

2022년 IoT기반 스마트 솔루션 개발자 양성과정



Embedded Application

7-Gyro Sensor

담당 교수 : 윤 종 이

010-9577-1696

ojo1696@naver.com

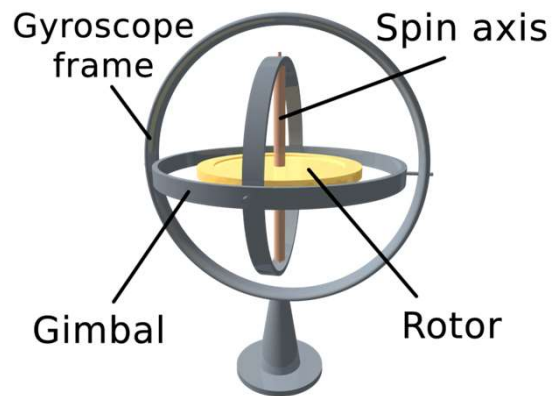
<https://cafe.naver.com/yoons2022>



충북대학교 공동훈련센터

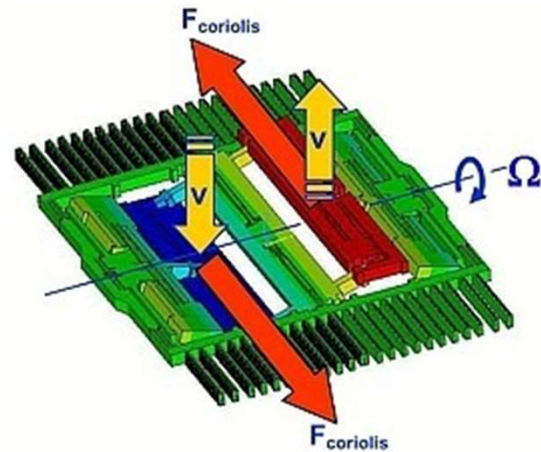
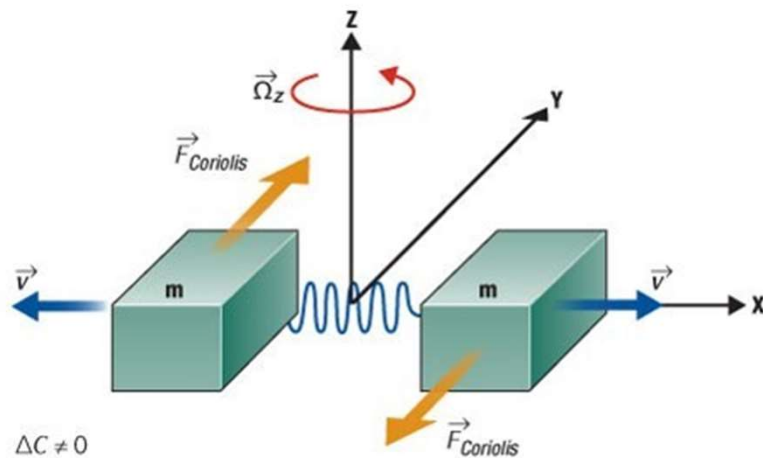
Gyro Sensor

- 자이로
 - 공간에서 자유로이 회전하도록 장치된 일종의 팽이
 - 지면에 수직인 조건에 상관 없이 모든 방면에 대해 각도의 변화를 측정
 - 단위 시간당 각도 변화량



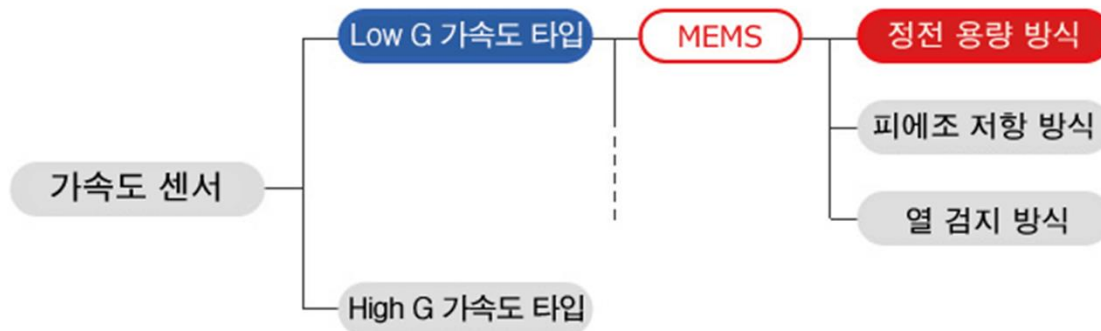
MEMS 기반의 자일로센서

- 코리올리의 힘을 측정

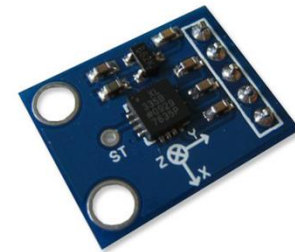
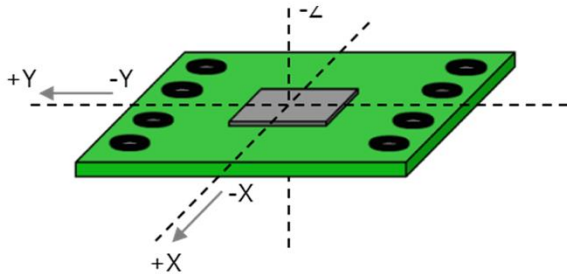


