Cloud to Device

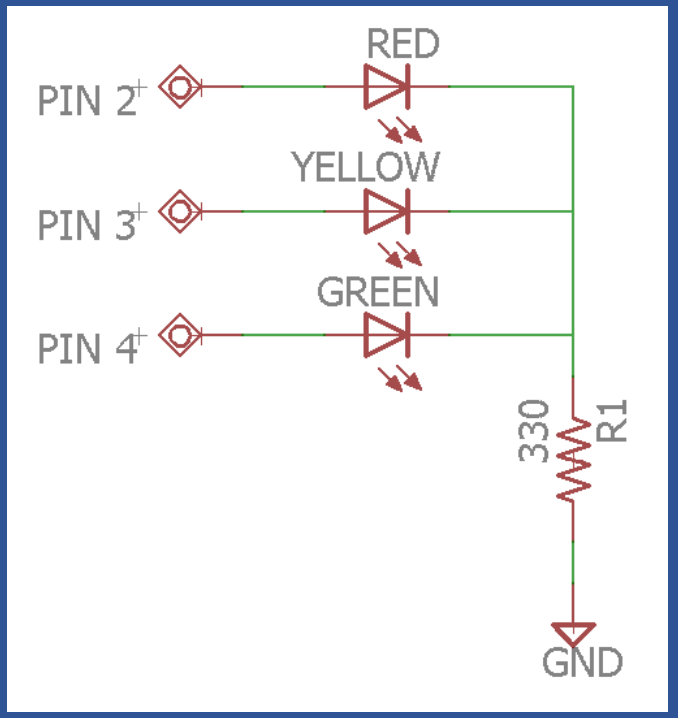
- Controlling Actuators
- Status Indicator
- Warning and Alert
- Power Electric device
- Sending command from Cloud to Device
- Responding to acknowledgement

