

솔라사이니지 매뉴얼

1. 태양광 발전 및 사물 인터넷으로 에너지 생산 모니터링 구성 내용

- 1-1. 공통 사항
- 1-2. 라즈베리파이
- 1-3. 모니터링 웹 서버

2. 공통 사항

- 2-1. JAVA 설치
- 2-2. Nodejs 설치

3. OM2M 설치 및 리소스 설정

- 3-1. OM2M 설치 및 실행
- 3-2. Solar Signage 리소스 설정

4. 라즈베리파이 설정

- 4-1. 고정 IP 설정
- 4-2. GPS 설정
- 4-3. Modbus 드라이버 설치
- 4-4. Solar-AE/gpsd-client.js
- 4-5. Modbus-IPE-master-v0.5
- 4-6. Bus Shelter

5. 모니터링 웹 서버 설정

- 5-1. [web]solar-wrapper 설정
- 5-2. [web]solar-ui 설정

6. 데몬 등록

- 6-1. 라즈베리파이
- 6-2. 모니터링 웹 서버

7. 유의사항

- 7-1 Modbus-IPE가 밤에 꺼지는 경우
- 7-2 UI Dashboard 부분의 변화

1. 태양광 발전 및 사물 인터넷으로 에너지 생산 모니터링 구성 내용

1-1. 공통 사항

- Nodejs, JAVA 설치

1-2. 라즈베리파이

- Modbus IPE, Solar-AE, Bus Shelter 설치
- 고정 IP 설정 필요
- 라즈베리파이 설정에서 상세히 설명

1-3. 모니터링 웹 서버

- Cloud Service 사용 (ex. Amazon Web Service, Google Cloud Platform)
- solar-ui 및 solar-wrapper 설치
- 모니터링 웹 서버 구축에서 상세히 설명

2. 공통 사항

2-1. JAVA 설치

Java LTS version 8 설치 스크립트

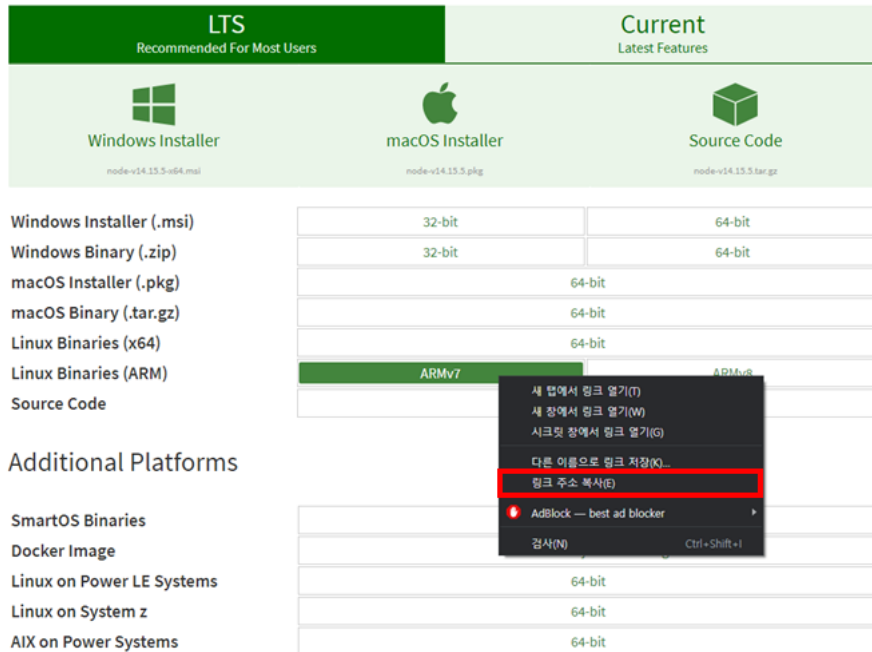
```
$ sudo apt update
$ sudo apt install openjdk-8-jdk
```

Java version 확인:

```
$ java -version
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.0_212-8u212-b01-1+rp11-b01)
OpenJDK Client VM (build 25.212-b01, mixed mode)
```

