

SISTEMA DE ADQUISICIÓN DE DATOS Y BASE DE TIEMPO PARA FUTURA SONDA ESTRATOSFÉRICA EN MISIÓN ESPACIAL DE ÓRBITA BAJA

Miguel Ángel Silva García

Instituto Tecnológico de Celaya, Tecnológico Nacional de México
12030542 @itcelaya.edu.mx

Alejandro Espinosa Calderón

Instituto Tecnológico de Celaya, Tecnológico Nacional de México
alejandro.espinosa @itcelaya.edu.mx

José Alfredo Padilla Medina

Instituto Tecnológico de Celaya, Tecnológico Nacional de México
alfredo.padilla @itcelaya.edu.mx

Mario Alberto Mendoza Bárcenas

Instituto Politécnico Nacional, Centro de Desarrollo Aeroespacial
mmendozab @ipn.mx

Lauro Santiago Cruz

Universidad Nacional Autónoma de México, Instituto de Ingeniería, Coordinación de Electrónica
lsc @pumas.ii.unam.mx

Rafael Prieto Meléndez

Universidad Nacional Autónoma de México, Grupo de Modelado y Simulación de Procesos
rafael.prieto @ccadet.unam.mx

Resumen

Los satélites de órbita baja (hasta 800 km) permiten la validación operativa de sistemas electrónicos y de instrumentación experimental, así como el monitoreo y registro de diversas variables físicas, dentro de tiempos de desarrollo reducidos. Además permiten ahorros económicos directos en su puesta en órbita y operación.

Antes de integrar satélites de órbita baja, se requiere validar experimentalmente, bajo condiciones de espacio cercano, componentes y equipos electrónicos, esto se hace en vuelos suborbitales.

En este artículo se describe el desarrollo de un sistema de adquisición de datos, enfocado al registro de datos de temperatura, que incluye un reloj de tiempo real para asociarla a los datos obtenidos como etiqueta de tiempo, almacenando los datos en una memoria SD. El núcleo computacional está basado en la tarjeta Arduino DUE, en torno a la cual se ha integrado un sistema de adquisición de datos, que incluye interfaces de conexión con periféricos, como un sensor de temperatura, un RTC, una pantalla LCD y una memoria SD. Adicionalmente, se discute y propone un esquema de futuras de pruebas del prototipo, en ambiente de termo-vacío, para certificación previa al vuelo en la sonda estratosférica.

Palabra(s) Clave(s): Arduino DUE, I2C, memoria SD, real time clock, SPI.

1. Introducción

La atmósfera se divide en diversas capas, la relación entre la altitud y la temperatura es distinta dependiendo de la capa atmosférica considerada: troposfera, estratosfera, mesosfera y termosfera. La estratosfera se sitúa entre la tropósfera y la mesósfera, y se extiende en una capa que va desde los 10 hasta los 50 km de altura aproximadamente. La temperatura aumenta progresivamente desde los $-55\text{ }^{\circ}\text{C}$ de la tropopausa hasta alcanzar los $0\text{ }^{\circ}\text{C}$ de la estratopausa, esto es debido principalmente a la absorción de las moléculas de ozono que absorben radiación electromagnética en la región del ultravioleta. En la parte baja de la estratósfera la temperatura es relativamente estable, y en toda la capa hay muy poca humedad.

La estratósfera es una de las capas más importantes de la atmósfera ya que reúne el 90% del ozono presente en la atmósfera y absorbe del 97% al 99% de la radiación ultravioleta de alta frecuencia, sin embargo, esta capa sufre cambios a partir del efecto invernadero, un problema creciente que pudiera alterar su dinámica global y con ello el espesor de la capa [1]. Es por esta razón que los

satélites de órbita baja cobran importancia al medir diferentes variables en esta capa de la atmósfera.

El ozono de la atmósfera se mide con equipos desde la tierra o desde el aire con aviones, globos y satélites. Algunos equipos miden el ozono directamente tomando muestras de aire y otros en forma remota. Los equipos usan técnicas ópticas tomando la luz solar y láser como fuentes de luz o usan reacciones químicas propias del ozono [2]. Aunque por el momento no se propone medir ozono, esta será una variable futura de monitoreo, una vez que se haya validado la funcionabilidad en la estratósfera del sistema propuesto en este artículo.

