Home Temperature Monitoring System

Bachelor of Technology

In

Electronics and Telecommunications Engineering

By

Raunaq Singh Sahni (D073)
Vipul Ajmera (D077)
Akshay Chhabra (D078)

Under the supervision of

Prof. Sharad Wagh
(Designation, EXTC Department, MPSTME)

SVKM's NMIMS University
(Deemed-to-be University)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING
Vile Parle (W), Mumbai-56

2013

CERTIFICATE

This is to certify that the project entitled **"Home Temperature Monitoring System"**, has been done by **Mr. Raunaq Sing Sahni, Mr. Vipul Ajmera & Mr. Akshay Chhabra** under my guidance and supervision for the degree of Bachelor of Technology in Electronics and Telecommunication of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

_____ _____

Name of Internal guide Examiner

(Intenal Guide)

Date

Place: Mumbai

_____

Prof. Vaishali Kulkarni

(HOD EXTC)

ACKNOWLEDGEMENT

Starting of with this project was never going to be an easy task, but we as a team worked on this together which made this possible. We'd like to thank our professors Mr. Sharad Wagh and who guided us throughout giving us ways to use different controllers and a proper method for preparing this report. A major level of appreciation for our group member Raunaq Singh Sahni who put in his vast experience and knowledge into this and I can say without any doubt that this project would have never worked out without him. Our lab assistants who adjusted with us and let us work freely and also lastly our college for giving us this opportunity to work on such a level which has proven to be a very good learning curve for our next year.

Akshay Chhabra
B.Tech (EXTC)
Roll No. D078

ABSTRACT

The project is a home temperature monitoring system that is controlled by a Pi, which can be accessed from anywhere there is an active Internet connection. It has worldwide access. There are many components to this project, including the temperature sensor, programming the Pi, setting up a NoSQL database, and establishing the Internet connectivity using usb tethering. We will separately discuss each component and provide clear steps on how we went about this.

Table of Contents

NO.	TOPIC	PAGE NUMBER
1	INTRODUCTION	**1**
2	LITERATURE REVIEW	**2**
3	PROBLEM STATEMENT	**3**
4	PROJECT DESCRIPTION	**6**
5	IMPLEMENTATION	**11**
6	RESULT	**14**
7	CONCULSION	**15**
8	REFERENCES	**16**

List of Figures

FIGURE NUMBER	NAME
1	TEMPERATURE SENSORS
2	RASPBERRY PI 2
3	GPIO PIN CONFIG
4	GPIO PINS
5	PI DESCRIPTION
6	BREADBOARD
7	PI COBLER
8	JUMPER WIRE
9	RIBBON CABLE
10	LAYOUT

List of Tables

- PIN DESCRIPTION OF DS18B120

INTRODUCTION

Our Home Temperature Monitoring System uses a Raspberry Pi as the main controller and DS18B20 temperature sensor which all in all records temperature in any time interval needed and this is displayed online and can be accessed anywhere with an internet connection on the webpage http://pi-temperature.appspot.com. The entire programming was done in python and we used Google App engine's zero system administration infrastructure to make the webapp that stores the readings displays them. The Occidentalis Linux distribution for Raspberry Pi (and Raspbian as of Dec 2012) includes support for the DS18B20 1-wire temperature sensor. These sensors come in a small three pin package like a transistor and are accurate digital devices. Since the Raspberry Pi has no ADC (Analog to Digital Converter), it cannot directly use an analog temperature sensor like the TMP36, making the DS18B20 a good choice for temperature sensing.

