1. INTRODUCTION

An efficient and accurate embedded access control system based on face recognition is very important for wide range of commercial and security application. Many countries are gradually adopting smart home security control system. The most important part of any home security system is accurately identifying visitor who enter and leave through the door. An entrance guard can be managed using passwords, RFID sensors, finger prints and face recognition methods. Face recognition is probably the most natural way to perform biometric authentication between human beings. Additionally, it is the second most popular biometric trait, after fingerprints.

Only few researchers have implemented the face recognition techniques in an embedded system for real time applications, such as a wireless door access control system. Most of the system was implementing a principle component analysis (PCA) algorithm for face recognition on hardware platform for its simplicity and dimensionality reduction. Wireless technologies like radio frequency identification (RFID), ultra wide band (UWB), and ZigBee etc. are used in access control systems.

The proposed system is a wireless access control system designed and developed for smart home environment. The paper proposes a Raspberry pi based door access control and home security system through webpage with WiFi based technology. The system identifies the visitor's presence, capture and transfers the image through email and/or an alert SMS via GSM network

automatically to home owner to recognize the visitors. The system capability to provide access through internet, where subject of received email is read by the developed algorithm fed into Raspberry pi and system responds to the corresponding instruction with high security. The user can directly login and interact with the embedded device in real time without the need to maintain an additional server. It has a variety of features such as energy efficient, intelligence, low cost, portability and high performance.

2. BLOCK DIAGRAM

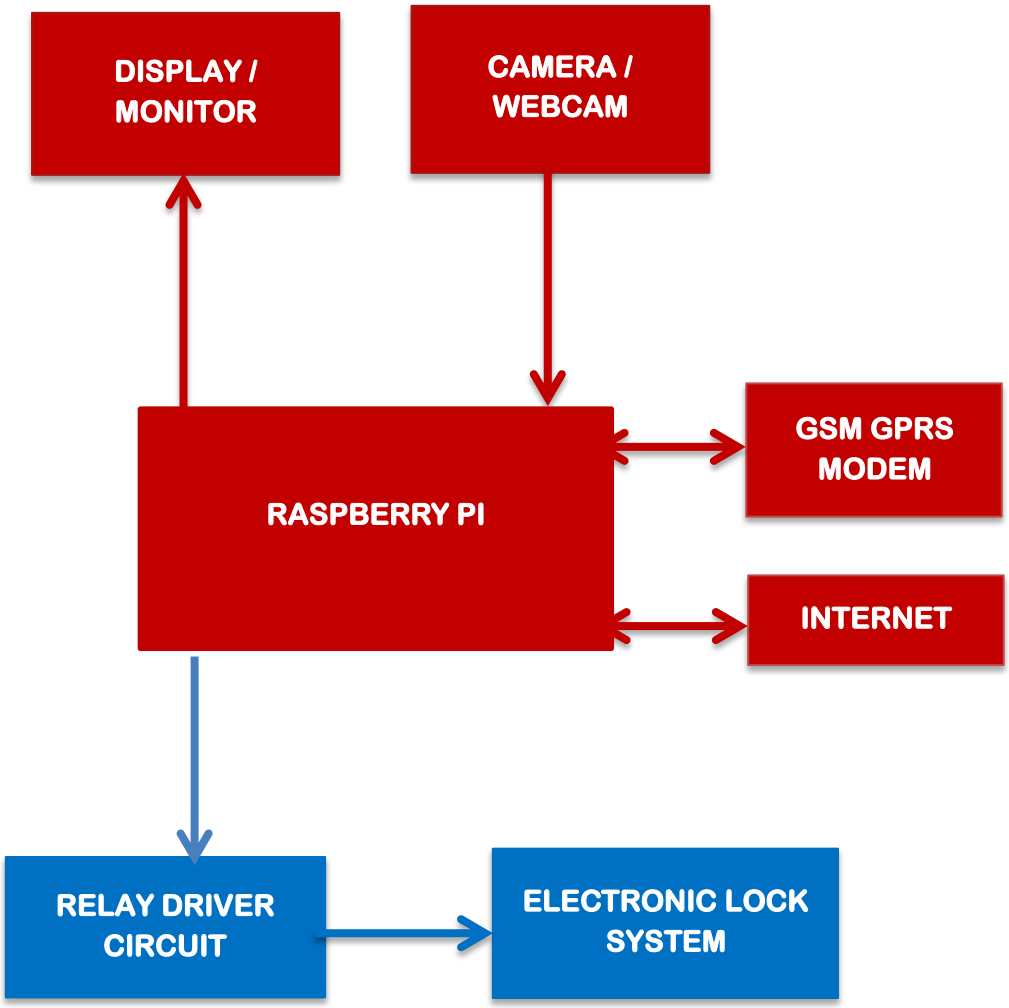


Fig: 2: - Block diagram

HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE

- Raspberry Pi
- ARM 1176 Processor
- Camera
- GSM Modem
- Relay

SOFTWARE

- Debian
- Raspbian
- Python
- Open Source Computer Vision Library

3. HARDWARE SPECIFICATIONS

3.1 RASPBERRY PI

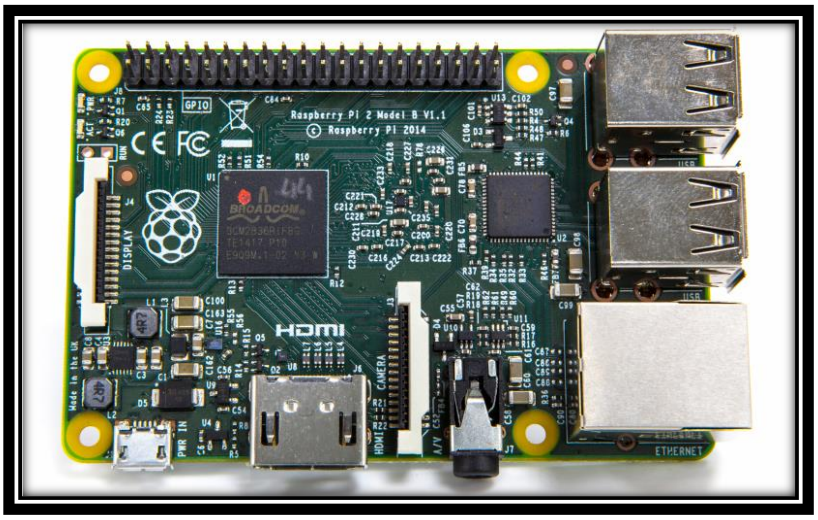


Fig:-Raspberry Pi

- A 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- Video Core IV 3D graphics core

The Raspberry Pi is a series of credit card-sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and developing countries.

All models feature a Broadcom system on a chip (SoC), which includes an ARM compatible central processing unit (CPU) and an on chip graphics processing unit (GPU, a Video Core IV). CPU speed ranges from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAM. Secure Digital SD cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes.

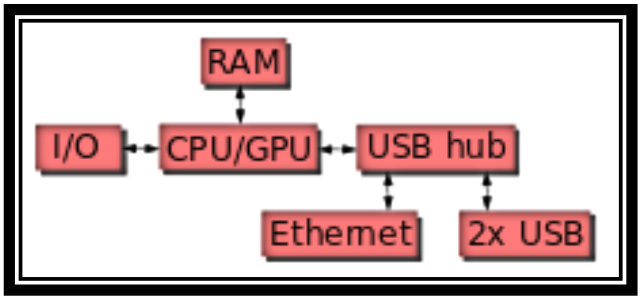
Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phono jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I²C. Some models have an 8P8C Ethernet port and the Pi 3 has on board Wi-Fi 802.11n and Bluetooth.

Hardware

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.

This block diagram depicts models *A*, *B*, *A*+, and *B*+. Model *A, A*+, and *Zero* lack the Ethernet and USB hub components. The Ethernet adapter is connected to an additional USB port. In model *A, A*+ and *Zero* the USB port is

connected directly to the system on a chip (SoC). On model *B+* and later models the USB/Ethernet chip contains a five-point USB hub, of which four ports are available, while model *B,* and only provides two. On the model *Zero*, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.



Processor

The Raspberry Pi model 2 uses a 32-bit 900 MHz quad-core ARM Cortex-A7 processor. The SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in older smartphones (such as iPhone, 3G, 3GS).

The Raspberry Pi is based on the Broadcom BCM2835 SoC, which includes an 700 MHz ARM1176JZF-S processor, Video Core IV graphics processing unit (GPU), and RAM. It has a Level 1 cache of 16 KB and a Level 2 cache of 128 KB. The Level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.The Raspberry Pi 2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache. The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

Operating systems

Various operating systems can be installed on the Pi models through SD cards, most use a MicroSD slot located on the bottom of the board. The Raspberry Pi primarily uses Linux-kernel-based operating systems.