가속도 센서[Accelerometer]

- 단위 시간 당 속도



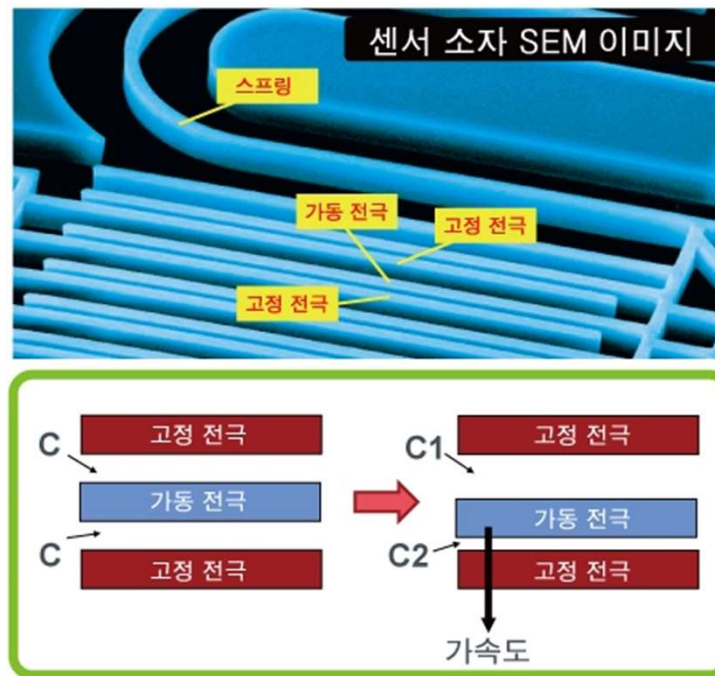
[측정 범위에 따른 가속도 센서 분류]



