



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

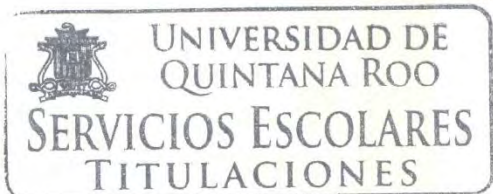
**IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE
DATOS PARA EL MONITOREO DE PARÁMETROS
AMBIENTALES SUBACUÁTICOS BASADOS EN UNA
ARQUITECTURA DE HARDWARE ABIERTO DE PLACA ÚNICA**

**TRABAJO DE TESIS
PARA OBTENER EL GRADO DE
INGENIERA EN REDES**

**PRESENTA
AIDA ELENA RIVAS CONTRERAS**

**DIRECTOR DE TESIS
MTI. VLADIMIR VENIAMIN CABAÑAS VICTORIA**

**ASESORES
DR. JAVIER VÁZQUEZ CASTILLO
DR. ALBERTO PEREIRA CORONA
MSI. LAURA YÉSICA DÁVALOS CASTILLA
MTI. MELISSA BLANQUETO ESTRADA**



CHETUMAL QUINTANA ROO, MÉXICO, OCTUBRE DE 2017



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

**TRABAJO DE TESIS ELABORADO BAJO SUPERVISIÓN DEL COMITÉ
DE ASESORÍA Y APROBADO COMO REQUISITO PARCIAL PARA
OBTENER EL GRADO DE:
INGENIERA EN REDES**

COMITÉ DE TRABAJO DE TESIS

DIRECTOR:



MTI. VLADIMIR VENIAMIN CABAÑAS VICTORIA

ASESOR:

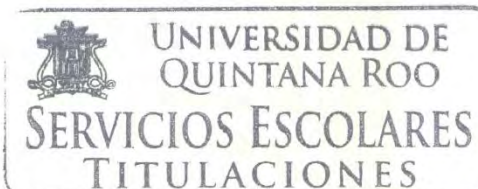


DR. JAVIER VÁZQUEZ CASTILLO

ASESOR:



DR. ALBERTO PEREIRA CORONA



Agradecimientos

A:

Mi papa y mi mama que desde pequeña me inculcaron que la educación es lo más importante que uno puede tener. Le agradezco a mi papa por nunca quejarse por todas las horas que trabajaba y por darme todo lo que le he necesitado. A mi mama por siempre asegurarse que tenía lo necesario y por siempre estar en todo evento en el que participaba, siendo siempre la que gritaba más fuerte. Les agradezco ambos por el amor incondicional y por enseñarme valores y morales.

Mi hermano por un pequeño nivel de paciencia que ayudo a no matarnos en estos años viviendo juntos. Le agradezco a mi hermana por los buenos viajes que hemos hecho y por presionarme todas las semanas en terminar mi tesis. A mi hijo/perro, Lord Bartholomew Bass, por siempre hacerme compañía y recibirme con tanto amor cuando llegaba a casa.

Mis abuelos por siempre quererme tanto, cada uno de su manera. A mi abuelita Marina por las semanas que venía a estar conmigo al año y siempre feliz de ayudar en lo que podía. Mi abuelita Nana por ser como otra madre para mí en los veranos y navidades que iba a El Salvador. Por todos los desvelos y ruido que aguantaste cuando la casa estaba llena de nietos. Le agradezco infinitamente a mi abuelito Caco que desde pequeña se preocupó que me aprendiera mi tabla de multiplicaciones y las reglas básicas de acentuación. Le doy gracias ya que, por él, me encantaban las matemáticas y por esa razón no me preocupé al entrar en una carrera de ingeniería. Al igual le agradezco a mi abuelo por nunca poner peros cuando le pedía que me consiga una partida de nacimiento actual para cualquier trámite que necesitaba aquí en México. Al resto de mi familia, tíos, tías, primos, primas, les agradezco por toda la atención y amor que siempre me han dado.

Mis amigos, Oscar Manzanilla, Luis Rodríguez, Ing. Araceli de la Cruz, Jairo Courtenay, Alfred Yam, Byron Sierra, Mauricio Morfin y el Ing. Walter Rivera, por todos esos momentos increíbles en la universidad. Por esas noches de desvelo trabajando en la tesis o en cualquier otro proyecto final. En especial los viajes que se hicieron y las fiestas en donde todo convivíamos sin pensar en el estrés de la Universidad. A todos mis compañeros de la carrera y amigos de otras carreras, Gerardo Tapia, les agradezco por toda ayuda académica o emocional sin

requerir algo a cambio. A mi mejor amiga, Hannah Montero, por esas conversaciones que me ayudaban a desestresarme y siempre ayudarme en cualquier problema, aunque estuviera lejos. A mi gran amigo de la infancia, Reni Reyes, que siempre me escribía de vez en cuando para preguntar como estoy y siempre hacerme reír.

La familia Reyes Ramos, por siempre visitar y hacerme sentir como que estoy en casa. Gracias por todas las noches de taquiza y doughnuts. Les agradezco por cada momento que pasamos aquí en Chetumal.

La Universidad de Quintana Roo por una gran experiencia académica en donde están bienvenidos estudiantes de todas partes del mundo. A la Dirección General de Cooperación Académica que me dio la oportunidad de hacer un intercambio nacional a la ciudad de Guadalajara. A todos los maestros que me ayudaron en mi formación profesional, les estoy muy agradecida.

La Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e Ingeniería y al maestro José Luis David Bonilla Carranza, por los tutoriales de Arduino que me ayudaron para iniciar mi proyecto de tesis.

Mi asesor de tesis, el maestro Vladimir Veniamin Cabañas Victoria por su constante ayuda. Sus conocimientos y dedicación hacia su trabajo me aseguraron su apoyo completo en mi proyecto. Le agradezco por nunca perder la confianza en mí y por guiarme, no solo en este proyecto, sino que también en mi formación académica.

Dedicatoria

A:

Todas las mujeres en ingeniería que a veces se sienten como que no encajan por ser tan pocas y al final del día eres *“one of the guys”*.

Los estudiantes extranjeros que se les dificulte acomodarse a un nuevo país, ciudad, y gente.

La carrera en Ingeniería en Redes, para que cualquier estudiante que necesite ayuda en este tema se pueda basar en este proyecto.

Mi esfuerzo y dedicación en este proyecto.

Resumen

El presente proyecto consiste en la implementación de un sistema de placa única para la adquisición de datos (parámetros ambientales como por ejemplo la temperatura) utilizando un Seeeduino Stalker v3.0 programado a través de Arduino UNO, tomando en cuenta la necesidad de que el sistema sea autosuficiente en el tema de la energía, por lo que se incorpora una batería recargable y un panel solar.

En el capítulo 1 se describe el contexto de la aplicación del sistema que se implementó para monitorear algunos parámetros que biólogos e investigadores podrían encontrar útiles para sus proyectos.

El capítulo 2 contiene los tópicos relevantes para el desarrollo de este trabajo: los componentes del sistema de adquisición de datos, su configuración e información técnica.

El capítulo 3 describe el desarrollo del trabajo, desde la configuración del sistema de placa única, su programación, la incorporación de sensores hasta la visualización del registro de la información.

El capítulo 4 contiene los resultados de la implementación del sistema de adquisición de datos y un análisis estadístico de las medidas tomadas.

El capítulo 5 contiene las conclusiones del proyecto basada en las pruebas y análisis del funcionamiento del sistema de adquisición de datos.

El capítulo 6 describe algunas posibilidades a futuros del sistema de adquisición de datos con la ayuda de dispositivos complementarios que se pueden agregar al sistema de placa única.

Contenido

Capítulo 1 Introducción.....	1
1.1 Introducción	1
1.2 Definición del problema:.....	1
1.3 Justificación	1
1.4 Objetivos	2
1.5 Metodología:	2
1.5.1 Ensamblado Físico:	2
1.5.2 Ensamblado Lógico:	2
1.5.3 Pruebas:	3
Capítulo 2 Marco Teórico	4
2.1 Sistema de Adquisición de Datos para el Monitoreo de Parámetros Ambientales.....	4
2.2 Sistemas de Adquisición de Datos en el Mercado	4
2.2.1 Termopar DI-808	4
2.2.2 Graphtec MT100.....	5
2.3 Sistemas de Adquisición de Datos basados en sistemas informáticos de placas Única.	5
2.4 Sistemas de cómputo de placa única.....	6
2.4.1. Arduino UNO	6
2.4.2 Seeeduino Stalker v3.0	6
2.4.3 Nanode Classic	7
2.4.4 Waspote	8
2.4.5 BeagleBone Black	9
2.4.6 Orange Pi One.....	9
2.4.7 Comparación de Especificaciones de Placas	10
2.5 Entorno de Desarrollo Integrado	11
2.5.1 Software Arduino	12
2.5.2 NetBeans.....	12
2.5.3 Eclipse.....	13
2.5.4 Code::Blocks	13
2.6 Sensores.....	14
2.6.1 Sensor de Temperatura de Termistor.....	14
2.6.1.1 Especificaciones del Sensor de Temperatura Grove	15

2.6.2 Sensor de Temperatura de Termopar	16
2.6.3 Sensor de Temperatura de Infrarrojos.....	16
2.6.4 Termómetro de Alcohol	16
2.7 Baterías.....	17
2.7.1 Batería de Polímero de Iones de Litio	17
2.7.2 Batería de Níquel-Hidruro Metálico	17
2.7.3 Batería de Níquel-Cadmio	18
2.8 Panel Solar	18
2.8.1 Panel Monocristalino	18
2.8.2 Panel Policristalino	19
2.9 Almacenamiento	20
Capítulo 3 Desarrollo.....	21
3.1 Fases	21
3.1.1 Componentes (Ensamblado lógico y físico)	21
3.1.2 Ensamblado lógico	22
3.1.2.1 Conexión de batería y de panel solar	22
3.1.2.2 Conexión de sensor de temperatura	22
3.1.3 Ensamblado físico	23
3.1.3.1 Conexión de batería y panel solar	23
3.1.3.2 Conexión de sensor de temperatura	23
3.1.4 Programación	24
3.1.4.1 Conexión puente para programación de Seeeduino Stalker v3.0	24
3.1.4.2 Conexión puente para programación de Seeeduino Stalker v3.0	25
3.2 Código:.....	25
Capítulo 4 Resultados	26
4.1 Prueba 1	26
4.1.1 Código cada minuto:.....	26
4.2 Prueba 2	29
4.2.1 Código cada 30 minutos:.....	29
4.3 Prueba 3	31
4.3.1 Código cada 60 minutos:.....	31
Capítulo 5 Conclusiones.....	33

Capítulo 6 Alcance	35
Referencias	36
Apéndices.....	40
Apéndice 1	40
Apéndice 2	43
Apéndice 3	46
Apéndice 4	49

Índice de Ilustraciones

Ilustración 1 Termopar DI-808 [5]	4
Ilustración 2 Graphtec MT100 [6]	5
Ilustración 3 Arduino UNO [9]	6
Ilustración 4 Partes de Seeeduino Stalker v3.0 [10]	7
Ilustración 5 Nanode Classic [13]	8
Ilustración 6 Waspote [14]	8
Ilustración 7 BeagleBone Black [15]	9
Ilustración 8 Orange Pi One [16]	9
Ilustración 9 Plataforma de Software Arduino	12
Ilustración 10 NetBeans	12
Ilustración 11 Eclipse	13
Ilustración 12 Code::Blocks	14
Ilustración 13 Sensor de Temperatura Grove [21]	15
Ilustración 14 Termómetro de Alcohol	16
Ilustración 15 Batería de Polímero de Iones de Litio	17
Ilustración 16 Batería de Níquel-Hidruro Metálico	17
Ilustración 17 Batería de Níquel-Cadmio	18
Ilustración 18 Panel Monocristalino	19
Ilustración 19 Panel Policristalino	19
Ilustración 20 Tarjeta microSD	20
Ilustración 21 Sistema de Adquisición de Datos	21
Ilustración 22 Cableado Batería y Panel Solar	23
Ilustración 23 Cableado Sensor de Temperatura Grove	23
Ilustración 24 Cableado Arduino UNO a Seeeduino Stalker v3.0	24

Índice de tablas

Tabla 1 Comparación de Especificaciones de Placas Computadoras [15] [16]	10
Tabla 2 Comparación de Especificaciones de Placas Microcontroladoras [9] [10] [14] [12]	11
Tabla 3 Especificaciones Sensor de Temperatura Grove v1.2 [21]	15
Tabla 4 Prueba de toma de temperatura cada minuto y desviación estándar	29
Tabla 5 Prueba de toma de temperatura cada 30 minutos y desviación estándar	30
Tabla 6 Prueba de toma de temperatura cada 60 minutos y desviación estándar	32

Índice de graficas

Grafica 1 Guía de Metodología	3
Grafica 2 Prueba de toma de temperatura cada minuto	29
Grafica 3 Prueba de toma de temperatura cada 30 minutos	31
Grafica 4 Prueba de toma de temperatura cada 60 minutos	32

