

## DEVELOPING A LOW COST DATALOGGER WITH ARDUINO.

Jiménez Buendía, M.<sup>1</sup>; Guerrero Arroniz, D.<sup>1</sup>; Ruiz Peñalver, L.<sup>1</sup>; Wesierski, P.<sup>2</sup>;  
Szczombrowski, A.<sup>2</sup>; Molina Martínez, J.M.<sup>1</sup>

<sup>1</sup> Universidad Politécnica de Cartagena, <sup>2</sup> Poznan University of Technology

A datalogger is a device capable of detecting and measuring electrical signals of various types (analog and digital), recording them and automatically store them in real time in a memory designated to this. In the market there are numerous brands and types of most versatile and robust dataloggers, however the great disadvantage of them is high cost, and in some cases programming difficulty.

That's why, we designed a datalogger based on Arduino platform. We created useful and flexible device, able to execute similar tasks, like other devices available on the market, but in lower cost. So our datalogger gives all input( analog /digital) as ATmega 328, which can be set by user, depending on chosen mode and automatically save on SD card.

**Keywords:** *Datalogger; Arduino; SD Memory*

## DESARROLLO DE UN DATALOGGER DE BAJO COSTE CON ARDUINO

Un datalogger es un dispositivo capaz de detectar y medir señales eléctricas de diversos tipos (tanto analógicas como digitales) para registrarlas y almacenarlas en tiempo real y de forma automática en una memoria destinada para tal fin. En el mercado existen numerosas marcas y tipos, la mayoría de gran versatilidad y robustez, sin embargo, el gran inconveniente que presentan es su elevado coste y en algunos casos, dificultad de programación.

Por ello, se ha desarrollado un datalogger con la plataforma Arduino con el fin de crear un dispositivo útil y flexible capaz de desarrollar funciones similares a las de otras soluciones del mercado pero con un coste considerablemente menor. Así, el datalogger con arduino aprovecha todas las entradas analógicas y digitales que posee el ATmega 328 (Microcontrolador de la placa Arduino) para ser configuradas de manera independiente midiendo las señales de diversos sensores que conectemos y registrándolas automáticamente en una memoria SD con el fin de poder trabajar y compartir los datos obtenidos de una forma sencilla.

**Palabras clave:** *Datalogger; Arduino; Memoria SD*

## 1. Introducción

La medición y adquisición de datos a través de sensores es una técnica cada vez más utilizada y necesaria en el ámbito del control de procesos o de la eficiencia energética. La tendencia actual es realizar estas medidas de forma automática y registrarlas en tiempo real a través de un sistema SCADA o con un dispositivo creado para tal fin. Uno de los usos más útiles y demandados es el de registrar las medidas de un determinado sistema periódicamente y de forma continuada para estudiar el comportamiento de dicho sistema a lo largo del tiempo y poder obtener datos reales acerca de su funcionamiento.

Para la tarea de recogida de datos automática, suelen utilizarse unos dispositivos denominados *dataloggers*, de reducido tamaño y gran versatilidad, capaces de realizar lecturas de sensores a través de sus numerosos puertos analógicos y digitales para posteriormente acondicionarlas al tipo de lectura que se está realizando (tensión, temperatura, humedad, resistencia, presión...) Existen numerosos modelos en el mercado capaces de realizar estas tareas como el DT80 de *Datataker* o el CR1000 de *Campbell*, mostrados en la Figura 1, con los que se ha trabajado y que han servido de base de pruebas y conocimientos para la realización de nuestro propio datalogger.

La necesidad de crear un dispositivo diferente a los que el mercado ofrece se debe a diversos motivos:

- **Coste.** La mayoría de los modelos que el mercado ofrece son de un coste relativamente elevado.
- **Utilidad.** Muchas opciones de medida que poseen dichos modelos no se usan generalmente, por lo que se quiere desarrollar un dispositivo versátil y sobre todo, que se adapte a las necesidades de medición.
- **Tamaño.** Se busca un dispositivo portable y de reducido tamaño, ideal para su implementación en cualquier sistema.
- **Consumo.** Dicho dispositivo ha de tener un consumo contenido y a ser posible que se alimente tanto con una batería interna como de la red eléctrica.
- **Programación sencilla.** Se busca que sea fácil de programar, configurar y que la tarea de obtención de datos sea sencilla.

