Microcontroller & Interfacing

CE205T Semester Project

Project Name: *"Whack a mole"*

A. Overview

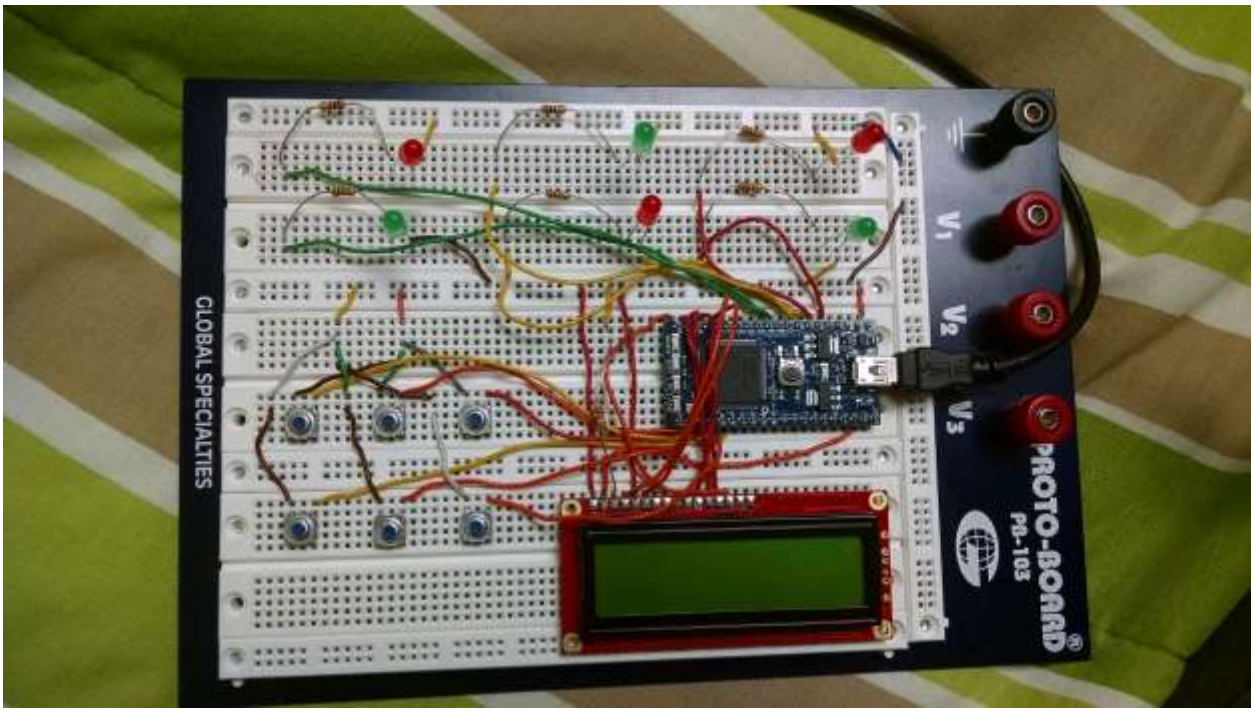
Overview:

The "Whack-a-Mole Game" project implemented on the STM32F407 microcontroller is a hardware-based adaptation of the classic arcade game. The project aims to recreate the interactive and engaging experience of the original game using the microcontroller's capabilities. The game involves a grid LEDs and Push buttons. It requires the player to hit the button crossponding to the LED that turns on.

Goals:

- It has a grid of 4x4 LEDs and 4x4 push buttons.
- Objective is to press the button crossponding to the turned on LED.
- Upon pressing correct button, score will increase (showed on 7 segment) and if no button is pressed or wrong button is pressed, game will reset.
- When game is over (player loses), a buzzer will make a sound.

The end result somewhat will look like this:



B. List of Components Used

Component	Cost	Reference Manual
STM32F407 (1)	16K	STM32F407/417 advanced Arm®-based 32-bit MCUs - Reference manual
10mm LEDs (16)	10x16 = 160	
Resistors (32)	4x32 = 128	
4 Pin Push Buttons (16)	10x16 = 160	
Passive Buzzer (1)	100	
LCD 16x2 (1)	450	
Vero board	3x 50 = 150	
Soldering Wire	2x 100=200	
Soldering Gun	From the lab 😊	