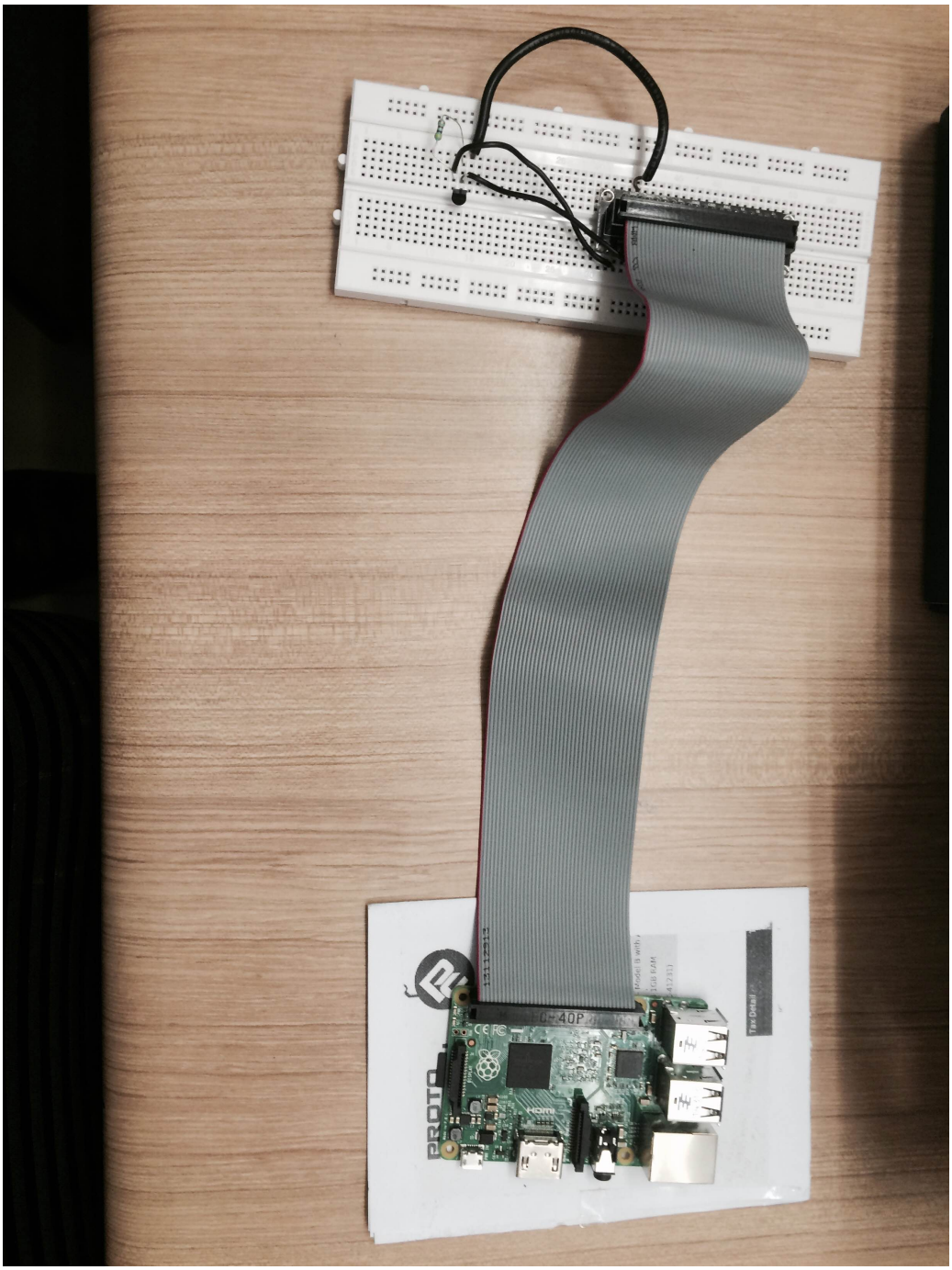
We have next the literature survey in which we show the research we did for the project. Next is the where we show our code and block diagrams in the problem statement. After which we describe the entire project and then concluding it with future scopes for this project.

