URSI GASS 2020, Rome, Italy, 29 August - 5 September 2020

A novel cost-effective Pressure Sensor based Smart Car park system

Pooja R Khanna [(1)], Gareth Howells* [(1)]
(1) University of Kent, Canterbury, CT2 7NZ

Abstract

With the increase in number of people using vehicles for transportation since last decade, traffic congestion is a major problem that requires to be solved effectively. Smart car park system is considered as one of the strategic solutions to this problem, which involves use of sensors to collect data. This paper proposes a novel low-cost smart car monitoring system to detect number of incoming and outgoing cars in and/or out of the car park using pressure sensors. This system also provides the data for the number of spaces available in the car park. Additionally, the paper also demonstrates the algorithm used to process the data obtained by the sensors to use it as a useful information.

1 Introduction

Increase in usage of vehicles is causing major traffic problems in major cities, which in-turn creates parking issues. Finding a free parking space has become a time consuming task at peak times of the day. A case where driver is not aware of the number of parking spaces available in the car park makes the traffic congestion even worse. Therefore, there are several solutions existing in the market that can provide data showing the number of cars available in the car park. Furthermore, it is significantly important to keep track of number of incoming and outgoing cars from the car park to ensure safety in an area [1]. The most commonly used system for this purpose involves installing cameras at car parks that detects the incoming and outgoing cars in the car park and thus displaying the number on the screen. These systems solely depend on the quality of camera installed, contrast and brightness of the picture, which can create room for errors. Therefore, the installation, implementation and maintenance of such a system is complex and expensive. Another solution is to install sensors at each of the parking spaces in the car park. Thus, these sensors will detect if the cars are parked at that spot and then update the system with the counts of overall available spaces in the car park [2]. Most of the car parks have more than 100 spaces and therefore, the installation and management of such a large number of sensors becomes complex too. Therefore, it is necessary to develop smarter as well as cost-efficient parking systems that are less complex and easier to maintain and install.

This paper proposes a cost-efficient and robust, Arduino and raspberry pi-based Car Park system to count the number of cars entering and exiting the car park. The data regarding the car count collected by the sensors is sent via radio using nRF24l01+ chip. The purpose of the system is to get a sense of traffic, understand the peak times in a day, and become a novel low cost solution to this problem. The pressure sensors are installed at the entries and/or exits of the car park along with a transmitter module. The transmitted data from each of the entries and/or exits of the car park is received by a single receiver module. Therefore, the receiver acts as a point of collection of data from all the transmitters installed at entries and exit points of the car park. The proposed car park system is tested and implemented at a large industrial site. Furthermore, this system is energy-efficient and consumes relatively lower power compared to other existing solutions in the market. Thus, changing of batteries is less frequent and thus contributes to easier maintenance of all the installed modules.

2 Related Work

A lot of research has been done to develop a car park system. Presently, the technology related with the smart parking system is based on the amalgamation of embedded modules, Wireless Sensor Network (WSN's), Sensor nodes and a cloud based system to store the data. Reference [3] represents an Arduino based car park system with an infrared (IR) sensors that checks the spaces available and sends the data to the server. A similar system described in [4] uses the Radio-Frequency Identification (RFID) tags for authorization, which connects the barricade to a motor to open the car park for the authorized personnel only if there is a space available. This increases the maintenance cost of the system, for instance, if a system is deployed at a larger scale, then, it would be required to issue large number of RFID tags for every new car entering the car park. The bigger issue would be maintaining and tracking of the cards if lost or if it passes in the wrong hands.

A cloud-based technology is used in [5], monitoring is done with the help of overhead passive infrared (PIR) and ultrasonic sensors at each slot in the car park. Each ultrasonic sensor is connected wirelessly to the Raspberry Pi using ESP8266 chip. The mobile app is used to book available spots in the park. This increases the cost of the whole system significantly as each of the sensor is connected individually with an ESP8266 and harder to maintain. A system proposed in [6] presents a similar low cost monitoring system, which uses three ultrasonic sensors (HC-SR04 one for each parking space) with one NodeMCU (esp8266) and connected to Raspberry Pi3. The Pi3 module collects all the data, then sends it to cloud every 30 secs, and updates the database. A User Interface is available to book the slots. This system in outdoors would be hard to maintain and will still be costlier. Since 40% of the traffic is due to cars roaming in a particular area to search for a parking space, a system proposed in [7] suggests a reservation-based parking system to avoid the congestion. The booking system is done via an application, which has GUI to view the free spots, and GPS is used to find the nearest car park. In this case, the system is designed only for Android cell users, excluding the Apple iOS users.

