Practical–14

Aim: What's GPIO & it's use, Interfacing & Programming of LED & RGB LED with Arduino UNO.

GPIO:

GPIO stands for General Purpose Input/Output. It's a standard interface used to connect microcontrollers to other electronic devices. For example, it can be used with sensors, diodes, displays, and System-on-Chip modules. Estimote Location Beacons (Hardware Revision F2.3 and later) are equipped with GPIO. It allows for providing external power supply, remote control of connected devices, broadcasting more contextual data, or defining contents for custom Bluetooth data packets.

Used For:

The most common use for GPIO is to operate custom electronics. Whether you're building your own robot arm or a DIY weather station, a GPIO interface lets you customize signals so that they operate your equipment correctly.

```
// Import required libraries
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>

// Replace with your network credentials
const char *ssid = "realme_8";
const char *password = "ankit123";

#define DHTPIN 5 // Digital pin connected to the DHT sensor

// Uncomment the type of sensor in use:
//#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

// current temperature & humidity, updated in loop()
float t = 0.0;
float h = 0.0;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);
```

```
// Generally, you should use "unsigned long" for variables that hold
time // The value will quickly become too large for an int to store
unsigned long previousMillis = 0; // will store last time DHT was updated

// Updates DHT readings every 10 seconds
const long interval = 10000;

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1"> <link
rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">

 <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">

  <style>
body{
  background-color: #241E4E;
  padding:0;
  margin:0;
}
.container{
  height: 350px;
  width: 350px;
  border-radius: 50%;
  position: absolute;
  margin: auto;
  top:0;
  bottom: 0;
  left:0;
  right: 0;
}
.snow{
  height: 40px;
  width: 235px;
  background-color: white;
  position: relative;
  left:60px;
  top:280px;
  border-radius: 8px;
}
.tree1{
  height: 0;
  width: 0;
  border-bottom: 120px solid #5b6b09;
  border-left: 20px solid transparent;
```

```
  border-right: 20px solid
  transparent; position: relative;
  left:70px;
  top:120px;
}
.tree2{
  height: 0;
  width: 0;
  border-bottom: 90px solid
  #718710; border-left: 24px solid
  transparent; border-right: 24px
  solid transparent; position: relative;
  left:90px;
  top:30px;
}
.house{
  position: relative;
  bottom:78px;
  left:146px;
}
.roof1{
  height: 60px;
  width: 12px;
  background-color: #7e110f;
  transform: skew(30deg);
  -webkit-transform:
  skew(30deg); position: relative;
  left:112px;
  bottom:18.5px;
}
.wall1{
  height: 0;
  width: 0;
  border-bottom: 53px solid
  #c44e0e; border-left:30px solid
  transparent; border-right:31px solid
  transparent; position: relative;
  bottom: 75px;
  left:68px;
}
.wall2{
  height:70px;
  background-color:
  #c44e0e; width: 150px;
  position: relative;
  bottom:75px;
  right: 21px;
}
.wall3{
  background-color: #b40f0c;
```

```
  height: 60px;
  width: 105px;
  transform: skew(-30deg);
  -webkit-transform: skew(-
  30deg); position: relative;
  bottom:201.5px;
  right: 16.5px;
}
.light{
  background-color:
  #ef7421; width: 90px;
  height: 50px;
  position: relative;
  top:20px;
}
.b1{
  background-color:
  #460701; height: 60px;
  width: 3px;
  position: relative;
  left:9px;
}
.b2{
  background-color:
  #460701; height:6px;
  width:10px;
  position: relative;
  bottom:6px;
}
.b4{
  background-color:
  #460701; height:59px;
  width: 3px;
  position: relative;
  left:103px;
  bottom:5px;
}
.b3{
  background-color:
  #7b0f0c; height: 5px;
  width: 106px;
  position: relative;
  top: 55px;
}
.w1,.w2{
  background-color:
  #700205; height: 30px;
  width:25px;
  position:relative;
}
```

```
.w1{
  top:5px;
  left:20px;
}
.w2{
  bottom:25px;
  left:50px;
}
.window{
  background-color:#89c6c5;
  height: 23px;
  width: 19px;
  border:2px solid white;
  position: relative;
  top:2px;
  left:3px;
}
.glass{
  height: 0;
  width: 0;
  border-top:23px solid
rgba(255,255,255,0.5); border-right: 17px
solid transparent; }
.w3{
  background-color: #982c19;
  height:18px;
  width: 16px;
  position: relative;
  top:40px;
  right: 3px;
  z-index: 1;
}
.window1{
  background-color:#89c6c5;
  height: 10px;
  width: 10px;
  border:1.3px solid white;
  position: relative;
  top:2.5px;
}
.glass1{
  height: 0;
  width: 0;
  border-top:10px solid
rgba(255,255,255,0.3); border-right: 10px
solid transparent; }
.door{
  background-color: #7f1414;
  height: 40px;
  width: 20px;
```

```
  position: relative;
  bottom:20px;
  left:112px;
  border-right: 2px solid
#ef2021; }
.handle{
  height: 8px;
  width: 4px;
  background-color:
  #cccccc; position: relative;
  top:16px;
  left:3px;
}
.chimney{
  background-color:
  #c85012; height:22px;
  width: 20px;
  transform: skew(30deg);
  -webkit-transform:
  skew(30deg); position: relative;
  bottom:75px;
  left:55px;
}
.shne1{
  background-color:
  #ef7421; height: 14px;
  width: 14px;
  position: relative;
  bottom: 20px;
  left:5px;
}
.shne2{
  background-color:
  #ef7421; height:10px;
  width: 20px;
  position: relative;
  bottom: 47px;
}
.top{
  background-color:
  #c85012; height:10px;
  width: 28px;
  position: relative;
  bottom:5px;
  right: 4px;
}
.smoke{
  position: relative;
  bottom:32px;
  right: 15px;
```

```
animation: smoke 5s infinite;
  -webkit-animation: smoke 5s infinite;
}
@keyframes smoke{
  50%{
    transform: translateX(-5px) translateY(-
    10px); opacity: 0.5;
  }
  100%{
    transform: translateX(10px) translateY(-
    20px); opacity: 0;
  }
}
@-webkit-keyframes smoke{
  50%{
    -webkit-transform: translateX(-5px) translateY(-
    10px); opacity: 0.5;
  }
  100%{
    -webkit-transform: translateX(10px) translateY(-
    20px); opacity: 0;
  }
}
.s1,.s2,.s3{
  background-color: #423f3f;
  height: 10px;
  opacity: 0.2;
  border-radius: 10px;
  position: relative;
  margin-bottom: 2px;
}
.s1{
  width:35px;
}
.s2{
