



Smart Toll Collection System on highways using embedded systems

Communication and information engineering

Mansoura University

2019 Graduates

Supervised by:

DR./ Mahmoud Saafan

ENG./ Hossam M. Balaha

Team Members:

Areej Ayman Helal

Ayah Ahmed Arfali

Esraa Shaban Sabry

Manar Ahmed Hesham

May Mohamed Abdel-Salam

➤ Acknowledgement:

First of all, all praise goes to the almighty Allah, for his great care, guidance in every step in our life and for the countless gifts we have been offered.

One of these gifts is those people who were assigned to give us a precious hand so we have to thank them.

We owe a particular debt to Prof. Dr. Mohammed Abdel Azim, Dean of Faculty of Engineering, Mansoura University, for his continuous encouragement and support to the students.

Words are not enough to express our deepest thanks and gratitude to our project supervisor Dr. Mahmoud Saafan, for all his support, practice, supervision and sincere advices to bring this project into light.

We're greatly indebted to Eng. Hossam Magdy Balaha , for his help, objective criticism and encouragement that helped us complete this work.

We have to thank all our teachers throughout our years of study, for guiding us, inspiring us, and for making us what we are today.

Finally , from our hearts we would like to thank our parents, we don't know how we would have made it this far without you, we appreciate your efforts and love in bringing us up to be better individuals and we promise we will work hard to make you proud, Thanks for Everything.

Lastly we acknowledge all the staff members of the department of Communication and Information Engineering (CIE).

➤ **Table of contents:**

Abstract	1
Chapter One: Introduction	3
Chapter Two: Microcontrollers	
2.1. Raspberry Pi	9
2.1.1. Raspberry Pi Models	9
2.1.2. Differences between Raspberry Pi models	10
2.1.3. Hardware Interfaces	11
2.1.4. Performance	12
2.1.5. Software (Operating System)	12
2.1.6. Raspberry Pi 3 Model B+	14
2.1.7. Raspbian Software Setup Steps	15
2.1.8. Accessing the raspberry remotely using Putty	24
2.1.9. Accessing the raspberry remotely using VNC	26
2.2. Arduino	30
2.2.1. Advantages of Arduino	30
2.2.2. Specifications of Arduino	31
2.2.3. Types of Arduino	32
2.2.4. Programming an Arduino	34
2.2.5. Download the Arduino Software (IDE)	35
Chapter Three: Web Server (XAMPP)	
3.1. Server	38
3.2. Web server	39
3.3. Web Server Function	39
3.4. Examples of Web Servers	40
3.5. Apache Web	41
3.6. XAMPP	41
3.6.1. Application areas of XAMPP	42
3.6.2. Installing XAMPP	42
3.6.3. The XAMPP Control Panel	46
3.6.4. Starting modules	47
3.6.5. Setting up XAMPP	47
3.6.6. Module administration	49
3.6.7. Testing XAMPP installation	50

Chapter Four: Database (XAMPP)

4.1. The basic processes that are supported by a DBMS	53
4.2. Conceptual Modeling	54
4.3. The Entity-Relationship Model	55
4.4. Short definitions of some of the basic terms that are used for describing important entity-relationship concepts	55
4.5. Entity-Relation Diagram	56
4.5.1. Symbols used in entity-relationship diagrams include	56
4.6. Database languages	56
4.7. SQL	57
4.7.1. SQL Command	57
4.8. Creating Database Using XAMPP	60

Chapter Five: Sensors

5.1. Factors of choosing a sensor	66
5.2. Classification of Sensors	66
5.2.1. Classification based on property	67
5.3. IR Sensor	67
5.3.1. Advantages	68
5.3.2. Disadvantages	68
5.3.3. How IR sensors work	68
5.3.4. Applications	69
5.3.5. Design of IR Circuit	69
5.3.6. IR with Raspberry pi	71
5.4. Laser Sensor	74
5.4.1. Different types of laser sensors	74
5.4.2. Photoelectric Laser Sensor Principles	75
5.4.3. Models	75
5.5. Load Cell	77
5.5.1. Connections	78

Chapter Six: Barcodes

6.1. Bar Code Design	81
6.1.1. Linear bar code	81
6.1.2. 2-D symbology	82
6.2. Symbologies	82
6.3. Bar Coding System Requirements	84
6.4. Key Benefits of Bar Coding	85
6.5. Common Bar Code Applications	85
6.6. Bar Code Print Technologies	87

Chapter Seven: PHP &MySQL

7.1. Usage	92
7.2. Applications of PHP Scripts	92
7.3. How to run a php file in XAMPP	93
7.4. Connecting to the database using PHP and MySQL	94
7.4.1. Functions used for database	94
7.5. Other PHP Functions used in our code	97

Chapter Eight: Python

8.1. IDLE	106
8.2. Basic Python usage	107
8.3. Python Libraries we used	110
8.3.1. Time	110
8.3.2. RPI.GPIO	110
8.3.3. Requests	112
8.3.4. Pygame	115

Chapter Nine: System

9.1. System Explanation	119
9.1.1. Raspberry Pi Connections	119
9.1.2. Raspberry Pi Code	120
9.1.3. XAMPP Server	122
9.1.4. MySQL Database	122
9.1.5. Apache Server	125
9.2. Features	128
9.2.1. Traffic Control	131
9.2.2. Web User/Admin Interface (Website).....	133
9.2.3. Our project prototype	138

Chapter Ten: Conclusion and Future Work

10.1. Conclusion	140
10.2. Future Work	142

References	145
-------------------------	-----

➤ **Abstract:**

The number of vehicles plying the highways keeps growing at a steady pace, leading to high maintenance costs. Toll collection was introduced as a means of raising funds for road maintenance, but the traditional method is usually slow and is prone to cause vehicular traffic congestion on the highways. In this book, a framework was proposed for Electronic Toll Collection (ETC) in smart and connected communities. The main components of the intelligent system architecture are the Laser sensor nodes, a Raspberry Pi, a Camera, web and mobile applications.

The Laser Sensor and Camera enable vehicle detection and classification, the Raspberry pi establishes a communication link to the back-end of the system. The central database and the web server are hosted in the Cloud while a mobile application is used for electronic transactions, subscription renewal, notification of toll payments, and for tracking toll payment history. In addition, a web dash board is provided for efficient toll administration.

The implementation of this system will improve the toll collection efficiency in terms of speed and flexibility. Overall, the contribution of this work extends to the domain of Intelligent Transportation System (ITS).

Smart Toll Collection on Highways

Chapter one

Introduction

As highways become more and more important in modern transportation networks, the inefficiencies of traditional toll collection systems tend to be unbearable for users because they often result in long vehicle queues and traffic congestion. ETC systems seek to address this problem by reducing the waiting time on the toll highways. Significant reduction in the delay period will minimize unproductive fuel consumption, air pollution, unnecessary road congestion, and increase road safety. The ETC system was basically designed to ease traffic flow during toll collection, and it has become an important part of ITS. In this era of rapid technology development, conventional systems are being replaced with automated systems.

Automation minimizes human interference, ensures reduction in time and energy, and increases efficiency. The ITS era emerged with the revolution in computer, communication, and embedded system technologies. ETC, as a part of ITS, has been extensively deployed to ameliorate the traffic delay, and it has drastically improved the efficiency of road operations. The use of roads is rapidly increasing as the number of vehicles plying them keeps growing, leading to high maintenance costs. Toll collection was introduced as a means of raising funds for road maintenance, but it is traditionally a slow process prone to cause vehicular traffic congestion.

Several efforts have been focused towards making the toll collection process faster and more transparent. The numerous advantages of the toll collection system make the whole effort worthwhile, a few of which include revenue generation for the maintenance, rehabilitation and reconstruction of roads, as well as road use demand management and control.

Therefore, automating this process is an indispensable task able to bring great benefits, and will help to increase the standard of living, while indirectly contributing to the goal of making cities smarter. In this book, we propose an architecture for ETC in smart and connected communities. The main components of the smart system architecture are: the Laser sensor nodes, a Raspberry Pi, a Camera, web and mobile applications.

Over last decades, electronic toll collection system have been implemented in united states and many other countries with a new improvement in it. By this we don't have to carry a handsome amount of cash with us relates to security as well. This system does not require any manual operation of toll barriers and collection of toll amounts, it is completely automated toll collection system.

When the Laser sensor is triggered the camera captures a picture of the car and sends it to the Raspberry pi, then the pi will send the picture to the server where it will be processed and if a barcode is detected it will search in the database for the car owner account and the toll will be collected, also the picture will be stored in the database for future use. If the barcode is not found in the database the server will send a warning to the police with the picture of the car. There is a web interface so that the administrators can monitor and control the database and have access to the toll history of car users and collect data about them. There is also a user interface where the user can track his toll history and car information.

We will be using load cell sensors and LCDs all over the highway to detect traffic jams and help control the traffic faster by detecting the problem once it happens and informing the authorities.

The load cells are used to detect when the flow of traffic is not normal. When the weight on the load cell is more than a specific weight for more than a certain duration of time that means that there's something wrong, it could be that there is an accident in that location or a traffic jam or a car broke down. In all cases a warning will be sent to the police with the location of the problem and it will be shown on all LCDs on the highway before that location to warn drivers to lower their speed according to their distance from the traffic jam to avoid making the traffic jam worse by arriving there fast.

We will be using Raspberry Pi 3 model B+, XAMPP web server and also for the database. We will use PHP as a server scripting language, SQL as a language for handling the database and Python as a programming language for the Raspberry Pi. We will use Arduino Uno to control the load cells and LCDs and to communicate with the Raspberry Pi. All these will be explained in detail in the next chapters then we will explain how the system works in details.

This new system will require some work to be done before it is installed on highways. First of all, each car number will have to be linked to a unique barcode which will be detected by the system because performing image processing on barcodes is easier, faster and more accurate than performing image processing on plate number especially if the plate number is in Arabic letters and numbers, that is why we choose barcodes rather than trying to process the ordinary plate numbers.

We will have to collect data about all cars and their owners and store them in the database in their respective tables. Each car owner will have to submit all the required information about themselves when they get their car license or when they renew their license. They have to submit their full name, a working mobile number, a valid email address, their national Id and a working bank account with a specific deposit amount which the toll will be collected from, in return they will be given a user account name and a password which will be used to access their account to monitor his toll collection, browse the latest highway news and submit any complaints or suggestions.

So we have to take all old plate numbers and print their unique barcodes on them and that will be a little expensive but in return it will make great profit and make life easier and faster for drivers and make it easier to monitor highways and store data about them.

For the hardware part of the project all old gates have to be replaced with new ones which have no booths for the cars to stop and pay the gate will just be over the road with a camera facing the incoming cars and another one to capture the back of the car when it passes the gate. A laser sensor will be installed before each gate to detect when a car passes so that the camera captures the picture so that the camera is not active all the time to save power and storage space from unimportant pictures.

Each gate will have a control unit next to it with all the required hardware and controllers and it is connected to the server, the internet, the nearest police station, hospital and fire department to send warnings in case of accidents or fires or detecting a car with is reported to be stolen or related to a crime.

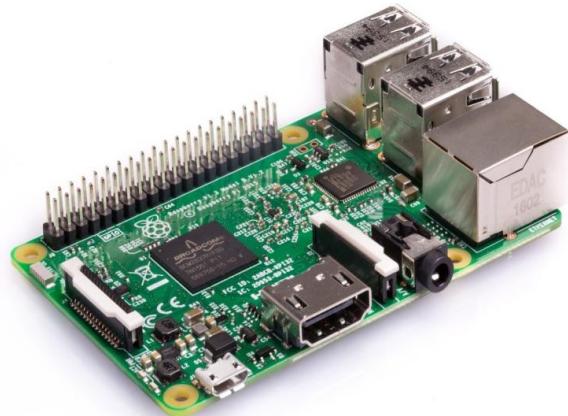
All over the road load cells will be installed every specific distance and next to each load cell we will have an LCD for advertisement in normal conditions but in case the load cell detects a traffic jam for whatever reason it will send a warning to the control unit, which will know its location and print a warning on all LCDs before that location with the location of the traffic jam and advices drivers to slow down their speed respectively to the distance between them and the traffic jam location so they don't reach the location fast and make the traffic jam worse, if they take a little bit more time to reach there they won't stop for a long time because the authorities will solve the problem quickly because they are notified immediately. No other changes will be made to the highways system.

Chapter two

Microcontrollers

➤ 2.1. Raspberry Pi:

Raspberry Pi is the third best-selling computer brand in the world. The Raspberry Pi is a credit card-sized computer that plugs into your TV or display, and a keyboard and mouse. You can use it to learn coding and to build electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high-definition video. The Raspberry Pi is being used by adults and children all over the world to learn programming and digital making.

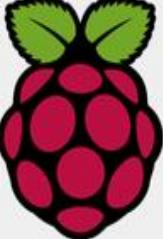


➤ 2.1.1. Raspberry Pi Models:

There have been three generations of Raspberry Pis: Pi 1, Pi 2, and Pi 3, and there has generally been a Model A and a Model B of most generations. Model A is a cheaper variant and tends to have reduced RAM and ports like USB and Ethernet. The Pi Zero is a spinoff of the original (Pi 1) generation, made even smaller and cheaper.

- Raspberry Pi 1 Model A+
- Raspberry Pi 1 Model B+
- Raspberry Pi 2 Model B
- Raspberry Pi 3 Model B
- Raspberry Pi 3 Model B+
- Raspberry Pi Zero
- Raspberry Pi Zero W
- Raspberry Pi Zero WH

➤ **2.1.2. Differences between Raspberry Pi models:**

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	\$35	\$5	\$35	\$35

The Model A+ is the low-cost variant of the Raspberry Pi. It has 512MB RAM (as of August 2016: earlier models have 256MB), one USB port, 40 GPIO pins, and no Ethernet port. The Model B+ is the final revision of the original Raspberry Pi. It has 512MB RAM, four USB ports, 40 GPIO pins, and an Ethernet port. In February 2015, it was superseded by the Pi 2 Model B, the second generation of the Raspberry Pi. The Pi 2 shares many specs with the Pi 1 B+, and originally used a 900MHz quad-core Arm Cortex-A7 CPU and has 1GB RAM. Some recent version of the Pi 2 (v1.2) now use a 900MHz Arm Cortex-A53 CPU.

The Pi 3 Model B was launched in February 2016. It uses a 1.2GHz 64-bit quad-core Arm Cortex-A53 CPU, has 1GB RAM, integrated 802.11n wireless LAN, and Bluetooth 4.1. The Pi 3 Model B+ was launched in March 2018. It uses a 1.4GHz 64-bit quad-core Arm Cortex-A53 CPU, has 1GB RAM, gigabit Ethernet, integrated 802.11ac/n wireless LAN, and Bluetooth 4.2. and is the model we recommend for use in schools, due to its flexibility for the learner.

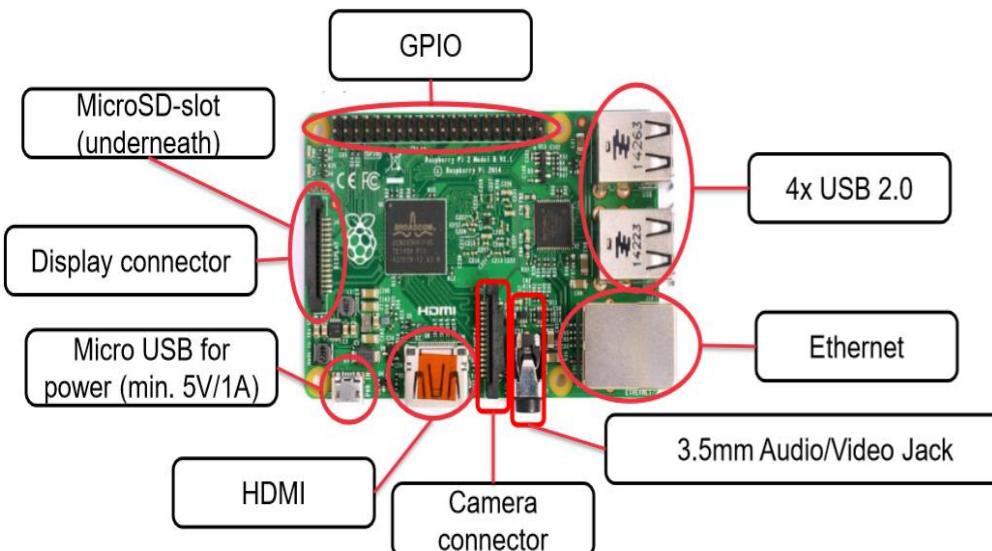
The Pi Zero and Pi Zero W/WH are half the size of a Model A+, with a 1GHz single-core CPU and 512MB RAM, and mini-HDMI and USB On-The-Go ports and a camera connector. The Pi Zero W also has integrated 802.11n wireless LAN and Bluetooth 4.1.

The Pi Zero WH is identical to the Zero W, but comes with a pre-soldered header. The Model A/A+ has one USB port, the Model B has two ports, and the Model B+, Pi 2 Model B, and Pi 3 Model B have four ports. These can be used to connect most USB 2.0 devices. Additional USB devices such as mice, keyboards, network adapters, and external storage can be connected via a USB hub. The Pi Zero and Pi Zero W have a single micro USB port, this requires a USB OTG cable to connect devices such as keyboards or hubs.

The final model (not described in the table above) is the Compute Module, which is intended for industrial applications. It is a small form factor device that connects to a carrier board, for example a circuit board inside an industrial product, and gives manufacturers an easy way to use the Raspberry Pi ecosystem in their own devices.

➤ 2.1.3. Hardware Interfaces:

Depending on the model, the Raspberry Pi has either 40 or 26 dedicated interface pins. In all cases, these include a UART, an I2C bus, a SPI bus with two chip selects, I2S audio, 3V3, 5V, and ground. The maximum number of GPIOs can theoretically be indefinitely expanded by making use of the I2C or SPI bus. There is also a dedicated CSI-2 camera port for the Raspberry Pi Camera Module, and a DSI display port for the Raspberry Pi LCD touchscreen display.



➤ **2.1.4. Performance:**

The GPU provides OpenGL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile encode and decode. The GPU is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general purpose compute and features a bunch of texture filtering and DMA infrastructure. This means that graphics capabilities are roughly equivalent to the original Xbox's level of performance. Overall real-world performance for models A, A+, B, B+, CM, Zero and Zero W is something like a 300MHz Pentium 2, only with much better graphics. The Pi 2 Model B is approximately equivalent to an Athlon Thunderbird running at 1.1GHz: again, it has the much higher-quality graphics that come from using the same GPU as in previous models. The Pi 3 Model B is around twice as fast as the Pi 2 Model B, depending on the benchmarks chosen.

➤ **2.1.5. Software (operating system):**

The recommended distribution (distro) is Raspbian, which is specifically designed for the Raspberry Pi and which our engineers are constantly optimising. The OS is stored on the SD card.

➤ **Operating systems:**

➤ **Raspbian**

A free Debian-based OS optimized for Raspberry Pi's hardware, Raspbian comes with all the basic programs and utilities you expect from a general-purpose operating system. Supported officially by the Raspberry foundation, this OS is popular for its fast performance and its more than 35,000 packages.

➤ **Ubuntu MATE**

Ubuntu MATE is a stable and simple OS, which brings a configurable yet still light-on-resources MATE desktop for its users. It is especially good for devices short on hardware specs, making it perfect for Raspberry Pi devices that can't run a composite desktop. MATE desktop comes with essential apps like a file manager, text editor, image viewer, system monitor, document viewer and terminal.

➤ **Snappy Ubuntu**

A lightweight edition of the popular Ubuntu OS aimed for clouds and devices, Snappy Ubuntu Core uses a minimal server image with the same system libraries. Applications run noticeably faster and are more reliable and secure because of the transactional systems management (like Docker); hence the term "Snappy".

➤ **Pidora**

Pidora is a remix of the well-known Fedora operating system for Raspberry Pi. Designed from the latest build of Fedora for the ARMv6 architecture, Pidora allows greater speed, and carries applications and components from the Fedora 20 package set.

➤ **Linutop**

An OS that can be quickly set up on a Raspberry Pi, Linutop uses a Raspbian-base with classic and lightweight XFCE graphical environment. It's also handy for secure professional uses, such as in Kiosks casting public access or in embedded systems like electronic devices.

➤ **SARPi**

Short for "Slackware ARM on a Raspberry Pi", SARPi is a community product of Slackware Linux enthusiasts. Considered widely as one of the best OS choice for Raspberry Pi, this can be installed on an 8 GB SD card. Although the ARM version doesn't support all the apps, but most applications (including essential ones) have been ported for the ARM architecture.

➤ **Arch Linux ARM**

A version of Arch Linux ported for ARM computers, Arch Linux ARM offers versions 6 and 7 for Raspberry Pi and Raspberry Pi 2 respectively. Its design philosophy promotes simplicity and user-centrism, ensuring that Linux users are in full control of the system.

➤ 2.1.6. Raspberry Pi 3 Model B+ : (which we will be using)

The Raspberry Pi 3 Model B features a quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz. This puts the Pi 3 roughly 50% faster than the Pi 2. Compared to the Pi 2, the RAM remains the same – 1GB of LPDDR2-900 SDRAM, and the graphics capabilities, provided by the VideoCore IV GPU, are the same as they ever were. As the leaked FCC docs will tell you, the Pi 3 now includes on-board 802.11n WiFi and Bluetooth 4.0. WiFi, wireless keyboards, and wireless mice now work out of the box.

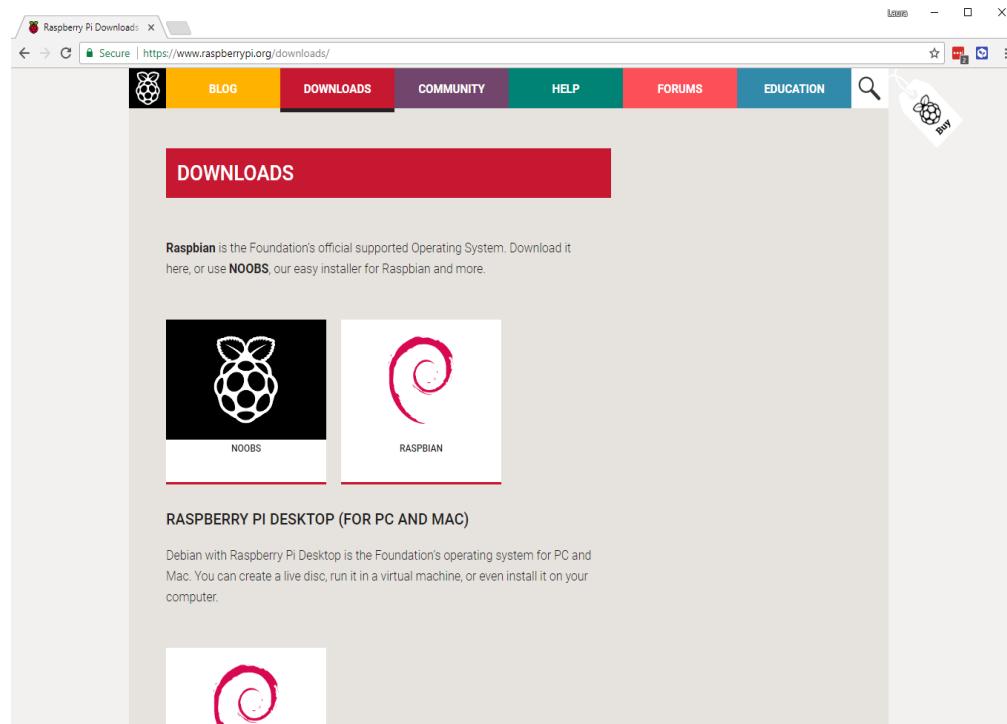
➤ Specification:

- **SoC:** Broadcom BCM2837 (roughly 50% faster than the Pi 2). Built specifically for the new Pi 3, the Broadcom BCM2837 system-on-chip (SoC) includes four high-performance ARM Cortex-A53 processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory, a VideoCore IV graphics processor, and is linked to a 1GB LPDDR2 memory module on the rear of the board.
- **CPU:** 1.2 GHZ quad-core ARM Cortex A53 (ARMv8 Instruction Set)
- **GPU:** Broadcom VideoCore IV @ 400 MHz
- **Memory:** 1 GB LPDDR2-900 SDRAM
- **USB ports:** 4
- **Network:** 10/100 MBPS Ethernet, 2.4GHz 802.11n Wireless LAN, Bluetooth 4.0
- **Bluetooth:** Bluetooth 4.1 Classic, Bluetooth Low Energy
- **Storage:** microSD
- **GPIO:** 40-pin header, populated. The Raspberry Pi 3 features the same 40-pin general-purpose input-output (GPIO) header as all the Pis going back to the Model B+ and Model A+. Any existing GPIO hardware will work without modification; the only change is a switch to which UART is exposed on the GPIO's pins, but that's handled internally by the operating system.
- **Ports:** HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- **Antenna:** There's no need to connect an external antenna to the Raspberry Pi 3. Its radios are connected to this chip antenna soldered directly to the board, in order to keep the size of the device to a minimum. Despite its diminutive stature, this antenna should be more than capable of picking up wireless LAN and Bluetooth signals – even through walls.

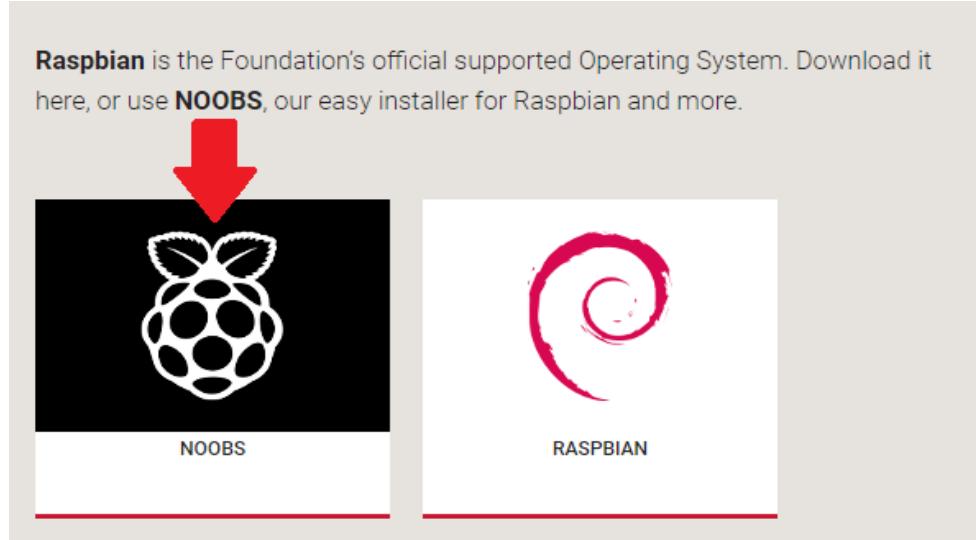
➤ 2.1.7. Raspbian Software Setup Steps:

1) Setup your SD card:

- Using the NOOBS software is the easiest way to install Raspbian on the SDcard.
- Visit the Raspberry Pi downloads page (<https://www.raspberrypi.org/downloads/>).



- Click the box linking to the NOOBS les.



- The simplest option is to download the zip archive of the files. Make sure to pay attention to where you save the archive, so that you can find it again quickly.

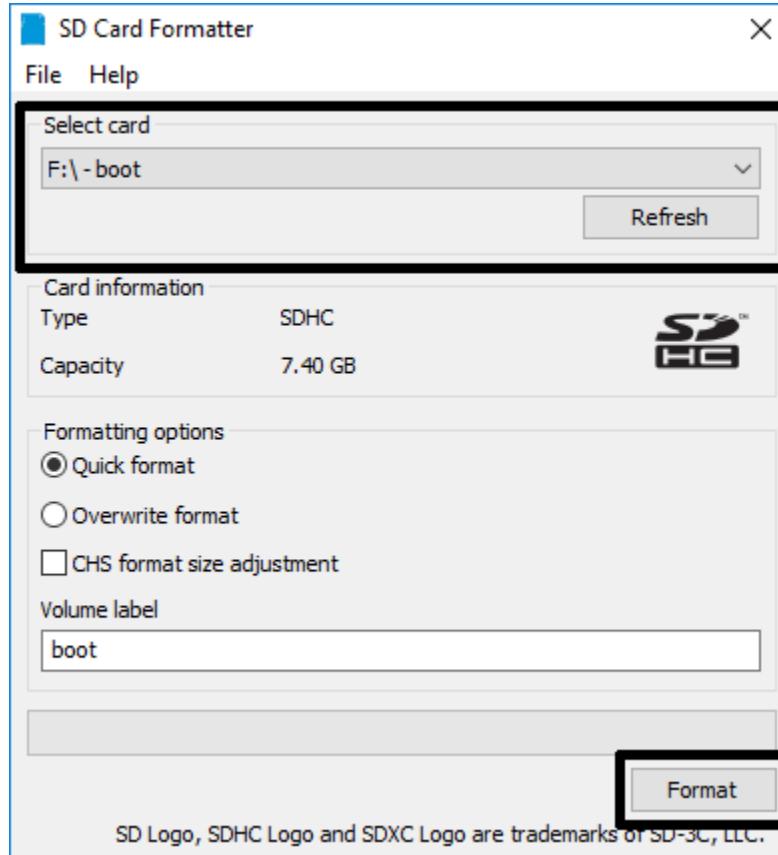
NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.

