

Sistema IoT de Alerta para Medicamentos Controlados

Elizabeth Lucas Bruno, Savio Lopes Rabelo, Sandro César Silveira Jucá

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE),
Eixo Tecnológico da Computação, Campus Maracanaú – CE – Brasil

{bethlbruno, saviorabelo.ti, sandro.juca}@gmail.com

Abstract. *This paper aims to analyze and describe the basic assembly of a prototype for a Alert IoT (Internet of Things) System for usage of Controlled Medicine. The system is composed by three components: the first one is a tabletop container with six drawers, each one with its own medicines and sensors. The second one is a online module which one allows the user to schedule the times he wants to take the medicine kept in a certain drawer. Finally, the third one component is a circuit microcontrolled by an ESP8266 that is able to fire two alarms: a visual one, connected to the drawer whose time the user scheduled in, and a sound alarm in the form of a beep. The proposed system was implemented in a real situation an it works as expected.*

Resumo. *Este artigo tem como objetivo a análise e descrição da montagem básica de um protótipo de um sistema de alerta IoT (Internet of Things) para uso de remédios controlados. O sistema é constituído por três componentes. O primeiro destes é um recipiente de mesa que possui seis gavetas para acomodar as cartelas de remédio de um determinado horário e os sensores de abertura dessas gavetas. O segundo é um formulário online que permite ao usuário preencher os horários que deseja tomar os remédios colocados em determinada gaveta. Finalmente, o terceiro componente é um circuito microcontrolado por um ESP8266 que dispara um sinal sonoro e acende o LED da gaveta cujo horário o usuário configurou para despertar. O sistema proposto foi implementado em uma situação real e funcionou como esperado.*

1. Introdução

O recente aumento na longevidade da vida das pessoas vem trazendo diversas patologias relacionadas com a idade e ao aumento da prevalência das patologias crônicas [dos Santos 2013]. Por isso, o uso de medicamentos controlados tem apresentado uma solução para esse problema. A falta de rigorosidade no horário desses medicamentos pode diminuir a eficácia do tratamento, ocasionar risco de desenvolvimento de complicações, aumentar substancialmente os custos humanos, sociais e econômicos da doença.

Em [Schemberger 2016] são citados dados do IBOPE que mostram que metade da população brasileira não segue à risca o tratamento proposto pelos médicos. Em proposta de solução, esse projeto porposto, de fácil implementação e manuseio, busca o auxílio de cuidadores e pacientes no cumprimento da restrição de horários de consumo dos remédios.

O ESP8266 [Espressif 2018] é um microcontrolador que acrescenta um eficiente suporte para aplicações online utilizando sistemas embarcados. Esse microcontrolador de 32 bits permite a comunicação com a rede sem fio pois possui pilha TCP/IP embarcada.

Os módulos baseados no microcontrolador ESP8266 representam um grande avanço na relação de custo-benefício e podem ser uma ferramenta muito interessante para soluções de Internet das Coisas (do inglês *Internet of Things*, IoT) [de Oliveira 2017]. Várias aplicações com esse dispositivo foram desenvolvidas na área de IoT [Thaker 2016], [Kodali and Soratkal 2016], [Singh and Saikia 2016], [Škraba et al. 2016]. IoT é uma área em ampla expansão que propõe ter dispositivos embarcados conectados à internet e interagindo com o ambiente e também entre outros dispositivos [Schwartz 2016].

2. Desenvolvimento

Nesta seção, são descritos os componentes e programas selecionados para o desenvolvimento do sistema IoT de alerta de medicamentos controlados.

2.1. Formulário Web

A rotina do sistema inicia a partir do preenchimento de um formulário *WEB*. Ao gravar os horários preenchidos pelo usuário, pressionando o botão Gravar Horário, o formulário *HTML* envia os valores preenchidos nos campos para uma página em *PHP* por meio de um *POST*. Essa página, endereçada como `ifce.sanusb.org/elizabeth/horariosjson.php`, recebe esses dados e os transforma em formato de um *JSON*. Finalmente, a partir desta página é enviado o *JSON* em formato de texto para o endereço fixo `ifce.sanusb.org/elizabeth/view` que tem a única funcionalidade de guardar esse conteúdo. A Figura 1 ilustra o formato do *JSON* gravado no arquivo *view*.



