VIETTEL DIGITAL TALENT

Control SIM7090 via ESP32

Mentor: Nguyễn Minh Thi

Nguyễn Trọng Tín
Trần Thọ Nhân

Agenda

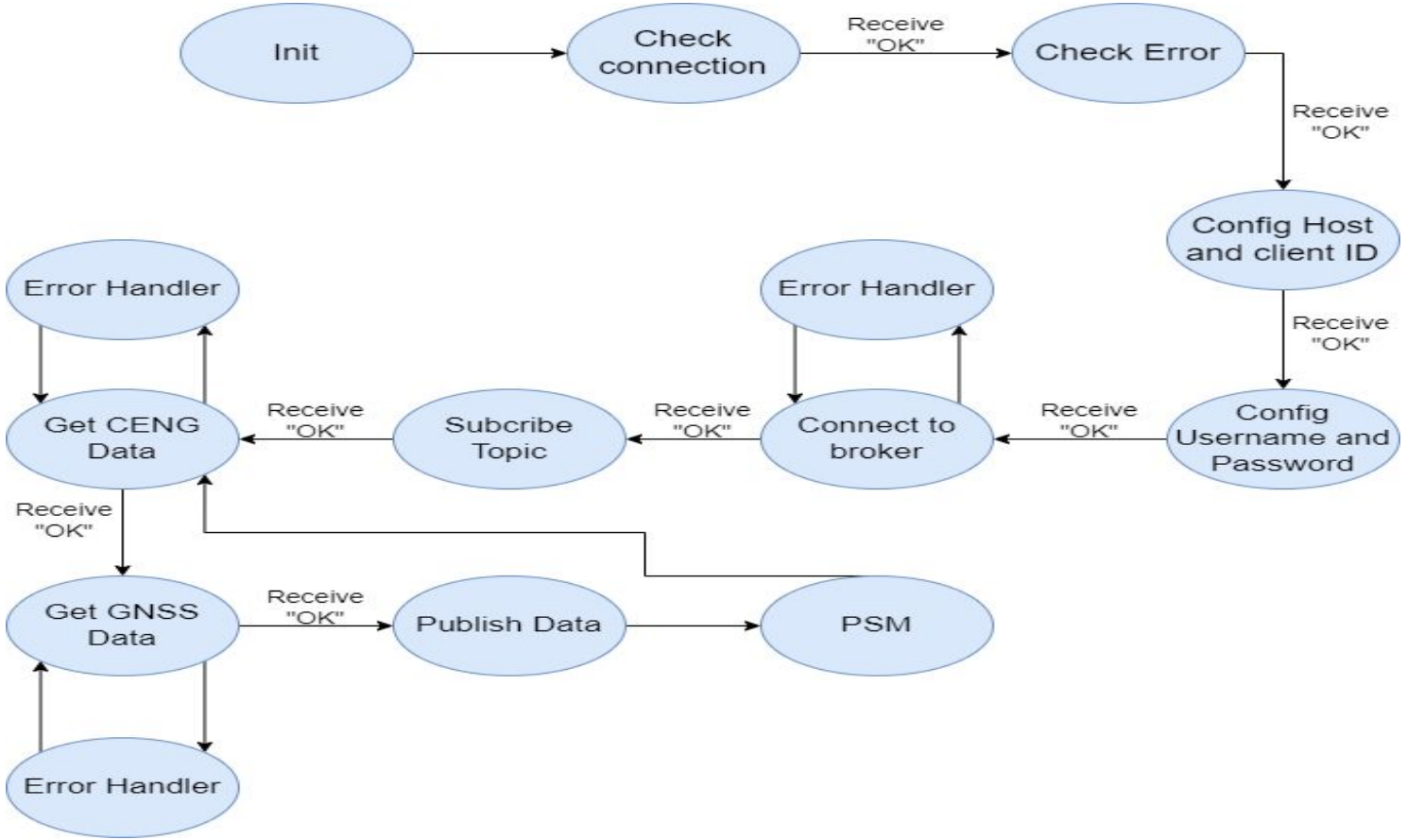
1. Chức năng
2. Finite-state machine và implementation
3. Dashboard
4. Link Dashboard

