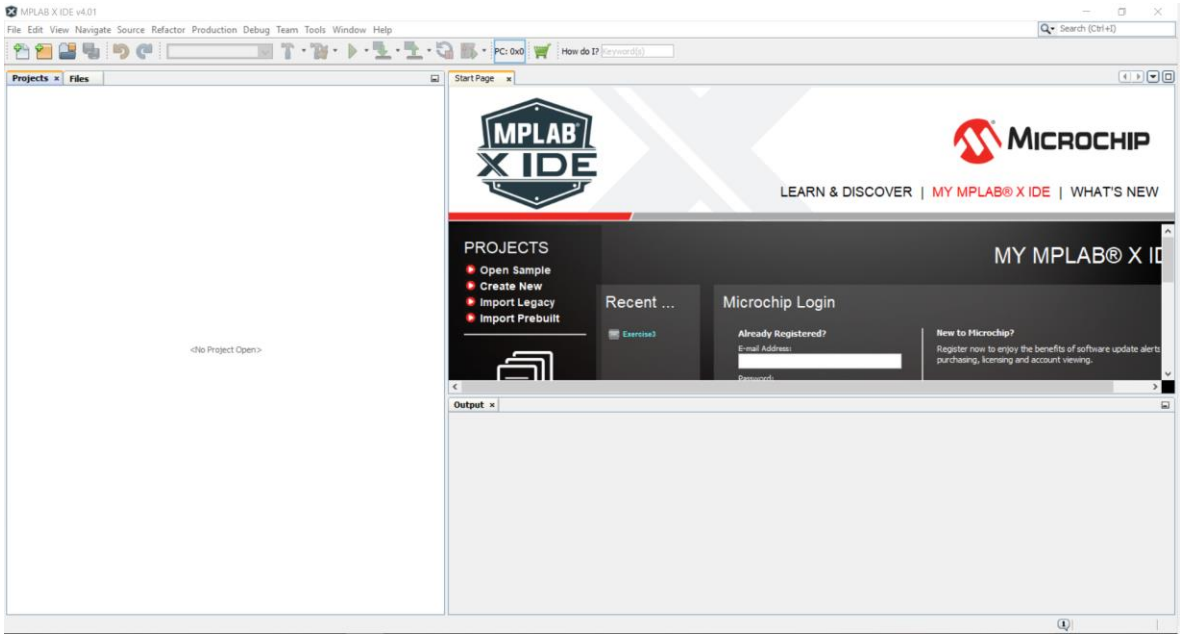
Introduction

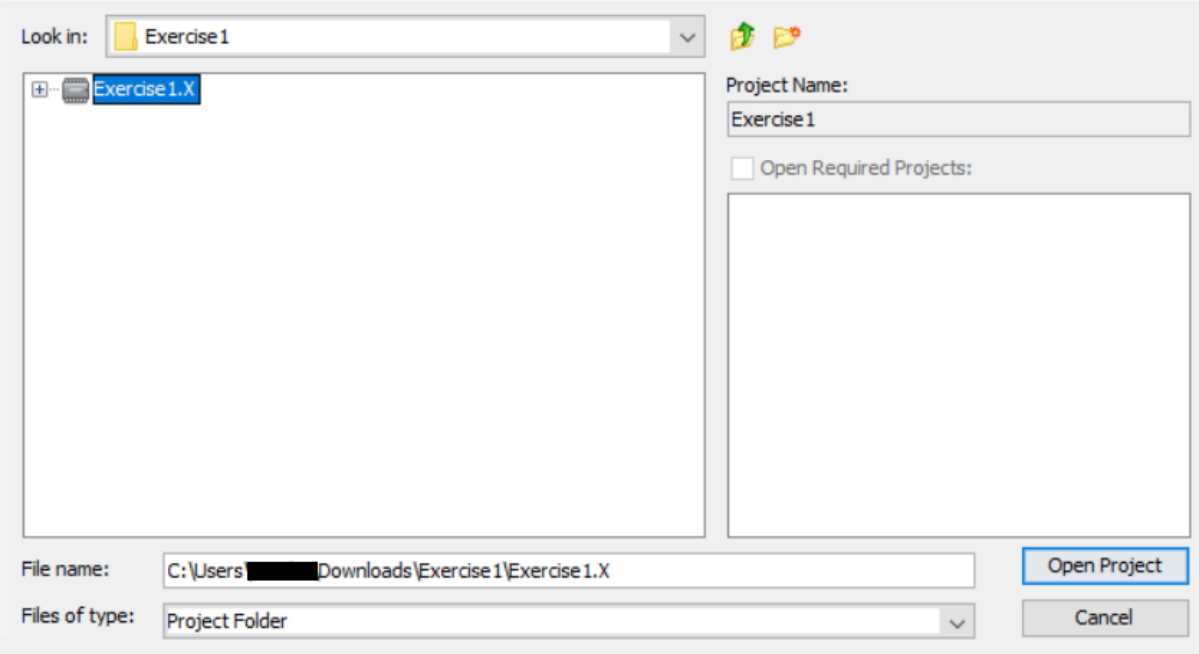
In this lab you will be working with microcontrollers. In order to program these controllers you will need to use a specific IDE called, MPLAB X IDE.

