

CANSAT BELGIUM 2016

see if life is possible on the atmosphere

VERSION
1.1



23/01/2016
a CAN in a SATellite

By INRASAT

Sacha Plat • Ahmed Ahmedov • Jérémie Sikyala
Ataberk Vuran • Ryan Danenberg • Yesukhei Delgermaa



Table of Contents

1. Introduction

1.1 Team and job organization.....	3
1.2 Mission objectives.....	5

2. Cansat description

2.1 Mission overview.....	6
2.2 Mechanical / structural design.....	7
2.3 Electronics design.....	9
2.3.1 Block diagram.....	9
2.3.2 Main diagram.....	9
2.4 Software design.....	10
2.5 Recovery system.....	15
2.6 Ground support equipment.....	15

3. Project Planning

3.1 Time schedule of the Cansat preparation.....	16
3.2 Resources estimation.....	17
3.2.1 Budget.....	17
3.2.2 External support.....	17
3.3 Test plan.....	18

4. Awareness programme

5. Restriction

5.1 Restriction for Cansat.....	19
5.2 Restriction for transceivers.....	19

1. Introduction

1.1 Team and job organization

Students:

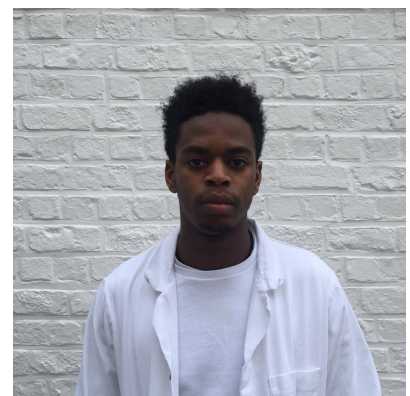
- Sacha Plat : English
- Ahmed Ahmedov : Solidworks
- Jérémie Sikyala : Solidworks + Mechanics
- Ataberk Vuran : English + Multipurpose
- Ryan Danenberg : Programming
- Yesukhei Delgermaa : Solidworks + Mechanics



Sacha Plat



Ahmed Ahmedov



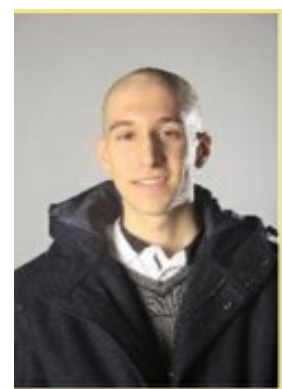
Jérémie Sikyala



Ryan Danenberg



Yesukhei Delgermaa



Ataberk Vuran

Tasks and areas of teachers

- PCB realisation and Cansat programming (Arduino) : Jöel Pochet (IT teacher)
- YAGI realisation : Therry Preys (CFAO teacher)
- Parachute realisation : Jean Morren (HELB's teacher)
- Programming of the ground station : Olivier Deldime (deputy director)
- Study of the electronic diagrams : Roberto Mazzeo (electronics teacher)
- Study of the physical and chemical aspects : Thierry Slagmolen (Science teacher)
- Writing of the intermediate reports in English : Jill Aerts (English teacher)
- Project promotion : the computer graphics section will realize a logo, a webpage and some t-shirts
- Pictures of the construction of the antenna and the Cansat : phoyography section from the school
- General coordination : Jöel Pochet and Roberto Mazzeo

Contact : - jpochet@gmail.com

- roberto.mazzeo@inraci.be

1.2 Mission Objectives

Because of the huge pollution, we will send a probe in the atmosphere to see if life is possible over there.

Primary mission:

The goal of the primary mission is to calculate the temperature and the pressure in the atmosphere. For this we will use three sensors (CTN: Fast Thermistore, LM35 : Slow temperature sensor and MPX4415: Pressure sensor).

- The different sensors are welded and tested.
- The Arduino program is realized.
- Some measures of temperature and pressure are made.
- The sending of information to Putty via transceivers works.

Secondary mission:

The secondary mission completes the primary mission, it consists in adding some modules in the can.

- Measure of the acceleration during the climb and during descent.
- Measure of the oxygen-level at its highest and during descent.
- Measure of the humidity at its highest and during descent.
- A buzzer will activate when the can arrives on the ground.
- Ground station with antenna YAGI which will draw graphs of t° , pressure, oxygen rate, humidity and acceleration according to the time.

