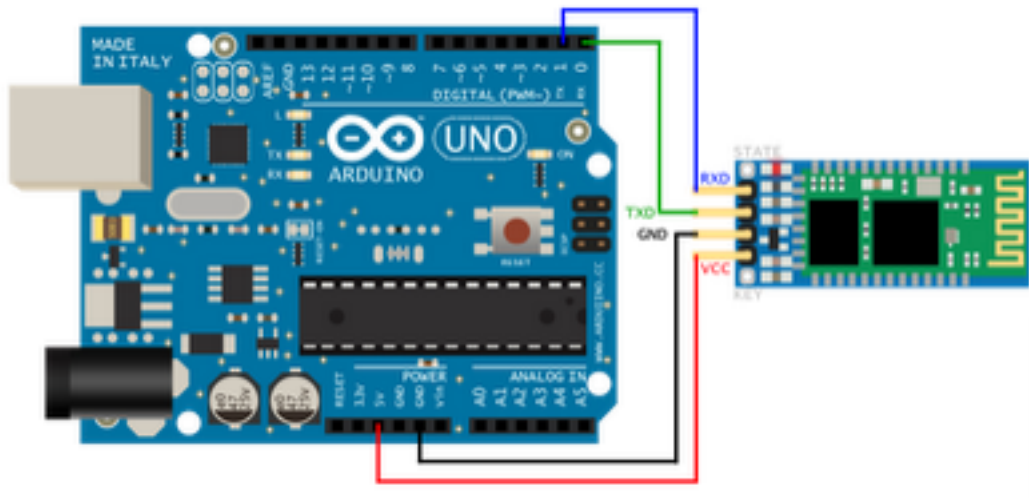


BLUETOOTH

Bluetooth JY-MCU module (Hobby Components) HC-06 module



Connection as above (4 pins)

BT Module HC-06

BT Name: see below

Pwd: see below

Baud rate default 9600

Can be "slave" only as based on HC-06 module.

TXD is the data received from remote

RXD is the data to send to remote

BT module configuration

Do not pair before this sketch, in order to send AT commands BT red LED must be flashing.

Set name to GanymedeBT, PIN to 0000, baud rate to 9600 using AT commands below:

```
/* BT module configure
BT flashing
send AT+NAMEGanymedeBT
send AT+PIN0000
send AT+BAUD4 (9600)
connections
Arduino      BT
5V           PIN 2 (VCC)
GND         PIN 3 (GND)
10          PIN 4 (TXD)
11          PIN 5 (RXD)
*/

#include <SoftwareSerial.h>

#define RXPIN 10
```

```

#define TXPIN 11

SoftwareSerial BT(RXPIN, TXPIN); // create class BT

void setup()
{
  Serial.begin(9600); // start USB
  BT.begin(9600); // start BT
}

void loop()
{
  if (BT.available())
  {
    Serial.write(BT.read()); // echo from BT
  }

  if (Serial.available())
  {
    BT.write(Serial.read()); // write AT command to BT
  }
}

```

Then System > Preferences > Bluetooth. “GanymedeBT” will be detected and can be [Pair]ed.

Serial Test BT

Use CoolTerm to transmit/receive BT module, and Serial Monitor to receive/transmit back.

```

/* echo test BT. Data sent on Serial Monitor to BT and back to CoolTerm, and
CoolTerm back to Serial Monitor
connections
Arduino      BT
5V           PIN 2 (VCC)
GND         PIN 3 (GND)
10 (RXPIN)   PIN 4 (TXD) - data transmitted from remote
11 (TXPIN)   PIN 5 (RXD) - remote to receive this data
- load sketch
- System > Preferences > Bluetooth "GanymedeBT" Pair
- CoolTerm > Options, select GanymedeBT serial port
- open IDE serial monitor
*/

#include <SoftwareSerial.h>

#define RXPIN 10
#define TXPIN 11

SoftwareSerial BT(RXPIN, TXPIN); // create BT class object

void setup()
{
  Serial.begin(9600); // init serial
  BT.begin(9600); // init BT
}

void loop()

```

```

{
  if (BT.available()) // if BT OK
  {
    Serial.write(BT.read()); // read from BT, write to USB
  }

  if (Serial.available()) // if USB OK
  {
    BT.write(Serial.read()); // read USB, write to BT
  }
}

```

Another Bluetooth project

This project uses the Morse generator `morseEnDecoder.h` to output morse to an LED. The messages to send are received by Bluetooth from the iMac.

1 System > Preferences > Bluetooth

2 Pair the GanymedeBT module

3 Open the terminal app and enter: `screen /dev/tty.GanymedeBT-DevB`

4 The terminal app will clear the screen and now act as a terminal, morse messages can be sent by typing them on the KB

5 Alternatively the CoolTerm app can be used. Options > Serial Port > GanymedeBT-DevB and Options > Terminal > Key Emulation > CR. Disconnect after use.

```

// morse sender - flash LED on 7, could be a relay
/*
connections
Arduino      BT
5V           PIN 2 (VCC)
GND         PIN 3 (GND)
10 (RXPIN)   PIN 4 (TXD) - data transmitted from remote
11 (TXPIN)   PIN 5 (RXD) - remote to receive this data
- load sketch
- System > Preferences > Bluetooth "GanymedeBT" Pair
- CoolTerm > Options, select GanymedeBT serial port
*/

#include <avr/pgmspace.h>
#include <MorseEnDecoder.h>
#include <SoftwareSerial.h>

#define WPM 5

#define LEDPIN 7
#define RXPIN 10
#define TXPIN 11

morseEncoder morseOut(LEDPIN);

SoftwareSerial GanymedeBT(RXPIN, TXPIN); // GanymedeBT class object in Arduino

long lastTrans;
long current;
boolean ended = true;

void setup() {

```

```
pinMode(LEDPIN, OUTPUT);

GanymedeBT.begin(9600);

morseOut.setspeed(WPM);

lastTrans = millis();
}

void loop() {

  current = millis();

  morseOut.encode();

  if(GanymedeBT.available() && morseOut.available()) {
    char text = GanymedeBT.read();
    GanymedeBT.write(text); // echo
    if(text == '\r') GanymedeBT.write('\n'); // when CR received add cq cq cq
m6kwh NL
    morseOut.write(text);
  }

  if(!morseOut.available()) {
    lastTrans = current;
    ended = false;
  }
}
```