A screenshot of a website page for NOOBS Lite. It features a central "NOOBS" section with the text "Offline and network install", "Version: 2.4.", and "Release date: 2017-01-12". Below this are two download buttons: "Download Torrent" and "Download ZIP". A large red arrow points down to the "Download ZIP" button. On either side of the central section are two smaller images of the Raspberry Pi logo on a card.

2) Format the SD card:

Anything that's stored on the SD card will be overwritten during formatting. So if the SD card currently has any files on it, e.g. from an older version of Raspbian, back these files up first to not lose them permanently.

- Visit the SD Association's website and download SD Formatter for Windows or Mac.
- Follow the instructions to install the software.
- Insert your SD card into the computer or laptop's SD card slot.
- In SD Formatter, select your SD card, and the format the card.



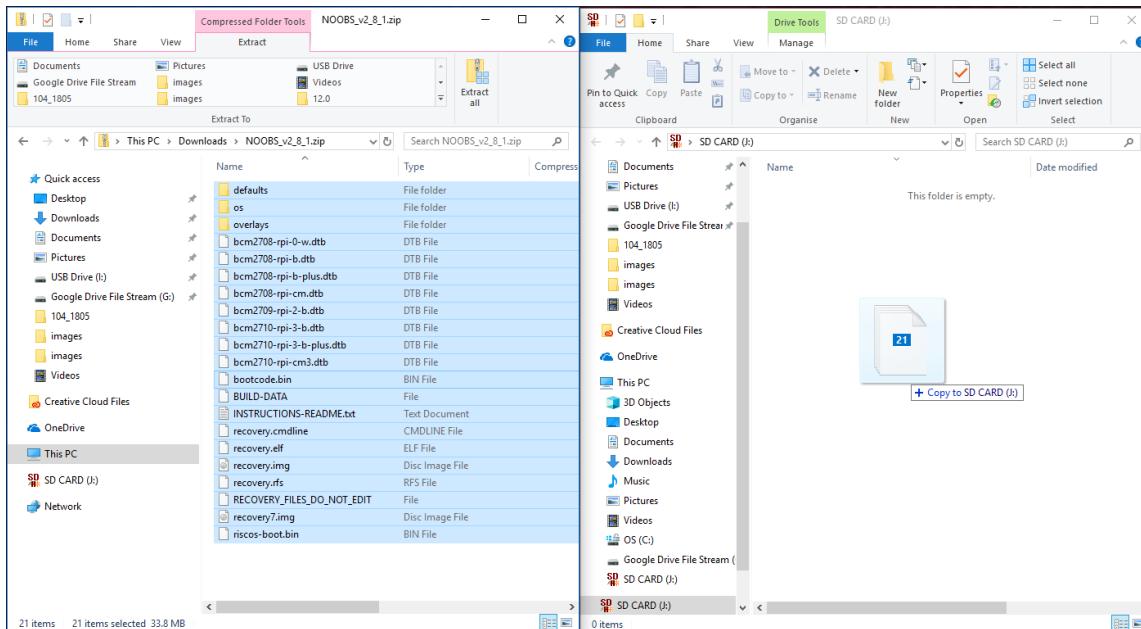
3) Extract NOOBS from the zip archive

Next, extract the files from the NOOBS zip archive you downloaded from the Raspberry Pi website.

- Find the downloaded archive — by default, it should be in your Downloads folder.
- Double-click on it to extract the files, and keep the resulting Explorer/Finder window open.

4) Copy the files

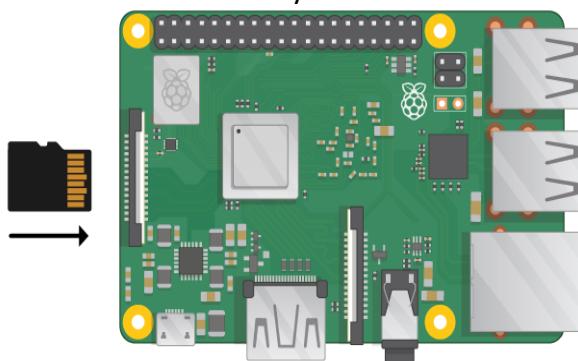
- Now open another Explorer/Finder window and navigate to the SD card. It's best to position the two windows side by side.
- Select all the files in the NOOBS folder and drag them into the SD card window to copy them to the card.
- Once the files have all been copied over, you can eject the SD card.



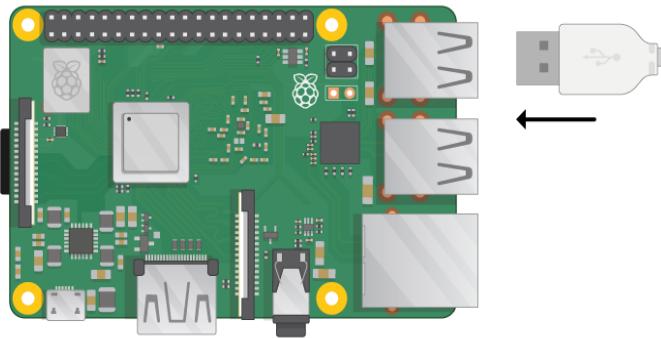
5) Connect your Raspberry Pi

Get everything connected. It's important to do this in the right order, so that all the components are safe.

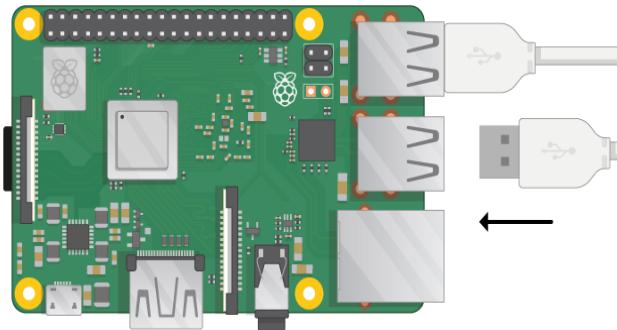
- Insert the SD card you've set up with Raspbian (via NOOBS) into the micro SD card slot at the underside of your Pi.



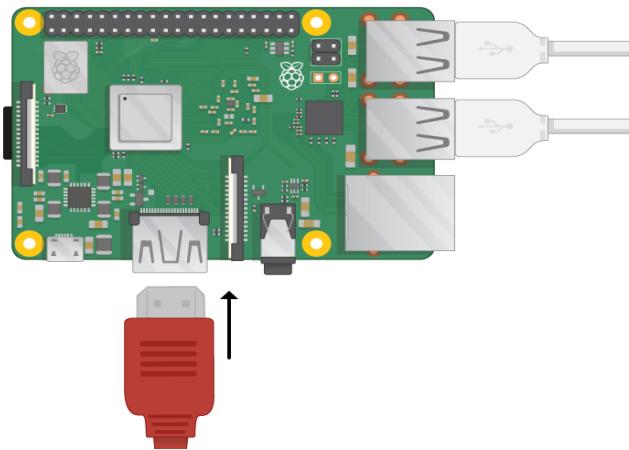
- ii. Find the USB cable for the mouse, and connect the mouse to a USB port on the Raspberry Pi (it doesn't matter which one).



- iii. Connect the keyboard in the same way.



- iv. Look at the HDMI port on the Raspberry Pi — notice that it has a large, flat side on top.

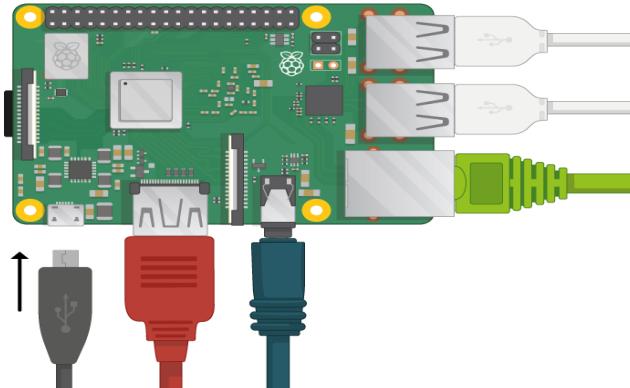


- v. Make sure the screen is plugged into a wall socket and turned on. Use a cable to connect the screen to the Pi's HDMI port — use an adapter if necessary.

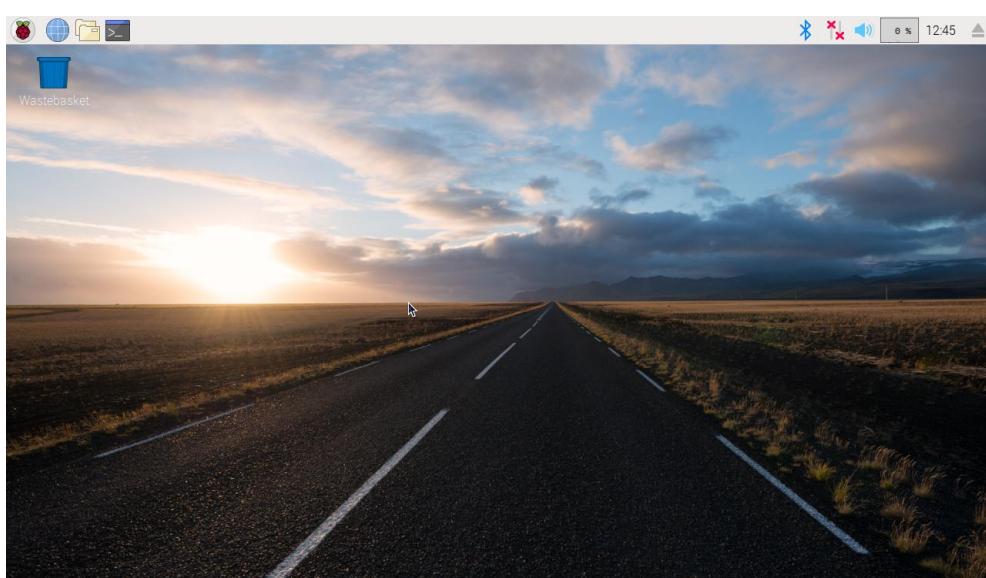
6) Start up your Raspberry Pi

Raspberry Pi doesn't have a power switch: as soon as it is connected to a power outlet, it will turn on. Notice that the Pi's micro USB power port has a longer flat side on top.

- i. Plug a micro USB power supply into a socket and connect it to your Pi's power port.



- ii. A red LED should light up on the Raspberry Pi, which indicates that the Pi is connected to power. As it starts up (this is also called booting), raspberries will appear in the top left-hand of your screen.
- iii. After a few seconds the Raspbian Desktop will appear.



7) Finish the setup

When you start your Raspberry Pi for the first time, the Welcome to Raspberry Pi application will pop up and guide you through the initial setup.



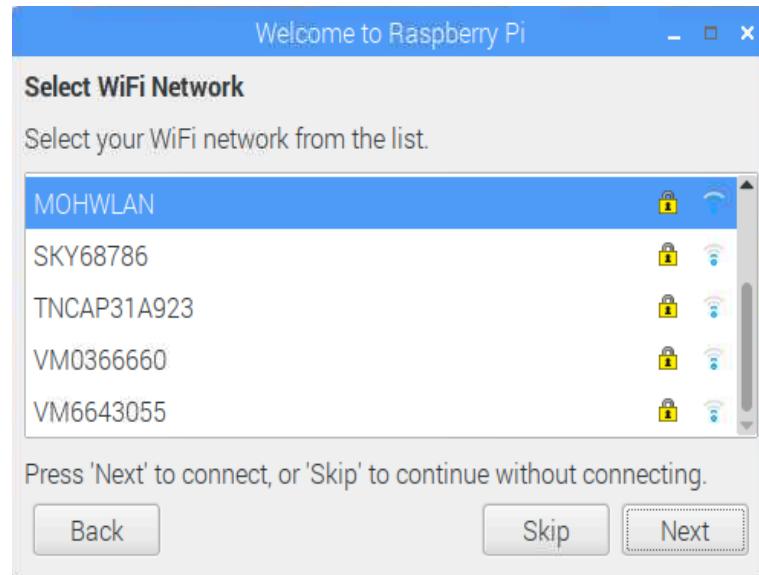
- i. Click Next to start the setup.
- ii. Set Country, Language, and Timezone, then click Next again.



- iii. Enter a new password for the Raspberry Pi and click Next.

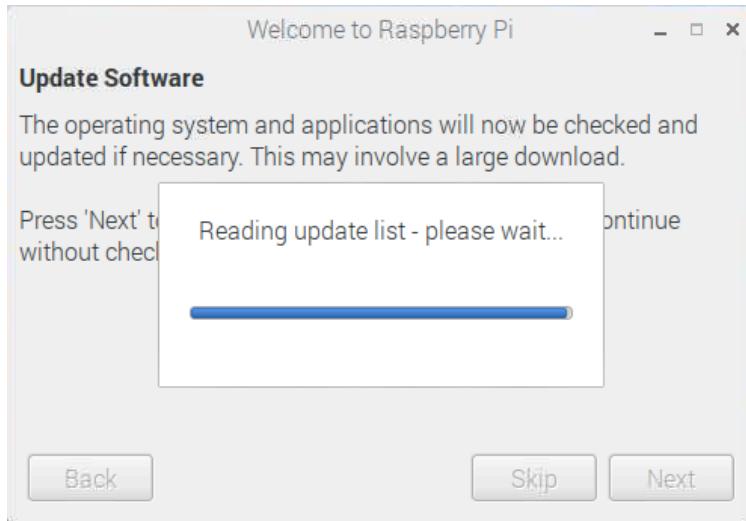


- iv. Connect to Wi-Fi network by selecting its name, entering the password, and clicking Next.



Note: if your Raspberry Pi model doesn't have wireless connectivity, you won't see this screen.

- v. Click Next let the wizard check for updates to Raspbian and install them (this might take a little while).



- vi. Click Done or Reboot to finish the setup.

Note: you will only need to reboot if that's necessary to complete an update.



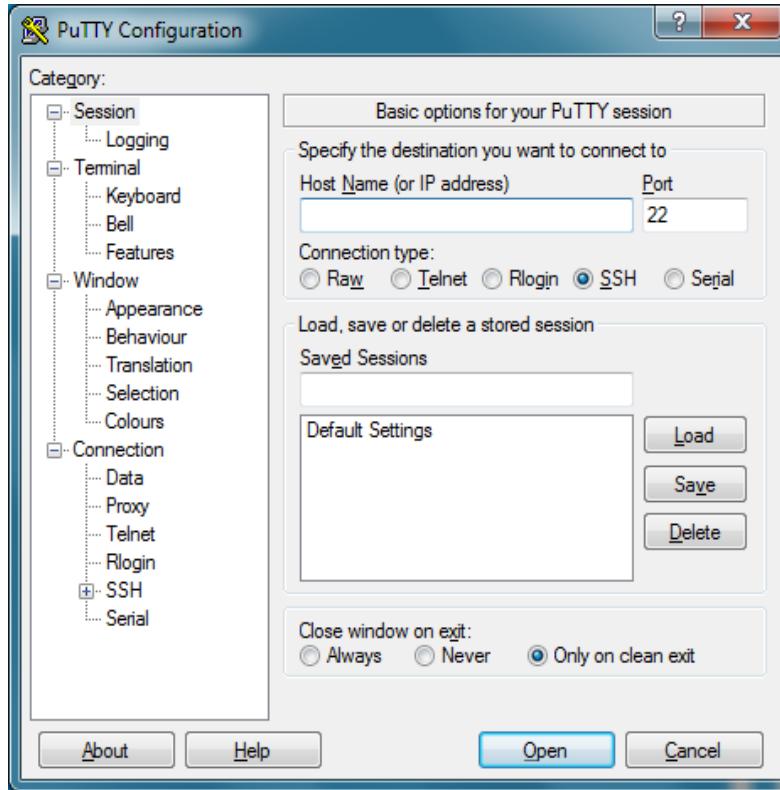
- vii. Now the setup is completed.

➤ 2.1.8. Accessing the raspberry remotely using Putty:

1) Add Raspberry Pi as a host

PuTTY does not include an installer package: it is a stand-alone .exe file. To download PuTTY use this link (<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>)

When it's run, configuration screen below:

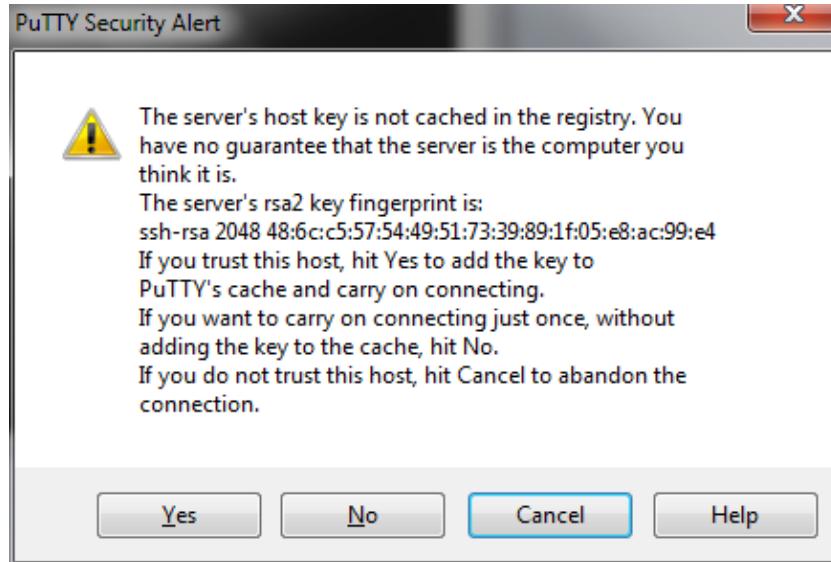


Type the IP address of the Pi into the Host Name field and click the Open button. If nothing happens and eventually a message saying Network error: Connection timed out appears, it is likely that you have entered the wrong IP address for the Pi.

If you do not know the IP address, type hostname -l in the Raspberry Pi command line.

2) Connect

When the connection works you will see the security warning shown below. You can safely ignore it, and click the 'Yes' button. You will only see this warning the first time PuTTY connects to a Raspberry Pi that it has not seen before.



You will now see the usual login prompt. Log in with the same username and password you would use on the Pi itself. The default login for Raspbian is pi with the password raspberry.

You should now have the Raspberry Pi prompt which will be identical to the one found on the Raspberry Pi itself.

pi@raspberrypi ~ \$

A screenshot of a terminal window titled "pi@raspberrypi: ~". The window shows a standard Linux login process:

```
login as: pi
pi@192.168.10.53's password:
Linux raspberrypi 3.12.25+ #700 PREEMPT Thu Jul 24 17:51:46 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul 27 14:59:17 2014 from 192.168.10.50
pi@raspberrypi ~ $
```

➤ 2.1.9. Accessing the raspberry remotely using VNC:

Sometimes it is not convenient to work directly on the Raspberry Pi. Maybe you would like to work on it from another device by remote control.

VNC (Virtual Network Computing) is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer (running VNC Server) from another computer or mobile device (running VNC Viewer). VNC Viewer transmits the keyboard and either mouse or touch events to VNC Server, and receives updates to the screen in return.

You will see the desktop of the Raspberry Pi inside a window on your computer or mobile device. You'll be able to control it as though you were working on the Raspberry Pi itself.



VNC Connect from RealVNC is included with Raspbian. It consists of both VNC Server, which allows you to control your Raspberry Pi remotely, and VNC Viewer, which allows you to control desktop computers remotely from your Raspberry Pi should you want to.

You must enable VNC Server before you can use it: instructions for this are given below. By default, VNC Server gives you remote access to the graphical desktop that is running on your Raspberry Pi, as though you were sitting in front of it. However, you can also use VNC Server to gain graphical remote access to your Raspberry Pi if it is headless or not running a graphical desktop. For more information on this, see Creating a virtual desktop, further below.

To download VNC visit this site.

(<https://www.realvnc.com/en/connect/download/vnc/windows/>)

➤ Enabling VNC Server

On your Raspberry Pi, run the following commands to make sure you have the latest version of VNC Connect:

```
sudo apt-get update  
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
```

Now enable VNC Server. You can do this graphically or at the command line.

- **Enabling VNC Server graphically**

- On your Raspberry Pi, boot into the graphical desktop.
- Select Menu > Preferences > Raspberry Pi Configuration > Interfaces.
- Ensure VNC is Enabled.

- **Enabling VNC Server at the command line**

You can enable VNC Server at the command line using raspi-config:

```
sudo raspi-config
```

Now, enable VNC Server by doing the following:

- Navigate to Interfacing Options.
- Scroll down and select VNC > Yes.

➤ Connecting to your Raspberry Pi with VNC Viewer

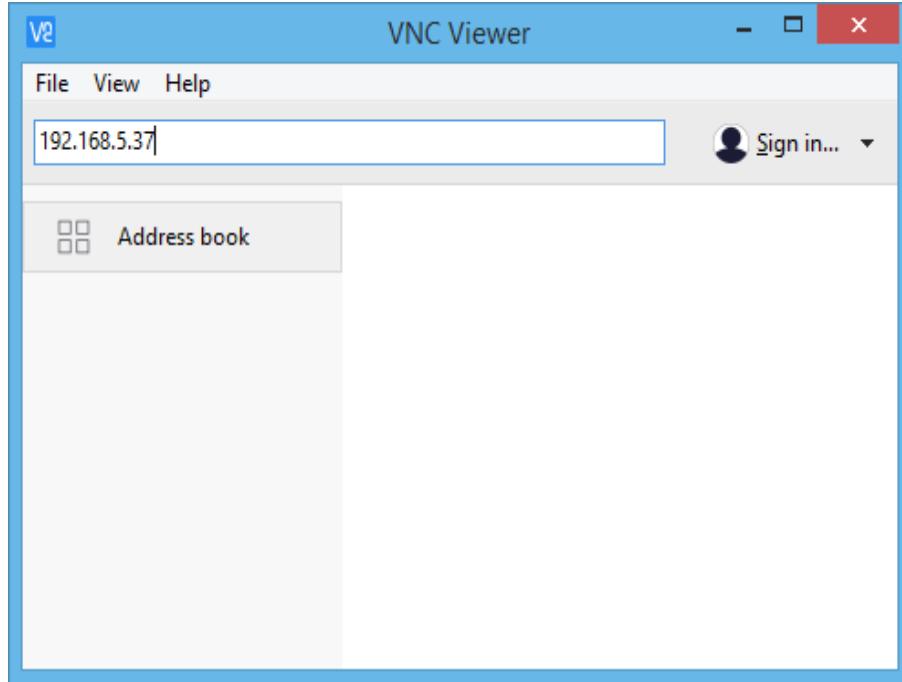
There are two ways to connect to your Raspberry Pi.

- **Establishing a direct connection**

Direct connections are quick and simple providing you're joined to the same private local network as your Raspberry Pi. For example, this might be a wired or wireless network at home, at school, or in the office).

- On your Raspberry Pi (using a terminal window or via SSH) use these instructions or run Establishing a direct connection

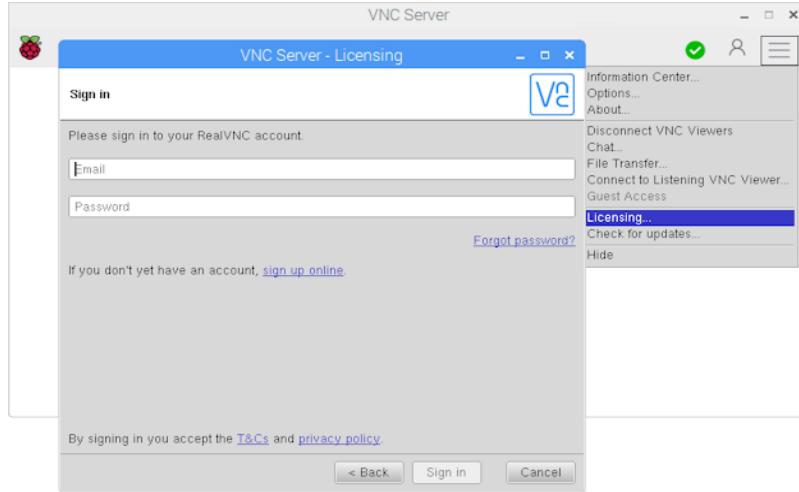
- On the device you'll use to take control download VNC Viewer. For best results, use the compatible app from RealVNC.
- Enter your Raspberry Pi's private IP address into VNC Viewer:



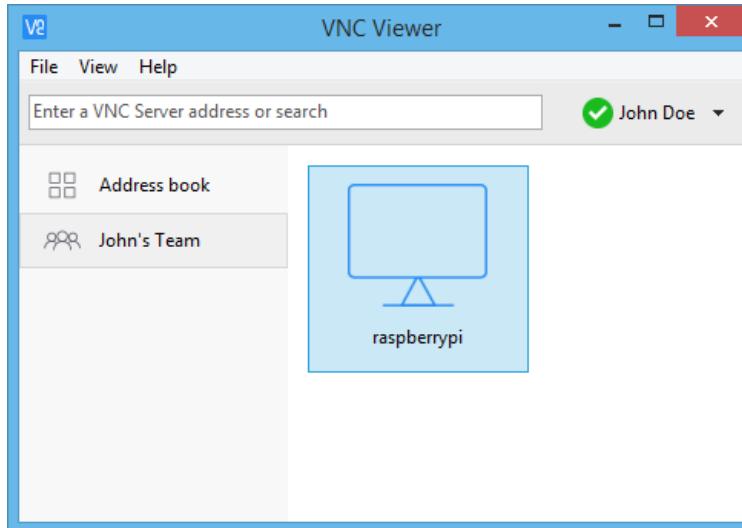
▪ Establishing a cloud connection

You are entitled to use RealVNC's cloud service for free, provided that remote access is for educational or non-commercial purposes only. Cloud connections are convenient and encrypted end-to-end. They are highly recommended for connecting to your Raspberry Pi over the internet. There's no firewall or router reconfiguration, and you don't need to know the IP address of your Raspberry Pi, or provide a static one.

- On your Raspberry Pi, sign in to VNC Server using your new RealVNC account credentials:



- On the device you'll use to take control, download VNC Viewer. You must use the compatible app from RealVNC.
- Sign in to VNC Viewer using the same RealVNC account credentials, and then either tap or click to connect to your Raspberry Pi:



➤ Authenticating to VNC Server

- To complete either a direct or cloud connection, you must authenticate to VNC Server.
- If you're connecting from the compatible VNC Viewer app from RealVNC, enter the user name and password you normally use to log in to your user account on the Raspberry Pi. By default, these credentials are **pi** and **raspberry**.

➤ 2.2. Arduino:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.



Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

➤ 2.2.1. Advantages of Arduino:

- **Inexpensive:** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- **Cross-platform:** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- **Simple, clear programming environment:** The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software:** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware :** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

➤ 2.2.2. Specifications of Arduino:

Arduinos contain a number of different parts and interfaces together on a single circuit board. The design has changed through the years, and some variations include other parts as well. But on a basic board, you're likely to find the following pieces:

- **A number of pins**, which are used to connect with various components you might want to use with the Arduino. These pins come in two varieties:
 - Digital pins, which can read and write a single state, on or off. Most Arduinos have 14 digital I/O pins.
 - Analog pins, which can read a range of values, and are useful for more fine-grained control. Most Arduinos have six of these analog pins.

These pins are arranged in a specific pattern, so that if you buy an add-on board designed to fit into them, typically called a “shield,” it should fit into most Arduino-compatible devices easily.

- **A power connector**, which provides power to both the device itself, and provides a low voltage which can power connected components like LEDs and various sensors, provided their power needs are reasonably low. The power connector can connect to either an AC adapter or a small battery.
- **A microcontroller**, the primary chip, which allows you to program the Arduino in order for it to be able to execute commands and make decisions based on various input. The exact chip varies depending on what type of Arduino you buy, but they are generally Atmel controllers, usually a ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560. The differences between these chips are subtle, but the biggest difference a beginner will notice is the different amounts of onboard memory.
- **A serial connector**, which on most newer boards is implemented through a standard USB port. This connector allows you to communicate to the board from your computer, as well as load new programs onto the device. Often times Arduinos can also be powered through the USB port, removing the need for a separate power connection.
- **A variety of other small components**, like an oscillator and/or a voltage regulator, which provide important capabilities to the board, although you typically don't interact with these directly; just know that they are there.

➤ 2.2.3. Types of Arduino:

•Arduino Uno :

The Arduino UNO R3 is a new board and by comparing with the previous Arduino boards it has some additional features. The Arduino UNO uses the Atmega16U2 instead of 8U2 and it allows faster transfer rate & more memory. There is no need of extra devices for the Linux & Mac and the ability to have the UNO show up as a keyboard, mouse, joystick, etc.



