Chapter 1

INTRODUCTION

Electrocardiography (ECG) is most widely used in vital signal sensing and health monitoring. This method provides useful diagnostic information about the cardiovascular system. Main objective here is to correctly detect changes in some physiological and pathological conditions of humans.

Coronary diseases are increasing from past few decades. Hence continuous monitoring of the ECG signal of high risk patients can play an important role in immediately detecting pathological signatures and arrhythmias. This information can be sent to a centralized health monitoring system for early analysis. Further preventative actions can be carried out.

1.1 Introduction to ECG monitoring system

Electrocardiography (ECG or EKG) is detection and recording of electrical activity of the heart over a period of time. Electrodes are connected to outer surface of the skin, generally above chest or hands. This noninvasive procedure of ECG recording is termed as electrocardiogram (also ECG or EKG). ECG tests are used to record the electrical activities of the heart. ECG is used to measure the regularity of heartbeats and heart rate. ECG determines the size and position of the heart chambers or presence of any damage to the heart. Effects of drugs or devices, such as pacemaker can be analyzed from ECG.

The electrodes along with ECG device detect and amplify the tiny electrical changes on the skin that are caused when the heart muscle depolarizes during each heartbeat. Across cell membrane and at rest position, heart muscle cell has a negative charge. This is called as the membrane potential. Influx of the positive actions, Na^{+} and Ca^{++}, decreases the negative charge towards zero. This action is called depolarization. It activates the mechanisms in the cell that cause it to contract. A healthy, normal heart will have an orderly progression of a wave of depolarization. This is triggered by the cells in the sinoatrial node, which spreads out

through the atrium. Then it passes through the atrioventracular node and then spreads all over the ventricles.

This small rise and fall in the voltage is detected by electrodes, placed on either side of the heart. It is displayed as a wavy line on a screen or paper. Rhythm of the heart and weaknesses in different parts of the heart muscle can be analyzed from the results obtained on the display or paper.

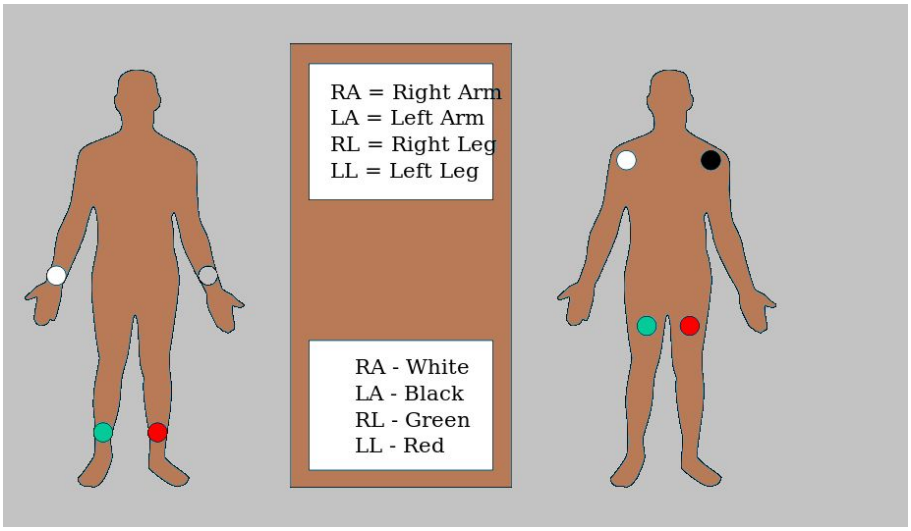


Fig.1.1. Electrode placement.

Two or more electrodes are used to detect ECG signal. They can be combined into a number of pairs. For example left arm (LA), right arm (RA) and left leg (LL) electrodes form three pairs LA+RA, LA+LL, and RA+LL as shown in the figure above.

Output from each pair of electrodes is known as a lead. ECGs can be classified into different types, based on the number of leads that are used to record it. For example 3 lead, 5 lead or 12 lead ECGs. In 12 lead ECG, 12 different electrical signals are recorded on a paper at approximately at the same time. It is often used as a one off recording of an ECG. 3 and 5-lead ECGs tend to be monitored continuously and viewed only on the screen of an appropriate monitoring device. Example can be during an operation or while being transported in an ambulance. In 3 or 5 lead ECG, permanent recording may or may not be provided depending upon the equipment.

A typical ECG cycle of heart beat consists of a P wave, a QRS complex, a T wave and a U wave as shown in figure below. These are normally visible in 50 to 75% of ECG's. The baseline voltage of the electrocardiogram is known as the isoelectric line. Isoelectric line is measured as the portion of the tracing following the T wave and preceding the next P wave.

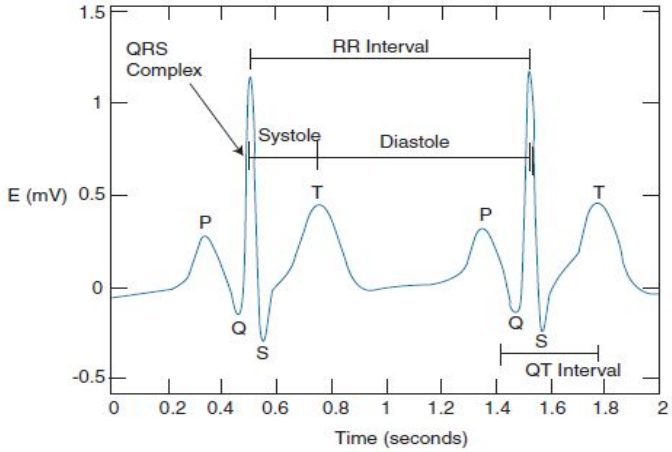


Fig. 1.2. ECG signal indicating valley peaks.

Earlier, original four deflections were noted. Fifth deflection was discovered after mathematical correction for artifacts were introduced by early amplifiers. Letters P, Q, R, S, and T were identified by Einthoven for the wave tracing.

The P peak is produced by a contraction of the atria. The R peak shows the ending of atrial contraction and the beginning of a ventricular contraction. Finally, the T peak marks the ending of a ventricular contraction. The magnitude of the R peak normally ranges from 0.1 to 1.5 mV. If the R peak is narrow and high, you have a strong heart. A complete heart cycle consists of systole and diastole phases. The former refers to the contractile (pumping) phase. Diastole refers to the resting (filling) phase.

1.2 Pulse Measurement

There are various methods for measuring pulse rates. For instance, one can manually measure someone's pulse rate by touching his wrist (radial artery) or neck (carotid artery). We can find a pulse in a person's elbow (brachial artery) and groin area (femoral artery). When we find a pulse, we count the number of beats per minute. This is a simple way to take measurement, but it's easy to make errors. For example, if you use your thumb, you might mistake your pulse for the other person's.

We can also measure pulse rates mechanically. Mechanical methods are based on the conversion of blood vessel rippling into electric signals via a strain or pressure sensor. A piezoelectric sensor is typically used for such an application, but the sensor position is sensitive to the vessel location and vessel displacement. Conventional tonometers are large and impractical for long-term use.

Another option is to take measurements optically. Optical tissue measurements are based on light absorption levels. Measuring the difference in light absorption in infrared and red bands enables you to determine the blood oxygen saturation level. This method yields more than a simple pulse rate measurement. Light modulation curve analysis can also detect respiratory system illnesses.

This proposed work involves pulsometers for electrical pulse rate signal detection [11]. Routinely this technique is used in hospitals as a tool for identifying cardiac disorders. It's the most accurate method for heart rate measurement. To detect electrical signals, we must place two or more electrodes on your patient's hands. Analyzing the ECG data from the electrodes provides a quantitative description of the heart's electrical activity. The average heart rate is calculated by measuring the time interval (the RR interval) between two consecutive R peaks, taking the average reciprocal of this value over a fixed period of time (usually 15, 30, or 60 s), and then scaling to units of beats per minute (bpm).

1.3 Wireless Sensor Network (WSN)

WSN contain number of sensor nodes which consist of sensing, data processing, and communicating components. These nodes can be deployed especially in regions where remote monitoring is not possible. Regions can be thick forest, inaccessible terrains or areas affected by disaster. With evolution in fabrication technologies, size of devices have become smaller, consume less power and are priced low. This motivation has made size of the sensor node to shrink, with extending its computation capabilities. Number of such small sensor nodes can be deployed in a large region. Because of their increased capabilities, these nodes can hop data to their neighboring node using advanced network protocols. Data can be sent to any distant node through these clustered nodes. These nodes together perform large data processing activities at very low power. Putting together, WSN's provide advantages like flexibility, fault tolerance, high sensing fidelity at low cost [1]. Rapid deployment characteristics of sensor networks fancy their use in remote patient monitoring system and sports training facilities. Remote care helps to measure vital body signals such as blood pressure and body temperature from a remote distance.

1.3.1 Body Sensor Network (BSN)

Advantages of WSN have helped to design Body Sensor Network technology (BSN). BSN consists of number of sensors attached on different areas of body called Body Area Network shown in figure 1.3 (BAN) [2]. BSN technology is very useful in patients which require long term attention [12]. Main task here is to transfer healthcare monitoring from the clinic to the healthcare center.

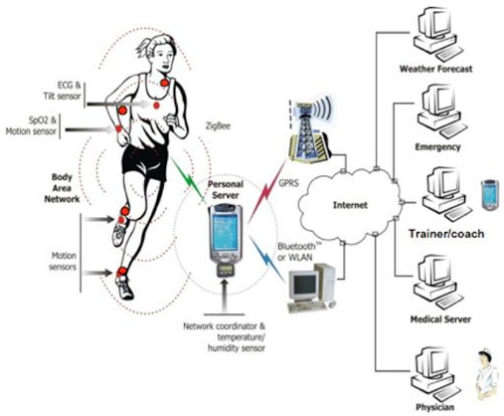


Fig. 1.3. Body Sensor Network.

Sensors on BAN act as nodes transmitting information across body shown in figure 1.3. These sensors individually collect data and transmit to the remote clinics or hospitals. This technique is very much helpful for elderly patients who face difficulty in visiting hospitals regularly and long term monitoring is recommended. Adhesive electrodes cannot be used for long term ECG monitoring. Nodes with non contact electrodes [3-5] are preferred for such patients. ECG monitoring system designed using microcontrollers require external amplification and filter circuits to acquire ECG signal. Thus, increase power consumption.

1.4 Problem Definition

Main difficulty faced in Wireless ECG detection system is, acquiring proper signal from patient's body. Signal can get corrupted from external sources like equipment noise, measurement noise (or electrode contact), electromyogram noise (muscle contraction), movement artifacts, baseline drift, instrument noise (ADC conversion process) and respiratory artifacts. To overcome these problems, additional circuitrary can be used. But this leads to increase in size, power consumption and cost, which are the major factors affecting WSN's. Other difficulty faced in WSN's is that number of packets transmitted/sec to the base station. More the number of packets, more load is imposed on network traffic. Increase in transmitter power consumption and reduction in network performance can be observed. This results in network congestion and reduced performance in receivers.

A system design involving high performance chip should be used to overcome problems faced in signal acquisition. Traffic on the network can be reduced by using an algorithm, which detect ECG signal accurately and less frequently.

This section provides an insight to the wireless ECG monitoring system. Literature review discussed in next section should help us choose best design to implement the ECG detection and transmission system.

1.5 Project Report Organization

The rest of the report is organized as follows:

Chapter 2 presents an overview of literature survey explaining the contributions of different people in designing low power wireless ECG monitoring systems.

Chapter 3 discusses about components used in ECG detection, transmitter and receiver systems.

Chapter 4 guides through the design flow of overall system.

Chapter 5 presents threshold adaptive algorithm for accurate ECG detection.

Chapter 6 provides information about the system requirements, in terms of the hardware and software requirements specifications.

Chapter 7 describes in detail the design and implementation considerations.

Chapter 8 shows classification of ECG signal, provides performance analysis of PSoC and wireless sensor from the obtained results.

Finally the report concludes with the scope for **future work** and **references**.

Chapter 2

LITERATURE SURVEY

Study involving designs of ECG devices which are small in size, easy to wear, convenient to use and consume less power are being carried out very extensively. Advantages of WSN have made a popular choice among other techniques in designing such portable devices.

2.1 Works related to low power wireless ECG system.

Ebrahim Nemati, M. Jamal Deen, and Tapas Mondal [6], did a research on "A Wireless Wearable ECG Sensor for Long-Term Applications". This work intends to build a small, portable two lead capacitive coupling ECG sensor and a wireless module. First objective of this work is to set up a low power ANT network for establishing connection, maintaining connection and channel operation. This is accomplished by using ANT node containing a engine and a microcontroller unit. Second objective is to impart some intelligence to switch node on and off. This intelligence is built based on their neighbor's nodes network activity.

Research work carried out by, Tsung-Heng Tsai, Jia-Hua Hong, Liang-Hung Wang, and Shuenn-Yuh Lee [7] on "Low Power Analog Integrated Circuits for Wireless ECG Acquisition Systems" contrasts WSN design using PSoC and Zigbee module. System on Chip (SoC) provides low power solution to acquire ECG signal. Zigbee protocol uses 2.4 GHz transmitter and receiver, which use low noise amplifiers, differential power splitter (DPS), and quadrature mixer for low-IF architecture. These blocks provide good conversion gain and signal switching, making it ideal for use in low power applications.

Paper involving work on, "A 700-µW Wireless Sensor Node SoC for Continuous Real-Time Health Monitoring" by Tee Hui Teo, Xinbo Qian, Pradeep Kumar Gopalakrishnan, Yee Shan Hwan, Kuruveettil Haridas, Chin Yann Pang, Hyouk-Kyu Cha and Minkyu Je [8] proposed a design of SoC that consumes 700 uWatt of power. This is a very low power system on chip. This can be configured as wireless sensor node for continuous and real time health monitoring. SoC transmits acquired ECG signal over 433.95 MHz channel. Transmitted

signal can be received by personal server like PDA. On PDA ECG signal is displayed and monitored. SoC consumes 700 uWatt with 0.7 V supply. This can run more than 200 hours, without changing it. This makes it a low power WSN solution.

Fei Hu, Yang Xiao [9] carried out a work on, "Congestion-Aware, Loss-Resilient Bio Monitoring Sensor Networking for Mobile Health Applications". Here they have proposed WSN design based on RF mote and Cypress's Programmable System on Chip (PSoC). PSoC's complex computational capabilities process different medical signals. Wavelet based signal decomposition technique extracts feature parameters. Based on these parameters normal or abnormal signal classification is done. RF mote transmits only abnormal signals to base station, thereby reducing network traffic, data throughput and transmission time.

Research work carried out by Fei Hu, Shruti Lakdawala, Qi Hao and Meikang Qiu [10] on , "Low-Power, Intelligent Sensor Hardware Interface for Medical Data Preprocessing", show WSN design using PSoC and MICA2 (Crossbow Inc.). PSoC is used to detect ECG signal. Thresholding algorithm is implemented in PSoC to limit the data transmission packets. Detected heart rate values are compared with stored values and transmission of data packet is done based on whether detected values are above or below threshold. This reduces congestion and data traffic over network. Low data rate reduces burden on MICA2 mote, thereby reducing power consumption. RF and MICA2 mote are complex to program, high cost and consume more memory space.

This section highlighted different work carried out in the field of wireless ECG monitoring system. Because of extended capabilities of PSoC, it makes best choice to use in ECG monitoring applications. Programming mote is complex and are priced high. Thus, simple, low cost transceiver module can overcome these difficulties. Next section proposes a wireless ECG monitoring system design.

Chapter 3

SYSTEM DESIGN

3.1 Block Diagram

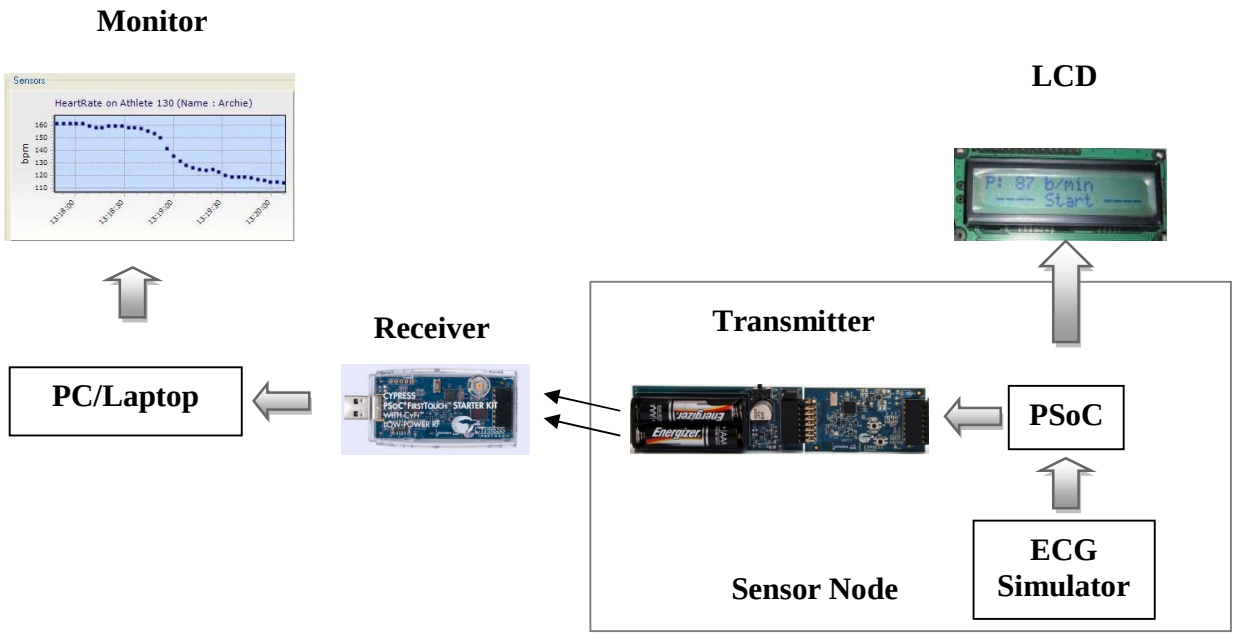


Fig. 3.1. Block Diagram of overall system.

Figure 3.1 shows the block diagram of wireless ECG monitoring system. Here the sensor node receives Electro Cardiogram (ECG) signals from the external simulator. The signal processing is done using Programmable System on Chip (PSoC) that contains amplifier, filters and Analog to Digital Converter (ADC). RF transmitter present on the sensor node transmits the heart signal to the base station. Base station can be a personal computer equipped with RF receiver. Computer is used for processing, storing and to displaying the result on monitor. GUI on monitor records and displays the heart rate in terms of BPM (Beats Per Minute). Alarm can be configured to indicate normal heart rate, low heart rate (bradycardia) and high heart rate (tachycardia).

3.2 ECG Signal Processing and Detection system.

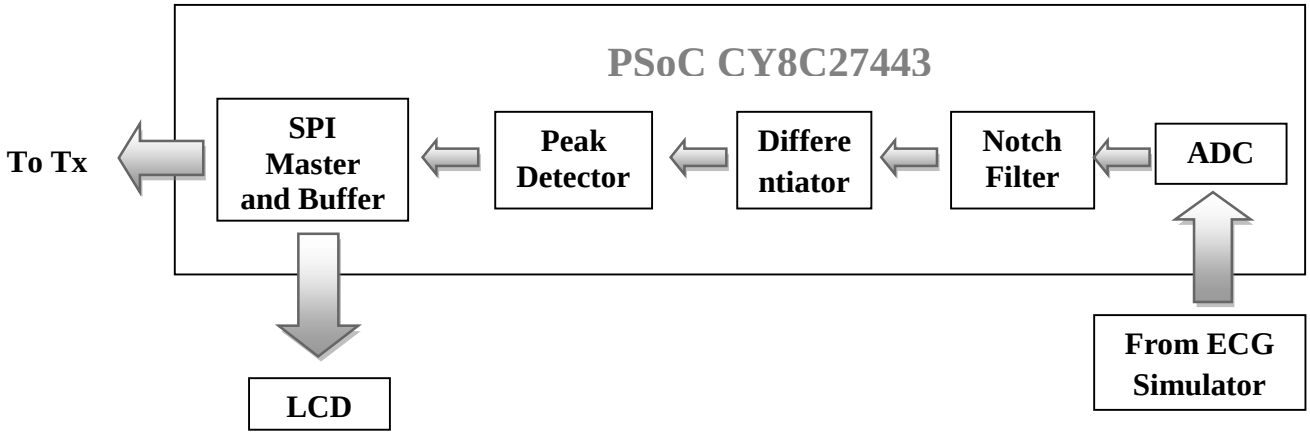


Fig. 3.2. Block Diagram of ECG signal processing and detection system.

Figure 3.2 shows block diagram of ECG signal processing and detection system. ECG signal from simulator is fed as input to this system. This system mainly contains Analog to Digital Converter (ADC), filters, peak detector and LCD. Description of the blocks is as given below,

PSoC: This block consists of Analog to Digital Converter (ADC), notch filter and differentiator implemented in PSoC CY8C27443. It has 8 bit microcontroller core, which achieve these functionalities.

ADC: Microcontroller processes digital data. Hence analog ECG signal from simulator should be converted into digital form. ECG signal is first amplified using Programmable Gain Amplifier (PGA) to the desired level. Then signal is digitized via a 14-bit incremental ADC. Processing of the ADC data stream is implemented in the firmware.

Filters: Digital filters are used to process the acquired ECG signal. It consists of second order IIR low pass notch filter and a differentiator.

- Notch filter: A second-order infinite impulse response (IIR) lowpass filters are used for power-line 50 and 60-Hz interference suppression. These filters have cut off frequency (Fc) of 25Hz. ECG signal passes through three cascaded low pass filters.
- Differentiator: After digital filtering, the pure ECG signal passes through the differentiator. Here the QRS complex of ECG signal is picked out. Differentiator

consists of a first order IIR high-pass filter with a cutoff frequency (Fc) of 70 Hz. This determines first order derivatives, which will be used to calculate pulse rate.

Peak detector: A smart peak detector with automatic threshold adjustment is used to detect number of R to R peaks. As the ECG signal exceeds the minimum threshold value, peak detector detects R peak. After 5 R-R peaks are detected, heat rate is calculated. Heart rate is calculated by scaling down R-R peaks detected in a given time into peaks detected per minute. This is measured in terms of Beats Per Minute (BPM).

SPI Master and Buffer: Heart rate calculated has to be sent to RF Transmitter for signal transmission to base station. There has to be a interface between ECG signal processing and detection system and RF transmitter. This achieved by implementing a Serial Peripheral Interface (SPI) between them. SPI master in full duplex mode is implemented at ECG signal processing and detection system as shown in figure below. SPI slave is implemented in RF transmitter module.