Uno de los principales problemas que se presentan con este tipo mediciones es el costo elevado de los equipos usados en estos satélites, además de su difícil adquisición para experimentos con el objetivo de contribuir a una investigación. Existen diversas propuestas para sistemas en satélites de baja altura [3,4,5,6], a diferencia de dichas propuestas en este artículo se propone un sistema de monitoreo basado en tecnología Arduino DUE, la cual es comercial, fácil de adquirir y de relativo bajo precio.

Arduino es una plataforma de prototipos electrónicos de código abierto basada en una placa con entradas y salidas, analógicas y digitales. El microcontrolador de la placa se programa usando el “Arduino Programming Language” (basado en Wiring) y el “Arduino Development Environment” (basado en Processing). Esta plataforma ofrece diversas ventajas sobre otros microcontroladores, tales como:

- Barato: Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras, muchas tarjetas Arduino cuestan menos de \$100 USD, a diferencia de otras tarjetas como la Basys 2 de Digilent cuyo valor esta alrededor de los \$150 USD, o la tarjeta Chipkit Uno también de Digilent, que, aunque su precio es similar a las tarjetas Arduino, cuenta con menos recursos y pines.
- Multiplataforma: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.

- Entorno de programación amigable y claro: El entorno de programación de Arduino es de amigable interpretación para principiantes, pero suficientemente flexible y complejo para que usuarios avanzados puedan aprovecharlo también.
- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++ [7].

En este artículo se presenta un sistema de medición de temperatura con etiqueta de tiempo (*timestamp*) utilizando un reloj de tiempo real (RTC) y almacenamiento de datos en memoria SD basado en tecnología Arduino, el cual se planea utilizar en un satélite de órbita baja. El principal objetivo de este sistema es ofrecer una alternativa más barata y sencilla para una sonda estratosférica, que permitirá validar sistemas para diseñar, posteriormente, un satélite de órbita baja.

2. Desarrollo

Para este proyecto se realizó el diseño e implementación de un primer prototipo en donde se utilizaron los siguientes materiales:

- Arduino DUE.
- Reloj en tiempo real DS1307.
- Módulo lector de tarjetas SD compatible con Arduino.
- Display LCD 16x2.

En figura 1 se puede observar la arquitectura del sistema con los bloques más importantes.

La tarjeta Arduino DUE utilizada es la primera tarjeta que cuenta un Microcontrolador de núcleo ARM de 32 bits Atmel SAM3X8E ARM Cortex-M3 a 84 Mhz, lo cual mejora las capacidades estándar de Arduino. Dentro de sus principales características cuenta 12 pines de entrada/salida digitales, 12 entradas

analógicas de 12 bits de resolución, dos módulos para comunicación I2C y una cabecera para comunicación SPI (figura 2) [7].

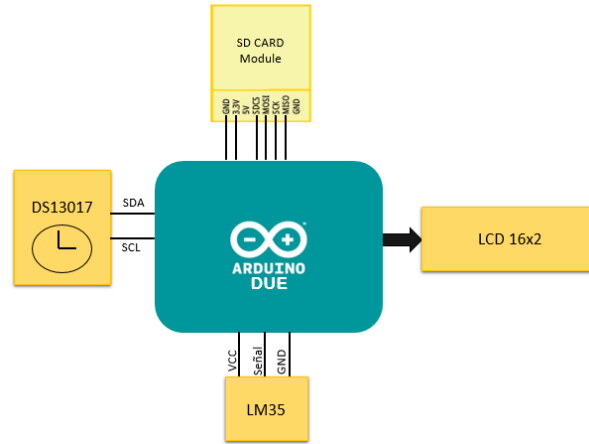


Figura 1 Arquitectura del sistema de adquisición de datos.

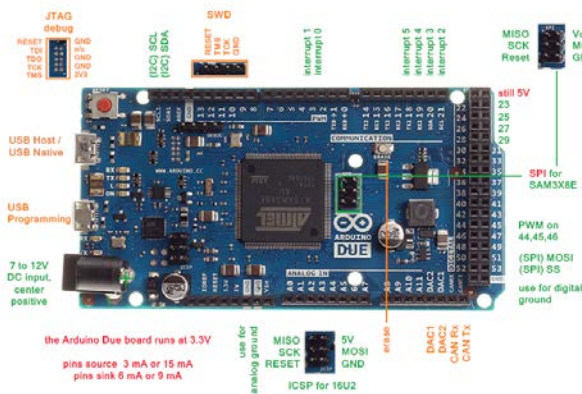


Figura 2 Tarjeta Arduino DUE.

