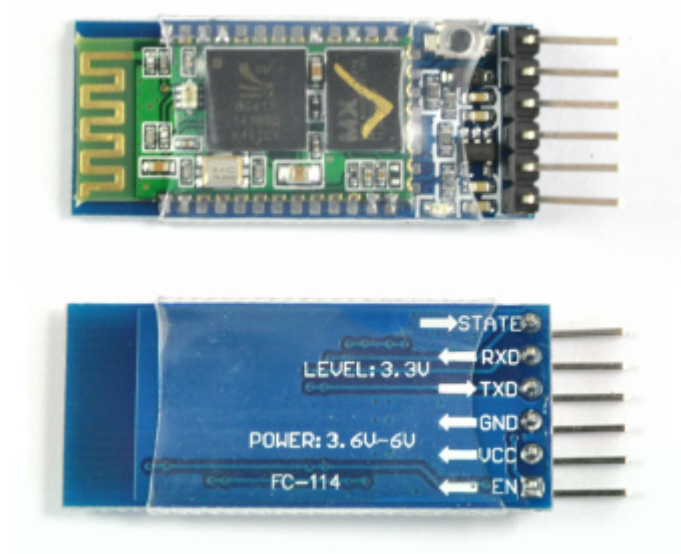


BLUETOOTH COMMUNICATION TROUGH HC05 MODULES

For this project, we have decided to allow communication between STM32F407 and Arduino board trough two HC-05 modules.

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication.



COMMAND and DATA TRANSFER MODES:

The module has two modes of operation: Command Mode where we can send AT commands to it, and Data Mode where it transmits and receives data to another Bluetooth module.

Default Settings

The default settings for new modules are

- Name = HC-05
- Password = 1234
- Baud rate in communication mode = 9600*
- Baud rate in AT/Command mode = 38400

*sometimes 38400

The default mode is DATA Mode, and it is used to transmit data over Bluetooth communication.

~~The default configuration is:~~

~~Baud Rate: 9600 bps, Data: 8 bits, Stop Bits: 1 bit, Parity: None, Handshake: None~~

~~Passkey: 1234~~

Device Name: HC-05

AT command mode allows you to interrogate the Bluetooth module and to change some of the settings: 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.

For the project, we need to change some of the configuration setup values in AT command mode.

There are two ways to operate in Command Mode: using a USB to TTL adapter, or using an Arduino board.

Link the hc-05 following this scheme (don't plug +5V pin for now)

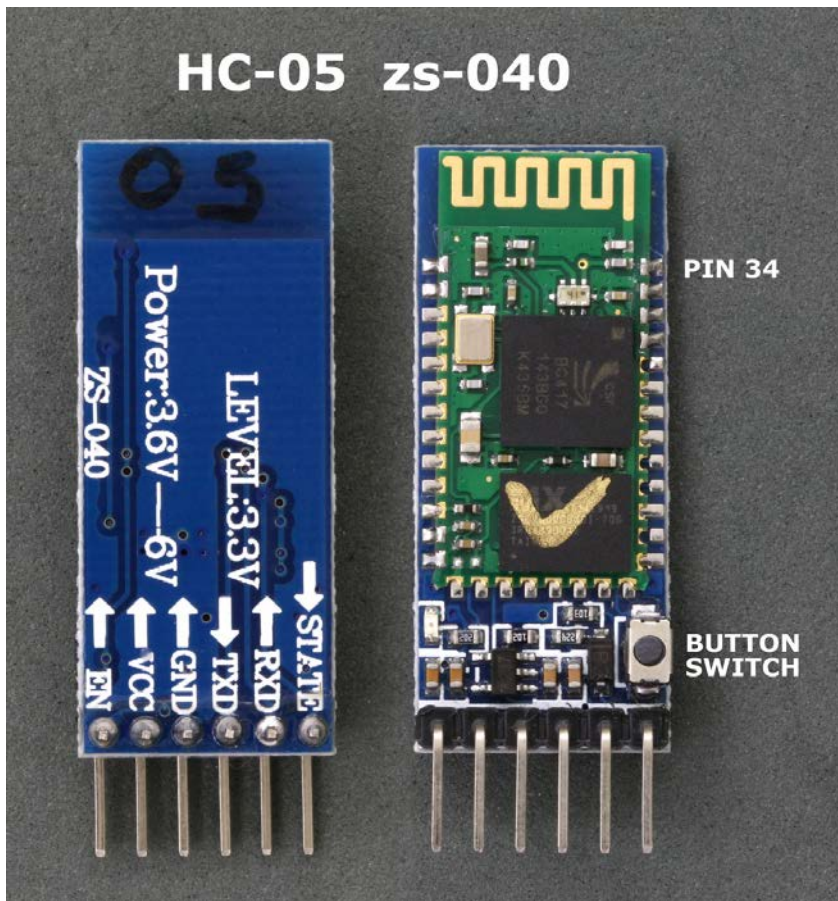
HC05	USB to TTL	Arduino
+5V →	Vcc	Vcc
Rx →	Tx	Tx
Tx →	Rx	Rx
Gnd →	Gnd	Gnd

