IoT Inertial Mesuament Unit Monitoring

Tanawin Siriwan 6203014610060 sec 1

Asst. prof. Dr.Supod Keawkorn

College of industrial technology
Mechatronics Engineering Technology Division

Topic

- Introduction
- Object
- Scope
- Process
- Priciple
- Test
- Problem
- Summarize

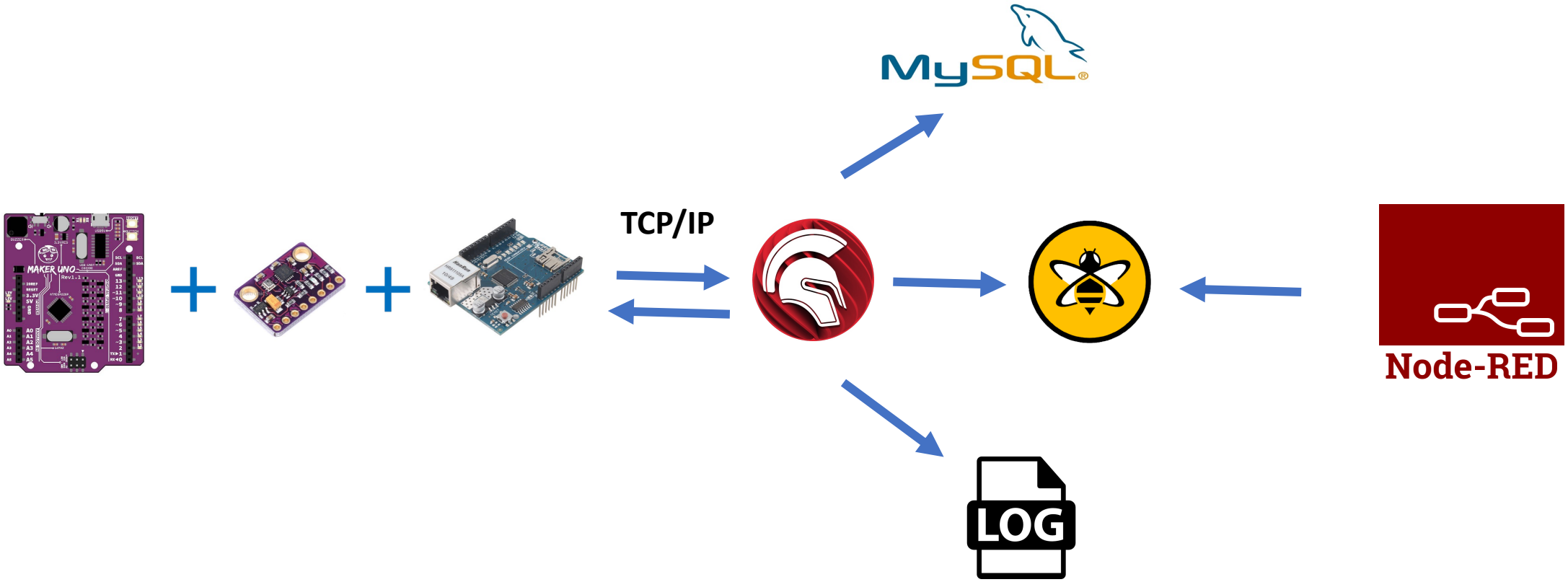
Objective

- For stutdy how to application VCL/FMX using delphi.
- For stutdy how to use delphi connect the controller using TCP/IP.
- For design IMU data monitoring system

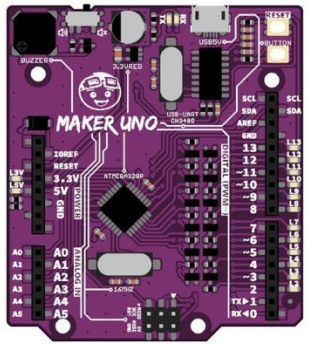
Scope

- Data monitoring
- Transive and resiceve data using TCP/IP
- Control 3D opject movment
- Loggging data file
- Sorted Data in MySQL database
- Send data to MQTT
- Monitoring on Nod-Red

IoT Inertial Mesuament Unit Monitoring



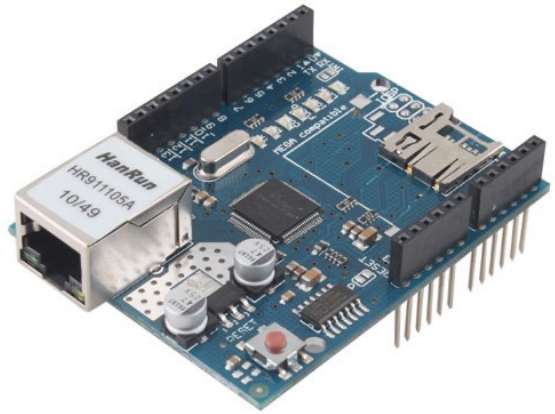
Arduino



SMD ATmega328P microcontroller.
USB Programming facilitated by the CH340.
Input voltage: USB 5V, from computer, power bank or standard USB adapter.
500mA (maximum) 3.3V voltage regulator.
0-5V outputs with 3.3V compatible inputs.
14 Digital I/O Pins (6 PWM outputs).
6 Analog Inputs.
ISP 6-pin Header.
32k Flash Memory.
16MHz Clock Speed.
R3 Shield Compatible.
LED array for 5V, 3.3V, TX, RX and all digital pins.
On board programmable push button (pin 2, need to configure as INPUT_PULLUP).
On board piezo buzzer (pin 8).
Utilize USB Micro-B socket.

Arduino Ethernetsheild

Shield	Name	Arduino® Ethernet Shield Rev 2
	SKU	A000024
	Compatibility	UNO, MEGA
Ethernet	Connector	RJ45
	Controller	W5500
	Speed	10/100 Mbps
	Communication	SPI
	Internal memory	32KB
	Maximum sockets	8 individual
	Supported protocols	IPv4, ICMP, TCP, UDP, ARP, IGMP, PPPoE, MQTT
Storage	Micro SD card slot	
Power	Operating voltage	5V
Connectors	TinkerKit	Yes, 6x



GY-91

MPU9250 + BMP280 module
(Three-axis gyroscope + triaxial accelerometer + triaxial magnetic field + pressure)
Module Model: GY-91
Use chip: MPU-9250 + BMP280
Power supply: 3-5v (internal low dropout regulator)
Communication: standard IIC / SPI communications protocol
-Chip 16bit AD converter, 16-bit data output
Gyroscopes range: ± 250 500 1000 2000 ° / s
Acceleration range: ± 2 ± 4 ± 8 ± 16g
Field range: ± 4800uT
Pressure range: 300-1100hPa

Arduino I2C Connected IMU

```ino
#include "SPI.h"
#include "Ethernet.h"
#include <MPU9250_asukiaaa.h>
#include <Adafruit_BMP280.h>

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
EthernetServer server(80);
int v;
float voltage;

int trigPin = 2;  //Assign D2 as trigPin
int echoPin = 4;  //Assign D4 as echoPin
long duration;
float distance;

Adafruit_BMP280 bme; // I2C
MPU9250_asukiaaa mySensor;
float aX, aY, aZ, aSqrt, gX, gY, gZ, mDirection, mX, mY, mZ;


void setup() {
  Serial.begin(115200); //Set Serial Communication
  Ethernet.begin(mac);  //
  server.begin();       //Start Arduino as Server role
  Serial.print("Arduino as Server Role IPaddress: ");
  Serial.println(Ethernet.localIP());

  while (!Serial);

#ifdef _ESP32_HAL_I2C_H_ // For ESP32
  Wire.begin(SDA_PIN, SCL_PIN);
  mySensor.setWire(&Wire);
#else
  Wire.begin();
  mySensor.setWire(&Wire);
#endif

  bme.begin();
  mySensor.beginAccel();
  mySensor.beginGyro();
  mySensor.beginMag();
}
```

```ino
void loop() {
  EthernetClient client = server.available(); //Wait connection from TCP/IP
  if (client) {
    Serial.println("Hi...New Client");
    while (client.connected()) {
      while (client.available()) {
        char data = client.read();
        Serial.println(data);
        switch (data) {
          case 'a':                   // data='a'--> Do nothing
            Serial.print("");
            client.println(""); //Arduino as Server send data to LabVIEW as Client
            break;
          case 'b':      // data='b' Read voltage from A0 and send back to LabVIEW
            if (mySensor.accelUpdate() == 0) {
              aX = mySensor.accelX();
              aY = mySensor.accelY();
              aZ = mySensor.accelZ();
              aSqrt = mySensor.accelSqrt();
              Serial.print(String(aX)); Serial.print(",");
              Serial.print(String(aY)); Serial.print(",");
              Serial.print(String(aZ)); Serial.print(",");
              Serial.print(String(aSqrt)); Serial.print(",");
              //-----------------------------------------------------//
              client.print(String(aX)); client.print(",");
              client.print(String(aY)); client.print(",");
              client.print(String(aZ)); client.print(",");
              client.print(String(aSqrt)); client.print(",");
            }
```

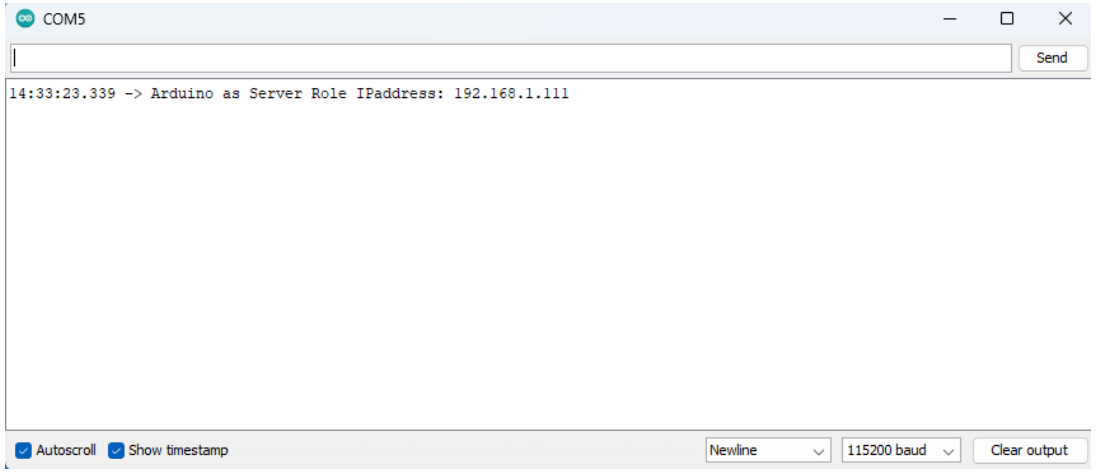
Arduino I2C Connected IMU

```
74            if (mySensor.gyroUpdate() == 0) {
75              gX = mySensor.gyroX();
76              gY = mySensor.gyroY();
77              gZ = mySensor.gyroZ();
78              client.print(String(gX)); client.print(",");
79              client.print(String(gY)); client.print(",");
80              client.print(String(gZ)); client.print(",");
81              //-------------------------------------------------//
82              Serial.print(String(gX)); Serial.print(",");
83              Serial.print(String(gY)); Serial.print(",");
84              Serial.print(String(gZ)); Serial.print(",");
85            }
86
87            if (mySensor.magUpdate() == 0) {
88              mX = mySensor.magX();
89              mY = mySensor.magY();
90              mZ = mySensor.magZ();
91              mDirection = mySensor.magHorizDirection();
92              Serial.print(String(mX)); Serial.print(",");
93              Serial.print(String(mY)); Serial.print(",");
94              Serial.print(String(mZ)); Serial.print(",");
95              Serial.print(String(mDirection)); Serial.print(",");
96              //-------------------------------------------------//
97              client.print(String(mX)); client.print(",");
98              client.print(String(mY)); client.print(",");
99              client.print(String(mZ)); client.print(",");
100             client.print(String(mDirection)); client.print(",");
101           }
```

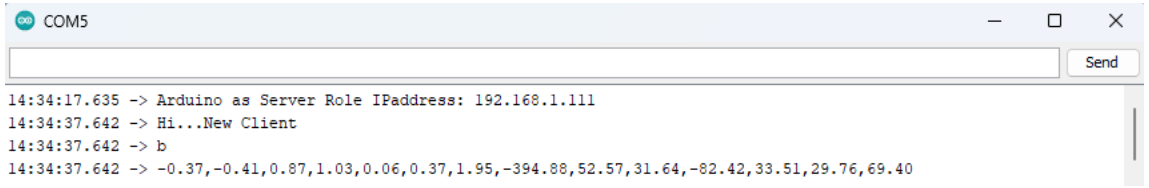
```
102
103           //  Serial.print("\tTemperature(*C): ");
104           Serial.print(bme.readTemperature()); Serial.print(",");
105           //-------------------------------------------------//
106           client.print(bme.readTemperature()); client.print(",");
107
108           //  Serial.print("\tPressure(Inches(Hg)): ");
109           Serial.print(bme.readPressure() / 3377); Serial.print(",");
110           //-------------------------------------------------//
111           client.print(bme.readPressure() / 3377); client.print(",");
112
113
114           //  Serial.print("\tApproxAltitude(m): ");
115           Serial.print(bme.readAltitude(1013.25)); // this should be adjusted to your local forcase
116           //-----------------------------------------//
117           client.print(bme.readAltitude(1013.25)); // this should be adjusted to your local forcase
118
119           Serial.println(""); // Add an empty line
120           //-----------------------------------------//
121           client.println("");
122           break;
123         }
124       }
125       }
126     }
127   }
```