Figura 1. Json gerado de acordo com o preenchimento do formulário

A Figura 2 ilustra o fluxo do tratamento de dados feito desde a submissão do formulário até o armazenamento final.

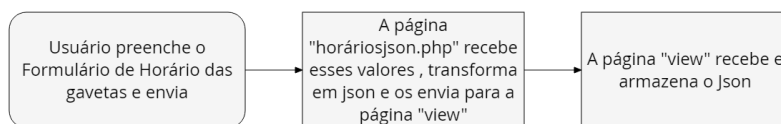


Figura 2. Fluxograma Web

2.2. Programação do Microcontrolador

A plataforma utilizada para desenvolvimento, compilação e gravação do código foi a Arduino IDE. A programação nessa parte foi desenvolvida em Linguagem C. Para o início

da rotina, o ESP8266 se conecta à rede Wifi utilizando a biblioteca ESP8266WiFi.h. Em seguida, utilizando a biblioteca time.h [Stoffregen 2016], o microcontrolador captura da Internet a data e horário real. Posteriormente, dentro de um *loop*, segue executando uma rotina de requisição do tipo *GET* para obter o *JSON* de dados, utilizando a biblioteca ESP8266HttpClient.h. Em sequência, com o auxílio da biblioteca ArduinoJson.h [Blanchon 2015], o *JSON* capturado é tratado e são extraídos os horários que o usuário programou. Finalmente os horários recebidos são comparados com o horário atual. A Figura 3 ilustra o fluxograma desse algoritmo.

