Lab 4: Interfacing Keypad

Instructor: Dr. Yanmin Gongea

Teaching Assistants: Francisco E Fernandes Jr and Khuong V. Nguyen

Spring 2019

Goals

1. Understand I/O matrix technique
2. Be familiar with keypad scanning algorithms

Grading Rubrics (Total = 100 points)

1. **Pre-lab assignment:** 10 points.
2. **Attendance and Class Participation:** 8 points.
3. **Code Organization:** 8 points.
4. **Lab demo questions:** 10 points.
5. **First Objective:** 50 points.
6. **Second Objective:** 14 points.

Pre-lab Assignment

1. Read Textbook Chapter 14.9 Keypad Scan.
2. Complete the register tables listed in the pre-lab (**10 points**).
    a. **Due date for Monday Labs:** April 22, 2019.
    b. **Due date for Wednesday Labs:** April 24, 2019.

Lab Objectives – Overview

1. **First Objective (50 points):**
    a. **Due date:**
        i. **For Monday labs:** April 29, 2019.
        ii. **For Wednesday labs:** May 01, 2019.
    b. Write a C program to read which key is being pressed in the keypad and display it in the LCD (the code for the LCD is given to you). For the first objective, you only need to display the numerical digits. You must use polling method to scan the keypad.

2. **Second Objective (14 points):**
    a. **Due date:**
        i. **For Monday labs:** April 29, 2019.
        ii. **For Wednesday labs:** May 01, 2019.
    b. You <u>MUST CHOOSE TWO</u> of the following options:
        i. When a key is pressed for a long time, generate a periodical input with an interval of 2 seconds.
        ii. Use the "*" key to delete the previous input. Pressing "*" key again keeps deleting the previous input.
        iii. Use the "#" key to repeat the previous inputs.
        iv. Detect and recognize if multiple keys are pressed simultaneously.

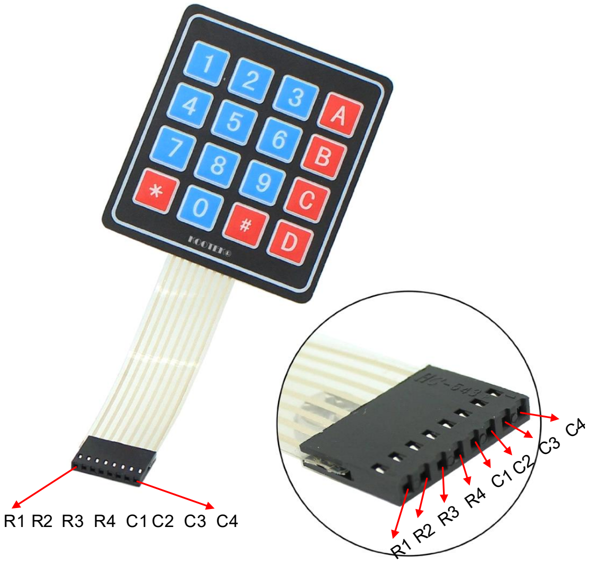
Keyboard Interface

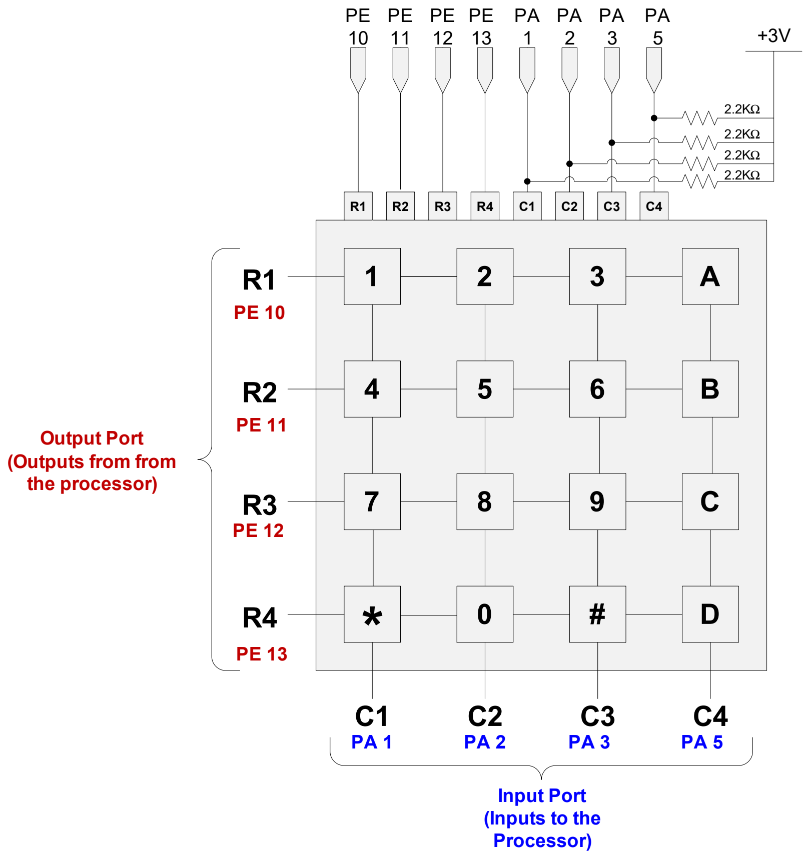
The 4x4 keypad used in this lab requires 8 pins (four row pins and four column pins). In this lab, the connection between the keypad and the discovery kit is as the following table.