El código implementado en la tarjeta Arduino DUE permite la comunicación y configuración de un reloj en tiempo real DS1307 Real-Time-Clock [8], el cual es un circuito integrado de ocho terminales con líneas de conexión a un bus I2C [9]. El DS1307, es un reloj/calendario en tiempo real que proporciona segundos, minutos, horas, día, mes y año. El final de fecha de mes se ajusta automáticamente durante meses menores de 31 días, incluyendo correcciones para el año bisiesto. El DS1307 tiene incorporado un circuito de sensor de tensión que detecta fallas de energía y cambia automáticamente al suministro de batería de respaldo.

También se realizó la lectura de un sensor de temperatura LM35 [10], el cual cuenta con una precisión calibrada de 1 °C y un rango que abarca desde -55 a +150 °C, una de las características más importantes es que tiene un comportamiento lineal, en donde cada 10 mV equivale a 1 °C.

Finalmente, los datos obtenidos del reloj DS1307 y el sensor LM35 son almacenados en una memoria SD a través de un módulo lector de tarjetas SD compatible con Arduino. Además la plataforma de Arduino cuenta con una librería para trabajar con este módulo en donde se hace uso del bus de comunicación SPI.

El código implementado en la tarjeta Arduino DUE se puede observar en el diagrama de flujo mostrado en figura 3.

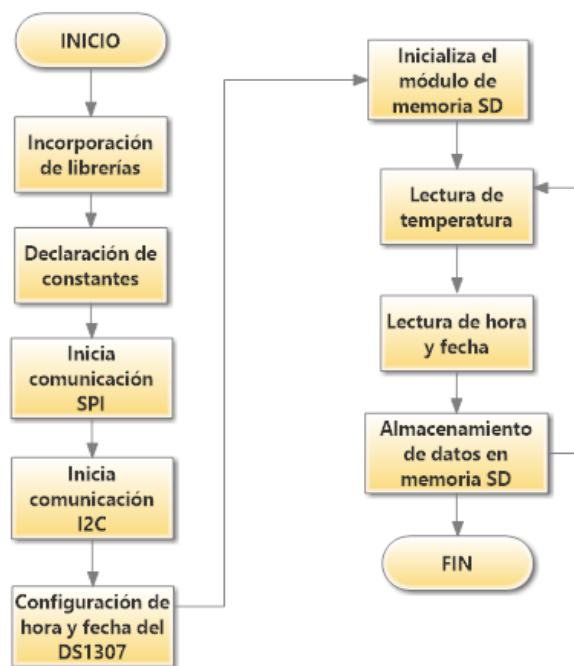


Figura 3 Diagrama de flujo de código implementado en la tarjeta Arduino DUE.

Las librerías que ofrece Arduino para la comunicación SPI, I2C, así como librerías para el módulo SD y el LCD facilitan su programación. Una lista de las librerías utilizadas se puede observar a continuación:

- “Wire.h”; Librería para el manejo del protocolo de comunicación I2C.
- “LiquidCrystal.h”; Librería para controlar un LCD 16x2.

- “RTCdue.h”; Librería para el reloj de tiempo real DS1307, específicamente para la tarjeta Arduino DUE.
- “SPI.h”; Librería para el manejo del protocolo de comunicación SPI.
- “SD.h”; Librería para el control del módulo SD Card.

Para la conexión del módulo SD se usaron los pines directamente del módulo SPI que se encuentra al centro de la tarjeta Arduino DUE (figura 4).

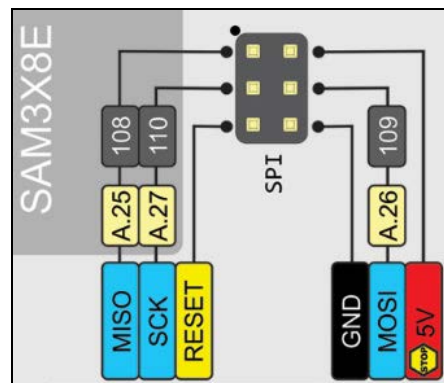


Figura 4 Configuración de pines para el módulo SPI en Arduino DUE.

