Universidad Carlos III de Madrid

Escuela Politécnica Superior

Perfusion Bioreactor for Liver Bioengineering

Bachelor Thesis
Trabajo de Fin de Grado
Biomedical Engineering

Author: David Cancio Villalonga
External Supervisor/ Director: Juan Francisco del Cañizo López
Internal Supervisor/Tutor: Pedro Baptista Almeida de Matos

Madrid, September 2016

We choose to go to the Moon! ... We choose to go to the moon and do the other things, not because they are easy, but because they are hard.
John F. Kennedy

So much universe, and so little time.
Terry Pratchett

Universidad Carlos III de Madrid

Acknowledgements

Before starting to talk about this bachelor thesis and its content, I think it is important to say that it has not only been a project, but also a trip, which has made me grow, mature, and has shown me things and possibilities that I did not know where even possible, finally taking me to the door of a future full of hope, now as a different person to the one that excitedly started Biomedical Engineering four years ago.

This is why I think it is essential to honestly show my deepest gratitude to everyone that has taught me, guided me, helped me or supported me along the way, finally making this possible.

I want to show my deepest gratitude and admiration to Dr. Juan F. del Cañizo, not only for allowing me to work in his research group laboratory (Laboratorio de Circulación Artificial) and trusting me, but also for his infinite patience, for teaching me so many things, for being always there when I needed assistance.

To Dr. Pedro Baptista, also for trusting in me and allowing me to work in his project, but especially for his support and for guiding me as much as he could even in the distance.

And to both of them for their incredibly kind and friendly treatment, and for showing and transmitting me their unconditional passion for science and this field.

To Charo, Miriam, Helena, Andrés, Nacho, Chubi, Robert and Judith, for making the lab a nice and comfortable place to work.

To all the professors of the degree of biomedical engineering, for their openness, attention and will to help.

To my family and friends, for their unconditional love, support and inspiration during this time, and to all my colleagues of the degree of biomedical engineering, for always lending a helping hand and accompanying me throughout this journey.

To Guadalupe, for being there calming and supporting me in the worst times and pushing me to always achieve the best.

Finally, I want to show my deepest gratitude to all the people who have shaped me and converted me into the person I am today, from teachers to colleagues.

Abstract

End-stage organ failure has grown to become one of the key challenges for the medical community because of the high number of patients in waiting list for a transplant and the severe shortage of suitable organ donors. These, together with population ageing, have created an accumulation phenomenon of patients which increases the severity of the problem.

New techniques for organ preservation, organ recovery from organs not suitable for transplant, and organ recellularization attempt to tackle this problem, appearing as some of the most promising solutions.

The aim of this bachelor thesis is to continue the development of a complex liver perfusion bioreactor in order to design and develop an efficient and repeatable method for organ perfusion, decellularization and recellularization, with the final objective being the creation of a perfusion bioreactor for liver bioengineering able to be used for organ perfusion and preservation, organ decellularization and organ recellularization, able to preserve cell structure, functionality, growth and control differentiation for up to 4 weeks, while avoiding contamination and automating as much as possible the process.

In order to do this, the bioreactor will include many sensors and data acquisition systems as well as control systems for pressure, flow rate, and temperature among others.

Universidad Carlos III de Madrid

Table of Contents

Contents

Acronyms/Abbreviations

IUPAC: International Union of Pure and Applied Chemistry.

ECM: Extracellular Matrix.

iPSC: Induced Pluripotent Stem Cell.

CS: Collins Solutions (used for cold organ preservation).

SSCS: Simple Static Cold Storage.

HMP: Hypothermic Machine Perfusion.

NMP: Normothermic Machine Perfusion.

PSF: Persufflation, gaseous oxygen perfusion.

UW: University of Wisconsin (referred to a solution used for cold organ preservation).

SFSS: small-for-size syndrome.

PBLB: Perfusion Bioreactor for Liver Bioengineering.

IDE: Integrated Development Environment.

GUI: Graphical User Interface.

1 Introduction

1.1 Definition and evolution

A bioreactor is defined by the IUPAC as *"any manufactured or engineered device or system that supports a biologically active environment."* [1] This wide definition applies to both vessels used for organism-driven chemical processes, as a fermenter, and to devices or systems designed to grow cell cultures, or tissues, thus being the base of tissue engineering. These bioreactors have also evolved from the most basic petri dish that allows for cell culturing, to the highly complex bioreactors used nowadays for tissue engineering.

Perfusion comes from the French verb "*perfuser*", which means to "pour over or through", and in physiology it is defined as the process of a body delivering blood to a capillary bed in its biological tissue. It was first completely described and studied in 1920 by August Krogh, who was awarded the Nobel Prize in Physiology and Medicine for his discovery of the mechanism of regulation of capillaries in skeletal muscle.[2]

The first documents we have about the idea or possibility of artificial perfusion date back to 1813, when the Dr. Le Gallois enunciated the hypothesis that any part of the organism could be kept alive for indefinite period of time if the heart could be substituted by a device able to pump arterial blood.[3] This hypothesis was studied and improved, until it eventually was implemented in an animal model in 1935, by Dr. Lindbergh [4] and Dr. Alexis Carrel [5], and in 1937 by Dr. Demikohv.[6]

Perfusion is an essential part of organism-emulating bioreactors, as it allows mimicking the natural pumping effect of the heart and irrigation of the tissue, as well as the different concentration gradients and pressures, which are essential for cell growth, differentiation, activity, and transport.

Nowadays, complex bioreactors aim to emulate the natural physiological and chemical conditions in the human body, in order to allow cellular growth to follow an as-close-as-natural path, which ideally could be altered as desired by the scientist. The breakthroughs in this field have allowed manufacture with the help of bioreactors of complex tissues and organs. [7] [8]

This project is based around the work developed in the LCA in the Hospital Gregorio Marañón, which objective was to develop a bioreactor for liver normothermic perfusion that allowed maintaining the liver alive and functional for a span of first 24 and later 48 hours and be able to successfully transplant it. Thus, the project presented in this bachelor thesis has already a solid base and has as objectives the improvement of this bioreactor system and continue to develop it towards an ideal perfusion bioreactor for liver recellularization able to preserve cell and tissue structure and functionality as well as avoiding contamination for up to 3-4 weeks. In order to do this, the project here presented is specially based around the implementation of new sensors and actuators for the control system.

1.2 Motivation and applicability

Nowadays, one of the main concerns in the medical field is the high demand for transplants, which highly exceeds the number of organ donors, thus creating an inequality that produces a steady increase in the number of patients waiting for a transplant, as patients in the waiting list start to accumulate. [9] Another problem that produces this high inequality between patients in the waiting list and organ donors is the growth of the population and thus patients, while the number of organ donors remains more or less the same, or even decreases in some cases thanks to advances in other fields such as automobile security. [10]

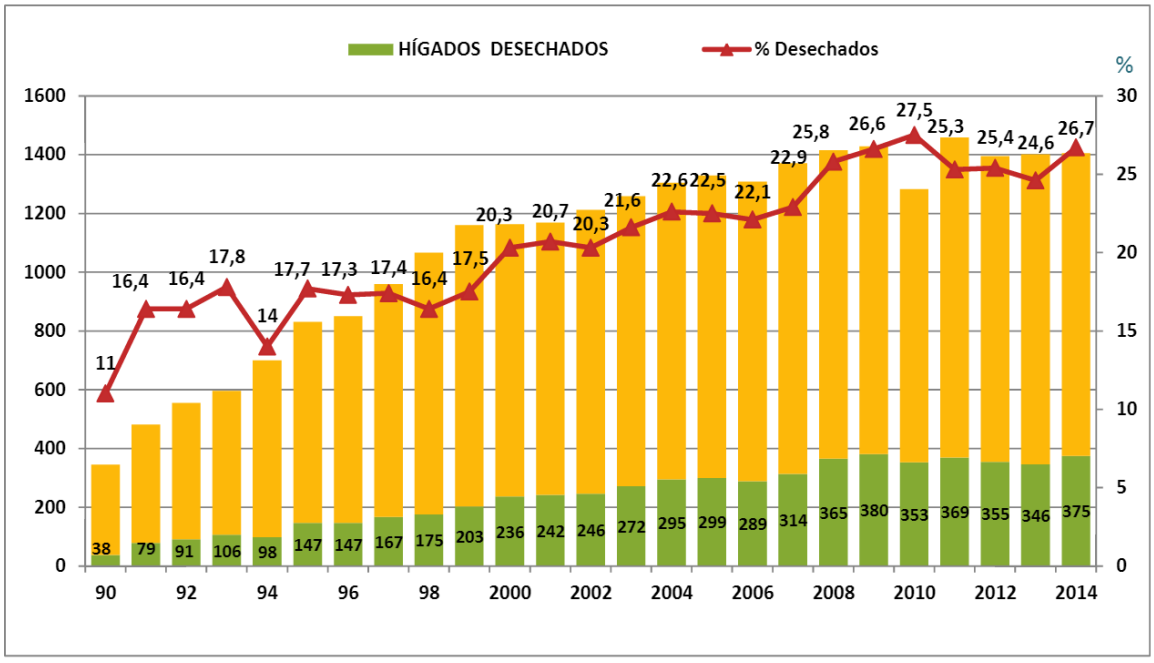
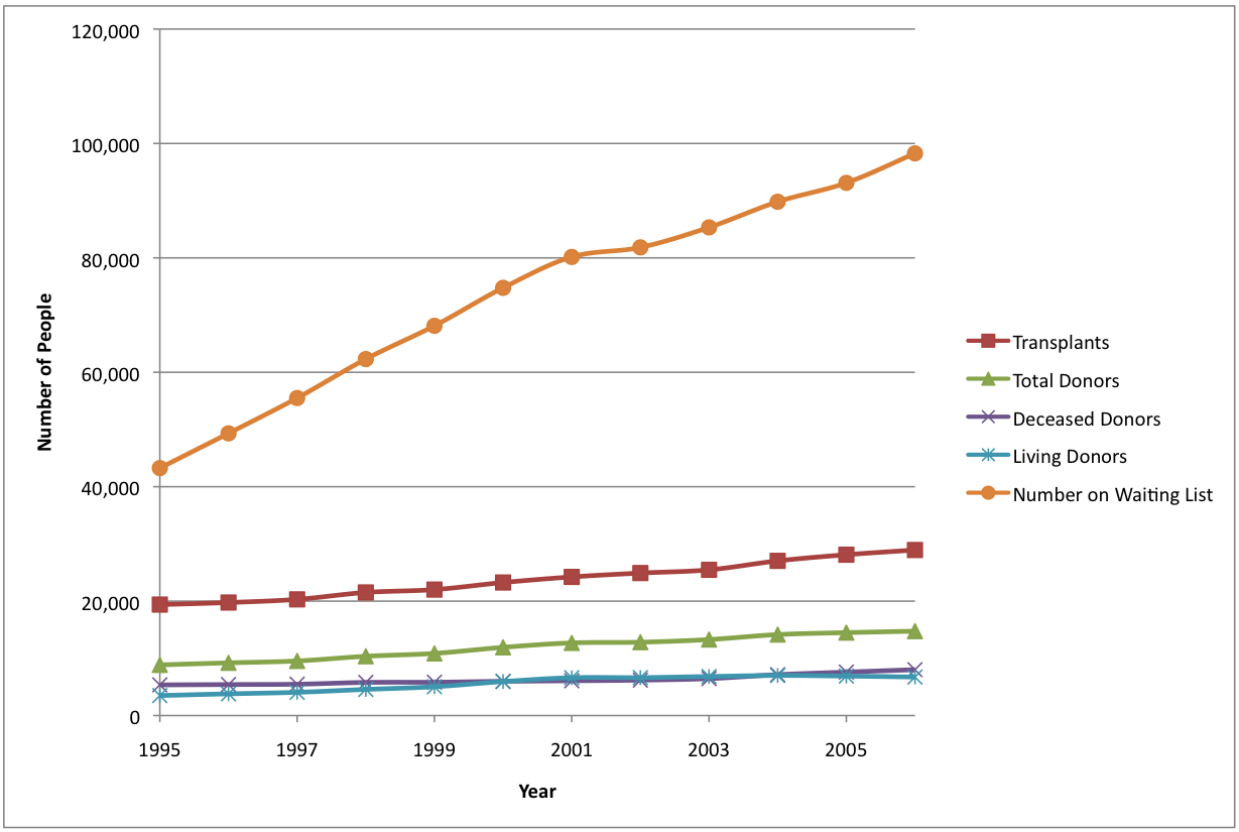


Fig. 4.7. Hígados no válidos (nº absoluto y porcentaje). 1990-2014

1 Total and rejected livers for transplant in Spain. 1990-2014



2: Yearly number of organ transplants, patients on waiting list, and living and deceased donors. Source: United Network for Organ Sharing (UNOS)

The development of a functional bioreactor for liver perfusion would greatly help to alleviate the liver shortage problem for transplants, by allowing better organ preservation, organ recovery, and allowing for the study and development in the future of functional recellularized livers.

1.2.1 Organ preservation

Organ preservation protocols have been one of the most important improvements in organ transplantation, by delivering donor organs well conserved and with high quality, as well as providing the time needed to find the most suitable patient for the donor organ.

Originally, in the 1960s, perfusion pumps were being developed for creating artificial perfusion circuits that would allow studying the physiology of organ function. Soon, it was discovered that the replacement of blood by synthetic perfusates, solutions made of electrolytes, solutes and vitamins, proportioned slower organ decay and injury. [5] [11]

By the same time it was discovered that the use of lower temperatures (0-5 degrees Celsius) altered metabolism helping to avoid organ damage during

surgery. It was discovered also that the most effective way of cooling an organ was via the whole vascular bed with perfusion of cold heparinised blood. The heparin was necessary to avoid clotting, but it still caused many problems such as vascular stasis on re-implantation. [12]

The first notable advance to organ preservation was the pioneer work by Collins and his colleagues in 1969, when they designed an acellular solution that mimicked the intracellular electrolyte balance of the mammalian cells, and it was the first advance towards cell preservation taking into account the changes produced by cooling. The solution had the added advantage of avoiding clotting problems due to the lack of cells and clotting factors. [13] The created solutions got to be known as "*Collins Solutions" (CS)*and became the standard procedure, allowing successful renal preservation for up to 36 hours.[14]

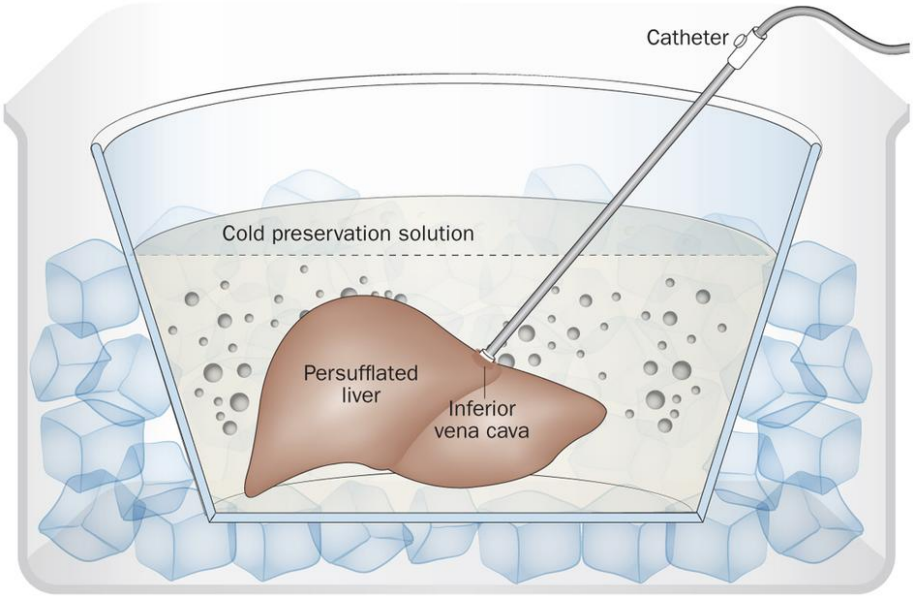
Alongside these discoveries, the research on continuous hypothermic machine perfusion (HMP) continued to develop, experimenting with oxygenated low temperature, low pressure perfusion,[15] [16] but due to the logistical problems a perfusion bioreactor presented, as well as the reliability of the equipment and machinery at the time, static flush cooling followed by ice storage, also known as simple static cold storage (SSCS) grew in popularity eventually becoming the most widely used preservation method, and the standard procedure.

