Discrete metered fluid injection: Development of a robust autonomous reagent based optical chemical sensor

By

Conor Slater, BEng.

Thesis presented to Dublin City University

Under the Supervision of

Dr. Brian Corcoran

and Prof. Dermot Diamond

School of Mechanical and Manufacturing Engineering

September 2009

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ____________ (Candidate)
ID No.: ___________
Date: _______

ACKNOWLEDGEMENTS

It would have been impossible for me to tackle this project all by myself. There are a number of people who have given advice and support over the years and allowed me to complete my work.

First, I would like to thank Peter Leitner from BMW who taught me how to manage a large project such as this one. In hindsight, I don’t think I would have had the ability to make the phosphate analyser a success without his previous instruction.

I don’t know whether or not I should thank Bill Yerazunis for convincing me that doing research was a good idea. But he should definitely be acknowledged for his advice at the initial stages of my project. I should also acknowledge the other research staff at Mitsubishi Electric Research Laboratories for teaching me how to produce top quality electronics, in particular Paul Dietz (Circuit Design), Jonathan Westhues (PCB Layout) and Darren Leigh (Microcontroller Programming).

The greatest contributor to this project was Christina McGraw who’s great knowledge and experience of analytical chemistry allowed me to develop the design of the phosphate analyser. I would like to think she helped me avoid a lot of mistakes. For advice on the optical components of this project I would like to thank Brian Lawless for helping me out, even though I did show up randomly at his office. Furthermore I would like to thank Kim Lau for helping me organise this thesis and his unending patience throughout the project.

I would like to thank Jörg Kutter for allowing me to visit his group at DTU Nanotech where Detlef Snakenborg helped me to design and fabricate what I consider state of the art polymer Labs-on-Chips. My colleagues Cormac, Daniel and Jer who have helped me out with numerous parts of the project and have kept me company (and sane) throughout cannot go without mention.

I would like to acknowledge John Cleary for his help with the analytical chemistry throughout this project and his assistance setting up the validation of the sensor at Osberstown. I would also like to thank the personnel at Osberstown Waste Water

Treatment Plant, specifically Fergal Cronin and Fergal Humphries for accommodating the validation of the analyser.

I would also like to thank Dr. Dermot Brabazon of the School of Mechanical and Manufacturing Engineering and Dr. Brendan O'Flynn of the Tyndall National Institude for taking the time act as examiners for this thesis.

Finally, I would like to thank my supervisors Brian Corcoran and Dermot Diamond for the opportunity to do a Master's Degree at DCU.

This work has been funded by a number grant agencies and organisations. Science Foundation Ireland grant: 07/CE/I1147, "CLARITY: Centre for Sensor Web Technologies", Enterprise Ireland grant: IP/2008/544, "Innovation Partnership scheme - Autonomous Phosphate Analyser for Water Quality Monitoring" and Enterprise Ireland grant: CFTD/08/111, "Wireless Autonomous Nutrient Detector". The initial funding for the analyser was provided by Mitsubishi Electric Research Laboratories and I would like to thank Joe Marks for giving the financial freedom to make my work possible.

ABSTRACT

This work is based on the development of an autonomous sensor for monitoring phosphate in lake and river water in-situ. The goal was to design, build and validate an analyser that is sensitive, low-power, and, most importantly, robust.

To tackle these design challenges a new concept of discrete metered fluid injection was developed. This concept combines robust solenoid metering pumps with a microfluidic lab-on-a-chip. Separate pumps can inject a precise volume of reagent and sample into the lab-on-a-chip. The layout of the microchannels within the chip direct the flow from the pumps to achieve precise mixing and the subsequent presentation of the reacted sample to a spectrophotometer for an absorbance measurement. This concept is scalable allowing pumps for dispensing calibration and cleaning solutions to be integrated into the design.

To validate whether this concept meets the design challenges a prototype analyser was constructed. The sensitivity of the analyser was shown to be comparable to commercial phosphate monitors in laboratory trials. Also the analyser was shown to be robust enough for long term operation when directly validated against a phosphate monitor at a waste water treatment plant over a period of almost 40 days.

TABLE OF CONTENTS

NOMENCLATURE

A	Absorbance	-
ε	Absortivity	-
b	Path Length	cm
c	Concentration	mol × cm^{-3}
I	Intensity	cd
Re	Reynold Number	-
ρ	Density	kg × m^{-3}
η	Viscosity	Pa × s
Q	Volumetric flow rate	m^3 × s^{-1}
P	Pressure	Pa
D	Coefficient of Diffusion	-
V	Velocity	m × s^{-1}
C	Compliance	m × N^{-1}
v	Voltage	V
i	Current	A
r	Resistance	Ω
c	Capacitance	F
q	Charge	C

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This work describes the development of a portable automated chemical analyser for determining the concentration of phosphate in natural water (i.e. river and lake water). Although a number of automated analysers are currently available for measuring phosphate, they are not portable enough to be deployed at a waterway to measure the phosphate in real time. Research into micro Total Analysis Systems (µ-TAS) was applied to design and build the portable analyser.

Phosphate analysers are normally described as Reagent Based Optical Chemical Sensor (ROCS) as phosphate sensing is usually based on a colorimetric chemical reagent. The theory and technology, such as microfluidic and lab-on-a-chip, associated µ-TAS is also described to provide the basis from where the portable phosphate analyser was designed. Current technology for phosphate analysis is reviewed along with previous research to develop a µ-TAS for phosphate measurement.

All the technologies were compared to highlight where there was an opportunity develop an analyser based on microtechnology that has comparable performance to current macro-technology. The goal set was to complete a µ-TAS analyser for phosphate that has comparable performance to a commercially available macro fluidic analyser demonstrating that µ-TAS can be a competitive and potentially disruptive technology as the prototype system developed costs much less and is more portable. A disruptive technology is defined as: "An innovation that creates a new (and unexpected) market by applying a different set of values. (e.g., the lower priced Model T Ford)" (Christensen, 2003)

A top-down approach is taken for describing the phosphate analyser developed. The analyser's design is outlined schematically indicating how each of the components functioned in the design. The design of the fluidic system is described in detail as this was the most critical of part of the analyser and where it differed significantly from past work. It is then shown how the actual analyser was constructed and its theoretical operating characteristics are detailed. A description of the how the performance characteristics of the analyser were determined is given and the results are compared to current technology for phosphate analysis to show that the design was a significant improvement on past research.

CHAPTER 1. LITERATURE REVIEW

1.1 Background

Some scientists believe that the next development in internet technology will be a “seamless interface between the molecular and digital worlds”. It is envisaged that a vast number of sensors will collect data from the real world and deposit it on internet data bases. These databases will be accessed by all manner of internet users and systems. At this moment sensing systems are not small, reliable, or cheap enough to realise this vision (Diamond, 2004).

The MySound project has realised part of this vision by deploying a number of sensing buoys at Long Island sound, New York, USA. These buoys are scattered throughout the sound and transmit data using wireless telecommunications in real time to an internet database. The data can be accessed via a web page where the user can view the latest and historical data. At the moment the buoys used by the project are large, expensive and as a result are limited in number. Improvements in the size and cost of sensors would see larger project deployments (Tedesco, 2003).

In 1990 a ‘Miniaturised total Analysis System’ (µ-TAS) was proposed. These systems would be miniaturised laboratories that could be deployed where they would measure a particular analyte on a continuous basis. These µ-TAS would be designed to use the unique flow patterns of fluids in microchannels (Manz, 1990).

The main drive in the realisation of µ-TAS is fundamental research into the area of microfluidics, the study of fluid flow at the µm scale. The unique features of microfluidics, has led to the development of the ink jet printer (Bassous, 1977) and in recent years there has been an explosion of research in microfluidic flow and microfluidic devices (Kamholz, 2004). This includes the developments of micro-pumps, valves, mixers, etc. that are used in µ-TAS.

Most microfluidic systems are partly integrated into a single unit, the Lab-on-a-Chip (LOC). A Lab-on-a-Chip integrates a number of functions such as mixing, pumping, reaction chambers etc. into a single chip. The concept is similar to the development of the electronic integrated circuit (IC) in the late 1950's where a number of electronic devices are integrated into a single chip. It is believed that like the development of ICs over the past 50 years (Going from 2 to 30 million+ components on a single chip), LOCs will integrate more and more microfluidic devices into a single chip in the coming years (Geschke, 2004). LOCs have mostly been applied into realising µ-TAS but some unique devices have been developed such as a LOC for controlling insect behaviour by injecting specific drugs. Such an approach was used to create a Micro-Air-Vehicle by attaching a LOC to a moth (Chung, 2008).

Many integrated µ-TAS have appeared in the literature. Systems developed such as the analyser for detecting ricin report promising results. However the majority of detection methods are not integrated into complete systems. The complete µ-TAS pictured in the study by Meager et al. is no more than a mock up of what a µ-TAS might look like (Meager, 2008).

This and many other systems like it give the impression that µ-TAS is a highly developed field and new products are just around the corner. But there are still many problems that need to be addressed before this happens. Currently researchers are still talking about the "potential" of microfluidics; "[Microfluidics] can potentially enable portability, reduced reagent consumption, reduced analysis time, and increased efficiency" (Pennathur, 2008). There are also many technical problems such as getting a sample from the macro world and analyse it in the micro world (Mariella, 2008).

1.1.1 *In-situ* water quality monitoring

Phosphate (PO_4^{3-}) is the most common form of the essential plant and animal nutrient phosphorus. This nutrient is a limiting factor in the growth of most aquatic ecosystems. Eutrophication is the over enrichment of nutrients in natural water. This over abundance of nutrients causes water systems to become overgrown with plants and unsightly algal blooms .

These algal blooms are sometimes toxic and so can have a devastating impact on drinking water supplies. This excessive growth is damaging to the aquatic ecosystem as the dissolved oxygen normally available to aquatic animals is depleted which can lead to decimation of fish populations. Human activity is the main source of excessive phosphate and other nutrients. This includes the dumping of untreated and treated waste water from sewerage systems, agricultural sources such as fertilizers and animal waste, and waste from industrial processes.

The European Water Framework Directive (2000) requires member states to achieve specific water quality goals by 2015 (European Council). In order to do this more European waterways will have to be sampled and sampled more frequently. The manpower required for this task does not exist within regulatory agencies and so there is an incentive for the development of automated methods of water quality monitoring. The water quality goal for phosphate is 0.3 mg/L PO_4^{3-}

Phosphate is normally monitored manually by collecting and filtering natural water samples which are stored and later analyzed in a laboratory. The obvious limitations of these sampling methods are the manpower and time cost. Other implications are the limited benefit that “snap-shot” monitoring of a particular waterway’s condition will give. Studies have shown the need to understand the spatial and temporal condition of a waterway to assess risk or to classify a particular water system (Jordan, 2005, 2007).

A recent review of environmental sensing technology highlighted the need for a more diverse selection of portable systems for water quality monitoring (Bogue, 2008). Analyser’s for nutrients such as phosphate, nitrate and ammonia are limited to cabinet sized instruments that are not portable and require considerable infrastructure to operate. Indeed, the study conducted by Jordan et al. involved building a hut at a remote location and installing and maintaining a large phosphate analyser.

1.1.2 Reagent based optical chemical sensors

A reagent based optical sensor (ROCS) is a sensor that is based on the utilisation of a analyte selective reagent for detection. The reagent is a chemical that changes its optical properties on contact with the target analyte. This change in optical properties is measured as a change in absorbance or a change in fluorescence. This type of sensor has the advantage of having a selective and charaterisable response to the target analyte.

There are two forms of this type of sensor. One where the light is directed through a dialysis membrane that encloses the reagent to a optical detector. In this setup the behaviour of reagent is reversible and the reagent is not renewed. The other method pumps new reagent between the emitter and detector for each measurement. The latter method requires a more complicated system but yields more stable results (DeGrandpre, 1999).

1.1.3 Beer-Lambert law

With all absorbance measurements the concentration can be determined by the Beer-Lambert Law shown in Equation 1.1, where A is the absorbance of the sample, ε is the absorbtivity of the compound, b is the path length of the sample and c is the concentration. The absorbance can be determined using I_0 as the intensity of the light before reaching the sample and I as the intensity of the light after passing through the sample. These equations show that the factors that will affect the sensitivity of this technique are the absorbance path length and the resolution of the light intensity measurement.

$$A = \varepsilon bc$$

$$A = log_{10}\left(\frac{I_0}{I}\right)$$

Equation 1.1: Beer-Lambert Law

1.2 Micro Total Analysis Systems

The term ‘miniaturised total analysis system’ (μ-TAS) was coined by Manz et al. (1990). It refers to miniaturised devices that perform chemical analyses that are normally confined to a fully equipped chemical laboratory operated by trained technicians. Such devices are based on flow injection analysis, chromatography and electrophoresis. In addition self calibration can be integrated into the design with electronics for automating the analyser. μ-TAS are employed for the continuous monitoring of chemicals in the environment, chemical production and for biomedical applications.

The initial goal of μ-TAS was to use miniaturisation to enhance the selectivity and lifetime of chemical sensors. An advantage of miniaturisation was to reduce the volume of reagent and sample consumed by the analyser (Reyes, 2002,Auroux, 2002). It was also considered that completely integrated systems could be developed that integrated miniaturied pumps, mixers and detectors (Elwenspoek, 1994.)

The typical µ-TAS is usually based around a Lab-on-a-Chip such as the system developed by Sequeira et al. (2002). This system used a reagent for detecting ammonia. The reagent was mixed with the sample in a microfluidic Lab-on-a-Chip that integrated a micro mixer and reaction coil. The reagent and sample were pumped using a bench top syringe pump and the ammonia was detected by performing an absorbance measurement on the reacted sample and reagent. Another example is a system that used a reagent and an amperometric detection for detecting glucose (Lammertyn, 2006). Other variations include µ-TAS that use electro-osmotic flow to move fluid through the LOC (Greenway, 1999, Petsul, 1999).

Some µ-TAS perform assays using LOCs that move fluid by using a number of passive forces, such as gravity or capillary action (Weigl , 2008). This type of device is easily miniaturised as no pumps are required and it has been shown that disposable LOCs can be created. This area of µ-TAS has shown success in generating commercial products such as the GRAVI® - chip (diagnoSwiss, 2009).

1.2.1 Fluid flow in micro-channels

Although the scale of a micro channel is miniature the number of molecules present in the fluid passing through is still in the order of billions meaning that the well established laws of fluid dynamics scale down to the micro-scale (Geschke, 2004). Gravsen et al. summarised the main formulas governing flow in microchannels in a review of micro fluidics (1993). The main feature of microfluidics is that fluid flow is normally laminar. This can be confirmed by calculating the Reynolds number for a particular fluid in a micro channel. The Reynolds number is a dimensionless value that is the ratio of inertial forces to the viscous forces in the fluid. Equation 1.2 details the calculation of the Reynolds number *Re*, where ρ is density d is the width or depth of a channel, η is the viscosity of the fluid and V is the velocity of the fluid is travelling at.

$$Re = \frac{\rho d V}{\eta}$$

Equation 1.2: Calculation of dimensionless Reynolds number

For flow to be laminar the Reynolds number has to be below 2300. For this to be the case the velocity would have to be low, the viscosity high or the value of *d* would have to be small. In microchannels the value of *d* is sufficiently small for the flow to be laminar, only in cases where fluids are of low viscosity (e.g. Air) or fluids which are travelling at exceptionally high speed does the flow become turbulent.

Fluid flow through a microchannel can be described using the Hagen-Poiseulle flow equation. This relates the volumetric flow rate Q through a channel to the fluidic resistance and the pressure difference between both ends of the channel ΔP Equation 1.3.

$$Q = \frac{\pi R^4}{8\eta L}\Delta P$$

Equation 1.3: Volumetric flow rate through a circular cross-sectional capillary

The fluid resistance is dependent on the length of the channel L, the viscosity of the fluid and the cross sectional shape of the channel. Equation 1.4 shows how to calculate the fluidic resistance of a channel of a circular cross section where R is the radius of channel cross section. The calculation for a square cross section is shown in Equation 1.5 where a is the length of the side of the channel.

$$\frac{8\eta L}{\pi R^4}$$

Equation 1.4: Fluidic resistance in circular cross-sectional channel

$$\frac{28.454\eta L}{a^4}$$

Equation 1.5: Fluidic resistance in square cross-sectional channel

1.2.2 Mixing in micro-channels

Since all flow in a microfluidic mixer is laminar the only mode of mixing liquids is through diffusion. The only options available to increase the efficiency of the mixing process is to maximise the interfacial area between the liquids to be mixed or to decrease the diffusion path as the diffusion coefficient and the concentration of the liquid are fixed values (Aoki, 2006).

$$x = \sqrt{2Dt}$$

Equation 1.6: Einstein-Smoluchowski relation

$$t = \frac{x^2}{2D}$$

Equation 1.7: Time to complete mixing

Diffusion happens by Brownian motion, the random movements of particles within a medium. The time it takes for a solution to mix completely by diffusion is governed by the Einstein-Smoluchowski relation. It is used to calculate the distance *x* a particle with a diffusion constant *D* will travel in the time *t* by Brownian motion. This relationship is used to calculate the time it will take to mix two solutions in a channel of given dimensions (see Equation 1.6 and Equation 1.7). In the equation as the dimension of *x* decreases the value of *t* will decrease at an exponential rate. This is how fluids can mix rapidly in microchannels

Figure 1.1 shows two fluids being mixed in a T-mixer. Both fluids are travelling at the same volumetric flow rate. The distance x is half the width of the channel leading from the T-mixer. When both fluids meet at the T junction they flow in parallel as the flow is laminar. Both fluids diffuse into each other resulting in a homogenous solution after a period of time. The length the fluid travels before it is mixed is determined by the velocity of the fluid leaving the mixer multiplied by the time it takes the two fluids to mix by diffusion.

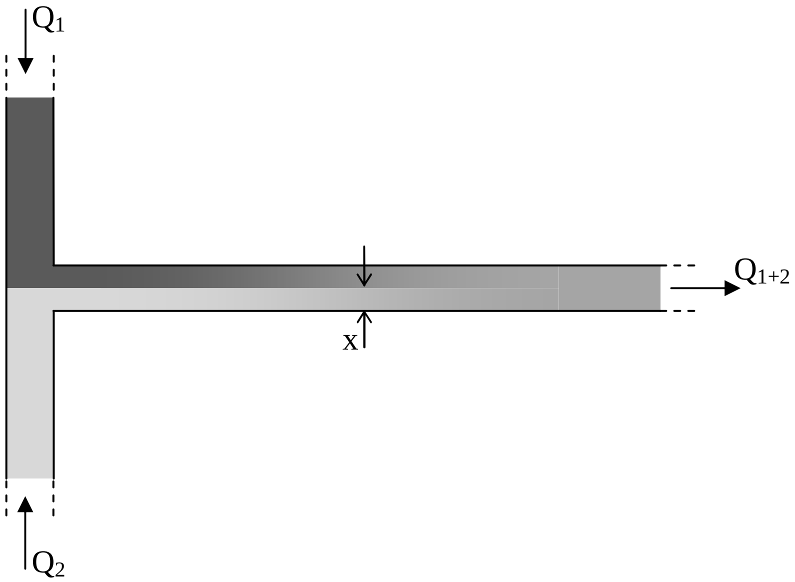


Figure 1.1: Molecular diffusion in a microfluidic T-mixer

The Peclét number relates the velocity of the fluid flow V to the diameter of the channel *d* and the coefficient of diffusion *D* (see Equation 1.8). This number is used as a representation of the efficiency of the mixer and is normally used for comparison (Hessel, 2004).

$$Pe = \frac{Vd}{D}$$

Equation 1.8: Calculation of Peclét number

The collision of two liquids at a T, Y or V junction is the most common method of mixing currently used by researchers due to that that they can be fabricated on a single chip layer. Many studies have been carried out on how flow rate and the channel geometry affect the mixing efficiency. The conclusions of these studies are that the sharper the collision angle the more efficient the mixing and this effect increases with flow rate. The study of these simple mixers have led to the development of extremely efficient mixers that colloid many fluid flows at different angles (Nagasawa, 2005). Also mixers that split the flow into many thin lamine and recombine them are very efficient as they reduce the value of *x* (Hessel, 2003, Hardt, 2003).

The mixing ratio between two fluids in a micro mixer is determined by the ratio of flow rate between the streams. If the flow rate of the two streams is identical the fluids will mix at a 1:1 ratio regardless of the absolute value of the flow rate (Takabayashi, 2008). If the volumetric flow rate of one stream is greater than the other the resulting mixed fluid will have a higher concentration of the fluid with the greater flow rate.

1.2.3 Modelling of microfluidic systems using equivalent electronic circuits

Given that there is no method to accurately describe the behaviour of microfluidic systems using a schematic drawing, a growing number of researchers borrow the symbols from electronic schematics. The basis for this is the analogy between Ohm's law and the Hagen-Poiseulle's equation (see Figure 1.2).

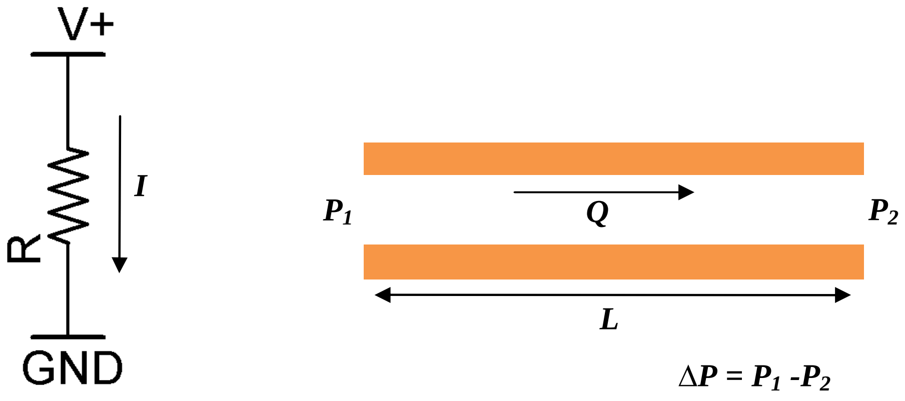


Figure 1.2: Comparison of resistive circuit and flow of fluid through a channel

In the analogy the pressure and voltage are analogous to each other, as is fluidic resistance and electric resistance, as well as volumetric flow rate and electric current. Since the relationship between each of variables is the same the analogy holds (see Equation 1.9 and Equation 1.10)

$$v = ir$$

Equation 1.9: Ohm's law

$$\Delta P = Q\,\frac{8\eta L}{\pi R^4}$$

Equation 1.10: Hagen-Poiseuille's equation

This method has been used by a number of researchers to design or describe microfluidic systems. Such as a flow focusing systems (Knight, 1998, Lee, 2001), systems for simultaneously generating a multitude of concentrations of fluids (Yamada, 2005, Sun, 2008) and micro pumps (Shen, 2008)

1.2.4 Pumping Technology

A large number of different types of pumps have been used with µTAS, these are usually bench top units such as peristaltic pumps, syringe pumps, piston pumps and pneumatic pumps (Pennathur, 2008). These pumps offer a wide range of performances and are suitable for microfluidic applications.

However most of these systems have high power consumption and are not compact enough for integration into a microfluidic chip and so they remain external to the system. Another method of pumping takes advantage of electro-osmotic flow (Haswell, 1997, Brask, 2006). Ions will flow through a conductive medium with a high potential difference across it (>1000V). The size limitation of the high voltage transformers limit the integration of this form of pumping into chips.

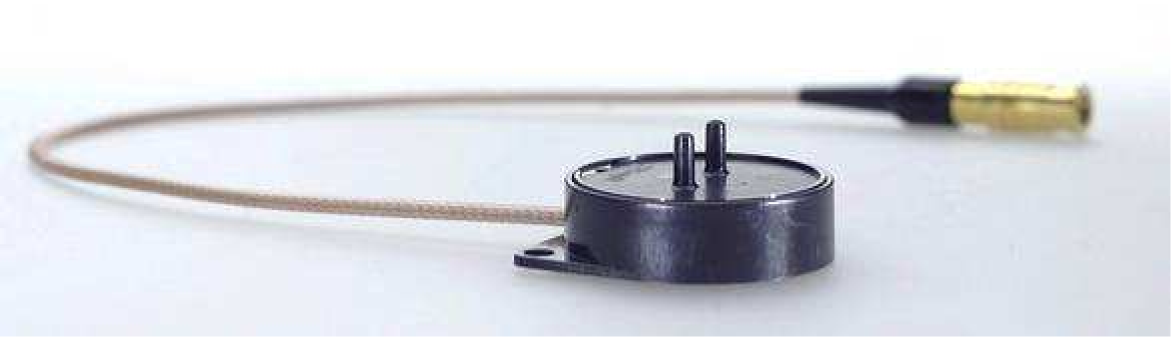


Figure 1.3: MDP1304 piezo electric actuated micropump from ThinXXS

Micro-pumps have been under development for over 25 years (Laser, 2004). Most of these pumps operate by reciprocation of a piston or diaphragm with a wide array technologies used for actuating them, including electro magnetism, piezo electrics and hydro gels (Nguyen, 2002). The direction of flow through the pump is normally rectified by a pair of micro check valves placed at the inlet and outlet of the pump. Although many designs of pump have been developed most are not reliable enough to be used in µTAS. Only one type of micro pump is commercially available at this time (see Figure 1.3).

1.2.5 Lab-on-a-Chip

A Lab-on-a-Chip (LOC) is the integration of a number of laboratory functions into a single chip such as the mixers and pumps described in this chapter. The development and role of the LOC is analogous to that of the integrated circuit.

In the early part of the 20th century there was practical limit to the complexity of electronic devices produced. It became increasingly difficult to reliably manufacture systems made up of many components. The more components in the device the less reliable it was. This problem was called the 'Tyranny of numbers' (Thorsen, 2002).

This problem was eventually solved with the development of the integrated circuit (IC) in the late 1950s. In an IC a large number of electronic devices are integrated into a single chip. It is this integration that allowed more and more complex electronic devices to be developed in subsequent years.

The development of the LOC is driven by the development of microfabrication technology. As the technology improves more and more functions can be integrated into a single chip. Early examples are hybrid systems where some functions such as mixing and detection are integrated into the LOC and pumps and valves remain external to the chip (Tamanaha, 2002). Some researchers have been able integrate all functions required to perform a bioassay into a single chip although the complexity of the chip made it difficult to manufacture (Liu, 2004). More recent LOCs have been able to integrate microvalves directly in to the chip (Hulme, 2009). Some chips have complex 3 dimensional layouts owing to their layered designs (Flachsbart, 2006, Morimoto, 2008)

LOCs are made from a large variety of materials. The most popular being silicon, glass, polycarbonate (PC), cyclic olefin copolymer (COC), polymethylmethacrylate (PMMA) and polydimethylsiloxane (PDMS) (Piruska, 2005). Materials are chosen depending on the application, for example PMMA has good optical properties where as COC has good biocompatibility characteristics (Yi, 2008). Chips made from low temperature co-fired ceramics have also appeared in the literature (Golonka, 2005)

Most recently, methods for rapid prototyping LOCs have been developed. This involves the assembly of prefabricated microfluidic blocks on an alignment board (Rhee, 2008), or using prefabricated microfluidic stickers that can be stacked to create an LOC (Bartolo, 2008). There is also a type of programmable LOC where a photopresponsive hydrogel can be used to create specific channel geometries by patterning with UV light (Sugiura, 2009).

1.3 Current technology and methods for phosphate detection

There are a number of commercially available and experimental technologies for monitoring phosphate. The term “Analyser” is used to describe an entire sensing system. Commercially available analysers are automated systems that are reagent based optical chemical sensors. A portable manually operated analyser is also presented with an experimental LOC based phosphate analyser. The performance of each of the analysers are summarised at the end of the chapter.

Some analyser’s quote their performance in terms of total phosphorus within the phosphate (mg/L P-PO_4^{3-}). This is done by calculating the mass of phosphorus in the phosphate sample. Phosphate is composed of 4 oxygen atoms and one phosphorus atom. Phosphorus has an atomic mass of roughly 31 and oxygen has an atomic mass of roughly 16. This gives phosphate an atomic mass of 95 ((16 × 4) + 31). The phosphorus in the phosphate makes up 33% of the atomic weight (31/95 = 0.3333). To calculate the total phosphorus in solution of phosphate the concentration of the phosphate is multiplied by 0.33. To make a comparison all analyser’s discussed here will have their characteristic quoted in terms of phosphate (PO_4^{3-})

1.3.1 Commercial Systems

For phosphate monitoring there are a number of commercially available systems. Both low end and high end versions are presented here along with a portable hand held phosphate analyser.

The AZTEC P100 (Severn-Trent, UK), (see Appendix A) is a low end online phosphate monitor. It is an automated system based on the molybdenum blue method of phosphate detection. Its features include sample digestion, two-point calibration, self cleaning and self diagnostics. The analyser uses a single piston pump and an array of valves for moving liquid around the system (see Figure 1.4). Two mixing units are used for mixing acid with the sample to digest it and for mixing the molybdenum blue reagent with the digested sample. The colour of the reacted sample is measured by a lamp, a 690 nm optical filter and a photodiode. The analyser is controlled and the data collected is logged by an on board microprocessor. A sample rate of 1 to 6 samples per hour can be set and the analyser has a limit of detection of 0.3 mg/L PO_4^{3-} and a detection range of 0.3 to 7.5 mg/L PO_4^{3-}. The analyser can be adapted to give a detection range of 0.9 to 22.5 mg/L PO_4^{3-}. The installation of one of these analysers costs about €70,000.

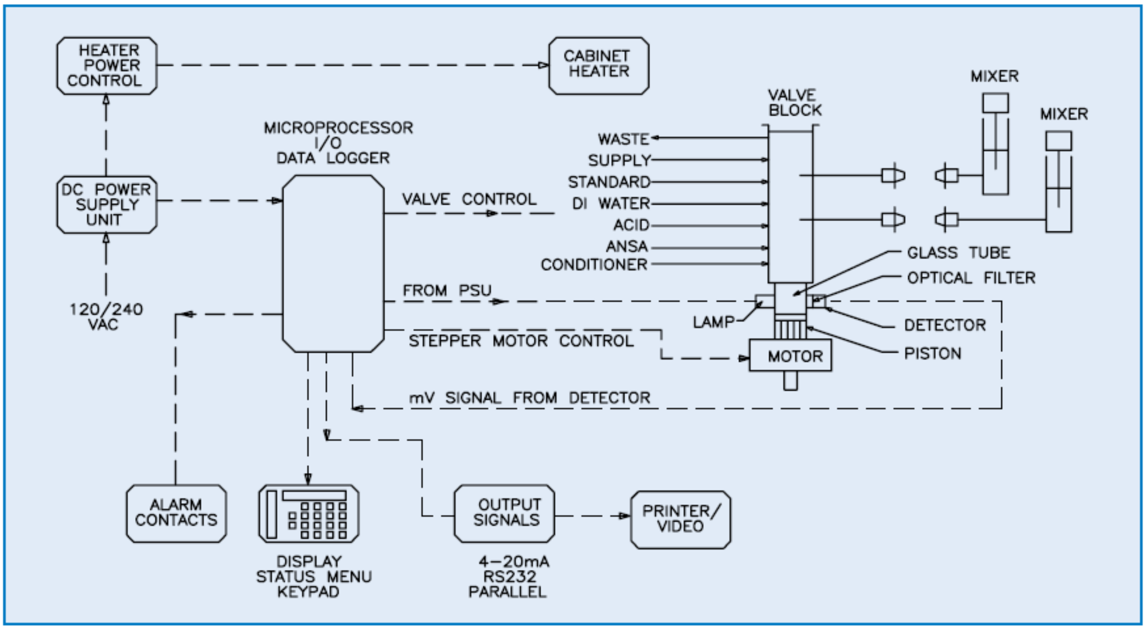


Figure 1.4: Schematic of AZTEC P100 (Appendix A)

A high end phosphate monitor is the TresCon OP210 (WTW, Germany), (see Appendix A). It is of similar design to the Aztec P100 as it is automated and has a two-point calibration routine. It is based on the vanadomolybdenumphosphoric acid method of phosphate detection. It is a more advanced system as it has a lower limit of detection of 0.15 mg/L PO_4^{3-} compared to the AZTEC P100. The detection range is also greater at 0.15 to 9 mg/L PO_4^{3-}**.**

The two systems described above are not portable and require mains electricity to operate so cannot be easily employed for measuring phosphate at a remote location. For measuring phosphate at remote locations a hand held colorimeter can be used (see Figure 1.5). Samples are manually collected and mixed with a reagent in a cuvette before being placed in the colorimeter for the analysis. The colorimeter is calibrated by measuring a blank before analysing the reacted sample. The Pocket Colorimeter II (Hach-Lange, Germany) (see Appendix A) is portable battery powered and can be adapted for a wide range of analytes. The version for measuring phosphate has a limit of detection of 0.15 mg/L and a range of 0.15 PO_4^{3-} to 4.5 mg/L PO_4^{3-}.

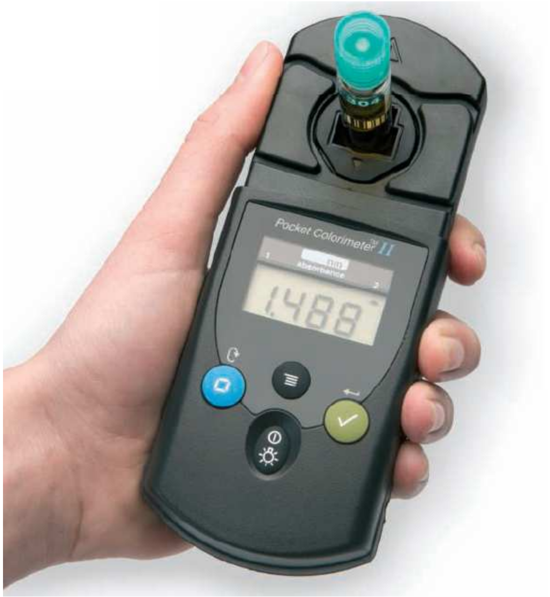


Figure 1.5: Hand held colorimeter (Pocket Colorimeter II, Hach-Lange, Germany)

1.3.2 Lab-on-a-Chip based phosphate analyser

The development of the LOC based phosphate analyser started with the work of Bowden et al. (2002). A flow injection analysis system that utilised the vanadomolybdenum phosphoric method of phosphate detection (Yellow method) was developed. The system used a bench top syringe pump to inject reagent and sample in to a Lab-on-a-Chip. The colour of reacted sample was measured by using a UV LED and bench top spectrophotometer. It was shown that reagent was suitable to be used in conjuction with microchannels. The LOC system was used to successfully detect phosphate in both phosphate standards and real river water samples. It was later demonstrated that the Yellow method was suitable for a field deployable instrument as the reagent used had a lifetime of up to a year (Bowden, 2003).

The portable phosphate analyser reported by McGraw et al. (2007) was an autonomous system based on a Lab-on-a-Chip and the yellow method of phosphate detection. It contained all the fluid handling and electronic components necessary to work without assistance for a period of 7 days at a sample rate of every 15 minutes. The analyser was split between two polycarbonate enclosures. The electronics for control and communications were contained in one enclosure. The fluidics and LED-Photodiode spectrophotometer were contained the other enclosure. An 8 meter communication cable connected the two parts of the analyser (see Figure 1.6).

The reagent and waste were stored in polyethylene bags and sample water was drawn through a 0.45 µm pore size filter positioned on the outside of the enclosure. A miniature dual channel peristaltic pump was used for pumping liquid through the fluidics, one pump channel for the reagent and the other for the sample water. For each measurement the pump was switched on for 10 minutes and pumped 20 µL of both reagent and sample.

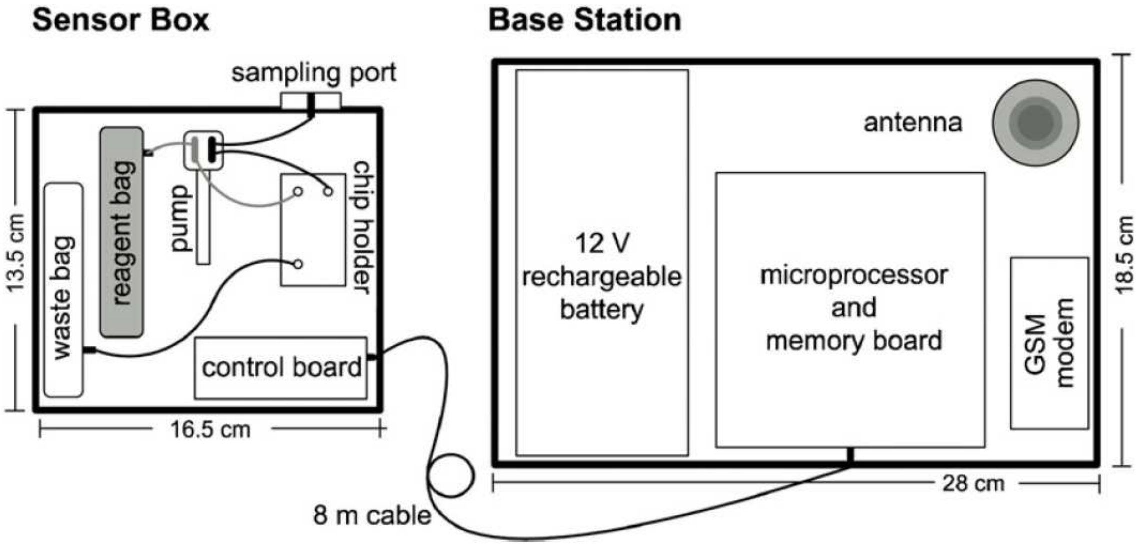


Figure 1.6: Layout of Lab-on-a-Chip phosphate analyser (McGraw, 2007, Cleary, 2008)

The lab-on-a-chip mixed the reagent and sample using a T-mixer (see Figure 1.7). After the reagent and sample reaches the optical window the flow is stopped and the reagent is allowed to react with the sample for 5 minutes. The optical window consists of a square cross-section channel of 200 × 200 µm. The channel's serpentine route winds between the LED and Photodiode in the spectrophotometer, this was done to maximise the area the light passes through. This resulted in a 200 µm path length for the absorbance measurement.

The analyser's electronics controlled the pump and took measurements from the LED-Photodiode spectrophotometer. It was also capable of storing up to 40 measurements with its internal memory. The GSM modem was used for communicating data. In operation the user would make a data call to the analyser using a computer equipped with a modem. Data could then be downloaded from the analyser's memory. A 12 V, 7 Ah lead acid battery was used to power the analyser.

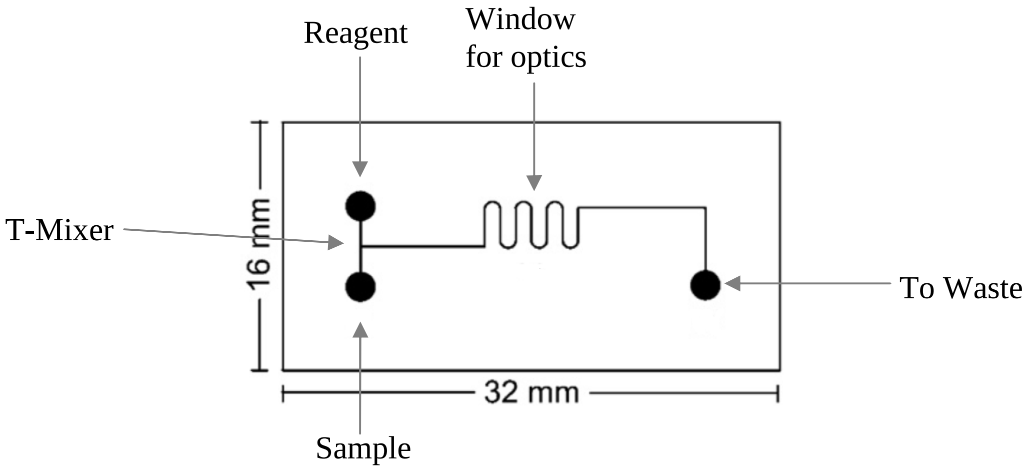


Figure 1.7: Layout of Lab-on-a-Chip

The analyser demonstrated a limit of detection of 0.9 mg/L PO_4^{3-} and a linear range of 0.9 to 60 mg/L PO_4^{3-}. However it was shown that the analyser's output drifted over time causing the limit of detection to increase to 4.5 mg/L PO_4^{3-}. This drift in was attributed to the staining of the optical window of the LOC by the reagent.

To reduce the drift in the LOC analyser Cleary et al. (2008) added a valve for switching between the sample and a blank solution. The valve was switched for every second measurement allowing a blank measurement to be made. This blank measurement was used to compensate for drift in the analyser. Figure 1.8 shows how the valve is integrated into the analyser.

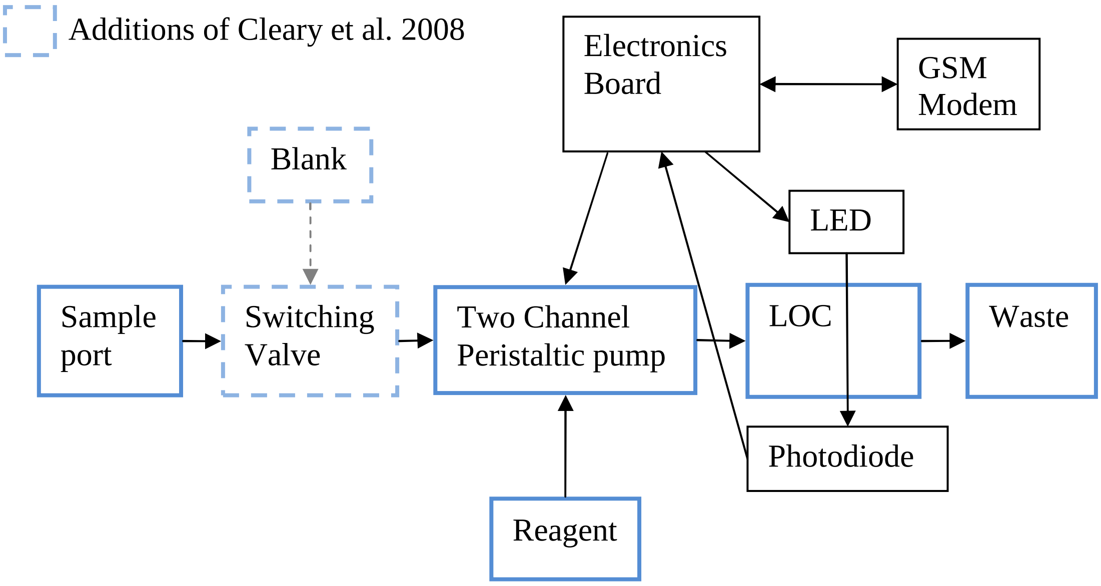


Figure 1.8: Schematic of Lab-on-a-Chip phosphate analyser (McGraw, 2007, Cleary, 2008)

The modified LOC analyser (Sensor) was validated against an AZTEC P100 phosphate analyser (Plant monitor). The results from the validation are shown in Figure 1.9. It can be seen from the data that although there is good correlation between both analysers, near the end of the plot there is a deviation. This demonstrates that the integration of the blank was not sufficient to completely eliminate drift in the LOC analyser’s output. It was suggested that a better calibration routine would be required to improve the data.

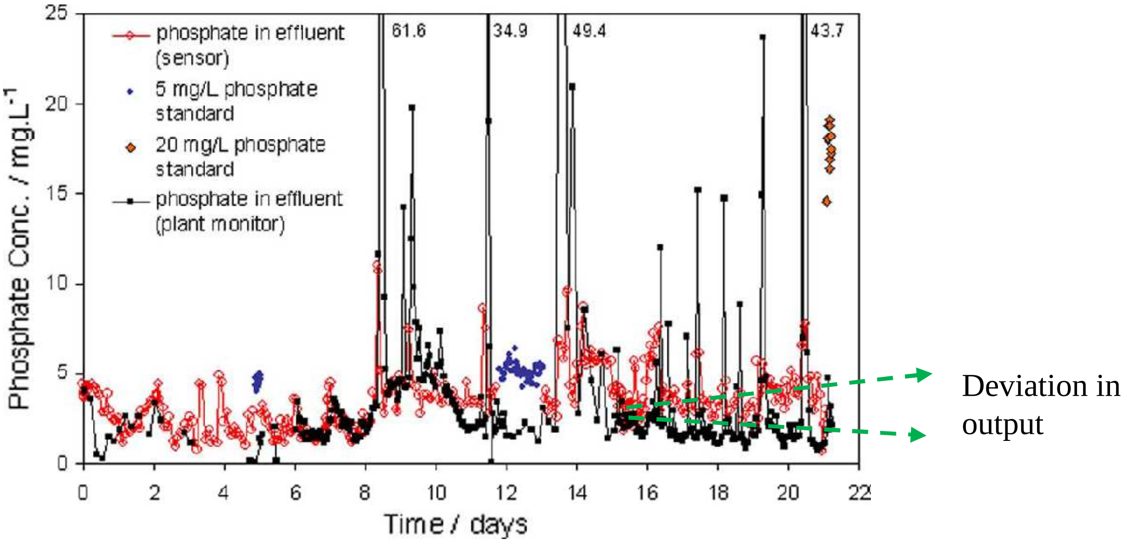


Figure 1.9: Data from 21 day validation at waste water treatment plant

In the plot there as a number of spikes in the data output of the plant monitor that do not appear on the plot of the LOC analyser. These were shown to be caused by the fine filter on the LOC analyser excluding larger particles. The plant monitor had a coarser filter and also included a digestion step in the phosphate measurement, so was able to measure the phosphate in samples with high amounts of particulate.

1.3.3 Comparison of phosphate analysers

The four phosphate analysers described in this chapter are summarised in Table 1.1. The AZTEC and TresCon analysers represent a low end and high end automated phosphate monitors respectively. The Pocket Colorimeter II is a portable handheld instrument that is operated manually. The LOC analyser can be described as an instrument that combines the automation of the Aztec and TresCon analysers with the portability of the Pocket Colorimeter II. What is clear from the comparison is that the detection performance of the LOC analyser needs to be improved in order to compete with commercial analysers.

Make	Developer	Detection Range (mg/L PO_4^{3-})	Comments
LOC Analyser	McGraw/Cleary	0.9 - 60	Portable, Automated
AZTEC P100	Severn-Trent	0.3 - 7.5	Automated
TresCon OP210	WTW	0.15 - 9	Automated
Pocket Colorimeter II	Hach-Lange	0.15 - 4.5	Portable

Table 1.1: Comparison of commercial and experimental phosphate analysers

The work of McGraw and Cleary represent steps towards a self contained portable analyser. It is clear that improvement in the sensors accuracy (Drift was present) precision (Limit of detection of 0.9 was not satisfactory) and lifetime (Analyser trial was only 21 days and lifetime of battery was 7 days) should be made to make the LOC analyser a viable alternative to current commercial technology.

1.4 Objectives

Having reviewed the current state of the art in self contained, automated µ-TAS for phosphate, performance and reliability is an issue for miniaturised analysers. But the compact size and low power consumption confer advantages in terms of deployment. In the case of commercial analysers the size and power consumption limits the number of places they can be deployed.

The goal of this work was to develop an autonomous portable analyser that was self contained, maintenance free and capable of maintaining the quality of the data output over its lifetime. The analyser developed was to have performance characteristics similar to low end commercial monitors. The purpose of developing an autonomous portable analyser was to provide a replacement for the current technology for phosphate monitoring that is cheaper and is easier to install.

The phosphate analyser was developed from the ground up rather than attempting to improve the design of McGraw and Cleary (2007, 2008). A key factor in its design was the use of microfluidics to reduce the amount of chemicals consumed per phosphate assay. Reliability of the analyser was based on limiting the number of moving parts, thus reducing the modes of failure. Also the power consumption of the components selected was considered in order to maximise the operational lifetime of the analyser.

The method of detection was the vandomolybdenumphosphoric acid ‘Yellow’ method of phosphate detection. This has been shown to work with LED-photodiode spectrophotometers (McGraw, 2007, Cleary, 2008). This method is ideal for automated systems as the reagent used has been shown to have a long lifetime (Bowden, 2003).

To demonstrate that the design of the analyser developed had improved equivalent performance and robustness, it was compared directly to a commercially available phosphate monitor installed at a waste water treatment plant to demonstrate that the miniaturised analyser could be used to substitute current technology.

Chapter 2 and Chapter 3 describe the design of the prototype phosphate analyser in detail as there were a number of system components. The entire design is summarised here. The layout of the analyser developed is shown in Figure 1.10. The analyser included bottles for storing the reagent, calibration solutions and cleaning solution; a sample port for collecting the water sample to be analysed; an array of solenoid pumps were used for pumping the required liquids through the microfluidic lab-on-a-chip (LOC). The LOC had the function of mixing the reagent and sample. It also presented the reacted sample to a photodiode and LED for an absorbance measurement. The analysed sample was then pumped to the waste storage. All of the fluid handling and analytical components were controlled by a microcontroller unit (MCU) that also performs the data acquisition and stores the data in a flash memory unit. A GSM modem was used to communicate the data via the SMS protocol to a laptop computer.

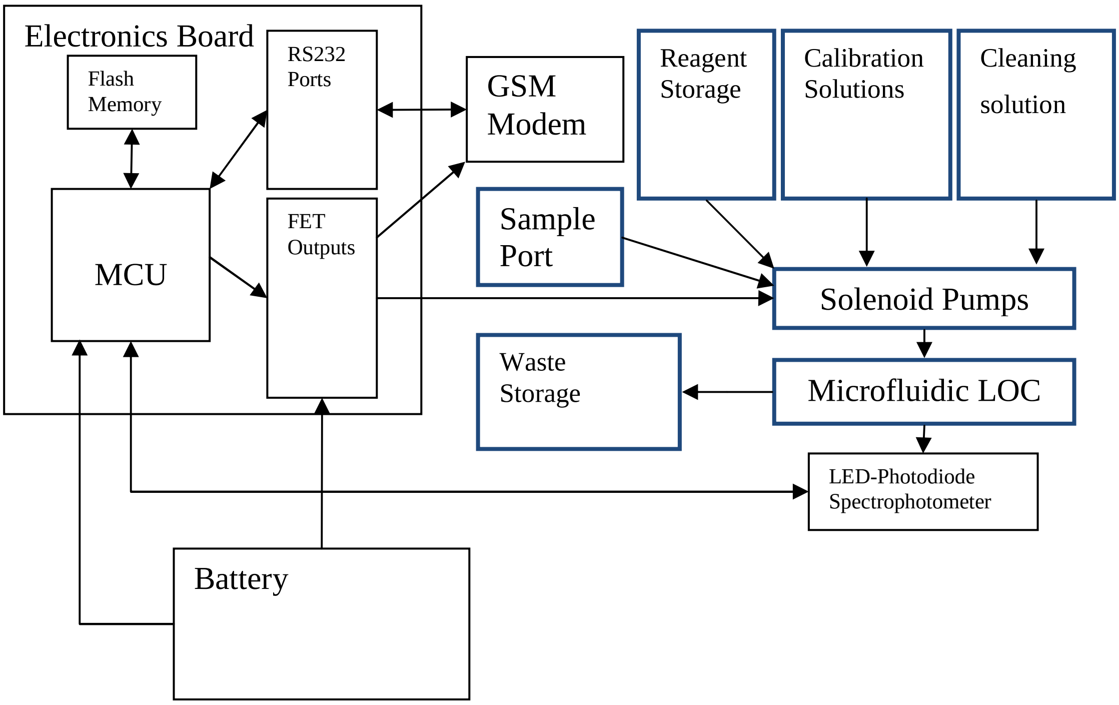


Figure 1.10: Schematic of LOC phosphate analyser

The difference between the design of this phosphate analyser and previous analysers was the selection of solenoid metering pumps for moving liquid in the system. This type of pump was chosen for its high reliability and simple design. Peristaltic pumps used in previous designs were problematic over a long period of time as the frictional forces applied to the tubing caused them to wear out.

It was shown how a solenoid pump, that does not deliver flow on a continuous basis but in discrete doses, can be employed in a suitable fluidic system integrating a microfluidic Lab-on-a-Chip. The LOC was fabricated using up to date fabrication methods for manufacturing chips with 3 dimensional layouts. The design included an LED-Photodiode spectrophotometer as this had been used successfully in conjunction with the ‘Yellow’ reagent and microfluidic analysers.

The analyser was controlled by a low power microcontroller unit (MCU). Integrated with the MCU was flash memory for storing data collected. Data could also communicate using a GSM modem. This type of communication was used as the infrastructure required for it to operate in available in most areas in Ireland. To make the design portable low power components were selected to so the analyser could run off a battery for a reasonable amount of time. A fused deposition modelling (FDM) 3D printer was used to manufacture parts to assemble the analyser. A robust enclosure was used to contain the whole analyser and protect it for its lifetime.

In operation the analyser mixes the reagent and sample water at a ratio of 1:1 and moves the reacted solution to an optical cuvette. The LED-Photodiode spectrophotometer then measures the absorbance. A 2-point calibration was carried out by mixing a blank and then phosphate standard with the reagent. The absorbance of the two solutions was measured to compensate for drift in the sensor. All data was time stamped and stored in the flash memory chip. The data collected was also periodically transmitted using the GSM modem.

Objective: Design and build specifications for the portable analyser for phosphate in this work were that it shall be:

- equivalent to low end commercial phosphate analysers in terms of limit of detection range of detection and sample frequency. The commercial analyser used as a design guide is the Aztec P100 (Severn Trent, UK). The detection range of this analyser was 0.3 – 7.5 mg/l PO_4^{3-} and had a maximum sample rate of 6 samples per hour.
- cheaper than low end commercial phosphate analysers. The Aztec P100 costs around €70,000 to install.
- easier to install than low end commercial phosphate analysers. The portable analyser should require a limited infrastructure in order for it to operate.

CHAPTER 2. DESIGN APPROACH FOR FLUIDICS

The design of the analyser included solenoid metering pumps. When actuated these pumps dispense a set amount of liquid. They are a proven technology and can have long life times, for example the pumps tested in this chapter have a lifetime of up to 20 million cycles (120SP, BioChem Valve, USA), (see Appendix A).

In this chapter an equivalent electronic circuit of a solenoid metering pump based on experimental results is presented. This circuit is used to demonstrate how mixing can be performed using a T-mixer, represented by a resistor network. Finally the complete concept for the phosphate analyser is presented. This design integrated an LED-photodiode spectrophotometer.

The final system design is presented schematically using an equivalent electronic circuit. The purpose was to show that the operation of a solenoid metering pump can be used efficiently by using theory of flow in microchannels. Since the presented system is theoretical it is not directly compared to a typical stop flow system.

2.1 Discrete metered fluid injection

The analyser was based on the injection of a precise amount of fluid into a network of micro-channels. Before the system was designed the behaviour of a metering pump was characterised. The pressure profile of the selected discharging pump was measured and a mathematical model was developed. This model was used to create an equivalent electronic circuit to describe the behaviour of the pump.

Since the fluid dispensed by these pumps is at a set volume no flow measuring technology was needed to determine the volume of fluid pumped. This is of great importance to microfluidic systems as flow in microchannels requires the integration of sophisticated instruments to measure flow on the µL scale. Another advantage is ease of control. The metering pumps are operated by supplying a voltage across the solenoid for a short period of time (~1 second). The majority of micropump technologies focus on pumps that supply continuous flow and are operated by supplying high frequency signals, in some cases at a high voltage (Amirouch, 2009).

2.1.1 Pressure profile of metering pumps

The pump selected for the experiment was the 120SP Micro Pump (Biochem Valve, USA). These pumps have size, power consumption, small volume dispensing and high reliability characteristics that make them suitable for miniaturised analysers.

This pump is based on the displacement of a VITON® diaphragm by a 12 V solenoid actuator. The flow through the pump is rectified by check valves, one positioned at the inlet and the other positioned at the outlet. This pump dispenses 20 µL for a complete stroke. According to the data sheet the maximum output pressure of the pump is 0.35 bar. However, no data on the precision or repeatability of this parameter is given.

Figure 2.1 shows a diagram of a solenoid metering pump. The pump is actuated by a solenoid actuator and a return spring. There is a plunger attached to the iron core of the solenoid and when this is pulled it creates suction that pulls the diaphragm. This happens when the solenoid is energised. When the diaphragm moves out, the volume in the pump chamber increases as fluid flows into the pumps chamber. When the solenoid off, the iron core and plunger are pushed by the return spring. The diaphragm is compressed pushing the liquid out of the pump chamber.

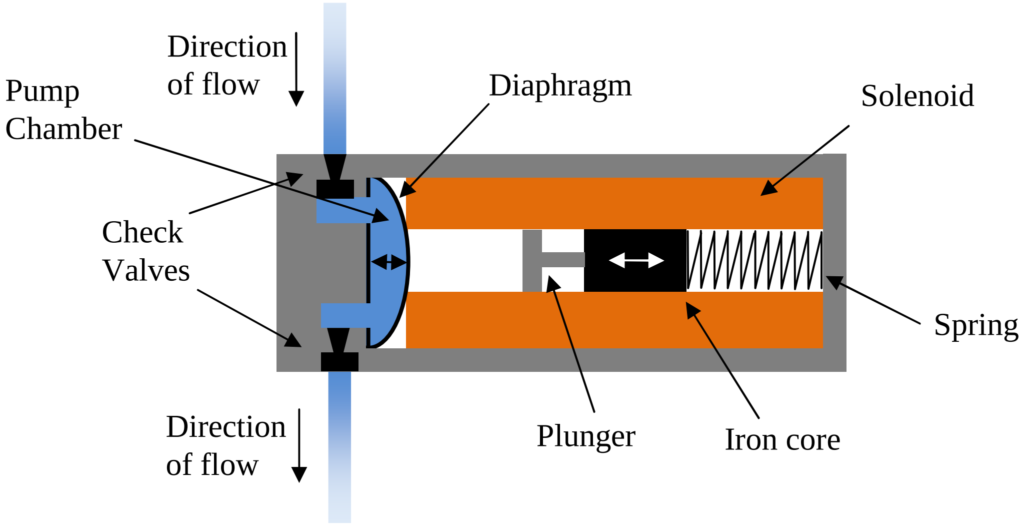


Figure 2.1: Solenoid metering pump

An experiment was run to acquire data detailing the pressure output of a solenoid metering pump during the return stroke. This information was needed to determine how fluid will flow when it is injected into a microchannel. The variation in pressure between different pumps of the same type and the repeatability of this pressure was measured. This information was useful when evaluating the final analyser.

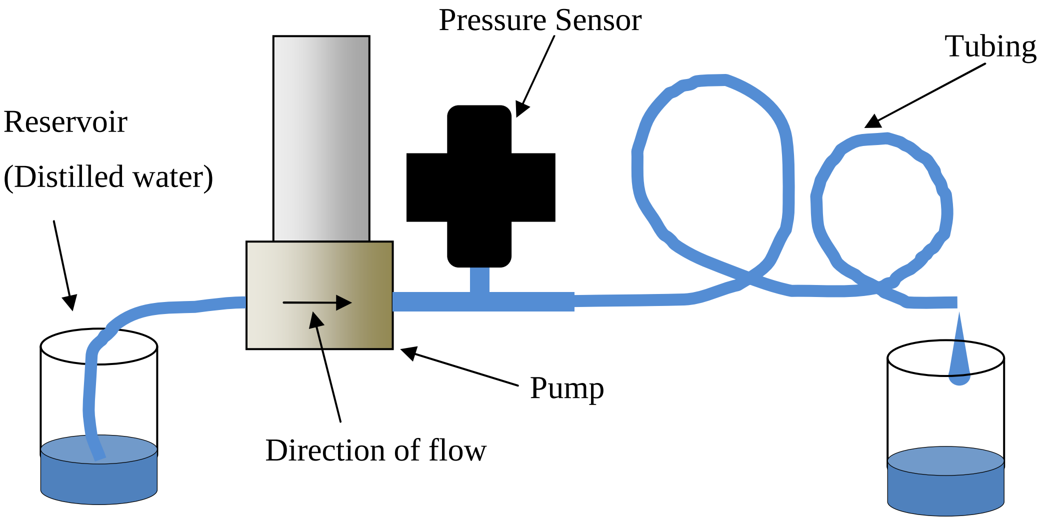


Figure 2.2: Experimental setup for measuring the pressure output of a 120SP micro pump

To measure the output pressure of the pump a piezo resistive differential pressure sensor (PK 80083, Honeywell, USA) was used. One port on the pressure sensor was connected to the outlet of the pump with a T-connector and the other was left open to the atmosphere. The open port on the T-connector was connected to a 20 cm length of TEFLON® tubing (Omnifit, UK). The internal diameter of the tubing was 0.8mm. The experimental setup is shown in Figure 2.2.

The output of the pressure sensor was measured by a CompactDAQ (National Instruments, USA) with a C series analogue input module (National Instruments, USA). A LabVIEW program was written to record the measured data to a Microsoft Excel file at a sample rate of 500 Hz.

Distilled water was used in the experiments. The pump was actuated by supplying a 12 V square pulse of 1000 ms. At the end of the pulse the pump began to dispense liquid.

Three separate pumps were tested. The pumps were never used before the experiment. Each pump was actuated three times and the pressure profile recorded. Point for point the mean of the measured pressure was calculated to give an average result for each of the pumps. The results for the three pumps are detailed in Figure 2.3.

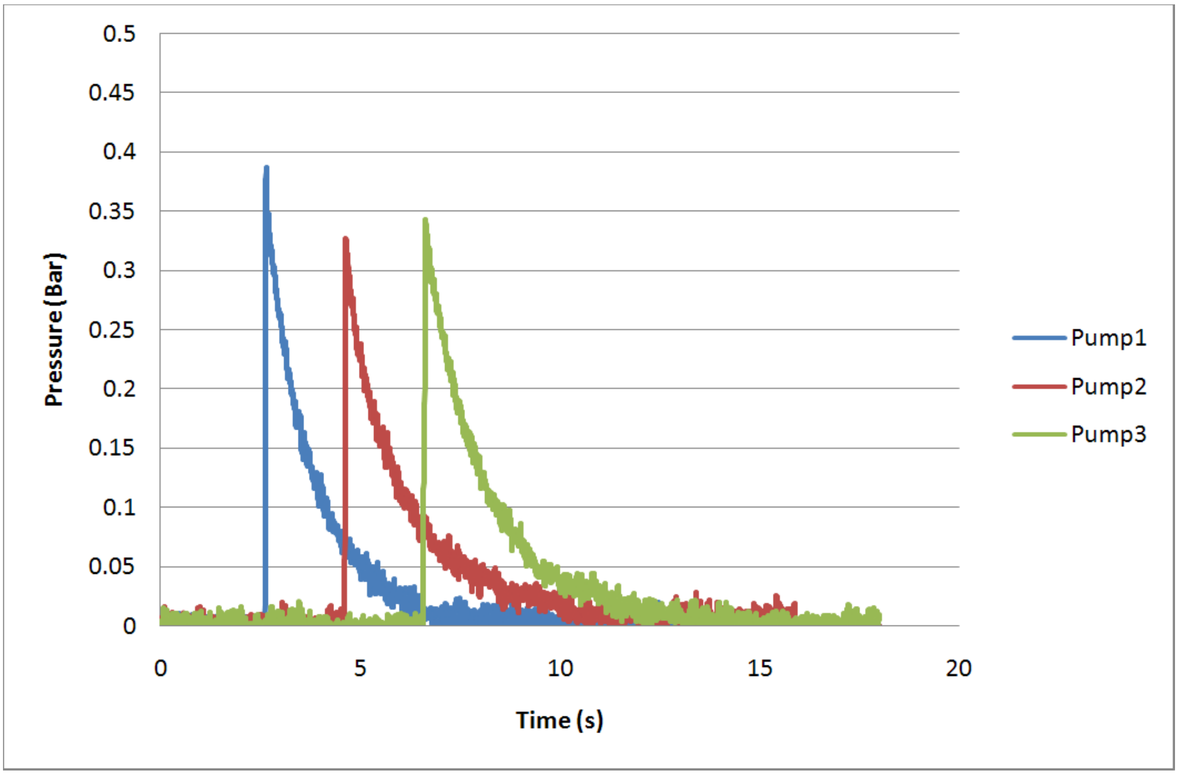


Figure 2.3: Pressure profile of metering pumps discharging fluid

The results from the pumps show a decay from the maximum pressure to zero. This exponential decay is analogous to the discharge of a capacitor in a Resistor-Capacitor circuit. This interpretation agrees with the model of a diaphragm pump developed by Shen et al. (2008). There was a variation in the maximum pressure between the pumps. This may be due to differences in the pumps manufacture. The affect of this variation in maximum pressure on the complete system using these pumps is shown in Chapter 4. For the rest of this chapter the behaviour of the pumps is assumed to be ideal.