  width: 27px;
  left:8px;
}
.s3{
  height:7px;
  width: 15px;
  left:25px;
}
.snowfall{
  background:
  url(https://i.imgur.com/T0mBFg2.png);
  background-size: 350px 200px;
  background-repeat: repeat;
  height: 350px;
  width: 350px;
  position:relative;
```

```
  bottom: 493px;
  border-radius: 50%;
  animation:snowfall 5s linear forwards infinite; -
webkit-animation:snowfall 5s linear forwards infinite; }
@keyframes snowfall{
  0%{
    background-position: 0 0;
  }
  20%{
    background-position: 20px 20px;
  }
  40%{
    background-position: 40px 40px;
  }
  60%{
    background-position: 60px 60px;
  }
  80%{
    background-position: 80px 80px;
  }
  100%{
    background-position: 100px 100px;
  }
}
@-webkit-keyframes snowfall{
  0%{
    background-position: 0 0;
  }
  20%{
    background-position: 20px 20px;
  }
  40%{
    background-position: 40px 40px;
  }
  60%{
    background-position: 60px 60px;
  }
  80%{
    background-position: 80px 80px;
  }
  100%{
    background-position: 100px 100px;
  }
}
.cover{
  background-color:
  rgba(255,255,255,0.2); height: 350px;
  width: 350px;
  border-radius: 50%;
```

```
  position: relative;
  bottom:842px;
}
.bottom{
  background-color:
  #c1272e; height:70px;
  width: 260px;
  position: relative;
  bottom:880px;
  left:47px;
}
.bt1,.bt2{
  background-color:
  #d33c3c; height: 20px;
  width:280px;
  position: relative;
  right: 10px;
  border-radius: 10px;
}
.bt1{
  bottom: 12px;
}
.bt2{
  top:40px;
}
.sn{
  background-color: white;
  width: 40px;
  height: 5px;
  transform: skew(30deg);
  -webkit-transform:
  skew(30deg); border-radius:
  5px;
  position: relative;
  bottom: 36px;
  left:5px;
}
.sn1{
  background-color: white;
  width: 30px;
  height: 5px;
  transform: skew(30deg);
  -webkit-transform:
  skew(30deg); border-radius:
  5px;
  position: relative;
  bottom: 41px;
  left:55px;
}
.dr1,.dr2,.dr3,.dr4{
  background-color: white;
  border-radius: 5px;
```

```
  position: relative;
  height: 10px;
  width: 4px;
}
.dr1{
  top:4px;
  left:5px;
}
.dr2{
  bottom:8px;
  left:15px;
}
.dr3{
  height: 8px;
  width: 8px;
  border-radius: 50%;
  left:29px;
  bottom:19px;
}
.dr4{
  left:22px;
}
.snw1{
  background-color: white;
  height: 15px;
  width: 25px;
  position: relative;
  bottom:30px;
  left:7px;
  border-radius: 20px 20px 0
0; }
.snw2{
  background-color: white;
  height: 12px;
  width: 12px;
  position: relative;
  bottom:40px;
  left:31px;
  border-radius: 50%
}
.sh1,.sh2,.sh3,.sh4,.sh5{
  background-color:
  #d33c3c; height: 6px;
  transform: skew(30deg);
  -webkit-transform:
  skew(30deg); width: 23px;
  border-radius: 15px;
  position: relative;
}
.sh1{
```

```
  bottom:56px;
  left:70px;
}
.sh2{
  bottom:68px;
  left:2px;
}
.sh3{
  bottom:95px;
  left:47px;
}
.sh4{
  bottom:108px;
  left:6px;
}
.sh5{
  bottom:95px;
  left:38px;
}
.fence{
  position: relative;
  bottom: 1008px;
  left:103px;
}
.fn1,.fn2,.fn3{
  background-color:
  #b40f0c; height: 25px;
  width: 7px;
  position: relative;
  border-radius: 25px 25px 0
0; }
.fn2{
  bottom:25px;
  left:12px;
}
.fn3{
  bottom:50px;
  left:24px;
}
.screw{
  background-color:
  #cccccc; height: 3px;
  width: 3px;
  position: relative;
  border-radius: 50%;
  left:1.6px;
  top:11px;
  z-index: 2;
}
.stck{
```

```
  background-color:
  #7d151f; position: relative;
  width: 46px;
  height: 6px;
  right: 8px;
  bottom:65px;
}


    html {
     font-family: Arial;
     display: inline-block;
     margin: 0px auto;
     text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem;
    } .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
<body>
<div class="container">
  <div
  class="snow"></div>
  <div class="tree1"></div>
  <div class="tree2"></div>
  <div class="house">
    <div class="roof1">
      <div class="b1"></div>
      <div class="b2"></div>
    </div>
    <div class="wall1">
      <div class="w3">
        <div class="window1">
          <div
        class="glass1"></div>
        </div>
      </div>
    </div>
    <div class="wall2">
      <div class="light">
        <div class="w1">
          <div class="window">
            <div
          class="glass"></div>
          </div>
        </div>
        <div class="w2">
```

```
<div class="window">
          <div
         class="glass"></div>
         </div>
       </div>
      </div>
      <div class="door">
       <div
      class="handle"></div>
      </div>
      <div
    class="snw1"></div> <div
    class="snw2"></div>
    </div>
    <div class="wall3">
      <div class="b3"></div>
      <div class="b4"></div>
      <div class="chimney">
       <div class="top">
         <div class="smoke">
           <div class="s1"></div>
           <div class="s2"></div>
           <div class="s3"></div>
         </div>
         <div
        class="shne1"></div> <div
        class="shne2"></div>
        </div>
      </div>
      <div class="sn">
        <div class="dr1"></div>
        <div class="dr2"></div>
        <div class="dr3"></div>
      </div>
      <div class="sn1">
        <div class="dr4"></div>
      </div>
      <div
    class="sh1"></div> <div
    class="sh2"></div> <div
    class="sh3"></div> <div
    class="sh4"></div> <div
    class="sh5"></div>
    </div>
  </div>
  <div
  class="snowfall"></div>
  <div class="cover"></div>
  <div class="bottom">
    <div class="bt1"></div>
    <div class="bt2"></div>
  </div>
  <div class="fence">
```

```
    <div class="fn1"> <divclass="screw"></div>
    <div class="fn2">
      <div class="screw"></div>
    </div>
    <div class="fn3">
      <div class="screw"></div>
    </div>
    <div class="stck"></div>
  </div>

  <h2>ESP8266 DHT Server</h2>
  <p>
    <i class="fas fa-thermometer-half"
    style="color:#059e8a;"></i> <span class="dht-
    labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">%</sup>
  </p>
</body>
<script>
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("temperature").innerHTML =
    this.responseText; }
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("humidity").innerHTML =
    this.responseText; }
  };
  xhttp.open("GET", "/humidity", true);
  xhttp.send();
}, 10000 ) ;
</script>
</html>)rawliteral";
```