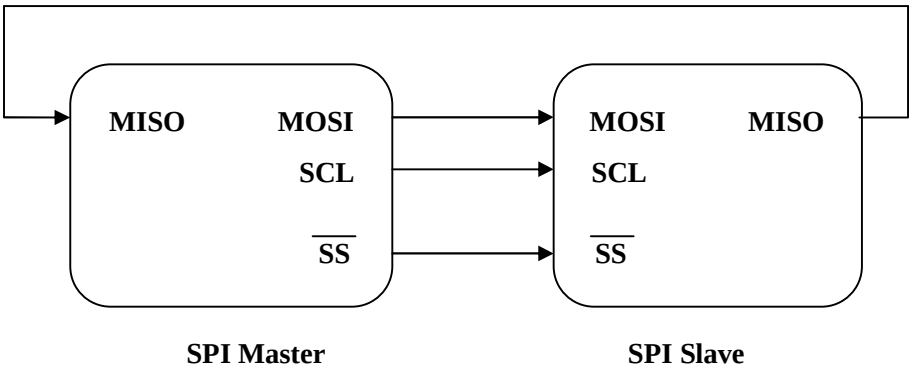


Fig. 3.3. Block Diagram of SPI in full duplex mode.

Following pins are used in SPI,

- MOSI is the Master Out Slave In line. It carries data from the master to the slave.
- MISO is the Master In Slave Out line. It carries data from the slave to the master.
- ~SS is the slave select line. This is used to select and enable a slave on the bus.
- SCLK is the clock signal. This synchronizes the data transmission in the system.

SPI in full duplex mode is used. When master wants to send a byte to the slave, it first loads a byte of data into its output shift register. SPI master controls the data transmission. It selects a slave as the destination. This is done by asserting low signal on ~SS line. The SCLK line is

then enabled and one bit of data is shifted out on the MOSI line with each rising edge of clock pulse. The slave shifts in the bit of data on the MOSI line on every falling clock edge. Received data is read from this input before next data arrives. To check consistency of the system, data has to be sent back to the master. Slave shifts data out from the output shift register on to the MISO line and the master reads in this data. Received data is put in the receive buffer register and displayed on LCD. Timing diagram representing various signals is shown below.

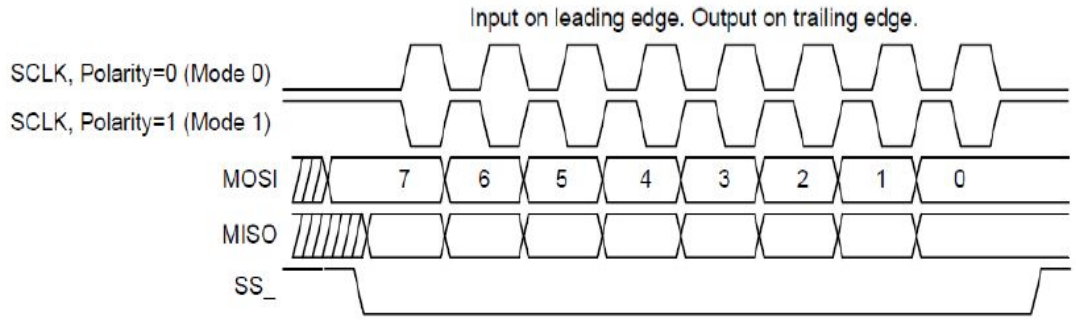


Fig. 3.4. Timing Diagram of SPI in full duplex mode.

In our work heart rate first is copied in transmitter buffer register. SPI master then sends heart rate serially, bit by bit to SPI slave on MOSI line. Before sending, it is buffered in external digital buffer and sent. This buffer is important to drive signal on MOSI line.

LCD: A 16x2 line LCD is used to display heart rate in terms of Beats per Minute (BPM). It also displays heart rate sent and received in hexadecimal on this LCD. If heart rate is less than 60 BPM, it displays message “Bradycardia”. If heart rate exceeds 90 BPM, it displays message “Tachycardia”, otherwise displays “Normal”.

3.3 ECG Signal Transmitter

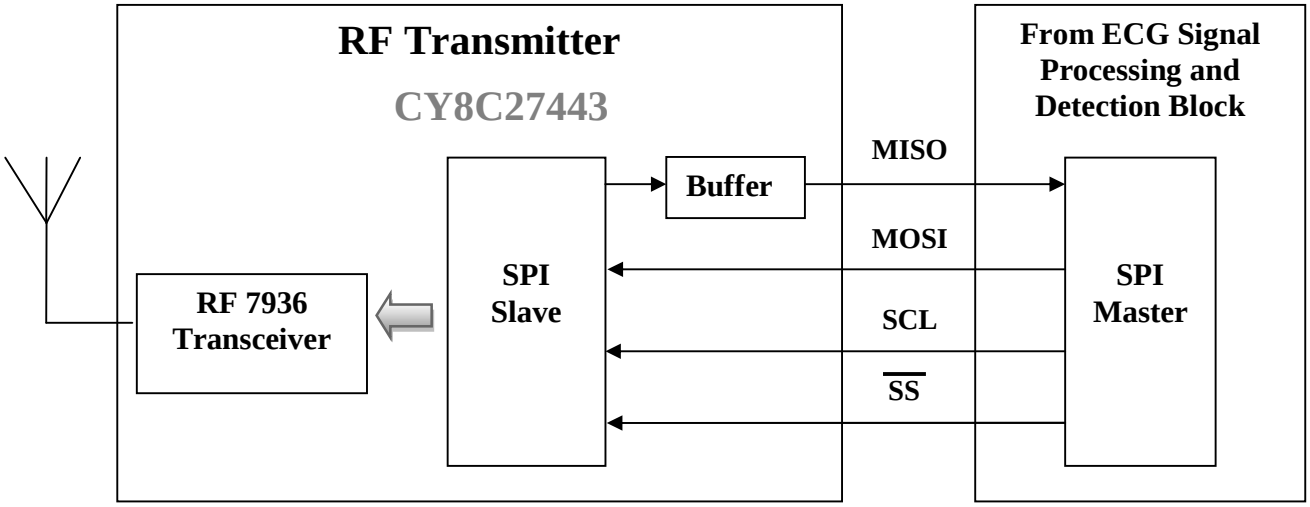


Fig. 3.5. Block Diagram of ECG signal Transmitter.

Cypress CyFi CY3271 [13] sensor node consists of an application PSoC and RF 7936 radio module. This application MCU is used to implement analog and digital blocks. It controls sensors, peripherals and RF transceiver. SPI, Buffer and CYFISNP modules are implemented in this application PSoC.

SPI Slave and Buffer: SPI slave receives buffered data from SPI master on MOSI line. Transmitter polls every second to check whether ~SS line is asserted or not. When ~SS line is held low, it starts receiving serial data bit by bit from master on MOSI line on every rising edge of clock pulse. Received data is put inside receive buffer register for further transmission. As transmitter doesn't have LCD to display received data, this data has to be sent back to ECG detection and processing system. This is achieved by buffering data on external buffer and sending it on MISO line of SPI slave.

RF Transceiver: Here transceiver acts as transmitter as well as receiver. Transmitter is used to send heart rate to base station using wireless link. We can obtain current configuration on network, bind status or change report timer rate from base station using GUI sense and control dashboard. Both RF transmitter and RF receiver (hub) use CYRF7936 2.4GHz transceiver for communication purpose.

3.4 2.4GHz CyFi Transceiver

Some of the important features of transceiver are listed below,

- 2.4-GHz direct sequence spread spectrum (DSSS) radio transceiver.
- Operates in the unlicensed worldwide industrial, scientific, and medical (ISM) band (2.400 GHz to 2.483 GHz).
- DSSS data rates up to 250 kbps and Gaussian frequency shift keying (GFSK) data rate of 1 Mbps.
- 80 discrete channels.
- Auto transaction sequencer (ATS) i.e no MCU intervention.
- Framing, length, CRC16, and auto acknowledge (ACK).
- Power management unit (PMU) for MCU.
- 16 byte transmit and receive separate FIFOs.
- Dynamic data rate reception.
- Receive signal strength indication (RSSI).
- 4-MHz SPI microcontroller interface.
- Up to +4 dBm output power.
- Up to -97 dBm receive sensitivity.

Figure below shows block diagram of CYRF 7936 transceiver block diagram.

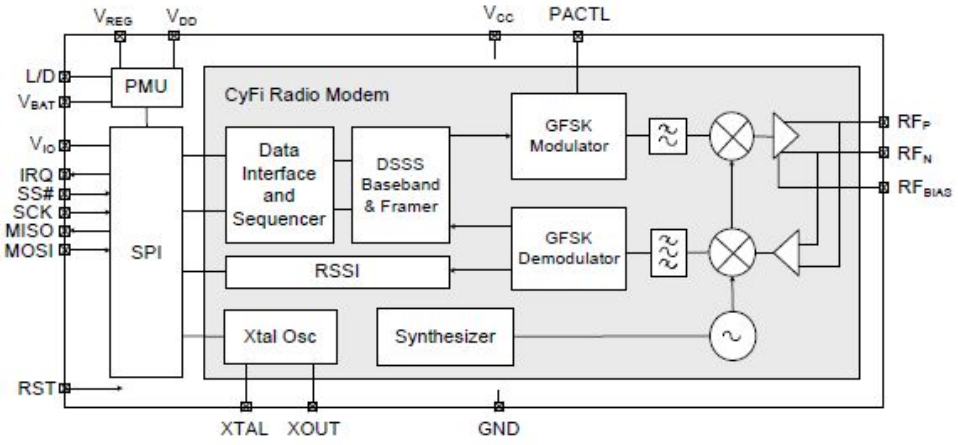


Fig. 3.6. Block Diagram of RF7936 Transceiver

The RF7936 IC is designed to implement wireless device links operating in the worldwide 2.4GHz ISM frequency band. This contains a 2.4GHz CyFi radio modem, featuring a 1Mbps GFSK front end radio, packet data buffering, packet framer, DSSS baseband controller and RSSI. For data transfer and device configuration, RF7936 uses a simple SPI interface. CyFi radio modem supports 98 discrete 1MHz channels. DSSS spreading, despreading, start-of-packet (SOP), end of packet (EOP) detection, and CRC16 generation and checking are performed by baseband. The baseband may be configured to automatically transmit ACK handshake packets whenever a valid packet is received. In receive mode, the device is always ready to receive data transmitted at any of the supported bit rates. But packet framing must be enabled before receiving. This enables the implementation of mixed rate systems in which different devices use different data rates. Implementation of dynamic data rate systems that use high data rates at shorter distances or in a low moderate interference environment or both are possible. It changes to lower data rates at longer distances or in high interference environments or both. In addition, the RF7936 IC has a power management unit (PMU). This allows direct connection of the device to any battery voltage supporting range of 1.8 V to 3.6 V. PMU conditions the battery voltage to provide the supply voltages required by the device and external devices.

For other details related to transceiver please refer Appendix A.

3.5 ECG Signal Receiver (Hub)

Figure 3.7 show overall receiver system. RF Receiver consists of RF transceiver, I2C application interface, memory management, I2C slave interface and data buffer. This receiver module consists of two CY8C24894 PSoC's. Slave PSoC controls RF transceiver for reception of data. Master PSoC provides USB to I2C bridge functionality to communicate with PC or laptop.

RF Transceiver: Here heart rate transmitted by transmitter gets received here. RF transceiver communicates with Star Network Protocol (SNP). It receives data packets sent by transmitter. Received data packets get written on application data buffer.

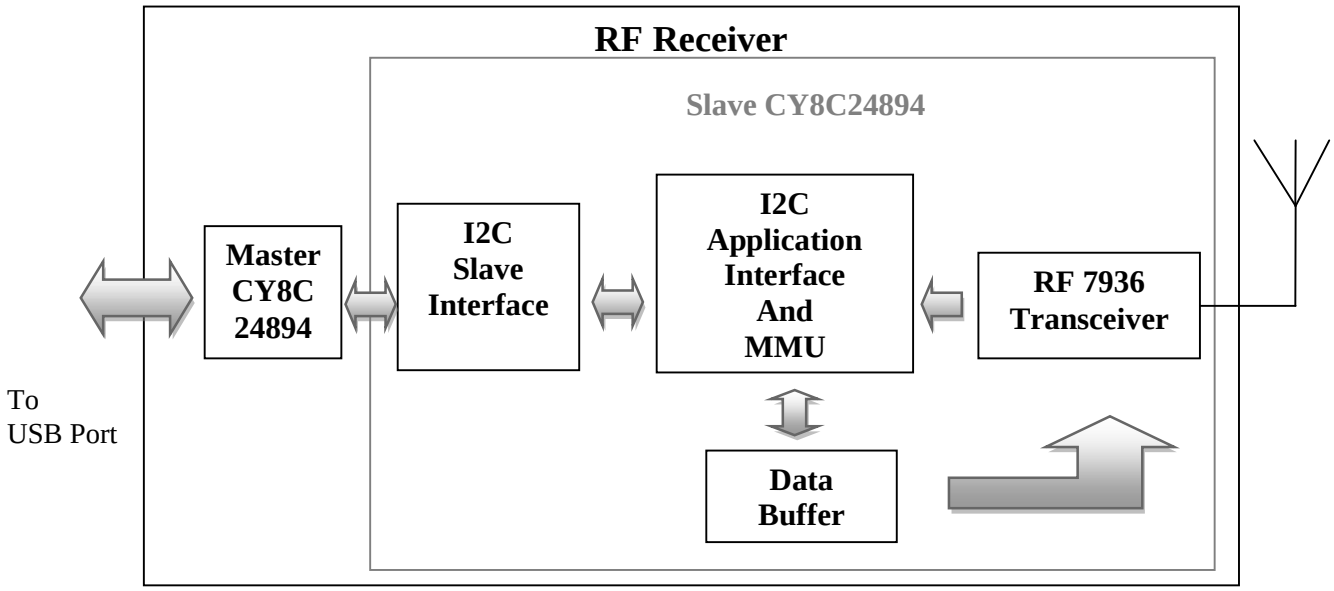


Fig. 3.7. Block Diagram of ECG signal receiver (Hub).

Memory Management: Memory Management Unit (MMU) is responsible for writing buffered data onto application hub data buffer. Master PSoC reads this data packet, by issuing command to buffer through I2C interface. Thus, GUI sense and control dashboard receives transmitted data here and outputs it on screen. MMU avoids any buffer overflows and receiving junk data when polling for data.

I2C Interface: Master and Slave PSoC's communicate with each other through this interface. Master sends command to read data from buffer. Slave sends data to Master through I2C slave interface. This read data is displayed on GUI Sense and Control Dashboard (SCD).

This section dealt with internal blocks and their functionalities of ECG detection, transmitter and receiver systems. Next section deals with work flow of system.

Chapter 4:

WORK PROCESS OF THE SYSTEM

4.1 ECG Signal Processing

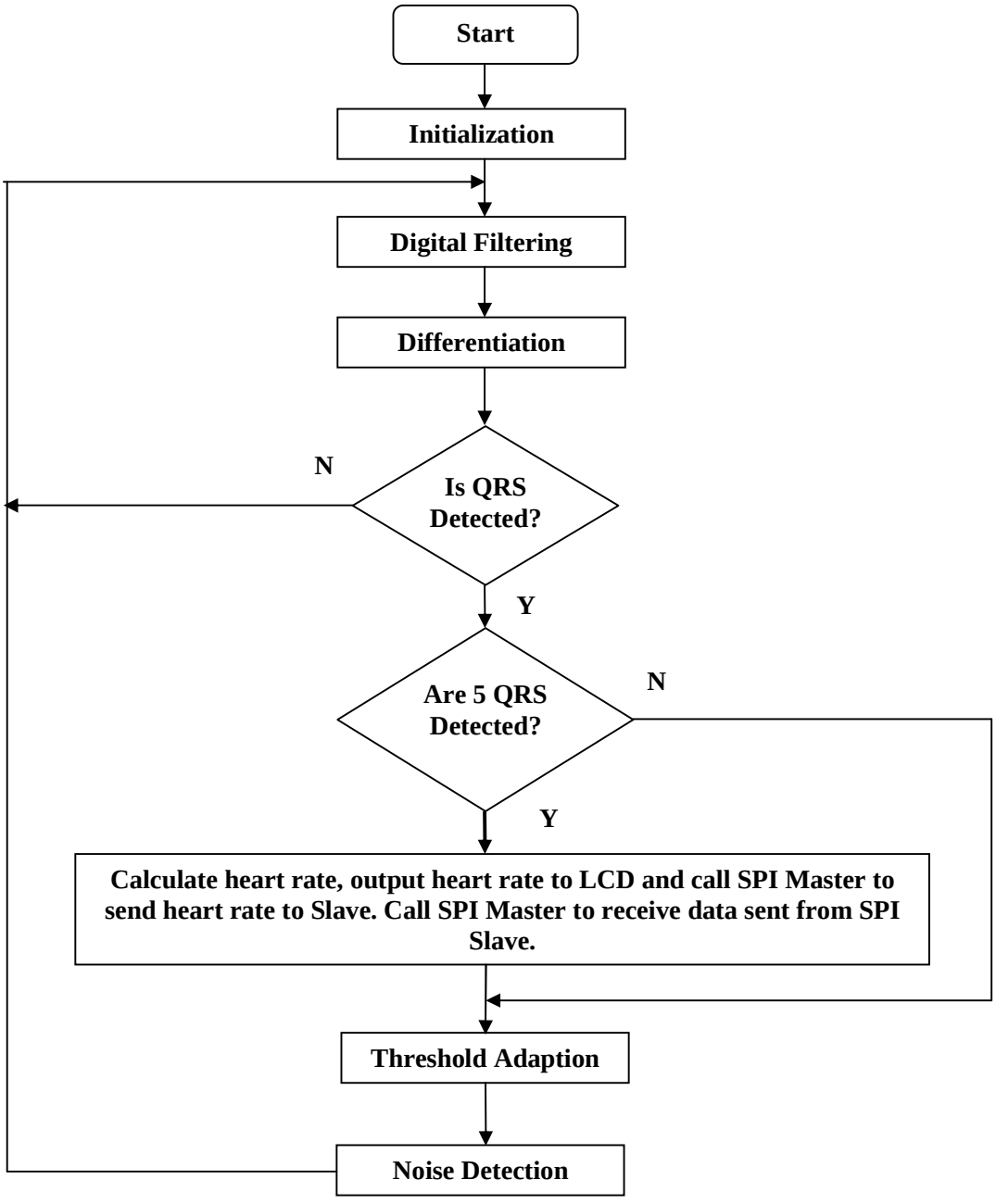


Fig. 4.1. Flow Chart of ECG signal processing.

After simulator is switched on, this block starts detecting ECG signal. This ECG signal is fed to ADC through PGA. After signal amplification ADC converts analog ECG signal into its equivalent digital discrete values. This is done because PSoC processes digital data and transmits it. Figure below shows ECG signal before digital filtering.

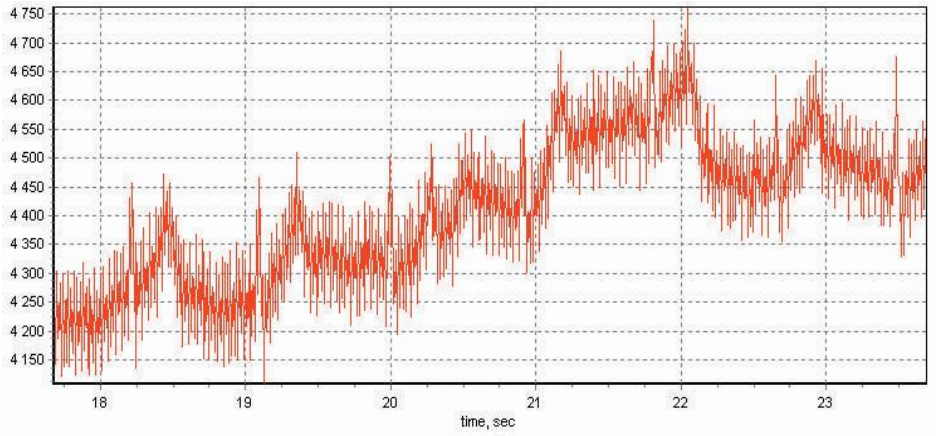


Fig. 4.2 (a). ECG signal before digital filtering.

ECG signal gets corrupted from sources like electrical interference from surrounding equipment, measurement (or electrode contact) noise, electromyogram noise (muscle contraction), movement artifacts, instrument noise (such as artifacts from the ADC conversion), baseline drift and respiratory artifacts. Thus ECG signal has to be filtered using digital filters.

Digitized signal from ADC passes through three cascaded second order IIR lowpass filters. Main purpose of using these filters is to remove power-line 50 and 60 Hz interference suppression. Figure below shows ECG signal after digital filtering.

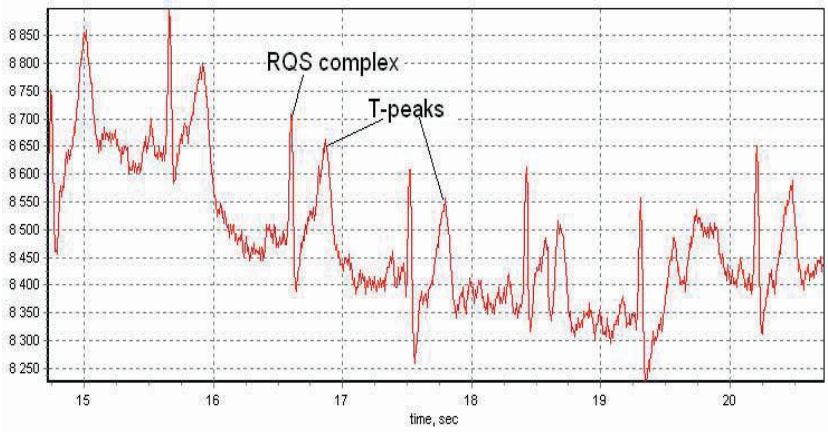


Fig. 4.2 (b). ECG signal after digital filtering.

Filtered ECG signal passes through the differentiator, where the QRS complex is picked out. Differentiator consists of first order IIR high-pass filter. Peak R wave is called as pulse beat. Pulse beats are detected by a smart peak detector with a threshold level that is automatically adjusted to increase the noise resistance. 5 such QRS complexes are detected and stored. Mean R-R pulse interval is chosen from these stored 5 complexes. Heart rate is calculated by taking reciprocal of this mean pulse interval value over a fixed period of time of 60 sec and then scaling to units of beats per minute (BPM). This calculated heart rate is made pass through averaging filter. Filter calculates average of last two detected heart rates. This is done to improve accuracy of the system. Finally ECG signal in terms of BPM is outputted on LCD. SPI Master is called, which sends the heart rate to the transmitter on MOSI line. SPI Master also receives the data which transmitter has received. This received data is also displayed on LCD. Purpose of receiving this heart rate back and displaying it is to check correctness of communication. Figure below shows differentiated ECG signal.

