CHAPTER-1
INTRODUCTION

The main objective of our project is to provide an image of the visitor to the owner irrespective of location of the owner. We also provide an alert message to the owner whenever the doorbell is pressed. We will also save the images in their drive so that they can view them in case there is no signal at the time when the person visits their house.

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

The internet of things (IoT) is the network of physical objects, devices, vehicles, buildings and other items which are embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.

The internet of things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more-direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit; when IoT is augmented with sensors and actuators, the technology becomes an instance of them or general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

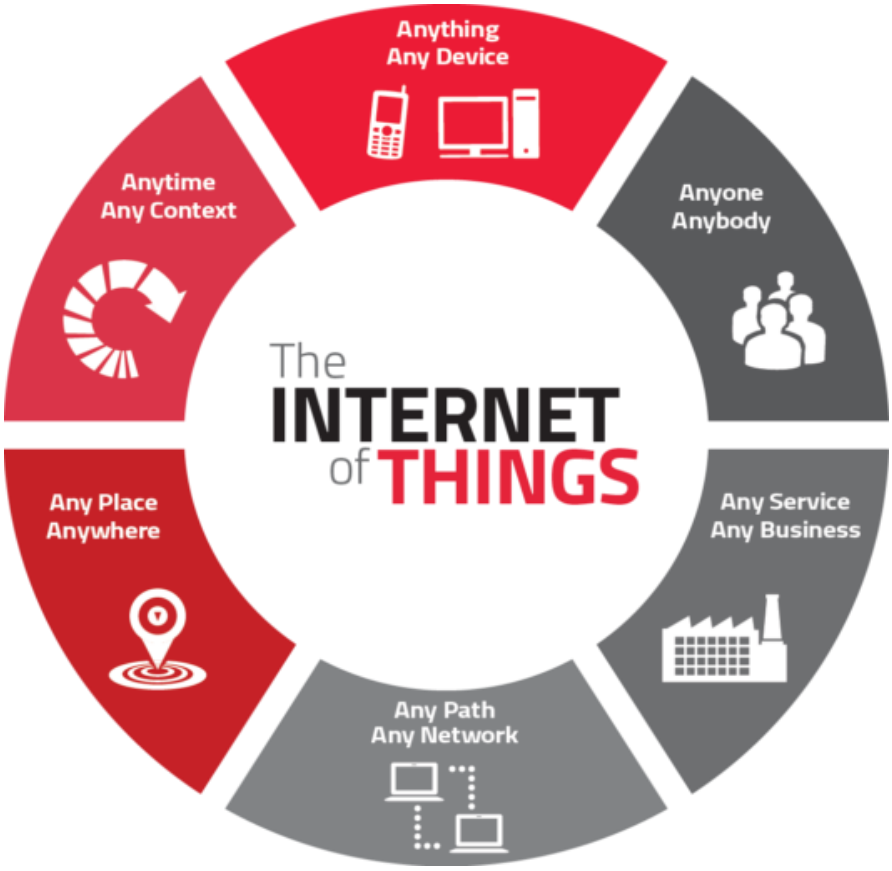


Fig 1.1: Architeture of IoT

Microcontrollers as the name suggests are small controllers. They are like single chip computers that are often embedded into other systems to function as processing/controlling unit. For example the remote control is using probably has microcontrollers inside that do decoding and other controlling functions.

This implementation is designed with ARM7TDMI processor. In these different modules such as LPC2148, GSM, GPS, IR pair, 16*2 LCD are used. It is implemented by using two modules Base station module and In-Bus module. The In-Bus module consists of IR pair to count the number of persons in the bus and to find out exact location of bus GPS is used and the information can be obtained as a message by using a GSM module.

Matrices are the basic elements of the MATLAB environment. A matrix is a two-dimensional array consisting of m rows and n columns. Special cases are column vectors (n = 1) and row vectors (m = 1).

GSM (Global System for Mobile communications) is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. GSM networks operate in four different frequency ranges. Most GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated.

CHAPTER-2
AIM & SCOPE

The project IDE is MATLAB and Keil software which process image. This software works on different platforms with minimum hardware and software requirements as stated.

We describe what features are in the scope of the software and what are not in the scope of the software developed.

In Scope:

- Visitor need not call to the Owner.
- Giving alert when the doorbell is pressed.
- The image of the visitor and an alert message is sent to Owner.
- Owner can call to the Visitor through the app if required.
- Visitor need not wait for a long time.

Out Scope:

- Never sends an alert message without pressing the doorbell.
- Owner doesn’t get any video recording.

2.1 EXISTING SYSTEM

The existing system is that the visitor should knock the door continuously or the visitor should call the owner to know whether the person is in the house or not.

2.2 PROPOSED SYSTEM

The proposed system is that when visitor presses the doorbell we can send an alert message to the owner so that the owner can be notified saying that someone is waiting for you at the door. Through our system the visitor can be benefitted by not waiting for a long time at the door. And Owner can also view the image of the visitor through our app.

2.3 FEASIBILITY STUDY

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Economical and Social feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Economic Feasibility
- Social Feasibility

2.3.1 Technical feasibility

The android app technically possible to implement with little amount of software and hardware costs. The technologies required for this sort of solutions are available in recent days like android phones. We can use this in our LINKAPP in any android phones. This mobile app is feasible to every android phone.

2.3.2 Economic Feasibility

The developed system is within the budget and this was achieved because most of the technologies used are freely available and the Hardware is also cost effective. Our system is economically feasible because open source technologies like Android SDK and MATLAB are used to develop our system. A user will not be burdened to use this system/device.

2.3.3 Social Feasibility

Our device is not developed in android studio, but we can also store all the images in the Google drive also. Many users own Android mobiles and also a Gmail account these days, than other mobiles such as Windows and IOS. In this regard, our device along with the app is socially feasible. As now we are providing this app only in English language.

CHAPTER-3
CONCEPTS & METHODS

3.1 PROBLEM DESCRIPTION

- The drawback of existing system is that the visitor should wait for a long time if he is not aware that the Owner is not in the house.
- Visitor should call the owner to know where he is or should wait till he comes.

3.2 PROPOSED SOLUTION

The proposed system is that whenever the doorbell is pressed an alert message will be sent to the owner indicating that someone is waiting for you at the door. To make it more advanced, a camera is also added, so that our IoT smart bell can take a picture of visitor, upload it and using FTP protocol and we are also storing these images in the Google drive.

Architecture Diagram

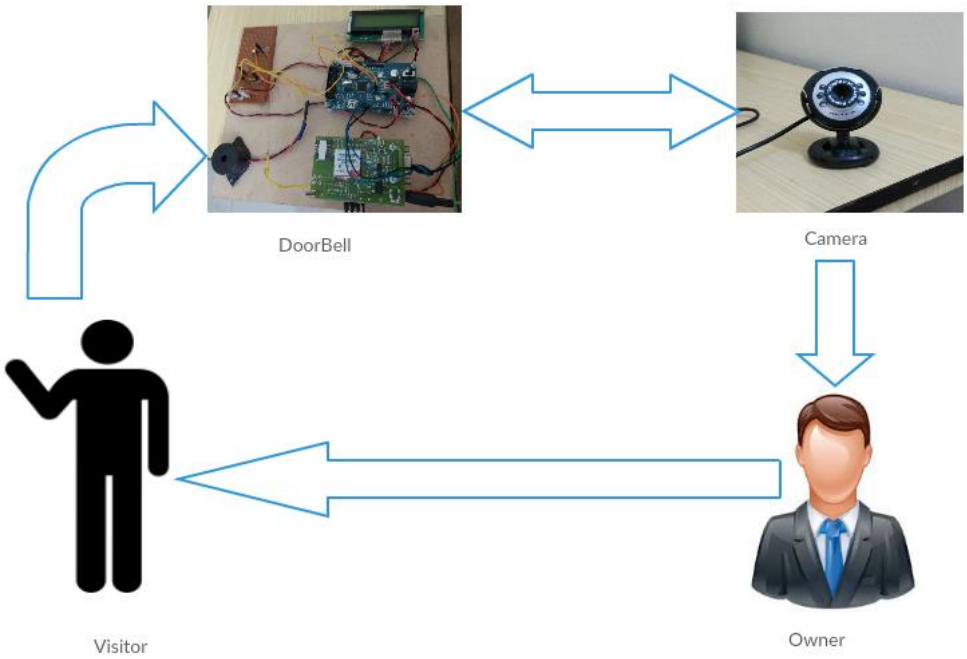


Fig 3.2.1 Architecture Diagram

3.3 SYSTEM ANALYSIS METHOD

Problem solving technique decomposes the system into component pieces for the purpose of studying how well those components work and interact to accomplish their work. It collects actual data and recommended feasible studying for improving the system functioning.

Objectives of System Analysis Method

- What is being done?
- How is being done?
- Who is doing this?
- When is he doing this?
- What is being done and how it is improved?

3.3.1 Use Case diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Fig 3.3.1.1 Use Case diagram

Here there are two actors namely Owner and Visitor, Owner should install the app and then if he is a new user then he should register and then login into the doorbell account. When the visitor presses the doorbell an alert message will be sent to the owner. In return the owner can call to the visitor.

3.3.2 Activity Diagram

In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

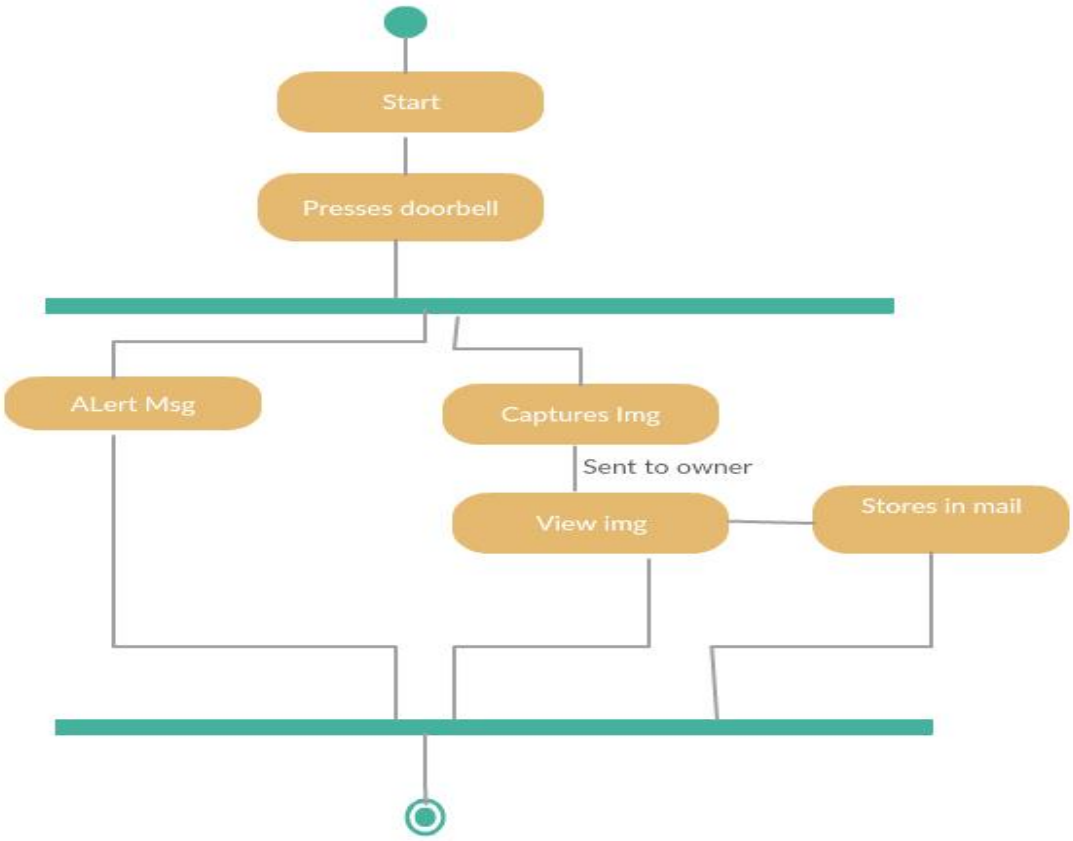


Fig 3.3.2.1 Activity diagram

The activity diagram tells about the working of the system, Here when visitor presses the doorbell an alert message will be sent to the registered mobile and also an image will be uploaded through the MATLAB so that the visitor is visible to the owner. Then owner can call to the visitor if he wishes. We are also trying to store all the images into the email given by the owner during registration process.

3.4 SYSTEM REQUIREMENTS

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of System requirements, is a generalisation of this first definition, giving the requirements to be met in the design of a system or sub-system. Typically an organisation starts with a set of Business requirements and then derives the System requirements from there.

Hardware Interfaces

- Arm-7 controller-3.3v
- GSM Modem-5v
- Voltage convertor-16/2
- Adapter-1 amp
- Jumper Wires
- Smart Phone
- Regulator
- Buzzer
- Button

Software Interfaces

- Operating system : Windows XP/7.
- Coding Language : JAVA,C
- Android studio
- MATLAB V9
- Keil v2

3.5 SYSTEM FEATURES

3.5.1 View Image

3.5.1.1 Description and Priority

The priority of this function is major. In this way, the owner will be easy to find the image of the visitor.

3.5.1.2 Stimulus/Response Sequences:

1.The user logs in to app.

2.The user views the owner's page.

3.The user clicks to view button.

4.The image can viewed.

3.5.1.3 Functional Requirements:

REQ 1: The user needs an Internet connection.

REQ 2: The user needs app

1. Purpose/Description: To view the image
2. Inputs: Click the view button
3. Processing: Get the image from MATLAB
4. Outputs: View image

3.5.2 Receiving a Message

3.5.2.1 Description and Priority:

It is one of the most important features of the System.

