

Bootloading 2560. Part I

What is needed:

1. Arduino UNO with a socketed ATMEGA328P 28-pin chip
2. Download and install the Arduino IDE: <https://www.arduino.cc/>
3. 6 jumper wires: <https://www.pololu.com/product/1700>

Programming the Arduino bootloader.

1. Start the Arduino IDE program.
2. Connect just the Arduino UNO and load the File -> Examples -> ArduinoISP sketch.
3. Select the Arduino UNO board under Tools and the default AVRISP mkII as programmer.
4. Select the Serial port the Arduino UNO is connected to and finally compile and upload the sketch.

After uploading is done, Disconnect the Arduino and proceed to make the connections as below:

| | | |
|----------|----|--------------------|
| UNO pins | -> | SD DDP ICSP header |
| 10 | -> | RESET |
| 11 | -> | MOSI |
| 12 | -> | MISO |
| 13 | -> | SCK |
| 5v | -> | VCC |
| GND | -> | GND |

Bootloading 2560. Part II

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 (I used Notepad++) and locate to the ATMEGA2560 section. Find the following line:

```
"mega.menu.cpu.atmega2560.bootloader.high_fuses=0xD8"
```

We must replace the value of "0xD8" with "0xD9". This is required 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.

Edit the line to the following. (You can keep the original line commented out by inserting a # as the first character.)

```
"#mega.menu.cpu.atmega2560.bootloader.high_fuses=0xD8"  
"mega.menu.cpu.atmega2560.bootloader.high_fuses=0xD9".
```

Save the changes and close the text editor.

Now we burn the bootloader. Connect the Arduino UNO board & open Arduino IDE, go to Tools, and select as following:

Board: Arduino Mega 2560.

Processor: ATMEGA2560.

Programmer: Arduino as ISP.

Port: (respective Arduino UNO port).

And now from Tools, click "Burn Bootloader". If everything is done correctly, the process will take some seconds and will finish with no errors.

SD DDP Programming Instructions

What is Needed:

Micro-USB cable

Pololu USB AVR programmer v2.1 - <https://www.pololu.com/product/3172>

Pololu USB AVR programmer software - <https://www.pololu.com/product/3172/resources>

ADAM_DDP_SD_Drive_Int_Power.ino.hex program file - <https://github.com/epearsoe/Coleco-ADAM-SD-DDP>

Xloader (Arduino .hex loader for Windows) - <https://www.hobbytronics.co.uk/arduino-xloader>

Procedure

NOTE: Disconnect the SD DDP from your ADAM computer by removing the 2 plugs before proceeding.

1. Download and install the Pololu AVR USB programmer software.
2. Download the ADAM_DDP_SD_Drive_Int_Power.ino.hex program file.
3. Download Xloader .hex loader software.
4. Attach the USB cable from the computer to the USB connector of the Pololu programming device. See Figure 1.
5. Launch the Pololu USB programming software program and configure the programmer settings as follows.

| | |
|----------------------|----------|
| ISP Frequency | 2400 kHz |
| Max ISP Frequency | 2400 kHz |
| Regulator mode | 5V |
| VCC output | Enabled |
| VCC output indicator | Steady |

Note: All other values stay default. Remember the 'Programming port' value for later.

6. Verify programmer LEDs are illuminated as per Figure 1 to validate that 5v power is being applied by the programmer. You can now safely close the Pololu USB programming software configuration program.

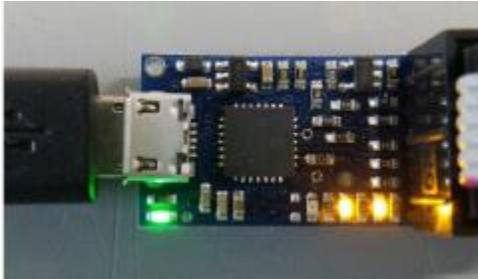


Figure 1 Pololu Programmer LEDs

7. Attach the 6-pin ribbon cable from the programmer to the ICSP connector on the SD DDP PCB. The red strip on the ribbon designates Pin-1 of the connector. See Figure 1 and 2 for details.

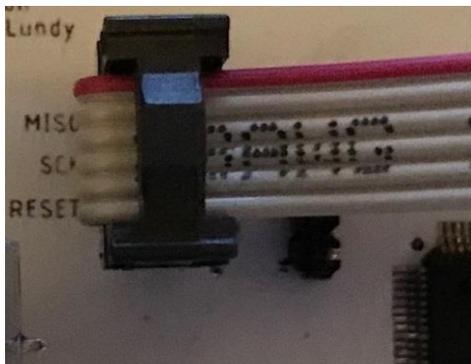


Figure 2 SD DDP ICSP Connection

8. Launch the Xloader updater program and configure Xloader according to Figure 3.

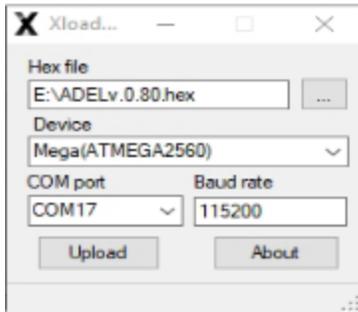


Figure 3 Xloader settings

9. Select the appropriate COM port for the device in the drop-down list that was generated for the 'Programming port' earlier in step 6.

10. Select the ADAM_DDP_SD_Drive_Int_Power.ino.hex file for the device.

11. Click Upload to update the device. The process will take approximately five seconds to perform. The LED on the SD DDP PCB should illuminate.

12. It is now safe to disconnect the 6-pin ribbon connector from the SD DDP PCB.

13. You can now re-connect the SD DDP to your ADAM or continue to the next step in the "Coleco ADAM SD DDP Build Guide and Info Pack.pdf" if building a new SD DDP.