Mahlet Alemayehu Zewde

Knee Angle Measurement System for Knee Implants

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Electronics
Thesis
October 2015

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Mahlet Zewde Knee Angle Measurement System for Knee Implants 39 pages + 8 appendices 5 October 2015
Degree	Bachelor of Engineering
Degree Programme	Electronics Engineering
Specialisation option	
Instructor(s)	Dr Eng Patrick Peligrims, Project Manager Matti Fischer, Principal Lecturer

The goal of this thesis project is building a knee angle measurement system for knee implants. Measuring the knee angle can be the new postoperative diagnosis method which assists in early diagnosis of total knee implants (TKAs).

This system is used to acquire magnetic field data from magneto sensors and process it using a microcontroller. Then, this data is transmitted wirelessly in to a receiver system (data logger) which logs the received data into a computer or USB.

The work was carried out in two phases. First, the work of previous students was modified and optimized. After this the analog magneto sensors were replaced with digital sensors, and the analog frontend ADS1296 (AFE) was omitted, resulting in a smaller and cheaper PCB.

Finally, it was possible to develop a system which can measure the magnetic field variation, which is dependent on the position of magnet and transmit it wirelessly via a wireless transmitter.

Further work can be done to modify this project by processing the sensor data acquired and associate it with the exact location of the magnet. Another possibility of further work is modifying the PCB further by using a microcontroller with an embedded wireless module such as CC430 by Texas Instruments.

Keywords	knee angle measurement, knee implants ,magneto sensors

Contents

1 Introduction

The objective of this thesis paper is to explain the process of solution development and implementation of a knee angle measurement system. The project took place in EMSYS (Embedded System Research Group) at Thomas More University College in Belgium. The project was carried out in a team of two students Mahlet Zewde (the writer of this theses paper) and Fikeraddis Lemma and a project manager Dr.Eng Patrick Peligrims. The implementation involves various phases and involves a number of students working on the topic in different phases of the implementation.
The project was a continuation of a thesis project of masters of engineering student Kevin Sebrechts, who came up with the idea of knee angle measurement system and implemented the first phase of the project.

The thesis project was proposed to me and my colleague by Dr.Eng Patrick Peligrims when we applied for an international project as Erasmus students at Thomas More University college of Belgium. The original plan was to modify the previous work by Kevin Sebrechts and make a few adjustments to result in a better communication and data acquisition. But, as we were very interested in the project, we decided to extend our stay and launch the second phase of the project using a new set of sensors and eliminating some unnecessary components resulting in more cost efficient and smaller circuitry.

2 Theoretical Background

2.1 Anatomy of Human Knee

Understanding the human knee movement was very crucial throughout this project. The human knee is a complex pivotal hinge joint which has three axis of movement. The joint consists of bones, tendons, muscles, ligaments and nerves. All of these play an important role for a successful movement.

The tibia, femur and patella are the three important bones in the knee joint. The tibia is the bigger bone located in the part below the knee; the fibula is the smaller bone below the knee; the femur is the bone located above the knee and the patella is the knee cap.

Ligaments connect these bones in the upper and lower leg. There are four important ligaments. The two cruciate ligaments, the anterior cruciate and posterior cruciate form a cross inside the knee joint and connect the tibia and the femur. The two other collateral ligaments, the medial collateral and lateral collateral are found outside of the joint.
These parts are shown on Figure 1 below.

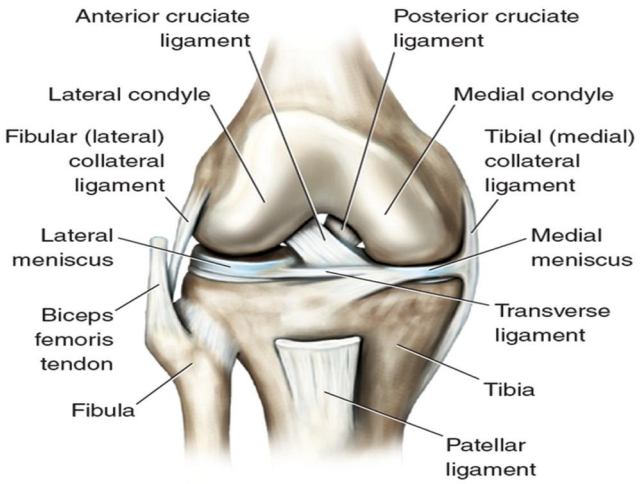


Figure 1. Human knee joint anatomy
Reprinted from: Behnke (2012) [1,248].

The human knee performs two kinds of motions in three axes of movement. These movements are rolling motion initiate and gliding motion. These movements are shown in figure 2.

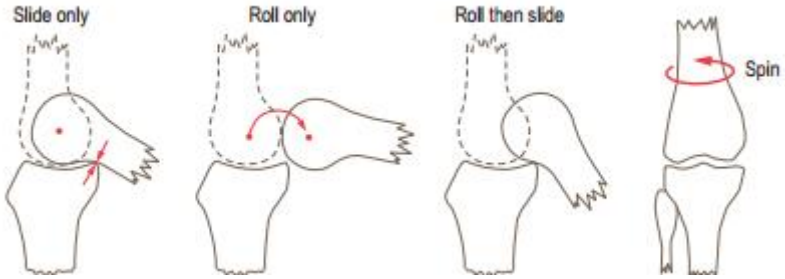


Figure 2. Rolling,Sliding and Spining of the knee
Adapted from: Palastanga and Soames (2012) [2, 20].

The three axes of movements are Anterior-Posterior flexion movement across the sagittal plane Medial-Lateral movement on transverse plane on and Proximal-Distal extension as can be seen on figure 3.

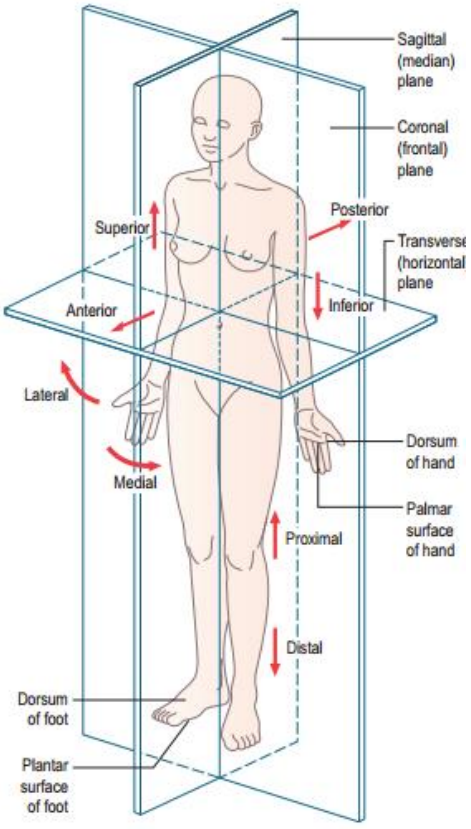


Figure 3. Axis of rotation of human body
Reprinted from: Palastanga and Soames (2012) [2, 1].

The human knee joint can be damaged due to various factors. Such as hereditary arthritis, osteoarthritis, rheumatoid arthritis and developmental abnormalities, for instance – while others are acute, such as avascular necrosis (bone death), obesity or emergency traumas. These conditions can lead to knee replacement surgery when other treatment options for example physical therapy are not effective.

2.2 Knee Replacement Implants

When a human knee joint is damaged, having a knee implant surgery can be an option. Mostly knee replacement surgeries take place in elderly people over 65 years of age. In some cases young people also get knee replacement surgeries when the joint is damaged during extreme sports or mobility is limited by some other causes.
During a knee implant surgery three bone surfaces might be replaced. These are the lower end of the femur bone (the femoral component), the upper end of the tibia (the tibial component) and the back surface of the patella (the patellar component). Patients with a major damage of the whole knee joint can get a total knee replacement surgery in which all of the three bone surfaces are replaced with an implant. Figure 4 shows the implants from Zimmer Biomet.

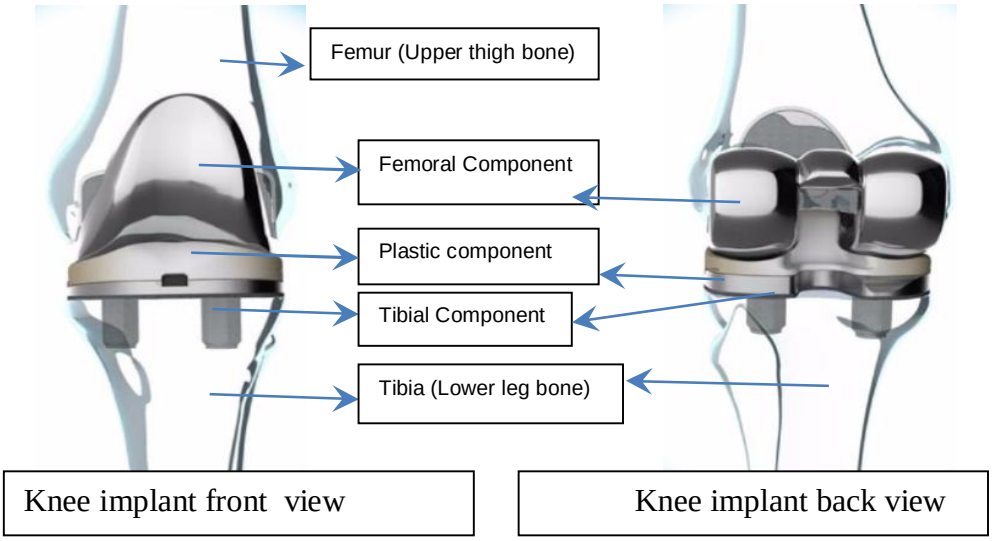


Figure 4. Total Knee implants
Adapted from: Knee Replacement Products (2015) [3]

In most cases the femoral and tibial component are made of titanium or cobalt-chromium based alloys, while the plastic part is made of polyethylene. The plastic component acts like a cushion between the femoral component and tibial component by reducing friction.

2.2.1 Complications of Knee Implant Surgery

Knee replacement surgeries help to improve the quality of life of patients. But, as any medical procedure, they bear few complications. These complications can be caused either by faulty implants or improper use of implant. These complications include loosening of the implant, dislocation, infection, bone loss, bone fracture etc..,

In some cases metallosis can happen. Metallosis occurs when the plastic component between the two metal components(femoral and tibial) wears off from over use and small metal pieces are released to the blood stream from the friction of the tibial and femoral component. [4.]

In cases when complications such us implant dislocation, loosening of implant or metalosis occur a revision surgery is required to fix the complication.

2.2.2 Revision Surgery

Knee replacements generally last more than 20 years, but not all patients experience the same results. For patients whose artificial joint wears out earlier than expected or who received a defective implant revision surgery is required. Any serious complication can also lead to revision surgery. In addition, younger recipients require revision surgery when the first implant wears out.

Revision surgery rates for knee replacements are quickly rising in the United States. In 2005, about 38,000 revision knee surgeries were performed. By 2030, estimates are there will be nearly 268,000 performed each year showing a 600 percent increase. [4.]

Implants that are used in revision surgeries typically have thicker, longer stems for added stability and to replace bone loss. In addition, surgeons typically implant a constrained (hinged) knee during revision surgery because it provides strength for severely damaged knees and weak ligaments. [4.]

2.3 Purpose of Knee Angle Measurement System

As discussed above the wear and tear of knee implants and surgical complications lead to revision surgeries in a lot of cases every year. This number can be reduced if there is a means to provide continuous diagnosis of the implants. This is where knee angle measurement system comes to the picture. Measuring the knee angle via a system built in the implants helps to provide continuous measurement data about the implant performance and leads to early detection of complications such as implant dislocation, loosening of implants and metallosis.

3 Realization of Knee Angle Measurement System

This solution consists of a knee unit: an in vivo system which would be part of knee implants; A Data Logger: an outside system used to receive data wirelessly from the Knee Unit and store the received data to USB or PC; and an outside power supply unit which would supply power wirelessly to the in vivo Knee Unit. Figure 5 shows the block diagram of this system.

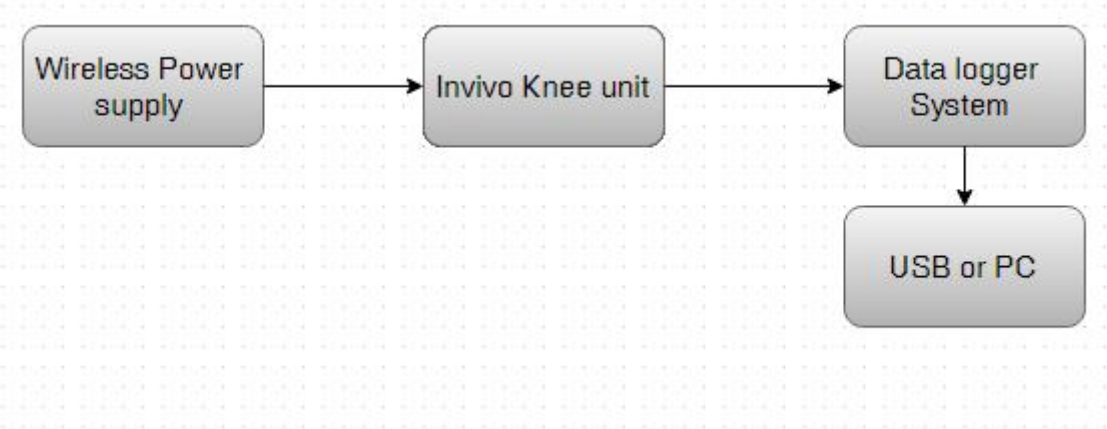


Figure 5. Block Diagram of Knee angle measurement system

The target system for this project was the in vivo Knee unit which consists of different peripheral Units such as: a magnet, 3 Magneto sensors, a Microcontroller (MSP430FR5724) and a wireless communication module (nRF24L01).

The magnet is located in the femoral component of the knee implant and the sensor system is placed in the plastic component right above the tibia as can be seen in figure 6. Thus, it was possible to detect the relative angle between the femur and the tibia. In healthy functioning implants the range of motion and the angle between the femur bone and the tibia is comparable to that of the natural knee. But if the knee implant has failed for some reason the range of motion and the angle is also greatly affected. Less range of motion and stiffness suggests complications in the knee implant and helps for early detection of complications.

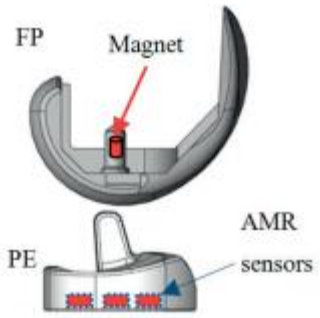


Figure 6. The magnet placed in the femoral part (FP) and the sensors located in the .polyethylene insert (PE) Adapted from ARAMI (2014) [5, 95]

The solution was implemented in two phases. First, the work of previous students was modified and optimized. Afterwards, the analog magneto sensors were replaced with digital sensors, and the analog frontend (AFE) was omitted, resulting in a smaller and cheaper PCB.

Power efficiency is an essential part of the entire scheme since the plan is to use an out vivo wireless inductive power supply unit in the future. In the new version of the code various programming techniques were used to save power resulting in a more energy efficient system. These measures include: changing the software SPI communication used in the previous software to hardware peripheral SPI, using interrupts to enter to low power mode and sleep the processor, and taking advantage of the ULP(Ultralow power) advisor of code composer studio for modifying the code accordingly.

3.1 Phase One of Knee Angle Measurement System Operation Principle

Phase one of the PCB design consists of different system integrations including a magnet, Linear/Angular/ Rotational displacement sensors(HMC1512), Low power 8channel 24-bit analogue front-end ADS1296, MSP430F57xx family mixed signal microcontroller (MSP4305724) and nRF24L01 single chip Transceiver. The operation principle of the whole system is explained step by step as follows.

As it can been seen on figure 7 the initial solution uses HMC412 analog magneto sensors from Honeywell electronics for sensing magnetic field intensity from the magnet located in the femoral component of the implant. This analog data is then transmitted to the ADS1296 analog front end (AFE), the main purpose of ADS1296 is to convert the analog data from the sensors in to 24 bit digital values, as the Msp430FR5724 microcontroller does not provide us with the required number of ADC channels for the three sensors. After the analog data from the three sensors is converted to a 24 bit digital values, the values are manipulated by the Msp430 microcontroller in to a 16 bit format which is suitable to represent the magnetic data with sufficient accuracy and in a format suitable to be transmitted wirelessly via the nRF24L01 wireless module.

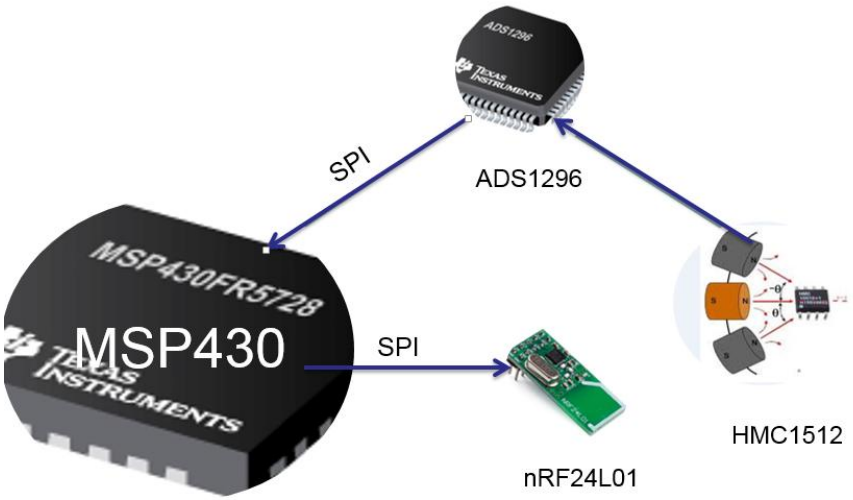


Figure 7. Flow diagram for the initial version of the Circuitry

Figure 7 also shows serial peripheral interface (SPI) protocol is used for communication.
In the following sections the components used in the phase one of the board design are discussed in detail

3.1.1 Microcontroller Unit (MSP430FR5728)

The Texas Instruments MSP430FR57xx family mixed signal micro-controller is the core of this project. It is an ultralow power microcontroller that consists of embedded FRAM non-volatile memory, ultra-low power 16 bit MSP430 CPU, and various peripherals for different applications. It includes 10-bit analog to digital converter, 16-channel comparator with internal voltage references generator and hysteresis capabilities, three enhanced serial channels capable of I^2C, SPI, or UART communication protocols, internal DMA, hardware multiplier, real time clock, three 16-bit timers and many more.