3.5.2.2 Stimulus/Response Sequences:

Visitor presses the doorbell

3.5.2.3 Functional Requirements:

REQ 1: The user needs an Internet connection.

REQ 2: The user needs to install app.

1. Purpose/Description: Owner receives an alert sound
2. Inputs: Pressing the doorbell
3. Processing: Take information from the cloud
4. Outputs: Owner can view the message

3.5.3Sending an Image

3.5.3.1 Description and Priority:

It is one of the most important features of the System.

3.5.3.2 Stimulus/Response Sequences:

Visitor presses the doorbell

3.5.3.3 Functional Requirements:

REQ 1: The user needs an Internet connection.

REQ 2: The user needs to install app.

1. Purpose/Description: Owner receives an Image
2. Inputs: Pressing the doorbell
3. Processing: Take information from the MATLAB
4. Outputs: Owner can view the message

3.5.4 Sending a Message

3.5.4.1 Description and Priority:

It is one of the most important features of the System.

3.5.4.2 Stimulus/Response Sequences:

Visitor presses the doorbell

3.5.4.3 Functional Requirements:

REQ 1: The user needs an Internet connection.

REQ 2: The user needs to install app.

1. Purpose/Description: To know someone is waiting at the door
2. Inputs: Pressing the doorbell
3. Processing: Take information from the device.
4. Outputs: Owner can view the message.

3.6 NON FUNCTIONAL REQUIREMENTS

3.6.1 Availability

The amount of time that is operational and available for use. As our system will be used by the user at any time the system must be available always. If there are any cases of updating they must be performed in a short interval of time without interrupting the normal services made available to the users.

3.6.2 Efficiency

Our system will effectively use its resources and is cost effective. All the resources can be effectively used by performing most of the validations at user side.

3.6.3Maintainability

Maintainability is the ability of a software to adapt to changes, improve over time, correct any bugs and be proactively fixed through preventive maintenance.

3.6.4 Usability

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

3.7 SYSTEM DESIGN

3.7.1 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

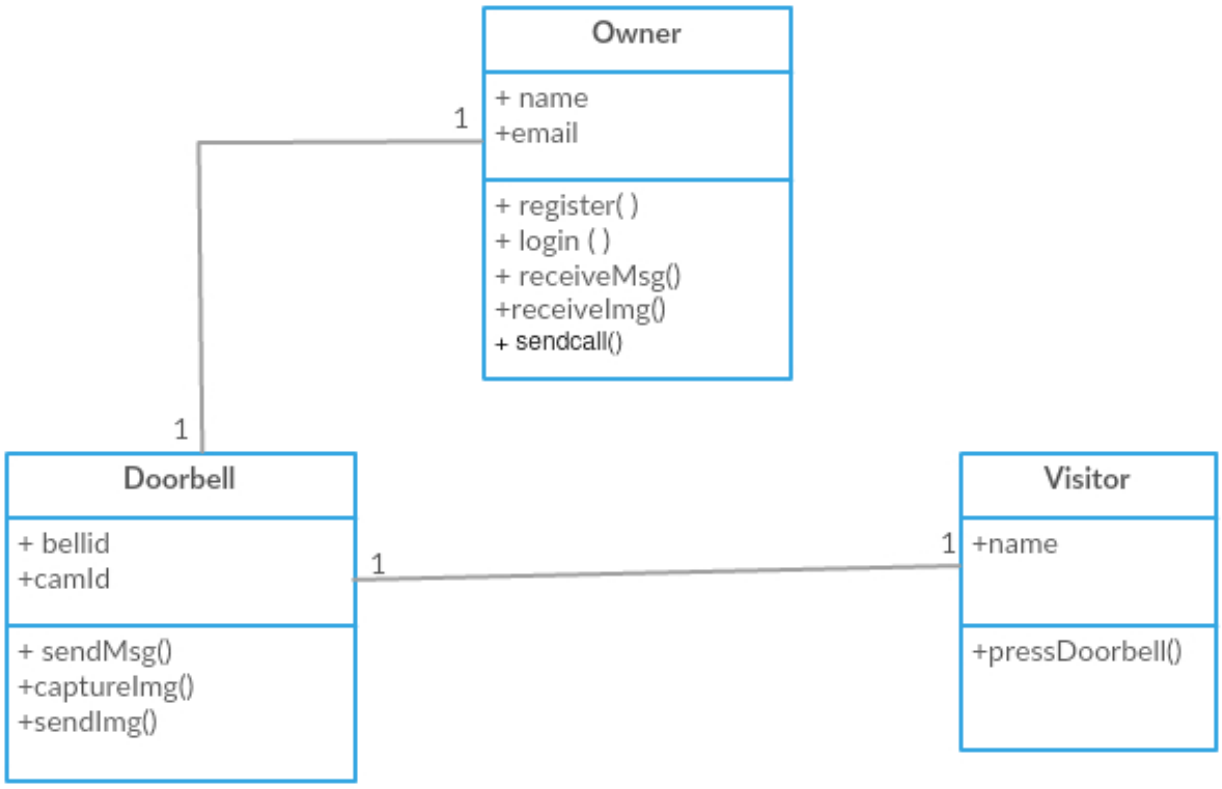


Fig 3.7.1.1: Class Diagram

Owner class is used to receive the message and also an image to the registered mobile number at the time of registration. The Doorbell class is used to send an image to the owner and also an alert message. Visitor can only press the doorbell.

3.7.2 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

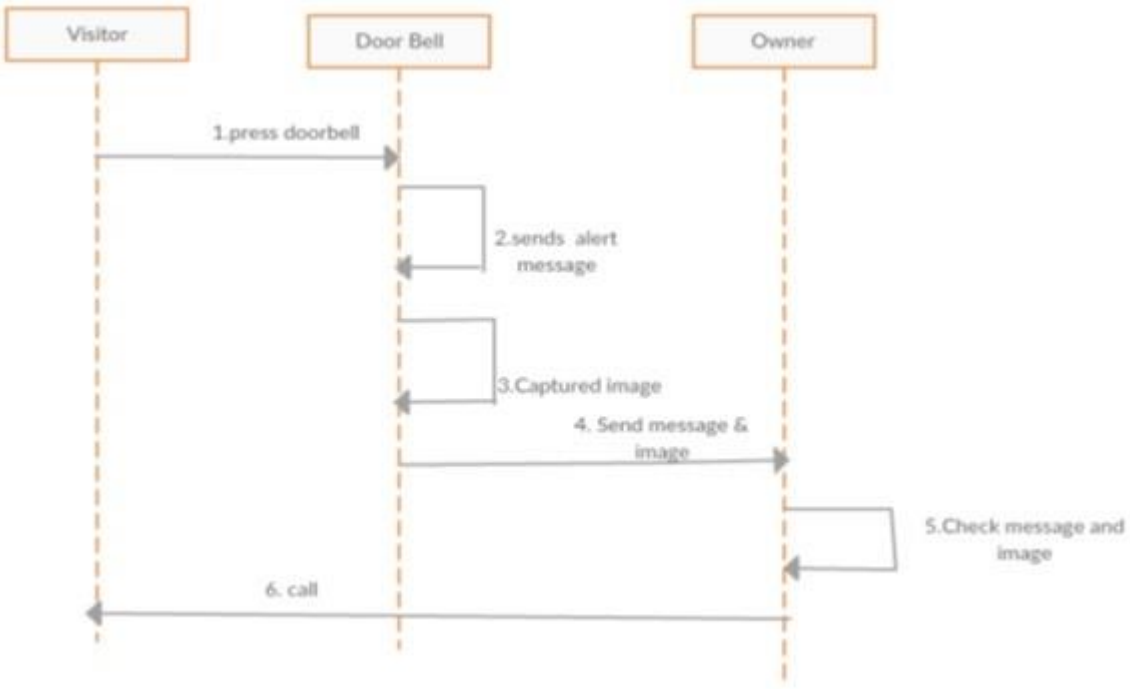


Fig: 3.7.2.1 Sequence Diagram

Initially the visitor should press the doorbell, then camera will capture the image and is sent to the app and also stores the images in the Google drive. We also send an alert message when the doorbell is pressed. When the owner receives the message he will be redirected to the app/server. He can call to the visitor if required.

CHAPTER-4
IMPLEMENTATION

4.1 TOOLS USED

HARD WARE

- Arm-7 controller-3.3v
- GSM Modem-5v
- Voltage convertor-16/2
- Adapter-1 amp
- Jumper Wires
- Smart Phone
- Liquid crystal display

SOFTWARE

- Operating system : Windows XP/7.
- Coding Language : JAVA,C
- Android studio
- MATLAB V9

4.1.1 ARM7 LPC2148 Microcontroller

ARM7 is one of the widely used micro-controller family in embedded system application. This section is humble effort for explaining basic features of ARM-7.**ARM** is a family of instruction set architectures for computer processors based on a reduced instruction set computing (RISC) architecture developed by British company ARM Holdings.

A RISC-based computer design approach means ARM processors require significantly fewer transistors than typical processors in average computers. This approach reduces costs, heat and power use. These are desirable traits for light, portable, battery-powered devices—including smart phones, laptops, tablet and notepad computers), and other embedded systems. A simpler design facilitates more efficient multi-core CPUs and

higher core counts at lower cost, providing higher processing power and improved energy efficiency for servers and supercomputers.

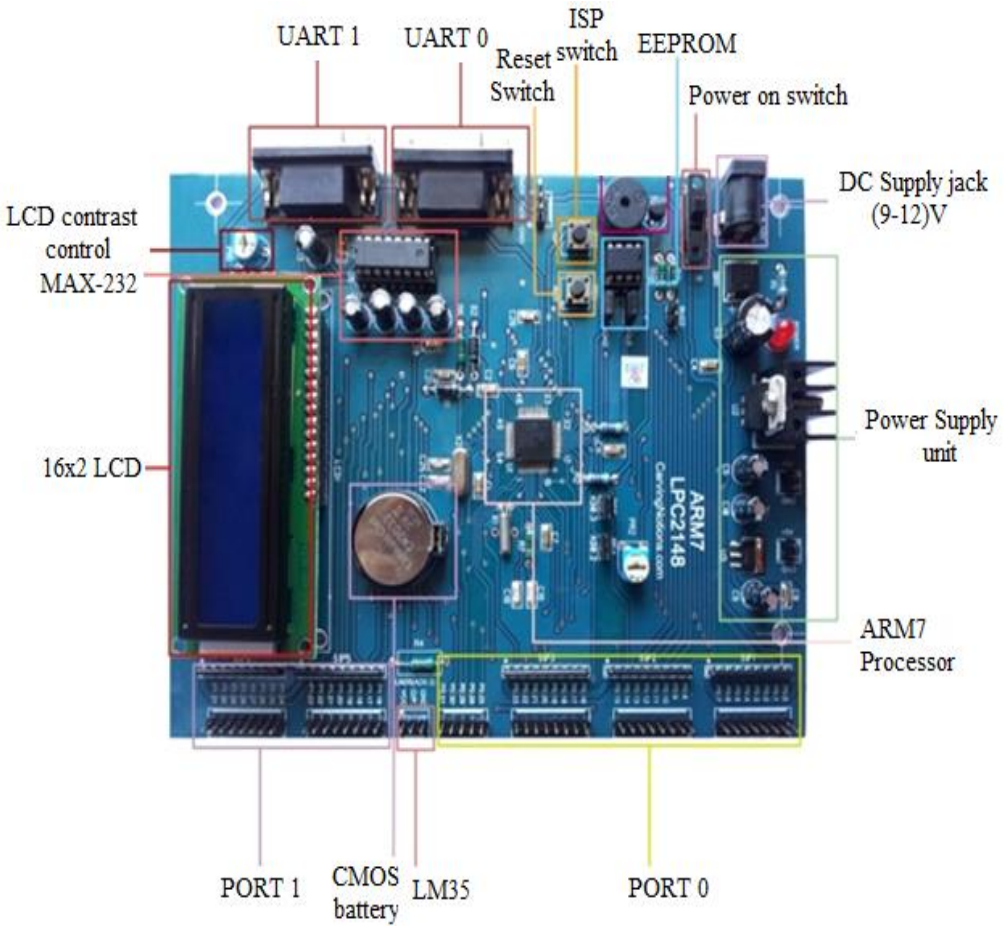


Fig 4.1.1 ARM7 LPC2148 Micro controller

ARM7 is most successful and widely used processor family in embedded system applications. So we have decided to choose ARM7 TDMI based NXP controller LPC2148. Also, ARM7 is a balance between classic and new Cortex series. ARM7 is excellent to get start with in terms of resources available on internet and quality documentation provided by NXP. It suits perfectly for beginners to get in-depth idea about hardware and software implementation.

LPC2148 is manufactured by NXP Semiconductor (Phillips) and it is preloaded with many in-built features and peripherals. This makes it more efficient and reliable choice for

an high-end application developer. I don't want to repeat list of features from User Manual. I request you to hold user manual while reading these tutorials. As we move further all details are discussed and almost all features are used throughout the series. The following is the pin diagram of LPC2148.