Índice de diagramas

Diagrama 1 Cableado Batería y Panel Solar	22
Diagrama 2 Cableado Sensor de Temperatura Grove	22
Diagrama 3 Cableado Arduino UNO a Seeeduino Stalker v3.0.....	25

Glosario

DAQ	Data Acquisition	<i>Adquisición de Datos</i>
PC	Personal Computer	<i>Computadora Personal</i>
SD	Secure Digital	
IDE	Integrated Development Environment	<i>Entorno de Desarrollo Integrado</i>
PWM	Pulse-Width Modulation	<i>Modulación por Ancho de Pulsos</i>
USB	Universal Serial Bus	<i>Bus Universal en Serie</i>
ICSP	In-Circuit Serial Programming	<i>Programación serial en circuito</i>
E/S		<i>Entrada/Salida</i>
DC	Direct Current	<i>Corriente Directa</i>
SRAM	Static Random Access Memory	<i>Memoria Estática de Acceso Aleatorio</i>
EEPROM	Electrical Erasable Programmable Read-Only Memory	<i>Memoria Eléctricamente Borrable Programable de Solo Lectura.</i>
MCU	Microcontroller Unit	<i>Microcontrolador</i>
RTC	Real Time Clock	<i>Reloj en Tiempo Real</i>
JST	Japan Solderless Terminal	<i>Terminal Sin Soldaduras Japón</i>
PCB	Printed Circuit Board	<i>Placa de Circuito Impreso</i>
UART	Universal Asynchronous Receiver-Transmitter	<i>Transmisor-Receptor Asíncrono Universal</i>
I ² C	Inter-Integrated Circuit	<i>Circuito Interintegrado</i>
SPI	Serial Peripheral Interface	<i>Interfaz de Periféricos Serie</i>
AC	Alternating Current	<i>Corriente Alterna</i>
DC	Direct Current	<i>Corriente Directa</i>

Capítulo 1 Introducción

1.1 Introducción

La adquisición de datos (DAQ) es el proceso de medir con una PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido. Un sistema DAQ consiste en sensores, hardware de medidas DAQ y una PC con software programable. Los sistemas DAQ basados en PC aprovechan la potencia del procesamiento, la productividad, la visualización y las habilidades de conectividad de las PCs estándares en la industria proporcionando una solución de medidas más potente, flexible y rentable. [1]

1.2 Definición del problema:

El monitoreo ambiental generalmente se basa en un sistema continuo de observación de medidas y evaluaciones para propósitos definidos; es una herramienta importante en el proceso de evaluación de impactos ambientales y en cualquier programa de seguimiento y control. [2]
[3]

Se necesitará un dispositivo capaz de hacer un monitoreo autónomo, adaptable a diferentes ambientes.

1.3 Justificación:

Monitorear diversos parámetros ambientales tiene como objetivo determinar cuáles y como se encuentran el estado de las cosas en materia ambiental de un entorno. Resulta ser una actividad de gran ayuda en lo que respecta al cuidado del medio ambiente. Esta propuesta pretende ser un sistema de adquisición de datos económico, móvil, autosuficiente y de largo despliegue que permita monitorear temperatura, grado de acidez o basicidad de una solución acuosa, oxígeno disuelto en agua, entre otros.

1.4 Objetivos

General:

Implementar un sistema de adquisición de datos para hacer un dispositivo energéticamente autónomo, que adquiera y almacene datos de parámetros ambientales para el análisis de la información de un ambiente en específico.

Específicos:

- i. Diseño
- ii. Construcción
- iii. Prueba

1.5 Metodología:

La metodología de este proyecto consistió en el ensamblado físico y lógico del sistema de adquisición de datos para posteriormente realizar algunas pruebas de mediciones.

1.5.1 Ensamblado Físico:

Para el ensamblado físico se conectó el Seeeduino Stalker v3.0 un sensor, en este caso de temperatura, para las mediciones y también una batería y panel solar. Finalmente se insertó la tarjeta microSD para el almacenamiento de datos que se registraron.

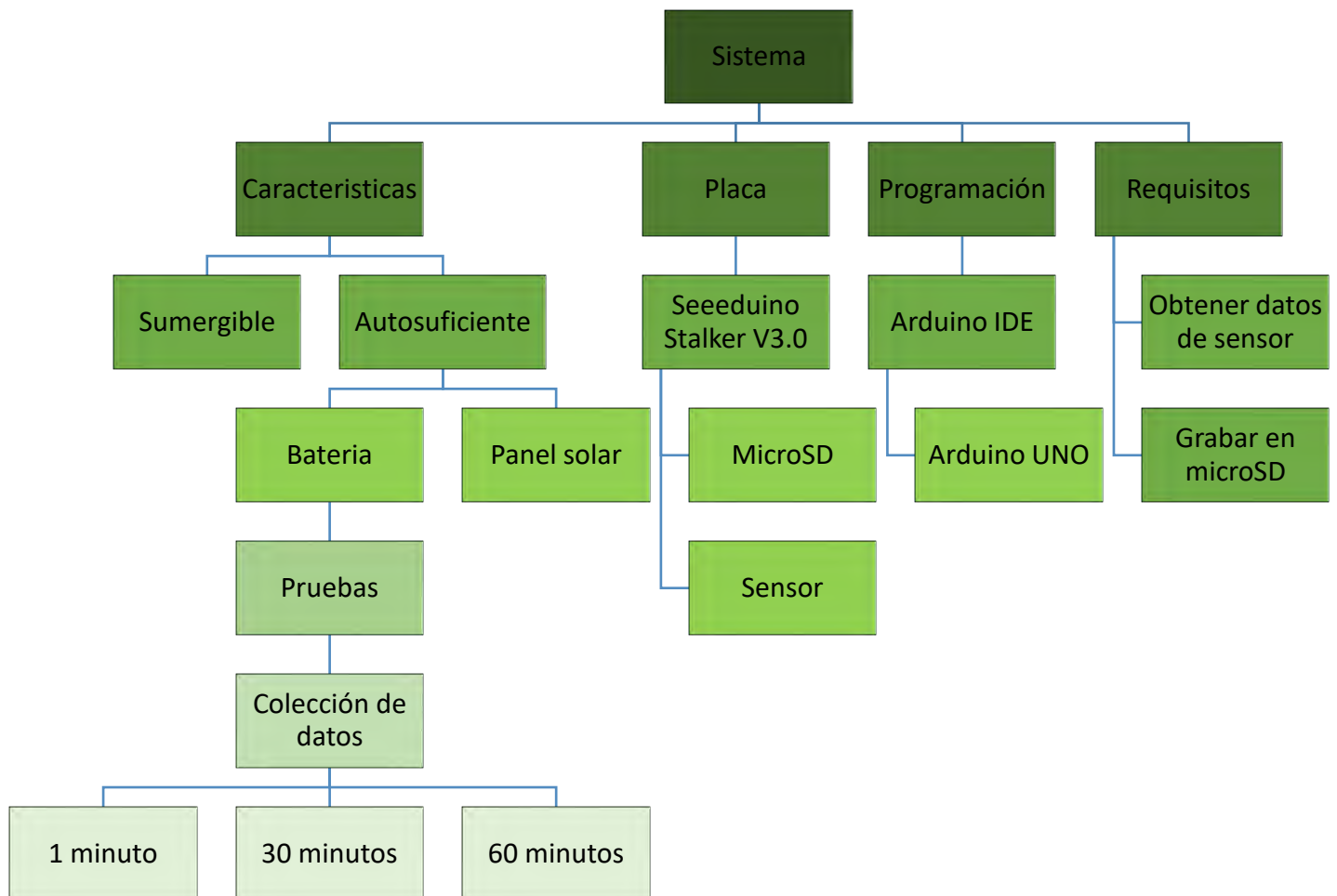
1.5.2 Ensamblado Lógico:

El ensamblado lógico consistió en la programación del Seeeduino Stalker v3.0. Para esto se utilizó el software de Arduino IDE. Primeramente, se conectó a un Arduino UNO para usar como programador y este a la computadora. La programación consistió en encender la tarjeta microSD, y se definió la fecha y hora actual. Luego se hizo una lectura del sensor y de la fecha y hora. Finalmente, estos datos se guardaron en la tarjeta microSD.

1.5.3 Pruebas:

Se tomaron pruebas de rendimiento de batería del sistema de adquisición de datos. Esto se basó en la duración de la batería con respecto al tiempo de toma de medición del sensor definido en cada prueba, esto será de 1, 30 y 60 minutos.

Se midió la temperatura con un termómetro de alcohol a la misma vez que el sistema de adquisición de datos tomaba medidas para comprobar que las mediciones sean correctas. Se comparo 60, 12 y 12 mediciones de las pruebas de 1, 30 y 60 minutos respectivamente.



Grafica 1 Guía de Metodología

Capítulo 2 Marco Teórico

2.1 Sistema de Adquisición de Datos para el Monitoreo de Parámetros Ambientales

Para llevar a cabo el sistema de adquisición de datos, se usó un Arduino UNO como programador, software de Arduino (IDE), Seeeduino Stalker V3.0, sensor de temperatura Grove v1.2, batería y panel solar y una tarjeta microSD.

La placa funciona como un puente en donde por un lado captura información del sensor y por otro graba la información adquirida en la tarjeta microSD.

Las placas de adquisición de datos se encargan de:

- Las conversiones de señales de analógica a digital.
- La comunicación con la PC.

Una parte fundamental en todo sistema de adquisición de datos es el elemento encargado de percibir la magnitud a medir. Los sensores informan de su entorno y además esa información es cuantificable, es decir, medible por algún instrumento. [4]

2.2 Sistemas de Adquisición de Datos en el Mercado

2.2.1 Termopar DI-808

Sistema de adquisición de datos de termopar y pulsos de ocho canales. Este dispositivo incorpora un servidor web. Almacena información en una memoria flash y/o a cualquier servidor remotamente accesible o correo electrónico. [5]



Ilustración 1 Termopar DI-808 [5]

2.2.2 Graphtec MT100

Sistemas de adquisición de datos de 10 canales de entrada análoga para voltaje, corriente o temperatura. Maneja 4 entradas lógicas de pulsos y almacena los datos en unidades USB Flash. Ofrece conexión ethernet para archivar automáticamente datos. [6]



Ilustración 2 Graphtec MT100 [6]

2.3 Sistemas de Adquisición de Datos basados en sistemas informáticos de placas Única.

Recientemente estudiantes del Tecnológico de Chetumal implementaron una red de sensores para el monitoreo en tiempo real de la calidad del agua en estanques de cultivo de tilapia. Conformando la red por nodos basados en tarjetas electrónicas Arduino y sensores para medir pH, temperatura, conductividad eléctrica y potencial de óxido-reducción. Con una comunicación inalámbrica, los nodos usan el protocolo ZigBee que permite la reconfiguración dinámica de la red en caso de falla. Miden nivel de acidez en el agua con un sensor de pH y miden la temperatura con un sensor de temperatura. Las lecturas de los sensores se almacenan en una base de datos MySql. [7]

2.4 Sistemas de cómputo de placa única

2.4.1. Arduino UNO

Arduino Uno es una placa microcontroladora basada en el ATmega328P. Plataforma abierta y versátil para el desarrollo de productos electrónicos muy enfocada a un público de no expertos. [8] Cuenta con 14 entradas y salidas digitales y 6 entradas analógicas. Tiene una memoria flash de 32 kB. [9]



Ilustración 3 Arduino UNO [9]

2.4.2 Seeeduino Stalker v3.0

Seeeduino Stalker v3.0 es una placa microcontroladora de código abierto con un nodo de red de sensores inalámbricos compatibles con Arduino, especialmente útil para aplicaciones de registro de datos en exteriores. Tiene un procesador ATmega328 con frecuencia 8MHz. Cuenta con 14 entradas y salidas digitales y 6 entradas analógicas. [10]

