Introduction to AVR Microprocessor Development Board

Daniel Murphy and Annie Guo

This document gives you a brief description on the AVR Microcontroller Development Board and how to use it.

1. Equipment Handling Precautions

A few of precautions are necessary in the experiment:

- Your experiment place, either the laboratory or home, can be an environment where it is quite easy for you to build up a static electric charge. **Electrostatic discharge can destroy electronic equipment**, including the circuits on the lab board, without giving any sign of doing so!

  Since you may be carrying a static charge without even realizing it, **you should always discharge yourself**. You can do this painlessly by touching a grounded conductor using a coin, a key or other metallic objects instead of your finger.

- Short circuits may damage certain devices. Please remove any metal items before handling the board. There are no dangerous voltages or currents that will harm you here, but this is a good habit to always act with caution and follow the designed procedure.

- Always turn the power off before connecting or disconnecting any I/O subsystems and don't place the board on a conductive surface when the power is on.

2. AVR Microcontroller Board

Figure 1 shows a picture of the assembled lab board that contains peripherals, connectors, and the Arduino MEGA 2560 R3 microcontroller board (at the back).

To program the board, the following software (available on the course website under the link References) should be first downloaded and installed

- avrdude.conf
- avrdude.exe
- arduino-1.0.6-windows.exe

To connect the board to your computer, use the USB Type-B port on the Arduino (left). A green light should be visible on the Arduino board. When using the lab computers, use the provided UAB cable instead of pulling you own one into the computer.

The communication between the board and the computer requires a correct COM port setting. To work out which COM port the Arduino is connect to, open the system properties of the computer by clicking the start button, right-clicking 'My Computer' then clicking properties. In the ‘Hardware’ tag, click ‘Device Manager’. Scroll down to ‘Ports (COM & LPT)’. There should be one item called ‘Arduino Mega 2560 (<portname>)’.

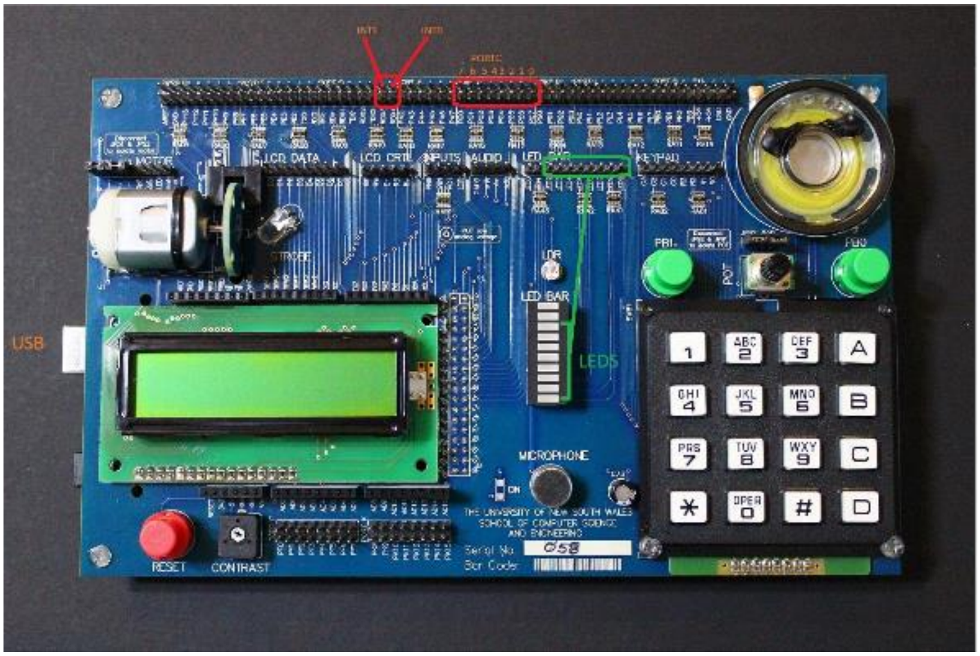
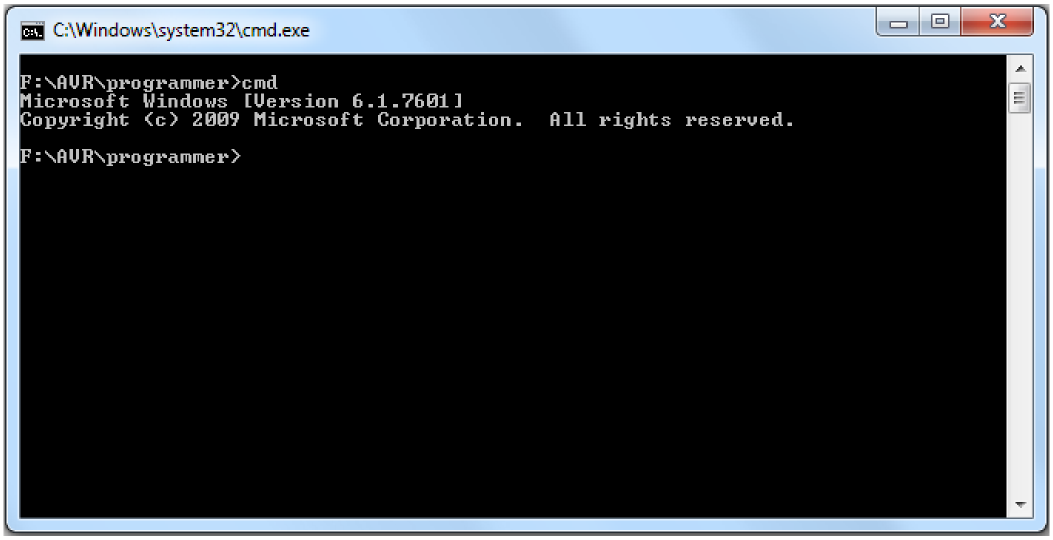