Arduino I2C Connected IMU

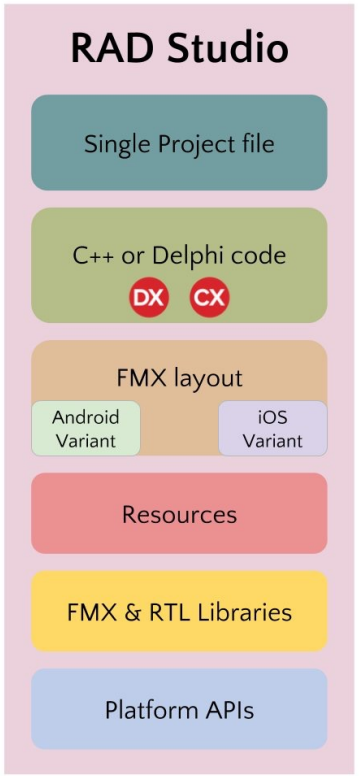
Arduino IP

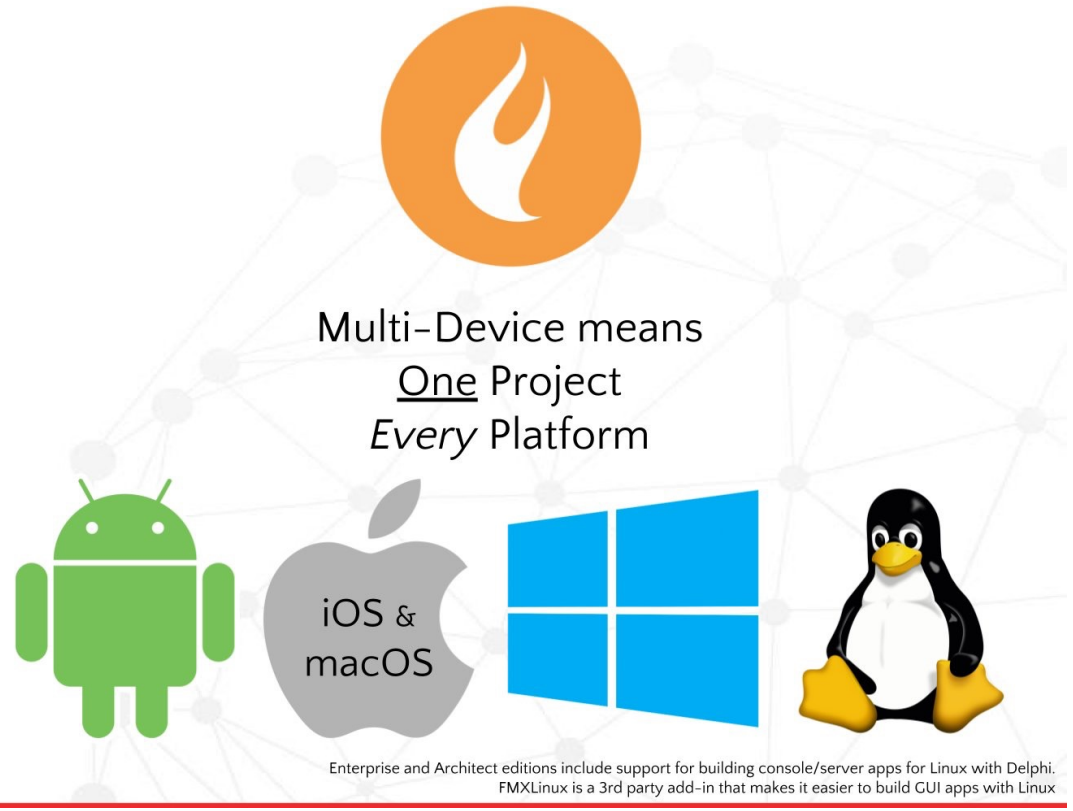


Delphi Connected

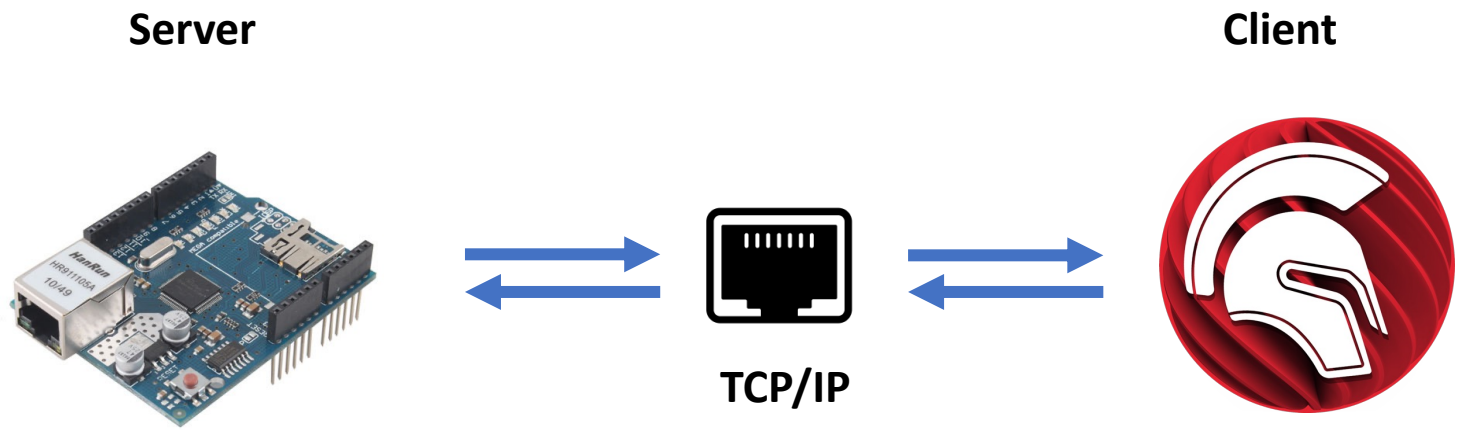


Impremention On FMX Application

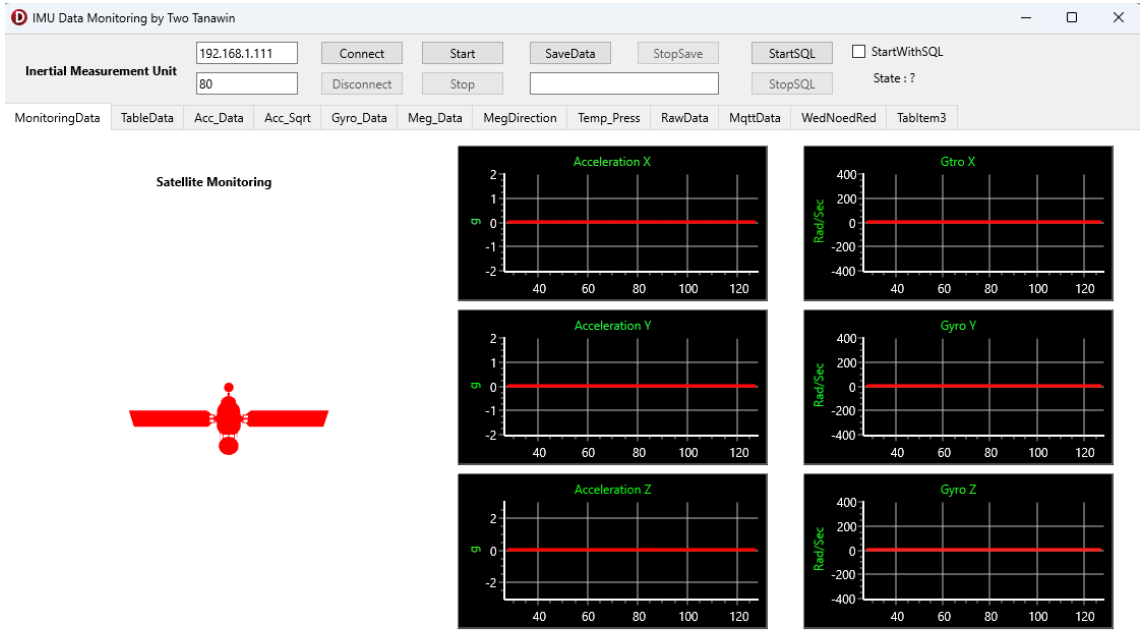




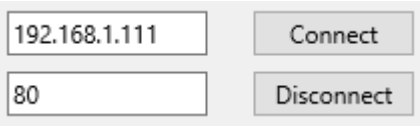
Delphi Conect Arduino Using TCP/IP



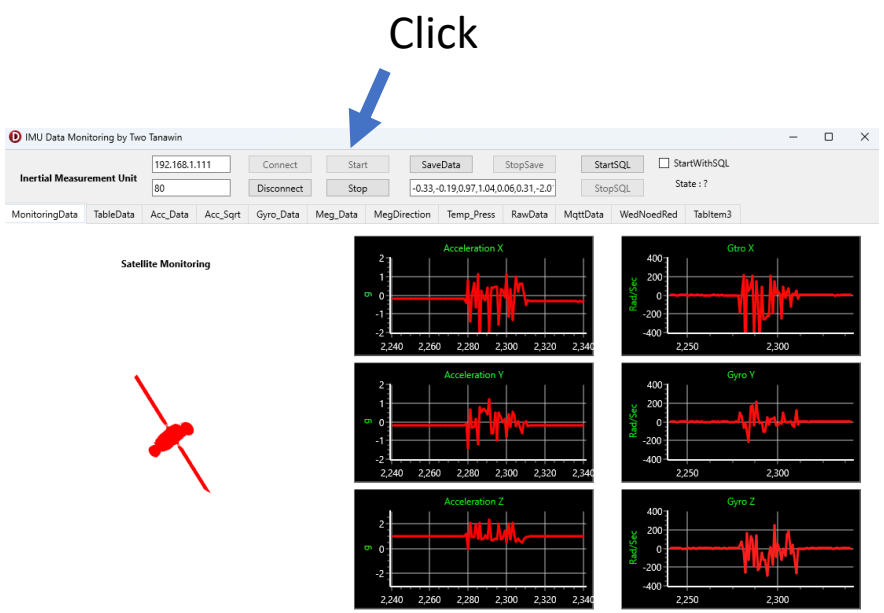
Delphi TCP/IP Connected Controller




```
procedure TForm1.ConnectClick(Sender: TObject);
begin
 IdTCPClient1.Connect;
end;

procedure TForm1.DisconnectClick(Sender: TObject);
begin
  IdTCPClient1.Disconnect;
end;
```



Delphi Start System




```
procedure TForm1.StartClick(Sender: TObject);
begin
if IdTCPClient1.Connected=False then
begin
  ShowMessage('Please Connect The Sever');
end
else
begin
  Start.Enabled:=False;
  Stop.Enabled:=True;
  StartClent;
  Timer1.Enabled:=True;
  if CheckBox1.IsChecked then
  begin
    ThreadSqlStart;
  end
  else
  begin
    EndThreadSql;
  end;
end;
end;
```

```
procedure TForm1.StopClick(Sender: TObject);
begin
StopClient;
EndThreadSQL;
Timer1.Enabled:=False;
Start.Enabled:=True;
Stop.Enabled:=False;
if CheckBox1.IsChecked then
begin
  EndThreadSQL;
end;
end;
```

Delphi Extrax Data

```
type
TClientRead = class(TThread)
private
protected
 procedure Execute; override;
end;
```

```
procedure TForm1.StartClent;
begin
 if MyCleint = nil then
  MyCleint := TClientRead.create(false);
end;
```

```
procedure TClientRead.Execute;
var
 Sl:TStringList;
 Data:String;
begin
  inherited;
  try
  repeat
  Application.ProcessMessages;
  Form1.IdTCPClient1.Socket.WriteLn('b'+#13#10);
  Form1.Edit3.Text:=Form1.IdTCPClient1.Socket.ReadLn;
  Form1.AddLog(Form1.Edit3.text);
  Sl:=TStringList.Create;
  Sl.CommaText:=Form1.Edit3.Text;  //Extrax string
   if Sl.Count = 14 then
   begin
   Form1.accData.aX:=Sl[0];
   Form1.accData.aY:=Sl[1];
   Form1.accData.aZ:=Sl[2];
   Form1.accData.rq:=Sl[3];
   Form1.gyroData.gyroX:=Sl[4];
   Form1.gyroData.gyroY:=Sl[5];
   Form1.gyroData.gyroZ:=Sl[6];
   Form1.magData.magX:=Sl[7];
   Form1.magData.magY:=Sl[8];
   Form1.magData.magZ:=Sl[9];
   Form1.magData.magH:=Sl[10];
   Form1.tempData.TempX:=Sl[11];
   Form1.tempData.PressX:=Sl[12];
   lastData:=Sl[13];
   end;
 if Assigned(Sl) then
 FreeAndNil(Sl);
 Form1.recMyData;
 until Terminated;
  Except
    ShowMessage('Please Connecte The Sever!');
  end;

end;
```

```
Type
  tAcc = record
    X:Single;
    Y:Single;
    Z:Single;
  end;

type
 TaccData = record
  aX:String;
  aY:String;
  aZ:String;
  rq:String;
 end;

type
 TgyroData = record
  gyroX:String;
  gyroY:String;
  gyroZ:String;
 end;

type
 TmagData = record
  magX:String;
  magY:String;
  magZ:String;
  magH:String;
 end;

type
 TtempData = record
  TempX:String;
  PressX:String;
 end;
```

