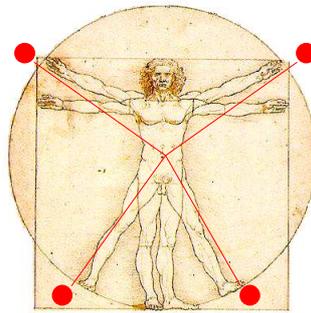


TECNOLOGÍ@ y DESARROLLO

Revista de Ciencia, Tecnología y Medio Ambiente

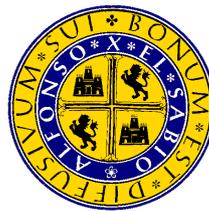
VOLUMEN VII. AÑO 2010

SEPARATA



ARDUDROP 1.0: DISPOSITIVO ELECTRÓNICO
PARA EL ESTUDIO DE LA HUMEDAD DEL SUELO

Miguel A. de Pablo Hdez. y C. de Pablo S.



UNIVERSIDAD ALFONSO X EL SABIO
Escuela Politécnica Superior
Villanueva de la Cañada (Madrid)

© Del texto: Miguel Ángel de Pablo Hernández y C. de Pablo S.
Noviembre,2010.

http://www.uax.es/publicaciones/archivos/TECMAD10_006.pdf

© De la edición: *Revista Tecnol@y desarrollo*

Escuela Politécnica Superior.

Universidad Alfonso X el Sabio.

28691, Villanueva de la Cañada (Madrid).

ISSN: 1696-8085

No está permitida la reproducción total o parcial de este artículo, ni su almacenamiento o transmisión ya sea electrónico, químico, mecánico, por fotocopia u otros métodos, sin permiso previo por escrito de la revista.

Tecnol@y desarrollo. ISSN 1696-8085. Vol.VII. 2010.

ARDUDROP 1.0: DISPOSITIVO ELECTRÓNICO PARA EL ESTUDIO DE LA HUMEDAD DEL SUELO

Miguel A. de Pablo Hdez.¹ y C. de Pablo S.

(1) Departamento de Geología. Universidad de Alcalá. Edificio de Ciencias. Campus Externo. Ctra. A-II km33,600. 28871 Alcalá de Henares, Madrid. España.
miguelangel.depablo@uah.es

RESUMEN:

En este trabajo se presenta un dispositivo electrónico para la adquisición de datos, mediante diferentes sensores, para el estudio de la evolución de la humedad en el suelo y su relación con parámetros ambientales como temperatura ambiental, temperatura al sol, humedad relativa e intensidad de la lluvia. Este dispositivo, en su versión 1.0, ha sido desarrollado mediante el uso de la plataforma Arduino (open-hardware) y el entorno de programación, en lenguaje C, Arduino v17. El código que lo controla también se presenta en este trabajo. Para el desarrollo del dispositivo ArduDrop 1.0 se han empleado sensores comerciales, pero se han desarrollado dos sensores (humedad del suelo y disdrómetro) propios basados en trabajos anteriores de otros autores. Este dispositivo es fácilmente clonable y mejorable sin necesidad de importantes conocimientos de electrónica ni programación, lo que junto con el reducido precio de los componentes y software, hace que sea de gran interés, tanto en el campo científico como en el didáctico.

PALABRAS CLAVE: *Agua, Temperatura, Humedad, Sensores, Open-hardware, Arduino.*

ABSTRACT

In this paper, an electronic device, whose purpose is the acquisition of data through various sensors is presented. The data (soil moisture, air temperature, relative humidity, and rain intensity) allow study the evolution of soil moisture depending of the environmental conditions. This device, named such as ArduDrop, at its 1.0 version here presented, was developed by the use of open-hardware: Arduino, and its programming environment in C language (Arduino v17). The complete firmware is also presented in this paper adequately commented. To develop the device ArduDrop 1.0, commercial sensors have been used, but other two (soil moisture and disdrometer) were developed for this device based on earlier works by other authors. ArduDrop is easily reproducible and improved without significant knowledge of electronics or programming, which together with the low price of components and software, makes it of great interest for both scientific and teaching applications.

KEY-WORDS: *Water, Temperature, Humidity, Sensors, Open-hardware, Arduino.*

SUMARIO: 1. Introducción, 2. Fundamento teórico, 3. Descripción del dispositivo, 4. Hardware, 5. Firmware, 6. Salida de datos, 7. Aplicaciones, 8. Mejoras, 9. Conclusiones generales, 10. Agradecimientos, 11. Referencias, 12. Direcciones web.

http://www.uax.es/publicaciones/archivos/TECMAD10_006.pdf

1. Introducción

El agua es tal vez el más importante de los recursos naturales, pues de él dependen la subsistencia de la población, la flora y la fauna, y la continuidad de la industria. En el actual estado del planeta, y con las presentes condiciones climáticas a escala global, este recurso y su gestión es causa de conflictos a escala local, regional, nacional e internacional. La protección del agua y su adecuada gestión y consumo es el objetivo prioritario de organismos internacionales, gobiernos nacionales y de organizaciones no gubernamentales. Muchos de los esfuerzos de estos organismos están dirigidos hacia el desarrollo de políticas de ahorro de agua, consumo responsable y formación adecuada a distintos niveles educativos. Un ejemplo de las políticas dedicada al ahorro del agua es el incentivo del uso de riego por goteo, tanto a nivel particular, como a nivel industrial. Pero en muchas ocasiones este tipo de técnicas se ven rechazadas debido al desconocimiento de sus beneficios, aunque en ocasiones también al alto coste de los dispositivos necesarios. Esto muestra cómo es necesaria una adecuada formación de la población y especialmente de los jóvenes. Los currículos educativos de Educación Secundaria en España recogen distintos aspectos relacionados con el agua, su gestión y consumo responsable. Pero también es necesario el desarrollo de tecnologías baratas que permitan a la población implicarse en el ahorro del agua en distintos ambientes (en el hogar, en la industria, etc.). Sólo las medidas sencillas y baratas serán adoptadas por una mayor cantidad de población.

La necesidad de ampliar las medidas para el uso responsable del agua, la necesidad de reflejar la necesidad del uso responsable del agua en la formación de la población, y la necesidad de desarrollar dispositivos económicos y sencillos, nos han llevado a desarrollar un dispositivo electrónico de bajo coste para el estudio de la humedad del suelo y su relación con las condiciones ambientales. Este dispositivo basado en el uso de hardware libre, y al que hemos denominado ArduDrop, no es más que una serie de sensores, gestionados por un microcontrolador, que adquieren datos que son almacenados en una tarjeta de memoria y que pueden posteriormente ser manejados para establecer un adecuado riego del terreno, entre otras aplicaciones. Lo que se presenta en este trabajo es un diseño inicial cuya finalidad no es el desarrollo de un dispositivo listo para su comercialización, sino una herramienta para ser usado con objetivos didácticos en muy diversos entornos y niveles, desde la educación secundaria hasta la universidad, o en asociaciones de distinto carácter. Pueden ser empleadas por particulares, agricultores, jardineros, estudiantes, profesores, etc. y su filosofía es que pueda ser mejorado, modificado o adaptado de acuerdo a las necesidades y objetivos del usuario, y cuya sencillez y atractivo permita su uso extendido en distintos entornos, y la concienciación en la conservación del agua como recurso natural.

Los autores queremos hacer notar desde el primer momento que somos conscientes de que este diseño inicial de ArduDrop 1.0 no es electrónicamente perfecto, y que puede ser mejorado notablemente, y que es posible realizar diversas modificaciones para que tenga mayores aplicaciones. Pero el objetivo de crear un dispositivo sencillo queda patente cuando los que hemos desarrollado ArduDrop no tenemos formación específica en electrónica o programación en lenguaje C, lo que da idea de las grandes posibilidades que tiene ArduDrop para que pueda ser mejorado y adaptado por especialistas, para que pueda ser empleado en el campo educativo de la electrónica para focalizar la atención de los estudiantes de ingenierías en la necesidad de desarrollar dispositivos enfocados en el adecuado uso y consumo del agua, y para ser empleado en los centros culturales, educativos, Educación Secundaria y Universitaria en el desarrollo de actividades relacionadas con el agua.

Así, el objetivo del presente trabajo es mostrar el primer prototipo en funcionamiento de ArduDrop, un dispositivo electrónico de bajo coste basado en el uso de hardware libre para la adquisición y almacenamiento de datos de humedad del suelo y otros parámetros relacionados con las condiciones meteorológicas locales. Para ello en las siguientes páginas se describe el dispositivo, se muestran las características de los distintos componentes empleados en su desarrollo, se muestra y describe el código programado en el microcontrolador del dispositivo, se muestran algunos datos recogidos con el mismo, y se discuten diversas aplicaciones, así como posibles mejoras o modificaciones que permitan a este dispositivo ser más eficaz y con mayores aplicaciones.

2. Fundamento teórico

La abundancia de agua en un territorio depende de las condiciones climáticas reinantes, pero su permanencia en él depende de diversos factores como son: las características del propio terreno: textura, estructura, permeabilidad, composición química, grado de alteración del terreno, etc., las características topográficas: pendiente, orientación, etc., y la presencia de vegetación: tipo, desarrollo, etc. Todos estos factores condicionan el balance de agua en el terreno (ej., Thornthwaite, 1948; Palmer and Havens, 1958; Pereira and Pruitt, 2004) de manera que se cumple la siguiente expresión:

$$\text{Entradas} = \text{Salidas} \pm \text{Variación del agua almacenada en el terreno}$$

La cantidad de agua que puede almacenar el terreno es lo que se denomina Capacidad de Campo, y es una propiedad intrínseca de cada tipo de terreno dependiente de sus características (textura, estructura, composición, permeabilidad, etc.). Las entradas en el

terreno vienen marcadas por la precipitación (lluvia, nieve, etc.), mientras que las salidas están constituidas por la evapotranspiración y la escorrentía (superficial y subterránea). La evapotranspiración es el agua evaporada a través de la vegetación existente en el terreno. Conocer con detalle este balance hídrico de una zona permitiría poder hacer un uso adecuado de los recursos hídricos, planificar el riego, mejorar las cosechas, adecuar las especies vegetales en un plan de reforestación, etc. Estos balances hídricos se realizan a partir de datos termopluviométricos tomados en estaciones meteorológicas, estaciones de aforos y estaciones agronómicas, de las que se obtienen datos de precipitación, escorrentía, y evapotranspiración, respectivamente.