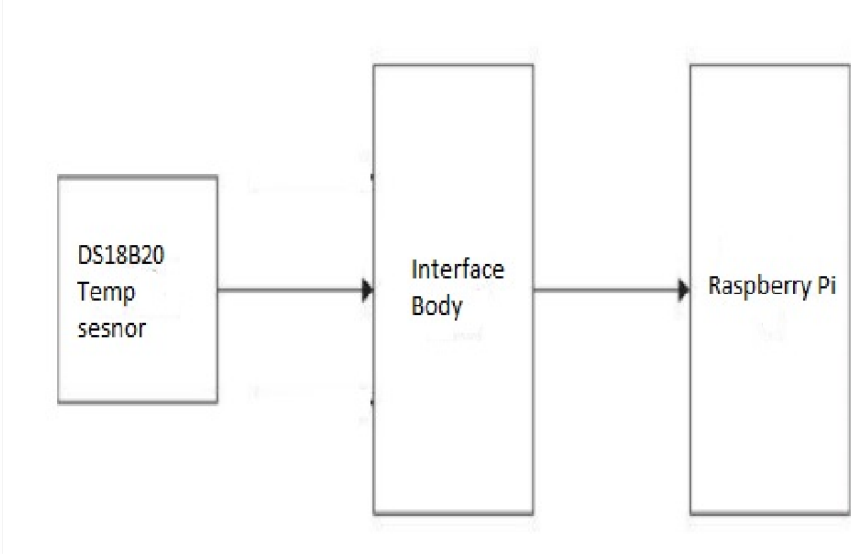
Literature Survey

We came across this topic in the book *The Internet of Things Projects, by Donald Norris, Mc Graw Hill*. The book explained how to make a Home temperature monitoring system using an analog temperature sensor. The basic understanding required to implement the project came from this source.

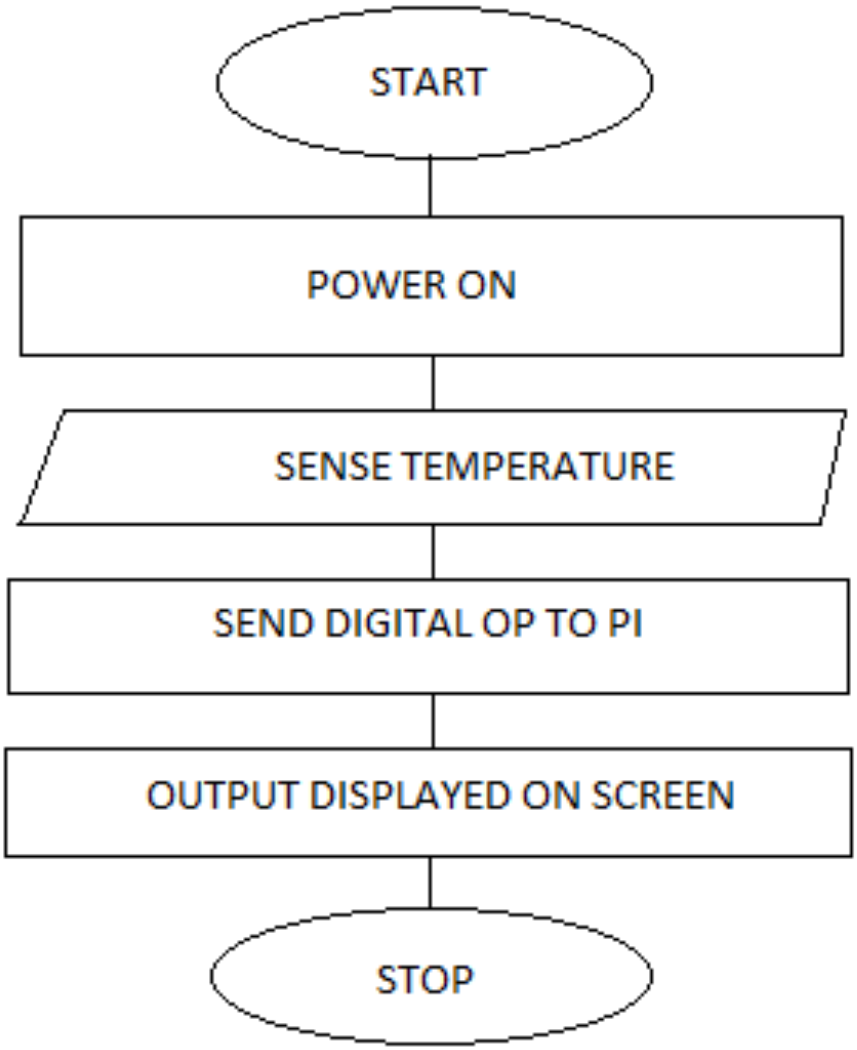
We then used adafruits web page for the system as our prvious components were analog and converting them to digital made the work easier.

