Chapter 1

Laboratory SST: Protocol and handshaking using Arduino XBee

In this session it is attempted to teach students the concept of protocol and handshaking between a pair of radio module once the communication is established. It is not attempted to practice the IEEE[1] protocol standards. These standards are already built in hardware modules. Instead this will be done using a simple hardware and software which will be implemented during this laboratory session.

A protocol is a set of rules or agreed upon guidelines for communication. When communicating it is important to agree on how to do so. If one party speaks French and one German the communications will most likely fail. If they both agree on a single language communications will work. On the Internet the set of communications protocols used is called Transmission Control Protocol (TCP) and Internet Protocol (IP). TCP/IP is actually a collection of various protocols that each have their own special function or purpose. These protocols have been established by international standards bodies and are used in almost all platforms and around the globe to ensure that all devices on the Internet can communicate successfully.

There are a variety of protocols currently in use for wireless networking. Arguably, the most prevalent is IEEE 802.11b. Equipment using IEEE802.11b is comparatively inexpensive. The IEEE802.11b wireless communication standard operates in the unregulated 2.4GHz frequency range. Unfortunately, so do many other devices such as cordless phones and baby monitors which can interfere with your wireless network traffic. The maximum speed for IEEE802.11b communications is 11Mbps [2].

Handshakes are computer speak for electronic acknowledgments. Before a particular computer function goes any further, the computer checks in to make sure that everything has happened as it should. If everything is satisfactory, it shakes hands, sealing the deal before the next step happens. A computer handshake is basically this: its a term used to describe the process of one computer establishing

[1] IEEE: Institute of Electrical and Electronics Engineers

[2] Mbps: Mega bit per second

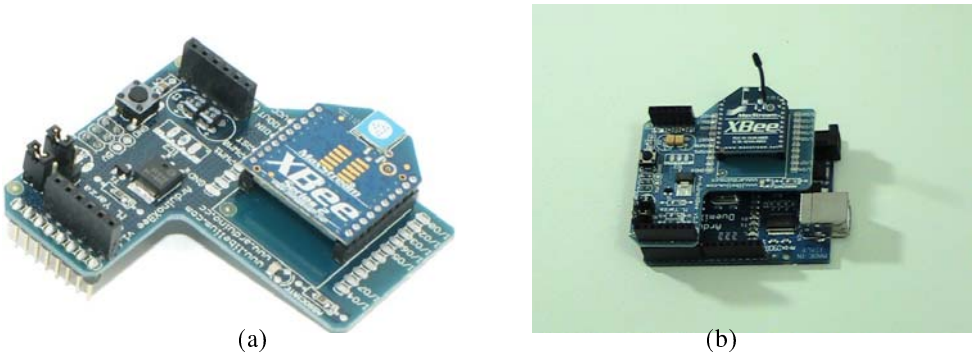
a connection with another computer or device. The handshake is often the steps of verifying the connection, the speed, and/or the authorization of the computer trying to connect to it. An example of handshaking is when a modem connects to another modem; the tones heard after the dialling is the handshake and can be thought of as the computers greeting each other.

1.1 ZigBee: XBee

ZigBee are small, low power digital radios operation in 2.4GHz frequency band. They are built based on the IEEE 802.15.4-2003 standard for Low-Rate Wireless Personal Area Networks (LR-WPANs) for wirelessly remote controls such as: Wireless light switches with lamps; electrical meters with in-home-displays, consumer electronics equipment or communication between machines in a factory. Data transmission rates vary from 20 kb/sec in the 868MHz frequency band to 250 kb/sec in the 2.4GHz frequency band. The output power of the radios is generally 0-20 dBm (1-100 mW). Transmission range is between 10 and 75 meters and up to 1500 meters for ZigBee PRO.

1.2 Arduino XBee

A Xbee shield allows an Arduino board to communicate wirelessly using Bee compatible modules (like Zigbee or BlueTooth Bee). It is designed to be used with XBee module from MaxStream. It can be used as a Serial Port / USB replacement. It is used to connect two Arduinos using Zigbee / Bluetooth Bee or connect a Arduino with PC Wirelessly. Two toggle switches decides how Rxd and Txd pins of Bee modules be connected to Arduino pins. These two switches provides options to connect RxD and TxD lines of Bee Modules to Arduino Hardware Serial Port or Digital pins 11 and 12 or FTDI RxD and Txd Pins. Figure 1.1 shows the XBee shield and Arduino mounted with XBee shield.



(a) (b)

Figure 1.1: (a) XBee shield; (b) Arduino mounted with XBee shield.

The modules are already reconfigured to operate as a transmitter or receiver. This is done using X-CTU software and has been explained during the lecture.

1.3 Circuit and programming Arduino

One group will be assigned to act as transmitter, while the other one as a receiver. Each group has two students. The Aduino's come with a Wireless shield installed. On the wireless shield there is switch to control the flow of data. If the switch is in the "USB" position, the Arduino can be programmed over the USB port. Also the serial monitor works normal. If the switch is in the "micro" position the wireless shield is connected to the serial port of the Arduino. Hence data can be transmitted and received through the wireless shield. The serial monitor can still be used to monitor the fow of data. Figure 1.2 shows the configuration of Arduino shield.

