Rohit N Singh, M.Chennakesavulu / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue4, July-August 2012, pp.2256-2263

Remote Radar Data Acquisition And Control Using Cdma Rf Link

Rohit N Singh*, M.Chennakesavulu**
*(Student, Department of E.C.E, RGMCET, Nandyal)
** (Asst. Professor, Department of E.C.E, RGMCET, Nandyal)

ABSTRACT

RADAR SEEKER is used in missile system for detection and tracking a target. It will be integrated in the nose cone of the missile. During testing and launching of the missile all necessary important parameters and status of the seeker are collected for analysis to ascertain the health of the Seeker. Similarly the configuration and control of the seeker is to ensure the proper mode of operation of seeker.

During launch campaign the technicians and engineers cannot go near the missile and the seeker Health, but the knowledge of seeker health is very much necessary for the launch campaign, so a remote Control and data acquisition system should be there to confirm the health of the seeker either through wired serial communication link or using RF Link.

The Aim of this project is to Design and Develop a "Remote Radar Data Acquisition and Control using CDMA RF Link " and to Test with an existing radar seeker.

Keywords – **RADAR SEEKER, QPSK, CORTEX –M3 (ARM Processor), SPZB260 (ZIGBEE Module),Costas loop.**

I. INTRODUCTION

. **Radar** (RAdio Detection And Ranging) is an object-detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. It can be used to detect aircraft, ships, spacecraft, guided missiles, motor vehicles, weather formations, and terrain. The **seeker** is a homing system perceives the target with its own radar , extracts tracking data from the received signal, and computes its own steering commands. As it closes on the target, a fixed angular error at the missile results in a decreasing linear error, providing the higher accuracy characteristic of homing guidance. An **active radar seeker** is basically a tracking Radar whose antenna is mounted on a stabilized platform so as to provide necessary isolation of the antenna from the body motion of the missile. Enabling the antenna to keep tracking the target and generating signals which are used in terminal guidance of the missile.

CDMA (Code Division Multiple Access) is a spread spectrum multiple access technique. Spread spectrum modulation was originally developed for military applications, where resistance to jamming is of major concern. A spread spectrum technique spreads the bandwidth of the data uniformly for the same transmitted power[1]. A spreading code is a pseudo-random code. These are studied and performance expressions are derived and confirmed by computational simulation using MALAB SIMULINK. The spread spectrum technique which is present in SPZB260 Zigbee module, is used in order to establish RF communication and it is controlled by ARM Cortex M3 Processor(LPC1768).

Section2 provides a brief description of Spread Spectrum Modulation scheme using QPSK, SPZB260 and ARM Cortex M3. Section 3 gives block diagram of Transmitter and Receiver. The procedure to implement transmitter and receiver using Simulink are explained in this section. Section 4 provides simulation results of transmitter-receiver, which are supporting the theory provided in the earlier sections in Simulink. Finally the work is concluded in section 5 and the scope for future work is explained.

II. GENERAL STRUCTURE OF THE SYSTEM

A. Definition of Spread Spectrum

Spread spectrum is a means of transmission in which the data sequence occupies a bandwidth in excess of the minimum bandwidth necessary to send it. The spectrum spreading is accomplished before transmission through the use of a code that is independent of data sequence. The same code is used in the receiver to despread the received signal so that the original data sequence may be recovered[1][2].

B. Working of DSSS

A conceptual diagram of DSSS system is given in fig 1. At the transmitter the digital binary information or data d(t) having a source bit of rate of $f_b = 1/T_b$ Where(f_b is bit rate and T_b is the bit duration) is XORed with spreading signal c(t) is a pseudonoise (PN) signal having chip rate of $f_c = 1/T_c$ (f_c is chip rate and T_c is the pulse duration)[2]. Where $f_c \gg f_b$

The data stream entering the modulator is converted by a serial to parallel converter into two separate data streams.

One stream, I(t) is in the phase and other Q(t), is quadrature phase ,. After obtaining the inphase and Quadrature signals ,we need to do modulation for the transmission of the signal. The inphase signal is multiplied by a carrier cosine wave and Quadrature signal is multiplied by sin signal. The both I(t) and Q(t) signal are summed to produce the transmitting signal with four phase signal.

In the Receiver the received signal or the modulated signal is first demodulated and then despread. So, for demodulation we use Costas loop. Costas loop is used to track the carrier signal and phase of modulated signal. It produces two output one is Inphase and the other quadrature phase. This two signal are passed through the parallel to serial convertor to obtain spreading signal. By using matched filter as despread we can obtain required binary data.