A diferencia de otros Arduinos la tarjeta Arduino DUE no cuenta con un pin asignado a la terminal CS “*Chip select*”, necesario para la comunicación SPI, por ello se debe especificar en el código cuál pin se va a utilizar. Los posibles pines a los que se les puede asignar esta función son: 4, 10 o 52. Para ello se necesita declarar el pin a utilizar dentro del “*Set up*” (figura 5).

```
//Declaración del pin 10 como Chip Select  
SPI.begin(10);
```

Figura 5 Asignación del pin 10 como *Chip select*.

Para conectar el módulo SD a la tarjeta Arduino DUE fue necesario realizar un divisor de voltaje para la conexión de pines (figura 6), de manera que fuera posible pasar de 5 V, que proporciona el módulo SPI, a 3.3 V, que requiere el módulo de SD.

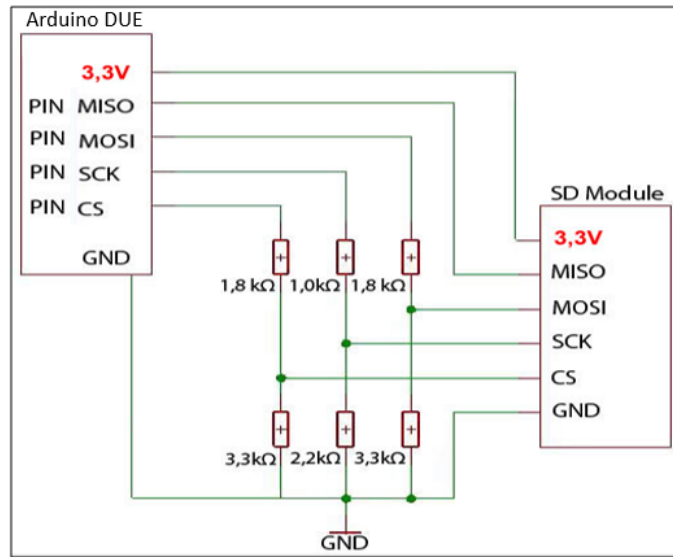


Figura 6 Divisor de voltaje para conexión de pines del módulo SD.

Además de utilizar la librería RTC, que sirve para configurar diferentes parámetros en el reloj DS1307 hecha específicamente para la tarjeta Arduino DUE, fue necesario agregar un par de líneas al código dentro del “*Void setup ()*” para asegurar que la comunicación I2C se configura de una manera correcta (figura 7), ya que sin estas líneas el protocolo de comunicación I2C no funciona correctamente con Arduino DUE.

```
#ifndef AVR
Wire.begin();//Se inicializa comunicacion i2c
#else
Wire1.begin(); // Shield I2C pins connect to alt I2C bus on Arduino Due
#endif
```

Figura 7 Líneas necesarias para establecer la comunicación I2C con el Arduino DUE.

El módulo del reloj DS1307 incluye una batería de 3 V para respaldo en caso de desconexión de la alimentación principal (figura 8). Este módulo presenta un problema para la tarjeta Arduino DUE, ya que las terminales SDA y SCL del módulo están conectadas a unas resistencias de *Pull-Up* a 5 V, al igual que la alimentación del circuito integrado, mientras que el voltaje máximo en los pines de I/O de Arduino DUE es de 3.3 V.

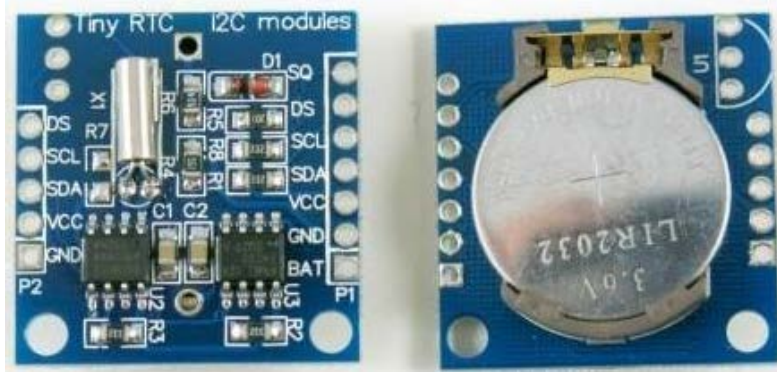


Figura 8 Módulo DS1307.

Para no dañar la tarjeta se revisó un diagrama esquemático del módulo DS1307 (figura 9) para ubicar las resistencias de *Pull-Up* que están conectadas a 5V. Una vez ubicadas las resistencias de *Pull-Up* en el módulo DS1307 se desoldaron y se remplazaron por un par de cables, de manera que fuera posible conectar unas resistencias de 10 k Ω a 3.3 V (figura 10), de esta manera no se dañará la tarjeta Arduino DUE.

