HIGH SPEED WIRELESS DATA ACQUISITION SYSTEM

By

Muhammad Nadeem

B.E. - Electrical Engineering,

N.E.D. University of Engineering & Technology,

Karachi/Pakistan 1992

A project

Presented to Ryerson University

In partial fulfillment of the

Requirement for the degree of

Master of Engineering

In the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2014

Declaration

I hereby declare that I am the sole author of this project report. This is a true copy of the report including any required final revisions.

I authorize Ryerson University to lend this report to other institutions or individuals for the purpose of scholarly research

I further authorize Ryerson University to reproduce this report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my report may be made electronically available to public.

Abstract

High speed wireless data acquisition system

Muhammad Nadeem, Master of Engineering, Electrical Engineering

Ryerson University, Toronto 2014

In a setup for a system long term stability and reliability test, sensors are used to measure physical quantities, affecting the behavior of the system, by sampling the sensor readings, convert it to digital numerical value and saving it for further detail analysis. The sensors are wired to a central location to collect and log data, due to extensive wiring requirements the setup is very difficult and sometimes even impossible to implement. This project presents an implementation of high speed wireless data acquisition system which samples sensors output at high speed (5 KHz), converts it to digital numerical form and sends wirelessly to central data gathering unit thus avoiding home run wiring from each sensor to central data gathering unit where it is logged on USB flash drive and send to PC for real time display. The implementation target was wireless link between a transmitter module, serving a maximum of 8 sensors at 5 KHz sampling rate and 16bit ADC resolution for each sensor, to data gathering unit. The implementation does fall short on specification on number of channels and sampling rate due to limitation of over the air data rate of the radio module, what we were able to achieve is 4 channels of 16 bit ADC resolution at 2 KHz sampling rate using radio module with 300 Kbps over the air data rate. Using different sensors and with different configurable settings the tests shows that the stored data at the data gathering unit and the data stored using wired data acquisition system has no difference. For future improvement radio module with over the air data rate (1.55 Mbps) which allows multiple transmitters connected wirelessly to a single data gathering unit providing more flexibility in sensor deployment. Even though the implementation falls short on some of the features but with using improved radio module and/or using some compression techniques on ADC data, before sending data wirelessly, these short comings could be overcome easily

Acknowledgement

I would like to thank my supervisor Dr. Krishna Dev. Kumar for his valuable time and dedication and thanks to my wife and kids for their patience and thanks to my friends for their encouragement.

Table of contents

Table of Figures

List of Tables

1 Introduction

A system under test requires a multitude of sensors to measure different physical quantities of the system. A system under test may require lots of physical quantities to be measured and requires a number of sensor to be deployed and here comes the real issue, wiring each sensor to the central data gathering unit would be cumbersome and sometimes not even possible therefore a data acquisition system that can read the sensors analog value and convert into digital form with required precision and send it wirelessly to a central data gathering unit where the data can be displayed on PC and saved on readily available USB flash drive for further processing is very appealing. Also sometimes it is required to test the system over time may be days or weeks to make sure the system is reliable and functionality is not deteriorated overtime, depending on the size of the flash drive weeks of logged data can be stored on multiple files for future processing, for application like this wireless data acquisition system is very convenient to use and deploy. The main challenge is to keep the transmitter and receiver's RF connection in sync and error free so that the data stored is termed reliable for further processing

Some of the present implementations fall short in one feature or the other. Sean Casey [1] implementation uses 12 bit analog to digital converter (ADC) resolution at 200 Hz sampling rate while Bhesania [2] uses 5 channels of 8 bit ADC resolution. Hua Fang et al. [3] uses low speed GPRS cell phone communication for their wireless link which is slow and using cell phone is not within the scope of this project. Chen Z et al. [4] uses a radio with over the air data rate of 250kb/s, ADC with 4 channels and resolution of 16 bit which is very close to the specification but still does not satisfies all the specifications. JinLin Hu et al. [5] uses Bluetooth as wireless communication medium which suffer from limited communication range for ADC they used 12 bit ADC converter which falls short on specification since the required specification calls for 16 bit ADC resolution. Even though this implementation did fall short in some features as well but by implementing future recommendations these shortcomings can be overcome very easily

1.1 System architecture

High speed data acquisition architecture consists of three main modules (see figure 1. System block diagram)

- Signal conditioning amplifier
- Transmitter module
- Receiver module

Signal conditioning amplifier (SCA) and transmitter module together are considered one unit and located close to the sensors whereas the sensors are wired to the SCA using 4-20mA current loop. The SCA is a

plug-in module to the transmitter module which has the capacity to accommodate 8 SCAs thus providing 8 conditioned analog signals to be processed by transmitter. Currently only 4 SCAs are used providing 4 channels for data acquisition only due to limitations on radio module which will be evident in radio communication section.

Main transmitter unit, which is microcontroller controlled, gathers conditioned analog signals from the SCAs and convert them to 16 bit digital numerical value at a maximum sampling rate of 2 KHz. These samples are buffered as they arrive and are sent periodically to the receiver module wirelessly.

The module which is remotely located from the transmitter module is the receiver module and is also microcontroller controlled. The function of the receives module is to receive and collect the ADC data acquired from the transmitter and log it to USB flash drive and at the same time sends it to PC for real time data display to visually make sure that the correct data is being acquired.

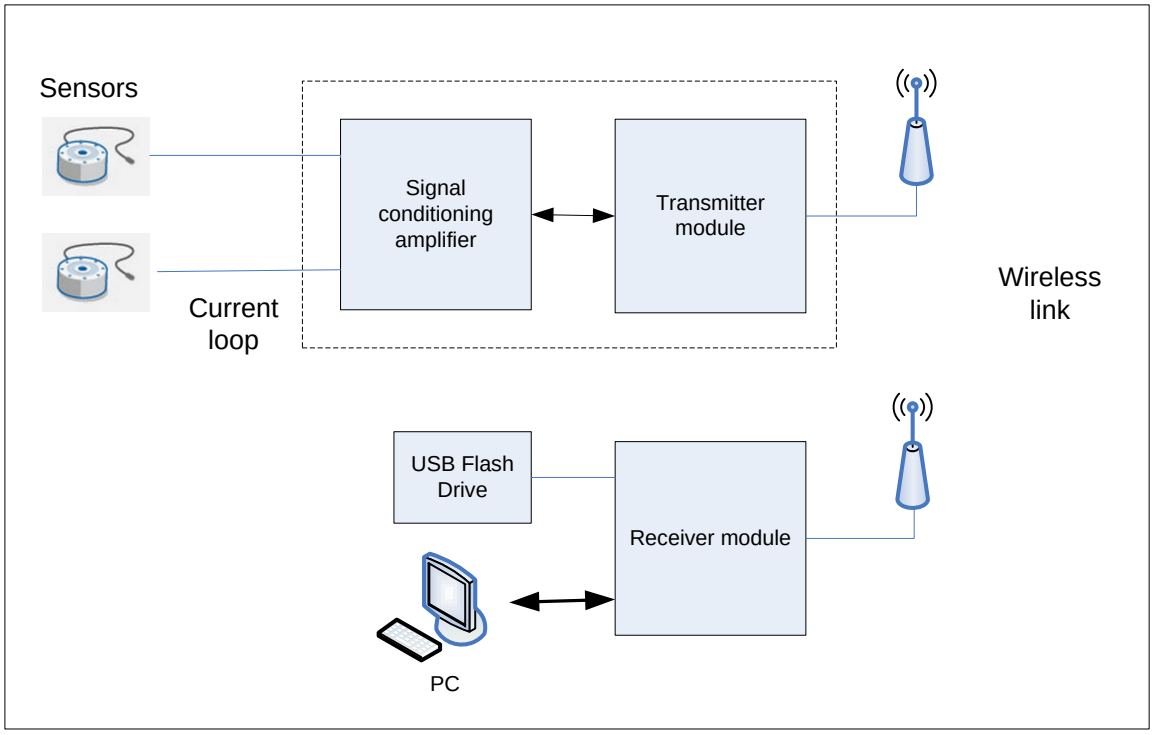

Figure 1. System Block Diagram

1.2 Analog to Digital conversion (ADC)

The process of measuring physical quantities involves a sensor that measures the physical quantity and produce appropriate analog signal depending on the variation of the physical quantity being measured. In

order to make this measurement, the sensor needs an electrical interface which can provide power to the sensor and at the same gets a varying electrical quantity based on sensor measurements. This interface is provided in the form of 4-20mA current loop which provide power and gets gives 4mA at its lowest level and 20mA at its highest level or 1V to 5V if measured across 250Ohms resistor. After getting the varying voltage from 1V to 5V the next stage is to convert this voltage to digital form using analog to digital converter chip. In the next section all the components described are discussed in detail in the same sequence as they are in actual design

1.2.1 Sensors

Sensors are used to measure physical quantities, such as temperature, pressure etc. into measurable electrical signal. Here is an example of few sensors that are commonly used in industry related to security, monitoring, consumer electronics and testing fields. (See Table 1 for different type of sensors and the physical quantity they measure). Sensors which are used and tested in this project are load, vibration and temperature sensors.

Table 1. Sensor types and measurements

Sensors	**Measurements**
Thermocouple, RTD, Thermistor	Temperature
Photo Sensor	Light, smoke
Microphone	Sound, ultra sound
Strain Gauge, Piezoelectric transducer	Force and pressure
Potentiometer, LVDT, Optical Encoder	Position and displacement
Accelerometer	Acceleration
pH Electrode	PH

1.2.2 4-20mA current loop

4-20mA current loop is the industry standard used for interfacing sensor, the interface provide power and get proportional analog signal to physical quantity measurement (see Figure 2 – 4-20mA interface)

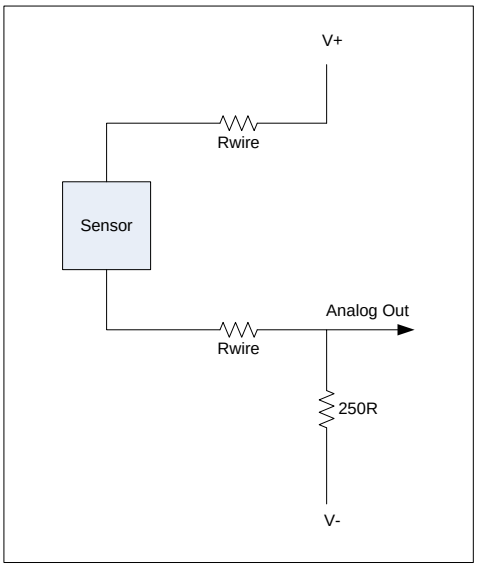

Figure 2. 4-20mA Interface

The advantage of using 4-20mA interface is that it's very robust and the fact that current not the voltage is the quantity is varying in proportion to physical quantity being measured and is not affected by the wire resistance from sensor to the A/D converter as long as the power supply can provide that current without dropping its voltage. Therefore no extra special care is required for optimizing wire length or wire resistance from sensor to the A/D converter

1.2.3 ADC converter

Following are the specifications for A/D conversion as required by the application.

