Getting Serial Data from Arduino to Raspberry Pi and saving it in MySQl database & Explaining How to run CoAP server on Raspberry pi:

In this tutorial I'm going to tell you how we can get the sensor data from Arduino Uno and sending it to the Raspberry pi through UART. Mostly We are going to do two important things. First we will use a CoAP server For making a GET request which will run on raspberry pi and in our CoAP browser we will get the DHT22 data immediately. And the second thing is we are going to make a mysql database in raspberry pi to store the value of DHT22 sensor.

**What is CoAP:**

**Constrained Application Protocol** (**CoAP**) is an Internet Application Protocol for constrained devices. It enables those constrained devices to communicate with the wider Internet using similar protocols. CoAP is designed for use between Devices on the same constrained network, between Devices and general nodes on the Internet, and between Devices on different constrained networks both joined by an Internet.

CoAP is designed to easily translate to [HTTP](https://en.wikipedia.org/wiki/HTTP) for simplified integration with the web, while also meeting specialized requirements such as [multicast](https://en.wikipedia.org/wiki/Multicast) support, very low overhead, and simplicity.

CoAP supports the basic methods of GET, POST, PUT, DELETE, which are easily mapped to HTTP.

Now here above was the brief description of CoAP. As I told you that you will get a lot of things about CoAP but as far as interfacing with real hardware is concerned there are very less stuff. So that's why I am going to tell you about this CoAP server and how we can interact with real hardware .

Step by Step we will follow the procedure:

1: In this step we will tell you that how we can run a CoAP server in our raspberry pi.

txThings Tutorial

In this tutorial, we will describe how to run a CoAP server on Raspberry Pi, and run the CoAP client on a PC. The CoAP server and clients are implemented using txThings, which is a Phyton implementation of CoAP.

# About txThings

txThings - CoAP library for Twisted framework

txThings is a Python implementation of Constrained Application Protocol (CoAP):

<http://tools.ietf.org/html/rfc7252>

txThings is based on Twisted - asynchronous I/O framework and networking engine written in Python.

<http://twistedmatrix.com/>

txThings uses MIT License (like Twisted itself).

<http://opensource.org/licenses/mit-license.php>

Copyright (c) 2012 Maciej Wasilak

<http://sixpinetrees.blogspot.com/>

# txThings Python Library Features

txThings has the following features:

* support for draft-ietf-core-coap-13 - including automatic piggyback/separate response handling. No caching support.
* support for draft-ietf-core-block-12 (no support for server initiative though - waiting for the resolution)
* limited support for RFC6690 (Core Link Format) - server only.

Other nice things:

* txThings works nicely on RaspberryPi
* txThings is compatible with [Kivy](http://kivy.org/) - brilliant new Python GUI library (I'll post some examples soon).
* txThings is fully asynchronous (thanks to twisted framework)

# txThings Installation HowTo

txThings is posted on [Github](https://github.com/siskin/txThings). The easiest way to get it is to clone the repository to your local machine using the command below:

*git clone git://github.com/siskin/txThings.git*

Library contains CoAP code (inside "iot" directory) and three examples:

* server.py - CoAP server that starts on localhost, port 5683 and hosts several resources
* client\_GET.py - example client which performs GET request to localhost, port 5683
* client\_PUT.py - example client which performs PUT request to localhost, port 5683

Client\_GET and client\_PUT both use port 61616 - to use them simultaneously change port number in one of the clients. Server will send blockwise responses for default settings. To use txThings you need Python 2.7 with Twisted installed (I suggest using the latest Twisted version, but older releases also work - tested with 11.1).

