Programming The RFD900 Custom Board From Borealis Labs

By Matthew Phillips

Forword:

The following instructions are very much aided by documentation made by Silicon14 on Instructables.com (https://www.instructables.com/ATMEGA2560-Standalone-Using-Arduino-UNO/); however, I found the instructions insufficient for this specific use case of the ATMega2560 and is therefore the reason for this document.

Parts:

Firstly, you will need a few parts.

- RFD900 Custom Board
- Arduino Uno Board
- USB to USB type B for Arduino Uno
- 6 Male to Female wires to connect to board pinouts
- A computer

Setting up the Arduino Uno:

To start off with, please make sure you have the Arduino IDE installed on your computer, if it is not you can find it here: https://downloads.arduino.cc/arduino-1.8.19-windows.exe

Installing IDE Library

We now need to install a library for the ATMega2560, the MegaCore library. To start off with, let's open the Arduino IDE and go to File --> Preferences and paste in:

https://mcudude.github.io/MegaCore/package_MCUdude_MegaCore_index.json

into the Additional Boards Manager URLs. Now we can install the library, which will be under tools --> board --> boards manager. From there you can search for MegaCore and install the library.

Uploading The Programmer Code

Now we're on to setting up the Arduino Uno board to act as a programmer for the ATMega2560 Microprocessor that is on the RFD900 board. To do so, plug in the Arduino Uno to your computer and "Connect just the Arduino UNO and load the File -> Examples -> ArduinoISP sketch. Select the Arduino UNO board under Tools and the default AVRISP mkII as programmer. Select the serial port (under tools-->ports-->arduino uno port) and finally compile and upload the sketch as you would normally do." (Silicon14). This just sets up the Arduino Uno so it may act as the programmer for the ATMega2560 on the RFD900 board.

From here you will want to disconnect the Arduino Uno and connect the RFD900 Board to the Uno with the following pinout:

UNO pins -> ATMEGA2560 pins

10 -> reset

11 -> MOSI

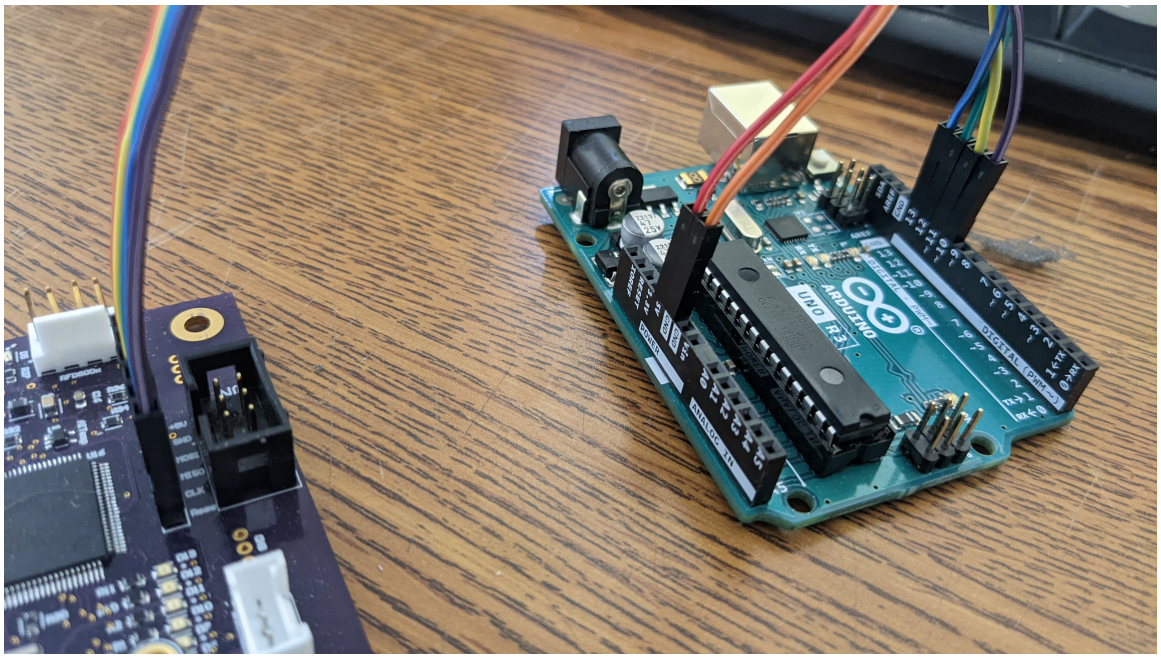
12 -> MISO

13 -> CLK

5v -> 5V

GND -> GND

(Above format credited to Silicon14)



Burning the Bootloader:

Editing Board Config File

“Once done with the connections, now find a file called "boards.txt" in your Arduino installation directory.

In my case, the path is "C:\Program Files (x86)\Arduino\hardware\arduino\avr".

Open it with a text editor [**as administrator**] (I used Notepad++) and locate to the ATMEGA2560 section. Find the line **"mega.menu.cpu.atmega2560.bootloader.high_fuses=0xD8"**

We have to replace the value of "**0xD8**" with "**0xD9**". This is so, because in stand-alone ATMEGA2560 chips, if the BOOTRST fuse is not set, the chip will be correctly programmed, but the programs will never run. So the line could look like this (we can keep the original line commented as I did.)

"mega.menu.cpu.atmega2560.bootloader.high_fuses=0xD9".

Save the changes and close the text editor.” (Silicon14)

Configuring

Now we can get all of our options out of the way. In boards we will want to go into the newly added MegaCore boards and select ATmega2560. Then we'll set the following under tools:

Clock --> "External 16 MHz"

BOD --> "BOD 2.7V"

EEPROM --> "EEPROM retained"

Compiler LTO --> "LTO disabled"

Pinout -->Arduino MEGA pinout

Bootloader --> "UART0"

Port --> whatever port the Arduino Uno is on (At this point you can plug the Arduino Uno back in)

Programmer --> "Arduino as ISP"

Burning

Now we can go up to tools --> Burn Bootloader

Programming:

Installing Libraries

Assuming that you are using the default software to collect data with the board, you will need to install a few libraries for the default code to work. All these libraries can be searched for, and installed, by going to sketch--> include libraries --> manage libraries. Those libraries are:

SparkFun u-blox Arduino Library

MS5xxx

Adafruit_LSM303_Accel

Adafruit_LIS2MDL

Editing config files

For the RFD900 board to communicate with the SD card and properly write to it, we also must edit a config file due to a mismatch in read/write speeds by default. So please open notepad as an administrator again and go to the directory:

C:\Users\<Username>\AppData\Local\Arduino15\packages\MegaCore\hardware\avr\2.1.1\libraries\SD\src

From there open SD.cpp and change the following:

```
  return card.init(SPI_HALF_SPEED, csPin) &&
         volume.init(card) &&
         root.openRoot(volume);
}
```

Original code

```
  return card.init(SPI_QUARTER_SPEED, csPin) &&
         volume.init(card) &&
         root.openRoot(volume);
}
```

Edited code

Uploading

Finally, we can now upload our code. To do so, please have your code ready and do Sketch --> Upload Using Programmer. You **must** upload using programmer, otherwise it will not upload the program correctly to the board. Once uploaded there should be two LEDs that begin to blink every few seconds that are labeled on the PCB to be D12 and D13. If these are blinking, we know that it is collecting data and storing it to the SD card. We can also check for a file on the SD card named PAYLOAD