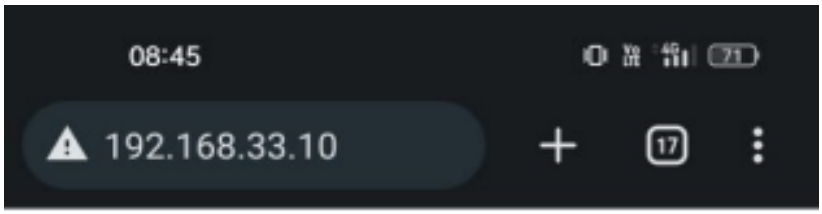
```
{
  // Serial.println(var);
  if (var == "TEMPERATURE")
  {
    return String(t);
  }
  else if (var == "HUMIDITY")
  {
    return String(h);
  }
  return String();
}

void setup()
{
  // Serial port for debugging purposes
  Serial.begin(115200);
  dht.begin();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.println(".");
  }

  // Print ESP8266 Local IP Address
  Serial.println(WiFi.localIP());

  // Route for root / web page
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) { request-
                    >send_P(200, "text/html", index_html, processor); });
          server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest
           *request) { request->send_P(200, "text/plain", String(t).c_str()); });
          server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest
       *request) { request->send_P(200, "text/plain", String(h).c_str()); });

  // Start server
  server.begin();
}

void loop()
{
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval)
```

```
 {
  // save the last time you updated the DHT values
  previousMillis = currentMillis;
```



```
  // Read temperature as Celsius (the default)
  float newT = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  // float newT = dht.readTemperature(true);
  // if temperature read failed, don't change t
  value if (isnan(newT))
  {
    Serial.println("Failed to read from DHT
  sensor!"); }
  else
  {
    t = newT;
    Serial.println(t);
  }
  // Read Humidity
  float newH = dht.readHumidity();
  // if humidity read failed, don't change h
  value if (isnan(newH))
  {
    Serial.println("Failed to read from DHT
  sensor!"); }
  else
  {
    h = newH;
    Serial.println(h);
  }
 }
}
```