# Setting up Raspberry-Pi

1. Install OS on Raspberry Pi (set the appropriate raspi-config for all,including ssh and desktop boot)
2. Install setup tools for python (such as pip)

[https://pip.pypa.io/en/stable/installing.html#install-pip](https://pip.pypa.io/en/stable/installing.html" \l "install-pip)

1. Install Twisted on Raspberry Pi

*sudo apt-get install python-pip*

*pip install twisted==15.1.0*

1. Install txThings on Raspberry Pi

*pip install txThings*

2. Now this is the time to move on Arduino. Here I've connected DHT22 sensor to an arduino and our arduino is connected to our raspberry pi(which is working here as an IOT gateway or server ) through USB cable to one of its USB port. But before going to connect Arduino to our raspberry pi we have to put a Arduino snippet for DHT22 interfacing to our arduino. Since we are using Raspberry pi remotely in ubuntu.

Connect your arduino to your PC to upload this code(Arduino code) Now just unplug your arduino from PC and connect it to your pi

3. Now we need to install some of the things to get data in raspberry pi

First one is Python-serial and then python-mysqldb

Now This URI we have to type in Mozilla Firefox

caop://<Raspberry pi IP Address>:5683

for that we need to do a little modification in our server.py code which we got while installing txThings.

You just check your server.py code and edit accordingly. Here some of the things I highlighted