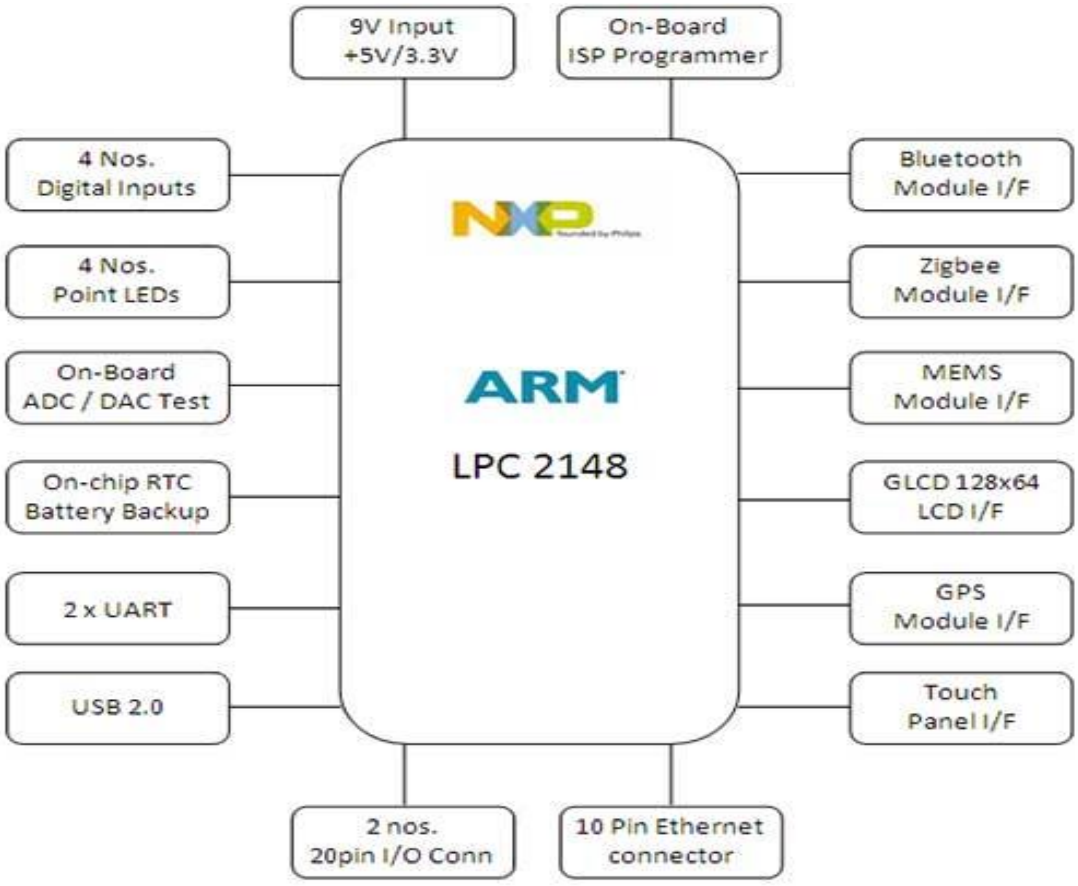


Fig: 4.1.2 Pin diagram of ARM7 LPC2148

MEMORY

LPC2148 has 32kB on chip SRAM and 512kB on chip FLASH memory. This chip has built in support up to 2kB end point USB RAM. This memory is more than enough for almost all applications. Let's understand the function of this huge memory space in LPC2148.

FLASH Memory System

The LPC2148 has 512kB flash memory. This memory may used for both code and data storage. The flash memory can be programmed by various ways

- Using serial built in **JTAG Interface**
- Using In-System Programming **(ISP)**
- By means of In-Application Programming **(IAP)** capabilities

The application program, using IAP functions may also erase and/or program the FLASH while the application is running. When the LPC2148 on chip boot loader is used, 500kB of flash memory is available for user code.

RAM Memory System

LPC2148 provides 32kB of static RAM which may be used for code and/or data storage. It may be accessed as 8-bit, 16-bit and 32-bits.

INPUT/OUTPUT PORTS (GPIO of LPC2148)

Understanding what is IO Ports and how to use them is very important. It's because when we see microchip, we'll find a black box i.e. IC with some pins. LPC2148 has two IO ports each of 32-bit wide, provided by 64 IO pins.

Technical Specifications of the ARM7 Board

- Processor : LPC2148
- Clock speed : 11.0592 MHz / 22.1184 MHz
- Clock Divisors : 6 (or) 12
- Real time Clock : DS1307 on I2C Bus /w Battery
- Data Memory : 24LCxx on I2C Bus
- LCD : 16x2 Backlit
- RS-232 : +9V, -9V levels
- Power : 7-15V AC/DC @ 500 mA
- Voltage Regulator : 5V On-board LM7805

PORT 0 is a 32-bit I/O port with individual direction controls for each bit. Total of 28 pins of the Port 0 can be used as a general purpose bi-directional digital I/Os while P0.31 provides digital output functions only. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pins P0.24, P0.26 and P0.27 are not available.

PORT 1 is a 32-bit bi-directional I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 0 through 15 of port 1 are not available.

PORT PINS	TYPE	DESCRIPTION
P0.0-P0.31 P1.16-P1.31	Input/output	General purpose input/output. The number of GPIOs actually available depends on the use of alternate functions.
P0.0/TXD0/ PWM1	I/o O O	**P0.0 —** General purpose input/output digital pin (GPIO). **TXD0 —** Transmitter output for UART0. **PWM1 —** Pulse Width Modulator output 1.
VSS 6, 18, 25, 42,50	I	**Ground:** 0 V reference.
VSSA 59	I	**Analog ground:** 0 V reference. This should nominally be the same voltage as VSS, but should be isolated to minimize noise and error.
VREF 63	I	ADC reference: This should be nominally less than or equal to the VDD voltage but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC(s) and DAC.

Table 4.1.1.1 Port Description

4.1.2 Liquid Crystal Display

A liquid crystal display (LCD) is a thin, flat panel used for electronically displaying information such as text, images, and moving pictures. Its uses include monitors for computers, televisions, instrument panels, and other devices ranging from aircraft cockpit displays, to every-day consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones.

LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segmentdisplays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters, the axes of transmission of which are (in most of the cases) perpendicular to each other. Without the liquid crystal between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer.

Among its major features are its lightweight construction, its portability, and its ability to be produced in much larger screen sizes than are practical for the construction of cathode ray tube (CRT) display technology. Its low electrical power consumption enables it to be used in battery-powered electronic equipment. It is an electronically-modulated optical device made up of any number of pixels filled with liquid crystals and arrayed in front of a light source (backlight) or reflector to produce images in colour or monochrome.

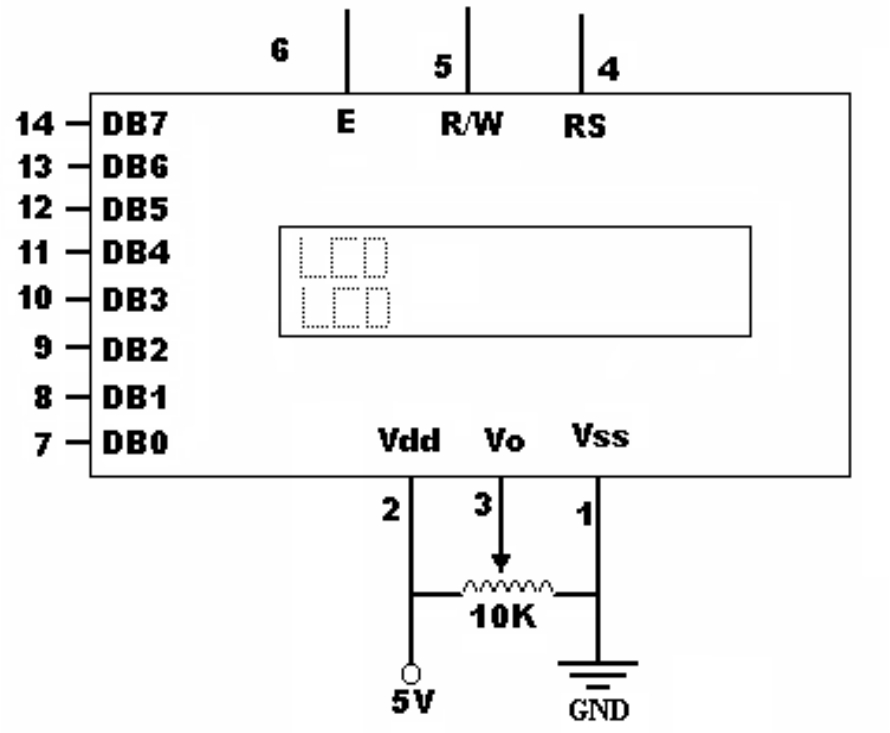


Fig 4.1.2.1 Pin description of LCD

The description of the different pins of the liquid crystal, display is shown in the figure 3.6. A **16x4 LCD** means it can display 16 characters per line and there are 4 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The LCD has 14 pins out of which 8 pins are data pins and the remaining are the enable, reset, read/write and power supply pins. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Pin description and their functions of LCD

PIN	SYMBOL	I/O	DESCRIPTION
1	VSS	--	Ground
2	VCC	--	+5V power supply
3	VEE	--	Power supply to control contrast
4	RS	I	RS=0 to select command register RS=1 to select data register
5	R/W	I	R/W=0 for write R/W=1 for read
6	EN	I/O	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

Table 4.2.1.1: Pin description and their functions of LCD

4.1.3 Keil Software

Keil an ARM Company makes C compilers, macro assemblers, real-time kernels, debuggers, simulators, integrated environments, evaluation boards, and emulators for ARM7/ARM9/Cortex-M3, XC16x/C16x/ST10, 251, and 8051 MCU families.

Keil development tools for the 8051 Microcontroller Architecture support every level of software developer from the professional applications engineer to the student just learning about embedded software development. When starting a new project, simply select the microcontroller you use from the Device Database and the µVision IDE sets all compiler, assembler, linker, and memory options for you. Keil is a cross compiler. So first we have to understand the concept of compilers and cross compilers. After then we shall learn how to work with keil.

Creating Your Own Application in µVision2

To create a new project in µVision2, you must:

1. Select Project - New Project.
2. Select a directory and enter the name of the project file.
3. Select Project - Select Device and select an 8051, 251, or C16x/ST10 device from the Device Database.
4. Create source files to add to the project.
5. Select Project - Targets, Groups, and Files. Add/Files, select Source Group1, and add the source files to the project.
6. Select Project - Options and set the tool options. Note when you select the target device from the Device Database all special options are set automatically. You typically only need to configure the memory map of your target hardware. Default memory model settings are optimal for most applications.
7. Select Project - Rebuild all target files or Build target.

EMBEDDED C

Use of embedded processors in passenger cars, mobile phones, medical equipment, aerospace systems and defence systems is widespread, and even everyday domestic appliances such as dish washers, televisions, washing machines and video recorders now include at least one such device.

Because most embedded projects have severe cost constraints, they tend to use low-cost processors like the 8051 family of devices considered in this book. These popular chips have very limited resources available most such devices have around 256 bytes (not megabytes!) of RAM, and the available processor power is around 1000 times less than that of a desktop processor. As a result, developing embedded software presents significant new challenges, even for experienced desktop programmers. If you have some programming experience - in C, C++ or Java - then this book and its accompanying CD will help make your move to the embedded world as quick and painless as possible.

4.1.4 MATLAB

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is a high-performance language for technical computing. It integrates *computation*, *visualization*, and *programming* environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated *data structures*, contains built-in editing and *debugging tools*, and supports *object-oriented programming*. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems.

MATLAB is an interactive system whose basic data element is an *array* that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful *built-in* routines that enable a very wide variety of computations. It also has

easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as *toolbox*. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other elders of applied science and engineering.

4.1.4.1 Starting MATLAB

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut *icon* (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains *other* windows. The major tools within or accessible from the desktop are:

- The Command Window
- The Command History
- The Workspace
- The Current Directory
- The Help Browser
- The Start button

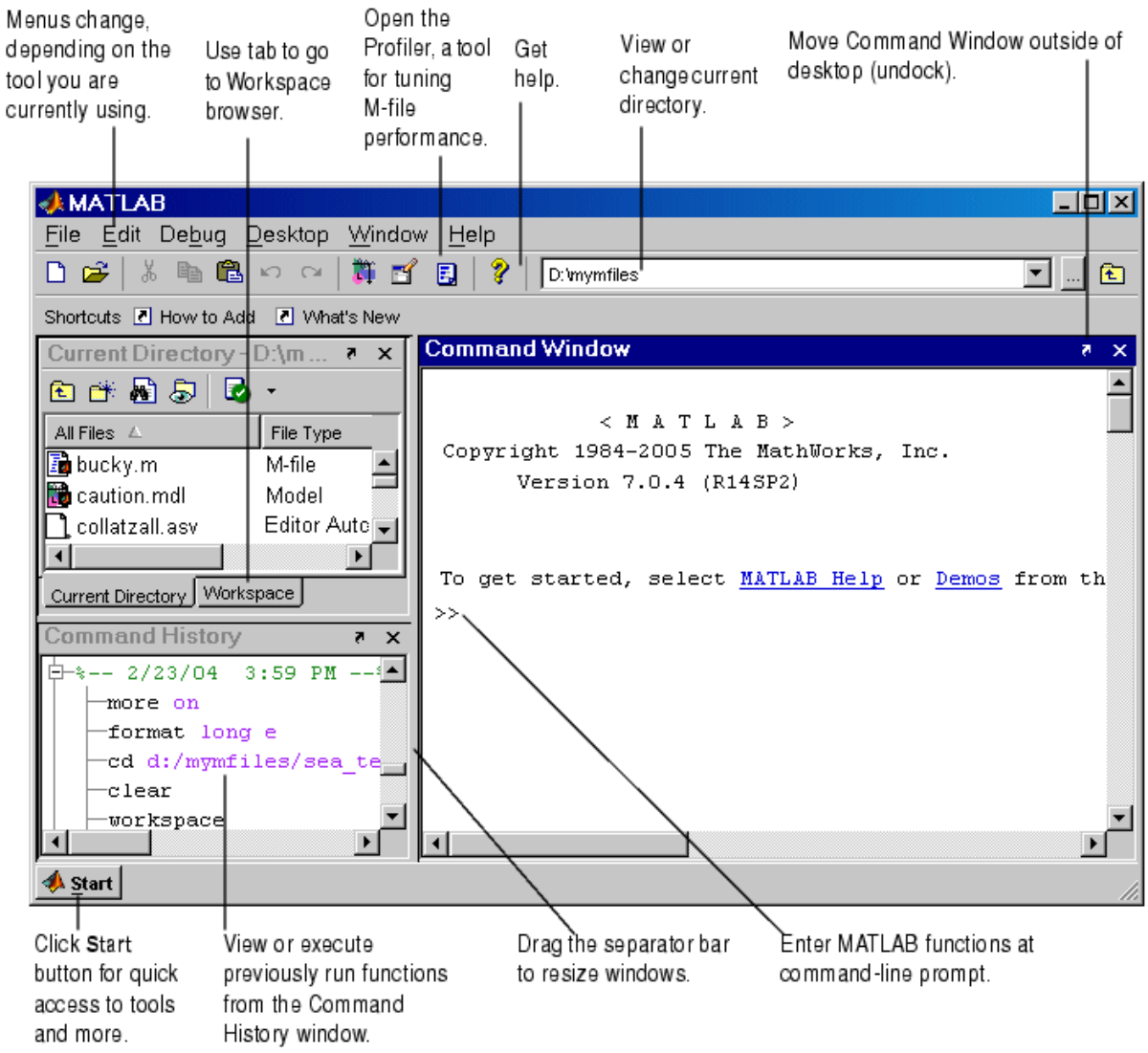


Fig 4.1.4.1.1 Graphical interface for MATLAB Workspace

When MATLAB is started for the first time, the screen looks like the one that shown in the Figure 4.1.4.1.1 this illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs. Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run. You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt:

