INTELLIGENT DOOR ACCESS MANAGEMENT SYSTEM

**ABSTRACT**

Now-a-days the Safe Home Control plays a key role in the security. This can be achieved more benefits through intelligent door access management which is used to identify the person through capturing images using Web Camera and allow access to those authorized one. The term access control refers to the practice of restricting entrance to a property or building or a room to authorized persons. Although there are various door access systems such by fingerprints, password, pin etc., which may have drawback that pin and password may be known by the other persons and finger prints may be stolen. We will use the face recognization in live to provide security.

**Key Words**:

Safe Home Security, Authorization, Access Management.

**List Of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Figure** **No.** | **Figure** **Name** | **Page** **No.** |
| 1 | Figure 5.1 | RASPBERRY PI | 12 |
| 2 | Figure 5.2 | FLAME SENSOR | 13 |
| 3 | Figure 5.3 | SMOKE SENSOR | 14 |
| 4 | Figure 5.4 | SERVO MOTOR | 15 |
| 5 | Figure 5.5 | DHT 11 | 17 |
| 6 | Figure 5.6 | DATA SHIELD | 17 |
| 7 | Figure 5.7 | WEB CAMERA | 18 |
| 8 | Figure 7.1 | RASPBERRYPI TO SERVO MOTOR | 26 |
| 9 | Figure 7.2 | RASPBERRYPI TO SMOKE SENSOR | 26 |
| 10 | Figure 7.3 | RASPBERRYPI TO FLAME SENSOR | 27 |
| 11 | Figure 7.4 | RASPBERRYPI TO DHT 11 | 27 |
| 12 | Figure 7.5 | RASPBERRYPI TO WEB CAMERA | 27 |
| 13 | Figure 7.6 | ENTIRE CIRCUIT CONNECTIONS | 28 |

**CONNECTIONS**

|  |  |
| --- | --- |
| **Raspberry Pi** | **Flame sensor** |
| GND | GND |
| +5V | VCC |
| GPIO18 | D0 |

|  |  |
| --- | --- |
| **Raspberry Pi** | **Servo Motor** |
| GND | GND(BROWN) |
| 3.3V | VCC(RED) |
| GPIO7 | DIGITAL(ORANGE) |

|  |  |
| --- | --- |
| **Raspberry Pi** | **DHT11** |
| GND | GND |
| 3.3V | VCC |
| GPIO14 | DATA |

|  |  |
| --- | --- |
| **Raspberry Pi** | **Data Shield** |
| 3.3V | VCC |
| GND | GND |
| GPIO5 | L2 |
| GPIO6 | L4 |
| GPIO10 | L6 |

|  |  |
| --- | --- |
| **Raspberry Pi** | **Smoke Sensor** |
| GND | GND |
| +5V | VCC |
| GPIO23 | DOUT |

**Chapter 1**

**INTRODUCTION**

* 1. **SAFE HOME SECURITY**

**Home security** is both the security hardware in place on a property as well as personal security practices. Security hardware includes doors, locks, alarm systems, lighting, motion detectors, security camera systems, etc. that are installed on a property; personal security involves practices such as ensuring doors are locked, alarms activated, windows closed, extra keys not hidden outside, etc. One main component to home security is finding an area for easy programming and interaction for users. A control panel is implemented to arm and disarm a home security system. The control panel is the main connection to the alarm company monitoring a home. It typically features a touchpad or buttons to easily maneuver the system, and some newer systems also feature voice control or wireless remotes (key fobs).

Door and window sensors are also implemented in most home security systems. One part of the system is installed on the door or window itself while the other part is installed on the frame of the door or window. The two part system fits secure within each other when a door or window is closed, creating a security circuit. The traditional surveillance cameras can be installed for enhanced home security, which allows a user to remotely watch a video stream. A more recent version of the camera is capable of change detection and notify a user of such change events. Some forecasts project the home security market as a whole will be worth $47 billion by 2020, with the DIY home security market worth $1.5 billion.

**1.2 PROBLEM** **STATEMENT**

Providing a Safe Home Security System using finger prints and password has a drawback that they may be known to others or it may be stolen. So we are trying to provide a better security system using face recognization.

**Chapter 2**

**REQUIREMENTS SPECIFICATION**

**2.1 The Python Programming Language:**

**Python** is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

[Python Package Index](https://en.wikipedia.org/wiki/Python_Package_Index) (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration
* Scientific computing
* Text processing
* Image processing

**2.2 Raspberry Pi Platform**

The Raspberry Pi Foundation provides Raspbian, a Debian-based Linux distribution for download, as well as third party Ubuntu, Windows 10 IOT Core, RISC OS, and specialised media centre distributions. It promotes Python and Scratch as the main programming languages, with support for many other languages. The default firmware is closed source, while an unofficial open source is available.  A Raspberry Pi Foundation sanctioned device, designed for educational purposes, that expands the Raspberry Pi's GPIO pins to allow interface with and control of LEDs, switches, analog and digital signals, sensors and other devices. It also includes an optional Arduino compatible controller to interface with the Pi.

**2.3 Virtual Network Computing(VNC)**

In computing, **Virtual Network Computing** (**VNC**) is a graphical desktop-sharing system that uses the Remote Frame Buffer protocol (RFB)to remotely control another computer. It transmits the keyboard and mouse events from one computer to another, relaying the graphical-screen updates back in the other direction, over a network.

