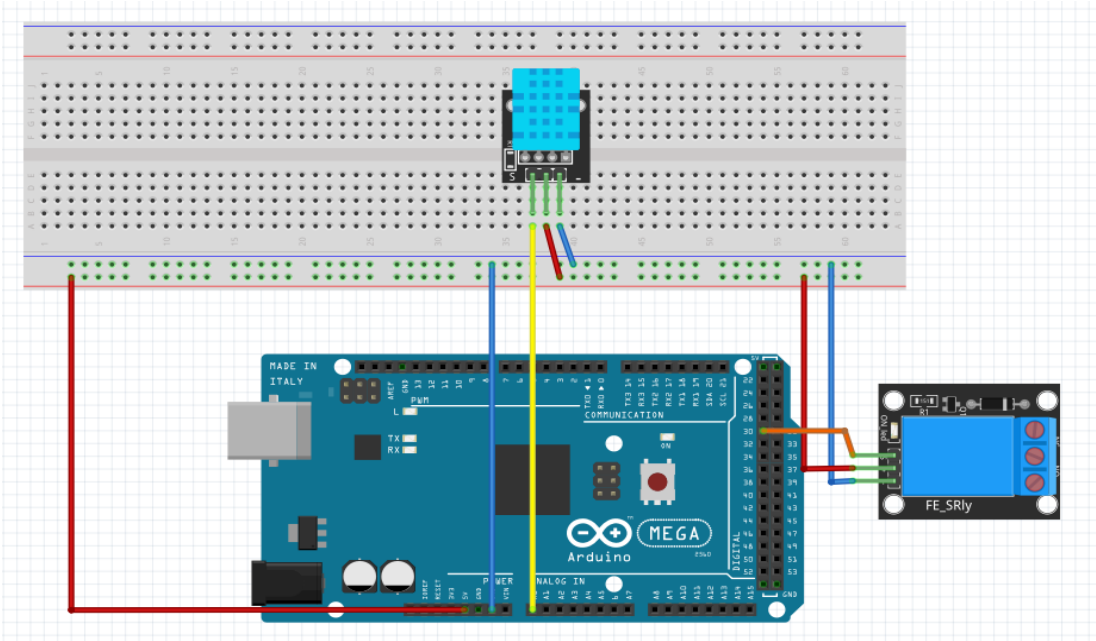


포트폴리오

프로젝트명	Visual Studio C#을 활용하여 온,습도 센서(DHT11)값, RELAY ON/OFF GUI 구현.
프로젝트 기간	2023.07.19~ 2023.07.19
상세내용	<p>1) 소 개: Visual Studio C#을 활용하여 온,습도 센서(DHT11)값, RELAY ON/OFF GUI 구현.</p> <p>2) 주요 기능: 1. Arduino 의 Mega 2560 보드와 BreadBoard 를 연결. 2. 온,습도 센서(DHT11)와 RELAY 를 BreadBoard 에 연결하고 회로를 구성. 3. Arduino IDE 를 이용하여 코드를 작성 후 업로드.</p> <p>3) 개발 환경 및 개발 언어: Arduino IDE, C 언어</p>
구성도	
상세 설명	<p>1. BreadBoard 에 온,습도 센서(DHT11)과 RELAY 를 연결.</p> <p>2. 온,습도 센서(DHT11)의 GND 는 GND 에 연결, '+' PIN 은 5V, Serial PIN 은 ANALOG PIN A0 에 연결.</p> <p>3. 온,습도 센서(DHT11)을 사용하기 위해서 구글에서 DHT11 arduino library 라고 검색하여 Arduino 홈페이지에서 DHT sensor library 를 다운받아 코딩할 파일에 DHT.cpp, DHT.h 파일을 넣어주어야 동작이 가능하다.</p> <p>4. GUI 구현을 위하여 Visual Studio 설치하여 C#으로 코딩.</p>

사용 코드
(Visual
Studio)

```
using System.IO.Ports; ///import와 똑같음

namespace Arduino_Communication
{
    참조 3개
    partial class Form1 : Form
    {
        private SerialPort _serialPort;
        String serial_data;
        참조 1개
        public Form1()
        {
            InitializeComponent();
            String[] portNames = SerialPort.GetPortNames();
            foreach (String portName in portNames)
            {
                this.textBox_PortNumber.Text = portName;
            }
            this._serialPort = new SerialPort();
        }

        참조 1개
        private void _serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
        {
            //throw new NotImplementedException();
            this.serial_data = this._serialPort.ReadLine();
            string[] parsingData = this.serial_data.Split(',');
            //온도값 빼오기
            string str_temperature = parsingData[0];
            //습도값 빼오기
            string str_humidity = parsingData[1];

            float float_temperature;
            float float_humidity;
            float.TryParse(str_temperature, out float_temperature);
            float.TryParse(str_humidity, out float_humidity);

            this.label_temperature.Invoke(new Action(() =>
            {
                label_temperature.Text = str_temperature;
            }));
        }
    }
}
```

```
        this.label_temperature.Invoke(new Action(() =>
        {
            label_temperature.Text = str_temperature;
        }));
        this.label_humidity.Invoke(new Action(() =>
        {
            label_humidity.Text = str_humidity;
        }));
        this.progressBar_Temperature.Invoke(new Action(() =>
        {
            progressBar_Temperature.Value = (int)float_temperature;
        }));
        this.progressBar_Humidity.Invoke(new Action(() =>
        {
            progressBar_Humidity.Value = (int)float_humidity;
        }));
    }

    참조 1개
    private void button_PortOpen_Click(object sender, EventArgs e)
    {
        if (this._serialPort.IsOpen)
        {
            this.textBox_PortNumber.Text = "이미 포트가 열려 있습니다.";
        }
        else
        {
            String temporary = this.textBox_PortNumber.Text;
            if (temporary.Equals(String.Empty))
            {
                Console.WriteLine("포트넘버가 입력이 안되었습니다.");
                return;
            }
            else
            {
                ///아두이노 시리얼 통신 기본 설정
                this._serialPort.PortName = this.textBox_PortNumber.Text; ///COM3
                this._serialPort.BaudRate = 115200;
                this._serialPort.DataBits = 8;
                this._serialPort.StopBits = StopBits.One;
                this._serialPort.Parity = Parity.None;
                this._serialPort.DataReceived += _serialPort1_DataReceived;
            }
        }
    }
}
```

