Diseño y desarrollo de un sistema embebido para una trampa pitfall con data logger

Mariano R. Droz [1], Juan A. Ramos [1], Pedro D. Benitez [1], Luz M. Zapata [1], Beatriz M. Diaz [2]

[1] Facultad de Ciencias de la Alimentación, Universidad Nacional de Entre Ríos, Av. Monseñor Tavella N° 1450, (CP 3200) Concordia, Entre Ríos, República Argentina
{mariano.droz, juan.ramos, pedrodaniel.benitez, luzmarina.zapata}@uner.edu.ar
http://www.fcal.uner.edu.ar

[2] Instituto Nacional de Tecnología Agropecuaria (INTA), Estación Experimental Agropecuaria Concordia (EEA Concordia), Ruta Provincial 22 y vías del Ferrocarril, (CP 3200) Estación Yuquerí, Concordia, Entre Ríos, República Argentina
diaz.beatriz@inta.gob.ar
https://inta.gob.ar/concordia

Abstract. Las trampas de caída (pitfall) son utilizadas en estudios de biodiversidad de los artrópodos del suelo. Los avances tecnológicos posibilitaron su evolución surgiendo las trampas de caída por tiempo. El objetivo fue diseñar y desarrollar un sistema embebido para un prototipo automatizado de pitfall dotada de un data logger y sensores meteorológicos y edáficos. La metodología incluyó actividades propias del proceso de diseño de software de tiempo real y, complementariamente, utilizó un enfoque iterativo e incremental. El prototipo se conformó por 3 módulos, el Data Logger, el Recolector y el de Alimentación. Se utilizó Arduino como plataforma de desarrollo y mediante 6 ensayos de laboratorio se comprobó que el sistema cumplió los requerimientos funcionales y que su diseño arquitectónico contribuye al mantenimiento y evolución del dispositivo. Los resultados indican la factibilidad de concretar ensayos a campo en diferentes ambientes productivos, contemplando en el futuro la utilización de Minería de Datos.

Keywords: Sistema embebido, Trampa de caída (pitfall) por tiempo, Data logger, Sensores, Artrópodos.

1 Introducción

Entre los instrumentos utilizados para llevar a cabo estudios de biodiversidad de los artrópodos del suelo, se encuentran las trampas de caída, denominadas también como "pitfall". El diseño básico de este tipo de trampa consiste en un contenedor enterrado en el suelo con la parte superior a ras de la superficie [1] y su uso se remonta a más de 120 años [2].

Éstas funcionan según el principio de que un artrópodo que se mueve en la superficie simplemente cae en un contenedor abierto (normalmente circular) excavado en el suelo. Los artrópodos generalmente mueren en una solución conservante y se devuelven a un laboratorio para su posterior clasificación e identificación [3]. De este modo, la trampa pitfall genera una estimación de la abundancia-actividad, considerando que la abundancia es un indicador de la actividad de los artrópodos, dando lugar así a su uso para estudios de biodiversidad de la fauna edáfica en diferentes escenarios, naturales o productivos [4, 5].

Producto de la evolución tecnológica de las trampas pitfall convencionales surgieron las trampas de caída por tiempo (time-sorting pitfall trap). Éstas conservan el mismo principio que las tradicionales, con la diferencia principal que el dispositivo recolector o contenedor va cambiando mediante una rotación automática cada cierto intervalo de tiempo previamente programado. Una vez cumplido el ciclo completo de rotación, se accede a la trampa y se retiran los contenedores para ser llevados al laboratorio [6, 7].

Más allá que, actualmente es posible diseñar y desarrollar unidades de registro de datos (data loggers) con hardware abierto y de bajo costo [8, 9], éstas trampas analizadas no utilizaron sensores de luz, de temperatura y humedad relativa ambiente, y de temperatura y humedad del suelo de manera simultánea.

Teniendo en cuenta lo expuesto, investigadores del Grupo Hortícola de la EEA Concordia del INTA plantearon a investigadores de la carrera Ingeniería en Mecatrónica de la Facultad de Ciencias de la Alimentación, la necesidad de avanzar en un desarrollo local adaptado a sus requerimientos.