After the standardization of SSCS and CS for organ preservation prior to transplant, research in SSCS focused in developing new solutions able to increase organ survivability and reduce even more organ damage due to hypoxia and ischemia. This research led to the invention of citrate solutions such as Marshall and Ross solutions, which never became very popular due to the shorter safe preservation period for organs [17], the University of Wisconsin solution (UW), and others such as Bretschneider's (Custodiol) Solution, Celsior Solution, Kyoto University Solution, or IGL-1 Solution.

The University of Wisconsin solution is today the gold standard for preservation of liver grafts since it was accepted 1989. It has demonstrated to have several advantages over Marshall and CS solutions, [18] [19] [20] and for other organs, for example in the case of the kidney, it allows for better physiological function and less histological damage than the CS. [21]

Other solutions have appeared to compete with UW for liver graft preservation, such as Bretschneider's Solution [22] or IGL-1 Solution [23], but Bretschneider's Solution appeared to cause biliary complications[24], and more study is required for IGL-1 use, although it shows promising results with smaller necrotic areas and ameliorated microcirculation of hepatic tissue after re-perfussion. [23]

Nowadays, of the main 4 different approaches to organ preservation, SSCS, HMP, normothermic machine perfusion (NMP) and persufflation (PSF,gaseous oxygen perfusion) [25], only SSCS is clinically approved currently for kidneys, livers, lungs, pancreas and heart, while HMP is only clinically approved for kidneys. The application of other methods for different organs is still in various stages of pre-clinical and early clinical studies. [14]



3 Liver preservation diagram by using persufflation and a cold preservation solution

In the specific case of the liver, SSCS with the first solutions such as CS allowed for preservation times below 10 hours, [26] [27] but the appearance of UW allowed to improve good preservation times up to 15 hours, [28] [27] becoming the gold standard. Nevertheless, HMP has showed promising results by providing improved early graft function and early resuscitation of energy metabolism.[29] Still, much larger trials are required for HMP.

NMP has also shown lately many promising results, starting to be considered as a new paradigm for liver preservation. [30] It has shown higher organ survivability, significantly higher levels of hepatocellular enzymes in the perfusate, as well has better results in bile production, factor V production, glucose metabolism and galactose clearance,[31] while also providing the advantage of easily assessing the viability of the graft before transplantation.[32] As in the case of the HMP, larger trials and more compact and reliable machinery is needed.

The perfusion system presented in this bachelor thesis is also the perfect machinery for HMP and NMP trials, extending its utility far beyond the decellularization and recellularization processes.

1.2.2 Organ recovery

It is not strange that the number of available organs for transplant is so low, if we take into account that a large percentage of the already low numbers of organs have to be discarded due to damage to the organ due to hypoxia, ischemia or other reasons, expected low survivability or biliary complications. One of the advantages of perfusion systems, independently of if they are HMP or NMP, is that they allow to assess the functionality and state of the organ before transplantation, thus checking if they are suitable for transplantation and in some cases allowing to enhance their functionality and recover normally not suitable organs.[33]

New approaches to organ preservation aim not only to minimize any injury related to the extraction and the preservation conditions, but also to recover organs that have suffered damage and improve the organ's viability. It has been shown that both temperature and oxygen supply allow to preserve and slightly recover marginal livers. Normothermic perfusions of up to 4 hours have shown to improve liver function and hepatobiliary parameters postischemia.[33] [32]

The perfusion bioreactor presented in this bachelor thesis presents accurate monitoring of many parameters and presents room for improvement by adding monitoring for other parameters, from proteins to oxygen consumption, allowing adjusting the conditions and obtaining data in real time, which can be a big advantage for organ recovery research.

In addition, some of the concerns in liver transplantation are the techniques used for partial liver transplantations, reduced liver grafts and extended resections in living donors. The application of cadaveric split or living donor liver transplantation has the potential to increase the pool of available organs, but to minimize the risk for living donors, the aim is to procure the least necessary liver volume, which can lead to the so-called small-for-size syndrome (SFSS) when the partial graft is unable to meet the functional demands. Lately, it has been shown that graft size and mass are not the exclusive limiting factors that may cause SFSS, but also the hemodynamic parameters of hepatic circulation. Control of the hepatic inflow, pressure and flow dynamics can be used to regulate organ perfusion and avoid SFSS. [34] [35] [36]

SFSS can thus be further studied as well as many other physiopathologies thanks to the perfusion system presented in this bachelor thesis, helping development of organ recovery systems. The accurate data and control provided in real time by the bioreactor can provide further insight into the behavior and response of the organ while under different stimuli.

1.2.3 Decellularization and recellularization

Organ recellularization is one of the new techniques that attempt to solve the problem of organ shortage, while also creating an infrastructure that proves useful for many other objectives such as drug experimentation and development,

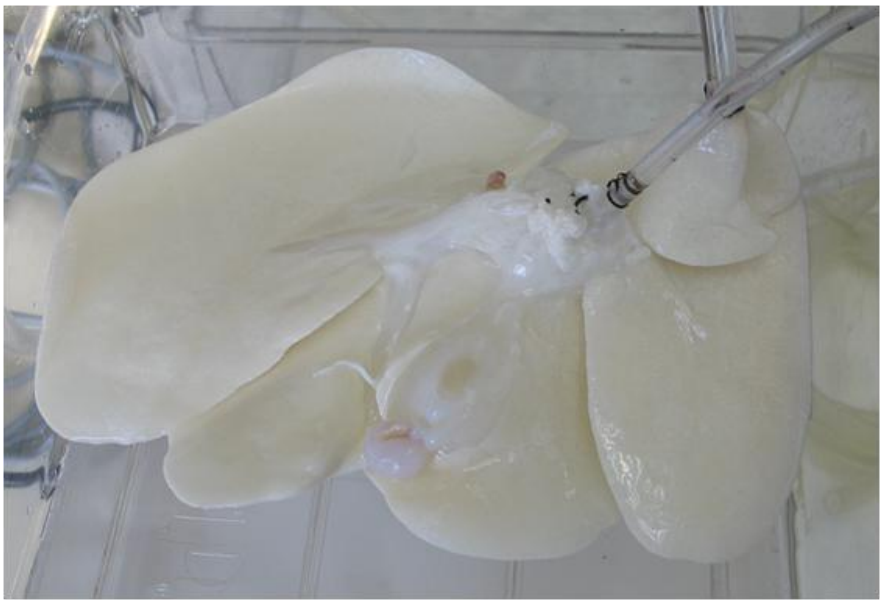
tissue engineering, or others. Organ decellularization presents itself as a technique with great potential, and nowadays, it is a required previous step for organ recellularization.

Organ decellularization was pioneered in 2006 at the McGowan Institute for Regenerative Medicine, at the University of Pittsburgh. Its objective is to isolate the extracellular matrix (ECM) of a tissue from its inhabiting cells, while maintaining as much as possible the composition and three-dimensional structure of the ECM, including the vascular network and the microenvironment, leaving the ECM as a scaffold of the original tissue or organ. [37]

All decellularization methods result in disruption of the architecture and potential loss of surface structure and composition in one way or another, it is up to the researchers choose which method is more desirable for their intentions. [38] The different agents used for decellularization can be classified as:

- **Chemical:** which include acids and bases, hypotonic and hypertonic solutions, detergents and alcohols.
- **Biological:** which include enzymes, chelating agents, or combination of several proteins and enzymes.
- **Physical & Miscellaneous:** which include temperature, force and pressure and electroporation.

Due to efficacy, ease of use and quality of the ECM obtained, two of the most used agents are SDS and Triton X-100, which are an ionic detergent and an non-ionic detergent, respectively. [39]



4 Pig liver ECM after succesful decelullarization

One of the advantages of this project is that the bioreactor can be used to obtain the ECM scaffold by organ decellularization that is going to be used later in the recellularization process. In order to do that, we only need to do a controlled perfusion of the organ using a solution with the detergent.

Organ recellularization belongs to the field of tissue engineering and consists basically in perfusing an organ ECM scaffold with media and differentiated cells, progenitor cells, adult stem cells, or iPSCs, allowing them to occupy the scaffold niches and differentiate and grow within the scaffold in a controlled manner, thus developing into the desired tissue and, if completely successful, creating a fully functional organ that can be transplanted. [40]

Two of the main advantages of organ recellularization are that almost any decellularized matrix could be used to treat any patient, and that since it allows using the patient's own cells to create the new organ, it greatly diminishes the risk of immune rejection by the patient's body.

A perfusion bioreactor is the ideal system to carry the decellularization and recellularization processes, as the organ's perfusion can be both done with detergents, in order to obtain the decellularized organ, and then with the desired media in order to grow the new tissues, while mimicking body homeostatic conditions.

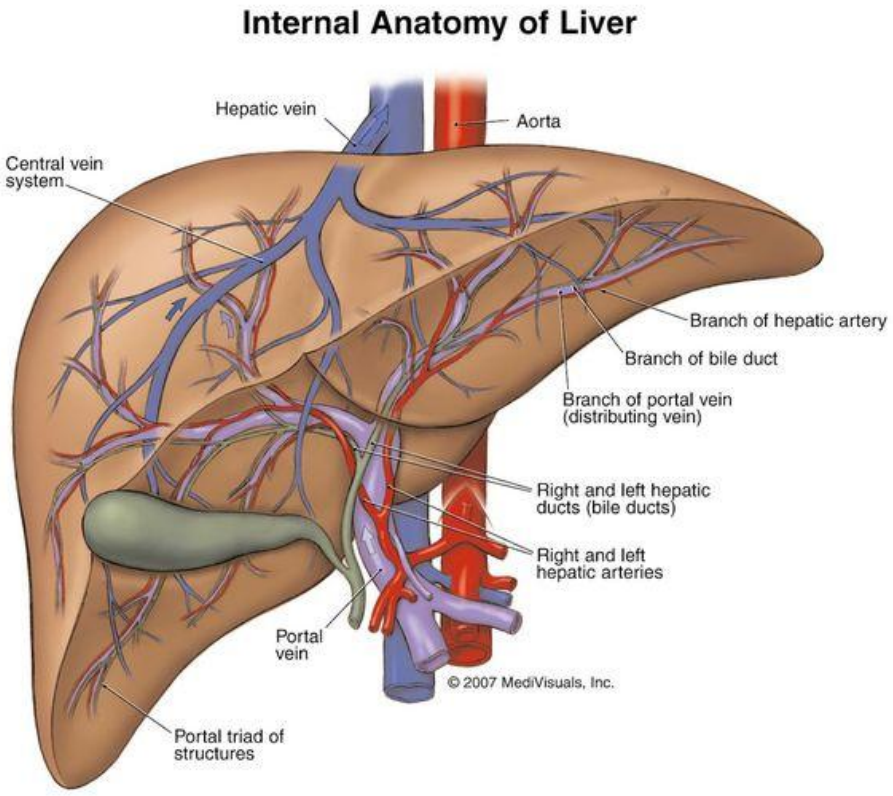
1.3 Background

It is essential to know and understand the liver in order to be able to maintain it and replicate it.

1.3.1 Anatomy of the liver

The liver is known to be the heaviest internal organ and the largest gland in the human body, contributing around 2% to the total body weight. It is located in the right upper quadrant of the abdominal cavity, extending across it.

It is formed by four lobes of unequal size and shape. From a frontal or diaphragmatic perspective, we can observe how the falciform ligament divides the liver into the largest lobes, the bigger right lobe and a smaller left lobe. The two additional lobes, caudate and quadrate, can be found extending from the posterior side of the right lobe. The caudate wraps around the inferior vena cava while the quadrate wraps around the gallbladder. [41]

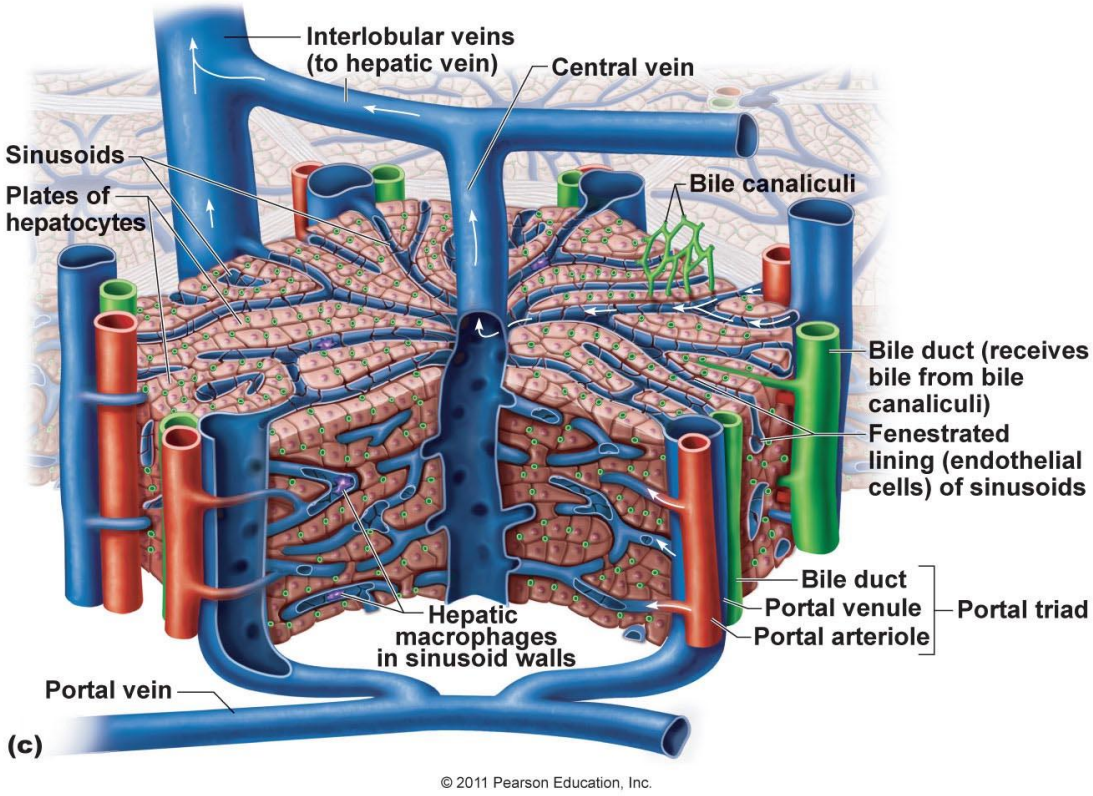


5 Liver gross anatomy

These four lobes are held together by a loose connective tissue known as areolar tissue, which surrounds the organ and the veins, arteries, ducts and nerves, forming a fibrous capsule called Glisson's capsule. This Glisson's capsule is firmly adhered to an external serous coat derived from peritoneum which completely surrounds the surface of the liver. [41]

Liver lobes are formed by hexagonal structures known as lobules, which are the basic functional units of the liver. These lobules are held together by the Glisson's capsule and formed by millions of hepatocytes, forming plates radiating from a central vein. Each of them is fed through a portal venule and a hepatic arteriole. [42]

Histologically, we can find parenchymal hepatocytes, which occupy up to 85% of the liver volume, and non-parenchymal cells, which constitute around 40% of the total of liver cells but less than a 10% of its volume. [42]