2-2. Nodejs 설치

<https://nodejs.org/en/download/>



Linux Binaries (ARM) - ARMv7 마우스 우클릭 - 링크 주소 복사

- 라즈베리 파이에 설치할 경우 (ARM architecture)

```
$ wget https://nodejs.org/dist/v14.15.5/node-v14.15.5-linux-armv7l.tar.xz
$ tar -xvf node-v14.15.5-linux-armv7l.tar.xz
$ sudo mv node-v14.15.5-linux-armv7l /opt/nodejs
$ sudo ln -s /opt/nodejs/bin/node /usr/bin/node
$ sudo ln -s /opt/nodejs/bin/npm /usr/bin/npm
$ sudo ln -s /opt/nodejs/bin/npx /usr/bin/npx
```

- 클라우드 웹 서버에 설치할 경우 (x86 architecture)

```
$ wget https://nodejs.org/download/release/v14.15.5/node-v14.15.5-linux-x64.tar.xz
$ tar -xvf node-v14.15.5-linux-x64.tar.xz
$ sudo mv node-v14.15.5-linux-x64 /opt/nodejs
$ sudo ln -s /opt/nodejs/bin/node /usr/bin/node
$ sudo ln -s /opt/nodejs/bin/npm /usr/bin/npm
$ sudo ln -s /opt/nodejs/bin/npx /usr/bin/npx
```

- node 16 버전 이상은 npm install 시 이슈 발생이 확인되어 14.15.5 버전 권고

3. OM2M 설치 및 리소스 설정

3-1. OM2M 설치 및 실행


- OM2M Server
 - <https://wiki.eclipse.org/OM2M/Download>

Official Releases

Download	Date	
Eclipse OM2M v1.4.1	February 15, 2021	Bug fi
Eclipse OM2M v1.4.0	August 5, 2020	New f
Eclipse OM2M v1.3.0	March 13, 2018	Bug fi
Eclipse OM2M v1.2.0	January 04, 2018	Bug fi
Eclipse OM2M v1.1.0	October 09, 2017	oneM
Eclipse OM2M v1.0.0	June 22, 2016	oneM
Eclipse OM2M v0.8.0	April 8, 2015	smart

Official Releases - 최신 버전 클릭

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.



Download from: Japan - Yamagata University (<https>)

File: [eclipse-om2m-v1-4-1.zip](#) SHA-512

[>> Select Another Mirror](#)

다운로드 클릭


Thank you for your download!

If the download doesn't start in a few seconds, please [click here](#) to start the download.

Eclipse technologies are 100% free and open source and used by millions of developers. Please contribute toward the future development of Eclipse projects.

Donate to the Eclipse Community

Contributions from users like you help fund the operations of the Eclipse Foundation. All donations are used to support the Eclipse Community.



- 새 탭에서 링크 열기(T)
- 새 창에서 링크 열기(W)
- 시크릿 창에서 링크 열기(G)
- 다른 이름으로 링크 저장(Q)
- 링크 주소 복사(C)**
- AdBlock — best ad blocker
- 검사(N) Ctrl+Shift+I

click here 우클릭 - 링크 주소 복사

```
$ wget https://ftp.yz.yamagata-u.ac.jp/pub/eclipse//om2m/releases/1.4.1/eclipse-om2m-v1-4-1.zip
$ unzip eclipse-om2m-v1-4-1.zip
$ mv eclipse-om2m-v1-4-1/eclipse-om2m-v1-4-1/ ~/om2m
$ rmdir eclipse-om2m-v1-4-1
$ cd ~/om2m/in-cse
$ sudo chmod +x start.sh
$ sudo ./start.sh
```

확인 : <http://<raspberry-pi-ip-address>:8080/webpage/>



username:

password:

Login

username : admin

password : admin

3-2. Solar Signage 리소스 설정

- Python Script 사용

- OM2M/init_ipe.py, OM2M/init_bus.py

1. init_ipe.py

```
host = '<om2m_server_address>'
port = '8080'

modbus_ipe_cnt_name = 'Modbus_IPE'
solar_ae_cnt_name = 'Solar_AE'

headers = {
    'X-M2M-Origin': '<om2m-id>:<om2m-passwd>',
    'Content-Type': '',
    'Cache-Control': 'no-cache',
}
```

