

Automação residencial: Comandando lâmpadas pelo *Telegram*

Parley Martins - 11/0038096 Tatielen Pereira - 12/0136074

Resumo—Este trabalho propõe a utilização de automação residencial para possibilitar ao usuário acender e apagar lâmpadas remotamente, utilizando integração com aplicativo no celular.

Index Terms—Automação Residencial, Telegram, bot, *smart house*

1 INTRODUÇÃO

Automação residencial é resultado da combinação de espaços residenciais, como sala, banheiro, quarto com tecnologias, para maior conforto, segurança, ou menos contato humano [1]. Estas tecnologias e ideias eram, até recentemente, consideradas sonhos de um futuro distante [2], sem uso prático, exceto no entretenimento.

No entanto com um mundo conectado pela internet, que mudou o jeito que as pessoas se comunicam e se relacionam, é normal que este conceito esteja cada vez mais próximo da realidade das pessoas. Para ter mais conforto, já é possível controlar pelo celular o volume das televisões (e outros aparelhos de som), o canal em que se está, a intensidade com que aparelhos devem funcionar, entre outras comodidades. Para ter mais segurança, é possível controlar luzes, sistema de alarmes, de detecção de movimentos, etc. Existem diversas empresas que fornecem esse tipo de serviço, mas eles ainda podem ter um custo muito elevado.

2 SOLUÇÃO

Para facilitar e desmistificar o acesso à automação residencial, a proposta deste projeto é implementar um sistema que possa controlar remotamente as lâmpadas de uma casa. O

usuário, após instalação do sistema físico, poderá utilizar seu *smartphone* para ligar e desligar as lâmpadas.

A interação com o usuário se dará através de um bot no aplicativo *Telegram*. Deve-se iniciar uma 'conversa' com o bot, e mandar o comando desejado (ligar ou desligar, por exemplo). Este irá mandar para o módulo wifi do sistema, que fará a comunicação com o MSP, desligando ou ligando a lâmpada selecionada.

Para fins deste trabalho, uma lâmpada e uma fonte de energia externas, controladas pela protoboard, serão utilizadas para facilitar a instalação e testes.

O *hardware* será composto, inicialmente, pelos seguintes itens:

- protoboard, para execução do sistema;
- microcontrolador MSP430, irá executar o controle da energia na lâmpada;
- módulo esp8266, proverá o acesso à rede wifi;
- lâmpada, para testes;
- fonte de energia, tanto para o microcontrolador quanto para a lâmpada.

O *software* embarcado no microcontrolador será escrito nas linguagens C e Assembly, enquanto o código do *bot* será desenvolvido utilizando Python 3. Os serviços serão conectados através do IFTTT, que conecta servidores de terceiros a outros serviços [3].

2.1 Requisitos

O *software* do microcontrolador deve corretamente identificar os comandos e apagar ou acender a lâmpada, conforme instrução recebida.

O *bot*, *software* que responde a comandos pré definidos automaticamente, deve ser integrado ao aplicativo *Telegram* e deve mandar instruções de ligar e de desligar a lâmpada.

O sistema completo, tanto hardware quanto *software*, deve ter acesso à internet para o funcionamento correto.

2.2 Benefícios

Este projeto tem como principal beneficiário o cidadão comum que quer ter um pouco do conforto que a automação residencial traz a sua casa. Além disso, ajudará na economia de energia, já que a pessoa pode mandar um comando de apagar determinada luz, mesmo a distância.

2.3 Resultados Parciais

Para realizar a comunicação entre a internet e o MSP foi escolhido o módulo ESP 8266, no entanto, até este ponto do trabalho, não foi possível comunicar os dois. O ESP parece não se conectar à internet ou receber os comandos AT que deveria. O ESP foi conectado à entrada e saída seriais do MSP, mas não deu resultado. Outra tentativa foi utilizar um conversor USB-serial para transmitir o *firmware* diretamente pro ESP, que também se provou infrutífero, talvez por algum erro de conexão.

O MSP foi configurado para receber um caractere na conexão UART, que deveria vir da internet, que trocará o estado da lâmpada ou lerá esse estado. Os caracteres aceitos são l, para ligar; d, para desligar; e e para ler o estado atual da lâmpada. Qualquer outro input resultará em nenhum retorno ou ação do MSP. O código abaixo inicializa o modo de comunicação UART com *baud rate* 9600 e *clock* em 1MHz. Caso algum dado seja recebido, uma interrupção do RX do UART será acionada. Esta interrupção lerá o valor recebido e tomará uma ação de acordo com o explicado acima. Para substituir a lâmpada, por enquanto, o LED conectado ao pino P1.6 do MSP está sendo utilizado.

Listing 1. lamp.c

```

1 #include <msp430g2553.h>
2 #include <legacymsp430.h>
3
4 #define RX BIT1
5 #define TX BIT2
6 #define LAMP BIT6
7
8 void send_data(unsigned char c);
9 void init_uart();
10
11 int main(void)
12 {
13     WDTCTL = WDTPW + WDTHOLD;
14
15     BCSCTL1 = CALBC1_1MHZ;
16     DCOCTL = CALDCO_1MHZ;
17
18     P1OUT &= ~LAMP;
19     P1DIR |= LAMP;
20
21     init_uart();
22     _BIS_SR(GIE);
23     return 0;
24 }
25
26 void send_data(unsigned char c)
27 {
28     while((IFG2&UCA0TXIFG)==0);
29     UCA0TXBUF = c;
30 }
31
32 void init_uart()
33 {
34     P1SEL2 = P1SEL = RX + TX;
35     UCA0CTL0 = 0;
36     UCA0CTL1 = UCSSEL_2;
37     UCA0BR0 = 104;
38     UCA0BR1 = 0;
39     UCA0MCTL = UCBRF_0 + UCBRS_1;
40     IE2 |= UCA0RXIE;
41 }
42
43 void send_state(char state[]){
44     char phrase[] = "Estado: ";
45     int i = 0;
46     for(i = 0; phrase[i] != '\0'; i++){
47         send_data(phrase[i]);
48     }
49     for(i = 0; state[i] != '\0'; i++){
50         send_data(state[i]);
51     }
52     send_data('\n');
53 }
54
55 interrupt(USCIAB0RX_VECTOR) set_lamp_state(
56     void){
57     unsigned char state = UCA0RXBUF;
58     if(state == 'l'){
59         P1OUT |= LAMP;
60     } else if(state == 'd') {
61         P1OUT &= ~LAMP;
62     } else if(state == 'e') {
63         if((P1OUT&LAMP)==0){
64             send_state("desligado");

```

```

65     } else{
        send_state("ligado");
67     }
    }
69 }

```

REFERÊNCIAS

- [1] F. Moraes, A. Amory, N. Calazans, E. Bezerra, and J. Petrini, "Using the can protocol and reconfigurable computing technology for web-based smart house automation," in *Integrated Circuits and Systems Design, 2001, 14th Symposium on*. IEEE, 2001, pp. 38–43.
- [2] A. GhaffarianHoseini, N. D. Dahlan, U. Berardi, A. GhaffarianHoseini, and N. Makaremi, "The essence of future smart houses: From embedding ict to adapting to sustainability principles," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 593 – 607, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1364032113001342>
- [3] S. Ovadia, "Automate the internet with "if this then that"(ifttt)," *Behavioral & social sciences librarian*, vol. 33, no. 4, pp. 208–211, 2014.