CPU

Operations (excluding program flow instructions) in the MSP430FR57xx family CPU are performed as a register in conjunction with seven addressing modes for source operand and destination operand which has four addressing modes. MSP430 CPU consists of 16 registers which are helpful in reducing execution time. The register-to-register operation execution time is one cycle of the CPU clock. The registers from R0 to R3 are used as program counter, stack pointer, status register and constant generator. The rest are general purpose registers. Connection of the peripherals to the CPU is done using data, address and control buses through instructions. [6, 88-89.]

Figure 8 shows the functional block diagram of the CPU for MSP430FR57xx family microcontrollers.

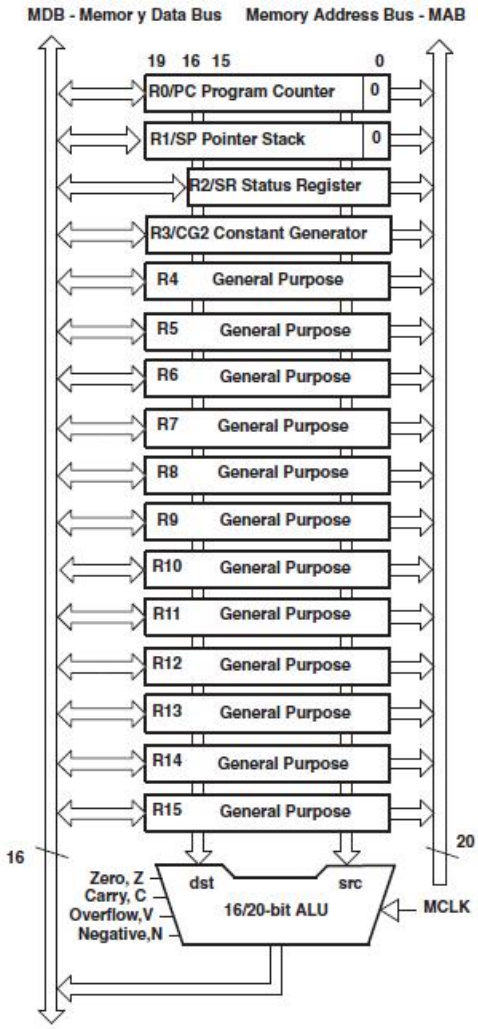


Figure 8. MSP430X CPU Block Diagram

Reprinted from: MSP430FR57xx Family User´s Guide. (2013) [6, 89]

Clock

The MSP430FR57xx family clock system consists up to five clock sources and four clock signals. These clock sources available on MSP430FR5724 are XT1CLK, VLOCLK and DCOCLK. These clocks provide clock for the four clock signals.
These clock signals are

- Auxiliary clock (ACLK): is used by individual peripheral modules.
- Master clock (MCLK): is used by CPU and system.
- Subsystem master clock (SMCLK): is used by individual peripheral modules.
- Module clock (MODCLK): Is used by various peripheral modules and the module oscillator (MODOSC)

All these clock software selectable as XT1CLK, VLOCLK, DCOCLK, and can be divided by 1, 2, 4, 8, 16, or 32. [6, 71-74.]

Initializing the master clock (MCLK) and other clocks is the first thing that takes place in any embedded system programming. Without the clock system there is no microcontroller. Thus it can be seen from Appendix 1 in the beginning of the code the clock signal is initialized, Listing 1 below shows the details of the clock initializing function.

```
void vInitClock(void)
{
      WDTCTL = WDTPW + WDTHOLD;
            //Stop Watch Dog Timer

      CSCTL0_H = 0xA5;
                  //CSKEY password. Always reads as 096h.

                              //Must be written as 0A5h when
writing in word mode; writing any other value in word mode
generates a PUC.

                              //After a correct password is
written and CS register accesses are enabled, a wrong
password write in byte mode disables the access, and no
PUC is generated

      CSCTL1 = DCOFSEL0 + DCOFSEL1;
            //DCO range 1MHz-8MHz, DCO frequency = 8
      CSCTL2 = SELA_3 + SELS_3 + SELM_3;
            //Set ACLK = MCLK = DCO
      CSCTL3 = DIVA_0 + DIVS_0 + DIVM_0;
            //set all dividers => f/1
      /*CSCTL4 not used => only for external oscillators*/
}
```

Listings 1. Code section showing the clock initializing function

From the listing 1 it can be observed that the auxiliary clock (ACLK) and master clock (MCLK) are selected as DCOCLK with no division to 8MHZ.

Operating modes

Msp430 has one active mode and seven software selectable low-power mode operations.
The device can wake up from the low power mode by an interrupt. It restores back to the normal operation after the execution of the interrupt. Each low power mode has different clock sources that are switched off to save power. In low power mode 0 the master clock is disabled; in low power mode 1 the master clock (MCLK) is disabled and the sub system master clock (SMCLK) is optionally active; in low power mode 2 the master clock (MCLK) and the sub system master are disabled; in low power mode 3 the master clock (MCLK) i and the sub system master are disabled; In low power mode 4 all clock sources are disabled. [6, 37.]

Interrupts are used to wake up the system from low power mode and enter active power [6, 38]. In this project port interrupt was used for the DRDY output pin of LIS3MDL to wake up the processor when data is available from magneto sensors. This can be seen in the source code provided in appendix 1.

GPIO

The general input output pins (GPIOs) for various peripherals are shown in the figure 9.

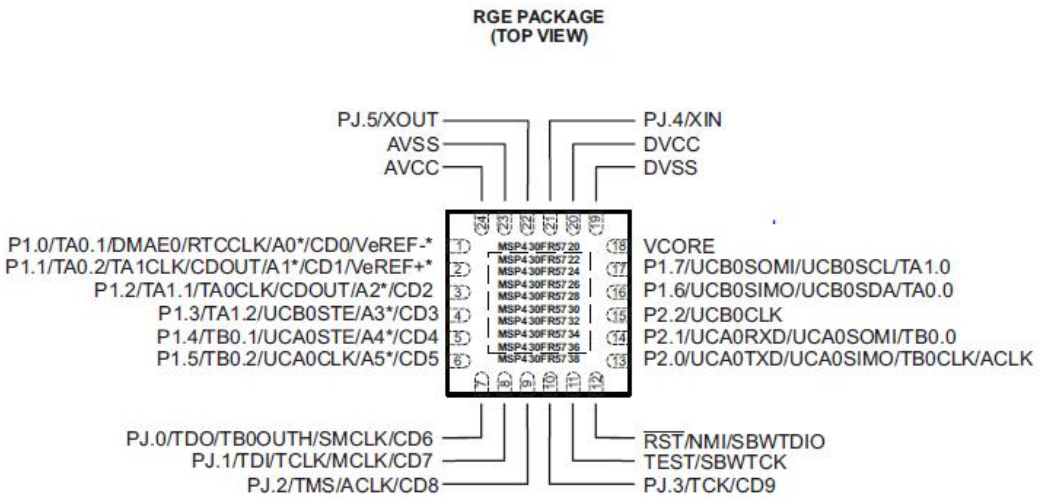


Figure 9. MSP430FR57xx Top view

Reprinted from: MSP430FR572x Mixed-Signal Microcontrollers. (2014) [7, 7]

As it can be observed from figure 9 above most pins have two or more multiplexed functions. If for example we take pin 1.7 we can see that it serves as a general input output pin (GPIO), a master in slave out pin (MOSI) for the serial peripheral interface in eUSCI_B0 SPI mode (UCB0SOMI), as a serial clock for I^2C serial interface in eUSCI_B0 I2C mode (UCB0SCL) and as a capture signal CCR0 capture for timer A1 (TA1).

The functionality and direction of each pin is controlled by a number of control registers. These registers are:

1. PxSEL1 and PxSEL0: These two registers control which peripheral function to be used. Table 1 below shows the different combinations of the two register values for selecting specific functions. [6,295.]

Table 1. Function Selection using PXSEL1 and PXSEL0
Reprinted MSP430FR57xx Family User´s Guide. (2013) [6,295]

PxSEL1	PxSEL0	I/O Function
0	0	General purpose I/O is selected
0	1	Primary module function is selected
1	0	Secondary module function is selected
1	1	Tertiary module function is selected

2. PxDIR: Controls the direction of an individual pin as an input or output pin. PxDIR=0 for input pin and PxDIR=1 for output pin. [6,294.]

3. PxREN: enables the pull up or pull down resister. Whether or not a pull up or pull down resister is controlled by PxOUT register. The PxREN and PxDIR register settings are shown in table 2.

Table 2. IO configuration using PxDIR, PxREN and PxOut
Reprinted from. MSP430FR57xx Family User´s Guide. (2013) [6,295]

PxDIR	PxREN	PxOUT	I/O Configuration
0	0	x	Input
0	1	0	Input with pulldown resistor
0	1	1	Input with pullup resistor
1	x	x	Output

Table 2 shows the different combinations of registers for selecting pull up or pull down resistors.

Enhanced Universal Serial Communication Interface (EUSCI)

There are two enhanced universal serial communication interfaces (EUSCI) in MSP430FR5724. These are EUSCI A and EUSCI B. EUSCI A contains a UART module and an SPI module, while EUSCI B contains an I2C module and another SPI module.
For this project we used both EUSCI A and EUSCI B SPI modules to communicate with the wireless module and the magneto sensors.

3.1.2 Sensor Unit Linear/ Angular/ Rotary Displacement Sensor (HMC1512)

Honeywell's HMC1512 is a high resolution, low power sensor with the ability of measuring the angle direction of a magnetic field from a magnet with a wide angular range of ±90° with $<0.05^0$ resolution and can give full scale output ranges of 120mV with 5V of power supply. Output is typical Wheatstone bridge. One of the advantages of using HMC sensor is it is small size with dimensions of 5 mm x 4 mm x 1.2mm total mounting envelop with pins less than 6 mm square.

Operation Principle

Honeywell´s HMC1512 uses the Anisotropic Magneto Resistive technology. Magnetoresistant materials are materials whose electrical resistance changes with change in the surrounding magnetic field. The Anisotropic Magneto Resistive effect is the effect resulting in change in the resistance of a current carrying magnetic material based on the relative angle between magnetic field with current [5,34]. This effect is demonstrated in figure 10 .

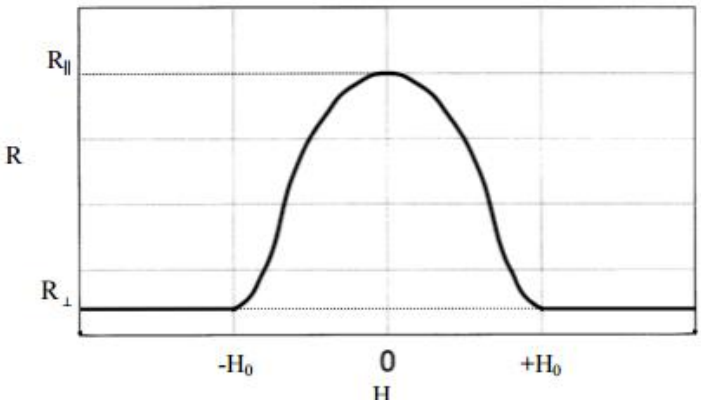


Figure 10. The Resistance of a ferromagnetic thin film as a function of magnetic field
Reprinted from: Arami (2014) [5, 34]

Where $R_{\parallel}$ is value of resistance when magnetic field is parallel to current, $R_{\perp}$ is value of resistance when magnetic field is perpendicular to current and H_0 is the value of the saturation magnetic field level of the AMR sensor. As it can be seen on figure 10 in the previous page the resistance value varies with the variation of the external magnetic field. The resistance is maximum when the applied external magnetic field is parallel to the current in the sensor and the resistance is minimum when the external magnetic field is perpendicular to the current.

Like most hall sensors HMC1512 also takes advantage of the bridge circuitry. As it can be seen in figure 11 in the next page the value of ΔV varies in relation to change in the resistance value of the magneto resistors. In a balanced state where there is no external field applied the value of the differential voltage Δv=0 or very close to zero.

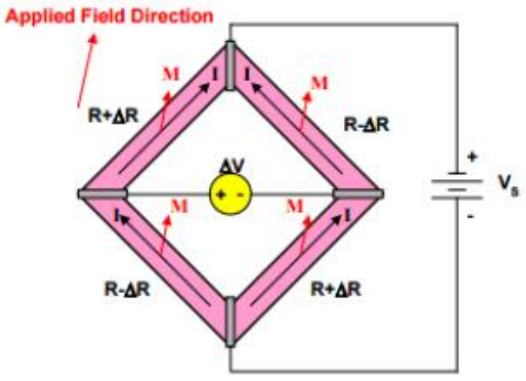


Figure 11. AMR Bridge
Reprinted from Applications of Magnetic Position Sensor. [8, 1]

The magnetic position based sensor HMC1512 contains AMR sensors that has to be used in saturation mode. The sensor contains two AMR bridges A and B sense $\pm 90^0$ position range. The two bridges have orientation of 45^0 from each other .AMR Bridge Copied from Applications of Magnetic Position Sensor. [8, 1]

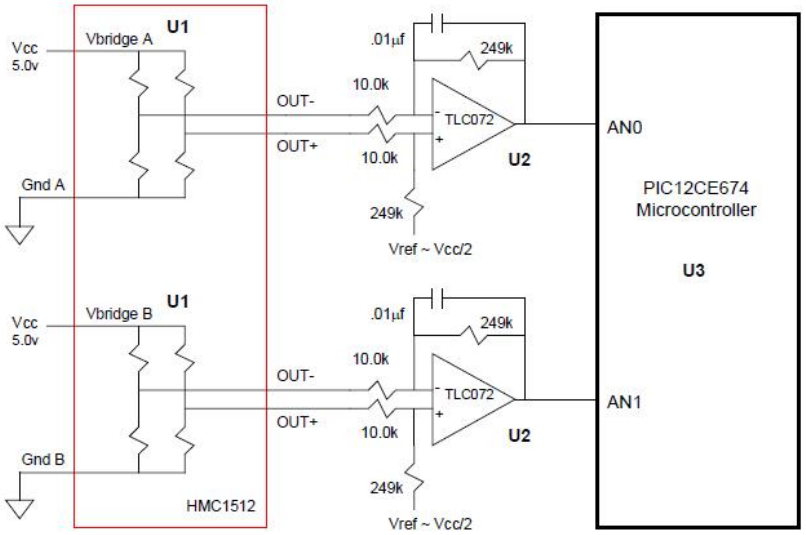


Figure 12. HMC1512 Circuit

Reprinted from Applications of Magnetic Position Sensor. [8, 4]

The differential output voltage for the sensor bridge A can be calculated as

$$\Delta VA = Vs\ S \sin(2\Theta) \quad (1)$$

where Δ VA is differential voltage in bridge A.

VS is the supply voltage

S is Material Constant

Θ is the magnetic field angle

And for sensor bridge B, the differential output voltage is given in equation 2.

$$\Delta VB = -Vs\ S \cos(2\Theta) \quad (2)$$

Where Δ VB is differential voltage in bridge B.

VS is the supply voltage

S is Material Constant

Θ is the magnetic field angle

The three sensors are placed in a triangular shape as shown in figure 13 which is necessary if we need to measure the angle accurately.

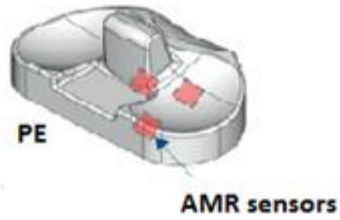


Figure 13. ARM sensors arranged in a triangular shape in the polyethylene part of the implant. Adapted from ARAMI (2014) [5, 95]

3.1.3 Analog Front End (ADS1296)

The ADS1296 Is a 24 bit six channel (ΔΣ) analog-to-digital converter (ADC) with built in programmable gain amplifiers (PGAs), internal reference and an on-board oscillator. ADS12xx family is mostly used in medical applications; medical electrocardiogram (ECG) and electroencephalogram (EEG).

The noise performance can be adjusted by the data rate and PGA (programmable gain amplifier) setting. There are seven PGA gain settings (1, 2, 3, 4, 6, 8 & 12). Reducing the data rate decreases the noise while increasing the PGA values drops the noise which is important in measuring low level bio-potential signals. The internal reference can be programmed to either 2.4 V or 4 V with a clock of 2.048 MHz internal oscillator.

Operation Principle of ADS1296 Analog Front End

Communication between the ADS1296 and other devices such as a microcontroller is achieved by Serial Peripheral Interface (SPI) protocol. The ADS contains 4 pins assigned for this purpose. These are Chip Select (CS), Serial Clock (SCLK), Data In (DIN) and Data Out (DOUT). These pins each have unique purposes.