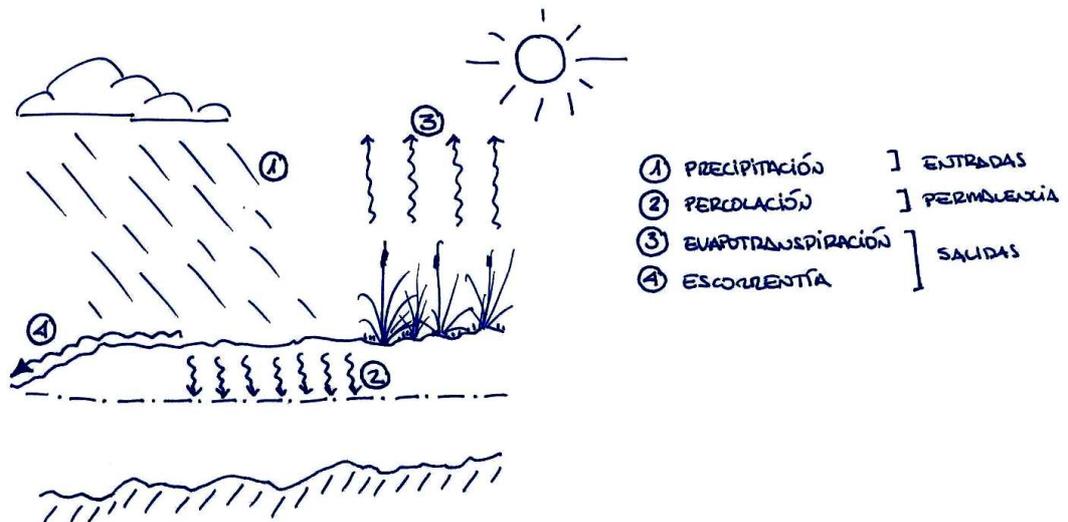


Figura 1: Esquema del balance hídrico en el terreno. (Fuente: elaboración propia)

Todos estos datos son empleados para realizar balances hídricos en extensas áreas, pero pueden ser ajustados mediante el uso de datos locales, que con una serie temporal de datos suficientemente larga, puede ayudar a perfeccionar los resultados haciendo los mismos más representativos del emplazamiento específico. Los datos termopluviométricos son sencillos de obtener, lo contrario de los datos de escorrentía y evapotranspiración, que requieren instrumentación más compleja o que debe ser instalada en emplazamientos que a veces no se encuentran cercanos al terreno de estudio. De hecho, muchos de los datos de evapotranspiración que se emplean en los balances hídricos son empíricos, pero las aproximaciones que se obtienen son suficientes para el entendimiento del ciclo del agua en una cuenca. La obtención de datos termopluviométricos de forma sencilla y económica puede ayudar a extender la red de este tipo de datos y a la consiguiente mejora de los balances hídricos y del uso del agua en cada terreno en función de sus características.

ArduDrop está diseñado para adquirir datos de la humedad del suelo a 3 cm de profundidad, así como otros datos sobre las condiciones ambientales. Para ello se cuenta con un sensor de temperatura y humedad relativa del aire a 22 cm de altura sobre el suelo con el fin de conocer las condiciones climáticas a las que el terreno se ve sometido. Por otro lado, para estimar la relación entre la humedad del suelo y la evaporación, se cuenta con un sensor de temperatura del aire al sol, y un sensor de luminosidad. Y finalmente, para conocer de forma aproximada las precipitaciones, se cuenta con un disdrómetro experimental que, aunque no permite derivar de forma precisa la precipitación en mm/h debido a la complejidad de estas estimaciones a través de disdrómetros (ej., Smith et al., 1993; Smith and De Veaux, 1994; Kourtellis et al., 2005, Licznar et al., 2008), sí permite un acercamiento inicial a la cantidad de agua que entra en el terreno por este medio.

Todos estos datos son, en definitiva, la base de los balances hídricos y, aunque no se pueden realizar cálculos numéricos de precisión, sí son un primer acercamiento al comportamiento del agua en un terreno determinado con todas sus características. Por tanto, el estudio de la evolución de los datos facilitados por estos sensores puede permitir derivar el comportamiento del suelo (su humedad) ante las distintas variaciones de las condiciones climáticas (precipitación y evaporación), con sus consiguientes aplicaciones. Por otra parte, la comparación de los datos con los facilitados por las estaciones meteorológicas y de aforos y agronómicas puede ayudar a complementar los modelos, y a comprobar y perfeccionar los resultados obtenidos por ArduDrop.

3. Descripción del dispositivo

ArduDrop 1.0 es el nombre dado al dispositivo que aquí se presenta (Figura 2). Su nombre deriva de las palabras “Arduino” y “Drop”. “Arduino” es el nombre de la plataforma libre para prototipos electrónicos en el que se basa este dispositivo (Link 1). “Drop” es la palabra inglesa de “gota”, con el que hemos querido reflejar la finalidad del dispositivo para el uso eficaz del agua.

El objetivo de ArduDrop es adquirir distintos tipos de datos ambientales para que éstos puedan ser posteriormente analizados con el fin de realizar una mejor gestión del agua. Para ello, el principal sensor, y que en parte ha condicionado el diseño de este dispositivo, es un sensor de humedad del suelo. Esta humedad, en condiciones naturales, depende en parte de las condiciones meteorológicas, por lo que ArduDrop 1.0 también dispone de distintos sensores para medir temperatura superficial al sol, luminosidad, temperatura ambiente, humedad ambiental y un sensor de intensidad de lluvia (disdrómetro). Todos estos sensores están conectados a un circuito electrónico

desarrollado por los autores para este proyecto en el interior de un adaptador de corriente reciclado. Esta electrónica está a su vez conectada a la placa principal en la que se encuentra el microprocesador y la tarjeta de memoria que almacena los datos adquiridos por los sensores. Con el fin de establecer la hora de las mediciones, el dispositivo cuenta con un reloj de tiempo real. Estos dispositivos se encuentran en el interior de una caja estanca de electrónica.

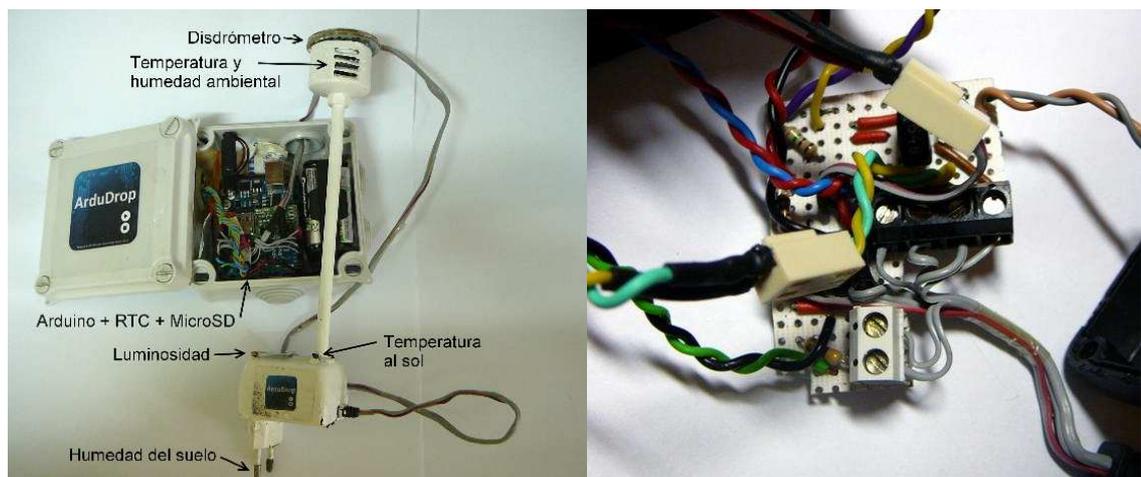


Figura 2: Dispositivo ArduDrop 1.0 (izqda) y electrónica del módulo de sensores (dcha)

El funcionamiento de ArduDrop es sencillo. Una vez conectado éste (hardware) a la fuente de alimentación (baterías, ordenador o red eléctrica con adaptador de corriente), el microcontrolador comienza a correr el programa grabado en su memoria interna (firmware). Una vez iniciado el programa, éste lee la hora y espera hasta las horas en punto para realizar las medidas. Mientras espera dicho momento, hace parpadear una pequeña luz situada en el exterior del dispositivo para indicar que ArduDrop está funcionando y en modo espera. Cuando el reloj del dispositivo indica la hora en punto, la luz se enciende de forma continua dando inicio a la toma de medidas de todos los sensores del dispositivo, empezando por el voltaje y la temperatura del microprocesador, la temperatura al sol, la temperatura ambiental, la humedad ambiental, la humedad del suelo, y la luminosidad. Todas estas medidas se toman en milisegundos, pero además el dispositivo mide la intensidad de la lluvia durante 30 segundos mediante el disdrómetro. Una vez completadas todas estas medidas, los datos adquiridos se guardan en formato ASCII (Figura 3) en una tarjeta de memoria flash microSD, y se apaga la luz, que comienza a parpadear durante una hora hasta la siguiente medida. Si el dispositivo se encuentra conectado a un ordenador, es posible visualizar los datos adquiridos en tiempo real mediante una conexión serie.

