TECHNISCHE UNIVERSITÄT
CHEMNITZ

Department of Electrical Engineering and Information Technology

Chair of Measurement and Sensor Technology

Project Documentation

"Project Lab Embedded Systems"

Group: 10

Members: Manoja Hirannaiah Anitha

Kshama Ramesh

Project: Automation and Video Monitoring using Raspberry Pi

Date: 2018-08-14

1 Abstract

The main objective of this project is to develop an interface for video monitoring by using Raspberry Pi B architecture. This project deals with Raspberry Pi processor, temperature sensor DHT-22, Raspberry Pi camera or webcam. This report presents the detection of real-time temperature and current video of the monitored system. In the presented system, it is possible to read the data from any adaptable sensor and monitoring it using a web-enabled device (computer, phone) web browser jointly with real-time video of the desired device. It is having capability to monitoring variables and also makes a visible state of such equipment using video features. Python language is used with DHT22 Temperature and Humidity Sensor. This last one is a popular sensor module used by hobbyists for implementing in a lot of IoT Projects. This sensor along with Raspberry Pi can also be used in:

- HVAC Systems
- Thermostats
- Home and Office Climate Control
- Weather Station

2 Member Responsibilities

Kshama Ramesh	Establishing the virtual screen, presentation and report
Manoja Hirannaiah Anitha	Programming, error rectification, presentation and report
Mozammil Hasan Arfi	Installation of Raspbian, server software, presentation and report

3 Functional Description

3.1 Overview

Temperature has an impact on almost all the activities surrounding us. A precise determination of temperature is a vital factor in countless industries and different fields of science. The temperature monitoring is crucial in lot of industries, like food industry, the workshop and pharmaceutical industry. Analog and digital Temperature sensors are available for sensing temperature for commercial purpose. Temperature sensors possessing temperature-dependent properties that can be measured electrically contain resistors, semiconductor mechanisms such as diodes, thermocouples, thermistors. This project aims at monitoring the real time temperature and relative humidity in a cost effective way. Here the monitoring node is raspberry pi. Programming language used for raspberry pi is Python. The Sensor utilized here is DHT22 temperature sensor. This sensor consists of thermistor and basic advantage of using DHT22 sensor is that it is economical and light in weight. The sensor is interfaced with the raspberry pi using jumper wires. The temperature is sensed using the sensor DHT22 and is read, stored and displayed by the raspberry pi kit.

In the later part of the report we will be discussing about the hardware component used, software supports and the connection setup.

3.2 Hardware

The project is having following hardware components. The detailed description of these components is below.

- Raspberry Pi
- DHT22 sensor
- 10k resistor (one per sensor)
- 8 GB micro SD CARD
- USB Power Supply
- USB Cable
- Breadboard & Jumper Wires
- Raspberry Pi Camera

This project is built on a breadboard, so no need to worry about soldering, or a designing PCB.

3.3.1 Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer which plugs into a computer monitor or TV, and requires a standard keyboard and mouse. Raspberry Pi is a dynamic microcontroller and runs with the Python programming language. In this project we have used Raspberry Pi B model B.
It includes:

- CPU: Quad-core 64-bit ARM Cortex A53 clocked at 1.2 GHz
- GPU: Video Core IV multimedia
- SoC: Broadcom BCM2837
- 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 Micro USB or GPIO header
- Size: 85.60mm × 56.5mm
- Weight: 45g (1.6 oz)

It has capability of a little device that allows people of all ages to explore computing. It's capable of doing everything you would expect a desktop/computer to do, from browsing the internet and playing high-definition video to making spreadsheets, word processing, and playing games . The Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array projects like smart home monitoring system, wireless motion sensor activated light and many more.

There are number of models available for raspberry pi based on their different features and different hardware configurations. Following are some model mentioned as per their releasing details.

- Raspberry Pi 3 B+ - Release Date 28 Mar 2018
- Raspberry Pi Zero WH - Release Date 12 Jan 2018
- Raspberry Pi Zero W - Release Date 28 Feb 2017
- Raspberry Pi 3 - Release Date 29 Feb 2016
- Raspberry Pi Zero - Release Date 30 Nov 2015
- Raspberry Pi 2 - Release Date 01 Feb 2015
- Raspberry Pi A+ - Release Date 10 Nov 2014
- Raspberry Pi B - Release Date 15 Feb 2012