The CS pin as the name implies selects the ADS for SPI communication. When this pin goes low SPI communication is enabled and ADS129x is selected so that the device

executes incoming commands every eight serial clock pulse. When the CS pin goes high any undergoing SPI communication is interrupted. . In CS high mode, the serial interface is reset, DIN and SCLK are ignored and DOUT enters a high impedance state. The data ready pin (DRDY) will be asserted after data has been converted independent of CS pin mode. Figure 14 below demonstrates the serial data output of ADS1298.

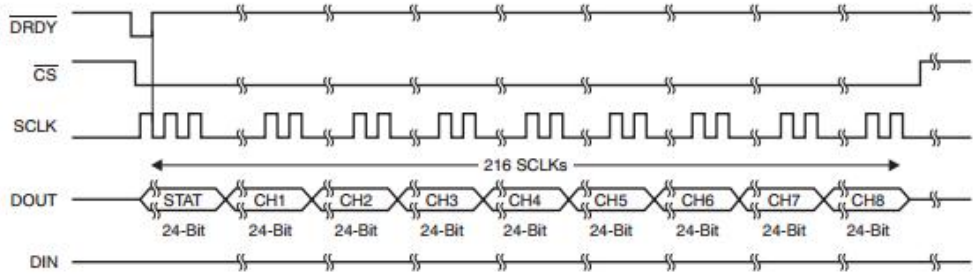


Figure 14. ADS1298 (Eight Channels) SPI Bus data output

Reprinted from: ADS129xADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for
Biopotential Measurements. (2015) [9, 37]

The SCLK pin is the serial peripheral interface (SPI) serial clock. It is used to move in commands and move out data from the device. The serial clock (SCLK) features a Schmitt-triggered input and clocks data on the DIN and DOUT pins into and out of the ADS129x. It is recommended that multiples of 8 SCLKs be presented every serial transfer to keep the interface in a normal operating mode. If the interface ceases to function because of extra serial clock signals, it can be reset by toggling CS high and back to low.

The minimum speed needed for the SCLK depends on the number of channels, number of resolution and output data rate. The calculation below assumes that there are no other commands issued between data captures. Figure 15 shows the timing requirement for the ADS1296.

tSCLK<(tDR– 4tCLK)/(NBITS × NCHANNELS + 24) (3)

Where tSCLK is serial clock

tCLK is the internal clock of ADS1296

tDR is the time period of the output data

NBITS is number of bits from each channel

NCHANNELS is the number of used ADS1296 channels

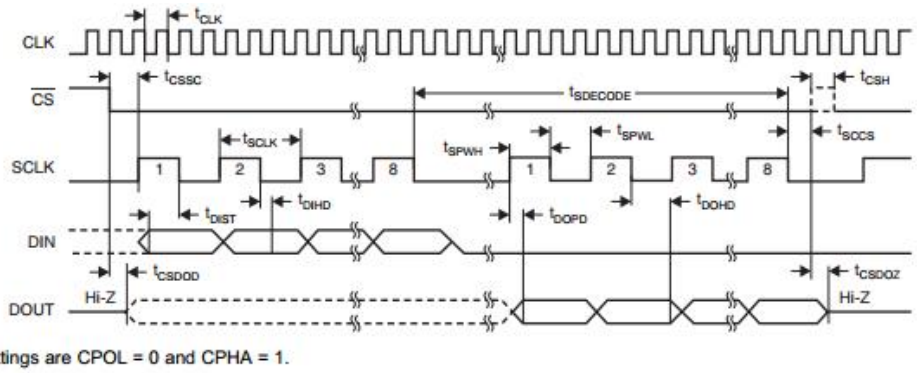


Figure 15. Serial Interface Timing Characteristics

Reprinted from: ADS129xADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements. (2015) [9, 15]

There are two serial clock (SCLK) clocking methods for multiple byte commands as shown below. For SCLK speeds that meet the command decode timing (tSDECODE) requirement ; SCLK is transmitted continuously when CS is low and on free running, SCLK operates when CS is high. [9, 60.] For faster SCLK speeds that do not meet the tSDECODE timing requirement, SCLK is transmitted in 8-bit bursts with a delay between bursts. This can be seen in the figure 16.

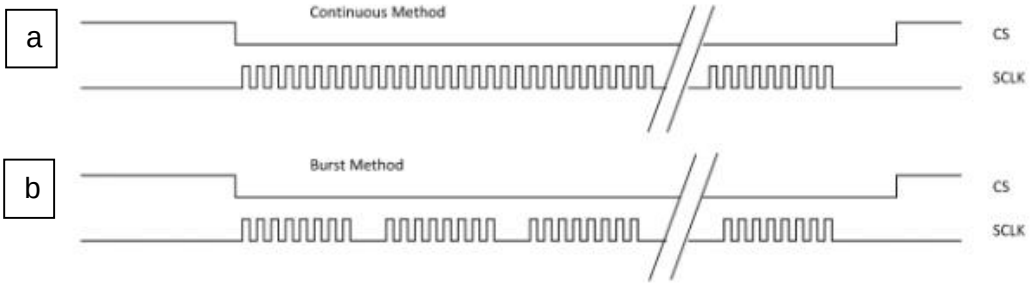


Figure 16. SCLK clocking method for ADS1296 a) Continuous clocking method. B) Burst method.

Reprinted from: ADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements. (2015) [9, 37]

The opcode commands, summarized in table 3, control and configure the operation of the ADS129x. The opcode commands are stand-alone, except for the register read and register write operations that require a second command byte plus data. CS can be taken high or held low between opcode commands but must stay low for the entire command operation (especially for multi-byte commands). [8, 38].

Table 3. Some Opcode Command definitions for ADS129x

Reprinted from: ADS129xADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements. (2015) [9, 38]

COMMAND	DESCRIPTION	FIRST BYTE	SECOND BYTE
System Commands			
WAKEUP	Wake-up from standby mode	0000 0010 (02h)	
STANDBY	Enter standby mode	0000 0100 (04h)	
RESET	Reset the device	0000 0110 (06h)	
START	Start/restart (synchronize) conversions	0000 1000 (08h)	
STOP	Stop conversion	0000 1010 (0Ah)	
Data Read Commands			
RDATAC	Enable Read Data Continuous mode. This mode is the default mode at power-up.(1)	0001 0000 (10h)	
SDATAC	Stop Read Data Continuously mode	0001 0001 (11h)	
RDATA	Read data by command; supports multiple read back.	0001 0010 (12h)	
Register Read Commands			
RREG	Read *n nnnn* registers starting at address *r rrrr*	001*r rrrr* (2xh)(2)	000*n nnnn* (2)
WREG	Write *n nnnn* registers starting at address *r rrrr*	010*r rrrr* (4xh)(2)	000*n nnnn* (2)

3.1.4 Single Chip 2.4 GHz Transceiver (nRF24L01)

The nRF24L01 is a single chip 2.4 GHz transceiver with an Enhanced ShockBrust™ baseband protocol engine which is convenient for ultralow power wireless applications. It has 2.400-2.4835 GHz frequency range that correlate with the world wide ISM frequency band.

Enhanced ShockBurst™ features

- 1 to 32 bytes dynamic payload length
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe Multi receiver for 1:6 star network

GFSK (Gaussian Frequency Shift Key) modulation is used in the radio front end. In addition it consists of a frequency channel, output power and air data rate with 250 kbps, 1 Mbps and 2 Mbps for nRF24L0.

We configure nRF24L01 with Serial Peripheral Interface (SPI). The registers and other necessary details are listed on the data sheet to make the configuration easy and understandable.

The high air data rate combined with two power saving modes make the nRF24L01+ very suitable for ultralow power designs. Internal voltage regulators ensure a high Power Supply Rejection Ratio (PSRR) and a wide power supply range.

Figure 17 shows the block diagram for nRF2L01 chip describing the functionality of each section.

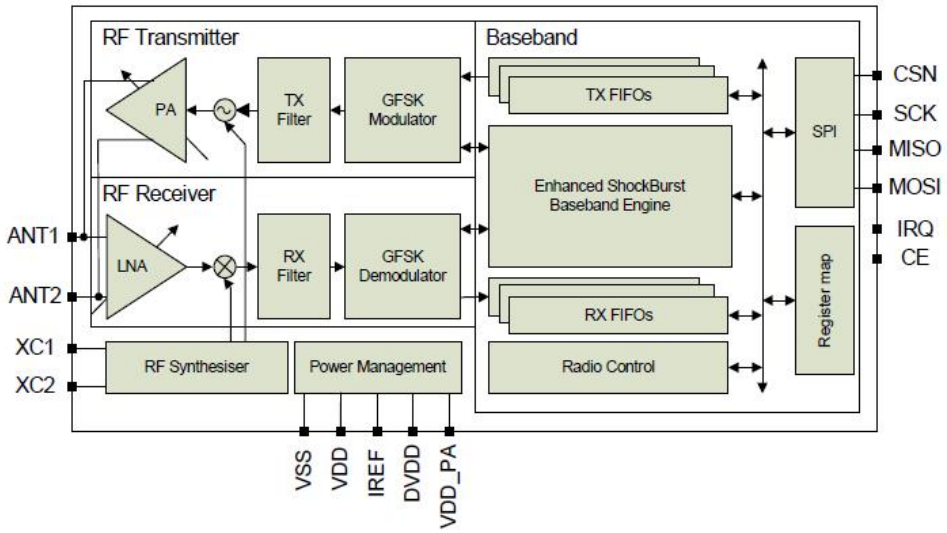


Figure 17. nRF24L01 block diagram

Reprinted from: nRF24L01 Single Chip 2.4 GHz Transceiver Product Specification. (2007) [10, 9]

Modes of operation

This wireless module has four operation modes these are:

- Power down mode: nRF2L01 is disabled with minimal current consumption.
- Standby mode: stand by modes 1 and 2 are used to reduce average current consumption and keep start times short.
- TX_mode: nRF2L01 radio is in sender mode.
- RX_mode: nRF2L01 radio is in sender mode. [10,20-21]

In this project we use the nRF2L01 radio in TX_mode, the selection of different modes takes place by writing to few control registers. Table 4 below shows how the operating modes are configured.

Table 4. Configuring operational modes
Reprinted from: nRF24L01 Single Chip 2.4GHz Transceiver Product Specification. (2007) [10, 21]

Mode	PWR_UP register	PRIM_RX register	CE	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO. Will empty all levels in TX FIFO[a].
TX mode	1	0	minimum 10µs high pulse	Data in TX FIFO.Will empty one level in TX FIFO[b].
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

The tx_mode selection in this project is shown in listings 2.

```
void TX_Mode(void)
{
      CE_LOW;

      SPI_Write_Buf(WRITE_REG + TX_ADDR, TX_ADDRESS,
TX_ADR_WIDTH);    // Writes TX_Address to nRF24L01
      SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS,
TX_ADR_WIDTH); // RX_Addr0 same as TX_Adr for Auto.Ack
      //SPI_Write_Buf(WR_TX_PLOAD, tx_buf, TX_PLOAD_WIDTH); //
Writes data to TX payload

      SPI_RW_Reg(WRITE_REG + EN_AA, 0x01);        // Enable
Auto.Ack:Pipe0
      SPI_RW_Reg(WRITE_REG + EN_RXADDR, 0x01);    // Enable Pipe0
      //SPI_RW_Reg(WRITE_REG + SETUP_RETR, 0x1a);      // 500us
+ 86us, 10 retransmits...
      SPI_RW_Reg(WRITE_REG + SETUP_RETR, 0x0a);   // 250us +
86us, 10 retransmits...
      SPI_RW_Reg(WRITE_REG + RF_CH, USED_RF_CH);  // Select RF
channel 40
      SPI_RW_Reg(WRITE_REG + RF_SETUP, 0x0C);     // TX_PWR:0dBm,
Data rate:2Mbps, LNA:HCURR
      SPI_RW_Reg(WRITE_REG + CONFIG, 0x0e);       // Set PWR_UP
bit, enable CRC(2 bytes) & Prim:TX. MAX_RT & TX_DS enabled..
      }
```

Listings 2. Selection of TX_mode

3.2. Phase two Knee Angle Measurement System Operation Principle

In the second phase the project Honeywell's HMC1512 was replaced with LIS3MDL from ST electronics. As the LIS3MDL provide us with digital output the analog front end ADS1296 was also eliminated. The block diagram for the second version of the PCB is shown below in figure 18.

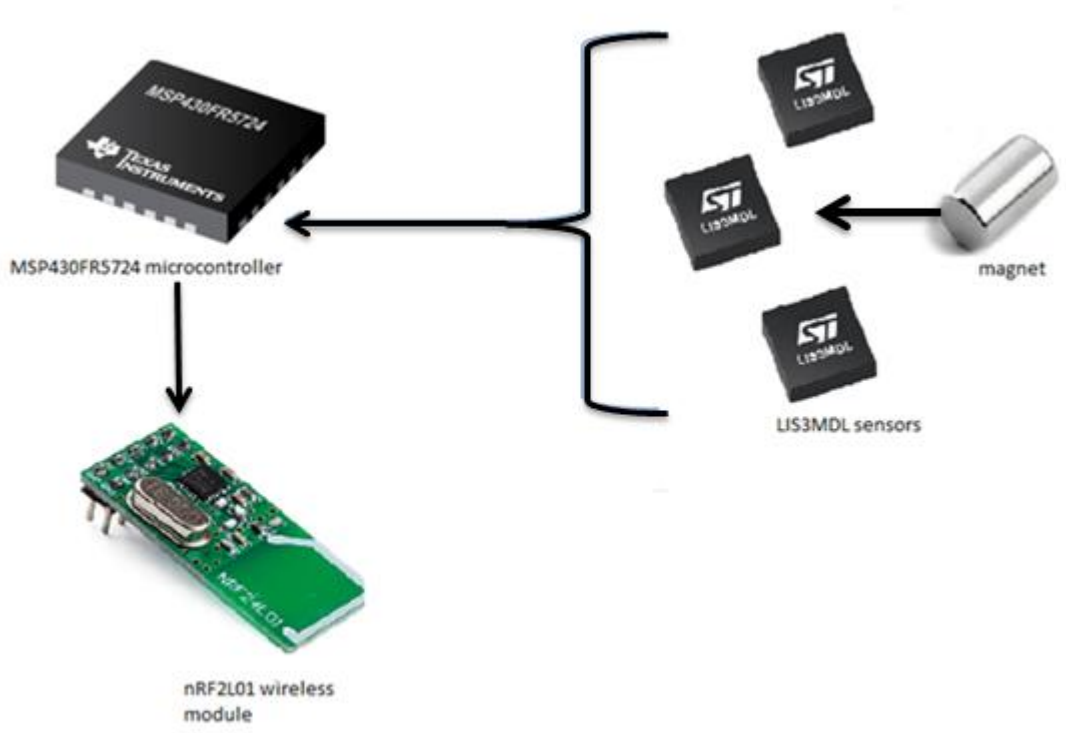


Figure 18. Block diagram of second version of knee PCB

As can be seen from the block diagram the important change in the modified version of knee is the addition of LIS3MDL magneto sensors which are explained in detail in the next versions.

3.2.1 LIS3MDL

The LIS3MDL is a digital magneto sensor from ST electronics. The fact that it has digital output, efficient power usage and the incredibly small size (2.0x2.0x1.0 mm) are few of the advantages of using this sensor.

Wide supply voltage, 1.9 V to 3.6 V, Independent IO supply (1.8 V) gauss selectable magnetic full scale of • ±4/ ±8/ ±12/ ±16 gauss, Continuous and single-conversion modes, 16-bit data output, Interrupt generator, Self-test, I 2C/SPI digital output interface ,Power-down mode/ low-power mode etc.., are some of the features of this sensor. [11, 1.]

The operation of the LIS3MDL sensor is similar to that of the HMC1512 sensors due to the fact that they both use bridge circuitry with anisotropic magneto resistors.As can be seen in figurel below there are 3 bridge circuits one for each axis(x,y,z).The out put of the bridge circuits is multiplexed, amplified and converted from analog to digital values. A control logic system is to configure serial communication and interrupt generation. These thins can be noted in figure 19.

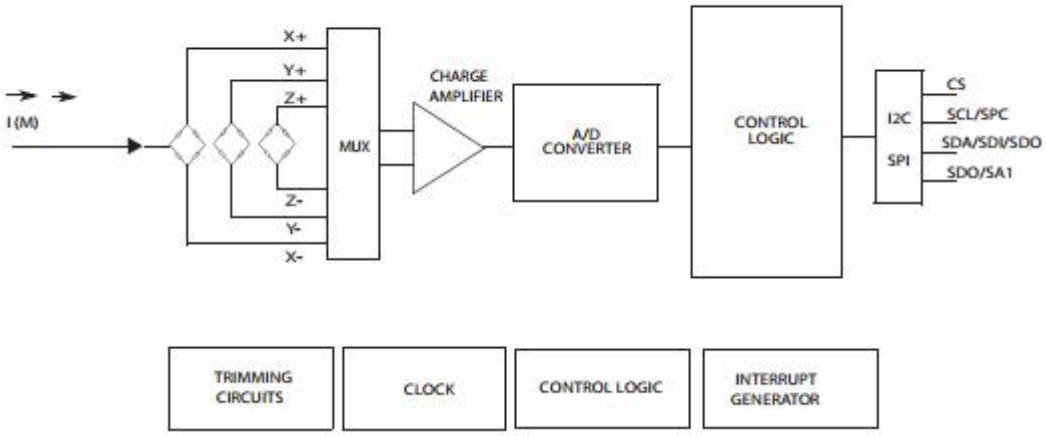


Figure 19. Block diagram of LIS3MDL

Reprinted from .LIS3MDL Digital output magnetic sensor: ultra-low-power, high-performance 3-axis magnetometer (2015) [11, 6]

The LIS3MDL sensors have in addition to the above mentioned features have built in temperature sensors.