6 Anatomy of a hepatic lobule

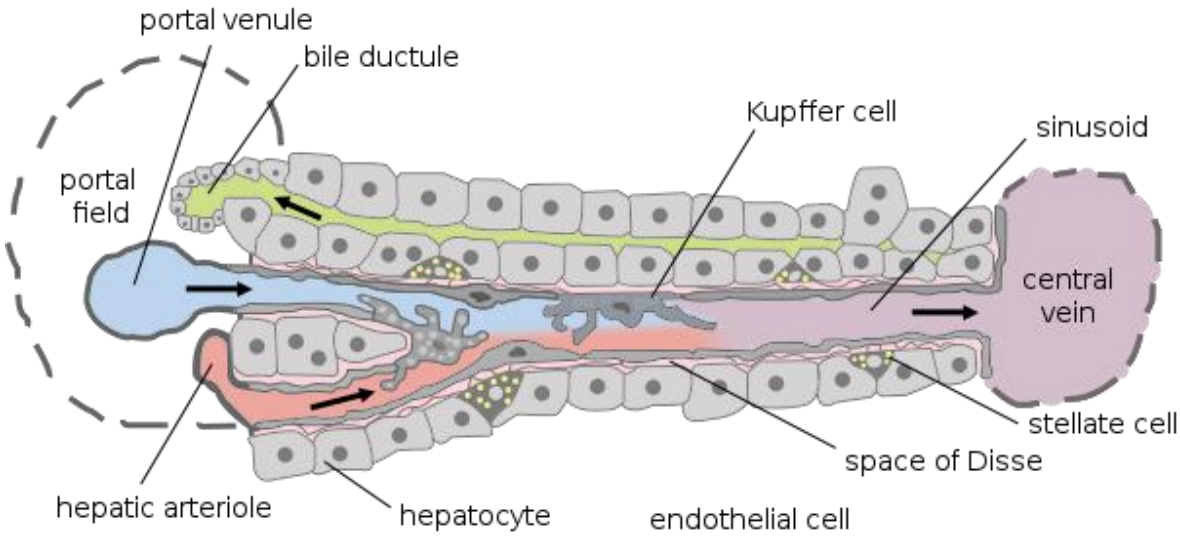
1.3.2 Hepatic irrigation

The more singular characteristic of the liver as an organ is that its main irrigation comes from the portal vein, instead of the hepatic artery. Up to 75% of the blood received by the organ is low-oxygenated venous blood that comes from the digestive tract, pancreas and spleen through the portal vein, while the other 25% is oxygenated blood received from the hepatic artery. Approximately half of the liver's oxygen demand is met by the venous blood and the other half by the arterial blood. [43] There also exist huge differences between the pressure in the portal vein and the hepatic artery, mainly due to the distance to the heart. Thus, Pressure at the portal vein for pig livers usually ranges from 12 to 17 mmHg where pressure at the hepatic artery ranges from 75 to 80 mmHg.

The blood coming from the portal venules and the blood coming from the hepatic arterioles mix upon reaching the sinusoids of the lobules, thus creating a phenomena where the cells of the lobules are never exposed to completely oxygenated blood. The sinusoids receive about 1350 milliliters of blood per minute, 1050 from the portal venules and 300 from the hepatic arterioles. [43]

The sinusoids are lined by sinusoidal endothelial cells and phagocytic Kupffer cells, which together form a fenestrated tissue, with fenestrae of around 110 nm in size. Between this fenestrated tissue and the hepatocytes there is a small space known as space of Disse or perisinusoidal space, which is filled with sinusoidal blood through the fenestrae and allows for better substance exchange between the blood and the hepatocytes, without the high stress produced by the blood flow. [43]

After passing through the sinusoids, the blood is collected at the central veins, from which it goes to the hepatic veins and finally arrives to the vena cava, from where it will return to the heart in order to be oxygenated in the lungs and start a new cycle.



7 Liver sinusoid

1.3.3 Physiology of the liver

The liver is the largest gland in the human body, and plays a big role in human metabolism. As such it has a wide range of functions, involving:

- **Digestion:** Bile is a yellowish brown fluid produced by the liver and concentrated in the gallbladder that helps in digestion. It acts as a surfactant, by helping to emulsify the lipids in food, thanks to their both hydrophilic and hydrophobic nature. In the absence of bile, fats become indigestible, causing severe problems. [42]

- **Metabolism:** The liver has a major role in carbohydrate, protein, amino acid and lipid metabolism. Maybe one of its more important tasks is to regulate the blood sugar, by performing glycogenesis and glycogenolysis, in order to transform glucose into glycogen and the other way around, and

gluconeogenesis, in order to synthetize glucose from amino acids, lactate or glycerol. Glycerol is produced by adipose and liver cells by lipolysis. [44]

- **Detoxification:** The liver breaks down bilirubin via glucuronidation, excreting it into bile. The liver also plays a major role in drug metabolism. Enzymes in hepatocytes metabolize toxins into inactive metabolites by breaking them down or by conjugating them with charged species. The liver also keeps hormone levels within homeostatic limits. [44]

- **Storage:** Essential nutrients, vitamins and minerals from the blood passing through the hepatic vascular system are stored in order to provide constant supply to the tissues in the body.[42]

- **Production:** Vital proteins components of blood plasma such as the coagulation factors prothrombin and fibrinogen, as well as albumins, which control the isotonic environment of the blood are produced in the liver. Other proteins involved in metabolism and transport, glucoproteins and lipoproteins are also produced by the liver. [42]

- **Immunity:** Kupffer cells that line in the sinusoids are a type of immobile macrophages that play an important role in immunity by digesting bacteria, fungi, parasites and cellular debris. The large volumes of blood passing through the hepatic portal system allow them to clean large blood volumes very quickly. [42]

2 Objectives

The objective of this bachelor thesis is none other than to continue the development of a perfusion bioreactor for liver bioengineering (PBLB) based on a bioreactor for liver perfusion designed and built by Dr. Juan Francisco del Cañizo and Lucía Gullón. This PBLB will allow continuing with the process of designing of an efficient and reproducible method for obtaining three-dimensional matrixes, as well as continue the research of organ recellularization. Thus, the obtained three-dimensional matrixes should be appropriate to be used in recellularization processes.

The final objective would be the creation of a liver perfusion bioreactor able to be used for normothermic isolated liver perfusion, organ decellularization, and to preserve cell structure, functionality, growth and control differentiation for up to 4 weeks for organ recellularization, while avoiding contamination and automating as much as possible the process.

In order to do this, the bioreactor will need to include the circuit for temperature-controlled isolated liver perfusion, sensors for different parameters and actuators in order to allow the user to remotely control the perfusion process and the organ response. It will also include several control systems for pressure, flow rate, pH, and temperature among others, allowing the user to leave the process in automatic mode if desired thanks to the control systems

The development of PBLB as described would also provide new data and useful methods to continue with many research areas. Also, the PBLB here described or variations of it may be used to solve other problems such as organ preservation and recovery, as presented in the introduction.

3 Materials & Methods

3.1 Design

The PBLB has at its core two things:

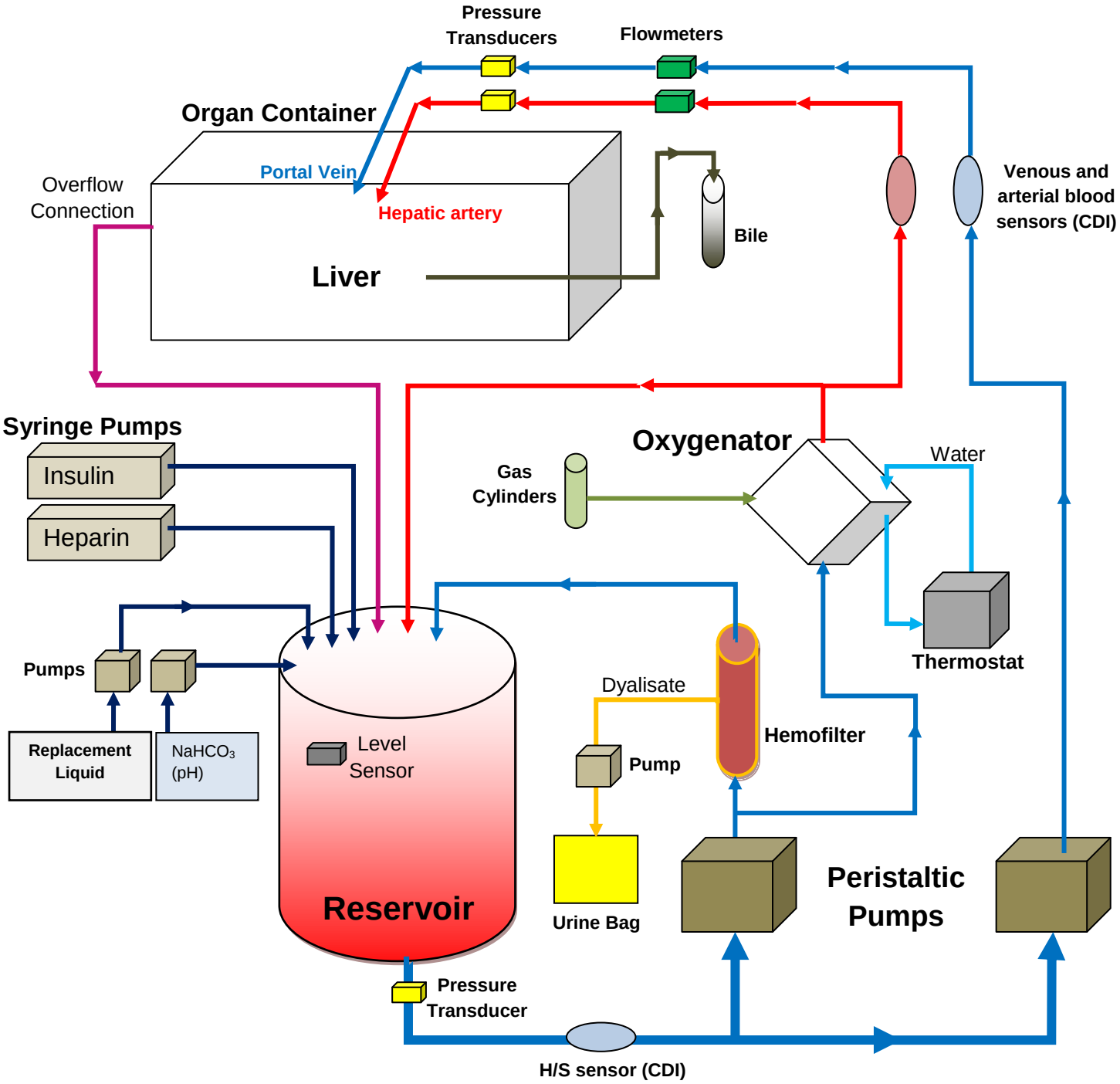
- **Physically**, a basic perfusion system formed by the elements cited in point 3.2. which, by itself could be used as a very simple and basic perfusion bioreactor.

- **Intelectually**, a central program that interprets and displays data wirelessly from a GUI in the computer, as well as controlling automatically or by user input different bioreactor parameters.

Over this core, a system of modular sensors and actuators are added in the form of modular boxes, that are each controlled thanks to a microprocessor and communicate wirelessly with the central program. The microprocessor in each of this modular boxes is in charge of controlling its actuators as commanded by the central program and receive and make a first interpretation of the data received from the different sensors.

This bachelor thesis is focused on the design, development, building, testing and programming of 2 of these modular boxes and their sensors or actuactors.The first box includes together with the microcontroller two flow measurement boards and is in charge of acquiring data from two flowmeters and control the two main peristaltic pumps of the system following the orders received by the central program.

The second box is in charge of the data acquisition from the level sensor, four temperature sensors, three weight sensors, that may also be used as volume sensors and receiving the information from the CDI Blood Parameter Monitoring System 500, an advanced machine with many integrated sensors such as pH, pCO_2, pO_2, potassium, temperature, oxygen saturation (SO_2), hematocrit, total hemoglobin, oxygen consumption (VO_2), base excess, bicarbonate and blood flow. As actuators, this second modular box only includes the control of the thermostat, in order to regulate the system temperature.

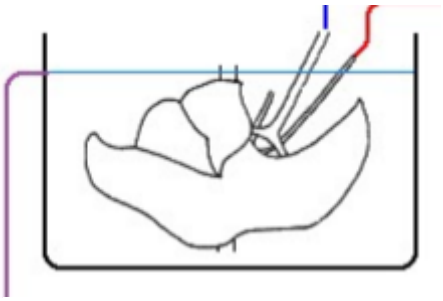

Pressure Transducers
Flowmeters
Organ Container
Overflow Connection
Portal Vein
Hepatic artery
Liver
Bile
Venous and arterial blood sensors (CDI)
Syringe Pumps
Insulin
Heparin
Oxygenator
Gas Cylinders
Water
Thermostat
Pumps
Replacement Liquid
NaHCO3 (pH)
Level Sensor
Reservoir
Dyalisate
Pump
Hemofilter
Urine Bag
Peristaltic Pumps
Pressure Transducer
H/S sensor (CDI)

3.2 Essential structure and plumbing

As seen in the design, the perfusion system is formed by many essential devices and a plumbing circuit in charge of carrying the blood or medium around without contamination.

The devices included in the essential structure replicate normal organ functions in order to emulate body physiology. These devices are:

- **Organ Container:** No organ container has been chosen yet since there has been no opportunity for trials, be it with rat or pig liver. In the case of pig liver perfusion though, the propylene container "500-type-Cage+Lid" from PANLAB, which is commercialized for rat storage, shows desirable properties such as an adequate volume of 7000cm^3 for the liver and is easy to modify for the perfusion required connections such as the overflow connection with reservoir.

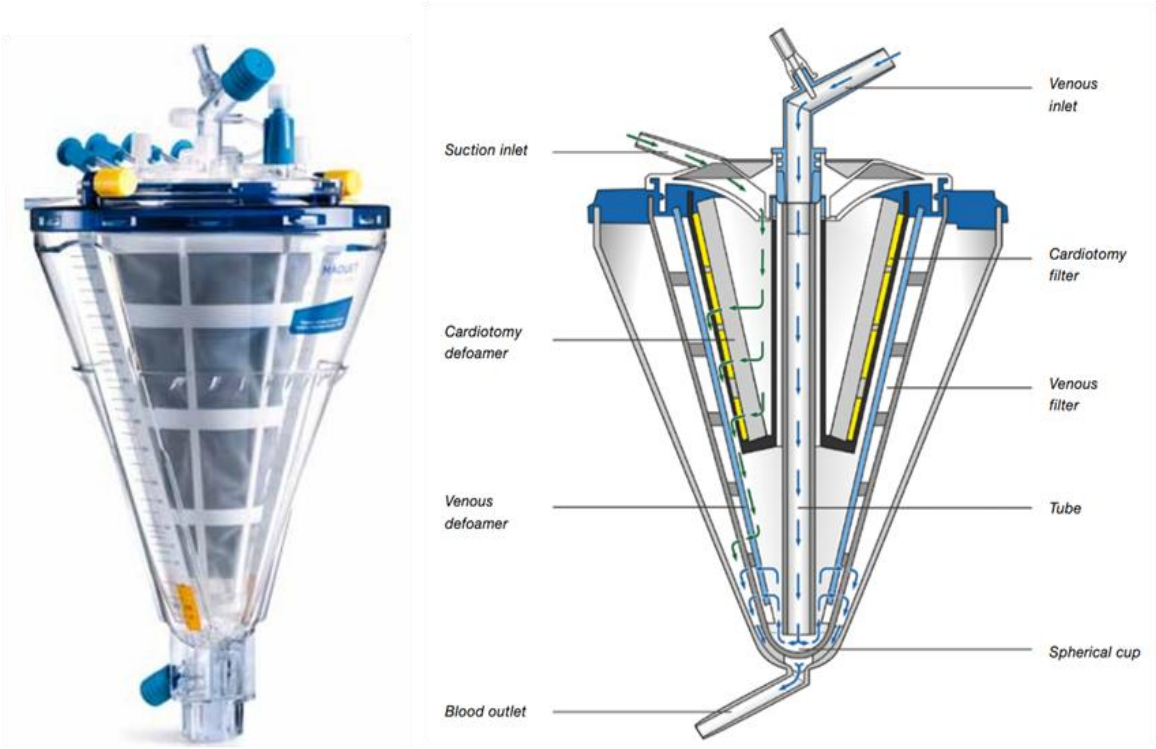


8 **Organ container schematic with overflow exit**

  In the event of a recellularization project, it would be highly recommendable to choose the smallest organ container possible such that the organ is still well hydrated and comfortable, since the increment in liquid volume required in order to keep the organ submerged, which may be only small problem for blood perfusions, can hugely increment the cost and difficulty of the project, due to the cost of making and maintaining a special medium with the required differentiation factors while keeping the aseptic conditions.