The ARM11 chip at the heart of the Pi (first generation models) is based on version 6 of the ARM which means that Ubuntu, and several other popular versions of Linux, do not run on this model. The primary supported operating system is Raspbian, although it is compatible with many others. The Raspberry Pi 2 has much wider support, including the current release of Ubuntu and Windows 10 IoT Core. No version of the Pi can run traditional Microsoft Windows.

The install manager for the Raspberry Pi is NOOBS. The operating systems included with NOOBS are:

- Arch Linux ARM
- OpenELEC
- OSMC (formerly Raspbmc) and the Kodi open source digital media cente.

- Puppy Linux
- RISC OS – is the operating system of the first ARM-based computer.

BLOCK DIAGRAM OF RASPBERRY PI 3

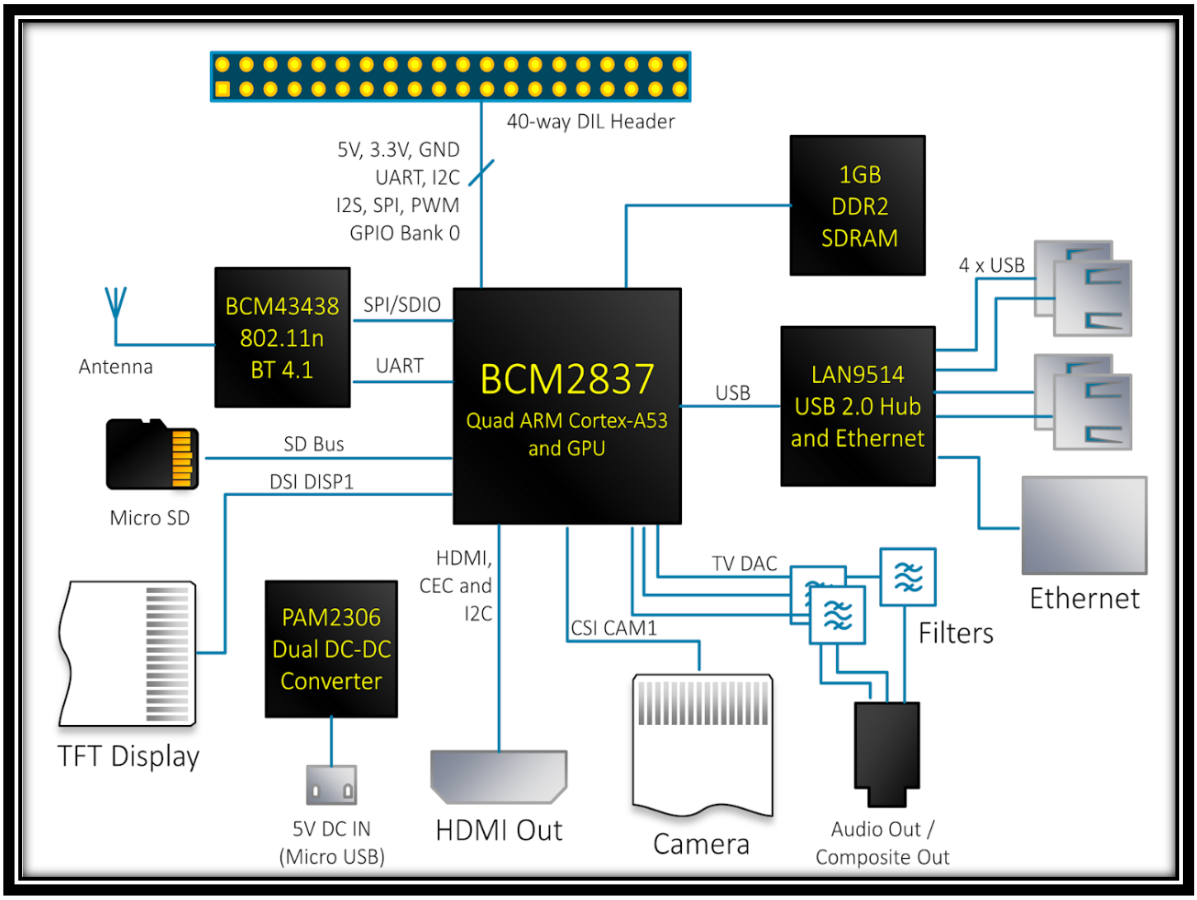


Fig: Block Diagram of Pi 3

3.2 ARM1176 PROCESSOR (SOC)

The ARM1176 applications processors deployed broadly in devices ranging from smart phones to digital TV's to eReaders, delivering media and browser performance, a secure computing environment, and performance up to 1GHz in low cost designs. The ARM1176JZ-S processor features ARM TrustZone technology for secure applications and ARM Jazelle technology for efficient embedded Java execution. Optional tightly coupled memories simplify ARM9™ processor migration and real-time design, while AMBA 3 AXI interfaces improve memory bus performance. DVFS support enables power optimization below the best-in-class nominal static and dynamic power of the ARM11 processor architecture.

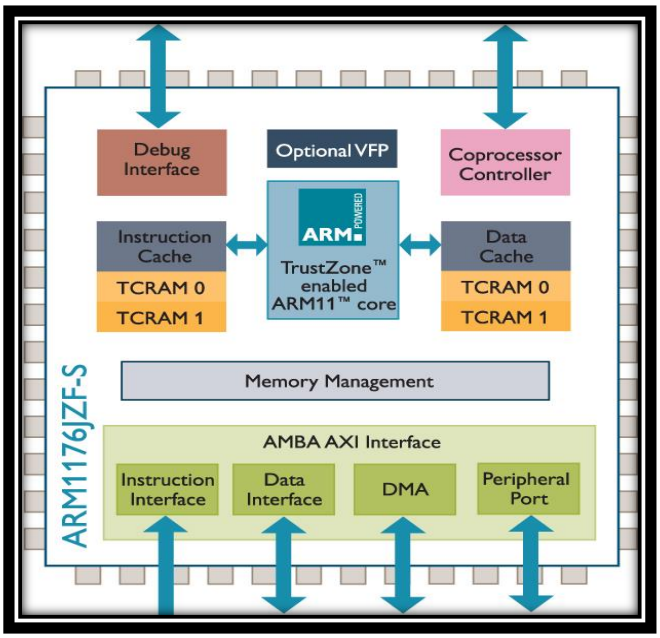


Fig: - ARM1176 Processor

Why ARM1176?

- Low risk and fast time to market
- High performance in low-cost designs
- Compelling end-user experience

ARM1176 Key Feature

- Performance to enable excellent end-user experience
- Product maturity enables rapid time to market and low risk
- Low Power Leadership

3.3. CAMERA

A **webcam** is a video camera that feeds or streams its image in real time to or through a computer to computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks via systems such as the internet, and email as an attachment. When sent to a remote location, the video stream may be saved, viewed or on sent there. Unlike an IP camera (which connects using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, or similar cable, or built into computer hardware, such as laptops. The term "webcam" (a clipped compound) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet.

Fig: - Webcam

Characteristics

Webcams are known for their low manufacturing cost and their high flexibility, making them the lowest-cost form of video telephony. Despite the low cost, the resolution offered at present (2015) is rather impressive, with low-end webcams offering resolutions of 320×240, medium webcams offering 640×480 resolution, and high-end webcams offering 1280×720 (aka 720p) or even 1920×1080 (aka 1080p) resolution. They have also become a source of security and privacy issues, as some built-in webcams can be remotely activated by spyware.

Uses

The most popular use of webcams is the establishment of video links, permitting computers to act as bs or videoconference stations. Other popular uses include security surveillance, computer vision, video broadcasting, and for recording social videos. The video streams provided by webcams can be used for a number of purposes, each using appropriate software.

3.4. GSM MODEM

A GSM modem is a specialized modem which accepts SIM card and operates over a subscription to a mobile operator, just like a mobile phone. A GSM modem has a serial connection. A GSM modem exposes an interface that allows applications such as to send and receive messages over the modem interface. A GSM modem is a wireless modem that works with wireless network. Computers use at commands (Attention commands) to control modems. The number of SMS that can be processed by a GSM per minute is about 6-10. A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, or it may be a mobile phone that provides GSM modem capabilities. A GSM modem exposes an interface that allows applications such as Now SMS to send and receive messages over the modem interface. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone. GSM modems can be a quick and efficient way to get started with SMS, because a special subscription to an SMS service provider is not required. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone.

