International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 3415-3424
© Research India Publications. http://www.ripublication.com

# Sensor Node Design with Dynamic Remote Reconfiguration and Analysis

**Hiren Patel**
[1]*Assistant Professor, EC Department, BVM Engineering College, Vallabh Vidyangar- 388120, Gujarat, India.*
[2]*Research Scholar, School of Engineering, RK University, Rajkot - 360020, Gujarat, India.*

**Prof. (Dr.) Vipul A. Shah**
*Faculty of Technology, Instrumentation & Control Engineering Department,*
*Dharmsinh Desai University, Nadiad - 387 001, Gujarat, India.*

## Abstract

Sensor Nodes (SN) are key components in a wireless sensor networks (WSN). With varying technologies, the sensor nodes have evolved to great extent considering processing, communication and power consumption. In this paper we have discussed MSP430 based sensor node development and its consumption for various activities performed during its functioning when deployed in field. Also we have proposed an algorithm to dynamically reconfigure the sensor node remotely to optimize lifetime of sensor node by dynamically varying resource setting to function efficiently under changing conditions.

## INTRODUCTION

Wireless sensor networks (WSN) is widely used to control and monitor the conditions remotely. Sensor nodes are equipped with various sensors and its conditioning circuits along with networking component for communication. Depending upon the application requirement, sensors for quantities such as temperature, pressure, level sensor, hall sensor, proximity sensor, accelerometers, and many more are added to the sensor node. Along with the sensors and its conditioning circuits, sensor nodes also contain a transceiver for forming wireless link with the other sensor nodes for communicating data in the network. Depending upon the requirement transceiver modules may vary and include any of\ these; Wi-Fi, Bluetooth, ZigBee or simply RF module.

The sensor nodes are deployed with the constraint of energy resources and limitations with a purpose to serve assigned task. To meet the need and survive with the resources available it has to adopt to energy conserving techniques that work at each layer of node while its design process[1]. Each sensors and communication module type along with the processing unit accounts for consumption that differs to great extent along with their features.

The design factors at various levels play an eminent role in energy conservation which mainly includes Hardware optimization and software optimization. The hardware optimization looks forward for Processing unit with low power consumption, Radio for communication with less energy consumption and other hardware peripherals[2] that consume power, while Software optimization criteria involve Time for active period i.e. duty cycle, Data reduction for communication and way the communication takes place between nodes.

In this paper we propose a sensor node design based on ultra-low power consuming MSP430 MCU which can be dynamically configured to alter the settings for optimizing the consumptions in sensor node. The paper is organized to show the review on sensor nodes and its design considerations along with the sensor node reconfiguration techniques in section two followed by the architecture we developed considering the review work done on various sensor node designs. After the proposed architecture design and its analysis we proposed the working of the algorithm for dynamic remote reconfiguration which is justified by our test results shown at the end with conclusion.

## LITERATURE REVIEW

In [3] the author has reviewed various sensor nodes such as Mica2/MicaZ Motes, TelosB Motes, Indriya-Zigbee Based WSN Development Platform[4], IRIS, iSense Core Wireless Module, Preon32-Wireless Module, Wasp Mote, WiSense Mote and panStamp NRG Mote. Also various transceiver modules are reviewed and compared such as NRF24L01, CC1101, RFM22B-S2\SMD Wireless Transceiver-434 MHz, TRM-433-LT, CC2500 and MC1322x. Each has varying capabilities that include Frequency, TX Power, RX Sensitivity, Interface, Data rate, operating voltage Supply current, Power down current, Stand-by current, Sleep current, Operating temperature, Range, Package and Cost.

There are various types of wireless sensor nodes designed and are in use for different application of wireless sensor networks. These sensors are built around variety of MCUs architectures including FPGAs and system on chip(SOC) on 0.18µm CMOS [5] [6][7]. The capability of processing unit used in sensor node varies to meet the application specific needs.

Depending upon the scenarios it may require the node size be small and compact enough looking to the constraints implied for deploying. In [8],the author has designed smallest sensor node with dimensions 3.9 mm × 3.9 mm × 3.5 mm to monitor humidity and temperature on 0.18µm 1P6M logic process. It uses RF-transmitter IC Si-4010 for wireless communication which consumes about 10mA during communication of it with sensor SHT-21.

In [9], the author presented design of sensor node for UV detection of flame using ZnSSe UV photodetector. It uses ZigBee for wireless communication and consumes about 2.3 mW from 3.3V source. As the sensor node is deployed in field with limited power source so energy conservation is important aspect for a sensor design. There are various techniques[10] developed for reducing power consumption such as dynamic voltage scaling, energy harvesting techniques, transmit power adjust, vary clock rate, frequency scaling and selective power supply scheme[6]. In one of paper[11], the author has addressed the gas sensing application and designed a sensor node that helps in reduction in power consumed using ATxmega324 by employing context aware approach with Pyroelectric Infrared (PIR) sensor. In this way the power hungry sensor acquisition may be avoided by getting coarse reading to act upon for further acquisition.