- **Reservoir:** A reservoir is required in order to store the majority of the blood or medium in the circuit. For pig liver perfusion a "Venous Cardiotomy Hardshell VHK 31000 reservoir" (Maquet, Göteborg, Sweden) is used. These reservoirs are usually used for open heart surgeries and are designed for perfusions. It has several inputs in the upper side to receive the blood or medium from the different parts of the circuit and mix them again, while having a single exit at the bottom for the blood or medium to enter the circuit again. It has a 1700 cm^3 total volume, and the conic shape helps to work with low volumes of up to 30 cm^3. It also has desirable characteristics such as being vacuum-tight and equipped with pressure relief and pressure valves, which also allow

them to drain, for example, after a decellularization process. The other main characteristic that makes the VHK 31000 a good option for a reservoir are the filters that it has incorporated, which depending on the input used, may be used or not as desired. These filters are a cardiotomy filter with volume and screen filter of 40 µm and a venous filter with a pore size of 64 µm. These filters are also designed to remove any possible air bubbles, facilitating a laminar flow. [45]



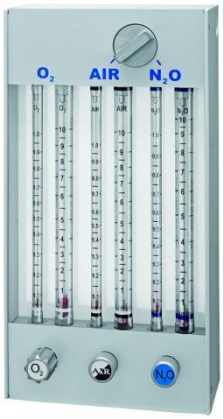
9 Venous Cardiotomy Hardshell VHK 31000 reservoir, made by Maquet

- **Oxygenator:** The oxygenator mimics the lungs by taking care of its main function: the gaseous exchange. It allows the blood or medium circulating in the system to be oxygenated and reduce its concentration of undesired gases. For pig liver perfusion, a common pediatric oxygenator QUADROX-i Pediatric (Maquet, Rastatt, Germany) is used. The oxygenator performs its function thanks to a double chamber system. In the first chamber there are gas fiber mats made of microporous polypropylene alternated with heat exchange mats of polyurethane. In the second chamber, gas fiber mats are arranged normally to each other, creating a mesh that provides optimal maximum surface contact and gaseous exchange, allowing a maximum flow rate of 2.8 l/min, and a priming volume of 99ml. The QUADROX-i oxygenator is also equipped with an arterial filter that eliminates microbubbles and particles that may alter the flow or damage the system.

A common problem of oxygenators in perfusion systems is that they act as resistances, producing a pressure drop and a need to increase pump force or velocity, which may cause other problems such as hemolysis. Thus, an objective for every oxygenator is to obtain good flows and gaseous exchange without creating a very high resistance.
This model of oxygenator also has the advantage of doubling its function by allowing the thermostat to effectively perform its function by having a double circuit where temperature-controlled water and the blood or medium can exchange heat without being in direct contact.[45]

10 Maquet Oxygenators QUADROX-i

- **Gas Blender:** The gas blender is a device that allows mixing different gases in order to supply the oxygenator with the desired concentrations of each gas. It has tubes used to see each gas pressure, which will traduce in gas concentration, as well as some simple wheel controls to manipulate the mix. For pig liver perfusion, a generic combination of oxygen, air and carbogen (30% CO_2, 70% air) is used. The required gas cylinders with the different gases or gaseous mixtures are provided by the Hospital Gregorio Marañón.



11 **Generic Gas Blender.**

- **Thermostat:** The thermostat mimics the thermoregulation capacity of warm-blooded animals to keep their body temperature stable. Due to the conditions of a perfusion system or a bioreactor, the best approach for keeping a given temperature is to use a secondary circuit so that a second liquid can exchange heat with the blood or medium without them being in direct contact. In the PBLB presented in this thesis, common tap water is used in the second circuit and it is heated by an immersive thermostat Thermotronic II (J.P.SELECTA, Barcelona, Spain). The heater coil of the thermostat is immersed in a reservoir with the water and a small pump and some tubing carry the water from the water reservoir to the oxygenator, where it exchanges heat with the PBLB blood or medium, and back to the water reservoir

12 **Generic Thermostat**

- **Hemofilter:** Also known as Flux Dialyzer, the hemofilter mimics the function of the kidneys by filtering the blood or medium of waste products of cellular activity. Kidneys filter the blood by means of diffusion

and pressure at the glomerular capillaries, which are imitated in the case of the hemofilter by a cylindrical bundle of hollow fibers, whose walls are made of a semi-permeable membrane while the end of the fibers are fixed on the top and bottom of the hemofilter, effectively separating the device into two cavities. These cavities are an internal one, with openings at top and bottom, through which the blood will flow, and a hollow cylindrical cavity outside the fibers. This second cavity will hold the filtered blood or medium, which is called dyalisate, and it will be pumped out of the hemofilter by a lateral opening into a urine bag. Different filters offer different pore size in the semi-permeable membrane, thus allowing to select by size what to filter.

The hemofilter used for pig liver perfusion is a Renaflo® II HF Minifilter™ Plus (Minntech Corp., Minneapolis, Minnesota, USA). The membrane of this hemofilters is made of Polysulfone, due to its characteristics, which include control of the pore size down to 40 nm and extremely high flow rates at very low different pressures, especially when compared to other options such as polypropylene or nylon. [46]

13 **Generic Hemofilter**

- **Peristaltic pumps:** The pumps are the core of every perfusion system, by mimicking the function of the heart and pumping the blood or medium across the circuit. Usually a system with two pumps is used, imitating the two closed circuits in the mammalian circulatory system and the different heart ventricles, one pumping oxygenated arterial blood and the other pumping venous blood. A single pump model can be developed

but it has several downsides with respect to a dual pump model, such as a much higher resistance, which translates in the need of a more powerful pump that can more easily damage the medium or blood with the increased force, for example by producing hemolysis.

Peristaltic pumps are a great option for a perfusion bioreactor for liver recellularization because of several reasons. The first, and one of the most important, is that in peristaltic pumps the transferred fluid never enters into direct contact with the fluid, in this case the blood or the medium. This greatly helps in avoiding contamination, as well as having the practical advantage of making cleaning and maintenance much easier by simply replacing the tubing.. Other reasons that make peristaltic pumps ideal are their controllability, delivery rate and their reliability. It is important to note that the peristaltic pumps are used in this project since the liquid used in recellularization will be medium, but for a normothermic isolated liver perfusion the liquid used would be blood and peristaltic pumps would not be a good option because they would cause hemolysis. For normothermic isolated liver perfusion, centrifugal pumps would be a better option.

The pumps used in this PBLB are Masterflex L/S 7523(Cole-Parmer, Illinois, United States). These pumps are formed by a Digital drive controller, which controls the pump speed and is easily accessible and configurable thanks to a LCD screen and a keypad, and would be the pump itself, and a Pump Head, which can be changed between different sizes depending on what is needed at the time. Depending on the pump head used, the Masterflex L/S 7523 can achieve flow rates between 0.001 mL/min up to 3400 mL/min.[47]

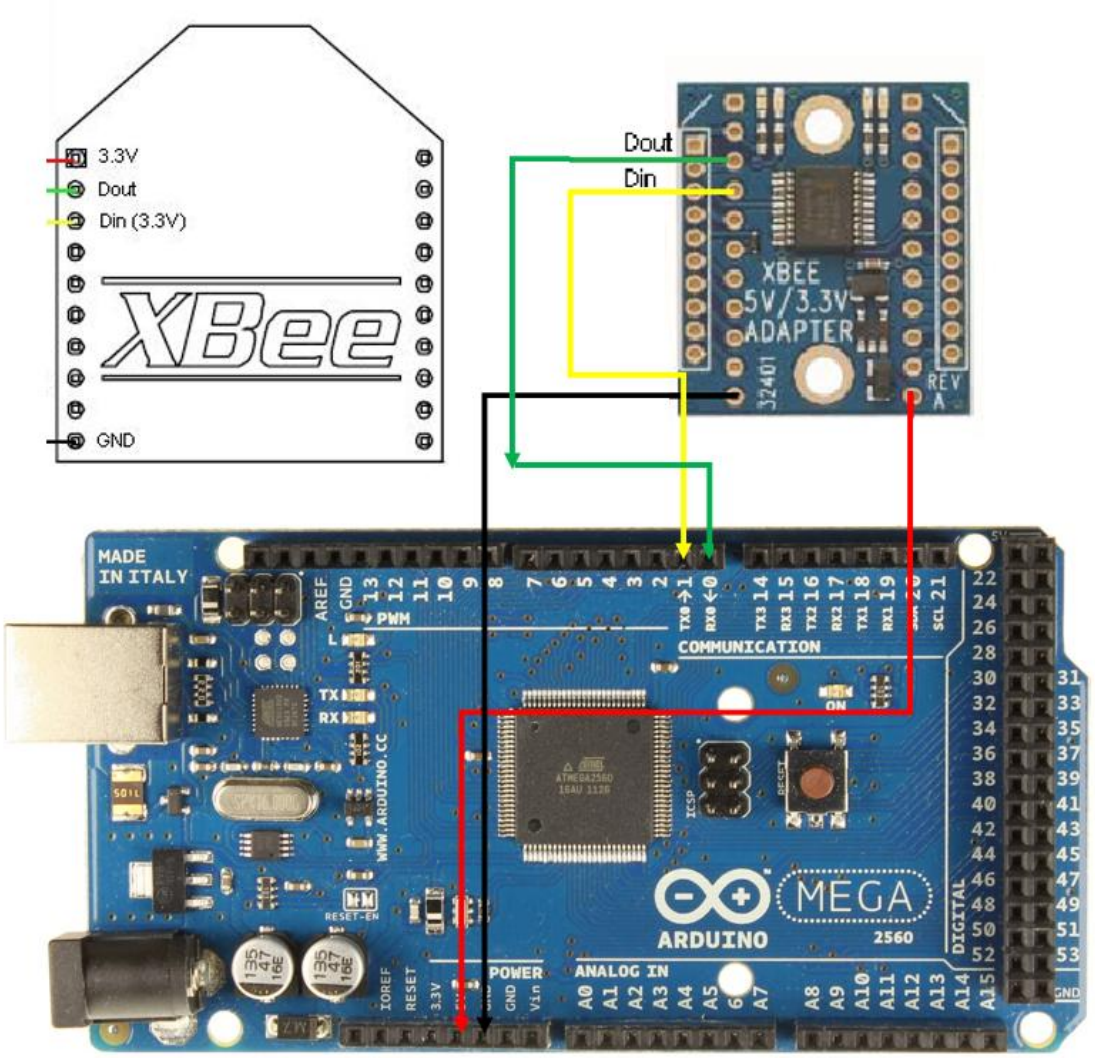
At the moment, the pump head being used is a Masterflex L/S 8-channel, 4 roller cartridge pump head, which allows for flow rates between 0.074 mL/min and 1400 mL/min, depending on the tubing diameter.

14 Masterflex pump L/S 7523 and Masterflex L/S 8-channel, 4 roller cartridge pump head

- **Plumbing:** For the circuit a silicon tube of 3/8" diameter has been used for the main areas and structure of the circuit, and a smaller silicon tube of 1/4” has been used for the zones with lower flow.

3.3 Circuitry & Modular Boxes

This bachelor thesis has focused heavily on the design, development, building, testing and integration of two new modular boxes with different sensors. As explained before, each of these boxes will have a microcontroller Arduino Mega 2560 and an Xbee S1 module for wireless communication with the computer. The microcontroller and the Xbee are communicated via serial port interface through transmission/reception cables at a rate of 9600 bauds, and the Xbee modules are supplied using the 5 V output from the microcontroller and a 5V / 3.3V adapter board as seen in the image, since the XBee has a requirement voltage of 3.3V



15 Arduino and Xbee adapter board connections

The power supply of each of the boxes consists in a generic AC adaptor (in this case from EDACPOWER Electronics Co. that produces a DC output of 12 V and a maximum of 4.5 A.

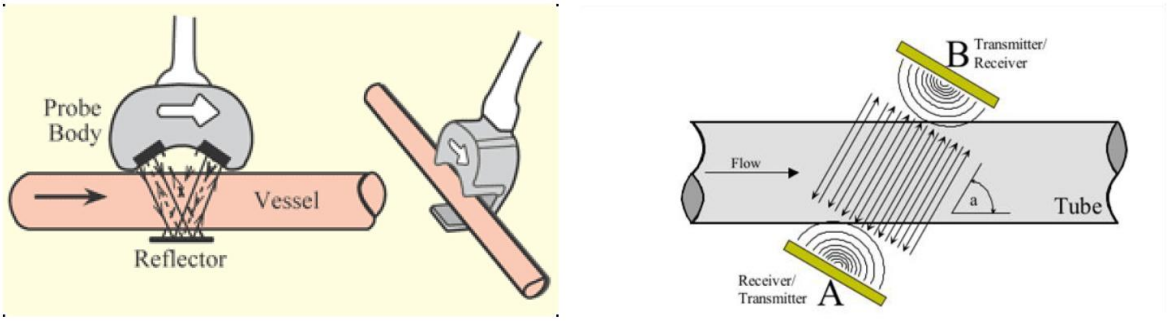
3.3.1 Module I

Modular box I has as objectives obtaining the data from the flow and controlling the main pumps. In order to gather the data from the flow a combination of a transducer or sensor and a board are needed. In order to measure the flow at two different points in the system, the hepatic artery and the portal vein, two of these boards are integrated in the box.

In this case, the boards are OEM Ultrasonic Flow Measuring System DIGIFLOW-EXT1, and the transducers are Ultrasonic Clamp-On Transducers (both developed by Emtec, gennevilliers, France). These transducers are indicated for non-invasive volumetric measurement of the liquid flowing through extracorporeal tubing systems, which represents the main advantage of these sensors as they do not need to be inside the flow or be in any contact with the liquid, which highly increases risk of contamination and could cause other problems such as protein or molecular adsorption, or flow disturbances.

16 Em-tec clamp-on flow transducer and flow measurement board

In order to do this, the transducers rely on an ultrasound transit-time method. A piezoceramic crystal is stimulated by a high frequency electric burst to send an ultrasound to a second piezoceramic crystal, which will receive the signal. By placing three or more of these piezoceramics and arranging them in a certain angle, we will receive the information of how the ultrasound is affected by the flow, both traveling in the direction of the flow and against it. This information is received by the boards, which filter it and obtain from it the average velocity of the fluid and the inner cross-sectional area of the tube, and from these the instant value for the flow volume can be calculated. The boards must be configured for the liquid that will be used as well as for the material of the tubing and sizes that will be mostly used, since these are parameters that highly affect the data.



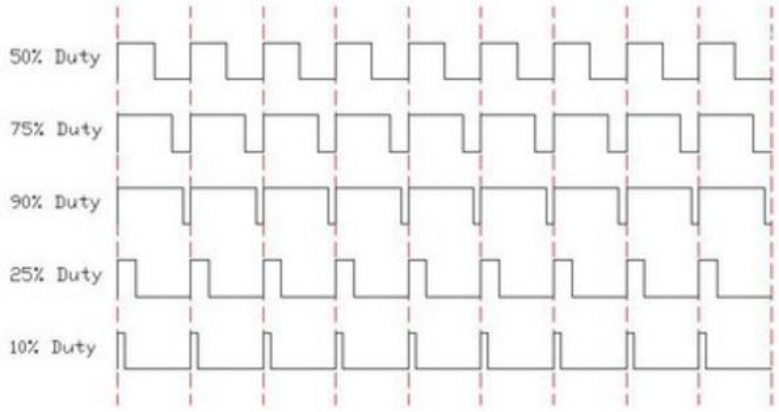
17 Schematic of the principle used by the flow transducers

Since we need to measure the flow at a minimum of two points, hepatic artery and portal vein, two flow transducers are needed, and since one board con only be connected to one flow transducer, two are needed inside the Module I box.

