ECE 445: Senior Design

FLEXIBLE ELECTRONICS VITALS SENSOR

March 11, 2013

Matthew Frank
Russell Geschrey
Project 15
TA: Justine Fortier
Professor Paul Carney

Contents

1 Introduction 3
1.1 Purpose 3
1.2 Objectives 3
1.2.1 Features 3
1.2.2 Benefits 3

2 Design Overview 4
2.1 Sensor 4
2.1.1 LF Wireless Interface and Microcontroller Pin Configurations 5
2.1.2 Operation of the TPS71433 (DC Power Supply System) 5
2.1.3 Operation of the TMS37157 (LF Wireless Interface) 6
2.1.4 Interface Between Microcontroller and LF Wireless interface 8
2.1.5 Interface Between Microcontroller and Sensors 8
2.1.5.1 MCP9808 Temperature Sensor 9
2.1.5.2 ADXL344 Ultra Low Power Accelerometer 10
2.1.5.3 ADS1191 IBIS Biopotential Measurement Chip 11
2.1.6 Microcontroller firmware and implementation 12
2.2 Transceiver 14
2.2.1 Transmission of Power and Data 14
2.2.2 Data Flow Diagram 15
2.2.3 Code Process 15

3 Requirements and Verification 17
3.1 Data Sensing and Manipulation 17
3.1.1 Sensors 17
3.1.2 Microcontroller 20
3.2 TPS71433 (DC Power Supply System) 21
3.3 PaLFI System 23
3.4 Tolerance Analysis 23

4 Ethical Considerations 24

5 Safety Considerations 25

6 Cost and Schedule 25
6.1 Cost Analysis 25
6.1.1 Labor 25
6.1.2 Parts 25
6.2 Timeline 26

7 References 28

1 Introduction

1.1 Purpose

This project was chosen because of an interest in wireless communication systems, namely BAN's (body area networks) and their recent integration with flexible electronics research here at the University of Illinois. This area of research, if successful, has the potential to greatly change health care and advance the field of bioinformatics.

1.2 Objectives

The goal is to create a small flexible electronics vitals sensor, about the size of an average bandage, which can measure a person's vitals such as ECG (heart rate), body movement and core body temperature. It will relay this information over the air in real time to a receiver, where it will be displayed in a graph format and stored for later analysis (such as for medical studies).

1.2.1 Features

Medical products such as heart rate monitors and temperature sensors are not uncommon in today's market. However, most of these products are relatively large, and require various wires which inhibit movement, or a battery which needs to be replaced and takes up a large volume of the device. The flexible electronics monitor is wire-free. In addition, its physical profile is low, which allows it to be used without getting in the way.

1.2.2 Benefits

- Allows for mobility while still being able to monitor patients.
- Less intrusive and more seamlessly integrated into everyday tasks.
- Capable of being used where previous "rigid" devices were unable to operate.
- Allows for studies that could not normally be performed with wired devices.
- Does not require battery replacement, which increases reliability and gives more flexibility.

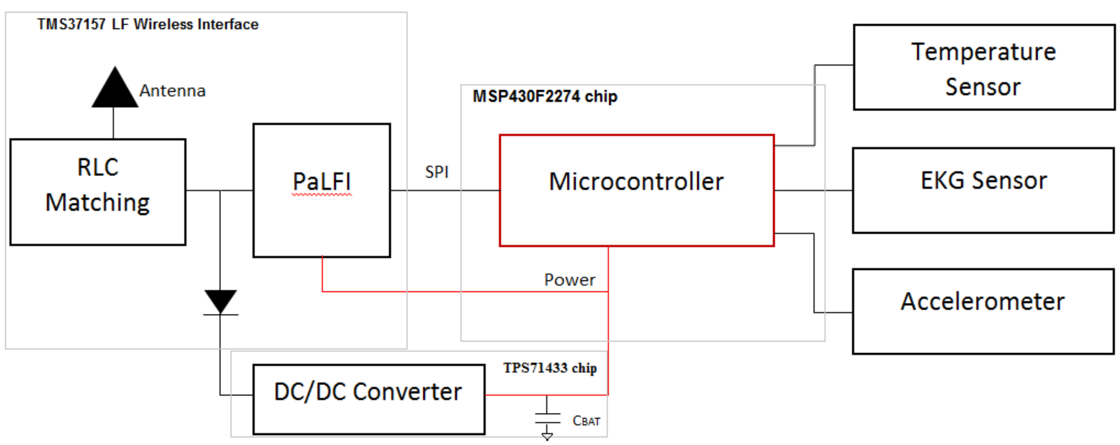
2 Design Overview

2.1 Sensor

Figure 1 is a top level block diagram of the sensing system. The PaLFI (**Pa**ssive **L**ow **F**requency **I**nterface) communication device is the source of data communication between the sensor and the receiver. It is connected via a 3 wire SPI interface. The microcontroller performs data sensing from the three sensors (heart rate, EKG, and temperature) and reports it to the PaLFI device when it is commanded. The DC / DC converter provides the power necessary for the sensor from the receiver's power transmission over RFID frequencies. Much more information about the interconnections will be presented in the following sections. To sum up, the sensor's operation is divided in four steps:

- Charge phase: Generate an RF field of 134.2 kHz from the reader to the wireless sensor module to charge the power capacitor.
- Downlink phase: Send command to wireless sensor to start measurement.
- Measurement and recharge phase: Trigger three measurements from microcontroller; recharge the power capacitor on the sensor device.
- Uplink phase: Send measurement results via RF interface (134.2 kHz) back to reader

