Lab 1. Digital Logic Implemented on a Microcontroller

All students do Lab 1 by themselves (no partner for Lab 1)

- Preparation
- Purpose
- System Requirements
- Procedure
  - Part a - Create Keil Project
  - Part b - Draw Flow Chart
  - Part c - Write Pseudocode
  - Part d - Write Assembly
- Demonstration
- Deliverables
- FAQ

Preparation

Read Chapters 1 and 2 of the book
Read Sections 3.1, 3.2, 3.3.1-3.3.5, 3.3.9, 3.3.10, 4.1.2 and 4.2 of the book
Install and run Keil uVision on your personal computer
Download, unzip, open, compile, and run these four projects (either in lab or on your personal computer)

http://edx-org-utaustinx.s3.amazonaws.com/UT601x/EE319K_InstallSpring2016.exe Labs 1-4

SimpleProject_4C123asm.zip	Random number generator with no I/O
GPIO_4C123asm.zip	Simple output to Port D
Switch_4C123asm.zip	Simple input from Port A, LED on Port F
InputOutput_4C123asm.zip	More complex with inputs and outputs to Port F
Lab1_EE319K_asm	Starter project with links to grading engine
Lab1_EE319K_C	Starter project for HW3

Purpose

The general purpose of this laboratory is to familiarize you with the software development steps using the **uVision** simulator. Starting with Lab 2, we will use uVision for both simulation and debugging on the real board, but for this lab, we will use just the 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.

System 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

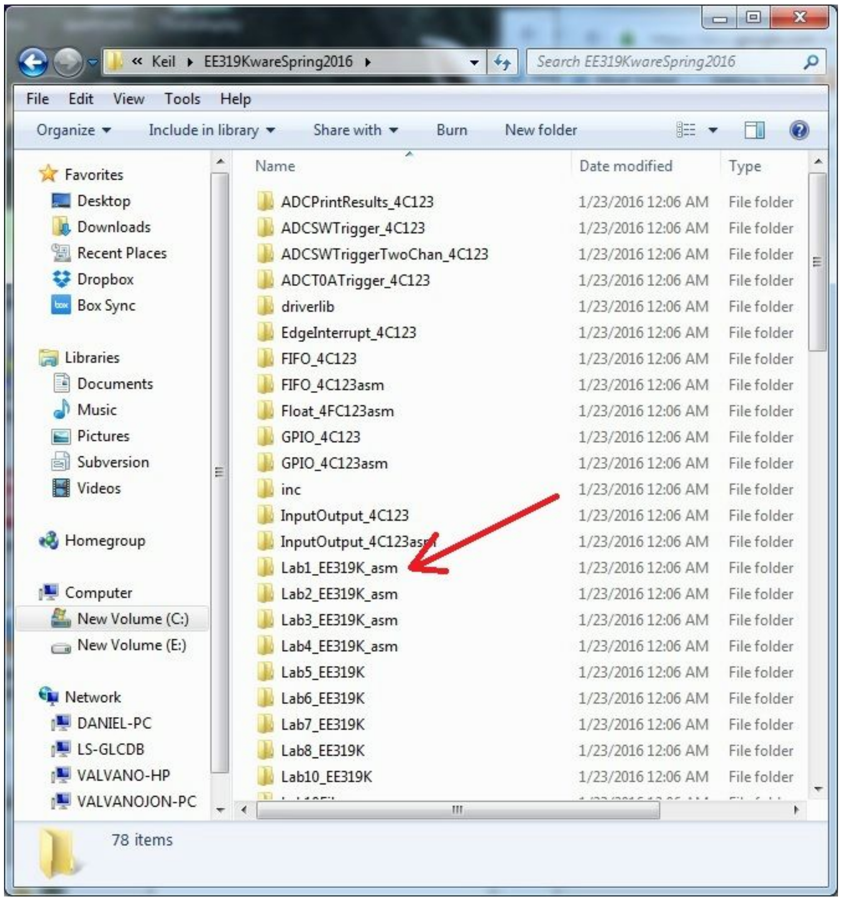
PE2 = (not(**PE3**)) and (not(**PE4**) and (not(**PE5**))

Procedure

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

Part a - Create Keil Project

To create a Lab 1 Project, perform these tasks. Find a place on your harddrive to save all your LM4F120/TM4C123 software. In Figure 1.1 it is called **EE319KwareSpring2016**, created when you install the .exe.

