Journal of Science and Technology Policy Management

Traffic systems in smart cities using LabVIEW

Aqeel Farooq, Wadee Alhalabi, Sara M. Alahmadi,

Traffic systems in smart cities using LabVIEW

Aqeel Farooq
Bright Inc., Jeddah, Saudi Arabia, and
Wadee Alhalabi and Sara M. Alahmadi
Department of Computer Science, Effat University, Jeddah, Saudi Arabia

Received 4 May 2017
Revised 11 June 2017
Accepted 17 July 2017

Abstract

Purpose – The purpose of this research work is to design and apply LabVIEW in the area of traffic maintenance and flow, by introducing improvements in the smart city. The objective is to introduce the automated human–machine interface (HMI) – a computer-based graphical user interface (GUI) – for measuring the traffic flow and detecting faults in poles.

Design/methodology/approach – This research paper is based on the use of LabVIEW for designing the HMI for a traffic system in a smart city. This includes considerable measures that are: smart flow of traffic, violation detection on the signal, fault measurement in the traffic pole, locking down of cars for emergency and measuring parameters inside the cars.

Findings – In this paper, the GUIs and the required circuitry for making improvements in the infrastructure of traffic systems have been designed and proposed, with their respective required hardware. Several measured conditions have been discussed in detail.

Research limitations/implications – PJRC Teensy 3.1 has been used because it contains enough general-purpose input–output (GPIO) pins required for monitoring parameters that are used for maintaining the necessary flow of traffic and monitor the proposed study case. A combination of sensors such as infrared, accelerometer, magnetic compass, temperature sensor, current sensors, ultrasonic sensor, fingerprint readers, etc. are used to create a monitoring environment for the application. Using Teensy and LabVIEW, the system costs less and is effective in terms of performance.

Practical implications – The microprocessor board shields for placing actuators and sensors and for attaching the input/output (I/O) to the LED indicators and display have been designed. A circuitry for scaling voltage, i.e. making sensor readings to read limits, has been designed. A combination of certain sensors, at different signals, will lead to a secure and more durable control of traffic. The proposed applications with its hardware and software cost less, are effective and can be easily used for making the city's traffic services smart. For alarm levels, the desired alarm level can be set from the front panel for certain conditions from the monitoring station. For this, virtual channels can be created for allowing the operator to set any random value for limits. If the sensor value crosses the alarm value, then the corresponding alarm displays an alert. The system works by using efficient decision-making techniques and stores the data along with the corresponding time of operation, for future decisions.

Originality/value – This study is an advanced research of its category because it combines the field of electrical engineering, computer science and traffic systems by using LabVIEW.

Keywords Sensors, Accelerometer, HMI, LabVIEW, PRJC Teensy 3.1, Virtual channels

Paper type Research paper

1. Introduction

Newly developed revolutionary technologies are being used in colonies to make cities smart. This is in accordance with the growth in the use of scientific methods for urban development. The use of new revolutionary technology includes the *use of automation, industrial control, embedded systems, artificial intelligence, self-optimization, cyber security,*

Journal of Science and Technology Policy Management
© Emerald Publishing Limited
2053-4620
DOI 10.1108/JSTPM-05-2017-0015

electronics and telecommunication techniques. For technology to behave smartly, the designed algorithms and decision-making techniques should be efficient and be designed for optimal conditions, often requiring fewer hardware resources. Often, the automation techniques with monitoring control over hardware enable the operators to monitor and control the conditions. Owing to the advancement in virtual reality, the quality of graphics used in automation is improving every day. In any case, an optimal condition is selected which ensures the lowest cost and efficient performance. Every area in a smart city needs to be considered for applying these concepts (Rida, 2017).

Within a few years, several cities around the globe have become more livable with a growth in their infrastructure. However, these days, there are few urban territories that are barely surviving and are facing large economic crises (Khatoun and Zeadally, 2017). Economies are under increased pressure because of the increase in energy consumption, changes in the environment and demands for health care and education systems. Each day, with the population boom, the demand for resources is growing and natural resource reserves, because of their rampant demand and use, are depleting. Electricity is mostly generated using minerals. With daily usage, mineral resources are gradually minimizing. So, several approaches are being used for making city systems smarter to limit the demand for natural resources, and thus maintaining optimal performance. For the city to be smart, the infrastructure including buildings and technology needs to be efficient and user-friendly (Berntzen, 2015). Health care in a smart city is also of considerable importance (Demirkan, 2013).

With the advancement in technology, computing machines and systems have opened a number of gateways, with the help of which pre-programmed devices can be created for smart conditions. Data can be used for storing customer usage record (history). Computer capabilities are advancing and use of programming languages and designed interfaces is increasing every day. Owing to these technologies, there is an interaction between user and technology, often defined by the term "human–machine interaction". The sensors obtain data from the real world as parameters and display it on a software interface. Similarly, data from the actions (clicks) in the software produce changes in the real world in the form of actuator actions such as movement of motors, lights display, etc. This interaction between the software and the real world allows the user to create an application software with various configurable hardware (*A combination of sensors, actuators, processors and circuitry*). Figure 1 shows the demonstration of the human–machine interaction.

Data from real-world devices such as sensors, converters, actuators (motors and rotating devices) and other controlled devices are being shared on the processing computer screen from where the user interacts with and controls the data, instead of visiting the real system (without going there). These days owing to improvements in computer graphics, data can be seen graphically. For example, if level sensors are placed in a water tank, then the human–machine interaction system allows the user to graphically visualize the water level on the processing computer. The water level can be controlled, if the control valve access is being shared on the processing computer. Real-world devices are connected with a computer using the data acquisition (DAQ) device.