Por ello, uno de los objetivos fue diseñar y desarrollar un sistema embebido para un prototipo automatizado de trampa de caída por tiempo, que utilice hardware abierto y de bajo costo, dotado de un data logger más completo, con pantalla y mayor cantidad de sensores, el cual permita medir simultáneamente la luz, la temperatura y humedad relativa ambiente, y la temperatura y humedad del suelo. A su vez, otro objetivo consistió en que el sistema cuente con una funcionalidad innovadora, que permita la fácil parametrización de los ensayos en estudios de biodiversidad de artrópodos de suelo y la posibilidad de soportar 2 modos de operación.

2 Metodología

Se efectuaron actividades propias de un proceso de diseño de software de tiempo real [10]: selección de la plataforma, identificación de estímulos/respuestas, análisis de temporización, diseño de procesos, diseño de algoritmos, diseño de datos y planificación del proceso.

Además, se efectuaron otras actividades relacionadas a la selección y adquisición de componentes accesorios, estructurales, mecánicos y electrónicos. Sumado a ello, se realizó el diseño e impresión de una placa de circuito impreso (PCB) y piezas 3D.

Teniendo en cuenta que el orden de las actividades en el proceso de diseño de software de tiempo real depende del tipo de sistema a desarrollar, así como de sus requerimientos de proceso y plataforma [10], se decidió utilizar también un enfoque iterativo e incremental [11].

La selección de este abordaje metodológico complementario radicó en que luego del análisis preliminar se arribó a la conclusión que sería práctico dividir el trabajo en partes más pequeñas o miniproyectos. Vale señalar que las iteraciones fueron controladas y planificadas [11], permitiendo en algunas instancias el avance de miniproyectos en paralelo.

3 Arquitectura del Hardware

El dispositivo está compuesto por 3 módulos, el *Módulo Data Logger (MDL)*, el *Módulo Recolector (MR)* y el *Módulo de Alimentación (MA)* (Fig. 1). Este diseño modular permite que cada módulo cuente con una funcionalidad específica y definida, estableciendo las interfaces de conexión y comunicación entre los mismos.

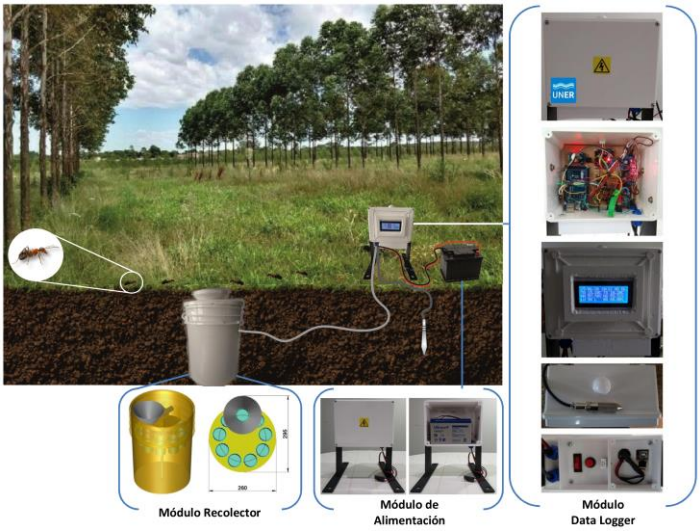

Fig. 1. Arquitectura del Hardware dispuesta en Módulos.

4 Sistema embebido

El software se programó utilizando el entorno de desarrollo Arduino IDE 1.8.10, una placa Arduino Mega 2560 Rev. 3 y cuenta con 1760 líneas de código. Para el diseño y desarrollo del sistema embebido se tuvieron en cuenta estos requerimientos funcionales (RF):

1) *Parametrizar el ensayo:* Permitir la configuración del ensayo a través de un archivo de configuración que se almacene en la tarjeta MicroSD.
2) *Controlar el funcionamiento del dispositivo:* Efectuar un control, al iniciar y durante su funcionamiento, de los principales componentes del dispositivo.