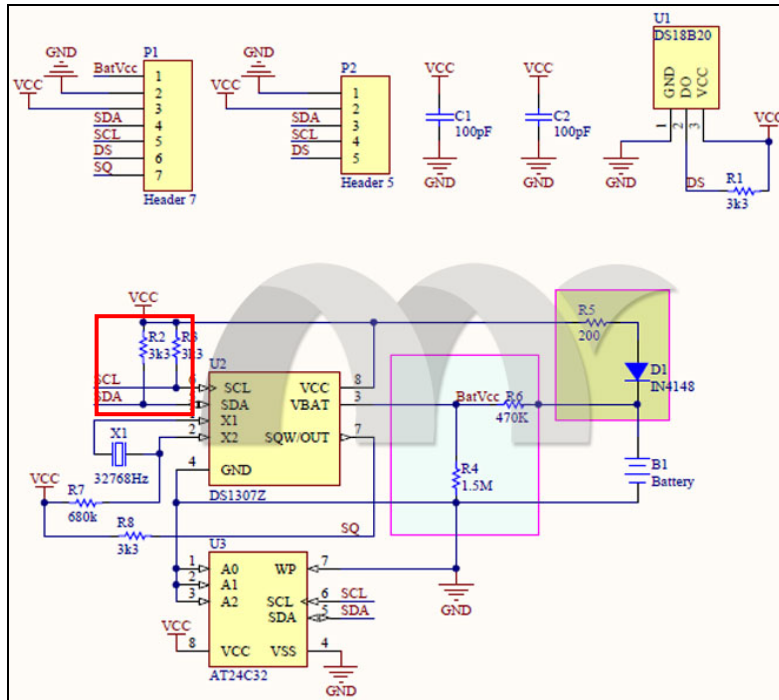


Figura 9 Diagrama esquemático del módulo DS1307.

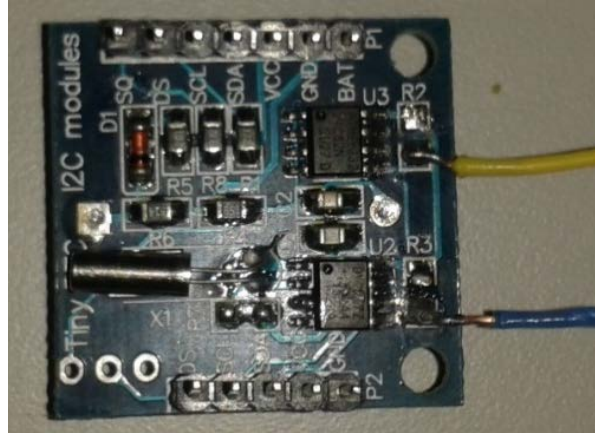


Figura 10 Módulo DS1307 modificado.

Para realizar la lectura de temperatura con el sensor LM35 se utilizó uno de los ocho convertidores ADC que dispone la tarjeta Arduino DUE, los cuales son de 12 bits de resolución por lo que nos entregarán valores desde 0 hasta 4095, en rangos de 0 a 3.3 V.

Para saber qué temperatura está midiendo el sensor es necesario conocer cuántos volts se están leyendo en la entrada analógica, por lo que se realizó la operación mostrada en ecuación 1.

$$X = \frac{\text{Dato} \times 3.3\text{v}}{4095} = V_{out} \quad (1)$$

Por lo tanto, para saber la cantidad de grados centígrados que representa dicho voltaje se realiza la operación mostrada en ecuación 2.

$$^{\circ}\text{C} = \frac{\text{Dato} \times 330}{4095} \quad (2)$$

Dado que la fórmula final tiene operaciones básicas, se pudo incluir en el código sin problemas.

3. Resultados

La figura 11 muestra el primer prototipo realizado en un protoboard. Una vez compilado el código se descargó a la tarjeta Arduino DUE y se comprobó su funcionamiento como se puede observar en las figuras 12 a 15.

