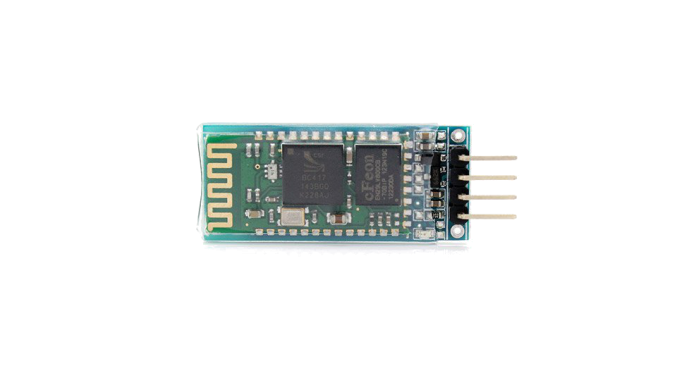
# JY-MCU Bluetooth Transceiver

# Introduction

###########ISM UHF must go into acronyms section



These are cheap but robust and stable Bluetooth to UART Wireless transceiver adapters. They have very low power consumption with a very high performance transceiver system. These adapters can be used in generally anything that requires a wireless connection with other devices. These modules can be used to help devices communicate wirelessly with other devices. Devices include: a Bluetooth Car hands free device, a mobile phone with a Bluetooth communication interface, Bluetooth GPS devices etc. For the purpose of this project this device is used to connect the ATmega328-PU microcontroller wirelessly to an Android phone. This device will allow for two way communication between the microcontroller and Android phone.

Bluetooth is wireless protocol used in the transfer of data using Ultra High Frequency (UHF) in the ISM band. Using UHF uses wide frequency bands that can transfer data at very high data rates using very low power within short distances. The UHF is better at penetrating objects and going through doors as apposed lower frequency bands. This protocol is very useful for indoor applications where objects can block the line of sight of two Bluetooth devices. Bluetooth used be standardized by IEEE and was formally known as IEEE 802.14.1. It is now manage Bluetooth Special Interest Group (SIG) which has 25000 members.

The Bluetooth protocol is found to very useful in the project mainly for it low power consumption. The range of operation is reasonable and the data rates are more than enough. The aim is to make the project very power efficient and Bluetooth fits that criterion of power efficiency.

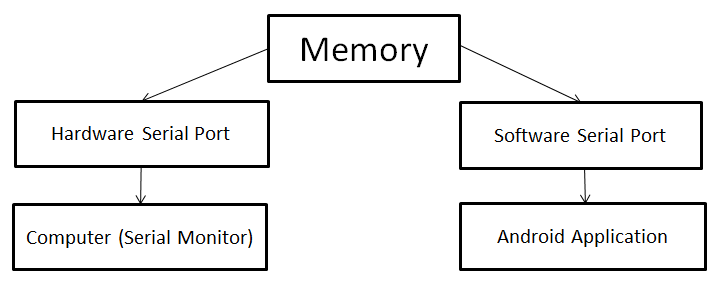
# Specifications

|  |  |
| --- | --- |
| Parameters |  |
| Bluetooth Version | V2.0+EDR (Enhanced Data Rate) |
| Wireless Protocol | Serial Port Protocol (SPP) 3Mbps |
| Pin sizes | Half hole size |
| Bluetooth Password | 1234 |
| Baud rate | Baud rate between 1200 and 1382400 (9600 default) |
| AT Command | Yes |
| Dimension | 4.4 cm x 1.6 x 0.7 cm |
| Voltage | 3.6 - 6 V |
| Current | 30 ~40 mA (pairing) & 8mA (communication) |
| Serial Protocol | 3.3 V level UART interface |
| Operation Temperature | -25 to 75 ℃ |
| Frequency | 2.4 GHz |
| Range | 10 meters |

# Communication Pins

|  |  |
| --- | --- |
| Pins | Description |
| 1 | Key |
| 2 | VCC (3.6-6V) |
| 3 | GND |
| 4 | TXD (3.3 V) |
| 5 | RXD (3.3 V) |