These boards are then connected to the arduino Mega microcontroller through a serial port each, communicating at a rate of 38400 bauds using 8 bits, 1 stop bit and without parity. The arduino microcontroller then reads the string received from the board and confirms it has the appropriate length and that it has been received correctly. Afterwards, the arduino saves the string and sends it to the computer wirelessly through the Xbee, as explained before.

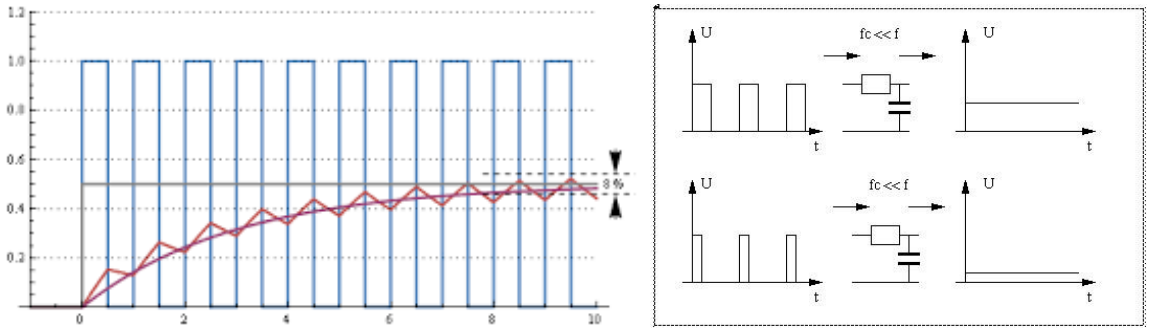
In the other hand, for pump control, the used pumps Masterflex L/S 7523 include a microcontroller inside and allow for RS-232 connection, as well as direct voltage or current input to control the speed. For the sake of simplicity, in this project the connections between the arduino microcontroller from module I and the pumps were made by simple voltage input. In this way, when the arduino microcontroller receives the order from the central program through the Xbee to set the pump speed at a certain value, it activates at the desired voltage the PWM port connected to the pump. The pump can be programmed through its GUI to stablish any desired relation between voltage and speed, but for the sake of simplicity, it has been set as 0 V being the pump stopped and 5 V, which is the maximum voltage that can be produced by the arduino microcontroller, to be the maximum pump speed.

A problem with this setup is that actually the signal produced by the arduino is not a true analog flat DC signal, but a form of square signal that varies between 0V and 5 V with a high frequency. It is also not a true square signal as it has what it's called duty cycle, which represents how much percentage of the cycle the signal is ON and which percentage is OFF, as we can see in the image.



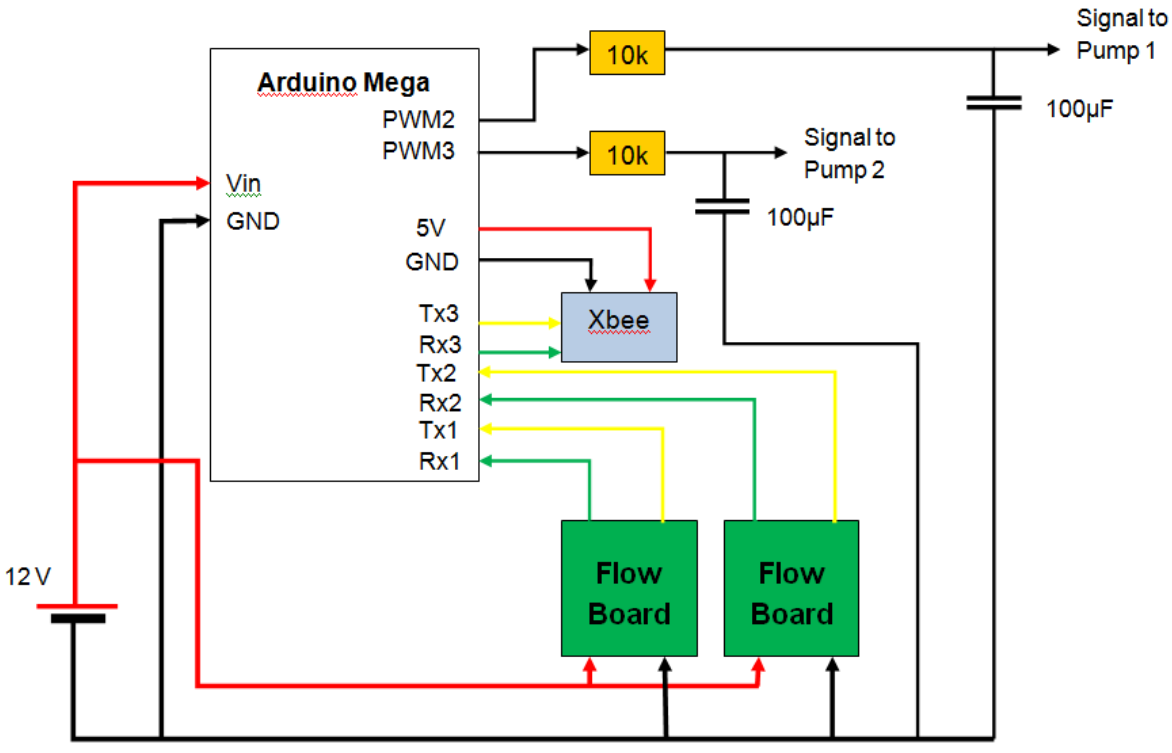
18 Duty cycles produced by the arduino microcontroller, instead of a DC signal

Many machines as well as humans cannot difference between a for example 50% duty cycle between 0V and 5V and a signal at 2.5V, but in this case, the pump control system is sensitive enough to perceive the alteration, which makes the pump oscillate around the desired speed instead of staying constant. In order to solve this problem, a low-pass filter circuit has been added to the signal sent by the arduino using a capacitor and a resistance, thus transforming the duty cycle into a real DC signal.



19 Effect of a low-pass filter on a square signal

Finally, this would be the aspect of the full circuit of module I:



20 Module I Circuit

And the final aspect of the Modular box I finished would be this:

21 Modular box I connected to one of the Masterflex L/S 7523 pumps

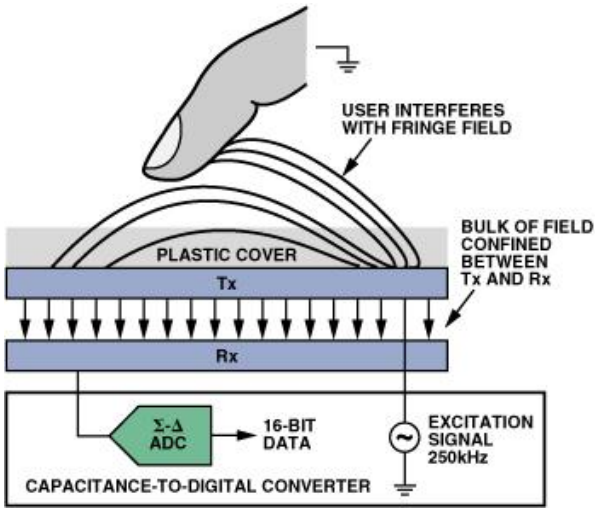
3.3.2 Module II

Modular box II has as objectives obtaining the data from the thermometers, the weight sensors, the level sensor and the CDI, making sure the received information is correct before storing it and sending it back to the central program, as well as controlling the thermostat for a set temperature.

The thermometers used are LM35, which produce a flat signal directly proportional to the temperature in degrees Celsius, being each degree equivalent to 10 mV, so, for example, a temperature of 35 ºC would be 350 mV, which is easily detected by the arduino microcontroller. So, in order to connect the thermometers we only needed to supply them with 5V and connect them to analog entries of our microcontroller. In order to be able to submerge our thermometers in the medium the thermometers were wrapped and topped with a metal piece that would easily conduct heat.

In order to turn on the thermostat when ordered by the central program, the arduino microcontroller just sends an ON signal which activates a relay, thus activating the thermostat, since it only needs an ON signal to start heating. When the sensors measure a temperature higher than the set one the thermostat is automatically turned off

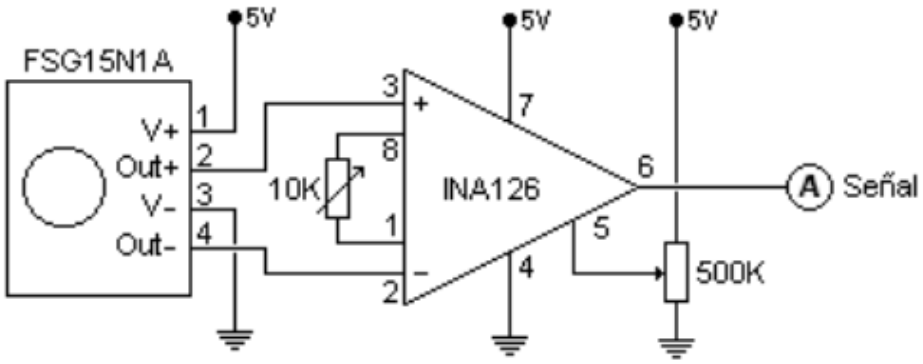
The level sensor used is a capacitive sensor, which uses the principles of a capacitor to detect if there is liquid in front of it or not. The level sensor is placed in the reservoir, at the minimum desired level. When the level of liquid falls below this point, the capacitive sensor will detect that now there is nothing interfering with the magnetic field, as the medium was interfering before, and it will stop sending an ON signal to the arduino microcontroller, which will in turn activate the pump to increase the liquid



22 Schematic of function of a capacitive sensor

The weight sensors used were FSG_15N1A, after trying to develop a system with extensiometric bands that did not get implemented ultimately due to a very low signal that caused problems, as it needed to be hugely amplified, up to 1000x and this caused a great increment in the error that was not acceptable for this type of project.

Sensors FSG_15N1a need an amplifier phase due to the low signal produced. For this, instrumentation amplifiers INA126 have been used, finally producing a signal with a maximum of 5V able to be read with precision by the arduino microcontroller.



23 Schematic of pressure sensors FSG15N1A and their amplification phase with an INA126.

Due to sensors FSG15N1A having a maximum weight that they can measure of 1500g, a small lever system has been designed to allow the sensor to measure higher weights. Knowing the weight and the density of the liquid used, be it medium or blood, the central processor can easily calculate the liquid volume lost.

Lastly, the microcontroller is connected to the CDI through a serial port with RS-232 communication. It detects that has correctly received one data string and stores and sends it back to the central program.

24 (Left) Finished modular box 2, with the top inputs for thermometers, the bottom ones from left to right: level and weights. To the right, the RS-232 connector for the CDI and below the connectors for the thermostats.

3.4 Software

The PBLB presented in this bachelor thesis is based upon a main control program that puts together, interprets and displays the data in a graphical manner as well as offering control over the whole system. **(ANNEX I)** This program runs in the computer and receives wirelessly all the data from the different sensors and module-boxes thanks to an XBee Series 1 (Digi, Minnesota, United States) module connected to the computer through an USB COMM port.

It has been designed and written by Dr. Francisco del Cañizo and by Lucia Gullón for a previous liver bioreactor, and few modifications allowed it to work properly with the new design. It is developed within an Integrated Development Environment (IDE) called GAMBAS3 and designed as an executable in order to be used independently without the need of GAMBAS.GAMBAS3 is a high-level programming language based on a variation of BASIC programming language which focuses on user-friendly interfaces, making easier the construction of the Graphical User Interfaces (GUI) that is used in the system, allowing the user to intuitively interact with the program. GAMBAS3 is an object-based IDE, similar to other IDEs such as LabVIEW, in which objects are each of the devices or parts that operate in the system. It is also important to mention that GAMBAS3 is designed to work on Linux and other Unix operating systems.

Additionally, each module-box receives and sends the information to control the system thanks to an Arduino Mega 2560 microcontroller (Arduino, Ivrea, Italy) that receives the information from each sensor and makes an initial interpretation of the data. These microcontrollers communicate with the computer and central computer thanks to other XBee Series1, and with the different actuators such as the pumps and the thermostat, when given the order from the central program.

The central GAMBAS 3 computer program can be explained by dividing it into two sections. First of all, the program receives and interprets the data received through the USB communication port provided by the XBee S1 module that receives the data wirelessly. This data, received in the form of string with a reference heading in order to split the string and recognize from where is the data, is registered in a logfile and stored as well as displayed in the GUI in real time. This allows the user to react and change parameters in real time as desired, which leads us to the second section of the program, user input and control.

The second section of the main program is focused on control and user input. There are several parameters that the user can adjust through controls in the GUI: speed of each pump for flow and pressure adjustment, temperature setting for the thermostat and control of the pumps in charge of liquid replenishment and filtration. These controls are implemented in the GUI through sliders, dials and

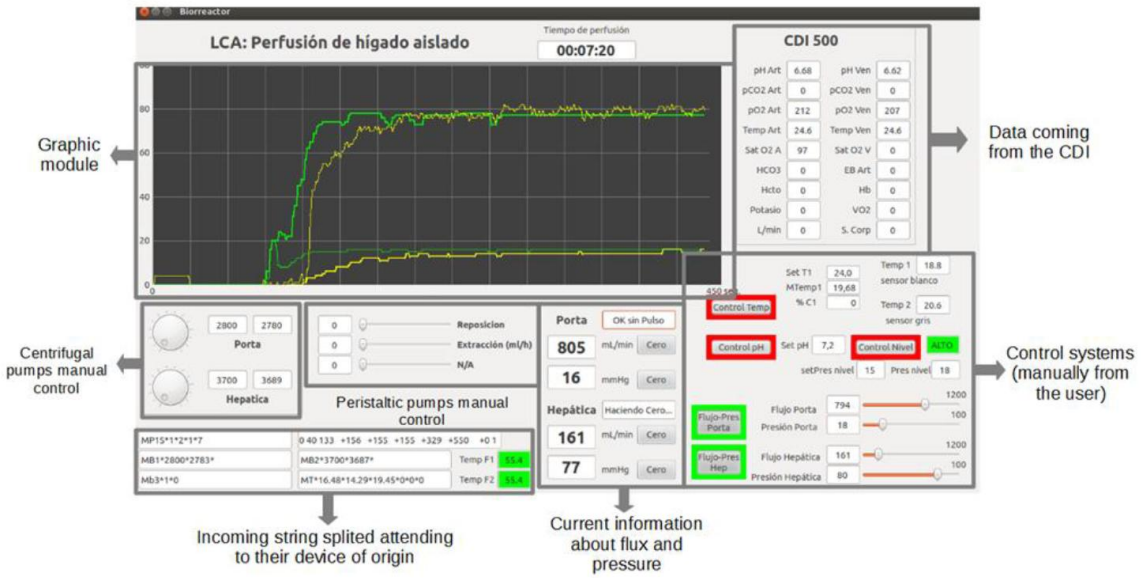
scroll bars. When one of these parameters is changed, the program sends the information through the Xbee S1, which will be received by other Xbees S1 in the module boxes. The program also includes control software that can be activated for certain parameters set by the user simply by pushing a button. When activated, the control software will attempt to maintain the parameters stable at the given values. The parameters included in the control software are temperature, level, pH, flow and pressure.

- **Temperature:** To control temperature the program automatically activates or deactivates the thermostat depending on the data received for the temperatures through the LM35 sensors.

- **Level:** Level control works by activating or deactivating the liquid replenishment pump when the level sensor activates, indicating that the liquid level in the reservoir is too low.

- **pH:** The control of the pH activates the pump that introduces Sodium Bicarbonate ($NaHCO_3$) into the system, thus raising the pH when the program receives the data from the CDI, measured at the arterial path, that the pH has fallen below the set point.

