Psychomotor Activity 2: Implementation + Results

S. No	Criteria	Performance	Description	Marks
1	**Branch, Loop & Subroutine Implementation (4)**	Level 4	Correctly implements the assigned branch, loop and subroutine program independently while following appropriate laboratory practices.	4
		Level 3	Implements the assigned branch, loop and subroutine program with minor errors or assistance and generally follows proper laboratory practices.	3
		Level 2	Partially implements the assigned branch, loop and subroutine program and requires considerable assistance or repeated guidance.	2
		Level 1	Implementation of the assigned branch, loop and subroutine program is largely incomplete, requires substantial assistance, or shows poor adherence to proper laboratory and safety practices.	1
2	**Execution & Debugging (2)**	Level 2	Correctly performs the required execution and debugging and verifies the expected operation.	2
		Level 1	Performs the basic execution and debugging with minor errors or assistance.	1
3	**Control-Flow & Result Analysis (4)**	Level 4	Obtains correct control-flow and result analysis and appropriately verifies and interprets the required experimental results.	4
		Level 3	Obtains mostly correct control-flow and result analysis with minor errors in verification or interpretation.	3
		Level 2	Obtains partially correct control-flow and result analysis with noticeable errors or omissions.	2
		Level 1	Control-flow and result analysis is largely incorrect or incomplete and requires substantial assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Affective)

Affective Activity: Lab Report + Viva Voce

S. No	Criteria	Performance	Description	Marks
1	**Lab Report Completeness & Organization (4)**	Level 4	Report is complete, well-organized, properly presented, and demonstrates responsible completion of the assigned laboratory work.	4
		Level 3	Report is complete with minor omissions or organizational issues and demonstrates satisfactory responsibility for the assigned work.	3
		Level 2	Report is partially complete with several omissions and demonstrates limited responsibility for the assigned work.	2
		Level 1	Report has major omissions or poor organization and demonstrates minimal responsibility for the assigned work.	1
2	**Control-Flow Output & Conclusion (2)**	Level 2	The control-flow output and conclusion are correctly presented and discussed, demonstrating ownership of the assigned work.	2
		Level 1	The control-flow output and conclusion are presented with minor errors or omissions.	1
3	**Viva Voce (4)**	Level 4	Clearly explains the assigned experiment, branch instructions, loops, subroutines, program explanation, working, and results, demonstrating individual responsibility, participation, and effective communication.	4
		Level 3	Explains most of the assigned experiment and results correctly, with minor difficulty, and demonstrates good individual participation.	3
		Level 2	Explains basic aspects of the assigned experiment with some prompting and demonstrates basic participation.	2
		Level 1	Demonstrates limited understanding or involvement and requires substantial prompting to explain the assigned work.	1
Total Obtained				**/10**

The student has not submitted the report and has not appeared in the Viva Voce.

Criteria	Description	Marks
Lab Report and Results Not Submitted	Student has not submitted the report and results file.	**0**
Absent in Viva Voce	Student is absent from the viva voce.	**0**

Instructor's Signature: _____ Date: _____

Experiment No 6

Keypad and Seven-Segment Display Interface with STM32

Objectives:

- To learn the basic input and output interface with STM32F103 Board
- Interfacing Keypad with STM32F103
- Interfacing the Seven Segment Display with STM32F103 to view the output

Keypad Interface:

Embedded system applications usually require large number of buttons connected (ex. calculator,cell phone). Keypad is one of the input devices that are commonly used in embedded systems which require large number of buttons. 4x3 matrix keypads have 12 buttons. We can connect these buttons directly to the GPIO pins of STM32F103, but it will eat up precious GPIO pins (If we have 12 buttons, then we need 12 GPIO pins). To avoid this trouble, keypad uses a technique that will save GPIO pins. In this technique, buttons are connected in a matrix (row/column) style.

