**Código fuente de: Programación para Arduino CubeSat**

**Lenguaje: Arduino**

**Fecha: 18/03/2018**

**Institución educativa: CBTis 168**

**Autor: Leonel Iván Fernández Carrillo**

static void smartdelay(unsigned long ms);

void TempHum\_satelite();

void Infrarojo\_satelite();

void Corriente\_satelite();

void gy\_87();

void gps\_satelite();

void Calidad\_aire();

void Rayos\_UV();

void FotoRes();

void Panel\_solar();

void BMP180\_satelite();

//GPS

#include <SoftwareSerial.h>

#include <TinyGPS.h>

TinyGPS gps;

int year;

byte month, day, hour, minute, second, hundredths;

unsigned long chars;

unsigned short sentences, failed\_checksum;

//Sensor de humedad y temperatura

#include <DHT.h>

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

//Sensor de corriente

float Sensibilidad=0.139;

float offset=0.100;

//Tarjeta SD

#include <SD.h>

File SDFILE;

//GY-87

#include "I2Cdev.h"

#include "MPU6050.h"

#include "HMC5883L.h"

static const char LED = 6;

static const float ACCEL\_SENS = 16384.0;

static const float GYRO\_SENS = 131.0;

HMC5883L mag;

int16\_t mx, my, mz;

MPU6050 accelgyro;

int16\_t ax, ay, az;

int16\_t gx, gy, gz;

#include <SFE\_BMP180.h>

SFE\_BMP180 bmp180;

double PresionNivelMar=1013.25;

void setup() {

Serial.begin(115200);

Serial1.begin(9600);

Serial2.begin(57600);

//temp hum

dht.begin();

Serial.print("Iniciando SD ...");

if (!SD.begin(4)) {

Serial.println("No se pudo inicializar");

return;

}

Serial.println("inicializacion exitosa");

//GY-87

boolean state = HIGH;

unsigned int count = 0;

while (!Serial && (count < 30) )

{

delay(200);

state = !state;

count++;

}

Wire.begin();

accelgyro.initialize();

Serial.print("Testing Accel/Gyro... ");

Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");

accelgyro.setI2CBypassEnabled(true); // set bypass mode

mag.initialize();

Serial.print("Testing Mag... ");

Serial.println(mag.testConnection() ? "HMC5883L connection successful" : "HMC5883L connection failed");

}

void loop() {

SDFILE = SD.open("Data.cvc", FILE\_WRITE);

if (SDFILE) {

} else {

}

gps\_satelite();

TempHum\_satelite();

Infrarojo\_satelite();

gy\_87();

BMP180\_satelite();

Calidad\_aire();

Rayos\_UV();

FotoRes();

Panel\_solar();

Corriente\_satelite();

SDFILE.close();

}

void gps\_satelite(){

smartdelay(500);

{

float latitude, longitude;

gps.f\_get\_position(&latitude, &longitude);

Serial.print("0;");

Serial.print(latitude,5);

Serial.print(",1;");

Serial.print(longitude,5);

Serial.print(",2;");

Serial.print(gps.f\_altitude());

Serial.print(",3;");

Serial2.print("0;");

Serial2.print(latitude,5);

Serial2.print(",1;");

Serial2.print(longitude,5);

Serial2.print(",2;");

Serial2.print(gps.f\_altitude());

Serial2.print(",3;");

SDFILE.print("0;");

SDFILE.print(latitude,5);

SDFILE.print(",1;");

SDFILE.print(longitude,5);

SDFILE.print(",2;");

SDFILE.print(gps.f\_altitude());

SDFILE.print(",3;");

gps.crack\_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths);

Serial.print(gps.f\_course());

Serial.print(",4;");

Serial.print(gps.f\_speed\_kmph());

Serial.print(",5;");

Serial.print(gps.satellites());

Serial.print(",6;");

Serial2.print(gps.f\_course());

Serial2.print(",4;");

Serial2.print(gps.f\_speed\_kmph());

Serial2.print(",5;");