Serial Communication

The LIS3MDL has both I2C and SPI digital interfaces. For this project we used the SPI communication thus we will explain the SPI communication interface in detail.

The SPI interface can be programmed to be either in 3-wire or 4-wire. When working in four wire mode the SPI uses chip select (CS), serial port clock (SPC), serial data input (SDI) and serial data output (SDO). Both the read register and write register

commands are completed in 16 clock pulses or in multiples of 8 in case of multiple byte read/write.

Bit 0 of every SPI command is the READ/WRITE bit when 0, data is written in to the device; When 1, data is read from the device. Bit 1 is the MS bit when 0, does not increment the address; when 1, increments the address in multiple writes. Bits 2-7 are address bits AD (5:0) which are the address field of the indexed register. Bits 8-15 are data DI (7:0) in write mode or DO (7:0) in read mode. This is the data that is written inside the device (MSb first). Bits after bit 16 represent further data in multiple byte writes.The SPI read and write protocol is shown in figure 20 below. [11, 18-21.]

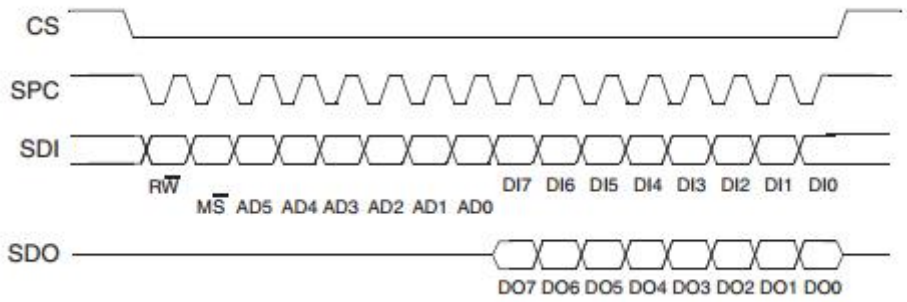


Figure 20. SPI read and write protocol

Reprinted from .LIS3MDL Digital output magnetic sensor: ultra-low-power, high-performance 3-axis magnetometer (2015) [11, 19]

Register Configuration

WHO_AM_I (0Fh: The WHO_AM_ I Register is the device identification register. It is a read only register; when read, it should have a value of 0x3D. Faulty values of these register suggest malfunction in the sensor serial interface.
As can be observed in appendix 1 the WHO_AM_I register is read in the beginning of the code to confirm the proper functionality of the serial communication.

CTRL_REG1 (20h): This register contains bits that enables temperature sensor (T_EN), selects the operative mode (OM),that selects the data output rate (DO),fast data rate selection (FAST_ODR) and self-test (ST) [11,24]. For this project the value of this register was set as 0x62 selecting ultra-high performance mode, fast output data rate enabled as illustrated in listing 3.

```
CS_PORT_OUT &= ~(CS_LIS1|CS_LIS2|CS_LIS3);

__delay_cycles(1);                  //wait

LIS_RW(CTRL_REG1);

//LIS_RW(0X22);                          //MEDIUM
PERFORMANCE MODE ON X AND Y AXIS, FAST ODR ENABLES,
ODR=560Hz
//LIS_RW(0X42);                          //HIGH PERFORMANCE
MODE ON X AND Y AXIS, FAST ODR ENABLES, ODR=300Hz
//LIS_RW(0X3C);                          //MEDIUM
PERFORMANCE MODE ON X AND Y AXIS, FAST ODR DISABLED,
ODR=80Hz
LIS_RW(0X62);                         //ULTRA HIGH
PERFORMANCE MODE ON X AND Y AXIS, FAST ODR ENABLES,
ODR=155Hz

__delay_cycles(5);                  //wait

CS_PORT_OUT |= (CS_LIS1|CS_LIS2|CS_LIS3);
```

Listing 3. Code segment from C code

CTRL_REG2 (21h): This register contains bits that control the full scale selection (FS), that can reboot the system (REBOOT) and software reset (SOFT_RST) [11,25]. The value of this register for this project was selected to be 0x40 selecting full scale value of 16 gauss as illustrated in listings 4.

```
CS_PORT_OUT &= ~(CS_LIS1|CS_LIS2|CS_LIS3);

__delay_cycles(1);                  //wait

LIS_RW(CTRL_REG2);

LIS_RW(0X60);                       // FULL SCALE=12GAUSS

//LIS_RW(0X00);                     // FULL SCALE=4GAUSS
//LIS_RW(0X20);                      // FULL SCALE=8GAUSS
//LIS_RW(0X40);                     // FULL SCALE=16GAUSS

__delay_cycles(5);                  //wait

CS_PORT_OUT |= (CS_LIS1|CS_LIS2|CS_LIS3);
```

Listing 4. Code segment from C code

CTRL_REG3 (22h):This register contains bits that configure the low power mode(LP), select the serial interface mode 4-wire or 3-wire(SIM) and operating mode selection (MD)[11,26]. As illustrated in listings 5 4_wire SPI mode with continuous mode is selected for this project.

```
CS_PORT_OUT &= ~(CS_LIS1|CS_LIS2|CS_LIS3);

__delay_cycles(1);                  //wait

LIS_RW(CTRL_REG3);

LIS_RW(0X00);                       // CONTINOUS CONVERSION
MODE

__delay_cycles(5);

CS_PORT_OUT |= (CS_LIS1|CS_LIS2|CS_LIS3);
```

Listing 5. Code segment from C code

CTRL_REG4 (23h): This register contains bits that control the operating mode selection for z axis (OMZ) and big little endian selection (BLE) [11, 26]. This register is set as 0x0E which means ultrahigh performance mode on the z axis and big little endian is enabled.

```
CS_PORT_OUT &= ~(CS_LIS1|CS_LIS2|CS_LIS3);

__delay_cycles(1);                      //wait

LIS_RW(CTRL_REG4);

//LIS_RW(0X06);                           // MEDIUM
PERFORMANCE MODE ON Z AXIS AND BIG LITTLE ENDIAN  ENABLED
//LIS_RW(0X0A);                           // HIGH
PERFORMANCE MODE ON Z AXIS AND BIG LITTLE ENDIAN  ENABLED
LIS_RW(0X0E);                           // ULTRA HIGH
PERFORMANCE MODE ON Z AXIS AND BIG LITTLE ENDIAN  ENABLED

__delay_cycles(5);                      //wait

CS_PORT_OUT |= (CS_LIS1|CS_LIS2|CS_LIS3);
```

Listing 6. Code segment from C code

Enabling the big little endian means the most significant bit (MSB) data is stored in the lower register address.

CTRL_REG5 (24h): contain bits that control the block data rate (BDU) and fast read option [11, 27].

4 Results and Discussion

4.1 Testing SPI Communication for Using ADS1296 in Phase 1

The performance of the SPI communication was checked throughout the progress of this project using logic analyzer hardware debugging tool and a logic software both from Saleae. This tools are shown in Figure 21 below.

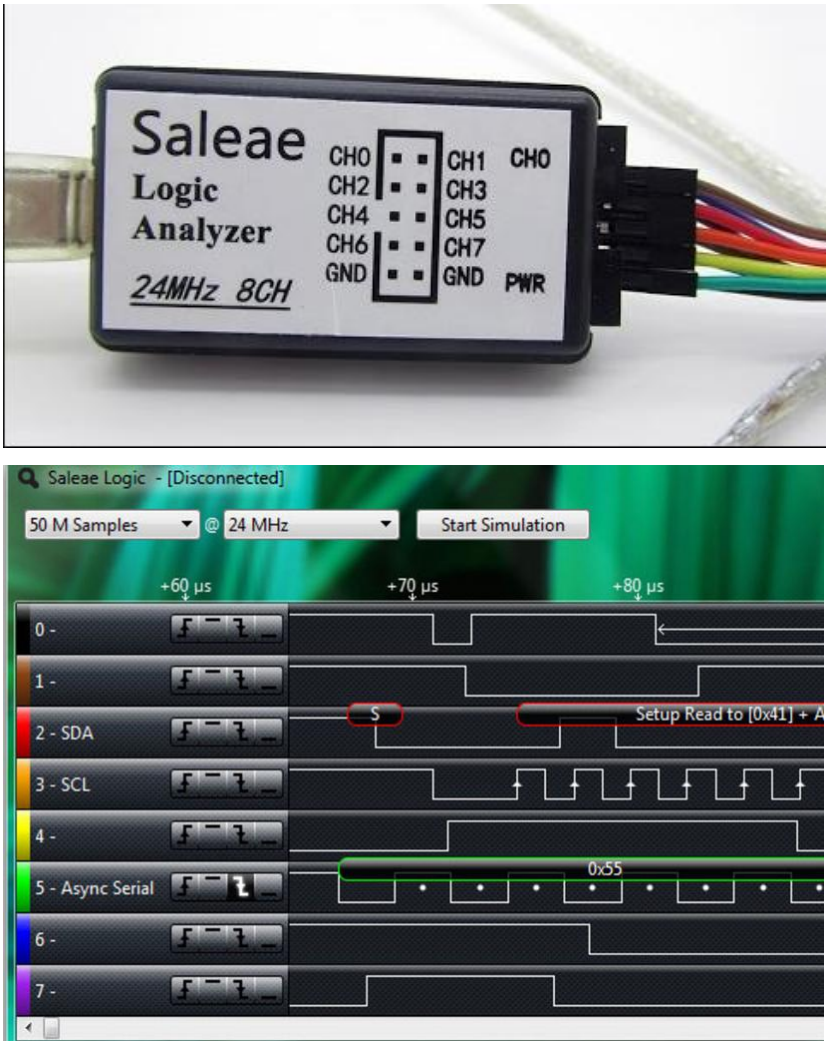


Figure 21. Testing devices A) saleae logic analyzer B) saleae logic application

The SPI communication results for the board in phase 1 with ADS1296 are shown in appendix 3 in detail.

4.2 Testing the performance of Phase 1 of the PCB

The performance of the phase 1 PCB was tested using a LabVIEW application developed specifically for this purpose. This application displays the magnetic data received by the receiver circuit from the knee. This application has three different views showing values in number, value gauge and graphs representation.

The results acquired using this LabVIEW application are presented in appendix 4 in detail where it can be seen only two of the nRF2L01 channels are showing the data. Which clearly dictates a malfunction in the system.

One of the possible causes of this malfunction is possible error in the wireless data transmission which was checked not to be true by sending a constant test data and viewing it in the LabVIEW application. The second cause could be a possible programming error in the manipulation of the magnetic data from 24 bit to 16 bit format. This was also tested by manipulating a constant data and viewing the outcome; which again was working as expected. These leave us with couple of other possibilities: malfunction in the ADS1296 chip and incorrect configuration in the ADS1296 timing. The hardware functionality was confirmed to work fine by running different hardware tests using oscilloscope. Thus the cause is decided to be either incorrect configuration of ADS1296 that do not comply with the timing requirement of the device or mistakes in the data manipulation. Detailed test results of this test set up are shown in appendix 4.

4.3 Testing SPI Communication for LIS3MDL in Phase 2

The performance of the SPI communication between the LIS3MDL sensor and MSPFR5724 checked throughout the progress of this project using logic analyzer hardware debugging tool and saleae logic software. The results obtained from this measurements are shown in appendix 8 measurements where the value of the who_am_I register is read and returned a value 0x3D which is the correct value. This alone proves that we are using the correct SPI settings to communicate with the LIS3MDL sensor.

4.4 Testing performance of LIS3MDL using UART board

A UART board was used in order to gather the output values of magnetic field sensed by the LIS3MDL sensor. The main aim of this test was to evaluate the performance of the LIS3MDL magnetic sensor by alternating full scale value of the LIS3MDL and varying the test set up. Four test setups were used for each full scale value. The circuit diagram used for this test is given below in Figure 22.

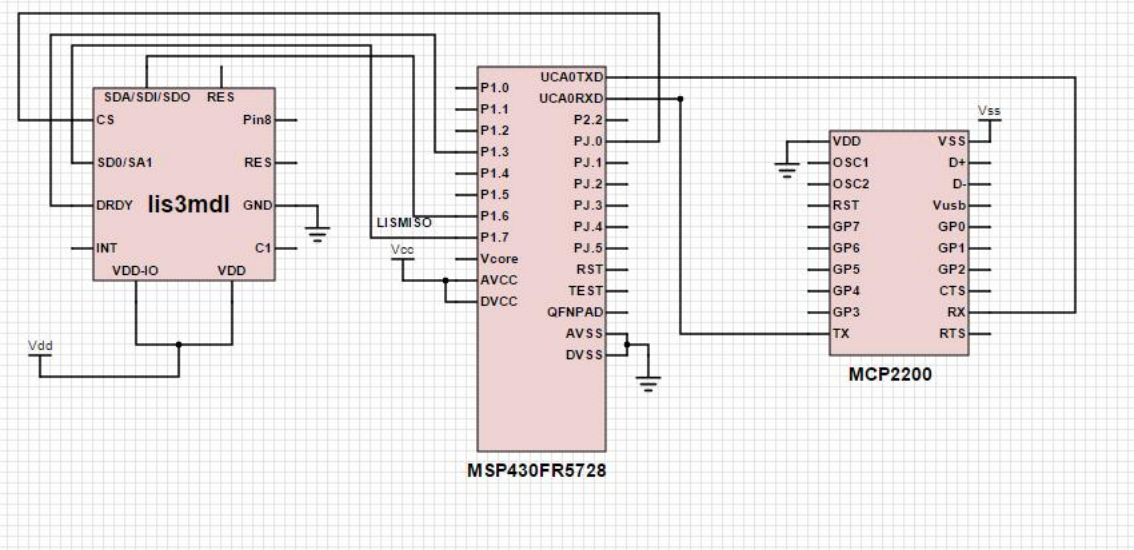


Figure 22. Prototype Schematics

The magnet was placed in various positions with respect to the LIS3MDL sensor and the output of the sensor data was sent to a computer via UART and the hex output values of the sensor were converted into Gauss using Excel sheet.

The various test set ups and the results obtained are discussed in detail the following pages.

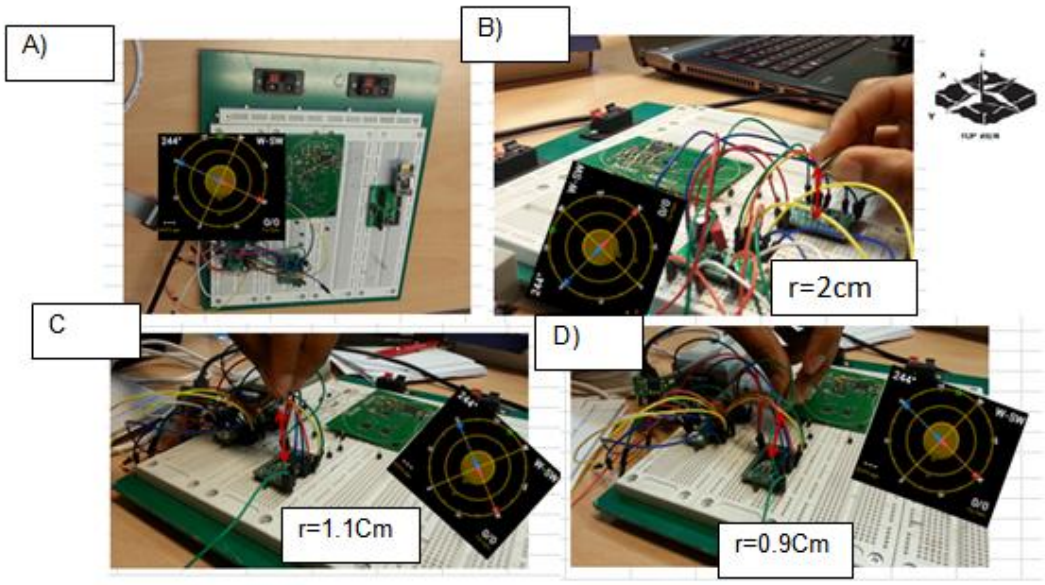


Figure 23. 4 Gauss Full scale A) No magnet applied B, C, D) Magnet applied in various distance and direction from the sensor.

No magnet is applied for the set up in figure 23 A. The resulting magnetic field values z axes are 52.35 milliGauss, 6.55 milliGauss and 1844.21 milliGauss respectively. The earth's magnetic north is the resultant value of the x, y and z magnetic data recorded. In the following mathematical computation the earth´s magnetic field is calculated.

$$Earth´s\ magneti\ field = \sqrt{x(Xaxis)^2 + (Yaxis)^2 + (Zaxis)^2} \quad (4)$$

$$\sqrt{52.35^2 + 6.55^2 + 1844.21^2} = 1844.21 \text{milliGasuss}$$

Since the earth´s magnetic field has a value between 250milliGauss and 650 milliGauss we can undoubtedly conclude the above result is not correct. But there is an explanation for this situation from the data sheet of LIS3MDL the sensor has zero gauss level of ±1Gauss for the test condition ±4 GS. [11, 8]

"….Zero-gauss level offset describes the deviation of an actual output signal from the ideal output if no magnetic field is present…."[11,13.]

Figure 23 B, C and D all show the measurement setup with the magnet held in a varying distance and angle from the LIS3MDL sensor. The appendix 5 tables B, C and D show the corresponding magnetic samples and their averages for these setups. For set up shown in figure 23 B the magnetic field strength is 2938.47 milliGauss in the x direction but has over flown across both y and z axes. Again as can be observed from appendix 5 table C in set up 3(figure 23 C) the magnetic data across z axes has been clipped and in set up 4 (figure 23 D) appendix 5 table D the magnetic data in all the three axes has been clipped.