In some cases the reconfiguration at software level and hardware level is also employed in sensor nodes by using Field Programmable Gate Array (FPGA) based sensor node[12]. In this case the reconfiguration file is sent to sensor node over wireless communication to alter configurations. Mostly a node designed uses some sort of operating system to manage the tasks on sensor node[13]. In [2], author discussed implementation of RTOS such as TinyOS, Contriki and AmbientRT. The OS implements dynamic reconfiguration with dynamic loadable modules that are preloaded in sensor node. In this way communication cost is reduced to transfer the code to node. It may also be required to dynamically reconfigure network autonomously and adapt to current network conditions[14].

## ARCHITECTURE OF A SENSOR NODE

Typically, a sensor node consists of a processing unit, transceiver and sensor unit with conditioning circuits. The features of a sensor node vary to great extent depending upon the application need. The sensor node is intended to communicate with sink and transfer the sensor data for accumulation at sink node or base station via cluster head. The sensor node is supposed to consume less power and efficiently transmit the processed data to the destination. So the controller used for the sensor node is chosen to be less power consuming and having overall features included required meeting the application needs.

The Figure 1 shows the typical sensor node where we have customized it with use of MSP430G2553 by Texas Instruments as processing unit and RF communication module – nRF24L01+ from Nordic Semiconductors. The RF module use SPI interface for communicating with MCU for data and configuration commands. The sensors module is interfaced with ADC of MCU via conditioning circuits for acquisition.

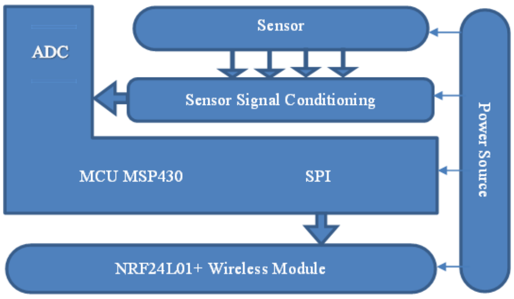

**Figure 1.** Sensor node Architecture

### a. Processing unit

The processing unit of sensor node plays a very eminent role when distributed processing is targeted in sensor network at node level. According to the processing type and application needs, the processor is selected. For our consideration to design a versatile sensor node with limited processing we have considered ultra-low power consuming MCU – MSP430G2553. It is equipped with the entire necessary peripheral interface required typically in a sensor node.

Along with the considerations of peripherals required, taking in to account consumptions, it has four low power modes that can be configured during idle mode of sensor node. Each mode functions differently by keeping the clock enabled for selected modules thereby reducing consumption of unused modules in idle mode. For different low power modes, the current consumption varies from 0.5µA to 56 µA. The active mode current consumption varies from 420µA to 4.75mA, which also depend on the operating frequency of MCU. The active mode consumption also depends on the supply voltage i.e. 2.2V to 3.6V, which means that variation in supply causes effect in consumption. Thus it can be said that operating frequency and supply voltage is directly proportional to consumption, so appropriately adjusting each may lead to conserve energy without affecting its functionality.

The operating voltage cannot be set programmatically from within the microcontroller until unless the respective circuit is added externally to vary input voltage by command from microcontroller pins. But the operating frequency of microcontroller can be changed programmatically from within by setting bits in respective clock control register[15]. The DCO bits (DCO.1 and DCO.0) and Range select bits (RSEL.3 to RSEL.1) are altered vary MCU frequency from 0.06MHz to 16MHz. DCO bits are used to select step size within the selected range of frequency by RSEL.x bits. On increasing the CPU frequency from 1MHz to 12 MHz the current consumption typically increases from 330µA to about 2.80mA. The technique of frequency scaling can be used only when the sufficient processing time is available between consecutive transmissions.

### b. ADC resolution and acquisition rate

Sensor nodes are meant for sensing the surrounding environment for change and report it back to the base station for activity detected. So, for a sensor node, Analog to Digital

Conversion (ADC) is most important part of the sensing system. Whenever there is change in the surrounding environment the change in the physical quantity of interest is also changing for which the monitoring application is build. To sense the particular change in the quantity a sensor is required which in turn senses the quantity and passes the signals to signal conditioning circuits. The Signal Conditioning circuits converts to appropriate electrical signals along with conditioning such as filtering, amplifying, etc… These conditioned signals are passed on to the ADC which converts the analog form to digital form that could be understandable by MCU for further processing.

The processing unit selected for sensor node design has a 10-bit ADC inbuilt on the chip. For ADCs, figure of merit (FOM) considers power, sampling rate and effective number of bits and relates each to derive inter-dependencies:

$$Figure\ of\ Merit\ (FOM) = \left(\frac{P}{F_S \times 2^{ENOB}}\right) \qquad \ldots. (1)$$

The power P is related directly to sampling rate which suggests trade-off between: sampling rate and resolution. As the resolution increases the energy consumed also increases and similar is the case with sampling rate as well. The ADC module in MCU support 10-bit resolution with successive approximation (SAR) core along with that it has sampling rate control up to 200ksps, reference generator and selector and data transfer controller.