- **Flow & Pressure:** The final and most important control system is developed to control flow and pressure. Since they are directly proportional parameters, they cannot be changed independently and the control system works for a given set of either flow or pressure parameters. Since each liver presents different characteristics and different resistance to flow, this control system is essential. An adequate blood intake is required for organ survivability or adequate perfusion in decellularization or recellularization processes, in order to irrigate the whole organ, but an excessive pressure may damage the tissue or the ECM. Thus, this control system needs to obtain the required flow levels or close as possible values without surpassing certain pressure levels. It is also important to note that during perfusion due to organ humidity, stress suffered and cell activity the organ resistance may change, and thus the control system needs to be able to adapt to any changes that may occur along the process.



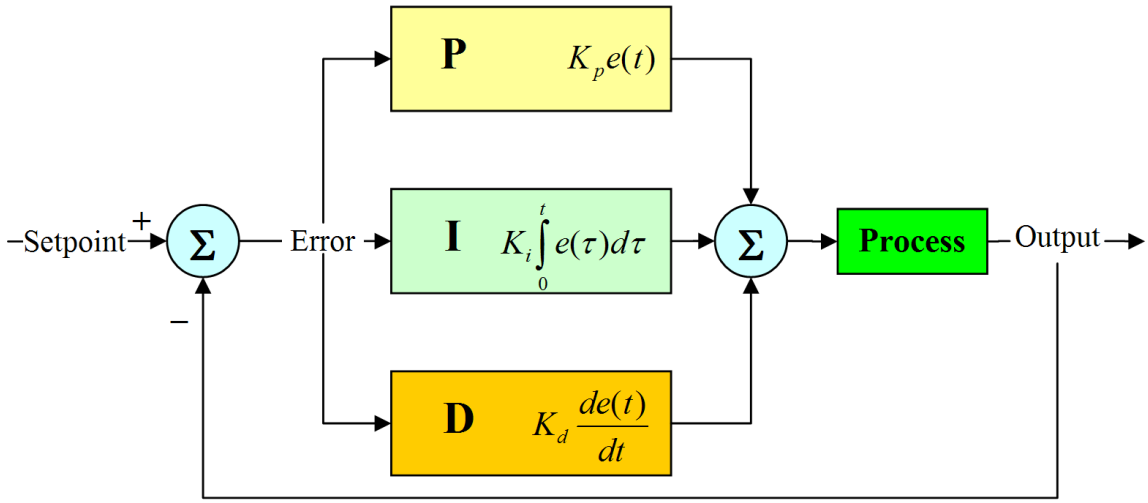
25 Bioreactor program on GAMBAS3 explained GUI. Picture extracted from "Normothermic Perfusion of an isolated liver: Control Software implementation, G. V., Lucía, 2014 [48]

The implemented control software for these parameters works thanks to different systems. For temperature, level and pH a simple activation/deactivation system is designed with a fast response in order to accurately control them with accuracy. Instead, for the flow & pressure control a more developed system is required due to the complexity of the problem. For flow & pressure control there are two different buttons in the GUI allowing to activate or deactivate the control system for the hepatic artery or the portal vein, so that they can be controlled independently.

The control system itself used for flow & pressure is a Proportional-Integral-Derivative (PID) loop feedback mechanism, and it is a new PID controller system that has been developed in order to work with the new pumps, which are different to the ones used previously by Dr. Francisco del Cañizo and Lucia Gullón. PID controllers continuously calculate an error value as the difference between the set value and the measured value. In this case, pressure set levels are preset as common pressure values for pig livers, being 12-17 mmHg in the portal vein and 75-80 mmHg in the hepatic artery. The set values for the flow will be provided by the user. Thus, the control system will increase or reduce the speed of the peristaltic pumps trying to meet the desired flow set values without surpassing the pressure set values. The real time information received from the different sensors, i.e., the flowmeters and the pressure transducers, will allow to obtain the required information for the feedback loop.

This type of double control where the pressure control is at a deeper level, overwriting the orders sent by the flow control system if needed, needs of two independent controllers, one for each parameter. Thus, each controller has to be adjusted independently in order to obtain a good control for them independently and when put together.

PID control mechanisms are feedback loops based on three components: a sensor a controller and an actuator. The sensors provide new data for the loop, providing information about the current state of the system. The controller calculates the adjustments that need to be done and sends the correct signal to the actuator, thus being the “brain” of the control system. Finally the actuator produces the adjustment ordered by the controller. With this three systems, a good PID control should reach as soon as possible a stability at the set values where the new data acquired will be the same, the controller should not need to make new adjustments and the actuator works always in the same way. In this case, the sensors would be the flowmeters and the pressure transducers, the controller would be the central computer program in GAMBAS 3 and the actuators would be the peristaltic pumps.



26 PID controller diagram

The PID feedback loop algorithm includes three different parameters that contribute to achieve the desired set value fast, and accurately. These parameters are the proportional, the derivative and the integral one, and the final action of the controller or adjustment is defined by the addition of each of these parameters, which are calculated each time independently from last received current values. Each of these parameters has a different function in how the PID works:

- **The proportional term** accounts for the present value of the error, this is, the real time difference between the set value and the current value.

It produces an output directly proportional to the current error, multiplying it by a constant K_p that is known as the proportional gain constant. A high proportional gain K_p will help the controller reach sooner the desired set value, but it can make the system unstable. A low proportional gain K_p in the other hand, will make the system less responsive, up to the point where it may be too small to adequately respond to system disturbances.

- **The integral term** is proportional to both the magnitude of the error and the duration of it. It is the sum of the current error over time and gives the adjustment that should have been made previously. It is multiplied by the integral gain K_i and added to the controller output. The function of the integral term is to accelerate the movement of the data towards the setpoint and eliminates the steady-state error that may be produced by the proportional term. However, a too low integral term will not have effect and a too high integral term since it uses past values can cause the value to overshoot the setpoint and oscillate around it.

- **The derivative term** of the error is calculated by finding the slope of the error over time and multiplying it by the derivative gain K_d. The derivative term predicts how the system will react and thus increases the stability of the system and settling time. However, because of the difficulty in tuning and in many times the low difference between a derivative term and not having a derivative term, many times derivative terms are not used, in which the PID controllers are also known as PI controllers. The need of low pass filters to adequately use the derivative term in many real-world applications also contributes to the low use of the derivative term.

$$u(t) = \underbrace{K_p e(t)}_{\text{Proportional}} + \underbrace{K_i \int_0^t e(\tau)d\tau}_{\text{Integral}} + \underbrace{K_d \frac{d}{dt} e(t)}_{\text{Derivative}}$$

27 PID general formula

In the PID present in central program the derivative term showed to not enhance the feedback control and thus, a simpler PI controller was developed, keeping the derivative gain K_d as zero.

In order to tune a PID controller there are several ways to do it, each one with its advantages and disadvantages. In this case, the PI control system was tuned manually since the response and setting time of the system was short, up to 3 minutes. The objectives of this PID control system were to obtain a stable control system with an accurate response that would not oscillate around the setpoint. In order to tune it, the derivative and integral terms were first set to zero and the proportional gain K_p was gradually changed until obtaining the smallest oscillation possible. After that, the integral gain K_i was adjusted in order to overcome the offset. Once these initial values were reached and attempt was made to include the derivative gain K_d, but it only destabilized the system, reason for which it was kept as 0.

This PI system takes into account thus the flow and pressure data to obtain a new velocity for the peristaltic pumps, which will in turn affect directly at the flow and pressure values obtained. Once the value for the velocity is obtained it is sent through the Xbee to the module in charge of the pumps, which sets the new velocitiy. Then the sensors receive again the data from the pressure transducers and the flowmeters and send the data back to the computer, which will run again the program and send a new order in regards to the velocity. Both the control for the hepatic artery and the portal vein can be active at the same time since they are

Lastly, the commands included for the module boxes, which are automatically used by the central program to communicate between the central program and the modular boxes are:

Module I:

- **MF1A:** gives back the information from the first flowmeter as MF1A*datastring
- **MF1B:** gives back the information from the second flowmeter as MF1B*datastring
- **MF1C:** gives back the information from both flowmeters as MF1A*datastring1*MF1B*datastring2
- **MB1P:** sets de speed for pump 1 at 0.
- **MB1VXXX:** sets de speed for pump 1 as the desired value. Currently input values go from 000 to 255 as they are the introduced by the central program. This command also returns pump speed as a percentage of maximum pump speed.
- **MB2P:** sets the speed for pump 2 at 0.
- **MB2VXXX:** As MB1V, sets de speed for pump 2 and also returns the pump speed as a percentage of the maximum one.
-

Module II:

- **MTAXXX**: sets de temperature for thermostat 1 activating the control system, being XXX the temperature in decimals and Celsius, for example, MTA250 would set the temperature to 25ºC
- **MTBXXX**: does the same as MTAXXX but for a possible second thermostat
- **MTI**: gives back the temperatures, de set temperatures, de state of the relays for control temperature at the moment, and the level. Everything as a string with the following structure: MT*temp1*temp2*temp3*temp4*set1*set2*Relay1*Relay2*Level
- **ML1I**: gives back the weights as ML1I*weight1*weight2*weight3
- **CDI**: gives back the data from the CDI as CDI1*datastring

4 Results & Conclusions

The presented design offers many advantages over current perfusion bioreactors, including size, ease of use, able to work without human surveillance for long periods of time, adaptability and control over different parameters. The modular design of the PBLB allows for easy implementation of new sensors, pumps or any other required machinery, as with the sensors required for adequate recellularization experiments. Also, the modular design allows for a more compact bioreactor.

The new modular boxes for the PBLB, more compact and integrating more functions than the old ones have been developed, built and tested, as well as some errors in the design and circuit problem has been corrected, such as incorrect amplifier design, failures in the wireless connection due to the metal box and others.

The new sensors for weight have been also built and tested, substituting the initially designed ones for more accurate sensors. They have shown to work correctly implemented with the new modular boxes, as well as other pre-fabricated sensors such as the flowmeters, the thermometers and the CDI.

The implemented software offers a user-friendly interface for easy data registration and parameter adjustment. As with the machinery and the physical case, the model with the central program and the external microcontrollers connected wirelessly allows for easy expansion of the software in order to control or register data from any new machinery as well as adding any new apps or data analysis that may be required for evaluation of the perfusion success.

Software for the modular boxes was developed successfully and correctly integrated with the existing central program, allowing for communication and correct function of the modular boxes wirelessly through the central program’s GUI in the computer.

5 Project Budget

5.1 Materials Costs

1 Permanent materials costs

Permanent	Price	Units	Cost for the project (€)
Thermotronic II (Thermostat)	1.240,00 €	1	1.240,00 €
Propylene container 500 tpye cage+lid (organ container)	20,00 €	1	20,00 €
Temperature sensors LM35	5,20 €	4	20,80 €
Alaris® CC Plus syringe pump	625,00 €	2	1.250,00 €
OEM Ultrasonic Flow Measuring System DIGIFLOW-EXT1	1.150,00 €	2	2.300,00 €
Ultrasonic Clamp-On Transducer	1.400,00 €	2	2.800,00 €
CDI Blood Parameter Monitoring System 500	28.156,00 €	1	28.156,00 €
VerderFlex EZ OEM pump	268,83 €	3	806,49 €
Masterflex L/S 7523	2.450,00 €	2	4.900,00 €
Masterflex L/S 4-roller cartridge pump head	620,00 €	2	1.240,00 €
Arduino Mega 2560	35,00 €	2	70,00 €
Xbee S1	30,00 €	3	90,00 €
Xbee 5V/3.3V adapter board	10,00 €	3	30,00 €
Pressure Sensors FSG15N1A	60,00 €	3	180,00 €
Boxes	15,00 €	2	30,00 €
Structure and gas blender	100,00 €	1	100,00 €
Cables and electronic devices	500,00 €	1	500,00 €
Computer and software	1.600,00 €	1	1.600,00 €
Total			**45.333,29 €**

2 Disposable materials costs

Disposable	Price	Units per perfusion	Cost for the project (€)
Oxygenator QUADROX-i Pediatric	3.100,00 €	1	3.100,00 €
VHK 31000 Venous Cardiotomy Hardshell Reservoir	100,00 €	1	100,00 €
Renaflo® II HF Minifilter TM Plus	50,00 €	1	50,00 €
TrueWave Disposable Pressure Transducers	20,00 €	3	60,00 €
CDI disposable sensors	130,00 €	1	130,00 €
Conduits and scaffolding	100,00 €	2	200,00 €
Total			**3.640,00 €**

5.2 Perfusion Costs

3 Perfusion costs

For pig liver normothermic perfusion:	Price	Units	Total cost (€)
Pig "Minipig"	320,00 €	1	320,00 €
Operating room and animal cares	3.100,00 €	1	3.100,00 €
Pathological anatomy technician	50,00 €	1	50,00 €
Replacement liquid and other infusions	100,00 €	1	100,00 €
Biochemical analysis	100,00 €	1	100,00 €
Perfusion experiment and results analysis	24.8 €/h	34 h	843,20 €
Total			**4.513,20 €**

5.3 Human Resources Costs

4 Human resources cost

Personnel:	Salary / hour (€)	Dedication (h)	Total cost (€)
Research and project investigation	24,80 €	90	2.232,00 €
Building and development	24,80 €	160	3.968,00 €
Software development: tests and optimization	24,80 €	120	2.976,00 €
Total			**9.176,00 €**

5.4 Total Costs

5 Total bioreactor setup costs

Total	Total cost (€)
Permanent parts	48.333,29 €
Disposable	3.640,00 €
Human resources	9.176,00 €
Total bioreactor price:	**61.149,29 €**

6 Future perspectives

The PBLB presented in this work is only the minimum base required to start adequately experimenting with liver recellularization. The addition of other several sensors could improve the PBLB functionality as well as making it able to be used in other environments, or in different experiments.

Although the idea of bioreactors is very old, these devices are still very new and there is plenty of room to upgrade. It would be interesting, if not required to add many new sensors assess cellular activity and performance. Some of these new sensors that are required to adequate recellularization experiments but were not added due their cost and the budget, as well as the time required to properly set them up are mainly the sensors for ammonium, glucose and lactate. Nevertheless, having sensors for other metabolites and nutrients, or in general excretions of the generated catabolism byproducts would be very useful for the research.

Once a bioreactor with all these sensors and other actuators is made, there would be still plenty of room to upgrade, from studying better pumps to cause less damage to the medium, to studying the flow and eliminating any disturbances in order to emulate the laminar flow that happens in the human body.

Other thing that would be interesting to study would be materials used, as proteins and molecules in the medium may be adsorbed by some material in the bioreactor, thus causing loss of the wanted protein.

We can conclude that even if it is a relatively new field, bioreactors have plenty of room to evolve and upgrade, and they will have a huge impact in the organ problem and in research in general.

7 Annexes

7.1 II: Software

7.1.1 Main Program

Made by Dr. Juan F. Del Cañizo and Lucía Gullón, with only some parameters changed.