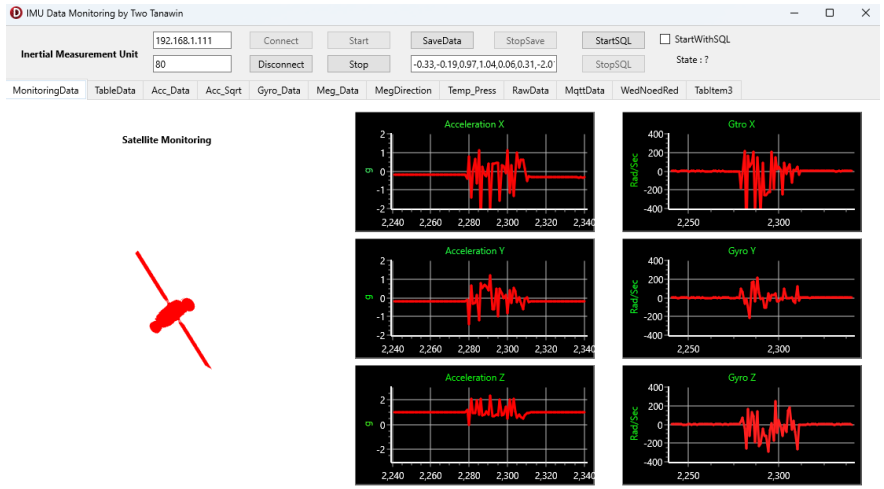
Delphi Plot Graph

```
procedure TMyPlot.Execute;
begin
  inherited;
  repeat
  Application.ProcessMessages;
  Sleep(100);
  Synchronize(Form1.SumPlot);
  until Terminated;
end;

procedure TForm1.SumPlot; // All Plot Data
begin
 PlotAccD();
 PlotGyroD();
 PlotNorm();
 PlotMag();
 PlotMagH();
 Teamp_Press();
end;
```

```
procedure TForm1.PlotAcc; //Plot Acc
begin
  PlotLineGraph(AccX, StrToFloat(accData.aX));
  PlotLineGraph(AccY, StrToFloat(accData.aY));
  PlotLineGraph(AccZ, StrToFloat(accData.aZ));
end;

procedure TForm1.PlotAccD; //Plot Acc
begin
  PlotLineGraph(AccXD, StrToFloat(accData.aX));
  PlotLineGraph(AccYD, StrToFloat(accData.aY));
  PlotLineGraph(AccZD, StrToFloat(accData.aZ));
end;

procedure TForm1.PlotGyro; //Plot Gyro
begin
  PlotLineGraph(GyroX, StrToFloat(gyroData.gyroX));
  PlotLineGraph(GyroY, StrToFloat(gyroData.gyroY));
  PlotLineGraph(GyroZ, StrToFloat(gyroData.gyroZ));
end;

procedure TForm1.PlotGyroD; //Plot Gyro
begin
  PlotLineGraph(GyroXD, StrToFloat(gyroData.gyroX));
  PlotLineGraph(GyroYD, StrToFloat(gyroData.gyroY));
  PlotLineGraph(GyroZD, StrToFloat(gyroData.gyroZ));
end;
```

```
procedure TForm1.PlotLineGraph(Graph: TLineSeries; Data: Real);
var
 tmpX:Double;
begin
 with Graph do
  begin
    if XValues.Count<100 then
     begin
      Add(Data);
     end
    else
     begin
      tmpX:=XValues[1]-XValues[0];
      Delete(0);
      AddXY( XValues.Last+tmpX, Data,'',clTeeColor);
     end;
  end;
end;
```

Graph In FMX Application

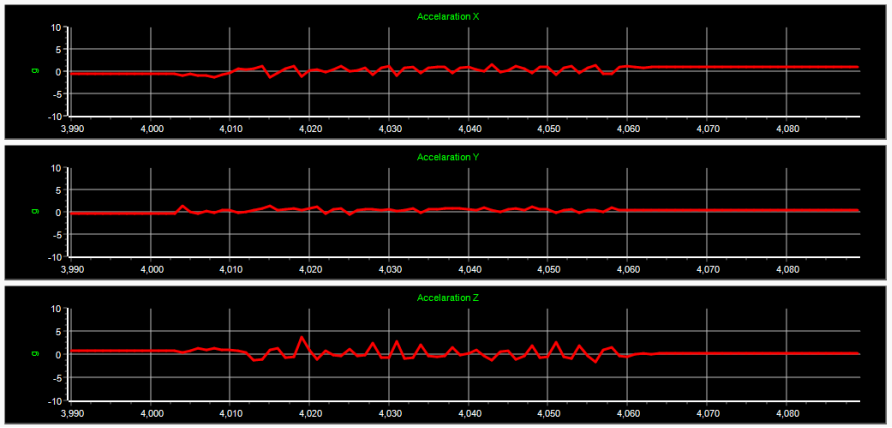



```
60  type
    Tmy3d = class(TThread)
      private
      procedure callSig();
      protected
      procedure Execute; override;
    end;

    procedure Tmy3d.callSig;
1330 var
      Sl:TStringList;
      Data:String;
      i:Integer;
    begin
      Application.ProcessMessages;
      Form1.Model3D1.Position.X := StrToFloat(Form1.accData.aX);
      Form1.Model3D1.Position.Y := StrToFloat(Form1.accData.aY);
      Form1.Model3D1.Position.Z := StrToFloat(Form1.accData.aZ);
      Form1.Model3D1.RotationAngle.X := StrToFloat(Form1.gyroData.gyroX);
1340  Form1.Model3D1.RotationAngle.Y := StrToFloat(Form1.gyroData.gyroY);
      Form1.Model3D1.RotationAngle.Z := StrToFloat(Form1.gyroData.gyroZ);
    end;

    procedure Tmy3d.Execute;
    begin
      inherited;
      repeat
      Application.ProcessMessages;
      Sleep(100);
1350  Synchronize(callSig);
      until Terminated;
    end;
```

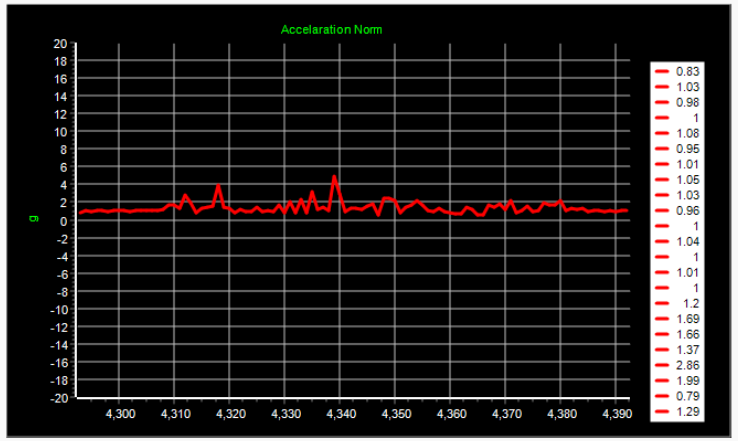
```
    type
    TModel = class(TThread)
    private
    protected
     procedure Execute; override;
    end;

    procedure TModel.Execute;
    begin
      inherited;
      repeat
      Application.ProcessMessages;
      Form1.PlotAcc();
      Form1.PlotGyro();
      Sleep(100);
      until Terminated;
1230 end;
```

