Martyn Currey

Mostly Arduino stuff

Arduino with HC-05 (ZS-040) Bluetooth module – AT MODE

Posted on **October 28, 2014**

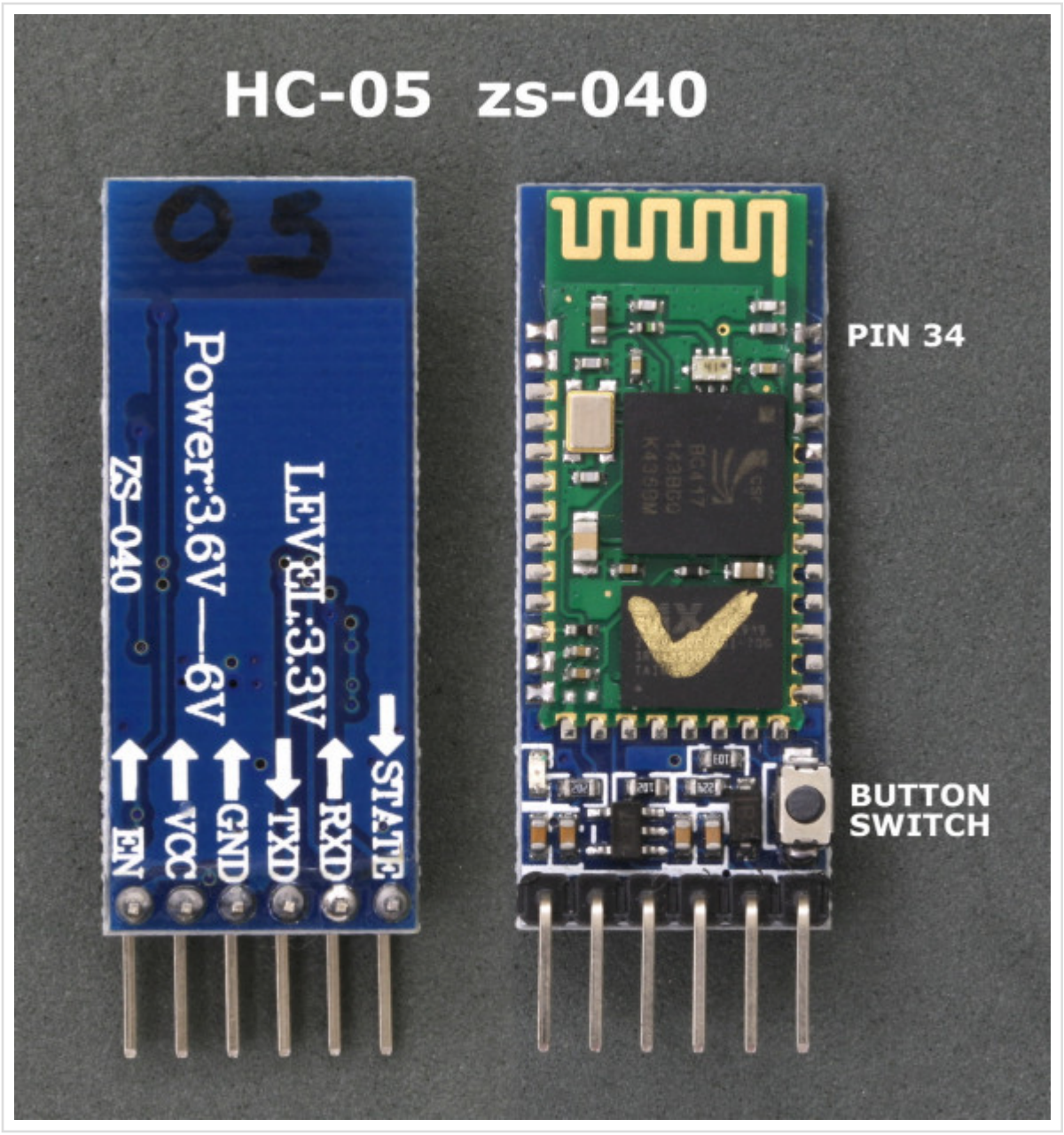
Updated 19.07.2015
Updated 26.07.2015

AT mode allows you to interrogate the BT module and to change some of the settings; things like the name, the baud rate, whether or not it operates in slave mode or master mode. When used as a master device AT commands allow you to connect to other Bluetooth slave devices.

There are many slightly different HC-05 modules, the modules I have are marked ZS-040 and have an EN pin rather than a KEY pin. They also have a small button switch just above the EN pin. They are based on the EGBT-045MS Bluetooth module.

Update: I now also have boards marked fc-114. See:
HC-05 FC-114 and HC-06 FC-114. First Look
HC-05 FC-114 and HC-06 FC-114. Part 2 – Basic AT commands
HC-05 FC-114 and HC-06 FC-114. Part 3 – Master Mode and Auto Connect

On the zs-040 modules there are 2 AT modes. I do not know if this is intentional but some commands only work when pin34 is HIGH. Other commands work when pin 34 is either HIGH or LOW. This fooled me for quite a while. For this post I have called the different modes “mini” AT mode and “full” AT mode.



To activate AT mode on the HC-05 zs-040 modules we can:

– 1. Hold the small button switch closed while powering on the module.

– 2. Set pin 34 HIGH (3.3v) when power on.

– 3. Close the small push button switch after the HC-05 is powered.

– 4. Pull pin 34 HIGH after powering the HC-05.

Method 1.

Enters AT mode with the built in AT mode baud rate of 38400. The baud rate cannot be changed by the user.

This method allows the module to enter AT mode on start but but does not keep pin 34 HIGH and uses the “mini” AT mode.

Method 2.
Enters AT mode with the built in AT mode baud rate of 38400. The baud rate cannot be changed by the user.
If you keep pin 34 HIGH you will enable the “full” AT mode which allows all AT commands to be used.
If you let pin 34 return LOW after power on then “mini” AT mode will be enabled.

Method 3.*
Enters “mini” AT mode using the user defined communication mode baud rate.

Method 4.*
Enters “full” AT mode using the user defined communication mode baud rate.

If pin 34 is kept HIGH then the HC-05 enters the “full” AT mode. If pin 34 is brought HIGH and returned to LOW it will put the module in to “mini” AT mode.

* added 21.07.2015

Method 1 and 2 are good in that you know the baud rate – it will always be 38400. This could be useful if you have modules other people have used or if you forget what communication mode baud rate you have previously set.

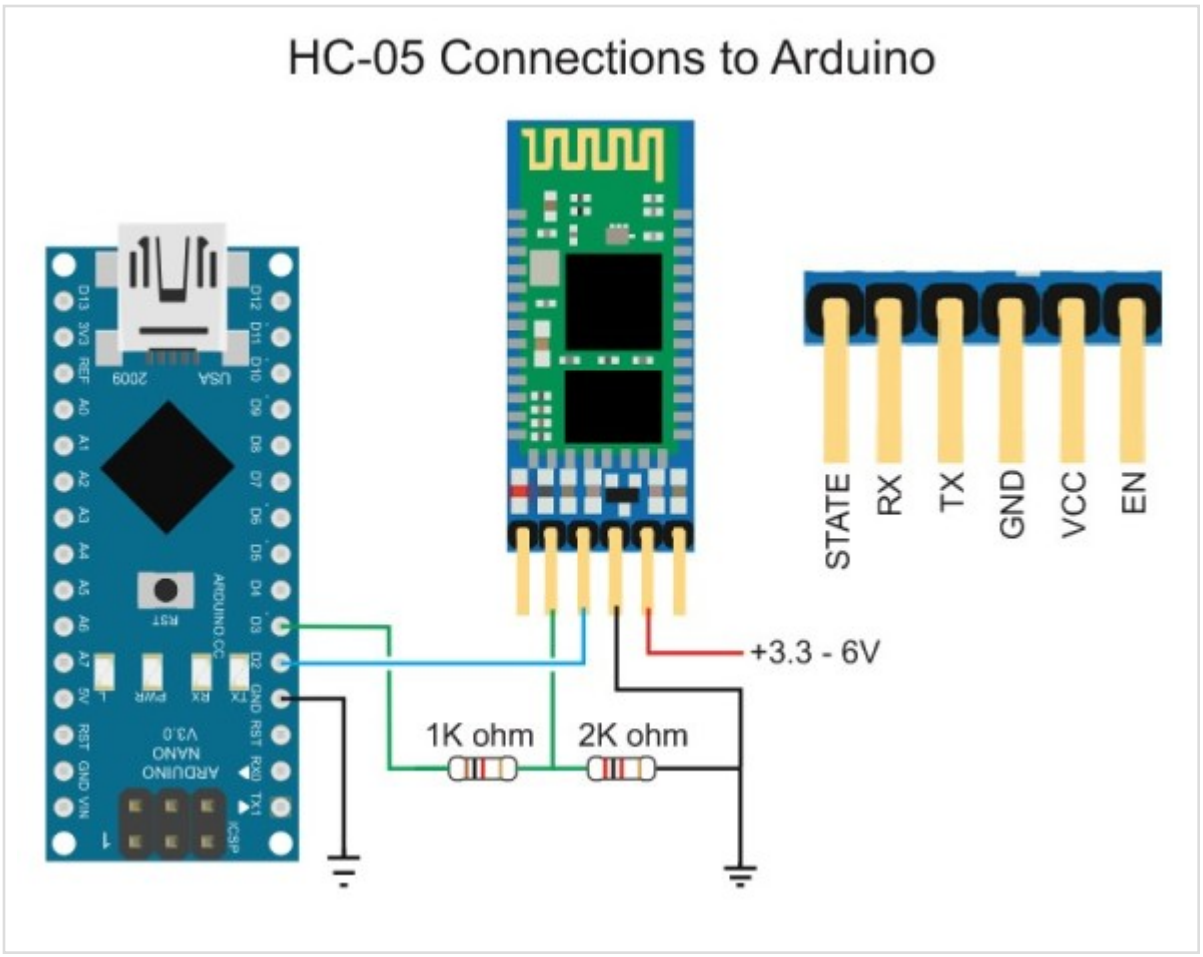
Method 3 and 4 adds convenience. You can enter AT mode, make changes and return back to communication mode without switching sketches and messing around with different baud rates.

I use software serial on Arduino pins 2 and 3 to talk to the HC-05. This means I can still use the hardware serial to talk to the serial monitor on a host computer.

Entering AT Mode Method 1. Use the button switch

The small push button switch, when closed, connects pin 34 to vcc which allows you to enter AT mode. Close the button switch when powering on the module. Once on you can release the button switch. However, this is not a full AT mode and some commands will not work. Commands such as AT+NAME?, AT+INQ, AT+RNAME? only work when pin 34 is HIGH. As soon as you release the button switch pin 34 returns LOW. If you want access to the extended commands you can simply close the switch just before issuing the AT command and release the button switch after the command has been sent.

