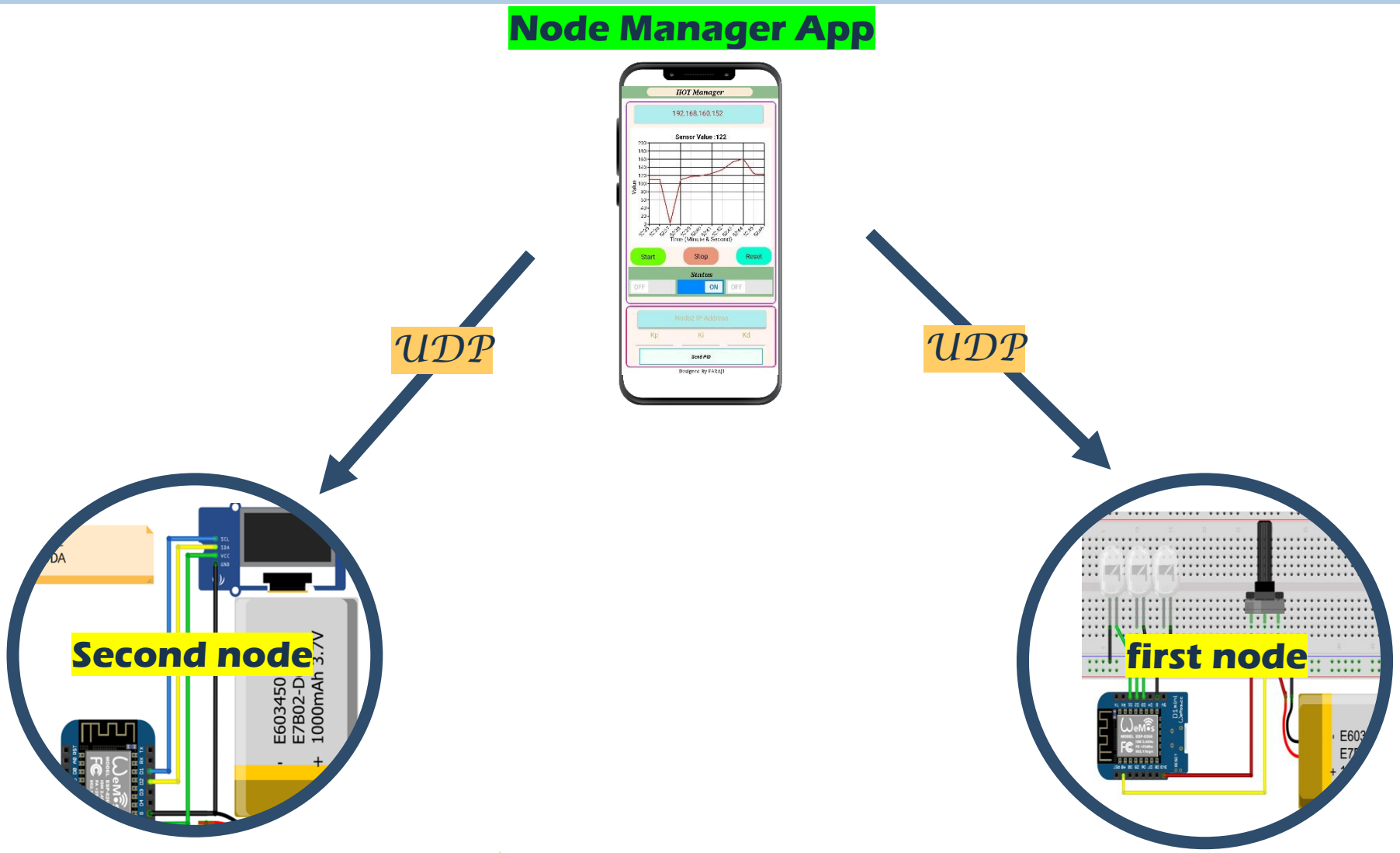
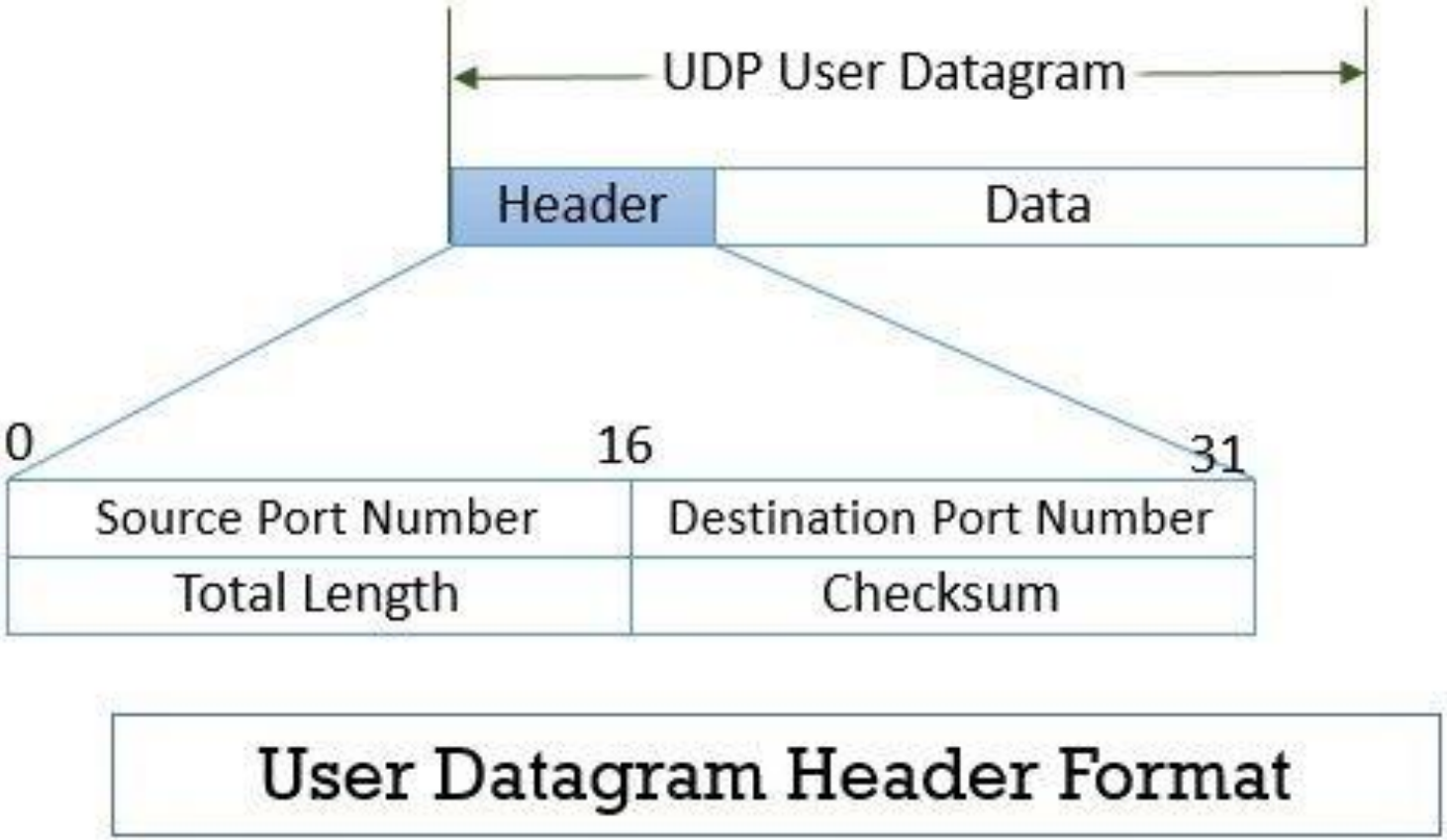
IIOT NODE Manager in Industrial Application