Row	R1 → PE 10	R2 → PE 11	R3 → PE 12	R4 → PE 13
Column	C1 → PA 1	C2 → PA 2	C3 → PA 3	C4 → PA 5

All pins of the input port (C1, C2, C3, and C4) are pulled up to 3V via a 2.2KΩ resistor. Within the processor, each GPIO pin can be pulled up via an internal resistor (between 20 and 55 KΩ, typically 40 KΩ). However, the internal pull-up capability is too weak and thus an external pull-up is required.

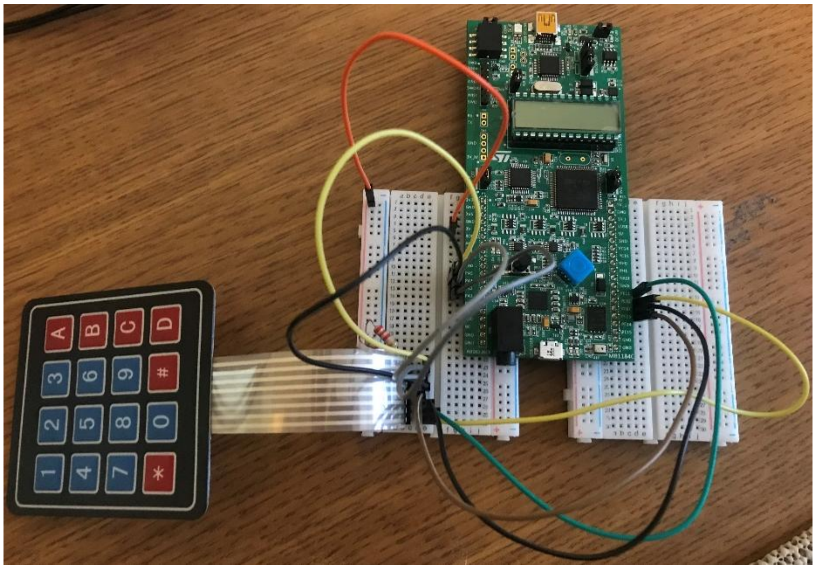
When looking at the front size of the keypad, the pins on the back from left to right are:
`R1 – R2 – R3 – R4 – C1 – C2 – C3 – C4.`

