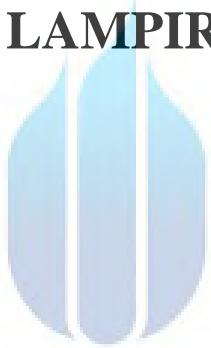


LAMPIRAN



UNIVERSITAS
MERCU BUANA

1. CODINGAN ARDUINO

```
#include <WiFiEsp.h>
#include <DistanceSRF04.h>
#include <Wire.h>
#include <WiFiEspUdp.h>

const char* ssid = "Laura"; //nama wifi
const char* pass = "melinda08"; //password
int status = WL_IDLE_STATUS;
WiFiEspClient client;

//IPAddress ipTujuan(192, 168, 0, 116);
IPAddress ipTujuan(192, 168, 0, 4);
unsigned int localPort = 8888; // local port to listen on

char packetBuffer[255];      // buffer to hold incoming packet
char ReplyBuffer[] = "ACK";   // a string to send back

WiFiEspUDP Udp;

const int buzzer1 = 22;
const int buzzer2 = 23;
const int buzzer3 = 24;
const int buzzer4 = 25;
```

```

const int PIN_TRIG = 2;
const int PIN_ECHO = 3;
const int PIN_TRIG2 = 4;
const int PIN_ECHO2 = 5;
const int PIN_TRIG3 = 6;
const int PIN_ECHO3 = 7;
const int PIN_TRIG4 = 8;
const int PIN_ECHO4 = 9;

```

```

DistanceSRF04 sonar, sonar2, sonar3, sonar4;
int tol = 1;

```

```

void setup() {
    Serial.begin(9600);
    Serial1.begin(115200);
    WiFi.init(&Serial1);
}

```

```

if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while (true);
}

```

```

while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
}

```



```
Serial.println(ssid);
status = WiFi.begin(ssid, pass);
}

Serial.println("You're connected to the network");
Serial.print("My IP address: ");
Serial.println(WiFi.localIP());

sonar.begin(PIN_ECHO, PIN_TRIG); // Penentuan pin Echo dan Trig
sonar2.begin(PIN_ECHO2, PIN_TRIG2);
sonar3.begin(PIN_ECHO3, PIN_TRIG3);
sonar4.begin(PIN_ECHO4, PIN_TRIG4);

pinMode(buzzer1, OUTPUT);
pinMode(buzzer2, OUTPUT);
pinMode(buzzer3, OUTPUT);
pinMode(buzzer4, OUTPUT);

Udp.begin(localPort);

Serial.print("Listening on port ");
Serial.println(localPort);
}

boolean hit = false;
```

```

int b1 = 35;
int b2 = 35;
int b3 = 35;
int b4 = 35;

void loop() {
    int jarak1 = sonar.getDistanceCentimeter();
    int jarak2 = sonar2.getDistanceCentimeter();
    int jarak3 = sonar3.getDistanceCentimeter();
    int jarak4 = sonar4.getDistanceCentimeter();

    if(jarak1>35){
        jarak1=0;
    }

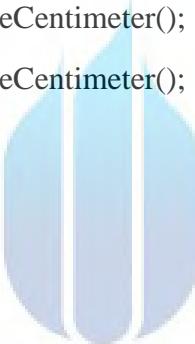
    if(jarak2>35){
        jarak2=0;
    }

    if(jarak3>35){
        jarak3=0;
    }

    if(jarak4>35){
        jarak4=0;
    }

    String inf = String(jarak1) + "=" + String(b1) + "#" + String(jarak2) + "=" +
    String(b2) + "#" + String(jarak3) + "=" + String(b3) + "#" + String(jarak4) + "=" +
    String(b4);
}

```



UNIVERSITAS

MERCU BUANA

```
cetak(inf);

if (hit == false) {
    hit = true;
}
else {
    hit = false;
}

if (hit == false) {
    if (jarak1 >= b1 - tol && jarak1 <= b1 + tol ) {
        digitalWrite(buzzer1,LOW);
        jarak1 = 0;
    }
    else {
        b1 = jarak1;
        digitalWrite(buzzer1,HIGH);
    }
}

if (jarak2 >= b2 - tol && jarak2 <= b2 + tol ) {
    digitalWrite(buzzer2,LOW);
    jarak2 = 0;
}
else {
    b2 = jarak2;
    // digitalWrite(buzzer2,HIGH);
```