Graph In FMX Application

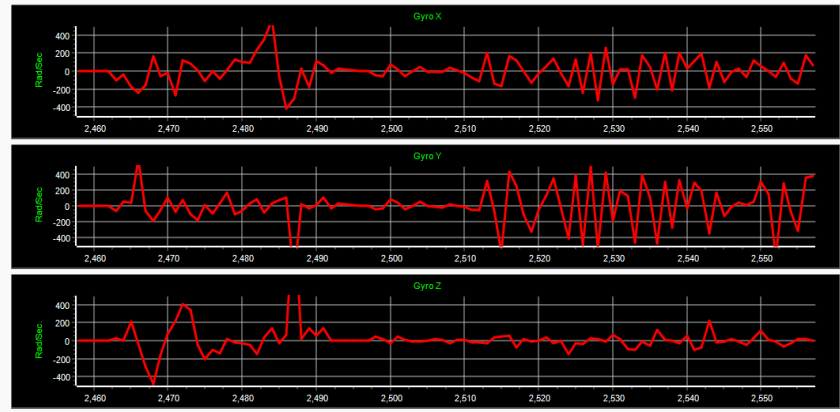
Acceleration XYZ



Acceleration Norm

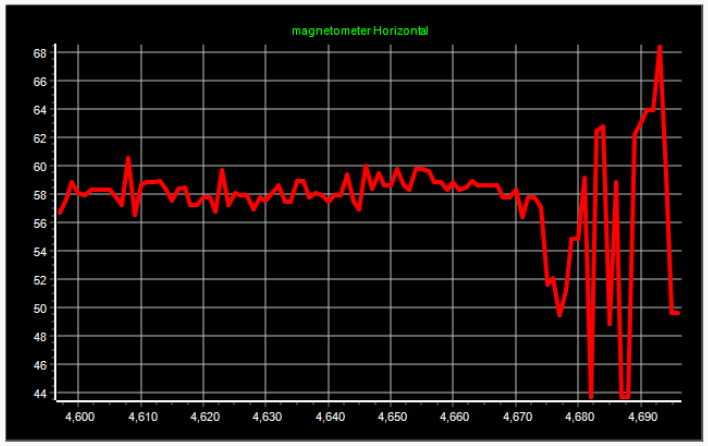


Gyro XYZ

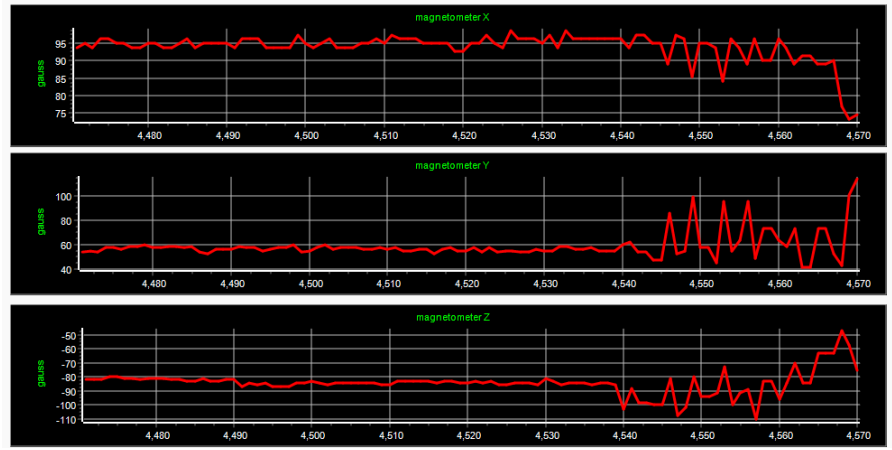


Graph In FMX Application

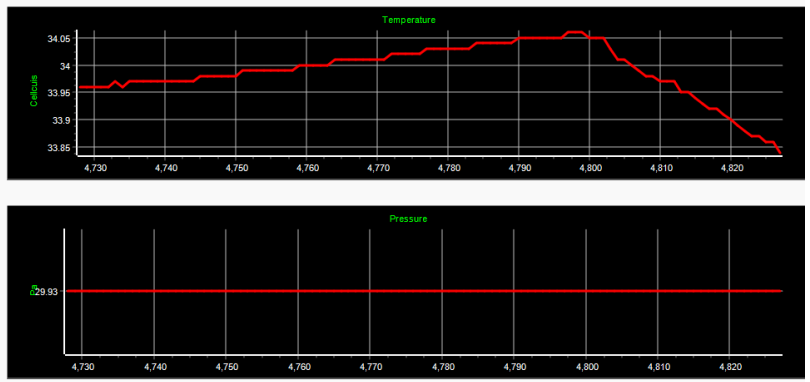
Magnetrometer Horizontal



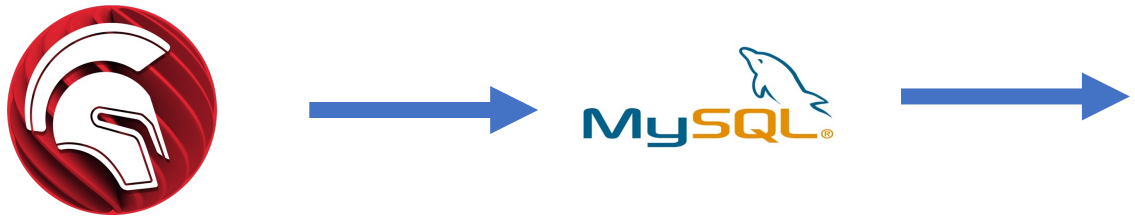
Magnetrometer XYZ

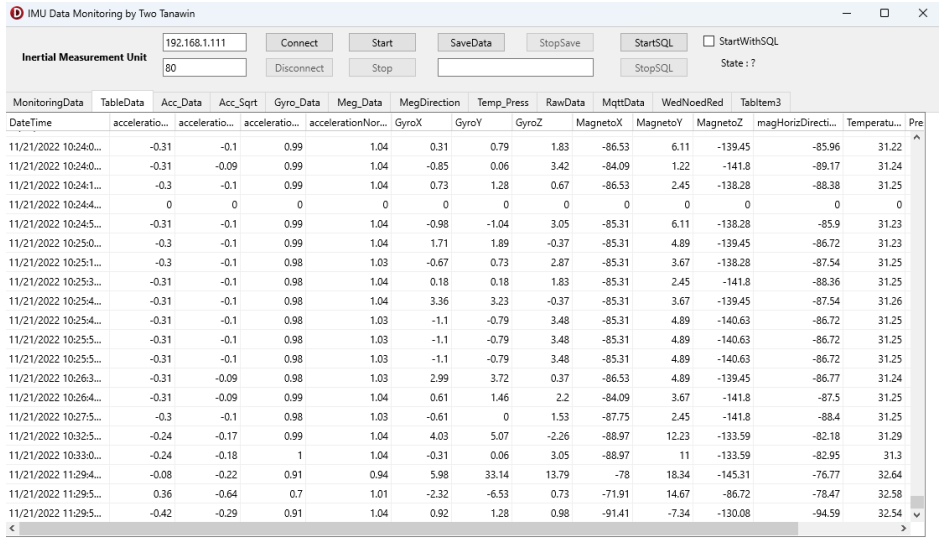


Temp & Pressure



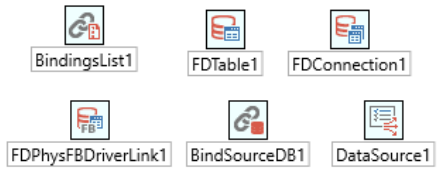
Delphi Store Data Using MySQL

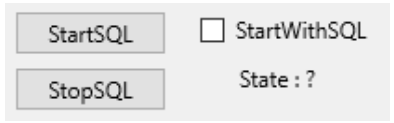




KMUTNB
INVENTION TO INNOVATION

Delphi Record Data Using MySQL





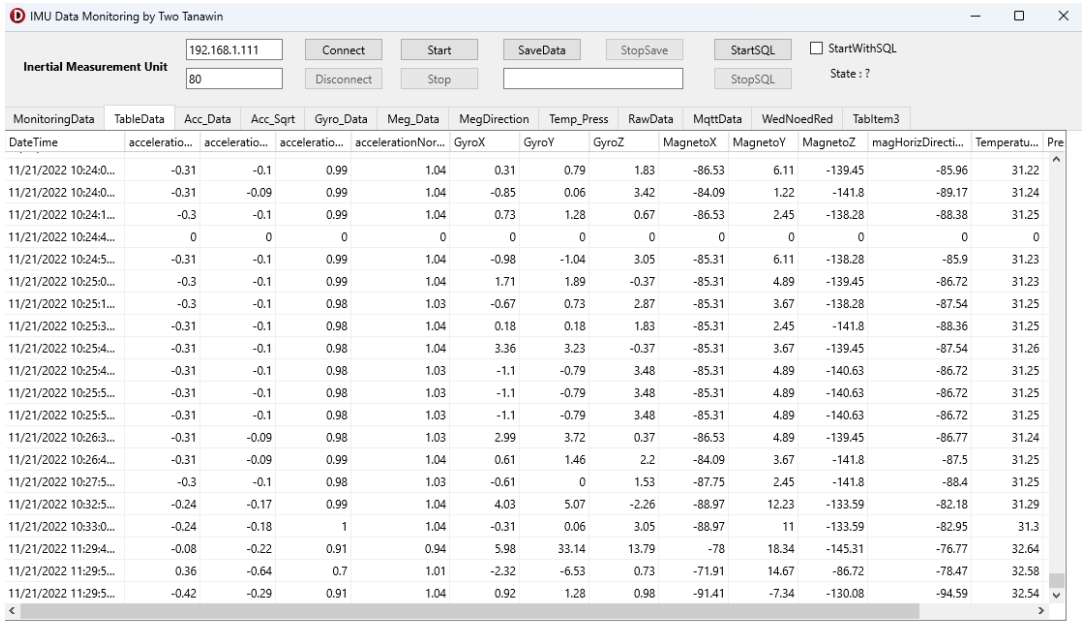

```
type
TMyThreadSQL = class(TThread)
private
LastData:String;
procedure DataToSQL(aX,aY,aZ,aN,gR,gP,gY,mgX,mgY,mgZ,mgH,temp,press,Alti:Single);
procedure ShowDataSQL();
protected
 procedure Execute; override;
end;
```

```
procedure TForm1.StartSQLClick(Sender: TObject);
begin
if Start.Enabled=True then
begin
 ShowMessage('Please Enable Start');
end
else
begin
ThreadSqlStart;
StartSQL.Enabled:=False;
StopSQL.Enabled:=True;
end;
end;

procedure TForm1.StopClick(Sender: TObject);
begin
StopClient;
EndThreadSQL;
Timer1.Enabled:=False;
Start.Enabled:=True;
Stop.Enabled:=False;
if CheckBox1.IsChecked then
begin
  EndThreadSQL;
end;
end;
```

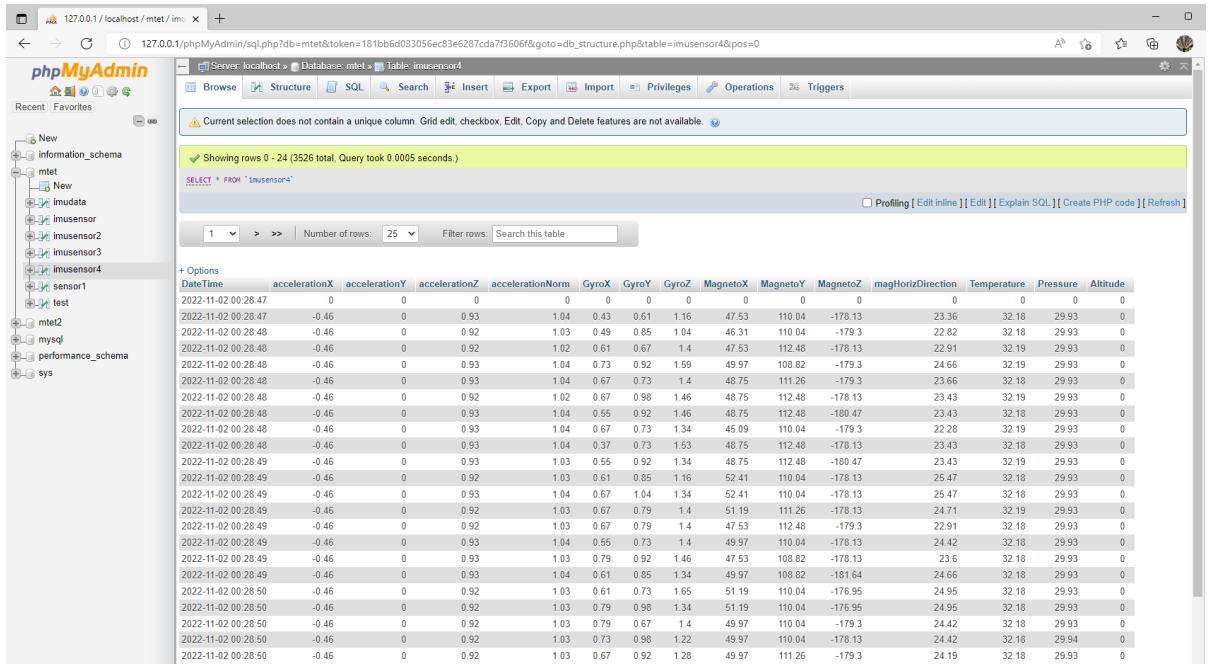
```
procedure TMyThreadSQL.Execute;
begin
  inherited;
    inherited;
    Form1.SQLStat.Text:='SQL is Start';
    lastData:='0';
    repeat
    Application.ProcessMessages;
    Synchronize(ShowDataSQL);
    Sleep(5000);
    until Terminated;
end;

procedure TMyThreadSQL.ShowDataSQL;
begin
 DataToSQL(StrToFloat(Form1.accData.aX),StrToFloat(Form1.accData.aY),StrToFloat(Form1.accData.aZ),StrToFloat(Form1.accData.rq),
 StrToFloat(Form1.gyroData.gyroX),StrToFloat(Form1.gyroData.gyroY),StrToFloat(Form1.gyroData.gyroZ),
 StrToFloat(Form1.magData.magX),StrToFloat(Form1.magData.magY),StrToFloat(Form1.magData.magZ),StrToFloat(Form1.magData.magH),
 StrToFloat(Form1.tempData.TempX),StrToFloat(Form1.tempData.PressX),StrToFloat(lastData));
end;
```