This technique of transferring real-world data (parameters, physical quantities such as speed, time etc.) into computer systems is often referred to as data acquisition. During the acquisition process, real-world data are mostly in the form of an analogue signal. The analogue signal is then converted into a digital signal before being processed in a microprocessor. This is done by a device named *Analogue to Digital converter* (ADC). The ADC is always shared between the sensor and the microprocessor (microcontroller). This

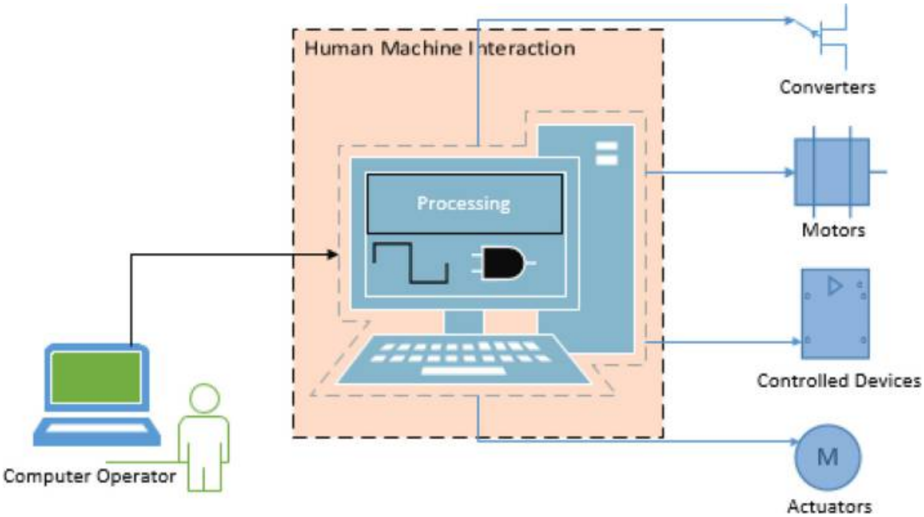


Figure 1.
Human–machine interaction

process will be the reverse when the data flow from the processor to the actuator (Walden, 1999). In this case, a device named *Digital to Analogue converter* (DAC) is used.

With the population boom, the number of cars on the road is also increasing, owing to which controlling heavy traffic is becoming difficult. Some countries use camera tracking for automatic detection of violations, but this system usually consumes considerable power and the resulting cost is higher. Owing to the large initial cost and power consumption cost, certain territories still find this system unfavorable. Some revolutionary ideas using sensors and modules such as GPS have been proposed (Hoh *et al.*, 2010).

In this research, a smart system for traffic management in the city has been proposed – by applying sensors, using actuators in cars, installing sensors and actuators inside traffic signal posts and alongside roads, etc. The system interface is proposed and designed using LabVIEW. The PJRC Teensy 3.1 is used as the DAQ device. Any standard PC running Windows or MAC can be used for monitoring and controlling actions. Monitoring sensors are installed in cars, inside traffic poles and alongside road. Sensors inside traffic poles are installed to detect faults. Traffic pole faults can be electrical and mechanical in a nature. A separate circuitry is designed for electrical and mechanical faults. However, citizen privacy needs to be considered because of the use of modules such GPS (Lu *et al.*, 2012).

The proposed hardware is a computer-controlled, low-cost and effective DAQ device using Teensy 3.1. The general-purpose input–output (GPIO) or the input/output (I/O) pins are sometimes regarded as Channels. The channels can be digital or analogue in nature. A digital channel takes the digital signal, whereas the analogue channel is capable of reading the signal value from any sensor. An analogue sensor can be connected with a digital sensor via an ADC. Using the Internet of Things (IoT) technology, the DAQ data can be visualized over the internet (Misbahuddin *et al.*, 2015).

2. Literature review

Building a Smart City is a broad innovation which covers the various aspects of urban and digital life. These include versatile improvements in various considerable sectors such as urban planning, sustainable development, smart grids, smart traffic, smart economic development, smart technologies and social development, etc. In the modern age, several vital phenomena have been appearing: information and communication technologies (ICTs), infrastructure development, implementation of smart algorithms in various fields, IoT, urbanization, smart industrial development and other latest emerging technologies. Technology and advancement during the 1990s contribute toward the economic growth in urban centers. All extraordinary developments (urbanization) have made the newly created urbanization possible and have led us to the transformation of rural areas into urban areas. These urban areas started to offer many opportunities in terms of education, work, social life and so on. These facilities (urbanization) keeps on improving with time, and the cycle of exploring culture begins (Cocchia, 2014).

The term “Smart City” was first used in 1998 (Van Bastelaer and Lobet-Maris, 1998). In the international context, to attain the list of purposes established in the Kyoto Protocol, the concept of making the city smart, often referred to as “Smart City”, was born, and it is now globally being used by many institutions and organization. Some examples of other alternative words are: Digital city, Wired City, Knowledge City, Cyber City, Green City and Advance City. All these words address the scale and complexity of the smart city (Cocchia, 2014).

Various scholars have addressed and debated on the term smart city since its first adaption in 1998. Several research works with different terminologies that identify the “Smart City” (Pardo and Tawewoo, 2011) and propose ideas about the definition of a smart city have been published. Some authors have attempted to analyze the context (Ishida, 2000). These authors have mainly focused on the literature search strategy. Alternative approaches for analyzing smart cities that include ICT attributes (such as digital media, broadband, wireless networks, etc.) in the urban city have also been discussed by various writers.

