



Traffic Lights Management System

Abstract geometric lines and shapes in the bottom right corner, including a large thin white line forming a wide 'V' shape, and several smaller white line segments and brackets.



Team members:

- Huda Essam Abdel-Aal
Fahmy
 - Mohamed Mahmoud
Ahmed Zaki
 - Arafat Ibrahim Ali
- 
- 



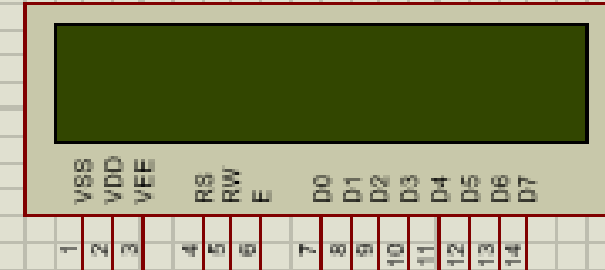
Hardware Components

- MCU: ATMEGA32
- LCD 16x2
- 6 LEDs
- 7 Resistors
- Push Button
- Timer Peripheral

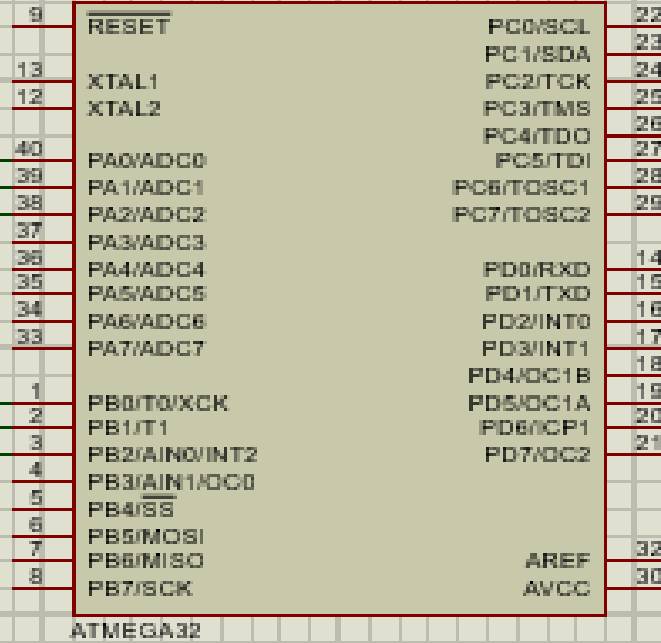
Hardware Architecture



LCD1
LM016L



U1



R2
300



D1
LED-GREEN

R3
300



D2
LED-YELLOW

R4
300



D3
LED-RED

R5
300



D4
LED-GREEN

R6
300



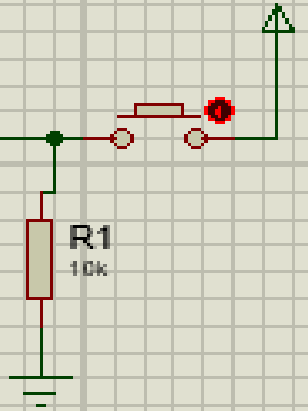
D5
LED-YELLOW

R7
300



D6
LED-RED

R1
10k





System Description

- This is an on-demand traffic system, which gives priority to pedestrians over vehicles.
- The system detects when a button is pressed, afterwards, based on its current state it will decide what to do. It allows pedestrians to walk by making sure cars are stopped first.



Software Architecture

- The software consists of 3 layers:
 - **MCAL**: we created modules to interact with timers, and GPIO.
 - **HAL**: we created drivers to interact with LCDs, LEDs, and Buttons.
 - **App**: The main system logic is implemented in a separate module. Which imports the HAL and MCAL drivers to take the needed actions on the AVR.



System Design

- The entire system is modeled as a state machine which operates in either one of two states: Normal State, or Pedestrian State.
- When the normal state is activated, the system operates like any normal traffic lights.
- When the button is pushed, a state transition to pedestrian mode is triggered.
- Depending on the current state and time, the LED lights are configured.



System Diagram

- A flowchart that describes system states transitions can be found at our project GitHub repository:
- [Traffic System State Diagram.pdf](#)

Thank you