IJCTA, 9(40), 2016, pp. 405-415 ISSN: 0974-5572

Zigbee Communication

Abdul Rauf EM* and E. Madhusudhana Reddy**

ABSTRACT

Smart grid technology places greater demand on need to develop faster and reliable means of wireless communication. This research project focus on building a small printed circuit board dedicated for wireless communication using Zigbee protocol. Zigbee is the most popular industry wireless mesh networking standard for connecting sensors, instrumentation and control systems. Zigbee, a specification for communication in a wireless personal area network (WPAN), has been a part of the "Internet of things." Zigbee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks.

Keywords: Zigbee, Communication, WPAN, Wireless, communication, networks, architecture

1. INTRODUCTION

Zigbee and IEEE 802.15.4 are low data rate wireless networking standards that can eliminate the costly and damage prone wiring in industrial control applications. The Zigbee standard enhances the IEEE 802.15.4 standard by providing a simple networking layer and standard application profiles that can be used to create interoperable multi-vendor consumer electronic solutions. Zigbee targets the application domain of low power, low duty cycle and low data rate requirement devices.

2. METHODOLOGY AND SCHEMEATIC OF PCB

2.1. Methodology

- Design a PCB (PCB 2) which can wirelessly transfer data.
- Getting PCB-2 build and solder components.
- Establish serial communication between DSP (PCB1) board and computer.
- Establish serial communication between PCB 2 and computer.
- Transfer data between two PCB 2s wirelessly using a computer.
- Establishing wireless communication between DSP boards.

2.2. Schematic of PCB –

Figure 1 shows the Schematic of the PCB designed using Orcad Capture software.

Following are the features available

1. Receives supply voltage from DSP board (PCB 1): The supply (3.3V) to power up PCB2 is taken from PCB 1 through 9 pin D-connector, thus reducing the number of supplies to be used and keeps the assembly compact.

* Department of Electronics Engineering, Y C College of Engineering,Nagpur, Maharashtra, India, *Emails: nehamahankal1@gmail.com, sandip.kakde@gmail.com*

** Department of Electronics Engineering, Rajiv Gandhi College of Engineering, Nagpur, Maharashtra, India, *Email: atish_khobragade@rediffmail.com*

2. Indicating LED to check if board is powered or not.
3. Jumper to reset the module: Shorting the reset jumper, does a physical reset of the device. It temporarily removes supply to the device and reapplies it. Reset pulse must be at least 200 ns wide.
4. Driver circuit to translate logic level to RS 232 voltage level and back: Enable accurate serial communication between two PCBs. A MAX3221 which is a one-driver and one-receiver RS-232 interface device is used for this purpose. The charge pump requires four small 0.1-ìF capacitors for operation from a 3.3-V supply. The MAX3221 is capable of running at data rates up to 250 kbps, while maintaining RS-232-compliant output levels. The IC is made to operate in FORCE ON mode so that when it is connected to DSP, serial communication between the two gets activated, else we will require an external supply to activate serial communication.
5. Hardware provision to activate sleep or active mode:Allows the module to sleep and wake according to the state of the Sleep_RQ pin (pin 9). Pin sleep mode is enabled by setting the SM command to 1. When Sleep_RQ is asserted (high), the module will finish any transmit or receive operations and enter a low power state. . The module will wake from sleep when the Sleep_RQ pin is de-asserted (low).
6. Decoupling capacitors: Eliminate the high frequency ripples in the supply voltage. Since parasitic inductance is typically high for large capacitors, so small and large capacitors are placed together in parallel to fully cover circuit bandwidth. High frequency tantalum capacitors are used as large capacitors though they are polarized because if we use other capacitors we will have to keep several units in parallel to achieve desired capacitance value.
7. Data transfer indicator: LED connected to pin 6 of Xbee act as indicator of data transmission. If the RSSI PWM functionality is enabled (P0 command), when the module receives data, the output of this pin is RSSI value of the packet, this output can be used to switch on LED during transmission.