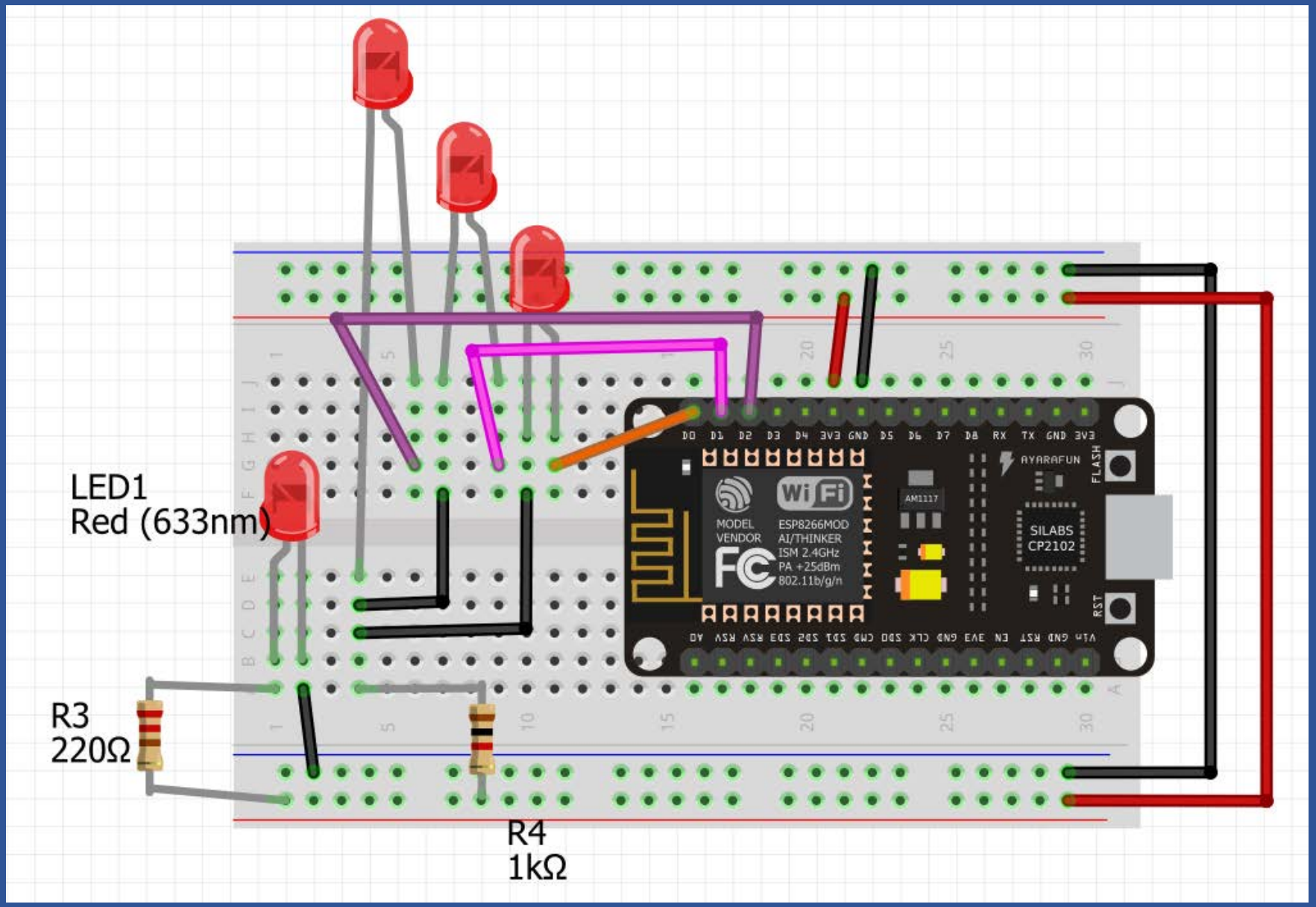
Controlling Actuators

1. LED Status indicator
2. Motor (Fan)
3. Buzzer (Warning & Alert)
4. Relay (Control Electric)

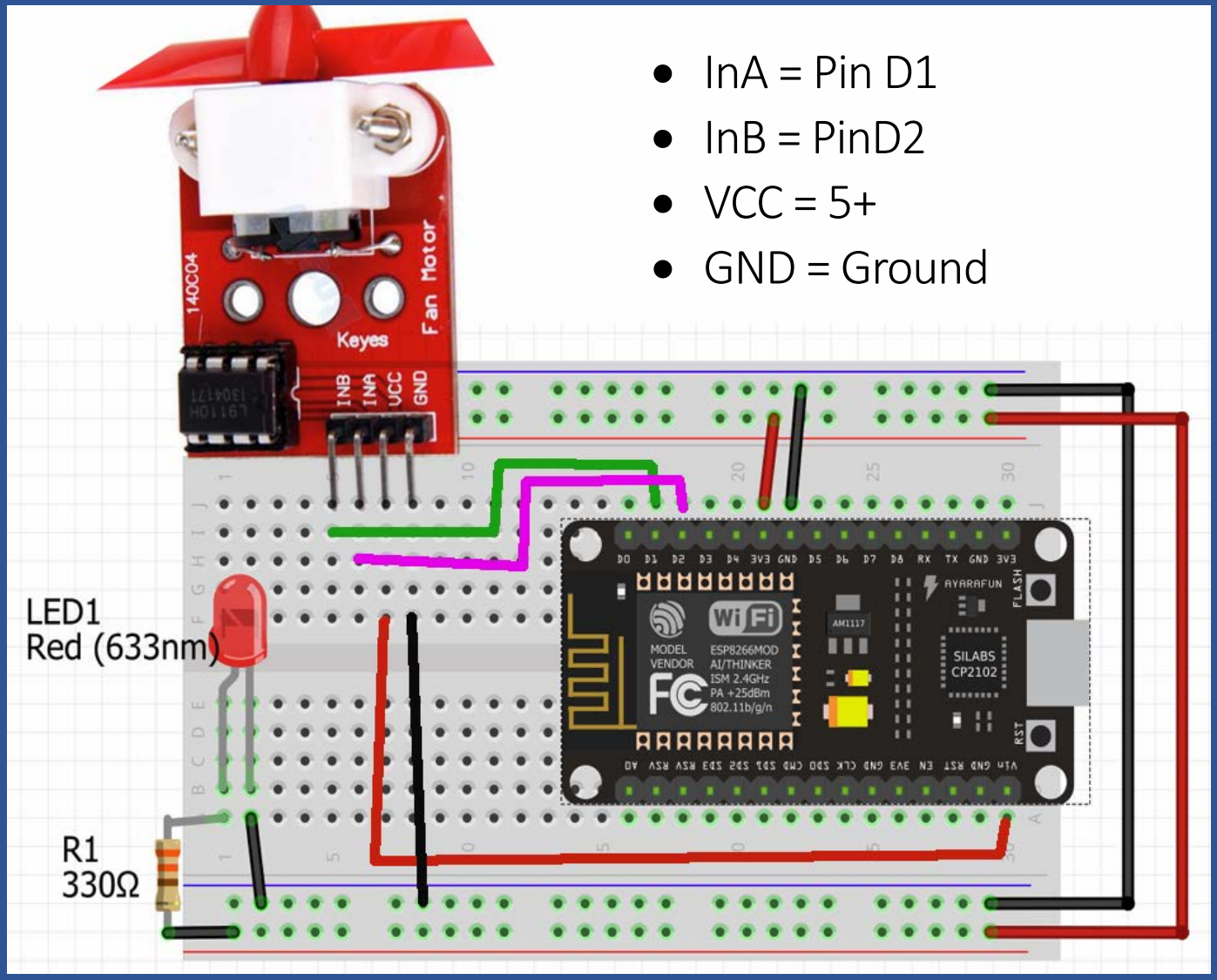
LED Status indicator

1. Blink 3 LEDs.
2. Control blink speed.
3. Control blink speed 3 LEDs.
4. Control LED brightness.
5. Create WinForm with 3 buttons to display status and control 3 LEDs
6. Control blink speed from C#
7. Control blink speed each LED C#
8. Control brightness from C#




LED1
Red (633nm)
R3
220Ω
R4
1kΩ
D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
AYARAFUN
FLASH
RST
AM1117
SILABS
CP2102
MODEL
VENDOR
ESP8266MOD
AI/THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n

Motor (Fan)



- InA = Pin D1
- InB = PinD2
- VCC = 5+
- GND = Ground

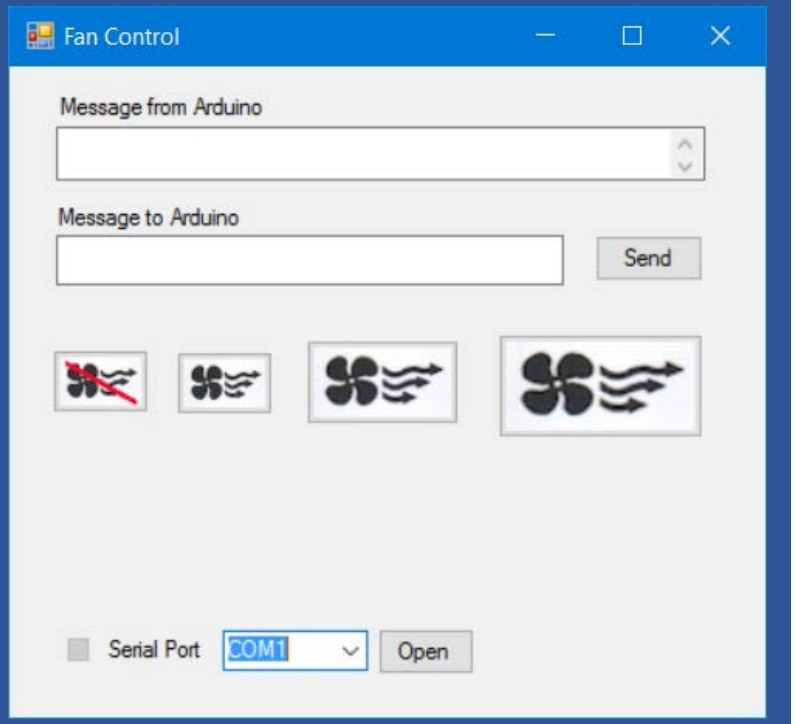
Fan1: Basic Code

```
1  int InA = D1;
2  int InB = D2;
3  void setup() {
4     pinMode(InA,OUTPUT);
5     pinMode(InB,OUTPUT);
6     digitalWrite(InA,LOW);
7     digitalWrite(InB,HIGH);
8    }
9
10 void loop() {
11 }
```

Fan2: Speed Control

```
1  int InA = D1;
2  int InB = D2;
3  void setup() {
4     pinMode(InA,OUTPUT);
5     pinMode(InB,OUTPUT);
6     digitalWrite(InA,LOW);
7     digitalWrite(InB,LOW);
8     digitalWrite(InA,HIGH);
9     delay(10);
10    analogWrite(InA,150);
11   }
12
13 void loop() {
14 }
```