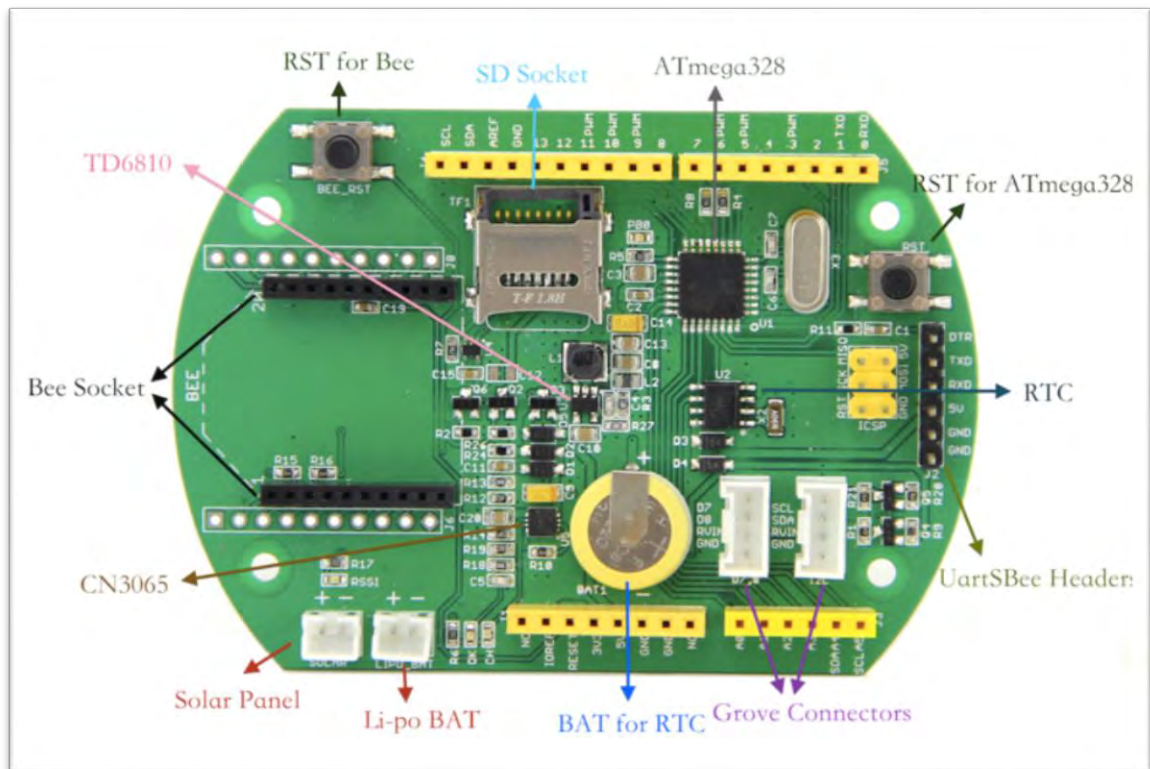


Ilustración 4 Partes de Seeduino Stalker v3.0 [10]

2.4.3 Nanode Classic

Nanode Classic es una placa microcontrolador de código abierto [11] que permite a este conectarse a Internet a través de un API y puedes incluso utilizarlo como servidor de páginas web simples permitiendo al usuario configurar el dispositivo. [8] Tiene un procesador ATmega328P con una frecuencia de 16MHz. Maneja una SRAM de 32kB. [12]

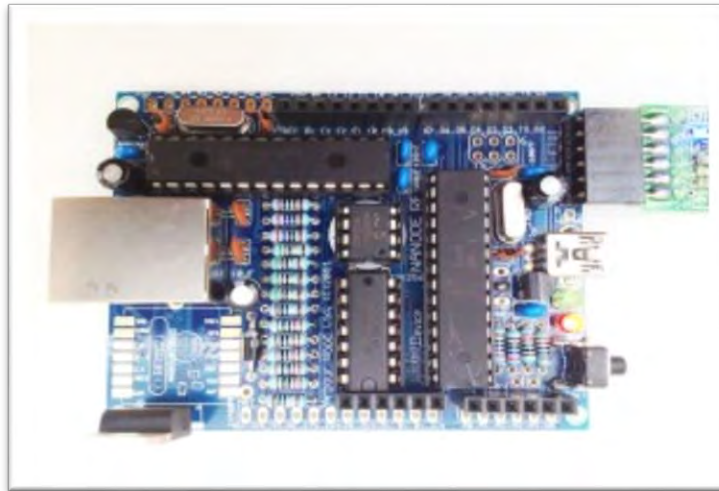


Ilustración 5 Nanode Classic [13]

2.4.4 Wasmote

Wasmote es una placa microcontroladora de código abierto diseñada para crear redes inalámbricas de sensores con unos requerimientos bastantes específicos y destinados a ser desplegados en un escenario real. Siendo un dispositivo específico para el internet de las cosas. [8] Tiene un procesador ATmega1281 con frecuencia de 32 kHz. Maneja una memoria flash de 128kB y cuenta con 8 entradas y salidas digitales y 7 entradas analógicas.

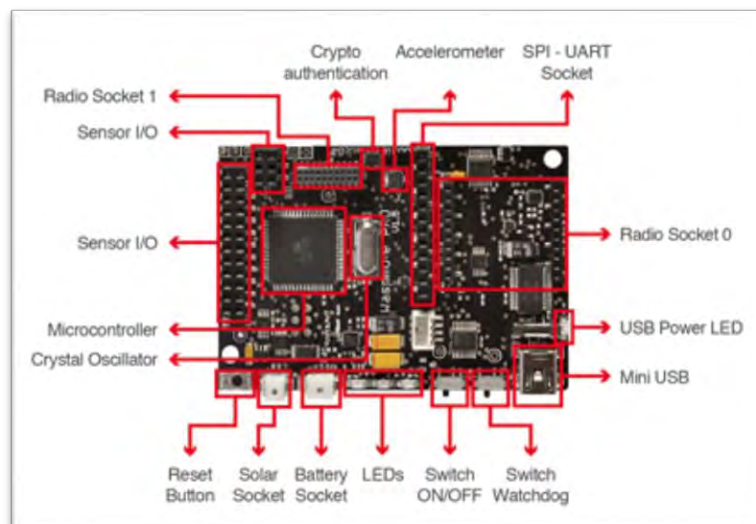


Ilustración 6 Wasmote [14]

2.4.5 BeagleBone Black

Una placa de hardware libre y código abierto de bajo consumo. Originalmente desarrollada como una placa educativa. [15] BeagleBone, está diseñado para funcionar a un nivel mucho más alto y tiene mucha más capacidad de proceso que Arduino. [8] Es una placa con procesador ARM Cortex-A8 de frecuencia 1GHz y un núcleo. Contiene interfaces USB y ethernet. [15]

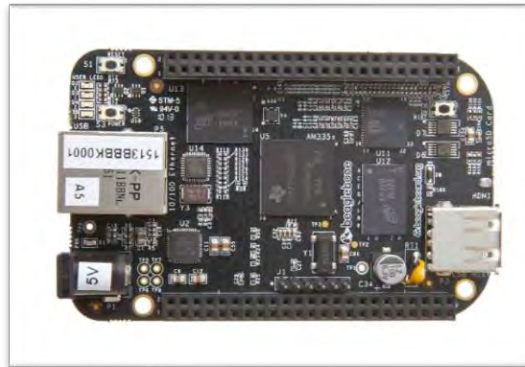


Ilustración 7 BeagleBone Black [15]

2.4.6 Orange Pi One

Placa de hardware abierto. Puede ejecutar Android 4.4, Ubuntu Debian, Raspbian Image, así como el Banana Pi Image. Es una placa con procesador ARM Cortex-A7 de frecuencia 1.2GHz y 4 núcleos. Contiene interfaces USB y ethernet alámbrica e inalámbrica y una ranura de almacenamiento microSD. [16]

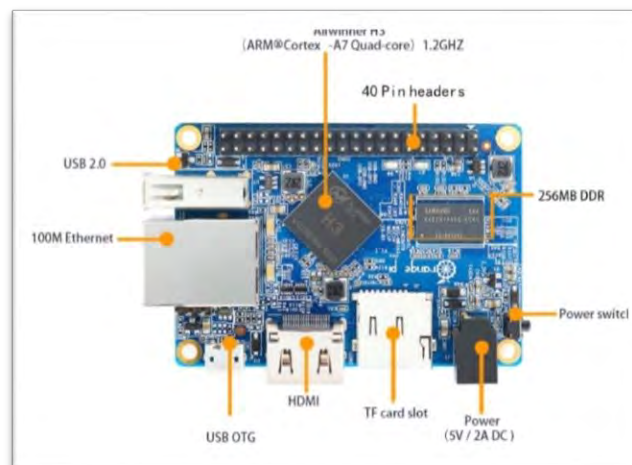


Ilustración 8 Orange Pi One [16]

2.4.7 Comparación de Especificaciones de Placas

			ORANGE PI ONE	BEAGLEBONE BLACK
CPU, GPU y Memoria	CPU	Arquitectura	ARM Cortex-A7	ARM Cortex-A8
		Núcleo	4	1
		Frecuencia	1.2 GHz	1 GHz
	GPU		ARM Mali-400 MP2 @600 MHz	PowerVR SGX530
	RAM	Tamaño	512 MB	512 MB
		Velocidad de Datos (MT/s)		
		Tamaño de ruta de datos (bits)	32	16
Tipo		DDR3	DDR3L	
Interfaces E/S y Puertos	PCIe		No	No
	USB	2.0	4	1
		3.0	No	No
		Dispositivo	OTG	Si
	Almacenamiento	En-Placa	No	4 GiB eMMC
		Ranuras flash	microSD	microSD
		SATA	No	No
	Redes	Ethernet	10/100	10/100
		Wi-Fi	b/g/h (RTL8189FTV)	No
	Comunicaciones	Bt.	No	No
		PC	Si	Si
		SPI	Si	Si
	I/O Genérico	GPIO	Si	66
		Análogo		12-bit ADC
Otras Interfaces		CSI, UART	CAN, UART	
Interfaces Audiovisual	Entrada MIC		nO	No
	Salida Audio		HDMI	No
	HDMI		1.x	No
	LVDS		No	No
	Otra Salida Video		Video compuesto	No
Sistema Operativo	Linux		Parcial	Si
	Android		Si	Si
	BSD		Si FreeBSD	Si OpenBSD
	Windows		No	Si CE 6, WEC 7
	Otro			Minix
Eléctrico	Voltaje Entrada		5 V	5 V
	Consumo de Energía (inactivo)			1.05 W
	Consumo de Energía (Maximo)		10 W	2.3 W
	Fuente de Poder		Conector DC de 4.0 mm o cabecera GPIO	miniUSB o conector DC o cabecera de expansión

Tabla 1 Comparación de Especificaciones de Placas Computadoras [15] [16]

		Arduino UNO	Seeeduino Stalker v3.0	Nanode Classic	Waspote
Procesador	Procesador	ATmega328P	ATmega328	ATmega328P	ATmega1281
	Frecuencia	16 MHz	8MHz	16 MHz	32 kHz
Voltaje		5V	3.3V	3.3V	3.3V
Flash (kB)		32	32		128
EEPROM (kB)		1			4
SRAM (kB)		2		32	8
E/S	E/S Digital (pins)	14	14		8
	E/S Digital con PWM (pins)	6	1 JST 1 USB		2 UARTs 1 I2C 1 SPI 1 USB
	Entrada Analógica (pins)	6	6		7

Tabla 2 Comparación de Especificaciones de Placas Microcontroladoras [9] [10] [14] [12]

2.5 Entorno de Desarrollo Integrado

Un IDE, es un software que contiene las herramientas básicas que necesita un desarrollador para programar y probar un software. Normalmente, contiene un editor de código, un compilador o un depurador que el desarrollador accede a través de una única interfaz gráfica de usuario (GUI). [17]

2.5.1 Software Arduino



Ilustración 9 Plataforma de Software Arduino

2.5.2 NetBeans

IDE gratis de código abierto. Principalmente usado para la creación de aplicaciones Java, pero soporte otros lenguajes como C, C++, Fortran, HTML 5, Java, PHP y más. [18]

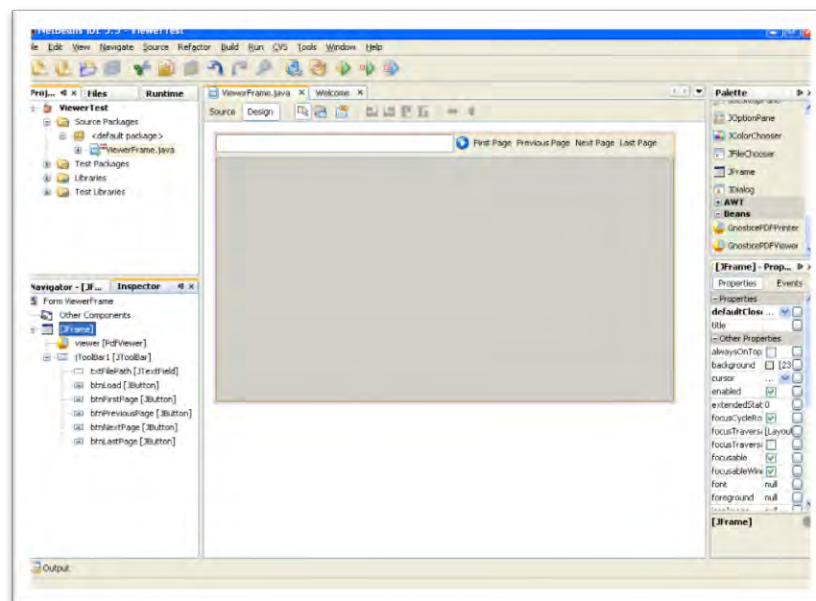


Ilustración 10 NetBeans

2.5.3 Eclipse

IDE gratuito con un editor de código abierto útil para principiantes y expertos. Originalmente para Java, pero ahora soporta lenguajes como C, C++, Perl, PHP, Ruby y más. [18]