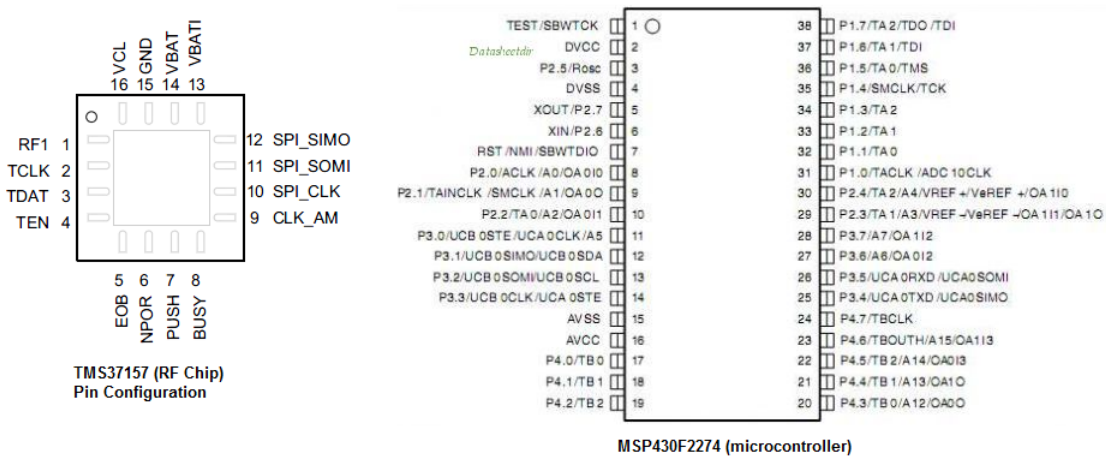
Figure 1: Top Level Block Diagram



2.1.1 LF Wireless Interface and Microcontroller Pin Configurations

The sections following this one are in-depth explanations of the performance of the LF Wireless Interface and the microcontroller. The figure below is the pin configuration for the following sections. This will help to clarify to the reader what pins are receiving what signals, and to be used as a general debugging guide.

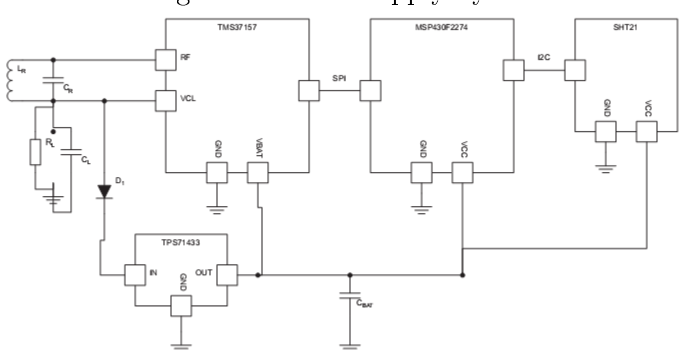
Figure 2: Pin configurations for the MSP430F2274 and TMS37157



2.1.2 Operation of the TPS71433 (DC Power Supply System)

The transmitter uses an external DC/DC converter attached to V_{CL} to generate a V_{BAT} / V_{CC} voltage out of the 134.2 kHz RF field. Figure 3 shows the basic principle of this circuit.

Figure 3: Power Supply System



The input of the DC/DC converter TPS71433 is connected to VCL via diode D_1. D_1

prevents the resonance circuit (consisting of L_R and C_R) from any disturbances coming from the dc/dc converter. Capacitor C_{BAT} stores the energy derived from the RF field. Using an external DC/DC converter instead of the internal of the TMS37157 overcomes two issues. The first advantage of an external DC/DC converter is that it can provide higher output currents in comparison to the internal regulator (80 mA compared to 5 mA). The second advantage using an external regulator is the simpler flow for the application and the firmware (see Table 1).

Table 1: Comparison of Internal vs. External Regulator

	Using Internal Regulator	Using external regulator
1	Charge Phase	Charge Phase
2	Send first MSP command to trigger measurement	Send MSP access command to trigger measurement
3	Send battery charge command to enable internal regulator to supply MSP and sensor	Keep RF field to supply MSP and sensor
4	Send second MSP access command to retrieve data from TMS37157	Disable RF field and wait for response from TMS37157

2.1.3 Operation of the TMS37157 (LF Wireless Interface)

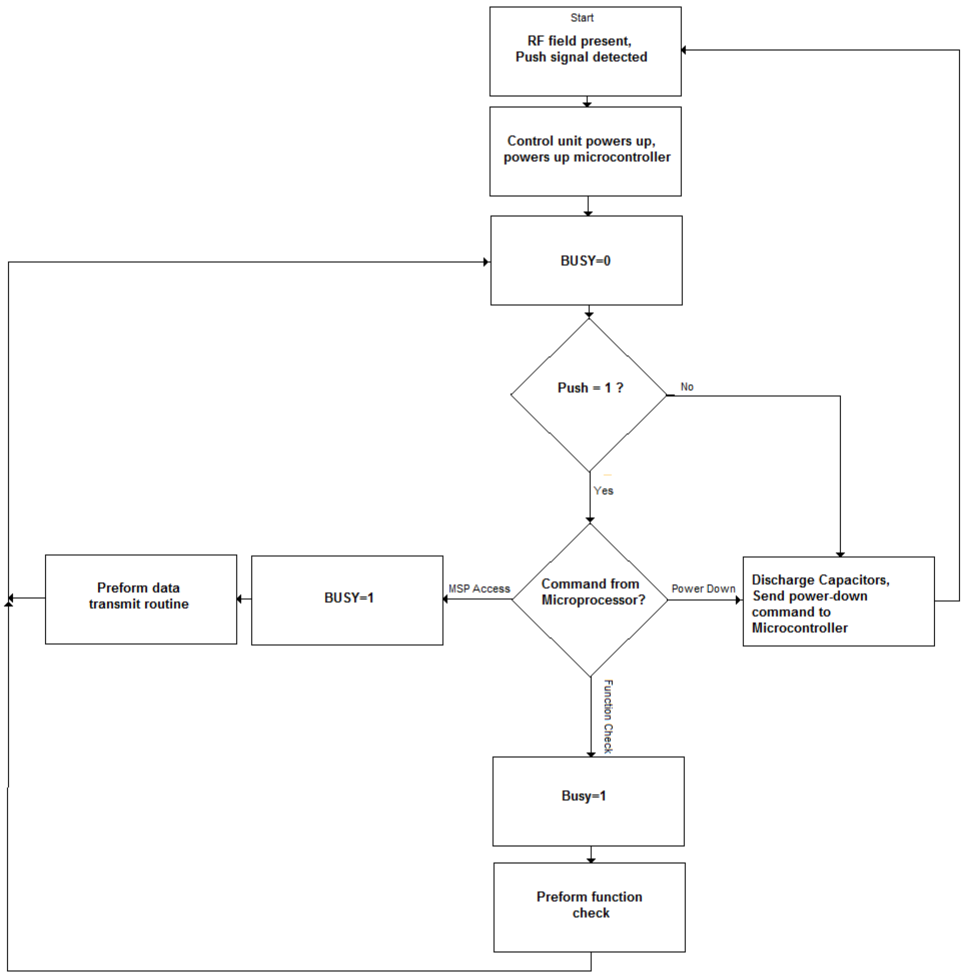
The following is a brief explanation of the TMS37157 and PaLFI operations. While the PaLFI preforms many functions, this section will talk about the functions that are used for our design.