Make the following connections
– BT VCC to Arduiono 5V
– BT GND to Arduino GND
– BT TX to Arduino D2
– BT RX to Arduino D3 through a voltage divider (3.3V)



Connect the Arduino to the host computer. The LED on the HC-05 should be blinking quickly at about 5 times a second.

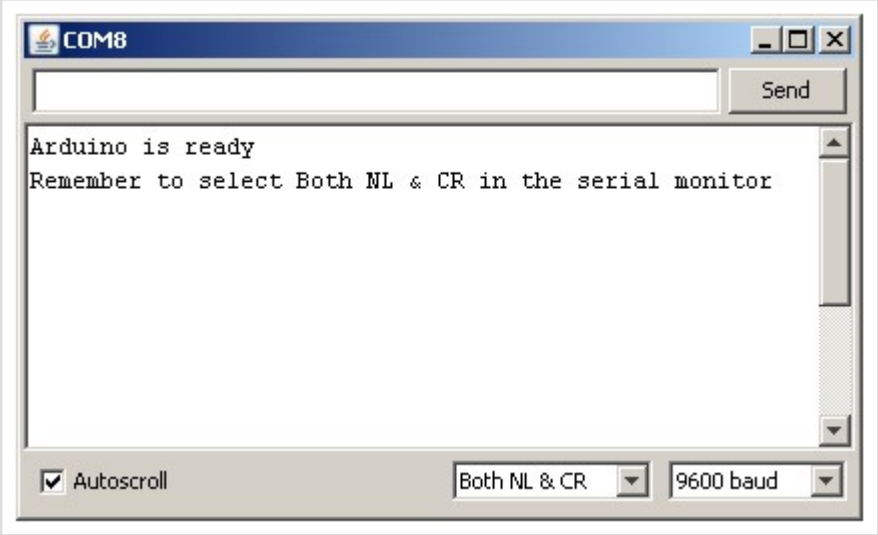
With the Arduino on, do the following
– Remove the 5V connection to BT VCC
– Press and hold the button switch on the BT module
– Re-connect BT VCC to 5V (while still pressing the button switch), the LED should come on.
– Release the button switch and the LED should be blinking slowly on/off once every couple of seconds. This indicates AT mode.

The following sketch is used to talk to the BT module. Run the sketch and put the HC-05 in to AT mode.

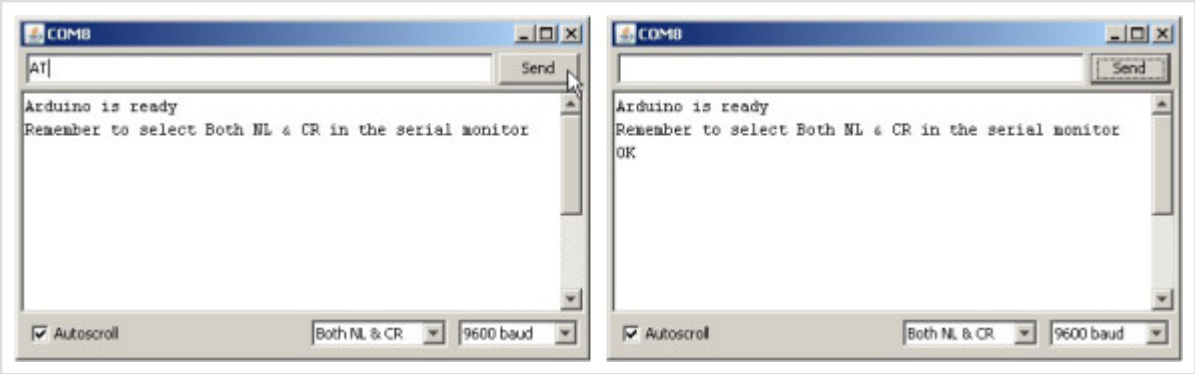
```
// Basic Bluetooth sketch HC-05_AT_MODE_01
// Communicate with a HC-05 using the serial monitor
//
// The HC-05 defaults to communication mode when first powered on you will need to manually enter A
// The default baud rate for AT mode is 38400
// See www.martyncurrey.com for details
//
```

```
#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3); // RX | TX
// Connect the HC-05 TX to Arduino pin 2 RX.
// Connect the HC-05 RX to Arduino pin 3 TX through a voltage divider.
//

char c = ' ';

void setup()
{
    Serial.begin(9600);
    Serial.println("Arduino is ready");
    Serial.println("Remember to select Both NL & CR in the serial monitor");

    // HC-05 default serial speed for AT mode is 38400
    BTserial.begin(38400);
}

void loop()
{

    // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTserial.available())
    {
        c = BTserial.read();
        Serial.write(c);
    }

    // Keep reading from Arduino Serial Monitor and send to HC-05
    if (Serial.available())
    {
        c =  Serial.read();
        BTserial.write(c);
    }

}
```

Here is the output on the serial monitor



The HC-05 expects a new line and a carriage return character at the end of each command so make sure “Both NL & CR” is selected at the bottom of the serial monitor. To confirm you are actually in AT mode, in the serial monitor type “AT” (no quotes) and hit Send. You should get an “OK”.



Entering AT Mode Method 2. Bring Pin 34 HIGH

If you do not wish the Arduino to control entering AT mode you can wire pin 34 to 3.3v, power on the HC-05 and then remove the 3.3v to pin 34. However, if you are doing this you might as well use the button switch.

This used to be my preferred method but I have found that once I have set up a module I seldom change it and it has become more convenient to use the button switch. The benefit of using a connection to pin 34 is once all the connections are made the Arduino can control the process. Also, certain commands only work when pin 34 is HIGH. Using this method allows you to control the state of pin 34 through the Arduino sketch.

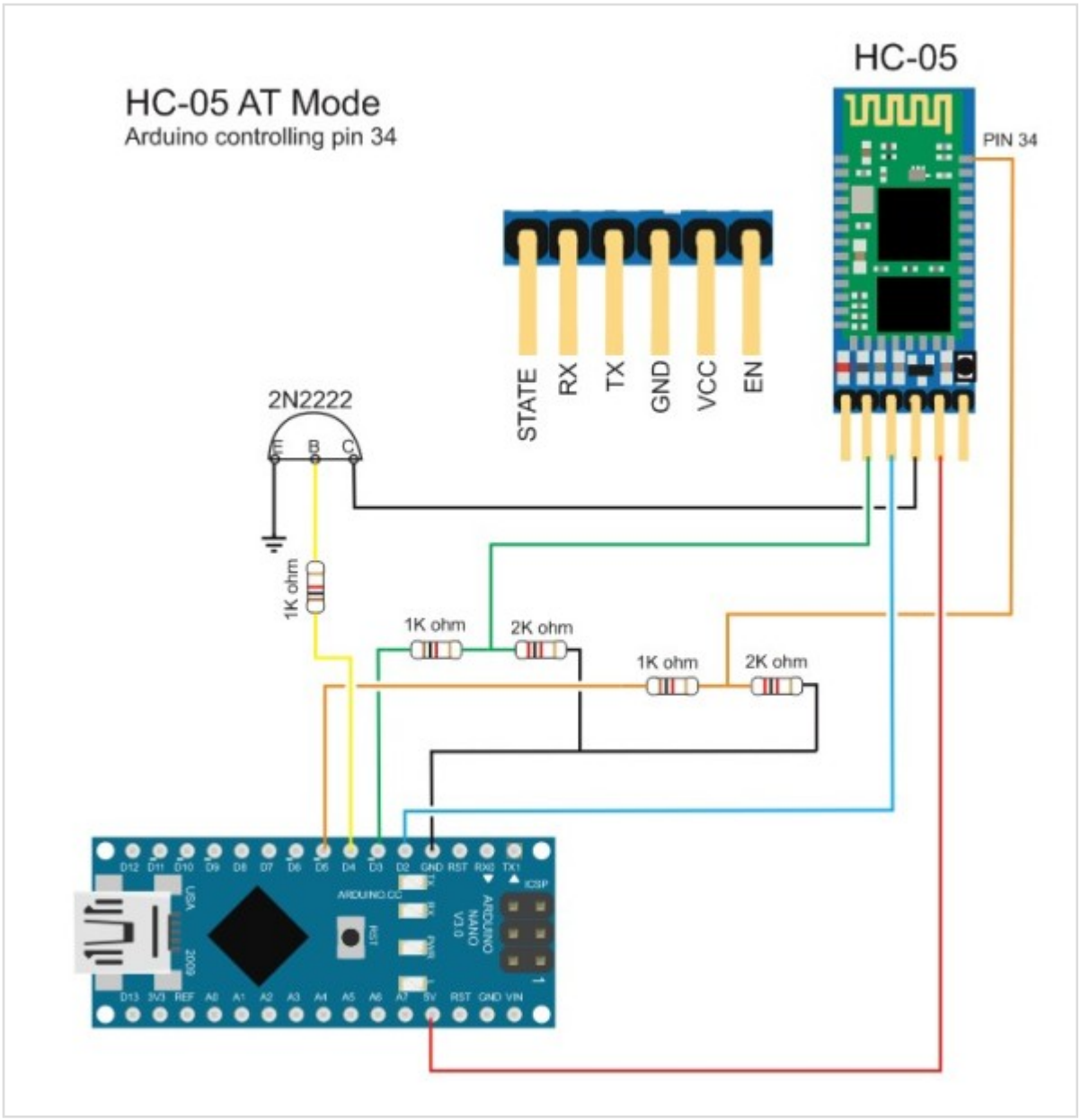
With the Arduino turned off make the following connections:
– BT VCC to 5V
– BT GND to pin C on a 2N222 transistor (or similar)
– BT TX to Arduino D2
– BT RX to Arduino D3 through a voltage divider
– 2N2222 B pin to 1K ohm resistor and then to Arduino pin D4
– 2N2222 E pin to GND
– Arduino D5 to a voltage divider and then to HC-05 pin 34

