

Desenvolvimento de dispositivo de sensoramento IoT

Francisco Wesley Melo Silva¹, Mateus Alves Malveira², Augusto Coelho Melo³

¹Campus Sobral – Universidade Federal do Ceará (UFC)
Rua Estanislau Frota – 62.010-720 – Sobral – CE – Brasil

wesleymelo@alu.ufc.br, mateusmalveira@alu.ufc.br, augusto.baden@gmail.com

Introdução

O presente trabalho mostra a elaboração de um sistema IoT (*Internet of Things*) capaz de fazer a leitura de 3 sensores, recebendo quatro tipos distintos de dados. Existem diversos serviços em nuvem capazes de acompanhar e gerenciar esse tipo de sistema, servidores e clientes *Web* como o ThingSpeak e o aplicativo *Blynk* são utilizados para receber valores e acionar e desligar cada sensor, bem como armazenamento dos dados transmitidos.

Metodologia

Primeiramente se deve ressaltar que foi utilizada a plataforma de programação (IDE) Arduino para codificar as rotinas do projeto. É importante que o NodeMCU esteja desconectado dos sensores no momento da gravação do arquivo à placa. Alguns sensores podem enviar níveis lógicos positivos para certas GPIO do ESP8266 e a mesma pode impedir a gravação do *sketch* quando se encontra em nível lógico diferente de 0 (zero).

Antes de efetuar a gravação do código no microcontrolador, é necessária uma breve configuração no serviço *ThingSpeak*, um aplicativo e API de Internet of Things (IoT) de código aberto para armazenar e recuperar dados de objetos (sensores) usando o protocolo HTTP pela Internet ou através de rede local. O ThingSpeak permite a criação de aplicativos de registro de sensores, rastreamento de localização e uma rede de objetos com atualizações de status. Ao acessar esse serviço *online*, o usuário deve fazer seu cadastro e escolher o plano de pagamento, embora a versão gratuita do serviço seja suficiente para a realização deste projeto.

Deve-se criar um novo “canal”, e nas definições dele são definidos campos de entradas dos dados (é possível cadastrar até 8 (oito) entradas). Neste projeto foram utilizados 4 (quatro) canais de dados, os valores obtidos podem ser conferidos no seguinte endereço: <https://thingspeak.com/channels/517629>. O primeiro teste de funcionamento do projeto se deu ao constatar que o ThingSpeak estava recebendo os valores dos sensores locais.

Após essa etapa, não foi mais necessária qualquer modificação no *hardware* do projeto.

O passo seguinte foi a utilização da plataforma *Blynk* que permite a leitura dos sensores através de um *smartphone*. Ao fazer o cadastro do aplicativo, o usuário recebe por e-mail um *token* de autenticação para ser incorporado no código que será gravado no microcontrolador. Esse curto processo já é suficiente para que o aplicativo esteja conectado ao NodeMCU, e possa ler os resultados de seus sensores, assim como acionar seus pinos ou desativá-los.

Basta que o usuário selecione o *widget* que deseja utilizar, por exemplo uma “tela” virtual para a leitura dos dados de determinado pino. É possível a adição de *widgets* que executam funções como notificação, em resposta a um evento detectado pelo sensor. Uma função importante são seus pinos virtuais, que enviam valores para variáveis predeterminadas no NodeMCU, podendo ser usadas para acionamentos, ativação, etc.

Neste projeto foram utilizados pinos virtuais para enviar valores *booleanos* para o microcontrolador e assim ativar ou desativar os sensores individualmente. Outra aplicação dos pinos virtuais que foi utilizada neste projeto foi que, os sensores digitais que recebem mais de um dado, como o DHT-11 que afere temperatura e humidade, utilizando um pino virtual e encaminhando os valores individualmente para um dispositivo de saída. Um ponto que se deve prestar atenção é que o Blynk não apaga os resultados das leituras anteriores antes de começar uma nova, mesmo que o sistema seja desligado.

Ao longo do desenvolvimento do código, foi utilizado o monitor serial da IDE, para verificar os valores de variáveis, *status* de conexão, instruções recebidas, etc.

Materiais utilizados

- 1 Resistor de 260 Ω ;
- 1 Resistor de 180 Ω ;
- Sensor LDR (cuja resistência varia de acordo com a luminosidade);
- NodeMCU (ESP8266) da marca LoLin;
- LED de cor vermelha;
- 2 *Protoboards* de 800 furos;
- 11 *Jumpers* machos;
- Sensor DHT - 11;
- Sensor HC-SR04;
- Cabo USB (*Univesal Serial Bus*).