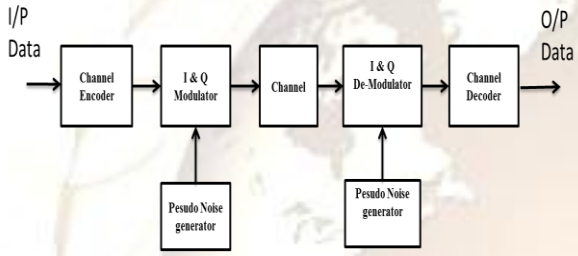

Figure 1: Block diagram of Spread Spectrum Communication system

C. Pseudo Noise Sequence

The types of spreading sequences are Gold sequences, maximum-length sequences, kasami sequences or walsh sequences. pseudo noise code generators are periodic in that the sequence that is produced repeats itself after some period of time. Such a periodic sequence is portrayed in fig 2. The Best known , best described PN sequences are maximal length. The generator contains type D flip-flops and is connected so that each data input except D0 is the input of the preceding flip-flop. Not all Q flip flop outputs need be connected to parity generator . the number of flip flop L and selection of which flip flop outputs are connected to parity generator determines the length and characteristics of the generated PN sequences. When the code is generated by maximum-length sequence , the value is $2^n - 1$, where n is the number of stages in the code generator[2].

$$L = 2^n - 1$$

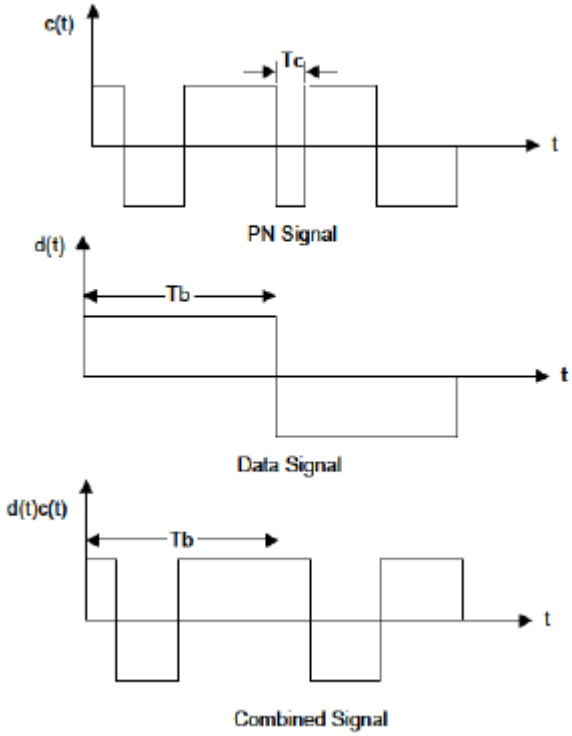

Figure 2: Wave form of Pseudo Random generator

Autocorrelation:

The auto correlation function for the periodic wave is defined as number of agreements less number of disagreements in a term by term comparison over one full period of sequence with cyclic shift (position τ) of the sequence itself:

$$Ra(\tau) = \int_{-\infty}^{\infty} f(t).f(t-\tau).dt$$

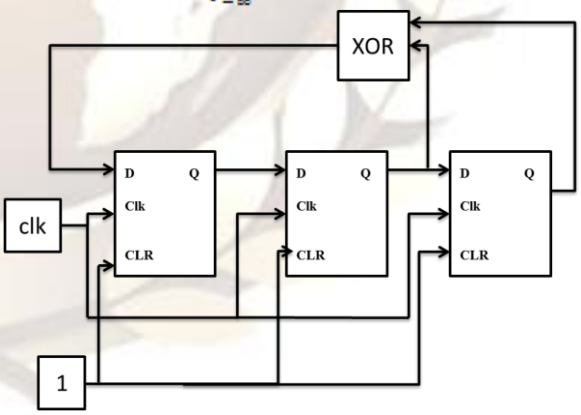

Figure 3: Block diagram of Pseudo Random generator

D. Direct Sequence QPSK

In QPSK, the data stream Inphase and Quadrature phase , with each stream having a symbol rate equal to half that of the incoming bit . Both I and Q are separately applied to multipliers. The Inphase multiplier is the carrier signal sinwt and Quadrature multiplier is the carrier signal

coswt. The I multiplier output signal has phase 90 and 270 degrees and Q multiplier output signal has phase 0 and 180 degrees. Figure 4 shows a typical QPSK waveform in the time domain[4].