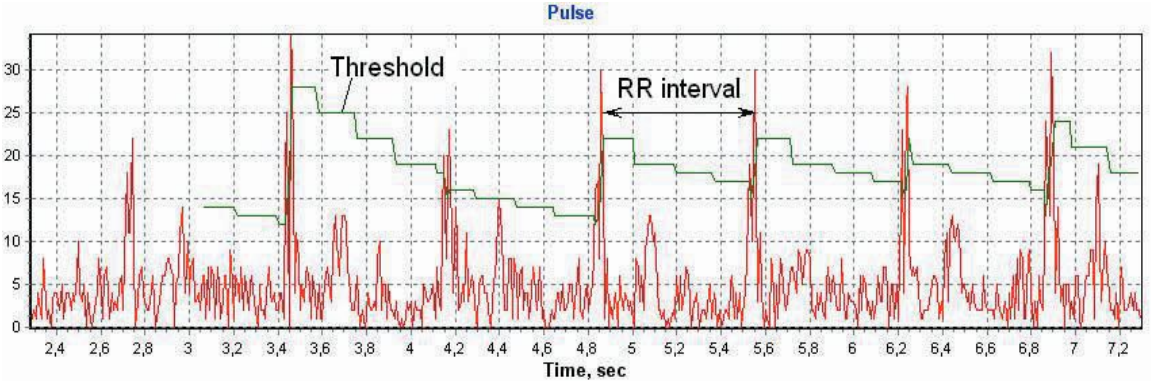


Fig. 4.2 (c). Differentiated ECG signal.

To reduce the detection of false beats, after every detected QRS complex, automatic threshold adaptive algorithm is run. This algorithm performs the automatic threshold level adjustment and noise detection in 240 ms as shown in figure above. Automatic threshold adaptive algorithm is discussed in the next section.

4.2 RF Transmitter

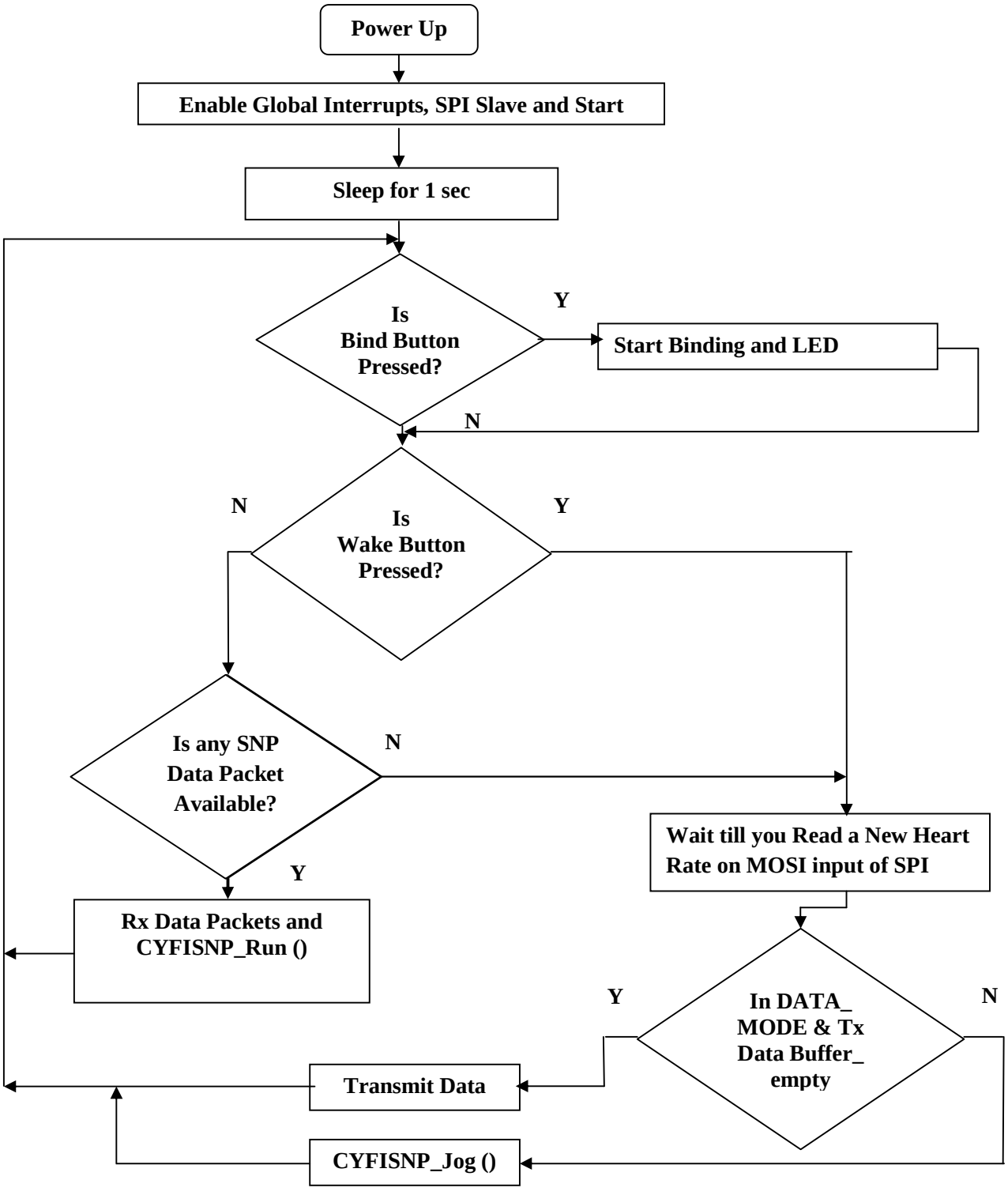


Fig. 4.3. Flow Chart of RF Transmitter

The firmware of RF Transmitter can be broken into the following parts:

- Power up initialization and setup
- Heart Rate Reading
- Main loop of the application performing general tasks

Power up initialization and setup: Program begins with enabling global interrupts. This should be called before CYFISNP protocol is invoked. CYFISNP_Start() starts and initializes the radio. Disable the CyFi Radio power management unit temporarily which saves around 32 uA power. CYFISNP protocol implements sleep timer. API CYFISNP_TimeSet starts and enables timers. These timers are dependent on sleep timer counts and sleep timer of one second is set here.

Heart Rate Reading and main loop application: Before we read the heart rate and transmit data, transmitter should be bound to receiver hub. Thus, first check for bind button press. If bind button is pressed, configure the node. Binding steps are explained in the later section. Scan the bind button press every cycle through the main application loop to detect a bind button request from the user.

There are two ways of reading of heart rate. First, by pressing wake button we can initiate heart rate reading manually. Second, application loop will initiate heart rate reading. If wake button is not pressed, then check whether any data is sent by receiver hub. Receiver can send some data back to transmitter on back channel to get information on configuration, bind or can update report timer. Report timer is not used in this project. If data packets are available then Call the CYFISNP_Run() API. This exercises the protocol and calls for Bind, connect, ping, and data modes. This helps in maintaining the RF link between the nodes and the hub. If there is no data available from receiver hub, then read heart rate.

SS line is checked repeatedly, till it is driven low by SPI master. Heart rate reading begins once SS line is driven low. SPI slave receives the buffered data serially, bit by bit and stores it in receive data buffer. This is then read and transmitted by transceiver. Also received data is sent back to heart rate detection system to check consistency.

To transmit data to receiver hub, it must first check whether transmitter is in “DATA_MODE” and transmit buffer is empty. If both are true, then load transmit data and transmit data by calling API CYFISNP_TxDataPut(&txApiPkt). Else call CYFISNP_Jog API to rouse the radio of any connect or ping mode timeouts.

4.3 RF Receiver (Hub)

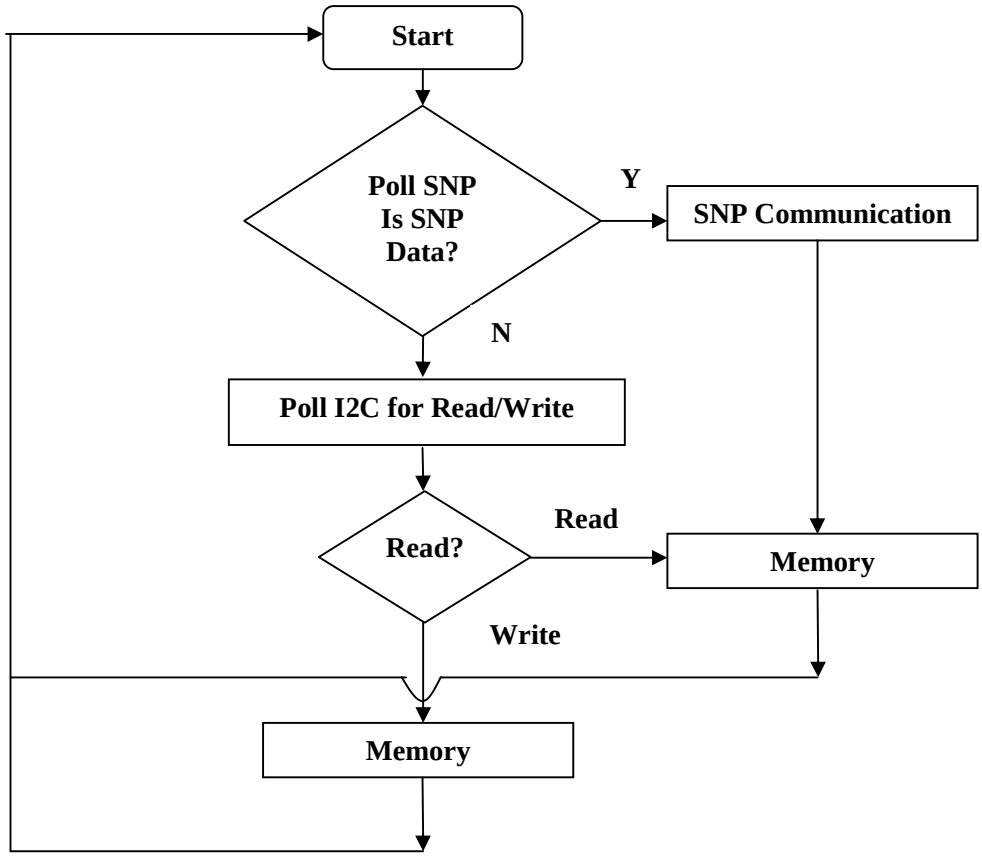


Fig. 4.4. Flow Chart of RF Receiver.

RF receiver Hub application consists of following three blocks,

- Communication with SNP
- Memory and Buffer Management
- I2C Application Interface Management

Communication with Star Network Protocol (SNP): This begins with Polling SNP at regular intervals. API ServeSNPPackets() is used for this purpose, which checks whether there is a need for servicing data. Keep polling the SNP at regular intervals to check if it contains data that needs to be serviced. It first polls the SNP for data CYFISNP API calls CYFISNP_RxDataPend to check whether previous data is pending, if yes then take necessary action. If the packet is of current data type, then it is written into an Application Buffer and the Memory management section is called which handles the packets in the buffer. If a data

packet is successfully written in the Application Buffer, then the Write pointer (pNextWrite) is incremented to point to the next location in the application buffer. Count value indicates number of packets in the buffer (numberOfPacketsInBuffer), this is incremented respectively. These two are implemented in an API called WriteBufferManager().

Memory and Buffer Management: Main functions of this block are to manage the data movement between the protocol data buffers, including the back channel data buffer and the Application firmware. Management algorithm prevents buffer overflows and prevents the I2C master bridge from receiving junk data when it polls the slave hub, when there are no packets in the buffer. This guarantees that data is read and written in an efficient manner. This saves latency, and also data is not overwritten or lost due to Buffer overflow error. The data buffer and the back channel data buffer are implemented as circular buffers. Size of data buffer = 252 bytes (18 bytes X 14). Back channel Data buffer size = 128 bytes (16 bytes X 8). API BufferManagement() is used to implement memory management in the hub firmware. This API is called immediately after the APIs WriteBufferManager() and ReadBufferManager(). This function checks for four important conditions. Takes actions based on the following conditions,

- Buffer Full
- Buffer Empty
- Read pointer has reached the last location in the buffer
- Write pointer has reached the last location in the buffer and the buffer is full or not

I2C Application Interface Management: The I2C is implemented using the EzI2Cs user module. I2C communication between the master CY8C24894 and the slave CY8C24894 is initiated by API CheckHostRequest(). After this API is called, the master CY8C24894 writes a command into the RAM buffer of the slave. Functions such as reading out data from the application buffer and freeing up buffer space, sending back channel data to a particular node, getting the node configuration of a specific node are implemented in various commands. Following switch case statements are used to check entered command.

The case statements are:

FETCH_NXT_PKT

- SEND_PKT
- GET_NODE_CFG
- UNBIND_NODE
- ENTER_BIND_MODE
- GET_PROTOCOL_STATE
- GET_LAST_BIND_RESULT

Hex values 0x01 to 0x07 assigned for every case and these are defined in the application header file FTRFHub.h. Here FETCH_NXT_PKT is executed only when a data packet is read by the I2C master and sent up to the host GUI. This occurs only when the application buffer is not empty. Then numberOfPacketsInBuffer is immediately decremented and BufferManagement() is called through the API ReadBufferManager(). All above mentioned master I2C requests have a response from the slave. These responses also have hex values 0x81 to 0x87 assigned to them. These are defined in the application header file FTRFHub.h.

Apart from performing above tasks, the hub application firmware also performs the following tasks to ensure proper functioning of the hub. They are,

- The bind button is scanned continuously to detect a button bind request from the user. This is Implemented by the API checkBindButton().
- A test back channel data packet is sent if a device requests a back channel data packet. This is done by setting BCDR bit when the host GUI has no data to send. This is implemented by the API SendBackChannelData(BYTE devID)

This section explained detailed work flow of overall system. Next section discusses thresholding algorithm used for accurate ECG detection.

Chapter 5:

ALGORITHM

5.1 Threshold adaptive algorithm

To reduce the detection of false beats, after every detected QRS complex, the algorithm performs the automatic threshold level adjustment and noise detection in 240 ms.

Step1-Find maximum absolute value in an interval of t1 = 200 ms after the last detected complex.

Step2 –Update threshold value with 0.75 × max [E], where E is the differentiated ECG.

Step3- Decrease threshold value till it reaches predefined value.

Step4- During interval between 200ms to 240ms following last QRS detection, if E becomes greater than threshold value then goto step 5. Else goto Step1.

Step5- Display “noise” is detected. Increase threshold value, recount current R-R interval. Goto Step1.

This section explained necessity of threshold adaptive algorithm to accurately detect ECG signal, detect and correct errors. Next section deals with hardware, software requirements to develop and run this application.

Chapter 6:

SYSTEM REQUIREMENTS

Tools	Purpose	Requirements	
		Type	**Minimum Requirement**
PSoC Designer 5.0	Integrated Development Environment	Processor Speed	1 GHz
		RAM	1 GHz
		O.S	Windows® XP (SP2 or higher)
Programmer	Programming	PSoC Programmer	Version 3.1
Cypress Sense and Control Dashboard	Graphic User Interface	CY3271 FTRF	Version 2.0

Table.6.1 showing system requirements.

This sectioned revealed details of various hardware and software requirements. Next section deals with implementation details of various blocks used in this design.

Chapter 7:

IMPLEMENTATION

7.1 PSoC Hardware platform

Cypress Semiconductor's family of PSoC microcontrollers are designed to replace traditional microcontroller-based system components with one low-cost programmable device. PSoC architecture has a central processor, data SRAM, flash program memory, configurable analog and digital blocks, programmable I/O ports, clock generator, and some other system resources. The M8C microprocessor follows 8-bit Harvard architecture and can operate on speeds of up to 24 MHz. The general purpose I/O pins (GPIO) allow flexibility to interface with external devices. Each pin's drive mode can select to predefined settings and interrupt option generates an interrupt in case of predefined events on a pin.

PSoC CY8C27443 with 28 pin PDIP package is used in our design. It contains 12 Rail-to-Rail Analog PSoC Blocks with 14-Bit ADCs, 9-Bit DACs, PGA's, Programmable Filters and Comparators. It also contains 8 Digital PSoC Blocks with 8 to 32 Bit Timers, Counters, and PWMs and Full-Duplex UARTs. It has inbuilt 256 bytes of RAM and 16 KB of program memory. Works on supply voltage ranging from 3.0 V to 5.25 V.

7.2 PSoC Software platform

The PSoC designer integrated development environment (IDE) is developed by Cypress Microsystems. PSoC Designer is the revolutionary IDE that can be used to customize PSoC to meet specific application requirements. PSoC Designer software accelerates system time to market. Applications can be developed using a library of pre characterized analog and digital peripherals in a drag and drop design style. Customized design leverage the dynamically generated API libraries of code. Finally we can debug and test developed designs with the integrated debug environment, including in-circuit emulation and standard software debug features.

The PSoC application code is developed in three main stages. They are Device Editor, Application editor and Debugger.

- Device editor (DE): allows the user to choose the functional component (e.g. ADC, filter, timer, counter, etc.) for each of the analog/digital blocks.

- Application Editor (AE): allows the user to write the code either in C language or assembly language. The analog/digital blocks must be initialized in the code using the generated component libraries. It also gives access to the interrupt service routine.

- Debugger: Debugging is the last step in the development process of PSoC. PSoC designer IDE provides an in-circuit-emulator, which allows debugging at the full operating speed of PSoC. It allows the user to define complex breakpoint events and large trace buffers that allow the user to monitor registers, memory locations, etc.

7.3 ECG Detection and Processing

ECG Detection and Processing block is implemented using PSoC CY8C27443. This microcontroller is configured using the device editor in PSoC Designer 5.0. In this system PGA, Analog to Digital Converter, SPI Master and digital buffer are implemented as software analog and digital blocks. Low Pass Filter and Differentiator are implemented in the program. Usage of each block is explained in brief below,

- PGA: This user module buffers ECG data received from simulator.

- ADC: PSoC processes digital data. Thus, analog ECG data is converted into its equivalent digital value using this user module.

- SPI Master: Heart rate has to be sent to transmitter module, to transmit data to the base station. Serial communication with transmitter is established using this user module.

- Digital Buffer: Digital data is sent on MOSI line of SPI master. This user module drives data on serial line and sends it bit by bit in synchronization with clock signal.

- LCD: This user module drives LCD. Various messages and Heart rate is displayed on LCD using this module.

7.3.1 Internal Schematic

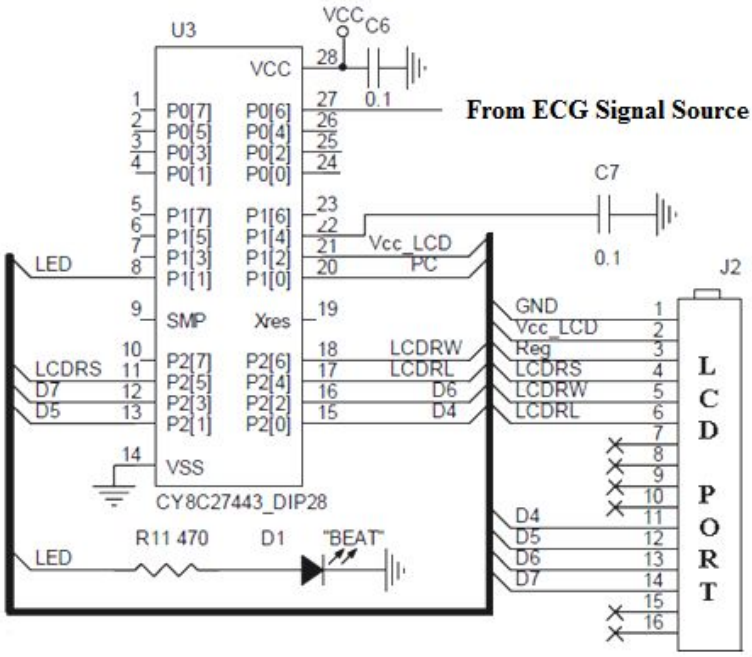


Fig. 7.1. Internal schematic.

Figure above shows internal schematic of ECG Detection and Processing. ECG signal from simulator passes to ADC through port P0[6]. ADC converts analog ECG signal into digital data readable by PSoC. Digital data obtained is made pass through three digital second order IIR low pass filters to suppress 50-60 Hz power line suppression. Filtered ECG signal is then passed through differentiator, which picks QRS complex. Differentiator consists of first order IIR high-pass filter. A smart peak detector with a threshold level that's automatically adjusted is used to increase the noise resistance that detects pulse beats. Heart rate is scaled to Beats Per Minute and is displayed on LCD. PGA, ADC and comparator are implemented in PSoC's analog blocks. SPI master, digital buffer and LCD are implemented in PSoC's digital blocks. These are explained in next section.

