

Program: Mechanical Engineering/Industrial Engineering

Course Number	MEC 733
Course Title	Microprocessor
Semester/Year	2022 W
Instructor	Dr. Siyuan He

Lab No.	1
---------	---

Lab Title	Familiarizing with the Elegoo Board and Arduino Assembly Programming
-----------	--

Section No.	04
Submission Date	02/18/2022
Due Date	02/18/2022

Name	Student ID
Cheng-Hao, Kao	xxxx23949

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*
<http://www.ryerson.ca/senate/policies/pol60.pdf>.

Abstract

This experiment examines how components on an Arduino board can be controlled through assembly language and the usage of *Arduino IDE*. Several jumper wires were used to connect the resistors, controller board, breadboard, and LEDs together. The whole device was later connected to a computer so that three previously created files (including a .ino file, .s file, and a .h file) can be uploaded to control the UNO R3 Controller Board. As a result, the designated LEDs were able to flash in the correct order as required.

Introduction

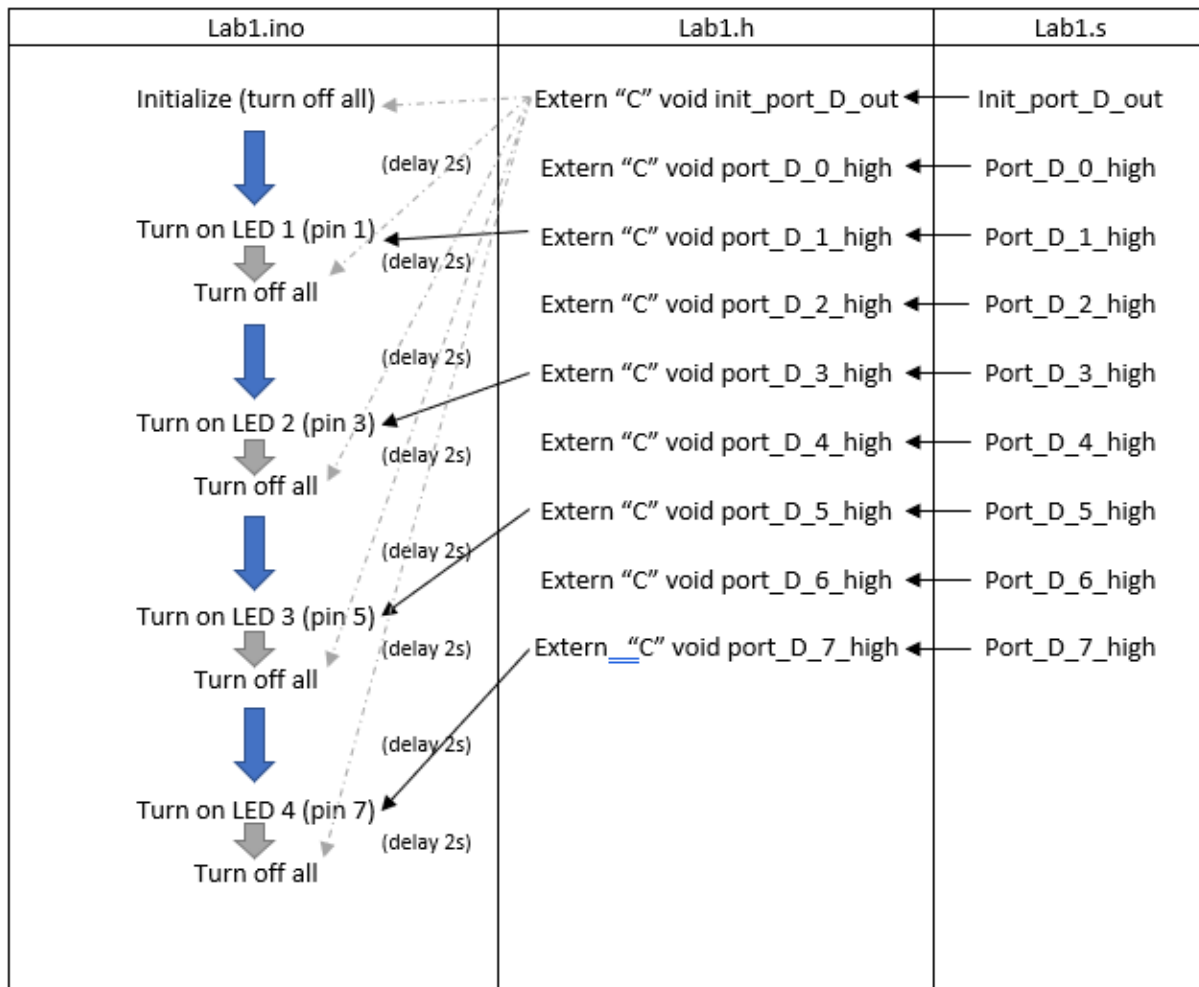
The purpose of this experiment is to explore simple assembly language instructions as *cbi* and *sbi*, as well as how DDRD (0X0A) and PORTD (0X0B) works. In specific, *cbi* and *sbi* will be used to control each bit of the DDRD as a means to determine whether each pin is an input or an output, whereas the bits on PORTD dictate whether each pin provides a high or low voltage when being used as an output [1].