>> for full version

EDU> for educational version

4.1.4.2 Error messages

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, *, in the following expression

```
>> x = 10;
>> 5x
??? 5x
Error: Unexpected MATLAB expression.
```

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. The preferred method is to use the *Help Browser*. The Help Browser can be started by selecting the ? icon from the desktop toolbar. On the other hand, information about any command is available by typing

```
>> help Command
```

Another way to get help is to use the lookfor command. The lookfor command differs from the help command. The help command searches for an exact function name match, while the lookfor command searches the quick summary information in each function for a match. For example, suppose that we were looking for a function to take *the inverse of a matrix*. Since MATLAB does not have a function named inverse, the command help inverse will produce nothing. On the other hand, the command lookfor inverse will produce detailed information, which includes the function of interest, inv.

```
>> lookfor inverse
```

Note - At this particular time of our study, it is important to emphasize one main point. Because MATLAB is a huge program; it is impossible to cover *all the details* of each function one by one. However, we will give you information how to get help. Here are some examples: Use on-line help to request info on a specific function

```
>> help sqrt
```

In the current version (MATLAB version 7), the doc function opens the on-line version of the help manual. This is very helpful for more complex commands.

```
>> doc plot
```

Use lookfor to and functions by keywords. The general form is

```
>> lookfor FunctionName
```

4.1.5 GSM

GSM also pioneered a low-cost, to the network carrier, alternative to voice calls, the Short message service (SMS, also called "text messaging"), which is now supported on other mobile standards as well. Another advantage is that the standard includes one worldwide Emergency telephone number, 112. This makes it easier for international travellers to connect to emergency services without knowing the local emergency number.

GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS).

GSM (Global System for Mobile communication) is a digital mobile telephony system that is widely used in Europe and other parts of the world. GSM uses a variation of time division multiple access (TDMA) and is the most widely used of the three digital wireless telephony technologies (TDMA, GSM, and CDMA). GSM digitizes and compresses data, then sends it down a channel with two other streams of user data, each in its own time slot. It operates at either the 900 MHz or 1800 MHz frequency band.

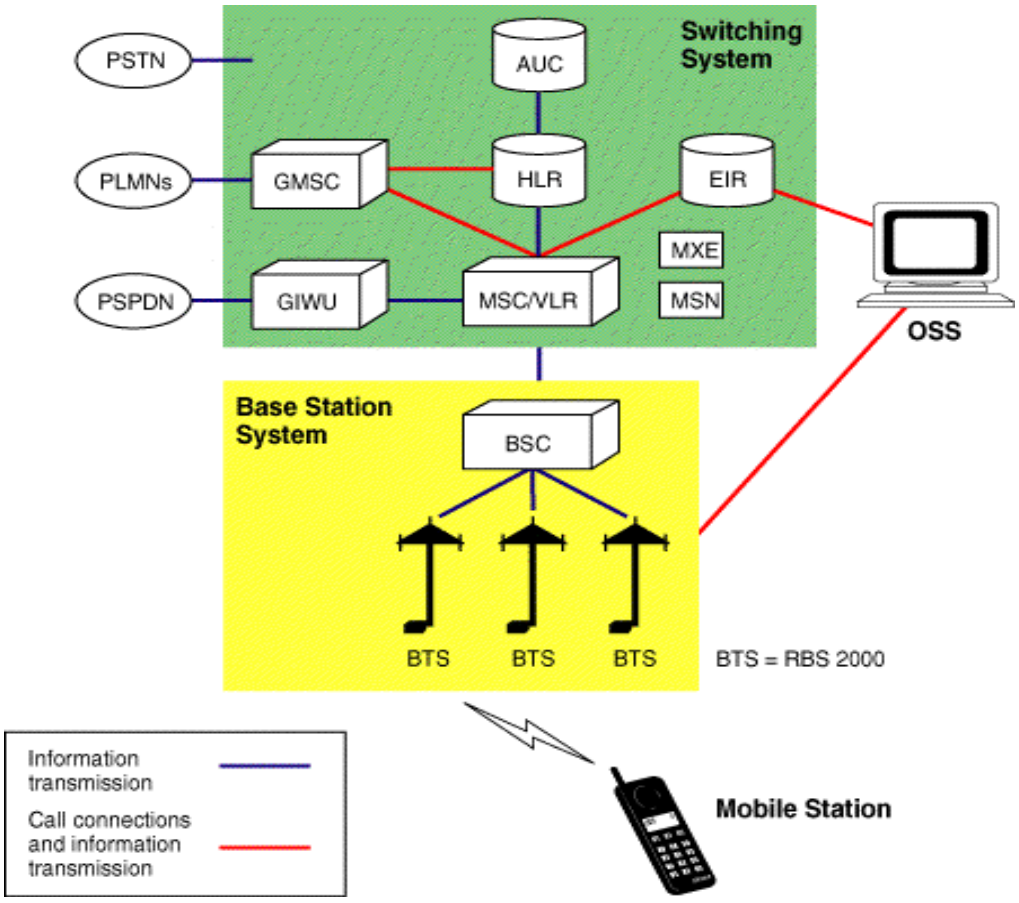


Fig 4.1.5.1 GSM Architecture

The Switching System:

The switching system (SS) is responsible for performing call processing and subscriber-related functions. The switching system includes the following functional units.

- ***Home location register (HLR)****:* The HLR is a database used for storage and management of subscriptions. The HLR is considered the most important database, as it stores permanent data about subscribers, including a subscriber's service profile, location information, and activity status. When an individual buys a subscription from one of the PCS operators, he or she is registered in the HLR of that operator.

- ***Mobile services switching centre (MSC):*** The MSC performs the telephony switching functions of the system. It controls calls to and from other telephone and data systems. It also performs such functions as toll ticketing, network interfacing, common channel signalling, and others.
- ***Visitor location register (VLR)****:* The VLR is a database that contains temporary information about subscribers that is needed by the MSC in order to service visiting subscribers. The VLR is always integrated with the MSC. When a mobile station roams into a new MSC area, the VLR connected to that MSC will request data about the mobile station from the HLR. Later, if the mobile station makes a call, the VLR will have the information needed for call setup without having to interrogate the HLR each time.
- ***Authentication centre (AUC)****:* A unit called the AUC provides authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call. The AUC protects network operators from different types of fraud found in today's cellular world.
- ***Equipment identity register (EIR)****:* The EIR is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized, or defective mobile stations. The AUC and EIR are implemented as stand-alone nodes or as a combined AUC/EIR node.

The Base Station System (BSS):

All radio-related functions are performed in the BSS, which consists of base station controllers (BSCs) and the base transceiver stations (BTSs).

- ***BSC****:* The BSC provides all the control functions and physical links between the MSC and BTS. It is a high-capacity switch that provides functions such as handover, cell configuration data, and control of radio frequency (RF) power levels in base transceiver stations. A number of BSCs are served by an MSC.
- ***BTS****:* The BTS handles the radio interface to the mobile station. The BTS is the radio equipment (transceivers and antennas) needed to service each cell in the network. A group of BTSs are controlled by a BSC.

The Operation and Support System

The operations and maintenance centre (OMC) is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralized, regional and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organizations.

4.1.6 BUZZER

A **buzzer** or **beeper** is a signalling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows.

It most commonly consists of a number of switches or sensors connected to a control unit that determines if and which button was pushed or a preset time has lapsed, and usually illuminates a light on the appropriate button or control panel, and sounds a warning in the form of a continuous or intermittent buzzing or beeping sound. Initially this device was based on an electromechanical system which was identical to an electric bell without the metal gong.

Often these units were anchored to a wall or ceiling and used the ceiling or wall as a sounding board. Another implementation with some AC-connected devices was to implement a circuit to make the AC current into a noise loud enough to drive a loudspeaker and hook this circuit up to a cheap 8-ohm speaker. Nowadays, it is more popular to use a ceramic-based piezoelectric sounder like a Son alert which makes a high-pitched tone. Usually these were hooked up to "driver" circuits which varied the pitch of the sound or pulsed the sound on and off.

In game shows it is also known as a "lockout system," because when one person signals ("buzzes in"), all others are locked out from signalling. Several game shows have large buzzer buttons which are identified as "plungers".

The word "buzzer" comes from the rasping noise that buzzers made when they were electromechanical devices, operated from stepped-down AC line voltage at 50 or 60 cycles. Other sounds commonly used to indicate that a button has been pressed are a ring or a beep.

4.1.7 Regulator

Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

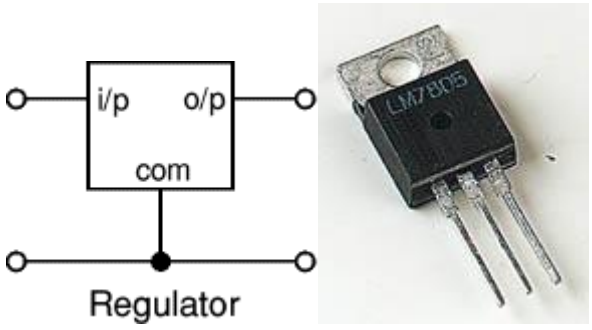


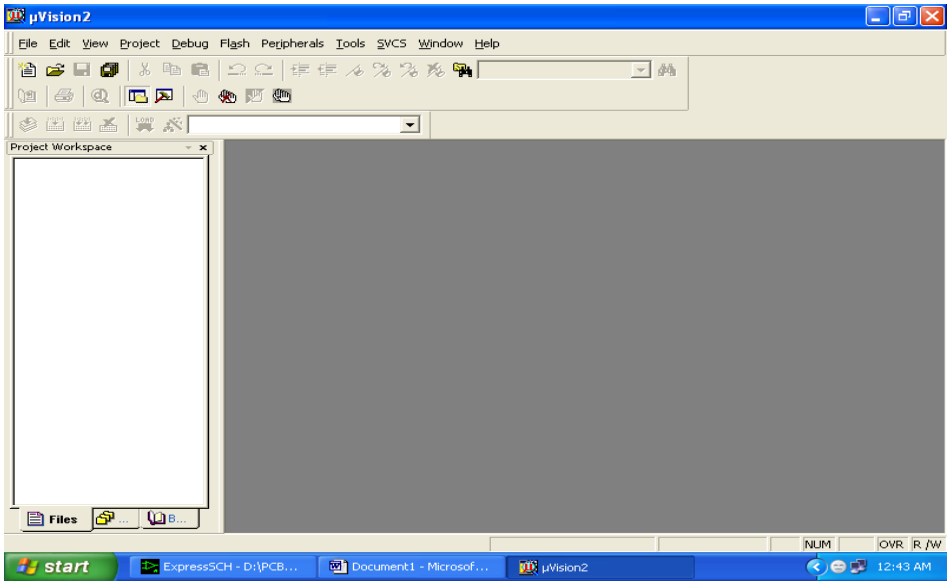
Fig 4.1.7.1 Regulator

Many of the fixed voltage regulator ICs have 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. They include a hole for attaching a heat sink if necessary.