Out of the all possible design factors the efficient data acquisition is most under explored so far as studies are concerned on previous work attempts. The energy spent on the acquisition is generally neglected in many of the work and studies, but still it consumes a portion of energy. It is not all the time that the data acquisition and communicating the same is required at the constant rate and thereby accumulating redundant data which is not of use. Even the quality of data matters so far as the energy consumption is concerned whereas sometimes the coarse reading does the function instead of fine-quality data. In such case the energy spent on acquisition of the fine readings can be reduced. The table 1 shows current consumption statistics shows for ADC module.

**Table 1.** ADC Module Consumptions

| Parameter (Current consumption) | Conditions | Typical | Units |
|---|---|---|---|
| ADC Supply Current | Vcc = 3V | 0.6 | mA |
| Internal reference voltage | Vcc = 3V | 0.25 | mA |
| Sampling rate of 200ksps | Vcc = 3V | 1.1 | mA |
| Sampling rate of 50ksps | Vcc = 3V | 0.5 | mA |

As seen from the table, the consumption for ADC account up to 1.95mA for initiating acquisition. By precisely selecting the sampling rate the consumption can be reduced to 0.5mA for 50ksps if it can meet the requirements. So far as acquisition is concerned, the factors that could be scaled to optimize consumption include sampling rate and resolution for ADC.

**c. Wireless module**

The sensor network is mostly formed over wireless connectivity so a wireless module is most essential and important part of sensor node. The wireless connectivity can be achieved in various way using different types of wireless modules, this includes Wi-Fi, Bluetooth, ZigBee and RF modules. Again it depends on the range and feature requirement while implementing network. These factors, range and data rate for communication, consumes the significant portion of energy. To meet the required throughput in network for divided timeslots, the wireless modules need to have appropriate data rate in order to handle the communication in given time. The range of module comes to the picture when in a scenario, the nodes are separated by distance apart than range of the radio should be high enough to communicate with the farthest node in group/cluster. As per observed fact, the power consumed to communicate the acquired data with the sink forms major part of total energy consumption in a sensor node.

We are using nRF24L01+ from Nordic Semiconductors for wireless connectivity, that operates on 2.4GHz frequency with 126 RF channels and suitable for ultra-low power wireless applications. It is having an embedded baseband protocol engine which supports packet communication. Enhanced Shock-Burst™ reduces system cost by handling all the high speed link layer operations at data link layer[16]. Enhanced Shock-Burst™ Packet frame as shown in figure 2 is formed of preamble (1 Byte), Address of receiver (3 to 5 byte), packet control field (9 bit), CRC (1 or 2 byte) and payload (0 to 32 bytes). The packet control field consists of payload length indicated by 6 most significant bits followed by packet identification bit which are of 2 bit in length. Lastly, least significant bit is for No_ACK to indicate acknowledgment from receiver. During the transmission mode the packet size can be from 8+24+9+8+8 = 57bits minimum to 8+40+9+32*8+16 = 329-bit maximum at a time.

For our algorithm we added 2 byte fields in 32bytes payload field that helps the receiver to know channel condition and the sequence number of packet respectively. The channel condition is indicated and estimated by packet error rate (PER), which show how link is prone to loss of data. The sequence number of packet is to be known when delivering chunks of data in multiple frames and identify packets lost by this number.

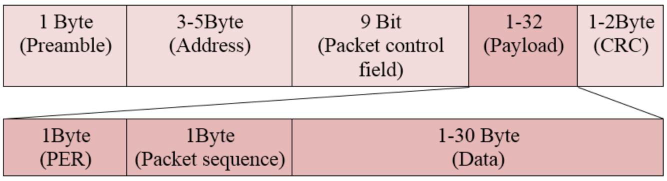

**Figure 2 .** An Enhanced ShockBurst Packet with payload (1-32)

The RF module takes serial peripheral Interface (SPI) to communicate with MCU for access of control registers to set user configurable parameters like frequency channel, output power and air data rate. The nRF24L01+ supports variable air data rate that vary from 250 kbps to 2Mbps. The high air data rates along with two power saving modes makes the nRF24L01+ very suitable for ultra-low power sensor node designs.

In RF module used, there are 4 distinct transmit power modes that are configurable as per the communication range to achieve. So there is a trade-off between power consumed to that of distance for transmission range. The transmission range can be extended by increasing transmit power which in turn will increase more current consumption. The Figure 3 shows 3D plot projections with axis indicating energy consumed, Current consumption for Transmit power setting in mA and Data length used to communicate in bytes. Maximum power is consumed when full data length of frame is utilized and transmitted same with full transmission power setting. The optimal values of data length and transmit power can be set to achieve energy conservation.

The Figure 4 shows the percentage consumption difference between consecutive power mode settings with varying data-length. The analysis states that the difference may vary from 0.6% to about 8% with varying power settings optimally.