Chức năng

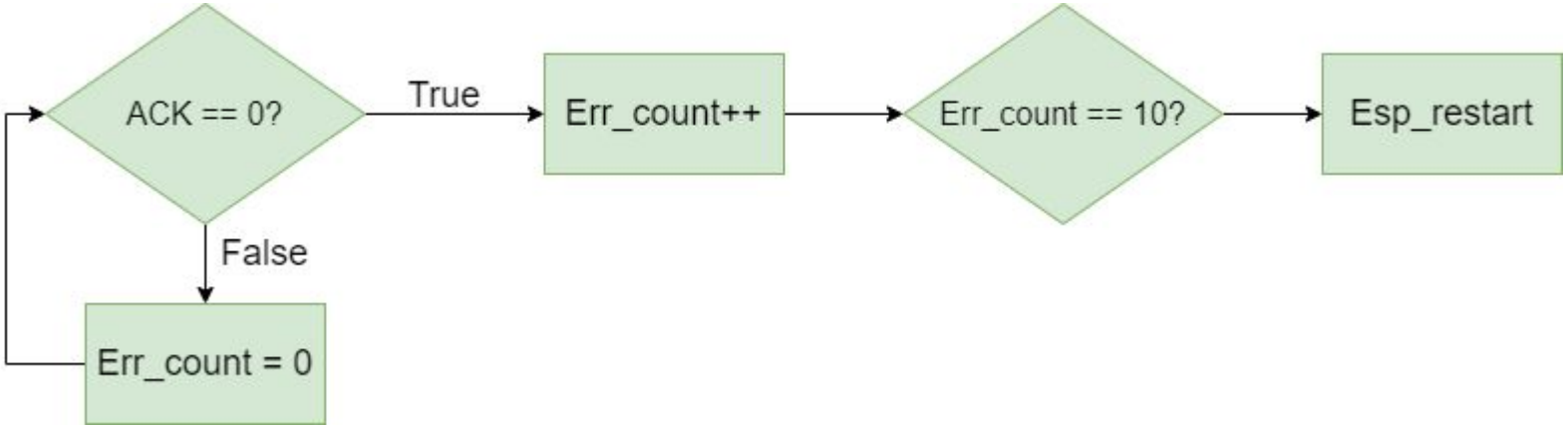
Chức năng:
Sử dụng vi điều khiển điều khiển module SIM NB-IoT gửi định kỳ 5 phút/lần lên nàng tảng IoT các dữ liệu:

- Toạ độ
- PCI
- RSRP
- RSRQ
- SINR
- cellID

Finite-state machine



Error Handler



Hệ thống sẽ phát hiện lỗi, nếu bị lỗi 10 lần liên tục sẽ tiến hành reset lại ESP32

Đối với việc ACK, chỉ check ở bước Connection, lấy CENG, lấy GNSS, bởi vì đây là bước lấy dữ liệu từ bên ngoài.

Các bước còn lại đa số là config nên hiếm khi có lỗi

CONFIG MQTT CONNETION

AT+SMCONF=\"URL\",\"demo.thingsboard.io\"\r");

AT+SMCONF="CLIENTID","135e5672-8c35-40e7-9854-ecd81a344ee7"
AT+SMCONF="USERNAME","4CndXRTNmiaqsORnNHd7"
AT+SMCONF="PASSWORD","4CndXRTNmiaqsORnNHd7"
AT+SMCONF?

