

Reporte CANSAT

2015

Algunas lecciones aprendidas



Documento realizado por alumnos y docentes del
Instituto San Felipe Neri
Buenos Aires - Argentina

(página intencionalmente en blanco)

Introducción; ¿Qué es CANSAT?

Es una competencia anual estudiantil para el diseño, construcción y lanzamiento de sistemas tipo espaciales, basados en la experiencia ARLISS (A Rocket Launch for International Science Students – Universidad de Stanford - USA), del tamaño de una lata de refrescos.

CANSAT en Argentina

En Argentina no es una competencia, sino que se trata de la difusión gratuita de una metodología de trabajo y estudio a través de la experimentación, empleando lanzadores reutilizables construidos por coheteros amateur. Se trata de diseñar, construir y probar experimentos de pequeño tamaño (5 centímetros de diámetro y 15 centímetros de longitud) que luego son lanzados por cohetes de la ACEMA hasta una altura de 1 km.

Cargas útiles del Instituto San Felipe Neri:

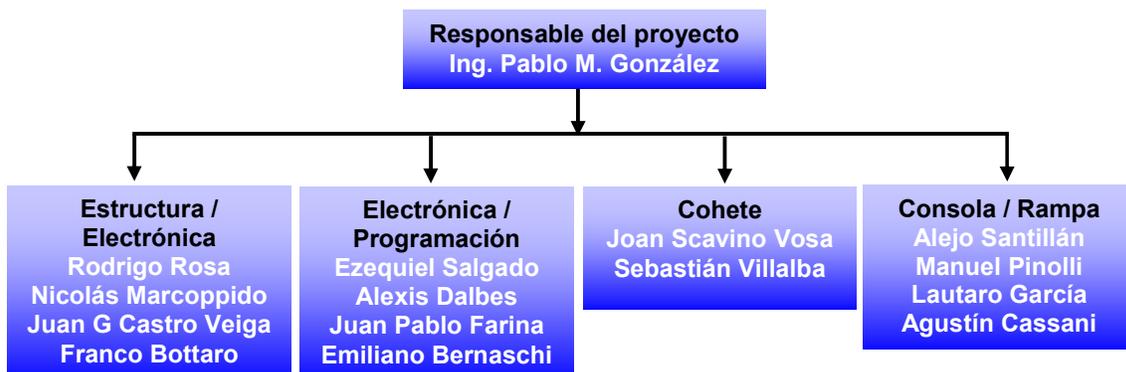
Nro	Año	CANSAT	Experiencia
1	2004	Sirio	Separación de fluidos en ambientes de aceleraciones extremas
2	2005	S/N	Acelerómetro mecánico
3	2006	Rigel	Acelerómetro mecánico
4	2006	Alhena	Cromatografía
5	2006	VH	Barómetro
6	2007	Rigel 2	Acelerómetro mecánico
7	2008	VH 2	Barómetro-altímetro
8	2008	S/N	Demostración de un circuito electrónico
9	2009	Ch ⁺	Radiobaliza y cámara
10	2009	S/N	Comportamiento de fluidos en microgravedad
11	2012	Ch2 ⁺	Altímetro, radiobaliza y cámara

*: con la colaboración de Escuela Técnica ORT

Nuestro proyecto para 2014/15

Nuestra idea fue utilizar un microprocesador Arduino y sensores para recoger información en vuelo, como primer paso para, en posteriores vuelos, realizar telemetría (transmisión radial simultánea de datos).

Equipo y roles



Objetivos de misión

Objetivos científicos:

- *Primario:* Medición del perfil térmico atmosférico mediante un sensor electrónico, almacenándolo en memoria para luego recuperar los datos y construir una curva altura – temperatura.
- *Secundario:* Medición de altura máxima mediante un sensor electrónico autónomo.

Objetivos Tecnológicos:

Certificación en vuelo de un computador Arduino (www.arduino.cc) programado para utilizar un sensor analógico.

Requerimientos

Performance: Se necesita elevar una carga útil a mas de 100 metros mediante un cohete, separarlo del cuerpo del vector en el apogeo, activar el paracaídas para un descenso lento y recuperar la carga útil.