These results implies that the full scale value of ±4GS is not sufficient for the given magnetic field as result the signal will be clipped and lead to inaccurate measurements. Increasing the Full Scale of the LIS3MDL can result in less clipping. In following pages the results obtained after full scale is increased to 12 Gauss using test setups in figure 24 are presented.

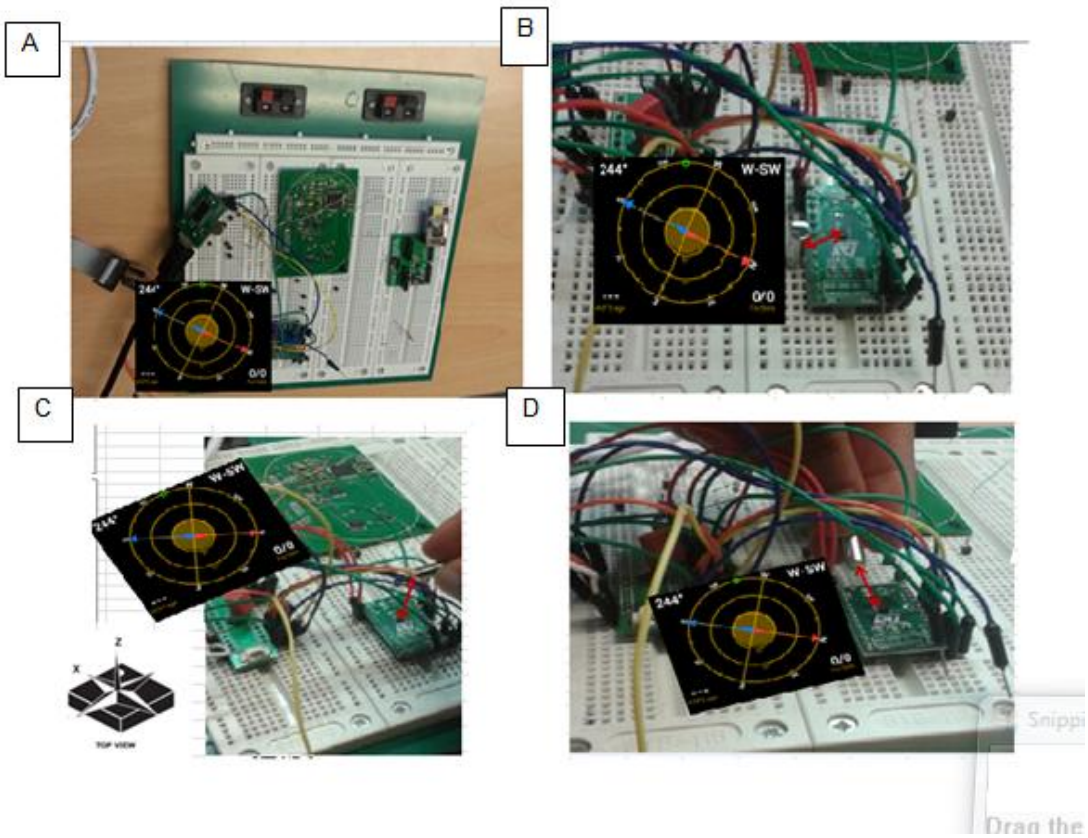


Figure 24. 12 Gauss Full scale A) No magnet applied B, C, D) Magnet applied in various distance and direction from the sensor

In the first set up as shown in figure 24A the magnetic field sensed by LIS3MDL was measured without applying any magnet and the results shown in appendix 6 A were obtained. As it can be seen from the appendix 6A the values obtained were very close to zero in the X Axis the average magnetic field of 10 samples is -17.42 milliGauss, in the y axis the value of the average is around -14.16 milliGauss and in the z axis this value is 1931.57 milliGauss. Taking the equivalent of the x, y and z axes values and using equation 4 we get:

$$\sqrt{17.42^2 + 14.16^2 + 1931.57^2} = 1931.57\text{milliGasuss}.$$

From the result above it can be observed again the value is not in the range of earth´s magnetic field and it can again be explained by the zero gauss level value.

As it can be observed from appendix 6 B for the corresponding setup in figure 24B, in all the three axes the magnetic value have over flown and the values obtained are all greater than 12Gauss which is the full scale value. This is caused by the magnet being placed in a very close distance from the LIS3MDL sensor.

As it can be observed from appendix 6 C which shows the results for setup 24C, the value of the magnetic field in the direction of Z axis have over flown and the values obtained are all greater than 12Gauss; this is caused due to the magnet´s orientation almost parallel to the Z axis. The average value of the magnetic field component across x and y axes for ten samples is and -8.8 Gauss and 4.4 Gauss respectively.

The set up shown figure 24 D resulted in magnetic field value over flow in the X Axis and value of 1.6 Gauss and -7.2 Gauss in the y and z axis respectively.

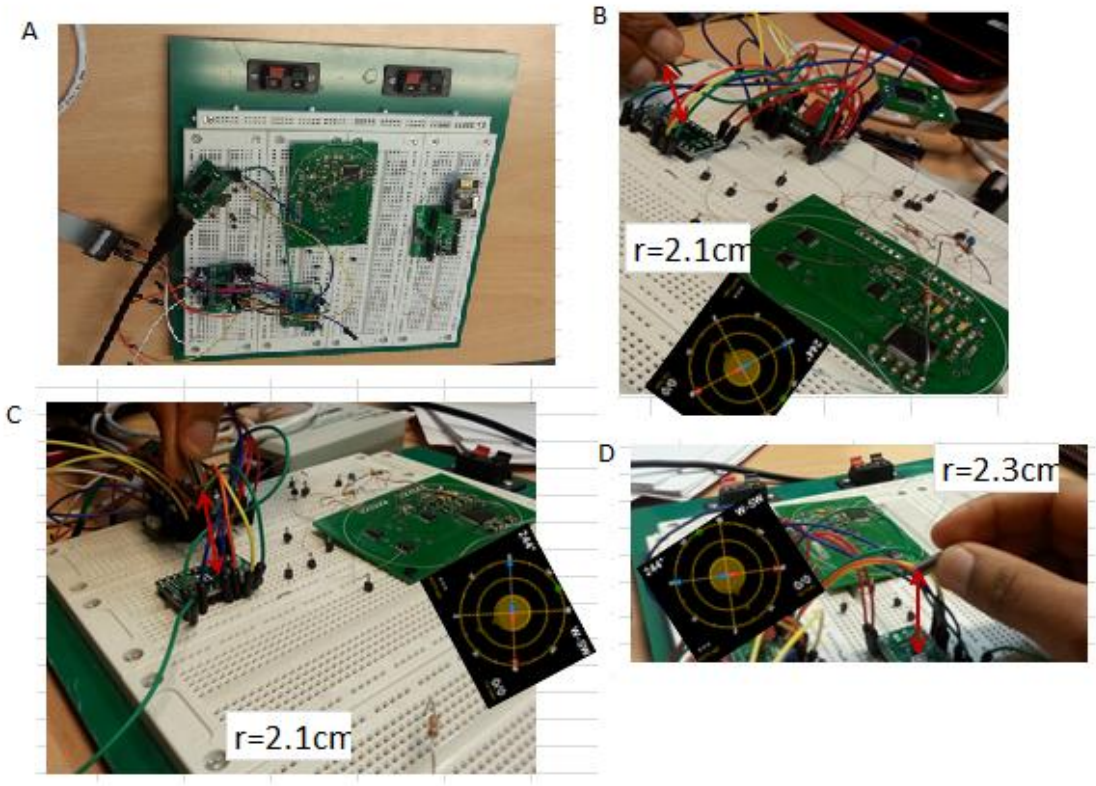


Figure 25. 16 Gauss Full scale A) No magnet applied B, C, D) Magnet applied in various distance and direction from the sensor

In the first set up shown in figure 25A the magnetic field sensed by LIS3MDL was measured without applying any magnet. The corresponding results for this setup shown in appendix 7A. As it can be seen from the appendix the values obtained were very close to zero in the X Axis the average magnetic field of 10 samples is -17.42 milliGauss, in the y axis the value of the average is around -32.09 milliGauss and in the z axis this value is 1559.44 milliGauss. Taking the equivalent of the x, y and z axes values and using equation 4 we get:

$$\sqrt{17.42^2 + 32.09^2 + 1559.44^2} = 1559.44 \text{ milliGasuss}.$$

From the result above it can be observed again the value is not in the range of earth´s magnetic field and it can again be explained by the zero gauss level value of the LIS3MDL sensor.

Appendix 7 B shows the magnetic values corresponding to the set up in figure 25B. From this appendix one can observe that the average of z axes components of the magnetic field have over flown and the values obtained are all greater than 16 Gauss which is the full scale value. This is caused by the magnet being oriented in a very close distance from the z axis of the sensor. In the X and Y axes the magnetic value is -7.2 Gauss and 4.2 Gauss respectively.
From appendix 7 C, the value of the magnetic field in the direction of Z axis have over flown and the values obtained are all greater than 16Gauss, this is caused due to the magnet´s orientation almost parallel to the Z axis as it can be seen from figure 25 C. The average value of the magnetic field across x and y axes for ten samples is and -6 Gauss and -6.6 Gauss respectively.

The set up shown figure 25 D resulted in magnetic field value of 2.8 Gauss, 6.1 Gauss and & 15.6 Gauss in the x, y and z axis respectively.

Notice the change of the magnetic field data throughout the results presented above and the corresponding distance of the magnet from the sensor.

4.5 Testing the performance of Phase 2 of the PCB

Once again the performance of phase 2 of the knee PCB was tested using the LabVIEW application presented in section 4.3. The results of this application are presented in appendix 9. As can be observed from the appendix the value gauges and graphs show a very clear and stable values in all the three axes.

5 Conclusion

The goal of this thesis work was to create a better version of the knee PCB by modifying previous works and making modifications in the embedded C program and have a functional knee system at the end.

The tests performed on the first version of the knee PCB imply a malfunction in the first version of the knee PCB. Implying the embedded software should be modified further and more tests and research should be made on the ADS1296 AFE (analog front end). However the tests performed on the second version of the PCB were satisfying and imply a good functional circuit and embedded software.

Couple of major works can be done in the future. First the statistical and mathematical analysis of the sensor´s magnetic data can be done to identify the exact angle from this data. Secondly a single chip with built in radio module can be used, which would result in a more compact system.

References

1 Robert S. Behnke. Kinetic Anatomy. United States of America: Human Kinetics Publishers; 2012.

2 Nigel Palastanga and Roger Soames. Anatomy and Human Movement Structure and Function. China: Elsevier Ltd.; 2012.

3 Zimmer Biomet. Knee Replacement Products [online]. Warsaw, Indiana USA: Zimmer Biomet; 2014.
URL: http://www.zimmer.com/medical-professionals/products/knee/persona-knee.html. Accessed 17 September 2015.

4 Knee Replacement Surgery [online]. Orlando, Florida, USA: www.drugwatch.com; 2015.
URL: http://www.drugwatch.com/knee-replacement/surgery/. Accessed 1 October 2015.

5 Arash Arami. Kinematics Estimation and Loosening Detection in Smart Knee Prostheses. Lausanne, Switzerland: ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2014.

6 Texas Instruments Incorporated. MSP430FR57xx Family User´s Guide [online]. Dallas, Texas, USA: Texas Instruments Incorporated; 2013.
URL: http://www.ti.com/lit/ug/slau272c/slau272c.pdf. Accessed 27 September 2015.

7 Texas Instruments Incorporated. MSP430FR572x Mixed-Signal Microcontrollers [online]. Dallas, Texas, USA: Texas Instruments Incorporated; 2014.
http://www.ti.com/lit/ds/symlink/msp430fr5722.pdf. Accessed 1 October 2015.

8 Honeywell. Applications of Magnetic Position Sensor [Online]. Morristown, New Jersey, United States: Honeywell International, Inc.;.
URL: http://www51.honeywell.com/aero/common/documents/Applications-of-Magnetic-Position-Sensors.pdf. Accessed (28/9/2015).

9 Texas Instruments Incorporated ADS129x Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements [Online]. Dallas, Texas, USA: Texas Instruments Incorporated; 2015.
URL: http://www.ti.com/lit/ds/symlink/ads1296.pdf. Accessed 27 September 2015.

10 Nordic Semiconductor. nRF24L01 Single Chip 2.4GHz Transceiver Product Specification [Online]. Trondheim, Norway: Nordic Semiconductor ASA; 2007.
URL: http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01. Accessed 27 September 2015.

11 STMicroelectronics. LIS3MDL Digital output magnetic sensor: ultra-low-power, high-performance 3-axis magnetometer [Online]. Geneva, Switzerland: STMicroelectronics N.V., 2015.
URL:
http://www.st.com/web/en/resource/technical/document/datasheet/DM00075867.pdf. Accessed 30 September 2015.

Appendix 1. Source Code main.c

```
/*!
 *THIS CODE HAS BEEN WRITTEN FOR THE USE WITH LIS3MDL MAGNETIC SENSOR FROM ST
ELECTRONICS
 *MSP430fr5724 MICROCONTROLLER Texas INSTRUMENTS AND nRF24L01  WIRELESS
TRANSMITTER FROM NORDIC SEMICONDUCTOR.
 *
 * SPI protocol is used to communicate between sensor,wireless module and
microcontroller.
 *  THIS code is to be used with version 3 (with LIS3MDL sensor) of the knee
PCB BEING DEVELOPED BY EMSYS RESEARCH LAB
 *
 *  THE CODE IS WRITTEN AND TESTED BY Mahlet Zewde and Fikeraddis Lemma-----
ERASMUS STUDENTS FROM METROPOLIA UAS FINLAND.
 *
 *
 *
 *  */



// includes ===================================

#include <msp430.h>
#include "gpio.h"
#include "globalUtils.h"
#include "SPI.h"
#include "nRF24L01.h"
#include "LIS3MDL.h"

unsigned char tx_buf[TX_PLOAD_WIDTH], I,i,LIS1,LIS2,LIS3;
unsigned short sensor1_status,sensor2_status,sensor3_status;
BYTE LIS_data[SENSORS][CHANNELS][BYTES];

//============================================================================
==
//
//!    main is the main function of the program.
//!    In this function the main initializations and data acquization  will
be done.
//
//============================================================================
==


int main(void)

{
      vInitClock();             // Initializing clock


      SPIB0Init();              // Initializing SPIB0
```

Appendix 1

```
__delay_cycles(5000000);    // wait for 0.625 sec

__delay_cycles(5000000);     // wait for 0.625 sec

      //=====================================================================
=========
      //
      //!Function check_ sensors used to check is the spi communication
between sensors and microcontroller is working
      //!if parameter I1,I2,I3 value is 0x3D then we know sensors are
working. To be used while debugging
      //!to be used only while debugging
      //
      //=====================================================================
=========

      /* unsigned char I1=0,I2=0,I3=0;

      CS_PORT_OUT &= ~(CS_LIS1);

                __delay_cycles(2);         // wait

                LIS_READ(READ+WHO_AM_I);      // READ WHO_AM_I REGISTER OF
LIS1

                I1= LIS_READ(0);

                __delay_cycles(10);    // wait

             CS_PORT_OUT |= (CS_LIS1);


             CS_PORT_OUT &= ~(CS_LIS2);

                       __delay_cycles(2);         // wait

                       LIS_READ(READ+WHO_AM_I);      // READ WHO_AM_I
REGISTER OF LIS1

                       I2= LIS_READ(0);

                       __delay_cycles(10);    // wait

                  CS_PORT_OUT |= (CS_LIS2);


                  CS_PORT_OUT &= ~(CS_LIS3);

                            __delay_cycles(2);         // wait

                            LIS_READ(READ+WHO_AM_I);      // READ WHO_AM_I
REGISTER OF LIS1
```

Appendix 1

```
                                I3= LIS_READ(0);


    __delay_cycles(10);    // wait

                            CS_PORT_OUT |= (CS_LIS3);

*/

       SPIA0Init();                 // Initializing SPIA0(wireless module SPI)

       LISInit() ;                //Initializing LIS 1,2 and 3

       RFinitPins();               //initializin RFmodule pins

       SPI_RW_Reg(FLUSH_RX,0);    // Flushing recieve FIFO for RF
       SPI_RW_Reg(FLUSH_TX,0);    // Flushing transmit FIFO for RF

     clearTX_DS();                           //Clearing TX_DS

       clearMAX_RT();                          //Clearing Max RT

       TX_Mode();                              //Setting TX mode

       SPI_RW_Reg(FLUSH_TX,0);                       // Flushing tranmsmit
FIFO

       vClearBuffer(LIS_data);                       //start with a clean
buffer


       LISInit() ;                //Initializing LIS 1,2 and 3

        clear_intflag();

       _bis_SR_register(LPM3_bits+GIE);  //ENABLES GLOBAL INTERRUPT AND
ENTERS LOPW POWER MODE 3


       while(1)

       {
              unsigned char y=0,x=0;

       if(LIS1==ready)                // check if LIS1 data is ready

       {
              LIS1=notready;


              for ( y=0;y<CHANNELS;y++)

                 {
```