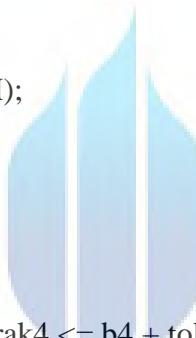


```
//tone (buzzer2, 765);
}

if (jarak3 >= b3 - tol && jarak3 <= b3 + tol ) {
    digitalWrite(buzzer3,LOW);
    jarak3 = 0;
}
else {
    b3 = jarak3;
    digitalWrite(buzzer3,HIGH);
    //tone (buzzer3, 556);
}

if (jarak4 >= b4 - tol && jarak4 <= b4 + tol ) {
    digitalWrite(buzzer4,LOW);
    jarak4 = 0;
}
else {
    b4 = jarak4;
    digitalWrite(buzzer4,HIGH);
    // tone (buzzer4, 856);
}

}
```



UNIVERSITAS
MERCU BUANA

```
String info = String(jarak1) + "#" + String(jarak2) + "#" + String(jarak3) + "#" +
String(jarak4) + "#";
cetak(info);
delay(1000);
}
```

```
void cetak(String info) {
Serial.println(info);

char charBuf[50];
info.toCharArray(charBuf, 50);
```



```
Udp.beginPacket(ipTujuan, localPort);
Udp.write(charBuf);
Udp.endPacket();
```

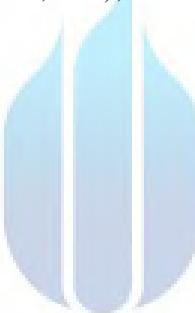
```
}
```

```
void loop2() {
int packetSize = Udp.parsePacket();
if (packetSize) {
Serial.print("Received packet of size ");
```

```
Serial.println(packetSize);
Serial.print("From ");
IPAddress remoteIp = Udp.remoteIP();
Serial.print(remoteIp);
Serial.print(", port ");
Serial.println(Udp.remotePort());

// read the packet into packetBufffer
int len = Udp.read(packetBuffer, 255);
if (len > 0) {
    packetBuffer[len] = 0;
}
Serial.println("Contents:");
Serial.println(packetBuffer);

// send a reply, to the IP address and port that sent us the packet we received
Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
Udp.write(ReplyBuffer);
Udp.endPacket();
}
```