<modbus_address>: Modbus IPE가 설치된 장치의 IP

<wrapper_addr>: solar wrapper가 실행 중인 장치의 IP

<om2m_server_address>: OM2M이 동작 중인 장치의 IP

<om2m-id>:<om2m-passwd>: om2m 아이디, 패스워드. 기본값 admin:admin

Modbus-IPE와 Solar의 AE, fcnt를 모두 생성하고 난 후 press any key 에서 아래의 설정을 진행한 후 엔터를 눌러 진행 (wrapper가 꺼져 있을 경우 Subscription이 생성되지 않음)

2. init_bus.py

```
bus_addr = '<bus_shelter_address>'

host = '<om2m_server_address>'
port = '8080'

ae_name = 'bus_shelter'
headers = {
    'X-M2M-Origin': '<om2m-id>:<om2m-passwd>',
    'Content-Type': '',
    'Cache-Control': 'no-cache',
}
```

<bus_shelter_address>: 버스 셸터가 설치된 장치의 IP

<om2m_server_address>: OM2M이 동작 중인 장치의 IP

<om2m-id>:<om2m-passwd>: om2m 아이디, 패스워드. 기본값 admin:admin

wrapper가 꺼져 있을 경우 Subscription이 생성되지 않음

4. 라즈베리파이 설정

4-1. 고정 IP 설정

1. 현재 연결되어 있는 IP 및 netmask 확인

```
$ ifconfig
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.100 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::9f6e:4cda:20d2:f90b prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:32:a2:16 txqueuelen 1000 (Ethernet)
    RX packets 235 bytes 26377 (25.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 98 bytes 14085 (13.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. gateway 확인 방법

```
$ netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 192.168.0.1 0.0.0.0 UG 0 0 0 wlan0
192.168.0.0 0.0.0.0 255.255.255.0 U 0 0 0 wlan0
```

3. `/etc/dhcpd.conf` 의 다음 코드 부분을 수정

```
$ sudo vim /etc/dhcpd.conf

interface wlan0
static ip_address=192.168.0.100/24
# static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
static netmask=255.255.255.0
```

`static ip_address` 부분에는 gateway에 맞춰서 고정할 ip 작성

`static routers` 부분에는 gateway 주소(맨 끝자리는 1 고정)

4. 네트워크 재부팅 및 라즈베리 파이 재부팅 진행

```
# 네트워크 재부팅
$ sudo /etc/init.d/networking restart

# 라즈베리 파이 재부팅
$ sudo reboot
```

4-2. GPS 설정

Reference : https://wiki.52pi.com/index.php/USB-Port-GPS_Module_SKU:EZ-0048

1. gpsd 관련 패키지 설치

```
$ sudo apt update && sudo apt -y install gpsd gpsd-clients python-gps
```

2. gpsd 서비스 등록 및 제어

```
# 서비스 등록
$ sudo systemctl enable gpsd.socket

# 서비스 시작
$ sudo systemctl start gpsd.socket

# 서비스 재시작
$ sudo systemctl restart gpsd.socket
```

```
# 서비스 상태 확인
$ sudo systemctl status gpsd.socket
```

3. /etc/default/gpsd 의 gpsd 설정 파일 수정

/dev 폴더 내부의 시리얼 포트 이름에 따라서 DEVICES 파라미터 수정, USB 케이블을 통해 라즈베리파이에 연결된 것이 /dev/ttyUSB0 면 nano 나 vim 을 통해 해당 부분을 아래와 같이 수정

```
$ grep -v "#" /etc/default/gpsd | grep -v "^$"
START_DAEMON="true"
USB_AUTO="true"
DEVICES="/dev/ttyUSB0"
GPSD_OPTIONS="-F /var/run/gpsd.sock"
```

서비스 재시작 : `sudo systemctl restart gpsd.socket`

GPS 모듈에서 정보 불러오기 : `sudo cgps -s`

4-3. Modbus 드라이버 설치

- Github 주소
 - <https://github.com/kasbert/epsolar-tracer>

```
$ git clone https://github.com/kasbert/epsolar-tracer
$ cd epsolar-tracer/xr_usb_serial_common-1a
$ make
$ sudo insmod ./xr_usb_serial_common.ko
```

- Make 오류 날 경우

```
$ sudo apt install raspberrypi-kernel
$ sudo apt install raspberrypi-kernel-headers
$ sudo reboot
```

- 디버깅 팁