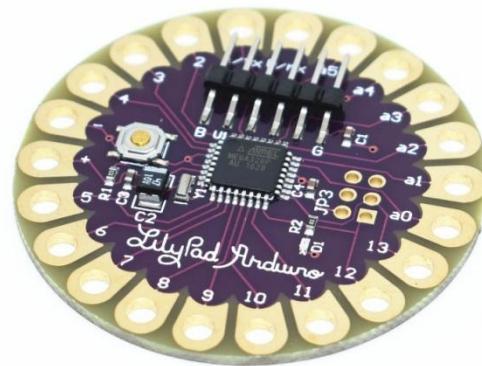
•Arduino Leonardo:

The Leonardo Arduino board is a Microcontroller board and it is based on the ATmega32u4 data sheet. This Arduino board has 20 digital input/out pins and from the total number of pins, seven pins are used for the pulse width modulation output and 12 pins are used as an analog input and there are the 16MHz crystal oscillator, a micro USB connection, RESET pin and power jack.



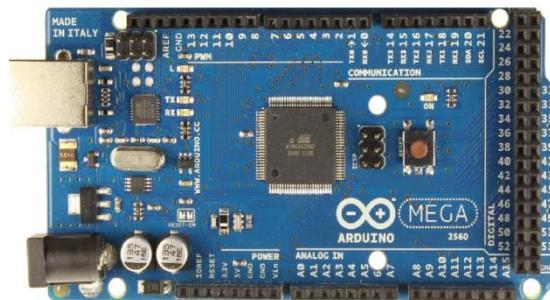
•Arduino LilyPad:

This board is an Arduino Programmable Microcontroller and it is designed to integrate easily into an e-textiles & wearable projects. The other Arduino boards have the same functionality like lightweight, round package designed to minimize snagging and profile, with wide tabs that can be sewn down and connected with conductive thread.



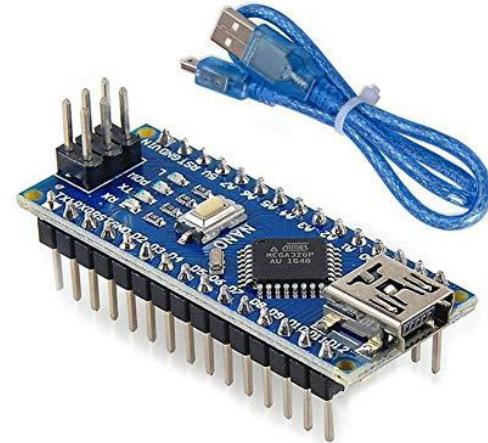
•Arduino Mega:

The Arduino Mega is a type of Microcontroller and it is based on the ATmega2560. It consists of 54 digital input/output pins and from the total pins 14 pins are used for the PWM output, 16 pins are used for the analog inputs, 4 pins are used for the hardware serial port of the UART.



•Arduino Nano:

Arduino Nano is a small, compatible, flexible and breadboard friendly Microcontroller board, developed by Arduino.cc in Italy, based on ATmega328p (Arduino Nano V3.x) / Atmega168 (Arduino Nano V3.x). It comes with exactly the same functionality as in Arduino UNO but quite in small size.



•Arduino Mini :

The Arduino Mini is a small microcontroller board originally based on the ATmega328P intended for use on breadboards and when space is at a premium. Because of its small size, connecting the Arduino Mini is a bit more complicated than a regular Arduino board.



➤ 2.2.4. Programming an Arduino:

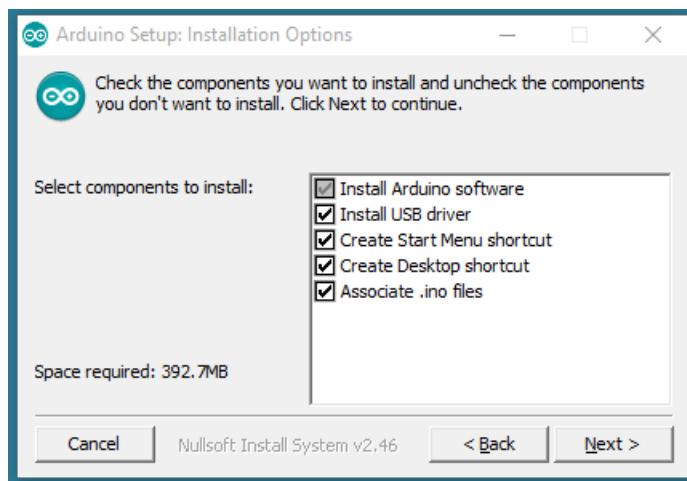
Most Arduino enthusiasts, especially when they are starting out, will choose to use the official integrated development environment (IDE) for the Arduino. The Arduino IDE is open source software which is written in Java and will work on a variety of platforms: Windows, Mac, and Linux. The IDE enables you to write code in a special environment with syntax highlighting and other features which will make coding easier, and then easily load your code onto the device with a simple click of a button.

- **2.2.5. Download the Arduino Software (IDE):**

Get the latest version from this link <https://www.arduino.cc/en/Main/Software>. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

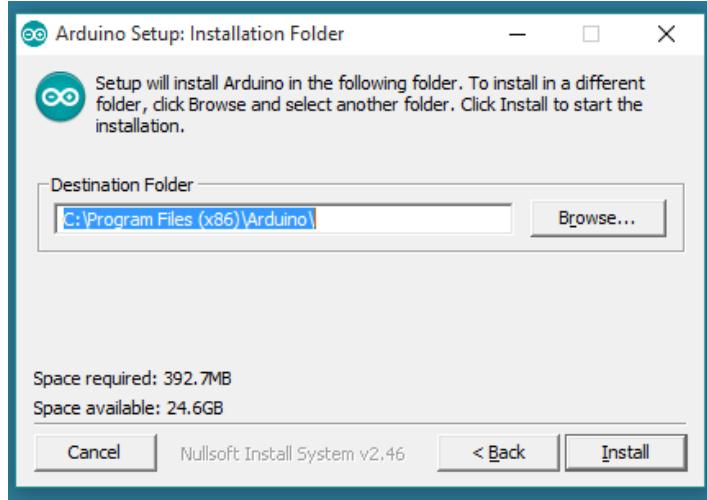
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

- Choose the components to install:

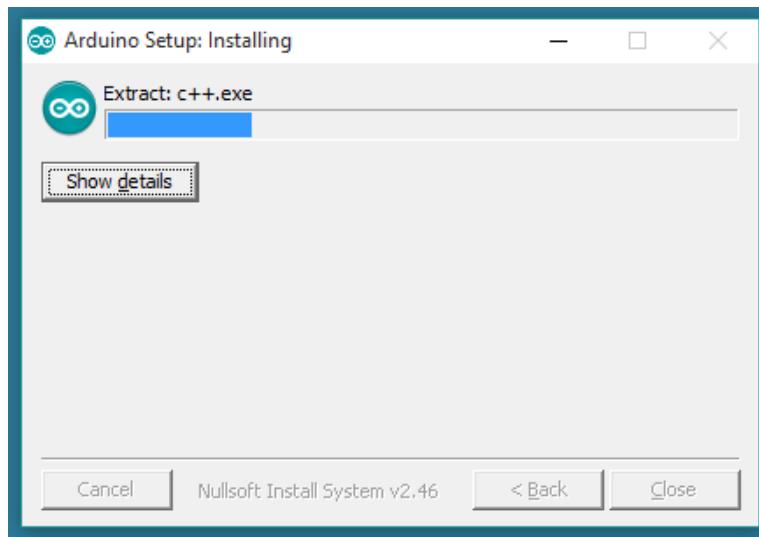


- Choose the installation directory (we suggest to keep the default one):

Smart Toll Collection on Highways

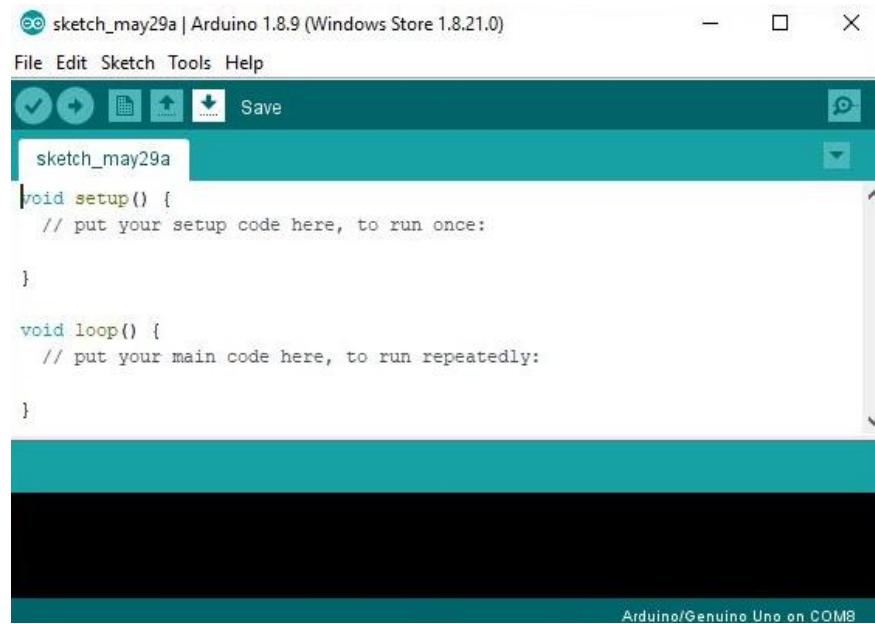


- The process will extract and install all the required files to execute properly the Arduino Software (IDE):



- When the Arduino Software (IDE) is properly installed you can go back to the Getting Started Home and choose your board from the list on the right of the page

Smart Toll Collection on Highways



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_may29a | Arduino 1.8.9 (Windows Store 1.8.21.0)". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Save, Run, and Upload. The main code editor window contains the following code:

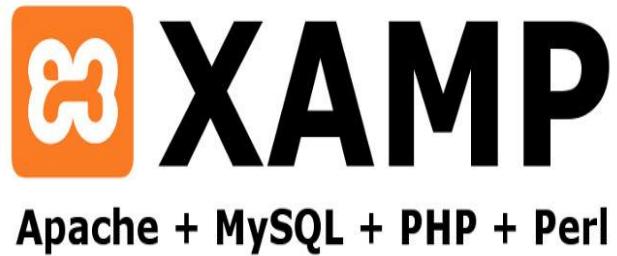
```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

In the bottom right corner of the code editor, it says "Arduino/Genuino Uno on COM8".

Chapter three

Web Server (XAMPP)

A Web server is a server that is responsible for accepting HTTP requests from web clients and serving them HTTP responses, usually in the form of web pages containing static (text, images etc) and dynamic (scripts) content.



➤ 3.1. Server:

- In a technical sense, a server is an instance of a computer program or device that accepts and responds to requests made by another program, known as a client. A good metaphor would be a customer (client) ordering a package then the mailman (server) delivering it to them or someone else.
- Servers are used to manage network resources. For example, a user may set up a server to control access to a network, send/receive e-mail, manage print jobs, or host a website. They are also proficient at performing intense calculations. Some servers are committed to a specific task, often referred to as dedicated. However, many servers today are shared servers which can take on the responsibility of e-mail, DNS, FTP, and even multiple websites in the case of a web server.
- Most servers are never turned off. Because they are commonly used to deliver services that are constantly required. Consequently, when servers fail, they can cause the network users and company many problems. To alleviate these issues, servers are commonly set up to be fault tolerant.

➤ **3.2. Web server:**

A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well. The process is an example of the client/server model. All computers that host Web sites must have Web server programs.

Web servers often come as part of a larger package of Internet- and intranet-related programs for serving email, downloading requests for File Transfer Protocol (FTP) files, and building and publishing Web pages. Considerations in choosing a Web server include how well it works with the operating system and other servers, its ability to handle server-side programming, security characteristics, and the particular publishing, search engine and site building tools that come with it.

➤ **3.3. Web Server Function:**

Web servers respond to client requests in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database.
- When a client sends a request for a web page, the web server searches for the requested page. If the requested page is found, then it will send it to the client with an HTTP response.
- If the requested web page is not found, the web server will send an HTTP response: Error 404 Not Found.
- If a client has requested for some other resources, then the web server will contact the application server and data store to construct the HTTP response.

➤ **3.4. Examples of Web Servers:**

1) Apache HTTP Server

This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.

2) Internet Information Services IIS

The Internet Information Server IIS is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms and maybe on upcoming new Windows versions also. IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.

3) Lighttpd

The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.

4) Sun Java System Web Server

This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.

5) Jigsaw Server

Jigsaw W3C's Server comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

➤ 3.5. Apache Web Server:

The Apache Web server has been the most popular and widely used Web server for the last decade. It is used by approximately 50% of all websites. Apache is cross-platform, lightweight, robust, and used in small companies as well as large corporations. Apache is also free and open-source. The Apache Web server has almost endless possibilities, due to its great modularity, which allows it to be integrated with numerous other applications. One of the most popular bundles is the XAMPP Web server application stack, which includes the Apache Web server alongside MySQL, PHP, Perl, and Python.

➤ 3.6. XAMPP

XAMPP is the title used for a **compilation of free software**. The name is an acronym, with each letter representing one of the five key components. The software packet contains the web server **Apache**, the relational database management system **MySQL** (or **MariaDB**), and the scripting languages **Perl** and **PHP**. The initial **X** stands for the operating systems that it works with: Linux, Windows, and Mac OS X.

- **Apache:** the open source web server Apache is the most widely used server worldwide for delivery of web content. The server application is made available as a free software by the Apache Software Foundation.
- **MySQL/MariaDB:** in MySQL, XAMPP contains one of the most popular relational database management systems in the world. In combination with the web server Apache and the scripting language PHP, MySQL offers data storage for web services. Current XAMPP versions have replaced MySQL with MariaDB (a community-developed fork of the MySQL project, made by the original developers).
- **PHP:** the server-side programming language PHP enables users to create dynamic websites or applications. PHP can be installed on all platforms and supports a number of diverse database systems.
- **Perl:** the scripting language Perl is used in system administration, web development, and network programming. Like PHP, Perl also enables users to programme dynamic web applications.

Alongside these core components, this free-to-use Apache distribution contains some other useful tools, which vary depending on your operating system. These tools include the mail server **Mercury**, the database administration tool **phpMyAdmin**, the web analytics software solutions **WebaLizer**, **OpenSSL**, and **Apache Tomcat**, and the FTP servers **FileZilla** or **ProFTPD**.

➤ 3.6.1. Application areas of XAMPP:

An XAMPP server can be installed and used with a single executable file quickly and easily, functioning as a local test system for Linux, Windows, and Mac OS X. The software packet contains the same components that are found on common web servers. Developers have the chance to test out their projects locally and to transfer them easily to productive systems. But XAMPP isn't suitable to use as a public server, because many safety features have been deliberately left out to simplify and speed up the system for testing.

➤ 3.6.2. Installing XAMPP:

1) Download

XAMPP is a release made available by the non-profit project Apache Friends. Versions with PHP 5.5, 5.6, or 7 are available for download on the [Apache Friends](#) website.

2) Run .exe file

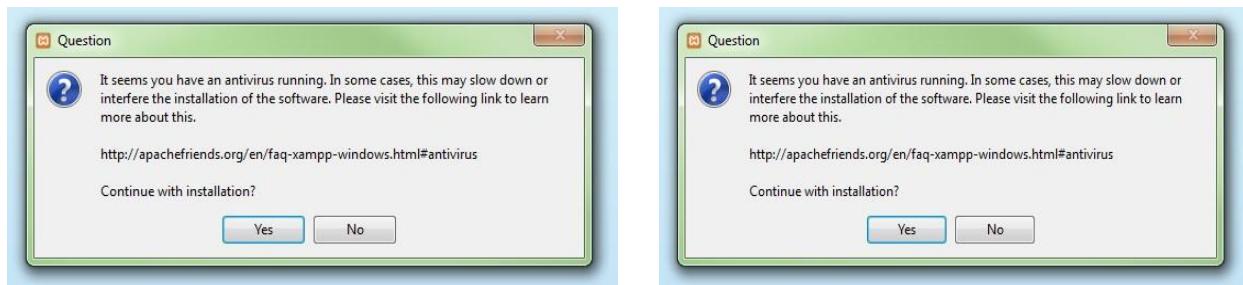
Once the software bundle has been downloaded, you can start the installation by double clicking on the file with the ending .exe.

3) Deactivate any antivirus software

Since an active antivirus programme can negatively affect the installation process, it's recommended to temporarily pause any antivirus software until all XAMPP components have successfully been installed.

4) Deactivate UAC

User Account Control (UAC) can interfere with the XAMPP installation because it limits writing access to the C: drive, so we recommend you deactivate this too for the duration of the installation process. To find out how to turn off your UAC, head to the [Microsoft Windows support pages](#).



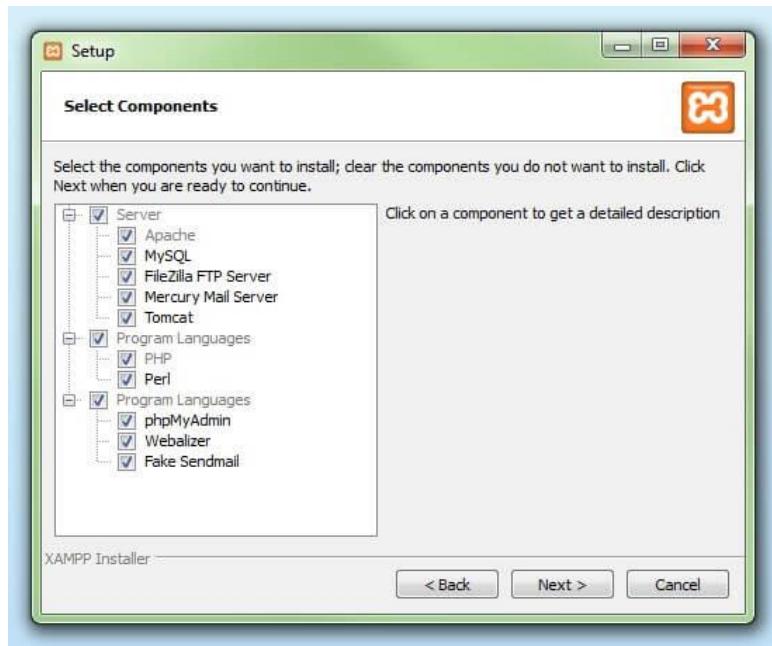
5) Start the setup wizard

After you've opened the .exe file (after deactivating your antivirus programme(s) and taken note of the User Account Control, the start screen of the XAMPP setup wizard should appear automatically. Click on 'Next' to configure the installation settings.



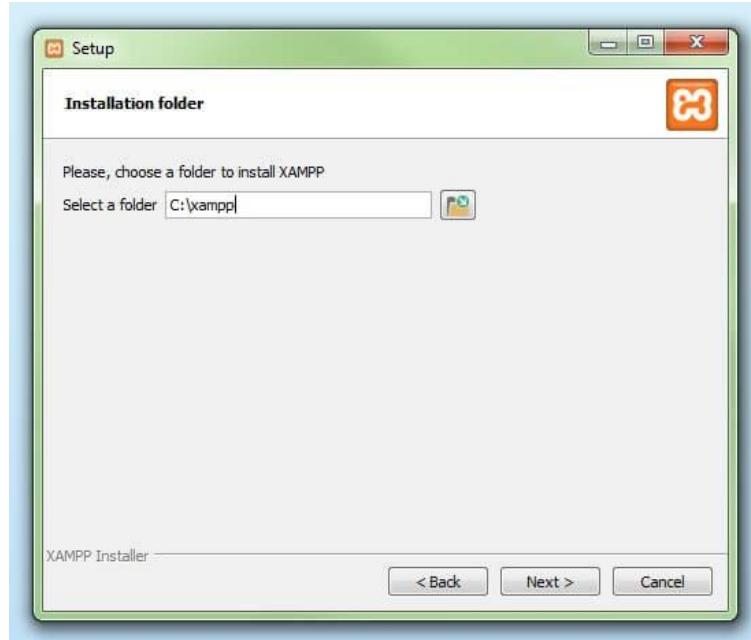
6) Choose software components

Under 'Select Components', you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click 'Next'.



7) Choose the installation directory

In this next step, you have the chance to choose where you'd like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name XAMPP will be created under C:\ for you. After you've chosen a location, click 'Next'.



8) Start the installation process

Once all the aforementioned preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. This process can take several minutes in total.

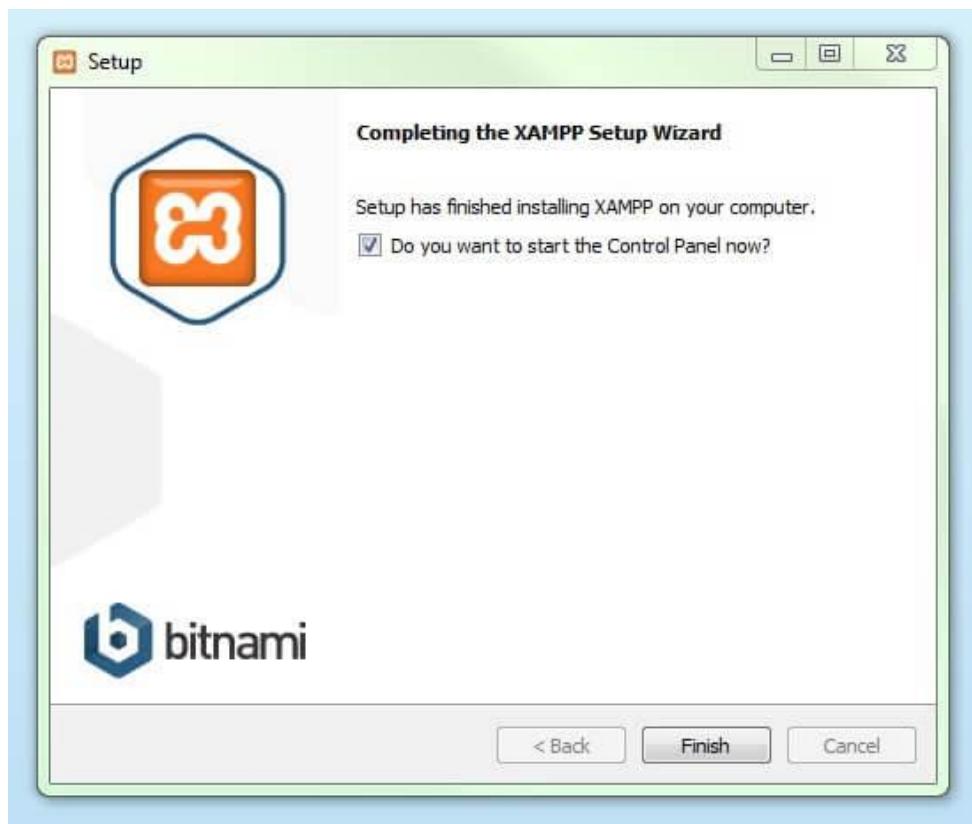


9) Windows Firewall blocking

Your Firewall may interrupt the installation process to block the some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network. Remember that making your XAMPP server available for public networks isn't recommended.

10) Complete installation

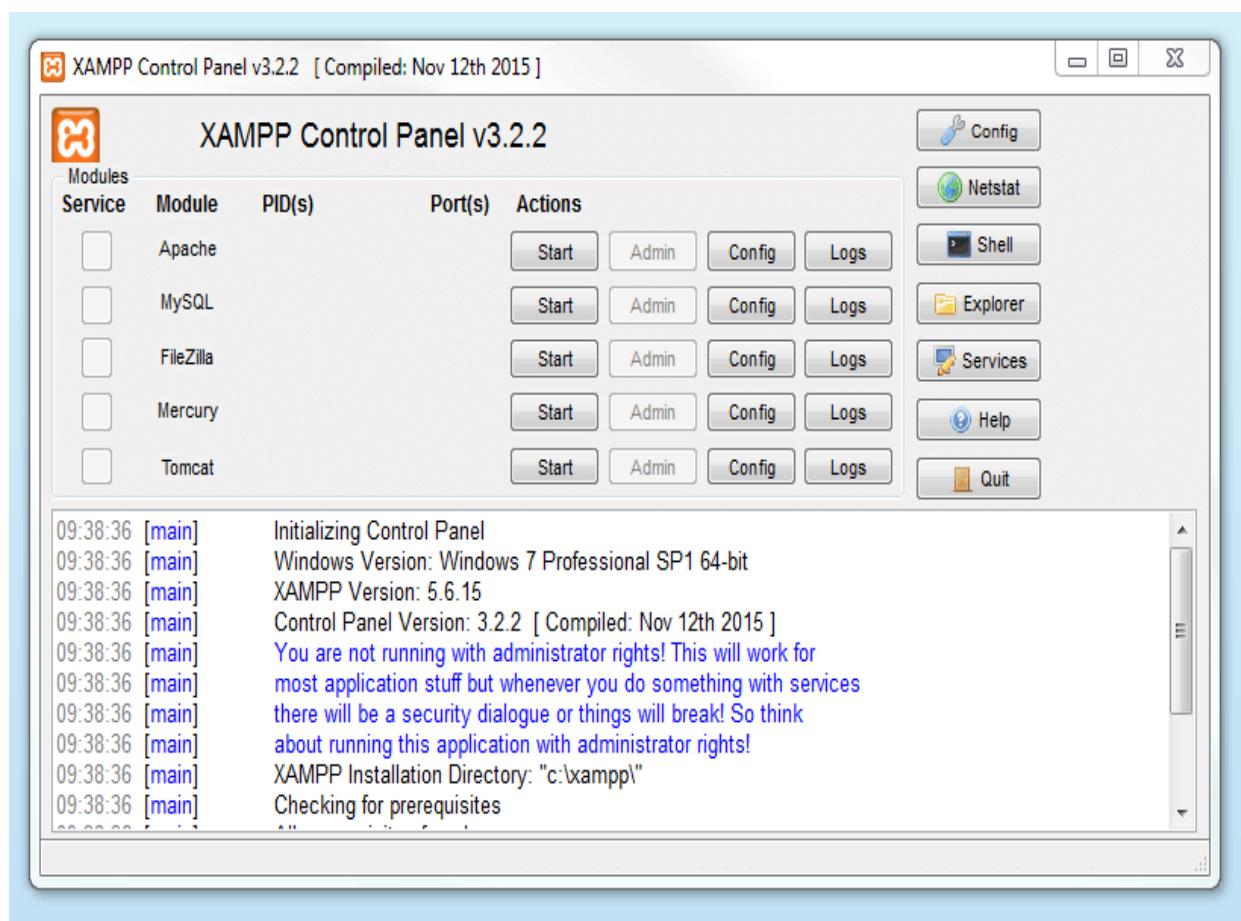
Once all the components are unpacked and installed, you can close the setup wizard by clicking on 'Finish'. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.



➤ 3.6.3. The XAMPP Control Panel:

Controls for the individual components of the test server can be reached through the XAMPP Control Panel. The clear user interface logs all actions and allows you to start or stop individual modules with a single. The XAMPP Control Panel also offers you various other buttons, including:

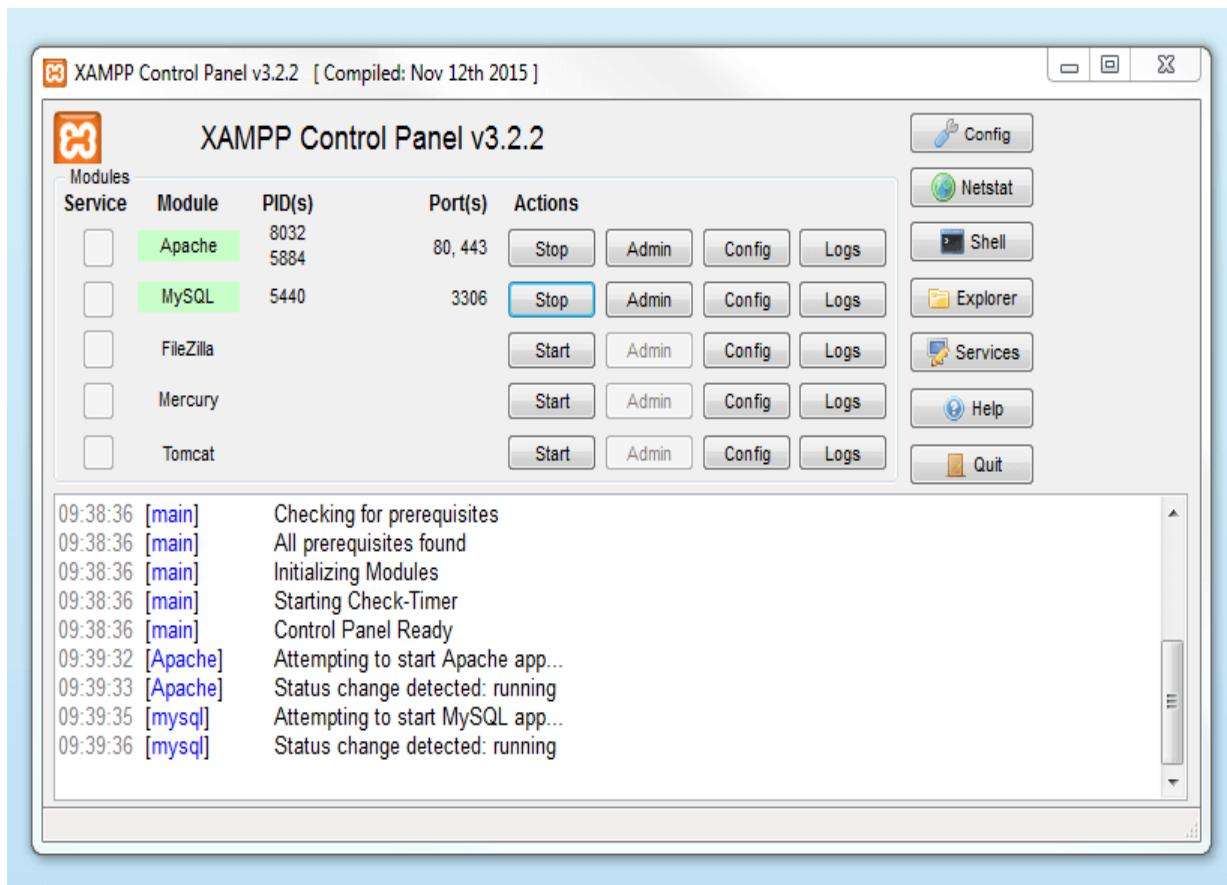
- Config: allows you to configure the XAMPP as well as the individual components
- Netstat: shows all running processes on the local computer
- Shell: opens a UNIX shell
- Explorer: opens the XAMPP folder in Windows Explorer
- Services: shows all services currently running in the background
- Help: offers links to user forums
- Quit: closes the XAMPP Control Panel



➤ 3.6.4. Starting modules:

Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under ‘Actions’. You can see which modules have been started because their names are highlighted green under the ‘Module’ title.

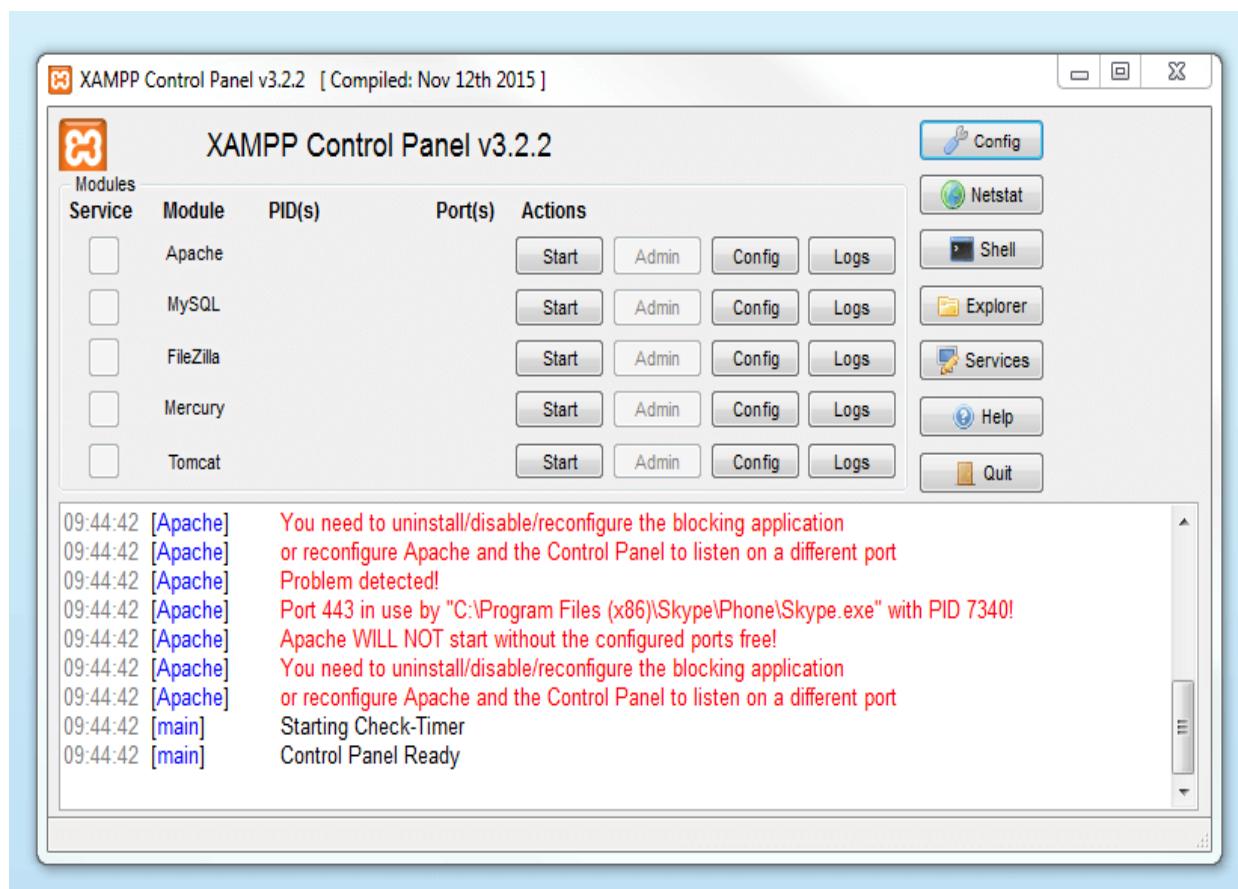
If a module can't be started as a result of an error, you'll be informed of this straight away in red font. A detailed error report can help you identify the cause of the issue.



➤ 3.6.5. Setting up XAMPP:

A common source of error connected with Apache is blocked ports. If you're using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programmes. In the example above, it's likely that the Tomcat port is being blocked, meaning the web server can't be started. There are three ways to solve this issue:

- Change the conflicting port: Let's assume for the sake of example that the instant messenger programme Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype's port settings. To do this, open the programme and navigate via 'Actions', 'Options', and 'Advanced', until you reach the 'Connections' menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.
- Change the XAMPP module port settings: Click the Config button for the module in question and open the files *httpd.conf* and *httpd-ssl.conf*. Replace port number 80 in *httpd.conf* and port number 443 in *httpd-ssl.conf* with any free ports, before saving the file data. Now click on the general Config button on the right-hand side and select 'Services and Ports Settings'. Customise the ports for the module server to reflect the changes in the *conf* files.
- End the conflicting programme: The simplest way to avoid port conflicts in the short term is to end the conflicting programme (Skype in this case). If you restart Skype after your XAMPP module servers are already running, it will select a different port and your issue will be resolved.



➤ 3.6.6. Module administration:

- Click on the Admin button of your Apache server to go to the web address of your web server. The Control Panel will now start in your standard browser, and you'll be led to the dashboard of your XAMPP's local host. The dashboard features numerous links to websites for useful information as well as the open source project [BitNami](#), which offers you many different applications for your XAMPP, like WordPress or other content management systems. Alternatively, you can reach the dashboard through `localhost/dashboard/`.



Welcome to XAMPP for Windows 5.6.15

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We have set up a site, translate.apachefriends.org, where users can contribute translations.

Install applications on XAMPP using Bitnami

Apache Friends and Bitnami are cooperating to make dozens of open source applications available on XAMPP, for free. Bitnami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the Bitnami XAMPP page for details on the currently available apps.



Blog

Privacy Policy

CDN provided by **fastly**

Copyright (c) 2015, Apache Friends

- You can use the Admin button of your database module to open phpMyAdmin. Here, you can manage the databases of your web projects that you're testing on your XAMPP. Alternatively, you can reach the administration section of your MySQL database via localhost/phpmyadmin/.

The screenshot shows the phpMyAdmin interface with the following details:

- General settings:** Server connection collation is set to utf8mb4_unicode_ci.
- Appearance settings:** Language is English (United Kingdom), Theme is pmahomme, and Font size is 82%.
- Database server:** Server: 127.0.0.1 via TCP/IP, Server type: MariaDB, Server version: 10.1.10-MariaDB - mariadb.org binary distribution, Protocol version: 10, User: root@localhost, Server charset: UTF-8 Unicode (utf8).
- Web server:** Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.15, Database client version: libmysql - mysqlnd 5.0.11-dev-20120503 - \$Id: 3c688b6bbc30d3ac34fdd4b7b5b787fe5555 \$, PHP extension: mysqli, PHP version: 5.6.15.
- phpMyAdmin:** Version information: 4.5.1, latest stable version: 4.7.0, Documentation, Wiki, Official Homepage, Contribute, Get support, List of changes.

➤ 3.6.7. Testing XAMPP installation:

To check whether the test server is installed and configured correctly, you have the option to create a PHP test page, store them on your XAMPP's local host, and retrieve them via the web browser.

- Open the XAMPP directory through the 'Explorer' button in the Control Panel and choose the folder *htdocs* (C:\xampp\htdocs for standard installations). This directory will store file data collected for web pages that you test on your XAMPP server. The *htdocs* folder should already contain data to help configuration of the web server. But you should store your own projects in a new folder (like 'Test Folder' for example).

- You can create a new PHP page easily by using the following content in your editor and storing it as *test.php* in your ‘*test*’ folder (C:\xampp\htdocs\test):