**VNC is platform-independent** – there are clients and servers for many GUI-based operating systems and for Java. Multiple clients may connect to a VNC server at the same time. Popular uses for this technology include remote technical support and accessing files on one's work computer from one's home computer, or vice versa.

**2.4 Raspberry Pi Specifications**

**SoC:** Broadcom BCM2837  
**CPU:** 4× ARM Cortex-A53, 1.2GHz  
**GPU:** Broadcom VideoCore IV  
**RAM:** 1GB LPDDR2 (900 MHz)  
**Networking:** 10/100 Ethernet, 2.4GHz 802.11n wireless  
**Bluetooth:** Bluetooth 4.1 Classic, Bluetooth Low Energy  
**Storage:** microSD  
**GPIO:** 40-pin header, populated  
**Ports:** HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

### 2.4.a Wireless radio

So small, its markings can only be properly seen through a microscope or magnifying glass, the Broadcom BCM43438 chip provides 2.4GHz 802.11n wireless LAN, Bluetooth Low Energy, and Bluetooth 4.1 Classic radio support. Cleverly built directly onto the board to keep costs down, rather than the more common fully qualified module approach, its only unused feature is a disconnected FM radio receiver.

## 2.4.b Antenna

There’s no need to connect an external antenna to the Raspberry Pi 3. Its radios are connected to this chip antenna soldered directly to the board, in order to keep the size of the device to a minimum. Despite its diminutive stature, this antenna should be more than capable of picking up wireless LAN and Bluetooth signals – even through walls.

## 2.4.c SoC

Built specifically for the new Pi 3, the Broadcom BCM2837 system-on-chip (SoC) includes four high-performance ARM Cortex-A53 processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory, a VideoCore IV graphics processor, and is linked to a 1GB LPDDR2 memory module on the rear of the board.

## 2.4.d GPIO

The Raspberry Pi 3 features the same 40-pin general-purpose input-output (GPIO) header as all the Pis going back to the Model B+ and Model A+. Any existing GPIO hardware will work without modification; the only change is a switch to which UART is exposed on the GPIO’s pins, but that’s handled internally by the operating system.

## 2.4.e USB chip

The Raspberry Pi 3 shares the same SMSC LAN9514 chip as its predecessor, the Raspberry Pi 2, adding 10/100 Ethernet connectivity and four USB channels to the board. As before, the SMSC chip connects to the SoC via a single USB channel, acting as a USB-to-Ethernet adaptor and USB hub.

**Chapter 3**

**REQUIREMENTS ANALYSIS**

Requirements analysis is also known as requirements engineering. It is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term requirements analysis can also be applied specifically to the analysis proper, as opposed to elicitation or documentation of the requirements, for instance. In our system the main requirements are .

**3.1 HARDWARE REQUIREMENTS:**

RASPBERRY PI

FALME SENSOR

SMOKE SENSOR

DHT

WEB CAM

SERVO MOTOR

DATA SHIELD

**3.2 SOFTWARE REQUIREMENTS:**

OPERATING SYSTEM : VNC Viewer

CODING LANGUAGE : Python

**Chapter 4**

**PROJECT WORKING PROCESS**

The Intelligent Door Access Management System will provides a another type of Safe Home Security being different from the other systems like fingerprint sensors and password type access systems. The different feature of the system is it used the Raspberry Web Camera to recognize the persons face and provide access to them when the identified face matches otherwise not. When a person tries to enter into the home the camera will take a picture and send it to the cloud and provides access to him/her whether it is a authorized one or not. If the person given the access then the door must be opened automatically, so we can use Servo Motor.

**4.1 Door Open/Close**

When the access is provided the motor will start rotating 180 degrees clockwise in order to open the door. The after the person entering into the home the door must be closed so that we can set some delay time, after the delay is completed the motor again rotates same 180 degrees in anti-clockwise.

In some cases we will leave our home without switching off lights and fans. So provide a solution to that cases here we have included here:

**4.2 Light On - Light Off/ Fan on - Fan Off**

For this we can provide connections to the Basic Data shield to Raspberry Pi and can control all the lights and fans with our device. We can switch on or switch off any light or fan in our home.

**4.3 Fire Sensor**

In some cases like there is a chance of occurring of fire accidents. To overcome this kind of actions we had included a fire sensor to identify and generate a signal when if any fire occurs. The sensor will have LED bulb, the fire occurs then it will glow and we get a signal as 1 which indicates that there is fire so we can immediately overcome the problem.

**4.4 Smoke Sensor**

However we can connect a smoke sensor to our Raspberry Pi. This will be helpful to us to identify the gas leakages in home. The sensor has a feature to identify the smoke when it occur and generate a status 0 when there is a fire. So we encounter these kind of problems we will find immediate olution to these problems.

