```
// INCLUDES

#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
#include "RTClib.h"
#include <SD.h>

// INSTANCES


File Data; //Logdatei
RTC_DS1307 RTC;

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
// Or, create it with a different I2C address (say for stacking)
// Adafruit_MotorShield AFMS = Adafruit_MotorShield(0x61);

// Select which 'port' M1, M2, M3 or M4. In this case, M1
Adafruit_DCMotor *myMotor = AFMS.getMotor(1);
// You can also make another motor on port M2
Adafruit_DCMotor *myOtherMotor = AFMS.getMotor(2);



// VARIABLES

//float SO2;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
 "Thursday", "Friday", "Saturday"};
long startTime = 0;
//bool error = 0; //log in case of initilization error

const int analogInPin_Flow = A0;  // Analog input pin that the flow sensor is
 attached to
int Flow_sensorValue = 0;

const int analogInPin_SO2 = A2;
int SO2_sensorValue = 0;

int outputValueFlow = 0;
int outputValueSO2 = 0;

const int analogInPin_Voltage = A3;
int halfVoltageValue = 0;
int VoltageValue = 0;

int val1;
int val2;
int val3;
int val4;

// define the used relais
// declare constants LOW or 0 is on and HIGH or 1 is off
```

```
#define HEADER "Date\tTime\tt[ms]\tSO2[ppm]\tSO2out\tFlow\tPumpSpeed\tD1\tD2\tD3
 \tD4\tVoltage"

#define RELAY_ON 0
#define RELAY_OFF 1

#define D1  3  //green wire
#define D2  4  //yellow wire
#define D3  5  //brown wire
#define D4  6  //orange wire

#define rcpin1 2   // remote control SA
#define rcpin2 7   // remote control SB
#define rcpin3 8  // remote control SC



#define SO2out 9 //output for remote control SO2 sensor value transmition


//INITIALIZATIONS


void setup () {


  digitalWrite(D1, RELAY_OFF);
  digitalWrite(D2, RELAY_OFF);
  digitalWrite(D3, RELAY_OFF);
  digitalWrite(D4, RELAY_OFF);


  pinMode(rcpin1, INPUT);
  pinMode(rcpin2, INPUT);
  pinMode(rcpin3, INPUT);
  //pinMode(rcpin4, INPUT);

  pinMode(D1, OUTPUT);
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);

  pinMode(SO2out, OUTPUT);


 analogReference(INTERNAL); //internal reference voltage set to 1,1V

  Serial.begin(9600); //Initilization
 // Wire.begin();

  Serial.println("Start System");

  if(!SD.begin(10)){ //Initilization of SD card; in case no SD card is inserted "0"
    will be logged
   Serial.print("SD-Card missing!");
  }
```

```
  if (! RTC.begin()) {
    Serial.println("Couldn't find RTC");
  }

  if (! RTC.isrunning()) {
    Serial.println("RTC is NOT running!");

    // following line sets the RTC to the date & time this sketch was compiled
    // RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  //AFMS.begin();  // create with the default frequency 1.6KHz
  AFMS.begin(1);  // OR with a different frequency, say 1KHz

  //Set the speed to start, from 0 (off) to 255 (max speed)
  myMotor->setSpeed(0);
  myMotor->run(RELEASE); //release = off

  myOtherMotor->setSpeed(255);
  myOtherMotor->run(FORWARD); //release = off

  Serial.println(HEADER); //print of header line

  Data = SD.open("Data.txt", FILE_WRITE); //creates txt file on sd card for data
   log
  Data.println(HEADER); //prints header line into txt file
  Data.close();

  delay(1000);


}

void loop () {

    startTime = millis(); //save start time

    if(Serial.available() > 19  &&  Serial.read() == 't') setRTCFromSerial(); //Set
      RTC by PC using SetRTC.bat
    while((millis()-startTime)<500); //Set delay to GPS time in case there is a
     difference

    SO2_sensorValue = analogRead(analogInPin_SO2);

    outputValueSO2 = map(SO2_sensorValue, 0, 60, 0, 255);
    analogWrite(SO2out, outputValueSO2);

    Flow_sensorValue = analogRead(analogInPin_Flow);
    //Serial.print(digitalRead(D1));

    halfVoltageValue = analogRead(analogInPin_Voltage);
    VoltageValue = halfVoltageValue*2;

  // Denuder Loops

                // 1. loop for rc SA
```

```
                //Serial.println(pulseIn(rcpin1, LOW));
                 val1 = pulseIn(rcpin1, INPUT);
                  if ((val1 > 13500))
                 {
                 digitalWrite(D1, RELAY_ON);