GSM MODEM Features

- Quad Band GSM/GPRS
- 850/900/1800/1900 Mhz
- GPRS multi-slot class 10/8
- GPRS Mobile station class B
- Compliant to GSM Phase 2/2+
- Class 4 (2W@850/900Mhz)
- Class 1(1W@1800/1900Mhz)
- Control via AT commands (GSM 07.07, 07.05 and enhanced AT commands)
- Operation Temperature (-20 deg C to +55 deg C)

Due to some compatibility issues that can exist with mobile phones, using a dedicated GSM modem is usually preferable to a GSM mobile phone. This is more of an issue with MMS messaging, where if you wish to be able to receive inbound MMS messages with the gateway, the modem interface on most GSM phones will only allow you to send MMS messages. This is because the mobile phone automatically processes received MMS message notifications without forwarding them via the modem interface.

Communication with the GSM modem is by using AT commands. To perform these tasks, a GSM modem must support an "extended AT command set" for sending/receiving SMS messages.

Commands for Reading and Sending SMS is as follows

1. **Set SMS mode**

   Command : AT+CMGF=1

2. **Send SMS**

   Command : AT+CMGS=<”number”><Crriagereturn>(new line)(>)<matter> <ctrl+z>

3. **Read SMS**

   Command : AT+CMGR=<location>

SIM300/SIM345 {GSM MODEM}

The SIM305/345 is a complete Dual-band/Quad-band GSM/GPRS solution in a compact plug-in module. Featuring an industry-standard interface, the SIM305/345 delivers GSM/GPRS 50/900/1800/1900MHz performance for Voice, SMS, Data, and Fax in a small form factor and with low power consumption. The leading features of SIM305/345 make it ideal for M2M applications, such as AMR, POS, Security, AVL etc.

General features

- Dual-Band GSM/GPRS 900/1800MHz
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Dimension: 58mm x 32mmx 3.9 mm
- Weight: <12g
- SIM application toolkit
- Supply voltage range 3.4 ... 4.5 V
- Low power consumption
- Normal operation temperature: -30°C to +80°C

- Restricted operation temperature: -40°C to -30°C and +80°C to +85°C
- Storage temperature: -45°C to +90°C

Specifications for data:

- GPRS class 10: max. 85.6 Kbps (downlink)
- PBCCH support
- Coding schemes CS 1, 2, 3, 4
- CSD up to 14.4 kbps
- USSD
- Non transparent mode

AT COMMANDS BASICS

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. Many of the commands that are used to control wired dial-up modems, such as ATD (Dial), ATA (Answer), ATH (Hook control) and ATO (Return to online data state), are also supported by GSM/GPRS modems and mobile phones. Besides this common AT command set, GSM/GPRS modems and mobile phones support an AT command set that is specific to the GSM technology, which includes SMS-related commands like AT+CMGS (Send SMS message), AT+CMSS (Send SMS message from storage), AT+CMGL (List SMS messages) and AT+CMGR (Read SMS messages).

Note that the starting "AT" is the prefix that informs the modem about

the start of a command line. It is not part of the AT command name. For example, D is the actual AT command name in ATD and +CMGS is the actual AT command name in AT+CMGS. However, some books and web sites use them interchangeably as the name of an AT command.

Here are some of the tasks that can be done using AT commands with a GSM/GPRS modem or mobile phone:

- Get basic information about the mobile phone or GSM/GPRS modem. For example, name of manufacturer (AT+CGMI), model number (AT+CGMM), IMEI number (International Mobile Equipment Identity) (AT+CGSN) and software version (AT+CGMR).
- Get basic information about the subscriber. For example, MSISDN (AT+CNUM) and IMSI number (International Mobile Subscriber Identity) (AT+CIMI).
- Get the current status of the mobile phone or GSM/GPRS modem. For example, mobile phone activity status (AT+CPAS), mobile network registration status (AT+CREG), radio signal strength (AT+CSQ), battery charge level and battery charging status (AT+CBC).
- Establish a data connection or voice connection to a remote modem (ATD, ATA, etc).
- Send and receive fax (ATD, ATA, AT+F*).
- Send (AT+CMGS, AT+CMSS), read (AT+CMGR, AT+CMGL), write (AT+CMGW) or delete (AT+CMGD) SMS messages and obtain

notifications of newly received SMS messages (AT+CNMI).

- Read (AT+CPBR), write (AT+CPBW) or search (AT+CPBF) phonebook entries.

Note that mobile phone manufacturers usually do not implement all AT commands, command parameters and parameter values in their mobile phones. Also, the behaviour of the implemented AT commands may be different from that defined in the standard. In general, GSM/GPRS modems designed for wireless applications have better support of AT commands than ordinary mobile phones.

In addition, some AT commands require the support of mobile network operators. For example, SMS over GPRS can be enabled on some GPRS mobile phones and GPRS modems with the +CGSMS command (command name in text: Select Service for MO SMS Messages). But if the mobile network operator does not support the transmission of SMS over GPRS, you cannot use this feature.

3.5. RELAY (ELECTROMECHANICAL SWITCH)

A **relay** is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and retransmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

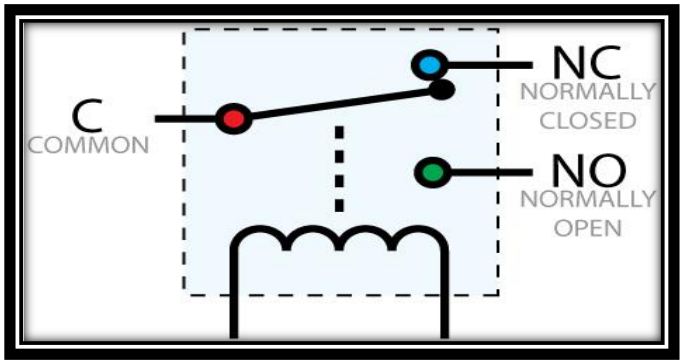


Fig: 3.5:-circuit of relay

BASIC DESIGN AND OPERATION

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core, an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two contacts in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay

pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB. When an electric current is passed through the coil it generates a magnetic field that activates the armature and the consequent movement of the movable contact either makes or breaks (depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduces arcing.

When the coil is energized with direct current, a diode is often placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to semiconductor circuit components. Such diodes were not widely used before the application of transistors as relay drivers, but soon became ubiquitous as early germanium transistors were easily destroyed by this surge. Some automotive relays include a diode inside the relay case. If the relay is driving a large, or especially a reactive load, there may be a similar problem of surge currents

around the relay output contacts. In this case a snubber circuit (a capacitor and resistor in series) across the contacts may absorb the surge. Suitably rated capacitors and the associated resistor are sold as a single packaged component for this commonplace use.

TYPES

- **Latching relay**

A ***latching relay*** (also called "impulse", "keep", or "stay" relays) maintains either contact position indefinitely without power applied to the coil. The advantage is that one coil consumes power only for an instant while the relay is being switched, and the relay contacts retain this setting across a power outage. A latching relay allows remote control of building lighting without the hum that may be produced from a continuously (AC) energized coil.

A ***stepping relay*** is a specialized kind of multi-way latching relay designed for early automatic telephone exchanges. An earth leakage circuit breaker includes a specialized latching relay. Very early computers often stored bits in a magnetically latching relay, such as ferreed or the later remreed in the 1ESS switch. Some early computers used ordinary relays as a kind of latch—they store bits in ordinary wire spring relays or reed relays by feeding an output wire back as an input, resulting in a feedback loop or sequential circuit. Such an electrically latching relay requires continuous power to maintain state, unlike magnetically latching relays or mechanically racheting relays.In computer memories, latching relays and other relays were replaced by delay line memory,

which in turn was replaced by a series of ever-faster and ever-smaller memory technologies.

- **Reed relay**

A ***reed relay*** is a reed switch enclosed in a solenoid. The switch has a set of contacts inside an evacuated or inert gas-filled glass tube which protects the contacts against atmospheric corrosion; the contacts are made of magnetic material that makes them move under the influence of the field of the enclosing solenoid or an external magnet.

- **Mercury-wetted relay**

A ***mercury-wetted reed relay*** is a form of reed relay in which the contacts are wetted with mercury. Such relays are used to switch low-voltage signals (one volt or less) where the mercury reduces the contact resistance and associated voltage drop, for low-current signals where surface contamination may make for a poor contact, or for high-speed applications where the mercury eliminates contact bounce. Mercury wetted relays are position-sensitive and must be mounted vertically to work properly. Because of the toxicity and expense of liquid mercury, these relays are now rarely used.

- **Mercury relay**

A ***mercury relay*** is a relay that uses mercury as the switching element. They are used where contact erosion would be a problem for conventional relay contacts. Owing to environmental considerations about significant amount of mercury used and modern alternatives, they are now comparatively uncommon.

- **Polarized relay**

A ***polarized relay*** places the armature between the poles of a permanent magnet to increase sensitivity. Polarized relays were used in middle 20th Century telephone exchanges to detect faint pulses and correct telegraphic distortion. The poles were on screws, so a technician could first adjust them for maximum sensitivity and then apply a bias spring to set the critical current that would operate the relay.

