INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

***ISSN: 2278 – 0211 (Online)***

# Secured Wireless Communication System

**Swapnil Vinodia**
Takshshila Institute Of Engg. & Technology, Jabalpur, India
**Sagar Padam**
Takshshila Institute Of Engg. & Technology, Jabalpur, India
**Swapnil Gupta**
Takshshila Institute Of Engg. & Technology, Jabalpur, India
**Prof.Shobit Verma**
Asst. Prof. In Takshshila Institute Of Engg. & Technology, Jabalpur, India

***Abstract:***

*In this we implement a wireless link between two Microcontroller units (MCUs). This link will be used to send and receive digital data. We will create a RF link. However as we know the RF circuits are little complicated so we will use readymade RF Modules. These are easily available. For the sake of security we are providing a address and checksum concept to use multiple receivers at a moment which will be beneficial in industries where sensors for different machines can be able to communicate directly, without mutual interferences.*

**1.Introduction**

In many situations a communication link between to devices becomes essential. This communication can be wired or wireless. If two devices are close to each other (like a MCU and a Memory) a wired link is preferred. However in many situations two devices are reasonably far apart. In that case a wireless link is preferred. Two popular wireless communication technologies are-

*1.1RF Module*

A RF Module is a small circuit pre built and tested. They come in Pair. One is RX or the receiver and other is a TX or Transmitter

*1.2.Working of RF Modules*

Here whatever digital data you input on "Data In" of TX is available on "Data Out" of RX. Say, if you set "data in" high, the "data out" will become high as well. But here lies the trick! The fact is that you cannot Keep Logic HIGH or LOW for a Long period of time, say for a few millisecond second. If you apply a logic low on "data in" the "data out" will become low but only for few milliseconds and it will start oscillating (become high/low repeatedly) after that. Same thing will happen if you set "data in" to high state.

Let us assume we have connected the RF modules with MCUs as shown below.

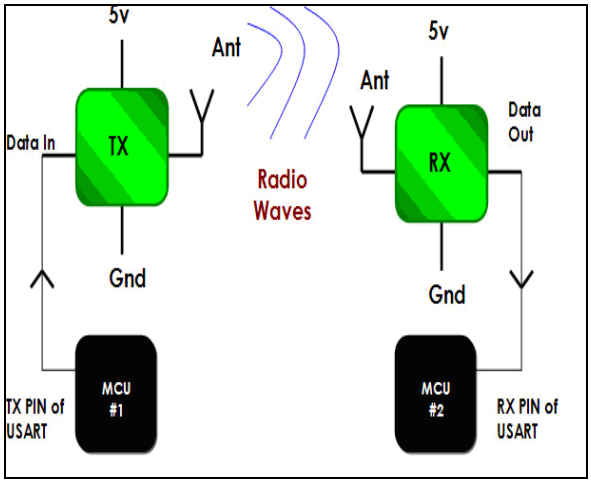


*Figure 1: RF Modules connected to USART of Microcontrollers*

**2.Software Introduction**

Let's move on to software part. Radio transmission is a bit more complicated than wired communications because you never know what radio signals are present on air. So all matters how transmitted signal is encoded. And this is a part where you have many choices: use hardware encoding like USART or writes your own based on one of many ending methods like NRZ, Manchester etc. In my example we have used AVR USART module to form data packs. Using hardware encoders solves many problems like synchronization, start and stop, various signal checks. But as long as we were practising you cannot rely on plain USART signal. Here you can actually improvize by adding various checks and so on.

We decided to form 4 byte data packages in order to send one byte information. These include:

- One dummy synchronization byte (10101010);
- One address byte – in case there are more receivers (or transmitters);
- One data byte;
- checksum which is actually a sum of address and data (address+data).

*2.1.Setting a BIT In Register*

Here our aim is to set (set to logical 1) any given bit (say bit 5) of a given register (say MYREG). The syntax is

MYREG=MYREG | 0b00100000;

The above code will SET bit 5 to 1 leaving all other bits unchanged. What the above code does is that it ORs each Bit of MYREG with each bit of 0b00100000 and store the value back in MYREG. If you know how logical OR works then you will get it.

In short you can write the same code as

MYREG|=0b00100000;

Now let's come to practical usage. In practice each bit has got a name according to its work/function. Say our BIT (the 5th bit) has got name ENABLE, and what it does is clear by its name, when we set it to 1 it enables the peripheral and when cleared (0) it disables it. So the right way to set it is.

MYREG|= (1<<ENABLE);

The << is called left shift operator. It shifts the bits of LHS variable left by the amount on its RHS variable. If you write

b=1<<3;

Then, 1 whose binary value is 00000001 is shifted 3 places to left which results in 00001000

So if ENABLE is defined as 5 (as enable is 5th bit) then

MYREG|= (1<<ENABLE);

Will result in

MYREG|= (1<<5);

Which again result in?