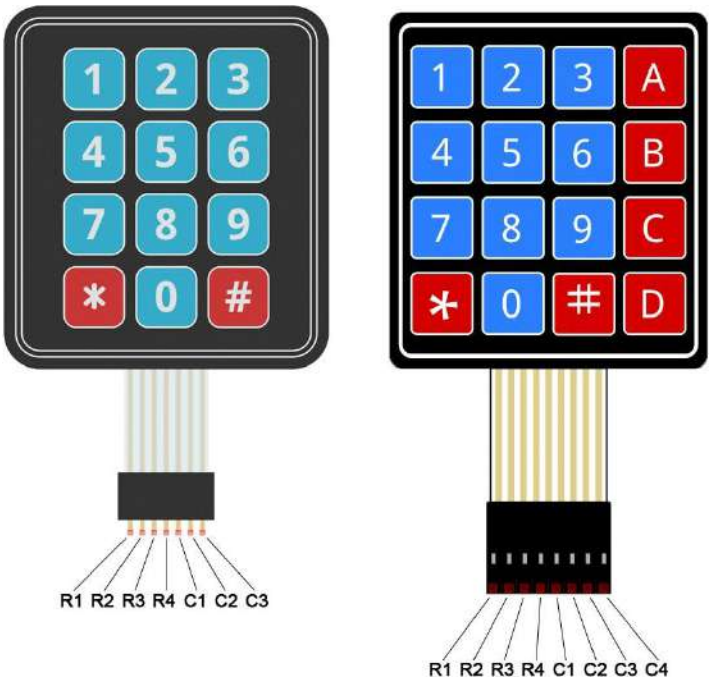


Figure 1: 4x3 and 4x4 Keypad

4x4 matrix keypads have 16 buttons. We can connect these buttons directly to the GPIO pins of STM32F103, but it will take 16 GPIO pins so the 4x4 keypad uses the matrix (row/column) technique just like in the 4x3 keypad.

The internal connection of the buttons in 4x3 matrix keypad is like this:

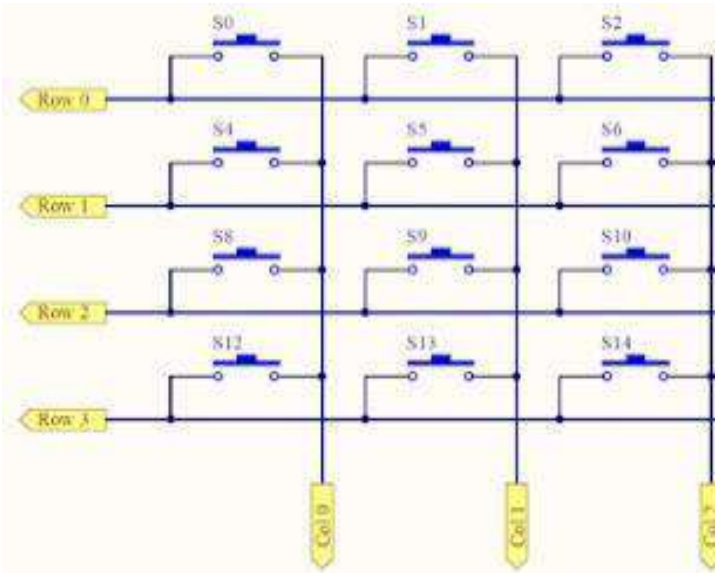


Figure 2: 4x3 Keypad Internal Circuitry

From the image above, for example if S8 is pressed, then Col-0 and Row 2 is connected. We can detect this by using a microcontroller. The microcontroller must scan the keypad by writing the columns and reading the rows.

To connect this keypad to the microcontroller, we need 7 GPIO pins. The column pins will be connected to the GPIO pins as open-drain or push-pull output. The row pins will be connected to the GPIO pins as inputs. At first, the column pins will be set to logic 1, and the other columnpins will be set to logic 0 condition. So, when no button pressed in this column, then all row pins will receive logic 0. When there is a button pressed on this column, then the corresponding row pin for that button will receive logic 1. The microcontroller can read the rows and decide what to display on the seven-segment display.

Seven-Segment Display Interface:

The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimaland Hex) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected

together to display numbers greater than ten.

