

Wireless Air Quality System

Ayşe Serra Şimşek
aserrasimsek@gmail.com



AMAÇ:

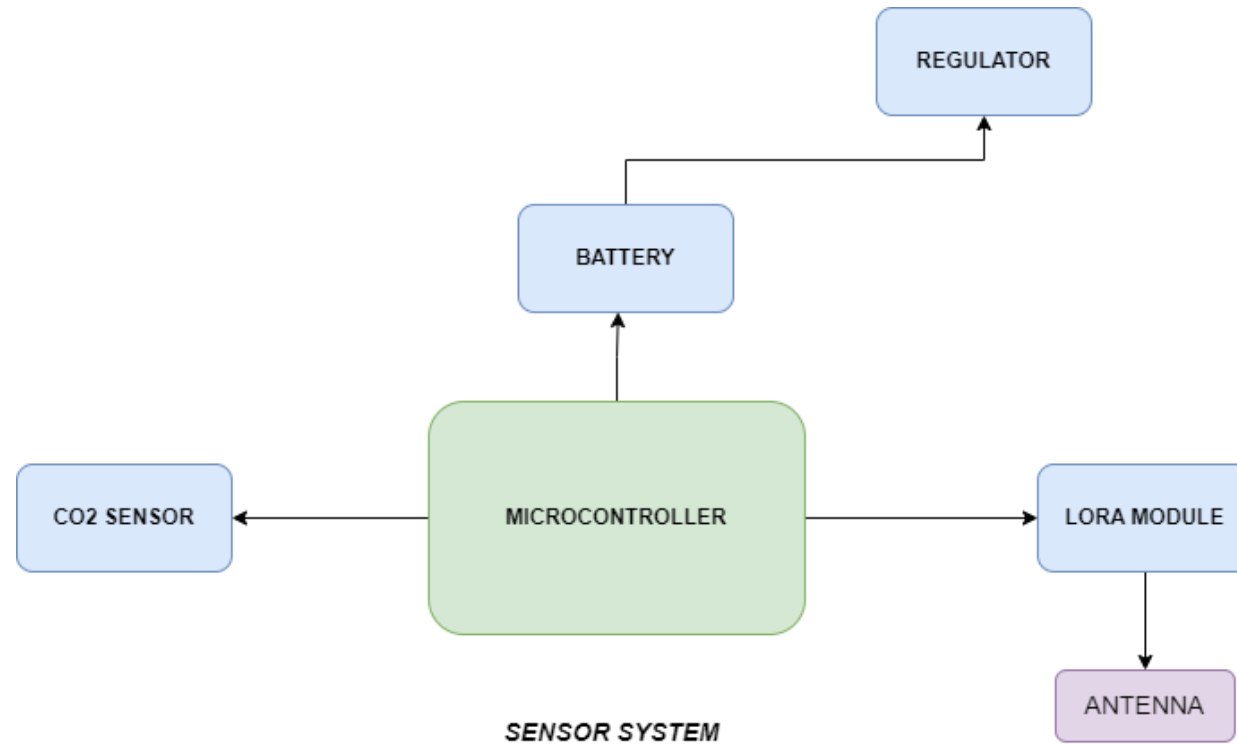
- Ofis vb. kapalı ortamlarda solunan hava bol miktarda karbondioksit içerir.
- %0,15 CO2'de yorgun hissedilmeye başlanır. %0,3 CO2'de alınan her 18 nefes yeniden kullanılan havadır.
- Baş ağrısı, kafa karışıklığı, yorgunluk ve nefes darlığı çok yüksek CO2 seviyesinin uyarı semptomlarıdır.
- Yani havadaki CO2 partiküllerinin oranı ne kadar yüksekse, birkaç kez solunmuş olma olasılığı da o kadar yüksektir.
- Bu solunan havada çeşitli virüsler taşınır. (Yakın dönemde yaşadığımız Covid salgını)
- Bununla birlikte ortamdaki CO2 artımı beklenen şekilde vücudun ihtiyacı olan oksijeni sağlayamaması anlamına gelir. Ofis vb. çalışma yerlerinde uzun süreli düşük oksijene maruz kalma, çalışanların veriminin düşüklüğünü de beraberinde getirecektir.

