



YOUR FREEDOM IN LEARNING

EE306 - Microprocessors

**Term Project**  
**Sliding Text**

June 8, 2020

Ali Bahadır Bulut  
#041602017

















Alp Gokcek  
#041701014

















Erdal Sidal Dogan  
#041702023

## 1 Project Description

In this project, the text has been shown by scrolling them on a 7-segment display. Which text to print on 7-segment displays has been chosen by using switches and has been adjusted the scroll speed of texts with the buttons. The main goal has been that this project was done by using the peripherals on the board. However, the project has been done via the internet browser using the CPULATOR ARMv7 System Simulator. The advantage of using this simulation site was to use eight 7-segment displays. ASCII codes have been used to describe letters and numbers in this project. ASCII equivalent of the characters has been converted to hexadecimal and implemented in the project. Additionally, A9 private timer has been used as a timer in the project.

Our design supports the characters provided in Figure 1.

															
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	

															
P	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	








						
4	5	6	7	8	9	Space

Figure 1: Sample output on 7-segment displays for input "EE306 Fun"

## 2 Sample Output



Figure 2: Sample output on 7-segment displays for input "EE306 Fun"



Figure 3: Sample output on 7-segment displays for input "1234567890"



Figure 4: Sample output on 7-segment displays for input "Mustafa Kemal Atatürk"

## 3 Flowcharts

### 3.1 MAIN

After the initialization, the program starts executing the *Main* subroutine. It consists of chaining other subroutines sequentially such as *SET\_WORD*, *CHECK\_PAUSE*, *SET\_SPEED*, *PRINT\_STRING*.

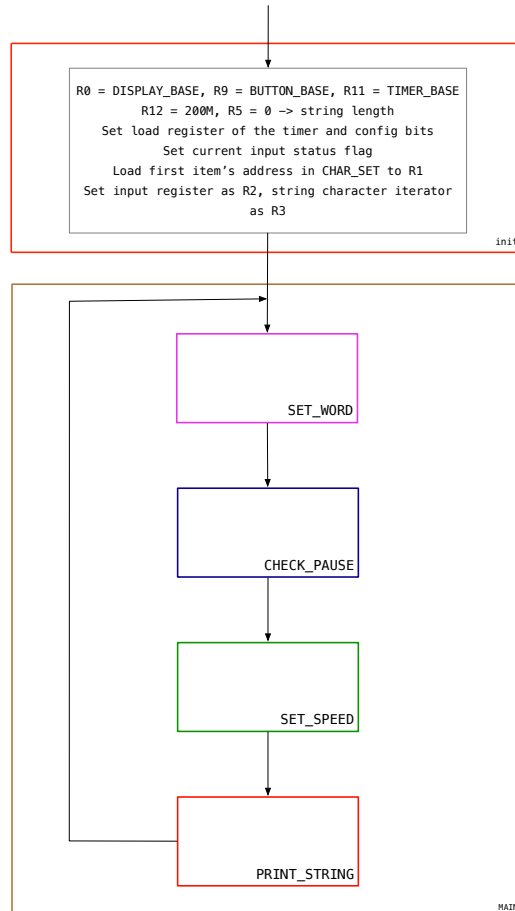


Figure 5: Flowchart of Main Loop

### 3.2 SET\_WORD

The program supports multiple words. Selection is being made by the 'Switch' input. The subroutine *SET\_WORD* is responsible for choosing the correct word according to the input.