This would require functions to be defined and is all to be done in a .s file. Such functions will then later be defined again as assembly functions in a .h file, allowing them to be called in by the commands in the .ino file [1]. A similar was done and presented to the class in *Week2_Example2.ino*, *Week2_Example2.s*, and *Week2_Example2.h* provided by the professor. A generic approach can be followed, but particular details of the codes would have to be adjusted and modified in order to achieve the wanted results from the Atmel Atmega 328 microcontroller and the Arduino board it was attached to.

Experimental Equipment

- UNO R3 Controller Board *1
 - USB Cable *1
 - Breadboard *1
 - LEDs (blue, red, green, yellow) *4
 - Resistors (220 Ohm) *4
 - Jumper Wires *5
-

Program Description



As shown by the flowchart, functions were defined in the .s file through instructions *sbi* and *cbi* which set the target values to 1 and 0, respectively. For instance, the function *init_port_D_out* was defined by setting all pin values on the DDRD to 1 and pin values on PORTD to 0. Similarly, *port_D_0_high* was accomplished by making the value corresponding to pin 0 on PORTD 1. Such functions were then defined as assembly functions in the .h file, such that they can be called in the .ino file.

These functions, when called in the .ino file, turn into actual commands that can be communicated between the Arduino program and the microprocessor. As an example, the functions *init_port_D_out* and *port_D_X_high* (X being an arbitrary number between 0 and 7) were repeatedly used to transmit signals to the microprocessor as a means to turn the LEDs on and off. Along with the *delay_ms(2000)* command, this allowed the LEDs to be controlled as desired, being turned on and off for a designated period of time (in this case 2 seconds).

Conclusion

Overall the experiment went successfully, with the blue, red, green, and yellow LEDs flashing in the expected sequence. Below is an image of the actual experimental setup within the lab.