According to the budget:

If the budget, the weight and the available place in the can allow it, we shall install additional modules.

- A GPS to facilitate the search of the can when it lands.
- A camera to take shots from the sky

2. Cansat description

2.1 Mission overview

The goal of the mission is to send a can with sensor modules to the atmosphere. The mission is divided in two parts, primary mission and secondary mission.

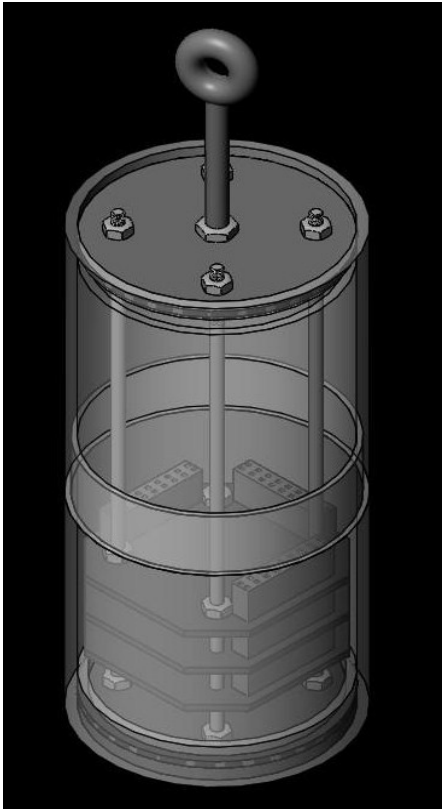


Key

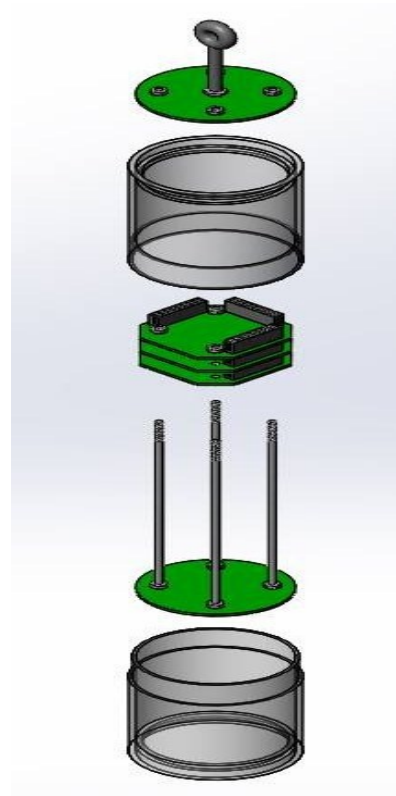
1. Take-off of the rocket.
2. Release of the can, beginning of the measurements and photos.
3. Opening of the parachute and beginning of the descent.
4. Landing of the can, launching of the buzzer.
5. The ground station (PC, transceiver and antenna) will receive and will display all measures.

2.2 Mechanics

Closed view of the can

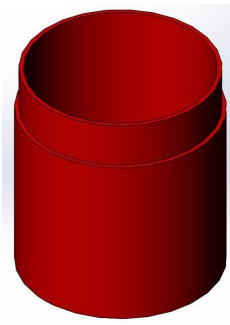
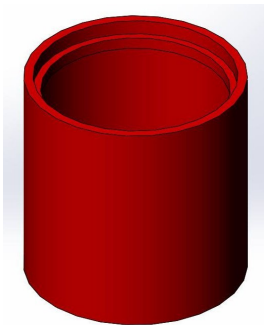


Fragmented view

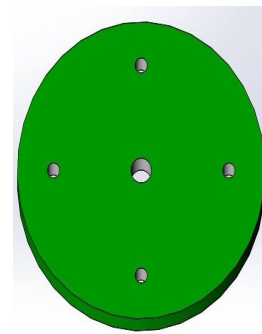
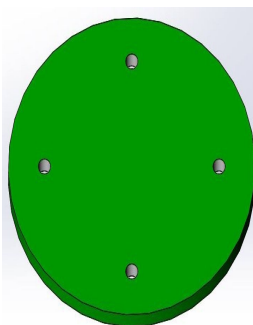