KMUTNB INVENTION TO INNOVATION

MySQL Database Result

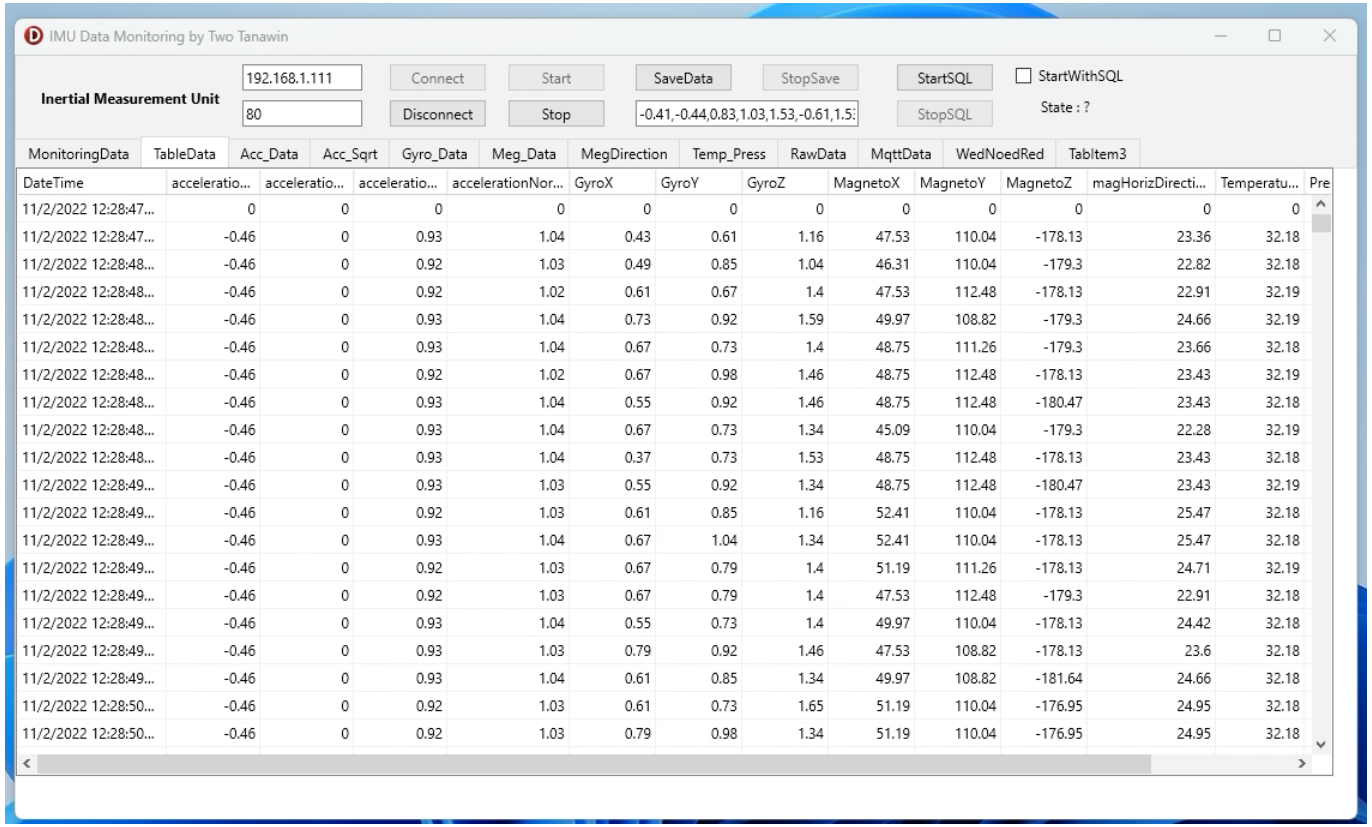


DateTime	acceleratio...	acceleratio...	acceleratio...	accelerationNor...	GyroX	GyroY	GyroZ	MagnetoX	MagnetoY	MagnetoZ	magHorizDirecti...	Temperatu...	Pre
11/21/2022 10:24:0...	-0.31	-0.1	0.99	1.04	0.31	0.79	1.83	-86.53	6.11	-139.45	-85.96	31.22	
11/21/2022 10:24:0...	-0.31	-0.09	0.99	1.04	-0.85	0.06	3.42	-84.09	1.22	-141.8	-89.17	31.24	
11/21/2022 10:24:1...	-0.3	-0.1	0.99	1.04	0.73	1.28	0.67	-86.53	2.45	-138.28	-88.38	31.25	
11/21/2022 10:24:4...	0	0	0	0	0	0	0	0	0	0	0	0	
11/21/2022 10:24:5...	-0.31	-0.1	0.99	1.04	-0.98	-1.04	3.05	-85.31	6.11	-138.28	-85.9	31.23	
11/21/2022 10:25:0...	-0.3	-0.1	0.99	1.04	1.71	1.89	-0.37	-85.31	4.89	-139.45	-86.72	31.23	
11/21/2022 10:25:1...	-0.3	-0.1	0.98	1.03	-0.67	0.73	2.87	-85.31	3.67	-138.28	-87.54	31.25	
11/21/2022 10:25:3...	-0.31	-0.1	0.98	1.04	0.18	0.18	1.83	-85.31	2.45	-141.8	-88.36	31.25	
11/21/2022 10:25:4...	-0.31	-0.1	0.98	1.04	3.36	3.23	-0.37	-85.31	3.67	-139.45	-87.54	31.26	
11/21/2022 10:25:4...	-0.31	-0.1	0.98	1.03	-1.1	-0.79	3.48	-85.31	4.89	-140.63	-86.72	31.25	
11/21/2022 10:25:5...	-0.31	-0.1	0.98	1.03	-1.1	-0.79	3.48	-85.31	4.89	-140.63	-86.72	31.25	
11/21/2022 10:25:5...	-0.31	-0.1	0.98	1.03	-1.1	-0.79	3.48	-85.31	4.89	-140.63	-86.72	31.25	
11/21/2022 10:26:3...	-0.31	-0.09	0.98	1.03	2.99	3.72	0.37	-86.53	4.89	-139.45	-86.77	31.24	
11/21/2022 10:26:4...	-0.31	-0.09	0.99	1.04	0.61	1.46	2.2	-84.09	3.67	-141.8	-87.5	31.25	
11/21/2022 10:27:5...	-0.3	-0.1	0.98	1.03	-0.61	0	1.53	-87.75	2.45	-141.8	-88.4	31.25	
11/21/2022 10:32:5...	-0.24	-0.17	0.99	1.04	4.03	5.07	-2.26	-88.97	12.23	-133.59	-82.18	31.29	
11/21/2022 10:33:0...	-0.24	-0.18	1	1.04	-0.31	0.06	3.05	-88.97	11	-133.59	-82.95	31.3	
11/21/2022 11:29:4...	-0.08	-0.22	0.91	0.94	5.98	33.14	13.79	-78	18.34	-145.31	-76.77	32.64	
11/21/2022 11:29:5...	0.36	-0.64	0.7	1.01	-2.32	-6.53	0.73	-71.91	14.67	-86.72	-78.47	32.58	
11/21/2022 11:29:5...	-0.42	-0.29	0.91	1.04	0.92	1.28	0.98	-91.41	-7.34	-130.08	-94.59	32.54	



DateTime	accelerationX	accelerationY	accelerationZ	accelerationNorm	GyroX	GyroY	GyroZ	MagnetoX	MagnetoY	MagnetoZ	magHorizDirection	Temperature	Pressure	Altitude
2022-11-02 00:28:47	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2022-11-02 00:28:47	-0.46	0	0.93	1.04	0.43	0.61	1.16	47.53	110.04	-178.13	23.36	32.18	29.93	0
2022-11-02 00:28:48	-0.46	0	0.92	1.03	0.49	0.85	1.04	46.31	110.04	-179.3	22.82	32.18	29.93	0
2022-11-02 00:28:48	-0.46	0	0.92	1.02	0.61	0.67	1.4	47.53	112.48	-178.13	22.91	32.19	29.93	0
2022-11-02 00:28:48	-0.46	0	0.93	1.04	0.73	0.92	1.59	49.97	108.82	-179.3	24.66	32.19	29.93	0
2022-11-02 00:28:48	-0.46	0	0.93	1.04	0.67	0.73	1.4	48.75	111.26	-179.3	23.66	32.18	29.93	0
2022-11-02 00:28:48	-0.46	0	0.92	1.02	0.67	0.98	1.46	48.75	112.48	-178.13	23.43	32.19	29.93	0
2022-11-02 00:28:48	-0.46	0	0.93	1.04	0.55	0.92	1.46	48.75	112.48	-180.47	23.43	32.18	29.93	0
2022-11-02 00:28:48	-0.46	0	0.93	1.04	0.67	0.73	1.34	45.09	110.04	-179.3	22.28	32.19	29.93	0
2022-11-02 00:28:48	-0.46	0	0.93	1.04	0.37	0.73	1.53	48.75	112.48	-178.13	23.43	32.18	29.93	0
2022-11-02 00:28:49	-0.46	0	0.93	1.03	0.55	0.92	1.34	48.75	112.48	-180.47	23.43	32.19	29.93	0
2022-11-02 00:28:49	-0.46	0	0.92	1.03	0.61	0.85	1.16	52.41	110.04	-178.13	25.47	32.18	29.93	0
2022-11-02 00:28:49	-0.46	0	0.93	1.04	0.67	1.04	1.34	52.41	110.04	-178.13	25.47	32.18	29.93	0
2022-11-02 00:28:49	-0.46	0	0.92	1.03	0.67	0.79	1.4	51.19	111.26	-178.13	24.71	32.19	29.93	0
2022-11-02 00:28:49	-0.46	0	0.92	1.03	0.67	0.79	1.4	47.53	112.48	-179.3	22.91	32.18	29.93	0
2022-11-02 00:28:49	-0.46	0	0.93	1.04	0.55	0.73	1.4	49.97	110.04	-178.13	24.42	32.18	29.93	0
2022-11-02 00:28:49	-0.46	0	0.93	1.03	0.79	0.92	1.46	47.53	108.82	-178.13	23.6	32.18	29.93	0
2022-11-02 00:28:49	-0.46	0	0.93	1.04	0.61	0.85	1.34	49.97	108.82	-181.64	24.66	32.18	29.93	0
2022-11-02 00:28:50	-0.46	0	0.92	1.03	0.61	0.73	1.65	51.19	110.04	-176.95	24.95	32.18	29.93	0
2022-11-02 00:28:50	-0.46	0	0.92	1.03	0.79	0.98	1.34	51.19	110.04	-176.95	24.95	32.18	29.93	0
2022-11-02 00:28:50	-0.46	0	0.92	1.03	0.79	0.67	1.4	49.97	110.04	-179.3	24.42	32.18	29.93	0
2022-11-02 00:28:50	-0.46	0	0.92	1.03	0.73	0.98	1.22	49.97	110.04	-178.13	24.42	32.18	29.94	0
2022-11-02 00:28:50	-0.46	0	0.92	1.03	0.67	0.92	1.28	49.97	111.26	-179.3	24.19	32.18	29.93	0

MySQL Database Result



DateTime	acceleratio...	acceleratio...	acceleratio...	accelerationNor...	GyroX	GyroY	GyroZ	MagnetoX	MagnetoY	MagnetoZ	magHorizDirecti...	Temperatu...	Pre
11/2/2022 12:28:47...	0	0	0	0	0	0	0	0	0	0	0	0	
11/2/2022 12:28:47...	-0.46	0	0.93	1.04	0.43	0.61	1.16	47.53	110.04	-178.13	23.36	32.18	
11/2/2022 12:28:48...	-0.46	0	0.92	1.03	0.49	0.85	1.04	46.31	110.04	-179.3	22.82	32.18	
11/2/2022 12:28:48...	-0.46	0	0.92	1.02	0.61	0.67	1.4	47.53	112.48	-178.13	22.91	32.19	
11/2/2022 12:28:48...	-0.46	0	0.93	1.04	0.73	0.92	1.59	49.97	108.82	-179.3	24.66	32.19	
11/2/2022 12:28:48...	-0.46	0	0.93	1.04	0.67	0.73	1.4	48.75	111.26	-179.3	23.66	32.18	
11/2/2022 12:28:48...	-0.46	0	0.92	1.02	0.67	0.98	1.46	48.75	112.48	-178.13	23.43	32.19	
11/2/2022 12:28:48...	-0.46	0	0.93	1.04	0.55	0.92	1.46	48.75	112.48	-180.47	23.43	32.18	
11/2/2022 12:28:48...	-0.46	0	0.93	1.04	0.67	0.73	1.34	45.09	110.04	-179.3	22.28	32.19	
11/2/2022 12:28:48...	-0.46	0	0.93	1.04	0.37	0.73	1.53	48.75	112.48	-178.13	23.43	32.18	
11/2/2022 12:28:49...	-0.46	0	0.93	1.03	0.55	0.92	1.34	48.75	112.48	-180.47	23.43	32.19	
11/2/2022 12:28:49...	-0.46	0	0.92	1.03	0.61	0.85	1.16	52.41	110.04	-178.13	25.47	32.18	
11/2/2022 12:28:49...	-0.46	0	0.93	1.04	0.67	1.04	1.34	52.41	110.04	-178.13	25.47	32.18	
11/2/2022 12:28:49...	-0.46	0	0.92	1.03	0.67	0.79	1.4	51.19	111.26	-178.13	24.71	32.19	
11/2/2022 12:28:49...	-0.46	0	0.92	1.03	0.67	0.79	1.4	47.53	112.48	-179.3	22.91	32.18	
11/2/2022 12:28:49...	-0.46	0	0.93	1.04	0.55	0.73	1.4	49.97	110.04	-178.13	24.42	32.18	
11/2/2022 12:28:49...	-0.46	0	0.93	1.03	0.79	0.92	1.46	47.53	108.82	-178.13	23.6	32.18	
11/2/2022 12:28:49...	-0.46	0	0.93	1.04	0.61	0.85	1.34	49.97	108.82	-181.64	24.66	32.18	
11/2/2022 12:28:50...	-0.46	0	0.92	1.03	0.61	0.73	1.65	51.19	110.04	-176.95	24.95	32.18	
11/2/2022 12:28:50...	-0.46	0	0.92	1.03	0.79	0.98	1.34	51.19	110.04	-176.95	24.95	32.18	

Delphi ini File

```
procedure TForm1.callini;
770 begin
  iniFilePath:=ExtractFilePath(ParamStr(0)) + 'cfg.ini';  // application exe
  if not FileExists(iniFilePath) then
    WriteIniCfg   // Set Object Config as Default
  else
    ReadIniCfg;
end;
```

```
1130 procedure TForm1.reCheck;
  begin
  if CheckBox1.IsChecked then
  begin
    StartSQL.Enabled:=False;
    StopSQL.Enabled:=False;
    ShowMessage('Start With SQL');
  end
  else
  begin
1140  StartSQL.Enabled:=True;
    StopSQL.Enabled:=False;
  end;
  end;
```

```
procedure TForm1.ReadIniCfg;
var
  vIni:TIniFile;
begin
  vIni:=TIniFile.Create(iniFilePath);
  try
   CheckBox1.IsChecked:=vIni.ReadBool(CheckBox1.ClassName, 'CheckBox1.Checked', False);
  finally
1110 vIni.Free;
  end;
end;
```

```
procedure TForm1.WriteIniCfg;
var
  vIni:TIniFile;
begin
  vIni:=TIniFile.Create(iniFilePath);
  try
   vIni.WriteBool(CheckBox1.ClassName, 'CheckBox1.Checked', CheckBox1.IsChecked);
540 finally
   vIni.Free;
  end;
end;
```

RecordIMU	11/21/2022 10:58 AM	File folder	
cfg	11/21/2022 12:21 PM	Configuration sett...	1 KB
FMX_IMU_TCP	11/21/2022 12:20 PM	Application	63,521 KB
Main.dcu	11/21/2022 12:10 PM	DCU File	54 KB
Project1	11/19/2022 11:02 AM	Application	63,437 KB
Unit1.dcu	11/19/2022 10:48 AM	DCU File	40 KB