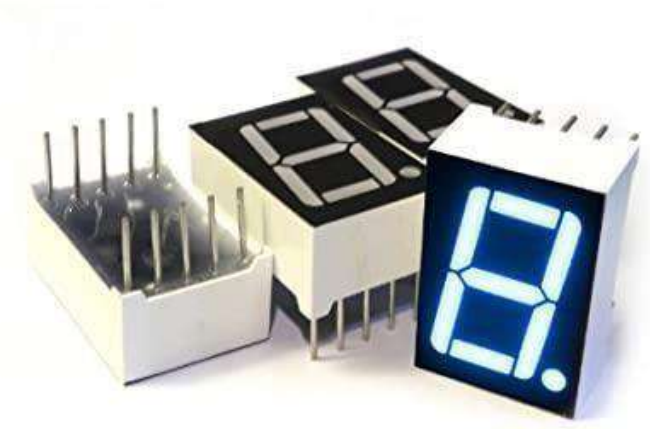


Figure 4: 7-Segment Displays

The displays common pin is generally used to identify which type of 7-segment display it is. As each LED has two connecting pins, one called the "Anode" and the other called the "Cathode", there are therefore two types of LED 7-segment display called: **Common Cathode** (CC) and **Common Anode** (CA).

The difference between the two displays, as their name suggests, is that the common cathode has all the cathodes of the 7-segments connected directly together and the common anode has all the anodes of the 7-segments connected together and is illuminated as follows.

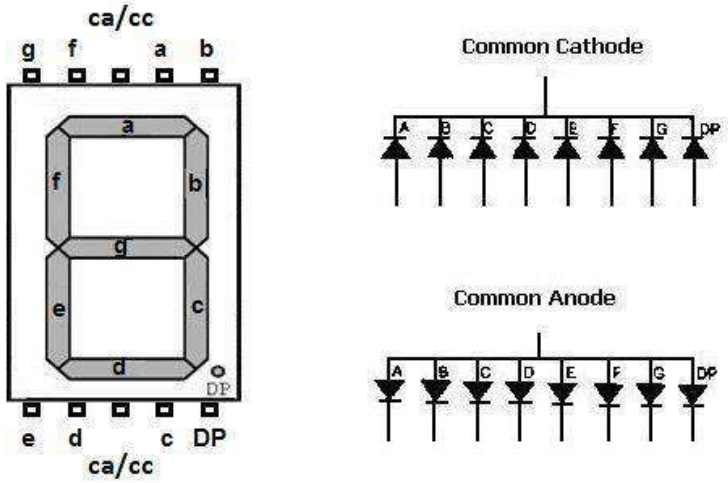


Figure 5: Pin Configuration and CA/CC circuitry

In this lab, seven-segment with common anode will be used

Required GPIO Configurations for Keypad:

```
void configureKeyPadpins(void)
{
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);

	//GPIO Configuration for Columns

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_13;                                   // Column 1
	GPIO_Init(GPIOA,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_14;                                   // Column 2
	GPIO_Init(GPIOA,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_15;                                   // Column 3
	GPIO_Init(GPIOA, &myLED);

	//GPIO Configuration for Rows
	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_IN_FLOATING;
	myLED.GPIO_Pin = GPIO_Pin_11;                                   // Row A
	GPIO_Init(GPIOA,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_IN_FLOATING;
	myLED.GPIO_Pin = GPIO_Pin_10;                                   // Row B
	GPIO_Init(GPIOA, &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_IN_FLOATING;
	myLED.GPIO_Pin = GPIO_Pin_9;                                    // Row C
	GPIO_Init(GPIOA, &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_IN_FLOATING;
	myLED.GPIO_Pin = GPIO_Pin_8;                                    // Row D
	GPIO_Init(GPIOA, &myLED);
}
```

Required GPIO Configurations for Seven-Segment Display:

Set Column Functions:

\

```
void configureLEDpins(void)
{
	RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB,    ENABLE);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_0;                    // Pin – A (7-Segment)
	GPIO_Init(GPIOB,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_1;                    // Pin – B (7-Segment)
	GPIO_Init(GPIOB,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_2;                    // Pin – C (7-Segment)
	GPIO_Init(GPIOB,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_3;                    // Pin – D (7-Segment)
	GPIO_Init(GPIOB,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_4;                    // Pin – E (7-Segment)
	GPIO_Init(GPIOB,  &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_5;                    // Pin – F (7-Segment)
	GPIO_Init(GPIOB, &myLED);

	myLED.GPIO_Speed = GPIO_Speed_2MHz;
	myLED.GPIO_Mode=GPIO_Mode_Out_PP;
	myLED.GPIO_Pin = GPIO_Pin_6;                    // Pin – G (7-Segment)
	GPIO_Init(GPIOB,  &myLED);
}
```

```
164  void SetColumnsLow(void)
165  {
166    GPIO_WriteBit(GPIOA, GPIO_Pin_13,0);       // Col-1 = 0
167    GPIO_WriteBit(GPIOA, GPIO_Pin_14,0);       // Col-2 = 0
168    GPIO_WriteBit(GPIOA, GPIO_Pin_15,0);       // Col-3 = 0
169
170    Col_1=0;
171    Col_2=0;
172    Col_3=0;
173  }
174  void SetColumn_1(void)
175  {
176    GPIO_WriteBit(GPIOA, GPIO_Pin_13,1);       // Col-1 = 1
177    GPIO_WriteBit(GPIOA, GPIO_Pin_14,0);       // Col-2 = 0
178    GPIO_WriteBit(GPIOA, GPIO_Pin_15,0);       // Col-3 = 0
179
180    Col_1=1;
181    Col_2=0;
182    Col_3=0;
183  }
184  void SetColumn_2(void)
185  {
186    GPIO_WriteBit(GPIOA, GPIO_Pin_13,0);       // Col-1 = 0
187    GPIO_WriteBit(GPIOA, GPIO_Pin_14,1);       // Col-2 = 1
188    GPIO_WriteBit(GPIOA, GPIO_Pin_15,0);       // Col-3 = 0
189
190    Col_1=0;
191    Col_2=1;
192    Col_3=0;
193  }
194  void SetColumn_3(void)
195  {
196    GPIO_WriteBit(GPIOA, GPIO_Pin_13,0);       // Col-1 = 0
197    GPIO_WriteBit(GPIOA, GPIO_Pin_14,0);       // Col-2 = 0
198    GPIO_WriteBit(GPIOA, GPIO_Pin_15,1);       // Col-3 = 1
199
200    Col_1=0;
201    Col_2=0;
202    Col_3=1;
203  }
```

Libraries to include:

```
1  #include "stm32f10x.h"
2  #include "stm32f10x_rcc.h"
3  #include "stm32f10x_gpio.h"
```

Main Function:

```
25  int main(void)
26  {
27
28      configureLEDpins();
29
30      configureKeyPadpins();
31
32      GPIOB->ODR = 0x00BF; //Displaying -
33
34      SetColumnsLow();
35      ms_Delay(10);
36
37      while (1)
38      {
39          SetColumn_1();
40          //ms_Delay(10);
41          KeyPad_Scan();
42
43          SetColumn_2();
44          //ms_Delay(10);
45          KeyPad_Scan();
46
47          SetColumn_3();
48          //ms_Delay(10);
49          KeyPad_Scan();
50      }
51  }
```

Programming in KEIL:

To program the interfaces in Keil, the first step is to create a new uVision Project and selectingthe device which is STM32F103C8.

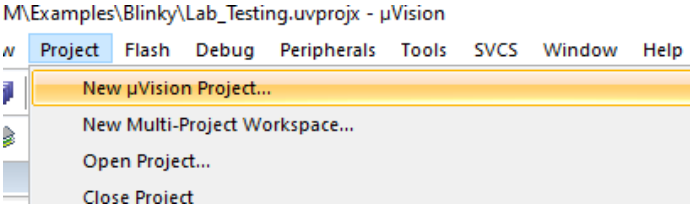


Figure 6: New Project Creation

After saving the project in your desired directory, the device will be selected in the „SelectDevice for Target“ dialogue box.