Temperature control:

```
' CONTROLADOR DE TEMPERATURAS

    Try Print #COMPort1, "MTI"; Chr$(13); Chr$(10)          'Devuelve el estado de las
temperaturas y los relés
    If Error Then
      TextMT.text = "Problema en la obtención de datos del termostato..."
    Endif
    If SetMTempA = SetMTemp1 Then
      DAControlTemp.Background = Color.Green
    Else
      DAControlTemp.Background = Color.Red
      cad = CInt(SetMTempA * 10)
      Print #COMPort1, "MTA"; Format(Str(cad), "000"); Chr$(13); Chr$(10)
    Endif
    Wait 0.03

Public Sub BControlNivel_Click()
  TimerNivel.Delay = 2000        'El control de la temperatura se realiza cada 2 segundos
   If TimerNivel.Enabled = False Then
 'Cuando se activa el control, el botón del mismo se vuelve verde, si no, rojo.
   DAControlNivel.Background = Color.Green
   TimerNivel.Enabled = True
  Else
   DAControlNivel.Background = Color.Red
   TimerNivel.Enabled = False
   SetVelBV1 = 0
  Endif

End
```

Section for pH control:

```
Public Sub TimerpH_Timer()    'Control de pH

 If TypeOf(Val(SetpH.Text)) = gb.Float Then ThpH = Val(SetpH.Text)
         'Lee el valor asignado al set para compararlo con la medida actual de pH

 If pHArt < ThpH Then

          'El control comienza si el pH está por debajo del set impuesto
   dif = ThpH - pHArt '
   seg = dif / 0.03

         'Segundos que la bomba estará activa en función de la distancia al pH ideal y al
volumen de carbónico expulsado por la bomba durante 1 segundo (experimental)

   seg = Round(seg, 0)                    'Redondeo del valor de los segundos
   SetVelBV3 = 200                        'Bomba en marcha a 200rpm
    Wait seg                              'Tiempo que está activada la bomba
    SetVelBV3 = 0                         'Para la bomba
    TimerpH.Delay = 120000 * seg
          'La bomba estará parada un tiempo proporcional al tiempo que estuvo encendida
(estimado, approx)
  Endif                        'El delay anterior (60000) que se tenía para el control, varía
End

Public Sub BControlpH_Click()              'Activa o desactiva el control de pH
  TimerpH.Delay = 60000                    'El control de pH se realiza cada minuto

    If TimerpH.Enabled = False Then
          'Una vez que se activa el control, el color del mismo se vuelve verde, si no, rojo

    DAControlpH.Background = Color.Green
    TimerpH.Enabled = True
   Else
    DAControlpH.Background = Color.Red
    TimerpH.Enabled = False

  Endif

End
```

Section for Flow and Pressure Control (pump velocity):

```
Public Sub TimerControlHep_Timer()
''**Variables para el control de la arteria hepatica**
'control de la presion
     Dim i As Integer
     Dim h As Integer
     Dim ErrPrHPro As Integer
     Dim ErrPrHDif As Integer
     Dim ErrPrHInt As Integer
     Dim CoefHP As Float
     Dim CoefHD As Float
     Dim CoefHI As Float
     Dim MedPrH As Float

'control del flujo
     Dim ErrFHPro As Integer
     Dim ErrFHDif As Integer
     Dim ErrFHInt As Integer
     Dim CoefFHP As Float
     Dim CoefFHD As Float
     Dim CoefFHI As Float
     Dim MedF2H As Float

'******************Control Hepatica*************************

CoefFHP = 0.05
CoefFHD = 0 '.0042 '.01 '.0001
CoefFHI = 0.0001 '.001 '.001

CoefHP = 0.02 '0.1 'Coeficientes PID
CoefHD = 0 '0.1
CoefHI = 0.001 '.003 '0.02

For i = 2 To 10
     ArrayPm2[i - 1] = ArrayPm2[i]
Next
ArrayPm2[10] = Pm2

For i = 1 To 10
     MedPrH = MedPrH + ArrayPm2[i]
Next
     MedPrH = MedPrH / 10
'Print "medH", MedPrH

For i = 2 To 10
ArrayF2[i - 1] = ArrayF2[i]
Next
ArrayF2[10] = Flujo2
For i = 1 To 10
```

```
MedF2H = MedF2H + ArrF2[i]
Next
MedF2H = MedF2H / 10
'Print "media", MedF1P

ErrFHInt = MedF2H - SetControlF2
ErrFHPro = Flujo2 - SetControlF2
ErrFHDif = Flujo2 - F2Ant

Print F2Ant
ErrPrHPro = Pm2 - SetControlPm2 'Error proporcional
ErrPrHDif = Pm2 - Pm2Ant 'Error diferencial
ErrPrHInt = MedPrH - SetControlPm2 'Error integral
Pm2Ant = Pm2

If MedPrH < SetControlPm2 Then

'* CONTROL PID DEL FLUJO DE LA ARTERIA HEPATICA (Flujo2);
SetControlF2 es el flujo que queremos en la Hepatica *

VelControlHep = VelControlHep - (CoefFHP * ErrFHPro) - (CoefFHD
* ErrFHDif) - (CoefFHI * ErrFHInt)
Print VelControlHep, CoefFHP * ErrFHPro, CoefFHD * ErrFHDif,
CoefFHI * ErrFHInt

Debug "Control flujo hepática"

Else '* CONTROL PID DE LA PRESION DE ARTERIA HEPATICA (Pm2);
SetControlPm2 es la presion que queremos en la Hepatica *
VelControlHep = VelControlHep - (CoefHP * ErrPrHPro) - (CoefHD
* ErrPrHDif) - (CoefHI * ErrPrHInt) 'Control de la velocidad

Debug "Control presión hepática"

Endif
If VelControlHep < 0 Then VelControlHep = 0

SetVelB2 = VelControlHep
If SetVelB2 > 50 Then SetVelB2 = 50 'Límite de velocidad
máxima: 5000rpm
If SetVelB2 < 0 Then SetVelB2 = 0

'Límite de velocidad mínima: 0rpm

'**************** TERMINA Control Hepatica*****************

End

Public Sub TimerControlPorta_Timer()
''**Variables para el control de la vena porta**
```

```
'control de la presion
Dim i As Integer
Dim h As Integer
Dim ErrPrPPro As Integer
Dim ErrPrPDif As Integer
Dim ErrPrPInt As Integer
Dim CoefPP As Float
Dim CoefPD As Float
Dim CoefPI As Float
Dim MedPrP As Float

'control del flujo

Dim ErrFPPro As Integer
Dim ErrFPDif As Integer
Dim ErrFPInt As Integer
Dim CoefFPP As Float
Dim CoefFPD As Float
Dim CoefFPI As Float
Dim MedF1P As Float

''**** CONTROL POR FLUJO DENTRO DE UNOS LÍMITES DE PRESIÓN ****

'Se controlan independientemente el flujo de la arteria y de la
porta. Es decir, la velocidad de las bombas centrífugas estará
en función del set del flujo que se haya establecido.
'Pero el control por flujo dejará de estar activo cuando la
presión supere los límites establecidos como set de presion. En
ese momento se activará el control de presión.
'Éste se encargará de que la presión no sobrepase estos límites
al variar las vueltas de las bombas. Los flujos cambiarán y ya
no serán los establecidos en el set previamente.

' ********************** Control Porta*******************

CoefFPP = 0.002
CoefFPD = 0 '0.0001
CoefFPI = 0.0001

CoefPP = 0.03 '0.1 'Coeficientes PID
CoefPD = 0 '0.1
CoefPI = 0.01 '0.02

For i = 2 To 10
ArrayPm1[i - 1] = ArrayPm1[i]
'Print "arrP", ArrayPm1[i]
Next
ArrayPm1[10] = Pm1
```

```
For i = 1 To 10
MedPrP = MedPrP + ArrayPm1[i]
Next
MedPrP = MedPrP / 10
' Print "med", MedPrP
For i = 2 To 10
ArrayF1[i - 1] = ArrayF1[i]
Next
ArrayF1[10] = Flujo1
For i = 1 To 10
MedF1P = MedF1P + ArrF1[i]
Next
MedF1P = MedF1P / 10
'Print "media", MedF1P

     ErrFPPro = Flujo1 - SetControlF1
     ErrFPDif = Flujo1 - F1Ant
     ErrFPInt = MedF1P - SetControlF1

     ErrPrPPro = Pm1 - SetControlPm1    'Error proporcional
     ErrPrPDif = Pm1 - Pm1Ant           'Error diferencial
     ErrPrPInt = MedPrP - SetControlPm1 'Error integral
     Pm1Ant = Pm1

If MedPrP < SetControlPm1 Then
     '* CONTROL PID DEL FLUJO DE LA VENA PORTA (Flujo1);
SetControlF1 es el flujo que queremos en la porta *

VelControlPorta = VelControlPorta - (CoefFPP * ErrFPPro) -
(CoefFPD * ErrFPDif) - (CoefFPI * ErrFPInt)
' Print VelControlPorta, CoefFPP * ErrFPPro, CoefFPD *
ErrFPDif, CoefFPI * ErrFPInt
Debug "Control flujo porta"

Else '* CONTROL PID DE LA PRESION DE VENA PORTA (Pm1);
SetControlPm1 es la presion que queremos en la Porta *
VelControlPorta = VelControlPorta - (CoefPP * ErrPrPPro) -
(CoefPD * ErrPrPDif) - (CoefPI * ErrPrPInt) 'Control de la
velocidad

Debug "Control presión porta"
Endif

If VelControlPorta < 0 Then VelControlPorta = 0
SetVelB1 = VelControlPorta
If SetVelB1 > 50 Then SetVelB1 = 50 'Límite de velocidad
máxima: 5000rpm
If SetVelB1 < 0 Then SetVelB1 = 0
'Límite de velocidad mínima: 0rpm

'************* TERMINA Control Porta *****************
```

```
End
```

7.1.2 Module I Program:

```
/*
 Recibe una cadena por la linea serie principal conectada al USB
 La manda por la linea serie 1
 La recibe por la linea serie 2
 La vuelve a mandar por la linea serie principal.
*/

char ByteR;               // Byte leido por la linea serie
String StrS = "";         // String para almecenar los datos entrantes
String StrS1 = "";        // String para almecenar los datos entrantes por ser1
String StrF1 = "";        // String para almecenar los datos del Flujometro 1
String StrF2 = "";        // String para almecenar los datos del Flujometro 2
int Pump1 = 2;
int Pump2 = 3;
int vel1 = 0;
int vel2 = 0;
int a=0;
int b=0;
int c=0;
boolean SFin = false;      // Bandera de string completa

void setup() {

  StrS.reserve(100);  // Espacio de memoria que se reserva a las strings
  StrS1.reserve(100);
  StrF1.reserve(60);
  StrF2.reserve(60);

  // initialize serial ports:
  Serial3.begin(9600);          // Baudios para la comunicación
  Serial1.begin(38400);
  Serial2.begin(38400);

  Serial3.println("Comienzo");  // Mensaje de inicio para comprobar la conexión.
}

void S1GetStr(void){
  char sIn = '*';
  StrF1 = "";
```

```
  if(Serial1.available()){
    while (StrF1.length() < 80){ // Longitud arbitraria (ésta debe ser >54 pero menor de 60 por los bytes de memoria)
      sIn = Serial1.read();   // Puede que sea >60 porque lee varias líneas (se asegura con un valor de 80)
      StrF1 += sIn;
    }
  }
  StrF1= StrF1.substring(0, StrF1.indexOf('\n')); // Lee la línea desde el primer caracter hasta el enter, sin incluir este último.
  if(StrF1.length() != 54){ // El número de caracteres máximos que debe tener el código que devuelve.
    delay (100);            // Espera 100 milisegundos
    if(Serial1.available()) S1GetStr();
  }
}

void S2GetStr(void){
  char sIn = '*';
  StrF2 = "";
  if(Serial2.available()){
    while (StrF2.length() < 80){
      sIn = Serial2.read();
      StrF2 += sIn;
    }
  }
  StrF2= StrF2.substring(0, StrF2.indexOf('\n'));
  if(StrF2.length() != 54){
    delay (100);
    if(Serial2.available()) S2GetStr();
  }
}

void loop() {
  int n;
  // ****************************** LEE LA LINEA SERIE 0 ******************************
  if (Serial3.available() > 0) {
    ByteR = (char)Serial3.read();      // lee el byte entrante:
    StrS += ByteR;
    if (ByteR == '\n') SFin = true;
  }
  if (SFin == true){          // Cadena Leida
    //    Inicio comandos para los flujómetros
    if ( StrS[0]== 'M' && StrS[1]== 'F' && StrS[2]== '1') {
      switch (StrS[3]) {
```

```
      case 'A':
        S1GetStr();
        Serial3.print("MF1A*");
        Serial3.println(StrF1);  // Imprime la información de cada flujómetro por separado en
función del comando (A,B o C).
        StrF1="";
        break;
      case 'B':
        S2GetStr();
        Serial3.print("MF1B*");
        Serial3.println(StrF2);
        StrF2="";
        break;
      case 'C':
        S1GetStr();
        Serial3.print("MF1A*");
        Serial3.println(StrF1);    // Para el comando MF1D se imprime la información de los
tres flujómetros.
        S2GetStr();
        Serial3.print("MF1B*");
        Serial3.println(StrF2);
        break;
      default:
        Serial3.println("MF1 Cmd Invalido");


      }
    }
    //  Inicio comandos para las bombas.
    if (StrS[0]== 'M' && StrS[1]== 'B')  {
      switch (StrS[2]) {
        case '1':
          if(StrS[3]=='P') vel1=0;
          if(StrS[3]=='V'){
          a = (int(StrS[4])-48);
          b = (int(StrS[5])-48);
          c = (int(StrS[6])-48);
          vel1= (a*100)+(b*10)+c;
          if (vel1>255) vel1=255;
          if (vel1<0) vel1=0;
          analogWrite(Pump1, vel1);
          Serial3.print("MB1V*");
          Serial3.print(vel1*100/255);
          Serial3.print("%");
          Serial3.println("*");
```

```
        }
       break;
       case '2':
        if(StrS[3]=='P') vel2=0;
        if(StrS[3]=='V'){
        a = (int(StrS[4])-48);
        b = (int(StrS[5])-48);
        c = (int(StrS[6])-48);
        vel2= (a*100)+(b*10)+c;
        if (vel2>255) vel2=255;
        if (vel2<0) vel2=0;
        analogWrite(Pump2, vel2);
        Serial3.print("MB2V*");
        Serial3.print(vel2*100/255);
        Serial3.print("%");
        Serial3.println("*");
        }
       break;
      default:
       Serial3.println("MV Cmd Invalido");
     }
   }
   StrS="";
   StrS1="";
   SFin = false;
  }
}
```

