

Segurança e monitoramento com RFID

IoT aplicada à segurança e ao monitoramento

Aubani Júnio Teixeira Cândido
Universidade de Brasília - UnB
Brasília-DF, Brasil
junio@aluno.unb.br

Elpidio Cândido De Araujo Bisneto
Universidade de Brasília - UnB
Brasília-DF, Brasil
elpidio.araujo@hotmail.com

Resumo— Uso de conceitos de Internet das Coisas, Internet of Things (IoT), para a implementação de um sistema de segurança e monitoramento de um ambiente através de acesso controlado de RFID e sincronizado em nuvem.

Keywords—monitoramento, segurança, IoT, RFID

I. JUSTIFICATIVA

Tendo em vista a necessidade de sistemas de segurança que controlem o acesso a um ambiente, exemplo de laboratórios, é necessário fazer o gerenciamento do acesso. Uma forma eficiente de realizar essa tarefa é usando a tecnologia de RFID aliada aos conceitos de IoT e armazenamento em nuvem. Proporcionando um controle daqueles que tiveram acesso e daqueles cadastrados para acessarem o ambiente.

II. OBJETIVO

A. Garantir segurança no acesso

Somente pessoas autorizadas por um dos administradores do ambiente, teriam acesso ao ambiente, impedindo que outros não convidados ou pessoas de foras tenham acesso ao local, garantindo integridade e sigilo. Tudo isso com a tecnologia de RFID.

B. Monitoramento do acesso

Mesmo tendo pessoas autorizadas acessando o ambiente é necessário que haja um monitoramento, informando quem acessou o ambiente, que horas acessou e quanto tempo ficou dentro do ambiente. Todos esses dados serão sincronizados em nuvem de forma criptografada, possibilitando consulta em outros dispositivos. Nessa parte também entra o conceito de IoT, fazendo uso de sensores que serão *Things* para interação com a internet via módulo Wi-Fi num servidor controlado.

III. TABELA DE MATERIAIS UTILIZADOS

UND	MATERIAIS	FABRICANTE
01	MSP430G2553LP	TEXAS INSTRUMENT
01	RFID MÓDULO RC522	NXP
03	CARTÕES COM RFID	-
01	MODULO WIFI - ESP8266 ESP-01	ESPRESSIF SYSTEMS
-	JUMPERS	-
01	PROTOBOARD	HIKARI
01	MODEM DSL-2401HN-T1C-NV	MITRASTAR
01	DISPLAY LCD COM MÓDULO 12C	-
01	BUZZER 5V	-
01	RESISTOR 1KΩ	-
01	RESISTOR 10KΩ	-

IV. HARDWARE

Para a realização desse projeto utilizamos os pinos (colocar os pinos) para fazer a comunicação entre o RFID e a MSP430, utilizamos as entradas para SPI e UART para realizar a comunicação com o módulo RFID (SPI) e módulo WiFi (UART), além disso foi utilizado um modem com o objetivo de mandar os dados que forem lidos para a nuvem.

Assim que o usuário aproxima o cartão no RFID, o sistema enviará esses dados para a nuvem, informando que alguém está tentando acessar o local. Se o cartão for cadastrado o acesso é

liberado, caso contrário, será mostrado acesso negado e registrado na nuvem.

Além de enviar para a nuvem, o sistema terá um aviso na hora do acesso. Quando o usuário aproximar o cartão o sistema emitirá um aviso sonoro através de um buzzer e em um display mostrará se o usuário tem acesso liberado ou não, além disso o led vermelho apagará na hora da liberação do acesso.

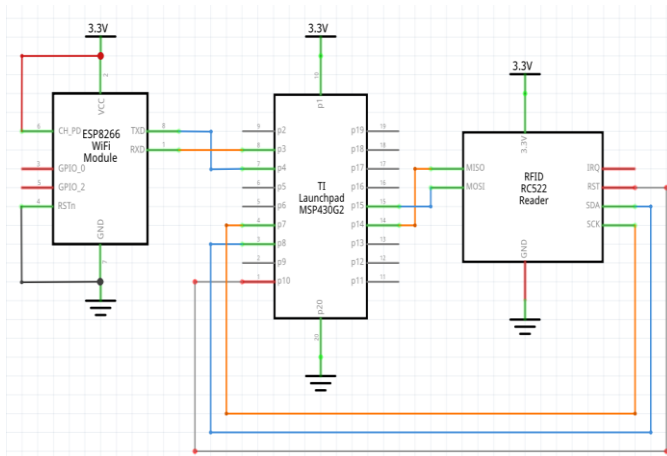


Figura 1 – Esquemático [2]