Việc config không quá phức tạp, đầu tiên là phải đúng URL của server sẽ sử dụng.
Đúng USERNAME và PASSWORD, 1 vài server chỉ yêu cầu password là id device là đủ.
Và phải publish lên đúng feed AT+SMPUB="v1/devices/me/telemetry",50,0,1

CONFIG MQTT CONNETION

```
+SMCONF:
CLIENTID: "135e5672-8c35-40e7-9854-ecd81a344ee7"
URL: "demo.thingsboard.io",1883
KEEPTIME: 60
USERNAME: "4CndXRTNmiaqsORnNHd7"
PASSWORD: "4CndXRTNmiaqsORnNHd7"
CLEANSS: 0
QOS: 0
TOPIC: ""
MESSAGE: ""
RETAIN: 0
```

Kết quả connect sẽ như sau:
lúc này chỉ cần AT+SMCONN và đợi 10-20s là xong

ENGINEERING MODE

ta dùng lệnh: **AT+CENG?** để lấy được các thông tin về cell như sau

```
+CENG: 1,1,1,LTE NB-IOT

+CENG: 0,"1791,367,-70,-59,-
11,15,45981,151089173,452,04,167"
```

Sau đó dùng vài giải thuật tách chuỗi cơ bản để lấy được các chỉ số cần thiết, ví dụ ở đây:

{"psi":367,"rsrp":-70,"rsrq":-11,"sinr":15,"cellID":151089173}

GNSS

ta dùng lệnh: **AT+SGNSCMD=1,0** để lấy được các thông tin về GNSS như sau

```
+SGNSCMD: 1,15:54:32,10.75638,106.71021,12.51,-2.91,-
1.60,1.03,144.58,0x188a6042ec0,447
```

Sau đó dùng vài giải thuật tách chuỗi cơ bản để lấy được kinh độ vĩ độ
{"latitude":10.75621,"longitude":106.71019}

Chú ý nhỏ ở đây ta không dùng CGSNPWR=1, CGNSINF?, bởi vì lâu hơn và phức tạp, khi GNSSPWR còn mở thì không thể kết nối CENG và gửi bảng tin, buộc phải tắt GNSSPWR, và đợi khoảng 30s-60s.

Trong khi đó **AT+SGNSCMD=1,0** lại gọn hơn và ổn định hơn

PSM

ta dùng lệnh: **AT+CEREG=4** để yêu cầu áp dụng PSM lên mạng, kích hoạt đăng ký mạng và thông tin vị trí mã kết quả không mong muốn

Sau đó: **AT+CPSMS=1,,,"10001010","00100001"**,
"100 01010" = 30s x 10 = 300s = 5min
“001 00001” = 1min x 1 = 1min
(3bit đầu là mã đơn vị thời gian, 5 bit sau là số lần nhân lên)

<Requested_Periodic-TAU>

Unit	Base	Min. in Second	Max. in Second
0	10min	2400	18600
1	1h	21600	111600
2	10h	144000	1116000
3	2sec	0	62
4	30sec	90	930
5	1min	960	1860
6	320h	1152000	35712000

Unit	Base	Min. in Second	Max. in Second
0	2sec	0	62
1	1min	120	1860
2	6min	2160	11160

Table 2 <Requested_Active-Time> of AT+CPSMS

PSM

Nghĩa là ta sẽ hoạt động trong 1min, nếu không hành động nào trong 1min IDLE tiếp theo, module sẽ vào trạng thái PSM deep-sleep tiêu thụ cực kì ít năng lượng (3.2microA) trong 3phut.

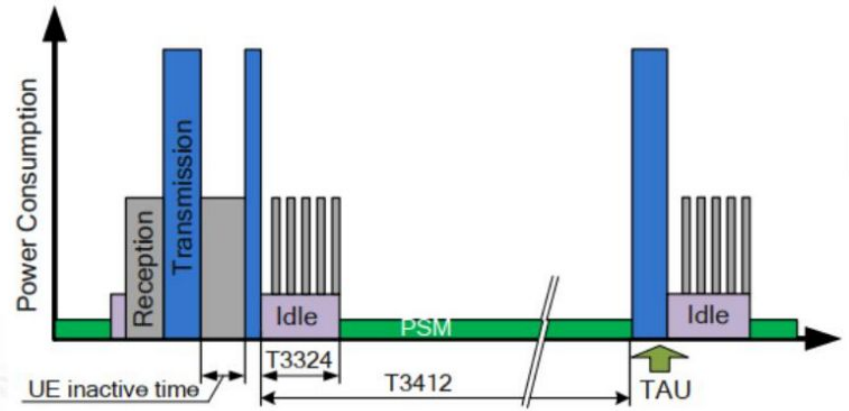


Figure1 PSM mode

Power Consumption
PSM 3.2 uA
Sleep 0.8 mA
Idle 10 mA

PSM

```
+CEREG: 1,"B39D","9017015",9,,,"00000000","01100000"

+CEREG: 1,"B39D","9017015",9,,,"00000000","10000100"

+CPSMRDP: 1,60,300,0,0,120
```