Drawing of the various parts

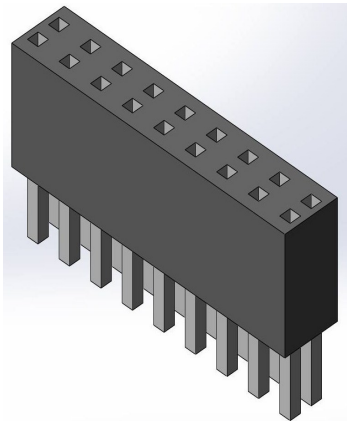
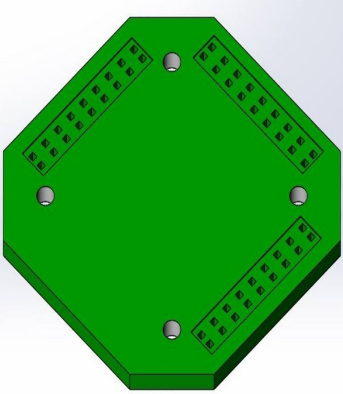
Case



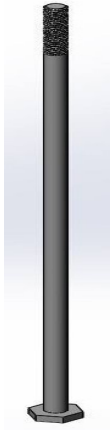
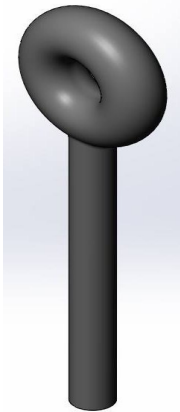
Base case



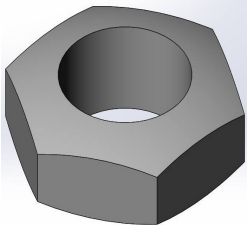
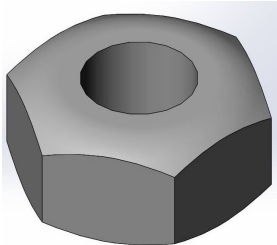
PCB and pins