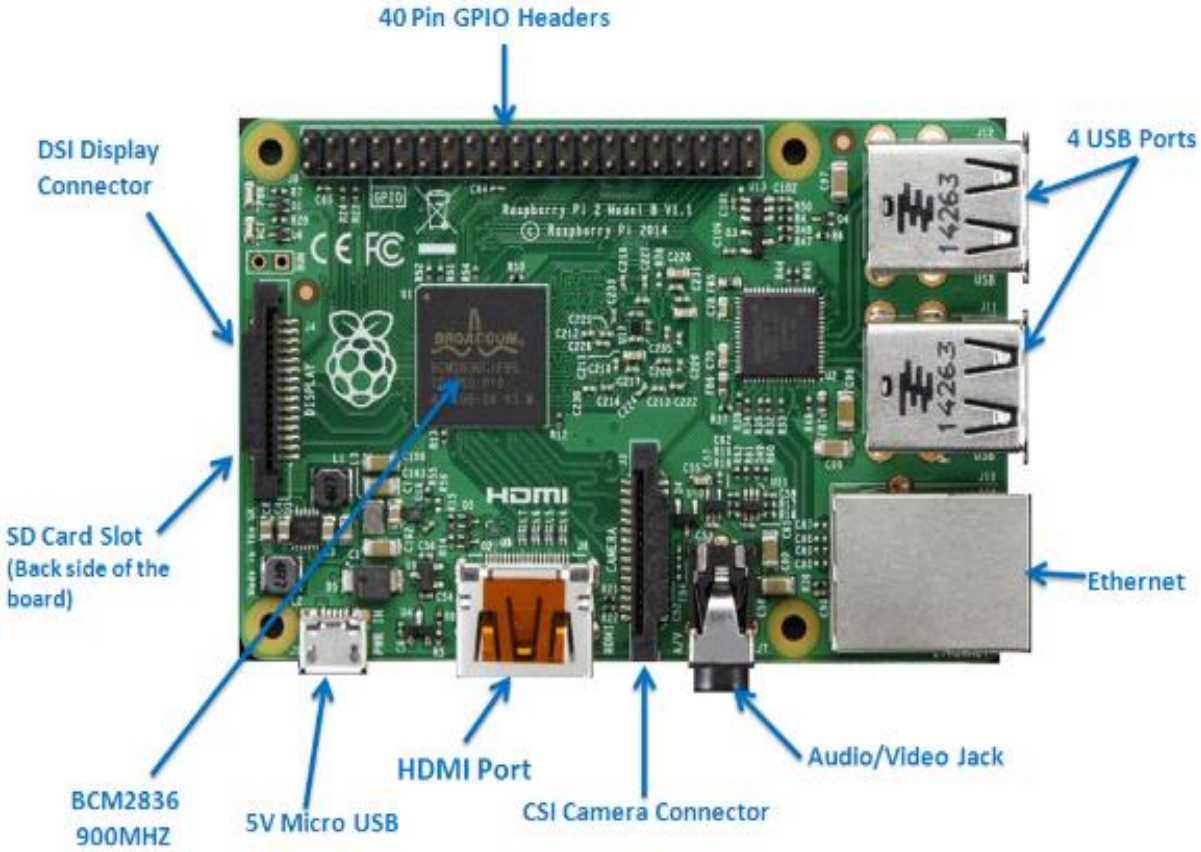


Fig1: Raspberry Pi 3 Model B

Raspberry Pi module is capable of performing various basic functionalities of multimedia, gaming and surveillance applications. The functionality of various components of Raspberry Pi is stated as below.