                  }
                  else {
                 digitalWrite(D1, RELAY_OFF);
                   }


  // 2. loop for rc SB
   //Serial.println(pulseIn(rcpin2, LOW));
  val2 = pulseIn(rcpin2, INPUT);
  if ((val2 > 13500))
      {
  digitalWrite(D2, RELAY_ON);
   }
  else {
    digitalWrite(D2, RELAY_OFF);
    }


                        // 3. loop for rc SC
                          // Serial.println(pulseIn(rcpin3, LOW));
                          val3 = pulseIn(rcpin3, INPUT);
                          if ((val3 > 13500))
                          {
                         digitalWrite(D3, RELAY_ON);
                          }
                             else {
                         digitalWrite(D3, RELAY_OFF);
                            }
//Using tubing D4 as a direct line to the SO2 sensor in case no denuder sampling is
  done

            if ((val3 > 13200)&&
                (val1 < 13500)&&
                (val2 < 13500)&&
                (val3 < 13500))
                 {
             digitalWrite(D4, RELAY_ON);
                  }
                  else {
              digitalWrite(D4, RELAY_OFF);
                        }



if ((val1 > 13500)||
      (val2 > 13500)||
      (val3 > 13200)
      )
      {
```

```
    outputValueFlow = map(Flow_sensorValue, 175, 475, 190, 170);
    myMotor->setSpeed(180);
    myMotor->run(FORWARD);


    }
    else {
    myMotor->run(RELEASE);
    //myMotor->setSpeed(200);
    //myMotor->run(FORWARD);
    }


    printDataToSD(); //print to sd card
    //printDataToSerial(); //send to serial port in order to debug

}


//PROCEDURE FOR SD LOGGING
void printDataToSD(){

  Data = SD.open("Data.txt", FILE_WRITE);
  DateTime now = RTC.now();

    Data.print(now.year(), DEC);
    Data.print('/');
    Data.print(now.month(), DEC);
    Data.print('/');
    Data.print(now.day(), DEC);
    Data.print("  ");
    //Data.print(daysOfTheWeek[now.dayOfTheWeek()]);
    Data.print(now.hour(), DEC);
    Data.print(':');
    Data.print(now.minute(), DEC);
    Data.print(':');
    Data.print(now.second(), DEC);
    Data.print("  ");
    Data.print(startTime);
    Data.print("  ");
    //Data.print("Flow_sensor = ");

    //Data.print("SO2_sensor = ");
    Data.print(SO2_sensorValue);
    Data.print("  ");
    Data.print(outputValueSO2);
    Data.print("  ");

    Data.print(Flow_sensorValue);
    Data.print("  ");
    Data.print(outputValueFlow);
    Data.print("  ");


    Data.print(digitalRead(D1));
    Data.print("  ");
```

```
    Data.print(digitalRead(D2));
    Data.print("  ");
    Data.print(digitalRead(D3));
    Data.print("  ");
    Data.print(digitalRead(D4));
    Data.print("  ");
    Data.println(VoltageValue);

    Data.close();

}

//PROCEDURE FOR SD PINTING
void printDataToSerial(){

  Data = SD.open("Data.txt", FILE_WRITE);
  DateTime now = RTC.now();

    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print("  ");
    //Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.print("  ");
    Serial.print(startTime);
    Serial.print("  ");
   //Serial.print("Flow_sensor = ");


    Serial.print(SO2_sensorValue);
    Serial.print("  ");
    Serial.print(outputValueSO2);
    Serial.print("  ");

    Serial.print(Flow_sensorValue);
    Serial.print("  ");
    Serial.print(outputValueFlow);
    Serial.print("  ");

    Serial.print("  ");
    Serial.print(digitalRead(D1));
    Serial.print("  ");
    Serial.print(digitalRead(D2));
    Serial.print("  ");
    Serial.print(digitalRead(D3));
    Serial.print("  ");
    Serial.print(digitalRead(D4));
    Serial.print("  ");
    Serial.println(VoltageValue);
```

```
}

    //RESET RTC WITH EXTERNAL PC

  void setRTCFromSerial(){
  uint8_t tag = Serial.parseInt();
  uint8_t monat = Serial.parseInt();
  uint16_t jahr = Serial.parseInt();
  uint8_t stunden = Serial.parseInt();
  uint8_t minuten = Serial.parseInt();
  uint8_t sekunden = Serial.parseInt();
  while(Serial.available()) Serial.read();
  RTC.adjust(DateTime(jahr, monat, tag, stunden, minuten, sekunden));
  }
```