

Lab 10: C Programming with Microchip Studio 7.0

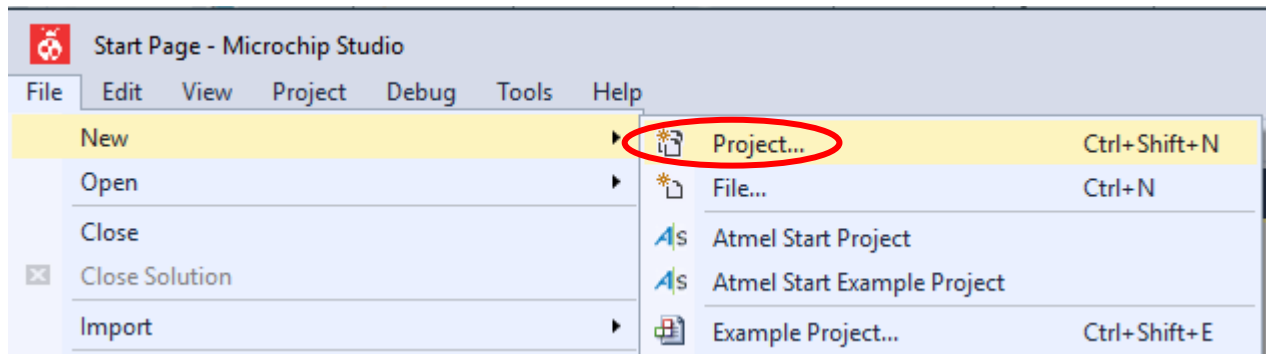
The goal of this lab is to introduce you to the C programming language. We will repeat the concepts learnt from the previous labs and use C instead of assembly programming. There are two parts in this lab. The first part introduces you to a blink program and the second is the LCD display program.

PART 1: LED blinking

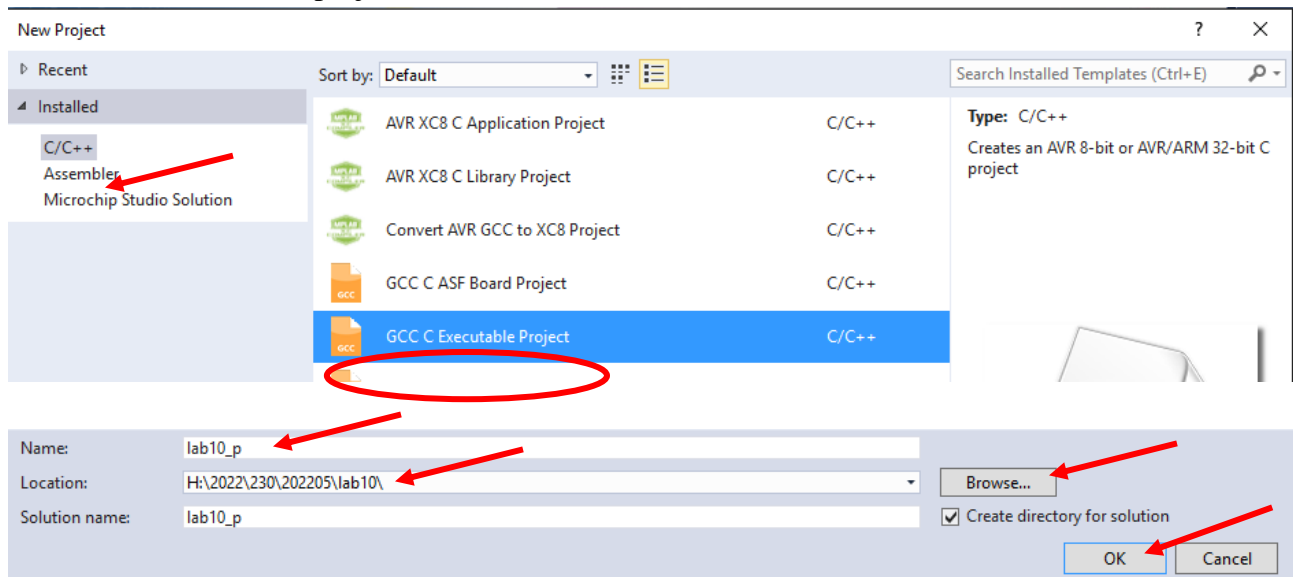
1.1 Create a C project in Microchip Studio 7.0

Start the Microchip Studio 7.0:

- From the menu, Click “File” -> “New” -> “Project...”.