Figura 1: Datalogger DT80 y CR1000



## 2. Objetivos

- Diseñar un datalogger a través de un microcontrolador ATmega programado por Arduino de la forma más sencilla posible que permita cumplir los requisitos anteriormente expuestos.
- Programar un código base que permita realizar las medidas, registrarlas en intervalos de tiempo y que pueda ser modificado y mejorado en el futuro.
- Crear una interfaz gráfica para la configuración del datalogger y la obtención de sus datos.
- Almacenar las medidas en una tarjeta de memoria extraíble (tipo SD) con el fin de que dichos datos puedan almacenarse y transportarse de forma sencilla y rápida.
- Sentar las bases e investigaciones necesarias para poder producir un dispositivo finalizado e introducirlo en el mercado.

## 3. Metodología y Materiales

### 3.1 Arduino

Arduino es una plataforma basada en una simple placa de circuito impreso y un entorno de desarrollo propio (IDE), a partir de la cual se pueden crear diversos dispositivos y desarrollar diferentes aplicaciones. Dispone de estructura de programación intuitiva y es una plataforma libre, por lo que existen una gran variedad de librerías destinadas a usos en concreto.

Además, otra de las consideraciones principales de elección de esta plataforma frente a los PIC reside en el importante hecho de que la propia placa Arduino permite reprogramar el microcontrolador Atmel que integra. Todo esto de una forma rápida y sencilla mediante la conexión a un ordenador por puerto USB.

También es importante la gran variedad de *shields* o placas de complementos existentes para esta plataforma en lo referido a la comunicación. En nuestro caso, utilizaremos un shield muy útil y extendido: el de la memoria SD.

### 3.2 Arduino Uno

#### Descripción General

Arduino UNO es una placa microcontrolador basada en el ATmeg328. Tiene 14 entradas/salidas digitales (de las cuales 6 proporcionan salida PWM), 16 entradas digitales, un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reseteo. Contiene todo lo necesario para hacer funcionar el microcontrolador; simplemente conectándolo al ordenador por el cable USB o alimentándolo con un transformador o batería para empezar.

**Figura 2: Arduino UNO**



### Características

- Microcontrolador: Atmel ATmega328
- Tensión de Operación (nivel lógico): 5 V
- Tensión de Entrada (recomendado): 7-12 V
- Tensión de Entrada (límites): 6-20 V
- Pines E/S Digitales: 14 (6 pueden usarse como salidas PWM)
- Entradas Analógicas: 6
- Corriente máx por cada PIN de E/S: 40 mA
- Memoria Flash: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Frecuencia de reloj: 16 MHz
- Dimensiones: 68,6mm x 53,3mm

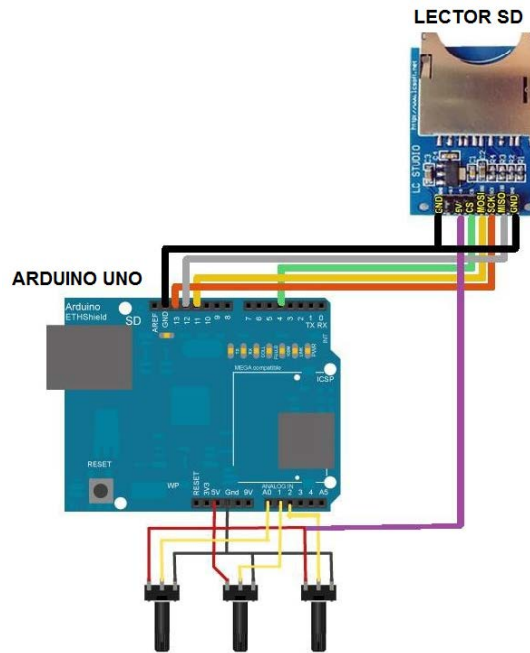
## **4. Resultados**

### **4.1 Implementación del hardware**

Al tratarse de un prototipo, se ha empleado la propia placa de entrenamiento de Arduino junto con una placa protoboard para las pruebas. Sin embargo, se está trabajando actualmente en la elaboración de una placa única diseñada para albergar el ATmega y las conexiones necesarias para el datalogger.