Segmentos

Terreno

Rampa y consola de disparo, equipos de soporte y rescate



Espacial

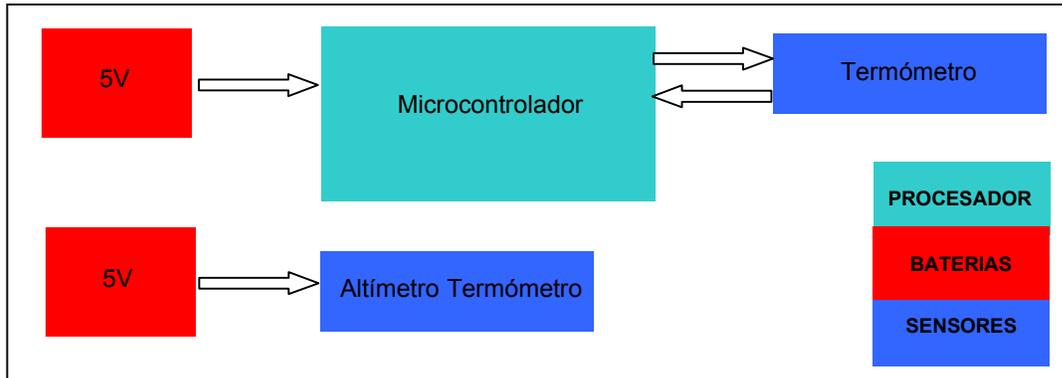
Cohete + CANSAT



Carga útil

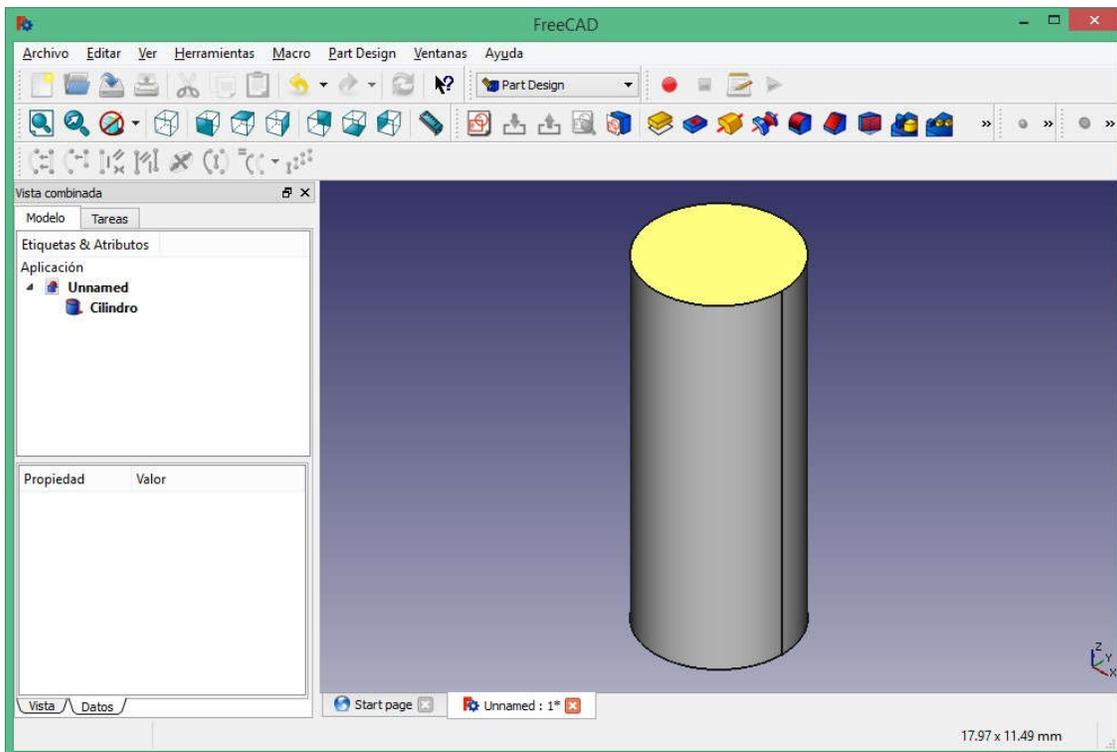
(se amplian los detalles en el Anexo Técnico al final del documento)