ÇÖZÜM:

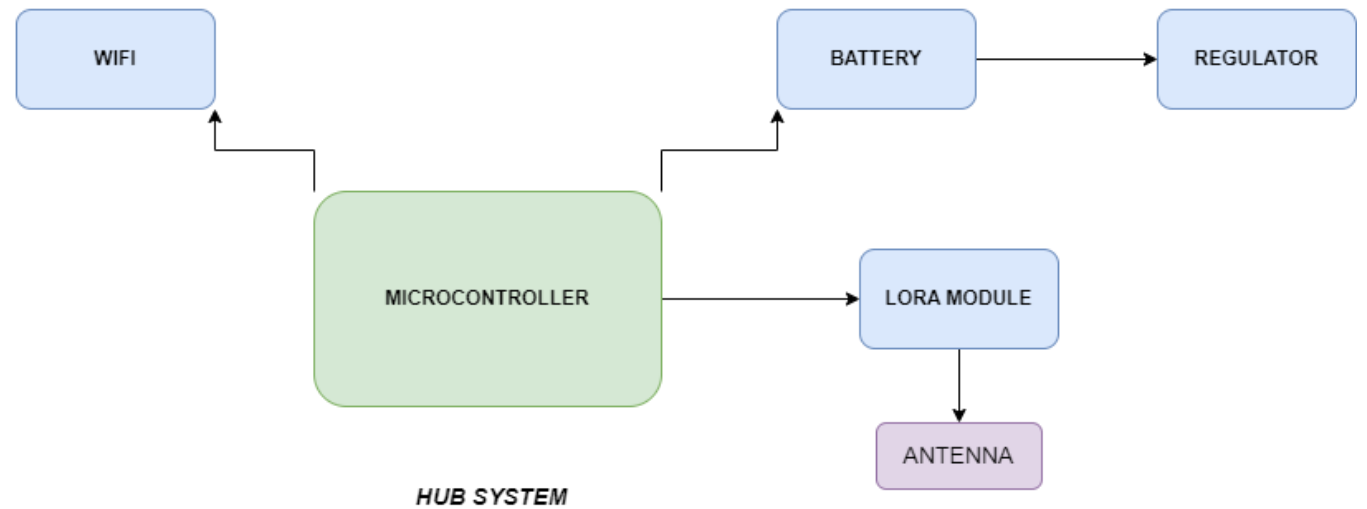
Öncelikle bir ofis odasını ele alırsak, bir veya birden fazla CO2 sensörünü odanın çeşitli yerlerine yerleştirerek havadaki karbondioksit oranlarını belirli periyotta ölçeriz. Ölçüm sonucu elde edilen veriler bir toplayıcı noktasında toplanır, CO2 miktarı kritik değerin üzerindeyse kullanıcıya bildirim gider. Aynı veri, havalandırma sisteminin (fan) aktif olması için de kullanılır.

Sistemi bina genelinde kurgularsak, sensörlü sistemin sayısını arttırmak yeterli olacaktır.

Sensör Sistemi:



HUB Sistemi:



Gerekli Komponentler:



- Nuvoton NuMaker
- LoRa modülü
- Sensirion CO2 sensörü

Sensörde I2C Üzerinden Veri Alınması

```
#include <stdio.h>
#include "NuMicro.h"

uint8_t gelen_veri[256];
uint8_t indis;
uint16_t mesaj;
void SYS_Init(void)
{

/* Unlock protected registers */
SYS_UnlockReg();

/* If the macros do not exist in your project, please refer to the related
clk.h in Header folder of the tool package */
/* Enable clock source */
CLK_EnableXtalRC(CLK_PWRCTL_HIRCEN_Msk);

/* Waiting for clock source ready */
CLK_WaitClockReady(CLK_STATUS_HIRCSTB_Msk);

/* Set HCLK clock */
CLK_SetHCLK(CLK_CLKSEL0_HCLKSEL_HIRC,
CLK_CLKDIV0_HCLK(1));

/* Set PCLK-related clock */
CLK->PCLKDIV = (CLK_PCLKDIV_APB0DIV_DIV1 |
CLK_PCLKDIV_APB1DIV_DIV1);

/* Enable IP clock */
CLK_EnableModuleClock(I2C0_MODULE);
CLK_EnableModuleClock(TMR0_MODULE);
CLK_EnableModuleClock(UART0_MODULE);
CLK_EnableSysTick(CLK_CLKSEL0_STCLKSEL_HIRC_DIV2, 0);

/* Set IP clock */
CLK_SetModuleClock(TMR0_MODULE,
CLK_CLKSEL1_TMR0SEL_HIRC, MODULE_NoMsk);