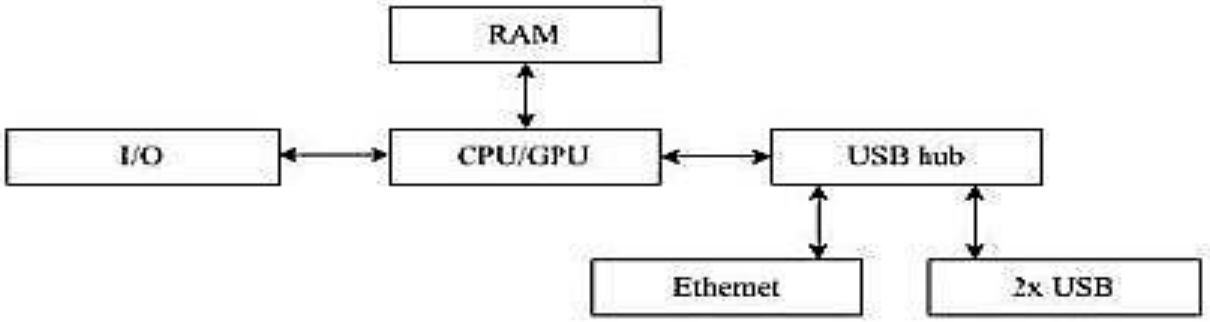


Fig 2 : Block Diagram of Raspberry Pi

a. Micro SD socket: SDIO (Secured Digital Input/Output) memory card is required to store the operating system and files. The operating system boots from Micro SD card, it is running a version of the Linux operating system (i.e. Raspbian OS). The maximum support of the SD card is about 32GB
b. Micro USB Power supply socket: Raspberry Pi Micro USB socket 5V, 2A. It also creates three different voltages 3.3V, 2.5V and 1.8V for the processor and Ethernet.
c. Ethernet socket: 10/100 BaseT Ethernet socket used for LAN network.
d. Audio output: 4-pole (TRRS) type connector, 3.5mm Jack, HDMI (High Definition Multimedia Interface), stereo audio and composite video.
e. Video output: HDMI (rev 1.3 & 1.4), composite RCA (PAL and NTSC). It is type of electrical connector commonly used to carry audio and video signals.
f. USB socket: Four Universal Serial Hub 2.0 connectors are available. The LAN9512, which is basically, turned the 1 USB port on the processor into 2 ports + Ethernet.
g. GPIO Connector: 40-pin 2.54mm (100mil) expansion header: 2x20 strip providing 27 GPIO pins as well as +3.3V, +5V and GND supply lines.
h. Broadcom BCM2835 SoC: 700MHz low power ARM 1176ZFS application processor is available on board.
i. Graphic Processing Unit: Dual core VideoCore IVR Multimedia Co-processor is used, which is having 1080p30 264 high profile decode.
j. Power requirements for board are 5V, 2A

3.3.2 DHT22 Sensor

The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

Simply connect the first pin on the left to 3-5V power, the second pin to data input pin (GPIO) and the rightmost pin to ground. Although it uses a single-wire to send data. For a multiple sensors, each one must have its own data pin. The arrangement is shown in below figure.

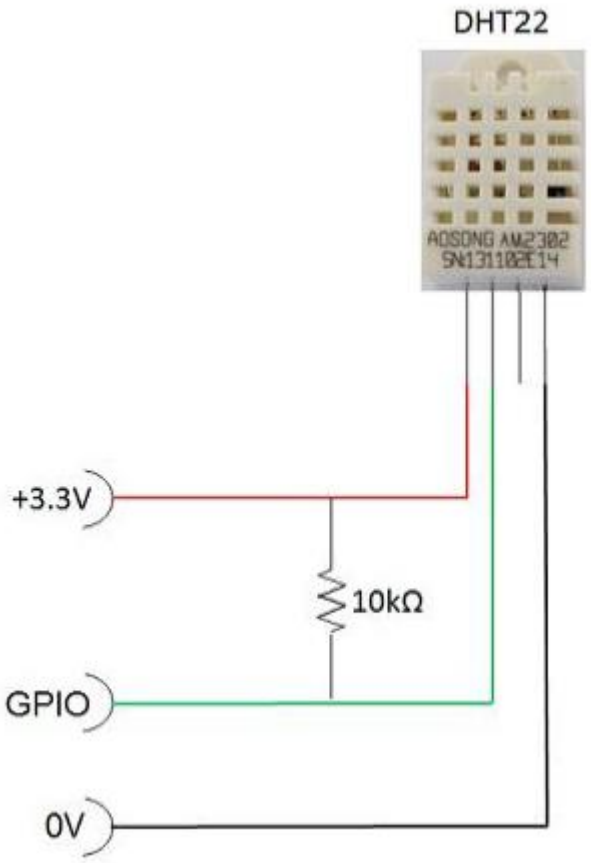


Fig 3 : DHT22 Sensor connections

The general Tachnical Details of this sensors type are:

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
- 4 pins, 0.1" spacing
- Weight (just the DHT22): 2.4g

3.3.3 Raspberry Pi Camera

Raspberry Pi camera is used to capture the video and images of monitorring area. It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. The camera works with all models of Raspberry Pi.