Consistió en un microcomputador Arduino, al cual se le conectó un sensor analógico de temperatura y una batería. Además, se agregó un altímetro-termómetro independiente, todo alojado en un soporte estructural:



Diseño de la estructura

Se utilizaron programas informáticos libres, como el *FREECAD*, para realizar la modelización de la estructura cilíndrica y realizar el *layout* de todos los componentes.



Construcción

La estructura se construyó en aluminio y poliestireno expandido tipo “foamboard”, formando placas tipo “sándwich”, cementadas con cianocrilato. Sus medidas eran 12 cm de largo y 4,8 cm de diámetro.

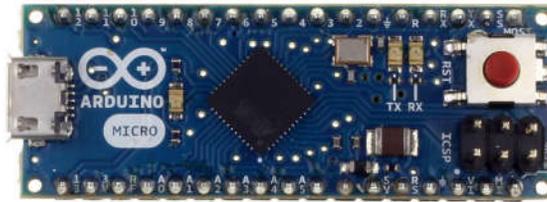


Verificaciones: sólo se midieron las dimensiones exteriores y se peso del conjunto (12 gramos), pero no se realizaron verificaciones mediante ensayos de integridad estructural.

Sistemas electrónicos

Microprocesador

Se optó por un microprocesador *Arduino Micro*, ya que se conecta en forma directa a un puerto USB, y las características de velocidad y memoria interna eran acordes al proyecto.

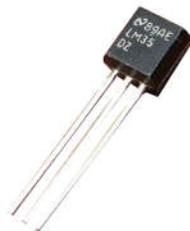


Sensor

Elegimos el sensor analógico de temperatura LM35, debido a su linealidad y a su precio. Se realizó la calibración usando un termómetro digital, determinándose la constante para aplicarla a la fórmula:

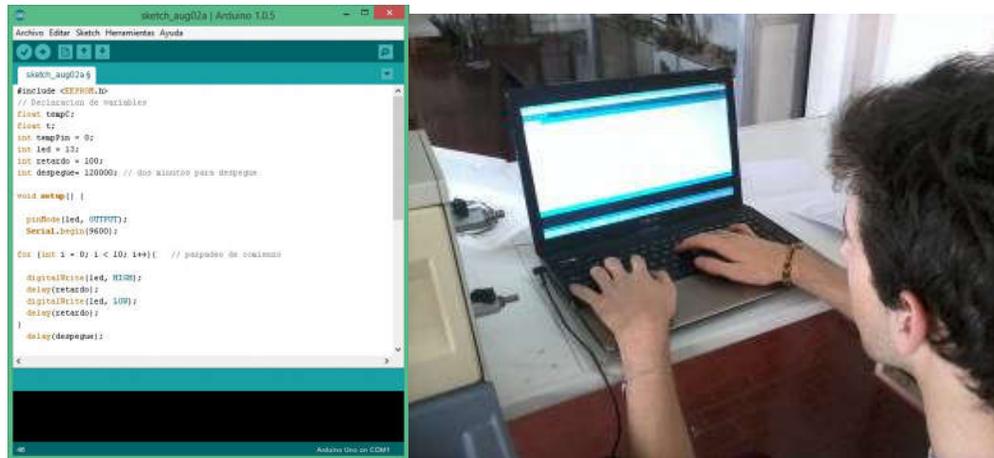
$$temp = (k \cdot t \cdot 100) / 1024$$

Siendo t el valor de la salida del sensor y k la constante, que en nuestro caso arrojó un valor cercano a 5.



