

Project Report

Author: Tong Zhang

zID: z5235242

1. General Description

This project aims to simulate traffic control of a short single lane road. The number of vehicles on east side and west side should be displayed on the LCD, and the traffic light patterns on both side should be displayed on the LED, with turning on representing green light, and turning off representing red light. Meantime, the direction and the number of cars on the road should be displayed on the LCD.

Two external interrupts (INT0 and INT1) are used to response to press PB0 and press PB1. In addition, three timer are developed. In details, Timer0 for the interval time between two adjacent vehicles to enter the road, Timer1 for simulating change the direction every 5 minutes when there is a congestion, and Timer2 for the 2 minutes that a car pass the road.

By analyzing the rules of traffic control, it could be concluded that two function should be applied in this project:

- a. Update and display the number of vehicles on the east and west would be used when (1). PB0 or PB1 pressed; (2). A car joins the road, the number on the corresponding side should decrease 1. So this function should be called in INT0, INT1 and Timer0. In INT0 and INT1, PB0 and PB1 would be pressed when a new vehicle detected by the sensor, thus the number of vehicles on the east or west side should increase 1, based on which button is pressed. In Timer0, after the interval time, a new vehicle would be allowed to join the road if is not full, thus the number of vehicle on the corresponding side should decrease 1.
- b. Update and display the road (including the direction and the number of vehicles) would be used when (1) a new vehicle joins the road which is not full, i.e. in Timer0; (2) a vehicle leaves the road, which is handled in Timer2; (3) the direction changes, i.e. in Timer1.

As for the special consideration: emergency, we handled it in the main function. In this condition, the LED light should flash as the pattern 0b00011000. Besides, Timer1 and Timer2 should be masked since no vehicle is allowed to join the road.

2. Project Design

a. Hardware Components and Interfacing Design

Hardware Components	Interfacing Design	Description	Details
LCD	Port A, Port C	Port C for LCD data; Port A for LCD control	Display the number of vehicles and the direction
LED	Port F	3 Pattern: 0b11000000, 0b00011000, 0b00000011	Display traffic light, light off for red, light on for green
Keypad	Port L	Connect PL7:4 to C3:0; Connect PL3:0 to R3:0	Press “*” for emergency; Press “*” again for quit
Push Button	Port D	INT0 for PB0; INT1 for PB1	PB0 for vehicle detected on east, PB1 for that on west

b. Code Structure

The code is divided into four sections:

(1) Initialize and Reset.

To make the code easy to understand and read, we use 'direction' to decide the direction the vehicles are allowed to pass. If direction = 0, the traffic light on the east side will turn on, and the vehicles on the road are going to the west. If direction = 1, the traffic light on the west side will turn on and vehicles on the road are going to east.

At the beginning, there is no car on both sides and the road. Thus the LCD screen is cleared. The traffic light on the east side is set green, which means the LEDs display the pattern '0b00000011', which represents a westbound traffic.

Since '*' locates at row 3, column 0, we set INITCOLMASK = 0xEF and INITROWMASK = 0x08 to scan '*'.

(2) INT0 and INT1 for PB0 and PB1 pressed.

INT0 and INT1 are set to detect falling edge, so ISCN0 = 0, ISCN1 = 1. When PB0 or PB1 pressed, a falling edge would be detected and then generate an interrupt. To enable the two interrupt, bit 0 and bit 1 for EIMSK must be set. In the two external interrupts, the number of vehicles would be updated (by increasing 1) and displayed on the left or right on LCD. After these routine, I bit in SREG should be set to enable the next interrupt.

(3) Timer and display the road (as a function).

TTCRnA is set '0x00' for applying Normal Mode. TTCRnB is set '0b00000011' to set Timer prescaler to /64.

In each Timer, the I bit in SREG would be reset to disable all maskable interrupts.

Timer0 is used for the interval time that two adjacent vehicles pass the road. In another word, after 0.5s, a new car could join the road continuous (i.e. from the same direction). Then the number of vehicles in the road should increase 1, and that on the east or west should decrease 1, according to the value of 'direction'. If direction = 0, the traffic light is set westbound, thus the number on the east side should decrease, else on the west side. Then the LCD display should be updated.

Timer1 is used to change the direction. If neither the east nor the west has vehicles, then the traffic light do not need to be changed. If only one side has vehicles, the traffic light on that side should turn green when the road is empty. If there is a congestion, then Timer1 would be applied to change the direction every 5 seconds, after the road being empty. The value of 'direction' is used here also, to decide how the traffic light should change.

Timer2 is used to decide if a car has passed the road. When passes, the number of vehicles on the road should decrease 1.

After each timer, the I Bit in SREG would be set again to enable the next interrupt and return.

