**NUST SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Lab Engineer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Semester/Section:\_\_\_\_\_\_\_\_\_\_\_\_ |

Department of Electrical Engineering

EE- 222: Microprocessor Systems

**LAB 10: Digital Logic Implemented on a Microcontroller**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student’s Name | Reg. # | Lab Conduct and Report | Viva | Total |
| 10 | 5 | 15 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Lab 10: Digital Logic Implemented on a Microcontroller

# Objective:

The general purpose of this laboratory is to familiarize you with the software development steps using the **uVision** simulator. You will learn how to perform digital input/output on parallel ports of the LM4F120/TM4C123. Software skills you will learn include port initialization, logic operations, and unconditional branching.

**Instruction Set TM4C123 files:**



**Introduction:**

6 General-Purpose I/O (GPIO) ports:

* Four 8-bit ports (A, B, C, D)
* One 6-bit port (E)
* One 5-bit port (F)

**I/O Ports and Control Registers:**

To initialize an I/O port for general use we perform seven steps. Steps two through four are needed only for the LM4F/TM4C microcontrollers.

1. First, we activate the clock for the port.
2. Second, we unlock the port; unlocking is needed only for pins PC3-0, PD7, PF0 on the LM4F and TM4C.
3. Third, we disable the analog function of the pin, because we will be using the pin for digital I/O.
4. Fourth, we clear bits in the **PCTL** to select regular digital function.
5. Fifth, we set its direction register.
6. Sixth, we clear bits in the alternate function register.
7. Lastly, we enable the digital port. We need to add a short delay between activating the clock and accessing the port registers.

The direction register specifies bit for bit whether the corresponding pins are input or output. A **DIR** bit of 0 means input and 1 means output.

**Sample Code:**

****

# Task Requirements:

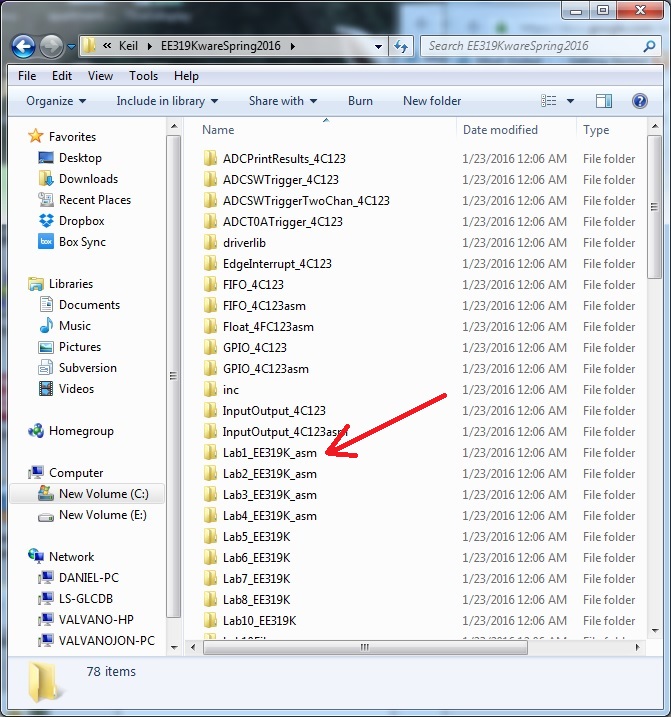
The specific device you will create is a digital lock with three binary switch inputs and one LED output. The LED output represents the lock, and the operator will toggle the switches in order to unlock the door. Let **PE2** (Port E bit 2) be the Boolean variable representing the lock (0 means LED is off and door is locked, 1 means LED is on and door is unlocked). Let **PE3** (Port E bit 3 is one input switch), **PE4** (Port E bit 4 is a second input switch) and **PE5** (Port E bit 5 is a third switch) be Boolean variables representing the state of the three switches (1 means the switch is not pressed, and 0 means the switch is pressed). The LED should come on if all three switches are pressed. The **PE2**=1 if and only if **PE5**=0 **PE4**=0 and **PE3**=0. The specific function you will implement is

**PE2** = (not(**PE3**)) and (not(**PE4**) and (not(**PE5**))

# Procedure:

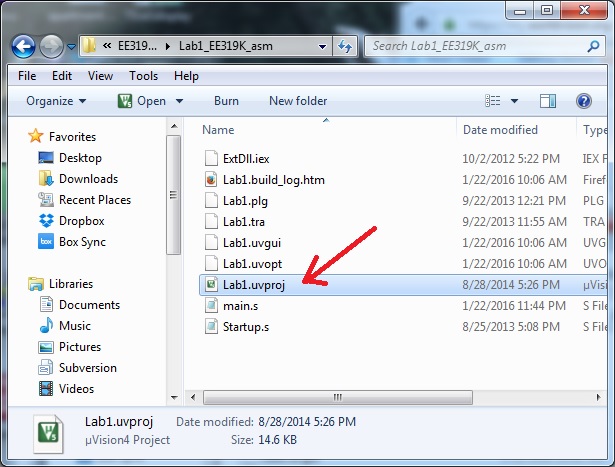
The basic approach to this lab will be to develop and debug your system using the simulator.