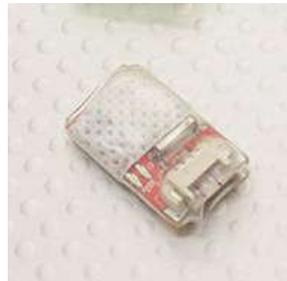
Programación

El ambiente de desarrollo integrado de Arduino (<https://www.arduino.cc/en/Main/Software>), que incluye un editor, un compilador y la herramienta de comunicación, fue el utilizado para programar en C las instrucciones necesarias para almacenar los datos del sensor. En función de la velocidad de descenso, se decidió programar dos mediciones por segundo.



Carga secundaria – sensor independiente

Para medir la altitud y la temperatura en función del tiempo y comparar los valores con los registrados por el computador de vuelo, se empleó un altímetro-termómetro HK usado en aeromodelismo:



Alimentación eléctrica

Utilizamos baterías en serie tipo 20XX (2025/2032, 3V), que conectadas en serie, entregaban el suficiente voltaje y amperaje para cubrir el tiempo de misión.

Integración y pruebas

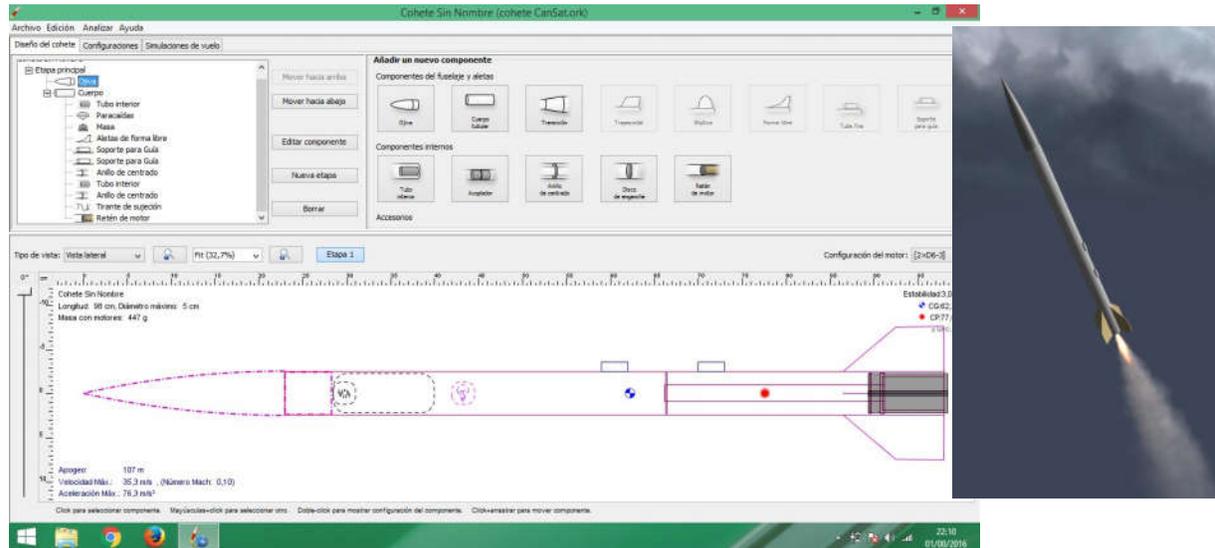
Se probaron todos los componentes por separado y se efectuaron pruebas una vez integrados. Los ensayos incluyeron pruebas en sencillas cámaras de baja presión, medición de consumo eléctrico, etc.



VECTOR

Diseño

Para el diseño del cohete se empleó el programa open source *OPEN ROCKET*, desarrollado en Java, que permite realizar el prototipo completo en forma de maqueta digital y admite la realización de simulaciones de vuelo para establecer si el diseño cumple con las performances esperadas.



Construcción

Se fabricó un cohete capaz de elevar el CANSAT a cotas de por lo menos 200 metros, para lo cual se construyó a partir de un tubo de 50 mm y una ojiva del mismo diámetro, con un motor comercial tipo "F" de la firma argentina *SkyTec*. Luego de lijarlo y darle una mano de impresión, se pegaron las aletas, se colocó la celda de motor, el pistón de presurización, el *shock cord* (unión entre la ojiva y el fuselaje) y los paracaídas. Por último se lo pintó con esmalte y se lo decoró.