2.1.2 Description of metering pump using equivalent electronic circuit

In order to design an analyser based on metering pumps an equivalent electronic circuit was developed that represents the behaviour of an individual pump. Electronic models were compared with fluidic models do determine equivalent parameters. This information is used to draw the schematic for the electronic circuit.

In electronic circuits capacitance is defined as detailed in Equation 2.1. Where c is capacitance, v is voltage and q is electric charge. This analogous to the equation for compliance as detailed in Equation 2.2. Where C is the compliance, ΔV is the volume of the pumping chamber and ΔP is the pressure difference between both sides of the diaphragm. Capacitance is analogous to compliance as capacitance it the ratio of charge to voltage and compliance is the ratio of volume to pressure applied.

$$c = \frac{q}{v}$$

Equation 2.1: Capacitance

$$C = \frac{\Delta V}{\Delta P}$$

Equation 2.2: Compliance of pump diaphragm

With an analogue between compliance and capacitance a Resistor Capacitor circuit that is analogous to a diaphragm and microfluidic system can be created. Equation 2.3 and Equation 2.4 detail the equations regarding the charging and discharging of a capacitor in a resistor-capacitor circuit. The charge of the capacitor Q is determined by the total charge the capacitor can hold Q_f, the time elapsed t and the time constant RC. The equation governing the discharge of a capacitor through a resistor is the inverse of Equation 2.3.

$$q = q_f\left(1 - e^{-t/rc}\right)$$

Equation 2.3: Charging of capacitor in RC circuit

$$q = q_f e^{-t/rc}$$

Equation 2.4: Discharge of capacitor in RC circuit

These equations can be converted to the fluidic equivalent by letting charge q be equivalent to the volume of the pump chamber ΔV, electric resistance r be analogous to fluidic resistance R, capacitance c be analogous to compliance C. As such, the equations for governing the behaviour of the metering pump are described in Equation 2.5 and Equation 2.6

$$\Delta V = \Delta \mathrm{V}_f\left(1 - e^{-t/RC}\right)$$

Equation 2.5: Fluid being drawn into metering pump

$$\Delta V = \Delta V_f e^{-t/RC}$$

Equation 2.6: Discharge of fluid from metering pump

Figure 2.4 shows a Resistor-Capacitor circuit and a diaphragm filled through a tube. The two systems are analogous to each other and the relationship between the two systems was used to draw an equivalent electronic circuit of the solenoid metering pump detailed in Figure 2.1.

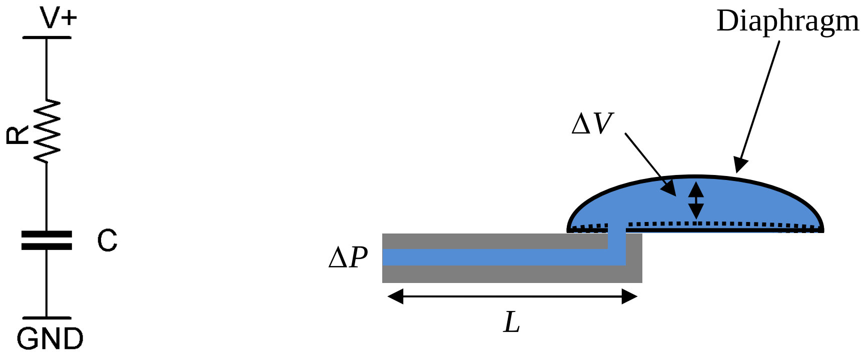


Figure 2.4: Resistor-Capacitor, Metering pump diaphragm analogy

To create the equivalent electronic circuit for the pump in Figure 2.1 a schematic was developed that shows the metering pump switching between the states were fluid is drawn into the pump (see Equation 2.5) and when the pump is discharging (see Equation 2.6). Figure 2.5 shows the equivalent electronic circuit for a solenoid actuated metering pump.

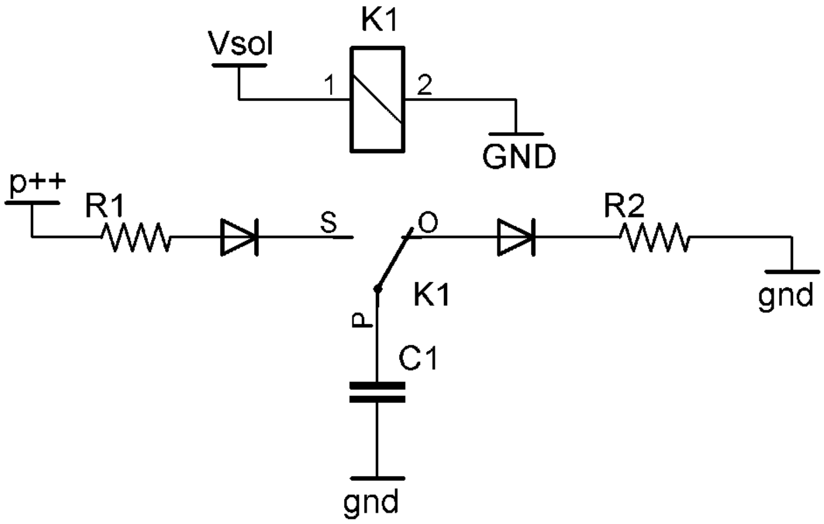


Figure 2.5: Equivalent electronic circuit for solenoid metering pump

C1 represents the compliance of the pump diaphragm and the two diodes represent the check valves of the pump. R1 represents the fluidic resistance of the tubing leading to the pump inlet and R2 represents the fluidic network the pump injects the fluid into. The solenoid actuator is represented by the relay K1. This switches the pump from drawing in fluid and discharging fluid. The switch is shown in the normally open position. The maximum pressure of the pump is represented by p_{++}. R1 is the fluidic resistance of the tubing leading from a fluid reservoir to the pump. It determines the length of time the solenoid has to be switched on for the pump to draw in the maximum amount of fluid.

The pump described in Figure 2.5 is assumed to be ideal. The inlet/outlet pressure, the volume pumped and performance of the check valves are based on fixed values. The circuit simulates of the ideal operation 120SP micro pump (BioChem Valve, USA). The variation in performance as detailed in the pump's data sheet (see Appendix A) is not incorporated into the equivalent electronic circuit. The circuit was used to analyse the fluid flow through the system. It is shown in Chapter 4 how random and systematic variation between different pumps affects the performance of the microfluidic system.

2.2 Fluid mixing using discrete metered fluid injection

Work by Yamada et al. (2006) has demonstrated that a number of mixing ratios can be achieved by varying the layout of the fluidic network. In the system the ratio of fluidic resistance of the two channels leading to a T-mixer inversely relate to the resulting mixing ratio. Their design was based on two key assumptions. That the viscosity of the two fluids to be mixed was identical as this will affect the fluidic resistance of the channels and, the pressure at the end of both channels leading to the T-mixer is equal. An important note is that the total flow rate and the coefficient of diffusion of the fluids to be mixed will not affect the ratio of mixing. The ratio of flow rates of the two streams meeting at the T-mixer inversely relate to the mixing ratio, the magnitude of the two flows will have no effect (Provided the Reynolds number of the flow is below 2300).

These properties can be used to create a system that mixes fluids accurately using solenoid metering pumps. A T-mixer is integrated for mixing fluids of the same viscosity. In the case where fluid is to be mixed at a 1:1 ratio the channels leading to the T-mixer were of equal fluidic resistance. A metering pump is connected to the beginning of each of the two channels leading to the T-mixer. The equivalent electronic circuit of the mixing system is shown in Figure 2.6.

R3 and R4 represent fluidic resistance of the micro channels leading to the T-Mixer. R5 represents the fluidic resistance of the outlet. The voltage sources ‘Reagent’ and ‘Sample’ represent the maximum pressure of the pumps. The pressures of the pumps were assumed to be equal. R1 and R2 are the fluidic resistances of the tubing leading to the inlets of the pumps. These values do not have to be equal, but they do determine the length of time it takes for the pumps to fill to their maximum volume.

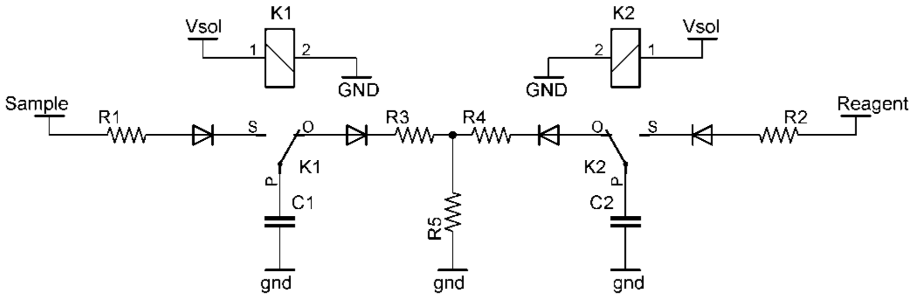


Figure 2.6: Equivalent electronic circuit for mixing of two fluids using metered fluid injection

In operation the solenoids K1 and K2 are switched on by V_{sol} which simultaneously flips the switches charging both capacitors. The solenoids remain on until both capacitors reach the same voltage as ‘Reagent’ and ‘Sample’.

When both capacitors are fully charged the solenoids are simultaneously deactivated flipping the switches back to the normally open positions. Both capacitors then begin to discharge through the resistor network. Since the resistance of R3 and R4 are equal and voltages of the capacitors start at the same level the current passing through R3 is equal to the current passing through R4. This relation holds true throughout the discharge cycle of both capacitors.

In fluidic terms when the solenoids are actuated the pumps draw fluid through the inlets and the diaphragms expand. When both pumps have drawn in the full amount of fluid the solenoids are switched off. At this point both pumps begin to inject fluid into the micro channel network. The fluidic resistances of the channels leading to the T-mixer are equal. When the pumps discharge they start at an equal pressure and that pressure decreases at an equal rate. The result is that the flow rate at both inlets of the T-mixer is equal for both fluid flows. When the fluid flows meet at the T-junction they mix at a 1:1 ratio.

In this system the flow rate from the pumps will start high but will eventually approach zero. The length of channel the fluids travel before they mix will start long but will shorten until it approaches zero as the flow rate decreases. As the flow is not continuous it is important in this system design that the internal volume of the micro channel network is less than the combined internal volume of the pumps. The time for the fluids to mix in this system is determined by the Einstein-Smoluchowski relation (Equation 2.7) where *t* is the time, x is half the channel width in cm and D is the coefficient of diffusion of the molecules to be mixed. It should be noted that the velocity of the fluids does not factor into this equation.

$$t = \frac{x^2}{2D}$$

Equation 2.7 Time to mix

2.3 Concept for ROCS using discrete metered fluid injection

Simply being able to mix a reagent with a water sample is not enough to create a fully functional reagent based optical chemical sensor (ROCS). It is necessary to integrate a spectrometer into the design to take an absorbance measurement of the reacted sample. This spectrophotometer has to be calibrated to determine what the baseline and gain of the absorbance measurement is. The design has to account for drift over time due to degradation of the spectrophotometers components, temperature variation and changes in ambient light. So this effect the ability for the analyser to perform a 2-point calibration is needed.

Also the performance of the pumps are not ideal in real operation. A number of factors are assumed to be equal between separate pumps. Differences in pump pressure and volume pumped are likely cause the analyser to drift from its predicted performance. The analyser needs to be calibrated initially to determine the magnitude of these systematic errors, as is shown in Chapter 4.

A problem specific to ROCS can also cause drift in both the baseline and gain. The work of McGraw et al. (2007) highlighted the need for a measurement of a blank to compensate for drift in the output of the phosphate analyser presented. Staining of the Lab-on-a-Chip caused the absorbance measurement to increase over time. However, with the addition of a blank measurement, drift was still present in analyser output (Cleary, 2008). Staining causes a decrease in resolution of the analyser as the gain is reduced.

It is also a requirement that the effect of this staining is reduced by the integration of a self cleaning function into the analyser. Cleaning can be achieved by pumping distilled water through the microchannels. This strategy is used by both the AZTEC P100 and TresCon OP210 to self clean (see Appendix A).

2.3.1 Description of phosphate analyser using equivalent electronic circuit

To describe how the fluidics of the completed analyser functioned, a schematic represented by an equivalent electronic circuit is shown in Figure 2.7.

The analyser consisted of 6 identical solenoid metering pumps that inject into a network of microchannels. To perform a phosphate assay, sample water was mixed with the reagent and the colour measured. The pump for the sample and the pump for the reagent were positioned opposing channels leading to the T-mixer. The pumps for the blank and phosphate standard were also positioned opposite to the pump for reagent. For mixing to be carried out at a 1:1 ratio the fluidic resistance of R13 must be equal to R14. Furthermore, R7, R8, R9 and R11 must also be equal resulting in a 1:1 mixing ratio for all solutions.

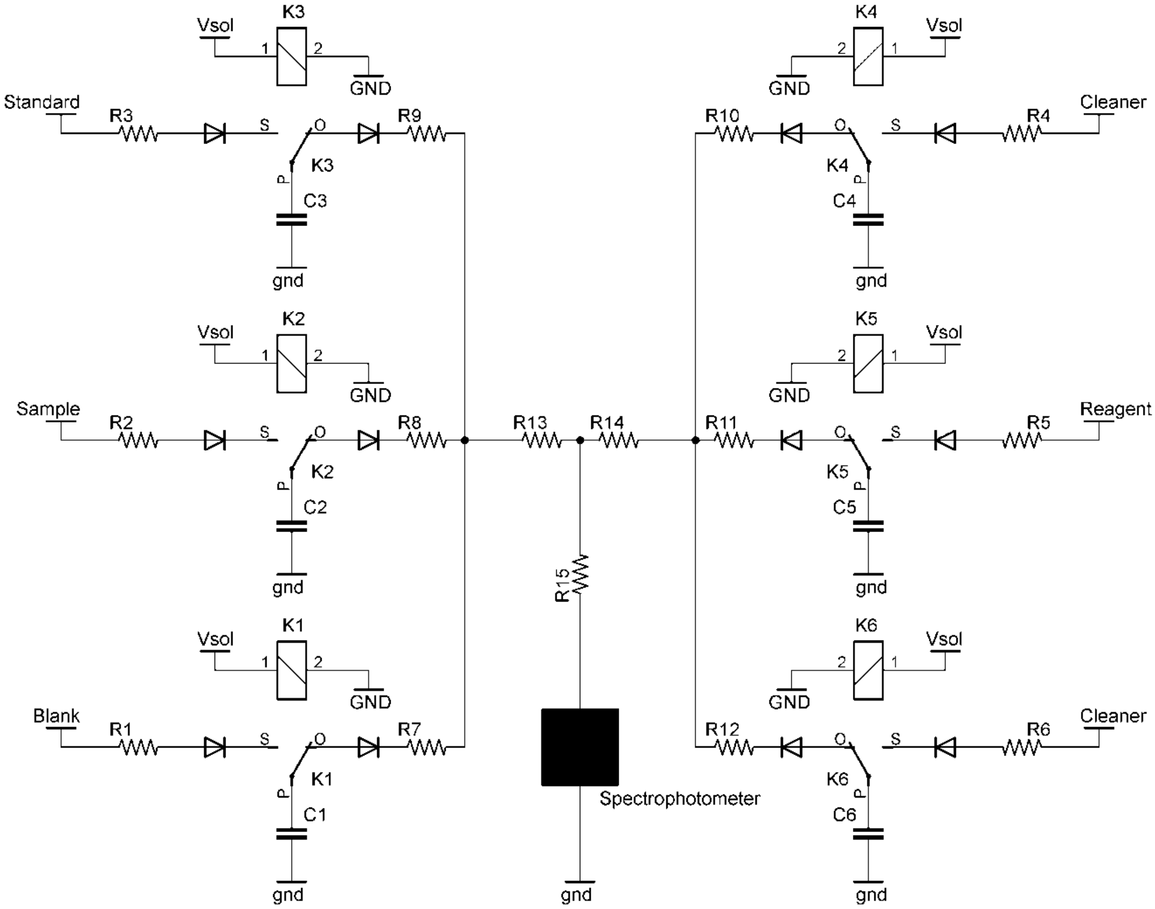


Figure 2.7: Equivalent electronic circuit for phosphate analyser

The pumps for the cleaning solution were positioned so that they inject from the same side as the reagent. Their positioning is not critical as the fluid they dispense was not mixed and must only wash through the microchannels at the spectrophotometer. The intention was to keep the design of the system symmetrical.

The check valves on the solenoid metering pumps were critical to the operation of this system, represented by diodes in Figure 2.7. They isolate the pumps from the microchannel network when they were not injecting fluid. This means regardless of what solution is being mixed with the reagent the effective fluidic system acted as described in Figure 2.6.

CHAPTER 3. CONSTRUCTION OF PHOSPHATE ANALYSER

This chapter describes the implementation of the fluidic system described in Chapter 2. To create the complete phosphate analyser the fluidic system was combined with an LED-Photodiode spectrophotometer, temperature sensor and the necessary chemicals. All the fluid storage and handling, electronic and optical components required to make the functioning analyser are detailed here.

The microfluidic lab-on-a-chip was an implementation of the microchannel network detailed in Figure 2.7. The chip contains a T-mixer, an area for presenting the reacted sample to the analyser's optics and, microchannels for connecting these components together with the inlets and outlet of the chip.

The chip was fabricated from poly methyl methacrylate (PMMA) using a micro milling machine, chip layers were machined individually and bonded together thermally after UV surface treatment to create a multilayered 3 dimensional chip. The full design and fabrication method are described in this chapter.

The reagent used was the vanadomolybdenumphosphoric acid method (yellow method) and has been shown to have a lifetime of over one year (Bowden, 2003). This reagent forms no precipitate so can be used in a microfluidic chip. In practice the reagent was mixed at a 1:1 ratio with the sampled water. This simplified the design of the fluidic chip. The reaction takes place over 2 to 60 minutes depending on temperature and absorbance can be measured below 400nm (Bowden, 2001, 2002). This allowed the use of an LED and Photodiode for the absorbance measurement (McGraw, 2007). The only major drawback of this reagent was that it is composed of an aggressive acid that readily corrodes stainless steel. This means that all wetted components had to be made from inert plastics.

It is also shown in this chapter how the phosphate analyser was integrated into a self contained field deployable platform. The analyser was contained in protective enclosure with a control board, communications system and a power supply.

3.1 Analyser fluidics

3.1.1 Pumps

The solenoid metering pump selected for the analyser was the 120SP miniature solenoid metering pump (BioChem Valve, USA). The pump’s wetted parts are made from inert materials thus making them suitable for the yellow phosphate reagent. The dimensions of the pump are 25 mm in diameter and 64 mm high. Although its dimensions are small enough for it to be integrated into a portable analyser the pump would not be classed as a micro pump (Laser, 2004).

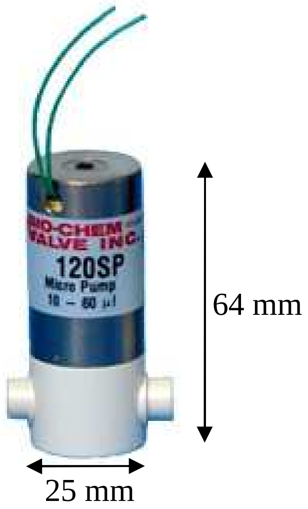


Figure 3.1: 120SP Solenoid metering pump

A portable system must be able to run off a battery for a reasonable amount of time. The pumps were also suitable due to their low power consumption. The solenoid in these pumps actuates at 12V and draws 320 mA. The maximum cycle rate of these pumps is 2 Hz, at a 50% duty cycle the maximum on time is 0.25 seconds. This equates to 26 × 10^{-5} Wh of power consumption per cycle. This level of power consumption would allow a 1.5V 1000mA AA battery actuate the pump nearly 6,000 times.

The amount of liquid dispensed by this pump is in the micro litre range. As the pump was not designed to be coupled with microfluidic channels the amount of liquid dispensed was in excess of what is needed. At a 20 µL dispense volume, the pump would move liquid 0.5 m down a 0.2 × 0.2 mm square cross sectional area capillary. The reliability of the 120SP was a key factor in its selection for the analyser. In normal use it is capable of up to 20 million cycles. This is enough cycles to pump 400 litres of fluid.

Pumps with smaller dispense volume are available but these do not have the same precision in pressure and dispense volume. Examples of this are the 090SP (BioChem Valve, USA), the dispense volume is 8 µL but the dispense volume precision is ± 25% (see Appendix A)

For mounting the pump has two M4 screw holes at the top. The pump can be mounted vertically in any direction or horizontally with the outlet facing upwards. Tubing is connected using the two ¼”-28 UNF ports for the inlet and outlet.

Six pumps were required for the completed analyser. Each of the fluids to be used had its own pump. This negated the need for valves in the analyser’s fluidics. There was one pump for sample water, one pump for the reagent, two pumps that were used for a two point calibration and a further two pumps for cleaning solutions. The check valves in the pump ensure that the pumps were isolated from the microchannel network when they were not being used.

3.1.2 Design of Lab-on-a-Chip

The microfluidic lab-on-a-chip mixed a number of fluids with a T-mixer. There was also a cuvette for holding the reacted sample for the analyser’s optics to perform an absorbance measurement.

With microfluidic devices special consideration must be made regarding air bubbles. The analyser design presented in Chapter 2 is not tolerant to bubbles. Any bubble present in the system would affect the level of phosphate measured. The round surface of the bubble and difference in refractive index between water and air would cause light to scatter, thus increasing the absorbance measurement. A bubble can also become trapped at a constriction or at a sharp turn (Jensen, 2004). If this were to happen in the optical cuvette in the analyser the absorbance measurement would be affected.

Bubbles can also affect the rate of fluid flow in a microchannel since air is compressible. If a bubble is within a channel section and a pressure is applied across it the bubble compresses. In this situation the fluid flow rate before the bubble is faster than fluid flow rate after the bubble. Additionally, when the pressure across the channel is reduced the bubble expands as the pressure normalises. This causes the fluid flow rate after the bubble to increase. In the case of an analyser where accurate mixing is a critical factor in its operation a bubble can have a serious effect on the performance.

A common method for eliminating bubbles from a microfluidic device is to introduce a bubble trap or debubbler into the design. This however would not be feasible in a portable analyser. A bubble trap, although simple in construction, merely traps bubbles and does not let them pass a certain point in the device. When too many bubbles are trapped it overflows, rendering the trap ineffective. Active debubblers, although effective, can be complicated devices that would be difficult to integrate into a portable analyser (Skelly, 2008).

As the integration of bubble trap or debubbler complicates the design of the analyser the strategy undertaken was to design a Lab-on-Chip that is tolerant to bubbles and not trap them. The layout allowed bubbles to pass through the LOC unobstructed. In this case when a bubble was present it affected the mixing and measurement performance of the system only once. In subsequent measurements the bubble was pushed through the LOC so that it did not affect further measurements. Bubbles can be considered analogous to noise on electronic signals, an interference that causes the system to deviate from the ideal.

Microchannels

The microchannels in the Lab-on-a-Chip had a cross-sectional area of 200 × 200 µm in the channels leading to and from the T-mixer. This will ensure the flow is laminar and the T-mixer will function as intended. Transitional flow and turbulent flow regimes are achieved when the Reynolds number exceeds 2000. In a channel of 200 × 200 µm cross sectional area with distilled water flowing through at room temperature (20^{o} C) the velocity of the liquid would have to exceed 10 m/s to achieve a transitional flow regime as calculated in Equation 3.1. *Re* is the Reynolds number, η is the viscosity of water, ρ is the density of water, D_h is hydraulic diameter of the channel and V is the velocity of the fluid flow.

$$Re = \frac{\rho D_h v}{\eta}$$

$$V = \frac{Re \cdot \eta}{\rho D_h}$$

$$V = \frac{2000 \times 1 \times 10^{-3}}{1000 \times 0.2 \times 10^{-3}}$$

$$V = 10\, {}^{\mathrm{m}}\!/_{\mathrm{s}}$$

Equation 3.1: Calculation of velocity needed for a transitional flow regime

Given that the pump selected had a pressure of 0.35 bar (35000 Pa) the length of the microchannel would have had to be short to reach this velocity. The volumetric flow is calculated in Equation 3.2. This value was used to calculate the length the channel would have to be to achieve this volumetric flow rate given that the pressure is 0.35 bar using the Hagen-Poiseulle flow equation Equation 3.3. Q is the volumetric flow rate in m/s, $28.454\eta L/a^4$ is the inverse of the fluidic resistance in channel of square cross section and ΔP is the pressure difference across the channel. The minimum length required is calculated as 3.2×10^{-13}. In the design of the LOC the length of the channel leading from the pump to the waste container would have to be less than the calculated value for the flow to become turbulent.

$$Q = a^2 \times v$$

$$Q = 0.04 \times 10^{-6} \times 10$$

$$Q = 0.4 \times 10^{-6}\ \mathrm{m^3 s^{-1}}$$

Equation 3.2: Calculation of volumetric flow rate

$$Q = \frac{28.454 \cdot \eta \cdot L}{a^4} \Delta P$$

$$L = \frac{Q \cdot a^4}{28.454 \cdot \eta \cdot \Delta P}$$

$$L = \frac{0.4 \times 10^{-6} \times 0.0008}{28.454 \times 0.001 \times 35000}$$

$$L = 3.2 \times 10^{-13} \text{ m}$$

Equation 3.3: Calculation of minimum channel length required for transitional flow

The layout of the microchannels integrated into the Lab-on-a-Chip are detailed in Figure 3.2. Each of the channels has a 200 × 200 µm cross sectional area. The length of each of the branches leading to the T-mixer is 2.5 mm. This is an arbitrary length, as explained in Chapter 2 the magnitude of the fluidic resistances is not important provided the flow remains laminar. What was important in the design of this LOC was that the fluidic resistances of all flows leading to the T-mixer were equal. This resulted in the 1:1 mixing ratio required for the analyser to function. The layout detailed in Figure 3.2 was fabricated into a single layer of the PMMA substrate.

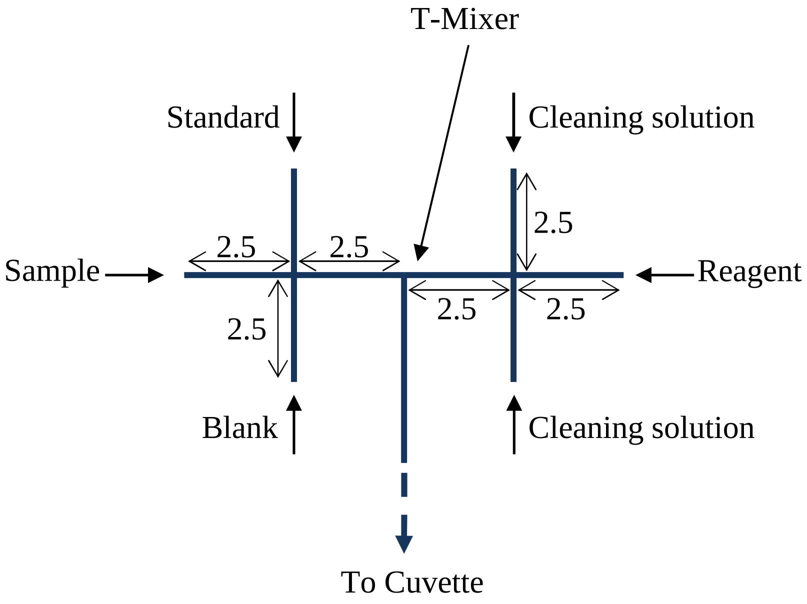


Figure 3.2: Layout of microchannels (Dimensions in mm)

Cuvette

The cuvette was the part of the LOC where the reacted sample was held while the absorbance was measured by the LED-Photodiode spectrophotometer that is described fully later in this chapter. The path length of the cuvette was made long to increase the limit of detection in the finished analyser. Since the design had two right angle turns the shape of the cuvette was such that it would be susceptible to trapping bubbles. Special attention described below was made to circumvent this problem.

The cuvette was a cylinder 1 mm in diameter and 3 mm in height. Fluid entered the cuvette through 200 × 200 µm channel with a square cross section. Once in the cuvette the fluid flows up and around the bend at the top of the cuvette (Figure 3.3). The end of the cuvette was curved and the cross section of the channel leading away from the cuvette was of the same shape.

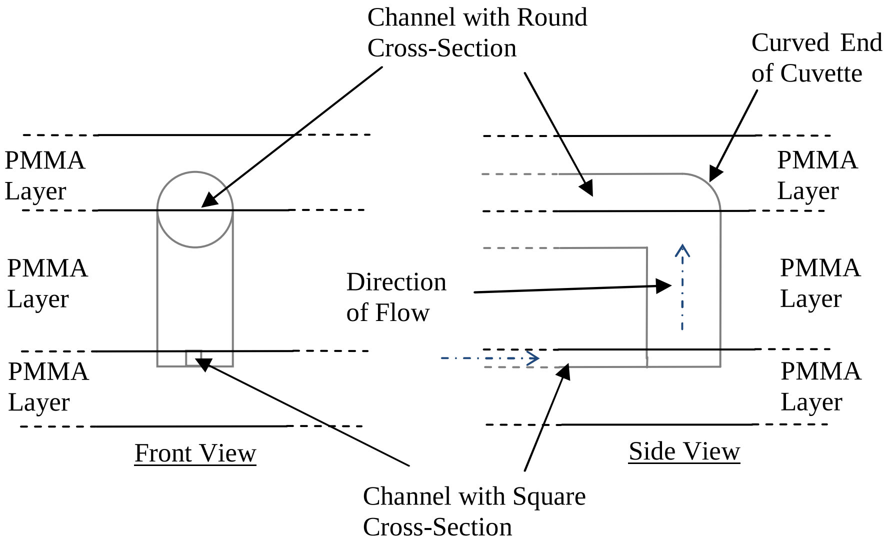


Figure 3.3: Layout of optical cuvette/fluidic via

This configuration allowed a bubble that was large enough to span the diameter of cuvette to pass through and not become trapped. As a bubble is spherical in shape it matched the shape of the cuvette cross section. When there was a pressure difference across the cuvette, caused by the actuation of the pump, fluid was unable to flow around the bubble and pressure applied across it.

Computational studies of bubbles in microchannels has shown that a bubble can be ejected by applying enough pressure across it (Jensen, 2004). In the case of the 200 × 200 µm channels with the square cross section the increased fluidic resistance caused a large pressure difference across the bubble and allowed any bubble to be easily ejected from the channel. However, if the cuvette had a square cross section it would have a lower fluidic resistance. Fluid would be able to flow around the bubble and pressure applied across it would not be great enough to move it along the channel.

At the end of the cuvette the channel turned at a right angle. This change in direction was likely to cause a bubble to become trapped in the corner. In the design the cuvette the cross section of the channel leading away was round. In this case there is no area where fluid would be able to flow around the bubble, thus the pressure across the bubble remained high enough to eject the bubble from the cuvette.

The 3 dimensional shape of cuvette layout was machined in 3 layers of PMMA. The microchannel leading to the cuvette was embedded in the bottom layer. The cuvette was a cylindrical hole through the PMMA layer above. The round cross section channel leading away from the cuvette was made of two sections. Both sections were semicircular channels machined into the top and middle layers which when mated formed a single channel with a round cross section. The same construction was used for allowing fluid to flow vertically between layers of the LOC.

Lab-on-a-Chip interconnects

The final part of the LOC design was the fluidic interconnects by which fluid entered and exited. PEEK® tubing which is rigid was inserted directly into the LOC via a friction fit. The type of tubing that was used was PEEK® tubing 1/6" outer diameter, 0.8 mm inner diameter (Upchurch Scientific, USA). The hole for receiving the tube was machined through two of the layers.

When assembling the LOC a piece of PEEK® tubing was cut to 20 mm in length and inserted into the hole. The diameter of the hole that the tube was inserted into was machined to the outer diameter of the tube with a H5 tolerance. This means that the tube was able to be inserted by hand, but was tight enough to give a reasonable seal. This method was similar to a method of creating fluidic interconnects using hollow steel pins instead of PEEK® tubing developed by Chen et al. (2009). To ensure the seal was water tight, epoxy glue (Araldite Epoxy Glue, Radionics Ltd., Ireland) was applied to the joint where the tube meets the top PMMA layer.

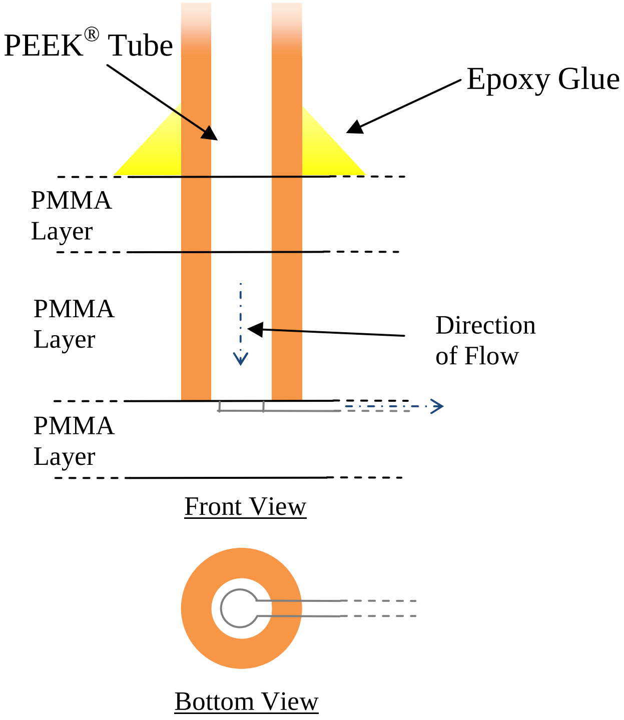


Figure 3.4: Left: layout of fluidic interconnect

Complete Lab-on-a-Chip

Figure 3.5 shows the three layers of the Lab-on-a-Chip. Each of the layers were bonded directly to create a three dimensional layout (see Figure 3.6). Fluid entered the LOC via the six inlet channels. These channels met at the T-mixer where flow continued to the cuvette via the forward channel. The fluid then flowed vertically through the cuvette and around the bend as detailed previously in Figure 3.3. The fluid flowed through the return channel to the fluidic via where it passed down to the bottom layer where the fluid flow reached the outlet. The total internal volume of the LOC was 23 µL.

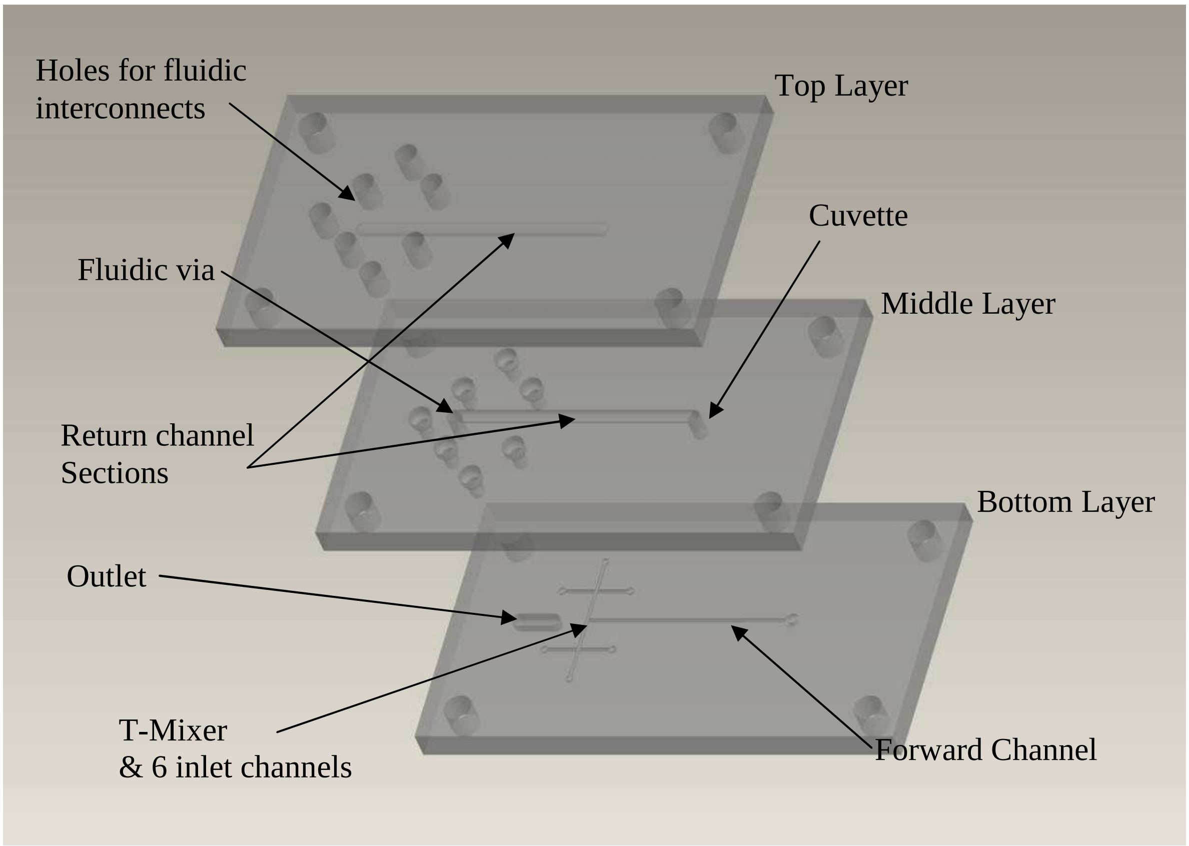


Figure 3.5: Exploded view of chip

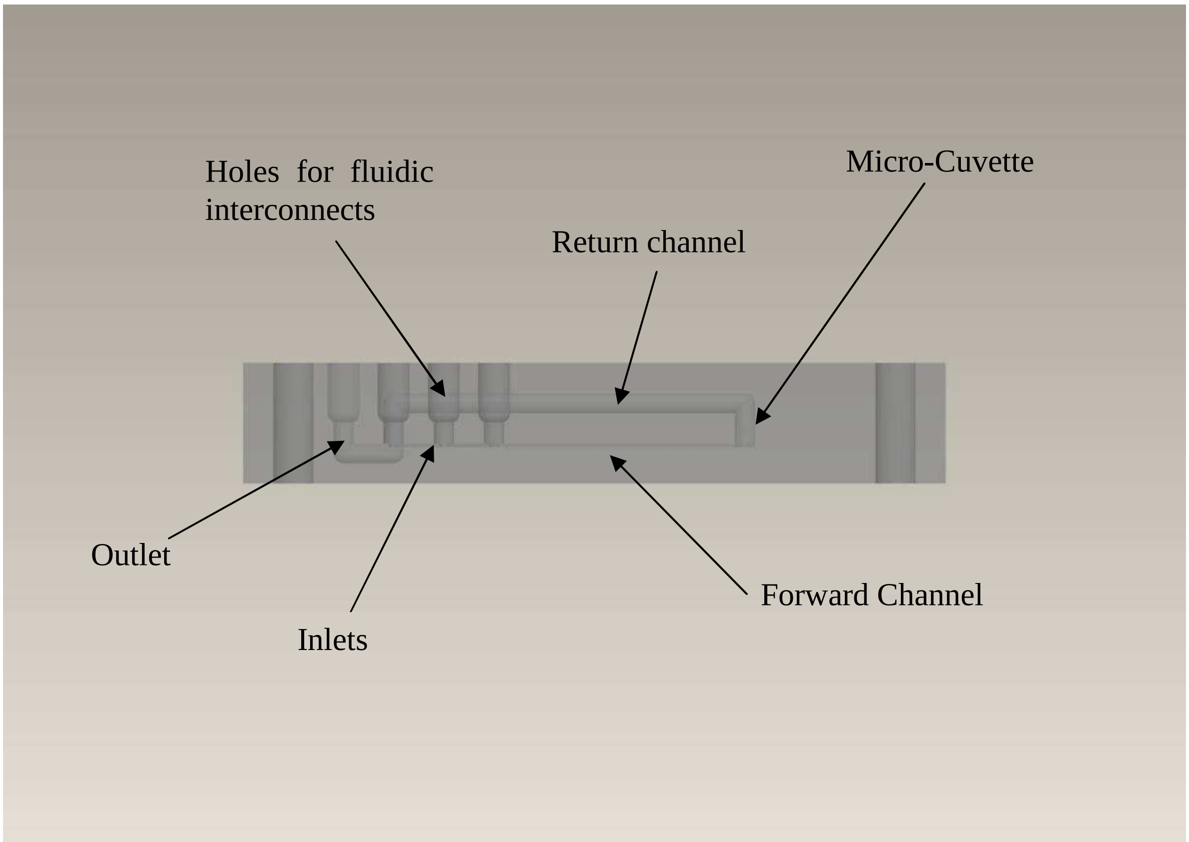


Figure 3.6: Side view of Lab-on-a-Chip showing 3 dimensional layout

The dimensions of the top and bottom layers were 35 × 20 × 1.5 mm and the dimensions of the middle layer were 35 × 20 × 2.5 mm. When all the layers were bonded directly together the resulting LOC is 35 × 20 × 5.5 mm in size.

3.1.3 Fabrication of the Lab-on-a-Chip

The chip design was manufactured by machining each of the layers using a micro-milling centre (CAT-3D-M6, DATRON, Germany). The machined layers were cleaned and the mating surfaces were treated by UV irradiation before the LOC was assembled in a jig and the layers bonded under heat and pressure.

The substrate used in fabricating the LOC was poly methyl methacrylate (PMMA). This polymer was used because it was optically transparent down to 254 nm wavelength of light and has low auto fluorescence (Piruska, 2005). PMMA is rigid and is soft enough to be milled effectively.

The main reason for its use in fabrication of the LOC was so the chip could be bonded using the low temperature direct bonding method developed by Tsao et al. (2007). In this bonding method the surfaces of the PMMA to be mated are irradiated by UV light at 184.9 nm and 253.7 nm. The light at 184.9 nm breaks O_2 in to free radicals and the light at 253.7 nm breaks the polymer chains at the surface of the PMMA. The free radicals form oxygen containing functional groups that are incorporated into the surface of the PMMA. This results in a more energetic surface that has enhanced hydrophilicity. The advantage of this method is that the polymer does not have to be heated above its glass transition temperature to from a bond thus the channels are not deformed.

Micromilling

Two thicknesses of PMMA sheet were used in the manufacture of the LOC. 1.5 mm thick sheets were used for the top and bottom layers and a 2.5 mm sheet was used for the middle layer. The sheets were cut into 90 × 90 mm sections and 6.5 mm diameter holes were drilled at each of the corners for mounting the sections on a CNC milling machine.

A 5mm thick sheet of PMMA was also cut into an 80 × 80 mm section to act as a support when the PMMA section to be machined was placed on the milling machine. This section also had 6.5 mm holes drilled to each corner. The purpose of the support was to prevent the substrate from flexing while it was being machined. PMMA was used for the support as it allowed machine tools to pass through the substrate without becoming damaged.

The CNC milling machine used was the CAT-3D-M6 (DATRON, Germany). It has a precision of 3 µm, a maximum spindle speed of 60,000 RPM, an 80 × 80 mm work area and has an automatic 4 slot tool changer. The CNC code was written using the Excalibur CAD/CAM package.

To mount the PMMA substrate the 5 mm PMMA support was placed on the table mount and the PMMA substrate was placed on top. Four M6 × 20 mm bolts were passed through the corner holes and fastened to the table mount to hold the substrate and support in place (see Figure 3.7).

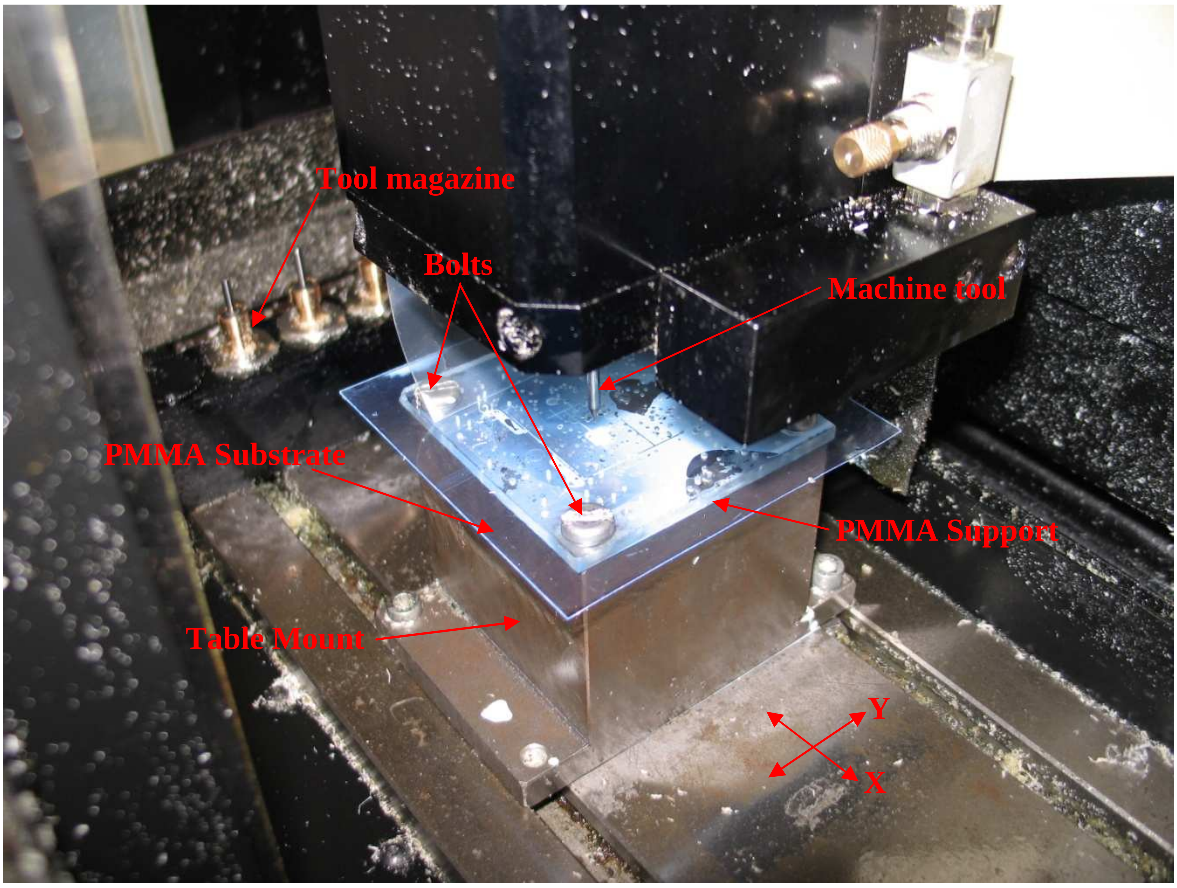


Figure 3.7: Machining of PMMA substrates on micromill

The milling of the LOC layers was split into a number of operations. First the channels were machined, then the holes for the via and cuvette, then the holes for the fluidic interconnects were milled to the correct diameter and finally the layer was cut from the substrate. The LOC's microchannels were machined into the 1.5 mm thick bottom layer, bottom half of the round cross-section return channel, the cuvette and via were machined into the 2.5 mm thick middle layer and the top half of the round cross-section return channel was machined into the 1.5 mm thick top layer.

Heat generated by the friction of the cutting tool can be problematic when machining polymers. The glass transition temperature of PMMA is approximately 90º (Tsao, 2007). Should the temperature of the machine tool rise above this level the substrate deforms and is not cut properly by the machine. In this case the substrate has to be discarded. To prevent this from happening distilled water was used as coolant.

When milling a channel of square cross-section an endmill was used. The 200 × 200 µm microchannels were machined using a 200 µm diameter endmill (DATRON, Germany). The channel was cut in 4 passes each stepping down 50 µm. A shallow dept of cut was used to reduce the amount of material being removed from the substrate during machining thus reducing the amount of torque being exerted on the machine tool. The feed rate was set to 2 mm per second as the slow speed reduces the amount of heat generated during milling. Due to the small size of the milling tool the spindle speed was set to 15,000 RPM.

The channel with the round cross-section was made of two semi circular cross-section channels. These channels were milled using a 1mm ballmill (DATRON, Germany). The ballmill cut through the substrate with 4 passes in 0.25 mm steps. The spindle speed was set to 9,000 RPM and the feed rate was 10 mm per second.

The holes for the cuvette and via were drilled using a 1 mm drill. The spindle speed of the drill was 6000 RPM. Material was removed from the hole in 100 µm steps each time the drill removed a step it is retracted fully from the hole to allow the debris to exit. The plunge rate of the drill was 2 mm per second.

The holes for the fluidic interconnects were machined using a 1 mm endmill. The hole was machined by contouring the diameter of the hole using the 1mm endmill. The diameter of the hole was 1.587 mm (1/16"), the diameter of the PEEK® tubing. The mill moves in a clockwise direction so that no burr was left on the outside of the hole. The mill cut down to the thickness of the substrate plus 0.5 mm in steps of 0.25 mm. The feed rate of the mill was 10 mm per second and the spindle speed was 9,000 RPM.

The final part of machining a chip layer was to cut it from the PMMA substrate. This was done with a 2 mm endmill. The endmill was set to a spindle speed of 7,000 RPM, a feed rate of 20 mm per second, and cut down through the substrate in 0.5 mm steps. The endmill moved around the LOC layer in a counter-clockwise direction so that a burr was not left behind. The 35 × 20 mm LOC layer was cut out by machining 0.5 mm deeper than the thickness of the substrate.

Figure 3.8 shows the 3 finished layers of the LOC. After machining the layers were sonicated in distilled water for 1 hour to remove the debris from milling. The layers were dried by blowing compressed nitrogen over them. They were then rinsed with isopropyl alcohol to remove any organic matter.

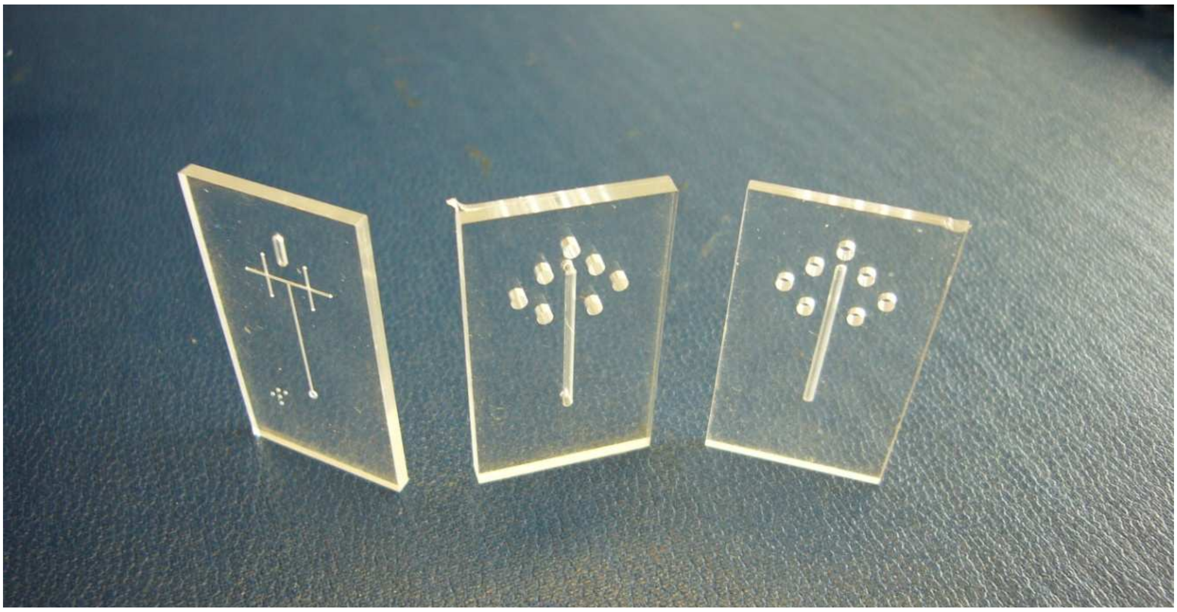


Figure 3.8: Layers of Lab-on-a-Chip

For the layers to be bonded together they were irradiated with UV light to make the mating surfaces hydrophilic. The lamp used was a 400W mercury UV lamp (2000-EC, Dymax, USA) that radiates light from 180 nm to 400 nm. The lamp was placed 10 cm above the chip layers and the surfaces were irradiated for 1 minute to make them hydrophilic. The top and bottom layers were irradiated on the side with their channels and the middle layer was irradiated on both sides.

The layers were then assembled using a custom jig. This was fabricated using a FDM 3D printer (Dimension SST 768, Dimension Printing, USA). Figure 3.9 shows the Lab-on-a-Chip being assembled in the jig. The jig was pressed using G-clamps exerting the required pressure of 3 MPa on the LOC layers (Tsao, 2007). The jig was then placed in an oven preheated to 85° for 2 hours so that the LOC layers bonded.

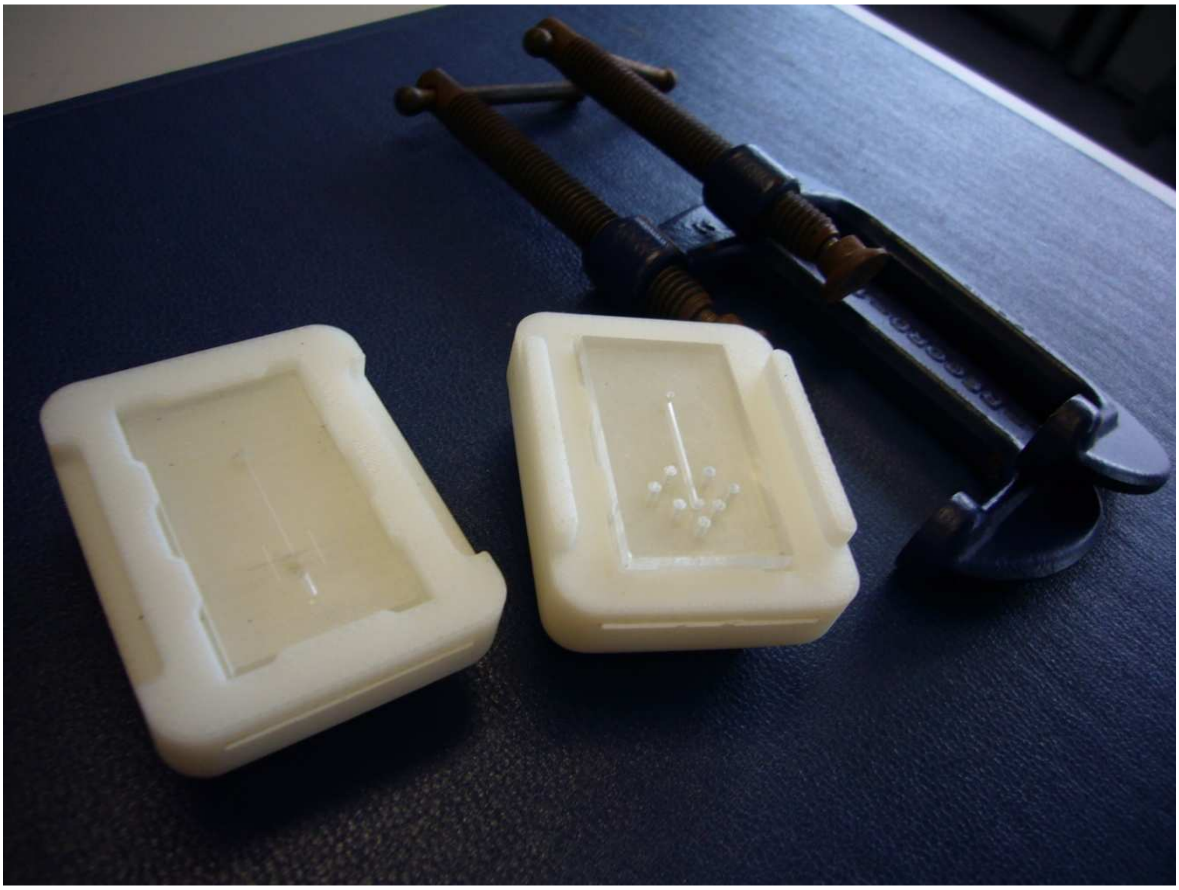


Figure 3.9: Assembly of Lab-on-a-Chip in 3D printed jig

The bonded LOC was removed from the jig and was allowed to cool to room temperature. Seven 20 mm lengths of 1/16" outer diameter PEEK® tubing (Upchurch Scientific, USA) were cut and inserted by hand into the holes for the fluidic interconnects. The LOC was then placed in an oven preheated to 85º for 30 minutes to relieve stress from around the inserted tubing. This was done to prevent cracking of the PMMA around the tubing (Chen, 2009). The completed LOC was shown in Figure 3.10. Epoxy glue was applied where the tubing meets the PMMA on the LOC to ensure that the interconnect was water tight.

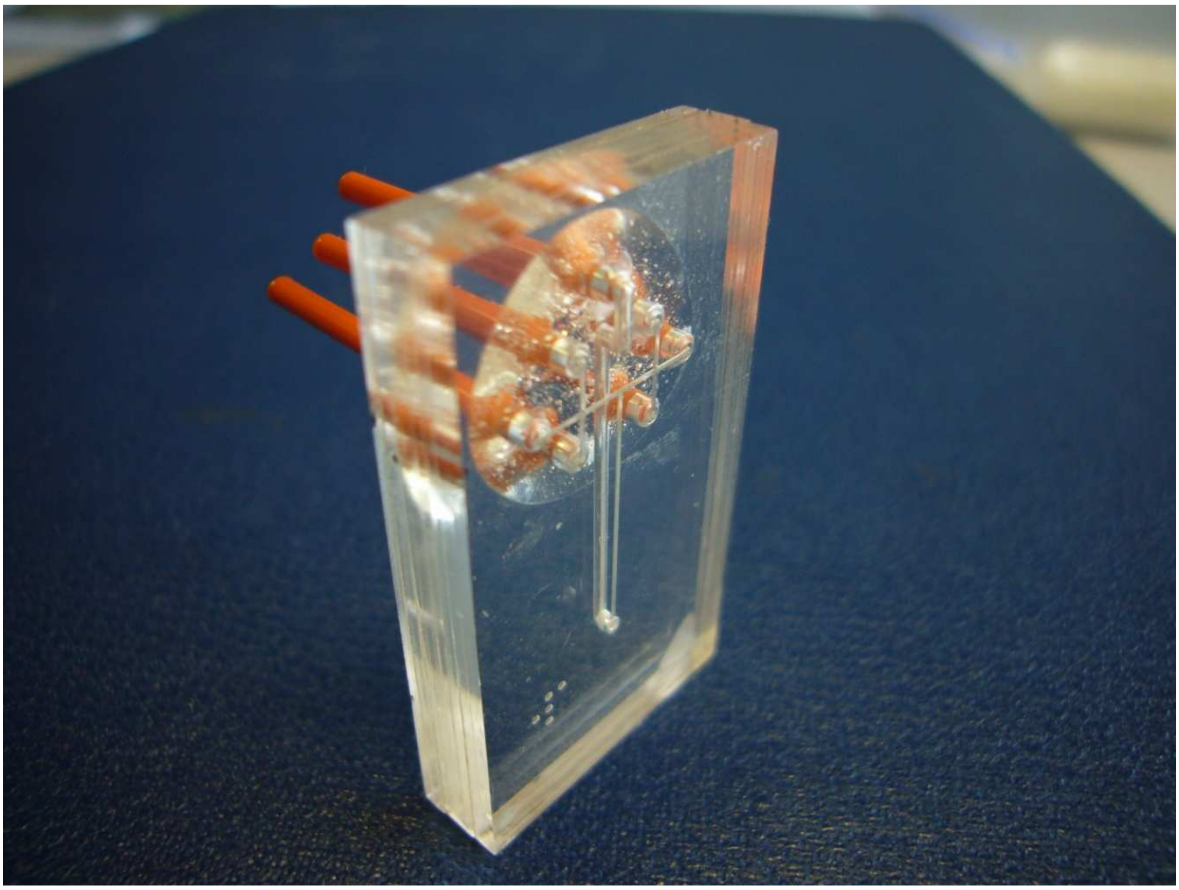


Figure 3.10: Completed Lab-on-a-Chip

3.1.4 Sample collection, fluid storage and plumbing

Sample port

The water sample was drawn into the analyser through a 0.45µm pore diameter filter membrane (Supor®, PALL Corporation, USA) that was held in place by polycarbonate plates that were 60 × 60 × 8 mm (see Figure 3.11) and sealed with TEFLON® O-rings (McMaster-CARR, USA). The sample port plates were manufactured by milling machine (M.A.D.S Inc., USA). The port had a ¼”-28 UNF port on the back to allow tubing to be connected. The sample port was mounted on the outside of the enclosure (described later in this chapter).

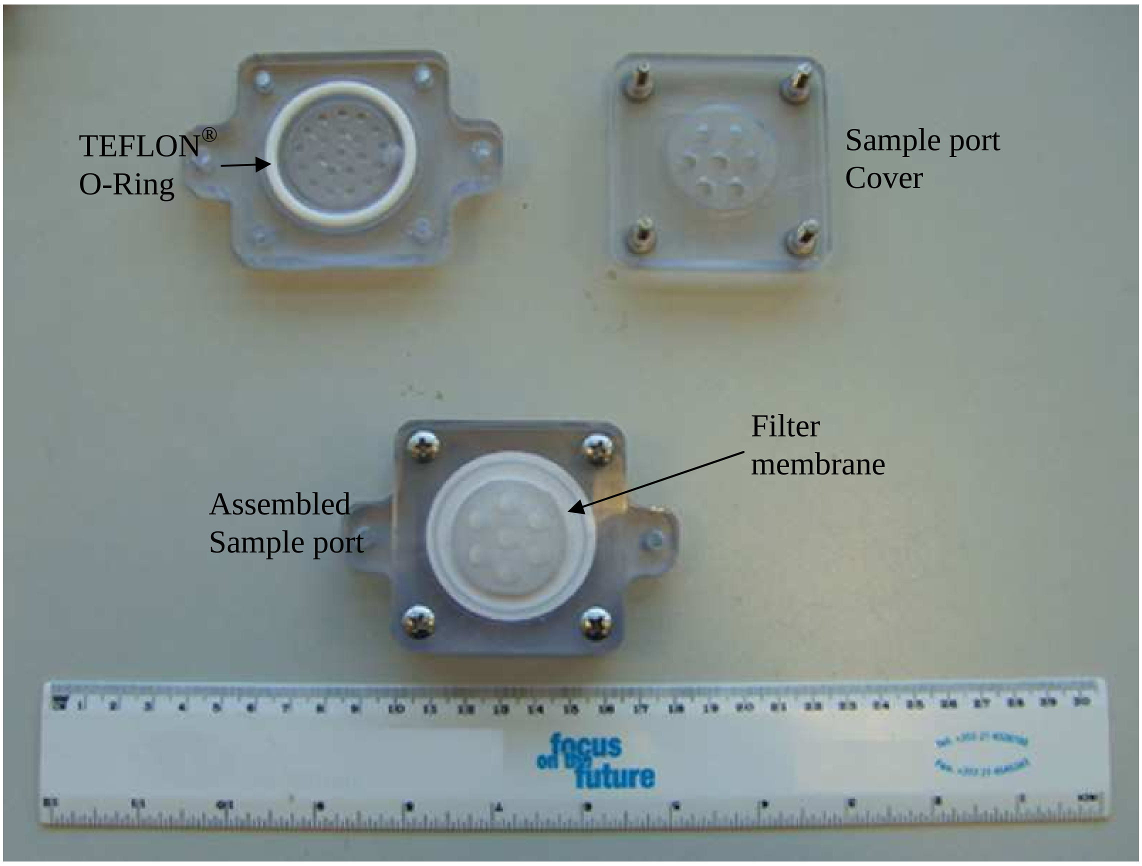


Figure 3.11: Components of sample port

Chemical storage

The reagent, calibration solutions and cleaning solutions were stored in five square 250mL high density poly ethylene (HDPE) bottles (NAlGENE 2018-0250, Nalge-Nunc, USA). Two square 1L HDPE bottles were used for storing waste chemicals (NALGENE 2018-1000, Nalge-Nunc, USA). Both types of bottle had 38 mm screw tops.

