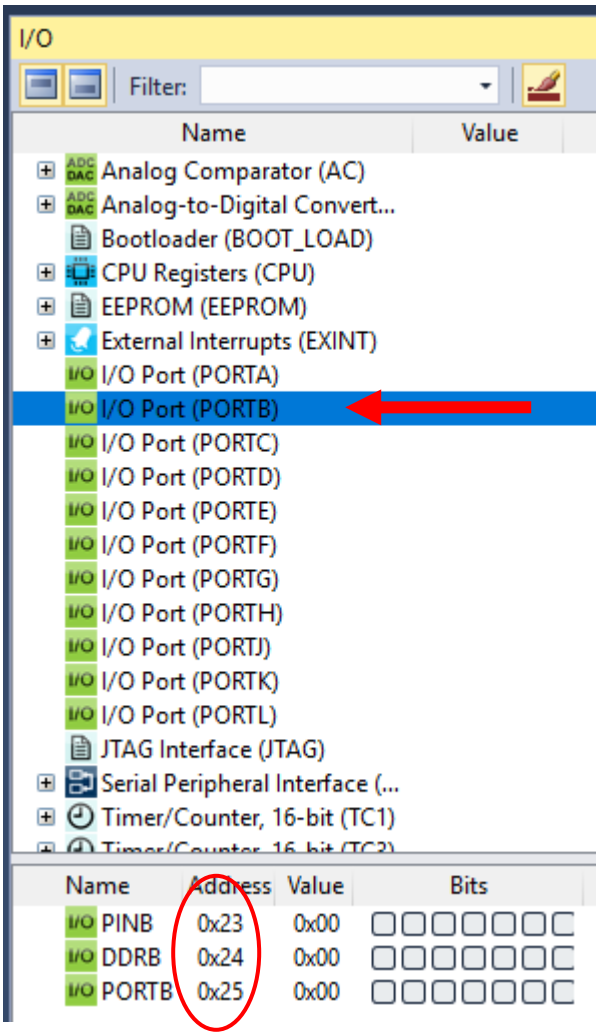
Lab 3 Introduction to MEGA 2560 Board and I/O Instructions

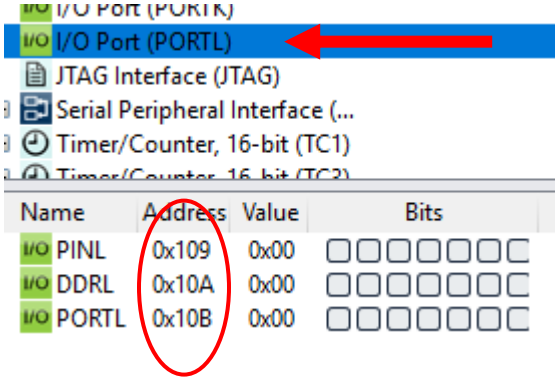
I. I/O Instructions

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 Atmel Studio 7.0 and observe the I/O View. To see the memory address of PORTB and PORTL, click on PORTB or PORTL:



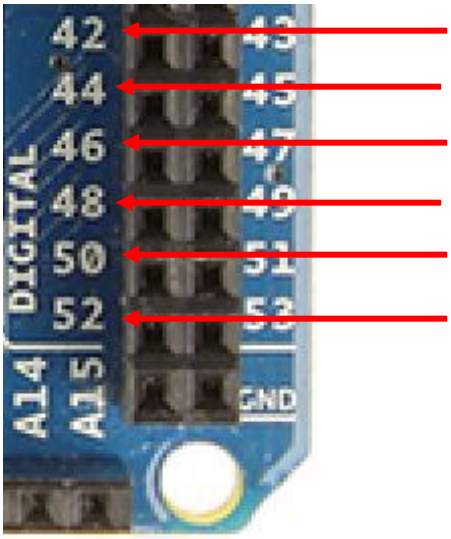
The memory address of PORTL:



Some ports are mapped to addresses smaller than or equal to 0x3F, for example, PORTB is mapped to 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 addresses which are bigger than 0x3F, they can't be accessed 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:




```
Port L: bit 7 PL7
Port L: bit 5 PL5
Port L: bit 3 PL3
Port L: bit 1 PL1
Port B: bit 3 PB3
Port B: bit 1 PB1
```

Create a new project named lab3. Write a small program to turn a few specific LED lights on. Type the following code:

```
.cseg
    ldi r16, 0xFF
    sts DDRL, r16 ;PORTL and PORTB as output
    out DDRB, r16

    ;set the top and the bottom lights on
    ldi r16, 0b10000000
    sts PORTL, r16
    ldi r16, 0b00000010
    out PORTB, r16

done:jmp done
```

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

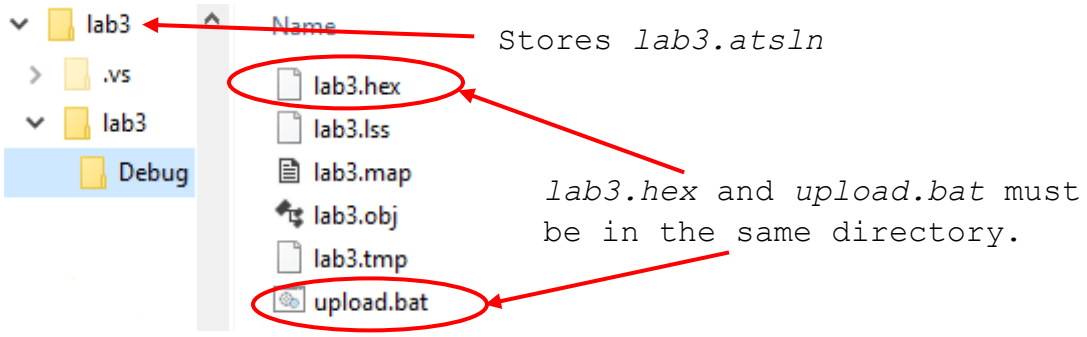
- Build and run the program above. Observe the changes of the registers and the I/O registers.
- **Note to which COM port the lab AVR board is connected to. This is different on each machine. You can check this by launch the Arduino IDE (refer to part III of the lab 1 note, page 6 – to be specific.)**
- In the example above, the Arduino Mega is connected to the computer using COM4. So 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:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avrdude.exe" -C
"C:\Program Files (x86)\Arduino\hardware\tools\avr\etc\avrdude.conf" -p
atmega2560 -c wiring -P COM4 -b 115200 -D -F -U flash:w:lab3.hex
```

You may need to change the path (e.g. `C:\Program Files (x86` and filename (e.g. lab3.hex) if the settings are different from this. Instead of typing such a long command, you can download "upload.bat" file to the same directory where your lab3.hex file 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 *lab3.hex* if your project is not called lab3. Learn some DOS commands, such as "cd" – change directory.

To find the *lab3.hex* file, refer to the following screen shot:



IV. Exercises:

1. Change the number loaded to r16, 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. Make the light blink, one at a time. At least two lights.

2. Download blink.asm and change some of the code such that at least two LED lights blink alternatively.