Fan3: Fan speed control from C#

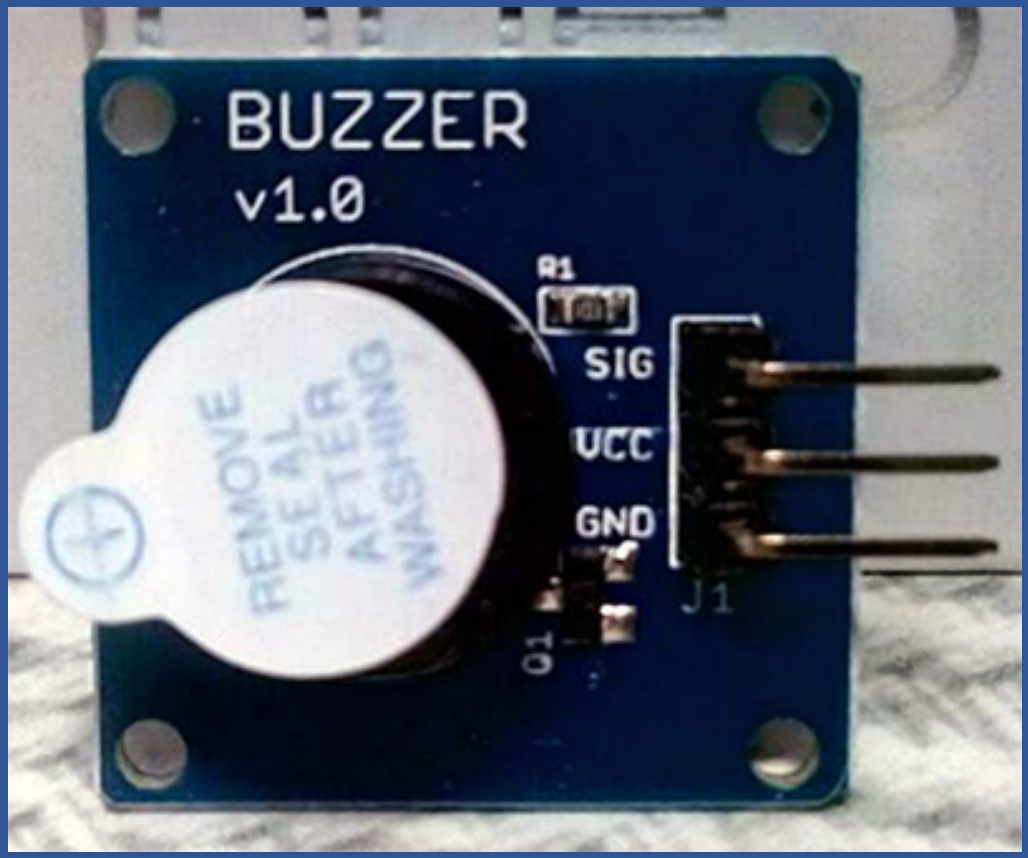


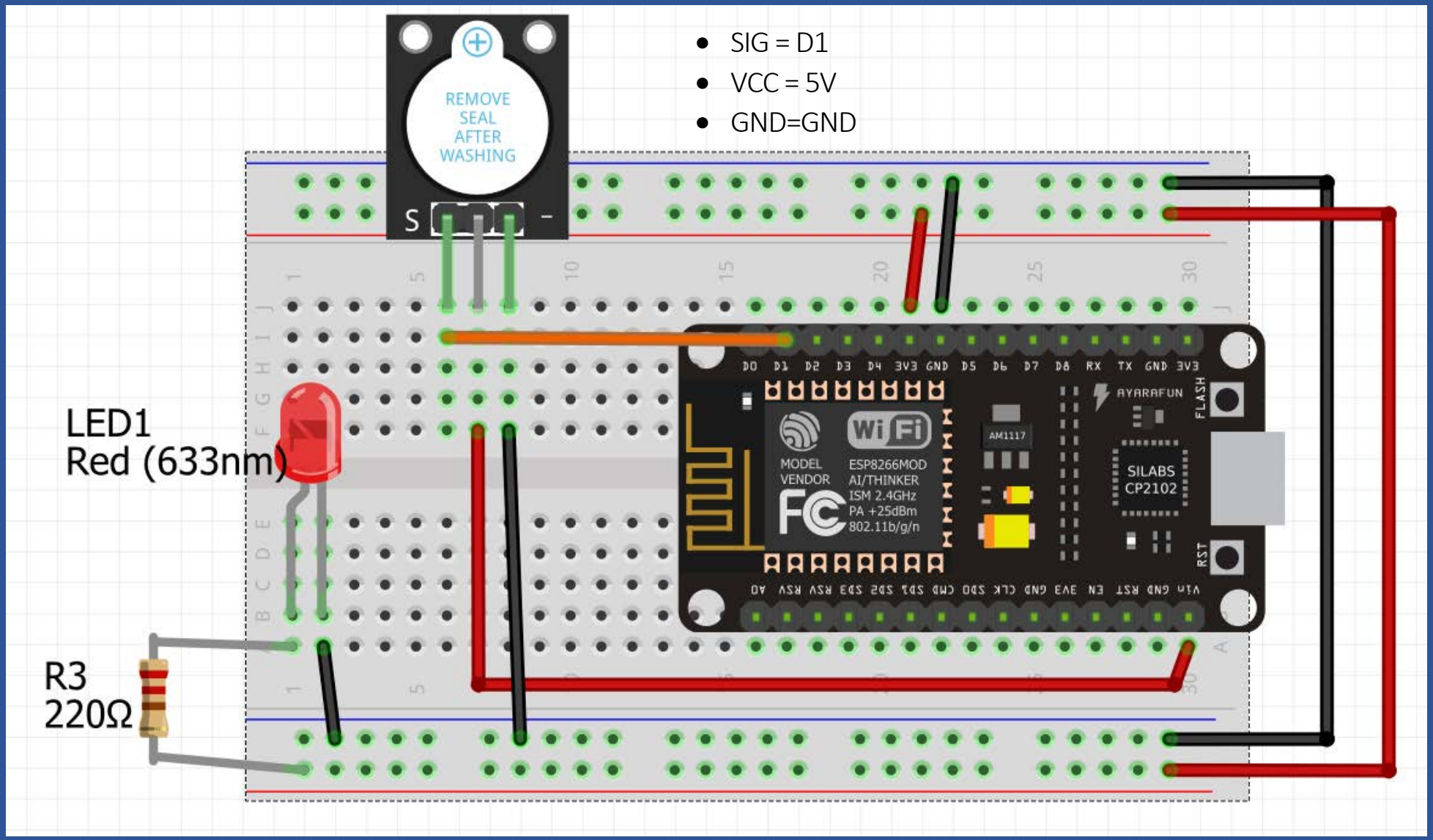

```
int InA = D1;
int InB = D2;

void setup() {
   pinMode(InA,OUTPUT);
   pinMode(InB,OUTPUT);
   digitalWrite(InA,LOW);
   digitalWrite(InB,LOW);
   Serial.begin(9600);
   Serial.println("Ready");
  }

void loop() {
    if (Serial.available() > 0) {
    String s = Serial.readString();
    int x = s.toInt();
    byte volts = (byte)x;
    Serial.print(s);
    digitalWrite(InA,HIGH);
    delay(100);
    analogWrite(InA, volts);
  }
}
```

Buzzer (Warning & Alert)