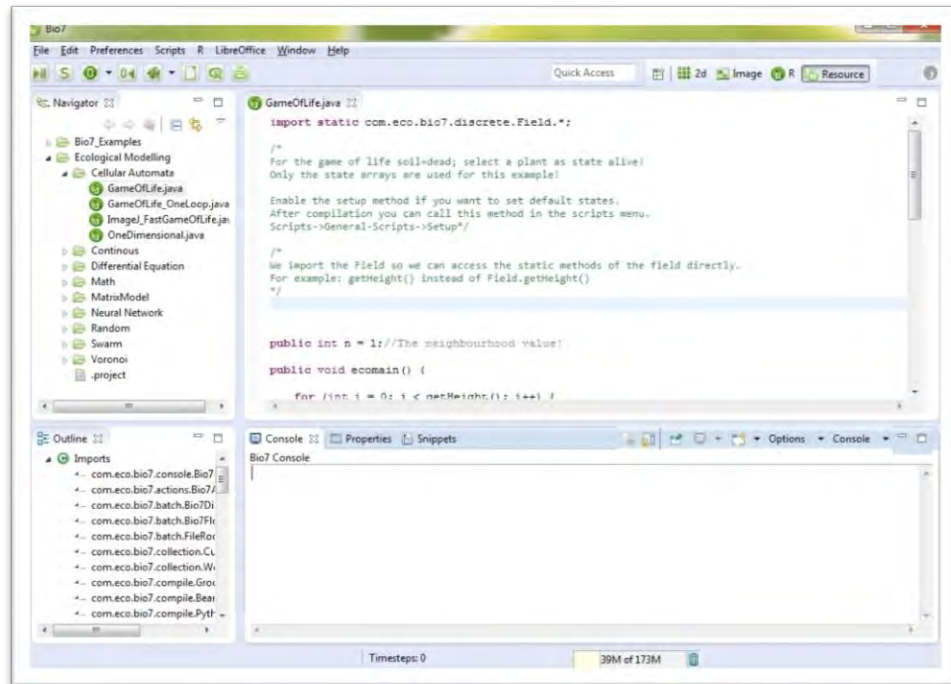


Ilustración 11 Eclipse

2.5.4 Code::Blocks

IDE gratis para código abierto. Funciona muy bien en toda plataforma y acepta lenguajes como C, C++ y Fortran. [18]

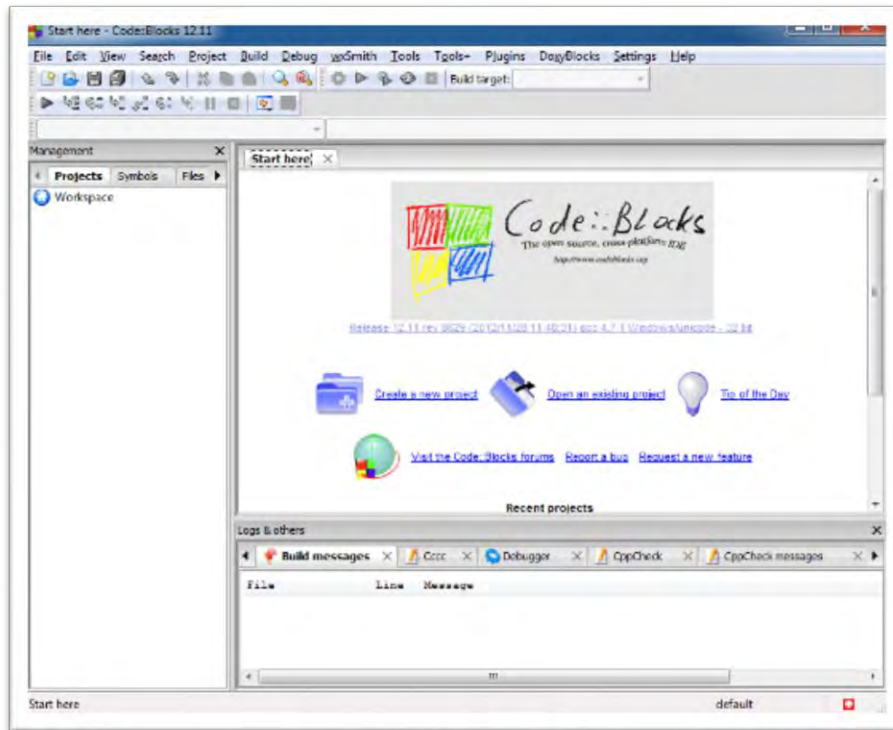


Ilustración 12 Code::Blocks

2.6 Sensores

Los sensores convierten un fenómeno físico en una señal eléctrica que se puede medir. Dependiendo del tipo de sensor, su salida eléctrica puede ser un voltaje, corriente, resistencia u otro atributo eléctrico que varía con el tiempo. [19] Realizan una conversión de energías y suministran información sobre el estado y tamaño de la magnitud. [4]

2.6.1 Sensor de Temperatura de Termistor

Compuestos de una mezcla sintetizada de óxidos metálicos, el termistor es esencialmente un semiconductor que se comporta como un resistor térmico. [20]

El sensor de temperatura Grove utiliza un termistor para detectar la temperatura ambiente. La resistencia de un termistor aumentará cuando la temperatura ambiente disminuya. Es esta característica que utilizamos para calcular la temperatura ambiente. [21] Por lo cual, cuando la temperatura sea menor habrá una corriente más alta y se consumirá más energía de la batería.

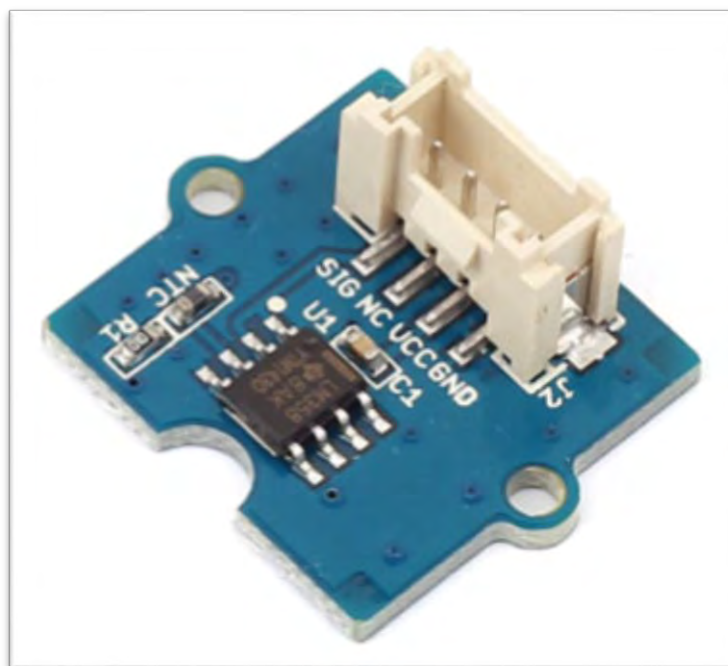


Ilustración 13 Sensor de Temperatura Grove [21]

2.6.1.1 Especificaciones del Sensor de Temperatura Grove

Parámetros	Valor
Voltaje	3.3 ~ 5V
Resistencia de potencia cero	100 K Ω
Tolerancia de resistencia	$\pm 1\%$
Rango de temperatura de funcionamiento	-40 ~ +125 °C
Constante B nominal	4250 ~ 4299K

Tabla 3 Especificaciones Sensor de Temperatura Grove v1.2 [21]

2.6.2 Sensor de Temperatura de Termopar

Un termopar se hace con dos alambres de distinto material unidos en un extremo, al aplicar temperatura en la unión de los metales se genera un voltaje muy pequeño, del orden de los milivolts el cual aumenta con la temperatura. [20]

2.6.3 Sensor de Temperatura de Infrarrojos

La tecnología subyacente para los Pirómetros de Radiación Infrarroja está basada en el principio que dice que todos los objetos emiten radiación a longitudes de onda ubicadas en la región infrarroja del espectro de radiación electromagnética. Los termómetros infrarrojos miden esta radiación y proporcionan una señal de salida calibrada en una variedad de rangos según los requisitos del cliente. [20]

2.6.4 Termómetro de Alcohol

Un termómetro de alcohol es un tipo de Termómetro de Líquido en Vidrio que utiliza la variación en el volumen del alcohol causada por cambios en temperatura. Su expansión se mide utilizando una escala grabada en el tubo del termómetro. [22]

Lecturas de $-40^{\circ}\text{C}/-40^{\circ}\text{F}$ a $50^{\circ}\text{C}/120^{\circ}\text{F}$.



Ilustración 14 Termómetro de Alcohol

2.7 Baterías

2.7.1 Batería de Polímero de Iones de Litio

Las baterías de polímero de iones de litio utilizan la electroquímica líquida de iones de litio en una matriz de polímeros conductores de iones que eliminan el electrolito libre dentro de la celda para producir energía. [23] Estas baterías son recargables.

Batería de 3.7V y 2200mA, provee energía al Seeduino Stalker V3.0.

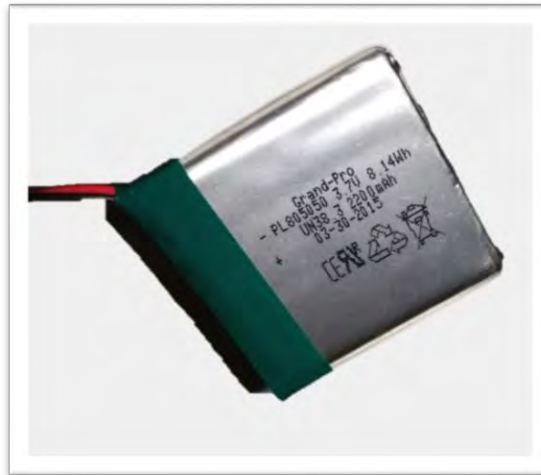


Ilustración 15 Batería de Polímero de Iones de Litio

2.7.2 Batería de Níquel-Hidruro Metálico

Una batería de níquel-hidruro metálico (Ni-MH) es un tipo de batería recargable que utiliza un ánodo de oxidróxido de níquel (NiOOH), como la batería de níquel cadmio, pero su cátodo es de una aleación de hidruro metálico. [24]



Ilustración 16 Batería de Níquel-Hidruro Metálico

2.7.3 Batería de Níquel-Cadmio

Las baterías de níquel-cadmio son recargables de uso doméstico e industrial. Utilizan un cátodo de hidróxido de níquel y un ánodo de un compuesto de cadmio. El electrolito es de hidróxido de potasio. Esta configuración de materiales permite recargar la batería una vez está agotada, para su reutilización. [25]



Ilustración 17 Batería de Níquel-Cadmio

2.8 Panel Solar

2.8.1 Panel Monocristalino

Las placas solares monocristalinas están compuestas por células monocristalinas. El modo más común de fabricación de células de silicio monocristalino (sc-Si) consiste en partir de un lingote de un único cristal de silicio, obtenido por los métodos de Czochralski (Cz) o zona flotante (FZ), y cortarlo en obleas que constituyen el sustrato sobre el que tendrá lugar todo el proceso restante (unión “p-n”, metalización, etc.). [26]

Panel Solar de 5.5V y 540mA con una dimensión de 160mmx138mm. Provee energía a la batería de polímero de litio-ion.

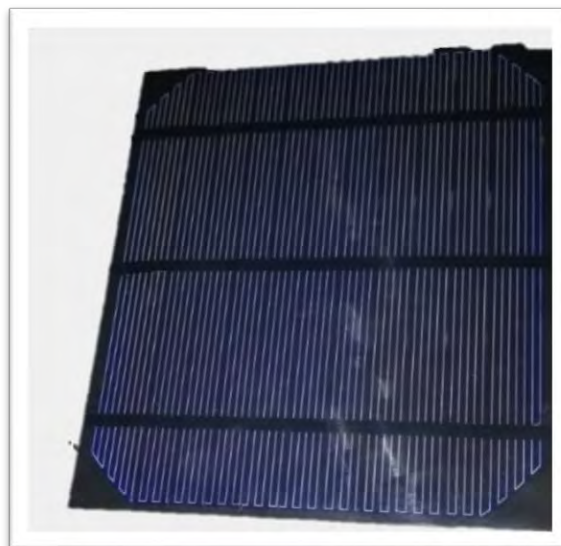


Ilustración 18 Panel Monocristalino

2.8.2 Panel Policristalino

Los paneles solares policristalinos están compuestos por células policristalinas. Podemos diferenciarlos por su color “azulado” y no poseen el chafalán en las esquinas como los monocristalinos. Las células de silicio policristalino (mc-Si) utilizan obleas de silicio como sustrato, éstas proceden del corte de un bloque de silicio que se ha dejado solidificar lentamente en un crisol y que está formado por muchos pequeños cristales de silicio. [26]

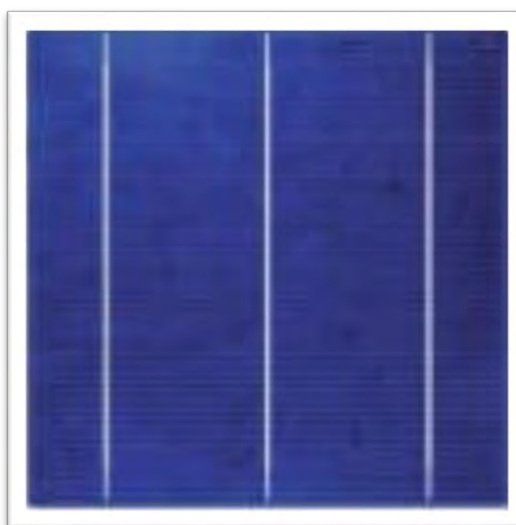


Ilustración 19 Panel Policristalino

2.9 Almacenamiento

Una tarjeta microSD es un tipo de tarjeta de memoria flash, permite la lectura y escritura de múltiples posiciones de memoria en la misma operación, extraíble utilizada para almacenar información. Las tarjetas microSD a veces se denominan μ SD o uSD. [27] Las tarjetas se utilizan en teléfonos móviles y otros dispositivos móviles. Es la tarjeta de memoria más pequeña que se puede comprar; a 15 mm x 11 mm x 1 mm. Tiene un almacenamiento de 2GiB.



