Sistema de Automação utilizando dispositivos micro controladores, módulo solar e plataforma de Internet das Coisas (IoT)

Automation System using micro controlled devices, solar module and Internet of Things Platform

Jonathan Galdino da Silva
Jonathansilva.2016@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Santa Helena, Paraná, Brasil
Itamar Iliuk
itamariliuk@utfpr.edu.br
Universidade Tecnológica Federal do paraná, Santa Helena, Paraná, Brasill

RESUMO

Este trabalho tem como objetivo descrever a construção de um sistema de automação utilizando um módulo solar e a placa de desenvolvimento NodeMCU ESP12. O micro controlador ESP12 irá obter os dados dos sensores de temperatura, umidade e umidade do solo, a cada 60 minutos, e enviar para a plataforma IoT construída no Node-RED. Após os dados serem enviados, o micro controlador entra em modo de economia de energia. A energia do sistema é fornecida por um módulo solar que irá carregar uma bateria, em caso de falta de energia por parte do módulo solar a bateria irá fornecer a energia para o sistema continuar rodando sem que ocorra perda de dados. A plataforma irá receber os dados, obter o horário correto de quando a mensagem foi recebida e, ao mesmo tempo, enviar para o banco de dados e para o dashboard. No dashboard, é possível visualizar os últimos dados recebidos em forma única e em gráficos.
PALAVRAS-CHAVE: Automação. Plataforma IoT. Módulo solar. Node-RED.

ABSTRACT

This paper aim to describe the construction of na automation system using a solar module and the NodeMCU ESP12 development board. The ESP12 micro controller will obtain the temperature, humidity and soil moisture from the sensors every 60 minutes and send it to the IoT platform built in the Node-RED. After the data is submitted the micro controller goes into power saving mode. The power of the system is provided by a solar module that will charge a battery, in case of a power failure by the solar module the battery will provide the power for the system to continue running without loss of data. The plataforma will receive the data, get the correct time when the message was received, and at the same time send it to the database and to the dashboard. In the dashboard, its possible to view the latest received data in a gauge and in charts.
KEYWORDS: Automation. IoT Plataform. Solar Module. Node-RED.

Recebido: 30 ago. 2018.
Aprovado: 04 out. 2018.

Direito autoral:

Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

INTRODUÇÃO

Este trabalho descreve um sistema de automação que foi criado utilizando a placa de desenvolvimento NodeMCU, um módulo solar e sensores. A energia do sistema é fornecida pelo módulo solar que irá carregar uma bateria de Li-Ion. Utilizando o carregador de baterias TP4056, é possível carregar a célula e alimentar o sistema ao mesmo tempo. O sistema se comunica diretamente com uma plataforma IoT (acrônimo de *Internet of Things*), criada através da ferramenta Node-RED. O sistema irá se comunicar com a plataforma, informando a leitura dos sensores, via protocolo MQTT (acrônimo de *Message Queuing Telemetry Transport*), que é um protocolo de mensagens leve para sensores e pequenos dispositivos móveis. Ao receber os dados, a plataforma IoT irá salvar os dados no banco de dados MongoDB e então, demonstrá-los numa *Dashboard*.

REFERENCIAL TEÓRICO-METODOLÓGICO

Segundo Zambarda (2014), a ideia de conectar objetos é discutida desde 1991, quando a conexão TCP/IP e a internet que conhecemos hoje começou a se popularizar. Em 1999, Kevin Ashton do MIT propôs o termo “Internet das Coisas" e dez anos depois escreveu o artigo “A Coisa da Internet das Coisas” para o RFID Journal. Segundo o autor a “Internet das Coisas” se refere a uma revolução tecnológica que tem como objetivo conectar objetos e equipamentos usados no dia a dia à rede mundial de computadores.

Node-RED é uma ferramenta de programação para Internet das Coisas, que conecta dispositivos de hardware, APIs e serviços online. Node-RED é construído em Node.js e fornece um editor de flows baseado em navegador que facilita a conexão de fluxos usando nós. (NODERED, 2018a). O Node-RED é composto de nodes, cada node tem um propósito, uns irão gerar dados, outros irão fazer alguma manipulação com os dados e outros irão apenas passar esses dados para outros nodes (NODERED, 2018b).