- **Machine tool relay**

A ***machine tool relay*** is a type standardized for industrial control of machine tools, transfer machines, and other sequential control. They are characterized by a large number of contacts (sometimes extendable in the field) which are easily converted from normally open to normally closed status, easily replaceable coils, and a form factor that allows compactly installing many relays in a control panel. Although such relays once were the backbone of automation in such industries as automobile assembly, the programmable logic controller (PLC) mostly displaced the machine tool relay from sequential control applications.

A relay allows circuits to be switched by electrical equipment: for example, a timer circuit with a relay could switch power at a preset time. For many years relays were the standard method of controlling industrial electronic systems. A number of relays could be used together to carry out complex functions (relay logic). The principle of relay logic is based on relays which

energize and de-energize associated contacts. Relay logic is the predecessor of ladder logic, which is commonly used in programmable logic controller.

- **Coaxial relay**

Where radio transmitters and receivers share one antenna, often a ***coaxial relay*** is used as a TR (transmit-receive) relay, which switches the antenna from the receiver to the transmitter. This protects the receiver from the high power of the transmitter. Such relays are often used in transceivers which combine transmitter and receiver in one unit. The relay contacts are designed not to reflect any radio frequency power back toward the source, and to provide very high isolation between receiver and transmitter terminals. The characteristic impedance of the relay is matched to the transmission line impedance of the system, for example, 50 ohms.

- **Time delay relay**

Timing relays are arranged for an intentional delay in operating their contacts. A very short (a fraction of a second) delay would use a copper disk between the armature and moving blade assembly. Current flowing in the disk maintains magnetic field for a short time, lengthening release time. For a slightly longer (up to a minute) delay, a dashpot is used. A dashpot is a piston filled with fluid that is allowed to escape slowly; both air-filled and oil-filled dashpots are used. The time period can be varied by increasing or decreasing the flow rate. For longer time periods, a mechanical clockwork timer is installed.

Some relays are constructed with a kind of "shock absorber" mechanism attached to the armature which prevents immediate, full motion when the coil is either energized or de-energized. This addition gives the relay the property of time-delay actuation. Time-delay relays can be constructed to delay armature motion on coil energization, de-energization, or both.

4. SOFTWARE SPECIFICATION

4.1. DEBIAN

Debian is a Unix-like computer operating system that is composed entirely of free software, most of which is under the GNU General Public License and packaged by a group of individuals participating in the Debian Project.

As one of the earliest Linux distributions, it was decided that Debian was to be developed openly and freely distributed in the spirit of the GNU Project. This decision drew the attention and support of the Free Software Foundation, which sponsored the project for one year from November 1994 to November 1995. Upon the ending of the sponsorship, the Debian Project formed the non-profit organisation Software in the Public Interest.

Features

Debian has access to online repositories that contain over 50,000 software packages making it one of the largest software compilations. Debian officially contains only free software, but non-free software can be downloaded from the Debian repositories and installed. Debian includes popular free programs such as LibreOffice, Firefox web browser, Evolution mail, K3b disc burner, VLC media player, GIMP image editor, and Evince document viewer. Debian is a popular choice for web servers (cf. LAMP).

4.2. RASPBIAN

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

Pros:

- Large and active community
- Low price for the capabilities offered

Cons:

- The Raspberry Pi 1's processor falls uncomfortably between the processor families that Debian has chosen to target. While Raspbian solves this to some degree an unofficial port will always give less certainty than an official one.
- 3D acceleration is not integrated with X or other standard mechanisms and the Raspberry Pi Foundation don't seem to show any interest in doing so. Therefore 3D applications will require Pi specific builds.
- A binary blob used by the GPU must be present on the SD card for the system to boot.
- While some hardware documentation has been released the documentation is sorely lacking.

4.3. PYTHON

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming.

Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and many language features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a mix of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. The design of Python offers some support for functional programming in the Lisp tradition. The language has map(), reduce() and filter() functions; list comprehensions, dictionaries, and sets; and generator expressions. Rather than requiring all desired functionality to be built into the language's core, Python was designed to be highly extensible.

Python can also be embedded in existing applications that need a programmable interface. This design of a small core language with a large standard library and an easily extensible interpreter was intended by Van Rossum from the start because of his frustrations with ABC, which espoused the opposite mindset.

Python's developers strive to avoid premature optimization, and moreover, reject patches to non-critical parts of CPython that would offer a marginal increase in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or try using PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is making it fun to use.

This is reflected in the origin of the name, which comes from Monty Python, and in an occasionally playful approach to tutorials and reference materials, such as using examples that refer to spam and eggs instead of the standard foo and bar.

Libraries

Python has a large standard library, commonly cited as one of Python's greatest strengths, providing tools suited to many tasks. This is deliberate and has been described as a "batteries included" Python philosophy. For Internet-facing applications, many standard formats and protocols (such as MIME and HTTP) are supported. Modules for creating graphical user interfaces, connecting to relational databases, pseudorandom number generators, arithmetic with arbitrary precision decimals, manipulating regular expressions, and doing unit testing are also included.

DEVELOPMENT

Python's development is conducted largely through the **Python Enhancement Proposal** (PEP) process. The PEP process is the primary mechanism for proposing major new features, for collecting community input on an issue, and for documenting the design decisions that have gone into Python. Outstanding PEPs are reviewed and commented upon by the Python community and by Van Rossum, the Python project's benevolent dictator for life.

4.4. OPEN SOURCE COMPUTER VISION LIBRARY (OPENCV)

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 7 million. The library is used extensively in companies, research groups and by governmental bodies.

History

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls.

The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. In mid-2008, OpenCV obtained corporate support from Willow Garage, and is now again under active development. A version 1.1 "pre-release" was released in October 2008.The second major release of the OpenCV was on October 2009.

OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations. In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.

Applications

OpenCV's application areas include:

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Stereopsis stereo vision: depth perception from 2 cameras
- Structure from motion (SFM