The device is powered on when the activation of a push signal is detected by an ultra low-power detection circuit. While waiting for a high signal at PUSH, the only active component in the PaLFI is a flip-flop, whose output is set when PUSH is set high. When this happens, the Control Unit is powered up and initialized. Also a command is sent to the DC/DC converter to power up the microcontroller. The Microcontroller can, after performing its desired actions, send a Power Down Command to the TMS37157, bringing the TMS37157 back down to the ultra low power mode (more on this in the next section).

The LF Wireless Interface contains all of the instructions to preform its operations in its EEPROM. When the BUSY signal is zero, the microcontroller is able to take in commands from the user at the transmitter. If no commands are given, the microcontroller cycles through a “push" routine until it is given a command or the RF field is powered down and the PUSH signal is no longer present.

Figure 4 (seen below) is the flow chart of the PaLFI used for our design. While the LF Wireless Interface itself has many control words, for the sake of simplicity it is broken into three main categories:

Figure 4: Flow chart of PaLFI Operations

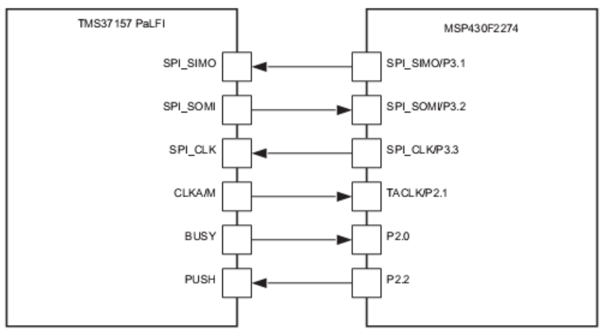


- The “Function Check" branch means a command was given to the LF Wireless Interface by the transmitter for it to report data on parts of the component (check to see if power levels are stable, ping the LF Wireless Interface, so on).
- “Power Down" branch gives the command to set the device into ultra low power mode.
- “MPS Access" is the command which will be the focus of the remainder of the report. This command will allow the mirocontroller to access the memory of the LF Wireless Interface and execute a program written to its memory. The program allows the chip to cycle through a transmit/charge/measure cycle over and over until the base station gives the command to halt the process.

2.1.4 Interface Between Microcontroller and LF Wireless interface

Figure 5 shows the data interface between MSP430F2274 (microcontroller) and TMS37157 (LF Wireless Interface). The LF Wireless Interface is connected to the microcontroller through a 3-wire SPI interface. The PaLFI takes all instructions from these three inputs, including the "MPS Access" command. To simplify communication between the microcontroller and LF Wireless Interface, the BUSY pin of the LF Wireless Interface is connected to the microcontroller. The BUSY pin indicates the readiness of the PaLFI to receive the next data byte from the sensors (which is given by the microcontroller). CLKAM is used for the antenna automatic tune feature. One of the functions in the "functions check" is the "autotrim routine". This routine uses the antenna as a voltage probe and tunes a variable capacitor on the PaLFI to ensure for optimum performance. This command can be called for manually, but it can also be called on in other functions (such as within the "MPS access" command).

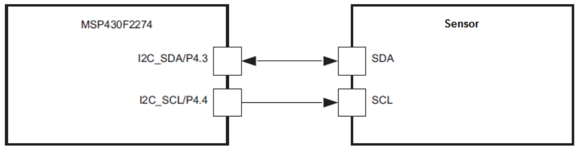
Figure 5: Microcontroller - PaLFI Data Interface



2.1.5 Interface Between Microcontroller and Sensors

This and the following sections describe the operation of each sensor and how it interfaces with the microcontroller, and ultimately the TMS37157 LF Wireless Interface. The microcontroller for this design was chosen for two reasons. Firstly, it consumes extremely low amounts of power compared to most on the market today (270 μA in active mode and 0.7 μA in standby). The second being that it supports a large amount of pins that can be configured to communicate data from various sensors. Figure 6 shows the common configuration between all of the sensors and the microcontroller.

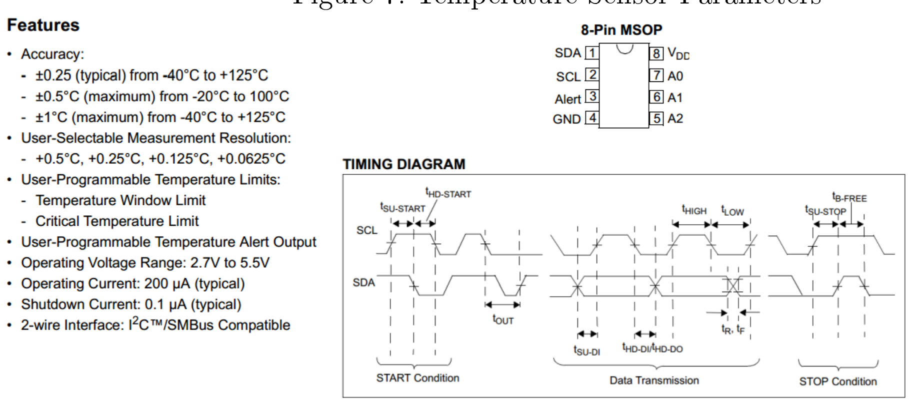
Figure 6: Microcontroller - Sensor interface



Each sensor is equipped with two I2C pin switches (SDA and a SCL)which, when given the correct control sequence will power up, measure, and power down. The corresponding data will then be stored into the LF wireless interface's memory and await its turn for transmission. The P1.x/P2.x as well as the P3.x/P4.x are all I/O ports that can be configured to take inputs and transmit data. Their specific uses and configurations will be discussed in the firmware section (where their values are programmed into the microcontroller).