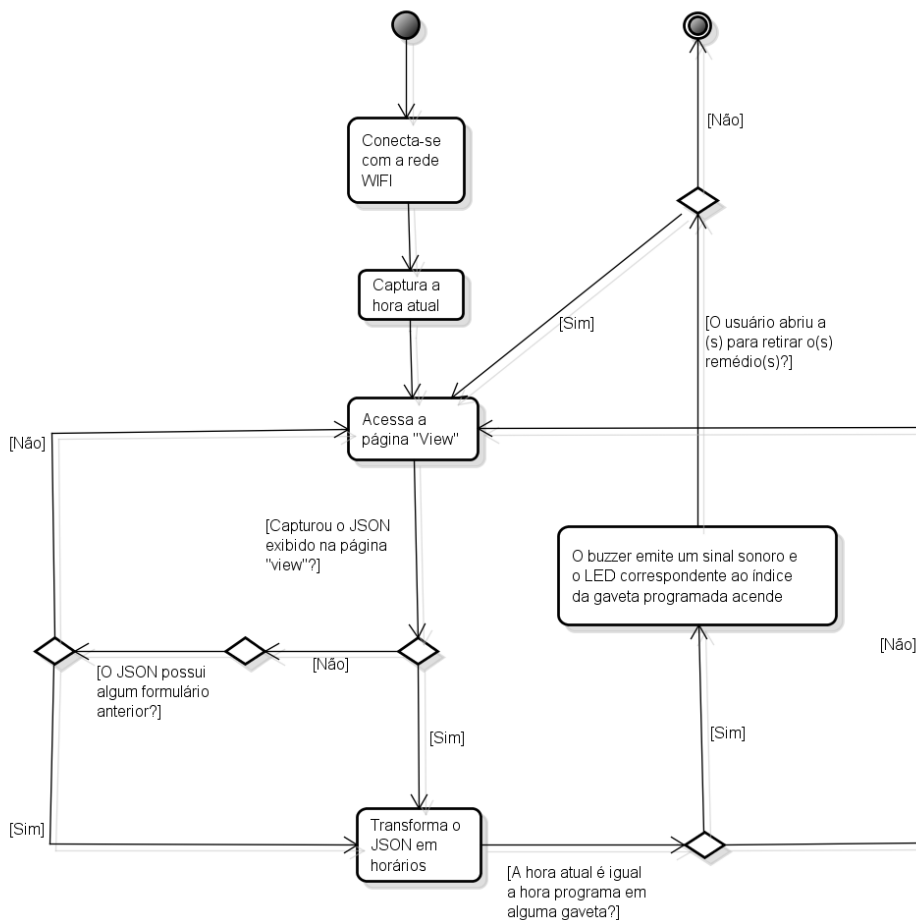


Figura 3. Fluxograma do software no ESP8266

Ao verificar que o horário atual é igual ao enviado pelo usuário, o LED correspondente à gaveta na qual o usuário preencheu acende e o *buzzer* emite um sinal sonoro. Para desativar o horário, basta que o usuário abra a gaveta para pegar o remédio.

Isso ocorre pois, quando as gavetas estão fechadas, um circuito condutor em série mantém o nível lógico alto, programado no pino D7. Ao abrir, o contato da gaveta aberta interrompe essa corrente, mudando o nível do pino D7. Quando isso ocorre, a função *attachInterrupt()*, que foi configurada para esse pino, é disparada. Essa função é utilizada para garantir uma comunicação contínua sobre o pino em que o botão foi conectado. Sendo ela disparada, chama outra função que modifica o sinal do LED e do *buzzer* para nível lógico baixo.

Além disso, para que o ESP8266 não reinicie com essa falta de corrente, é utilizada em seguida da função de interrupção, a *detachInterrupt()*, para desativar a interrupção naquele momento. A Figura 4 ilustra o sensor de corrente.

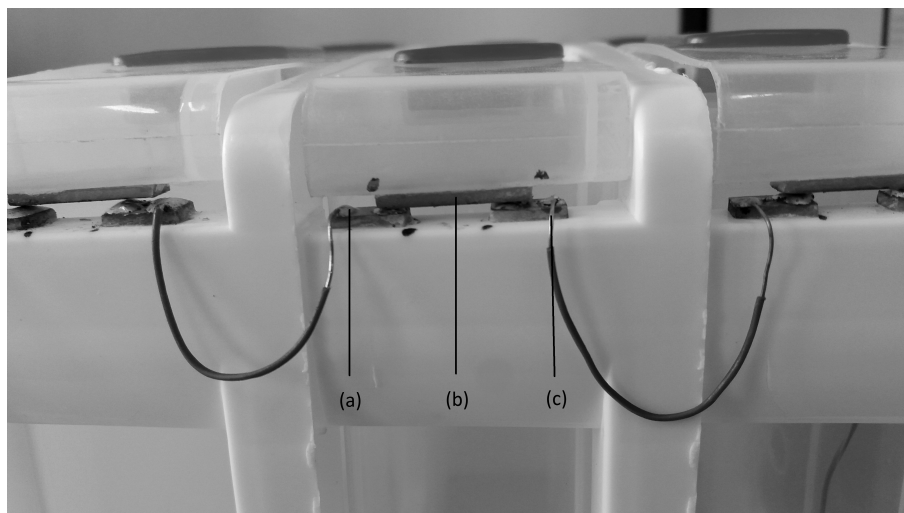


Figura 4. Sensor de corrente fechado. (a) Placa de cobre estanhada na estrutura da gaveta (b) Placa de cobre estanhada na gaveta (c) Fio de cobre

2.3. Circuito Desenvolvido

Utilizando a placa de ensaios, ou seja, a *proto-board*, foram ligados os seis LEDs que representam o índice das respectivas gavetas do recipiente nos pinos de D0 a D5 do microcontrolador ESP8266. No pino D8 foi ligado um emissor de sinal sonoro, também conhecido como *buzzer*. O pino D7 foi designado para detecção de interrupção. Neste pino, um fio de cobre foi conectado de forma que, ao manter as gavetas fechadas, é estabelecido um circuito em série. Para fechar esse circuito em série, o fim do mesmo foi conectado a um pino *GND*. A Figura 5 mostra o resultado do circuito montado.

