Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Engenharia Elétrica
ELE0629 – Tópicos Especiais em Sistemas Embarcados – 2020.2

Projeto de aplicação em IoT

Pedro Henrique de Souza Fonsêca dos Santos

Introdução

Com a evolução da Internet das coisas (*Internet of Things* ou simplesmente IoT), os avanços na área da domótica são cada vez mais constantes. Domótica nada mais é que o termo usado para caracterizar a integração dos mecanismos automáticos de um espaço residencial, satisfazendo as necessidades de segurança, conforto e comunicação. Realizando um projeto de domótica, é possível automatizar iluminação, segurança e climatização, além de outros avanças possíveis.

Nesse projeto será mostrada uma aplicação IoT que se utiliza do microcontrolador ESP32, da Espressif, o sensor de temperatura e umidade DHT22, o sensor de luminosidade BH1750 e o sensor de presença PIR HC-SR501. A comunicação com o microcontrolador será feita por meio do aplicativo Blynk pelo Wi-Fi.

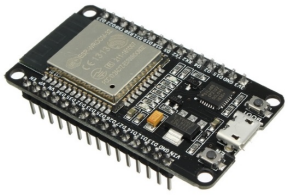

Figura 1: ESP32

Figura 2: Aplicativo Blynk

Desenvolvimento

O projeto foi pensado com as seguintes funcionalidades:

- Controle de lâmpadas pelo aplicativo e pelo interruptor físico. Esse item foi simulado por um botão e um led e sua funcionalidade é parecida com um *three-way* utilizado em instalações elétricas. Ao apertar o interruptor físico ou o do aplicativo, será ligado ou desligado. Uma luz no aplicativo indicará também o estado da luz;
- Controle de um alarme com sensor de presença. Terá um botão para armar o alarme. Enquanto ele estiver armado, checará se o sensor recebe nível alto, indicando que há presença. Quando isso ocorrer, uma luz no aplicativo será ligada;
- Coleta de dados de temperatura, umidade e luminosidade pelo microcontrolador e mostrados no aplicativo.

1 Controle das lâmpadas

O controle das lâmpadas será feito de uma forma a lembrar o funcionamento do interruptor *three-way* feito em instalações residências. Uma ativação no interruptor do aplicativo será sentida e mudará o estado de uma das variáveis que controlam a lâmpada. A outra variável será alterada quando uma ativação no interruptor físico for feita. Dessa forma, ao alterar qualquer uma das duas, o estado será alterado entre ligado ou desligado. Um led no aplicativo indicará o estado atual da iluminação.

2 Controle do alarme

O funcionamento do alarme dependerá inicialmente da sua ativação. Ele só será armado quando o utilizador desejar. Haverá um botão no aplicativo para isso. Enquanto o alarme não estiver armado, ele ignorará os dados vindos do sensor de presença. Já quando ele estiver ligado, haverá um led no aplicativo que será acionado quando o sensor de presença enviar o valor alto, indicando presença no local.

3 Coleta de dados

A coleta de dados é feita de forma bem simples. Os sensores utilizados são todos digitais, não necessitando do conversor analógico-digital do microcontrolador. Cada um deles tem sua própria API na IDE utilizada e basta definí-los para coletar os dados. O DHT22 enviará os dados de temperatura e umidade, utilizando comunicação serial One Wire, enquanto o BH1750 enviará o de luminosidade, utilizando I^2C. Os valores serão exibidos em displays no aplicativo.

4 Blynk

O Blynk é um aplicativo utilizado para projetos que utilizem microcontroladores conectados a um servidor MQTT. Ao utilizá-lo, ele gerará um token para ser utilizado no código do projeto e nos dará várias opções de Widgets para utilizar. Widgets são elementos de interações, como botões, displays, gráficos. A API dentro da IDE utilizada faz toda a conexão com a internet e com o servidor automaticamente, ficando apenas o trabalho de escolhermos quais funções de cada Widget.

5 Código completo

O código completo se encontra a seguir:

```
#define BLYNK_PRINT Serial
#define DHTPIN 15
#define DHTTYPE DHT22
#define BUTTON 5
#define PIR 4
#define LED 2

#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <Wire.h>
#include <BH1750.h>

// Definicao dos dados
BlynkTimer timer;
WidgetLED led1(V2);
WidgetLED led2(V3);
DHT dht(DHTPIN, DHTTYPE);
BH1750 lightMeter;

// Definicoes das configuracoes da internet
char auth[] = "TOKEN_DO_BLYNK";
char ssid[] = "NOME_DA_REDE";
char pass[] = "SENHA_DA_REDE";

int interruptor_blynk = 0;
int alarme_blynk = 0;
int interruptor_fisico = 0;

void sendSensor()
{
  float temp = dht.readTemperature();
  float umid = dht.readHumidity();
  uint16_t lux = lightMeter.readLightLevel();

  if (isnan(temp) || isnan(umid)) {
    return;
  }

  Blynk.virtualWrite(V4, temp);       // Enviar temperatura
  Blynk.virtualWrite(V5, umid);       // Enviar umidade
  Blynk.virtualWrite(V6, lux);        // Enviar luminosidade

  delay(100);
}
```

```
48
49  // Recebendo valor do interruptor
50  BLYNK_WRITE(V0)
51  {
52    if(param.asInt() == 1)
53    {
54      interruptor_blynk = !interruptor_blynk;
55    }
56  }
57
58  // Recebendo valor do alarme
59  BLYNK_WRITE(V1)
60  {
61    alarme_blynk = param.asInt();
62  }
63
64  void setup()
65  {
66    Serial.begin(115200);
67    dht.begin();
68    Wire.begin();
69    lightMeter.begin();
70
71    pinMode(LED, OUTPUT);
72
73    Blynk.begin(auth, ssid, pass);
74    timer.setInterval(1000L, sendSensor);
75  }
76
77  void loop()
78  {
79    Blynk.run();
80    timer.run();
81
82    int presence = digitalRead(PIR);
83
84    // Configuracao da lampada
85    if(digitalRead(BUTTON) == 1)
86    {
87      interruptor_fisico = !interruptor_fisico;
88    }
89
90    if(interruptor_blynk == 0 && interruptor_fisico == 1)
91    {
92      digitalWrite(LED, HIGH);
93      led1.on();
94    }
95    else if(interruptor_blynk == 1 && interruptor_fisico == 0)
96    {
97      digitalWrite(LED, HIGH);
```

```
98      led1.on();
99    }
100   else if(interruptor_blynk == 0 && interruptor_fisico == 0)
101   {
102     digitalWrite(LED, LOW);
103     led1.off();
104   }
105   else if(interruptor_blynk == 1 && interruptor_fisico == 1)
106   {
107     digitalWrite(LED, LOW);
108     led1.off();
109   }
110   // Configuracao do alarme
111
112   if(alarme_blynk == 1 && presence == 1)
113   {
114     led2.on();
115   }
116   else
117   {
118     led2.off();
119   }
120   delay(100);
121 }
```

Resultados e discussão

O projeto se mostrou funcional e útil. Numa aplicação utilizando um relé e uma lâmpada de algum cômodo da casa, ele satisfaria as expectativas. Nas figuras 3 e 4 a seguir é possível ver os estados de quando o led está ligado ou desligado e o alarme desarmado. Esses estados mudam com o acionamento dos dois interruptores. Também é possível ver os dados coletados dos sensores de temperatura, umidade e luminosidade ao lado.

