ANKARA YILDIRIM BEYAZIT UNIVERSITY

EE226 INTRODUCTION TO MICROPROCESSORS

LABORATORY MANUAL - 1

-Assembly Programming-

Purpose

- The general purpose of this laboratory is to familiarize you with the software development steps using the Keil uVision simulator. In the following labs, we will use uVision for debugging on the real board, but for this lab, we will just use the **simulator.**
- You will learn how to perform **digital input/output** on parallel ports of the **TM4C123**. Software skills you will learn include **port initialization**, **logic operations**, and **unconditional branching**.
- **Do not use any conditional branches in your solution**! We want you to think of the solution in terms of logical and shift operations. Logical operations include **AND, ORR** and **EOR**. Shift operations include **LSL**(logical shift left) and **LSR**(logical shift right).

System Requirements

- The objective of this system is to implement an **even/odd parity system**. There are three bits of inputs and one bit of output. The output is in positive logic: outputing a 1 will turn on the LED, outputing a 0 will turn off the LED. *Inputs are negative logic: meaning if the switch not pressed is the input is 1, if the switch is pressed the input is 0.*
    - **PE3** is an input,
    - **PE4** is an input,
    - **PE5** is an input,
    - **PE2** is an output.
- The specific operation of this system
    - Initialize Port E to make PE3, PE4, PE5 inputs and PE2 an output,
    - Make the output 1 if there is an **even** number of switches pressed, otherwise make the output 0. **(changes according to last digit of Student ID number)**
    - Make the output 1 if there is an **odd** number of switches pressed, otherwise make the output 0. **(changes according to last digit of Student ID number)**
    - Over and over, read the inputs, calculate the parity bit and set the parity bit at the output

- For example, even parity is an algorithm used in communication systems to detect errors during transmission. Consider the three inputs as a 3-bit data value, such that if the input switch is pressed, that data bit is 1. Your system will add one output bit, creating a 4-bit value, such that the number of ones, considered as one 4-bit value will always be even. The communication system (if there were one) sends the 4-bit value as a message (containing the 3-bit data plus parity), and the receiver could detect if one of the bits were to be flipped during transmission.

- The input data refers to the switch, not the input. The following table illustrates the expected behavior relative to output **PE2** as a function of inputs **PE3**, **PE4**, **PE5** (negative logic with respect to the switches).

PE3	**PE4**	**PE5**	**PE2**	
0	0	0	1/0	3 switches pressed, odd
0	0	1	0/1	2 switches pressed, even
0	1	0	0/1	2 switches pressed, even
0	1	1	1/0	1 switch pressed, odd
1	0	0	0/1	2 switches pressed, even
1	0	1	1/1	1 switch pressed, odd
1	1	0	1/0	1 switch pressed, odd
1	1	1	0/1	no switch pressed, even

Laboratory Work

1. Verify Keil Project for Lab1 is present and runs

To work on Lab 1, perform these tasks. Find a place on your hard drive to save all your TM4C123 software. In Figure 1 it is called **EE319KwareSpring2021**, created when you install the .exe. Download and unzip the starter configuration from:

http://users.ece.utexas.edu/~valvano/Volume1/EE319K_Install.exe

into this location.