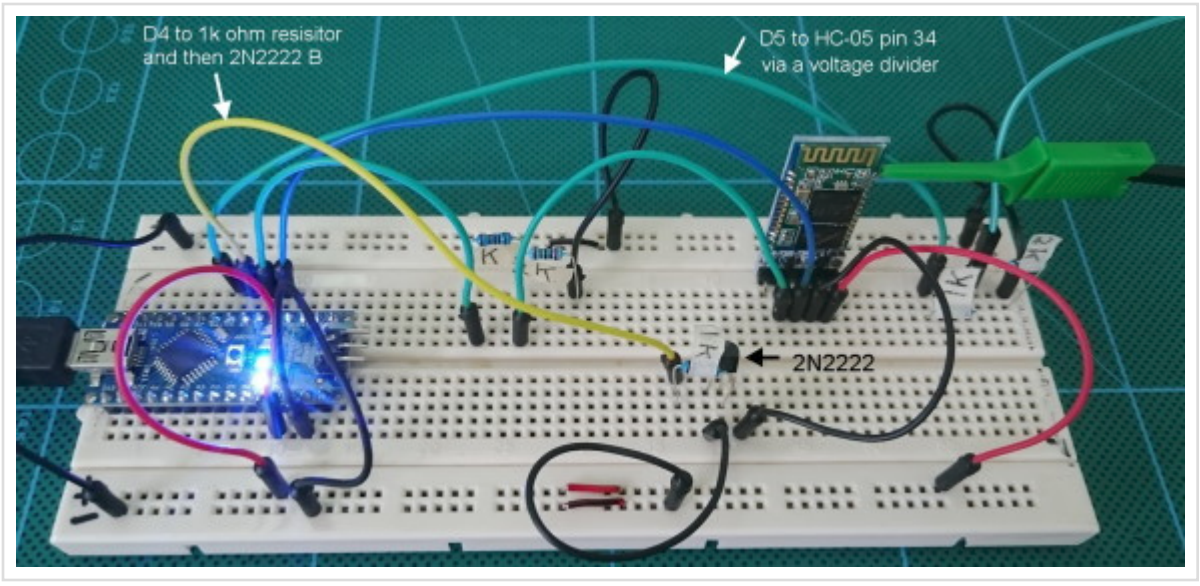
NOTE: I have previously powered the HC-05 directly from one of the Arduino digital pins and it seemed to work fine, however, you should not really do this. The HC-05 can draw up to 40mA which is too much for a regular Arduino pin to handle. Pins on the Arduino are rated for an absolute max of 40mA but the recommended is 20mA. The 5V pin out on the Arduino can handle up to 200mA so its OK to use the 5V out.

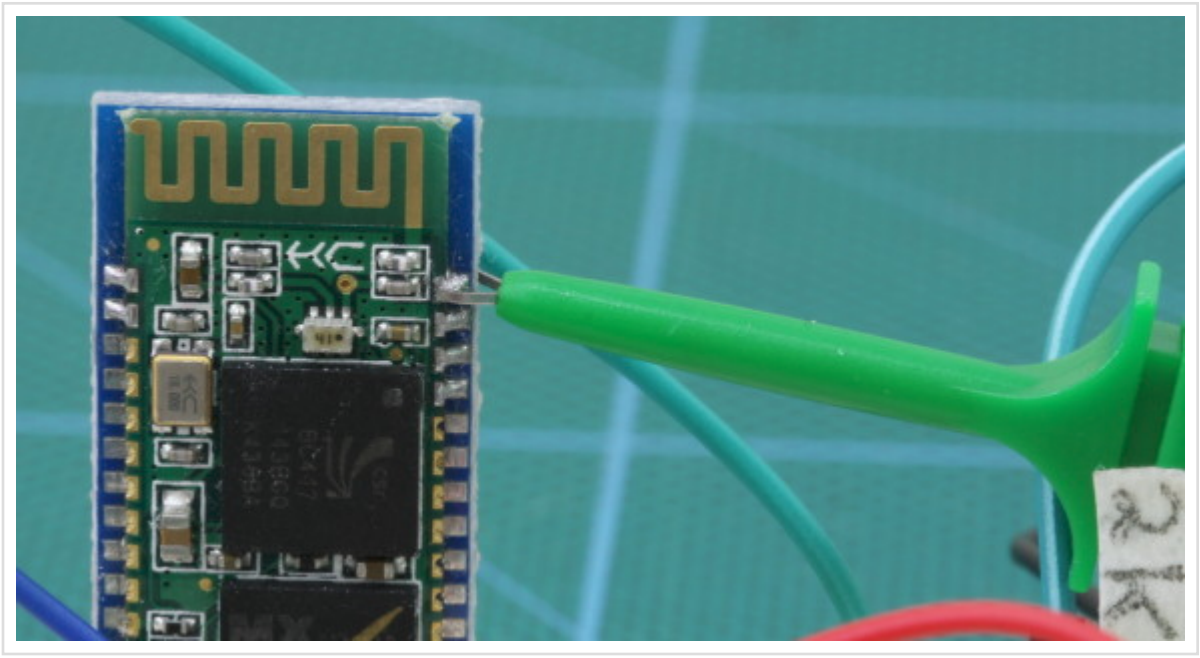
I have used a transistor as a switch. Connect the transistor between HC-05 and GND and then use the Arduino to turn the

transistor on and off and thus turn the HC-05 on and off.

Should be like this







HC-05 with pin 34 connected by a probe clip. You can also simply hold a piece of wire to the pin. For a long term solution you would need to solder a wire to pin 34.

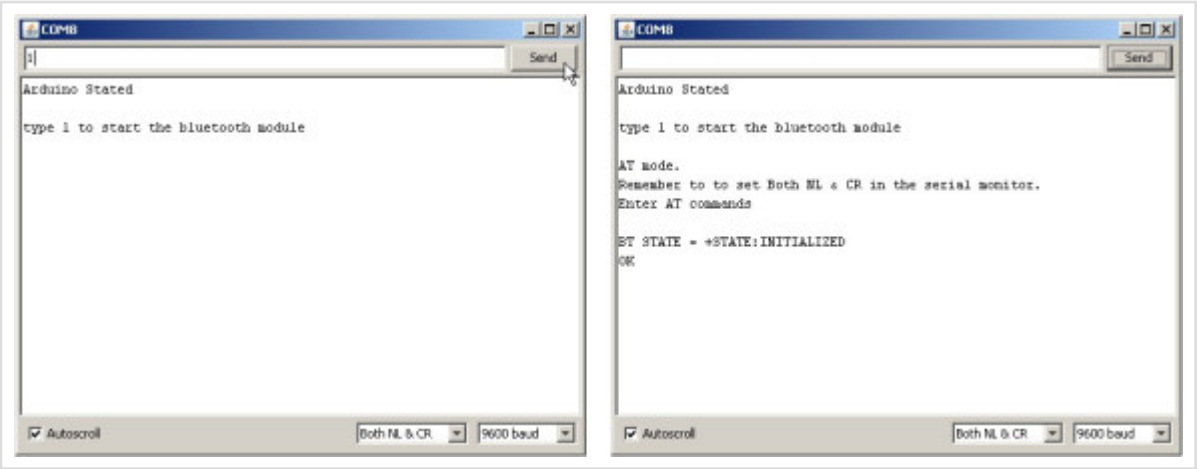
Compile and upload the following sketch

```
// Basic Bluetooth sketch HC-05_AT_MODE_02
// Connect the HC-05 module and communicate using the serial monitor
// Arduino automates entering AT mode
//
// The default baud rate for AT mode is 38400
// See www.martyncurrey.com for details
//
//
```

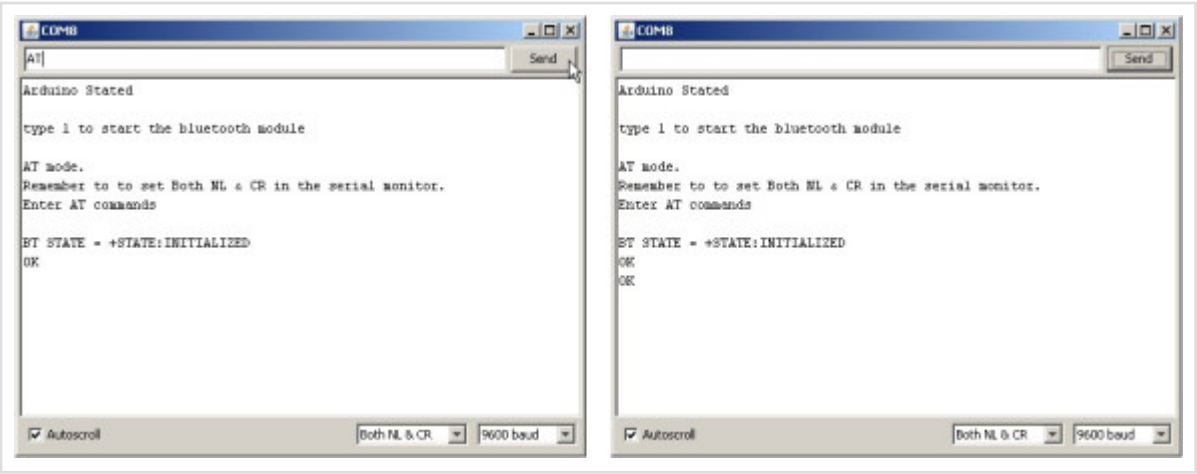
```
//  Pins
//  BT VCC to Arduino pin 3
//  BT GND to 2N2222 C
//  BT RX (through a voltage divider) to Arduino pin D3
//  BT TX  to Arduino pin D2 (no need voltage divider)
//  BT pin 34 (through a voltage divider) to Arduino pin D5
//  2N2222 B to 1K resistor then to D4
//  2N2222 E to GND
//
// When a command is entered in to the serial monitor on the computer
// the Arduino will relay it to the bluetooth module and display the result.
//
//
//  Process
//  Set pin 34 HIGH
//  Turn on the HC-05
//  send a test command
//

//  Use the LED on pin 13 to indicate when the HC-05 is in AT mode
const byte  LEDPIN = 13;
const byte BT_POWERPIN = 5;
const byte BT_PIN34 = 4;

char c = ' ';

#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3); // RX | TX

void setup()
{
    pinMode(LEDPIN, OUTPUT);
    pinMode(BT_POWERPIN, OUTPUT);
    pinMode(BT_PIN34, OUTPUT);

    // some modules require the BT pins to be LOW at first
    digitalWrite(BT_POWERPIN, LOW);
    digitalWrite(BT_PIN34, LOW);
    delay (500);

    Serial.begin(9600);     // communication with the host computer
    //while (!Serial)   { ; }

    Serial.println("Arduino Started");
    Serial.println("");
    Serial.println("type 1 to start the bluetooth module");

    while (c!='1')
    {
        if (Serial.available())
        {
            c =  Serial.read();
        }
    }
```

```
    // set pin 34 HIGH
    digitalWrite(BT_PIN34, HIGH);
    delay(100);

    // turn on the HC-05
    digitalWrite(BT_POWERPIN, HIGH);

    // show we have turned on the BT module
    digitalWrite(LEDPIN, HIGH);

    delay (500);

    // Can now turn off the power to pin 34.
    // Some commands only work when pin 34 is HIGH.
    // To use all commands comment out the next line
    //digitalWrite(BT_PIN34, LOW);


    // Start the software serial - baud rate for AT mode is 38400
    BTserial.begin(38400);


    Serial.println(" ");
    Serial.println("AT mode.");
    Serial.println("Remember to to set Both NL & CR in the serial monitor.");

    Serial.println("Enter AT commands");
    Serial.println("");

    Serial.print("BT STATE = ");
    BTserial.println("AT+STATE" );
}

void loop()
{

    // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTserial.available())
    {
        c = BTserial.read();
        Serial.write(c);
    }

    // Keep reading from Arduino Serial Monitor and send to HC-05
    if (Serial.available())
    {
        c =  Serial.read();
        BTserial.write(c);
    }

}
```

