ILLUMINATIONSPACE
ADVANCED SCIENCE RESEARCH CENTER
THE GRADUATE CENTER
CITY UNIVERSITY OF NEW YORK

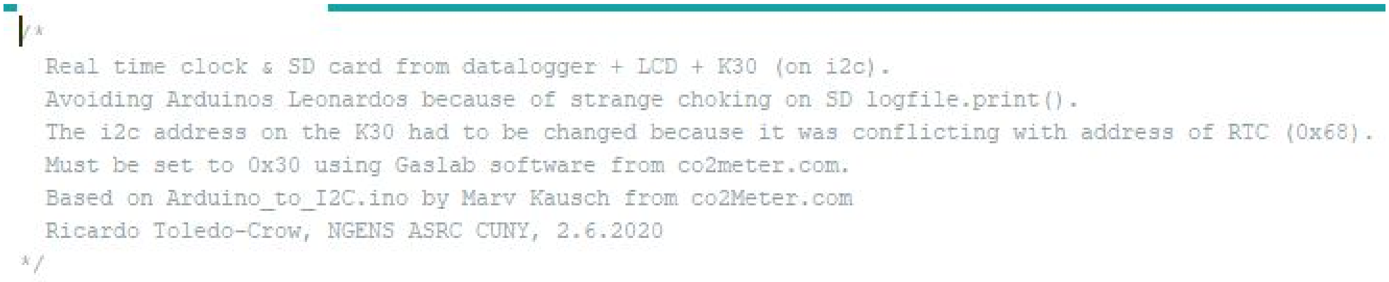
CO2 Sensor Arduino Code Explanation

By Peter Christakos

Introduction

This is the introduction to your code where you describe what it's function is and any notes about compatability and things to look out for. This is like the instruction manual to the code

a. To insert a comment into your code, start with
b. /*   */
   i. You can also disable a piece of code using this or the // notation

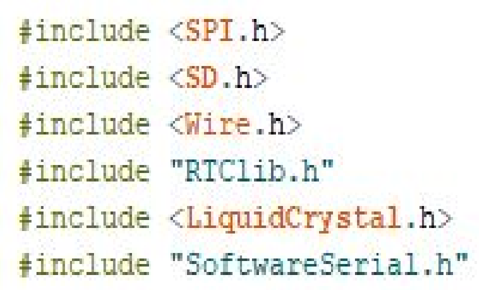


```
/*
  Real time clock & SD card from datalogger + LCD + K30 (on i2c).
  Avoiding Arduinos Leonardos because of strange choking on SD logfile.print().
  The i2c address on the K30 had to be changed because it was conflicting with address of RTC (0x68).
  Must be set to 0x30 using Gaslab software from co2meter.com.
  Based on Arduino_to_I2C.ino by Marv Kausch from co2Meter.com
  Ricardo Toledo-Crow, NGENS ASRC CUNY, 2.6.2020
*/
```

2.

Libraries

This bit of code is calling an external library. Libraries are other files of code that can be referenced which include more variables and commands.

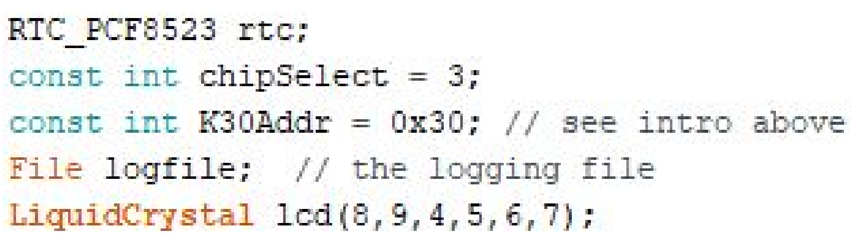


```
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
#include <LiquidCrystal.h>
#include "SoftwareSerial.h"
```

1.