Countless debates regarding cybersecurity and privacy solutions in smart cities have been conducted. These debates have considered various sectors including government, health care, critical infrastructure, smart buildings, transportation, etc. These ideas include making the government sector an e-government sector and the health sector an e-health sector and applying IoT technology to various other sectors in the urban territory. Numerous privacy analyses, issues, models and future privacy recommendations have been proposed.

Researchers’ strong inclination toward concentrating on cities has led to an increase in positive and negative aspects, as well as the development of advanced technologies. The smart city revolution, at the same time, has caused an increase in urban culture, which has led to economic growth and production of new jobs. On the other hand, this revolution had led population boom in urban cities, thus resulting in problems such as traffic jams, increase in carbon dioxide emissions, greenhouse gas emissions and waste disposal in healthy environments. Moreover, the problem of considerable traffic, itself, is a problem. Owing to heavy traffic, the number of road-traffic accidents has increased multifold. According to the US Department of Health and Human Services, accidents in the USA have killed more than 35,000 people in 2015. Most of the times, accident effects the trees and earth’s natural environment. In this scenario, to save the planet and improve people’s health, the idea of making improvements in the smart city emerged, that is, paying attention towards and finding solution for issues such as traffic and pollution. In the past, various ideas including smart algorithms and agents and using IoT technology have been suggested (Chen and Cheng, 2010). Numerous other approaches such the vehicular ad hoc network (VANET) have been proposed (Khekare and Sakhare, 2013).

In this paper, the idea of traffic management in the smart city by using the LabVIEW-dedicated DAQ has been discussed. A low-cost microcontroller (PJRC Teensy 3.1) has been used as the DAQ device. The system is divided in to two parts to simplify the understanding of the system. One part deals with detection and measurement on the traffic signal pole, whereas other part deals with managing the strength of traffic on the road. The idea involves connecting the sensors from the road and the traffic poles to a microcontroller (PJRC Teensy 3.1) device. The microcontroller (PRJC Teensy 3.1 – DAQ Device) is then connected with a station PC on which the LabVIEW is running, allowing the operator to actively monitor and control the shared parameters on the road and from the traffic pole. These parameters help in controlling the traffic strength on the road. Accidents due to heavy traffic in a certain lane can be avoided. The main aim of the paper is to use sensing modules and actuators to maintain the traffic system (flow) in a smart city.

Furthermore, modules are proposed to install sensors in cars so that they can halt the vehicle in the range of a traffic signal by using the proposed smart mechanism. Details of modules have been explained in detail in the following sections.

3. System architecture

3.1 LabVIEW-based learning platform: introduction to LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a programming software that allows the user to write programs in a graphical language. The LabVIEW code is written in a graphical programming language. A simple LabVIEW program has two display panels, i.e. the front panel and the block diagram panel. The front panel contains the control and display parameter which normally takes the input and shows the output, whereas the block diagram panel contains the graphical code, for instance, if the user makes a graphical calculator using LabVIEW (Travis and Kring, 2006). In this case, the front panel contains the body and graphics (*the complete calculator will be displayed in the front panel*), whereas the block diagram contains the code written in a graphical language. LabVIEW has a support for several hardware, including *FPGA* (*field programmable gate array*), PLC (*programmable logic controllers*), DAQ (*data acquisition device*), signal processing boards, etc. Third-party hardware, which mainly includes *NI DAQ*, *Compact RIO*, *Arduino*, *Digilent*, *PJRC*, *Spark fun*, etc., is also supported. Some libraries such *virtual instrument software architecture* (VISA) toolset need to be installed, to get started. Using LabVIEW, certain different interfaces for the various applications can be designed for different applications (Oppermann, 2002).

Graphical programming is supported in LabVIEW, but by using built-in compilers, the graphical code can be converted into equivalent C-codes. LabVIEW can be used for designing interfaces. Using various libraries such as advanced mathematics and statistics, data processing, data storing, data managing, signal generation, processing and acquisition, etc., with their supports for LabVIEW, can help in running programs efficiently. LabVIEW has many applications in engineering studies and can be readily used for teaching control systems (Ertugrul, 1999).

With LabVIEW, *NI VISA* needs to be installed on the machine to allow for data acquisition and provide support for several hardware. There are some libraries that are required with the use of third-party hardware's such as LINX. LINX (*LabVIEW Maker Hub*) is the library support for the hardware control, with which numerous possible applications can be created. With the help of LabVIEW, applications for smart cities with the aid of the IoT technology can be designed (Gea *et al.*, 2013).

3.2 System diagram

"Data acquisition is the process of measuring and detecting the electrical quantity such as current, voltage, etc. and display it graphically on the computer screen".

A complete DAQ system consists of electronic modules (sensors, actuators), DAQ measurement hardware and a computer with programmable software. As the context of this paper, LabVIEW is used as programmable software and the micro-controller (PJRC Teensy 3.1) is the DAQ device. Figure 2 depicts the block diagram of the system (traffic pole fault detection) using PRJC Teensy 3.1 as the DAQ device. While designing the LabVIEW algorithmic code, the feedback approach is considered. The feedback approach allows efficient control of the hardware (Skogestad and Postlethwaite, 2007).

The electrical signals from the sensor modules (ultrasonic, accelerometer, digital compass, GPS) are being transferred to the PC using the Teensy DAQ device. The voltage and current levels are displayed graphically. From the DAQ graphical PC screen, the operator can operate the system. Factors such as traffic signal status, parametric quantities and sensor readings (graphically) are available on the PC software screen.