Open the serial monitor, enter a 1 and hit Send



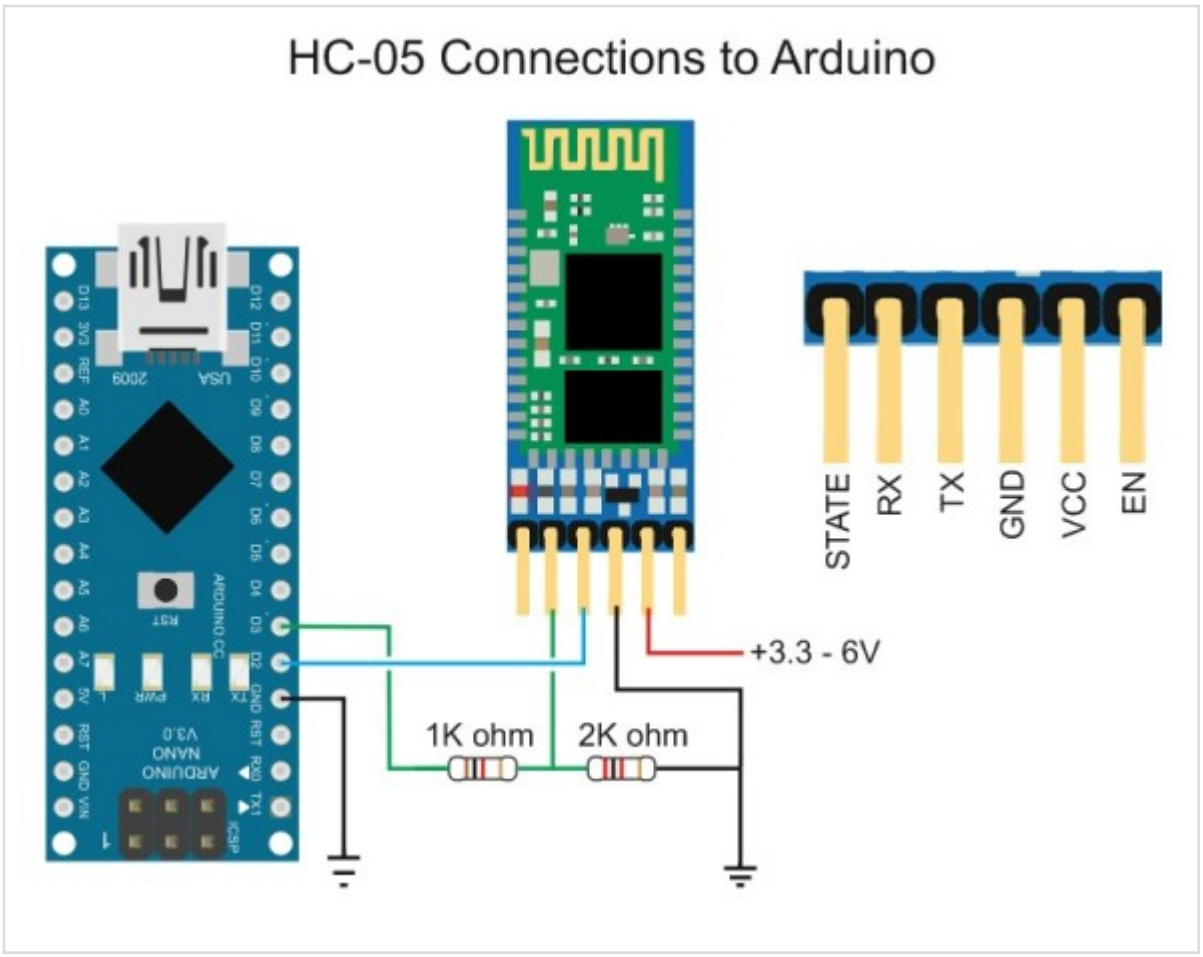
To confirm we are actually in AT mode enter AT and hit Send. If you do not get a reply enter AT again. Sometimes there are extra characters in the buffer.



Entering AT Mode Method 3. Close the small push button switch after the HC-05 is powered.

This method enters the “mini” AT mode using the baud rate defined for communication mode. This means you can use communication mode and enter AT mode without uploading a new sketch but not all commands will work.

Set up the HC-05 as below:



Upload the following sketch.

```
// Basic Bluetooth sketch HC-05_02_9600+ECHO
// Connect the HC-05 module and communicate using the serial monitor
//
// The HC-05 defaults to commincation mode when first powered on.
// The default baud rate for communication mode is 9600. Your module may have a different speed.
//

#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3); // RX | TX
// Connect the HC-05 TX to Arduino pin 2 RX.
// Connect the HC-05 RX to Arduino pin 3 TX through a voltage divider.

char c = ' ';

void setup()
{
    Serial.begin(9600);
    Serial.println("Arduino is ready");

    // HC-05 default serial speed for communication mode is 9600
    BTserial.begin(9600);
    Serial.println("BTserial started at 9600");
```

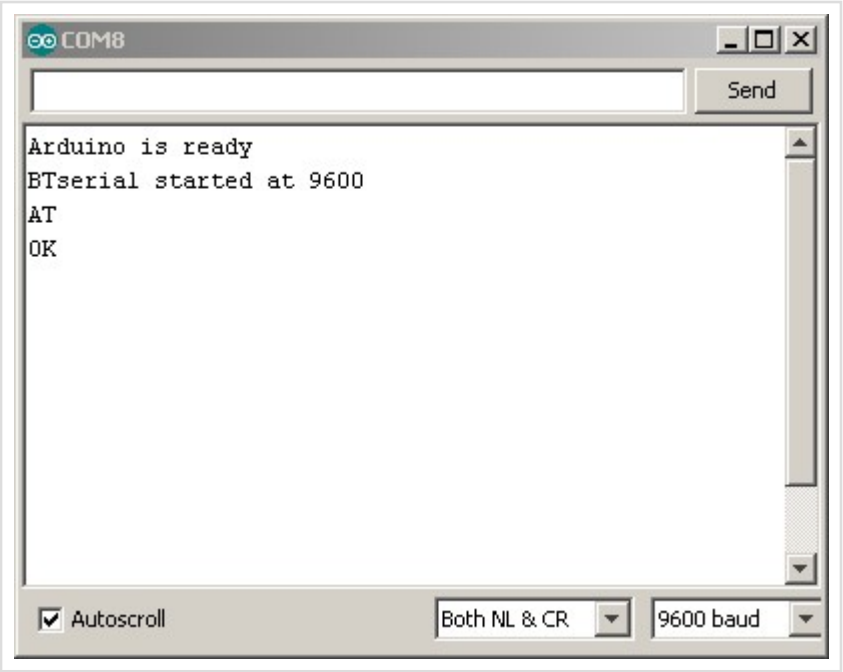
```
}

void loop()
{
    // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTserial.available())
    {
        c = BTserial.read();
        Serial.write(c);
    }

    // Keep reading from Arduino Serial Monitor and send to HC-05
    if (Serial.available())
    {
        c =  Serial.read();

        // Copy the serial data back to to the serial monitor.
        // This makes it easy to follow the commands and replies
        Serial.write(c);
        BTserial.write(c);
    }

}
```

The HC-05 should be in communication mode with the LED on the HC-05 blinking about 5 times a second. This indicates the module is waiting for a connection or to be paired.

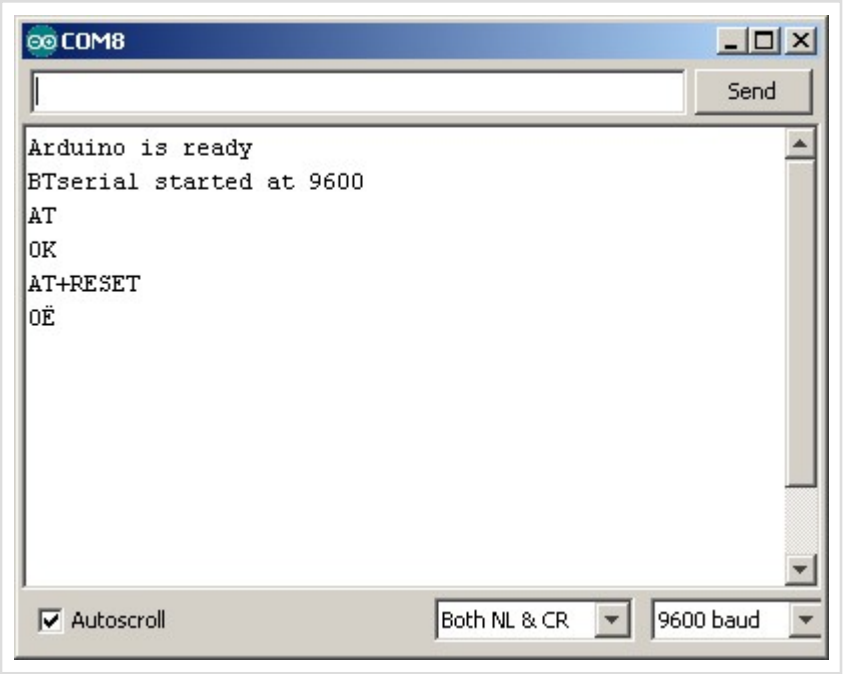
Open the serial monitor.
Press the small button switch.
Release the small button switch.

That's it. You are now in "mini" AT mode. The LED does not change. It still blinks quickly at 5 times a second.

In the serial monitor enter "AT" (no quotes) and hit Send. You should get an "OK"



To return to communication mode, enter the “AT+RESET” command.