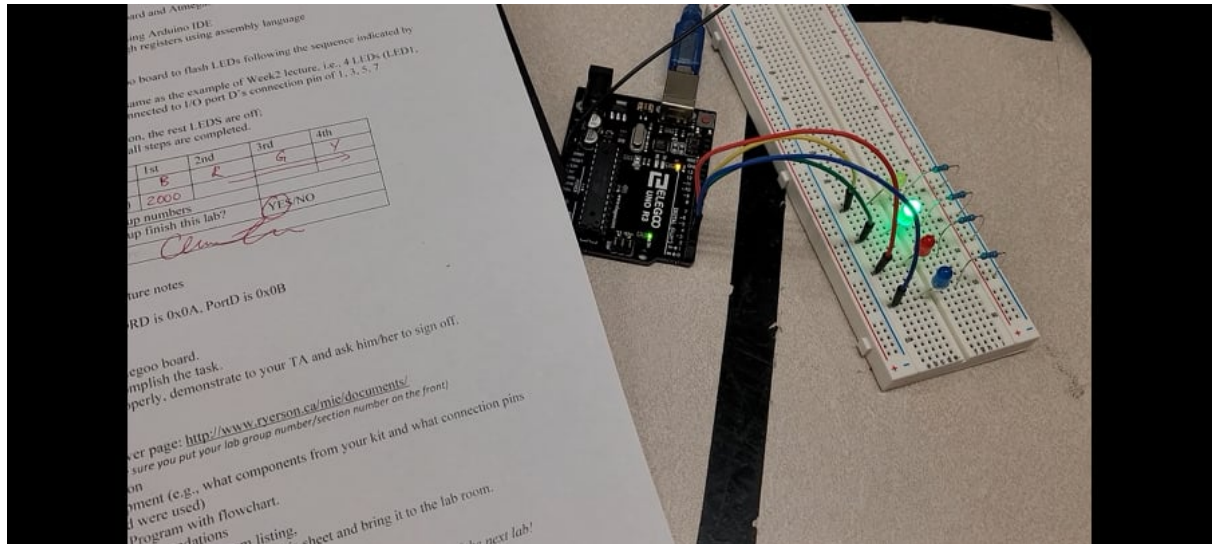


Figure 1 - Experimental Setup of the UNO R# Controller Board

Noticeably, the blue (1st) LED flashed twice before the red (2nd) LED started flashing. This was an unexpected result as the initialization process was only meant to make all pins to output mode while setting the voltage output to be low at the same time. The codes were checked and no errors were found. It was later confirmed by the TA that such an outcome was normal and was due to random signals assigned to the pins that took place prior to the initialization process.

Recommendation

Considering this was the first lab and the procedures were relatively straightforward, no substantial recommendations were required or made. However, it is expected that as the lab proceeds, experiments would soon be more complex and that recommendations would be needed to improve the quality as well the speed of experiment execution.