Designed By Faraji

❑ Topology

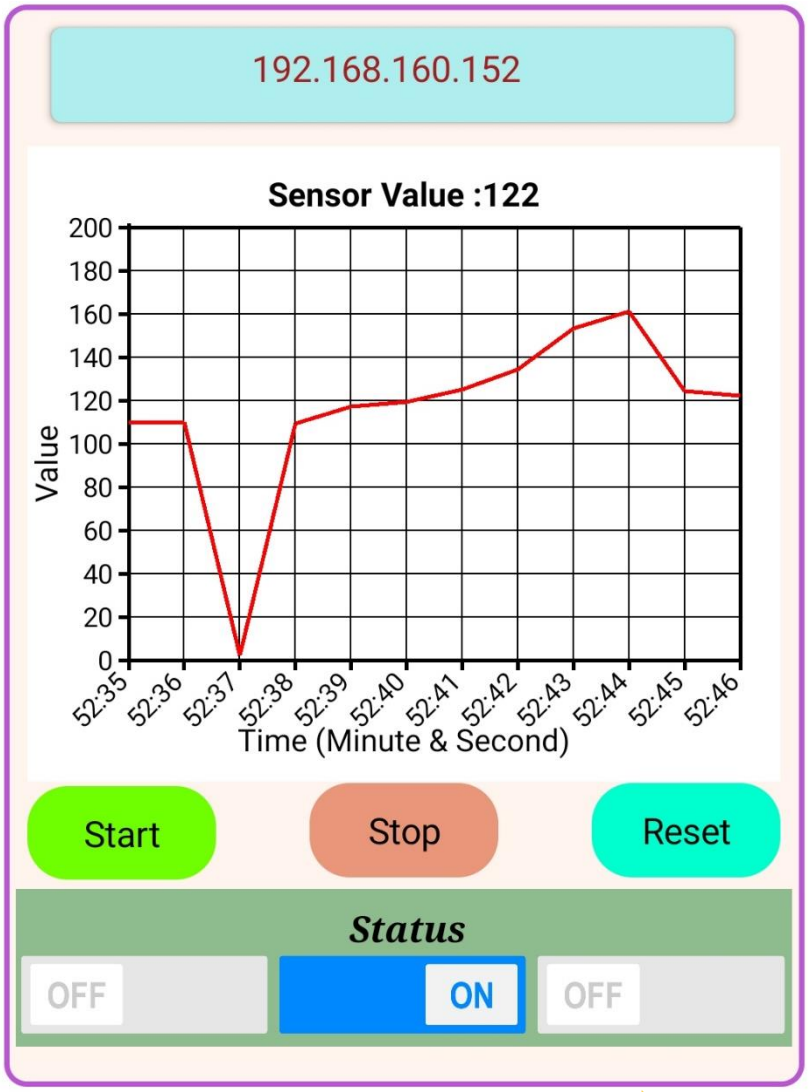

Node Manager App
IOT Manager
192.168.160.152
Sensor Value :122
Start
Stop
Reset
Status
ON
UDP
UDP
Second node
first node