3) *Mostrar información para control:* Informar al operador, a través del display LCD, el resultado de los controles iniciales, los parámetros configurados, el modo y estado de funcionamiento.
4) *Registrar datos:* Registrar, cada cierto intervalo de tiempo configurado, la fecha, la hora, las mediciones de los sensores y el voltímetro, el número de rotación y el número de recipiente colector (vaso) utilizado.
5) *Cambiar el recipiente colector:* Rotar el disco del *Módulo Recolector* cada cierto intervalo de tiempo configurado para utilizar otro recipiente colector.
6) *Ahorrar energía:* Cuando el dispositivo no efectúe alguna tarea entrar en modo reposo hasta el próximo evento programado, o bien, hasta que el operador accione el botón de encendido del display LCD.
7) *Continuar un ensayo en proceso:* Retomar un ensayo en proceso cuando el mismo haya sido interrumpido.
8) *Registrar los principales eventos del sistema:* Llevar una bitácora con la información de los principales eventos del sistema.
9) *Soportar 2 modos de operación:* Permitir que el operador, a través del archivo de configuración que se almacene en la tarjeta MicroSD, indique el modo de funcionamiento del dispositivo: *Modo Data Logger* donde se omite el RF N° 5 y se reduce la información proporcionada para el RF N° 3, y el *Modo Completo (Modo Pitfall)* que abarca la totalidad de los RF.

Una vez analizados los requerimientos funcionales y luego del proceso de desarrollo guiado por el enfoque metodológico previamente descripto, se construyó el sistema embebido que se presenta en la Fig. 2.

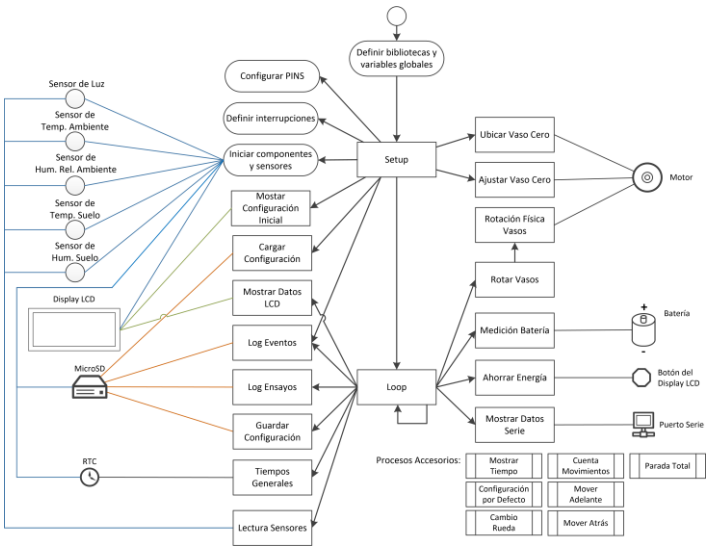

Fig. 2. Arquitectura del sistema embebido (procesos y principales componentes).

En la plataforma Arduino [12] existen 2 procesos fundamentales, *Setup* que se ejecuta sólo una vez al iniciar y *Loop* que se ejecuta indefinidamente hasta que la

placa se apaga. A partir de ello, a continuación se describen cada uno de los principales componentes del código fuente representado en la Fig 2.

- *Definir bibliotecas y variables globales:* Llamada a las bibliotecas necesarias para utilizar los sensores, los componentes (el display LCD, la tarjeta MicroSD y el Real Time Clock RTC), el protocolo I2C (utilizado por el RTC, el display LCD, el sensor de luz y el sensor de temperatura y humedad del suelo) y el modo de ahorro de energía. Además, se efectúa la declaración de las constantes y variables globales, los prototipos de las funciones y los procedimientos.

- *Configurar PINS:* Se declaran los PIN a utilizar y si los mismos serán empleados como entrada o salida.

- *Definir interrupciones:* Se definen las 2 interrupciones utilizadas y el modo de disparo de las mismas. Una interrupción está asociada al botón de encendido y apagado del display LCD y, la otra, al sensor fotoeléctrico dispuesto en el *Módulo Recolector* para controlar los movimientos.

- *Iniciar componentes y sensores:* Se inician todos los sensores y los componentes utilizados. Asimismo, se efectúa un control de los mismos para detectar el estado de su funcionamiento, informando el resultado en el display LCD (Fig. 3). La continuidad del proceso de inicio depende de un resultado satisfactorio de dichos controles. Permite cubrir parcialmente el RF N° 3.

Fig. 3. a) Pantalla al *Iniciar componentes y sensores*. **b)** y **c)** Pantallas del procedimiento *Mostrar Configuración Inicial*.

- *Cargar Configuración:* Procedimiento que permite cumplir con los RF N° 1, 7 y 9. Lee y procesa el archivo de configuración CONFIG.INI (Tabla 1) desde la tarjeta MicroSD, si la operación es exitosa, las variables de operación se establecen con los valores leídos; si se encuentran errores durante esta operación, se invoca al procedimiento *Configuración por Defecto* para establecer los valores de las principales variables del sistema.

- *Ubicar Vaso Cero:* Procedimiento invocado cuando el modo de operación es completo. Se ocupa de ubicar el vaso cero, es decir, el vaso residual con el que comienza un ensayo hasta que se cumplen las condiciones para el primer movimiento.

- *Ajustar Vaso Cero:* En el modo de operación completo, este procedimiento es invocado una vez que se ha ubicado el vaso cero, se efectúa un movimiento de ajuste para que dicho vaso quede correctamente centrado con respecto al embudo.

- *Mostrar Configuración Inicial:* Este procedimiento se invoca en la etapa final del *Setup* y se encarga de mostrar en el display LCD los valores de las principales variables. De este modo, el operador verifica que el ensayo inicie con los parámetros deseados. Este es uno de los procedimientos que se ocupa del RF N° 3. Las pantallas con esta información se encuentran en la Fig 3.

- *Log Eventos:* Procedimiento cuyo objetivo es dar cumplimiento al RF N° 8. Para ello, escribe los datos en el archivo EVENTOS.CSV residente en la tarjeta MicroSD (Fig. 4a). Es invocado al finalizar el *Setup* para registrar el arranque del dispositivo y en el *Loop* cuando resulta necesario.

- *Log Ensayos:* Procedimiento cuyo objetivo es dar cumplimiento al RF N° 4. Para ello, escribe los datos en el archivo ENSAYOS.CSV residente en la tarjeta MicroSD (Fig. 4b).

a) EVENTOS.CSV

```
FECHA;HORA;TIPO;EVENTO
21/5/2020;9:12:24;S;ENCENDIDO
21/5/2020;9:12:24;E;AHORRO POR 225 BLOQUES DE 8 SEGUNDOS
21/5/2020;9:12:44;E;ACTIVO
21/5/2020;9:12:57;D;LCD ON
21/5/2020;9:12:57;E;AHORRO POR 221 BLOQUES DE 8 SEGUNDOS
```

b) ENSAYOS.CSV

```
FECHA;HORA;LUZ;TEMP_AMB;HUM_AMB;TEMP_SUELO;HUM_SUELO;V_BAT;PORC_BAT;ROTACION;VASO
21/5/2020;9:48:49;9;20.7000007629;74.0999984741;21.0999984741;83.3480300903;12.67;90;0;0
21/5/2020;10:17:39;20;22.6000003814;67.5000000000;21.2899971008;83.1058959960;12.65;90;1;1
21/5/2020;11:30:50;12;23.3999996185;66.0999984741;22.2799987792;82.0081253051;12.59;80;1;1
21/5/2020;12:44:2;4;24.5000000000;65.3000030517;21.9499969482;84.4952392578;12.53;80;1;1
```

Fig. 4. Formato de los archivos EVENTOS.CSV y ENSAYOS.CSV.