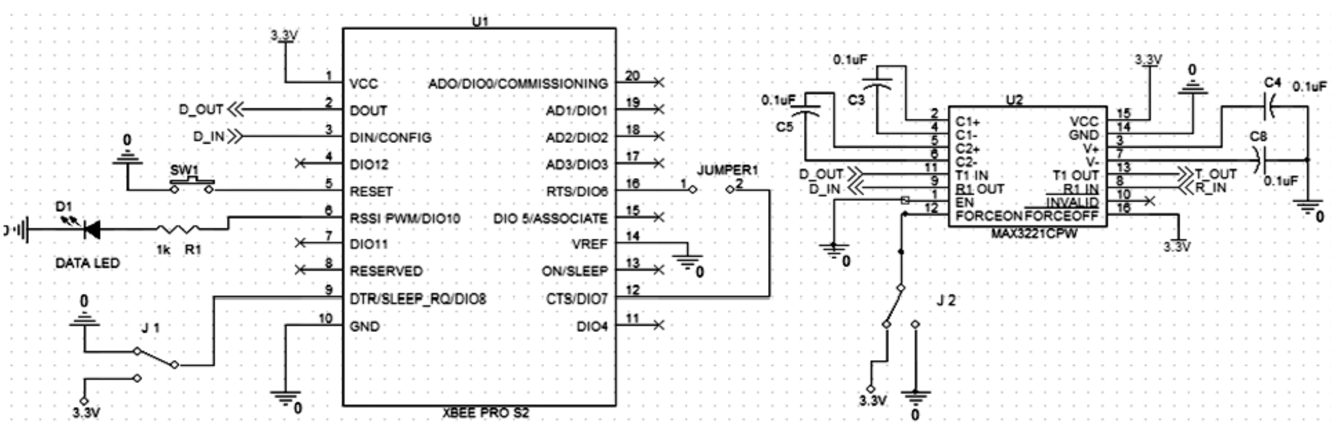


Figure 1: Schematic of PCB-2

3. PCB DESIGN AND DATA TRANSFER

3.1. Components on PCB

The two main components on PCB are Xbee pro S2B and max 3221.

3.1.1. Xbee pro S2B

The XBee-PRO RF Modules are designed to operate within the Zigbee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices. The modules operate within the ISM 2.4 GHz frequency band. Main features of this IC are:

- Transmission distance up to 1500m outdoor and up to 90m indoor.
- RF data rate 250 Kbits /second.
- In built antenna for data transmission and receiving.
- 2 pin universal asynchronous receiver/transmitter (UART) serial communication.

3.1.2. MAX 3221

This IC translates the logic voltage levels to RS 232 voltage levels enabling the serial data transmission between PCB1 and PCB2 for a longer distance. The power block of this driver increases, inverts, and regulates voltage at V+ and V- pins using a charge pump that requires four external capacitors. Auto-power-down feature for driver is controlled by FORCEON and FORCEOFF inputs. The driver is made to operate in FORCE ON mode. A jumper (J2) is provided to enable auto-power-down feature as and when required.

PCB layout: Layout of the PCB is designed using Orcad Layout plus software. Individual component footprints were designed and they were assembled appropriately to get the desired PCB design. Figure 2 shows the PCB-2 layout.

PCB specifications: Type: 2 layer, Board thickness- 1.6 mm, Copper thickness-35 microns, Size: 5.20 X 5.20 cm

Specifications: HASL Finish, Solder Mask- green, Legend- white.

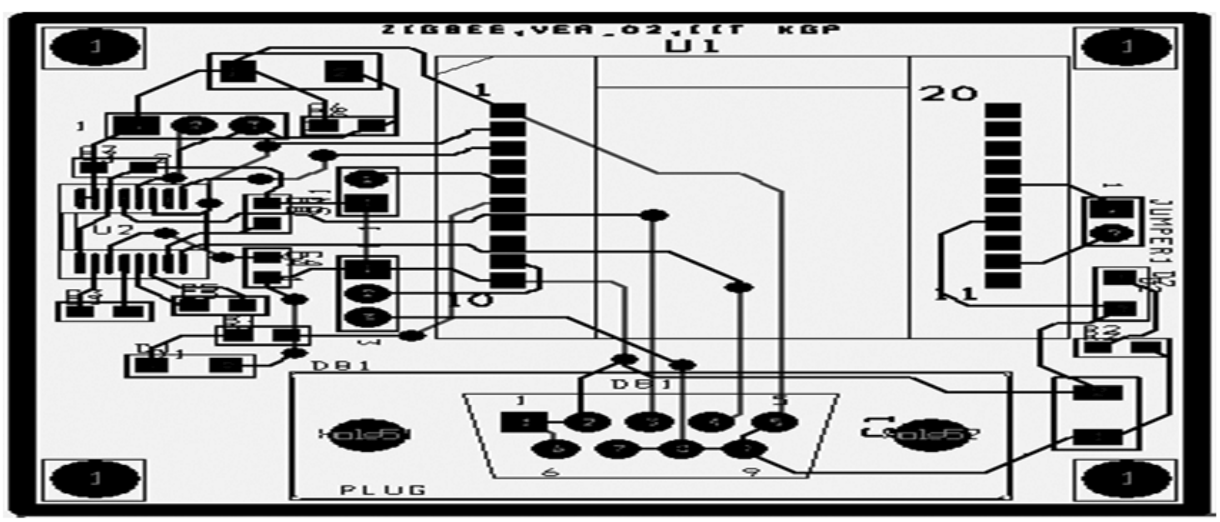


Figure 2: PCB-2 layout

3.2. Serial data transfer between PC and PCB1 –

The serial communication interface of DSP lf 2407 has the following features:

- Programmable bit rates to over 65,000 different speeds through a 16-bit baud select register.
- Programmable data word length from one to eight bits.
- Programmable stop bits of either one or two bits.
- Transmitter and receiver can be operated by interrupts or by polling using status flags.
- Separate enable bits for transmitter and receiver interrupts (except break).
- NRZ (non-return-to-zero) format.

