GEBZE TEKNİK

ÜNİVERSİTESİ ELEKTRONİK

MÜHENDİSLİĞİ

ELEC334

Microprocessors

PROJECT01

Yakup Yıldız

171024006

Introduction

The goal is for this project is to implement a randomized counter in Assembly. A 4-digit 7-segment display should be connected that can show my school ID's last 4 digit which is "4006". When the code is not countdown this ID should display which is called idle state. There is a button for generate a random number and then start counting down to 0. The generated random number should be between 1000-9999. When counter reaches 0, the number 0000 should be displayed for second, then the code should back to idle state waiting for the next button press. Pressing the button while counting down should pause counting, and pressing again should resume counting.

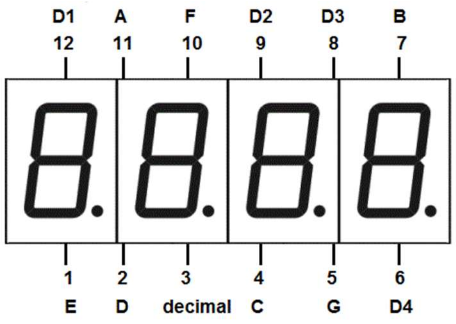
Randomized Counter

First of all I determined tasks to begin this Project. These tasks helped to keep things steady.

Tasks:

1:

Understanding of 4-Digit 7-Segment Display structure and ligting up a digit.



2:

Ligting up a 4-Digit 7-Segment Display and understanding its structure. Determining the pin connections between Microcontroller and 4-Digit 7-Segment Display.

PA0-Decimal	PB0-Ex.LED
PA1-A	PB2-Button
PA4-B	PB4-D1
PA5-C	PB5-D2
PA12-D	PB9-D3
PA11-E	PB8-D4
PA6-E	
PA7-G	

3:

Assigning numbers on assembly for digits. The first thing to do was preparing a table that helped me to define a hex numbers for numbers.

Pin / Number	PA1-A	PA4-B	PA5-C	PA12-D	PA11-E	PA6-F	PA7-G
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	0	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1

Table 1.

HEX Digit Codes	
0	0001_1000_0111_0010 = 0x1872
1	0000_0000_0011_0000 = 0x30
2	0001_1000_1001_0010 = 0x1892
3	0001_0000_1011_0010 = 0x10B2
4	0000_0000_1111_0000 = 0xF0
5	0001_0000_1110_0010 = 0x10E2
6	0001_1000_1110_0010 = 0x18E2
7	0000_0000_0011_0010 = 0x32
8	0001_1000_1111_0010 = 0x18F2
9	0001_0000_1111_0010 = 0x10F2

Table 2.

4:

Displaying my ID on 4-Digit 7-Segment Display. In multiplexed applications, all the digit segments are driven in parallel at the same time, but only the common pin of the required digit is enabled. By enabling or disabling the digits so fast that it gives the impression to the eye that both displays are ON at the same time as the human eye cannot differentiate it when the speed is too high. This technique is based on the principle of Persistence of Vision of our eyes.