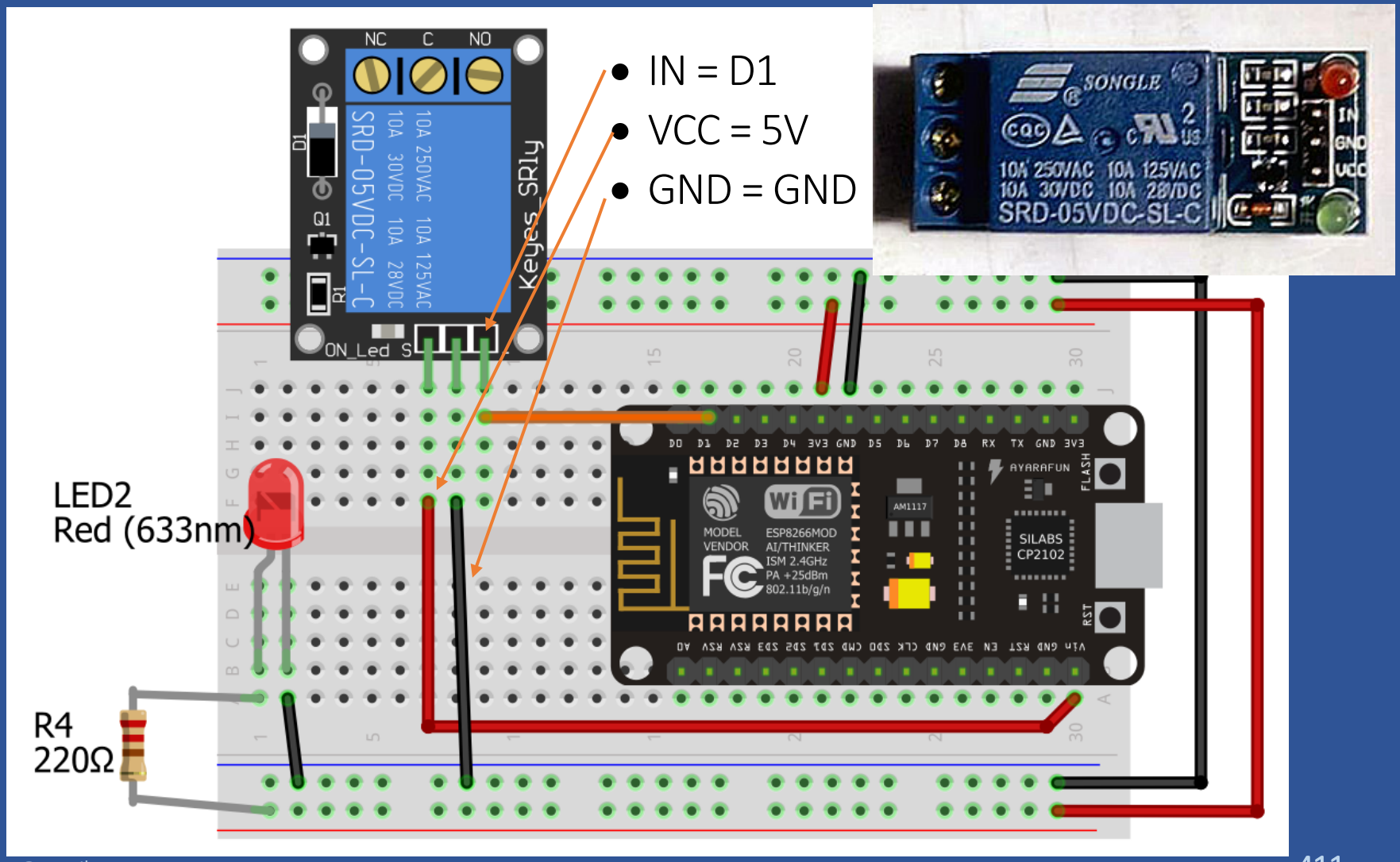


- SIG = D1
- VCC = 5V
- GND=GND

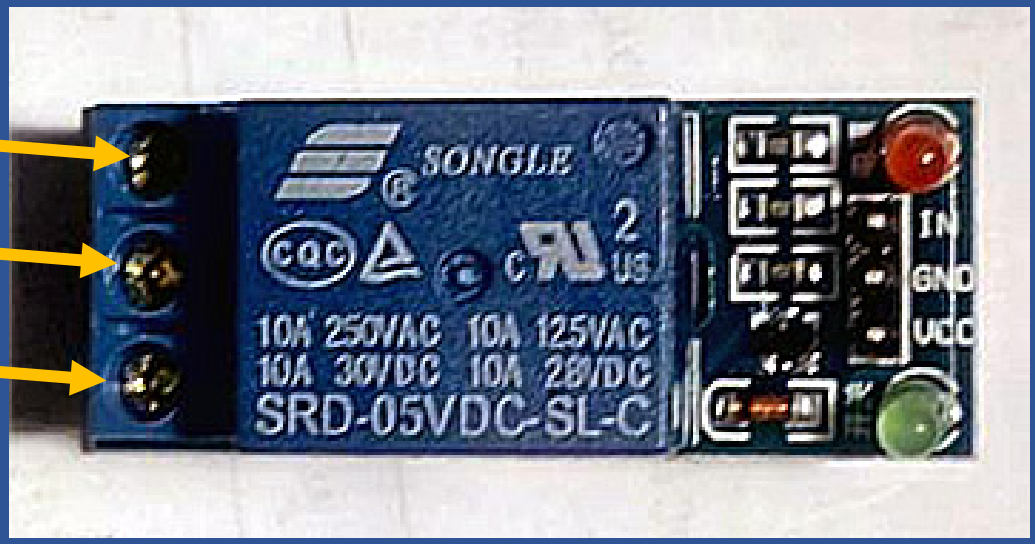
Buzzer Lab

1. Send SOS Morse code very 5 sec
2. Alarm when the temperature reach the certain point

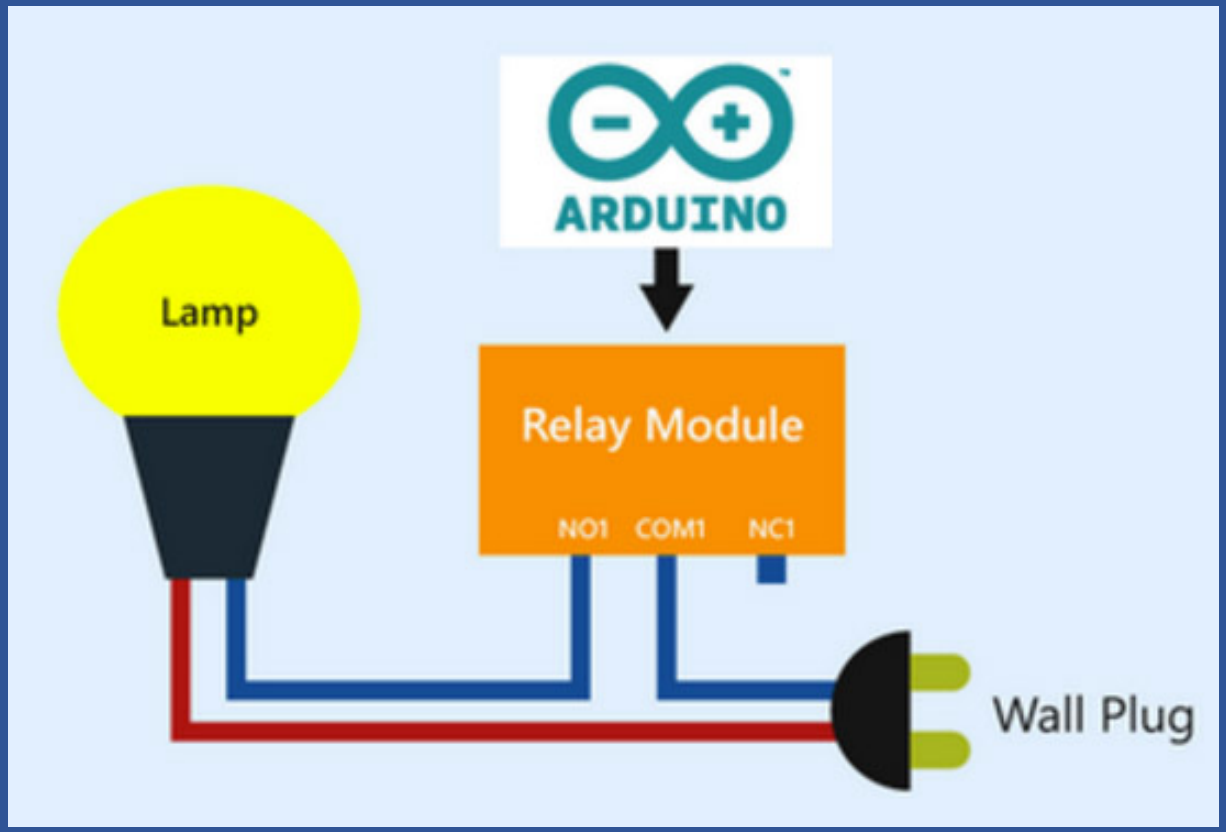
Relay (Electric control)