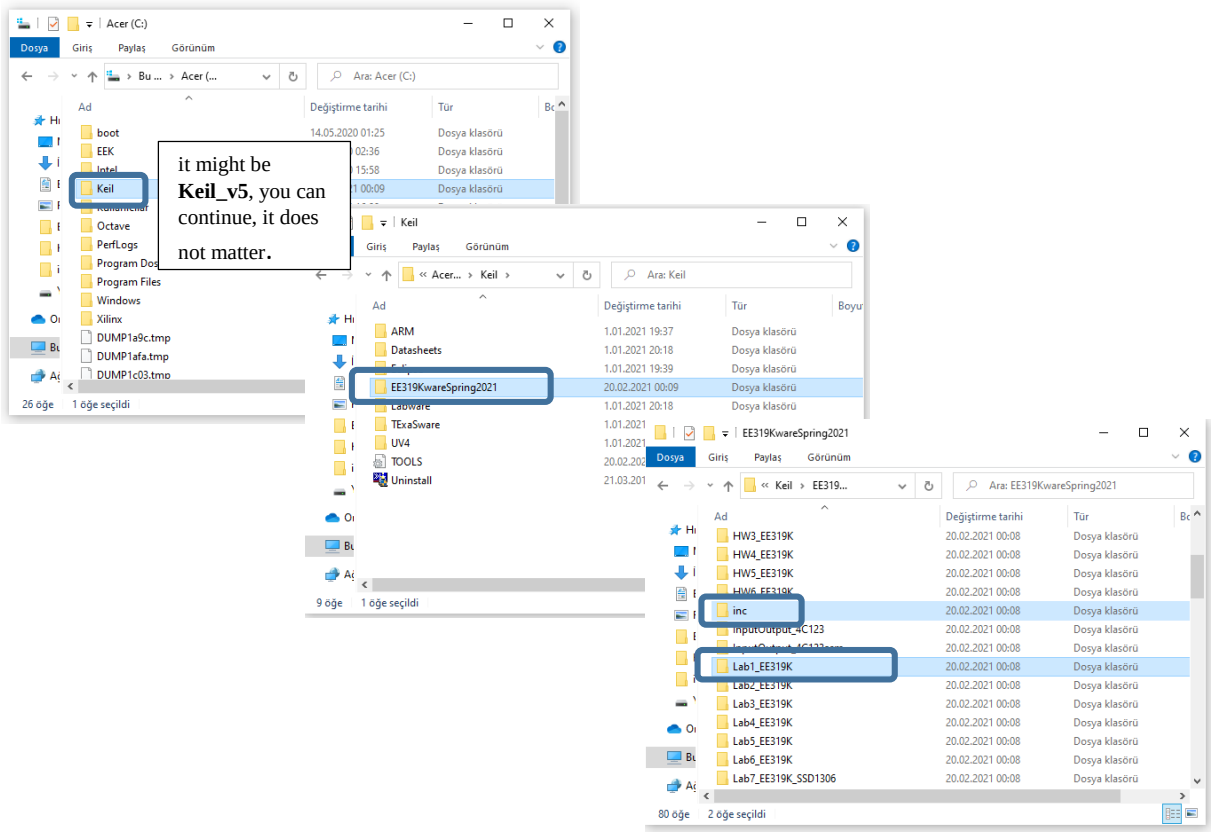

Figure 1: Directory structure with your Lab1

- It is important for the directory structure to look like Figure 1. Notice the directory relationship between the lab folders and the **inc** (include) folder. Begin with the **Lab1_EE319K** project in the folder **EE319KwareSpring2021**. Either double click the **uvprojx** file or open the project from within uVision. Make sure you can compile it and run on the simulator. **Startup.s** contains assembly source code to define the stack, reset vector, and interrupt vectors. **main.s** will contain your assembly source code for this lab. You will edit the **main.s** file.

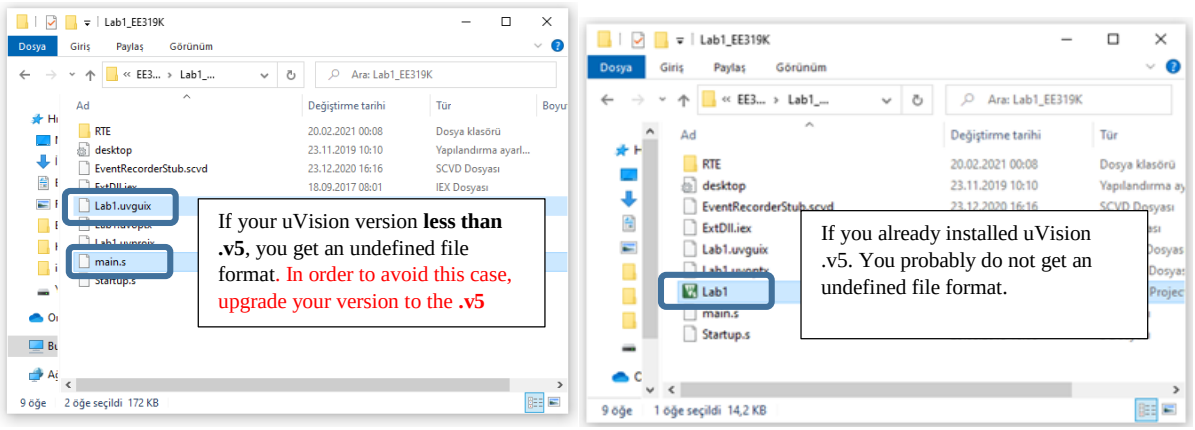

Figure 2: Start Keil by opening ***Lab1.uvguix***

- To run the Lab 1 simulator, you must check two things. First, execute Project->Options and select the Debug tab. The debug parameter field must include **-dEE319KLab1**. Second, the **EE319KLab1.dll** file must be present in your Keil\ARM\BIN folder (the EE319K DLLs should have been put there by the installer).