Connection was made between the PC and lf2407 using 9 pin D-plug. Rin of the PCB1 was connected to T_out of PC and vice-versa. The connection is described in table 1

Table 1
D - plugs configuration

Function	*PCB1 D-pug pin no.*	*PC D-plug pin no.*
R in	3	2
T out	2	3
Ground	5	5

3.3. Transmitting and receiving data

In PC we can use “hyper terminal” or any other serial communication software to receive and transmit data serially. DSP programme should do the following:

Initialization:

- Appropriate number of parity bits, stop bits and length of data are selected.
- Transmitter and receiver are enabled and the module is reset.
- Interrupts are disabled to send and receive data by polling.
- Appropriate Baud rate is selected by configuring Baud rate registers.

Steps to transmit data:

- Check if the buffer is ready to accept data from the DSP if ready transfer data to transmit buffer else delay sending for the next time.
- Data from the transmit buffer is shifted onto the transmit pin, one bit at a time.

Steps to receive data:

- Check if any new data has arrived through serial interface.
- If new data arrived, load the data from the receiver buffer and transfer it to the desired memory location.

Care should be taken to ensure that both DSP and PC serial communication interface are set with sane baud rate, parity, stop bits, etc.

3.4. Serial data transfer between PC and PCB2

Serial communication between PC and PCB2 is done using X-CTU software of Digi international. Select appropriate baud rate, stop bits, parity and data bits before communication. R_in of the PCB1 was connected to T_out of PC and vice-versa. Pin configurations of respective D- plugs are shown in table 2

Table 2
D - plugs configuration

Function	*PCB2 D-pug pin no.*	*PC D-plug pin no.*
R in	2	2
T out	3	3
Ground	5	5

Zigbee defines three different device types: coordinator, router, and end device.

A coordinator has the following characteristics:

- Selects a channel and PAN ID to start the network
- Can allow routers and end devices to join the network
- Can assist in routing data
- Cannot sleep—should be mains powered
- Can buffer RF data packets for sleeping end device children.

A router has the following characteristics:

- Must join a Zigbee PAN before it can transmit, receive, or route data
- After joining, it behaves similar to that of a coordinator.

An end device has the following characteristics:

- Must join a Zigbee PAN before it can transmit or receive data
- Cannot allow devices to join the network
- Must always transmit and receive RF data through its parent. It can't route data.
- Can enter low power modes to conserve power and can be battery-powered.

Starting a Coordinator:

1. Set SC and ID to the desired scan channels and PAN ID values.
2. If SC or ID is changed from the default, issue the WR command to save the changes.
3. The permit joining attribute on the coordinator is configured with the NJ command.

After starting a coordinator (that is allowing joins), the following steps will cause a router to join the network:

1. Set ID to the desired 64-bit PAN ID, or to 0 to join any PAN.
2. Set SC to the list of channels to scan to find a valid network. Once the router has joined, the OP and CH commands will indicate the operating 64-bit PAN ID and channel the router joined.
3. The MY command will reflect the 16-bit address the router received when it joined.

If a coordinator or router is there which allows joining then starting of end device is similar to that of a router. Figure 3 shows the serial data transfer between PC and PCB-2

4. ESTABLISHING WIRELESS COMMUNICATION

All Zigbee devices have two different addresses, a 64-bit and a 16-bit address. The 64-bit address is a unique device address assigned during manufacturing. A device receives a 16-bit address when it joins a Zigbee network. These addresses are used during sending and receiving data. The Xbee modules support both transparent and API (Application Programming Interface) serial interfaces. When operating in transparent mode, the modules act as a serial line replacement. All UART data received through the DIN pin is queued up for RF transmission. When RF data is received, the data is sent out through the DOUT pin. The module configuration parameters are configured using the AT command mode interface. API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and

