MICRO-CONTROLLER BASED OPTIMIZED TRAFFIC CONTROLLER

COURSE: MICROCONTROLLERS FOR MECHATRONICS
MECH 6621

PRESENTED BY:
BHAVIKA JAGIASI
(9713336)
SHIVAM NIGAM
(9573437)

PROFESSOR:
DR YOUMIN ZHANG
The project aims to provide following functions:

- A high priority is given to the highway traffic

- Service the by-pass(or service road) according to its traffic (number of cars)

- Notify the cars on highway for smooth or congested traffic, depending on traffic on service road on LCD

- Check the speed of car on highway and flash the photo-radar if it exceeds speed limit

- Control a railway barrier to stop traffic on service road when a train is coming regardless of other traffic conditions
- Highway road light
- Service road light
- Car on service road
- LCD display
- Car exceed speed limit
- Railway barrier
FUNCTIONING
- Highway road light = Green
- Service road light = Red
- Car on service road = No
- LCD display = -
- Car exceed speed limit = -
- Railway barrier = Up
1. CAR ON SERVICE ROAD
Highway road light = Yellow
Service road light = Yellow
Car on service road = Yes
LCD display = Free flow
Car exceed speed limit = -
Railway barrier = Up
2. WAIT FOR CAR TO CROSS
- Highway road light = Red
- Service road light = Green
- Car on service road = Yes
- LCD display = Free flow
- Car exceed speed limit = -
- Railway barrier = Up
3. MORE THAN 2 CARS ON SERVICE ROAD
- Highway road light = Red
- Service road light = Green
- Car on service road = Yes
- LCD display = Congestion
- Car exceed speed limit = -
- Railway barrier = Up
4. CAR ON HIGHWAY
- Highway road light = Green
- Service road light = Red
- Car on service road = No
- LCD display = -
- Car exceed speed limit = No
- Railway barrier = Up
5. CAR EXCEEDS SPEED LIMIT
- Highway road light = Green
- Service road light = Red
- Car on service road = No
- LCD display = -
- Car exceed speed limit = Yes
- Railway barrier = Up
6. RAILWAY BARRIER
- Highway road light = Green
- Service road light = Red
- Car on service road = Does not matter
- LCD display = -
- Car exceed speed limit = -
- Railway barrier = Down
HARDWARE INTERFACING
LED FOR CONGESTION DETECTION

FOR RAILWAY BARRIER
SERVO MOTOR INTERFACING
IR SENSOR INTERFACING
IR TRANSMITTER & RECEIVER
SOFTWARE DESCRIPTION
We have used the following high priority interrupts:

<table>
<thead>
<tr>
<th>Interrupt</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT0</td>
<td>To turn on motor</td>
</tr>
<tr>
<td>INT1</td>
<td>To generate PWM</td>
</tr>
<tr>
<td>INT2</td>
<td>To detect speed in speedometer</td>
</tr>
<tr>
<td>TMR0</td>
<td>To calculate speed of car in speedometer</td>
</tr>
<tr>
<td>TMR1</td>
<td>To detect the traffic on service road</td>
</tr>
<tr>
<td>TMR2</td>
<td>To calculate number of cars on service road</td>
</tr>
</tbody>
</table>

Also, TMR2 is used as a low priority interrupt:

- INT0: To turn on motor
- TMR1: To generate PWM
- INT1: To detect speed in speedometer
- TMR2: To calculate speed of car in speedometer
- INT2: To detect the traffic on service road
- TMR0: To calculate number of cars on service road
CONGESTION DETECTION

INT0 SUBROUTINE

Wait for interrupt, car comes

WREG = 1

Is \( N_{\text{CAR}} > 0 \) ?

Load timer0 with 0x1200h

Increment \( N_{\text{CAR}} \)

Stop

YES

NO

Clear timer0 flag

Set bit 0 of FLAG_TIMER

Stop

TIMER0 SUBROUTINE

TIMER0 is used as a 16 bit timer, detection on rising edge with 1:64 pre-scale. It generates a delay of 2 sec.
TIMER0 = 2 SEC
CONGESTION DETECTION (CONT'D.)

SUBROUTINE FOR TRAFFIC LIGHT DISPLAY

Start

Clear bit 0 of FLAG_TIMER

WREG = 2

Is \( N_{\text{CAR}} > 2 \)?

YES

Display “FREE FLOW” on LCD

Display “CONGESTION” on LCD

Both signals go to YELLOW, call DELAY

Service road = green, highway = red

Delay = \( N_{\text{CAR}} \times 4.9 \) sec

Both signals go to YELLOW, call DELAY

Service road = red, highway = green

Stop
DELAY OF 4.9 SEC

C:\\project_p1(2).asm

DELAY2:
- MOV LW D'1'
- MOVWF C1
- MOV LW D'200'
- MOVWF C2
- MOV LW D'SD'
- MOVWF C3

DELAY1:
- DECISZ C1,1
- GOTO DELAY1
- DECISZ C2,1
- GOTO DELAY1
- DECISZ C3,1
- GOTO DELAY1
- RETURN

SUBROUTINE FOR SPEEDOMETER

INT1_ISR:
- RETURN

; INITIATING LCD
LCD_SETUP: MOV LW D'150' ; COUNT FOR 1.5 MS DELAY
- MOVWT DELAY_MSG

Stopwatch
- Synch
- Instruction Cycles 9625015
- Total Simulated 9625027
- Zero
- Time (Secs) 4.912508
- 4.912514
- Processor Frequency (MHz) 8.000000
CONGESTION DETECTED
Initialize Timer1 as high priority, 16-bit timer, rising edge, 1:2 prescale and TMR1H= 255, TMR1L = 156.

