

## Lab 9: C programming with AVR Studio 4

The goal of this lab is to introduce you to the C programming language. We will repeat the concepts learnt from the previous labs and use C instead of assembly programming. There are two parts in this lab. The first part introduces you to a blink program and the second is the LCD display program.

**Submit lab9.c and display.c by 5:00pm on March 23<sup>rd</sup>, 2018.**

### PART 0: Lab Evaluations

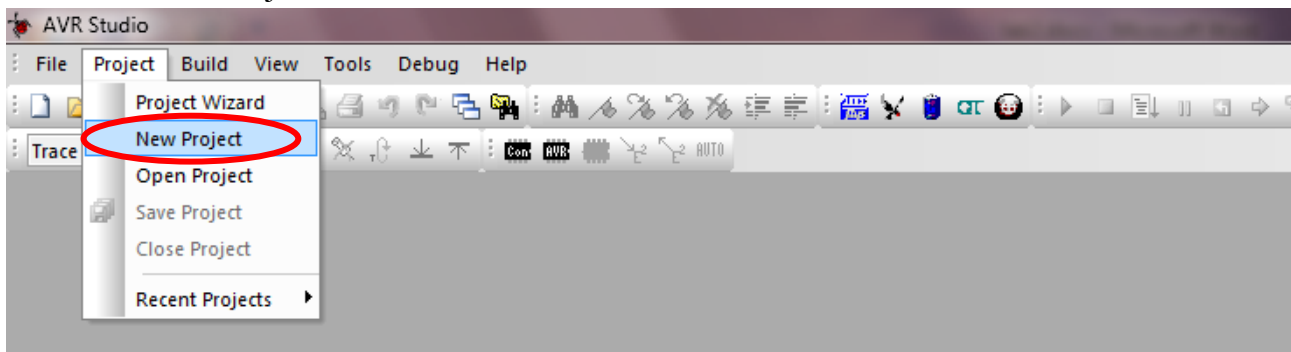
- Lab Evaluations URL: <https://evals.csc.uvic.ca/>
- Lab Evaluations instruction file is in the same directory as lab9.
- The full names of the four Lab instructors are Philip Alipour, Nirmala Gnanaratnam, Deepak Kumar and Victoria Li. Select the lab instructor for the lab you have been attending this term.
- Need a volunteer to report any problems to [insupport@csc.uvic.ca](mailto:insupport@csc.uvic.ca) or 250-472-5900.

### PART 1: LED blinking

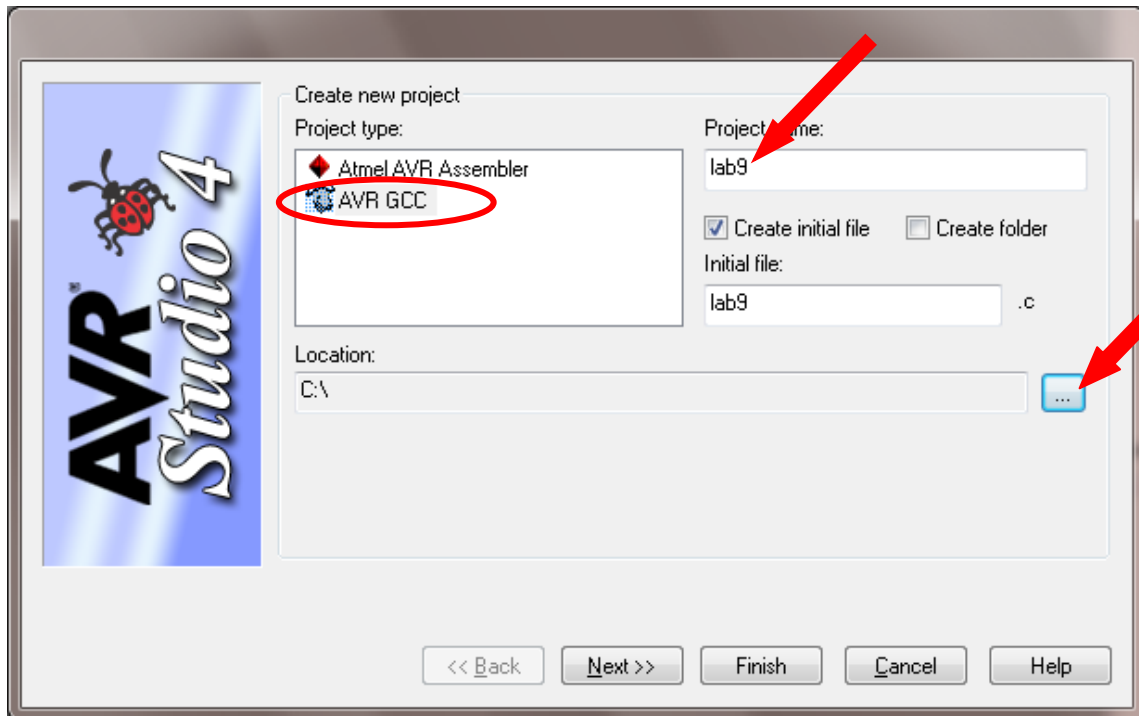
#### 1.1 Create a C project in AVR Studio 4