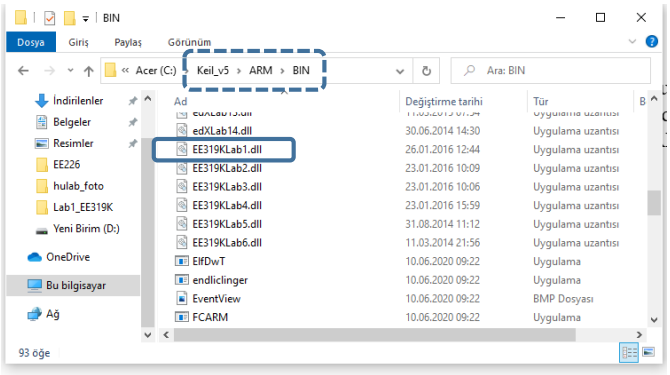

Figure 3: EE319KLab.dll checking

- You should **build** and **debug** your code consecutively.

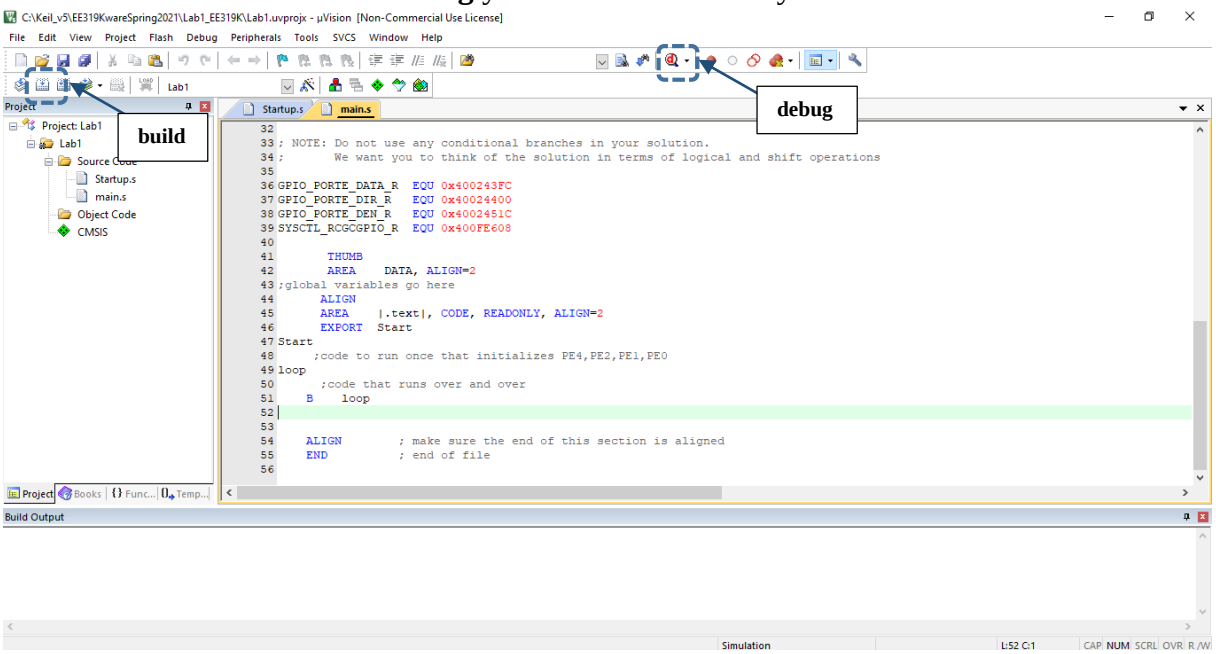

Figure 4: **Build** and **Debug** Icons

- After **debug**ging your code, if **TExaS Lab1** simulator is not open, you will click the following icons consecutively.