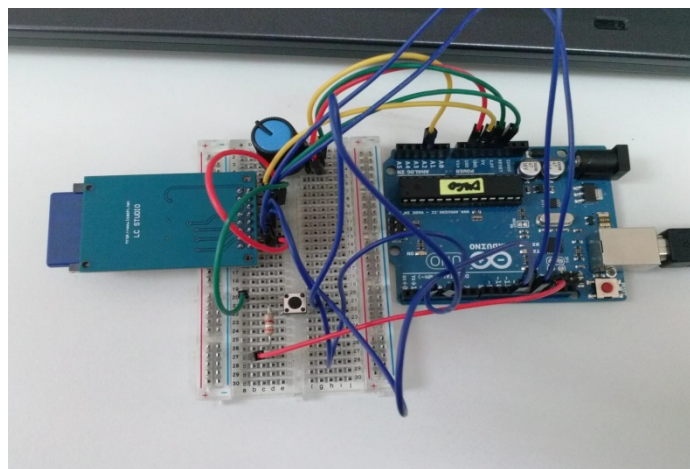
En el siguiente esquema se puede observar una posible conexión de sensores (en este caso, potenciómetros analógicos) a la placa arduino, cuyas señales posteriormente son procesadas y almacenadas en la memoria SD. Tal y como se ha dicho al principio, nuestro datalogger es plenamente configurable y capaz de leer señales analógicas y digitales, por lo que este es solo una de las muchas configuraciones que podemos realizar.

**Figura 3: Esquema con 3 entradas analógicas**



En la siguiente figura se puede ver una imagen real del prototipo. En este caso se han conectado un pulsador (señal digital) y un potenciómetro (señal analógica).

**Figura 4: Circuito con entrada analógica y digital**



En cuanto a las conexiones necesarias para el lector o *shield* SD, encontramos:

Figura 6: Conexiones lector SD



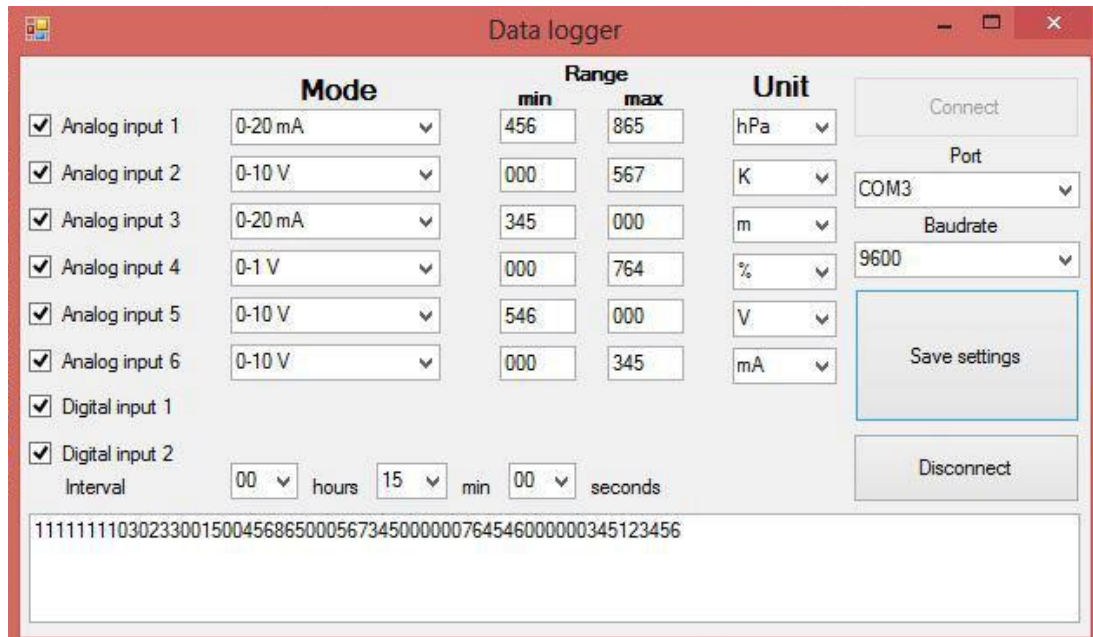
- GND Masa del lector
- 5V Alimentación del lector. También puede alimentarse a 3.3V
- CS *Chip Select*. Para seleccionar un esclavo en el caso de disponer de varios dispositivos conectados a la vez.
- MOSI/MISO Entrada y salida de datos del puerto serie.
- SCK *Source Clock*. Entrada del reloj para sincronizar el dispositivo.

#### 4.2 Características del código programado

Con el código diseñado a través de la plataforma Arduino se genera un archivo de texto que contiene los parámetros del tipo de lectura que se va a realizar. Dichos parámetros se configuran a través de una interfaz gráfica diseñada mediante Microsoft Visual Basic. Se pueden configurar los siguientes:

- Selección de hasta 8 lecturas diferentes (6 analógicas y 2 digitales) de forma simultánea.
- Modo de lectura: Para medir tensión (de 0-1 V o de 0-10 V) o para medir intensidad (0-20mA o 4-20mA).
- Establecimiento del rango de medida del sensor a leer mediante sus valores mínimos y máximos.
- Conversión de unidades. Se puede elegir la unidad de medida de los sensores para convertir directamente y obtener los valores de dicha unidad en vez de los valores de tensión o intensidad leídos por el sensor.
- Selección del puerto de comunicaciones y la velocidad de transferencia.
- Determinación del intervalo de almacenamiento de valores en horas, minutos y segundos.

Figura 7: Interfaz de configuración



Una vez configurados los parámetros que determinan la lectura a medir, se almacena un archivo .txt que se almacena en la memoria SD del Arduino. De esta manera, cuando conectamos nuestro sensor y comenzamos a medir, el dispositivo lee en primer lugar el archivo de opciones y se configura internamente para comenzar las lecturas.

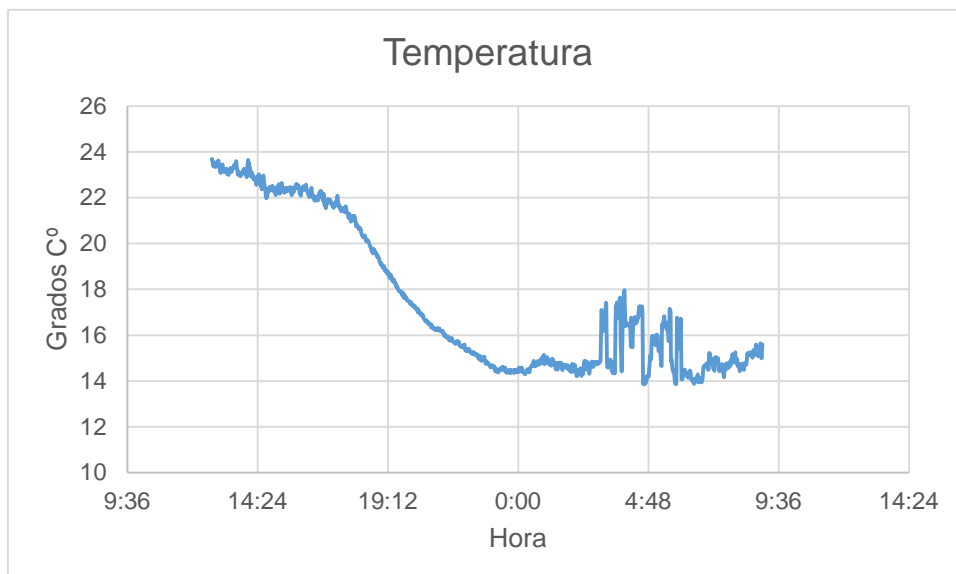
### 4.3 Validación en campo

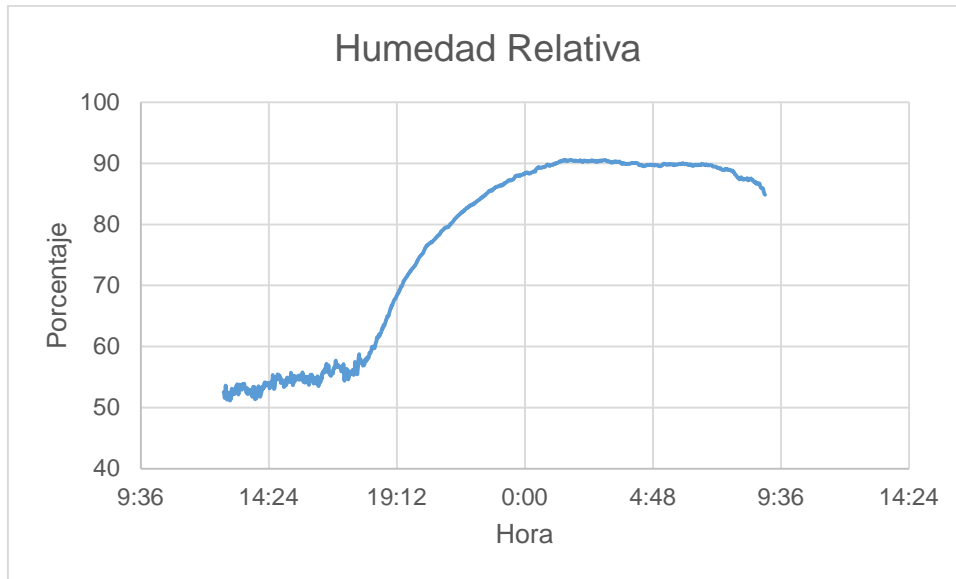
Para su validación se han utilizado los siguientes sensores de una estación agroclimática que recoge sus datos mediante un CR1000. El prototipo se ha puesto en paralelo con ella obteniéndose los siguientes resultados:

- Sensor de temperatura y Humedad relativa (HMP45-ACT)

Rango de salida = 0.008-1 (V)

Rango de medida = -40 a +60 (°C) para temperatura. 0 a 100 (%) para humedad.





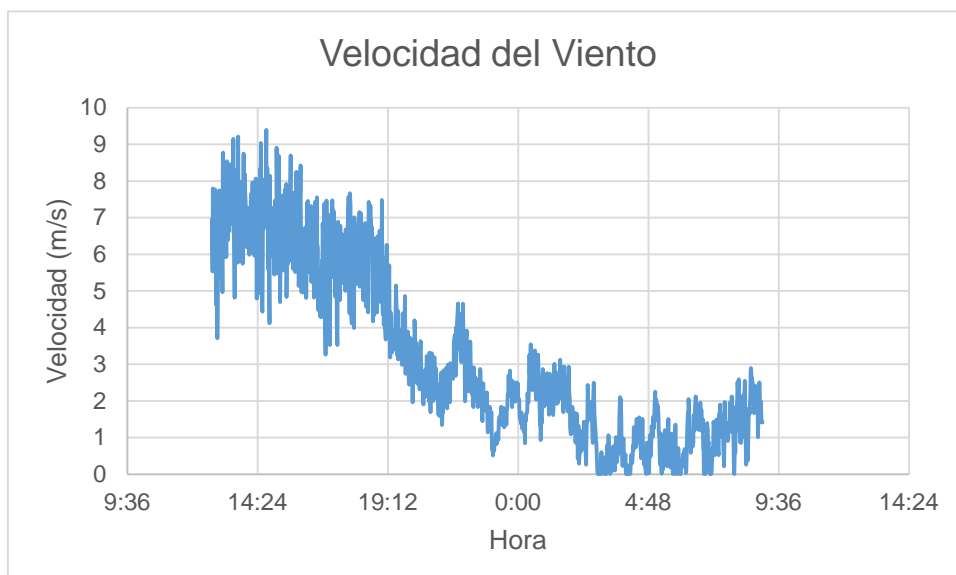
• Anemoveleta

Rango de salida (Dirección del viento) = Potenciómetro 0-12 (V)

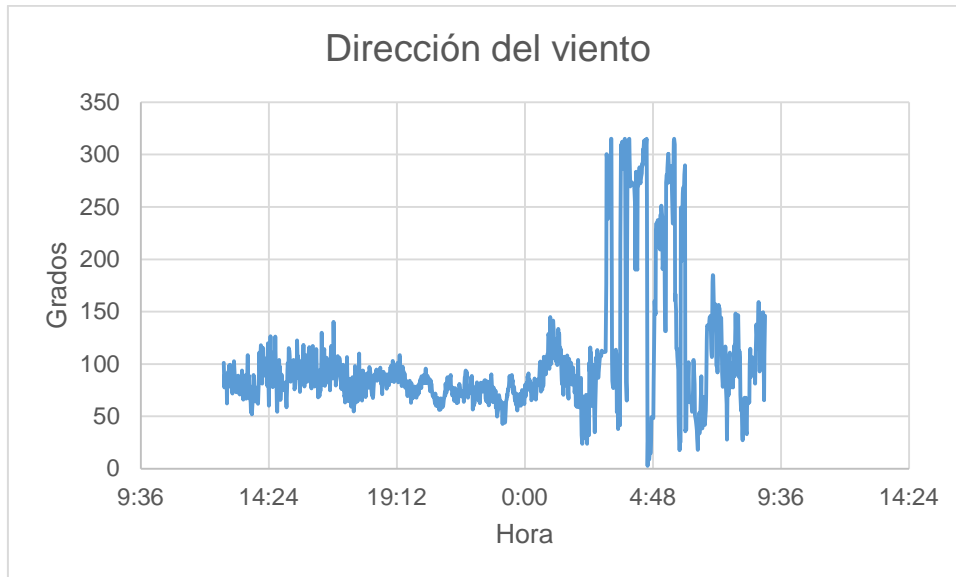
Rango de medida = 0 a 355 (°)

