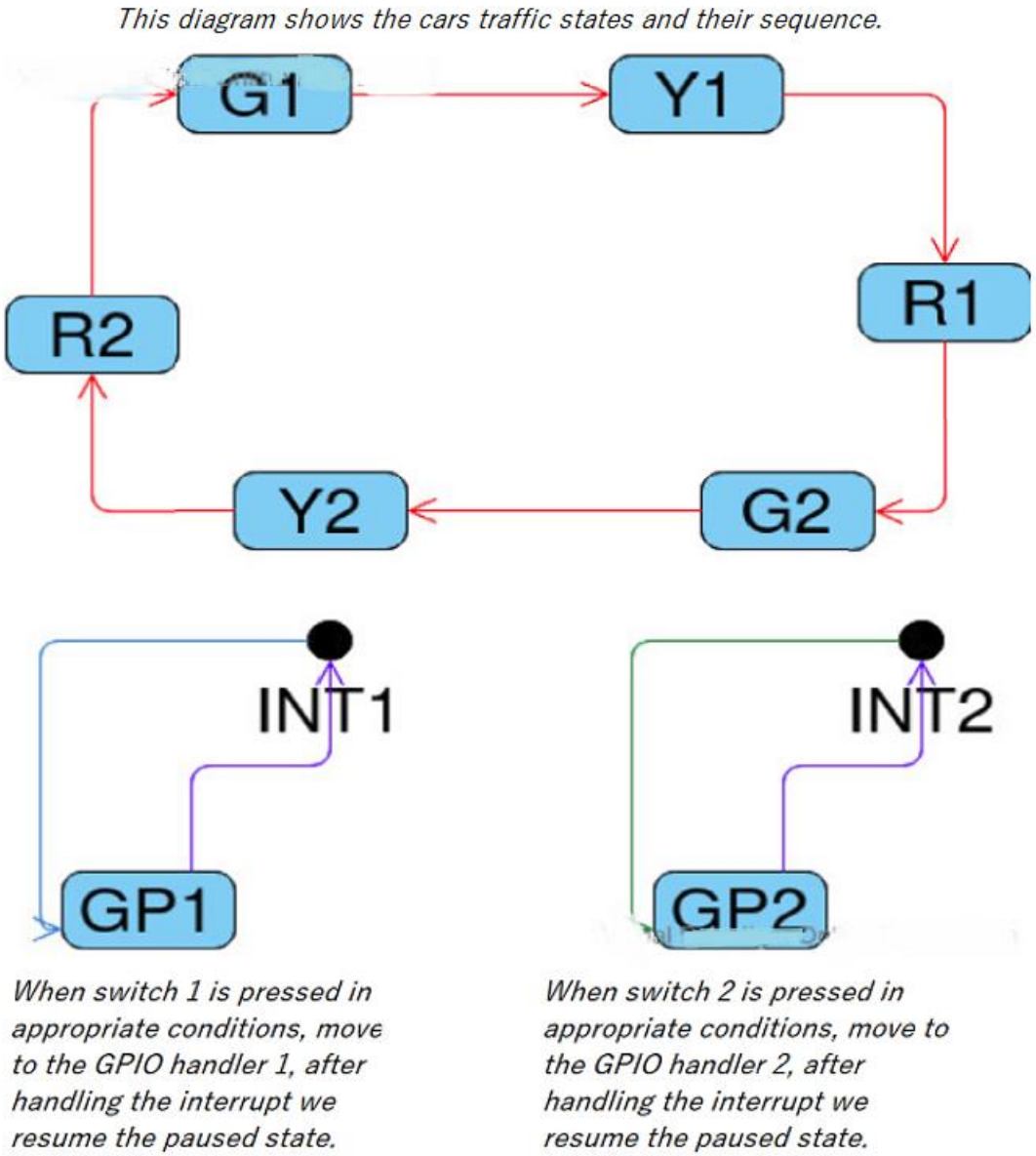
Traffic Light System

Code Flow and Description



- In the first state diagram it is shown that the 6 states of the two traffic lights have the same handler, once a state is done an interrupt happens to change to the next state considering the timeout for each state.