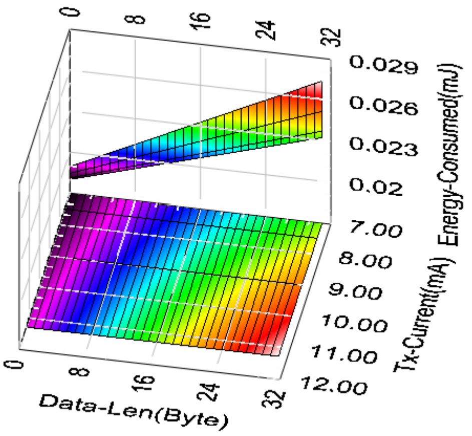

**Figure 3.** Energy consumed for varying transmission range and data length

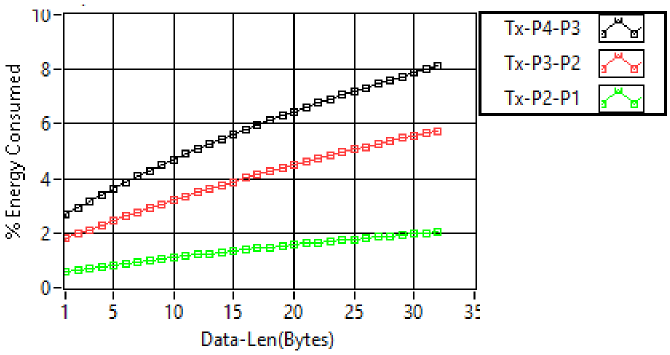

**Figure 4.** Energy Consumed (in %) with varying Data Length

### i. Energy per bit:

The energy spent to transmit a bit depends on consumption for startup of module along with the power consumed by electronic circuits and radio module. Energy per bit($E_{Bit}$) for a typical wireless module can be given by following formula:

$$E_{bit} = \frac{E_{start}}{L} + \frac{P_{Ele}+P_{RF}}{R_s}\left(1+\frac{H}{L}\right) \qquad \ldots.(2)$$

Where L = length of packet i.e. payload size in bits, H is header size in bits, $E_{start}$ = radio startup transient, $R_s$ is symbol rate and $P_{ELE}$ is power consumed by electronic circuit. Thus energy per bit for RF module used can be estimated as:

$$E_{bit} = \frac{E_{start}}{L} + \frac{P_{Ele}+P_{RF}}{R_s}\left(1+\frac{P+A+PC+CRC}{L}\right) \qquad \ldots.(3)$$

Where, P is preamble of 1 byte, A is address that can vary from 3 to 5 bytes, PC is packet control field of 9-bits and lastly CRC bytes which can be 1 or 2 bytes depending upon the configuration.

### ii. Current consumption with acknowledge:

The average current consumed for transmitting the packet with auto retransmission enabled for lost packet can be given as:

$$I_{avg} = \frac{I_{spi} \times t_{spi\,data-rate} + I_{stby1} \times t_{stby1} + ARC\left\{I_{tx-stl} \times t_{tx-stl} + I_{tx} \times t_{tx} + I_{rx-stl} \times t_{rx-stl} + I_{rx} \times t_{rx} + I_{stby2} \times t_{ARD}\right\}}{t_{spi\,data-rate} + t_{stby1} + ARC\left\{t_{tx-stl} + t_{tx} + t_{rx-stl} + t_{rx} + t_{ARD}\right\}} \qquad \ldots.(4)$$

Where ARC is Automatic retransmission count, $t_{ARD}$ is time delay set between consecutive auto-retransmission for lost packet. Both these parameters can be programmatically configured in RF module to control the energy consumption while communicating data over wireless link.

Typically transmitter settling time($t_{tx-stl}$) and receiver settling time($t_{rx-stl}$) is nearly equal to 130µSec. The time take to transmit the data in the air is given by $t_{tx}$ as:

$$t_{tx} = \frac{packet\ length}{data\ rate} \qquad \ldots.(5)$$

### iii. Startup consumption:

Apart from the component directly accounting for consumption proportional to the data, indirect component that affects the consumption is the power consumed for turning the module on for transmission each time it wakes up. $E_{start}$ is the energy consumed prior the transmitter is ready for transmission. If the data length is too short i.e. Start-up Time ($t_{start}$) > Data Length(L)/Datarate(Rs) the consumption for start-up will dominate the overall the consumption.

The average current consumed during startup of the module from off state to idle state i.e. standby-1 state can be given as:

$$I_{on(Avg)} = \frac{I_{pwr-on} \times t_{pwr-on} + I_{pwr-dn} \times t_{pwr-dn} + I_{pwrup} \times t_{pwrup} + I_{stby1} \times t_{stby1}}{t_{pwr-on} + t_{pwr-dn} + t_{pwrup} + t_{stby1}} \quad \text{.... (6)}$$

Where, $t_{pwr-on} = 100mSec$ take by module for going from off state to power down state, $t_{pwrup} = 1.5mSec$ is time typically taken for crystal oscillator startup, $t_{pwr-dn}$ = time spent by module in power down mode till programmatically PWR_UP bit is set in command register by user and $t_{stby1}$ = time module stays in standby-1 mode prior going to active receive or transmit mode. $I_{on(Avg)}$ is consumed only once it is power on, but if node does ON-OFF switching of RF module then the overhead will incur each time it power-up for communication.