CLK_SetModuleClock(UART0_MODULE,
CLK_CLKSEL1_UART0SEL_PCLK0, CLK_CLKDIV0_UART0(1));
```

```
/* Update System Core Clock */
/* User can use SystemCoreClockUpdate() to calculate
SystemCoreClock. */
SystemCoreClockUpdate();

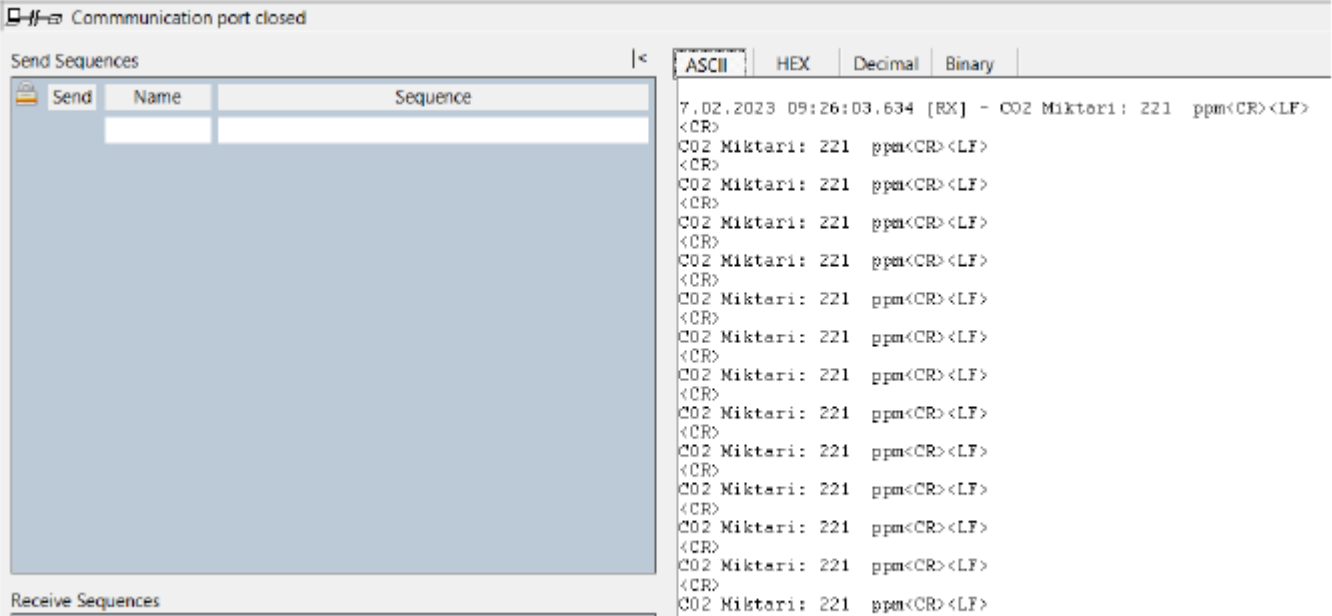
SYS->GPA_MFPH = 0x00000000;
SYS->GPA_MFPL = 0x00000000;
SYS->GPB_MFPH = SYS_GPB_MFPH_PB13MFP_UART0_TXD |
SYS_GPB_MFPH_PB12MFP_UART0_RXD;
SYS->GPB_MFPL = SYS_GPB_MFPL_PB5MFP_I2C0_SCL |
SYS_GPB_MFPL_PB4MFP_I2C0_SDA;
SYS->GPC_MFPL = 0x00000000;
SYS->GPF_MFPH = 0x00000000;
SYS->GPF_MFPL = SYS_GPF_MFPL_PF1MFP_ICE_CLK |
SYS_GPF_MFPL_PF0MFP_ICE_DAT;

/* Lock protected registers */
SYS_LockReg();
}
int main()
{
SYS_Init();

UART_Open(UART0, 115200);
I2C_Open(I2C0, 100000);

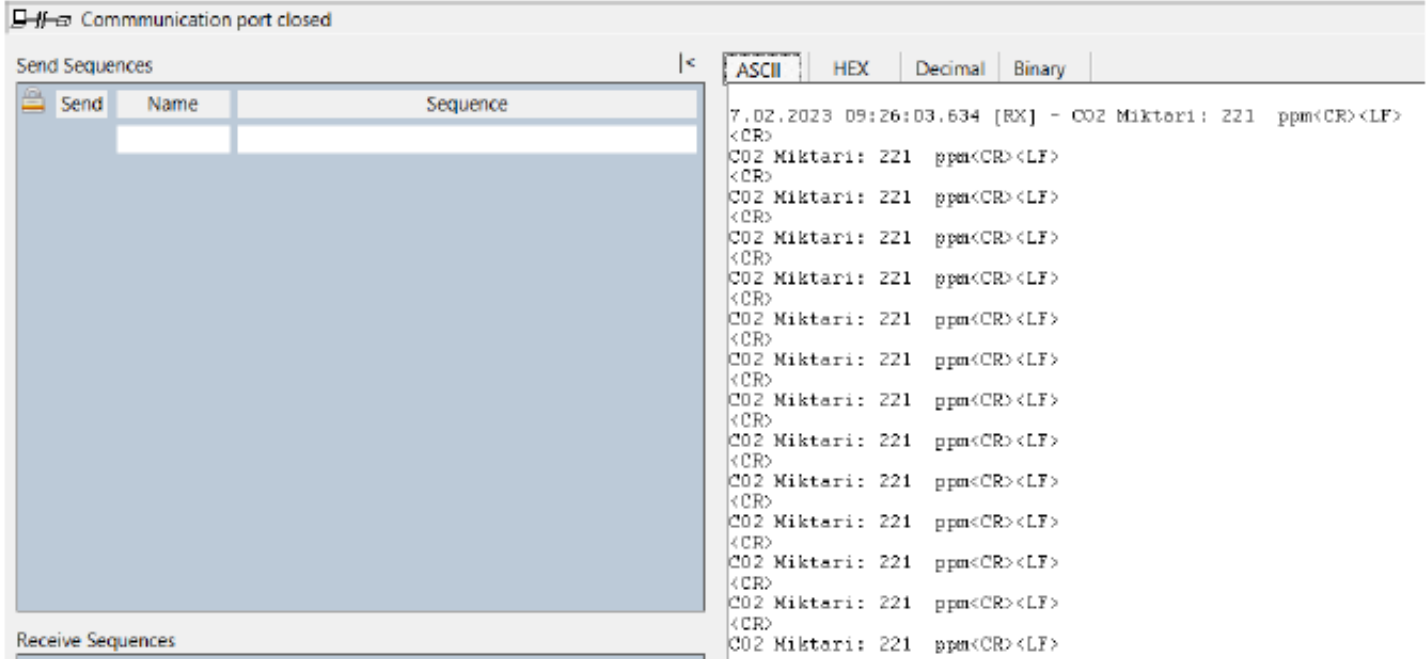
for(indis=0; indis<128; indis++){
I2C_ReadMultiBytesOneReg(I2C0, 0x61, 0xC3, gelen_veri, 1);
CLK_SysTickDelay(1000000);
mesaj=gelen_veri[0];
printf("CO2 Miktarı: %d ppm\n\r", mesaj);
}
NVIC_EnableIRQ(I2C0_IRQn);
while(1); }
```