Appendix 1

```
                    for( x=0;x<BYTES;x++)


                         {
                          CS_PORT_OUT &= ~(CS_LIS1);
                          __delay_cycles(2);  //wait
                    LIS_RW(READ + OUT_X_L+x+2*y);
                          LIS_data[0][y][x] = LIS_READ(0);   // Read data
                           __delay_cycles(12);  //wait
                                CS_PORT_OUT |= (CS_LIS1);
                    }
               }
}


if(LIS2==ready)                    // check if LIS2 data is ready
    {
         LIS2=notready;
   for ( y=0;y<CHANNELS;y++)
              {
              for( x=0;x<BYTES;x++)
               {
                    CS_PORT_OUT &= ~(CS_LIS2);
                    __delay_cycles(2);  //wait
                LIS_RW(READ + OUT_X_L+x+2*y);// Read data
              LIS_data[1][y][x] = LIS_READ(0);
              _delay_cycles(12);  //wait
              CS_PORT_OUT |= CS_LIS2;
              }
               }
}
```

Appendix 1

```
        if(LIS3==ready)                    // check if LIS3 data is ready
        {
               LIS3=notready;


          for ( y=0;y<CHANNELS;y++)

                       {

                       for( x=0;x<BYTES;x++)

                       {

                              CS_PORT_OUT &= ~(CS_LIS3);

                              __delay_cycles(2);  //wait

                            LIS_RW(READ + OUT_X_L+x+2*y);// Read data

                              LIS_data[2][y][x] = LIS_READ(0);

                              _delay_cycles(12);  //wait

                     CS_PORT_OUT |= CS_LIS3;

                }
                       }

        }


               unsigned char h,i,j,k=0,status;

               for(h=0;h<SENSORS;h++)
               {

                       for(i=0;i<CHANNELS;i++)

                       {

                              for(j=0;j<BYTES;j++)

                              {

                       tx_buf[k]=LIS_data[h][i][j];  // copying data from
LIS_data to tx_buf

                       k++;
                             }
                    }
                }
```

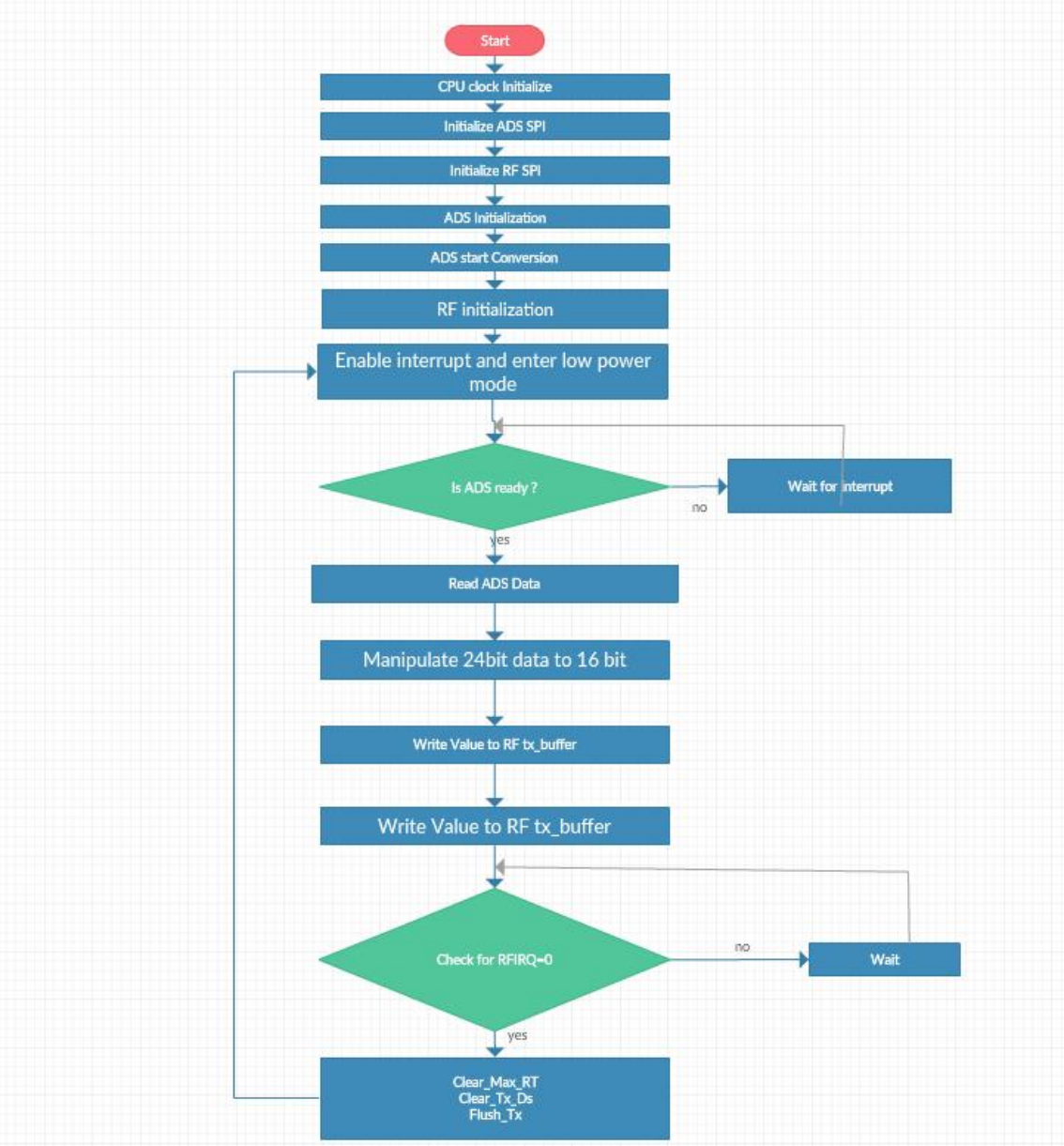
Appendix 1

```
            k=0;

                SPI_Write_Buf(WR_TX_PLOAD, tx_buf, TX_PLOAD_WIDTH);   //
writing the payload data



                CE_Pulse();               //start transmit


                while (GEN_PORT_IN & RF_IRQ);  //wait for RF_IRQ


                status=readStatus();
                if(status & 0x10)        //look for MAX_RT
                      status=clearMAX_RT();
                if(status & 0x20)       //look for TX_DS
                      status=clearTX_DS();
                SPI_RW_Reg(FLUSH_TX,0);  //flush TX FIFO


                vClearArray( tx_buf, TX_PLOAD_WIDTH);  //clear tx_buf
                vClearBuffer(LIS_data);              //clear LIS_DATA
                _bis_SR_register(LPM3_bits+GIE);  //ENABLES GLOBAL
INTERRUPT AND ENTERS LOW POWER MODE 3
      }
}
```

Appendix 1

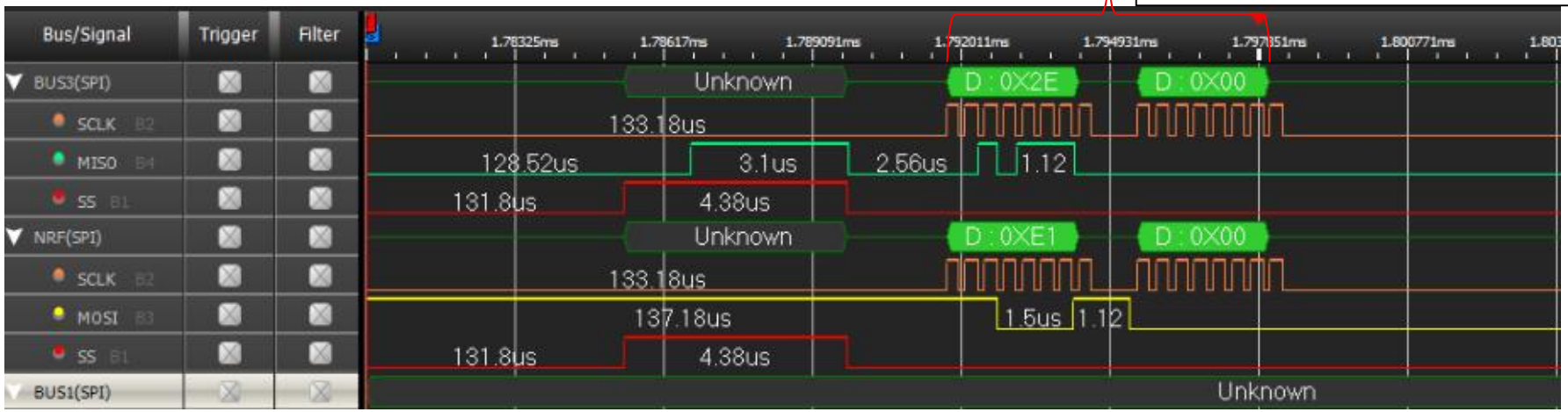
```
//====================================================================
==
//
//!    PRAGMA VECTOR PORT ONE IS THE INTERRUPT SERVICE ROUTINE
//!    THE PROGRAM ENTERS THIS ROUTINE IF ANY INTERRUPTS HAS HAPPENED IN PORT
1
//!    IN THIS ROUTINE THE PROGRAM MODIFIES THE STATUS OF THE SENSORS TO
READY
//!    ALSO THE ROUTINE CLEARS THE INTERRUPT FLAGS
//
//====================================================================
==


#pragma vector=PORT1_VECTOR

__interrupt void PORT_1(void)

{


if(P1IFG&LIS1_DRDY)                // check for interrupt flag
{


P1IFG&=~LIS1_DRDY;

                    LIS1=ready;


}

if(P1IFG&LIS2_DRDY)                // check for interrupt flag
{

      P1IFG&=~LIS2_DRDY;
      LIS2=ready;


}

if(P1IFG&LIS3_DRDY)                // check for interrupt
{


       P1IFG&=~LIS3_DRDY;

       LIS3=ready;
}


      _bic_SR_register_on_exit(LPM3_bits+GIE);


}
```

Appendix 2. Code Flow Diagram

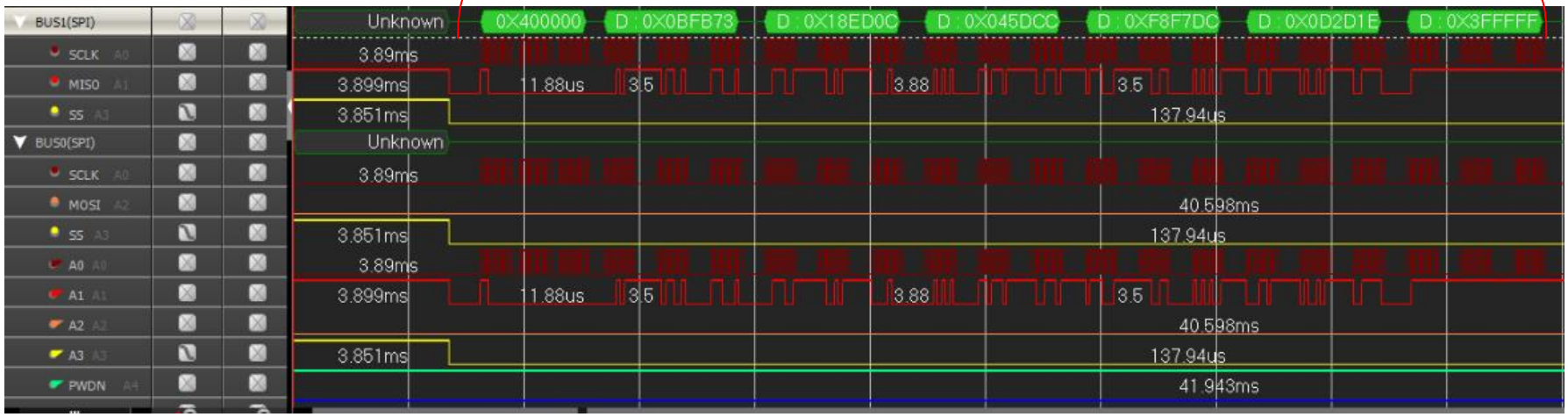


Appendix 3. SPI Communication Test for Phase 1 Knee PCB

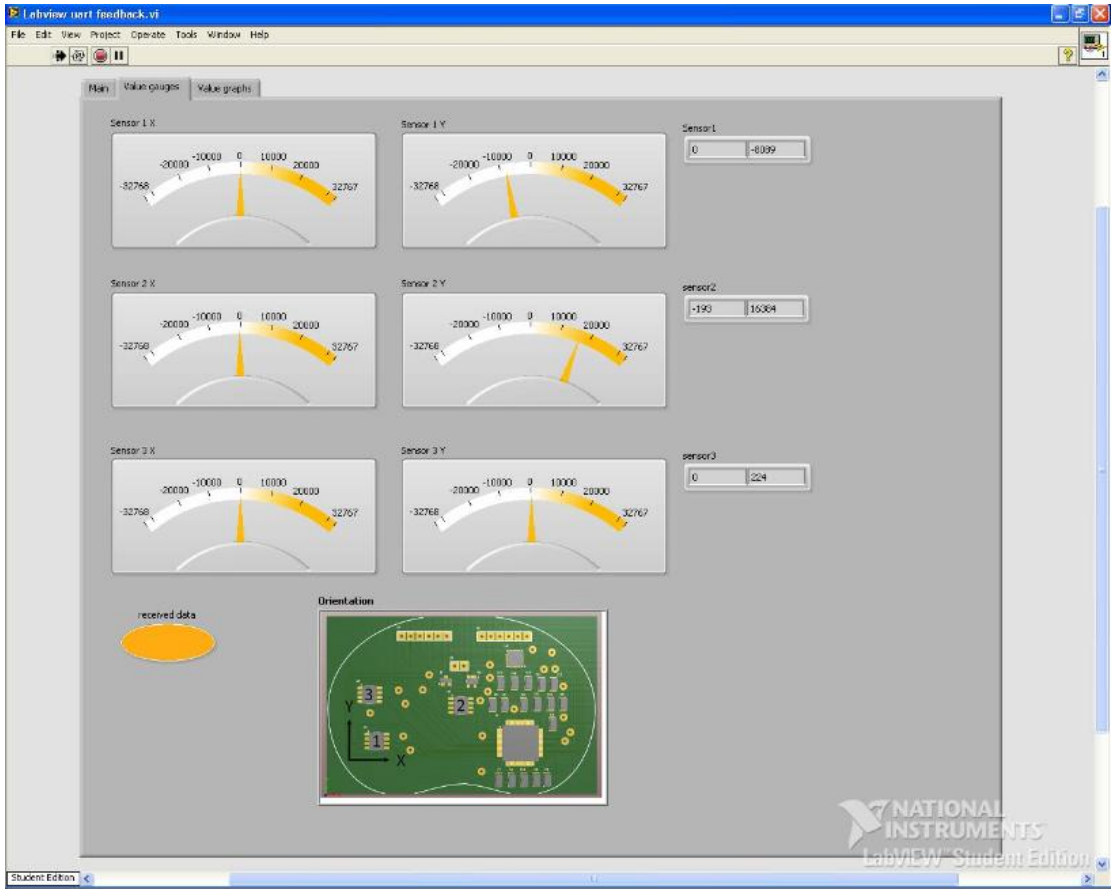
Flush TX buffer (0xE1-→0x00)

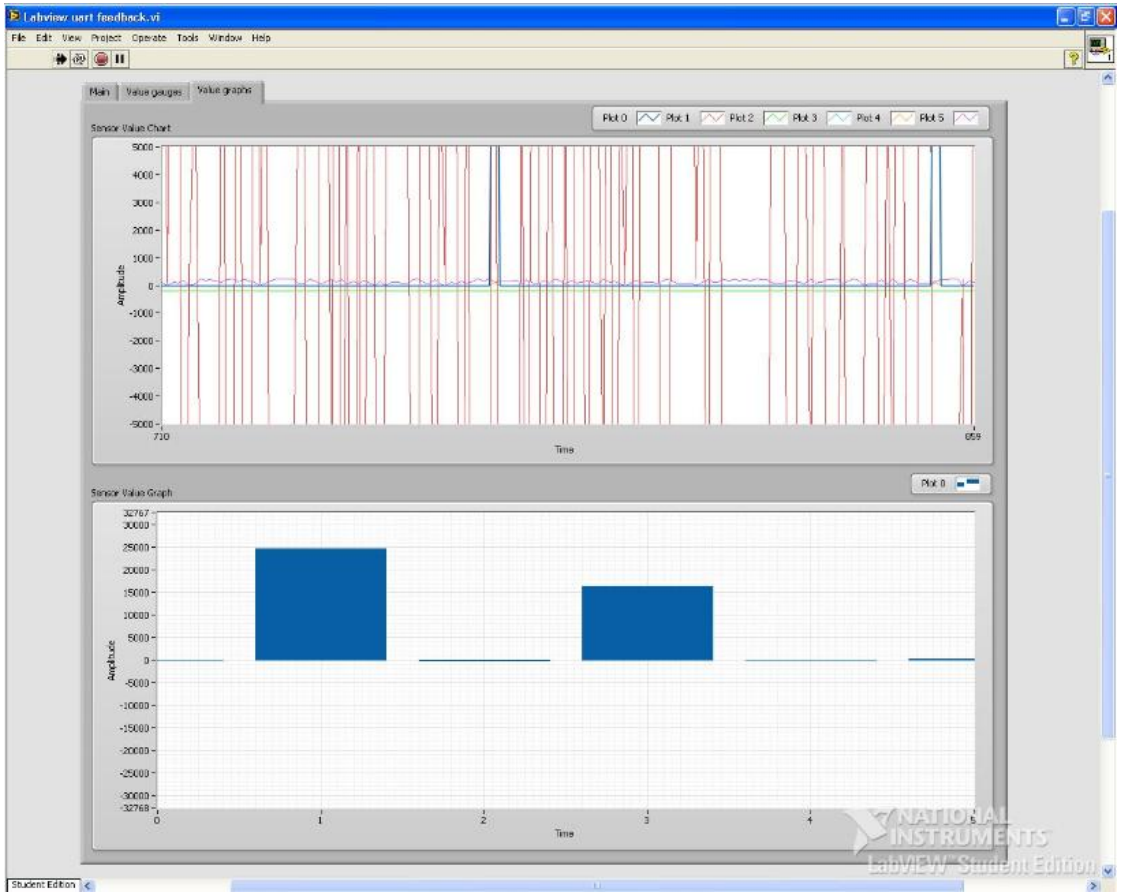


Receiving Data from ADS module



Appendix 4. LabView Test of Phase 1 of the PCB




Labview uart feedback.vi
File Edit View Project Operate Tools Window Help
Main
Value gauges
Value graphs
Sensor Value Chart
Plot 0
Plot 1
Plot 2
Plot 3
Plot 4
Plot 5
Amplitude
Time
Sensor Value Graph
Plot 0
Amplitude
Time
NATIONAL INSTRUMENTS
LabVIEW Student Edition
Student Edition