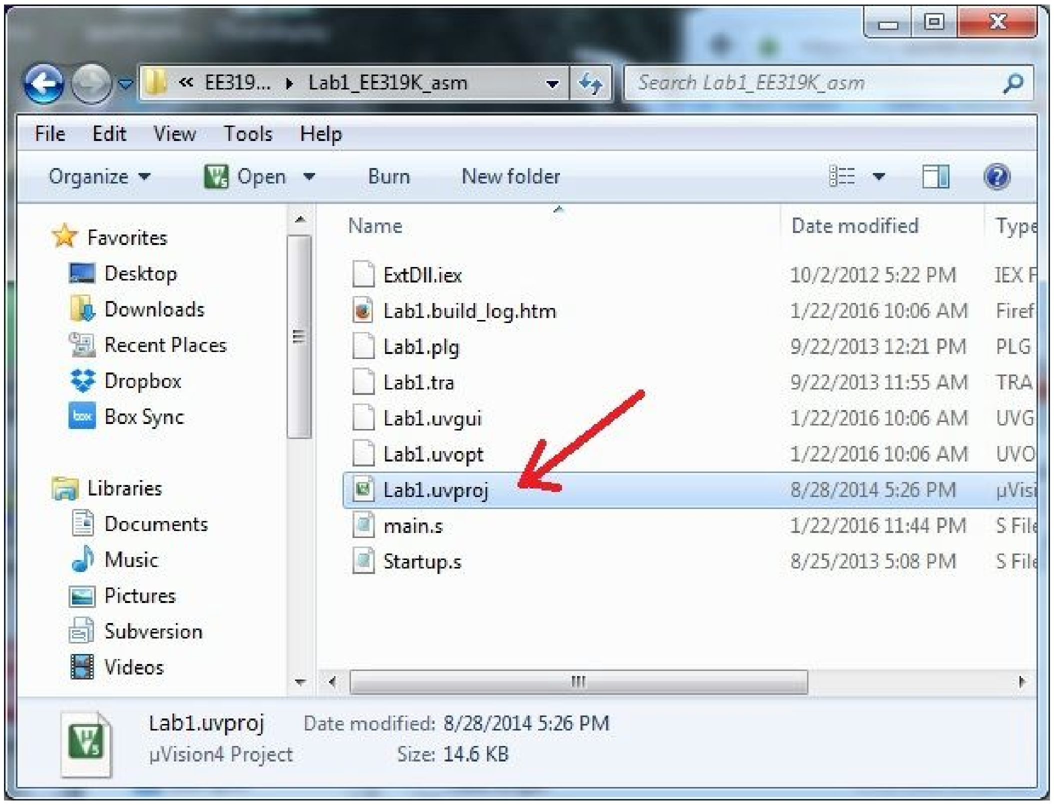


Name	Date modified	Type
ADCPrintResults_4C123	1/23/2016 12:06 AM	File folder
ADCSWTrigger_4C123	1/23/2016 12:06 AM	File folder
ADCSWTriggerTwoChan_4C123	1/23/2016 12:06 AM	File folder
ADCT0ATrigger_4C123	1/23/2016 12:06 AM	File folder
driverlib	1/23/2016 12:06 AM	File folder
EdgeInterrupt_4C123	1/23/2016 12:06 AM	File folder
FIFO_4C123	1/23/2016 12:06 AM	File folder
FIFO_4C123asm	1/23/2016 12:06 AM	File folder
Float_4FC123asm	1/23/2016 12:06 AM	File folder
GPIO_4C123	1/23/2016 12:06 AM	File folder
GPIO_4C123asm	1/23/2016 12:06 AM	File folder
inc	1/23/2016 12:06 AM	File folder
InputOutput_4C123	1/23/2016 12:06 AM	File folder
InputOutput_4C123asm	1/23/2016 12:06 AM	File folder
Lab1_EE319K_asm	1/23/2016 12:06 AM	File folder
Lab2_EE319K_asm	1/23/2016 12:06 AM	File folder
Lab3_EE319K_asm	1/23/2016 12:06 AM	File folder
Lab4_EE319K_asm	1/23/2016 12:06 AM	File folder
Lab5_EE319K	1/23/2016 12:06 AM	File folder
Lab6_EE319K	1/23/2016 12:06 AM	File folder
Lab7_EE319K	1/23/2016 12:06 AM	File folder
Lab8_EE319K	1/23/2016 12:06 AM	File folder
Lab10_EE319K	1/23/2016 12:06 AM	File folder