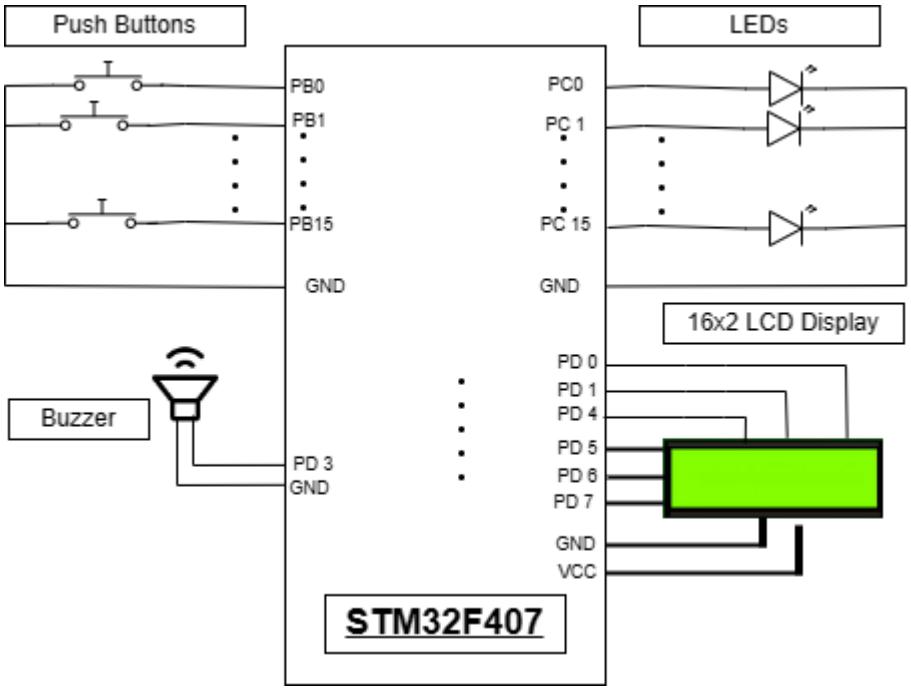
C. Peripherals of STM Microcontroller being used

In this project, we will be using Following GPIOs:

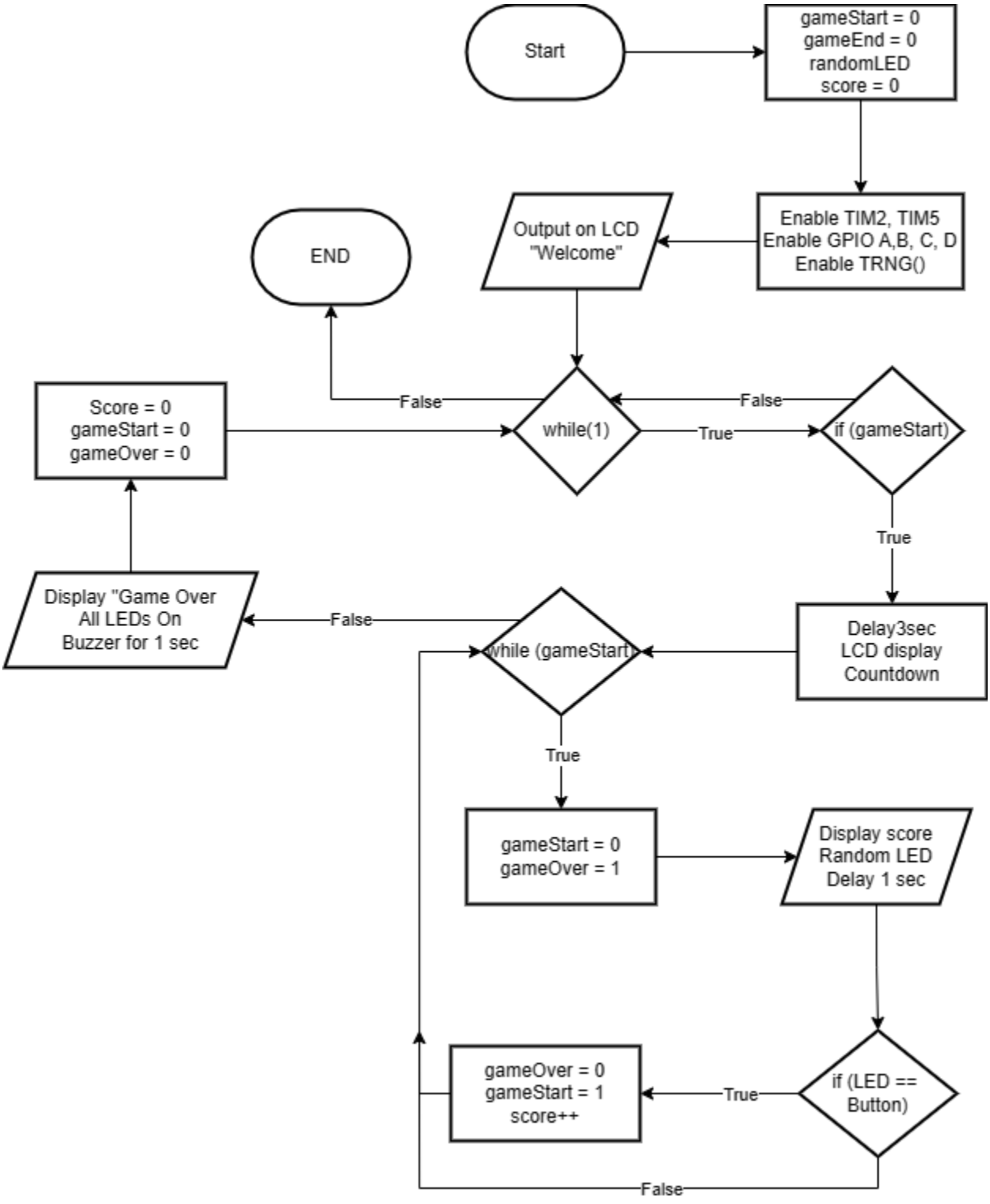
- **GPIOC (RCC->AHB1ENR):** All 16 (0 to 15) pins for LEDs.
- **GPIOB (RCC->AHB1ENR), (RCC->AHB2ENR):** All 16 (0 to 15) pins for push buttons (external interupts).
- **GPIOD (RCC->AHB1ENR):** PD0, PD1 and PD4 to PD7 will be used for LCD displays to display score.
- **GPIOD (RCC->AHB1ENR):** PD3 to connect a buzzer.
- **GPIOA (RCC->AHB1ENR):** PA0 to start the game.
- **TIM2 (RCC->APB1ENR):** Timer 2 to add 0.5 seconds delay to change LEDs.
- **TIM5 (RCC->APB1ENR):** Timer 5 for 3 seconds delay.
- **RNG (RCC->AHB2ENR):** Use to generate true random number.

