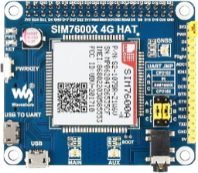
A green circuit board with many ports

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. The device is built using a Raspberry Pi, a SIM7600x GSM modem, a temperature sensor, and PiJuice.

**Features**

* User friendly web page at device’s IP to check configuration or change any settings.
* Configuration via SMS (Admin-only feature). Type "help" to see available commands.
* Replies to "status" and "time" commands sent via SMS for regular users.
* History logs to keep track of all the events that occurred in the Monitor.

**Lab Monitor Installation Dates and Information**

1. **172.17.175.252**

* Installed in ORT lab on 15th October 2024.
* Username: pi
* Password: pi

**Replicating device**

1. **Hardware**

* You will need:
  + Raspberry Pi
  + WaveShare SIM7600G-H hat
  + Antenna
  + PiJuice
  + Waterproof DS18B20 Temperature sensor
  + A Sim Card with unlimited SMS plan.
* Insert the sim card into the SIM7600x hat
* Place it 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.
* Enable serial interface of the PI using `*sudo raspi-config*`.

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**

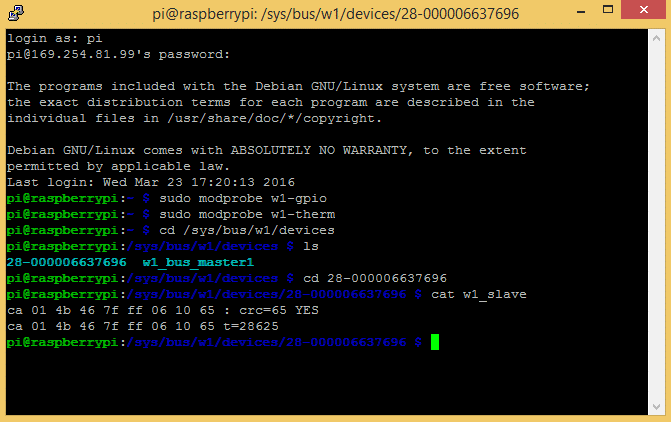
* Make sure you have 'timesyncd' installed on your pi by using the '*systemctl status systemd-timesyncd*' command. If not install using '*sudo apt-get install systemd-timesyncd*'.
* 'sudo nano /etc/systemd/timesyncd.conf' and mention your custom NTP server as 'NTP=172.17.223.167' (Change the IP to any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

**A computer screen shot of a program

Description automatically generated**

1. **Connecting the Temperature Sensor**

* 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 send data to PI.
* To enable One Wire Interface, at the command prompt, enter sudo nano /boot/config.txt, then add this to the bottom of the file: dtoverlay=w1-gpio,gpiopin=XX (pin of your choice), add all the pins that you have connected the sensors to. Exit Nano
* Similarly, add `*w1-gpio*` and `*w1-therm*` to `*/etc/modules*` file to load kernel drivers that are used to read temperature on boot.
* Reboot the Pi with `*sudo reboot*`.
* Change directories to the `*/sys/bus/w1/devices*` folder by entering `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)

1. **Software**

* Download the source code from GitHub using this [link](https://github.com/omkarsindha/GSM-Alarm). (or ask Pedro for the source code)
* 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`.*
* `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 ‘cd /etc/nginx/sites-available/default’.
* Remove everything and paste:

server {  
 listen 80;  
 server\_name \_;

location / {

proxy\_pass http://127.0.0.1:8080;

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 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.
* If any unexpected bug disrupts the service, `*sudo systemctl restart monitor.service*`

**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 10 seconds for the web server to spin up.
* 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.