Figure 1.1. Directory structure with your Lab1.

It is important for the directory structure to look like Figure 1.1. Download and unzip the starter configuration from http://edx-org-utaustinx.s3.amazonaws.com/UT601x/EE319K_InstallSpring2016.exe into this location. We will place our projects in folders like my Lab 1.

Find the four example projects listed above. Begin with the **Lab1_EE319K_asm** project. 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.



Name	Date modified	Type
ExtDll.iex	10/2/2012 5:22 PM	IEX F
Lab1.build_log.htm	1/22/2016 10:06 AM	Firef
Lab1.plg	9/22/2013 12:21 PM	PLG
Lab1.tra	9/22/2013 11:55 AM	TRA
Lab1.uvgui	1/22/2016 10:06 AM	UVG
Lab1.uvopt	1/22/2016 10:06 AM	UVO
Lab1.uvproj	8/28/2014 5:26 PM	µVisi
main.s	1/22/2016 11:44 PM	S File
Startup.s	8/25/2013 5:08 PM	S File

Lab1.uvproj Date modified: 8/28/2014 5:26 PM
µVision4 Project Size: 14.6 KB

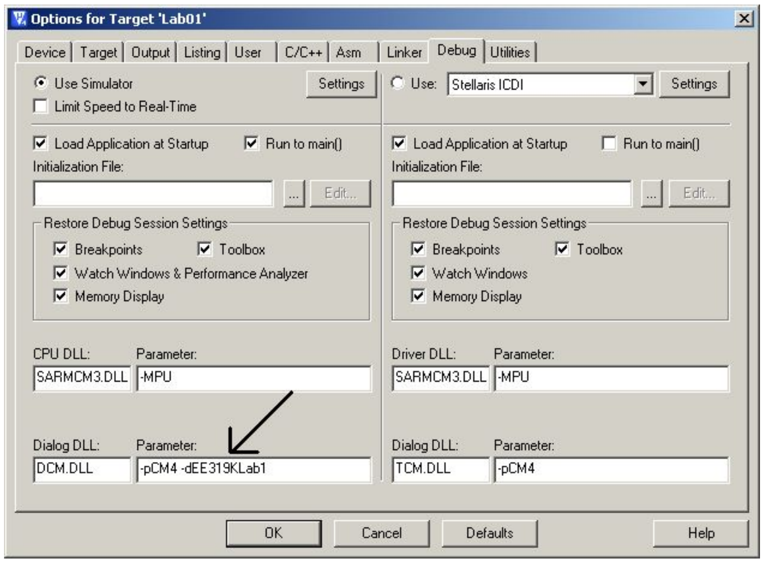
Figure 1.2. Start Keil by opening the ***Lab1.uvproj*** *file.*