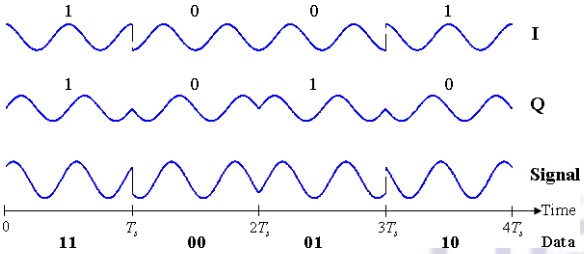

Figure 4: Wave form In phase and quadrature of QPSK modulation

E. Costas Loop

Costas loop is use for carrier recovery and phase detection which is used as demodulation circuit.

In Costas Loop the incoming signal is mixed with the output of the VCO, both before its phase is shifted and after its phase is shifted by 90 degrees. These two outputs are then filtered, multiplied together, filtered again and to control the frequency of the voltage controlled oscillator. The decoded spreading data stream can be taken from the output of the mixer output[1].

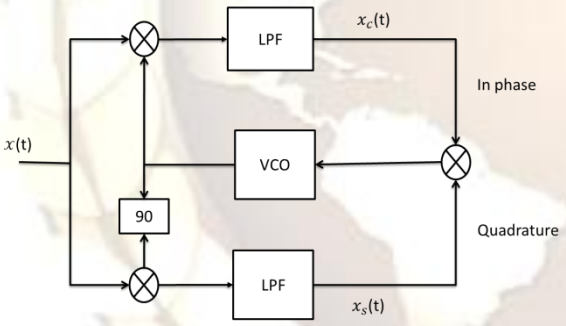

Figure 5: Block diagram of Costas Loop Carrier and phase recovery

$$x_c(t) = A\cos\psi(t) + n_c(t)$$

$$x_s(t) = A\cos\psi(t) + n_s(t)$$

When these two outputs are multiplied together, the product is

$$x_c(t)x_s(t) = \frac{A^2}{2}\sin\psi(t) + n_{eq}(t)$$

F. Hardware Requirement

Here we use the Arm Cortex M3 Processor to Control the Spread Spectrum Module (i.e SPBZB260) via SPI mode Here, we use two SPZB260 and two LPC1768.

The first pair is connected to seeker via serial communication (UART) via LPC1768 which acts as client and other pair is connected to the system . were command and data sent and receive using GUI via serial communication(UART). Which is also called basestation. The following fig 6 show the the CDMA communication between the seeker and user.

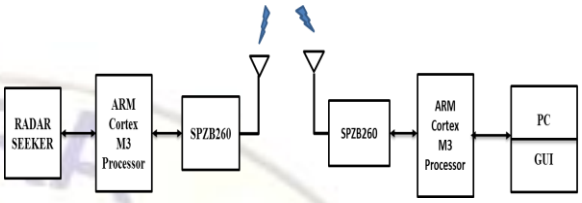

Figure 6: Block diagram of CDMA TransRecevier system

i) SPZB260

The SPZB260 integrates a 2.4 GHz, IEEE802.15.4- compliant transceiver. SPZB260 exposes access to the EmberZnet API across a standard SPI module allowing application to develop on a host platform.

Some features of SPZB260 Zigbee Transceiver are as follows[6]:

- Integrated 2.4 GHz, IEEE 802,15,4-compliant transceiver:
  - 3 dBm nominal TX output power
  - -95 dBm RX sensitivity
  - + 5 dBm in boost mode
  - RX filtering for co-existence with IEEE 802.11g and Bluetooth devices
  - Integrated VCO and loop filter
- Integrated IEEE 802.15.4 PHY and MAC
- Controlled by a standard serial line for an easy interface of host microcontrollers (SPI)
- Embedded flash and integrated RAM for program and data storage
- On board 24 MHz stable crystal
- Integrated RC oscillator (typ. 10 kHz) for low power operation
- 1 µA power consumption in deep sleep mode
- Watchdog timer and power-on reset
- Pins available for non-intrusive debug interface (SIF)
- Single supply voltage 2.1 to 3.6 Vdc
- Available link and activity outputs for external indication / monitor
- CE compliant (a)
- FCC compliant (FCC ID:S9NZB260A) (a)

