DESIGN AND IMPLEMENTATION OF REAL TIME TRAFFIC LIGHT SYSTEM USING RASPBERRY PI AND OPEN CV

Mrs.V. Vidyalakshmi[1]

[1]Asst. Professor, Velammal Engineering College

Varun J, Rahul P[2], Shiam S T[3], Naveen Balaji T J[4]

[2-4]Velammal Engineering College

---***---

ABSTRACT: The solution we provide for Traffic management by having a special intelligence which the images of road feed from the cameras (PC Camera) at traffic junctions for real time traffic density calculation using image processing. It also focuses on the algorithm for switching the traffic lights according to vehicle density on the road, thereby aiming at reducing the traffic congestion on roads which will help lower the number of accidents. In turn, it will provide safe transit for people and reduce fuel consumption and waiting time. It will also provide significant data which will help in future road planning and analysis. In further stages multiple traffic lights can be synchronized with each other with an aim of even less traffic congestion and free flow of traffic. The vehicles are detected by the system through images instead of using electronic sensors embedded in the pavement. A camera will be placed alongside the traffic light. It will capture images sequences. Image processing is a better technique to control the state change of the traffic light. It shows that it can decrease the traffic congestion and avoids the time being wasted by a green light on an empty road. It is also more reliable in estimating vehicle presence because it uses actual traffic images. The red signal becomes the green signal when the ambulance arrives by using ZigBee. With the help of ZigBee all the red signals will be turned to green in order to provide a clear way for emergency vehicles (Ambulance).

General Terms: Image recognition, Traffic control, Open CV.

Keywords: Traffic congestion, Raspberry pi, Arduino, ZigBee, Python, Machine Learning, Open CV.

1. INTRODUCTION

Traffic-jam is a very big problem in developing cities, In fact it's ever increasing day-by-day nature makes it difficult to find where the traffic density is more in real time, so that to schedule a better traffic signal control and effective traffic routing. The root cause of this can be of different situations like congestion in traffic like insufficient Road width, Road conditions due to weather, unrestrained demand, large delay of Red Light etc. While insufficient capacity and unrestrained demand are somewhere interrelated, the delay of respective light is hard coded and not dependent on traffic. Indeed, manual control is must, Therefore, in order to reduce man's power, the need for simulating and optimizing traffic control to satisfy the increasing demand arises. Technology in the recent past uses image processing for surveillance and safety, which is widely used in vehicle and traffic management for traveler information. The traffic density estimation can also be achieved using Image Processing.

2. BLOCK DIAGRAM AND WORKING

Camera should be faced in each direction to monitor the traffic density by having a number of boundaries of each vehicle on the road. When the traffic density of one direction is high, green signal will turn in that direction and rest will be in red. Similarly, by analyzing the order of traffic density traffic signal should be changed in every direction. With the help of ZigBee all the red signals will be turned to green in order to provide a clear way for emergency vehicles. So that traffic signaling system becomes automatic using image processing through open CV and python.

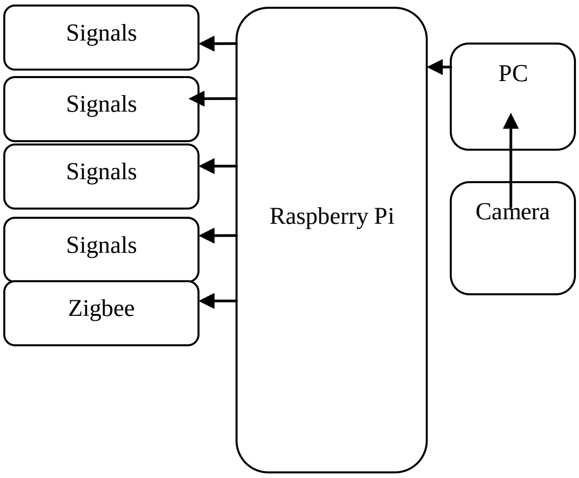


Figure 1: Block Diagram of Traffic Light System

3. COMPONENTS

3.1 Camera

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video. For example, Apple's iSight camera, which is built into Apple laptops, iMacs and a number of iPhones, can be used for video chat sessions, using the iChat instant messaging program (now called Messages). Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires a lot of bandwidth, such streams usually use compressed formats. The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

Figure 2: Camera

3.2 ZigBee

The XBee/XBee-PRO RF Modules are designed to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices. The modules operate within the ISM 2.4 GHz frequency band. The XBee/XBee-PRO ZB firmware release can be installed on XBee modules. This firmware is compatible with the ZigBee 2007 specification, while the ZNet 2.5 firmware is based on Ember's proprietary "designed for ZigBee" mesh stack (EmberZNet 2.5). ZB and ZNet 2.5 firmware are similar in nature, but not over-the-air compatible. Devices running ZNet 2.5 firmware cannot talk to devices running the ZB firmware.