Appendix

- Lab1.ino

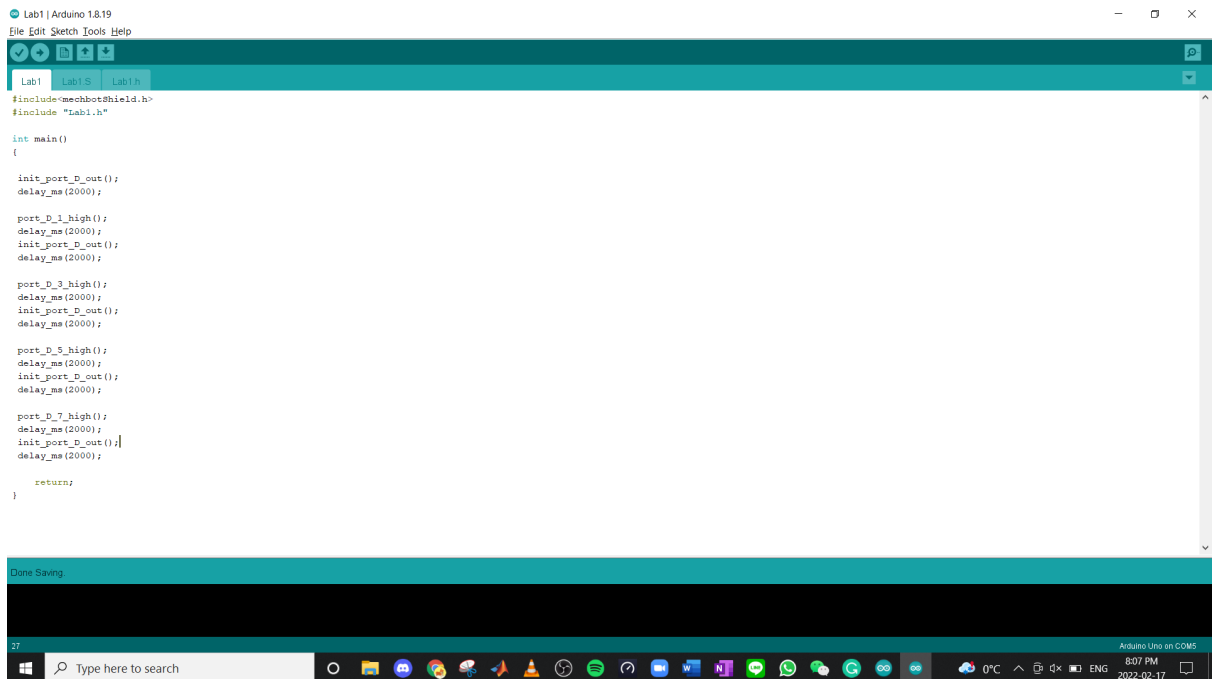


Figure 2 - Screenshot of Lab1.ino File

- `#include<mechbotShield.h>`
`#include "Lab1.h"`

```
int main()
{
```

```
    init_port_D_out();
    delay_ms(2000);
```

```
    port_D_1_high();
    delay_ms(2000);
    init_port_D_out();
    delay_ms(2000);
```

```
    port_D_3_high();
    delay_ms(2000);
    init_port_D_out();
    delay_ms(2000);
```

```
    port_D_5_high();
    delay_ms(2000);
    init_port_D_out();
    delay_ms(2000);
```

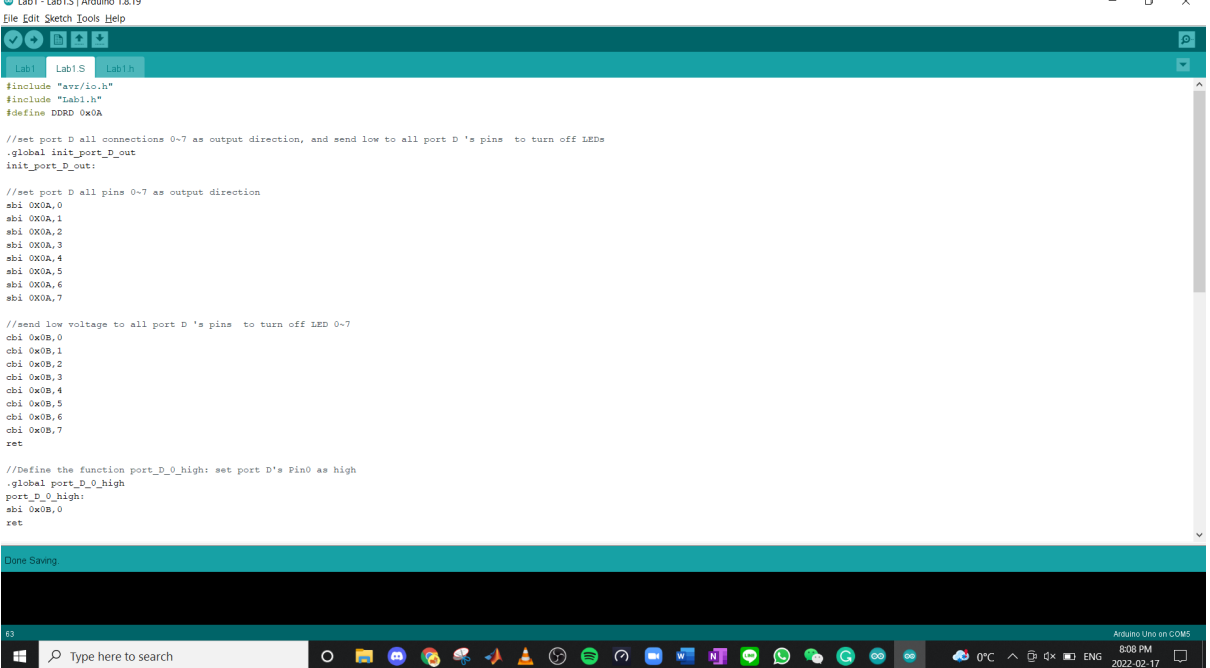
```
    port_D_7_high();
    delay_ms(2000);
    init_port_D_out();
```



```
delay_ms(2000);
```