HOW TO ENTER COMMAND MODE IF YOUR HC05 HAS KEY PIN

HC05	USB to TTL	Arduino
Key →	Vcc	3.3V

KEY: This pin has to be pulled high to enter AT mode.

HOW TO ENTER COMMAND MODE IF YOUR HC05 HAS EN PIN INSTEAD OF KEY PIN



To activate AT mode on these HC-05 modules, pin 34 needs to be HIGH on power up. The small push button switch connects pin 34 to +3.3v so we can either:

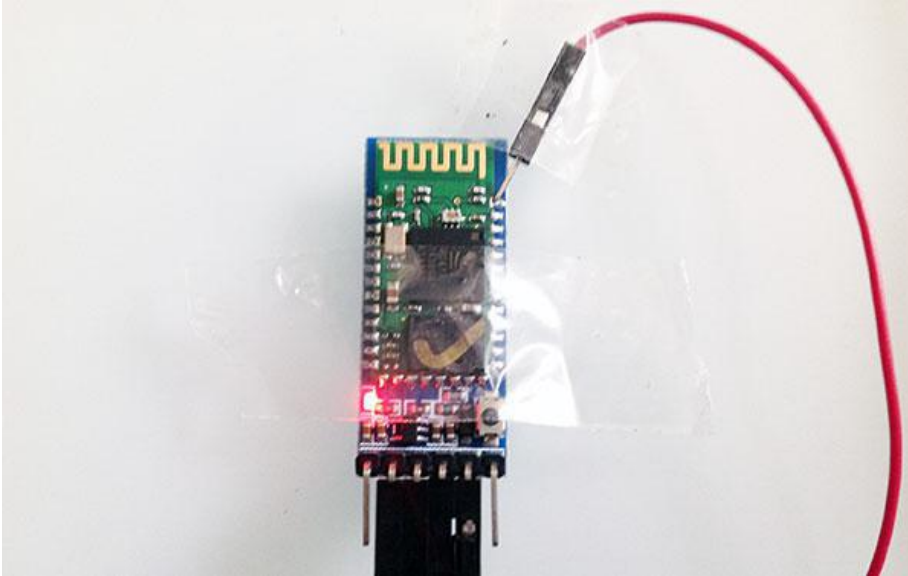
- connect pin 34 directly to +3v3v and power on, or
- hold the button switch closed when starting the module.

(METHOD 1) Using the button switch to enter AT command mode using 38400 baud rate:

1. Remove power from the module
2. Hold the small button switch closed while powering on the module.
3. Press and hold the button switch.
4. While still holding the button switch closed, apply power.
5. When you see the LED come on you can release the button switch.

If the Bluetooth module led is flashing every 2 seconds (slower than usual) that means that we have successfully entered in the AT command mode.

(METHOD 2) Using pin 34 to enter full AT command mode using 38400 baud rate.



1. Remove power from the module
2. Make a connection between pin 34 and +3.3v
3. Reapply power.

If the Bluetooth module led is flashing every 2 seconds (slower than usual) that means that we have successfully entered in the AT command mode.

SETUP THE COMMUNICATION

Final scheme:

Stm32 → hc-05 (master) ↔ bluetooth communication ↔ hc-05 (slave) → Arduino

Once the HC-05 is in Command mode, we can use “Arduino Serial Monitor” to set the HC-05 module.

After we run the Serial Monitor, we need to select “Both NL and CR”, as well as, “38400 baud” which is the default baud rate in command mode. Now, we are ready to send commands.