Another cloud-based system is presented in [8], an Internet of Things (IoT) based network architecture, proposes a system based on performance metrics based on the least cost to find a parking spot. RFID technology is used in the car park, creates a WSN, each node sends the real-time data. The design consists of two RFID readers on the entry and exit respectively to get the percentage of free spaces, nearest possible space, map of the local car park and authentication of the cars on site. These RFID readers are connected to the Arduino, which controls the door and sends the data to the Web-server. The paper [9] implements an Intelligent Smart Parking Algorithm (ISPA). The design includes various sensors such as CMOS, Ultrasonic, OCR (Optical Character Recognition) and Speed sensors to create a hybrid parking mechanism. These are used for number plate identification, Vehicle detection, OCR working softwares and speed detection respectively. The system uses Raspberry Pi and Arduino (to control motor) to combine all the sensors described. It registers the users to the system to allocate the parking space based on the source and destination provided by them. They also provide the smart card that is used by the user to provide details of the smart card once he is outside the vehicle. This card contains the barcode, which stores all the information of the user. This is a very tedious cycle to update the smart card information every time a car enters a carpark especially during peak times.

3 System Architecture

Any smart parking system consists of three basic components i.e. Sensors (to accumulate the data), Microcontroller (for processing) and Internet connection (so the data is easily accessible to authenticated users).

Sensors: This design uses the Differential Pressure sensors, MPX5010DP as shown in Figure 1. These sensors are piezoresistive transducers are the analog sensors, used along with microcontroller. These sensors combine thin film metallization, advanced micromachining techniques and bipolar processing to provide the analog output signal, which is directly proportional to the pressure applied.

Figure 1 Differential Pressure Sensor MPX5010DP [10]

The microcontroller board used here is Arduino UNO with Atmega328P. It has 14 digital I/O pins, 6 analog pins that are used to connect two differential sensors. This board is used at the transmitter end with the NRF24L01 chip with PA and LNA circuit on board used for radio communication to transmit/receive data. The NRF24L01 RF module is a 2.4 GHz transceiver module (as shown in Figure 2) with an onboard antenna used for long-range transmission [11].

Figure 2 NRF24l01 transceiver module [12]

The second nRF24l01 module used at the receiver end in combination the Adafruit Feather Huzzah module (as shown in Figure 3) to receive data from the transmitter module. The microcontroller has the on-board ESP8266 WiFi module.

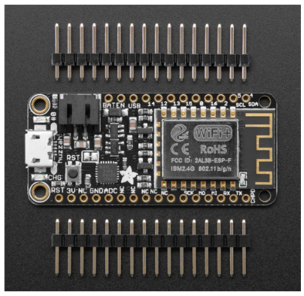

Figure 3 Adafruit Huzzah Feather [13]

Raspberry Pi: The system uses Pi2 with Node-Red to send the data to the cloud. Node-Red programming is done to save the data to Raspberry Pi, if ever the WiFi goes down, the data is stored locally.

4 Working

The low-cost smart parking system comprises of two transmitters and one receiver. The transmitters send the data to a single receiver with a unique number to understand which transmitter the data is coming from. Each transmitter consists of two pressure sensors MPX5010DP with two tubes attached to it, which changes the pressure readings when a car goes over it. The car park does not have a definite entry/exit point. So, the openings can be used either as entry or exit point hence, making the algorithm accordingly to detect the incoming and outgoing cars. This is the novel way of detecting incoming/outgoing and counting cars to know available/used parking spaces. Every time a car passes over, the transmitter sends the data that is an array of four numbers, which has identification number of the transmitter, incoming car or outgoing car and lastly the number of triggers (which gives insight to the cars entering straight or at an angle) with a hashtag to separate the consecutive payloads. This data is then combined at raspberry pi to count the spaces left. This data is logged onto a feed with three different streams showing the total incoming, outgoing and no. of spaces available. Figure 4 shows the layout of the car park where the sensors are placed at the entry and/or exit points. These modules are placed in a waterproof box.

Figure 4 Layout of the car park

The two entry/exit points are where the car park sensors are placed with the tubes glued to the ground across the opening, making the installation easier.

The figures 5 and 6 show the prototype of the whole system i.e. the transmitter (which is same for both the entrances) and a receiver with the Raspberry Pi.