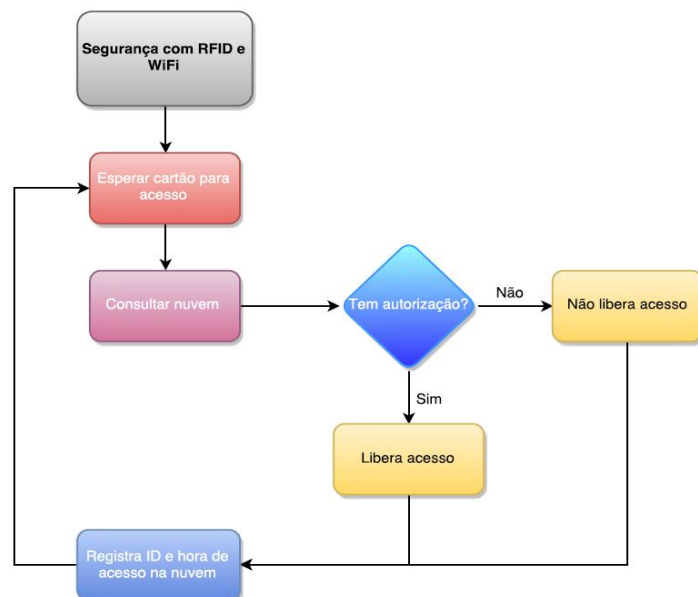


Figura 2 – Fluxograma

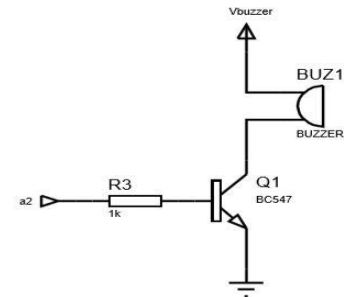


Figura 4 – Esquemático do aviso sonoro

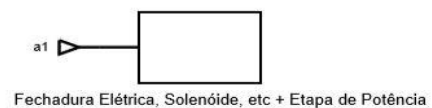


Figura 5 – Esquemático do servo motor simulando a porta.

V. SOFTWARE

O projeto foi iniciado em códigos separados com o software "Energia". Primeiro foi feito um código que conecta à rede WiFi, usando nome e senha da rede, para isso foi utilizado comunicação UART nos pinos 3 e 4 do MSP430.

Após foi feito o código para reconhecimento do RFID, com uma biblioteca livre no GitHub [1]. O sistema usa SPI, e as conexões são mostradas na Figura 1.

Em seguida, foi feita uma otimização para a comunicação UART buscando uma melhor performance para o módulo WiFi. Essa otimização foi realizada escrevendo todo o código que permitisse a manipulação da UART usando os registradores do próprio MSP430.

Após essa etapa foi feito o código utilizando o TIMER A para acender um Led da MSP430, mostrando que o usuário teve acesso liberado, o Led permanece aceso por cerca de 5s.

A etapa seguinte foi implementar o protocolo Tcp/IP em um servidor que funcionaria como nuvem para prover informações ao MSP430. O servidor recebe 5 bytes do ID, caso seja autorizado, responde enviando 1, caso contrário, envia 0 ao *client*.

O código foi passado do "Energia" para CCS de acordo com a figura 3.

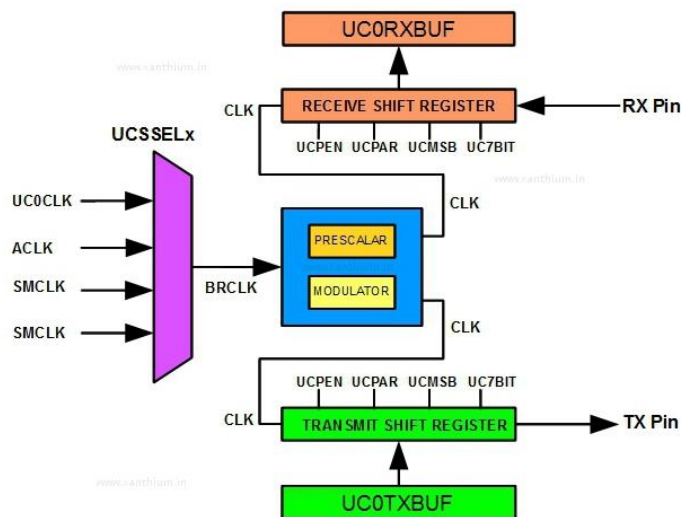


Figura 3 – Diagrama dos registradores para UART [3].