The bottle caps used (Q-Series, Bio-Chem Valve) fit a 38 mm thread and had ¼"-28 UNF ports and were fitted with TEFLON® check valves which allowed air into the bottle to replace the liquid pumped out without letting gasses from the chemicals to escape and corrode the systems components (see Figure 3.12).

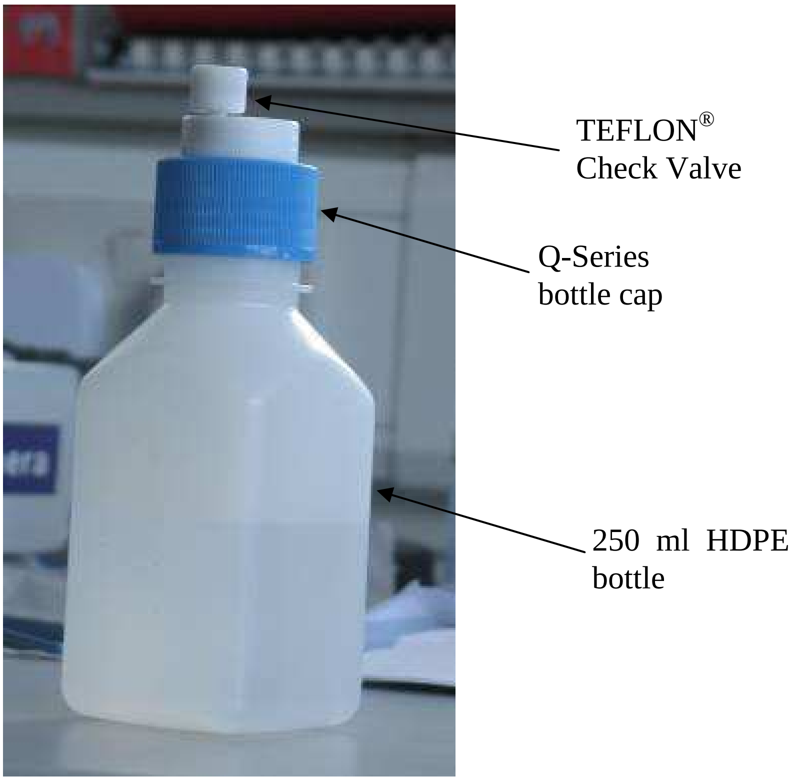


Figure 3.12: 250 ml HDPE bottle with Q-series bottle cap and TEFLON® check valve

Tubing and interconnects

TEFLON® tubing of 0.3 and 0.8mm ID with an outer diameter of 1/16" was used for the fluidic connections between components. Tubing made from TEFLON® has good strength and is chemical resistant while having a small inner and outer diameter and is still flexible. Tubing like PEEK is too rigid to be used. TYGON® is flexible but has a large outer diameter and is used mostly on barb connectors as it is too flexible to be fitted with a nut and ferrule.

The end of the TEFLON® tubing was fitted with a cone shaped ferrule and a ¼”-28 UNF nut (Omni-Lok, Omnifit, UK). The fitting could be assembled by hand and required no tools other than a scalpel for cutting tubes to the correct lengths. The end of the tube was cut flat then the nut was passed over the tube. The ferrule was then placed on the end. Nut was then screwed into a flat bottomed ¼”-28 UNF port. The ferrule was squeezed against the tube holding it securely and the end of the cone was squeezed against the bottom of the ¼”-28 UNF port creating a seal

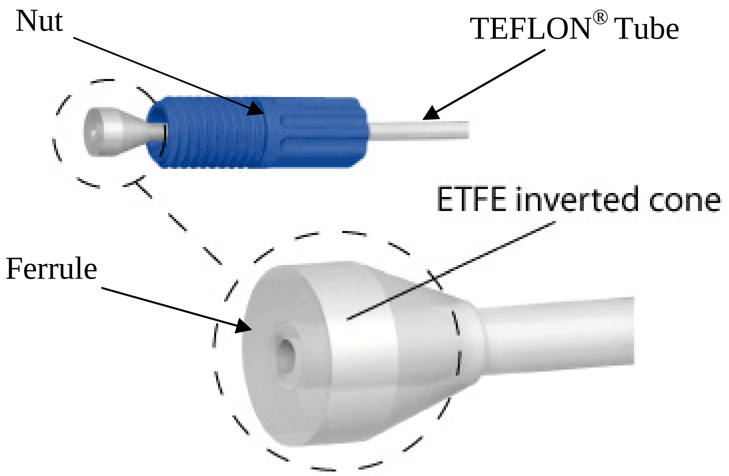


Figure 3.13: Diagram of Omni-Lok Inverted Cone Fitting System (Omni-Lok, UK)

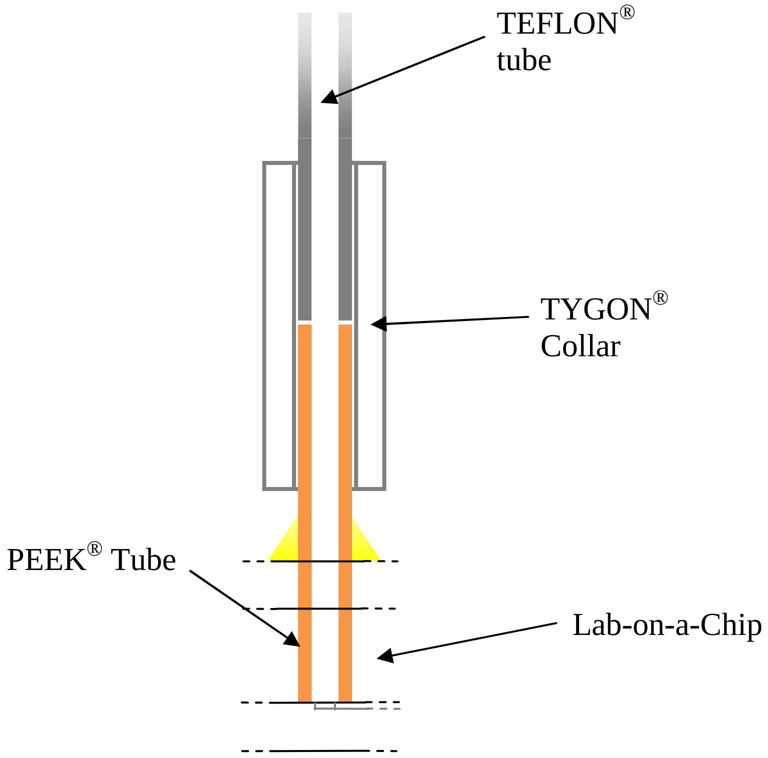


Figure 3.14: Connection of TEFLON® tubing to Lab-on-a-Chip

Six lengths of 0.8 mm inner diameter tubing were cut to 250 mm in length and both ends were fitted with the ¼”-28 UNF nut and ferrule. The tubing was used to connect the sample port and the bottles to the six pumps. Six 120 mm lengths of 0.3 mm inner diameter tubing were cut and one end was fitted with the ¼”-28 UNF nut and ferrule. The other end was connected to the LOC by using a 1.5 mm inner diameter section of TYGON® tubing (Upchurch Scientific, USA), (see Figure 3.14).

3.2 Reagents and standards

The solutions required for the completed analyser were 250 ml of reagent, 250 ml of 10 mg/L phosphate standard and 750 ml of a blank 0 mg/L phosphate standard. 250 ml of the blank standard was used in calibration and the other 500 ml was used as a cleaning solution. The chemicals used in the analyser were prepared in the same way as McGraw et al. (2007) and Cleary et al. (2008). All solutions were prepared with Milli-Q deionized water (Milli-Ro Plus 30 System, Millipore, USA).

3.2.1 Blank and cleaning solution

The blank and cleaning solution were composed of the same solution. It was made by mixing 1 L of deionized water with 2.25 ml of sulphuric acid (Fisher Chemicals, UK). The sulphuric acid was used to preserve the blank and cleaning solution.

3.2.2 Phosphate standards

A 50 mg/L stock solution of phosphorus, in the form of phosphate, was made by dissolving 0.6585 g of anhydrous potassium dihydrogen phosphate (Fluka, Switzerland) in 1 L of deionized water. 2.25 ml of sulfuric acid (Fisher Chemicals, UK) was added to preserve the stock solution with the blank and cleaning solution. 0 – 50 mg/L standard solutions were prepared from dilutions of the stock solution with deionized water.

3.2.3 'Yellow' Reagent

The reagent was prepared by dissolving 7.143 g of ammonium molybdate (A-7302, Sigma-Aldrich, UK) and 0.358 g of ammonium metavanadate (20555-9, Sigma-Aldrich, UK) in 95 mL of concentrated hydrochloric acid (37% wt. in water) (926, BDH Laboratory Supplies, UK) and 905 mL of water.

3.3 Electronics for analyser

3.3.1 Design of LED-photodiode spectrophotometer

An integral part of the phosphate analyser's design was an LED-Photodiode spectrophotometer. As stated before the compact size and low power consumption make the design ideal for portable analysers. Such spectrometers are used successfully in phosphate monitors (TresCon OP210, WTW, Germany) and handheld colorimetric instruments (POCKET Colorimeter II, Hach-Lange, Germany).

In order to design the LED spectrophotometer the characteristics of the 'yellow' phosphate reagent had to be quantified. Figure 3.15 shows the absorbance spectra for 12 separate solutions of phosphate ranging from 0.025 mg/L to 10mg/L. Each of the solutions was mixed at a 1:1 ratio with the phosphate reagent in a 10 × 10 mm square cuvette before the absorbance is measured by a benchtop UV-Vis spectrophotometer (Agilent Technologies, USA). The absorbance of deionised water was also included in the experiment as a reference.

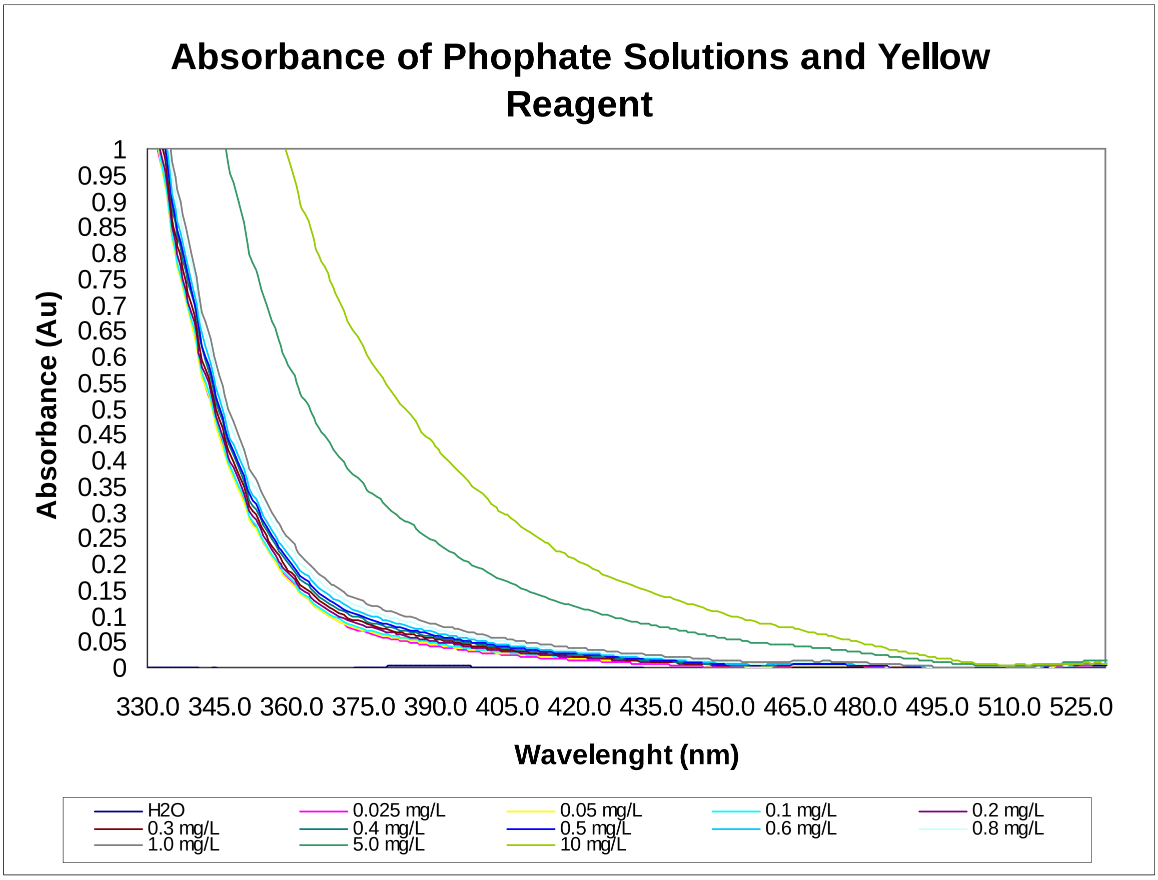


Figure 3.15: Absorbance spectra of varying concentrations of phosphate mixed with reagent

In plot it can be seen that solutions of phosphate with the reagent absorb light below 400nm and that this absorbance is proportional to the amount of phosphate in the solution as predicted by the Beer-Lambert law. The absorbance increases exponentially as the light wave length decreases. In order for the spectrophotometer to be suitable for use with the yellow reagent an LED that emits light at a peak wavelength that is absorbed by the reagent had to be selected. This LED had to be coupled with a photodiode that was sensitive to the peak wavelength of the LED.

An LED of peak wavelength of 375 nm was chosen for the spectrophotometer (NSHU590A, Nichia, Japan) (see Figure 3.16). This wavelength was chosen as it is below 400 nm but not below 360 nm where the absorbance of the phosphate reagent increases rapidly. A wavelength below 360nm would not be desirable as the peak wavelength of an LED varies with temperature and this would cause an increased level of noise in the absorbance measurement.

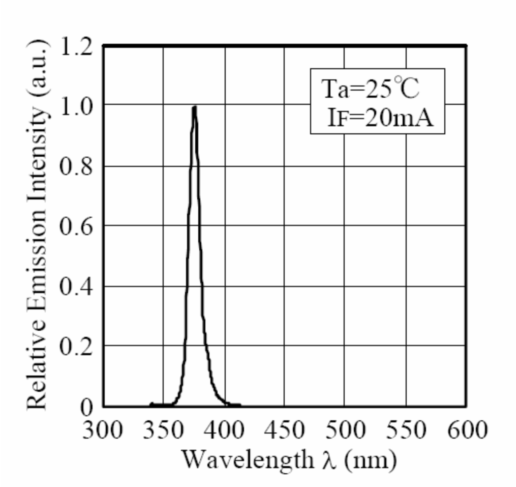


Figure 3.16: Emission spectrum of UV LED (NSHU590A, Nichia, Japan)

To limit the effect of ambient light on the spectrophotometer a photodiode that is sensitive to UV light was chosen. Figure 3.17 shows the responsivity of the UV photodiode (EDP-440-0, Roithner Lasertechnik, Austria). It has a peak responsivity of 440 nm and a bandwidth that includes 375 nm required for the photodiode to function with the UV LED chosen.

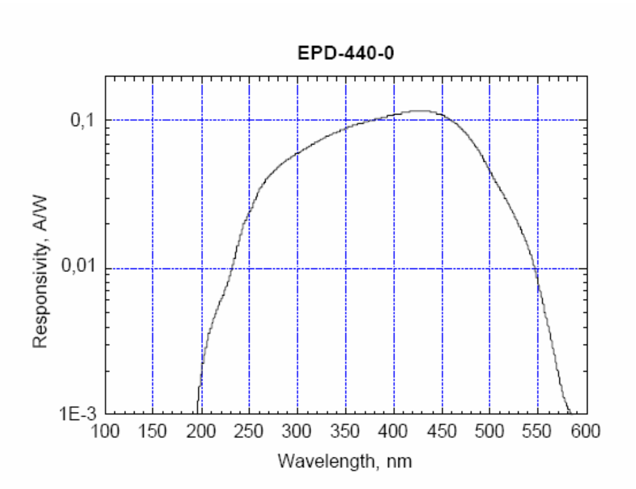


Figure 3.17: Responsivity spectrum of photodiode (EDP-440-0, Roithner Lasertechnik, Austria)

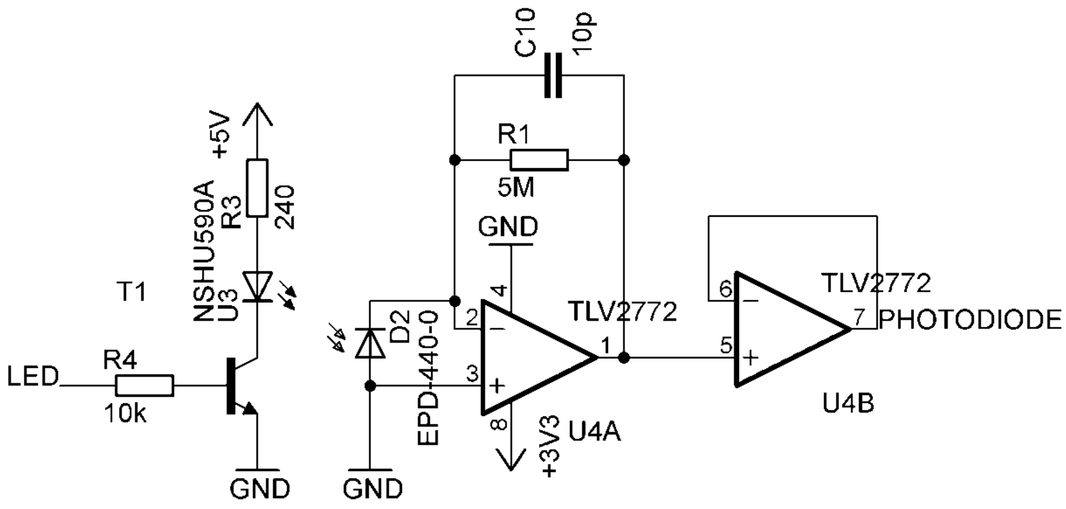


Figure 3.18: Trans-impedance amplifier circuit

The LED was switched by a bipolar transistor. The voltage power supply was smoothed and held at 5V by a linear voltage regulator (TPS76950DBV, Texas Instruments, USA). The photocurrent of the photodiode was converted to a voltage and amplified by a trans-impedance amplifier, (see Figure 3.18). The operational amplifier (TLV2772, Texas Instruments, USA) was powered through a 3.3 V linear voltage regulator (LP2985-33DBV, Texas Instruments, USA) to give an output range of 0 – 3.3 V.

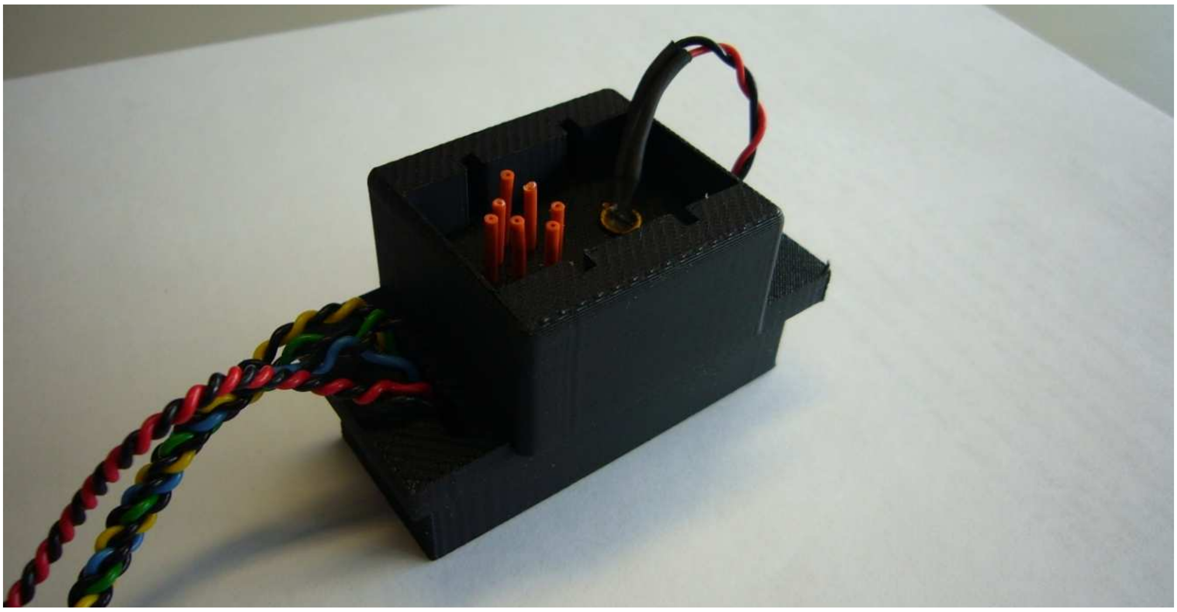


Figure 3.19: Holder for Lab-on-a-Chip and spectrophotometer

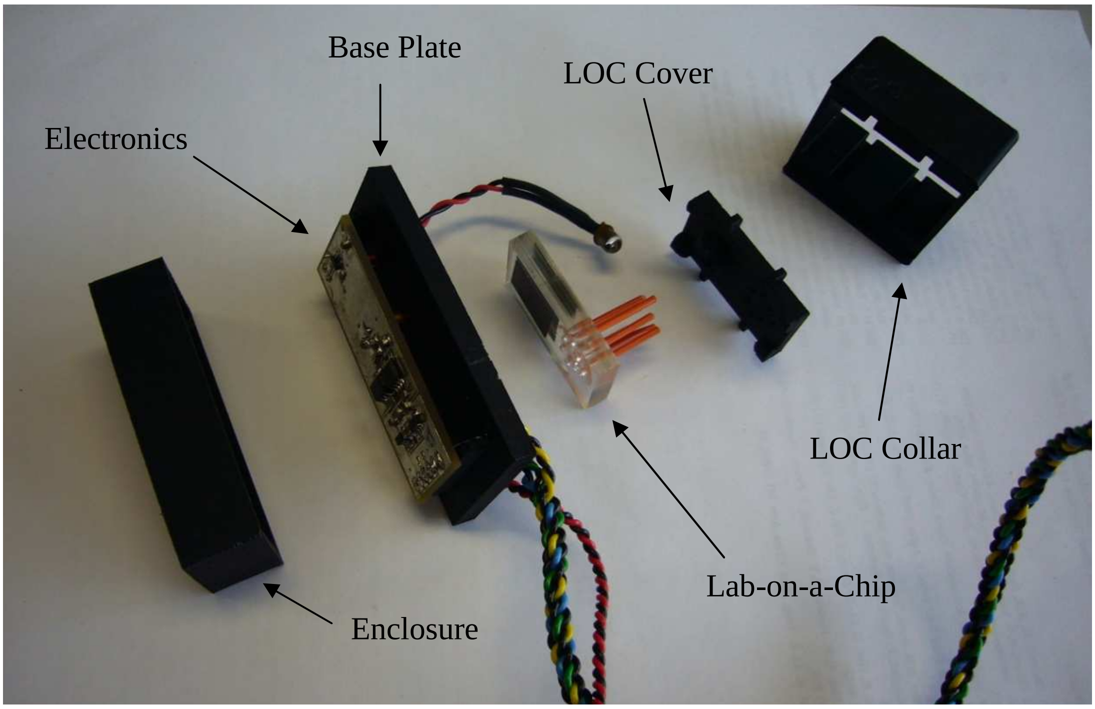


Figure 3.20: Components of holder for LOC and spectrophotometer

The LED-Photodiode spectrophotometer had to be coupled with the cuvette in the LOC. A 3D printed holder was designed to do this (Dimension SST 768, Dimension Printing, USA). This method was also used successfully for the alignment of fibre optics by Scarmagnani et al. (2008). The holder was made up of four parts, a base plate, an enclosure for the electronics, a collar for the LOC and a cover for LOC (see Figure 3.19 and Figure

3.20). The base plate held the photodiode and electronics in place. The enclosure was placed over the electronics to protect them. The collar held the LOC in place over the photodiode and the LOC cover held the LED in place over the LOC. The holder was printed in black to block ambient light from interfering with the spectrophotometer.

The arrangement of the LED and photodiode in relation to the cuvette in the LOC is shown in Figure 3.21. Two masks in the form of a 10 × 10 mm black insulation tape (Radionics, Ireland) with a 1mm diameter hole punched through the centre were placed on either side of the LOC. This was done so that the majority of the light entering the photodiode passed through the solution held in the cuvette.

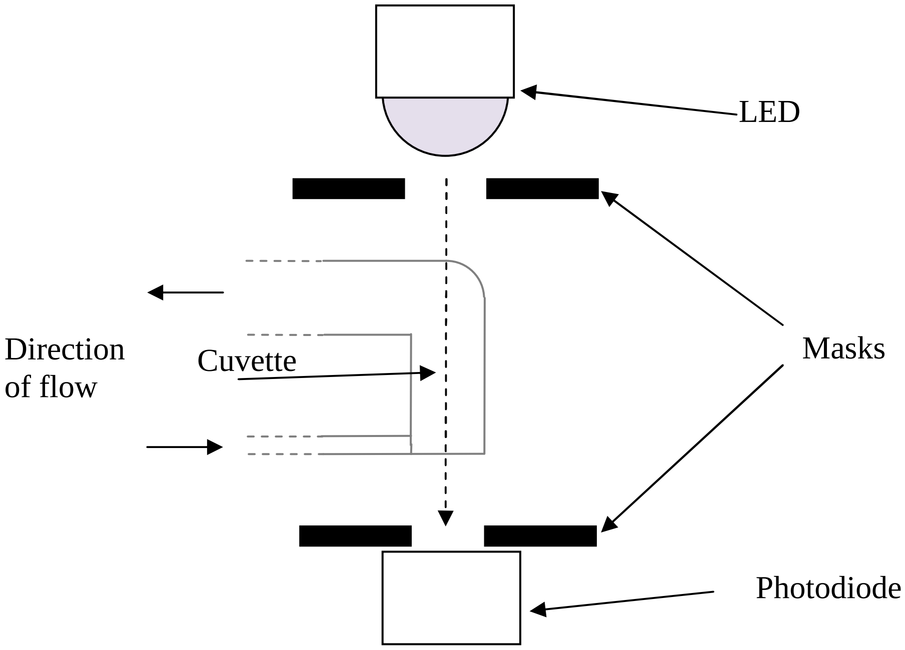


Figure 3.21: Positioning of LED and photodiode relative to cuvette

3.3.2 Temperature sensor

The temperature sensor used for the analyser was the LM335 precision temperature sensor (National Semiconductor, USA). This temperature sensor was used as it is easy to calibrate and its output is on a linear scale. The sensor was integrated into the holder for the LOC and spectrophotometer to give the temperature of the fluid in the LOC.

3.4 Standard operating procedure of sensor

For the analyser to accurately detect phosphate it must go through a set procedure to complete the assay. Before the analyser was used for detecting phosphate it was primed to remove any air that may be present in the LOC and all the tubing in the fluidics. This was done by actuating each of the analyser's pumps in sequence until no more air is passes through the outlet of the LOC. This process is semi automatic, the used initiates the priming sequence and inspected the outlet tube until no more air passes from the outlet.

When performing a phosphate assay the first step was to purge the tubing leading from the sample port to the sample pump. This replaced the sample water from the previous assay with new sample water. The length of the tubing was 250 mm this equates to roughly 125 µL of fluid. In order to purge this volume the sample pump was actuated 10 times which purged 200 µL of fluid. Next the LOC was purged with cleaning solution. This is done by simultaneously actuating the two pumps for cleaning solution thus purging the LOC with 40 µL.

Then the reagent and phosphate standard, a solution of known concentration of phosphate, were simultaneously injected into the LOC and allowed to react. The reaction time is dependent on temperature as demonstrated by Cleary et al. (2008). Figure 3.22 shows how the reaction reaches equilibrium after a period of time at 10º C. For this reason the analyser was programmed to wait a number of minutes in order for the reaction to complete. The reaction time varies between 2 and 60 minutes.

After the phosphate standard is reacted an absorbance measurement was taken. The LED emitter was pulsed for 1 second so that it warms up. Then the LED was pulsed for 100 ms during which time the ADC connected to the photodiode transimpedance amplifier output was sampled. 100 ms was as this was approximately the amount of time it takes for the ADC to take a sample, The LED was pulsed and the ADC sampled a further 127 times and the mean of all the ADC samples was calculated. The LED is pulsed to minimise the amount of time the LED is on, thus reducing the current consumed. The number returned from the ADC ranges from 0 to 4095 depending on the intensity of the light measured. Although not necessary, the calculated mean was then subtracted from 4095 to give the inverse of the light intensity reading. This gives a pseudo-absorbance measurement that makes the raw data from the spectrophotometer easier to read. This number is then stored and the LOC is purged with 40 µL of cleaning solution.

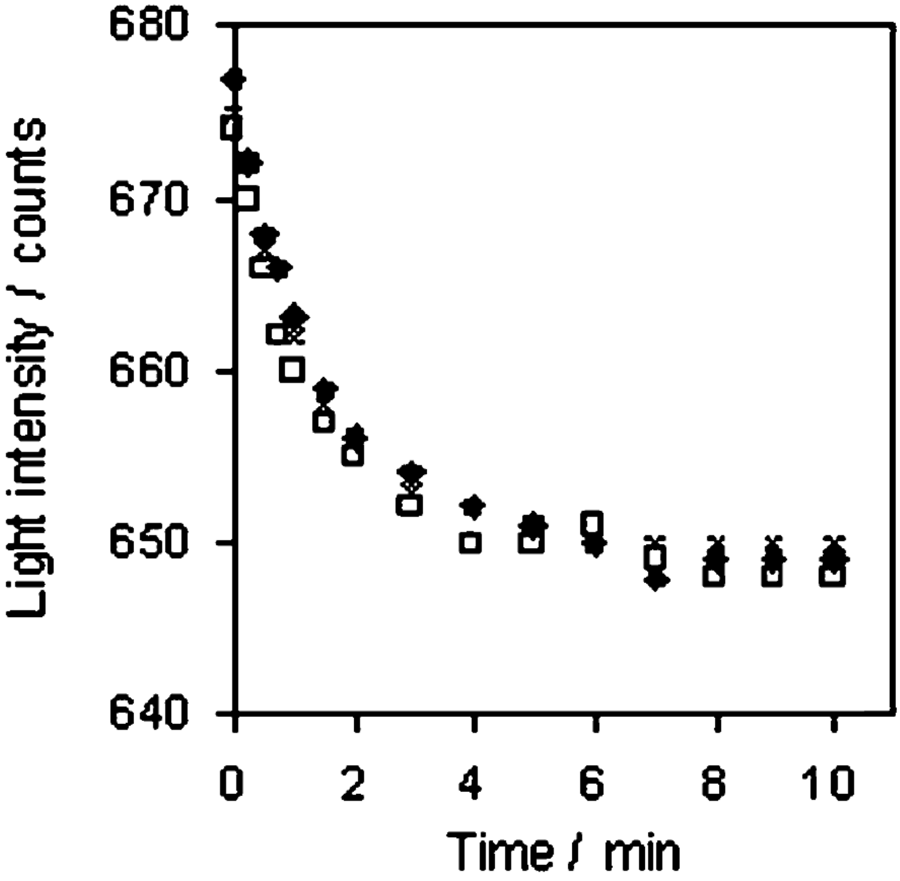


Figure 3.22: Reaction time of yellow reagent at 10º C (Cleary, 2008)

The sample and reagent were then injected into the LOC and allowed to react for the same amount of time as the standard. When the solution had reacted the absorbance was measured by the LED-Photodiode spectrophotometer using the same procedure used when measuring the reacted standard. The mean of 128 measurements was calculated the inverse of this value was stored. The LOC was purged for a third time with 40 µL of cleaning solution.

The final step was to inject the blank and reagent into the LOC. The solution was left to react for the same amount of time as the standard and the sample. After which the LED-Photodiode took 128 measurements and the mean was calculated. The inverse of the mean was stored. The final action was to inject 40 µL of cleaning solution into the LOC to purge it. The analyser was then left idle until the next assay. Figure 3.23details the operation of phosphate analyser.

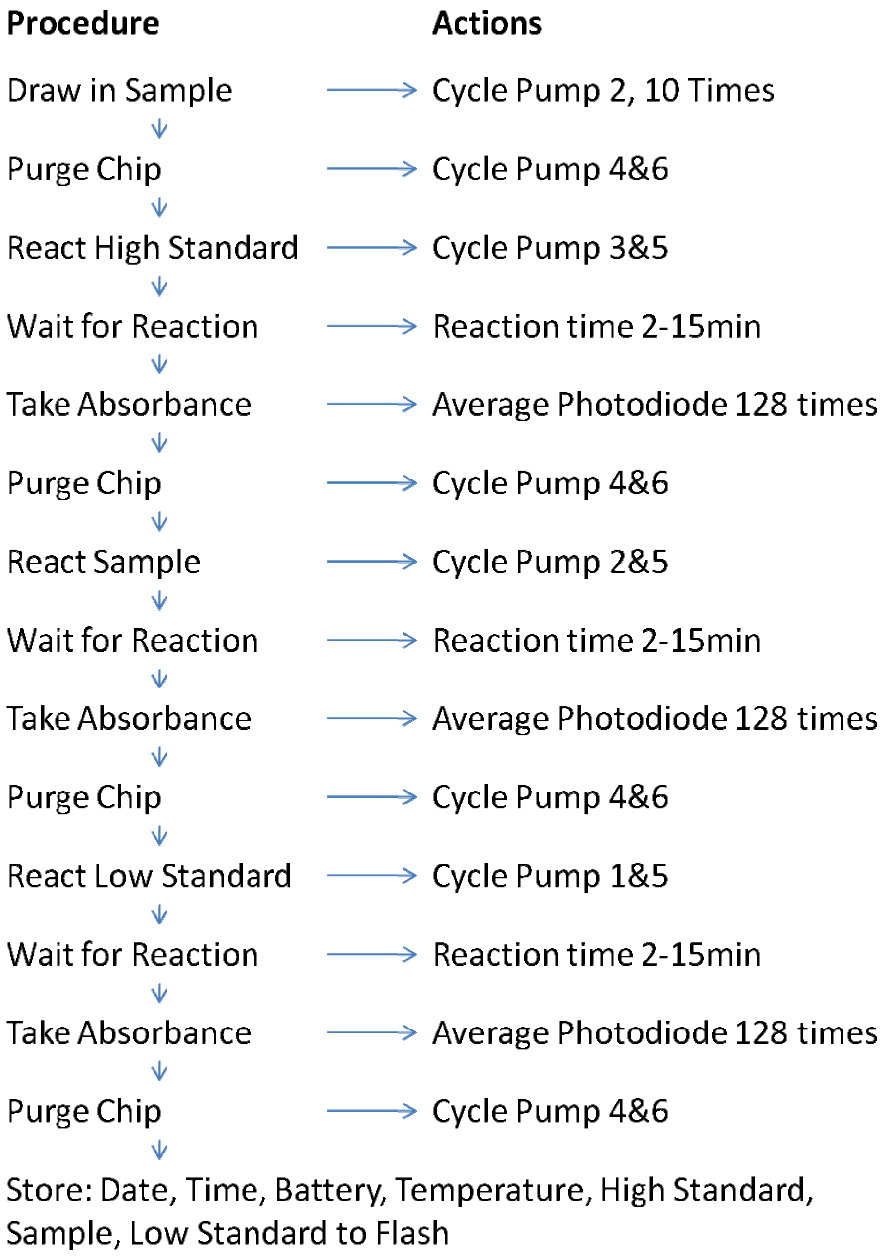


Figure 3.23: Flow chart of standard operating procedure of phosphate analyser

3.5 Power Supply

The analyser's power was provided by a 12V 7Ah lead acid gel cell (Maplin Electronics, Ireland). This type of battery was used as it has a high charge density, is able to be recharged, can supply the high amounts of current needed to operate the communications system and the solenoid pumps without damaging itself or dropping a significant amount of voltage. A 2A fuse was connected to the negative terminal of the battery to protect the battery from a short circuit.

3.6 Electronic control board

The function of the electronic control board was to carry out the functions of the analyser. It coordinates the actuation sequence of the pumps with data acquisition from the spectrophotometer. It also stores the data collected and operates the communication hardware. The board consists of a microcontroller unit (MCU), a flash memory chip, circuitry for operating the pumps and LED-Photodiode spectrophotometer, and circuitry for communicating with other peripheral devices.

The MCU used was the MSP430F449 (Texas Instruments, USA). This was chosen for its low power consumption during both active (280 µA) and sleep modes (1.1 µA). A complete open source compiler is also available (mspgcc.sourceforge.net). The MSP430F449 has, along with numerous digital input and outputs, an 8 channel 12 bit analogue to digital converter (ADC) with a 0 to 3.3V range and two universal asynchronous receive and transmit (UART) ports. The MCU can select to run off either of two clock crystals. One 8 MHz crystal was used when the MCU was active and a second 32,768 Hz was used when the MCU was in standby.

To control the phosphate sensors pumps the MCU utilised an array of Field Effect Transistors (IRLML2502, International Rectifier, USA). A Field Effect Transistor (FET) was used for two reasons. First, a minute amount of current is required to switch the transistor. Second, the dielectric in the FET protects the MCU from the relatively high current required for actuating the pumps, negating the need for opto-couplers. The FETs maximum current is 4.2 A and the outputs were connected to a screw terminal where the leads of the pumps were connected to.

To save power another array of FETs were used to control the power supply to all the peripheral devices. This meant that components were only on when they needed to be and when the system was idle the only part receiving power was the MCU.

The MCU’s ADC collects data from photodiodes trans-impedance amplifier circuit and the + terminal in the battery. The battery voltage was reduced from 12 V to the 3.3 V range of the ADC by a potential divider. The output of the temperature sensor was connected directly to the ADC. The LED was controlled by a GPIO (General Programmable Input Output) port on the MCU.

Data collected was stored on a flash memory IC (M25P20, ST Microelectronics, USA). The low power device (1 µA in standby mode) has a memory capacity of 2 megabits. The IC communicated via the Serial Peripheral Interface (SPI) protocol. This was chosen as the protocol can be easily implemented in software. Four GPIO pins were used to; send and receive data; send clock signal to synchronise the IC with the MCU; and a chip select output. If required additional SPI devices could be connected to the same clock and communication bus, each would only require an additional chip select output for the device.

The two UARTs were converted to the RS232 protocol with a RS232 driver IC (MAX3232, Texas Instruments, USA). The output of the driver was connected to a male DE9 connector making the control system DTE (Data Terminal Equipment) compliant. The goal of which was to allow peripheral devices that use the RS232 protocol to be connected directly to the electronic control board.

3.7 Communications

A GSM modem is used for communication when the system is deployed. The main reason for using this communications protocol is the extensive coverage of the GSM network in Ireland. Since the GSM infrastructure is already in place the only hardware cost in the communications is the GSM modem itself. Also the system can be deployed any where there is a GSM signal.

The modem used was the Siemens MC35iT (Siemens, Germany). The modem was connected to the MCU via the RS232 protocol and was operated by an extended AT command set which allows the GSM functionality to be used.

The GSM modems SMS (Short Messaging Service) is utilised to send data. The data can be sent to another GSM modem, a mobile phone or an online SMS server. The modem was only on when sending data, a period less than 30 seconds, otherwise it was completely powered down. The GSM modem selected could also be used to receive commands via SMS, a function that is not implemented in this work.

3.8 Enclosure

The systems enclosure provided two functions. To protect the systems components from the outside environment and it also allowed the components to be mounted securely. The enclosure must be tough enough to survive the external operating environment and must be radio transparent to allow the GSM modem to communicate. A final and important requirement was that it must have some sort of pressure equalisation system. Sample water drawn in must be stored in the waste containers. Therefore the volume of liquid in the waste containers increased during each assay. Air displaced during each measurement must be allowed to escape the waste container and the enclosure so the enclosure's internal pressure does not increase affecting the operation of the pumps.

The enclosure selected was the Pelican™ 1430 case (Peli Products, USA). This Polypropylene case was resistant to the chemicals used and was completely water tight. Although tough, the case could be easily machined and cut allowing the sample port to be drilled through and self tapping screws to be used to mount components. The GSM modem functioned normally inside the box. Also standard on Pelican™ cases is a GORE-TEX® membrane that acts as a pressure equalisation valve. GORE-TEX® is a form of molecular filter that allows gasses to pass through but not liquids. This ensures that the pressure inside the enclosure was equalised and the internal components were protected from moisture.

The internal dimensions of the Pelican™ 1430 is 345mm long, 146mm wide and 297mm deep. The case has a hinged lid that is 51mm deep leaving 246mm of dept in the bottom part. The components were mounted in three distinct sections.

The bottom of the case held the system's consumables, chemicals and battery. The battery sat in the centre to keep the case's centre of gravity below the carrying handle. The bottles were placed around the battery to hold it in place.

The electronics were mounted in the lid section of the case. The case lid has six mounting holds four of which were used to mount the control system board using self tapping screws. The GSM modem was secured to the lid using adhesive backed VELCRO®.

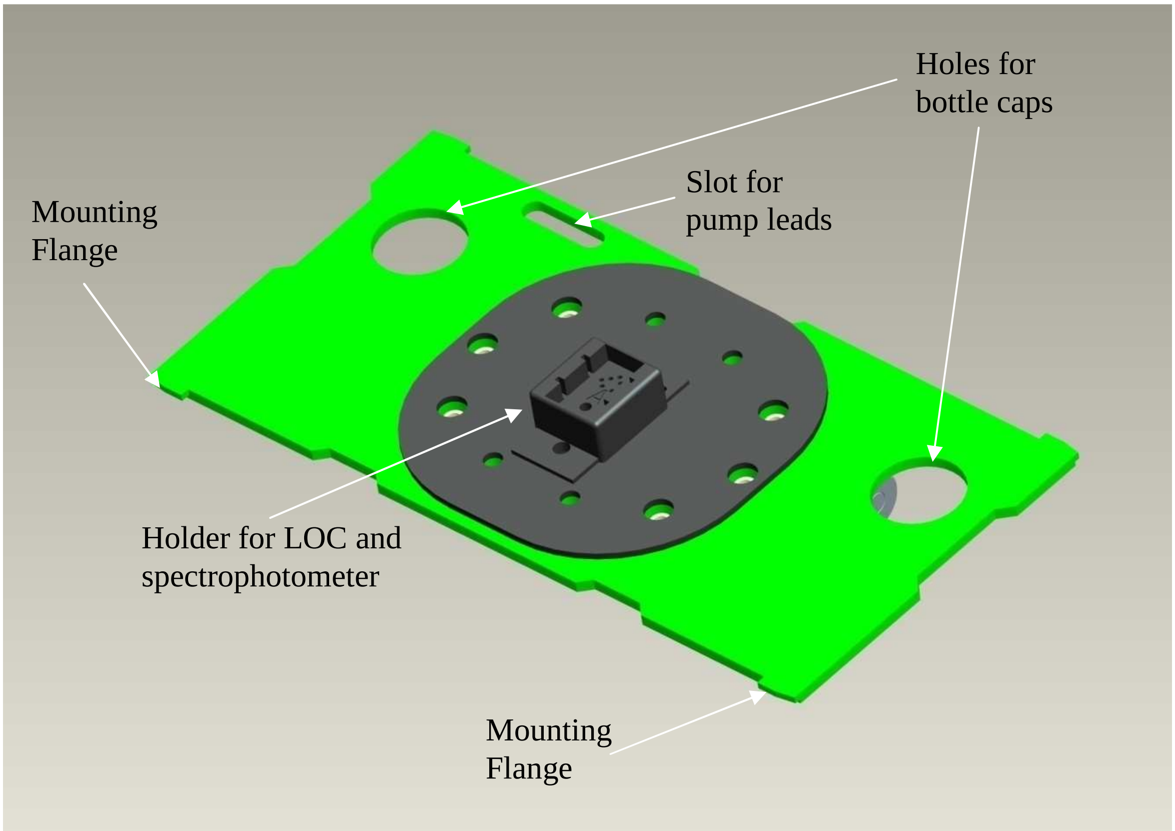


Figure 3.24: 3D model of mount (Top view)

A custom made ABS mount was manufactured using a FDM 3D printer (Dimension SST 768, Dimension Printing, USA), (see Figure 3.24 and Figure 3.25). The mount held all the solenoid pumps in position around the microfluidic chip holder, which fitted into a slot at the centre of the mount. It consisted of four sections as the 3D printer's work area was too small to print it in its entirety. Two plates supported the weight of the components. Another two sections, one for holding the pumps and another for the microfluidic chip holder fitted on either side of the plates. All the sections were fastened together using M6 nylon nuts and M6 × 40 nylon bolts (Radionics, Ireland). The mounting plate was fitted at the opening of the bottom section of the enclosure and is held in place by the mounting flanges. There were two holes for the bottle caps of the waste containers as these were taller that the depth of the bottom section of the enclosure.

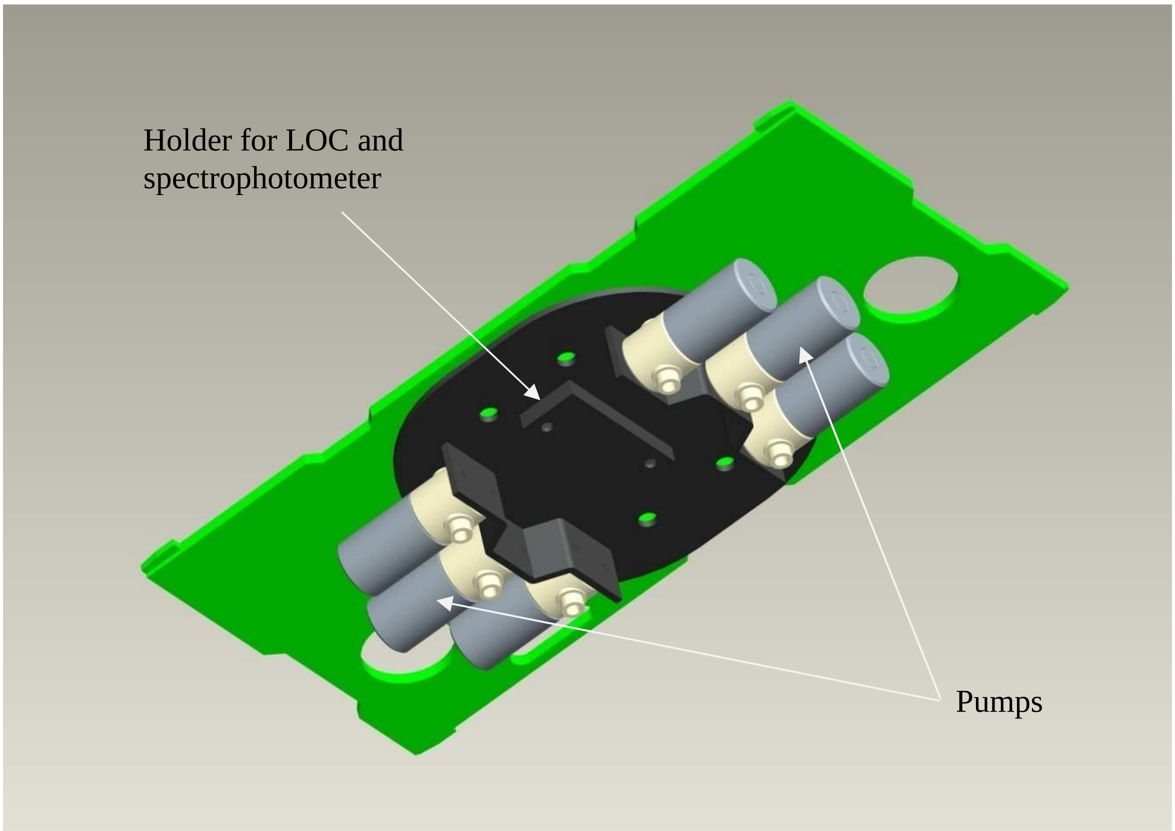


Figure 3.25: 3D model of mount (Underside)

The reason for the stratified mounting was to protect critical components in the event of a leak. Components at the bottom the enclosure were sealed so they are not affected by water. The electronics were most likely to be damaged by liquid so were mounted at the top. The remaining components were mounted in the middle of the case. One exception to this rule was the battery. It was placed at the bottom because of its weight. The battery however is sealed with only the contacts at the top being exposed. Liquid would have to fill the bottom 90 mm of the case to cause a sort circuit.

Figure 3.26 shows the layout of the components in bottom section of the enclosure. The five 250 ml bottles, two 1 L bottles and the battery were located here. The underside of the top plate is also visible in the figure. The pumps were secured in place using two M4 thread mounting holes located at the top of the pump. TEFLON® tubing can be seen connecting the bottles to the pumps.

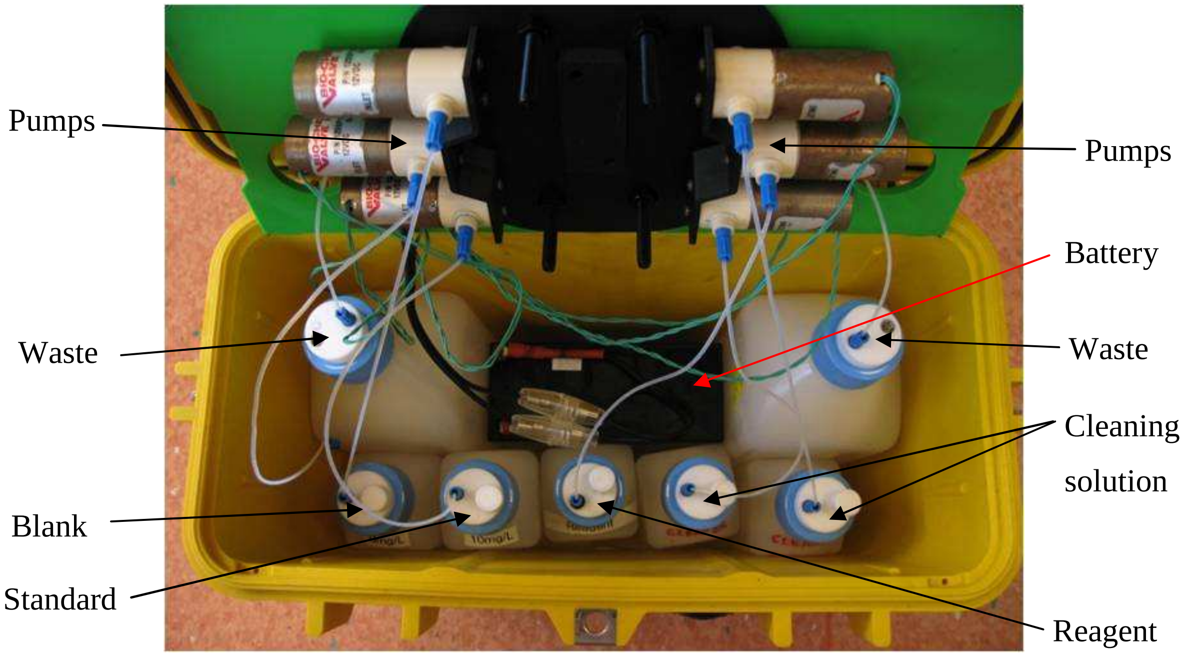


Figure 3.26: Inside view of phosphate analyser showing pumps and chemical storage

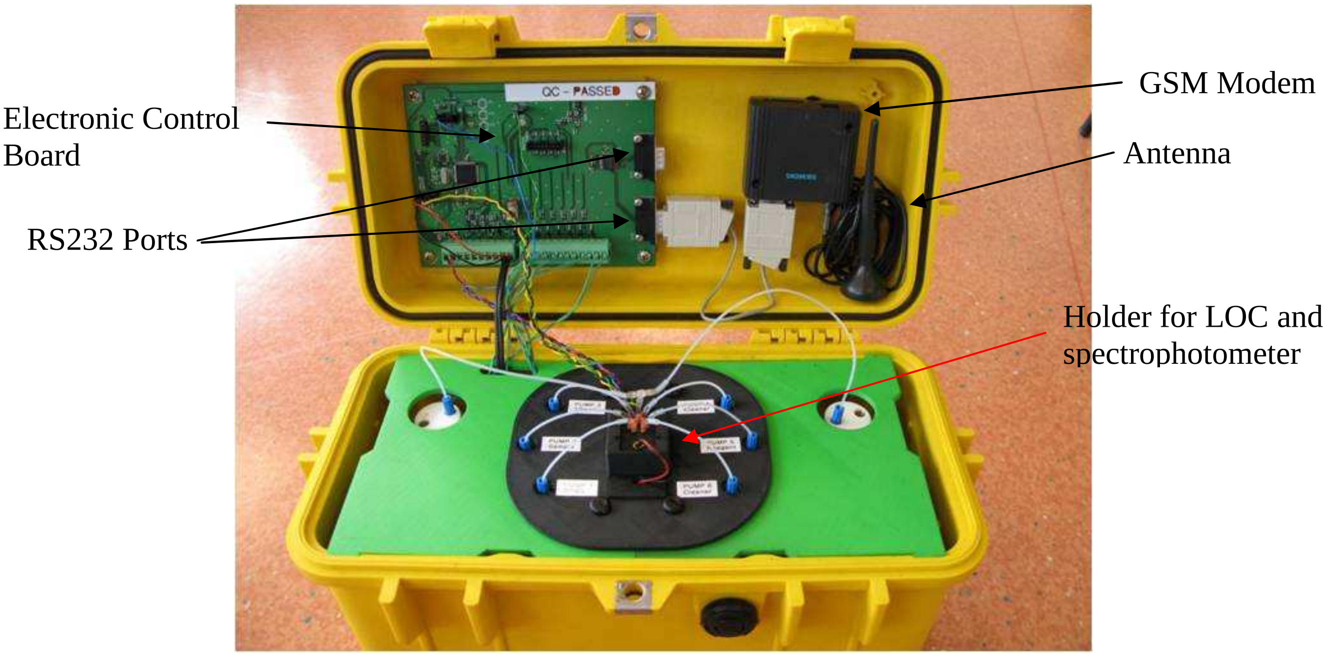


Figure 3.27: Inside view of phosphate analyser showing top plate and electronics

Figure 3.27 shows the phosphate analyser with the top plate in position. The location of the LED-Photodiode spectrophotometer is visible along with the TEFLON® tubing connected to the LOC. The electronic control board and GSM modem were located at the top of the enclosure.

The sample port was mounted on the side of the case. A hole was drilled through the enclosure wall and the ¼”-28 UNF port on the sample port is aligned with the hole. A further two holes were drilled through the enclosure for two bolts that fix the sample port to the enclosure. Polyurethane adhesive (Heinkle GmbH, Germany) was applied to the back of the sample port to seal it against the enclosure.

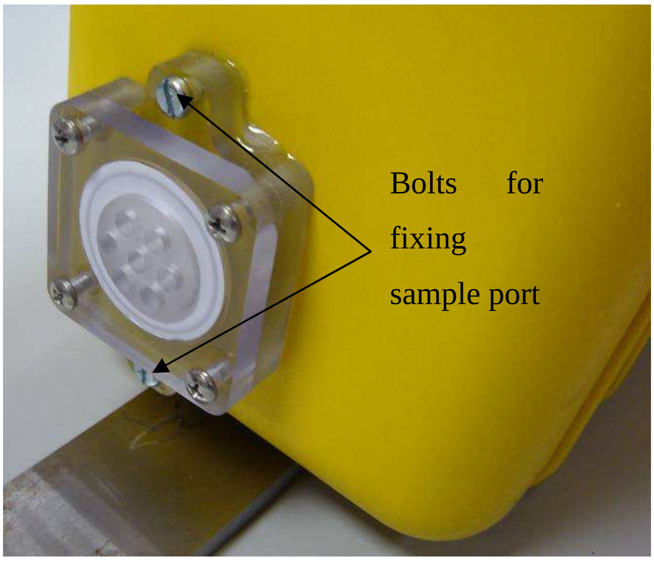


Figure 3.28: Sample port mounted on the outside of the enclosure.

The phosphate analyser is deployed by simply submerging the sample port beneath the water line (see Figure 3.29) Chapter 4 presents how the analyser was deployed at a waste water treatment plant.

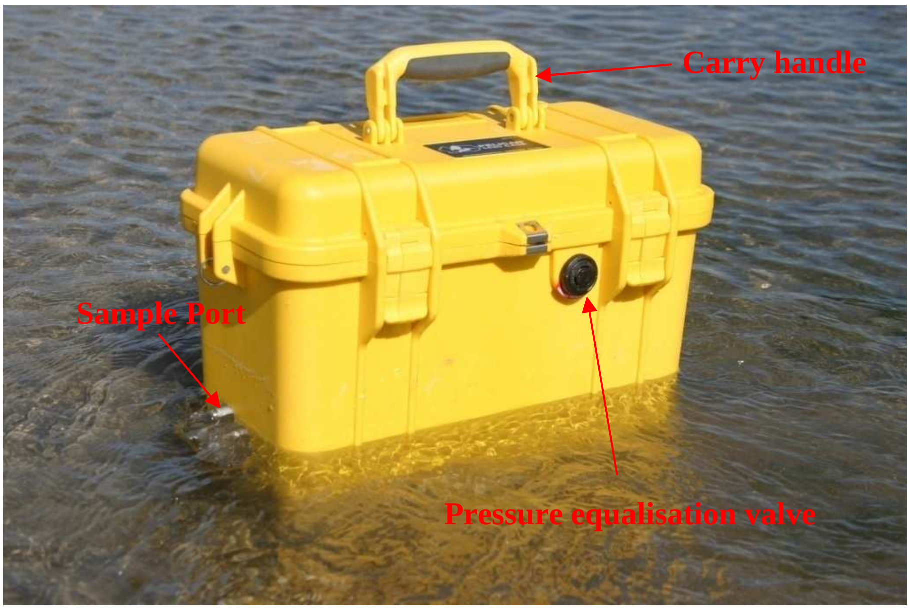


Figure 3.29: Complete phosphate analyser deployed

3.9 Software functionality

The majority of the phosphate analyser's functionality was derived from its software. It had its own operating system and the various functions are described here. All of the software was written in C and was compiled using the MSPGCC compiler.

Users could operate the analyser by connecting a serial cable to the top RS232 port. Hyperterminal™ a telecommunications program available as part of most versions of Microsoft Windows was used to communicate with the system. The control board communicated at a baud rate of 115200 bits per second, each character sent contained 8 bits, one stop bit and no parity check is done.

The analyser was powered on by pushing the on switch located in the centre of the control board down wards. If the RS232 cable was plugged in and the hyperterminal program was open with the right settings the analyser printed out a message saying that it is ready to start and the user was prompted to hit the return key to start the analyser and access the command line interface. Users manually controlled each of the analysers functions or could program a sample rate and the system could disconnected and ran autonomously.

The command line interface input and output information in the following format:

```
$CMD_NAME,DATA_1,DATA_2,DATA_3,DAT…   …DATA_N<CR>
```

Where '$' signifies the start of the command, CMD_NAME is the name of the command, DATA_1 to DATA_N are the data to be input or output to the analyser. All commands end with the carriage return character <CR> better known as the return key on a standard keyboard. A list of commands can be seen in Table 3.1.

Command	Format	Arguments
Set time and date	$POSET,<Date>,<Time>	<Date> - YYYYMMDD <Time> - HHMM
Program Analyser Parameters	$POPRM,<Start Time>,<Sample Rate>,<Reaction Time>	<Start Time> - HHMM <Sample Rate> - MM <Reaction Time> MM
Prime system	$POPRE,<Pump>,<Times>	<Pump> - Number of pump to be primed 'ALL' primes all pumps in sequence <Times> - Number of times the pumps is actuated
Purges Lab-on-a-Chip	$POPGE,<Times>	<Time> number of times cleaning solution is injected
Operate flash memory	$FLASH,<Operation>,<Argument>	<Operation> - Is either 'ERASE' to erase the flash device or 'READ' to read from the flash <Argument> - is used after 'READ' is used. 'ALL' prints the entire contents of the flash. 'LATEST' prints the most recent data. 'YYYYMMDD' prints all the data for that particular day.
Test all components	$TEST	

Table 3.1: Commands for phosphate analyser

The analyser maintains time by use of an internal real time clock. The 32,768 Hz crystal was used to keep the clock in time. An internal timer in the MCU was used to count the number of oscillations of the crystal. Every 32,768 oscillations of the crystal equates to 1 second. Each time the count reaches 32,768 one second was added to the RTC. There was a function for converting the incrementing of the seconds into minute, hours, day, months and years so the analyser maintains an internal calendar. The main function of the internal clock was to keep the sample rate and reaction time of the analyser accurate. The other function was to ensure that the data logged is accurately time stamped.

Data acquired by the analyser was stored in the flash memory chip. Data was formatted into a 16 byte data block that contained the number of the assay, the time and date of the assay, the battery voltage, the temperature, the absorbance of the phosphate standard, the absorbance of the sample and the absorbance of blank. The 2 Mbit capacity of the flash memory allowed 16,384 data blocks (or phosphate assays) to be stored, which was far in excess of the amount of data the analyser would generate. Data could be read and erased using the commands described in Table 3.1.

The GSM modem was operated automatically by the control board. Every five data blocks were sent by the SMS protocol to a phone number programmed into the analyser. There were no commands for changing the destination phone number or operating the GSM modem via the command line interface. The number was entered when the program for the analyser was compiled.

3.10 Chemical and Power Consumption

The analyser had two types of consumables, the power stored in the battery and the chemicals stored in the bottles. In order to determine the analysers life time both consumables are detailed here.

Table 3.2 details the fluid consumption of the analyser as per assay. The total amount of fluid consumed by the analyser was 480 µL and this volume was stored in the waste bottles after an assay was complete. This allows for 4166 assays to be performed before the waste containers were full. The fluid that limited the lifetime of the analyser was the cleaning solution. There was enough cleaning solution in the bottles for 3,125 assays.

Actions	Blank (µl)	Sample (µl)	Standard (µl)	Cleaner1 (µl)	Reagent (µl)	Cleaner2 (µl)	Fluid to waste (µl)
- Draw in new sample		200					200
- Purge LOC				20		20	40
- Mix Standard and Reagent			20		20		40
- Purge LOC				20		20	40
- Mix Sample and Reagent		20			20		40
- Purge LOC				20		20	40
- Mix Blank and Reagent	20				20		40
- Purge LOC				20		20	40
Fluid Consumed (µl)	20	220	20	80	60	80	480

Table 3.2: Fluid consumption of phosphate analyser for one assay

Table 3.3 details the power consumption of the analyser's main components and the data was obtained from the component's datasheets. In the case of the electronics board the current consumption was measured using a multimeter for both the standby and active states.