Sensörde I2C Üzerinden Veri Alınması



Sensör Verisi

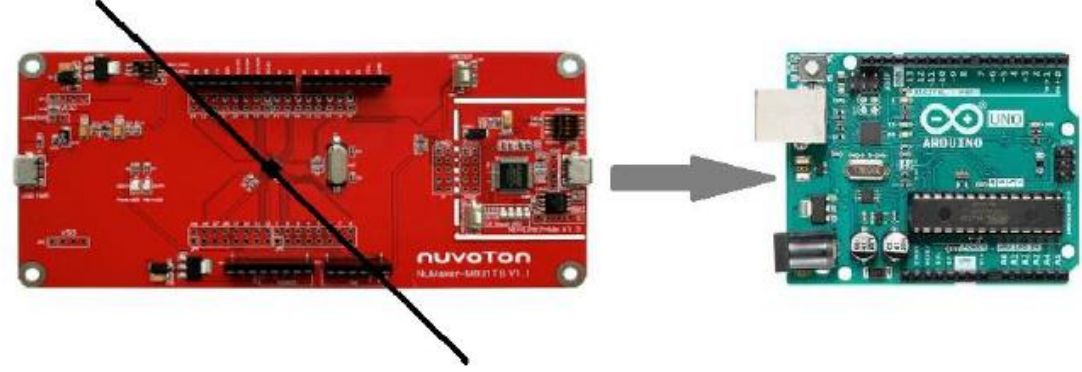
Sensörde I2C Üzerinden Veri Alınması



Sensör Verisi

Projenin Revize Edilmesi:

Revize:



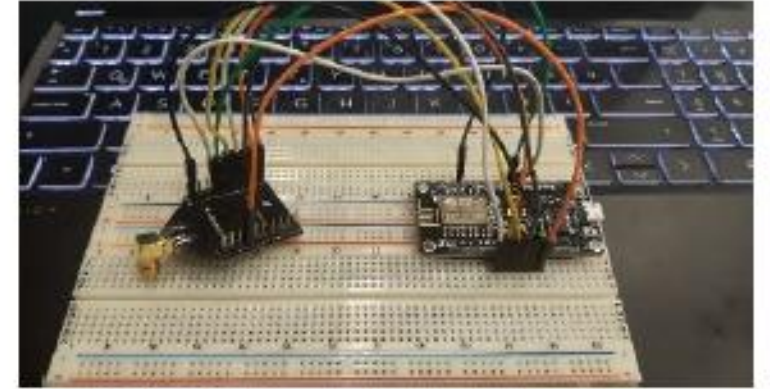
Nuvoton işlemciyi kullanarak SCD30 sensörü ile sağlıklı veri alınamamıştır. Staj süresinin kısa oluşundan dolayı Nuvoton yerine Arduino'ya geçilmiştir.



Komponentler

Lora ve ESP8266 Denemeleri:

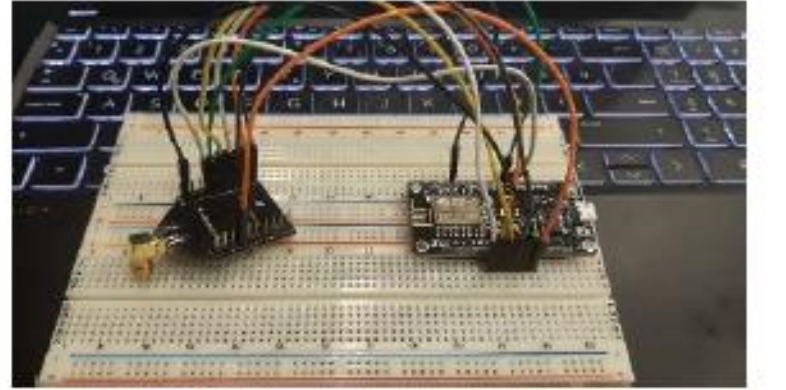
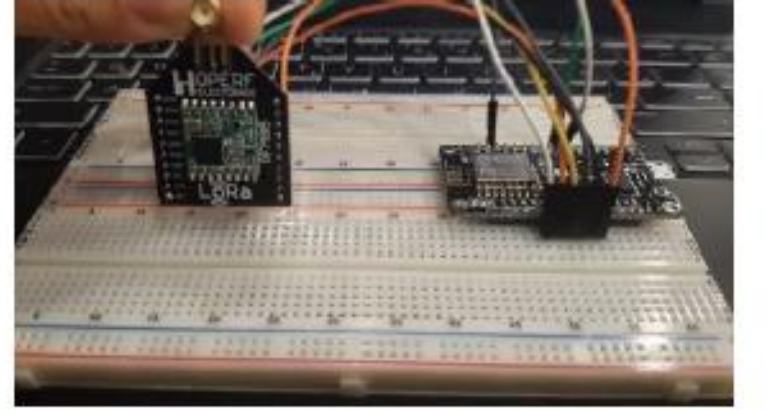
```
17:43:06.677 -> ***CO2 Olcum Verisi***  
17:43:06.677 ->  
17:43:06.722 -> Veri aliniyor..  
17:43:06.722 ->  
17:43:06.722 -> Olcum araligi: 2 saniye  
17:43:06.722 ->  
17:43:08.857 ->  
17:43:08.857 -> Sicaklik: 25.10 degrees C  
17:43:08.857 ->  
17:43:08.857 -> Bagil Nem: 35.53 %  
17:43:08.857 ->  
17:43:08.857 -> CO2 Yogunlugu: 42.348 ppm  
17:43:08.857 ->  
17:43:10.943 ->  
17:43:10.943 -> sicaklik: 25.13 degrees C  
17:43:10.943 ->  
17:43:10.943 -> Bagil Nem: 35.45 %  
17:43:10.943 ->  
17:43:10.943 -> CO2 Yogunlugu: 1724.395 ppm  
17:43:10.943 ->  
17:43:12.944 ->  
17:43:12.944 -> Sicaklik: 25.14 degrees C  
17:43:12.944 ->  
17:43:12.944 -> Bagil Nem: 35.39 %  
17:43:12.944 ->  
17:43:12.944 -> CO2 Yogunlugu: 1750.275 ppm
```



Haberleşme Değişikliği:

LoRa modüllerinde oluşan donanımsal aksaklık sonucu Bluetooth haberleşmesine geçilmiştir. Sensörden alınan veriler Arduino ile uyumlu olan **HM-10 Bluetooth** modülü ile telefona gönderilmiştir.

```
17:43:06.677 -> ***CO2 Ölçüm Verisi***
17:43:06.677 ->
17:43:06.722 -> Veri alınıyor..
17:43:06.722 ->
17:43:06.722 -> Ölçüm aralığı: 2 saniye
17:43:06.722 ->
17:43:08.857 ->
17:43:08.857 -> Sıcaklık: 25.10 degrees C
17:43:08.857 ->
17:43:08.857 -> Bağıl Nem: 35.53 %
17:43:08.857 ->
17:43:08.857 -> CO2 Yoğunluğu: 42.348 ppm
17:43:08.857 ->
17:43:10.943 ->
17:43:10.943 -> Sıcaklık: 25.13 degrees C
17:43:10.943 ->
17:43:10.943 -> Bağıl Nem: 35.45 %
17:43:10.943 ->
17:43:10.943 -> CO2 Yoğunluğu: 1724.395 ppm
17:43:10.943 ->
17:43:12.944 ->
17:43:12.944 -> Sıcaklık: 25.14 degrees C
17:43:12.944 ->
17:43:12.944 -> Bağıl Nem: 35.39 %
17:43:12.944 ->
17:43:12.944 -> CO2 Yoğunluğu: 1750.275 ppm
```