- A/D channels for interfacing sensors
- 5 KHz sampling rate
- 16 bit A/D conversion resolution

The ADC used in transmitter module is a 16 bit analog to digital converter with maximum sampling rate of 100kSps. The ADC has the following features which makes it suitable for this project

- 100KSps maximum sampling rate
- Maximum non-linearity of +/-1.5LSB
- S/(N+D): 94DB Typ @ 45 KHz
- THD: -110 DB Typ @ 45 KHz
- Works on single supply of +5V
- Differential input range of +/-2.5V
- Parallel bus (16bit) and serial SPI interface

The report is organized as follows: In Radio Communication section the radio modules used in this project is discussed, their features, their interface to the main board, how they communicate and their mode of communication is discussed. In Hardware section complete system hardware design of all three modules in the system are presented along with the interface used by each modules and the overall structure of the hardware is described. The Firmware section details the architecture, code flow chart and system configuration of the firmware. In Testing section, test procedure and test results are discussed. In Future work section new and improved hard recommendations are described, the description of hardware and software modifications and firmware flow chart, for future work, are presented, finally Appendix shows some of the research used for selecting hardware and development tools for future work.

2 Radio Communication

Radio communication is a very critical and challenging part of this project. The communication should be reliable and consistent over time to insure reliability of the logged data since critical decisions are to be made based on this data. Therefore correction methods are used to make sure each packet from transmitter to receiver gets there without any error. Instead of creating a new radio module, readily available radio modules are used. There are two options available for radio modules the first 'radio-1' is a FSK transceiver while the other 'radio-2' is a DSS transceiver

2.1 Radio FSK (Radio-1)

Radio-1is a 900 MHz FSK transceiver, working at a maximum over the air data rate of 300 Kbps. The radio has a line of sight range of around 3000 feet and uses a maximum of 8 selectable channels (see Table-2 for available channels for selection). The radio module transceiver boasts a receive sensitivity of -113dBm and transmit signal strength of 12.5 dBm.

Table 2. Channel Selection for Radio -1

Channel Selection				
CHN_SEL2	**CHN_SEL1**	**CHN_SEL**	**CHANNEL**	**FREQUENCY**
0	0	0	3	903.37
0	0	1	15	906.37
0	1	0	21	907.87
0	1	1	27	909.37
1	0	0	39	912.37
1	0	1	51	915.37
1	1	0	69	919.87
1	1	1	75	921.37

Radio module uses RS232 and control signals to communicate with the host processor. The radio also provides control signals in the form of 'Ready' and 'RSSI' signals. 'Ready' signal indicates that the data is present in the receiver as received by the transmitter whereas 'RSSI' signal indicates the signal strength

of the data received is within the allowed signal strength level and the data received is actually data and not noise (see Figure-3 Radio interface for Radio-1)

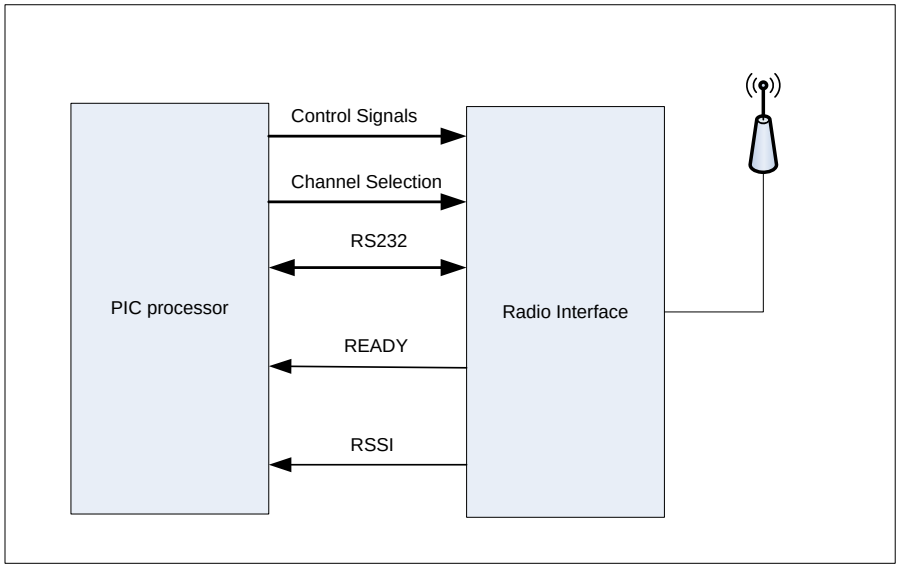

Figure 3. Radio Interface for Radio-1

Other control signals are also provided such as signals to select RS232 baud rate and setting direction of data transfer to the transceiver. Data packets are send to the radio module using RS232 interface, channel selection is used for selecting a quiet channel in otherwise RF busy environment, READY signal is used to indicate the module has received data from another module and in ready to be fetched, RSSI signal is used to indicate that the signal strength of receive data and whether or not the software should accept or reject the data based on minimum RSSI requirement

2.2 Radio DSS 1.55Mbps (Radio-2)

Radio-1 falls short on one important feature i.e. over the air data rate and as describe in subsequent section that this limitation has led to scaling down some features namely number of channels used and sampling rate. The most important benefit of using Radio DSS from AVLAN is its high data rate which allows not only to have communication at required specification of number of channels and sampling rate but also allows us to use a number of transmitter modules with one receiver modules as data gathering unit. Following are the main features and specification of the module

- 1.54 Mbps DSSS Radio with High Speed SPI and UART Data Interfaces
- FCC/IC Certified modular approval
- 128 Bit AES Encryption – FIPS 197 NIST Certified
- 148 dB link budget at 900 MHz enables exceptional range
- 4 Watts EIRP with 15 dBi antennae (+21 dB conducted)
- -112 dB receiver sensitivity with 15 dBi antennae (-98dB at port)
- TDMA MAC supports up to 63 concurrent subscribers
- Low power consumption: 0.75 Watts in transmit, 0.5 Watts in receive, 100 μW in standby
- Wide Temperature range: -40° to +85° C
- Narrow occupied bandwidth allows 12 channels at 900 MHz
- Small form factor allows easy integration: 47.5 x 51 x 7 mm, 10 grams
- Frequency Channels: 2.0833 MHz spacing and 1.75 MHz occupied bandwidth

Radio communicates at a maximum over the air rate of 1.55 Mb/s and allows subscriber transmitters talking to one master receiver. The MAC communication protocol is build inside the radio which provides proper signaling so that all the radio subscribers are synchronized with the master receiver.

To reduce data over head of sending packets the send data should be as large as possible to offset the extra bytes included within the frame structure (see Figure-4 radio communication between client and server). Maximum 4 subscriber transmitters can be used with one master receiver forming a data acquisition network

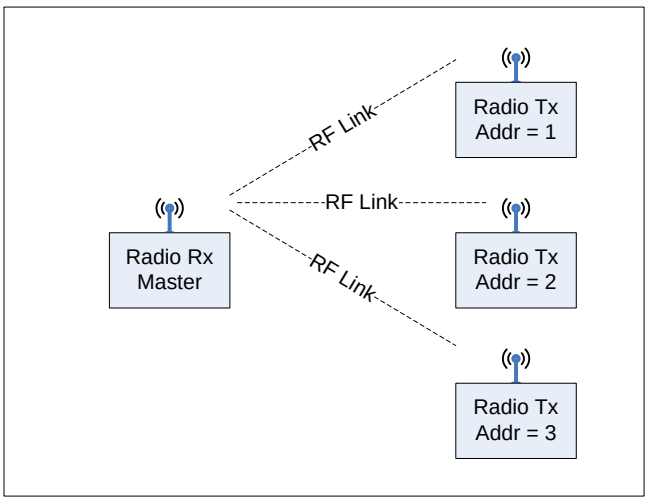

Figure 4. Radio communication between client (Radio Rx) and server (Radio Tx)

2.2.1 Radio Interface

Interface to the microcontroller uses SPI port to communicate with radio at a maxim bit rate of 7Mb/s, this bit rate is SPI data bit rate and not over the air date rate which is 1.55Mb/s (see Figure-5 Radio interface for Radio-2).

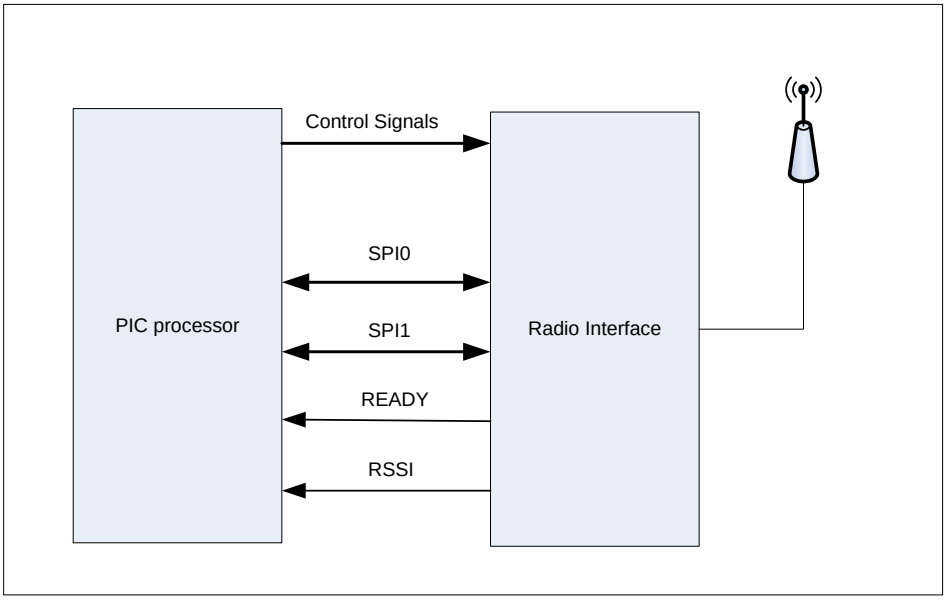

Figure 5. Radio Interface for Radio-2

2.2.2 Syncing Radio

Each radio module has unique address to get identified on the network. For synchronizing each radio module, associated with the transmitter, to the master radio module, associated with the receiver, the master radio send a sync character along with its address when the subscriber transmitter receives the sync character and its own address then it will transmit a packet of data, after receiving the data the receiver send the sync to another transmitter and receives its data and this process keeps on going until a stop signal is send to all the slave transmitters