Variables

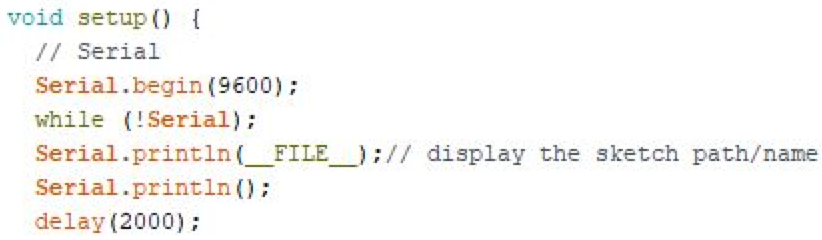
Variables are used to label different values



```
RTC_PCF8523 rtc;
const int chipSelect = 3;
const int K30Addr = 0x30; // see intro above
File logfile;  // the logging file
LiquidCrystal lcd(8,9,4,5,6,7);
```

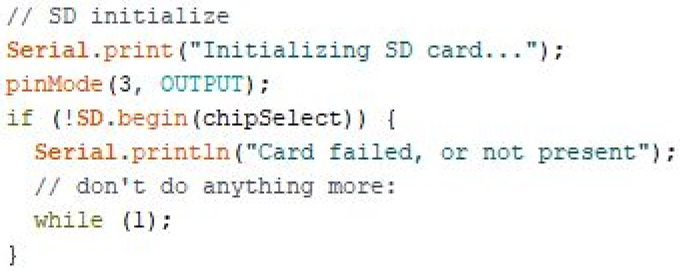
Setup

The set up defines different parameters and conidtions for each component



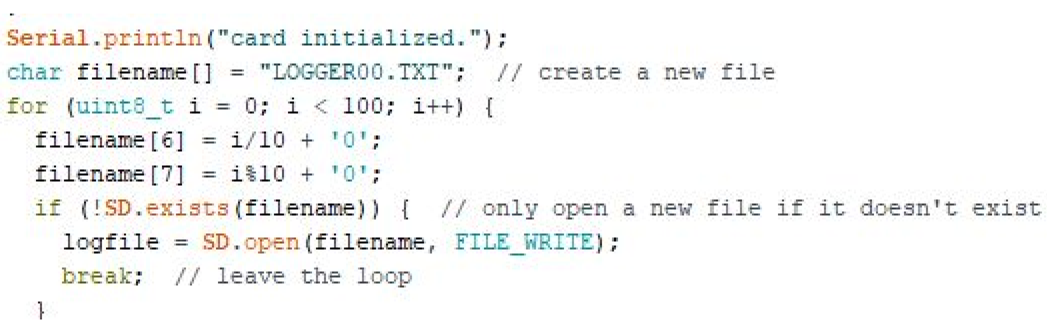
```
void setup() {
  // Serial
  Serial.begin(9600);
  while (!Serial);
  Serial.println(__FILE__);// display the sketch path/name
  Serial.println();
  delay(2000);
```

//Serial Communication: This tells the arduino to exchange information at 9600 bits per second, With a delay of 2000.