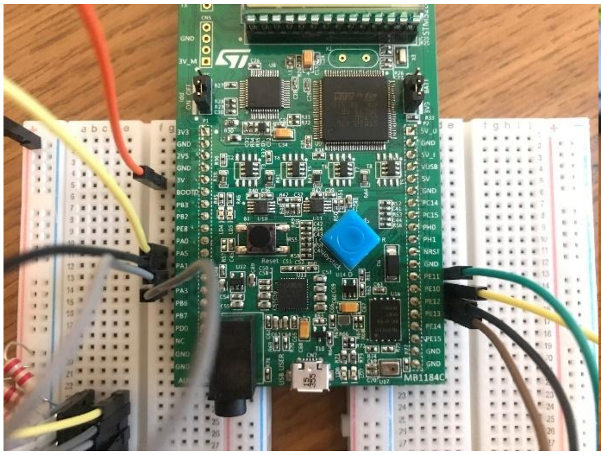


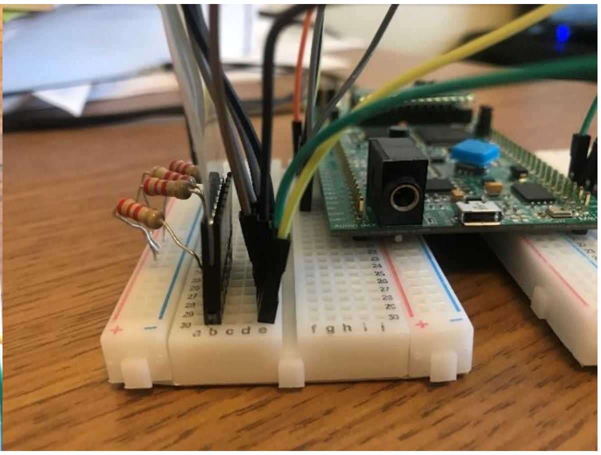


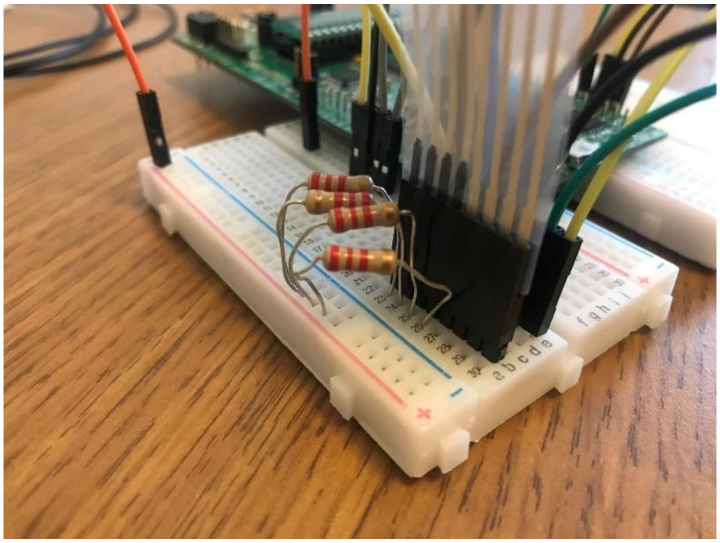
The maximum current a GPIO pin can source or sink is 20 mA. When calculating the value of external pull-up resisters, make sure that the current should not exceed 20 mA.

$$\frac{3V}{2.2K\Omega} = 1.4\ mA$$

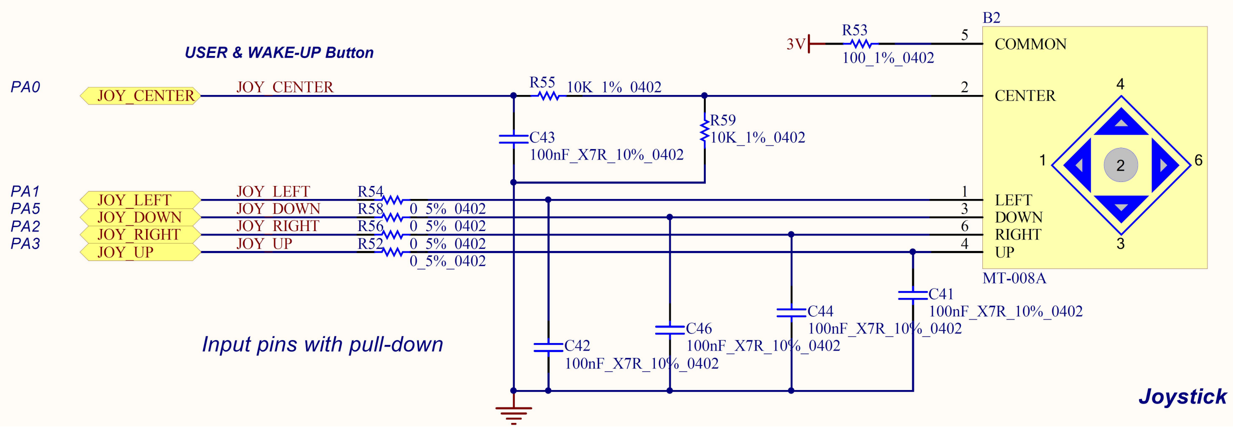








On the STM32L4 board, all pins in the input port (PA1, PA2, PA3, and PA5) are connected to ground via a 100nF capacitor, as shown in the figure below. Therefore, due to these capacitors, the voltage output on these pins won't immediately changes to Vcc or ground. ***A very short delay should be added before reading the input port.*** Specifically, delays are needed between setting GPIOE outputs and reading GPIOA inputs.



Lab 4: Startup Code

- A startup code is provided on D2L under Lab 4 section (filename: ***Lab 4 – Startup Code.zip***) containing the following files: **main.c**, **LCD.c**, **keypad.c, LCD.h, keypad.h** and **stm32l476xx.h**.
  - **Download** and **extract** the startup code.
  - Create a new C Project using System Workbench for STM32 IDE.
  - **Delete** the **Utilities** folder that is automatically created by the IDE.
  - **Move all *.c* files** to your project's **src** folder.
  - **Move all *.h* files** to your project's **inc** folder.

  - To complete the first objective (50 points), you only have to write code in the ***keypad.c*** file.
  - More specifically, you should complete two methods:
    - ***Keypad_Init()***:
      - This is based on this and previous pre-labs (complete the missing masks).
    - ***Keypad_Scan()***:
      - You should complete this method by following the scanning algorithm found on **Figure 14-26** in the textbook.
      - This method is mostly empty. You should figure out by yourselves the correct code to be written. To help you, try follow the comments located in the **keypad.c** file.

  - To complete the second objective (14 points), you will have to figure it out by yourself, but you will have to also modify the **main.c** file.

  - For this lab all the functions related to displaying digits in the LCD is given to you (**LCD.c**) you do not need to write any code related to the LCD.