7.3.2 PSoC internal Schematic

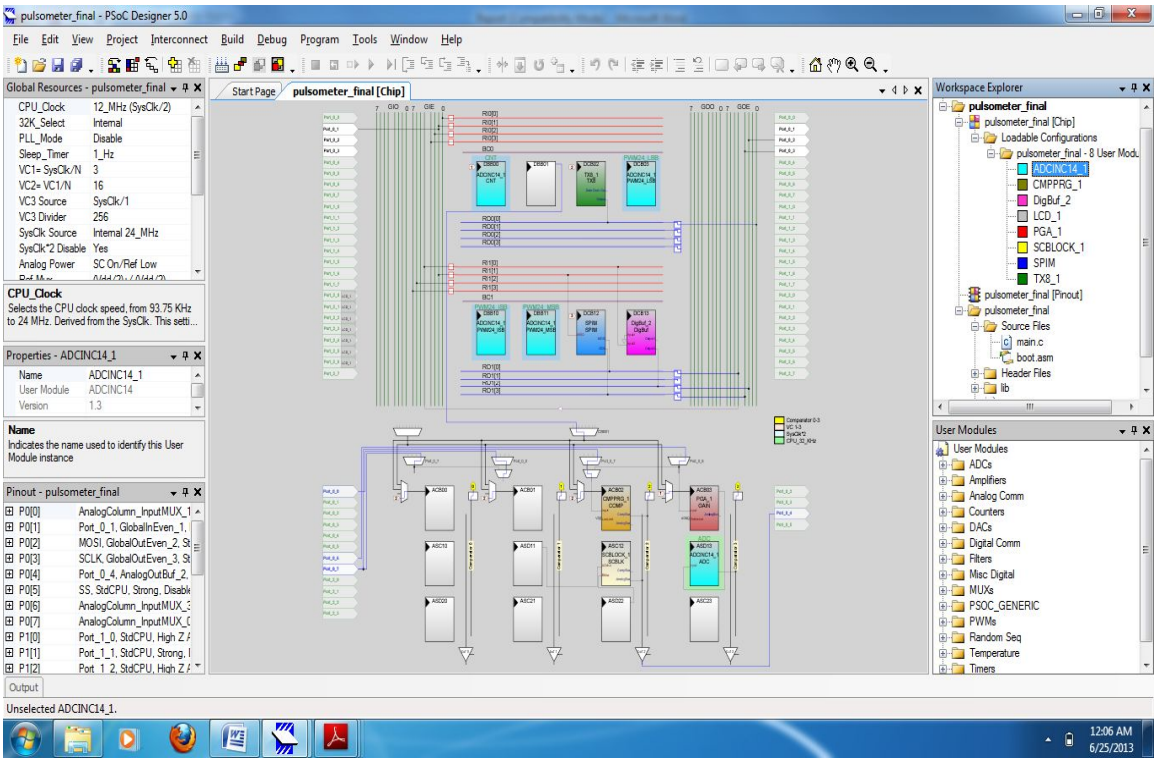


Fig. 7.2 . PSoC internal Schematic.

Global Resources are configured with following values.

CPU_Clock: Selects clock speed from 93.75 KHz to 24MHz. It is derived from SysCk.CPU_Clock selected here is SysClk/2 i.e 24/2=12MHz.

32K_Select: Selects internal low speed 32KHz clock speed for internal block operations.

PLL_Mode: Generates SynClock with greater clock accuracy. But can be unstable in noisy system if 32 KHz clock is not stable. PLL_Mode is disabled here.

Sleep_Timer: Selects time of sleep interrupt. Affects watchdog timer for every 3 sleep time cycles if enabled. Sleep timer of 1HZ is selected here.

VC1: It is virtual clock. It is derived from SysClk/N. Here N is selected as 3. Hence, VC1=24/3=8 MHz.

VC2: Virtual clock derived from VC1/N. Here N is selected as 16. Hence VC2=8/16=0.5 MHz.

VC3: Virtual clock source derived from VC1. Hence VC3=SysClk/1=24 MHz. VC3 is derived from VC3 Divided by VC3 Divider. VC3 Divider value =256. Hence VC3=VC3/256 =24/256= 93.75 KHz.

SysClkSource: Selects clock source from internal clock source or external port P1[4].
Ref Mux: Selects referential multiplexer value from various sources. Here value selected is Vdd/2+-Vdd/2.

AGNDBypass: It is enabled. The PSoC only handles signals of positive polarity with respect to VSS. An artificial ground is constructed on the chip to provide a reference point for signals of both polarities. This reference is called Analog Ground.

Buffer Amplifier (PGA):

Features:

- Thirty-one user-programmable gain settings with a maximum gain of 16.0.
- All other PSoC Devices: thirty-three user-programmable gain settings with a maximum gain of 48.0.
- High impedance input.
- Single-ended output with selectable reference.

ECG signal from simulator is fed to buffer amplifier via port P0[6]. A Programmable Gain Amplifier is used as buffer amplifier for this purpose. The PGA User Module implements an Op-Amp based non-inverting amplifier with user-programmable gain. This amplifier has high input impedance, wide bandwidth, and selectable reference.
The PGA User Module amplifies an internal or externally-applied signal. This signal can be referenced to the internal analog ground, Vss or other selected references.

The gain, of the programmable gain amplifier, is set by programming the selectable tap in a resistor array and the feedback tap in a continuous time analog PSoC block. The gain, input, reference, and output bus enable are set by the user from tables of values in the Device Editor. For gains greater than or equal to one, the top of the resistor string is connected to the Op-Amp output and the resistor tap is connected to the inverting input of the Op-Amp.

PGA_1 Properties:

Gain: Required gain is achieved by choosing appropriate values of R_a and R_b in $(1+R_b/R_a)$. Gain selected here is 48.

Input: From simulator ECG signal is fed to port P0[6]. AnalogColoumn_InputMUX3 is connected to port P0[6]. Filtered signal is amplified in this PGA block.

Reference: The gain of the PGA is referenced to a "ground" value selected by the user. Choices include AGND (on-chip analog ground), Vss, and adjacent continuous time (CT) and switched capacitor (SC) blocks. CT and SC block connections allow for adjustable offsets as a controlled reference voltage. Reference of AGND is chosen here.

AnalogBus: The gain block output may be routed through the analog PSoC block array's network of local interconnections and/or through an analog output bus. AnalogBus is disabled here.

Analog to Digital Conversion (ADC):

Features:

- 14-bit resolution.
- Sample rate from 2 to 120 samples per second (sps).
- Input range from Vss to Vdd.
- Integrating converter provides good normal mode rejection.
- Internal or external clock.

The ADCINC14 is an integrating ADC with 14 bits of resolution. It can be configured to remove unwanted high frequencies by optimizing the integrate time. Input voltage ranges, including rail-to-rail, may be measured by configuring the proper reference voltage and analog ground. The result format is selectable between signed or unsigned, based on an input voltage between Vref and +Vref centered at AGND. Sample rates from 2 to over 120 sps are achievable, depending on the selection of the DataClock and CalcTime parameters. The programming interface allows the user to specify the number of sequential samples to be taken or to select continuous sampling. The CPU load varies with the input level.

To make the integrator function as an incremental ADC, the following digital resources are used:

- An 8 bit counter to accumulate the number of cycles that the output is positive.
- A 17 bit PWM to measure the integrate time and gate the clock into the 8-bit counter.

A single DataClock is connected to the 8-bit counter, the 17 bit PWM (implemented from a 24 bit PWM), and the analog column clock which connects to the analog SC PSoC block. The analog column clock is actually two clocks φ1 and φ2, which are generated from the data clock. These two additional clocks are exactly one-fourth the frequency of the data clock. This means that the PWM and counter operate 4 times faster than required and therefore, need to accumulate 16 bits worth of data.

The counter is implemented with an 8 bit digital block for the LSB and a software counter for the MSB. Each time the hardware counter overflows, an interrupt is generated and the upper MSB of the counter is incremented. This allows the ADCINC14 module to be implemented with only four digital blocks instead of five.

The 24 bit PWM is programmed to output a high signal that is 216 times the DataClock. The PWM output will be low for the time it takes to do the minimum result calculations and to reset the integrator. This low time can also be adjusted to help provide a more exact sample rate in combination with the DataClock. The total period of the PWM is the sum of the integrate time and the CalcTime. Sample rate is calculated using following formula,

$$\text{SampleRate} = \frac{\text{DataClock}}{2^{\text{Bits}+2}+\text{CalcTime}}$$

When the first reading is initiated, the PWM configuration is calculated, the integrator is reset, and the counter is reset to FFh. The initial delay will always be at least that of the calculation time. The PWM is initialized only prior to the first reading. After the compare and period registers are set once, they do not have to be re-initialized. When the PWM count is less than or equal to the integrate value, the output goes high, enabling the 8-bit counter to count down. The output of the PWM stays high until the counter reaches zero. At this point, the clock to the 8 bit counter is disabled and the PWM interrupt is generated. The initial value of this 8 bit software counter is set to 214/64 times the most negative value. Each time the 8 bit counter overflows, the interrupt for the 8 bit counter is executed and the software counter (MSB) is incremented by one.

Counter will increment on every positive transition of the DataClock when the input to the ADC is greater than or equal to the most positive value, the 8 bit. If the input to the ADC is less than or equal to the most negative input value, the 8 bit counter will never decrement and therefore, never generate an interrupt. An input near analog ground under ideal conditions allows the counter to increment for its half of the time. So it is easy to see that, depending on the input voltage level, the amount of interrupts from the 8 bit counter will vary from 0 to (216)/256. This means that it is possible that the processor could be interrupted a maximum of 65536/256 or 256 times during the integrate period. With the sample time taking so long for a higher resolution result, it is unreasonable to expect the processor to wait while a sample is being processed. Primary communication between the ADC routine and the main program is a flag. It may be polled. When the LSB of ADCINC14_bfStatus has a non zero value, the new data becomes available in ADCINC14_iResult. API’s are available to check the data flag and retrieve data.

This data handler was designed to be poll based. Interrupt based data handler if it is desired, the user may insert own data handler code into the interrupt routine ADCINC14_CNT_ISR, located in the assembly file ADCINC14INT.asm. The point to best insert code is clearly marked.

ADCINC14_1 Properties:
Input: Input to ADC is fed from buffer amplifier (ACB03).

ClockPhase: This is used to synchronize the output of one switched capacitor analog PSoC block to the input of another. Here swap setting is selected. Swap setting allows the phases to be swapped so that the input signal is now acquired during φ2

Clock: Clock source is from VC1. Clock frequency is, VC1=24/3=8 MHz.

CalcTime: The CalcTime is the amount of time it takes the CPU to calculate intermediate integration result before the next integrate cycle can start. It is given by,

$$\text{CalcTime} >= \frac{\text{DataClock} * 200}{\text{CPU Clock}}$$

With DataClock=8MHz and CPU Clock=12MHz, we get CalcTime=133. Hence, CalcTime used here is 140 data clocks.

DataFormat: This selection determines in what format the result will be returned. If “Signed” is selected, the data range will be between -8182 and 8191. When the unsigned format is selected, the data range is between 0 and 16,383.Finally converted digital signal is fed to Low Pass Filter for further processing. Low pass filters, differentiator and peak detector are implemented in the program.

SPI Master (SPIM):

Features:

- Serial Peripheral Interconnect (SPI) Master protocol.
- SPI clocking modes 0, 1, 2 and 3.

- Input sources are selectable for clock and MISO.
- Output sources are selectable routing for MOSI and SCLK.
- Programmable interrupt on SPI done condition.
- SPI Slave devices can be independently selected.

SPIM Properties:

Clock: SPIM is clocked by one of 16 possible sources. Global I/O busses may be used to connect the clock input to an external pin or a clock function generated by a different PSoC block. The 48 MHz clock, the CPU_32 kHz clock, one of the divided clocks 24V1 or 24V2 or another PSoC block output can be specified as the SPIM clock input. The clock rate must be set to two times the desired bit rate. One data bit is transmitted and/or received for every two input clocks. SPIM clock derived here is VC3= 93.75 KHz.

MISO: The Master-In-Slave-Out input signal can be routed to one of 16 possible input sources. Allowable MISO sources include the global I/O buses, high, low, analog comparator buses, or another PSoC block can be specified to supply the MISO input. Thus, MISO input is derived from Row_1_Input_1.

MOSI: The Master-Out-Slave-In output of the SPIM can be routed to one of the global output buses. The global output bus can then be connected to an external pin or to another PSoC block for further processing. Thus, MOSI input is derived from Row_1_Output_0.

SCLK: This output clock is generated by the SPI Master. It is normally routed out through one of the global output lines to a port pin, then connected to a SPI slave. This clock defines the effective bit transfer rate. Thus, SCLK is derived from Row_1_Output_3.

Interrupt Mode: This option determines when an interrupt will be generated for the TX block. TxRegEmpty option is selected here. The "TxRegEmpty" option causes an interrupt to be generated as soon as the data has been transferred from the Data register to the Shift register. Choosing the other option, "TxComplete" delays the interrupt until the last bit is shifted out of the Shift register. This second option is useful in that it is important to know when the character has been completely sent. The first option, "TxRegEmpty" is best used to maximize the output of the transmitter. This allows a byte to be loaded while the previous byte is being sent.

ClockSync: In the PSoC devices, digital blocks may provide clock sources in addition to the system clocks. Digital clock sources may be chained in ripple fashion. This introduces skew with respect to the system clocks. Thus to remove any skew and for proper operation of this block Sync To SysClk is selected.

Digital Buffers:

Features:

- Two Digital Buffers.
- Input1 can be inverted.
- Can be used to generate an interrupt on the rising edge of Output1.

DigBuf is a set of two digital buffers. It can be mapped onto any digital block. The API provides functions to enable or disable the Output1 interrupt capability.

DigiBuf_2 Properties:

Default Load Status: Determines the whether the block defaults to an enabled or disabled state at the time the User Module loads. If this parameter is set to Enable there is no need to call the Start function to enable the User Module. If it set to disable then the buffer functionality will not be turned on until the Start function is called. This status is enabled.

Input1: The number of sources for Input1 varies depending on the digital block the UM is placed. These sources include the Analog Comparator buses and the row input and output buses. Input 1 is not used here.
Input2: The number of sources for Input2 varies depending on the digital block the UM is placed. These sources include the 48 MHz oscillator output, lower frequency VC1, VC2 and VC3 divided down from the 24 MHz system clock and other PSoC blocks and external inputs routed through global inputs and outputs. Input is derived from Row_1_Input_0 line.

Input2 ClockSync: If the Output2 signal of the DigBuf User Modules is to be used as a input signal or a clock for other blocks within the PSoC it is recommended that Input2 is synchronized with one of the internal system clocks. Clock is synchronized with this input.

Output1: The output may be routed to one of four global output signals. This output is not used here.

Output2: The output may be routed to one of four global output signals. Thus buffered signal is outputted on Output line Row_1_Output_2 line.

InvertInput1: Inverts the value of Input1, if selected. Normal input is used here.

LCD:

Features:

- Hitachi HD44780 LCD display driver chip protocol.
- Uses seven I/O pins.
- Simple routines provided to print RAM or ROM strings.
- Simple routines provided to print numbers.
- Simple routines provided to display horizontal and vertical bar graphs.
- Uses only a single I/O port.

The LCD Tool Box User Module is a set of library routines that writes text strings and formatted numbers to a common two or four line LCD module. Character graphics feature of these LCD modules are supported by using vertical and horizontal bar graphs. This module was developed specifically for the Hitachi HD44780 two line by 16 character LCD display driver chip. But it will work for many other four line displays. This library uses the 4 bit interface mode to limit the number of I/O pins required.

LCD_1 Properties:

LCDPort: Selects which PSoC I/O port is used to interface to the LCD display module. Port 2 is used as LCD port here.
Bargraph: Selects whether the bargraph functions are enabled. If disabled, the bargraph code is not generated, saving ROM space. Bargraph is enebled here.

7.4 Filters implemented in firmware

Second order IIR Low Pass Filter

A cascade of three second-order IIR lowpass filters the ADC samples. In general, an IIR filter is represented by an equation in which the output signal at a given instant is obtained as a linear combination of I/O signal samples at earlier times. These filters are designed to suppress the 50- and 60-Hz interference. Each has the following transfer function with a cutoff frequency of 25 Hz:

$$H(z) = \frac{0.0625(1+Z^{-2})+0.125z^{-1}}{1-0.75z^{-1}+0.25z^{-2}}$$

Differentiator

The first-order IIR high-pass filter with a cutoff frequency (FC) of 70 Hz determines the first derivative of the ECG signal used for pulse rate calculations. The derivative picks out the QRS complex from the ECG signal.

$$H(z) = \frac{0.5(1-z^{-1})}{1-0.015625z^{-1}}$$

Peak Detector

A smart peak detector with automatic threshold adjustment is used to detect number of R to R peaks. Numbers of peaks detected are scaled down to display heart beats in Beats Per Minute (BPM).

7.5 RF Transmitter

The CY3271 FTRF Kit contains RF expansion board and RF hub. RF Expansion Board contains PSoC CY8C27443. This is the application controller. It controls the CyFi Radio and other components on the board. Since the RF expansion card has its own PSoC, it can be operated by removing it from the battery pack and inserting it into your target hardware or other development platforms.

The heart rate Sensor node design consists of the following modules:

- SPI Slave: Heart rate transmitted by the transmitter is received with the help of this user module.
- Digital Buffer: Received data is sent back to SPI master of ECG Detection and Processing system on MISO line of SPI slave. This is done to check the continuity in the communication. To drive this MISO line, data is buffered and sent back.
- CYFISNP: This user module implements and controls complete wireless link.

Node application PSoC contains the PSoC CY8C27443, which is the application Micro Controller Unit (MCU). It controls the radio transceiver CYRF7936 and sensors.
The RF Expansion board contains the onboard thermistor, which can be used to read temperature from thermistor. ISSP pins are used for programming PSoC. This board is provided with two LED's and two push button switches.
PSoC Node application PSoC has five General Purpose Input Output pins (GPIO). It also has two I2C pins Serial Data line (SDA) and serial clock (SCL). These pins, through RF expansion board can perform Input/output operation with external application board. Following four GPIO pins are connected with SPI master of ECG acquisition system. Heart rate is read and sent back with the help of these GPIO pins.

- GPIO 2 (pin no 16): Serial Clock (SCLK).
- GPIO 3 (pin no 14): Slave Select (~SS).
- GPIO 5 (pin no 15): Master Out Slave In (MOSI).
- GPIO 6 (pin no 13): Master In Slave Out (MISO).

This male expansion port is used to either program application PSoC or to connect to battery pack. When connected to battery pack, it acts as a independent node.

Female sensor receptacle is used to connect either multifunction expansion board or external input. Multifunction expansion board has ambient light detector, capsense and thermistor for temperature measurement. This board can be interfaced to read data from multifunction board. Female receptacles are internally connected to RF expansion port. Through this port external communication takes place. Internal schematic is as shown in the figure below.

Here application PSoC controls the transmission of data. Data to be transmitted is loaded into the txApiPkt buffer. By calling API this data is sent onto MOSI line of transceiver. Then data is sent out via RF transmitter. Detailed working is explained in the earlier sections.

7.5.1 PSoC internal Schematic

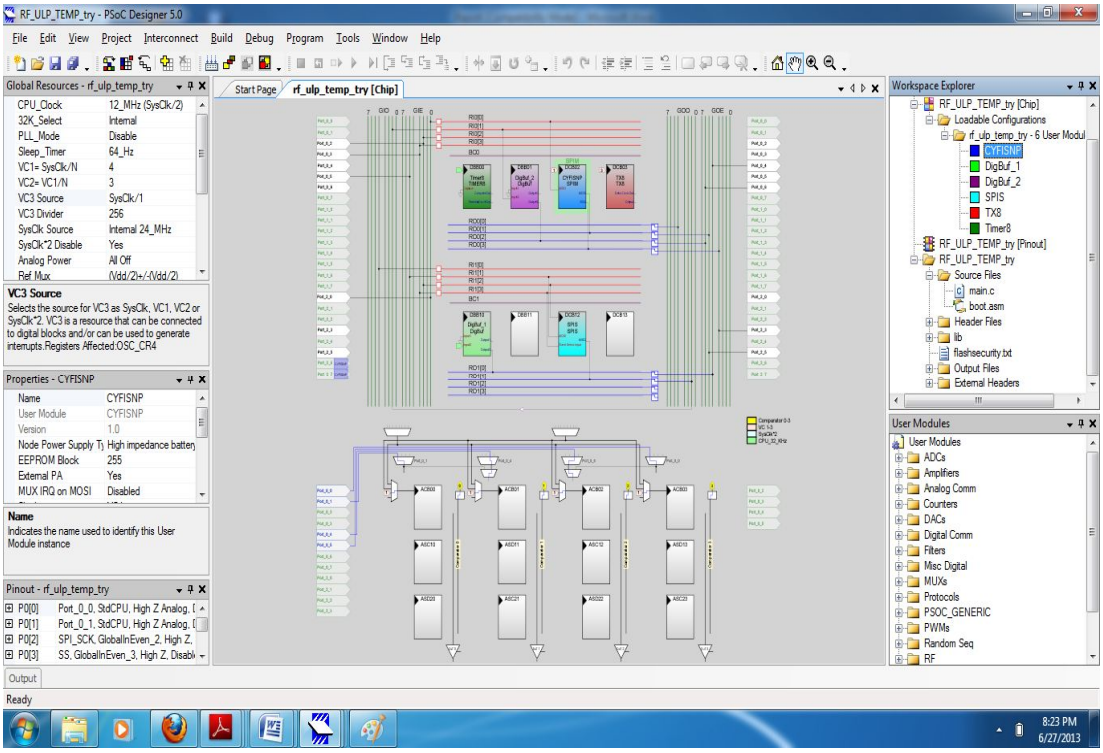


Fig. 7.3. RF transmitter PSoC schematic

SPI Slave Properties:

SCLK: SPIS is clocked by the SPI Master generated SCLK signal through Row_1_Input_2.

MOSI: The Master-Out-Slave-In input signal is routed from master on Row_1_Input_1 line.

~SS: The Slave Select input signal is routed from master on Row_1_Input_3 line.

MISO: The Master-In-Slave-Out output signal is routed to master on Row_1_Outpt_1 line.

Interrupt Mode: The "TxRegEmpty" option is selected here.

InvertMOSI: Is selected as normal.

Digital Buffers: Similar digital buffers discussed in ECG Detection and Processing system is used with following properties.

DigiBuf_2 Properties:

Default Load Status: This status is enabled here.

Input1: Input 1 is not used here.

Input2: Input is derived from Row_0_Input_1 line.

Input2 ClockSync: Clock is synchronized with this input.

Output1: This output is not used here.

Output2: Buffered signal is outputted on Output line Row_0_Output_2 line.

InvertInput1: Normal input is used here.

CYFISNP

Features:

- Protocol stacks implements hub or node functionality to support a wireless star network consisting of one hub and up to 250 nodes.
- It provides reliable two-way communication between a hub and node.

- Dynamic data rate (up to 1 Mbps) and output power according to the channel noise level and packet loss rate.
- Operates in the unlicensed worldwide Industrial, Scientific, and Medical (ISM) band (2.400-2.483GHz).
- With +20dBm output power, range translates over 400m of Line Of Sight (LOS).

CyFi Star Network Protocol Stack (CYFISNP) user module address up to 250 general purpose nodes. It provides reliable two way communication between the hub and nodes. The hub is assumed to be wall powered. Nodes may be either wall powered or powered by an alkaline (low impedance) or coin-cell (high impedance) battery.