- *Ahorrar Energía:* Procedimiento que es ejecutado cuando se dispara la interrupción al presionar el botón de encendido y apagado del display LCD. Establece el valor de una variable para determinar si el dispositivo debe o no activar el ahorro de energía hasta el próximo evento programado. Es fundamental para dar cumplimiento al RF N° 6.

- *Tiempos Generales:* Procedimiento que es invocado en muchas partes del código ya que actualiza la fecha, la hora y variables relacionadas al tiempo, las cuales son muy importantes para el funcionamiento del sistema.

- *Lectura Sensores:* Procedimiento encargado de efectuar las lecturas de los sensores y verificar el funcionamiento de los mismos.

- *Medición Batería:* Procedimiento que mediante un voltímetro efectúa la medición del voltaje de carga de la batería y determina el porcentaje de autonomía restante.

- *Rotar Vasos:* Procedimiento que controla y determina el número de rotación, el vaso actual y el siguiente, verificando que los valores se encuentren de acuerdo a lo programado. Luego de ello, invoca al siguiente procedimiento que efectúa y controla la rotación física de los vasos. Es el principal procedimiento que se ocupa del RF N° 5.

- *Rotación Física Vasos:* Procedimiento que se ocupa de controlar y accionar el funcionamiento del *Módulo Recolector*, efectuando la rotación de los vasos y controlando los movimientos necesarios para que cada vaso quede alineado al embudo.

- *Mostrar Datos LCD:* Proporciona información sobre la fecha, la hora, el modo de operación, los errores encontrados, las mediciones de los sensores y los tiempos de operación. Este es uno de los procedimientos que se ocupa del RF N° 3. Las pantallas con esta información se encuentran en la Fig 5.

Fig. 5. Pantallas del procedimiento *Mostrar Datos LCD*.

- *Guardar Configuración:* Procedimiento que hace persistentes en el archivo CONFIG.INI de la tarjeta MicroSD los cambios de los parámetros del ensayo a medida que los mismos son actualizados. Este procedimiento en combinación con *Cargar Configuración* permite dar cumplimiento a los RF N° 7 y 9.
- *Mostrar Datos Serie:* Procedimiento empleado durante el desarrollo del sistema, permite conocer los valores de las diferentes variables y efectuar la corrección de errores y la depuración del sistema.
- *Mostrar Tiempo:* Procedimiento que convierte el tiempo pasado como parámetro a una notación +/-HHH:MM:SS. Es utilizado por el procedimiento *Mostrar Datos LCD* para proporcionar al operador la información de los tiempos en una notación entendible, permitiendo controlar el número de caracteres utilizados en el display LCD.
- *Configuración por Defecto:* Procedimiento que parametriza el ensayo con valores por defecto, es invocado en el *Setup* sólo cuando el procedimiento *Cargar Configuración* ha presentado algún error.
- *Cambio Rueda:* Procedimiento asociado a la interrupción empleada al iniciar el dispositivo durante el *Setup*. Es invocado por la interrupción que se dispara cuando el sensor fotoeléctrico del *Módulo Recolector* detecta un cambio de señal de alto a bajo (FALLING). Funciona de manera complementaria con el procedimiento *Ubicar Vaso Cero,* durante la configuración inicial del dispositivo.
- *Cuenta Movimientos:* Procedimiento ligado a la interrupción que se dispara cuando el sensor fotoeléctrico del *Módulo Recolector* detecta un cambio de señal (CHANGE), luego que el vaso cero ha sido ubicado. Actúa complementariamente con los procedimientos *Ajustar Vaso Cero* y *Rotación Física Vasos*, permitiendo moverse correctamente de un vaso a otro.
- *Mover Adelante:* Procedimiento que activa el motor para que se mueva en un sentido creciente al número de vaso. Es invocado por los procedimientos *Ubicar Vaso Cero* y *Rotación Física Vasos*.
- *Mover Atrás:* Procedimiento que activa el motor para que se mueva en un sentido decreciente al número de vaso. Es invocado por el procedimiento *Ajustar Vaso Cero*.
- *Parada Total:* Procedimiento que detiene el motor. Es invocado por los procedimientos *Ubicar Vaso Cero, Ajustar Vaso Cero* y *Rotación Física Vasos.*