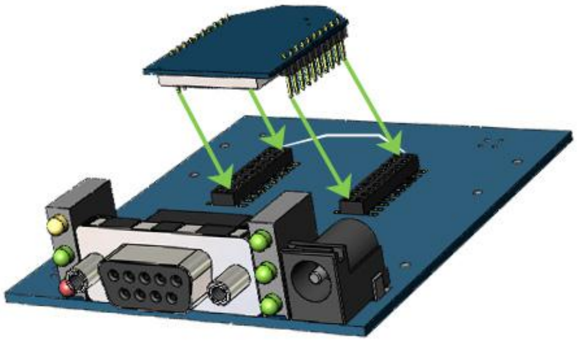


Figure 3: ZigBee Module Mounting to an RS232 Interface Board

The XBee modules were designed to mount into a receptacle (socket) and therefore do not require any soldering when mounting it to a board. The XBee-PRO Development Kits contain RS- 232 and USB interface boards which use two 20-pin receptacles to receive modules. Figure shows XBee-PRO Module Mounting to an RS-232 Interface Board. The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions.

3.2.1 Key Features

- High performance, Low cost.
- Indoor/Urban: Upto 300' (100 m).
- Outdoor line-of-sight: upto 1 mile (1.6 km).
- Transmit power output: 100 mW (20 dBm).
- Receiver sensitivity: -102 dBm.
- RFdata rate: 250,000bps.

3.2.2 Advanced Networking and Security

- Retries and Acknowledgements
- DSSS (Direct Sequence Spread Spectrum)
- Each direct sequence channel has over 65,000unique network addresses available.
- Point-to-point topology.
- Point-to-multipoint topology.
- Self-routing, self-healing and fault tolerant.
- Mesh networking.

3.2.3 Low Power

- TX Current: 295 mA (@3.3 V).
- RX Current: 45 mA (@3.3 V).
- Power-down Current: <1 µA @ 25oC.

3.2.4 Easy-to-Use

- No configuration necessary for out-of-box.
- RF Communications.
- AT and API command modes for configuring module parameters.
- Small form factor.
- Extensive command set.
- Free X-CTU software (Testing and Configuration software).

3.2.5 RF Module Operation

The XBee OEM RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device. Data enters the module UART through the DIN (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted. Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module. The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits). The XBee modules maintain small buffers to collect received serial and RF data, which is illustrated in the figure below. The serial receive buffer collects incoming serial characters and holds them until they can be processed. The serial transmit buffer collects data that is received via the RF link that will be transmitted out the UART. When serial data enters the RF module through the DIN Pin (pin 3), the data is stored in the serial receive buffer until it can be processed. Under certain conditions, the module may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the module, CTS flow control may be required to avoid overflowing the serial receive buffer. If the module is receiving a continuous stream of RF data, the data in the serial receive buffer will not be transmitted until the module is no longer receiving RF data. If the module is transmitting an RF data packet, the module may need to discover the destination address or establish a route to the destination. After transmitting the data, the module may need to retransmit the data if an acknowledgment is not received, or if the transmission is a broadcast. These issues could delay the processing of data in the serial receive buffer. When RF data is received, the data is moved into the serial transmit buffer and sent out the UART. If the serial transmit buffer becomes full enough such that all data in a received RF packet won't fit in the serial transmit buffer, the entire RF data packet is dropped. If the RF data rate is set higher than the interface data rate of the module, the module could receive data faster than it can send the data to the host. If the host does not allow the module to transmit data out from the serial transmit buffer because of being held off by hardware flow control.

3.2.6 RS232 Communications

Information being transferred between data processing equipment and peripherals is in the form of digital data which is transmitted in either a serial or parallel mode. Parallel communications are used mainly for connections between test instruments or computers and printers, while serial is often used between computers and other peripherals. Serial transmission involves the sending of data one bit at a time, over a single communications line. In contrast, parallel communications require at least as many lines as there are bits in a word being transmitted (for an8-bit word, minimum of 8 lines are needed).Serial transmission is beneficial for long distance communications, whereas parallel is designed for short distances or when very high transmission rates are required.

3.2.6.1 Standards

One of the advantages of a serial system is that it lends itself to transmission over telephone lines. The serial digital data can be converted by modem, placed onto a standard voice-grade telephone line, and converted back to serial digital data at the receiving end of the line by another modem. Officially, RS-232 is defined as the ”Interface between data terminal equipment and data communications equipment using serial binary data exchange.” This definition defines data terminal equipment (DTE) as the computer, while data communications equipment (DCE) is the modem. A modem cable has pin-to-pin connections, and is designed to connect a DTE device to a DCE device.

3.2.6.2 Interfaces

In addition to communications between computer equipment over telephone lines, RS-232 is now widely used for direct connections between data acquisition devices and computer systems. As in the definition of RS-232, the computer is data transmission equipment (DTE). However, many interface products are not data communications equipment (DCE). Null modem cables are designed for this situation; rather than having the pin- to-pin connections of modem cables, null modem cables have different internal wiring to allow DTE devices to communicate with one another.