Para exibir as informações, foi utilizado o node-red-dashboard. Este módulo fornece um conjunto de nós no Node-RED para criar rapidamente um painel de dados ao vivo (NODERED, 2018c). O node-red-dashboard foi instalado diretamente pelo pallete do Node-RED. O pallete é uma ferramenta que permite instalar outros nós diretamente da interface do Node-RED.

MongoDB é um banco de dados NoSQL orientado a documentos desenvolvido pela empresa 10gen e lançado no ano de 2009. Foi construído utilizando a linguagem de programação C++ (DEVMEDIA, 2014) e armazena seus dados no formato BSON, que é um JSON em formato binário, porém superior a este quando comparado ao espaço e a velocidade de busca (NAYAK et al., 2013).

Foi utilizado o módulo WiFi ESP8266 NodeMCU, pra enviar dados de temperatura e umidade, utilizando o sensor DHT22O módulo WiFi ESP8266 NodeMCU é uma placa de desenvolvimento que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão 3.3V. O NodeMCU possui antena embutida e conector micro-usb para conexão ao computador, além de 11 pinos de I/O e conversor analógico-digital (FILIPEFLOP, 2018). A programação pode ser

feita usando a linguagem de programação LUA ou a IDE do Arduino, utilizando a comunicação via cabo micro-usb.

O MQTT (YUAN, 2017) foi inventado e desenvolvido inicialmente pela IBM no final dos anos 90. Ele é um protocolo de mensagens com suporte para a comunicação assíncrona entre as partes. O MQTT é um protocolo de aplicação leve, baseado na topologia *publish/subscribe* para envio de mensagens (FIREWARELAB, 2016).

Para coletar os dados de temperatura e umidade do ar, foi utilizado o sensor DHT22 de modelo AM2302. Este sensor consome 1,5mA quando está medindo e 50uA quando está em modo *standby* (AOSONG,2018).

Para coletar a umidade do solo, foi utilizado um sensor que é constituído por duas sondas (hastes de metal) separadas à distância fixa por algum material isolante (GARDENBOT, 2017).

Como fonte de energia do sistema, foi utilizado um módulo solar da Sunny International Power Limited, de modelo SFM5-P. Esse módulo solar tem capacidade de gerar 7V de tensão e 1A de corrente. O módulo opera num intervalo de -40ºC até 80ºC de temperatura.

Para carregar a bateria de Li-Ion, foi utilizado o módulo carregador de baterias TP4056. Este módulo possui proteção contra sobrecarga, isto é, mesmo se a bateria estiver totalmente carregada, o TP4056 para de carregar a bateria além de indicar o através de um *led*.

Uma célula Li-Ion foi utilizada para alimentar o sistema quando não há captação de energia solar. Esta célula foi obtida através da desmontagem de uma bateria de notebook.

Foi utilizado um regulador de tensão LM2596, este regulador é um conversador DC-DC do tipo Step-Down (TEXAS INSTRUMENTS, 2016). A tensão de saída pode ser ajustada através de um *trimpot*. A tensão de entrada deve ser 1,5V maior que a saída.

PROTOTIPAÇÃO

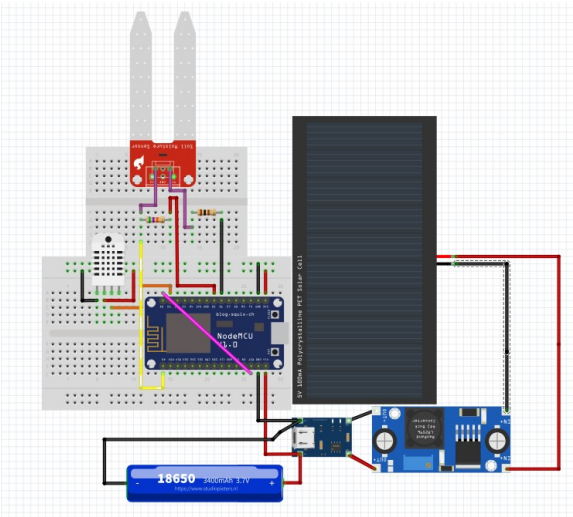
Toda a prototipação do sistema foi realizada em *Protoboards* (Figura 1).