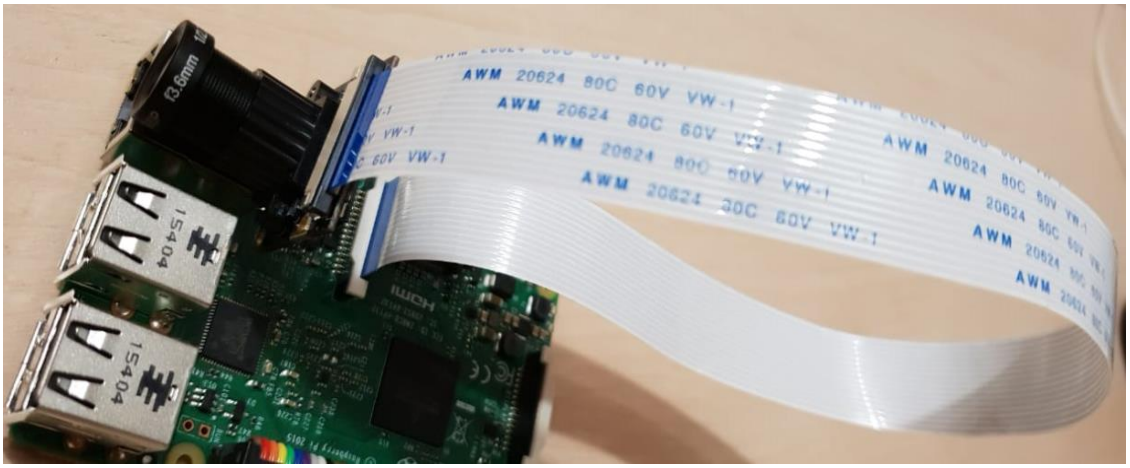


Fig 4 : Raspberry Pi Camera

3.3.4 System Architecture

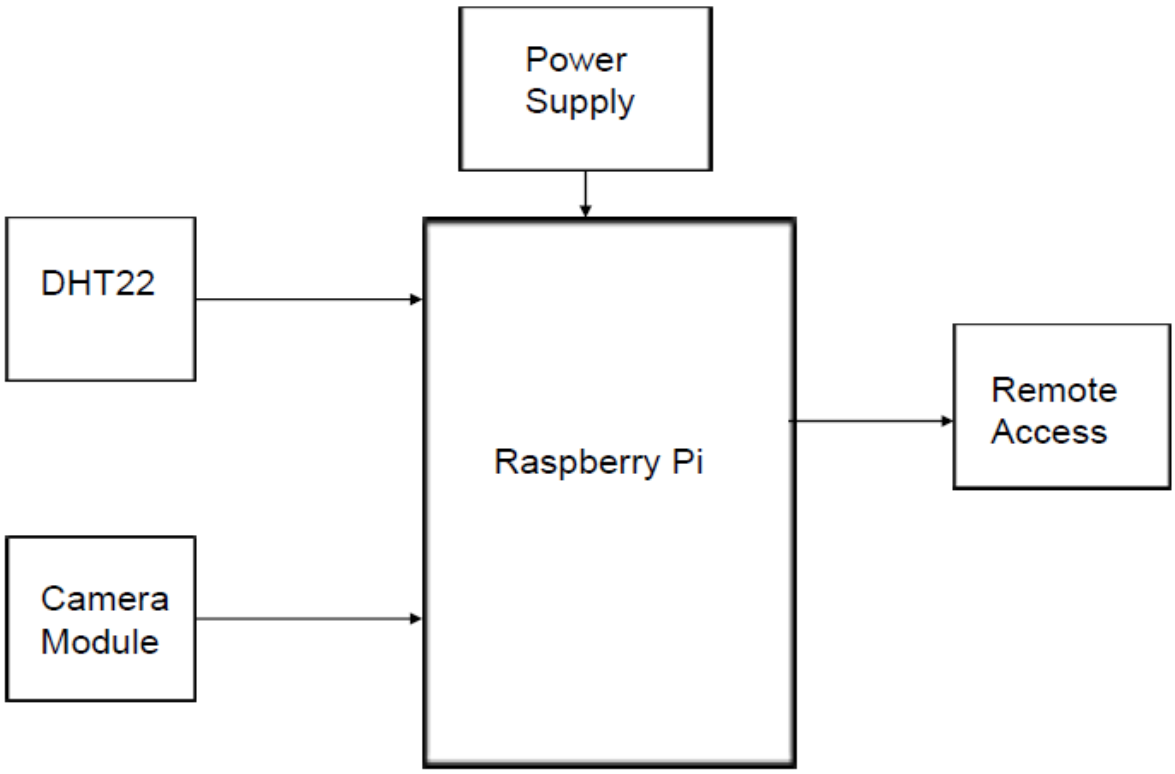


Fig 5: System architecture

3.3.5 Proposed system

The proposed method aims at continuously monitoring the real time temperature in a cost effective way by polling sensor at fixed interval of time. Here the monitoring node is raspberry pi. The Sensor utilized here is DHT22 Temperature/Humidity Sensor. The sensor is connected

to the raspberry pi kit using jumper wire.The raspberry pi kit can be used to store and display the real time temperature and relative humidity. The raspberry kit is programmed using python language. The block diagram of the proposed method is shown in figure below. The temperature is displayed in degree celsius or Fahrenheit as required. To know the current temperature and relative humidity at remote location, the user can log on raspberry Pi with all mentioned steps and programming. Raspberry pi processed data will be updated continuously as per given input command. User can also get to know the stored data on timely basis but here in this project we are not doing this. We only dsiplay the the real time temperature paarmeter qith a video.