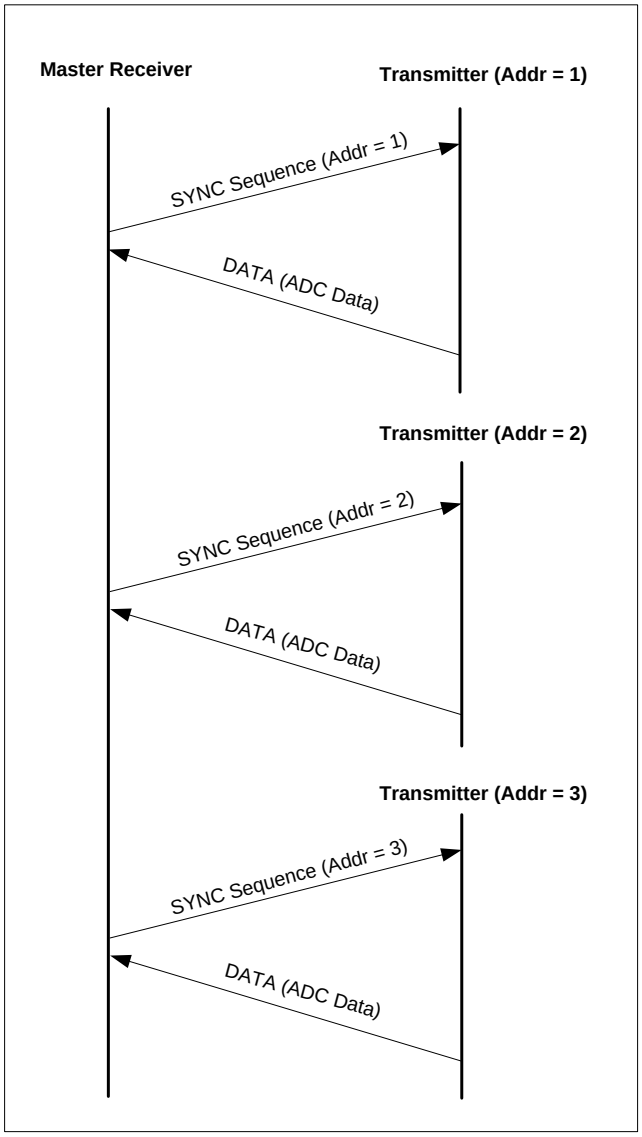

Figure 6. RF Sync between one master receiver and multiple transmitters

2.3 Over the air data rate requirement

To calculate what would be the minimum over the air data rate required in order to have consistent and reliable communication between the transmitter and receiver

Bit Rate = Number of channels x Sampling rate x number of bits per sample + 10% (overhead)

2.3.1 Radio-1

Bit rate = 4 x 5 KHz x 16 + 10% = 352 Kbps

No if we are using radio 1 then it's beyond its capabilities since the maximum over the air rate is 300kbps, therefore if we can reduce the sampling frequency to 3 KHz then let see what would be the required bit rate

Bit rate = 4 x 3 KHz x 16 + 10% = 212 Kbps

2.3.2 Radio-2

Bit rate = 4 x 5 KHz x 16 + 10% = 352 Kbps

This is well within the capability of radio-2. Since radio -2 is also capable of client server architecture with a single client talking to multiple servers so we can calculate how many servers can be used with one client

Number of servers = Maximum over the air rate / maximum over the air of one server

Number of servers = 1.55 Mbps/352 Kbps = 4

Therefore we can use 4 servers with one client transceiver

3 HARDWARE

Hardware consists of three main modules, signal conditioning amplifier, transmitter module and receiver module. Each hardware module is microcontroller controlled and uses PIC series microcontroller.

3.1 Transmitter

Transmitter consists of a main transmitter mother board and plugged in signal conditioning and amplifier boards. Following sections will describes the hardware architecture of signal conditioning amplifier and main transmitter board

3.1.1 Signal conditioning amplifier

Signal conditioning amplifier performs a multitude of tasks before signal is presented to A/D converter. SCA consist of a number of hardware signal conditioning blocks (see Figure-7 Signal conditioning and amplifier block diagram) to perform specific functions required by the module. Following is the description of each functional block

Depending on the type of sensor the power supply requirement could be different. To accommodate different power supply requirements for both positive and negative supply the signal conditioning amplifier provides a configurable selection for two different positive power supplies and two different negative power supplies. The total of power supply combinations for a given sensor is therefore 4.

The next block in the SCA is the differential amplifier. The differential amplifier is also configurable and can select differential input, single ended positive input or single ended negative input from each sensor output. Again the reason for this scheme to accommodate different type of sensor requirements even though the preferred scheme is differential which helps eliminate any common noise present on signal line

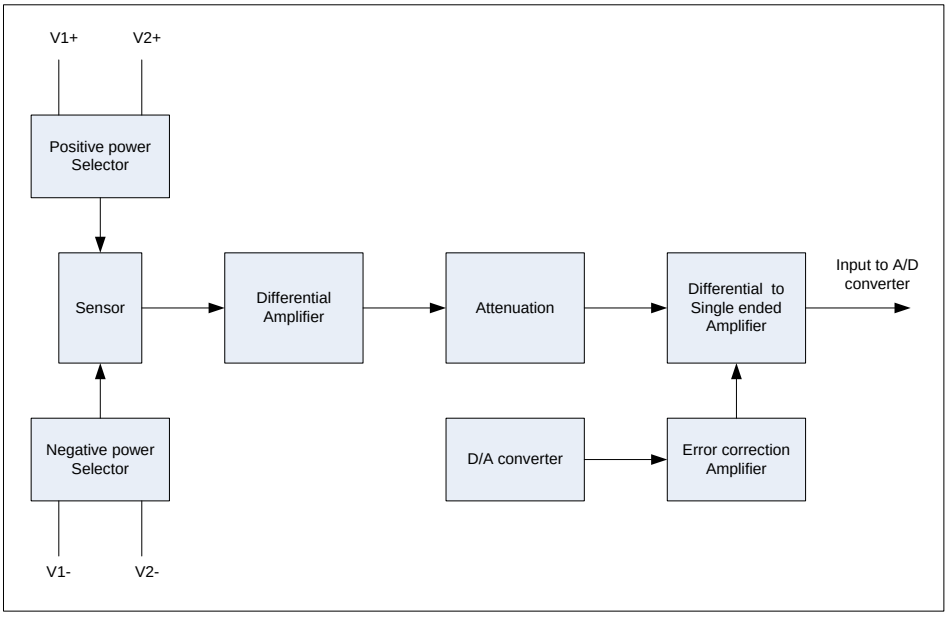

Figure 7. SAC block diagram

The next functional block is signal attenuation block. This block is also configurable and uses R-2R ladder as signal attenuator. The R-2R is controlled by microcontroller by 8 bit signal attenuation logic signal. The next block converts differential signal to single ended signal appropriate for ADC converter. The block also has an amplifier with output controlled by a D/A converter, the output of which is interfaced to the differential to single ended amplifier. The function of the D/A converter is to provide configurable shifted level of the signal to A/D converter.

3.1.2 Interface

Interface to the SCA to the microcontroller includes (See figure. 8 for interface to microcontroller) the following signals

- Select signal for power supply selection
- Signal to set attenuation
- Signal for D/A conversion for selecting signal output level

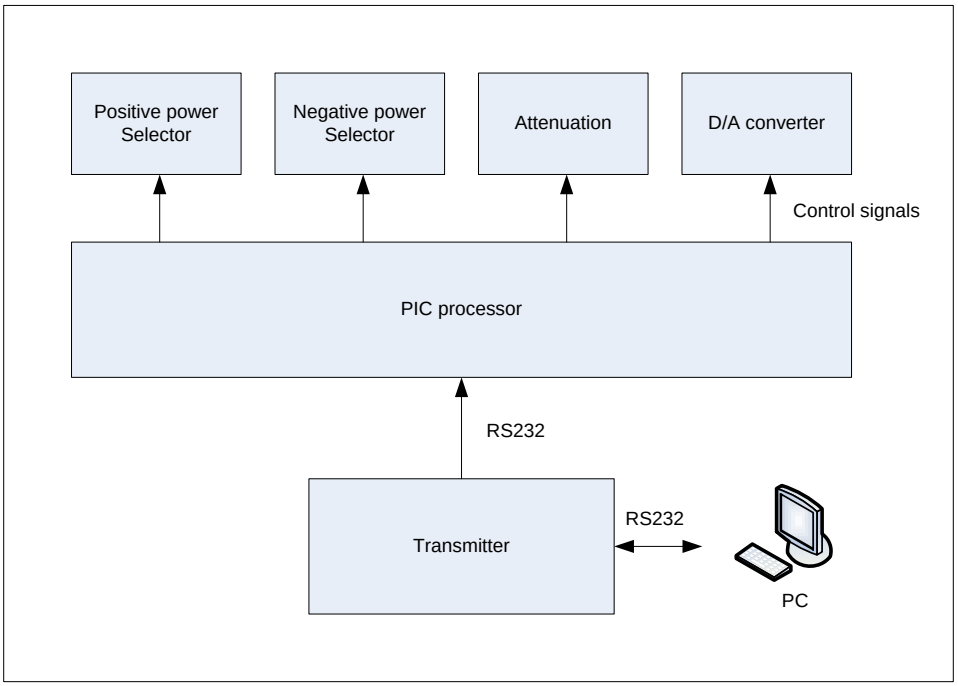

Figure 8. SAC Control

3.1.3 Transmitter main board

Transmitter main board is also microcontroller controlled and uses PIC series microcontroller. (See Figure-9 - Transmitter module block diagram). Transmitter main board consist of the following components

- 16 bit A/D converter using parallel bus interface
- D/A converter acting as error correcting amplifier uses SPI interface
- PC interface using RS-232
- Radio interface using RS232
- Signal conditioning and amplifier (AMP) interface using control signals and RS232

