

```

#!/usr/bin/python2.7
# -*- coding: utf-8 -*-
# Balise meteo radioamateur sur raspberry et python 2.7 par F5SAZ
#pression atmosphérique, température, force et direction du vent.*
#*****version 1.85 bêta 2016*****
#*****


import Adafruit_BMP.BMP085 as BMP085
import datetime
import time
import locale
import os
import sys
import RPi.GPIO as GPIO
from subprocess import call
sys.path.insert(0, "/home/pi/yoctopython/")          # suivant config sur votre
micro
from yocto_api import *
from yocto_anbutton import *

errormsg=YRefParam()

locale.setlocale(locale.LC_TIME, '')                  # Francisation de la date

GPIO.setmode(GPIO.BCM)                                # utilisation des n° de ports
du processeur
GPIO.setup(23, GPIO.OUT)                               # mise en sortie du port GPIO23
(broche 16)

i = 0

while 1:

    # mise en veille de la balise de 22h à 06h00

        v = datetime.datetime.now().strftime('%H')      # control de l'heure

        if v == "22":                                  # 22h00 dodo jusqu'à 6h00
            time.sleep(28800000)                      # fin du dodo 8 heures plus
tard

    # reglage de la tempo toutes les 15 minutes

        h = datetime.datetime.now().strftime('%M,%S') # control des minutes et
secondes

        if h == "15,00" or h == "30,00" or h == "45,00" or h == "00,00":

    # initialise l'api

        if YAPI.RegisterHub("usb", errormsg)!= YAPI.SUCCESS:
            sys.exit("init error"+errormsg.value)

    # cherche l'entree avec le nom logique entree1

        channel= YAnButton.FindAnButton('direction')
        channel2= YAnButton.FindAnButton('vitesse')

    # verifie que l'entre a bien été trouvée

```

```

if not(channel.isOnline()):
    sys.exit("pas trouvé d'entrée avec le nom logique 'direction' ")

if not(channel2.isOnline()):
    sys.exit("pas trouvé d'entrée avec le nom logique 'vitesse' ")

# affiche la valeur de l'entree en question

source = (channel.get_calibratedValue())      #variable direction du vent
canal 1

pulse = (channel2.get_pulseCounter())          #variable impulsion
tempo = (channel2.get_pulseTimer())            #variable temps

# rose des vents!!

if source <= 945 and source >= 893:           #ces valeur sont a adapter
suivant
    direction = "nord"
elif source <= 892 and source >= 796:          #la girouette utilise
    direction = "nord nord este"                 #cardineaux en phonetique!!
elif source <= 795 and source >= 709:
    direction = "nord este"
elif source <= 708 and source >= 629:
    direction = "este nord este"
elif source <= 628 and source >= 556:
    direction = "este"
elif source <= 555 and source >= 490:
    direction = "este sude este"
elif source <= 489 and source >= 403:
    direction = "sude este"
elif source <= 402 and source >= 322:
    direction = "sude sude este"
elif source <= 321 and source >= 272:
    direction = "sude"
elif source <= 271 and source >= 228:
    direction = "sude sude oueste"
elif source <= 227 and source >= 186:
    direction = "sude oueste"
elif source <= 185 and source >= 147:
    direction = "oueste sude oueste"
elif source <= 146 and source >= 110:
    direction = "oueste"
elif source <= 109 and source >= 76:
    direction = "oueste nord oueste"
elif source <= 75 and source >= 44:
    direction = "nord oueste"
elif source <= 43 and source >= 14:
    direction = "nord nord oueste"
elif source <= 13:
    direction = "nord"

# vitesse du vent

p = pulse / 5.                                #sur mon anemo.

```

```

t = temps /1000.                                     #conversion de miliseconde en
seconde

f = p / t                                         #determination de la frequence
                                                #formule: 2*pi*r*f = en metre

par seconde

vitvent = 2*3.14*0.070*f*3600/1000                #vitesse vent en Km/h

vitevt = int(vitvent)                                #conversion en entier pour
supprimer la virgule

vitv = str(vitevt)                                   #conversion des chiffres en
texte

# TEMPERATURE et PRESSION

sensor = BMP085.BMP085()

degre_c = sensor.read_temperature()                 #variable temperature
pression = sensor.read_pressure()/100 + 5          #variable pression (+5) a
ajuster

degre_b = int(degre_c)                               #conversion en entier pour
supprimer la virgule

tp = str(degre_b)                                    #conversion des chiffres en
texte
pr = str(pression)                                 #pour en chaine de caracteres

# DATE et HEURE

ce_jour = time.strftime("%A %d %B %Y ")           #variable date
cette_heure = time.strftime("%H")                  #variable heure
cette_minute = time.strftime("%M")                 #variable minute

# MISE EN FORME DES PHRASES

phrase0 = "balise météo experimentale du radioclub de granville F 5 K
A Q "
phrase1 = ce_jour + cette_heure+ " heure "+cette_minute+ " minute "
phrase2 = "température "+tp+" degré celcius "
phrase3 = "préssion atmosphérique "+pr+" hectopascale "
phrase4 = "vent de "+direction+ " vitesse "+vitv+" kilométreheure"

phrase =(phasel)+tp+" "+pr+" "+vitv+" "+direction    #phrase à
enregistrer

phrase9 =(phrase2)+(phrase3)                         #3eme phrase à emettre

# GESTION du FICHIER

fichier_meteo = open("meteo.txt", "a")            #ouverture du fichier
meteo.txt
fichier_meteo.write("\n")                          #retour a la ligne
fichier_meteo.write(phrase)                        #ecriture de la phrase
fichier_meteo.close()                            #fermeture du fichier

# GESTION DE LA PAROLE ET PASSAGE EN EMISSION

```

```

phrase5 = phrase0.replace(" ", "-")           #remplace les espace par des -
pour espeak
phrase6 = phrasel.replace(" ", "-")
phrase7 = phrase9.replace(" ", "-")
phrase8 = phrase4.replace(" ", "-")

GPIO.output(23, GPIO.HIGH)                   #emission

time.sleep(1)                                #pause 1 seconde

call(["espeak", "-vfr+15", "-p60", "-s170", phrase5])      #voie femme en
francais
call(["espeak", "-vfr+15", "-p60", "-s170", phrase6])      #p pour la
tonalité
call(["espeak", "-vfr+15", "-p60", "-s170", phrase7])      #s pour la
vitesse de diction
call(["espeak", "-vfr+15", "-p60", "-s170", phrase8])

GPIO.output(23, GPIO.LOW)                     #arret emission

channel2.resetCounter()                      #remise a zero du compteur
YAPI.Sleep(780000)                          #pause de 13 minutes

```