Start the AVR Studio 4.

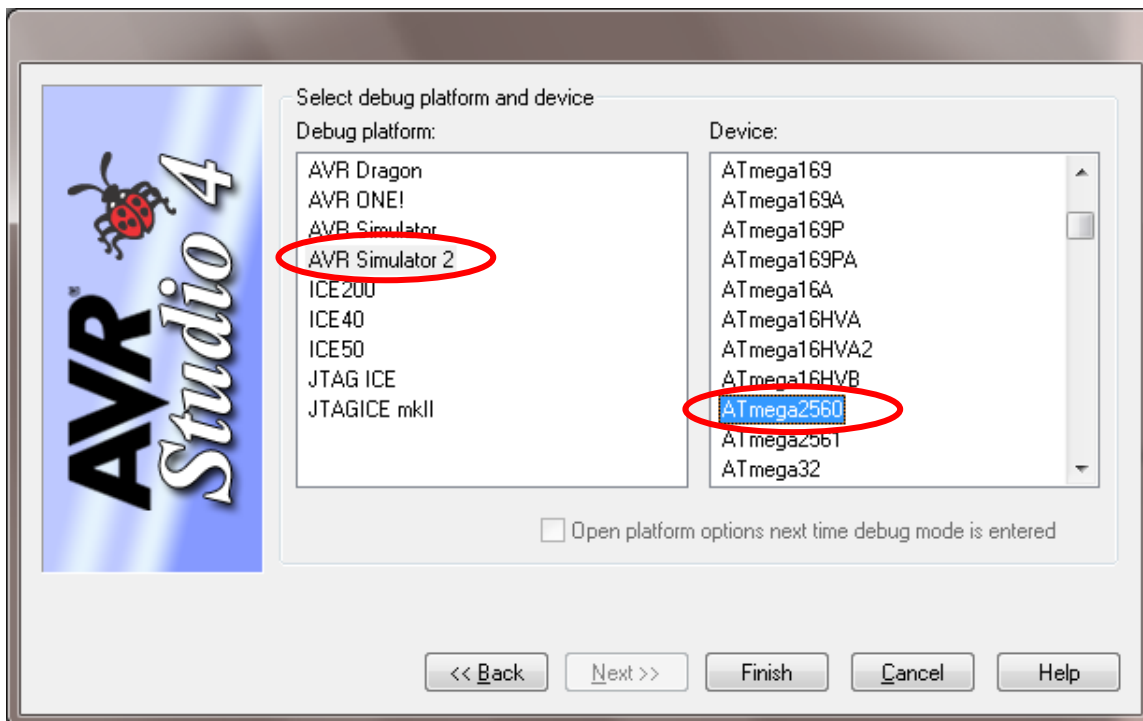
- Select “New Project”.