Haberleşme Değişikliği:

```
#include <SoftwareSerial.h>
#include <Adafruit_SCD30.h>
Adafruit_SCD30 sensor;
SoftwareSerial Blu(2, 3); // HM-10 Bağlantı pinleri rx=2, tx=3

void setup() {

  Serial.begin(9600);
  Blu.begin(9600);
  while (!Serial) delay(10);
  Serial.println("***CO2 Olcum Verisi***\n");

  if (!sensor.begin()) {
    Serial.println("Failed to find SCD30 chip");
    while (1){ delay(10);}
  }

  Serial.println("Veri aliniyor..\n");

  Serial.print("Olcum araligi: ");
  Serial.print(sensor.getMeasurementInterval()); // 2 saniye
  Serial.println(" saniye\n"); }

void loop() {

  if (sensor.dataReady()){

    if (!sensor.read()){
      Serial.println("Hata");
      return; }

    Serial.print("\nCO2 Yogunlugu: ");
    Serial.print(sensor.CO2);
    Serial.println(" ppm");
    Serial.println("");
    Blu.print("\nCO2 Yogunlugu: ");
    Blu.print(sensor.CO2);
    Blu.println(" ppm");

    if(2000 > sensor.CO2 && sensor.CO2 > 1300 ){

      Serial.println("Hava kalitesi dusuk seviyede!");
      Blu.println("---->Hava kalitesi dusuk seviyede! "); }

    else if(5000 > sensor.CO2 && sensor.CO2 > 2000){

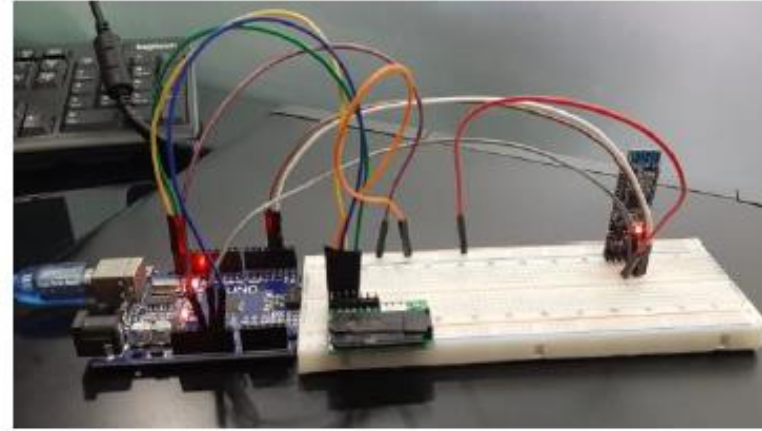
      Serial.println("CO2 miktarı yuksek seviyede!");
      Blu.println("---->CO2 miktarı yuksek seviyede!"); }

    else if(sensor.CO2 > 5000){

      Serial.println("CO2 miktarı tehlikeli seviyede!");
      Blu.println("---->CO2 miktarı tehlikeli seviyede!"); }

    delay(100);
  }
}
```


Sonuç:



```
Arduino Bluetooth C...
HMSOft
F8:30:02:2A:86:35
Connected
Serial present
CO2 Yogunlugu: 1097.42 ppm
CO2 Yogunlugu: 1096.61 ppm
CO2 Yogunlugu: 1096.75 ppm
CO2 Yogunlugu: 1097.45 ppm
CO2 Yogunlugu: 0.00 ppm
CO2 Yogunlugu: 1028.36 ppm
CO2 Yogunlugu: 1068.06 ppm
CO2 Yogunlugu: 1075.23 ppm
CO2 Yogunlugu: 2559.50 ppm
-->CO2 miktarı yüksek seviyede!
CO2 Yogunlugu: 3095.04 ppm
-->CO2 miktarı yüksek seviyede!
```

Devre ve Sensör Verisi



TEŞEKKÜRLER.

SORULAR ?