MODO DE ECONOMIA DE ENERGIA

O ESP8266 utiliza o pino Digital 0 para religar o sistema quando o mesmo entra modo de economia *deepSleep*. Sendo assim, foi realizado uma conexão entre o pino D0 e o pino RST (*reset*). Assim, quando o modo de economia estourar o tempo pré-definido por código, o pino D0 irá dar um pulso religando o sistema.

O consumo do sistema durante a execução do código é de 3.80V, enquanto no modo de economia de energia *deepSleep* o consumo é de aproximadamente 0.78V.

Figura 1 – Prototipação do sistema

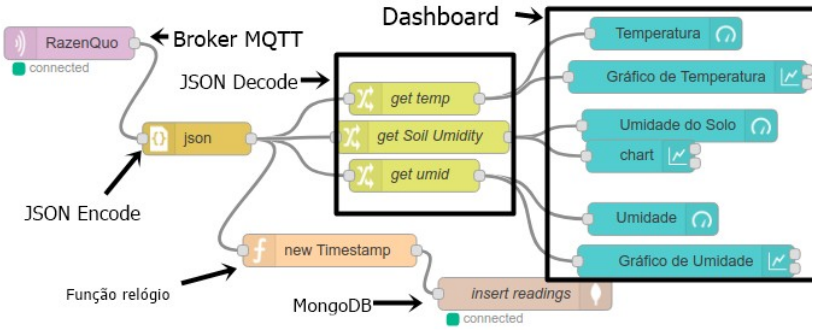


Fonte: Fritzing, 2018.

CRIAÇÃO DA DASHBOARD

Utilizando a ferramenta Node-RED, foi criado um *flow*. O *flow* é o ambiente de desenvolvimento dentro do Node-RED. O *flow* criado neste trabalho é demonstrado na Figura 2 e comentada logo em seguida.

Figura 2 – Exemplo de figura



O *Broker* MQTT irá receber os dados do sistema, em forma de *String*. Utilizando o *node* denominado como “JSON Encode”, essa *String* é transformada em objeto JSON. O *node Function* denominado como “Função relógio”, obtém o horário em que a mesagem chegou e envia para o *node* denominado como “MongoDB”. Este *node* faz a persistência dos dados no banco de dados. Quando a mensagem é transformada em objeto JSON, dois *nodes* extraem os dados (denominado como “JSON Decode”), ao mesmo tempo, e em seguida enviam para os *nodes* (*Gauge* e *Chart)* denominados como “*Dashboard”*.

CONCLUSÃO

O modo de energia de economia *deepSleep* foi o ponto chave nesse trabalho visto que por utilizá-lo, foi possível prolongar a energia na bateria e manter o sistema sempre ligado.

A combinação do módulo solar e o módulo carregador de baterias foi bastante satisfatório, visto que o módulo carregador de baterias faz o chaveamento inteligente e se a bateria estiver totalmente carregada a energia é direcionada para o sistema, reaproveitando a sobra de energia e prolongando a vida útil da bateria.