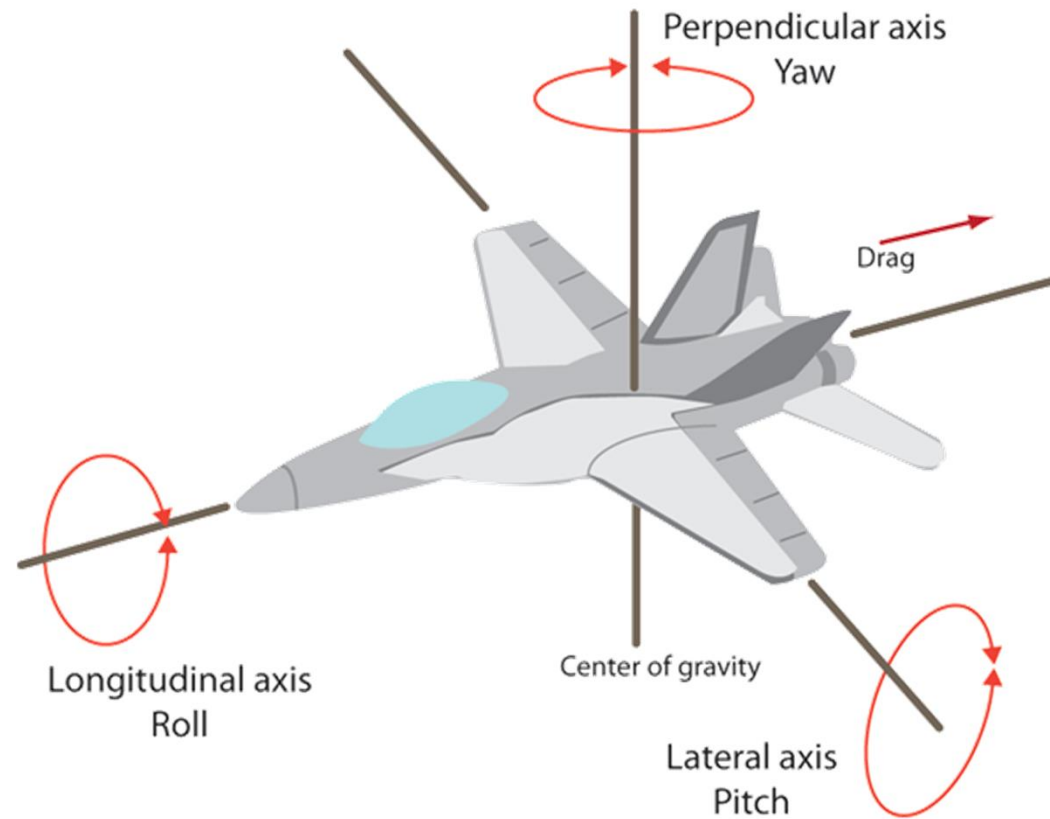
충북대학교 공동훈련센터

MEMS 기반의 가속도 센서

- MEMS 기술을 사용한 정전 용량 방식의 가속도 센서
- 센서 소자는 Si로 만들어진 고정 전극과 가동 전극, 스프링으로 구성



Roll, Pitch, Yaw

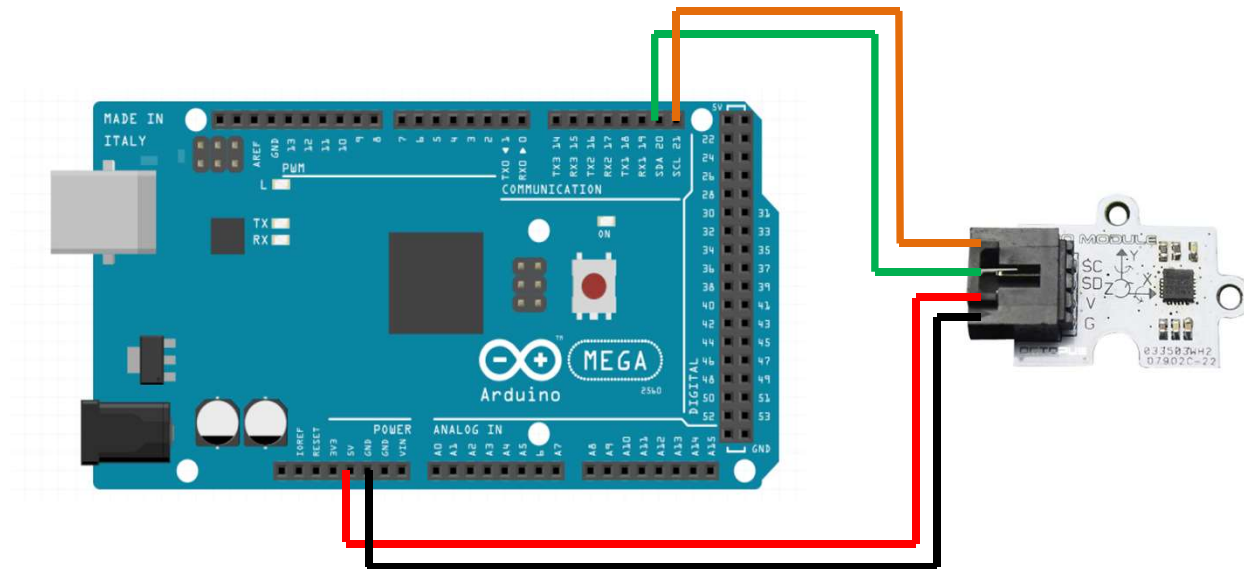


Octopus : ITG3205

센서 모듈 외형	모듈 항목	모듈 항목의 내용
	자이로 센서	ITG3205
	인터페이스	I2C Serial Interface
	ADC 분해능	16Bit
	동작 전압	3.3V/5V
	크기	28x46mm
각속도를 측정할 수 있는 센서 모듈		



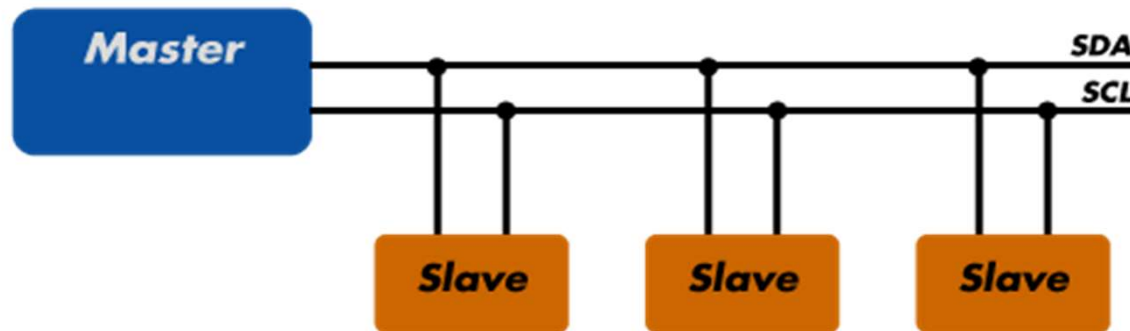
Wiring



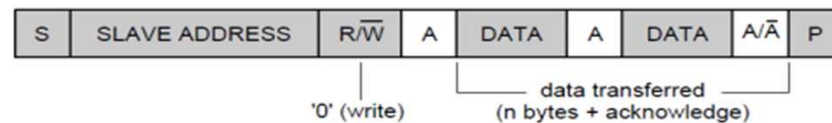
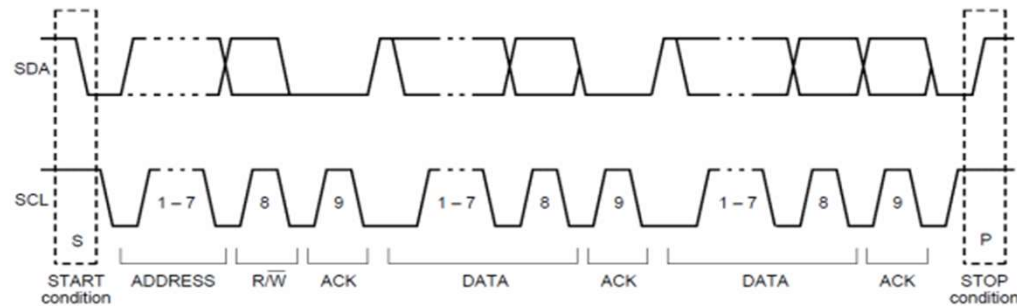
충북대학교 공동훈련센터