- Select “Project Type” → AVR GCC, name the project "lab9", click "..." button to choose a location (H drive)



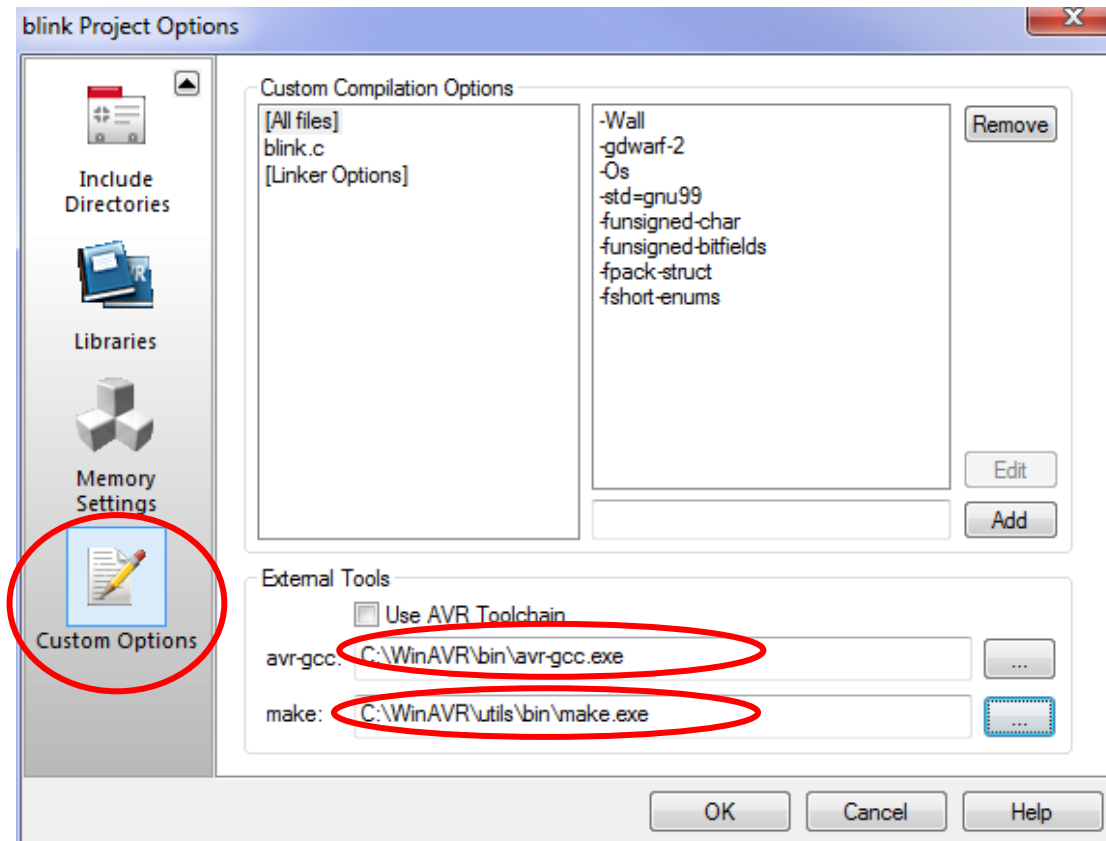
- Go to → Next.
- Select a platform: Debug Platform → AVR Simulator 2
- Select a Device: Device → ATmega2560. Click the "Finish" button.



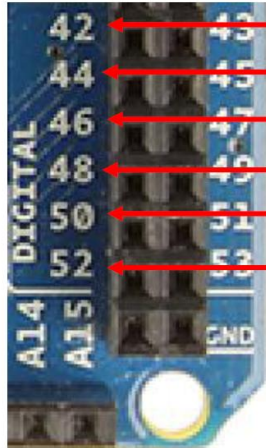
## 1.2 Compile C project

After creating a C project, it opens up the project space. There are issues with the avr-gcc compiler within the AVR Studio. Hence we need to replace this. **This step needs to be done for every project you create.**