5. PROGRAM

```
#import
import RPi.GPIO as GPIO
import time
import sys,numpy,os,cv2
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEBase import MIMEBase
from email.MIMEText import MIMEText
from email.Utils import COMMASPACE,formatdate
from email import Encoders
from os import fork, chdir, setsid, umask
from sys import exit
import RPi.GPIO as GPIO
import time
import os
import syslog
import numpy as np
import cv2
import cv2.cv as cv
import serial
from curses import ascii

port = serial.Serial(
        port='/dev/ttyS0',
        baudrate=9600,
        parity=serial.PARITY_NONE,
        stopbits=serial.STOPBITS_ONE,
        bytesize=serial.EIGHTBITS,
        timeout=2
)

number="+919496369807"
message="unauthorized access  to Vehicle"

size = 4
fn_haar = 'haarcascade.xml'
```

```
fn_dir = 'att_faces'

# Define GPIO to LCD mapping
LCD_RS = 18
LCD_E  = 22
LCD_D4 = 10
LCD_D5 = 9
LCD_D6 = 11
LCD_D7 = 16

Lock_switch = 21
Lock_door = 20


# Define some device constants
LCD_WIDTH = 16    # Maximum characters per line
LCD_CHR = True
LCD_CMD = False

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line

# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005

Authenticated=["Renjith","Kiran","test"]

# Part 1: Create fisherRecognizer
print('Training...')
# Create a list of images and a list of corresponding names
(images, lables, names, id) = ([], [], {}, 0)
for (subdirs, dirs, files) in os.walk(fn_dir):
    for subdir in dirs:
        names[id] = subdir
        subjectpath = os.path.join(fn_dir, subdir)
        for filename in os.listdir(subjectpath):
            path = subjectpath + '/' + filename
```

```
            lable = id
            images.append(cv2.imread(path, 0))
            lables.append(int(lable))

        id += 1
(im_width, im_height) = (112, 92)

# Create a Numpy array from the two lists above
(images, lables) = [numpy.array(lis) for lis in [images, lables]]

# OpenCV trains a model from the images
# NOTE FOR OpenCV2: remove '.face'
model = cv2.createFisherFaceRecognizer()
model.train(images, lables)


# Part 2: Use fisherRecognizer on camera stream
haar_cascade = cv2.CascadeClassifier(fn_haar)
webcam = cv2.VideoCapture(0)

def main():
  # Main program block
  GPIO.setwarnings(False)
  GPIO.setmode(GPIO.BCM)       # Use BCM GPIO numbers
  GPIO.setup(LCD_E, GPIO.OUT)  # E
  GPIO.setup(LCD_RS, GPIO.OUT) # RS
  GPIO.setup(LCD_D4, GPIO.OUT) # DB4
  GPIO.setup(LCD_D5, GPIO.OUT) # DB5
  GPIO.setup(LCD_D6, GPIO.OUT) # DB6
  GPIO.setup(LCD_D7, GPIO.OUT) # DB7

  GPIO.setup(Lock_switch, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
  GPIO.setup(Lock_door, GPIO.OUT)
  GPIO.output(Lock_door, 0)

  # Initialise display
  lcd_init()
```

```
  lcd_string("   ioT  Based   ",LCD_LINE_1)
  lcd_string("ACCESS   CONTROL",LCD_LINE_2)
  time.sleep(4)
  lcd_byte(0x01, LCD_CMD)

  while True:

    if GPIO.input(Lock_switch)==0:
      GPIO.output(Lock_door, 0)
      lcd_string(" PRESS   SWITCH ",LCD_LINE_1)
      lcd_string("     --ON--     ",LCD_LINE_2)
    else:
      open_door()
    time.sleep(3)

def sendMail(to,subject,text,files=[]):
    assert type(to)==list
    assert type(files)==list
    me="myprojectcep@gmail.com"   # from mail address
    pas="perumon123"  # from email addreess password
    msg=MIMEMultipart()
    msg['From']=me
    msg['To']=COMMASPACE.join(to)
    msg['Date']=formatdate(localtime=True)
    msg['subject']=subject

    msg.attach(MIMEText(text))

    for file in files:
            part=MIMEBase('application',"octet-stream")
            part.set_payload(open(image_path,"rb").read())
            Encoders.encode_base64(part)
            part.add_header('content-
Disposition','attachment',filename='Camera.jpg')
            msg.attach(part)

    server=smtplib.SMTP('smtp.gmail.com:587')
    server.ehlo_or_helo_if_needed()
```

```
    server.starttls()
    server.ehlo_or_helo_if_needed()
    server.login(me,pas)
    server.sendmail(me,to,msg.as_string())
    server.quit()

my_subject='Alert from Home'
image_name='Image'
image_path='/home/pi/Desktop/ioT_Access_Control/'+image_name+'.png'
att_name=image_name+'.png'

def lcd_init():
  # Initialise display
  lcd_byte(0x33,LCD_CMD) # 110011 Initialise
  lcd_byte(0x32,LCD_CMD) # 110010 Initialise
  lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
  lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
  lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
  lcd_byte(0x01,LCD_CMD) # 000001 Clear display
  time.sleep(E_DELAY)

def lcd_byte(bits, mode):
  # Send byte to data pins
  # bits = data
  # mode = True  for character
  #        False for command

  GPIO.output(LCD_RS, mode) # RS

  # High bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x10==0x10:
    GPIO.output(LCD_D4, True)
  if bits&0x20==0x20:
    GPIO.output(LCD_D5, True)
```

```
  if bits&0x40==0x40:
    GPIO.output(LCD_D6, True)
  if bits&0x80==0x80:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  lcd_toggle_enable()

  # Low bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x01==0x01:
    GPIO.output(LCD_D4, True)
  if bits&0x02==0x02:
    GPIO.output(LCD_D5, True)
  if bits&0x04==0x04:
    GPIO.output(LCD_D6, True)
  if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  lcd_toggle_enable()

def lcd_toggle_enable():
  # Toggle enable
  time.sleep(E_DELAY)
  GPIO.output(LCD_E, True)
  time.sleep(E_PULSE)
  GPIO.output(LCD_E, False)
  time.sleep(E_DELAY)

def lcd_string(message,line):
  # Send string to display

  message = message.ljust(LCD_WIDTH," ")
```

```
lcd_byte(line, LCD_CMD)

  for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)

def open_door():
  global access_sucess
  lcd_string("      ----      ",LCD_LINE_1)
  lcd_string("  FACE   CAMERA ",LCD_LINE_2)
  time.sleep(5)
  lcd_byte(0x01, LCD_CMD)
  access_sucess=0
  for ll in range(10):
    repeated_match()
  if access_sucess==True:
    lcd_string("Authentication  ",LCD_LINE_1)
    lcd_string("         Success",LCD_LINE_2)
    print "Authentication got"
    GPIO.output(Lock_door, 1)
    time.sleep(1)
    for ll in range(10):
       tt=ll+1
       time.sleep(1)
    GPIO.output(Lock_door, 0)


  else:
     i=0
     lcd_string("Authentication  ",LCD_LINE_1)
     lcd_string("         Denied",LCD_LINE_2)
     print "Authentication Denied"
     #sms unauthorized access

     #mail photo
     sendMail(["renjith@beginow.in"],
                    my_subject,
                    "Image Received from Home, Found an Intruder",
                    [att_name,"ignore_text_file.txt"])
```

```
GPIO.output(Lock_door, 0)
     time.sleep(15)
     #send_msg(message,number)
def repeated_match():
    global access_sucess
    j=0
    for j in range(5):
        (rval, frame) = webcam.read()
        frame=cv2.flip(frame,1,0)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        mini = cv2.resize(gray, (gray.shape[1] / size, gray.shape[0] / size))
        faces = haar_cascade.detectMultiScale(mini)
        cv2.imwrite("Image.png",gray)
        for i in range(len(faces)):
            face_i = faces[i]
            (x, y, w, h) = [v * size for v in face_i]
            face = gray[y:y + h, x:x + w]
            face_resize = cv2.resize(face, (im_width, im_height))

            # Try to recognize the face
            prediction = model.predict(face_resize)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)

            # Write the name of recognized face
            # [1]
            if prediction[1]<200:
                cv2.putText(frame,
                '%s - %.0f' % (names[prediction[0]],prediction[1]),
                (x-10, y-10), cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
                cv2.imshow('OpenCV', frame)
                print (names[prediction[0]])
                for qq in range(3):
                  if names[prediction[0]]==Authenticated[qq]:
                    print "sucessd"
                    access_sucess=1
            else:
                cv2.putText(frame,
                'Unknown',
```

```
        (x-10, y-10), cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
       key = cv2.waitKey(10)
 if key == 27:
          break

def gsm_init():
    port.write("AT\r\n")
    c=port.read(17)
    print c
    port.write("AT+CMGF=1\r\n")
    c=port.read(17)
    print c

def send_msg(msg,num):
    port.write("AT+CMGF=1\r\n")
    c=port.read(15)
    print c
    port.write('AT+CMGS="%s"\r\n' % num)
    c=port.read(15)
    print c
    port.write(msg)
    c=port.read(15)
    print c
    port.write(ascii.ctrl('z'))
    c=port.read(15)
    print c

if __name__ == '__main__':

  try:
    main()
  except KeyboardInterrupt:
    pass
  finally:
    lcd_byte(0x01, LCD_CMD)
    lcd_string("Goodbye!",LCD_LINE_1)
    GPIO.cleanup()
```