Initialize variable counters, COUNT_1 = 1, VAL_PWM = 0, REP1 = 50, REP2 = 255.

Wait for interrupt

Is TMR1IF = 1?

YES

Clear flag and stop timer.

NO

Is COUNT_1 = 1?

YES

Is COUNT_1 = 2 OR 3?

YES

Is COUNT_1 = 4?

YES

Reset COUNT_1 = 1

NO

Reload timer with D’156’ and start.

RETFIE FAST
TIMER1 INTERRUPTS AT EVERY 0.1MS

BSF PORTA, 2
BSF PORTA, 3
BSF PORTA, 4
;DISABLE ALL OTHER INTERRUPTS AND TIMERS
BCF INTCON3, INT1IE
BCF INTCON3, INT2IE
BCF INTCON, INT0IE
BCF PIE1, TMR2IE
BCF PIE3, TMR8IE
BSF T1CON, 0 ;ENABLE TIMER
BSF FLAG_TIMER, 1
RETIE FAST

TMR1_ISR: BCF PIE1, TMR1IF
BCF T1CON, 0 ;STOP TIMER
BITSET: BITG PORTA, 0 ;TOGGLE LED RA0

MOVW D'1'
CPFSEQ COUNT_1
GOTC NXT
GOTO PWM_M
NXT: BITG PORTA, 0
MOVW D'2'
CPFSEQ COUNT_1
GOTC NXT2
GOTO PWM_R

Stopwatch
Synch Instruction Cycles: 211 335
Zero Time (uSecs): 105.500000 167.500000
Processor Frequency (MHz): 8.000000

Asynch Pin / Register Actions Advanced Pin / Register Clock Stimulus Register Injection Other

Fire Pin / SFR Action Width Units Comments / Message
> INTO Set High
> INTO Set Low
Pulse width = 1.5ms
Load WREG = 154, REP2 = 255

Is VAL_PWM > WREG (154)?

NO

RC2 = low, increment VAL_PWM

YES

Load WREG = 170

Is VAL_PWM > WREG (170)?

NO

RC2 = high, increment VAL_PWM

YES

Clear VAL_PWM

Decrement REP

Is REP = 0?

NO

YES

Reset REP2 = 255 and Increment COUNT_1

Pulse width = 2ms
Start

Load REG = 2

Is INT1IF = 1?

Decrement REG

Is REG = 0?

Stop the timer 2

Start timer 2

Is TMR2 IF = 1?

Increment REG14

Load value of TMR2 in TMR_VAL

Is TMR_VAL > 55?

PHOTO-RADAR ON

Stop

Copy the value of TMR2 in TMR_VAL

Copy the value of REG14 in REG12

Time = TMR_VAL * REG12 * 4 * prescale * postscale * 125ns

Speed = Distance / Time

Stop
**DISCUSSION**

- It was our effort to combine maximum number of techniques that we learned in this course.

- It was challenging to use 6 interrupts together in one program, but it was best to give all but one the high priority. We learned:
  - It is very important to clear flags immediately after entering in the subroutine.
  - It is good to keep like subroutines nearby.
  - If required, all other interrupts may be disabled when inside in a major subroutine.

- We also learned that each motor is different, and unlike other subroutines, the same PWM generation program may not be used with all motors. It is best to test PWM generation using LOGIC ANALYZER as the values change for every code depending on number of instruction cycles.

- Also, the synchronization between two interrupt in case of speedometer was difficult to achieve, which we rectified by using TMR2 in low priority.

- We initially planned to use LCD as explained, but due to certain synchronization errors, we used LEDs to display our results.
  - A bit change at RB7 can generate a high priority interrupt
  - If disabled, it does not detect the corresponding pin during LCD functioning
  - We were unable to overcome this error
Our project provides various options but can be improved further more:

- IR sensors are not powerful, hence other stronger types of sensors would be required for real time application.

- We assumed that the highway has at a time only one car. For a regular 6 lane highway, we need to have 6 sensors, one for each lane.

- If a car is too slow, it might over run the timer values, hence detect very high speed instead of too slow.

- If the sensors are too close, or there are IR reflectors, or high voltage wires near the receiver, it might interfere.

- If traffic load is high on service road, the controller will work in the same way as a regular controller.

- The speed which is retrieved from our speedometer is not very accurate as there is no resource for calibration.
This kind of a controller is already implemented in a lot places. The most popular one being a photo-radar.

Although the technique of counting car on service road is not very popular, yet a similar approach is still used in controller of trains.

Also, the speed detection which is done in our program is not very popular either.

All these techniques are easy to implement with very simple and user friendly integrations.

The incoming traffic is optimized, in terms of saving time for traffic on highway.

The vehicles become more fuel efficient too.

Photo-radar can (and does too, to) help prevent accidents due to speeding and help regulate traffic law enforcement on roads, specially the ones which may not be easily monitored by physical presence.

Congestion detection technique can help direct the route on highway through a secondary route, to prevent further congestion.
CONCLUSION

- The project is a very efficient way of optimizing traffic, with redefinition of threshold values for a real time application.

- The project works to control traffic on both roads according to traffic on service road, as well as indicate congestion for more than 2 cars.

- It also works to put a barrier down for an incoming train.

- Also, we were able to detect a speeding car.
THANK YOU