

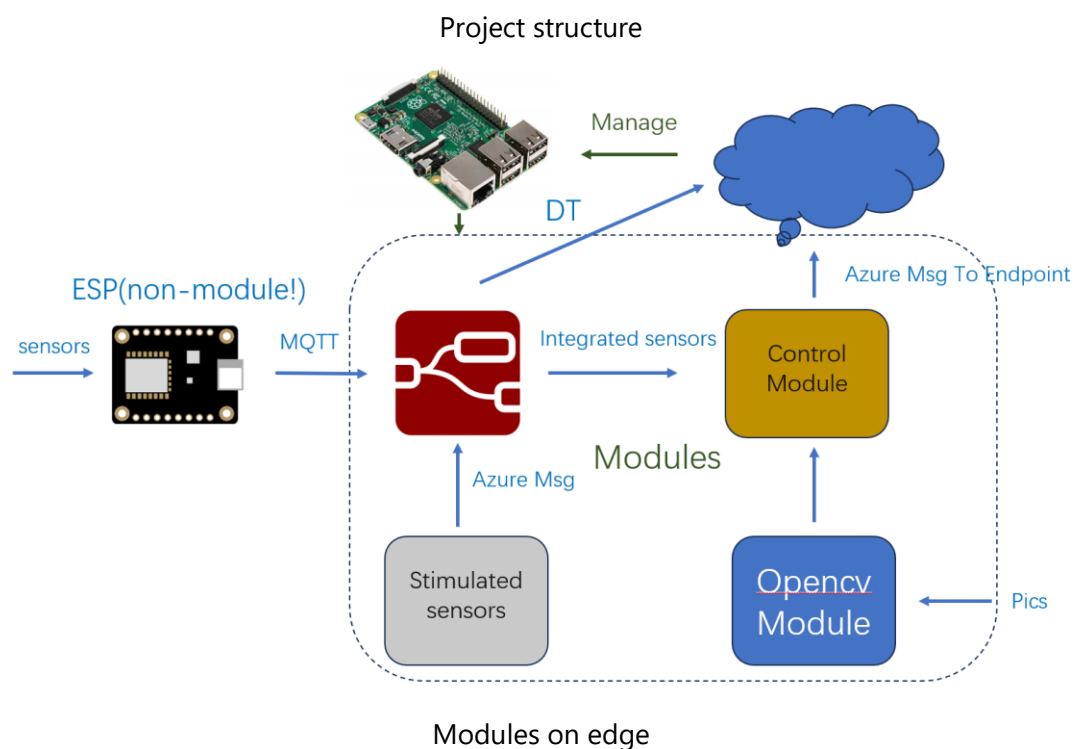
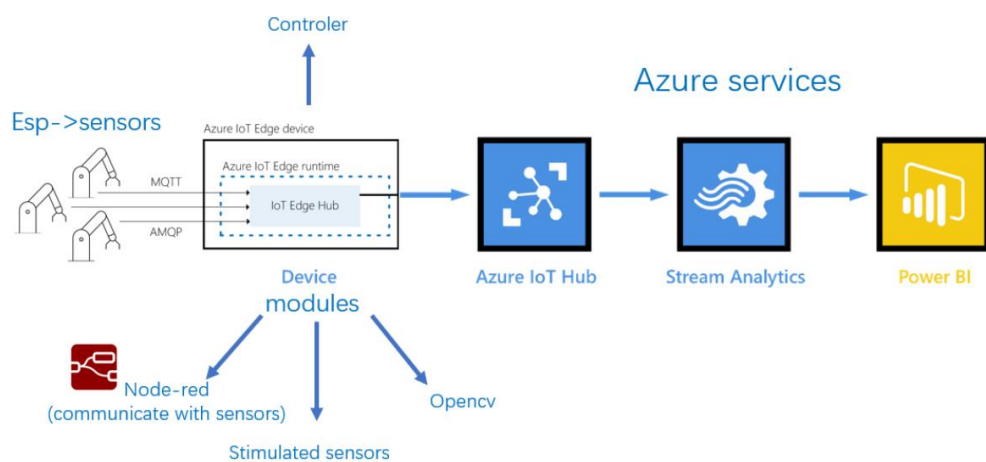
Pervasive Computing Implementation——smart-pi-on-edge

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 ESP8266, Raspberry Pi with 4 modules and Azure Cloud services.