HOW TO PAIR THE TWO MODULES

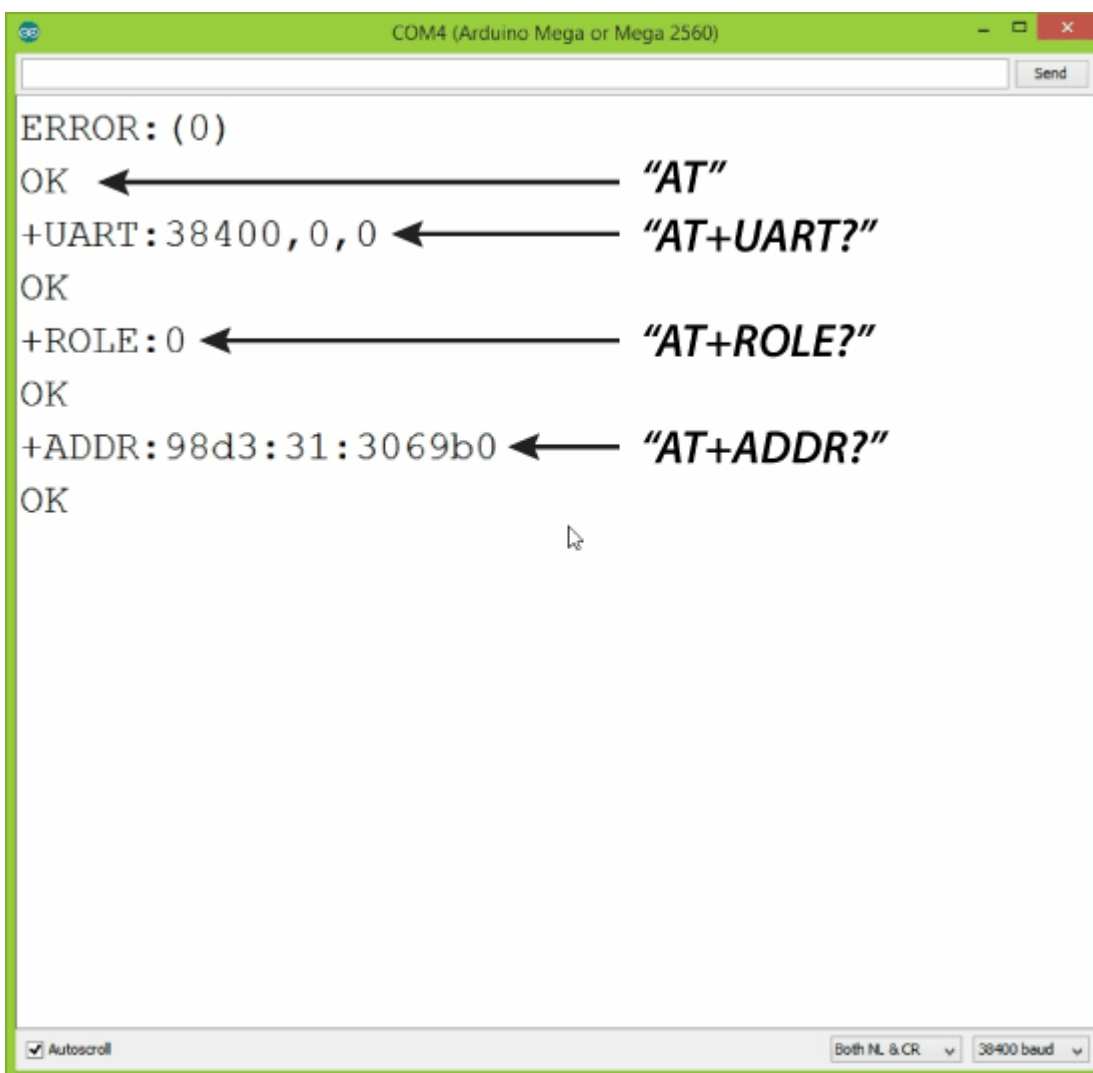
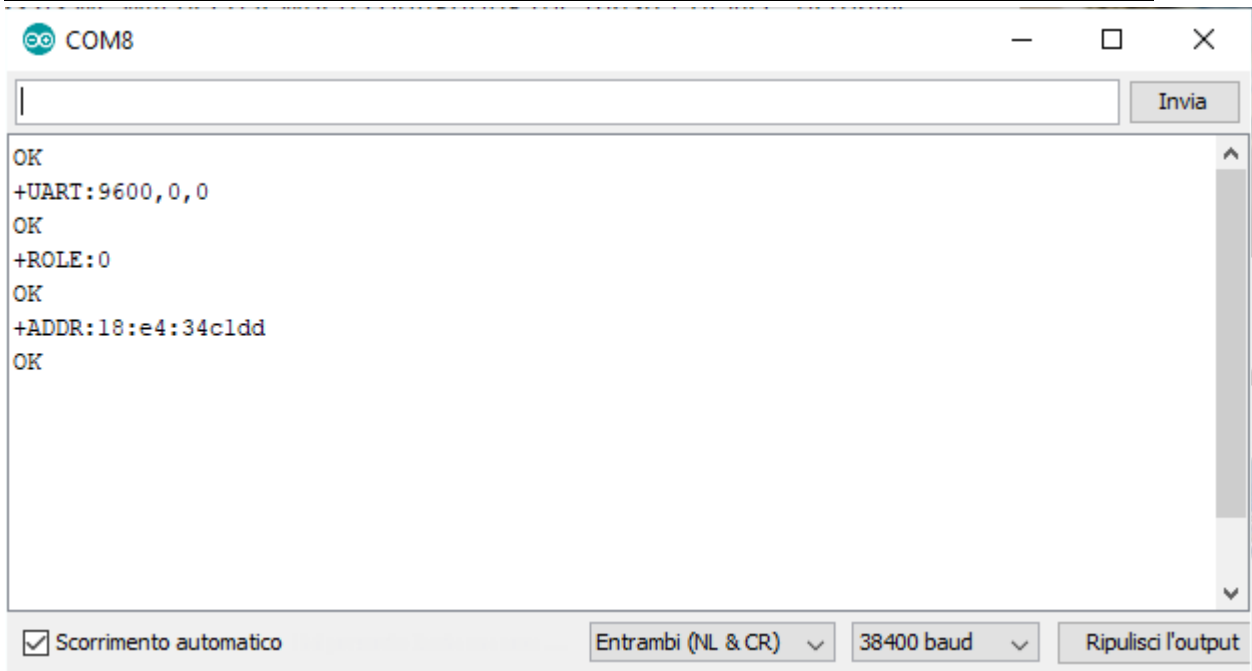
SLAVE CONFIGURATION (this hc-05 module will be used on the Arduino Board)