Problem Statement





The above is the flow diagram of the home temperature monitoring system in which first the sensor connected to pi using a breadboard and bus senses the temperature and sends through the interfacing system to pi which records and shows the output on the screen.


START
POWER ON
SENSE TEMPERATURE
SEND DIGITAL OP TO PI
OUTPUT DISPLAYED ON SCREEN
STOP

Project Description

Hardware:

1) Temperature Sensor

Fig 1: DS18B120 sensors

Table 1: pins description of DS18B20

Pin Number	Description	Remarks
1	GND	Ground
2	DQ	Data input output pins
3	Vdd	Power supply volage

It is the voltage given out by this sensor that accounts for the temperature reading. Although the DS18B20 just looks like a regular transistor, there is actually quite a lot going on inside.The chip includes the special 1-wire serial interface as well as control logic and the temperature sensor itself. Its output pin sends digital messages and Raspbian/Occidentalis includes an interface to read those messages. Power for reading, writing and performing temperature conversions can be derived from data itself without need of external power source.

2) Raspberry Pi 2

Fig 2 : Raspberry pi 2

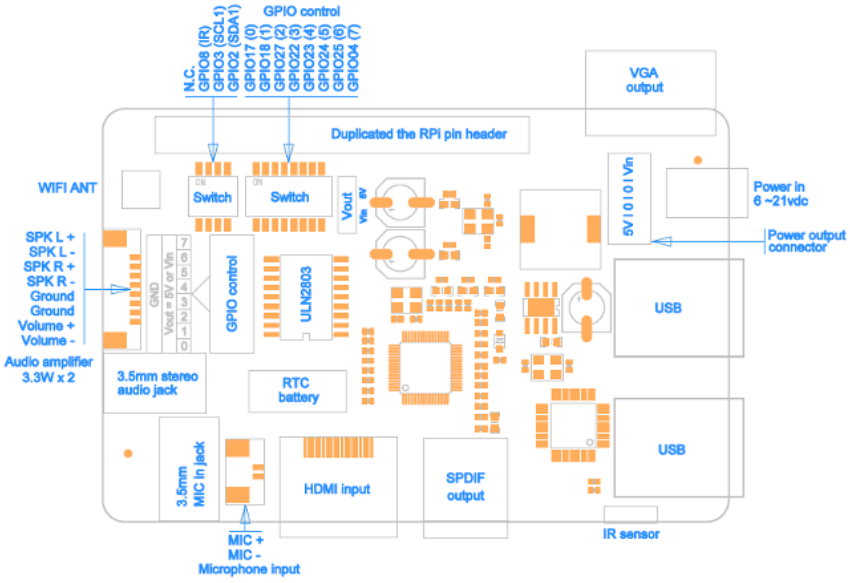


Fig 3 : Pi Descriptiom

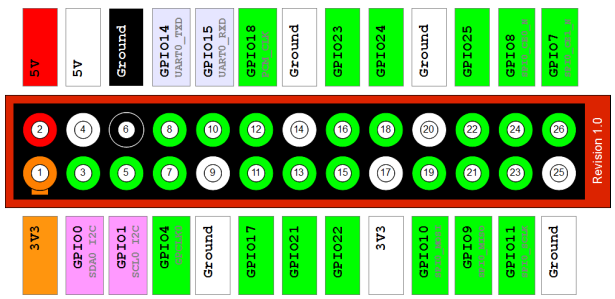


Fig 4: GPIO pin config

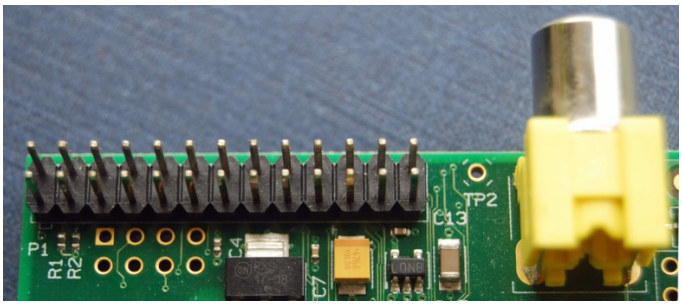


Fig 5 : GPIO pins

3) Breadboard

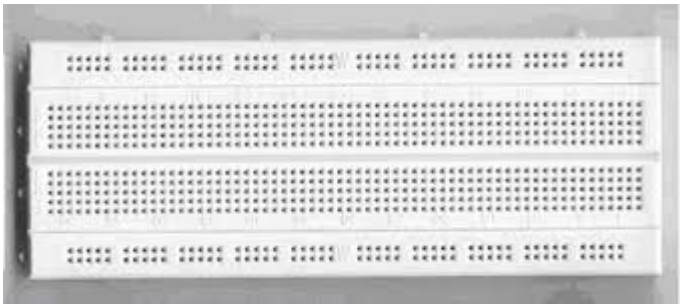


Fig 6 : Breadboard

4) Pi Cobbler

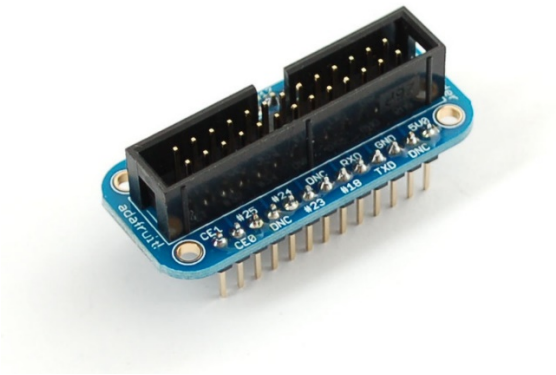


Fig 7 : Pi cobbler

5) Jumper wires

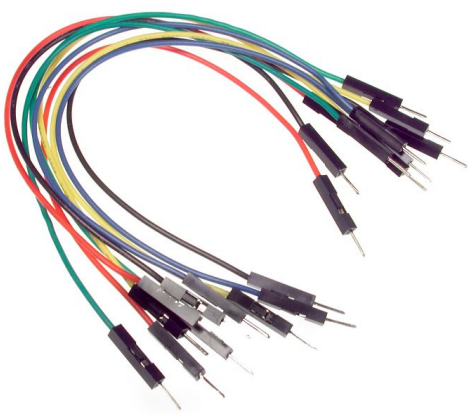


Fig 8 : Juper wires

6) Ribbon Cable

Fig 9 : Ribbon Cable

7) Layout

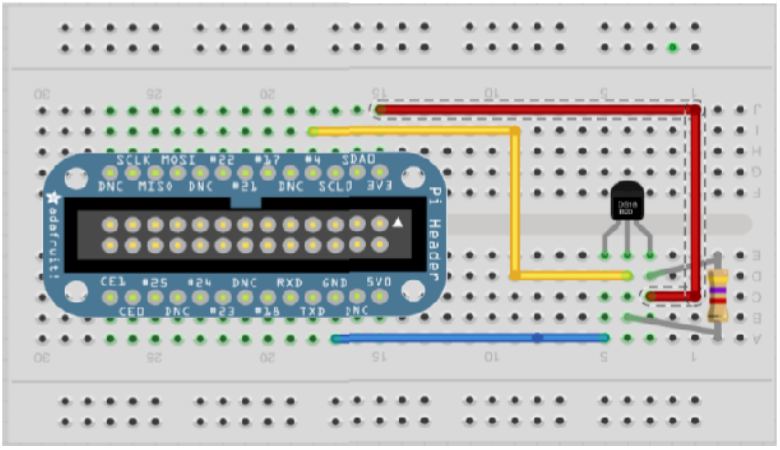


Fig 10 : Breadboard Layout

The breadboard layout for just the basic DS18B20 is shown above.The DS18B20 "1-wire" sensors can be connected in parallel - unlike nearly any other sensor sold! All sensors should share the same pins, but you only need one 4.7K resistor for all of them.The resistor is used as a 'pullup' for the data-line, and is required to keep the data transfer stable

