# Design and Implementation of a Vehicle Parking Monitoring System

Preshen Goobiah, 1355880

School of Electrical and Information Engineering,
University of the Witwatersrand, Johannesburg 2050, South Africa

***Abstract*—This report documents the design and implementation of a vehicle parking monitoring system using a ATMEGA328p microcontroller which computed the various algorithms necessary. The programming (in the Assembly language) of the microcontroller, done using Atmel®Studio 7, was used in conjunction with the appropriate hardware which demonstrated the number of vehicles in a parking space.**

***Index Terms*—Light Emitting Diode (LED), Red, Green and Blue (RGB), Pulse Width Modulation (PWM), Binary Coded Decimal (BCD)**

## I. Introduction

Finding parking in a shopping mall or any parkade, is a problem that needs to be addressed. There exists solutions to this, such as the LED occupancy sensors[4], which are widely implemented in South Africa, where the commuter visually searches for a sensor that is green which indicates that the particular parking bay is free. This is highly inefficient as there may be no free parking bays, causing the commuter to be delayed in his/her travels. A software controlled entering/exiting system alleviates the problem of entering a parking lot that is full. The aforementioned microcontroller executes the algorithms necessary; with the integration of TCST1103 optocouplers, a 7 segment display to display the number of cars parked, an RGB LED to indicate whether a vehicle should enter and other hardware components - solves this inefficiency. This system can be implemented in conjunction with existing solutions, to ease parking problems.

## II. Success Criteria, Assumptions and Constraints

### A. Success Criteria

TCST1103 optocouplers were specified to be used as sensors to detect vehicles from their way of passage (entrance or exit). At the entrance, a RGB LED indicates whether a vehicle should enter the premise; orange for default, green for entry and red for no entry (premise is at capacity). A calculation was to be made to count the number of vehicles in the parking lot and display the result on a 7 segment display. Upon entry/exit a sound must be generated, varying the intensity of the sound - corresponding with the number of vehicles present. The sound intensity is to be varied using the PWM function of the microcontroller.

### B. Assumptions

All hardware components exhibit behavior according to manufacturer specifications and mixing the colors red and green will give required orange.

### C. Constraints

The coding of the microcontroller was restricted to AVR Assembly language.

## III. Design and Methodology

### A. Sleep mode

Microcontroller was set to sleep in idle mode which stops the CPU to save power until an activation condition (a vehicle triggers the entrance/exit sensor) is met, consequently an external interrupt occurs.

### B. External interrupts

In order to implement the system, the external interrupt function of the microcontroller was used. It was chosen instead of checking the output voltage of the optocoupler at specified time intervals (polling) as vehicles may enter or exit after long periods of time. It was setup by enabling global interrupts, subsequently the INT0 (used for entrance) and INT1 (used for exit) bits in EIMRSK register. The respective interrupt handlers were set to trigger on rising (voltage LOW to HIGH) by enabling ISC00 and ISC01 bits in the EICRA register. This configuration of triggering was used to avoid triggering a single interrupt twice on a single instance. Additionally, a delay of 9 milliseconds was added to cater for debouncing.

*1) Optocoupler (TCST1103):* The TCST1103 optocouplers (gap sensors) facilitated the external interrupts. The output of the sensors were connected to PORTD1 and PORTD2, which are the fixed ports for the two external interrupts [5]. An Analogue to Digital conversion (ADC) was not chosen to be used for the input signal from the gap sensor. The hardware provided a sufficient HIGH voltage ($\approx 1.25$ V) for an external interrupt to occur [1] The emitter and detector of the optocoupler were protected by $150\ \Omega$ and $10\ k\Omega$ resistors.

