Blue Tooth – Basics
Using BT-Arduino

Dr. Lynn Fuller
Webpage: http://people.rit.edu/lfeee/
Microelectronic Engineering
Rochester Institute of Technology
82 Lomb Memorial Drive
Rochester, NY 14623-5604
Tel (585) 475-2035

Email: Lynn.Fuller@rit.edu
MicroE webpage: http://www.microe.rit.edu
Bluetooth is a proprietary open wireless technology standard for exchanging data over short distances (using short wavelength radio transmissions in the ISM band from 2400-2480 MHz) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security. Created by telecoms vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

http://en.wikipedia.org/wiki/Bluetooth
DEFINITIONS

Arduino – refers to a project that provides open source hardware and software to learn by doing projects with micro controllers.

Arduino-BT – one of the several Arduino hardware platforms available containing a micro controller, power regulator, Blue Tooth Transceiver, interconnect pins, sockets, crystal, LEDs and other components.

Shield – an add on hardware board that plugs into the Arduino micro controller platform and provides additional capabilities such as analog signal processing, WiFi, etc.

Processing – a “C” based software programming tool to create graphical output and communicate with hardware platforms such as the Arduino Uno.

Sketch – name for the “C” programs used by “Processing” and by “Arduino” software to make the hardware do something and to process the results.
Wireless Microsystems

- Capacitor Sensor
- Bluetooth Transceiver
- Wireless Platform Breadboard
- 5” x 8” Breadboard
- 2.5” x 3” BT-Arduino
- 1” x 1” PCB
- 2mm x 3mm Custom CMOS & MEMS
Bluetooth Basics

BLUETOOTH

Bluetooth V2.0 and V2.1 Rate ~3Mbit/s
Bluetooth V3.0 Rate ~24Mbit/s
Bluetooth V4.0 Rate ~24Mbit/s + Low Energy

Hardware to transmit and receive radio signals
Transceivers, Dongles both with antennas
Bluetooth Basics

BLUETOOTH POWER CLASSES

The following table compares the available Bluetooth power classes:

<table>
<thead>
<tr>
<th>Power Class</th>
<th>Maximum Power</th>
<th>Operating Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>100mW (20dBm)</td>
<td>100 meters</td>
</tr>
<tr>
<td>Class 2</td>
<td>2.5mW (4dBm)</td>
<td>10 meters</td>
</tr>
<tr>
<td>Class 3</td>
<td>1mW (0dBm)</td>
<td>1 meter</td>
</tr>
</tbody>
</table>

The actual range for each power class may vary depending upon environmental factors where the Bluetooth device is used.

• If you wish to communicate over the 100m range, you will need a class 1 Bluetooth device at both ends.

• If you wish to communicate over the 10m range, you can have a class 1 or class 2 device at both ends.
PERSONAL AREA NETWORKS (PAN)

At any given time, data can be transferred between the master and one other device (except for the little-used broadcast mode). The master chooses which slave device to address; typically, it switches rapidly from one device to another in a round-robin fashion. Since it is the master that chooses which slave to address, whereas a slave is (in theory) supposed to listen in each receive slot, being a master is a lighter burden than being a slave. Being a master of seven slaves is possible; being a slave of more than one master is difficult. The specification is vague as to required behavior in scatter nets.
Our definition for Microsystems is the integration of MEMS sensors and actuators with CMOS electronics to provide solutions for a wide variety of applications including automotive, military, aerospace, consumer and biomedical.
ARDUINO BLUETOOTH

Microcontroller ATmega328
Operating Voltage 5V
Input Voltage (limits) 1.2-5.5V
Digital I/O Pins 14
(of which 6 provide PWM output)
Analog Input Pins 6
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA
Flash Memory 32 KB (ATmega328)
of which 0.5 KB used by bootloader
SRAM 2 KB (ATmega328)
EEPROM 1 KB (ATmega328)
Clock Speed 16 MHz

BT-Arduino Development Board
2” x 2 ¾”
~$100
BLUEGIGA WT11 MODULE USED WITH ARDUINO

WT11 is a next-generation, class 1, Bluetooth 2.1 + EDR module. It's a highly integrated Bluetooth module, containing all the necessary elements from Bluetooth radio to antenna, and a fully implemented iWRAP protocol stack.

**Key Features:**
- Bluetooth class 1
- Integrated chip antenna or U.FL connector
- USB, UART, GPIO, AIO and PCM interfaces
- 802.11 co-existence interface
- 8MB flash memory
- Size: 35 x 14 x 2.3mm
- Simple iWRAP firmware for controlling Bluetooth wireless technology
- Industrial temperature range from -40C to +85C
- RoHS compliant
- Up to 14 supported Bluetooth profiles in iWRAP firmware
- Also available with HCI firmware
- Support for on-board applications
- Fully qualified end product with Bluetooth 2.1 + EDR, CE, IC and FCC

**Description**

Bluegiga WT11 provides an ideal solution for developers that want to integrate Bluetooth wireless technology into their design with limited knowledge of Bluetooth and RF technologies. By using WT11 combined with Bluegiga's complete development, testing and verification service offering and excellent developer support, OEMs and designers ensure that their products reach the market rapidly and cost-effectively in relation to time and resources. Bluegiga has in-house knowledge of both software and hardware - offering customers a single point of contact for all Bluetooth related issues.