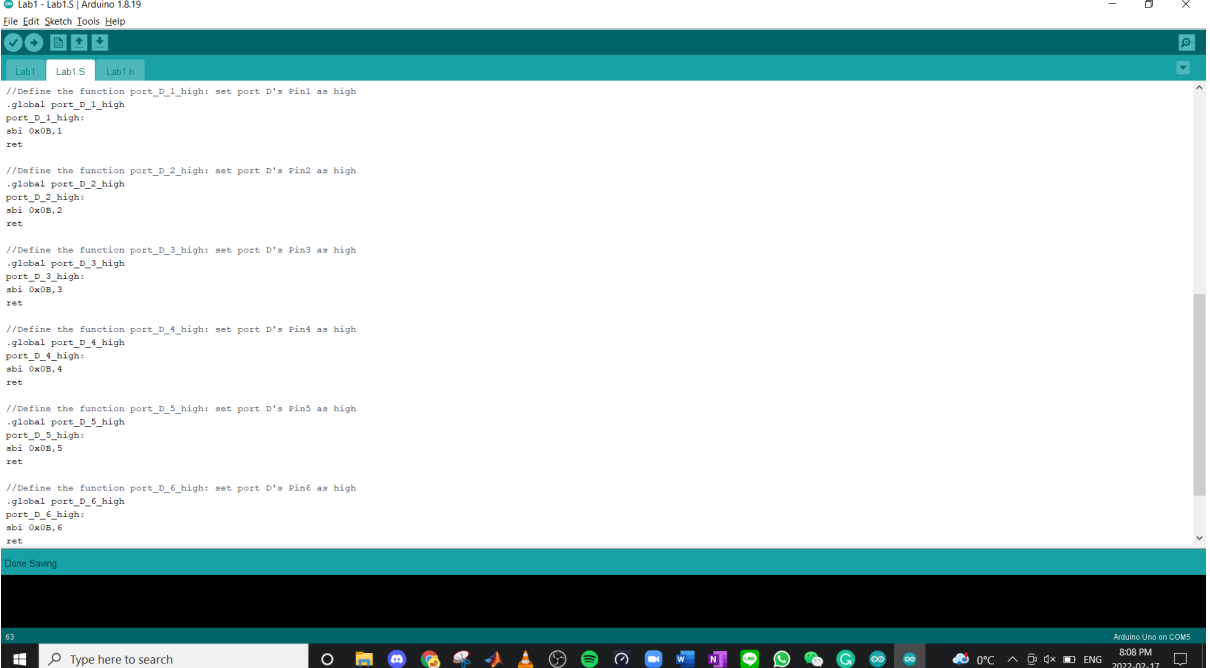
```
return;  
}
```

- Lab1.S



```
Lab1 - Lab1.S | Arduino 1.8.19  
File Edit Sketch Tools Help  
Lab1 Lab1.S Lab1.h  
#include "avr/io.h"  
#include "Lab1.h"  
#define DDRD 0x0A  
  
//set port D all connections 0-7 as output direction, and send low to all port D 's pins to turn off LEDs  
.global init_port_D_out  
init_port_D_out:  
  
//set port D all pins 0-7 as output direction  
sbi 0x0A,0  
sbi 0x0A,1  
sbi 0x0A,2  
sbi 0x0A,3  
sbi 0x0A,4  
sbi 0x0A,5  
sbi 0x0A,6  
sbi 0x0A,7  
  
//send low voltage to all port D 's pins to turn off LED 0-7  
cbi 0x0B,0  
cbi 0x0B,1  
cbi 0x0B,2  
cbi 0x0B,3  
cbi 0x0B,4  
cbi 0x0B,5  
cbi 0x0B,6  
cbi 0x0B,7  
ret  
  
//Define the function port_D_0_high: set port D's Pin0 as high  
.global port_D_0_high  
port_D_0_high:  
sbi 0x0B,0  
ret
```