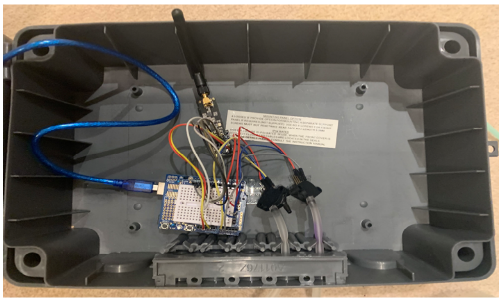

Figure 5 Transmitter End of the system

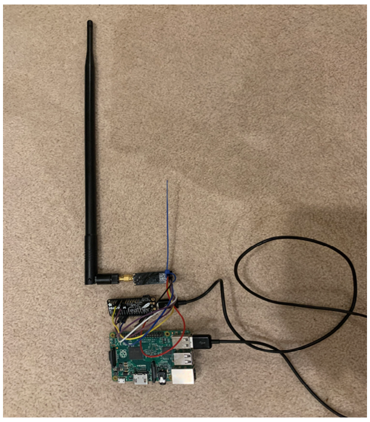

Figure 6 Receiver End of the system

5 Algorithm

The algorithm shows the harmonious working of transmitter and receiver. The Sensing side sends the trigger to the receiver when a car is detected. This car can either be coming in/ going out which is decided on the transmitter edge microcontroller and sends the data accordingly. The first part of the algorithm is executable on the Arduino at the transmitter end of the prototype. The Arduino waits for 1.5 seconds after the car goes over to get total number of triggers. This step is important because a car if enters the carpark at an angle, each wheel would give a separate trigger. The second scenario is when the car does not enter at an angle so the front two wheels and back two wheels apply the pressure at the same time, hence getting two triggers. The third scenario, when a car enters at an angle and then becomes straight or vice-versa, number of triggers would be three. All of this processing is done on the Transmitter end. After the car is detected, with the help of RF module used here, the data is sent to the receiver immediately. The receiver is connected with the raspberry Pi through USB connection to send the data on the serial port. The raspberry Pi executes a Node-Red program to segregate the data to know the available spaces in a car park. Figure 7 shows the algorithm in the form of a flowchart.

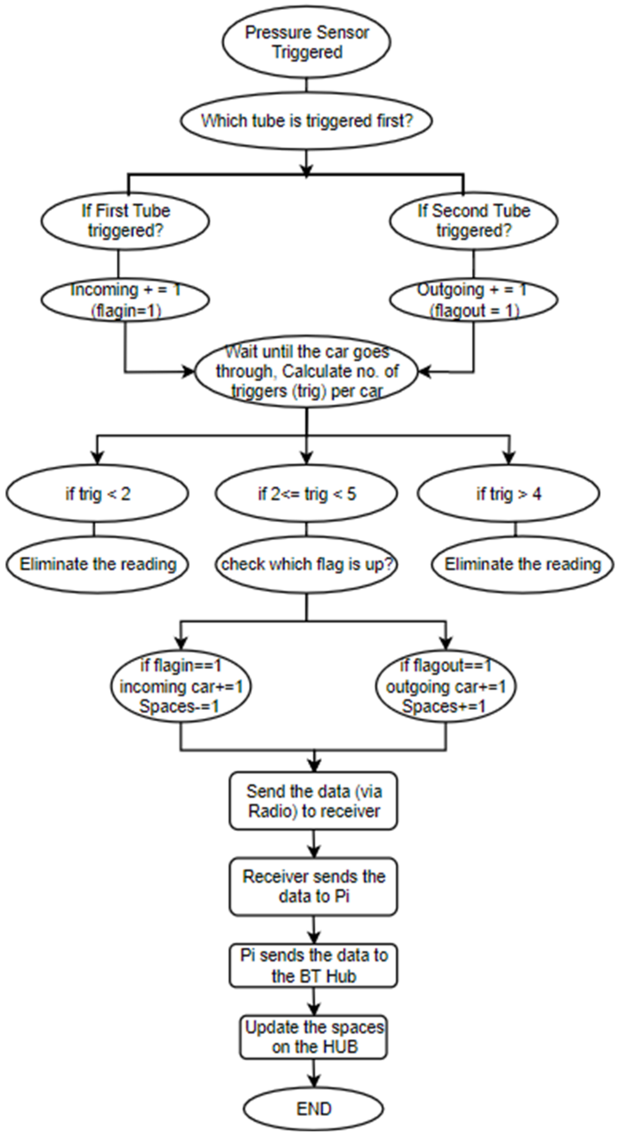

Figure 7 Algorithm Design

6 Results

This section shows the data hub showing the different streams for the incoming and outgoing. To access the values, we need to subscribe to the data feed to get the real time values. The Figure 8 shows the picture of the data feed and Figure 9 values stored in each stream.