3.2.6.3 Cabling Options

RS-232 cables are commonly available with 4, 9 or 25-pin wiring. The 25-pin cable connects every pin; the 9-pin cables do not include many of the uncommonly used connections; 4-pin cables provide the bare minimum connections, and have jumpers to provide “handshaking” for those devices that require it. These jumpers connect pins 4, 5 and 8, and also pins 6 and 20. The advent of the IBM PC AT has created a new wrinkle in RS-232communications. Rather than having the standard 25-pinconnector, this computer and many new expansion boards for PC's feature a 9-pin serial port. To connect this port to a standard 25-pin port, a 9-to-25-pin adaptor cable can be utilized, or the user can create his own cable specifically for that purpose.

3.2.6.4 Selecting a Cable

The major consideration in choosing an RS-232 cable is what devices are to be connected? First, are you connecting two DTE devices (null modem cable) or a DTE device to a DCE device (modem cable)? Second, what connectors are required on each end, male or female, 25-pin or 9-pin (AT style)? Usually, it is recommended that the user obtain the two devices to be connected, and then determine which cable is required.

3.2.7 Max232

The MAX232 from Maxim was the first IC which in one package contains the necessary drivers (two) and receivers (also two), to adapt the RS-232 signal voltage levels to TTL logic. It became popular, because it just needs one voltage (+5V) and generates the necessary RS-232 voltage levels (approx. -10V and +10V) internally. This greatly simplified the design of circuitry. Circuitry designers no longer need to design and build a power supply with three voltages (e.g. -12V, +5V, and +12V), but could just provide one +5V power supply, e.g. with the help of a simple 78x05 voltage converter. The MAX232 has a successor, the MAX232A. The ICs are almost identical, however, the MAX232A is much more often used (and easier to get) than the original MAX232, and the MAX232A only needs external capacitors 1/10th the capacity of what the original MAX232 needs

3.3 Raspberry Pi

Raspberry Pi is a small single-board Computer developed in UK by Raspberry Pi foundation to promote the teaching of computer science in schools and in developing countries. Original model become far more popular than anticipated sealing outside of its target market, for uses such as robots.

3.3.1 History

Raspberry Pi has mainly three generations Raspberry Pi 1, Raspberry Pi 2, Raspberry Pi 3 and also a reduce simple inexpensive Raspberry Pi zero. The first model of Raspberry Pi was launched in February 2012 i.e. Raspberry Pi 1 Model B followed by a simple inexpensive Model A. In April 2014 "Compute Model" for embedded application Raspberry Pi 1 model B+ improved versions of A and B was launched. In November 2015 with reduced I/O and GPIO Raspberry Pi zero came into market. In February 20-15 advance model with 40 GPIO pins, Ethernet, 4 USB slots Raspberry Pi 2 was launched. In February 2016 an upgraded model with inbuilt Bluetooth and Wi-Fi Raspberry Pi 3 Model B was launched. Recently in February 2017, "Raspberry Pi – Zero W" with in- built Wi-Fi and Bluetooth came into the market.

3.3.2 Features

The heart of the Raspberry Pi is a Broadcom System on Chip (SOC) which includes ARM compatible CPU and on-chip graphic processing unit and Vediocore IV. The key feature from First generation to the Third generation includes:

- CPU speed ranges from 700 MHz to 1.2 GHz.
- On board Memory (RAM) ranges from 256 MB to 1 GB.
- USB slot differs from 1 slot to USB slots.
- HDMI, composite video output and 3.5mm phone jack.
- Low level output is provided by GPIO pins which support common. Protocols like I2C (inter-integrated circuit).
- Ethernet 8 Position 8 Contact (8P8C).

3.3.3 Processor

The processor at the heart of the Raspberry Pi is a Broadcom BCM28XX. This is the Broadcom System on Chip (SOC) chip use in the Raspberry Pi. The processor from first to third generations includes:

- Raspberry Pi 1: Broadcom BCM2835 SOC with 700MHz CPU speed, L2 cache of 128kb with ARM compatibility AR1176JZF-S (ARMv6) 32-bit RISC ARM.
- Raspberry Pi 2: Broadcom BCM 2836 SOC with 900MHz CPU speed, L2 cache of 256kb with 32-bit quad-core ARM cortex-A7 (ARMv7).
- Raspberry Pi 3: Broadcom BCM2837 SOC with 1.2GHz 64-bit quad-core –A53 with 512 kb shared L2 cache (64-bit instruction set ARMv8).

3.3.4 Linux

Linux is a free open source operating system and it belongs to the Unix operating systems. Actually Linux means the kernel itself which is the heart of the operating system and handles the communication between the user and hardware. Normally Linux is used to refer to the whole Linux distribution. (Upton, E. & Halfacree, G. 2012, 28.). Linux distribution is a collection of software based on the Linux Kernel. It consists of the GNU-project's components and applications. Because Linux is an open source project, anyone can modify and distribute it. That is the reason why there are many variations of Linux distributions. Most popular distributions are Ubuntu, Red Hat Linux, Debian GNU/Linux and SuSe Linux. (Kuutti, W. & Rantala, A. 2007, 2.).