Delphi Data Logging

```
type
TrecData = class(TThread)
  private
  protected
  procedure Execute; override;
end;
```

```
function TForm1.FloatToStr2(Data: Single): String;
begin
  Result:= Format('%0.2f', [Data]);
end;
```

```
procedure TForm1.recDataStart;
begin
if MyRec = nil then
  MyRec := TrecData.Create(False);
end;

procedure TForm1.recDataStop;
begin
if MyRec <> nil then
begin
  Myrec.Free;
  Myrec.Terminate;
  Myrec := nil;
end;
end;
```

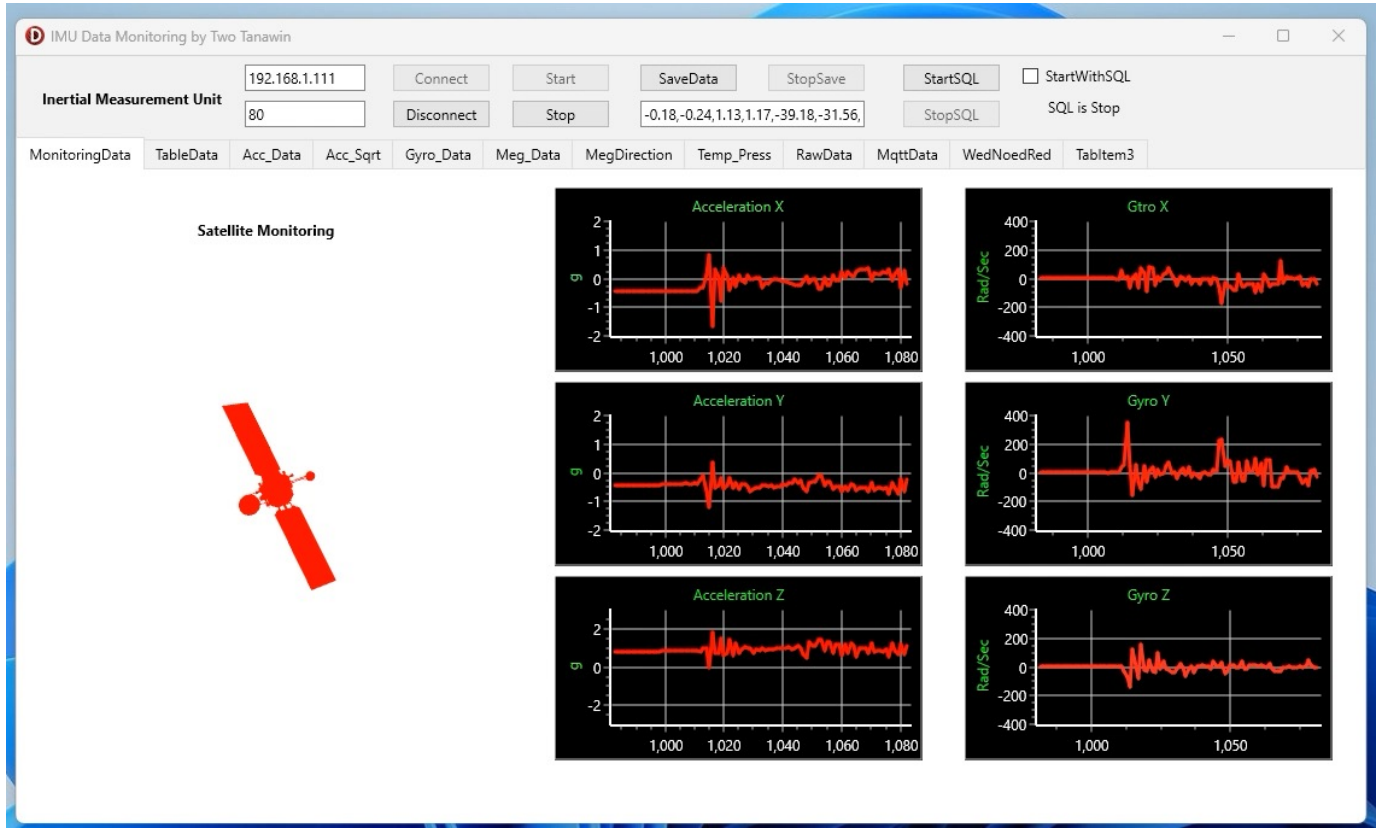
```
procedure TForm1.SaveDataClick(Sender: TObject);
var
 Buff:String;
 I,x:integer;
 S:String;
begin
if Start.Enabled=True then
begin
 ShowMessage('Please Enable Start');
end
else
begin
recDataStart;
SaveData.Enabled:=False;
StopSave.Enabled:=True;
end;
end;
```

```
procedure TForm1.StopSaveClick(Sender: TObject);
begin
recDataStop;
StopSave.Enabled:=False;
SaveData.Enabled:=True;
end;
```

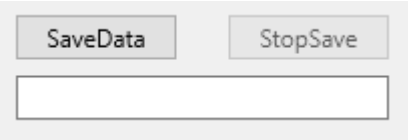
```
procedure TrecData.Execute;
var
 Buff:String;
 I,x:integer;
begin
  inherited;
  i:=0;
   AssignFile(Form1.MyDataFile,'RecordIMU\'+FormatDateTime('hh_nn_ss', Now)+Form1.MyDataPath); {Assigns the Filename}
   ReWrite(Form1.MyDataFile); {Create a new file }
 Buff:='No, Time, AccX, AccY, AccZ, AccNorm, GyroY, GyroZ';
 Writeln(Form1.MyDataFile, Buff);
  repeat
    Application.ProcessMessages;
    Inc(i);
    Sleep(100);
    Form1.MyData.accX:=StrToFloat(Form1.accData.aX);
    Form1.MyData.accY:=StrToFloat(Form1.accData.aY);
    Form1.MyData.accZ:=StrToFloat(Form1.accData.aZ);
    Form1.MyData.accNorm:=StrToFloat(Form1.accData.rq);
    Form1.MyData.gyroR:=StrToFloat(Form1.gyroData.gyroX);
    Form1.MyData.gyroP:=StrToFloat(Form1.gyroData.gyroY);
    Form1.MyData.gyroY:=StrToFloat(Form1.gyroData.gyroZ);
    Buff := IntToStr(i)+','+FormatDateTime('dd-mm-yyyy hh:nn:ss', Now)+','+Form1.FloatToStr2(Form1.MyImu[i].accX)+','
    +Form1.FloatToStr2(Form1.MyImu[i].accY)+','+Form1.FloatToStr2(Form1.MyImu[i].accy)+','+Form1.FloatToStr2(Form1.MyImu[i].accNorm)
    +','+Form1.FloatToStr2(Form1.MyImu[i].gyroR)+','+Form1.FloatToStr2(Form1.MyImu[i].gyroP)+','+Form1.FloatToStr2(Form1.MyImu[i].gyroY);
    Writeln(Form1.MyDataFile, Buff);
  until Terminated;
  Closefile(Form1.MyDataFile); {Closes file }  //save
end;
```

KMUTNB INVENTION TO INNOVATION

Delphi Data Logging



Delphi Data Logging



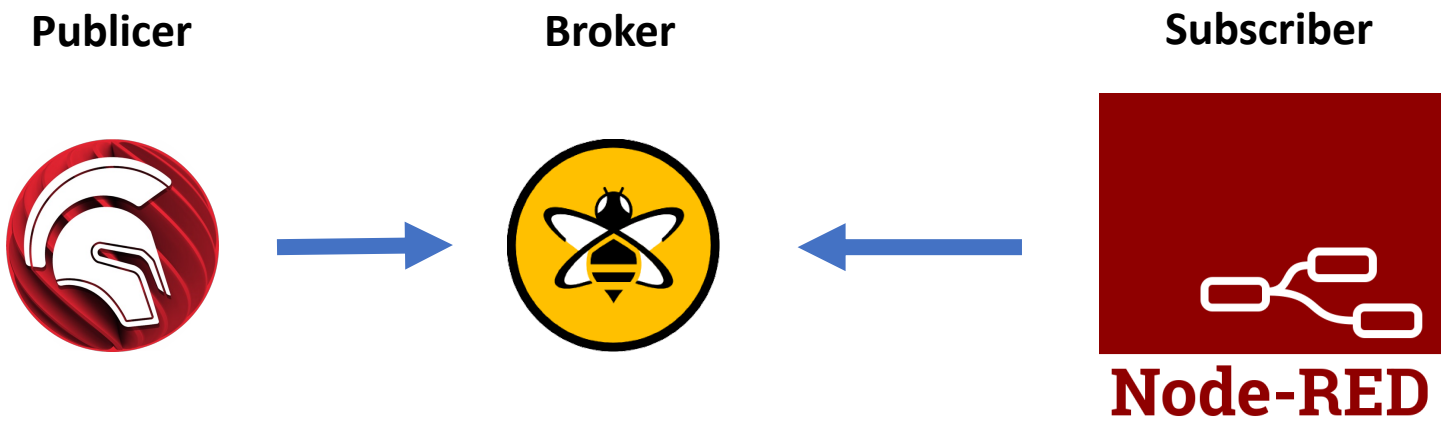
RecordIMU	11/21/2022 10:58 AM	File folder	
cfg	11/21/2022 12:21 PM	Configuration sett...	1 KB
FMX_IMU_TCP	11/21/2022 12:20 PM	Application	63,521 KB
Main.dcu	11/21/2022 12:10 PM	DCU File	54 KB
Project1	11/19/2022 11:02 AM	Application	63,437 KB
Unit1.dcu	11/19/2022 10:48 AM	DCU File	40 KB

10_58_03MyFile - Notepad

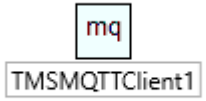
File Edit View

```
No, Time, AccX, AccY, AccZ, AccNorm, GyroY, GyroZ
1,21-11-2022 10:58:03,0.22,-0.38,-0.38,0.89,-50.66,76.29,-79.53
2,21-11-2022 10:58:03,-0.11,-0.52,-0.52,0.99,262.21,-12.15,35.64
3,21-11-2022 10:58:03,-0.28,1.00,1.00,1.68,-224.85,74.58,-151.37
4,21-11-2022 10:58:03,0.20,-1.46,-1.46,1.70,54.75,-54.69,58.59
5,21-11-2022 10:58:03,-2.51,1.85,1.85,4.56,267.88,52.12,45.23
6,21-11-2022 10:58:03,0.34,-0.80,-0.80,1.20,-346.92,-40.53,-108.03
7,21-11-2022 10:58:03,0.35,-0.26,-0.26,0.87,-180.85,88.01,-287.48
8,21-11-2022 10:58:04,0.22,-0.84,-0.84,1.03,259.95,-121.70,216.37
9,21-11-2022 10:58:04,-0.03,0.65,0.65,1.86,-432.19,-196.47,-163.94
10,21-11-2022 10:58:04,0.99,-1.16,-1.16,1.59,46.94,-132.81,196.90
11,21-11-2022 10:58:04,-1.41,1.73,1.73,3.43,-156.43,-208.01,-48.95
12,21-11-2022 10:58:04,0.46,-1.79,-1.79,1.93,-95.70,-62.38,37.41
13,21-11-2022 10:58:04,0.97,-0.96,-0.96,1.45,77.70,-33.14,134.70
14,21-11-2022 10:58:04,0.94,-0.33,-0.33,1.28,-187.19,203.31,-180.97
15,21-11-2022 10:58:04,-0.88,1.02,1.02,2.50,361.57,152.47,78.37
16,21-11-2022 10:58:04,0.40,-0.07,-0.07,0.86,-194.40,48.58,-211.98
17,21-11-2022 10:58:04,0.66,-0.22,-0.22,1.09,295.23,16.05,231.08
18,21-11-2022 10:58:05,0.32,0.00,0.00,1.41,-234.31,-14.53,-188.23
19,21-11-2022 10:58:05,0.63,-2.03,-2.03,2.16,49.74,-130.98,126.04
20,21-11-2022 10:58:05,-1.27,1.52,1.52,3.17,-112.55,-144.47,-35.95
21,21-11-2022 10:58:05,0.55,-1.51,-1.51,1.74,-246.28,17.21,-68.79
22,21-11-2022 10:58:05,0.13,0.34,0.34,1.66,361.57,203.61,132.20
23,21-11-2022 10:58:05,0.23,-0.15,-0.15,1.21,-253.11,35.46,-237.79
24,21-11-2022 10:58:05,0.68,-0.13,-0.13,0.88,-91.61,156.80,-224.67
25,21-11-2022 10:58:05,-0.26,1.59,1.59,2.81,-429.50,-110.60,-33.14
26,21-11-2022 10:58:05,-0.14,-0.49,-0.49,1.40,-475.83,-145.63,-239.14
27,21-11-2022 10:58:05,1.35,-1.06,-1.06,1.80,-73.91,210.33,-326.54
28,21-11-2022 10:58:06,0.49,-0.11,-0.11,1.21,119.63,-146.97,277.34
29,21-11-2022 10:58:06,0.58,-1.11,-1.11,1.45,-368.16,16.24,-163.88
```