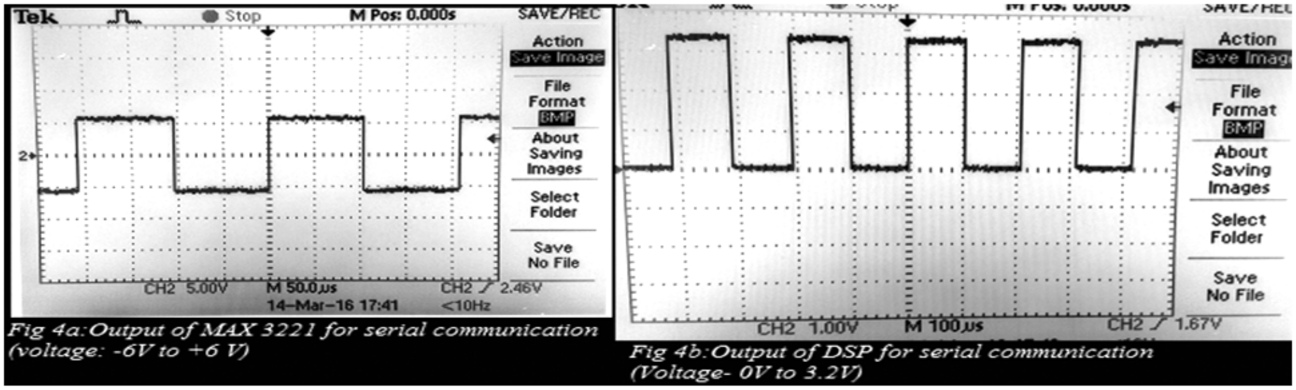


Fig 4a:Output of MAX 3221 for serial communication (voltage: -6V to +6 V)

Fig 4b:Output of DSP for serial communication (Voltage- 0V to 3.2V)

Figure 3: Output of DSP and MAX 3321 for serially transferring binary number 01010101 between PC and PCB-2.

leaving the module is contained in frames that define operations or events within the module. The API operation facilitates many operations such as:

- Transmitting data to multiple destinations without entering Command Mode.
- Receive success/failure status of each transmitted RF packet.
- Identify the source address of each received packet.

Frames supported by the algorithm made for this project are

1. AT command queue parameter value: Table 3 shows the AT command queue parameter value. It allows module parameters to be queried or set. The queued parameter values are not applied until either the “AT Command” (0x08) API type or the AC (Apply Changes) command or a flash request is issued.
2. Transmit Request: Table 4 shows the transmit data parameter. A Transmit Request API frame causes the module to send data as an RF packet to the specified destination.
3. Zigbee receive packet: When the module receives an RF packet (Table 5), it is sent out the UART using this message type.
4. AT command response: In response to an AT Command message (Table 6), the module will send an AT Command Response message.

In future if new frames are to be added, slight modifications can be made in the below mentioned ZIGBEE_TRANSMIT and SERIAL_COMMUNICATION algorithms (modification to be made in that part where decision regarding frame type is made). The configuration mode of -XCTU can be used to select appropriate networking, addressing, security and serial communication parameters from PC while from DSP we can use below mentioned algorithm to set the same .Xbee modules are operated in API mode. To select API mode:

XCTU -> Update firmware -> (Select Product family, function set, firmware version) -> finish

Selection of API mode can done using only X-CTU software, it can’t be done from DSP. Ensure that all modules used have same firmware version, else may lead to compatibility issues. Use console mode to connect Xbee module to PC, generate different frames (transmit, AT command, etc.), send data and receive data from other modules.

The PCB developed enables the user tosend and receive 8 byte data using Zigbee protocol, read and modify module parameter values like PAN ID, Baud rate,etc. Connection between PCB-1 and PCB-2 can be made using D plugs.Pin configurations of respective D- plugs are shown below:

Function	*PCB2 D-pug pin no.*	*PCB1 D-plug pin no.*
R in	2	3
T out	3	2
Ground	5	5

R_in of the PCB1 is connected to T_out of PCB-2 and vice-versa.

Most of the software part is driven by three functions:

1. SETUP_FRAME: It initializes the control variables and sets up data to be transmitted in appropriate memory locations. Inside this function we need to give input regarding what type of frame we wish to transmit.

Table 3
AT command queue parameter value

Start Delimiter		Example:0x7E	
Length	LSB	0x00	Number of bytes between the length and the checksum
	MSB	0x05	
Frame-specific Data	Frame Type	0x09	
	Frame ID	0x01	Identifies the UART data frame for the host to correlate with a subsequent, ACK(acknowledgement). If set to 0, no response is sent.
	AT command	0x42	Command Name - Two ASCII characters that identify the AT Command.
		0x44	
	Parameter value	0x07	If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
Checksum		0x68	0xFF - the 8 bit sum of bytes from Frame type to this byte.

Table 4
Transmit data

Start Delimiter		0x7E	
Length	MSB	0x00	Number of bytes between the length and the checksum
	LSB	0x16	
Frame Specific data	Frame type	0x10	
	Frame ID	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to 0, no response is sent.
	64-bit Destination Address	0x00	Set to the 64-bit address of the destination device. The following addresses are also supported: 0x0000000000000000 - Reserved 64-bit address for the coordinator 0x000000000000FFFF - Broadcast address
		0x13	
		0xA2	
		0x00	
		0x40	
		0x0A	
		0x01	
		0x27	
	16-bit Destination Network Address	0xFF	Set to the 16-bit address of the destination device, if known. Set to 0xFFFE if the address is unknown, or if sending a broadcast.
		0xFE	
	Broadcast Radius	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.
	Options	0x00	0x01 - Disable retries and route repair, 0x20 - Enable APS encryption (if EE=1), 0x40 - Use the extended transmission timeout
	RF Data	0x54	Data that is sent to the destination device
		0x78	
		0x44	
		0x61	
		0x74	
		0x61	
		0x30	
		0x41	
Checksum		0x13	0xFF - the 8 bit sum of bytes from Frame type to this byte.