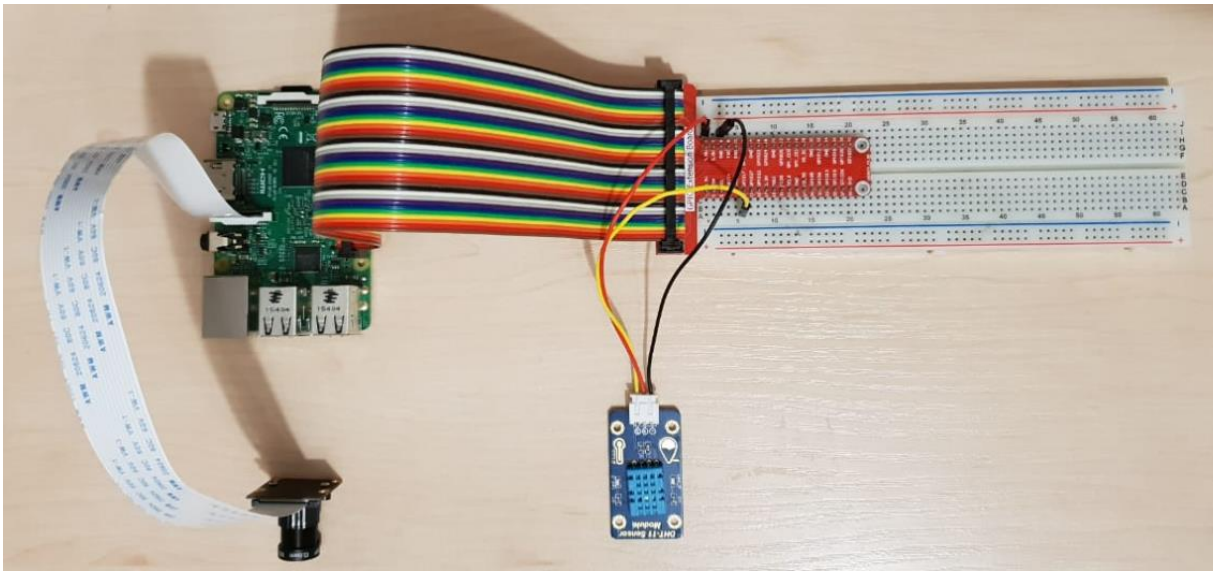


Fig 6: Complete setup of Proposed System

3.3 Software

First of all Raspberry pi has to be prepared for that we need to dowload Raspbian Jessie . It's an operating system manager that makes it easy to download, install, and set up your Raspberry Pi. When you first boot up Raspbian Jessie, you'll get a selection of OS to choose from.It makes getting started with Pi easy, and includes a bunch of different operating systems to choose from.The Raspbian is the official operating system of the Raspberry Pi.Raspbian is a version of Linux built operating system build specifically for the Raspberry Pi. Here raspberrypi 3 model b has been used.

Jessie installation:

1. Insert an SD card that is minimum of 8GB or greater in size into your computer.

2. Format the SD card using the platform-specific instructions below:

i.Download the SD Association's Formatting Tool from https://www.sdcard.org/downloads/formatter_4/eula_wi ndows/

ii. Install and run the Formatting Tool on your machine

iii. Set "FORMAT SIZE ADJUSTMENT" option to "ON" in the "Options" menu

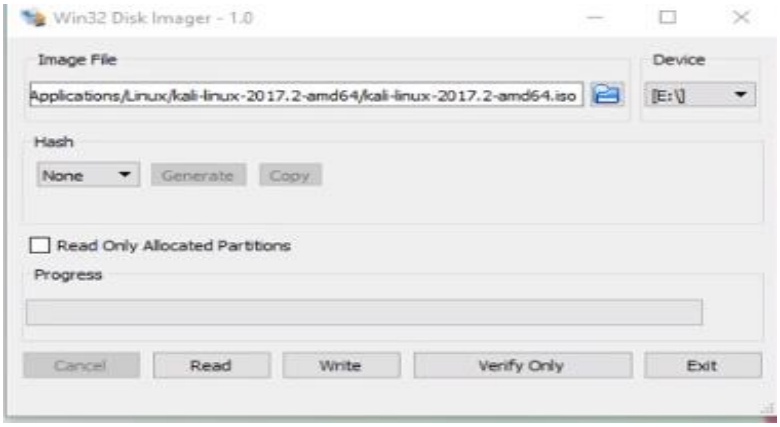
iv. Check that the SD card you inserted matches the one selected by the Tool

v. Click the "Format" button 3. Extract the files contained in this Raspbian zip file.

4. Copy the extracted files onto the SD card that you just formatted so that this file is at the root directory of the SD card. Please note that in some cases it may extract the files into a folder, if this is the case then please copy across the files from inside the folder rather than the folder itself.

5. Insert the SD card into raspberry Pi and connect the power supply. Use Win32-DiskImager to boot the Image file to Raspberry.

