Lab 1: Interfacing Joystick and LEDs

Graduate Teaching Assistants:

Francisco E. Fernandes Jr.
feferna@okstate.edu

Khuong Vinh Nguyen
Khuong.V.Nguyen@okstate.edu

School of Electrical and Computer Engineering
Oklahoma State University

Spring 2019

Lab 1 Goals

- Get familiar with the **System Workbench** software development environment
- Create a **C** project for **STM32L4 discovery kit** and program the kit
- **Learn basics of GPIO input and output configuration:** input/output, push pull, open-drain, pull up/down, GPIO speeds
- Program GPIO registers to perform simple digital I/O input (interfacing the joystick) and output (interfacing LED)

Lab 1 Schedule

- **Pre-lab Assignment** (10 points):
  - **For Monday labs:** Due on Feb. 11, 2019.
  - **For Wednesday labs:** Due on Feb. 13, 2019.
- **Lab 1** will take a total of **three weeks:**
  - **In each week, you will have to complete some assignment!**
  - **Week 1:**
    - Complete and show to a T.A. the **code to initialize the GPIO** clocks and pins (10 points).
  - **Week 2:**
    - Complete and demo to a T.A. the **primary objective** of this lab (40 points).
  - **Week 3:**
    - Complete and demo to a T.A. the **secondary objective** of this lab (14 points).

Grading Policy for Lab 1

<table>
<tr><th colspan="2">Description</th><th>Points</th><th>Due date for Monday labs</th><th>Due data for Wednesday labs</th></tr>
<tr><td colspan="2">Pre-lab assignment</td><td>10 points</td><td>Feb. 11</td><td>Feb. 13</td></tr>
<tr><td colspan="2">Attendance and Class Participation</td><td>8 points</td><td>Feb. 11, 18, 25</td><td>Feb. 13, 20, 27</td></tr>
<tr><td colspan="2">Code organization</td><td>8 points</td><td>N/A</td><td>N/A</td></tr>
<tr><td colspan="2">Lab demo questions</td><td>10 points</td><td>Feb. 25</td><td>Feb. 27</td></tr>
<tr><td rowspan="2"><b>Primary objective</b></td><td>GPIO initialization</td><td><b>10 points</b></td><td><b>Feb. 11</b></td><td><b>Feb. 13</b></td></tr>
<tr><td>Lab logic</td><td><b>40 points</b></td><td><b>Feb. 18</b></td><td><b>Feb. 20</b></td></tr>
<tr><td colspan="2"><b>Secondary objective</b></td><td><b>14 points</b></td><td><b>Feb. 25</b></td><td><b>Feb. 27</b></td></tr>
<tr><td colspan="2"><b>Total:</b></td><td><b>100 points</b></td><td></td><td></td></tr>
</table>

Grading Policy for Lab 1

- Code organization:

```
#include <stdio.h>

int main() {
  printf("Hello World!");
  return 0;
}
```

Good organization!

✓

```
#include <stdio.h>




int main() {  printf("Hello World!");
              return 0;}
```

Bad organization!

✗

Lab 1 Objectives

- **Primary Objective:**
  - **Write a C program that uses the onboard joystick to control both the red and green LEDs as follows:**
    - Toggle red LED when the right button is pushed;
    - Toggle green LED when the left button is pushed;
    - Set both LEDs to on when the up button is pushed;
    - Set both LEDs to off when the down button is pushed.