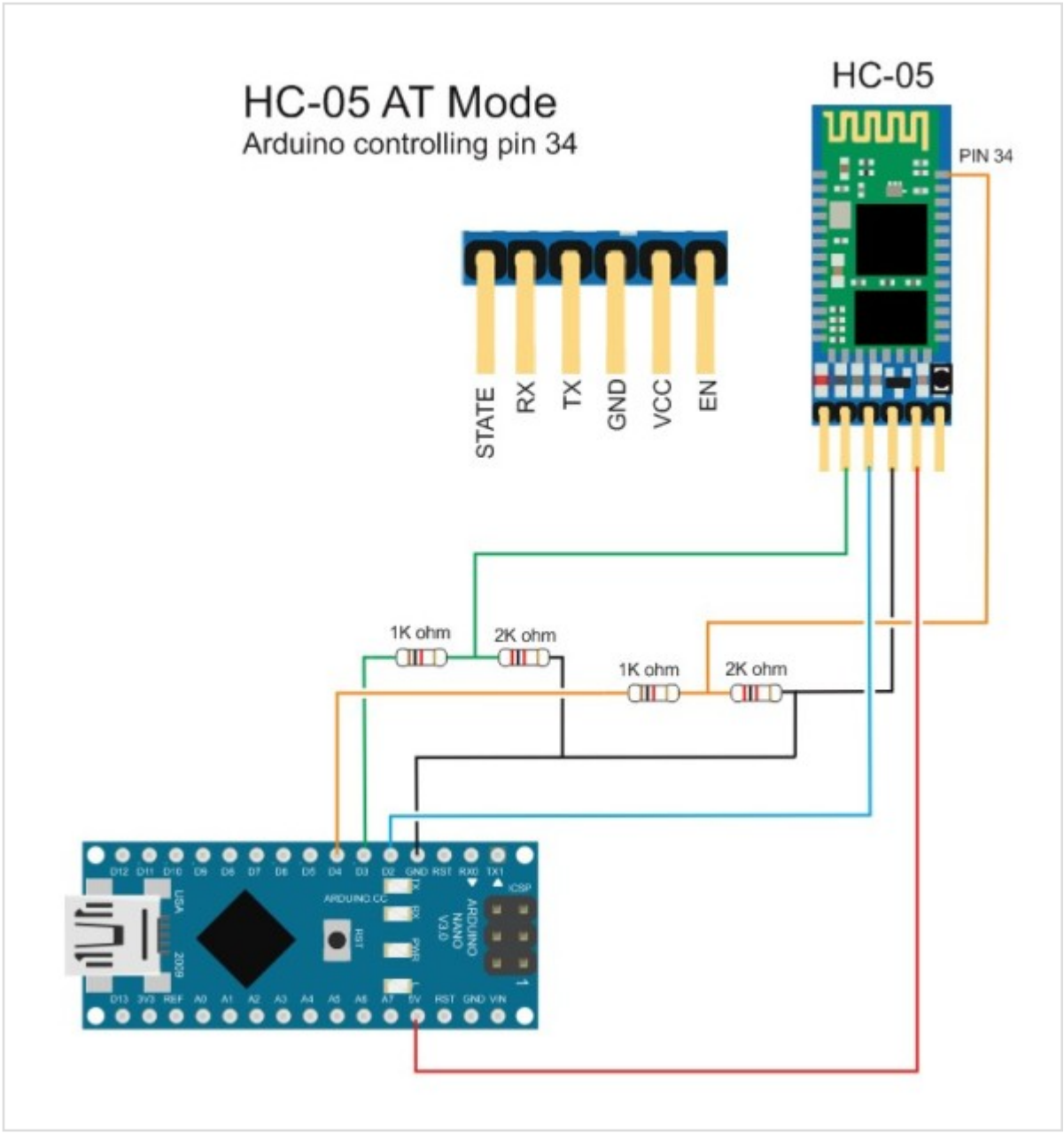
Entering AT Mode Method 4. Pull pin 34 HIGH after powering the HC-05.

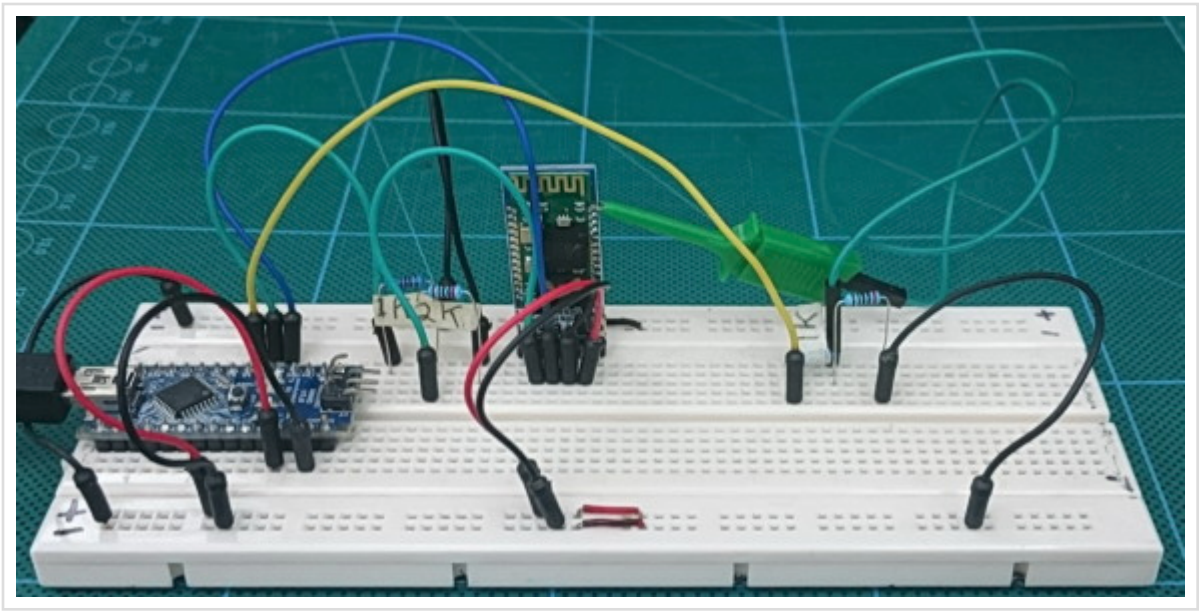
This method allows you to enter and exit AT mode from within a sketch. This can be handy when you run a set up sketch

or when you want the user to be able to make changes. For example change the name of the HC-05. The sketch could ask the user for the new name and then set it without the need to change sketches or any manual operation from the user.

Because pin 34 is being pulled HIGH the HC-05 will enter “full” AT mode and all commands will work.

Make the following connections
– D2 (RX) to HC-05 TX
– D3 (TX) to voltage divider and then HC-05 RX
– D4 to voltage divider and then to pin 34
– HC-05 vcc to 5v
– HC-05 GND to common GND


HC-05 AT Mode
Arduino controlling pin 34
HC-05
PIN 34
STATE
RX
TX
GND
VCC
EN
1K ohm
2K ohm
1K ohm
2K ohm



Upload the following sketch. Once the sketch is uploaded open the serial monitor.

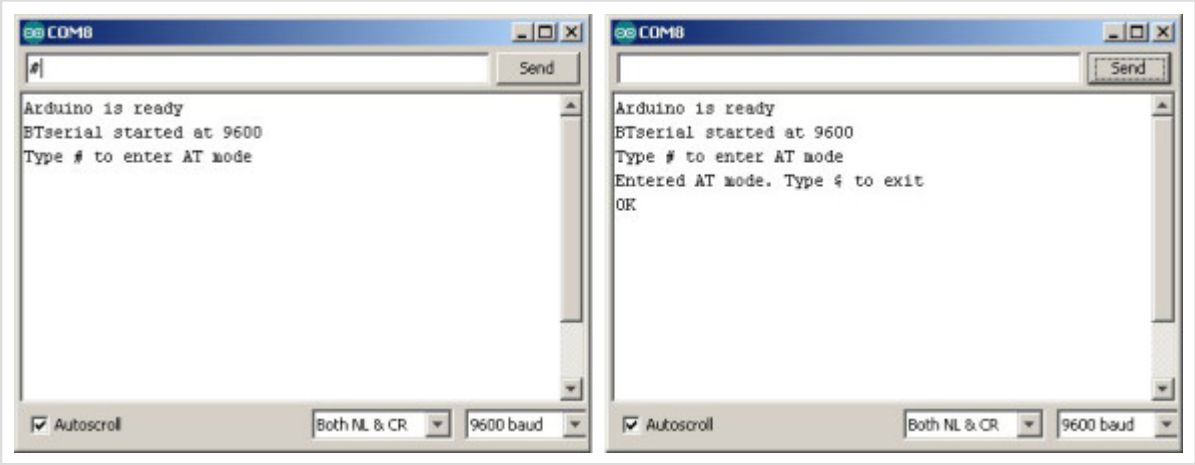
```
// Basic Bluetooth sketch HC-05_03_AT_MODE_Controlled
// Connect the HC-05 module and communicate using the serial monitor
//
// The HC-05 defaults to communication mode when first powered on.
// The default baud rate for communication mode is 9600. Your module may have a different speed.
// This sketch allows the user to enter AT mode on the HC-05
//
//
// Pins
// D2 (RX) to HC-05 TX
// D3 (TX) to voltage divider and then HC-05 RX
// D4 to voltage divider and then to pin 34
// HC-05 vcc to 5v
// HC-05 GND to common GND
//

#include <SoftwareSerial.h>
SoftwareSerial BTserial(2, 3); // RX | TX
// Connect the HC-05 TX to Arduino pin 2 RX.
// Connect the HC-05 RX to Arduino pin 3 TX through a voltage divider.

char c = ' ';
byte ATmodePin=4;

void setup()
{
    // set up the pin used to turn on AT mode
    pinMode(ATmodePin, OUTPUT);
    digitalWrite(ATmodePin, LOW);

    // Start the serial monitor
    Serial.begin(9600);
```

```
    Serial.println("Arduino is ready");

    // HC-05 default serial speed for communication mode is 9600
    BTserial.begin(9600);
    Serial.println("BTserial started at 9600");
    Serial.println("Type # to enter AT mode");
}

void loop()
{

    // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTserial.available())
    {
        c = BTserial.read();
        Serial.write(c);
    }

    // Keep reading from Arduino Serial Monitor and send to HC-05
    if (Serial.available())
    {
        c =  Serial.read();

        if (c=='#')  // enter AT mode
        {
            digitalWrite(ATmodePin, HIGH);
            Serial.print("Entered AT mode. Type $ to exit");
        }

        else if (c=='$') // exit AT mode by reseting the HC-05
        {
            digitalWrite(ATmodePin, LOW);
            BTserial.print("AT+RESET\n\r");
            Serial.print("AT+RESET\n\r");
        }

        else
        {
              // Copy the serial data back to to the serial monitor.
              // This makes it easy to follow the commands and replies
              Serial.write(c);
              BTserial.write(c);
        }

    }

}
```