3.3.5 History of Linux

Linux is a Unix compatible operating system where the operating system's kernel has been reprogrammed. Because of the compatibility most of the free applica-tions programmed for Unix are also available for Linux. In 1973 Unix was repro-grammed in C programming language instead of the assembly code. At that point Unix reached its current outfit. Unix supported multiple users and it was also easy to transfer to new digital machines. Later Unix was given to Universities for further development. (Kuutti, W. & Rantala, A. 2007, 5-6.). Linux got started in the early 1990s when Linus Torvalds got tired of MS-DOS operating system and decided to create a new operating system for the Intel's cheapx86 processors. At the time there was already available Minix operating system for microcomputers. However, mainly for teaching purposes created Minix was not good enough for Torvalds. (Kuutti, W. & Rantala, A. 2007, 5-6.). In October 1991 Linus Torvalds released the first unofficial Linux and the first offi-cial Linux version was released in March 1994. Nowadays Linus Torvalds is still partly developing and supporting the kernel's further development. (Kuutti, W. & Rantala, A. 2007, 5-6.).

3.3.6 Linux Compared to Windows

When comparing Linux and Windows as operating systems, one of the major differences are that Linux is an open-source project and Windows is a closed-source project. In the closed-source project the users sees only the finished product but do not know how it has been done. In open-source projects everything is made fully visible to the public. (Upton, E. & Halfacree, G. 2012, 13-14.). In practice this can be seen in Linux's easy customization for different platforms. This process is called porting. There are several distributions ported to the Rasp-berry Pi's BCM2835 chip. One of the distributions is called Raspbian Wheezy. (Up-ton, E. & Halfacree, G. 2012, 14.).

3.3.7 Raspbian Wheezy

Raspbian Wheezy is a free operating system based on Debian distribution. It is created by a small team of developers who are fans of Raspberry Pi. Raspbian is optimized for the Raspberry Pi's hardware and it comes with over 35 000 packag-es and pre-compiled software. Raspbian is still under active development and it aims to improve the stability and performance of the Debian packages. (Raspbian [Ref. 15.2.2015]). Raspbian is officially recommended for beginners and it includes the graphical desktop environment called LXDE. Raspbian Wheezy is one of the fastest ways to setup and get the RasPi running. (McManus, S. & Cook, M. 2013, 20.).

3.3.8 Programming Languages

There are considerable numbers of programming languages which have been adapted for Raspberry Pi. Python programming language is recommended by The Raspberry Pi foundation especially for the beginners. Basically any programming language which can be compiled for ARMv6 can run on the Raspberry Pi. There-fore the users are not restricted to use only the Python. On the Raspberry Pi there are preinstalled several languages for example C, C++, Java, Scratch and Ruby. (Raspberry Pi Foundation [Ref. 5.2.2015].).

3.3.9 Python Programming Language

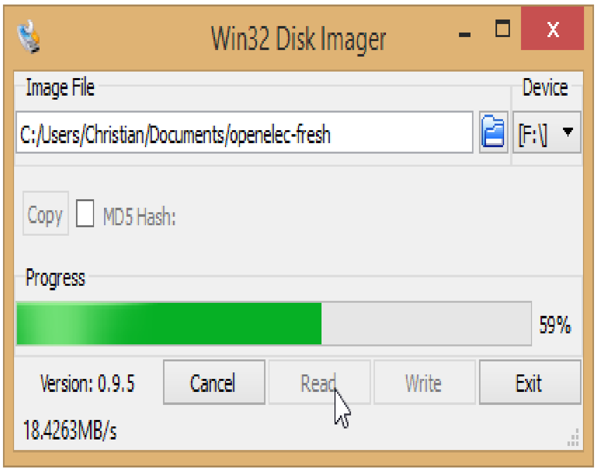
Python programming language is developed in the late 1980s at the National Re-search Institute by Guido van Rossum. Python has grown in popularity, and it is widely used commercially. (Upton, E. & Halfacree, G. 2012, 152.). Python is a flexible and powerful programming language but still it is easy to learn and follow. The clear syntax of Python makes it a valuable tool for users who wants to learn programming. This is one of the reasons why it is recommended by the Raspberry Pi Foundation. Python is published under an open-source license and it is available for different operating systems. Python runs on Linux, OS X and Windows computer systems. (Upton, E. & Halfacree, G. 2012, 152.). Cross-platform support guarantees that the programs which are written in Python are also compatible in other platforms. There are few exceptions where the pro-grams are not compatible. For instance, when the Python is addressed to use the specific hardware such like Raspberry Pi's GPIO. (Upton, E. & Halfacree, G. 2012, 152.).