Lab 1 Objectives

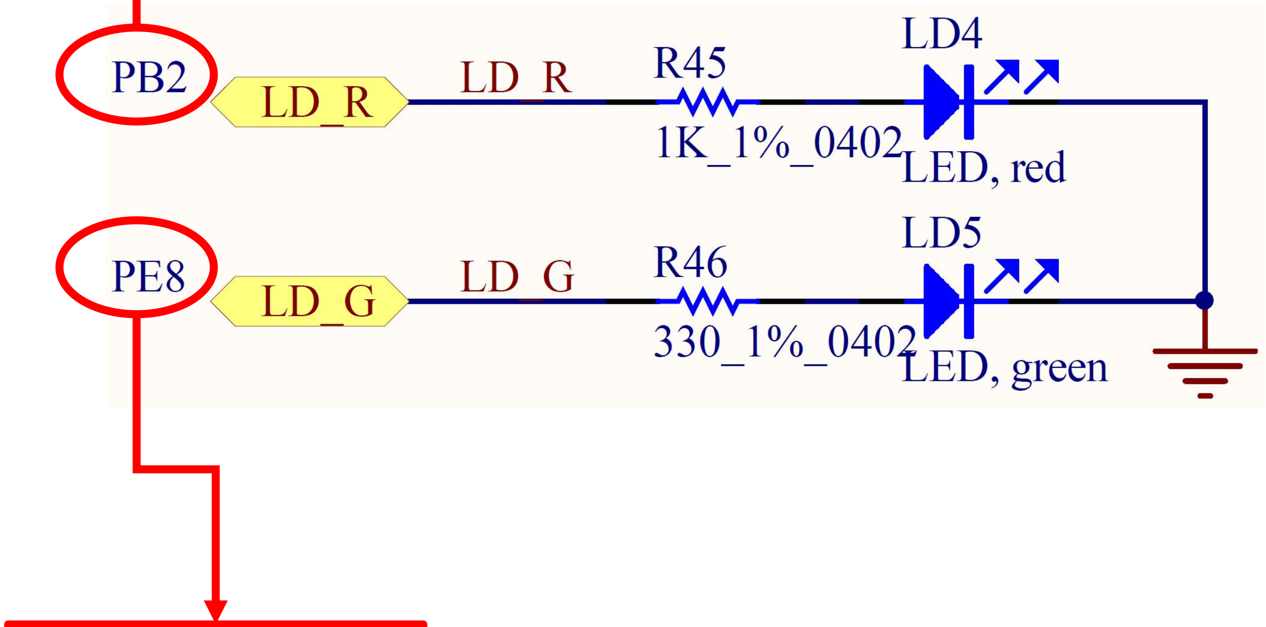
- **Secondary Objective:**
  - You must write a program to implement only **ONE** of the following options (choose the one it is the easiest for you):
    - Write a **C program** to make the **RED LED** send out SOS in Morse code (· · · – – – · · ·) if the **joystick's middle button** is pressed. DOT is on for ¼ second and DASH is on for ½ second, with ¼ second between these light-ons.

OR

-
  -
    - Write an **Assembly program** that re-implements the primary objective.

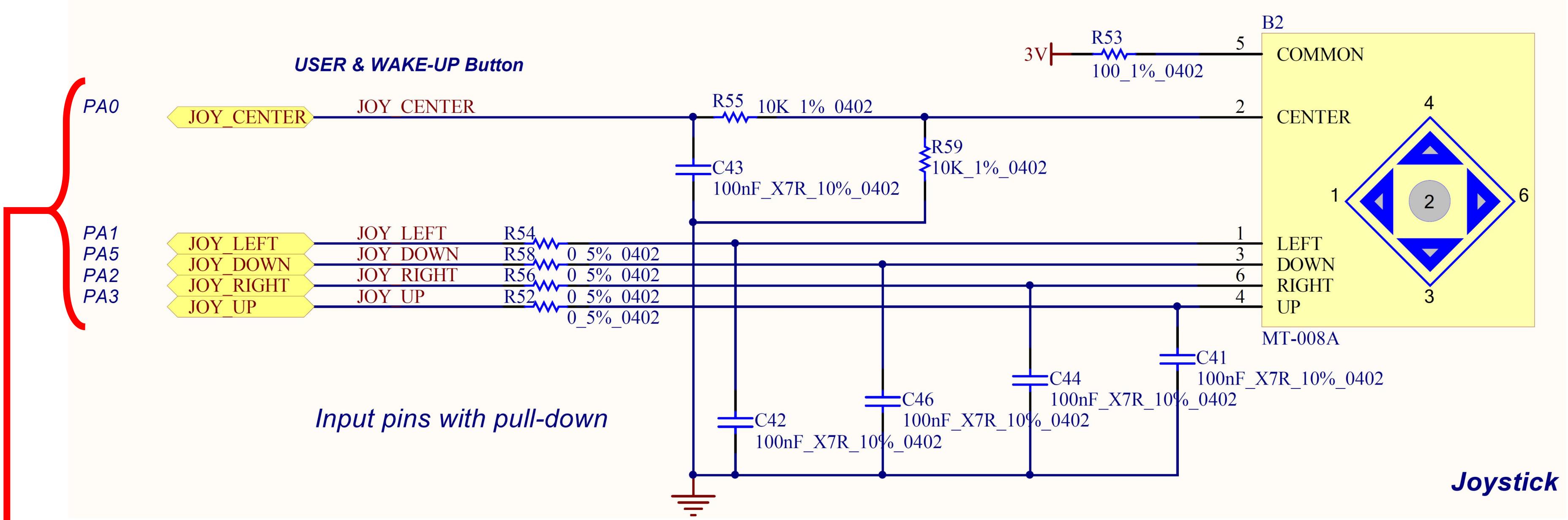
LEDs on the board

GPIO port B, pin 2!



GPIO port E, pin 8!