I2C 통신

- I2C(Inter-Integrated Circuit)
- 데이터라인(SDA)과 클럭라인(SCL) 2라인의 연결만 이루어 진다
- 1 Master- 다수 Slave관계로 통신
- 아두이노에서는 기본으로 제공하는 Wire라이브러리를 사용



Communication Structure



from master to slave
 from slave to master

A = acknowledge (SDA LOW)
 \bar{A} = not acknowledge (SDA HIGH)
 S = START condition
 P = STOP condition



충북대학교 공동훈련센터

Wire 함수

함 수	기 능
begin()	Wire라이브러리를 초기화하고 I2C통신을 시작한다
requestFrom(address, quantity)	특정주소의 장치(address)에게 지정한 데이터의 양(quantity)만큼 데이터를 요청한다
beginTransmission(address)	마스터장치가 특정주소(address)의 슬레이브 장치로 데이터 전송을 시작한다
endTransmission()	beginTransmission()에 의해 시작된 데이터 전송을 끝낸다
available()	read()함수로 수신가능한 데이터의 바이트수를 반환한다
read()	requestFrom()함수 호출에 의해 슬레이브가 마스터에게 전송한 데이터 한 바이트를 읽어서 반환한다.
write(value)	마스터가 슬레이브에게 데이터를 전송하거나 슬레이브가 마스터에게 데이터를 전송할때 큐에 데이터를 입력하기 위해 사용한다



ITG3205 Register

- WHO_AM_I register : 0x68

Addr Hex	Addr Decimal	Register Name	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	0	WHO_AM_I	R/W	-	ID						-	
15	21	SMPLRT_DIV	R/W	SMPLRT_DIV								
16	22	DLPF_FS	R/W	-	-	-	FS_SEL		DLPF_CFG			
17	23	INT_CFG	R/W	ACTL	OPEN	LATCH_INT_EN	INT_ANYRD_2CLEAR	-	ITG_RDY_EN	-	RAW_RDY_EN	
1A	26	INT_STATUS	R	-	-	-	-	-	ITG_RDY	-	RAW_DATA_RDY	
1B	27	TEMP_OUT_H	R	TEMP_OUT_H								
1C	28	TEMP_OUT_L	R	TEMP_OUT_L								
1D	29	GYRO_XOUT_H	R	GYRO_XOUT_H								
1E	30	GYRO_XOUT_L	R	GYRO_XOUT_L								
1F	31	GYRO_YOUT_H	R	GYRO_YOUT_H								
20	32	GYRO_YOUT_L	R	GYRO_YOUT_L								
21	33	GYRO_ZOUT_H	R	GYRO_ZOUT_H								
22	34	GYRO_ZOUT_L	R	GYRO_ZOUT_L								
3E	62	PWR_MGM	R/W	H_RESET	SLEEP	STBY_XG	STBY_YG	STBY_ZG	CLK_SEL			



Gyro setup(), loop()

```
#include <Wire.h>

#define GYRO_Address 0x68
#define G_SMPLRT_DIV 0x15
#define G_DLPF_FS 0x16
#define G_INT_CFG 0x17
#define G_PWR_MGM 0x3E
#define G_TO_READ 6 // axis x, y, z

int gyro[3];

void setup( ) {
  Serial.begin(115200);
  Wire.begin( );
  initGyro( );
}
```

```
void loop( ) {
  getGyroscopeData(gyro);
  Serial.print(" X = ");
  Serial.print(gyro[0]);
  Serial.print(", Y = ");
  Serial.print(gyro[1]);
  Serial.print(", Z= ");
  Serial.println(gyro[2]);
  delay(100);
}
```



Gyro; get Gyro

```
void initGyro( ){  
    writeTo(GYRO_Address, G_PWR_MGM, 0x00);  
    writeTo(GYRO_Address, G_SMPLRT_DIV, 0x07);  
    writeTo(GYRO_Address, G_DLPF_FS, 0x1E);  
    writeTo(GYRO_Address, G_INT_CFG, 0x00);  
}  
  
void getGyroscopeData(int * result){  
    int regAddress = 0x1D;  
    int temp, x, y, z;  
    byte buff[G_TO_READ];  
  
    readFrom(GYRO_Address, regAddress, G_TO_READ, buff);  
    result[0] = (buff[0] << 8) | buff[1];  
    result[1] = (buff[2] << 8) | buff[3];  
    result[2] = (buff[4] << 8) | buff[5];  
}
```



