UDACITY

egyptfwd initiative

ON-DEMAND TRAFFIC LIGHT CONTROL

Embedded systems professional track

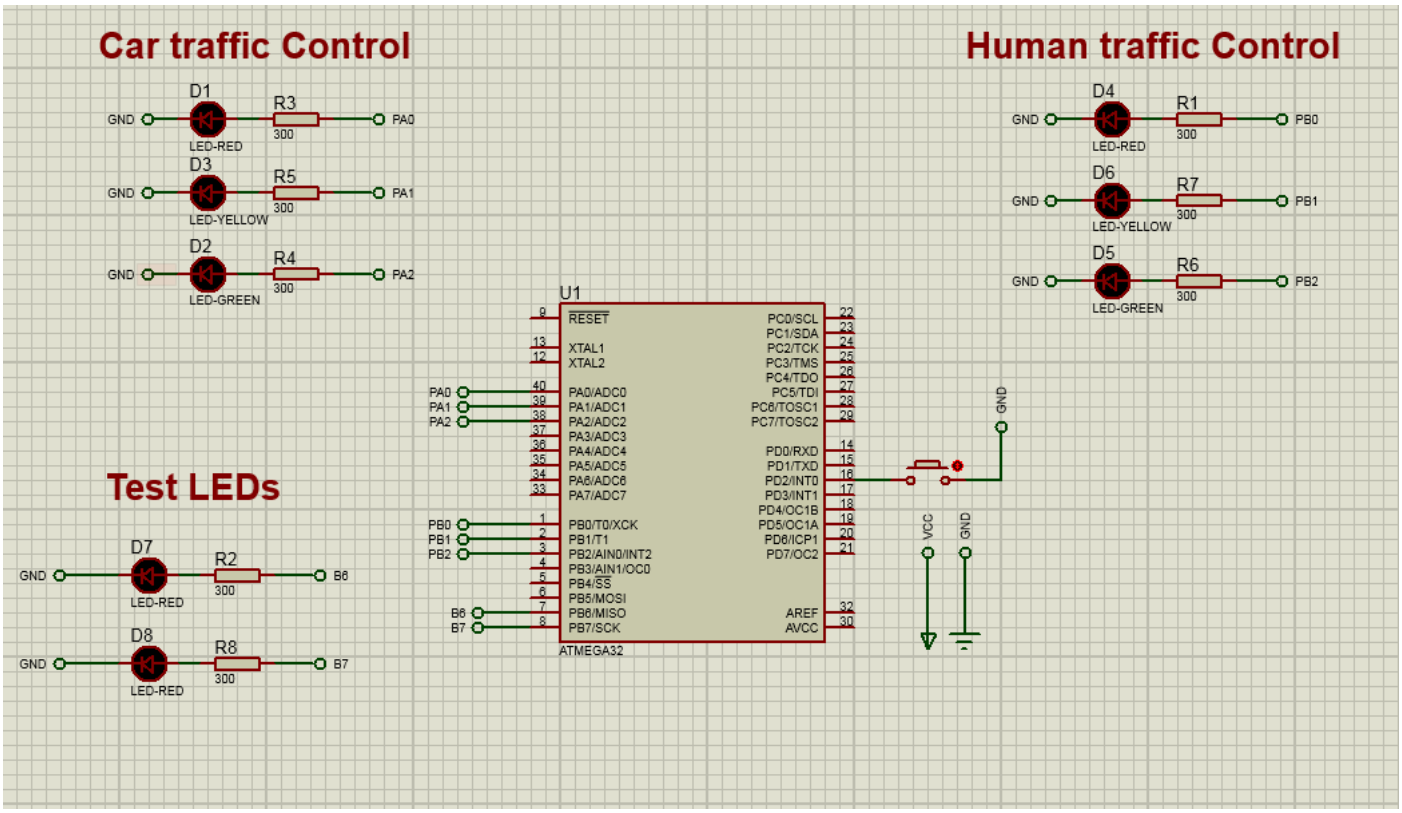
Project documentation

Ahmed Elshety

On-demand traffic control system

System description

The system aims to provide an on-demand traffic control system. It includes a pedestrian button to allow for pedestrians to pass.



System Functionality

The system can detect when the button is pressed. Afterwards, based on current state it would decide what to do .It allows pedestrians to walk by making sure cars are stopped first.

System Design

2.1 System Requirements

The system consists of:

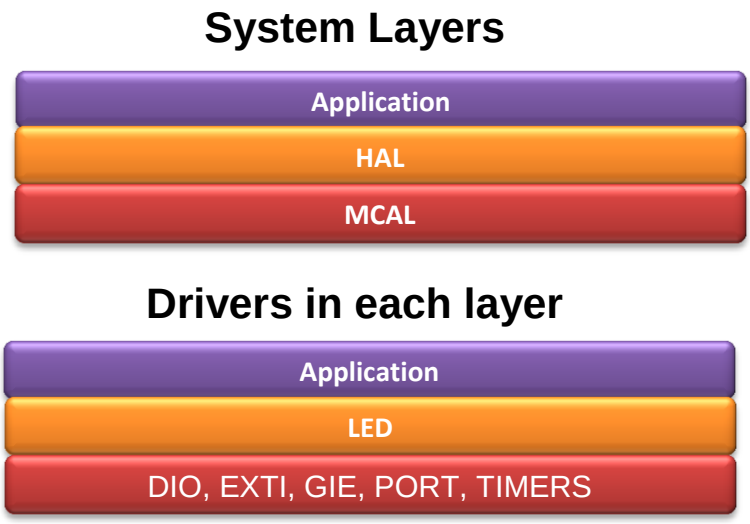
- AVR Atmega32 (1MHz)
- 8 300 Ohm resistors
- 1 Push Button
- 4 Red LEDs
- 2 Green LEDs
- 2 Yellow LEDs

2.2 Operating Environment

The program has been tested on Proteus simulator. It should be used in traffic light control systems on streets with a pedestrian push button included to allow for full system functionality.