Chú ý +CPSMRDP: 1,60,300,0,0,120
60s ở đây là thời gian ILDE cho đến khi deep-sleep
300s chính là 5p cố định giữa 2 lần ta gửi bản tin

Hết thời gian IDLE: module tự động ngắt toàn bộ kết nối "DEACTIVE" và "ENTER PSM"

```
+APP PDP: 0,DEACTIVE

QCIMGBOOTTYPE : 1


+CPSMSTATUS: "ENTER PSM"
```

```
SMS Ready

+CPSMSTATUS: "EXIT PSM"
```

Đủ 5p: ta chủ động kích PWRKEY=0 để wakeup module. Module sẽ dậy nhanh hơn việc tắt nguồn, nhưng vẫn bị mất toàn bộ kết nối

PSM

Điểm mạnh của chế độ này là tiêu thụ cực ít năng lượng
nhưng đánh đổi sẽ mất toàn bộ kết nối khi đi ngủ và phải khởi động lại toàn bộ kết nối khi thức dậy.

Với yêu cầu đề bài là gửi bảng tin mỗi 5p, như vậy ta có 1p để gửi, 1 phút IDLE đợi tín hiệu trả lời từ cell, và 3p deepsleep. Thời gian ngủ là quá ít, nhưng lại phải kết nối lại liên tục sẽ gây ra lỗi không ổn định.

Chế độ này nên sử dụng nếu chu kì gửi bảng tin là vài tiếng hoặc 1 ngày thì tốt hơn.

=> ta chọn chế độ DRX tiếp theo.

eDRX

ta dùng lệnh: **"AT+CEDRXS=1,5,"0010"** để yêu cầu áp dụng eDRX

Trong đó 0010 nghĩa là chu kì DRX sẽ là 20.48s. Nghĩa là mỗi 20.48s thì module mới hoạt động 1 lần, còn lại sẽ vào chế độ standby (tiêu thụ 0.8mA)

```
+CEDRXRDP: 5,"0010","0010","0000"

OK
```

Power Consumption

PSM 3.2 uA

Sleep 0.8 mA

Idle 10 mA

eDRX value, octet 3 (bit 4 to 1)

The octet contains the eDRX value field. The values are listed in table3.

4	3	2	1	eDRX cycle length duration
0	0	0	0	5.12 seconds
0	0	0	1	10.24 seconds
0	0	1	0	20.48 seconds
0	0	1	1	40.96 seconds
0	1	0	0	61.44 seconds
0	1	0	1	81.92 seconds
0	1	1	0	102.4 seconds
0	1	1	1	122.88 seconds
1	0	0	0	143.36 seconds
1	0	0	1	163.84 seconds
1	0	1	0	327.68 seconds
1	0	1	1	655.36 seconds
1	1	0	0	1310.72 seconds
1	1	0	1	2621.44 seconds
1	1	1	0	5242.88 seconds
1	1	1	1	10485.76 seconds

eDRX

Như vậy, mặc dù tiêu thụ nhiều hơn PSM
Nhưng mà lợi thế sẽ duy trì được kết nối, với yêu cầu gửi bảng tin liên tục mỗi 5p thì eDRX chiếm lợi thế hơn

Ta không chọn eDRX quá 163s, bởi vì sẽ làm delay rất lâu với chu kì yêu cầu gửi 5p, và mặc định module sẽ vào deep sleep với delay > 163s. gây ra mất kết nối như PSM.

