**Mini kit CO2 sampling equipment**

**1. List of components**

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Details | Number required | Operating temperature (deg C)\* |
| CO2 sensor | SprintIR®-W, 0-5% with flow through adaptor. Gas Sensing Solutions, Cumbernauld, UK | 1 | 0 to 50 |
| Air pump | D220 BL, TCS micropumps Ltd, Faversham, UK | 1 | -20 to 100 |
| Microcontroller | Arduino Nano | 1 |  |
| Rocker switch | On/off | 2 |  |
| Momentary Pushbutton switch |  | 1 |  |
| Electrical connector | Wago cable connector (222-515; 5 channel) | 3 |  |
| OLED display | 128X64 0.96" OLED I2C | 1 |  |
| Zener Diode | 1N4953B | 1 |  |
| Resistor | 10k-ohm | 1 |  |
| Dupont cables |  |  |  |
| Electrical hook-up wire |  |  |  |
| Battery | 9v PP9 | 1 | -5 to 55 |
| Tubing | Isoversinic, Saint Gobain, France |  | -20 to 200 |
| Coupling | PMCD42-02, Colder Products Company, USA | 1 |  |
| Coupling | PMCD16-02, Colder Products Company, USA | 1 |  |

\*Based on manufacturer’s specifications

**2. Electrical connections schematic**



**3. Tubing connections schematic**



**4. Arduino code**

#include <SoftwareSerial.h>

SoftwareSerial portOne(10, 11); //TX = digital pin 10, RX = digital pin 11

#include <SPI.h>

#include <Wire.h>

/\*

 Universal 8bit Graphics Library, https://github.com/olikraus/u8glib/

 Copyright (c) 2012, olikraus@gmail.com

 All rights reserved.

 Redistribution and use in source and binary forms, with or without modification,

 are permitted provided that the following conditions are met:

 \* Redistributions of source code must retain the above copyright notice, this list

 of conditions and the following disclaimer.

 \* Redistributions in binary form must reproduce the above copyright notice, this

 list of conditions and the following disclaimer in the documentation and/or other

 materials provided with the distribution.

 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND

 CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,

 INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

 DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR

 CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

 SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

 NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

 LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

 CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

 STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

 ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF

 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

#include "U8glib.h"

U8GLIB\_SSD1306\_128X64 u8g(U8G\_I2C\_OPT\_NONE|U8G\_I2C\_OPT\_DEV\_0);

long timer = 0;

char string[6];

char inData[60];

int index;

int zeroPin = 5;

float CO2ppm = 0;

String sCO2ppm = "";

int zerotext = 0;

void setup()

{

 Serial.begin(9600);

 portOne.begin(9600);

 portOne.flush();

 pinMode(zeroPin, INPUT);

 String setFilter = "A 16\r\n"; //set the digital filter on the SPRINT. Number: 1 is no filtering, 32 when factory-supplied.

 portOne.println(setFilter);

 timer = millis();

}

void loop()

{

 if (millis() - timer > 500) {

 zerotext = 0;

 portOne.listen();

 while (portOne.available() > 0) {

 char inByte = portOne.read();

 if (inByte == 'Z') {

 index = 0;

 sCO2ppm = "";

 }

 inData[index] = inByte;

 index++;

 }

 sCO2ppm = " " + sCO2ppm + inData[2] + inData[3] + inData[4] + inData[5] + inData[6];

 CO2ppm = sCO2ppm.toFloat();

 CO2ppm = CO2ppm \* 10;

 Serial.println(CO2ppm, 0);

 if (digitalRead(zeroPin) == HIGH) {

 String sendcalib = "U\r\n";

 portOne.println(sendcalib);

 zerotext = 1;

 }

 u8g.firstPage();

 do {

 draw();

 } while ( u8g.nextPage() );

 timer = millis();

 }

}

void draw(void) {

 u8g.setFont(u8g\_font\_fur11);

 u8g.setPrintPos(0, 12);

 u8g.print("CO2 ppm: ");

 if (zerotext == 1) {

 u8g.setPrintPos(64, 60);

 u8g.print("zeroing");

 }

 u8g.setFont(u8g\_font\_fur20);

 u8g.setPrintPos(20, 44);

 u8g.print(CO2ppm, 0);

}

**5. Sampling procedures**

**Atmospheric CO2 sampling:** Sample air is first pumped through the Mini-kit and out to atmosphere via the soda lime trap to remove traces of the previous sample. Clips (red bars) are then removed from the molecular sieve cartridge and placed on the soda lime cartridge (as shown below) to force the air flow through the molecular sieve. Sampling is performed until 3 to 10 ml of CO2 has been trapped on the molecular sieve (the sieve will trap 100 % of the CO2 until a volume of ca. 10 ml has been collected, and therefore the sampling time is determined from the flow rate (450 ml/min) and CO2 concentration of the air being sampled). An air filter at the gas inlet can be used to prevent particles entering the magnesium perchlorate cartridge.



**Closed chamber sampling:** First, atmospheric CO2 is removed from the chamber to prevent it from contaminating the sample. This is done using the setup below by pumping the chamber air through the soda lime cartridge and back to the chamber in a closed loop until the CO2 sensor indicates no CO2 remains. Clips are retained on the molecular sieve cartridge and no clips are present on the soda lime cartridge.



The chamber is then left to accumulate CO2 (the Mini kit can be disconnected and used on another chamber if required). When sufficient (> 3 ml) CO2 has accumulated, the chamber air is pumped through a molecular sieve cartridge. The closed loop is used again, but this time the clips on the molecular sieve trap are removed and placed on the soda lime cartridge before starting the pump. The amount of CO2 trapped on the molecular sieve can be estimated from the volume of the chamber and the decrease in CO2 concentration during sampling (i.e. the difference represents the amount of CO2 trapped).

