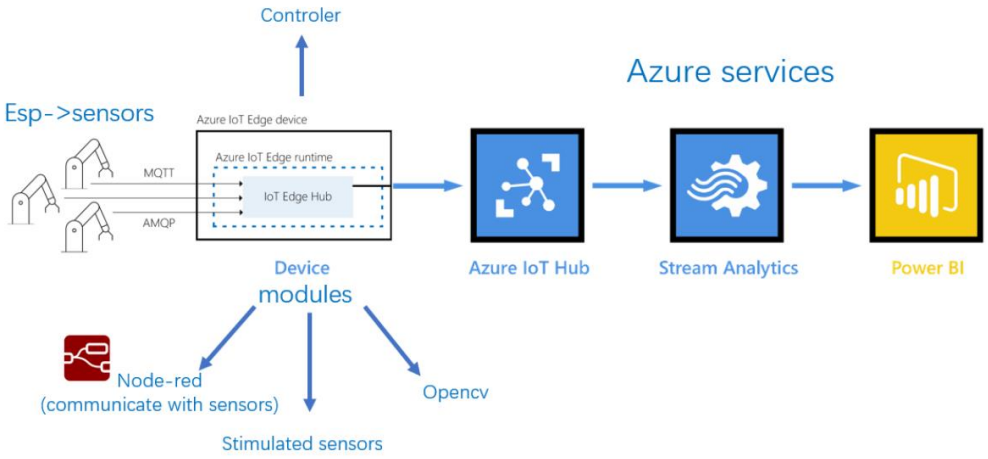
Pervasive Computing Implementation——smart-pi-on-edge

Introduction

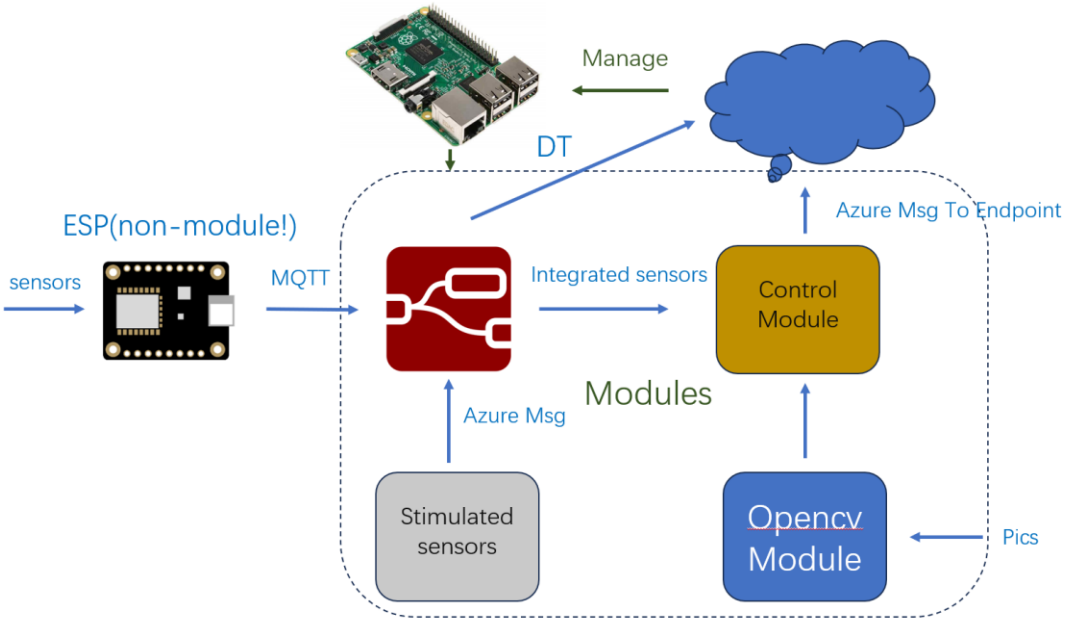
This project aims to explore the implementation of Pervasive Computing in the context of Internet of Things (IoT). Pervasive Computing refers to the integration of smart devices and technologies into our everyday surroundings, enabling seamless connectivity and intelligent interactions. By combining the power of ESP8266 with sensors, Raspberry Pi as edge device with multiple modules, and Azure Cloud services, this project creates a scalable IoT system.

Project Overview

The project revolves around building a IoT system that utilizes various components to enable pervasive connectivity and intelligent control. The key components of the project include ESP826, Raspberry Pi with 4 modules and Azure Cloud services.



Project structure



Modules on edge

Project Structure

The key components used in the project include:

Raspberry Pi:

The Raspberry Pi serves as the edge device, hosting the Node-RED environment and running the edge modules. It acts as a central hub for data processing and coordination at the edge of the cloud.