```
# 시스템에 연결된 USB UART 확인
$ lsusb
Bus 001 Device 004: ID 0424:7800 Standard Microsystems Corp.
Bus 001 Device 003: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

# Exar USB UART용 CDC-ACM 드라이버가 설치되지 않았는지 확인
$ ls /dev/tty*
CDC-ACM 드라이버 삭제와 드라이버 설치:
# rmmod cdc-acm
# modprobe -r usbserial
# modprobe usbserial
# insmod ./xr_usb_serial_common.ko
```

- 홈 디렉터리에 epsolar.sh 생성

```
rmmod cdc-acm
modprobe -r usbserial
modprobe usbserial
insmod ~/epsolar-tracer/xr_usb_serial_common-1a/xr_usb_serial_common.ko
```

- epsolar.sh 에 실행 권한 부여

```
$ sudo chmod +x ~/epsolar
```

4-4. Solar-AE/gpsd-client.js

- GPS 값과 현재 시간을 서버에 올려주는 역할

- GPS 설정을 먼저 진행
- om2m 서버 주소 수정 (gpsd-client.js)
- DeviceInfo flex con

<http://192.168.0.100:8080/~in-cse/fcnt-892985304>

```

- in-name
  |
  | acp_admin
  | SDT_Home_Monitoring_Application_ACP
  | ACP_Device_Admin_1615293416157
  | SDT_Home_Monitoring_Application
  | SDT_IPE
  | Modbus_IPE
  | Solar_AE
  |
  | fcnt_812075511
  | fcnt 892985304

```

```

//import request from 'request-promise';

const request = require('request-promise')
const Gpsd = require('node-gpsd-client')
var moment = require('moment');

const GPS_URL = 'http://<om2m-server-ip>:8080/~in-cse/fcnt-892985304';

```

- GPSD nodejs 실행

```

$ npm install
$ node gpsd-client.js

```

4-5. Modbus-IPE-master-v0.5

- USB Driver 설정 먼저 진행
- `config.js` 파일에서 OM2M 서버 주소 및 fcnt (flex container) 변경

```

export const slaveConfig = {
  port: "/dev/ttyXRUSB0",
  baudRate: 115200,
  parity: "none",
  dataBits: 8,
  stopBits: 1
};

export const cseUrl = 'http://<om2m-server-ip>:8080';

export const fcntUrls = {
  battery: '/~/in-cse/fcnt-101716600',
  energyGeneration: '/~/in-cse/fcnt-416305009',
  energyConsumption: '/~/in-cse/fcnt-379612474'
};

```

- 실행

```

$ npm install
$ npm start

```

4-6. Bus Shelter

- 실행

```
$ npm install
$ node app.js
```

5. 모니터링 웹 서버 설정

5-1. [web]solar-wrapper 설정

- OM2M 서버 주소 변경 : `server_express.js`

```
var om2mServer = "http://<om2m-server-ip>:8080";

var om2mServer_BATTERY = om2mServer + "/~/in-cse/<battery-fcnt>";
var om2mServer_GPS = om2mServer + "/~/in-cse/<deviceinfo-fcnt>";
var om2mServer_AD = om2mServer + "/~/in-cse/<userinfo-fcnt>";

// If you don't use bus shelter, comment this block
var om2mServer_LED1 = om2mServer + "/~/in-cse/<led1-cnt>";
var om2mServer_LED2 = om2mServer + "/~/in-cse/<led2-cnt>";
var om2mServer_AWNING = om2mServer + "/~/in-cse/<awning-cnt>";

var default_AD_URL = "https://www.youtube.com/embed/xLD8oWRmIAE";
```

- `om2mServer_BATTERY` : Modbus_IPE 밑의 battery flex container 주소
- `om2mServer_GPS` : Solar_AE 밑의 deviceinfo flex container 주소
- `om2mServer_AD` : Solar_AE 밑의 userinfo flex container 주소
- `default_AD_URL` : 사용자 광고 값을 못 얻어올 때 default 설정 값
- `om2mServer_LED1` : bus_shelter 밑의 LED1 container 주소
- `om2mServer_LED2` : bus_shelter 밑의 LED2 container 주소
- `om2mServer_AWNING` : bus_shelter 밑의 AWNING container 주소

- 실행