2.3 Input & Output Formats

The only system input is in the form of the pedestrian push button. When it comes to output it handles 6 LEDs at once given the current state, time and push button press state.



Development environment preparation

1. Create Layers folders
2. Create modules folders in each layer
3. Create necessary .h and .c file for each module
4. Add header file guards
5. Create the main.c file that will call your application

Implement the MCAL layer

1- Create folders for each driver: DIO, EXTI, GIE, PORT, TIMERS

2- Implement (DIO, EXTI, GIE, PORT, TIMERS) modules:

- ✓ Fill in interface.h and register.h, private.h and config.h files with functions' prototypes, MACROS and registers address
- ✓ Implement program.c functions

Implement the HAL layer

Create folders for LED driver:

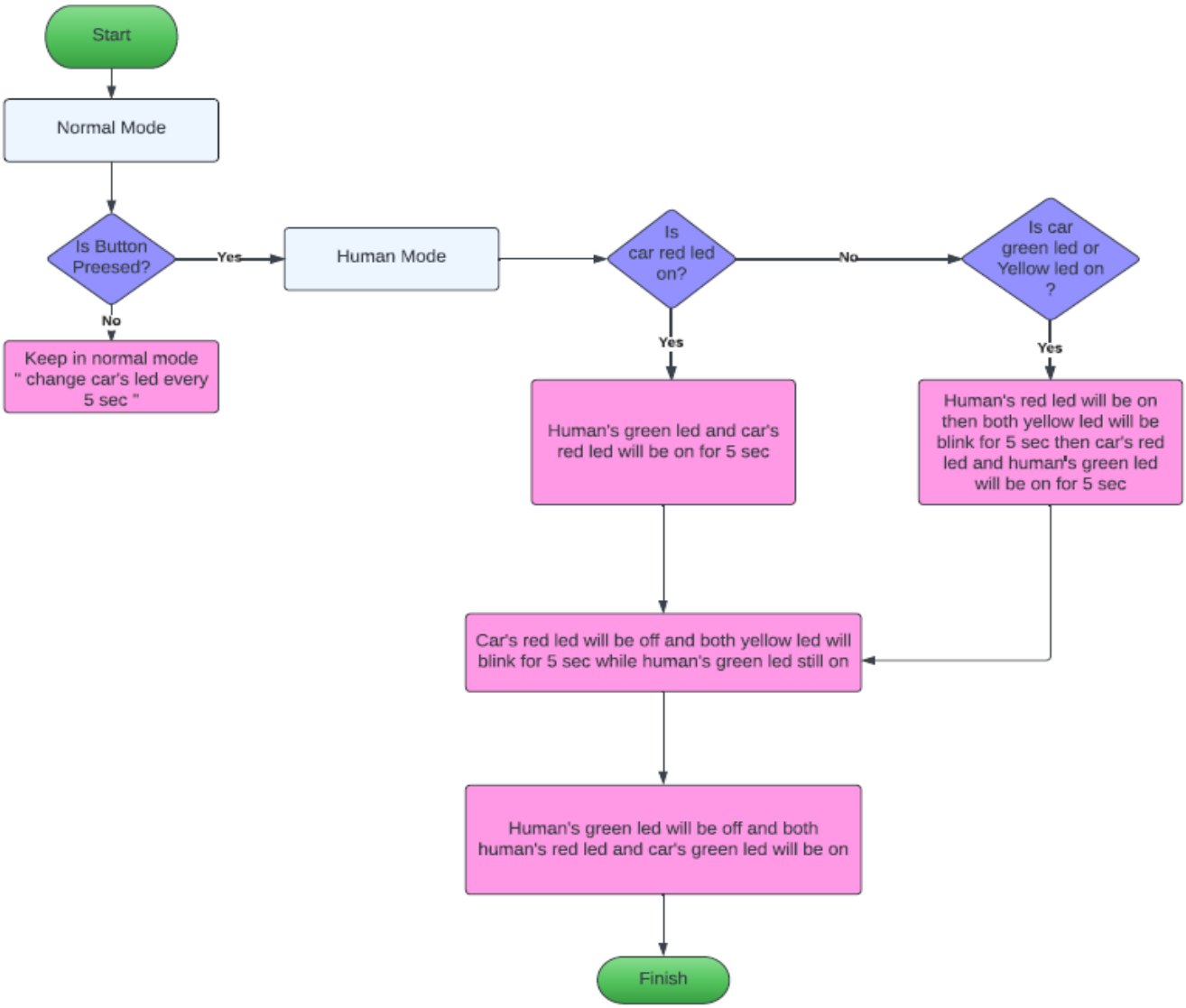
- ✓ Fill in LED_interface.h and LED_config.h file with functions' prototypes and MACORS
- ✓ Implement program.c functions (LED_u8TurnOffLed, LED_u8TurnOnLed, LED_u8ToggleLed)

Implement the application

1- Fill in application.h file with functions' prototypes

2- Implement AppStart() and AppInit() functions

Flow Chart



System constraints:

1. Dealing with the long press:

We solve it in a way that whether the press was short or long it has the same effect on the system and the interruption will occur only once till it is totally executed.

2. Dealing with the double press:

We solve it in a way that we neglect any press after the first one till the interruption is totally executed.