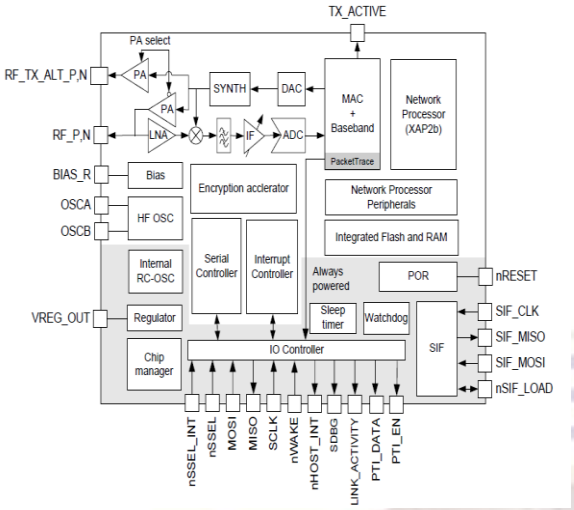

Figure 7: Block diagram of SPZB260

ii) ARM Processor

The LPC1768 is an 32bit microcontroller .The peripheral complement of the LPC1768 includes up to 512 kB of flash memory, up to 64 kB of data memory, Ethernet MAC, a USB interface that can be configured as either Host, Device, or OTG, 8 channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I2C interfaces, 2-input plus 2-output I2S interface, 8 channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, 4 general purpose timers, 6-output general purpose PWM, ultra-low power RTC with separate battery supply, and up to 70 general purpose I/O pins[5].

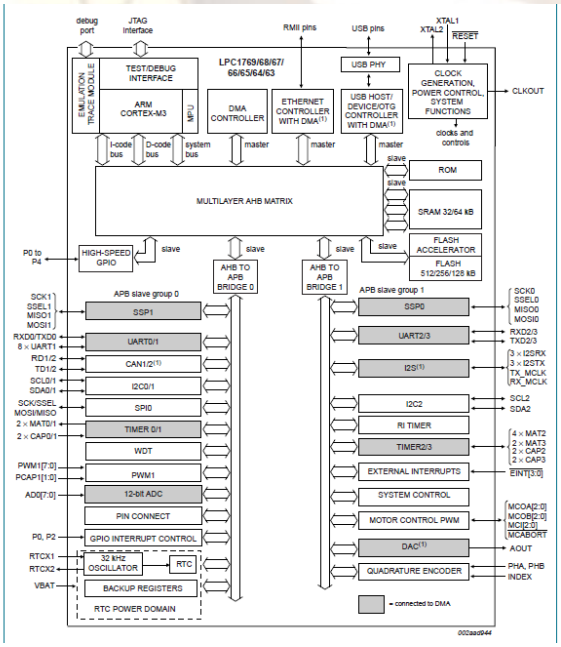

Figure 8: Block diagram of ARM CORTEX M3 Controller

iii) Interfacing of SPZB260 with ARM Controller

ARM Cortex M3 (LPC1768) use SPI Mode to Interface the SPZB260.The Programming of the LPC1768 is done by using Keil Uvision4 MDK.

The SPI transaction is as follows. The basic SPZB260 SPI transaction is half-duplex to ensure proper framing and to give the SPZB260 adequate response time. The basic transaction, as shown in Figure 9, is composed of three sections: Command, Wait, and Response. The transaction can be considered analogous to a function call. The Command section is the function call, and the Response section is the return value. The clock used for SPI transaction is 2 MHz For every 1 byte of data transfer or received a 8bit clock pulse is to be generated by LPC1768 Microcontroller.

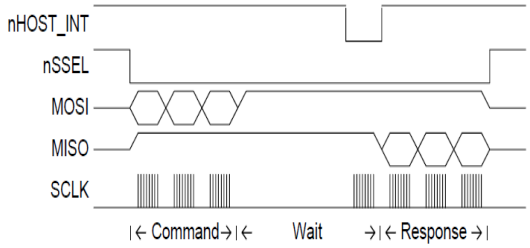

Figure 9: SPI transaction between microcontroller and SPZB260

a) Command Section

The LPC1768 microcontroller begins the transaction by asserting the Slave Select and then sending a command to the SPZB260. This command can be of any length from 2 to 128 bytes and must not begin with 0xFF. During the Command section, the SPZB260 will respond with only 0xFF. The LPC1768 should ignore data on MISO during the Command section. Once the LPC1768 has completed transmission of the entire message, the transaction moves to the Wait section.

b) Wait Section

The Wait section is a period of time during which the SPZB260 may be processing the command or performing other operations. Note that this section can be any length of time up to 200 milliseconds. Because of the variable size of the Wait section, an interrupt-driven or polling-driven method is suggested for clocking the SPI as opposed to a DMA method. Since the SPZB260 can require up to 200 milliseconds to respond, as long as the Host keeps Slave Select active, the LPC1768 can perform other tasks while waiting for a Response. To determine when a Response is ready, use one of two methods: Clock the SPI until the SPZB260 transmits a byte other than 0xFF. Interrupt on the falling edge of nHOST_INT. The first method, clocking the SPI, is recommended due to simplicity in implementing. During the Wait section, the