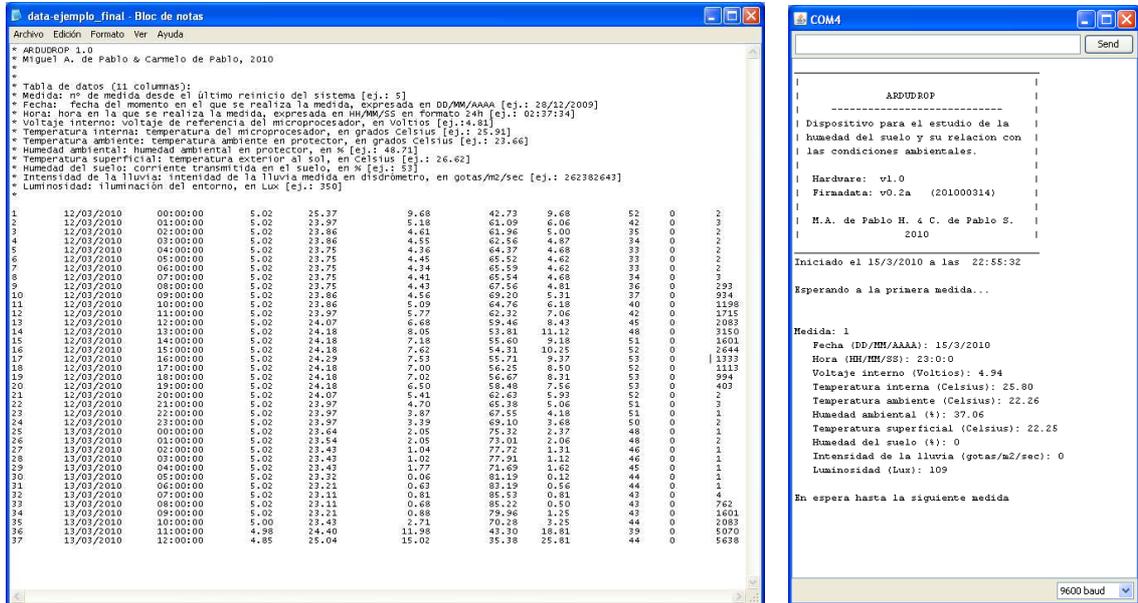


Figura 3: Ejemplos de visualización de datos adquiridos por ArduDrop 1.0 en el fichero ASCII grabado en la memoria flash (izqda.), y en pantalla, a través del puerto serie, en tiempo real (dcha.).

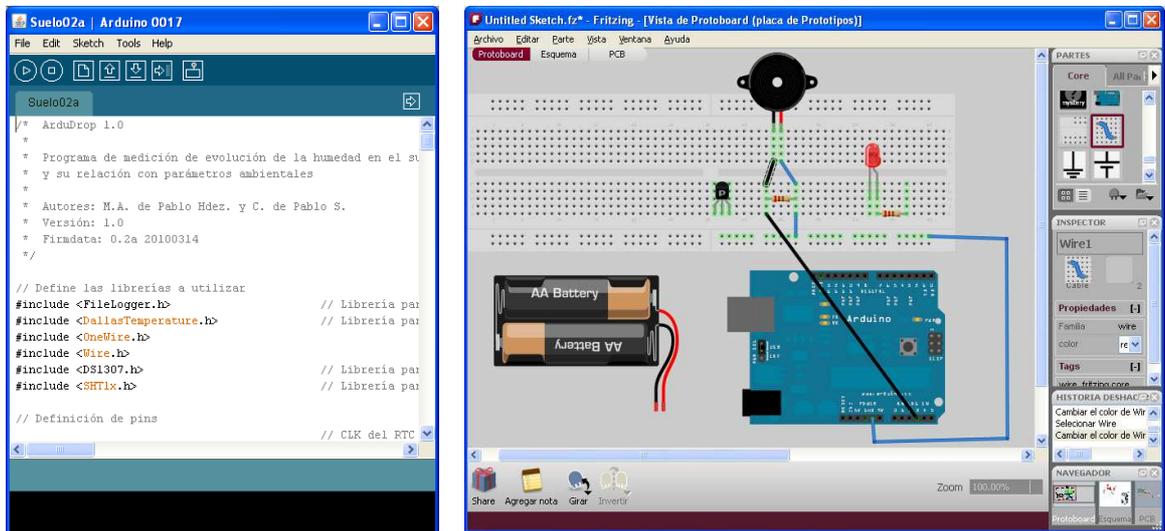


Figura 4: Entorno de programación del dispositivo con la plataforma Arduino v17 (izqda.) y de modelización del circuito electrónico con el programa Fritzing (dcha.).

Este funcionamiento ha sido programado de forma sencilla con el entorno de desarrollo también denominado Arduino (Link 2), en lenguaje de programación C, lo que junto con el programa de modelización electrónica de prototipos Fritzing (Link 3), ha permitido el desarrollo del dispositivo y del programa que lo controla. La modificación del código del programa, como se verá más adelante permitiría realizar medidas con otras frecuencias, mostrar los datos en una ventana gráfica en el ordenador, conectar o desconectar sensores, etc. adaptándolo a las necesidades del usuario.

Aunque todos los sensores empleados funcionan con una corriente de hasta 5V, éste dispositivo se puede alimentar con una corriente de hasta 12V. La forma de alimentar este dispositivo es variada, y puede hacerse mediante baterías, una conexión USB a un ordenador, o mediante un conector de corriente a la red eléctrica con el respectivo adaptador. Nosotros hemos optado por el uso de 6 baterías NiH AA (1.2 V, 2450 mAh) recargables, conectadas a una placa solar (9.5V, 310mA) con un circuito cargador, con el fin de hacerlo autónomo e independiente de la corriente eléctrica. Es necesario comentar que una alimentación inferior a 7V puede producir desestabilización de la placa Arduino y conllevar errores en las medidas. Sin embargo, nosotros no hemos observado este problema en las pruebas realizadas con los prototipos anteriores y el descrito en este trabajo. En cualquier caso, el desarrollo del cargador de baterías (ej., Link 4) no es el objeto de este trabajo, pero da una idea de cómo es posible la instalación de ArduDrop en cualquier ubicación).

Excepto la electrónica, todos los materiales empleados en el desarrollo de ArduDrop son reciclados: desde una lata de refrescos para la superficie sensible del disdrómetro, cables de equipos electrónicos viejos para las conexiones, hasta materiales aislantes. Dos de los sensores (disdrómetro y sensor de humedad del suelo) han sido también desarrollados específicamente para este trabajo, basado en ideas o en circuitos y sensores existentes desarrollados por otros autores, equipos científicos y empresas especializadas (Link 5, Link 6, Link 7; Salmi et al., 2008; Iseman, 2009). Los objetivos de hacerlo de esta forma han sido: reducir los costes de producción del prototipo, servir como ejemplo educativo del uso de materiales reciclados, y simplificar el dispositivo. Este carácter educativo es también parte de la filosofía de la plataforma y entorno de desarrollo Arduino y del programa Fritzing, con los que se ha ideado el dispositivo, y que permite múltiples modificaciones y mejoras del presente prototipo. Sin embargo, es posible incluir sensores comerciales para la medida de la humedad del suelo (ej., Link 8), así como para la medida real de la precipitación (Link 9), aunque este último tipo de dispositivos son voluminosos y más costosos. Pero ArduDrop está abierto a mejoras y modificaciones, incluyendo la modificación de algunos de los sensores o la inclusión de otros nuevos.

4. Hardware

ArduDrop se basa en el uso de la plataforma libre de desarrollo de prototipos electrónicos y programación de microcontroladores Arduino (Link 1). Además se han empleado diferentes sensores comerciales. ArduDrop se encuentra dividido en dos partes. La primera de ellas consiste en la placa base del dispositivo con un reloj de tiempo real para sincronizar las mediciones, la alimentación y la tarjeta de memoria en la que se guardan los datos. Este conjunto se encuentra instalado en el interior de una caja estanca de electrónica. La segunda parte del dispositivo, conectada mediante un cable de nueve hilos a la placa principal, consiste en la base de un transformador de corriente reciclado en cuyo interior se encuentra la electrónica de los distintos sensores, al que se le ha añadido un pequeño mástil para el alojamiento de un protector de radiación para el sensor de temperatura y humedad ambiental, y el disdrómetro para la medición de la intensidad de la lluvia.

