```
/*
Auteur: Noah Anonyme
Interface: arduino
Nom du projet:
Description:
Toolbox: vittascience
Mode: code

Projet généré par Vittascience.
Ce fichier contient le code textuel ainsi que le code blocs. Il peut être importé de nouveau
sur l'interface http://vittascience.com/arduino

*/

#include <Wire.h>
#include <paj7620.h>
#include <TM1637.h>

#define PIN_4_DIGIT_DISPLAY_CLK_2  2
#define PIN_4_DIGIT_DISPLAY_DIO_3  3
#define PIN_MOISTURE_SENSOR_A1  A1
#define WATER_THRESHOLD 100
#define ATTINY1_HIGH_ADDR 0x78
#define ATTINY2_LOW_ADDR 0x77

TM1637 tm1637_2(PIN_4_DIGIT_DISPLAY_CLK_2, PIN_4_DIGIT_DISPLAY_DIO_3);

uint8_t water_low_data[8] = {0};
uint8_t water_high_data[12] = {0};
String geste;
String mem_geste_valid;
int8_t tm_digits[4];

String getGestureType() {
  uint8_t data = 0;
  paj7620ReadReg(0x43, 1, &data);
  if (data == GES_RIGHT_FLAG) return "right";
  else if (data == GES_LEFT_FLAG) return "left";
  else if (data == GES_UP_FLAG) return "up";
  else if (data == GES_DOWN_FLAG) return "down";
  else if (data == GES_FORWARD_FLAG) return "forward";
  else if (data == GES_BACKWARD_FLAG) return "backward";
  else if (data == GES_CLOCKWISE_FLAG) return "clockwise";
  else if (data == GES_COUNT_CLOCKWISE_FLAG) return "anticlockwise";
  else if (data == GES_WAVE_FLAG) return "wave";
  else return "gesture not detected";
}

uint16_t getAnalogMean(uint8_t pin, long n) {
  int sum = 0;
  for (int i = 0; i < n; i++) {
    sum += analogRead(pin);
    delay(1);
  }
  return sum / n;
}

void getHigh12SectionValue() {
  memset(water_high_data, 0, sizeof(water_high_data));
  Wire.requestFrom(ATTINY1_HIGH_ADDR, 12);
  while (12 != Wire.available());
  for (int i = 0; i < 12; i++) {
    water_high_data[i] = Wire.read();
  }
  delay(10);
}

void getLow8SectionValue(void) {
  memset(water_low_data, 0, sizeof(water_low_data));
  Wire.requestFrom(ATTINY2_LOW_ADDR, 8);
  while (8 != Wire.available());
  for (int i = 0; i < 8 ; i++) {
    water_low_data[i] = Wire.read(); // receive a byte as character
  }
  delay(10);
}

int getWaterLevel() {
  uint32_t touch_val = 0;
  uint8_t trig_section = 0;
  getLow8SectionValue();
  getHigh12SectionValue();
  for (int i = 0 ; i < 8; i++) {
    if (water_low_data[i] > WATER_THRESHOLD) {
      touch_val |= 1 << i;
    }
  }
  for (int i = 0 ; i < 12; i++) {
    if (water_high_data[i] > WATER_THRESHOLD) {
      touch_val |= (uint32_t)1 << (8 + i);
    }
  }
  while (touch_val & 0x01) {
    trig_section++;
    touch_val >>= 1;
  }
  return trig_section * 5;
}

void setTemperature(int8_t digits[], float t) {
  if (t<100 && t>-10) {
    if (t<-1) {
      digits[0] = '-';
      digits[1] = (int8_t)abs((int)t);
    } else {
      digits[0] = (int8_t)((int)t/10);
      digits[1] = (int8_t)((int)t%10);
    }
  } else {
    digits[0] = '_';
    digits[1] = '_';
  }
  digits[2] = ' ';
  digits[3] = ' ';
}


void setup() {
  paj7620Init();
  pinMode(PIN_4_DIGIT_DISPLAY_CLK_2, OUTPUT);
  pinMode(PIN_4_DIGIT_DISPLAY_DIO_3, OUTPUT);
  tm1637_2.init();
  tm1637_2.set(7); // Maximum brightness
  pinMode(PIN_MOISTURE_SENSOR_A1, INPUT);
  Wire.begin();
  pinMode(6, OUTPUT);
}

void loop() {
  geste = getGestureType();
  if (geste != "gesture not detected") {
    mem_geste_valid = geste;
  }
  if (mem_geste_valid == "right") {
    tm1637_2.displayNum(getAnalogMean(PIN_MOISTURE_SENSOR_A1, 5));
  } else if (mem_geste_valid == "left") {
    setTemperature(tm_digits, getWaterLevel());
    tm1637_2.display(tm_digits);
  } else if (mem_geste_valid == "up") {
    tm1637_2.displayNum(0);
  }
  if (getWaterLevel() > 10 && getAnalogMean(PIN_MOISTURE_SENSOR_A1, 5) < 250) {
    analogWrite(6, (HIGH == 0) ? 0 : 255);
    while (!(getWaterLevel() < 10 || getAnalogMean(PIN_MOISTURE_SENSOR_A1, 5) >= 500)
    ) {}
    analogWrite(6, (LOW == 0) ? 0 : 255);
  } else if (getWaterLevel() > 5 && getAnalogMean(PIN_MOISTURE_SENSOR_A1, 5) < 100) {
    analogWrite(6, (HIGH == 0) ? 0 : 255);
    while (!(getWaterLevel() <= 1 || getAnalogMean(PIN_MOISTURE_SENSOR_A1, 5) >= 500)) {}
    analogWrite(6, (LOW == 0) ? 0 : 255);
  }
}
```