Cabe señalar que el RF N° 2 es abordado en el *Setup* al momento de *Iniciar componentes y sensores* y *Cargar Configuración.* Luego, durante el *Loop* en los procedimientos: *Lectura Sensores, Log Eventos, Log Sensores, Guardar Configuración, Rotación Física Vasos* y *Tiempos Generales*.

5 Ensayos de Laboratorio

Mediante el archivo CONFIG.INI (Tabla 1) el operador configura el ensayo a realizar. A su vez, a medida que el dispositivo se encuentra operativo, se actualiza el valor de ciertas variables del archivo. Más allá que durante el diseño y desarrollo las pruebas del sistema fueron una etapa insoslayable del proceso, es preciso señalar 6 ensayos de laboratorio que evaluaron el sistema bajo diferentes condiciones.

Tabla 1. Archivos CONFIG.INI con los parámetros de los ensayos realizados.

Ensayo N° 1: Con ahorro de energía en bloques de 8 s. Sin reinicio. Inicio y fin 02/05/2020.	**Ensayo N° 2:** Con ahorro de energía en bloques de 1 s. Con reinicio. Inicio 03/05/2020 y fin 04/05/2020.	**Ensayo N° 3:** Con ahorro de energía en bloques de 1 s. Con reinicio y pausa. Inicio 19/05/2020 y fin 20/05/2020.
TiempoEsperaLog;10 TiempoIntervaloLog;10 TiempoUltimoLog;0 TiempoEsperaRot;10 TiempoIntervaloRot;10 TiempoUltimaRot;0 TiempoInicioRot;0 TiempoFinRot;0 vaso;0 rotacion;0 cant_vasos;8 cant_max_rotacion;3 estadoRot;1	TiempoEsperaLog;30 TiempoIntervaloLog;1800 TiempoUltimoLog;0 TiempoEsperaRot;60 TiempoIntervaloRot;3600 TiempoUltimaRot;0 TiempoInicioRot;0 TiempoFinRot;0 vaso;0 rotacion;0 cant_vasos;8 cant_max_rotacion;3 estadoRot;1	TiempoEsperaLog;1800 TiempoIntervaloLog;3600 TiempoUltimoLog;0 TiempoEsperaRot;3600 TiempoIntervaloRot;10800 TiempoUltimaRot;0 TiempoInicioRot;0 TiempoFinRot;0 vaso;0 rotacion;0 cant_vasos;8 cant_max_rotacion;1 estadoRot;1 ModoOperacion;1
Ensayo N° 4: Con ahorro de energía en bloques de 8 s. Con reinicio y pausa. Inicio 21/05/2020 y fin 22/05/2020.	**Ensayo N° 5:** Con ahorro de energía en bloques de 1 s. Sin reinicio. Inicio 30/05/2020 y fin 31/05/2020.	**Ensayo N° 6:** Sin ahorro de energía en bloques de 1 s. Sin reinicio. Inicio 23/06/2020 y fin 24/06/2020.
TiempoEsperaLog;1800 TiempoIntervaloLog;3600 TiempoUltimoLog;0 TiempoEsperaRot;3600 TiempoIntervaloRot;10800 TiempoUltimaRot;0 TiempoInicioRot;0 TiempoFinRot;0 vaso;0 rotacion;0 cant_vasos;8 cant_max_rotacion;1 estadoRot;1 ModoOperacion;1	TiempoEsperaLog;1800 TiempoIntervaloLog;1800 TiempoUltimoLog;0 TiempoEsperaRot;3600 TiempoIntervaloRot;3600 TiempoUltimaRot;0 TiempoInicioRot;0 TiempoFinRot;0 vaso;0 rotacion;0 cant_vasos;8 cant_max_rotacion;1 estadoRot;1 ModoOperacion;2	TiempoEsperaLog;1800 TiempoIntervaloLog;3600 TiempoUltimoLog;0 TiempoEsperaRot;3600 TiempoIntervaloRot;10800 TiempoUltimaRot;0 TiempoInicioRot;0 TiempoFinRot;0 vaso;0 rotacion;0 cant_vasos;8 cant_max_rotacion;1 estadoRot;1 ModoOperacion;1