7.1.3 Module II Program:

```
/*
* Adquisición de datos del Biorreactor. Trata:
*    - Temperaturas
 *    - Pesos
 *    - Nivel
 *    - CDI

* Utiliza un Arduino Mega.
 * Linea serie a 9600 baudios para comunicar con el XBee. Linea serie a 38400 baudios para comunicar con el CDI.
 *
 * Tratamos de que el control de temperatura lo haga el arduino y el PC sólo tendrá que enviar el set de temperatura.
 * Para ello usaremos los reles.
```

```
*
_____________________________________________________________________
*
* COMANDOS:
*
* MTAxxx      Pone el set de temperatura 1 en decimas de grado (255 = 25.5 ºC)
* MTBxxx      Pone el set de temperatura 2 en decimas de grado (246 = 24.6 ºC)
* MTI         Devuelve las temperaturas y los sets y el estado de los reles
* CDI       Devuelve la información del CDI
* ML1I        Devuelve los pesos.
*
*
* FORMATO DEVUELTO:
*
* MT*Temp1*Temp2*Temp3*Temp4*SetT1*SetT2*SR1*SR2*Nivel\n
* ML1I*Peso1*Peso2*Peso3\n
* CDI1*StrCDI2\n (String del CDI)
*/

long sumTemp1;        // Sumas de las temperaturas (reseteadas cada segundo) Van
sumando las nuevas mediciones de los pines.
long sumTemp2;
long sumTemp3;
long sumTemp4;
float Temp1;          // Temperatura final que será enviada cada segundo.
float Temp2;
float Temp3;
float Temp4;
float SetT1;          // Temperaturas predefinidas.
float SetT2;

int Peso1;   // Pesos enviados cada segundo.
int Peso2;
int Peso3;
long sumPeso1;   // Suma de los pesos, reseteados cada segundo.
long sumPeso2;
long sumPeso3;
boolean Nivel;

const int NivelPin = 10;       // Entrada del Nivel.
const int REL1 = 6;         // Salida para el Rele 1.
const int REL2 = 7;         // Salida para el Rele 2.
int SR1;             // Estado del rele 1
int SR2;             // Estado del rele 2
```

```
unsigned long tact;
unsigned long tant;
unsigned long paso;
unsigned long contmseg;    // contador de milisegundos
unsigned long contcseg;    // contador de centesimas de segundo
unsigned long contdseg;    // contador de decimas de segundo
unsigned long contseg;     // contador de segundos

char ByteR;               // Byte leido por la linea serie
String StrS = "";         // String para almecenar los datos entrantes
boolean SFin;      // Bandera de string completa
char ByteRCDI;            // Byte leído por la linea serie (proveniente del CDI)
String StrCDI1 = "";      // String para almacenar los datos del CDI
String StrCDI2 = "";      // String para mandar los datos del CDI
boolean SFinCDI;   // Bandera de string completa para el CDI

// ***************              Rutina de inicializacion          **************



void setup() {

pinMode (NivelPin, INPUT);
pinMode (REL1, OUTPUT);
pinMode (REL2, OUTPUT);

//Serial.begin (9600); // Para pruebas con el ordenador.
Serial3.begin(9600); // Conector al XBee
Serial1.begin(38400); // Conector al RS-232 (CDI)

StrS.reserve(100);
StrCDI1.reserve(100);
StrCDI2.reserve(100);

contmseg = 0;
tant = micros();
 SetT1 = 24.0;
 SetT2 = 0.0;
}

void loop() {

  int a,b,c,d;
```

```
tact = micros();
paso = tact - tant;

if(paso >= 1000){      // ***** Cada milisegundo **********************************
  contmseg++;

// ************************** LEE LA LINEA SERIE DEL XBEE ****************************

  if (Serial3.available() > 0) {
    ByteR = (char)Serial3.read();      // lee el byte entrante:
    StrS += ByteR;
    if (ByteR == '\n') SFin = true;
  }
  if (SFin == true){          // Cadena Leida
    //Serial.print (" * ");
      if ( StrS[0]== 'M' && StrS[1]== 'L' && StrS[2]=='1' && StrS[3]=='I'){
        Serial3.print ("ML1I*");
        Serial3.print (Peso1);
        Serial3.print ("*");
        Serial3.print (Peso2);
        Serial3.print ("*");
        Serial3.println (Peso3);
        }
      else if (StrS[0]=='C' && StrS[1]=='D' && StrS[2]=='I') {
        Serial3.print("CDI1*");
        Serial3.println (StrCDI2);
        }
      else if ( StrS[0]=='M' && StrS[1]=='T') {
        switch (StrS[2]){
          case 'A':
            a=(int(StrS[3])-48)*100; b=(int(StrS[4])-48)*10; d=(int(StrS[5])-48);
            c=a+b+d;
            SetT1 = c / 10.0;
            if(SetT1 > 50) SetT1 = 50; if(SetT1 < 0) SetT1 = 0;
          break;
          case 'B':
            a=(int(StrS[3])-48)*100; b=(int(StrS[4])-48)*10; d=(int(StrS[5])-48);
            c=a+b+d;
            SetT2 = c / 10.0;
            if(SetT2 > 50) SetT2 = 50; if(SetT2 < 0) SetT2 = 0;
          break;
          case 'I':
            Serial3.print("MT*");
            Serial3.print(Temp1); Serial3.print("*");
```

```
            Serial3.print(Temp2); Serial3.print("*");
            Serial3.print(Temp3); Serial3.print("*");
            Serial3.print(Temp4); Serial3.print("*");
            Serial3.print(SetT1); Serial3.print("*");
            Serial3.print(SetT2); Serial3.print("*");
            Serial3.print(SR1);   Serial3.print("*");
            Serial3.print(SR2);   Serial3.print("*");
            Serial3.println(Nivel);
          break;
        }
       }
       StrS="";
       SFin = false;
   }
 // ************************* LEE LA LINEA SERIE DEL CDI ***************************

 if (Serial1.available() > 0); {
   ByteRCDI = (char)Serial1.read();     // Lee el byte entrante.
   StrCDI1 += ByteRCDI;
   if (ByteRCDI == '\n') SFinCDI = true;
 }
 if (SFinCDI == true) {
   StrCDI2 = StrCDI1;
   StrCDI1 = "";
   SFinCDI = false;
 }
 // ************************* FIN LECTURA DE LINEAS SERIE ***************************

  paso = 0;
  tant = micros();
 }

// ****************************** Cada centesima de segundo *******************
 if(contmseg >= 10){ // Cada centesima de segundo
   contcseg++;
  Nivel = digitalRead (NivelPin); // Lee el nivel del reservorio.

  sumPeso1 = sumPeso1 + analogRead (A0);    // Adquisición de datos de los pesos.
  sumPeso2 = sumPeso2 + analogRead (A1);
  sumPeso3 = sumPeso3 + analogRead (A2);

  sumTemp1 = sumTemp1 + analogRead (A3);     // Adquisición de datos de las
temperaturas.
  sumTemp2 = sumTemp2 + analogRead (A4);
```

```
  sumTemp3 = sumTemp3 + analogRead (A5);
  sumTemp4 = sumTemp4 + analogRead (A6);

   contmseg = 0;
 }

// ****************************   Cada decima de segundo  ******************
 if(contcseg >= 10){
   contdseg++;
   contcseg = 0;
 }

// ****************************        Cada segundo       ******************
 if(contdseg >= 10){
//    contseg++;
   Temp1 = (sumTemp1 / 100) * 500 /1024;  // Divido por 100 para obtener los grados.
   Temp2 = (sumTemp2 / 100) * 500 /1024;
   Temp3 = (sumTemp3 / 100) * 500 /1024;
   Temp4 = (sumTemp4 / 100) * 500 /1024;

// Control de los relés para activar el termostato.
   if (Temp1 > SetT1 + 0.2){ digitalWrite(REL1, LOW); SR1 = 0;}
   if (Temp1 < SetT1 - 0.2){ digitalWrite(REL1, HIGH);  SR1 = 1;}
   if (SetT1 == 0) {digitalWrite(REL1, LOW); SR1 = 0;}
   if (Temp2 > SetT2 + 0.2){ digitalWrite(REL1, LOW); SR1 = 0;}
   if (Temp2 < SetT2 - 0.2){ digitalWrite(REL1, HIGH);  SR1 = 1;}
   if (SetT2 == 0) { digitalWrite(REL1, LOW); SR1 = 0;}

   sumTemp1 = 0;
   sumTemp2 = 0;
   sumTemp3 = 0;
   sumTemp4 = 0;

   // Hasta aquí hace todos los cálculos para las temperaturas, así como maneja los reles
y reinicia los sumatorios.

   // Peso1 = (sumPeso1 /100) *2.6455 - 365.08;
   Peso1 = (sumPeso1/100)/1024    - 130.2528;
   if (Peso1 < 0) Peso1 = 0;
   // Peso2 = (sumPeso2 /100) *2.6455 - 365.08;
   Peso2 = sumPeso2/100;
   if (Peso2 < 0) Peso2 = 0;
   // Peso3 = (sumPeso3 /100) *2.6455 - 365.08;
   Peso3 = sumPeso3/100           - 266;
```

```
    if (Peso3 < 0) Peso3 = 0;

    sumPeso1 = 0;
    sumPeso2 = 0;
    sumPeso3 = 0;

    contdseg = 0;
  }
}
```

8 Bibliography & References

1. IUPAC. *Compendium of Chemical Terminology*. *IUPAC Compendium of Chemical Terminology* (2014). doi:10.1351/goldbook.I03352

2. Krogh, A. The number and distribution of capillaries in muscles with calculations of the oxygen pressure head necessary for supplying the tissue. *J. Physiol.* **52,** 409–415 (1919).

3. Le Gallois, M., Nancrede, N. C. (Trans) & Nancrede, J. G. (Trans). Experiments on the principle of life. in *Experiments on the principle of life, and particularly on the principle of the motions of the heart, and on the seat of this principle….* 23–228 (M Thomas, 1813). doi:10.1037/13894-001

4. Lindbergh, C. A. AN APPARATUS FOR THE CULTURE OF WHOLE ORGANS. *J. Exp. Med.* **62,** 409 (1935).

5. Carrel, A. The culture of whole organs. *J. Exp. Med.* **65,** 515 (1937).

6. Langer, R. M. Vladimir P. Demikhov, a Pioneer of Organ Transplantation. *Transplant. Proc.* **43,** 1221–1222 (2011).

7. Lee, W. *et al.* Multi-layered culture of human skin fibroblasts and keratinocytes through three-dimensional freeform fabrication. *Biomaterials* **30,** 1587–1595 (2009).

8. Haglund, L. *et al.* Development of a Bioreactor for Axially Loaded Intervertebral Disc Organ Culture. *Tissue Eng. Part C Methods* **17,** 1011–1019 (2011).

9. Saidi, R. F. Changing pattern of organ donation and utilization in the USA. *Int. J. organ Transplant. Med.* **3,** 149–56 (2012).

10. Saidi, R. F. & Hejazii Kenari, S. K. Challenges of organ shortage for transplantation: solutions and opportunities. *Int. J. organ Transplant. Med.* **5,** 87–96 (2014).

11. Locke, F. S. & Rosenheim, O. Contributions to the physiology of the isolated heart: The consumption of dextrose by mammalian cardiac muscle. *J. Physiol.* **36,** 205–20 (1907).

12. CALNE, R. Y., PEGG, D. E., PRYSE-DAVIES, J. & BROWN, F. L. Renal preservation by ice-cooling: an experimental study relating kidney transplantation from cadavers. *Br. Med. J.* **2,** 651–5 (1963).

13. Collins, G. M., Bravo-Shugarman, M. & Terasaki, P. I. Kidney preservation for transportation. Initial perfusion and 30 hours' ice storage. *Lancet (London, England)* **2,** 1219–22 (1969).

14. Guibert, E. E. *et al.* Organ Preservation: Current Concepts and New Strategies for the Next Decade. *Transfus. Med. Hemother.* **38,** 125–142 (2011).

15. HUMPHRIES, A. L., RUSSELL, R., GREGORY, J., CARTER, R. H. & MORETZ, W. H. Hypothermic perfusion of the canine kidney for 48 hours followed by reimplantation. *Am. Surg.* **30,** 748–52 (1964).

16. Humphries, A. L., Russell, R., Stoddard, L. D. & Moretz, W. H. Successful five-day kidney preservation. Perfusion with hypothermic, diluted plasma. *Invest. Urol.* **5,** 609–18 (1968).

17. Howden, B. *et al.* Studies in renal preservation using a rat kidney transplant model: II. The effect of reflushing with citrate. *Transplantation* **37,** 52–4 (1984).

18. Testa, G. *et al.* Histidine-tryptophan-ketoglutarate versus University of Wisconsin solution in living donor liver transplantation: results of a prospective study. *Liver Transpl.* **9,** 822–6 (2003).

19. Mangus, R. S. *et al.* Comparison of histidine-tryptophan-ketoglutarate solution and University of Wisconsin solution in extended criteria liver donors. *Liver Transpl.* **14,** 365–73 (2008).

20. Cavallari, A. *et al.* A multicenter pilot prospective study comparing Celsior and University of Wisconsin preserving solutions for use in liver transplantation. *Liver Transpl.* **9,** 814–21 (2003).

21. Ueda, Y. *et al.* The UW solution for canine kidney preservation. Its specific effect on renal hemodynamics and microvasculature. *Transplantation* **48,** 913–8 (1989).

22. Rayya, F. *et al.* Comparison of histidine-tryptophan-ketoglutarate solution and University of Wisconsin solution in adult liver transplantation. *Transplant. Proc.* **40,** 891–4 (2008).

23. Franco-Gou, R. *et al.* New preservation strategies for preventing liver grafts against cold ischemia reperfusion injury. *J. Gastroenterol. Hepatol.* **22,** 1120–6 (2007).

24. Groenewoud, A. F. & Thorogood, J. Current status of the Eurotransplant randomized multicenter study comparing kidney graft preservation with histidine-tryptophan-ketogluterate, University of Wisconsin, and Euro-Collins solutions. The HTK Study Group. *Transplant. Proc.* **25,** 1582–5 (1993).

25. Suszynski, T. M. *et al.* Persufflation (or gaseous oxygen perfusion) as a method of organ preservation. *Cryobiology* **64,** 125–43 (2012).

26. Benichou, J. *et al.* Canine and human liver preservation for 6 to 18 hr by

cold infusion. *Transplantation* **24,** 407–11 (1977).

27. Badger, I. L. *et al.* Human hepatic preservation using Marshall's solution and University of Wisconsin solution in a controlled, prospective trial. *Transplant. Proc.* **22,** 2183–4 (1990).

28. Kalayoglu, M. *et al.* Extended preservation of the liver for clinical transplantation. *Lancet (London, England)* **1,** 617–9 (1988).

29. Changani, K. K. *et al.* Non-invasive assessment of ATP regeneration potential of the preserved donor liver. A 31P MRS study in pig liver. *J. Hepatol.* **26,** 336–42 (1997).

30. Brockmann, J. *et al.* Normothermic perfusion: a new paradigm for organ preservation. *Ann. Surg.* **250,** 1–6 (2009).

31. Imber, C. J. *et al.* Advantages of normothermic perfusion over cold storage in liver preservation. *Transplantation* **73,** 701–9 (2002).

32. Schön, M. R. *et al.* Liver transplantation after organ preservation with normothermic extracorporeal perfusion. *Ann. Surg.* **233,** 114–23 (2001).

33. Bruinsma, B. G. *et al.* Subnormothermic Machine Perfusion for *Ex Vivo* Preservation and Recovery of the Human Liver for Transplantation. *Am. J. Transplant.* **14,** 1400–1409 (2014).

34. Tanaka, K. & Ogura, Y. 'Small-for-size graft' and 'small-for-size syndrome' in living donor liver transplantation. *Yonsei Medical Journal* **45,** 1089–1094 (2004).

35. Dahm, F., Georgiev, P. & Clavien, P.-A. Small-for-size syndrome after partial liver transplantation: definition, mechanisms of disease and clinical implications. *Am. J. Transplant* **5,** 2605–10 (2005).

36. Asencio, J. M., Vaquero, J., Olmedilla, L. & García Sabrido, J. L. 'Small-for-flow'; syndrome: shifting the 'size' paradigm. *Med. Hypotheses* **80,** 573–7 (2013).

37. Gilbert, T. W., Sellaro, T. L. & Badylak, S. F. Decellularization of tissues and organs. *Biomaterials* **27,** 3675–3683 (2006).

38. Crapo, P. M., Gilbert, T. W. & Badylak, S. F. An overview of tissue and whole organ decellularization processes. *Biomaterials* **32,** 3233–3243 (2011).

39. Kasimir, M. T. *et al.* Comparison of different decellularization procedures of porcine heart valves. *Int. J. Artif. Organs* **26,** 421–7 (2003).

40. Badylak, S. F., Weiss, D. J., Caplan, A. & Macchiarini, P. Engineered whole organs and complex tissues. *Lancet* **379,** 943–952 (2012).

41. Abdel-Misih, S. R. Z. & Bloomston, M. Liver anatomy. *Surg. Clin. North Am.* **90,** 643–53 (2010).

42. Michael H, R. & Wojciech, P. *Histology: A Text and Atlas*. *Journal of Chemical Information and Modeling* **53,** (2013).

43. Carroll, R. G. & Abdel-Rahman, A. A. *Textbook of Medical Physiology*. *The Comprehensive Pharmacology Reference* (Guyton and Hall, 2011). doi:10.1016/B978-008055232-3.60298-3

44. Tortora, G. J. & Derrickson, B. H. in *Principles of anatomy and physiology* 4 (2008).

45. MAQUET. QUADROX-i Oxygenators & VHK reservoirs Datasheet. (2015).

46. Minntech. Renaflo II Hemofilters Datasheet. (2010).

47. Cole-Parmer. *Operating Manual: L / S ® Pump Drives*. **11,**

48. Gullón Vega, L., Jorcano Noval, L. & Del Cañizo, J. F. Normothermic perfusion of an isolated liver: Control Software Implementation. (Carlos III of Madrid, 2014).