```
<html>
<head>
<title>PHP-Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

- The last step now is to open your web browser and load your PHP page via *localhost/test/test.php*. If your browser window displays the words ‘Hello World’, then you’ve successfully installed and configured your XAMPP.

Chapter four

Database (XAMPP)

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

Information and data are different. Information is understood by a person. Data are values stored on a passive medium like a computer disk. The purpose of a database management system (DBMS) is to bridge the gap between information and data - the data stored in memory or on disk must be converted to usable information.



➤ 4.1. The basic processes that are supported by a DBMS are:

- **Data definition** – Creation, modification and removal of definitions that define the organization of the data.
- **Update** – Insertion, modification, and deletion of the actual data.
- **Retrieval** – Providing information in a form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
- **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.

A database is a model of a real world system. The contents (sometimes called the extension) of a database represent the state of what is being modeled. Changes in the database represent events occurring in the environment that change the state of what is being modeled. It is appropriate to structure a database to mirror what it is intended to model.

Databases can be analyzed at different levels:

- Conceptual Modeling
- Logical Modeling
- Physical Modeling

➤ **4.2. Conceptual Modeling:**

Conceptual-level concepts permit us to model the application world in terms that are independent of any particular data (logical) model. Conceptual models provide a framework for developing a database schema from the top to the bottom in the process of a database design. This Section examines the entity-relationship model and the object-oriented model as representatives of conceptual modeling. The entity-relationship model is widely used and the object-oriented model is gaining more acceptance for non-traditional databases.

➤ **4.3. The Entity-Relationship Model:**

It is a tool for analyzing the semantic features of an application that are independent of events. This approach includes a graphical notation, which depicts entity classes as rectangles, relationships as diamonds, and attributes as circles or ovals. For complex situation, a partial entity-relationship diagram may be used to present a summary of the entities and relationships that do not include the details of the attributes. The entity-relationship diagram provides a convenient method for visualizing the interrelationships among entities in a given application.

This tool has proven to be useful in making the transition from an information application description to a formal database schema. The entity-relationship model is used for describing the conceptual schema of an enterprise without attention to the efficiency of the physical database design. The entity-relationship diagrams are then turned into a logical schema in which the database is actually implemented.

➤ **4.4. Short definitions of some of the basic terms that are used for describing important entity-relationship concepts are:**

- **Entity:** An entity is a thing that exists and is distinguishable.
 - (a) **Entity instance:** An instance is a particular occurrence of an entity. For example, each person is an instance of an entity Person, each car is an instance of an entity Car, etc.
 - (b) **Entity class:** A group of similar entities is called an entity class or entity type. An entity class has common attributes.
- **Attributes:** Attributes describe properties of entities and relationships.
 - (a) **Simple and composite attributes:** A simple attribute is the smallest semantic unit of data, which are atomic (no internal structure). A composite attribute can be subdivided into parts, e.g., address (street, city, state, zip).
 - (b) **Single and multivalued attributes:** Single attributes have a single value for a particular entity. Multivalued attributes have multiple values of an attribute for a particular entity; e.g., degrees or courses that a student can have or take.
- **Relationships:** A relationship is a connection between entities. For example, a relationship between PERSONS and AUTOMOBILES could be an "OWNS" relationship. That is to say, automobiles are owned by people.
 - The mapping cardinality of a relationship indicates the number of instances in entity E1 that can or must be associated with instances in entity E2:
 - (a) **One-One Relationship:** For each entity instance in one entity there is at most one associated entity instance in the other entity.
 - (b) **Many-One Relationships:** One entity instance in entity E2 is associated with zero or more entity instances in entity E1, but each entity instance in E1 is associated with at most one entity instance in E2.
 - (c) **Many-Many Relationships:** There are no restrictions on how many entity instances in either entity are associated with a single entity instance in the other.
- **Keys:** The key uniquely differentiates one entity instance from all others in the entity. A key is an identifier.
 - (a) **Primary Key:** Identifier used to uniquely identify one particular instance of an entity. A primary key can be one or more attributes. It must be unique within the domain. Its value should not change over time. It must always have a value.

(b) Concatenated Key: Key made up of parts which when combined become a unique identifier. Multiple attribute keys are concatenated keys.

(c) Foreign Keys: Foreign keys reference a related table through the primary key of that related table.

➤ 4.5. Entity-Relation Diagram

➤ 4.5.1. Symbols used in entity-relationship diagrams include:

- **Rectangles:** represent ENTITY or ENTITY CLASSES
- **Circles:** represent ATTRIBUTES
- **Diamonds:** represent RELATIONSHIPS
- **Arcs:** connect entities to relationships. Arcs are also used to connect attributes to entities. Some styles of entity-relationship diagrams use arrows and double arrows to indicate the one and the many in relationships. Some use forks etc.
- **Underlined attributes:** identify keys of entities.

➤ 4.6. Database languages

Database languages are special-purpose languages, which allow one or more of the following tasks, sometimes distinguished as sublanguages:

- **Data control language (DCL)** – controls access to data;
- **Data definition language (DDL)** – defines data types such as creating, altering, or dropping and the relationships among them;
- **Data manipulation language (DML)** – performs tasks such as inserting, updating, or deleting data occurrences;
- **Data query language (DQL)** – allows searching for information and computing derived information.

Database languages are specific to a particular data model. Notable examples include:

- **SQL:** combines the roles of data definition, data manipulation, and query in a single language. It was one of the first commercial languages for the relational model, although it departs in some respects from the relational model. SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. The standards have been regularly enhanced since and is supported (with varying degrees of conformance) by all mainstream commercial relational DBMSs.

- **OQL:** is an object model language standard (from the Object Data Management Group). It has influenced the design of some of the newer query languages like JDOQL and EJB QL.
- **XQuery:** is a standard XML query language implemented by XML database systems such as MarkLogic and eXist, by relational databases with XML capability such as Oracle and DB2, and also by in-memory XML processors such as Saxon.
- **SQL/XML:** combines XQuery with SQL.

➤ 4.7. SQL

Structured Query language (SQL) pronounced as "S-Q-L" or sometimes as "See-Quel" is actually the standard language for dealing with Relational Databases.

SQL programming can be effectively used to insert, search, update, delete database records. That doesn't mean SQL cannot do things beyond that. In fact it can do lot of things including, but not limited to, optimizing and maintenance of databases. Relational databases like MySQL Database, Oracle, Ms SQL server, Sybase, etc uses SQL. SQL syntaxes used in these databases are almost similar, except the fact that some are using few different syntaxes and even proprietary SQL syntaxes.

➤ 4.7.1. SQL Command

SQL defines following ways to manipulate data stored in an RDBMS.

▪ **DDL: Data Definition Language**

This includes changes to the structure of the table like creation of table, altering table, deleting a table etc.

All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Command	Description
Create	to create new table or database
Alter	for alteration
Truncate	delete data from table
Drop	to drop a table
Rename	to rename a table

▪ **DML: Data Manipulation Language**

DML commands are used for manipulating the data stored in the table and not the table itself.

DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

Command	Description
Insert	to insert a new row
Update	to update existing row
Delete	to delete a row
Merge	merging two rows or two tables

▪ **TCL: Transaction Control Language**

These commands are to keep a check on other commands and their affect on the database. These commands can annul changes made by other commands by rolling the data back to its original state. It can also make any temporary change permanent.

Command	Description
Commit	to permanently save
Rollback	to undo change
Savepoint	to save temporarily

- **DCL: Data Control Language**

Data control language are the commands to grant and take back authority from any database user.

Command	Description
Grant	grant permission of right
Revoke	take back permission.

- **DQL: Data Query Language**

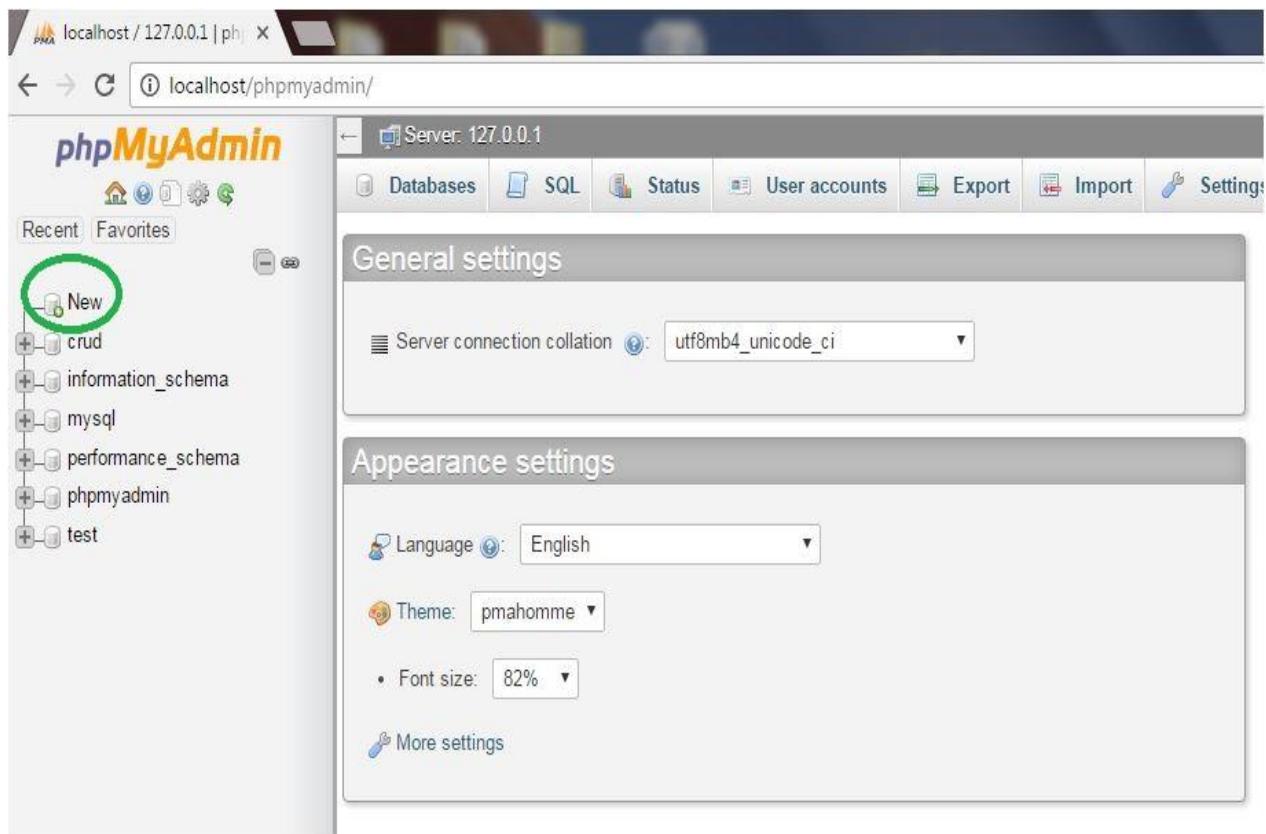
Data query language is used to fetch data from tables based on conditions that we can easily apply.

Command	Description
Select	retrieve records from one or more table

➤ 4.8. Creating Database Using XAMPP

▪ Step 1

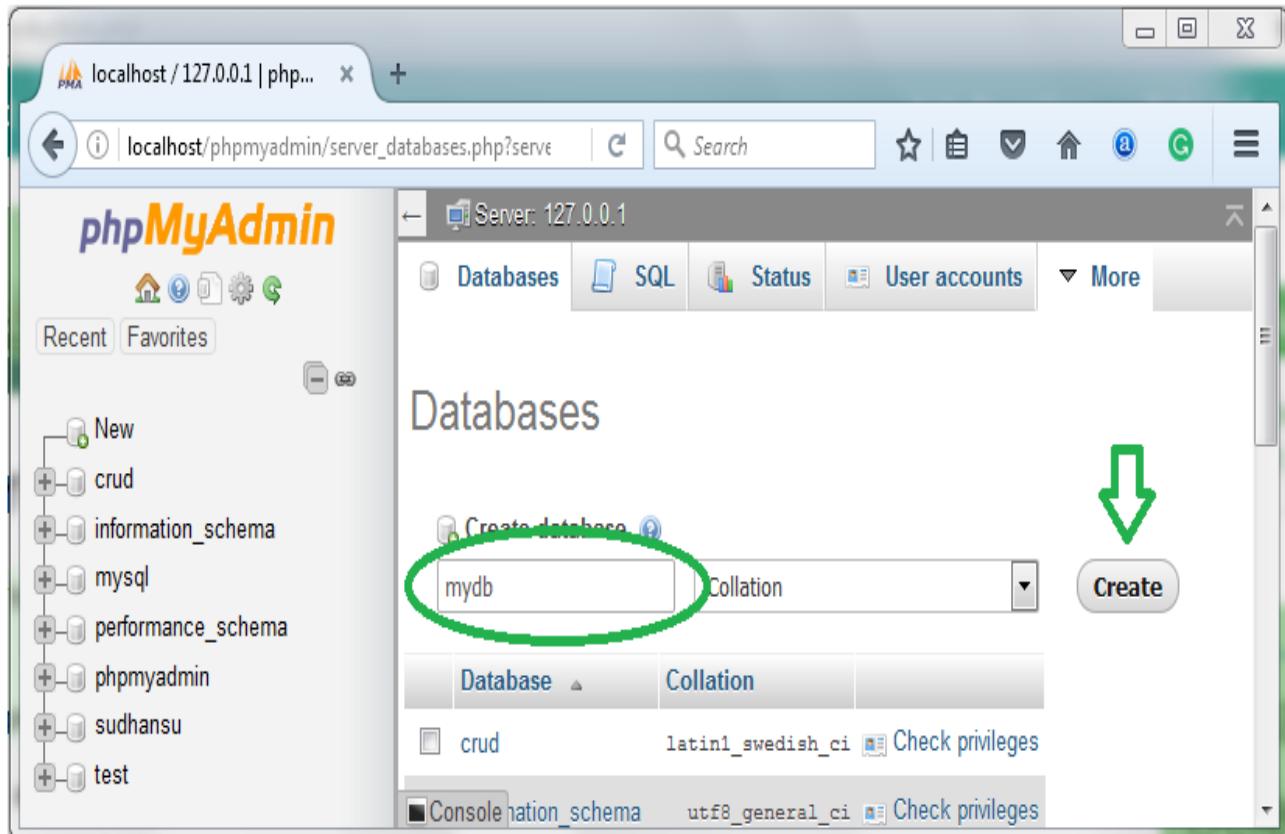
First of all start your XAMPP server. Then go to the link bar of your browser and type **localhost/phpmyadmin** and click enter. Then you find a page like this



Then click on the new button that is available on the sidebar of that page.

▪ Step 2

After click on the new button you find a page like that.



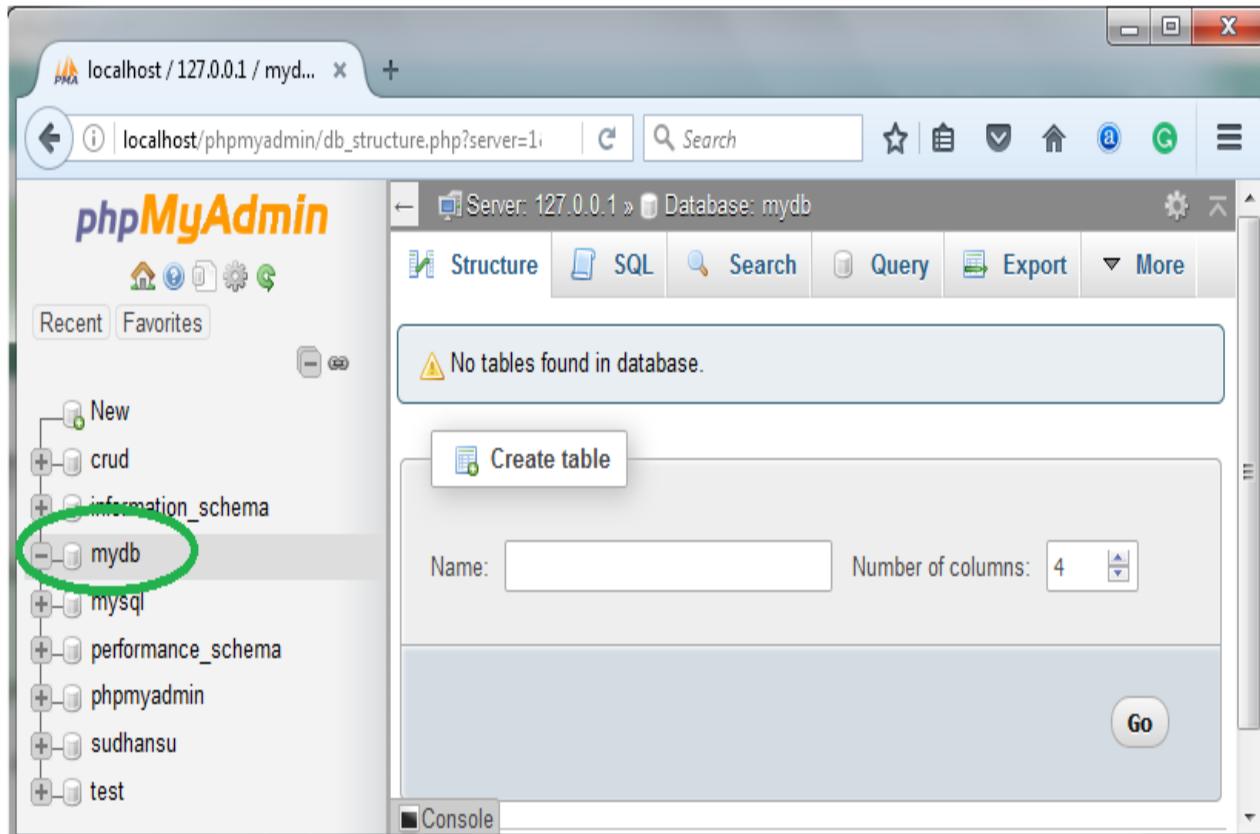
Here put a name of your own choice on the create database field and click on the **create** button and you get a successful message on the screen that you create database successfully and will appear in the database list.

Now let's **create a table in PhpMyAdmin** under the database that we have created currently.

Note: You can't create table if you not have a database so first create database.

▪ Step 3

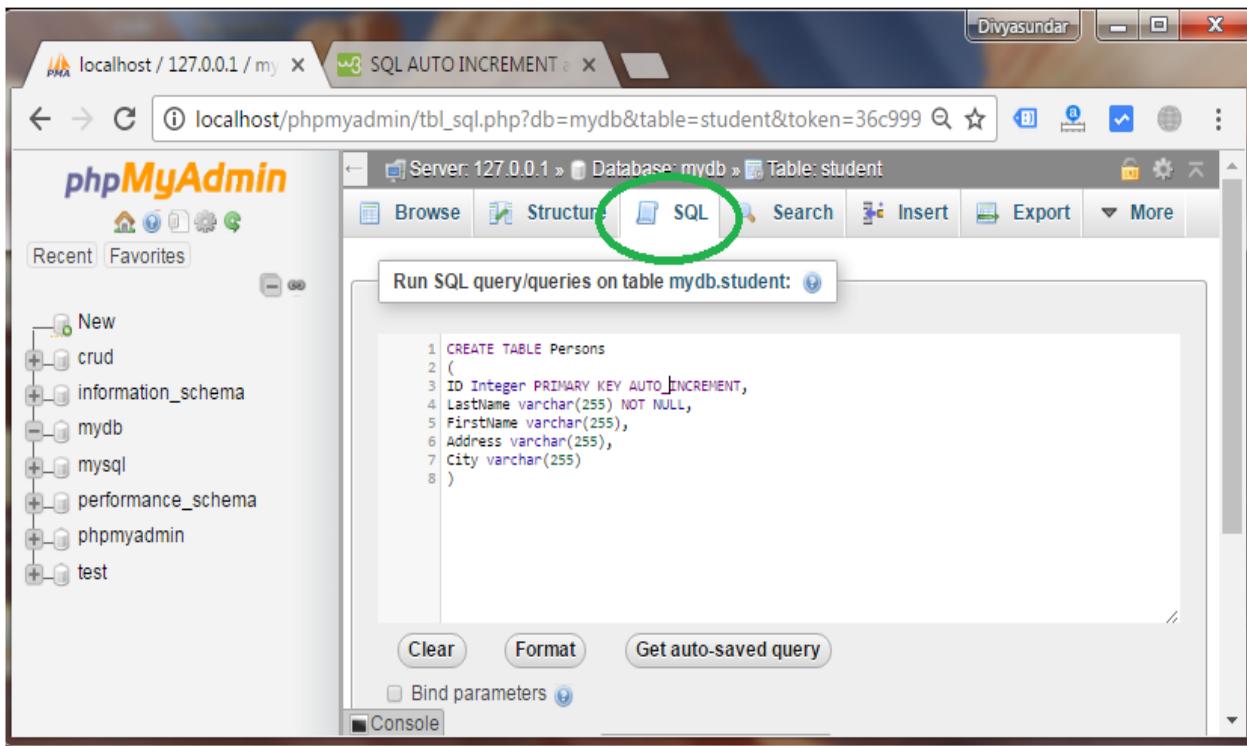
Click on the database name in which under you create a table. After click on the database name you will find a page like that.