- Go to Project → Configuration Options
- Go to the last item “Custom Options” on the left pane.
- Uncheck “Use AVR Toolchain” under External Tools at the bottom
- **avr-gcc:**           **C:\WinAVR\bin\avr-gcc.exe**
- **make:**           **C:\WinAVR\utils\bin\make.exe**



- Copy the contents of lab9.c provided to you in the Lab09 folder on Connex
- Read and try to understand the code



42	Port L: bit 7 PL7
44	Port L: bit 5 PL5
46	Port L: bit 3 PL3
48	Port L: bit 1 PL1
50	Port B: bit 3 PL3
52	Port B: bit 1 PL1

```
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

/*
 * arduino pins 42, 44, 46, 48, 50, 52
 * and Gnd (ground)
 * Pin 42 Port L: bit 7 (PL7)
 * Pin 44 Port L: bit 5 (PL5)
 * Pin 46 Port L: bit 3 (PL3)
 * Pin 48 Port L: bit 1 (PL1)
 * Pin 50 Port B: bit 3 (PB3)
 * Pin 52 Port B: bit 1 (PB1)
 */
int main (void)
{
    /* set PORTL and PORTB for output*/
    DDRL = 0xFF;
    DDRB = 0xFF;

    for ( ; ; )
    {
        /* set PORTL.7 high and PORTL.3 high
         * 1000 1000
         *
         * set PORTB all low
         * 0000 0000
         */
        PORTL = 0x88;
        PORTB = 0x00;

        _delay_ms(500);
    }

    return 1;
}
```

- Build the code.
- Check the build window below to see if there are any possible errors
- If the program compiled with no errors it is time to load the binary hex file to the unit
  - Upload the lab9.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:lab9.hex`  
 You may need to change the [path](#) and [filename](#) if the settings are different from this.
  - Or, download upload.bat to a directory named "default" in your lab9.c directory. Open a command window and get to the same directory where upload.bat is stored, and type upload.bat. **You need to change the file name if your project is not called lab9.**

### 1.3. Exercises:

Download lab9.c. Implement the following Pseudo-code:

turn the first led on;

1 second delay;

for (;){

    turn the current led off

    1 second delay;

    turn the next led on; //wrap around when appropriate

    1 second delay;