Resultados

Foram obtidos os resultados que eram esperados durante a elaboração do projeto. O NodeMCU foi capaz de fazer leituras dos sensores e enviá-las para plataformas *online*. Também foi possível ler os dados obtidos dos sensores através de um *smartphone* e controlar o funcionamento dos mesmos através do mesmo aplicativo. A seguir estão dispostas algumas imagens do funcionamento do projeto e o link para um vídeo que demonstra o seu funcionamento.

Imagens

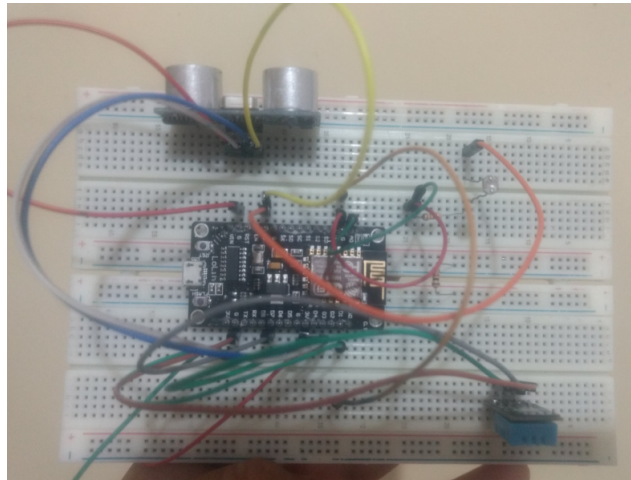


Figure 1. Imagem das *protoboards* com todo o componentes.



Figure 2. Vista do sensor de distância



Figure 3. Vista do sensor de temperatura e humidade

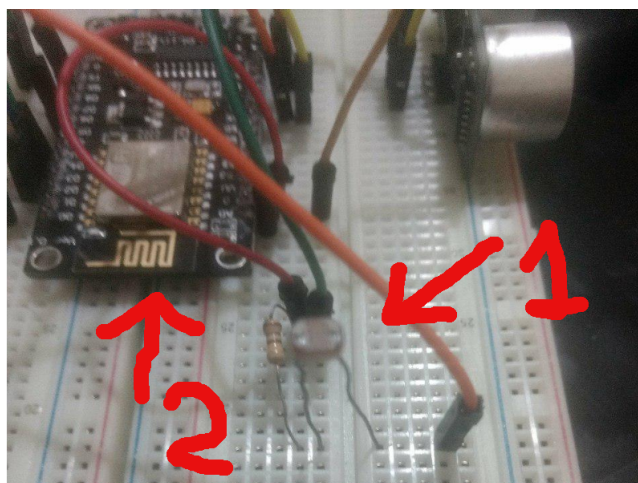


Figure 4. Vista do sensor LDR (luminosidade) (1) e do NodeMCU (2).



Figure 5. Captura de tela do smartphone mostrando valores aferidos através de *mini-displays* e com os botões de ativação e desativação dos sensores.

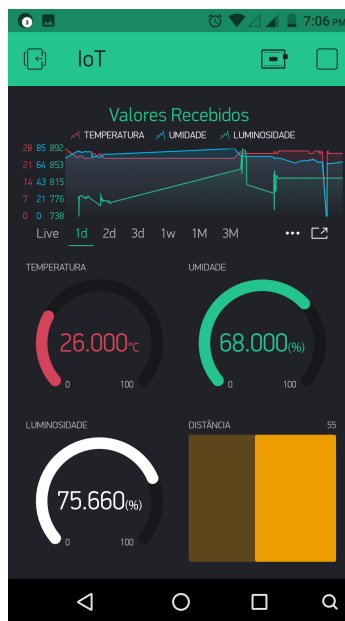


Figure 6. Screenshot do aplicativo Blynk mostrando os dados aferidos através de ponteiros, gráficos e barra.

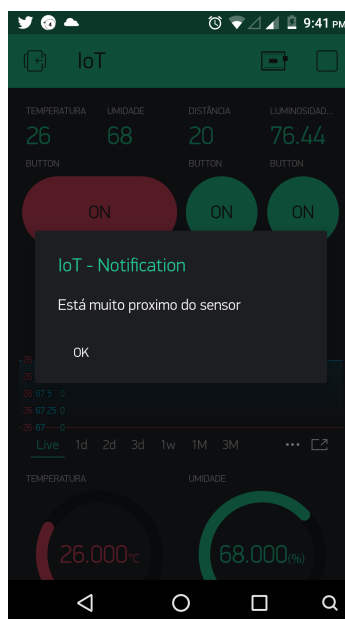


Figure 7. Notificação do aplicativo Blynk após o sensor detectar uma condição especificada, acionada pelo sensor de distância.

