Simulation of T-type crossroad on FRDM-KL27z board

Project IMP
Brno University of Technologies

Pavel Yadlouski (xyadlo00)
September, 2020

Contents

1 Introduction

Aim of this project is to implement an algorithm for controlling traffic lights at an intersection with pedestrian crossings of "request" type. Traffic lights are represented by LEDs on shield with schematic representation of T-type intersection. For each pedestrian crossing there is button. Signal from button represents that pedestrian came to crossing and send request for crossing the road by pressing on corresponding button. Shield with LEDs and buttons is connected to FRDM-KL27z board.

2 Architecture

Intersection is divided into 3 logical parts: bottom, central and top part.

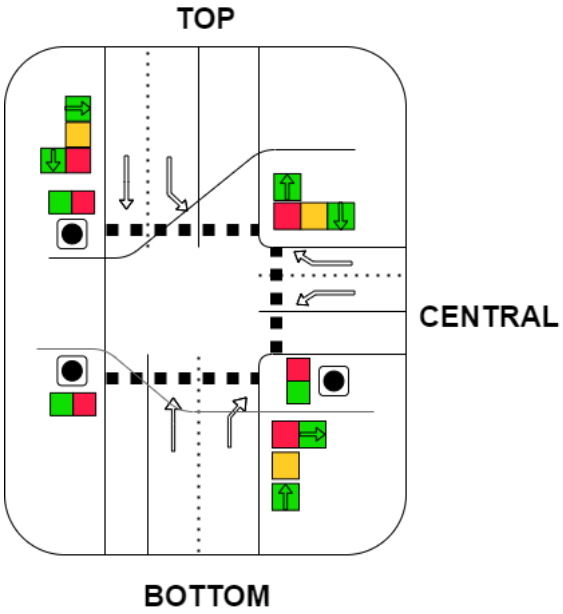

Figure 1: Logical division of intersection

In initial state bottom and top traffic lights are red and central traffic light has both green LEDs on. Check for any request is handled before starting new cycle. Cycle include sequence of steps from the initial state to the opposite side of semaphore and back to the initial state. Also any handler of interrupts ends in this initial state. The reason for this is that function for processing corresponding combination of buttons are called before new cycle, so loop can continue from initial state after request processing. One cycle in total takes 18 seconds. Taking one traffic light as an example:

1. 6 seconds in red
2. 3 seconds in orange
3. 6 seconds in green
4. 3 seconds in orange

3 Implementation

In main infinite loop of the program there is switching of all car traffic lights. For switching light on individual traffic light there are 4 functions. Each function switch light from one light to the next in corresponding order.

1. red → orange – *change_red_orange_**side**()*
2. orange → green – *change_orange_green_**side**()*
3. green → orange – *change_green_orange_**side**()*
4. orange → red – *change_orange_red_**side**()*

where ***side*** is one of three available sides: bottom, central or top. Each function contains simple setting and clearing bits of LEDs in corresponding registers. In functions *change_orange_green_**side**()* there is switch case that enables both green LEDs or only one of them based on its argument *arrow*.

3.1 Interrupts handling

When interrupts came, program switch to handler *PORTB_PORTC_PORTD_PORTE_IRQHandler()*. In this handler flag for corresponding button is set. Flags is a boolean value ***side**_button* So, during any cycle this flags can be set. When system is in initial state before next standard cycle based on combination of button flags appropriate function is called.

- *handler_**side**_only()* – for processing request from the only one button
- *handler_**side**_**side**_()* – for processing request that came from two of three button

Here ***side*** means one of three sides (bottom, central, top) where button is placed.

4 Problems and limitations