Figure 3 - Screenshot of Lab1.s File (1st of 3)



```
Lab1 - Lab1.S | Arduino 1.8.19  
File Edit Sketch Tools Help  
Lab1 Lab1.S Lab1.h  
  
//Define the function port_D_1_high: set port D's Pin1 as high  
.global port_D_1_high  
port_D_1_high:  
sbi 0x0B,1  
ret  
  
//Define the function port_D_2_high: set port D's Pin2 as high  
.global port_D_2_high  
port_D_2_high:  
sbi 0x0B,2  
ret  
  
//Define the function port_D_3_high: set port D's Pin3 as high  
.global port_D_3_high  
port_D_3_high:  
sbi 0x0B,3  
ret  
  
//Define the function port_D_4_high: set port D's Pin4 as high  
.global port_D_4_high  
port_D_4_high:  
sbi 0x0B,4  
ret  
  
//Define the function port_D_5_high: set port D's Pin5 as high  
.global port_D_5_high  
port_D_5_high:  
sbi 0x0B,5  
ret  
  
//Define the function port_D_6_high: set port D's Pin6 as high  
.global port_D_6_high  
port_D_6_high:  
sbi 0x0B,6  
ret
```

Figure 4 - Screenshot of Lab1.s File (2nd of 3)

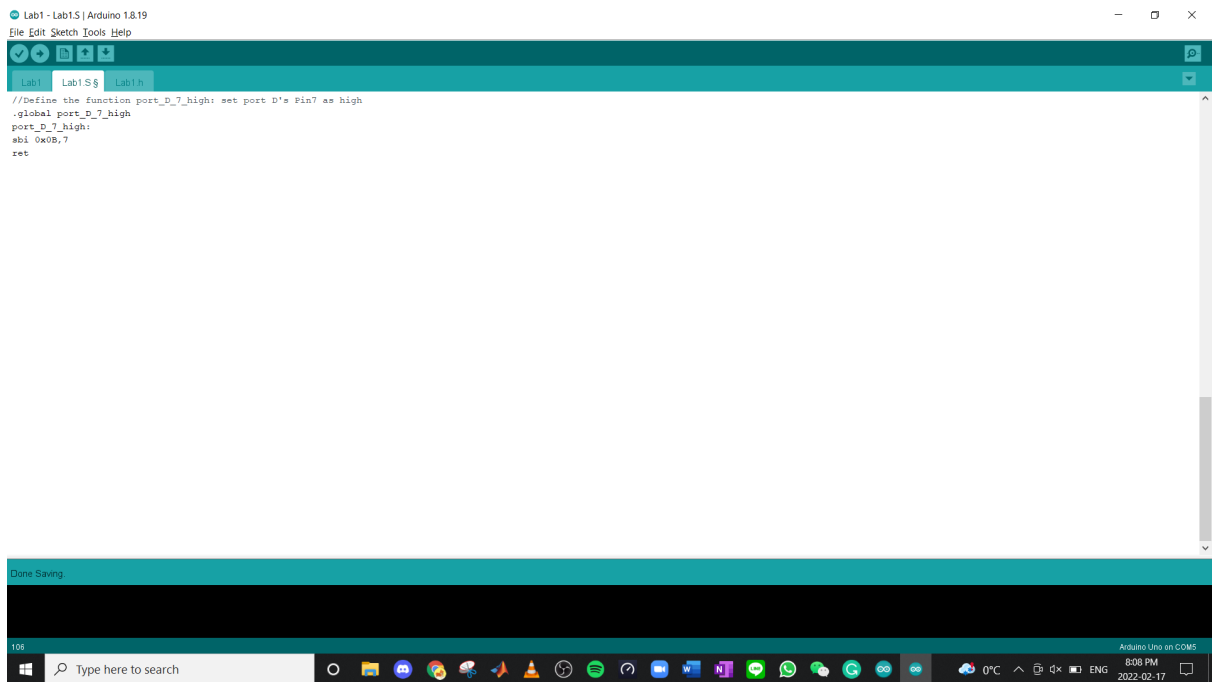


Figure 5 - Screenshot of Lab1.s File (3rd of 3)

```
#include "avr/io.h"
#include "Lab1.h"
#define DDRD 0x0A

//set port D all connections 0~7 as output direction, and send low to all port D
's pins  to turn off LEDs
.global init_port_D_out
init_port_D_out:

//set port D all pins 0~7 as output direction
sbi 0X0A,0
sbi 0X0A,1