Gyro; wire

```
void writeTo(int DEVICE, byte address, byte val){
    Wire.beginTransmission(DEVICE);    //start transmission to ACC
    Wire.write(address);                // send register address
    Wire.write(val);                   // send value to write
    Wire.endTransmission( );           //end transmission
}

void readFrom(int DEVICE, byte address, int num, byte buff[ ]){
    int k = 0;
    Wire.beginTransmission(DEVICE);    //start transmission to ACC
    Wire.write(address);                //sends address to read from
    Wire.endTransmission( );           //end transmission
    Wire.beginTransmission(DEVICE);    //start transmission to ACC
    Wire.requestFrom(DEVICE, num);      //request 6 bytes from ACC
    while( Wire.available( ) ) {        //ACC may send less than requested (abnormal)
        buff[k] = Wire.read( );         //receive a byte
        k++;
    }
    Wire.endTransmission( );           //end transmission
}
```

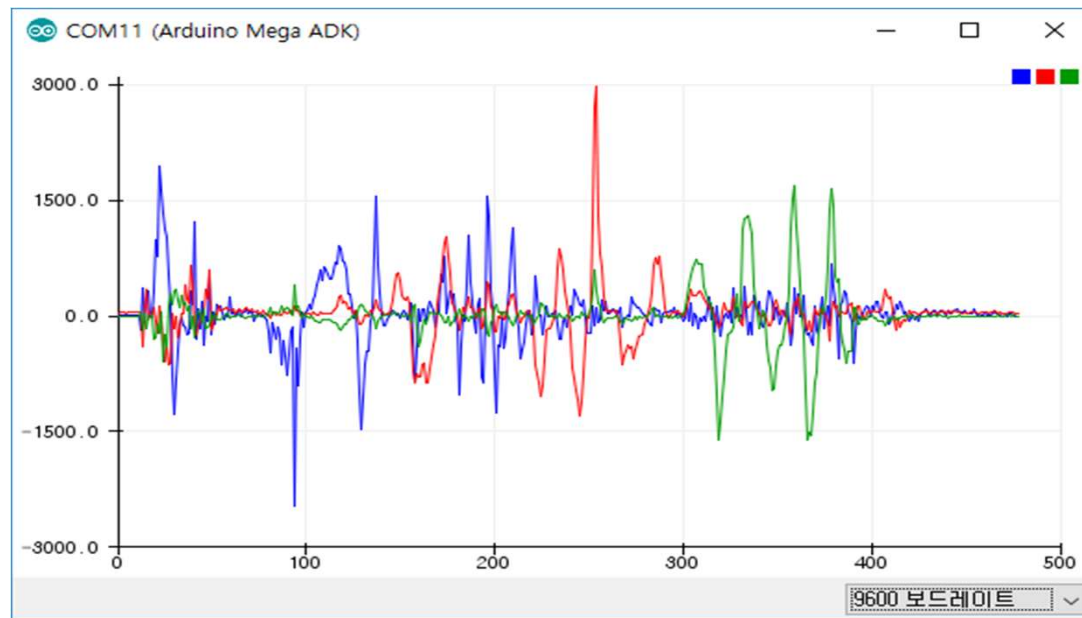


Serial Monitor



충북대학교 공동훈련센터

Serial Plotter



충북대학교 공동훈련센터

Gyro setup(), loop()

```
#include <Wire.h>

#define GYRO_Address 0x68
#define G_SMPLRT_DIV 0x15
#define G_DLPF_FS 0x16
#define G_INT_CFG 0x17
#define G_PWR_MGM 0x3E
#define G_TO_READ 6 // axis x, y, z

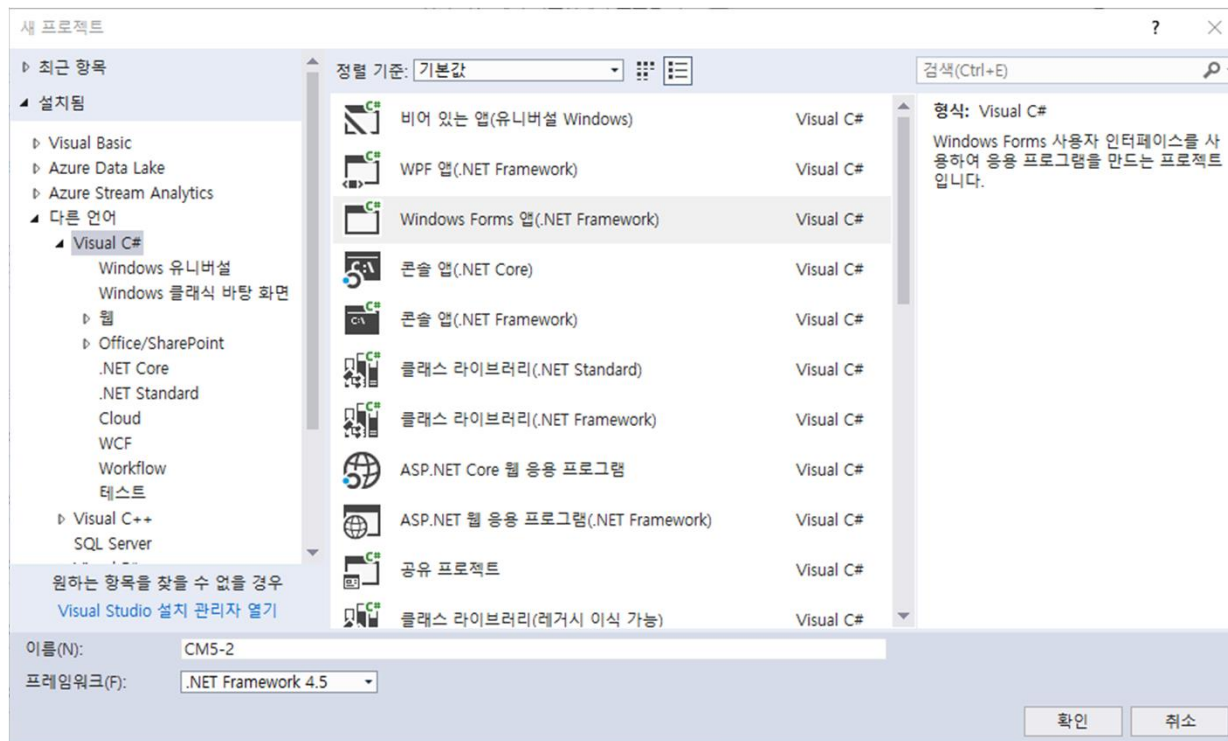
int gyro[3];

void setup( ) {
  Serial.begin(115200);
  Wire.begin( );
  initGyro( );
}
```

```
void loop( ) {
  getGyroscopeData(gyro);
  Serial.print('@');
  Serial.print(gyro[0]);
  Serial.print(',');
  Serial.print(gyro[1]);
  Serial.print(',');
  Serial.print(gyro[2]);
  Serial.print('\n');
  delay(100);
}
```