```
$ npm install
$ node server_express.js
```

5-2. [web]solar-ui 설정

- OM2M 서버 주소 변경 : `config.js`

```
//export const UI_SOCKET_URL = "http://192.168.0.202:19997";
export const UI_SOCKET_URL = "http://<web-server-ip>:19997";

//export const WRAPPER_URL = "http://192.168.0.202:19998";
export const WRAPPER_URL = "http://<web-server-ip>:19998"

export const PUSH_CHARGING_URL = WRAPPER_URL + "/charging";
export const PUSH_DISCHARGING_URL = WRAPPER_URL + "/discharging";

// If you don't use bus shelter, comment this block
export const PUSH_LED1_URL = WRAPPER_URL + "/led1";
export const PUSH_LED2_URL = WRAPPER_URL + "/led2";
export const PUSH_AWNING_URL = WRAPPER_URL + "/awning";

export const STARTTIME_WRAPPER_URL = WRAPPER_URL + "/starttime";

export const GPS_WRAPPER_URL = WRAPPER_URL + "/gps";

export const AD_WRAPPER_URL = WRAPPER_URL + "/ad";
```

IP 주소는 cloud service의 IP 사용

- 실행

```
$ npm install
$ npm run dev
```

- 날씨 예보 서비스, Google Cloud Platform : API Key 변경 필요

- `index.html`, `app.js`

```
// index.html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Solar Signage</title>
  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDjiacZxDWas3jpR-CdDWeUGv651E5eoGI&libraries=places"></script>
  <!--<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC-lCjpg1xbw-nsCc11Si8Ldg2LKYizqI4&libraries=places"></script>
  <!--<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC-lCjpg1xbw-nsCc11Si8Ldg2LKYizqI4&libraries=places"></script>
</head>

<body>
  <div id="app"></div>
  <script src="dist/build.js"></script>
</body>

</html>
```

6. 데몬 등록

- 재부팅 시 자동 실행 프로그램 등록
- 데몬 경로 `etc/systemd/system`

/home/solar/solar-signage/raspberryPi_IPE/Modbus-IPE-master-v0.5

6-1. 라즈베리파이

1. epsolar driver setting : `sudo nano /etc/systemd/system/epsolar.service`

```
[Unit]
Description = Epsolar driver setting
After = om2m.service

[Service]
WorkingDirectory = /home/pi
ExecStart = sudo ./epsolar.sh

[Install]
WantedBy = multi-user.target
```

2. Solar AE 실행 : `sudo nano /etc/systemd/system/solarAE.service`

```
[Unit]
Description = Solar AE script
After = epsolar.service

[Service]
WorkingDirectory = /home/pi/project/Solar-AE
ExecStart = node gpsd-client.js

[Install]
WantedBy = multi-user.target
```

3. Modbus IPE 실행 : `sudo nano /etc/systemd/system/modbusipe.service`

```
[Unit]
Description = Modbus IPE script
After = solarAE.service

[Service]
WorkingDirectory = /home/pi/project/Modbus-IPE-master-v0.5
ExecStart = npm start

[Install]
WantedBy = multi-user.target
```

4. bus_shelter 실행: `sudo nano /etc/systemd/system/busshelter.service`

```
[Unit]
Description = Bus Shelter script
After = modbusipe.service

[Service]
WorkingDirectory = /home/pi/project/bus_shelter
ExecStart = node app.js

[Install]
WantedBy = multi-user.target
```

5. 서비스 활성화 및 시작

```
# 활성화
$ sudo systemctl enable epsolar.service
$ sudo systemctl enable solarAE.service
$ sudo systemctl enable modbusipe.service
$ sudo systemctl enable busshelter.service

# 시작
$ sudo systemctl start epsolar.service
$ sudo systemctl start solarAE.service
$ sudo systemctl start modbusipe.service
$ sudo systemctl start busshelter.service
```

6-2. 모니터링 웹 서버

1. Solar wrapper 실행 : `sudo nano /etc/systemd/system/solarwrapper.service`

```
[Unit]
Description = Solar Wrapper script
After = network.target

[Service]
WorkingDirectory = /home/username/solar-signage-v1.0/[web]solar-wrapper
ExecStart = sudo node server_express.js

[Install]
WantedBy = multi-user.target
```

2. Solar UI 실행 : `sudo nano /etc/systemd/system/solar.service`