**4.5 DHT**

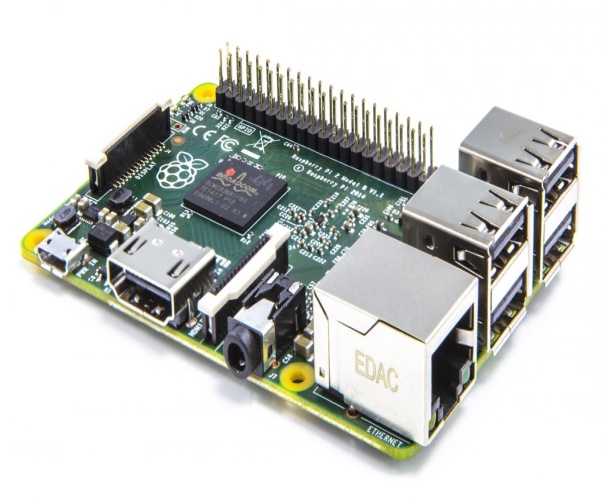
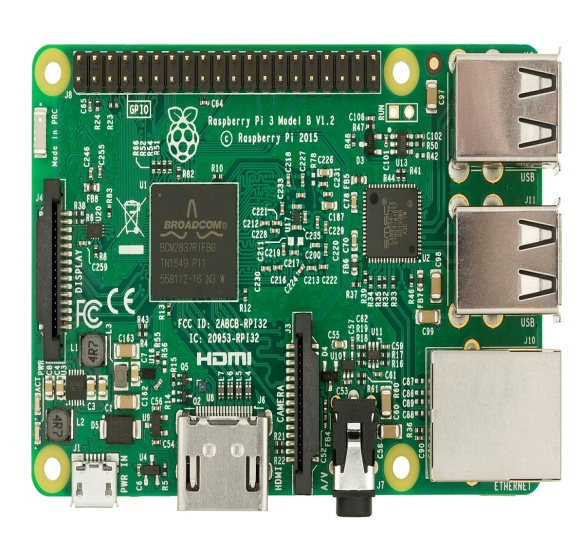
In order to identify the temperature and humidity conditions in our home the DHT sensor will provide this feature. The sensor will identify the room temperature and humidity conditions and will display on the screen. So according to that conditions we will manage our safety arrangements.

After making all the connections to the Raspberry Pi we can create a mobile application for the entire for project using MIT App Inventor and we can control each and everything through our mobile phone from anywhere. In this way the system will provide a better safe home security system providing better safety through face recognization which is unique for everyone and can’t be possible to use one person access another one to more extent.

**Chapter 5**

**COMPONENTS USED**

**5.1 RASPBERRY PI**

**5.1.a FEATURES**

The Raspberry Pi 3 will cost the same as its predecessor, but feature much more powerful hardware. [Bluetooth](https://www.wired.co.uk/topic/bluetooth) will be built into the board for the first time, and is powered by a Quad Core Broadcom BCM2837 64bit ARMv8 processor. The Pi 3 runs at 1.2 GHz, compared to the Pi 2's 900MHz, and also has an upgraded power system, and the same four USB ports and extendable 'naked board' design as the Pi 2.

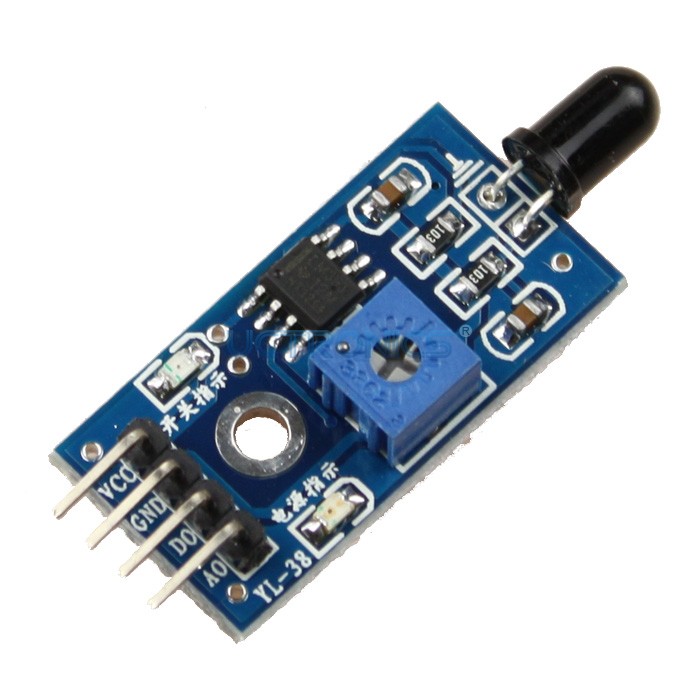
**5.1.b SPECIFICATIONS**

* **CPU**: Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
* **GPU**: 400MHz VideoCore IV multimedia
* **Memory**: 1GB LPDDR2-900 SDRAM (i.e. 900MHz)
* **USB** **ports**: 4
* **Video** **outputs**: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
* **Network**: 10/100Mbps Ethernet and 802.11n Wireless LAN
* **Peripherals**: 17 GPIO plus specific functions, and HAT ID bus
* **Bluetooth**: 4.1
* **Power** **source**: 5 V via MicroUSB or GPIO header
* **Size**: 85.60mm × 56.5mm
* **Weight**: 45g (1.6 oz)

**5.1.c WORKING PRICIPLE**

The working principal is like any modern PC. They include a processor, memory, storage (SD card) video (HDMI), network (10/100 Ethernet) AND IO (four USB 2.0 ports). The power supply is a micro USB connector, and it’s happiest with 5–5.2 volts at 2.5A. The heavier current is in case you plug high draw devices into the USB port, otherwise it can get by on 1.5A supply.

**5.2 FLAME SENSOR**



**5.2.a FEATURES**

High Photo Sensitivity

Fast Response Time

Sensitivity adjustable

**5.2.b SPECIFICATIONS**

Working voltage : 3.3V - 5V

Detect Range : 60 degrees

Digital/Analog Output

On-board LM393 chip

**Dimension:** 3.2cm x 1.4cm

**5.2.c WORKING PRICIPLE**

A **flame detector** is a sensor designed to detect and respond to the presence of a flame or fire, allowing **flame detection**. Responses to a detected flame depend on the installation, but can include sounding an alarm, deactivating a fuel line and activating a fire suppression system. When used in applications such as industrial furnaces, their role is to provide confirmation that the furnace is working properly.

**5.3 SMOKE SENSOR**



**5.3.a FEATURES**

High sensitivity to carbon monoxide

Stable and long life

**5.3.b APPLICATION**

Domestic gas leakage detector

Industrial CO detector

Portable gas detector

**5.3.c WORKING PRICIPLE**

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. **Gas sensors** are very important part of such systems.  Small like a nose, gas sensors spontaneously react to the gas present, thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state.  **Gas sensors** are available in wide specifications depending on the sensitivity levels, type of gas to be sensed, physical dimensions and numerous other factors

**5.4 SERVO MOTOR**



**5.4.a FEATURES**

The servo motor is specialized for high-response, high-precision positioning. As a motor capable of accurate rotation angle and speed control, it can be used for a variety of equipment.

**5.4.b SPECIFICATIONS**

**Size** : 38 x 11.5 x 24mm

**Weight** : 17g

**Speed** : 0.14sec/60degrees

**Torque** : 2.5kgf-cm

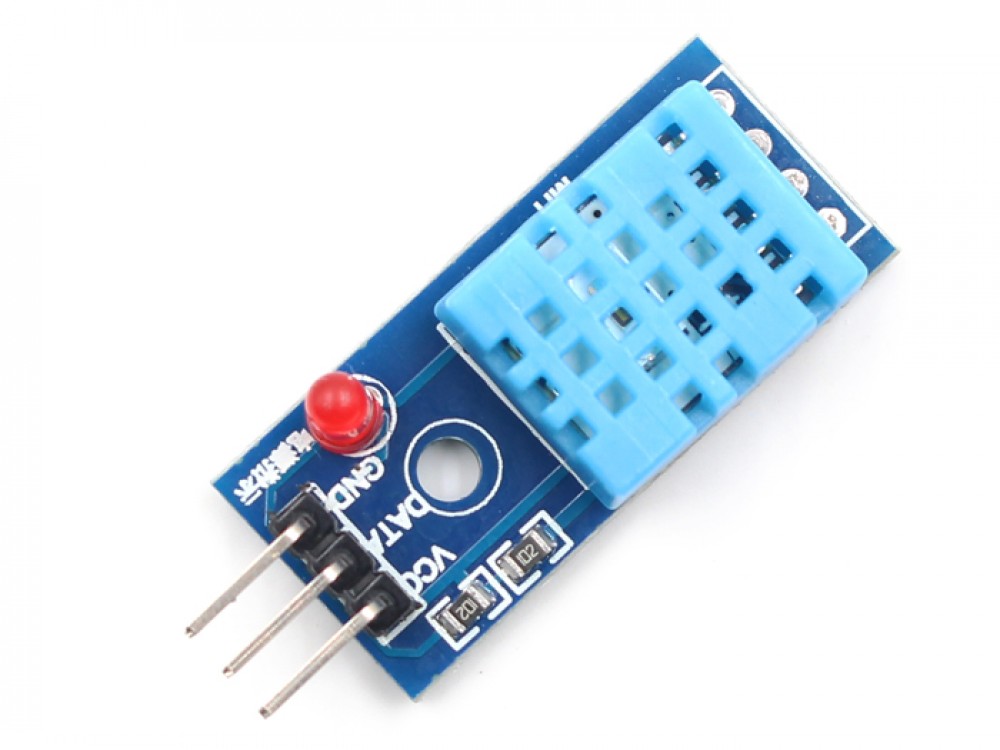
**Voltage** : .8V-6.0V

**Connector** **type** : JR type (Yellow: Signal, Red: VCC, Brown:GND)

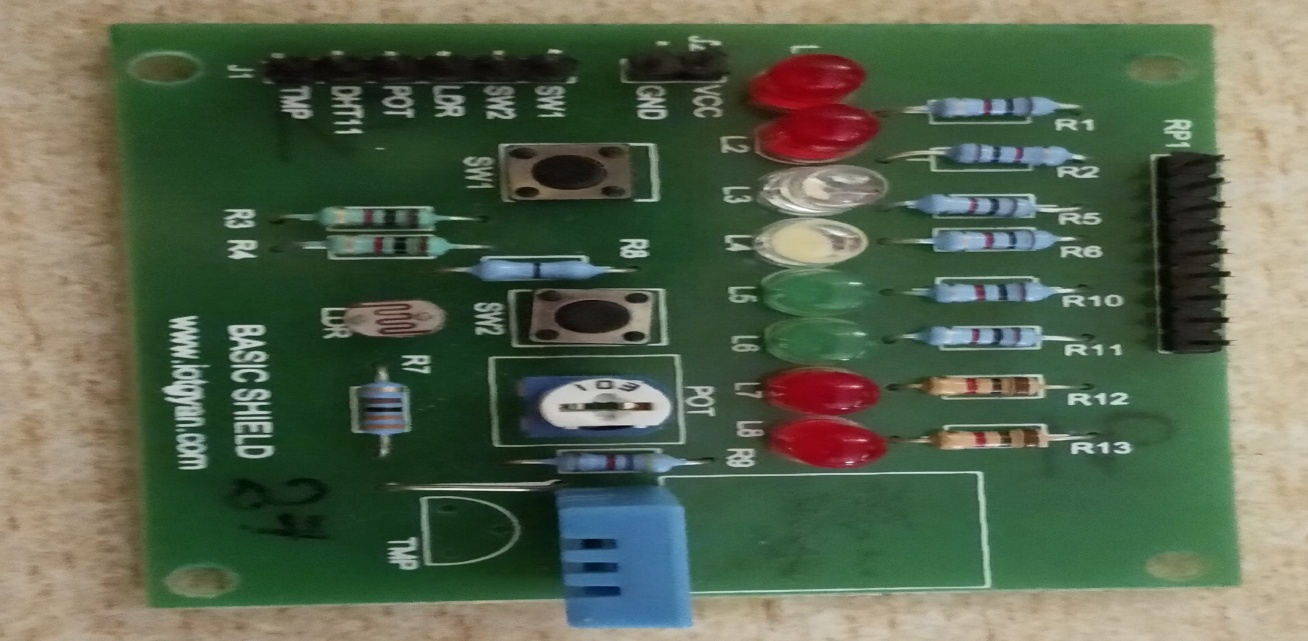
**5.4.c WORKING PRICIPLE**

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly and a controlling circuit. First of all we use gear assembly to reduce RPM and to increase torque of motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now difference between these two signals, one comes from potentiometer and another comes from other source, will be processed in feedback mechanism and output will be provided in term of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with potentiometer and as motor rotates so the potentiometer and it will generate a signal. So as the potentiometer’s angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

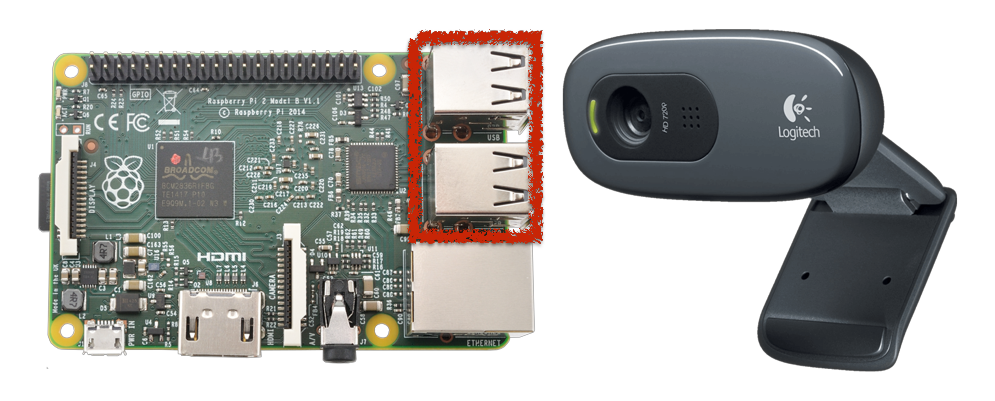
**5.5 DHT11**



**5.6 DATA SHIELD**

****

**5.7 WEB CAMERA**



import cv2

# initialize the camera

cam = cv2.VideoCapture(0)

ret, image = cam.read()

if ret:

cv2.imshow('SnapshotTest',image)

cv2.waitKey(0)

cv2.destroyWindow('SnapshotTest')

cv2.imwrite('/home/pi/book/output/SnapshotTest.jpg',image)

cam.release()

**Chapter 6**

**FINAL CODE IMPLEMENTATION**

# -\*- coding: utf-8 -\*-

import RPi.GPIO as GPIO

import time

import numpy as np

import cv2

from datetime import datetime

import httplib, urllib, os, glob, requests , urllib2

import dht11

fire = 0

gas = 0

temperature = 0

humidity = 0

Doorbell =4

pir = 21

firesensor = 18

gassensor = 23

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(Doorbell , GPIO.IN)

GPIO.setup(pir , GPIO.IN)

GPIO.setup(gassensor, GPIO.IN)

GPIO.setup(firesensor , GPIO.IN)

GPIO.setup(7, GPIO.OUT)

p = GPIO.PWM(7, 50)

p.start(2.5)

instance = dht11.DHT11(pin = 14)

global previous\_state

previous\_state = False

global current\_state

current\_state = False

GPIO.setup(5, GPIO.OUT)

GPIO.output(5, GPIO.LOW)

GPIO.setup(6, GPIO.OUT)

GPIO.output(6, GPIO.LOW)

GPIO.setup(10, GPIO.OUT)

GPIO.output(10, GPIO.LOW)

base\_url = "http://smartsecurity.thesmartbridge.com/API/update?key=71486723251"

def doorcontrol():

url = "http://smartsecurity.thesmartbridge.com/API/get\_talkback?userkey=71486723251"

connect = urllib2.urlopen(url)

response= connect.read()