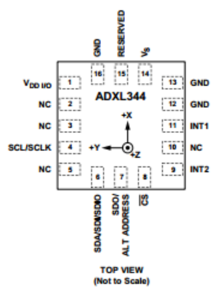
2.1.5.1 MCP9808 Temperature Sensor The chip used for temperature sensing is the MCP9808 ±.5°C Maximum Accuracy Digital Temperature Sensor. This chip was chosen because of its low power consumption, its 2-wire Interface: I2C/SMBus Compatibility (mentioned above) and its digital output. All of the essentials of the chip are displayed below.

Figure 7: Temperature Sensor Parameters



2.1.5.2 ADXL344 Ultra Low Power Accelerometer The accelerometer used for this design is the ADXL344 is a versatile 3-axis, digital-output, accelerometer. The device was chosen because its selectable measurement range and bandwidth allow it to be configured for low power (yet still useful) measurements. All of the necessary data on the accelerometer is displayed below.

Figure 8: Accelerometer Sensor Parameters



NOTES
1. NC = NO INTERNAL CONNECTION.

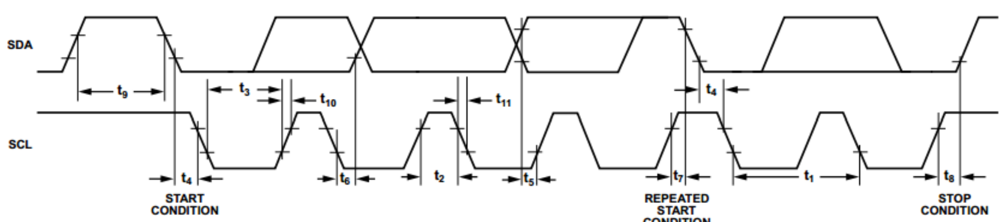
Pin Configuration (Top View)

. Typical Current Consumption vs. Data Rate
(T_A = 25°C, V_S = 2.6 V, $V_{DD I/O}$ = 1.8 V)

Output Data Rate (Hz)	Bandwidth (Hz)	Rate Code	I_{DD} (µA)
3200	1600	1111	140
1600	800	1110	90
800	400	1101	140
400	200	1100	140
200	100	1011	140
100	50	1010	140
50	25	1001	90
25	12.5	1000	55
12.5	6.25	0111	40
6.25	3.13	0110	31
3.13	1.56	0101	27
1.56	0.78	0100	23
0.78	0.39	0011	23
0.39	0.20	0010	23
0.20	0.10	0001	23
0.10	0.05	0000	23

Pin Function Descriptions

Pin No.	Mnemonic	Description
1	$V_{DD I/O}$	Digital Interface Supply Voltage.
2	NC	Not Internally Connected.
3	NC	Not Internally Connected.
4	SCL/SCLK	Serial Communications Clock.
5	NC	Not Internally Connected.
6	SDA/SDI/SDIO	Serial Data (I²C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).
7	SDO/ALT ADDRESS	Serial Data Output (SPI 4-Wire)/Alternate I²C Address Select (I²C).
8	$\overline{CS}$	Chip Select.
9	INT2	Interrupt 2 Output.
10	NC	Not Internally Connected.
11	INT1	Interrupt 1 Output.
12	GND	Must be connected to ground.
13	GND	Must be connected to ground.
14	V_S	Supply Voltage.
15	RESERVED	Reserved. This pin must be connected to V_S.
16	GND	Must be connected to ground.

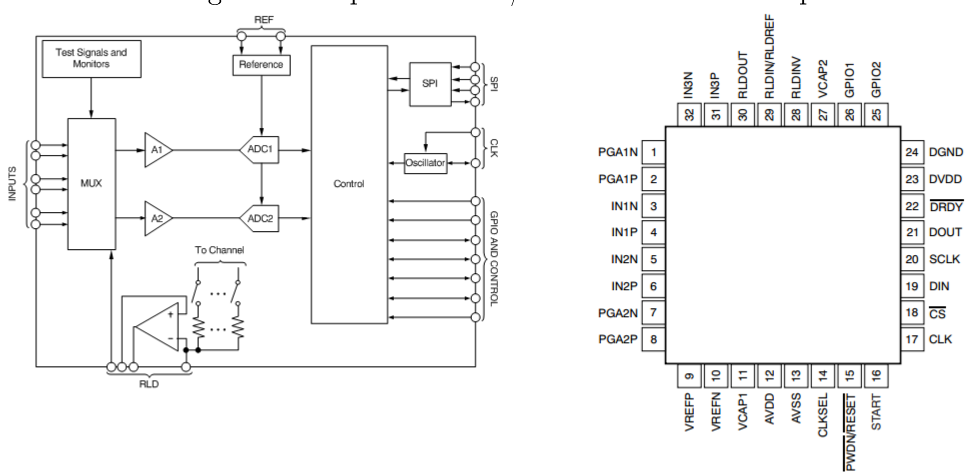


I²C Timing Diagram

2.1.5.3 ADS1191 IBIS Biopotential Measurement Chip The ECG signal from the body will be measured using two conductive pads on the device which will contact the skin and have a typical voltage difference of up to 400μV and a frequency range of .01 to 250 Hz. This signal will then be amplified and filtered, and converted to digital information to be transmitted through the microcontroller and to the data display. There are three important factors to consider in the ECG design: patient safety, signal amplification, and signal filtering.

Because the sensor pads are designed to detect minute bio-voltage potentials, we will need to use an operational amplifier which will bring up our power consumption a lot. In addition, we will be needing to use a method which will allow the signals to be digitally stored in the PaLFI. For this reason we use the ADS119x IBIS Model Low-Power, 2-Channel, 16-Bit Analog Front-End for Biopotential Measurement chip. The internal workings of this chip are displayed below.