Serial2.print(gps.satellites());

Serial2.print(",6;");

SDFILE.print(gps.f\_course());

SDFILE.print(",4;");

SDFILE.print(gps.f\_speed\_kmph());

SDFILE.print(",5;");

SDFILE.print(gps.satellites());

SDFILE.print(",6;");

gps.stats(&chars, &sentences, &failed\_checksum);

}

}

static void smartdelay(unsigned long ms){

unsigned long start = millis();

do

{

while (Serial1.available())

gps.encode(Serial1.read());

} while (millis() - start < ms);

}

void TempHum\_satelite(){

float h = dht.readHumidity();

float t = dht.readTemperature();

Serial.print(h);

Serial.print(",7;");

Serial.print(t);

Serial.print(",8;");

Serial2.print(h);

Serial2.print(",7;");

Serial2.print(t);

Serial2.print(",8;");

SDFILE.print(h);

SDFILE.print(",7;");

SDFILE.print(t);

SDFILE.print(",8;");

}

void Infrarojo\_satelite(){

int measure = analogRead(0);

Serial.print(measure);

Serial.print(",9;");

Serial2.print(measure);

Serial2.print(",9;");

SDFILE.print(measure);

SDFILE.print(",9;");

}

void Corriente\_satelite(){

float I=get\_corriente(200);

Serial.print(",22;");

Serial.print(I,3);

Serial.println();

Serial2.print(",22;");

Serial2.print(I,3);

Serial2.println();

SDFILE.print(",22;");

SDFILE.print(I,3);

SDFILE.println();

}

float get\_corriente(int n\_muestras){

float voltajeSensor;

float corriente=0;

for(int i=0;i<n\_muestras;i++)

{

voltajeSensor = analogRead(A1) \* (5.0 / 1023.0);

corriente=corriente+(voltajeSensor-2.5)/Sensibilidad;

}

corriente=corriente/n\_muestras;

return(corriente);

}

void Calidad\_aire(){

int aire = analogRead(2);

Serial.print(",20;");

Serial.print(aire);

Serial2.print(",20;");

Serial2.print(aire);

SDFILE.print(",20;");

SDFILE.print(aire);

}

void Rayos\_UV(){

Serial.print(",24;");

SDFILE.print(",24;");

Serial2.print(",24;");

int UV\_Val\_RAMBAL;

int UV;

UV\_Val\_RAMBAL = analogRead(3);

if(UV\_Val\_RAMBAL < 10) {

Serial2.print("0");

SDFILE.print("0");

Serial.print("0"); }

else { if(UV\_Val\_RAMBAL < 46) {

Serial2.print("1");

SDFILE.print("1");

Serial.print("1"); }

else { if(UV\_Val\_RAMBAL < 65) {

Serial2.print("2");

SDFILE.print("2");

Serial.print("2"); }

else { if(UV\_Val\_RAMBAL < 83) {

Serial2.print("3");

SDFILE.print("3");

Serial.print("3"); }

else { if(UV\_Val\_RAMBAL < 103) {

Serial2.print("4");

SDFILE.print("4");

Serial.print("4"); }

else { if(UV\_Val\_RAMBAL < 124) {

Serial2.print("5");

SDFILE.print("5");

Serial.print("5"); }

else { if(UV\_Val\_RAMBAL < 142) {

Serial2.print("6");

SDFILE.print("6");

Serial.print("6"); }

else { if(UV\_Val\_RAMBAL < 163) {

Serial2.print("7");

SDFILE.print("7");

Serial.print("7"); }

else { if(UV\_Val\_RAMBAL < 180) {

Serial2.print("8");

SDFILE.print("8");

Serial.print("8"); }

else { if(UV\_Val\_RAMBAL < 200) {

Serial2.print("9");

SDFILE.print("9");

Serial.print("9"); }

else { if(UV\_Val\_RAMBAL < 221) {

Serial2.print("10");

SDFILE.print("10");

Serial.print("10"); }

else { if(UV\_Val\_RAMBAL < 239) {

Serial2.print("11");

