



Alimentador automático para animais

Pedro Zaffalon da Silva | pedro.zaffalon@uel.br

Universidade Estadual de Londrina

Tutorial de IoT

A área de IoT busca a automação das atividades diárias, através da conexão de objetos cotidianos com a internet. A internet das coisas é um conceito extremamente amplo, devido a grande quantidade de "coisas" que podemos conectar à internet e suas diferentes aplicações. Com isso em mente, esse tutorial busca introduzir esse conceito com um exemplo simples.

O projeto consiste no desenvolvimento de um alimentador automático para animais de estimação e um app para controlá-lo, contendo as seguintes funcionalidades:

- Um botão para despejar a ração;
- Controle da quantidade de ração despejada;
- Mostrar no aplicativo a porcentagem de ração restante no recipiente;
- Notificar quando a ração foi despejada, quando restar menos de 20% da capacidade total e quando ela acabar.

1 Componentes

- ESP8266 NodeMCU;
- Servo Motor;
- Sensor ultrassônico;
- madeira (base da estrutura);
- Galão de água (recipiente para ração);

2 Blynk

Para o desenvolvimento do aplicativo, será utilizado o aplicativo Blynk, que permite a criação de app simples para projeto de IoT e fácil conexão com o hardware.

Depois de baixá-lo será necessário cadastrar-se e criar um novo projeto. Durante a criação ele vai gerar um token, que será utilizado para a conexão com o ESP8266 NodeMCU.

Após isso, será possível começar a implementação do aplicativo, que é feita adicionando widgets.

3 Conexão com o NodeMCU

Resumidamente, será necessário instalar as bibliotecas do Blynk e do ESP8266 na IDE do Arduino, além do seguinte código:

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
}
```

A conexão será realizada no setup, utilizando as variáveis ssid, pass e auth, respectivamente o ssid e a senha da rede wifi na qual o NodeMCU será conectada e o token gerado pelo Blynk durante a criação do projeto.

Assista este tutorial para uma explicação mais detalhada:
<https://www.youtube.com/watch?v=FhS44hGk1Lc>

4 Setup e loop

Nesta parte será utilizado o seguinte código:

```
BlynkTimer timer;

void setup()
{
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
}
```

```
timer.setInterval(intervalo, evento);  
}  
  
void loop()  
{  
  Blynk.run();  
  timer.run();  
}
```

O `Blynk.begin` é utilizado para a conexão, conforme explicado anteriormente, enquanto na próxima linha, irá definir que o timer declarado no começo do código irá realizar a função evento a cada intervalo. Nesta função deve estar todo o funcionamento do alimentador.

Por fim no loop as funções do Blynk e o timer são iniciados.

5 Utilizando o Blynk

Para cada widget adicionada, é possível conectar ele não só à um pin analógico ou digital do NodeMCU, mas também com um virtual pin.

5.1 Virtual pin

O virtual pin não é um pin físico, e tem como função a troca de informações entre o aplicativo e o hardware, o que será útil para lidar com os dados do sensor e com o servo.

Para acessar informações do aplicativo é utilizada a função `BLYNK_WRITE`:

```
BLYNK_WRITE(IdDoVirtualPin){  
  int i = param.asInt();  
}
```

Esta função atribui à variável `i` o valor definido no virtual pin passado como parâmetro. Por padrão o valor de um virtual pin é String, mas é possível convertê-lo para outros tipos, ou tratá-la como um vetor, conforme os exemplos abaixo:

```
param.asFloat();  
param.asDouble();  
param.asStr();  
param[0].asInt();
```

A função `BLYNK_WRITE` é acionada toda vez que o valor do virtual pin for atualizado. É importante ter cuidado com essa função, pois ela deve estar fora de loops e outras funções.

Para enviar valores ao aplicativo é utilizada a função:

```
Blynk.virtualWrite(IdDoDigitalPin, valor);
```

5.2 Widgets

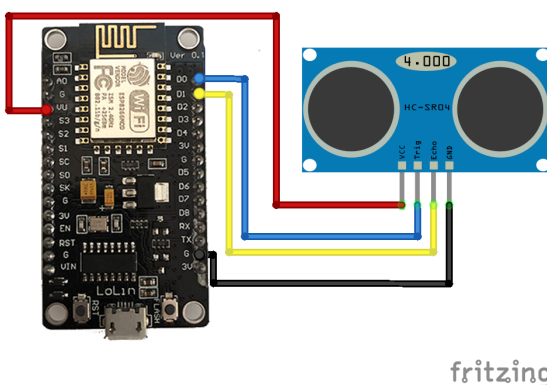
Widgets que podem ser úteis para esse projeto:

- Button: para despejar ração;
- Value Display: para informar a porcentagem da ração restante;
- Numerical input: para informar por quantos segundos a ração será despejada (controle da quantidade de ração);
- Push notification: Permite fazer notificações.

Para fazer notificações utilizá-se o comando:

```
Blynk.notify("texto da notificação");
```

6 Sensor



fritzing

Para utilizar o sensor é necessária a biblioteca `<Ultrasonic.h>`, que pode ser baixada no link: <https://github.com/flipeflop/Ultrasonic>.

Código para utilizar o sensor para calcular a porcentagem de ração restante:

```
Ultrasonic sensor(TRIGGER_PIN, ECHO_PIN);  
  
float porcentagem(){  
    float restante = tamanhoRecipiente - sensor.Ranging(CM);  
    return restante*100/tamanhoRecipiente;  
}
```

Neste código uma variável para o sensor é declarada, e com ela, a distância até a ração é calculada (considerando que o sensor está no topo do recipiente). Utilizando esta distância e o tamanho do recipiente, a porcentagem de ração restante é calculada.

7 Servo

Para utilizar o servo é necessário utilizar a biblioteca <Servo.h>.
Exemplo da utilização do servo:

```
Servo servo1;

void colocarRacao(int tempo){
  servo1.write(90);
  delay(tempo*1000);
  servo1.write(0);
}

void setup()
{
  servo1.attach(pin);
}
```

A função colocarRacao inicialmente define o ângulo do servo para 90^o, permitindo o despejo de ração, e depois de uma quantidade de segundos, definida pela variável tempo, fechava, voltando para 0^o.

8 Estrutura



A estrutura será similar a da imagem acima, porém mais simples, com a base feita com madeira e utilizando o galão de água como recipiente em cima. O sensor ficará no topo do galão e o servo controlando a saída de ração.



Referências

- [1] <http://docs.blynk.cc>
- [2] <https://medium.com/swlh/real-time-iot-app-with-react-firebase-esp8266-e2e673efd1bf>
- [3] <https://www.petlove.com.br/cachorro/acessorios-para-alimentacao/alimentadores-e-bebedouros-automaticos>