Item	mA (Standby)	mA (Active)	Voltage	mW (Standby)	mW (Active)
Spectrophotometer	0	21	5	0	105
Pump	0	320	12	0	3840
GSM modem	0	600	10	0	6000
Electronics Board	0.5	5	3.3	1.65	16.5

Table 3.3: Power consumption of analyser's main components

The power consumption for each of the components per assay is shown if Table 3.4. The total power consumed was 0.023 Wh per assay. Using the 12 V 7 Ah (84 Wh) battery in the analyser this equates to 3,652 assays per battery charge. As the total number of assays was limited to 3,125 by the chemicals stored, only 71.9 Wh of the battery power was used for performing phosphate assays. The rest of the battery charge was used to keep the analyser on standby. Using the remaining charge and with standby current of 0.5 mA at 3.3

V (0.00165 Wh) the analyser could theoretically be on standby for 7,333 hours or 305 days.

Item	Time (Standby)	Time (Active)	Times used	Total usage	Wh (Active)
Spectrophotometer	0	14	3	42	0.001225
Pump	0	0.25	24	6	0.006400
GSM modem	0	30	0.2	6	0.010000
Electronics Board	∞	44.25	27.2	1203.6	0.005517
					0.0231415

Table 3.4: Power usage of components per phosphate assay

3.11 Cost analysis

To compare the phosphate analyser to commercial technology a cost analysis is detailed in Table 3.5. The costs for the electronic board, LOC and 3D printed components were estimated as they were of custom manufacture. The total cost of the analyser's components came to €2,372. It was expected that cost of selling this analyser would range from €5,000 to €10,000.

Item	Manufacturer	Quantity	Price $	Price €	Total €
Metering Pumps	Bio-Chem Valve	6	210.3	151.42	908.50
Bottle Caps	Bio-Chem Valve	8	42.37	30.51	244.05
Check Valves	Bio-Chem Valve	5	33.58	24.18	120.89
1/4" 28 Nuts * 10	Bio-Chem Valve	2	19.12	13.77	27.53
1/4" 28 Ferules * 10	Bio-Chem Valve	2	19.68	14.17	28.34
250ml Bottles	Nalge-Nunc	6	8.08	5.82	34.91
1000ml Bottles	Nalge-Nunc	2	14.92	10.74	21.48
Top Plate	NCSR	1	0	100.00	100.00
Electronics	Beta Layout	1	0	500.00	500.00
GSM Modem	Siemens	1	0	250.00	250.00
Lead Acid Battery	Maplin	1	0	32.99	32.99
LOC Holder	NCSR	1	0	20.00	20.00
LOC Chip	NCSR	1	0	15.00	15.00
Pelican Case 1430	Pelican	1	94.95	68.36	68.36
					€ 2,372.05

Table 3.5: Cost analysis of completed phosphate analyser

CHAPTER 4. EVALUATION OF PHOSPHATE ANALYSER

The analyser described in Chapter 2 details a purely theoretical system based on the operation of ideal components. However the actual analyser presented in Chapter 3 had a number of random and systematic errors that affected its performance. This chapter details the testing of the analyser to measure the extent of these errors. It is demonstrated how the analyser uses the two point calibration to maintain accuracy over hundreds of assays. In addition the factors that determine the analyser’s precision were identified.

In order for the performance of the analyser to be ascertained it was run continuously to measure solutions of known concentrations of phosphate. The analyser was run continuously to show that its reliability over an extended period of time. 661 consecutive assays were performed over a period of 7 days.

Later in this to chapter it is demonstrated that the Lab-on-a-Chip phosphate analyser is a viable alternative to commercial monitors as it was validated against a commercial phosphate monitor of similar performance characteristics.

4.1 Set up of analyser

Phosphate standards of 0, 1, 2, 5, 10, 20 and 50 mg/L PO_4^{3-} were prepared as samples for the analyser. Each of the samples was stored in seven 250 ml HDPE bottles (Nalge-Nunc, USA). The analyser itself had 250 ml of 10 mg/L PO_4^{3-} phosphate standard, 250 ml of blank, 500 ml of cleaning solution and 250 ml reagent stored. The two 1 L waste containers were empty at the start of the experiment. The sample rate of the analyser was set to every 15 minutes and the reaction time was 3 minutes.

The sample port was not used in the experiments. Sample water was drawn directly from the 250 ml bottles positioned outside the enclosure using a 300 mm length of 0.8 mm inner diameter TEFLON® tubing (Omni-Lok, UK) connected directly to the sample pump with a ¼"-28 UNF nut and ferrule (Omni-Lok, UK).

The analyser started with a sample of 0 mg/L phosphate. Calibrations were carried out at sample number 84, 180 and 574. In a calibration 1 mg/L PO_4^{3-} was measured for four assays, then the sample was changed to 2 mg/L PO_4^{3-} for four assays. The samples were changed by removing the TEFLON® tube connected to the sample pump from one sample bottle and dipping it into the next sample bottle.

The same was done for 5, 10, 20 and 50 mg/L PO_4^{3-}. Each phosphate standard was measured four times before the next highest standard was measured. After the 50 mg/L standard was measured the sample was switched to 0 mg/L phosphate. In the third calibration the 50 mg/L standard was not used and the analyser measured the 0 mg/L phosphate standard after the 20 mg/L standard was measured four times. All the data was recorded to the phosphate analyser's flash memory chip and was retrieved after the experiment. The GSM modem was not used in the experiment and the analyser was powered by a 12V power supply.

4.2 Accuracy of the analyser

Figure 4.1 shows the raw data from the continuous run. The start of each of the calibrations is marked. There was an apparent relationship between temperature and the measurements of the LED spectrophotometer as reflected in the variation of the measurement of the blank. As detailed in the datasheet for the NSHU590A UV LED (see Appendix A), ambient temperature affects the intensity of the light output and peak wavelength thus affecting the output of the spectrophotometer. There was also a drift in the measurements of the LED-Photodiode spectrophotometer. This was most likely caused by staining of the Lab-on-a-Chip.

There are a number of spikes and variations in the raw data shown in Figure 4.1. These variations were averaged out so they have no affect on the determination of accuracy of the analyser. It is shown later in this chapter how these variations affect the precision of the phosphate measurement.

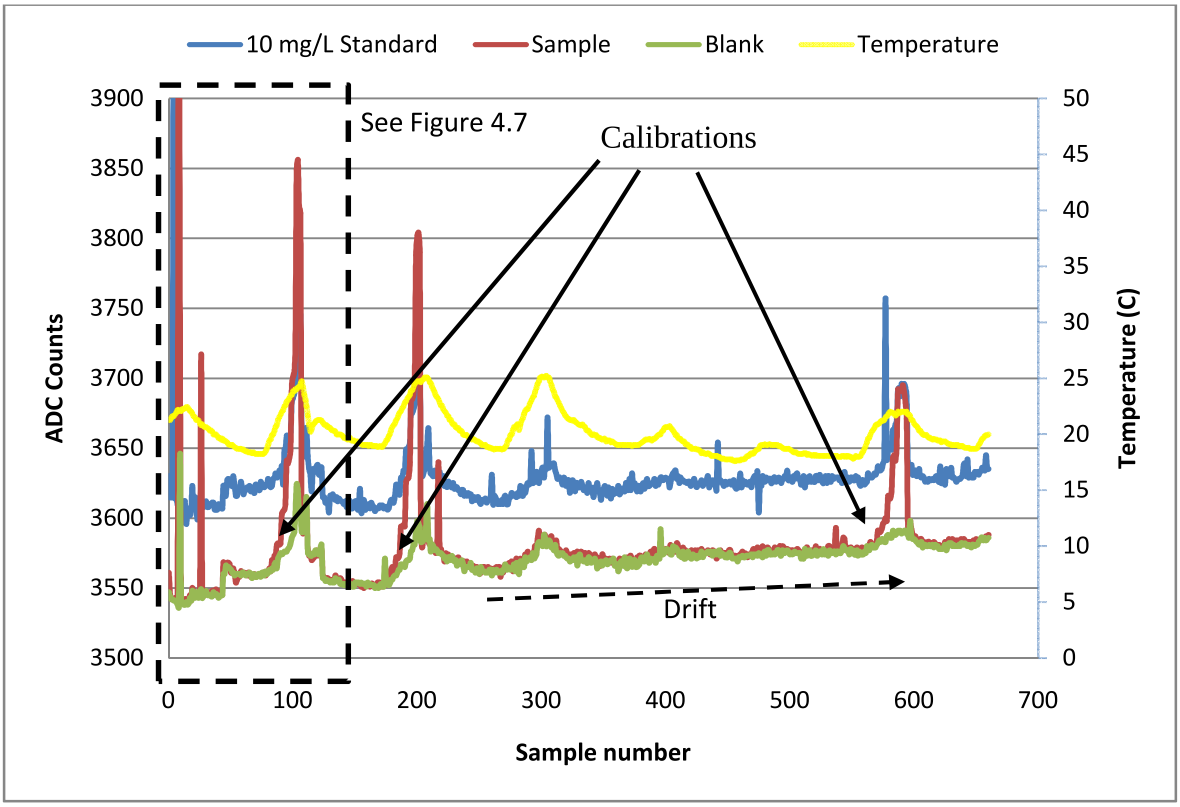


Figure 4.1: Raw data from long term experiment

4.2.1 Drift in analyser's output

To further investigate the affects of temperature and staining on the phosphate analyser measurement of the blank is plotted against the measured temperature (see Figure 4.2). It can be seen more clearly that the measurement had a dependence on temperature. However this dependence is not linear. Most of the data are scattered throughout the plot and not positioned along a single line as would be expected if temperature was the only factor affecting the drift. This is evidence that other factors, such as staining, cause drift in the analyser output.

In the experiment three calibrations were carried out. If staining was responsible for drift in the analyser it would be expected that the amount of staining would increase after measuring a high level of phosphate as an increased amount of vanadomolbdenumphosphoric acid is formed during these times. In the experiment there were three occasions where an excessive amount of phosphate was present in the LOC. This corresponds to the three drifts in the measurement of the blank. Although there is most likely other factors that affect the output of the phosphate analyser it is shown in Figure 4.2 that the two main causes of drift in the analyser were temperature and staining.

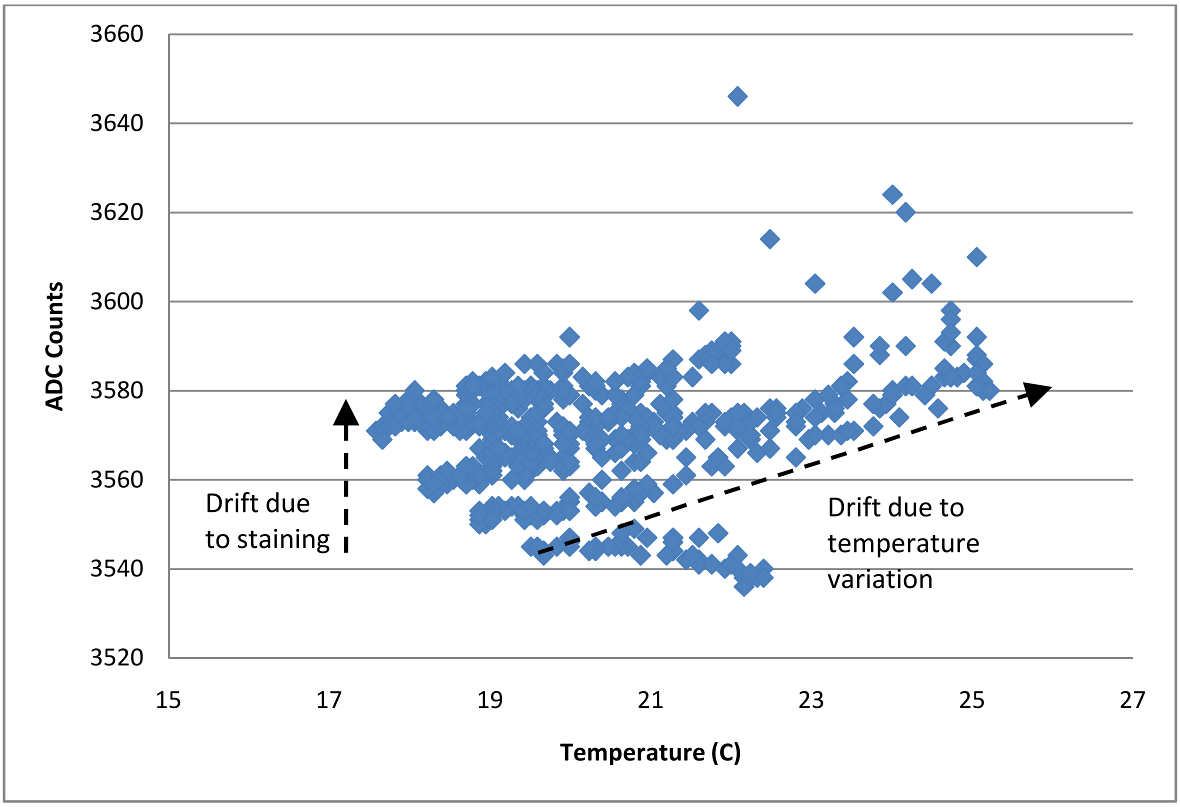


Figure 4.2: Measurement of blank versus temperature

4.2.2 Two - point calibration

It should be noted that since staining is related to the level of phosphate measured it cannot be compensated for by simply measuring the level of drift over time and subtracting it. When measuring phosphate in the environment the phosphate levels would fluctuate randomly and this would lead to the amount of drift on the analyser output being random.

In order to compensate for the variations in the analyser's output, known values were integrated into the output of the analyser in the form of a two point calibration. Each sample taken was compared to a blank and a phosphate standard. The two values were used to compensate for drift in the analyser's baseline and gain.

Figure 4.4 shows the effect of integrating the blank measurement into the measurement of the sample and standard. This was done by calculating the absorbance of the sample and standard in respect to the blank being the baseline value. Equation 4.1 and Equation 4.2 detail how the absorbance was calculated using the raw ADC counts for the measurements of the sample, standard and blank. The calculated data are plotted in Figure 4.3.

$$A_{sample} = log_{10}\left(\frac{ADC_{Sample}}{ADC_{blank}}\right)$$

Equation 4.1: Calculation of absorbance of sample

$$A_{standard} = log_{10}\left(\frac{ADC_{standard}}{ADC_{blank}}\right)$$

Equation 4.2: Calculation of absorbance of standard

The plot shows how the baseline of the analyser’s output was normalised. This is most evident in between calibrations when the analyser was measuring the 0 mg/L PO_4^{3-} standard, the output was close to zero where it is expected to be. What is shown when the absorbance of the standard is plotted is that there is a downward drift. This was a drift in the gain of the analyser’s output due to staining. It would be reasonable to suggest that there is a similar effect on the calculated absorbance of the sample. Figure 4.3 shows that the integration of only the blank measurement was not sufficient to fully stabilise the output of the analyser although it does show that calculation of the absorbance had a significant effect on improving the analyser’s output. Also visible was the sharp increase in the absorbance of the standard when samples with high concentrations of phosphate were introduced in to the analyser. This sharp increase was due to excessive staining of the LOC’s cuvette.

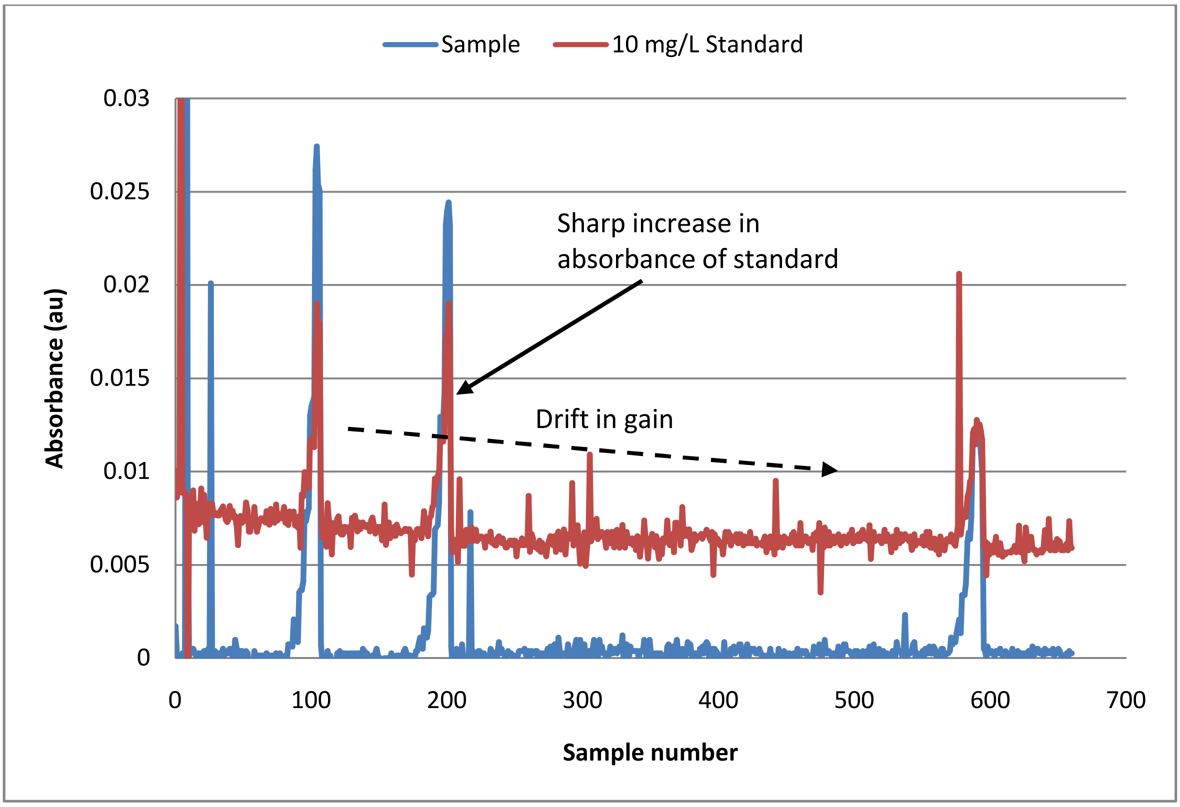


Figure 4.3: Normalisation of sample and standard by integration of blank measurement

Figure 4.4 shows plot of the analysers output after the absorbance of the standard was integrated into the sample measurement. The concentration of the sample was calculated using Equation 4.3. The known value of the phosphate standard used to calibrate the analyser (*standard*) which was 10 mg/L PO_4^{3-}. This was multiplied by the ratio of absorbance of the sample (A_{sample}) and standard ($A_{standard}$) to give the phosphate concentration of the sample.

$$sample = standard \times \left(\frac{A_{sample}}{A_{standard}}\right)$$

Equation 4.3: Calculation of sample concentration

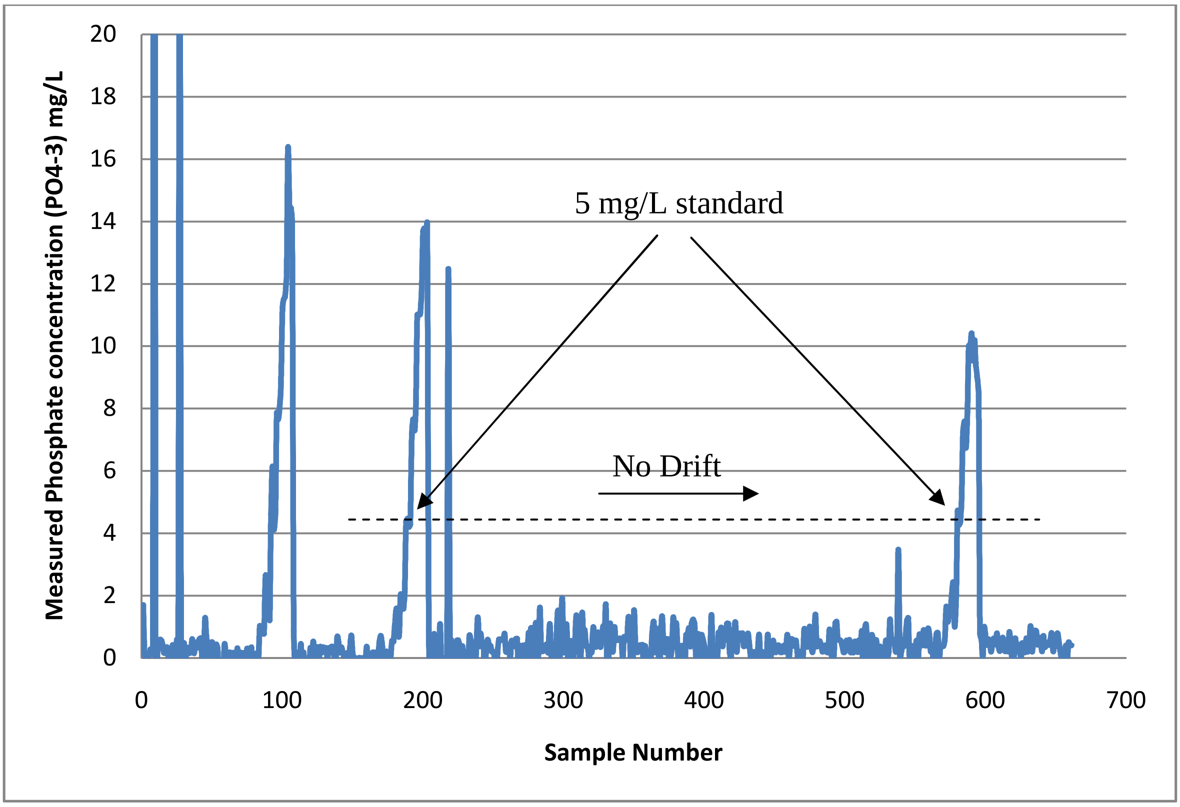


Figure 4.4: Data normalised by integration of blank and standard measurements

In the plot there is no drift between the second and last calibration run. This is particularly noticeable between the measurements for the 5 mg/L PO_4^{3-} standard. However, there was still an error between the measured concentration and the actual concentration of the sample.

4.2.3 Compensation of systematic errors

In order to understand the performance of the system and to quantify the deviation of the analyser's output from the expected performance the output was plotted against the actual concentration of the phosphate standards.

Using the normalised data plotted in Figure 4.4, all 12 measurements that were taken for each of the phosphate standards were gathered together and the mean was calculated. The mean of the measured concentrations are plotted against the actual concentrations of the phosphate standards (see Figure 4.5). It can be seen that the output of the analyser had a linear relationship to the actual concentration of the phosphate standards from 1 to 5 mg/L PO_4^{3-}. For higher concentrations of phosphate the analysers output became saturated. This non linearity was caused by the increase in absorbance of the standard when higher concentrations of phosphate were sampled (see Figure 4.3).

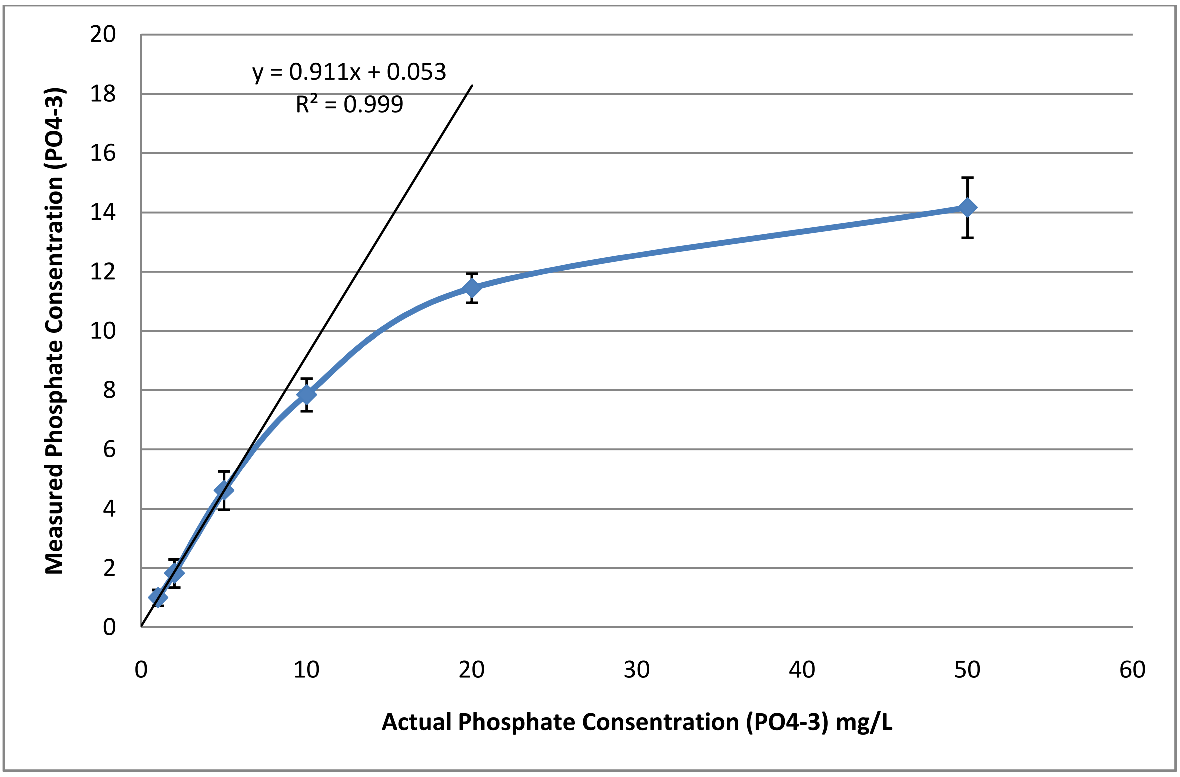


Figure 4.5: Calibration curve of analyser

The linearity of the analyser was calculated as *y = 0.911x + 0.053*. If the operation of the analyser was ideal the linearity would be calculated as y = x. The deviation from the ideal was an error in the gain and baseline of the analyser. The cause of this was most likely a difference in pressure between separate pumps as observed in Figure 2.3 in Chapter 2. In order that the output of the analyser was accurate it was compensated by using the inverse of the linearity (see Equation 4.4). All of the measured phosphate concentrations were compensated for and the results plotted in Figure 4.6. With the systematic errors in the analyser corrected the linearity was calculated as *y = x + 0.005*.

$$Actual\ Phosphate = \left(Measured\ Phosphate \times \frac{1}{0.911}\right) - 0.053$$

Equation 4.4: Calculation of actual phosphate concentration

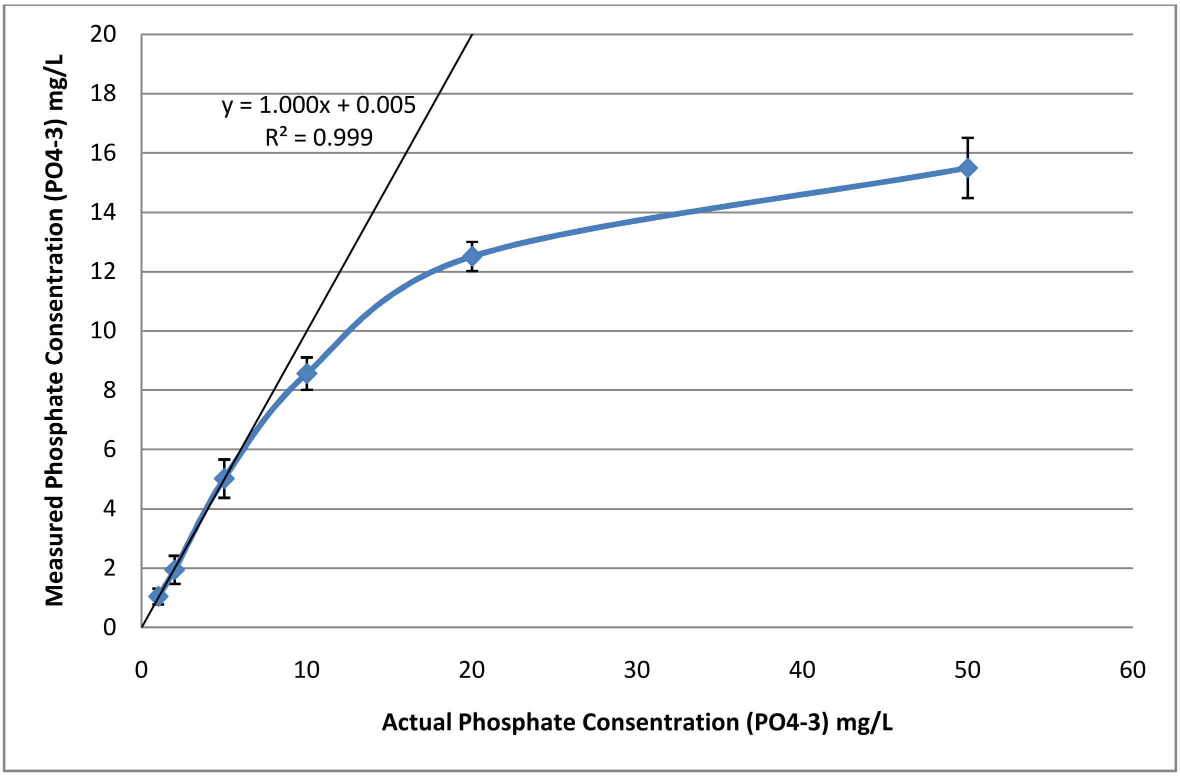


Figure 4.6: Calibration curve of analyser with systematic errors corrected

4.3 Limits of detection and precision

The precision of the analyser was determined by the stability of the output signal. There were a number of interferences that affected this stability, the stability of the spectrophotometer, the stability of the output pressure of the pumps while mixing, and bubbles that were present in the fluidic system.

Number	ADC Count	Mean	Standard Deviation
1	3542	3542.4	0.5164
2	3542		
3	3542		
4	3543		
5	3543		
6	3543		
7	3543		
8	3542		
9	3542		
10	3542		

Table 4.1: Stability of spectrophotometer

To investigate the stability of the optics, prior to the experiment 10 consecutive readings of the spectrometer where taken. These where done with the fully assembled phosphate analyser. Deionised water was injected into the microcuvette of the LOC before the experiment. Each reading was taken using the same method when the analyser is measuring absorbance described in Chapter 3. 128 reading were taken and the mean calculated. The results are shown in Table 4.1. The reading from the spectrophotometer only varies by one count showing that it was highly stable.

Figure 4.7 shows the first 150 samples of the experiment. What is seen in the first part of the plot are 4 spikes in the data. They appear on the measurement for the standard, sample and blank. These spikes were caused by bubbles passing through the analyser's fluidics. The bubbles were sometimes caught between the LED and photodiode in the spectrophotometer and they caused the emitter's light to scatter and not reach the photodiode. The output was the inverse of the measured light intensity so the presence of a bubble caused a high value. In the next assay the bubble was pushed out of the way of the spectrophotometer when pumps inject fluid into the LOC. The presence of bubbles caused the analyser to output spurious data and these points should be discarded.

To quantify the precision of the analyser the limit of detection (LOD) was calculated. Statistically it was assumed that the measurements taken by the LOC analyser had a normal distribution around the actual concentration of the phosphate sample. In a normal distribution 99.7 % of the phosphate measurements are within 3 standard deviations from the actual concentration of the sample. The LOD was defined here as 3 times the standard deviation of the measurement (Equation 4.5). This means that there is a 99.7 % certainty that measurement taken by the LOC analyser has a maximum error of the LOD quoted.

$$Limit\ of detection = 3 \times Standard\ deviation$$

Equation 4.5: Limit of detection

For the 20 readings marked on the plot in Figure 4.7 the phosphate concentration was calculated using Equation 4.1, Equation 4.2, Equation 4.3, and Equation 4.4. The standard deviation for the 20 measurements was calculated and multiplied by 3 to give the limit of detection which was given as 0.3 mg/L PO_4^{3-}. The maximum error in measurements taken by the LOC analyser characterised here was 0.3 mg/L PO_4^{3-}.

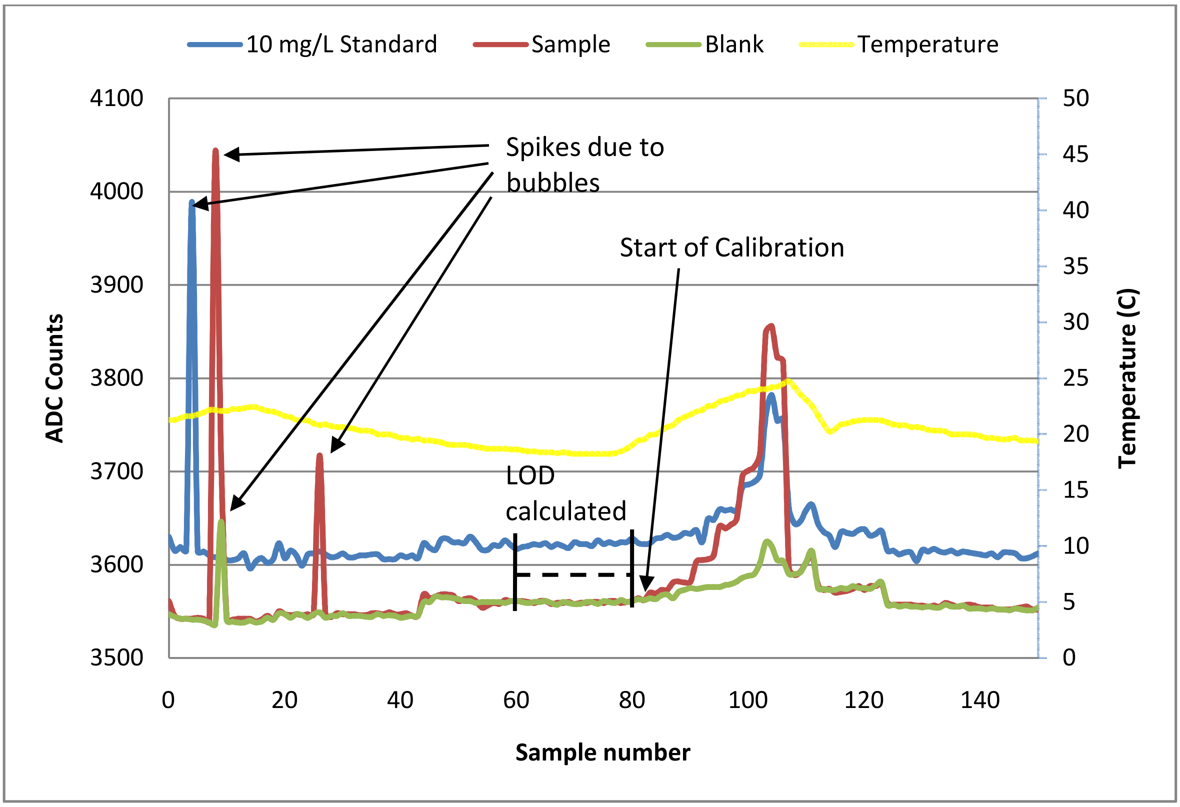


Figure 4.7: Details of bubbles in analyser's output

4.4 Validation against commercially available analyser

The analyser selected for the validation was the AZTEC P100 (Severn-Trent, UK). It has a limit of detection of 0.3 mg/L and a linear range of 0.3 to 7.5 mg/L PO_4^{3-}. This compares well with the LOC phosphate analyser's limit of detection of 0.3 mg/L and linear range of 0.3 to 5 mg/L PO_4^{3-} so it is used for the validation. An AZTEC P100 was installed at a Water Water Treatment Plant (WWTP) located Osberstown, Co. Kildare and was used for the validation the LOC phosphate analyser.

The commercial phosphate analyser was installed to monitor the final effluent leaving the WWTP. It worked alongside a nitrate (NO_2^-) and a nitrite (NO_3^-) monitor and was used to sample the phosphate concentration of the effluent every 15 minutes. The measured phosphate concentration was stored in a database on a nearby computer.

Figure 4.8 shows the set up of the analyser validation. In the effluent monitoring system, effluent flowed through a 100 µm filter to each of the monitors. The effluent then flowed back to a waste reservoir. The LOC analyser was connected to the same sample line as the other monitors using a modified sample port.

The sample port had a poly ethylene cover glued over it (see Figure 4.9). The cover had two 1/8" inner diameter barbed connectors for attaching tubing. 1/8" inner diameter silicone tubing was used to circulate effluent from the sample stream. The analyser could then draw sample water through the filter membrane.

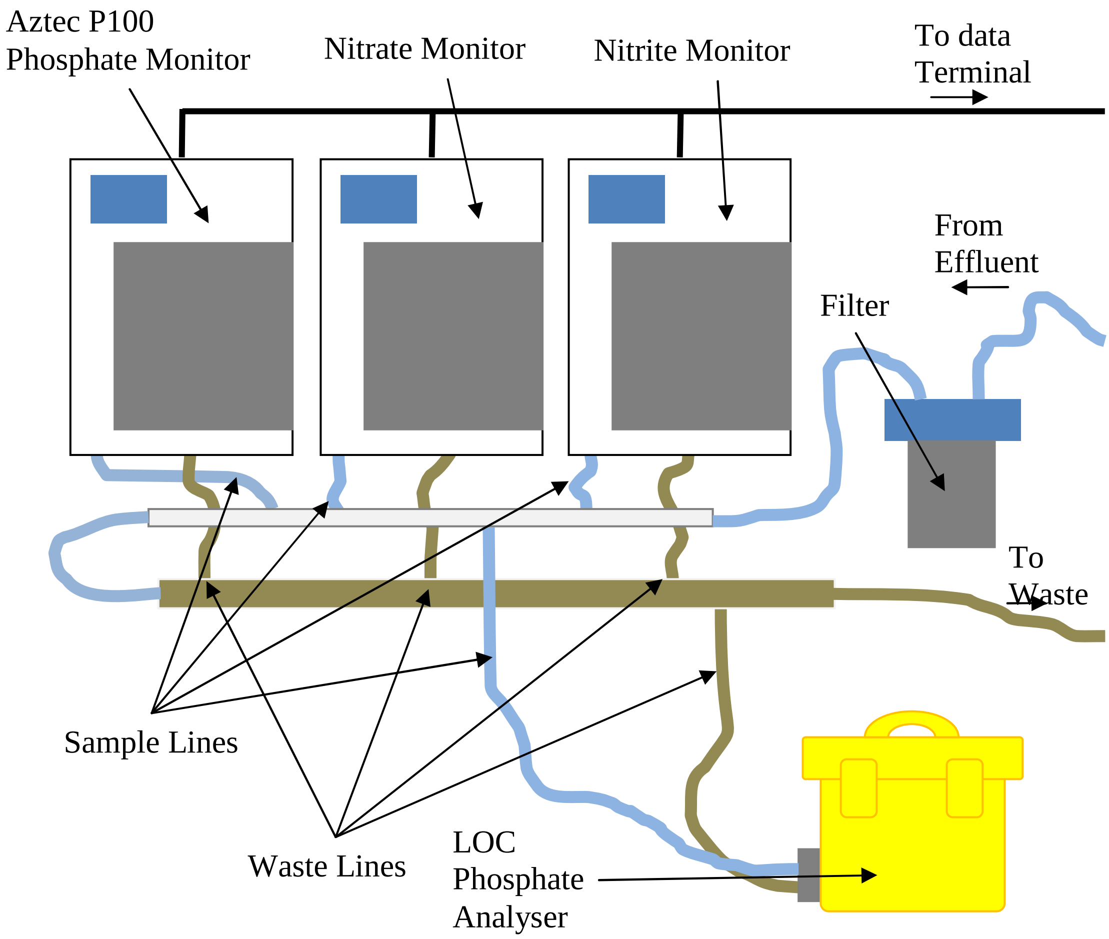


Figure 4.8: Setup of phosphate analyser at WWTP monitoring station

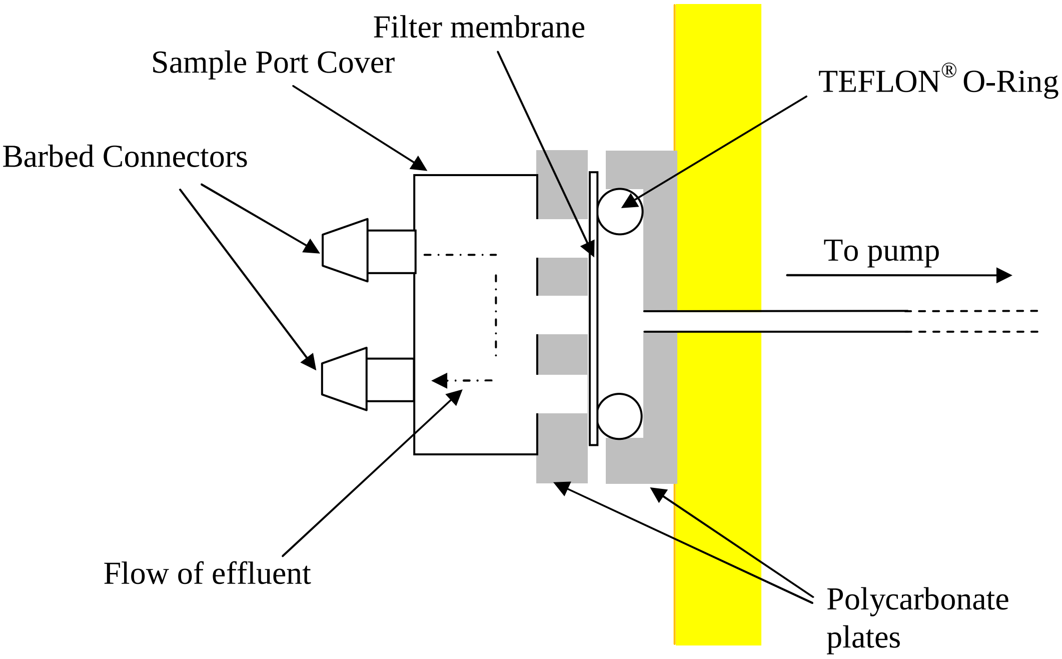


Figure 4.9: Schematic of modified sample port

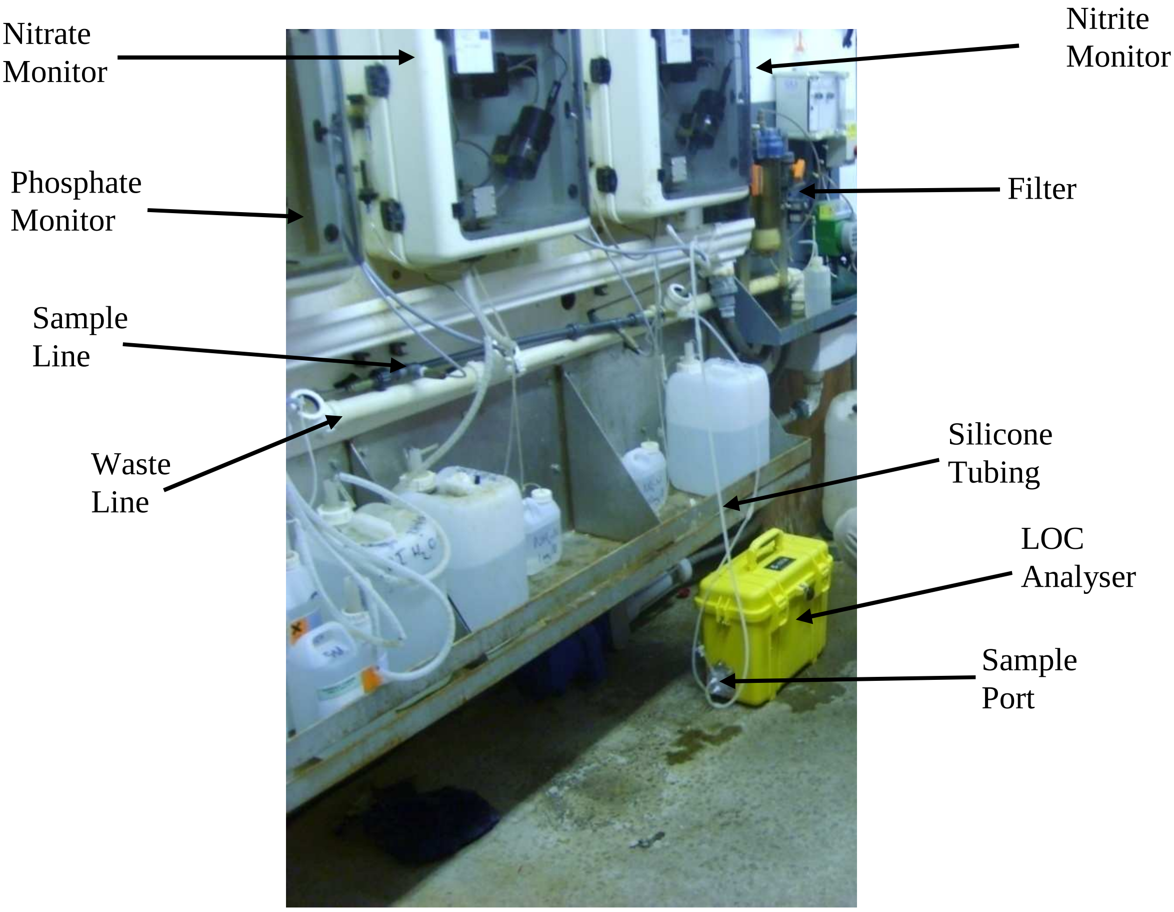


Figure 4.10: Phosphate analyser at WWTP

Figure 4.10 shows the LOC analyser at the WWTP. It was positioned below the three other monitors and the silicone tubing for circulating the effluent is visible in the picture. In the validation the LOC analyser was powered by connecting 12 V adaptor to mains electricity.

The validation ran for 38 days with a total of 910 assays being performed by the LOC analyser. The sample rate was set to every 60 minutes and the reaction time for each sample was 8 minutes. The GSM modem was used for sending the data collected to a mobile phone. The data was transmitted every 5 hours and was also stored on the flash memory chip. The AZTEC P100 was set to a sample rate of 15 minutes with the data being stored on the computer terminal. Only the samples taken by the AZTEC P100 at the same time as the LOC analyser were used in this validation (i.e. 3 out of the 4 measurements were discarded). After 450 samples the filter on the sample port became clogged with solids. The filter was changed and the validation was allowed to continue.

4.5 Comparison of results

The plot in Figure 4.11 shows the result of the validation. The raw data from the LOC analyser was normalised by using the blank and standard measurements as detailed earlier in this Chapter. The output of the Aztec P100 was the total phosphorus (P-PO_4^{3-}) in the sample so the output of the LOC analyser was converted to total phosphorus by multiplying the measured phosphate concentration by 0.333. A 5 point moving average was performed on both sets of data to show that both analysers follow the same trend in measured phosphate concentration. There are a number of places where the trends deviate from one another and these are shown in more detail in Figure 4.12, Figure 4.13 and Figure 4.14.

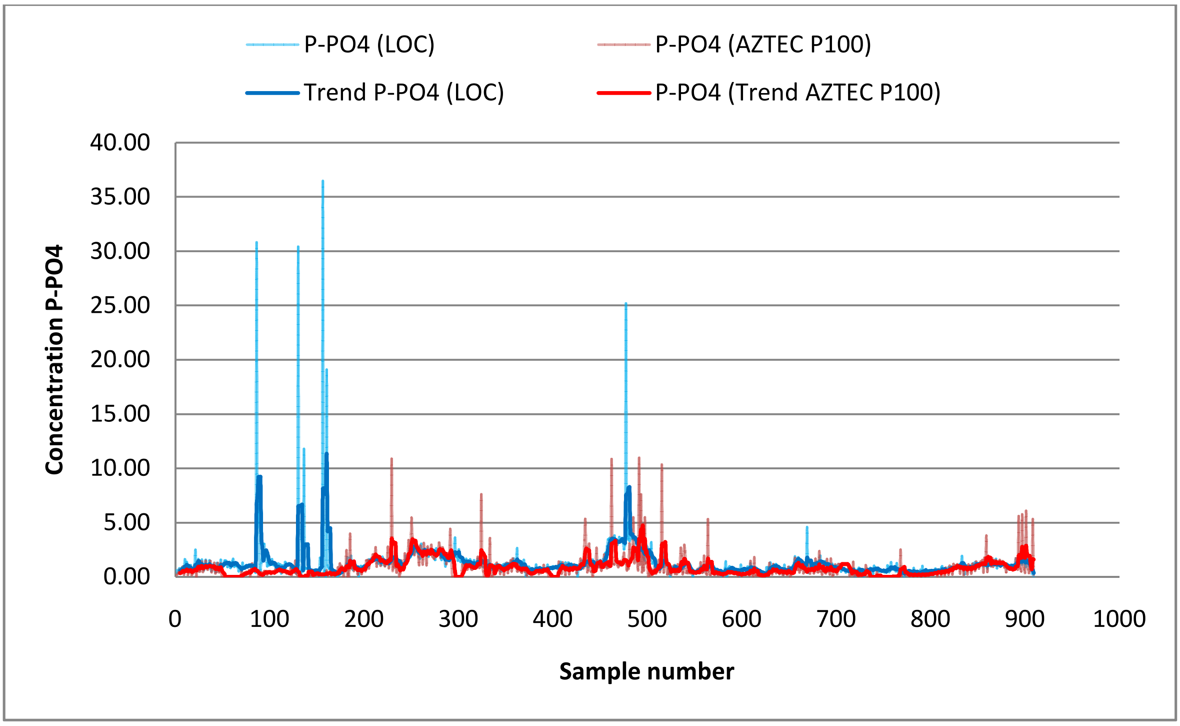


Figure 4.11: Comparison of output from LOC analyser and AZTEC P100 phosphate monitor

Figure 4.12 shows a number of spikes in the plot of the measured concentration from the LOC analyser that do not appear on the plot form the Aztec analyser. These were caused by bubbles in the optical cuvette in the LOC analyser. The bubbles scatter light from the emitter causing the absorbance measurement to be unusually high. It should be noted that bubbles only affected a small number of measurements after which the LOC analyser resumes correlation with the AZTEC P100. Methods for differentiating real high concentrations of phosphate and bubbles are discussed in Chapter 5.

There is a single point where the AZTEC P100 detected a higher concentration of phosphate compared to the LOC analyser. This could be due to the Aztec P100 being able to detect phosphates in particulate that is excluded by the 0.45 µm pore diameter filter on the LOC analyser. This same discrepancy was seen in the data from the phosphate analyser used by Cleary et al. (2008).

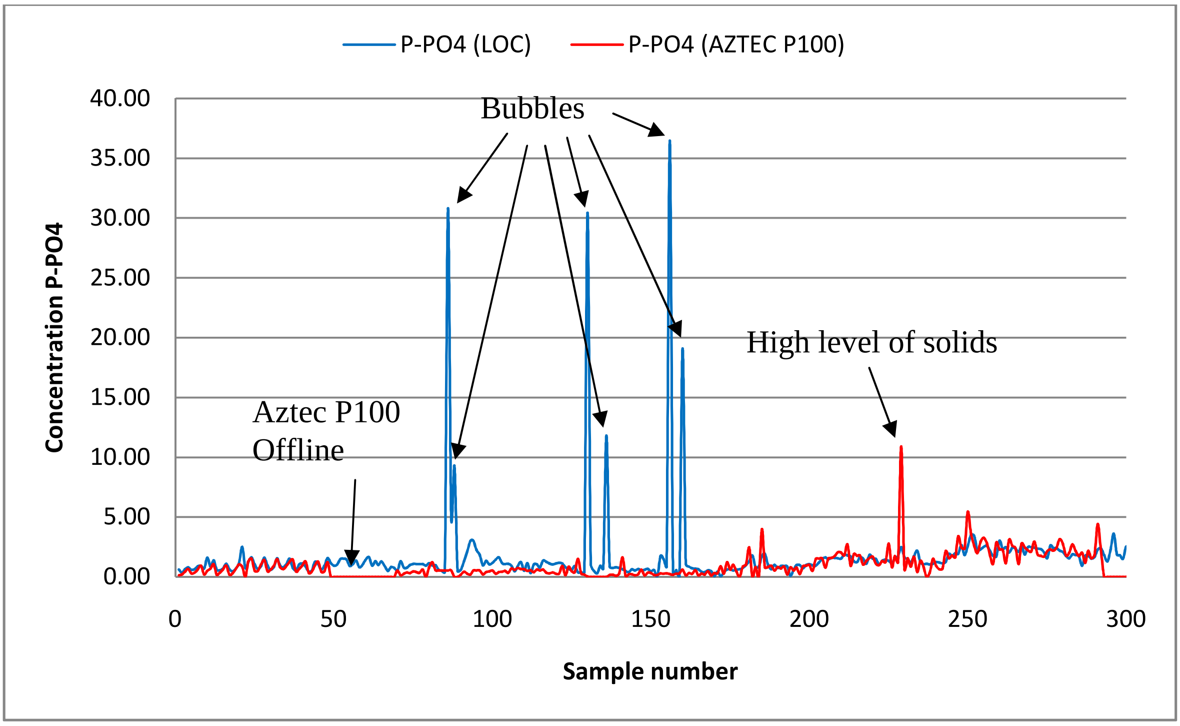


Figure 4.12: Beginning of validation showing good correlation and interference by bubbles

The effect of high amounts of particulate on both analysers is more evident in Figure 4.13. What is seen is a rapid deviation in measurements by the LOC analyser from the Aztec P100. This was caused by the modified sample port becoming clogged with high amounts of particulate. This particulate shows up in the measurements made by the Aztec P100 as number of spikes in measured concentration.

The sample port was removed, cleaned and the filter was replaced at about sample number 500. It can be seen that the measurements made by both analysers resumed correlation after the filter change and continue to do so until the end of the validation (see Figure 4.14).

There are a number times throughout the plot that the Aztec P100 was offline. This was due to the fact during the validation tests the Aztec P100 had to be serviced because of the failure of a component or chemicals were depleted (see Figure 4.14).

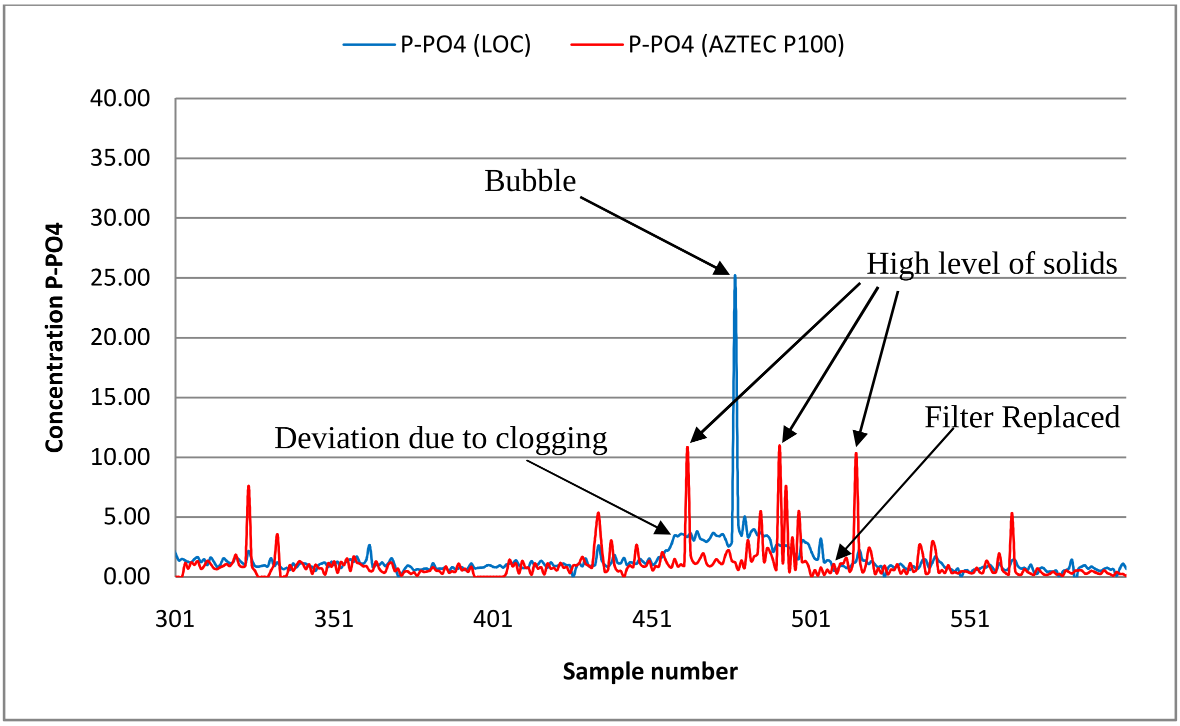


Figure 4.13: Detail of midway through validation showing effect of clogged filter

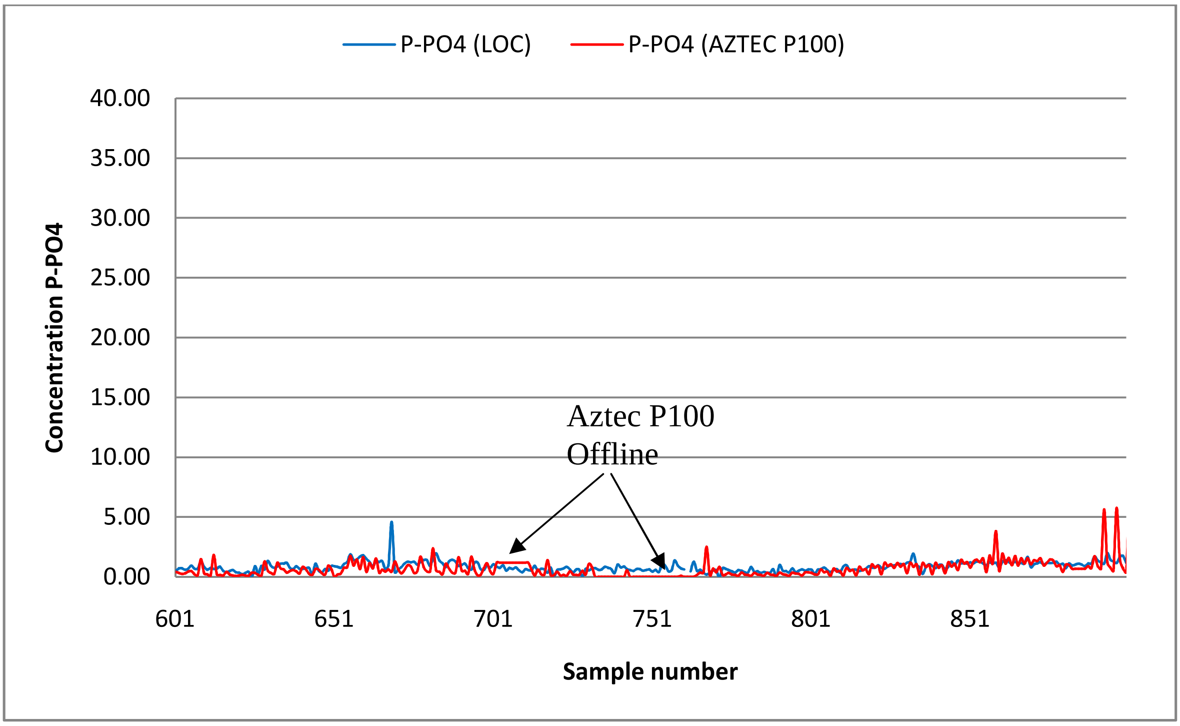


Figure 4.14: Detail of the final of validation showing good correlation until end

CHAPTER 5. DISCUSSION & CONCLUSIONS

The principles of fluid flow in microchannels were applied to design and construct a prototype phosphate analyser based on a Lab-on-a-Chip that was both compact and reliable. The analyser demonstrated improved detection performance and reliability compared to previous designs of LOC phosphate analysers. The improvements in performance were significant enough such that the LOC analyser can compete with commercially available phosphate analysers. It is clear that the LOC analyser and others like it are potentially a disruptive technology. That would displace and challenge current methods of phosphate monitoring.

The phosphate analyser designed in this work is based on a microfluidic Lab-on-a-Chip. It was shown that it is possible to make an analyser using solenoid metering pumps. The equivalent electronic circuit was an effective method to aid designing a fluidic system that accomplished all of the required actions to make a self calibrating analyser. The analyser could be built using mostly commercial parts. The custom built component of the analyser was the polymer Lab-on-a-Chip that is the implementation of the fluid system. It was shown that current fabrication techniques for LOCs are sophisticated enough to create a chip with the 3 dimensional layout required for the analyser to operate.

An analysis on the theoretical time between services of the analyser based on the battery life was performed. It was shown that from the selection of low power components that the life time was significantly long with the ability to perform 3,125 phosphate assays on one battery charge and also a standby time of 305 days. The LOC phosphate analyser developed by McGraw et al. which reported a battery life of 7 days at a sample rate of every 15 minute running off a 12 V, 7 Ah battery. The LOC analyser developed here would operate for 32 days using the same battery, a 460% increase in battery life for the same sample rate.

In the laboratory test of the analyser it was shown to be accurate over an extended period of time and over hundreds of assays. The accuracy was guaranteed by including a frequent 2-point calibration resulting in the analyser having no drift over time.

The LOC analyser also demonstrated an improved limit of detection over previous designs of LOC analyser due to the longer path length of the optical cuvette compared to the optical window used by McGraw et al. (2007). The 3 mm length compared to the 0.2 mm length resulted in a 300% decrease in the limit of detection. The improvement in the limit of detection is also due to the stability provided by the two-point calibration. The analyser compensates for drift so does not suffer from the increase in limit of detection reported by McGraw et al.

In comparison to the previous designs of LOC analysers there was a decrease in the linear range of the new LOC analyser. The non linearity is introduced by the absorbance measurement of the standard. When higher levels of phosphate are sample the measurement of the standard is affected by staining and this affects calculation of the concentration of phosphate in the sample. Previous designs of LOC analyser did not have a measurement of a standard to introduce this type of error.

Table 5.1 shows a comparison of commercial phosphate analysers with the LOC analyser developed here (LOC analyser II). It can be seen it has similar performance characteristics to the low end commercial analyser (Aztec P100). The linear range is less than expected and caused by the LOC becoming stained while measuring higher levels of phosphate.

Make	Developer	Detection Range (mg/L PO_4^{3-})	Comments
LOC Analyser II	**Slater**	**0.3 - 5**	**Portable, Automated**
LOC Analyser	McGraw/Cleary	0.9 - 60	Portable, Automated
Aztec P100	Severn-Trent	0.3 - 7.5	Automated
TresCon OP210	WTW	0.15 - 9	Automated
Pocket Colorimeter II	Hach-Lange	0.15 - 4.5	Portable

Table 5.1: Updated comparison of commercial and experimental phosphate analysers

During the validation of the developed LOC analyser against the commercially available Aztec P100, the LOC analyser operated for 38 days with only one fault. The sample port became clogged half way through and had to be cleaned and have the filter replaced. It was shown throughout the validation that the LOC analyser's output had excellent correlation with the Aztec P100. This is an improvement over the 21 day validation of the LOC analyser developed by Cleary et al. (2008). The new design of the LOC analyser was able to provide high quality data over this longer period of time.

Given that the LOC has similar performance characteristics to the Aztec P100 and has been proven to be reliable over a long period of time, it is conceivable that the LOC analyser could be used to substitute current commercial technology. In terms of cost the LOC analyser could be sold for between €5,000 and €10,000. This is a fraction of the €70,000 need to install a similar currently available commercial system.

LOC analyser also has the added benefit of being able to run off a battery, and so it can be deployed at remote locations. This would also allow the LOC analyser to replace the hand held colorimeter in some situations where high a level of precision is not required.

5.1 Recomendations

This work has shown that a µ-TAS can be constructed that can offer similar performance to commercially available microfluidic analysers but at reduced size, reagent consumption and power consumption. However this is not an optimised solution, there are improvements that can be made to a number of the analyser's key components.

The LOC analyser developed here can be improved further by miniaturisation of its individual components in particular the pumps. If the volume dispensed by the pumps was reduced this would reduce the amount of fluids used per assay decreasing the overall size of the analyser.

The optics of the LOC are not optimised for the absorbance measurements. The integration of fibre optics and waveguides would decrease the volume of sample needed and a more efficient optical set up could improve the limits of detection.

The linear range of the analyser is limited by staining of the LOC by the reagent. If cleaning of LOC was improved by using something more aggressive, then the linear range of the analyser could be increased.

