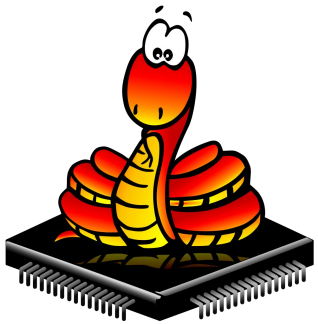


Sparrow
MicroPython

Thursday, October the 3rd 2023

Antoine Chauvin & Hugo Reimeringer

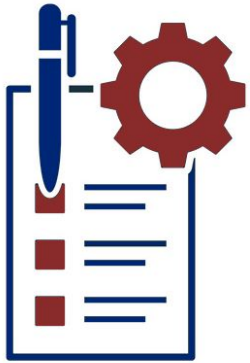
Summary

I - Save the data on the card

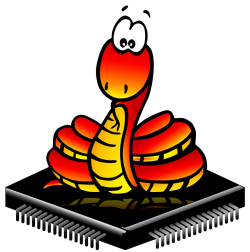
II - Code !

- a) CNES's Specifications

- b) Our Specifications



I - Save data on the card



Save data on the card

During the launch, we want to save the data collected, so that we can understand what happened and correct it for the next launch if necessary
We will store them in the flash memory of the raspberry pi pico, which has 2 MB of storage.

```
0.0,1013.307,-0.385772,-1.542367,1.414415,19.28628,21.20769,-19.78714,139.6866,0
0.132,1013.314,-0.443902,-2.102703,13.03668,-12.02016,-2.212099,-58.26637,131.8504,0
0.264,1013.307,-0.385772,6.03218,-11.90064,18.87427,21.05928,-31.19217,112.9546,0
0.408,1013.319,-0.4861784,-2.381694,-5.525634,17.1397,20.76881,26.6137,129.6129,0
0.54,1013.323,-0.5231702,-0.321045,0.4061622,9.144312,16.0347,21.47535,130.9675,0
0.672,1013.316,-0.4597557,-0.2674835,0.1248168,9.166679,10.95048,16.05036,132.0242,0
0.803,1013.327,-0.5548775,-0.3051532,0.0953874,9.097815,8.289967,12.72939,132.4086,0
0.971,1013.301,-0.3382111,-0.366955,0.09244445,9.061321,6.413312,10.18528,132.6009,0
1.103,1013.315,-0.4491866,-0.3981502,0.1254054,9.05367,4.864625,8.077888,133.4883,0
1.234,1013.317,-0.4703248,-0.4122763,0.1089249,9.071917,3.618366,6.404073,134.1274,0
1.366,1013.328,-0.5601621,-0.4205165,0.08655857,9.073682,2.76055,5.193414,134.3977,0
1.498,1013.313,-0.4333329,-0.4110991,0.1112793,9.053082,1.812851,4.098213,134.9435,0
1.629,1013.319,-0.4861784,-0.4093333,0.1248168,9.080745,1.291156,3.375273,135.2252,0
1.765,1013.322,-0.5126011,-0.4258138,0.0747868,9.055436,0.8082389,2.821803,135.306,0
1.896,1013.316,-0.4650402,-0.4634835,0.09067868,9.098402,0.4902275,2.422391,135.2634,0
2.028,1013.319,-0.4861784,-0.4058018,0.1012733,9.091928,0.2468318,2.068096,135.3809,0
2.16,1013.327,-0.5548775,-0.416985,0.07596397,9.124889,-0.1064008,1.790239,135.4626,0
2.292,1013.307,-0.385772,-0.3899099,0.105982,9.104877,-0.4843437,1.623716,135.3328,0
2.423,1013.309,-0.4016256,-0.4193394,0.0794955,9.086631,-0.4595116,1.44393,135.3358,0
2.555,1013.323,-0.5178857,-0.4110991,0.06124925,9.065442,-0.614615,1.296664,135.2601,0
2.687,1013.31,-0.4121947,-0.4034474,0.09009009,9.098402,-0.8636005,1.234417,135.2179,0
2.818,1013.324,-0.5284548,-0.4134535,0.1101021,9.064853,-0.4177602,1.216483,135.2006,0
2.95,1013.312,-0.4280484,-0.4216937,0.07360962,9.04543,-0.6058527,1.14023,135.1824,0
3.082,1013.314,-0.4491866,-0.4628949,0.0747868,9.102523,-0.3993829,1.069544,135.1893,0
3.213,1013.309,-0.4069102,-0.4010931,0.09479879,9.078391,-0.2133336,1.044217,135.1714,0
3.348,1013.314,-0.443902,-0.4063904,0.1142222,9.097815,-0.4789176,1.08496,135.1588,0
3.48,1013.323,-0.5231702,-0.477021,0.09891892,9.087808,-0.4417304,1.07609,135.1387,0
3.612,1013.315,-0.4544711,-0.4352313,0.0794955,9.07015,-0.1284059,1.032572,135.114,0
3.755,1013.331,-0.5865848,-0.39403,0.09715314,9.053082,-0.6087778,1.024658,135.1092,0
```