To create a Lab 1 Project, perform these tasks. Find a place on your hard drive to save all your LM4F120/TM4C123 software. In Figure 1.1 it is called **EE319KwareSpring2016**,



*Figure 1.1. Directory structure with your Lab1.*

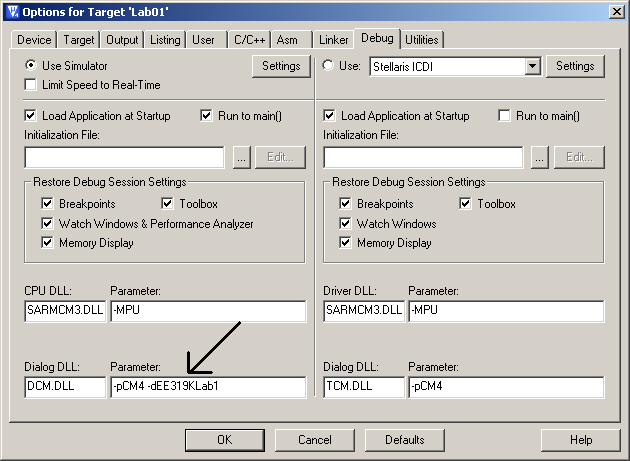
Copy the **Lab1\_EE319K\_asm** project on desktop. Notice the folder for each project is in **EE319KwareSpring2016**. Either double click the **uvproj** file or open the project from within uVision. Make sure you can compile it and run on the simulator. Please contact your TA if the starter project does not compile or run on the simulator. **Startup.s** contains assembly source code to define the stack, reset vector, and interrupt vectors. All projects in this class will include this file, and you should not need to make any changes to the **Startup.s** file. **main.s** will contain your assembly source code for this lab. You will edit the **main.s** file.



*Figure 1.2. Start Keil by opening the* ***Lab1.uvproj*** *file.*

Add your names and the most recent date to the comments at the top of main.s.

To run the Lab 1 simulator, you must do 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 added to your Keil\ARM\BIN folder. The debugging dynamic link libraries should go into Keil\ARM\BIN when you install the .exe.



*Figure 1.3. Debug the software using the simulator (DCM.DLL -pCM4 -dEE319KLab1).*

## Write Assembly

You will write assembly code that inputs from PE3, PE4, and PE5 and outputs to PE2. You can copy and paste the address definitions for Port E from the **lm4f120.s** or **tm4c123gh6pm.s** file

**THUMB**

**AREA DATA, ALIGN=2**

**;global variables go here**

**ALIGN**

**AREA |.text|, CODE, READONLY, ALIGN=2**

**EXPORT Start**

**Start**

**;initialization code goes here**

**loop**

**;the body of the code goes here**

**B loop**

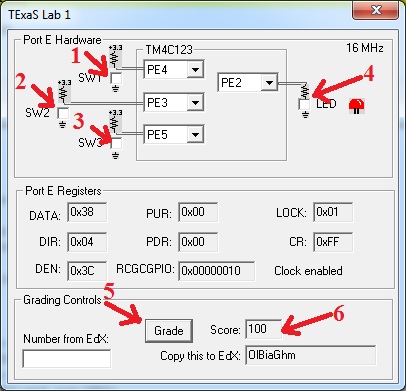
**ALIGN**

**END**

*Program 1.1. Assembly language template.*

The opening comments include: file name, overall objectives, hardware connections, specific functions, author name, TA, and date. The **equ** pseudo-op is used to define port addresses. Global variables are declared in RAM, and the main program is placed in EEPROM. The 32-bit contents at ROM address 0x00000004 define where the computer will begin execution after a power is turned on or after the reset button is pressed.

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 (1) (2) and (3). We observe output pin in the window (4). You can also see the other DIR AFSEL DEN and PUR registers. To have the simulator test your Lab 1 solution, click the **Grade** button (5). Your score will be shown in the **Score** box (6). The **Grading Controls** are for the MOOC and are not used for any part of your grade in EE319K this semester



*Figure 1.4. In simulation mode, we interact with virtual hardware.*

# Demonstration

During the demonstration, you will be asked to run your program to verify proper operation. You should be able to single step your program and explain what your program is doing and why. You need to know how to set and clear breakpoints. You should know how to visualize Port E input/output in the simulator.