❑ UDP Protocol



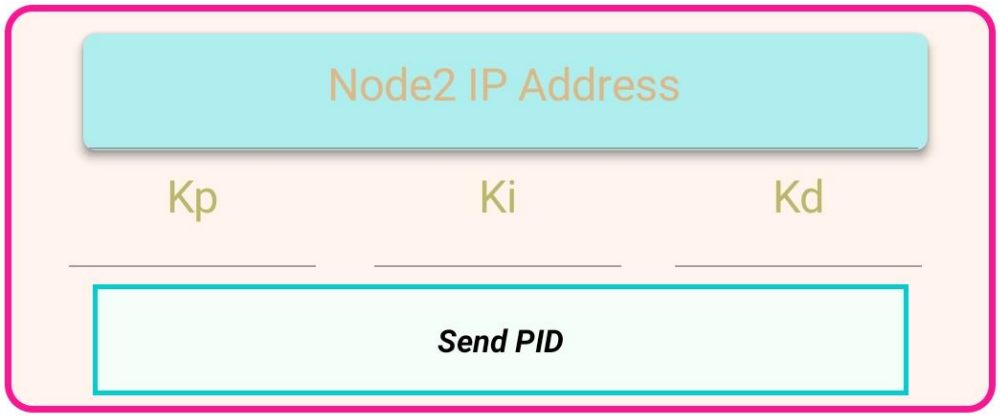
User Datagram Header Format

App ❑



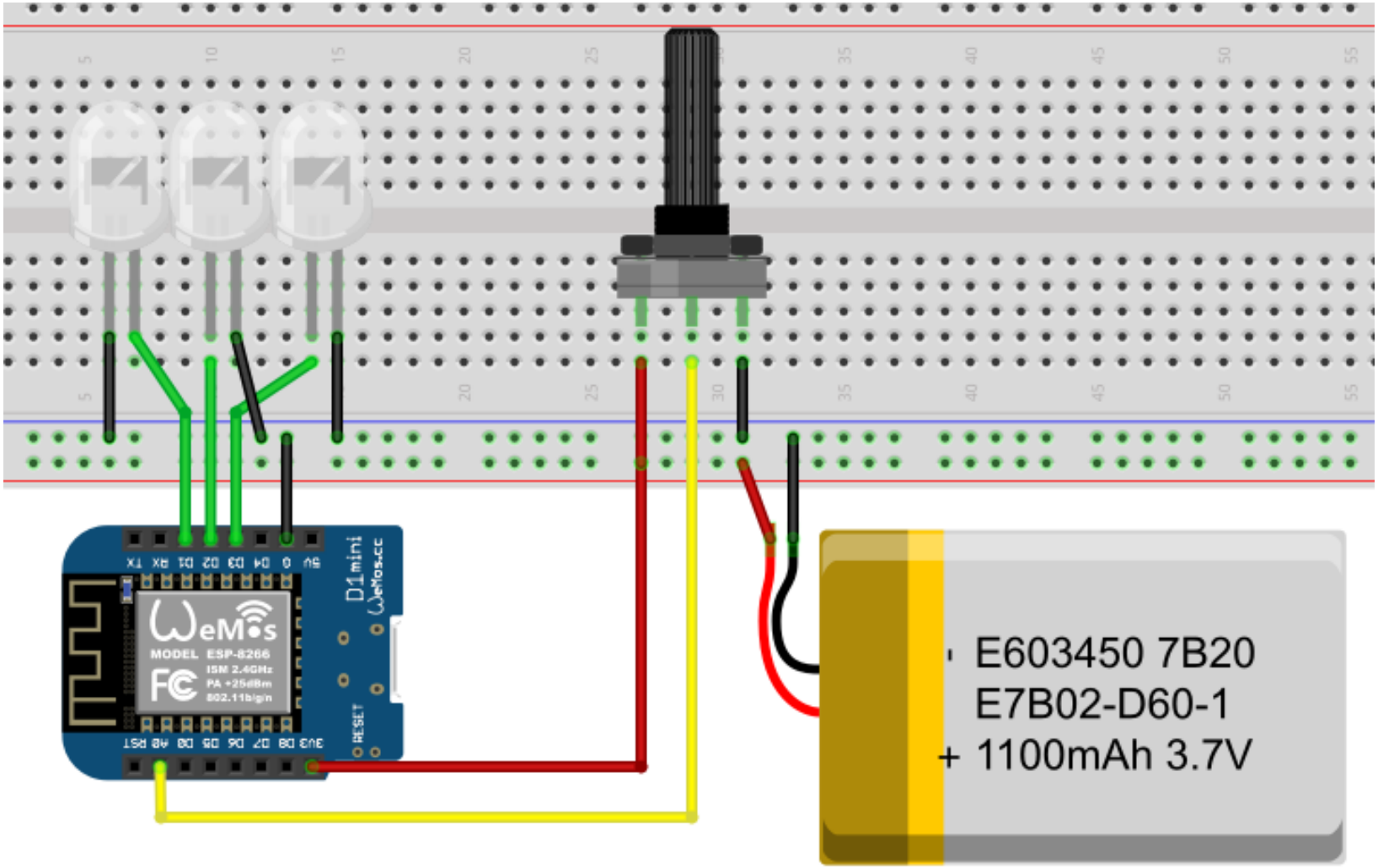
first node

Second node

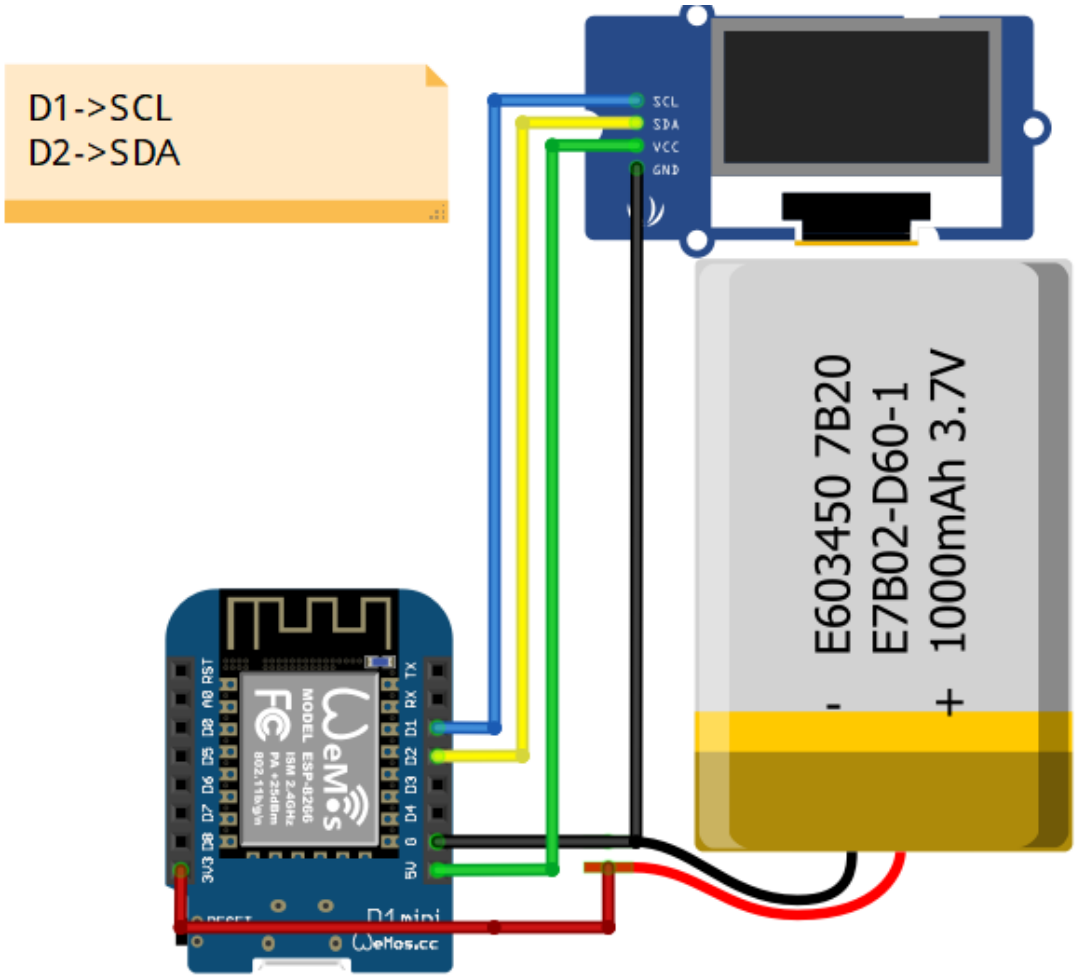


Designed By FARAII

❑ Node 1


D1 mini
WeMos.cc
WeMos
MODEL ESP-8266
ISM 2.4GHz
PA +25dBm
802.11b/g/n
RESET
E603450 7B20
E7B02-D60-1
+ 1100mAh 3.7V

❑ Node 2


D1->SCL
D2->SDA
SCL
SDA
VCC
GND
E603450 7B20
E7B02-D60-1
- +
1000mAh 3.7V
D1 mini
WeMos.cc

❑ Node 1 Arduino Code

❖ Define Variables

```
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>

WiFiUDP Udp;
unsigned int UdpPort = 4210;  // local port to listen on
char incomingPacket[255];  // buffer for incoming packets
int sensorValue = 0;
const int analogInPin = A0;
char buffer[40];

String strCon;
```

❑ Node 1 Arduino Code

```
void setup()
{
  pinMode(0, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

  Serial.begin(115200);
  Serial.println();

  WiFi.begin("SM_FARAJI", "seyed1*4=

  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.print("Connected, IP address: ");
  Serial.println(WiFi.localIP());
  Udp.begin(UdpPort);
 }
```

Set-Up Pins ❖
Connection with Central Node ❖

❑ Node 1 Arduino Code

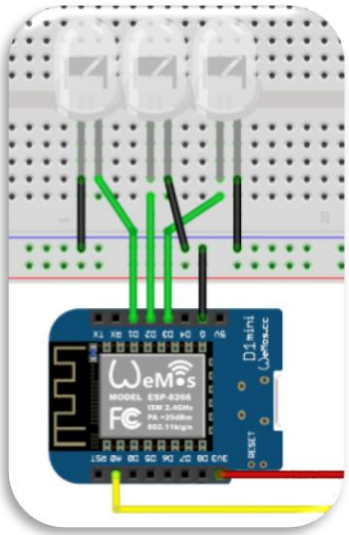
- Receive Commands from App via UDP protocol

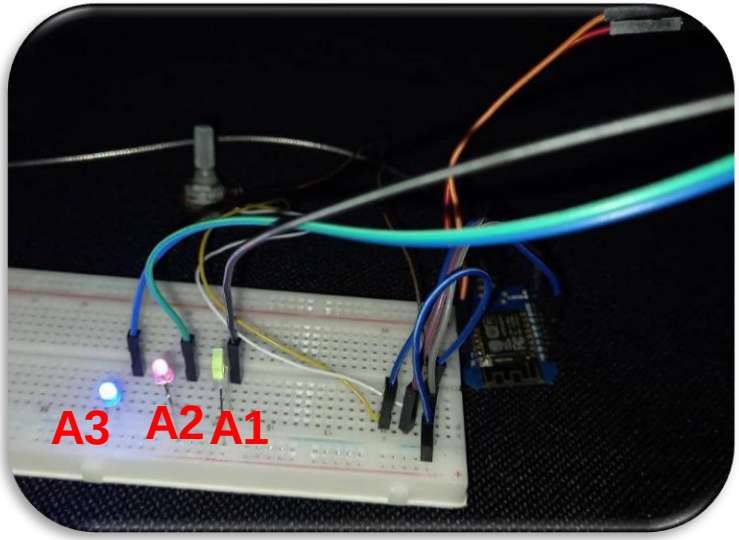
```
void loop()
{
  int packetSize = Udp.parsePacket();
  if (packetSize)
  {
    int len = Udp.read(incomingPacket, 255);
    if (len > 0)
    {
      incomingPacket[len] = 0;
    }
    Serial.println(incomingPacket);
    strCon = incomingPacket;
```

❑ Node 1 Arduino Code

- Execute Commands Based on Received Message

```
  if (strCon=="A1On") {
    digitalWrite(5, HIGH);
  }
  else if (strCon=="A1Off") {
    digitalWrite(5, LOW);
  }
  if (strCon=="A2On") {
    digitalWrite(4,HIGH);
  }
  else if (strCon=="A2Off") {
    digitalWrite(4,LOW);
  }
  if (strCon=="A3On") {
    digitalWrite(0, HIGH);
  }
  else if (strCon=="A3Off") {
    digitalWrite(0, LOW);
  }
}
```

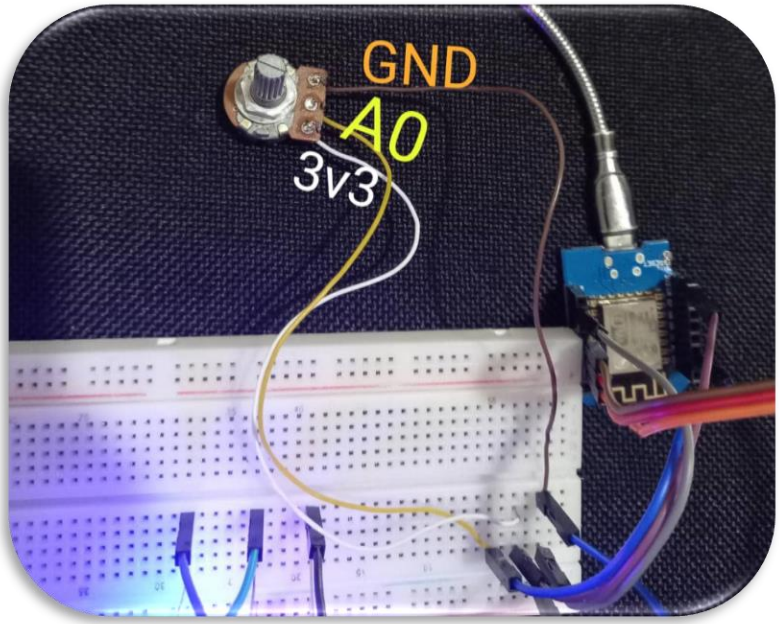




❑ Node 1 Arduino Code

❖ Send Sensor Data to App via UDP

```
    sensorValue = analogRead(analogInPin);
    sensorValue = map(sensorValue, 0, 1023, 0, 200);
    sprintf(buffer, "%d", sensorValue); //Value:
    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
    Udp.write(buffer);
    Udp.endPacket();
    Serial.println(buffer);

  delay(400);
}
```