Datavisualization Using Node-RED



Delphi Conect MQTT




```
430  const
       ChatChannel = 'sensor1';
```

```
procedure TForm1.CallMqtt;
begin
  TMSMQTTClient1.BrokerHostName := 'broker.mqttdashboard.com';
  TMSMQTTClient1.Connect();
  memo2.Lines.Add('Connect');
end;
```

```
procedure TForm1.RunMqtt;
begin
  TMSMQTTClient1.Publish(ChatChannel, TMSMQTTClient1.ClientID+'!'+ memo2.Lines.Text);
  memo2.Lines.Clear;
1170 end;
```

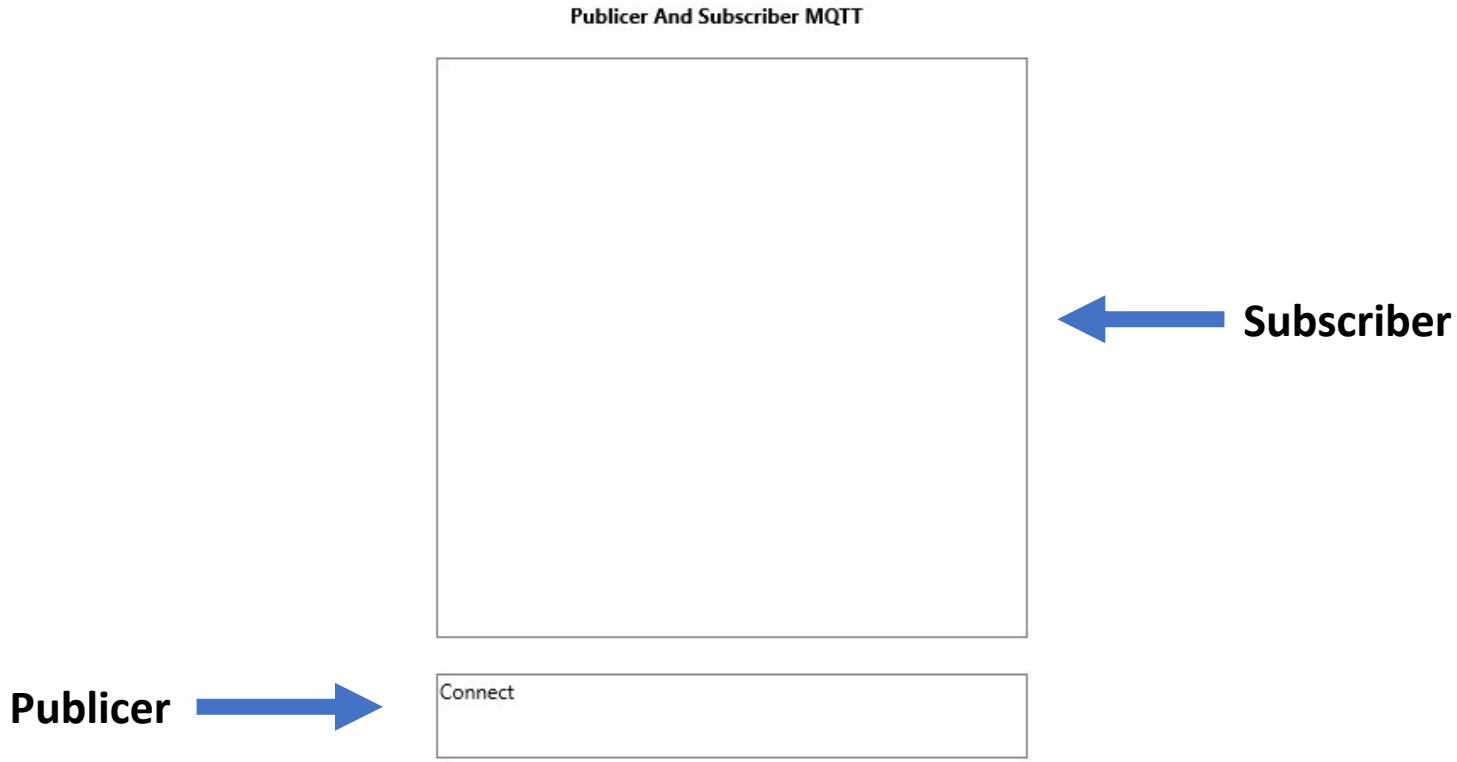
Delphi Connect MQTT

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
Application.ProcessMessages;
RunMqtt;
Memo2.Lines.Add(','+accData.aX+','+accData.aY+','+accData.aZ+','+gyroData.gyroX+','+gyroData.gyroY+','+gyroData.gyroZ);
end;
```

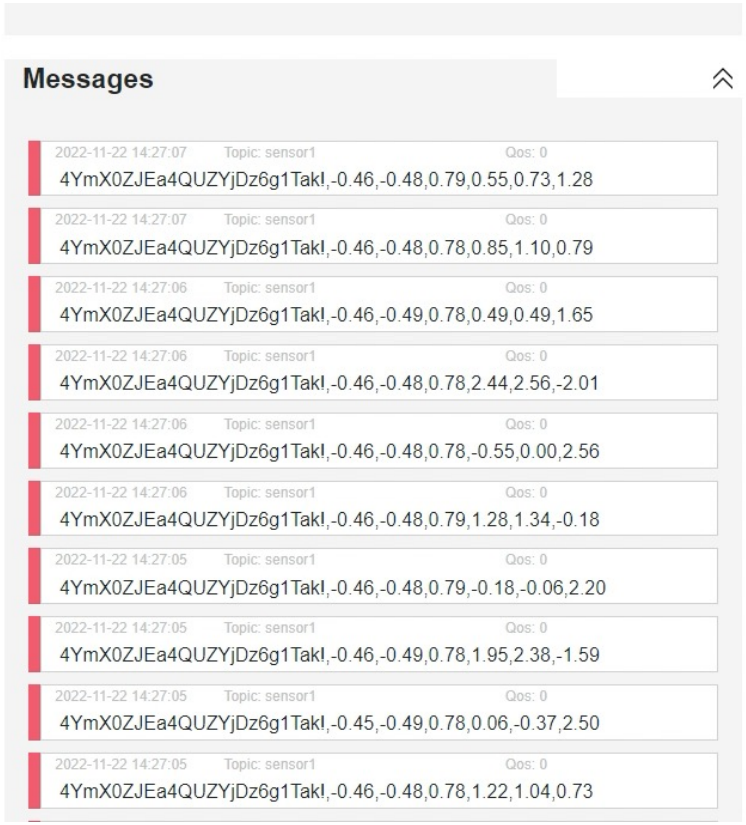
```
procedure TForm1.TMSMQTTClient1ConnectedStatusChanged(ASender: TObject;
  const AConnected: Boolean; AStatus: TTMSMQTTConnectionStatus);
begin
  if AConnected then
  begin
    TMSMQTTClient1.Subscribe(ChatChannel);
  end
  else begin
    case AStatus of
      csConnectionRejected_InvalidProtocolVersion,
      csConnectionRejected_InvalidIdentifier,
      csConnectionRejected_ServerUnavailable,
      csConnectionRejected_InvalidCredentials,
      csConnectionRejected_ClientNotAuthorized:; // the connection is rejected by broker
      csConnectionLost:; // the connection with the broker is lost
      csConnecting:; // The client is trying to connect to the broker
      csReconnecting:; // The client is trying to reconnect to the broker
    end;
  end;

end;
```

```
procedure TForm1.TMSMQTTClient1PublishReceived(ASender: TObject;
  APacketID: Word; ATopic: string; APayload: TArray<System.Byte>)
var
  msg,orig: string;
  vp: integer;
  alright: boolean;
begin
  msg := TEncoding.UTF8.GetString(APayload);

  vp := pos('!', msg);

  if vp > 0 then
  begin
    orig := copy(msg,1,vp-1);
    alright := orig <> TMSMQTTClient1.ClientID;

    msg := copy(msg, vp + 1, Length(msg));
    AddMessage(msg, alright);
  end;
end;
```

Delphi Connect MQTT



Delphi Connect MQTT

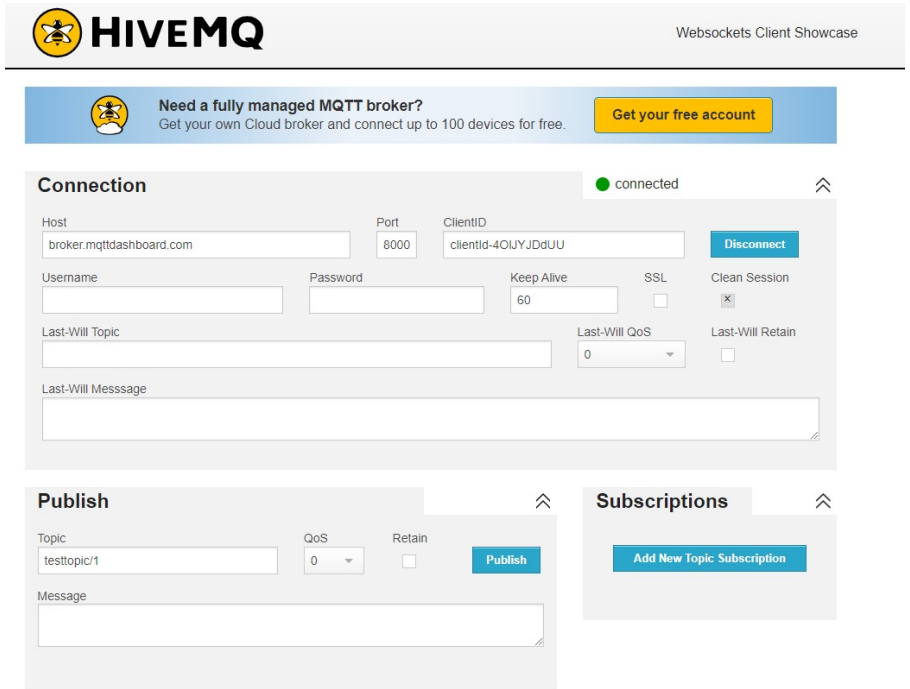


Form Creat & Close

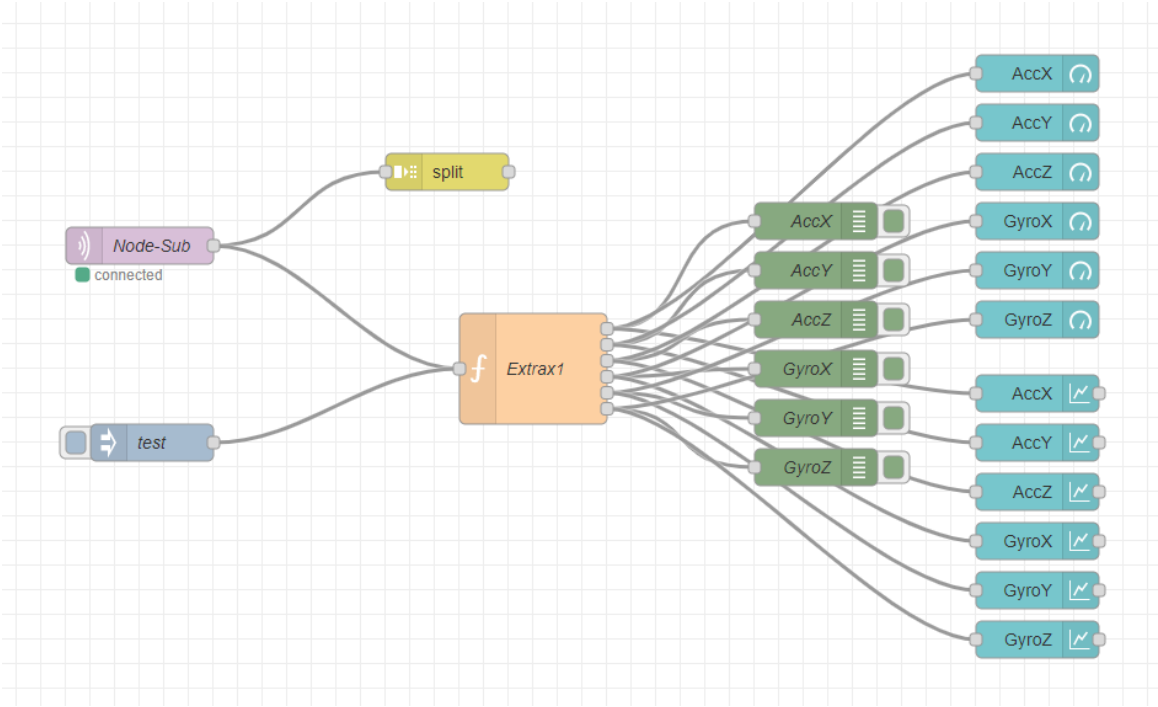
```
procedure TForm1.FormCreate(Sender: TObject);
var
  I:Integer;
begin
IdTCPClient1.Host := Edit1.Text;
IdTCPClient1.Port := StrToInt(Edit2.Text);
PreModel;
SetZero;
ThreadPlotStart;
callMqtt;
Thread3dStart;
MyDataPath:=ExtractFilePath(Application.Name)+'MyFile.txt';
Disconnect.Enabled:=False;
Stop.Enabled:=False;
StopSQL.Enabled:=False;
StopSave.Enabled:=False;
callini;
reCheck;
end;
```