3.3 Hardware-in-the-loop simulation with LabVIEW and Teensy board

With the help of LabVIEW Maker Hub Library, LabVIEW supports numerous microprocessor device families that can be used as DAQ devices. These models include *Arduino, Digilent, Teensy, SparkFun* and a few other microprocessors (Barrett, 2013). Library functions such as digital read, digital write, analog read, analog write, PWM write, etc. are supported for graphical programming. These elemental blocks can be joined with each other to create a complete graphical program.

By using LabVIEW Maker Hub Library, data can be obtained from the DAQ device. The connected sensors' data can be monitored in LabVIEW. Similarly, the output hardware can be controlled via graphical actions on the front panel window. To make a program that never stops, a "while loop" is drawn outside. Inside the graphical while loop, a code in defined using blocks. The block such as *analog write, digital read* allows to read and write the data. Sensors can be connected with a channel (*Teensy GPIO- I/O pin*) and the data can

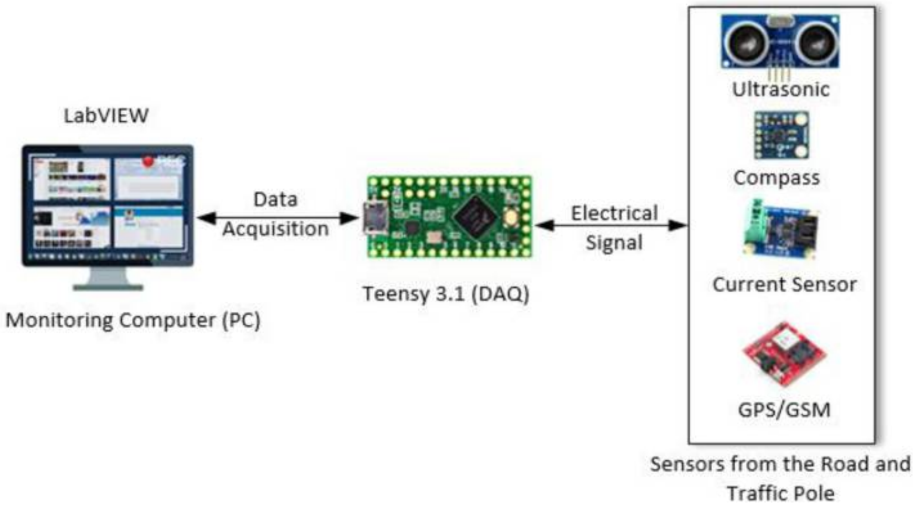


Figure 2.
System diagram (Case – 1: sensor from the traffic signal)

be read. After reading the data, actuating actions can be triggered with the occurrence of certain points or limits (*This means that if a certain set limit is past, then actuators such as motors and the LED can be allowed to run using output channels*). Output channels help in displaying the data and for triggering the actions in a normal or in an abnormal case.

To write a program using the LINX library in LabVIEW, an OPEN block is required to be placed at the start, and the last block must be a CLOSE block. The I/O blocks such as *analog write, analog read, digital write, digital read, PWM pulse, I2C, SPI* and *UART communications* are placed between the OPEN and CLOSE blocks. The *analog write* blocks can produce a voltage from 0 V to V_{cc} (*biasing voltage or applied voltage*). The *analog read* can read any value from 0 V to V_{cc} (*biasing voltage or applied voltage*). The *digital write* can produce LOW (*0 V*) and HIGH (*5 V or 3.3 V*) voltage values, while the *digital read* can read only HIGH and LOW values. The *PWM pulse* block can be used to produce a square wave duty cycle from 0 to 100 per cent (*the duty value average to produce a DC value*). There are some sensor libraries available that can be considered, but for the sensors whose libraries are not available, scaling and mathematical blocks are required to be added to obtain real-time physical quantity. The UTILITIES (*in the LINX Library*) block allows the programmer to change the *Arduino* frequency and configure the Wi-Fi and other.

3.4 VR Lab system architecture

The complete set of hardware is named as VR Lab system hardware. In this research work, the idea for managing, improving, realizing and constructing the traffic system, a non-violating (no one can cross the traffic signals and make violations) system has been discussed and proposed. The DAQ device used for this purpose is the PRJC Teensy 3.1. The sensors and the hardware modules such as the accelerometer, magnetic compass, ultrasonic sensors, GPS/GSM modules and current sensors help establish the quality of the system for monitoring the faults in the traffic pole and control the traffic flow. The sensors are required to be connected to the PRJC Teensy 3.1, thereby allowing data acquisition between the PC and the Teensy 3.1 on the LabVIEW front panel. The actuating actions can be effected to control the output parameters from the same front panel. The front panel can be visualized at selected remote stations via the LabVIEW Web server. (*A LabVIEW program is allowed to be visualized over the Web with appropriate credentials.*) The complete explanation of the system hardware is divided into parts detailed in the following sections.

3.5 Traffic pole

There are two types of faults that can occur in the traffic pole (electrical and mechanical). During an electrical fault, one or more of the traffic signal lights usually stops working owing to faults such as *power loss, wire cut, hardware damage* (*usually burning of the internal hardware component*), whereas the mechanical faults include bending or movement of the pole from its installed position. This usually occurs because of high wind pressure and moisture in the air that reacts with metal and makes it weak over time. There are three lights in every traffic pole. The Hall effect current sensor ACS712 can be connected in series with every traffic light to measure the current over time. If the current stops after X years, this means that a fault occurred in the light. The accelerometer and compass modules such as HMC5883L, if installed in the electrical pole, help in detecting the bending or movement of the pole from its installed position.