Rango de salida (Velocidad del viento) = La rotación de la hélice de 4 palas produce una señal eléctrica en corriente alterna, de frecuencia proporcional a la velocidad del viento. Esta señal es inducida en una bobina estacionaria por el giro de un imán permanente de 6 polos, solidario al eje de las hélices.

Rango de medida = 0 a 20 m/s



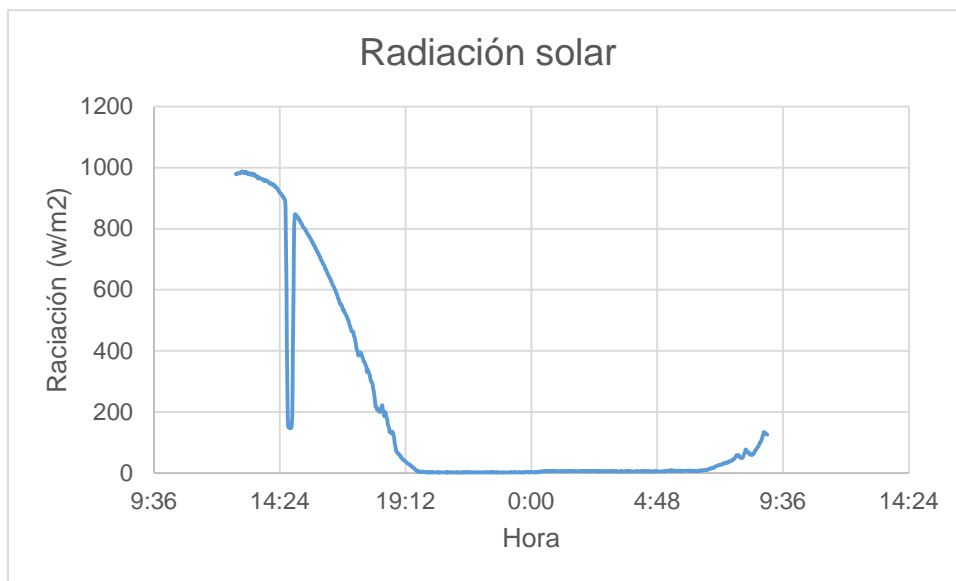




• Piranómetro (CMP-3)

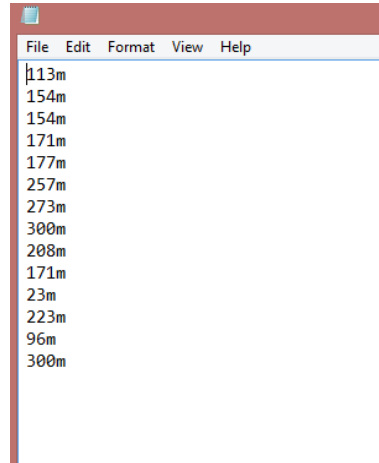
Rango de salida = 0-20 (mv)

Rango de medida = 0 a 1000W/m<sup>2</sup>



Las medidas se van registrando en la memoria SD a través de un archivo de texto que las almacena según el intervalo de tiempo anteriormente establecido.

**Figura 8: Ejemplo de resultados**



Nota: Simulación de lecturas de un sensor de proximidad que mide distancias en metros, cuyo rango oscila entre los 23 y los 230 metros.

Todas las medidas se registran en columnas separadas por comas, por lo que es muy sencillo exportar los datos a una hoja de datos excel para obtener de ellas más cálculos y procesar los datos gráficamente.

## 5. Conclusiones

Se confirman las hipótesis de partida, verificando el coste bajo de los componentes utilizados, la versatilidad de las medidas que se pueden realizar, el consumo bajo, se alimenta el sistema por el puerto usb del ordenador, la facilidad de programación utilizando Visual Basic y C simplificado, sin punteros.

El desarrollo futuro pasará por la sustitución de la placa de prototipo por placas de circuito impreso diseñadas para los periféricos a utilizar.