You should rename the Lab1 starter folder to include your EIDs. Add your names and the most recent date to the comments at the top of main.s. This code shows the basic template for the first few labs. You will not need global variables in this lab.

```
       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.

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.



(•) Use Simulator
[ ] Limit Speed to Real-Time
[✓] Load Application at Startup [✓] Run to main()
Initialization File:
Restore Debug Session Settings
[✓] Breakpoints [✓] Toolbox
[✓] Watch Windows & Performance Analyzer
[✓] Memory Display
CPU DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: DCM.DLL Parameter: -pCM4 -dEE319KLab1

( ) Use: Stellaris ICDI
[✓] Load Application at Startup [ ] Run to main()
Initialization File:
Restore Debug Session Settings
[✓] Breakpoints [✓] Toolbox
[✓] Watch Windows
[✓] Memory Display
Driver DLL: SARMCM3.DLL Parameter: -MPU
Dialog DLL: TCM.DLL Parameter: -pCM4

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

Part b - Draw Flow Chart

Write a flowchart for this program. We expect 5 to 15 symbols in the flow chart.

Part c - Write Pseudocode

Write pseudocode for this program. We expect 5 to 10 steps in the pseudo code. You may use any syntax you wish, but the algorithm should be clear.

Part d - Write Assembly

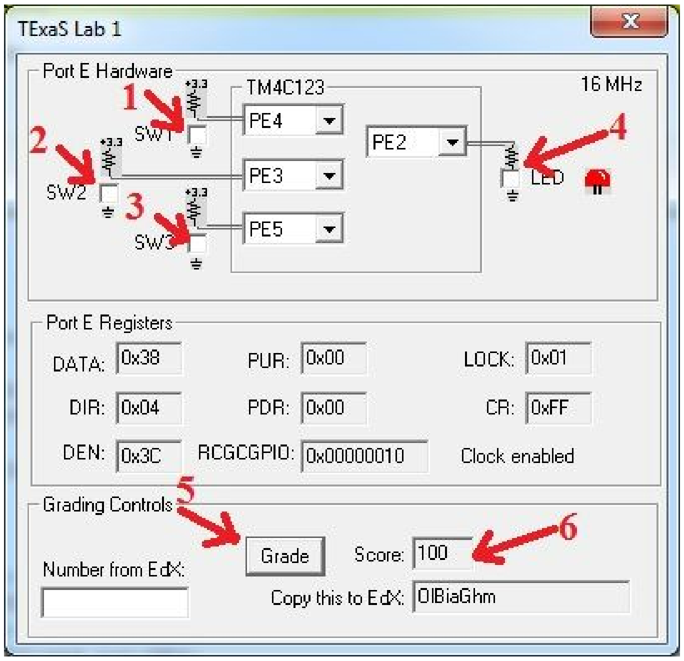
You will write assembly code that inputs from PE3, PE4, and PE5 and outputs to PE2. You can copy and paste the address definitions for Port E from the **lm4f120.s** or **tm4c123gh6pm.s** file (listed below).

```
GPIO_PORTE_DATA_R        EQU    0x400243FC
GPIO_PORTE_DIR_R         EQU    0x40024400
GPIO_PORTE_AFSEL_R       EQU    0x40024420
GPIO_PORTE_DEN_R         EQU    0x4002451C
GPIO_PORTE_AMSEL_R       EQU    0x40024528
GPIO_PORTE_PCTL_R        EQU    0x4002452C
SYSCTL_RCGC2_R           EQU    0x400FE108
```

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.