It might also be possible to add more features to the phosphate analyser. As well as storing the data collected in the flash memory, samples could be stored in a microfluidic auto sampler such as the one developed by Boukellal et al. (2009) and the data could be verified later by High Performance Laser Chromatography (HPLC).

Microfluidic fuel cells could also be utilised in future designs (Kjeang, 2009). Fuels cells although under development have the potential for more efficient energy storage on a micro scale increasing the lifetime of future analysers.

REFERENCES

Amirouch, F., Zhou, Y., Johnson, T. (2009). Current micropump technologies and their biomedical applications. *Microsystems Technology*, 15, 647-666.

Aoki, N., Mae, K. (2006). Effects of channel geometry on mixing performance of micromixers using collision of fluid segments. *Chemical Engineering Journal*, 118, 189-197.

Auroux, P. -A., Iossifidis, D., Reyes, D., Manz, A. (2002). Micro total analysis systems. 2. Analytical standard operations and applications. *Analytical Chemistry*, 74(12), 2637-2652.

Balsev, S., Jorgensen, A. M., Bilenberg, B., Mogensen, K. B., Snakenborg, D., Geschke O., Kutter, J. P., Kristensen, A. (2006). Lab-on-a-Chip with optical transducers. *Lab on a Chip*, 6, 213-217.

Bartolo, D, Degre, G., Nghe, P., Struder, V. (2008). Microfluidic stickers. *Lab on a Chip*, 8, 274-279.

Bassous, E., Taub, H. H., Kuhn, L. (1977). Ink jet printing nozzle arrays etched in silicon. *Applied Physics Letters*, 31(2), 135-137.

Bogue, R. (2007). Optical chemical sensors for industrial applications. *Sensor Review*, 27(2), 86-90.

Bogue, R. (2008). Environmental sensing: strategies, technologies and applications. *Sensor Review*, 28(4), 275-282.

Boukellal, H., Selimovic, S., Jia, Y., Cristobal, G., Fraden, S. (2009). Simple robust storage of drops and fluids in a microfluidic device. *Lab on a Chip*, 9, 331-338.

Bowden, M., Sequeira M., Krog, J. P., Gravsen, P., Diamond, D. (2002). A prototype industrial sensing system for phosphorus based on micro system technology. *The Analyst*, 127, 1-4.

Bowden M., Sequiera, M., Krog, J. P., Gravsen, P., Diamond, D. (2002). Analysis of river water samples utilising a prototype industrial sensing system for phosphorus based on micro-system technology. *Journal of Environmental Monitoring*, 4, 767-771.

Bowden, M., Geschke, O., Kutter, J. P., Diamond, D. (2003). CO2 laser microfabrication of an integrated polymer microfluidic manifold for the determination of phosphorus. *Lab on a Chip*, 3, 221-223.

Bowden, M., Diamond, D. (2003). The determination of phosphorus in a microfluidic manifold demonstrating long-term reagent lifetime and chemical stability utilising a colorimetric method. *Sensors and Actuators B: Chemical*, 90, 170-174.

Braschler, T., Metref, L., Zvitov-Marabi, R., van Lintel, H., Demierre, Theytaz, J., Renaud, P. (2007). A simple pneumatic setup for driving microfluidics. *Lab on a Chip,* 7, 420-422.

Brask, A., Snakenborg, D., Kutter, J. P., Bruss, H. (2006). AC electroosmotic pump with bubble-free palladium electrodes and rectifying polymer membrane valves. *Lab on a Chip,* 6, 280-288.

Chen, C. F., Liu, J., Hromada, L. P., Tsao, C. W., Chang, C. C., DeVoe, D. L. (2009). High-pressure needle interface for thermoplastic microfluidics. *Lab on a Chip,* 9, 50-55.

Chung, A. J., Erickson, D. (2008). Engineering insect flight metabolic using immature stage implanted microfluidics. *Lab on a Chip*, 9, 669-676.

Cleary, J., Slater, C., McGraw, C. M., Diamond, D. (2008). An autonomous microfluidic sensor for phosphate: On-site analysis of treated wastewater. *IEEE Sensors Journal,* 8(5), 508-516.

Christensen, C. M. (2003), The innovator's dilemma : the revolutionary book that will change the way you do business, New York: HarperCollins

DeGrandpre, M. D., Baehr, M. M. (1999). Calibration-free optical chemical sensors. Analytical Chemistry, 71(6), 1152-1159.

diagnoSwiss (2009) Website. Available from: www.diagnoswiss.com 27-6-2009.

Diamond, D. (2004). Internet-scale sensing. *Analytical Chemistry*, 279-286.

Dopper, J., Clemens, M., Ehrfeld, W., Jung, S., Kamper, K. -P., Lehr, H. (1997). Micro gear pumps for dosing of viscous fluids. *Journal of Micromechanics and Microengineering*, 7, 230-232.

Einstein, A. (1905), Investigations on the theory of, the Brownian movement. *Annalen der Physik*, 17, 549.

European Council (2000). Establishing a framework for Community action in the field of water policy. Directive 2000/60/EC of the European Parliament and of the Council.

Geschke, O., Klank, H., Telleman, P. (2004). Microsystem engineering of lab-on-a-chip devices. *Wiley-VCH.*

Elwenspoek, M. Lammerink, T. S. J., Miyake, R., Fluitman, J. H. J. (1994). Towards integrated microliquid handling systems. *Journal of Micromechanics and Microengineering*, 4, 227-245.

Flachsbart, B. R., Wong. K., Innacone, J. M., Abante, E. N., Vlach, R. L., Rauchfuss, P. A., Bohn, P. W., Sweedler, J. V., Shannon, M. A. (2006). Design and fabrication of a multilayered polymer microfluidic chip with nanofluidic interconnects via adhesive contact printing. *Lab on a Chip*, 6, 667-674.

Golonka, L. J., Roguszczak, H., Zawada, T., Radojewski, J., Grabowska, I., Chudy, M., Dybko, A., Brzozka, Z., Stadnik, D. (2005). LTCC based microfluidic system with optical detection. Sensors and Actuators B: Chemical, 111-112, 396-402.

Gravsen, P., Branebjerg, J., Jensen, O. S. (1993). Microfluidics - a review. Journal of Micromechanics and Microengineering, 3, 168-182.

Greenway, G. M., Haswell, S. J., Petsul, P. H. (1999). Characterisation of a micro-total analytical system for the determination of nitrate with spectrophotometric detection. *Analytica Chimca Acta*, 387, 1-10

Hardt, S., Schonfeld, F. (2003). Laminar mixing in different interdigital micromixers: II. numerical simulations. *AIChE Journal*, 49(3), 578-564.

Haswell, S. J. (1997). Development and operating characteristics of micro flow injection analysis systems based on electroosmotic flow. *The Analyst*, 122, 1-10.

Hessel, V., Hardt, S., Lowe, H., Schonfeld, F. (2003). Laminar mixing in different interdigital micromixers: I. Experimental characterisation. *AIChE Journal*, 49(3), 566-577.

Hessel, V., Lowe, H., Schonfeld, F. (2004). Micromixers - a review on passive and active mixing principles. *Chemical Engineering Science*, 60, 2479-2501.

Hulme, S. E., Shevkoplyas, S. S., Whitesides, G. M. (2009). Incorporation of prefabricated screw, pneumatic, and solenoid, valves into microfluidic devices. *Lab on a Chip*, 9, 79-86

Jensen, M. J., Goranović, G., Bruus, H. (2004). The clogging pressure of bubbles in hydrophilic microchannel contractions. *Journal of Micromechanics and Microengineering*, 14, 876-883.

Jordan, P., Arnscheidt, J., McGrogan, H., McCormick, S. (2005). High-resolution phosphorus transfers at the catchment scale: the hidden importance of non-storm transfers. *Hydrology and Earth System Sciences*, 9(6), 68-691.

Jordan, P., Arnscheidt, J., McGrogan, H., McCormick, S. (2007). Characterising phosphorus transfers in rural catchments using a continuous bank-side analyser. *Hydrology and Earth System Sciences*, 11(1), 372-381.

Kamholz, A. E. (2004). Proliferation of microfluidics in literature and intellectual property. *Lab on a Chip,* 4, 16-20.

Kjeang, E., Djilali, N., Sinton, D. (2009). Microfluidic fuel cells: a review. *Journal of Power Sources,* 186, 353-369.

Knight, J. B., Vishwanath, A., Brody, J. P., Austin, R. H. (1998). Hydrodynamic focusing on a silicon chip: Mixing nanoliters in microseconds. *Physical Review Letters*, 80(17), 3863-3866.

Kockmann, N., Kiefer, T., Engler, M., Woias, P. (2006). Convertive mixing and chemical reactions in microchannels with high flow rates. *Sensor Actuators B: Chemical*, 117, 495-508.

Lammertyn, J., Verboven, P., Veraverbeke, E. A., Vermeir, S., Irudayaraj, J., Nicolai, B. M. (2006). Analysis of fluid flow and reaction kinetics in a flow injection analysis biosensor. *Sensor and Actuators B: Chemical*, 114, 728-736.

Laser, D. J., Santiago, J. G. (2003). A review of micropumps. *Journal of Micromechanics and Microengineering*, 14, 35-64.

Lee, G. -B., Hwei, B. -H., Huang, G. -R. (2001). Micromachined prefocused M × N flow switches for continuous multi-sample injection. *Journal of Micromechanics and Microengineering*, 11, 654-661.

Liu, R. H., Yang, J., Lenigk, R., Bonanno, J., Grodzinski, P. (2004). Self-contained, fully integrated biochip for sample preparation, polymerase chain reaction amplification, and DNA microarray detection. *Analytical Chemistry*, 76, 1824-1831.

Manz, A., Graber, N., Widmer, H. M. (1990). Miniaturized total chemical analysis systems: a novel concept for chemical sensing. Sensors and Actuators B: Chemical, 1, 244-248.

Mariella, R. (2008). Sample preparation: the weak link in microfluidics-based biodetection. *Biomedical Microdevices*, 10, 777-784.

Martz, T. R., Carr, J. J., French, C. R., DeGrandpre, M. D. (2003). A submersible autonomous sensor for spectrophotometric pH measurements of natural waters. *Analytical Chemistry*, 75(8), 1844-1850.

Meager, R. J., Hatch, A. V., Renzi, R. F., Singh, A. K. (2008). An integrated microfluidic platform for sensitive and rapid detection of biological toxins. *Lab on a Chip*, 8, 2046-2053.

McGraw, C. M., Stitzel, S. E., Cleary, J., Slater, C., Diamond, D. (2007). Autonomous microfluidic system for phosphate detection. *Talanta*, 71, 1180-1185.

Morimoto, T., Konishi, S. (2008). Layer-to-layer parallel fluidic transportation system by addressable fluidic gate arrays. *Lab on chip*, 8, 1552-1556.

Nagasawa, H., Aoki, N., Mae, K. (2005). Design of a new micromixer for instant mixing based on the collision of micro segments. Chemical Engineering Technology, 28(3), 324-330.

Nguyen, N. -T., Huang, X., Chuan, T. K. (2002). MEMS-micropumps: a review. *Transactions of the ASME*, 124, 384-392.

Oh, K. W., Ahn, C. H. (2006). A review of microvalves. Journal of Micromechanics and Microengineering, 16, 13-19.

Pennathur S. (2008). Flow control in microfluidics: are the workhorse flows adequate? *Lab on a Chip*, 8, 383-387.

Pennathur, S., Meinhart, . D., Soh, H. T. (2008). How to exploit the features of microfluidics technology. *Lab on a Chip*, 8, 20-22.

Petsul, P. H., Greenway, G. M., Haswell, S. J. (2001). The development of an on-chip micro-flow injection analysis of nitrate with a cadmium redactor. *Analytica Chimica Acta*, 428, 155-161.

Piruska, A., Nikcevic, I., Lee, S. H., Ahn, C., Heineman, W. R., Limbach, P. A., Seliskar, C. J. (2005). The autofluorescence of plastic materials and chips measured under laser irradiation. *Lab on a Chip*, 5, 1348-1354.

Reyes, D. R., Iossifidis, D., Auroux, P. -A., Manz, A. (2002). Micro total analysis system. 1. introduction, theory, and technology. *Analytical Chemistry*, 74(12), 2623-2636.

Rhee, M., Burns, M. A. (2008). Microfluidic assembly blocks. *Lab on a Chip*, 8, 1365-1373.

Scarmagnani, S., Walsh, Z., Slater, C., Alhashimy, N., Paull, B., Macka, M., Diamond, D. (2008). Polystyrene bead-based system for optical sensing using spiropyran photoswitches. *Journal of Materials Chemistry*, 18, 5063-5071.

Seidel, M. P., DeGrandpre, M. D., Dickson, A. G. (2008). A sensor for in situ indicator-based measurements of seawater pH. *Marine Chemistry*, 109, 18-28.

Sequeira, M., Diamond, D. (2002). Progress in the realisation of an autonomous environmental monitoring device for ammonia. *Trends in analytical chemistry*, 21(12), 816-827.

Sequeira, M., Bowden, M., Minogue, E., Diamond, D. (2002). Towards autonomous environmental monitoring systems. *Talanta*, 56, 355-363.

Shen, M., Yamahata, C., Gijs, M. A. M. (2008). A high performance compact electromagnetic actuator for a PMMA ball-valve micropump. *Journal of Micromechanics and Microengineering*, 18, 1-9.

Skelly, A. M., Voldman, J. (2008). An active bubble trap and debubbler for microfluidic systems. *Lab on a Chip*, 8, 1733-1737.

Snakenborg, D., Perozziello, G., Geschke, O., Kutter, J. P. (2006) A fast and reliable way to establish fluidic connections to planar microchips. *Journal of Micromechanics and Microengineering*, 17, 98-103.

Sugiura, S., Szilagyi, A., Sumaru, K., Hattori, K., Takagi, T., Filipcsei, G., Zrinyi, M., Kanamori, T. (2009). On-demand microfluidic control by micropatterned light irradiation of a photoresponsive hydrogel sheet. *Lab on a Chip*, 9, 196-198.

Sun, K., Wang, Z., Jiang, X. (2008). Modular microfluidic for gradient generation. *Lab on a Chip*, 8, 1536-1543.

Takabayashi, Y., Fujino, T., Korenaga, T. (2008). Direct observation of dispersion and mixing processes in microfluidic systems. *Analytical Sciences*, 24, 1481-1485.

Tamanaha, C. R., Whitman, L. J., Cotton, R. J. (2002). Hybrid macro-micro fluidics system for a chip-based biosensor. *Journal of Micromechanics and Microengineering*, 12, 7-17.

Tedesco, M., Bohlen, W. F., Howard-Strobel, M. M., Cohen, D. R., Tebeau, P. A. (2003). The mysound project: building an estuary-wide monitoring network for long island sound, U.S.A. *Enviromental Monitoring and Assessment*, 81, 35-42.

Thorsen, T., Maerkl, S. J., Quake, S. R. (2002). Microfluidic large-scale integration. *Science*, 298, 580-584.

Tsao, C. W., Hromada, L., Liu, J., Kumar, P., DeVoe, D. L. (2007). Low temperature bonding of PMMA and COC microfluidic substrates using UV/ozone surface treatment. *Lab on a Chip*, 7, 499-505.

Weigl, B., Domingo, G., LaBarre, P., Gerlach, J. (2008). Towards non- and minimally instrumented, microfluidics-based diagnostic devices. *Lab on a Chip*, 8, 1999-2014.

Yamada, M., Seki, M. (2005). Hydrodynamic filtration for on-chip particle concentration and classification utilizing microfluidics. *Lab on a Chip*, 5, 1233-1239.

Yamada, M., Hirano, T., Yasuda, M., Seki, M. (2006). A microfluidic flow distributor generating stepwise concentrations for high-throughput biochemical processing. *Lab on a Chip*, 6, 179-184.

Yi, L., Xiaodong, W., Fan, Y. (2008). Microfluidic chip made of COP (cyclo-olefin polymer) and comparion to PMMA (polymethylmethacrylate) microfluidic chip. *Journal of materials processing technology*, 208, 63-69.

PUBLICATIONS

Slater, C., Cleary, J., Lau, K. T., Snakenborg, D., Corcoran, B., Kutter, J. P., Diamond, D. (2009). Validation of a fully autonomous phosphate analyser based on a microfluidic lab-on-a-chip. *Journal of Water Science and Technology*, (Submitted).

Cleary, J., Slater, C., McGraw, C. M. & Diamond, D. (2008) An autonomous microfluidic sensor for phosphate: on-site analysis of treated wastewater. *IEEE Sensors Journal*, 8(5), pp. 508-515.

McGraw, C. M., Stitzel, S., Cleary, J., Slater, C. & Diamond, D. (2007) Autonomous microfluidic system for phosphate detection. *Talanta*, 71(3), pp. 1180-1185.

Slater, C., Cleary, J., McGraw, C. M., Yerazunis, W. S., Lau, K. -T. & Diamond, D. (2007) Autonomous field-deployable device for the measurement of phosphate in natural water. *The proceedings of the Conference on Advanced Environmental, Chemical, and Biological Sensing Technologies V*, Boston USA, pp. L7550-L7550.

Slater, C., Corcoran, B. & Diamond, D. (2007) Fabrication of a multi-layer microfluidic chip from PMMA. *Proceedings of the 24th International Manufacturing Conference,* Waterford IRELAND.

APPENDIX

A. Datasheets:

1. Severn Trent Aztec P100
2. WTW TresCon OP210
3. Hach-Lange POCKET Colorimeter II
4. 120SP Solenoid Metering Pump
5. Honeywell, PK80083

B. Technical Drawings:

1. CHIP_BOTTOM
2. CHIP_MIDDLE
3. CHIP_TOP
4. JIG_BOTTOM
5. JIG_TOP
6. CHIP_COVER
7. CHIP_COVER_MOUNT
8. CHIP_MOUNT
9. CHIP_ENCLOSURE
10. PUMP_MOUNT_TOP
11. PUMP_MOUNT
12. TOP_PLATE_LEFT
13. TOP_PLATER_RIGHT

C. Electronics:

1. Photodiode Circuit Schematic
2. Photodiode Circuit PCB Layout
3. Nichia NSHU590
4. Roithner Laser Technik EPD-440-0
5. Texas Instruments TLV277x
6. Texas Instruments LP2985LV
7. Texas Instruments TPS7695
8. National Semiconductor LM335
9. Control Board Schematic
10. Control Board PCB Layout
11. Texas Instruments MSP430F449

12. International Rectifier IRLML2502

13. ST Microelectronics M25P20

14. Texas Instruments MAX3232

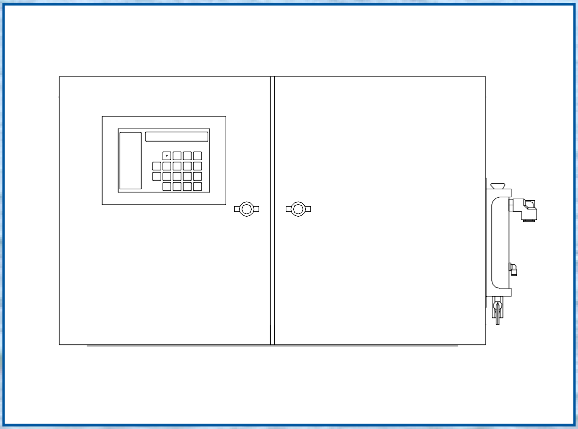
D. Software

1. adc12.
2. batterymonitor
3. command
4. gsm
5. GSMPo-II
6. phosphate
7. photodiode
8. pindefs
9. pumps
10. spi_flash
11. system
12. temp_sensor
13. uart
14. utility

AZTEC®
Phosphate Monitor
Series P100

SEVERN
TRENT
SERVICES

CAPITAL CONTROLS



- Continuous on-line monitoring
- Two-point auto-calibration
- Self-diagnostics
- Self-cleaning optics
- Multiple sample streams
- Microprocessor controlled
- Data logging
- Low service requirement
- Programmable sample frequency

The Capital Controls AZTEC® Phosphate Monitor, Model P100 is a reliable, continuous on-line process instrument. The microprocessor-based technology makes the operation easy and user-friendly.

The measurement is based on the recognized and approved molybdenum blue method. An adaptation of this method is also described in “Standard Methods”, 18th Edition.

Accuracy and reproducibility are obtained through a programmable automatic calibration feature. This two-point calibration, based on deionized water and a standard solution, represents the only tuning needed for continuous operation.

The sample head combines a precision-engineered pump with the optical measuring cell. The optical cell is self-cleaning further ensuring an accurate and reproducible system.

Sample flow and reagents are monitored by level detectors and the monitor is designed to operate for a minimum of 20 days before reagent replacement.

The P100 is designed to NEMA 4 standards and uses corrosion-resistant materials.

The microprocessor provides between one and six weeks of data logging. Standard outputs include RS232, 4-20 mAdc, parallel printer port, and alarm relays.

Applications

- **Drinking Water:** Raw and finished water monitoring and control
- **Surface Water:** Monitor for treatment against algae blooms
- **Wastewater:** Effluent compliance and process control
- **Industrial Wastewater:** Effluent compliance
- **Cooling Towers:** Chemical package control
- **Boiler Water:** Process control

Design Features

- **Automatic Calibration**: A two-point calibration using deionized water and a phosphate standard ensures highly accurate and reproducible results
- **Communications:** Serial (RS232) and parallel (centronics printer port) and 4-20 mAdc outputs are standard
- **Self-diagnostics:** The monitor will indicate system faults and shut down in the event of a power or sample flow failure. The unit will automatically restart once these are restored.
- **Microprocessor controlled:** Installation and operation is user-friendly with control variables such as date, time, alarm limits, calibration interval and print mode that can be entered via the keypad
- Multi-stream: In addition to the standard single stream unit, an optional triple stream unit is available
- **Data logging:** The microprocessor has a data logging capability with storage in excess of one week of data
- **Sample frequency:** The number of samples per hour is fully programmable. This reduces maintenance frequency and reagent consumption.

Principle of Operation

The colorimetric method is based upon the reaction of molybdenum-blue in an acidic sample. This optical method passes light through a 20 mm sample cell at a wavelength of 690 nm.

A discrete sample of water is collected by the pump at user-programmable intervals. A sample blank is measured to compensate for color or turbidity of the sample and then the sample is transferred to a reaction chamber where acid is added. A conditioning reagent and color reagent allow a color to develop in direct proportion to the concentration of orthophosphate in the sample. This solution is drawn into the combined pump/optical cell and the transmission of light is measured using a photo-diode detector and a 690 nm optical filter.

The output from the detector is converted by a microprocessor into mg/l P or mg/l PO_4.

In addition, a pre-programmed, two-point automatic calibration ensures the accuracy of this analyzer. The piston pump provides accurate reproducible and a continually cleaned sample cell. (Figure 1)

Technical Data

Series P100

Quality Standards: ISO 9001 Certified
Measuring Ranges:
Low: 0.3-7.5 mg/l phosphate or 0.1-2.5 mg/l phosphorous
High: 0.9-22.5 mg/l phosphate or 0.3-7.5 mg/l phosphorous.
Accuracy: ±0.01 to ±0.3 mg/l P
Resolution: 0.01 mg/l
Repeatability: <2% of reading
Sample Flow: 200 to 500 ml/min.
Sample Temperature Range: 1-35°C (33-96°F)
Operating Temperature Range: 0-35°C (32-96°F)
Power: 110 Vac, ±6%, 60 Hz, 120/240 Vac, ±6%, 50/60 Hz. Supply to be stable and generally free of voltage dips/surges, excessive switching spikes and transient noise. 5 Amp fuse.
Power Consumption: 240 watts
Outputs: RS232, parallel (centronics), isolated 4-20 mAdc into 1000 ohms maximum. Each value is held until the reading is updated
Alarms: 4 dry contacts rated 10 amps @ 240 Vac maximum
Chemical Reagents:
Acid reagent
Conditioning reagent
Indicator reagent ANSA
Deionized water
Standard solution
Sample Frequency: 1 to 6 measurements per hour
Cabinet: NEMA 4 Three-section industrial, wall cabinet, mild steel construction with phosphate etch prime and epoxy powder texture finish.
Dimensions: 21 1/2” (545 mm) H x 35 3/8” (899 mm) W x 11 1/4” (285 mm) D
Shipping Weight: 125 lb. (60 kgs)

Model Information Code

Model P100 __ __

Voltage
1 - 120 Vac, 50/60 Hz, single phase

2 - 240 Vac, 50/60 Hz, single phase

Sample stream
0 - Triple sample stream
1 - Single sample stream

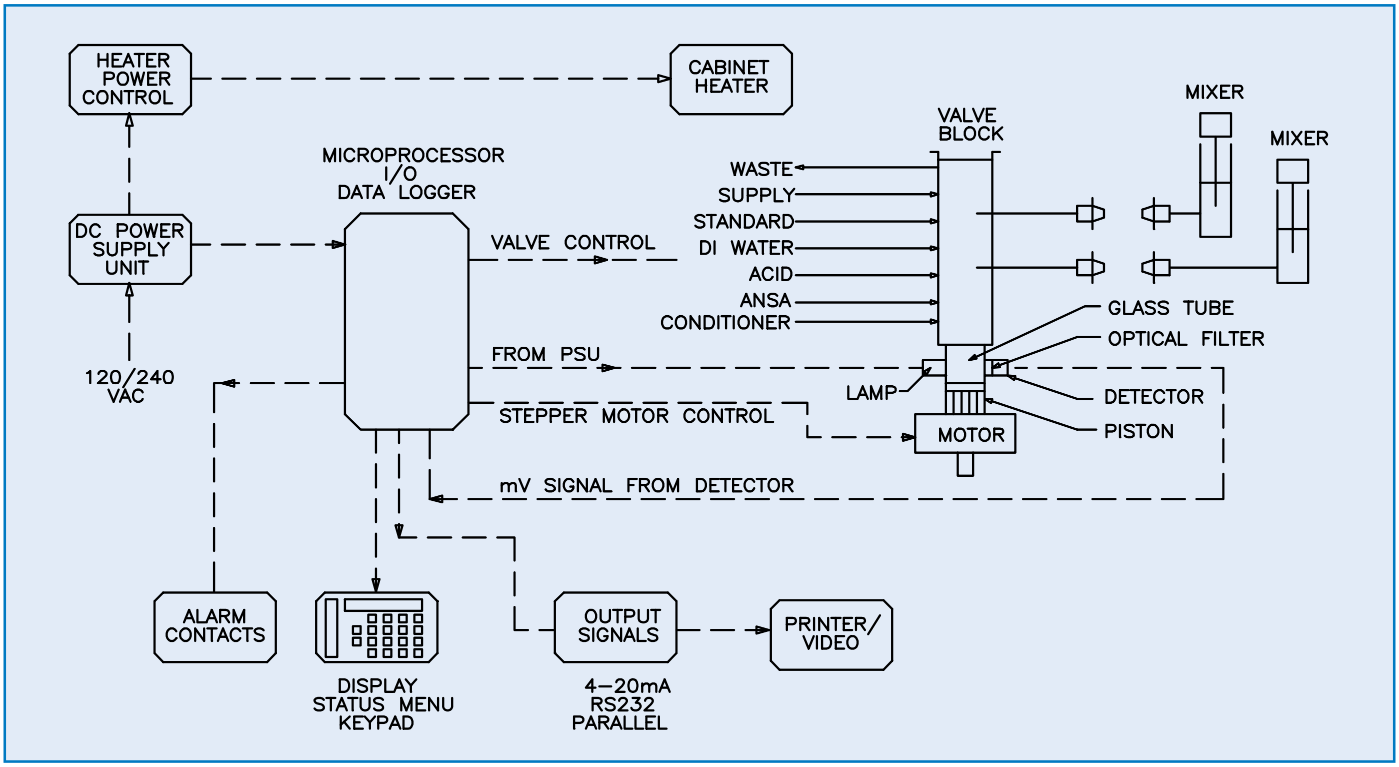


Figure 1

Warranty and Capability

Capital Controls offers a one (1) year limited warranty on the P100 Phosphate Monitor.

Capital Controls is ISO 9001 certified to provide quality and precision materials. Disinfection technologies, water quality monitors and instrumentation for water and wastewater are areas of specialization. Over 35 years of industrial and municipal application experience in the water and wastewater industries is incorporated into the equipment design to provide high quality comprehensive solutions for the global market.

Brief Specification

The phosphate monitor shall provide on-line, continuous analysis of a water sample using the molybdenum-blue method. The method shall incorporate a blank measurement on the untreated sample. The range of the monitor shall be selectable from a low range of 0.3 to 7.5 mg/l PO_4 to a high range of 0.4 to 22.5 mg/l PO_4.

The cabinet shall be constructed of mild steel and be suitably rugged for long term plant operation, water-resistant to NEMA 4 standard. The sensor and electronics shall be mounted within separate compartments to ensure a safe working environment.

Sampling frequencies shall be programmable between 1 and 6 times per hour. Automatic two-point calibrations shall be programmable between 1 and 4 times per day. The unit shall be configurable to a 3-stream unit within the standard enclosure.

The unit shall feature a programmable isolated 4-20 mAdc output, RS232c port, parallel printer port and alarm relay contacts.

The unit shall operate on standard clean power supply within 6% of specification.

Design improvements may be made without notice.

Represented by:

SEVERN
TRENT
SERVICES

CAPITAL CONTROLS

Severn Trent Water Purification, Inc.
3000 Advance Lane Colmar, PA 18915
Tel: 215-997-4000 • Fax: 215-997-4062
Web: www.severntrentservices.com
E-mail: marketing@severntrentservices.com

UNITED KINGDOM • UNITED STATES • HONG KONG
INDIA • ITALY • MALAYSIA

Phosphate

Phosphate

Phosphorus compounds – in particular ortho-phosphate PO_4^{3-} – are considered to be the limiting nutrients in most stagnant and flowing waters. An increase in their concentration caused by higher input (wastewater, avulsion etc.) results directly in increasing eutrophication of the water with known effects such as increased growth of algae, oxygen depletion as far as anoxia in the deeper regions, etc.

Phosphorus Compounds in Water

Phosphorus occurs in 3 compounds in natural waters:

- inorganic, dissolved **ortho-phosphate**
- dissolved organic phosphorus compounds
- particulate phosphorus (bound in biomass or attached to particles),

which add up to the **total of phosphorus content P_{Total}**, an important parameter in monitoring wastewater treatment plant effluents.

Measuring Methods and Digestion

There are two methods available for determining phosphate or phosphorus concentrations:

- **Molybdenum blue method**
- **Vanadate/molybdate method (yellow method)**

Both techniques are based on the measurement of **ortho-phosphate**. **Digestion** of both dissolved organic as well as particulate phosphorus compounds is therefore mandatory for determining the **total P** content. In addition, an unfiltered sample must be acquired in order to include all solid matters in the digestion process. Digestion is usually performed by heating the sample with peroxodisulfate and sulfuric acid.

Elimination of Phosphates in Wastewater

To meet the required limits of P concentration in the effluent, the modern wastewater treatment facility has two methods available:

- **Biological elimination of phosphates "Bio-P":**
  incorporation of phosphate in microbial biomass (usually in combination with a preliminary anaerobic stage to stimulate luxury consumption of phosphate and intracellular storage as polyphosphate)
- **Chemical-physical elimination of phosphates:**
  Chemical precipitation of ortho-phosphates using metallic salts (usually Fe^{3+} or Al^{3+}). The use of ortho-phosphate analyzers for effective control and regulation of precipitations results in considerable savings.

Regulation according to P Concentration

With a continuous monitor PO_4 analyzer, the operator of water treatment plants can realize significant cost savings.
(*cf. Application Report PO4 1609 2003 01e*)

Molybdenum blue method

In an acidic medium, ortho-phosphates bond with ammonium molybdate to form molybdenic phosphoric acid. With the aid of a reducing agent this forms phosphorus molybdenum blue compound. Photometrical measurement of dye intensity can be performed at 880 nm.

Vanadate/molybdate method (yellow method)

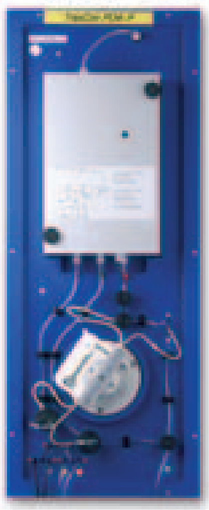
In acids, ortho-phosphate ions react with ammonium molybdate and ammonium vanadate to form yellow ammonium phosphoric vanadomolybdate. This can be photometrically analyzed at 380 nm.

TresCon®

OP 210

TresCon® OP 210

Phosphate Analyzer Module



On-line orthophosphate measurement

- Control or feedback control of chemical phosphate precipitation, e.g. precipitating agent addition with simultaneous precipitation
- Monitoring biological phosphate elimination
- Measuring the phosphate pollution in natural waters
- Monitoring the phosphate concentration in the drinking water

Measuring Principle

The PO_4 module uses the vanadate/molybdate method (yellow method) for determining the orthophosphate content. A reagent reacts with phosphate in the sample to color the sample solution yellow. The intensity of this color is recorded photometrically and evaluated as a measure of the phosphate content.

- **Yellow method**
- **Continuous background compensation**
- **Continuous/Discontinuous operation selectable**
- **Can be used in weakly polluted water without sample preparation**

IP 54 CE 2 Years Warranty

Technical Data OP 210

Resolution (Display)	Measuring range 1: 0.01 mg/l or µmol/l Measuring range 2: 0.1 mg/l or µmol/l Measuring range 3: 0.1 mg/l or µmol/l
Accuracy	±2% of the measured value ±0.01 mg/l PO_4-P (Measuring range 1) ±2% of the measured value ±0.1 mg/l PO_4-P (Measuring range 2 and 3)
Coefficient of variation for method	2% (for all measuring ranges)
Response time	< 4 min to measured value (after alteration in concentration at module input)
Measuring interval	Quasi-continuous measurement; 5, 10, 15, 20, 25 or 30 min settings
Calibration	Automatic 2-point calibration (time and interval selectable)
Background correction	Continuous background compensation based on new WTW algorithm
Sample input	Approx. 0.06 l/h, solid content <50 mg/l (e.g. sewage treatment plant effluent)
Consumption	Reagent, 10 l: 60/155/310/465 days with cont. /10/20/30 min measuring intervals Standard B 1.5 l: 90 days with 24 h calibration interval Cleaning solution, 1.5 l: 45 days with 24 h cleaning interval
Maintenance interval	Every 6 months

Measuring Range 1		
	mg/l	µmol/l
PO_4-P	0.05 - 3.00	1.5 - 100
PO_4	0.15 - 9.00	1.5 - 100

Measuring Range 2		
	mg/l	µmol/l
PO_4-P	0.1 - 10.0	3 - 320
PO_4	0.3 - 30.0	3 - 320

Measuring Range 3		
	mg/l	µmol/l
PO_4-P	0.1 - 25.0	3 - 800
PO_4	0.3 - 80.0	3 - 800

Ordering Information OP 210

Separate TresCon® analyzer module for Orthophosphate for extension of an existing TresCon® system (requires 1 measuring place)		**Order No.**
OP 210/ MB 1	Module for Orthophosphate: Measuring range 1	820 004
OP 210/ MB 2	Module for Orthophosphate: Measuring range 2	820 005
OP 210/ MB 3	Module for Orthophosphate: Measuring range 3	820 006
TresCon®-basic instrument with analysis module OP 210 for ortho-phosphate (wall mounting, space for 2 further modules)		**Order No.**
TresCon® P 211/MB1	Orthophosphate, Measuring range 1	8A-40030
TresCon® P 211/MB2	Orthophosphate, Measuring range 2	8A-50030
TresCon® P 211/MB3	Orthophosphate, Measuring range 3	8A-60030
TresCon® Uno single parameter system ortho-phosphate with analysis module OP 210		**Order No.**
TCU/P211-MB1	TresCon® Uno for Orthophosphate: Measuring range 1	820 104
TCU/P211-MB2	TresCon® Uno for Orthophosphate: Measuring range 2	820 105
TCU/P211-MB3	TresCon® Uno for Orthophosphate: Measuring range 3	820 106
	Accessories and Consumables see brochure "Product Details"	

For information visit **www.WTW.com** for a customer care center near you or inside US: call **WTW 800 645 5999**.

Parameter section
Dissolved Oxygen
pH/ORP
Conductivity
Turbidity/ Suspended Solids
Nitrogen
Phosphate
Carbon: COD/TOC/DOC/ BOD/SAC

TresCon®

OP 510

Total phosphorus analyzer module

- On-line analysis of total phosphorus for wastewater treatment plant effluent
- Rapid analysis at 10 min intervals
- 2-point calibration – high degree of accuracy
- Automatic Cleaning
- Automatic Monitoring
- Blue method

On-line P_{Total} measurement

- Monitoring the effluent from wastewater treatment plant for P_{Total}
- Monitoring phosphorus pollution in natural waters

	Measuring Range*	
	mg/l	µmol/l
P_{Total}	0.01 ... 3.00/ 6.00*	0.3 ... 100/ 200*

* by continuous sample dilution in a 1:1 ratio

IP 54 CE 2 Years Warranty

TresCon® OP 510

Measuring Principle

The P_{Total} module consists of two units: in the first unit (digestion unit) the sample undergoes a chemical-thermal digestion; in the second unit the total phosphorus content is determined.

During the **digestion** all the phosphorus compounds contained in the sample are converted to orthophosphate; this can be determined photometrically. The phosphorus compounds are oxidized by peroxodisulfate under acidic conditions. This process is accelerated by overpressure and an increased temperature so that very short digestion times are achieved.

The subsequent **analysis** is by the molybdenum blue method. The sample is mixed with a molybdate reagent which reacts with phosphate via an intermediate chemical step to form a blue coloration. The intensity of this coloration is a measure of the original concentration of the phosphate ions. It is measured photometrically and evaluated.

Technical Data

Resolution (Display)	Range: 0.01 ... 3.00 mg/l : 0.01 mg/l 0.30 ... 100 µmol/l : 0.1 µmol/l
Accuracy	±3% of the measured value ±0.05 mg/l P_{Total}
Measuring principle	Photometric reference beam method after digestion
Measuring method	Molybdenum blue method
Coefficient of variation for method	1.5%
Measuring interval	10, 15, 20, 25, 30 or 60 min can be set (DIN EN measurement with 30 min digestion at approx. 248 °F/120 °C)
Calibration	Fully automatic 2-point calibration
Consumption	Reagents A, B, C, D: 10/15/20/30/60 days with 10/15/20/30/60 min measuring interval Standard, 1.5 l: 70 days with 24 h calibration interval Cleaning solution, 1.5 l: 60 days with 24 h cleaning interval
Maintenance interval	Every 6 months

Ordering Information Total Phosphorus OP 510

		Order No.
OP 510	Separate TresCon® analyzer module for total phosphorus for extension of an existing TresCon® system (requires 2 TresCon® measuring places)	820 011
TresCon® P 511	TresCon®-basic instrument with analysis module OP 510 for total phosphorus (wall mounting, space for 1 further module)	8A-8X030
	Accessories and Consumables see brochure "Product Details"	

Homogenizer available on demand (see brochure "Product Details")

PRODUCT INFORMATION
LABORATORY ANALYSIS
COLORIMETER
POCKET COLORIMETER II

POCKET Colorimeter II — Small in size, big on precision

The POCKET Colorimeter II has already proven its worth, but now it can serve additional application areas: These single parameter meters can now be used for key parameters to evaluate **LANGE cuvette tests** in addition to HACH tests. This dual functionality enables you to make the most of two key benefits: cost-effective meters combined with reagents for various quality standards. The result? Outstanding portable water analysis.
The numerous mini colorimeters on offer can serve a broad range of parameters from A (aluminium) to Z (zinc). Extensive measuring ranges for parameters such as ammonium, COD and chlorine extend the scope to include wastewater and drinking water applications.

HACH LANGE

UNITED FOR WATER QUALITY

40 POCKET mini colorimeters: Cost-effective, accurate field testing

→ **Extremely flexible; 40 single-parameter meters**

→ **Suitable for various quality standards owing to a diverse range of tests**

→ **Compact and simple to operate**

→ **Reliable results with no mains connection**

→ **Sturdy design, IP 67 standard**

From aluminium to zinc (A-Z), the diverse range of single-parameter instruments makes this mini meter a giant in the field of water analysis.

Proven analysis: Now with even greater flexibility

The POCKET Colorimeter II delivers rapid, reliable results for any application thanks to consistent LANGE cuvette tests and practical HACH tests.

Each mini meter can be used with the pre-configured calibration curves or self-programvming by the user for special applications. This makes it possible to determine a parameter using both feed and drain samples.

Optimum operation

You can benefit from the high quality design specifications:

- Simple operation: just four buttons are required to operate all functions.
- Safe handling: proven LANGE cuvette tests exclude sources of error instantly. ADDISTA standard solutions result in accurate quality assurance.
- All-weather equipment: optional backlighting, large digit display.

Strong on detail

These lightweight instruments weigh just 230 grams despite their sturdy design, but they can tolerate a lot! They are waterproof in line with IP 67, which is ideal for all weather conditions. The POCKETS are also suitable for prolonged use, with the batteries lasting for 2,000 analyses. All models come with a fully equipped carrying case.

Handy carrying case: The POCKET II and its accessories are safely stored and are ideal for field testing.

Parameters from aluminium to zinc (A-Z): The POCKET II evaluates PERMACHEM, ACCUVAC and LANGE cuvette tests.

LANGE reagents are easily assessed using the adapter.

Easy to carry: The lightweight yet sturdy design, combined with simple operation make the mini meter an unbeatable option for field testing.

Order information: LANGE cuvette tests & accessories

CUVETTE TESTS	ORDER NO.
Ammonium (Xn, N)*	
2–47 mg/l NH_4-N	LCK 303
0.015–2 mg/l NH_4-N	LCK 304
1–12 mg/l NH_4-N	LCK 305
Chloride (T, C)	
1.0–1,000	LCK 311
Chlorine (free, total)	
0.05–2 mg/l Cl_2/O_3	LCK 310
COD (T, C)	
15–150 mg/l	LCK 314
50–300 mg/l	LCK 614
150–1,000 mg/l	LCK 114
1,000–10,000 mg/l	LCK 014
Formaldehyde	
0.5–10 mg/l	LCK 325
Phosphate (C, Xn)	
0.05–1.5 mg/l PO_4-P	LCK 349
0.5–5 mg/l PO_4-P	LCK 348
2–20 mg/l PO_4-P	LCK 350
Zinc (Xn)	
0.2–6.0	LCK 360
ADDISTA standard solutions LCA700-709, see document	DOC062. 52.00269
ACCESSORIES & SPARE PARTS	
Pipette 0.2–1.0 ml	BBP 078
Pipette tips 0.2–1.0 ml	BBP 079
Pipette 1.0–5.0 ml	BBP 065
Pipette tips 1.0–5.0 ml	BBP 068
Empty cuvettes	LCW 919
POCKET light protection cap	LZV 759
Adapter for LANGE cuvette tests	5954600
Stands for LANGE cuvettes	LYW 915
LT 200 thermostat, Single block, for COD + PO_4 + Zn	LTV082.99. 10002

*Abbreviations of hazard symbols for reagents, see next page

Measuring with the POCKET II for use with LANGE cuvette tests

Out in the field: Samples are added directly for the pre-configured LANGE cuvette test. The cuvette is ready for use after a short development time.

During this waiting time, the POCKET is set to zero using the blank value cuvettes supplied. All it takes is the touch of a button. The POCKET is then ready to measure the sample cuvette.

Once the light protection cap has been fitted, measurement can start: Simply press the button to read. The large display will show the result.

Order information

POCKET COLORIMETER II IN CARRYING CASE

Meter for parameters incl. reagents	Measurement range [mg/l]	Method	No. of tests	Order no.
Aluminium (Xi)*	0.1–0.8	Aluminon	100	5870025
Ammonium (C, Xn)	0.01–0.8 NH_3-N	Salicylate	100	5870040
Bromine	0.05–4.5/0.2–10.0	DPD	50–100	5870001
Copper	0.04–5.0	Bicinchoninate	100	5870019
Chlorine, free + total	0.02–2.0/0.10–8.0	DPD	50	5870000
Chlorine, free + total **pH**	0.10–10.0 Cl_2 6.0–8.5 pH	DPD Phenol red	100 100+pH	5870012
Chlorine dioxide	0.05–5.0	DPD, Glycine	100	5870051
Chromium VI (Xi)	0.01–0.7	Diphenylcarbazide	100	5870017
Fluoride (C)	0.10–2.00	SPADNS	50	5870005
Iron (Xn)	0.02–5.0	FerroVer	100	5870022
Iron (Xn)	0.01–1.7	TPTZ	50	5870016
Lead (C)	0.005–0.15	Column extraction	20	5870021
Manganese (T, N)	0.01–0.70	PAN	100	5870018
Manganese (Xi, O)	0.20–20.0	Periodate oxidation	100	5870015
Molybdate	0.02–3.00/ 0.10–12.0	Ternary complex	100	5870010
Nickel + cobalt (T)	0.01–1.00 Ni 0.02–2.00 Co	PAN	100	5870020
Nitrate (T, N)	1.5–100 NO_3	Cd reduction	100	5870002
Oxygen (Xi)	0.20–10.0	HRDO	50	5870003
Ozone (Xn)	0.01–0.25/ 0.01–0.75	Indigo trisulphonate	50	5870004
Phosphate (ortho/total) (Xi)	0.02–3.00 PO_4	PhosVer 3	100	5870006
Phosphonate with UV lamp (Xi, Xn, O)	0.10–2.5/ 1.0–125 PO_4	PhosVer 3/UV	100	5870008
Silica (Xi)	1.0–100 SiO_2	Molybdate	100	5870034
Sulphate (Xn)	2.0–70	Turbidity	100	5870029
Zinc (Xn, N)	0.02–3.0	Zincon	100	5870009

POCKETS II FOR SPECIAL APPLICATIONS

Wavelength 420 nm	0–2.5 extinction			5870042
Wavelength 450 nm	0–2.5 extinction	Also for immunoassays		5870045
Wavelength 500 nm	0–2.5 extinction			5870050
Wavelength 528 nm	0–2.5 extinction			5870052
Wavelength 550 nm	0–2.5 extinction			5870055
Wavelength 580 nm	0–2.5 extinction			5870058
Wavelength 600 nm	0–2.5 extinction			5870060
Wavelength 665 nm	0–2.5 extinction			5870065

POCKETS II FOR LANGE CUVETTE TESTS**

Kit for ammonium	0.015–47 NH_4-N	LCK 303, LCK 304, LCK 305 (Xn, N)*	5953000V.01
Kit for chloride	1.0–1,000	LCK 311 (T, C)	5953000V.07
Kit for chlorine	0.05–2 Cl_2/O_3	LCK 310	5953000V.02
Kit for COD	15–300	LCK 314, LCK 614 (T, C)	5953000V.03
Kit for COD	150–10,000	LCK 114, LCK 014 (T, C)	5953000V.04
Kit for formaldehyde	0.5–10	LCK 325	5953000V.08
Kit for phosphate	0.05–20 PO_4-P	LCK 349, LCK 348, LCK 350 (C, Xn)	5953000V.05
Kit for zinc	0.2–6.0 Zn	LCK 350 (Xn)	5953000V.06

* Abbreviations of hazard symbols for reagents ** Cuvette tests must be ordered separately.

HACH LANGE Services

Contact us to place an order, request information or receive technical support.

Assurance of legal compliance, together with environmental protection via collection service for recycling of used reagents.

www.hach-lange.com
up to date and secure, with downloads, information and e-shop.

Hazard symbols for reagents

O	oxidising
T	Toxic
Xn	Harmful
Xi	Irritant
C	Corrosive
N	Dangerous for the environment

HACH LANGE GMBH
Willstätterstraße 11
D-40549 Düsseldorf
Tel. +49 (0)2 11 52 88-0
Fax +49 (0)2 11 52 88-143
info@hach-lange.de
www.hach-lange.com

UNITED FOR WATER QUALITY

BIO-CHEM VALVE™ INC.

Self-Priming Micro Pumps

Inside.....

Inert solenoid operated self-priming fixed displacement diaphragm pumps

- High accuracy dispense settings from 8µl to 250µl
- Low power consumption / minimal heat generation
- Low internal volume
- Positive shut off
- High cycle life - up to 20 million dispense cycles
- Choice of inert wetted materials

Broad range of dispense settings and flow rates

The diaphragm pumps are factory set for discrete outputs ranging from 8µl to 250µl. The pumps can be cycled at up to 2.0 Hz for the smallest version and 1.6 Hz for the largest version. Flow rates reach up to 25 ml / minute. Depending on the dispense setting, very high accuracy can be achieved with deviations from set-point of less than 1%. For optimal accuracy, the pumps should be used for the transfer of fluids between un-pressurized containers.

Self-priming

At startup, the pump is able to draw air. The suction created by the larger pumps is sufficient to pull liquids from an un-pressurized container located up to 1.3 meters (4' 3") beneath the pump. Once the pump is primed, it is able to generate around 5 psi (0.3 bar) pressure, equating to 3.5 meters (11' 6") of water.

Wide assortment of inert wetted materials available

The pumps provide a non-metallic, inert fluid path for the dispensing of high purity or aggressive fluids. The standard pump body is made of PPS (polyphenylsulfide). Other materials available for the pump body include PTFE, PEEK™ and Delrin®. The elastomers that can be used for the diaphragms and check valves include PTFE, EPDM, and Viton®.

High reliability

The pumps are designed for continuous duty. They are guaranteed for up to 20 million actuations, corresponding to nearly 3,000 hours of continuous use at a 2 Hz cycle rate.

Specifications

Pump series & weight

The self-priming pumps are offered in four pump series, distinguished by the solenoid shell sizes:

Valve Series	Shell Diameter	Weight
090SP	0.75 inches	2 oz. (60 g)
120SP	1.00 inches	5 oz. (140 g)
130SP	1.00 inches	6 oz. (170 g)
150SP	1.50 inches	16 oz. (450 g)

Electrical

Valve Series	Voltage	Power @ 70^{0}F (21^{0}C)	Current @ 70^{0}F (21^{0}C)	Effective continuous power at max cycle rate
090SP	12 Vdc	2.6 Watts	0.22 amps	0.8 Watts
090SP	24 Vdc	2.6 Watts	0.11 amps	0.8 Watts
120SP	12 Vdc	4.0 Watts	0.32 amps	1.2 Watts
120SP	24 Vdc	4.0 Watts	0.16 amps	1.2 Watts
130SP	12 Vdc	4.0 Watts	0.32 amps	1.2 Watts
130SP	24 Vdc	4.0 Watts	0.16 amps	1.2 Watts
150SP	12 Vdc	8.0 Watts	0.66 amps	3.2 Watts
150SP	24 Vdc	8.0 Watts	0.33 amps	3.2 Watts

Lead Wires

All lead wires are Teflon® coated. Different lengths of lead wires and terminal connectors can be provided. Please consult Bio-Chem Valve and Omnifit concerning non-standard lead wire lengths and the addition of terminal connectors.

Valve Series	Lead wire length	Wire thickness
090SP	15 in. / 38cm	26 gauge
120SP	15 in. / 38cm	26 gauge
130SP	15 in. / 38cm	26 gauge
150SP	15 in. / 38cm	22 gauge

Specifications (contd.)

Pressure limits

To attain optimal dispense accuracy, pressure on both the inlet and the outlet side of the pump must be kept between ± 0.5 psi (0.035 bar) and 0 psi (equating to a head of ± 12 in / 30 cm of water).

During the pump's up-stroke, suction is created on the inlet. Positive pressure is generated at the outlet during the down-stroke. When the pump is not actuated, it will shut off flow as long as the pressure on the inlet does not exceed the maximum holding pressure. To ensure correct operation, pressure on the inlet side should never exceed 2 psi even when the pump is in the closed position. The check valves in the pump prevent fluid from flowing against the intended flow direction.

Valve Series	Inlet suction (priming w/air)	Inlet suction (after priming)	Outlet pressure
090SP	1 psi / 0.07 bar	3 psi / 0.20 bar	3 psi / 0.20 bar
120SP	2 psi / 0.15 bar	5 psi / 0.35 bar	5 psi / 0.35 bar
130SP	2 psi / 0.15 bar	5 psi / 0.35 bar	5 psi / 0.35 bar
150SP	2 psi / 0.15 bar	5 psi / 0.35 bar	5 psi / 0.35 bar

Cycle rates

To operate the solenoid micro pumps, first place the rated voltage over the pump's coil (causing the intake stroke) and then turn the voltage off again (causing the output stroke). To achieve the optimal dispense accuracy, the pump must remain in the on and in the off position for a minimum amount of time. To operate the pumps at less than the maximum cycle rate, the on-time should remain unchanged and the off-time should be lengthened appropriately.

Valve Series	Minimum on time	Minimum off time	Maximum cycle rate
090SP	150 msec	350 msec	2.0 Hz
120SP	150 msec	350 msec	2.0 Hz
130SP	150 msec	350 msec	2.0 Hz
150SP	200 msec	400 msec	1.6 Hz

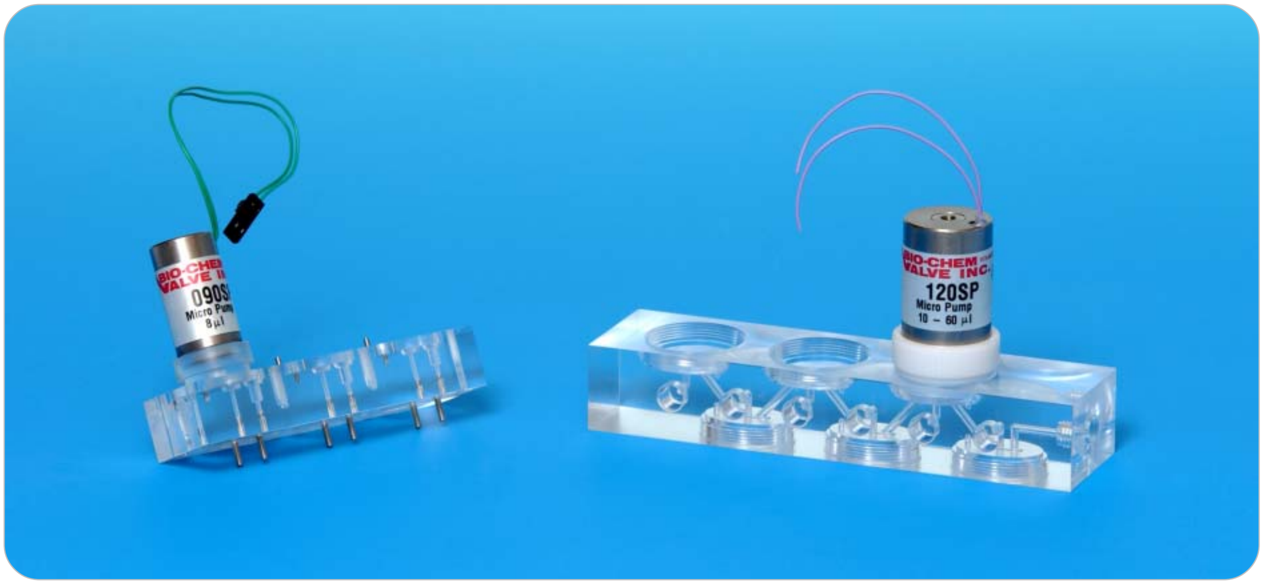
Dispense volume accuracy

Valve Series		Minimum setting	Maximum setting	Max flow rate
090SP	Dispense volume:	8 µl	8 µl	1.0 ml / min
	Set-point accuracy:	+/- 25%	+/- 25%	
	Repeatability:	+/- 5%	+/- 5%	
120SP	Dispense volume:	10 µl	60 µl	7.2 ml / min
	Set-point accuracy:	+/- 4%	+/- 2%	
	Repeatability:	+/- 3%	+/- 1%	
130SP & 120SP w/PTFE dia.	Dispense volume:	10 µl	50 µl	6.0 ml / min
	Set-point accuracy:	+/- 5%	+/- 5%	
	Repeatability:	+/- 3%	+/- 1%	
150SP	Dispense volume:	100 µl	250 µl	25 ml / min
	Set-point accuracy:	+/- 5%	+/- 3%	
	Repeatability:	+/- 1%	+/- 0.5%	

Specifications (contd.)

Manifold mounted pumps

The Bio-Chem Valve diaphragm pumps can be mounted directly on manifolds. Many configurations are possible to meet the specific needs of the application. Please consult with Bio-Chem Valve regarding manifold mounted pump designs.



Ports and tubing

To achieve an optimal dispense accuracy, the tubing should conform to the recommended inner diameters stated below and friction losses should be kept to a minimum.

For more information regarding 1/4"-28 fittings and tubing sets, please consult the Omni-Lok^{TM} Fitting System and Fitting Systems specification sheets.

Pump Series	Standard ports	Recommended inlet tubing size	Recommended outlet tubing size
090SP	1/4"-28 flat bottom	≥ 1/32" I.D.	≥ 1/32" I.D.
120SP	1/4"-28 flat bottom	≥ 1/32" I.D.	≥ 1/32" I.D.
130SP	1/4"-28 flat bottom	≥ 1/32" I.D.	≥ 1/32" I.D.
150SP	5/16"-24 flat bottom	≥ 1/8" I.D.	≥ 1/8" I.D.

Internal volume

Pump Series	Internal volume
090SP	130 µl
120SP	105 µl
130SP	105 µl
150SP	710 µl

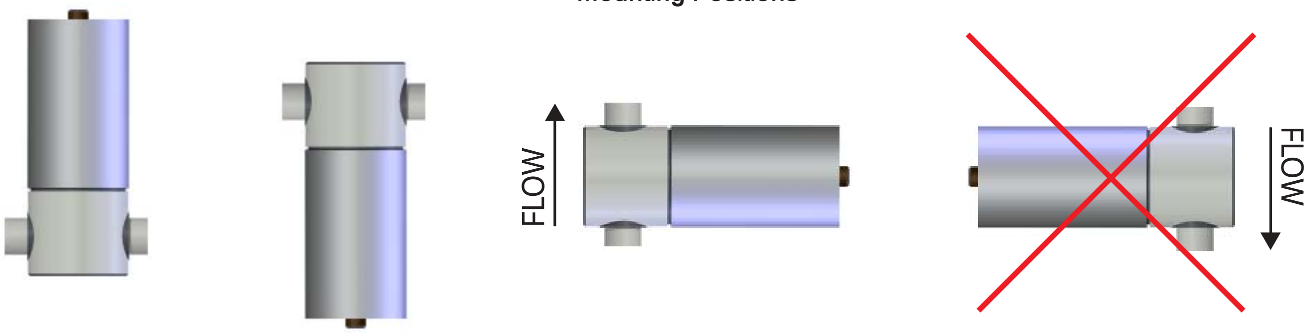
Specifications (contd.)

Mounting

For mounting clips, rings and flanges, please see the Mounting Accessories and Options specification sheet. Two mounting holes in the body permit bottom mounting in the 120SP series pumps. Please see the dimensional diagrams on the following pages for further information.

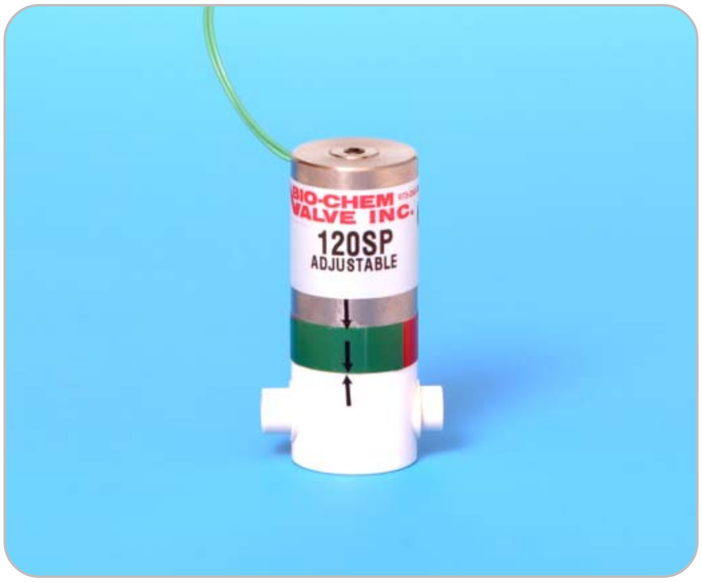
The pumps should be installed with the solenoid portion of the pump pointing upwards, downwards, or in a horizontal position with the outlet on top. The horizontal position with the outlet on the bottom could affect the output accuracy.

Mounting Positions



Adjustable pumps

In cases where the dispense volume for the application is unknown, Bio-Chem Valve offers an adjustable pump designed to be used as an R&D tool by the instrument design engineer to help determine the optimal dispense setting. By turning the shell on the pump, the engineer can compensate for the instrument's friction losses or pressure imbalances. Once the correct pump setting and dispense volume have been determined, the engineer can provide the dispense volume to the factory. This information will be used to manufacture a factory-set (non-adjustable) pump specific to the application.



Pump Series	Dispense range
090SP - adjustable	8 µl to 12 µl
120SP - adjustable	10 µl to 60 µl
150SP - adjustable	100 µl to 250 µl

Aside from the adjustability of these pumps, all other specifications are identical with those stated in this product data sheet for the same pump series.

For part numbers of the adjustable pump versions, please contact Bio-Chem Valve and Omnifit.

Specifications (contd.)