Figure 9: Amplifier and A/D converter for ECG pads



The device has a programmable gain which allows the analog ECG signals to be amplified and then converted to a digital signal.

A significant difference between the ECG device and the previous sensors is that it does not use the IC2 configuration for communication, but rather an older SPI single BUS interface. Luckily, the P3.1 port of the microcontroller is equipped to deal with this very type of interface and is used in this configuration to serve its purpose.

2.1.6 Microcontroller firmware and implementation

Below is the flow chart for the microcontroller implementation. The microcontroller has essentially two routines that it runs through, the "Push interrupt" and the "busy" interrupt. A press on the PUSH button activates the PaLFI, reads the PCU state and "page 2" (a memory location in the PaLFI). If page 2 is 0x00, the autotrim routine is executed to trim the resonance circuit of the PaLFI to 134.2 kHz. If page 2 is 0x01, nothing is done, and the microcontroller returns to standby. In case of a BUSY interrupt, an MSP Access Command is executed. In this case, the 6 bytes transmitted from the reader to TMS37157 are read via SPI command. This is followed by all of the sensor measurements. A more detailed description of these operations are seen in figure 11 .

Figure 10: Microcontroller flow chart

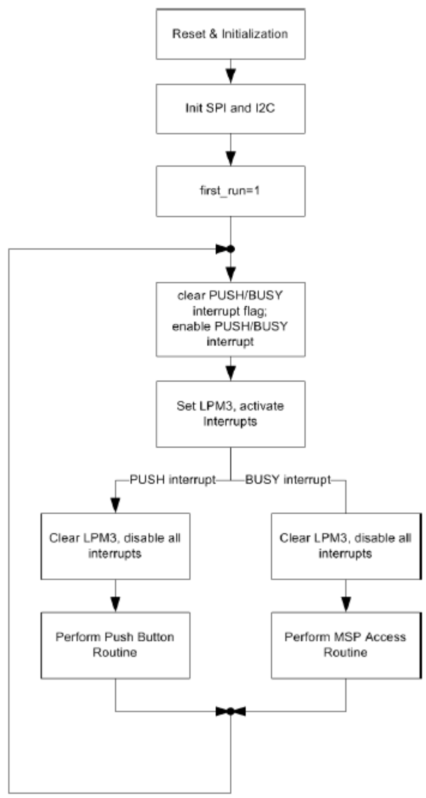
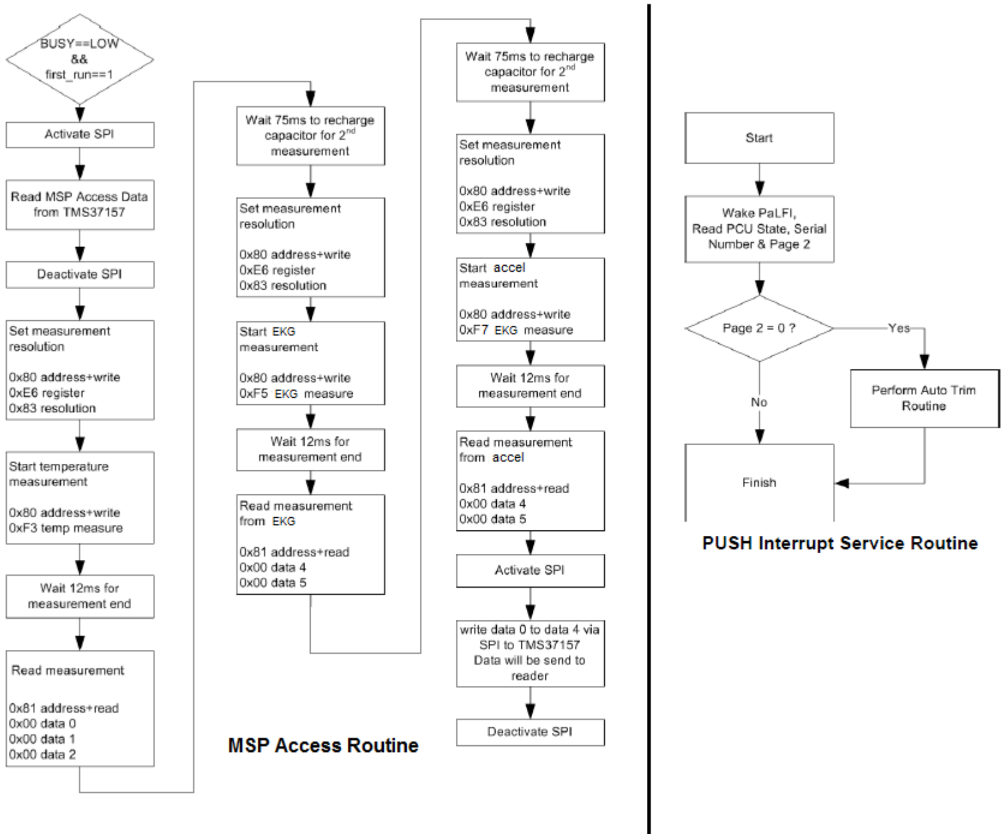


Figure 11: Microcontroller flow chart



2.2 Transceiver

The transceiver will be composed of the Base Station PaLFI reader connected to a computer and powered via a USB interface. There is an option available to have a higher power PaLFI reader (RI-RFM-007B) with a separate power supply, which we will implement if the device requires a greater range after the design has been proven to work. For now, the transceiver and antenna that is part of the development kit will be sufficient. Also, it is important to understand that the transceiver is so named due to its transmission / receiving of power and data which is powered by an external source, and the transponder is named due to its lack of an external power source.