Appendix 5. Test Data when Full Scale is 4 Gauss

A) Set Up 1 4 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	0178	015E	0179	0182	017E	014F	0179	0153	014A	014A	
	In Decimal	376	350	377	386	382	335	377	339	330	330	358.20
	In milliGauss	54.955	51.155	55.101	56.416	55.832	48.962	55.101	49.547	48.232	48.232	52.35
Y Axis	In HEX	0016	0035	002B	002E	0019	0030	0031	003C	0033	0033	
	In Decimal	22	53	43	46	25	48	49	60	51	51	44.80
	In milliGauss	3.215	7.746	6.285	6.723	3.654	7.016	7.162	8.769	7.454	7.454	6.55
Z Axis	In HEX	3155	3194	3176	3149	313D	3175	30F3	3156	3156	30EC	
	In Decimal	12629	12692	12662	12617	12605	12661	12531	12630	12630	12524	12618.10
	In milliGauss	1845.805	1855.013	1850.629	1844.051	1842.298	1850.482	1831.482	1845.952	1845.952	1830.459	1844.21

B) Set Up 2 4 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	4F69	4F8D	4F10	4E65	4E03	4E0F	4E36	4E82	4E3F	4DE6	
	In Decimal	20329	20365	20240	20069	19971	19983	20022	20098	20031	19942	20105.00
	In milliGauss	2971.207	2976.469	2958.199	2933.207	2918.883	2920.637	2926.337	2937.445	2927.653	2914.645	2938.47
Y Axis	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	IN mliliGAUSS	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.10
Z Axis	In HEX	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	
	In Decimal	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768.00
	In milliGauss	-4789.243	-4789.243	-4789.243	-4789.243	-4789.243	-4789.243	-4789.243	-4789.243	-4789.243	-4789.243	-4789.24

C) Set Up 3 4 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	AC7F	ACE4	AD53	ADE6	AF11	AFD1	B04F	B0AB	B0F9	B135	
	In Decimal	-21377	-21276	-21165	-21018	-20719	-20527	-20401	-20309	-20231	-20171	-20719.40
	In milliGauss	-3124.379	-3109.617	-3093.394	-3071.909	-3028.208	-3000.146	-2981.731	-2968.284	-2956.884	-2948.115	-3028.27
Y Axis	In HEX	D22F	D2C3	D3A0	D4BE	D747	D8AD	DA0E	DB4D	DCD1	DDF2	
	In Decimal	-11729	-11581	-11360	-11074	-10425	-10067	-9714	-9395	-9007	-8718	-10307.00
	In milliGauss	-1714.265	-1692.634	-1660.333	-1618.533	-1523.677	-1471.353	-1419.760	-1373.137	-1316.428	-1274.189	-1506.43
Z Axis	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	IN milli GAUSS	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.10

D) Set Up 4 4 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	In milliGauss	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.10
Y Axis	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	In milliGauss	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.10
Z Axis	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	In milliGauss	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.097	4789.10

Appendix 6. Test Data when Full Scale is 12 Gauss

A) Set Up 1 12 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	00AE	00B0	00A5	00AE	00A7	00AE	00AE	00AA	00AE	00AD	
	In Decimal	174	176	165	174	167	174	174	170	174	173	172.10
	In milliGauss	76.282	77.159	72.337	76.282	73.214	76.282	76.282	74.529	76.282	75.844	75.45
Y Axis	In HEX	002A	0025	001E	0023	001F	0023	0022	001B	001B	0019	
	In Decimal	42	37	30	35	31	35	34	27	27	25	32.30
	In milliGauss	18.413	16.221	13.152	15.344	13.591	15.344	14.906	11.837	11.837	10.960	14.16
Z Axis	In HEX	1148	1127	112C	1150	113E	112E	1121	113C	1115	1152	
	In Decimal	4424	4391	4396	4432	4414	4398	4385	4412	4373	4434	4405.90
	In milliGauss	1939.500	1925.033	1927.225	1943.008	1935.116	1928.102	1922.403	1934.239	1917.142	1943.884	1931.57

B) Set Up 2 12 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	
	In Decimal	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768.00
	In milliGauss	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.63
Y Axis	In HEX	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	
	In Decimal	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768.00
	In milliGauss	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.63
Z Axis	In HEX	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	
	In Decimal	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768.00
	In milliGauss	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.63

C) Set Up 3 12 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	B0C7	B118	B165	B196	B1B7	B1C5	B1EB	B1F6	B1DB	B19A	
	In Decimal	-20281	-20200	-20123	-20074	-20041	-20027	-19989	-19978	-20005	-20070	-20078.80
	In milliGauss	-8891.276	-8855.765	-8822.007	-8800.526	-8786.058	-8779.921	-8763.262	-8758.439	-8770.276	-8798.7725	-8802.63
Y Axis	In HEX	286F	285D	283D	2824	2819	2809	27CF	27BD	27A3	2780	
	In Decimal	10351	10333	10301	10276	10265	10249	10191	10173	10147	10112	10239.80
	In milliGauss	4537.922	4530.030	4516.002	4505.041	4500.219	4493.205	4467.777	4459.886	4448.487	4433.1434	4489.17
Z Axis	In HEX	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	
	In Decimal	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768.00
	In milliGauss	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.629	-14365.63

D) Set Up 4 12 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	In milliGauss	14365.191	14365.191	14365.191	14365.191	14365.191	14365.191	14365.191	14365.191	14365.191	14365.1907	14365.19
Y Axis	In HEX	F151	F158	F14E	F154	F157	F158	F158	F158	F14E	F157	
	In Decimal	-3759	-3752	-3762	-3756	-3753	-3752	-3752	-3752	-3762	-3753	-3755.30
	In milliGauss	-1647.961	-1644.893	-1649.277	-1646.646	-1645.331	-1644.892	-1644.892	-1644.892	-1649.276	-1645.3310	-1646.34
Z Axis	In HEX	BF59	BF7E	BF99	BFEB	BFE7	BFEC	C027	C00C	C053	C070	
	In Decimal	-16551	-16514	-16487	-16405	-16409	-16404	-16345	-16372	-16301	-16272	-16406.00
	In milliGauss	-7256.028	-7239.807	-7227.970	-7192.021	-7193.775	-7191.583	-7165.717	-7177.554	-7146.427	-7133.7133	-7192.46

Appendix 7. Test Data when Full Scale is 16 Gauss

A) Set Up 1 16 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	FFDF	FFE7	FFE5	FFE1	FFE1	FFE1	FFE2	FFE1	FFE0	FFE5	
	In Decimal	-33	-25	-27	-31	-31	-31	-30	-31	-32	-27	-29.80
	In milliGauss	-19.2870	-14.6113	-15.7802	-18.1181	-18.1181	-18.118	-17.534	-18.118	-18.703	-15.7802	-17.42
Y Axis	In HEX	FFC9	FFC8	FFC8	FFCA	FFBF	FFC9	FFD8	FFCC	FFC4	FFC8	
	In Decimal	-55	-56	-56	-54	-65	-55	-40	-52	-60	-56	-54.90
	In milliGauss	-32.145	-32.729	-32.729	-31.561	-37.990	-32.145	-23.378	-30.392	-35.067	-32.729	-32.09
Z Axis	In HEX	0A7D	0A58	0A7B	0A69	0A6F	0A67	0A6B	0A6F	0A69	0A68	
	In Decimal	2685	2648	2683	2665	2671	2663	2667	2671	2665	2664	2668.20
	In milliGauss	1569.258	1547.633	1568.089	1557.569	1561.075	1556.400	1558.738	1561.075	1557.569	1556.984	1559.44

B) Set Up 2 16 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	CF80	CFDB	D01B	D052	D06B	D07D	D070	D044	D028	D016	
	In Decimal	-12416	-12325	-12261	-12206	-12181	-12163	-12176	-12220	-12248	-12266	-12246.20
	In milliGauss	-7256.575	-7203.390	-7165.985	-7133.840	-7119.229	-7108.708	-7116.306	-7142.022	-7158.387	-7168.907	-7157.33
Y Axis	In HEX	1BA0	1BD5	1BEC	1BD8	1BC8	1BB3	1BB5	1BB2	1BAB	1BAC	
	In Decimal	7072	7125	7148	7128	7112	7091	7093	7090	7083	7084	7102.60
	In milliGauss	4133.255	4164.231	4177.674	4165.985	4156.634	4144.360	4145.529	4143.776	4139.684	4140.269	4151.14
Z Axis	In HEX	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	
	In Decimal	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768	-32768.00
	In milliGauss	-19151.374	-19151.374	-19151.374	-19151.374	-19151.374	-19151.374	-19151.374	-19151.374	-19151.374	-19151.374	-19151.37

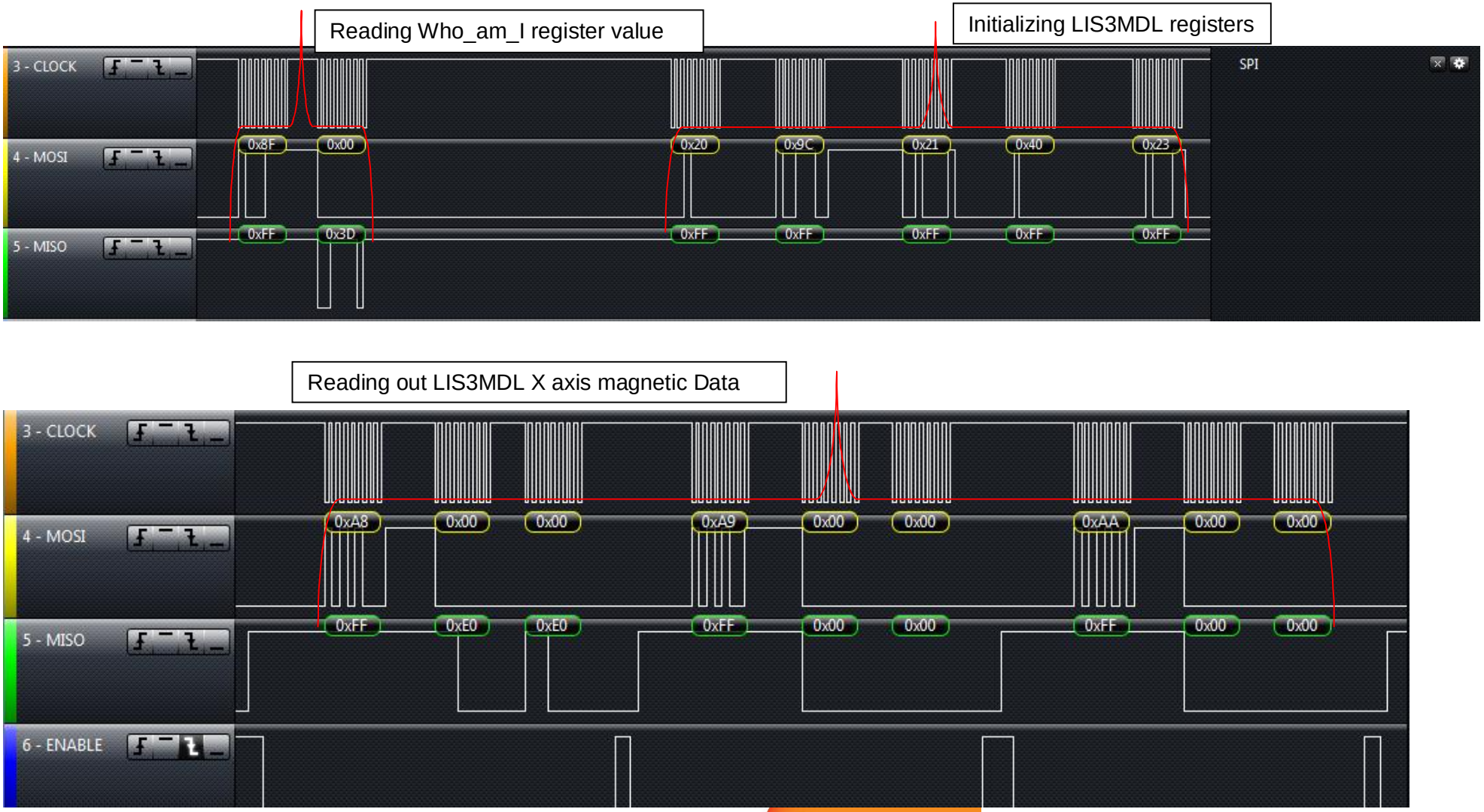
C Set Up 3 16 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	D7AD	D7E5	D836	D82B	D81B	D814	D7FB	D7EC	D7EB	D7F1	
	In Decimal	-10323	-10267	-10186	-10197	-10213	-10220	-10245	-10260	-10261	-10255	-10242.70
	In milliGauss	-6033.3139	-6000.5845	-5953.2437	-5959.6727	-5969.0240	-5973.1151	-5987.7265	-5996.4933	-5997.0777	-5993.5710	-5986.38
Y AXIS	In HEX	D364	D39D	D3E2	D416	D420	D41A	D3D7	D3AF	D39B	D3B1	
	In Decimal	-11420	-11363	-11294	-11242	-11232	-11238	-11305	-11345	-11365	-11343	-11314.70
	IN milli GAUSS	-6674.459	-6641.146	-6600.818	-6570.427	-6564.582	-6568.089	-6607.247	-6630.625	-6642.314	-6629.457	-6612.92
Z AXIS	In HEX	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	7FFF	
	In Decimal	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767	32767.00
	In milliGauss	19150.789	19150.789	19150.789	19150.789	19150.789	19150.789	19150.789	19150.789	19150.789	19150.789	19150.79

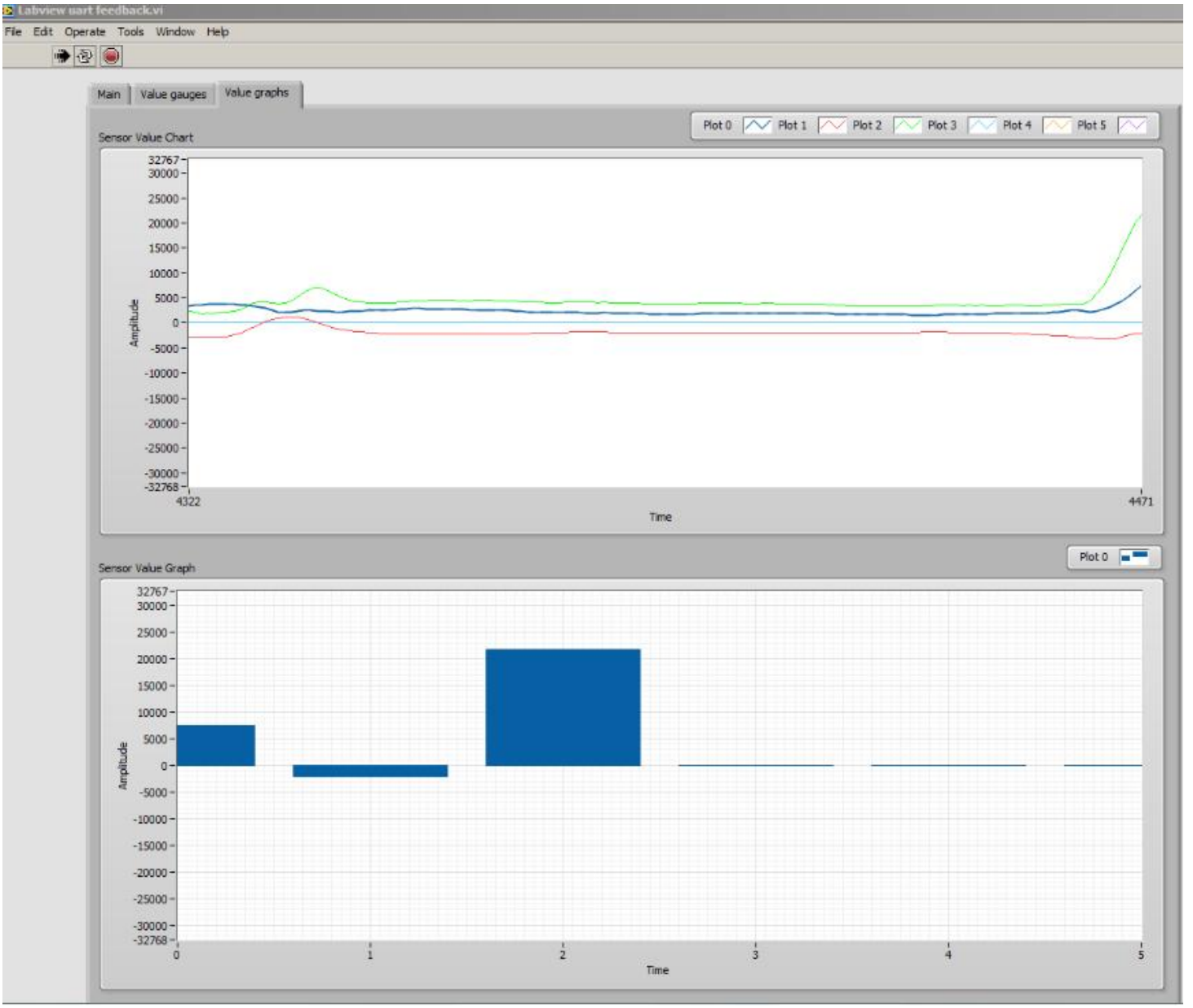
D) Set Up 4 16 Gauss Full Scale

		sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10	AVERAGE
X Axis	In HEX	ED93	ED8E	EDA1	EDA0	ED88	ED7F	ED71	ED76	ED80	ED97	
	In Decimal	-4717	-4722	-4703	-4704	-4728	-4737	-4751	-4746	-4736	-4713	-4725.70
	In milliGauss	-2756.8673	-2759.7896	-2748.6850	-2749.2694	-2763.2963	-2768.5564	-2776.7387	-2773.8165	-2767.9719	-2754.5295	-2761.95
Y Axis	In HEX	E836	E82C	E83D	E836	E829	E81A	E807	E7FF	E7EA	E7EC	
	In Decimal	-6090	-6100	-6083	-6090	-6103	-6118	-6137	-6145	-6166	-6164	-6119.60
	In milliGauss	-3559.3220	-3565.1666	-3555.2309	-3559.3220	-3566.9199	-3575.6867	-3586.7914	-3591.4670	-3603.7405	-3602.5716	-3576.62
Z Axis	In HEX	684B	6837	67F6	67EC	6818	687A	68C8	6922	693A	6936	
	In Decimal	26699	26679	26614	26604	26648	26746	26824	26914	26938	26934	26760.00
	In milliGauss	15604.3250	15592.6359	15554.6464	15548.8019	15574.5178	15631.7943	15677.3816	15729.9825	15744.0094	15741.6715	15639.98

