**Assignment on Assembly language**

**Topic: Race Signal with buzzer**

**Taha Mohamed Alzein(269055)**

**Muhammad Nadeem (266704)**

**Fadi Atia Dasus(266265)**

**Oskars Arajs(266534)**

**Balkis Ibrahim(260092)**

**Supervisor**

[Christian Flinker Sandbeck](https://via.itslearning.com/Dashboard/)

**ICT Engineering, VIA University College, Horsens**

**3rd Semester (05-12-2018)**

**Table of content**

[**Abstract**](#_1t3h5sf) **3**

[**1 Introduction**](#_aa7v5xb17dj7) **3**

[**2 Requirement**](#_77ocpjbb0f4c) **4**

[**3 Analysis**](#_jve8li63msnk) **5**

[**4 Design**](#_7pzpyp9bzc7p) **6**

[**5 Implementation**](#_7pzpyp9bzc7p) **6**

[**7 Conclusion**](#_7pzpyp9bzc7p) **8**

[**Reference**](#_7pzpyp9bzc7p) **8**

# 

# 

# 

# 

# 

# 

# *Abstract*

The aim of this project is providing a race signal that is implemented in assembly language. This device will control the starting point of the race by changing the color of the LEDs. By pressing the button, Red led will on and players should be active, so race should start when led is green and buzzer emit the sound. Developing this tool has gone through different stages which will be discussed in detail in this report.

# 1 Introduction

The goal of this assignment is to get familiar with assembly language programming. This assignment presents the analysis, implementation and test of a Race Signal with buzzer system. If you are a beginner then this assignment is very essential.

The idea of a Race Signal with buzzer is not new, but It can be useful in the future. This assignment doesn't use many parts but the important parts are red, yellow, green leds and active buzzer.

Device will be control by the button and it will start out by turning the red led on to indicate that all the players are in the queue when the led switches to yellow, It means that the players should be ready to start and at last led is green which shows the players to run now. The device will emit also a sound which is synchronized with the green light.

# 2 Requirement

## 2.1 Functional requirements

1. The device contains three light (red, yellow, green).

2. The system must be able to turn the red light on when the button is pressed

3. The system must be able to switch the between the three lights with a delay of one second after each light.

4. The system must emit a sound which is synchronized with the green light.

## 2.3 Non-Functional Requirements

1. The software should be written in assembler language.
2. The appliance should be build using Arduino AVR and basic component.
3. The project should interact with some sensors and actuators.

# 3 Analysis

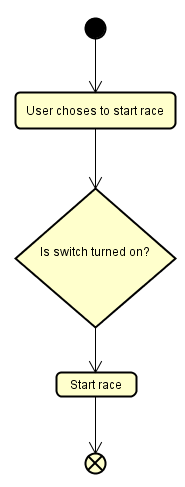


Figure 1. Activity diagram

The activity diagram shows the steps the user takes to get to the required result. In this case, these are the steps needed for the Race light to work properly.

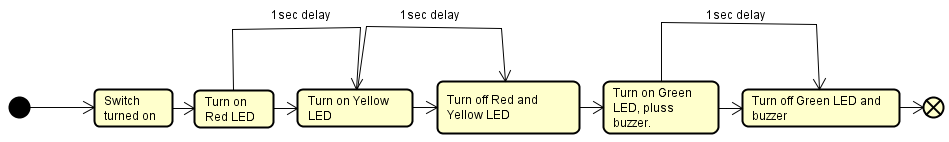


Figure 2. State machine diagram

The state machine diagram shows the steps the device takes to reach the desired endpoint of the program. It shows all the delays between each step and shows the actions taken after each step.

# 4 Design

The device was designed with the following components:  
 1) ATMega2560 AVR microcontroller

2) Breadboard  
 3) Connecting cables  
 3) 10k resistor

4) 3LEDs (Red, Yellow, Green)  
 5) Push button

The workflow of the device as shown in figure 2 indicates that when the Button is pressed it sends a signal to PortB to turn on the Red led. After a delay of 1 second, the signal is copied to via register 23 to PortA to turn on the Yellow light. After a delay of 1sec the Red and Yellow LEDs will turn off and the Green led will turn on with a sound signal coming from the active buzzer. And after a 1 second delay, the Green led and buzzer will turn offending the sequence.

# 5 Implementation

The device was implemented in AVR Assembly language using an ATMega2560 microcontroller with a 16MHz Central Processing Unit.  
The circuit is built by connecting a push button to PD0, Red led to PB0, Yellow led to PA0 and Green led with buzzer to PC0.

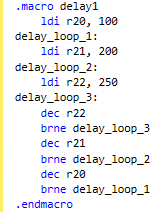


Figure 3. delay1 Macro

The macro represented in figure 3 is was made to function as a delay of 1 second.

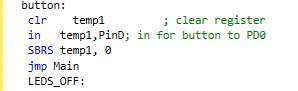


Figure 4. Button class

The button function represented in figure 3. shows how the signal is copied to the temporary register 16 (temp1). The command line SBRS (Skip if bit in register is set) means that bit stored in register 16 is 0 skip the next line of command and continue. This ensures that if there is no input from the user the device won’t turn on.

**6 Test and Result**

The black box technique was used to test the device ability to act the way that we expected.

We connected the Arduino to the laptop and we upload the code. After we pressed the button the red light was on, it waited for a second to switched to yellow and also one second to green. when the green light was on the buzzer starts to emit a sound. in the end, everything was off.

Obviously, the device is built, programmed and behaves in a good manner.

# 7 Conclusion

Building this project turned out more difficult then what the estimate was. But following up on the analysis and design section, the project was executed. By following the Analysis section we determined the steps necessary for a successful main scenario. And from the design section we were able to construct a functioning device using all the necessary components. And from the implementation we concluded that the assembly language code that was made for the device is working as it should.

# Reference

[Mazidi, 2011]: Muhammad Ali Mazidi, Sarmad Naimi and Sepehr Naimi - The AVR Microcontroller and Embedded System using Assembly and C

AVR Instruction Set Manual

http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf