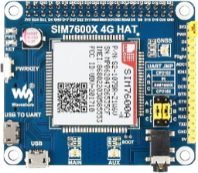
A close-up of a circuit board

Description automatically generatedA green circuit board with many ports

Description automatically generatedA close-up of a computer chip

Description automatically generated**Lab Monitor**

**About**

The Lab Monitor is a custom-built device designed to monitor lab temperature and power outages. It sends SMS alerts to a list of phone numbers in case of high temperature, power changes, and provides daily reports.

**Features**

* **User friendly web interface**: Accessible at device’s IP to check sensor status, device configuration or change settings.
* **SMS Configuration** **via SMS**: Type "help" to see available commands, replies to "status" and "time" commands sent via SMS for non-admin user.
* **History Logs**: Tracks all events and activities recorded by the Lab Monitor.

**Lab Monitor Installation Dates and Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Device IP** | **Location** | **Installation Date** | **Username** | **Password** |
| 172.17.175.252 | ORT Lab | 15th October 2024 | pi | pi |
| 172.16.97.201 | PE Lab | 17th December 2024 | pi | pi |

**Replicating device**

1. **Hardware**

* You will need:
  + Raspberry Pi
  + SD Card
  + WaveShare SIM7600G-H Hat (GSM modem)
  + GSM Antenna
  + PiJuice (UPS)
  + Breakout Board
  + Waterproof DS18B20 Temperature sensor
  + Sim Card (Unlimited SMS plan)
  + Extendable height Raspberry pi case.

**Steps to Assemble:**

* Install Raspbian Operating system on SD card using “Raspberry PI Installer” application.
* Insert the SD card into the Raspberry pi and sim card into the SIM7600x hat
* Place SIM7600x on top of Raspberry Pi and place the PiJuice on top of the SIM7600x hat.
* Give power to PI to turn the device on or you can power PiJuice and click the ‘SW1’ button to turn the device on.

1. **Serial**

* Enable serial interface of the PI using `*sudo raspi-config*` then choose

Interfacing Options>Serial >No >Yes.

1. **IP Address**

* Click on the connection icon on the top right corner of the desktop.
* Navigate to Advanced Options > Edit Connections > Ethernet Wired Connection 1 > IPv4 Settings.
* Edit the IP Address as needed.
* After making the changes, reboot the Raspberry PI and connect it to the Lab Switch.

1. **Sync Time**

* Ensure 'timesyncd' installed on your pi by using the '*systemctl status systemd-timesyncd*' command. If not install using '*sudo apt-get install systemd-timesyncd*'.
* 'sudo nano /etc/systemd/timesyncd.conf' and mention your custom NTP server as 'NTP=172.17.223.167' (Change the IP to any NTP server that is pingable)
* Restart the service 'sudo systemctl restart systemd-timesyncd'.
* Check if Status shows ‘Contacted’ to time server instead of ‘Contacting’.

**A computer screen shot of a program

Description automatically generated**

1. **Connecting the Temperature Sensor**

Note: Use 2.7k pullup resistors for wire with length more than 50 feet for shorter use 4.7k pullup.

* Place the Breakout board on top of PiJuice Connect the sensors onto the board.
* The sensor has 3 wires, red for voltage, black for ground and yellow for sending data to raspberry pi that uses one wire interface.
* To enable One Wire Interface, at the command prompt, `use sudo nano */boot/firmware/config.txt*`, then add to the file: ‘dtoverlay=w1-gpio, gpiopin=X’ (pin of your choice), add all the pins that you have connected the sensors to. Exit Nano

A screenshot of a computer program

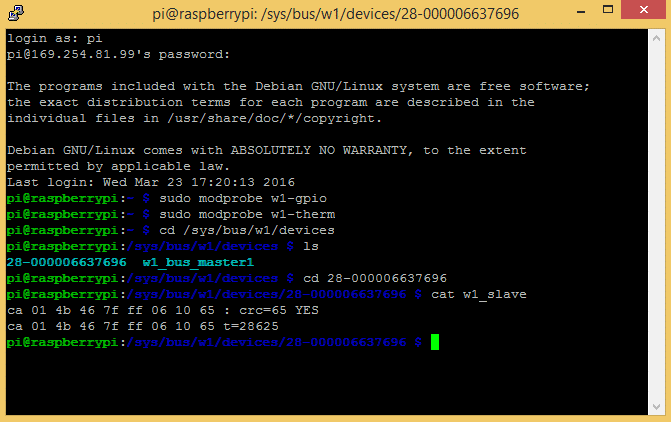
Description automatically generated

* Similarly, add `*w1-gpio*` and `*w1-therm*` to `*/etc/modules*` file to load kernel drivers that are used to read temperature on boot.

A screen shot of a computer

Description automatically generated

* Reboot the Pi with `*sudo reboot*`.
* Enter `cd /sys/bus/w1/devices` and enter ls to list the devices:
* Enter `*cd 28-000006637696*` (Change numbers to the folder name in your pi)
* Enter `*cat w1\_slave*` which will show the raw temperature reading output by the sensor:

[](https://www.circuitbasics.com/wp-content/uploads/2016/03/Raspberry-Pi-DS18B20-Temperature-Sensor-Tutorial-DS18B20-Raw-Output.png)

* If you are adding Sensors after the Lab Monitor has been deployed as service, you will need to restart the service (sudo systemctl restart monitor.service) to detect new sensors.

1. **Software**

* Download the source code from Stash.

Install the dependencies in requirnments.txt and install Pi Juice package using `*sudo apt install pijuice-base`* also install systemd for logging purposes using *`sudo apt-get install python3-systemd`.*

* You might have to change the UART device in the code (SIM7600x.py) according to PI you are using, the user UART device of Raspberry Pi 2B/Zero is ttyAMA0, and ttyS0 of Raspberry Pi 3B.
* `cd` into the project folder and run the project using `python main.py`.
* This should start the Lab Monitor and spin up the web server. The web page should be made available on the device local host’s port 5000. (127.0.0.1:5000)

1. **Changing ports.**

* To access the webpage from the wider network you must set up a Reverse Proxy using nginx.
* To install it, ‘sudo apt-get install nginx’ then enter ‘nano /etc/nginx/sites-available/default’.
* Remove everything and paste:

server {  
 listen 80;  
 server\_name \_;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

* Restart the nginx service using ‘sudo systemctl restart nginx’

1. **Deploying Lab Monitor as a service**

* To have the Lab Monitor Server run perpetually, you must deploy it as a service.
* To achieve that, run **`***sudo nano /etc/systemd/system/monitor.service***`** and paste the following configuration into that file.

[Unit]

Description=Server Service

After=multi-user.target

[Service]

Type=simple

WorkingDirectory=/home/pi/GSM-Alarm

ExecStart=/usr/bin/python3 /home/pi/GSM-Alarm/main.py

Restart=always

RestartSec=3

StandardOutput=journal

StandardError=journal

SyslogIdentifier=LabMonitor

User=pi

[Install]

WantedBy=multi-user.target

* Run the commands `*sudo systemctl daemon-reload*`, `*sudo systemctl enable monitor.service*` and `*sudo systemctl start monitor.service*` one by one to start the service.

**Debugging**

* To check live logs write `*journalctl -u monitor.service -f* `to see live logs of the Lab Monitor. To check older logs,` *journalctl -u monitor.service --since "2024-12-10 9:30:00" --until "2024-12-10 10:56:00"`* (This will show logs from 9:30 AM to 10:56PM).
* If any unexpected bug disrupts the service, `*sudo systemctl restart monitor.service*`.
* If the sensor is showing a reading of 85°C there is bad 1 wire communication or the sensor is not getting enough voltage.
* If a sensor does not show up in ‘28-00000xxxxxxx’ format, the sensor is sending bad signal. Try reducing the pullup resistor and try again.
* If the Monitor keeps sending power lost message, connect the power directly to the PiJuice instead of the raspberry pi.

**How to Use**

1. **Power on the device**

* Plug the device into power to turn it on.
* Connect the PI to a network switch and power it on.

1. **Access the Web Page**

* Wait about 15 seconds for the web server to spin up and the monitor to discover all sensors.
* Open a web browser and navigate to the device's IP address.
* You can see the device configuration and can navigate to the help page, settings page and the history page from on the website.