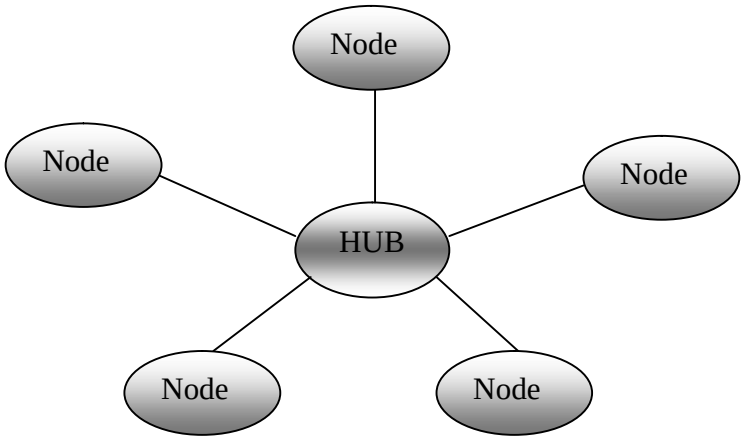


Fig. 7.4. SPIS Block diagram.

Definitions

Node: An MCU with a radio and other hardware that reports data wirelessly to a hub.

Hub: An MCU with a radio that receives data wirelessly from one or more nodes.

Manufacturing ID: Each CyFi radio modem contains a 4-byte manufacturing ID (MID) that has been laser fused into the radio silicon during manufacturing.

Addressing: Every node in a Star Network has an 8-bit device ID that ranges from 1 to 250 (Device ID0 is reserved). The Device ID uniquely identifies a particular node. It passes this to the application as part of the packet structure. When initially connecting, a node may either request that a Device ID be assigned by the hub (on the fly assignment), or the node may specify its own device ID (preassigned device ID). A single network can either support on the fly device ID assignment or preassigned device ID, but not a mix of both.

The device ID is passed up to the application (either in the MCU or software application) as part of the payload.

CYFISNP Properties

Node Power Supply Type: Low impedance alkaline battery has sufficiently low internal resistance to power the node continuously when needed. Node idles in sleep to conserve power.

EEPROM Block: This parameter defines the Flash block number used to store network parameters. Note that a Node device uses one Flash block and a Hub device can use several blocks (each available Device Id needs 8 bytes, so the number of blocks varies with the Maximum number of nodes parameter). These blocks must be marked as unprotected in the flashsecurity.txt file. Here node uses 255 flash block.

External PA: This parameter controls operation of external power amplifier. Choose Yes or No to indicate whether your project uses an external PA. External PA is used here.

MUX IRQ on MOSI: The CYRF793x device IRQ function may be multiplexed onto the MOSI pin. This is disabled here.

Clock: Selects the SPIM block clocking source. The clock rate must be set to two times the desired SPI bit rate. One data bit is transmitted and/or received for every two input clocks. Refer to CyFi radio data sheets for supported SPI bit rate ranges. Here clock source is chosen as VC1. VC1=SysClk/4=24MHz/4=6MHz.

MISO: The Master-In-Slave-Out input signal. This input line must be routed to the MISO pin on the radio. Signal is routed from Row_0_Input_0 line.

MOSI: The Master-Out-Slave-In output signal. This output line must be routed to the MOSI pin on the radio. Signal is routed to Row_0_Onput_1 line.

SClk: This output clock is generated by the SPI Master. This output line must be routed to the SCK pin on the radio. Signal is routed to Row_0_Onput_3 line.

IRQ_Port, IRQ_Pin: This pin selects the IRQ. This pin must be connected to the IRQ pin of the CYRF793x device when the MUX IRQ on the MOSI parameter is disabled. This pin must be connected to the MOSI pin of the CYRF793x device when the MUX IRQ on MOSI parameter is enabled. Thus for interrupt request Port_2 and pin Port_2_6 is selected.

nSS_Port, nSS_Pin: Selects the SlaveSelect pin. This output line must be connected to the SS pin on the radio. Slave select Port_2 and pin Port_2_7 is used.

First Channel: This parameter controls the low limit of the RF channel range used by the CYFISNP protocol to ensure regulatory compliance by preventing or reducing out-of-band RF emissions that may occur depending on RF circuitry design (ex. non-linearities in an external power amplifier may cause out-of-band RF emissions). Thus, channel 10 (2.412 GHz) is selected.

Last Channel: This parameter controls the high limit of the RF channel range used by the CYFISNP protocol to ensure regulatory compliance by preventing or reducing out-of-band RF emissions that may occur depending on RF circuitry design (ex. non-linearities in an external power amplifier may cause out-of-band RF emissions). Thus, channel 58 (2.460 GHz) is selected.

7.5.2. Data Transmission mode

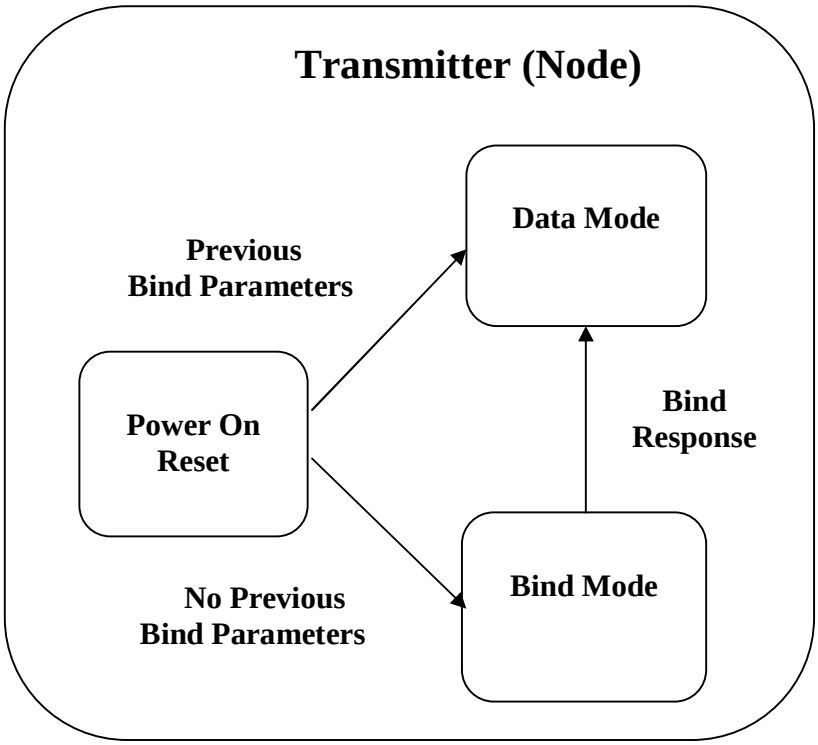


Fig. 7.5. Transmitter State diagram.

Node Power On Reset:

If the node has network parameters stored in its Flash memory, then the node moves to connect mode. If no network parameters are available, then the node waits in idle sleep mode until a user-initiated event causes the node to enter bind mode.

Node Bind Mode (Button Bind) :

The node assumes the bind network parameters and transmits hub bind requests. If an AutoACK is received, the node enables its receiver and listens for an immediate hub bind response. If a hub bind response is not received, the node moves to the next channel. If a hub bind response is received, the node stores the hub bind information in Flash and moves to connect mode. Upon delivering a bind response,the hub automatically exits bind mode.

Node Data Mode: When the node application has data to send to the hub the node transmits a DATA packet and listens for an AutoACK. If an AutoACK is not received, the node retransmits the packet. If the node does not receive an AutoACK after n DATA_PACKETS_RETRIES of retransmissions of the data packet it assumes the hub has changed channels and then enters node ping mode. If the nodes still fails to contact the hub, it assumes the channel has become unavailable due to excessive interference and goes to connect mode timeout. The packet to be transmitted is held until the node ping or connect is successful and then transmitted. Transmit data packets are never discard.

Refer Appendix B for steps involved in configuring node.

7.6 RF Receiver (Hub)

The PC Bridge consists of two CY8C28494 processors. One device is used for the Master Microprocessor that provides USB to I2C bridge functionality in addition to programming support for all kit elements. The second CY8C28494 processor acts as the wireless hub and communicates with the SCD SW application via an I2C interface to the master processor USB/I2C bridge. The Wireless Hub application use the CyFiSNP user module configured as a hub to communicate with the wireless nodes. All configuration and node data is communicated over a I2C interface.

The slave CY8C24894 is configured using the Device Editor in PSoC Designer 5.0. The bridge uses the CYFISNP, EzI2Cs, LED (Red and Green), and TX8 user modules. The usage of each user module in the application is described:

- CYFISNP: This user module implements the entire Star network wireless protocol and all protocol modes, in addition to low level radio communication and radio control by the MCU.
- EzI2C: This user module implements the I2C slave functionality and takes care of data communication through the I2C interface with the master CY8C24894.
- LED: There are two instances of this user module: one is configured as RED and the other as GREEN. These implement the API to turn ON/OFF the LEDs according to the needs of the application. The application firmware can just call simple APIs to manipulate the LEDs.

The PC Bridge consists of the Hub CY8C24894, the Master CY8C24894, and the CYRF7936 2.4 GHZ CyFi Transceiver. It contains a 16-pin connector to connect to the RF Expansion Board or the MultiFunction board, for application data exchange or ISSP programming. The FTPC Bridge is the interface bridge between the expansion cards, your PC, and the various applications.

Since the FTPC Bridge enumerates as a special type of 'composite device' that contains a PSoC Mini-Prog interface, the standard PSoC Programmer utility can identify and communicate with the FTPC bridge. This ensures that your FirstTouch RF Kit is automatically compatible with PSoC Designer.

Female expansion port is used to program RF expansion board (RF transmitter in our application). Hex file with the help of PSoC programmer can be downloaded onto expansion board.

Slave PSoC CY8C24894 controls the transceiver to receive the data. Received data is written onto the data buffer. Master reads the data through I2C interface and sends it GUI sense and control dashboard to output it on the screen. Detailed working is explained in earlier sections.

RF Hub program with default values is provided with the kit. Firmware is downloaded in this RF hub to make it capable to receive data and communicate with GUI.

7.6.1. Data Reception mode

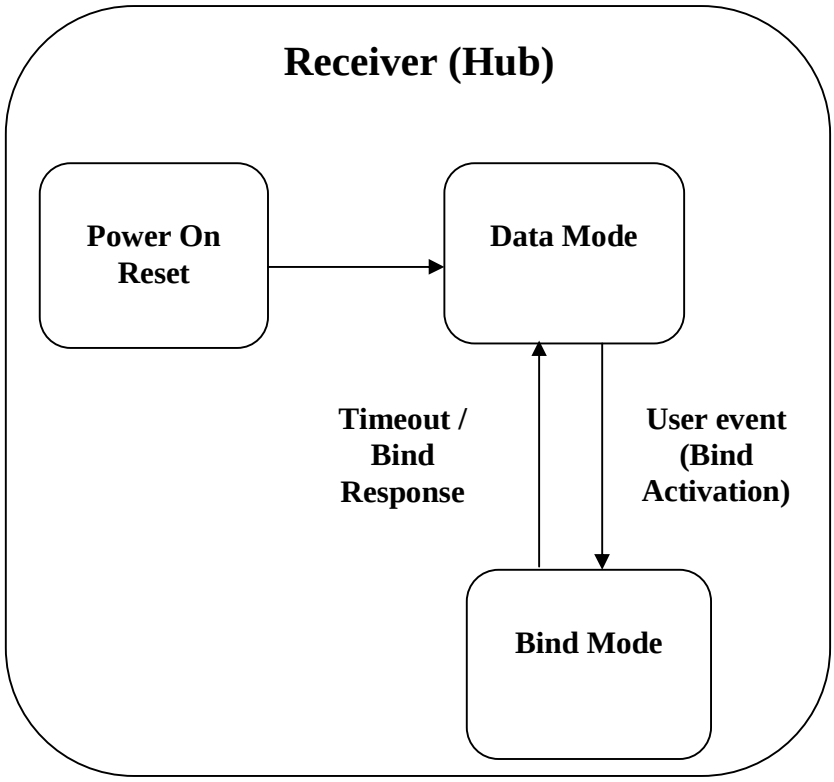


Fig.7.6 . Receiver State diagram.

Hub Data Mode

Data mode allows application data to be transmitted from a node to the hub. The hub continuously listens for data packets from nodes. When valid data is received from a node, the hub returns an AutoACK to the node and sends the data to the application. The hub

monitors the interference level and moves to ping mode if the RSSI interference threshold RSSI_NOISE_THRESHOLD is reached. This ensures that the hub is operating on a quiet channel and is capable of receiving node packets.

Hub Bind Mode

Hub bind mode is usually entered by a user event (button bind) and changes from the hub network parameters to the bind network parameters. The hub listens for a bind request on each channel for approximately 320 ms before selecting the next channel using the channel selection algorithm to mitigate channel interference. Normally, the first bind request received by the hub results in storing the node information in the hub's Flash. A subsequently received bind request (when the hub has the bind information stored in its Flash) will result in returning a bind response to the node and completing the bind process.

Unbind

An `unbind' mechanism allows the hub to unbind (and possibly replace) a previously bound node. At the hub, unbinding the node is done by zeroing the Flash EEP_DEV_REC entry for the particular device ID.

Sensor node is configured with the help of SCD GUI. Follow steps mentioned in Appendix B for binding hub with node. Other options like recording and alarm configuration is explained in this appendix section.

This section dealt with implementation details of analog, digital modules in ECG detection processing system, RF transmitter and RF hub. Next section presents with results.

Chapter 8

RESULTS

8.1 Heart Rate Classification

Sensor node detects three types of heart rates. They are Bradycardia, Normal and Tachycardia. This sensor node is connected to patient's body.

Bradycardia: If heart rate falls below 60 BPM, the condition will be called as "Bradycardia". This is demonstrated in following figures.

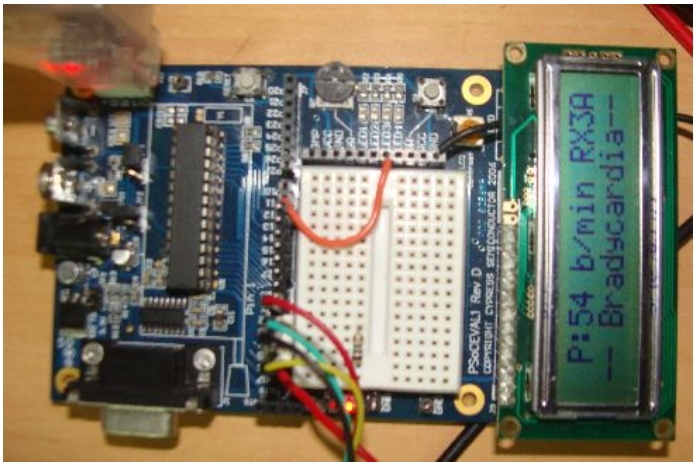


Figure.8.1(a). Bradycardia detected in ECG detection and processing system.

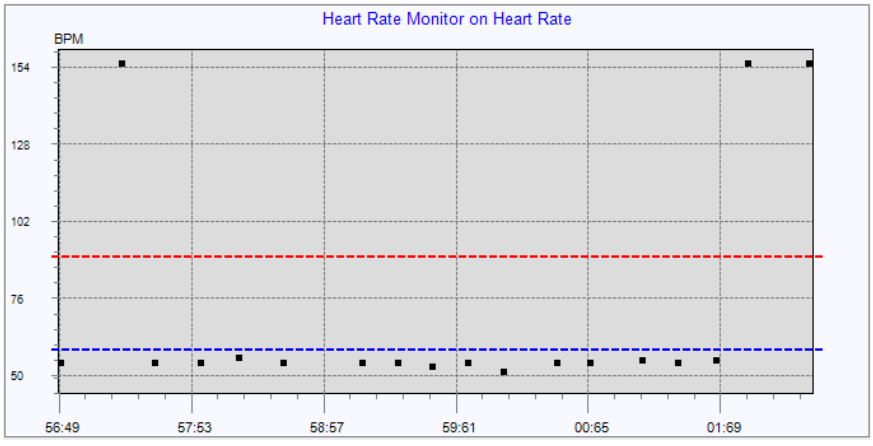


Figure.8.1 (b). SCD Graphical display of Bradycardia status.

Note: Sense and Control Dashboard displays graphical and textual data in decimal number format.

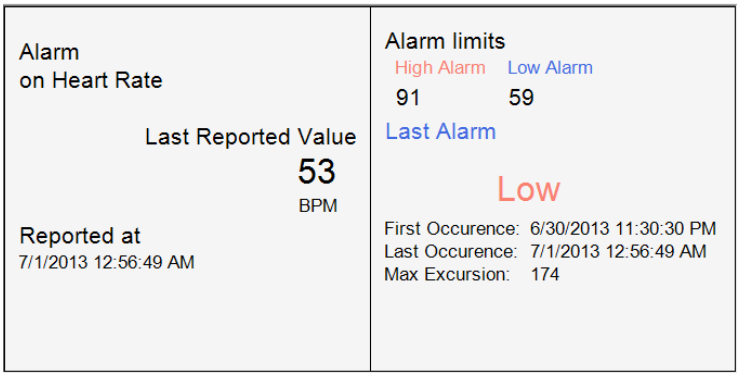


Figure.8.1 (c). Alarm indicating low heart rate (Bradycardia) on SCD.

Note: Console displays message received in Hexadecimal number format.

HubVendor ID	HubProductID	HubSerial No	NodeID	Resp	Rem	Len	RSSI	Time Stamp	Message In Hex
0x04B4	0xF115	8819DDC34222	01	00	00	01	13	01:09:10	34
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	01:09:33	35
0x04B4	0xF115	8819DDC34222	01	00	00	01	0D	01:09:49	35
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	01:10:32	36
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	01:10:48	36
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	01:11:07	33
0x04B4	0xF115	8819DDC34222	01	00	00	01	11	01:11:48	36
0x04B4	0xF115	8819DDC34222	01	00	00	01	10	01:12:11	34
0x04B4	0xF115	8819DDC34222	01	00	00	01	11	01:12:45	35
0x04B4	0xF115	8819DDC34222	01	00	00	01	11	01:13:03	36

Table.8.1. Low (Bradycardia) signal detection display on debug console of SCD

Normal Heart Rate: Normal heart rate ranges from 60 to 90 BPM. This is shown in following figures.

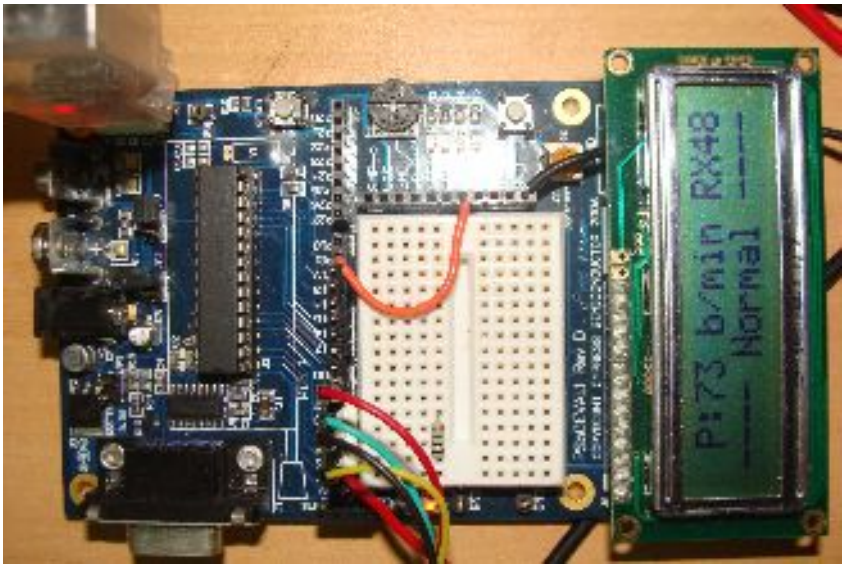


Figure.8.2 (a). Normal heart rate in ECG detection and processing system.

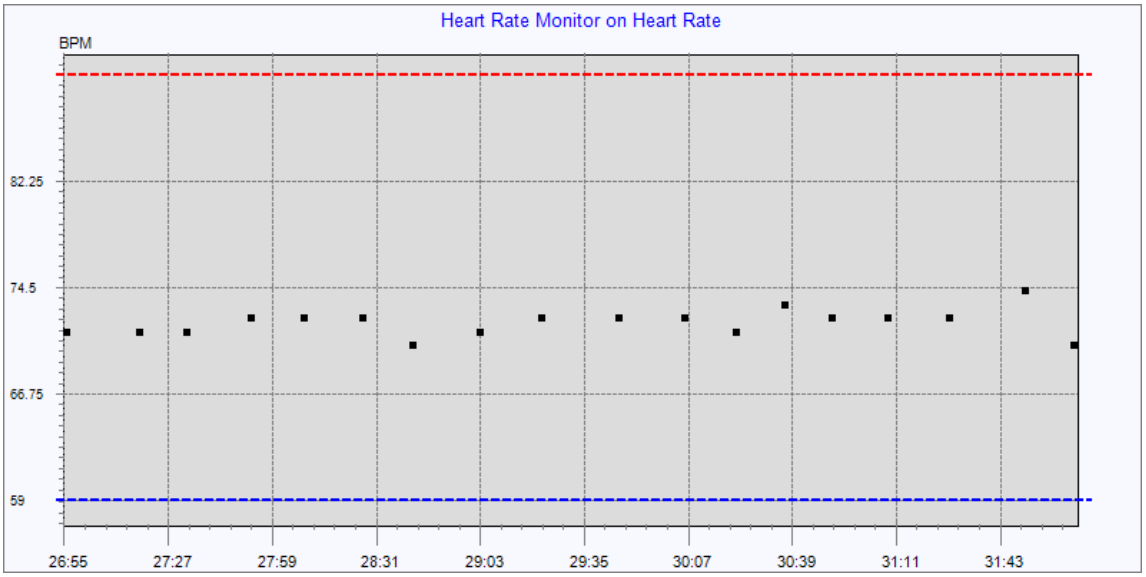


Figure.8.2 (b). SCD Graphical display of Normal heart rate.

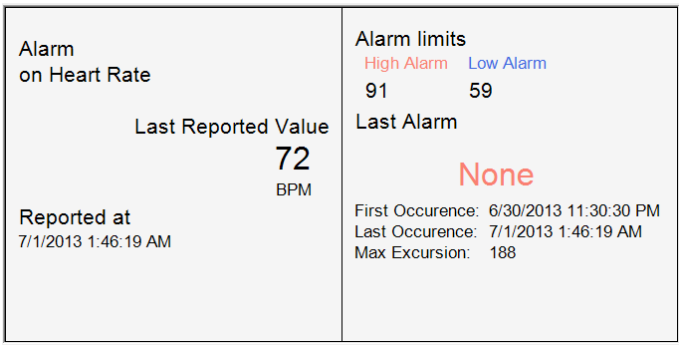


Figure.8.2 (c). Alarm indicating Normal heart rate on SCD.