#response = urllib.urlopen(url).read()

print response

try:

if response == "DOOROPEN":

p.ChangeDutyCycle(12.5) #180°

if response == "DOORCLOSE":

p.ChangeDutyCycle(2.5) #0°

if response == "LIGHT1ON":

GPIO.output(5,GPIO.HIGH)

print "light1 is on"

if response == "LIGHT1OFF":

GPIO.output(5,GPIO.LOW)

print "light1 is off"

if response == "LIGHT2ON":

GPIO.output(10,GPIO.HIGH)

print "light2 is on"

if response == "LIGHT2OFF":

GPIO.output(10,GPIO.LOW)

print "light2 is off"

if response == "FANON":

GPIO.output(6,GPIO.HIGH)

print "fan is on"

if response == "FANOFF":

GPIO.output(6,GPIO.LOW)

print "fanoff is off"

except keyboardinterrupt:

p.stop()

def sensors():

gas = GPIO.input(gassensor)

print " GAS STATUS " +str(gas)

fire = GPIO.input(firesensor)

print " fire STATUS " +str(fire)

result = instance.read()

if result.is\_valid():

temperature = result.temperature

humidity = result.humidity

print temperature

print humidity

else:

print "no data from sensor"

temperature = 33

humidity = 44

url2 = base\_url + "&field1="+str(temperature)+"&field2="+str(humidity)+"&field3="+str(0)+"&field4="+str(gas)+"&field5="+str(current\_state)

print(url2)

f = urllib2.urlopen(url2)

print f.read()

f.close()

def camera():

count =1

current\_state = GPIO.input(Doorbell)

current\_state1 = GPIO.input(pir)

print " button state = " + str(current\_state)

print " pir state = " + str(current\_state1)

if current\_state == 0 or current\_state1 ==1:

cap = cv2.VideoCapture(0)

ret, frame = cap.read()

print "Saving Photo"

picname = datetime.now().strftime("%y-%m-%d-%H-%M")

picname = picname+str(count)+'.jpg'

cv2.imwrite(picname, frame)

url1 = 'http://smartsecurity.thesmartbridge.com/API/image\_upload/71486723251'

files = {'fileToUpload': open(picname, 'rb')}

r = requests.post(url1, files=files)

print r.text

time.sleep(5)

cap.release()

count=count+1

while True:

previous\_state = current\_state

sensors()

camera()

time.sleep(2)

doorcontrol()

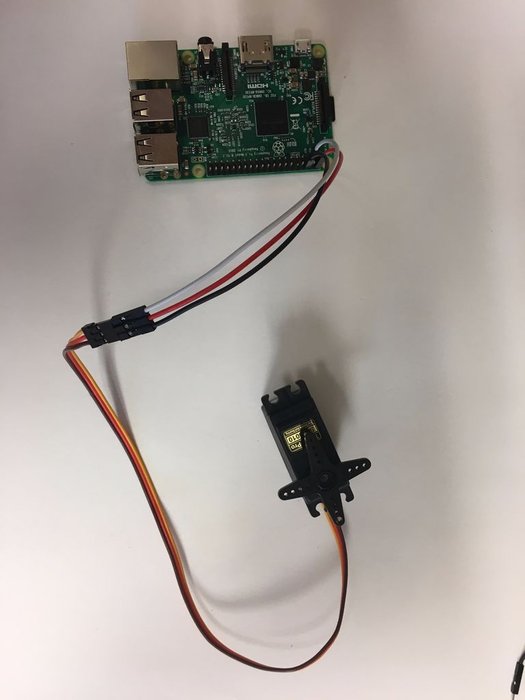
#upload()

