2023

Compteur de palonniers

OPTIONS AUTO-EXTINCTION

Farid ADDA

#include <SoftwareSerial.h>

#include <simpleBouton.h>

#include "Heures.h"

SoftwareSerial Serial(0,4); // RX ,TX

#define automaintien 1

simpleBouton capteur(2);    // bouton en broche D2 Cablage : pin---BP---GND

Heures heures;

unsigned long previousTime=0;

unsigned long currentTime=0;           // Temps millis() en cours

int minuterie = 0;                     // Temps de fonctionnement

int duree =3600;                       // Temps de réarmement de la minuterie

int compteur=0;                        // Compteur

String str="";                         // Lecture du port série

void setup()

{

  Serial.begin(9600);

  Serial.setTimeout(100);

  pinMode(automaintien,OUTPUT);

  minuterie=duree;

  digitalWrite(automaintien,HIGH);

}

void loop()

{

   str=Serial.readString();

   // Raz en appuyant sur l'écran

  if(str=="raz")

  {

    compteur=0;

    Serial.print("cpt.val=");

    Serial.print(compteur);

    ValideNextion();

  }

**// Capteur comptage**

  capteur.actualiser();

  if(capteur.vientDEtreEnfonce()&& millis()>2000)   // Evite l'incrémentation à la mise sous tension

  {

      compteur++;

      Serial.print("cpt.val=");

      Serial.print(compteur);

      ValideNextion();

      minuterie= duree ;      // Relance la minuterie à chaque passage

  }

**// Décomptage et test minuterie**

   currentTime=millis();

   if((currentTime-previousTime)>1000)

   {

        previousTime=currentTime;

        String decompte = "tps.txt=\"" + heures.Affiche(minuterie)+"\"";

        Serial.print(decompte);

        ValideNextion();

        minuterie--;

   }

   if(minuterie<=0 || str=="off")

   {

       digitalWrite(automaintien,LOW);     // Coupe l'alimentation

   }

}

**// Validation Nextion**

void ValideNextion()

{

  Serial.write(0xFF);

  Serial.write(0xFF);

  Serial.write(0xFF);

}

**Bibliothèque**

**Heure.h**

#include "Arduino.h"

class Heures

{

private:

  uint8\_t minutes;

  uint8\_t secondes;

  char minSec[5];

public:

  String Affiche(int temps);

};

**Heure.cpp**

#include "Heures.h"

String Heures::Affiche(int temps)

{

  minutes = temps / 60;

  secondes = temps % 60;

  sprintf(minSec, "%0.2d:%0.2d", minutes, secondes);

  return String (minSec);

}