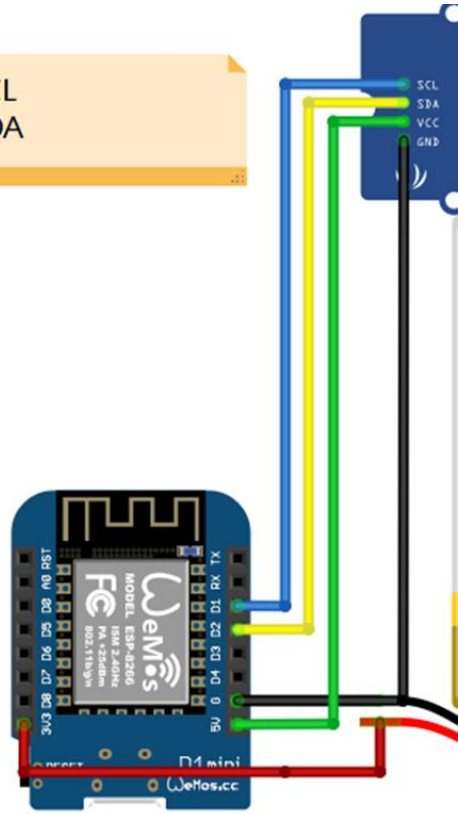


❑ Node 2 Arduino Code

❖ Define Variables

```
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>

//////////////////////////////////OLED/////////////////////////////////////
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
//////////////////////////////////////////////////////////////////////////

WiFiUDP Udp;
unsigned int UdpPort = 4210;  // local port to listen on
char incomingPacket[255];  // buffer for incoming packets
int sensorValue = 0;
const int analogInPin = A0;
char buffer[40];

String strCon;
```

D1->SCL
D2->SDA



❑ Node 2 Arduino Code

❖ Define a Function for getting numbers from strings data

B4A:

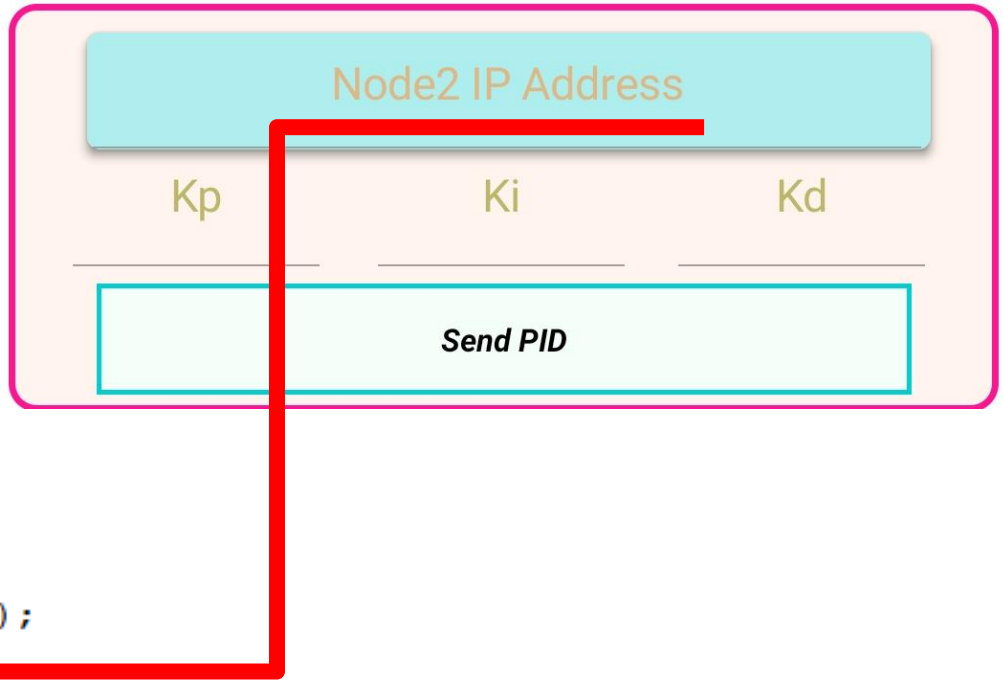
```
PID = EditTextKp.Text & "S" & EditTextKi.Text & "S" & EditTextKd.Text
UdpP.Initialize(PID.GetBytes("ASCII"), EditTextDestinationIP2.Text, 4210)
```

```
String getValue(String data, char separator, int index)

int found = 0;
int strIndex[] = { 0, -1 };
int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
        found++;
        strIndex[0] = strIndex[1] + 1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
}
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
```

❑ Node 2 Arduino Code

- Initial Setup For OLED
- Connection Setup for second node

```
void setup()
{
  Serial.begin(115200);
  Serial.println();

  WiFi.begin("SM_FARAJI", "1*4=1400");
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.print("Connected, IP address: ");
  Serial.println(WiFi.localIP());
  Udp.begin(UdpPort);
```



❑ Node 2 Arduino Code

❖ Initial Setup For OLED

```
//////////////////////////Setup OLED//////////
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(1000);
  display.clearDisplay();

  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 10);
  // Display static text
  display.println("PID TUNER");
  display.display();
}
```

❑ Node 2 Arduino Code

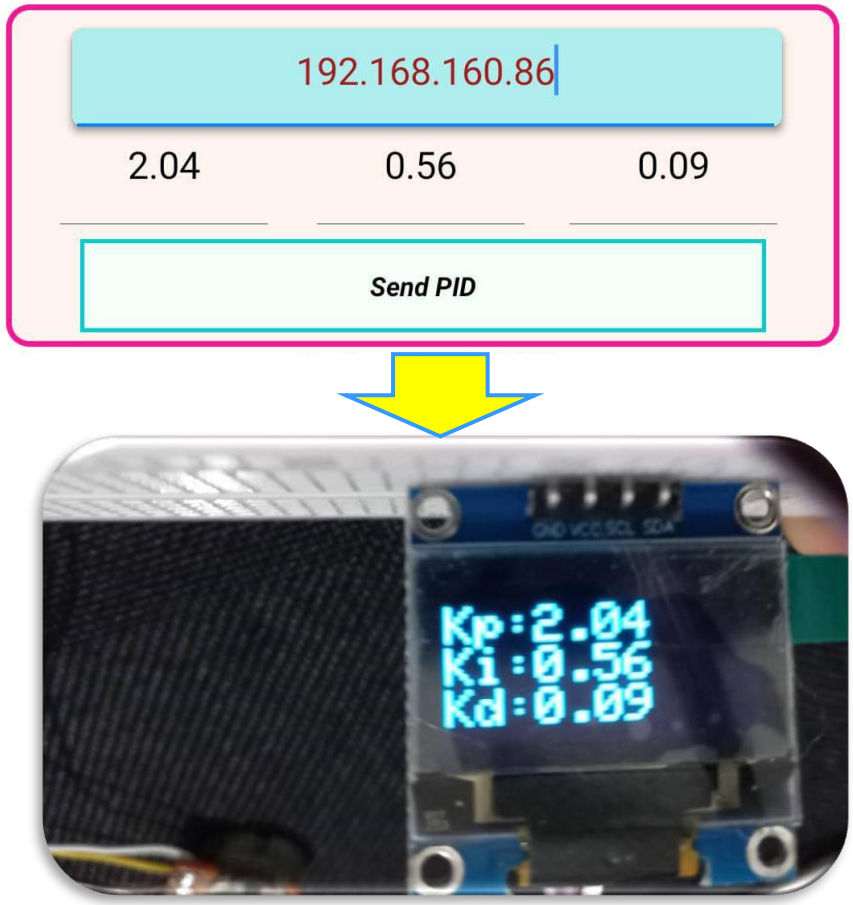
Receiving PID Setup And Display in OLED ❖

```
void loop()
{
  int packetSize = Udp.parsePacket();
  if (packetSize)
  {
    int len = Udp.read(incomingPacket, 255);
    if (len > 0)
    {
      incomingPacket[len] = 0;
    }
    //Serial.println(incomingPacket);
    strCon = incomingPacket;
    String Kp = getValue(strCon, 'S', 0);
    String Ki = getValue(strCon, 'S', 1);
    String Kd = getValue(strCon, 'S', 2);

    Serial.println("Kp:" +Kp);
    Serial.println("Ki:" + Ki);
    Serial.println("Kd:" + Kd);
    Serial.println("");
```

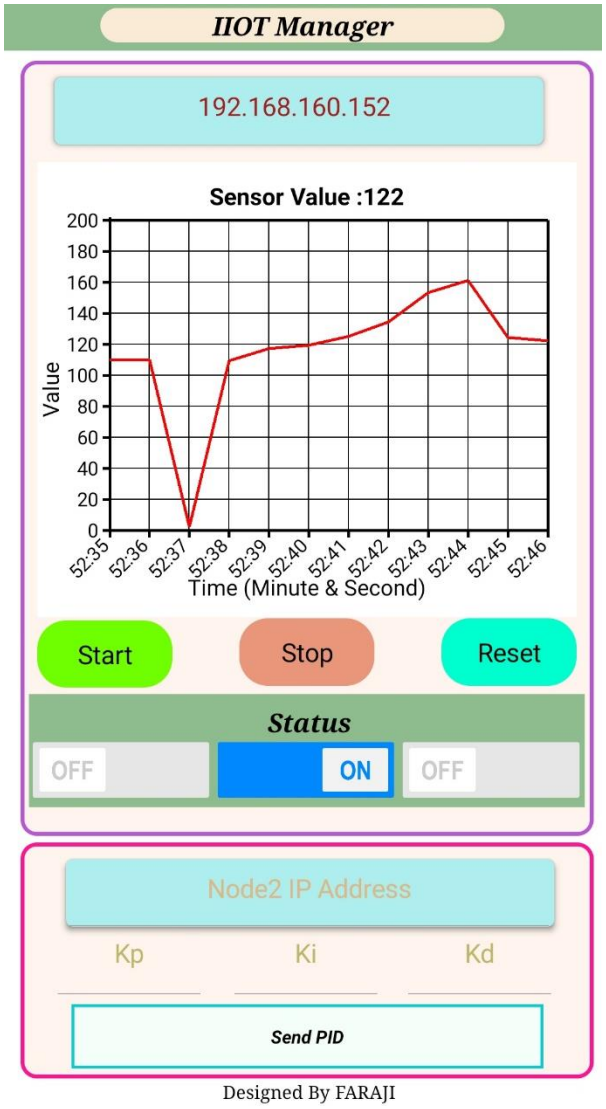
❑ Node 2 Arduino Code

❖ Receiving PID Setup And Display in OLED

```
    display.clearDisplay();

    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.setCursor(0, 10);
    // Display static text
    display.println("Kp:"+Kp);
    display.println("Ki:"+Ki);
    display.println("Kd:"+Kd);
    display.display();

   delay(400);
}
```



❑ B4A Coding

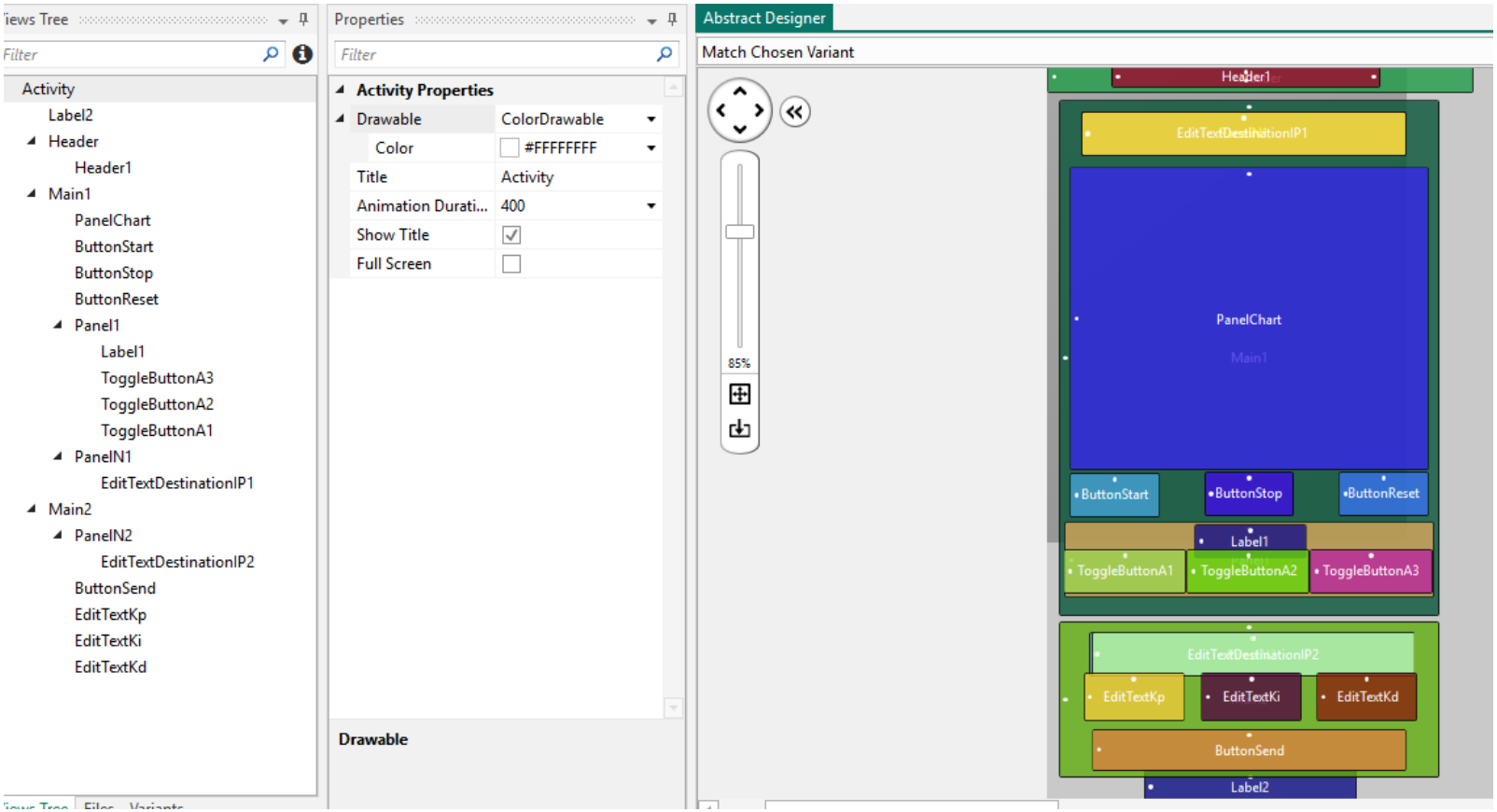


Designed By FARAJI

- Explain B4A code

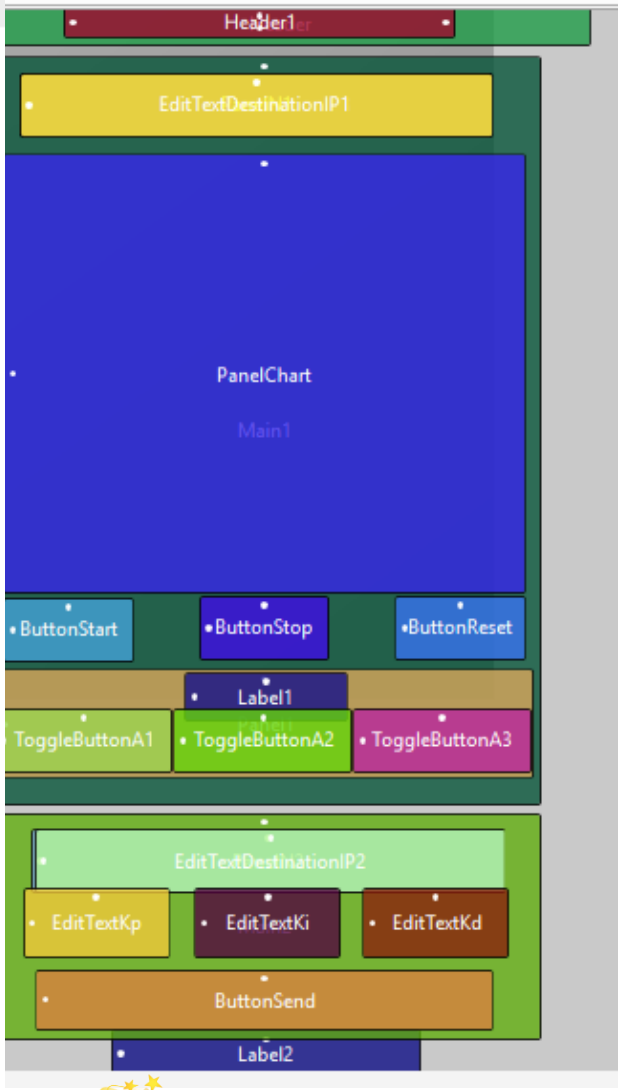
❑ B4A Coding

B4A Designer ❖



❑ B4A Coding

- Activity
  - Label2
  - Header
    - Header1
  - Main1
    - PanelChart
    - ButtonStart
    - ButtonStop
    - ButtonReset
    - Panel1
      - Label1
      - ToggleButtonA3
      - ToggleButtonA2
      - ToggleButtonA1
    - PanelN1
      - EditTextDestinationIP1
  - Main2
    - PanelN2
      - EditTextDestinationIP2
    - ButtonSend
    - EditTextKp
    - EditTextKi
    - EditTextKd



B4A Designer Parts ❖

❑ B4A Coding

Main Part of Code ❖

Main | Charts

Globals

```
1  #Region  Project Attributes
2      #ApplicationLabel: IIOT App
3      #VersionCode: 1
4      #VersionName:
5      'SupportedOrientations possible values: unspecified, landscape or portrait.
6      #SupportedOrientations: unspecified
7      #CanInstallToExternalStorage: False
8  #End Region
9
10 #Region  Activity Attributes
11     #FullScreen: True
12     #IncludeTitle: False
13 #End Region
14
15 Sub Process_Globals
16     'for Graph
17     Private Timer1 As Timer
18
19     'for udp connection
20     Public UdpS As UDPSocket
21     Public UdpP As UDPPacket
22     Private server As ServerSocket
23     Public ShrIP, ShrPort As String
24 End Sub
```

❑ B4A Coding

```
26  Sub Globals
27      'for Graph
28      Dim Val As Float
29      Dim ValBuf(12) As Float
30      Dim ss, mm As Byte
31      Dim tm, tmbuf(12) As String
32      Private PanelChart As Panel
33      Private LabelValue As Label
34      Private ButtonStart As Button
35      Private ButtonStop As Button
36      Private ButtonReset As Button
37
38      'for udp
39      Dim ImgToggleButton As BitmapDrawable
40      Private ToggleButtonA1 As ToggleButton
41      Private ToggleButtonA2 As ToggleButton
42      Private ToggleButtonA3 As ToggleButton
43      Private EditTextDestinationIP1 As EditTe
44      Public Value As String
45
46      Private ButtonSend As Button
47      Private EditTextDestinationIP2 As EditTe
48      Private EditTextKp As EditText
49      Private EditTextKi As EditText
50      Private EditTextKd As EditText
51      Public PID As String
52      End Sub
```

Define vaiables ❖

❑ B4A Coding

Initial Set-Up ❖

```
54  Sub Activity_Create(FirstTime As Boolean)
55      Activity.LoadLayout("1")
56      'for Graph
57      Timer1.Initialize("Timer1",1000)
58      DateTime.DateFormat = "HH:mm:ss"
59
60      ButtonReset.Enabled = True
61      ButtonStart.Enabled = True
62      'ButtonStop.Enabled = False
63      Reset_Graph
64      '''''''''''''''''''''''''''''''''for udp''''''''''''
65      Private cd As ColorDrawable
66      cd.Initialize(Colors.Transparent, 0)
67      EditTextDestinationIP1.Background = cd
68      If FirstTime Then
69          UdpS.Initialize("UdpSEvent", 2000, 1024)
70      End If
71      EditTextDestinationIP1.Text = ShrIP
72
73
74      ImgToggleButton.Initialize(LoadBitmap(File.DirAssets, "Toggle off.png"))
75      ImgToggleButton.Gravity = Gravity.FILL
76      ToggleButtonA1.Background = ImgToggleButton
77      ToggleButtonA2.Background = ImgToggleButton
78      ToggleButtonA3.Background = ImgToggleButton
79
80  End Sub
```

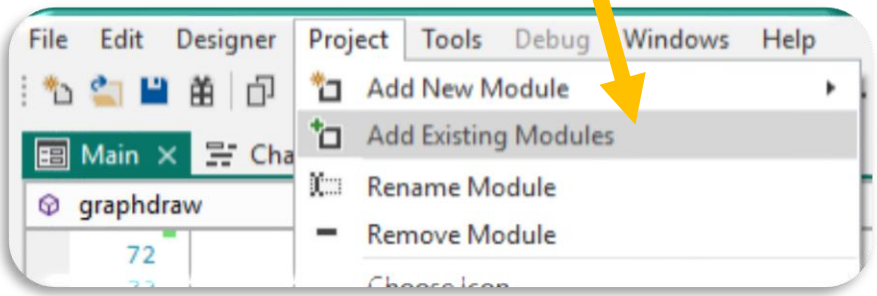
❑ B4A Coding

```
83  Sub graphdraw
84      Dim LD As LineData
85      LD.Initialize
86      LD.Target = PanelChart
87      Charts.AddLineColor(LD, Colors.Red)
88
89      Charts.AddLinePoint(LD, tmbuf(0), ValBuf(0), True)
90      Charts.AddLinePoint(LD, tmbuf(1), ValBuf(1), True)
91      Charts.AddLinePoint(LD, tmbuf(2), ValBuf(2), True)
92      Charts.AddLinePoint(LD, tmbuf(3), ValBuf(3), True)
93      Charts.AddLinePoint(LD, tmbuf(4), ValBuf(4), True)
94      Charts.AddLinePoint(LD, tmbuf(5), ValBuf(5), True)
95      Charts.AddLinePoint(LD, tmbuf(6), ValBuf(6), True)
96      Charts.AddLinePoint(LD, tmbuf(7), ValBuf(7), True)
97      Charts.AddLinePoint(LD, tmbuf(8), ValBuf(8), True)
98      Charts.AddLinePoint(LD, tmbuf(9), ValBuf(9), True)
99      Charts.AddLinePoint(LD, tmbuf(10), ValBuf(10), True)
100     Charts.AddLinePoint(LD, tmbuf(11), ValBuf(11), True)
101
102     'text around graph
103     Dim G As Graph
104     G.Initialize
105     G.Title = "Sensor Value :" & Value
106     'time_now = DateTime.Date(DateTime.Now)
107     G.XAxis = "Time (Minute & Second)"
108     G.YAxis = "Value"
109     G.YStart = 0
110     G.YEnd = 200
111     'G.YInterval = G.YEnd / 10
112     G.YInterval = 20
113     G.AxisColor = Colors.Black
114     Charts.DrawLineChart(G, LD, Colors.White)
115 End Sub
```

❖ Define Number Of Points in Horizontal and Vertical Axis

❖ **First Download Chart Library and :add it to Project**

Android Charts Framework | B4X Programming Forum



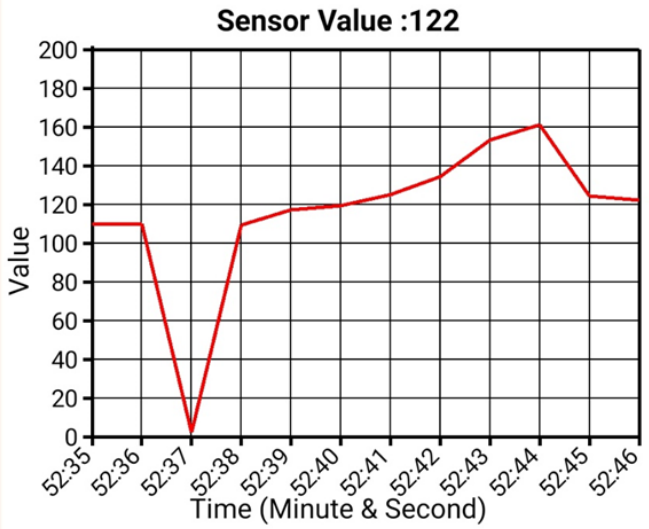
❑ B4A Coding

```
117 Sub TImer1_Tick
118     Val = Value 'Rnd(0,200)
119
120     ValBuf(0) = ValBuf(1)
121     ValBuf(1) = ValBuf(2)
122     ValBuf(2) = ValBuf(3)
123     ValBuf(3) = ValBuf(4)
124     ValBuf(4) = ValBuf(5)
125     ValBuf(5) = ValBuf(6)
126     ValBuf(6) = ValBuf(7)
127     ValBuf(7) = ValBuf(8)
128     ValBuf(8) = ValBuf(9)
129     ValBuf(9) = ValBuf(10)
130     ValBuf(10) = ValBuf(11)
131     ValBuf(11) = Val
132     ss = DateTime.GetSecond(DateTime.Now)
133     mm = DateTime.GetMinute(DateTime.Now)
134     tm = mm & ":" & ss
135
136     tmbuf(0) = tmbuf(1)
137     tmbuf(1) = tmbuf(2)
138     tmbuf(2) = tmbuf(3)
139     tmbuf(3) = tmbuf(4)
140     tmbuf(4) = tmbuf(5)
141     tmbuf(5) = tmbuf(6)
142     tmbuf(6) = tmbuf(7)
143     tmbuf(7) = tmbuf(8)
144     tmbuf(8) = tmbuf(9)
145     tmbuf(9) = tmbuf(10)
146     tmbuf(10) = tmbuf(11)
147     tmbuf(11) = tm
148     'LabelValue.Text = "Sensor Value : " & Value
149     graphdraw
150 End Sub
```