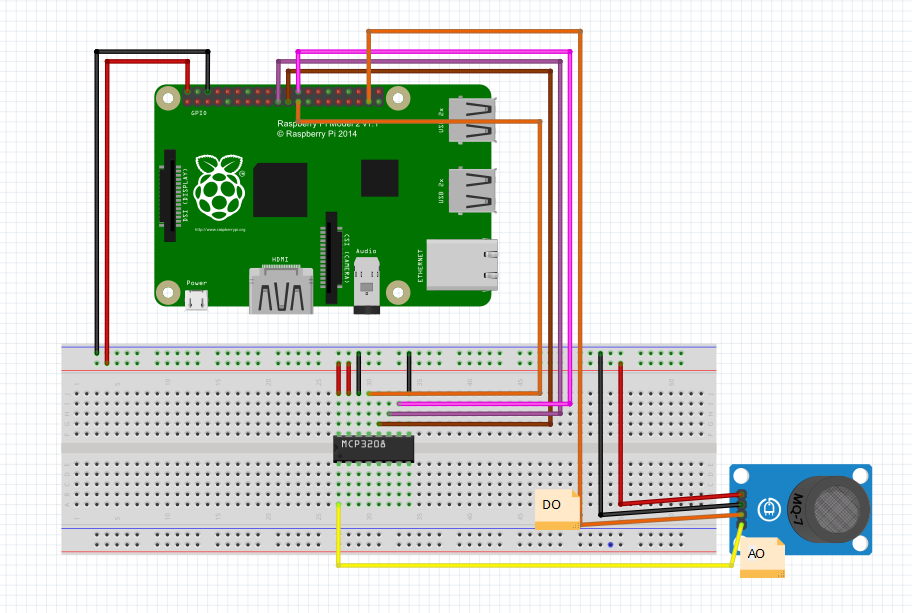
**Chapter 7**

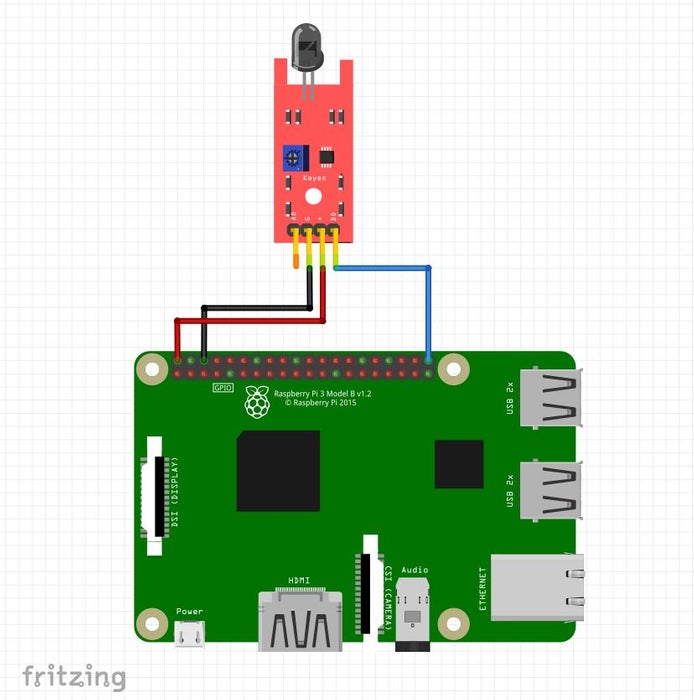
**CONNECTIONS FROM**

**RASPBERRY PI TO SENSORS**

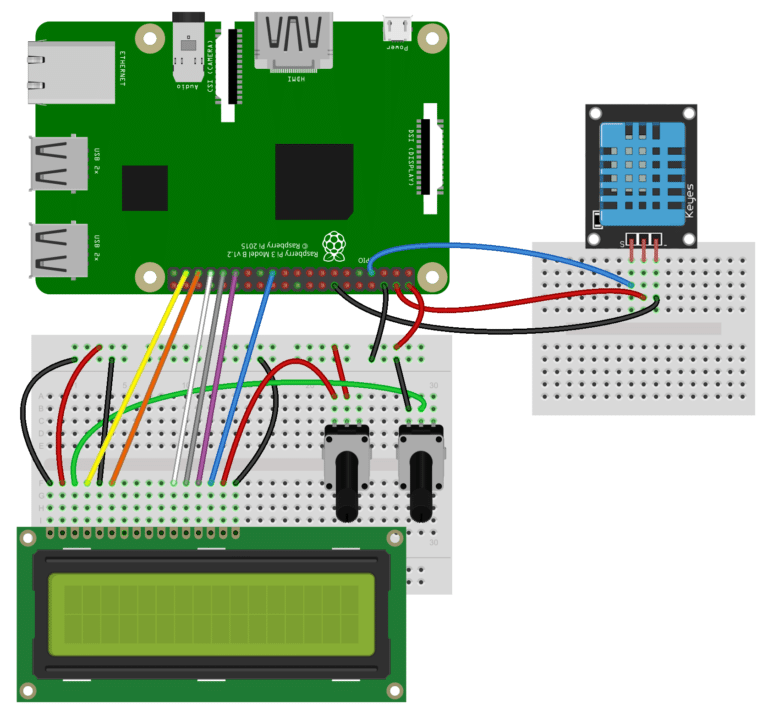
**7.1 RASPBERRY TO SERVO MOTOR**



**7.2 RASPBERRY TO SMOKE SENSOR** 

**7.3 RASPBERRY TO FLAME SENSOR**

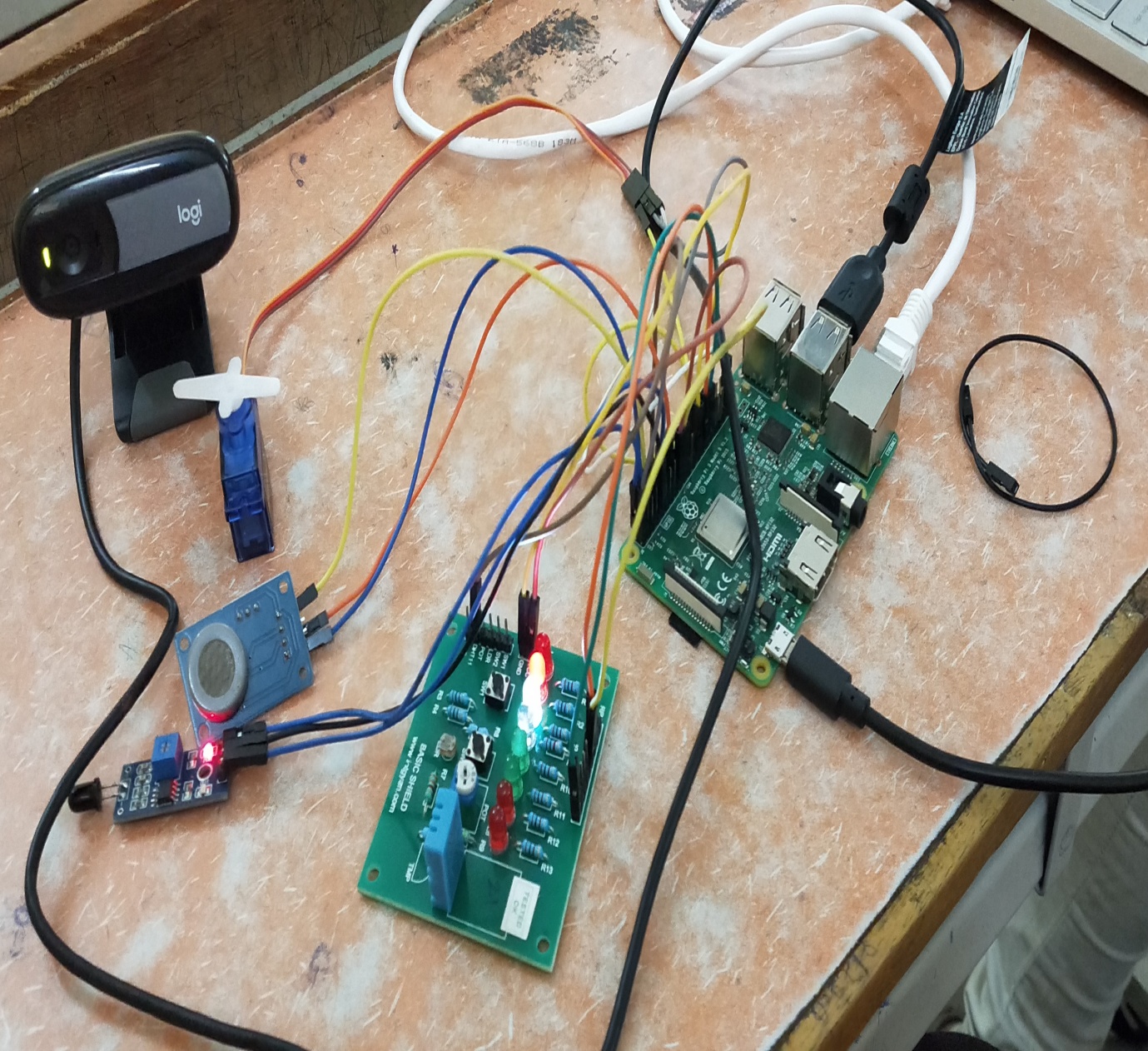
**7.4 RASPBERRY TO DHT SENSOR**



**7.5 RASPBERRY TO WEBCAMERA**



**7.6 CIRCUIT CONNECTIONS**

****

**Chapter 8**

**INSTALLATIONS**

**OPEN CV INSTALLATION**

First of all, we update the package lists:

sudo apt-get update && sudo apt-get upgrade && sudo rpi-update

A reboot is necessary if it has been updated.

sudo reboot

Then you can install all the important tools and libraries needed for OpenCV (installation takes a few minutes).

sudo apt-get install build-essential git cmake pkg-config libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libgtk2.0-dev libatlas-base-dev gfortran

If everything worked, we could clone OpenCV from git. This step also takes a few minutes.

git clone https://github.com/Itseez/opencv.git && cd opencv &&git checkout 3.0.0

Whether version 3.0 or 2.4 of OpenCV is taken is up to you. Depending on the application, one of the versions may be better suited.

Afterwards, OpenCV can be compiled. You can either use Python 2.7 or Python 3+. There are some differences between the versions, especially as some libraries are not (yet) executable with Python 3+. However, this mainly affects smaller libraries, as common libraries (NumPy, SciPy, etc.) usually provide the respective files for both versions.