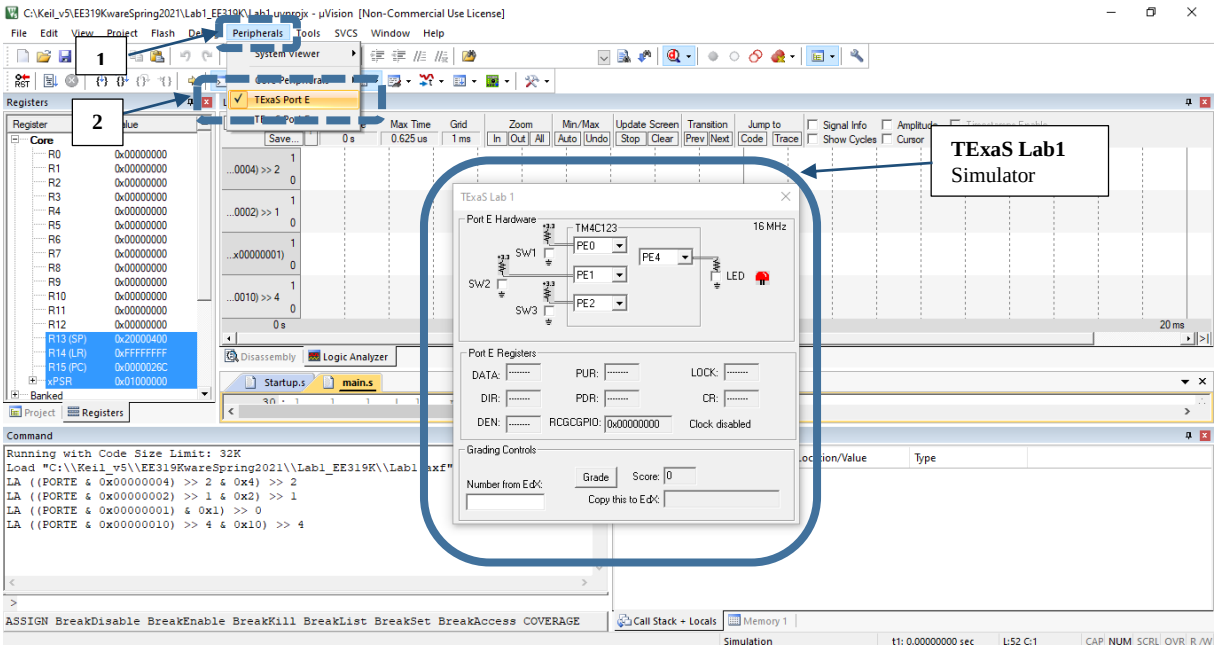

Figure 4: TExaS Lab1 Simulator and Keil Environment

- In order to start your code execution click the **run** icon.

