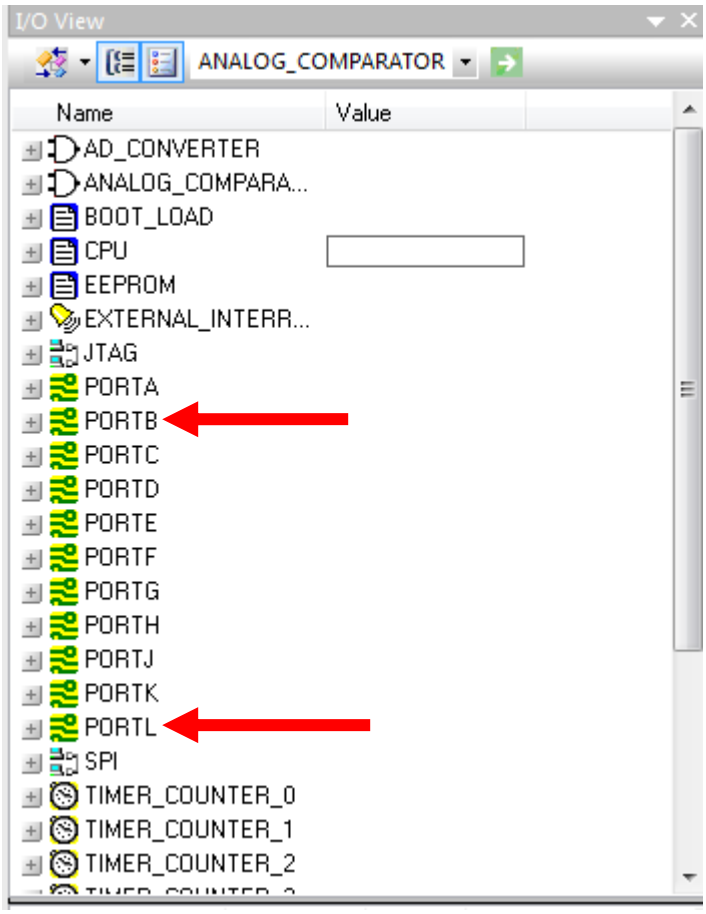
Lab 3 Introduction to MEGA 2560 Board and I/O Instructions

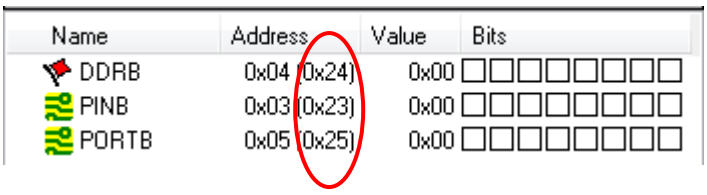
Submit lab3BlinkOne.asm at the end of your lab, not your project.

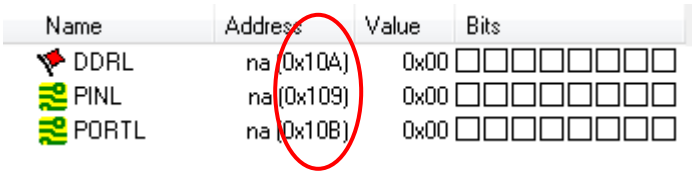
I. I/O Instructions

In the AVR MEGA 2560 microcontroller, each Input/Output (I/O) port has three associated registers, the data direction register (DDRx), output register (PORTx), and input register (PINx). These registers correspond to addresses in the data space accessible by the processor (the first 0x200 bytes in SRAM).

Launch AVR Studio 4 and observe the I/O View. Expand PORTB and PORTL



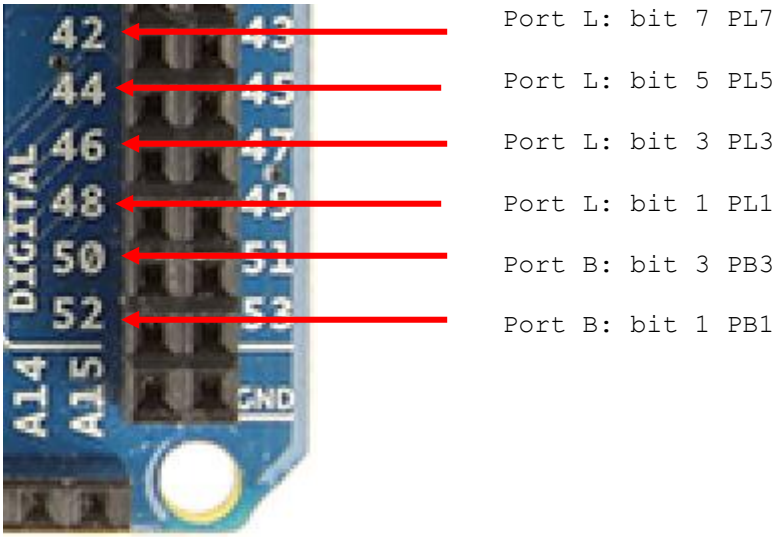




Some ports are mapped to two addresses, for example, PORTB is mapped to 0x05 and 0x25, you may use IN/OUT instructions to transfer data between registers and SRAM (data memory) for address between 0x00 and 0x3F, but some ports such as PORTL are mapped to one addresses which are bigger than 0x3F, they can't be manipulated using the IN/OUT instructions, what are we going to do? We use LDS and STS instructions instead. In AVR, Ports A to G use Port Mapped I/O (separate addresses from memory) and Ports H to K use Memory Mapped I/O (usage is similar to any memory location). Port Mapped addresses have to use separate In/Out instructions while Memory mapped use LDS and STS.

II. Pins and Ports

The six LEDs are associated with six pins and the pins are mapped to some bits of two ports: PORTB and PORTL:

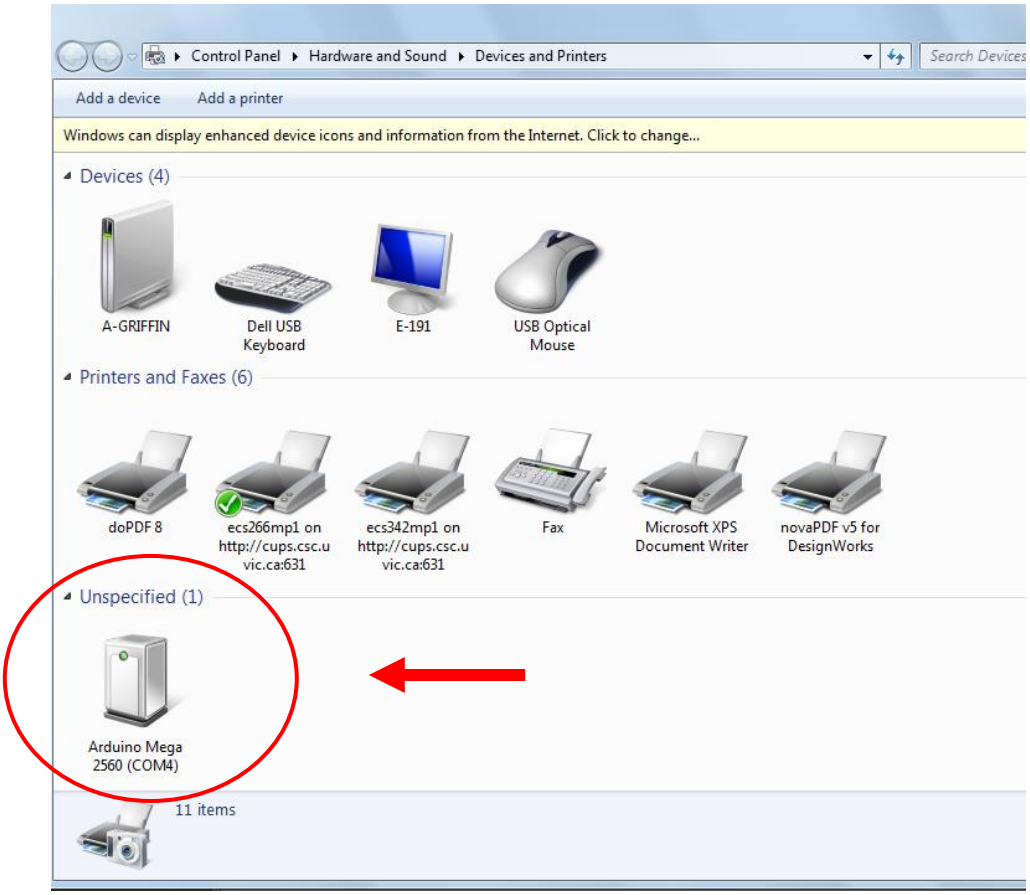


Create a new project named lab3. Write a small program to turn a specific LED on. Type the following code. Note that the I/O registers - 0x10B and 0x10A - are specified in hexadecimal numbers. The two I/O registers are given names, PORTL and DDRL, by using the .equ directives to the port numbers (the memory addresses in data memory). Once equated, the I/O registers are referred to by names instead of numbers in the program.

```
;pins.asm
;learn which port is mapped to which pin and how to turn it on/off
.equ PORTL = 0x10B
.equ DDRL  = 0x10A

    ;configure PORTL as output by setting DDRL to 1
    ldi r16, 0xff
    sts DDRL, r16

    ;set LED at pin 48 on
    ldi r19, 0x02
    sts PORTL, r19

done:jmp done
```

III. Build and download the .hex file to the board:

- Build the program above. Observe the changes of the registers and the I/O registers.
- **Note to which COM port the lab AVR board is connected to. This is different on each machine. You can check this by going to Start -> Devices and Printers.**



- In the above example, the Arduino Mega is on COM4. So please check the port number of Arduino Mega on the machine you are currently working on.

- Upload the lab3.hex file to the board by typing the following command at the command window:

```
"C:\WinAVR\bin\avrdude" -C "C:\WinAVR\bin\avrdude.conf" -p atmega2560 -c
wiring -P COM4 -b 115200 -D -F -U flash:w:filename.hex
```

You may need to change the path and filename if the settings are different from this. Or, download "upload.bat" file to the same directory where your lab3.asm program is stored. Modify the com port (-P COM4) to reflect the one on your machine. Open a command window get to the same directory where upload.bat is stored, and type upload.bat. "upload.bat" is a batch file. It is another way to type a command. You need to change the file name if your project is not called lab3. Learn some DOS command, such as "cd" – change directory.

IV. Exercises:

1. Change the number transferred to r19, rebuild the program and upload it to the board, observe which LED is on. Change the number if you want to turn on the LEDs mapped to pins 46, 44, 42. Change the code if you want to turn on the LEDs mapped to pins 50 and 52.

2. Download blinkOne.asm and change some of the code such that only one LED blinks.

Submit lab3BlinkOne.asm at the end of your lab.

This lab is derived from the Chapters 2 and 3 of your textbook (Some Assembly Required by Timothy S. Margush)