Ta nên chọn trong khoảng 20s-50s, nhưng nhóm chọn 20s là thời gian vừa đủ để tiết kiệm năng lượng và cũng không quá delay, rất an toàn.

NOTE:

If the cycle length is greater than or equal to 163.84 s, the module will enter into deep sleep.

The wake up requires the following conditions:

1) ***Cycle length timer is expired***
2) ***Pulling PWRKEY to low level (Typ. 800ms)***
3) ***Pulling RTC_EINT to low level***

Dashboard

Hiển thị trên bản đồ

- Hiển thị các thông tin trên bản đồ khi click, kiểm tra thông số RSRP và hiển thị màu và đánh dấu tương ứng với theo giá trị của RSRP
- Marker hiển thị giá trị mới nhất, các dữ liệu cũ hơn hiển thị bằng các vòng tròn theo màu tương ứng với giá trị của RSRP

RSRP	Color	Marker
> -70 dBm	Green	
-70 dBm to -105 dBm	Blue	
-106 dBm to -120 dBm	Orange	
< -120 dBm	Red	
No valuevalue	Black	

Dashboard

Các thông tin hiển thị trên bản đồ

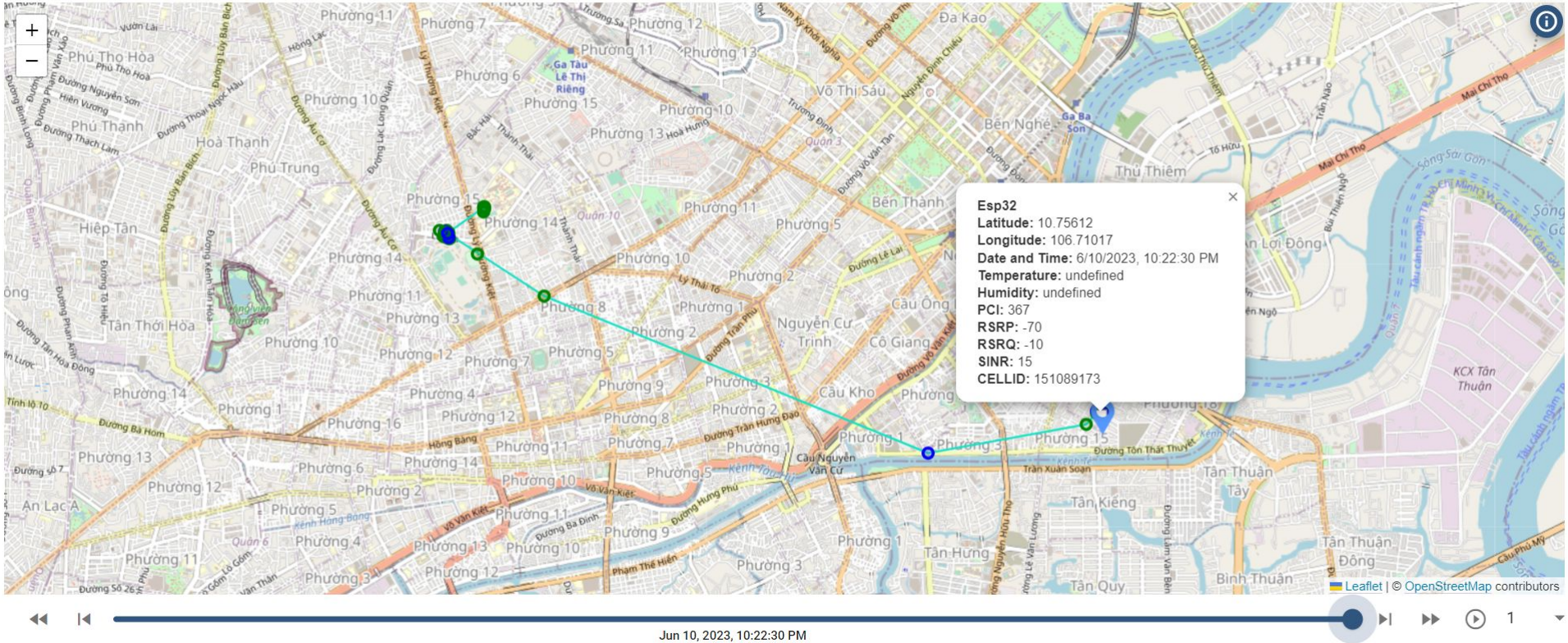
- Esp32: tên thiết bị
- Latitude: vĩ độ hiện tại của thiết bị
- Longitude: kinh độ hiện tại của thiết bị
- Date and Time: thời gian nhận được dữ liệu
- Temperature: nhiệt độ
- humidity: độ ẩm
- PCI: Physical Cell ID
- RSRP: Công suất của tín hiệu tham chiếu
- RSRQ: Chất lượng của tín hiệu tham chiếu nhận được
- SINR: Tỷ lệ tín hiệu nhiễu cộng với nhiễu
- CELLID: cell id, ở định dạng hexadecimal

