AP Lab 14

Haseeb Ali Bhatti

112727

BSCS 5C

/\*

\* IoT Hub Raspberry Pi NodeJS - Microsoft Sample Code - Copyright (c) 2017 - Licensed MIT

\*/

const wpi = require('wiring-pi');

const Client = require('azure-iot-device').Client;

const Message = require('azure-iot-device').Message;

const Protocol = require('azure-iot-device-mqtt').Mqtt;

const BME280 = require('bme280-sensor');

const BME280\_OPTION = {

i2cBusNo: 1, // defaults to 1

i2cAddress: BME280.BME280\_DEFAULT\_I2C\_ADDRESS() // defaults to 0x77

};

const connectionString = '[Your IoT hub device connection string]';

const LEDPin = 4;

var sendingMessage = false;

var messageId = 0;

var client, sensor;

var blinkLEDTimeout = null;

function getMessage(cb) {

messageId++;

sensor.readSensorData()

.then(function (data) {

cb(JSON.stringify({

messageId: messageId,

deviceId: 'Raspberry Pi Web Client',

temperature: data.temperature\_C,

humidity: data.humidity

}), data.temperature\_C > 30);

})

.catch(function (err) {

console.error('Failed to read out sensor data: ' + err);

});

}

function sendMessage() {

if (!sendingMessage) { return; }

getMessage(function (content, temperatureAlert) {

var message = new Message(content);

message.properties.add('temperatureAlert', temperatureAlert.toString());

console.log('Sending message: ' + content);

client.sendEvent(message, function (err) {

if (err) {

console.error('Failed to send message to Azure IoT Hub');

} else {

blinkLED();

console.log('Message sent to Azure IoT Hub');

}

});

});

}

function onStart(request, response) {

console.log('Try to invoke method start(' + request.payload + ')');

sendingMessage = true;

response.send(200, 'Successully start sending message to cloud', function (err) {

if (err) {

console.error('[IoT hub Client] Failed sending a method response:\n' + err.message);

}

});

}

function onStop(request, response) {

console.log('Try to invoke method stop(' + request.payload + ')');

sendingMessage = false;

response.send(200, 'Successully stop sending message to cloud', function (err) {

if (err) {

console.error('[IoT hub Client] Failed sending a method response:\n' + err.message);

}

});

}

function receiveMessageCallback(msg) {

blinkLED();

var message = msg.getData().toString('utf-8');

client.complete(msg, function () {

console.log('Receive message: ' + message);

});

}

function blinkLED() {

// Light up LED for 500 ms

if(blinkLEDTimeout) {

clearTimeout(blinkLEDTimeout);

}

wpi.digitalWrite(LEDPin, 1);

blinkLEDTimeout = setTimeout(function () {

wpi.digitalWrite(LEDPin, 0);

}, 500);

}

// set up wiring

wpi.setup('wpi');

wpi.pinMode(LEDPin, wpi.OUTPUT);

sensor = new BME280(BME280\_OPTION);

sensor.init()

.then(function () {

sendingMessage = true;

})

.catch(function (err) {

console.error(err.message || err);

});

// create a client

client = Client.fromConnectionString(connectionString, Protocol);

client.open(function (err) {

if (err) {

console.error('[IoT hub Client] Connect error: ' + err.message);

return;

}

// set C2D and device method callback

client.onDeviceMethod('start', onStart);

client.onDeviceMethod('stop', onStop);

client.on('message', receiveMessageCallback);

setInterval(sendMessage, 2000);

});

GitHub link: https://github.com/haseeb21603/AP\_lab14