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.

```
1  },
2  },
3  },
4  },
5  },
6  },
7  },
8  },
9  },
10 },
11 },
12 },
13 },
14 },
15 },
16 },
17 },
18 },
19 },
20 },
21 },
22 },
23 },
24 },
25 },
26 },
27 },
28 },
29 },
30 },
31 },
32 },
33 },
34 },
35 },
36 },
37 },
38 },
39 },
40 },
41 },
42 },
43 },
44 },
45 },
46 },
47 },
48 },
49 },
50 },
51 },
52 },
53 },
54 },
55 },
56 },
57 },
58 },
59 },
60 },
61 },
62 },
63 },
64 },
65 },
66 },
67 },
68 },
69 },
70 },
71 },
72 },
73 },
74 },
75 },
76 },
77 },
78 },
79 },
80 },
81 },
82 },
83 },
84 },
85 },
86 },
87 },
88 },
89 },
90 },
91 },
92 },
93 },
94 },
95 },
96 },
97 },
98 },
99 },
100 }
```

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:

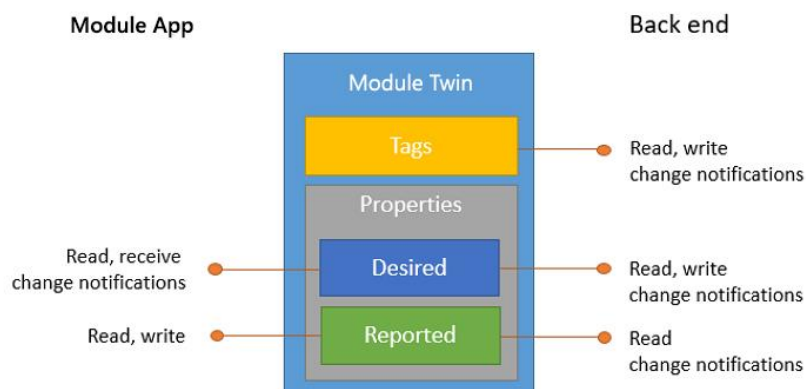
- Edge computing with direct integration of sensors, enabling real-time data processing and decision-making at the edge.
- Development and integration of multiple modules, leveraging Azure IoT Edge Modules for enhanced portability and scalability.

-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 twin

```
58     "timeStamp": {
59       "$lastUpdated": "2023-06-17T22:05:51.7757525Z"
60     },
61   },
62   "$version": 2586,
63   "ambient": {
64     "humidity": 25,
65     "temperature": 20.712653538650063
66   },
67   "detect": 1,
68   "humidity": 25,
69   "light": 9,
70   "machine": {
71     "pressure": 10.039168173527258,
72     "temperature": 100.34380952318372
73   },
74   "moduleStatus": "[object Object]",
75   "room": 0,
76   "temperature": 20.88388654448998,
77   "timeCreated": "2023-06-21T17:02:45.4046092Z",
78   "timeStamp": "2023-06-17T22:05:50.1307755Z"
79 },
80 },
81 "status": "enabled",
82 "statusUpdateTime": "0001-01-01T00:00:00.0000000Z",
83 "lastActivityTime": "2023-06-21T16:56:34.1436957Z",
84 "connectionState": "Connected",
85 "cloudToDeviceMessageCount": 0,
86 "authenticationType": "sas"
87 }
```