sbi 0X0A,2
sbi 0X0A,3
sbi 0X0A,4
sbi 0X0A,5
sbi 0X0A,6
sbi 0X0A,7

//send low voltage to all port D 's pins  to turn off LED 0~7
cbi 0x0B,0
cbi 0x0B,1
cbi 0x0B,2
cbi 0x0B,3
cbi 0x0B,4
cbi 0x0B,5
cbi 0x0B,6
cbi 0x0B,7
ret

//Define the function port_D_0_high: set port D's Pin0 as high
```

```
.global port_D_0_high  
port_D_0_high:  
sbi 0x0B,0  
ret
```

```
//Define the function port_D_1_high: set port D's Pin1 as high  
.global port_D_1_high  
port_D_1_high:  
sbi 0x0B,1  
ret
```

```
//Define the function port_D_2_high: set port D's Pin2 as high  
.global port_D_2_high  
port_D_2_high:  
sbi 0x0B,2  
ret
```

```
//Define the function port_D_3_high: set port D's Pin3 as high  
.global port_D_3_high  
port_D_3_high:  
sbi 0x0B,3  
ret
```

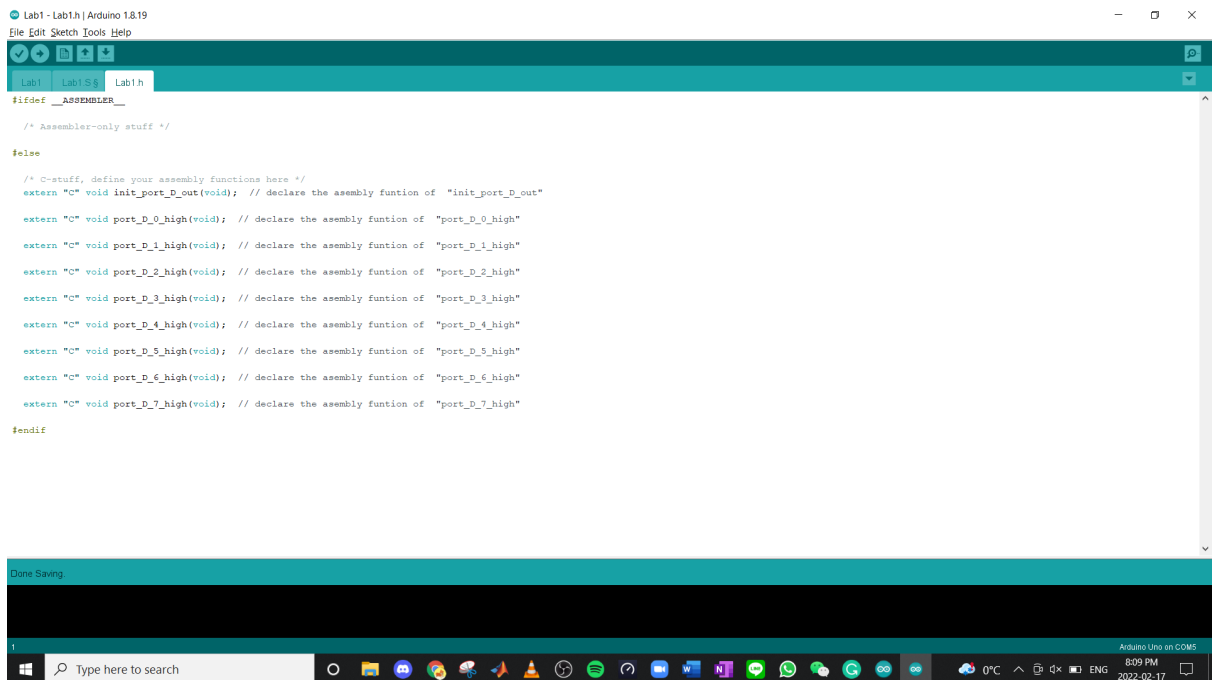
```
//Define the function port_D_4_high: set port D's Pin4 as high  
.global port_D_4_high  
port_D_4_high:  
sbi 0x0B,4  
ret
```

```
//Define the function port_D_5_high: set port D's Pin5 as high  
.global port_D_5_high  
port_D_5_high:  
sbi 0x0B,5  
ret
```

```
//Define the function port_D_6_high: set port D's Pin6 as high  
.global port_D_6_high  
port_D_6_high:  
sbi 0x0B,6  
ret
```

```
//Define the function port_D_7_high: set port D's Pin7 as high  
.global port_D_7_high  
port_D_7_high:  
sbi 0x0B,7  
ret
```