Las variables de este archivo cumplen las siguientes funciones. *TiempoEsperaLog:* Segundos a esperar para registrar los datos luego del arranque del dispositivo. *TiempoIntervaloLog:* Intervalo en segundos entre cada registro en el archivo ENSAYOS.CSV. *TiempoUltimoLog:* Tiempo de la última escritura en el archivo ENSAYOS.CSV. *TiempoEsperaRot:* Segundos a esperar para rotar los vasos luego del arranque del dispositivo. *TiempoIntervaloRot:* Segundos entre cada rotación de los vasos. *TiempoUltimaRot:* Tiempo de la última rotación de los vasos.

TiempoInicioRot: Tiempo en que se inició la rotación de los vasos. *TiempoFinRot:* Tiempo en que finalizó la rotación de los vasos. *vaso:* Número de vaso actualmente utilizado. *rotacion:* Número de rotación en curso. *cant_vasos:* Cantidad de vasos del dispositivo. *cant_max_rotacion:* Número de rotaciones (vueltas) del ensayo. *estadoRot:* Estado de la rotación (1 - ESPERA, 2 - ROTANDO, 3 - FIN, 4 - DETENIDO). *ModoOperacion:* 1 - Modo *Pitfall*, otro valor Modo *Data Logger*.

Durante la ejecución del sistema, las siguientes variables son actualizadas: *TiempoUltimoLog*, *TiempoUltimaRot*, *TiempoInicioRot*, *TiempoFinRot*, *vaso*, *rotacion* y *estadoRot*. En el modo de operación *Data Logger* sólo se consideran las variables *TiempoEsperaLog*, *TiempoIntervaloLog*, *TiempoUltimoLog* y *ModoOperacion,* las demás variables no son tenidas en cuenta.

6 Resultados

El *Ensayo N° 1* permitió comprobar el funcionamiento correcto del *MR* en combinación con el *MDL*, allí se detectaron ajustes a realizar en el procedimiento *LogEventos*. El *Ensayo N° 2* posibilitó determinar que la autonomía de la batería (12 Volts 7 Ah) no superó las 18 h, siendo necesaria una recarga o un remplazo de la misma durante su ejecución.

A partir del *Ensayo N° 3* se incorporó el parámetro *ModoOperacion* y contribuyó a comprobar que más allá de ampliar los períodos de tiempo, la autonomía de la batería continuaba sin superar las 18 h; por otra parte, cuando se interrumpía un ensayo, el período de tiempo de permanencia de un vaso comenzaba desde 0. Así, el *Ensayo N° 4* confirmó lo analizado en los 2 ensayos previos respecto a la autonomía de la batería, también permitió establecer que la autonomía no estaba relacionada respecto a la utilización de bloques de ahorro de energía de 1 s u 8 s, y posibilitó verificar que se subsanó el inconveniente detectado en el ensayo anterior, el cual ante un reinicio el período de tiempo de permanencia de un vaso comenzaba desde 0.

Mediante el *Ensayo N° 5* se comprobó el correcto funcionamiento del sistema en el modo de operación *Data Logger*. A su vez, terminó de comprobar algo detectado en los ensayos previos, relacionado a que el uso reiterado de la función *LowPower.powerDown* de la biblioteca *LowPower.h*, si bien contribuye a una mayor autonomía de la batería, tiene como desventaja que introduce un margen de tiempo superior entre 6 y 7 min en los intervalos temporales de rotación y registro de datos. En tanto, en el *Ensayo N° 6* se corroboró que sin la funcionalidad de ahorro de energía el sistema sólo cuenta con una autonomía de 15 h; no obstante, los intervalos temporales de rotación y registro de datos son más precisos, con un margen de +/- 6 s.