Hub VendorID	Hub ProductID	HubSerialN	NodeID	Resp	Rem	Len	RSSI	Time Stamp	Message In Hex
0x04B4	0xF115	8819DDC34222	01	00	00	01	16	11:50:23	47
0x04B4	0xF115	8819DDC34222	01	00	00	01	15	11:50:51	47
0x04B4	0xF115	8819DDC34222	01	00	00	01	15	11:51:08	49
0x04B4	0xF115	8819DDC34222	01	00	00	01	15	11:51:25	48
0x04B4	0xF115	8819DDC34222	01	00	00	01	13	11:51:40	47
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	11:52:00	48
0x04B4	0xF115	8819DDC34222	01	00	00	01	13	11:52:15	46
0x04B4	0xF115	8819DDC34222	01	00	00	01	13	11:52:31	49
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	11:52:59	48

Table.8.2. Normal signal detection display on debug console of SCD.

Tachycardia: It is the stage heart beats at the rate above 90 BPM. This is shown with the help of following figures.

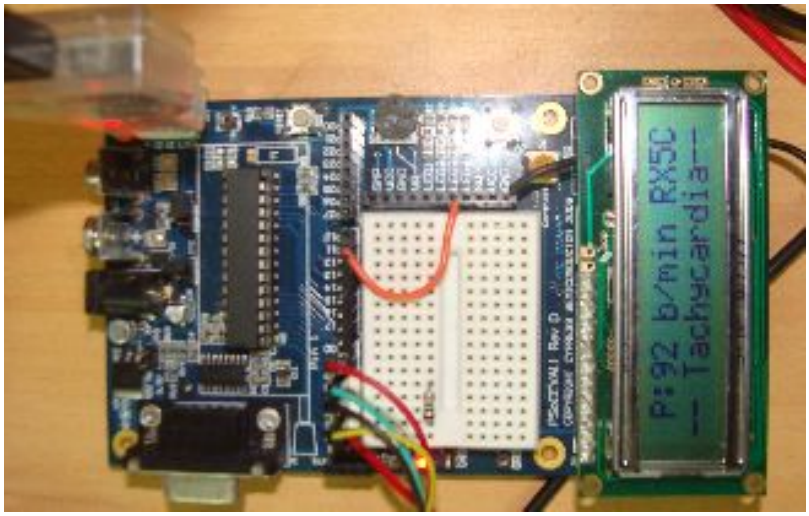


Figure.8.3 (a). Tachycardia in ECG detection and processing system.

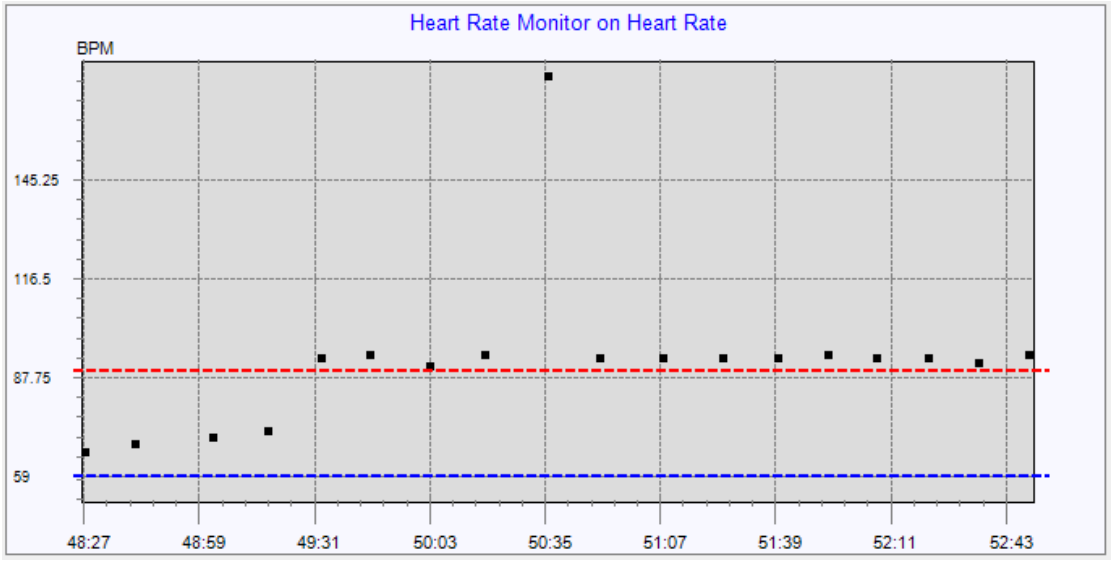


Figure.8.3 (b). SCD Graphical display of Tachycardia status.

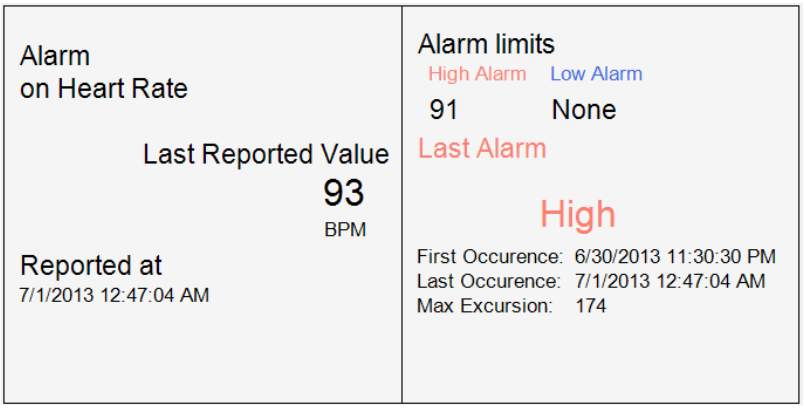


Figure.8.3 (c). Alarm indicating high heart rate (Tachycardia) on SCD

Hub VendorID	Hub ProductID	HubSerialN	NodeID	Resp	Rem	Len	RSSI	Time Stamp	Message In Hex
0x04B4	0xF115	8819DDC34222	01	00	00	01	11	12:42:13	5C
0x04B4	0xF115	8819DDC34222	01	00	00	01	13	12:42:27	5C
0x04B4	0xF115	8819DDC34222	01	00	00	01	13	12:42:41	5D
0x04B4	0xF115	8819DDC34222	01	00	00	01	10	12:42:55	5D
0x04B4	0xF115	8819DDC34222	01	00	00	01	0F	12:43:10	5D
0x04B4	0xF115	8819DDC34222	01	00	00	01	0C	12:43:24	5B
0x04B4	0xF115	8819DDC34222	01	00	00	01	0B	12:43:41	5C
0x04B4	0xF115	8819DDC34222	01	00	00	01	12	12:43:57	5D
0x04B4	0xF115	8819DDC34222	01	00	00	01	0C	12:44:12	5D
0x04B4	0xF115	8819DDC34222	01	00	00	01	09	12:44:48	5C

Table.8.3. High (Tachycardia) signal detection display on debug console of SCD.

8.2 CyFi sensor node Performance Analysis

CyFi CY3271 sensor nodes performance is compared with that of MICA 2 mote. Cypress Microsystems have designed a power calculator using simple excel sheet and is freely available for their users to calculate power. This helps developers to estimate the power consumption in the chip. Power consumption is calculated based on number of analog/digital peripheral blocks used, frequency of clock used for each of them, the number of pins and rows (on which the signals are routed) to be driven, etc. Table 8.4 show power calculator result of sensor node. Sensor node operates on 3.3 V and use 6MHz for its internal operations. CYFISNP operates on 6MHz. Provides clock to radio on Row_0_Output_3 line at 6MHz and consumes 0.095mA of current. Similarly this module sends data to radio module on Row_0_Output_1 line at 6MHz, consuming 0.095mA of current. Data at node is sent back to PSoC on Row_1_Output_1, at 93.75 KHz speed consuming 0.008mA of power. MOSI, SS and SCLK data from master are received on GPIO lines at 9.75 KHz, consume 0.035mA. Overall power consumption including reference circuit current is calculated as 3.452mA.

Estimated PSoC Power Calculations for CY8C27xxx Family Device											
Version 1.0									Enter parameters in yellow		
project name	CY 3271 Sensor Node								See estimated result in green		
part number	CY8C27443			**Cypress MicroSystems, Inc.**					Indicates something is wrong		
pin count	28										
								OpAmp Bias	0		
CPU Clock	12.000	MHz		SysClkx2Disable	1						
Vdd	3.30	V							Power		
								Analog Block	Level (0-3)		
CPU Current	2.764	mA						ACB00	0	0.000	mA
								ACB01	0	0.000	mA
								ACB02	0	0.000	mA
	Block Clk			GPIO	GPIO Clk			ACB03	0	0.000	mA
Digital Block	(MHz)			Count	(MHz)			ASC10	0	0.000	mA
DBB00	0.032	0.005	mA	3	0.094	0.034	mA	ASD11	0	0.000	mA
DBB01	0.000	0.000	mA	0	0.000	0.000	mA	ASC12	0	0.000	mA
DCB02	6.000	0.066	mA	0	0.000	0.000	mA	ASD13	0	0.000	mA
DCB03	0.000	0.000	mA	0	0.000	0.000	mA	ASD20	0	0.000	mA
DBB10	0.000	0.000	mA	0	0.000	0.000	mA	ASC21	0	0.000	mA
DBB11	0.000	0.000	mA	0	0.000	0.000	mA	ASD22	0	0.000	mA
DCB12	0.000	0.000	mA	0	0.000	0.000	mA	ASC23	0	0.000	mA
DCB13	0.000	0.000	mA	0	0.000	0.000	mA	Analog Block Current		0.000	mA
Digital Block Current		0.071	mA	GPIO Current		0.034	mA				
								A_Buffer Power	0		
	Row Clk			Analog Power				Analog			
Row	(MHz)			All Off	0			Output Buffer			
Row_0_Output_0	0.000	0.000	mA	SC Off/Ref Low	0	0.000	mA	AnalogOutBuf_0	0	0.000	mA
Row_0_Output_1	6.000	0.095	mA	SC Off/Ref Med	0	0.000	mA	AnalogOutBuf_1	0	0.000	mA
Row_0_Output_2	0.000	0.000	mA	SC Off/Ref High	0	0.000	mA	AnalogOutBuf_2	0	0.000	mA
Row_0_Output_3	6.000	0.095	mA	SC On/Ref Low	1	0.385	mA	AnalogOutBuf_3	0	0.000	mA
Row_1_Output_0	0.000	0.000	mA	SC On/Ref Med	0	0.000	mA	Analog Output Buffer Current		0.000	mA
Row_1_Output_1	0.094	0.008	mA	SC On/Ref High	0	0.000	mA				
Row_1_Output_2	0.000	0.000	mA	Reference Circuit Current		0.385	mA	**Estimated Total PSoC Current**		**3.452**	**mA**
Row_1_Output_3	0.000	0.000	mA								
Row Current		0.198	mA					**Estimated Total PSoC Power**		**11.393**	**mW**

Table 8.4. Power consumption of application PSoC in intelligent sensor node.

CASE I: 100% of the time below threshold							
	Mica 2 Mote	**Model 1**			**Model 2**		
Radio		**% Current Consumed**	**% Current Reduced**	**Observed Value**	**% Current Consumed**	**% Current Reduced**	**Observed Value**
Current Transmit in mA	12	8%	92%	0.96	8%	92%	1
Current Receive in mA	8	0%	0%	--	0%	0%	--
Current Sleep in mA	0.002	92%	8%	0.0018	83%	18%	0.0016
PSoC							
Active Current in mA	3.774	100%	0%	3.774	91%	9%	3.452
Current Consumption in mA	23.7740	20%	80%	4.7358	19%	81%	4.4536
% Current Consumption Reduced by Model 2 over Model 1:		5.96%					
CASE II: 50% of the time below threshold							
	Mica 2 Mote	**Model 1**			**Model 2**		
Radio		**% Current Consumed**	**% Current Reduced**	**Observed Value**	**% Current Consumed**	**% Current Reduced**	**Observed Value**
Current Transmit in mA	12	44%	56%	5.28	37%	63%	4.45
Current Receive in mA	8	0%	0%	--	0%	0%	--
Current Sleep in mA	0.002	56%	44%	0.00112	54%	46%	0.001
PSoC							
Active Current in mA	3.774	100%	0%	3.774	91%	9%	3.452
Current Consumption in mA	23.7740	38%	62%	9.05512	33%	68%	7.903
% Current Consumption Reduced by Model 2 over Model 1:		12.72%					
CASE III: 0% of the time below threshold							
	Mica 2 Mote	**Model 1**			**Model 2**		
Radio		**% Current Consumed**	**% Current Reduced**	**Observed Value**	**% Current Consumed**	**% Current Reduced**	**Observed Value**
Current Transmit in mA	12	80%	20%	9.6	74%	26%	8.9
Current Receive in mA	8	0%	0%	--	0%	0%	--
Current Sleep in µA	0.002	20%	80%	0.0004	17%	83%	0.0003
PSoC							
Active Current in mA	3.774	100%	0%	3.774	91%	9%	3.452
Current Consumption in mA	23.7740	56%	44%	13.3744	52%	48%	12.3523
% Current Consumption Reduced by Model 2 over Model 1:		7.64%					

Table 8.5. Performance evaluation of sensor node systems with PSoC interface.

Table 8.5 shows the comparison results between the implemented PSoC interfaced CyFi system (Model 2) and thresholding system (Model 1). Model 1 is implemented using PSoC and MICA2 mote [10]. For performance analysis purpose it is assumed that Model 2 transmits data only when abnormal (bradycardia or tachycardia) heart activity is observed. Model 2 compares its performance with standard MICA2 mote. Whereas, Model 2 compares performance with Model 1 and standard MICA2 mote. Three cases have been considered for performance comparisons.

Case1: 100% of the time ECG signal is below threshold. Model 1 never transmits the data. This is same case for Model 2, as heart rates are never transmitted to base station when normal heart rate is observed.

Case 2: 50% of the time ECG signal is below threshold. Model 1 sends three packets per second or 180 packets per minute. In Model 2 node transmits data when it detects heart rate as either bradycardia or tachycardia. Model 2 assumes tachycardia heart condition for analysis purpose, as highest heart rate is possible here. Thus for one minute maximum of 120BPM is considered. Then number of data transmission packets per minute is calculated as 120. For 50% below threshold, 60 packets per minute or one packet per second is transmitted.

Case 3: 0% of the time ECG signal is below threshold. Model1 transmits 28 packets per second or 1680 packets per minute. Model2 transmits 120 packets per minute or two packets per second. Electrical pulse detection technique employed in Model2 reduces number of packets transmitted per minute by considerable extent.

8.3 ECG Detection and Processing system performance analysis

The power consumption of ECG detection and processing system the PSoC is calculated using the power calculator, as shown in table 8.6. System operates on 3.3V battery and use 12MHz clock for internal operation consuming total of 2.76mA current. Switched capacitor block is started with medium power. PGA and ADC blocks are started in High power mode. Thus total analog block current consumption will be 5.4mA. Analog reference block, GPIO, SPI and Buffer including above current consumptions result in total of 9.469mA of current consumption.

Estimated PSoC Power Calculations for CY8C27xxx Family Device											
Version 1.0									Enter parameters in yellow		
project name	ECG DETECTION AND PROCESSING SYSTEM								See estimated result in green		
part number	CY8C27443			Cypress MicroSystems, Inc.					Indicates something is wrong		
pin count	28										
								OpAmp Bias	0		
CPU Clock	12.000	MHz		SysClkx2Disable	1						
Vdd	3.30	V							Power		
								Analog Block	Level (0-3)		
CPU Current	2.764	mA						ACB00	0	0.000	mA
								ACB01	0	0.000	mA
								ACB02	0	0.000	mA
	Block Clk			GPIO	GPIO Clk			ACB03	3	2.400	mA
Digital Block	(MHz)			Count	(MHz)			ASC10	0	0.000	mA
DBB00	8.000	0.087	mA	1	0.094	0.011	mA	ASD11	0	0.000	mA
DBB01	0.000	0.000	mA	0	0.000	0.000	mA	ASC12	2	0.600	mA
DCB02	0.000	0.000	mA	0	0.000	0.000	mA	ASD13	3	2.400	mA
DCB03	0.000	0.000	mA	0	0.000	0.000	mA	ASD20	0	0.000	mA
DBB10	0.000	0.000	mA	0	0.000	0.000	mA	ASC21	0	0.000	mA
DBB11	0.000	0.000	mA	0	0.000	0.000	mA	ASD22	0	0.000	mA
DCB12	0.094	0.006	mA	0	0.000	0.000	mA	ASC23	0	0.000	mA
DCB13	0.000	0.000	mA	0	0.000	0.000	mA	Analog Block Current		5.400	mA
Digital Block Current		0.092	mA	GPIO Current		0.011	mA				
								A_Buffer Power	0		
	Row Clk			Analog Power				Analog			
Row	(MHz)			All Off	0			Output Buffer			
Row_0_Output_0	0.000	0.000	mA	SC Off/Ref Low	0	0.000	mA	AnalogOutBuf_0	0	0.000	mA
Row_0_Output_1	0.000	0.000	mA	SC Off/Ref Med	0	0.000	mA	AnalogOutBuf_1	0	0.000	mA
Row_0_Output_2	0.000	0.000	mA	SC Off/Ref High	0	0.000	mA	AnalogOutBuf_2	1	0.800	mA
Row_0_Output_3	0.000	0.000	mA	SC On/Ref Low	1	0.385	mA	AnalogOutBuf_3	0	0.000	mA
Row_1_Output_0	0.094	0.008	mA	SC On/Ref Med	0	0.000	mA	Analog Output Buffer Current		0.800	mA
Row_1_Output_1	0.000	0.000	mA	SC On/Ref High	0	0.000	mA				
Row_1_Output_2	0.000	0.000	mA	Reference Circuit Current		0.385	mA	**Estimated Total PSoC Current**		**9.469**	**mA**
Row_1_Output_3	0.094	0.008	mA								
Row Current		0.016	mA					**Estimated Total PSoC Power**		**31.247**	**mW**

Table. 8.6. Power consumption of ECG detection and processing system.

Chapter 9

CONCLUSION AND FUTURE WORK

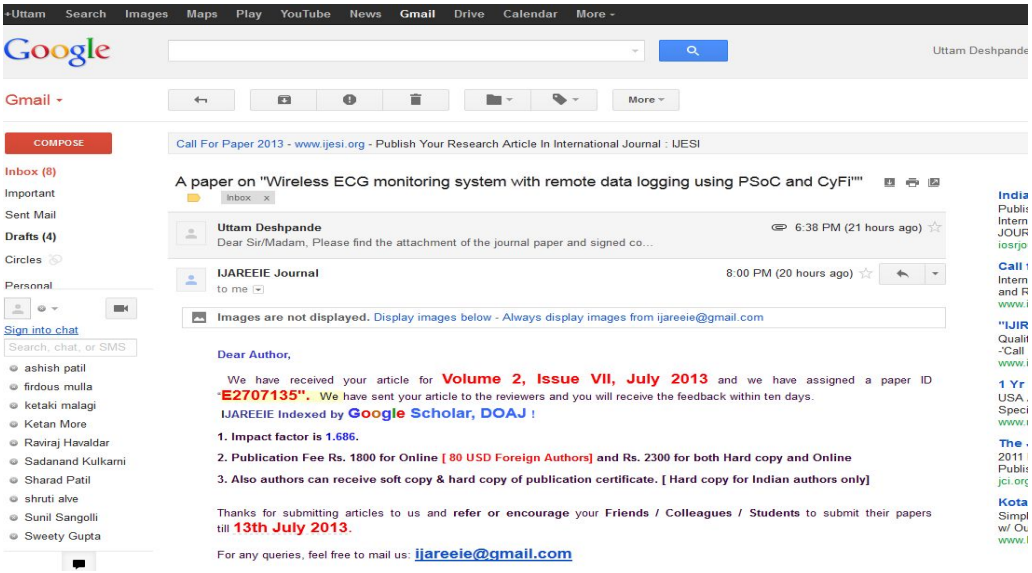
This proposed work implements a low power design by interfacing PSoC with CyFi CY3271 sensor node. ECG detection and processing system demonstrates its ability to detect ECG signal. Threshold adaptive algorithm was developed to facilitate accurate ECG detection. Heart rate classification helps to determine status of heart. It was observed that in comparison with thresholding technique employed by PSoC interfaced MICA2, electrical pulse detection method employed in this work decreases number of packets transmitted per second. Intelligent sensor node enables transmission only when proper data is made available to it. Observations showed that, the node PSoC's current consumption was less than that of MICA2. Reduced data packet and intelligence of sensor node contributes in reduction of overall systems current consumption. This will lead to reduction of data traffic at individual nodes on sensor network and improves lifetime of sensor node. GUI sense and control dashboard demonstrates how it graphs the received data with time and store it in simple .txt file form. This helps for analyzing data for later use. Alarm feature of SCD alerts users whether heart rate received is low (bradycardia), normal or high (tachycardia).

CY3271 is strong enough to handle radio as well as node application PSoC. Since network protocol stack uses code less than 6Kbytes, entire ECG detection and wireless transmission system application can be developed on CY 3271. PSoC used in detection system and sensor node consume about 9.5mA and 3.452mA respectively. Require individual battery to power up. Complete application designed on CY3271 should curtail current consumption and cost. Small size, light weight, low power consumption and ability to perform complex computations make PSoC one of the best choices in developing medical applications.

Publication in International Journal

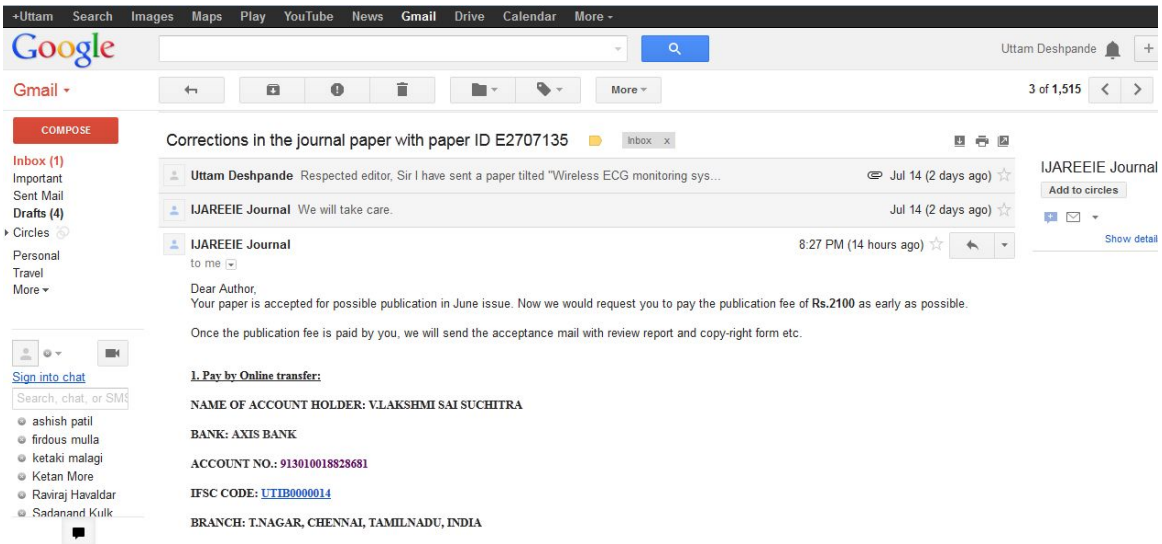
A paper titled “Wireless ECG monitoring system with remote data logging using PSoC and CyFi” is sent for publication in “International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)” on 10-07-2013.