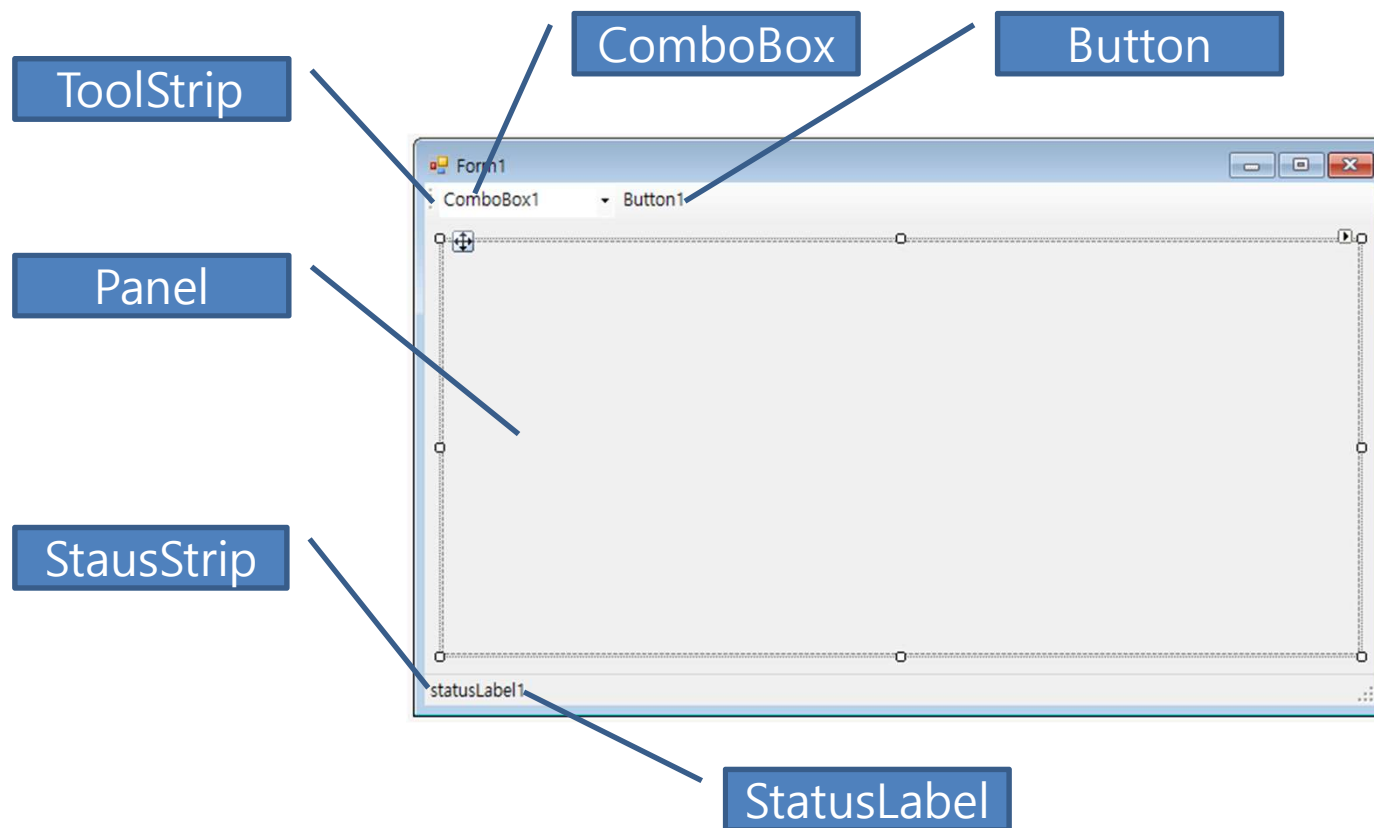


C# Gyro Demo :새 프로젝트

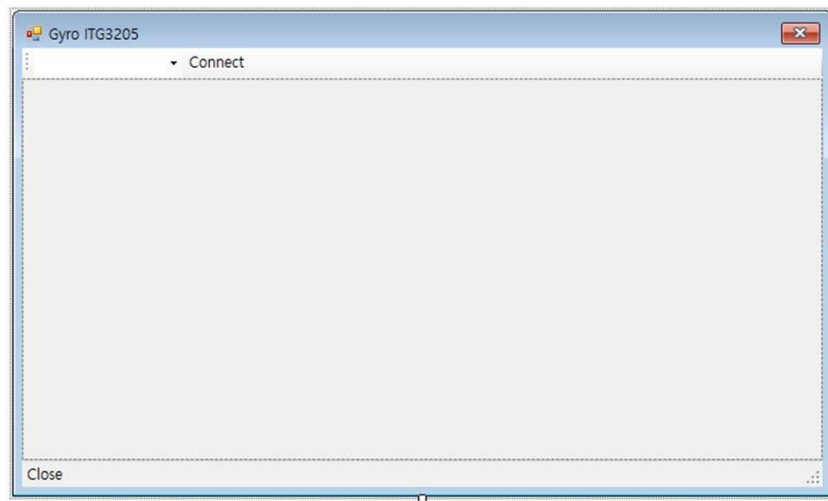


충북대학교 공동훈련센터

Form Design



Property

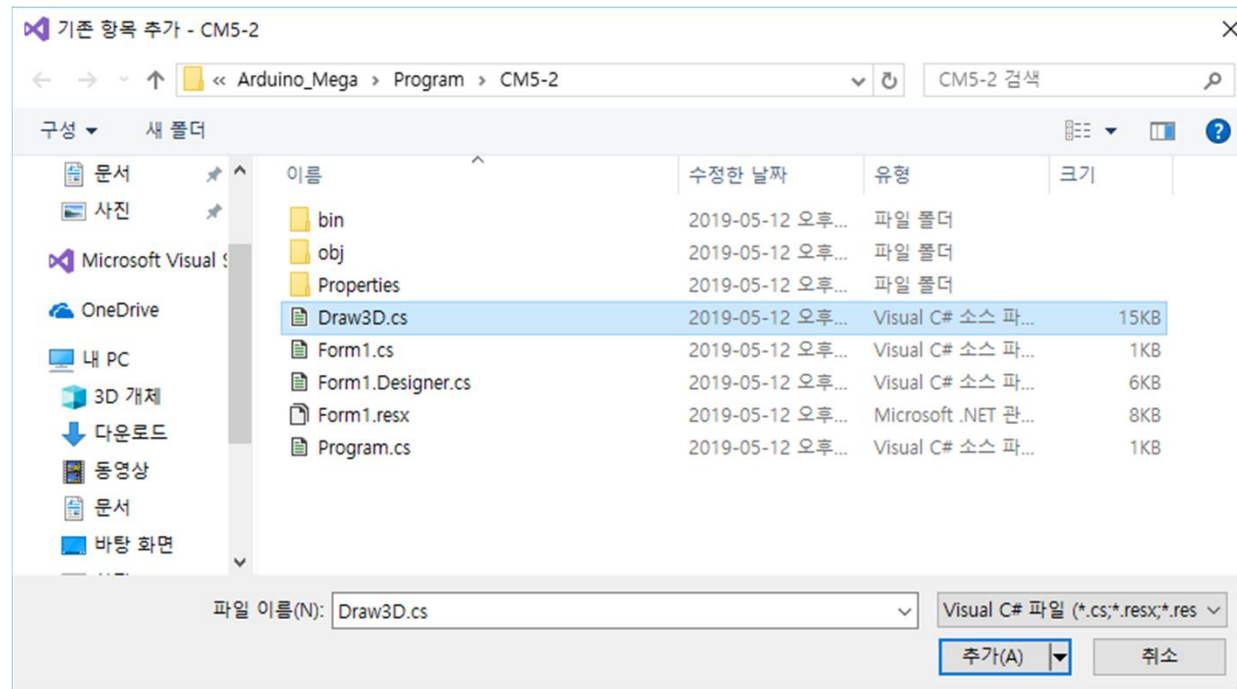


Form1	
Text	Gyro ITG3205
MaximizeBox	False
MinimizeBox	False
Panel1	
Dock	Fill
ToolStripComboBox1	
Name	cmbComPort
ToolStripButton1	
Name	btnConnect
Text	Connect
ToolStripStatusLabel1	
Name	Status
Text	Close



Class 파일 추가

- {솔루션 탐색기}[프로젝트]{Mouse+right} [추가][기존 항목] Draw3D.cs



Define

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;

namespace CM5_2
{
    public partial class Form1 : Form
    {
        SerialPort ComPort = new SerialPort();
        private delegate void SetTextDelegate(string getString);

        Cube cube;
        private float Xaxis = 0;
        private float Yaxis = 0;
        private float Zaxis = 0;
    }
}
```



Form