Electrical hazard



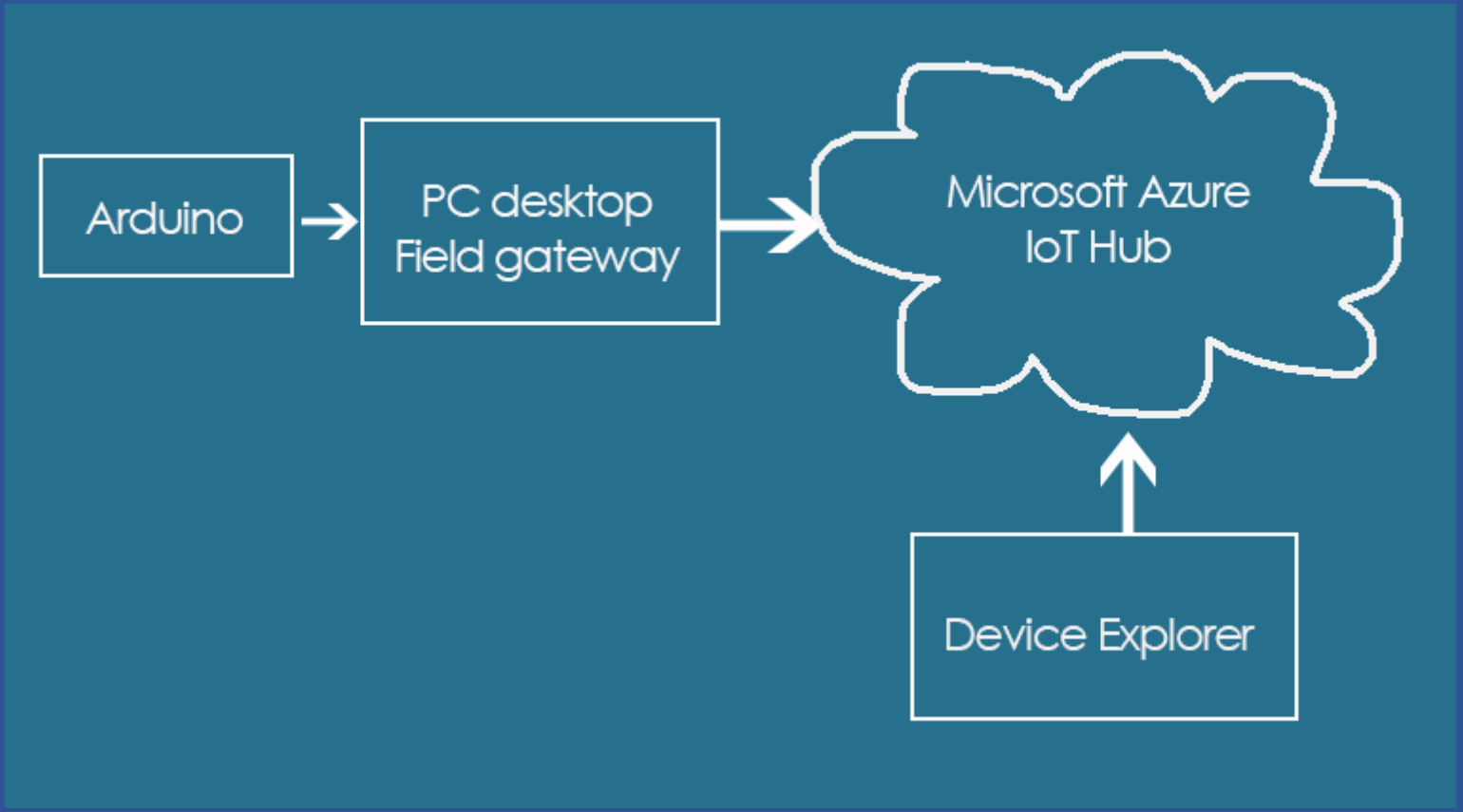
Arduino electric control schematic diagram