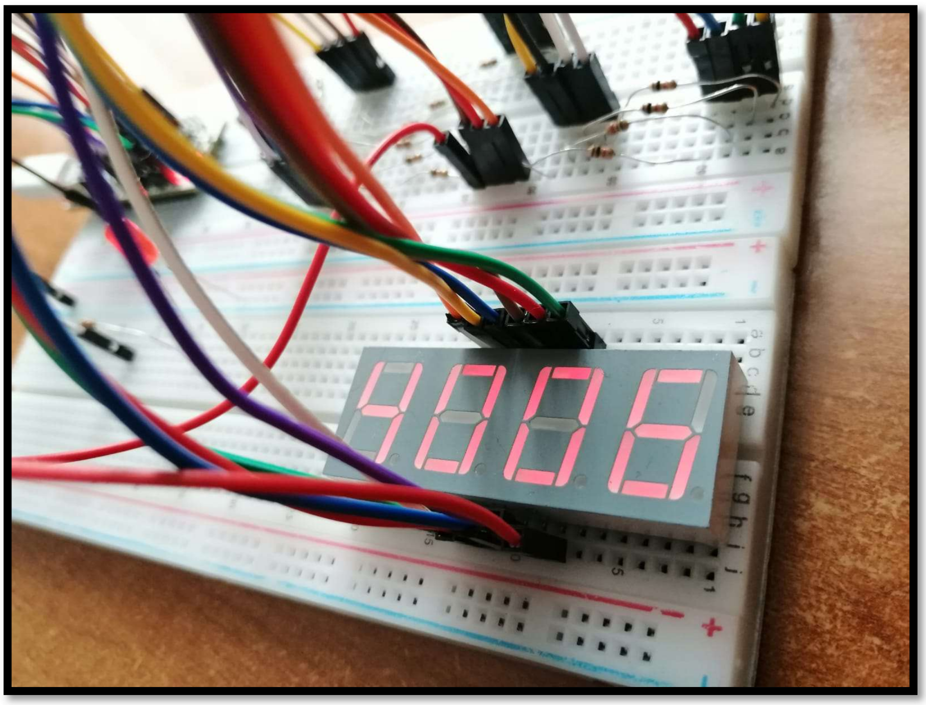


Figure 1. ID:4006

5:

Creating a random number generator. A register that adds one to itself accoring to the number of iterations.

6:

A 4-digit random number was generated, but it is necessary to examine the number in each digit separetely by separating it into its digits. At the same time, to achieve a more functional operation by adding countdown process while seperating random number into its digits.

7:

When the countdown is over, to obrain the desired 0000 number to see on the 4-Digit 7-Segment Display screen. At the same time, external LED turns on and off while 0s are displayed on the screen.

8:

Pressing the button while counting down should pause counting, and pressing again should resume counting.

How the Code Works:

Proceeded sequentialy in the code, sticking to the given tasks. First of all defined peripheral addreses after that mode assignments are made as pins input, output. Created an infinitely spinning loop that calls idle function. The desired 4-Digit ID is printed in the function named idle. At he same time, in the idle function, it was checked wheter the button was pressed or not. If the button is pressed go to a random number generation function was called. In the random number function, the random number was created by the register in the idle increasing one by one. In this way, a random number was created. The created was too large fort his shrinking has been done. Since a number between 1000 and 9999 is requested, the balue 1000 was added after taking the mod according to 8999 and assign it. Since the number created is 4 digits, but the required number of digits is one, an algorithm was established accordingly and the obtained number was parsed into digits. The number seperated to the digits was inserted into the countdown process and printed on the 4-Digit 7-Segment Display. When the countdown is finished, the numbers 0000 was obtained on the screen. At the same time, the external LED was turned on while the numbers 0000 were displayed on the screen and the external LED was turned off when the state was switched to the idle state.