```
[Unit]
Description = Solar script
After = solarwrapper.service

[Service]
WorkingDirectory = /home/username/solar-signage-v1.0/[web]solar-ui
ExecStart = npm run dev

[Install]
WantedBy = multi-user.target
```

3. 서비스 활성화 및 시작

```
# 활성화
$ sudo systemctl enable solarwrapper.service
$ sudo systemctl enable solar.service
```

```
# 시작
$ sudo systemctl start solarwrapper.service
$ sudo systemctl start solar.service
```

- 서비스 종료 및 비활성화

```
# 종료
$ sudo systemctl stop 서비스이름

# 비활성화
$ sudo systemctl disable 서비스이름
```

- 서비스 상태 확인

- `sudo systemctl status 서비스이름`

- om2m 서비스: `sudo vi /etc/systemd/system/om2m.service`

```
[Unit]
Description = OM2M script
After = network.target

[Service]
WorkingDirectory = /home/pi/OM2M/in-cse
ExecStart = sudo ./start.sh

[Install]
WantedBy = multi-user.target
```

활성화 : `sudo systemctl enable om2m.service`

시작: `sudo systemctl start om2m.service`

7. 유의사항

7-1 Modbus-IPE가 밤에 꺼지는 경우

- 보완코드가 작성된 modbus-ipe를 적용

<https://github.com/seslabSJU/2022-SOLARSIGNAGE>

await을 쓰는 부분에 try catch문을 활용하여 에러 핸들을 잡음

7-2 UI Dashboard 부분의 변화

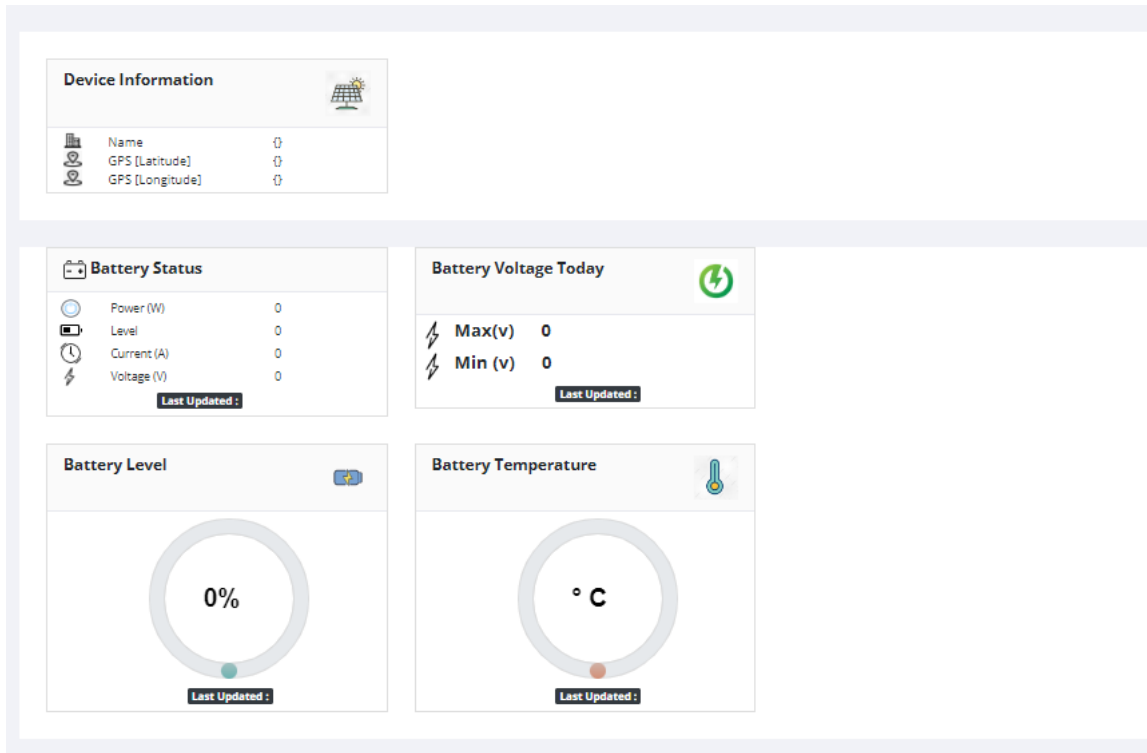
- 버스쉘터 대시보드의 변화

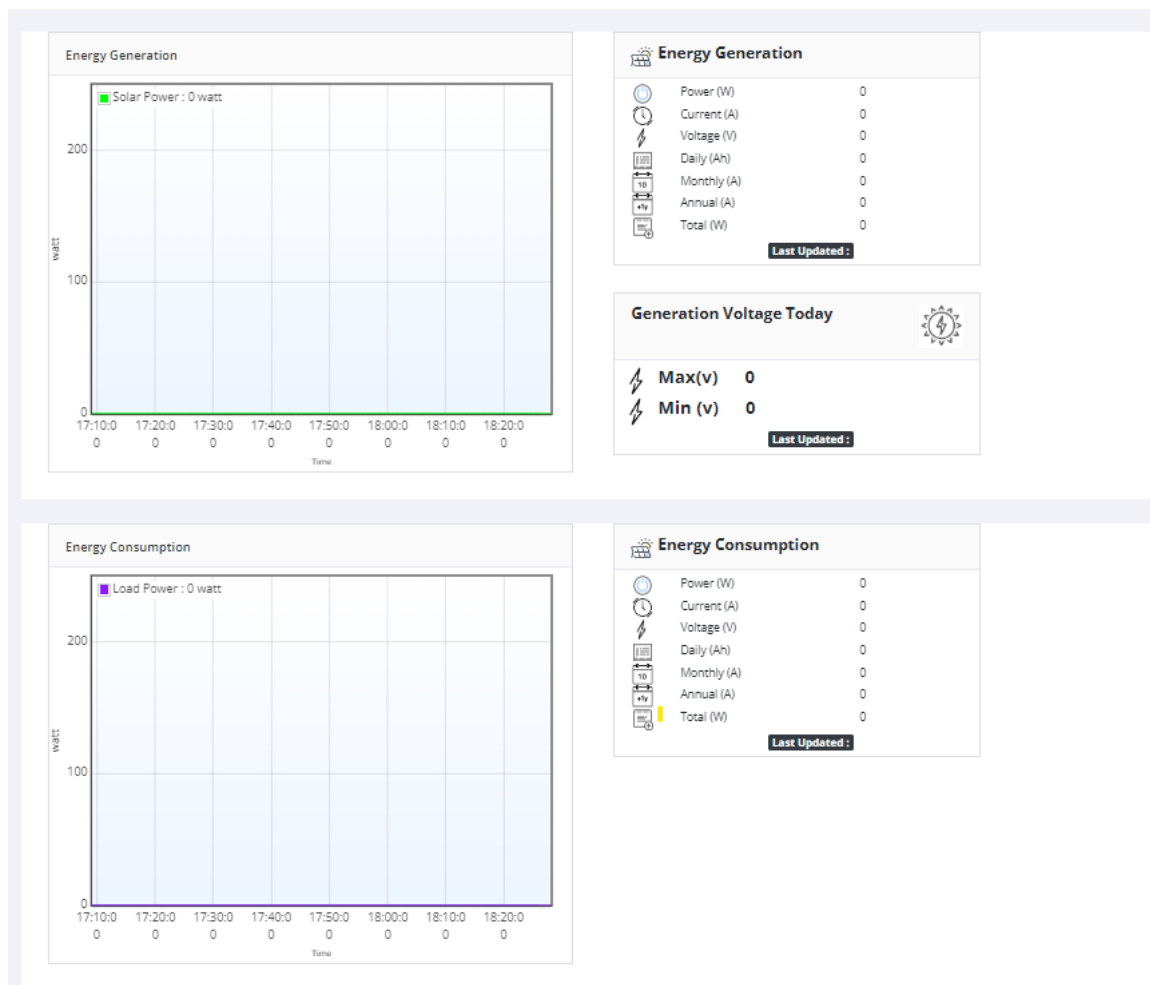
<https://github.com/seslabSJU/2022-SOLARSIGNAGE>

1. 메뉴 간략화

버스쉘터에서 필요한 부분만 간략화 하여 메뉴를 간략화하였음

2. 대시보드의 변화





battery, generation, consumption 부분을 합쳐 좀 더 데이터를 편하게 볼 수 있게 함.