### iv. Comparison with duty cycle:

It is not always the case that node is communicating data, in fact once transmission is done successfully it has to wait for some time for its turn to transmit next chunk of data. So, we considered here two modes wherein either the RF module of the node is turned off when it is working in Duty-Cycle mode or pushed to standby mode when node is working in reconfiguration enabled mode.

In duty cycle concept the node is dropped to asleep for a period of time and when the time comes the node wakes up to act as a sensor node and perform the function as per schedule in sync with nodes in cluster. In such case the node may miss the events which are continuous and may not be suitable for the scenarios to the extent it is intended.

The Figure 5 shows the graph of energy consumed per bit versus time the node is in idle condition. We have compared two scenarios, duty-cycle mode and standby mode. In duty-cycle mode the node is turned off after communication is over while in standby mode, the node is pushed to standby mode after the communication is over. In both the scenarios, for the time node is idle will account for the consumption which leads to increase in energy per bit value. The comparison shows that when the node is working in standby mode its consumption is less until the idle time increases above 11.85ms. At idle time equal to 11.85ms the consumption is equivalent to that of duty cycle mode, after which it increases and turns to be inefficient due to more power loss in idle time.

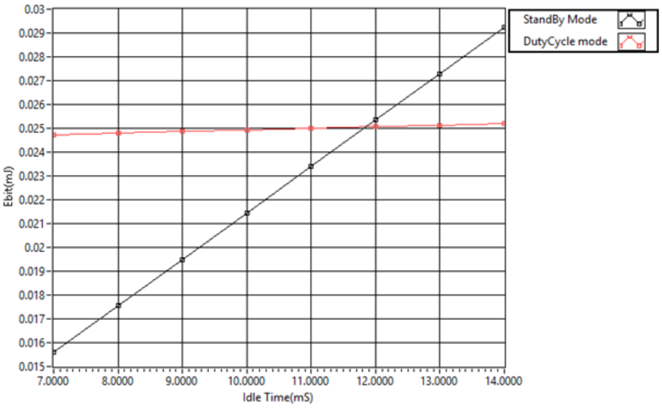

**Figure 5.** Energy per Bit consumed versus idle time

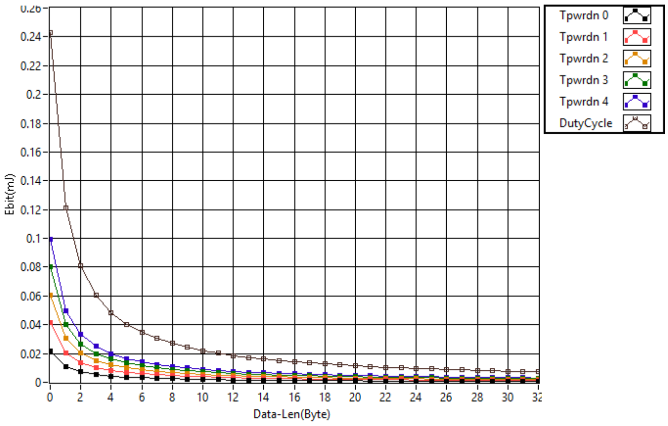

**Figure 6.** Energy per Bit consumed versus Data Length

The Figure 6 shows the graph of energy consumed per bit versus data length, that vary from 1 to 32 bytes per transmission, along with stand-by time indicated by Tpwrdn0 to Tpwrdn4 in idle condition. As data length increases the energy per bit decreases taking into account the idle time between consecutive transmissions. The analysis shows that, for less data to be communicated, it costs more to node in terms of power consumption.

### v. Throughput

Along with Energy per bit, other very important metric that gives measure of reliability with efficiency is throughput of the module. It depicts the packet acceptance rate thus the reliability with energy efficiency. Throughput(T) can be considered as measure of successful transmission by the RF-module for point to point single transmission and can be estimated as:

$$T = \left(\frac{L}{L+P+A+PC+CRC}\right) \times (1 - BER)^{L+P+A+PC+CRC} \quad \ldots.(7)$$

Where BER is Bit Error rate and L+P+A+PC+CRC is total packet size. [17]Bit Error Rate (BER) depend on the receiver sensitivity limit, the average number of errors increases as received power approaches receiver sensitivity limit or lower. Thus the error due to noise at such instances is stochastic in nature. The accurate value of BER can be estimated by iteration of system performance in environment it is deployed into. Typically, BER can be given as:

$$BER = \left(\frac{Number\ of\ Bit\ Errors\ made}{Total\ number\ of\ Bits\ received}\right) \quad \ldots.(8)$$

The Figure 7 and figure 8 shows the plot for throughput with reference to varying data length and effect of packet error rate(PER). The analysis shows that at a given value of packet error rate, the throughput increases to certain extent after then it starts decreasing with increase in packet length. So if estimated PER for the channel is known the optimum value of the packet length can be set to receive better throughput with energy efficiency. The Figure 9 shows the graph for estimated optimal value of data-length at a given packet error rate observed for particular channel. By setting optimal data-length the overall throughput for the communication link can be increased.