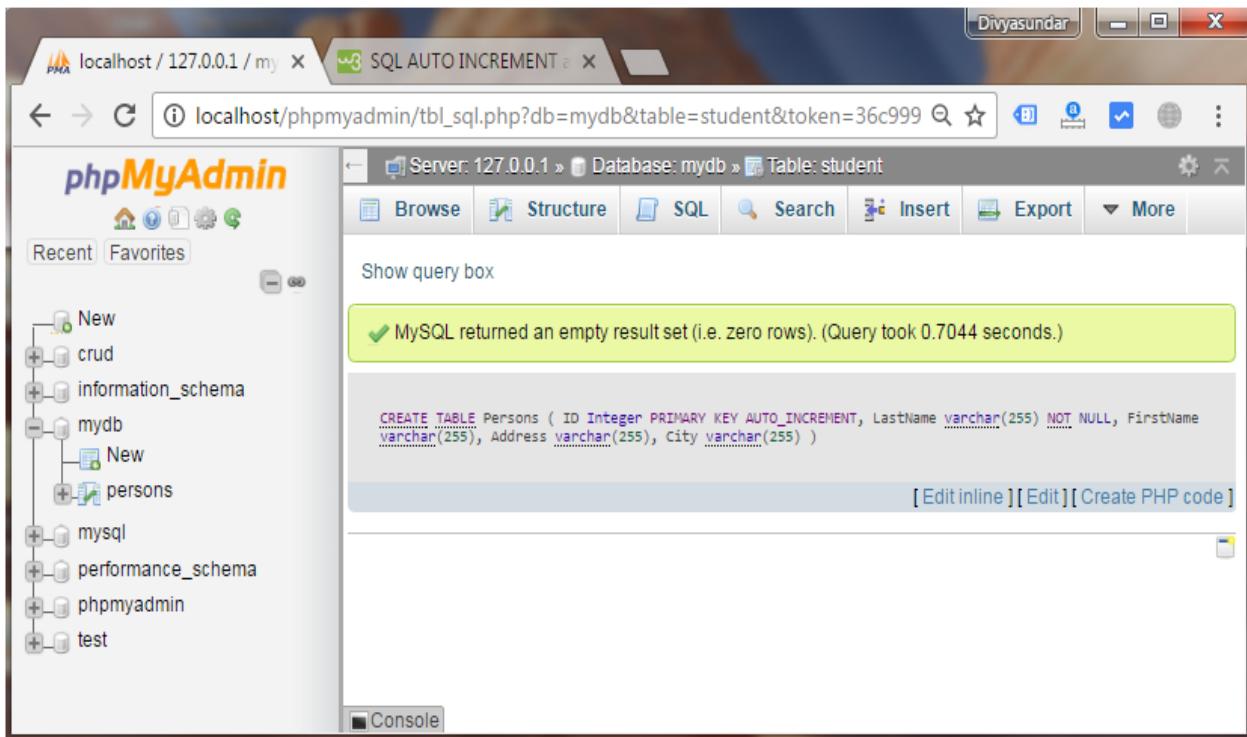
Here you have two options to create a table, the first one is using **structure** and the second one is using **SQL**.

- If you want to create table in structure option then put your table name on the create table name field and choose the number of columns and click on the go button.
- If you want to create a table by writing SQL Query simply click on the SQL button on the page and write your query and click on the go button.

Smart Toll Collection on Highways



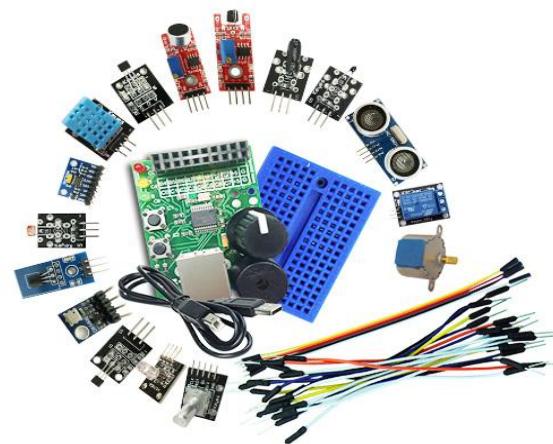
- If your SQL query is correct you find a page like that. Now you can add tables and edit them and insert data using SQL queries.



Chapter five

Sensors

A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.



Sensors are sophisticated devices that are frequently used to detect and respond to electrical or optical signals. It converts the physical parameter into a signal which can be measured electrically. Let's explain the example of temperature. The mercury in the glass thermometer expands and contracts the liquid to convert the measured temperature which can be read by a viewer on the calibrated glass tube.

Sensors are pervasive. They are embedded in our bodies, automobiles, airplanes, cellular telephones, radios, chemical plants, industrial plants and countless other applications. Active element of a sensor is called a transducer.

A transducer is a device which converts one form of energy to another. When the input is a physical quantity and the output is electrical then it's a sensor. When the input is electrical and the output a physical quantity then it's an actuator.

➤ 5.1. Factors of choosing a sensor:

There are certain factors which have to be considered when we choose a sensor like:

➤ Environmental Factors:

- Temperature range
- Humidity effects
- Size
- Power consumption
- Ruggedness
- Corrosion

➤ Economic Factors:

- Cost
- Availability
- Lifetime

➤ Sensor Characteristics:

- Sensitivity and Accuracy
- Range
- Stability
- Response time
- Frequency response
- Error
- Repeatability

➤ 5.2. Classification of Sensors

The sensors are classified into the following criteria:

- Primary Input quantity (Measurand)
- Transduction principles (Using physical and chemical effects)
- Material and Technology
- Property
- Application

➤ **Transduction principle** is the fundamental criteria which are followed for an efficient approach. Usually, material and technology criteria are chosen by the development engineering group.

➤ **5.2.1. Classification based on property is as given below:**

- **Temperature** - Thermistors, thermocouples, RTD's, IC and many more.
- **Pressure** - Fibre optic, vacuum, elastic liquid based manometers, LVDT, electronic.
- **Flow** - Electromagnetic, differential pressure, positional displacement, thermal mass.
- **Level Sensors** - Differential pressure, ultrasonic radio frequency, radar, thermal displacement.
- **Proximity and displacement** - LVDT, photoelectric, capacitive, magnetic, ultrasonic.
- **Biosensors** - Resonant mirror, electrochemical, surface Plasmon resonance, Light addressable potentio-metric.
- **Image** - Charge coupled devices, CMOS
- **Gas and chemical** - Semiconductor, Infrared, Conductance, Electrochemical.
- **Acceleration** - Gyroscopes, Accelerometers.
- **Others** - Moisture, humidity sensor, Speed sensor, mass, Tilt sensor, force, viscosity.

➤ **We will be using IR sensors for our project prototype.**

➤ **5.3. IR Sensor:**



This device emits and/or detects infrared radiation to sense a particular phase in the environment. Generally, thermal radiation is emitted by all the objects in the infrared spectrum. The infrared sensor detects this type of radiation which is not visible to human eye.

➤ **5.3.1. Advantages:**

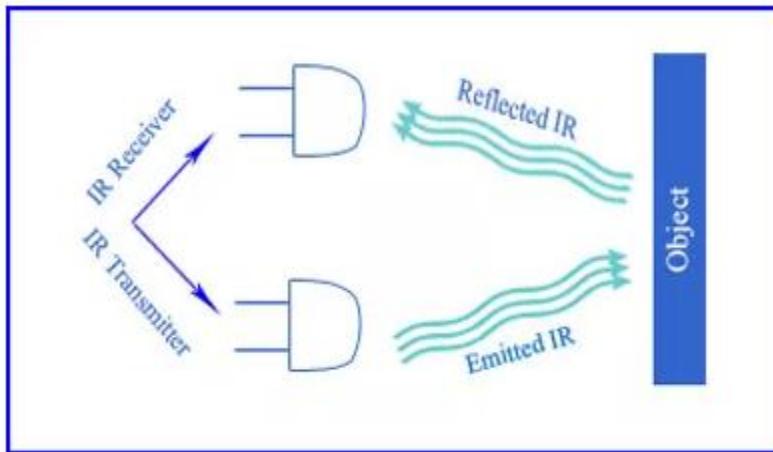
- Easy for interfacing
- Readily available in market

➤ **5.3.2. Disadvantages:**

- Disturbed by noises in the surrounding such as radiations, ambient light etc.

➤ **5.3.3. How IR sensors work:**

The basic idea is to make use of IR LEDs to send the infrared waves to the object. Another IR diode of the same type is to be used to detect the reflected wave from the object. The diagram is shown below.



When IR receiver is subjected to infrared light, a voltage difference is produced across the leads. Less voltage which is produced can be hardly detected and hence operational amplifiers (Op-amps) are used to detect the low voltages accurately.

By measuring the distance of the object from the receiver sensor, the electrical property of IR sensor components can be used to measure the distance of an object. The fact when IR receiver is subjected to light, a potential difference is produced across the leads.

➤ 5.3.4. Applications:

- **Thermography** – According to the black body radiation law, it is possible to view the environment with or without visible illumination using thermography
- **Heating** – Infrared can be used to cook and heat food items. They can take away ice from the wings of an aircraft. They are popular in industrial field such as, print dying, forming plastics, and plastic welding.
- **Spectroscopy** – This technique is used to identify the molecules by analyzing the constituent bonds. This technique uses light radiation to study organic compounds.
- **Meteorology** – Cloud heights, calculate land and surface temperature is possible when weather satellites are equipped with scanning radiometers.
- **Photobiomodulation** – This is used for chemotherapy in cancer patients. This is used to treat anti herpes virus.
- **Climatology** – Monitoring the energy exchange between the atmosphere and earth.
- **Communications** – Infra red laser provide light for optical fibre communication. These radiations are also used for short range communications among mobiles and computer peripherals.

➤ 5.3.5. Design of IR Circuit :

IR sensor used to detect any movement made in front of the two sensors which is IR led and photodiode.

The range of IR sensor can be adjusted using variable resistor (preset). The range depends on Quality and specification of IR led and Photodiode used.

IR sensor can be used for making hobby projects like Line follower robot without using a microcontroller, obstacle detector robot or obstacle avoider robot.

IR sensor can also be used for home automation like automatic water dispenser, automatic sliding door, and automatic sliding door.

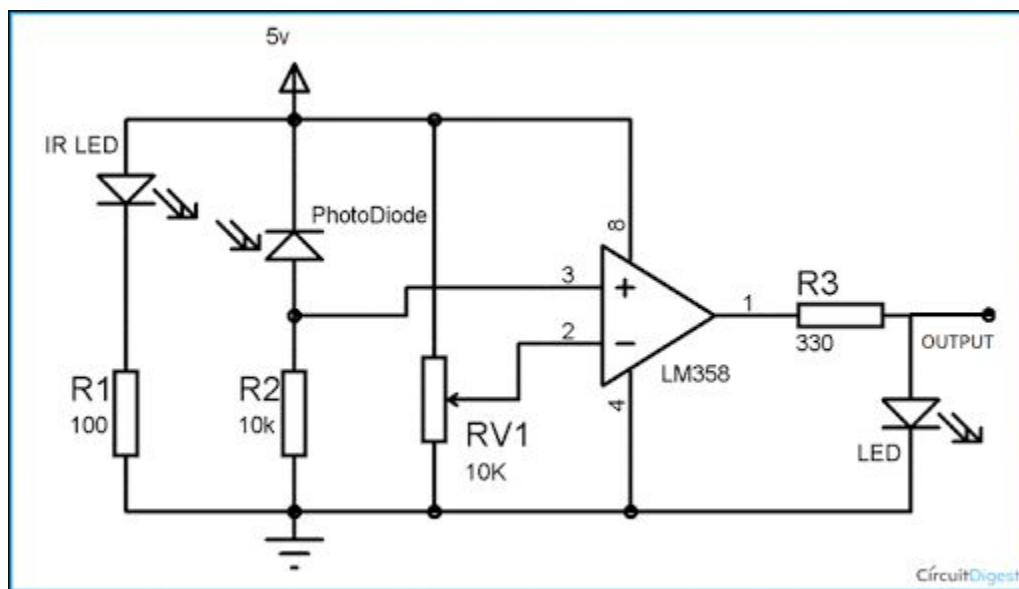
Since it has wide range of applications it can also be used for security purpose for detecting trespassing and you can create your own applications too.

➤ **Steps:**

• **Step 1:** Gather the Components and Tools

- 1 x 8 pin IC base
- 1 x LM358
- 1 x photodiode(5mm)
- 1 x IR led(5mm)
- 1 x 10k preset (variable resistor)
- 1 x 10k ohm resistor
- few wires and male header pins
- soldering iron and solder
- wire cutter

• **Step 2:** Placing the Components



• **Step 3:** Done! Just Anything as Output.

➤ 5.3.6. IR with Raspberry pi:

➤ Programming your Raspberry Pi:

Here we are using Python Programming language for programming Raspberry Pi. There are many ways to program your Raspberry Pi but we are using the **Python 3 IDE**, since it is the most used one. We will talk about few commands which we are going to use in Python program:

- We are going to import GPIO file from library, below function enables us to program GPIO pins of Raspberry Pi. We are also renaming “GPIO” to “IO”, so in the program whenever we want to refer to GPIO pins we will use the word ‘IO’.

```
import RPi.GPIO as IO
```

- Sometimes, when the GPIO pins, which we are trying to use, might be doing some other functions. In that case, we will receive warnings while executing the program. Below command tells the PI to ignore the warnings and proceed with the program.

```
IO.setwarnings(False)
```

- We can refer the GPIO pins of Raspberry Pi, either by pin number on board or by their function number. Like ‘PIN 29’ on the board is ‘GPIO5’. So we tell here either we are going to represent the pin here by ‘29’ or ‘5’.

```
IO.setmode (IO.BCM)
```

- We are setting 3 pins as input/output pins. The two output pins will control the LED and the input pin will read signal from the IR sensor.

```
IO.setup(2,IO.OUT) #GPIO 2 -> Red LED as output
```

```
IO.setup(3,IO.OUT) #GPIO 3 -> Green LED as output
```

```
IO.setup(14,IO.IN) #GPIO 14 -> IR sensor as input
```

- Now we have to turn off the Green LED and turn on the Red LED when the object is far. This can be done by checking the GPIO14 pin.

```
if(IO.input(14)==True): #object is far away  
  
    IO.output(2,True) #Red led ON  
  
    IO.output(3,False) # Green led OFF
```

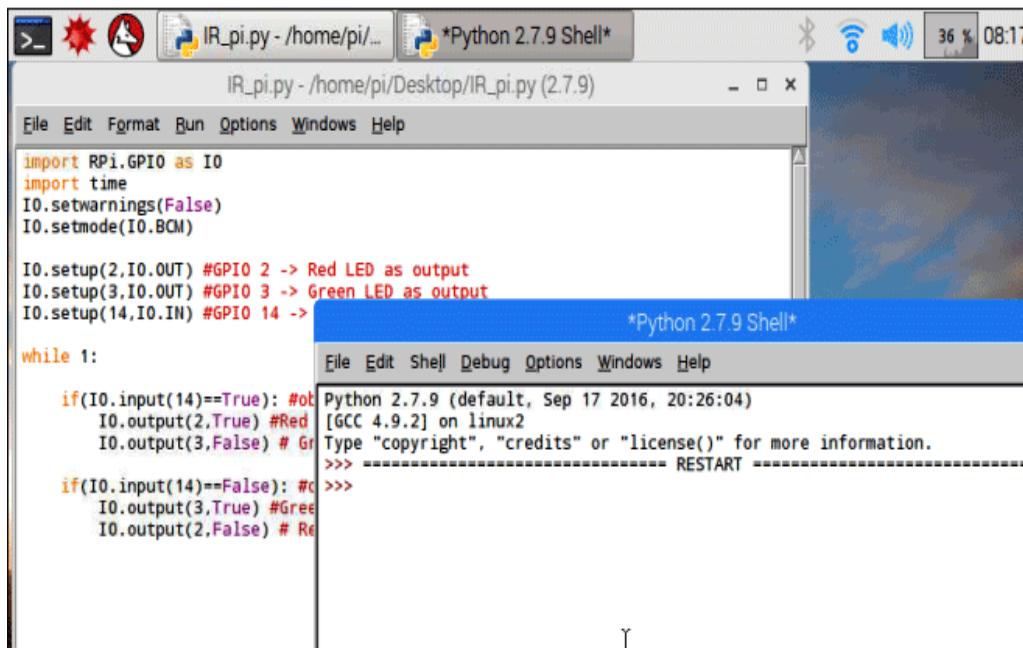
- Similarly we have to turn on the Green LED and turn off the Red LED when the object is near.

```
if(IO.input(14)==False): #object is near  
  
    IO.output(3,True) #Green led ON  
  
    IO.output(2,False) # Red led OFF
```

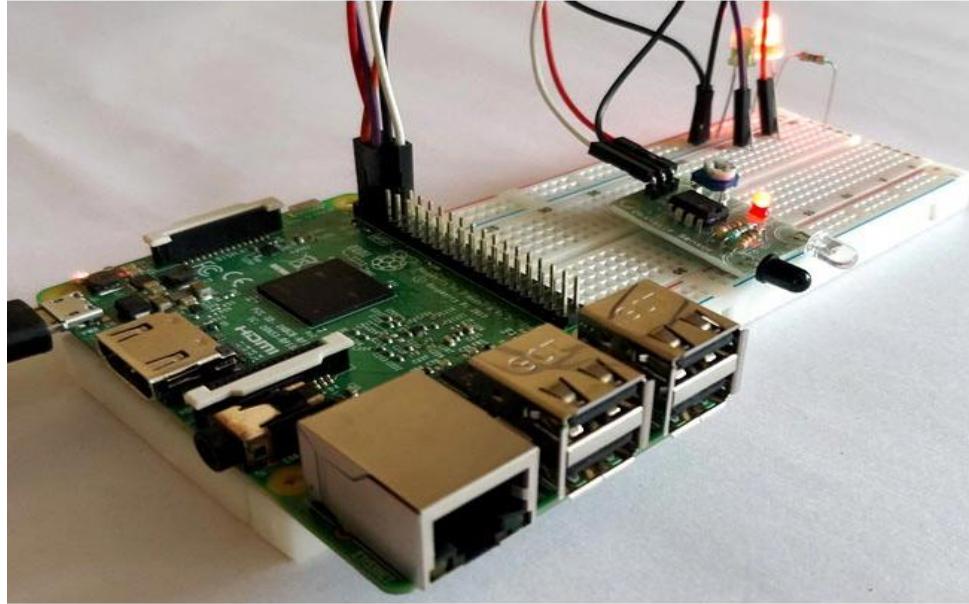
- Below command is used as forever loop, with this command the statements inside this loop will be executed continuously.

```
While 1:
```

- Once you have created your python code, execute it using the run command. If the program is executed without any errors you should get the following screen.



- You should also see the red colour LED going high when there is no object in front of the sensor as shown below.



- Now, bring something close to the IR led and you should notice the red LED turning off and the Green turning on.

➤ The Full Code:

```
import RPi.GPIO as IO
import time
IO.setwarnings(False)
IO.setmode(IO.BCM)

IO.setup(2,IO.OUT) #GPIO 2 -> Red LED as output
IO.setup(3,IO.OUT) #GPIO 3 -> Green LED as output
IO.setup(14,IO.IN) #GPIO 14 -> IR sensor as input

while 1:
    if(IO.input(14)==True): #object is far away
        IO.output(2,True) #Red led ON
        IO.output(3,False) # Green led OFF

    if(IO.input(14)==False): #object is near
        IO.output(3,True) #Green led ON
        IO.output(2,False) # Red led OFF
```

➤ 5.4. Laser Sensor:

Laser sensors come in several configurations, with some detecting presence and others measuring distance. A proximity type laser sensor, also called a laser photoelectric sensor, is commonly used to detect presence of a part. The laser distance sensors use a focused, coherent light to measure distance to a target object.

Laser sensors generally work well in a dirty environment since the focused light can “burn” through dust. The focused beam also enables long sensing distances, and detection of small objects or targets through small openings. Laser sensors are often used in process monitoring and closed-loop feedback control systems. Material handling is a popular application to enable positioning of cranes, gantries and automatic guided vehicles. A few of the many other applications include component alignment, height measurement, robot positioning and weld head location.

At times, shiny or transparent objects can cause problems. Because the laser distance sensor detects reflected or through-beam light, transparency and surface reflectivity may cause complications. Applications requiring bouncing a laser off a shiny surface or looking through a transparent surface should be carefully tested to make sure the measurement works as required. For example, the laser may need to be mounted at a slight angle to a shiny surface or adjusted to a lower intensity to properly detect a shiny object, while the intensity may need to be increased to burn through a transparent object.

➤ 5.4.1. Different types of laser sensors:

- Charge-coupled devices (CCD)
- Complimentary metal oxide semiconductors (CMOS)
- Position-sensitive detectors (PSD)
- Photoelectric sensors

Depending on the requirements of a particular application, one or more laser sensors might operate simultaneously within a piece of equipment. All but the photoelectric laser sensors function by triangulated signal reception. A laser diode emits a beam of light toward an object. The beam diffuses and reflects back to the sensor, which interprets the data and provides the required information.

CCD sensors contain millions of tiny cells that convert received light into electrons. These charges then cross a chip that interprets the data. The CCD sensors produce high quality and low noise images, without regard to color, texture or quantity of light. Manufacturers frequently combine CCD sensors with PSD technology for image reproduction. CCD sensors are usually more expensive and require more electricity than other sensor types.

CMOS sensors also contain millions of cells or pixels, which convert light into energy, but the wiring and transistors within the sensor are constructed so that data from each cell can be obtained individually without the need to transfer energy over a chip. CMOS sensors operate at close or long-distance ranges, regardless of light intensity or reflectivity. This type of sensor is more susceptible to noise than a CMOS sensor and may not produce as clear an image. A CMOS might be combined with a CCD sensor, forming a laser displacement sensor that is commonly used in industrial laser measurement.

PSD sensors can locate a beam of light in one or two dimensions. For example, when the sensor surface receives the signal, the position might be determined both horizontally and vertically. This type of laser sensor is generally suitable for use in both short and long range applications. The technology is often employed by the military, as the laser sensor accurately detects motion, position, and vibration.

Photoelectric laser emits an infrared or visible beam of light aimed at and received by an opposing photoelectric sensor detector. These laser sensors are usually designed to detect the absence or presence of objects. When the light beam traveling to the sensor is disrupted, the laser sensor relays a signal, and the equipment performs a specific function. Photoelectric lasers might be used to count the items being transported along a conveyor belt or to provide a barrier as part of a security system. The sensitivity of photoelectric sensors varies, but some are quite sensitive. Certain models are capable of detecting objects at close range that measure no more than 0.40 inches (1 mm) in diameter. The length of the beam transmitted also varies, with some units able to transmit up to 197 feet (60 m).

➤ 5.4.2. Photoelectric Laser Sensor Principles:

A photoelectric sensor emits a light beam (visible or infrared) from its light-emitting element. A reflective-type photoelectric sensor is used to detect the light beam reflected from the target. A thrubeam type sensor is used to measure the change in light quantity caused by the target crossing the optical axis.

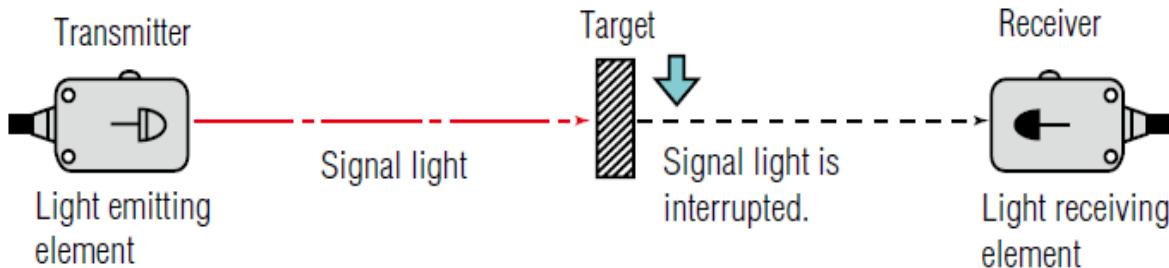
A beam of light is emitted from the light emitting element and is received by the light receiving element.



➤ 5.4.3. Models:

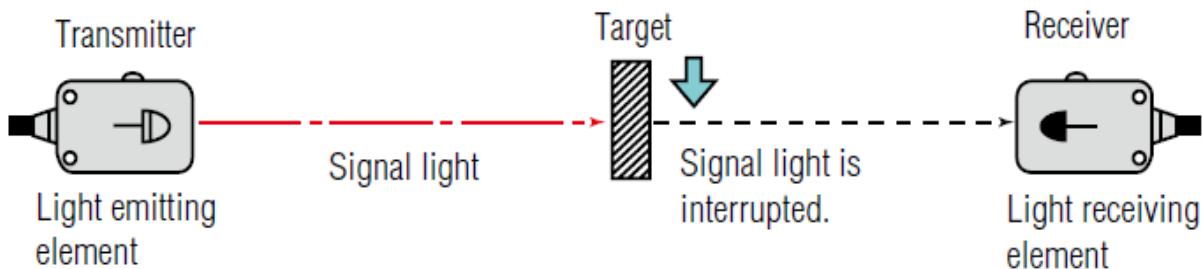
- **Reflective model:**

Both the light emitting and light receiving elements are contained in a single housing. The sensor receives the light reflected from the target.



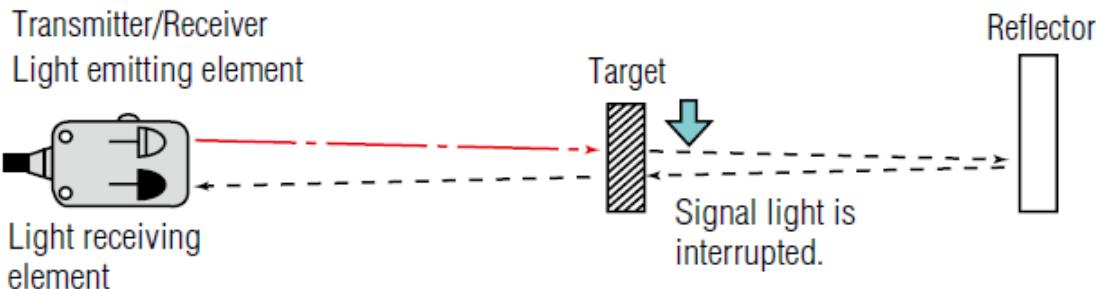
- **Thrubeam model:**

The transmitter and receiver are separated. When the target is between the transmitter and receiver, the light is interrupted



- **Retroreflective model:**

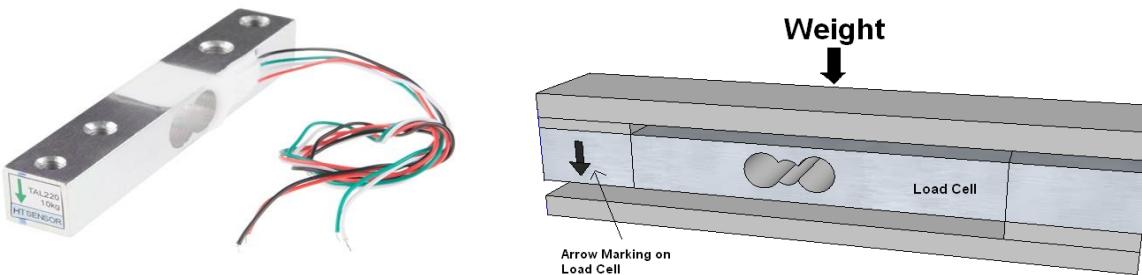
Both the light emitting and light receiving elements are contained in same housing. The light from the emitting element hits the reflector and returns to the light receiving element. When a target is present, the light is interrupted.



➤ 5.5. Load Cell:

Load cell is a type of transducer which performs the functionality of converting force into an electric output which can be measured. You can find load cell at the heart of any weighing machine or electric scales. This type of transducer is highly accurate which provides user with required information that is difficult to obtain by other technology owing to certain commercial factors.

It is basically a device that measures strain and then converts force into electric energy which serves as measurement for scientists and workers. The strain measurement by load cells helps in maintaining integrity of the unit under pressure and protects people and equipment nearby.