Table 5
Receive packet

Start Delimiter		0x7E	
Length	MSB	0x00	Number of bytes between the length and the checksum
	LSB	0x14	
Frame Specific data	Frame type	0x90	
	64-bit Source Address	0x00	64-bit address of sender. Set to 0xFFFFFFFFFFFFFFFF (unknown 64-bit address) if the sender's 64-bit address is unknown.
		0x13	
		0xA2	
		0x00	
		0x40	
		0xE6	
		0x78	
		0x24	
	16-bit Source Network Address	0x00	16-bit address of sender
		0x00	
	Options	0x02	0x01 - Packet Acknowledged, 0x02 - Packet was a broadcast packet, 0x20 - Packet encrypted with APS encryption, 0x40 - Packet was sent from an end device (if known)
	RF Data	0x00	Received RF data
		0x00	
		0x00	
		0x00	
		0x00	
		0x00	
		0x00	
		0x00	
Checksum		0xA1	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

Table 6
at command message

Start Delimiter		Example:0x7E	
Length	LSB	0x00	**Number of bytes between the length and the checksum**
	MSB	0x05	
Frame-specific Data	Frame Type	0x88	
	Frame ID	0x01	Identifies the UART data frame being reported. Note: If Frame ID = 0 in AT Command Mode, no AT Command Response will be given.
	AT command	0x42	Command Name - Two ASCII characters that identify the AT Command.
		0x44	
	Command status	0x00	0 = OK, 1 = ERROR, 2 = Invalid Command, 3 = Invalid Parameter , 4 = Tx Failure register.
	Command Data	0xFF	Register data in binary format.
Checksum		0xF0	0xFF - the 8 bit sum of bytes from Frame type to this byte.

2. ZIGBEE_TRANSMIT: It checks if there is any new data to transmit or if the transmission of current data frame completed. Based on type of frame it moves pointer to corresponding memory location and resets pointer once transmission is over. It also calculates the checksum, which is used to validate the received and transmitted data. If the data frame is of type "AT command queue parameter value" type, then this function issues an flash command so that the modified parameter writes its value to permanent memory location so that even after a power down, modified value will be retained. Figure 4 shows the flow chart for zigbee transmission