Hook and support

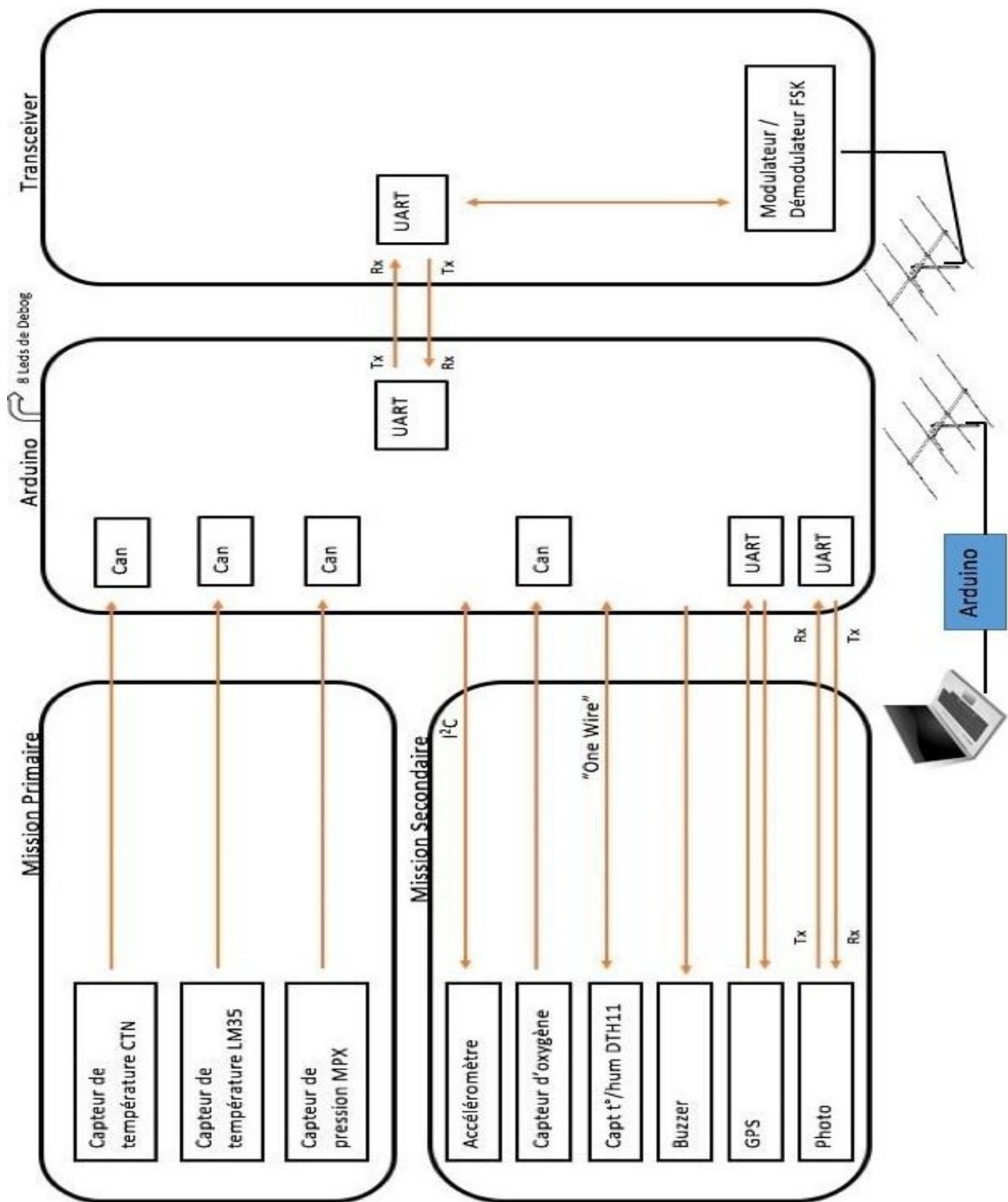


Nuts



2.3 Electronics design

2.3.1 Block diagram



2.3.2 Schematics

See later ...

2.4 Software design

A course on processing will be organized on Tuesday, 02/02/16 by Mr Deldime

The Arduino card of the Cansat will be programmed in code C.

The PC application of the ground station will be scheduled in JAVA with Processing.

Program which sends the temperature and the pressure (J. Pochet):

```
void screen(const char *msg)
{Serial.print(msg); }

void screen_val(float val)
{Serial.println(val); }

void send_ER(const char *msg)
{Serial1.print(msg); }

void send_val_ER(float val)
{Serial1.println(val); }

float ctn_temp(int val)
{
float temp = 0;
float tension = ((float)val * 4.88); //mV
float resistance = (50000000.0 / tension) - 10000.0;
float rapport_mesure = resistance / 10000; //10000 <-> 25 celcius
float borne_res_petite, borne_res_grande, borne_temp_min;
if (rapport_mesure < 0.5330)
temp = 40;
else if (rapport_mesure > 5.505)
temp = -10;
else if (rapport_mesure <= 5.505 && rapport_mesure > 4.216)
{
borne_res_petite = 4.216;
borne_res_grande = 5.505;
borne_temp_min = -10;
}
else if (rapport_mesure <= 4.216 && rapport_mesure > 3.255)
{
borne_res_petite = 3.255;
borne_res_grande = 4.216;
borne_temp_min = -5;
}
else if (rapport_mesure <= 3.255 && rapport_mesure > 2.534 )
{
borne_res_petite = 2.534;
borne_res_grande = 3.255;
borne_temp_min = 0;
}
else if (rapport_mesure <= 2.534 && rapport_mesure > 1.987 )
{
borne_res_petite = 1.987;
borne_res_grande = 2.534;
borne_temp_min = 5;
}
}
```

```

else if (rapport_mesure <= 1.987 && rapport_mesure > 1.57 )
{
borne_res_petite = 1.57 ;
borne_res_grande = 1.987;
borne_temp_min = 10;
}
else if (rapport_mesure <= 1.57 && rapport_mesure > 1.249 )
{
borne_res_petite = 1.249;
borne_res_grande = 1.57;
borne_temp_min = 15;
}
else if (rapport_mesure <= 1.249 && rapport_mesure > 1 )
{
borne_res_petite = 1;
borne_res_grande = 1.249 ;
borne_temp_min = 20;
}
else if (rapport_mesure <= 1 && rapport_mesure > 0.8059 )
{
borne_res_petite = 0.8059;
borne_res_grande = 1 ;
borne_temp_min = 25;
}
else if (rapport_mesure <= 0.8059 && rapport_mesure > 0.6535 )
{
borne_res_petite = 0.6535;
borne_res_grande = 0.8059;
borne_temp_min = 30;
}
else if (rapport_mesure <= 0.6535 && rapport_mesure > 0.5330 )
{
borne_res_petite = 0.5330;
borne_res_grande = 0.6535;
borne_temp_min = 35;
}
if (temp==0)
{
temp = borne_temp_min ;
temp += 5*(borne_res_grande - rapport_mesure)/(borne_res_grande-borne_res_petite);
}
return temp;
}

float LM35_temp(int val)
{
float tension = ((float)val * 4.88); //mV
return tension / 10;
}

float give_pression(int val)
{
float tension = ((float)val * 4.88); //mV
return (tension + 475.0)/ 45.0; // en kPa
}