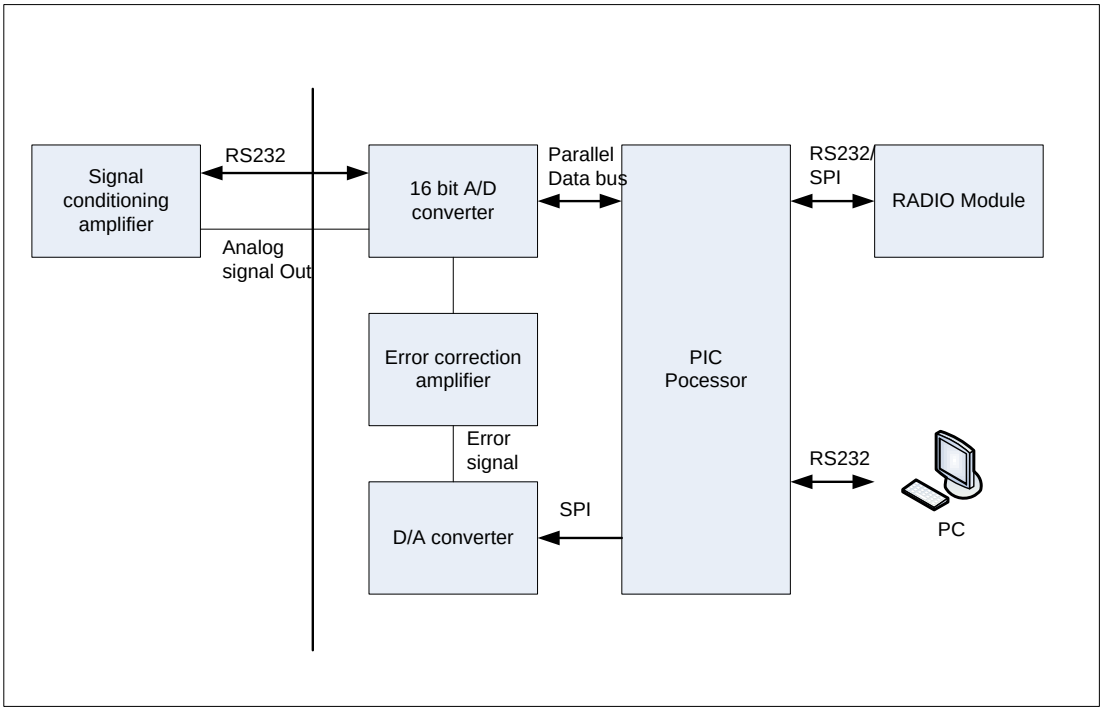

Figure 9. Transmitter module block diagram

3.1.3.1 16 bit A/D converter

16bit A/D converter is used for A/D conversion. Input to the ADC is differential signal to minimize noise and the digital interface is 8/16 bit multiplex data bus connected to the microcontroller

3.1.3.2 D/A converter

A single 14-/16 bit DAC is used for error correcting the A/D reading; it uses one direction SPI to interface with the microcontroller

3.1.3.3 PC interface

Uses standard RS232 interface at 38400 baud rate

3.1.3.4 Radio interface

Uses RS232 if radio module 1 is used and SPI if radio module 2 is used

3.2 Receiver

Receiver or data gathering unit is also microcontroller controlled and uses PIC series microcontroller. The receiver uses the following interfaces with the microcontroller. (See Figure 10 – Receiver module block diagram)

- Radio for receiving A/D reading from the transmitter
- USB host interface to connect USB flash drive
- PC interface
- Real time clock

3.2.1 Radio interface

Uses TTL level RS232 to interface radio to the microcontroller.

3.2.2 USB Host

The processor includes USB host peripheral to accommodate USB host devices for example in this USB flash drive. Standard USB flash drives which supports USB 2.0 standard with FAT32 file format can be used here to save data files

3.2.3 PC interface

Uses standard RS232 interface at 38400 baud rate

3.2.4 Real time clock

Real time clock uses I2C interface with the microcontroller. The clock is used to time stamped the A/D data frame before saving it to USB flash drive

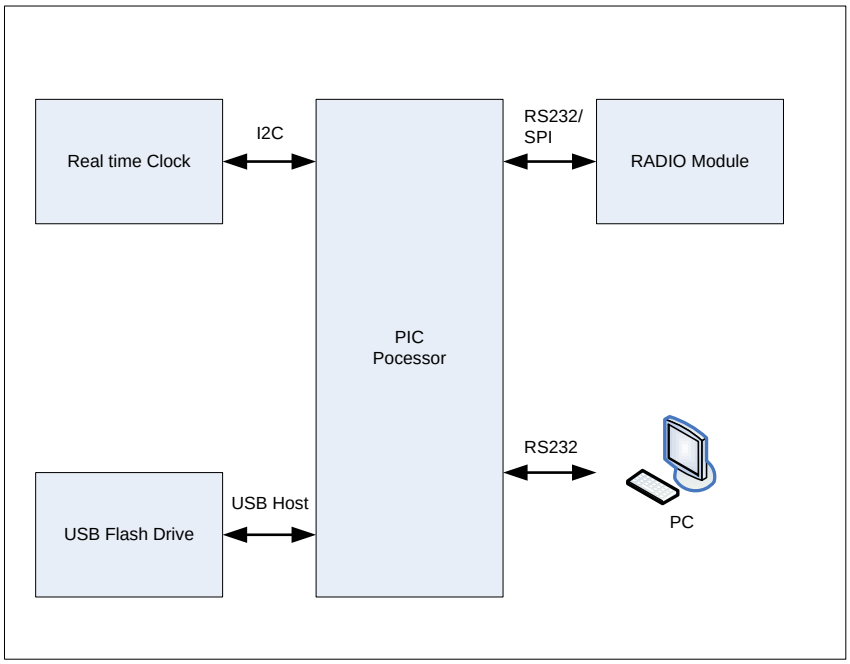

Figure 10. Receiver module block diagram

4 FIRMWARE

Firmware is mostly written in C language with the exception of startup code written in assembly language. Firmware architecture uses foreground/background with foreground as main application code while the background process involves interrupt service routines. Firmware code flow will be presented for SCA, transmitter and receiver in the next section

4.1 Transmitter

Transmitter firmware consists of foreground and background processes. Foreground process is initialization and main loop which runs forever while the background process comprises of interrupt service routines. Figure 11 shows the code flow chart for the foreground process while Figure 12 shows the code flow chart for background process.