Dashboard

Hiển thị trên bản đồ

Location

Esp32



Dashboard

Bảng hiển thị

- Hiển thị thông tin về thời gian nhận dữ liệu gửi lên, các thông số về longitude, latitude, pcci, rsrp, rsrq, sinr, cellID.

- Các thông số về humidity và temperature sẽ được gửi khi có sensor gắn vào ESP32, nếu không nhận giá trị null.

Dashboard

Bảng hiển thị

Table

Realtime - last day

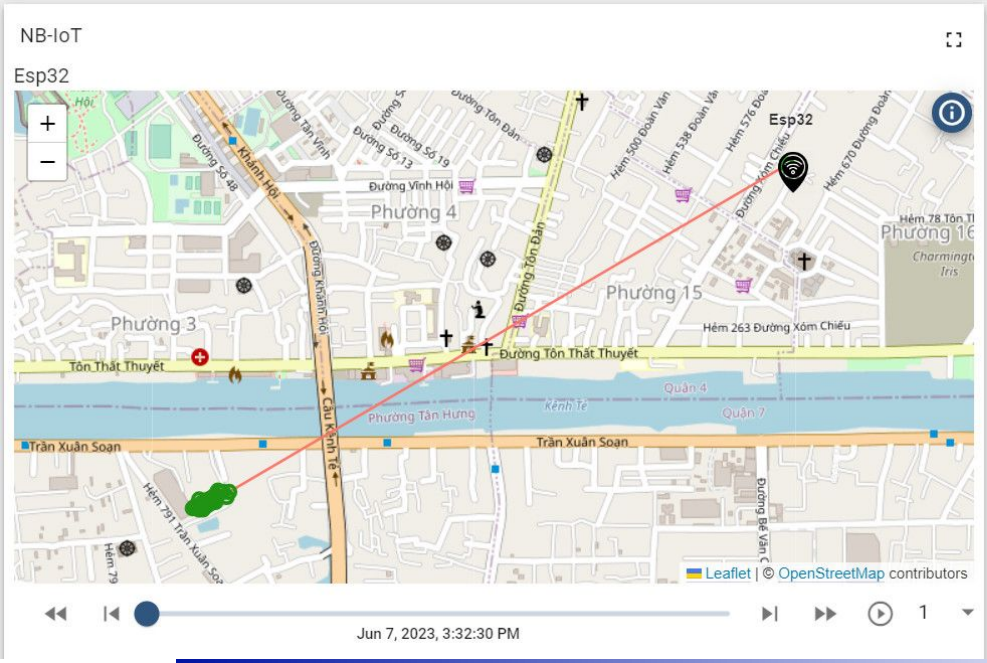
Timestamp ↓	latitude	longitude	pci	rsrp	rsrq	sinr	cellID
2023-06-10 22:01:05	10.75636	106.71027	367	-69	-10	13	151089173
2023-06-10 21:56:03	10.75635	106.71028	367	-76	-11	7	151089173
2023-06-10 21:51:03	10.75629	106.71037	367	-72	-10	13	151089173
2023-06-10 21:46:03	10.75634	106.71022	11	-83	-12	3	152064535
2023-06-10 21:41:03	10.75614	106.71012	419	-89	-20	-8	151762967
2023-06-10 21:36:06	10.75627	106.71016	367	-75	-10	7	151089173
2023-06-10 21:11:22	10.75643	106.71041	367	-69	-11	12	151089173
2023-06-10 21:06:28	10.75622	106.71029	419	-72	-12	2	151762967
2023-06-10 21:01:21	10.75634	106.71005	419	-73	-10	5	151762967
2023-06-10 20:53:12	10.75635	106.71023	367	-59	-10	20	151089173