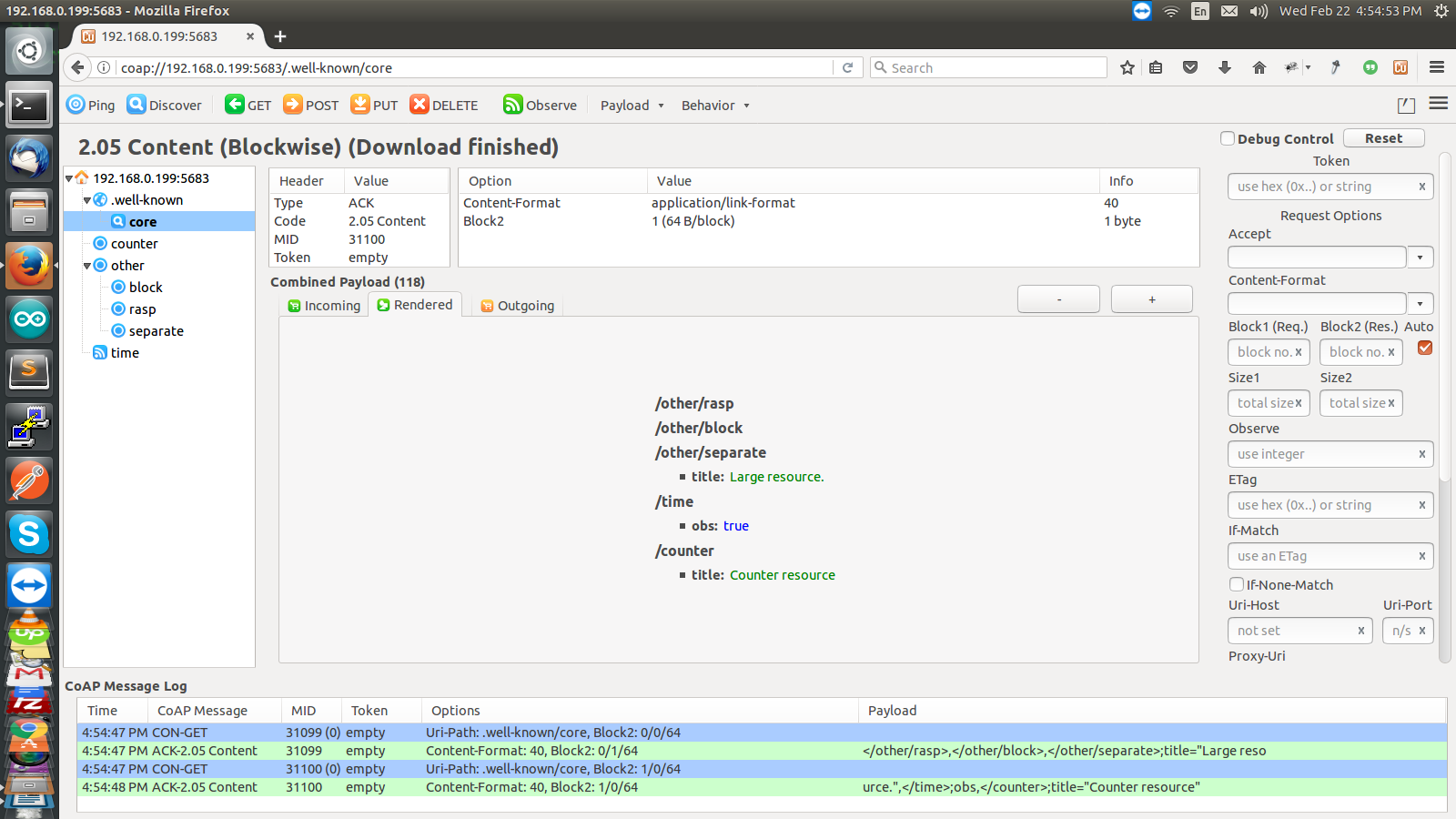
Now we need to run our server.py code

It'll take a little time

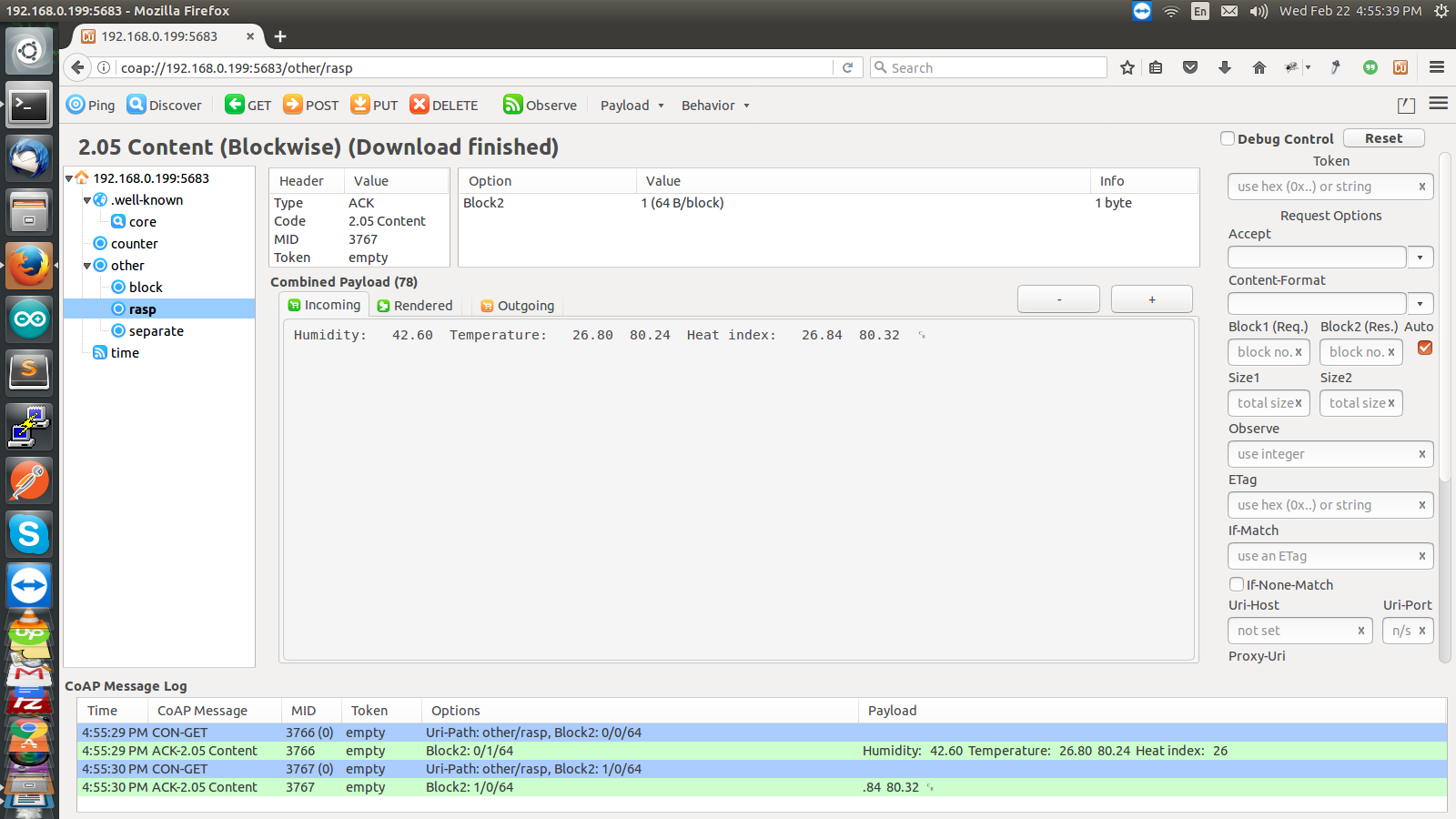
Here(Figure 1)you can see the temperature and humidity in Copper on firefox