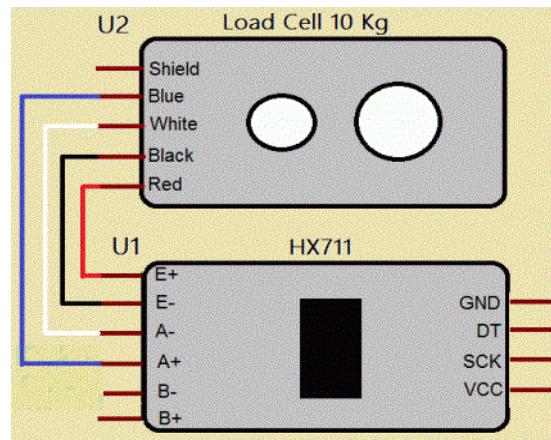


To connect load cell to Arduino Uno, a chip named HX711 is needed between the two. HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor. The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. Onchip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization. There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

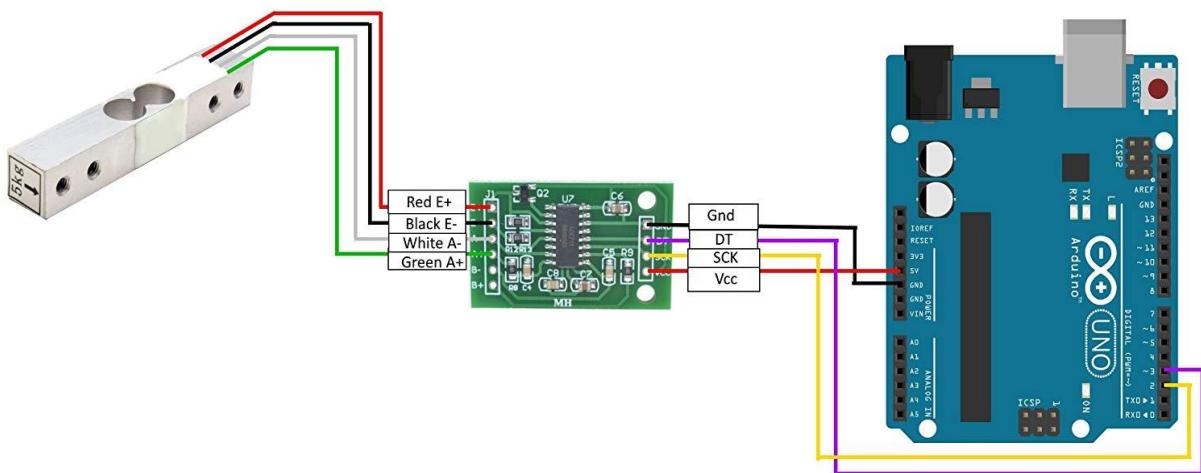
➤ 5.5.1. Connections:

Most Load cell have four wires red, black, green and white. On HX711 board you will find E+, E-, A+, A- and B+, B- connections:

- Red wire to E+
- Black wire to E-
- Green wire to A-
- White wire to A+



HX711 Module operates at 5V and communication is done using serial SDA and SCK pins.
Connection to Arduino:



Chapter six

Barcodes

Automatic identification and data collection (AIDC), also known as Auto ID or Keyless Data Entry, is the generic term for a number of technologies that help eliminate human error and reduce time and labor by replacing manual methods of data entry and data collection. Bar coding is one of several AIDC technologies that also include magnetic striping, smart cards, radio frequency identification (RFID), and Wireless LANs (WLANS).

QR Code



1D Barcode



1 234567 890128 >

2D Barcode



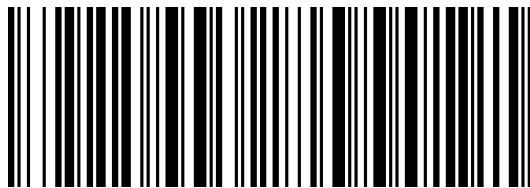
A barcode is a visual, machine-readable representation of data; the data usually describes something about the object that carries the barcode. Bar coding is a fast, easy and accurate data entry. Once bar-coded data is collected via a bar code scanner, activity can be tracked efficiently and accurately at speeds not possible using manual data entry and reading systems. Bar codes, such as the familiar Universal Product Code (UPC) symbol used on packaged goods, first appeared in grocery stores in the early 1970s. Aided by new technologies such as mobile and wireless printing, bar coding has evolved into a productivity enhancement tool widely used by business and industry for collecting and processing information.

Bar codes are fast. Speeding both data entry and data collection, they allow instantaneous, real-time data capture and exchange. Bar codes are also accurate. Studies show that the entry and read error rates when using bar code technology is approximately one error in one million characters, vs. one error per every 300 characters using manual key entry. Bar codes permit companies to track information and activity as it occurs, allowing their decisions to be based on concrete, current, and accurate information.

➤ **6.1. Bar Code Design:**

Bar codes encode data such as part number, serial number, supplier number, quantity, or transaction code into the form of black and white stripes or “bars.” A number of bar code standards have been developed and refined over the years into accepted languages called “symbologies”. Bar code symbologies can be either linear or two-dimensional.

- **6.1.1. Linear bar code:**



A linear bar code symbology consists of a single row of dark lines and white spaces of varying but specified width and height, as indicated by the example below.

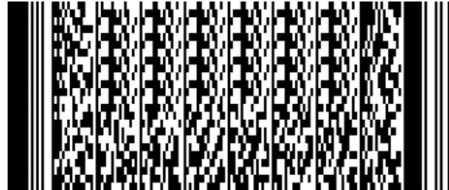
➤ **The most popular Linear barcode types:**

- Code 39 (Code 3 of 9)
- Code 128
- Interleaved 2 of 5 (I2 of 5)
- EAN8 and EAN13
- Databar
- Intelligent Mail
- Postnet

➤ **Linear Barcode Facts:**

- Linear Barcodes use start and stop characters to determine the beginning and end of barcodes.
- Linear Barcode cannot hold/encode an enormous amount of data.
- Linear Barcode Scanners are less expensive than 2D Barcode Scanners.
- Linear Barcodes may require a check character calculation.
- Code 128 and Code 39 are very popular Linear barcodes types.
- Code 128, Code 39, and Code 93 can encode alphanumeric data.
- Code 39, Interleaved 2 of 5, and Postnet do not require check digits.
- Interleaved 2 of 5 and Code 128 C encode numeric data in pairs.
- Databar has several subsets including: Limited, Expanded, and Omni-Directional.

- **6.1.2. 2-D symbology:**



2-D barcodes “stack” information to allow more information to be stored. They are configured either as stacked linear bar codes, or as matrix symbols that use regularly shaped black or white cells to encode data. A one-inch by one-inch 2-D matrix symbology can encode and store the entire U.S. Constitution!

➤ **The most well-known 2-D barcodes:**

- Data Matrix
- QR Code
- Aztec
- PDF417
- Maxicode

➤ **2-D Barcodes Facts:**

- 2-D Barcodes can encode hundreds of characters.
- 2-D Barcodes require a 2-D scanner to scan them.
- With the same data, 2-D Barcodes, compared to Linear barcodes, are smaller.
- Instead of check digits, 2-D barcodes use Error Correction.
- MicroPDF417 and MacroPDF417 are symbologies derived from PDF417.
- Acuity CiMatrix / Siemens invented the Data Matrix ECC200 symbology and placed it in the public domain.

➤ **6.2. Symbologies:**

The mapping between messages and barcodes is called a symbology. The specification of a symbology includes the encoding of the message into bars and spaces, any required start and stop markers, the size of the quiet zone required to be before and after the barcode, and the computation of a checksum.

Linear symbologies can be classified mainly by two properties:

- **Continuous vs. discrete:**

- Characters in discrete symbologies are composed of n bars and $n - 1$ spaces. There is an additional space between characters, but it does not convey information, and may have any width as long as it is not confused with the end of the code.
- Characters in continuous symbologies are composed of n bars and n spaces, and usually abut, with one character ending with a space and the next beginning with a bar, or vice versa. A special end pattern that has bars on both ends is required to end the code.

- **Two-width vs. many-width**

- A two-width, also called a binary bar code, contains bars and spaces of two widths, "wide" and "narrow". The precise width of the wide bars and spaces is not critical; typically it is permitted to be anywhere between 2 and 3 times the width of the narrow equivalents.
- Some other symbologies use bars of two different heights (POSTNET), or the presence or absence of bars (CPC Binary Barcode). These are normally also considered binary bar codes.
- Bars and spaces in many-width symbologies are all multiples of a basic width called the module; most such codes use four widths of 1, 2, 3 and 4 modules.

Some symbologies use interleaving. The first character is encoded using black bars of varying width. The second character is then encoded by varying the width of the white spaces between these bars. Thus characters are encoded in pairs over the same section of the barcode. Interleaved 2 of 5 is an example of this.

Stacked symbologies repeat a given linear symbology vertically.

The most common among the many 2D symbologies are matrix codes, which feature square or dot-shaped modules arranged on a grid pattern. 2D symbologies also come in circular and other patterns and may employ steganography, hiding modules within an image (for example, DataGlyphs).

Linear symbologies are optimized for laser scanners, which sweep a light beam across the barcode in a straight line, reading a slice of the barcode light-dark patterns.

Scanning at an angle makes the modules appear wider, but does not change the width ratios. Stacked symbologies are also optimized for laser scanning, with the laser making multiple passes across the barcode.

➤ **6.3. Bar Coding System Requirements:**

To establish a bar code system for AIDC, four primary components are required. They are a bar code printer, a label for item tracking, scanning equipment for data collection, and an external database for bar code data capture and relay.

1. The Bar Code Printer:

Bar codes are typically printed by one of four types of printers: dot matrix, ink jet, laser, or thermal. They will be discussed in detail later on in this chapter.

2. The Bar Code Label:

Labels are commonly applied on parts, product subassemblies, products, packages and shipping containers, allowing the item to be identified and tracked during internal processing or after it is shipped or sold. An item label can contain any combination of text, graphics, or barcode information, but it is the bar code symbology that facilitates the item tracking process. Bar code labels can be applied to anything—or anyone—that needs to be identified or tracked. Bar codes are not confined to adhesive labels, but also can be printed on receipt paper, tickets, tags and plastic cards, depending on the application and the selected printer.

3. Scanning Equipment for Data Collection:

In the data collection phase, scanners are used to instantly and accurately read, capture and decipher the information contained in the bar code label. Scanners read information much faster and more reliably than humans, thus significantly reducing the likelihood of error. Scanners also act as decoders, converting bar code information into a signal that can be understood by a computer system linked to a scanner.

4. Data Capture Via an External Database:

The fourth and final component of a bar code system is the external database. Bar code applications commonly rely on the availability of external-data computer systems to match an item's unique bar code with pertinent information about the article from a related database. The computer mainframe collects and interprets the data transmitted from the scanner and links the bar code information to a detailed data file on that item. Such data files contain information such as a detailed product description, price, and inventory quantity, enabling transactions and activity to be monitored in real-time. Without this external database, the bar code can have no value.

➤ 6.4. Key Benefits of Bar Coding:

Implementing a bar code system offers tremendous advantages. Among the most compelling benefits are:

- **Accuracy:**

Bar coding increases accuracy by reducing the likelihood of human errors from mislabeled or misread items. Good print quality is important to ensure that the scanner can accurately read the bar code.

- **Ease of Use:**

With the proper hardware and software in place, the automation provided by a bar code system greatly simplifies information collection, processing and tracking.

- **Uniform Data Collection:**

Standardized bar code symbologies and compliance labeling ensure that bar code information is captured and relayed in a fashion that is universally understood and accepted.

- **Timely Feedback:**

Data is captured in real-time, enabling decisions to be made from current information.

- **Improved Productivity:**

Automation of formerly manual tasks increases efficiency and allows personnel to be employed in other, more productive areas. Bar coding also can improve a company's quality by improving the monitoring and tracking of manufacturing processes.

- **Increased Profitability:**

Increased efficiencies from bar coding—such as improved workflow and reduced error rates—enable companies to save costs and improve their bottom line, as well as potentially improve sales through heightened customer satisfaction. In fact, the return on investment for an AIDC system is typically one year, and often substantially less.

➤ 6.5. Common Bar Code Applications

Generally, any industry or company can utilize bar coding to track and improve current processes and operations. The convenience of bar coding surrounds us in everyday applications. Some of the most common bar code applications are:

- **Manufacturing Processes and Supply Chain Management:**

Manufacturing businesses depend on a well-coordinated chain of events to make their operations work effectively. Many companies initiated bar code labeling at the shipping dock to support their customers' compliance requirements. Today's ERP (Enterprise Resource Planning) systems depend on bar coding and data collection systems to provide information crucial to the entire manufacturing operation.

- **Time and Attendance:**

Any company that uses time clocks can provide employees with badges printed with bar codes. The bar codes, scanned at clock stations, can provide attendance data directly to a computerized payroll program.

- **Package Delivery:**

For common carriers such as UPS® or FedEx®, the bar code label enables the package to be tracked as it passes through diverse sorting hubs en route to its ultimate destination. Throughout the package's journey, each sorting hub scans the package to register its receipt before passing it onward. Thus, if the package is ever misplaced or its arrival delayed, it can be tracked by its bar code tracking number to the exact point in the process where it halted.

- **Healthcare:**

The use of bar coding in healthcare increases the accuracy of medication administration, provides management tracking and reporting tools, enhances documentation and ensures greater patient safety. From patient ID wristbands to specimen and prescription labeling, bar code printers and supplies offer dependable solutions.

- **Hospitality and Travel:**

Bar code label and receipt printing is frequently used for "queue busting." With shorter lines and fewer hassles, guests have more time to relax. Orders are processed more quickly so lines move faster and customers are served more quickly. Other bar code printing examples include airline baggage tags, amusement park wristbands, event tickets, and boarding passes.

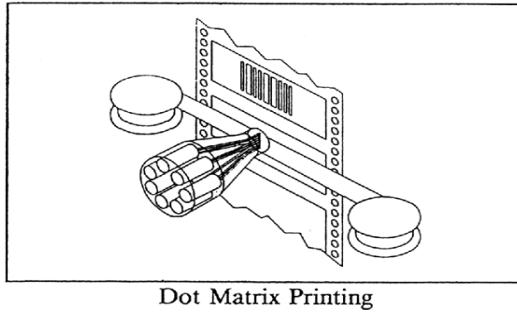
- **Security:**

Bar codes can be printed on plastic ID and keyless entry cards to aid in access control. In the airline industry, bar codes are being used to link baggage to passengers and to track the movement of people, luggage and cargo. Bar code labels are also used successfully in law enforcement to track evidence

➤ 6.6. Bar Code Print Technologies:

- **Dot Matrix:**

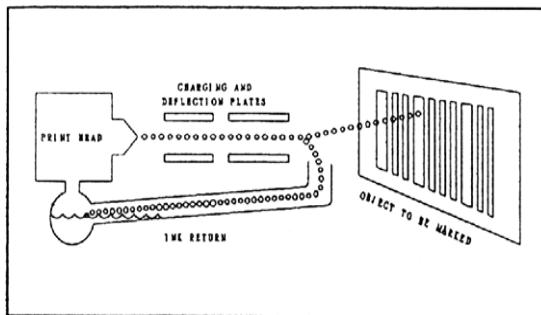
Dot matrix print technology is one of the oldest techniques used for on-site label printing. The typical dot matrix bar code printer is a modified line printer requiring pin-feed paper stock. Solenoid-driven needles strike an ink-coated nylon ribbon, transferring ink onto the paper or label. The image is built up dot-by-dot in a matrix as the needle and paper are moved relative to one another.



Dot Matrix Printing

- **Ink Jet:**

Ink jet printing is a common direct marking process and a favorite on high-speed production lines. Ink droplets are selectively deflected between a moving product and an ink return channel. Ink jet printing is used primarily for printing cartons or product packages with bar codes and human-readable data at very high speed.



Deflection Ink Jet

- **Laser:**

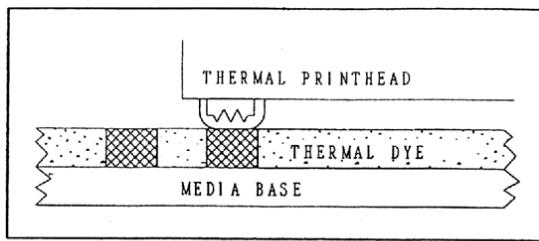
The laser printer works much like a photocopier; it projects controlled streams of ions onto the surface of a print drum, resulting in a charged image. The charged image then selectively attracts toner particles, transferring the image onto the paper substrate by means of pressure. The pressure from the printhead and drum then fuse the image to the paper, creating the image.

- **Thermal:**

Thermal printing is classified as either direct thermal or thermal transfer. The two technologies are suited to different applications.

- **Direct Thermal Printing:**

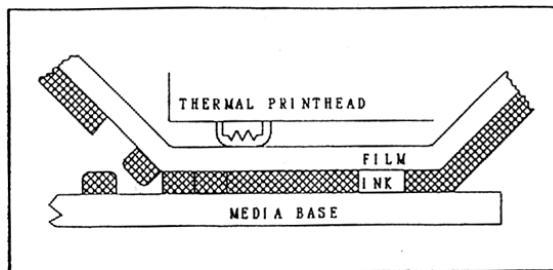
Direct thermal printing is an old technology, originally designed for copiers and fax machines, that utilizes chemically coated paper. It has since been transformed into a highly successful technology for bar coding. The direct thermal printhead consists of a long, linear array of tiny resistive heating elements (about 100 to 300 per inch) that are arranged perpendicular to the paper flow. Each printhead element locally heats an area directly below it on the chemically coated paper. This produces a chemical reaction that causes a black dot to form in that area. The image is built by rows of dots that are formed as the media passes beneath the active edge of the printhead.



Direct Thermal Printing

- **Thermal Transfer:**

Thermal transfer printers use the same basic technology as direct thermal printers, but replace the chemically coated material with a non-sensitized face stock and a special, inked ribbon. A durable, polyester ribbon film coated with dry thermal transfer ink is placed between the thermal printhead and label. The thermal printhead transfers the ink onto the label surface, where it cools and anchors to the media surface. The polyester ribbon is then peeled away, leaving behind a stable, passive image.



Thermal Transfer Printing

➤ **Bar Code Print Technology Matrix:**

Technology	Print Quality	Scanner Readability	Initial Installation Cost	Long-Term Maintenance Cost	Materials Waste
<i>Dot Matrix</i>	Fair	Low	Low/Moderate	Moderate/High	High
<i>Ink Jet</i>	Moderate	Low/Moderate	High	Moderate/High	High
<i>Laser</i>	Moderate	Moderate	Moderate/High	Moderate/High	High
<i>Direct Thermal</i>	Modera te/ Excelle nt	Moderate/ Excellent	Moderate/High	Low	Low
<i>Thermal Transfer</i>	Excellent	Excellent	Moderate/High	Low	Low

Chapter seven

PHP & MySQL

Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism: Hypertext Preprocessor.



PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP code may be executed with a command line interface (CLI) [is a means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines)].

A program which handles the interface is called a command language interpreter or shell (computing)], embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable.

The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

➤ 7.1. Usage:

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications.

PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). Most web hosting providers support PHP for use by their clients.

It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

You can use PHP to create dynamic web pages, collect form data, and send or receive cookies.

PHP enables you to choose not only the operating system of your choice but also allows you to have choices to use a web server that you are familiar with. It also enables beginners and professionals to write scripts in their own ways as it allows procedural as well as object-oriented programming.

PHP not only enables you to output HTML but also lets you include images, PDFs, videos, and sounds. PHP can auto-generate XHTML and XML files.

PHP provides support to protocols like LDAP, HTTP, COM, POP3, etc. It also supports WDDX complex data exchange.

➤ 7.2. Applications of PHP Scripts:

- **Server-Side Scripting**

Server side scripting is the first purpose of PHP. All you need to start working on a desktop PC with PHP is a PHP Parser, a webserver (such as Apache) and a web browser like Google Chrome.

- **Command Line Scripting**

If you want to use PHP on Linux or task scheduler on Windows, then you don't really need a web server, but only a PHP Parser. This is called "command line scripting".

- **Desktop Applications**

Although, PHP is not a suitable language for development of desktop applications, but it supports some advanced features like PHP-GTK which is basically an extension of PHP. PHP-GTK provides object-oriented user interface.

➤ 7.3. How to run a php file in XAMPP:

Step 1:

Start the XAMPP program. When started, XAMPP loads itself into your icon tray. The icon is orange with a white bone-like shape in its center. Single-click the icon to expand the Control Panel. Click on the "Start" button next to "Apache" to start your Apache Web server. When Apache is running, the word "Running" will appear next to it, highlighted in green. Also start "MySQL" if your PHP scripts depend on a MySQL database to run.

Step 2:

Place your PHP files in the "htdocs" folder located under the "XAMPP" folder on your C: drive. The file path is "C:\xampp\htdocs" for your Web server. Make sure your PHP files are saved as such; they must have the ".php" file extension. Open up any Web browser on your desktop and enter "localhost" into the address box. The browser will open a list of files stored under the "htdocs" folder on your computer. Click on the link to a PHP file and open it to run a script.

Step 3:

Create any folders you need to test PHP files in under the "htdocs" folder. If you create a folder named "scripts," then use the address "localhost/scripts" to open them in your browser.

➤ 7.4. Connecting to the database using PHP and MySQL:

As noted previously, one of the most important factors driving PHP's popularity over the last couple of years has been its support for a variety of databases, including MySQL, mSQL, Oracle, and Microsoft Access. By simplifying and streamlining database access, PHP enables developers to build complex datadriven web applications while enjoying short development cycles.

Support for MySQL has been available in PHP since version 3.x, and has gradually improved over subsequent releases. PHP 5.0 promises even better integration with the latest version of MySQL: the new MySQL extension in PHP 5.0 provides developers with both function and object-oriented APIs to common MySQL functions, and includes support for new and upcoming MySQL features like transactions, stored procedures, and prepared statements.

➤ 7.4.1. Functions used for database:

- **PHP mysqli_connect() Function:**

The mysqli_connect() function opens a new connection to the MySQL server.

- **Syntax:**

```
mysqli_connect(host,username,password,dbname,port,socket);
```

Parameter	Description
<i>Host</i>	Optional. Specifies a host name or an IP address
<i>Username</i>	Optional. Specifies the MySQL username
<i>Password</i>	Optional. Specifies the MySQL password
<i>Dbname</i>	Optional. Specifies the default database to be used
<i>Port</i>	Optional. Specifies the port number to attempt to connect to the MySQL server
<i>Socket</i>	Optional. Specifies the socket or named pipe to be used

- **Return Value:**

Returns an object representing the connection to the MySQL server

- **PHP mysqli_connect_errno() Function:**

The mysqli_connect_errno() function returns the error code from the last connection error, if any.

- **Syntax:**

```
mysqli_connect_errno();
```

- **Return Value:**

Returns an error code value. Zero if no error occurred

- **PHP mysqli_connect_error() Function:**

The mysqli_connect_error() function returns the error description from the last connection error, if any.

- **Syntax**

```
mysqli_connect_error();
```

- **Return Value:**

Returns a string that describes the error. NULL if no error occurred

- **PHP mysqli_query() Function:**

The mysqli_query() function performs a query against the database.

- **Syntax**

```
mysqli_query(connection,query,resultmode);
```

Parameter	Description
<i>Connection</i>	Required. Specifies the MySQL connection to use
<i>Query</i>	Required. Specifies the query string
<i>Resultmode</i>	Optional. A constant. Either: <ul style="list-style-type: none">• MYSQLI_USE_RESULT (Use this if we have to retrieve large amount of data)• MYSQLI_STORE_RESULT (This is default)

– **Return Value:**

For successful SELECT, SHOW, DESCRIBE, or EXPLAIN queries it will return a mysqli_result object. For other successful queries it will return TRUE. FALSE on failure

- **PHP mysqli_fetch_assoc() Function**

The mysqli_fetch_assoc() function fetches a result row as an associative array.

Note: Fieldnames returned from this function are case-sensitive.

– **Syntax:**

```
mysqli_fetch_assoc(result);
```

Parameter	Description
<i>Result</i>	Required. Specifies a result set identifier returned by mysqli_query(), mysqli_store_result() or mysqli_use_result()

– **Return Value:**

Returns an associative array of strings representing the fetched row. NULL if there are no more rows in result-set

- **PHP mysqli_free_result() Function:**

The mysqli_free_result() function frees the memory associated with the result.

- **Syntax:**

```
mysqli_free_result(result);
```

Parameter	Description
<i>Result</i>	Required. Specifies a result set identifier returned by mysqli_query(), mysqli_store_result() or mysqli_use_result()

- **PHP mysqli_close() Function:**

The mysqli_close() function closes a previously opened database connection.

- **Syntax:**

```
mysqli_close(connection);
```

Parameter	Description
<i>Connection</i>	Required. Specifies the MySQL connection to close

➤ 7.5. Other PHP Functions used in our code:

- **PHP pathinfo() Function:**

The pathinfo() function returns an array that contains information about a path.

The following array elements are returned:

- [dirname]
- [basename]
- [extension]

– **Syntax:**

`pathinfo(path,options)`

Parameter	Description
Path	Required. Specifies the path to check
Options	Optional. Specifies which array elements to return. Default is all Possible values: <ul style="list-style-type: none">• PATHINFO_DIRNAME - return only dirname• PATHINFO_BASENAME - return only basename• PATHINFO_EXTENSION - return only extension

Note: The `pathinfo()` function returns a string if not all elements are requested.

• **PHP `move_uploaded_file()` Function**

The `move_uploaded_file()` function moves an uploaded file to a new location.

This function returns TRUE on success, or FALSE on failure.

– **Syntax:**

`move_uploaded_file(file,newloc)`

Parameter	Description
file	Required. Specifies the file to be moved
newloc	Required. Specifies the new location for the file

Note: This function only works on files uploaded via HTTP POST.

Note: If the destination file already exists, it will be overwritten.

- **PHP filesize() Function:**

The filesize() function returns the size of the specified file.

This function returns the file size in bytes on success or FALSE on failure.

- **Syntax:**

```
filesize(filename)
```

Parameter	Description
Filename	Required. Specifies the file to check

Note: The result of this function are cached. Use clearstatcache() to clear the cache.

- **PHP explode() Function:**

The explode() function breaks a string into an array.

Note: The "separator" parameter cannot be an empty string.

Note: This function is binary-safe.

- **Syntax**

```
explode(separator,string,limit)
```

Parameter	Description
separator	Required. Specifies where to break the string
string	Required. The string to split
limit	Optional. Specifies the number of array elements to return. Possible values: <ul style="list-style-type: none">• Greater than 0 - Returns an array with a maximum of <i>limit</i> element(s)• Less than 0 - Returns an array except for the last - <i>limit</i> elements()• 0 - Returns an array with one element

- **PHP strlen() Function**

The `strlen()` function returns the length of a string.

- **Syntax:**

```
strlen(string)
```

Parameter	Description
<i>String</i>	Required. Specifies the string to check

- **Return Value:**

Returns the length of a string on success, and 0 if the string is empty

- **PHP fopen() Function**

The `fopen()` function opens a file or URL. If `fopen()` fails, it returns FALSE and an error on failure. You can hide the error output by adding an '@' in front of the function name.

- **Syntax**

```
fopen(filename,mode,include_path,context);
```

Parameter	Description
<i>filename</i>	Required. Specifies the file or URL to open
<i>mode</i>	Required. Specifies the type of access you require to the file/stream. Possible values: <ul style="list-style-type: none">• "r" (Read only. Starts at the beginning of the file)• "r+" (Read/Write. Starts at the beginning of the file)• "w" (Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist)• "w+" (Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist)• "a" (Write only. Opens and writes to the end of the file or creates a new file if it doesn't exist)

	<ul style="list-style-type: none"> creates a new file if it doesn't exist) "a+" (Read/Write. Preserves file content by writing to the end of the file) "x" (Write only. Creates a new file. Returns FALSE and an error if file already exists) "x+" (Read/Write. Creates a new file. Returns FALSE and an error if file already exists)
include_path	Optional. Set this parameter to '1' if you want to search for the file in the include_path (in php.ini) as well
context	Optional. Specifies the context of the file handle. Context is a set of options that can modify the behavior of a stream

Note: When writing to a text file, be sure to use the correct line-ending character! Unix systems use \n, Windows systems use \r\n, and Macintosh systems use \r as the line ending character. Windows offers a translation flag ('t') which will translate \n to \r\n when working with the file. You can also use 'b' to force binary mode. To use these flags, specify either 'b' or 't' as the last character of the mode parameter.

- **PHP fwrite() Function**

The fwrite() writes to an open file. The function will stop at the end of the file or when it reaches the specified length, whichever comes first.

This function returns the number of bytes written, or FALSE on failure.

- Syntax

`fwrite(file,string,length)`

Parameter	Description
file	Required. Specifies the open file to write to
string	Required. Specifies the string to write to the open file
length	Optional. Specifies the maximum number of bytes to write

Tip: This function is binary-safe (meaning that both binary data, like images, and character data can be written with this function).

- **PHP fclose() Function:**

The fclose() function closes an open file.

This function returns TRUE on success or FALSE on failure.