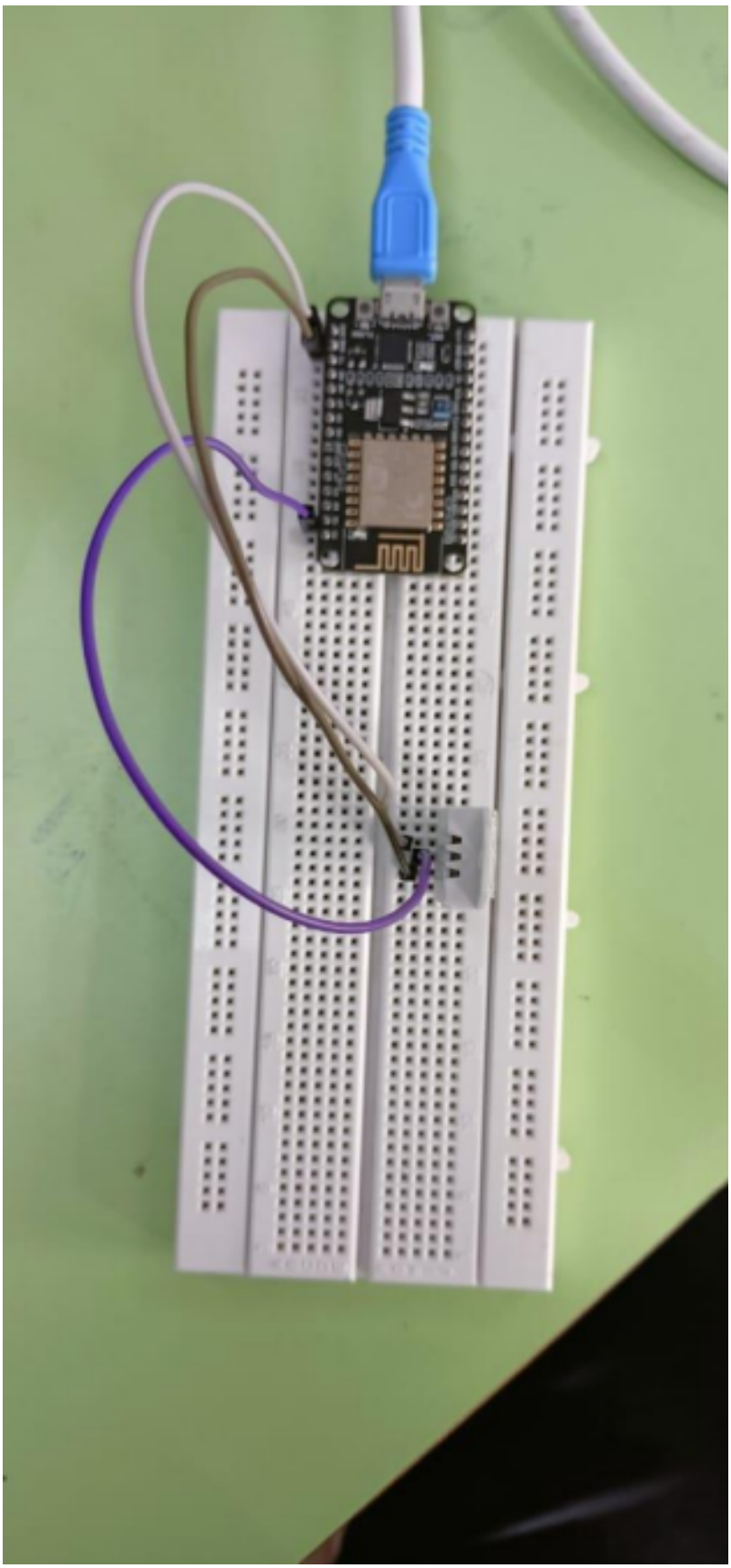
Output:

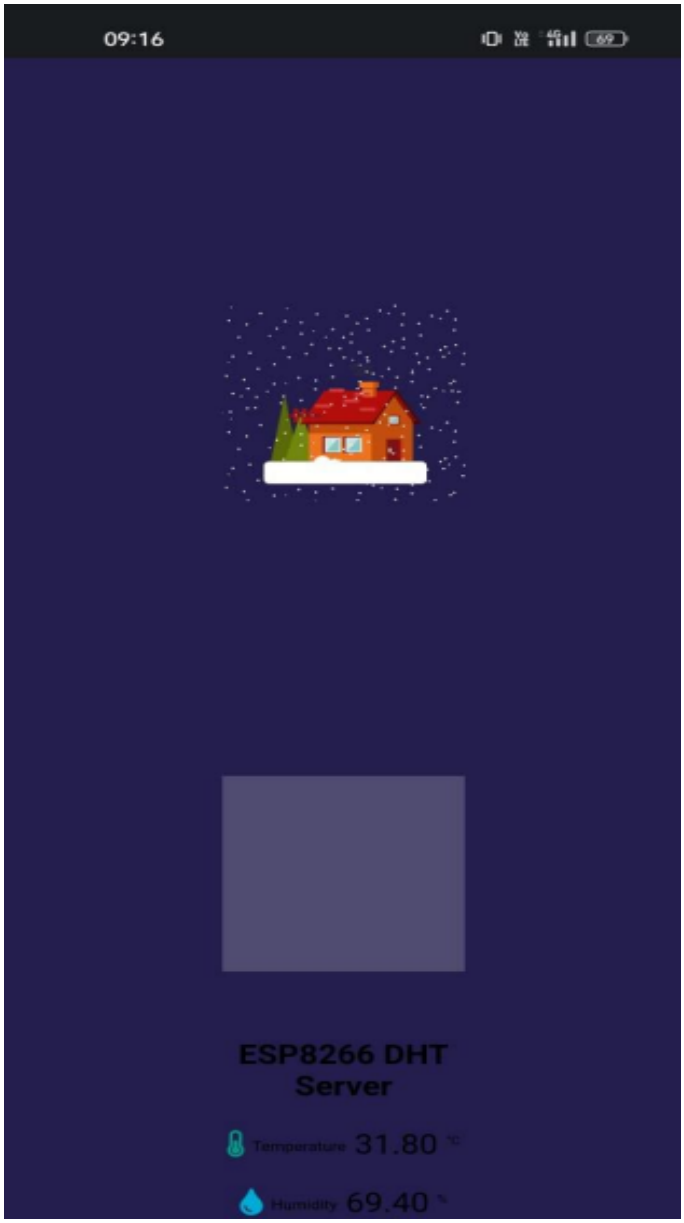


ESP8266 DHT Server

Temperature 32.00 °C

Humidity 70.90 %




09:16
ESP8266 DHT
Server
Temperature 31.80 °C
Humidity 69.40 %