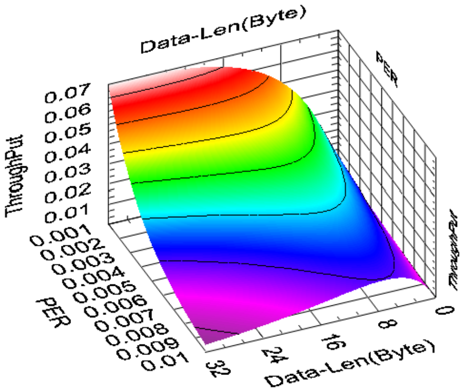

**Figure 7.** Contour Plot for Throughput with respect to change in Data-Length and PER

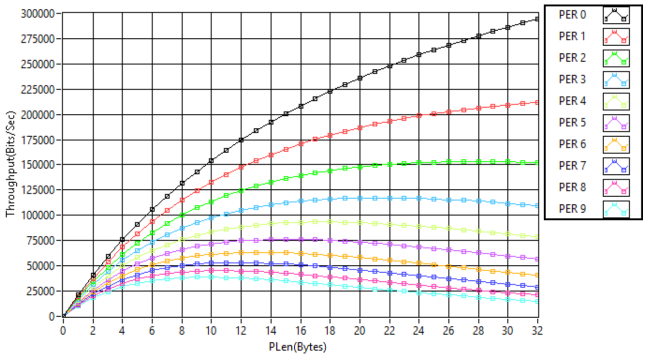

**Figure 8.** Throughput versus Data Length in bytes with varying Packet error rates

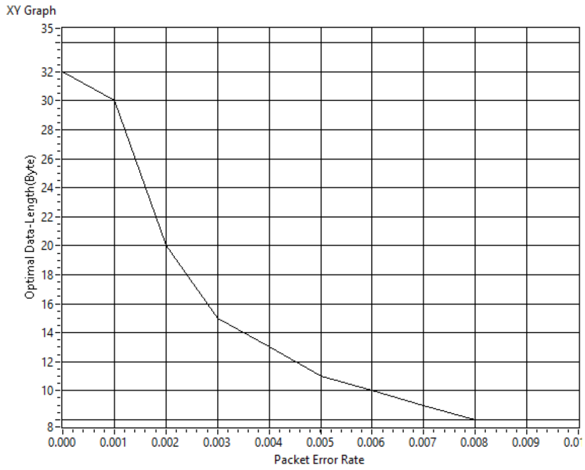

**Figure 9.** Optimal Data-Length versus Packet error rate

## REMOTE RECONFIGURATION ALGORITHM

The energy consumption can be significantly reduced by dynamically optimizing the acquisition and communication setting through software. This technique of the optimization in energy consumption directly deals with altering configuration of the peripherals to reduce the energy consumed in acquiring the data from the sensor, processing it and communicating the same. It requires the algorithmic approach to alter the acquisition settings at definite point of decision to avoid loss of data. For instance, the alternation of sampling rate in the configuration can be made when the data acquired is varying with low frequency so that low frequency sampling do not loss any data.

### a. Frame for reconfiguration command

The command frame is formed of two bytes. The reconfiguration command is meant to know the status and configure the sensor node in particular tuned mode so as to minimize consumption. The parameters that can be tuned are data rate, power setting and retransmission settings for RF module on a sensor node along with MCU modes and ADC settings for sampling rate, channel and reference voltage selection.

The first byte of command specifies what type of command it is and for whom it is issued to i.e. is it for RF module or MSP. Also it specifies what this command is intended to do i.e. read the status of the registers configured or to set the new configurations by writing the command-word (specified in bytes after 1st command) into the address register which is given in command itself.

As given in Table 2, the first-byte can be classified into four categories by the combination of two most significant bits of frame. These bits take the form depending on the access it is meant for i.e. RF module access, MSP430 access for clock or ADC and thirdly it may be for miscellaneous service request related to data, setting, completion status of task or Digital Input Outputs status (DIO). For command issued to access RF module, the last 5 bits takes the form of address of RF-module registers that vary from 0x00 to 0x17 to perform read or write operation. The first-byte can be of 1 byte if it is intended to read the status and can be upto 6 byte(max) if it is supposed to change the settings.

Similarly, for MSP only the 1st byte of command is used to read the functionality configured or to change the functionality related to clock frequency using Digital Controlled Oscillator (DCO) bits or ADC setting.