# Run Copper on Firefox

The Copper (Cu) CoAP user-agent is an add-on for the Firefox Web browser. It allows browsing, bookmarking, and direct interaction with CoAP resources. Simply enter a CoAP URI into the address bar.

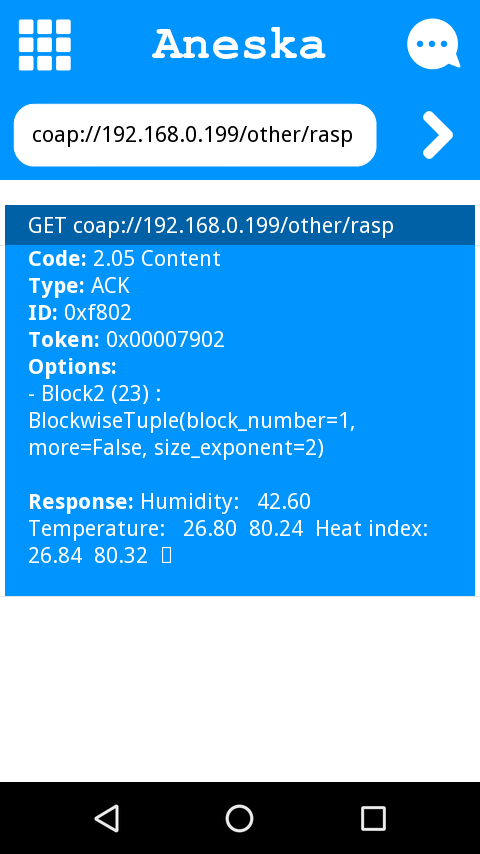


**Figure** 1**. CORE**



**Figure 1. Humidity Temperature and Heat Index**

For getting the data in our smartphone we need to install Aneska. At the end section of the video you can see how we can get data in our smartphone.



**Figure 2. Humidity and Temperature In Smartphone**

**The second part of this project is creating a database of DHT22 data:**

For that you have to install Apache Mysql and php and then save data in phpmyadmin.

You can easily get about these installation in many links. So I'm not going to demonstrate that. In short you have to set up a LAMP server in your raspberry pi. I am assuming you have install these necessary things.

So Lat's get started

Go to localhost/phpmyadmin or <pi IP address>/phpmyadmin if you are using pi remotely.

Create Database give it some name

then create table for that

Here I've made 6 columns

For getting the data in database from arduino serial out we have to write a small script in pi.