In this tutorial, I use Python 2.7. If you already have Python installed and want to know which version is installed, you can simply enter python into the console and get the exact version at the beginning (the command for Python 3+ is python3). If you do not have a Python installed, you can install it by following the steps below:

sudo apt-get install python2.7-dev

We also need the package management tool pip, which installs NumPy right away:

cd ~ && wget https://bootstrap.pypa.io/get-pip.py && sudo python get-pip.py

Now we can simply install via pip NumPy. NumPy is a library that makes it very easy to perform array operations in Python.

pip install numpy

But now to compile OpenCV. For this purpose, a build folder must be created in which the compiled files land:

cd ~/opencv && mkdir build && cd build

cmake -D CMAKE\_BUILD\_TYPE=RELEASE

-D CMAKE\_INSTALL\_PREFIX=/usr/local

-D INSTALL\_PYTHON\_EXAMPLES=ON

-D INSTALL\_C\_EXAMPLES=ON

-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib/modules

-D BUILD\_EXAMPLES=ON ..

Now you can finally compile. This step takes (depending on Raspberry Pi model) quite a long time (on my Pi 2 about an hour). To use all four cores to compile on the Raspberry Pi 2, type in the following:

make -j4

If the compilation has worked without problems, we can install OpenCV:

sudo make install && sudo ldconfig

To capture the images using web camera first we need to install the web cam feature and write the following commands:

Sudo apt-fswebcam

To get OpenCV installation for coding we need to do:

Sudo apt-get install python.

Done!