Message routing in the endpoint of device

ESP8266:

It has 2 sensors: light detector and human detector. It is connected to local Wi-Fi network and facilitating MQTT communication with the Node-RED gateway. The ESP8266 module represents a lower-level device that **cannot directly connect to the cloud**.

Stimulated sensors(**module**):

Installed from edge marketplace. This module represents the capability to **directly connect** sensor devices to the cloud (such as Azure SDK).

Node-RED(**module**):

It acts as the edge module and gateway, responsible for receiving data from the ESP8266 modules over MQTT, and messages from Stimulated sensors, routed in the endpoint of the edge device. It has the ability to aggregate input from multiple different sources and output digital twins, and send messages. In addition, the node-red dashboard is also deployed. Node-RED plays a crucial role as a gateway. It serves as a bridge in the IoT system.

OpenCV(**module**):

The OpenCV library is used to read and analyze images captured by the connected devices, allowing the system to determine if there are any changes in the room environment.

Control module(**module**):

Performing various tasks, including image processing using OpenCV and controlling the virtual LED lights based on sensor information. It also integrates messages into the final output from device, to the endpoint of Azure iot hub.

Azure Cloud services:

The project integrates with Azure IoT Hub, Azure Stream Analytics, and Power BI to

enable data visualization and analysis,.

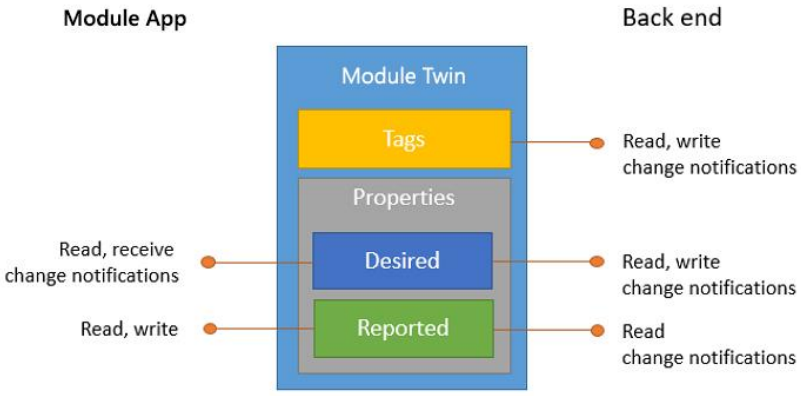
Interesting aspects:

- Edge computing with direct integration of sensors, enabling real-time data processing and decision-making at the edge.

- Development and integration of multiple modules, leveraging Azure IoT Edge Modules for enhanced portability and scalability.

-docker
Docker simplifies the packaging and deployment process, ensuring consistency and portability across different edge devices. By containerizing the modules, they can be easily managed and deployed from the cloud.

-Module **Twins** for Remote Monitoring and Management.
through module twins, we can easily monitor the state of modules, update configurations, and perform actions on specific modules from the cloud.



Module Identity Twin

NodeREDonIoTEdge

Save

The module twin for 'NodeREDonIoTEdge' is shown below. You can add tags and desired properties to your module t


```
                "timeStamp": {
                    "$lastUpdated": "2023-06-17T22:05:51.7757525Z"
                }
            },
            "$version": 2586,
            "ambient": {
                "humidity": 25,
                "temperature": 20.712653538650063
            },
            "detect": 1,
            "humidity": 25,
            "light": 9,
            "machine": {
                "pressure": 10.039168173527258,
                "temperature": 100.34380952318372
            },
            "moduleStatus": "[object Object]",
            "room": 0,
            "temperature": 20.88388654448998,
            "timeCreated": "2023-06-21T17:02:45.4046092Z",
            "timeStamp": "2023-06-17T22:05:50.1307755Z"
        }
    },
    "status": "enabled",
    "statusUpdateTime": "0001-01-01T00:00:00.0000000Z",
    "lastActivityTime": "2023-06-21T16:56:34.1436957Z",
    "connectionState": "Connected",
    "cloudToDeviceMessageCount": 0,
    "authenticationType": "sas"
}
```