4.1.8 Compiling of Software's

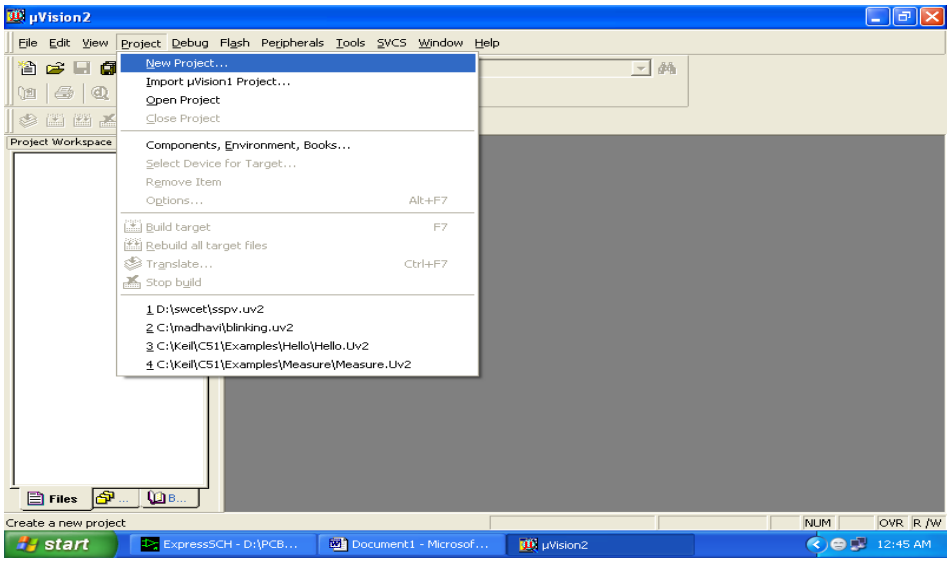
1. Click on the Keil Vision Icon on Desktop

2. The following fig will appear

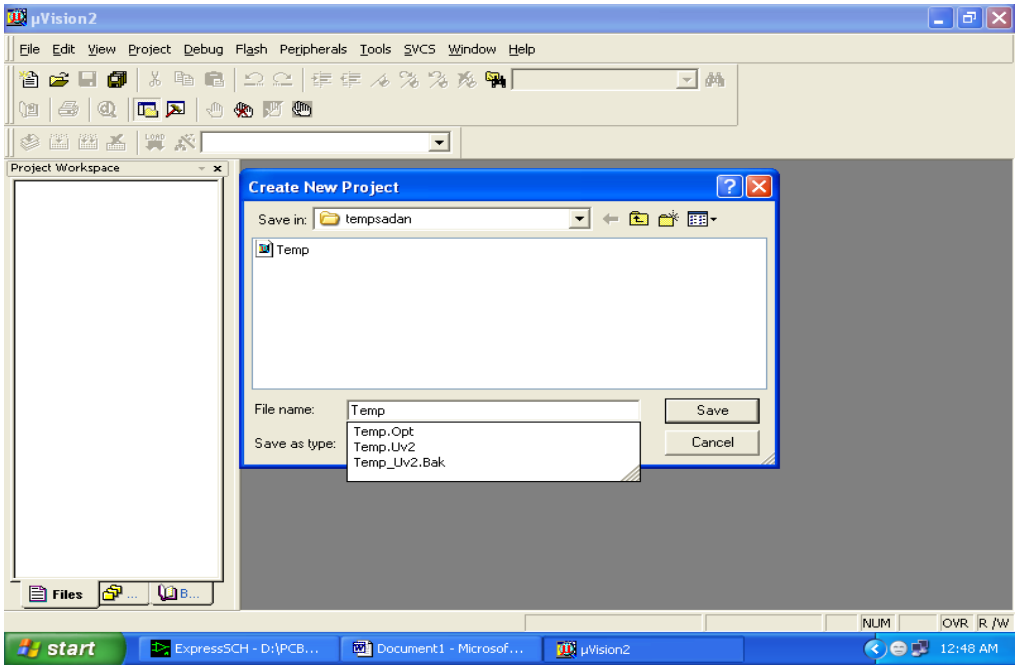


3. Click on the Project menu from the title bar

4. Then Click on New Project



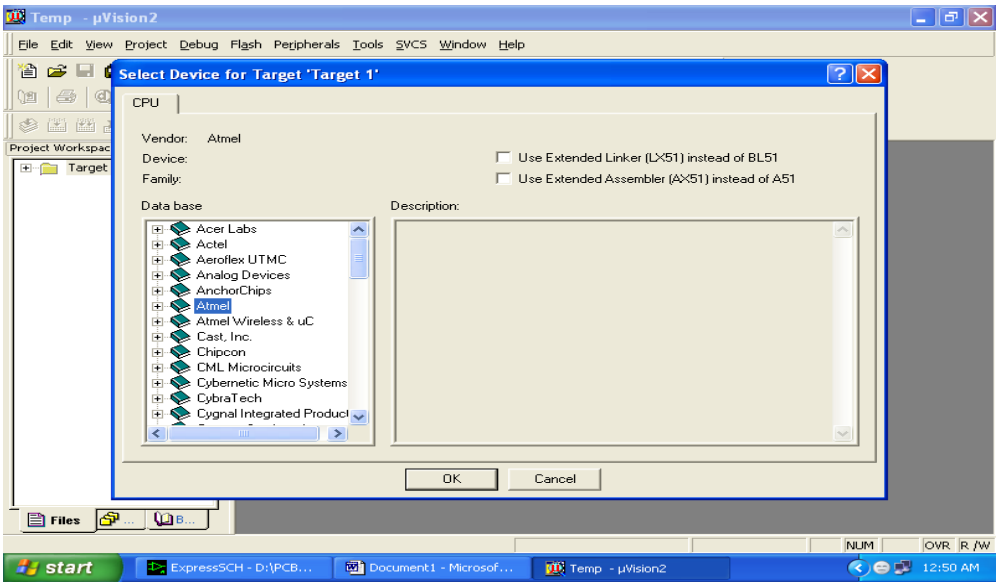
5. Save the Project by typing suitable project name with no extension in your own folder sited in either C:\ or D:\



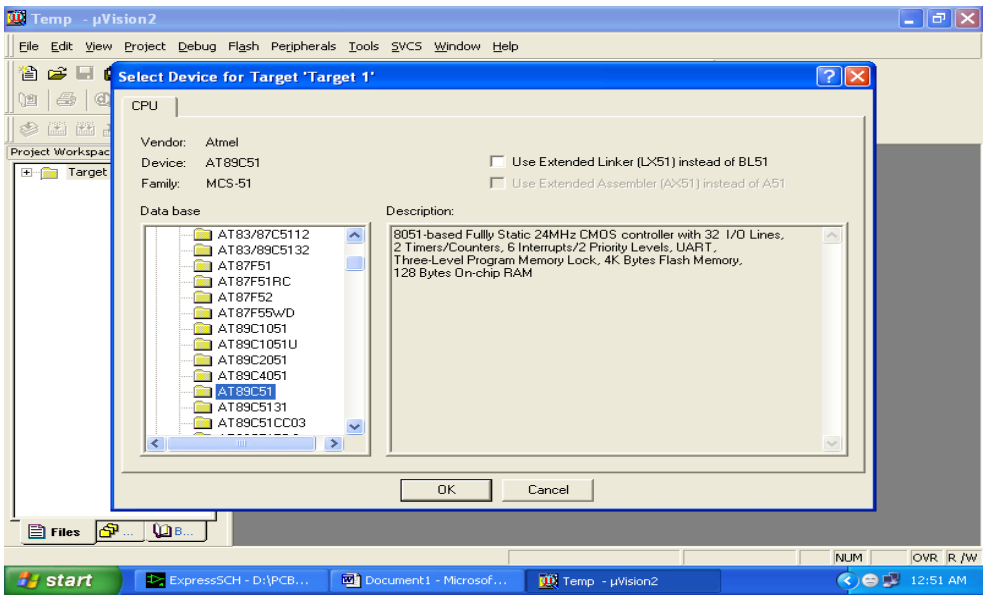
6. Then Click on Save button above.

7. Select the component for u r project. i.e. Atmel……

8. Click on the + Symbol beside of Atmel

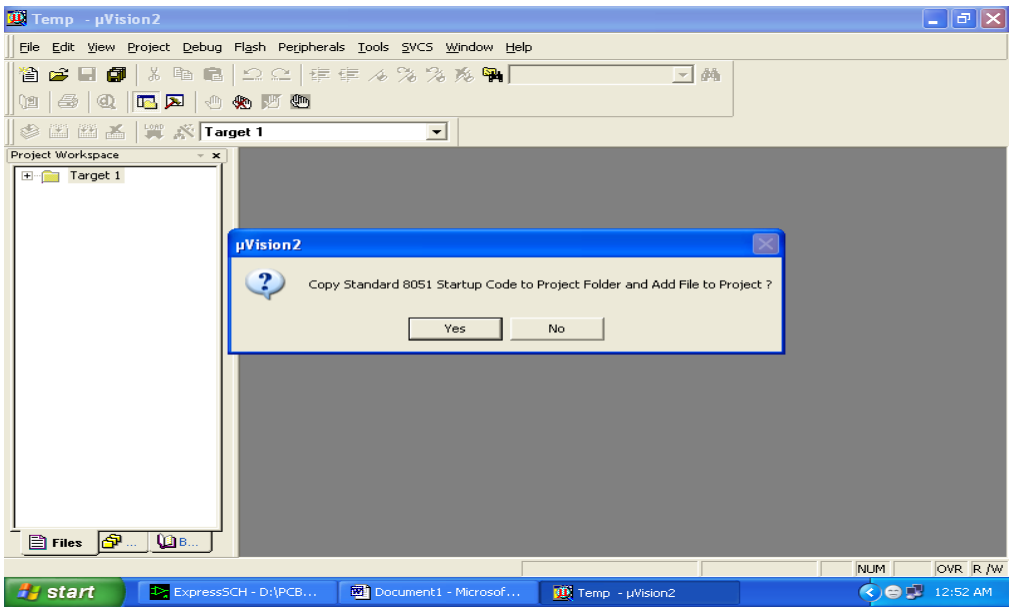


9. Select AT89C51 as shown below



10. Then Click on "OK"

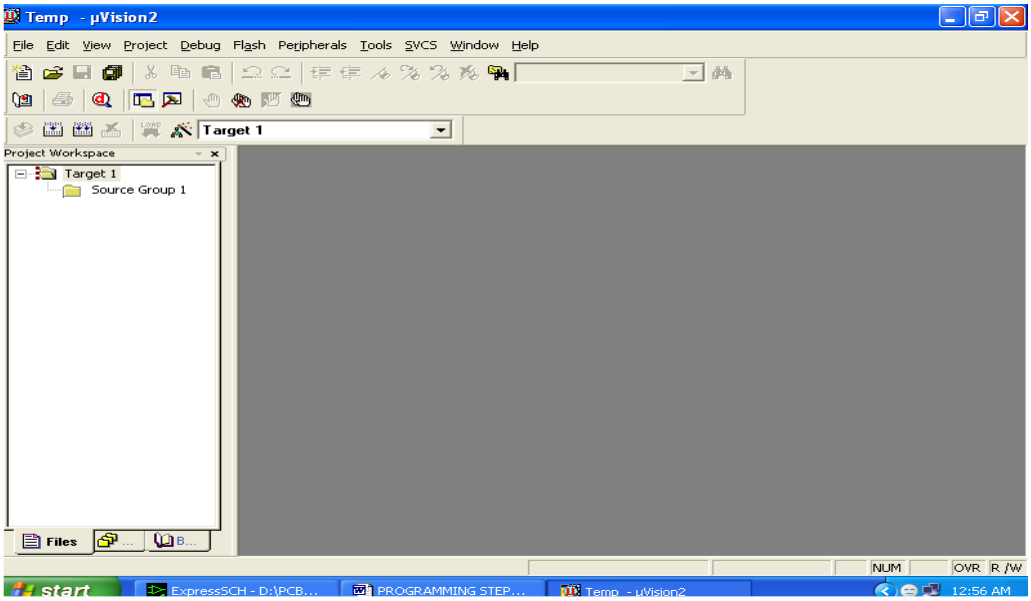
11. The Following fig will appear



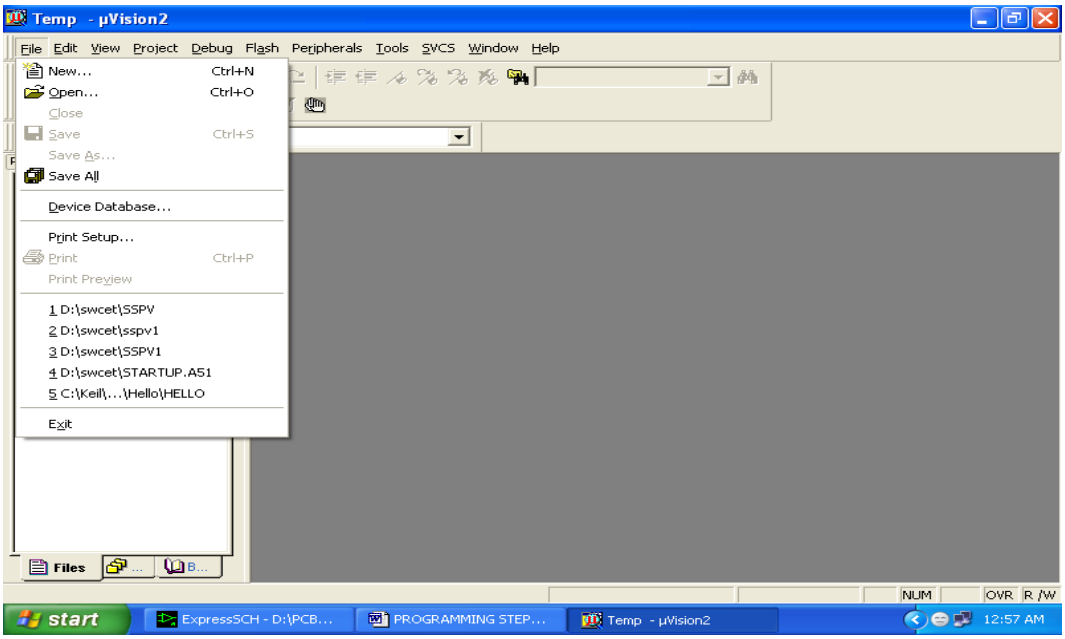
12. Then Click either YES or NO……mostly “NO”.