6. CIRUIT DIAGRAM

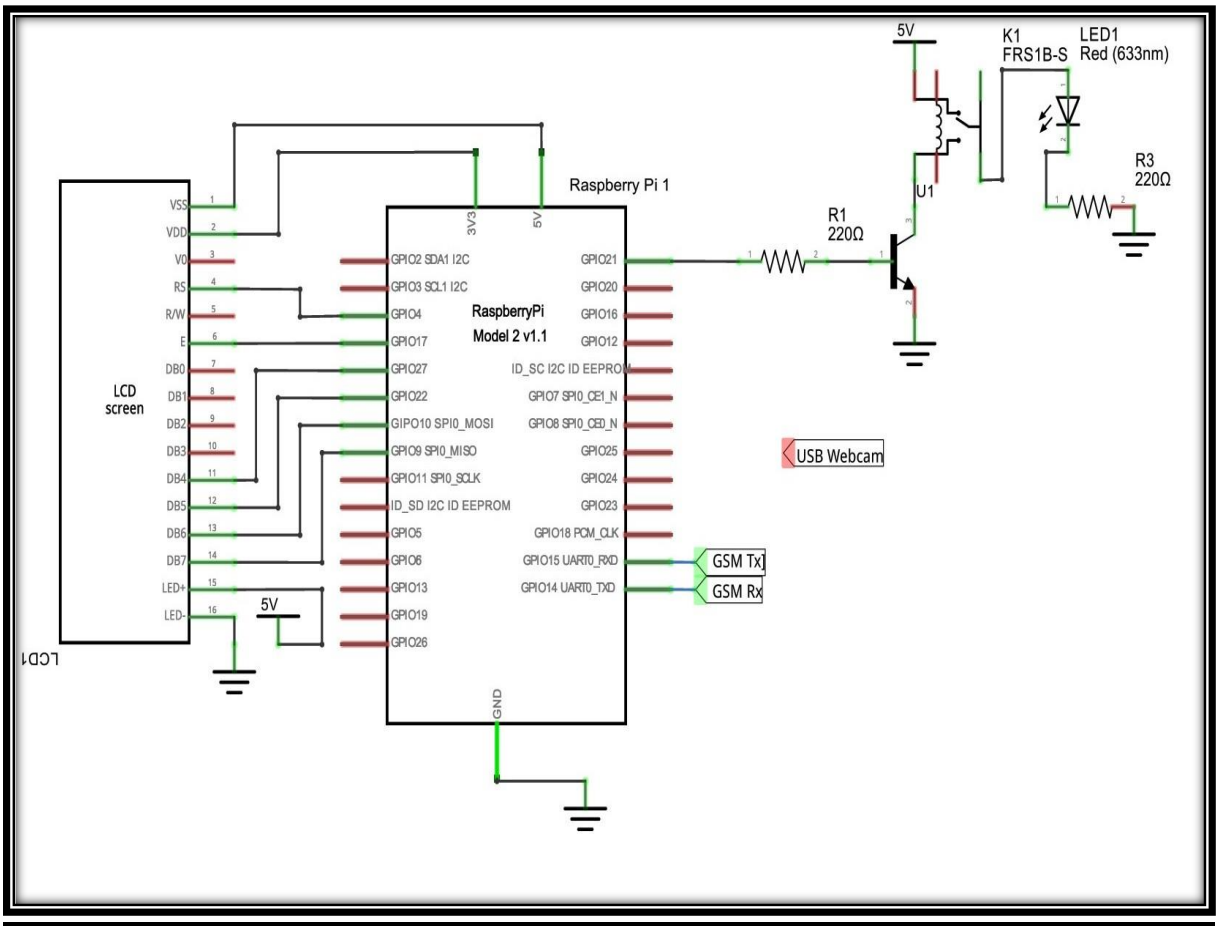


Fig: - circuit diagram

7. CIRCUIT DIRGRAM EXPLANATION

The Raspberry Pi is a credit card sized single-board computer with an open-source platform that has a thriving community of its own, similar to that of the Arduino. It can be used in various types of projects from beginners learning how to code to hobbyists designing home automation systems. There are a few versions of the Raspberry Pi, but the latest version, has improved upon its predecessor in terms of both form and functionality.

The Raspberry Pi Model B features:

- More GPIO
- More USB
- Micro SD
- Lower power consumption
- Better audio
- Neater form factor

This higher-spec variant increases the Raspberry pi GPIO pin count from 26 to 40 pins. There are now four USB 2.0 ports compared to two on the Model B. The SD card slot has been replaced with a more modern push-push type micro SD slot. It consumes slightly less power, provides better audio quality and has a cleaner form factor.

SoC: Broadcom BCM2835 media processor (datasheet, BCM2835 datasheet errata, unofficial pinout, BCM2835 Register documentation - based on GPU source code) system-on-chip featuring:

CPU core: ARM1176JZF-S ARM11 core clocked at 700 MHz; ARM VFP. The ARM11 core implements the ARMv6 Architecture. For details on ARM instruction sets and naming conventions, see ARM architecture and List of ARM microprocessor cores.

GPU core: a Broadcom VideoCore IV GPU providing OpenGL ES 1.1, OpenGL ES 2.0, hardware-accelerated OpenVG 1.1, Open EGL, OpenMAX and 1080p30 H.264 high-profile decode. There are 24 GFLOPS of general purpose compute and a bunch of texture filtering and DMA infrastructure. Eben worked on the architecture team for this and the Raspberry Pi team are looking at how they can make some of the proprietary features available to application programmers

DSP core: There is a DSP, but there isn't currently a public API (Liz thinks the BC team are keen to make one available at some point) thread 256 MiB of (Hynix MobileDDR2 or Samsung Mobile DRAM) SDRAM (or 512 MB Mobile DRAM on later boards). The RAM is physically stacked on top of the Broadcom media processor (package-on-package technology). Here is a photo of the SDRAM (left) and BCM2835 (right) ball grid arrays on JamesH's finger. You are looking at the bottom side. The BCM2835 top side has a land grid array which matches the SDRAM ball grid array. Here is a highly magnified side view of the SDRAM stacked on top of the BCM2835 stacked on top of the PCB PoP stack (you can see why its job can only be done by robots!).

LAN9512 (Data Brief | Data Sheet) **(Model B)** providing:

10/100 Mbit/s Ethernet (Auto-MDIX), 2x USB 2.0

S1: Micro USB power jack (5 V - Power Only)

S2: DSI interface. 15-pin surface mounted flat flex connector, providing two data lanes, one clock lane, 3.3 V and GND.

S3: HDMI connector providing type A HDMI 1.3a out

S4: Composite Video connector: RCA

S5: MIPI CSI-2 interface. 15-pin surface mounted flat flex connector.

S6: Audio connector: 3.5mm stereo jack (output only)

S8: SD/MMC/SDIO memory card slot (underside)

S7: Either 1x USB 2.0 **(Model A)** 2x USB 2.0 **(Model B)**

P1: 26-pin (2x13) 2.54 mm header expansion, providing: see Low-level peripherals

8 GPIOs at 3.3 V

2-pin UART serial console, 3.3 V TTL (debug); or 2 GPIOs at 3.3 V

I²C interface (3.3 V); or 2 GPIOs at 3.3 V

SPI interface (3.3 V); or 5 GPIOs at 3.3 V, 3.3 V, 5 V and GND supply pins

ARM JTAG (if pins are reconfigured in software - on Revision1.0 boards one signal would also need to be taken from S5) I²S interface (if pins are reconfigured in software, hardware hack may be required[5]) P2: 8-pin 2.54 mm header expansion (header not fitted on Revision 2.0 boards), providing GPU JTAG (P3: 7-pin 2.54 mm header expansion, providing LAN9512 JTAG (pin 6 is nofit for locating) Transistors are used in order to drive the relay, as raspberry pi can't drive a 12V relay directly using its GPIO pin, and also LCD is provided in order to display settings for user interface.

8. PCB FABRICATION

Printed Circuit Broad (PCB) is a mechanical assembly consisting of layers of fiberglass sheet laminated with etched copper patterns. It is used to mount electronic parts in a rigid manner suitable for packaging.

The type of integrated circuit components used in the fabrication process has an important role in the design of PCB. The conductor width, spacing between the signal conductors etc, are calculated to give optimum wave impedance of the conductor's lines. Optimum wave impedance gives minimum delay or rising and trailing edge of the pulse in digital circuit.

Art Work Generation

The generation of PCB artwork should be considered as the first step of the PCB manufacturing process. The artwork is generated at 1:1:1 or 4:1 scale according to the accuracy needed. Ink drawing on a transparent drawing paper or cut up and strip method are the methods used for the art work generation.

Routing

Presently artwork generation is not used for the PCB fabrication. Instead there are many types of software available for the routing of PCBs. Mainly used software's are CAD SOFTWARE EAGLE, ORCAD, TRAXMAKER, EASYPCB, PORTAL etc. Here we make use of CAD SOFT EAGLE.

1. Manual. Traces are placed manually as done in the traditional method where you change the path of the trace every time you click the mouse.

2. Follow-me. This highly interactive method combines the power of an auto router with the control and flexibility of manual routing.

3. **Auto Router.** This fully automated method will auto route an entire trace by clicking on a rats net line

Then using a laser printer solution prints the routed diagram. Laser printer is very affordable, fast and good quality. The printer used must have at least 600dpi resolution for all but the simplest PCB swill require only 300DPI resolution. It is very important that the printer produces the good solid black with no toner pinholes.

When using tracing paper or drafting film, always use manual paper feed, and set the straightest possible paper output path, to keep the artwork as flat as possible and minimize jamming. The printed diagram is then converted into film by using vertically mounted cameras.

Screen-printing

Screen-printing is arguably the most versatile of all printing process. It can be used to print on a wide variety of substrates, including paper, paper board, plastics, glass, metals, posters, labels, decals, signage, and all types of textiles and electronic circuit boards. The advantage of screenwriting over other print processes is that the press can print on substrates of any shape, thickness and size.

A significant characteristic of screen-printing is that a greater thickness of the ink can be applied to the substrate than is possible with other printing techniques. This allows for some very interesting effects that are not possible using other printing methods. Because of the simplicity of the

application process, a wider range of inks and dyes are available for use in screen-printing than for use in any other printing process.

Screen Printing Process Overview

Screen-printing consists of three elements: the screen which is the image carrier, the squeegee; and ink. The screen-printing process uses a porous mesh stretched tightly over a frame made of wood or metal. Proper tension is essential to accurate color registration. The mesh is made of porous fabric or stainless steel mesh. A stencil is produced on the screen either manually or photo chemically. The stencil defines the image to be printed in other printing technologies this would be referred to as the image plate.