Implementation

The following steps have been taken by us to implement this project:

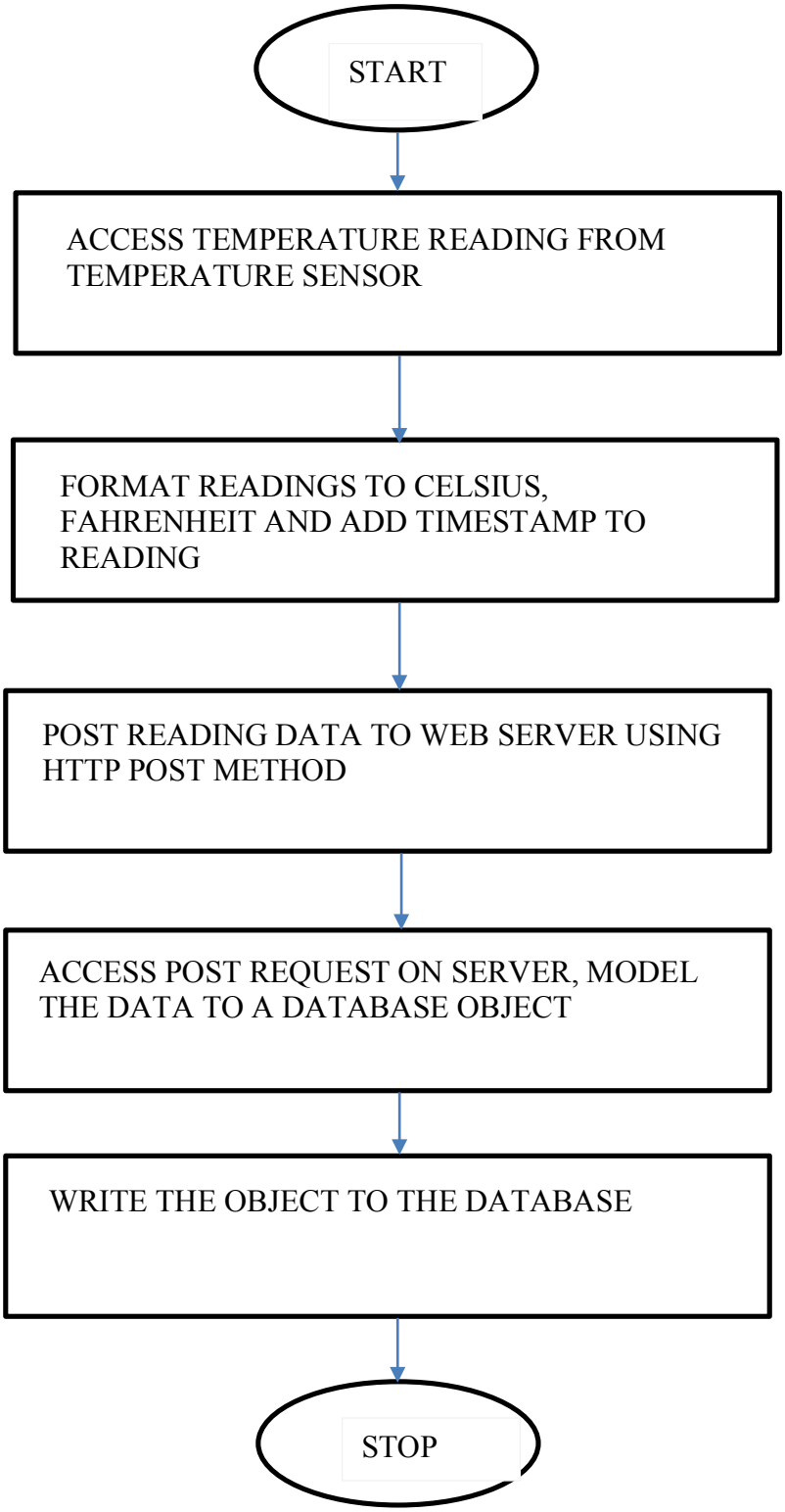
1. Setup Raspberry Pi – we set up the raspberry pi using the bootable NOOBS sdcard and installed the latest Raspbion Linux Distribution as the OS of the microcontroller. After setting it up, we enabled all hardware peripherals required by configuring the pi on the terminal.