A continuación se describen cada uno de los sensores y partes electrónicas del dispositivo ArduDrop 1.0:

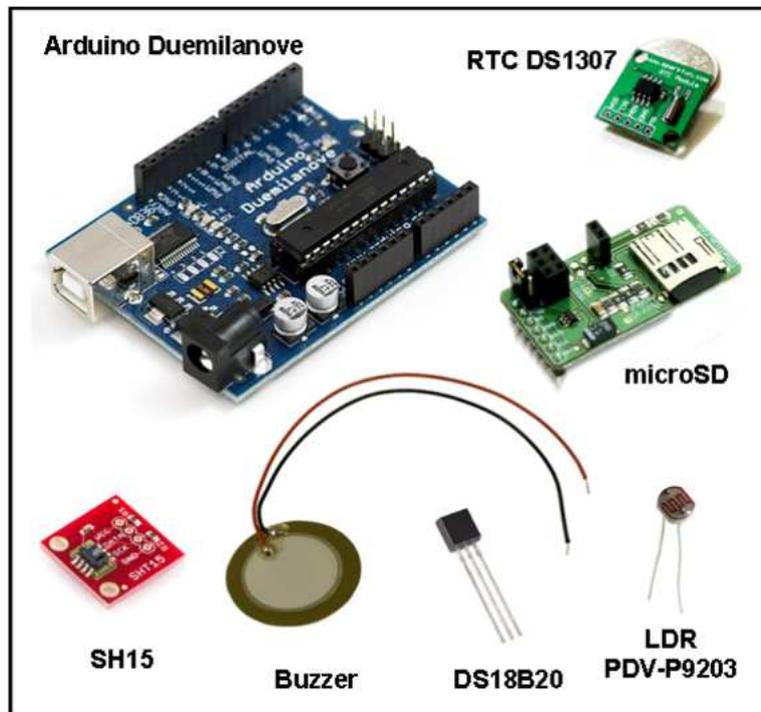


Figura 5: Placa Arduino y sensores empleados en el dispositivo ArduDrop 1.0

4.1. Arduino Duemilanove

La placa Arduino es la base del dispositivo ArduDrop. Arduino es una plataforma física abierta con un microcontrolador y creada para el desarrollo de dispositivos electrónicos basados en diferentes sensores, motores, etc. (Link 1). Actualmente existen diferentes placas Arduino, pero la versión básica más reciente es la placa Arduino Duemilanove cuyo microcontrolador es un Atmel ATmega328 (Link 10). Esta placa dispone de 14 pins de entrada y salida digitales, 6 pins analógicos de entrada, un oscilador de 16 MHz, un conector USB, un conector de corriente, cabezales ICSP y un botón de reinicio (Figura 5). Esta placa opera a 5V, pero se puede alimentar con una corriente de entre 7 y 12 V y un máximo de 50 mA. El microcontrolador, operativo a temperaturas de entre -25°C y 85°C, dispone de una memoria flash de 32Kb, una memoria RAM de 2Kb y otra memoria EEPROM de 1Kb. En la memoria flash del microcontrolador almacena el código de funcionamiento o firmware desarrollado para el dispositivo. Sin embargo, esta memoria no es suficiente para el almacenaje de gran cantidad de datos, por lo que en este proyecto se ha empleado un componente adicional para añadir capacidad de almacenaje en una memoria flash de 2Gb en una tarjeta microSD. Su programación, en este proyecto, se ha realizado a través del entorno de programación Arduino v017 (Link 2, Figura 4) a través de una conexión USB.

4.2. Reloj interno

La obtención de la fecha y hora real se realiza a través del uso de un reloj de tiempo real. Para ello se ha empleado el componente DS1307 RTC de Sparkfun (Link 11). Este componente (Figura 5) debe programarse previamente, pero dispone de una pila de botón para evitar la pérdida de la fecha y hora en caso de pérdida de corriente en la placa base (Arduino) a la que se encuentra conectado para el volcado de datos. Este componente es capaz de facilitar hora, minuto y segundo, día de la semana, día del mes, mes, y año gracias a su memoria no volátil de 56 bytes. Tiene un reducido consumo (0.5 mA) y es operativo entre -40°C y +85°C. Su conexión es a través de cuatro cables: +5V, GND, SCL, y SDA, estos últimos conectados a los pin 4 y 5 analógicos de la placa Arduino, respectivamente.

4.3. Módulo microSD

Los datos recibidos por cada uno de los sensores así como del reloj de tiempo real son almacenados en una memoria flash consistente en una tarjeta microSD introducida en un módulo (Figura 5) especialmente desarrollado por Libelium (Link 12) para instalar en la placa Arduino. Su conexión es a través del protocolo I2C, a los pins digitales 8 a 12 de la misma. La tarjeta microSD empleada tiene una capacidad de 2Gb, formateada en modo FAT, en la que es necesario grabar previamente un fichero en formato ASCII con la cabecera del fichero de datos. El formato de los datos a grabar en dicho fichero se realiza por código en el firmware.

4.4. Humedad del suelo

La medida de la humedad del suelo se basa en la resistencia del terreno al paso de una corriente eléctrica, que varía en función del tipo de terreno y de la cantidad de agua. Para realizar esta medición no se ha empleado un sensor comercial, sino que se ha desarrollado uno basado en otros dispositivos y circuitos, algunos de ellos para su uso con la plataforma Arduino (Iseman, 2009; Link 8; Link 13). Como elementos sensores se ha empleado los bornes metálicos de un enchufe de un cargador de baterías, aunque se emplearon clavos de acero galvanizado en versiones preliminares del dispositivo ArduDrop. La calibración de este sensor se realiza mediante electrónica y código, realizando ensayos con distinta humedad sobre el terreno en cuestión a monitorizar, ajustando la resistencia variable del circuito, y programando los cálculos adecuados en el código para la conversión de la señal entrante en % de humedad del suelo.

4.5. Temperatura y humedad ambiental

La temperatura y humedad ambiental se miden mediante el componente SH15 (Figura 5) montado sobre su propia placa de Sensirion (Link 14). Este sensor es operativo entre -40°C y $+125^{\circ}\text{C}$ de temperatura y de 0% a 100% de humedad relativa. Tiene una resolución de $\pm 0.3^{\circ}\text{C}$ en la temperatura y de $\pm 2\%$ en la humedad relativa. Este sensor está conectado a +3V, GND de la placa Arduino para la alimentación, y a los pin digitales 5 y 6 para la transmisión de datos, sin necesidad de componentes adicionales a los incluidos en la propia placa. Para evitarlos daños de la humedad sobre el circuito impreso, el componente se encuentra envuelto en plástico quedando al descubierto únicamente la parte sensible del circuito. Por otro lado, para evitar el efecto de la radiación solar se ha instalado el dispositivo en el interior de un protector de radiación realizado para tal fin mediante un tapón de PVC de 40 mm de diámetro al que se le han realizado diversas ranuras para permitir el acceso del aire pero no de la luz solar, a la vez que la ventilación adecuada del sensor.

4.6. Disdrómetro

El disdrómetro es un dispositivo capaz de sentir las gotas de lluvia, basado en el uso de un sensor piezoeléctrico (Figura 5) instalado en sentido inverso con una resistencia puente, es decir, produciendo un pulso eléctrico al recibir una vibración, en lugar de producir vibraciones sonoras debidas a una corriente eléctrica (Link 15). Este dispositivo no consume corriente eléctrica, al estar conectado a negativo y a un pin analógico de la placa Arduino. La calibración y sensibilidad de este sensor se realiza a través del código. Para amplificar la señal, el dispositivo piezoeléctrico se encuentra pegado a una superficie impermeable levemente convexa realizada a partir de la base de una lata de refresco. El sensor se encuentra pegado en el interior de esta superficie y protegido en su parte inferior con una placa de metacrilato. El conjunto se encuentra instalado en la parte superior del

protector de radiación anteriormente descrito, y separado de él mediante unas almohadillas de espuma expandida para evitar interferencias entre el disdrómetro y el resto del dispositivo.

4.7. *Temperatura exterior*

La temperatura exterior al sol y a nivel de la superficie, se realiza con un sensor digital de Maxim (Link 16) DS18B20 (Figura 5). Este sensor, ya calibrado de fábrica, tiene una resolución de $\pm 0.5^{\circ}\text{C}$ entre -10°C y $+85^{\circ}\text{C}$. Usando la tecnología 1-Wire es posible conectar el sensor a la placa base a través de sólo dos pins: +5V y el pin digital 2. La patilla GND va conectada, mediante una resistencia, a +5V de la placa Arduino.

4.8. *Luminosidad*

Para medir la luminosidad del emplazamiento, se ha empleado una resistencia eléctrica dependiente de la luminosidad, conocidas como LDR. Se ha empleado la célula fotoresistente CdS modelo PDV-P9203 (Figura 5), que responde al espectro visible (400-700 nm), y es operativa desde -30 hasta $+75^{\circ}\text{C}$, con sensibilidad $0.9 \Omega/\text{Lux}$. Este sensor se ha instalado mediante una resistencia puente de $10\text{K}\Omega$ al pin analógico 0 de la placa Arduino.

4.9. *Otros componentes*

Además de estos sensores, se han empleado componentes electrónicos adicionales, incluyendo resistencias, leds, transistores, etc. cuya ubicación en el circuito se muestra en la Figura 6. En la tabla 1 se muestra un listado de todos los componentes empleados en el desarrollo de la electrónica del dispositivo ArduDrop 1.0, sin incluir los sensores. No se incluyen en este listado las herramientas necesarias para el montaje del dispositivo como soldador, estaño o la propia caja para el dispositivo o los sensores, el protector de radiación, etc.

Tabla 1: listado de componentes empleados en el desarrollo de ArduDrop 1.0

1 led rojo 5 mm diámetro	1 terminal de tornillo de 3 cables
1 transistor 2N2222A	3 terminales de tornillo de 2 cables
1 resistencia 4,7 K Ω carbono 5% 1/4W	9 terminales pin macho
1 resistencia 233 Ω carbono 5% 1/4W	Tubo termoretractil
1 resistencia 100 Ω carbono 5% 1/4W	1 cable plano de 9 hilos
1 resistencia 10 K Ω carbono 5% 1/4W	Cables para prototipado 1 mm diámetro
1 resistencia 1 M Ω carbono 5% 1/4W	Cables para prototipado 0,5 mm diámetro
1 resistencia variable 10 K Ω	Placa microperforada de prototipado

4.10. Circuito electrónico

Todos los sensores se encuentran conectados a una placa de prototipado microperforada en la que se han soldado los distintos componentes de los circuitos de los diferentes sensores, compartiendo alimentación y masa. Además se ha incluido un circuito adicional para un led que señala el funcionamiento del dispositivo. Esta placa ha sido diseñada para ser introducida en el interior de un transformador de corriente reciclado que se ha empleado como base para los sensores. El conjunto de los sensores está conectado con la placa Arduino mediante un cable plano de nueve hilos (Figura 5). En la Figura 6 se muestra el esquema del circuito electrónico final del dispositivo ArduDrop 1.0.