Wetted materials

Valve Series		Standard materials	Options
090SP	Body:	PPS	PEEK™
	Diaphragm:	PTFE	
	Check valves:	EPDM	Viton®, Perfluoroelastomer
120SP	Body:	PPS	Delrin®, PEEK™
	Diaphragm:	EPDM	PTFE
	Check valves:	EPDM	Viton®, Perfluoroelastomer
130SP	Body:	PTFE	
	Diaphragm:	PTFE	
	Check valves:	Perfluoroelastomer	
150SP	Body:	PPS	PEEK™
	Diaphragm:	EPDM	
	Check valves:	EPDM	

Ordering Information

1	Select pump series	090SP, 120SP, 130SP, 150SP
2	Indicate voltage	12 VDC, 24 VDC
3	Indicate dispense setting (µl)	8 (090SP), 10, 20, 30, 40, 50, 60 (120SP), 10, 20, 30, 40, 50 (130SP), 100, 125, 150, 175, 200, 225, 250 (150SP)
4	Indicate body material (See optional materials in chart above.)	1 (PTFE, 130SP only), 4 (PPS), 5 (PEEK), 6 (Delrin®, 120SP only)
5	Indicate diaphragm material (See optional materials in chart above.)	E (EPDM), T (PTFE)
6	Indicate check valve material (See optional materials in chart above.)	E (EPDM), V (Viton®), P (Perfluoroelastomer)

Part Number Example:

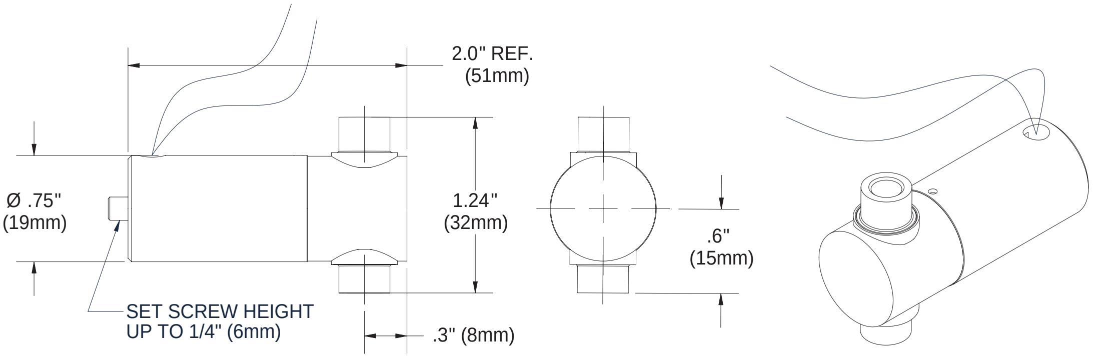
120SP	24	50	-	5	T	P
Pump Series	Voltage	Dispense setting		Body material	Diaphragm material	Check valve material

Please see our other product specification sheets for the following accessories:

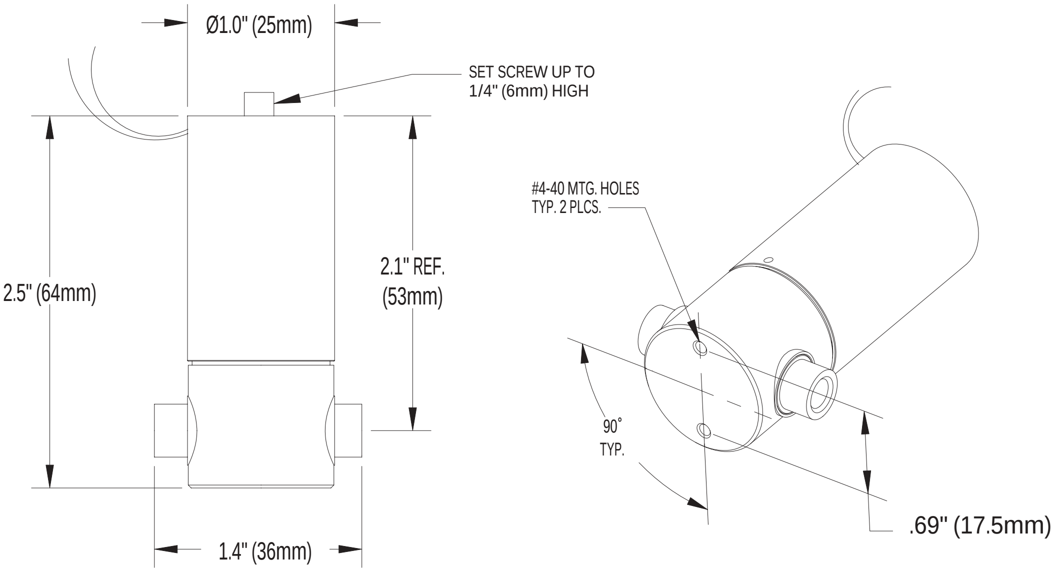
- Omni-Lok™ fittings and tubing sets
- CoolCube™ control module
- Mounting clips

Installation Drawings

090SP

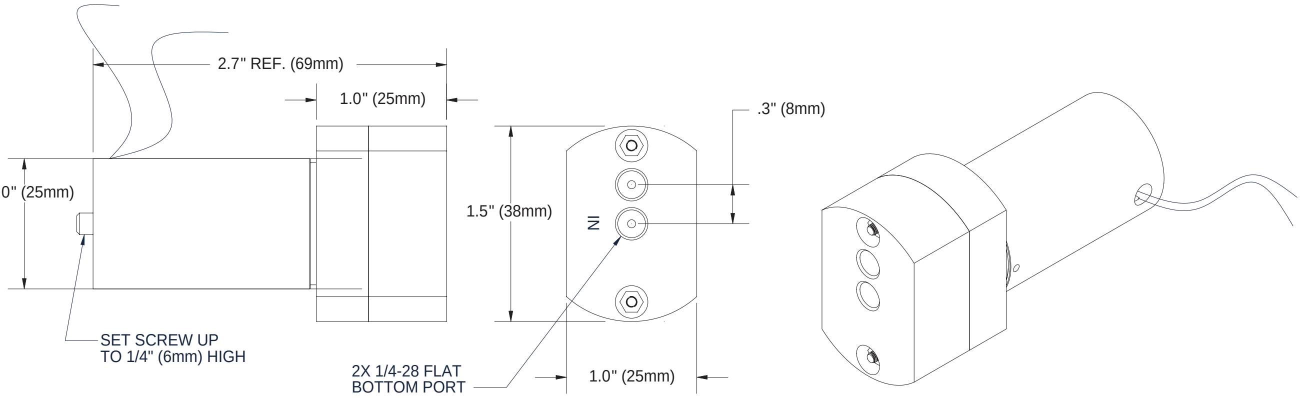


120SP

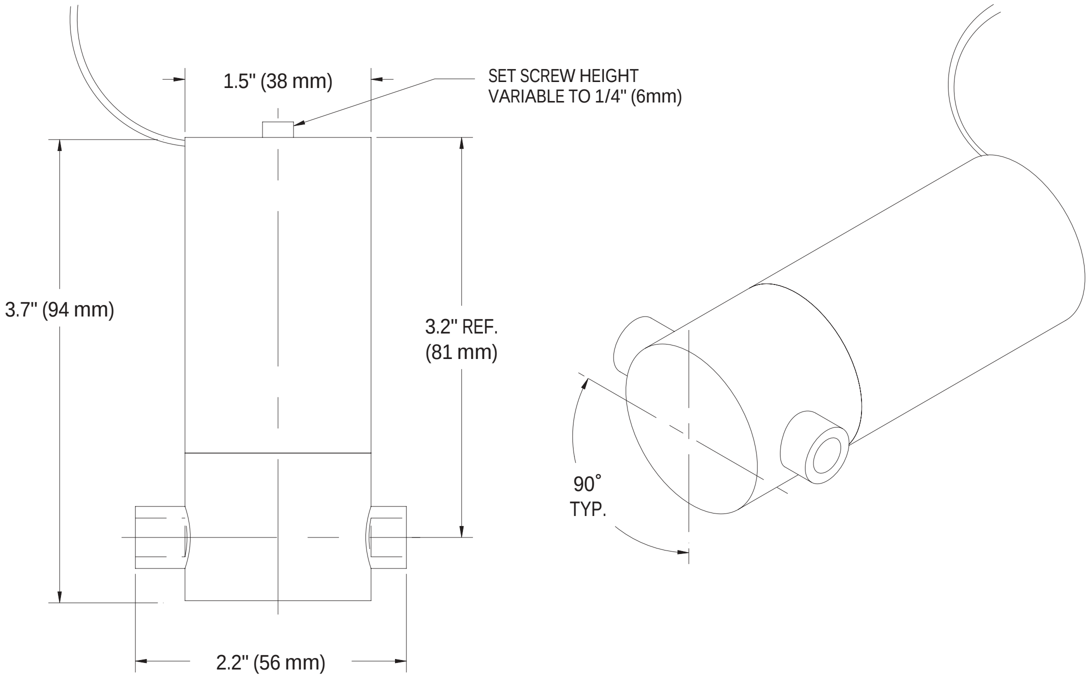


Installation Drawings (contd.)

130SP



150SP



Trademarks:

CoolCube™ is a trademark of Bio-Chem Valve Inc.
Delrin®, Teflon®, **Viton®** are registered trademarks of E.I. du Pont de Nemours and Company
Omni-Lok™ is a trademark of Omnifit Ltd.
PEEK™ is a trademark of Victrex plc

msscientific Chromatographie-Handel GmbH
Gneisenaustrasse 66/67 ·10961 Berlin ·Germany
Fon: +49 30 6270 6087 ·Fax: +49 30 6270 6089
e-mail: info@msscientific.de ·Web: www.msscientific.de

Bio-Chem Valve / Omnifit, 2 College Park, Coldhams Lane, Cambridge, CB1 3HD UK
t: +44 (0) 1223 416642 f: +44 (0) 1223 416787 e: sales@omnifit.com w: omnifit.com

Rev. 0805

Bio-Chem Valve / Omnifit, 85 Fulton Street, Boonton, NJ 07005 USA
t: 973 263 3001 f: 973 263 2880 e: info@bio-chemvalve.com w: bio-chemvalve.com

A HALMA GROUP COMPANY

Honeywell

Installation Instructions for the **26PC Series Pressure Sensors**

ISSUE 4

PK 80083

⚠ WARNING

PERSONAL INJURY

DO NOT USE these products as safety or emergency stop devices, or in any other application where failure of the product could result in personal injury.

Failure to comply with these instructions could result in death or serious injury.

GENERAL INFORMATION

26PC Series pressure sensors are four active element piezoresistive bridges. When pressure is applied, the resistance changes and the 26PC provides an output signal proportional to the input pressure.

Gage pressure is measured with respect to ambient pressure. When applied pressure increases, the differential voltage V2 - V4 increases. As pressure decreases, differential voltage V2 - V4 decreases.

Differential sensors provide a differential voltage proportional to the pressure differential between port P2 and P1. As DP = P2 – P1 increases, differential voltage V2 - V4 increases. As DP = P2 – P1 decreases, differential voltage V2 - V4 decreases.

SOLDERING

Limit soldering to 315 °C [600 °F] maximum, with duration of 10 seconds maximum.

CLEANING

Proper cleaning fluids should be selected, based on the type of contaminants to be removed. Honeywell recommends alcohols or fluorinated solvents. Do not immerse the sensor.

26PC SERIES PERFORMANCE CHARACTERISTICS

at 10.0 ± 0.01 Vdc Excitation, 25 °C

	Min.	Typ.	Max.	Units
Excitation	—	10	16	Vdc
Response Time	—	—	1.0	ms
Input Resistance*	5.5 k	7.5 k	11.5 k	Ohm
Output Resistance*	1.5 k	2.5 k	3.0 k	Ohm
Weight		2		gram

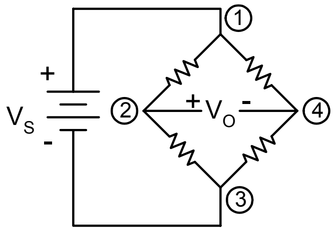
* Measured using a 1 mA current

ENVIRONMENTAL SPECIFICATIONS

Operating Temperature	-40 °C to 85 °C [-40 °F to 185 °F]
Storage Temperature	-55 °C to 100 °C [-67 °F to 212 °F]
Compensated Temperature	0 °C to 50 °C [32 °F to 122 °F]
Shock	Qualification tested to 150 g
Vibration	MIL-STD-202, Method 213 (0 kHz to 2 kHz, 20 G sine)

Note: For media compatibility specifications, refer to catalog or web site: www.honeywell.com/sensing

26PC CIRCUIT TERMINATION



Pin 1 = Vs (+)	Pin 4 = Output - (V4)
Pin 2 = Output + (V2)	Pin 3 = Ground (-)

Pin 1 is notched

Pin 2 is next to Pin 1, etc.

26PC LASER BRANDING SCHEME (Note: Alternate form is the entire catalog listing)

CATALOG LISTING STAMP

<u>6</u> <u>B</u> <u>F</u> <u>6</u> <u>G</u> <u>XX</u>

XX = Special (Sequentially numbered)
G = Pressure measurement: G = Gage, D = Differential
6 = Termination style: 2 = 2 x 2, 5 = Wire Harness, or 6 = 1 x 4
F = Media seal: F = Fluorosilicone, E = EPDM, N = Neoprene, S = Silicone, V= Viton
B = Range (psi) : A = 1, B = 5, C = 15, D = 30, F = 100, G = 250, J = 38, K = 38
6 = Type: (6 = Compensated)

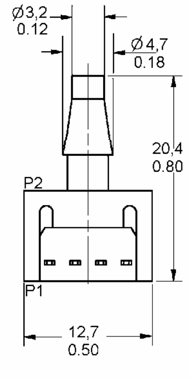
DATE STAMP CODE

2002 **10** (Note: Alternate date code is 102002)

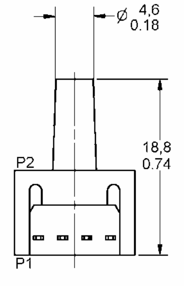
10 = Week in the year
2002 = Year

ADDITIONAL PORT VARIATIONS (dimensions for reference only) mm/in

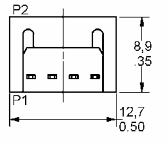
B Barbed



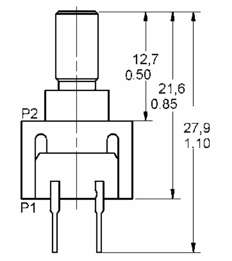
C Luer



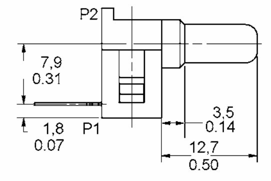
D Modular



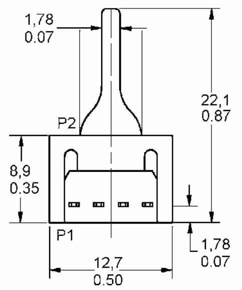
H M5 Thread



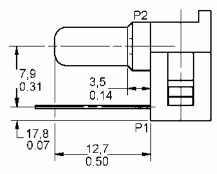
I 90° Port



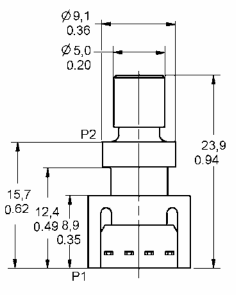
J Needle *



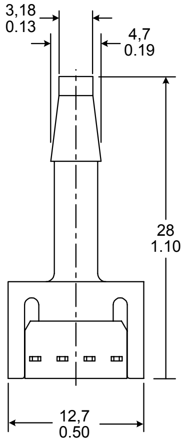
K Reverse 90° Port



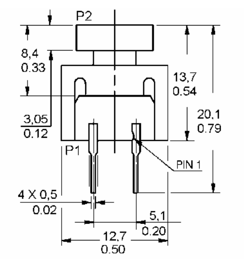
M ¼-28 Thread**



T Long Barbed



S Manifold

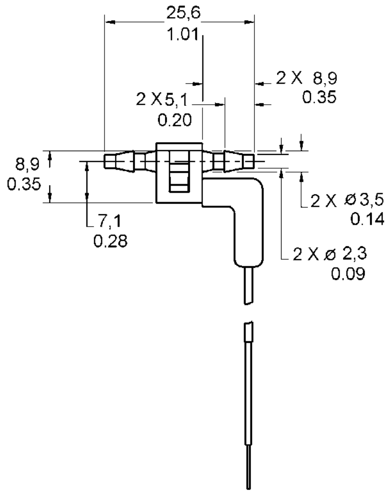


* Recommended torque for sealing is 4 in-lb. Do not exceed 6 in-lb of torque. Use size 007 O-Ring. O-Ring counterbore dimensions are 0.04 ± .005 in D x 0.300 ± .003 in Dia.

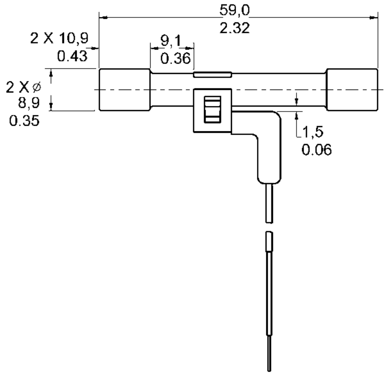
** Recommended torque for sealing is 8 in-lb. Do not exceed 12 in-lb. Use size 009 O-Ring. O-Ring counterbore dimensions are .040 ± .002 in D x 0.360 ± .003 in Dia.

ADDITIONAL PORT VARIATIONS (dimensions for reference only) mm/in

G Small Flow Through



N or U Large Flow Through

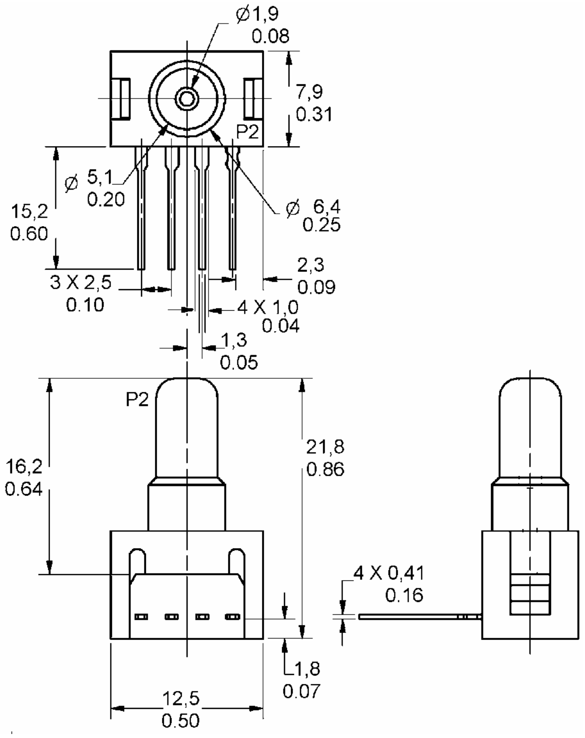


MOUNTING DIMENSIONS (dimensions for reference only) mm/in

GAGE SENSOR

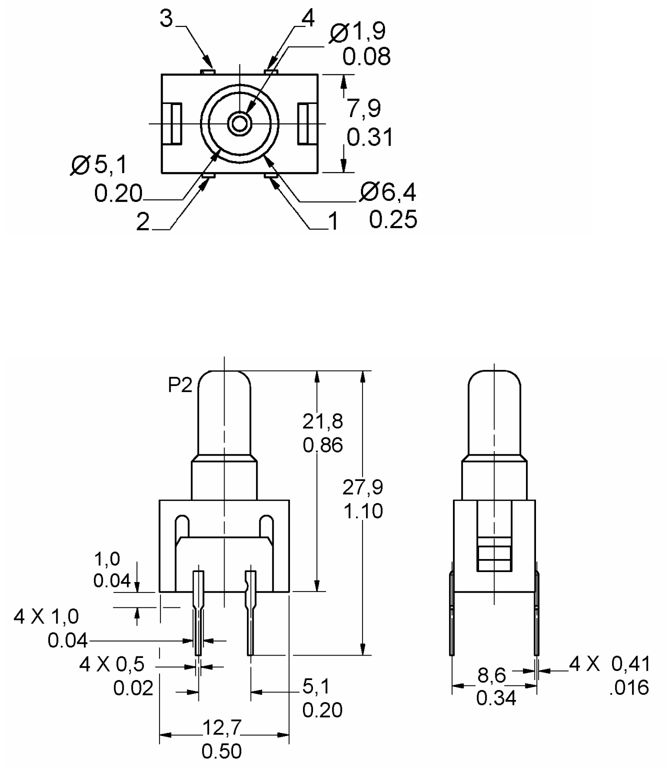
1 x 4 Termination (Style 6) Port Style A, Straight

Pin 1 is notched, Pin 2 is next to Pin 1, etc.



2 x 2 Termination (Style 2) Port Style A, Straight

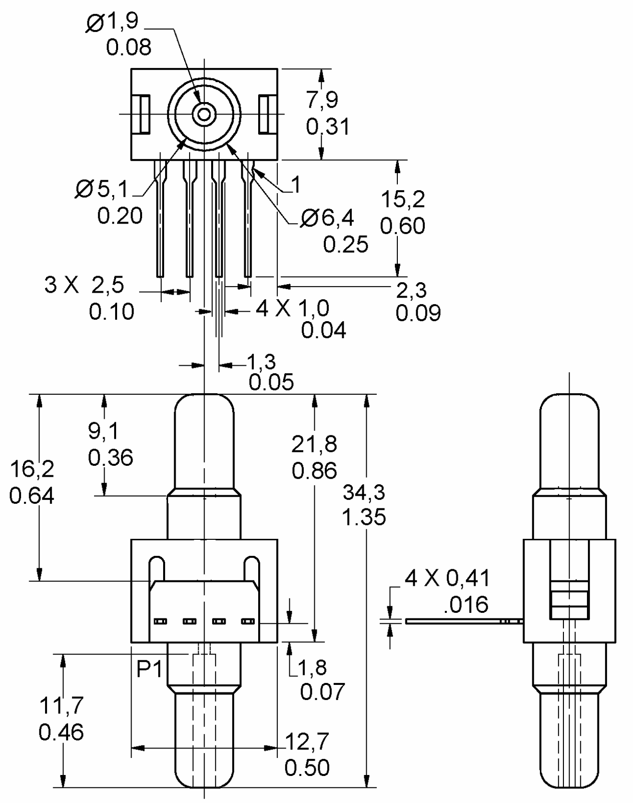
Pin 1 is notched, Pin 1 is shown at lower right corner. Pins 2, 3, 4 are clockwise.



MOUNTING DIMENSIONS (for reference only) mm/in

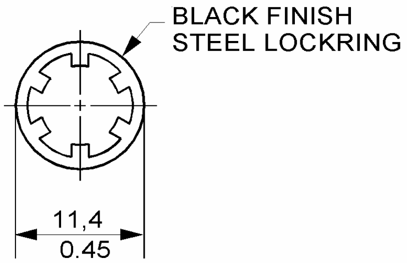
DIFFERENTIAL SENSOR

1 X 4 Termination (Style 6) Port Style A, Straight (Only)



ACCESSORY

PC-10182 — Steel Lockring



WARRANTY/REMEDY

Honeywell warrants goods of its manufacture as being free of defective materials and faulty workmanship. Contact your local sales office for warranty information. If warranted goods are returned to Honeywell during the period of coverage, Honeywell will repair or replace without charge those items it finds defective. The foregoing is Buyer’s sole remedy and is **in lieu of all other warranties, expressed or implied, including those of merchantability and fitness for a particular purpose.**

Specifications may change without notice. The information we supply is believed to be accurate and reliable as of this printing. However, we assume no responsibility for its use.

While we provide application assistance personally, through our literature and the Honeywell web site, it is up to the customer to determine the suitability of the product in the application.

For application assistance, current specifications, or name of the nearest Authorized Distributor, contact a nearby sales office. Or call:

1-800-537-6945 USA
1-800-737-3360 Canada
1-815-235-6847 International

FAX

1-815-235-6545 USA

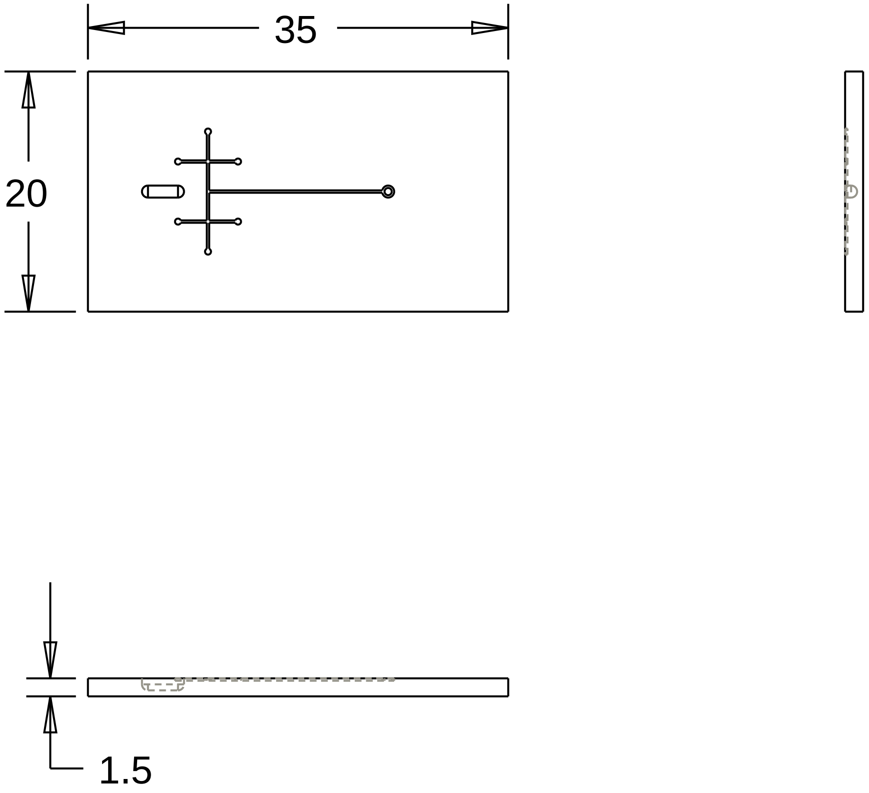
INTERNET

www.honeywell.com/sensing
info.sc@honeywell.com

Honeywell

Automation and Control Products
Honeywell
11 West Spring Street
Freeport, Illinois 61032

www.honeywell.com/sensing



Location:

Dublin City University

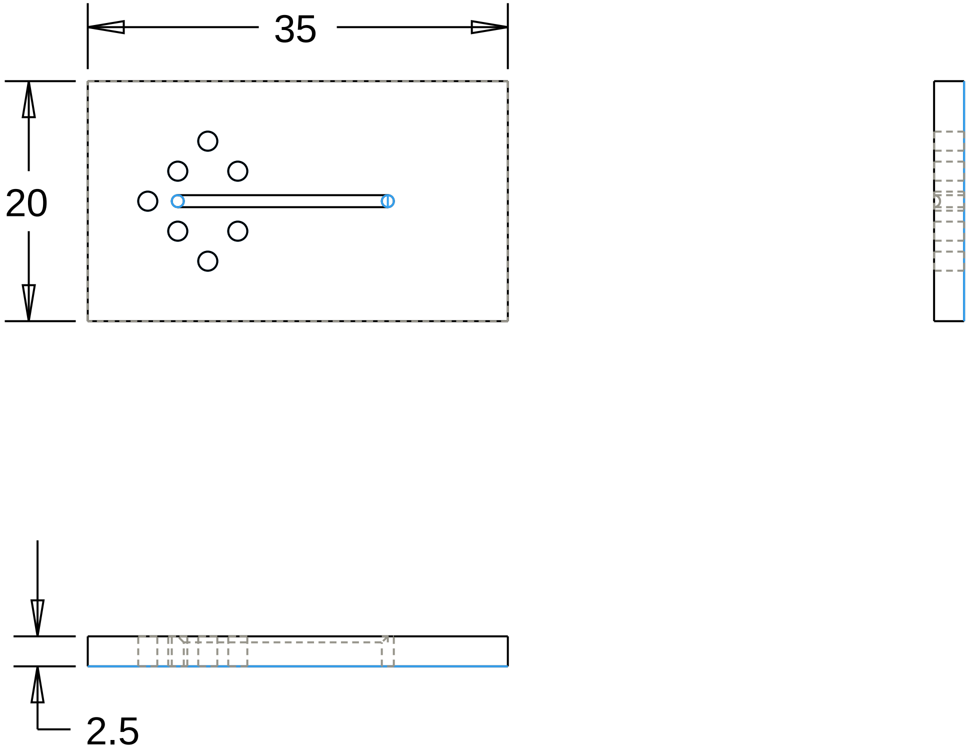
Title:

PO4_CHIP_BOTTOM

Name: Conor Slater	Date: 13-7-2009

Scale:

1:1



Location:

Dublin City University

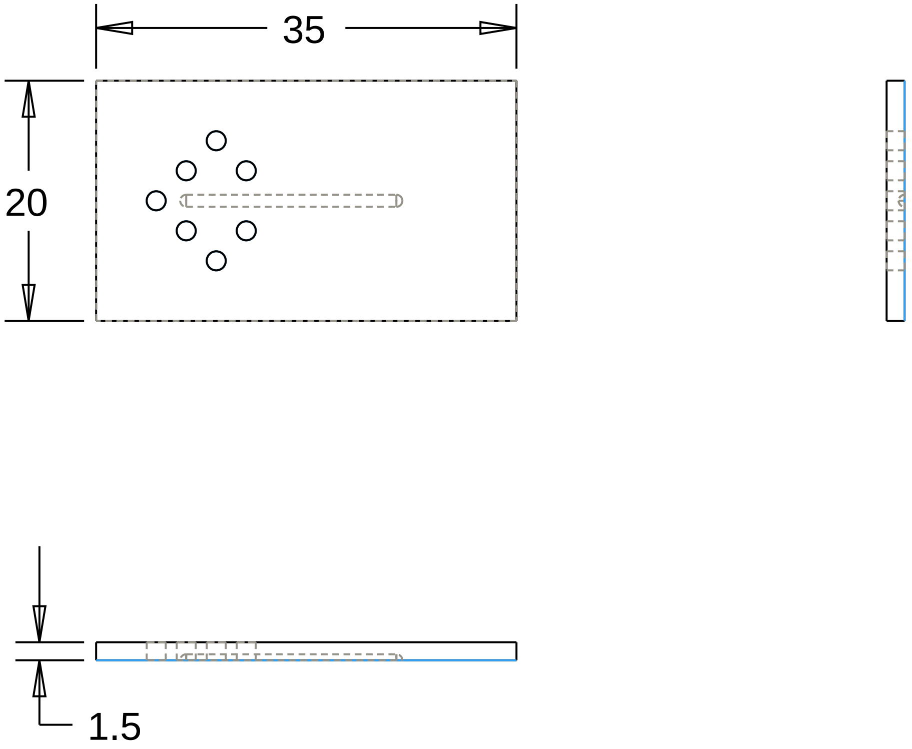
Title:

PO4_CHIP_MIDDLE

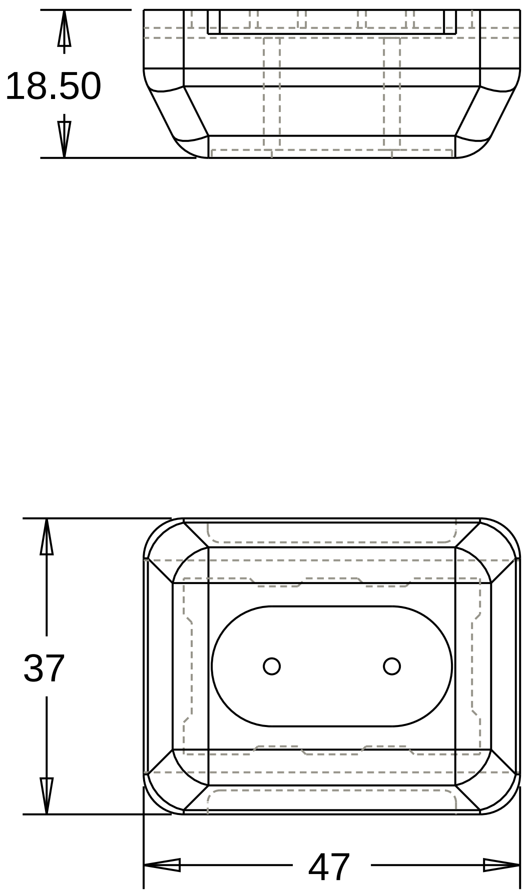
Name:	Date:
Conor Slater	13-7-2009

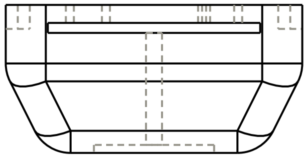
Scale:

1:1

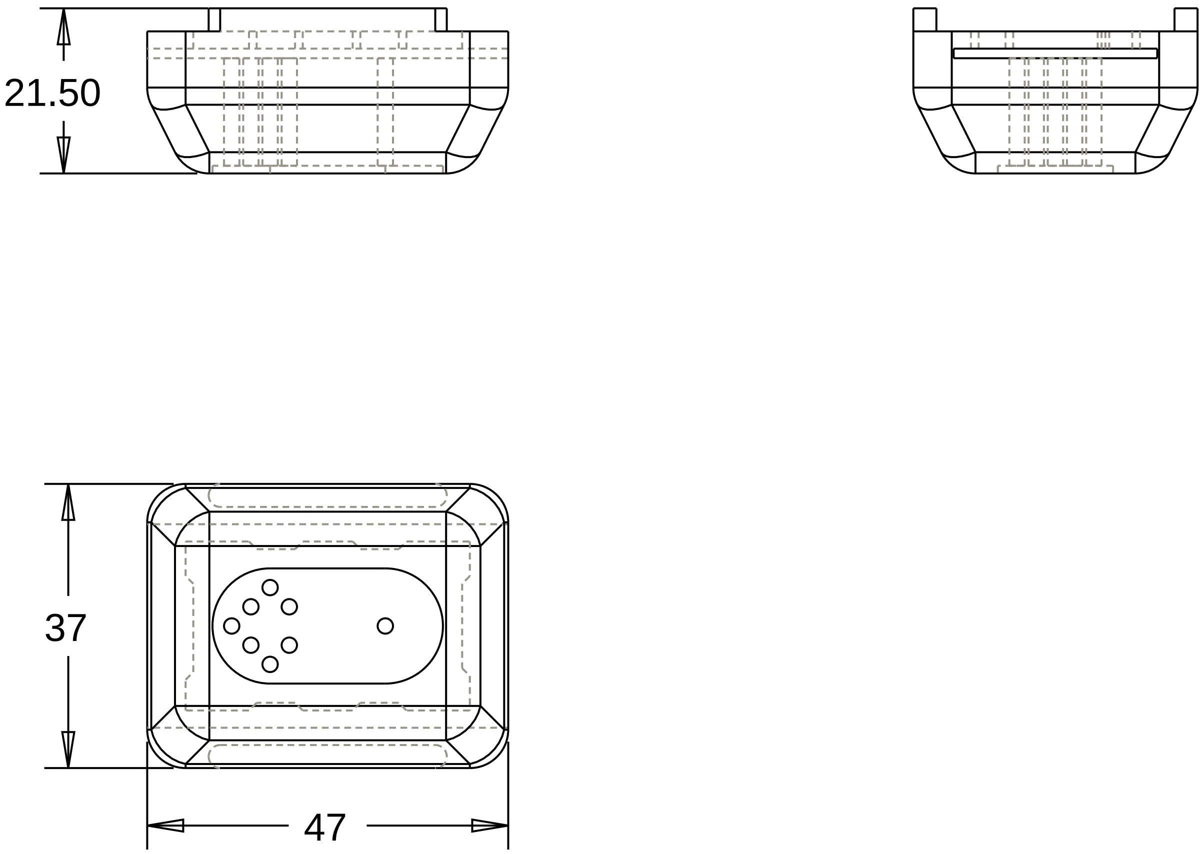


Location: Dublin City University	
Title: PO4_CHIP_TOP	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:1	

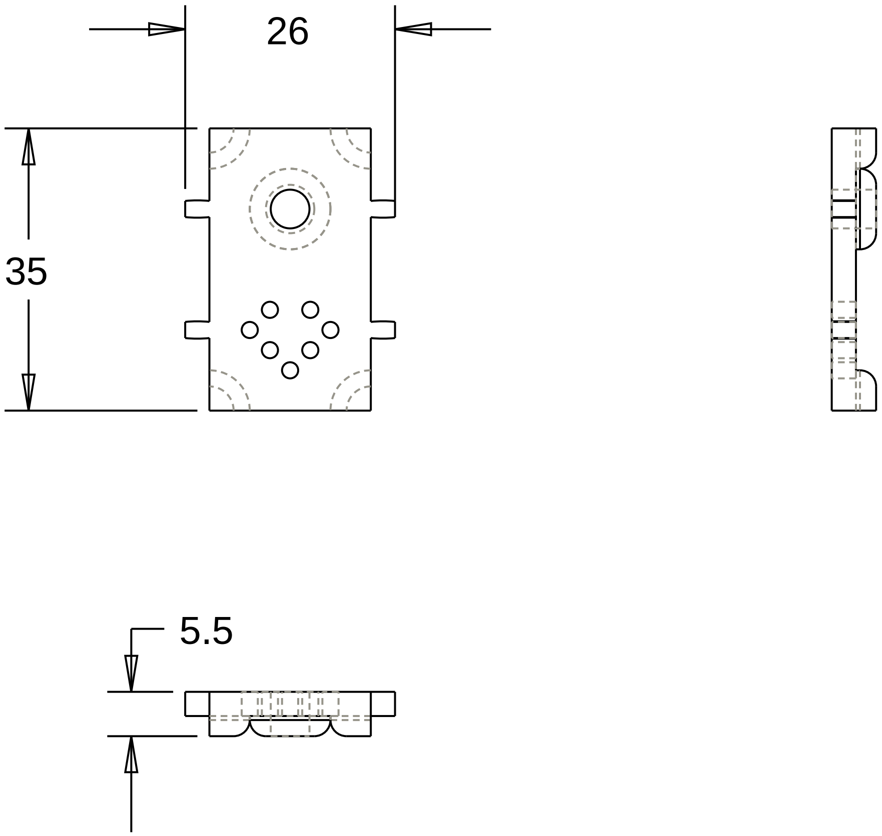




Location:	
Dublin City University	
Title:	
ASSEMBLY_JIG_BOTTOM	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:1	



Location: Dublin City University	
Title: ASSEMBLY_JIG_TOP	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:1	



Location:

Dublin City University

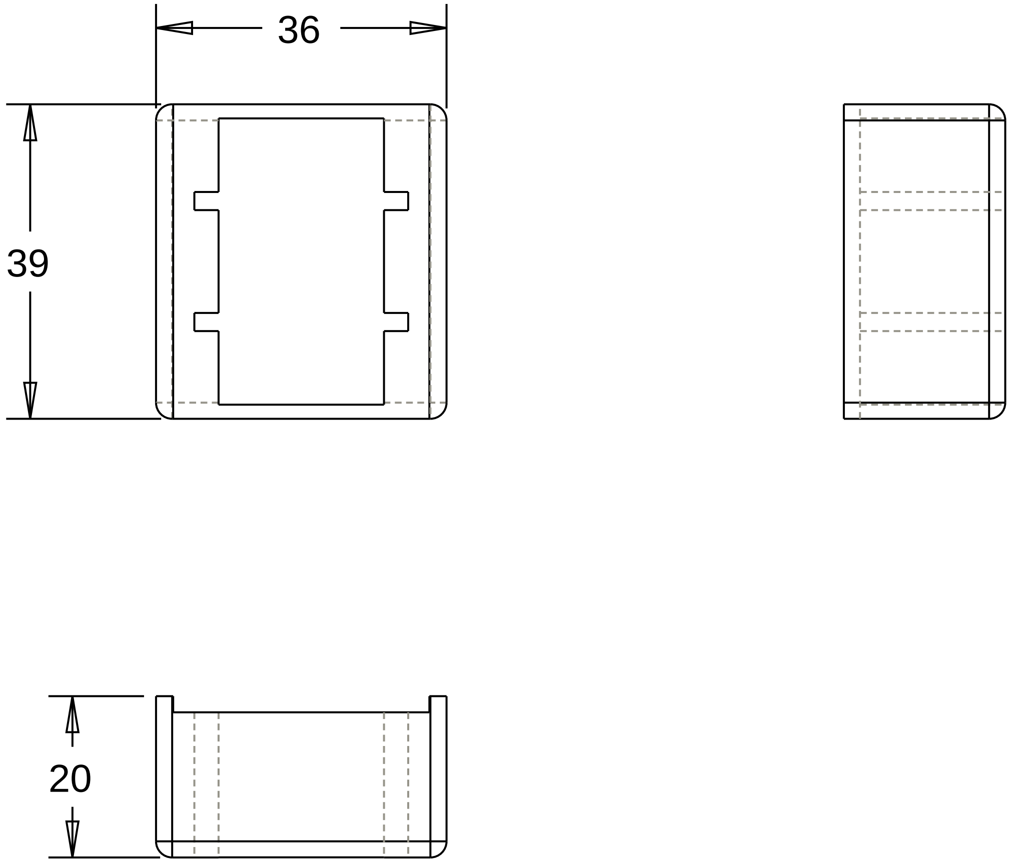
Title:

CHIP_COVER

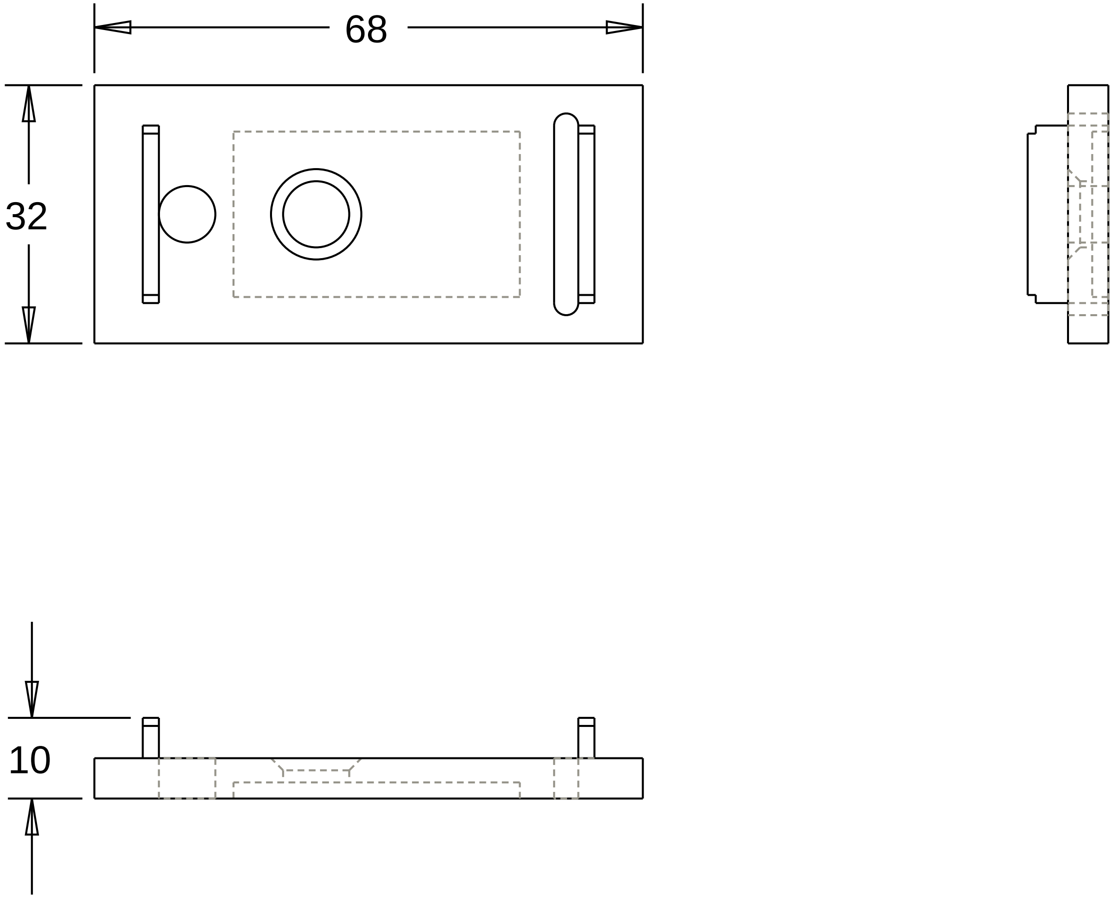
Name: Conor Slater	Date: 13-7-2009

Scale:

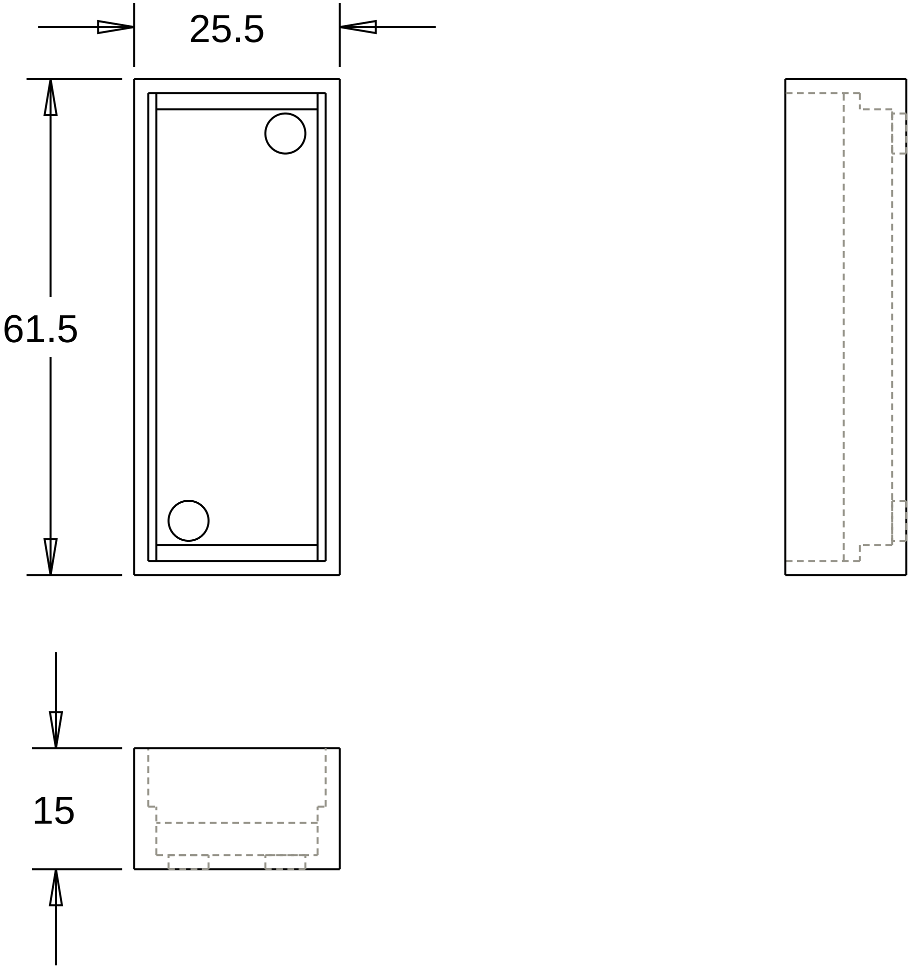
1:1



Location: Dublin City University	
Title: CHIP_COVER_MOUNT	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:1	



Location: Dublin City University	
Title: CHIP_MOUNT	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:1	



Location:

Dublin City University

Title:

CHIP_ENCLOSURE

Name:

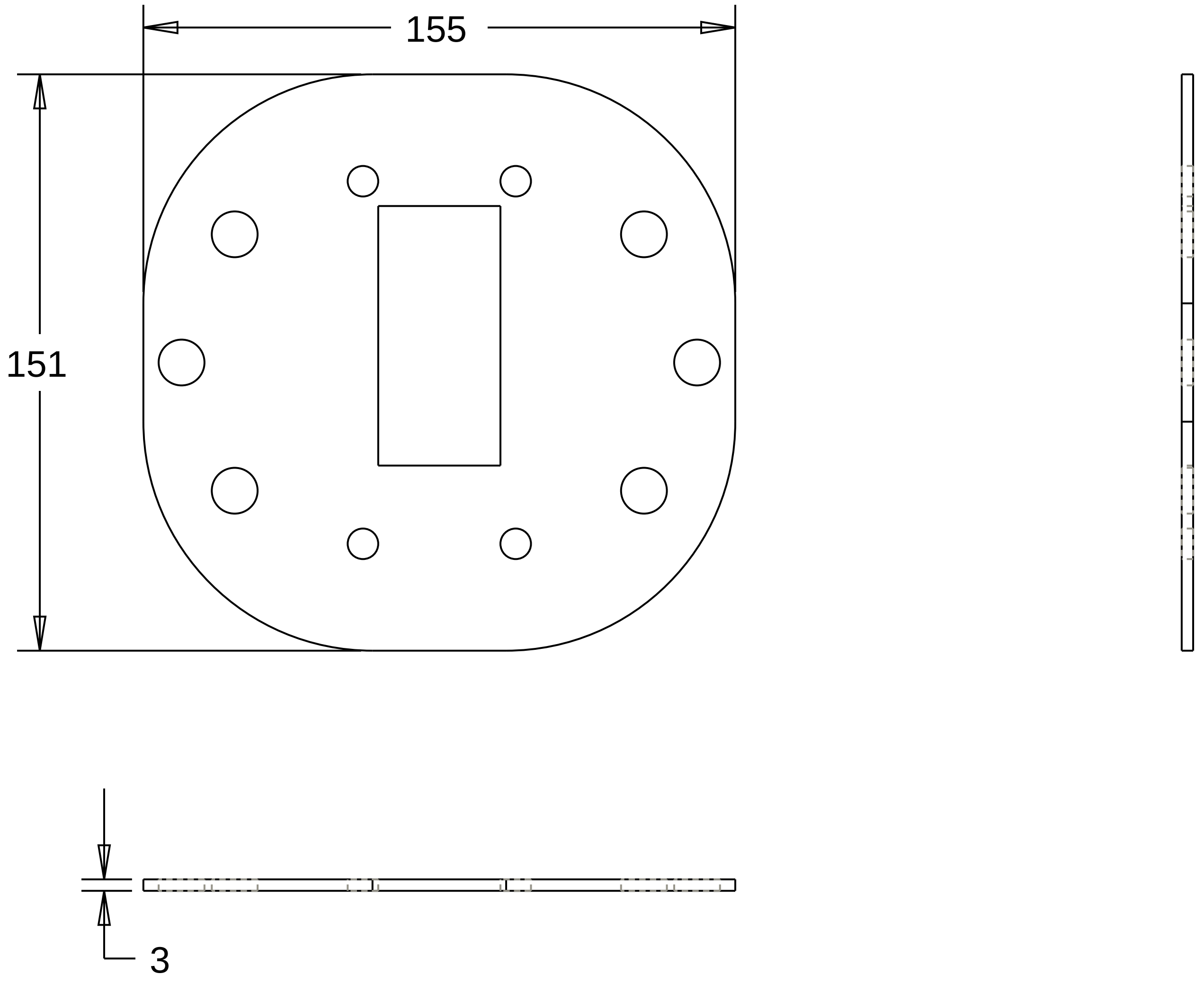
Conor Slater

Date:

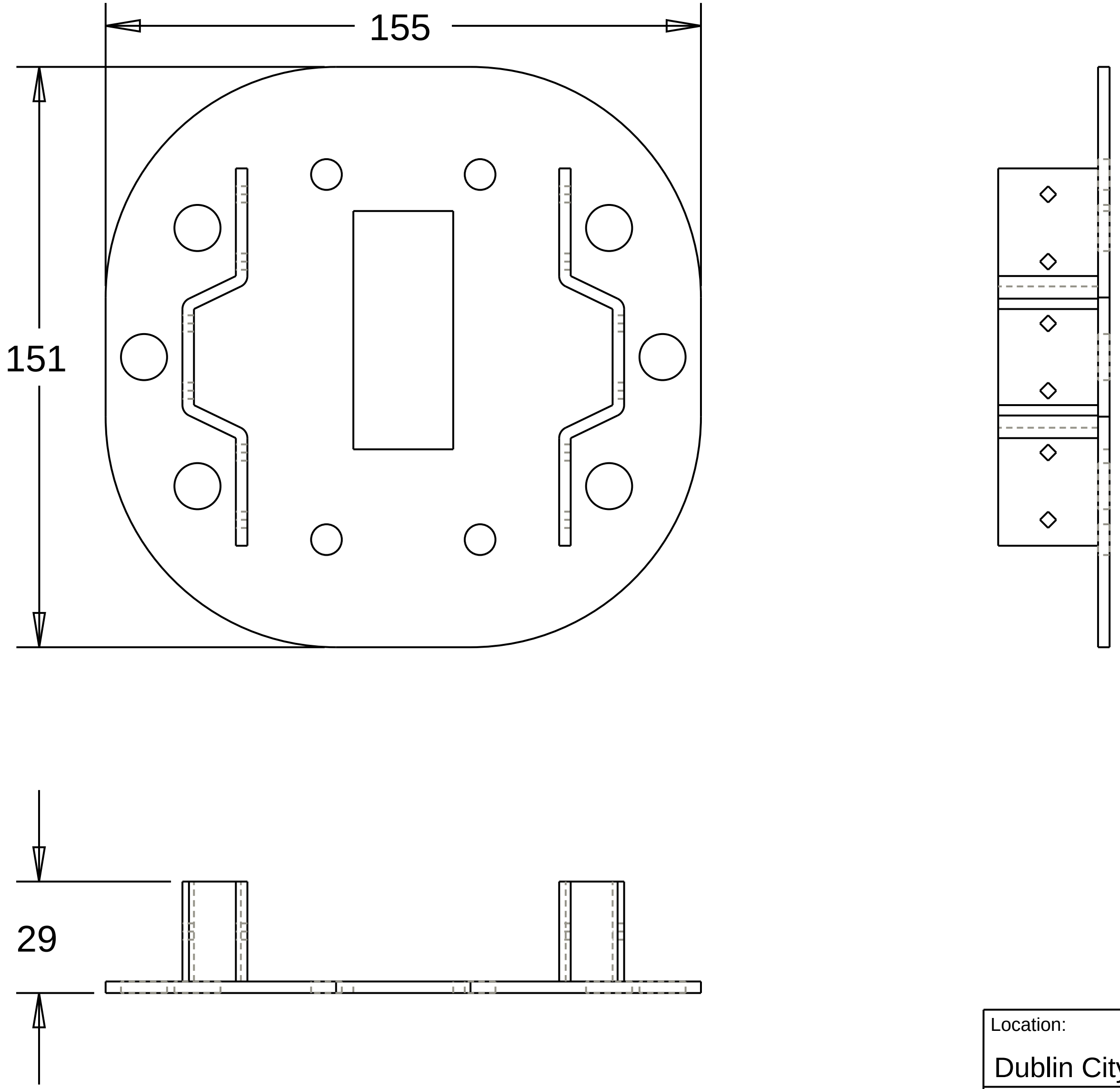
13-7-2009

Scale:

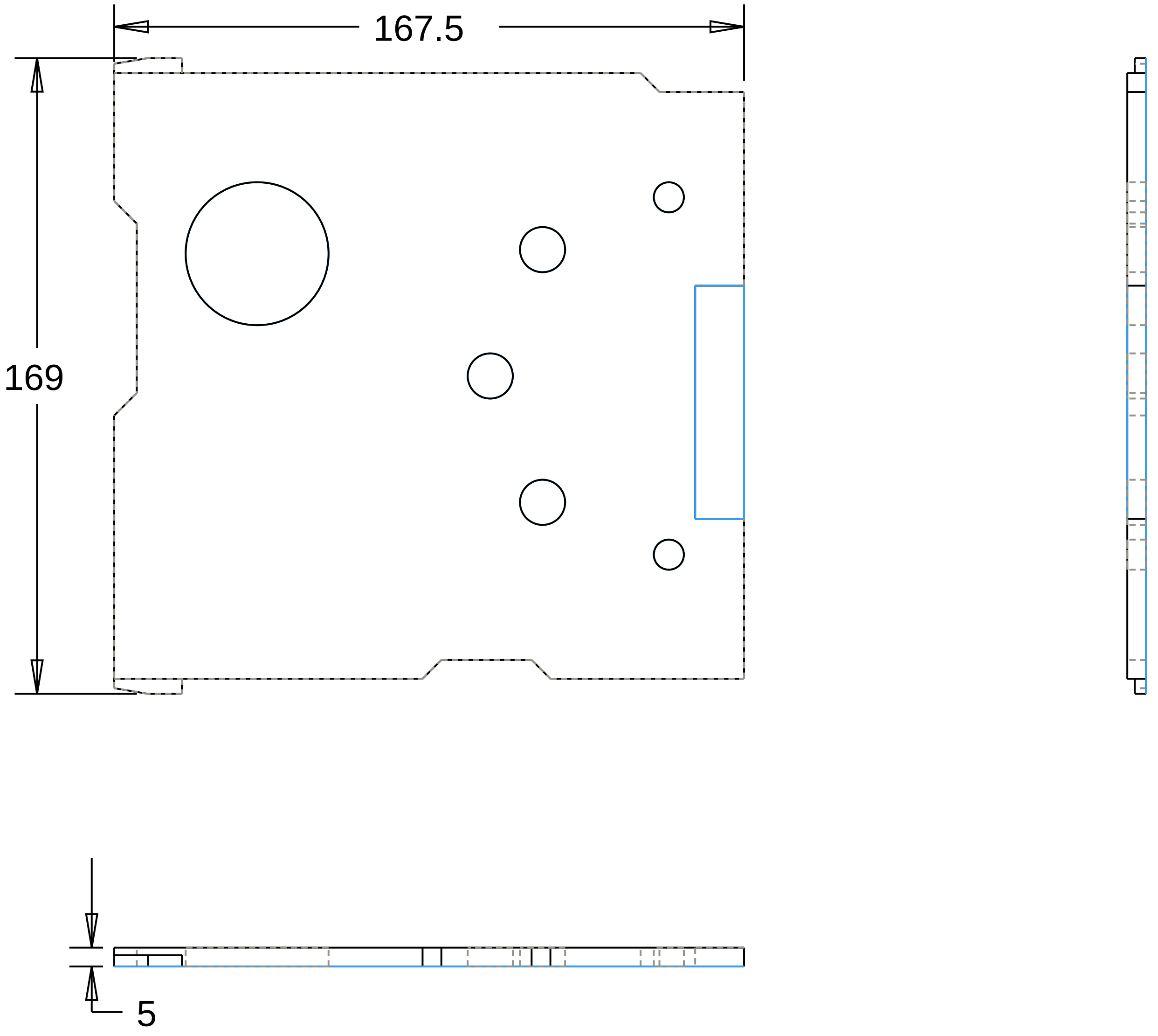
1:1



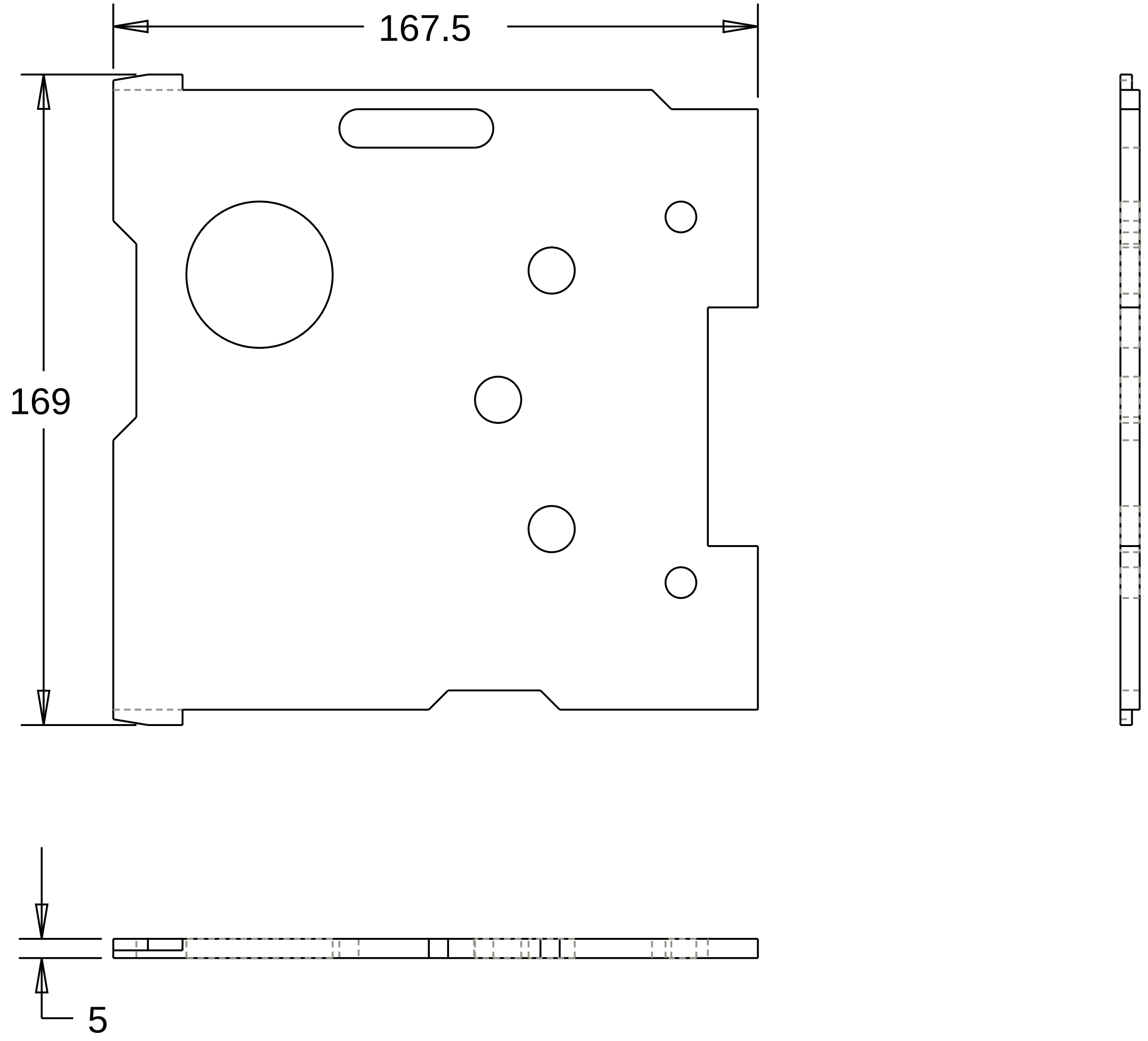
Location: Dublin City University	
Title: CHIP_HOLDER_TOP	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:2	



Location: Dublin City University	
Title: CHIP_HOLDER	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:2	



Location: Dublin City University	
Title: TOP_PLATE_LEFT	
Name: Conor Slater	Date: 13-7-2009
Scale: 1:2	



Location:

Dublin City University

Title:

TOP_PLATE_RIGHT

Name:

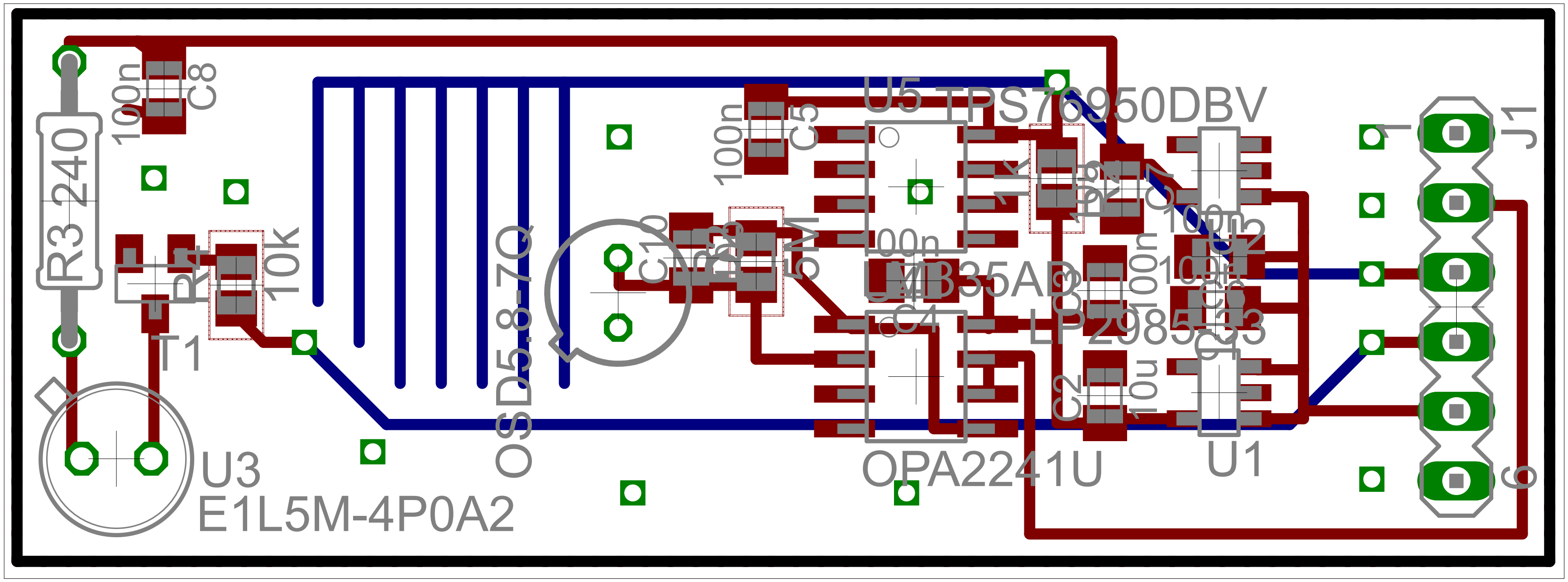
Conor Slater

Date:

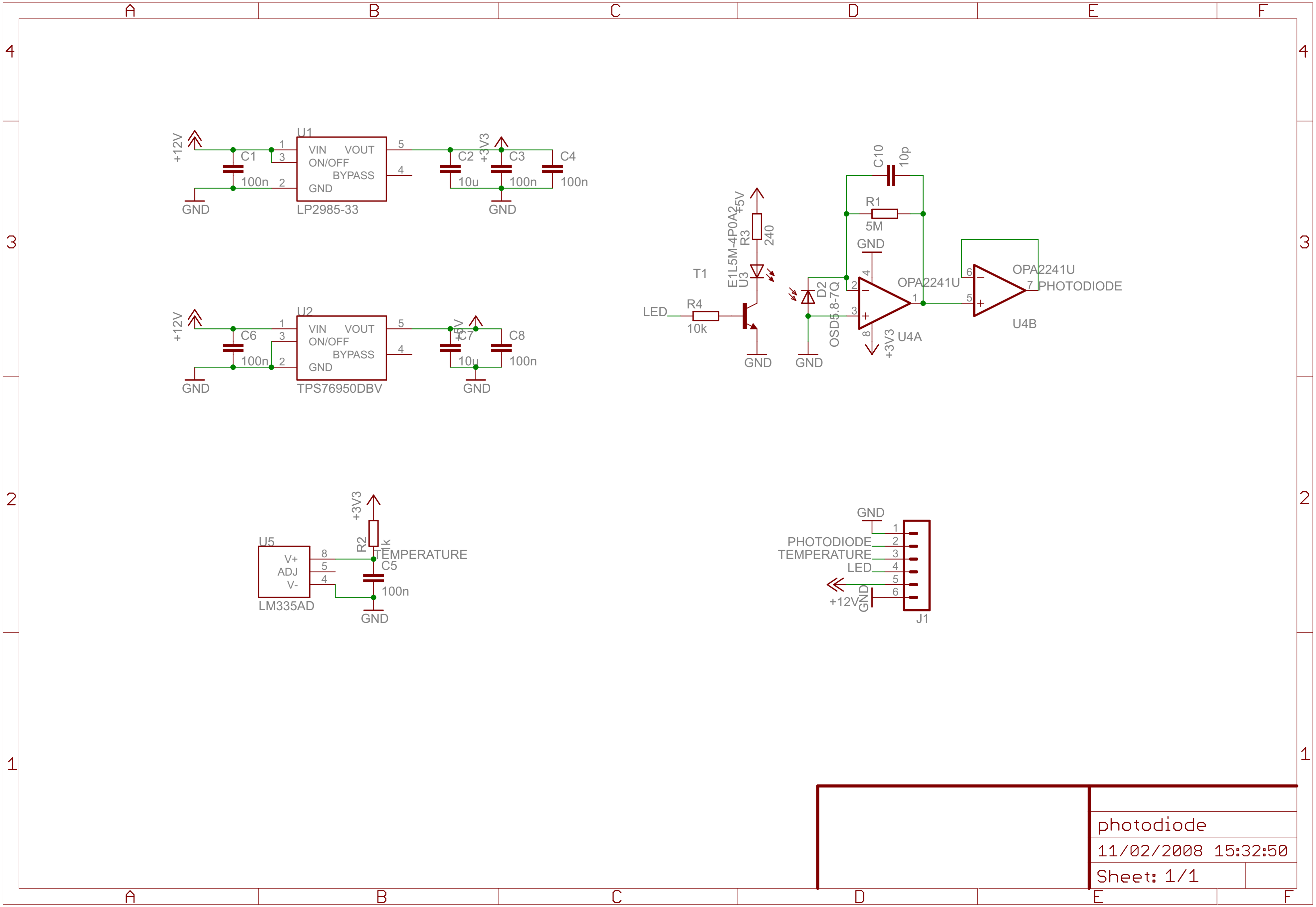
13-7-2009

Scale:

1:2

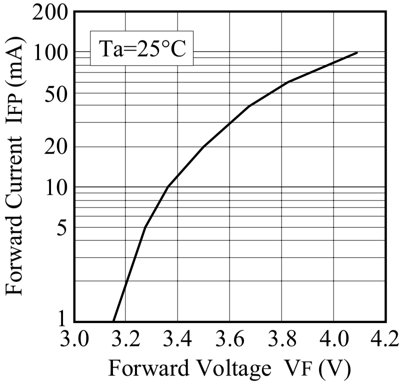


17/07/2009 18:24:19 f=4.00 C:\Documents and Settings\SlaterC\My Documents\Work\PCB\GSMPo4-II\photodiode.brd

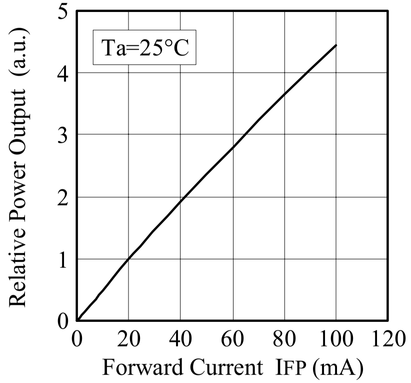


17/07/2009 18:23:25 C:\Documents and Settings\SlaterC\My Documents\Work\PCB\GSMPo4-II\photodiode.sch (Sheet: 1/1)

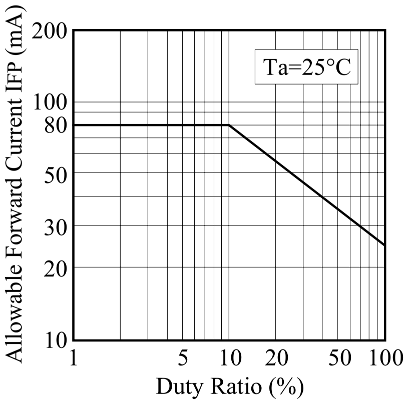
■ Forward Voltage vs.
Forward Current



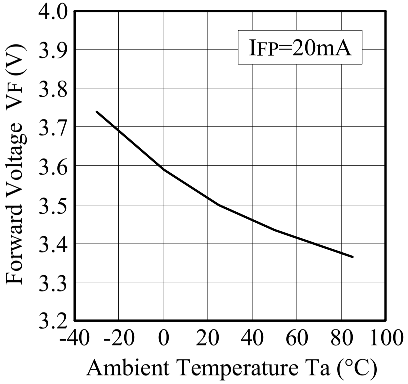
■ Forward Current vs.
Relative Power Output



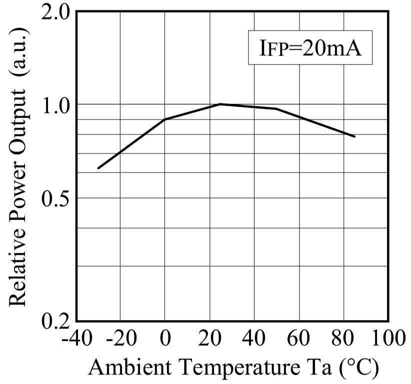
■ Duty Ratio vs.
Allowable Forward Current



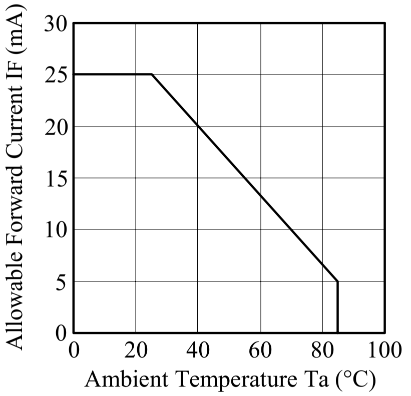
■ Ambient Temperature vs.
Forward Voltage



■ Ambient Temperature vs.
Relative Power Output

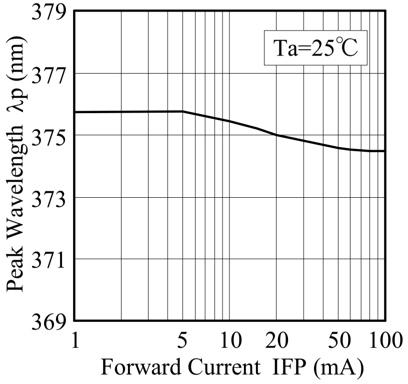


■ Ambient Temperature vs.
Allowable Forward Current

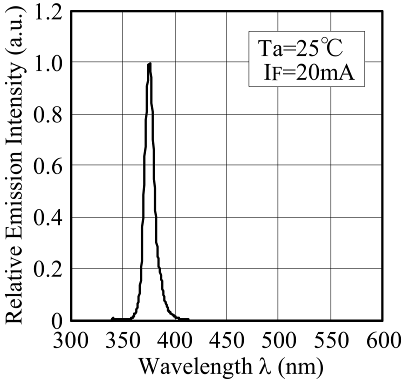


NICHIA CORPORATION	Model	NSHUxxxA
	Title	TYP.CHARACTERISTICS
	No.	040121421401

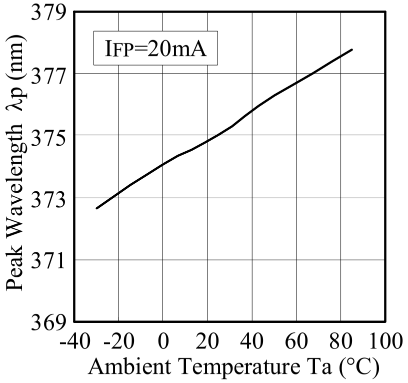
■ Forward Current vs.
Peak Wavelength



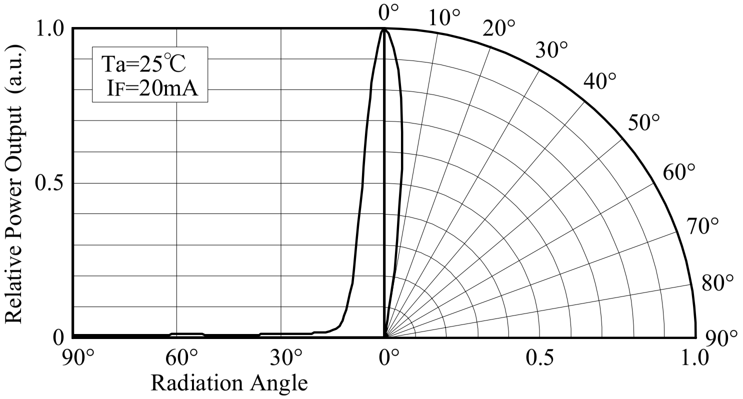
■ Spectrum



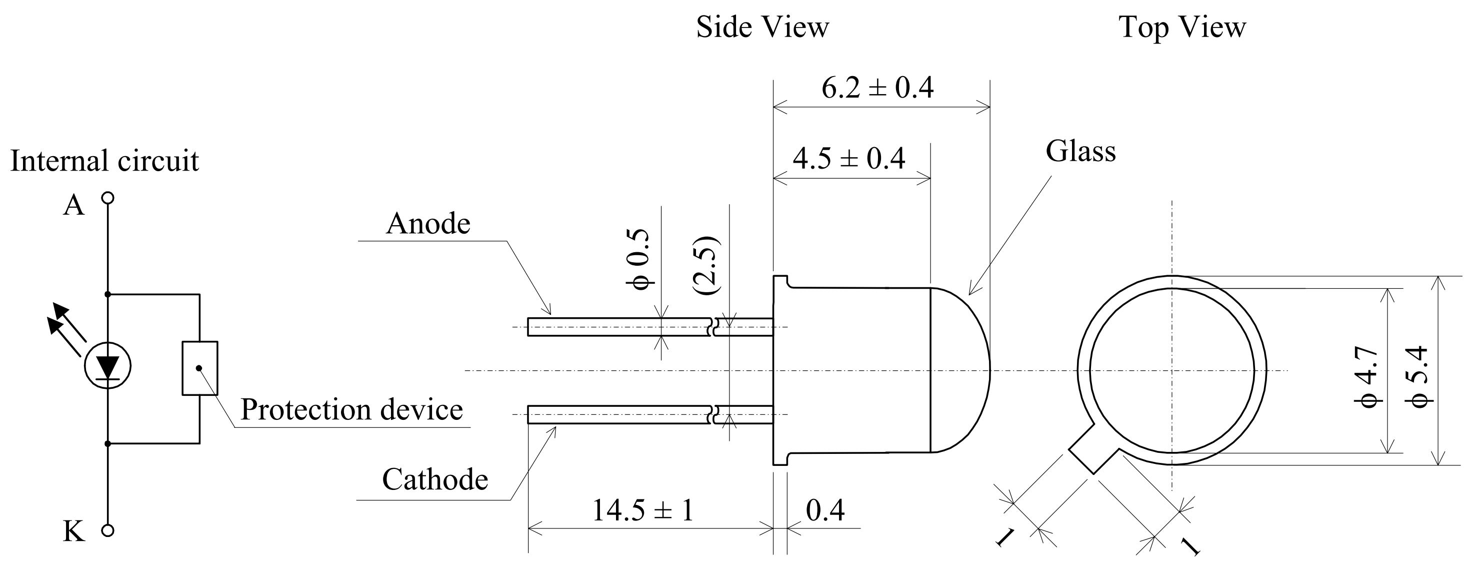
■ Ambient Temperature vs.
Peak Wavelength



■ Directivity (NSHU590A)



NICHIA CORPORATION	Model	NSHU590A
	Title	TYP.CHARACTERISTICS
	No.	031202200913



∗ NSHU590A has a protection device built in as a protection circuit against static electricity.

ITEM	MATERIALS
GLASS	Hard Glass
CAP	Ni Plating Iron Alloy
LEAD	Au Plating Iron Alloy

NICHIA CORPORATION

Model NSHU590A

Title OUTLINE DIMENSIONS

No. 020705200151

Unit mm

5/1 Scale

Allow ±0.2

GaP – UV-Photodiodes (190 - 570 nm) EPD-440-0

Schottky barrier type

Description

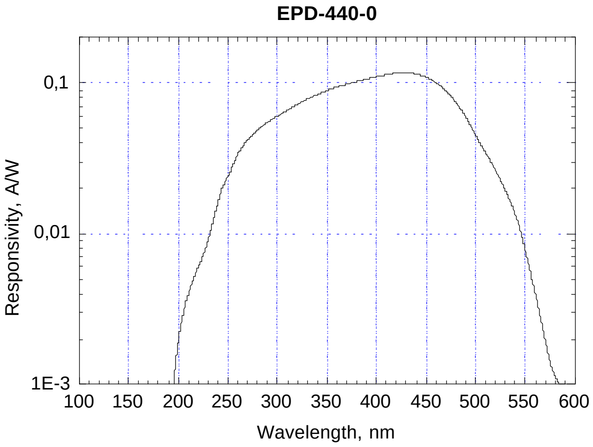
Wide bandwidth and high spectral sensitivity in the UV and visible range (190 nm - 570 nm), low cost chip based on GaP, large active areas are possible

Applications

Medical engineering (dermatology), output check of UV - lamps and gas burner flame, measurement and control of ecological parameters, radiation control for a solarium, UV water purification facilities

Features

Mounted in hermetically sealed TO-packages with UV glass window, different active areas are available



Parameter	Units	Symbol	EPD-440-0/0.9	EPD-440-0/1.4	EPD-440-0/2.5	EPD-440-0/3.6
Chip sizes	mm		0.9 x 0.9	1.4 x 1.4	2.5 x 2.5	3.6 x 3.6
Active area	mm²	A	0.7	1.2	4.8	10.9
Max. dark current at V_R = 0.01V V_R = 1V	pA	I_D	1 10	2 20	4 40	8 80
Package			TO-46	TO-46	TO-39	TO-39
Spectral range at 0,01 maximum	nm	λ_{min}-λ_{max}	190 – 570 (with UV-glass)			
Spectral bandwidth at 50%	nm	$\Delta\lambda_{0,5}$	180			
Peak sensitivity wavelength	nm	λ_p	440			
Typical responsivity at λ_p	A/W	$S_{ë}$	0.12			
Temperature coefficient of I_D	times/K	TCI_D	1.07			
Typical rise and fall time at V_R = -5V and 50 Ω load	ns	t_r t_f	0.7 13	0.8 30	1 140	1 140
Maximal reverse voltage at I_R=100 µA	V	V_R	10			
Operating temperature range	°C	T_{amb}	-40 to +125			
Storage temperature range	°C	T_{stg}	-40 to +125			

- **High Slew Rate . . . 10.5 V/µs Typ**
- **High-Gain Bandwidth . . . 5.1 MHz Typ**
- **Supply Voltage Range 2.5 V to 5.5 V**
- **Rail-to-Rail Output**
- **360 µV Input Offset Voltage**
- **Low Distortion Driving 600-Ω 0.005% THD+N**
- **1 mA Supply Current (Per Channel)**
- **17 nV/√Hz Input Noise Voltage**
- **2 pA Input Bias Current**
- **Characterized From T_A = –55°C to 125°C**
- **Available in MSOP and SOT-23 Packages**
- **Micropower Shutdown Mode . . . I_{DD} < 1 µA**
- **Available in Q-Temp Automotive**
  High Reliability Automotive Applications
  Configuration Control / Print Support
  Qualification to Automotive Standards

description

The TLV277x CMOS operational amplifier family combines high slew rate and bandwidth, rail-to-rail output swing, high output drive, and excellent dc precision. The device provides 10.5 V/µs of slew rate and 5.1 MHz of bandwidth while only consuming 1 mA of supply current per channel. This ac performance is much higher than current competitive CMOS amplifiers. The rail-to-rail output swing and high output drive make these devices a good choice for driving the analog input or reference of analog-to-digital converters. These devices also have low distortion while driving a 600-Ω load for use in telecom systems.

These amplifiers have a 360-µV input offset voltage, a 17 nV/√Hz input noise voltage, and a 2-pA input bias current for measurement, medical, and industrial applications. The TLV277x family is also specified across an extended temperature range (–40°C to 125°C), making it useful for automotive systems, and the military temperature range (–55°C to 125°C), for military systems.

These devices operate from a 2.5-V to 5.5-V single supply voltage and are characterized at 2.7 V and 5 V. The single-supply operation and low power consumption make these devices a good solution for portable applications. The following table lists the packages available.

FAMILY PACKAGE TABLE

DEVICE	NUMBER OF CHANNELS	PACKAGE TYPES								SHUTDOWN	UNIVERSAL EVM BOARD
		PDIP	CDIP	SOIC	SOT-23	TSSOP	MSOP	LCCC	CPAK		
TLV2770	1	8	—	8	—	—	8	—	—	Yes	Refer to the EVM Selection Guide (Lit# SLOU060)
TLV2771	1	—	—	8	5	—	—	—	—	—	
TLV2772	2	8	8	8	—	8	8	20	10	—	
TLV2773	2	14	—	14	—	—	10	—	—	Yes	
TLV2774	4	14	—	14	—	14	—	—	—	—	
TLV2775	4	16	—	16	—	16	—	—	—	Yes	

A SELECTION OF SINGLE-SUPPLY OPERATIONAL AMPLIFIER PRODUCTS†

DEVICE	V_{DD} (V)	BW (MHz)	SLEW RATE (V/µs)	I_{DD} (per channel) (µA)	RAIL-TO-RAIL
TLV277X	2.5 – 6.0	5.1	10.5	1000	O
TLV247X	2.7 – 6.0	2.8	1.5	600	I/O
TLV245X	2.7 – 6.0	0.22	0.11	23	I/O
TLV246X	2.7 – 6.0	6.4	1.6	550	I/O

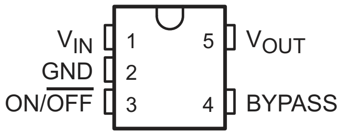
† All specifications measured at 5 V.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

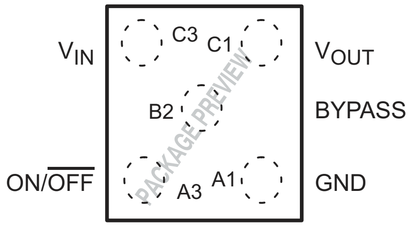
LP2985LV
150-mA LOW-NOISE, LOW-DROPOUT REGULATOR WITH SHUTDOWN FOR OUTPUT VOLTAGES <2.3 V

- **Available in the Texas Instruments NanoStar™ and NanoFree™ Wafer Chip Scale Packages**
- **Output Tolerance of:**
  − 1% (A Grade)
  − 1.5% (Standard Grade)
- **Ultra Low Dropout, Typically 280 mV at Full Load of 150 mA**
- **Wide V_{IN} Range . . . 16 V (Max)**
- **Low I_Q . . . 850 µA at Full Load at 150 mA**
- **Shutdown Current . . . 0.01 µA Typ**
- **Low Noise . . . 30 μV_{RMS} With 10-nF Bypass Capacitor**
- **Stable With Low ESR Capacitors, Including Ceramic**
- **Over-Current and Thermal Protection**
- **High Peak Current Capability**
- **For V_{OUT} Options ≥2.5 V, See LP2985 Data Sheet**
- **Portable Applications**
  − Cellular Phones
  − Palmtop and Laptop Computers
  − Personal Digital Assistants (PDAs)
  − Digital Cameras and Camcorders
  − CD Players
  − MP3 Players

DBV (SOT-23) PACKAGE
(TOP VIEW)



YEQ, YEU, YZQ, OR YZU (WCSP) PACKAGE
(TOP VIEW)



description/ordering information

The LP2985LV family of fixed-output, low-dropout regulators offers exceptional, cost-effective performance for both portable and nonportable applications. Available in voltages of 1.25 V, 1.35 V, 1.5 V, 1.7 V, 1.8 V, and 2 V, the family has an output tolerance of 1% for the A version (1.5% for the non-A version), and is capable of delivering 150-mA continuous load current. Standard regulator features, such as over-current and over-temperature protection, are included.

The LP2985LV has a host of features that makes the regulator an ideal candidate for a variety of portable applications:

- Low dropout: A PNP pass element allows a typical dropout of 280 mV at 150-mA load current.
- Low quiescent current: The use of a vertical PNP process allows for quiescent currents that are considerably lower than those associated with traditional lateral PNP regulators.
- Shutdown: A shutdown feature is available, allowing the regulator to consume only 0.01 µA when the $\overline{\text{ON/OFF}}$ pin is pulled low.
- Low-ESR-capacitor friendly: The regulator is stable with low ESR capacitors, allowing for the use of small, inexpensive ceramic capacitors in cost-sensitive applications.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

NanoStar and NanoFree are trademarks of Texas Instruments.

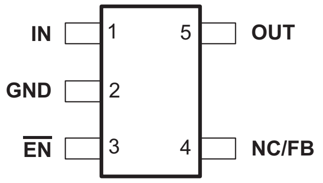
PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TPS76901, TPS76912, TPS76915, TPS76918, TPS76925
TPS76927, TPS76928, TPS76930, TPS76933, TPS76950
ULTRALOW-POWER 100-mA LOW-DROPOUT LINEAR REGULATORS

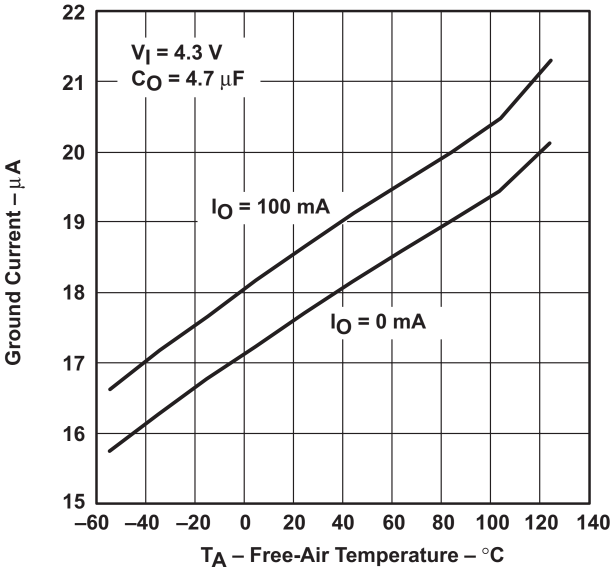
SLVS203D – JUNE 1999 – REVISED APRIL 2000

- **100-mA Low-Dropout Regulator**
- **Available in 1.2-V, 1.5-V, 1.8-V, 2.5-V, 2.7-V, 2.8-V, 3.0-V, 3.3-V, and 5-V Fixed-Output and Adjustable Versions**
- **Only 17 μA Quiescent Current at 100 mA**
- **1 μA Quiescent Current in Standby Mode**
- **Dropout Voltage Typically 71 mV at 100mA**
- **Over Current Limitation**
- **–40°C to 125°C Operating Junction Temperature Range**
- **5-Pin SOT-23 (DBV) Package**

DBV PACKAGE
(TOP VIEW)



TPS76933
GROUND CURRENT
vs
FREE-AIR TEMPERATURE



description

The TPS769xx family of low-dropout (LDO) voltage regulators offers the benefits of low dropout voltage, ultralow-power operation, and miniaturized packaging. These regulators feature low dropout voltages and ultralow quiescent current compared to conventional LDO regulators. Offered in a 5-terminal small outline integrated-circuit SOT-23 package, the TPS769xx series devices are ideal for micropower operations and where board space is at a premium.

A combination of new circuit design and process innovation has enabled the usual PNP pass transistor to be replaced by a PMOS pass element. Because the PMOS pass element behaves as a low-value resistor, the dropout voltage is very low, typically 71 mV at 100 mA of load current (TPS76950), and is directly proportional to the load current. Since the PMOS pass element is a voltage-driven device, the quiescent current is ultralow (28 μA maximum) and is stable over the entire range of output load current (0 mA to 100 mA). Intended for use in portable systems such as laptops and cellular phones, the ultralow-dropout voltage feature and ultralow-power operation result in a significant increase in system battery operating life.

The TPS769xx also features a logic-enabled sleep mode to shut down the regulator, reducing quiescent current to 1 μA typical at T_J = 25°C. The TPS769xx is offered in 1.2-V, 1.5-V, 1.8-V, 2.5-V, 2.7-V, 2.8-V, 3.0-V, 3.3-V, and 5-V fixed-voltage versions and in a variable version (programmable over the range of 1.2 V to 5.5 V).

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

National Semiconductor

February 1995

LM135/LM235/LM335, LM135A/LM235A/LM335A Precision Temperature Sensors

General Description

The LM135 series are precision, easily-calibrated, integrated circuit temperature sensors. Operating as a 2-terminal zener, the LM135 has a breakdown voltage directly proportional to absolute temperature at +10 mV/°K. With less than 1Ω dynamic impedance the device operates over a current range of 400 μA to 5 mA with virtually no change in performance. When calibrated at 25°C the LM135 has typically less than 1°C error over a 100°C temperature range. Unlike other sensors the LM135 has a linear output.

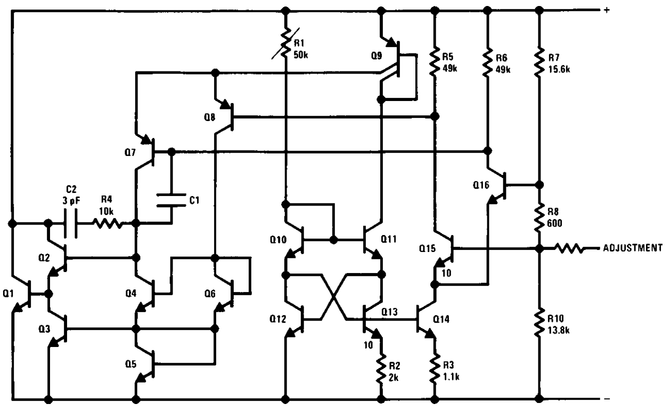
Applications for the LM135 include almost any type of temperature sensing over a −55°C to +150°C temperature range. The low impedance and linear output make interfacing to readout or control circuitry especially easy.

The LM135 operates over a −55°C to +150°C temperature range while the LM235 operates over a −40°C to +125°C temperature range. The LM335 operates from −40°C to +100°C. The LM135/LM235/LM335 are available packaged in hermetic TO-46 transistor packages while the LM335 is also available in plastic TO-92 packages.

Features

- Directly calibrated in °Kelvin
- 1°C initial accuracy available
- Operates from 400 μA to 5 mA
- Less than 1Ω dynamic impedance
- Easily calibrated
- Wide operating temperature range
- 200°C overrange
- Low cost

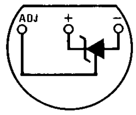
Schematic Diagram



TL/H/5698-1

Connection Diagrams

TO-92
Plastic Package

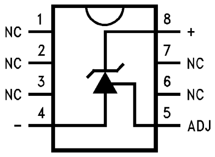


TL/H/5698-8

Bottom View

Order Number LM335Z or LM335AZ
See NS Package Number Z03A

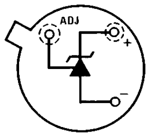
SO-8
Surface Mount Package



TL/H/5698-25

Order Number LM335M or LM335AM
See NS Package Number M08A

TO-46
Metal Can Package*



TL/H/5698-26

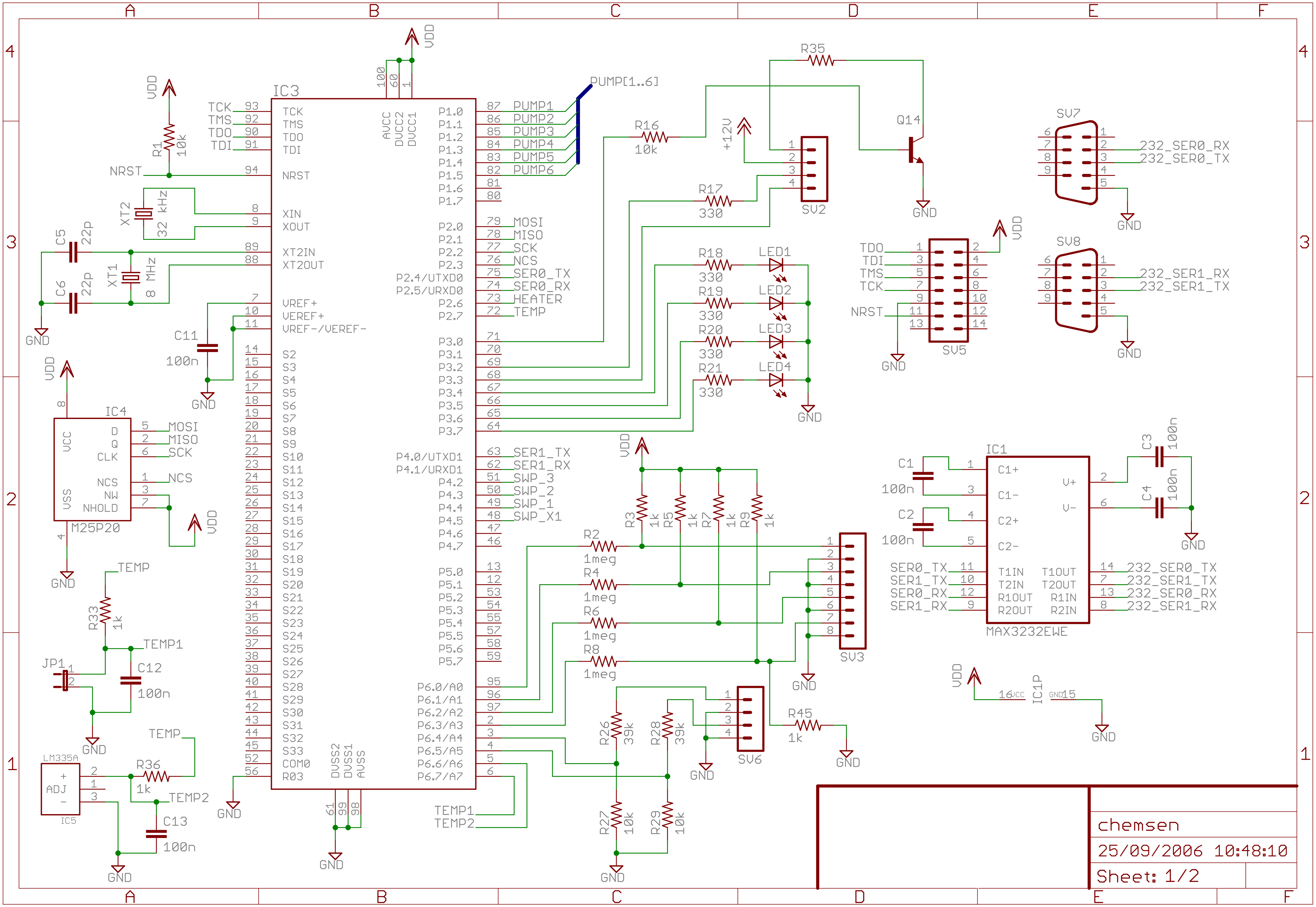
Bottom View

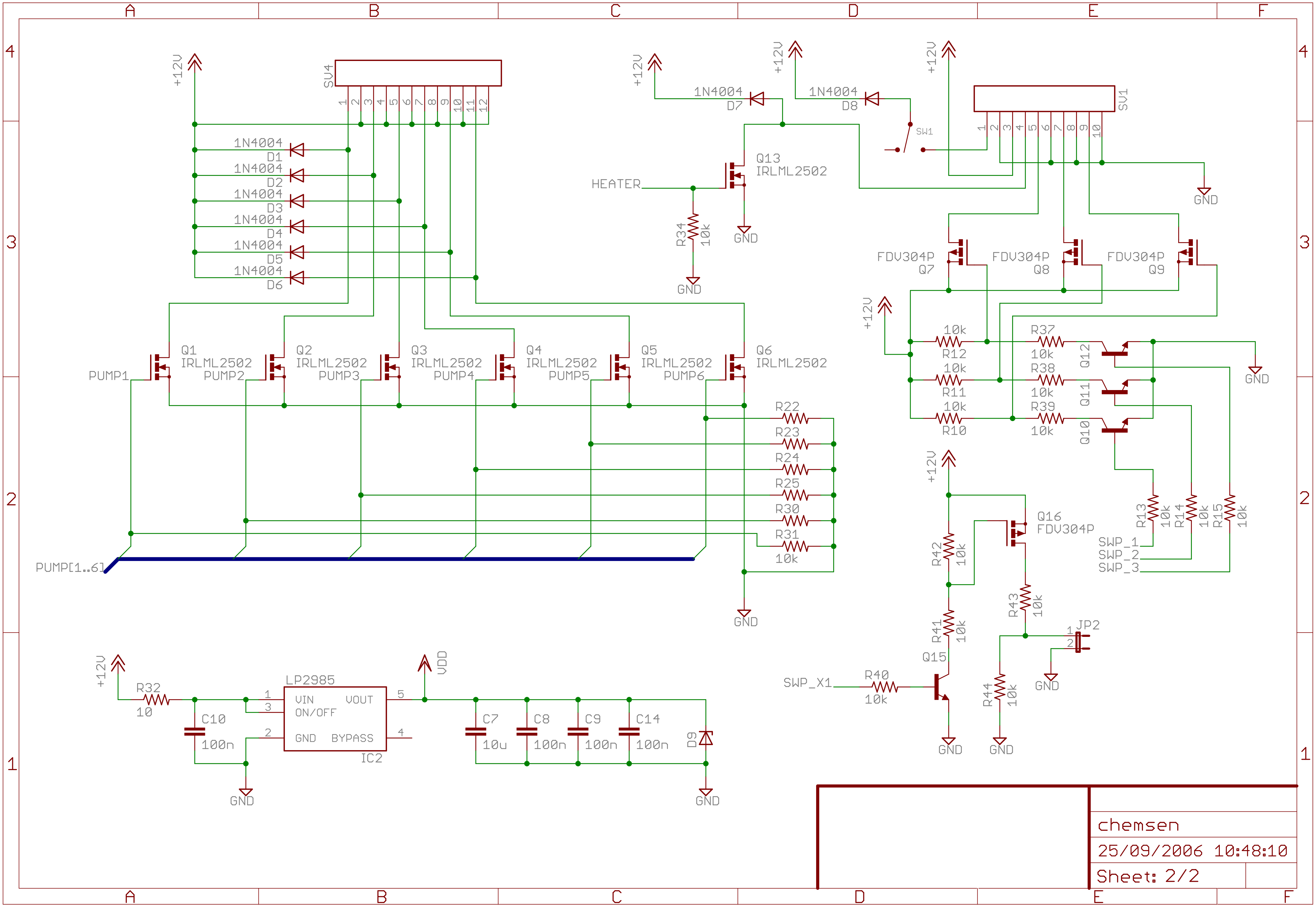
*Case is connected to negative pin

Order Number LM135H, LM135H-MIL, LM235H, LM335H, LM135AH, LM235AH or LM335AH
See NS Package Number H03H

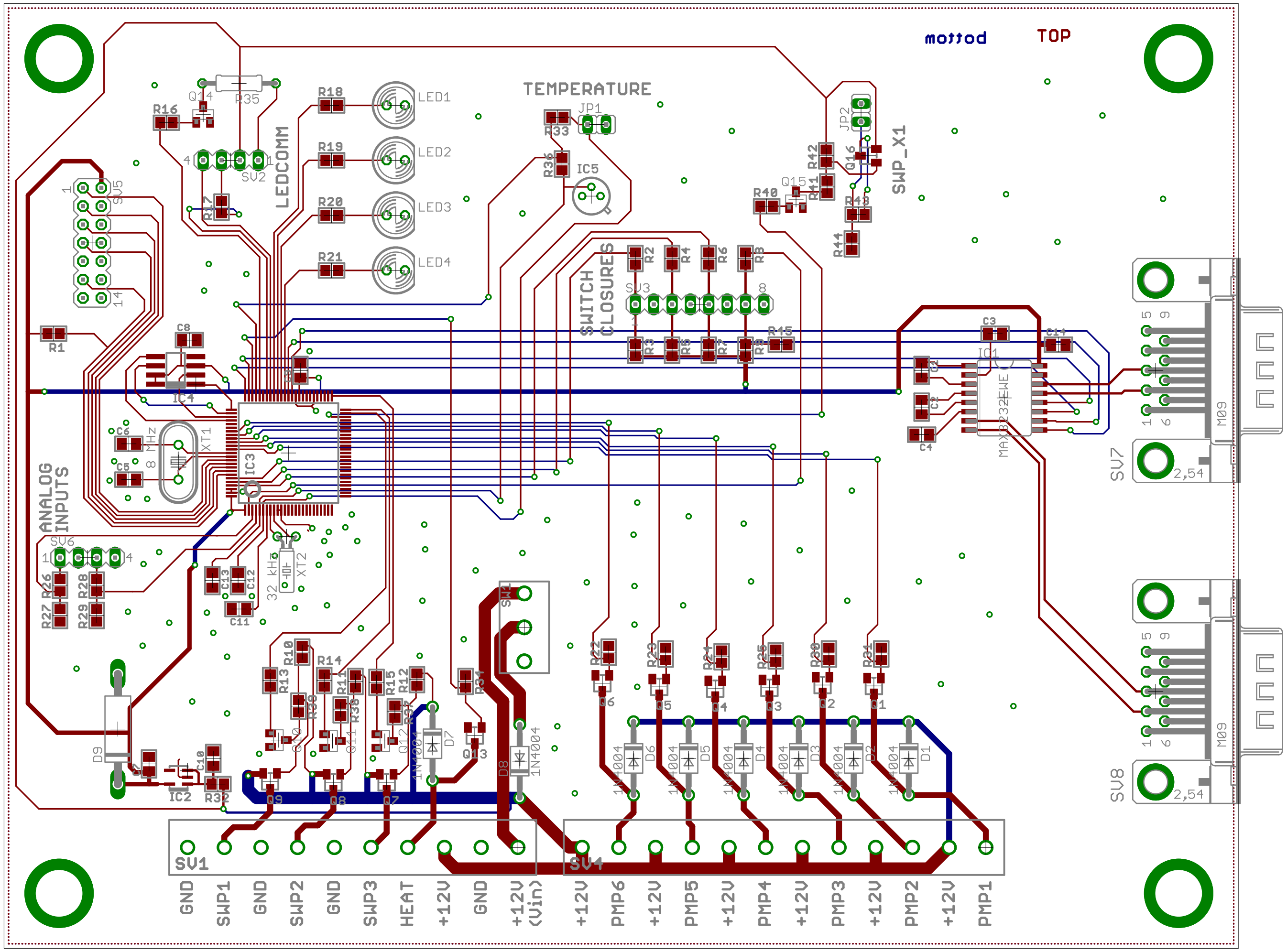
LM135/LM235/LM335, LM135A/LM235A/LM335A Precision Temperature Sensors

RRD-B30M115/Printed in U. S. A.

A
B
C
D
E
F
4
3
2
1
VDD
IC3
100
60
1
AVCC
DVCC2
DVCC1
TCK 93
TMS 92
TDO 90
TDI 91
NRST 94
XIN 8
XOUT 9
XT2IN 89
XT2OUT 88
VREF+ 7
VEREF+ 10
VREF-/VEREF- 11
S2 14
S3 15
S4 16
S5 17
S6 18
S7 19
S8 20
S9 21
S10 22
S11 23
S12 24
S13 25
S14 26
S15 27
S16 28
S17 29
S18 30
S19 31
S20 32
S21 33
S22 34
S23 35
S24 36
S25 37
S26 38
S27 39
S28 40
S29 41
S30 42
S31 43
S32 44
S33 45
COM0 52
R03 56
DVSS2
DVSS1
AVSS
61
99
98
P1.0 87 PUMP1
P1.1 86 PUMP2
P1.2 85 PUMP3
P1.3 84 PUMP4
P1.4 83 PUMP5
P1.5 82 PUMP6
P1.6 81
P1.7 80
P2.0 79 MOSI
P2.1 78 MISO
P2.2 77 SCK
P2.3 76 NCS
P2.4/UTXD0 75 SER0_TX
P2.5/URXD0 74 SER0_RX
P2.6 73 HEATER
P2.7 72 TEMP
P3.0 71
P3.1 70
P3.2 69
P3.3 68
P3.4 67
P3.5 66
P3.6 65
P3.7 64
P4.0/UTXD1 63 SER1_TX
P4.1/URXD1 62 SER1_RX
P4.2 51 SWP_3
P4.3 50 SWP_2
P4.4 49 SWP_1
P4.5 48 SWP_X1
P4.6 47
P4.7 46
P5.0 13
P5.1 12
P5.2 53
P5.3 54
P5.4 55
P5.5 57
P5.6 58
P5.7 59
P6.0/A0 95
P6.1/A1 96
P6.2/A2 97
P6.3/A3 2
P6.4/A4 3
P6.5/A5 4
P6.6/A6 5
P6.7/A7 6
TEMP1
TEMP2
PUMP[1..6]
R1 10k
XT2 32 kHz
C5 22p
C6 22p
XT1 8 MHz
C11 100n
GND
IC4
VCC
VSS
D 5 MOSI
Q 2 MISO
CLK 6 SCK
NCS 1 NCS
NW 3
NHOLD 7
8
4
M25P20
TEMP
R33 1k
TEMP1
JP1
C12 100n
TEMP
R36 1k
TEMP2
LM335A
ADJ
IC5
C13 100n
R16 10k
+12V
R17 330
SV2
R35
Q14
R18 330
R19 330
R20 330
R21 330
LED1
LED2
LED3
LED4
SV7
232_SER0_RX
232_SER0_TX
SV5
TDO
TDI
TMS
TCK
NRST
SV8
232_SER1_RX
232_SER1_TX
R3 1k
R5 1k
R7 1k
R9 1k
R2 1meg
R4 1meg
R6 1meg
R8 1meg
SV3
R26 39k
R28 39k
R27 10k
R29 10k
SV6
R45 1k
IC1
C1 100n
C2 100n
C3 100n
C4 100n
C1+
C1-
C2+
C2-
V+
V-
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
SER0_TX 11
SER1_TX 10
SER0_RX 12
SER1_RX 9
14 232_SER0_TX
7 232_SER1_TX
13 232_SER0_RX
8 232_SER1_RX
MAX3232EWE
16 VCC
GND 15
IC1P
chemsen
25/09/2006 10:48:10
Sheet: 1/2



17/07/2009 18:21:07 C:\Documents and Settings\SlaterC\My Documents\Work\PCB\GSMPo4\chemsen.sch (Sheet: 2/2)



17/07/2009 18:21:36 C:\Documents and Settings\SlaterC\My Documents\Work\PCB\GSMPo4\chemsen.brd

- **Low Supply-Voltage Range, 1.8 V to 3.6 V**
- **Ultralow-Power Consumption:**
  - **Active Mode: 280 µA at 1 MHz, 2.2 V**
  - **Standby Mode: 1.1 µA**
  - **Off Mode (RAM Retention): 0.1 µA**
- **Five Power Saving Modes**
- **Wake-Up From Standby Mode in 6 µs**
- **16-Bit RISC Architecture, 125-ns Instruction Cycle Time**
- **12-Bit A/D Converter With Internal Reference, Sample-and-Hold and Autoscan Feature**
- **16-Bit Timer With Three† or Seven‡ Capture/Compare-With-Shadow Registers, Timer_B**
- **16-Bit Timer With Three Capture/Compare Registers, Timer_A**
- **On-Chip Comparator**
- **Serial Communication Interface (USART), Select Asynchronous UART or Synchronous SPI by Software; Two USARTs (USART0, USART1) In MSP430x44x Devices One USART (USART0) In MSP430x43x Devices**
- **Brownout Detector**
- **Supply Voltage Supervisor/Monitor With Programmable Level Detection**
- **Serial Onboard Programming, No External Programming Voltage Needed Programmable Code Protection by Security Fuse**
- **Integrated LCD Driver for Up to 160 Segments**
- **Family Members Include:**
  - **MSP430F435:**
    16KB+256B Flash Memory,
    512B RAM
  - **MSP430F436:**
    24KB+256B Flash Memory,
    1KB RAM
  - **MSP430F437:**
    32KB+256B Flash Memory,
    1KB RAM
  - **MSP430F447:**
    32KB+256B Flash Memory,
    1KB RAM
  - **MSP430F448:**
    48KB+256B Flash Memory,
    2KB RAM
  - **MSP430F449:**
    60KB+256B Flash Memory,
    2KB RAM
- **For Complete Module Descriptions, See The MSP430x4xx Family User's Guide, Literature Number SLAU056**

† 'F435, 'F436, and 'F437 devices
‡ 'F447, 'F448, and 'F449 devices

description

The Texas Instruments MSP430 series is an ultralow-power microcontroller family consisting of several devices featuring different sets of modules targeted to various applications. The microcontroller is designed to be battery operated for use in extended-time applications. The MSP430 achieves maximum code efficiency with its 16-bit RISC architecture, 16-bit CPU-integrated registers, and a constant generator. The digitally-controlled oscillator provides wake-up from low-power mode to active mode in less than 6 µs. The MSP430x43x and the MSP430x44x series are microcontroller configurations with two built-in 16-bit timers, a fast 12-bit A/D converter, one or two universal serial synchronous/asynchronous communication interfaces (USART), 48 I/O pins, and a liquid crystal driver (LCD) with up to 160 segments.

Typical applications include sensor systems that capture analog signals, convert them to digital values, and process and transmit the data to a host system, or process this data and displays it on a LCD panel. The timers make the configurations ideal for industrial control applications such as ripple counters, digital motor control, EE-meters, hand-held meters, etc. The hardware multiplier enhances the performance and offers a broad code and hardware-compatible family solution.

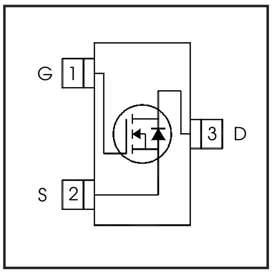
Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

International
IOR Rectifier

IRLML2502

HEXFET® Power MOSFET

- Ultra Low On-Resistance
- N-Channel MOSFET
- SOT-23 Footprint
- Low Profile (<1.1mm)
- Available in Tape and Reel
- Fast Switching



V_{DSS} = 20V

$R_{DS(on)}$ = 0.045Ω

Micro3™

Description

These N-Channel MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET® power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in battery and load management.

A thermally enhanced large pad leadframe has been incorporated into the standard SOT-23 package to produce a HEXFET Power MOSFET with the industry's smallest footprint. This package, dubbed the Micro3™, is ideal for applications where printed circuit board space is at a premium. The low profile (<1.1mm) of the Micro3 allows it to fit easily into extremely thin application environments such as portable electronics and PCMCIA cards. The thermal resistance and power dissipation are the best available.

Absolute Maximum Ratings

	Parameter	Max.	Units
V_{DS}	Drain- Source Voltage	20	V
I_D @ T_A = 25°C	Continuous Drain Current, V_{GS} @ 4.5V	4.2	A
I_D @ T_A= 70°C	Continuous Drain Current, V_{GS} @ 4.5V	3.4	
I_{DM}	Pulsed Drain Current ①	33	
P_D @T_A = 25°C	Power Dissipation	1.25	W
P_D @T_A = 70°C	Power Dissipation	0.8	
	Linear Derating Factor	0.01	W/°C
V_{GS}	Gate-to-Source Voltage	± 12	V
T_J, T_{STG}	Junction and Storage Temperature Range	-55 to + 150	°C

Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JA}$	Maximum Junction-to-Ambient③	75	100	°C/W

04/30/03

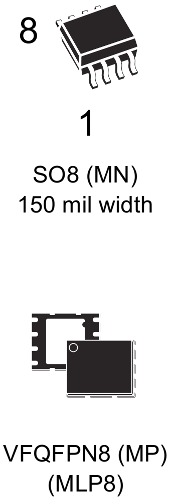
M25P20

2 Mbit, Low Voltage, Serial Flash Memory With 25 MHz SPI Bus Interface

FEATURES SUMMARY

- 2 Mbit of Flash Memory
- Page Program (up to 256 Bytes) in 1.5ms (typical)
- Sector Erase (512 Kbit) in 2 s (typical)
- Bulk Erase (2 Mbit) in 3 s (typical)
- 2.7 V to 3.6 V Single Supply Voltage
- SPI Bus Compatible Serial Interface
- 25 MHz Clock Rate (maximum)
- Deep Power-down Mode 1 µA (typical)
- Electronic Signature (11h)
- More than 100,000 Erase/Program Cycles per Sector
- More than 20 Year Data Retention

Figure 1. Packages



SO8 (MN)
150 mil width

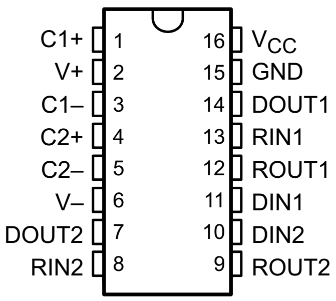
VFQFPN8 (MP)
(MLP8)

MAX3232
3-V TO 5.5-V MULTICHANNEL RS-232 LINE DRIVER/RECEIVER

SLLS410F – JANUARY 2000 – REVISED AUGUST 2002

- **Meets or Exceeds the Requirements of TIA/EIA-232-F and ITU v.28 Standards**
- **Operates With 3-V to 5.5-V V_{CC} Supply**
- **Operates up to 250 kbit/s**
- **Two Drivers and Two Receivers**
- **Low Supply Current . . . 300 μA Typical**
- **External Capacitors . . . 4 × 0.1 μF**
- **Accepts 5-V Logic Input With 3.3-V Supply**
- **Designed to Be Interchangeable With Maxim MAX3232**
- **RS-232 Bus-Pin ESD Protection Exceeds ±15 kV Using Human-Body Model (HBM)**
- **Applications**
  - **Battery-Powered Systems, PDAs, Notebooks, Laptops, Palmtop PCs, and Hand-Held Equipment**

D, DB, DW, OR PW PACKAGE
(TOP VIEW)



description/ordering information

The MAX3232 device consists of two line drivers, two line receivers, and a dual charge-pump circuit with ±15-kV ESD protection pin to pin (serial-port connection pins, including GND). The device meets the requirements of TIA/EIA-232-F and provides the electrical interface between an asynchronous communication controller and the serial-port connector. The charge pump and four small external capacitors allow operation from a single 3-V to 5.5-V supply. The devices operate at data signaling rates up to 250 kbit/s and a maximum of 30-V/μs driver output slew rate.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–0°C to 70°C	SOIC – D	Tube	MAX3232CD	MAX3232C
		Tape and reel	MAX3232CDR	
	SOIC – DW	Tube	MAX3232CDW	MAX3232C
		Tape and reel	MAX3232CDWR	
	SSOP – DB	Tape and reel	MAX3232CDBR	MA3232C
	TSSOP – PW	Tape and reel	MAX3232CPWR	MA3232C
–40°C to 85°C	SOIC – D	Tube	MAX3232ID	MAX3232I
		Tape and reel	MAX3232IDR	
	SOIC – DW	Tube	MAX3232IDW	MAX3232I
		Tape and reel	MAX3232IDWR	
	SSOP – DB	Tape and reel	MAX3232IDBR	MB3232I
	TSSOP – PW	Tape and reel	MAX3232IPWR	MB3232I