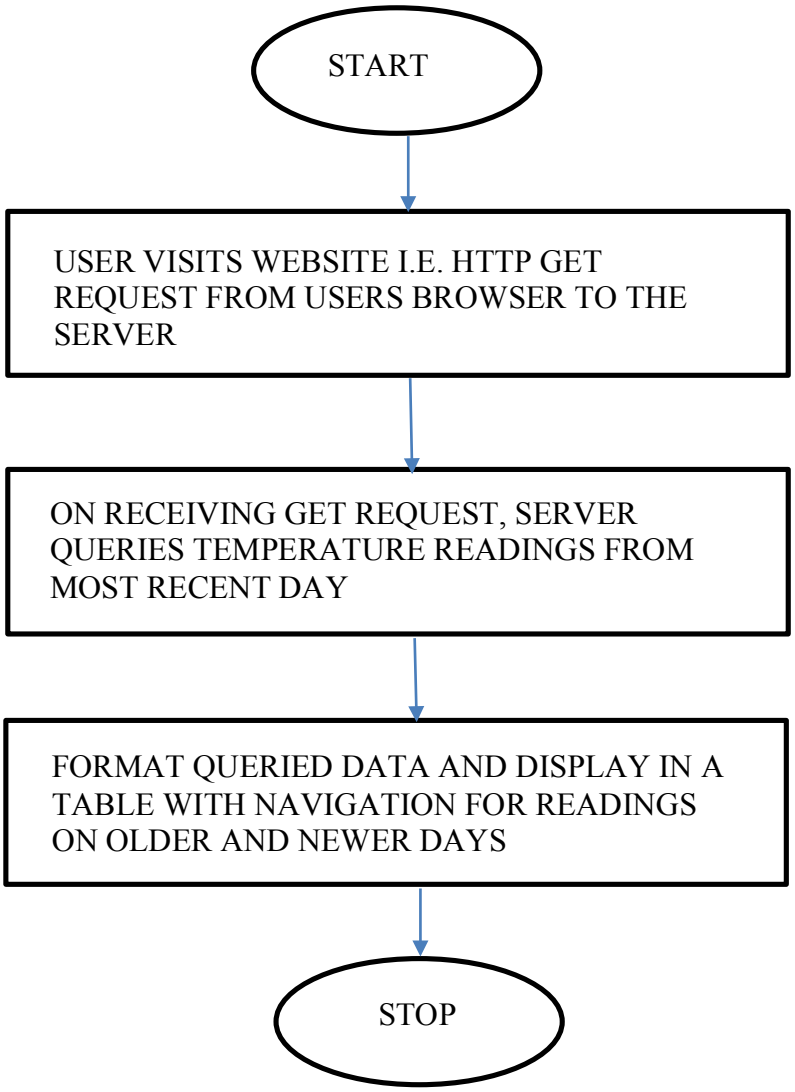
2. Setup up USB Tethering internet – in order to tether internet on the pi using an iPhone we had to install some required packages using Linux's package manager apt-get. We then mount the iphone to the pi using the packages installed. We also wrote an executable script to automate the mounting of the iPhone everytime it's plugged in so that the pi connects to the internet automatically after plugging it in.