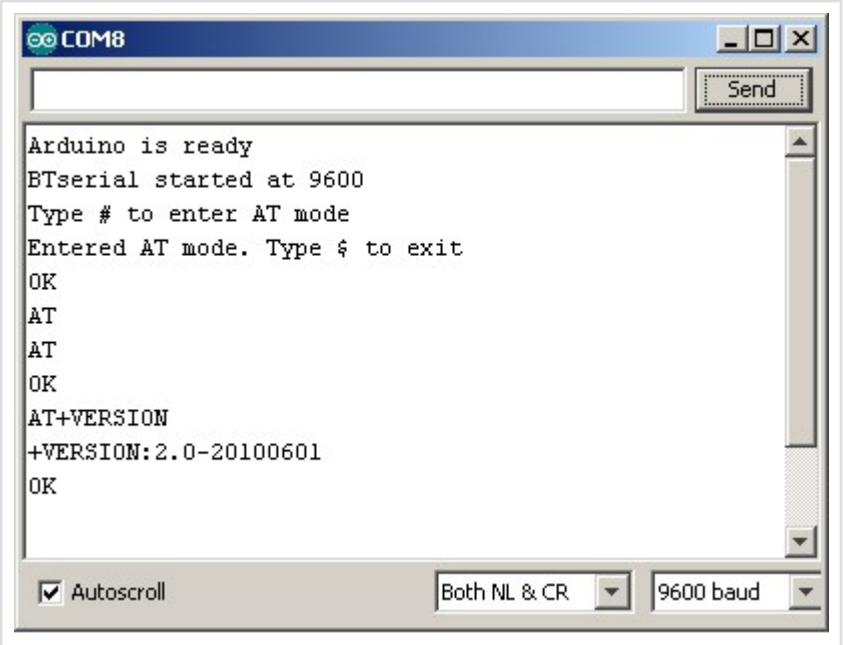
The sketch is similar to the other HC-05 sketches, it copies what is entered in to the serial monitor to the HC-05 and what ever it receives from the HC-05 it sends to the serial monitor. The extra bit is # and $. Entering a “#” (no quotes) puts the HC-05 in to AT mode and entering a “$” (no quotes) returns to communication mode. It returns the HC-05 to communication mode by resetting the module.

To enter AT mode type “#”

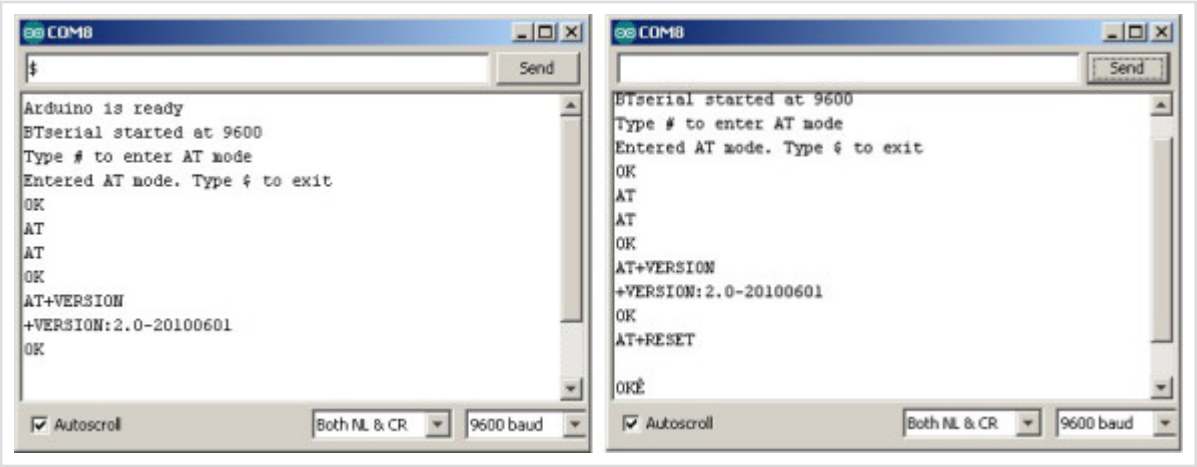
To exit AT mode type “$”



You can confirm you are in AT mode by entering “AT” or “AT+VERSION”



To leave At mode type “$” (no quotes)



AT commands

The HC-05 expects commands to include a carriage return and newline characters (\r\n). You can add these automatically in the serial monitor by selecting Both NL & CR at the bottom of the window.

You can also enter them manually in the form AT\r\n. If you forget to add carriage return and newline characters the HC-05 will not respond.

Example commands

AT – simple feedback request. Will return
OK

AT+VERSION – returns the firmware version. In my case it returns
+VERSION:2.0-20100601
OK

AT+STATE – returns the current state of the module
+STATE:INITIALIZED
OK

AT+ROLE – the possible values are ; 0 – Slave, 1 – Master, 2 – Slave-Loop

Returns
+ROLE:0
OK

To change to Master Mode, enter AT+ROLE=1, returns:
OK

AT+UART – returns the baud rate used by the HC-05 in communication mode. The default for the modules I have is 9600.
Returns:
+UART:9600,0,0
OK

To change the baud rate to 38400 – AT+UART=38400,0,0
Returns:
OK

Windows does not support baud rates above 115200. If you accidentally set the baud rate higher than 115200 you will not be able to use communication mode. You should still be able to enter AT mode at 38400 using method 1 or method 2 above and change the communication mode baud rate to something Windows can handle.

AT+NAME
Querying the modules name with AT+NAME? only works in “full” At mode. If you cannot get AT+NAME? to work you need to bring pin34 HIGH.
Changing the modules name with AT+NAME=newname works in “full” AT mode and “mini” AT mode.

What you should get is:
AT+NAME?, returns
+NAME:HC-05
OK
(or something similar depending what your module is called)

Other commands that require pin 34 to be HIGH are AT+INQ and AT+RNAME. This is not a complete list through.

Full list of AT commands

AT COMMAND LISTING

	COMMAND	FUNCTION
1	AT	Test UART Connection
2	AT+RESET	Reset Device
3	AT+VERSION	Query firmware version
4	AT+ORGL	Restore settings to Factory Defaults
5	AT+ADDR	Query Device Bluetooth Address
6	AT+NAME	Query/Set Device Name
7	AT+RNAME	Query Remote Bluetooth Device's Name
8	AT+ROLE	Query/Set Device Role
9	AT+CLASS	Query/Set Class of Device CoD
10	AT+IAC	Query/Set Inquire Access Code
11	AT+INQM	Query/Set Inquire Access Mode
12	AT+PSWD	Query/Set Pairing Passkey
13	AT+UART	Query/Set UART parameter
14	AT+CMODE	Query/Set Connection Mode
15	AT+BIND	Query/Set Binding Bluetooth Address
16	AT+POLAR	Query/Set LED Output Polarity
17	AT+PIO	Set/Reset a User I/O pin
18	AT+MPIO	Set/Reset multiple User I/O pin
19	AT+MPIOY	Query User I/O pin
20	AT+RSCAN	Query/Set Scanning Parameters
21	AT+SNIFF	Query/Set SNIFF Energy Savings Parameters
22	AT+SENM	Query/Set Security & Encryption Modes
23	AT+RMAAD	Delete Authenticated Device from List
24	AT+FSAD	Find Device from Authenticated Device List
25	AT+ADCN	Query Total Number of Device from Authenticated Device List
26	AT+MRAD	Query Most Recently Used Authenticated Device
27	AT+STATE	Query Current Status of the Device
28	AT+INIT	Initialize SPP Profile
29	AT+INQ	Query Nearby Discoverable Devices
30	AT+INQC	Cancel Search for Discoverable Devices
31	AT+PAIR	Device Pairing
32	AT+LINK	Connect to a Remote Device
33	AT+DISC	Disconnect from a Remote Device
34	AT+ENSNIFF	Enter Energy Saving mode
35	AT+EXSNIFF	Exit Energy Saving mode

ERROR CODES

ERROR CODE	VERBOSE
0	Command Error/Invalid Command
1	Results in default value
2	PSKEY write error
3	Device name is too long (>32 characters)
4	No device name specified (0 length)
5	Bluetooth address NAP is too long
6	Bluetooth address UAP is too long
7	Bluetooth address LAP is too long
8	PIO map not specified (0 length)
9	Invalid PIO port Number entered
A	Device Class not specified (0 length)
B	Device Class too long
C	Inquire Access Code not Specified (0 length)
D	Inquire Access Code too long
E	Invalid Inquire Access Code entered
F	Pairing Password not specified (0 length)
10	Pairing Password too long (> 16 characters)
11	Invalid Role entered
12	Invalid Baud Rate entered
13	Invalid Stop Bit entered
14	Invalid Parity Bit entered
15	No device in the Pairing List
16	SPP not initialized
17	SPP already initialized
18	Invalid Inquiry Mode
19	Inquiry Timeout occurred
1A	Invalid/zero length address entered
1B	Invalid Security Mode entered
1C	Invalid Encryption Mode entered

This list is taken from the EGBT-045MS bluetooth module user guide and not all commands may be supported or work straight away. For example AT+NAME? only works when pin 34 is HIGH.

For more information look at the HC-05 user guide or the EGBT-046S/EGBT-045MS user guide

This entry was posted in **Arduino**, **Bluetooth** and tagged **arduino**, **AT mode**, **Bluetooth**, **HC-05** by **Martyn**. Bookmark the **permalink [http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-at-mode/]** .

65 THOUGHTS ON “ARDUINO WITH HC-05 (ZS-040) BLUETOOTH MODULE – AT MODE”

montassar
on **February 13, 2015 at 3:17 pm** said:

Thank you so much , you’re article is the only that answered my question about how to get the zs40 into AT mode.

Brian
on **July 23, 2015 at 4:00 pm** said:

+1 I agree with montassar. I was not able to get all the AT com-

mands until I read this posting. The at+name? was making me nuts. Now, I hold the button and the command works.

Thanks!

MEDo
on **March 4, 2015 at 9:31 pm** said:

Hi there, I have tried both ways and it did not work what you think could be the problem?

PABLO
on **March 4, 2015 at 9:38 pm** said:

hey, i have this problem, the comunication it's not going well, it send to me characters like this: Ï§†…. I tried a lot of ways and it's always the same, :
AT mode.
Remember to to set Both NL & CR in the serial monitor.
BT STATE =
««ª ª'•ªFU
b•´ŠÄ……Ï§†…
i think it is a problem in the comuniction between the arduino whit my module HC-05 zs404

MEDo
on **March 4, 2015 at 9:47 pm** said:

This is what happens when you have the wrong baud rate set for BT-serial.