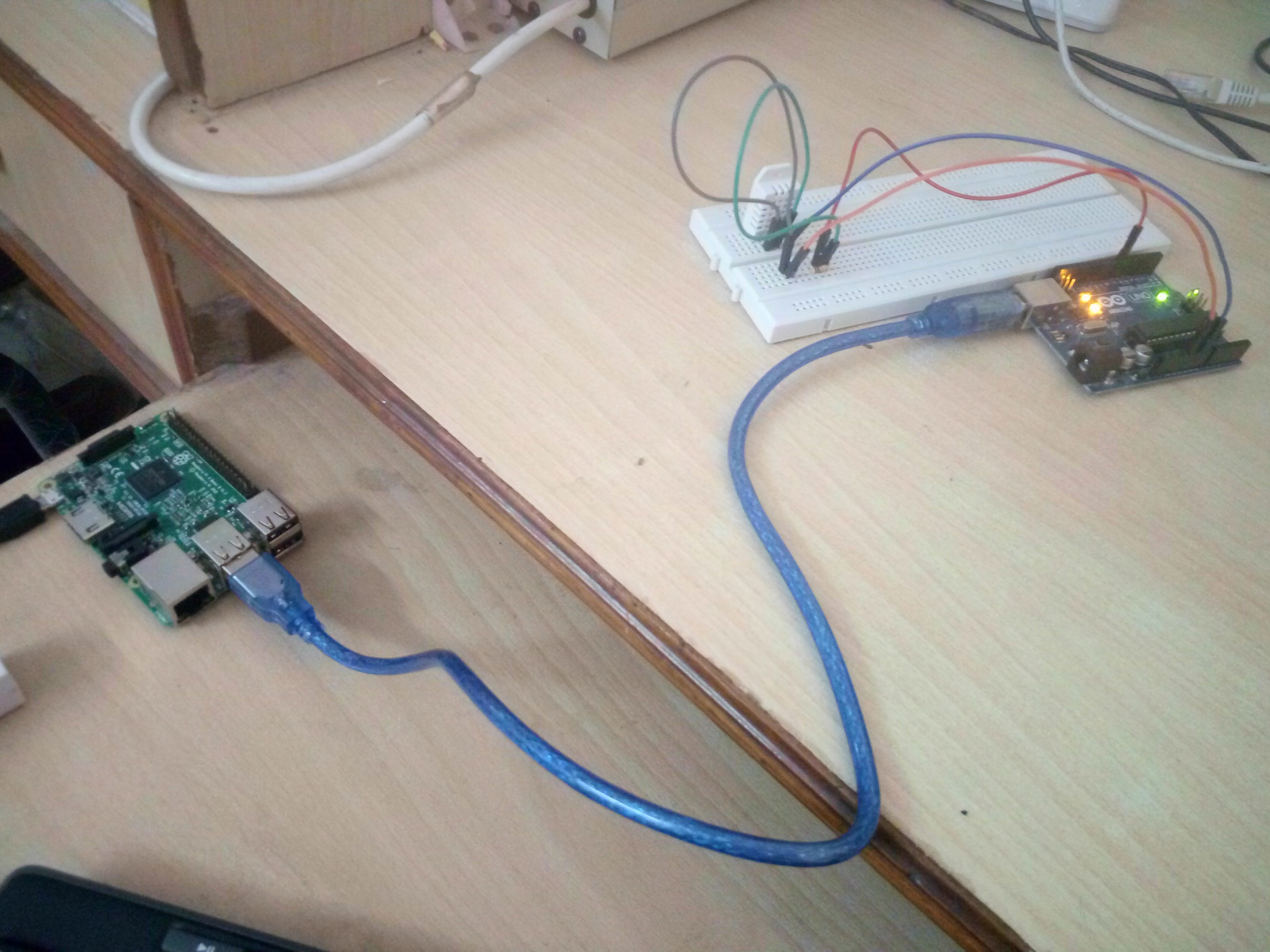
This is the script(You can find all the files in our github page)for getting data in mysql datbase from arduino.