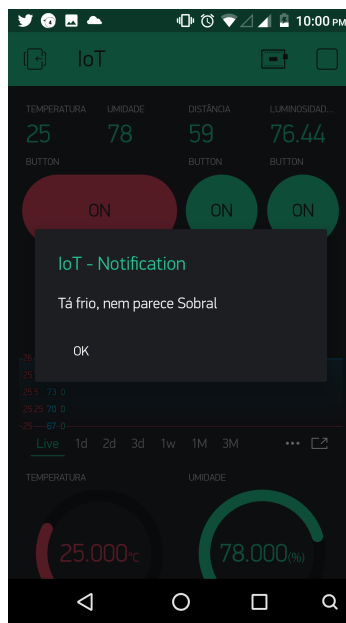


Figure 8. Outro exemplo de notificação, aciado por um evento no sensor de temperatura.

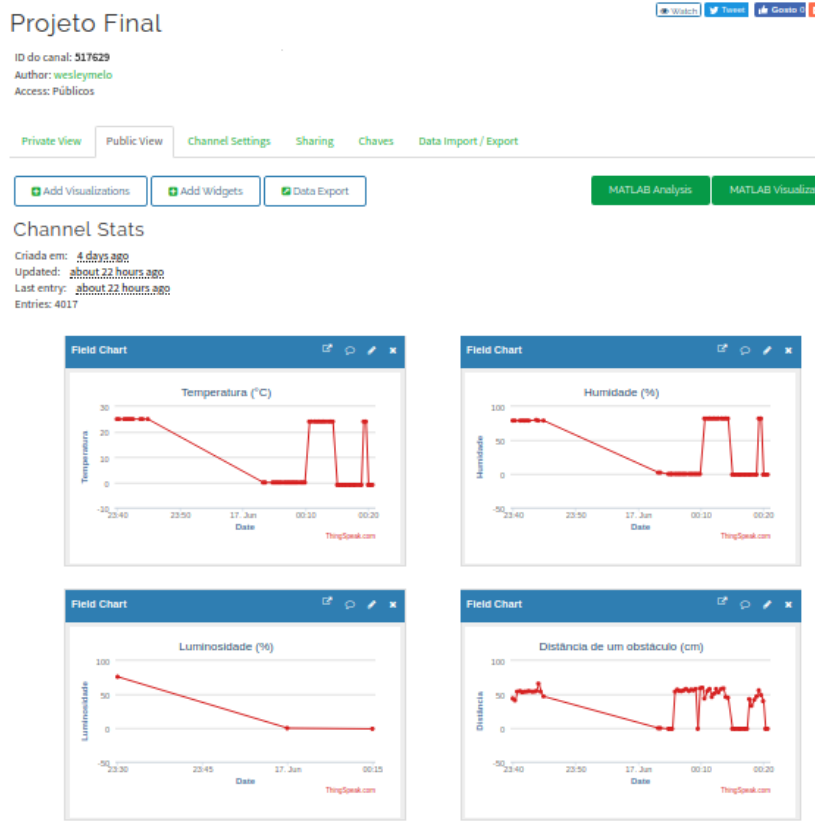


Figure 9. Screenshot do site ThingSpeak, mostrando os valores obtidos através dos sensores

Video

O vídeo do link a seguir mostra o funcionamento do sistema.

<https://youtu.be/kkukCwKl6qg>

Código-fonte

O código-fonte está escrito na linguagem C++, embora a plataforma de desenvolvimento do NodeMCU recomende que se utilize a linguagem *Lua*, foi levado em consideração a compatibilidade com o grande número de bibliotecas da plataforma Arduino que podem ser utilizadas no NodeMCU (ESP8266).

O arquivo “projetofinal_ino” é o que deve ser gravado no microcontrolador. No mesmo repositório estão disponíveis as bibliotecas que foram importadas ao projeto para utilização dos sensores e do aplicativo *Blynk*.

Repósitório de código

O código-fonte utilizado neste projeto está disponível em: <https://www.github.com/fwesleyms/projetofinaltpcs2> e encontra-se sob licença AGPL 3.

Divisão do trabalho entre a equipe

Table 1. Divisão de tarefas para o desenvolvimento do projeto.

	Wesley Melo	Mateus Malveira	Augusto Coelho
<i>Brainstorm</i>	X	X	X
Montagem do Circuito			X
Escrita do Código	X		
Integração ao <i>ThingSpeak</i>		X	
Integração do <i>Blynk</i>	X		
Registro de imagens			X
Registro em vídeo	X		
Escrita do Relatório	X	X	X
Montagem da maquete	X	X	X

References

- ANIPINDI, K. (2014). *An Introduction to ThingSpeak*. code-project.com, 15th edition.
- Arduino.CC (2018). *Arduino: Getting Started with Arduino and Genuino products*. Arduino.CC.
- Blynk (2018). *Blynk Documents*. Blynk LLC.
- Mati, D. (2013). *Indoor environmental quality station + Bluetooth + ThingSpeak*. Instructables.