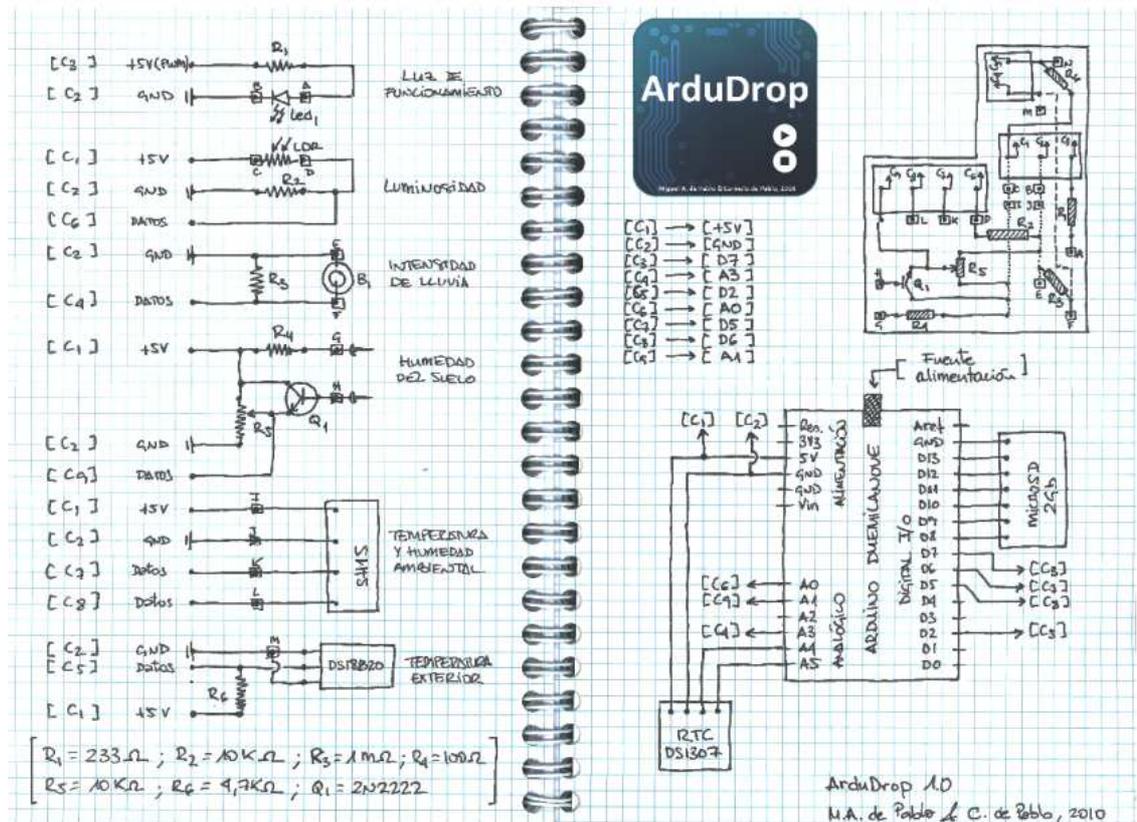


Figura 6: Esquema del circuito electrónico de ArduDrop 1.0 (Fuente: Elaboración propia)

5. Firmware

Las medidas realizadas por los sensores de ArduDrop, el control del tiempo para realizar dichas medidas, y la grabación de los datos en la memoria flash, están controladas por el código grabado en la memoria del microcontrolador de la placa base Arduino. Este código está programado sobre el Entorno de Desarrollo también denominado Arduino (Link 2), mediante el uso del lenguaje C. Este código, o firmware, está especialmente desarrollado para la configuración electrónica de la versión 1.0 de ArduDrop. Variaciones en las conexiones o en la electrónica del dispositivo podría derivar en el mal funcionamiento del dispositivo o en la obtención de datos erróneos.

La versión actual del firmware es la 0.2a, y con ella se realizan medidas a través de los diferentes sensores una vez cada hora, guardándose los datos en la memoria flash y manteniendo una pequeña luz parpadeando mientras se espera a la siguiente medida. El código incluye una serie de rutinas que permiten visualizar los datos en el ordenador a través del puerto serie del mismo.

Para que la placa base Arduino Duemilanove reconozca los sensores y dispositivos conectados a ella, realice la medición a través de los sensores, convierta la señal en las respectivas unidades de medida y grabe los datos en la tarjeta de memoria, es necesario emplear distintas librerías que simplifican el código del firmware. Estas librerías han sido desarrolladas por distintos autores y se encuentran en continua actualización, por lo que su correcto funcionamiento no está asegurado con el presente código y configuración electrónica para futuras actualizaciones, ni nuevas versiones de la plataforma Arduino v17. Sin embargo, no suelen producirse importantes modificaciones y las correcciones de las librerías suelen ser rápidas a través de la página de Arduino y su foro de usuarios (Link 1, Link 17). Las librerías empleadas son: (1) Wire y DS1307, para conectar el reloj de tiempo real DS1307, (2) DallasTemperature y OneWire, para conectar el sensor de temperatura DS18B20, (3) SH1x, para conectar el sensor de temperatura ambiente y humedad relativa SH15, y (4) FileLogger, para la grabación de los datos en la memoria flash microSD.

Además de realizar medidas de humedad del suelo, temperatura al sol, luminosidad, temperatura ambiental, humedad relativa e intensidad de lluvia, que son el objetivo del dispositivo, se ha programado la lectura del voltaje interno del microprocesador y su temperatura. Para ello no se requiere sensores, sino que se emplean las propias características del microprocesador (Link 18; Link 19). De esta forma, ante posibles problemas en el dispositivo es posible reconocer si dichos problemas se deben a la falta de alimentación o al sobrecalentamiento del circuito integrado, lo que ayuda a localizar posibles problemas en el dispositivo.

El funcionamiento del dispositivo, mencionado anteriormente en el apartado 2 de este trabajo, está recogido en el código del firmware. El cuadro 1 muestra la secuencia de comandos en pseudocódigo que ha sido la base de programación en el entorno Arduino 17.

Cuadro 1: Pseudocódigo del firmware 0.2a del dispositivo ArduDrop 1.0.

1. Definición de los pins
2. Declaración y configuración de las librerías
3. Definición de las constantes y variables
4. CONFIGURACIÓN INICIAL
4.1. Configuración de los pins
4.2. Configuración de la comunicación por el puerto serie (*)
4.3. Mostrar la pantalla de inicio en el puerto serie (*)
5. DESARROLLO DEL PROGRAMA
5.1. Comprueba la hora
5.2. Si es el momento de medir...
5.2.1. Enciende la luz de forma permanente
5.2.2. Realiza la medida a través de todos los sensores
5.2.3. Muestra los datos a través del puerto serie (*)
5.2.4. Graba los datos en la memoria flash
5.2.5. Apaga la luz
5.2.6. Espera una hora mientras hace parpadear una luz cada 5 segundos
5.3. Si no es la hora...
5.3.1. Comprueba la hora
5.3.2. Calcula el tiempo de espera hasta las mediciones
5.3.2. Hace parpadear una luz cada 5s durante el tiempo restante

Las instrucciones marcadas con asterisco (*) en el pseudocódigo son opcionales para mostrar datos a través del puerto serie de un ordenador. La eliminación de las mismas o su comentario puede ayudar a reducir el tamaño del fichero compilado final (y a la liberación de memoria que permita que el programa sea más eficiente y sin fallos), que en la actualidad es de unos 18800 bytes, dentro de la capacidad de la memoria flash interna del microprocesador ATmega328 de la placa base Arduino empleada.

Con el fin de mantener el código legible se ha optado por: comentar las distintas operaciones e instrucciones dentro del propio código y crear funciones para las distintas operaciones (lectura de cada sensor, grabación de datos, etc.). De esta forma, el código es más accesible y permite localizar con mayor precisión cada una de las instrucciones, lo que sin duda es de gran ayuda para realizar futuras modificaciones del código para su mejora o ampliación.

En el Anexo 1 se muestra el código final completo y comentado de la versión 0.2a del firmware del dispositivo ArduDrop 1.0, en funcionamiento durante 3 meses.