Microsoft Azure

Search resources, services, and docs (G

Home > bobofan-iothub | IoT Edge > Pi > IoT Edge Module Details >

Module Identity Twin

EdgeContorlModule

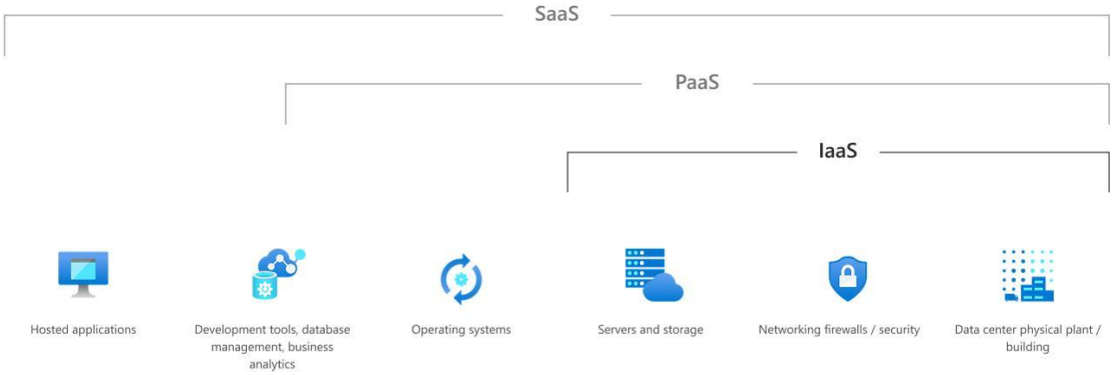
Save

The module twin for 'EdgeContorlModule' is shown below. You can add tags and desired properties to your module tw


```
            },
            "$version": 23,
            "LIGHT_THRESHOLD": 10,
            "PIC_DIFF_THRESHOLD": 30
        },
        "reported": {
            "$metadata": {
                "$lastUpdated": "2023-06-21T17:03:18.1182300Z",
                "LED": {
                    "$lastUpdated": "2023-06-21T17:03:18.1182300Z"
                },
                "lastProcessedMessage": {
                    "$lastUpdated": "2023-06-21T17:03:18.1182300Z"
                },
                "test_ceshi": {
                    "$lastUpdated": "2023-06-21T15:26:11.6605030Z"
                }
            },
            "$version": 503,
            "LED": false,
            "lastProcessedMessage": "06/21/2023 17:03:14",
            "test_ceshi": "123"
        }
    },
    "status": "enabled",
    "statusUpdateTime": "0001-01-01T00:00:00.0000000Z",
    "lastActivityTime": "2023-06-21T16:56:36.1120498Z",
    "connectionState": "Connected",
    "cloudToDeviceMessageCount": 0,
    "authenticationType": "sas"
```

Microservices Architecture for Modular Development
Each Azure IoT Edge module represents a specific microservice, encapsulating a distinct functionality or service within the IoT system. These modules can communicate with each other, forming a cohesive system. This modular approach facilitates independent development, deployment, and scaling of the system, allowing for easier management and flexibility as the project expands.

PaaS in the Project Context
In the project, the use of Azure IoT Edge modules and their deployment with Docker aligns with the Platform as a Service (PaaS) model. This approach provides a managed platform where developers can focus on building and deploying modules without the need to manage the underlying infrastructure. The platform takes care of managing the edge devices and provides services for module management, communication, and monitoring, enabling faster development and deployment of the IoT system.



Challenges:
-Development of modules using Azure IoT SDK and deployment using Docker.

-Effective utilization of the Azure platform, including **Azure Central, Power BI**, Azure Functions, and other services to achieve desired functionality.

-System integration, leveraging modular design principles and containerization techniques to simplify the overall system architecture and improve interoperability.