```

```

void setup() {
pinMode(16, OUTPUT); // init LED
Serial.begin(9600);
Serial1.begin(19200); // init serial1 pour ER
// serial is the USB connection
// for Transmitter Serial1, Serial2 or Serial 3 are used
// dependant on the used poort
}

void loop() {
int val;
float temp_ctn = 25;
float temp_LM35 = 25;
float pression = 101.325;
pression = give_pression(analogRead(A0));
temp_LM35 = LM35_temp(analogRead(A1));
temp_ctn = ctn_temp(analogRead(A2));
screen("temperature ctn en CELCIUS: ");
screen_val(temp_ctn);
screen("temperature lm35 en CELCIUS: ");
screen_val(temp_LM35);
screen("pression en kPa : ");
screen_val(pression);
screen("\n");
send_ER("temperature ctn en CELCIUS: ");
send_val_ER(temp_ctn);
send_ER("temperature lm35 en CELCIUS: ");
send_val_ER(temp_LM35);
send_ER("pression en kPa : ");
send_val_ER(pression);
send_ER("\n");
delay(2000);
}

```

Program test of the DTH11

(https://tkkrlab.nl/wiki/Arduino_KY-015_Temperature_and_humidity_sensor_module):

```
int DHpin = 8;
byte dat [5];
byte read_data () {
  byte data;
  for (int i = 0; i < 8; i ++) {
    if (digitalRead (DHpin) == LOW) {
      while (digitalRead (DHpin) == LOW); // wait for 50us
      delayMicroseconds (30); // determine the duration of the high level to determine
                               //the data is '0 'or '1'
      if (digitalRead (DHpin) == HIGH)
        data |= (1 << (7-i)); // high front and low in the post
      while (digitalRead (DHpin) == HIGH); // data '1 ', wait for the next one receiver
    }
  }
  return data;
}

void start_test () {
  digitalWrite (DHpin, LOW); // bus down, send start signal
  delay (30); // delay greater than 18ms, so DHT11 start signal can be detected
  digitalWrite (DHpin, HIGH);
  delayMicroseconds (40); // Wait for DHT11 response
  pinMode (DHpin, INPUT);
  while (digitalRead (DHpin) == HIGH);
  delayMicroseconds (80); // DHT11 response, pulled the bus 80us
  if (digitalRead (DHpin) == LOW);
  delayMicroseconds (80); // DHT11 80us after the bus pulled to start sending data

  for (int i = 0; i < 4; i ++) // receive temperature and humidity data, the parity bit
    dat[i] = read_data (); //is not considered
  pinMode (DHpin, OUTPUT);
  digitalWrite (DHpin, HIGH); // send data once after releasing the bus, wait for the
                               //host to open the next Start signal
}

void setup () {
  Serial.begin (9600);
  pinMode (DHpin, OUTPUT);
}

void loop () {
  start_test ();
  Serial.print ("Current humidity =");
  Serial.print (dat [0], DEC); // display the humidity-bit integer;
  Serial.print ('.');
  Serial.print (dat [1], DEC); // display the humidity decimal places;
  Serial.println ('%');
  Serial.print ("Current temperature =");
  Serial.print (dat [2], DEC); // display the temperature of integer bits;
  Serial.print ('.');
  Serial.print (dat [3], DEC); // display the temperature of decimal places;
  Serial.println ('C');
  delay (700);}
```