**Table 2.** Command Frame - 1st Byte

| **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
|---|---|---|---|---|---|---|---|
| 0 0 = NRF24l01+ | | 0 = Read Status<br>1= Write Command | Register address of NRF24l01+ | | | | |
| 0 1 = MSP430 (CPU - DCO) | | | DCO bits:<br>DCO.1 and DCO.0 | | Range Select bits:<br>RSEL.3 to RSEL.1 | | |
| 1 0 = MSP430 (ADC) | | | ADC Sampling Rate<br>0 = 50ksps<br>1 = 100ksps | Ref voltage<br>0 = 1.5V<br>1 = 2.5V | ADC Channels<br>000 = A0<br>001 = A1<br>010 = A2<br>011 = A3<br>100 = A4<br>101 = A5<br>110 = A6<br>111 = A7 or temperature | | |
| 1 1 = Service Request | | 00 0000 = Data Request<br>00 0001 = Request all configuration settings<br>00 0010 = Exit reconfiguration algorithm<br>00 0011 = Reconfiguration complete – Successful<br>00 0100 = Reconfiguration failed – resend<br>00 0101 = 00 1000 = DIO.0 to DIO.3 status<br>00 0111 = Request Estimated Distance between SN and CH<br>00 1000 = Request Estimated PER for channel | | | | | |

Second Byte: If the first byte command is of “Write” type than the 2nd byte will be having the byte that is to be written to NRF24L01+ to change the settings of sensor node. In case, first byte is issued to read the status of the sensor node configuration than the configuration setting will be send in reply to it by the sensor node addressed.

### b. Remote Reconfiguration Algorithm (RRA)

Base-station or sink node is the node where all the sensor-node data are accumulated. There are certain instances during the process monitoring when the reconfiguration may lead to conserve the energy. For instance, when the cluster head wants to go with selective acquisition or else when cluster head finds that process has gone stationary and variations in sensor readings are low enough cluster head may decide to reconfigure sensor node to avoid redundant data acquisition. The solution to save acquisition energy can be derived by taking into consideration the data thresholds and its accuracy. Algorithmically deriving the instant when the variation of the data is less and accuracy required is not of higher significance until the thresholds are nearby. As and when the data required to be very precisely taken for sensitive variations in process than the sampling frequency can be increased.

Under such scenario if process tends to become non-stationary than sensor node may find the need to re-update its configuration by reporting it back to cluster head. Along with that, if the cluster heads are changed or are mobile in the cluster than the reconfiguration of RF-module is inevitable to conserve power by adjusting the radio range to optimal level. The reconfiguration of channel is also required if error rate for channel is more and affecting consumption loss in communicating. In such case PER value is taken into consideration to act upon for the changes in transmission power.

```
Algorithm 1: Remote Reconfiguration Algorithm (RRA) – PER Scenario
Input: Incoming packets from the sensor node
Output: Commands to reconfigure setting in sensor node
For all nodes in cluster do
    Check the PER field value in packet from sensor node
    If (PER > $P_{Threshold}$) then
        $S_i$ = Receive(S); Read configurations of $i^{th}$ node in cluster
        $S_i^*$= Identify the new configuration values
        Send(Invoke Reconfiguration, $S_i^*$)
        while (!Acknowledge);
        Send(Exit-command)
        end
    Process further with the Data received
end
```

**Figure 10.** Remote Reconfiguration Algorithm(RRA) - PER Scenario

```
Algorithm 2: Reconfiguration Service Routine(RSR).
Input: Incoming packets from the Cluster Head(CH)
Output: Acknowledge with reconfiguration completion status to Cluster Head
C=Receive(Service Request command)
If (C == Read configurations) then
    Send(S) to Cluster head
else if(C == write command) then
    write(command) to respective register
    if (reconfiguration == successful) then
        Send(Reconfiguration complete)
    else
        Send(Reconfiguration failed)
    end
while(!Receive(Exit reconfiguration));
end
```

**Figure 11.** Reconfiguration Service Routine(RSR) Algorithm

### c. How to apply Algorithm

Whenever cluster head finds the need for reconfiguration from the sensor node, it initiates Remote Reconfiguration Algorithm (RRA) as shown in figure 10 for scenario when packet errors causing more power loss. During this, the first thing performed is reading of the configuration settings in particular sensor node. After the configurations setting are known the new configuration settings and their commands are identified. The commands for new configuration are sent to the node and after that host will wait to receive the completion command. Once completion status is received the host will send the exit command and return to the main application program.

The sensor node is running the application program with default configurations. It reads the data from the sensors and sends it to the host. In midst, if it receives any data from cluster head than a receive interrupt is generated and it executes Reconfiguration Service Routine (RSR). During this routine it identifies the command type i.e. whether it is read-status type, write-settings type or service request. It executes the appropriate sequence of commands and sends the completion status to the host and resumes task as shown in Figure 11. For communicating the data, the nearby nodes are identified by the transmitting node and the transmission power is adjusted to appropriate level to save power. The algorithm to detect the nodes in nearby vicinity is implemented, to get list of nodes in neighborhood and communicate with them at appropriate power levels.

