BESC Modbus IOT SDK Integrated System

Index

BESC Modbus IOT SDK Integrated System 1
Create Project and Contract 2
Formula and Baseline Integration 2
Application and program installation 2
Configure Environment File 3
Configure Devices and Project setting 4
Basic Setting of the project: 4
Poll: 4
Devices: 6
Complete Poll sample for serial connection: 7
Complete Poll sample for TCP connection: 8
Start the application 10
Devices Data 10
Debugging 10
Make an exe file if bat file can't run 11
Update SDK version 11

Create Project and Contract

1. Register an account on https://portal.besc.online.
2. Click on the form link which will be redirected to https://kyc.besc.online, complete the KYC form registration.
3. Once approved by BESC, login to registered account and proceed to **Create Project**.
4. After project is created, create a contract and awaits vote approval by stakeholders.

Formula and Baseline Integration

Baseline reading and Schema file uploaded during contract creation process will be use as a reference to generate the correct energy record.

Get Baseline Sample:

```
var baseline = await helper.getBaseline(host_client, keypair);
```

Get Formula Sample:

```
var formulas = await helper.getAllFormulas(host_client, keypair);

var f1 = formulas["EnergyEfficiency"].duplicate();

// formula Name is not fixed
f1.applyFieldsValues({h: 10, w: 10, d: 10});
```

Any modification of formula and baseline has to be done by updating existing contract

---

Application and program installation

There is a program that need to be installed first which is the NodeJs. It is provided in the folder, named "node-v{version}.msi"

There is also optional text editor, notepad++ available to install. It will make text editing much easier.

Configure Environment File

There is a file named ".env" in the folder, it will be setting for the application

Sample:

```
APIKEY=DXDSdLeJ07zEL4LML39qf9IzMVTt1Q0q
PROJECT_ID=Testing
REPEAT_EVERY_MINUTES=30
BESC_ESS_API_PATH=
DISABLE_DATA_SEND=
```

Explanation:

Key	Description
APIKEY	A secret string provided by BESC to send the energy data to the ESS-API.
PROJECT_ID	A string provided by BESC to represent the project, it must be used with the correct APIKEY.
REPEAT_EVERY_MINUTES	Repeat the reading and data pushing to ESS-API every n minutes.
BESC_ESS_API_PATH	It is the URL to the ESS-API, it will pass to the SDK. If it is empty, SDK will use default URL.
DISABLE_DATA_SEND	Leave it empty for the application to send data to ESS-API, if not empty, it will stop the send of data to ESS-API. It is used for data reading testing without sending the data out

---

Configure Devices and Project setting

There will be a config.json in the folder, this is when all the device will be configurated along with project details.

This is the overall layout of the config.json :

```
{
        "ProjectName": "Testing",
        "Location": "101.1212, 112.1133",
        "AverageRT": 0,
        "Polls": [
            ...
        ]
}
```

Basic Setting of the project:

Name	Description
ProjectName	The project name given to this group of energy data.
Location	A string reprensetative of the project location, it should be the longitude and latitude of the GPS location
AverageRT	Set the average RT
Polls	It is the data pulling with different set of setting

Inside the polls, you will need to set each poll.

Poll:

Poll sample for serial connection:

Using Serial Setting (only for direct port connection)

Name	Description	Possible Value
type	The type of modbus connection	TCP (for TCP setting), Serial (for serial setting)

Name	Description	Possible Value
protocol	The protocol used for the modbus serial connection	RTU/ ASCII
port	The port which will used by the modbus serial connection.	Windows - COM{number} (eg. COM1, COM2) Linux - /dev/ttyS{number} (eg. /dev/ttyS0, /dev/ttyS1)
baudRate	Baud rate setting of the modbus serial connection.	Eg. 9600
dataBits	Data bits setting of the modbus serial connection	Eg. 8
stopBits	Stop bits setting of the modbus serial connection	Eg. 1
parity	The parity setting of the modbus serial connection	none, odd, even
Devices	The devices included in the modbus poll, it can be multiple devices with different address setting.	List of devices

Poll sample for TCP connection:

Using TCP Setting

Name	Description	Possible Value
type	The type of modbus connection	TCP (for TCP setting), Serial (for serial setting)
host	The host where the modbus needed to connect to.	Any IP address (127.0.0.1)
port	The port number which will used by the modbus TCP connection.	Any, usually use port 502

Name	Description	Possible Value
Devices	The devices included in the modbus poll, it can be multiple devices with different address setting.	List of devices

Devices:

Sample of device

```
{
        "name": "Device 1",
        "deviceNum": 1,
        "address": 1,
        "dataType": "int",
        "mod": "",
        "registerLength": 2,
        "registerType": 3,
        "registerBit": 16,
        "endian": "BE"
}
```

Devices is a collection of the device in the modbus poll.