If we type just “AT” which is a test command we should get back the message “OK”.

Then if we type “AT+UART?” we should get back the message that shows the default baud rate which is 9600. (REMEMBER, we will use the same baud rate on the Arduino board)

Then if we type “AT+ROLE?” we will get back a message “+ROLE=0” which means that the Bluetooth device is in slave mode. If we type “AT+ADDR?” we will get back the address of the Bluetooth module and it should look something like this: 18:e4:34cldd.

Now we need to write down this address, as we will need it when configuring the master device.



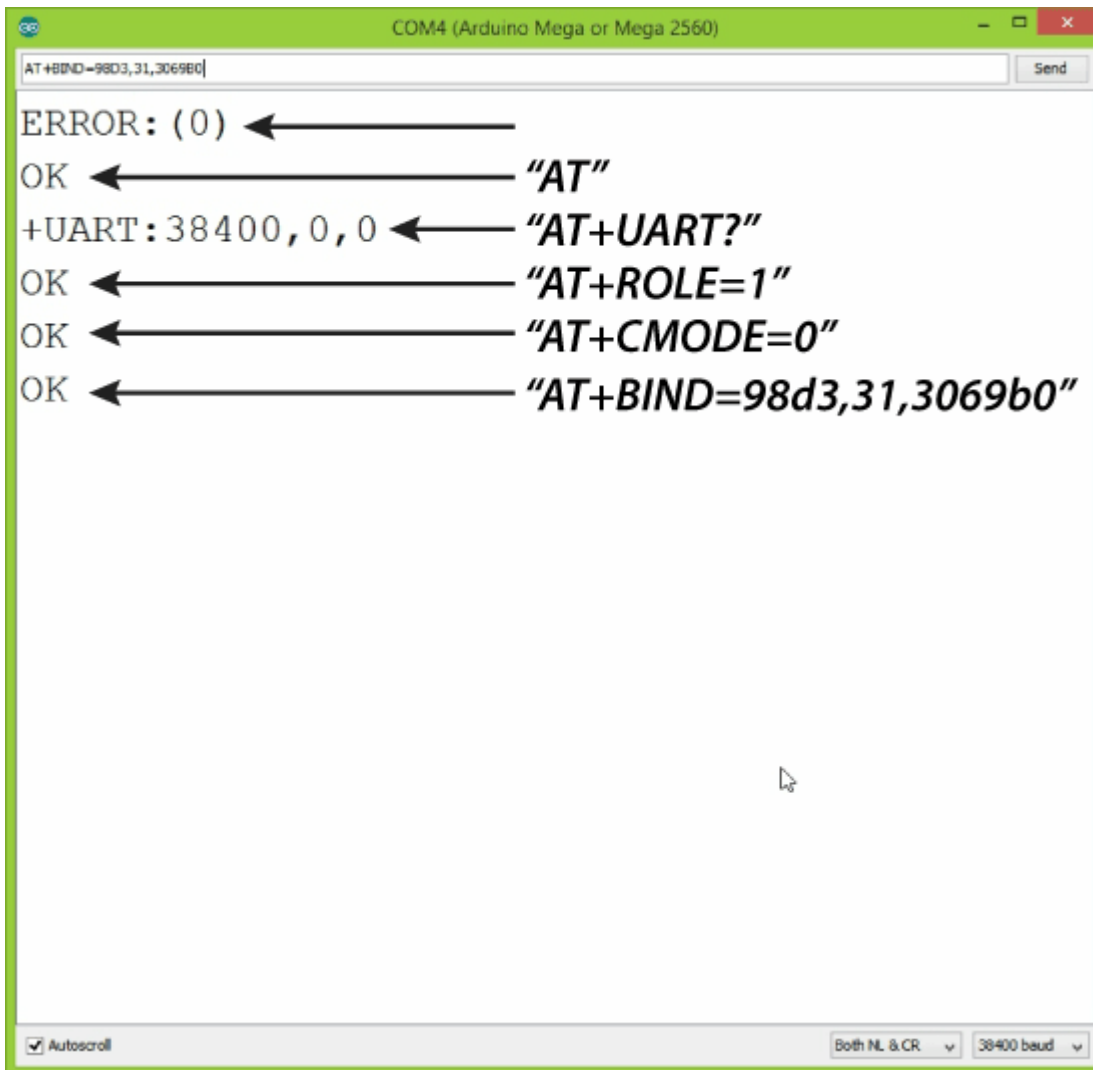
MASTER CONFIGURATION (this hc-05 module will be used on the STM32F407 Discovery Board)

The STM32F407 uses 19200 baud rate to communicate on USART1, so we need to change the hc-05 baud rate, in addition to pair the master with his slave.

To speedup the baudrate to 19200 bauds, we have to type "AT+UART=19200,0,0".

Then by typing "AT+ROLE=1" we will set the Bluetooth module as a master device. After this using the "AT+CMODE=0" we will set the connect mode to "fixed address" and using the "AT+BIND=XX,XXXX,XX" command we will set the address of the slave device that we previously wrote down. (**Use commas, instead of colons!!**)

"AT+BIND=18,e4,34cldd" in our case.



Now, the two modules are paired. Let's take a look at the code.

STM32F407 CODE (Transmitter)

The code loaded on the STM32F407 board is an adaptation of the code "2_serial" that we saw at lesson.

The code uses the USART2 channel to communicate with the HC-05 module.

So, we should link the hc-05 rx pin to PA2 pin (USART2 TX) and hc-05 tx pin to PA3 (USART2 RX)

PA2 (USART2 TX)

PA3 (USART2 RX)

In our project, we only send strings from SMT32F407 to Arduino (not vice versa) so the code will only write on the PA2 pin. (It is not necessary to link hc-05 tx to PA3 pin for our purpose)

To send a string, we just need to use the “write” function implemented in serial.cpp class.

ARDUINO CODE (Receiver)

“ArduinoBluetoothTest.txt” contains a sketch with allow us to send and receive strings trough Bluetooth communication.

We decided to use PIN 11 (TX) and PIN 12 (RX) to communicate with the hc-05 module.

So, we should link the hc-05 rx pin to PIN 11 and hc-05 tx pin to PIN 12.

In our project we only send strings from SMT32F407 to Arduino (not viceversa) so the code will only read from PIN 12. (It’s not necessary to link hc-05 rx to PIN 11 pin for our purpose)

To read a string from the hc-05 on Arduino is enough to use the read() function of the SoftwareSerial class.