– Syntax

`fclose(file)`

Parameter	Description
File	Required. Specifies the file to close

- **PHP DateTime Function**

The date/time functions allow you to get the date and time from the server where your PHP script runs. You can then use the date/time functions to format the date and time in several ways.

Note: These functions depend on the locale settings of your server. Remember to take daylight saving time and leap years into consideration when working with these functions.

Runtime Configuration:

The behavior of these functions is affected by settings in `php.ini`:

Name	Description	Default	PHP Version
<code>date.timezone</code>	The default timezone (used by all date/time functions)	""	PHP 5.1
<code>date.default_latitude</code>	The default latitude (used by <code>date_sunrise()</code> and <code>date_sunset()</code>)	"31.7667"	PHP 5.0
<code>date.default_longitude</code>	The default longitude (used by <code>date_sunrise()</code> and <code>date_sunset()</code>)	"35.2333"	PHP 5.0
<code>date.sunrise_zenith</code>	The default sunrise zenith (used by <code>date_sunrise()</code> and <code>date_sunset()</code>)	"90.83"	PHP 5.0
<code>date.sunset_zenith</code>	The default sunset zenith (used by <code>date_sunrise()</code> and <code>date_sunset()</code>)	"90.83"	PHP 5.0

• PHP date_modify() Function

The date_modify() function modifies the timestamp.

– Syntax

`date_modify(object,modify);`

Parameter	Description
<i>object</i>	Required. Specifies a DateTime object returned by date_create()
<i>modify</i>	Required. Specifies a date/time string

– Return Value:

Returns a DateTime object on success. FALSE on failure

• PHP date_format() Function

The date_format() function returns a date formatted according to the specified format.

– Syntax:

`date_format(object,format);`

Parameter	Description
<i>Object</i>	Required. Specifies a DateTime object returned by date_create()
<i>Format</i>	Required. Specifies the format for the date. The following characters can be used: <ul style="list-style-type: none">• d - The day of the month (from 01 to 31)• D - A textual representation of a day (three letters)• m - A numeric representation of a month (from 01 to 12)• M - A short textual representation of a month (three letters)• Y - A four digit representation of a year• y - A two digit representation of a year• a - Lowercase am or pm• A - Uppercase AM or PM• h - 12-hour format of an hour (01 to 12)• H - 24-hour format of an hour (00 to 23)• i - Minutes with leading zeros (00 to 59)• s - Seconds, with leading zeros (00 to 59)

Chapter eight

Python programming

Python is an interpreted, high-level, general-purpose programming language.

Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.



Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and with Raspberry Pi lets you connect your project to the real world.

Python syntax is very clean, with an emphasis on readability and uses standard English keywords. Start by opening IDLE from the desktop.

➤ 8.1. IDLE:

The easiest introduction to Python is through IDLE, a Python development environment.

Open IDLE from the Desktop or applications menu:



IDLE gives you a REPL (Read-Evaluate-Print-Loop), which is a prompt you can enter Python commands into. Because it's a REPL, you even get the output of commands printed to the screen without using print.

Note: two versions of Python are available — Python 2 and Python 3. Python 3 was first released in 2008 and Python 2 development ended with 2.7, which was released in 2010. Python 3 is recommended, but Python 2 is available for legacy applications which do not support Python 3 yet.

IDLE also has syntax highlighting built in and some support for autocompletion. You can look back on the history of the commands you've entered in the REPL with Alt + P (previous) and Alt + N (next).

➤ 8.2. Basic Python usage:

- **Indentation**

Some languages use curly braces { and } to wrap around lines of code which belong together, and leave it to the writer to indent these lines to appear visually nested. However, Python does not use curly braces but instead requires indentation for nesting. The indentation is necessary here. A second line indented would be a part of the loop, and a second line not indented would be outside of the loop. For example:

```
for i in range(2):  
    print("A")  
    print("B")
```

- **Variables**

To save a value to a variable, assign it like so:

```
name = "Bob"  
age = 15
```

Note that data types were not specified with these variables, as types are inferred, and can be changed later.

- **Comments**

Comments are ignored in the program but there for you to leave notes, and are denoted by the hash # symbol. Multi-line comments use triple quotes like so:

```
"""  
This is a very simple Python program that prints "Hello".  
That's all it does.  
"""
```

- **Lists**

Python also has lists (called arrays in some languages) which are collections of data of any type:

```
numbers = [1, 2, 3]
```

Lists are denoted by the use of square brackets [] and each item is separated by a comma.

- **Iteration**

Some data types are iterable, which means you can loop over the values they contain. For example a list:

```
numbers = [1, 2, 3]
```

```
for number in numbers:  
    print(number)
```

This takes each item in the list numbers and prints out the item:

Note we used the word number to denote each item. This is merely the word we chose for this - it's recommended you choose descriptive words for variables - using plurals for lists, and singular for each item makes sense. It makes it easier to understand when reading.

Other data types are iterable, for example the string:

```
dog_name = "BINGO"  
for char in dog_name:  
    print(char)
```

This loops over each character and prints them out:

- **Range**

The integer data type is not iterable and trying to iterate over it will produce an error.

However you can make an iterable object using the range function:

```
for i in range(3):  
    print(i)
```

range(5) contains the numbers 0, 1, 2, 3 and 4 (five numbers in total). To get the numbers 1 to 5 (inclusive) use range(1, 6).

- **Length**

You can use functions like len to find the length of a string or a list:

```
name = "Jamie"  
print(len(name)) # 5  
names = ["Bob", "Jane", "James", "Alice"]  
print(len(names)) # 4
```

- **If statements**

You can use if statements for control flow:

```
name = "Joe"  
if len(name) > 3:  
    print("Nice name,")  
    print(name)  
else:  
    print("That's a short name,")  
    print(name)
```

➤ 8.3. Python Libraries we used:

➤ 8.3.1. Time:

This module provides various time-related functions. like:

- **time.sleep(secs):**

Suspend execution of the calling thread for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time. The actual suspension time may be less than that requested because any caught signal will terminate the sleep() following execution of that signal's catching routine. Also, the suspension time may be longer than requested by an arbitrary amount because of the scheduling of other activity in the system.

- **time.gmtime([secs]):**

Convert a time expressed in seconds since the epoch to a struct_time in UTC in which the dst flag is always zero. If secs is not provided or None, the current time as returned by time() is used. Fractions of a second are ignored.

- **time.localtime([secs]):**

Like gmtime() but converts to local time. If secs is not provided or None, the current time as returned by time() is used. The dst flag is set to 1 when DST applies to the given time.

➤ 8.3.2. RPi.GPIO:

This package provides a class to control the GPIO on a Raspberry Pi.

To import the RPi.GPIO module:

```
" import RPi.GPIO as GPIO "
```

By doing it this way, you can refer to it as just GPIO through the rest of your script.

Pin numbering:

There are two ways of numbering the IO pins on a Raspberry Pi within RPi.GPIO. The first is using the BOARD numbering system. This refers to the pin numbers on the P1 header of the Raspberry Pi board. The advantage of using this numbering system is that your hardware will always work, regardless of the board revision of the RPi. You will not need to rewire your connector or change your code.

The second numbering system is the BCM numbers. This is a lower level way of working - it refers to the channel numbers on the Broadcom SOC. You have to always work with a diagram of which channel number goes to which pin on the RPi board. Your script could break between revisions of Raspberry Pi boards.

To specify which you are using using (mandatory):

```
GPIO.setmode(GPIO.BOARD)  
# or  
GPIO.setmode(GPIO.BCM)
```

To detect which pin numbering system has been set (for example, by another Python module):

```
mode = GPIO.getmode()  
The mode will be GPIO.BOARD, GPIO.BCM or None
```

Warnings:

It is possible that you have more than one script/circuit on the GPIO of your Raspberry Pi. As a result of this, if RPi.GPIO detects that a pin has been configured to something other than the default (input), you get a warning when you try to configure a script.

To disable these warnings:

```
GPIO.setwarnings(False)
```

Setup up a channel:

You need to set up every channel you are using as an input or an output.

To configure a channel as an input:

```
GPIO.setup(channel, GPIO.IN)  
(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)).
```

To set up a channel as an output:

```
GPIO.setup(channel, GPIO.OUT)  
(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)).
```

You can also specify an initial value for your output channel:

```
GPIO.setup(channel, GPIO.OUT, initial=GPIO.HIGH)
```

Input:

To read the value of a GPIO pin:

```
GPIO.input(channel)
```

(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)). This will return either 0 / GPIO.LOW / False or 1 / GPIO.HIGH / True.

Output:

To set the output state of a GPIO pin:

`GPIO.output(channel, state)`

(where channel is the channel number based on the numbering system you have specified (BOARD or BCM)).

State can be 0 / GPIO.LOW / False or 1 / GPIO.HIGH / True.

Cleanup:

At the end of any program, it is good practice to clean up any resources you might have used. This is no different with RPi.GPIO. By returning all channels you have used back to inputs with no pull up/down, you can avoid accidental damage to your RPi by shorting out the pins. Note that this will only clean up GPIO channels that your script has used. Note that `GPIO.cleanup()` also clears the pin numbering system in use.

To clean up at the end of your script:

`GPIO.cleanup()`

It is possible that don't want to clean up every channel leaving some set up when your program exits. **You can clean up individual channels, a list or a tuple of channels:**

`GPIO.cleanup(channel)`

`GPIO.cleanup((channel1, channel2))`

`GPIO.cleanup([channel1, channel2])`

➤ 8.3.3. Requests:

Dealing with HTTP requests is not an easy task in any programming language. If we talk about Python, it comes with two built-in modules, `urllib` and `urllib2`, to handle HTTP related operation. Both modules come with a different set of functionalities and many times they need to be used together. The main drawback of using `urllib` is that it is confusing (few methods are available in both `urllib`, `urllib2`), the documentation is not clear and we need to write a lot of code to make even a simple HTTP request.

To make these things simpler, one easy-to-use third-party library, known as Requests, is available and most developers prefer to use it instead of `urllib/urllib2`. It is an Apache2 licensed HTTP library powered by `urllib3` and `httplib`.

- **Installing the Requests Module:**

Installing this package, like most other Python packages, is pretty straight-forward. You can either download the [Requests source code](#) from Github and install it or use pip:

```
$ pip install requests
```

To verify the installation, you can try to import it like below:

```
import requests
```

If you don't receive any errors importing the module, then it was successful.

- **Making a GET Request:**

GET is by far the most used HTTP method. We can use GET request to retrieve data from any destination. Let me start with a simple example first. Suppose we want to fetch the content of the home page of our website and print out the resultin HTML data. Using the Requests module, we can do it like below:

```
import requests
r = requests.get('https://api.github.com/events')
print(r.content)
```

It will print the response in an encoded form. If you want to see the actual text result of the HTML page, you can read the .text property of this object. Similarly, the status_code property prints the current status code of the URL:

```
import requests
r = requests.get('https://api.github.com/events')
print(r.text)
print(r.status_code)
```

requests will decode the raw content and show you the result. If you want to check what type of encoding is used by requests, you can print out this value by calling .encoding. Even the type of encoding can be changed by changing its value.

- **Passing Parameters in GET**

In some cases, you'll need to pass parameters along with your GET requests, which take the form of query strings. To do this, we need to pass these values in the params parameter, as shown below:

```
import requests
payload = {'user_name': 'admin', 'password': 'password'}
r = requests.get('http://httpbin.org/get', params=payload)
print(r.url)
print(r.text)
```

Here, we are assigning our parameter values to the payload variable, and then to the GET request via params. The above code will return the following output:

```
http://httpbin.org/get?password=password&user_name=admin
{"args": {"password": "password", "user_name": "admin"}, "headers": {"Accept": "*/*", "Accept-Encoding": "gzip, deflate", "Connection": "close", "Host": "httpbin.org", "User-Agent": "python-requests/2.9.1"}, "origin": "103.9.74.222", "url": "http://httpbin.org/get?password=password&user_name=admin"}
```

As you can see, the Requests library automatically turned our dictionary of parameters to a query string and attached it to the URL.

Note: that you need to be careful what kind of data you pass via GET requests since the payload is visible in the URL, as you can see in the output above.

- **Making POST Requests**

HTTP POST requests are opposite of the GET requests as it is meant for sending data to a server as opposed to retrieving it. Although, POST requests can also receive data within the response, just like GET requests.

Instead of using the get() method, we need to use the post() method. For passing an argument, we can pass it inside the data parameter:

```
import requests
payload = {'user_name': 'admin', 'password': 'password'}
r = requests.post("http://httpbin.org/post", data=payload)
print(r.url)
print(r.text)
```

Output:

```
http://httpbin.org/post
{"args": {}, "data": "", "files": {}, "form": {"password": "password", "user_name": "admin"}, "headers": {"Accept": "*/*", "Accept-Encoding": "gzip, deflate", "Connection": "close", "Content-Length": "33", "Content-Type": "application/x-www-form-urlencoded", "Host": "httpbin.org", "User-Agent": "python-requests/2.9.1"}, "json": null, "origin": "103.9.74.222", "url": "http://httpbin.org/post"}
```

The data will be "form-encoded" by default. You can also pass more complicated header requests like a tuple if multiple values have same key, a string instead of a dictionary, or a multipart encoded file.

- **Sending Files with POST**

Sometimes we need to send one or more files simultaneously to the server. For example, if a user is submitting a form and the form includes different form-fields for uploading files, like user profile picture, user resume, etc. Requests can handle multiple files on a single request. This can be achieved by putting the files to a list of tuples, like below:

```
import requests
url = 'http://httpbin.org/post'
file_list = [
    ('image', ('image1.jpg', open('image1.jpg', 'rb'), 'image/png')),
    ('image', ('image2.jpg', open('image2.jpg', 'rb'), 'image/png'))
]
r = requests.post(url, files=file_list)
print(r.text)
```

The tuples containing the files' information are in the form (field_name, file_info).

➤ 8.3.4. Pygame:

Pygame is a Python wrapper module for the SDL multimedia library. It contains python functions and classes that will allow you to use SDL's support for playing cdroms, audio and video output, and keyboard, mouse and joystick input.

Pygame comes with support for interfacing cameras, allowing you to capture still images, watch live streams, and do some simple computer vision.

- **Import and Init:**

```
import pygame
import pygame.camera
from pygame.locals import *

pygame.init()
pygame.camera.init()
```

- **Capturing a Single Image:**

Now we will go over the simplest case of opening a camera and capturing a frame as a surface. In the below example, we assume that there is a camera at /dev/video0 on the computer, and initialize it with a size of 640 by 480. The surface called image is whatever the camera was seeing when get_image() was called.

```
cam = pygame.camera.Camera("/dev/video0", (640,480))
cam.start()
image = cam.get_image()
```

- **Listing Connected Cameras:**

If you don't know the exact path of the camera, We can ask the module to provide a list of cameras attached to the computer and initialize the first camera in the list.

```
camlist = pygame.camera.list_cameras()
if camlist:
    cam = pygame.camera.Camera(camlist[0], (640,480))
```

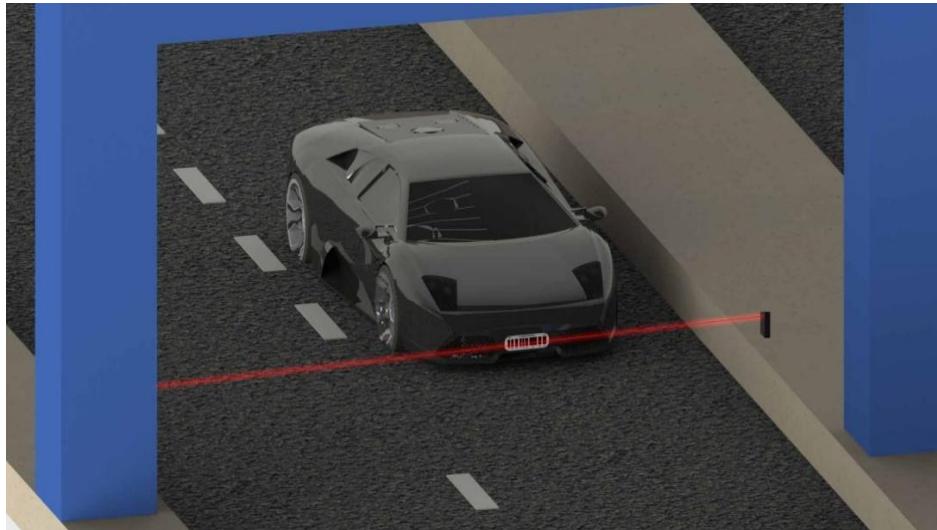
- **Using Camera Controls:**

Most cameras support controls like flipping the image and changing brightness. `set_controls()` and `get_controls()` can be used at any point after using `start()`.

```
cam.set_controls(hflip = True, vflip = False)
print camera.get_controls()
```


Chapter nine

System



This system does not require any manual operation of toll barriers and collection of toll amounts, it is completely automated toll collection system.

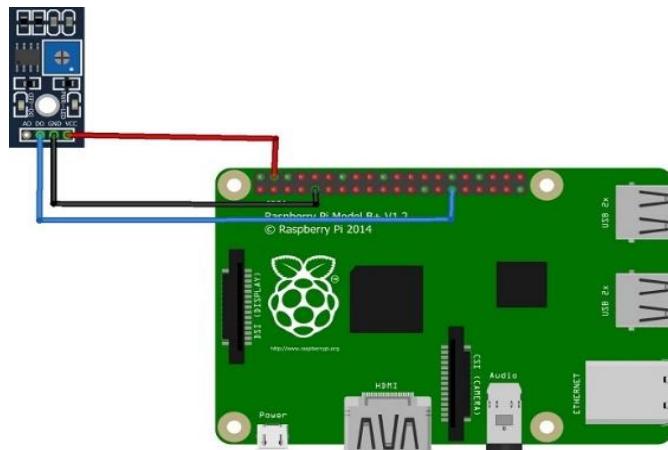
When the Laser sensor is triggered the camera captures a picture of the car and sends it to the Raspberry pi, then the pi will send the picture to the server where it will be processed and if a barcode is detected it will search in the database for the car owner account and the toll will be collected, also the picture will be stored in the database for future use. If the barcode is not found in the database the server will send a warning to the police with the picture of the car. There is a web interface so that the administrators can monitor and control the database and have access to the toll history of car users and collect data about them. There is also a user interface where the user can track his toll history and car information.

➤ 9.1. System Explanation:

➤ 9.1.1. Raspberry Pi Connections:

The raspberry pi is the provider of power to the whole system, it will provide the system with common Vcc (5v) and common GND (0v).

The IR (infrared) sensor detector is connected to the raspberry pi using pin 8 in raspberry pi and power connections. The IR emitter is connected with a resistance of $100\ \Omega$ to the power connections.



The IR sensor detector connection with Raspberry Pi

The camera used to capture the pictures of passing cars is connected to the raspberry pi using one of the USB ports. It will only capture a picture if the IR sensor is triggered.



The USB camera connection with Raspberry Pi

➤ 9.1.2. Raspberry Pi Code:

- Importing the required libraries:

```
import time
import serial
import RPi.GPIO as GPIO
import pygame
import pygame.camera
from pygame.locals import *
import requests
import os
```

- Setting the pin direction connected to the IR sensor as input:

```
GPIO.setmode(GPIO.BOARD)
GPIO.setup(8,GPIO.IN)
```

- Setting the camera device name and the name of the picture captured and size:

```
DEVICE = '/dev/video0'
SIZE = (640, 480)
FILENAME = 'capture.jpg'
```

- Initializing the serial connection between Raspberry pi and Arduino and setting the device number:

```
ser=serial.Serial("/dev/ttyACM0", 9600) #change ACM number as found
from ls /dev/tty/ACM*
ser.baudrate=9600
```

- Initializing the pygame library used to control the USB camera and initializing the camera and giving the pygame the active camera number then starting the camera to be ready to capture:

```
pygame.init()
pygame.camera.init()
camera = pygame.camera.Camera(pygame.camera.list_cameras()[0])
camera.start()
```

- Reading the input from the pin connected to the IR detector and then checking if the sensor is triggered meaning that something broke the connection between the IR emitter and detector. If the sensor is triggered (`PIR == 1`), the camera will be activated and an image will be captured and saved in the home folder in the raspberry pi by the name determined in the above code “capture.jpg”. Then the image will be sent to the server using the server IP address (`192.168.1.150`) and the php file name (`ProjectCode.php`) which the image will be sent to.

```
try:  
    PIR= GPIO.input(8)  
    if(PIR == 1):  
        print("there's a motion")  
        img = camera.get_image()  
        pygame.image.save(img, FILENAME)  
        url = 'http://192.168.1.150/ProjectCode.php'  
        files = {'image': open('capture.jpg', 'rb')}
```

- Then checking for response from the server that the image is received or else print “Error”.

```
try:  
    pass  
    response= requests.post(url, files=files ,  
timeout=60)  
    print response  
except Exception as e :  
    print 'Error: {}'.format(e)  
    print 'timeout error'
```

- Reading the serial data sent from the Arduino and sending the alert location to the server to store it in the database:

```
read_ser=ser.readline()  
if("Hello" in read_ser):  
    print("hiiiii")  
    url = "http://192.168.1.150/send.php"  
    d = dict()  
    d['message'] = "2"  
    r = requests.post(url, data=d)  
    print (r.text)
```

- Reading the IR sensor and checking if it is triggered or not and reading the serial data are put in an infinite loop (`while (True)`) and then outside the while loop we will stop the camera using this line (`camera.stop()`)

➤ 9.1.3. XAMPP Server:

The server will receive the image from the raspberry pi and process it using php code on Apache web server to detect the barcode if found. Then the server will search in the MySQL server database we created using the barcode detected to get the vehicle owner's information and bank account to collect the toll amount.

➤ 9.1.4. MySQL Database:

We used xampp MySQL server to create the database using the “localhost/phpmyadmin” page.

We have five main tables: cars, users, admins, images and alerts.

- **Cars table** stores the data about all cars in the system as follows:

#	Name	Type	Collation	Attributes	Null	Default
1	car_id	int(10)			No	None
2	car_number	varchar(9)	latin1_swedish_ci		No	None
3	car_barcode	varchar(20)	latin1_swedish_ci		No	None
4	car_status	char(1)	latin1_swedish_ci		No	None
5	user_id	int(10)		UNSIGNED	No	None
6	car_logs	int(10)			No	None
7	last_updated	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP

For example:

car_id	car_number	car_barcode	car_status	user_id	car_logs	last_updated
1	NTB395	89472164841	Y	3	11	2019-05-27 14:12:09
2	SON9432	56317994133	N	1	7	2019-05-27 14:12:26

- **Users table** stores the data about all car users in the system as follows:

#	Name	Type	Collation	Attributes	Null	Default
1	user_id 	int(10)		UNSIGNED	No	None
2	user_name	varchar(30)	latin1_swedish_ci		No	None
3	user_phone_no	varchar(40)	latin1_swedish_ci		No	None
4	user_email 	varchar(100)	latin1_swedish_ci		No	None
5	user_national_id	bigint(30)			No	None
6	user_bank_account	bigint(30)			No	None
7	account_username	varchar(100)	latin1_swedish_ci		No	None
8	account_password	varchar(100)	latin1_swedish_ci		No	None

For example:

user_id	user_name	user_phone_no	user_email	user_national_id	user_bank_account	account_username	account_password
1	Manar Ahmed	01020045617	manarhesham19@hotmail.com	29603161203381	5264024201627551	manar	e8692ea1e0ceb010756edac08f2bf3ac
2	Mohamed Hesham	01008556369	moh_hesham@hotmail.com	29603456154489	1589236498754562	mohamed	81dc9bdb52d04dc20036dbd8313ed055

- **Users and cars tables are connected using foreign key constraints on the user_id column in each table.**

- **Admins table** stores the data about all administrators in the system:

#	Name	Type	Collation	Attributes	Null	Default
1	id 	int(11)			No	None
2	username	varchar(100)	latin1_swedish_ci		No	None
3	email	varchar(100)	latin1_swedish_ci		No	None
4	password	varchar(100)	latin1_swedish_ci		No	None

For example:

id	username	email	password
1	manar	manar19@hotmail.com	463eac4220a4388b6b60ed100d2430d8
2	may	may@yahoo.com	9a4b6f884971dcba4a5172876b335baab

- **Images table** stores the images captured by the camera which contain valid barcodes (the images are stored in the server in htdocs folder in a folder called images) as follows:

#	Name	Type	Collation	Attributes	Null	Default
1	id 	int(20)			No	None
2	image_path	varchar(200)	latin1_swedish_ci		No	None
3	image_name	varchar(100)	latin1_swedish_ci		No	None
4	image_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP
5	car_barcode	varchar(20)	latin1_swedish_ci		No	None

For example:

id	image_path	image_name	image_time	car_barcode
1	images/1556884065.jpg	1556884065	2019-05-03 13:47:45	2345678

- **Alerts table** stores the location, time and duration of alerts detected by the load cell on the road (which will be explained later in this chapter) as follows:

#	Name	Type	Collation	Attributes	Null	Default
1	id 	int(11)			No	None
2	alert_location	varchar(200)	latin1_swedish_ci		No	None
3	alert_time	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP
4	alert_duration_mins	int(20)			No	None

For example:

id	alert_location	alert_time	alert_duration_mins
42	kilo 200	2019-05-21 13:20:16	10
43	kilo 200	2019-05-21 13:22:10	5

➤ 9.1.5. Apache Server:

We used Apache server to run the project php code that processes the image sent to the server and searches the database. We also used it to host our user/admin web interface (which will be mentioned later in this chapter).

- **PHP Code:**

- Connecting to the database to be able to access it:

```
$user = 'root';
$pass = '';
$db = 'project_db';
$conn = mysqli_connect('localhost', $user, $pass, $db);
if(mysqli_connect_errno())
{
    die("Connection failed" .mysqli_connect_error());
}
```

- Receiving the image from the raspberry pi and storing it in a folder called “images” in “htdocs” folder:

```
$info = pathinfo($_FILES['image']['name']);
$ext = $info['extension'];
$newname = round(microtime(true));
$target = 'images/' . $newname . '.' . $ext;
move_uploaded_file($_FILES['image']['tmp_name'], $target);
```

- Scanning the image for barcodes and when found will be stored in a variable named “barcode”:

```
exec('C:\\"Program Files (x86)"\ZBar\bin\zbarimg -q
C:\xampp\htdocs\images\' . $newname . '.' . $ext, $result);
$value = $result[0];
$results = explode(":", $value);
$barcode = $results[1];
```

- Checking if a barcode is read and storing it in a folder called “barcodes”:

```
if (strlen($barcode) > 0) {
    $f = fopen('C:\xampp\htdocs\barcodes\' . $newname .
'.txt', 'w');
    fwrite($f, $barcode);
    fclose($f);
```

- After saving the barcode, we will fetch from the data the last time the log of the car with this barcode was updated:

```
$sql="select last_updated from cars where car_barcode like  
'%{$barcode}%'";  
$res = mysqli_query($conn , $sql);  
if (!$res) {  
    die("Query Failed!");}  
while ($row=$res->fetch_assoc()) {  
    $last_time = $row["last_updated"];}  
mysqli_free_result($res);
```

- Then we get the current time and compare it with the last time the car's log was updated if the difference between the two is bigger than one minute then add one to the car logs and collect toll amount but if it is less than one minutes then nothing will happen because maybe the car was captured twice or more in under one minute:

```
$now = new DateTime();  
$now->modify('-1 minute');  
$time = $now->format('Y-m-d H:i:s');  
if($time > $last_time)  
{
```

- If the if condition is true meaning we will collect the toll amount, we will fetch the user id from the cars table where the barcode equals the detected barcode and then we will fetch his mobile number and bank account to withdraw the toll amount from or if he has a prepaid account the money will be withdrawn automatically.
- Money transfer process will be through a Gateway. Payment gateway is a merchant service provided by an e-commerce application service provider that authorizes credit card or direct payment processing for e-businesses, online retailers, In other words, it is responsible for the quick and secure transfer of money. The payment gateway encrypts the sensitive data that are sent between the driver, the government and the bank. All sensitive data are stored on a secure server of the gateway. This means that the integration can be performed in association with the developers of gateway, they will take care of integrating the code with the API key provided by the gateway in our application.

```
$sql="select user_id from cars where car_barcode like
'%{$barcode}%'";
$res = mysqli_query($conn , $sql);
while ($row=$res->fetch_assoc())
{
    $id = $row["user_id"];
}
mysqli_free_result($res);
$sql="select user_bank_account from users where user_id like
'%{$id}%'";
$res = mysqli_query($conn , $sql);
while ($row=$res->fetch_assoc())
{
    $bank_account = $row["user_bank_account"];
}
mysqli_free_result($res);
```

- Then we will increment the car logs column of the car with the detected barcode by one:

```
$sql="Update cars SET car_logs=car_logs+1 WHERE car_barcode like
'%{$barcode}%'";
$res = mysqli_query($conn , $sql);
```

- Then we will save the image from which the barcode was detected in the images table, we will save the image path, name and barcode:

```
$sql="insert into images ( image_path , image_name , car_barcode)
values ('xampp/htdocs/images' , '.$newname.'.jpeg' , '$barcode'
)';
$res = mysqli_query($conn , $sql);
mysqli_free_result($res);
```

- This way the barcode part is finished in the code. Now we will check if an alert is sent from the Raspberry pi and if found, determine which load cell the alert was sent from and determining its location and then store it in the alerts table in the database with its location, duration, date and time. Each load cell has a specific number to indicate its location. After storing the alert in the database we will send a warning to the police for them to check the reason for this alert and solve it quickly:

```
$location = null;
if (isset($_POST['message'])) {
    $location = $_POST['message'];
    if($location=="2")
    {
        $sql="select alert_time from alerts where
alert_location='kilo 200'";
        $res = mysqli_query($conn , $sql);
        while($row=$res->fetch_assoc())
        {
            $last_time = $row["alert_time"];
        }
        mysqli_free_result($res);
        $now = new DateTime();
        $now->modify('-1 minute');
        $time = $now->format('Y-m-d H:i:s');

        if($time > $last_time)
        {
            $sql="insert into alerts (alert_location) values ('kilo
200')";
            $res = mysqli_query($conn , $sql);
        }
    }
}
```

- Then we will close the connection with the database:

```
mysqli_close($conn);
```

➤ 9.2. Features:

➤ 9.2.1. Traffic Control:

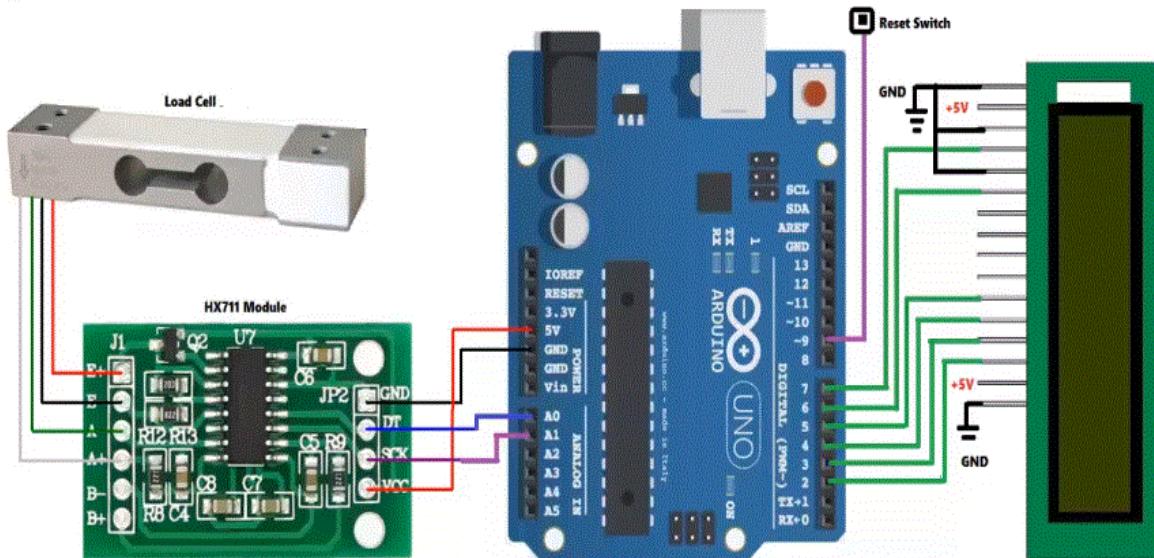
The load cells are used to detect when the flow of traffic is not normal. When the weight on the load cell is more than a specific weight for more than a certain duration of time that means that there's something wrong, it could be that there is an accident in that location or a traffic jam or a car broke down. In all cases a warning will be sent to the police with the location of the problem and it will be shown on all LCDs on the highway before that location to warn drivers to lower their speed according to their distance from the traffic jam to avoid making the traffic jam worse by arriving there fast.