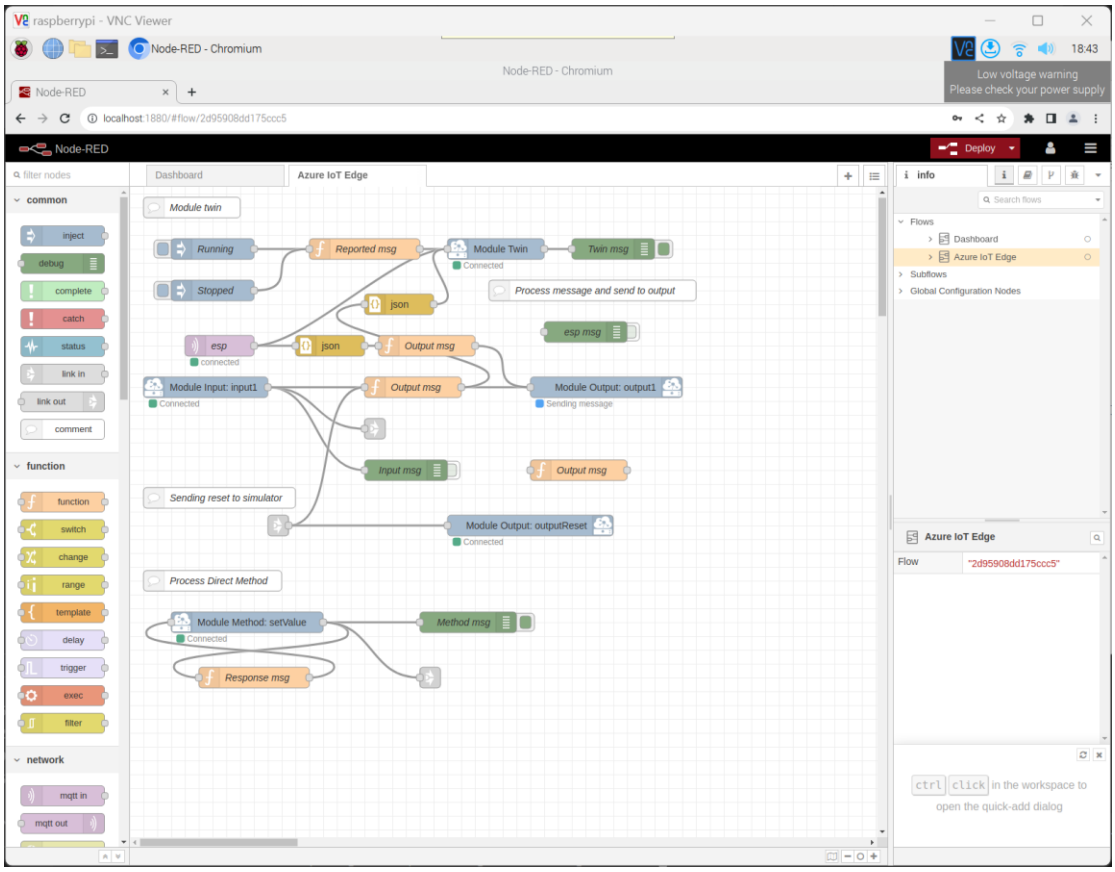
Possible extensions:
- Further development on Digital twins(Twin Graphs & their visualization (Explorer))
- The use of Node-RED as a **gateway** and the scalability for **expanding downstream devices** (Reference: Scaling Node-RED Horizontally for High Availability).
https://www.narendranaidu.com/2016/07/scaling-node-red-horizontally-for-high.html
Different topic/ Shared Subscription/ Other modules and extensions
- Further development on SaaS （iot central）. Dashbroad on the cloud.

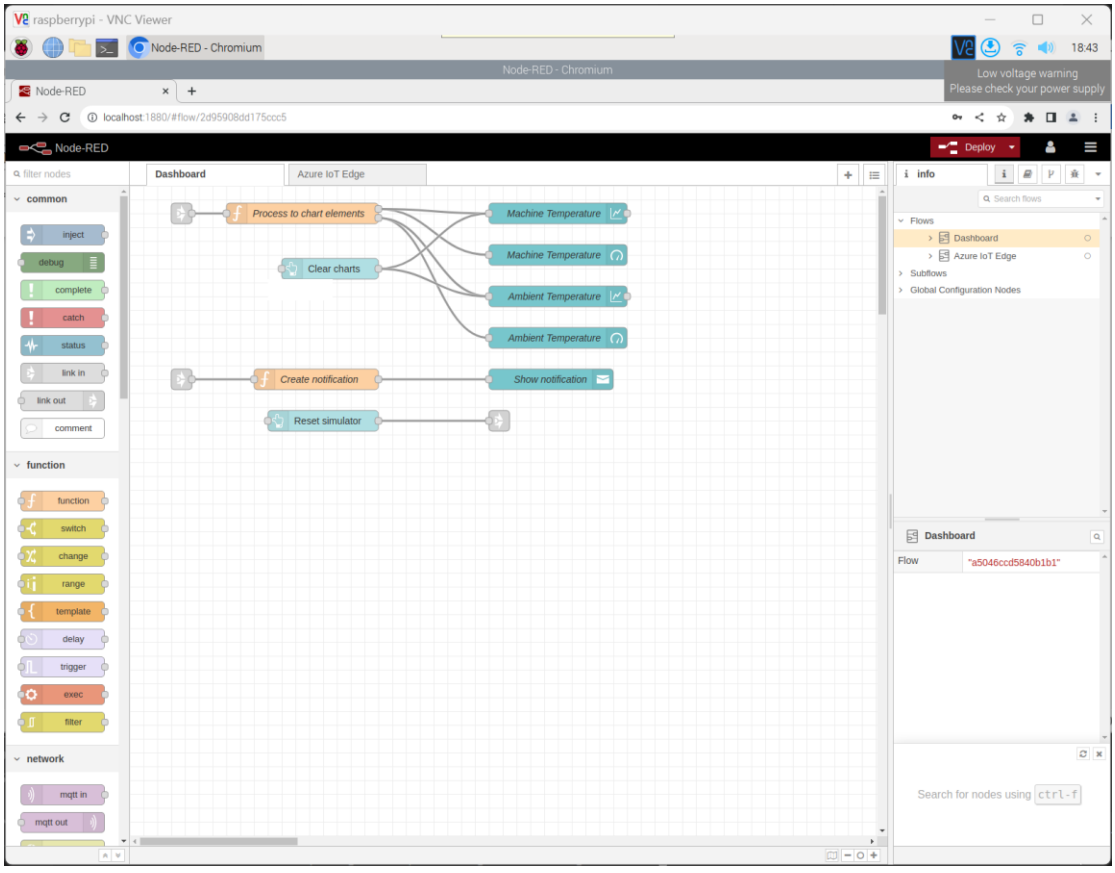
```
edgeHub                     stopped          Stopped 23 minutes ago  mcr.microsoft.com/azureiotedge-hub:1.4
boyang@raspberrypi:~ $ iotedge list
NAME                        STATUS           DESCRIPTION      CONFIG
EdgeContorlModule           running          Up 18 minutes    boboedge.azurecr.io/edgecontorlmodule:0.1.1-arm32v7
NodeREDonIoTEdge            running          Up 17 minutes    marketplace.azurecr.io/1604376875527/wedx-nodered:latest
OpencvModule                running          Up 17 minutes    boboedge.azurecr.io/opencvmodule:0.1.2-arm32v7
SimulatedTemperatureSensor  running          Up 18 minutes    mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:latest
edgeAgent                   running          Up 19 minutes    mcr.microsoft.com/azureiotedge-agent:1.4
edgeHub                     running          Up 18 minutes    mcr.microsoft.com/azureiotedge-hub:1.4
boyang@raspberrypi:~ $
```