3.3.10 Installing OS to Raspberry pi B+:

Raspberry Pi doesn't come with an operating system. This is not a weakness, however, rather a feature that means you can choose from a wide of OSs, each of which can be flashed to an SD card (or microSD card for the Raspberry Pi B+) in a few simple steps. Here's how to get a new OS installed and running on your Pi – and how to clone your perfect setup for quick disaster recovery. Operating systems such as the recommended Raspbian, ArchLinux, Risc OS and even Android come ready to run on your Raspberry Pi. I'll show you the two main ways add an operating system – and once you've got your Pi setup how you want it, we'll look at how to clone the card so that it can be restored following errors (or for temporary reuse of your SD card). The following tutorials assume that you have a basic Raspberry Pi package and Windows to manage your SD card writing and cloning. Flash An OS To SD And Boot your Raspberry Pi. Whichever operating system you download for your Raspberry Pi, the process of writing it to an SD card is the same. However, there are some differences in SD card writing between desktop operating systems. You'll also need to ensure that your card is blank and formatted, and at least 2 GB. For Windows users (and also those of you with Mac OS X) the tool of choice for writing a Raspberry Pi OS image to SD card is SD Formatter, from the SD Association. With the card inserted into your computer's card reader, and ensuring you have the correct Drive letter selected in the drop down menu, open the Option menu and select Full (Erase) and On. This ensures that the full capacity of the storage card will be available. Click OK, then Format to begin. To write the disk image, use Win32DiskImager, available from Sourceforge. You may need to run with administrator privileges. Select the correct drive letter for your SD card, browse to the image file and click Write to commence the process. Win32DiskImager will inform you when the data has been written. If writing the disk image seems too much hassle or is beyond your abilities, it is possible to purchase SD cards with Raspbian pre-installed. Booting Raspbian for the first time with Raspbian installed, you'll need to login with the following credentials: Username: pi. Password: raspberry. For other operating systems, check the documentation to find the default login credentials. Remember that the password will not be displayed as you type it; there are no Windows-style * symbols representing the letters. Instead, it will appear that you haven't entered a password. This is a security feature in Linux to prevent people guessing the length of your passphrase.

3.3.11 Disaster Management: Cloning the OS

The point at which you're happy with the installed OS is precisely when you should make a clone of it. This is a full backup image of the SD card saved to your hard disk drive (or second SD card) that can be easily flashed should you run into trouble. Windows users can again employ Win32DiskImager here, ensuring that the perfected Raspberry Pi SD card is inserted in your Windows card reader before launching the utility. Find the Image File field, and browse to the folder where you will be saving your SD card image; in the Device box, select the drive letter of the SD card, and after confirming both are correct, click Read to begin the cloning, following any on-screen instructions. Give the process time to complete before proceeding. Should you need to revert the SD card back to its "perfect" state later on, you'll need to remove all partitions (WIN+R to launch Disk Management, right-click the SD card and select Delete**)** before restoring the image. You might also reformat using SD Formatter, as explained above. Again using Win32DiskImager, browse for the image file, set the Device drive letter and then click Write. Once the process is complete, your SD card will be reverted to the "perfect" configuration you worked on earlier, and your Raspberry Pi should be ready to put to use once more.



Win32 Disk Imager

Image File: C:/Users/Christian/Documents/openelec-fresh

Device: [F:\]

□ MD5 Hash:

Progress: 59%

Version: 0.9.5

18.4263MB/s

3.3.12 Raspberry Pi Configuration

As said earlier Raspberry Pi comes without any peripheral devices. The first thing to do is to unpack RasPi and protect it with an enclosure (Figure 3). Raspberry Pi can be installed to the protective enclosure without using any tools. The enclosure has plastic clips which are holding the Raspberry Pi in its place. After Raspberry Pi has been installed to enclosure and well protected, all the nec-essary peripherals can be attached to it. Just like any other computer, Raspberry Pi needs some basic devices such as display which is connected via the HDMI cable, the mouse and the keyboard, and the internet connection cable. Before plugging the power cable, MicroSD-card should be checked if it is flashed and prepared with an operating system. Also it is recommendable to create a backup folder of the MicroSD-card just in case of complications. The MicroSD-card can be checked with a card-reader. The card-reader can be found from most of the laptops and desktop computers. Insert the MicroSD-card into the card-reader and check that there is something stored in the MicroSD-card. If everything looks good, take the MicroSD-card and plug it into the Raspberry Pi. Now the power cable can be connected. Raspberry Pi does not have any kind of power switch so it will start up immediately when the power cable is connected to it. At the start up text starts to flow on the monitor and shortly after that there appears a configuration menu. The configura-tion menu is called Raspi-config. In Raspi-config it is possible to change some of the settings on Raspberry. Expand Filesystem, where it is necessary to check that RasPi can use the whole memory capacity of the MicroSD-card. Otherwise the memory can run out fast. Internationalisation Options, where it is possible to choose between differ-ent languages and the time zones. Advanced Options, if the internet cable has been plugged in, it is possible to update RasPi to the latest version available. (McManus, S. & Cook, M. 2013, 38.). It is recommendable that users who do not have so much experience with Linux operating systems should choose the English language because then help and advice can be found more easily from the internet. It is possible to get back to the Raspi-config and change the settings also after the first setup by typing the following command into the terminal: sudo raspi-config. After making the changes on the Raspberry Pi's settings, the settings can be ac-cepted by choosing the Finish option. Now the terminal view should appear and it might be asking for the username and the password. The username in Raspian Wheezy is by default **pi** and the password should be **raspberry**. Notice that these are written in small letters. The Linux is letter case sensitive and it will recognize the difference between small and capital letters. The next step is logging in to Raspberry and instead of the graphical environment there will be a command console flashing. However, the graphical environment, or so called desktop view, can be started by entering the command: Startx. Now Raspberry will be loading for a while and a few seconds later there will appear a more user friendly desktop view. It is recommended to learn how to use the command console as it makes some of the actions faster than doing them in the desktop view. So far the basic configurations are made for the Raspberry. There might still be some things that are not working correctly. For instance, the keyboard layout might be defined to be in UK style which is the default keyboard layout setting on Rasp-berry Pi. This can be frustrating and annoying. The layout can be changed easily by opening the LXTerminal which opens the command console. Open the key-board file in the command console with the nano text editor by typing the following command: sudo nano /etc/default/keyboard. The keyboard configuration file (Figure 5) will appear and it can be modified. The keyboard layout can be changed by replacing the XKBLAYOUT value as shown in Figure 5. After the file is edited it can be saved by pressing CTRL + O key combi-nation.