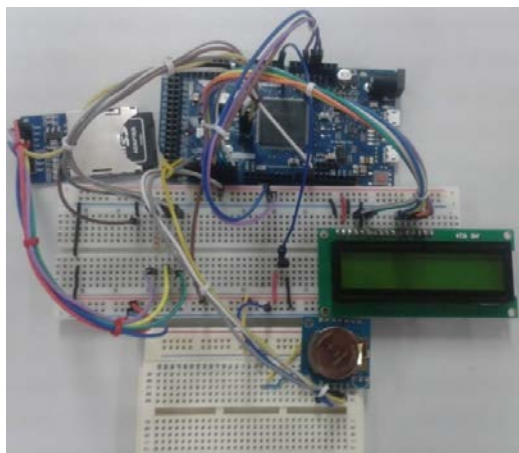


Figura 11 Conexión del primer prototipo.



Figura 12 Mensaje "Bienvenido, ITC Celaya".



Figura13 Mensaje "Inicializando SD".

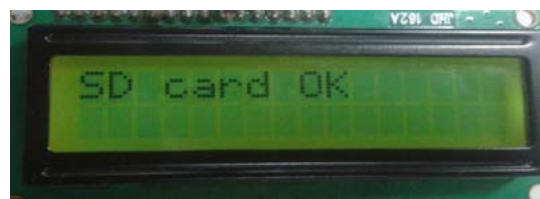


Figura 14 Mensaje "SD card ok".



Figura 15 Temperatura y tiempo.

Los mensajes observados en la LCD 16x2 también pueden ser monitoreados a través del monitor serial del programa Arduino IDE. En este monitor también se puede observar que la hora y fecha de las mediciones fueron sincronizadas con la hora y fecha de compilación (figura 16).

Finalmente, después de que el sistema realizará diversas mediciones de temperatura se comprobó que la memoria SD almacenó correctamente los datos (figura 17). Este módulo soporta una memoria de hasta 2GB, lo que permite registrar aproximadamente 45 millones de lecturas de temperatura con su etiqueta de tiempo.

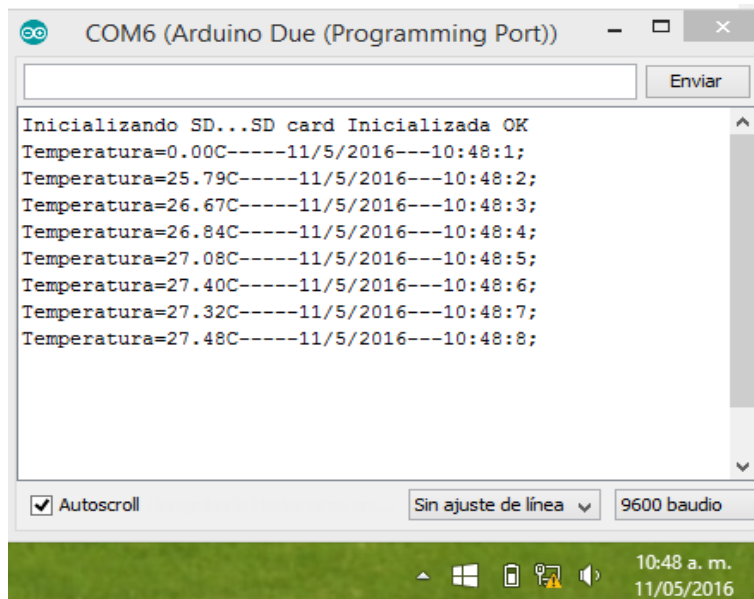


Figura 16 Mensajes en el monitor serial de Arduino.

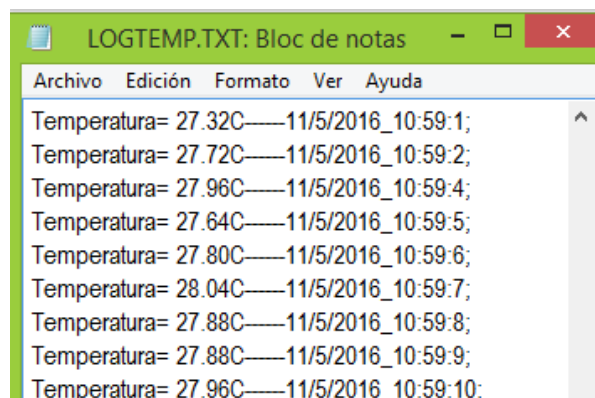


Figura 17 Contenido de memoria SD.