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

```
#include "adc12.h"

void adc_set_channel(volatile unsigned int ch) //selects and configures
channel
 {
  ADC12CTL0 =               //ADC12SC |         //Start Conversion
                                        //ENC |                   //Enable
                                        Conversion
                                        //ADC12TOVIE |  //Set Timer Flow
                                        Interupt Enable
                                        //ADC12OVIE |   //Set Overflow
                                        Interupt Enable
                                        ADC12ON |       //Enable ADC Core
((ch==10 || ch==1) ?    REFON : 0) |    //If channel 10 or 1 ref on
((ch==10 || ch==1) ?    REF2_5V : 0) |//If channel 10 or 1 ref is 2.5V
                                        //MSC |         //Set Multiple
                                        Sample Conversion
                                        SHT0_7 |            //Sample timer0
                                        is 1024
                                        SHT1_7;             //Sample timer1
                                        is 1024

  ADC12CTL1 =               //ADC12BUSY |   //Set ADC Busy
                                        CONSEQ_0 |      //Single Channel
                                        Conversion
                            ADC12SSEL_2 | //Master Clock
                            ADC12DIV_7 |    //Clock Divide
                            //ISSH |                //Invert Samplehold
                            Signal
                            //SHP |                 //Set Sample Hold Pulse
                            Mode
                            SHS_0;              //Sample hold source
                            ADC12SC bit
                            //(ch<<12);         //Memory Select

  ADC12MCTL0 =              ch |
 ((ch==10 || ch==1) ? SREF_1 : SREF_0) | //If channel is 10 or 4 then ref
 2.5V
                                        EOS;
 }

void adc_off()
 {
  ADC12CTL0 = ADC12CTL1 = 0x0000;
 }

volatile unsigned int adc_read() //Take Sample
 {

    SET_BITS(ADC12CTL0, ADC12ON); //Enable ADC Core
    SET_BITS(ADC12CTL0, ENC); //Enable Conversion

    SET_BITS(ADC12CTL0, ADC12SC); //Start Conversion
    busy_wait(0x0fff); //Hold Sample
    CLR_BITS(ADC12CTL0, ADC12SC); //Stop Conversion

    while(ADC12IFG == 0){nop();} //Wait for flag

        CLR_BITS(ADC12CTL0, ENC); //Disable Conversion
        CLR_BITS(ADC12CTL0, ADC12ON); //Disable ADC Core

        return ADC12MEM0;

 }
```

```
#ifndef _ADC12_H
#define _ADC12_H

#include "utility.h"
#include "pin_defs.h"

/* routines for manipulating the ADCs */

/*** forward references ***/

void adc_set_channel(volatile unsigned int ch); //Also switches ADC on
void adc_off();
volatile unsigned int adc_read();

#endif /* _ADC12_H */
```

```
#include "batterymonitor.h"

volatile short get_battery()
 {

    volatile unsigned int i;
    unsigned long adc = 0;

    adc_set_channel(4);

    for(i=0;i<128;i++)
     {
      adc += adc_read();
     }

    adc_off();

    adc = adc/i;

    return adc;

 }
```

```
#ifndef _BATTERYMONITOR_H
#define _BATTERYMONITOR_H

#include "adc12.h"

/* routines for manipulating battery */

/*** forward references ***/

volatile short get_battery();

#endif /* _BATTERYMONITOR_H */
```

```
#include <string.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <ctype.h>

#include "pumps.h"
#include "temp_sensor.h"
#include "batterymonitor.h"
#include "photodiode.h"
#include "phosphate.h"
#include "spi_flash.h"
#include "uart.h"
#include "gsm.h"
#include "system.h"
#include "command.h"

char * delims = "$, \r";

void cmd_echo(void)
 {
    if(u0rx_echo == 0)
     {
      u0rx_echo = 1;
    printf0("$ECHO,ON\r\n");
     }
    else
     {
      u0rx_echo = 0;
    printf0("$ECHO,OFF\r\n");
     }
  return;
 }

void cmd_poset(char * cmd)
 {
    cmd = strtok(NULL, delims);
    if(cmd == NULL)
     {
        printf0("$POSET,20%02u%02u%02u", system.year, system.month, system.
        day);
    printf0(",%02u%02u\r\n", system.hour, system.minute);
    return;
     }
    else if(isdigit(*cmd) && (strlen(cmd) == 8) )
     {

        char cyear[] = {cmd[2],cmd[3]};
        system.year = atoi(&cyear[0]);

        char cmonth[] = {cmd[4],cmd[5]};
        system.month = atoi(&cmonth[0]);

        char cday[] = {cmd[6],cmd[7]};
        system.day = atoi(&cday[0]);

        cmd = strtok(NULL, delims);
        if( isdigit(*cmd) && (strlen(cmd) == 4))
         {
            char chour[] = {cmd[0],cmd[1]};
            system.hour = atoi(&chour[0]);

            char cminute[] = {cmd[2],cmd[3]};
            system.minute = atoi(&cminute[0]);
         }
        else
         {
            printf0("$POSET,ERROR\r\n");
            return;
         }
        printf0("$POSET,20%02u%02u%02u,%02u%02u,OK\r\n",
        system.year,system.month,system.day,system.hour,system.minute);
     }
    else
     {
        printf0("$POSET,ERROR\r\n");
        return;
     }

 }

void cmd_poprm(char * cmd)
{
    //volatile unsigned char poprm[0x000F];
    cmd = strtok(NULL, delims);
    char buffer[3];

    buffer[2] = '\0';

    if(cmd == '\0')
    {
        printf0("$POPRM,%02u%02u,%u,%u\r\n",system.sample_hour,system.
        sample_minute,system.sample_time,system.reaction_time);
        return;
    }

    // Set Sample Time
    buffer[0] = *cmd;               buffer[1] = *(++cmd);
    system.sample_hour = atoi(&buffer[0]);
    buffer[0] = *(++cmd);               buffer[1] = *(++cmd);
    system.sample_minute = atoi(&buffer[0]);

    // Set Sample Rate
    cmd = strtok(NULL, delims); //if(cmd == '\0')return;
    system.sample_time = atoi(cmd);

    // Set Reaction Time
    cmd = strtok(NULL, delims); //if(cmd == '\0')return;
    system.reaction_time = atoi(cmd);

  printf0("$POPRM,%02u%02u,%u,%u,OK\r\n",system.sample_hour,system.
    sample_minute,system.sample_time,system.reaction_time);
  return;
 }

void cmd_popre(char * cmd)
 {
    volatile unsigned char i, j, k=0, pumps, pump_number, pump_times;
    cmd = strtok(NULL, delims);
    if(cmd == '\0')
     {
    printf0("$POPRE,ERROR");
    return;
     }

    pump_number = atoi(cmd);
    pumps = (0x01 << (pump_number-1));
    cmd = strtok(NULL, delims);
    pump_times = atoi(cmd);

  if(pumps > 0x20)
     {
      printf0("$POPRE,ERROR\r\n");
      return;
     }

  if(pumps == 0x00)
   {
    pumps = 0x01;
    k = 5;
   }

  for(i=0;i<pump_times;i++)
   {
    for(j=0;j<=k;j++)
```

```
     {
      pump((pumps << j), ON_TIME, OFF_TIME);
     }
   }

    printf0("$POPRE,%u,%u,OK\r\n",pump_number,pump_times);

 }

void cmd_popge(char * cmd)
 {

    volatile unsigned char i, times;
    STORAGE store;

  cmd = strtok(NULL, delims);
    if(cmd == '\0')
     {
    printf0("$POPGE,ERROR\r\n");
    return;
     }

    times = atoi(cmd);

    for(i=0;i<times;i++)
        {
     pump(PHOSLOW_PUMP_1 | PHOSHIGH_PUMP_3 | CLEANER_PUMP_4 |
     CLEANER_PUMP_6, ON_TIME, OFF_TIME);
     get_phosphate_test(&store, 1);
         printf0("$TEST,%u", store.count);
     printf0(",20%02u%02u%02u", store.year, store.month, store.day);
     printf0(",%02u%02u", store.hour, store.minute);
     printf0(",%u", store.battery);
     printf0(",%u", store.temp);
     printf0(",%u", store.phos_high);
     printf0(",%u", store.sample);
     printf0(",%u,OK\r\n", store.phos_low);
        }
    //pump(CLEANER_PUMP_4 | CLEANER_PUMP_6, ON_TIME, OFF_TIME);

  printf0("$POPGE,%u,OK\r\n",times);

 }

void cmd_pomse(char * cmd)
 {

  STORAGE store;

  cmd = strtok(NULL, delims);

  if(cmd == '\0')
     {
    printf0("No arguments, Type $HELP,POMSE for more details\r\n");
    return;
     }

  pump(CLEANER_PUMP_4 | CLEANER_PUMP_6, ON_TIME, OFF_TIME);

  get_phosphate(&store, atoi(cmd));

  printf0("$POMSE,%u", store.count);
  printf0(",20%02u%02u%02u", store.year, store.month, store.day);
  printf0(",%02u%02u", store.hour, store.minute);
  printf0(",%u", store.battery);
  printf0(",%u", store.temp);
  printf0(",%u", store.phos_high);
  printf0(",%u", store.sample);
  printf0(",%u", store.phos_low);
  printf0(",OK\r\n");

 }
```

```
void cmd_porqt(void)
 {
    STORAGE store;
    volatile unsigned short index = 0;

    index = system.nextFreeDataBlock-1;
    readDataBlock(index,&store);
  printf0("$PORQT");
    printf0(",%u", store.count);
  printf0(",20%02u%02u%02u", store.year, store.month, store.day);
  printf0(",%02u%02u", store.hour, store.minute);
  printf0(",%u", store.battery);
  printf0(",%u", store.temp);
  printf0(",%u", store.phos_high);
  printf0(",%u", store.sample);
  printf0(",%u", store.phos_low);
  printf0(",OK\r\n");

  return;
 }

void cmd_flash(char * cmd)
{

    STORAGE store;
    volatile unsigned char brk = 0;
    volatile unsigned short index = 0;

  //store.count = 0;
    cmd = strtok(NULL, delims);

    if(!strcmp("READ",cmd))
     {
        cmd = strtok(NULL, delims);

        while(index < system.nextFreeDataBlock && brk != 1)
         {

          printf0("$FLASH,READ");

        if(!strcmp("LATEST",cmd))
         {
          printf0(",LATEST");
          index = system.nextFreeDataBlock-1;
         }
        else if(!strcmp("ALL",cmd))
         {
          printf0(",ALL");
        //displayAllStoredData();
         }
        else if(!strcmp("DATE",cmd))
         {
            printf0(",DATE");
            //Implement - write date-to-index function
            brk = 1;
         }
        else if(isdigit(*cmd))
         {
            index = atoi(cmd);
            brk = 1;
         }
        else
         {
            printf0("$FLASH,ERROR\r\n");
            return;
         }

        readDataBlock(index,&store);
        printf0(",%u", store.count);
      printf0(",20%02u%02u%02u", store.year, store.month, store.day);
      printf0(",%02u%02u", store.hour, store.minute);
      printf0(",%u", store.battery);
      printf0(",%u", store.temp);
      printf0(",%u", store.phos_high);
      printf0(",%u", store.sample);
      printf0(",%u", store.phos_low);
      printf0(",OK\r\n");

      index++;
         }
     }

    else if(!strcmp("ERASE",cmd))
     {
        printf0("$FLASH,ERASE");

        if((cmd=strtok(NULL, delims)) == '\0')
         {
            spi_flash_bulk_erase();
         }
        else if(isdigit(*cmd) && atoi(cmd)<=3)
         {
            spi_flash_sector_erase(atoi(cmd));
          printf0(",%u",atoi(cmd));
            //printf0(",ERROR\r\n");
         }
        printf0(",OK\r\n");
     }
    else
     {
        printf0(",ERROR\r\n");
     }

}

void cmd_test(char * cmd)
 {
    //volatile unsigned short stdev = 4095;
    STORAGE store;

    //while(1)
     //{
      //printf0("$TEST,%x,%x\r", TAR, TAIV);
     //}

    //SET_BITS(ADC_OUTPORT, ADC_0);

    //while(stdev>15)
     //{
      //stdev = get_photodiode_stdev(16);
      //printf0("$TEST,%u,OK\r\n", stdev);
     //}

    //daq_enable();
    //while(1)
     //{
      //printf0("$TEST,%u", get_photodiode());
     //}
    //daq_disable();



    get_phosphate_test(&store, 1);
    //text_test();

  printf0("$TEST,%u", store.count);
  printf0(",20%02u%02u%02u", store.year, store.month, store.day);
  printf0(",%02u%02u", store.hour, store.minute);
  printf0(",%u", store.battery);
  printf0(",%u", store.temp);
  printf0(",%u", store.phos_high);
  printf0(",%u", store.sample);
  printf0(",%u,OK\r\n", store.phos_low);

  //CLR_BITS(ADC_OUTPORT, ADC_0);
 }

void parse_command(char * part)
 {
    char * cmd;
    char * split = "$, \r";

  printf0("\r\n");

    cmd = strtok(part,split);

  if(u0rx_buff_read[0] != '$' && u0rx_buff_read[0] != 13 && u0rx_echo == 1)
        printf0("Bad Command;\r\nAll commands must start with a '$'\r\n");
    else if(!strcmp("ECHO",cmd))
        cmd_echo();
    else if(!strcmp("POSET",cmd))
        cmd_poset(cmd);
    else if(!strcmp("POPRM",cmd))
        cmd_poprm(cmd);
    else if(!strcmp("POPRE",cmd))
        cmd_popre(cmd);
    else if(!strcmp("POPGE",cmd))
        cmd_popge(cmd);
    else if(!strcmp("POMSE",cmd))
        cmd_pomse(cmd);
    else if(!strcmp("PORQT",cmd))
        cmd_porqt();
    else if(!strcmp("FLASH",cmd))
        cmd_flash(cmd);
    else if(!strcmp("TEST",cmd))
        cmd_test(cmd);
    else if(u0rx_buff_read[0] == 13)
        nop();
    else
        printf0("INVALID COMMAND\r\n");

  printf0(">>");
    clear_uart0_buffers();
 }

void start(void)
 {

  set_defaults();
    basic_timer_setup();
    timer_a_setup(0x7FFF);

  printf0("\r\nPort 0: MSP430 Running the GSMPO4-II 115200bps\r\n");
  printf1("\r\nPort 1: MSP430 Running the GSMPO4-II 115200bps\r\n");

  system.nextFreeDataBlock = findNextSlot();

    printf0("Press Return to Start");

    system.flag = u0rx_flag;
  while(system.flag == u0rx_flag){nop();}
    printf0("\r                     \r");

    printf0(">>");

    return;
 }
```

```
#ifndef _CMD_IMP_H
#define _CMD_IMP_H

/*** forward references ***/

//extern void inline LED1_on() {SET_BITS(LED_OUTPORT, LED_1); }
//extern void inline LED1_off() {CLR_BITS(LED_OUTPORT, LED_1); }
//extern void inline LED2_on() {SET_BITS(LED_OUTPORT, LED_2); }
//extern void inline LED2_off() {CLR_BITS(LED_OUTPORT, LED_2); }
//extern void inline LED3_on() {SET_BITS(LED_OUTPORT, LED_3); }
//extern void inline LED3_off() {CLR_BITS(LED_OUTPORT, LED_3); }
//extern void inline LED4_on() {SET_BITS(LED_OUTPORT, LED_4); }
//extern void inline LED4_off() {CLR_BITS(LED_OUTPORT, LED_4); }

//extern void inline LEDs_on() {SET_BITS(LED_OUTPORT, LED_1 | LED_2 | LED_3
| LED_4); }
//extern void inline LEDs_off() {CLR_BITS(LED_OUTPORT, LED_1 | LED_2 |
LED_3 | LED_4); }

void start(void);
void parse_command(char * part);

#endif /* _CMD_IMP_H */
```

```
#include <stdio.h>

#include "system.h"
#include "gsm.h"
#include "uart.h"
#include "utility.h"

volatile unsigned char debug = 0;

void gsm_on()
 {
    gsm_igt_on();
    busier_wait(0x0040);
    gsm_igt_off();
    busier_wait(0x0080);
 }

void gsm_off()
 {
    gsm_pd_on();
    busier_wait(0x0040);
    gsm_pd_off();
    //busier_wait(0x0080);
 }

void text_data(DATATEXT * text)
 {

    volatile unsigned int i;

    gsm_on();
    busier_wait(0x0080);

  printf1("AT+CMGF=1"); //Set Modem to Textmode
  putchar_u1tx(13);
  busier_wait(0x0040);

  printf1("AT+CMGD=1"); //Clear Text Message
  putchar_u1tx(13);
  busier_wait(0x0040);

  printf1("AT+CMGW=1"); //Write the message
  putchar_u1tx(13);
  busier_wait(0x0040);

  printf1("$POTXT,%u,20%02u%02u%02u,%02u%02u,%u,%u;",text->count,
                                                     text->year,
                                                     text->month,
                                                     text->day,
                                                     text->hour,
                                                     text->minute,
                                                     text->sample_time,
                                                     text->reaction_time);

  for(i=0;i<=4;i++)
   {
    printf1("%u,%u,%u,%u,%u;",text->battery[i],
                              text->temp[i],
                              text->phos_high[i],
                              text->sample[i],
                              text->phos_low[i]);
   }

  busier_wait(0x0040);
  putchar_u1tx(26); //ctrl-z
  busier_wait(0x0040);

  //printf1("AT+CMSS=1,0866088342,129"); //Send Message
  //putchar_u1tx(13);
  //busier_wait(0x0200);

  printf1("AT+CMSS=1,0868935040,129");
  putchar_u1tx(13);
  busier_wait(0x0200);

  //printf1("AT+CMSS=1,0871236656,129");
  //putchar_u1tx(13);
  //busier_wait(0x0200);

  printf1("AT+CMGD=1"); //Clear Text Message
  putchar_u1tx(13);
  busier_wait(0x0040);

  gsm_off();

 }

void text_test()
 {

    //volatile unsigned int i;

    gsm_on();
    busier_wait(0x0080);

  printf1("AT+CMGF=1"); //Set Modem to Textmode
  putchar_u1tx(13);
  busier_wait(0x0040);

  printf1("AT+CMGD=1"); //Clear Text Message
  putchar_u1tx(13);
  busier_wait(0x0040);

  printf1("AT+CMGW=1"); //Write the message
  putchar_u1tx(13);
  busier_wait(0x0040);

  printf1("$POTXT,OK");

  /*for(i=0;i<=4;i++)
   {
    printf1("%u,%u,%u,%u,%u;",text->battery[i],
                              text->temp[i],
                              text->phos_high[i],
                              text->sample[i],
                              text->phos_low[i]);
   }*/

  busier_wait(0x0040);
  putchar_u1tx(26); //ctrl-z
  busier_wait(0x0040);

  //printf1("AT+CMSS=1,**********,129"); //Send Message
  //putchar_u1tx(13);
  //busier_wait(0x0200);

  printf1("AT+CMSS=1,**********,129"); //Send Message
  putchar_u1tx(13);
  busier_wait(0x0200);

  printf1("AT+CMGD=1"); //Clear Text Message
  putchar_u1tx(13);
  busier_wait(0x0040);

  gsm_off();

 }

/*char sms_command(char * number, char * text)
{
    char * s;
    sprintf(s,"AT+CMGW=""%s""\r",number);

    printf1("%s",s);

    return 'f';
```

```
}*/

void send_sms(char * num, char * msg)
{
    //cmd = strtok(NULL,",");

    if(num == '\0' || msg == '\0')
    {
        printf0("Incorrect Useage\r\n");
        return;
    }

    gsm_on();
    printf0("To: %s\r\nMessage: %s\n\r",num,msg);
    printf1("AT+CMGR=1\r");
    //busier_wait(0x00FF);
    unsigned char i;

    for(i=0;i<200;i++)
    {
            printf0("[%s]  ",&u1rx_buff_read[0]);
    }

    /*

    printf1();
    uprintf(putchar_u1tx,"AT+CMGR=1"); //Set Modem to Textmode
  putchar_u1tx(13);

    uprintf(putchar_u1tx,"ATD017007604");
    putchar_u1tx(13);

    busier_wait(0x0FFF);
    printf0("%s\r\n",u1rx_buff_read);

  //uprintf(putchar_u1tx,"ATH");
    //putchar_u1tx(13);*/

    gsm_off();
}
```

```
#ifndef _GSM_H
#define _GSM_H

#include "system.h"
#include "utility.h"
#include "pin_defs.h"

/* routines for manipulating switchable power */
extern void inline gsm_igt_on()     { SET_BITS(SWP_OUTPORT, SWP_2); }
extern void inline gsm_igt_off()    { CLR_BITS(SWP_OUTPORT, SWP_2); }
extern void inline gsm_pd_on()      { SET_BITS(SWP_OUTPORT, SWP_3); }
extern void inline gsm_pd_off()     { CLR_BITS(SWP_OUTPORT, SWP_3); }


/*** forward references ***/
void ring();
void text_done();
void text_test();
void text_data(DATATEXT * text);

void send_sms(char * num, char * msg);

#endif /* _GSM_H */
```

```
#include "pin_defs.h"
#include "system.h"
#include "utility.h"
#include "command.h"
#include "phosphate.h"
#include "spi_flash.h"
#include "gsm.h"

#include "GSMPo-II.h"


int main(void)
 {

    //Function Variables
    STORAGE store;
    DATATEXT text;

  //volatile unsigned short i;

  //initialize watchdog timer
  WDTCTL =  //WDTIS0 |      //Timer Interval Select0
                    //WDTIS1 |      //Timer Interval Select1
                    //WDTSSEL |     //Selects ACLK as Source
                    //WDTCNTCL |    //Clears Count Value
                    //WDTTMSEL |    //Interval Timer Enable
                    //WDTNMI |      //Non Maskable Interupt Enable
                    //WDTNMIES |    //Non Maskable Interupt Edge Select
                    WDTHOLD |       //Timer Stop
                    WDTPW;          //Password.

  /* set up the system clock */
  /* We want an 8 MHz MCLK from our 32768 Hz xtal, so
     FLLDx divider is 2 and FLL divider (N) is 122,
     modulation is ON, FLL is on, etc. */
  SCFQCTL = 122;
  SCFI0 = FLLD_2 | FN_2;
  FLL_CTL0 = //SMCLKOFF |
             XT2OFF |
             DCOPLUS |
             XCAP10PF; /* why called 10PF? should be smaller */
  FLL_CTL1 = FLL_DIV_1;

  /* set default outputs before we switch any I/O pins to "output" */
  P1OUT  = 0x00;
  P2OUT  = 0x08; //Flash chip select 0x08 is set high on startup
  P3OUT  = 0x00;
  P4OUT  = 0x00;
  P5OUT  = 0x00;
  P6OUT  = 0x04; //ADC22 chip select 0x04 is set high on startup

  /* select either I/O or peripheral for pin (I/O = 0, peripheral = 1) */
  P1SEL  = P1_FUNCTION_PINS;
  P2SEL  = P2_FUNCTION_PINS;
  P3SEL  = P3_FUNCTION_PINS;
  P4SEL  = P4_FUNCTION_PINS;
  P5SEL  = P5_FUNCTION_PINS;
  P6SEL  = P6_FUNCTION_PINS;

  /* port direction registers (out = 1 / in = 0) */
  P1DIR  = ~P1_INPUT_PINS;
  P2DIR  = ~P2_INPUT_PINS;
  P3DIR  = ~P3_INPUT_PINS;
  P4DIR  = ~P4_INPUT_PINS;
  P5DIR  = ~P5_INPUT_PINS;
  P6DIR  = P6_INPUT_PINS | 0x01 | 0x08; //A0 and A3 are outputs

  /* interrupt edge select (0 = positive / 1 = negative) */
  P1IES  = 0x0000;
  P2IES  = 0x0000;

  /* interrupt enables (0 = disabled / 1 = enabled) */
  P1IE   = 0x0000;
```

```
P2IE   = 0x0000;

  //uart0_setup(0x0682, 0x00DE); //4800
  //uart0_setup(0x0341, 0x0009); //9600
  uart0_setup(0x0045, 0x0029); //115200
uart1_setup(0x0045, 0x0029); //115200

/* wait while clock settles */
busy_wait(0xffff);

eint(); //Enable Interupt

start();

LPM3;

  while(1)
   {
      system.flag = u0rx_flag;
      while(system.flag == u0rx_flag)
      {

          if((system.hour == system.sample_hour)&&(system.minute ==
          system.sample_minute))
           {

              system.count = (system.count > 4) ? 0 : system.count;
              system.sample_minute = system.minute + system.sample_time;
              system.sample_hour = system.hour +
              (system.sample_minute/60);
              system.sample_minute = (system.sample_minute>=60) ?
              (system.sample_minute-60):system.sample_minute;
              system.sample_hour = (system.sample_hour>=24) ? (system.
              sample_hour-24):system.sample_hour;

        get_phosphate(&store, system.reaction_time);

      printf0("$POMSE,%u", store.count);
        printf0(",20%02u%02u%02u", store.year, store.month, store.day);
          printf0(",%02u%02u", store.hour, store.minute);
          printf0(",%u", store.battery);
          printf0(",%u", store.temp);
          printf0(",%u", store.phos_high);
          printf0(",%u", store.sample);
          printf0(",%u", store.phos_low);
          printf0(",OK\r\n");

      if(system.count == 0)
       {
        text.count = store.count;
        text.year = store.year;
        text.month = store.month;
        text.day = store.day;

        text.hour = store.hour;
        text.minute = store.minute;

        text.sample_time = system.sample_time;
        text.reaction_time = system.reaction_time;
       }

      text.battery[system.count] = store.battery;
      text.temp[system.count] = store.temp;
      text.phos_high[system.count] = store.phos_high;
      text.sample[system.count] = store.sample;
      text.phos_low[system.count] = store.phos_low;

        if(system.count == 4)
         {
          text_data(&text);
         }
        system.count++;
      }
```

```
        }

     parse_command(u0rx_buff_read);
    }

 }

/* print string (no formatting) using pchar as putchar routine */
/*void uprint_string(void (*func)(char c), const char *s)
 {
  for (; *s; s++) func(*s);
 }*/
```

```
#ifndef _PO_H
#define _PO_H

/*** forward references ***/

extern void inline LED1_on() {SET_BITS(LED_OUTPORT, LED_1); }
extern void inline LED1_off() {CLR_BITS(LED_OUTPORT, LED_1); }
extern void inline LED2_on() {SET_BITS(LED_OUTPORT, LED_2); }
extern void inline LED2_off() {CLR_BITS(LED_OUTPORT, LED_2); }
extern void inline LED3_on() {SET_BITS(LED_OUTPORT, LED_3); }
extern void inline LED3_off() {CLR_BITS(LED_OUTPORT, LED_3); }
extern void inline LED4_on() {SET_BITS(LED_OUTPORT, LED_4); }
extern void inline LED4_off() {CLR_BITS(LED_OUTPORT, LED_4); }

//extern void inline LEDs_on() {SET_BITS(LED_OUTPORT, LED_1 | LED_2 | LED_3
| LED_4); }
//extern void inline LEDs_off() {CLR_BITS(LED_OUTPORT, LED_1 | LED_2 |
LED_3 | LED_4); }

//extern void inline LED_on(volatile unsigned char led_number)
{SET_BITS(LED_OUTPORT, led_number); }
//extern void inline LED_off(volatile unsigned char led_number)
{CLR_BITS(LED_OUTPORT, led_number); }

void uprint_string(void (*func)(char c), const char *s);

#endif /* _PO_H */
```

```
#include "utility.h"
#include "pumps.h"
#include "temp_sensor.h"
#include "batterymonitor.h"
#include "photodiode.h"
#include "spi_flash.h"
#include "system.h"
#include "phosphate.h"

#include "uart.h"

/*void get_phosphate_calibrate(CALIBRATION * calibrate)
 {
  volatile unsigned char i,j;

  calibrate->year = system.year;
  calibrate->month = system.month;
  calibrate->day = system.day;
  calibrate->hour = system.hour;
    calibrate->minute = system.minute;
  calibrate->battery = get_battery();
    calibrate->temp = get_temp(BOARD);

  for(i=0;i<3;i++)
   {

    for(j=0;j<20;j++)
     {
      pump_on(i+1,0x20,0xc0);
     }

    calibrate->phos[i] = get_photodiode();

    for(j=0;j<20;j++)
     {
      pump_mix(CLEANER1,CLEANER2,0x20,0xc0);
     }

   }

  for(j=0;j<20;j++)
   {
    pump_on(5,0x20,0xc0);
   }

  calibrate->phos[3] = get_photodiode();

  for(j=0;j<20;j++)
   {
    pump_mix(CLEANER1,CLEANER2,0x20,0xc0);
   }

  return;

 }*/

void get_phosphate_test(STORAGE * store, volatile unsigned char
reaction_time)
 {
  store->count = system.nextFreeDataBlock;
  store->year = system.year;
  store->month = system.month;
  store->day = system.day;
  store->hour = system.hour;
    store->minute = system.minute;
  store->battery = get_battery();
    store->temp = get_temp_avg(BOARD,1);
  store->phos_high = get_photodiode_avg(WARM_UP, 0x80, 128);
  store->sample = get_photodiode_avg(WARM_UP, 0x80, 128);
  store->phos_low = get_photodiode_avg(WARM_UP, 0x80, 128);

  //store_datablock(store);
  //writeDataBlock(store, system.nextFreeDataBlock);
 }

void wait_reaction(volatile unsigned char reaction_time)
 {
    volatile unsigned char minute, wait_second, wait_minute;

    //LPM3; //Low power mode
    //LPM3_EXIT; //Exit low power mode

  wait_second = system.second;
    wait_minute = system.minute + reaction_time;
  wait_minute = (wait_minute>=60) ? (wait_minute-60):wait_minute;

  //busy_wait(0xffff);
  //printf0("%u %u %u %u\r\n",wait_minute, system.minute, wait_second,
  system.second);

    minute = system.minute;
    while((system.minute != wait_minute) || (system.second != wait_second))
     {
        busy_wait(0xffff);

        if((system.minute >= wait_minute-2) && ((system.second >=
        wait_second) || (system.minute >= wait_minute)))
         {
            //LED_on = 1;
            //daq_enable();
            //uv_on();
            //printf0("$POSTT,%02u:%02u,%02u:%02u,", wait_minute,
          //                                        wait_second,
          //                                        system.minute,
          //                                        system.second);

            //printf0("%04u,%04u,%04u,OK      \r", get_battery(),
          //                                      get_temp_avg(BOARD,1),
          //
          get_photodiode_avg(WARM_UP, 0x10, 1));
         }
        //else
         //{
          printf0("$POSTT,%02u:%02u,%02u:%02u,OK            \r",
          wait_minute,

                                                                 wait_secon
                                                                 d,
                                                                 system.
                                                                 minute,
                                                                 system.
                                                                 second);
         //}

        //minute = (minute>=60) ? (minute-60):minute;
        //if(minute == system.minute)
         //{
          //printf0("%u\r\n",get_photodiode());
          //minute = system.minute + 1;
         //}
     }

    //uv_off();
    //daq_disable();

 }

void get_phosphate(STORAGE * store, volatile unsigned char reaction_time)
 {
    volatile unsigned char i, chip = COUVETTE;//SERPENTINE;
  //volatile unsigned char minute;

  store->count = system.nextFreeDataBlock;
  store->year = system.year;
  store->month = system.month;
  store->day = system.day;
  store->hour = system.hour;
    store->minute = system.minute;
  store->battery = get_battery();
    store->temp = get_temp_avg(BOARD,128);

  //CLR_BITS(ADC_OUTPORT, ADC_3); //Peristaltic Pump Off

  for(i=0;i<=19;i++) //get new sample
   {
      pump(SAMPLE_PUMP_2, ON_TIME, OFF_TIME);
     }

  for(i=0;i<=1;i++)
    pump(CLEANER_PUMP_4 | CLEANER_PUMP_6, ON_TIME, OFF_TIME);

  //get_photodiode_avg(WARM_UP, 0x80, 128);

  for(i=0;i<=chip;i++)
    pump(PHOSHIGH_PUMP_3 | REAGENT_PUMP_5, ON_TIME, OFF_TIME);

  wait_reaction(reaction_time);

  store->phos_high = get_photodiode_avg(WARM_UP, 0x80, 128);

  //uv_off();
    //daq_disable();

    for(i=0;i<=1;i++)
      pump(CLEANER_PUMP_4 | CLEANER_PUMP_6, ON_TIME, OFF_TIME);

    for(i=0;i<=chip;i++)
      pump(SAMPLE_PUMP_2 | REAGENT_PUMP_5, ON_TIME, OFF_TIME);

    wait_reaction(reaction_time);

  store->sample = get_photodiode_avg(WARM_UP, 0x80, 128);

  //uv_off();
    //daq_disable();

  for(i=0;i<=1;i++)
    pump(CLEANER_PUMP_4 | CLEANER_PUMP_6, ON_TIME, OFF_TIME);

    for(i=0;i<=chip;i++)
      pump(PHOSLOW_PUMP_1 | REAGENT_PUMP_5, ON_TIME, OFF_TIME);

    wait_reaction(reaction_time);

  store->phos_low = get_photodiode_avg(WARM_UP, 0x80, 128);

  //uv_off();
    //daq_disable();

  //SET_BITS(ADC_OUTPORT, ADC_3); //Peristaltic Pump On

    for(i=0;i<=1;i++)
      pump(PHOSLOW_PUMP_1 | CLEANER_PUMP_4 | CLEANER_PUMP_6, ON_TIME,
      OFF_TIME);
    //pump(PHOSLOW_PUMP_1 | PHOSHIGH_PUMP_3 | CLEANER_PUMP_4 |
    CLEANER_PUMP_6, ON_TIME, OFF_TIME);


  writeDataBlock(store, system.nextFreeDataBlock);

    return;

 }
```

```
#ifndef _PHOSPHATE_H
#define _PHOSPHATE_H

#include "system.h"

#define SERPENTINE 0 //Changed from 3.
#define COUVETTE 0

/*** forward references ***/

//void get_phosphate_calibrate(CALIBRATION * calibrate);
void get_phosphate_test(STORAGE * store, volatile unsigned char
reaction_time);
void get_phosphate(STORAGE * store, volatile unsigned char reaction_time);

#endif /* _PHOSPHATE_H */
```

```
//#include<conio.h>
#include "photodiode.h"
#include "adc12.h"
#include "uart.h"

#include <math.h>

volatile unsigned short get_mean(volatile unsigned short *value, volatile
unsigned char n)
 {
    volatile unsigned char i;
    volatile unsigned short mean;
  //find mean
  for(i=0; i<n; i++)
   {
    mean+=value[i];
   }
  mean=mean/n;

  return(mean);
 }

volatile unsigned short get_stdev(volatile unsigned short *value, volatile
unsigned char n)
 {
    volatile unsigned char i;
    volatile unsigned short mean = 0, stdev = 0, dev2[16];
    volatile short dev[16];

    /*value[0] = 16;
    value[1] = 1;
    value[2] = 2;
    value[3] = 3;
    value[4] = 4;
    value[5] = 5;
    value[6] = 6;
    value[7] = 7;
    value[8] = 8;
    value[9] = 9;
    value[10] = 10;
    value[11] = 11;
    value[12] = 12;
    value[13] = 13;
    value[14] = 14;
    value[15] = 15;*/

  //find mean
  for(i=0; i<n; i++)
   {
    mean+=value[i];
   }
  mean=mean/n;

  //find deviation
  for(i=0; i<n; i++)
   {
    dev[i] = value[i] - mean;
   }

  //find variance
  for(i=0; i<n; i++)
   {
    dev2[i] = dev[i]*dev[i];
   }

  for(i=0; i<n; i++)
   {
    stdev+=dev2[i];
   }
  stdev=stdev/n;

  stdev = sqrtf(stdev);
  return(stdev);

 }

volatile unsigned short get_photodiode()
 {
    volatile unsigned short adc;

    adc_set_channel(1);

  uv_on();
    busy_wait(0xffff);
    adc = adc_read();
    uv_off();
    busy_wait(0xffff);
    adc_off();

    adc = adc;

    return(adc);
 }

volatile unsigned short get_photodiode_avg(volatile unsigned char warm,
volatile unsigned char warm_time, volatile unsigned char n)
 {

    volatile unsigned char i;
    volatile unsigned long abs = 0;

    daq_enable();
    busier_wait(0x0001);

  if(warm == WARM_UP)
   {
      uv_on();
      busier_wait(warm_time);
      uv_off();
   }

    for(i=0;i<n;i++)
     {
        abs += get_photodiode();
     }
    abs = 4095 - (abs/n);

    //busy_wait(0xffff);
    daq_disable();

    return(abs);

 }

volatile unsigned short get_photodiode_fourier(volatile unsigned char warm,
volatile unsigned char warm_time, volatile unsigned char n)
 {

    volatile unsigned char i;
    volatile unsigned short abs = 0;
    volatile unsigned long acu = 0;

    daq_enable();
    busier_wait(0x0001);

  if(warm == WARM_UP)
   {
      uv_on();
      busier_wait(warm_time);
      uv_off();
   }

    for(i=0;i<n;i++)
     {
        acu += get_photodiode_avg(NO_WARM_UP, 0x00, 64);
     }
    abs = 4095 - (acu & 0xFFF);

    //busy_wait(0xffff);
    daq_disable();

    return(abs);

 }

volatile unsigned short get_photodiode_stdev(volatile unsigned char n)
 {
    volatile unsigned char i,j;
  volatile unsigned short stdev, value[n], value_mean[n];

  daq_enable();
    busier_wait(0x0040);

  //uv_on();
  for(i=0;i<n;i++)
   {
    for(j=0;j<n;j++)
     {
      value[j] = get_photodiode();
     }
    value_mean[i] = get_mean(value, n);
   }

  stdev = get_stdev(value_mean, n);

  //busy_wait(0xffff);
  daq_disable();

  return(stdev);
 }
```

```
#ifndef _PHOTODIODE_H
#define _PHOTODIODE_H

#include "utility.h"
#include "pin_defs.h"

#define WARM_UP 1
#define NO_WARM_UP 0

/* routines for manipulating emitter and photodiode */
extern void inline uv_on() {SET_BITS(DAQ_OUTPORT, DAQ_EMMITTER); }
extern void inline uv_off() {CLR_BITS(DAQ_OUTPORT, DAQ_EMMITTER); }

extern void inline daq_enable()     { SET_BITS(SWP_OUTPORT, SWP_1); }
extern void inline daq_disable()    { CLR_BITS(SWP_OUTPORT, SWP_1); }

/*** forward references ***/

volatile unsigned short get_photodiode();
volatile unsigned short get_photodiode_avg(volatile unsigned char warm,
volatile unsigned char warm_time, volatile unsigned char n);
volatile unsigned short get_photodiode_fourier(volatile unsigned char warm,
volatile unsigned char warm_time, volatile unsigned char n);
volatile unsigned short get_photodiode_stdev(volatile unsigned char n);

#endif /* _PHOTODIODE_H */
```

```
#ifndef _PIN_DEFS_H
#define _PIN_DEFS_H

#include <io.h>

/* I/O port definitions */
#define PUMP_OUTPORT        P1OUT
#define PUMP_INPORT       P1IN
#define PUMP_DIRPORT        P1DIR

#define FLASH_OUTPORT       P2OUT
#define FLASH_INPORT          P2IN
#define FLASH_DIRPORT       P2DIR

#define LED_OUTPORT         P3OUT
#define LED_INPORT          P3IN
#define LED_DIRPORT         P3DIR

#define SWP_OUTPORT         P4OUT
#define SWP_INPORT          P4IN
#define SWP_DIRPORT         P4DIR

#define SPARE_OUTPORT         P5OUT
#define SPARE_INPORT          P5IN
#define SPARE_DIRPORT         P5DIR

#define ADC_OUTPORT         P6OUT
#define ADC_INPORT          P6IN
#define ADC_DIRPORT         P6DIR

#define DAQ_OUTPORT         P6OUT
#define DAQ_INPORT          P6IN
#define DAQ_DIRPORT         P6DIR

#define PERI_OUTPORT          P6OUT
#define PERI_INPORT         P6IN
#define PERI_DIRPORT          P6DIR


/* PUMP port pin definitions */
#define PUMP_1                0x01
#define PUMP_2              0x02
#define PUMP_3              0x04
#define PUMP_4              0x08
#define PUMP_5                0x10
#define PUMP_6                0x20

#define PHOSLOW_PUMP_1      0x01
#define SAMPLE_PUMP_2         0x02
#define PHOSHIGH_PUMP_3     0x04
#define CLEANER_PUMP_4      0x08
#define REAGENT_PUMP_5      0x10
#define CLEANER_PUMP_6      0x20

/* FLASH port pin definitions */
#define FLASH_MOSI          0x01
#define FLASH_MISO          0x02
#define FLASH_SCK               0x04
#define FLASH_NCS               0x08
#define SER0_TX               0x10
#define SER0_RX               0x20
#define HEATER                0x40
#define TEMP_ENABLE         0x80

#define UTXD0                   0x10
#define URXD0                   0x20

/* LED port pin definitions */
#define UV_LED              0x01
#define LED_UNUSED          0x02
#define LED_SENSE_A         0x04
#define UV_LED_POWER          0x08
```

```
#define LED_1                   0x10
#define LED_2                   0x20
#define LED_3                   0x40
#define LED_4                   0x80

/* Switchable Power port pin definitions */
#define SER1_TX             0x01
#define SER1_RX             0x02
#define SWP_3                 0x04
#define SWP_2                 0x08
#define SWP_1                 0x10
#define SWP_X1              0x20
#define SWP_UNUSED_1        0x40
#define SWP_UNUSED_2        0x80

#define UTXD1                 0x01
#define URXD1                 0x02

/* SPARE port pin definitions */
#define SPARE_PIN0          0x01
#define SPARE_PIN1          0x02
#define SPARE_PIN2          0x04
#define SPARE_PIN3          0x08
#define SPARE_PIN4          0x10
#define SPARE_PIN5          0x20
#define SPARE_PIN6          0x40
#define SPARE_PIN7          0x80

/* ADC port pin definitions */
#define ADC_0                       0x01
#define ADC_1                       0x02
#define ADC_2                       0x04
#define ADC_3                       0x08
#define ADC_4                       0x10
#define ADC_5                       0x20
#define ADC_6                       0x40
#define ADC_7                       0x80

/* Data Aqusition Board Definitions*/
#define DAQ_EMMITTER      0x01
#define DAQ_PHOTODIODE    0x02
#define DAQ_TEMPERATURE   0x04


/* Pin for Peristaltic Pump*/
#define PERI_PUMP               0x08

/* function (as opposed to I/O) select pins */
#define P1_FUNCTION_PINS    0x00
#define P2_FUNCTION_PINS    SER0_TX | SER0_RX
#define P3_FUNCTION_PINS    0x00
#define P4_FUNCTION_PINS    SER1_TX | SER1_RX
#define P5_FUNCTION_PINS    0x00
#define P6_FUNCTION_PINS    0x00 /*don't forget, fix later*/

/* inputs pins (as opposed to output) */
#define P1_INPUT_PINS         0x00
#define P2_INPUT_PINS         0x00
#define P3_INPUT_PINS         0x00
#define P4_INPUT_PINS         0x00 /* we'll handle the ADC input manually
*/
#define P5_INPUT_PINS         0x00
#define P6_INPUT_PINS         0x00

/* unused pins */
#if 0
#define P1_UNUSED_PINS      0xc0
#define P2_UNUSED_PINS      0x00    /* UART 0 pins */
#define P3_UNUSED_PINS      0x02
#define P4_UNUSED_PINS      0xc0    /* 0x03 are UART 1 pins */
#define P5_UNUSED_PINS      0x00
#define P6_UNUSED_PINS      0xf0
#endif
```

```
#endif /* _PIN_DEFS_H */
```

```
#ifndef _PUMPS_H
#define _PUMPS_H

#include "utility.h"
#include "pin_defs.h"

#define ON_TIME 0x20
#define OFF_TIME 0x30

extern void inline pump(volatile unsigned char pumps,
                                                  volatile unsigned char
                                                  on_time,
                                                  volatile unsigned char
                                                  off_time)
 {
    SET_BITS(PUMP_OUTPORT, pumps)
    busier_wait(on_time);
    CLR_BITS(PUMP_OUTPORT, pumps)
    busier_wait(off_time);
 }

#endif /* _PUMPS_H */
```

```
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <ctype.h>
#include <string.h>
#include "utility.h"
#include "spi_flash.h"

unsigned char STORAGE_buffer[sizeof(STORAGE)];
unsigned short return_buffer[8];

void clock()
 {
  flash_clock_high();
  flash_clock_low();
 }

volatile unsigned char spi_flash_receive_byte()
 {
    volatile unsigned char i,byte=0;
  for(i=0;i<=7;i++)
   {
    byte = byte << 1;
    byte = byte | ((FLASH_INPORT & FLASH_MISO)>>1);
    clock();
   }
  return(byte);
 }
void spi_flash_send_byte(volatile unsigned char byte)
 {
  volatile unsigned char i,bit;
  for(i=0;i<=7;i++)
   {
    bit = byte << i;
    bit = bit & 0x80;
        if(bit==0x80)
         {
          SET_BITS(FLASH_OUTPORT, FLASH_MOSI);
         }
        else
         {
          CLR_BITS(FLASH_OUTPORT, FLASH_MOSI);
         }
    clock();
   }
  return;
 }

volatile unsigned char spi_flash_test()
 {
  volatile unsigned char byte;
  flash_select();
    spi_flash_send_byte(RES);
    spi_flash_send_byte(DUMM);
    spi_flash_send_byte(DUMM);
    spi_flash_send_byte(DUMM);
    byte = spi_flash_receive_byte();
    flash_deselect();
    if(byte == 0x11)
     {
      return(0);
     }
    else
     {
      return(1);
     }
 }

void spi_flash_write_byte(volatile unsigned char byte,
                                                    volatile unsigned char
                                                    sector,
                                                    volatile unsigned short
                                                    address)
 {
    volatile unsigned char x, y;

    x = address & 0x00FF;
    y = (address >> 8) & 0x00FF;

  flash_select();
  spi_flash_send_byte(WREN);
  flash_deselect();

  flash_select();
  spi_flash_send_byte(PP);
  spi_flash_send_byte(sector);
  spi_flash_send_byte(y);
  spi_flash_send_byte(x);
  spi_flash_send_byte(byte);
  flash_deselect();

  // Make it wait until the writing is over
  flash_select();
  spi_flash_send_byte(RDSR);
  while((spi_flash_receive_byte()& 0x01) == 0x01);
  flash_deselect();

  return;
 }

volatile unsigned char spi_flash_read_byte(volatile unsigned char sector,
                                                                   volatile
                                                                   unsigned
                                                                   short
                                                                   address)
 {
    volatile unsigned char byte, x, y;

    x = address & 0x00FF;
    y = (address >> 8) & 0x00FF;

  flash_select();
  spi_flash_send_byte(READ);
  spi_flash_send_byte(sector);
  spi_flash_send_byte(y);
  spi_flash_send_byte(x);
  byte = spi_flash_receive_byte();
  flash_deselect();

  return(byte);

 }

void spi_flash_bulk_erase()
 {
  flash_select();
  spi_flash_send_byte(WREN);
  flash_deselect();

  flash_select();
  spi_flash_send_byte(BE);
  flash_deselect();

  flash_select();
  spi_flash_send_byte(RDSR);
  while((spi_flash_receive_byte()& 0x02) == 0x02);
  flash_deselect();

  system.nextFreeDataBlock = findNextSlot();

 }

void spi_flash_sector_erase(volatile unsigned char sector)
{
    //printf0("Erasing Sector [%u]\r\n",sector);
    flash_select();
  spi_flash_send_byte(WREN);
  flash_deselect();

  flash_select();
  spi_flash_send_byte(SE);
  spi_flash_send_byte(sector);
  spi_flash_send_byte(0x00);
  spi_flash_send_byte(0x00);
  flash_deselect();

  flash_select();
  spi_flash_send_byte(RDSR);
  while((spi_flash_receive_byte()& 0x02) == 0x02);
  flash_deselect();
  busier_wait(50);
}

/**
 * indexToAddress
 * converts an index to a specified location in memory in the form of a
 DataBlockAddress
 * @param block the passed index to be converted to a viable address in
 flash memory
 * @return a struct that stores the address and sector for storing
 */
DataBlockAddress indexToAddress(volatile unsigned short block) {
    DataBlockAddress db;

    if(block < 0x1000) {
        db.sector = 1;
        db.address = block*DATABLOCK_SIZE;
    }
    else if(block < 0x2000) {
        db.sector = 2;
        db.address = (block-0x1000)*DATABLOCK_SIZE;
    }
    else if(block < 0x3000) {
        db.sector = 3;
        db.address = (block-0x2000)*DATABLOCK_SIZE;
    }
    return db;
}

/**
 * addressToIndex
 * @param DataBlockAddress a struct that holds the memory address
 * @return a short that holds the wanted block memory address
 */
unsigned short addressToIndex(DataBlockAddress db) {
    unsigned short index = 0;

    if(db.sector == 1) {
        index = db.address/DATABLOCK_SIZE;
    }
    else if(db.sector == 2) {
        index = db.address/DATABLOCK_SIZE;
        index += 0x1000;
    }
    else if(db.sector == 3) {
        index = db.address/DATABLOCK_SIZE;
        index += 0x2000;
    }
    return index;
}

/**
 * writeDataBlock
 * will store a dataBlock a selected memory slot
 * @param dataBlock pointer to the STORAGE struct that holds data to be
 stored
 * @param index index to store selected dataBlock in memory [0-12,287] i.e.
 12,288 spaces
 */
void writeDataBlock(STORAGE * dataBlock, volatile unsigned short index) {
    // (1) - declare variables
    volatile unsigned char i;       // Loop index
    unsigned char * memptr;         // Character pointer to cast struct
    bytes to
    DataBlockAddress db = indexToAddress(index);        // for index
    conversion  to address

    if(db.sector>3 || db.address>0xFFF0) {
        return;
    }

    // (2) - create char pointer and point to address of storage pointer
    memptr = (void*)dataBlock;

    // (3) - store the values in spi flash memory at specified location
    for(i=0;i<DATABLOCK_SIZE;i++)   {
        spi_flash_write_byte(*(memptr++),db.sector,(db.address)+i);
    }
    system.nextFreeDataBlock++;
}

void storeData(STORAGE * dataBlock) {
    volatile unsigned char i;       // Loop index
    unsigned char * memptr;         // Character pointer to cast struct
    bytes to
    DataBlockAddress db = indexToAddress(system.nextFreeDataBlock);     //
    for index conversion    to address

    dataBlock->count = system.nextFreeDataBlock;

    if(db.sector>3 || db.address>0xFFF0) {
        return;
    }

    // (2) - create char pointer and point to address of storage pointer
    memptr = (void*)dataBlock;

    // (3) - store the values in spi flash memory at specified location
    for(i=0;i<DATABLOCK_SIZE;i++)   {
        spi_flash_write_byte(*(memptr++),db.sector,(db.address)+i);
    }
    system.nextFreeDataBlock++;
}

/**
 * readDataBlock
 * will store a dataBlock a selected memory slot
 * @param index index to store selected dataBlock in slot [0-12,287] i.e.
 12,288 spaces
 * @param dataBlock pointer to the STORAGE struct that will hold the wanted
 data
 */
void readDataBlock(volatile unsigned short index, STORAGE * dataBlock) {
    // (1) - declare variables
    DataBlockAddress db = indexToAddress(index);    // Get address from
    index
    unsigned char * memptr;
    volatile unsigned char i;                               // Loop Index

    // (2) - point to address of storage pointer
    memptr = (void*)dataBlock;

    // (3) - store the bytes in space allocated for dataBlock
    for(i=0;i<DATABLOCK_SIZE;i++)   {
        memptr[i] = spi_flash_read_byte(db.sector,db.address+i);
    }
}

/**
 * findNextSlot
 * will scan through the stored data blocks and locate the next
 */
unsigned short findNextSlot() {

    volatile unsigned short i;      // loop index
    STORAGE temp;

    // Cycles through all stored data blocks
    for(i=0;i<0x3000;i++) {
        readDataBlock(i,&temp);
        if(temp.count != i) {
            return i;
        }
    }
    return 0;
}
```

```
#ifndef _SPI_FLASH_H
#define _SPI_FLASH_H

#include "utility.h"
#include "pin_defs.h"
#include "system.h"

#define WREN 0x06 //Write Enable
#define WRDI 0x04 //Write Disable
#define RDID 0x9F //Read Identification
#define RDSR 0x05 //Read Status Register
#define WRSR 0x01 //Write Status Register
#define READ 0x03 //Read Data Bytes
#define FAST_READ 0x0B //Read Data Bytest at Higher Speed
#define PP   0x02 //Page Program
#define SE   0xD8 //Sector Erase
#define BE   0xC7 //Bulk Erase
#define DP   0xB9 //Bulk Erase
#define RES  0xAB //Release From Deep Sleep
#define DUMM 0x00 //Dummy Byte

#define DATABLOCK_SIZE 16       // Final size of the datablock in bytes
(128 bits)

typedef struct {
    unsigned short address;
    unsigned char sector;
}DataBlockAddress;


/* routines for manipulating flash memory */

extern void inline flash_select()
{
    CLR_BITS(FLASH_OUTPORT, FLASH_NCS);
    CLR_BITS(FLASH_DIRPORT, FLASH_MISO);
}

extern void inline flash_deselect()
{
    SET_BITS(FLASH_OUTPORT, FLASH_NCS);
    SET_BITS(FLASH_DIRPORT, FLASH_MISO);
}

extern void inline flash_clock_high()   { SET_BITS(FLASH_OUTPORT,
FLASH_SCK); }
extern void inline flash_clock_low()    { CLR_BITS(FLASH_OUTPORT,
FLASH_SCK); }

/*** forward references ***/
void spi_flash_bulk_erase();
void spi_flash_sector_erase(volatile unsigned char sector);
void writeDataBlock(STORAGE * dataBlock, volatile unsigned short index);
void readDataBlock(volatile unsigned short index, STORAGE * dataBlock);
unsigned short findNextSlot();
void displayAllStoredData();
#endif /* _SPI_FLASH_H */
```

```
#include <io.h>
#include <signal.h>

#include "utility.h"
#include "system.h"

volatile SYSTEM system;

volatile unsigned int phoshigh[12], phoslow[12], sample[12], battery[12],
board_temp[12], chamber_temp[12], internal_temp[12];
static unsigned char month30[4] = {4,6,9,11};

void set_defaults()
 {

    system.year = 8;
    system.month = 1;
    system.day = 1;

    system.hour = 0;
    system.minute = 0;
    system.second = 0;

    system.count = 0;

    system.sample_hour = 12;
    system.sample_minute = 0;
  system.sample_time = 10;
  system.reaction_time = 8;

 }

void basic_timer_setup()
 {
  BTCNT1 = 0x00;
  BTCNT2 = 0x00;

  BTCTL = BT_fCLK2_DIV128 | //Interupt Select
                    BT_fLCD_1K |            //For controlling LCD Frequency
                    BTDIV;                      //fCLK2 = ACLK:256
                //BTRESET;                  // BT is reset and BTIFG is
                reset if this bit is set
                //BTHOLD |                  // BT1 is held if this bit is
                set
                //BTSSEL;                   // fBT = fMCLK (main
                clock)BTCTL = BTDIV | BT_fCLK2_DIV128;

  SET_BITS(IE2, BTIE);      //Enable Interupt
 }

interrupt (BASICTIMER_VECTOR) rtc(void)
 {

    BTCNT1 = BTCNT2 = 0x00;

  IFG2 = IFG2 & ~BTIFG;
  system.second++;

  if(system.second==60)
   {
    system.minute++;
    system.second = 0;
    if(system.minute==60)
     {
        system.hour++;
      system.minute = 0;
      if(system.hour==24)
       {
        system.day++;
        system.hour = 0;
        if(system.day==28 && system.month == 2)
         {
          system.month++;
          system.day = 1;
         }
        //else if(system.day==29 && system.month == 2)
         //{
          //system.month++;
          //system.day = 1;
         //}
          else if(system.day==0)
         {
          system.month++;
          system.day = 1;
         }
          else if(system.day==31 && (system.month == month30[0] || system.
          month == month30[1] || system.month == month30[2] || system.month
          == month30 [3]))
         {
          system.month++;
          system.day = 1;
         }

        if(system.month == 13)
             {
                system.year++;
             }
       }
     }
   }
 }

void timer_a_setup(volatile unsigned short period)
 {
    TAR = 0x00;

    TACCR0 = period;
    //TACCR1 = 0x0FFF;

    //TACCTL1 = CCIE;

  TACTL = TASSEL_ACLK | //Clock Select TASSELx
                    ID_DIV1 |     //Clock Divider IDx
                    //MC_UPTO_CCR0 |    //Timer Mode Control
                    TAIE;           //Timer Interrupt enable
                  //TACLR;          //Timer Clear
 }

interrupt (TIMERA0_VECTOR) pulse(void)
 {
  //LED_on(0x10);
  //busy_wait(0xffff);
  //LED_off(0x10);
 }
```

```
#ifndef _SYSTEM_H
#define _SYSTEM_H

typedef struct {
    unsigned int year:7;        // + 07 bits
    unsigned int month:4;       // + 11 bits
    unsigned int day:5;         // + 16 bits
    unsigned int hour:6;        // + 22 bits
    unsigned int minute:7;  // + 28 bits
    unsigned int second:6;

    unsigned int flag:1;
    unsigned int count:3;

    unsigned int sample_hour:6;
    unsigned int sample_minute:8;
    unsigned int sample_time:8;
    unsigned int reaction_time:6;

    unsigned int nextFreeDataBlock:14;

}SYSTEM;

typedef struct {
    unsigned int count:14;      // +14
    unsigned int year:7;            // +21
    unsigned int month:4;           // +25
    unsigned int day:5;             // +30
    unsigned int hour:6;            // +36
    unsigned int minute:6;      // +42

    unsigned int temp:12;           // +54
    unsigned int phos_high:12;// +66
    unsigned int phos_low:12;   // +78
    unsigned int sample:12;   // +90
    unsigned int battery:12;  // +102

    unsigned long reserved:26;  // +128
}STORAGE;

typedef struct {
  unsigned int count:14;        // +14

  unsigned int year:7;          // +21
    unsigned int month:4;               // +25
    unsigned int day:5;                 // +30

    unsigned int hour:6;                // +36
    unsigned int minute:6;          // +42

    unsigned int sample_time:8;   // +50
    unsigned int reaction_time:6; // +56

    unsigned int battery[5];
    unsigned int temp[5];
  unsigned int phos_high[5];
  unsigned int sample[5];
  unsigned int phos_low[5];

}DATATEXT;

/*typedef struct {
    unsigned int count:14;      // +14
    unsigned int year:7;            // +21
    unsigned int month:4;           // +25
    unsigned int day:5;             // +30
    unsigned int hour:6;            // +36
    unsigned int minute:6;      // +42

  unsigned int battery:12;  // +54
    unsigned int temp:12;           // +66

  unsigned short phos[6];
}CALIBRATION;*/

extern volatile SYSTEM system;

//extern volatile unsigned int phoshigh[12], phoslow[12], sample[12],
battery[12], board_temp[12], chamber_temp[12], internal_temp[12];

/*** forward references ***/
void set_defaults();
void basic_timer_setup();
void timer_a_setup(volatile unsigned short period);

//extern void inline LED_on(volatile unsigned char led_number)
{SET_BITS(LED_OUTPORT, led_number); }
//extern void inline LED_off(volatile unsigned char led_number)
{CLR_BITS(LED_OUTPORT, led_number); }

#endif /* _SYSTEM_H */
```

```
#include "temp_sensor.h"

volatile short get_temp_avg(volatile unsigned short sensor, volatile
unsigned char n)
 {

    volatile unsigned int i;
    unsigned long adc = 0;
    //volatile float volt, temp;

    switch(sensor)
     {
      case BOARD:
       temp_enable();
       adc_set_channel(6);
       break;
      case CHAMBER:
       adc_set_channel(2);
       break;
      case INTERNAL:
       adc_set_channel(10);
       break;
      default:
       return 0;
     }

    for(i=0;i<n;i++)
     {
      adc += adc_read();
     }
    temp_disable();
    adc_off();

    adc = adc/n;

    /*if(sensor == INTERNAL)
     {
      volt = (2.5 * adc)/4096;
      temp = (volt - 0.986)/0.00355;
      return temp;
     }*/

    return adc;

 }
```

```
#ifndef _TEMP_SENSOR_H
#define _TEMP_SENSOR_H

#include "pin_defs.h"
#include "adc12.h"

#define BOARD 1
#define CHAMBER 2
#define INTERNAL 3

/* routines for manipulating switchable power */
extern void inline temp_enable()    { SET_BITS(FLASH_OUTPORT, TEMP_ENABLE);
}
extern void inline temp_disable()   { CLR_BITS(FLASH_OUTPORT, TEMP_ENABLE);
}

/*** forward references ***/

volatile short get_temp_avg(volatile unsigned short sensor, volatile
unsigned char n);

#endif /* _TEMP_SENSOR_H */
```

```
#include "uart.h"

/* serial buffer variables */
static volatile unsigned int u0tx_count;
static volatile unsigned int u0tx_in_p, u0tx_out_p;
static volatile char u0tx_buff[U0TX_BUFF_SIZE];

volatile char u0rx_echo, u0rx_flag;
volatile unsigned int u0rx_buff_count;

volatile char* u0rx_buff_write;
volatile char* u0rx_buff_read;

static volatile char u0rx_buff_a[U0RX_BUFF_SIZE];
static volatile char u0rx_buff_b[U0RX_BUFF_SIZE];

void uart0_setup(unsigned int uxbr, unsigned char mod) {
  /* select UART0 pins as uctrl function instead of I/O */
  P2SEL  = UTXD0 | URXD0;

  /* configure UART0 */
  SET_BITS(U0CTL, SWRST);
  SET_BITS(U0CTL, CHAR);  /* select 8N1, asynch */
  U0TCTL = SSEL_SMCLK;
  U0RCTL = 0;
  U0BR0 = uxbr & 0xff;
  U0BR1 = uxbr >> 8;
  U0MCTL = mod;
  SET_BITS(ME1, UTXE0 | URXE0);
  CLR_BITS(U0CTL, SWRST);

  /* serial port transmit ring buffers are initially empty */
  u0tx_count = 0;
  u0tx_in_p = 0;
  u0tx_out_p = 0;

  /* serial port receive buffers are initially empty and flag is set to 0
  */
  u0rx_echo = 1;
  u0rx_flag = 0;
  u0rx_buff_count = 0;
  u0rx_buff_write = u0rx_buff_a;
  u0rx_buff_write = u0rx_buff_b;

  /* enable UART RX interrupts (tx interrupt set in xmit routine) */
  SET_BITS(IE1, URXIE0);
}
void clear_uart0_buffers(void)
{
    volatile unsigned char i;
    for(i=0;i<U0RX_BUFF_SIZE;i++)
    {
        u0rx_buff_a[i] = '\0';
        u0rx_buff_b[i] = '\0';
    }
}
/* This is the uart 0 receive interrupt. */
interrupt (UART0RX_VECTOR) uart0_rcv_isr(void)
 {

  if(U0RXBUF == 3)
   {
    FCTL1 = 0;
   }

  if(u0rx_echo == 1)
   {
    U0TXBUF = U0RXBUF;
   }

  u0rx_buff_write[u0rx_buff_count] = U0RXBUF;
  u0rx_buff_count++;
  if(U0RXBUF == 8)
   {
    u0rx_buff_count = u0rx_buff_count-2;
    u0rx_buff_write[u0rx_buff_count] = 0;
   }

  if(u0rx_buff_write[u0rx_buff_count - 1] == 10) //Discard New Line
  Characters
   {
    u0rx_buff_write[u0rx_buff_count - 1] = 0;
   }

  if(u0rx_buff_count > U0RX_BUFF_SIZE || u0rx_buff_write[u0rx_buff_count -
  1] == 13)
   {
    u0rx_flag = u0rx_flag ^ 1;

    if(u0rx_flag == 1)
     {
      u0rx_buff_write = u0rx_buff_b;
      u0rx_buff_read = u0rx_buff_a;
     }
    else
     {
      u0rx_buff_write = u0rx_buff_a;
      u0rx_buff_read = u0rx_buff_b;
     }

    u0rx_buff_count = 0;
   }
 }

/* This is the uart 0 transmit interrupt, set to work with a ring
   buffer so we can send a few bytes without busy waiting. */
interrupt (UART0TX_VECTOR) uart0_xmit_isr(void) {
  /* send char out UART0 transmitter */
  U0TXBUF = u0tx_buff[u0tx_out_p++];

  /* wrap output ptr if necessary */
  if (u0tx_out_p > U0TX_BUFF_SIZE - 1) u0tx_out_p = 0;

  /* update count */
  u0tx_count--;

  /* if that was the last one, disable further interrupts */
  if (u0tx_count == 0) CLR_BITS(IE1, UTXIE0);
}

/* This routine will send a byte through uart 0.  If there is room in
   the ring buffer, it will insert the byte into the buffer and return
   immediately.  If there is no room in the buffer, it will wait until
   there is room. */
void putchar_u0tx(char data) {
  /* wait until there's room in the buffer */
  while (u0tx_count >= U0TX_BUFF_SIZE);

  /* put it in there ... */
  u0tx_buff[u0tx_in_p++] = data;

  /* wrap the input ptr if necessary ... */
  if (u0tx_in_p > U0TX_BUFF_SIZE - 1) u0tx_in_p = 0;

  /* force an interrupt now that data is ready to be sent, update count...
  */
  dint();
  SET_BITS(IE1, UTXIE0);
  u0tx_count++;
  eint();
}


void printf0(const char *fmt, ...)
{
    const char *p;
    unsigned char i;
    char s[80];
    va_list argp;

    va_start(argp, fmt);

    for(p = fmt; *p != '\0'; p++)
    {
        if(*p != '%')
        {
            putchar_u0tx(*p);
            continue;
        }

        switch(*++p)
        {
            case 'c':
                putchar_u0tx(va_arg(argp, int));
                break;
            case 'd':
                sprintf(s,"%d",va_arg(argp, int));
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
            case 'o':
                sprintf(s,"%o",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
            case 's':
                sprintf(s,"%s",va_arg(argp, char *));
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
            case 'u':
                sprintf(s,"%u",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
            case 'x':
                sprintf(s,"%x",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
            case 'X':
                sprintf(s,"%X",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
            case '%':
                putchar_u0tx('%');
                break;
            case '0':
                switch(*++p)
                {
                    case '2':
                        sprintf(s,"%02u",va_arg(argp, unsigned int));
                        break;
                    case '3':
                        sprintf(s,"%03u",va_arg(argp, unsigned int));
                        break;
                    case '4':
                        sprintf(s,"%04u",va_arg(argp, unsigned int));
                        break;
                }
                p++;
                for(i=0;i<strlen(s);i++)
                    putchar_u0tx(s[i]);
                break;
        }
    }
    va_end(argp);
```

```
}
```

```
#include "uart.h"

/* serial buffer variables */
static volatile unsigned int u1tx_count;
static volatile unsigned int u1tx_in_p, u1tx_out_p;
static volatile char u1tx_buff[U1TX_BUFF_SIZE];

volatile char u1rx_echo, u1rx_flag;
volatile unsigned int u1rx_buff_count;

volatile char* u1rx_buff_write;
volatile char* u1rx_buff_read;

static volatile char u1rx_buff_a[U0RX_BUFF_SIZE];
static volatile char u1rx_buff_b[U0RX_BUFF_SIZE];

void uart1_setup(unsigned int uxbr, unsigned char mod) {
  /* select UART1 pins as uctrl function instead of I/O */
  P4SEL  = UTXD1 | URXD1;

  /* configure UART1 */
  SET_BITS(U1CTL, SWRST);
  SET_BITS(U1CTL, CHAR);  /* select 8N1, asynch */
  U1TCTL = SSEL_SMCLK;
  U1RCTL = 0;
  U1BR0 = uxbr & 0xff;
  U1BR1 = uxbr >> 8;
  U1MCTL = mod;
  SET_BITS(ME2, UTXE1 | URXE1);
  CLR_BITS(U1CTL, SWRST);

  /* serial port transmit ring buffers are initially empty */
  u1tx_count = 0;
  u1tx_in_p = 0;
  u1tx_out_p = 0;

    /* serial port receive buffers are initially empty and flag is set to 0
    */
  u1rx_echo = 0;
  u1rx_flag = 0;
  u1rx_buff_count = 0;
  u1rx_buff_write = u1rx_buff_a;
  u1rx_buff_write = u1rx_buff_b;

  /* enable UART RX interrupts (tx interrupt set in xmit routine) */
  SET_BITS(IE2, URXIE1);
}

/* This is the uart 1 receive interrupt. */
interrupt (UART1RX_VECTOR) uart1_rcv_isr(void)
{

  if(u1rx_echo == 1)
   {
    U1TXBUF = U1RXBUF;
   }

  u1rx_buff_write[u1rx_buff_count] = U1RXBUF;
  u1rx_buff_count++;

  if(U1RXBUF == 8)
   {
    u1rx_buff_count = u1rx_buff_count-2;
    u1rx_buff_write[u1rx_buff_count] = 0;
   }

  if(u1rx_buff_write[u1rx_buff_count - 1] == 10) //Discard New Line
  Characters
   {
    u1rx_buff_write[u1rx_buff_count - 1] = 0;
   }

  if(u1rx_buff_count > U1RX_BUFF_SIZE || u1rx_buff_write[u1rx_buff_count -
  1] == 13)
   {
    u1rx_flag = u1rx_flag ^ 1;

    if(u1rx_flag == 1)
     {
      u1rx_buff_write = u1rx_buff_a;
      u1rx_buff_read = u1rx_buff_b;
     }
    else
     {
      u1rx_buff_write = u1rx_buff_b;
      u1rx_buff_read = u1rx_buff_a;
     }

    u1rx_buff_count = 0;
   }
}

/* This is the uart 1 transmit interrupt, set to work with a ring
   buffer so we can send a few bytes without busy waiting. */
interrupt (UART1TX_VECTOR) uart1_xmit_isr(void) {
  /* send char out UART1 transmitter */
  U1TXBUF = u1tx_buff[u1tx_out_p++];

  /* wrap output ptr if necessary */
  if (u1tx_out_p > U1TX_BUFF_SIZE - 1) u1tx_out_p = 0;

  /* update count */
  u1tx_count--;

  /* if that was the last one, disable further interrupts */
  if (u1tx_count == 0) CLR_BITS(IE2, UTXIE1);
}

/* this routine will send a byte through uart 1.  If there is room in
   the ring buffer, it will insert the byte into the buffer and return
   immediately.  If there is no room in the buffer, it will wait until
   there is room. */
void putchar_u1tx(char data) {
  /* wait until there's room in the buffer */
  while (u1tx_count >= U1TX_BUFF_SIZE);

  /* put it in there ... */
  u1tx_buff[u1tx_in_p++] = data;

  /* wrap the input ptr if necessary ... */
  if (u1tx_in_p > U1TX_BUFF_SIZE - 1) u1tx_in_p = 0;

  /* force an interrupt now that data is ready to be sent, update count...
  */
  dint();
  SET_BITS(IE2, UTXIE1);
  u1tx_count++;
  eint();
}

void printf1(const char *fmt, ...)
{
    const char *p;
    unsigned char i;
    char s[80];
    va_list argp;

    va_start(argp, fmt);

    for(p = fmt; *p != '\0'; p++)
    {
        if(*p != '%')
        {
            putchar_u1tx(*p);
            continue;
        }
        switch(*++p)
        {
            case 'c':
                putchar_u1tx(va_arg(argp, int));
                break;
            case 'd':
                sprintf(s,"%d",va_arg(argp, int));
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
            case 'o':
                sprintf(s,"%o",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
            case 's':
                sprintf(s,"%s",va_arg(argp, char *));
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
            case 'u':
                sprintf(s,"%u",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
            case 'x':
                sprintf(s,"%x",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
            case 'X':
                sprintf(s,"%X",va_arg(argp, unsigned int));
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
            case '%':
                putchar_u1tx('%');
                break;
            case '0':
                switch(*++p)
                {
                    case '2':
                        sprintf(s,"%02u",va_arg(argp, unsigned int));
                        break;
                    case '3':
                        sprintf(s,"%03u",va_arg(argp, unsigned int));
                        break;
                }
                p++;
                for(i=0;i<strlen(s);i++)
                    putchar_u1tx(s[i]);
                break;
        }
    }
    va_end(argp);
}

void just_return()
{

}
```

```
#ifndef _UART_H
#define _UART_H

#include <stdio.h>
#include <string.h>
#include <signal.h>
#include "utility.h"
#include "pin_defs.h"

#define U0TX_BUFF_SIZE 0x000F
#define U0RX_BUFF_SIZE 0x005F

#define U1TX_BUFF_SIZE 0x000F
#define U1RX_BUFF_SIZE 0x005F

extern volatile char u0rx_echo, u0rx_flag;
extern volatile unsigned int u0rx_buff_count;
extern volatile char* u0rx_buff_write;
extern volatile char* u0rx_buff_read;

extern volatile char u1rx_echo, u1rx_flag;
extern volatile unsigned int u1rx_buff_count;
extern volatile char* u1rx_buff_write;
extern volatile char* u1rx_buff_read;

/*** forward references ***/
void uart0_setup(unsigned int uxbr, unsigned char mod);
void uart1_setup(unsigned int uxbr, unsigned char mod);

void clear_uart0_buffers(void);

extern void putchar_u0tx(char data);
extern void putchar_u1tx(char data);

extern void printf0(const char *fmt, ...);
extern void printf1(const char *fmt, ...);

extern void just_return();
#endif /* _UART_H */
```

```
#ifndef _UTILITY_H
#define _UTILITY_H

#include <iomacros.h>
#include "uart.h"

typedef char* string;
typedef char byte;
typedef unsigned char ubyte;
typedef volatile char vchar;
typedef volatile unsigned char vuchar;

/*
#define BUSY_WAIT(x) asm("mov %0, r5\n1:\tnop\n\tdec r5\n\tjnz 1b"\
 : : "i" (x) : "r5", "cc");
*/

/* this C code is equivalent to the inline assembler code above */
extern inline volatile void busy_wait(unsigned int x) {
  unsigned int i;
  for (i = x; i > 0; i--) nop();
}

extern inline volatile void busier_wait(unsigned int x) {
  unsigned int i;
  for (i = x; i > 0; i--) busy_wait(0xffff);
}

/* set SFR with command. Usefull for bitwise operations  */
#define DO_SFR_CMD(cmd,s,v)                       \
({                                                \
__asm__ __volatile__( #cmd "    %[mask], %[port]" \
: [port] "=m" ((__typeof(s))s)                    \
: [mask] "Imr" ((__typeof__(s))v));               \
})

#define SET_BITS(port, mask) DO_SFR_CMD(bis.b, port, mask);
#define CLR_BITS(port, mask) DO_SFR_CMD(bic.b, port, mask);
#define SET_BITS_W(port, mask) DO_SFR_CMD(bis, port, mask);
#define CLR_BITS_W(port, mask) DO_SFR_CMD(bic, port, mask);

#endif /* _UTILITY_H */
```