Traffic Lights Mangement System

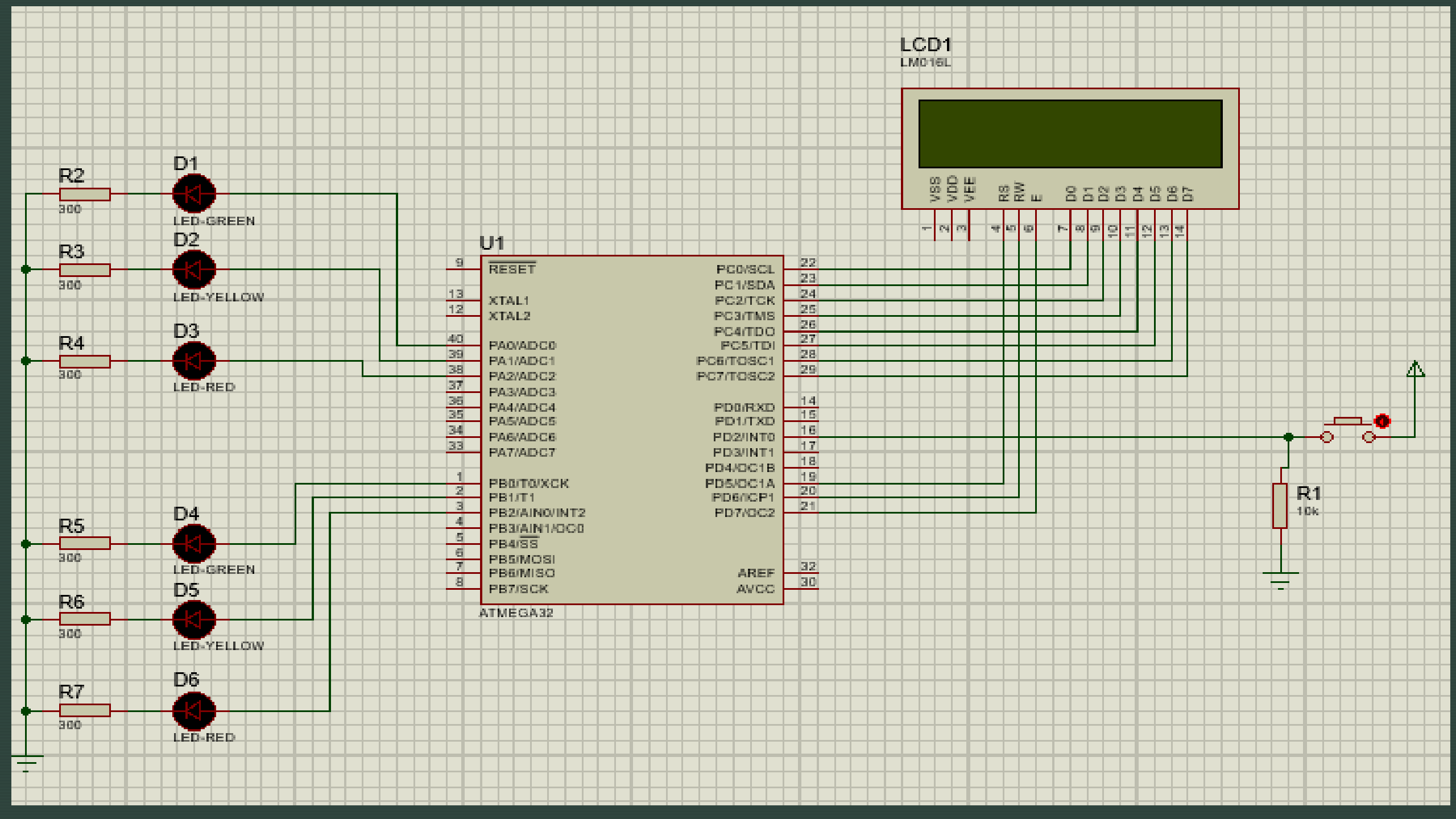
Team members:

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

Hardware Components

- MCU: ATMGEA32
- LCD 16x2
- 6 LEDs
- 7 Resistors
- Push Button
- Timer Peripheral

Hardware Architecture


LCD1
LM016L
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA32
R1
10k
R2
300
R3
300
R4
300
R5
300
R6
300
R7
300
D1
LED-GREEN
D2
LED-YELLOW
D3
LED-RED
D4
LED-GREEN
D5
LED-YELLOW
D6
LED-RED

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 `assets/Traffic System State Diagram.pdf`

Thank you