- Lab1.h



```

Lab1 - Lab1.h | Arduino 1.8.19
File Edit Sketch Tools Help

Lab1 Lab1.S Lab1.h

#ifndef __ASSEMBLER__
/* Assembler-only stuff */

#else
/* C-stuff, define your assembly functions here */
extern "C" void init_port_D_out(void); // declare the assembly funtion of "init_port_D_out"
extern "C" void port_D_0_high(void); // declare the assembly funtion of "port_D_0_high"
extern "C" void port_D_1_high(void); // declare the assembly funtion of "port_D_1_high"
extern "C" void port_D_2_high(void); // declare the assembly funtion of "port_D_2_high"
extern "C" void port_D_3_high(void); // declare the assembly funtion of "port_D_3_high"
extern "C" void port_D_4_high(void); // declare the assembly funtion of "port_D_4_high"
extern "C" void port_D_5_high(void); // declare the assembly funtion of "port_D_5_high"
extern "C" void port_D_6_high(void); // declare the assembly funtion of "port_D_6_high"
extern "C" void port_D_7_high(void); // declare the assembly funtion of "port_D_7_high"

#endif
  
```

Figure 6 - Screenshot of Lab1.h File

```

#ifndef __ASSEMBLER__

/* Assembler-only stuff */

#else

/* C-stuff, define your assembly functions here */
extern "C" void init_port_D_out(void); // declare the assembly funtion of
"init_port_D_out"

extern "C" void port_D_0_high(void); // declare the aseembly funtion of
"port_D_0_high"

extern "C" void port_D_1_high(void); // declare the aseembly funtion of
"port_D_1_high"

extern "C" void port_D_2_high(void); // declare the aseembly funtion of
"port_D_2_high"

extern "C" void port_D_3_high(void); // declare the aseembly funtion of
"port_D_3_high"

extern "C" void port_D_4_high(void); // declare the aseembly funtion of
"port_D_4_high"

extern "C" void port_D_5_high(void); // declare the aseembly funtion of
"port_D_5_high"
  
```

```
extern "C" void port_D_6_high(void); // declare the assembly function of  
"port_D_6_high"
```

```
extern "C" void port_D_7_high(void); // declare the assembly function of  
"port_D_7_high"
```

```
#endif
```

References

- [1] S. Y. He, "Microprocessor Systems," 20-Jan-2022.
- [2] S. Y. He, "Week2_example." Toronto, 20-Jan-2022.