ELE201 H25 Assignment – 2 (Microcontrollers part)

This is a document for the second assignment in ELE201 – Microcontrollers and Data Network for the **microcontrollers part.** You must also deliver the data network related part of your assignment for the whole assignment to be approved.

In this assignment, you will answer two questions. In each question, you will set up a given circuit in the figure and write a code as the question requires.

If the question requires you to use a specific pin on your STM32F767 board, please use it. Otherwise, you are expected to find a suitable pin using datasheet/CubeMx for the task.

Please be explicit when you are writing your code so that the other people who read your code should understand what you are doing. Adding a README file to your assignment and/or writing a header comment where you state your pins and their roles is **highly encouraged.**

When you deliver your assignment on Canvas, please attach everything that is relevant i.e. a picture of your circuit (or a working video), your code, your CubeMx file etc. Each of you might have a different approach. Therefore, you can make the judgement of what to include in your attachment to show the best of your answer. If you attach the picture/video of your circuit which is not easy to trace, then you may attach a diagram – there you can reuse the figures in from the assignment itself.

You can work in groups with **max 2 people** in each group. Remember that **each group member should upload the solutions to Canvas**.

Good luck!

1. In this exercise, you will design a simple traffic light system with 3 LEDs,1 push button and 4 resistors as shown in Figure-1. Your system should work according to these principles:

   PART 1

   a. As the system starts, only LED_RED should be on.
   b. The LED_RED should stay on for 20 seconds.
   c. After, LED_RED and LED_YELLOW should stay on for 5 seconds.
   d. Finally, only LED_GREEN should switch on, and stay on for 10 seconds.
   e. Repeat the process infinitely.

   PART 2

   a. Your push button represents a pedestrian button by the road. When the button is pressed,
      i. If the LED_GREEN is on, the LED_YELLOW should turn on *together with* the LED_GREEN for 5 seconds, and then only LED_RED should turn on.
      ii. If the LED_RED is on, then add 10 seconds extra to its on-time.

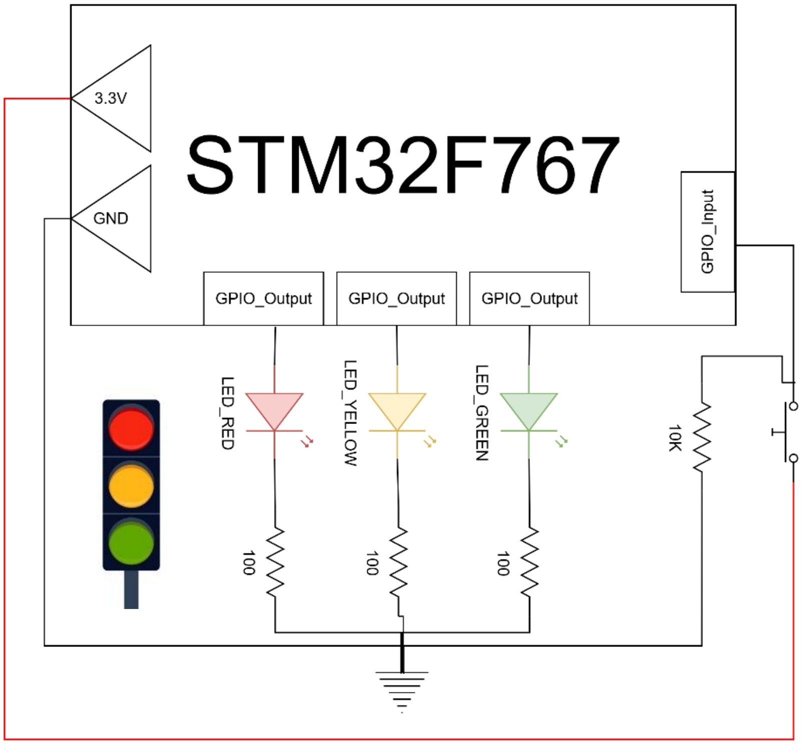

Figure 1: Traffic light circuit

2. In this question, you will develop a *reaction time* game. It looks like this:

Contrary to what is shown in the picture above, you will use only 2 push buttons and 2 LEDs instead of an LCD screen. See Figure 2 for the circuit diagram. The circuit diagram consists of 2 push buttons, 3 LEDs and 5 resistors.

Each user will interact with only one button throughout the game. One LED (LD3) is assigned as reflex LED and will switch on randomly within 20 seconds after the system starts. You may use the given generateRandomNumber() function at the end of this question if you wish. (Remember that we are programming in C and you may use any C-libraries and syntax to build your algorithm)

The game rules are as follows:

- As soon as LD3 switches on, users are supposed to press their button.
- The person who presses the button first, will win the round. To indicate that this person's LED will turn on.
- If a person presses the button BEFORE the LD3 turns on, then the other person will win. To indicate that, the other person's LED will turn on.
- The game ends after each round. To restart, the players must use the reset button on the STM32F767 (the black button).