UNIVERSITAS

MERCU BUANA

2. CODINGAN VISUAL BASIC

```
3. using System;
4. using System.Collections.Generic;
5. using System.ComponentModel;
6. using System.Data;
7. using System.Drawing;
8. using System.Text;
9. using System.Windows.Forms;
10. using System;
11. using System.Net;
12. using System.Net.Sockets;
13. using System.Text;
14. using System.Threading.Tasks;
15.
16.
17. namespace Blaney
18. {
19.     public partial class frmMain : Form
20.     {
21.         Radar _radar; N I V E R S I T A S
22.         Timer t = new Timer();
23.         Random rnd = new Random();
24.
25.         private const int MyPort = 8888;
26.         private UdpClient Client;
27.
28.         public frmMain()
29.         {
30.             InitializeComponent();
31.             // internal item update timer
32.             t.Interval = 60;
33.             t.Tick += new EventHandler(t_Tick);
34.             t.Enabled = true;
35.
36.             Client = new UdpClient(MyPort);
37.             Client.BeginReceive(DataReceived, null);
38.         }
39.
```

```
40.     RadarItem item1 = new SquareRadarItem(1, 8, 190, 60);
41.     RadarItem item2 = new SquareRadarItem(2, 8, 45, 45);
42.     RadarItem item3 = new SquareRadarItem(3, 8, 30, 30);
43.     RadarItem item4 = new SquareRadarItem(4, 8, 35, 35);
44.
45.     int GetDelta()
46.     {
47.         int i = rnd.Next(0, 2);
48.         if (i == 0)
49.             i--;
50.         return i;
51.     }
52.
53.     void t_Tick(object sender, EventArgs e)
54.     {
55.         try
56.         {
57.
58.             String txt = txtstatus.Text;
59.             String[] ar = txt.Split('#');
60.
61.             String y1 = ar[0];
62.             String y2 = ar[1];
63.             String y3 = ar[2];
64.             String y4 = ar[3];
65.
66.             // String x = textBox1.Text;
67.             // String y = textBox2.Text;
68.             int xx1 = 40;
69.             int yy1 = Convert.ToInt32(y1);
70.
71.             int xx2 = -40;
72.             int yy2 = Convert.ToInt32(y2);
73.
74.             int xx3 = -128;
75.             int yy3 = Convert.ToInt32(y3);
76.
77.             int xx4 = 128;
78.             int yy4 = Convert.ToInt32(y4);
79.
80.             item1.Azimuth = xx1;//GetDelta();
81.             item1.Elevation = Convert.ToInt32(yy1); // GetDelta();
82.             _radar.AddItem(item1);
```

```
83.  
84.     item2.Azimuth = xx2;//GetDelta();  
85.     item2.Elevation = Convert.ToInt32(yy2); // GetDelta();  
86.     _radar.AddItem(item2);  
87.  
88.     item3.Azimuth = xx3;//GetDelta();  
89.     item3.Elevation = Convert.ToInt32(yy3); // GetDelta();  
90.     _radar.AddItem(item3);  
91.  
92.     item4.Azimuth = xx4;//GetDelta();  
93.     item4.Elevation = Convert.ToInt32(yy4); // GetDelta();  
94.     _radar.AddItem(item4);  
95. }  
96. catch (Exception ee) { }  
97.  
98.  
99.  
100.    }  
101.  
102.  
103.  
104.    private void DataReceived(IAsyncResult ar)  
105.    {  
106.        IPEndPoint ip = new IPEndPoint(IPAddress.Any, MyPort);  
107.        byte[] data;  
108.        try  
109.        {  
110.            data = Client.EndReceive(ar, ref ip);  
111.  
112.            if (data.Length == 0)  
113.                return; // No more to receive  
114.            Client.BeginReceive(DataReceived, null);  
115.        }  
116.        catch (ObjectDisposedException)  
117.        {  
118.            return; // Connection closed  
119.        }  
120.  
121.        // Send the data to the UI thread  
122.        this.BeginInvoke((Action<IPEndPoint,  
123.                           string>)DataReceivedUI, ip, Encoding.UTF8.GetString(data));  
124.    }
```

```
125.     private void DataReceivedUI(IPEndPoint endPoint, string data)
126.     {
127.         txtstatus.Text = data+Environment.NewLine;
128.     }
129.
130.    private void frmMain_Load(object sender, EventArgs e)
131.    {
132.        _radar = new Radar(pictureBox1.Width);
133.        pictureBox1.Image = _radar.Image;
134.        _radar.ImageUpdate += new
135.            ImageUpdateHandler(_radar_ImageUpdate);
136.        _radar.DrawScanInterval = 60;
137.        _radar.DrawScanLine = true;
138.    }
139.
140.    void _radar_ImageUpdate(object sender, ImageUpdateEventArgs
141.    e)
142.    {
143.        // this event is important to catch!
144.        pictureBox1.Image = e.Image;
145.    }
146.    private void button1_Click(object sender, EventArgs e)
147.    {
148.        // return 0;
149.    }
150.
151.    }
152.
153.
154.
155.    private void label3_Click(object sender, System.EventArgs e)
156.    {
157.    }
158.    }
159.
160.    private void timer1_Tick(object sender, System.EventArgs e)
161.    {
162.        txtJam.Text = DateTime.Now.ToString("dd MMM yyyy");
163.        txtTanggal.Text = DateTime.Now.ToString("HH:mm:ss");
164.    }
165.
```

```
166.     private void button1_Click_1(object sender, System.EventArgs e)
167.     {
168.         String x = textBox1.Text;
169.         String y = textBox2.Text;
170.
171.         item1.Azimuth = Convert.ToInt32(x); //GetDelta();
172.         item1.Elevation = Convert.ToInt32(y); // GetDelta();
173.         _radar.AddItem(item1);
174.
175.     }
176.
177.     private void label1_Click(object sender, System.EventArgs e)
178.     {
179.
180.     }
181.
182.
183.
184. }
185.
186.
187.
188. }
189.
190.
```