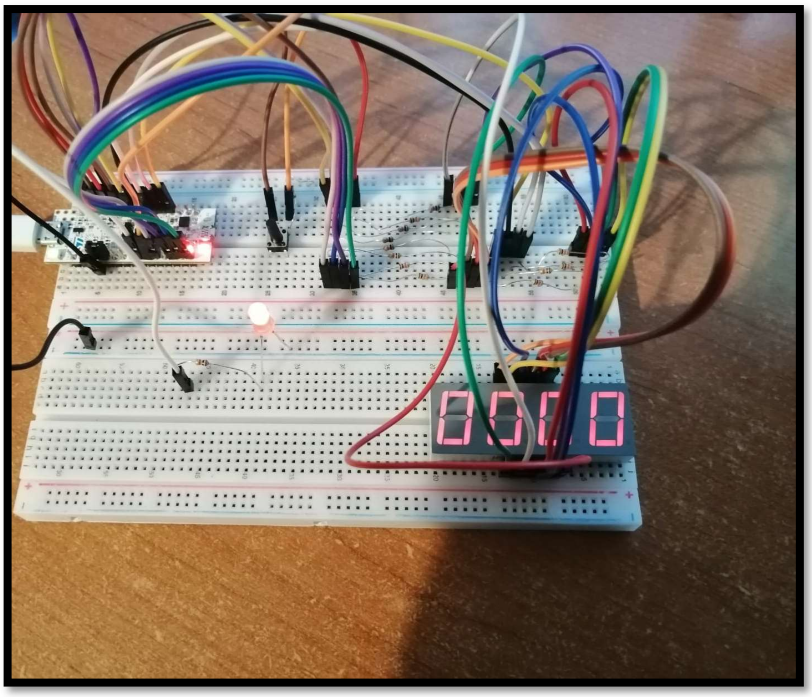
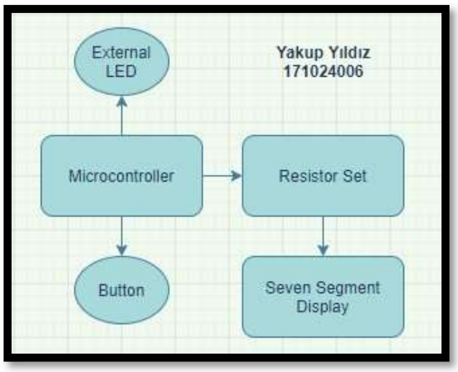
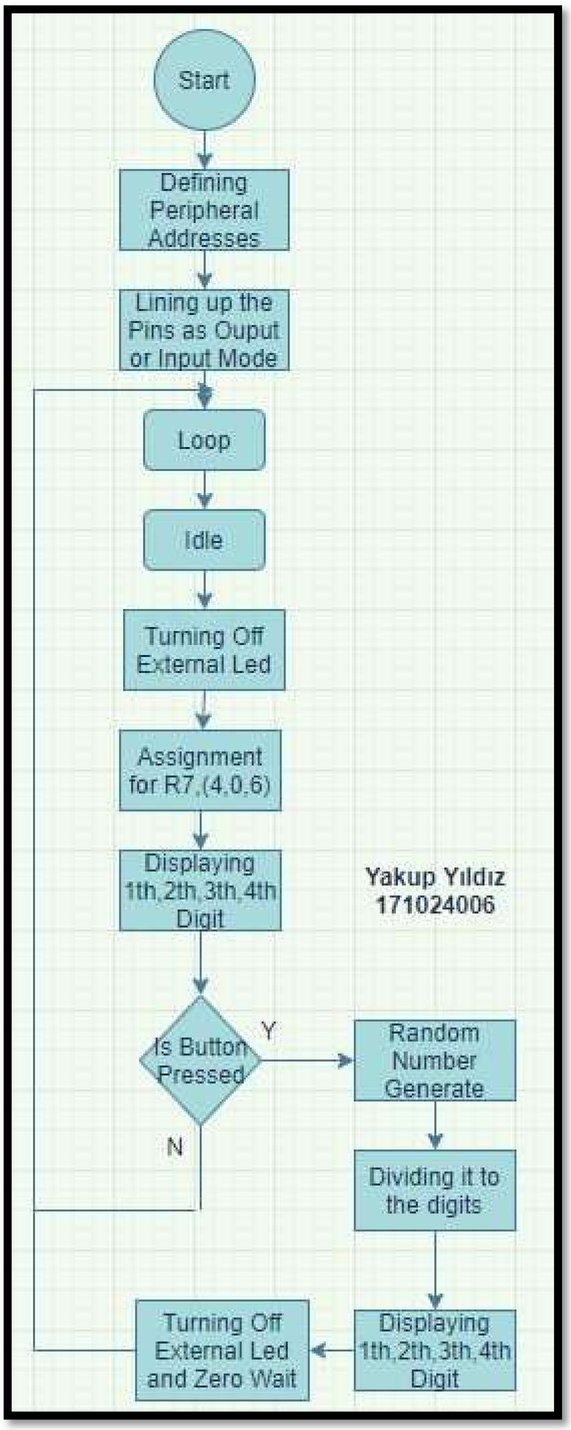


Figure 2. External Led ON, 0000 Display

Block Diagram:

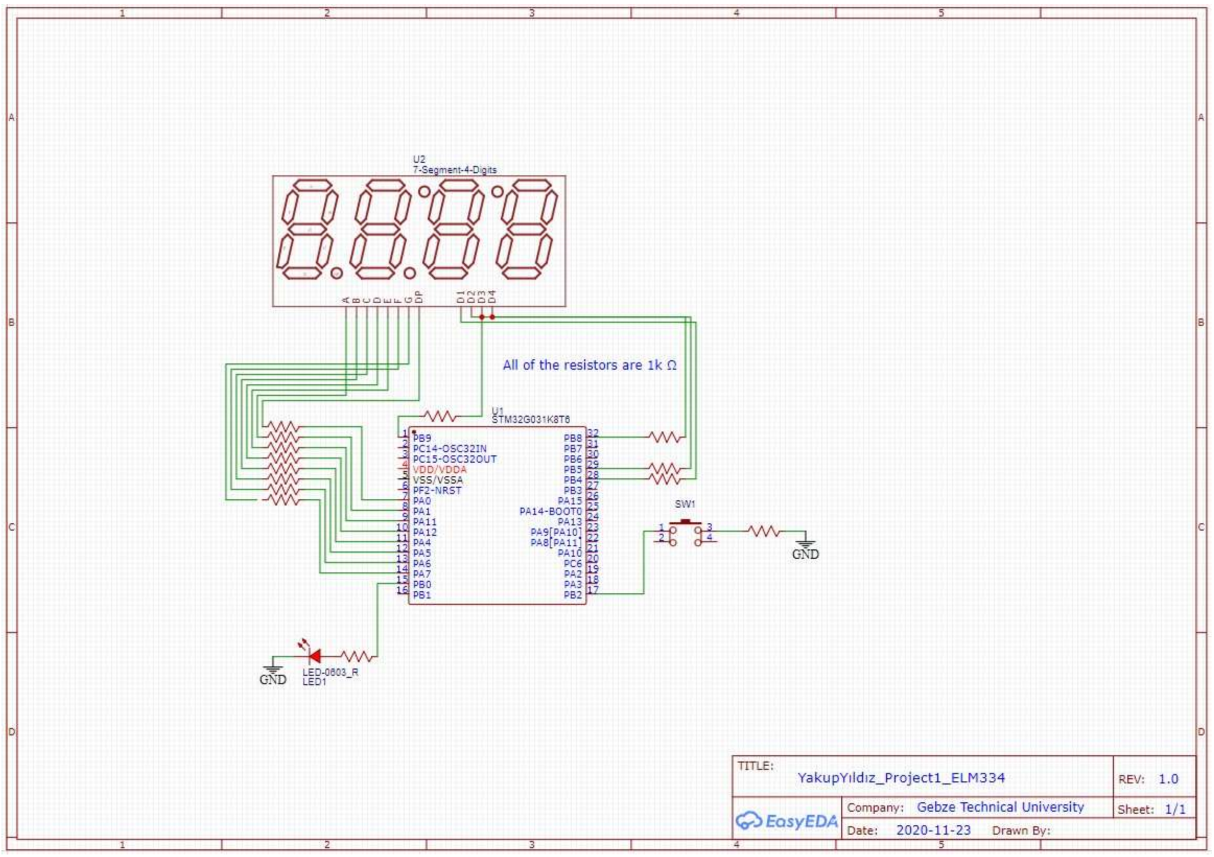


Flowchart:



Start
Defining Peripheral Addresses
Lining up the Pins as Ouput or Input Mode
Loop
Idle
Turning Off External Led
Assignment for R7,(4,0,6)
Displaying 1th,2th,3th,4th Digit
Yakup Yıldız 171024006
Is Button Pressed
Y
N
Random Number Generate
Dividing it to the digits
Displaying 1th,2th,3th,4th Digit
Turning Off External Led and Zero Wait

Schematic:



Results and Comments:

Seven of the assigned tasks were successfully accomplished. The desired randomizer has been successfully created. Unfortunately it could not be done due to not enough time fort he last task. It was a challenging project. Because expressing something in assembly is much more diffucult than programming languages. To give an example, a instruction that can be done in a single function in C language can be done in more than one function in assembly. Assembly is long and tedious to write initialy. In the other side, assembly give a complete control of code and can Access machine dependent register and I/O. One of the most challengin things for me in this Project was the pop and push instructios. The transactions made were broken due to the interference of the linked places. As a solution to this, I avoided using push and pop instructions as much as possible. The second most challenging thing for me was the start date of the project. I could have started earlier and did it more comfortably and maybe I could have done the last task of this project.

Referance

- *4-Digit 7-Segment Display datasheet*
- *rm0444_reference_manual*
- *arm_architecture_v6m_reference_manual*

Parts List:

Microcontroller-STM32G031KT6	105 ₺
1 x LED	0.16₺
1 x BUTTON	0.24₺
14 x 1kΩ Resistor (1/8W)	0,84₺
2 x Breadboard	15₺
27 x Jumper Cable	3₺
Sum	124.24₺