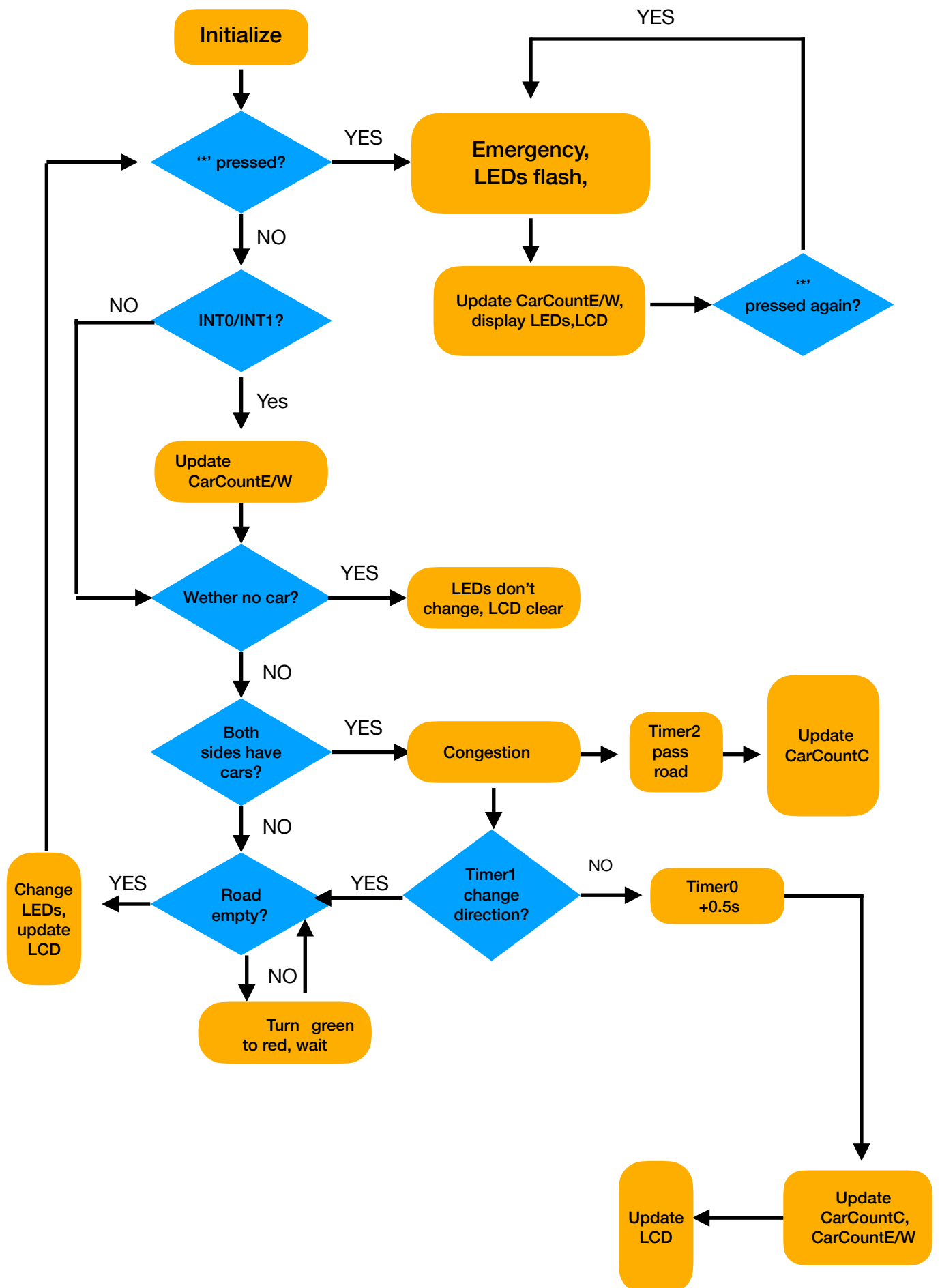
(4) Detect '*' pressed and Emergency (in 'main') .

Emergency is handled in the main function. As described in (1), by scanning row3 and column0 (*), if '*' is pressed, then jump to handle with emergency.

It should be noticed here that the Emergency actually holds higher priority than Timer0 and Timer2, since in emergency, no vehicle would be allowed to enter the road. This is achieved in the function 'emergency'. TIMSK0 and TIMSK2 are all set 0x00 to disable Timer interrupts. Then EmergencyFlag would be set to 1, which means the Emergency is not ended yet.

Then the LEDs would flash by using the macro 'flash'. Next, scan the keypad to see if '*' is pressed again. If not, the emergency function will be excused again (as a loop), else exit emergency, clear EmergencyFlag, and enable TIMSK0 and TIMSK2.

c. Execution Flow



3. Group Contribution

The source code for this project is the work of Bashayr Altulahi, Logan Li and me.

Name	Contribution	Details
Bashayr Altulayhi	RESET and Emergency	Initialization; Interfacing; Main and Emergency
Logan Li	INT0 and INT1	PB0 and PB1; LEDs display; CarCountEast/West display
Tong Zhang	Timers and General Logic	General logic of the project; LCD display of the road; Timers