This is an organised version of the previous project (AWS complete service). It makes use of the classes structures in node js.

There are three levels of scripts. The highest and the executed one is the task, i.e. TempSensortask and LEDtask. The middle level is the device, i.e. TempSensordevice and LEDdevice. Each script calls the lower one and inherits some functionalities from it. The Thing script is the lowest and is the one that all scripts end up using as every device is a Thing at the end.

More closely, the Thing is the general class that all the devices use, which contain the general functions (connecting to AWS, registering, delta-ing.. etc.). For instance, The TempSensordevice inherits all the functionalities of the Thing, and adds other functions that is only to be used in the Temperature Sensor case such as processing the Enocean telegram to extract the temperature. The TempSensortask does the tasks of the LED as a device (turns on and off).

One thing to note about this exact version of the service is that we extended the LED task from turning the LED on/off by the pi to sending data via Bluetooth to an HC-05 chip and hence to an Arduino to turn the LED on/off. You can go one step back by eliminating the Bluetooth part and decapsulate the onSwitchOn/ onSwitchOn functions out of the Bluetooth.connect in the following code:

bluetooth.connect('00:18:E5:03:67:DA', '1', function(err, connection){

MyLED.onSwitchOn(function(){

LedPin.writeSync(1);

connection.write(new Buffer('1', 'utf-8'), () => {

console.log("wrote");

});

console.log("LED is on");

})

MyLED.onSwitchOff(function(){

LedPin.writeSync(0);

connection.write(new Buffer('0', 'utf-8'), () => {

console.log("wrote");

});

console.log("LED is off");

})

});