참조 1개

```
private void button_Relay_On_Click(object sender, EventArgs e)
{
    this._serialPort.WriteLine("SW_ON");
}
```

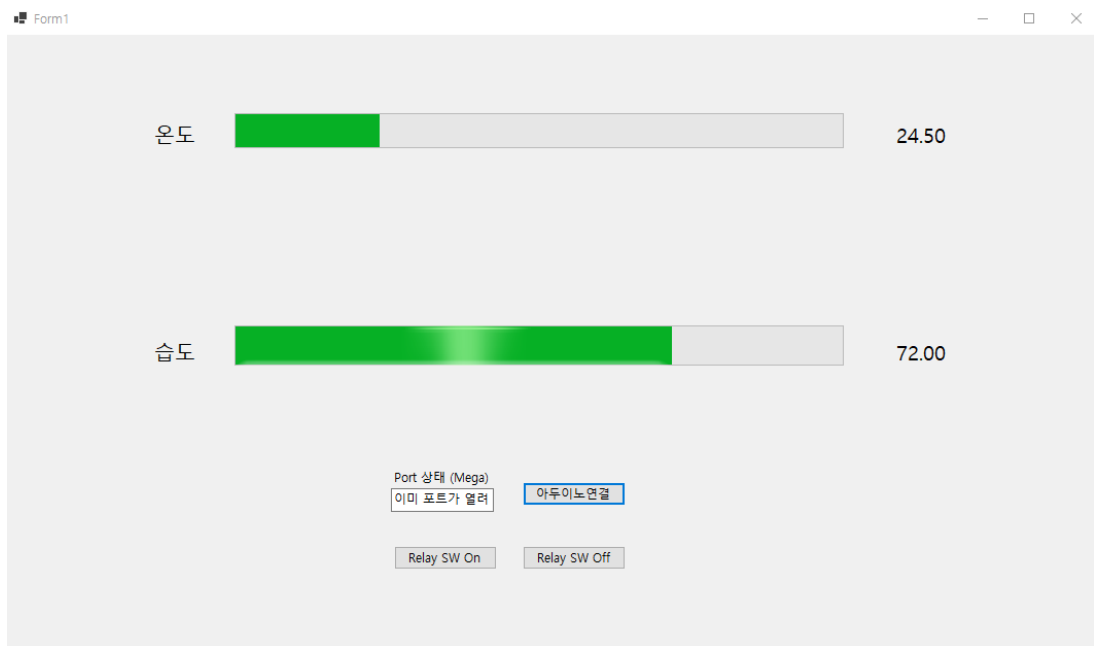
참조 1개

```
private void button_Relay_Off_Click(object sender, EventArgs e)
{
    this._serialPort.WriteLine("SW_OFF");
}
```

사용 코드
(Arduino)

```
#include "DHT.h"
class DHT dht(A0, 11);
const uint8_t RELAY_SW {30U};
void setup() {
    dht.begin();
    Serial.begin(115200UL); //COM 통신용
    pinMode(RELAY_SW, OUTPUT); //RELAY
}
void loop() {
    float temperature = 0.0;
    float percentHumidity = 0.0F;
    if(dht.read()) {
        temperature = dht.readTemperature();
        percentHumidity = dht.readHumidity();}
    if(Serial.available()) {
        String str_command = Serial.readString(); // SW_ON
        str_command.trim();
        str_command.toUpperCase();
        if(str_command == "SW_ON")
        { digitalWrite(RELAY_SW, HIGH);
          delay(500UL);
        } else if(str_command == "SW_OFF") {
          digitalWrite(RELAY_SW, LOW);
          delay(500UL);
        } else {
        }
    }
    Serial.print(temperature);
    Serial.print(",");
    Serial.println(percentHumidity);
    delay(700UL);
}
```

결과



Visual Studio 를 통해서 GUI 를 구현하여 아두이노와 연결되면 온도, 습도를 아두이노에서 받아오면 GUI 화면에 온도, 습도 값이 출력되며, RELAY 도 연동되어 있어 RELAY 스위치 ON/OFF 도 가능 하게 구현 하였습니다.

이 방법을 통해서 원격지에서 온도, 습도를 모니터링 하여 RELAY 를 통해서 장비들을 ON/OFF 가 가능하도록 구현이 가능할 것 입니다.