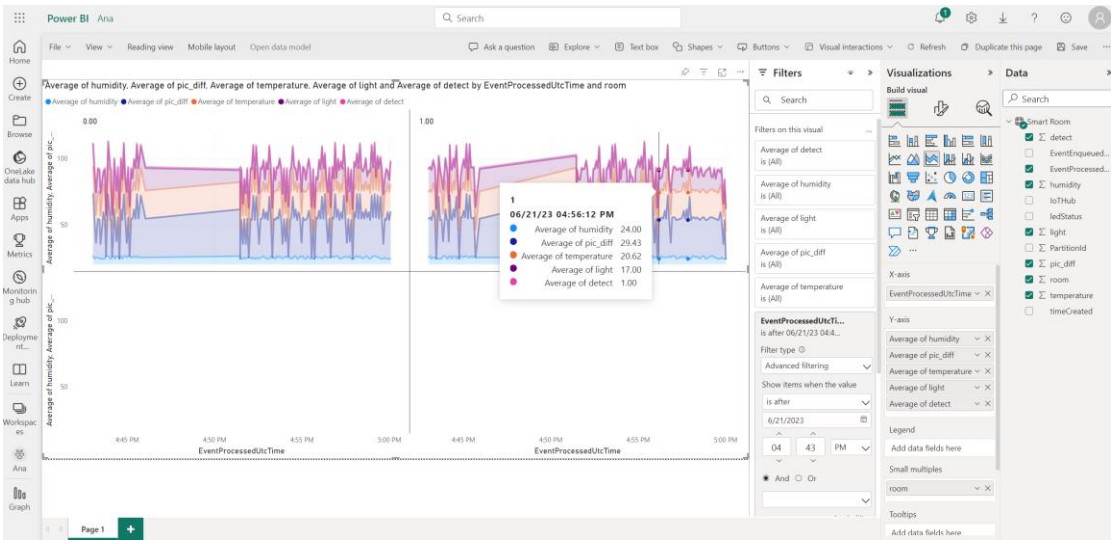
```
    1 reference
18  public class Machine
19  {
        0 references
20      public double? temperature {get; set;}
        0 references
21      public double? pressure {get; set;}
22  }
    1 reference
23  public class Ambient
24  {
```