Screen printing ink is applied to the substrate by placing the screen over the material. Ink with a paint-like consistency is placed on to the top of the screen. Ink is then forced through the fine mesh openings using a squeegee that is drawn across the screen, applying pressure thereby forcing the ink through the open areas where no stencil is applied, thus forming an image on the printing substrate. The diameter of the threads and the thread count of the mesh will determine how much ink is deposited onto the substrates.

Etching

In all subtractive PCB process, etching is one of the most important steps. The final copper pattern is formed by selective removal of all unwanted copper, which is not protected by an etch resist. There are two basic ways that you can remove unwanted copper from copper-clad substrates to form electronic circuits: mechanical etching and chemical milling (etching).

Mechanical Etching

It involves the use of a precise numerically controlled multi-axis machine tool and a special milling cutter to remove a narrow strip of copper from the boundary of each pad and trace. The removal of this copper electrically isolates the circuit element from the rest of the foil.

Chemical Etching

It relies on the action of any one of a family of corrosive liquids to dissolve away-unwanted copper in order to define the desired circuit pattern. But in practice, factors like under-etching and overhang compliance the etching process.

Under Etching

During etching process etching must progress vertically. But in practice etching takes place in the sideways which attacks the pattern below the etch resist. Under etching can be minimized by keeping the etching, time as short as possible and by pressurized perpendicular discharge of the etched towards the surface to be etched.

Rinsing

After etching is over, the ferric chloride contaminated surface is cleaned. A simple spray water rinse is a dip in a 5% oxalic acid solution to remove the iron and copper salts.

Plating

Plating of metal can be accomplished on a copper pattern by three methods:

1) Immersion plating

2) Electrolysis plating

3) Electroplating

Immersion plating

It is the deposition of metallic coating on a substrate, by chemical replacement, from a solution of a salt of the coating metal. Advantages of immersion plating are simplicity, minor capital expenses and increase in deposits. Tin and its alloys and gold are the two most commonly used coating metals.

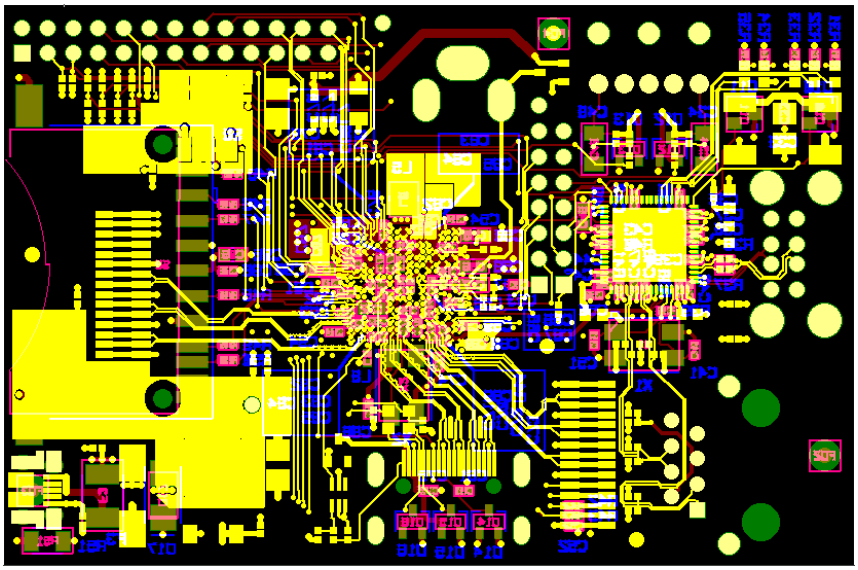


Fig: - Pi 3 PCB

9. SOLDERING DETAIL

Soldering techniques

Soldering is an important skill for electrical technician. Good soldering is important for proper operation of equipment.

Solder is an alloy of tin and lead. The solder that is most used is 60/40 solder. This means that it is made from 60%tin and 40% lead. Solder melts at a temperature of about 400 degree Fahrenheit. For solder to adhere to join, the parts must to enough to melt the solder. Rosin flux is contained inside the solder. It is called rosin-core solder. A good mechanical joint must be made when soldering. Heat is then applied until the material rare hot. When they are hot, solder is applied to the joint. The heat of the metal parts is used to melt the solder. Only a small amount of heat should be used sparingly. The joint should appear smooth and thin. If it does not, it could be a "cold" solder joint. This is called a “cold joint". Care should be taken not to damage PCB when soldering parts on to them. Small, low wattage irons should be used with PCB and semiconductor devices.

Need of flux

Flux is needed for achieving desired clean lines of the surface. Most metals tend to form compounds with atmospheric oxygen, which leads a coating of oxide even at room temperature, react chemically with oxides and disperse the reaction products. Fluxes are applied before and during soldering.

Soldering Tools

To facilitate soldering work, various tools are necessary. The most essential tools in the soldering practice are:

Soldering iron

A soldering should supply sufficient heat to melt solder by heat transfer, when the iron tip is applied to the connection to the soldered. There are two general classes of soldering irons.

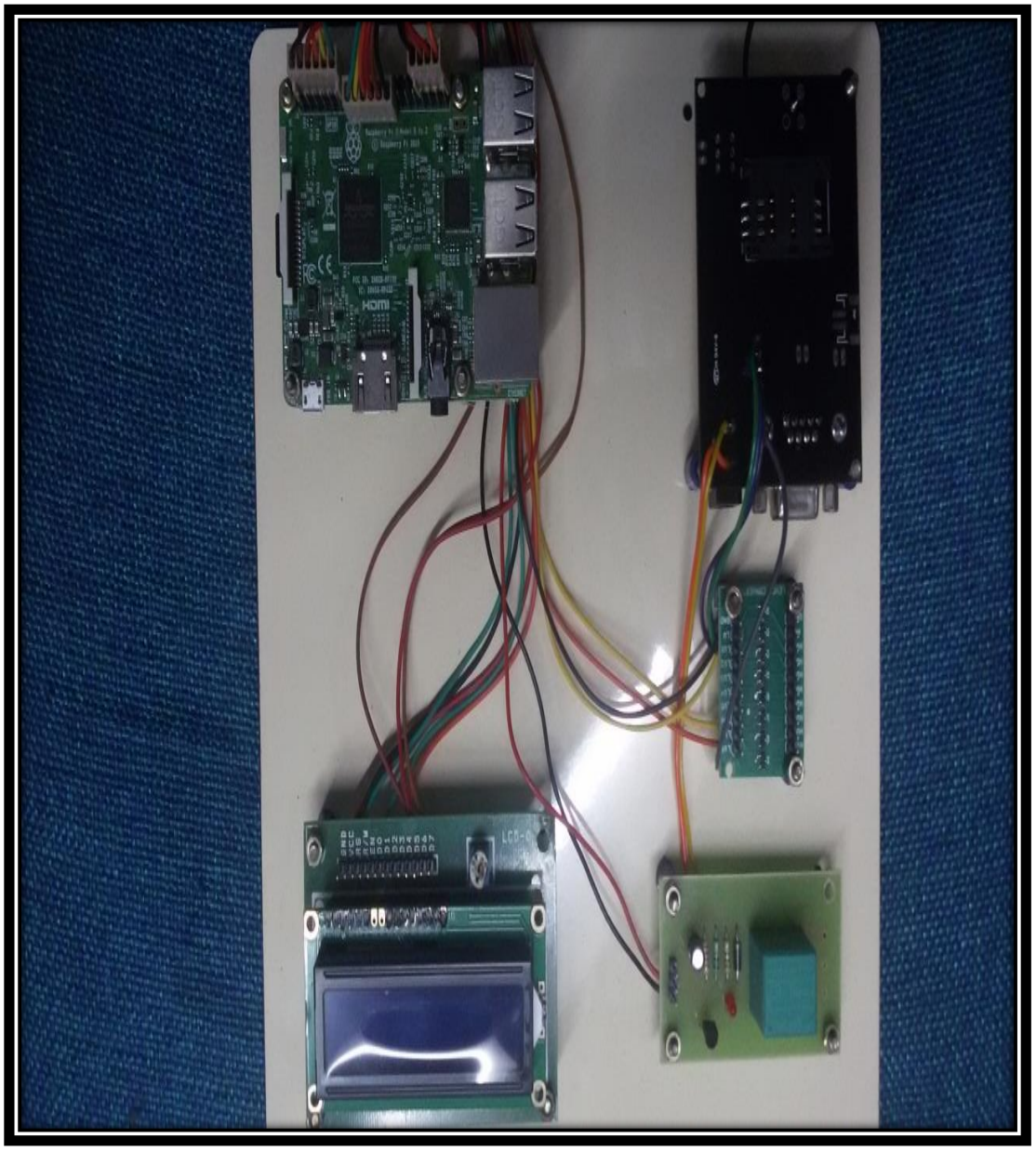
a) **Soldering pencils:** Soldering pencils are lightweight soldering tools, which can generate as little as 10W or as much as SOW. A) 25W is well suited for light duty works such as soldering on PCBs.