}

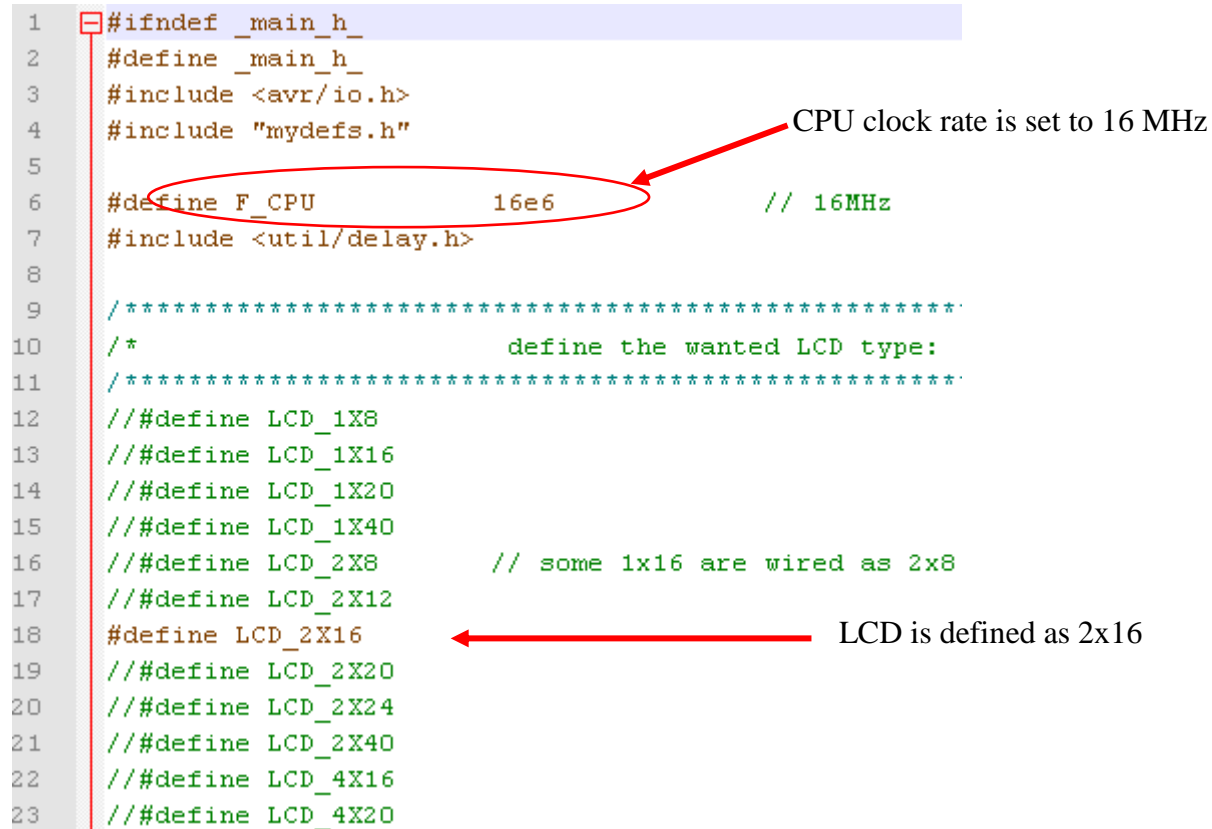
**PART 2: LCD Display**

**2.1** Write a program to calculate the length of a string and display the result on the LCD display. Import the file **main.c**. Down load the LCD driver software to the project directory. The driver is obtained online from <https://www.avrfreaks.net/>.

- Create an **avr-gcc** project called display using the AVR Studio 4.
- Make sure to use the WinAVR tools, not the AVR Toolchain (see 1.2 above).
- Import the lcd\_driver files into your project space. The driver files are:

```
mydefs.h
lcd_drv.h
lcd_drv.c
main.h
main.c
```

- Open **main.h**. The LCD is configured as a 2x16 display (two lines with 16 characters on each). The CPU clock rate is set to 16 MHz:



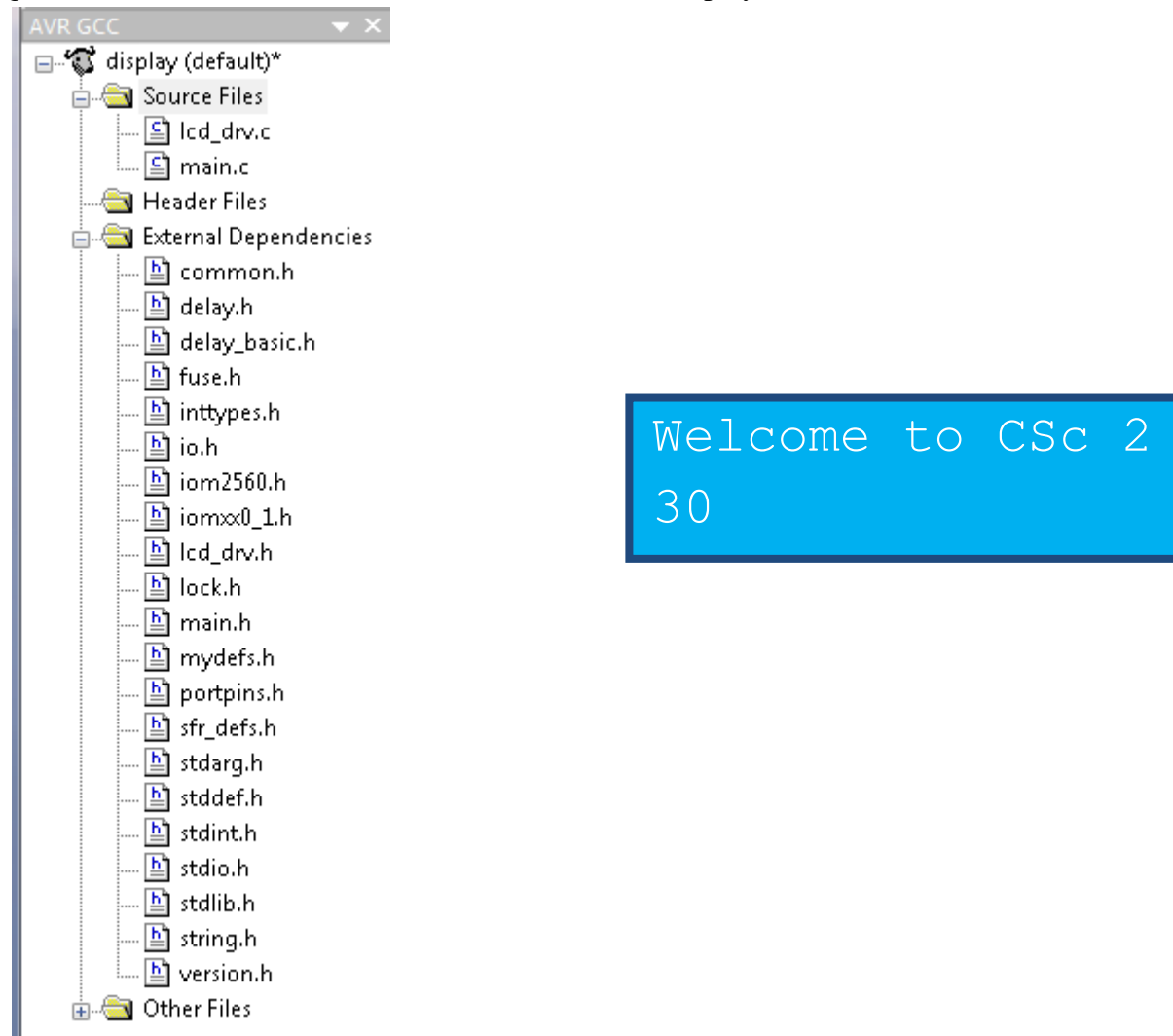
```
1  #ifndef _main_h_
2  #define _main_h_
3  #include <avr/io.h>
4  #include "mydefs.h"
5
6  #define F_CPU 16e6 // 16MHz
7  #include <util/delay.h>
8
9  /*****
10  /*          define the wanted LCD type:
11  /*****
12  // #define LCD_1X8
13  // #define LCD_1X16
14  // #define LCD_1X20
15  // #define LCD_1X40
16  // #define LCD_2X8 // some 1x16 are wired as 2x8
17  // #define LCD_2X12
18  #define LCD_2X16
19  // #define LCD_2X20
20  // #define LCD_2X24
21  // #define LCD_2X40
22  // #define LCD_4X16
23  // #define LCD_4X20
```

- The LCD is connected to the board through specific set of digital pins as outlined in the HD44780 tutorial. These are already defined under “define the LCD connections” section of **main.h** with the appropriate port pins.
- Study the simple **main.c** program:

```
int main( void )
{
    lcd_init();           ← Initialize the LCD display

    lcd_xy( 0, 0 );       ← Set the cursor to position (0, 0) – (col, row)
    /* lcd_puts takes a pointer to a null terminated
    * string and displays it at the current cursor position.
    *
    * In this call, I'm using a constant string
    */
    lcd_puts("Welcome to CSc 230"); ← Show the string on LCD display. It
                                     will continue with displaying the
                                     extra characters on the second line.
}
```

g. The file structure of the AVR studio and the LCD display:



**2.2 Exercise:** create a new file called `display.c` and add it to the project. Based on the code in `main.c`, modify `display.c` and let it show a string on the first line, then show the number of characters of that string on the second line. Let them blink for 0.5 second. (Remove `main.c` from the project.)

**Submit lab9.c and display.c by 5:00pm on March 23<sup>rd</sup>, 2018.**