Following is the snapshot of the acknowledgment received by IJAREEIE editor.



Acceptance mail was received on 15-7-2013. Paper will be published in the June issue.

Following is the snapshot of acceptance mail.



REFERENCES

1. *I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Computer. Network. (Elsevier), 2006.*
2. *N. A. Khan, N. Javaid, Z. A. Khan, M. Jaffar, U. Rafiq, A. Bibi , "Ubiquitous HealthCare in Wireless Body Area Networks", IEEE, 2012.*
3. *Fred Chen, Henry Wu, Pei-Lan Hsu, Brad Stronger, Robert Sheridan, and Hongshen Ma, "SmartPad: A Wireless, Adhesive-Electrode-Free, Autonomous ECG Acquisition System", IEEE, 2008.*
4. *Chulsung Park and Pai H. Chou, Ying Bai, Robert Matthews, and Andrew Hibbs, "An Ultra-Wearable, Wireless, Low Power ECG Monitoring System", IEEE, 2006.*
5. *Massot B, Gehin C, Nocua R, Dittmar A, McAdams E,"A wearable, low-power, health-monitoring instrumentation based on a Programmable System-on-Chip", IEEE, 2009.*
6. *Ebrahim Nemati, M. Jamal Deen, and Tapas Mondal, "A Wireless Wearable ECG Sensor for Long-Term Applications", IEEE, 2012.*
7. *TsungHeng Tsai, Jia-Hua Hong,Liang-Hung Wang and Shuenn-Yuh Lee,"Low Power Analog Integrated Circuits for Wireless ECG Acquisition Systems",IEEE,2009.*
8. *Tee Hui Teo, Xinbo Qian, Pradeep Kumar Gopalakrishnan, Yee Shan Hwan, Kuruveettil Haridas, Chin Yann Pang, Hyouk-Kyu Cha, Minkyu Je, "A 700-µW Wireless Sensor Node SoC for Continuous Real-Time Health Monitoring", IEEE, 2010.*
9. *Fei Hu, Yang Xiao,"Congestion-Aware, Loss-Resilient Bio-Monitoring Sensor Networking for Mobile Health Applications", IEEE, 2009.*
10. *Fei Hu, Shruti Lakdawala, Qi Hao and Meikang Qiu, "Low-Power, Intelligent Sensor Hardware Interface for Medical Data Preprocessing", IEEE, 2009.*
11. *Serhiy Matviyenko, Cypress application note AN2284, "Low Cost EKG Pulsometer", 2006.*
12. *Archana Yarlagadda, Applications Engineer, Cypress Semiconductor Corp "Designing a Wireless Heart Rate Monitor with Remote Data Logging", 2009. [Online] Available: http://www.wirelessdesignmag.com.*
13. *(2008) The CY 3271 sensor node website. [Online]. Available:http://www.cypress.com/CY3271sensornode.*

APPENDIX A

CyFi Transceiver

Data Transmission Modes: The CyFi radio transceiver supports two different data transmission modes,

- GFSK mode, data is transmitted at 1 Mbps, without any DSSS.
- 8DR mode, DSSS is enabled and eight bits are encoded in each derived code symbol transmitted.

Both 64 chip and 32 chip pseudo noise (PN) codes are supported in 8DR mode. In general, lower data rates reduce packet error rate in any given environment.

Packet Framing: The CYRF7936 IC device supports the following data packet framing features,

- SOP: Packets begin with a two symbol SOP marker. The SOP_CODE_ADR PN code used for the SOP is different from that used for the "body" of the packet, and if necessary may be a different length. SOP must be configured to be the same length on both sides of the link.
- Length: This is the first eight bits after the SOP symbol and is transmitted at the payload data rate. An EOP condition is inferred after reception of the number of bytes defined in the length field, plus two bytes for the CRC16.
- CRC16: The device may be configured to append a 16 bit CRC16 to each packet. The CRC16 uses the USB CRC polynomial with the added programmability of the seed. If enabled, the receiver verifies the calculated CRC16 for the payload data against the received value in the CRC16 field. The seed value for the CRC16 calculation is configurable, and the CRC16 transmitted may be calculated using either the loaded seed value or a zero seed. The received data CRC16 is checked against both the configured and zero CRC16 seeds. CRC16 detects the following errors:
  i. One bit in error.
  ii. Any two bits in error.
  iii. Any odd number of bits in error.
  iv. An error burst as wide as the checksum itself.

Figure 1 shows an example packet with SOP, CRC16, and lengths fields enabled and Figure 2 shows a standard ACK packet.

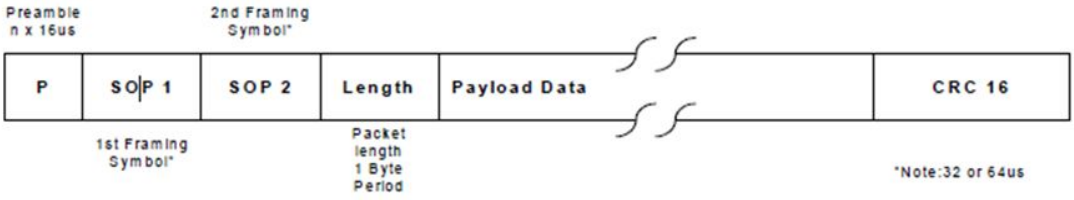


Fig. 1. Packet indicating various fields.

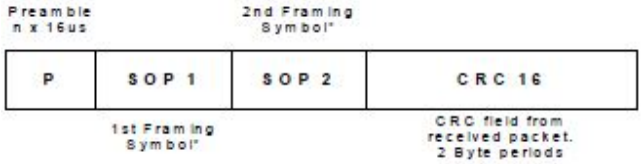


Fig. 2. ACK packet.

Packet Buffers: All data transmission and reception use the 16 byte packet buffers, one for transmission and one for reception. The transmit buffer allows loading a complete packet of up to 16 bytes of payload data in one burst SPI transaction. This is then transmitted with no further MCU intervention. Similarly, the receive buffer allows receiving an entire packet of payload data up to 16 bytes with no firmware intervention required until the packet reception is complete. Maximum packet length depends on the accuracy of the clock on each end of the link. Packet lengths up to 40 bytes are supported when the data between the transmitter and receiver crystals is 60 ppm or better. Interrupts are provided to allow an MCU to use the transmit and receive buffers as FIFOs. When transmitting a packet longer than 16 bytes, the MCU can load 16 bytes initially, and add further bytes to the transmit buffer as transmission of data creates space in the buffer. Similarly, when receiving packets longer than 16 bytes, the MCU must fetch received data from the FIFO periodically during packet reception to prevent it from overflowing.

Auto Transaction Sequencer (ATS): The CYRF7936 IC provides automated support for transmission and reception of acknowledged data packets.

When transmitting in transaction mode, the device,

- Starts the crystal and synthesizer
- It enters transmit mode

- It transmits the packet in the transmit buffer
- It transitions to receive mode and waits for an ACK packet
- Transactions to the end state when an ACK packet is received or a timeout period expires

Similarly, when receiving in transaction mode, the device,

- Waits in receive mode for a valid packet to be received
- Transitions to transmit mode, transmits an ACK packet
- Transitions to the transaction end state.

The contents of the packet buffers are not affected by the transmission or reception of ACK packets. In each case, the entire packet transaction takes place without any need for MCU firmware action (as long as packets of 16 bytes or less are used). To transmit data, the MCU must load the data packet to be transmitted, set the length, and set the TX GO bit. Similarly, when receiving packets in transaction mode, firmware must retrieve the fully received packet in response to an interrupt request indicating reception of a packet.

Data Rates: The CYRF7936 IC supports the following data rates by combining the PN code lengths and data transmission modes described in the previous sections.

- 1000 kbps (GFSK)
- 250 kbps (32 chip 8DR)
- 125 kbps (64 chip 8DR)

2.4-GHz CyFi Radio Modem:

The CyFi radio modem is a dual conversion low IF architecture optimized for power, range, and robustness. The CyFi radio modem employs channel-matched filters to achieve high performance in the presence of interference. An integrated power amplifier (PA) provides up to +4 dBm transmit power, with an output power control range of 34 dB in seven steps. The supply current of the device is reduced as the RF output power is reduced.

PA Setting	Typical Output Power (dBm)
7	+4
6	0
5	-5
4	-13
3	-18
2	-24
1	-30
0	-35

Table.1. Internal PA output power.

Frequency Synthesizer: Prior to transmission or reception, the frequency synthesizer must settle. The settling time varies depending on the channel, 25 fast channels are provided with a maximum settling time of 100 µs. The 'fast channels' (less than 100µs settling time) are every third channel, starting at 0 up to and including 72 (for example, 0, 3, 6 ….66, 69, 72).

Baseband and Framer: The baseband and framer blocks provide the DSSS encoding and decoding, SOP generation and reception, CRC16 generation and checking, and EOP detection and length field.

Packet Buffers and Radio Configuration Registers: Packet data and configuration registers are accessed through the SPI interface. All configuration registers are directly addressed through the address field in the SPI packet. Configuration registers allow configuration of DSSS PN codes, data rate, operating mode, interrupt masks, interrupt status, and so on.

SPI Interface: The CYRF7936 IC has an SPI interface supporting communication between an application MCU and one or more slave devices (including the CYRF7936). The SPI interface supports single-byte and multi-byte serial transfers using either 4-pin or 3-pin interfacing. The SPI communications interface consists of slave select (~SS), serial clock (SCK), MOSI, MISO, or SDAT.

SPI communication is described as follows:

- Command direction (bit 7) = '1' enables SPI write transaction. When it equals a '0', it enables SPI read transactions.
- Command increment (bit 6) = '1' enables SPI auto address increment. When set, the address field automatically increments at the end of each data byte in a burst access. Otherwise the same address is accessed.
- Six bits of address.
- Eight bits of data.

The device receives SCK from an application MCU on the SCK pin. Data from the application MCU is shifted in on the MOSI pin. Data to the application MCU is shifted out on the MISO pin. The active LOW ~SS pin must be asserted to initiate an SPI transfer. The application MCU can initiate SPI data transfers using a multibyte transaction. The first byte is the Command/Address byte and the following bytes are the data bytes. SPI transaction format is as shown in table below. Data transfer is shown in Figure 15.

Parameter	**Byte 1**			**Byte 1+N**
Bit Number	7	6	[5:0]	[7:0]
Bit Name	DIR	INC	Address	Data

Table.2. SPI transaction format.

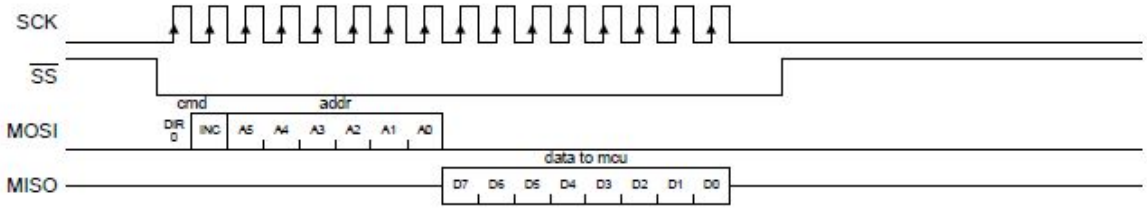


Figure. 3. SPI Single Read Sequence

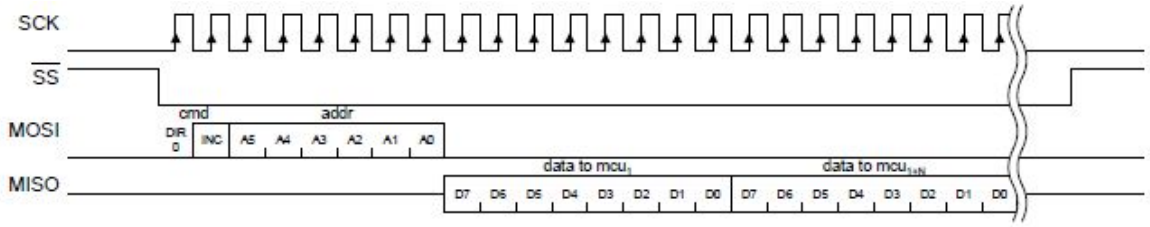


Figure. 4. SPI Incrementing Burst Read Sequence

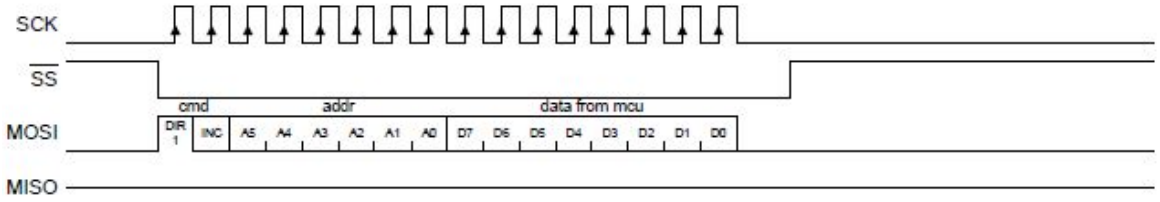


Figure.5. SPI Single Write Sequence

Figure. 6. SPI Incrementing Burst Write Sequence

The SPI communications interface has a burst mechanism, where the first byte can be followed by as many data bytes as required. A burst transaction is terminated by deasserting

the slave select (~SS = 1). The SPI communications interface single read and burst read sequences are shown in Figure 3 and 4 respectively.

The SPI communications interface single write and burst write sequences are shown in Figure 5 and 6 respectively.

This interface may be optionally operated in a 3 pin mode with the MISO and MOSI functions combined in a single bidirectional data pin (SDAT). When using the 3 pin mode, firmware must ensure that the MOSI pin on the MCU is in a high-impedance state except when MOSI is actively transmitting data. The device registers may be written to or read from one byte at a time, or several sequential register locations may be written or read in a single SPI transaction using incrementing burst mode. In addition to single byte configuration registers, the device includes register files. Register files are FIFOs written to and read from using no incrementing burst SPI transactions. The IRQ pin function may be optionally multiplexed to the MOSI pin. When this option is enabled, the IRQ function is not available while the ~SS pin is LOW. When using this configuration, firmware must ensure that the MOSI pin on the MCU is in a high impedance state whenever the ~SS pin is HIGH. The SPI interface is not dependent on the internal 12 MHz clock. Registers may therefore be read from or written to when the device is in sleep mode, and the 12 MHz oscillator disabled. The SPI interface and the IRQ and RST pins have a separate voltage reference pin (VIO). This enables the device to interface directly to MCUs operating at voltages below the CYRF7936 IC supply voltage.

Interrupts: The device provides an interrupt output, which is configurable to indicate the occurrence of different events. The IRQ pin can be programmed to be either active HIGH or active LOW, it can be a CMOS or open drain output. The CYRF7936 IC features three sets of interrupts: transmit, receive, and system interrupts. These interrupts all share a single pin (IRQ), but can be independently enabled or disabled. The contents of the enable registers are preserved when switching between transmit and receive modes. If more than one interrupt is enabled at any time, it is necessary to read the relevant status register to determine which event caused the IRQ pin to assert. Even when a given interrupt source is disabled, the status of the condition that otherwise causes an interrupt can be determined by reading the

appropriate status register. It is therefore possible to use devices without the IRQ pin, by polling the status registers to wait for an event, rather than using the IRQ pin. Clocks A 12 MHz crystal (30 ppm or better) is directly connected between XTAL and GND without the need for external capacitors. A digital clock out function is provided, with selectable output frequencies of 0.75, 1.5, 3, 6, or 12 MHz. This output may be used to clock an external microcontroller (MCU) or ASIC. This output is enabled by default, but may be disabled. The requirements to directly connect the crystal to the XTAL pin and GND are:

- Nominal frequency: 12 MHz
- Operating mode: Fundamental mode
- Resonance mode: Parallel resonant
- Frequency stability: ±30 ppm
- Series resistance: <60 Ω
- Load capacitance: 10 pF
- Drive level: 100 µW

Power Management: The operating voltage of the device is 1.8 V to 3.6 V DC, which is applied to the VBAT pin. The device can be shut down to a fully static sleep mode by writing to the FRC END = 1 and END STATE = 000 bits in the XACT_CFG_ADR register over the SPI interface. The device enters sleep mode within 35 µs after the last SCK positive edge at the end of this SPI transaction. Alternatively, the device may be configured to automatically enter sleep mode after completing the packet transmission or reception. When in sleep mode, the on-chip oscillator is stopped, but the SPI interface remains functional. The device wakes from sleep mode automatically when the device is commanded to enter transmit or receive mode. When resuming from sleep mode, there is a short delay while the oscillator restarts. The device can be configured to assert the IRQ pin when the oscillator has stabilized.

APPENDIX B

Configuring hardware and Sense and Control Dashboard

Steps to write and execute program:

1. Install PSoC Designer 5.0, PSoC Programmer 3.0 and Sense and Control dashboard 2.0 provided with CY3271 kit.

2. Open new project.

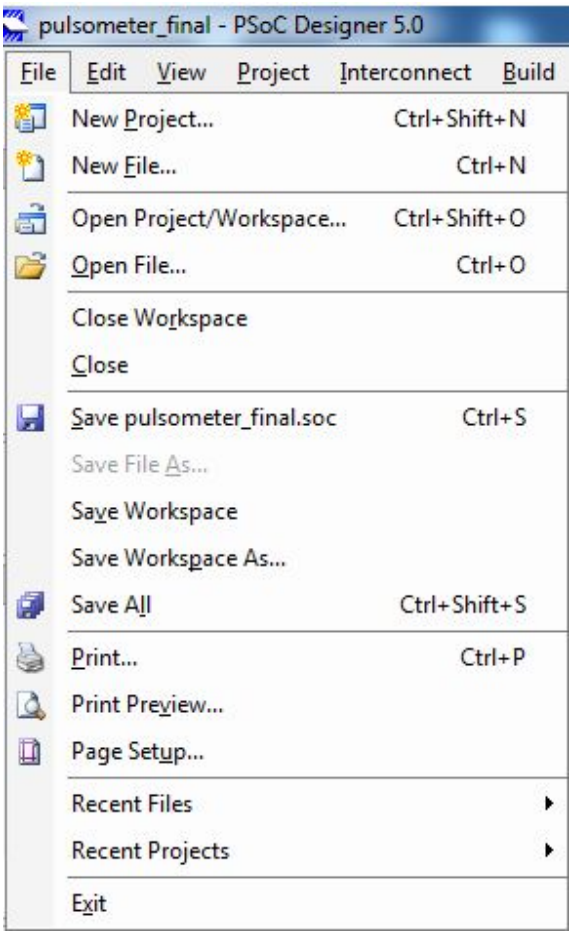


Figure. 1. Creating new project.

For the required application, select and configure user modules from global resources. Route signals from/to different pins on PSoC.

3. Open .c file and write the program for the given application.

4. Build the project to check any errors. Build generates .Hex file with the name of the project.

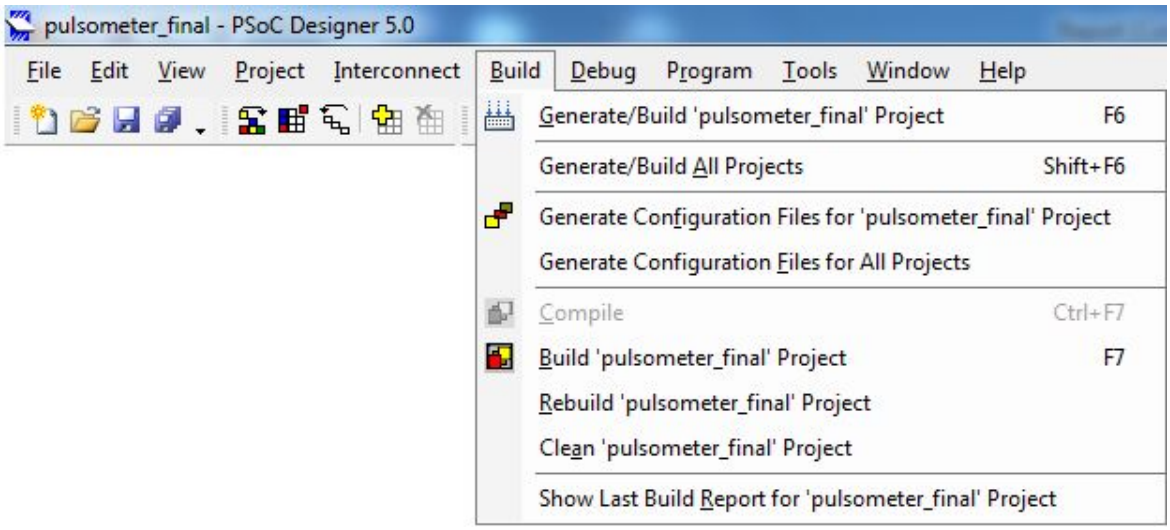


Figure. 2. Build project.

5. Connect PSoC MiniProg on the free available USB port.

Figure. 3. PSoC MiniProg.