Verificaciones

Dado que se demoró la construcción, no hubo tiempo de realizar verificaciones dinámicas ni pruebas de integración con la carga útil.

<https://picasaweb.google.com/102610685240741156409/PrepCansat2015?authkey=Gv1sRgCOW8xpuPq4zkWA>

Rampa, consola de disparo y sistemas de apoyo en tierra

Se recicló una rampa de lanzamiento tipo torre ajustable, construida por alumnos de años anteriores, cementando y renovando la bulonería.

El disparador también se recicló, agregando una batería. Se comprobó el funcionamiento del sistema de comunicaciones y se acondicionó una caja de herramientas con los elementos necesarios para el montaje, ajuste y posible reparación de todos los sistemas.

JORNADA DE LANZAMIENTO – FASE 1

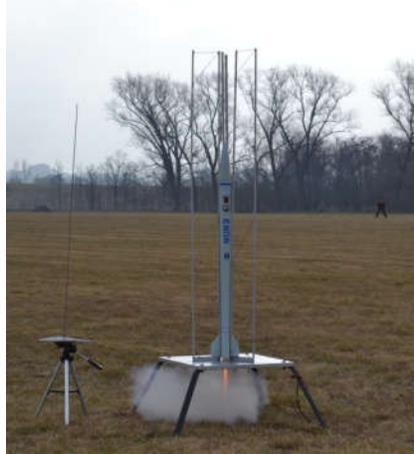
El día 1 de julio de 2015 por la mañana, partimos hacia el campo de vuelo de la Agrupación Aeromodelista “Pucará” (<http://www.agrupacionpucara.com>). Nos reunimos con integrantes de la Escuela técnica ORT, de Buenos Aires, para compartir la jornada.

Cada integrante del equipo tenía un rol asignado: LCO (*Launch Control Officer*, Oficial de Lanzamiento), RSO (*Range Safety Officer*, Oficial de Seguridad), Encargado de carga útil, Jefe de Rampa y Encargado del equipo de rescate.

Luego de un reconocimiento del terreno, se estableció el lugar de lanzamiento y se realizaron dos tiros de reglaje, para reubicar el lugar de la rampa.

A las 11:45, el cohete **R2D2 elevó al CANSAT 2015**, culminando así el trabajo de varios meses de construcción, pruebas y preparaciones de todos los sistemas.

El vuelo no alcanzó la altura deseada para cumplir el objetivo primario, pero se rescató la carga útil y se recuperaron los datos.



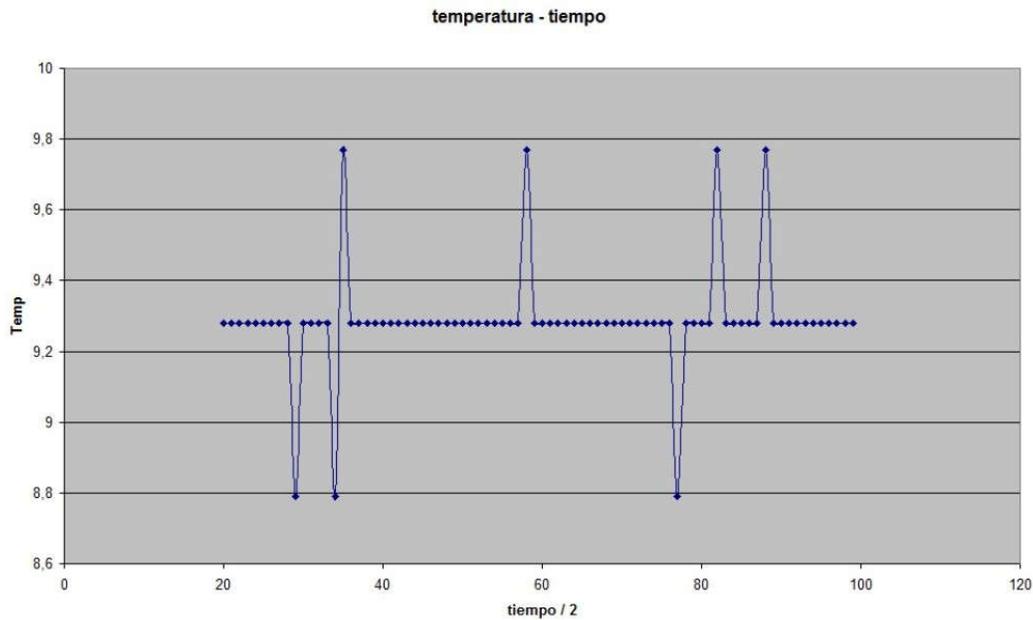
Para acceder al detalle de las fotos y un video de las actividades pueden consultarse los siguientes vínculos:

<https://picasaweb.google.com/102610685240741156409/Cansat2015#slideshow/6171526547686936418>

<https://youtu.be/3tq2dOckhgl>

Datos obtenidos

La placa Arduino Micro funciona con memoria EPROM (no es volátil frente a un fallo de alimentación eléctrica), por lo que el mismo día pudimos rescatar los datos de temperatura obtenidos durante el vuelo, con lo que se construyó este perfil térmico:



El sensor independiente de altitud y temperatura realizó el siguiente registro:



De los gráficos se desprende que los datos de temperatura se ajustan bastante (alrededor de los 10°C) y la altura alcanzada fue de 35 metros (cada división corresponde a una altura de 10 m). El dato de altura parte de -50 ya que en el montaje se encendió y apagó varias veces el altímetro, ocasionando la descalibración del sensor de presión.

JORNADA DE LANZAMIENTO – FASE 2

A partir de los resultados parciales de la Fase 1, se diseñó y construyó un nuevo cohete para repetir la experiencia en busca de cumplir con los objetivos primarios. En esta oportunidad, se construiría una bahía de carga dentro del cohete para el experimento.

El día 3 de diciembre de 2015 por la mañana, partimos nuevamente hacia el campo de vuelo de la Agrupación Aeromodelista “Pucará”.

Se reasignaron los roles de campo para corregir los errores de la primera fase. Luego de un lanzamiento o tiro de reglaje, se ubicó la rampa principal con el disparador.

Hacia el mediodía, el cohete **BB8 elevó al CANSAT 2015 (12B)**. El vuelo alcanzó esta vez la altura deseada para cumplir el objetivo primario, estableciendo un nuevo record, rescatándose la carga útil luego de un extenso rastrillaje y se recuperaron los datos.

Para acceder al detalle de las fotos y un video de las actividades pueden consultarse los siguientes vínculos:

<https://picasaweb.google.com/102610685240741156409/CANSAT2015Fase231215#slideshow/6224794387797804626>