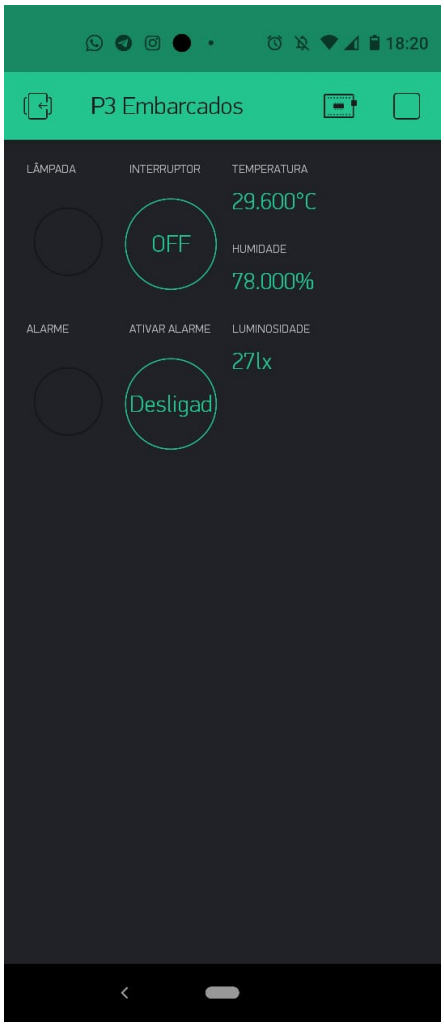

Figura 3: Lâmpada desligada

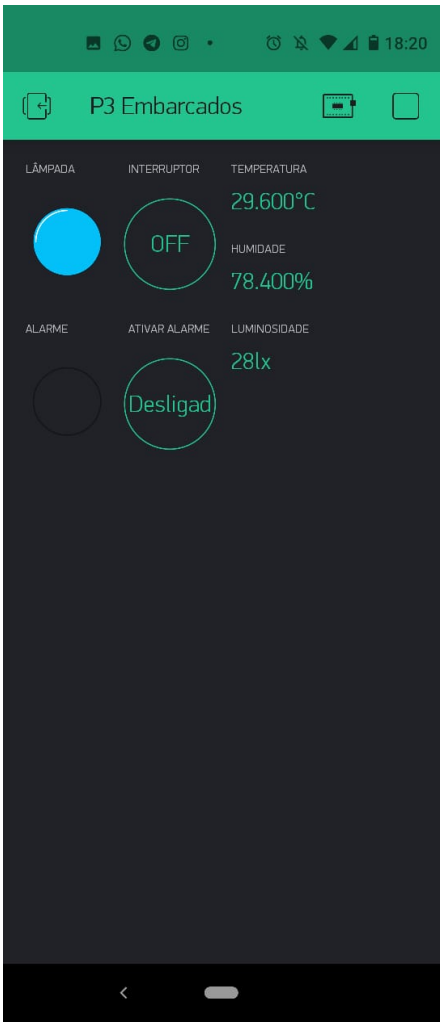

Figura 4: Lâmpada ligada

Nas figuras 5 e 6, é possivel ver os estados de quando o alarme está armado, mas sem sentir presença e quando ele está armado e sentindo a presença, que é o momento em que ele deverá acionar o widget do led.