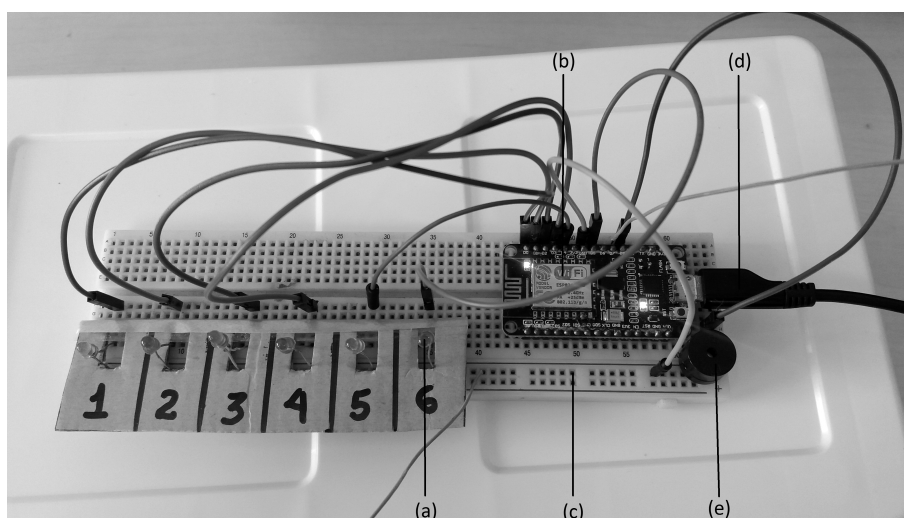


Figura 5. Circuito desenvolvido (a) Led de índice de gaveta (b) ESP8266 (c) Corrente GND (d) Alimentação MicroUSB 5v (e) Buzzer

3. Resultados

A Figura 6 ilustra a interface disponibilizada em ifce.sanusb.org/elizabeth/formulario remedios.html, para que o usuário preencha com os horários que deseja tomar os remédios guardados em determinada gaveta.



The screenshot shows a web browser window with the URL ifce.sanusb.org/elizabeth/formulario remedios.html. The page title is "Horário Programado para sinalização das Gavetas". It contains five sections, each for a drawer: "Primeira Gaveta", "Segunda Gaveta", "Terceira Gaveta", "Quarta Gaveta", and "Quinta Gaveta". Each section has a time selection interface with "H:" for hours and "M:" for minutes, followed by "min".

Figura 6. Formulário de horários HTML

A Figura 7 o circuito integrado ao recipiente. Os fios azuis ligam as gavetas em série, conectados ao circuito controlado pelo ESP8266. As gavetas foram enumeradas e a tabela física indica qual o índice da gaveta está o remédio para o qual o sistema alarma.