- Click on “C/C++” on the left hand side, then click on “GCC C Executable Project” for the project type. Click on the “Browse...” button to choose a location (H drive), type “lab10_p” for the Name of the project, then click on the “OK” button. Refer to the screen shot below:



- The rest of the steps are similar to what we have done in the previous labs. You don't need to set entry file.

1.2 Compile C project

After the C project is created, it opens up the project space. Download “CSC230.h” file and save it in the same directory as the “main.c”. The file structure looks like this:

Name	Date modified
Debug	7/12/20
lab10_p.componentinfo.xml	7/12/20
lab10_p.cproj	7/12/20
main.c	7/12/20
CSC230.h	7/12/20

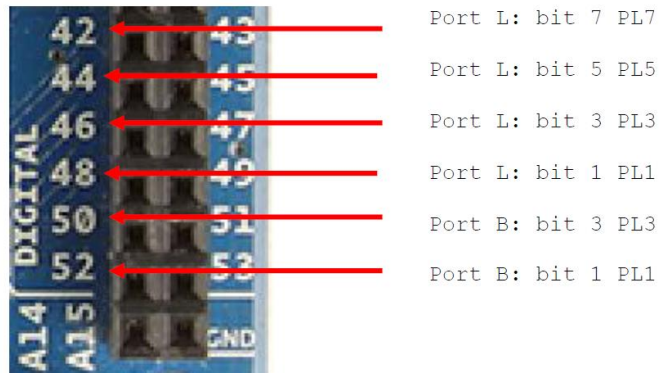
In the main.c, remove the code, including “#include <avr/io.h> “, then type the following code:

```
#include "CSC230.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;
    while (1)
    {
        /* 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;
}
```

Recall the LED lights:



Build the code and upload the .hex file to the board (the same as the previous labs).

1.3. Exercise:

Download lab10.c. Implement the following Pseudo-code (Note in C, there must be only one main method in each project. You must remove main.c from your project and add lab10.c to the project).