Relay Lab

1. Turn the light on when it is dark
2. Turn on the coffee machine at 7:30 AM

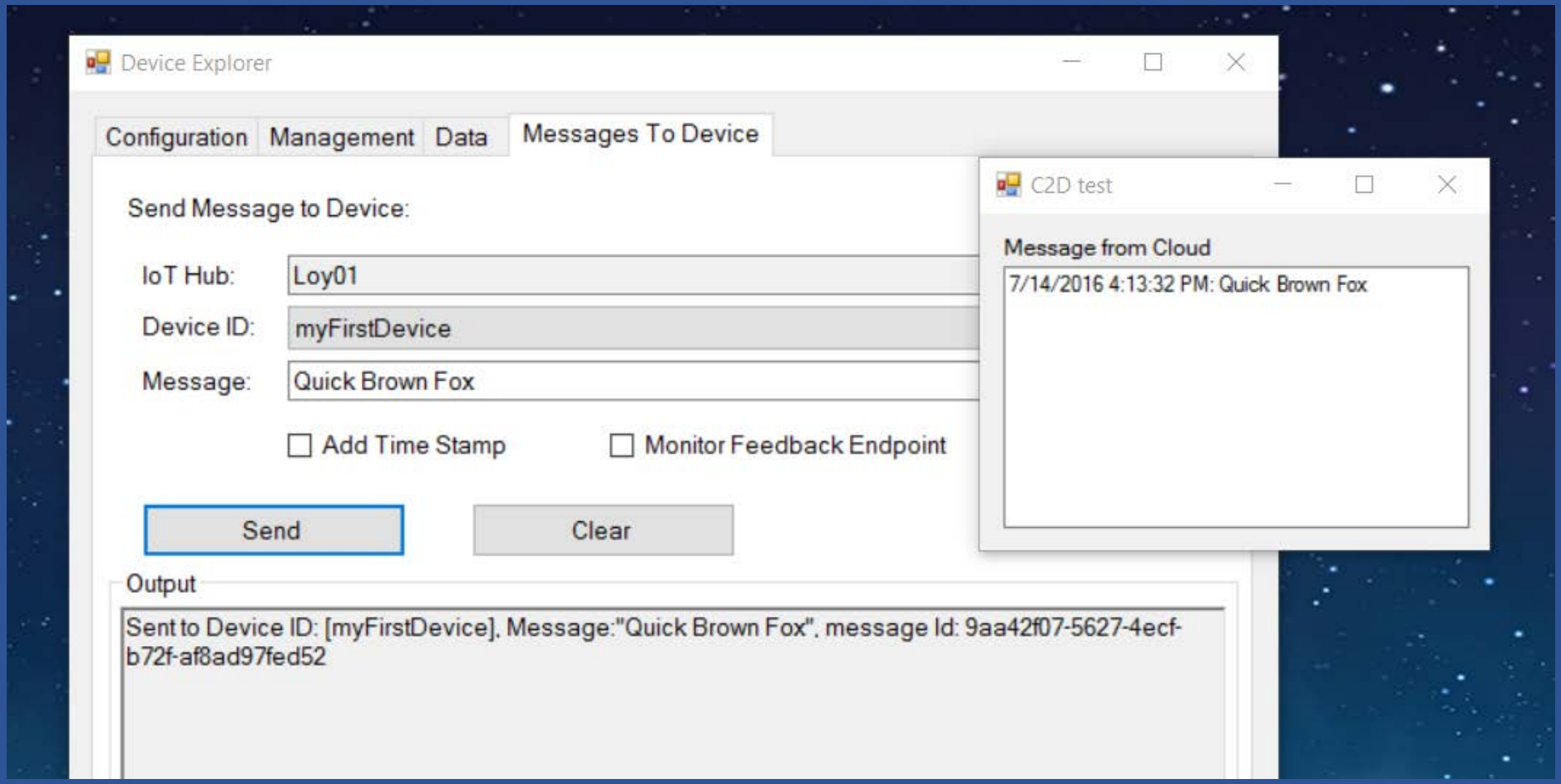
Sending command from Cloud to Device



Receive message from Cloud

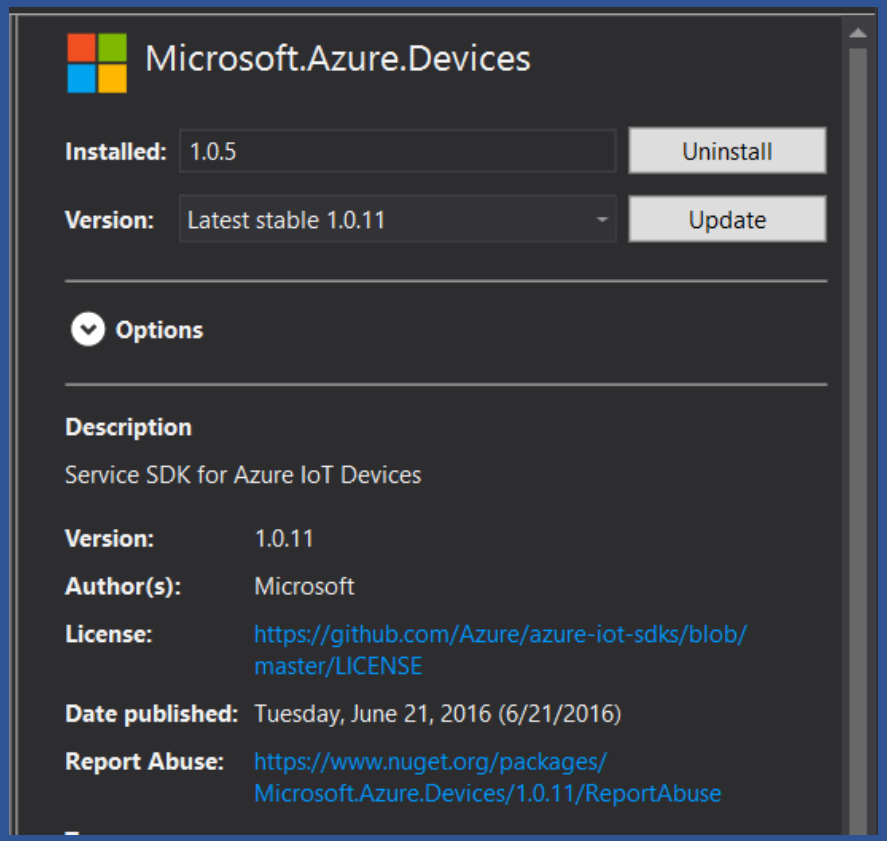
1. Create C# WinApp
2. Add Nuget “Microsoft.Azure.Devices.Client” package
3. Add TextBox
4. Add code
5. Use Device Explorer to send test message
6. If receive message “On” turn on LED
7. When receive message “Off” turn on LED