URL:https://sourceforge.net/projects/win32diskimager/



In the box load the image raspberry stretch or raspberry jessie and click write button.

6. Connect raspberry pi to the monitor via HDMI cable.

7. Download the Python program that will poll the sensor(s) dht22.py dht22.py is used for monitoring temperature and humidity using the DHT22 sensor. Once your Raspberry Pi is up and running we setup Python to talk to the GPIO pins. The GPIO pins are our interface to the DHT22 Temperature/Humidity sensor. Not all the libraries we need to make this project are pre-loaded on the Raspberry Pi. We require the Adafruit GPIO Python library and the Adafruit DHT22 library.

8.Open the terminal in raspberry and type

a. $ sudo apt-get update

b.$ sudo apt-get upgrade

c.$ sudo apt-get install tightvncserver.

9. Now install flask using the command

a. sudo pip install flask

Python is used for current application as it has following advantages:

Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. Python is designed to be highly readable . Python is a simplest, dynamic, interpreted, object oriented language. Python interpreters allowing Python code to run on a wide variety of system.

FOR SECURED SERVER HOST (SSH connection):

In linux:

1.Use Fing mobile app from android playstore we can it we will get the Ip address of raspberry

2.ssh pi@<ip-address of pi>login:pi

password:raspberry (login and password is default for all)

For Windows:

Download putty and tightvnc server

putty :https://www.putty.org/

Tightvncserver: https://www.tightvnc.com/licensing-tvnserver.php

In putty type the ip address in session

select ssh→X11→ enable→ save the port → click open

Login:pi

Password:raspberry

Activate motion in raspberry pi:

1. We will be using the terminal, so open the terminal on the Pi or connect to it via SSH.

2. To begin, first, update the Raspberry Pi, so you're running on the latest version.

```
sudo apt-get update
sudo apt-get upgrade
```

3.Depending on the version of Raspbian you're using you will need to do a few different steps.We start by removing libraries that may conflict with the newer packages. These may or may not already exist on your copy of Raspbian.

```
sudo apt-get remove libavcodec-extra-56 libavformat56 libavresample2 libavutil54
```

4. Download and install the following packages by inserting the following commands into the terminal.

```
wget https://github.com/ccrisan/motioneye/wiki/precompiled/ffmpeg_3.1.1-1_armhf.deb
sudo dpkg -i ffmpeg_3.1.1-1_armhf.deb
```

5. Now we need to install the following packages. We will need these as the Motion software relies on them.

```
sudo apt-get install curl libssl-dev libcurl4-openssl-dev libjpeg-dev libx264-142 libavcodec56 libavformat56 libmysqlclient18 libswscale3 libpq5
```

6. With those packages installed we can now grab the latest version of the motion software and install it. To do this run the following commands.

```
wget https://github.com/Motion-Project/motion/releases/download/release-4.0.1/pi_jessie_motion_4.0.1-1_armhf.deb
sudo dpkg -i pi_jessie_motion_4.0.1-1_armhf.deb
```

Raspbian Stretch

1. First, install the following packages. This command will work both on the full and lite version of Raspbian Stretch.

```
sudo apt-get install libmariadbclient18 libpq5 libavcodec57 libavformat57 libavutil55 libswscale4
```

2.Next, download the Motion deb file from the GitHub and install it using the dpkg command.

```
sudo wget https://github.com/Motion-Project/motion/releases/download/release-4.0.1/pi_stretch_motion_4.0.1-1_armhf.deb
sudo dpkg -i pi_stretch_motion_4.0.1-1_armhf.deb
```

That’s all you need to do before moving on to configuring Motion so that it will run on your Pi.

Configuring Motion

1. Now we need to make some edits to the configuration file (motion.conf)

```
sudo nano /etc/motion/motion.conf
```

2. Find the following lines and change them to the following.

- daemon on
- stream_localhost off

Note: Change the following two lines from on to off if you're having issues with the stream freezing whenever motion occurs.

- output_pictures off
- ffmpeg_output_movies off

Optional (Don't include the text in brackets)

- stream_maxrate 100 (This will allow for real-time streaming but requires more bandwidth & resources)
- framerate 100 (This will allow for 100 frames to be captured per second allowing for smoother video*)*
- width 640 (This changes the width of the image displayed)
- height 480 (This changes the height of the image displayed)

3. Now we need to setup up the daemon. First, we need to edit the *motion* file.

```
sudo nano /etc/default/motion
```

4. Find the following line and change it to the following:

```
start_motion_daemon=yes
```

5. Once you're done, simply save and exit by pressing *ctrl+x* then *y*.

6. Now make sure the camera is connected and run the following line:

```
sudo service motion start
```

7. If you need to stop the service, simply run the following command:

```
sudo service motion stop
```

8. Now you should be able to check out the Webcam Stream at the IP address of our Pi so in your browser go to the following address:

```
192.168.1.103:8081
```