### C. PWM

PWM is completely hardware controlled in the ATMEGA328p. This function of the microcontroller was utilized to generate a wave at a particular frequency and vary the sound intensity of the wave. Timer 0 was used (in PWM mode), which is an 8 bit register, as we do not need a high resolution for speaker sound levels. In order to simplify the sound generation/variation the Fast PWM mode was chosen. It was setup by enabling the WGM02, WGM01 and WGM00 bits in the TCCR0A and TCCR0B register. The timer clock frequency was set to a prescaling value of $256$ by enabling the CS02 bit in TCCR0B. COM0B1 was enabled to clear the OC0B pin on a compare match (used in duty cycle calculation), A value of $40$ was loaded into the OCR0A register therefore a waveform was generated with a frequency of a $1524\ Hz$ (see equation 1). The duty cycle, which defines the time ratio of PORTD5 being ON and OFF - has a perceived decrease/increase in output voltage, consequently sound intensity varies - was defined by the value of OCR0B. In order to achieve the goal of varying the sound intensity as the number of vehicles increase/decrease. A simple mathematical manipulation was made to the value of OCR0B i.e. the duty cycle, with the number of vehicles present - on every interrupt. The corresponding "analogous" voltage was sent to PORTD5 (see equation 2).

$$PWM_{Frequency} = \frac{\frac{CPUClock_{Frequency}}{Timer_{prescaler}}}{OCR0A + 1} \tag{1}$$

$$V_{out} = \frac{DutyCycle}{255} * V_{dd} \tag{2}$$

*1) LM386 and Speaker:* The oscillations on the speaker cone correspond to frequency of the wave generated which produce a sound. This sound of $1524\ Hz$ was output at a voltage varying from 0 to $V_{dd}$ (defined by the duty cycle). By introducing the LM386 amplifier to the circuit there was a default gain of 20 on the output voltage [3], which made the audio levels distinct.

### D. Visual Output details

*1) 7 Segment Display:* A common anode 7 segment display was used, which sets a single LED segment to ON if the input voltage is LOW. It was protected by a 220 Ω resistor as per $R = \frac{V_{dd} - V_{LED}}{I_{LED}}$. The number of vehicles present was illustrated by - using the 16 bit Z pointer register, which pointed to binary representations of the numbers 0-9; corresponding to LED segments a-g on the 7 segment display. This was done by using the '.db' directive. A problem arose due to there not being 7 pins on a particular PORT available to use, due to individual ports being restricted for the PWM and external interrupt functions, which resulted in a binary value being split up amongst ports. PORTC5:0 and PORTB4 was used to illuminate the segments of the display. Fig. 2

*2) RGB LED:* Common anode RGB was used. The colors were achieved by enabling/disabling PORTB5 - connected to red cathode and PORTB2 - connected to green cathode. The orange color was achieved by placing a $10\ k\Omega$ resistor on green cathode to limit the green light when mixing with red light which protected by a $220\ \Omega$ resistor as seen in Fig. 1. Table I shows the outputs on the RGB LED provided a status/event in the parking lot has occurred.

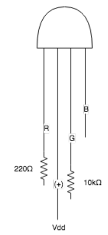


Fig. 1. RGB LED with resistors.

TABLE I
STATUS/EVENT AND OUTPUT ON RGB LED

| Status/Event | Red | Green | Blue |
|---|---|---|---|
| Parking Empty | ON | ON | OFF |
| Vehicle approaches entrance (Vehicle count less than 9) | OFF | ON | OFF |
| Vehicle approaches entrance (Vehicle count less is 9) | ON | OFF | OFF |
| Parking full | ON | OFF | OFF |
| Vehicle exits | ON | ON | OFF |

### E. Overall

An overall illustration of the setup is shown in Fig. 2

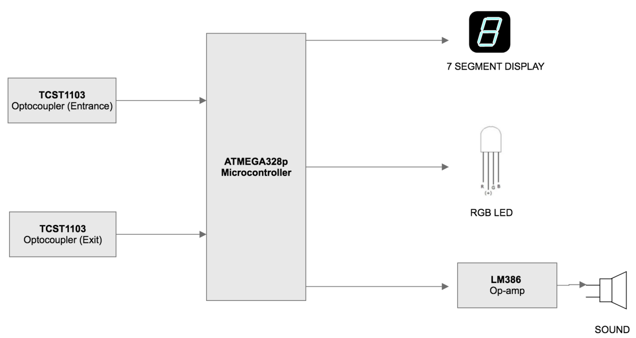


Fig. 2. Block Diagram showing setup.