- In the pedestrian traffic light once we push the button a specific handler is called to pause the traffic light and turn it red while the pedestrian traffic light is turned green for 2 seconds, when the time is met an interrupt happens to resume the car traffic lights where it was paused.

Functions and Functionality

void init_portA(void)

Input
Non
Output
Non
Functionality
Initializes pedestrian traffic light

void init_portB(void)

Input
Non
Output
Non
Functionality
Initializes cars traffic light

void init_portF(void)

Input
Non
Output
Non
Functionality
Initializes pedestrian push buttons

void timer0_init()

Input
Non
Output
Non
Functionality
Initializes cars traffic light timer

void timer1_init()

Input
Non
Output
Non
Functionality
Initializes pedestrian traffic light timer

void update_state()

Input

Non

Output

Non

Functionality

If state is more than 5 (State when a pedestrian requested to pass)

1. Set traffic light 1 to red
2. Set traffic light 2 to red
3. Set pedestrian light 1 to green
4. Set pedestrian light 2 to green

Else

If the system is within case 0

1. Set traffic light 1 to green
2. Set traffic Light 2 to red
3. Set pedestrian ight 1 to red
4. Set pedestrian light 2 to green

If the system is within case 1

1. Set traffic light 1 to yellow
2. Set traffic Light 2 to red
3. Set pedestrian ight 1 to red
4. Set pedestrian light 2 to green

If the system is within case 2

1. Set traffic light 1 to red
2. Set traffic Light 2 to red
3. Set pedestrian ight 1 to red
4. Set pedestrian light 2 to red

If the system is within case 3

1. Set traffic light 1 to red
2. Set traffic Light 2 to green
3. Set pedestrian ight 1 to green
4. Set pedestrian light 2 to red

If the system is within case 4

1. Set traffic light 1 to red
2. Set traffic Light 2 to yellow
3. Set pedestrian ight 1 to green
4. Set pedestrian light 2 to red

If the system is within case 5

1. Set traffic light 1 to red
2. Set traffic Light 2 to red
3. Set pedestrian ight 1 to red
4. Set pedestrian light 2 to red

void timer0_set(int ms, void (*callback)(void))

Input

ms : an integer indicating time in millisecond to be used in delay
callback : pointer to the function invoked in case of an interrupt

Output

Non

Functionality

Sets the reload register of timer0 with a specific number of ticks

void timer0_callback()

Input

Non

Output

Non

Functionality

1. Update the state
2. Delays timer_0 specific duration in milliseconds according to the current state

void gpio_handler()

Input

Non

Output

Non

Functionality

1. Checks which button pressed then call its handler
2. Clears the interrupt flag

void gpio1_handler()

Input

Non

Output

Non

Functionality

If the state within 0 : 2 range

1. Prevent handler from accepting push button request
2. Set traffic light 1 to red
3. Set traffic light 2 to red
4. Set pedestrian light 1 to green
5. Set pedestrian light 2 to green
6. Pauses car traffic light routines for 2 seconds

void gpio2_handler()

Input

Non

Output

Non

Functionality

If the state within 3 : 5 range

1. Prevent handler from accepting push button request
2. Set traffic light 1 to red
3. Set traffic light 2 to red
4. Set pedestrian light 1 to green
5. Set pedestrian light 2 to green
6. Pauses car traffic light routines for 2 seconds

void timer1_set(int ms, void (*callback)(void))

Input

ms : an integer indicating time in millisecond to be used in delay
Callback : pointer to the function invoked in case of an interrupt

Output

Non

Functionality

Sets the reload register of timer1 with a specific number of ticks

void timer1_callback()

Input

Non

Output

Non

Functionality

1. Updates the state when interrupt happened
2. Clears the flag
3. Enables timer0 again
4. Handles the case to delay the request of the pedestrian to cross if the button was pressed after 1 second from the end of the Period of pedestrian crossing

void one_sec_delay()

Input

Non

Output

Non

Functionality

1. Clears the flag
2. Allows handler from accepting push button request

int ticks_from_ms(int ms)

Input

ms : an integer indicating time in millisecond to be used in delay

Output

The number of ticks

Functionality

Changes milliseconds into ticks