

Wireless Programming of Arduino using Bluetooth

The following instructions provide steps to allow programming and reprogramming of the Arduino boards. The standard Arduino board will be used in this procedure but the method applies for the mini as well. The Bluetooth module used is WRL-00582 BlueSMiRF from Sparkfun Electronics with built-in ceramic antenna.

The procedure has 3 parts.

- 1) Programming the BT module for 19200-baud communication.
- 2) Adding a hardware reset circuit.
- 3) Adding a reset function to your program.

Changing the BT module baud rate

Because program uploading takes place at 19200 baud, it is necessary to change the baud rate of the BT module. This can be accomplished using HyperTerminal after communication is established at the default 9600, 8,N, 1 setting. (CR- Return Key).

Enter +++<CR> to get the module in 'command' mode. Module will respond OK.
Enter ATSW20,79,0,0,1<CR>. This command will change the module communication parameters to 19200 baud, No Parity and 1 stop bit.
Enter ATSI,8<CR> to confirm programming. The module should respond with

```
OK
004F,0000,0000
```

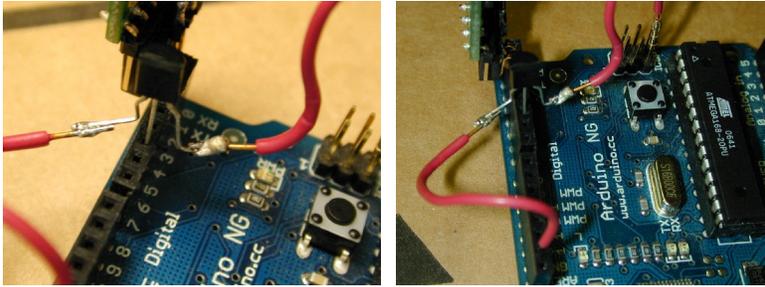
4F is 79 in hex. This confirms the module is programmed for 19200 baud communication.

Enter ATMD<CR> to get the module back into 'Data' mode. When the module responds with OK, it is ready for data communication at 19200 baud.

For a detailed instructions on programming the module refer to the AT command set available at http://www.sparkfun.com/commerce/product_info.php?products_id=582.

Adding a hardware reset

Unless your Arduino board is near your programming PC and not inside a box, in which case you can use the USB cable for programming it, you will need some means of resetting the board to get it ready to accept a new program. This can be accomplished using a general purpose NPN transistor.



In this example we will use the 2n3904 and the connection is simple. The collector goes to the reset pin, the emitter goes to ground and the base connects to any available digital output, we will use DIO5 for this exercise. Once the board is setup this way, all you need to do to get the board ready to accept a new program is turn DIO5 HIGH, that will turn the collector LOW thereby resetting the board and activating the Bootloader.

Adding a reset function to your program.

Next you need a mechanism within your regular program to detect when a new program is available for uploading. This is achieved by including a code for reading a command within your main program like...

```
serbyte = Serial.read();
if (serbyte == 'b') { // check if command is 'b'
  dloadprog();
}
```

In this case, sending the character 'b' will reset the board by running the function below where bootpin is 5.

```
void dloadprog() {
  Serial.println("Device will reboot in 18 seconds. Have new program ready.");
  delay(18000);
  digitalWrite(bootpin, HIGH);
}
```

In the setup function include the line
Serial.begin(19200);

To upload a new program

Get your new program ready in your Arduino IDE with the code given above included. Open your serial monitor select 19200-baud and send 'b'. As soon as you see the message "Device will reboot..." Press the upload button. Wait and confirm program is uploaded properly by monitoring the debug window.

NOTES:

The delay included in the `dloadprog()` is very important. I have found accessing the serial port in wireless mode (BT) takes significantly longer than when you are connected wired with a USB cable. On my PC it took anywhere from 16 ~ 20 seconds. This delay could be different in yours and may also depend on other processes that I am not aware of. You need to incorporate the correct delay in order to catch the boot window in time after the board is reset. To estimate this delay press on the serial monitor button and measure the time it takes the serial window to appear. Repeat this 3 to 5 times, average it, subtract 3seconds from the average and use that value for the delay.