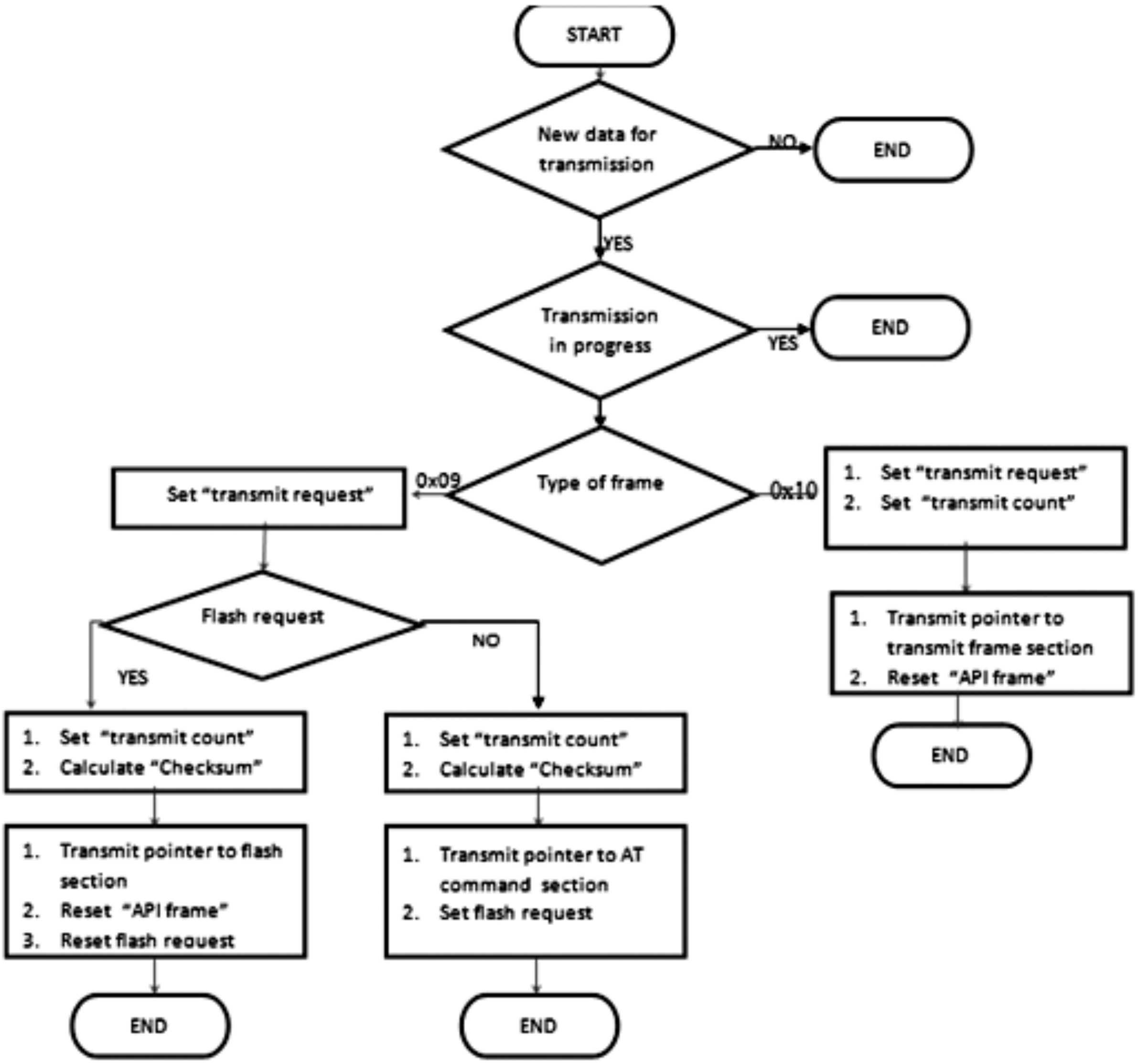


Figure 4: Flow chart for ZIGBEE_TRANSMIT

3. SERIAL_COMMUNCATION: It takes care of the data transfer between PCB-1 and PCB-2. It has two subparts, one to handle incoming data and other to handle outgoing data. Figure 5 and 6 shows the flow chart for handling incoming and transmitting data