SPZB260 will transmit only 0xFF and will ignore all incoming data until the Response is ready. When the SPZB260 transmits a byte other than 0xFF, the transaction has officially moved into the Response section.

c) Response Section

When the SPZB260 transmits a byte other than 0xFF, the transaction has officially moved into the Response section. The data format is the same format used in the Command section. The response can be of any length from 2 to 128 bytes and will not begin with 0xFF. Depending on the actual response, the length of the response is known from the first or second byte and this length should be used by the Host to clock out exactly the correct number of bytes. Once all bytes have been clocked, it is allowable for the LPC1768 to deassert chip select. Since the LPC1768 is in control of clocking the SPI, there are no ACKs or similar signals needed back from the Host because the SPZB260 will assume the LPC1768 could accept the bytes being clocked on the SPI[7].

SOFTWARES USED

1. Keil`uVision4 software for embedded C programming.
2. Flash Magic software programmer for dumping code into ARM-cortex M3 LPC 1768 Microcontroller
3. Debugger used is CooCox CoIDE
4.Visual Basic for GUI which used to send command to LPC1768 via serial port.

III. IMPLEMENTATION OF TRANSRECEIVER MODEL IN MATLAB/SIMULINK

A. *CDMA Transmitter in Simulink*

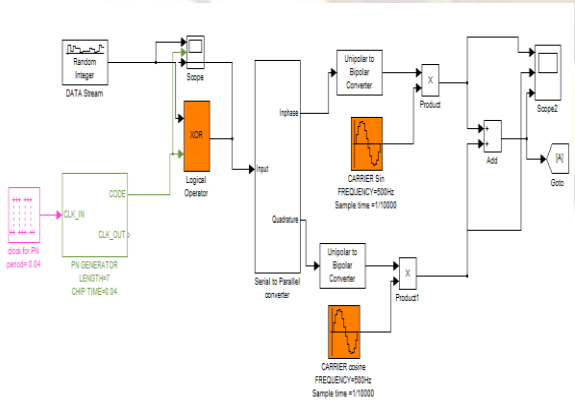

Figure 10: Implementation of CDMA Transmitter in Simulink:

i. Generating binary data stream: *By using Random integer generator block in the* communication tool box, we can generate binary data stream of 250Kbps. By adjusting the parameters like M-ary number, initial seed, sample time and output data type, we can achieve the fixed binary stream. In a real time scenario, this data stream is supplied by application that will generate information to be transmitted.

ii. ***Generating PN sequence***: PN code is generated using D-flip-flop. we can generate 7 bit-PN sequence chip rate of 0.04ms by using 3 D- flip-flop and a XOR gate to generate 7 bit PN code.

iii. Serial to parallel converter implementation: By using flip-flops in Simulink extras tool box, we can get the parallel data from the serial data. The necessary instruments are one clock, one JK flip flop and two D flipflops. The initial conditions of the flip-flops using is zero and the period of the clock was decided by the input data stream. By this way we can easily generate the parallel data technically called as inphase and Quadrature data. Here necessary one bit offset delay is provided by the D flip flop itself. multiplied to get a Direct spread spectrum signal.

iv. Performing Modulation: After obtaining the inphase and Quadrature signals we need to do modulation for the transmission of the signal. Generally we do this with the help of high frequency(500HZ and sample time of 1/10000) sinusoidal carrier. By using sine wave block in Signal Processing Tool Box, sine wave can be generated by adjusting the parameters like amplitude, frequency, sample time, phase and sine type. Now the inphase signal after half sine pulse shaping is multiplied by a sine wave and Quadrature is multiplied by its cosine signal which is nothing but 90 degree phase shift of original sinusoidal carrier.

v. Output of the Transmitter: Addition of both inphase and Quadrature signals after modulation, generates the required transmitter output. The required output signal is generated by using sum block in commonly used blocks. There will be no phase transitions in the output, which is an advantageous property[8].