PABLO
on **March 4, 2015 at 11:01 pm** said:

YES the led blink, every thing seems to be allright except for the Ï§†

> **Martyn**
> on **March 5, 2015 at 5:22 am** said:
>
> It sounds like you have the wrong baud rate.
>
> The modules I have use different speeds; AT mode is 38400, and communication mode is 9600.
>
> // Start the software serial – baud rate for AT mode is 38400
> BTserial.begin(38400);
>
> Remember there are 2 different serial communications.
> 1 – The arduino to the computer at 9600
> 2 – The blue tooth module to the Arduino at 38400
>
> The baud rate shown in the serial monitor is for the communication between Arduino and computer.

PABLO
on **March 5, 2015 at 9:34 pm** said:

this is the last scetch i'm using. i also tried change the baud ratio in the BTserial, i tried whith 1200, 2400, 4800, 9600 etc, but i only have "response" using 38400. I even used the AltsoftSerial library and diden't work. I check in the net and some other few people has the same problem but no one seems find a solution.

char serialByte = '0';
const byte LEDPIN = 13;

#include

```
SoftwareSerial BTserial(10, 11); // RX | TX

void setup()
{
pinMode(LEDPIN, OUTPUT);

// communication with the host computer
Serial.begin(9600);

// Start the software serial – baud rate for AT mode is 38400
BTserial.begin(38400);

// LED to show we have started the serial channels
digitalWrite(LEDPIN, HIGH);

// Give the user time to enter AT mode
// wait for the user to enter a '1' to start
Serial.println("After entering AT mode, type 1″);
while (serialByte !='1′)
{
serialByte = Serial.read();
}

Serial.println(" ");
Serial.println("AT mode.");
Serial.println("Remember to to set Both NL & CR in the serial monitor.");

Serial.print("BT STATE = ");
BTserial.println("AT+STATE" );
Serial.println(" ");
delay(100);

}

void loop()
{
// listen for communication from the BT module and then write it to the serial
monitor
if ( BTserial.available() ) { Serial.write( BTserial.read() ); }

// listen for user input and send it to the HC-05
```

if (Serial.available()) { BTserial.write(Serial.read()); }
}

and this its what happen when i executed it

After entering AT mode, type 1

AT mode.
Remember to to set Both NL & CR in the serial monitor.
BT STATE =
««ª ª'•ªFE
b•´ŠÄ……Ï§†…

Martyn
on **March 14, 2015 at 4:44 am** said:

I can't duplicate the problem. I do get some garbage when I use an incorrect baud rate but it is intermittent. As soon as I use 38400 everything works fine.

Try different pins, for example use Pin 2 and 3 and reseat all the connections.

Mohammad Hasan
on **April 26, 2015 at 4:25 pm** said:

try to remove the arduino IDE and reinstall it again, then go the same steps you made .

Mohammad Hasan
on **April 26, 2015 at 4:27 pm** said:

try to remove the arduino IDE and reinstall it again.

Fibo
on **May 26, 2015 at 6:01 am** said:

hey, did you find the solution to the problem? I got the exact same result as yours on my serial monitor. Thanks in advance!

Martyn
on **May 26, 2015 at 6:18 am** said:

The closest I have been to recreating the problem is through using the wrong baud rate.

Try the different baud rates and note the responses you get. One of then should work but it may not be 38400.

Duke Tan
on **April 1, 2015 at 3:50 pm** said:

Hi, I got two problems with this HC-05 module.
1. When I type AT+NAME? I got no response. I do not understand why is that so, I got all responses for AT+ROLE? AT+PSWD? but not AT+NAME? Can answer this ?

2. I can't search this HC-05 Bluetooth module with my ios devices. I can find it with other android devices but not my iphone or ipad. How to solve ?

Martyn
on **April 3, 2015 at 4:14 am** said:

Updated reply.

AT+NAME? only works when pin34 is HIGH.
If using the button switch, close the switch just before sending the command.

> The HC-05 and HC-06 do not work with idevices. Apple products do not support the Bluetooth protocol that the HC-05 /06 units use. If you want to connect to an idevice you will need a BT module that uses BLE / Bluetooth 4.0.

mehmeteray
on **May 5, 2015 at 10:04 pm** said:

how to configure master mod? how to pair android?

Martyn
on **May 6, 2015 at 4:42 am** said:

What do you mean master mode? If you mean AT mode then see above.

To know how to pair, see http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module/

mehmeteray
on **May 6, 2015 at 3:52 pm** said:

AT+INIT and AT+INQ not responding.

> **Martyn**
> on **June 22, 2015 at 7:08 am** said:
>
> You need to make pin 34 HIGH.
>
> Either connect pin 34 directly to 3.3v or close the little button switch just before you send the command. The button switch connects pin 34 to vcc 3.3v when closed

ram
on **July 4, 2015 at 5:37 am** said:

we are using ARDUINO MEGA and we also encountered the same problem after entering “AT” then “1”, the serial monitor respond is “BT STATE”. we tried “38400” and “9600” but didnt get any response.

but when we used ARDUINO UNO. we got an “OKAY” respond from the serial monitor.

WHY IS THAT SO?
we really need to use the ARDUINO MEGA

any help?

Martyn
on **July 4, 2015 at 6:00 am** said:

The mega has additional hardware serial channels (I think it has 4 in total). Instead of using software serial use one of the other hardware channels. The additional hardware serials are on pins 14 to 19.

See https://www.arduino.cc/en/Tutorial/MultiSerialMega on setting it up.

ram
on **July 4, 2015 at 7:53 am** said:

is it similar when configuring ZS 040 using arduino NANO?

Martyn
on **July 5, 2015 at 5:54 am** said:

I have just added a intro guide for the Arduino Mega. Hope this helps.

ram
on **July 7, 2015 at 4:24 pm** said:

very much appreciated
thank you :)

Aiman
on **July 11, 2015 at 6:07 pm** said:

Hello Martyn! Thank you for sharing your work. I have successfully connected my uno with hc05 and due with hc05 too and have paired them using AT+LINK Both the hc05 have started blinking together now i want to transfer data from uno to due I have connected a microphone with uno and here is the receiver code of uno :
#include

```
#define rxPin 11
#define txPin 10
SoftwareSerial myserial= SoftwareSerial(rxPin,txPin);
int analogpin=0;
void setup(){
Serial.begin(9600);
myserial.begin(9600);
}
void loop(){

int b=analogRead(analogpin);
Serial.println(b);
delay(1000);
myserial.write(b);

}
```

And on due i have used this code

```
void setup()
{
Serial.begin(9600);
Serial1.begin(9600);
)
void loop()
{
int a=Serial1.read();
Serial.println(a);
}
```

But the data appearing on the serial monitor of uno is not going on due .
Can you please help its urgent

Martyn
on **July 13, 2015 at 10:00 am** said:

Confirm that they are actually talking to each other. You can do this with 2 serial monitors. You can also set one to send test data; for example “1234” every second.

shar

on **July 22, 2015 at 9:11 am** said:

hi, how did you paired the two HC 05 bluetooth modules?

my serial monitor always respond “FAIL” during the “AT+LINK” command.

i need help thank you:)

ram
on **July 18, 2015 at 6:03 pm** said:

Starting the bluetooth module

AT mode.
Remember to to set Both NL & CR in the serial monitor.
Enter AT commands

BT STATE = +STATE:INITIALIZED
OK
ERROR:(16)
OK
+INQ:3014:8:190258,1F00,7FFF
OK
+RNAME:CARDEA_SLAVE
OK
FAIL
+INQ:3014:8:190258,1F00,7FFF
OK

any idea why do we get a “FAIL” response? thank you

Martyn
on **July 19, 2015 at 8:55 am** said:

RNAME requires the mac address of the remote device: See page 10 of the EGBT-046S/EGBT-045MS user guide. You can download

the user guide from https://docs.google.com/file/d/0BxdLxDCD6Hid-SkRaRTVuNERrQjg/edit

7. Query Remote Bluetooth Device's Name

Command
AT+RNAME?<addr>

Response
+NAME:<name>
OK

where <name> = Device name
<addr> = 48 bit bluetooth address
in NAP,UAP,LAP format

Example: Query remote Bluetooth device having address = 00:02:72:0A:3C:7F
Bluetooth address in NA:UAP:LAP format = 0002:72:0A3C7F
From Host controller:
AT+RNAME?0002,72,0A3C7F
EGBT-045MS response if remote device name is "HC-05"
+NAME:HC-05
OK
EGBT-045MS response if remote device name is unresolved
FAIL

I have a HC-06 module with the address = 30:14:10:17:11:79
This means the command = AT+RNAME?3014,10,171179
and this gets me:
+RNAME:HC-06_1
OK

The HC-06 is name "HC-06_1″

RNAME only works for me when pin 34 on the HC-05 is HIGH. When pin 34 is LOW or not connected I get nothing, no reply and no error message.

Are you able to use AT+INQ and AT+RNAME with pin 34 LOW or not connected?

On the modules I have I have to set pin 34 HIGH before the commands give a reply.

ram
on **July 20, 2015 at 7:07 am** said:

the serial monitor responds properly AT+INQ and AT+RNAME, pin 34 is HIGH.
we only get “FAIL” respond evertytime we use AT+LINK.

we cannot pair the master and slave module

ram
on **July 20, 2015 at 9:12 am** said:

the serial monitor is not responding in AT+INIT, AT+INQ, AT+LINK when we are using the METHOD 1. USE THE BUTTON SWITCH

any help? thanks

Martyn
on **July 20, 2015 at 9:20 am** said:

You need to keep pin 34 HIGH. Certain commands like AT+INQ etc need pin 34 to be HIGH. If pin 34 is LOW or not connected then the module does not respond (no reply and no error message).

This means keeping the button switch closed or adding a connection to pin 34 directly. The button switch, when closed, connects pin 34 to +3.3v

I am going to solder a wire to pin 34 to make switching HIGH/LOW a bit easier to do. A wire connected to the top of the button switch should also work.

I have updated the guide and the new information may be helpful to you.

Pingback: Connecting 2 Arduinos by Bluetooth using a HC-05 and a HC-06: Easy Method Using CMODE | Martyn Currey

shar
on **July 22, 2015 at 9:05 am** said:

do you have an idea on how to pair two HC 05 bluetooth modules using the command “AT+LINK”? thanks :)

Martyn
on **July 23, 2015 at 5:03 am** said:

I will try to do a write up on this but in the meantime here are the basic steps

Pin34 on the HC-05 needs to be HIGH for all commands to work.