Microsoft Azure Search resources, services, and docs (G

Home > bobofan-iiotHub | 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

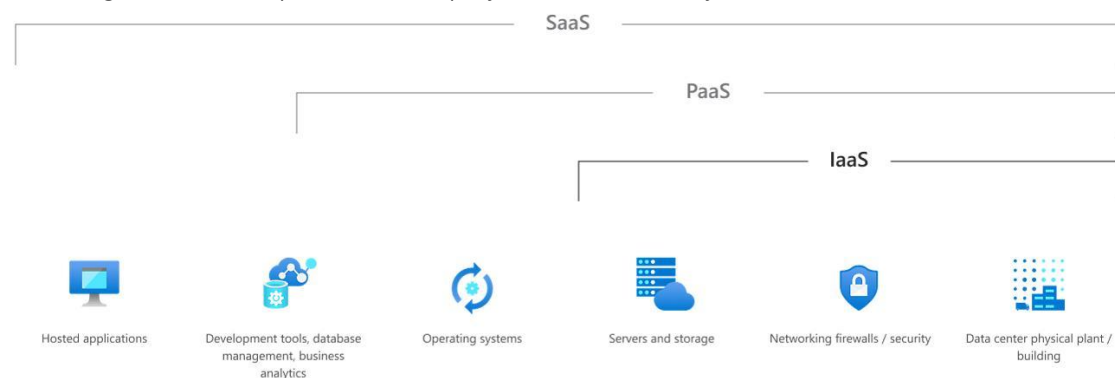
```
21   },
22   "$version": 23,
23   "LIGHT_THRESHOLD": 10,
24   "PIC_DIFF_THRESHOLD": 30
25 },
26 "reported": {
27   "$metadata": {
28     "$lastUpdated": "2023-06-21T17:03:18.1182300Z",
29     "LED": {
30       "$lastUpdated": "2023-06-21T17:03:18.1182300Z"
31     },
32     "lastProcessedMessage": {
33       "$lastUpdated": "2023-06-21T17:03:18.1182300Z"
34     },
35     "test_ceshi": {
36       "$lastUpdated": "2023-06-21T15:26:11.6605030Z"
37     }
38   },
39   "$version": 503,
40   "LED": false,
41   "lastProcessedMessage": "06/21/2023 17:03:14",
42   "test_ceshi": "123"
43 },
44 },
45 "status": "enabled",
46 "statusUpdateTime": "0001-01-01T00:00:00.0000000Z",
47 "lastActivityTime": "2023-06-21T16:56:36.1120498Z",
48 "connectionState": "Connected",
49 "cloudToDeviceMessageCount": 0,
50 "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    bobeedge.azurecr.io/edgecontorlmodule:0.1.1-arm32v7
NodeREDonIoTEdge                   running    Up 17 minutes    marketplace.azurecr.io/1604376875527/wedx-nodered:latest
OpencvModule                        running    Up 17 minutes    bobeedge.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 {
20     0 references
21     public double? temperature {get; set;}
22     0 references
23     public double? pressure {get; set;}
24 }
25 1 reference
26 public class Ambient
27 {