## IV. RESULTS

The algorithms, implemented through external interrupts, efficiently performed the stipulated project requirements. The increment and decrement of the 7 segment display worked seamlessly with no visible flicker, although there was a 1 second delay between subsequent vehicle entries/exits i.e. interrupts. The sound generated did increase/decrease in intensity, although a distinct difference between vehicle five to vehicle nine (exit and entry) was not heard. There were glitches found when the 'sbi' instruction was called on PORTB5, this was caused by leakage current in the breadboard used. A high frequency tone was heard on the speaker, possibly due to noise. The system which had to be coded in assembly restricted the implementation of streamlined/extra functionality due to time constraints and limited resources available. Initially 3 regular LED's were used, as a misinterpretation of the project requirements but due to the modularity of the assembly code, it was easy to change to a single RGB LED.

## V. Design Tradeoffs

### A. 7 Segment Display

The 7 segment display restricted the system to track only 9 vehicles which is not practical, in turn the sound intensity fell into a minute range which aided in implementing the system.

### B. External Interrupts

Due to the microcontroller having the capability of a maximum of two external interrupts - only two optocouplers could be connected at any given time, at the same time it made implementation/coding easier.

## VI. Possible Improvements

### A. Low Pass Filter

Introducing a low pass RC filter for the output voltage signal, sent the amplifier, will attenuate the high frequency noise as mentioned in IV.

### B. Increasing sound-loudness capacity

Only a slight difference in the sound intensity can be heard when the number of vehicles increase/decrease. A solution to this problem would be to add a 10 $\mu F$ capacitor between pins 1 and 8 of the LM386 amplifier to increase the gain to 200 [3]. Additionally, we can increase the number of speakers connected to the amplifier.

### C. Additional optocoupler

The system cannot precisely track the number of vehicles in the parking, in the event of a vehicle approaching the sensor but not entering.A possible improvement to the design would be to add an additional optocoupler at the entrance. The first optocoupler would be for a vehicle that approaches - to change the colour of the the RGB LED. The second optocoupler would be for a vehicle that enters the parking - to increment the vehicle count.

### D. BCD 7 Segment decoder

The system can include the use of a BCD decoder to convert the binary values of the vehicle count to display on the 7 segment. This would streamline the code and remove excess data from the microcontroller's SRAM.

## VII. Conclusion

The integration between external interrupts and the optocoupler were simple. The most challenging part of coding the system, included; using the PWM function (to produce a wave and alter it) coherently with the counting segment of the code and displaying on the 7 segment without seven ports being freely available to use. The overall project was challenging as many errors such as debouncing, glitches and stack pointer arose. The demonstration of the system should be to the satisfaction of the marker as all project requirements were fulfilled.

## References

[1] Vishay. (2017). TCST1103 Datasheet. [online] Available at: `www.vishay.com/docs/83764/tcst1103.pdf` [Accessed 19 Sep. 2017].
[2] Arduino. (2017). The use of PWM in Arduino. [online] Available at: `www.arduino.cc/en/Tutorial/PWM` [Accessed 17 Sep. 2017].
[3] Texas Instruments. (2017). LM386 Datasheet. [online] Available at: `https://goo.gl/5ym5q7` [Accessed 19 Sep. 2017].
[4] Kinzey, B., Royer, M., Myer, M. and Sullivan, G. (2017). The Use of LED occupancy sensors. [online] pnnl.gov. Available at: `www.goo.gl/cf5Uq7` [Accessed 19 Sep. 2017].
[5] Atmel. (2017). ATmega328p Datasheet. [online] Available at: `www.goo.gl/i2CpoA` [Accessed 19 Sep. 2017].