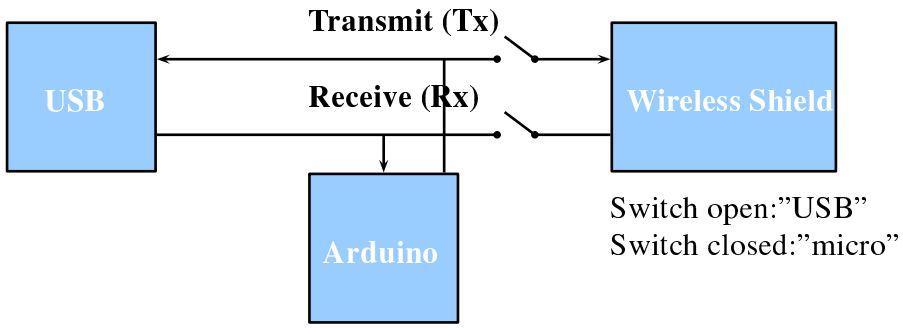


Figure 1.2: Configuring the wireless shield in Arduino environment.

When transferring data over a serial port, you are actually sending over 8 bit code. These codes can be interpreted as numbers, but also as character. Each character (0 .. 1, a..z, A.. Z, space, return,...) has its own code which is stadardized, the American Code for Information Interchange (ASCII). You can find code table on the internet. Wheter a number is to be interperted as a value or a character depends on how you declare yor variables;

```
Int x = 1;

Serial.print(x) // you send the ASCII code'1', which is 49

Serial.write(x) // you send 1

Char y = '1';

Serial.print(x) // you send the ASCII code'1', which is 49

Serial.write(x) // you send 49
```

1.3.1 Transmit side

Build a circuit having three switches. Connect the switches to digital pins 3, 4, and 5. If a switch is connected to a digital input of the Arduino, it is necessary that the input have a well defined voltage of either 5V or 0V. The way to realize

this is to use a pull up resistor circuit as shown in Figure 1.3. Note that the 5V and the GND are from the Arduino board. State of the switch can be read using the **digitalRead**(pin) command. Write a program that read sequentially the state of each switch in the loop and sends one character (a number or letter) to the serial port. Check the program using the serial port.

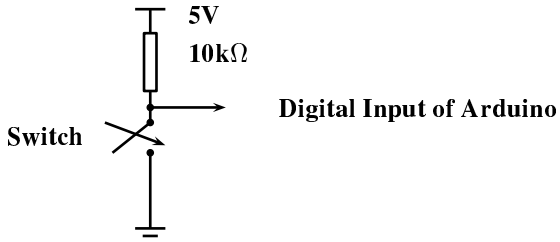


Figure 1.3: Pull up circuit.

1.3.2 Receive side

At the receive side build a circuit having three LEDs. Connect the LEDs to digital pins 3, 4, and 5. If a LED is connected to a digital pin of the Arduino, it is necessary that to use a suitable resistor depending on the current and the voltage that LED is required. Figure 1.4. Each LED can be turned on and off using the **digitalWrite**(pin) command. Write a program that sequentially turning on and off the LEDs in the loop.

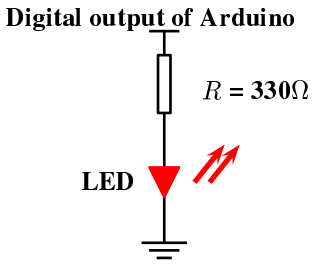


Figure 1.4: LED circuit.

1.4 Communication

It is the attempted to build communication between the transmitter and receiver modules to carry out a number of instructions. The communication should be based on well defined protocol and handshaking. Let us define a certain protocol and corresponding handshaking. It is very important that the XBee shield is mounted after the program is uploaded in Arduino. Once the program is uploaded mount then the XBee shield. The communication set-up is then ready to exchange data. Note the follwoing:

- **Serial.available**() to check if data is available at the receiver.
- **Serial.read**() to read data Byte at the receiver.
- **Serial.write**() to write data at the transmitter.

The protocol and handshaking are defined as follows:

First Handshaking

- An acknowledgement statement i.e., ”Are you ready?” is sent from the transmitter to the receiver.
- An acknowledge statement i.e., ”I am ready” is sent from the receiver to the transmitter.

The transmitter and the receiver are ready to exchange data. Program the modules such that after each action from transmitter side i.e., closing or releasing the switch, the LEDs would sequentially turned on or off. After each sequence a handshaking confirmation is sent back to the transmitter. Hence:

Protocol and Handshaking

- If the 1^{st} switch is closed then the first LED will be tuned on.
- A handshaking message is sent that the command is performed.
- If the 1^{st} switch is released then the first LED will be turned off.
- A handshaking message is sent that the command is performed.
- And so on.