The reading from the current sensor is used in displaying the traffic signal graphics in LabVIEW. This creates a monitoring view. The LabVIEW front panel is connected with the GSM module to send an alert over the network in case of a fault in a certain traffic signal. Traffic data can be visualized over the Web (Li *et al.*, 2016) (Figure 3).

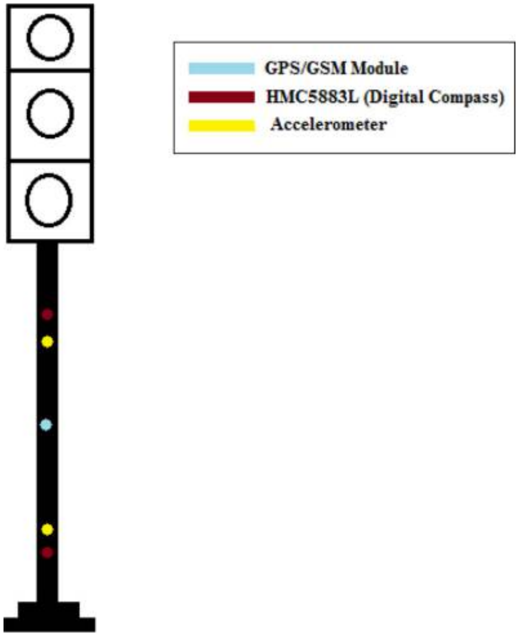


Figure 3.
Traffic poles with sensors' (modules) positions

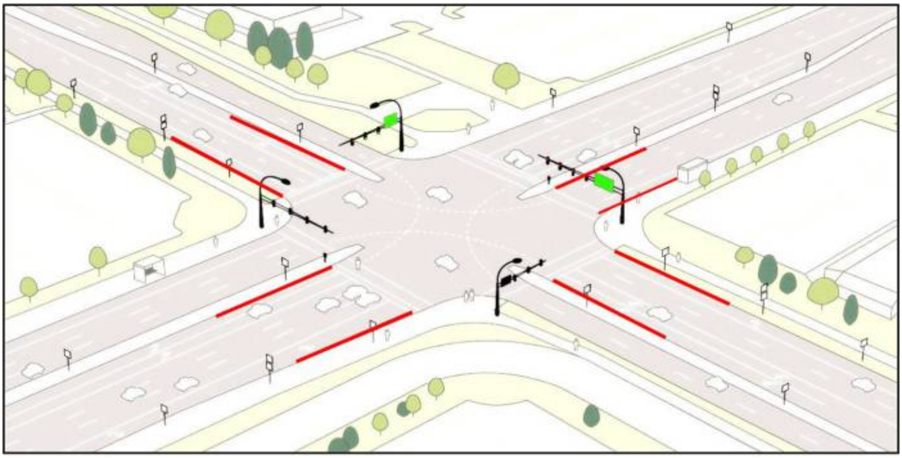
3.6 Sensor on the road

The sensor networks on the road allow managing the traffic flow. The idea under consideration is smartly switching the traffic in smart cities. For instance, if there are four lanes in a road and they are being controlled by traffic signals, every traffic lane is ON for 120 s; with one lane ON, all others will be OFF. In case of LOW traffic on a specific lane, it is not a good idea for the specific lane to remain ON for 120 s. After 30 s, the traffic signal will be switched to the more traffic lane and it will turn itself OFF. The traffic on the road can be monitored using ultrasonic sensors' array placement on the road. An ultrasonic sensor consists of a receiver and a transmitter. In every lane, an ultrasonic sensor needs to be installed with the transmitter at one side and the receiver at the other side. The receiver and the transmitter measure the distance, in case of no traffic. This "no traffic" distance is considered as the reference value. If the newly attained value is lesser than the reference value, then there is traffic on the road. Hence, by using ultrasonic sensors, the traffic strength (distance) can be gauged, thereby allowing the smart flow of traffic in any lane. Figure 4 shows the ultrasonic sensor positions on the road with four lanes of traffic.

Using ultrasonic sensors, a programmed hardware can measure the distance after 30 s on a lane with moving traffic. If there is no traffic on the lane, then switching will occur. For example, if *Lane-1* is ON and other lanes (*Lane-2, Lane-3, Lane-4*) are OFF, then after 30 s, the signal on *Lane-1* is ON; in case there is no traffic on *Lane-1*, *Lane-1* will be OFF and traffic will switch to *Lane-2* (*making Lane-2 ON*).

3.7 Sensors in the car

It is necessary to monitor cars that are in the range of the traffic signals. The GPS module in the car allows the detection of the car position on the road. There is a GPS module on the traffic signal pole. If the car GPS comes near the traffic signal (GPS module – detection from



Note: Labeled as red

Figure 4. Sensor positions on the road

the co-ordinates) and the traffic signal is ON, then the station sends a message to the car. Upon receiving the message, the car will be turned OFF, if its GPS co-ordinates are in range of the traffic signal with the status ON. In this way, GPS/GSM allows avoiding traffic signal violation, thus making it a no-traffic-violation city.

3.8 Workings of the system

The traffic pole faults (electrical and mechanical) are being monitored by the sensing modules. The GPS/GSM modules in the car lock down the car, if the car is in the range of the traffic signal with status ON (to avoid violations). The car can also be locked down from the remote stations' front panel to revert criminal activity (using LabVIEW front panel). There are sensors on the road for smart switching of the lane to maintain the traffic flow.