Download lab10.c from lab10 folder of the course website. Add it to your project. Implement the following pseudocode:

```

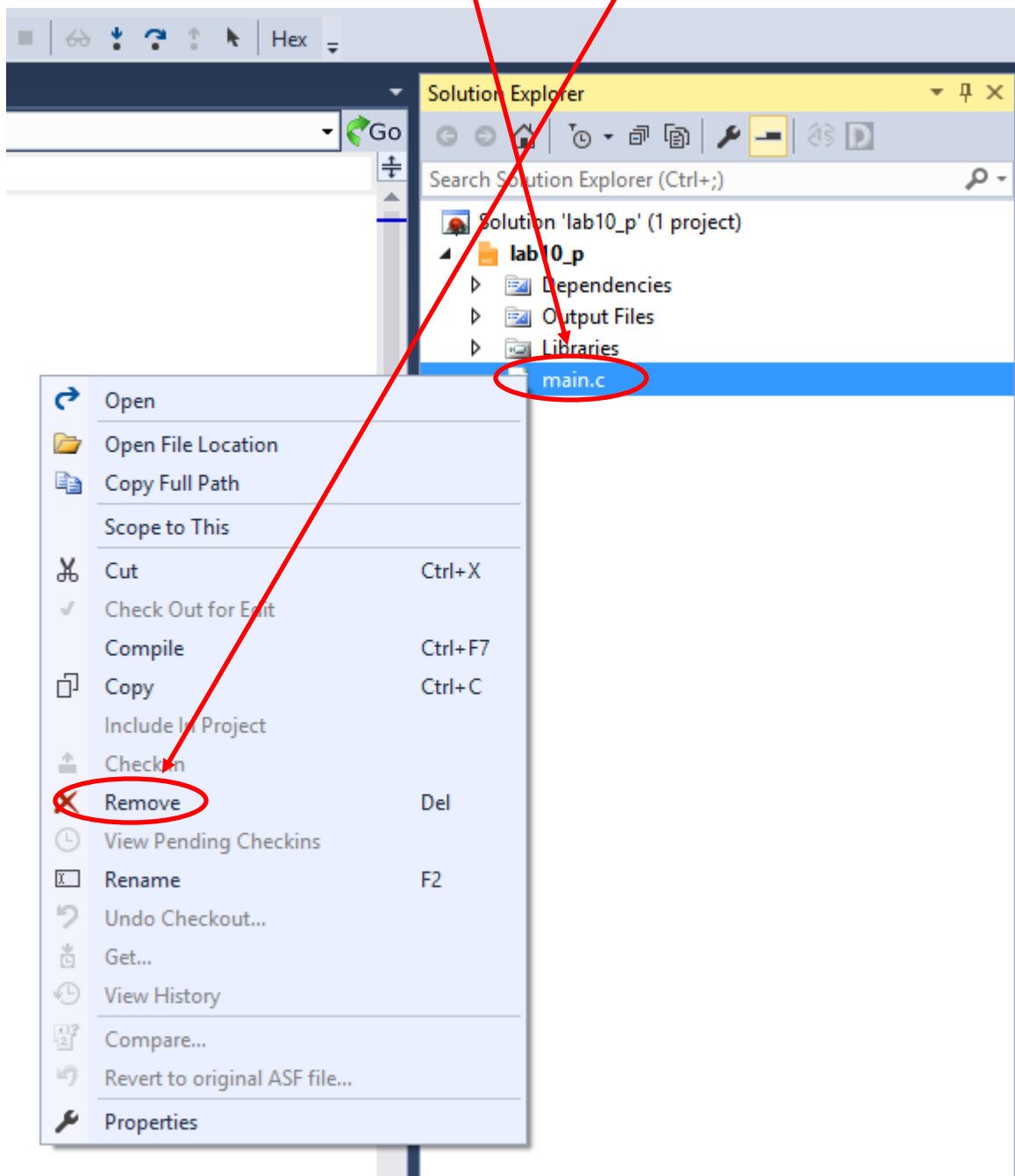
turn the first led on;
1 second delay;
for (; ){
    turn the current led off
    1 second delay;
    turn the next led on; //wrap around when appropriate
    1 second delay;
}

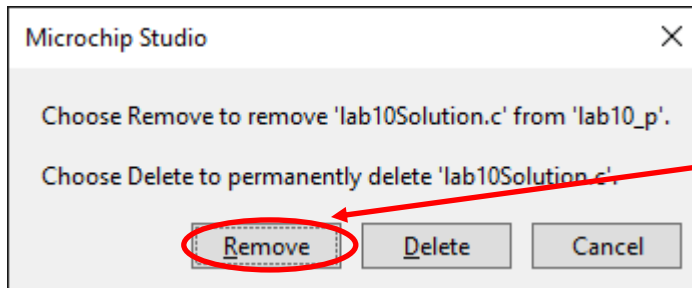
```

How to remove main.c from your project?

Refer to the diagram on the next page:

In Microchip “Solution Explorer”,
right click on “main.c”, in the pull
down menu, click “Remove”