Ilustración 20 Tarjeta microSD

Capítulo 3 Desarrollo

3.1 Fases

3.1.1 Componentes (Ensamblado lógico y físico)

El sistema de adquisición de datos necesita poder obtener datos de un sensor, por ejemplo, de temperatura, y la capacidad de almacenamiento

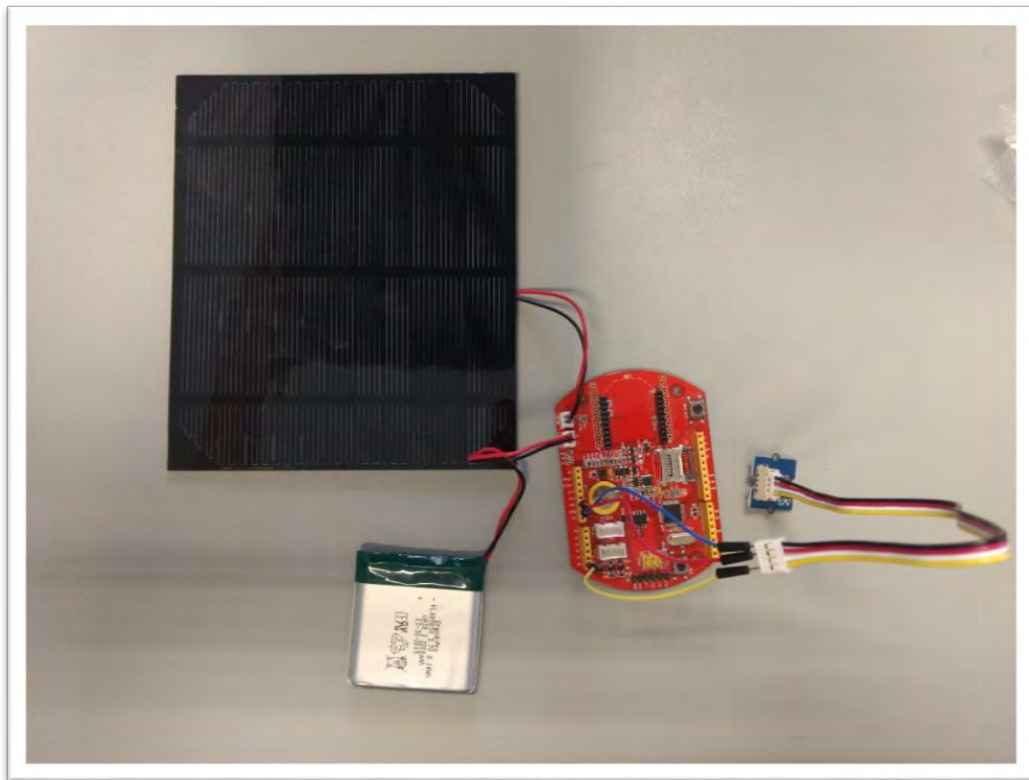


Ilustración 21 Sistema de Adquisición de Datos

3.1.2 Ensamblado lógico

3.1.2.1 Conexión de batería y de panel solar

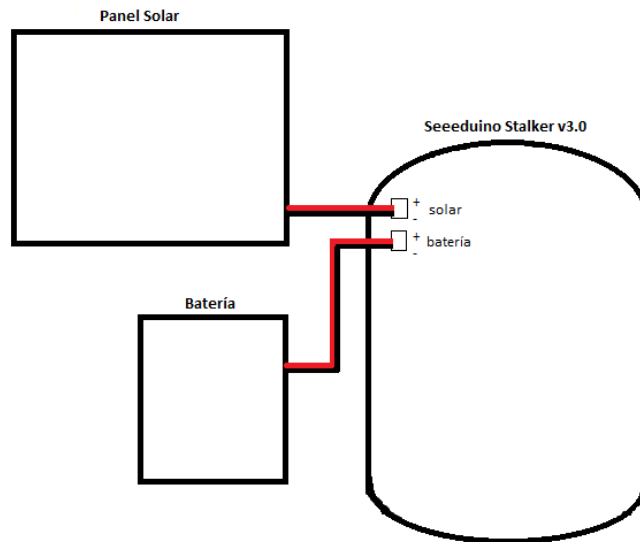


Diagrama 1 Cableado Batería y Panel Solar

3.1.2.2 Conexión de sensor de temperatura

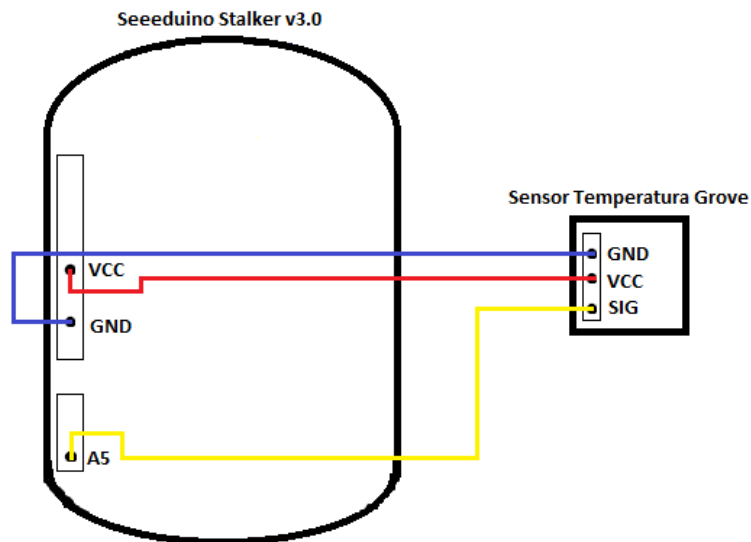


Diagrama 2 Cableado Sensor de Temperatura Grove

3.1.3 Ensamblado físico

3.1.3.1 Conexión de batería y panel solar

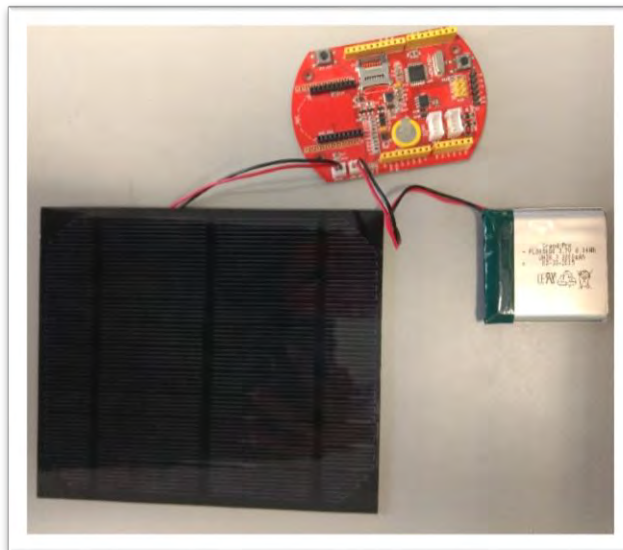


Ilustración 22 Cableado Batería y Panel Solar

3.1.3.2 Conexión de sensor de temperatura

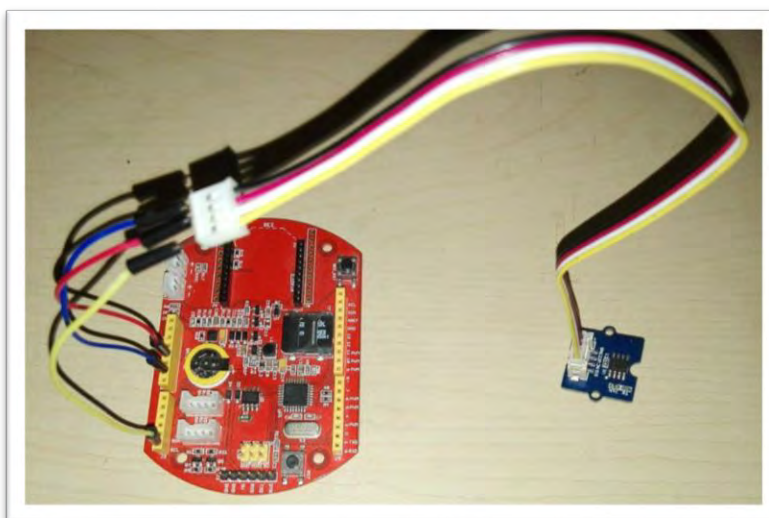


Ilustración 23 Cableado Sensor de Temperatura Grove

3.1.4 Programación

La programación se llevó acabo en el Arduino IDE usando un Arduino UNO como programador del Seeeduino Stalker V3.0. Ya que el Seeeduino Stalker v3.0 viene listo para usar con un UartsBee como programador, pero para ser programado se usaron cables para acomodar el Arduino UNO como el programador.

3.1.4.1 Conexión puente para programación de Seeeduino Stalker v3.0

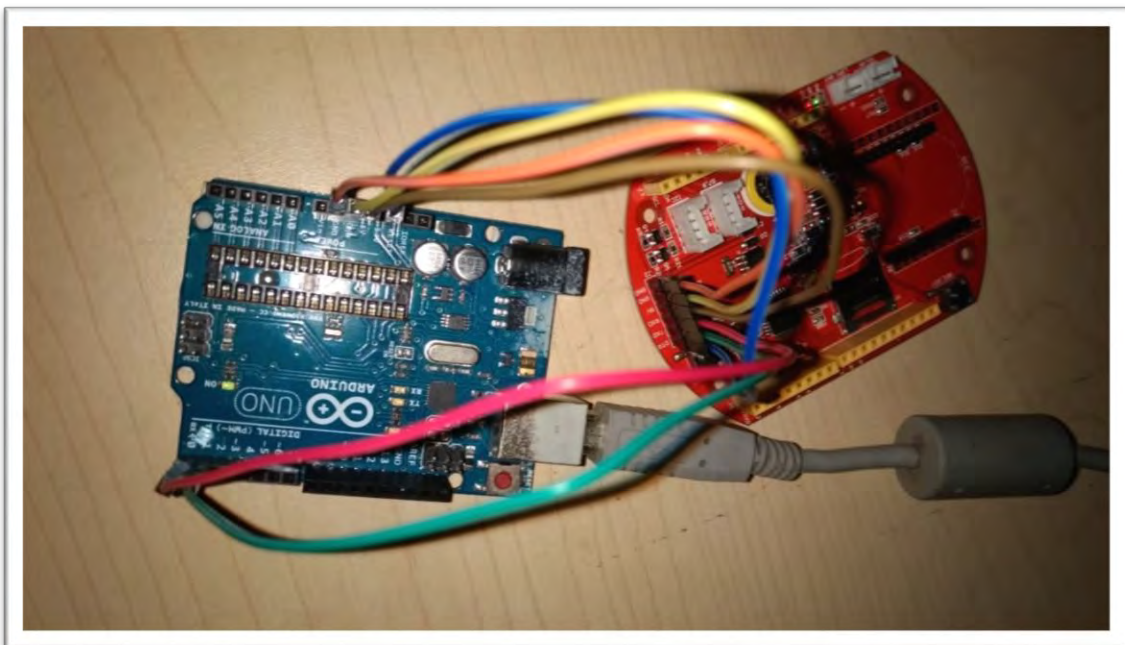


Ilustración 24 Cableado Arduino UNO a Seeeduino Stalker v3.0

3.1.4.2 Conexión puente para programación de Seeeduino Stalker v3.0

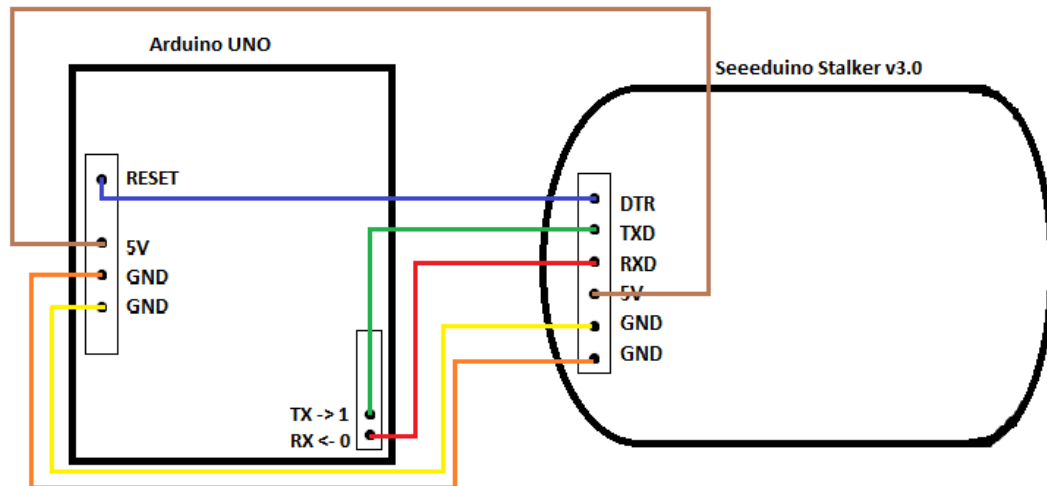


Diagrama 3 Cableado Arduino UNO a Seeeduino Stalker v3.0

Para usar el Arduino UNO como el programador se tuvo que quitar el Atmel ATmega 328, ya que, si se dejaba, el código se programaba en el Arduino UNO y no en el Seeeduino Stalker v3.0.