Name	Description	Possible Value
name	The name given to the device, it must be unique in the whole config file.	Any
deviceNum	The device id used by modbus.	Any integer (eg. 1, 2, 3)
address	The registers address of the data needed to read.	Any integer, the address same as ModScan (without -1 manually)
dataType	The data type represent by the registers addresses.	"int", "float", "floating_point"
mod	The modification do to the reading after getting the value from device. The reading is represent by the "reading" keyword. It can be used to do mathematical expression. Can be left empty.	Eg. "reading / 10", "reading * 10"

Name	Description	Possible Value
registerLength	The number of registers to represent a single reading	1, 2, 4
registerType	The registers type that need to be read from.	3 - Holding Register 4 - Input Register
registerBit	The total bit represent by a single registers address.	8, 16, 32
endian	The endianess of the registers address reading	BE - Big Endian LE - Little Endian. Most of the device will be BE.

Complete Poll sample for serial connection:

```
{
	"type": "Serial",
	"protocol": "RTU",
	"port": "COM5",
	"baudRate": 9600,
	"dataBits": 8,
	"stopBits": 1,
	"parity": "none",
	"Devices": [
	  {
	    "name": "Device 1",
	    "deviceNum": 1,
	    "address": 1,
	    "dataType": "int",
	    "mod": "",
	    "registerLength": 2,
	    "registerType": 3,
	    "registerBit": 16,
	    "endian": "BE"
	  },
	  {
	    "name": "Device 2",
	    "deviceNum": 3,
	    "address": 3,
	    "dataType": "int",
	    "mod": "",
	    "registerLength": 2,
	    "registerType": 3,
	    "registerBit": 16,
```

```
            "endian": "BE"
        }
      ]
    }
```

Complete Poll sample for TCP connection:

```
{
        "type": "TCP",
        "host": "127.0.0.1",
        "port": 502,
        "Devices": [
          {
            "name": "Device 4",
            "deviceNum": 1,
            "address": 1,
            "dataType": "int",
            "mod": "",
            "registerLength": 2,
            "registerType": 3,
            "registerBit": 16,
            "endian": "BE"
          },
          {
            "name": "Device 3",
            "deviceNum": 2,
            "address": 3,
            "dataType": "int",
            "mod": "",
            "registerLength": 2,
            "registerType": 3,
            "registerBit": 16,
            "endian": "BE"
          }
        ]
    }
```

You can combine both together with comma in between, like this:

```
{
        "type": "Serial",
        "protocol": "RTU",
        "port": "COM5",
        "baudRate": 9600,
        "dataBits": 8,
        "stopBits": 1,
        "parity": "none",
        "Devices": [
          {
            "name": "Device 1",
```

```
            "deviceNum": 1,
            "address": 1,
            "dataType": "int",
            "mod": "",
            "registerLength": 2,
            "registerType": 3,
            "registerBit": 16,
            "endian": "BE"
          },
          {
            "name": "Device 2",
            "deviceNum": 3,
            "address": 3,
            "dataType": "int",
            "mod": "",
            "registerLength": 2,
            "registerType": 3,
            "registerBit": 16,
            "endian": "BE"
          }
        ]
    },
{
        "type": "TCP",
        "host": "127.0.0.1",
        "port": 502,
        "Devices": [
          {
            "name": "Device 4",
            "deviceNum": 1,
            "address": 1,
            "dataType": "int",
            "mod": "",
            "registerLength": 2,
            "registerType": 3,
            "registerBit": 16,
            "endian": "BE"
          },
          {
            "name": "Device 3",
            "deviceNum": 2,
            "address": 3,
            "dataType": "int",
            "mod": "",
            "registerLength": 2,
            "registerType": 3,
            "registerBit": 16,
            "endian": "BE"
          }
        ]
    }
```

Start the application

Just run the start.bat if using windows system, else run the start.sh for linux.

Devices Data

There will be a file called "deviceData.json" in the folder, it will be auto generated and keeping a list of the recorded devices energy record.

Sample :


```
{
 "Devices": [{
    "name": "Device1",
    "energy": 291,
    "Efficiency": 3215.4696132596687,
 }, {
    "name": "Device2",
    "energy": 300,
    "Efficiency": 3314.9171270718234,
 }, {
    "name": "Device3",
    "energy": 305,
    "Efficiency": 3370.1657458563536,
 }, {
    "name": "Device4",
    "energy": 310,
    "Efficiency": 3425.414364640884,
 }]
}
```


Please reset the energy of the device in this file if you are resetting the device's reading registers

---

Debugging

All output logs will be inserted into logs.log and all error log will be inserted into error.log.

Make an exe file if bat file can't run

1. In the case you can't run the start.bat, please install the exe named "advancedbattoexeconverter" provided in the folder.
2. After installation, right click the start.bat and select "compile with Advanced Bat to Exe", a start.exe will be created.
3. You should be able to start running the application by double click the start.exe

---

Update SDK version

This section is for updating of the BESC Client SDK, it is still work in progress.