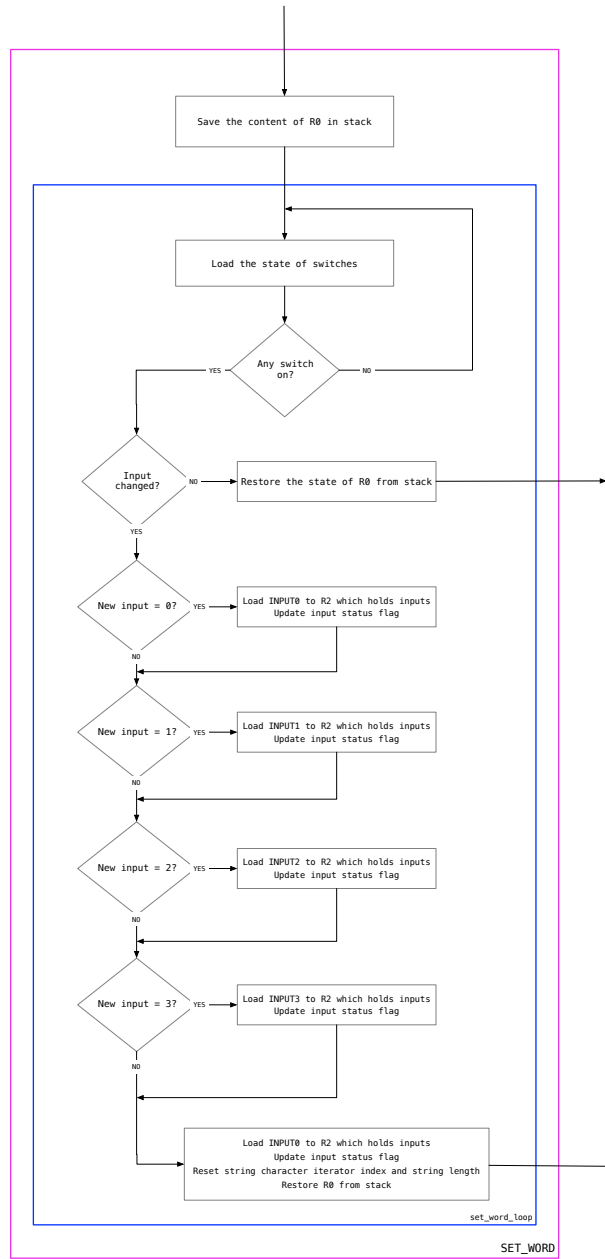


Figure 6: Flowchart of Set Word Subroutine

### 3.3 SET\_SPEED & CHECK\_PAUSE

As the names suggests, *SET\_SPEED* and *CHECK\_PAUSE* subroutines are pausing the slider and sets its speed. Both of these functions are triggered by certain combinations of push buttons. They check the latest push button configuration pause or increase/decrease the speed accordingly. There are 3 different predefined levels of speed;