3. Sensing the temperature – Using the components and the circuit diagram for our project, we implemented the circuit and wrote the necessary code to read the temperature from the sensor using Python 2.7. The sensor DS18B20 provides an 8 to 12 bit digital output corresponding to the temperature sensed. This temperature can be accessed on the pi from the device number of the temperature sensor. Our temperature reading program accesses this device and formats the output in the required way.

4. Getting the readings online – In order to display the temperature on http://pi-temperature.appspot.com, we have written a web application in Python and hosted it using Google's App Engine platform. The pi sends the reading to the server using an HTTP Post method. The HTTP post method request sent has been implemented using an external library for Python. Our web app receives this data and writes it to the NoSQL database. When a user visits the website, our app aggregates the data saved on the database according to dates and displays it in a table. Many external libraries and frameworks like webapp2, jinja2, and bootstrap css framework have been used to achieve the result. Expertise required to implement this app were acquired by previous web app development experience.

Software : 1)Posting temperature readings to database




START
USER VISITS WEBSITE I.E. HTTP GET REQUEST FROM USERS BROWSER TO THE SERVER
ON RECEIVING GET REQUEST, SERVER QUERIES TEMPERATURE READINGS FROM MOST RECENT DAY
FORMAT QUERIED DATA AND DISPLAY IN A TABLE WITH NAVIGATION FOR READINGS ON OLDER AND NEWER DAYS
STOP

RESULT

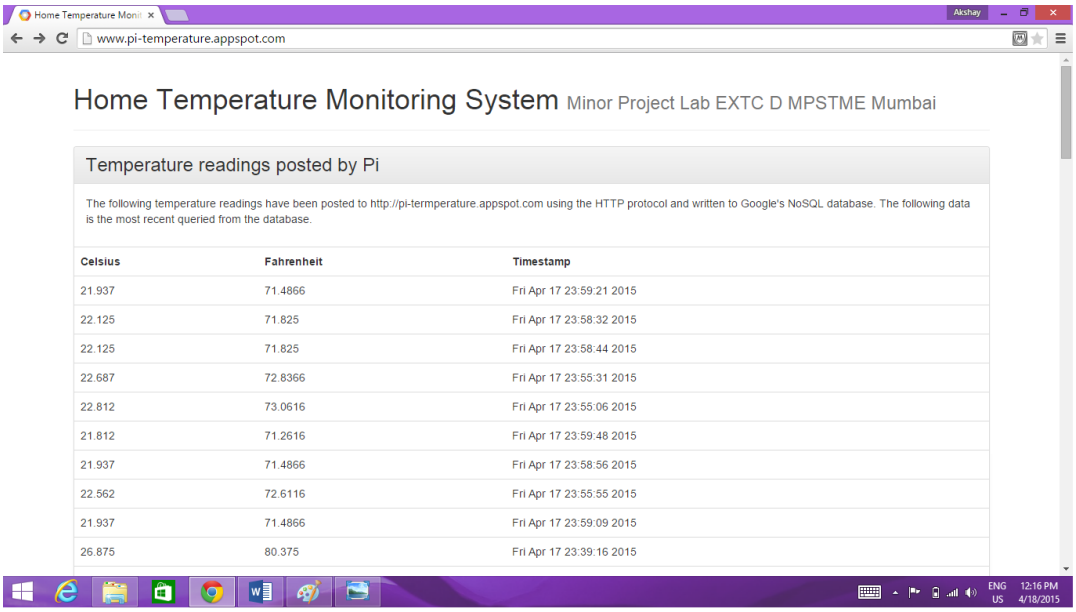


Fig 12: Final Output

In the figure we show our final output as seen on the webpage

Conclusion

By this experiment the Home Temperature Monitoring System was created and it successfully reads temperature and further maintains a record of it on the webpage which can keep udating itself automatically in the time interval specified.

Refferences

- Intenet of Things: Projects, by Donald Norris, Mc Graw Hill 2015
- Adafruit Raspberry Pi Lesson 11, created by Simon Monk, 2015
- Google App Engine Developer Documentation
- Raspberry Pi Documentation
- Bootstrap Framework Documentation
- Jinaj2 Documentation