A figura 3 mostra os registradores utilizados para fazer a manipulação do UART no MSP430. O algoritmo consiste em usar esses registradores fazendo interrupção no TX e RX, ou seja, ao receber ou enviar pelo UART, executar um ISR

```
interrupt(USCIAB0TX_VECTOR) USCI0TX_ISR(void)
{
    //Pega os valores passados nos parâmetros da função
    byte* buf = Serial_UART.TxBuffer;
    volatile int* i = &Serial_UART.iTx;
    volatile byte* lock = &Serial_UART.lockTX;
    *i++;

    // Compara se o valor é maior ou igual com o tamanho que deve ler
    // >= caso o length seja 0, ele lê 1 vez da mesma forma.
    if (*i >= Serial_UART.lengthTX)
    {
        *lock = 0;
        UC0IE &= UCA0TXIE;
    }

    else //Coloca o valor para ser enviado no BUF do TX
        UCA0TXBUF = buf[*i];

    //Desabilita a flag do TX
    //IFG2 &= ~UCA0TXIFG;
}
```

Figura 4 – Código de interrupção TX – UART

A figura 4 mostra o trecho de código de interrupção para envio de dados pelo Serial UART. Ele usa parâmetros dentro da classe “Serial_UART” para fazer o envio correta das informações. Mais trechos do código se encontram no link: <https://github.com/Metalus/Microcontroladores/tree/master/Projeto%20Final>.

A outra parte do sistema de monitoramento utiliza o RFID, nos primeiros testes foi utilizado bibliotecas do próprio Energia para avaliar o funcionamento do módulo. Após, foi realizado o código para o SPI utilizando os registradores do próprio MSP430, entretanto, diferentemente do código para UART, nesse não foi utilizado rotinas de interrupção. Considerando que o SPI tem modo de operação *full-duplex*, o código se torna mais eficiente escrevendo uma função para ler e escrever ao mesmo

tempo, a figura 5 mostra o trecho de envio e recebimento do SPI.

```
unsigned char Serial_SPIClass::transfer(unsigned char data)
{
    UCB0TXBUF = data; // enviar o byte
    while (UCB0STAT & UCBSY); //aguardar o registrador
    return UCB0RXBUF; // ler o valor recebido
}
```

Figura 5 – Código SPI.

```
char rfid[] = { serNum[0], serNum[1], serNum[2], serNum[3], serNum[4] };
SendCommand("AT+CIPSEND=5\r\n",1000);
Serial_UART.print(rfid);
delays(1000);

Serial_UART.readStr(recv, 0, 9);

char result = recv[7];
if (result == '1')
{
    digitalWriteA(1, BIT0, HIGH);
    delays(5000);
    digitalWriteA(1, BIT0, LOW);
}
else
{
    digitalWriteA(1, BIT0, LOW);
}
```

A figura 6 mostra o trecho do código para acender o Led na hora que o usuário é liberado. (Explicar a ideia do timer a nessa parte).

VI. REQUISITOS

Uma placa MSP430, um módulo Wi-Fi, um módulo de RFID, Cartão RFID e uma nuvem para armazenamento dos dados. Estudo sobre protocolo TCP, conceitos de IoT, funcionamento de RFID e comunicação com a plataforma em nuvem utilizada.

VII. BENEFÍCIOS

O sistema irá se beneficiar de tecnologia mais atual para a autorização de pessoas como o RFID, sendo cada cartão único para ter acesso ao ambiente. Uso de sistema em nuvem para monitoramento, tornando-o eficaz para análise de controle ao ambiente. Uso de conceitos de IoT, para uma nova ponta de tecnologia, onde temos sensores como *Things*, mantendo essa abstração o sistema pode ser complementado para interação com outros. Baixo custo de projeto e de consumo.

VIII. RESULTADO

A dupla conseguiu passar os códigos do Energia para o CCS, escrevendo todos os códigos que estão em anexo, a MSP430 fez a comunicação bem com o modulo RFID, já o Modulo WiFi não houve uma Comunicação boa, a dupla tentou encontrar algum erro de código ou de montagem na protoboard, porém não foi encontrado nenhum erro nesses quesitos, foi substituído o modulo e a comunicação voltou. Ao passar um cartão cadastrado pelo sensor de RFDI, o sistema mostrou que o

acesso foi liberado e o Led da placa ficou aceso por cerca de 5 segundos, ao passar o cartão não cadastrado o sistema mostrou o acesso negado.

IX. DISCURSÕES

A dupla teve dificuldade na implementação da comunicação UART por sincronizar o tempo com o delay do módulo WiFi, há atrasos entre um comando e outro para o funcionamento correto, então foi necessário usar o Timer A para controlar esse atraso e mandar os comandos no momento correto. Além disso foi utilizado a comunicação SPI para o módulo de RFID, nessa comunicação o funcionamento foi mais

simples, o fato de ser uma comunicação *Full-Duplex* permite enviar um byte e retornar o valor recebido na própria função.

REFERÊNCIAS

- [1] fmilburn3. Biblioteca RC522 para MSP430. Disponível em <https://github.com/fmilburn3/CardReader_RFID_RC522>
- [2] Energia. Mapeamento dos pinos. Disponível em <<http://energia.nu/pin-maps/>>
- [3] Xanthium. Imagem do diagrama dos registradores para UART no MSP430. Disponível em <<http://www.xanthium.in/sites/default/files/site-images/serial-com-msp430-uart/uart-msp430-block-dia.jpg>>