- Normal: 1 sec
- Fast: .25 sec
- Slow: 4 sec

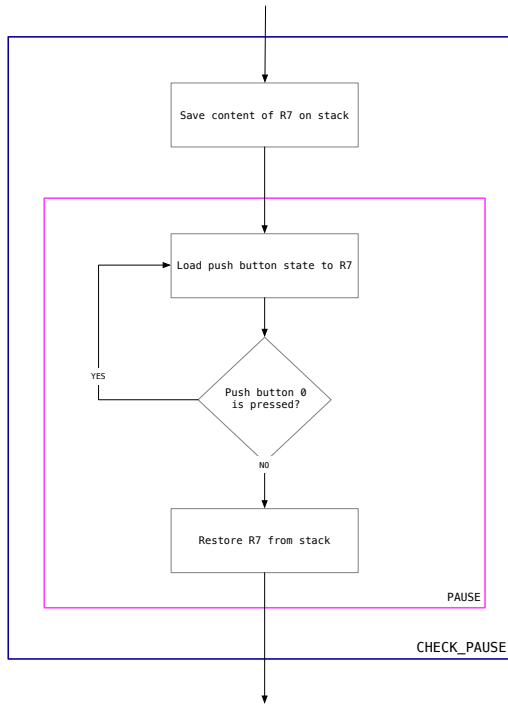


Figure 7: Flowchart of Check Pause Subroutine

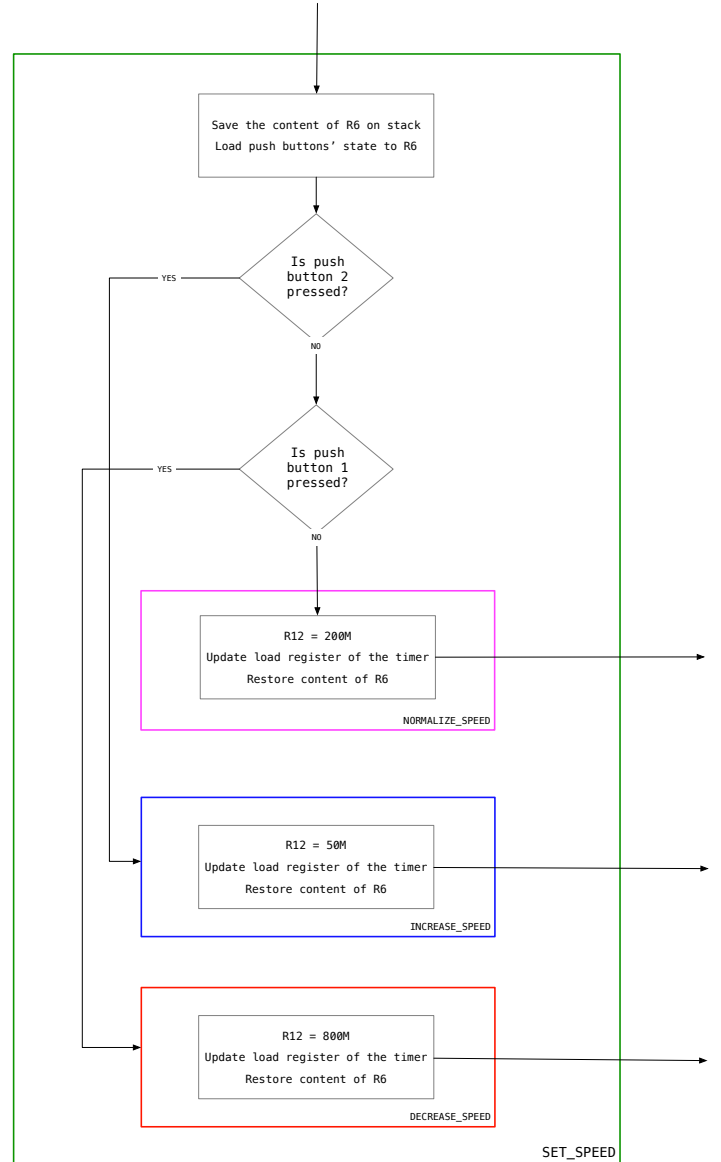


Figure 8: Flowchart of Set Speed Subroutine

### 3.4 PRINT\_STRING & DELAY & FIND\_LOC

#### 3.4.1 DELAY

The A9 Private Timer is being used for delays. This subroutine checks the timer configuration continuously until it finishes its countdown, after it finished this subroutine resets it and start over again when called next time.

#### 3.4.2 PRINT\_STRING

This subroutine determines which letter to print relying on the input and which 7segment to print it.

#### 3.4.3 FIND\_LOC

The representation of possible inputs, which are capital and lowercase letters along with numbers, are stored sequentially in the memory. This subroutine is used for locating the correct 7-segment representation in the memory.

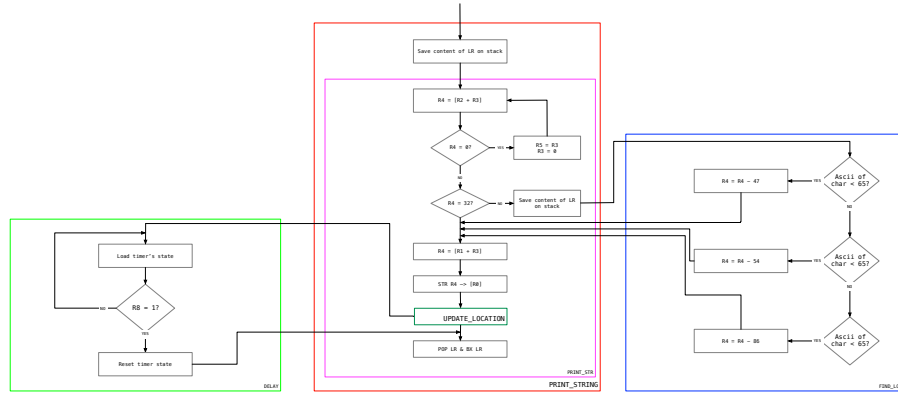


Figure 9: Flowchart of Print String Subroutine

#### 3.4.4 UPDATE\_LOCATION

This subroutine determines which 7-segment to write next character. Given that there are jumps between memory locations of 7-segments and circular loop, it has been decided that its best to handle these operations in a different subroutine. First we set it to be the 7-segment that is right next to the previous one. Then we check if has exceeded any limits, if it did, reset the position accordingly.

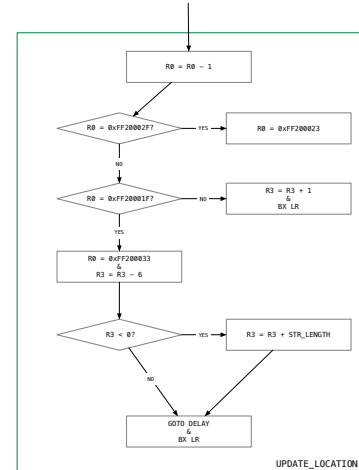


Figure 10: Flowchart of Update Location Subroutine