2.2.1 Transmission of Power and Data

The electromagnetic link between the Reader and the sensor system is maintained through a 134.2kHz Frequency Modulated, digital data encoding scheme known as NRZ (Non Return to Zero). Figure 12 shows the basic communication components and Figure 13 shows the NRZ basic principle of operation. The typical data low bit frequency is 134.7 kHz, and the typical high bit frequency is 123.7 kHz. The low and high bits have different durations due to one bit being equivalent to 16 RF cycles of the transmitting frequency. The details of the data transmission are not immediately relevant to this project. However, one important command that must be discussed is the "Battery Charge" command. This command will simply send an RF pulse of length decided in the software of the transmitter in order to charge the capacitor on the sensing device. This command is only ended once the next command is transmitted to the sensing device. This is the length of time that will need to vary in order to charge the sensing device to an adequate power.

Figure 12: RF Link Diagram

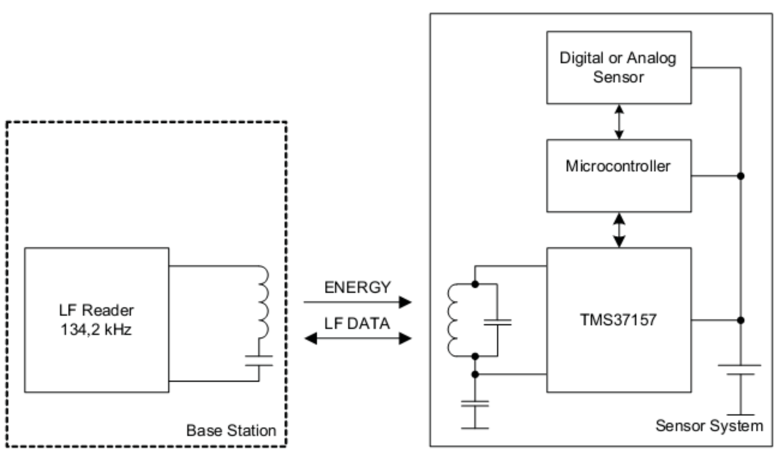
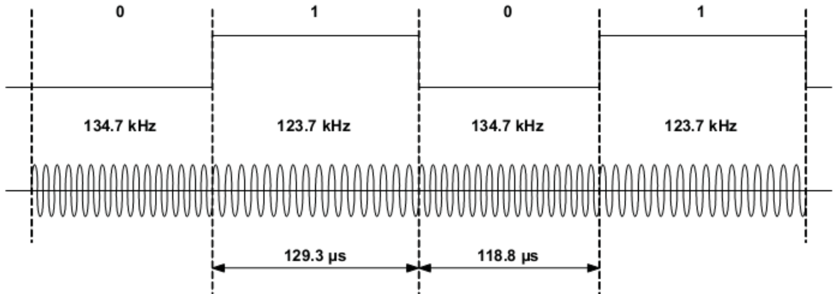


Figure 13: NRZ Encoding Scheme



2.2.2 Data Flow Diagram

The software used in the data transceiver is implemented using the RFID development software included in the TI's eZ430-TMS37157 Development Tool. The source code written to perform all the necessary functions will be written in C and programmed into the microcontroller while it is powered by a development battery over the air. Once the firmware is loaded, then the device can operate without a battery. The software contains many built in functions for I2C and SPI communications between the sensors. Shown below in Figure 14 is an example of the code that would be used to read a value into the microcontroller from the temperature sensor. The other code snippets are fairly similar, with different register addressing.

Figure 14: Temperature Sensor communication code

```
i2cStart();
i2cWrite(0x80);   // 0b10000000  sensor addres + write attempt
i2cWrite(0xF3);   // trigger temp measure; no hold master
Delay_ms_LPM(12); // wait for measurment end
i2cStart();
i2cWrite(0x81);   // 0b10000000  sensor addres + read attempt
ucMSPAccessData[0] = i2cRead(1);   // get meas value MSByte
ucMSPAccessData[1] = i2cRead(1);   // get meas value LSByte
i2cRead(1);       // get CRC8 value
i2cStop();
```

2.2.3 Code Process

As stated in the design overview, the code flow of data measurements will have several periods of charging the capacitor interspersed between the measurement, downlink, and uplink phases. The exact timing of these phases is still unknown, as it depends on tests of distance, power consumption, and many different modes for which the power consumption and charge time is dependent on testing of the components once they are connected to each other. However, the basic sequence will be the same:

- Charge phase: Generate an RF field of 134.2 kHz from the reader to the wireless sensor module to charge the power capacitor.
- Downlink phase: Send command to wireless sensor to start measurement.
- Measurement and recharge phase: Trigger three measurements from microcontroller; recharge the power capacitor on the sensor device.
- Uplink phase: Send measurement results via RF interface (134.2 kHz) back to reader

The development program has data plotting and storing functions integrated, and will display the measurements in a graph format.

3 Requirements and Verification

3.1 Data Sensing and Manipulation

3.1.1 Sensors

Table 2: Temperature Sensor

Requirement	Verification
1) When the microcontroller gives the "power up" command to the temperature sensor, the steady-state current rises from .1 μ A to 200 μA (requires working microcontroller)	Using a multimeter, measure the current at the V_{DC} of the sensor. Before the "power on" command is given 0.1 μA should be read. After the power on command is given, $200 μA should be read.
2) When the "Start condition" (shown in figure 7) for the temperature sensor is given across the I2C BUS, the sensor actively begins to send measurements to the Microcontroller (requires working microcontroller)	Attach a digital waveform oscilloscope to the SCL and SDA pins of the device. Observe the "Start Condition" being given by the microcontroller. Verify measurements are being taken by observing the data across the BUS. When heat is applied to the temperature sensor, observe the binary value of the temperature sensor rise.
3) When the "Stop Condition" is given (shown in figure 7) is given across the I2C BUS, the temperature sensor stops transmitting data across the BUS (requires working microcontroller)	Ensure when the "Stop Condition" is given by the microcontroller that no more values are put on the BUS by the sensor.
4) When the "Power Down" command is given, the current lowers from 200μA to $.1 \mu$A	Using a multimeter, measure the current at the V_DC of the sensor. Before the "power down" command is given 200 μ A should be read. After the power on command is given, $.1 \mu$A should be read.