```

问题 4 输出 调试控制台 终端 JUPYTER 串行监视器 AZURE

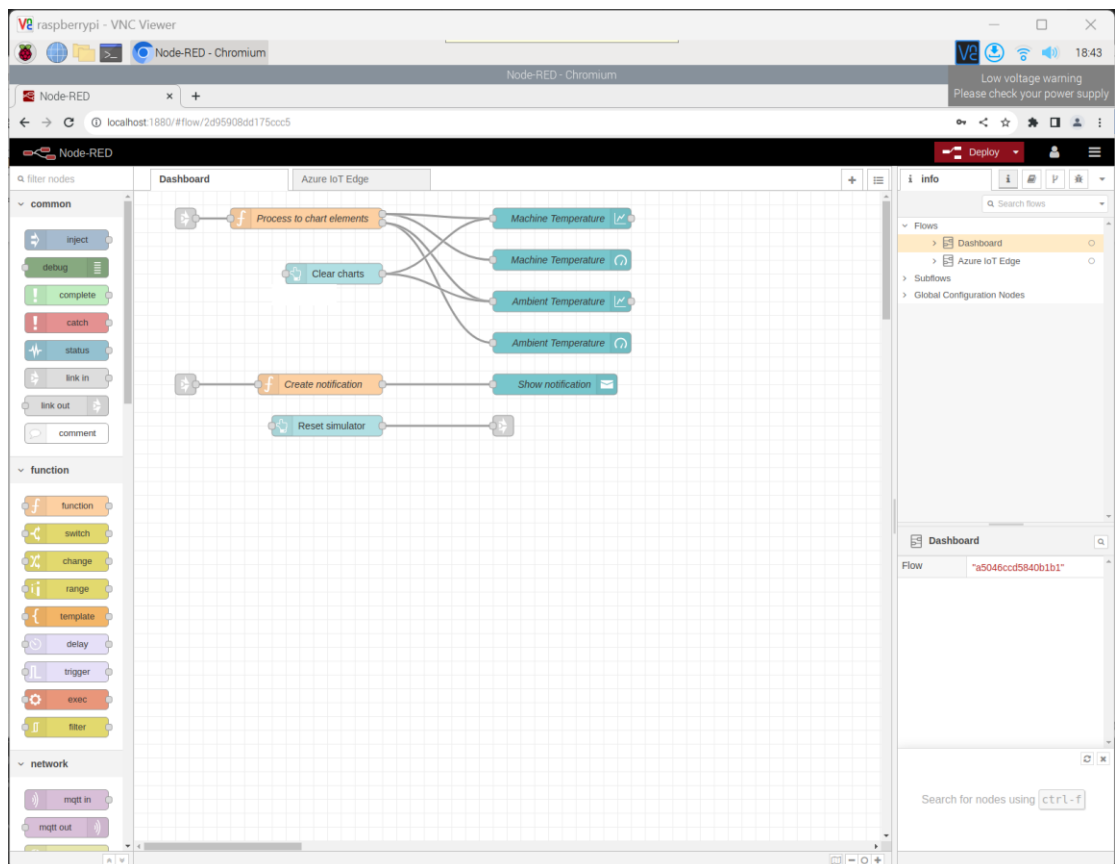
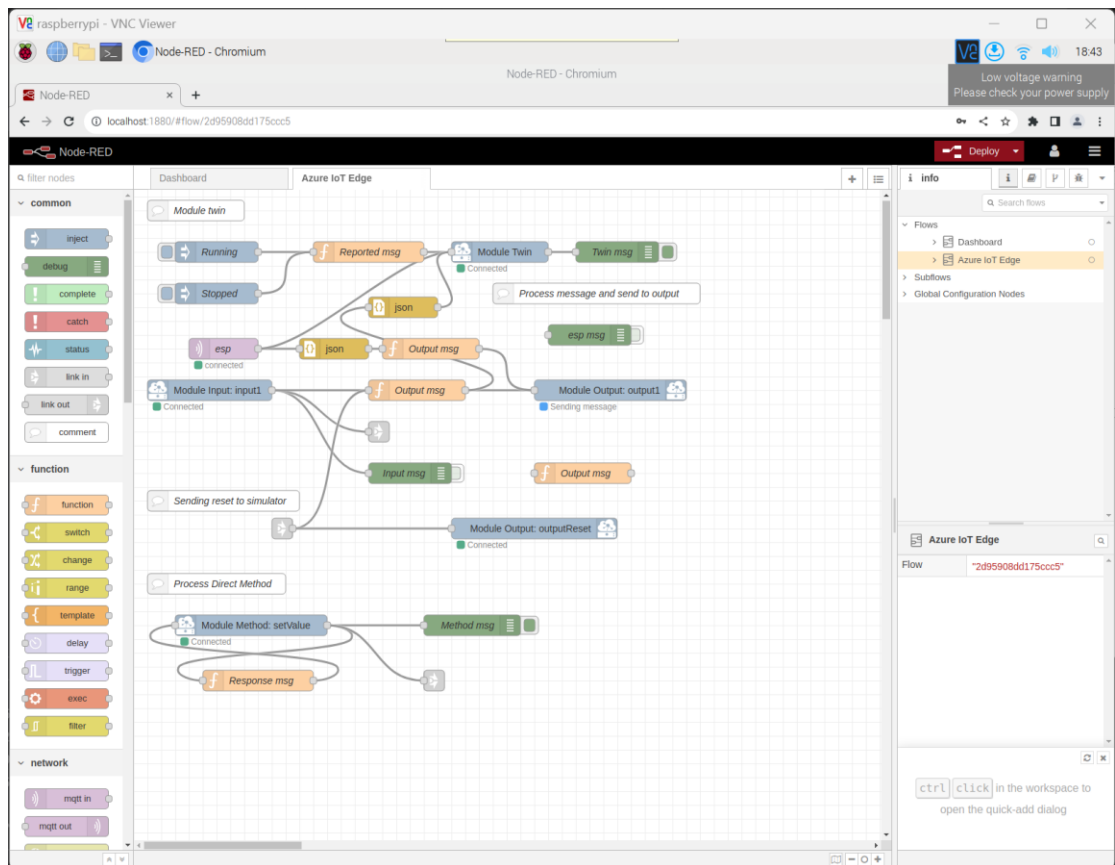
```

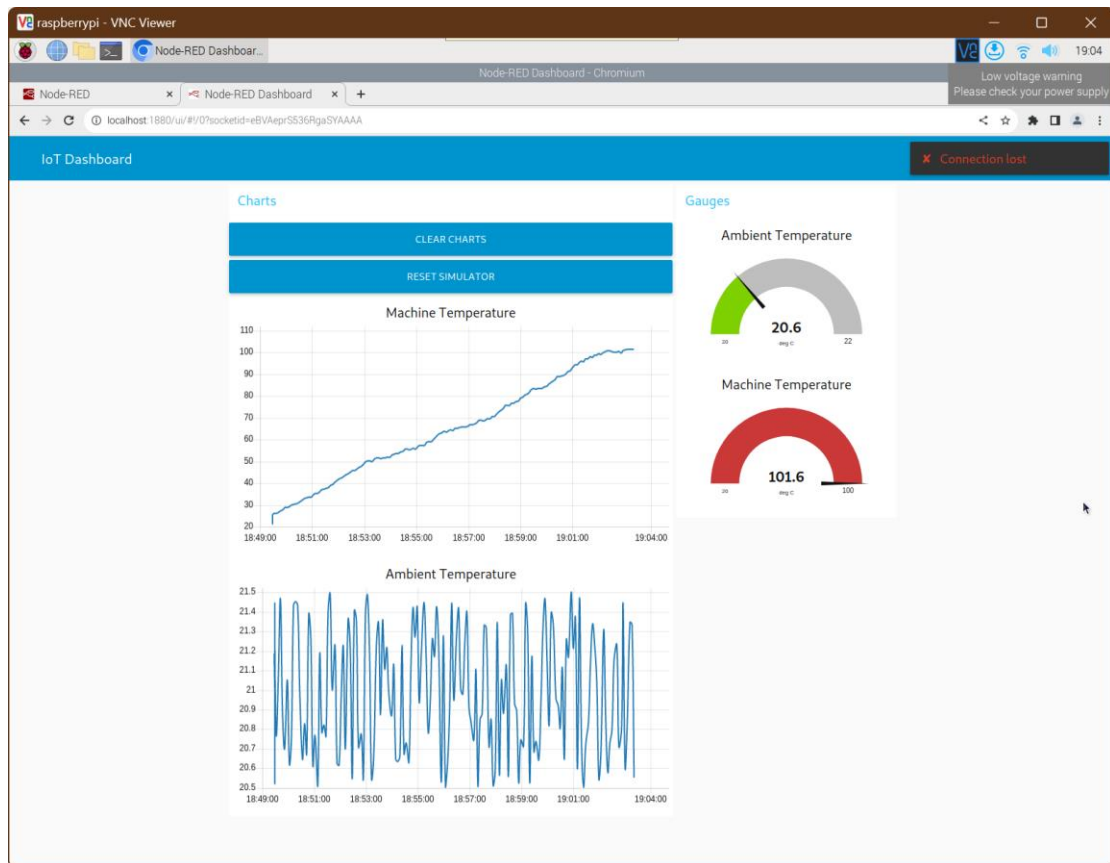
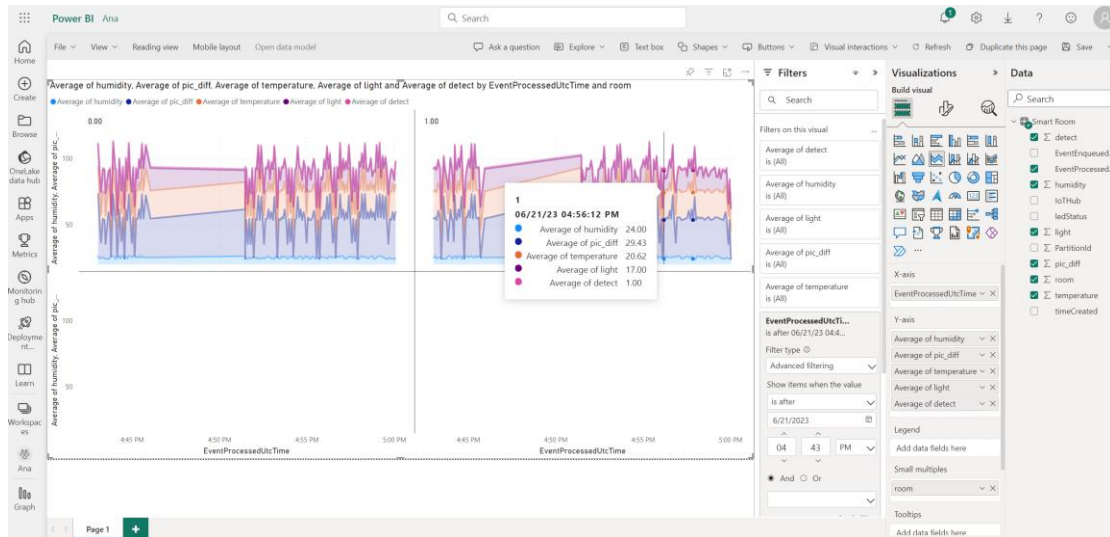
{
  "light": 10,
  "detect": 1,
  "room": 0,
  "ledStatus": true,
  "temperature": 21.499626802731516,
  "humidity": 25
}

[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
}