➤ Arduino and load cell Connections:

The electrical signals generated by Load cell is in few millivolts, so they need to be further amplify by some amplifier and hence HX711 Weighing Sensor comes into picture. **HX711 Weighing Sensor Module** has HX711 chip, which is a 24 high precision A/D converter (Analog to digital converter). HX711 has two analog input channels and we can get gain up to 128 by programming these channels. **So HX711 module amplifies the low electric output of Load cells and then this amplified & digitally converted signal is fed into the Arduino to derive the weight.**

Load cell is connected with HX711 Load cell Amplifier using four wires. These four wires are Red, Black, White and Green/Blue. There may be slight variation in colors of wires from module to module.

Connections for this project are easy and schematic is given below. 16x2 LCD pins RS, EN, d4, d5, d6, and d7 are connected with pin number 12, 11, 4, 5, 6 and 7 of Arduino respectively. HX711 Module's DT and SCK pins are directly connected with Arduino's pin A0 and A1. Load cell connections with HX711 module are already explained earlier and also shown in the below circuit diagram.



➤ Arduino Code:

- Including the required libraries for interacting with the LCD and the HX711 chip:

```
#include <HX711.h>
#include <LiquidCrystal.h>
```

- Giving values to the variables used with the HX711 chip scale function and initializing it:

```
#define calibration_factor -7050.0 //This value is obtained using
the SparkFun_HX711_Calibration sketch
#define DOUT 3
#define CLK 2
HX711 scale;
```

- Initializing the pins of the LCD and determining which pins they are connected to in the Arduino:

```
const int rs = 12, en = 11, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

- Initializing the LCD and HX711 scale function:

```
void setup() {
    lcd.begin(16, 2);
    Serial.begin(9600);
    Serial.println("HX711 scale demo");
    scale.begin(DOUT, CLK);
    scale.set_scale(calibration_factor);
    scale.tare();
    Serial.println("\nReadings:");
}
```

- Reading the weight from the Load cell and storing it in a float variable ‘weight’:

```
float weight = abs((scale.get_units()) * -1);
delay(500);
```

- Checking if the measured weight is larger than the specified weight and it remained more than that for the specified duration and then printing the warning on the LCD and if not print the advertisement on the LCD:

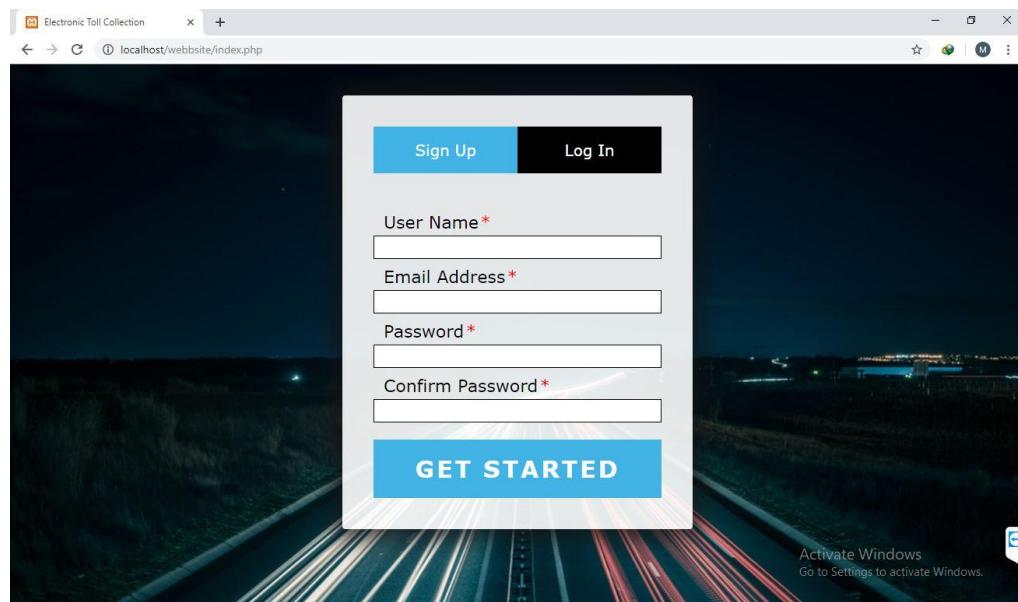
```
if (x >= 1.5) {  
    int y = millis();  
    if (y > 2000) {  
        lcd.clear();  
        lcd.print(" Traffic Jam");  
        Serial.println(data);  
    }  
}  
else {  
    lcd.clear();  
    lcd.print(" Advertisement");  
    Serial.print(" Advertisement");  
}
```

➤ 9.2.2. Web User/Admin Interface (Website):

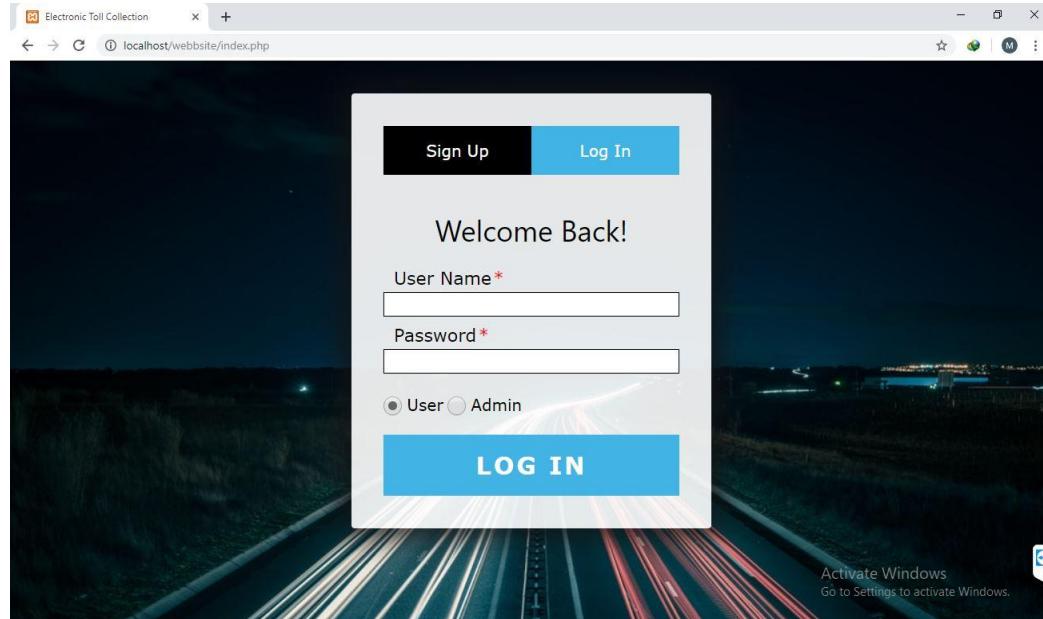
We created a web interface to make it easier for the user to keep track of his data and how many times he paid the toll and the date of the last time he passed the gate, and to make it easier for the administrators to access the database and search and update the data of users and cars in the system.

➤ User Interface:

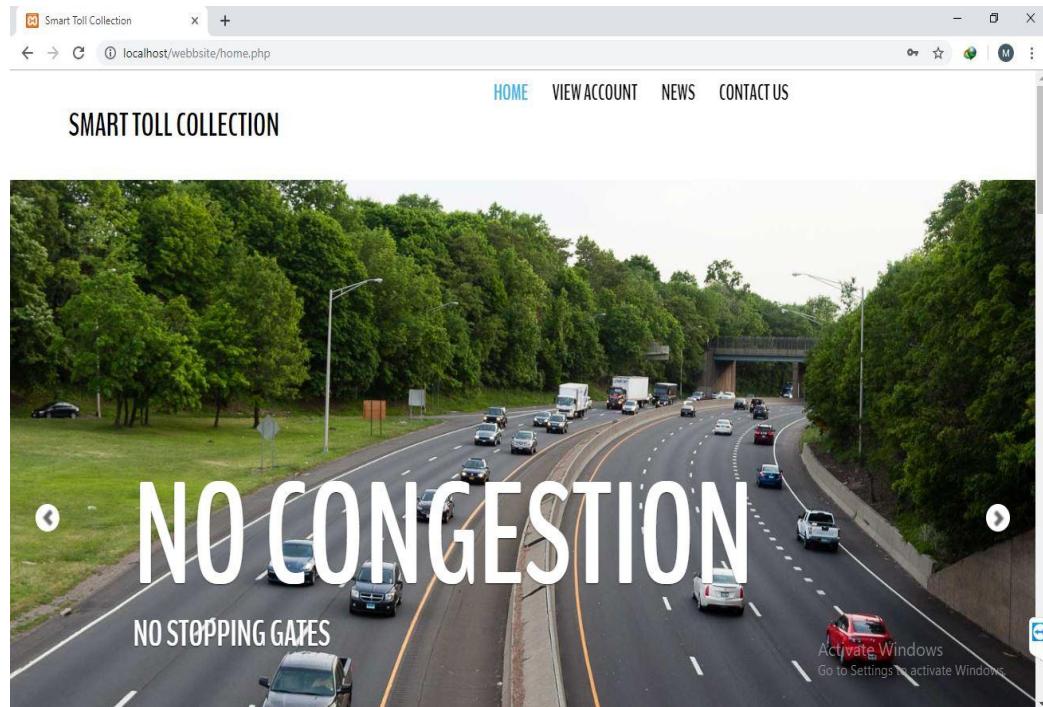
Each user has an account with the email which is registered under his personal data, to access the website he must sign up the very first time with the same email and give it a password and then they can login anytime after that.



Smart Toll Collection on Highways



Login Form For User



Home Page

The screenshot shows a web browser window titled "Smart Toll Collection" with the URL "localhost/websit/home.php". The page has a blue header with the text "SMART TOLL COLLECTION" and navigation links for "HOME", "VIEW ACCOUNT", "NEWS", and "CONTACT US". Below the header, the main content area has a large white title "ACCOUNT INFORMATION". Underneath, it displays user information:
User Name: manar
Full Name: Manar Ahmed
Phone Number: 01020045617
Email: manarhesham19@hotmail.com
National Id: 29603161203381
Bank Account Number: 5264024201627551
A table below shows toll history:

Cars Owned	Last Toll Date	Number of Times
SON9432	2019-05-27 14:12:26	1

Displaying The user information

The screenshot shows a web browser window titled "Smart Toll Collection" with the URL "localhost/websit/home.php". The page has a blue header with the text "SMART TOLL COLLECTION" and navigation links for "HOME", "VIEW ACCOUNT", "NEWS", and "CONTACT US". Below the header, the main content area has a large white title "LATEST NEWS". Underneath, it displays recent news items:
Recent Egypt Highway News
Three news cards are shown:

- Egypt's Top 10 Roads: Exploring the National Roads Project
- 2019 Accomplishment: National Road Project puts Egypt 75 globally in road quality
- Egypt's new concrete highway connecting Cairo with Suez

A small "Activate Windows" watermark is visible in the bottom right corner.

Latest Highways and Roads News

Smart Toll Collection

localhost/website/home.php

HOME VIEWACCOUNT NEWS CONTACT US

SMART TOLL COLLECTION

CONTACT US

CONTACT INFORMATION

Monday - Friday: 9:30 AM to 6:30 PM

Address: 123 Some Street, California, US, CP 123

Phone: (01) 999-1235

Fax: (01) 999-1234

Email: info@domain.com

LEAVE US A MESSAGE

Name

Email

Phone No:

Message

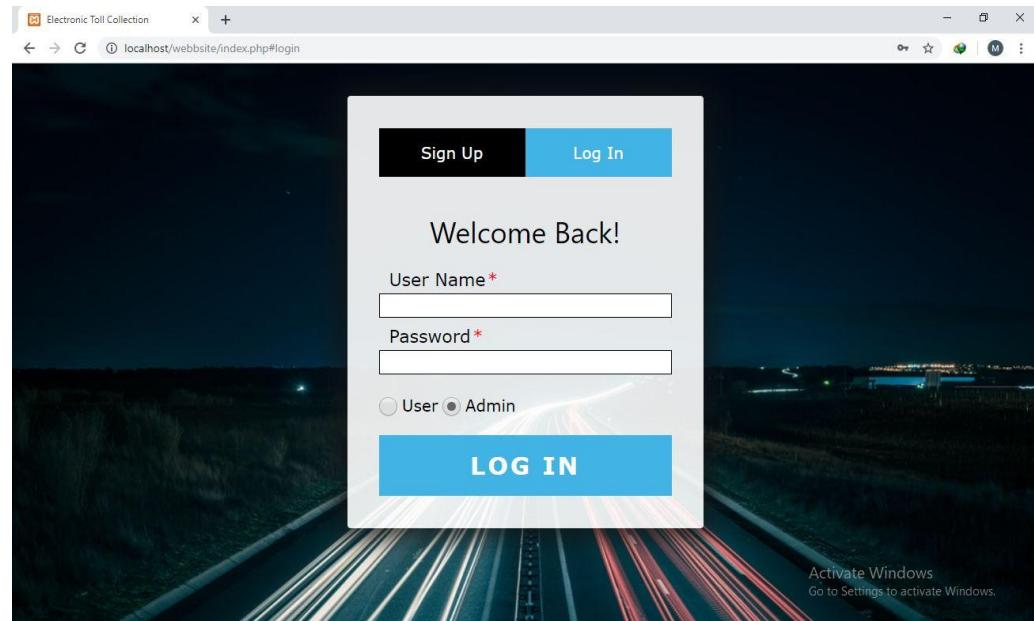
SUBMIT

Activate Windows Go to Settings to activate Windows.

Contact Form to Send Complaints or Suggestions

➤ Admin Interface:

- Each admin has an account to login with to be able to access the database and see all table contents and be able to change data and run queries and update data and search the data using user names and car plate numbers.



Smart Toll Collection on Highways

- Admins can't sign up they are only added to the database by another admin using the admin web interface or by database developers who deal directly with the database.
- Admins can add users data and cars data to the database, they can delete users and cars if needed and they can update their data also.
- Admins can search tables and see all data like this:

The screenshot shows a web browser window titled "Smart Toll Collection". The URL is "localhost/websit/homeadmin.php#". The page has a blue header with the title "SMART TOLL COLLECTION" and a welcome message "WELCOME, MAY". On the left, there is a sidebar with navigation links: "Cars and Users", "Cars", "Users", "Alerts", "Admins", and "Images". The main content area is titled "CARS AND USERS TABLE" and contains a table with the following data:

Car Number	Car Barcode	Owner Name	Owner Number	Owner Email	Owner Bank Account	Owner National ID
NTB395	89472164841	May Abdelsalam	01287855119	mayabdelrahman22@gmail.com	8949236498754562	29605489600158
SON9432	56317994133	Manar Ahmed	01020045617	manarhesham19@hotmail.com	5264024201627551	29603161203381
DQT3742	91412287456	Mohamed Hesham	01008556369	moh_hesham@hotmail.com	1589236498754562	29603456154489
DSR5821	42368123365	May Abdelsalam	01287855119	mayabdelrahman22@gmail.com	8949236498754562	29605489600158
HEP213	28479328941	Esraa Shaban	01097704142	esraa@yahoo.com	1545697789456992	29603456154485

At the bottom, there is a copyright notice "smarttollcollection.com All right reserved", social media icons for Facebook, Twitter, and Google+, and an "Activate Windows" message.

The screenshot shows a web browser window titled "Smart Toll Collection". The URL is "localhost/websit/homeadmin.php#". The page has a blue header with the title "SMART TOLL COLLECTION" and a welcome message "WELCOME, MAY". On the left, there is a sidebar with navigation links: "Cars and Users", "Cars", "Users", "Alerts", "Admins", and "Images". The main content area is titled "CARS TABLE" and contains a table with the following data:

Car Number	Cars ID	Car Barcode	Car Status	Car Owner ID	Car Logs	Last Log
NTB395	1	89472164841	Pays	3	11	2019-06-01 14:59:31
SON9432	2	56317994133	Exempted	1	1	2019-06-01 14:59:21
DQT3742	3	91412287456	Exempted	2	0	2019-06-01 14:59:33
DSR5821	4	42368123365	Pays	3	5	2019-06-01 14:59:39
HEP213	5	28479328941	Pays	4	0	2019-06-01 14:59:46

At the bottom, there is a copyright notice "smarttollcollection.com All right reserved", social media icons for Facebook, Twitter, and Google+, and an "Activate Windows" message.

Smart Toll Collection on Highways

Smart Toll Collection - localhost/websit/homeadmin.php#

SMART TOLL COLLECTION

WELCOME, MAY

- Cars and Users
- Cars
- Users
- Alerts
- Admins
- Images

USERS TABLE

User Name	User ID	User Number	User Email	User Bank Account	User National ID	Account Name	Account Password
Manar Ahmed	1	01020045617	manarhesham19@hotmail.com	5264024201627551	29603161203381	manar	*****
Mohamed Hesham	2	01008556369	moh_hesham@hotmail.com	1589236498754562	29603456154489	mohamed	*****
May Abdelsalam	3	01287855119	mayabdelaslam22@gmail.com	8949236498754562	29605489600158	mayy	*****
Esraa Shaban	4	01097704142	esraa@yahoo.com	1545697789456992	29603456154485	esraa	*****

Smart Toll Collection - localhost/websit/homeadmin.php#

SMART TOLL COLLECTION

WELCOME, MAY

- Cars and Users
- Cars
- Users
- Alerts
- Admins
- Images

ALERTS TABLE

Alert Location	Alert ID	Alert Date and Time	Alert Duration in Minutes
kilo 200	1	2019-06-02 01:44:22	10
kilo 30	2	2019-06-02 01:45:34	5
kilo 70	3	2019-06-02 01:47:50	2
kilo 100	4	2019-06-02 01:47:54	30
kilo 130	5	2019-06-02 01:48:03	1

smarttollcollection.com All rights reserved

Smart Toll Collection - localhost/websit/homeadmin.php#

SMART TOLL COLLECTION

WELCOME, MAY

- Cars and Users
- Cars
- Users
- Alerts
- Admins
- Images

ADMINS TABLE

Admin User Name	Admin ID	Admin Email	Password
manar	1	manar19@hotmail.com	*****
may	2	may@yahoo.com	*****
rana	3	rana21@hotmail.com	*****
areej	4	areej@hotmail.com	*****

smarttollcollection.com All rights reserved

➤ **9.3. Our project prototype (Maquette):**



Chapter ten

Conclusion and Future Work

➤ 10.1. Conclusion:

In Conclusion, we choose this project because the number of vehicles plying the highways keeps growing at a steady pace, leading to high maintenance costs and the inefficiencies of traditional toll collection systems tend to be unbearable for users because they often result in long vehicle queues and traffic congestion.

- Smart toll collection systems seek to address this problem by reducing the waiting time on the toll highways. Significant reduction in the delay period will minimize unproductive fuel consumption, air pollution, unnecessary road congestion, and increase road safety.
- The implementation of this system will improve the toll collection efficiency in terms of speed and flexibility. Overall, the contribution of this work extends to the domain of Intelligent Transportation System (ITS). In this era of rapid technology development, conventional systems are being replaced with automated systems.
- Automation minimizes human interference, ensures reduction in time and energy, and increases efficiency. Therefore, automating this process is an indispensable task able to bring great benefits, and will help to increase the standard of living, while indirectly contributing to the goal of making cities smarter.
- By this we don't have to carry a handsome amount of cash with us relates to security as well. This system does not require any manual operation of toll barriers and collection of toll amounts, it is completely automated toll collection system.

- We made a prototype for ETC in smart and connected communities. The main components of the intelligent system architecture are the Laser sensor nodes, a Raspberry Pi, a Camera and web applications.
- The Laser Sensor and Camera enable vehicle detection and classification, the Raspberry pi establishes a communication link to the back-end of the system. The central database and the web server are hosted in the Cloud while a web application is used for electronic transactions, subscription renewal, notification of toll payments, and for tracking toll payment history. In addition, a web dash board is provided for efficient toll administration.
- When the Laser sensor is triggered the camera captures a picture of the car and sends it to the Raspberry pi, then the pi will send the picture to the server where it will be processed and if a barcode is detected it will search in the database for the car owner account and the toll will be collected, also the picture will be stored in the database for future use. If the barcode is not found in the database the server will send a warning to the police with the picture of the car. There is a web interface so that the administrators can monitor and control the database and have access to the toll history of car users and collect data about them. There is also a user interface where the user can track his toll history and car information.
- The load cells are used to detect when the flow of traffic is not normal. When the weight on the load cell is more than a specific weight for more than a certain duration of time that means that there's something wrong, it could be that there is an accident in that location or a traffic jam or a car broke down. In all cases a warning will be sent to the police with the location of the problem and it will be shown on all LCDs on the highway before that location to warn drivers to lower their speed according to their distance from the traffic jam to avoid making the traffic jam worse by arriving there fast.
- We used Raspberry Pi 3 model B+, XAMPP web server and also for the database. We used PHP as a server scripting language, SQL as a language for handling the database and Python as a programming language for the Raspberry Pi. We used Arduino Uno to control the load cells and LCDs and to communicate with the Raspberry Pi.

➤ 10.2. Future Work:

This new system will require some work to be done before it is installed on highways.

- First of all, each car number will have to be linked to a unique barcode which will be detected by the system because performing image processing on barcodes is easier, faster and more accurate than performing image processing on plate number especially if the plate number is in Arabic letters and numbers, that is why we choose barcodes rather than trying to process the ordinary plate numbers.
- We will have to collect data about all cars and their owners and store them in the database in their respective tables. Each car owner will have to submit all the required information about themselves when they get their car license or when they renew their license. They have to submit their full name, a working mobile number, a valid email address, their national Id and a working bank account with a specific deposit amount which the toll will be collected from, in return they will be given a user account name and a password which will be used to access their account to monitor his toll collection, browse the latest highway news and submit any complaints or suggestions.
- We have to take all old plate numbers and print their unique barcodes on them and that will be a little expensive but in return it will make great profit and make life easier and faster for drivers and make it easier to monitor highways and store data about them.
- For the hardware part of the project all old gates have to be replaced with new ones which have no booths for the cars to stop and pay the gate will just be over the road with a camera facing the incoming cars and another one to capture the back of the car when it passes the gate. A laser sensor will be installed before each gate to detect when a car passes so that the camera captures the picture so that the camera is not active all the time to save power and storage space from unimportant pictures.

- Each gate will have a control unit next to it with all the required hardware and controllers and it is connected to the server, the internet, the nearest police station, hospital and fire department to send warnings in case of accidents or fires or detecting a car with is reported to be stolen or related to a crime.
- All over the road load cells will be installed every specific distance and next to each load cell we will have an LCD for advertisement in normal conditions but in case the load cell detects a traffic jam for whatever reason it will send a warning to the control unit, which will know its location and print a warning on all LCDs before that location with the location of the traffic jam and advices drivers to slow down their speed respectively to the distance between them and the traffic jam location so they don't reach the location fast and make the traffic jam worse, if they take a little bit more time to reach there they won't stop for a long time because the authorities will solve the problem quickly because they are notified immediately. No other changes will be made to the highways system.
- In the Future we will develop a mobile application for users to make it easier to access their account and there will be an option an e-wallet to be connected to the mobile application for car owners that don't want to give their bank account

➤ **References:**

1. <https://brainschildblog.wordpress.com/2016/05/30/raspberry-pi-3-is-out-now-specs-benchmarks-more/>
2. <https://www.raspberrypi.org/documentation/faqs/#introduction>
3. <https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>
4. <https://www.raspberrypi.org/documentation/remote-access/vnc/>
5. <https://www.raspberrypi.org/documentation/remote-access/vnc/>
6. <https://www.raspberrypi.org/documentation/remote-access/ssh/windows.md>
7. <https://www.arduino.cc/en/guide/environment>
8. <https://www.arduino.cc/en/Guide/Introduction>
9. <https://www.techwalla.com/articles/how-to-sell-server-space>
10. <https://whatis.techtarget.com/definition/server>
11. <https://www.computerhope.com/jargon/w/webserv.htm>
12. https://www.tutorialspoint.com/internet_technologies/pdf/web_servers.pdf
13. <https://www.dedoimedo.com/computers/www.dedoimedo.com-apache-web-server-lm.pdf>
14. https://en.wikipedia.org/wiki/Database#Terminology_and_overview
15. G563 Quantitative Paleontology Department of Geological Sciences | P. David Polly 1 Introduction to MySQL and phpMyAdmin
16. <https://pureinfotech.com/install-xampp-windows-10/>
17. <https://en.wikipedia.org/wiki/XAMPP#Etymology>
18. INSTALLING, CONFIGURING, AND DEVELOPING WITH XAMPP by Dalibor D. Dvorski, March 2007 Skills Canada – Ontario
19. <https://www.rudrra.com/products/load-cell.html>
20. <https://www.instructables.com/id/How-to-Interface-HX711-Balance-Module-With-Load-Ce/>

21. <https://circuits4you.com/2016/11/25/hx711-arduino-load-cell/>
22. The Basics of Bar Coding
23. BARCODE EDUCATIONAL GUIDE , IDAutomation.com, Inc. | IDAutomation.com, Inc. and Benjamin Johnson, Technical Support & Development
24. <https://www.cloudways.com/blog/how-to-start-php-programming/>
25. <https://en.wikipedia.org/wiki/PHP#Use>
26. <https://www.w3schools.com/php/>
27. https://www.w3schools.com/php/php_mysql_connect.asp
28. <https://www.raspberrypi.org/documentation/usage/python/>
29. [https://en.wikipedia.org/wiki/Python_\(programming_language\)#Features_and_philosophy](https://en.wikipedia.org/wiki/Python_(programming_language)#Features_and_philosophy)