**Used on ArduinoBT Development Board**

Arduino Bootloader (reconfigurable)
Baud Rate Set at 115200
Device Name Set to: ARDUINOBT
Access Code Set to: 12345
**Bluetooth 4.0 Single Mode Modules**

Bluetooth 4.0 is a new, open standard developed by the Bluetooth SIG. It's targeted to address the needs of new modern wireless applications such as ultra-low power consumption, fast connection times, reliability and security. Bluetooth low energy consumes 10-20 times less power and is able to transmit data 50 times quicker than classical Bluetooth solutions. **Comparison between Classic Bluetooth and Bluetooth Low Energy**

*BLE112 Bluetooth 4.0 Single Mode USB Dongle* is a single mode USB device.

*BLE112 Bluetooth 4.0 Module* is a Bluetooth 4.0 single mode device targeted for low power sensors and accessories.
**SETTING UP BLUETOOTH ARDUINO**

**Blink Program Test**

**Install BT adapter**
*DO NOT install the software with the adapter, it will keep on scanning the ports and slow down the reaction of Arduino. Let the Windows native driver work and ignore the other failed installation service, because only the serial communication is necessary.

**Find Bluetooth device**
*Turn on the board, then right click on the Bluetooth icon appear on the bottom right corner>Add a device>choose ARDUINOBT and input the pass code 12345>pair success

**Check the automatic port setting**
*Right click on the Bluetooth icon appears on the bottom right corner>Open Settings>COM Ports
Set the port number and restart PC
*Right click on Computer icon>Manage>Device Manager>Ports(COM & LPT)>right click on the port(Standard Serial over BT link(COM#))>Properties>Port Settings>Bits per second=115200 and Advanced>COM Port Number
*Choose a low port number that has not been used by other devices

**Choose the correct board at Arduino IDE**
*Tools>Board>Arduino BT w/Atmega328

**Choose the correct serial port.** Only both outgoing and incoming ports show up in the menu, the board can work and respond correctly.
*Tools>Serial Port>choose the outgoing port you set

**Connect the LED between pin 13 and GND**

**Upload the blink program**
*Type in the code>Compile>Upload  (press the reset button after binary sketch size showing up)

**Blink!!**
/* Arduino BT Configuration File  Feb 05 2007 */
/* Upload and run this file <<once>> to configure the name or other variables of your BT module. */
/* Don't forget to reset your module so that the new commands are executed. */
/* Massimo Banzi */
/*
int LED = 13;       // select the pin for the LED
int RESET = 7;      // BT module uses pin 7 for reset

void setup() {
  pinMode(LED, OUTPUT); // declare the LED and BT RESET pins as output
  pinMode(RESET, OUTPUT);
  Serial.begin(115200); // connect to the serial port

digitalWrite(RESET, HIGH);
delay(10);
digitalWrite(RESET, LOW);
delay(2000);
CHANGING DEVICE NAME AND AUTHORIZATION CODE

// the following "SET BT" commands are one-time commands
// that only need to be uploaded to your BT module and run
// when you want to change the BlueGiga's configuration.
// ok. First, the easy to change configuration commands:

    Serial.println("SET BT NAME ChenArduino");
    // you can change the name of your module here.
    // No spaces allowed in the name; names can be up to 256 characters.
    Serial.println("SET BT NAME KarlMarx");  // example

    Serial.println("SET BT AUTH * 123123");
    // 12345 == 0 to 16 digit Bluetooth passkey/PIN code
    Serial.println("SET BT AUTH * 987654321");  // example

void loop() {
    digitalWrite(LED, HIGH);
    delay(100);
    digitalWrite(LED, LOW);
    Serial.println("goodjob!");
    delay(1000);
}
Arduino-BT running blink and serial port write every second uses 20 mA at 3 volts or 60 mW

AA Battery Stores ~2000 mAh of Charge Q
Time = Q / I = 2000 / 20 = 100 hours

CR2032 Battery Stores ~200 mAh of Charge Q
Time = 200/20 = 10 hours

1 month is 720 hours
Current = Q / time
I = 2.8 mA

<table>
<thead>
<tr>
<th>Battery Type</th>
<th>Capacity (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>12000</td>
</tr>
<tr>
<td>C</td>
<td>6000</td>
</tr>
<tr>
<td>AA</td>
<td>2000</td>
</tr>
<tr>
<td>AAA</td>
<td>1000</td>
</tr>
<tr>
<td>N</td>
<td>650</td>
</tr>
<tr>
<td>9 Volt</td>
<td>500</td>
</tr>
<tr>
<td>6 Volt Lantern</td>
<td>11000</td>
</tr>
</tbody>
</table>
BATTERIES

Batteries in series doubles the voltage but maintains the same capacity.
Batteries in parallel doubles the capacity but maintains the same voltage.

Four AA connected batteries as shown give 4000mAh at 2.4V

2000 mAh, at 1.2V
REFERENCES

4. www.bluetooth.com
6. Bluetooth BTD211 Micro USB 2.1 Adapter, BlueSoleil software support from Azio Corporation, www aziocorp.com
1. None
2. None