13. Now your project is ready to USE.

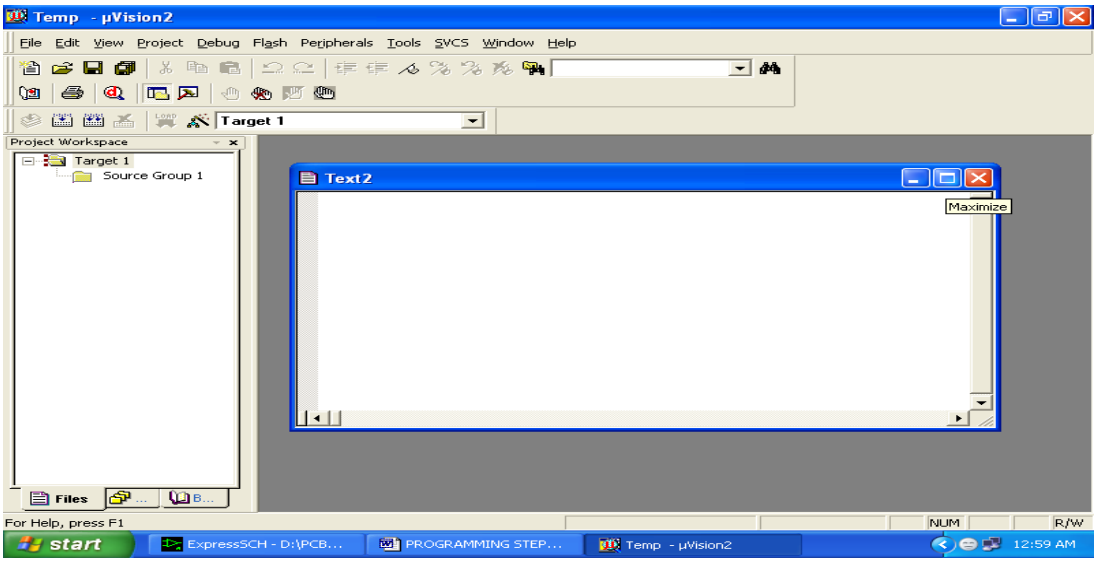
14. Now double click on the Target1, you would get another option “Source group 1” as shown in next page.



15. Click on the file option from menu bar and select “new”.

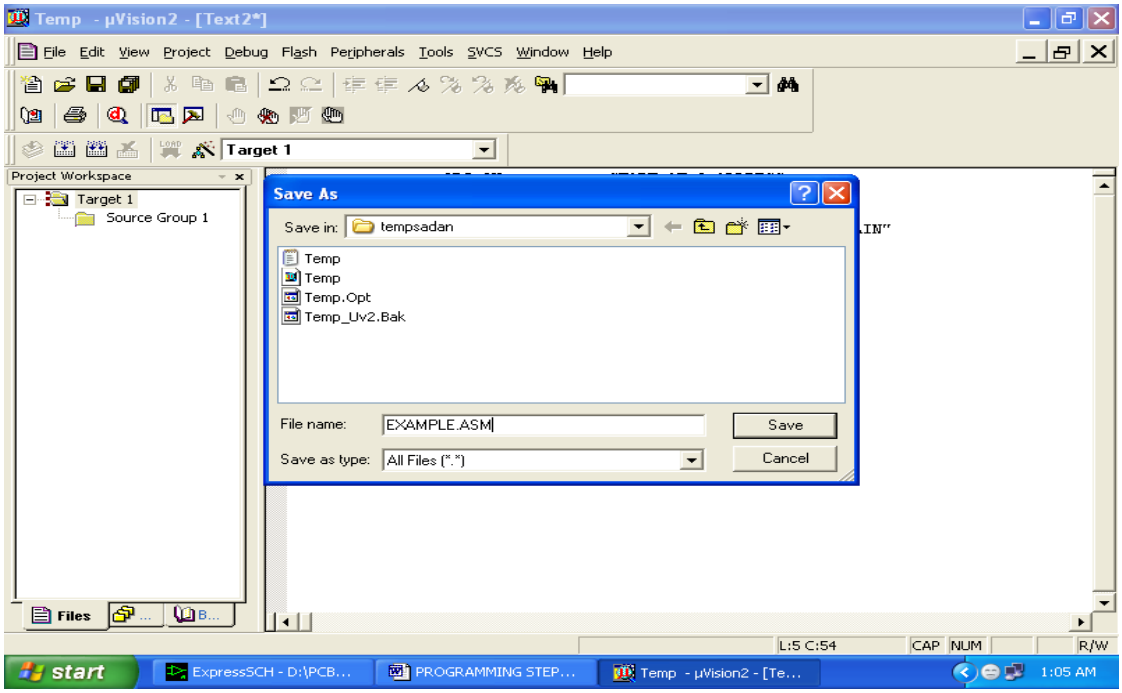


16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.

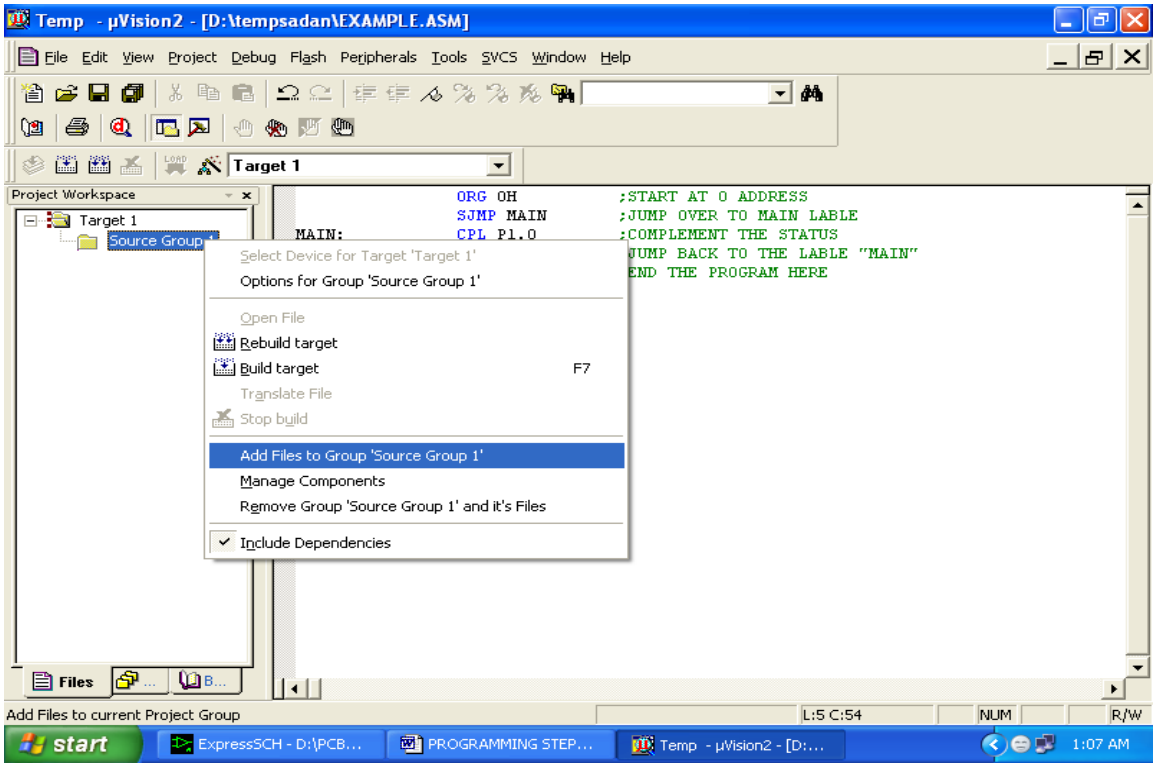


17. Now start writing program in either in "EMBEDDED C" or "ASM".

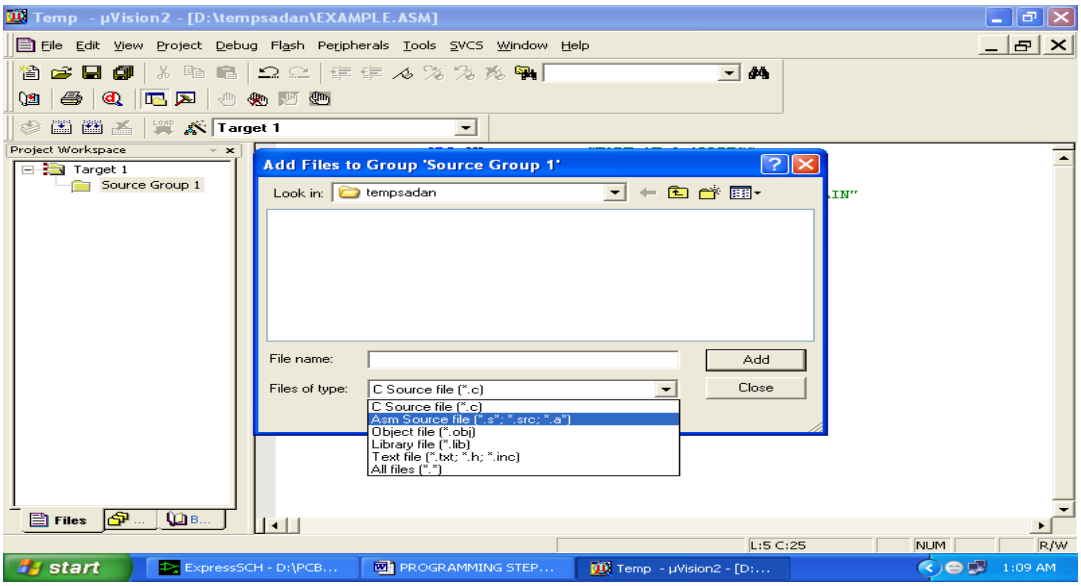
18. For a program written in Assembly, then save it with extension ". asm" and for "EMBEDDED C" based program save it with extension " .C"



19. Now right click on Source group 1 and click on “Add files to Group Source”.



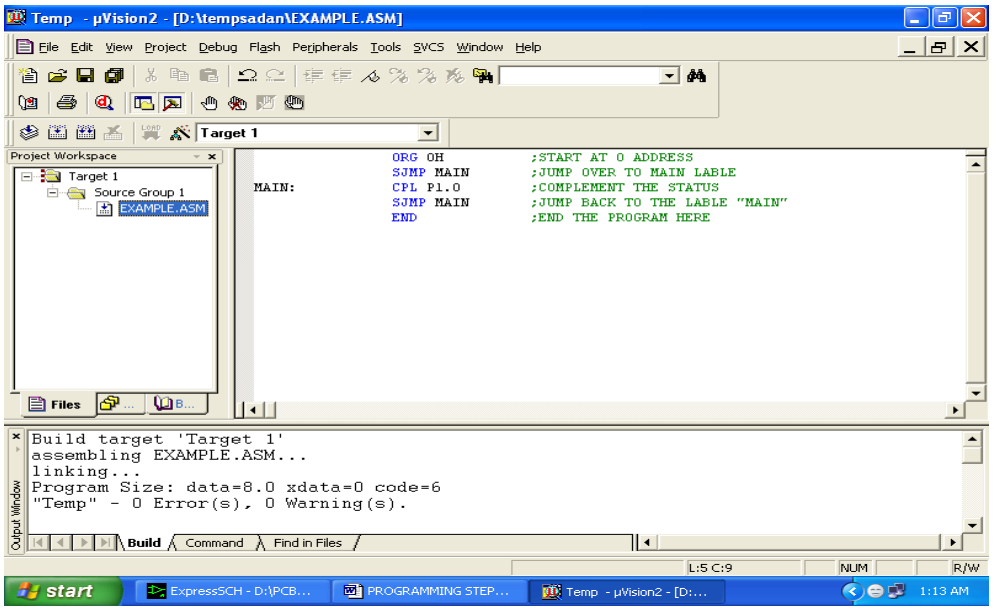
20. Now you will get another window, on which by default “EMBEDDED C” files will appear.



21. Now select as per your file extension given while saving the file

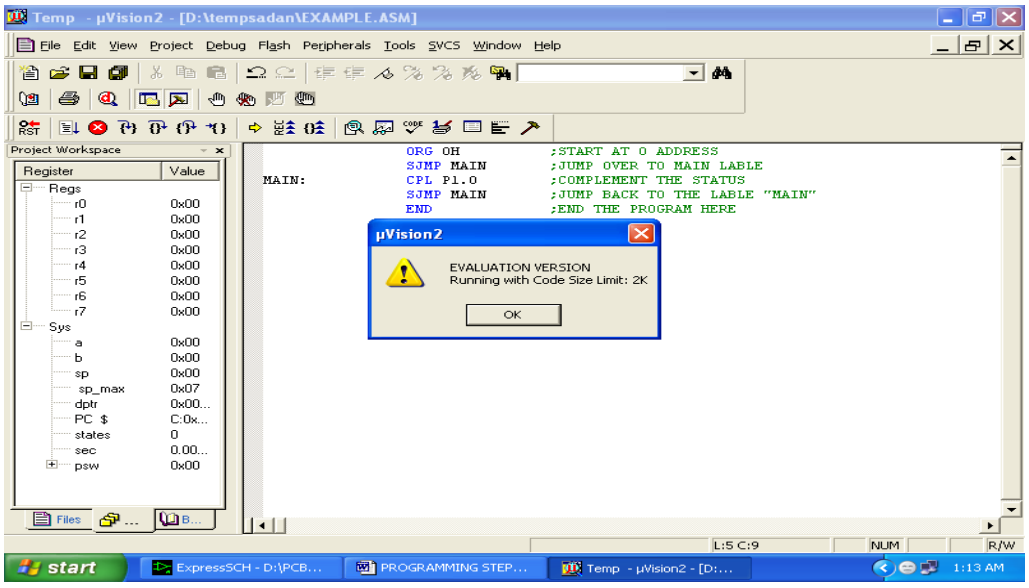
22. Click only one time on option "**ADD**".

