/\* Gasthon project

 \*

 \* tools.hpp

 \*

 \* members of the project :

 \* - YAGHI Anthony

 \* - SCHAAF Hugo

 \* - LANDURE Raphaël

 \*

 \* Author : SCHAAF Hugo

 \*

 \* IS3 - ENIB

 \* 01/2019

 \*/

#ifndef TOOLS\_HPP

#define TOOLS\_HPP

#include "Arduino.h"

#include <Servo.h>

// default delay to enter in emergency mode

#define DELAY 5000 // in ms

namespace tools

{

 //---------------------------------------------------------------------------------------------//

 struct Ventilation

 {

 // constructor

 Ventilation(uint8\_t pin0, uint8\_t pin1, uint32\_t wait\_t= DELAY):

 pin\_balance{pin0}, pin\_orient{pin1}, start\_time{0}, wait\_t{wait\_t},

 emergency\_count{false}, doRun{false}

 {

 };

 // run it once to initialize the ventilation system

 void begin()

 {

 servo\_balance.attach(pin\_balance);

 servo\_orient.attach(pin\_orient);

 servo\_balance.write(0);

 servo\_orient.write(0);

 };

 uint8\_t run()

 {

 // if no need to ventilate, exit

 if(!doRun) return 0;

 Serial.println("doRun");

 if(!emergency\_count)

 {

 emergency\_count = true;

 start\_time = millis();

 }

 if(millis() - start\_time > wait\_t)

 {

 Serial.println("emergency");

 servo\_orient.write(90);

 }

 servo\_balance.write(20);

 delay(250);

 servo\_balance.write(0);

 delay(250);

 return 1;

 };

 void set()

 {

 if(!doRun)

 {

 Serial.println("set");

 doRun = true;

 }

 };

 // reset to initial position

 void reset()

 {

 if(doRun)

 {

 Serial.println("reset");

 servo\_balance.write(0);

 servo\_orient.write(0);

 emergency\_count = false;

 doRun = false;

 }

 };

 // motorisation members

 Servo servo\_balance, servo\_orient;

 uint8\_t pin\_balance, pin\_orient;

 // timer members

 int32\_t start\_time;

 uint32\_t wait\_t;

 // flags

 bool emergency\_count, doRun;

 };

 //---------------------------------------------------------------------------------------------//

 inline

 void displayTemp(float temp)

 {

 static float prev\_temp = temp+1.;

 if(temp > prev\_temp + 0.3 || temp < prev\_temp - 0.3)

 {

 prev\_temp = temp;

 Serial.println("candle temperature : "+String(temp)+" °C");

 }

 };

}// namespace tools

#endif