Appendix 8. SPI Communication Test For Phase 1 Knee PCB

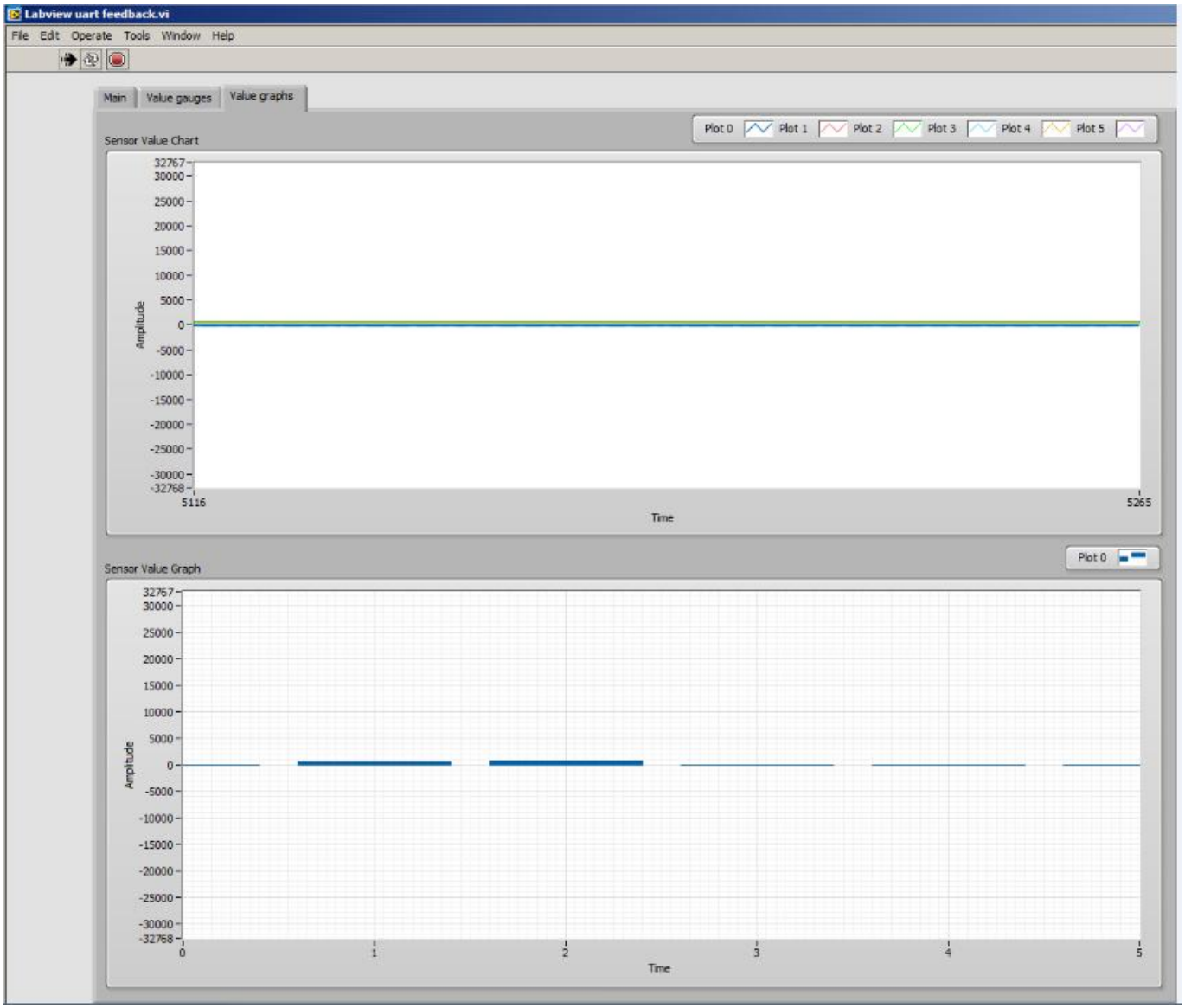


Appendix 9. LabView Test of Phase 2 of the PCB



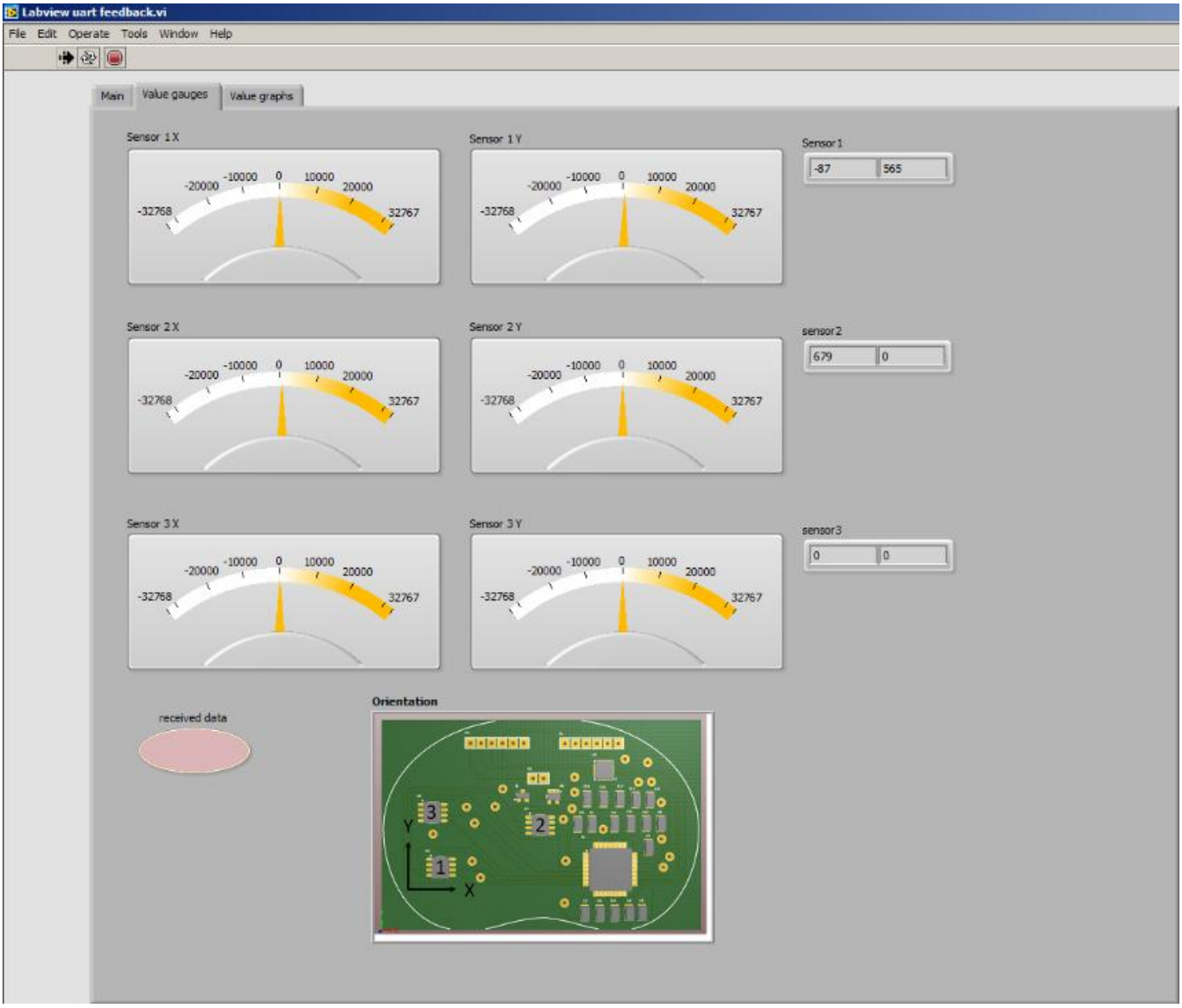
Magnetic field value graph and chart when there is a magnet from certain distance

Appendix 9



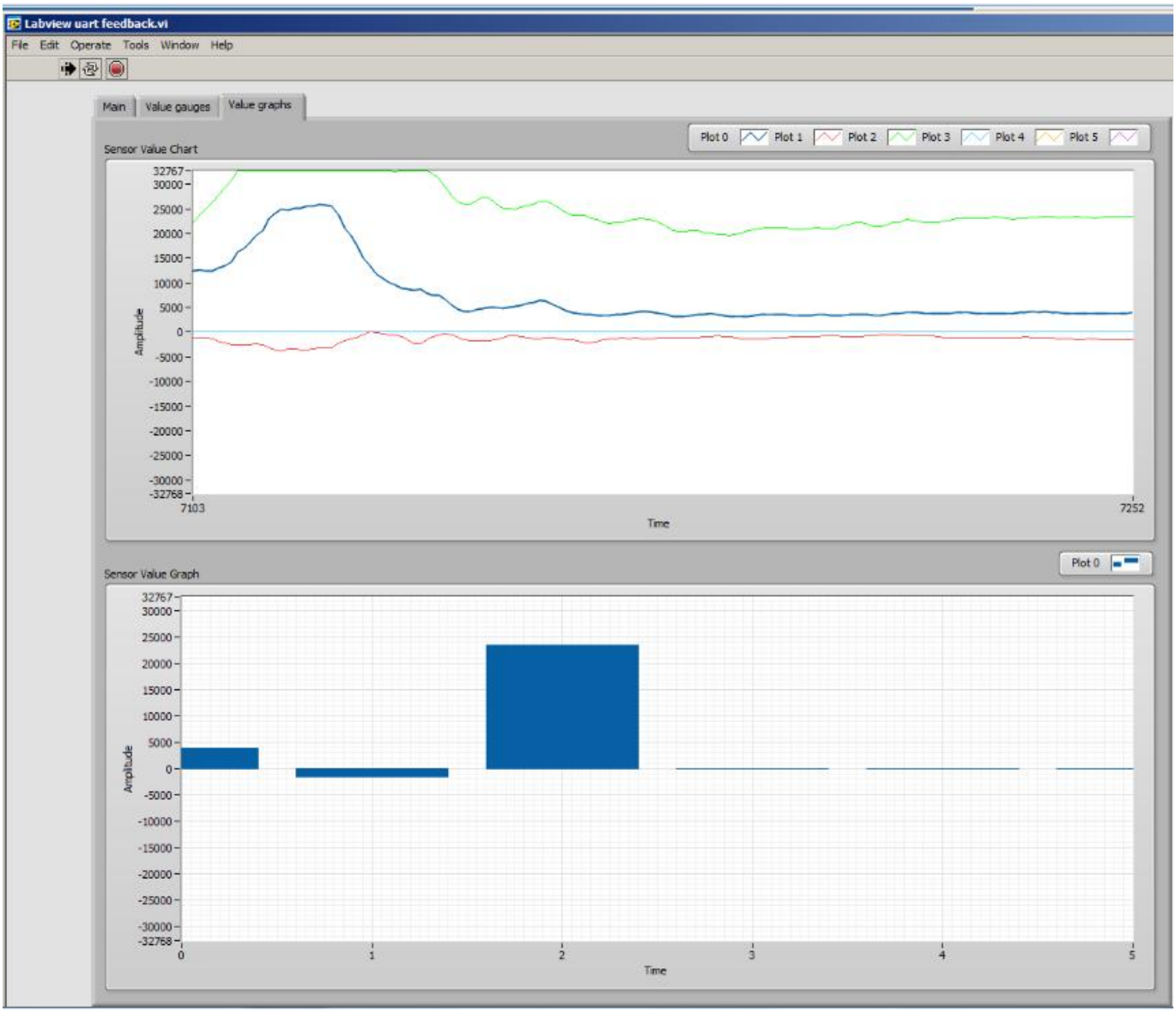
Magnetic field value graph and chart when there is no magnet applied

Appendix 9



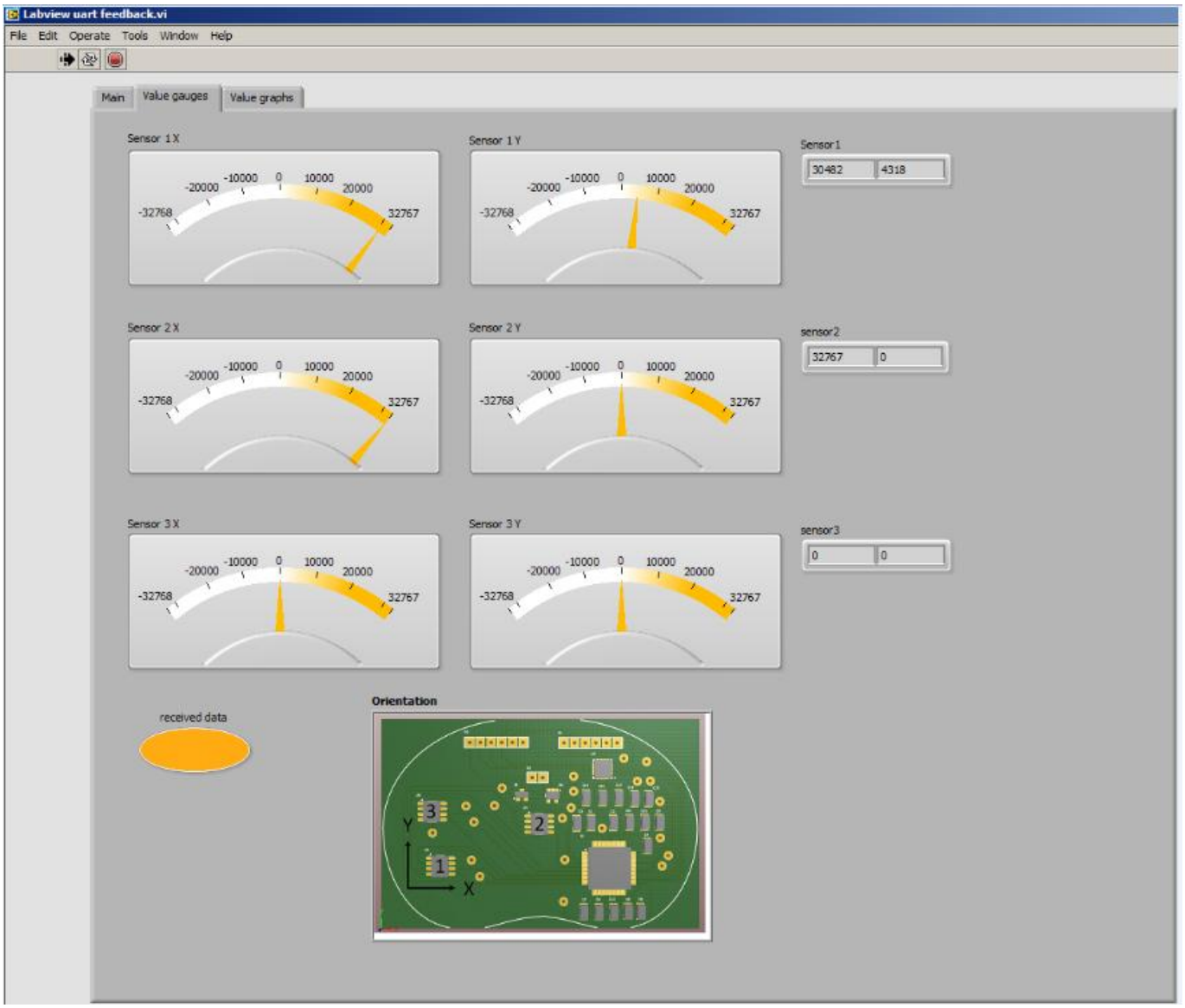
Sensor values when there is no magnet applied

Appendix 9



Magnetic field values along the z axis clipped

Appendix 9



Magnetic field values along the x and y axis over flown