B. *CDMA Receiver in Simulink*

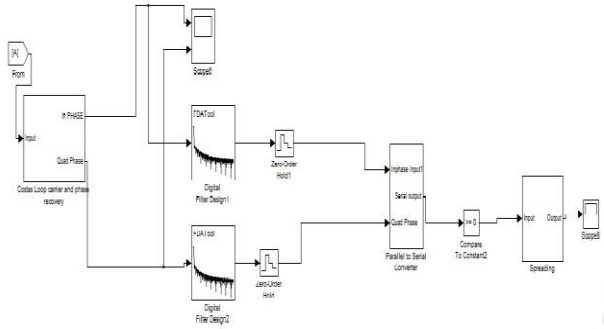

Figure 11: Implementation of CDMA Receiver in Simulink:

i. Multiplying with RF carrier: By using Costas loop is use for carrier recovery and phase detection which is used as demodulation circuit. In Costas Loop the incoming signal is mixed with the output of the VCO, both before its phase is shifted and after its phase is shifted by 90 degrees. These two outputs are then filtered, multiplied together, filtered again and to control the frequency of the voltage controlled oscillator. The decoded spreading data stream can be taken from the output of the mixer output.

ii. Sampling and Thresholding :

a. Sampling: By using the zero order hold circuit in the Simulink discrete menu, a sample and hold circuit was generated. It samples the signal for every T time period. By setting the sample time in this block, adjust the time period T in zero order hold circuit.

b. Thresholding: By using compare to constant block in the Simulink logic and bit operations menu. By setting operator, constant value and output data type parameters, we can get the comparator circuit, which compares the sampled data with the predefined threshold value and detects whether the transmitted data is „1" or „0".

iii. Parallel to serial conversion: By using switch block in the Simulink signal rooting menu, convert the parallel data into serial data. By setting the threshold value and the criterion for parallel to serial conversion, convert parallel data into serial data

iv. Despreading: The resulting data coming after serial to parallel conversion is multiplied with the delayed PN sequence. So that original is recovered data with small amount of delay. The incoming bit stream and the resultant output both are same but with a small amount of delay.

IV. SIMULATION RESULTS

A. At the transmitter end

The following figures demonstrate simulation results for CDMA transmission system. The results are displayed in the form of snapshots of scope signals.