3.3.12.1 Installing the Pi NoIR camera module

The Raspberry Pi's NoIR camera module board comes in anti-static plastic bag. It is fast and easy to install. The camera module can be mounted to the protective case's cover, where is reserved slot for the camera. (Figure 12) It is screwed with two small screws, and the ribbon cable is connected to the Raspberry Pi's camera connection port. The connection port is located between the 3.5mm audio jack and the HDMI socket. The connection port's clip has to be pulled up before plugging the camera module's ribbon cable on its place After mounting the camera module, it is required to enable the camera module from the Raspi-config configuration tool and then Raspberry Pi has to be rebooted so that the changes will take effect.

3.3.12.2 Taking pictures with NoIR camera

In Raspbian there are built in functions for the camera module. With these built in functions it is possible to take pictures and record videos, just to try out proper function of the camera module. One of these built in functions or commands is called "raspistill". Raspistill -v -o first_image.jpg. After typing the command above into terminal a preview window is started up. The preview window is running for 5 seconds, and then Raspberry takes the picture, and saves it to the file called first_image.jpg. Parameters -v stands for verbose information during the run and with the -o parameter it is possible to give filename for the output file.Other simple and useful parameters which can be added into raspistill command are: image width **w,** image height **h,** image quality **q,** flip the image vertically **vf,** flip the image horizontally **hf,** image rotation **rot.** A complete parameter list can be found from the RaspiCam documentation. (RaspiCam Documentation. 2013. 5-18.).

3.3.12.3 Creating Python script for taking pictures

First things to consider before creating the script which takes the picture and stores it automatically are: where the picture is stored, finding the right parameters for the picture so that the image quality and size does not suffer too much. After a while, some limitations for the pictures are found. The size and quality are reduced to minimize the picture size on the hard drive. The Quality of 75% and the resolution of 1280x720 pixels are sufficient. With these parameters the picture size on the hard drive is around 500KB. That is good starting point, and trade-off be-tween picture quality and available space for picture saving. All the pictures which are taken by the Python script will be saved to the own folder with current timestamp filename. The folder is located at /var/www/camera/. Apache2 is hosting the folder so that the pictures are available on the website. Creating the script starts with placing the shebang information and importing the necessary libraries. These libraries are datetime, picamera and time.

3.3.13 Architecture

The processor at the heart of the Raspberry Pi system is a Broadcom BCM2837 system-on-chip (SoC) multimedia processor. This means that the vast majority of the system's components, including its central and graphics processing units along with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB memory chip at the centre of the board. It's not just this SoC design that makes the BCM2837 different to the processor found in your desktop or laptop, however. It also uses a different instruction set architecture (ISA), known as ARM. The BCM2837 SoC, located beneath a Hynix memory chip Developed by Acorn Computers back in the late 1980s, the ARM architecture is a relatively uncommon sight in the desktop world. Where it excels, however, is in mobile devices: the phone in your pocket almost certainly has at least one ARM-based processing core hidden away inside. Its combination of a simple reduced instruction set (RISC) architecture and low power draw make it the perfect choice over desktop chips with high power demands and complex instruction set (CISC) architectures. The ARM-based BCM2837 is the secret of how the Raspberry Pi is able to operate on just the 5V 1A power supply provided by the onboard micro-USB port. It's also the reason why you won't find any heat-sinks on the device: the chip's low power draw directly translates into very little waste heat, even during complicated processing tasks. It does, however, mean that the Raspberry Pi isn't compatible with traditional PC software. The majority of software for desktops and laptops is built with the x86 instruction set architecture in mind, as found in processors from the likes of AMD, Intel and VIA. As a result, it won't run on the ARM-based Raspberry Pi. The BCM2837 uses a generation of ARM's processor design known as ARM11, which in turn is designed around a version of the instruction set architecture known as ARMv6. This is worth remembering: ARMv6 is a lightweight and powerful architecture, but has a rival in the more advanced ARMv7 architecture used by the ARM Cortex family of processors. Software developed for ARMv7, like software developed for x86, is sadly not compatible with the Raspberry Pi's BCM2837—although developers can usually convert the software to make it suitable. That's not to say you're going to be restricted in your choices. As you'll discover later in the book, there is plenty of software available for the ARMv6 instruction set, and as the Raspberry Pi's popularity continues to grow, that will only increase. In this book, you'll also learn how to create your own software for the Pi even if you have no experience with programming