MYREG|= (0b00100000);

Now a beginner would ask "What's the Advantage?" And once you know it you would realize that advantage is immense!

*2.2.Flow Chart Of Data Reception Algorithm- Simple Algorithm For Sending And Receiving Data*

For this we created a simple mechanism. The steps are shown below.

- We begin transmission by sending character 'A'
- We again send one more 'A'
- Then we send the actual data.
- Now we send the inverse of data. That is all 0's are converted to 1 and vice-versa.
- We end the packet by sending 'Z'

In this way we create a simple packet based transmission with error detection. Now in the RX side MCU our program follows the algorithm given below.

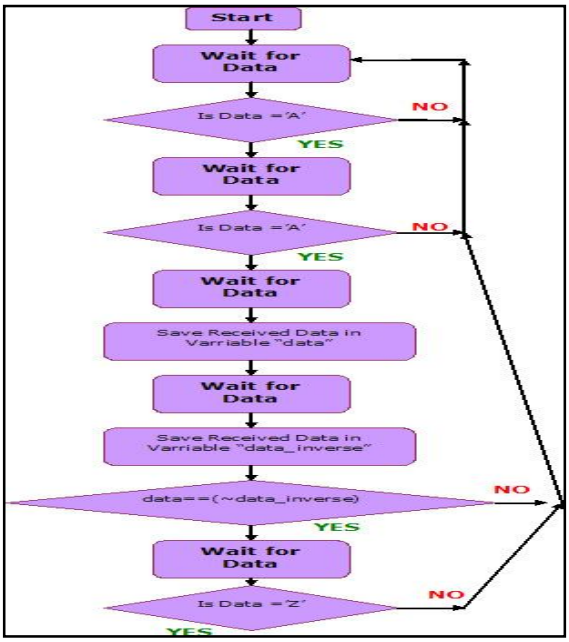


*Figure 1: Data reception algorithm*

*2.3.Circuit For Transmitter*

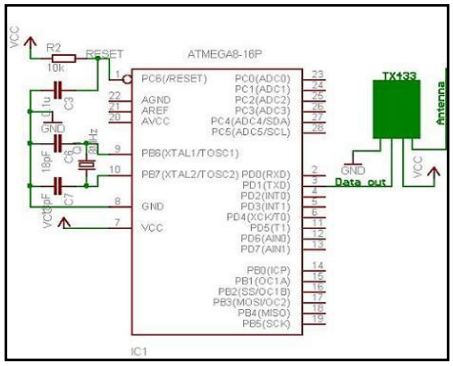


*Figure 2: Circuit for transmitter*

*2.4.Circuit For Receiver*

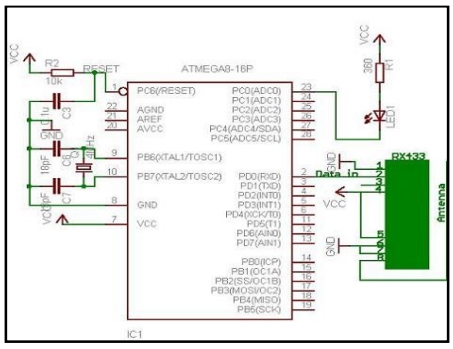


*Figure 3: Circuit for receiver*

**3. Future Enhancement**

As this project is based on the wireless communication. The word secured is mentioned because we can use this project for highly security areas. This project secures the message signal that is transmitted, and in which continuously varying of the bit takes place in every microsecond so, it becomes quite impossible for the other party to encode the information. This information can be encoded successfully at the receiver end only when the receiver is aware of the correct code of the bits. So it may be used as one of the most successful medium for transferring the secret and important information for the purpose of high security. So future work can also be done in the above mentioned direction .

**4.Reference**


1. L. Zhou and Z. Haas, “Securing ad hoc networks,” IEEE Network Magazine, pp. 24–30, Nov./Dec. 1999.
2. H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for wireless sensor networks,” in Proc. IEEE Symp. Security Privacy, May 2003.
3. C. Shannon, “Communication theory of secrecy systems,” Bell Syst. Tech. J., vol. 29, pp. 656–715, 1949.
4. A. Wyner, “The wire-tap channel,” Bell Syst. Tech. J., vol. 54, pp. 1355–1387, 1975.A. A. Hassan, W. E. Stark, J. E. Hershey, and S. Chennakeshu,
5. “Cryptographic key agreement for mobile radio,” Digital Signal Processing, vol. 6, no. 207-212, 1996.
6. H. Koorapaty, A. A. Hassan, and S. Chennakeshi, “Secure information Transmission for mobile radio,” IEEE Commun. Letts. pp. 52–55, Feb. 2000.
7. Ajay V.deshmukh (Tata McGraw hill education)publish-2010