6.- Salida de datos

Cuadro 2: Ejemplo de datos grabados en tarjeta microSD en formato ASCII, precedidos de la cabecera preexistente en el fichero "datos.log"

* ARDUDROP 1.0										
* Miguel A. de Pablo y C. de Pablo S., 2010										
*										
* Tabla de datos (11 columnas):										
* Medida: nº de medida desde el último reinicio del sistema [ej.: 5]										
* Fecha: fecha del momento en el que se realiza la medida, expresada en DD/MM/AAAA [ej.: 28/12/2009]										
* Hora: hora en la que se realiza la medida, expresada en HH/MM/SS en formato 24h [ej.: 02:37:34]										
* Voltaje interno: voltaje de referencia del microprocesador, en Voltios [ej.:4.81]										
* Temperatura interna: temperatura del microprocesador, en grados Celsius [ej.: 25.91]										
* Temperatura ambiente: temperatura ambiente en protector, en grados Celsius [ej.: 23.66]										
* Humedad ambiental: humedad ambiental en protector, en % [ej.: 48.71]										
* Temperatura superficial: temperatura exterior al sol, en Celsius [ej.: 26.62]										
* Humedad del suelo: corriente transmitida en el suelo, en % [ej.: 53]										
* Intensidad de la lluvia: intensidad de la lluvia medida en disdrómetro, en gotas/m2/sec [ej.: 262382643]										
* Luminosidad: iluminación del entorno, en Lux [ej.: 350]										
*										
1	12/03/2010	00:00:00	5.02	25.37	9.68	42.73	9.68	52	0	2
2	12/03/2010	01:00:00	5.02	23.97	5.18	61.09	6.06	42	0	3
3	12/03/2010	02:00:00	5.02	23.86	4.61	61.96	5.00	35	0	2
4	12/03/2010	03:00:00	5.02	23.86	4.55	62.56	4.87	34	0	2
5	12/03/2010	04:00:00	5.02	23.75	4.36	64.37	4.68	33	0	2
6	12/03/2010	05:00:00	5.02	23.75	4.45	65.52	4.62	33	0	2
7	12/03/2010	06:00:00	5.02	23.75	4.34	65.59	4.62	33	0	2
8	12/03/2010	07:00:00	5.02	23.75	4.41	65.54	4.68	34	0	3
9	12/03/2010	08:00:00	5.02	23.75	4.43	67.56	4.81	36	0	293
10	12/03/2010	09:00:00	5.02	23.86	4.56	69.20	5.31	37	0	934
11	12/03/2010	10:00:00	5.02	23.86	5.09	64.76	6.18	40	0	1198
12	12/03/2010	11:00:00	5.02	23.97	5.77	62.32	7.06	42	0	1715
13	12/03/2010	12:00:00	5.02	24.07	6.68	59.46	8.43	45	0	2083
14	12/03/2010	13:00:00	5.02	24.18	8.05	53.81	11.12	48	0	3150
15	12/03/2010	14:00:00	5.02	24.18	7.18	55.60	9.18	51	0	1601
16	12/03/2010	15:00:00	5.02	24.18	7.62	54.31	10.25	52	0	2644
17	12/03/2010	16:00:00	5.02	24.29	7.53	55.71	9.37	53	0	1333
18	12/03/2010	17:00:00	5.02	24.18	7.00	56.25	8.50	52	0	1113
19	12/03/2010	18:00:00	5.02	24.18	7.02	56.67	8.31	53	0	994
20	12/03/2010	19:00:00	5.02	24.18	6.50	58.48	7.56	53	0	403
21	12/03/2010	20:00:00	5.02	24.07	5.41	62.63	5.93	52	0	2
22	12/03/2010	21:00:00	5.02	23.97	4.70	65.38	5.06	51	0	3
23	12/03/2010	22:00:00	5.02	23.97	3.87	67.55	4.18	51	0	1
24	12/03/2010	23:00:00	5.02	23.97	3.39	69.10	3.68	50	0	2
25	13/03/2010	00:00:00	5.02	23.64	2.05	75.32	2.37	48	0	1

Con el firmware (versión 0.2a) es posible grabar los datos en un fichero de texto en formato ASCII en la tarjeta de memoria flash microSD con que se ha dotado al dispositivo. Esta salida de datos (Figura 3) es compatible con la visualización en tiempo real de los mismos al ser adquiridos (Figura 3) gracias a la comunicación con un

ordenador a través del puerto serie. Sin embargo, puesto que los datos se adquieren cada hora, esta salida de datos ha sido desarrollada más como un sistema de control del funcionamiento del dispositivo que como una forma de volcar datos en el equipo informático.

El almacenamiento de los datos en un fichero de texto en la memoria flash, por el uso de la librería FileLogger anteriormente mencionada, requiere la preexistencia de un fichero de al menos un carácter y 1Kb de tamaño. Hemos aprovechado este requerimiento para crear un fichero de texto en formato ASCII de nombre “datos.log” (nombre y extensión modificables a través del firmware) en el que hemos incluido una cabecera que precede a los datos que se graban en el mismo, y que ayuda al usuario al entendimiento de los datos visualizados (Cuadro 2), en columnas separadas por espacios.

7.- Aplicaciones

ArduDrop 1.0 tiene diversas aplicaciones posibles, comenzando por el campo de la didáctica. Dentro de este campo, el dispositivo puede ser empleado en muy diversos ámbitos y niveles educativos, incluyendo Educación Secundaria, Bachillerato, Formación Profesional y Universidad. En el campo de las enseñanzas técnicas, el dispositivo electrónico puede ser empleado para la enseñanza de electrónica y sus aplicaciones en el medio ambiente, incentivando al alumno a desarrollar otros dispositivos que faciliten, simplifiquen o reduzcan costes de las investigaciones que se llevan a cabo en la actualidad. En el campo de las enseñanzas científicas, el dispositivo abre una vía al desarrollo de investigaciones viables gracias al uso de instrumentos de bajo coste, o en regiones en vías de desarrollo, donde la instalación masiva de sensores estándar implican costes difíciles de asumir, y que es además donde mayores necesidades hay de monitorizar la evolución del agua en el suelo, y de realizar un uso y gestión responsable de los recursos hídricos.

Desde el punto de vista científico, este dispositivo debe ser calibrado de forma adecuada para el tipo de terreno en el que se sitúe, con el fin de que los valores de humedad que devuelve el sensor sean valores realistas. El sensor de intensidad de lluvia o disdrómetro, también debe ser calibrado por código para establecer la intensidad de lluvia mínima que puede reconocer sin mezclarse con el ruido instrumental. El resto de los sensores no requieren de calibración. Todos estos sensores permiten desarrollar interesantes trabajos de relación entre los distintos parámetros para cada emplazamiento, y su relación con los valores de precipitación, escorrentía y evapotranspiración de las estaciones cercanas al punto de ubicación del sensor. Por otro lado, es posible estudiar la

relación entre parámetros como iluminación, temperatura y humedad del suelo para establecer valores aproximados de evapotranspiración y estudiar su relación con los valores estándar generalmente empleados en los balances hídricos.

Finalmente, desde el punto de vista aplicado, este dispositivo, por su sencillez de construcción, lo reducido de sus costes y su simplicidad operativa, es adecuado para realizar amplios estudios de detalle de la evolución del agua en el suelo para aplicar a la mejora del regadío, la caracterización del entorno en el que se realizarán reforestaciones para la selección adecuada de la vegetación a emplear, etc. (Allen et al., 1998). Por estas mismas razones, la utilización de este dispositivo tiene interés en áreas afectadas en mayor o menor medida por la desertización y desertificación, así como en áreas en vías de desarrollo, ya que en estos casos, la instrumentalización no suele ser extensa debido a los altos costes de instalación y mantenimiento de sensores, que suelen ser empleados en otro tipo de necesidades de la población.

8.- Mejoras del dispositivo

Esta primera versión funcional del dispositivo ArduDrop está abierta a numerosas mejoras, tanto en la parte de hardware, como en la de software. Desde el punto de vista de los sensores, es necesario mejorar la calibración del disdrómetro para que tenga una mayor sensibilidad y sea capaz de medir lluvias más finas. Sería de interés incluso incluir un sensor de lluvia capaz de detectar incluso niebla para que tenga un mayor rango de medida que el disdrómetro incluido en ArduDrop. Es posible incluir otros sensores que permitan un mejor conocimiento de las variaciones de las condiciones en el terreno y en el ambiente. El primer sensor de interés es un barómetro capaz de medir la presión atmosférica. También resultaría de interés la instalación de sensores capaces de medir la radiación incidente, especialmente la infrarroja y la ultravioleta. Esto puede hacerse para obtener valores relativos (mediante el uso de componentes led infrarrojos y ultravioletas como receptores), o absolutos mediante el uso de sensores ópticos específicos. Para el caso de la humedad del suelo, es posible emplear sensores comerciales cuyo precio es similar al de otros de los sensores ya empleados en el dispositivo, lo que permitiría una mayor fiabilidad de los datos obtenidos, aunque de igual manera se requeriría de calibraciones con el terreno en el que se ubique. Por otro lado, el estudio de la humedad del suelo, también requeriría determinar dicha humedad a diferentes profundidades, lo cual hace interesante la opción de incluir más de un sensor de humedad del suelo en el dispositivo y así poder estudiar con más detalle la evolución de la humedad en el terreno.

En cuanto a la electrónica, es necesario realizar diversas mejoras. En primer lugar es fundamental la reducción del consumo que permita al dispositivo funcionar con pilas recargables a su vez alimentadas por una placa solar, y que pueda funcionar incluso en largos periodos de nubosidad. Esta mejora puede realizarse a través de modificaciones de la electrónica y del código del firmware. Una forma de favorecer el menor consumo es el empleo de un reloj de tiempo real que sustituya al RTC DS1307 y que sea capaz de programación de alarmas de tal forma que la placa base pueda entrar en modo “hibernación” hasta que la alarma lo active para realizar las medidas. De esta forma el procesador no necesita leer el RTC de forma continua hasta que llega la hora programada en el firmware para realizar las mediciones. Otra forma de reducir el consumo es la desconexión de la corriente de los sensores hasta unos instantes antes de realizar las medidas, y su desconexión una vez que los datos han sido grabados. Esta mejora, además, permitiría que el sensor de humedad del suelo no enviara corriente de forma continua al terreno, reduciendo así los fenómenos de electrólisis que a largo plazo deterioran el propio sensor y falsean las medidas de humedad del terreno.

Otro tipo de mejoras están enfocadas a la transmisión y volcado de datos desde el Arduino a un ordenador, un dispositivo PDA o directamente a Internet. Para ello es posible la instalación de componentes bluetooth, radiofrecuencias, wifi o ethernet. Por una parte esto permitiría el control del funcionamiento del dispositivo a distancia, y del volcado de datos de forma continua manteniendo una copia de seguridad en la tarjeta de memoria flash microSD. La conexión bluetooth permitiría el volcado de datos a través de un equipo PDA, y sin necesidad de manipular la caja que contiene el dispositivo, evitando así que una mala manipulación permita la entrada de agua. Las conexiones a través de sistemas de radiofrecuencias (radio y wifi) permitiría la interacción a muy larga distancia. Por otra parte, el volcando de datos a través de Internet mediante la propia placa Arduino y otros componentes adicionales permitiría un seguimiento continuo y completamente remoto del equipo, aunque este produciría un mayor consumo de corriente, y una conexión a Internet (aunque puede hacerse vía wifi).