```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
ThreadModelStop;
EndThreadSQL;
EndThreadPlot;
recDataStop;
EndThread3d;
StopClient;
Timer1.Enabled:=False;
Application.Terminate;
end;
```

Hive MQTT

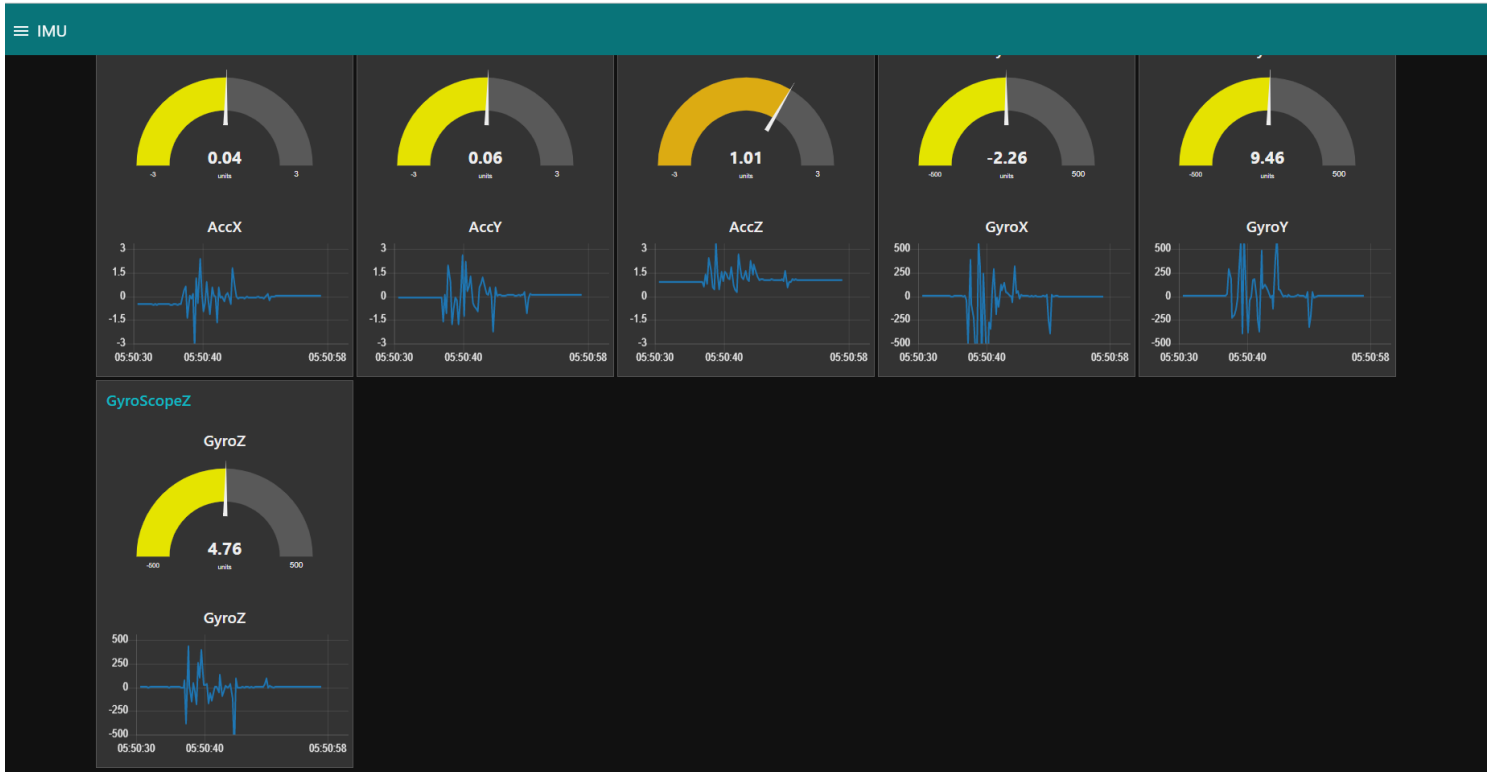

HIVEMQ
Websockets Client Showcase
Need a fully managed MQTT broker?
Get your own Cloud broker and connect up to 100 devices for free.
Get your free account
Connection
connected
Host
broker.mqttdashboard.com
Port
8000
ClientID
clientId-4OlJYJDdUU
Disconnect
Username
Password
Keep Alive
60
SSL
Clean Session
Last-Will Topic
Last-Will QoS
0
Last-Will Retain
Last-Will Messsage
Publish
Topic
testtopic/1
QoS
0
Retain
Publish
Message
Subscriptions
Add New Topic Subscription

Node-Red Extrax Data

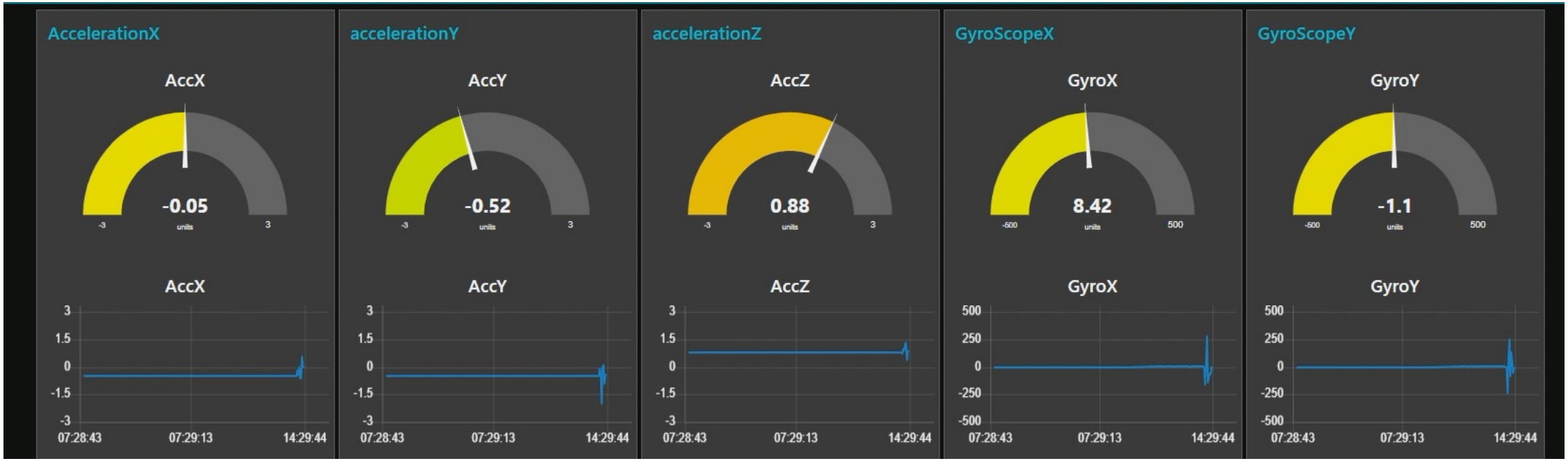



```
var outputMsgs = [];
var words = msg.payload.split(",");
// for (var w in words) {
//     outputMsgs.push({ payload: words[w] });
//  }
var text = { payload: words[0] };
var accx = { payload: words[1] };
var accy = { payload: words[2] };
var accz = { payload: words[3] };
var gyrox = { payload: words[4] };
var gyroy = { payload: words[5] };
var gyroz = { payload: words[6] };
// var out7 = { payload: words[7] };
return [accx, accy, accz, gyrox, gyroy, gyroz];
```

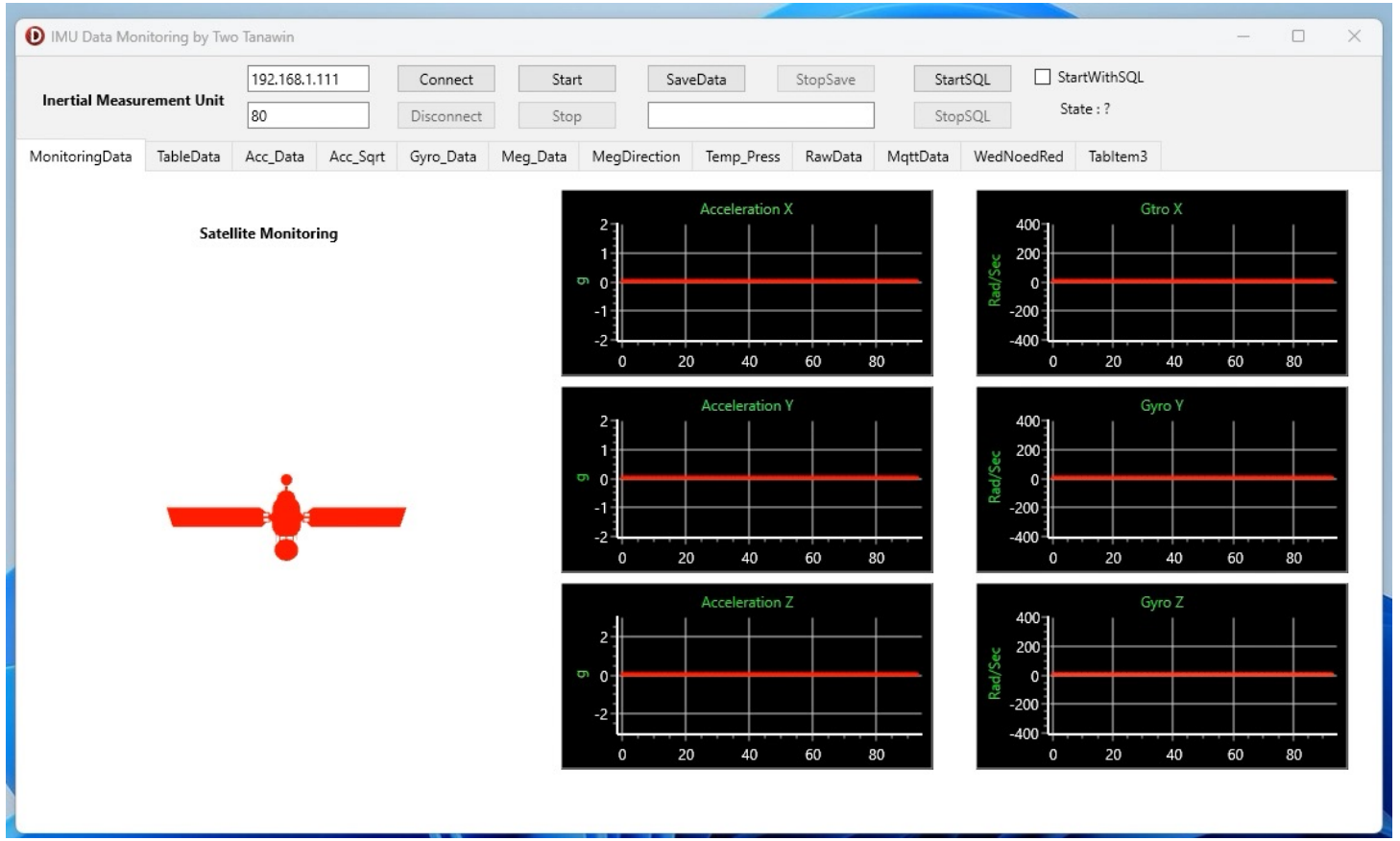
Datavisualization Using Node-RED



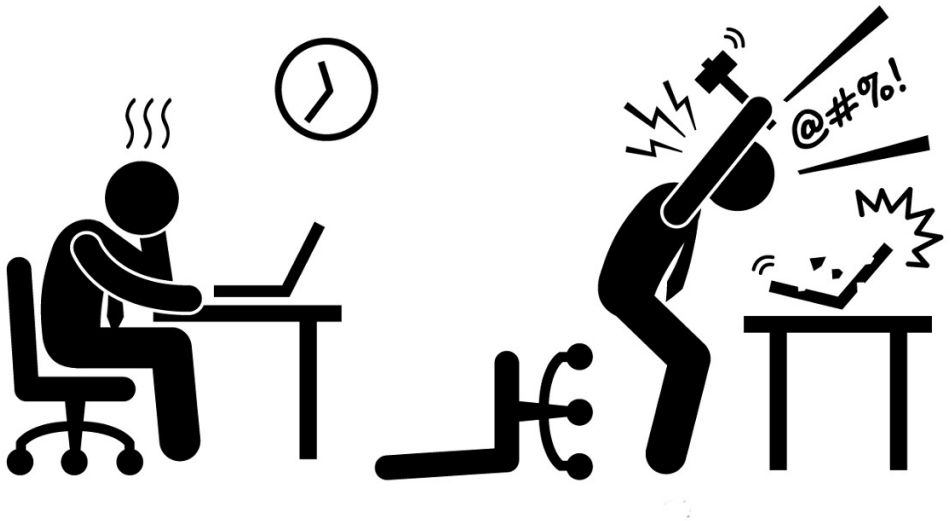
Datavisualization Using Node-RED



IoT Inertial Mesuament Unit Monitoring



Problem



Applied

Delploy Other Device

Monitoring Real Machine

THE END THX