La programación consistió en tres pasos para la iniciación, que solo se ejecuta una vez.

1. Asignar fecha y hora
2. Encender tarjeta microSD
3. Verificar inicio de tarjeta microSD

Para la lectura del sensor y escritura de la tarjeta microSD se ejecutaron los siguientes pasos en repetición dependiendo del tiempo programado.

1. Lectura de sensor
2. Lectura de fecha y hora
3. Escribir las lecturas en un archivo CSV en la tarjeta microSD

3.2 Código:

[Apéndice 4](#)

Capítulo 4 Resultados

Para el análisis de datos se llevaron a cabo dos tipos de pruebas, la primera, siendo la prueba de batería y la segunda es una prueba de control de calidad del sistema de adquisición de datos.

Para el monitoreo se hicieron tres pruebas para verificar si el funcionamiento del código afectaba la carga de la batería. Estas pruebas consistieron en poner el sistema de adquisición de datos en uso únicamente con la batería hasta que se le terminara la carga en tiempos de lectura del sensor de 1 minuto, 30 y 60 minutos, fecha y hora.

Las pruebas para el control de calidad del sistema de adquisición de datos se hicieron de manera paralela al anteriormente mencionado. Este consistió en tomar medidas con un termómetro de alcohol a la misma vez que el sistema de adquisición de datos y por lo tanto luego comparar que las mediciones sean parecidas. Al igual se calculó la desviación estándar, una medida de dispersión, de las medidas para poder saber cuánto se alejan los valores con respecto a la media. [30]

En base a la fórmula de Ley de Ohm, $I=V/R$, en donde I es corriente, V es voltaje y R es resistencia, se justificó la razón por la duración de vida de la batería en los diferentes parámetros de tiempo en el código utilizados para la prueba. Por uso del sensor de temperatura, el parámetro ambiental es el que influyo más la vida de la batería que el funcionamiento del código.

4.1 Prueba 1

En la primera prueba se programó el sistema de adquisición de datos para que tome la medida de temperatura cada minuto por una hora y con un termómetro de alcohol se tomó la temperatura al mismo tiempo.

4.1.1 Código cada minuto:

[Apéndice 1](#)

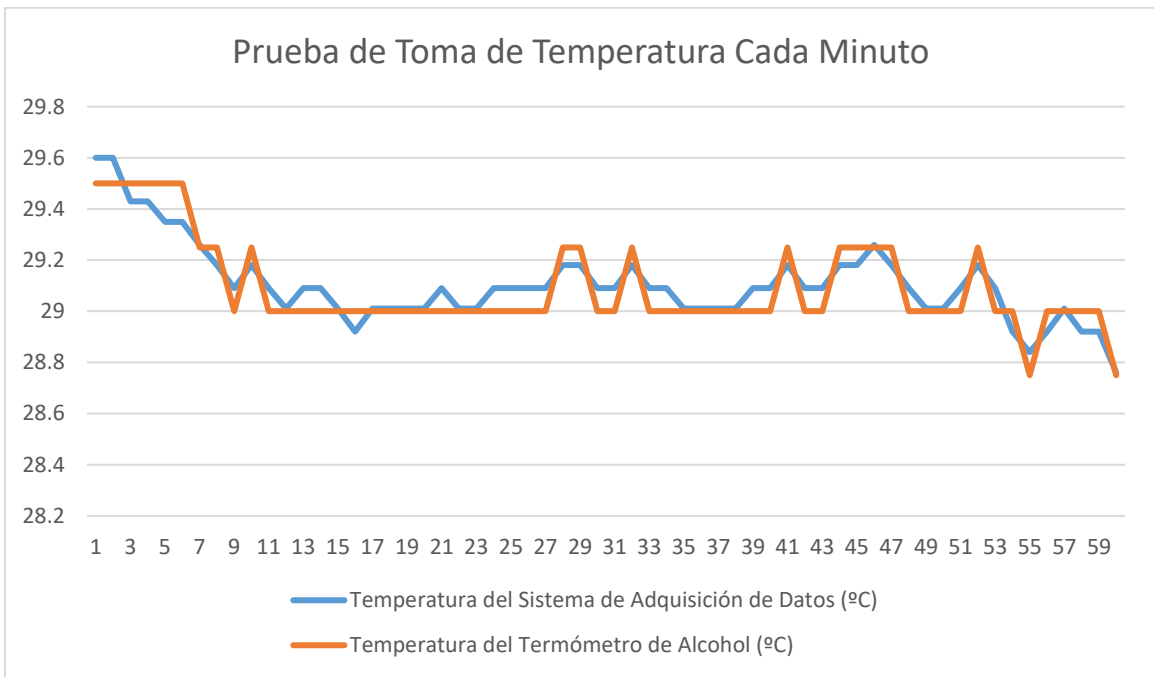
La carga de la batería duró aproximadamente 14.37 días.

Temperatura del Sistema de Adquisición de Datos (°C)	Temperatura del Termómetro de Alcohol (°C)	Desviación Estándar $\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{n}}$
29.6	29.5	0.005
29.6	29.5	0.045
29.43	29.5	0.040
29.43	29.5	0.040
29.35	29.5	0.040
29.35	29.5	0.040
29.26	29.25	0.040
29.18	29.25	0.005
29.09	29	0.005
29.18	29.25	0.005
29.09	29	0.005
29.01	29	0.005
29.09	29	0.005
29.09	29	0.005
29.01	29	0.005
28.92	29	0.005
29.01	29	0.005
29.01	29	0.005
29.01	29	0.005
29.01	29	0.005
29.09	29	0.005
29.01	29	0.005
29.01	29	0.045
29.09	29	0.045
29.09	29	0.045
29.09	29	0.045

29.09	29	0.045
29.18	29.25	0.045
29.18	29.25	0.045
29.09	29	0.045
29.09	29	0.045
29.18	29.25	0.045
29.09	29	0.045
29.09	29	0.045
29.01	29	0.045
29.01	29	0.045
29.01	29	0.045
29.01	29	0.045
29.09	29	0.045
29.09	29	0.045
29.18	29.25	0.045
29.09	29	0.045
29.09	29	0.035
29.18	29.25	0.035
29.18	29.25	0.035
29.26	29.25	0.035
29.18	29.25	0.035
29.09	29	0.035
29.01	29	0.035
29.01	29	0.035
29.09	29	0.035
29.18	29.25	0.035
29.09	29	0.005
28.92	29	0.005
28.84	28.75	0.075

28.92	29	0.075
29.01	29	0.035
28.92	29	0.035
28.92	29	0.050
28.76	28.75	0.050

Tabla 4 Prueba de toma de temperatura cada minuto y desviación estándar



Grafica 2 Prueba de toma de temperatura cada minuto

4.2 Prueba 2

En la segunda prueba se programó el sistema de adquisición de datos para que tome la medida de temperatura cada 30 minutos por seis horas y con un termómetro de alcohol se tomó la temperatura al mismo tiempo.

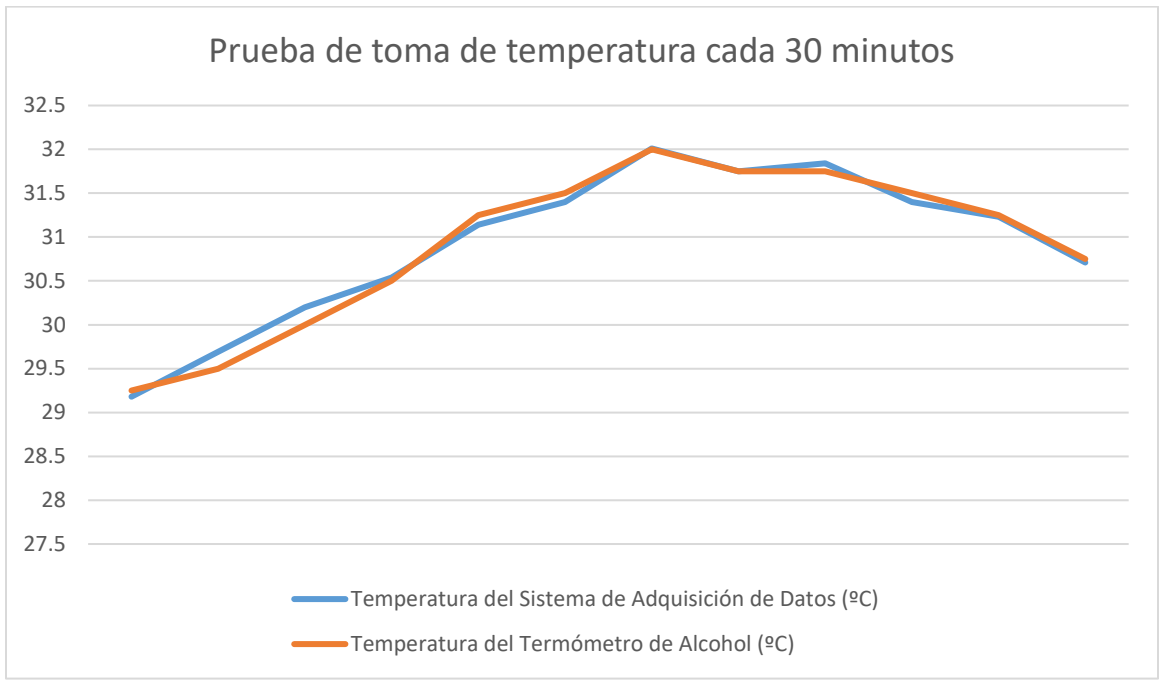
4.2.1 Código cada 30 minutos:

[Apéndice 2](#)

La carga de la batería duró aproximadamente 13.9 días.

Temperatura del Sistema de Adquisición de Datos (°C)	Temperatura del Termómetro de Alcohol (°C)	Desviación Estándar $\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{n}}$
29.18	29.25	0.035
29.69	29.5	0.095
30.2	30	0.100
30.54	30.5	0.020
31.14	31.25	0.055
31.4	31.5	0.050
32.01	32	0.005
31.75	31.75	0.000
31.84	31.75	0.045
31.4	31.5	0.050
31.23	31.25	0.010
30.71	30.75	0.020

Tabla 5 Prueba de toma de temperatura cada 30 minutos y desviación estándar



Grafica 3 Prueba de toma de temperatura cada 30 minutos

4.3 Prueba 3

En la tercera prueba se programó el sistema de adquisición de datos para que tome la medida de temperatura cada 60 minutos por 12 horas y con un termómetro de alcohol se tomó la temperatura al mismo tiempo.

4.3.1 Código cada 60 minutos:

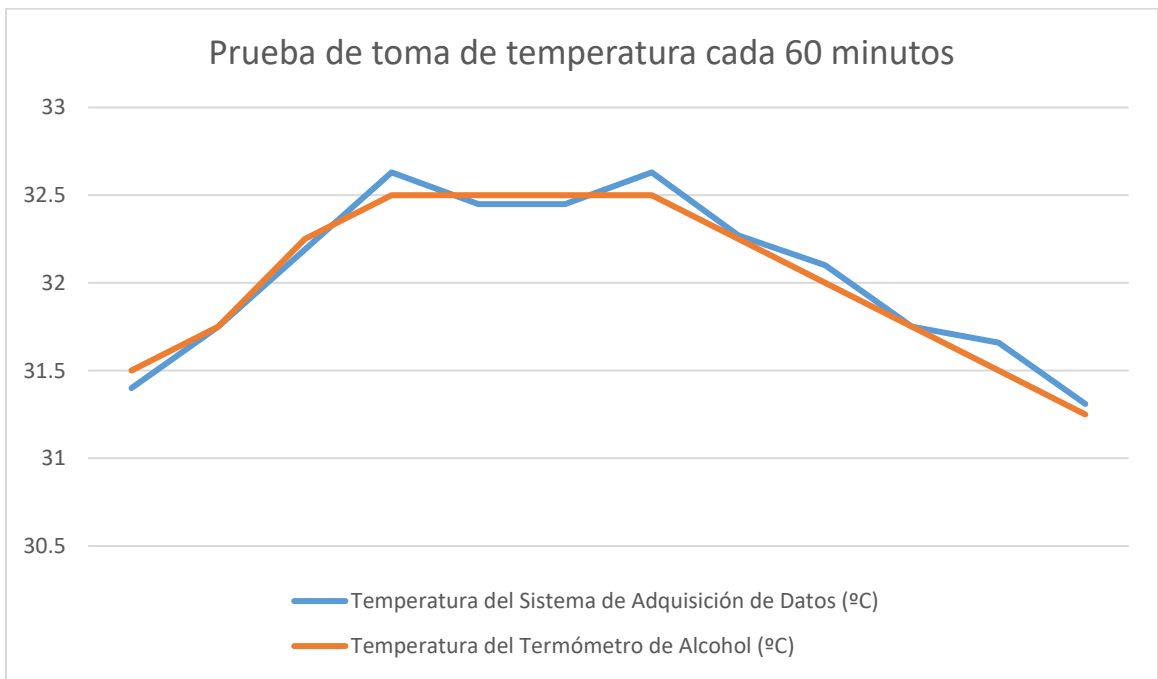
[Apéndice 3](#)

La carga de la batería duró aproximadamente 13.08 días.