All of your code will be written and compiled by this IDE.
In addition, you have been provided with project files. These files contain the main file and necessary libraries to make all of the used functions work. As such, you may look into the libraries for a better understanding of the functions, but do not edit them.
When you first open this program, you should be presented with a similar screen.

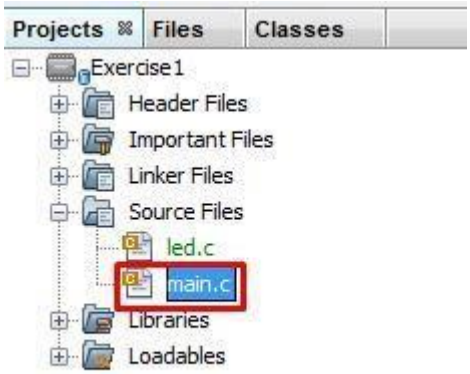


In order to open a project file choose *Open Project* from toolbar.

Then navigate to a downloaded and extracted project file.



First, you will have to open *main.c* file. To do this, choose *Source Files* and navigate to *main.c*.



It is important to remind you again that you should only edit *main.c* file, otherwise you could corrupt your project. In the *main.c* file edit only the designated rows or add new functions when necessary. This means that you should mostly limit your code to the *int main(void)* function. Editing anything else, like the *#pragma* lines, could also corrupt your project.

```
int main(void)
{

    while(1)
    {
        ///sinu kood siia!
    }
    return (1);
}
```

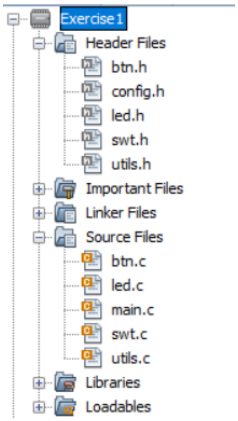
To test your code, you must upload it to the training board. Click on the *Make and Program Device Main Project* button on the toolbar.

To upload your code, you must have the training board connected to your PC via USB cable and the device turned on. Make sure you plug the cable to *Debug* port.

Exercise 1

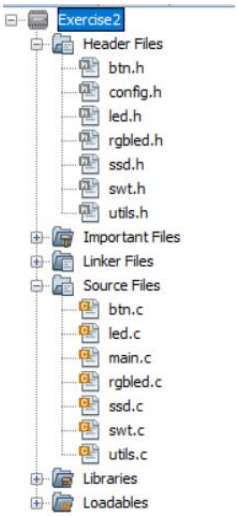
Program a single, simple logical gate (AND,OR,NAND,NOR,XOR,XNOR) given to you by your teacher. The input states are given by 2 rightmost switches. LED indicators above the switches must light up while the switch is ON. The leftmost LED is the output and must light up, if the result of the logical operation is True.

First, you will need to make sure that every necessary library is included in your project.

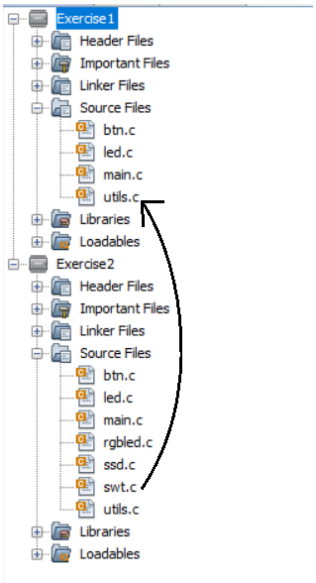


To use the switches, you will need to include switches library *swt.c* and to use the LED indicators, make sure you have *led.c* included. Both of these should be under the *Source Files*. Also, make sure the corresponding *swt.h* and *led.h* are under the *Header Files*.

If you should lack any of the corresponding *Header Files* or *Source Files*, then there are a couple of ways to fix the issue. The easiest one would be to open a different project file, that contains all the necessary libraries. For example, *Exercise2*.



The harder approach is to take the *Header* and *Source Files* from a different project. Open two different projects. Then drag the corresponding *Source File* to your target project.



Do the same for the *Header Files*, as you need both for the functions to work.
After you have done this, check to make sure that you also have the appropriate *#include* lines at the top of the file.

```
#include "btn.h"
#include "swt.h"
#include "led.h"
```

These are necessary for using any of the functions.

To use functions from the libraries they must be initialized.

To initialize the switches, add SWT_Init() at the beginning of the main function, and to initialize the LED indicators, add LED_Init()to the main function.
These initializations should be done outside of the *while(1)* loop in the main function. Anything that you want to continually run on the board, such as the LED and switch functions, should be located in the *while(1)* loop.

Now, to know at which state a switch is currently, you will have to call a `SWT_GetValue(int x)` function. It will return an integer (1 if the switch is ON and 0 if the switch is off). It takes 1 parameter, an integer which points to the switch you want to read (for the rightmost switch the parameter equals 0, next switch equals 1 and so on).

Next you will have to turn on some LED indicators. The function to do that is `LED_SetValue(int x, int y)`. It is a void type function, that takes 2 parameters. The first is the number of the LED you desire to turn on or off and the second is if you want to turn the led on, in which case it is 1, or off, in which case it is 0. The order of the LED-s follow the same principle as switches, the rightmost one equals 0 and the on left of it equals 1 and so on and so forth.