Como se puede observar el código final solo ocupa un 9% de del espacio disponible en la tarjeta (figura 18). Esto permitirá en un futuro incluir más funciones de medición y procesamiento en el dispositivo.

```
Subido
El Sketch usa 50,224 bytes (9%) del espacio de almacenamiento de programa. El máximo es 524,288 bytes.
Atmel SMART device 0x285e0a60 found
Erase flash
done in 0.032 seconds
```

Figura 18 Cantidad de recursos utilizados en la tarjeta Arduino DUE.

Las mediciones de temperatura del sensor LM35 se compraron con las mediciones realizadas con un termómetro de mercurio en un baño térmico, en donde se registraron las mediciones cada 60 segundos (figura 19).

Figura 19 Sensor LM35 y termómetro en baño térmico.

Los datos obtenidos del experimento anterior se muestran en la figura 20. Se obtuvo un error promedio de 2.71%.

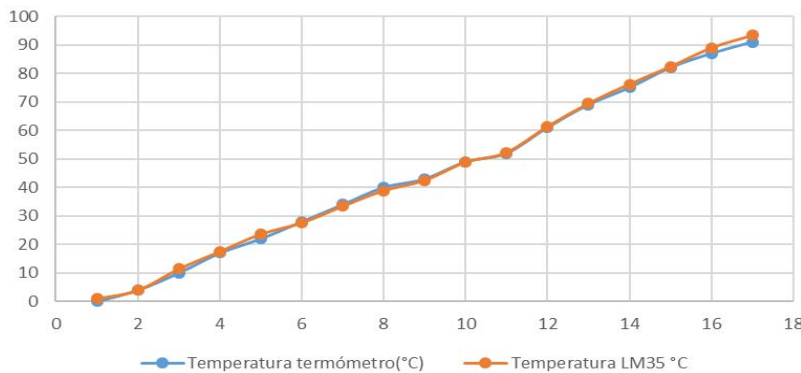


Figura 20 Comparación de lecturas entre termómetro de mercurio y LM35.

4. Discusión

El desarrollo de proyectos en la plataforma de Arduino tiene la ventaja de ofrecer una solución barata en comparación con otras plataformas de programación más sofisticadas, sin embargo, para este proyecto los recursos que ofrece la tarjeta Arduino DUE, así como las librerías utilizadas, fueron suficientes para llevar a cabo un primer prototipo funcional que puede ser utilizado en pruebas iniciales con satélites de baja altura.

Con el desarrollo de este proyecto es posible implementar un sistema de medición de temperatura con etiqueta de tiempo y almacenamiento, todo en tiempo real utilizando una tarjeta y componentes de bajo costo. Otra ventaja del sistema propuesto es que las mediciones capturadas y almacenadas ocupan muy poco espacio dentro de la memoria SD de 2 GB, por lo que se podrían tener millones de mediciones que pueden ser útiles para el análisis de resultados.

Es importante mencionar que el microcontrolador AT91SAM3X8E que incluye la tarjeta Arduino DUE, trabaja en un rango de temperatura de -40 a 85 °C, lo cual será muy importante al momento de realizar las primeras pruebas en un satélite de baja altura, ya que la temperatura en la estratósfera aumenta progresivamente desde los -55 °C hasta alcanzar los 0 °C, por lo que deberán realizarse pruebas para comprobar el desempeño de la tarjeta bajo estas condiciones.

Otro aspecto importante a considerar es el sensor utilizado para medir temperatura, ya que de momento se está utilizando un sensor LM35, pero en futuras pruebas se espera utilizar una sonda térmica 107L.

5. Conclusiones

En este trabajo se demostró que se puede implementar un sistema de medición de temperatura, con referencia de un reloj en tiempo real con almacenamiento en memoria SD en una plataforma comercial como Arduino DUE. Esta plataforma permite una reducción considerable de los costos que implican este tipo de pruebas para mediciones de diferentes variables en la estratósfera.

Antes de realizar pruebas reales en un satélite de órbita baja, será necesario hacer pruebas de desempeño del sistema a baja temperatura y presión. Con estas

pruebas se podrá observar su comportamiento ante condiciones semejantes a las que se encuentran en la estratósfera. De esta manera se podrá comprobar si es viable, o no, implementar este tipo de sistemas con tarjetas Arduino, ofreciendo una alternativa mucho más barata en comparación a los sistemas convencionales. Estas pruebas se realizarán en una cámara de termovacío del Centro de Investigaciones Nucleares de la UNAM en meses próximos.

Entre las funciones futuras a incluir en el sistema están la medición de ozono, la obtención de imágenes, y la medición con sonda térmica 107L.