Temperatura del Sistema de Adquisición de Datos (°C)	Temperatura del Termómetro de Alcohol (°C)	Desviación Estándar $\sigma = \sqrt{\frac{\sum(x - \bar{x})^2}{n}}$
31.4	31.5	0.050
31.75	31.75	0.000
32.19	32.25	0.030

32.63	32.5	0.065
32.45	32.5	0.025
32.45	32.5	0.025
32.63	32.5	0.065
32.27	32.25	0.010
32.1	32	0.050
31.75	31.75	0.000
31.66	31.5	0.080
31.31	31.25	0.030

Tabla 6 Prueba de toma de temperatura cada 60 minutos y desviación estándar



Grafica 4 Prueba de toma de temperatura cada 60 minutos

Capítulo 5 Conclusiones

El cuidado del medio ambiente es una responsabilidad de todos los seres humanos, es un tema de gran importancia, ya que de ello depende nuestra supervivencia. En este sentido el monitoreo ambiental es una actividad esencial que permite medir las condiciones del medio ambiente, información de muy alta relevancia para biólogos, investigadores, autoridades ambientales, gobierno etc.

El sistema propuesto en su fase inicial incorpora un sensor el cual brinda un monitoreo “constante” de la temperatura de un cuerpo de agua, o de cualquier otro lugar, posteriormente incorporará mediciones de grado de acidez o basicidad de una solución acuosa, oxígeno disuelto en agua, entre otros; lo cual brindará información relevante para la elaboración de proyectos que conllevan un impacto ambiental.

Para realizar un monitoreo ambiental eficiente, se debe tomar en cuenta que éste debe de llevarse a cabo desde la planeación de algún proyecto y durante evaluación observando los estándares de calidad.

Una vez realizado el proceso de adquisición de datos, se realiza el análisis de los datos obtenidos para generar información cuantificable tal como los niveles de contaminación, cambios drásticos de temperatura, acidez, humedad y muchos más. El registro de esta información puede ser graficada o tabulada para facilitar su estudio y a su vez comparable con datos anteriores para poder determinar cambios importantes

Durante el proceso de monitoreo se debe llevar un control de calidad en las mediciones, que garanticen la precisión del sistema de adquisición de datos y la manera que se manejan los registros. Para el análisis de datos se deben usar estrategias estadísticas para poder organizar y analizar todos los registros y llegar a una conclusión correcta.

El sistema propuesto proyecta una vida útil de aproximadamente 3 años, ya que la batería de polímero de iones de litio se desgasta. Es recomendable hacer un cambio de batería cada 2 años para tener un buen funcionamiento. El panel solar debe someterse

a mantenimiento para garantizar un nivel adecuado de energía. Es recomendable limpiar el panel solar cada 3 meses. En caso de que el panel solar se ubique en un lugar con mucho polvo, la limpieza deberá ser más frecuente.

La unidad de almacenamiento (2 GiB) tiene una capacidad algo superior a los 25,000,000 de registros (archivos csv). Si el registro de datos se realiza cada hora, el sistema almacenaría datos por aproximadamente 48 años.

Este proyecto aporta al tema de monitoreo como un sistema fácil de usar y autosuficiente para utilizarse especialmente en áreas remotas en donde no se necesita la presencia de un operador, donde se requiera manejo de monitoreo a mediano plazo. Cuenta con características de portabilidad ya que es un sistema pequeño, de bajo consumo de energía, sencillo y de bajo costo.

Capítulo 6 Alcance

Como va avanzando la tecnología, es importante que cualquier sistema tenga posibilidades de mejora. La facilidad de usar la placa Seeeduino Stalker v3.0 permite esto. Ya que la placa incluye entradas digitales y analógicas, se pueden agregar más sensores de monitoreo y antenas inalámbricas para el uso remoto. Al igual que la creación de un programa que grafique todos los resultados, ya siendo al momento con el uso remoto o a la finalización de la adquisición de datos con la información en la tarjeta microSD.

La compañía SeeedStudio maneja módulos compatibles con el Seeeduino Stalker v3.0 para un monitoreo más conveniente.

- Xbee – monitoreo inalámbrico
- GPSBee – monitoreo de GPS
- BluetoothBee – monitoreo en tiempo real vía bluetooth para uso en dispositivos móviles.
- RFBee – control remoto

Referencias

- [1] N. Instruments, «National Instruments,» National Instruments, [En línea]. Available: <http://www.ni.com/data-acquisition/what-is/esa/>. [Último acceso: 12 mayo 2017].
- [2] J. C. Jr, «Engineering within ecological constraints,» de *Perspectives on Ecology and Engineering*, Washington DC, National Academy Press, 1996, pp. 13-30.
- [3] Environmentally Sustainable Development, «How Can We Monitor Air and Water,» de *Monitoring Environmental Progress: A Report on Work in Progress*, Washington D.C, The World Bank, 1995, pp. 27-37.
- [4] H. Mora, «Repositorio Institucional de la Universidad de Alicante,» 2002. [En línea]. Available: <http://hdl.handle.net/10045/19119>. [Último acceso: 1 june 2017].
- [5] DATAQ Instruments, «Thermocouple Model DI-808,» [En línea]. Available: <https://www.dataq.com/products/di-808/>. [Último acceso: 9 septiembre 2017].
- [6] DATAQ Instruments, «Graptec MT100,» [En línea]. Available: <https://www.dataq.com/products/graptec/mt100/>. [Último acceso: 9 septiembre 2017].
- [7] Agustín Esquivel-Pat, Gian N. Cuello-Nicholson, Doris V. Tun-Caamal, Isaías May-Canché, Blandy B. Pamplona-Solís, Raquel I. Saavedra-Vargas, «Implementación de una red de sensores para el monitoreo en tiempo real de la calidad del agua,» *Innovación para la Vinculación FOMIX - Quintana Roo*, pp. 44-54, Julio - Diciembre 2015.
- [8] I. Lozano, «Cuatro alternativas a Arduino: BeagleBone, Raspberry Pi, Nanode y Waspote,» 9 enero 2013. [En línea]. Available: <https://blogthinkbig.com/4-alternativas-arduino-beaglebone-raspberrypi-nanode-waspote/>. [Último acceso: 6 septiembre 2017].
- [9] Arduino, «Arduino UNO & Genuino UNO,» Arduino, [En línea]. Available: <https://www.arduino.cc/en/main/arduinoBoardUno>. [Último acceso: 28 abril 2017].
- [10] Seeed Studio, «Seeeduino Stalker V3,» 20 Julio 2016. [En línea]. Available: http://wiki.seeedstudio.com/wiki/Seeeduino-Stalker_v3. [Último acceso: 28 abril 2017].
- [11] Nanode, «What is Nanode,» 2012. [En línea]. Available: <http://www.nanode.eu/what-is-nanode/>. [Último acceso: 6 septiembre 2017].
- [12] Nanode, «Products,» [En línea]. Available: <http://www.nanode.eu/products>. [Último acceso: 6 septiembre 2017].

- [13] Nanode, «Nanode Classic,» 2012. [En línea]. Available: <http://www.nanode.eu/classic-build/>. [Último acceso: 6 septiembre 2017].
- [14] Libelium, «Hardware,» [En línea]. Available: <http://www.libelium.com/products/waspmote/hardware/>. [Último acceso: 6 septiembre 2017].
- [15] Beagle Board, «BeagleBone,» 3 febrero 2017. [En línea]. Available: <http://beagleboard.org/bone>. [Último acceso: 6 septiembre 2017].
- [16] Orange Pi, «Orange Pi One,» [En línea]. Available: <http://www.orangepi.org/orangepione/>. [Último acceso: 6 septiembre 2017].
- [17] M. Rouse y V. Silverthorne, «Integrated Development Environment (IDE),» TechTarget, 3 junio 2016. [En línea]. Available: <http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>. [Último acceso: agosto 2017].
- [18] C. Arsenault, «Best IDE Software - a List of the Top 10,» 11 julio 2017. [En línea]. Available: <https://www.keycdn.com/blog/best-ide/>. [Último acceso: 6 septiembre 2017].
- [19] National Instruments, «¿Qué es un Sensor?,» [En línea]. Available: <http://www.ni.com/data-acquisition/what-is/esa/#sensors>. [Último acceso: 23 agosto 2017].
- [20] J. Bausá Aragonés, C. García Gómez, B. Zaragozaí , A. Gil Martínez, D. Moreno Campos y A. Galiana Llinares, «Sensores de Temperatura,» [En línea]. Available: http://server-die.alc.upv.es/asignaturas/LSED/2003-04/0.Sens_Temp/ARCHIVOS/SensoresTemperatura.pdf. [Último acceso: 6 septiembre 2017].
- [21] Seeed, «Grove - Temperature Sensor V1.2,» [En línea]. Available: http://wiki.seeed.cc/Grove-Temperature_Sensor_V1.2/. [Último acceso: 28 abril 2017].
- [22] P. J. Camilleri, «Thermometers,» [En línea]. Available: <http://staff.um.edu.mt/pcam2/Resources/Thermometers.pdf>. [Último acceso: 15 abril 2017].
- [23] Woodbank Communications Ltd., «Rechargeable Lithium Batteries,» 2005. [En línea]. Available: <http://www.mpoweruk.com/lithiumS.htm>. [Último acceso: 1 junio 2017].
- [24] Linear, «2GIG Bateria,» [En línea]. Available: http://www.inalarm.mx/2gig/pdf/documentos_enlace/Bateria-2GIG.pdf. [Último acceso: 8 septiembre 2017].

- [25] EcuRed, «Baterías de Ni-Cd,» [En línea]. Available: https://www.ecured.cu/Bater%C3%ADas_de_Ni-Cd. [Último acceso: 8 septiembre 2017].
- [26] SunFields Europe, «Tipos de paneles solares en el sector fotovoltaico,» [En línea]. Available: <https://www.sfe-solar.com/noticias/articulos/tipos-de-paneles-solares-fotovoltaicos/>. [Último acceso: 8 septiembre 2017].
- [27] M-Short, «Sparkfun,» [En línea]. Available: <https://learn.sparkfun.com/tutorials/sd-cards-and-writing-images>. [Último acceso: 1 junio 2017].
- [28] F. M. Ladarola y L. Mora, «Qué es la desviación estándar y como interpretarla #1,» 11 noviembre 2009. [En línea]. Available: <https://tradingcenter.wordpress.com/2009/11/11/que-es-la-desviacion-estandar-y-como-interpretarla-1/>. [Último acceso: 24 agosto 2017].
- [29] R. J. Schweers, «Repositorio Institucional de la UNLP,» 2002. [En línea]. Available: <http://hdl.handle.net/10915/3835>. [Último acceso: 2 junio 2017].
- [30] Seeed, «Seeeduino Stalker V3 - Waterproof Solar Kit,» [En línea]. Available: http://wiki.seeed.cc/Seeeduino_Stalker_V3-Waterproof_Solar_Kit/. [Último acceso: 2017 junio 16].
- [31] Arduino, «Getting Started with Arduino and Genuino UNO,» Arduino, [En línea]. Available: <https://www.arduino.cc/en/Guide/ArduinoUno#toc1>. [Último acceso: 28 abril 2017].
- [32] OMEGA, «Sensor de temperatura,» [En línea]. Available: <http://es.omega.com/prodinfo/medicion-temperatura.html>. [Último acceso: 15 febrero 2017].
- [33] C. San Martín S, F. Torres V, R. Barrientos S y M. Sandoval D, «Monitoreo y control de temperatura de un estanque de agua entre Chile y España usando redes de alta velocidad,» *Revista Facultad de Ingeniería*, vol. 11, pp. 41-46, 2003.
- [34] J. L. D. B. Carranza, «Inventores,» 1 agosto 2016. [En línea]. Available: <http://blog.inventores.mx/curso-kit-inventores-v1/>. [Último acceso: 17 enero 2017].
- [35] Z. G, G. W., Z. G, Z. J, Z. Y y X. D, «Design of Environmental Parameters Monitoring System for Watermelon Seedlings Based on Wireless Sensor Networks,» 22 junio 2010. [En línea]. Available: <http://naturalspublishing.com/files/published/3hblr4122h28mx.pdf>. [Último acceso: 21 mayo 2017].
- [36] K. Appelt, «Hunker,» [En línea]. Available: <https://www.hunker.com/13408232/mercury-thermometer-vs-alcohol-thermometer>. [Último acceso: 15 abril 2017].