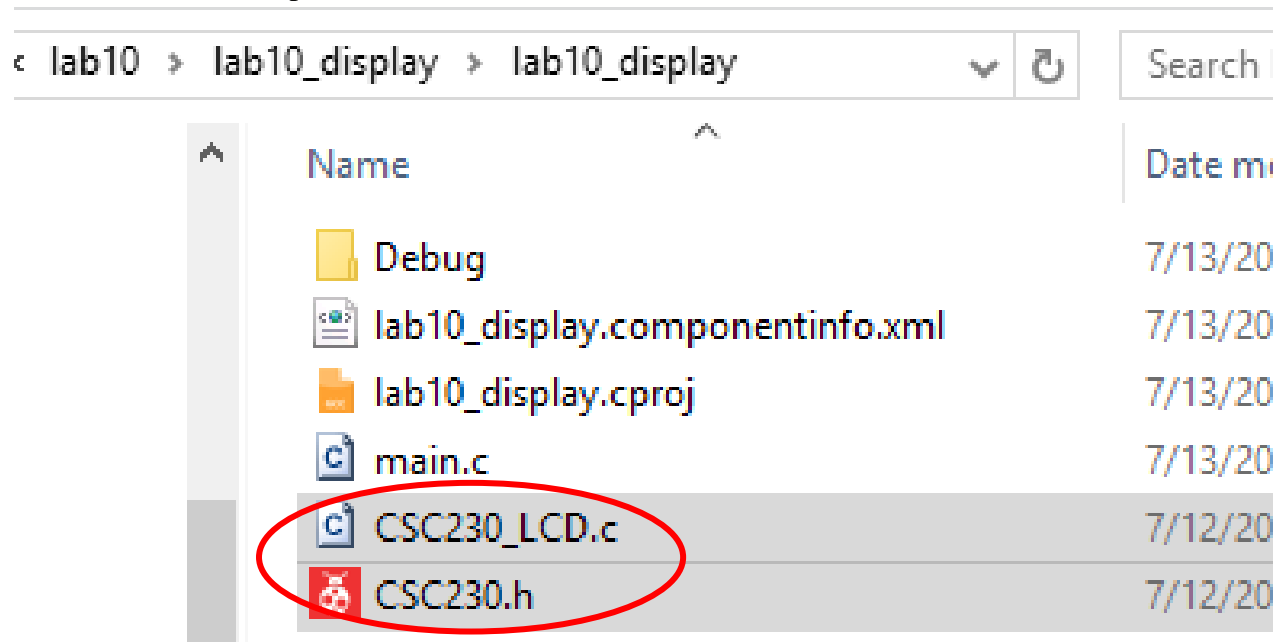


On the new pop up window, click on the "Remove" button.

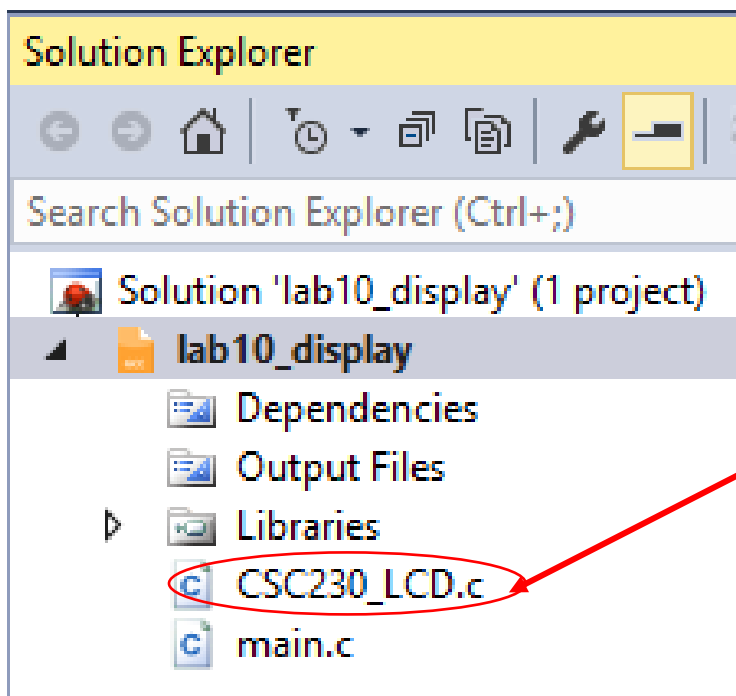
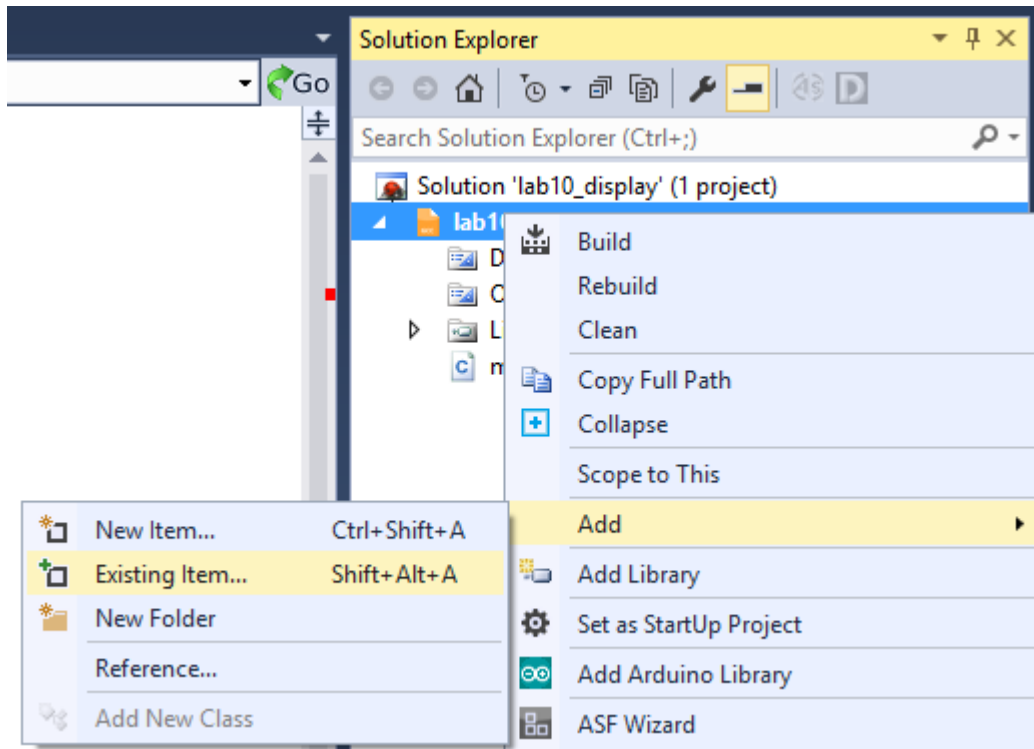
PART 2: LCD Display