Figure 1: AVR Development Board

A demonstration is given below on how to run a code (test.hex) on the board.

Download the file ‘programmer.zip’ from the Labs page on the course website. Extract the files and open the folder.

Run the batch file ‘console.bat’ to open up a command window as shown in Figure 2.



```
F:\AVR\programmer>cmd
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

F:\AVR\programmer>
```

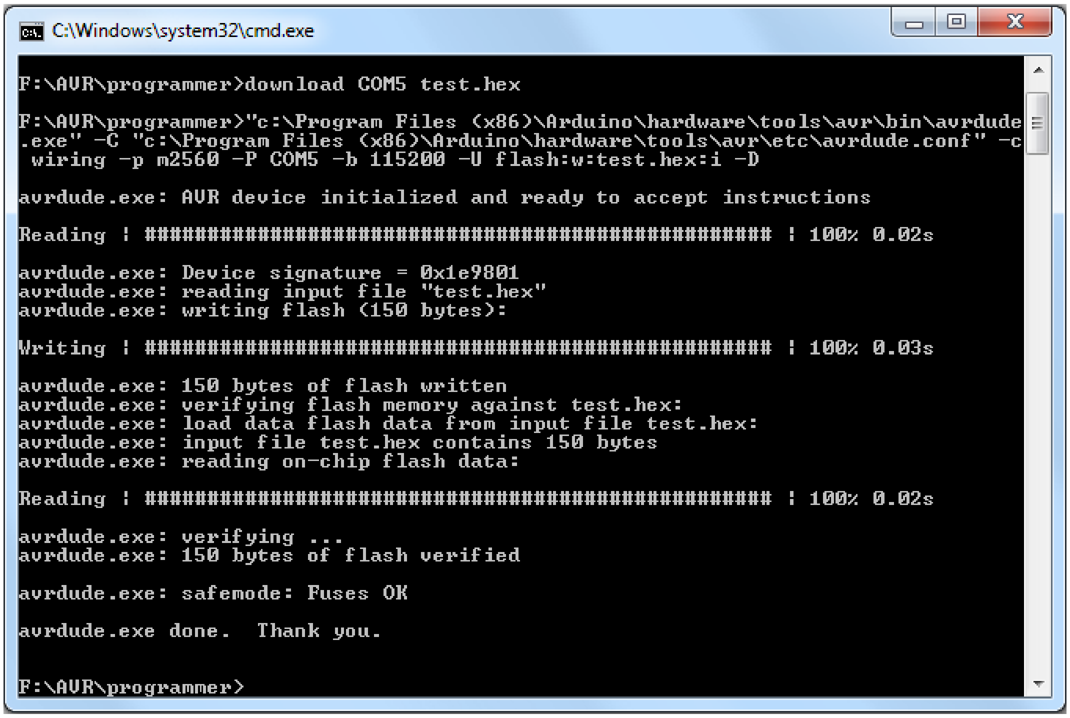
Figure 2

In the window, type the following command to load the test program in the folder with the following download command:

```
download <comport> test.hex
```

Here <comport> is the name of the port you found earlier. E.g. COM5.

A message will be returned to indicate the success of the downloading, as shown in Figure 3.



```
F:\AVR\programmer>download COM5 test.hex

F:\AVR\programmer>"c:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude
.exe" -C "c:\Program Files (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf" -c
 wiring -p m2560 -P COM5 -b 115200 -U flash:w:test.hex:i -D

avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude.exe: Device signature = 0x1e9801
avrdude.exe: reading input file "test.hex"
avrdude.exe: writing flash (150 bytes):

Writing | ################################################## | 100% 0.03s

avrdude.exe: 150 bytes of flash written
avrdude.exe: verifying flash memory against test.hex:
avrdude.exe: load data flash data from input file test.hex:
avrdude.exe: input file test.hex contains 150 bytes
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 0.02s

avrdude.exe: verifying ...
avrdude.exe: 150 bytes of flash verified

avrdude.exe: safemode: Fuses OK

avrdude.exe done.  Thank you.


F:\AVR\programmer>
```

Figure 3

Once the download is done and the LEDs are connected correctly, the LEDs should start flashing.

You can run your own code generated by the AVR Studio. To download it, replace ‘test.hex’ with the path to your code.

For example,

```
download <comport> F:\AVR\test2.hex
```