# Test Procedure



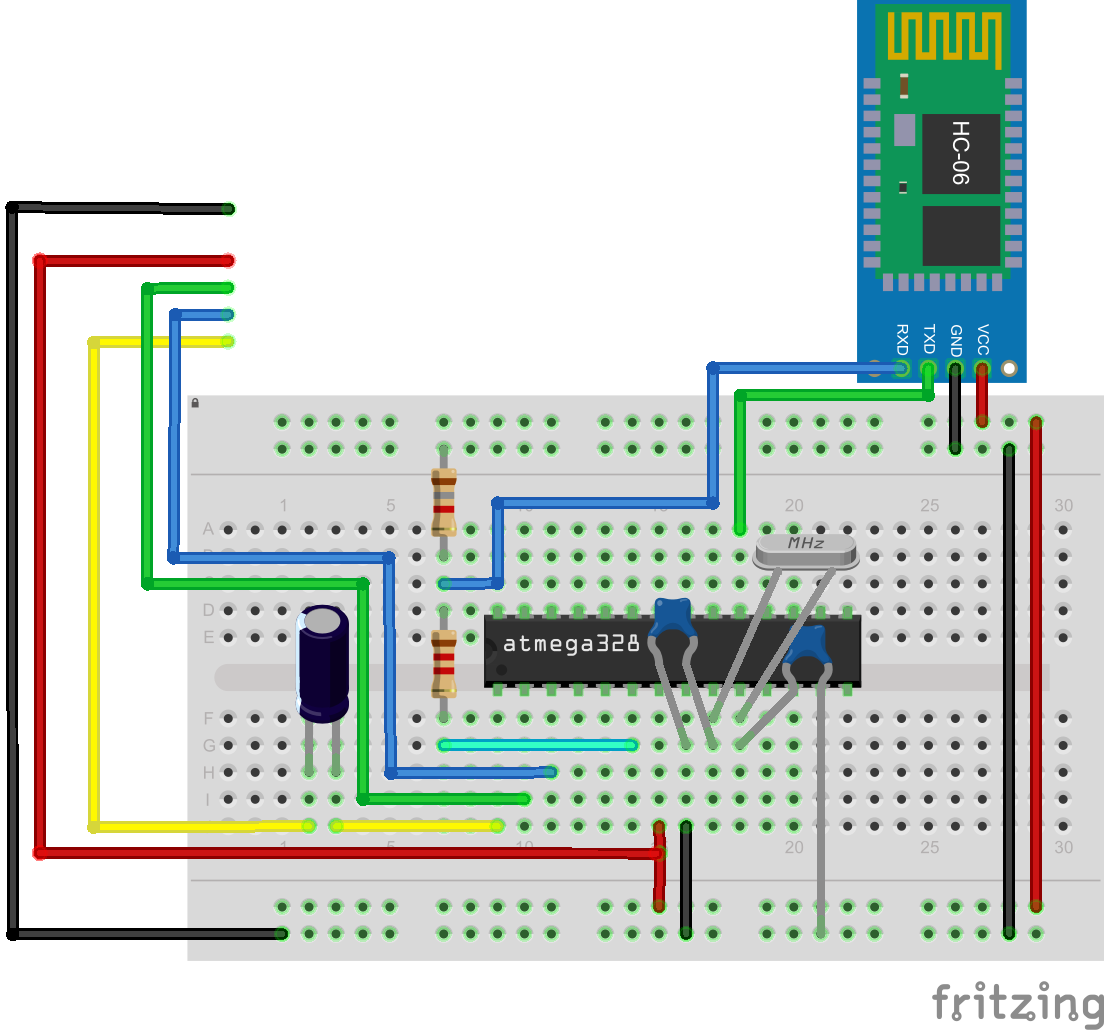
This is at test procedure that can be taken to test whether the Bluetooth adapter is operational and to get a sense of how it works. The first thing is to find an android application on the Google Play Store that can send and receive characters. One such application called “Bluetooth Terminal” is able to send and receive characters. And lastly the circuit is set to test the Bluetooth protocol. The hardware serial port and a setup Software Serial port are used in this test. As mentioned earlier the software serial port is created by making IO pins behave like a serial port. The hardware and software serial port will share the same space of memory. The hardware serial monitor will control data transfer between the computer and the memory. The software serial port will control data between the android application and the memory.

# Components

* JY-MCU Bluetooth Adapter.
* 1.2 kΩ, 1.8kΩ & 1 kΩ Resistors.
* Breadboard.
* Jumper cables.
* LED.
* Bluetooth android application from Google Play Store.

# Schematic

The schematic provided shows how the Bluetooth module is implemented into the circuit. As there is only one UART port provided by the microcontroller it was decided a Software Serial port be setup for the Bluetooth module. Pins 13 (RXD) and 4 (TXD) are used to establish this software serial port.



# Code

#include <SoftwareSerial.h>

char value**;** // Variable to receive data from the serial port

void setup**()** **{**

Serial**.**begin**(**9600**);** // Init serial communication at 9600bps

bluetooth**.**begin**(**9600**);**

**}**

void loop**()** **{**

**if(** Serial**.**available**()** **)** // If data is available to read

**{**

val **=** Serial**.**read**();** // Read it and store it in value.

bluetooth**.**write**(**val**);** // Software serial transmits the saved data to android application

**}**

**if(** bluetooth**.**available**()** **)** // If data is available to read

**{**

val **=** Serial**.**read**();** // Read it and store it in value

Serial**.**write**(**val**);** // Hardware serial transmits the saved data to the serial monitor

**}**

**}**

References

<http://upcommons.upc.edu/pfc/bitstream/2099.1/23666/7/Annex3Datasheet%20m%C3%B2dul%20Bluetooth%20JY-MCU.pdf>

<http://en.wikipedia.org/wiki/Bluetooth>

<http://upcommons.upc.edu/pfc/bitstream/2099.1/23666/7/Annex3Datasheet%20m%C3%B2dul%20Bluetooth%20JY-MCU.pdf>

<http://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Bluetooth.svg/671px-Bluetooth.svg.png>

<http://robotosh.blogspot.ie/2012/07/arduino-jy-mcu-bluetooth.html>