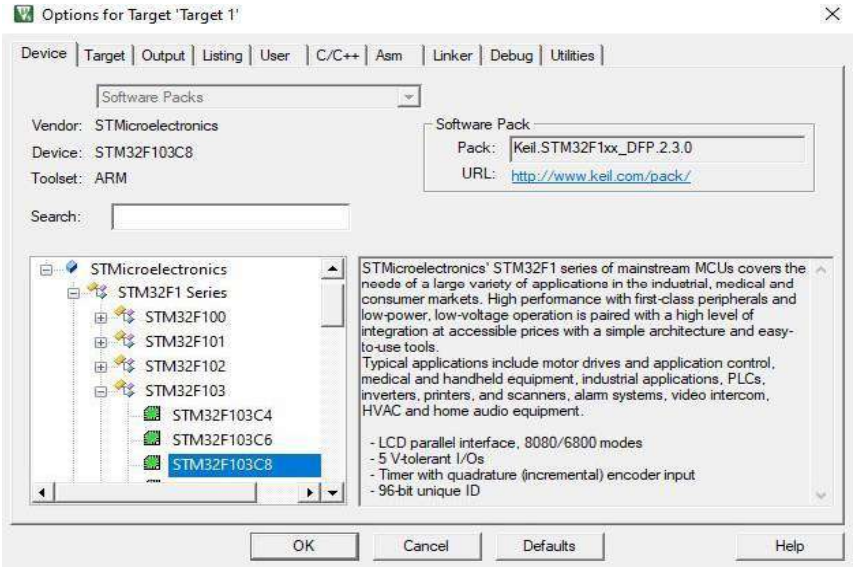


Figure 7: Device Selection

After that, include the libraries as shown in the figure below:

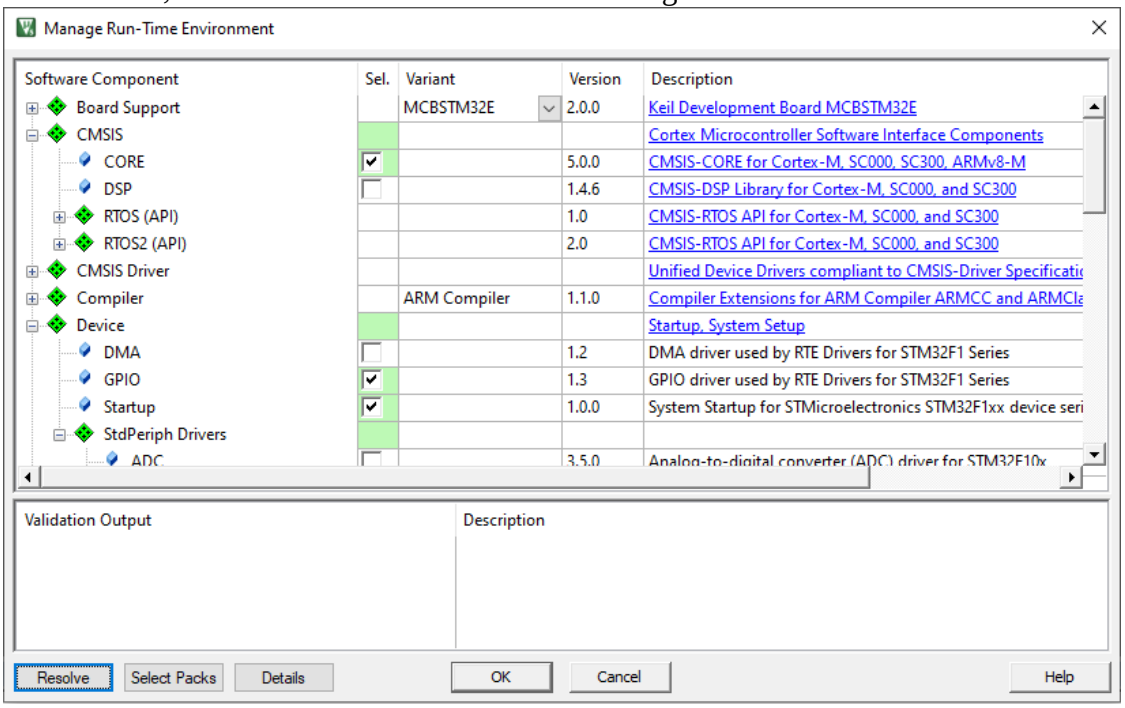


Figure 8: Adding required libraries

Then add a new item into your source group.