Table 3: EKG Sensor

Requirement	**Verification**
1) When the microcontroller gives the "power up" command to the ECG the steady-state current rises from 3.3μA to 200 μA (requires working microcontroller)	Using a multimeter, measure the current at the V_DC of the sensor. Before the "power on" command is given 3.3 μA should be read. After the power on command is given, $200 μA should be read.
2) When the "Start condition" for the ECG sensor is given across the I2C BUS, the sensor actively begins to send measurements to the Microcontroller (requires working microcontroller)	Attach a digital waveform oscilloscope to the SCL and SDA pins of the device. Observe the "Start Condition" being given by the microcontroller.
3) When the "Stop Condition" is given is given across the I2C BUS, the ECG sensor stops transmitting data across the BUS (requires working microcontroller)	Ensure when the "Stop Condition" is given by the microcontroller that no more values are put on the BUS by the sensor.
4) When the "Power Down" command is given, the current lowers from 200 μA to .3μA	Using a multimeter, measure the current at the V_{DC} of the accelerometer. Before the "power down" command is given 200 μA should be read. After the power on command is given, .3μA should be read.

Table 4: Accelerometer

Requirement	**Verification**
1) When the microcontroller gives the "power up" command to the Accelerometer, the steady-state current rises from .2μA to 23μA (requires working microcontroller)	Using a multimeter, measure the current at the V_{DC} of the sensor. Before the "power on" command is given .2μA should be read. After the power on command is given, 23μA should be read.
2) When the "Start condition" (shown in figure 8) for the Accelerometer sensor is given across the I2C BUS, the sensor actively begins to send measurements to the Microcontroller (requires working microcontroller)	Attach a digital waveform oscilloscope to the SCL and SDA pins of the device. Observe the "Start Condition" being given by the microcontroller. Verify measurements are being taken by observing the data across the BUS. When the accelerometer is moved around, observe a change in the data being transmitted across the BUS.
3) When the "Stop Condition" is given (shown in figure 8) is given across the I2C BUS, the temperature sensor stops transmitting data across the BUS (requires working microcontroller)	Ensure when the "Stop Condition" is given by the microcontroller that no more values are put on the BUS by the sensor.
4) When the "Power Down" command is given, the current lowers from 23μA to .2μA	Using a multimeter, measure the current at the V_DC of the accelerometer. Before the "power down" command is given 23μA should be read. After the power on command is given, .2μA should be read.

3.1.2 Microcontroller

Table 5: Microcontroller

Requirement	Verification
1. Ensure the microcontroller "powers on" when the PaLFI receives a "PUSH" command	Simulate transmitting the "PUSH" command to the PaLFI. Then use the development tool "test" command to ensure the microcontroller is powered on. A "test successful" prompt will display if the microcontroller is successfully powered on.
2. Ensure the microcontroller can successfully perform the MSP access routine	Use the development tool to instruct the microprocessor to issue the "MSP Access" command. Write some value to its memory, then read that value back at the transceiver using the "Raw Data" Tab of the development software.

3.2 TPS71433 (DC Power Supply System)

Table 6: Power Supply System

Requirement	**Verification**
1) The Capacitor CBAT must provide enough power to all systems when it is needed.	The device recieves all the data requested and it is within a margin of error of 10%
a. The microcontroller must receive a voltage from the capacitor greater than 1.8V over a time greater than 4ms.	a. Using the development board, send a 75ms power pulse to the transciever to charge the capacitor, and connect an oscilloscope to the capacitor leads. Measure the time it takes for the voltage to decrease to below 1.8V.
b. The Temperature Sensor must receive a voltage from the capacitor greater than 2.7V over a time greater than 40ms	b. Using the development board connected to the temperature sensor via the I2C data pins of the microcontroller, send a 75ms power pulse to the transciever to charge the capacitor, and connect an oscilloscope to the capacitor leads. Send a 'Start measure' command to the temperature sensor via the microcontroller, and measure the time it takes for the capacitor to reach a voltage of 2.7V.
c. The accelerometer receives a voltage from the capacitor greater than 1.7V for a time greater than 40ms	c. Using the development board connected to the accelerometer via the I2C data pins of the microcontroller, send a 75ms power pulse to the transciever to charge the capacitor, and connect an oscilloscope to the capacitor leads. Send a 'Start Measure' command to the accelerometer via the microcontroller, and measure the time it takes for the capacitor to reach a voltage of 1.7V.
d. The ECG measurement circuit receives a voltage from the capacitor greater than 2.7V for a time greater than 20ms.	d. Using the development board connected to the EKG sensor via the I2C data pins of the microcontroller, send a 75ms power pulse to the transciever to charge the capacitor, and connect an oscilloscope to the capacitor leads. Send a 'Start Measure' command to the EKG sensor via the microcontroller, and measure the time it takes for the capacitor to reach a voltage of 2.7V.
e. Each sensing component is getting a signal from the microcontroller and able to power on with a charge time that is less than 100ms.	e. Send a 100ms pulse of "battery charge" command to the PaLFI transponder. Power on, measure, and store data from each of the measurement devices separately from the others, and be sure that the data received is coherent and within a margin of error of 10%

Table 7: DC/DC Converter

Requirement	**Verification**
When a simulated "PUSH" function is applied to the RF Chip, the Power supplying capacitor charges to a value between 2.1-2.9V (min-max range for single transmit operation)	A voltmeter will be used to measure the potential across the capacitor to ensure it is in the desired voltage range. If the PaLFI and microcontroller are operational, this can also be done by sending the device a "battery check" ping. If the "battery check" is successful, the receiver will display an "operation success" prompt.

3.3 PaLFI System

Table 8: PaLFI System

Requirement	Verification
When PUSH signal is transmitted, PaLFI activates and powers device on	Using a Multimeter/oscilloscope, monitor the PUSH pin on the PaLFI. When the PUSH signal is transmitted, the wire connected to the PUSH pin becomes a digital "high" (binary 1).
When a simulated "MSP access" instruction is given, a cycle of data measurement is transmitted	Use the development tool to pre-load data into the PaLFI memory. Simulate a "MSP access" command on the SPI pins of the PaLFI. Observe the output at the receiver and ensure it matches the simulated data stored in memory.
When the "Power down" instruction is given, the device is powered down	Simulate a "Power Down" command on the SPI pins of the PaLFI. Measure the capacitor to ensure it is completely discharged. Then measure the voltage across the PALFI to ensure it is not consuming power.
When the "Power down" instruction is given, the device is powered down into standby mode	Check the DC amperage going to the Vcc pin of the microcontroller and the PaLFI using a multimeter across the jumper of the development kit.