Items per page: 10

Dashboard

Bảng hiển thị khi được gắn vào module đo nhiệt, ẩm bằng giao tiếp RS485 với rapshberry Pi CM4

Xem thêm ở **nhánh CM4** trên github



New Timeseries table

Realtime - last 30 minutes

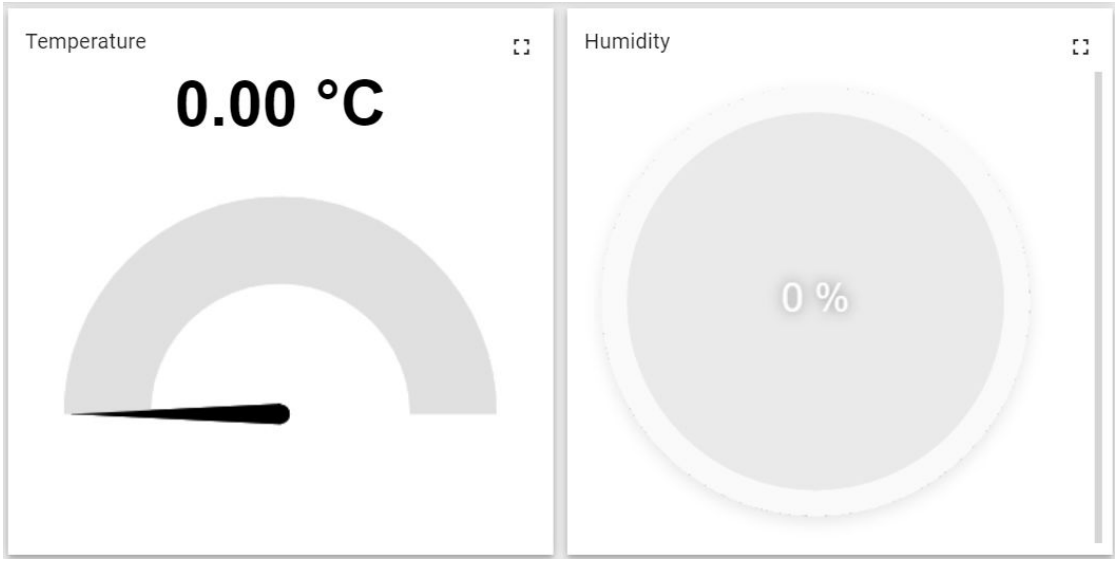
Timestamp	latitude	longitude	psi	rsrp	rsrq	sinr	cellID	mois	temp
2023-06-08 12:47:45			489	-65	-10	18	7842825	33	2371
2023-06-08 12:45:57			489	-59	-10	20	7842825	22	2359
2023-06-08 12:44:08			489	-59	-10	20	7842825	22	2359
2023-06-08 12:42:18			489	-59	-10	19	7842825	33	2368
2023-06-08 12:40:27			489	-60	-10	15	7842825	33	2365
2023-06-08 12:38:39			489	-60	-10	21	7842825	44	2365
2023-06-08 12:36:38			489	-60	-10	21	7842825	33	2368
2023-06-08 12:34:50			489	-59	-10	15	7842825	44	2374
2023-06-08 12:33:00			489	-60	-10	21	7842825	33	2377