```
private void Form1_Load(object sender, EventArgs e)
{
    cmbComPort.Items.Clear();
    var portName = System.IO.Ports.SerialPort.GetPortNames();
    cmbComPort.Items.AddRange(portName);
    cmbComPort.SelectedIndex = cmbComPort.Items.Count - 1;

    cube = new Cube(300, 30, 400);
    render();
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (ComPort.IsOpen)
    {
        ComPort.Close();
        ComPort.Dispose();
        ComPort = null;
    }
    Status.Text = "Form Closing.";
}
```



btnConnect_Click

```
private void btnConnect_Click(object sender, EventArgs e)
{
    if (btnConnect.Text == "Connect")
    {
        if (ComPort.IsOpen)
        {
            Status.Text = "Already used.";
        }
        else
        {
            ComPort.PortName = cmbComPort.Text;
            ComPort.BaudRate = 115200;
            ComPort.DataBits = 8;
            ComPort.Parity = Parity.None;
            ComPort.StopBits = StopBits.One;
            ComPort.Handshake = Handshake.None;
            ComPort.Open();
            ComPort.DiscardInBuffer();
            btnConnect.Text = "Close";
            Status.Text = "Port Open.";
        }
    }
    else
    {
        ComPort.Close();
        Status.Text = "Port Close.";
        btnConnect.Text = "Connect";
    }
}
```



DataReceived

```
private void DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    string rxd = ComPort.ReadTo("\n");
    this.BeginInvoke(new SetTextDelegate(SerialReceived), new object[] { rxd });
}

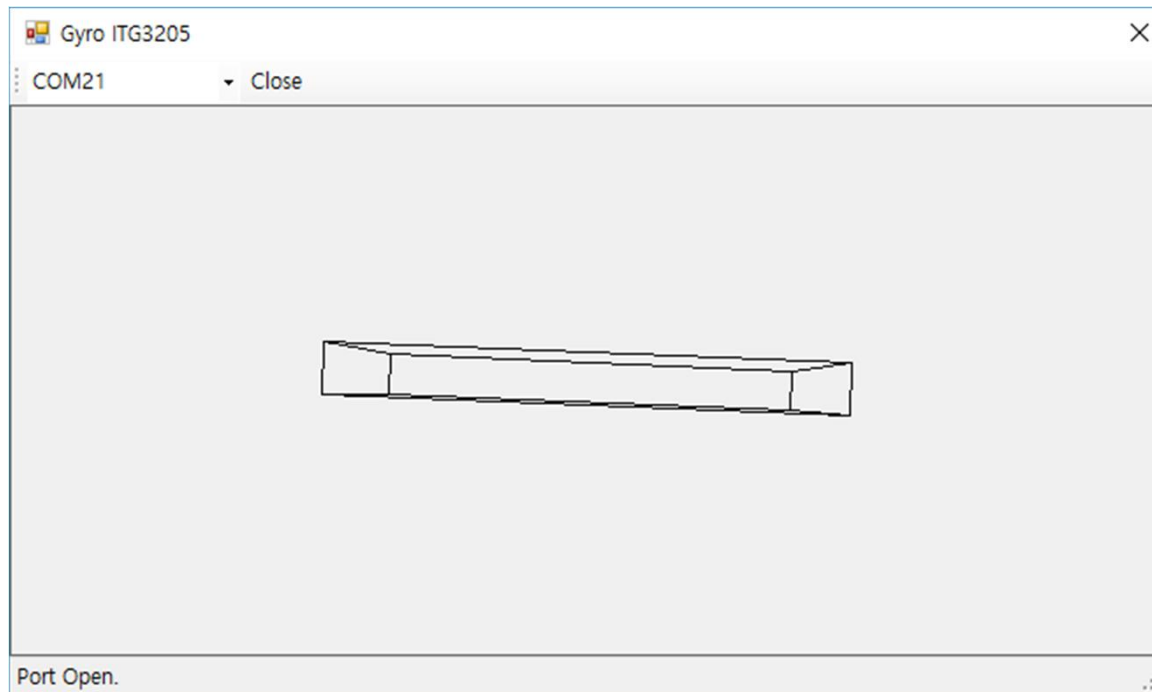
private void SerialReceived(string inString)
{
    string Code = inString.Substring(0, 1);
    string Gyro = inString.Substring(1);

    if (Code == "@")
    {
        string[] Axis = Gyro.Split(',');

        Xaxis = (float)(Convert.ToInt16(Axis[0]) / 10.0);
        Yaxis = (float)(Convert.ToInt16(Axis[1]) / 10.0);
        Zaxis = (float)(Convert.ToInt16(Axis[2]) / 10.0);
        render();
    }
}
```






Debug Run



충북대학교 공동훈련센터

Files

lot Base SW Developer-2019 > Arduino_Mega > Program > CM5-2 > bin > Debug				▼	🔄
이름	수정한 날짜	유형	크기		
 CM5-2.exe	2019-05-12 오후...	응용 프로그램	18KB		
 CM5-2.exe.config	2019-05-12 오후...	XML Configuratio...	1KB		
 CM5-2.pdb	2019-05-12 오후...	프로그램 디버그 ...	38KB		