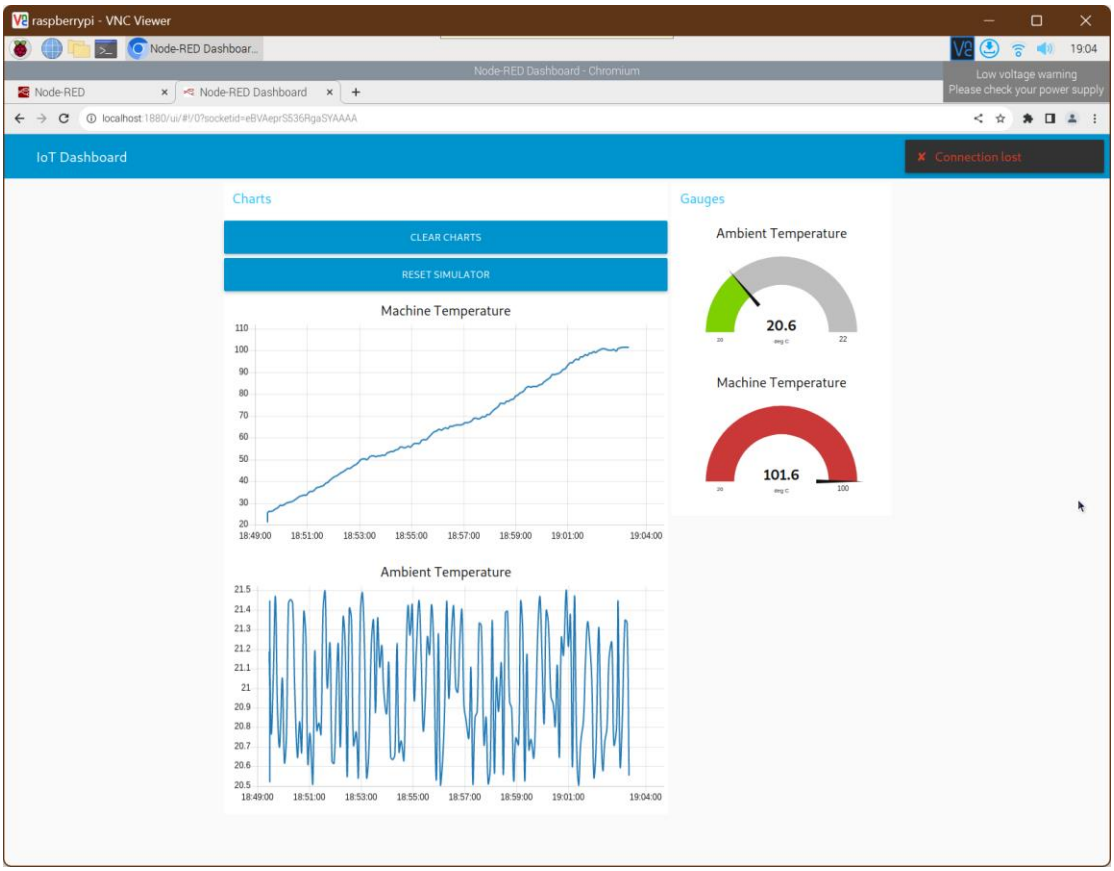
问题 4 输出 调试控制台 终端 JUPYTER 串行监视器 AZURE

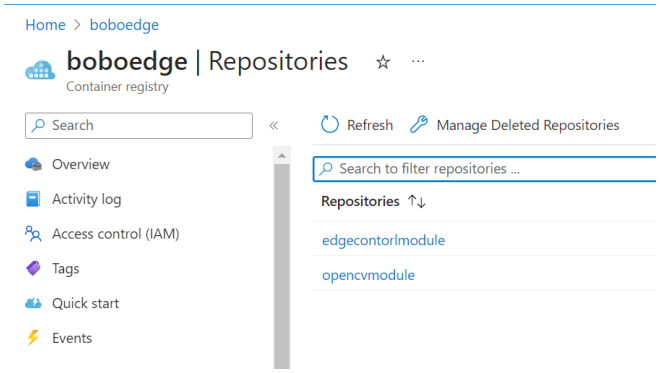
```
  "detect": 1,
  "room": 0,
  "ledStatus": true,
  "temperature": 21.499626802731516,
  "humidity": 25
}
[IoTHubMonitor] [6:34:50 PM] Message received from [Pi/EdgeContorlModule]:
{
  "timeCreated": "2023-06-21T16:34:46.3542405Z",
  "pic_diff": 47.109272,
  "light": 10,
  "detect": 1,
  "room": 1,
  "ledStatus": false,
  "temperature": 21.499626802731516,
  "humidity": 25
}
[IoTHubMonitor] [6:34:52 PM] Message received from [Pi/EdgeContorlModule]:
{
  "timeCreated": "2023-06-21T16:34:51.3747963Z",
  "pic_diff": 35.499813,
  "light": 10,
  "detect": 1,
  "room": 0,
  "ledStatus": false,
  "temperature": 21.44856636095934,
  "humidity": 25
}
```