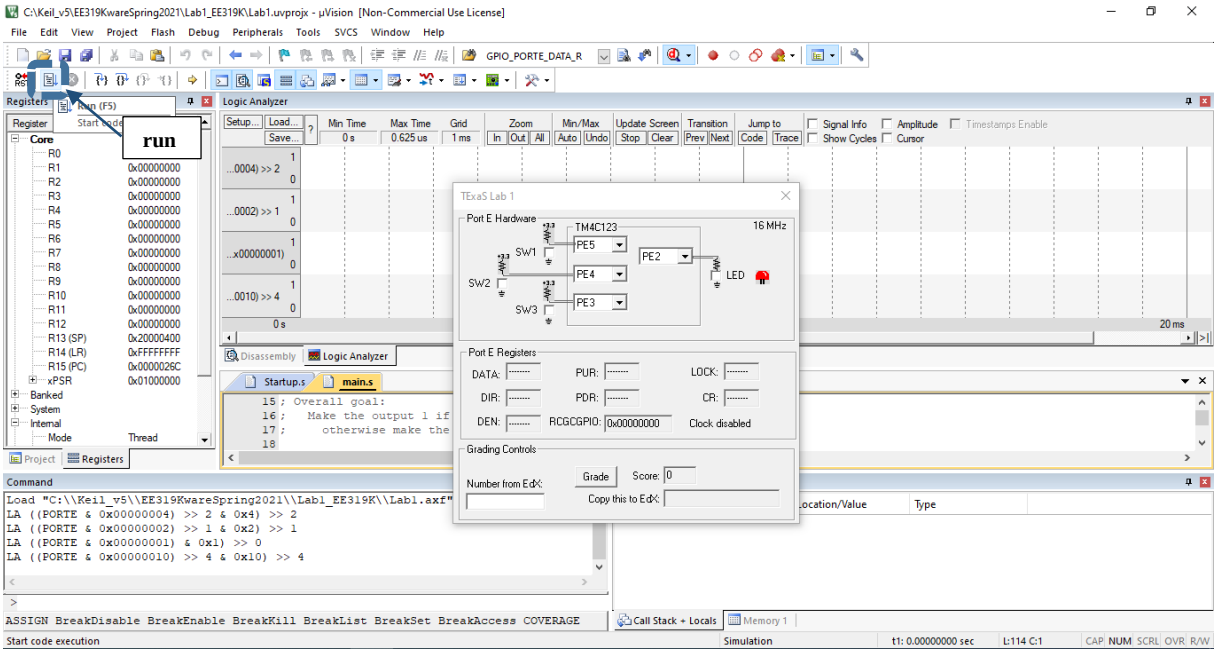

Figure 5: **Run** icon in Keil Environment

2. Draw Flowchart

Write a flowchart for this program. We expect 5 to 15 symbols in the flowchart. A flowchart describes the algorithm used to solve the problem and is a visual equivalent of pseudocode. You can define any problem which you are desired, then you should show how can you solve this problem by drawing flowchart.

3. Write Pseudocode

Write pseudocode for this program. We expect 5 to 10 steps in the pseudocode. You may use any syntax you wish, but the algorithm should be clear. Note, pseudocode ought to embody the algorithm and therefore be language blind. The same pseudocode can serve as an aid to writing the solution out in either assembly or C (or any other language).

4. Write Assembly

You will write assembly code that inputs from PE3, PE4, PE5 and outputs to PE2. The address definitions for Port E are listed below, and these are placed in the starter file main.s:

```
GPIO_PORTE_DATA_R EQU 0x400243FC
GPIO_PORTE_DIR_R EQU 0x40024400
GPIO_PORTE_DEN_R EQU 0x4002451C
SYSCTL_RCGCGPIO_R EQU 0x400FE608
```

To interact with the I/O during simulation, **make sure that View->Periodic Window Update is checked or the simulator will not update!** Then, execute the **Peripherals->TExaS Port E** command. When running the simulator, we check and uncheck bits in the I/O Port box to change input pins. We observe output pin in the window. You can also see the other registers, such as DIR DEN and RCGCGPIO.

- You should use specified even/odd parity checking according to your student ID number
  - If your student ID number is **even**, you should make **even** parity checking,
  - If your student ID number is **odd**, you should make **odd** parity checking.

References

[1] http://users.ece.utexas.edu/~valvano/Volume1/, Access Date: February 22, 2021.

Appendix

- When you are installing EE319K program, if you get a MS Smart Screen warning in the Figure A1, you will click **Additional info (Ek bilgi)**

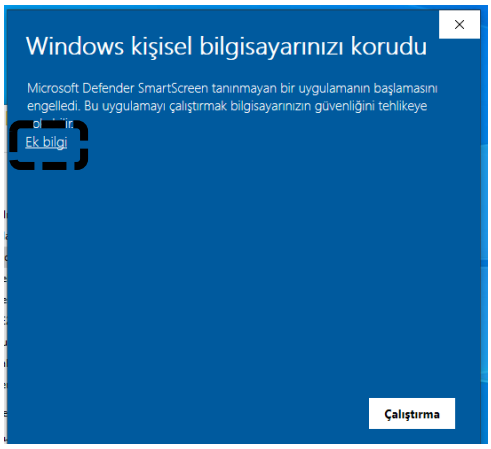

Figure A1: MS Smart Screen warning - 1

- Then, click **Still run (Yinede de Çalıştır)** as shown in Figure A2.

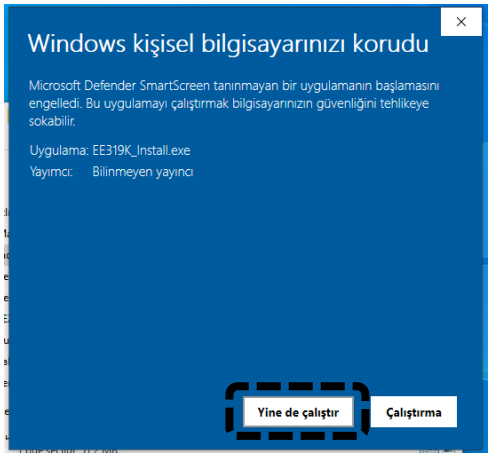

Figure A2: MS Smart Screen warning - 2