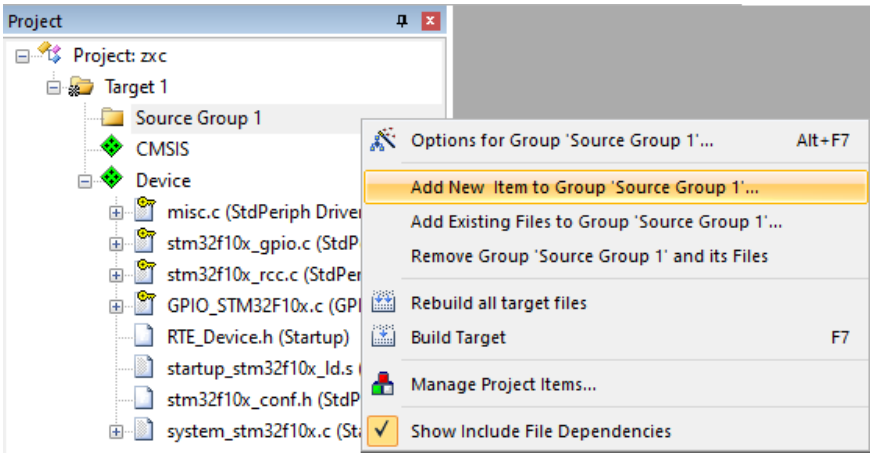


Figure 9: Adding new file

Select the “C File (.c)” option, name your C file and save it in your desired directory.

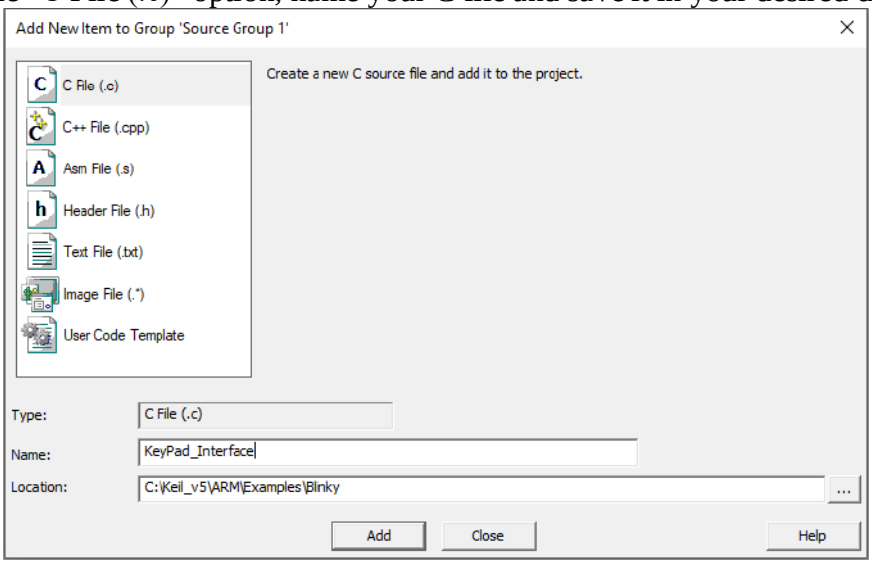


Figure 10: Adding C file

Lab Task (Psychomotor)

Activity 01:

Write C-Program to interface 3x4 keypad and Seven Segment display with STM32F103 board. Program will read the keys pressed by the user on the keypad and display the number on Seven Segment display. Implement the design using STM32F103 board and show the results to lab instructor.