6. Open PSoC Programmer, select target device family, device and download project name.Hex file onto target device.

7. Supply power by clicking on Toggle device power from PSoC programmer.

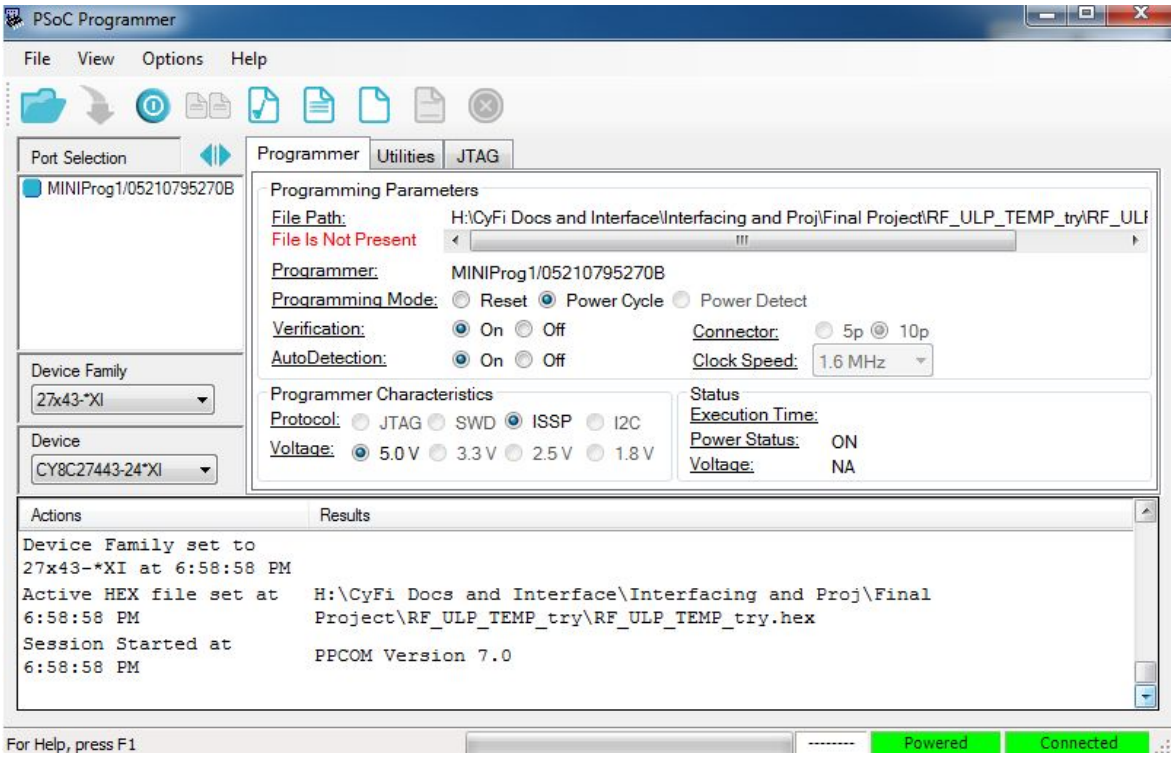


Figure. 4. PSoC Programmer.

Configuring ECG signal processing and detection system:

1. Repeat above steps 1 to 4.
2. Connect kit PSoC Eval1 CY3210 kit to PSoC MiniProg.
3. Follow above steps 5 to 7.
4. Connect ECG simulator to pin P0[6] on CY3210 kit, and turn it on.
5. Observe heart rate on the output.

Configuring Node:

1. Repeat steps 1 to 4 from program write and execute procedure.
2. Connect the RF PC Bridge into any free USB port on your computer. The blue LED should start flashing indicating that the PC Bridge is enumerated on the USB bus.
3. Insert RF expansion board male header to PC bridge female expansion port.
4. Select target family, device and program the RF expansion board by selecting the RF_ULP_TEMP_try.Hex file.
5. Close PSoC programmer.

6. Connect MOSI, MISO, ~SS, SCLK pins of PSOC kit CY3210 with pins of application PSoC of RF expansion board.

7. Insert the coin cell into the battery board + side up.

Configuring Hub:

1. Connect the RF PC Bridge into any free USB port on your computer. The blue LED should start flashing indicating that the PC Bridge is enumerated on the USB bus.

2. Program the PC Bridge slave processor by selecting the RF_HUB.Hex file.

3. Close PSoC Programmer.

4. Open the SCD Dashboard software GUI that is installed with the installation pack.

To access the SCD Dashboard software, go to: Start > Programs > Cypress > Cypress Sense and Control Dashboard > Cypress Sense and Control Dashboard.

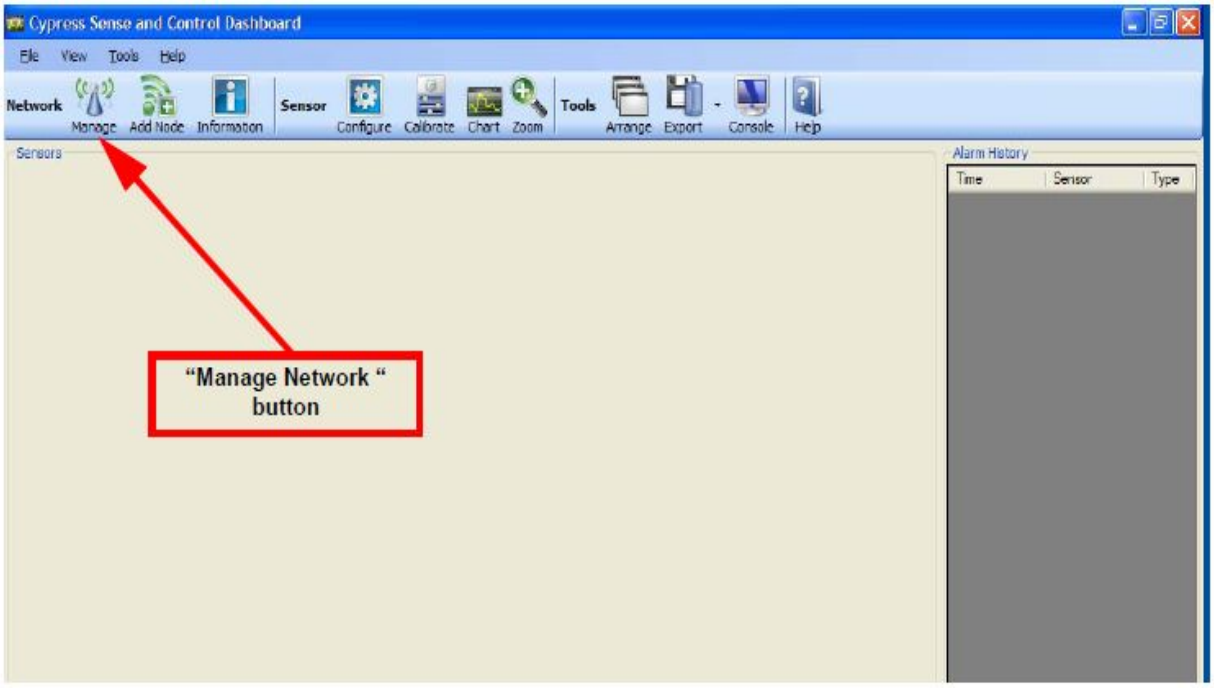


Figure. 5. Manage Network.

5. In the Manage Network screen, click Add to add a new node. On the Node Binding screen, click Begin Binding. When the green LED on the bridge lights up, the bridge is in bind mode (there is no need to press the button).

Figure. 6. Node Binding.

6. Place the RF Expansion Board in Bind mode, by pressing the Bind button on the board when instructed by the GUI.

7. Verify the success of the bind and click next. After it is bound, click next. Then name the node and click Finish.

8. In the Configure Node screen, click Add to define the sensors for this node.

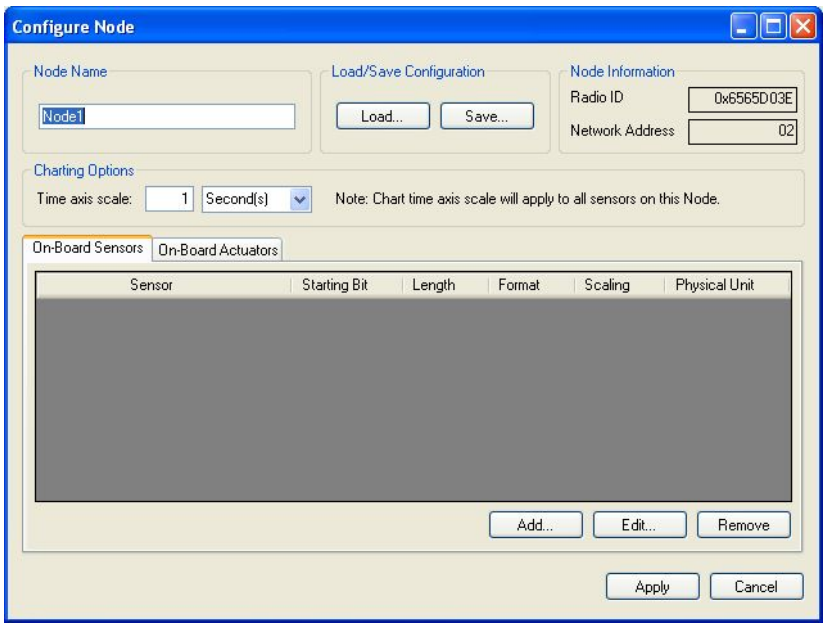


Figure. 7. Configure Node.

9. Configure the sensor as shown in figure below.

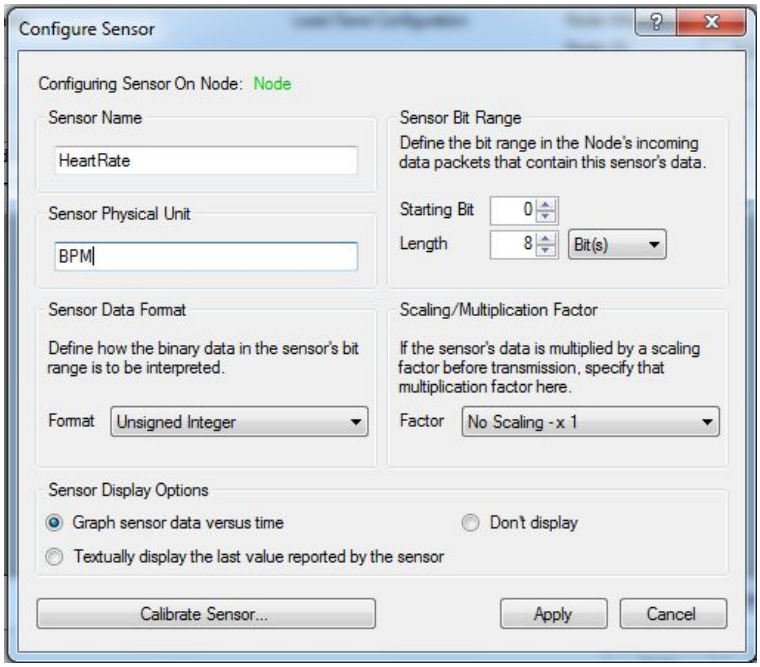


Figure. 8. Configure Sensor.

10. Select graphical mode of data display. The data is displayed in graphical format on the Sense and Control Dashboard screen.

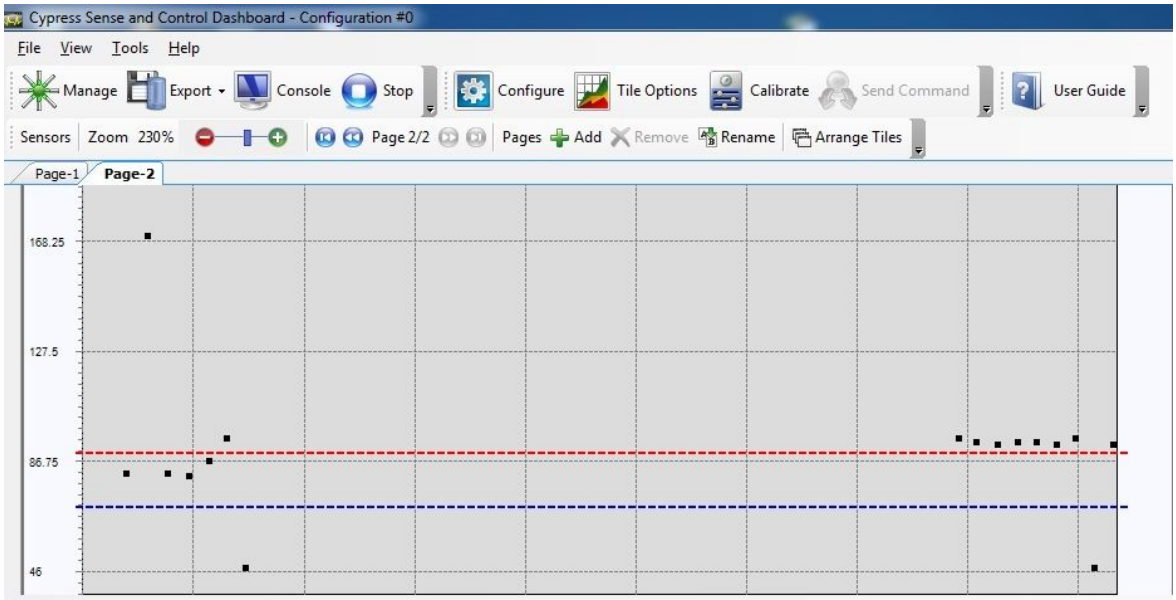


Figure. 9. Cypress SCD.

11. Or Select textual mode of data display. The data is displayed in textual format on the SCD screen.

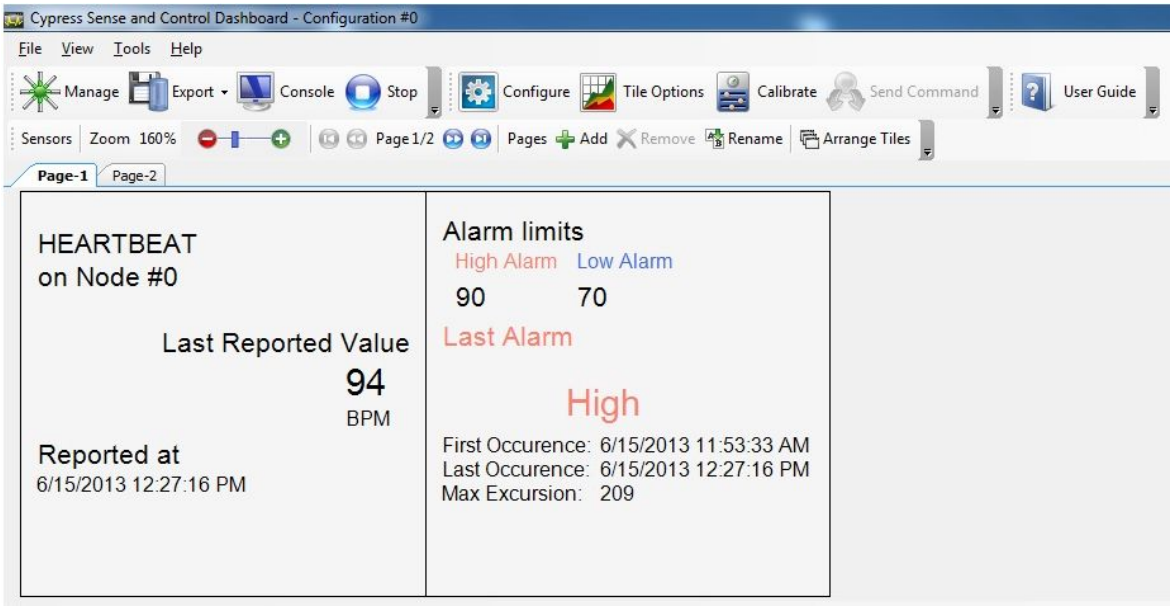


Figure. 10. Textual data and Alarm.

12. Here alarm can be configured by clicking Configure>Tile Options>Set High Alarm>Set Low Alarm.

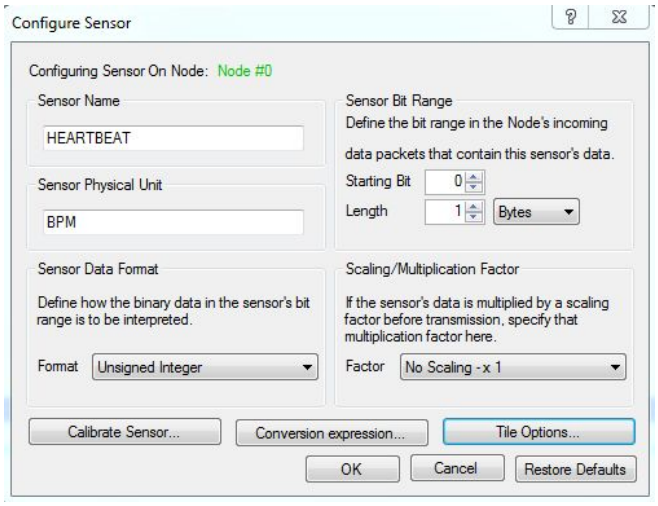


Figure. 11. Alarm configuring.

Here High Alarm is set as 90 BPM (Tachycardia) and Low Alarm as 60 BPM (Bradycardia).

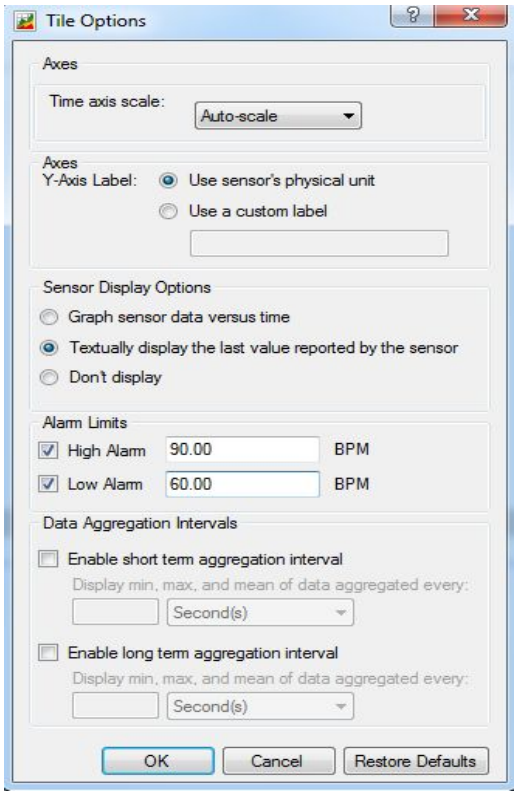


Figure. 12.Setting Alarm values.

13. We can also see the output on console. Here data is displayed in terms of Hexadecimal number. We can check the system configuration, Bind result and also can change report timer if configured. Console displays following fields with data.

Resp - Response on the command using mask 0x80.

Rem - (Remaining Message) Indicates the number of node data packets (messages) left in the node packet queue. If the queue is empty prior to execution of this command, 0xFF is returned in this byte.

RSSI - Level of signal of wireless node.

The Fetch Message button allows manual message fetching (not on timer).

Polling

Enable Polling: Starts the timer that fetches messages.

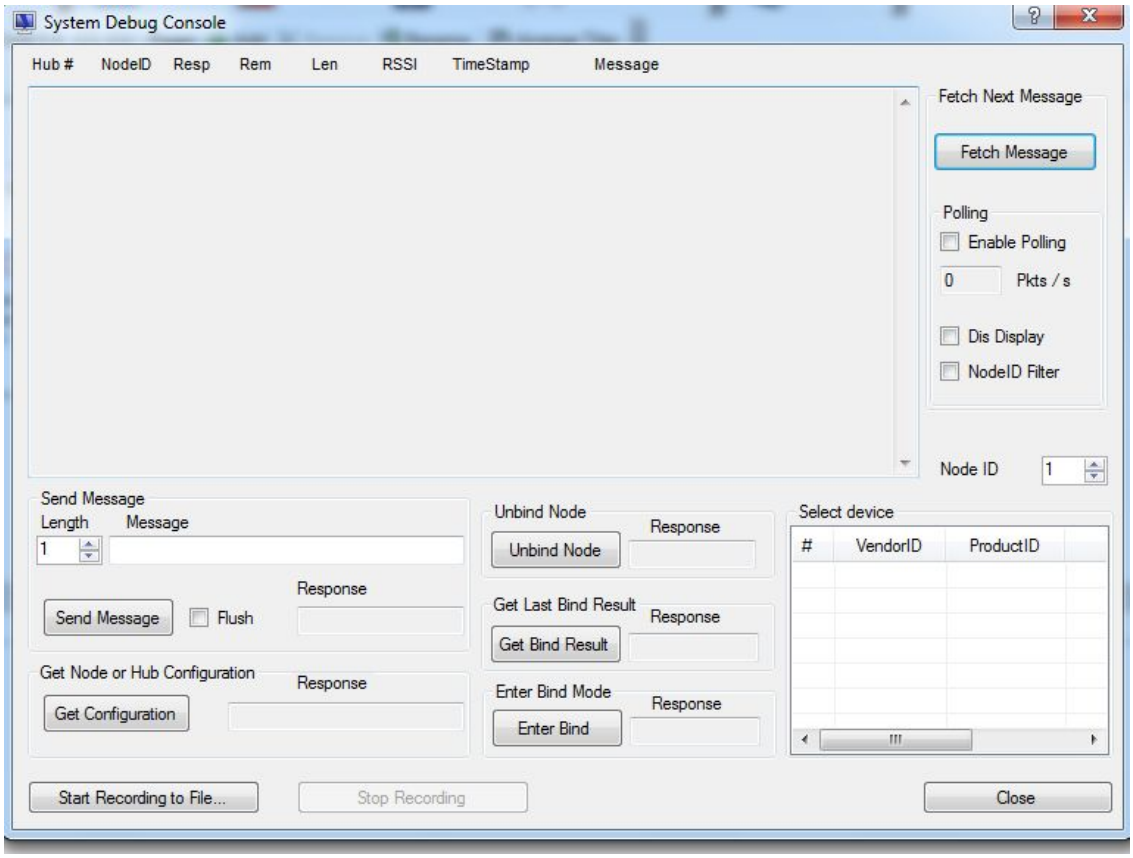


Figure. 13. Setting console to receive data.

Dis Display: Disables message display in the text window.

NodeID Filter: Displays only messages of Node ID set on the right panel of the screen.

Start Recording to File : Allows storing information to the file.

Stop Recording: Selecting this button stops storing information to the file.

Start Recording to File: Allows storing information to the file.

Send Message: You can send the message entered in the Message box by clicking this button. The status of the sent message and the response is received in the Response text box.

Flush: The flush option when checked will have all the sent messages flushed.

Get Configuration: Use this option to know the node configuration, status of the command sent and the node ID.

Unbind Node: Use this button if you choose to unbind the node. On choosing so, the system pops up a warning that unbinding the node will discontinue data acquisition and that the node will be removed from the system configuration.

Get Bind Result: Use this button to know the result of the bind action. The result will be populated in the Response field.

Enter Bind: Use this button to bind the node to the hub. When you select this button, the Select device displays the list of the nodes.

Remove the node from the system configuration : If you choose this option, the debug console warns you that all its associated sensors, actuators, and output rules will be removed from the system configuration.