Testing Receive message from Cloud

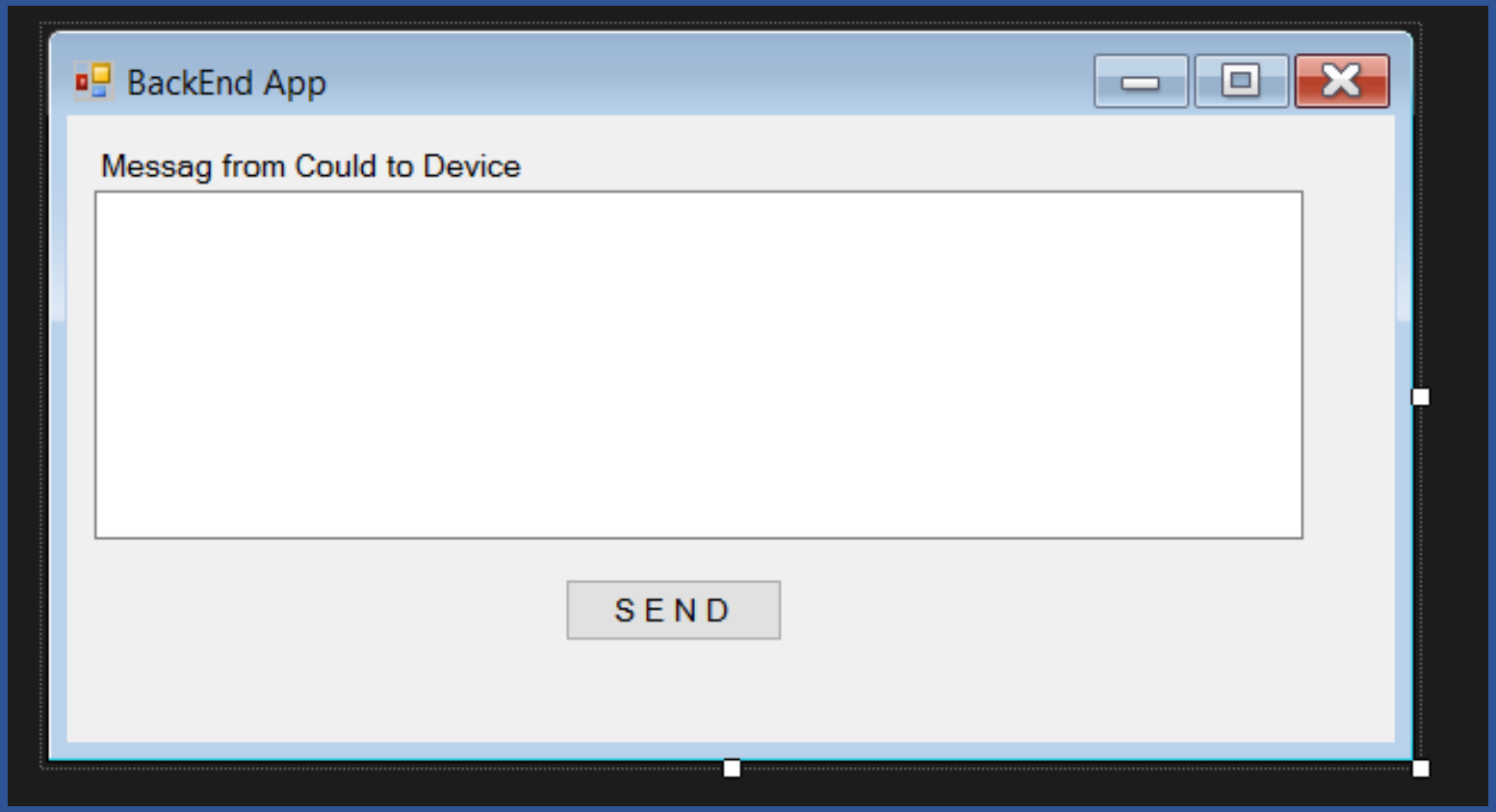


Create C# App to send C2D message

1. Create C# WinForm
2. Add NuGet “Microsoft Azure Devices” package



3.Add TextBox name = “textBoxC2D” to Form1 and a button



4.Add Name space to Form1

```
using Microsoft.Azure.Devices;
using System;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

5.Add Class fields and method SendCloudToDeviceMessageAsync()

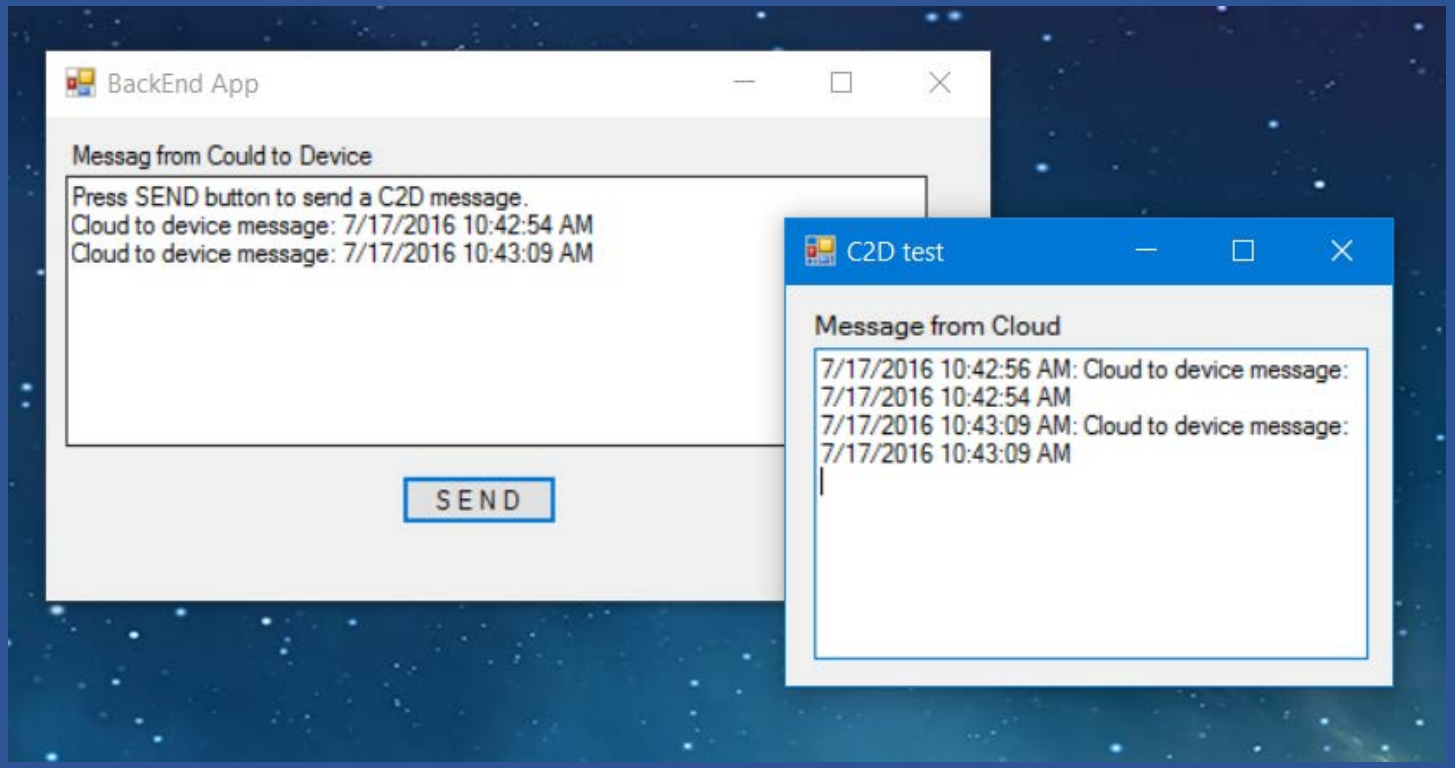
```
public partial class Form1 : Form
{
    ServiceClient serviceClient;
    string connectionString = "HostName=Loy01.azure-devices.net;SharedAccessKeyN
    const string commandMessage = "Cloud to device message: ";
    const string deviceName = "myFirstDevice";

    1 reference
    public Form1()
    {
        InitializeComponent();
    }
    1 reference
    private async Task SendCloudToDeviceMessageAsync()
    {
        string message = "Cloud to device message: " + DateTime.Now;
        var commandMessage = new az.Message(Encoding.ASCII.GetBytes(message));
        commandMessage.Ack = DeliveryAcknowledgement.Full;
        await serviceClient.SendAsync(deviceName, commandMessage);
        textBoxCouldTX.Invoke(new Action(() =>
        { textBoxCouldTX.AppendText(message + "\r\n"); }));
    }
```

6. Add Form_Load and buttonSend_Click code