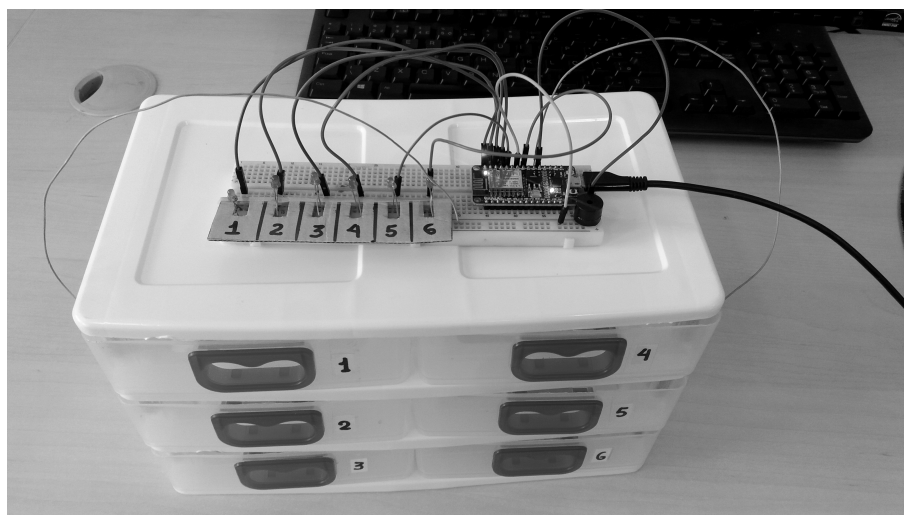


Figura 7. Circuito integrado ao recipiente

4. Conclusão

Durante a construção do projeto proposto, a aplicação da estrutura plástica adquirida não foi adequada para esse tipo de protótipo devido à fragilidade. Além disso, mostrou-se a necessidade de algum componente para manter o contato pressionado e garantir a passagem de corrente contínua quando todas as gavetas estão fechadas.

Outra dificuldade encontrada no desenvolvimento do projeto, levou as gavetas a serem colocadas em série conectadas em um só pino, devido a falta de pinos adicionais do modelo do ESP8266. Em caso de utilizar um microcontrolador com mais pinos disponíveis, seria possível destinar um pino detector de interrupção para cada gaveta.

Durante o desenvolvimento desse projeto, foi possível constatar a facilidade do manuseio do ESP8266 e a grande possibilidade de adaptação para tornar vários tipos de tarefas manuais em automáticas. Além disso, esse trabalho mostrou a construção de uma ferramenta simples para solucionar um problema frequente e recorrente de forma econômica e funcional. O projeto proposto apresentou também a descrição da montagem básica de um protótipo de um sistema IoT de alerta para uso de remédios controlados. Esse sistema proposto foi implementado em uma situação real e funcionou como projetado.

Referências

- Blanchon, B. (2015). C++ json library for iot, simple and efficient. Disponível em <https://github.com/bblanchon/ArduinoJson>. Acesso em abril de 2018.
- de Oliveira, S. (2017). *Internet das Coisas com ESP8266, Arduino e Raspberry Pi*. Novatec Editora.
- dos Santos, N. M. L. (2013). Desenvolvimento de um dispensador automático de medicamentos.
- Espressif (2018). Esp8266ex datasheet. Disponível em https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. Acesso em março de 2018.
- Kodali, R. K. and Soratkal, S. (2016). Mqtt based home automation system using esp8266. In *Humanitarian Technology Conference (R10-HTC), 2016 IEEE Region 10*, pages 1–5. IEEE.
- Schemberger, M. (2016). Dispensador de medicamentos automático microcontrolado. Disponível em <http://www.faculdadespontagrossa.com.br/revistas/index.php/technoeng/article/download/197/165>. Acesso em abril de 2018.
- Schwartz, M. (2016). *Internet of Things with ESP8266*. Packt Publishing Ltd.
- Singh, P. and Saikia, S. (2016). Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and esp8266 wifi module. In *Humanitarian Technology Conference (R10-HTC), 2016 IEEE Region 10*, pages 1–4. IEEE.
- Škraba, A., Koložvari, A., Kofjač, D., Stojanović, R., Stanovov, V., and Semekin, E. (2016). Streaming pulse data to the cloud with bluetooth le or nodemcu esp8266. In *Embedded Computing (MECO), 2016 5th Mediterranean Conference on*, pages 428–431. IEEE.
- Stoffregen, P. (2016). Arduino time library. Disponível em <https://github.com/PaulStoffregen/Time>. Acesso em abril de 2018.
- Thaker, T. (2016). Esp8266 based implementation of wireless sensor network with linux based web-server. In *Colossal Data Analysis and Networking (CDAN), Symposium on*, pages 1–5. IEEE.