9. If the webpage isn't loading, try restarting the service.

```
sudo service motion restart
```

10. If you're using a Raspberry Pi camera, then you will need to do a few extra steps that are mentioned below.

1.First make sure the camera is switched on within raspi config, enter the following command and then enable the camera. (You will need to restart once you have done this).

```
sudo raspi-config
```

2. Now open up the modules file by entering the following line.

```
sudo nano /etc/modules
```

3. Enter the following line at the bottom of the file if it doesn't already exist, once done save & exit by pressing ctrl+x then y.

```
bcm2835-v4l2
```

4. Now reboot the Pi, and the stream should now work.

```
sudo reboot
```

5. You should now be able to access the Raspberry Pi webcam stream by going to the Pi's IP address on port 8081.

```
192.168.1.103:8081
```

You should now have a fully working Raspberry Pi camera server that is accessible within your local network. If you want to allow external access to the camera, then please follow my instructions below.

3.3.1 Execution code of DHT22:

```
from flask import Flask
from flask import render_template, request
import RPi.GPIO as GPIO
import time
import os
import sys
from signal import pause
import Adafruit_DHT as dht

app = Flask(__name__,template_folder='templates')
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

#read data using pin17
h,t = dht.read_retry(dht.DHT22,17)
Temp='{0:0.1f}*C'.format(t)
time.sleep(1)
print "DOne"
a=1
@app.route("/")
def index():
        return render_template('index.html',Temp=Temp)
        # Read Temperature pin 17
        #h,t = dht.read_retry(dht.DHT22,17)
        #Temp=('{0:0.1f}*C'.format(t))
        #if t is not None:
                #print('Temp={0:0.1f}*C  Humidity={1:0.1f}%' .format(t, h))
                #return render_template('index.html')
                #Temp='{0:0.1f}*C'.format(t)

if __name__ == "__main__":
        print "Start"
        debug = True
        app.run(host='192.168.0.102',port=2222)
```

