Code for Displaying Time and Temperature:

var lcd = new jsUpmI2cLcd.Jhd1313m1(6, 0x3E, 0x62);

var groveSensor = require('jsupm\_grove');

var today = setInterval(function ()

{

var d = new Date();

var b= d.toTimeString();

lcd.setColor(0, 255, 0);

// Go to the 2nd row, 6th character (0-indexed)

lcd.setCursor(0,0);

lcd.write(b);

var celsius = temp.value();

var fahrenheit = celsius \* 9.0/5.0 + 32.0;

var t = Math.round(fahrenheit);

lcd.setCursor(1, 1);

lcd.write(t+" \*F");

v.saveValue(t);

}, 1000);

Code for Sending SMS:

var twilio = require('twilio');

// Create a new REST API client to make authenticated requests against the

// twilio back end

var TWILIO\_ACCOUNT\_SID = '' ;

var TWILIO\_AUTH\_TOKEN = '';

var OUTGOING\_NUMBER = '';

var TWILIO\_NUMBER = '';

var client = new twilio.RestClient(TWILIO\_ACCOUNT\_SID, TWILIO\_AUTH\_TOKEN);

// Pass in parameters to the REST API using an object literal notation. The

// REST client will handle authentication and response serialzation for you.

client.sms.messages.create({

to:OUTGOING\_NUMBER,

from:TWILIO\_NUMBER,

body:'Hi, sending from my Edison SmartWatch'

}, function(error, message) {

// The HTTP request to Twilio will run asynchronously. This callback

// function will be called when a response is received from Twilio

// The "error" variable will contain error information, if any.

// If the request was successful, this value will be "falsy"

if (!error) {

// The second argument to the callback will contain the information

// sent back by Twilio for the request. In this case, it is the

// information about the text messsage you just sent:

console.log('Success! The SID for this SMS message is:');

console.log(message.sid);

console.log('Message sent on:');

console.log(message.dateCreated);

} else {

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

}

});

Code for Accelerometer and Gyroscope:

var accelrCompassSensor = require('jsupm\_lsm303');

// Instantiate LSM303 compass on I2C

var myAccelrCompass = new accelrCompassSensor.LSM303(0);

var successFail, coords, outputStr, accel;

var myInterval = setInterval(function()

{

// Load coordinates into LSM303 object

successFail = myAccelrCompass.getCoordinates();

// in XYZ order. The sensor returns XZY,

// but the driver compensates and makes it XYZ

coords = myAccelrCompass.getRawCoorData();

// Print out the X, Y, and Z coordinate data using two different methods

outputStr = "coor: rX " + coords.getitem(0)

+ " - rY " + coords.getitem(1)

+ " - rZ " + coords.getitem(2);

console.log(outputStr);

outputStr = "coor: gX " + myAccelrCompass.getCoorX()

+ " - gY " + myAccelrCompass.getCoorY()

+ " - gZ " + myAccelrCompass.getCoorZ();

console.log(outputStr);

// Get and print out the heading

console.log("heading: " + myAccelrCompass.getHeading());

// Get the acceleration

myAccelrCompass.getAcceleration();

accel = myAccelrCompass.getRawAccelData();

// Print out the X, Y, and Z acceleration data using two different methods

outputStr = "acc: rX " + accel.getitem(0)

+ " - rY " + accel.getitem(1)

+ " - Z " + accel.getitem(2);

console.log(outputStr);

outputStr = "acc: gX " + myAccelrCompass.getAccelX()

+ " - gY " + myAccelrCompass.getAccelY()

+ " - gZ " + myAccelrCompass.getAccelZ();

console.log(outputStr);

console.log(" ");

}, 1000);

// Print message when exiting

process.on('SIGINT', function()

{

clearInterval(myInterval);

myAccelrCompass = null;

accelrCompassSensor.cleanUp();

accelrCompassSensor = null;

console.log("Exiting");

process.exit(0);

});

Code for turning on the flashlight with button press:

var groveSensor = require('jsupm\_grove');

// Create the button object using GPIO pin 0

var button = new groveSensor.GroveButton(2);

// Read the input and print, waiting one second between readings

function readButtonValue() {

console.log(button.name() + " value is " + button.value());

var v=button.value();

if(v==1){ led.on();}

if(v==0){ led.off();}

}

setInterval(readButtonValue, 1000);

Sending data to Cloud:

var ubidots = require('ubidots');

var client = ubidots.createClient('YOUR-API-KEY');

client.auth(function () {

this.getDatasources(function (err, data) {

console.log(data.results);

});

var ds = this.getDatasource('xxxxxxxx');

ds.getVariables(function (err, data) {

console.log(data.results);

});

ds.getDetails(function (err, details) {

console.log(details);

});

var v = this.getVariable('xxxxxxx');

v.getDetails(function (err, details) {

console.log(details);

});

v.getValues(function (err, data) {

console.log(data.results);

});