D. Block Diagram/Schematic

Pin Connectivity/ Block Diagram:



E. Flow Chart (Required at the time of final submission)



F. CEP (Project Complexity) Attributes - Describe Briefly

Attribute	Description	Complexity Level in your project
WP1: Depth of knowledge	The project shall involve in-depth engineering knowledge related to the area of Microprocessors, Microcontrollers & Interfacing [WK-4, Engineering Specialization].	In our course we get to know the concept of interrupts and timers which are used in such a way that when a button is pressed interrupt is generated which further decides what is happen next in our program. We used the concept of usart a bit in order to get the connections of lcd
WP2: Range of conflicting requirements	The project has multiple conflicting requirements in terms of optimal usage of peripheral resources available on a Microcontroller.	Our 2-digit multilpex7 segment display was working perfectly when it was run in a separate module but when we added it in our code it was not giving the correct output when we checked internet for help it gave us the error that overload peripherals so that is why we used lcd display for that. Also, Keil was not 16 handling the interrupts al a time correctly so we shifted to CUBE ide
WP5 Extent of applicable codes	The projects expose the students to broadly defined problems which require the development of codes that may be partially outside those encompassed by well-documented standards.	In our project we used true random generator of STM32 in order to create a random value to one a random led. The reset variable value is generated depending upon our interrupt. Then we used some variables in order to store the value of score also.
WP7 Interdependence	The projects shall have multiple components at the hardware and software level.	We faced the main challenge of dealing with the interrupts as there were many interrupts and their interrupt handlers. We used some other functions such as reset pin in polling as our interrupts were full.

G. Code

CODE is present in the Whack-it folder attached on github

Final Project Model Display:

References [Negative Marking of 20% if this section is skipped]

- **BUZZER**: https://components101.com/misc/buzzer-pinout-working-datasheet
- **LCD:**
  https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet
  https://www.youtube.com/watch?v=_NoSyFVUNdU
  https://www.youtube.com/watch?v=ITTBWSQTi3c
- **CUBE-IDE:**
  https://www.youtube.com/watch?v=gL8OoS9E1rw&list=PLnMKNibPkDnFCosVVv98U5dCuIE6T3Iy8
- **STM32F407VGtx:**
  https://www.st.com/en/microcontrollers-microprocessors/stm32f407vg.html
- **INTERUPTS:**

https://deepbluembedded.com/stm32-interrupts-tutorial-nvic-exti/

- **TIMERS:** https://www.st.com/resource/en/product_training/STM32L4_WDG_TIMERS_GPTIM.pdf
- **USART:**
- **TRUE RANDOM NUMBER GENERATOR:** https://www.st.com/resource/en/application_note/an4230-random-number-generation-validation-using-nist-statistical-test-suite-for-stm32-microcontrollers-stmicroelectronics.pdf