3.3.2 Template for webpages:

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
</head>
<body>
<img src="http://192.168.0.102:8081/" /> <!--Enter the IP Address of your Raspberry Pi-->
<div style="float:right">
</div>
<div style=" height:720px; width:480px; float:right;">
<center>
<h1><span style="color:#5C5C5C;">Temperature</span><span style="color:#139442">
pi</span></h1>
<h2>DHT_22</h2><br><br>
</div>
<h2>Current Temperature : {{Temp}}<h2>
<footer>
    <p>&copy;TUC Group:10 <i class="icon-beaker"></i></p>
</footer>
</body>
</html>
```

4 Description of files

Describe the provided documents within this project containing

- Folder structure
- Filenames + content

project_documentation.docx	project documentation
schedule_presentation.docx	schedule presentation slides
final_presentation.pptx	final presentation slides
additional_files	
>diagrams	diagrams and visualisations
>layouts	schematics and circuit diagrams
>source_code	source code for Arduino IDE
>>documentation	doxygen source code documentation

5 Results

By doing all the above mentioned setup and programming we are able to monitoring the video having real time temperature and humidity changes along with the date and time. The below pictures is taken directly from the end results of varying temperatures. For getting range of temperature we have used hair drier to inject hot air in DST sensor, and by refreshing the web page we can have the real time change of temperature and humidity as required in the project scope. We have taken 3 state of temperature changes. The outputs are shown below.

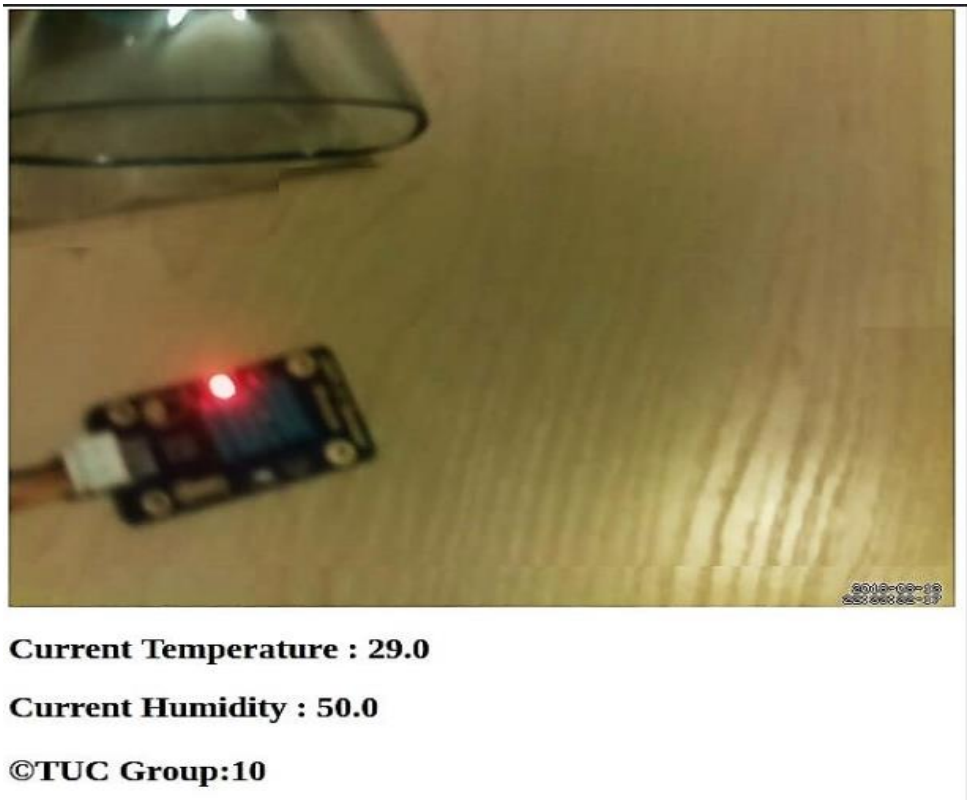
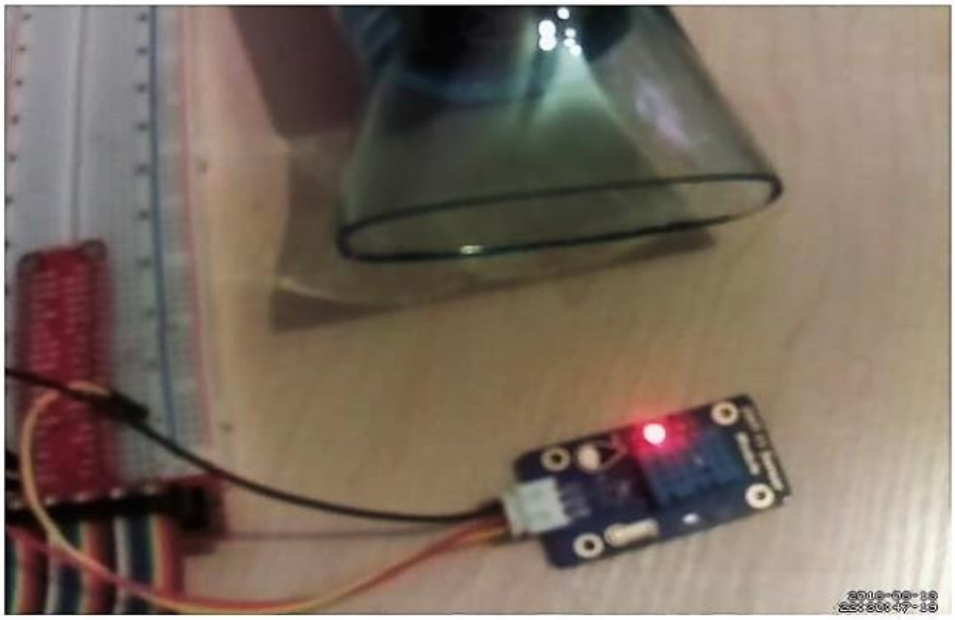


Fig 7: Real Time readings at normal temperature



Current Temperature : 48.0

Current Humidity : 18.0

©TUC Group:10

Fig 8: Real Time readings after inserting hot air bolw by hair drier

Current Temperature : 50.0

Current Humidity : 19.0

©TUC Group:10

Fig 9: Real Time readings after inserting more hot air bolw by hair drier

6 Conclusion

The smart surveillance system has been aimed to design in such a way that it can fulfil the needs of the user for particular surveillance area. It has countless applications and can be used in different environments and scenarios. For instance, at one scenario it can be used by any person working in industry to be aware of the activity being happened at their own working places, in their absence, while at another instance it can be used for spy purposes at bank lockers, storage houses. Another application is to provide information to the user about what is happening in surveillance area by notification

7 References

[1] International Journal of Advanced Research(IJAR) in Computer and Communication Engineering-Vol. 5-October 2016` Raspberry Pi Based Weather Monitoring System Meetali V. Rasal, Prof. Jaideep G. Rana.

[2] Using Raspberry Pi to sense temperature and relative humidity .

Volume: 04 Issue: 02 | Feb -2017 International Research Journal of Engineering and Technology

[3] DHT humidity sensing on Raspberry Pi

https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/overview

[4] Flask web development

http://flask.pocoo.org/

[6] Raspberry Pi Projects Book by Mike Cook and W. Andrew Robinson

[7] Learning Python with Raspberry Pi Book by Alex Bradbury and Ben Everard