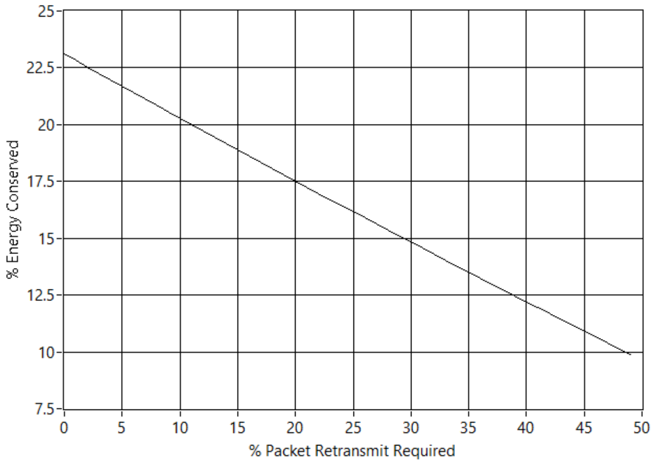

**Figure 12.** % of Energy Conserved versus % Packet Retransmit Required

## RESULTS AND CONCLUSION

By varying the acceptable packet error thresholds, the energy consumptions can be reduced to optimize sensor node lifetime. The Figure 12 shows the graph of % energy conserved with increasing packet error thresholds when applying dynamic reconfiguration over static transmit power settings. Keeping threshold at the said PER of 28.05% for BER of 0.1% with receiver sensitivity of -85dbm 15.36% energy could be conserved over static configurations. So far as acquisition is concerned, about 25.64% can be conserved by switching acquisition to lower sampling rate as compared to higher sampling rate.

The drawback of the algorithm can be overcome by embedding additional intelligence in the node itself i.e. by making node capable of taking decisions for changing the configuration of acquisition hardware along with the RF power setting. In the next stage of extension, addition of the intelligence in node will be worked upon with the decision taking algorithm that intelligently decides the mode of operation on node to reduce power consumption.

## REFERENCES

[1] H. Patel and V. Shah, “A Review on Energy consumption and conservation techniques for Sensor node in WSN,” in *International conference on Signal Processing, Communication, Power and Embedded System*, 2016.

[2] S. Misra and E. Eronu, “Implementing Reconfigurable Wireless Sensor Networks : The Embedded Operating System Approach,” *Embed. Syst. - High Perform. Syst. Appl. Proj.*, pp. 221–232, 2012.

[3] R. P. Narayanan, T. V. Sarath, V. V. Vineeth, and R. P. Narayanan, “Survey on Motes Used in Wireless Sensor Networks: Performance & Parametric Analysis,” *Wirel. Sens. Netw.*, vol. 8, no. April, pp. 51–60, 2016.

[4] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, “Indriya : A Low-Cost , 3D Wireless Sensor Network Testbed Indriya : A Large-Scale Sensor Network Testbed,” pp. 302–316, 2012.

[5] T. H. Teo, S. Member, X. Qian, P. K. Gopalakrishnan, and S. Member, “A 700- W Wireless Sensor Node SoC for Continuous Real-Time Health Monitoring,” vol. 45, no. 11, pp. 2292–2299, 2010.

[6] D.-S. Kim, B.-S. Kim, K.-H. Won, and M.-S. Kang, “A wireless sensor node SoC with a profiled power management unit for IR controllable digital consumer devices,” *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2282–2287, 2010.

[7] J. A. Guevara, E. A. Vargas, A. F. Fatecha, F. Barrero, and S. Member, “Dynamically Recon gurable WSN Node Based on ISO/IEC/IEEE 21451 TEDS,” *IEEE Sens. J.*, vol. 15, no. 5, pp. 2567–2576, 2015.

[8] Jian Lu, H. Okada, T. Itoh, T. Harada, and R. Maeda, “Toward the World Smallest Wireless Sensor Nodes With Ultralow Power Consumption,” *IEEE Sens. J.*, vol. 14, no. 6, pp. 2035–2041, 2014.

[9] P. Cheong *et al.*, “A ZigBee-Based Wireless Sensor Network Node for Ultraviolet Detection of Flame,” vol. 58, no. 11, pp. 5271–5277, 2011.

[10] B. H. Calhoun *et al.*, “Design considerations for ultra-low energy wireless microsensor nodes,” *IEEE Trans. Comput.*, vol. 54, no. 6, pp. 727–740, 2005.

[11] A. Somov, A. Baranov, D. Spirjakin, and R. Passerone, “Circuit Design and Power Consumption Analysis of Wireless Gas Sensor Nodes: One-Sensor Versus Two-Sensor Approach,” *IEEE Sens. J.*, vol. 14, no. 6, pp. 2056–2063, 2014.

[12] Y. E. Krasteva, J. Portilla, E. De La Torre, and T. Riesgo, “Embedded runtime reconfigurable nodes for wireless sensor networks applications,” *IEEE Sens. J.*, vol. 11, no. 9, pp. 1800–1810, 2011.

[13] M. Healy, T. Newe, E. Lewis, and C. C. C. Em, “Wireless Sensor Node Hardware : A Review,” pp. 621–624, 2008.

[14] W. L. Lee, A. Datta, and R. Cardell-Oliver, “WinMS: Wireless sensor network-management system,” no. June, pp. 1–21, 2006.

[15] Texas Instruments, “MSP430G2x53 Datasheet,” no. April 2011, p. 75, 2013.

[16] N. Semiconductor, “nRF24L01+ Single Chip 2.4GHz Transceiver,” *Product Specification*, 2008. .

[17] Nordic Semiconductor, “RF Performance Test Guidelines,” *White Paper*, 2008. .