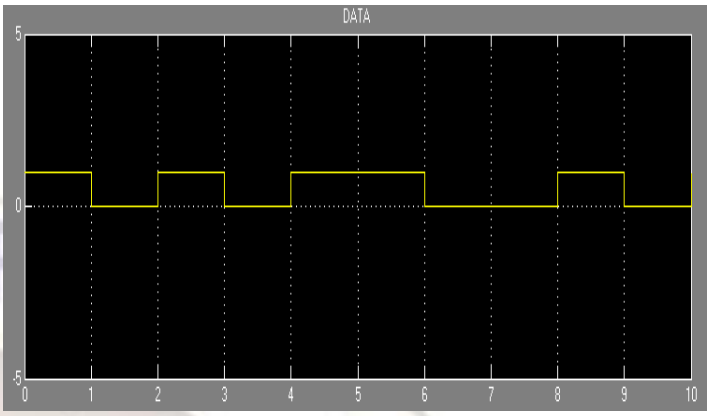

Figure 12: Binary data generated by Random integer Generator

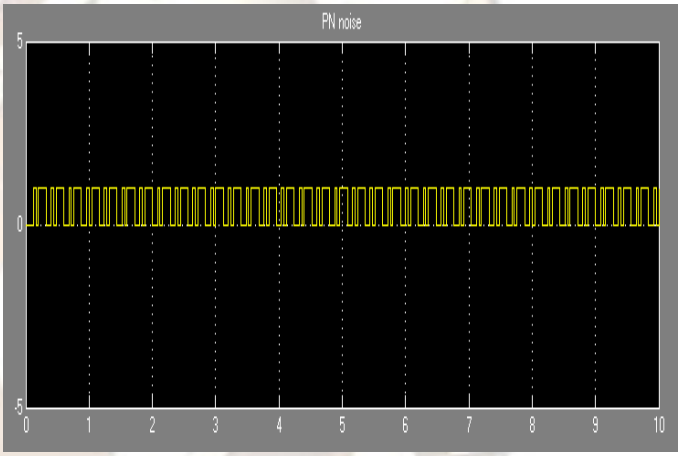

Figure 13: PN Generator

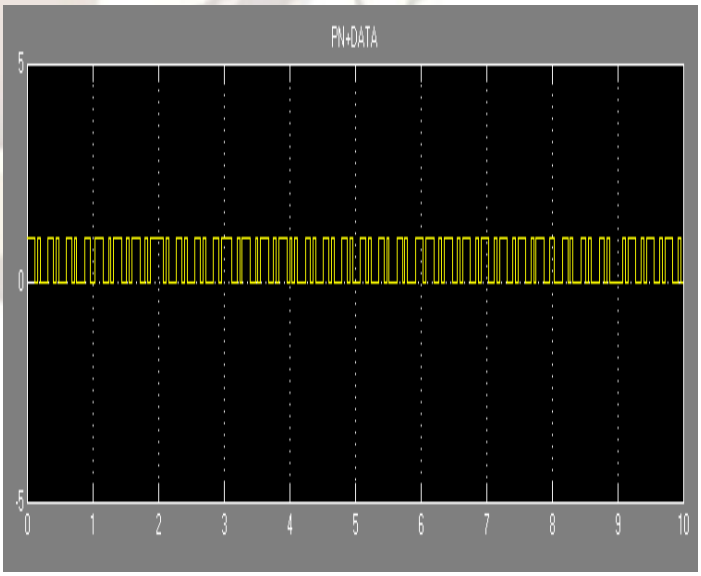

Figure 14: PN + Data

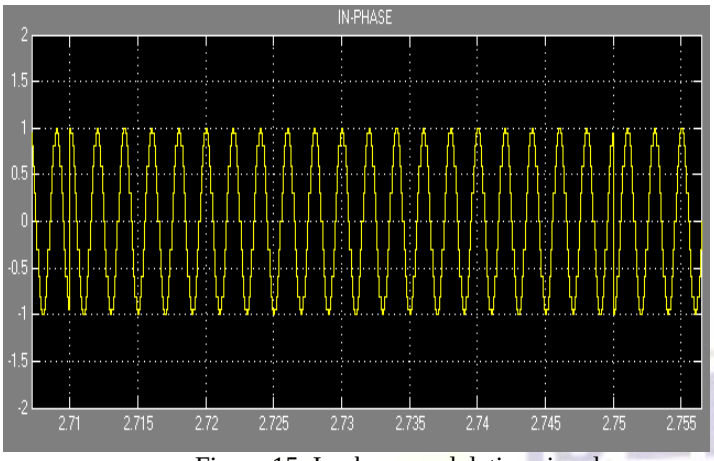

Figure 15: In phase modulation signal

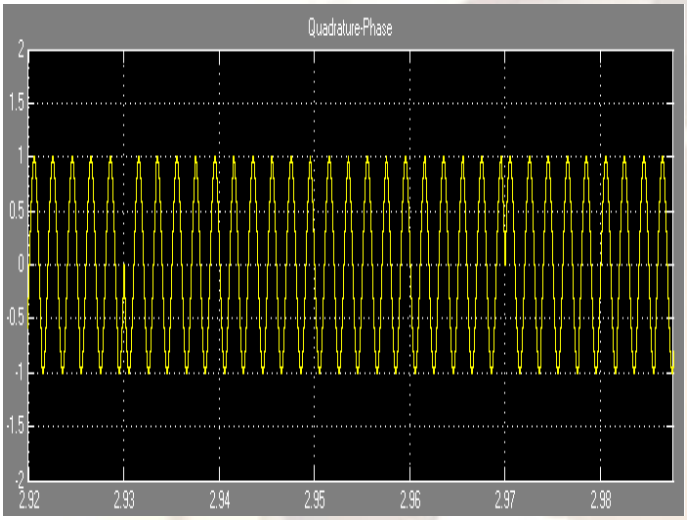

Figure 16: Quadrature phase modulation signal

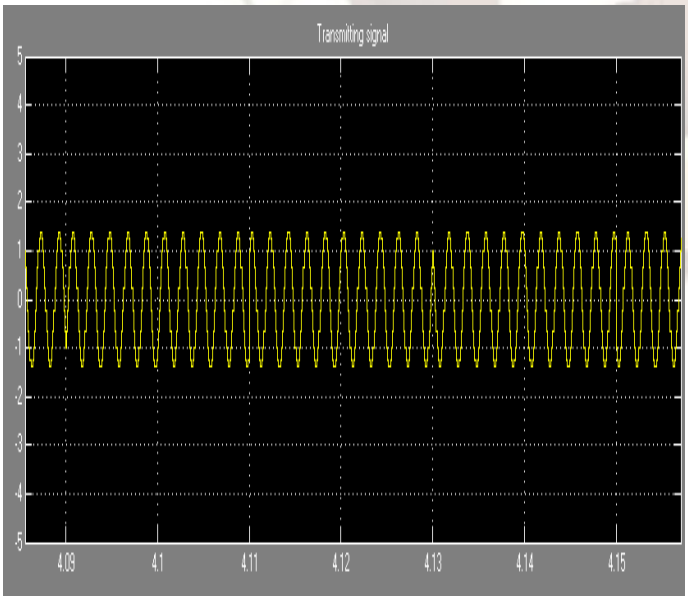

Figure 17: DSSS signal to transmit

B. At the receiver end

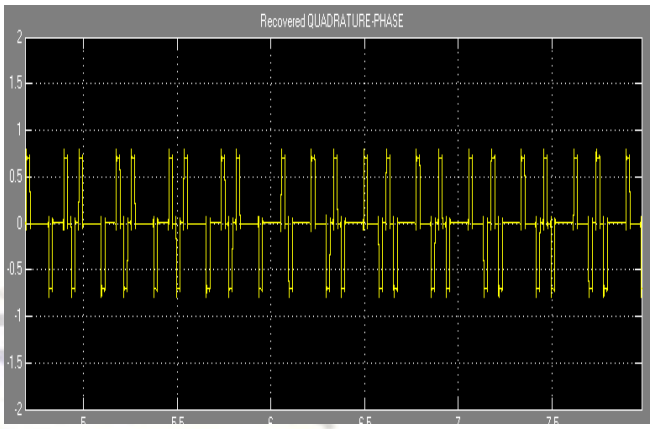

Figure 18: Recovered In phase stream using costas loop

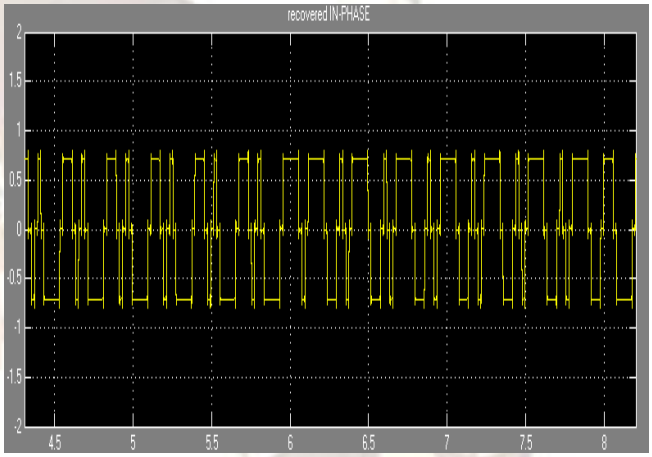

Figure 19: Recovered Quad phase stream using costas loop

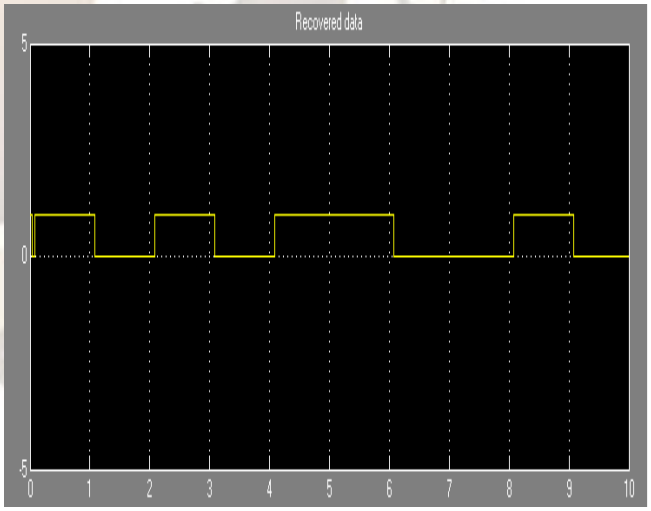

Figure 20: Recovered Data stream with small delay

CONCLUSION

CDMA RF link is used for achieving the communication between transmitters and receivers. Simulation results were plotted. Hard ware implementation is to be done using cortexM3 processor and SPZB260 Zigbee module for RF communications. In future, we aspire to improvise the design with many (2 to 3) client to control the other system of the missile.

REFERENCES

1. George R Cooper ,Clare D McGillem.".Modern Communications and Spread Spectrum" 1986 .
2. Dr.Kamilo Feher "Wireless Digital Communication." . June 2003.
3. http://en.wikipedia.org/wiki/Pseudorandom_noise.
4. http://en.wikipedia.org/wiki/PSK.
5. http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.PDF
6. http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00171682.
7. Kanna, Ravikanath. *Design of Zigbee Transmitter Receiver IEEE 802.15.4 using Matlab/Simulink.* Masters Thesis, Rourkela, Odhisha: National Institute of Technology, 2011.
8. mathworks. 2012. www.mathworks.com/help/toolbox/comm/ref/alignsignals.html (accessed March 5, 2012)