3.9 Accuracy and cyber security

The LabVIEW server panel is displayed over the network, not over the internet. The network should use a secure cable (without internet) to prevent any hacking activity.

3.10 Data logger

Storing the data for future processing is called data logging (Arini *et al.*, 2015). Using LabVIEW, the data from the sensors' value can be stored, logged and kept for traffic record. The car data, traffic faults and road sensors data can be logged using the LabVIEW data logger.

4. Experimental methods

The idea for placing sensors and modules on the road, traffic poles and cars has been proposed. If these sensors and modules are installed, then faults in the traffic poles (using current sensors, magnetic compasses and accelerometers) can be detected, traffic on the lane (using ultrasonic sensors) can be detected and automatic stop action of the car in the range of the signal (using GPS/GSM) can be effected. The station will send messages to the car whose co-ordinates are near to the traffic signal poles' GPS co-ordinates (*with traffic signal* *status ON to avoid violation*). This helps in achieving a smart city that is free of traffic signal violations. The cars can be locked down in their position for avoiding criminal activity manually (using LabVIEW front panel by just typing the car number). In smart cities, maintaining the flow of traffic and transportation is very important (Zhu *et al.*, 2016).

4.1 Methodology

4.1.1 Getting device-ready: uploading the firmware. To get started, the microcontroller (Teensy 3.1) needs to be directly connected with the PC that has LabVIEW and the Maker Hub Library needs to be uploaded in the microcontroller (Teensy 3.1). To upload the firmware, go to Tools > Maker Hub > LINX > LINX Firmware Wizard. Afterward, the microcontroller (Teensy 3.1) needs to be selected from the list of available devices and the firmware can be uploaded.

The sensors from the road and traffic signal can be connected with a microcontroller (Teensy 3.1), GPIO now.

4.1.2 LabVIEW simulation. The LabVIEW program contains two display windows. One is named as the front panel (the one which is available at the remote service station) and the other is a block diagram (contains the block diagram code).

The proposed example of the LabVIEW monitoring screen for a four-lane road with fault prompt is shown in Figure 5.

The above proposed graphic layout will be at the operator station that is connected to the street. The mapping on the front panel is a graphical view of the original traffic signal status on the lanes (*Lane-1, Lane-2, Lane-3 And Lane-4*). The fault status information and the mobile number for alert, ultrasonic sensor distance measurement for a specific lane can also be visualized using the same panel. The proposed graphics can be customized or split into multiple GUIs.

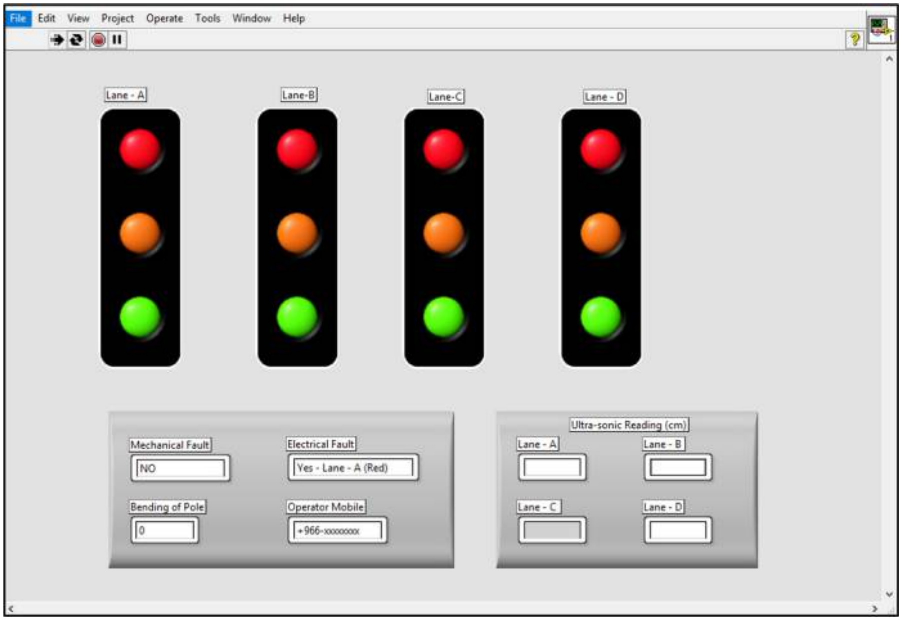


Figure 5.
Proposed LabVIEW graphics

5. Performance evaluation

The system involves computer and computing devices. The power of computer is usually considered, as its processing speed is *alternatively referred to as working speed*. To conduct performance evaluation for the system, working speed is required to be calculated and compared with standard available DAQ devices.

The working speed of the system greatly depends on the number of operation (*executions in the processors*). The time used during this process is regarded as the "time factor". The time factor defines the time that an input action takes to produce an output. This time factor is usually of the order of ns(10^{-9} s). This time factor depends on the frequency, clock speed and architecture of the microprocessor and the monitoring CPU (PC CPU – *normally Intel or AMD Core*). Sometimes, heating of the CPU adds delay in the process; therefore, by improving the thermal properties, the time factor can be improved.