Value points and shift them to the left every second ❖

```
275 Sub UdpSEvent_PacketArrived (Packet As UDPPacket)
276     Value = BytesToString(Packet.Data, Packet.Offset, Packet.Length, "ASCII")
277 End Sub
```

192.168.160.152



❑ B4A Coding

reset, stop & start Part of graph ❖

```
163 Sub ButtonStart_Click
164     Timer1.Enabled = True
165     ButtonStart.Enabled = False
166     ButtonReset.Enabled = False
167     ButtonStop.Enabled = True
168 End Sub
169
170 Sub ButtonStop_Click
171     Timer1.Enabled = False
172     ButtonStop.Enabled = False
173     ButtonStart.Enabled = True
174     ButtonReset.Enabled = True
175 End Sub
176
177 Sub ButtonReset_Click
178     Reset_Graph
179 End Sub
180
181 Sub Reset_Graph
182     ValBuf(0) = 0
183     ValBuf(1) = 0
184     ValBuf(2) = 0
185     ValBuf(3) = 0
186     ValBuf(4) = 0
187     ValBuf(5) = 0
188     ValBuf(6) = 0
189     ValBuf(7) = 0
190     ValBuf(8) = 0
191     ValBuf(9) = 0
192     ValBuf(10) = 0
193     ValBuf(11) = 0
194     graphdraw
195 End Sub
```

❑ B4A Coding

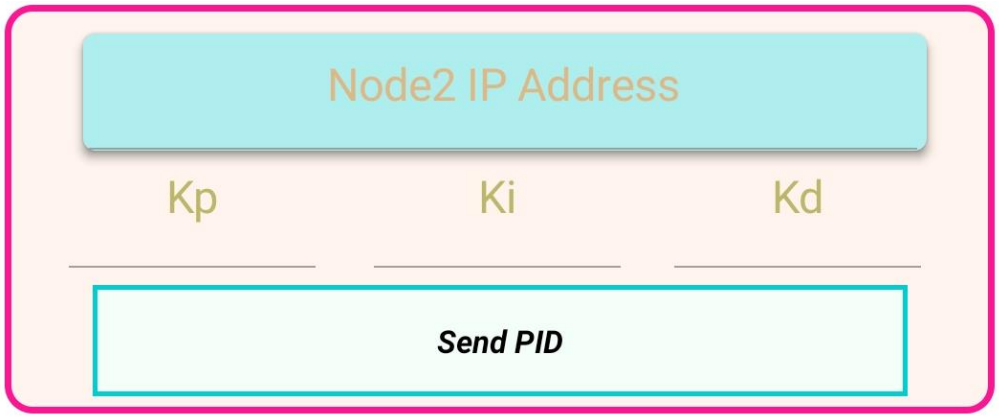
❖ Part in 1st node toggle Button

```
197 Sub ToggleButtonA1_CheckedChange(Checked As Boolean)
198     If EditTextDestinationIP1.Text = "" Then
199         ToastMessageShow("Destination IP and Destination Port cannot be empty !!!",False)
200         Return
201     End If
202
203
204     If Checked = False Then
205         ImgToggleButton.Initialize(LoadBitmap(File.DirAssets, "Toggle off.png"))
206         ImgToggleButton.Gravity = Gravity.FILL
207         ToggleButtonA1.Background = ImgToggleButton
208         UdpP.Initialize("A1Off".GetBytes("ASCII"), EditTextDestinationIP1.Text, 4210) 'Edi
209         UdpS.Send(UdpP)
210
211         ToastMessageShow("Node1:S1 Off", False)
212     Else
213         ImgToggleButton.Initialize(LoadBitmap(File.DirAssets, "Toggle on.png"))
214         ImgToggleButton.Gravity = Gravity.FILL
215         ToggleButtonA1.Background = ImgToggleButton
216         UdpP.Initialize("A1On".GetBytes("ASCII"), EditTextDestinationIP1.Text, 4210) 'Edit
217         UdpS.Send(UdpP)
218
219         ToastMessageShow("Node2:S1 On", False)
220     End If
221 End Sub
```

❑ B4A Coding

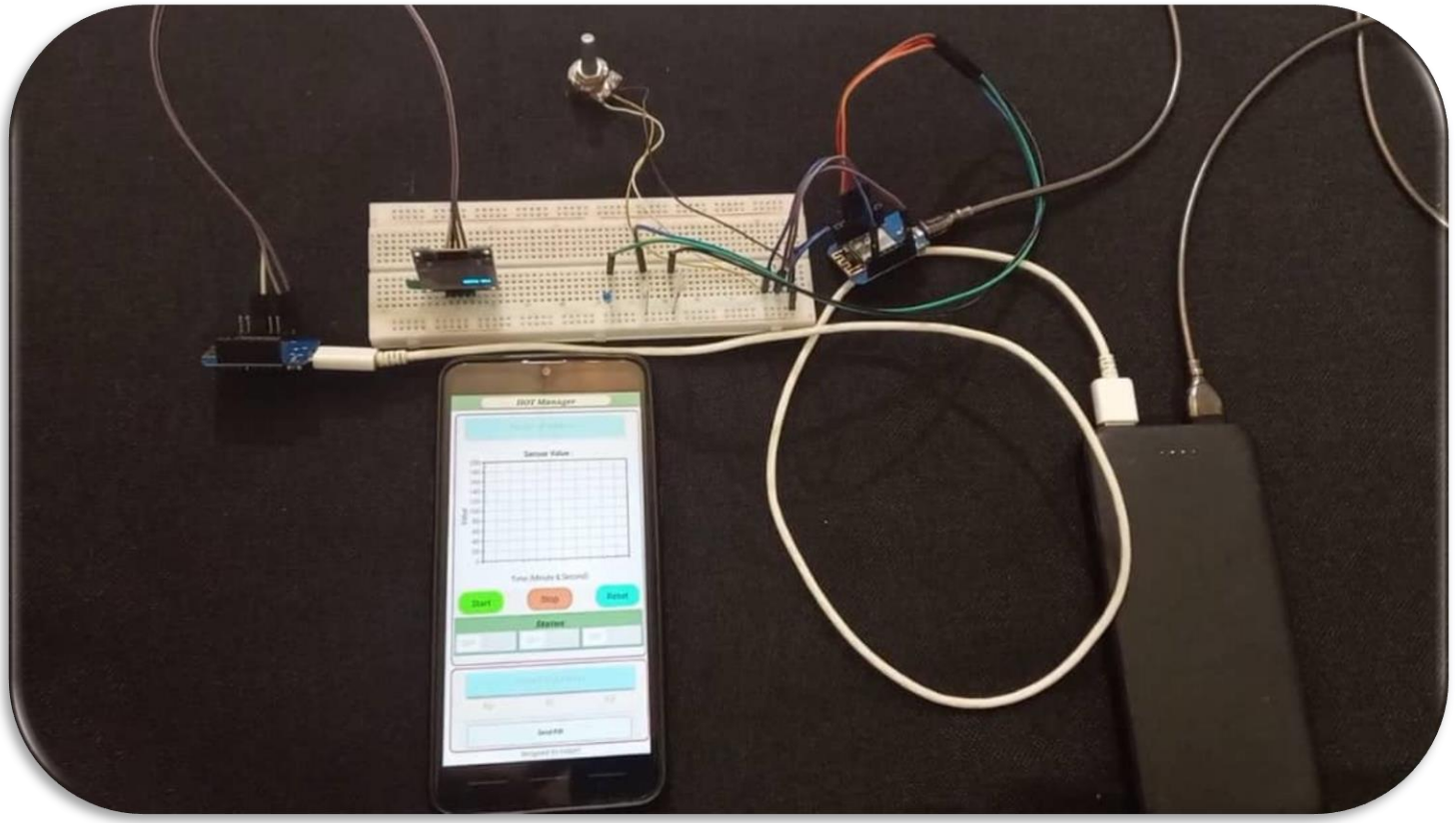
❖ Send PIDs To 2nd Node

```
280  Sub ButtonSend_Click
281      If EditTextDestinationIP2.Text = "" Or EditTextKp.Text = "" Or EditTextKi.Text = "" Or EditTextKd.Te
282          ToastMessageShow("Destination IP,Kp,Ki and Kd cannot be empty !!!",False)
283          Return
284      End If
285      PID = EditTextKp.Text & "S" & EditTextKi.Text & "S" & EditTextKd.Text
286      UdpP.Initialize(PID.GetBytes("ASCII"), EditTextDestinationIP2.Text, 4210) 'EditTextDestinationPort.T
287      UdpS.Send(UdpP)
288      ToastMessageShow("Sending PID.",False)
289  End Sub
```