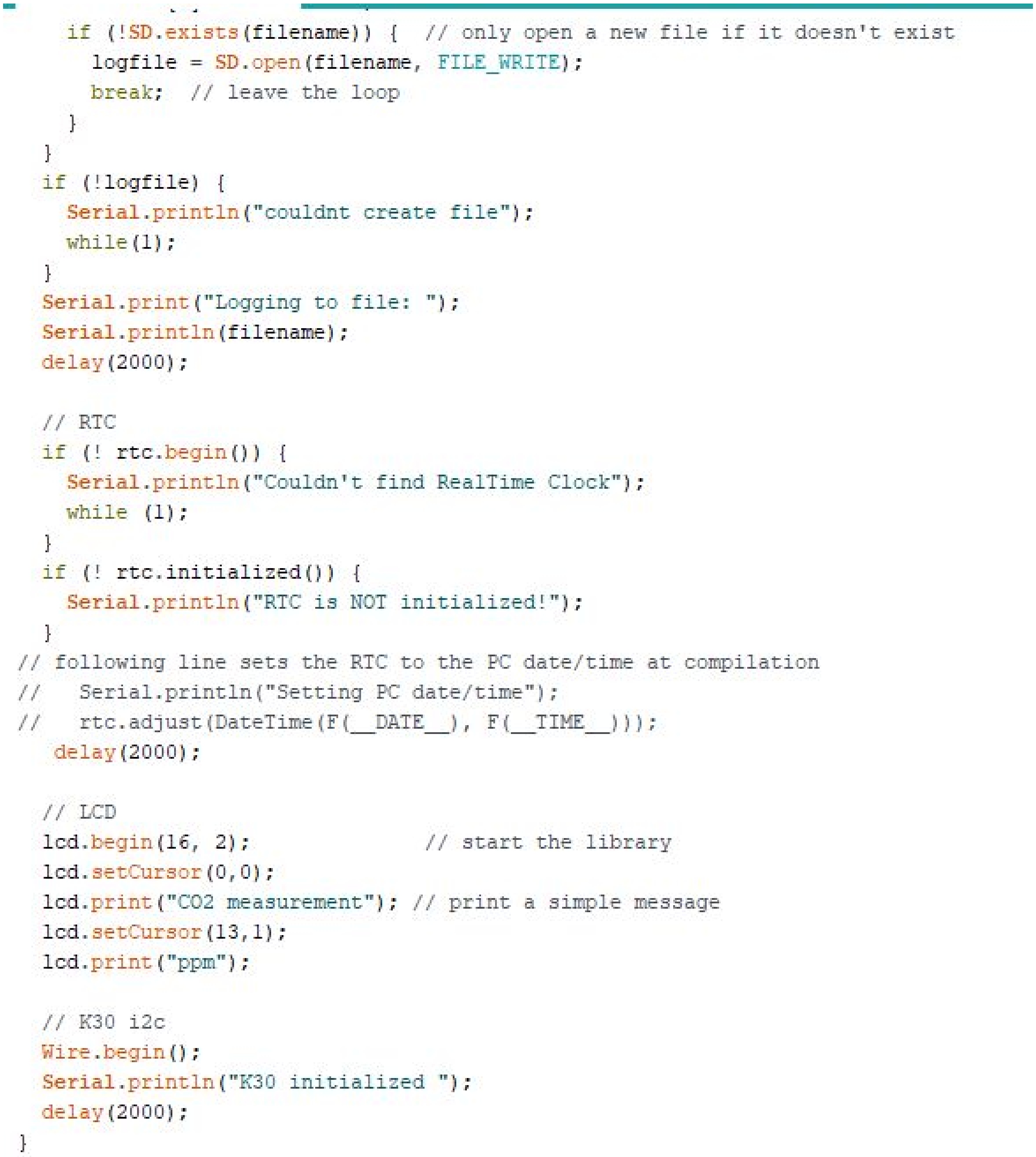
```
  // SD initialize
  Serial.print("Initializing SD card...");
  pinMode(3, OUTPUT);
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    while (1);
  }
```

//SD Card Storage: Starts the use of the SD card to store the data being given. But if there is no SD card in the arduino it will not do anything at all.



```
  Serial.println("card initialized.");
  char filename[] = "LOGGER00.TXT";  // create a new file
  for (uint8_t i = 0; i < 100; i++) {
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (!SD.exists(filename)) {  // only open a new file if it doesn't exist
      logfile = SD.open(filename, FILE_WRITE);
      break;  // leave the loop
    }
```

//Creates a file to display the information that is being given off by the arduino.
Does not open another display if one already exist.



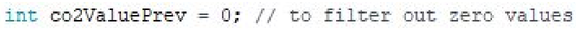
```
    if (!SD.exists(filename)) {  // only open a new file if it doesn't exist
      logfile = SD.open(filename, FILE_WRITE);
      break;  // leave the loop
    }
  }
  if (!logfile) {
    Serial.println("couldnt create file");
    while(1);
  }
  Serial.print("Logging to file: ");
  Serial.println(filename);
  delay(2000);

  // RTC
  if (! rtc.begin()) {
    Serial.println("Couldn't find RealTime Clock");
    while (1);
  }
  if (! rtc.initialized()) {
    Serial.println("RTC is NOT initialized!");
  }
// following line sets the RTC to the PC date/time at compilation
//   Serial.println("Setting PC date/time");
//   rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
   delay(2000);

  // LCD
  lcd.begin(16, 2);              // start the library
  lcd.setCursor(0,0);
  lcd.print("CO2 measurement"); // print a simple message
  lcd.setCursor(13,1);
  lcd.print("ppm");

  // K30 i2c
  Wire.begin();
  Serial.println("K30 initialized ");
  delay(2000);
}
```