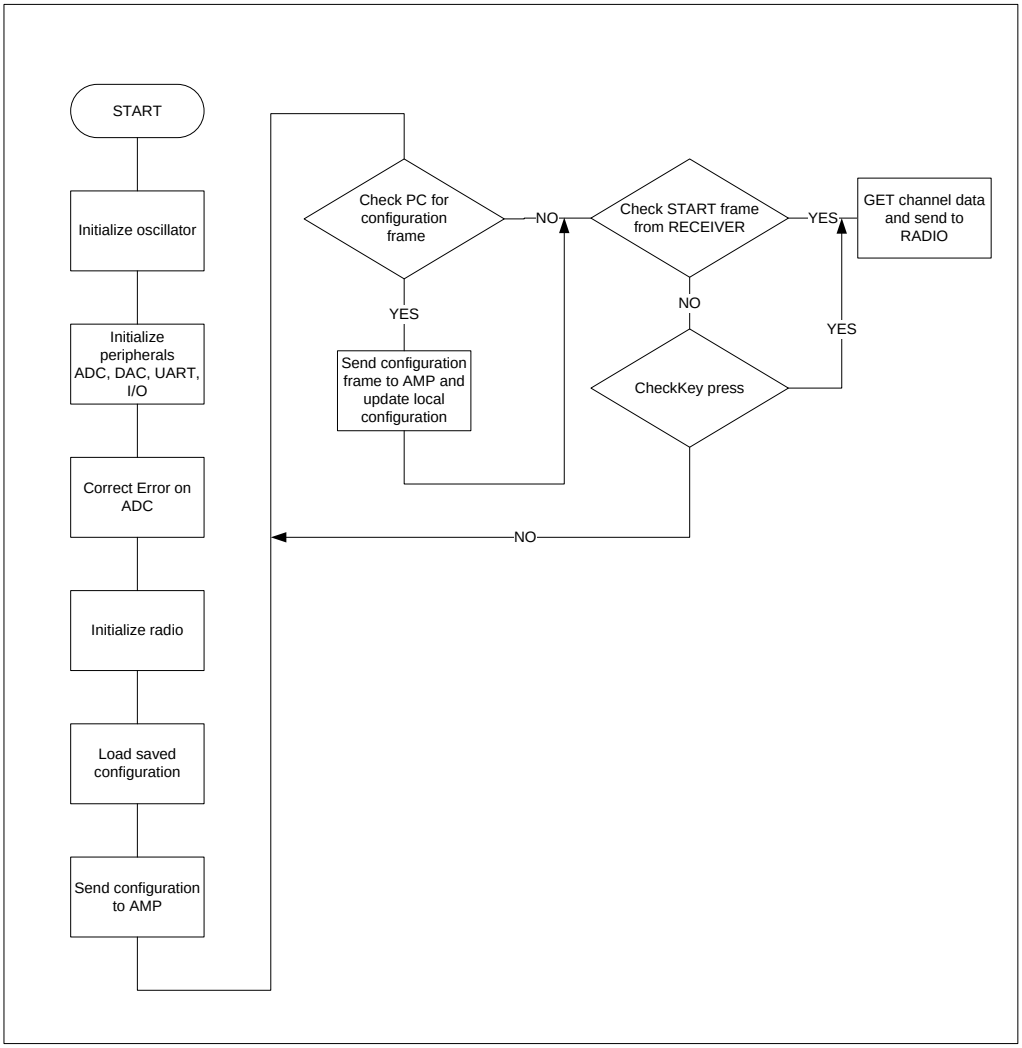

Figure 11. Transmitter module main loop flow chart

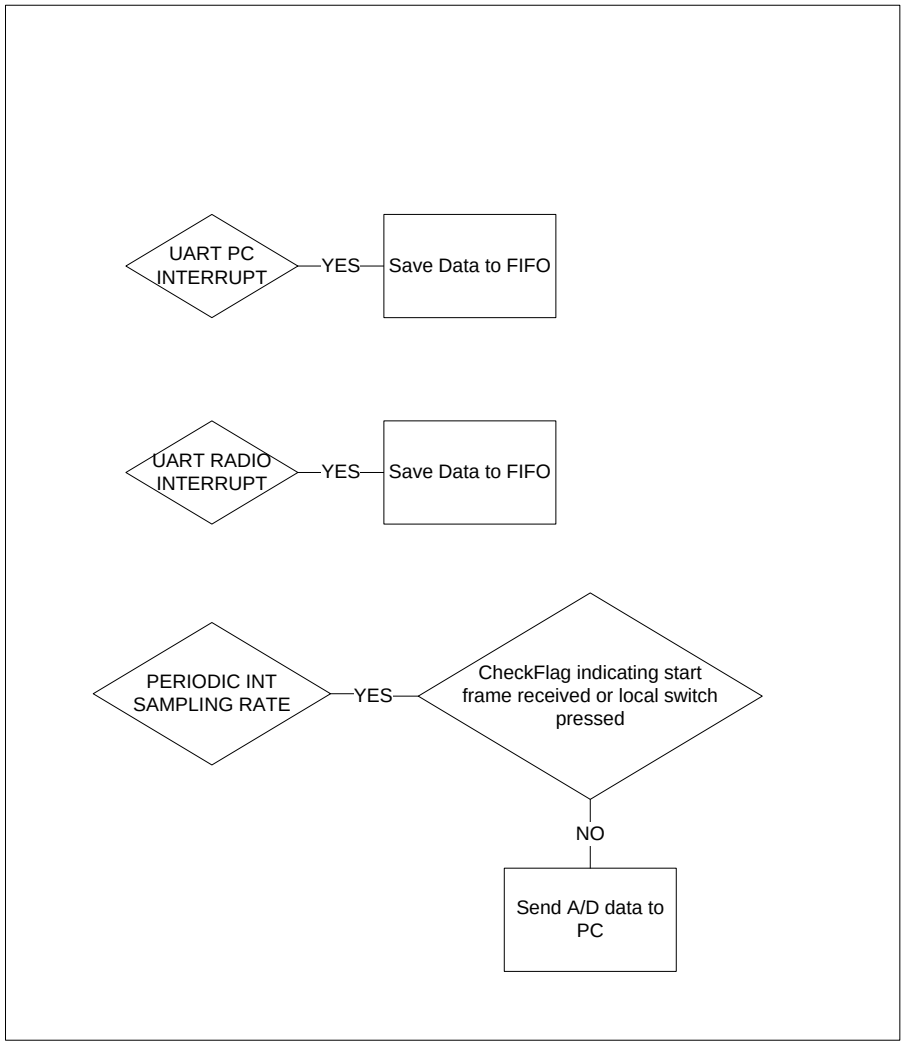

Figure 12. Transmitter module ISR flow chart

Interrupt service routines are basically data providers and the main foreground loop is the data consumer. The transmitter firmware constantly gathers data from the ADC and stores it in a FIFO. The FIFO buffer should be large enough so that any slow down on part of wireless link should not lose any data. There could be many reasons for wireless link to slow down for example too much noise at the receiver antenna from other RF sources, attenuation due to distance between transmitter and receiver, signal attenuation due to obstacles etc.

4.2 Receiver

Receiver communication consists of foreground loop called main loop and a number of interrupts serving the foreground loop in the background (see Figure 13 for Receiver main loop flow chart and Figure 14 for Interrupt service routines flow chart).

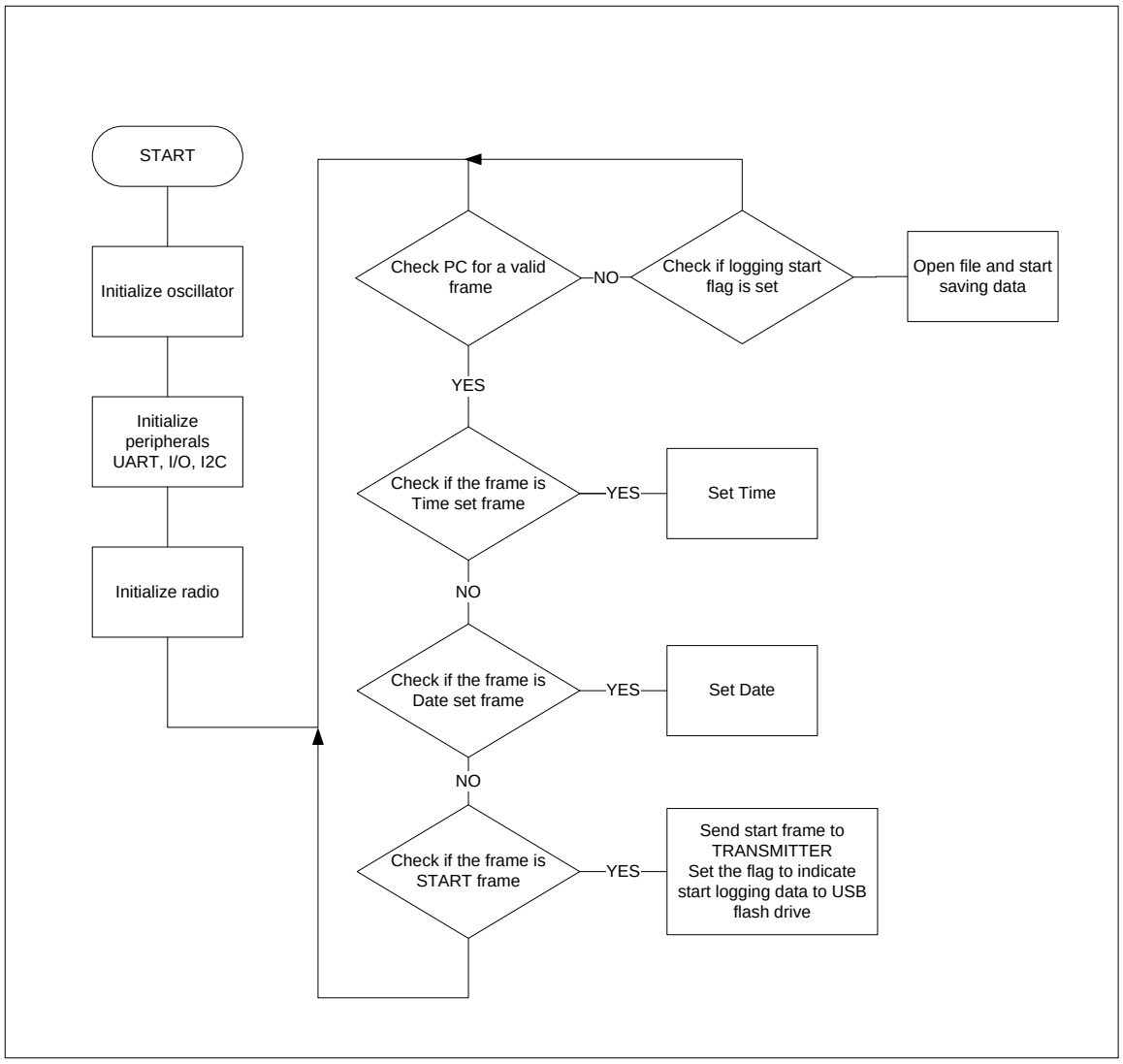

Figure 13. Receiver module main loop flow chart

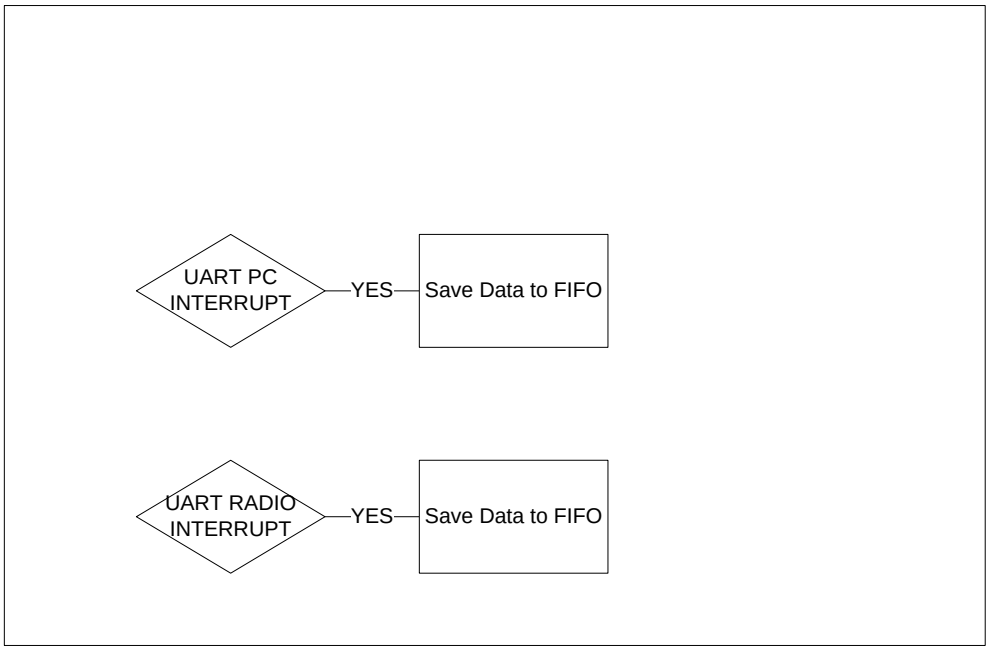

Figure 14. Receiver module ISR flow chart

Optimum method should be used when writing data on USB flash drive. Since USB flash drives are flash based the most efficient way to write data on flash is to write data in size equivalent to sector size. Writing data size equivalent to sector size increases the USB data throughput to the flash drive which prevents any FIFO buffer overflow in the receiver firmware

4.3 Signal conditioning and Amplifier

Following is the flow chart for signal conditioning and amplifier firmware (see Figure 15 for Signal conditioning and amplifier main loop flow chart and Figure 16 for Interrupt service routines flow chart). AMP is configured at start-up of transmitter and/or when configuration data is send to the transmitter from PC

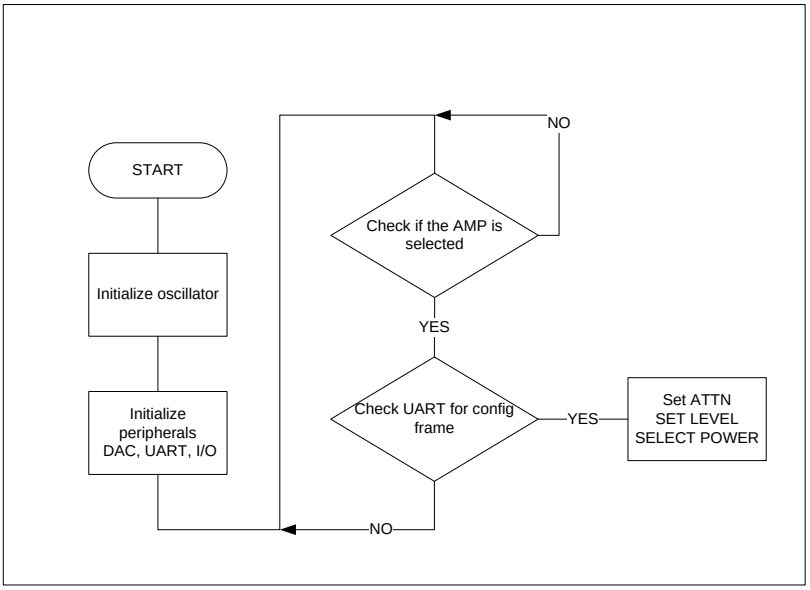

Figure 15. SCA main loop flow chart

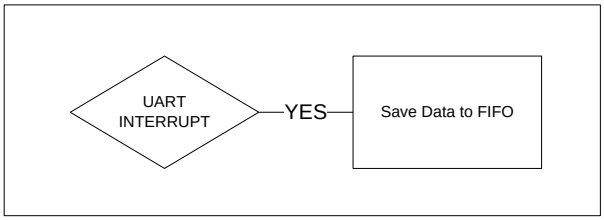

Figure 16. SCA ISR flow char

4.4 Configuration

As for any processor controlled device there is almost always a need for configuration data used for defining the behavior of the system. In case of power loss or system restart configuration should be restored to last known good configuration which usually happens at system start up. In case the system is powered the very first time there should be at least some default configuration loaded into the system so the system can function with a known good configuration (see Figure 17 for Power up sequence)