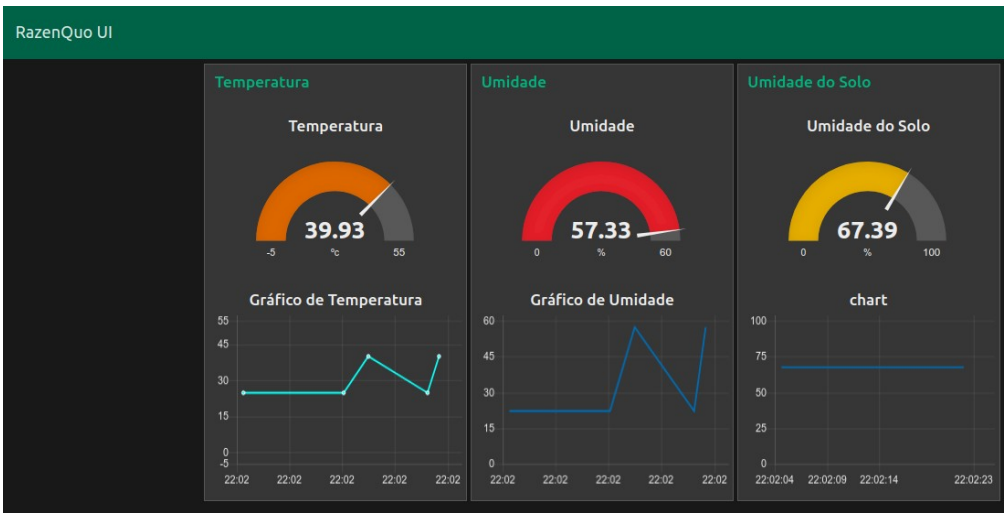
O sistema ficou ligado por um período de 8 horas, com a intenção de validar o sistema. Os seguintes dados (Tabela 1) foram persistidos no banco de dados.

Tabela 1 – Dados coletados pela plataforma

	Dados		
Horário	Temperatura	Umidade	Umidade do Solo
11:27:24	18.7	99.9	87
12:26:18	20.2	99.9	87
13:23:54	21.6	97.3	87
14:22:54	21.3	90.5	87
15:21:59	22.3	91	86
16:20:37	23.1	89.3	86
17:19:34	23.2	94.1	85
18:18:40	23	95.8	85

A criação de uma *dashboard* utilizando o plugin *node-red-dashboard* é bem simples, facilitando o desenvolvimento do trabalho e a mostra dos dados. A *dashboard* pode ser visualizada na Figura 3.

Figura 3 – Exemplo de figura



REFERÊNCIAS

AOSONG. Digital-outpu relatve humidity & temperature sensor/module. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. Acesso em: 28 Ago. 2018.

DEVMEDIA. Introdução ao MongoDB. Disponível em: < https://www.devmedia.com.br/introducao-ao-mongodb/30792>. Acesso em 21 Ago. 2018.

ESPRESSIF. ESP8266 Low Power Solutions. Disponível em: < https://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf>. Acesso em: 29 Ago. 2018.

FIREWARELAB. Apresentação do protocolo MQTT como alternativa para comunicação IoT. Disponível em: < http://fiwarelabsp.org/2016/07/apresentacao-do-protocolo-mqtt/>. Acesso em: 12 maio 2018.

FRITZING. Fritzing. Disponível em: <http://fritzing.org/home/>. Acesso em: 29 Ago. 2018.

GARDENBOT. The Soil Moisture Sensor. Disponível em: < http://gardenbot.org/howTo/soilMoisture/>. Acesso em: 29 Ago. 2018.

NAYAK, A; PORIYA, A; POOJARY, D. Type of NOSQL Databases and its Comparison with Relational Databases. International Journal of Applied Information Systems. v. 5, n. 4, 2013.

NODERED. Flow-based programming for the Internet of Things. 2018a. Disponivel em: <https://nodered.org>. Acesso em: 12 maio 2018.

NODERED. About. 2018b. Disponível em: <https://nodered.org/about/>. Acesso em: 14 maio 2018.

NODERED. Node-red-dashboard. 2018c. Disponível em: <https://flows.nodered.org/node/node-red-dashboard>. Acesso em: 12 maio 2018.

TEXAS INSTRUMENTS. **LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3-A Step-Down Voltage Regulator**. Disponível em: <http://www.ti.com/lit/ds/symlink/lm2596.pdf>. Acesso em: 4 Ago. 2018.

YUAN, M. **Conhecendo o MQTT**. Disponível em <https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>. Acesso em: 27 jun. 2018.

AGRADECIMENTOS

O autor agradece ao orientador do trabalho, por todo o aprendizado e por tê-lo guiado com sabedoria.