6. Bibliografía y Referencias

- [1] Astronomía educativa. <http://www.astromia.com/tierraluna/capatmosfera.htm>. Abril 2016.
- [2] D. W. Fahey, "20 preguntas y respuestas sobre la capa de Ozono". Evaluación Científica del Agotamiento de Ozono. 2002. Pp. 1-7.
- [3] El-Bayoumi, Abdullah, "A new Checkout-and-Testing-Equipment (CTE) for a satellite Telemetry using LabVIEW". Aerospace Conference. IEEE. 2015. Pp. 1-9.
- [4] Interorbital Systems, Satellite kits. http://www.interorbital.com/interorbital_06222015_030.htm. Agosto 2016.
- [5] Ayyaz, M. Naeem, M. Riaz Suddle, Shakeel Zahid, "Systems design of an economical and general-purpose on-board computer for low-earth-orbit micro-satellites". Second International Conference on. IEEE. 2008. Pp. 1-9.
- [6] TubeSat kit specifications and pricing, version 2.0, Interorbital System. Mojave Air & Space Port California. Estados Unidos. 2013.
- [7] McRoberts, M. Levy, B. Wootton, *Beginning Arduino*. 2010. Apress. New York.
- [8] DS1307 64 x 8 Serial I2C Real-Time Clock. Maxim Integrated Products, Sunnyvale, CA. USA. 2015.
- [9] F. Leens, "An introduction to I2C and SPI protocols". Instrumentation & Measurement Magazine. IEEE. Vol. 2. 2009. Pp 8-13.

[10] LM35 Precision Centigrade Temperature Sensor. Texas Instruments. Dallas, Texas, USA. 2016.

7. Autores

Miguel Angel Silva García. Estudiante de 9° semestre de la carrera de Ingeniería Electrónica en el Instituto Tecnológico de Celaya.

Dr. Alejandro Espinosa Calderón. Recibió su título de Doctorado de la Universidad Autónoma de Querétaro, en Querétaro, Qro., Mex., en 2012, de donde honores, la medalla al mérito académico, y el premio a la mejor tesis doctoral. Ha participado en eventos científicos internacionales en EEUA, Italia y China. Ha recibido premios internacionales de la OSA y USAID, ambos de EEUA. Desde 2013 trabaja en el Departamento. de Ing. Eléctrica y Electrónica del ITCelaya. Sus áreas de interés incluyen: instrumentación electrónica y procesamiento en sistemas digitales.

Dr. Mario Alberto Mendoza Bárcenas. Obtuvo su título de Doctor en Ingeniería Eléctrica en la Universidad Nacional Autónoma de México. Actualmente se desempeña como investigador en la coordinación de Fomento al Desarrollo de la Industria Nacional del Centro de Desarrollo Aeroespacial del Instituto Politécnico Nacional. Trabaja en proyectos multiinstitucionales y multidisciplinarios de desarrollo de instrumentación para vehículos aeroespaciales.

M.I. Lauro Santiago Cruz. Obtuvo su título de Maestro en Electrónica en la División de Estudios de Posgrado de la Facultad de Ingeniería de la UNAM. Actualmente se desempeña como docente en la Facultad de Ingeniería - UNAM y desarrolla proyectos de investigación aplicada en la Coordinación de Electrónica del Instituto de Ingeniería de la UNAM.

M.I. Rafael Prieto Meléndez. Recibió el título Maestro en Ingeniería, en la División de Estudios de Posgrado de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, en 1999. Es profesor de asignatura definitivo en la División de Ciencias Básicas y profesor de asignatura en la División de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México. Sus líneas de trabajo comprenden: Sistemas Embebidos basados en Microcontroladores y FPGA; y Modelado y Simulación de Sistemas.

Dr. José Alfredo Padilla Medina. Recibió su grado de Doctor en Filosofía del Centro de Investigaciones en Óptica, en León, Gto., Mex., en 2003. Actualmente es profesor-investigador de tiempo completo en el Dpto. de Ing. Eléctrica y Electrónica del ITCelaya. Pertenece al Sistema Nacional de Investigadores con nivel I. Sus áreas de interés incluyen: procesamiento de imágenes digitales en el visible e infrarrojo, el sistema visual humano, y la lógica difusa.

Los autores agradecen al Tecnológico Nacional de México por el apoyo económico para la realización de este proyecto.