An example of a csv file containing all the data collected from the card

Save data on the card

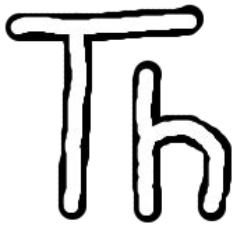
Exercise : Save the acceleration measured by the IMU

One line for each 0.1s

Given functions :

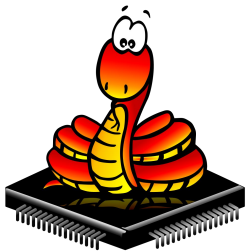
```
file = open("data.csv", "w") #creation and opneinf of a CSV file in writing mode
#Type Program Logic Here
file.write(str(value) + ",") #writing data in the opened file
file.close()                 #the file is closed

timer = time.ticks_ms()      #returns an increasing counter with an arbitrary reference point
dt = time.ticks_diff(t2, t1) #substraction t2-t1 for ticks_ms() values
```



MicroPython

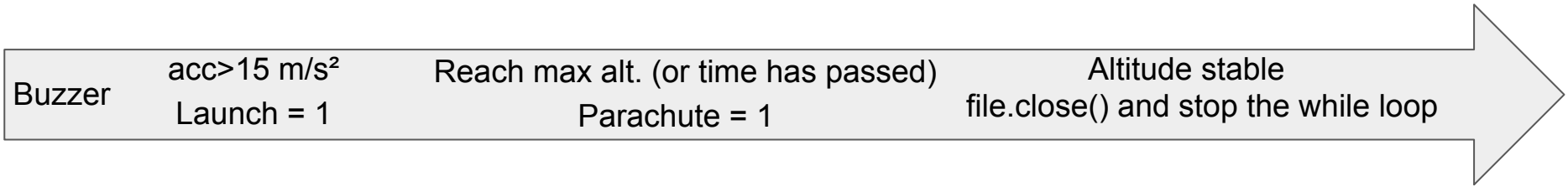
II - Code !



CNES's specifications

- The parachute is deployed when reaching the maximum of altitude (move the servo motor).
  - For safety reasons, we check with a timer that its deployment is within a certain range of time so that it does not deploy either too soon or too late in the case of a false measure.
  - The sensors have some noise on the measures. Therefore, we consider that we have reached the maximum of altitude when 5 values of altitude in a row are decreasing.
- Before the launch, you need to have a buzzer on the rocket that indicates that the card is ready for takeoff. (But we don't have it yet 😢 so you can just add a useless function start_buzzer to remember)

The values are arbitrary and can change with the calculations you'll make on the rocket later



Our specifications

- We consider that the rocket is launched when we measure an acceleration superior to 15 m/s²
- We consider that the rocket has landed when the altitude is stable with an error of 1 meter for 10 values in a row. At this point, we can stop close the data file and stop the while loop.
- Split the code in different files (+ add comments to your code) :
  - mpu9250.py
  - lps22hbtr.py
  - servo.py
  - main.py
- Make sure that you have enough storage capacity (2MB) in regards to the frequency of the main loop (for a margin, you can take a duration of flight of 3 minutes)

On obtient la correction d'altitude en fonction de la pression atmosphérique :

$$z - z_1 = \frac{T_0}{a}\left(1 - \left(\frac{p}{p_1}\right)^{\frac{Ra}{Mg}}\right)$$

L'argument de la fonction est p, la valeur est z. Les autres symboles représentent des constantes. Numériquement,

$$z - z_1 = 44330.8m\left(1 - \left(\frac{p}{p_1}\right)^{0.190289}\right)$$

https://www.deleze.name/marcel/sec2/applmaths/pression-altitude/pression-altitude.pdf

The values are arbitrary and can change with the calculations you'll make on the rocket later