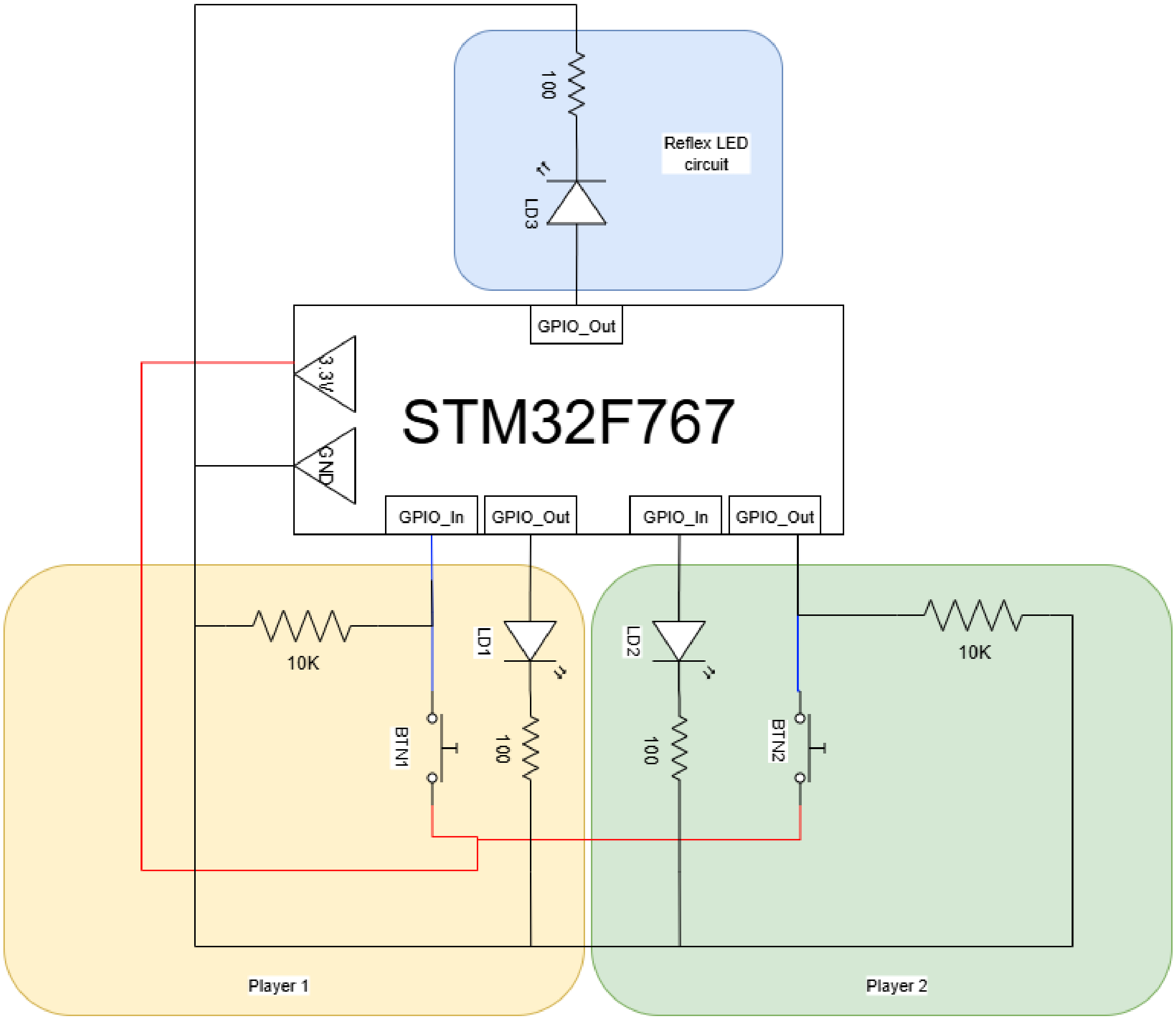

Figure 2: Reaction game circuit

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// Here is the ready random number generator function for you
// You can add it before your main() in the project
int generateRandomNumber(int min, int max) {
    int range_size = max - min + 1;
    return min + (rand() % range_size);
}

// Here is the example usage
int main() {
    srand(time(NULL)); // Seed the random number generator once

    int randomNumber1 = generateRandomNumber(1, 10); // Random number between 1 and 10
    printf("Random number between 1 and 10: %d\n", randomNumber1);

    int randomNumber2 = generateRandomNumber(50, 100); // Random number between 50 and 100
    printf("Random number between 50 and 100: %d\n", randomNumber2);

    return 0;
}
```