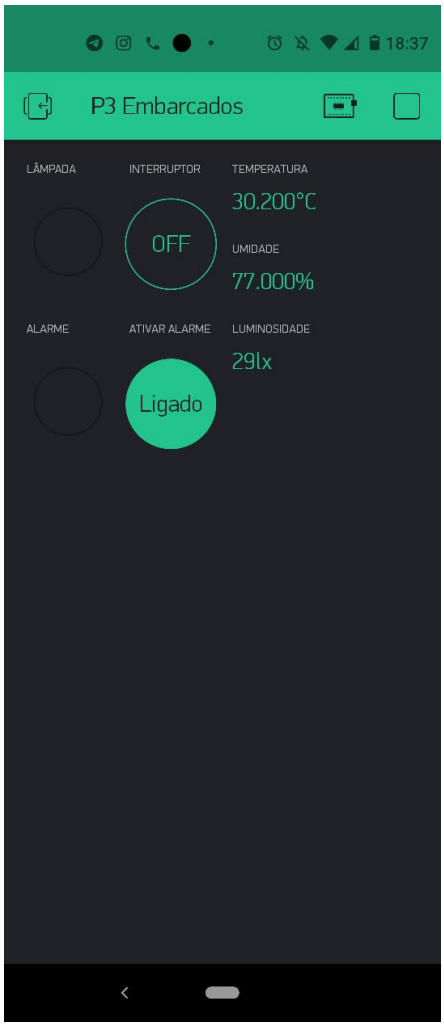

Figura 5: Alarme armado sem presença

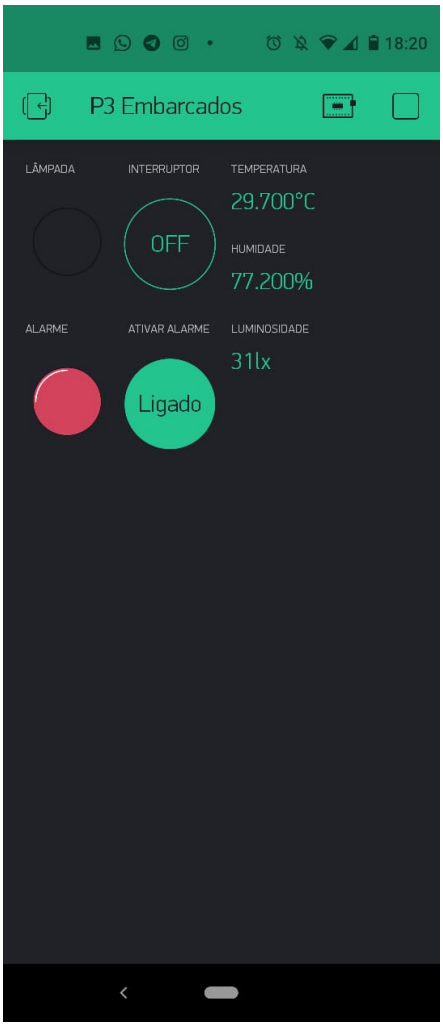

Figura 6: Alarme armado com presença

Na figura 7 é possível ver o protótipo do equipamento. A luz azul do ESP32 foi o que serviu para simular a lâmpada e botão serviu para simular o interruptor. Também é possível ver os sensor de presença, temperatura e umidade, e o de luminosidade.

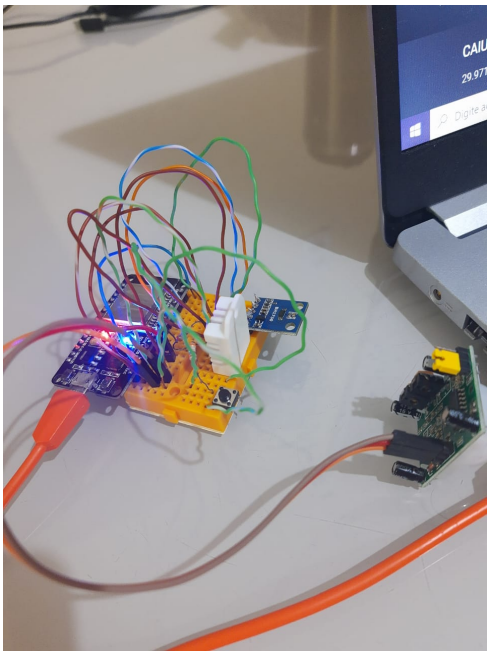

Figura 7: Protótipo do equipamento.

Conclusão

Por fim, foi possível criar uma aplicação IoT de domótica de forma prática com o aplicativo Blynk. Além disso, há várias possibilidades de incremento ao projeto. O aplicativo tem uma certa limitação em relação aos Widgets, mas levando em consideração que seria possível criar várias zonas de alarme e automatizar diversas lâmpadas na casa, ou ainda criar novas funcionalidades, como controle de ar-condicionado, televisão, entre outros.