El firmware también puede mejorarse en diferentes aspectos. Por un lado mediante la preparación de rutinas que permitan el cálculo del punto de rocío, pero también de máximas, medias y mínimas diarias de los datos adquiridos por cada uno de los sensores, y que podrían ser almacenados en un fichero de texto adicional. También la realización de mayor número de medidas a lo largo del día (por ejemplo cada 5 minutos), con el fin de obtener una señal más detallada de cada sensor. Finalmente, aumentando las medidas (10 medidas en vez de una en cada evento) y calculando la media para cada evento, se reducirían errores debidos al ruido en la electrónica, aunque en el prototipo aquí presentado, estos son despreciables dados los resultados obtenidos.

9.- Conclusiones generales

El dispositivo ArduDrop 1.0 es un ejemplo de desarrollo de tecnología de bajo coste mediante el uso de hardware y software libre para la monitorización de la humedad del suelo y de diversas constantes ambientales que pueden ayudar a comprender, de forma local, la evolución del agua en el terreno. Este dispositivo es de fácil construcción y no requiere de grandes conocimientos de electrónica ni programación, y puede ser fácilmente mejorable y ampliable por personal especialmente formado en estos ámbitos. El objetivo de este dispositivo es realizar un acercamiento cualitativo pero con datos reales al balance hidrológico de una zona determinada. ArduDrop 1.0 tiene múltiples aplicaciones, tanto en el campo de la didáctica (en ámbitos técnicos y científicos), como en el del medio ambiente, ya que una serie larga de datos puede ayudar a conocer la evolución de la humedad del terreno en función de las condiciones ambientales y, con ello, a realizar riegos adecuados o a la selección de vegetación capaz de adaptarse a las condiciones existentes, entre otras aplicaciones.

En este trabajo se ha presentado el dispositivo y los sensores que lo constituyen, así como el programa que lo controla y un ejemplo de los datos obtenidos.

10.- Agradecimientos

Este trabajo ha sido posible gracias a la plataforma libre (open-hardware) Arduino, así como al entorno de desarrollo Arduino 017 (www.arduino.cc), pero especialmente a la comunidad de usuarios de Arduino a través de su foro. Algunas partes del código han sido propuestas o depuradas por algunos de sus usuarios, que también han sido de gran importancia gracias a los tutoriales desarrollados por algunos de ellos en el pasado para el manejo de distintos sensores, y al desarrollo de librerías para su programación más sencilla. La comunidad Arduino ha sido de gran ayuda para suplir nuestros escasos conocimientos en electrónica de microcontroladores y de programación en lenguaje C. Nuestro agradecimiento a la Revista *Tecnología@ y Desarrollo* por su interés en este trabajo, y especialmente a Julio Merino y a los revisores anónimos cuyos comentarios han ayudado a mejorar el trabajo. Finalmente, los autores deseamos agradecer especialmente a nuestra familia su paciencia por las horas en las que hemos estado enfrascados en el desarrollo del dispositivo.

11.- Referencias

- ALLEN, R.G., PEREIRA, L.S., RAES, D., SMITH, M. (1998) “*Crop evapotranspiration – guidelines for computing crop water requirements*” Irrigation and drainage paper, 56. FAO, Rome, Italy. 300 pp.
- ISEMAN, L. (2009) “*The Garduino garden controller*”. Make magazine, 18. 91- 101.
- KOURTELLIS, A.G., KASPARIS, T., LINWOOD, J., LANE, J. (2005). “*Disdrometer calibration using an adaptive signal processing algorithm*”. Proc. Oceans,2005, Washington, DC, MTS/IEEE, 2572–2577.
- LICZNAR, P., ŁOMOTOWSKI, J., BŁON’ SKI, S. (2008). “*Microprocessor field impactometer calibration: Do we measure drops’ momentum or their kinetic energy?*” Journal of Atmospheric and Oceanic Technology, 25. 742-753.
- PALMER, W.C., and Havens, H.A. (1958) “*A graphical technique for determining evapotranspiration by the Thornthwaite method*”. Monthly Weather Review, 86. 123-128.
- PEREIRA, A.R., PRUIT, W.O. (2004). “*Adaptation of the Thornthwaite scheme for estimating daily reference evapotranspiration*”. Agricultural Water Management, 66. 251–257.
- SALMI, A., ELOMAA, L., KOPSALA, P., LAUKKANEN; E. (2008) “*Piezoelectric Vaisala Raincap® rain sensor applied to drop size distribution measurement*”. Technical Conference on Meteorological and Environmental Instruments and Methods of Observation, Abstract 2-11. 7 páginas.
- SMITH, J.A., DE VEAUX, R.D. (1994) “*A stochastic model relating rainfall intensity and raindrop processes*” Water Resources Research, 30(3), 651–664.
- SMITH, P.L., LIU, Z., JOSS, J. (1993) “*A study of sampling variability effects in raindrop size observations*”. Journal of Applied Meteorology, 32. 1259–1269.
- THORNTHWAITE, C.W. (1948) “*An approach toward a rational classification of climate*”. Geographic Review, 38. 55-94.

12.- Direcciones web

- 1: <http://www.arduino.cc/>
- 2: <http://arduino.cc/en/Main/Software>
- 3: <http://fritzing.org/>
- 4: <http://www.instructables.com/id/Self-Sufficient-Arduino-Board/>
- 5: http://www.botanicalls.com/archived_kits/twitter/
- 6: http://www.instructables.com/id/Garduino_Gardening_Arduino/
- 7: <http://www.vaisala.com/weather/applications/oem-precipitation.html>

24. Miguel A. de Pablo Hdez. y C. de Pablo S.

- 8: <http://www.vegetronix.com/Soil-Moisture-Probe.phtml>
- 9: http://www.sparkfun.com/commerce/product_info.php?products_id=8942
- 10: <http://arduino.cc/en/Main/ArduinoBoardDuemilanove>
- 11: http://www.sparkfun.com/commerce/product_info.php?products_id=99
- 12: http://www.libelium.com/tienda/catalog/product_info.php?products_id=66
- 13: http://www.botanicalls.com/archived_kits/twitter/
- 14: <http://www.sensirion.cc/product/8000/236.html>
- 15: <http://www.instructables.com/id/Make-an-acoustic-rain-gauge-disdrometer/>
- 16: http://www.maxim-ic.com/quick_view2.cfm?qv_pk=2812
- 17: <http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl>
- 18: <http://code.google.com/p/tinkerit/wiki/SecretThermometer>
- 19: <http://code.google.com/p/tinkerit/wiki/SecretVoltmeter>

ANEXO 1 Código comentado del Firmware versión 0.2a del dispositivo ArduDrop 1.0

```

/* ArduDrop 1.0
 *
 * Programa de medición de evolución de la humedad en el suelo
 * y su relación con parámetros ambientales
 *
 * Autores: M.A. de Pablo Hdez. y C. de Pablo S.
 * Versión: 1.0
 * Firmdata: 0.2a 20100314
 */

// Define las librerías a utilizar
#include <FileLogger.h>           // Librería para almacenamiento de datos en microSD
#include <DallasTemperature.h>   // Librería para el sensor de temperatura DS18B20
#include <OneWire.h>
#include <Wire.h>
#include <DS1307.h>              // Librería para el RTC DS1307
#include <SHT1x.h>              // Librería par el sensor de temperatura y humedad SH15

// Definición de pins
                                // CLK del RTC DS1307 en An5
                                // DATA del RTC DS1307 en An4
#define temp 2                  // Define el pin digital que recibe datos de temperatura DS18B20
#define temphumdat 5           // Define el pin digital de datos temperatura-humedad SH15
#define temphumcl 6            // Define el pin digital de reloj temperatura-humedad SH15
#define vida 7                 // Define el pin digital que envía corriente al led de vida
#define MEM_PW 8               // Define el pin de alimentación de la tarjeta microSD
#define luz 0                  // Define el pin analógico que recibe datos de luminosidad LDR
#define suelo 1                // Define el pin analógico que recibe datos de humedad del suelo
#define lluvia 3               // Define el pin analógico que recibe datos de lluvia

// Activación de librerías
OneWire oneWire(temp);        // Configuración librería termómetro DS18B20
DallasTemperature sensors(&oneWire);
SHT1x sht1x(temphumdat, temphumcl); // Configuración librería sensor temperatura y humedad SH15

// Definición de constantes
const int sensibilidad = 30;   // Define la sensibilidad del disdrómetro
const float area=0.001654;    // Define el área de medida del sensor disdrómetro en m^2

// Definición de variables
float innerVcc;                // Define la variable de voltaje interno
float innertemp;               // Define la variable de temperatura interna
long contador = 1;            // Define el contador de datos almacenados
long lux = 0;                 // Define la variable que guarda datos de luminosidad (LUX)
float tempext = 0;            // Define la variable que guarda datos de temperatura (°C)