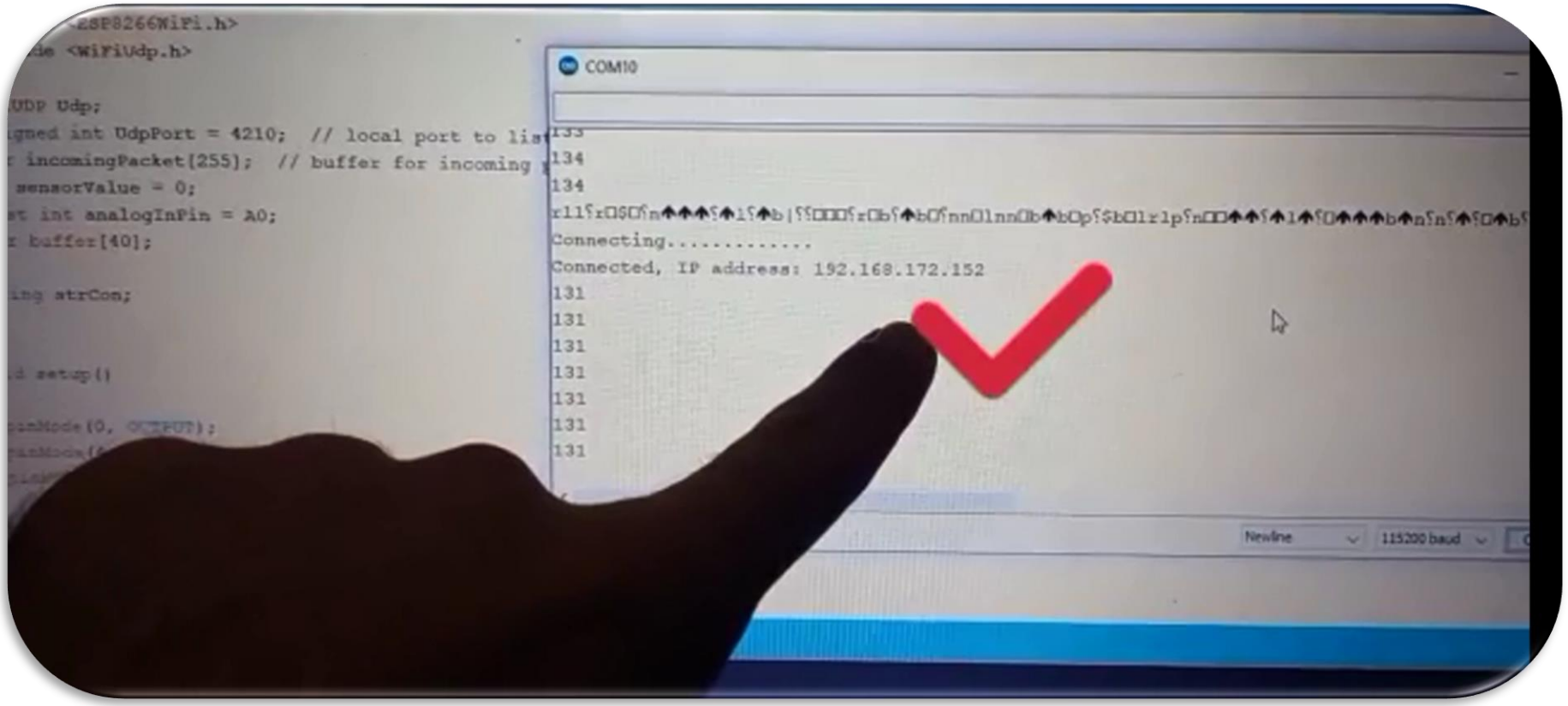
❑ Set up 2 nodes with sensors and modules

View of nodes ❖



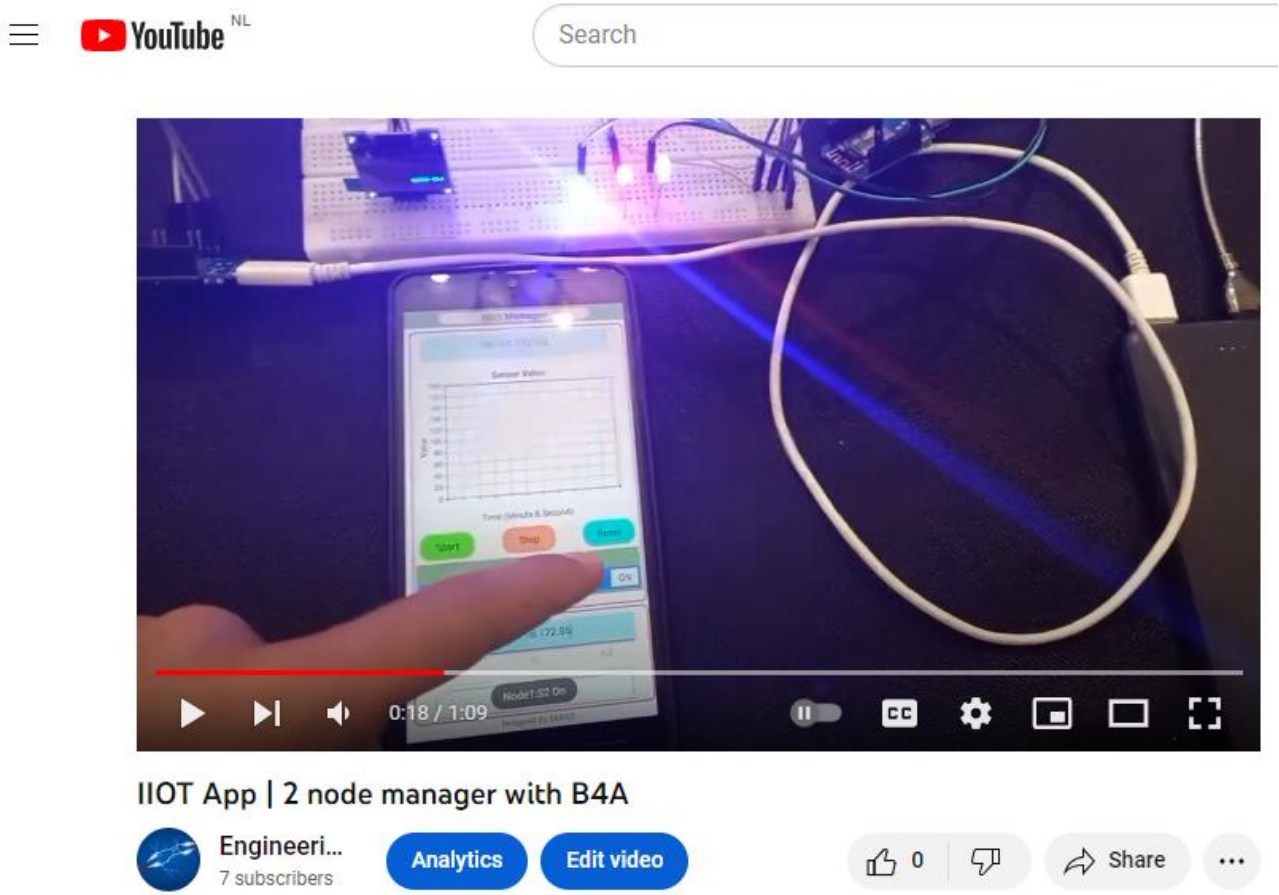
❑ Find IP Addresses

- Find each Node IP via connecting it to main wifi node and see ip in serial port.



❑ Test and Final result

❖ https://www.youtube.com/watch?v=kqubm8_RC5Y



❑ Development Ideas

- ❖ Using ESP-Now Protocol to increase range of connection
- ❖ Plot multiple sensors
- ❖ Using FireBase DataBase and Add Remote control via internet with several Operators
- ❖ Automatic control and Alarming

❑ Addresses

github.com/SmFaraji

Projects channel -> t.me/EngineeringLab
aparat channel -> www.aparat.com/EngineeringLab
YouTube channel -> https://www.youtube.com/@sm_faraji