Program test for the measuring of oxygen

([http://www.seeedstudio.com/wiki/Grove_-_Gas_Sensor\(O%E2%82%82\)](http://www.seeedstudio.com/wiki/Grove_-_Gas_Sensor(O%E2%82%82))):

```
#include <math.h>
void setup()
{
  Serial.begin(9600); //Start the Serial connection
}
void loop()
{
  float sensorValue;
  float sensorVoltage;
  float Value_O2;
  sensorValue = analogRead(A0);
  sensorVoltage = (sensorValue/1024)*5.0;
  Serial.print("Tension sensorVoltage is ");
  Serial.print(sensorVoltage);
  Serial.println("V");
  sensorVoltage = (sensorVoltage/121)*10000; //sensorVoltage = sensorVoltage/201*10000;
  Serial.print("Courant sensor is ");
  Serial.print(sensorVoltage);
  Serial.println("uA");
  Value_O2 = sensorVoltage/7.43;
  Serial.print("Concentration of O2 is ");
  Serial.print(Value_O2,1);
  Serial.println("%");
  Serial.println(" ");
  delay(1000);
}
```

2.5 Recovery system

To facilitate the recovery of the can, it will be equipped with a buzzer which will emit a sound signal when it arrives on the ground.

The can will be equipped with a parachute which will it slow down so that we have the time to take measures when it is high up.

We receive the assistance of Mr Morren for the conception of the parachute and the calculation of its size. The beginning its realization began on Friday 23/01.

Formula to calculate the scale of the parachute:

$$D = \frac{6,1\sqrt{m}}{V}$$

m = the mass = 0,35 Kg

V = speed = 8 m/s

D = diameter = 45 cm

2.6 Ground support equipment

On the ground, it will be a laptop with the transceiver and the YAGI antenna.

On the laptop, a program will show the different measures (temperature, pressure, acceleration, rate of oxygen, rate of humidity) according to the time in the form a graph.

3. Project Planning

3.1 Time schedule of the cansat preparation

	Designed	Prototyped	Constructed	Tested
Primary mission				
Electronics	V	V	V	V
Mechanics	V	V		
Software	V	V	V	V
Secondary mission				
MPU6050	V	V	V	
Oxygen	V	V	V	V
DTH11	V	V	V	V
Buzzer	V	V	V	V
YAGI	V			

3.2 Ressource estimation

3.2.1 Bugdet

Total budget = 500€

Kit = 300€

Rest = 200€

The current spendings are :

- Oxygen sensor : 37,98 euro (Seedstudio)
- Accelerometer (MPU6050): 1,59 euro (Aliexpress)
- Humidity sensor (DTH11) : 1,35 euro (Aliexpress)
- Buzzer : 2,60 for a batch of 20 buzzer (Aliexpress)
- Transistor BS170 : 0,28 euro (RS components)
- 3 * Condenser 100nF: 3 * 0,22 euro = 0,66 euro (RS components)

Total = 44,46€

Still to plan:

- PCB, Connector,..
- Battery
- Equipment and cable for the antenna
- Equipement for the parachute
- ...

3.2.2 External support

For the moment, all the coaching is made by the professors of the Inraci (electronics, computer graphics and photography section).

We also receive some help from Mr Jean Morren who was chosen for his interest for the project and for his experience about parachutes.

3.3 Test plan

When the YAGI antenna will be realized, we shall make a test to check the data reception.

At the reception of the sensors (accelerometer, oxygen, humidity, buzzer), for safety, we made a test with an ARDUINO card before integrating them into the project.

The sensors work normally, but it we have to calibrate it. (test mesure : 23,4 % O₂)

The LM35 is a temperature-sensor whose response time is very slow.

4. Outreach programme

Web site, Logo and T-shirt (computer graphics section).

<http://cansat.inraci.be>

A facebook page is going to be created and livened up by Jérémie Sikyala.

5. Restriction

5.1 Restriction for Cansat

Characteristics	Restriction
Height of the Cansat	115 mm
Weight of the Cansat	350 g
Diameter of the Cansat	66 mm
Current max available	800 mA
Autonomy	3 hours
Acceleration resistance	20 G

5.2 Restriction for transceivers

Characteristics	Restriction
Carrier	434,43 MHz
Power max of broadcast	13 dBm = 20mW
Air baudrate	19200 bps
USB baudrate	19200 bps
Quarter of wave antenna	17 cm