5.1 Performance evaluation – 1: time factor calculation

Using the Intel Core – i5, 3.0 GHz processor and Teensy 3.1, the time factor is:

$$Time\ \ factor = \frac{1}{CPU\ \ frequency} + \frac{1}{Teensy\ \ Frequency}$$

$$Time\ \ factor = \frac{1}{3.0\ \ GHz} + \frac{1}{72\,MHz} = 0.33\,ns + 0.013\,us$$

The time factor is quite low and comparable with the advanced control systems such as *distributed control systems* (DCS), *field graphic programmable arrays* (FPGA), *digital signal processing* (DSP) boards and PLCs (*programmable logic controller*), even though there is large cut down of cost in this discussed case. The VI (*virtual instruments – which is the LabVIEW file extension*) file is programmed with a GUI panel. The written program can be run by pressing the run button on the top bar.

5.2 Performance evaluation – 2: time estimation for the complete process

The estimation time depends on the processing power of the PC, GPS/GSM and Teensy 3.1. In case of an accidental lockdown of a car, the estimation time using Core – i3 processor with 2.2 GHz CPU containing 4 cores, is:

$$Estimation\ \ Time = \frac{1}{CPU\ \ frequency} + \frac{1}{Teensy\ \ Frequency} + Message\ \ sending\ \ time$$

$$Estimation\ \ Time = \frac{1}{2.2\,GHz} + \frac{1}{72\,MHz} + Message\ \ sending\ \ time$$

Using the message sending time as 1 s, the car can be stopped in a maximum duration of 1.1 s. The data acquisition unit (E6500) manufactured by SUPCON has a sampling speed of 0.125 s (SUPCON E6500 DAQ Device, 2016). The proposed system has a messaging service included with time approximation, so, for comparison, if the messaging service is excluded, the sampling time or time estimation for one cycle is 0.1 s, which is better than the most of the industrially available DAQ devices.

5.3 Performance evaluation – 3: accuracy

If the car on the road is travelling with the speed of 100 ms^{-1}, then the car can be turned OFF before it covers a distance of 90 m (100 ms^{-1}/1.1 s). For speed up to 30 ms^{-1} (normal speed in the city), the accuracy is 27 m (30 ms^{-1}/1.1 s). The car, after being stopped, will send its current co-ordinates to the station (if the car needs to be locked down in the case of suspected criminal activity).

6. Advantages and disadvantages of the system

6.1 Advantages of the research

- The proposed system is a low-cost system with high accuracy.
- The proposed system can stop a car (for criminal action), monitor traffic pole faults and can act as a smart switching system for signals, thus making the probability of lesser or no traffic signal violation.
- The system is designed using LabVIEW which can be used to design any other automation-based application in a smart city.
- Using a GPS module, the speed of cars can be detected over a certain distance. Automated violation fines can be send, in case of speeding.
- Smart fault monitoring can be done from the remote station.
- The system has its separate network which is not over the internet, thus there are little chances of hacking.
- Drivers, cars and roads can be categorized based on their condition of operation and behavior.
- Behavioral analysis of the system can be performed on the basis of its performance.
- Further sensors can be applied to measure further parameters that can help in improving the overall globalization (making every one install it) of the system.
- The proposed DAQ device is better in terms of performance than already available DAQ devices such as SUPCON E6500.

6.2 Disadvantages of the research

- Using the current system with connected sensors, violations such as using the phone while driving cannot be detected.
- The system uses GSM technology owing to which a delay in the performance occurs, but the delay can be reduced with the use of the faster communication modules (*Bluetooth, Wi-Fi, Ethernet, antenna*, *etc.*) over certain measured distances.

7. Conclusions

LabVIEW is a software that can be used with some processor families to create a real-time application that can control hardware using a GUI. The interface can also monitor the hardware parameters (sensor data from the real world). With the help of software (LabVIEW), the desired output conditions can be achieved and monitored.

In this research, a traffic application in a smart city is proposed using LabVIEW with Teensy 3.1 as the DAQ device. From the LabVIEW front panel, traffic can be detected (whether there is traffic or not – using ultrasonic sensors) for smart switching of traffic lanes and maintaining a reasonable flow of traffic (by avoiding crowd). The faults (electrical power loss or mechanical) on the traffic pole can be detected and monitored, and upon the occurrence of fault, alerts can be send to the operator and the maintenance team during offline hours. Cars that are in the range of the traffic signal with the ON status can also be locked. Ultrasonic sensors are placed on the road for measuring distances. The maximum distance occurs in the case there is no traffic on the road. When there is traffic, the distance seen by ultrasonic sensor will be lower than the maximum value owing to traffic in the lane. Using this scheme, traffic can be detected, and with programmed hardware inside the traffic poles, switching of traffic lanes using decision-making algorithms can be done. The cars and the traffic poles are embedded with GPS/GSM modules. When car co-ordinates are in the range of the ON signal, then message commands will be sent to the car for automatic and efficiently stopping.

In short, the idea to improve the traffic system in a smart city is proposed using LabVIEW software with Teensy 3.1 as the DAQ device. Using a combination of sensors and modules, different parameters can be measured on the road, from the pole and from moving cars. Using the modules such GPS and GSM allows the cars to be stopped automatically if they are in the range of the ON signal. This creates "a Smart City" with no traffic signals violation.

The performance analysis is done on the system, time factor (working speed) and time estimation for the speed is calculated. The system is then compared with an industrially available DAQ device (E6500) from SUPCON. Upon comparison, it was found that the custom-made LabVIEW-based system for traffic is faster, as it contains a faster microcontroller (Teensy 3.1).

References

Arini, N.S., Krieger, K. and Rowe, S.M. (2015), "Data logging for media consumption studies", Google Patents.

Barrett, S.F. (2013), "Arduino microcontroller processing for everyone! Synthesis lectures on digital circuits and systems", Vol. 8 No. 4, pp. 1-513.