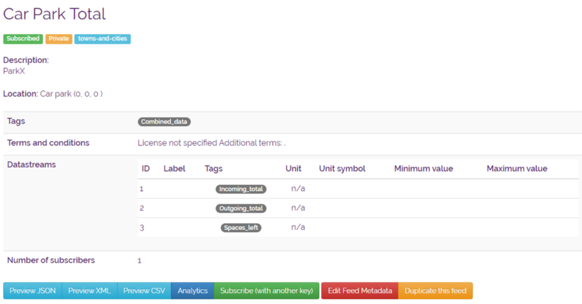

Figure 8 Subscribed car park feed


```
▼<feed id="a4593bf4-d9b3-45d4-a233-83ef15112e13">
   <creator>pooja</creator>
   <updated>Thu, 30 Jan 2020 13:53:37 GMT</updated>
   <title>Car Park Total</title>
   <url>/feeds/a4593bf4-d9b3-45d4-a233-83ef15112e13</url>
   <private>true</private>
   <description>ParkX</description>
   <tag>Combined_data</tag>
 ▼<location>
     <name>Car park</name>
     <lon>0</lon>
     <lat>0</lat>
     <ele>0</ele>
   </location>
 ▼<datastream id="3">
     <tag>Spaces_left</tag>
     <current_time>Wed, 05 Feb 2020 11:03:14 GMT</current_time>
     <current_value>227</current_value>
     <unit_type>contextDependentUnits</unit_type>
     <unit_text>n/a</unit_text>
   </datastream>
 ▼<datastream id="2">
     <tag>Outgoing_total</tag>
     <current_time>Wed, 05 Feb 2020 11:03:14 GMT</current_time>
     <current_value>0</current_value>
     <unit_type>contextDependentUnits</unit_type>
     <unit_text>n/a</unit_text>
   </datastream>
 ▼<datastream id="1">
     <tag>Incoming_total</tag>
     <current_time>Wed, 05 Feb 2020 11:03:14 GMT</current_time>
     <current_value>23</current_value>
     <unit_type>contextDependentUnits</unit_type>
     <unit_text>n/a</unit_text>
   </datastream>
 </feed>
```


Figure 9 shows individual stream values

7 Conclusion

This proposed smart parking system prototype focuses on becoming a novel low-cost, low-maintenance solution to detect and count the cars. It uses a pressure sensor to detect and monitor incoming and outgoing cars from two ends of the car park and does the further processing on the raspberry pi to get the total number of cars going in and out to calculate the spaces available in the car park.

8 Acknowledgements

This research and implementation work were undertaken in collaboration with British Telecom.

9 References

1. R. Stiawan, A. Kusumadjati, N. Aminah, M. Djamal and S. Viridi, "An Ultrasonic Sensor System for Vehicle Detection Application", *Journal of Physics: Conference Series*, vol. 1204, p. 012017, 2019.

2. P. Verma and M. Verma, "Techniques for Smart & innovative parking, critical observations and Future Directions: A review", in *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, India, 2015.

3. A. Osmani, A. Gawade, M. Nikam and S. Wavare, "Research paper on Smart City Parking System" Vol-2 Issue-3 2016, IJARIIE-ISSN(O)-2395-4396.

4. H. Chaudhary, P. Bansal and B. Valarmathi, "CAR parking system using Arduino," *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, 2017, pp. 1-5.

5. A. Khanna and R. Anand, "IoT based smart parking system," *2016 International Conference on Internet of Things and Applications (IOTA)*, Pune, 2016, pp. 266-270.

6. D. Vakula and Y. K. Kolli, "Low cost smart parking system for smart cities," *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, 2017, pp.280-284.

7. S. Kazi, S. Nuzhat, A. Nashrah and Q. Rameeza, "Smart Parking System to Reduce Traffic Congestion," *2018 International Conference on Smart City and Emerging Technology (ICSCET)*, Mumbai, 2018, pp. 1-4.

8. T. N. Pham, M. Tsai, D. B. Nguyen, C. Dow and D. Deng, "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies," in *IEEE Access*, vol. 3, pp.1581-1591, 2015.

9. D. Kanteti, D. V. S. Srikar and T. K. Ramesh, "Intelligent smart parking algorithm," *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Bangalore, 2017, pp. 1018-1022. doi: 10.1109/SmartTechCon.2017.8358524.

10. Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated", *Mouser.co.uk*, 2012. [Online]. Available: https://www.mouser.co.uk/datasheet/2/302/MPX5010-1127094.pdf.

11. NRF24L01+ Single Chip 2.4GHz Transceiver Preliminary Product Specification v.10", *Sparkfun.com*, 2008. [Online]. Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf.

12. NRF24L01+PA+LNA 2.4GHz Wireless Transceiver Module - 1100 Meters [NRF24L01+PA+LNA] - US $2.40 HAOYU Electronics : Make Engineers Job Easier", *Hotmcu.com*, 2008. [Online]. Available: https://www.hotmcu.com/nrf24l01palna-24ghz-wireless-transceiver-module-1100-meters-p-276.html.

13. Adafruit Industries, "Adafruit Feather HUZZAH with ESP8266 - Loose Headers", *Adafruit.com*. [Online]. Available: https://www.adafruit.com/product/2821#.