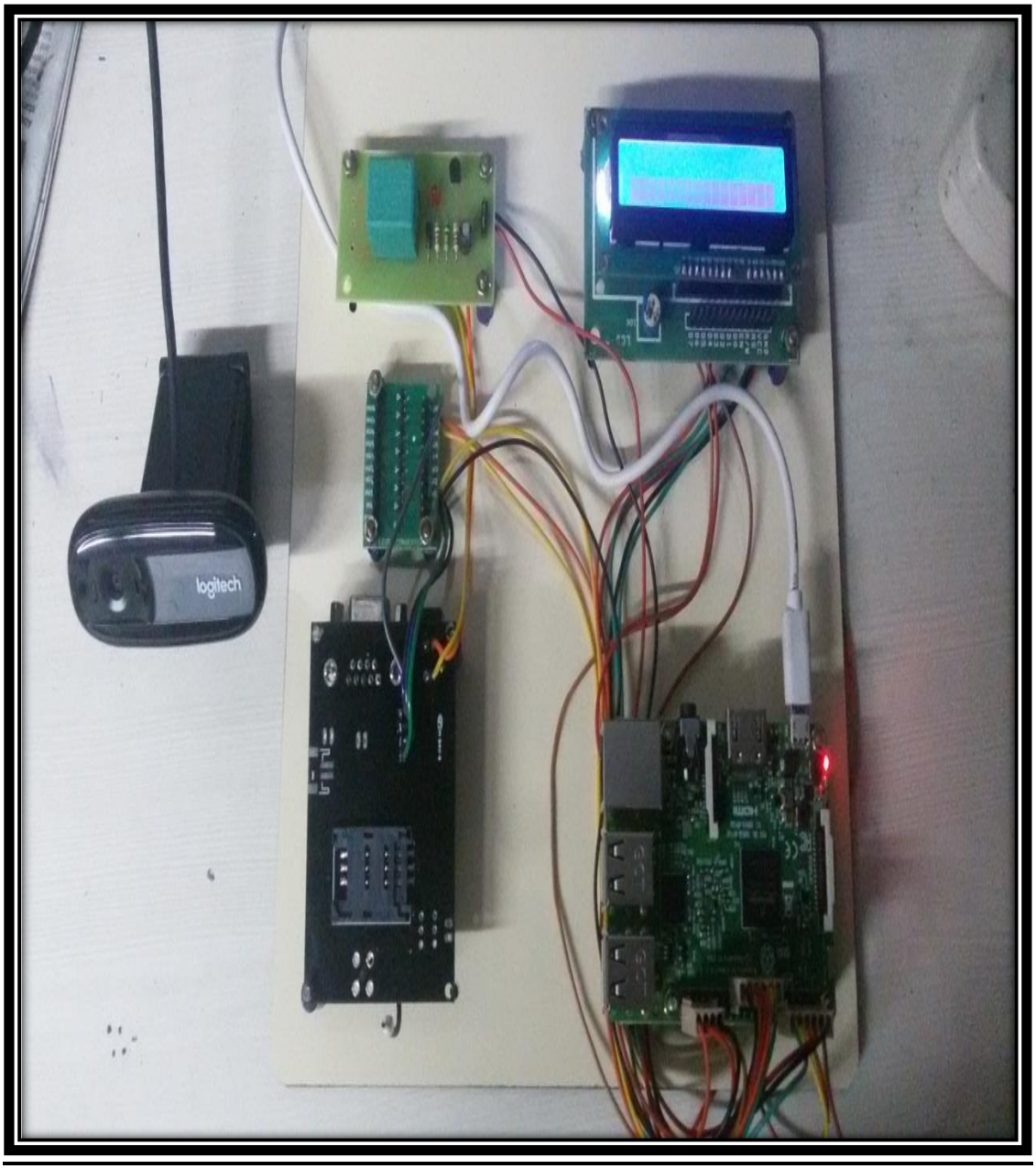
b) **Soldering gun**: A gun is heavier and generates more heat than the average pencils. Soldering of heavy-duty conductors requires the use of a gun because it can generate enough heat to quickly being a heavy metal joint at the proper soldering temperature. These soldering tools are called gun-soldering station.

Strippers and bending tools

Strippers are used to remove insulation from the wire. Bending tools are those having smooth bending surface so that they do not cause any damage to the component.

10. FINAL VIEW




logitech

11. ADVANTAGE

Prevention of Theft

An access control system's primary task is to restrict access. This is critical when access to a person's account information is sufficient to steal or alter the owner's identity. Many websites that require personal information for their services, especially those that need a person's credit card information or a Social Security number, are tasked with having some sort of access control system in place to keep this information secure.

Varying Levels of Security

As technology has increased with time, so have these control systems. Simple four-digit PIN information secure. For example, there are now locks with biometric scans that can be attached to locks in the home. The Biometrics Institute states that there are several types of scans. These scan-based locks make it impossible for someone to open and password are not the only options available to a person who wants to keep the door to a person's home without having the right physical features, voice or fingerprint. In some instances, such as with large businesses, the combination of both a biometric scan and a password is used to create an ideal level of security.

12. DISADVANTAGE

Hacking

Access control systems can be hacked. When a system is hacked, a person has access to several people's information, depending on where the information is stored. Wired reported how one hacker created a chip that allowed access into secure buildings, for example. Not only does hacking an access control system make it possible for the hacker to take information from one source, but the hacker can also use that information to get through other control systems legitimately without being caught. Despite access control systems increasing in security, there are still instances where they can be tampered with and broken into.

IOT Risks and Security Implications

Although smart technology implementation can provide great business opportunities, devices are not always designed with security in mind. This can be especially problematic for systems that connect to your corporate network or the Internet, such as Wi-Fi and “cloud-controlled” solutions. If not secured and managed, these systems can serve as a gateway for hackers to gain access to your network and data. Some risks of IOT devices may include:

- Installation of malware, providing persistent access to the local or wide-area network, and impacting network availability, systems and/or data.
- Ability for intruders to disable alarms, cameras, or locking systems, gaining physical access to your business.
- Unauthorized control of devices, which could result in failure of the equipment, loss of use, or similar damage to your business operations.

13. FUTURE SCOPE

- extending security to remote locations and gates where conduit and wiring may be difficult to reach the area;
- adding security to non-critical areas, such as cabinets, utility closets, elevators and conference rooms;
- installing in structures with poor cabling, including buildings with asbestos and other issues affecting wiring, as well as for non-critical doors in very old buildings with concrete or cinderblock walls that could make it difficult to run cables to interior doors;
- deploying in small businesses with simple, key-based security that are seeking greater convenience (over mechanical keys) with basic security functionality; or
- when interior doors are pre-ordered/prepped for wireless locks as part of a new install. In this case, running wire may still be more cost-effective, depending on whether the building is pre-wired for Wi-Fi, if the doors are located in high-traffic areas, and many other factors.

14. CONCLUSION

This paper presents the design and the implementation of an interactive home security system with the GSM, Wifi communication and Web-enabled measurement and control systems. The Web based monitor and automatic control of equipment is forming a trend in automation field. Replacing PC with low-cost single chip processor which can make administrators to get parameters of different remote devices and send control information to field equipment's at any time through Internet.

The GSM is an excellent choice for this due to its extensive coverage. Since SMS is a text based protocol, even the most basic GSM systems can have an access to the status of the devices or make changes on these states. The complete system is secured through a login E-mail and Webpage password based authentication. The design is completely wireless and integrated with the software to form a low cost, robust and easily operable system. WiFi communication makes the system easy to install. The GSM, E-mail and Web based controlled duplex communication system provides a powerful decision making device concept for adaptation to several smart home scenarios.

15. REFERENCE

- Jinsoo Han; Chang-Sic Choi; Ilwoo Lee, "More efficient home energy management system based on ZigBee communication and infrared remote controls," *Consumer Electronics, IEEE Transactions on* , vol.57, no.1, pp.85,89, February 2011.

- Chia-Hung Lien; Ying-Wen Bai; Ming-Bo Lin, "Remote-Controllable Power Outlet System for Home Power Management," *Consumer Electronics, IEEE Transactions on* , vol.53, no.4, pp.1634,1641, Nov. 2007.

- Vernon, S.; Joshi, S.S., "Brain–Muscle–Computer Interface: Mobile-Phone Prototype Development and Testing," *Information Technology in Biomedicine, IEEE Transactions on* , vol.15, no.4, pp.531,538, July 2011.

- www.google.co.in

- https://www.wikipedia.org/

- Erdem, H.; Uner, A., "A multi-channel remote controller for home and office appliances," *Consumer Electronics, IEEE Transactions on* , vol.55,
- no.4, pp.2184,2189, November 2009.