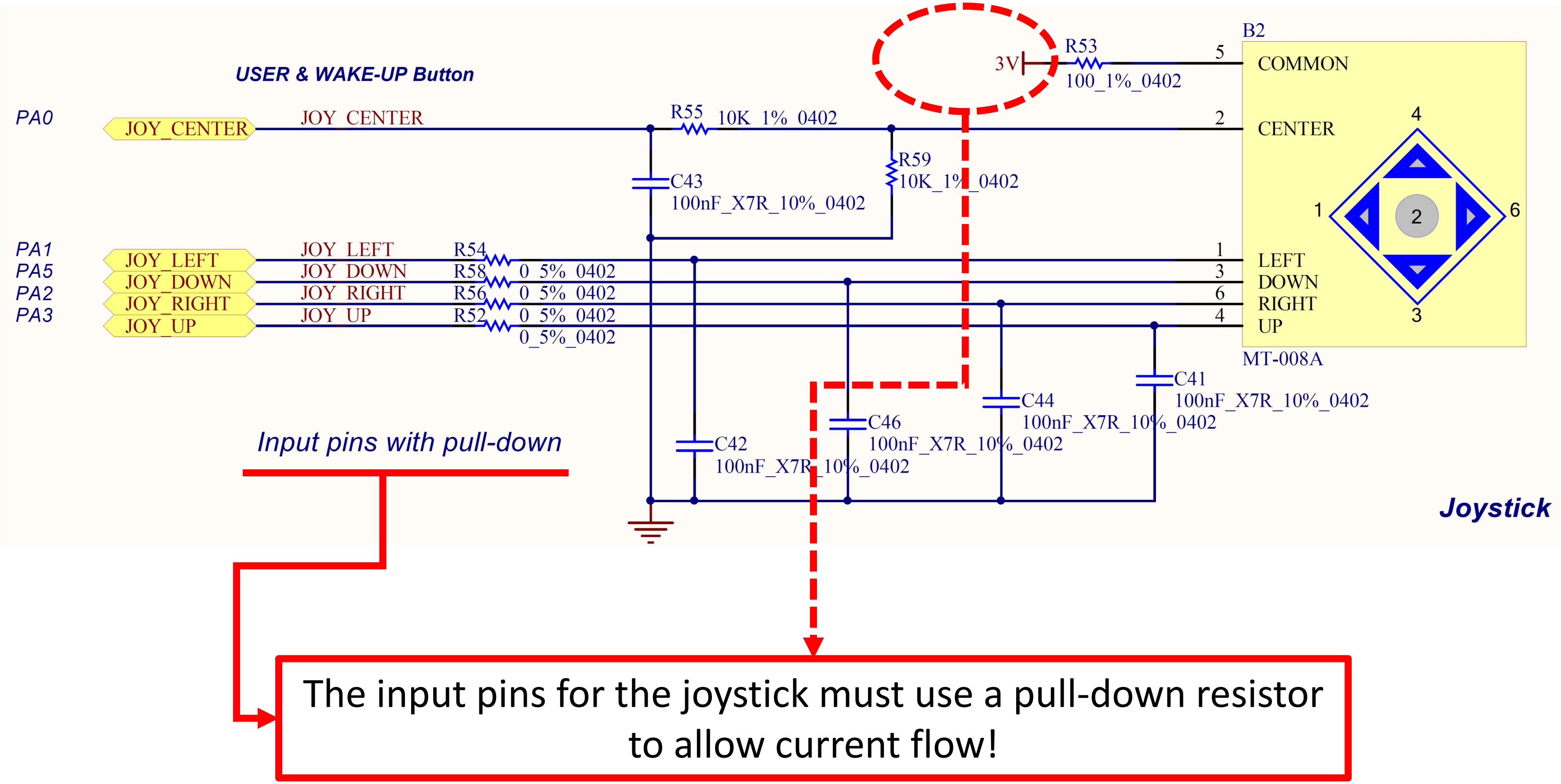
Joystick on the board



The joystick is connected as follow:

- **Center:** PA0 (GPIO port A, pin 0)
- **Left:** PA1 (GPIO port A, pin 1)
- **Right:** PA2 (GPIO port A, pin 2)
- **Up:** PA3 (GPIO port A, pin 3)
- **Down:** PA5 (GPIO port A, pin 5)

Joystick on the board



The input pins for the joystick must use a pull-down resistor to allow current flow!

Lab 1: step-by-step

1. **Enable the clock** to GPIO port A, B and E.
2. Configure PB2 (blue LED) and PE8 (green LED) as **output**
3. Configure PB2 and PE8 as **push-pull mode.**
4. Configure PB2 and PE8 output type as **No Pull-up No Pull-down.**
5. Configure PA0, PA1, PA2, PA3 and PA5 as **input.**
6. Configure PA0, PA1, PA2, PA3 and PA5 as **Pull-down.**
7. **Wait and verify** if any joystick position is pressed.

Lab 1: step-by-step

- To help you, a start-up code is available on D2L. Use it to create your project.
- The start-up code contains some helpful comments. **Read them!**
- All your code should be written inside the **main()** function located in the **main.c** file.

For next class

- Complete and demo to a T.A. the **primary objective** of this lab.