1. make sure the baud rates on the HC-05 and the HC-06 are the same
2. make sure the passwords on the HC-05 and the HC-06 are the same
3. find the address of the slave module
4. pair with the slave device
5. bind the slave device
6. set the HC-05 to only connect with paired devices
7. link to the slave device

1 and 2 are self-explanatory

3. find the address of the slave module. You can do this using the HC-05:

Turn on the HC-06. Turn on the HC-05

On the HC-05 do the following:
– Clear any previously paired devices – AT+RMAAD
– Put the HC-05 in Master Mode – AT+ROLE=1
– After changing the mode you may need to reset the module – AT+RESET
– Allow the HC-05 to connect to any device – AT+CMODE=0
– Set inquiry to search for 5 devices and 9 seconds – AT+INQM=0,5,9
– Initiate the SPP profile – AT+INIT (if SPP is already active you will get an error(17) which you can ignore)
– Search for other devices – AT+INQ

This will bring up a list of found devices. One of them should be the HC-06.
The reply should be in the format +INQ:address,type,signal
The address will show something like
+INQ:15:FF:F3241B,1F00,7FFF

To use the address in AT commands you need to change the colons into commas. If you get more than one found device you check the names with: AT+RNAME?15,FF,F3241B

Once you have confirmed you have the correct module and correct address you need to pair it with the HC-05

4. pair with the slave device – AT+PAIR=<addr>,<timeout>
e.g. AT+PAIR=15,FF,F3241B,9
If the HC-05 cannot pair with the HC-06 within 9 seconds you will get an error message.

5. bind the slave device – AT+BIND=<address>

6 set the HC-05 to connect to bound devices only – AT+CMODE=1

7 Link to the slave device – AT+LINK=<address>

Once set, on start up the HC-05 should automatically connect to the HC-06.

Let me know how you get on.

shar
on **July 25, 2015 at 10:03 am** said:

it did work BUT…

once i turn OFF the the MASTER module, and then i turn it on again, the MASTER module WOULD NOT CONNECT automatically to the SLAVE module.

HOW CAN I AUTOMATICALLY THE MASTER TO THE SLAVE MODULE?

THANK YOU VERY MUCH FOR YOUR HELP :)

Martyn
on **July 25, 2015 at 4:01 pm** said:

I have just written up the guide. See http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-pair-bind-and-link/

Martyn
on **July 26, 2015 at 6:05 am** said:

I have just added a video to the http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-pair-bind-and-link/ post.

The video shows the HC-05 automatically connecting to the HC-06. Have tried this many times now and I seldom have a problem. They do appear to connect more quickly if the HC-06 is turned on first.

ram

on **July 30, 2015 at 5:27 pm** said:

than you for your blog martyn! its really a great help :)

Pingback: Connecting 2 Arduinos by Bluetooth using a HC-05 and a HC-06: Pair, Bind, and Link | Martyn Currey

peppermint
on **July 29, 2015 at 5:49 am** said:

Hello, do you happen to know any way in which an hc-05 would suddenly malfunction (as in not respond to AT commands and serial communication problems)? Or have you heard of any such cases?

I've already gone through several hc-05's.
I have absolutely no idea why this happens as there isn't any commonality between the malfunctioning (it doesn't happen as a direct result of something – one moment, it works and when I try again later on it just suddenly doesn't).

Martyn
on **July 29, 2015 at 6:03 am** said:

I haven't come across this. My modules have been fairly robust – dropped on the floor, connected the wrong way, etc.

If the LED is working normally then you know the module is still alive?

Are you using a voltage divider on the RX pin? You can damage the mini Bluetooth module by connecting 5V directly to it.

peppermint
on **July 30, 2015 at 10:24 am** said:

Turns out I was sending 5v signals to BT Rx pin despite connecting 3.3v to the vcc. I guess that was slowly damaging the modules. Hopefully no more of them will die.

Thank you for the advice!

renren
on **July 30, 2015 at 5:26 pm** said:

hi martyn,

i have read from other source that it is possible to connect 2 slaves to only 1 master module if your bluetooth module supports “multi instance SPP”

do you have an idea on how to do this? and how to determine if your bluetooth module supports the “multi instance SPP” ?

thank you :)

Martyn
on **July 31, 2015 at 12:33 pm** said:

sorry, I have never done this. It does sound interesting and if possible may be an easy way to create a small network. Not sure when I will get time to investigate though.

Celso Ferrarini
on **August 7, 2015 at 2:59 am** said:

Hello,
Thanks for sharing fine information.
Instructions for Arduino UNO, it only worked on ports 10-11 9600 x 9600
I have the HC-05 zs-040 with the button on breakout board over EN Pin.

First you have to program arduino UNO with the code below, without the data pins plugged otherwise, it does not upload, then you connect data cables to pins 10rx-11 tx with voltage divider and power arduino with the module fully connected, then just open the terminal and push the small button on the module for some seconds, the led continues to flash fast as before (I didn´t manage to work with it blinking slow) all the commands work, some you have to hold the small button while tapping enter. Hope it throws some light to others with the same problem

```
// Basic Bluetooth sketch HC-06_01
// Connect the Hc-06 module and communicate using the serial monitor
//
// The HC-06 defaults to AT mode when first powered on.
// The default baud rate is 9600
// The Hc-06 requires all AT commands to be in uppercase. NL+CR should not be added to the command string

#include
SoftwareSerial BTserial(10, 11); // RX | TX
// Connect the HC-06 TX to the Arduino RX on pin 2.
// Connect the HC-06 RX to the Arduino TX on pin 3 through a voltage divider.
//

void setup()
{
Serial.begin(9600);
Serial.println("Enter AT commands:");

// HC-06 default serial speed is 9600
BTserial.begin(9600);
}

void loop()
{
```

```
// Keep reading from HC-06 and send to Arduino Serial Monitor
if (BTserial.available())
{
Serial.write(BTserial.read());
}

// Keep reading from Arduino Serial Monitor and send to HC-06
if (Serial.available())
{
BTserial.write(Serial.read());
}

}
```

Douglas
on **August 28, 2015 at 8:41 pm** said:

Hello,
I am unable to connect to the AT mode, only the third method worked but the settings are reduced. I have a GW-040 model, used this tutorial to be identical. Does anyone know any specific tutorial for this model (GW-040) or that works.

Martyn
on **August 29, 2015 at 2:30 am** said:

I'm not familiar with the GW-040 boards and I can't find anything on-line. Can you post a link to the ones you purchased.

It does appear that the same breakout board is now being used for different bluetooth modules. I have some new boards marked as FC-114. These use the same breakout board but run a different firmware and act very differently to the zs-040s.

Is it a HC-05 and does it have 6 pins?
Does the breakout board look to be the same as the zs-040?

Have you tried bringing pin 34 HIGH?

What firmware is the module using?

Michael
on **September 27, 2015 at 10:39 pm** said:

Hey Martyn
I just bought two HC-05 boards marked as FC-114, and in no way I can put it in Full AT mode, do you have new post to solve this problem
Thanks
Michael

Martyn
on **September 28, 2015 at 4:51 am** said:

The FC-114 boards I have have the Bolutek firmware which starts in AT mode. See http://www.martyncurrey.com/hc-05-fc-114-and-hc-06-fc-114-first-look/ and http://www.martyncurrey.com/hc-05-fc-114-and-hc-06-fc-114-part-2-basic-at-commands/.

Bianca
on **September 4, 2015 at 1:56 pm** said:

Dear Martyn!
THANK YOU SOOOO MUCH!!!! I was trying to get my HC-05 into AT-Mode for 2 days and your tutorial was the only one which worked :-D!

Cheers
Bianca

Martin M
on **September 13, 2015 at 6:17 pm** said:

Hi, I'm trying to connect my zs-040 module with my android phone using the AT+LINK command but the only response i get is FAIL. All other commands are working. what could be the problem with AT+LINK=AA,70,39F04A

Martyn
on **September 14, 2015 at 2:40 am** said:

You didn't say but I am assuming you have the HC-05.

I'm not 100% certain but I believe you have to initiate the connection from the phone.

Are you able to establish a connection from the phone to the BT module when using the phone to initiate the connection?
If you have the HC-05, does the phone appear when you scan for bluetooth devices using the HC-05?

Martin M
on **September 18, 2015 at 2:08 pm** said:

Sorry for the late response. Yes I'm using the HC-05 module. I can see the module on my phone's Bluetooth device list even before pairing but I'm not able to initiate a connection using my phone. I can only pair using the module after searching for my phone's address and using the AT+PAIR command. After pairing, my phone shows "paired but not connected". I want it to show something like "Connected"
My main interest is this command AT+LINK. Under what conditions does it work? It always shows FAIL every time i try.

Martyn
on **September 19, 2015 at 3:39 am** said:

After you have paired the devices you need to use a third party software to actually make a connection.

Have a look at http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-in-slave-mode/. Here I use and android app called Bluetooth Terminal. It is the app that makes the connection but it can only connect to paired devices.

I have never tried using the HC-05 to make a connection to an android device, I have always used the Android device to initiate the connection, when I get time I will try some experiments. I'm very busy at work at the moment so not sure when this will be.

Linking has to be done in a certain order; BIND, CMODE, LINK. See http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc-06-pair-bind-and-link/ for details.

Martin M
on **September 28, 2015 at 1:17 pm** said:

Thank you very much

Pingback: HC-05 and HC-06 zs-040 Bluetooth modules. First Look | Martyn Currey

Pingback: Quick Links | Martyn Currey

Daniel
on **October 2, 2015 at 12:37 am** said:

Hi Martyn, first of all i wanna tell you it's great to have all this information about hc 05. Thanks for sharing the information needed. I have a problem and hope you can help me, when sending an AT command, I can only send one at a time, and after that, no value is return, I have to disconnect the VCC

from BT and reconnect it to send another AT command, what you think can be the solution ?.

Martyn
on **October 2, 2015 at 6:49 am** said:

Can you explain what you mean by one at a time. Do you mean the first command works but the second,third etc does not?
Are you sure you are in AT mode?
Does “AT” work, do you get the “OK” reply?

What model HC-05 do you have? Do you have the zs-050?
Do you have the correct line endings set?
Can you make contact with another device, for example an Android device?

Daniel
on **October 3, 2015 at 12:54 am** said:

Exactly, first command works but the second,third etc doesnt.
im sure that AT mode is on cause led blinks each 2 seconds (aprox), and when send an AT command i ge an OK, but only the firts time.
I got a zs-040, and i already tried some AT comands. Also i tried to connect with my android, and it works.

Martyn
on **October 3, 2015 at 5:33 am** said:

Do you have “Both NL & CR” set in the serial monitor?

using sketch “Basic Bluetooth sketch HC-05_02_9600+ECHO” what do you get in the serial monitor?

Daniel
on **October 3, 2015 at 10:56 pm** said:

I Used the sketch “Basic Bluetooth sketch HC-05_02_9600+ECHO” and works great! Thanks Martyn, you are Awesome!

> **Martyn**
> on **October 4, 2015 at 2:11 am** said:
>
> Glad to hear you got it working

Gerardo Hernandez
on **October 9, 2015 at 7:32 pm** said:

Great article. It works great.