If I run this script I can see the updated data in database

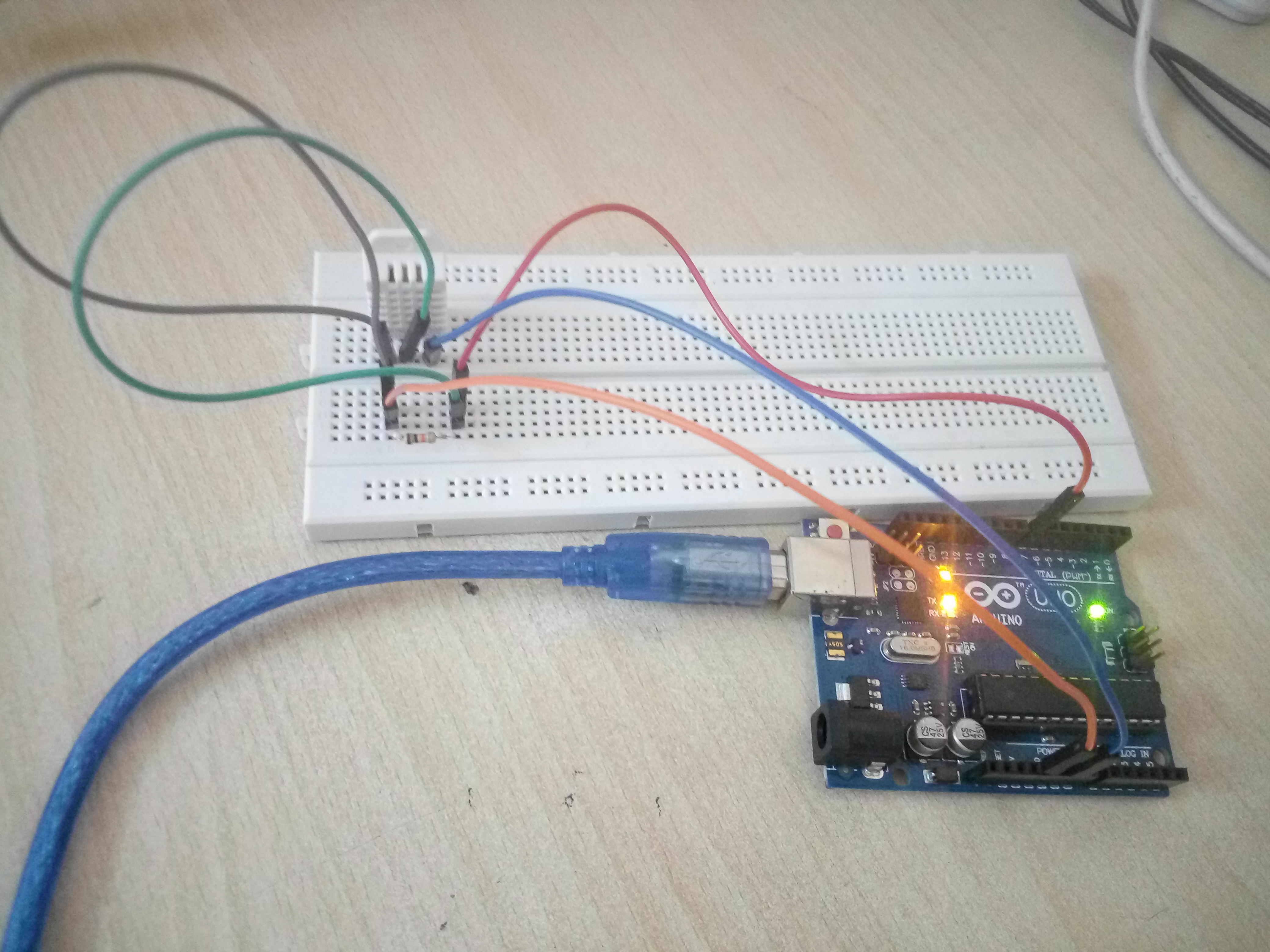
But if we want this thing automatically let's say for every single minute then we have to edit crontab -e

All the process Regarding this project is explained in our video

**circuit Images of our project:**

****

**Figure 3. Arduino Interfacing with DHT22 and connected to Pi with USB**



**Figure 4. Arduino Connection with DHT22**