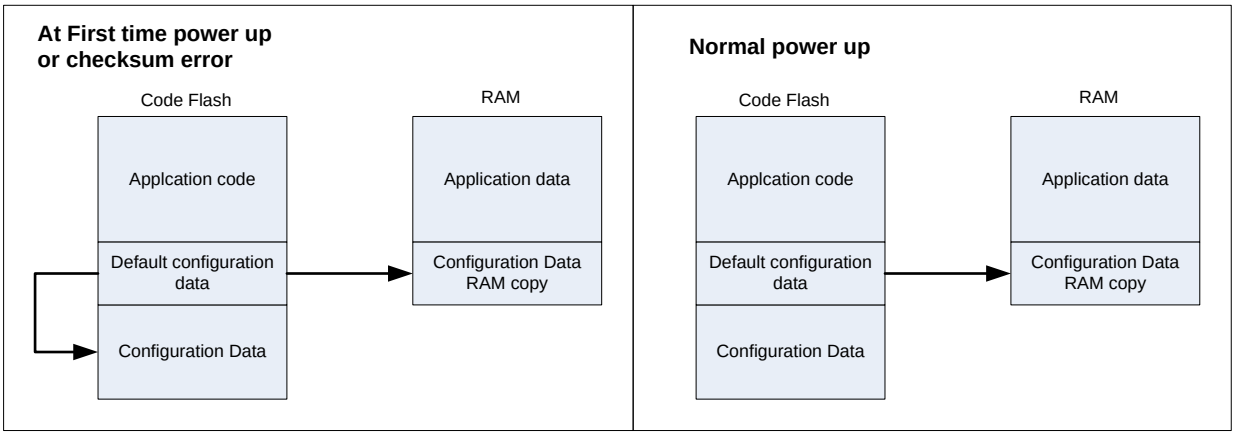

Figure 17. Power up sequence: Left – first time power up, Right - Normal power up

Configuration data can be user editable or factory set or both. In case of receiver there is no user configurable data, all the configuration is fixed and hard coded in the firmware. As far as transmitter is concerned it has both user editable and factory set configuration. Configuration data for transmitter is actually for both the transmitter main board and SCA board. Following shows the configuration data for transmitter board and its explanation of its purpose and use.

4.4.1 Configuration data

Channel selection: indicate that this configuration data is for which channel one out of maximum 8 channels can be selected.

Channel active: indicate if the channel is active or not.

Attenuation: level of attenuation required before the signal is presented to A/D converter.

DAC value: DAC value for setting signal level before presented to A/D converter.

Sampling rate: this field is user configurable and does not pertain to a specific channel rate it's a system wide configuration for transmitter board only. Sampling rate is can be selected anywhere from 300 samples/sec to a maximum of 5 K samples/sec.

Checksum: used for data integrity.

Using code flash for configuration is not straight forward, since one cannot write into flash memory code space from where the code is fetched and executed, solution to this problem is to copy the code, which dealt with writing data into flash memory, into RAM and execute the block of code from RAM. This way the code is not executed from the same space as the data resides (see Figure 18 – Code Execution from RAM) .

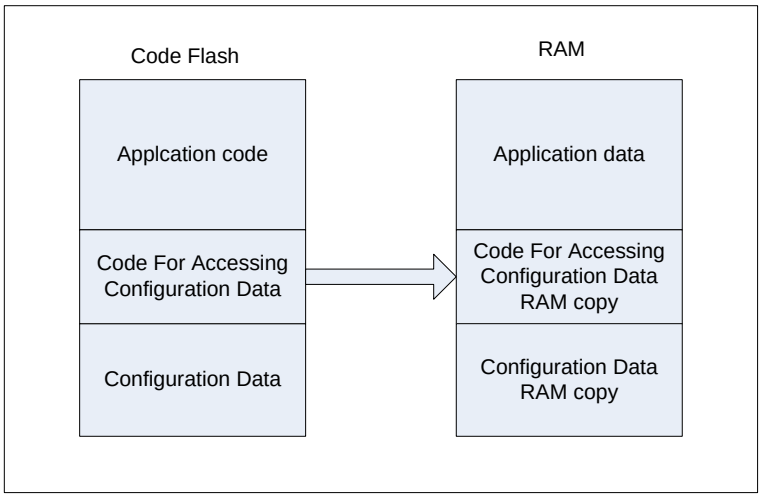

Figure 18. Code Execution from RAM

5 Testing

The system is tested with vibration sensor which is set to produce a value of -1 to +1 volt vibrating at 50Hz. The results are shown using PC interface software that displays data graphically (see Figure 19 for real time plot and Figure 20 for saved data plot).

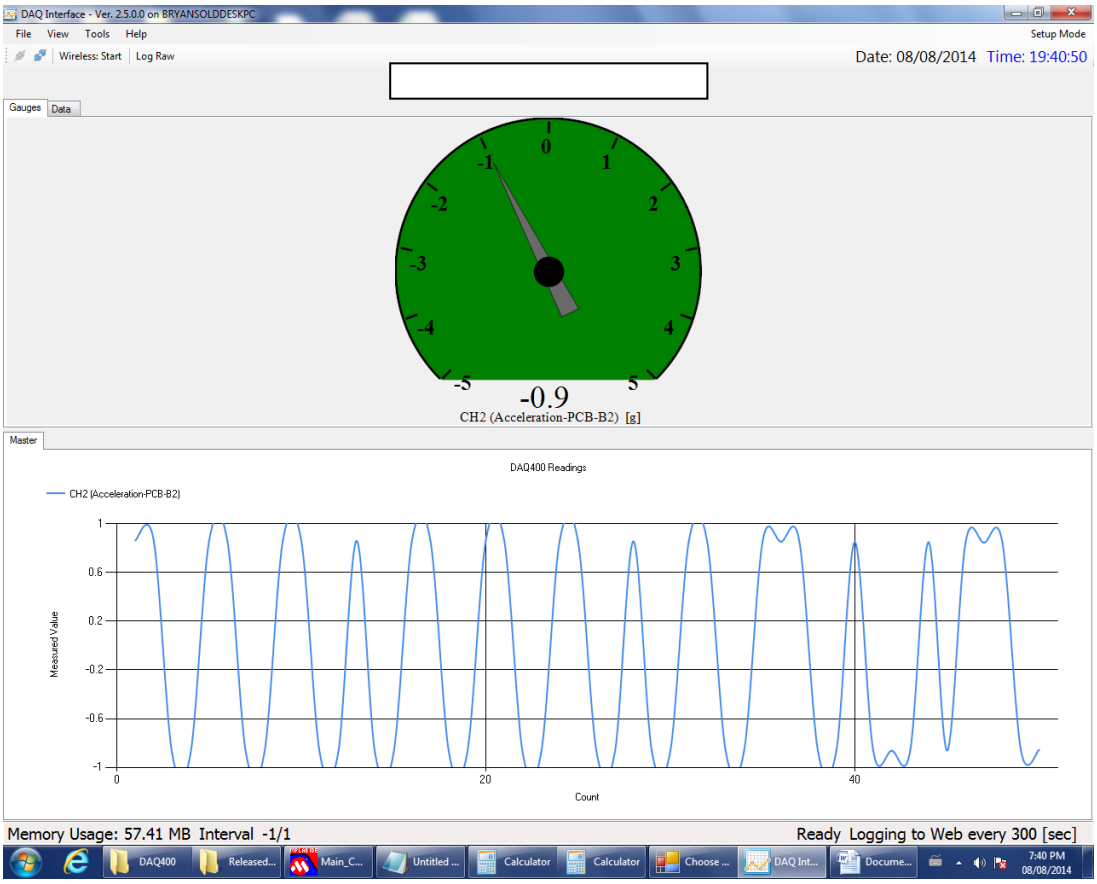

Figure 19. Real Time Data Plot

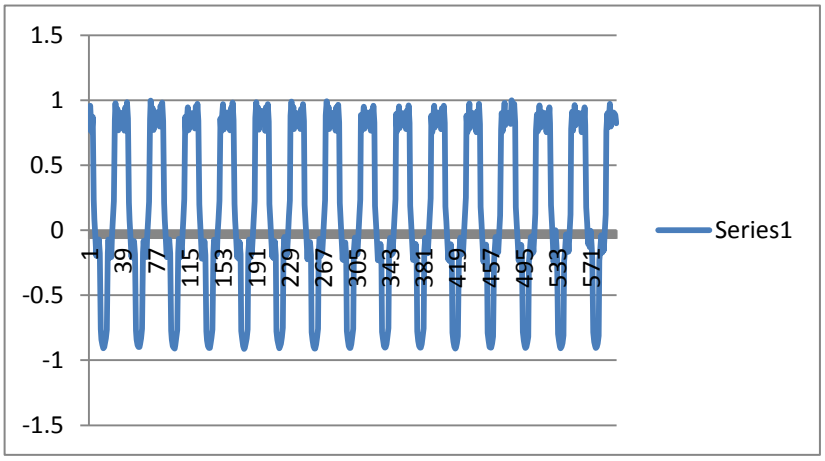

Figure 20. Saved Data Plot

6 Future work

For future work following are the area of improvement

- More powerful processor with reasonable RAM for buffering A/D values
- More versatile A/D converter
- Improved USB host interface
- Real time operating system for more structured code

Let's divide the future improvements in terms of hardware and firmware and discuss them individually.