Items per page: 10 1 – 10 of 16

Dashboard

Widget hiển thị thông tin về nhiệt độ và độ ẩm

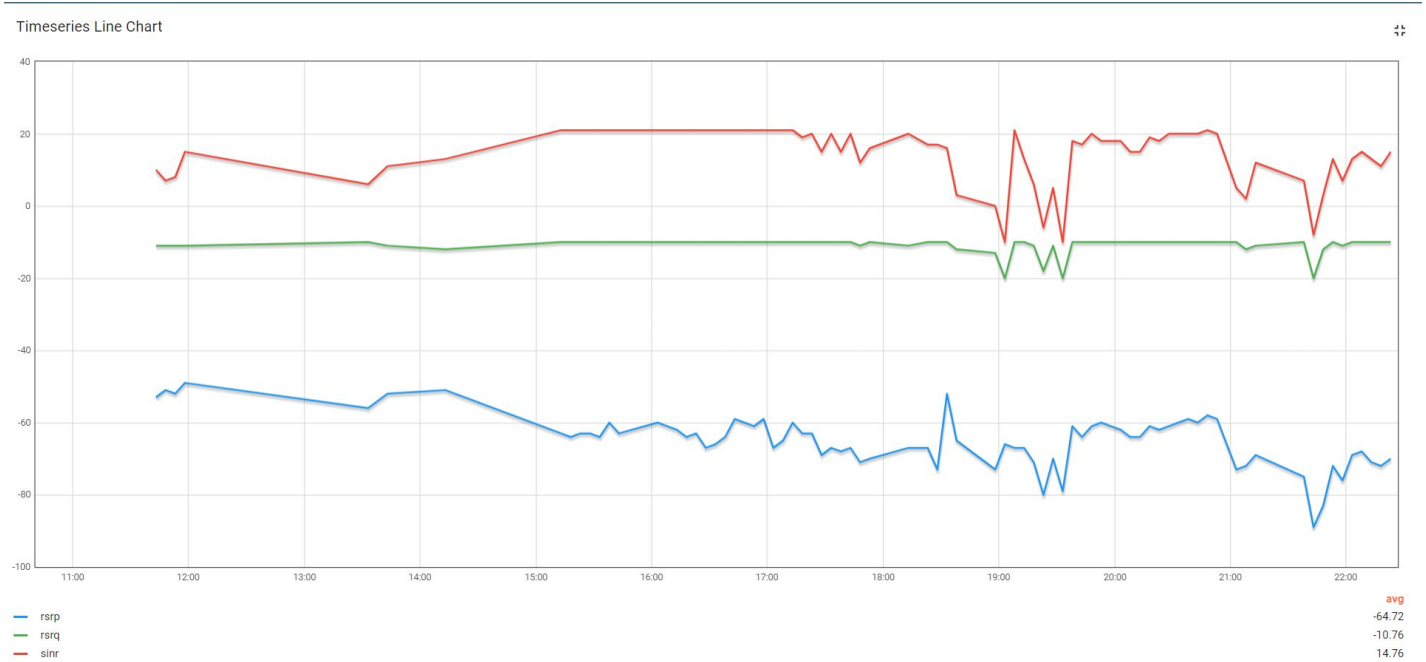
- Thông tin về nhiệt độ và độ ẩm sẽ được hiển thị bằng các widget như hình:



Dashboard

Timeseries line chart

- Biểu diễn sự thay đổi giá trị theo thời gian của 3 thông số RSRP, RSRQ, SINR



Link

Link Dashboard:

https://demo.thingsboard.io/dashboard/b42e5330-fcad-11ed-9029-87706d0da53c?publicId=676aaee0-fcad-11ed-9029-87706d0da53c

Link GITHUB:

JackWrion/NBIOT (github.com)

https://github.com/JackWrion/NBIOT

bao gồm
nhánh main: là chương trình ổn định sử dụng code cứng không có chế độ tiết kiệm năng lượng
nhánh PSM: sử dụng chế độ PSM cho module SIMCOM7090G
nhánh DRX: sử dụng chế độ eDRX cho module SIMCOM7090G
nhánh CM4: Kết nối với CM4 rashpbery thông qua RS485 lấy nhiệt, ẩm