**Contents**

**1.System Requirements (or Prerequisites)**

Software and hardware requirements

**2.Overview of System or Product Features**

**3.Minimal Quick Start Configuration**

Python (Version 3.9 or greater)

Firmware Patch Server with Linux Operating

Repository Server with Linux Operating System

Raspberry Pi (Model B+) with Raspbian Operating System.

DHT-11 Sensor.

Paramiko Framework of Python

Paho-Mqtt library

Mqtt Broker – Eclipse Mosquitto

Adafruit Libraries for DHT-11 sensor.

**4.Set-Up Configurations**

Machine 1 Setup

Machine 2 Setup

Machine 3 Setup

End Nodes

**5.Maintenance and Error Messages and Troubleshooting**

Python script debugging

**1) System Requirements (or Prerequisites)**

* Windows Operating System (Version 10 or greater)
* Linux Operating System (Version 18.04 or greater)
* Python (Version 3.9 or greater)
* Firmware Patch Server with Linux Operating System.
* Repository Server with Linux Operating System.
* Raspberry Pi (Model B+) with Raspbian Operating System.
* 16 GB Memory Card
* DHT-11 Sensor.
* Paramiko Framework of Python for Secure File Transfer Protocol.
* Paho-Mqtt library to access the Mqtt functionalities through Python program.
* Mqtt Broker – Eclipse Mosquitto
* Adafruit Libraries for DHT-11 sensor.

**2) Overview of System or Product Features**

* **Python (Version 3.9 or greater)**

Python latest version compatible with your device configuration must be downloaded.link to download and instruction is given below.

<https://www.python.org/downloads/release/python-390/>

* **Firmware Patch Server with Linux Operating System**

Firmware patch server will have linux operating system for network cofigurations and linux commands.Installation of latest version of ubuntu on your system.A video link is given below downloading instructions.

<https://ubuntu.com/desktop>

* **Raspberry Pi (Model B+) with Raspbian Operating System**

The Raspberry Pi 3B+ features dual-band IEEE 802.11b/g/n/ac WiFi, Bluetooth 4.2, and Gigabit Ethernet (limited to approximately 300 Mbit/s by the USB 2.0 bus between it and the SoC). The Raspberry Pi 4 has full gigabit Ethernet (throughput is not limited as it is not funnelled via the USB chip.

Instructions to setup the Raspberry pi 3B+ are given in the link below.

<https://www.techrepublic.com/article/how-to-set-up-your-raspberry-pi-3-model-b/>

* **DHT-11 Sensor**

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels.The datasheet for DHT-11 sensor is given below.

<http://robocraft.ru/files/datasheet/DHT11.pdf>

* **Paramiko Framework**

Paramiko is a Python (2.7, 3.4+) implementation of the SSHv2 protocol, providing both client and server functionality. While it leverages a Python C extension for low level cryptography (Cryptography), Paramiko itself is a pure Python interface around SSH networking concepts.

**Paramiko.sftp\_client.**

SFTPClient provides SFTP (Secure File Transfer Protocol) features to move files between computers. After starting an sftp session via

**ssh = client.open\_sftp()**

you can access all of these methods.

**SFTP**: <http://docs.paramiko.org/en/2.3/api/sftp.html#>

If you just want to move files on the remote host, then you can use "local" file transfer methods with shutil.

**shutil**: <https://docs.python.org/3/library/shutil.html>

* **Paho-Mqtt library**

MQTT is a kind of lightweight IoT messaging protocol based on the publish/subscribe model, which can provide real-time and reliable messaging service for IoT devices, only using very little code and bandwidth. It is suitable for devices with limited hardware resources and the network environment with limited bandwidth. Therefore, MQTT protocol is widely used in IoT, mobile internet, IoV, electricity power, and other industries.

This article mainly introduces how to use the paho-mqtt client and implement connection, subscribe, messaging, and other functions between the client and MQTT broker, in the Python project.

<https://www.emqx.io/blog/how-to-use-mqtt-in-python>

* **Mqtt Broker – Eclipse Mosquitto**

Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.

The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for Internet of Things messaging such as with low power sensors or mobile devices such as phones, embedded computers or microcontrollers.