6.1 Hardware

Following are the three proposed hardware design, each one has their own pros and cons. Each of the options will be discussed

6.1.1 Option 1

This option mostly uses the current hardware with the exception of using a dedicated chip for logging data to USB flash drive; the chip receives digital data from serial port and save it to a file in USB flash drive. The logging chip has command interface to create file for read and write, delete file, create directory etc. The logging chip has USB host interface and corresponding stack for Mass Storage Device to communicate and control USB flash drive. Using this option will free up the processor time which it used to maintain USB stack and more time will be available for other receiver functions. Option 1 also uses external serial EEPROM for storing configuration (see Figure 21 for Hardware option 1)

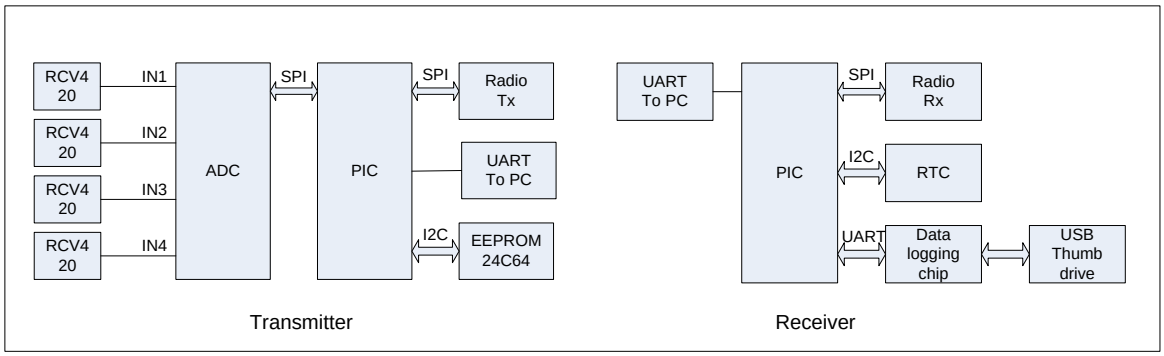

Figure 21. Hardware Option 1

6.1.2 Option 2

Option 2 uses cortex arm processor, the reason for recommending arm cortex processor is its availability of tools and the community support one can get since they are hugely popular and supported heavily by on-line community. The development tools like compiler, linker, debugger and integrated development environment are freely available and are of good quality. On the hardware side lots of low cost development hardware is available for development before deciding on the final design (see Appendix for detail). As for firmware development process lots of ports are available for operating system, file system, networking stack, USB stack and number of libraries with community support. Industry Standard gcc and gcc++ compilers are freely available with sample code and good documentation (see Figure 22 for Hardware option 2)

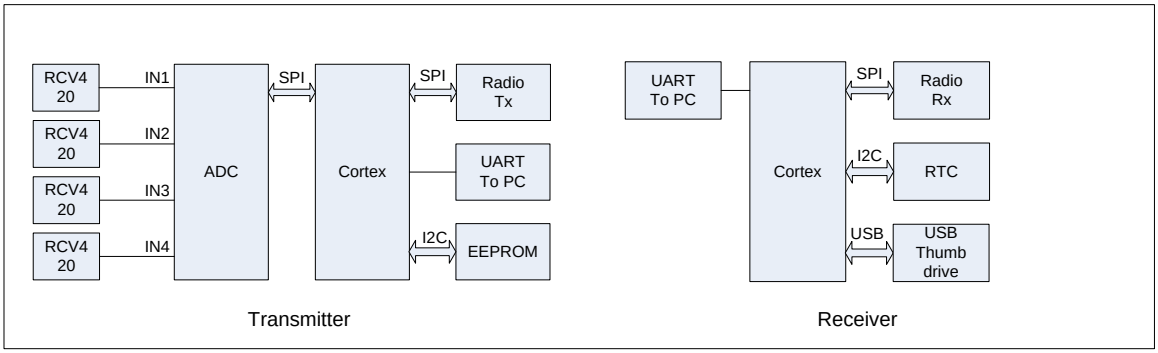

Figure 22. Hardware Option 2

6.1.3 Option 3

Option 3 uses much more powerful processors with memory management unit running Linux operating system. The benefits of using Linux operating system are many, as it is heavily supported by on-line community with lots of libraries, tools, debugger are available (see Figure 23 for Hardware option 3).

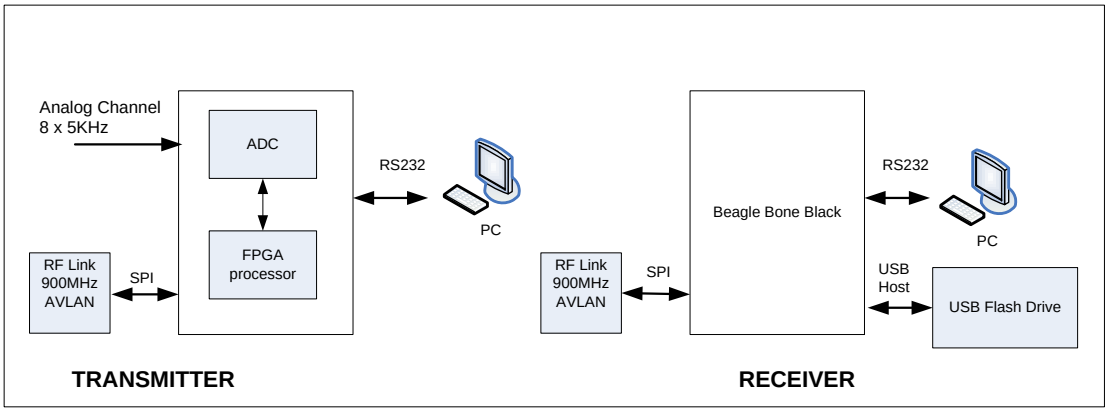

Figure 23. Hardware Option 3

6.2 Firmware

To make the firmware more structured and more flexible, inclusion of real time operating system like free rtos, tinyOS or uCOS-II is recommended. Real time operating system makes the firmware more structured, easily maintainable and more efficient. Instead of one single endless loop the firmware is structured into tasks where each task is an endless while loop performing specific function assigned to it.

6.2.1 Transmitter firmware

Transmitter firmware can be structured into the following tasks (see Figure 24 for Transmitter module tasks)

- A/D conversion task
- RF communication task
- Configuration task
- Supervisory task

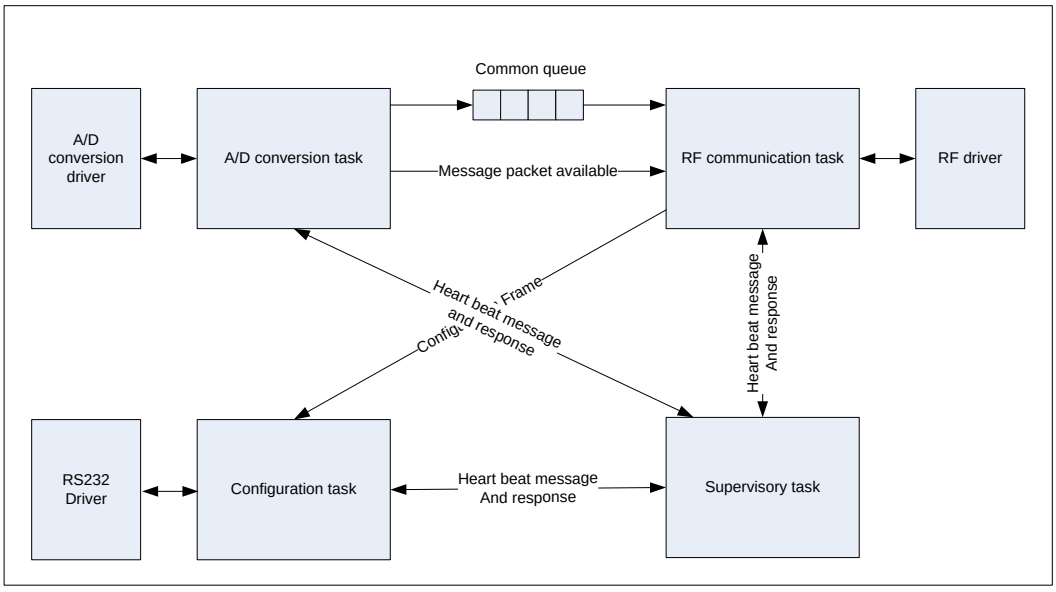

Figure 24. Transmitter module tasks

6.2.1.1 A/D conversion task

A/D conversion task is synchronized with A/D conversion driver and RF communication task. A/D converted data from A/D driver is saved in common queue access by both A/D conversion task and RF communication task and a message is send to RF task that the packet in the queue is waiting to be delivered to the radio communication driver to be send out to the radio

6.2.1.2 RF communication task

RF communication task is synchronized with A/D conversion task and waits for the message from A/D conversion task that a complete packet is waiting to be delivered to the radio module. When this message is receive the RF communication task send the packet to the RF driver to be delivered to the RF module. This task also monitors the receive RF communication for configuration frame from the receiver module, if configuration frame is received from receiver module that frame will be send to configuration task and configuration task will with that frame as if it is received from PC since configuration is allowed from remote PC as well

6.2.1.3 Configuration task

Configuration task takes care of the communication with PC to get configuration data and also responsible for saving and retrieving configuration from non-volatile memory. This tasks monitors the communication from PC and waits for a valid frame, depending on the frame type the task acts accordingly for example if a frame is received for setting a given channel active the task sends a message to A/D conversion task to start reading A/D converted value for that particular channel

6.2.1.4 Supervisory task

Supervisory task supervises all the other tasks to ensure their wellbeing by sending a heartbeat message al periodic interval to each task and receive their response to this message, if a given task does not respond within a certain interval as a first response to this situation, the supervisory task forcefully re-initialized the un responsive task and waits for the response from the next heart beat message. If for example the un-responsive task still does not respond the supervisory task restarts the whole system

6.2.2 Receiver firmware

Receiver firmware can also be structured into the following tasks (see Figure 25 for Receiver module tasks)