Activity 02:

Design and implement a Digital Lock by interfacing Keypad and 7-Segment with the STM32F103 board. The design will get 4-Digit numeric pin code as input from the user. If the pin is correct the 7-Segment will display U (for unlocked state) and if the input is incorrect or not entered yet the 7-Segment will keep on displaying L (for locked state).

Lab Task (Cognitive)

Question 1:

A common-cathode seven-segment display is connected to PB0-PB6 with segments a-g mapped to bits 0-6 (a = bit 0, g = bit 6). The decimal point (bit 7) is not used and is kept OFF.
A 3x4 keypad (rows: 1 2 3 / 4 5 6 / 7 8 9 / * 0 #) is scanned by driving columns and reading rows with pull-up resistors.

a) Determine the hexadecimal codes required to display the digits 4 and 2. **[4 Marks]**

b) Determine the code required to display digit 4 if a common-anode display were used instead (8-bit value, decimal point OFF). **[2 Marks]**

c) Key '5' is pressed. Describe how the scan routine identifies it and state the purpose of the pull-up resistors on the row lines. **[4 Marks]**

LABORATORY SKILLS ASSESSMENT (Cognitive)

Obtained Marks	**/10**

Instructor's Signature: _____ Date: _____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Psychomotor Activity 1: Experimental Setup

S. No	**Criteria**	**Performance**	**Description**	**Marks**
1	**Keypad & Seven-Segment Hardware Setup (4)**	Level 4	Correctly prepares the required keypad and seven-segment hardware setup for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required keypad and seven-segment hardware setup with minor errors or assistance.	3
		Level 2	Partially prepares the required keypad and seven-segment hardware setup and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required keypad and seven-segment hardware setup and requires substantial assistance.	1
2	**Pin Mapping & Port Configuration (4)**	Level 4	Correctly prepares the required pin mapping and port configuration for the experiment independently and according to the laboratory requirements.	4
		Level 3	Correctly prepares the required pin mapping and port configuration with minor errors or assistance.	3
		Level 2	Partially prepares the required pin mapping and port configuration and requires considerable assistance.	2
		Level 1	Shows limited ability to prepare the required pin mapping and port configuration and requires substantial assistance.	1
3	**Build & Simulation Parameters (2)**	Level 2	Correctly sets the required build and simulation parameters for the experiment.	2
		Level 1	Sets the basic build and simulation parameters with minor errors or assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	**Description**	**Marks**
Absent	Student is absent from the laboratory activity.	**0**

Psychomotor Activity 2: Implementation + Results

S. No	Criteria	Performance	Description	Marks
1	**Keypad & Display C Program (4)**	Level 4	Correctly implements the assigned keypad and display C program independently while following appropriate laboratory practices.	4
		Level 3	Implements the assigned keypad and display C program with minor errors or assistance and generally follows proper laboratory practices.	3
		Level 2	Partially implements the assigned keypad and display C program and requires considerable assistance or repeated guidance.	2
		Level 1	Implementation of the assigned keypad and display C program is largely incomplete, requires substantial assistance, or shows poor adherence to proper laboratory and safety practices.	1
2	**Execution & Debugging (2)**	Level 2	Correctly performs the required execution and debugging and verifies the expected operation.	2
		Level 1	Performs the basic execution and debugging with minor errors or assistance.	1
3	**Key-Detection & Display Output Analysis (4)**	Level 4	Obtains correct key-detection and display output analysis and appropriately verifies and interprets the required experimental results.	4
		Level 3	Obtains mostly correct key-detection and display output analysis with minor errors in verification or interpretation.	3
		Level 2	Obtains partially correct key-detection and display output analysis with noticeable errors or omissions.	2
		Level 1	Key-detection and display output analysis is largely incorrect or incomplete and requires substantial assistance.	1
Total Obtained				**/10**

If a student is absent:

Criteria	Description	Marks
Absent	Student is absent from the laboratory activity.	**0**

Instructor's Signature: _____ Date: _____