**Tài liệu mô tả cấu trúc & hướng dẫn sử dụng Gateway**

Link github: https://github.com/dangnguyenquang/USM

Author: dangnguyenquang

Collaborators: NMD555

1. **Thông tin server và serial port**

* kafkaServer = '173.249.14.91:29092'; // địa chỉ kafkaServer
* portName = 'COM5'; // địa chỉ cổng serial
* intervalMillisecond = 60000; // interval của từng thiết bị được set cứng
* expectedDataCount = 17; // số lượng data mong muốn nhận được từ serialPort.
* baudRate = 115200 // tốc độ baud

1. **Thông tin hệ thống :**
2. **Cấu trúc file:**

app.js có nhiệm vụ nhận thông tin từ các thiết bị và liên lạc với server qua kafka

1. **Cấu trúc data nhận được từ serialPort**
2. Dãy bit hex từ gateway gửi đến thiết bị (res): **05 05 01 AA**

Chú thích:

05 05: address (03 03, 04 04) // là địa chỉ của máy

01: channel

1. Dãy bit hex nhận gateway nhận được từ thiết bị (req):

**05 05 01 00 00 00 00 00 01 01 00 0F 00 01 00 00 13**

Chú thích:

05 05: address (03 03, 04 04)

01: islamp (có bo lamp hay không)

00 00: red (dừng) // màu đèn

00 00: yellow (nghỉ) // màu đèn

00 01: green (chạy) // màu đèn

01: iscount (có bo đếm hay không)

00 0F: số lượng sản phẩm (hex) (max: 65535) // dãy bit dưới dạng hex

00 01: nút start/stop

00 00: bỏ

13: sum // check xem data có phải data rác hay không

1. **Mô tả gatewway hoạt động:**

Vì không có database nên dữ liệu của các máy sẽ được set cứng và đưa vào 1 mảng chứa những object:

let device = [

  {

    hexArray: [0x03, 0x03, 0x01, 0xAA],

// *chứa địa chỉ của thiết bị*

    preIndexs: 0,

// *sản phẩm đã hoàn thành ở lần cập nhật trước*

    indexs: 0,

// *số sản phẩm đã hoàn thành trong 1 lot*

    lastedBtn: '0',

// *trạng thái của btn trước đó*

    status: 0,

// *trạng thái đèn (0: red, 1: yellow, 2: green)*

    jsonDataAlive: {},

// *dữ liệu được gửi đi thông qua kafka (alive)*

    jsonDataCount: {},

// *dữ liệu được gửi đi thông qua kafka (count)*

    alive: false

// *thiết bị có đang sống hay không*

  },

];

Lưu ý: Các thông số trong mảng sẽ được cập nhật sau 5s/thiết bị (mỗi thiết bị được cập nhật cách nhau 5s, hết 1 vòng sẽ quay lại)

Biến alive tồn tại với mục đích nếu có thiết bị chết sẽ không gửi data rác đến kafka server

1. Bản tin Start (topic: mes\_StartIOTRequest):

Bản tin được gửi đi mỗi khi gateway trạng thái của btn thay đổi từ off sang on (00 00 to 00 01)

1. Bản tin Stop (topic: mes\_StopIOTRequest):

Bản tin được gửi đi mỗi khi gateway trạng thái của btn thay đổi từ off sang on (00 01 to 00 00)

1. Bản tin đếm (topic: mes\_StopIOTRequest):

Bản tin được gửi sau mỗi 60s.

Số lượng sản phẩm trong 1 lot được tính như sau:

*if (combinedValue < currentDevice.preIndexs) {*

*currentDevice.indexs += combinedValue + 65535 - currentDevice.preIndexs + 1;*

*} else {*

*currentDevice.indexs += combinedValue - currentDevice.preIndexs;*

*}*

*currentDevice.preIndexs = combinedValue;* // **Cập nhật lại giá trị của preIndexs**

Chú thích:

combinedValue: Số lượng sản phẩm nhận được thông qua serialPort

currentDevice.indexs: Số lượng sản phẩm trong lot hiện tại

currentDevice.preIndexs: Số lượng sản phẩm trong lần cập nhật trước

Giải thích: Vì số lượng bit trong mảng là có hạn (00 00: chỉ được số lượng tối đa là 65535 tương ứng với FF FF, nếu quá số lượng này số sẽ chuyển về 00 00) nên cần phải có điều kiện if để cộng dồn số lượng sản phẩm của máy nếu dãy số reset.

1. Bản tin alive (topic: mes\_AliveIOTRequest)

Bản tin này cũng được gửi sau mỗi 60s