

```

#include <SD.h>
#include <MsTimer2.h>

int val;
int DATA = 3;
int CLK = 2;
float encoderPos = -1;
int encoderPinALast = LOW;
int n = LOW;
int t=5000;
float rpm=0;
float rev=0;

int chipSelect = 4; //Variable de Mirdo SD Card
File file; // Archivo para leer y escribir

void setup() {

    pinMode (DATA, INPUT);
    pinMode (CLK, INPUT);

    Serial.begin(9600);
    pinMode(chipSelect, OUTPUT);

    if (!SD.begin(chipSelect)) { // Para inicializar la tarjeta SD
        Serial.println("Tarjeta SD no pudo ser inicializada"); // Si no se ha insertado la tarjeta o no se inició
    }
    if (SD.exists("file.xls")) { // Si el documento existe se borrará
        Serial.println("El documento ya existe");
        if (SD.remove("file.xls") == true) {
            Serial.println("El Archivo se borró correctamente");
        } else {
            Serial.println("No se pudo borrar el archivo");
        }
    }
}

```

```

MsTimer2::set(t, guardar_rpm); // Función que se repite cada 5s
MsTimer2::start();
}

void guardar_rpm() {
    Serial.print (encoderPos);
    Serial.print ("/");
    rpm=( encoderPos /20 ) * 12;
    encoderPos=0;
    Serial.println (rpm);
    file = SD.open("file.xls", FILE_WRITE); // abre el documento para escribir en el
    if (file) {
        if (rpm!=0)
        {
            file.println(rpm); // escribir rpm en el documento
        }
        file.close(); // para cerrar el archivo
    } else {
        Serial.println("No se puede escribir en el documento");
    }
}

void loop() {
    n = digitalRead(DATA);
    if ((encoderPinALast == LOW) && (n == HIGH)) {
        if (digitalRead(CLK) == LOW) {
            //encoderPos--;
        } else {
            encoderPos++;
        }
    }
    encoderPinALast = n;
}

```