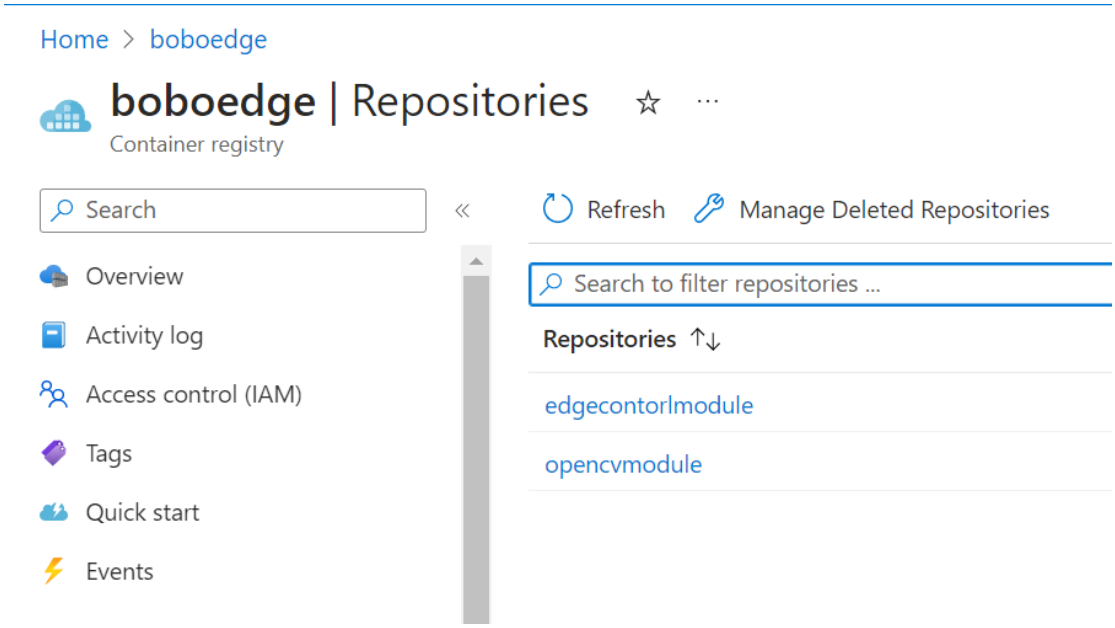

raspberrypi - VNC Viewer
Node-RED - Chromium
18:43
Low voltage warning
Please check your power supply
Node-RED
localhost:1880/#flow/2d95908dd175ccc5
Deploy
Dashboard
Azure IoT Edge
Module twin
Running
Reported msg
Module Twin
Connected
Twin msg
Stopped
json
Process message and send to output
esp
connected
json
Output msg
esp msg
Module Input: input1
Connected
Output msg
Module Output: output1
Sending message
Input msg
Output msg
Sending reset to simulator
Module Output: outputReset
Connected
Process Direct Method
Module Method: setValue
Connected
Method msg
Response msg
info
Flows
Dashboard
Azure IoT Edge
Subflows
Global Configuration Nodes
Azure IoT Edge
Flow
"2d95908dd175ccc5"
ctrl click in the workspace to open the quick-add dialog
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
delay
trigger
exec
filter
network
mqtt in
mqtt out