3.3.14 HDMI video

A better-quality picture can be obtained using the HDMI (High Definition Multimedia Interface) connector, the only port found on the bottom of the Pi. Unlike the analogue composite connection, the HDMI port provides a high-speed digital connection for pixel-perfect pictures on both computer monitors and high-definition TV sets. Using the HDMI port, a Pi can display images at the Full HD 1920x1080 resolution of most modern HDTV sets. At this resolution, significantly more detail is available on the screen. If you're hoping to use the Pi with an existing computer monitor, you may find that your display doesn't have an HDMI input. That's not a disaster: the digital signals present on the HDMI cable map to a common computer monitor standard called DVI (Digital Video Interconnect). By purchasing an HDMI-to-DVI cable, you'll be able to connect the Pi's HDMI port to a monitor with DVI-D connectivity. If your monitor has a VGA input—a D-shaped connector with 15 pins, typically coloured silver and blue—the Raspberry Pi can't connect to it. Adapters are available that will take in a digital DVI signal and convert it to an analogue VGA signal, but these are expensive and bulky. The best option here is simply to buy a more-modern monitor with a DVI or HDMI input.

3.3.15 Flashing the SD card

To prepare a blank SD card for use with the Raspberry Pi, you'll need to flash an operating system onto the card. While this is slightly more complicated than simply dragging and dropping files onto the card, it shouldn't take more than a few minutes to complete. Firstly, you'll need to decide which Linux distribution you would like to use with your Raspberry Pi. Each has its advantages and disadvantages. Don't worry if you change your mind later and want to try a different version of Linux: an SD card can be flashed again with a new operating system at any point. The most up-to-date list of Linux releases compatible with the Pi is available from the Raspberry Pi website at The Foundation provides BitTorrent links for each distribution. These are small files that can be used with BitTorrent software to download the files from other users. Using these links is an efficient and fast way to distribute large files, and keeps the Foundation's download servers from becoming overloaded. To use a BitTorrent link, you'll need to have a compatible client installed. If you don't already have a BitTorrent client installed, download one and install it before trying to download the Raspberry Pi Linux distribution. One client for Windows, OS X and Linux is µTorrent, Which distribution you choose to download is up to you. Instructions in the rest of the book will be based on the Debian Raspberry Pi distribution, a good choice for beginners. Where possible, we'll give you instructions for other distributions as well. Linux distributions for the Raspberry Pi are provided as a single image file, compressed to make it faster to download. Once you've downloaded the Zip archive (a compressed file, which takes less time to download than the uncompressed files would) for your chosen distribution, you'll need to decompress it somewhere on your system. In most operating systems, you can simply double-click the file to open it, and then choose Extract or Unzip to retrieve the contents. After you've decompressed the archive, you'll end up with two separate files. The file ending in sha1 is a hash, which can be used to verify that the download hasn't been corrupted in transit. The file ending in img contains an exact copy of an SD card set up by the distribution's creators in a way that the Raspberry Pi understands. This is the file that needs to be flashed to the SD card.