- USB host task
- RF communication task
- Configuration task
- Supervisory task

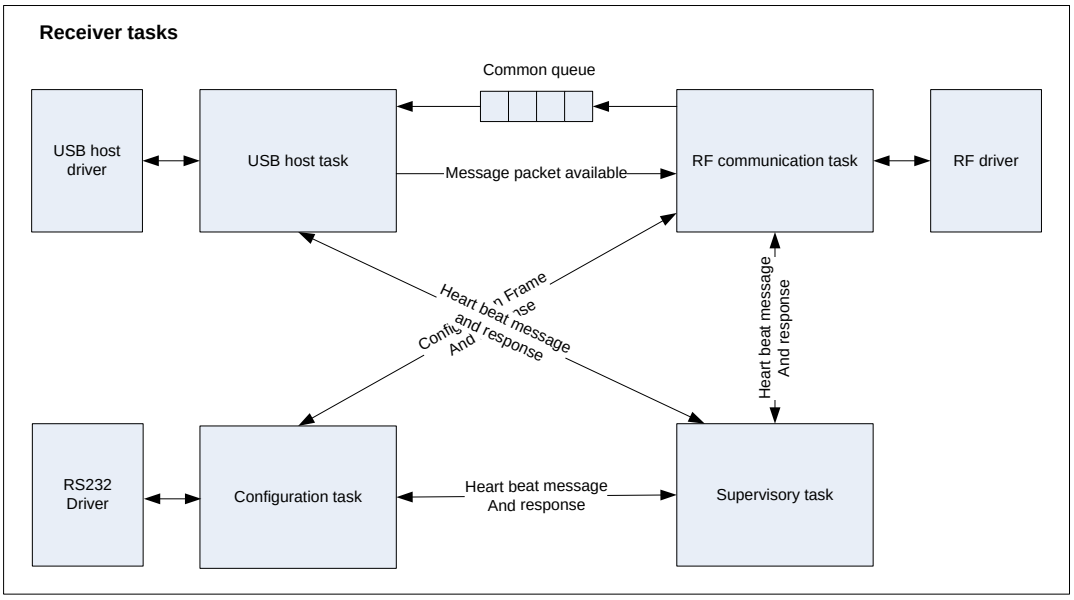

Figure 25. Receiver module tasks

6.2.2.1 USB host task

USB host task synchronizes with RF communication task. Any A/D conversion packet received from the RF driver to RF communication task is saved on a common queue accessed by both USB host task and RF communication task and a message is send to USB host task indicating that a valid A/D conversion frame is waiting to be fetched from the queue and should be saved on USB flash drive. USB host task after receiving this message saved the packet to USB flash drive

6.2.2.2 RF communication task

RF communication task synchronizes with USB host task and configuration task. Any A/D conversion packet received from RF driver to RF communication task is saved on a common queue accessed by both USB host task and RF communication task and a message is send to USB host task indicating a valid A/D conversion frame is waiting to be fetched.

6.2.2.3 Configuration task

Configuration task synchronizes with RF communication task. Any configuration or startup frame receiver from RS232 driver to configuration task is send to RF communication task, as a message, to be sent to transmitter via RF driver. Response packet receive from the RF driver to RF communication task is send to configuration task as response message. The configuration task waits for a certain duration for the response message before retry to send the original message

6.2.2.4 Supervisory task

Supervisory task works the same way as in transmitter task

7 Conclusions

The project does fall short on some features but we were able to log and monitor 4 channels of analog data sampled at 2 KHz. Radio-1 does have some limitations as the highest over the air data rate is 300 Kbps and we have seen from the calculations that it is not sufficient data rate to monitor and logs 4 channels at 5 KHz sampling, which is the ultimate goal. Nevertheless the future work does promise to overcome this bottleneck by using radio-2 which has the maximum over the air data rate of 1.55 Mbps. Also for future work more powerful and resourceful processor will make a difference make the hardware and firmware more structured especially for firmware using real time operating system streamlines the development process and makes it more robust and easier to maintain.

As far as testing data is concerned it's difficult to include the test data on this report since each data file is tens of Mega Byte in size and a dedicated PC software is used to plot the data and make some sense of the results

References

1. Sean Casey (2005) “ Design of a small form factor high-frequency wireless data acquisition and logging system” Master’s thesis, Department of Computer Science, University of Massachusetts Lowell.
2. Alpaben Bhesania (2001). “Wireless data acquisition system for health monitoring of civil infrastructure” Master’s thesis, Department of Electrical and Computer Engineering, Southern Illinois University Carbondale.
3. Hua Fang; Ming Tang; Lian Peng, "Wireless data acquisition system based on ARM," *Electrical and Control Engineering (ICECE), 2011 International Conference on* , vol., no., pp.2817,2820, 16-18 Sept. 2011
4. Chen Z, Casciati F. A low-noise, real-time, wireless data acquisition system for structural monitoring applications. Structural Control and Health Monitoring 2014;21(7):1118-1136
5. JinLin Hu , Shangyou Zeng , Zhengzhen Zhang . The Design of Wireless Data Acquisition System Based on STM32 and Virtual Instrument. 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing 2012 2012 SEPT.:1-4

Appendix

Processor

Apart from interfaces CPU is common for both modules and there are two options available for selecting processor

1. PIC series
    - 16bit processor from microchip
2. LPC3250 is an ARM processor from NXP with external memory interface. Main features pertaining to the requirements are
    - External memory interface (DRAM)
    - Dual SPI (one for radio and one for A/D converter)
    - UART (to communicate with PC)
    - USB host (to interface with Mass Storage Device – USB flash drive)

Evaluation boards

Embest

(see Figure 26 for Embest evaluation board)

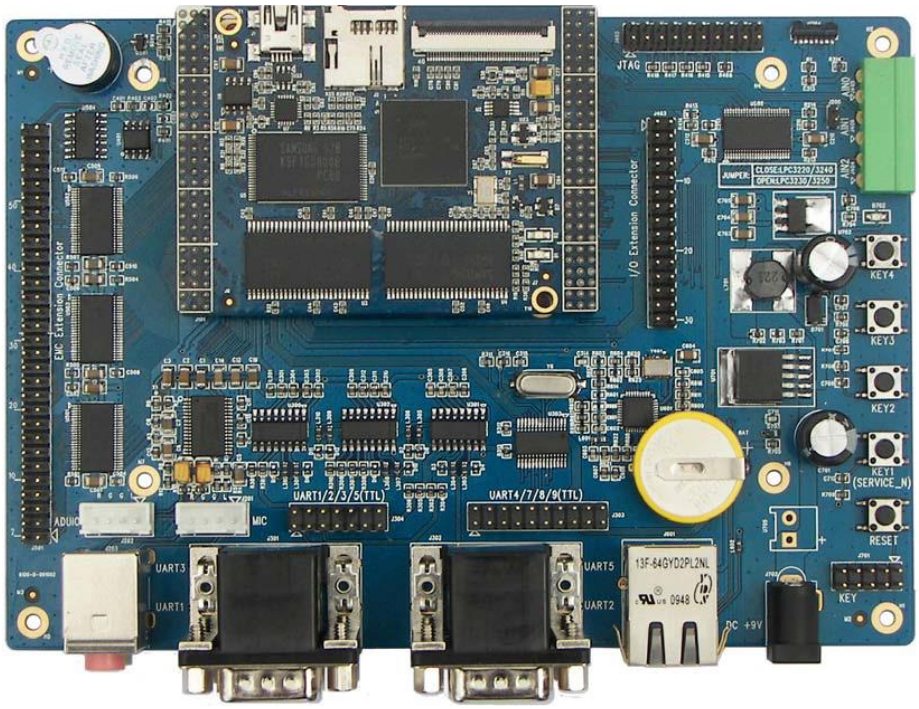

Figure 26. LPC-3250 Evaluation board from Embest

The board comes with a CPU adder board (mini3250), and this adder board can be purchased separately to plug-in the own prototype board. The board is available from Embest and their web site is www.embedinfo.com

Development IDE is available from http://lpcxpresso.code-red-tech.com/LPCXpresso/ and it is not free, it's about $1000 value.

Phytec

Another evaluation board is available from www.phytec.com this also comes with mother board and a plugin CPU board to be plug-in in the prototype board (see Figure 27 phytec evaluation board and Figure 28 for plug-in CPU board)

Figure 27. LPC-3250 Evaluation board from Phytec

Figure 28. LPC-3250 plug-in CPU board

Supports multiple JTAG dongles like jlink, ARM real view ICE etc. for code development EmIDE is available free of charge

A/D converter evaluation board

(See Figure 29 for ADAS3023 evaluation board)

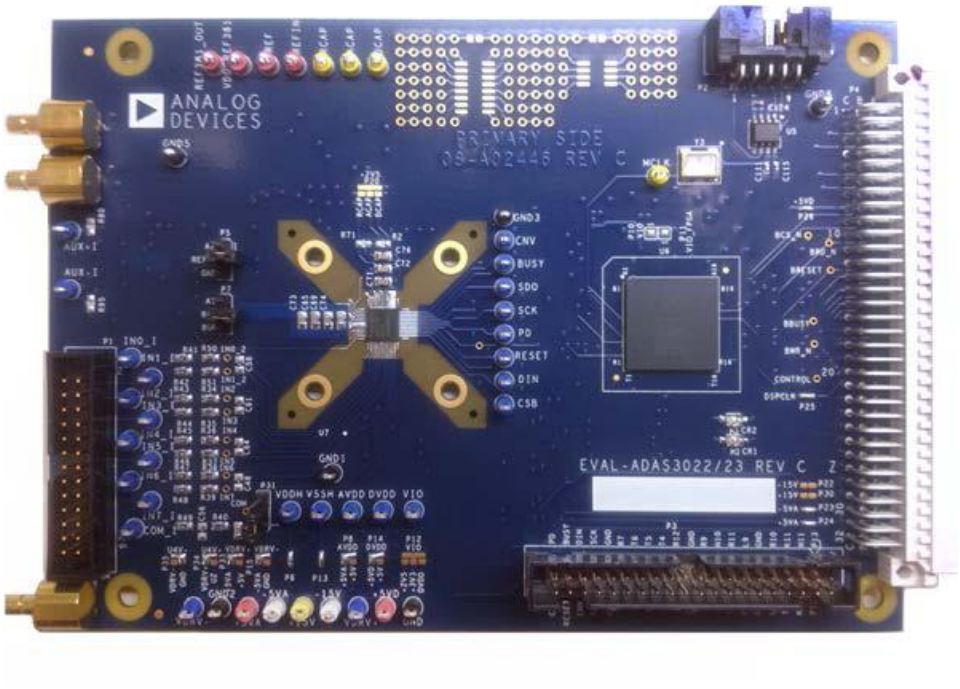

Figure 29. ADAS3203 A/D converter evaluation board from Analog Devices

VNC2 Evaluation board

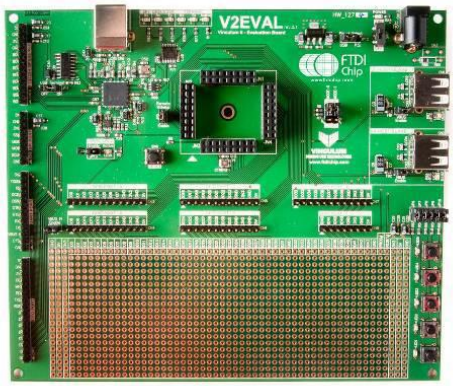

Figure 30. VNC2 Evaluation board from FTDI

Pre-compiled USB host to UART Bridge can be downloaded for evaluation, it has a serial port to connect to the board and/or to PC for evaluation and send/receive command and data (see Figure 30 for VNC2 evaluation board)