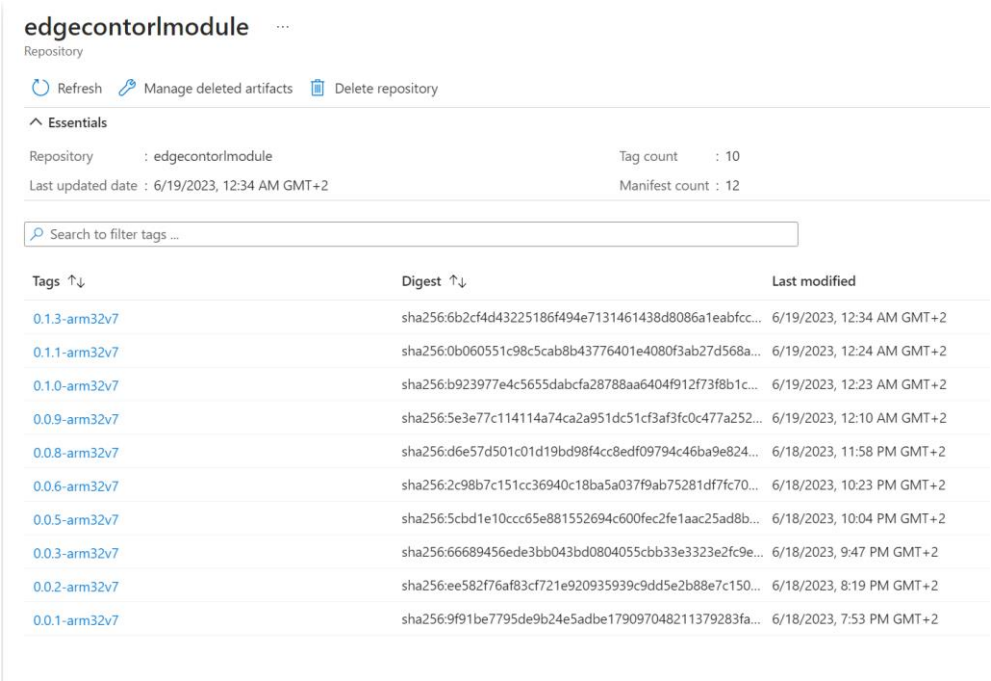

raspberrypi - VNC Viewer
Node-RED - Chromium
18:43
Low voltage warning
Please check your power supply
Node-RED
localhost:1880/#flow/2d95908dd175ccc5
Deploy
Dashboard
Azure IoT Edge
Process to chart elements
Machine Temperature
Machine Temperature
Clear charts
Ambient Temperature
Ambient Temperature
Create notification
Show notification
Reset simulator
info
Flows
Dashboard
Azure IoT Edge
Subflows
Global Configuration Nodes
Dashboard
Flow
"a5046ccd5840b1b1"
Search for nodes using ctrl-f
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
delay
trigger
exec
filter
network
mqtt in
mqtt out


Power BI
Average of humidity, Average of pic_diff, Average of temperature, Average of light and Average of detect by EventProcessedUtcTime and room
Filters
Visualizations
Data
06/21/23 04:56:12 PM
Average of humidity 24.00
Average of pic_diff 29.43
Average of temperature 20.62
Average of light 17.00
Average of detect 1.00
EventProcessedUtcTime
Page 1


raspberrypi - VNC Viewer
Node-RED Dashboard - Chromium
Low voltage warning
Please check your power supply
Node-RED
Node-RED Dashboard
localhost:1880/ui/#!/0?socketid=eBVAeprS536RgaSYAAAA
IoT Dashboard
Connection lost
Charts
CLEAR CHARTS
RESET SIMULATOR
Machine Temperature
Ambient Temperature
Gauges
Ambient Temperature
20.6
Machine Temperature
101.6


Home > boboedge
boboedge | Repositories
Container registry
Search
Refresh
Manage Deleted Repositories
Overview
Activity log
Access control (IAM)
Tags
Quick start
Events
Search to filter repositories ...
Repositories
edgecontorlmodule
opencvmodule


Home > boboedge
boboedge | Repositories
Container registry
Search
Refresh
Manage Deleted Repositories
Overview
Activity log
Access control (IAM)
Tags
Quick start
Events
Search to filter repositories ...
Repositories
edgecontorlmodule
opencvmodule


edgecontorlmodule
Repository
Refresh
Manage deleted artifacts
Delete repository
Essentials
Repository : edgecontorlmodule
Last updated date : 6/19/2023, 12:34 AM GMT+2
Tag count : 10
Manifest count : 12
Search to filter tags ...
Tags
Digest
Last modified
0.1.3-arm32v7 sha256:6b2cf4d43225186f494e7131461438d8086a1eabfcc... 6/19/2023, 12:34 AM GMT+2
0.1.1-arm32v7 sha256:0b060551c98c5cab8b43776401e4080f3ab27d568a... 6/19/2023, 12:24 AM GMT+2
0.1.0-arm32v7 sha256:b923977e4c5655dabcfa28788aa6404f912f73f8b1c... 6/19/2023, 12:23 AM GMT+2
0.0.9-arm32v7 sha256:5e3e77c114114a74ca2a951dc51cf3af3fc0c477a252... 6/19/2023, 12:10 AM GMT+2
0.0.8-arm32v7 sha256:d6e57d501c01d19bd98f4cc8edf09794c46ba9e824... 6/18/2023, 11:58 PM GMT+2
0.0.6-arm32v7 sha256:2c98b7c151cc36940c18ba5a037f9ab75281df7fc70... 6/18/2023, 10:23 PM GMT+2
0.0.5-arm32v7 sha256:5cbd1e10ccc65e881552694c600fec2fe1aac25ad8b... 6/18/2023, 10:04 PM GMT+2
0.0.3-arm32v7 sha256:66689456ede3bb043bd0804055cbb33e3323e2fc9e... 6/18/2023, 9:47 PM GMT+2
0.0.2-arm32v7 sha256:ee582f76af83cf721e920935939c9dd5e2b88e7c150... 6/18/2023, 8:19 PM GMT+2
0.0.1-arm32v7 sha256:9f91be7795de9b24e5adbe179097048211379283fa... 6/18/2023, 7:53 PM GMT+2