```





Home > boboedge

boboedge | Repositories

Container registry

Search



Refresh



Manage Deleted Repositories

Overview

Activity log

Access control (IAM)

Tags

Quick start

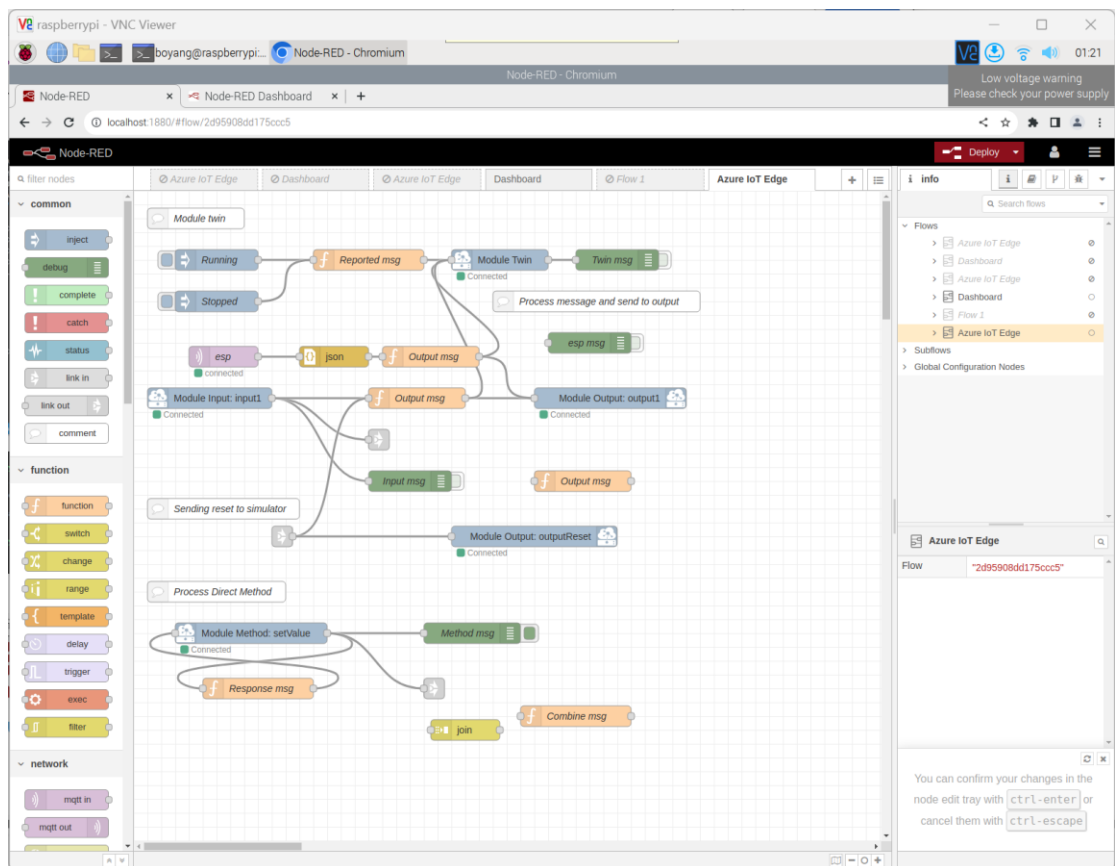
Events

Search to filter repositories ...

Repositories ↑↓

edgecontorlmodule

opencvmodule





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

Tag count : 10

Last updated date : 6/19/2023, 12:34 AM GMT+2

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:b923977e4c5655dabcf28788aa6404f912f73f8b1c... | 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 |