Coding for cloud:

/\*

Sketch generated by the Arduino IoT Cloud Thing "Untitled"

https://create.arduino.cc/cloud/things/861e6ad4-a5ed-48bf-92ef-53214b4bc142

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

CloudAngle x;

CloudAngle y;

CloudAngle z;

CloudLight lED;

CloudLength dISTANCE;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions

which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.

\*/

#include <Adafruit\_MPU6050.h>

#include <Adafruit\_Sensor.h>

#include <Wire.h>

Adafruit\_MPU6050 mpu;

#include "thingProperties.h"

void setup() {

if (!mpu.begin()) {

Serial.println("Failed to find MPU6050 chip");

while (1) {

delay(10);

}

}

Serial.println("MPU6050 Found!");

// set accelerometer range to +-8G

mpu.setAccelerometerRange(MPU6050\_RANGE\_8\_G);

// set gyro range to +- 500 deg/s

mpu.setGyroRange(MPU6050\_RANGE\_500\_DEG);

// set filter bandwidth to 21 Hz

mpu.setFilterBandwidth(MPU6050\_BAND\_21\_HZ);

delay(100);

// Initialize serial and wait for port to open:

Serial.begin(9600);

// This delay gives the chance to wait for a Serial Monitor without blocking if none is found

delay(1500);

// Defined in thingProperties.h

initProperties();

// Connect to Arduino IoT Cloud

ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/\*

The following function allows you to obtain more information

related to the state of network and IoT Cloud connection and errors

the higher number the more granular information you’ll get.

The default is 0 (only errors).

Maximum is 4

\*/

setDebugMessageLevel(2);

ArduinoCloud.printDebugInfo();

}

void loop() {

ArduinoCloud.update();

// Your code here

/\* Get new sensor events with the readings \*/

/\* Get new sensor events with the readings \*/

sensors\_event\_t a, g, temp;

mpu.getEvent(&a, &g, &temp);

/\* Print out the readings \*/

Serial.print("Temperature: ");

Serial.print(temp.temperature);

Serial.println(" degC");

Serial.println("");

delay(1000);

}

/\*

Since LED is READ\_WRITE variable, onLEDChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onLEDChange() {

// Add your code here to act upon LED change

}

/\*

Since X is READ\_WRITE variable, onXChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onXChange() {

Serial.print("Rotation X: ");

Serial.print(g.gyro.x);

Serial.print("Acceleration X: ");

Serial.print(a.acceleration.x);

x=a.acceleration.x;

// Add your code here to act upon X change

}

/\*

Since Y is READ\_WRITE variable, onYChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onYChange() {

Serial.print(", Y: ");

Serial.print(a.acceleration.y);

Serial.print(", Y: ");

Serial.print(g.gyro.y);

y=a.acceleration.y;

// Add your code here to act upon Y change

}

/\*

Since Z is READ\_WRITE variable, onZChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onZChange() {

Serial.print(", Z: ");

Serial.print(a.acceleration.z);

Serial.println(" m/s^2");

z=a.acceleration.z;

// Add your code here to act upon Z change

}

/\*

Since DISTANCE is READ\_WRITE variable, onDISTANCEChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onDISTANCEChange() {

// Add your code here to act upon DISTANCE change

}