<https://youtu.be/zvS6CkjlFgC>

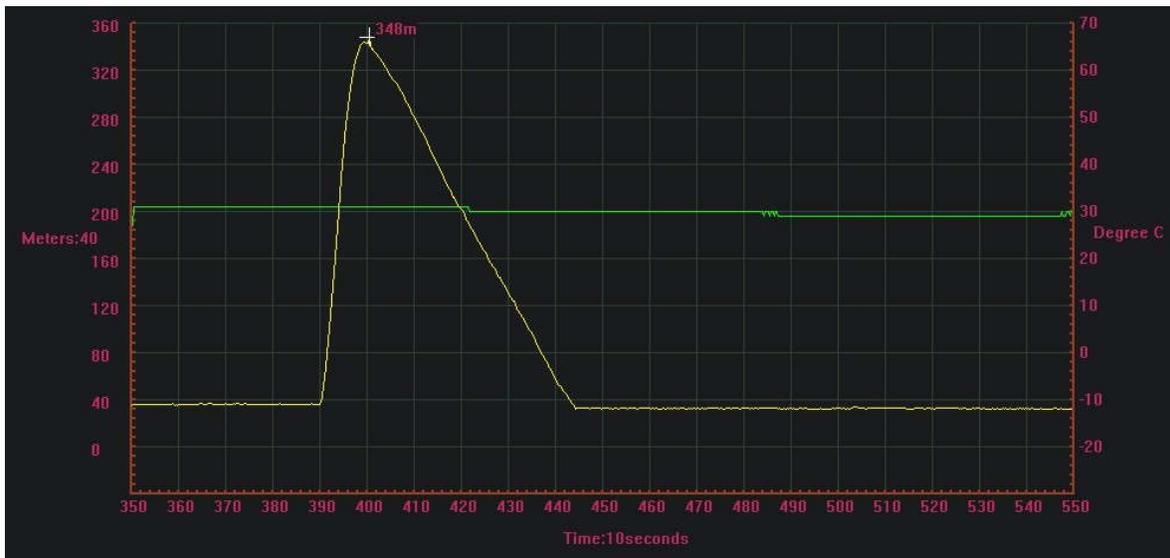


Datos obtenidos

Se obtuvieron datos de temperatura y altitud e imágenes desde el apogeo con una microcámara situada a bordo:



El sensor independiente de altitud y temperatura realizó el siguiente registro:



Puede notarse que el conjunto llegó al apogeo en menos de 10 segundos, alcanzando una altura de 310 metros (348 m– 38 m snm), registrando variaciones de temperatura de 31°C a 29°C.

ALGUNAS LECCIONES APRENDIDAS

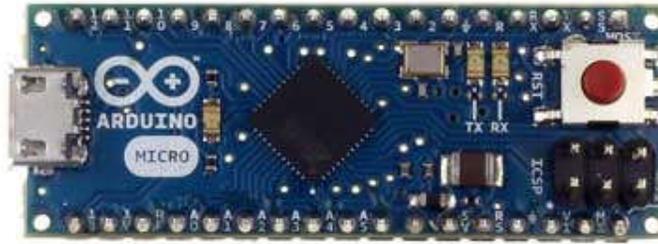
A partir de dos reuniones posteriores a las jornadas de lanzamiento, pudieron establecerse algunas conclusiones preliminares:

- Todo el trabajo fue un desafío permanente para los alumnos, y el aprendizaje a partir del trabajo realizado fue significativo.
- Es necesario enseñar una metodología acorde con el desafío tecnológico que implica el diseño, la construcción y la prueba de una carga útil.
- Si bien se aplicó una metodología, no se respetó adecuadamente a la hora de realizar las tareas.
- La adecuada planificación de los tiempos y el cumplimiento de un calendario es esencial para tener el suficiente margen temporal para corregir errores.
- Es necesario comenzar la capacitación al comienzo del ciclo lectivo, para culminar necesariamente hacia el final del mismo, de esta forma se mantiene el entusiasmo y permite el adecuado cierre de actividades.
- Es fundamental someter a pruebas todos los componentes del sistema completo, para minimizar los errores al momento de lanzamiento.
- Para simplificar el proceso, puede separarse la construcción del vector y los sistemas de soporte en tierra (tarea que la ACEMA puede realizar) y concentrarse en la carga útil.

Ing. Pablo M. González
Docente ISFN – ORT - UTN

ANEXO TECNICO

ARDUINO MICRO



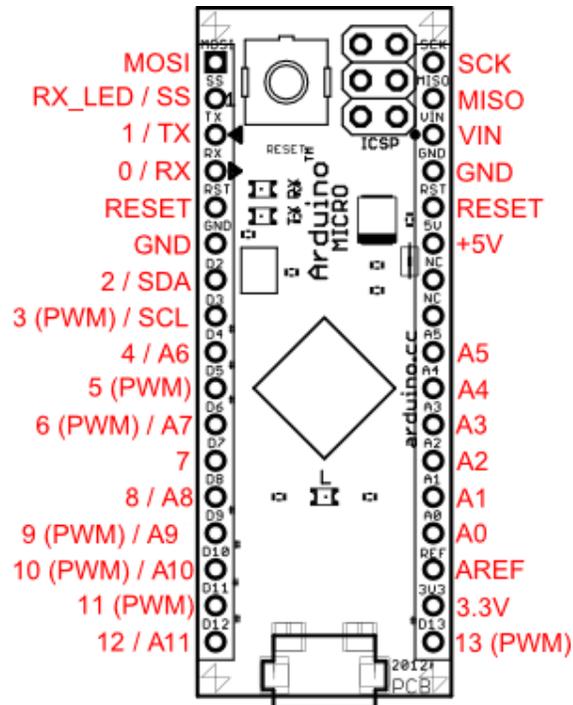
Overview

The Arduino Micro is a microcontroller board based on the ATmega32u4 ([datasheet](#)), developed in conjunction with [Adafruit](#). It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a micro USB cable to get started. It has a form factor that enables it to be easily placed on a breadboard.