3.3.16 GPIO Port

The Raspberry Pi's GPIO port is located on the top-left of the printed circuit board, labelled P1. It's a 54-pin port, fitted with two rows of 13 male 2.54 mm headers at the factory. The spacing of these headers is particularly important: 2.54 mm pin spacing (0.1 inches in imperial measurements) is a very common sight in electronics, and is the standard spacing for prototyping platforms that include stripboards and breadboards. Each pin of the GPIO port has its own purpose, with several pins working together to form particular circuits. The layout of the GPIO port. The Raspberry Pi's GPIO port and its pin definitions Pin numbers for the GPIO port are split into two rows, with the bottom row taking the odd numbers and the top row the even numbers. It's important to keep this in mind when working with the Pi's GPIO port: most other devices use a different system for numbering pins, and because there are no markings on the Pi itself, it's easy to get confused as to which pin is which. Although the Pi's GPIO port provides a 5 V power supply, tapped from the incoming power on the micro-USB hub, on Pin 2, the Pi's internal workings are based on 3.3 V logic. This means that the components on the Pi work from a 3.3 V power supply. If you're planning on creating a circuit that will interface with the Pi through its GPIO port, make sure you are using components compatible with 3.3 V logic or are passing the circuit through a voltage regulator before it reaches the Pi. Connecting a 5 V supply to any pin on the Raspberry Pi's GPIO port, or directly shorting either of the power supply pins (Pin 1 and Pin 2) to any other pin will result in damage to the Pi. Because the port is wired directly to pins on the Broadcom BCM2837 SoC processor, you will not be able to repair any damage you do to it. Always be extra careful when working around the GPIO port. The GPIO port provides seven pins for general-purpose use by default: Pin 11, Pin 12, Pin 13, Pin 15, Pin 16, Pin 18 and Pin 22. Additionally, Pin 7—while defaulting to providing a clock signal for general purpose use—can also be used as a generalpurpose pin, giving eight pins in total. These pins can be toggled between two states: high, where they are providing a positive voltage of 3.3 V; and low, where they are equal to ground or 0 V. This equates to the 1 and 0 of binary logic, and can be used to turn other components on or off. You'll learn more about this later in the chapter. The Pi's internal logic operates at 3.3 V. This is in contrast to many common microcontroller devices, such as the popular Arduino and its variants, which typically operate at 5 V. Devices designed for the Arduino may not work with the Pi unless a level translator or optical isolator is used between the two. Likewise, connecting pins on a 5 V microcontroller directly to the Raspberry Pi's GPIO port will not work and may permanently damage the Pi. In addition to these general-purpose pins, the GPIO port has pins dedicated to particular buses. These buses are described in the following subsections

4. CONCLUSIONS

In this project we introduced machine learning technology for traffic control. This project has two major phases:

- Blinking of traffic light according to the traffic level present on the road.
- This system manages traffic when any necessary vehicle comes. For example ambulance, fire brigade.Proposed system will have wider future scope that user can get traffic information on mobile phone.

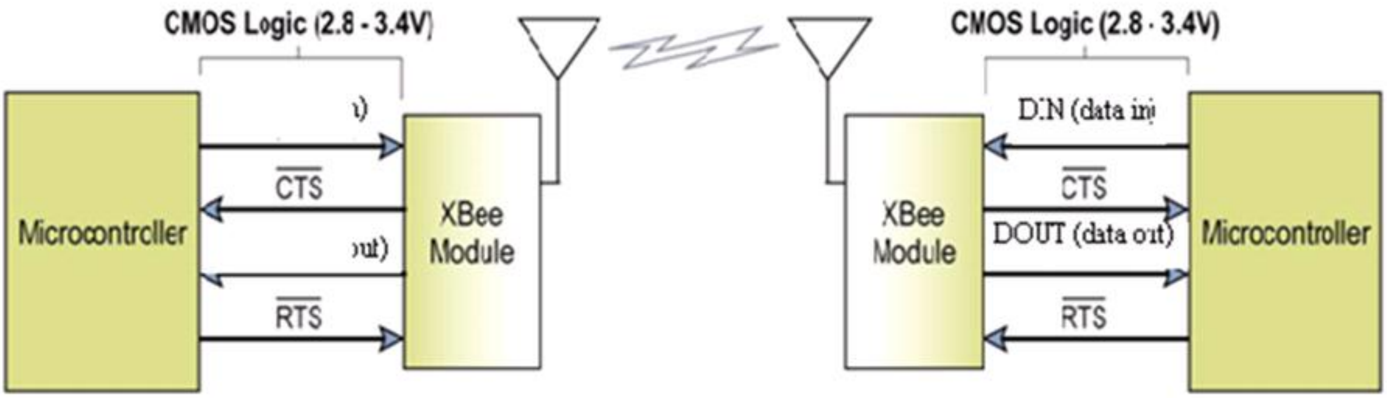


Figure 4: ZigBee UART Dataflow

5. ACKNOWLEDGMENTS

This work is an outcome of rigorous work done under the supervision and guidance of our project guides. We like to pay our sincere thanks to them, for their kind support and cooperation.

6. REFERENCES

[1] Michigan Traffic Safety Summit – February 28, 2006 – "Traffic safety is a health issue saving lives is ultimate measure of success."

[2] https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812013. The Economic and Societal Impact Of Motor Vehicle Crashes, 2010 (Revised).

[3] Automated traffic monitoring system using computer vision - 2016 International Conference on ICT in Business Industry Government (ICTBIG) [4] Danda B. Rawat and Kayhan Zrar Ghafoor, "Smart Cities Cybersecurity and Privacy", Elsevier Press, ISBN: 9780128150320, Dec. 2018

[4] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[5] S. Jeschke, C. Brecher, H. Song and Danda B. Rawat (Eds.), "Industrial Internet of Things: Cyber-manufacturing Systems," ISBN 978-3-319-42559-7, Springer, Switzerland, 2016.

[6] Danda B. Rawat and C. Bajracharya, "Vehicular Cyber Physical Systems: Adaptive Connectivity and Security," ISBN 978-3 -319-44494-9, Springer, 2016