```
;****************** main.s ***************
; Program written by: ***Your Name**update this***
; Date Created: 1/22/2016
; Last Modified: 1/22/2016
; Section ***Tuesday 1-2***update this***
; Instructor: ***Ramesh Yerraballi**update this***
; Lab number: 1
; Brief description of the program
; The overall objective of this system is a digital lock
; Hardware connections
;  PE3 is switch input  (1 means switch not pressed, 0 means switch pressed)
;  PE4 is switch input  (1 means switch not pressed, 0 means switch pressed)
;  PE5 is switch input  (1 means switch not pressed, 0 means switch pressed)
;  PE2 is LED output (0 means door is locked, 1 means door is unlocked)
; The specific operation of this system is to
;   unlock if all three switches are pressed
```
Program 1.2. Required comments at the top of every file.

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



DATA:	0x38	PUR:	0x00	LOCK:	0x01
DIR:	0x04	PDR:	0x00	CR:	0xFF
DEN:	0x3C	RCGCGPIO:	0x00000010	Clock enabled	

Score: 100
Copy this to EdX: OlBiaGhm

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.

Deliverables

Items 1-4 are one pdf file uploaded to Canvas, have this file open during demo.

0. Lab 1 grading sheet. You can print it yourself or pick up a copy in lab. You fill out the information at the top.
1. Flowchart of the system
2. Pseudo code for the algorithm
3. Assembly source code of your final **main.s** program
4. One screenshot of the Port E window, showing PE2, PE3, PE4, and PE5 showing the case with the LED on.
5. Optional Feedback : http://goo.gl/forms/rBsP9NTxSy

FAQ

The list of FAQ below are populated from Piazza over the semesters (thanks to the contributions of all past TAs and students). More questions may be posted so please check back regularly.

1. Once the "lock" is "unlocked," should the program stop checking and remain unlocked or should the user be able to "re-lock" it by un-pressing the switches?
   Your program should loop, so yes. After the switches are no longer in "un-lock" configuration, the LED should turn off, entering into the "locked" configuration.
2. Our program works as expected when stepping through but when it is run through it does not. What could be causing this? What is an effective way to debug when our debugging method says that the program is working fine but the actual running of the program says otherwise?
   First check if the “Periodic Window Update” under the “View" tab is on when you are in debugging mode. Also, some run-time errors can occur when setting up the clock register which don't appear when single stepping. Look over and ensure you are writing to the correct register and only affecting those bits required to activate Port E, as well as give enough time for it to start up.
3. What is the best way to include the source code for the main.s into the PDF?
   One way to include the source code for the main.s into the PDF is to copy / paste the code into a Word document and then combine that with all of your other deliverables, depending on the lab, and then converting that file to a PDF. Please do not only include a screenshot.
4. For the flowchart and pseudocode, do we need to start at the very beginning with all the initialization tasks, or just at the actual logic for locking and unlocking the lock?
   Your flowchart should cover the entire program that you write. Therefore, initialization should be included. How you represent initialization is up to you. See Part b of Lab 2 as an example.
5. Are we allowed to branch?
   Only unconditional branches are allowed in this lab. You are required to implement this lab using only Boolean logic such as AND, OR, and shifts etc.
6. Do both members of our lab group need to turn in a pdf, or do we just turn in one for the two of us?
   Please both submit you pdf deliverables on Canvas.

7. When I try to run the debugger, I get: Error: Could not load file 'C:\Program Files (x86)\Keil\EE319KwareSpring2016\Lab1_EE319K\Lab1.axf'. Debugger aborted! What should I do?
   You most likely forgot to build your project before running the debugger. If that doesn't solve the problem, try running Keil as an administrator.
8. When I try to build, it gives the errors: Build target 'Lab1', error - cannot create command input file '.\startup._ia', error - cannot create command input file '.\main._ia', Target not created! What should I do?
   Try running Keil as an administrator
9. Where can I find the addresses for the different ports?
   Register definitions for the microcontroller may be found here:
   http://users.ece.utexas.edu/~valvano/arm/tm4c123gh6pm.s
10. I edited my code but nothing changes when I re-run it in the debugger!
    If you made changes to your code "in debug mode", you most likely have not re-built your project and therefore your debugger is running the old version of your code. Exit out of debug mode and rebuild your code for the changes to take effect.
11. The grader that comes with the simulator says I got x grade for the lab. Am I good to go/ in trouble?
    The grader is not used officially in this course at all. In fact we encourage you to debug your program on your own rather than relying on the output messages from the grader. Your grade for the lab will be determined by the TA's during checkout
12. I cannot find the interactive UI simulator with switches and LEDs when I am in debugging mode!
    Enable "TExaS Port x" under the "Peripherals" tab. If you do not see the Ports that you are expecting, you most likely do not have the correct DLL file listed in "Debug" tab in "Options for Target" under "Project" tab.
13. Keil tells me I have tons of build errors but nothing seems to be wrong with my assembly code.
    Make sure you instructions are indented. Only the labels are aligned all the way to the left.