The Micro is similar to the Arduino Leonardo in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Micro to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port.

Summary

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
Length	48 mm
Width	18 mm
Weight	13 g



Pin Mapping of the Arduino Micro displays the complete functioning for all the pins, to use them as in the Leonardo.

Communication

The Micro has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega32U4 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). The 32U4 also allows for serial (CDC) communication over USB and appears as a virtual com port to software on the computer. The chip also acts as a full speed USB 2.0 device, using standard USB COM drivers. [On Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on others Micro's digital pins.

The ATmega32U4 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

The Micro appears as a generic keyboard and mouse, and can be programmed to control these input devices using the [Keyboard and Mouse](#) classes.

PROGRAMACIÓN

Registro en vuelo

```
#include <EEPROM.h>
// Declaracion de variables
float tempC;
float t;
int tempPin = 0;
int led = 13;
int retardo = 100;
int despegue= 120000; // dos minutos para despegue

void setup() {

    pinMode(led, OUTPUT);
    Serial.begin(9600);

    for (int i = 0; i < 10; i++){ // parpadeo de comienzo

        digitalWrite(led, HIGH);
        delay(retardo);
        digitalWrite(led, LOW);
        delay(retardo);
    }
    delay(despegue);

    for (int i = 0; i < 200; i++){ // 200 datos

        tempC = analogRead(tempPin);

        t = (5.0 * tempC * 100.0)/1024.0;

        Serial.print(t); // SACAR
        Serial.print(" grados Celsius\n"); // (solo para monitor)

        digitalWrite(led, HIGH); // dos datos por segundo
        delay(250);
        digitalWrite(led, LOW);
        delay(250);

        EEPROM.write(i, tempC);
    }
}

void loop()
{
}
```

Lectura en tierra

```
#include <EEPROM.h>

int a = 0;

float value;
float t;

void setup()
{
  Serial.begin(9600);

  for (int i = 0; i < 200; i++){
    value = EEPROM.read(i);
    // conversion t = (5.0 * value * 100.0)/1024.0;

    Serial.print(i);
    Serial.print(";");
    Serial.print(t,2);
    Serial.println();

    delay(500);
  }
}

void loop()
{
}
```

HK-A



The HobbyKing®™ Altimeter is a stand-alone, small, light and convenient unit for recording altitude and temperature changes throughout your flight. Once you have finished flying, you simply plug the altimeter into your computer via the supplied USB interface and download all the recorded flight data which displays an Altitude/Temperature/time graph for each flight. Powered by your RX, the H-King Altimeter is small and light enough to be fitted to almost any model and comes with mounting foam & instructions.

Specs:

Dimensions: **21x13x5mm**

Weight: **1.3g**

Connector: **1.25mm pin**

Operating Voltage: **3.7~8.4v**

Working Current: **<7mA**

Upload Interface: **USB**

Height Range: **-500~9000m**

Height Accuracy: **0.3m**

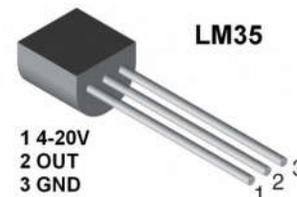
Operating Temp Range: **-20c~65c**

Temperature Accuracy: **1c**

Sampling Frequency: **1hz/2hz/4hz/8hz**

Storage Capacity: **63h(1hz)**

LM35 (sensor de temperatura)



<http://pdf.datasheetcatalog.net/datasheet/nationalsemiconductor/DS005516.PDF>