- [37] EPA, «Mercury Thermometers,» 13 septiembre 2016. [En línea]. Available: <https://www.epa.gov/mercury/mercury-thermometers>. [Último acceso: 15 abril 2017].
- [38] A. B. Perez, «Calcular la desviación estándar paso a paso,» [En línea]. Available: <https://es.khanacademy.org/math/probability/data-distributions-a1/summarizing-spread-distributions/a/calculating-standard-deviation-step-by-step>. [Último acceso: 24 agosto 2017].
- [39] Disfruta Las Matemáticas, «Varianza y desviación estándar,» 2011. [En línea]. Available: <http://www.disfrutalasmaticas.com/datos/desviacion-estandar.html>. [Último acceso: 24 agosto 2017].
- [40] DATAQ Instruments, «DI-730-USB,» [En línea]. Available: <https://www.dataq.com/products/di-730/di-730-usb.html>. [Último acceso: 9 septiembre 2017].

Apéndices

Apéndice 1

```
#include <math.h>
#include <Wire.h>
#include "DS1337.h"
#include <SPI.h>
#include <SD.h>

const int B=4275;           // B valor de termistor
const int R0 = 100000;     // R0 = 100k
const int pinSensorTemp = A5; // Sensor de Temperatura conectado a pin A5

DS1337 RTC;

char weekDay[][4] = {"Dom", "Lun", "Mar", "Mie", "Jue", "Vie", "Sab" };
DateTime dt(2017, 5, 18, 20, 54, 30, 4); //año, mes, día, hora, min, seg, día

void setup()
{
  Serial.begin(19200);
  Wire.begin();
  RTC.begin();
  RTC.adjust(dt); //fecha definida como dt

  pinMode(4,OUTPUT); //pin de control de encendido o apagado.
  digitalWrite(4,LOW); //Encender tarjeta microSD.

  Serial.print("Encendiendo tarjeta microSD...");

  // Revisando si tarjeta microSD puede iniciar
  if (!SD.begin(10)) //Chipselect en pin 10
  {
    Serial.println("Tarjeta microSD no pudo iniciar, o no se encuentra");
    return;
  }
  Serial.println("Tarjeta microSD encontrada e inicializada.");
}

void loop()
{
```

```
digitalWrite(LED_BUILTIN, HIGH); // cada que escriba en microSD, encender LED

// Lectura de Temperatura con el sensor
int a = analogRead(pinSensorTemp);
float R = 1023.0/((float)a)-1.0;
R = 100000.0*R;
float temperatura=1.0/(log(R/100000.0)/B+1/298.15)-273.15; //convertir temperatura
en centígrado

//Fecha y Hora
DateTime now = RTC.now(); //fecha y hora actual

// En caso de que se le agregue un display al Seeeduino o querer monitorearlo con el
serial monitor
// No comentar si requiere uso, pero esta comentado para ahorrar batería por el
procesamiento
// Imp Fecha
/*Serial.print(now.year(), DEC);
Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.date(), DEC);
Serial.print(' ');
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
Serial.print(weekday[now.dayOfWeek()]);
Serial.println();

//Imp Temperatura
Serial.print("Temperatura = ");
Serial.println(temperatura);
Serial.println();*/

//Escribir en microSD
File logFile = SD.open("DATALOG.CSV", FILE_WRITE);

if(logFile) {
logFile.print(now.year(), DEC);
```

```
    logFile.print('/');
    logFile.print(now.month(), DEC);
    logFile.print('/');
    logFile.print(now.date(), DEC);
    logFile.print(',');
    logFile.print(now.hour(), DEC);
    logFile.print(':');
    logFile.print(now.minute(), DEC);
    logFile.print(':');
    logFile.print(now.second(), DEC);
    logFile.print(',');
    logFile.print(weekDay[now.dayOfWeek()]);
    logFile.print(',');
    logFile.println(temperatura);
    logFile.print(',');
    logFile.close();
}

    delay(30000); //1 min
}
```

Apéndice 2

```
#include <math.h>
#include <Wire.h>
#include "DS1337.h"
#include <SPI.h>
#include <SD.h>

const int B=4275;           // B valor de termistor
const int R0 = 100000;     // R0 = 100k
const int pinSensorTemp = A5; // Sensor de Temperatura conectado a pin A5

DS1337 RTC;

char weekDay[][4] = {"Dom", "Lun", "Mar", "Mie", "Jue", "Vie", "Sab" };
DateTime dt(2017, 5, 18, 20, 54, 30, 4); //año, mes, día, hora, min, seg, dia

void setup()
{
  Serial.begin(19200);
  Wire.begin();
  RTC.begin();
  RTC.adjust(dt); //fecha definido como dt

  pinMode(4,OUTPUT); //pin de control de encendido o apagado.
  digitalWrite(4,LOW); //Encender tarjeta microSD.

  Serial.print("Encendiendo tarjeta microSD...");

  // Revisando si tarjeta microSD puede iniciar
  if (!SD.begin(10)) //Chipselect en pin 10
  {
    Serial.println("Tarjeta microSD no pudo iniciar, o no se encuentra");
    return;
  }
  Serial.println("Tarjeta microSD encontrada e inicializada.");
}

void loop()
{
```

```
digitalWrite(LED_BUILTIN, HIGH); // cada que escriba en microSD, encender LED

// Lectura de Temperatura con el sensor
int a = analogRead(pinSensorTemp);
float R = 1023.0/((float)a)-1.0;
R = 100000.0*R;
float temperatura=1.0/(log(R/100000.0)/B+1/298.15)-273.15; //convertir temperatura
en centígrado

//Fecha y Hora
DateTime now = RTC.now(); //fecha y hora actual

// En caso de que se le agregue un display al Seeeduino o querer monitorearlo con el
serial monitor
// No comentar si requiere uso, pero esta comentado para ahorrar batería por el
procesamiento
// Imp Fecha
/*Serial.print(now.year(), DEC);
Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.date(), DEC);
Serial.print(' ');
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
Serial.print(weekday[now.dayOfWeek()]);
Serial.println();

//Imp Temperatura
Serial.print("Temperatura = ");
Serial.println(temperatura);
Serial.println();*/

//Escribir en microSD
File logFile = SD.open("DATALOG.CSV", FILE_WRITE);

if(logFile) {
logFile.print(now.year(), DEC);
```

```
    logFile.print('/');
    logFile.print(now.month(), DEC);
    logFile.print('/');
    logFile.print(now.date(), DEC);
    logFile.print(',');
    logFile.print(now.hour(), DEC);
    logFile.print(':');
    logFile.print(now.minute(), DEC);
    logFile.print(':');
    logFile.print(now.second(), DEC);
    logFile.print(',');
    logFile.print(weekDay[now.dayOfWeek()]);
    logFile.print(',');
    logFile.println(temperatura);
    logFile.print(',');
    logFile.close();
}

    delay(900000); //30 min
}
```


Apéndice 3

```
#include <math.h>
#include <Wire.h>
#include "DS1337.h"
#include <SPI.h>
#include <SD.h>

const int B=4275;          // B valor de termistor
const int R0 = 100000;    // R0 = 100k
const int pinSensorTemp = A5; // Sensor de Temperatura conectado a pin A5

DS1337 RTC;

char weekDay[][4] = {"Dom", "Lun", "Mar", "Mie", "Jue", "Vie", "Sab" };
DateTime dt(2017, 5, 18, 20, 54, 30, 4); //año, mes, día, hora, min, seg, dia

void setup()
{
  Serial.begin(19200);
  Wire.begin();
  RTC.begin();
  RTC.adjust(dt); //fecha definida como dt

  pinMode(4,OUTPUT); //pin de control de encendido o apagado.
  digitalWrite(4,LOW); //Encender tarjeta microSD.

  Serial.print("Encendiendo tarjeta microSD...");

  // Revisando si tarjeta microSD puede iniciar
  if (!SD.begin(10)) //Chipselect en pin 10
  {
    Serial.println("Tarjeta microSD no pudo iniciar, o no se encuentra");
    return;
  }
  Serial.println("Tarjeta microSD encontrada e inicializada.");
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH); // cada que escriba en microSD, encender LED
```

```
// Lectura de Temperatura con el sensor
  int a = analogRead(pinSensorTemp);
  float R = 1023.0/((float)a)-1.0;
  R = 100000.0*R;
  float temperatura=1.0/(log(R/100000.0)/B+1/298.15)-273.15; //convertir temperatura
en centígrado

//Fecha y Hora
  DateTime now = RTC.now(); //fecha y hora actual

// En caso de que se le agregue un display al Seeeduino o querer monitorearlo con el
serial monitor
// No comentar si requiere uso, pero esta comentado para ahorrar batería por el
procesamiento
// Imp Fecha
  /*Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.date(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  Serial.print(weekday[now.dayOfWeek()]);
  Serial.println();

//Imp Temperatura
  Serial.print("Temperatura = ");
  Serial.println(temperatura);
  Serial.println();*/

//Escribir en microSD
  File logFile = SD.open("DATALOG.CSV", FILE_WRITE);

  if(logFile) {
    logFile.print(now.year(), DEC);
    logFile.print('/');
```

```
    logFile.print(now.month(), DEC);
    logFile.print('/');
    logFile.print(now.date(), DEC);
    logFile.print(',');
    logFile.print(now.hour(), DEC);
    logFile.print(':');
    logFile.print(now.minute(), DEC);
    logFile.print(':');
    logFile.print(now.second(), DEC);
    logFile.print(',');
    logFile.print(weekDay[now.dayOfWeek()]);
    logFile.print(',');
    logFile.println(temperatura);
    logFile.print(',');
    logFile.close();
}

    delay(1800000); //60 min
}
```

Apéndice 4

```
#include <math.h>
#include <Wire.h>
#include "DS1337.h"
#include <SPI.h>
#include <SD.h>

const int B=4275;           // B valor de termistor
const int R0 = 100000;     // R0 = 100k
const int pinSensorTemp = A5; // Sensor de Temperatura conectado a pin A5

DS1337 RTC;

char weekDay[][4] = {"Dom", "Lun", "Mar", "Mie", "Jue", "Vie", "Sab" };
DateTime dt(2017, 5, 18, 20, 54, 30, 4); //año, mes, día, hora, min, seg, dia

void setup()
{
  Serial.begin(19200);
  Wire.begin();
  RTC.begin();
  RTC.adjust(dt); //fecha definida como dt

  pinMode(4,OUTPUT); //pin de control de encendido o apagado.
  digitalWrite(4,LOW); //Encender tarjeta microSD.

  Serial.print("Encendiendo tarjeta microSD...");

  // Revisando si tarjeta microSD puede iniciar
  if (!SD.begin(10)) //Chipselect en pin 10
  {
    Serial.println("Tarjeta microSD no pudo iniciar, o no se encuentra");
    return;
  }
  Serial.println("Tarjeta microSD encontrada e inicializada.");
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH); // cada que escriba en microSD, encender LED
```

```
// Lectura de Temperatura con el sensor
  int a = analogRead(pinSensorTemp);
  float R = 1023.0/((float)a)-1.0;
  R = 100000.0*R;
  float temperatura=1.0/(log(R/100000.0)/B+1/298.15)-273.15; //convertir temperatura
en centígrado

//Fecha y Hora
  DateTime now = RTC.now(); //fecha y hora actual

// En caso de que se le agregue un display al Seeeduino o querer monitorearlo con el
serial monitor
// No comentar si requiere uso, pero esta comentado para ahorrar batería por el
procesamiento
// Imp Fecha
  /*Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.date(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  Serial.print(weekday[now.dayOfWeek()]);
  Serial.println();*/

//Imp Temperatura
  Serial.print("Temperatura = ");
  Serial.println(temperatura);
  Serial.println();*/

//Escribir en microSD
  File logFile = SD.open("DATALOG.CSV", FILE_WRITE);

  if(logFile) {
    logFile.print(now.year(), DEC);
    logFile.print('/');
    logFile.print(now.month(), DEC);
```

```
    logFile.print('/');  
    logFile.print(now.date(), DEC);  
    logFile.print(',');  
    logFile.print(now.hour(), DEC);  
    logFile.print(':');  
    logFile.print(now.minute(), DEC);  
    logFile.print(':');  
    logFile.print(now.second(), DEC);  
    logFile.print(',');  
    logFile.print(weekDay[now.dayOfWeek()]);  
    logFile.print(',');  
    logFile.println(temperatura);  
    logFile.print(',');  
    logFile.close();  
    }  
  
    delay(1800000); //60 min  
}
```