23. Now Press function key F7 to compile. Any error will appear if so happen.



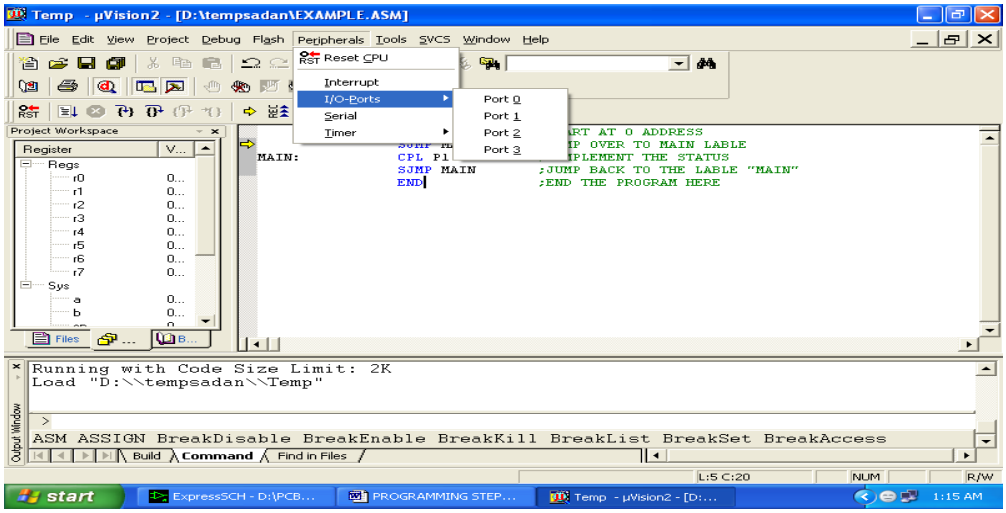
24. If the file contains no error, then press Control+F5 simultaneously.

25. The new window is as follows

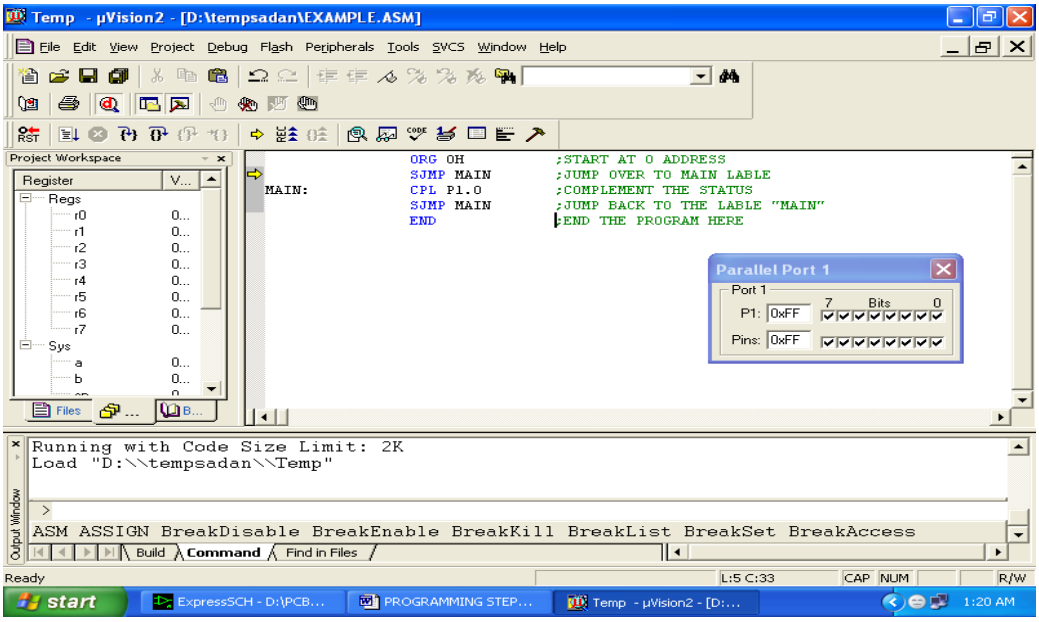


26. Then Click “OK”.

27. Now click on the Peripherals from menu bar, and check your required port as shown in fig below.



28. Drag the port a side and click in the program file.



29. Now keep Pressing function key “F11” slowly and observe.

30. You are running your program successfully.

4.2 COMPONENT DIGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams. It does not describe the functionality of the system but it describes the components used to make those functionalities.

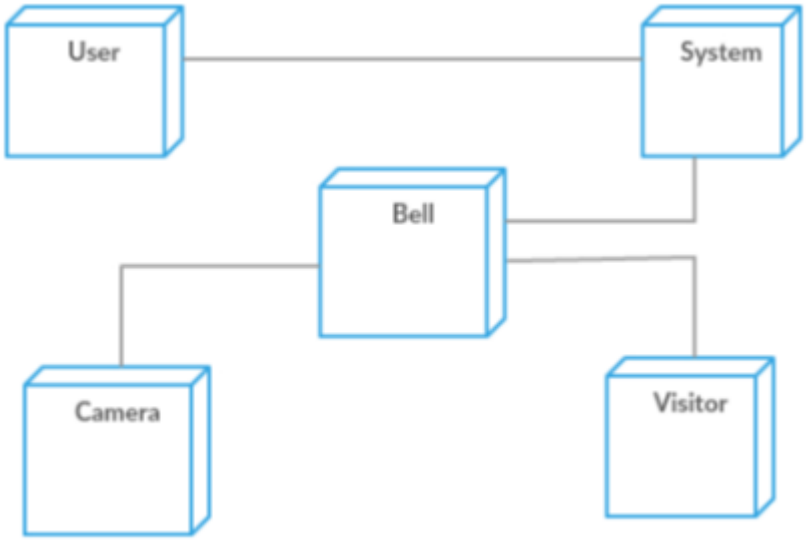


Fig 4.2.1 Component Diagram

The above diagram discuss about the software and hardware requirements that are used in the development of the device.

4.3 PSEUDO CODE/ALGORITHMS

MATLAB CODE

```
while 1==1
   fopen(com);      %opening serial com port
   rec=fscanf(com);  %scaning com port for 10 sec
   fclose(com);         %closing com port
   a=size(rec);         %checking for data receive or not
   if a>0
       count=count+1;                    %no of person count
       image=getsnapshot(vid);        %capturing an image from camera
       face=vision.CascadeObjectDetector;   %creating objective for face recognition tool
       box=step(face,image);                %analing image for face recognition
      imshow(image);
title('SNAP SHOT FROM WEBCAM');
image=insertText(image, [0 0], num2str(round(clock)));   %inserting time and date in the
image
       siz=size(box);   %verifing for face in the image
     if siz>0%and sending the data

             IFaces = insertObjectAnnotation(image, 'rectangle', box, 'Face');   %to the
controller
       imshow(IFaces), title('Detected faces');
   fopen(com);
        fprintf(com,'*detect#');
            fclose(com);
       else
        imshow(image), title('NO FACE DETECTED');
        fopen(com);    %Sending no person status to
        fprintf(com,'*notdtect#');         %controller board
         fclose(com);
```

```
 end
while 1==1
        fopen(com);
         dat=fscanf(com);         %Waiting for perons
         fclose(com);              %leaving time to
                 if dat>0                  %write end time in
            image=insertText(image,[450 0], num2str(round(clock)));        %image
       imshow(image);
            break
         end
      end
   imshow(image);          %displaying image
mkdir('D:',date);          %creating the folder with the name of date
   nm=strcat('C:\Users\Admin\Google Drive\Face capture',date,'pic',num2str(count),'.jpg');
```

Embedded C code:

```
void gsm_msg()
{
  ser_tx1("AT+CMGS=\"");
  delay(50);
  ser_tx1("9493033886\"\r\n");
  delay(5000);
  ser_tx1("Person is waiting. Please check authourization");
  delay(100);
  tx1(0x1a);
  delay(10000);
}
```

4.4 DEPLOYMENT DIAGRAM

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

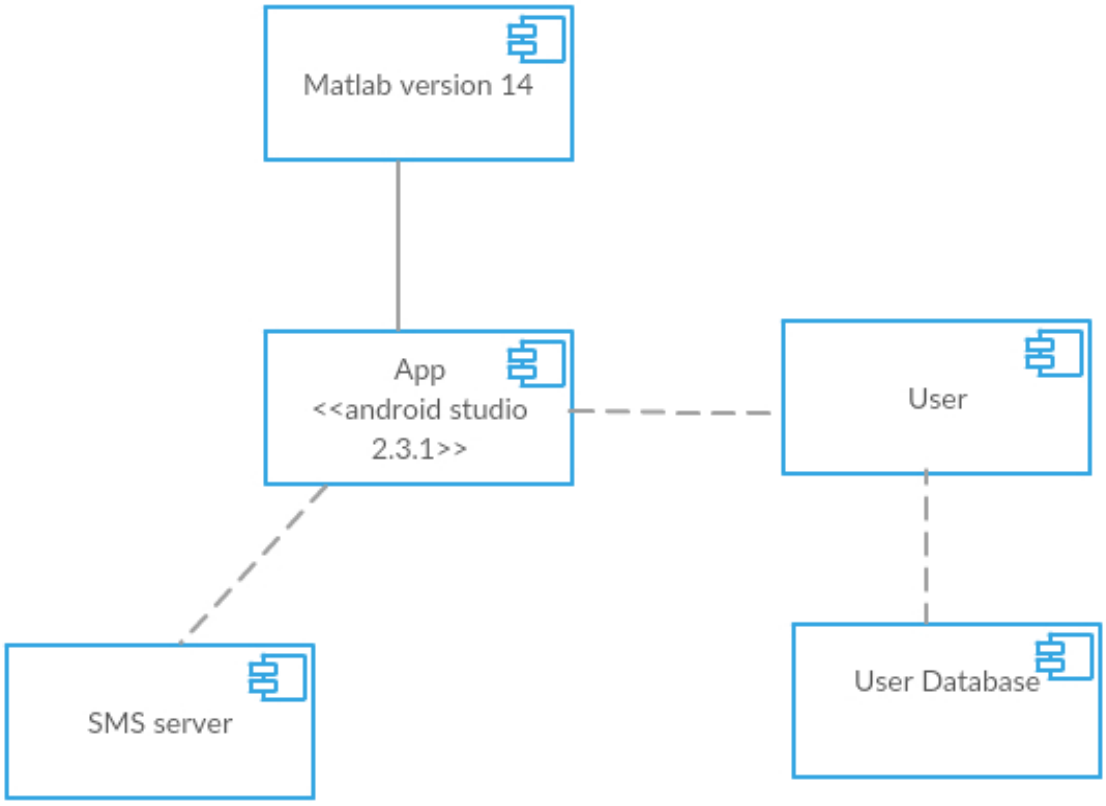


Fig.4.4.1 Deployment Diagram

The above diagram deals with the software components used in the development of the device

4.5 SCREENSHOTS

Initially visitor should press the doorbell.

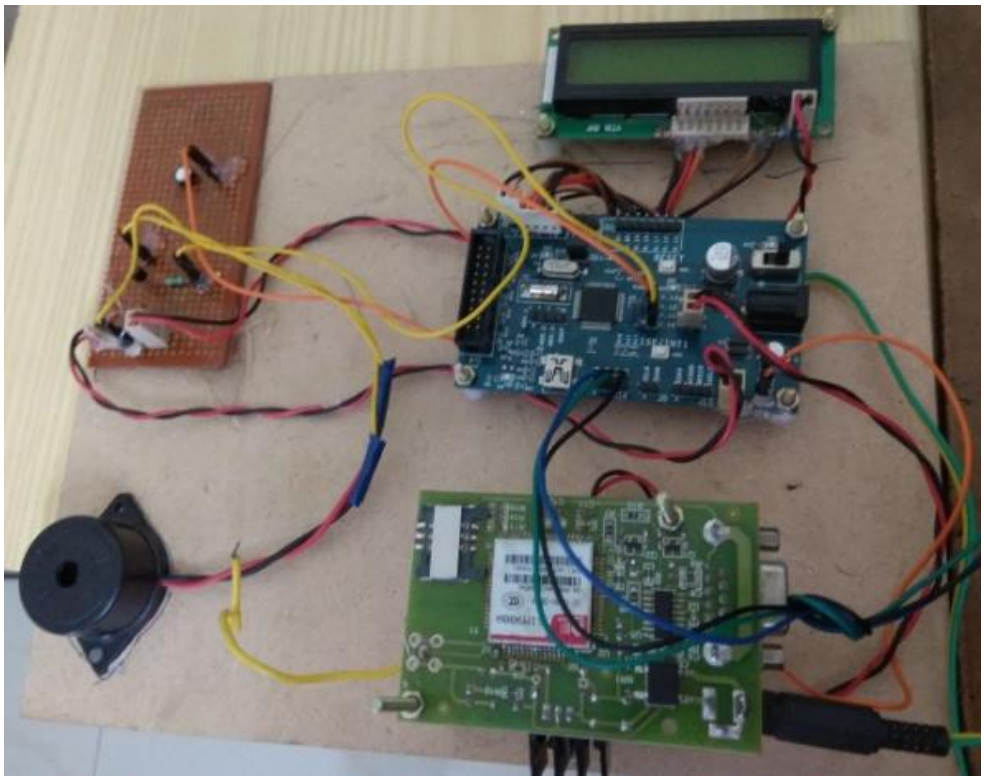


Fig.4.5.1 Device

This is the device that is used for the visitor to press the doorbell i.e.., it is a smart bell that sends an SMS and an alert message to the Owner.

When the visitor presses the doorbell an alert message will be sent to the owner.