Berntzen, L. (2015), *Smart Cities, Smart Buildings, Smart Users*, Keynote, DigitalWorld 2015, Lisbon.

Chen, B. and Cheng, H.H. (2010), "A review of the applications of agent technology in traffic and transportation systems", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11 No. 2.

Cocchia, A. (2014), "Smart and digital city: a systematic literature review", *Smart City (How to Create Public and Economic Value with High Technology in Urban Space*, Springer, Progress in IS, pp. 13-43.

Demirkan, H. (2013), "A smart healthcare systems framework", *IT Professional*, Vol. 15 No. 5, pp. 38-45.

Ertugrul, N. (1999), "Towards virtual laboratories: a survey of LabVIEW-based teaching/learning tools and future trends".

Gea, T., Paradells, J., Lamarca, M. and Roldan, D. (2013), "Smart cities as an application of Internet of Things: experiences and lessons learnt in Barcelona", *Proceeding 7th International Conference Innovation Mobile Internet Services in Ubiquitous Computing (IMIS)*, pp. 552-557.

Hoh, B., Gruteser, M., Xiong, H. and Alrabady, A. (2010), "Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking", *IEEE Transaction Mobile Computing*, Vol. 9 No. 8, pp. 1089-1107.

Ishida, T. (2000), "Understanding digital cities", in Isbister, T.I. (Ed.), *Digital Cities*, LNCS, vol. 1765, Springer, Berlin, pp. 7-17.

Khatoun, R. and Zeadally, S. (2016), "Smart cities: basic concepts, architectural issues, and research opportunities", *Communications of the ACM*, Vol. 59 No. 8, pp. 46-57.

Khekare, G.S. and Sakhare, A.V. (2013), "A smart city framework for intelligent traffic system using VANET", 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), Kottayam, pp. 302-305.

Li, X., Zhihan, L.V., Wang, W., Zhang, B., Hu, J., Yin, L. and Feng, S. (2016), "WebVRGIS based traffic analysis and visualization system", *Advances in Engineering Software*, Vol. 93, pp. 1-8.

Lu, R., Lin, X., Luan, T.H., Liang, X. and Shen, X. (2012), "Pseudonym changing at social spots: an effective strategy for location privacy in VANETs", *IEEE Transactions on Vehicular Technology*, Vol. 61 No. 1, pp. 86-96.

Misbahuddin, S., Zubairi, J.A., Saggaf, A., Basuni, J., Wadany, S.A. and Al-Sofi, A. (2015), "IoT based dynamic road traffic management for smart cities" 2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET), Islamabad, pp. 1-5.

NESIMI ERTUGRUL (1999), "Towards Virtual Laboratories: a Survey of LabVIEW-based Teaching/Learning tools and future trends".

Oppermann, R. (2002), "User-interface design", *Handbook on Information Technologies for Education and Training*, Springer, pp. 233-248.

Pardo, T. and Tawewoo, N. (2011), "Conceptualizing smart city with dimensions of technology, people, and institutions", *Proceedings of the 12th Annual international Conference on Digital Government Research*, ACM, New York, NY, pp. 282-291.

Rida, K. (2017), "Cybersecurity and privacy solutions in smart cities", *IEEE Communications Magzine*, Vol. 55 No. 3, pp. 51-59.

Skogestad, S. and Postlethwaite, I. (2007), *Multivariable Feedback Control: Analysis and Design*, Wiley, New York, NY, Vol. 2.

SUPCON E6500 DAQ Device (2016), available at: http://en.supcon.com/product/detail-id-42.html

Travis, J. and Kring, J. (2006), *LabVIEW for Everyone: Graphical Programming Made Easy and Fun (National Instruments Virtual Instrumentation Series)*, Prentice Hall PTR.

Van Bastelaer, B. and Lobet-Maris, C. (1998), "Social learning regarding multimedia developments at a local level: the case of digital cities", in van Bastelaer, B. and Lobet-Maris, C. (Eds), *SLIM Research—DG XII—TSER Program, Final integrated study—Social learning in the public sector*, University of Namur.

Walden, R.H. (1999), "Analog-to-digital converter survey and analysis", *IEEE Journal on Selected Areas in Communications*, Vol. 17 No. 4, pp. 539-550.

Zhu, F., Li, Z., Chen, S. and Xiong, G. (2016), "Parallel transportation management and control system and its applications in building smart cities", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17 No. 6, pp. 1576-1585.

About the authors

Aqeel Farooq has a Bachelor in Electrical Engineering from UET, Lahore. He has research interest in Electrical Engineering and virtual reality systems. He has a preferred badge from Freelancer.com and is currently working with a company in Bright Inc., Jeddah, Saudi Arabia. He often visits and works as a Researcher at the Virtual Reality Center, Effat University, Jeddah, Saudi Arabia. Aqeel Farooq is the corresponding author and can be contacted at: aqeelfarooq92@gmail.com

Wadee Al Halabi got his master and PhD from the University of Miami, department of Electrical and Computer Engineering in 2004 and 2008 respectively. He worked as an electronics technician, systems' engineer, then he moved to work as a lecturer and a professor. He has a vast experience in the industry, as he worked for Jeddah desalination and power generation plant, Saudi Arabia Monetary Agency, Makkah College of Tech., and King Abdulaziz University.

Sara M. Alahmadi is currently working as an Assistant in the Department of Computer Science, Effat University, Jeddah. She is interested in writing and doing research work as she is the co-author for this paper.