2.1 Write a program to calculate the length of a string and display the length on the LCD display at row 0, column 3.

- Create a new C project named **lab10_display**.
- Download the files **CSC230.h** and **CSC230LCD.c** to the same directory as the file **main.c**. Open the Window's File Explorer, the file structure looks like this:



- In the Solution Explorer pane of Microchip Studio, add file **CSC230LCD.c** to the project. The Solution Explorer looks like this:



After downloading both *CSC230.h* and *CSC230_LCD.c* into the project directory, right click on the project name -> Add -> Existing Item...

Choose *CSC230_LCD.c*, then click on “Add” button.

CSC230_LCD.c should be added to the project and the Solution Explorer looks like the left hand side.

d. The LCD is connected to the board through specific set of digital pins as outlined in the HD44780 tutorial. These are already defined in **CSC230.h** and other driver files.

f. Type the following code in **main.c**:

```

/*
 * lab10_display.c
 *
 * Created: 7/17/2019 9:23:22 AM
 * Author : vli
 */

#include "CSC230.h"

int main(void)
{
    /* Replace with your application code */
    lcd_init(); ← Initialize the LCD display

    while (1)
    {
        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 CSC230."); ← Show the string on LCD display. It
                                         continues to display the extra
                                         characters on the second line.

        lcd_xy( 4, 1 );

        /* This function will delay for the 500 milliseconds */
        _delay_ms(500);

        /*
         * Here using a buffer. Note that this isn't the normal
         * way to initialize a C-string, but I wanted to illustrate
         * how they are created.
         */

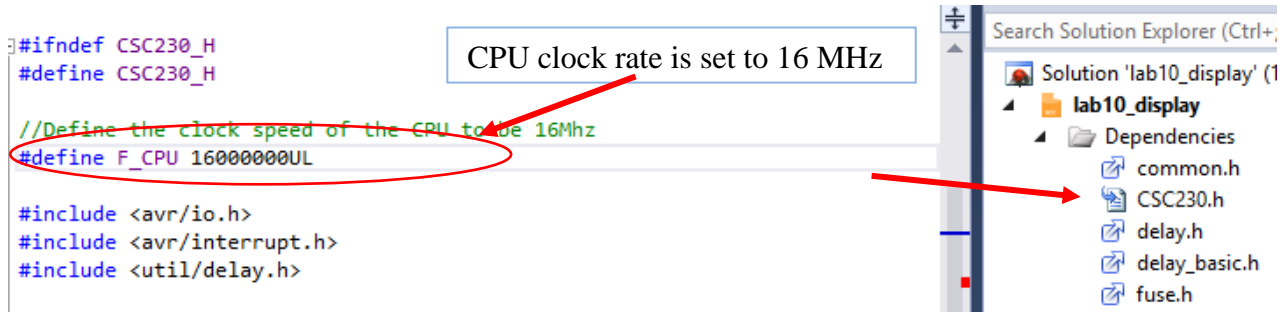
        char msg[10];
        msg[0] = 'H';
        msg[1] = 'i';
        msg[2] = 0;

        lcd_puts(msg);
        char msg1[5] = "Uvic"; //another way to declare and initialize a string

        lcd_xy(8,1);
        lcd_puts(msg1);
    }
}

```

e. Open **CSC230.h**. The CPU clock rate is set to 16 MHz:



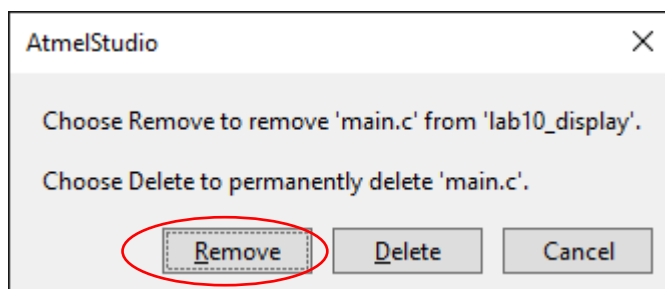
g. Build and upload the project (same steps as the previous labs). The LCD display looks like this:



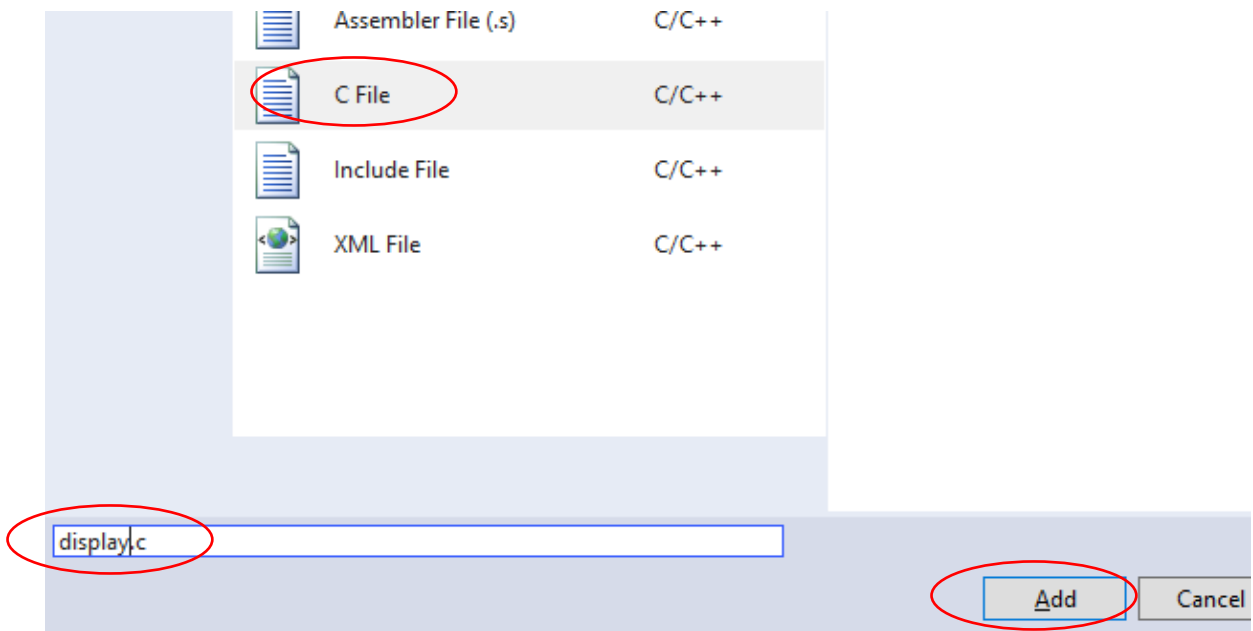
2.2 Exercise: Write a program to show a string on row 0, column 1. The length of the string ≤ 15 . Then show the length of that string on row 1, column 1. Make both blink for 0.5 seconds. In C, there must be one main method in a project, you need to remove **main.c** from the project (refer to pages 3-5 on how to remove a file from the project). Then create a new file called **display.c** and add it to the project. Based on the code in **main.c**, modify **display.c** for exercise 2.2.

Step by step instructions:

- a. **Remove main.c:** In Microchip Solution Explorer, right click on **main.c** -> Remove -> Click on the "Remove" button.



- b. **Create display.c:** Right click on the project -> Add -> Click on **New Item...** -> Click on **C File** and type **display.c** -> Click on the **Add** button:



c. Using **main.c** as an example, do exercise 2.2.

Hint:

You may find the following information helpful:

<https://cplusplus.com/reference/cstring/strlen/>

Must write `#include <string.h>` at the top of the file in order to use this method.

e.g. `int n=strlen(str);` //get the length of a string

<https://cplusplus.com/reference/cstdlib/itoa/>

Must write `#include <stdlib.h>` at the top of the file in order to use this method

e.g. `char n_as_str[5];`

`int n=90;`

`itoa(n,n_as_str, 10);` //convert an int to an array of char