7 Conclusiones y Trabajo Futuro

Ha sido posible diseñar y desarrollar un sistema embebido para un prototipo automatizado de trampa de caída por tiempo, dotado de un data logger parametrizable y más completo, con sensores de luz, temperatura y humedad relativa ambiente, y

temperatura y humedad del suelo. Los ensayos fueron satisfactorios y permitieron la evolución del sistema, cumpliendo los requerimientos funcionales. El diseño arquitectónico contribuye al mantenimiento y evolución del dispositivo, resultando factible su utilización en estudios de biodiversidad de artrópodos de suelo.

El enfoque de ahorro de energía utilizado si bien contribuye a preservar la autonomía de la batería, también conlleva a la pérdida de exactitud en los tiempos. En consecuencia, este aspecto tendrá que ser revisado.

Como instancia posterior, se tiene previsto dotar al *Módulo de Alimentación* de una batería de mayor capacidad para prolongar la autonomía, a partir de ello, realizar ensayos de campo en diferentes ambientes productivos de la EEA Concordia del INTA, lo cual abre la posibilidad de efectuar comparaciones con dispositivos similares previamente desarrollados y, a su vez, realizar investigaciones de artrópodos de suelo mediante un abordaje novedoso que utilice herramientas estadísticas y Minería de Datos.

Referencias

1. Hohbein, R.R., Conway, C.J.: Pitfall traps: A review of methods for estimating arthropod abundance. Wildl. Soc. Bull. 42, 597-606 (2018). https://doi.org/10.1002/wsb.928
2. Dahl, F.: Vergleichende Untersuchungen Über die Lebensweise wirbelloser Aasfresser. Sitzungberichte – K. Preuss. Akad. der wissenschaften. 17-30 (1896)
3. Greg Sherley and Ian Stringer: DOCCM-248862 Invertebrates: pitfall trapping v1. 0. Dep. Conserv. Te Papa Atawhai, Wellingt. 1-30 (2016)
4. Brown, G.R., Matthews, I.M.: A review of extensive variation in the design of pitfall traps and a proposal for a standard pitfall trap design for monitoring ground-active arthropod biodiversity. Ecol. Evol. 6, 3953-3964 (2016). https://doi.org/10.1002/ece3.2176
5. Knapp, M., Saska, P., Knappová, J., Vonička, P., Moravec, P., Kůrka, A., Anděl, P.: The habitat-specific effects of highway proximity on ground-dwelling arthropods: Implications for biodiversity conservation. Biol. Conserv. 164, 22-29 (2013). https://doi.org/10.1016/j.biocon.2013.04.012
6. Buchholz, S.: Design of a time-sorting pitfall trap for surface-active arthropods. Entomol. Exp. Appl. 133, 100-103 (2009). https://doi.org/10.1111/j.1570-7458.2009.00902.x
7. McMunn, M.S.: A time-sorting pitfall trap and temperature datalogger for the sampling of surface-active arthropods. HardwareX. 1, 38-45 (2017). https://doi.org/10.1016/j.ohx.2017.02.001
8. Wickert, A.D., Sandell, C.T., Schulz, B., Ng, G.C.: Open-source Arduino-compatible data loggers designed for field research. 2065-2076 (2019)
9. Baker, E.: Open source data logger for low-cost environmental monitoring. Biodivers. Data J. 2, (2014). https://doi.org/10.3897/BDJ.2.e1059
10. Sommerville, I.: Software Embebido. En: Ingeniería de Software. pp. 537-564. Pearson Educación S.A. (2011)
11. Jacobson, I., Booch, G., Rumbaugh, J.: El Proceso Unificado de Desarrollo de Software, (2000)
12. Arduino AG: What is Arduino?, https://www.arduino.cc/en/Guide/Introduction