3.4 Tolerance Analysis

The main constraint on the functionality and effectiveness of this device is the power available to the components from RFID energy transmission and storage in a capacitor. The total power consumed by each of the devices according to their datasheets is shown in the table below.

Table 9: Power Consumption

Device	Standby Power (μW)	Active Power (μW)
PaLFI	0	300
Microcontroller	1.26	486
Temperature Sensor	0.27	540
Accelerometer	0.52	104
ECG Sensor	10	600
DC/DC Converter	0 (only powered on when in range)	3.2

4 Ethical Considerations

The IEEE code of ethics in Section 7.8 of IEEE Policies contains the following items which are relevant to this project:

- *to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;*
  This device is intended to monitor the health and welfare of human beings, therefore it must be relied upon to provide correct results when used correctly, and not cause any harm to the user while in use. Any potential danger in the device's normal operations will be documented and disclosed in the final report.

- *to be honest and realistic in stating claims or estimates based on available data;*
  Each of the components has the relevant data and claims that this project expects in the design section of this report. The stated claims have been compared to other available data to be sure that they are realistic.

- *to improve the understanding of technology; its appropriate application, and potential consequences;*
  This project's purpose is to further the knowledge and innovation in the newly expanding field of BAN's and health informatics. Any potential consequences of this knowledge will be duly addressed in the final report.

- *to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;*
  The team will only attempt technological tasks for others until after being qualified in training and disclosure of any and all risks involved.

- *to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;*

All contributions and sources have been cited in the appropriate References section of this report.

5 Safety Considerations

The testing and assembly of components will adhere to the Lab Safety Rules. Since this device will have electric current near a human subject, it will comply to the "Safe Current Limits for Electromedical Apparatus," which it indeed does satisfy.

6 Cost and Schedule

6.1 Cost Analysis

The total cost of this project consists of labor and parts costs which total **$13,301.31**, as detailed in the following two sections.

6.1.1 Labor

Table 10 shows the labor costs of this project.

Table 10: Labor Costs

Team member	Hourly rate	Number of hours invested	Labor Cost
Matt	$40	160	$6,400
Russ	$40	160	$6,400
		Total:	**$12,800**

6.1.2 Parts

Table 11 shows the parts cost of this project.

Table 11: Parts Costs

Part # and Description	**Cost**	**Quantity**	**Total Price**
TMS37157 (PaLFI)	$1.25	1	$1.25
MSP430F2274 microcontroller	$2.98	1	$2.98
TPS71433 DC / DC converter	$1.10	1	$1.10
eZ430-TMS37157 Development Tool	$199.99	1	$199.99
RI-RFM-007B High Power Reader Module	$289.04	1	$289.04
ADXL344 (Accelerometer)	$1.45	1	$1.45
ADS 119x (Biopotential Measuring Chip)	$2.14	1	$2.14
MCP9808 (Temperature Sensor)	$1.36	1	$1.36
Analog components	$2.00	x	$2.00
		Total:	**$501.31**

6.2 Timeline

Table 12 indicates the timeline for project completion, and the team member in charge of each specific task that will need to be completed.

Table 12: Schedule of tasks and team member in charge

Week	**Date**	**Tasks**	**Team Member**
1	2/3 - 2/6	Write Project Proposal	Matt
	2/6	Sign up for mock design review	Matt
	2/6	Attend Fabrication Seminar – MRL	Russ
2	All week	Test possible antennas in lab	Russ
	All week	Research RF harvesting	Matt
3	All week	Simulate RF Harvesting circuit	Russ
	All week	Research receiver implementations	Matt
4	All week	Prepare for Design Review	Matt
	2/27	Design Review - 9am	Russ & Matt
	All week	Simulate and revise design	Russ
5	3/4	Order parts	Matt
	All week	Begin writing software	Russ
6	3/11	Individual Progress Reports Due	Russ & Matt
	All week	Begin building the receiver parts and software	Matt
	All week	Begin preliminary tests of transmitter	Russ
7	All week	Analyze results of last week's test	Matt
	All week	Have full circuit built on Eagle board	Russ
8	3/25	Sign up for mock-up demos	Matt
	All week	Test receiver design against requirements	Matt
	All week	Test transmitter design against requirements	Russ
9	4/1	Mock up Presentation	Russ & Matt
	All week	Test full system with reciever and transmitter	Matt
	All week	Order extra parts, complete extra fabrication	Russ
10	4/8	Test final revision	Matt
	All week	Make necessary changes, order parts, etc	Russ
	All week	Simulate final revision	Matt
11	4/15	Test final revision	Russ
	4/15	Sign up and Prepare for Demo	Matt
	All week	Begin writing report	Russ
12	All week	Prepare for Demo	Matt
	4/22-25	Demos	Russ & Matt
13	4/29-30	Final Presentation	Russ & Matt

7 References

1. MSP430x22x2, MSP430x22x4 Mixed Signal Microcontroller data sheet

2. MSP430x2xx Family User's Guide

3. TMS37157 Passive Low-Frequency Interface (PaLFI) Device With EEPROM and 134.2-kHz Transponder data sheet 4. eZ430-TMS37157 Development Tool User's Guide

5. SPI Library for TMS37157

6. ADXL344 3-Axis, ±2 g/±4 g/±8 g/±16 g Ultralow Power Digital MEMS Accelerometer Data Sheet

7. ADS1294/ADS1294R/ADS1296/ADS1296R/ADS1298/ADS1298R Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements Data sheet

8. MCP9808 ±0.5° C Maximum Accuracy Digital Temperature Sensor

9. Sensirion Low-Power Battery-Less Wireless Temperature Sensor Example configuration review sheet