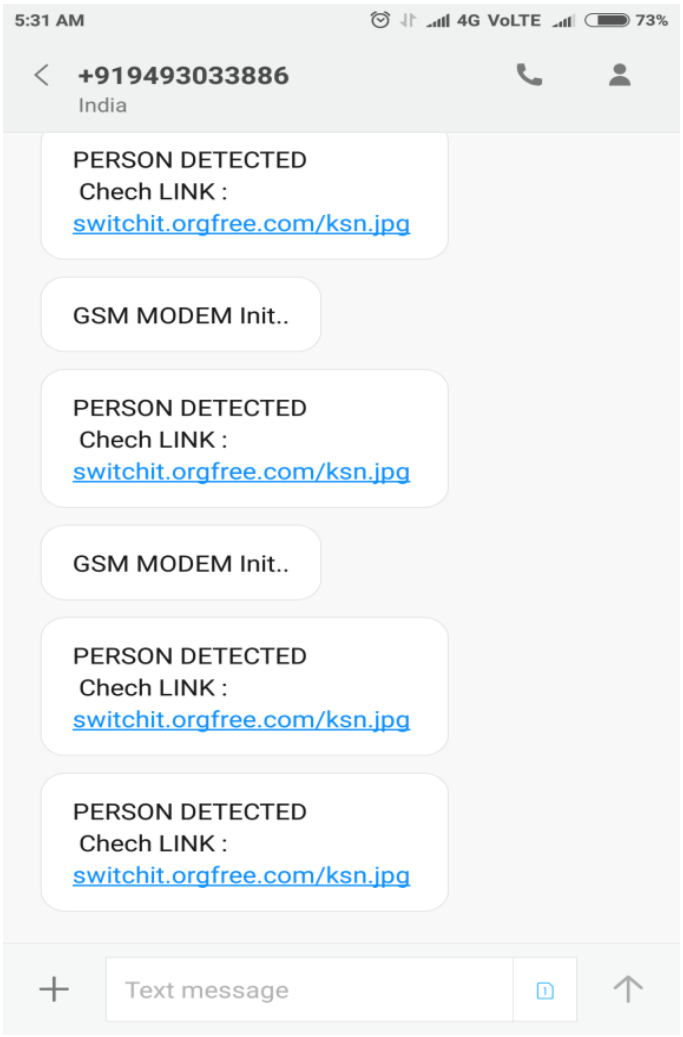


Fig 4.5.2 Alert Message

The above image/screenshot describes the SMS or alert message sent to the owner when the visitor presses the doorbell.

Parallelly an image of the visitor is captured and sent to the Mobile App.

Fig.4.5.3 Image sent to app

The above image/screenshot describes about the image sent to the app that is installed by the owner. The image is sent when the visitor presses the doorbell.

And all the images are also stored in Google drive.

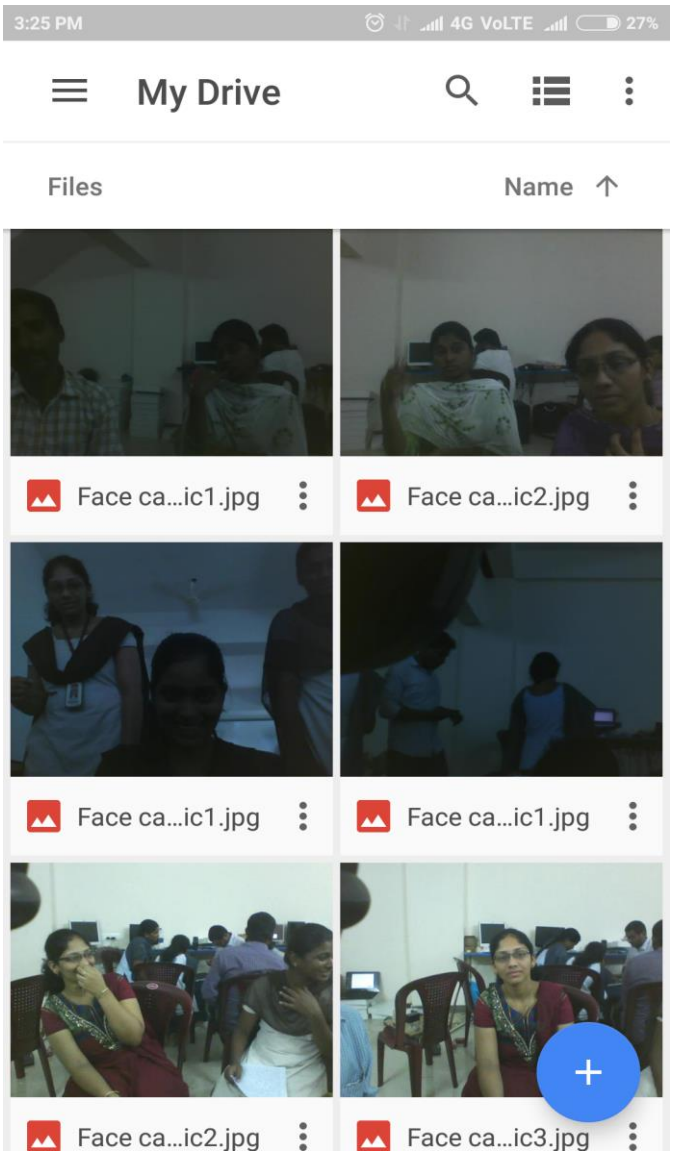


Fig 4.5.4 Images saved to drive

We are trying to save the images in the drive, so that the user can check images at any time if the owner will not receive SMS when the visitor rings the doorbell.

CHAPTER-5
TESTING

In general, testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. For example, in software development, product objectives are sometimes tested by product user representatives. When the design is complete, coding follows and the finished code is then tested at the unit or module level by each programmer; at the component level by the group of programmers involved; and at the system level when all components are combined together. At early or late stages, a product or service may also be tested for usability.

5.1 TESTING TECHNOLOGIES

Software Testing

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: blackbox testing and whitebox testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

System Testing

Testing is a process of finding faults with the package. To test the software there are several testing methodologies.

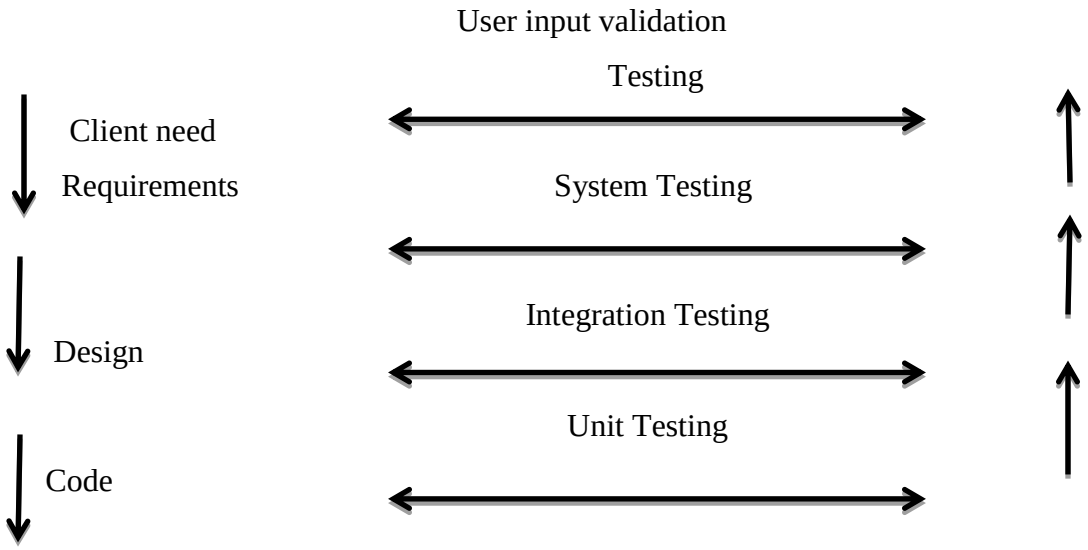
Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- Testing is a process of executing a program with the intent of finding an error.
- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.
- The tests are inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

Levels of Testing

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below…



Types of Testing

- Unit Testing
- System Testing
- User Input Validation Testing

Unit Testing

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins. In this project each service can be thought of a module. There are modules like User Registration and capturing of image, sending an alert message. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error.

System Testing

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal to see if software meets its requirements.

User input validation

The user input must be validated to check the image. The fields should also not be empty.

5.2 TEST CASES

Test Case 1

1	**Test case 1**	**Sending Image**
2	**Precondition**	Press Doorbell
3	**Description**	Visitor has to press the doorbell
4	**Test Steps**	Visitor has to press the doorbell when he visits the house
5	**Expected Output**	Image of the visitor should be sent to the Owner
6	**Actual Output**	If the doorbell is pressed then the image of the visitor is sent to the Owner.
7	**Status**	**Pass**

Table.5.2.1 Sending Image

Test case 2

1	**Test case 2**	**Sending SMS**
2	**Pre-condition**	Press the Doorbell
3	**Description**	Visitor has to press the doorbell
4	**Test Steps**	If the visitor presses the doorbell then an alert message will be sent to the Owner.
5	**Expected Output**	An alert message will be sent to the Owner when visitor presses the doorbell.
6	**Actual Output**	Alert message is received by the owner.
7	**Status**	**Pass**

Table.5.2.2 Sending SMS

CHAPTER-6
CONCLUSION & FUTURE SCOPE

We discuss about the elements of Smart phone and home system network. We actualized the discontinuous notice framework using the current doorbell work. There are a few preferences. For instance, it permits the visitor to work framework advantageously. Right now, the framework is restricted in that just a single Web cam gadget can be usable. In the future, voice recognition and face recognition functionalities can be expanded to fortify the security.

It can be further enhanced in such a way that we can upload even a video to our app to check what is happening in front of the door. The alert message can also be converted to voice message. We can even place a sensor and a button can be provided in the app such that when a button is pressed we can open the door.

BIBLIOGRAPHY

[1]. Park Kwang-ro, "Standard Technology Trend Home Network", Korea Information Technology Association, vol. 8, 2005.

[2] http://ieeexplore.ieee.org/document/7890101/?anchor=references

[3] Home network technology, IT/information technology magazine, http://i-bada.blogspot.kr.

[4] M. Day, Lotus, j. Rosenberg, H. Sugano, Fujitsu, “A Model for Presence and instant messaging”, article number: 2778

[5] Se-Hwan Park and Jong-Kyu Park “Security technology trend of IoTs”, proceedings of the ITFE Summer Conference. 2016.9.1

[6] Kim Hye Young and Park Hyun Joo, “Internet of Things security and implementation examples”, Korea Information Processing Society, Volume 22, Issue 2 (2015), pp.33-38.

[7]. M. Day, J. Rosenberg Lotus, H. Sugano Fujitsu, "A Model for Presence and instant messaging", pp. 2778.

[8] Kim Hye Young, Park Hyun Joo, "Internet of Things security and implementation examples", Korea Information Processing Society, vol. 22, no. 2, pp. 33-38, 2015.

[9] Vamsikrishna Patchava, Hari Babu Kandala, P. Ravi Babu, "A Smart Bell Notification system using IoT", IEEE International Conference on Samrt Sensors and Systems(IC-SSS), 2015.

[10]Chaouchi, Hakima. The Internet of Things. London: Wiley-ISTE, 2010.

[11]Chabanne, Herve, Pascal Urien, and Jean-Ferdinand Susini. RFID and the Internet of Things. London: ISTE, 2011.

[12]Brown, Eric (20 September 2016). "21 Open Source Projects for IoT". Linux.com. Retrieved 23 October 2016.

[13] "Internet of Things Global Standards Initiative". ITU. Retrieved 26 June 2015.

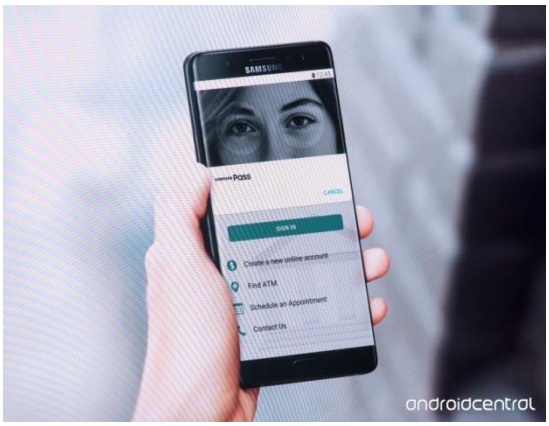
[14] http://www.faz.net/aktuell/wirtschaft/diginomics/grosse-internationale-allianz-gegen-cyber-attacken-15451953-p2.html?printPagedArticle=true#pageIndex_1

USER MANUAL

IOT SMART BELL NOTIFICATION SYSTEM USING ANDROID APPLICATION can be used in the following manner

1. Only the App is installed in owner smart phone where as he receives the image of visitor when the doorbell is pressed.

2. Owner receives an alert message to his smart phone that someone knocking the door.



2. The captured image is uploaded to Google Drive.

PUBLICATION DETAILS

Paper Acceptance Mail From IREJournals - International Peer Reviewed Journal Inbox x

IRE Journals <irejournals@gmail.com> Mar 21 (

to me

Images are not displayed. Display images below - Always display images from irejournals@gmail.com

Dear Author,

We are thankful for your contribution to Research Community of IREJournals.

We are pleased to inform you that your paper

Registration ID: 1700375

Title of the Paper: " IOT Smart Bell Notification System:Design And Implementation "

Author's Name: D.Pravallika

has been **Accepted for publishing** in **IREJournals,** which will be published in **IREJournals Volume 1, Issue 9 March 2018. Congratulations!**

At least one of the authors should complete the following step (before **30th March, 2018**):