The Mosquitto project also provides a C library for implementing MQTT clients, and the very popular mosquitto\_pub and mosquitto\_sub command line MQTT clients.

<https://hub.docker.com/_/eclipse-mosquitto/>

* **Adafruit Libraries**

DHT11 Sensor works with the principle of one-wire system. The value of temperature and humidity is sensed by the sensor and then transmitted through the output pin as serial data. We can then read these data by using I/O pin on a MCU/MPU. To understand how these values are read you would have to read through the datasheet of the DHT11 sensor, but for now to keep things simple we will use a library to talk with the DHT11 sensor.

The DHT11 library provided by Adafruit can be used for DHT11, DHT22 and other one wire temperature sensors as well. The procedure to install the DHT11 library is also similar to the one followed for installing LCD library. The only line that would change is the link of the GitHub page on which the DHT library is saved.

<https://circuitdigest.com/microcontroller-projects/raspberry-pi-dht11-interfacing-with-python-code>

**3) Minimal Quick Start Configuration**

* **Machine 1 Setup**

1. Operating System – windows 10
2. Scripting -Preinstalled Python 3.9 or above ,   
   Libraries-Paramiko framework for SFTP ,Paho Framework for MQTT.
3. Networking Configuration- Local IP Configured, Port 22 for SFTP and FTP,Port 1883 for MQTT.

* **Machine 2 Setup**

1. Operating System- Linux (ubuntu 10).
2. Scripting - Preinstalled Python 3.9 or above, Libraries-Paramiko framework for SFTP ,Paho Framework for MQTT, MQTT Broker -Mosquitto.
3. Networking Configuration- Private IP Server Setup, Port 22 for SFTPPort 1883 for MQTT

* **Machine 3 Setup**

1. Operating System- Raspberry Pi (Model B+) with Raspbian Operating System.
2. Scripting- Inbuilt thonny python ide.
3. Networking Configuration- Local IP configured.

* **End Nodes** – DHT 11 Sensor Interfacing with Raspberry Pi.

**4) Set-Up Configurations**

* Machine 1 is a local network configured with Windows Operating System, where the Firmware File is stored and transferred.
* The files are been uploaded from Machine 1 (Local Server) to Machine 2 (Repository Server) through SFTP, and the protocol used here is FTP. These files are transferred through .put .
* The modules that are developed will provide the following functionality:
* Hashing of files, Hashing provides with the Encryption of file and helps to uniquely identify a file and Replacing Old files with the new one.
* Compression of files in a particular format.
* All the tasks i.e., The Hashing, Replacing and Compression of the files will take place at Machine 1 itself even before the files are transferred to Machine 2.

Python Script for Firmware transfer from Machine 1 to Machine 2.

* Machine 2 is a Private Network Configured Linux Operating System (Ubuntu) with the Python Script for Mqtt Client Server Configuration with Machine 2 and 3 and Firmware File Transfer from Machine 2 to 3.
* After file transfer, compressed files will be present on Machine 2.
* Firstly Machine 2 will publish Availability message to Machine 3 using MQTT protocol.
* After publishing the message, MQTT Port will be closed and immediately after that, at the same time SFTP Port will be opened for file transfer i.e., First message will be published through the port 1883 and then the port will be closed, and after that Port 22 will be opened for file transfer.

Python Script for Firmware transfer from Machine 2 to 3.

* Machine 3 is Raspberry Pi (Model B+) with Raspbian Operating System and inbuilt thio python Scripting.
* When the file is transferred to Machine 3 (Raspberry Pi), it will update the firmware at the Sensor Nodes with the New Firmware.
* To access the Mqtt functionalities through python program, we need to install the Mqtt Python libraries on Raspberry Pi.
* To read the received data in python program, we need python mqtt library.
* The most popular mqtt library for python is Paho-mqtt library.

**5) Maintenance and Error Messages and Troubleshooting**

* **Python script debugging**

Python script needs to be tested and debug for errors and compilation purpose it can be done through Python IDE's like jupyter notebook,pycharm etc.Link for downloading Pycharm ide is given below.

<https://www.jetbrains.com/pycharm/download/#section=windows>