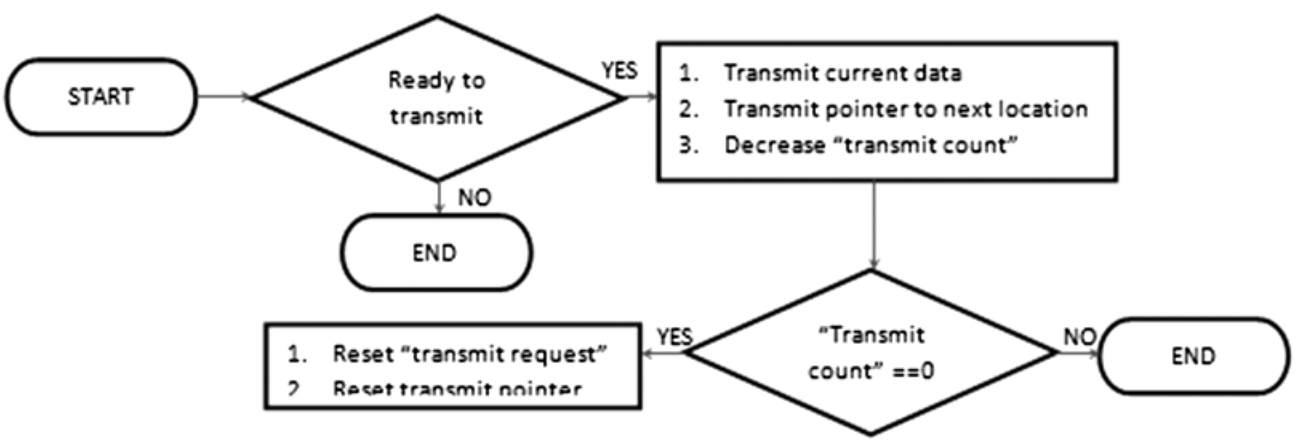


Figure 5: Flow chart for the sub section handling transmitting data

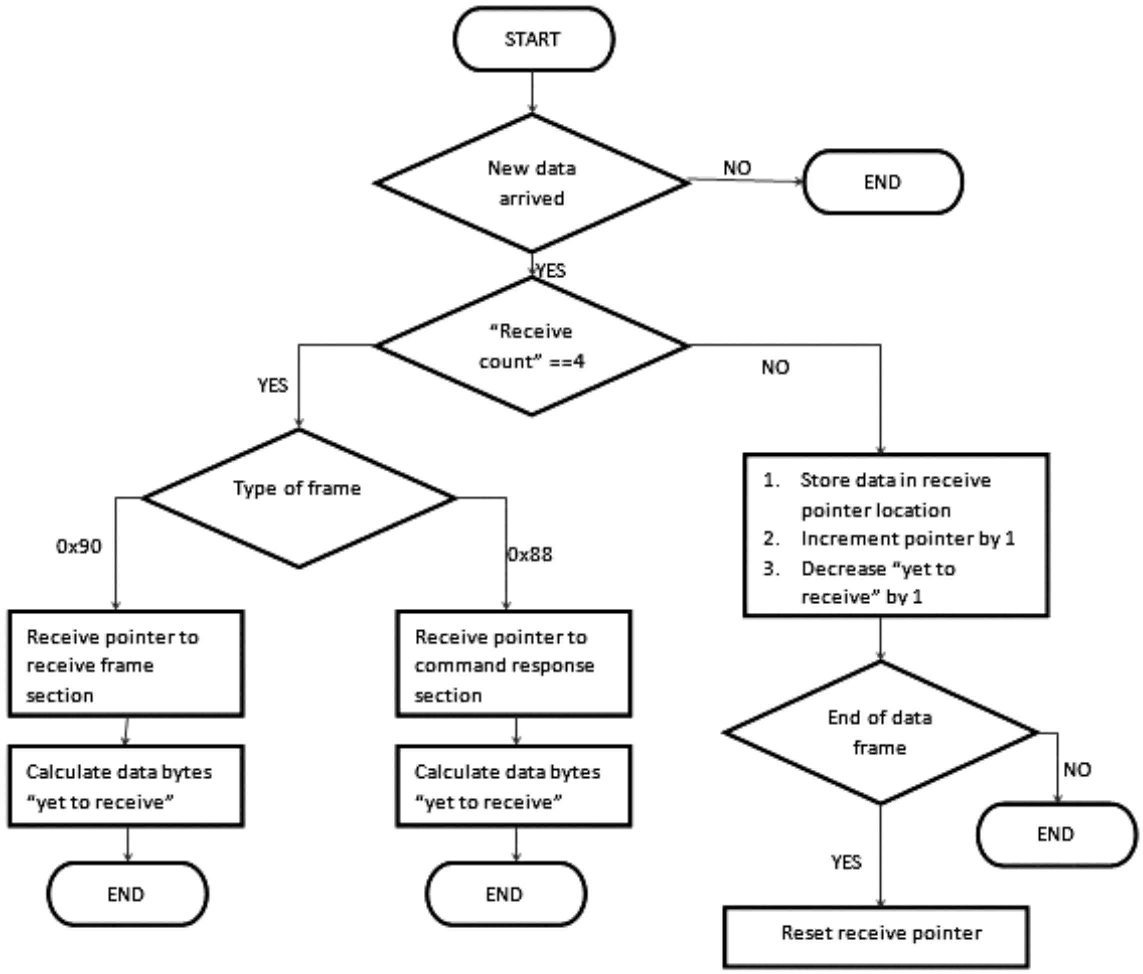
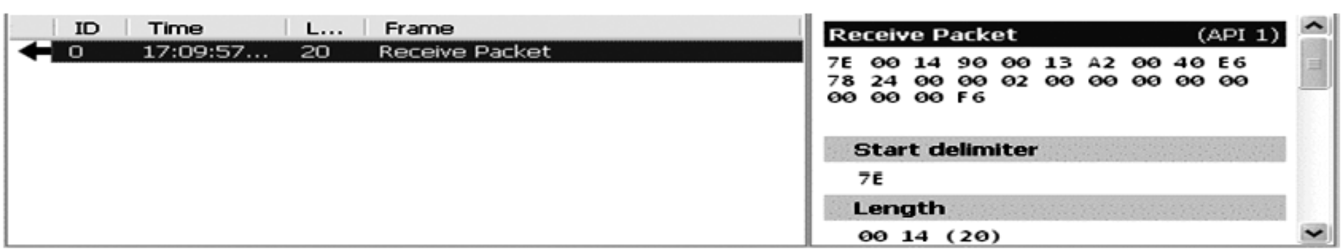


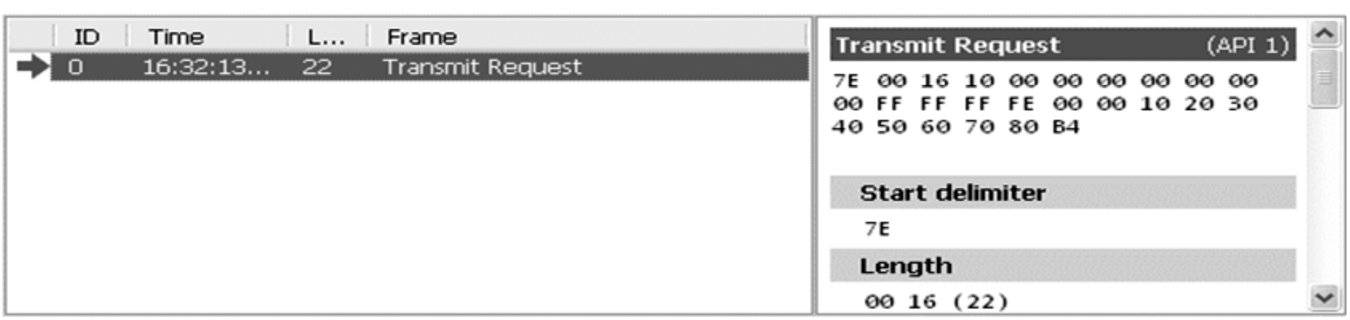
Figure 6: Flow chart for the sub section handling incoming data