```

```

float ambtemp = 0;           // Define la variable que guarda la temperatura ambiental SH15 (°C)
float humedad = 0;          // Define la variable que guarda la humedad ambiental SH15 (%)
float suelohum = 0;         // Define la variable que guarda los datos de humedad del suelo (%)
unsigned long intensidad=0; // Define la variable de intensidad de lluvia (gotas/horas/m2)
unsigned short periodo=30000; // Define los periodos de espera en milisegundos
unsigned int year = 0000;   // Define la variable que guarda el año
unsigned short int month = 00; // Define la variable que guarda el mes
unsigned short int day = 00; // Define la variable que guarda el día
unsigned short int hora = 00; // Define la variable que guarda la hora
unsigned short int minuto = 00; // Define la variable que guarda el minuto
unsigned short int segundo = 00; // Define la variable que guarda los segundos

void setup(){
  pinMode(MEM_PW, OUTPUT); // Establece el modo del pin del alimentación tarjeta microSD
  digitalWrite(MEM_PW, HIGH);
  pinMode(vida, OUTPUT); // Establece el modo del pin del led
  Serial.begin(9600); // Inicia comunicaciones (*)
  sensors.begin(); // Inicia el sensor de temperatura DS18B20
  tiempo(); // Lee la hora y fecha del RTC ds1307
  SplashScreen(); // Muestra la pantalla de inicia (*)
}

void loop(){
  minuto=RTC.get(DS1307_MIN,true); // Lee la fecha y hora del reloj DS1307
  if (minuto == 00){ // Si es la hora en punto inicia las mediciones
    digitalWrite(vida, HIGH); // Enciende el led mientras realiza las lecturas
    tiempo(); // Toma la hora
    delay(20);
    readVcc(); // Lee el voltaje interior
    delay(20);
    midegotas(); // Mide la intensidad de lluvia (disdrómetro)
    delay(20);
    readTemp(); // Mide la temperatura interna
    delay(20);
    temperatura(); // Mide la temperatura exterior (DS18B20)
    delay(20);
    sueloH(); // Mide la humedad del suelo
    delay (20);
    tempambiental(); // Lee temperatura y humedad (SH15)
    delay (20);
    humambiental(); // Mide la humedad ambiental (SH15)
    delay (20);
    luminosidad(); // Mide la luminosidad (LDR)
    mostrar(); // Muestra los datos a través del puerto serial (*)
    grabar(); // Graba los datos en formato ascii en tarjeta microSD
    contador = contador + 1; // Actualiza el contador de medidas
  }
}

```

```

digitalWrite(vida, LOW);           // Apaga el led al acabar el proceso
periodo=58000;                    // Establece el periodo de espera para siguiente medida
espera();                          // Hace parpadear el led mientras espera
}
periodo= (60-minuto-1)*1000;      // Establece el periodo de espera hasta iniciar
espera();                          // Espera hasta la siguiente medida
}

// Hace parpadear el led de vida
void espera(){
  unsigned long millis();
  long startTime = millis();
  while(millis() < startTime + periodo) { // Enciende la luz una vez cada 5s durante 40s
    digitalWrite(vida, HIGH);
    delay(25);
    digitalWrite(vida, LOW);
    delay(4075);
  }
  return;
}

// Obtiene la temperatura interna de dispositivo Arduino
long readTemp() {
  long readTemp=0;
  ADMUX = _BV(REFS1) | _BV(REFS0) | _BV(MUX3); // Lee la temperatura interna
  delay(20);
  ADCSRA |= _BV(ADSC); // Convierte los datos
  while (bit_is_set(ADCSRA,ADSC));
  readTemp = ADCL;
  readTemp |= ADCH<<8;
  innertemp = (readTemp - 125) * 1075; //Convierte los datos en x10^-4 °C
  innertemp = innertemp / 10000; // Convierte los datos en °C
  return innertemp;
}

//Lee el voltaje interno
long readVcc() {
  long readVcc=0;
  ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
  delay(2);
  ADCSRA |= _BV(ADSC); // Convierte los datos
  while (bit_is_set(ADCSRA,ADSC));
  readVcc = ADCL;
  readVcc |= ADCH<<8;
}

```

```
innerVcc = 1126400L / readVcc;           // Voltaje interno mV
innerVcc = innerVcc / 1000;             // Voltaje interno en V
return innerVcc;
}

// Lee la temperatura superficial (D18B20)
float temperatura() {
    sensors.requestTemperatures();      // Pide información a los sensores
    tempext=sensors.getTempCByIndex(0); // Lee el sensor (°C)
    return tempext;
}

// Mide la humedad del suelo
int sueloH(){
    suelohum=analogRead(suelo);          // Lee el sensor
    suelohum=(suelohum*0.1904761905);    // Calcula la humedad *100/525 (%)
    return suelohum;
}

// Mide la temperatura ambiental (SH15)
float tempambiental(){
    ambtemp = sht1x.readTemperatureC(); // Lee el sensor
    return ambtemp;
}

// Mide la humedad ambiental (SH15)
float humambiental(){
    humedad = sht1x.readHumidity();     // Lee el sensor
    return humedad;
}

// Mide la luminosidad (LDR)
unsigned int luminosidad(){
    float photocellReading0 = analogRead(luz); // Lee el sensor
    float Vout0=photocellReading0*0.0048828125; // Calcula el voltaje de referencia
    lux=500/(10*((5-Vout0)/Vout0));           // Convierte los datos a Lux
    return lux;
}

// Mide la intensidad de lluvia (disdrómetro)
unsigned long midegotas() {
    volatile unsigned long gotas=0;
    unsigned long millis();
    long startTime = millis();
    while(millis() < startTime + 30000) { // Mide el número de gotas en 30 segundos
        int sensorReading = analogRead(lluvia);
    }
}
```

```

    if (sensorReading >= sensibilidad) {
        gotas=gotas+1;
    }
}
intensidad = ((gotas / area) / 30);           // Calcula la intensidad de la lluvia en gotas/m2/sec
}

// Pantalla de inicio (*)
void SplashScreen () {
    Serial.println(" ");
    Serial.println("----- ARDUDROP 1.0 -----");
    Serial.println("-----");
    Serial.println(" Hardware: v1.0 ");
    Serial.println(" Firmdata: v0.2a (201000315) ");
    Serial.println(" ");
    Serial.println(" M.A. de Pablo & C. de Pablo S., 2010 ");
    Serial.println("-----");
    Serial.println();
    Serial.print("Iniciado el ");
    Serial.print(day);
    Serial.print("/");
    Serial.print(month);
    Serial.print("/");
    Serial.print(year);
    Serial.print(" a las ");
    Serial.print(hora);
    Serial.print(":");
    Serial.print(minuto);
    Serial.print(":");
    Serial.println(segundo);
    Serial.println();
    Serial.print("Esperando a la primera medida...");
    Serial.println();
    return;
}

// Muestra las mediciones a través del puerto serie (*)
void mostrar() {
    Serial.println();
    Serial.println();
    Serial.print("Medida: ");
    Serial.println(contador);
    Serial.print(" Fecha (DD/MM/AAAA): ");
    Serial.print(day);
    Serial.print("/");
    Serial.print(month);

```

```
Serial.print("/");
Serial.println(year);
Serial.print(" Hora (HH/MM/SS): ");
Serial.print(hora);
Serial.print(":");
Serial.print(minuto);
Serial.print(":");
Serial.println(segundo);
Serial.print(" Voltaje interno (Voltios): ");
Serial.println(innerVcc);
Serial.print(" Temperatura interna (Celsius): ");
Serial.println(innertemp);
Serial.print(" Temperatura ambiente (Celsius): ");
Serial.println(ambtemp);
Serial.print(" Humedad ambiental (%): ");
Serial.println(humedad);
Serial.print(" Temperatura superficial (Celsius): ");
Serial.println(tempext);
Serial.print(" Humedad del suelo (%): ");
Serial.println(suelohum, DEC);
Serial.print(" Intensidad de la lluvia (gotas/m2/sec): ");
Serial.println(intensidad);
Serial.print(" Luminosidad (Lux): ");
Serial.println(lux, DEC);
Serial.println();
Serial.println("En espera hasta la siguiente medida");
Serial.println();
return;
}

// Graba los datos en formato ascii en un soporte microSD
void grabar(){
// Convierte los datos decimales en valores enteros para ser grabados
volatile short innerVcc1= (int)innerVcc;
volatile short innerVcc2= (abs)((int)((innerVcc-innerVcc1)*100));
volatile short innertemp1= (int)innertemp;
volatile short innertemp2= (abs)((int)((innertemp-innertemp1)*100));
volatile short ambtemp1= (int)ambtemp;
volatile short ambtemp2= (abs)((int)((ambtemp-ambtemp1)*100));
volatile short humedad1= (int)humedad;
volatile short humedad2= (abs)((int)((humedad-humedad1)*100));
volatile short tempext1= (int)tempext;
volatile short tempext2= (abs)((int)((tempext-tempext1)*100));
volatile short suelohum1=(int)suelohum;
char message[200];
printf(message,"%ld\t %02u/%02u/%u\t %02u:%02u:%02u\t // Da formato a los datos
%hd.%02hd\t %4hd.%02hd\t
```

```
%4hd.%02hd\t %3hd.%02hd\t %4hd.%02hd\t %3hu\t %lu\t %lu\n", contador, day, month, year, hora,
minuto, segundo, innerVcc1, innerVcc2, innertemp1, innertemp2, ambtemp1, ambtemp2, humedad1,
humedad2, tempext1, tempext2, suelohum1, intensidad, lux);
  unsigned long length = strlen(message);
  FileLogger::append("datos.log", (byte*)message, length);           // Graba los datos
  return;
}

// Lee el Reloj de Tiempo Real (DS1307)
void tiempo(){
  day=RTC.get(DS1307_DATE, true);           // Lee la fecha actual
  month=RTC.get(DS1307_MTH,false);
  year=RTC.get(DS1307_YR,false);
  hora=RTC.get(DS1307_HR,true);           //Lee la hora actual
  minuto=RTC.get(DS1307_MIN,true);
  segundo=RTC.get(DS1307_SEC,true);
  return;
}
```