```
private void Form1_Load(object sender, EventArgs e)
{
    serviceClient = ServiceClient.CreateFromConnectionString(connectionString);
    string msg = "Press SEND button to send a C2D message.";
    textBoxCouldTX.Invoke(new Action(() =>
    { textBoxCouldTX.AppendText(msg + "\r\n"); }));
}
1 reference
private void buttonSend_Click(object sender, EventArgs e)
{
    SendCloudToDeviceMessageAsync().Wait(1);
}
```

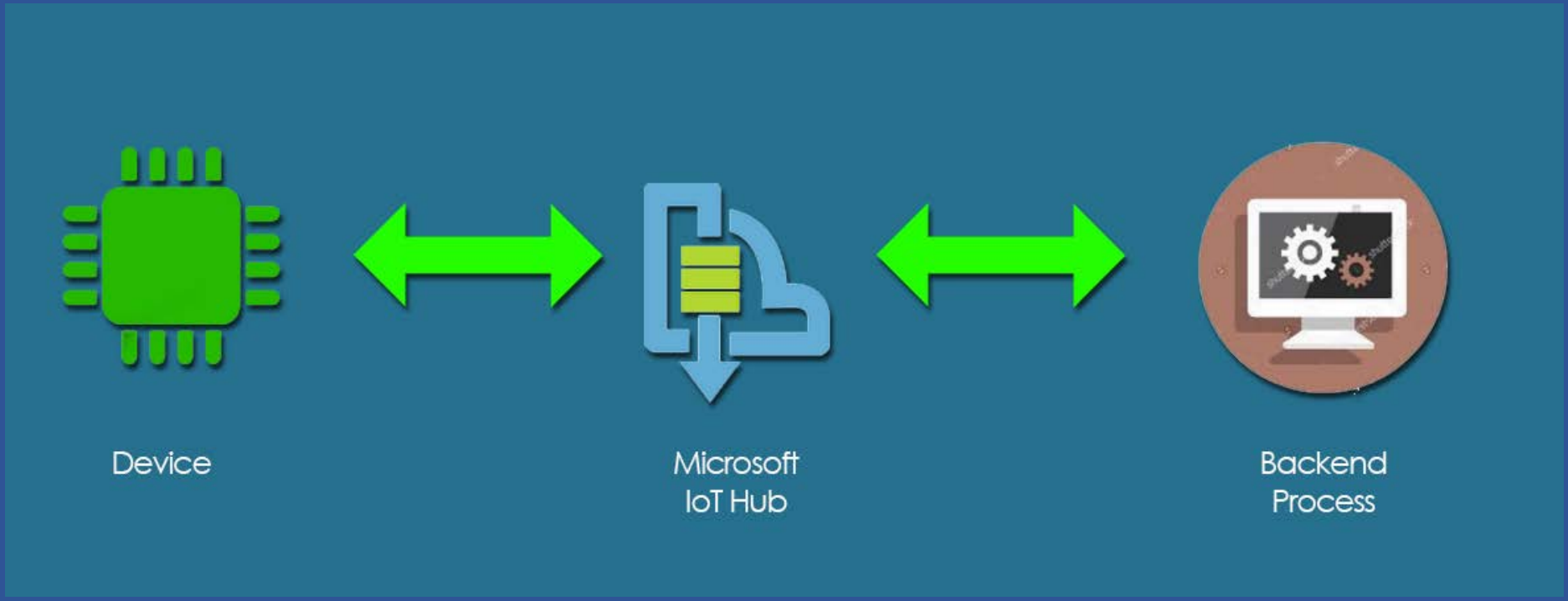
7. Test Program
    a. Run Program Receive message from Cloud (p415)
    b. Run Program send C2D (p417)
    c. Click send button



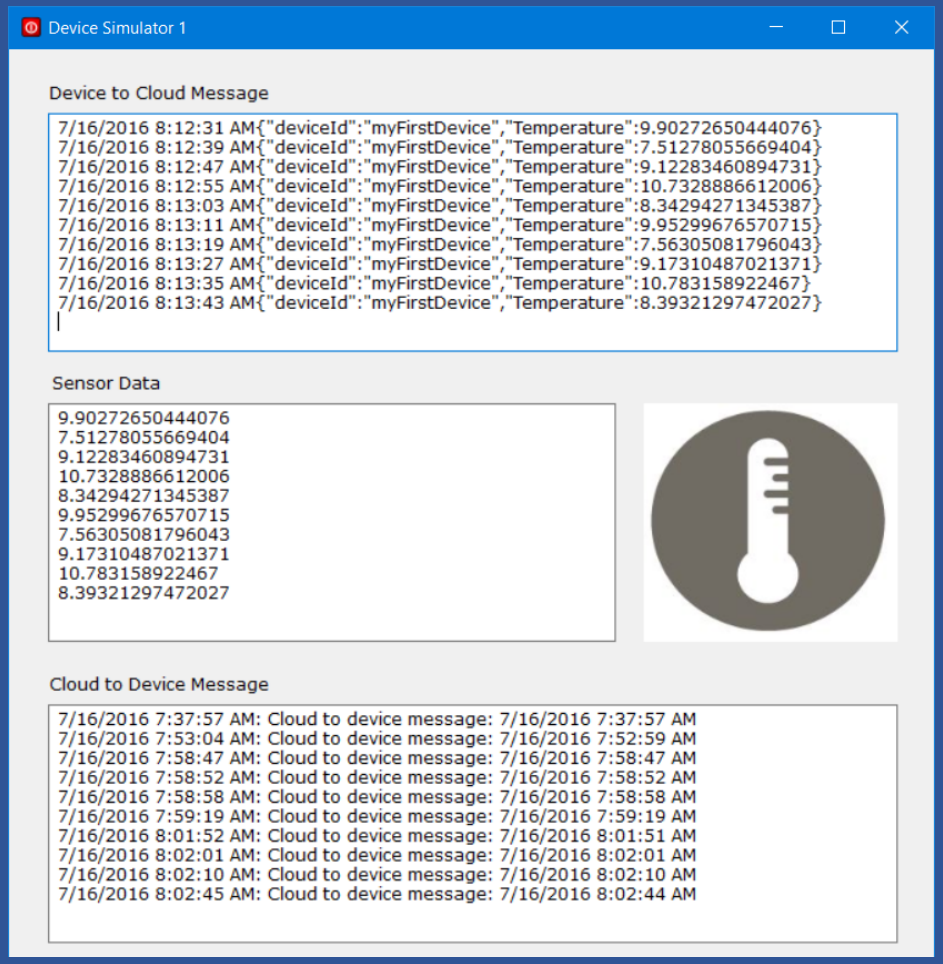
C2D Lab

1. Send message Normal, Warning or Alarm to device from cloud. Device shows status using LED normal = green, alarm = yellow, alarm = red
2. Send command from cloud to turn on the Fan when Temperature reaches a certain point
3. Send command from cloud to Buzz Alarm when detect a movement

Responding to acknowledgement



Run device simulator



Create and Run Backend App

(modify from p417 WinApp)