Result:

Received frame from PC to DSP lf2407



Transmitted frame from DSP lf2407 to PC



Transmitted data from DSP lf 2407

0x000003A0	0x007E	0x0000	0x0016
0x000003A3	0x0010	0x0000	0x0000
0x000003A6	0x0000	0x0000	0x0000
0x000003A9	0x0000	0x0000	0x00FF
0x000003AC	0x00FF	0x00FF	0x00FE
0x000003AF	0x0000	0x0000	0x0000
0x000003B2	0x0000	0x0000	0x0000
0x000003B5	0x0000	0x0000	0x0000
0x000003B8	0x0000	0x00F4	0x42F7
0x000003BB	0x3A4D	0x14FE	0xAF30
0x000003BE	0x7F20	0xF2A9	0x1D5D
0x000003C1	0x6D9F	0x657B	0xFF08

Received data from PC to DSP To lf2407

0x000003B2	0x0000	0x0000	0x0000
0x000003B5	0x0000	0x0000	0x0000
0x000003B8	0x0000	0x00F4	0x007E
0x000003BB	0x0000	0x0014	0x0090
0x000003BE	0x0000	0x0013	0x00A2
0x000003C1	0x0000	0x0040	0x00E6
0x000003C4	0x0078	0x0027	0x005C
0x000003C7	0x00C3	0x0002	0x0010
0x000003CA	0x0020	0x0030	0x0040
0x000003CD	0x0050	0x0060	0x0070
0x000003D0	0x0080	0x0094	0x0369
0x000003D3	0xA1EE	0xEE21	0x6081
0x000003D6	0xEE0A	0x487D	0x68B4

Received data between two DSP lf2407

0x000003B8	0x0018	0x0050	0x007E
0x000003BB	0x0000	0x0014	0x0090
0x000003BE	0x0000	0x0013	0x00A2
0x000003C1	0x0000	0x0040	0x00E6
0x000003C4	0x0078	0x0027	0x005C
0x000003C7	0x00C3	0x0002	0x0011
0x000003CA	0x0012	0x0013	0x0014
0x000003CD	0x0015	0x0016	0x0017
0x000003D0	0x0018	0x0030	0x0369
0x000003D3	0xA1EE	0xEE21	0x6081
0x000003D6	0xEE0A	0x487D	0x68B4

Transmitted data from DSP lf2407 to another DSP lf2407

0x3a0

0x000003A0	0x007E	0x0000	0x0016
0x000003A3	0x0010	0x0000	0x0000
0x000003A6	0x0000	0x0000	0x0000
0x000003A9	0x0000	0x0000	0x00FF
0x000003AC	0x00FF	0x00FF	0x00FE
0x000003AF	0x0000	0x0000	0x0011
0x000003B2	0x0012	0x0013	0x0014
0x000003B5	0x0015	0x0016	0x0017
0x000003B8	0x0018	0x0050	0x02F7
0x000003BB	0x3A6D	0x10FE	0x2FB4
0x000003BE	0x7C20	0xF2A9	0x1D5D

5. CONCLUSION

The intention of this work was to build a small printed circuit board dedicated for wireless communication using Zigbee protocol. Zigbee is the most popular industry wireless mesh networking standard for connecting sensors, instrumentation and control systems. Zigbee, a specification for communication in a wireless personal area network (WPAN), has been a part of the "Internet of things." Zigbee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks. The next step of this work is to integrate Zigbee wireless communication hardware to PCB-1 itself instead of using a separate PCB (PCB-2).

REFERENCES

[1] XBee®/XBee-PRO® ZB RF Modules user manual

[2] TMS320LF/LC240x DSP Controllers, Reference Guide, Texas Instruments.

[3] MAX3221 3-V to 5.5-V RS-232 Line Driver and Receiver data sheet, Texas Instruments

[4] ZigBee specification version 2006, ZigBee document 064112, 2006

[5] Dust network Inc. http://dust-inc.com/flash-index.shtm

[6] Motes, smart dust sensors, wireless sensor networks., http://www.xbow.com/

[7] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, " Efficient in-network movingobject tracking in wireless sensor networks", IEEE Trans. Mobile Computing , 5(8):1044–1056, 2006

[8] Y.-C. Tseng and M.-S. Pan, "Quick covergecast in ZigBee/IEEE 802.15.4tree-based wireless sensor networks", In Proc. of ACM Int'l Workshopon Mobility Management and Wireless Access (MobiWAC), 2006

[9] P. Rohitha, P. Ranjeeth Kumar, N. Adinarayana and T. Venkat Narayana Rao, "Wireless Networking througg Zigbee Technology", International Journal of Advance Research in Computer Science and Software Engineering, ISSN : 2277 128X, Vol.2 Issue 7, 2012