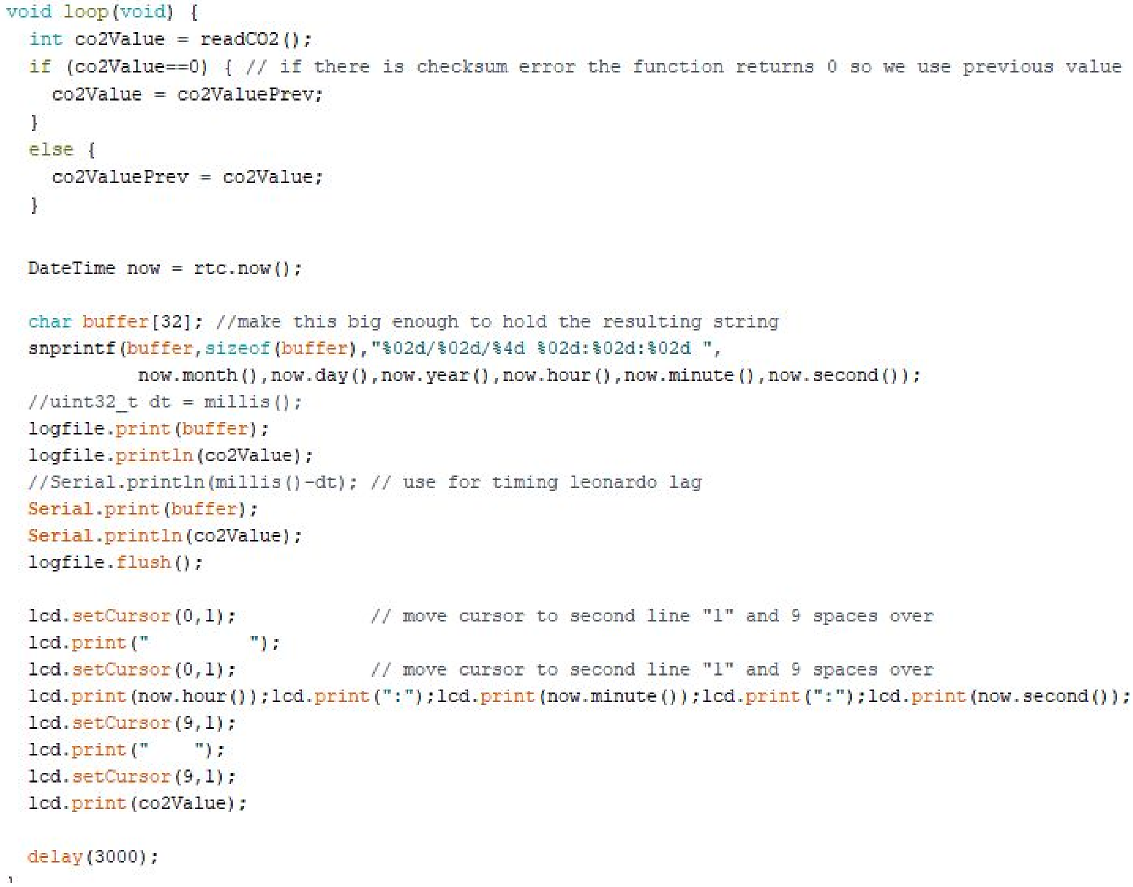
//Prints the information displayed on the logger shield and given off the censor onto the arduino program. The date, time and the measurement of C02.



```
int co2ValuePrev = 0; // to filter out zero values
```

The Loop

The loop is a set of commands which run over and over again. For our code the arduino is constantly reading from the CO2 sensor and writing to both the logger and the LCD Shield.



```
void loop(void) {
  int co2Value = readCO2();
  if (co2Value==0) { // if there is checksum error the function returns 0 so we use previous value
    co2Value = co2ValuePrev;
  }
  else {
    co2ValuePrev = co2Value;
  }


  DateTime now = rtc.now();

  char buffer[32]; //make this big enough to hold the resulting string
  snprintf(buffer,sizeof(buffer),"%02d/%02d/%4d %02d:%02d:%02d ",
            now.month(),now.day(),now.year(),now.hour(),now.minute(),now.second());
  //uint32_t dt = millis();
  logfile.print(buffer);
  logfile.println(co2Value);
  //Serial.println(millis()-dt); // use for timing leonardo lag
  Serial.print(buffer);
  Serial.println(co2Value);
  logfile.flush();

  lcd.setCursor(0,1);            // move cursor to second line "1" and 9 spaces over
  lcd.print("         ");
  lcd.setCursor(0,1);            // move cursor to second line "1" and 9 spaces over
  lcd.print(now.hour());lcd.print(":");lcd.print(now.minute());lcd.print(":");lcd.print(now.second());
  lcd.setCursor(9,1);
  lcd.print("    ");
  lcd.setCursor(9,1);
  lcd.print(co2Value);

  delay(3000);
}
```