SDFILE.print("11");

Serial.print("11"); }

else {

Serial2.print("12");

SDFILE.print("12");

Serial.print("12"); }

}}}}}}}}}}}

}

void FotoRes(){

int fres = analogRead(4);

Serial.print(",21;");

Serial.print(fres);

Serial2.print(",21;");

Serial2.print(fres);

SDFILE.print(",21;");

SDFILE.print(fres);

}

void Panel\_solar(){

int solar = analogRead(5);

Serial.print(",22;");

Serial.print(solar);

Serial2.print(",22;");

Serial2.print(solar);

SDFILE.print(",22;");

SDFILE.print(solar);

}

void gy\_87(){

static unsigned long ms = 0;

static boolean state = HIGH;

if (millis() - ms > 100)

{

accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

Serial.print(ax/ACCEL\_SENS);

Serial.print(",10;");

Serial.print(ay/ACCEL\_SENS);

Serial.print(",11;");

Serial.print(az/ACCEL\_SENS);

Serial.print(",12;");

Serial.print(gx/GYRO\_SENS);

Serial.print(",13;");

Serial.print(gy/GYRO\_SENS);

Serial.print(",14;");

Serial.print(gz/GYRO\_SENS);

Serial.print(",15;");

SDFILE.print(ax/ACCEL\_SENS);

SDFILE.print(",10;");

SDFILE.print(ay/ACCEL\_SENS);

SDFILE.print(",11;");

SDFILE.print(az/ACCEL\_SENS);

SDFILE.print(",12;");

SDFILE.print(gx/GYRO\_SENS);

SDFILE.print(",13;");

SDFILE.print(gy/GYRO\_SENS);

SDFILE.print(",14;");

SDFILE.print(gz/GYRO\_SENS);

SDFILE.print(",15;");

Serial2.print(ax/ACCEL\_SENS);

Serial2.print(",10;");

Serial2.print(ay/ACCEL\_SENS);

Serial2.print(",11;");

Serial2.print(az/ACCEL\_SENS);

Serial2.print(",12;");

Serial2.print(gx/GYRO\_SENS);

Serial2.print(",13;");

Serial2.print(gy/GYRO\_SENS);

Serial2.print(",14;");

Serial2.print(gz/GYRO\_SENS);

Serial2.print(",15;");

mag.getHeading(&mx, &my, &mz);

Serial.print(mx);

Serial.print(",16;");

Serial.print(my);

Serial.print(",17;");

Serial.print(mz);

Serial.print(",18;");

SDFILE.print(mx);

SDFILE.print(",16;");

SDFILE.print(my);

SDFILE.print(",17;");

SDFILE.print(mz);

SDFILE.print(",18;");

Serial2.print(mx);

Serial2.print(",16;");

Serial2.print(my);

Serial2.print(",17;");

Serial2.print(mz);

Serial2.print(",18;");

float heading = atan2(my, mx);

}

}

void BMP180\_satelite(){

Serial.print(",18;");

Serial2.print(",18;");

SDFILE.print(",18;");

char status;

double T,P,A;

status = bmp180.startTemperature();

if (status != 0)

{

delay(status);

status = bmp180.getTemperature(T);

if (status != 0)

{

status = bmp180.startPressure(3);

if (status != 0)

{

delay(status);

status = bmp180.getPressure(P,T);

if (status != 0)

{

Serial.print(T);

Serial.print(",19;");

Serial.print(P);

A= bmp180.altitude(P,PresionNivelMar);

Serial.print(",23;");

Serial.print(A);

Serial2.print(T);

Serial2.print(",19;");

Serial2.print(P);

A= bmp180.altitude(P,PresionNivelMar);

Serial2.print(",23;");

Serial2.print(A);

SDFILE.print(T);

SDFILE.print(",19;");

SDFILE.print(P);

A= bmp180.altitude(P,PresionNivelMar);

SDFILE.print(",23;");

SDFILE.print(A);

}

}

}

}

}