

Software Embarcado

Tópicos Especiais em Redes de Telecomunicações

Apresentação

Alisson Cavalcante e Silva

Acadêmica

Mestrado Engenharia Eletrônica

Linha de Pesquisa: Rede de Computadores e Sistemas Distribuídos

Profissional

Marinha do Brasil

Analista de Segurança da Informação Digital

Proposta

**Controle Automatizado para Escritórios e
Laboratórios por meio de Redes de RF with use
Arduino**

Proposta

Motivação:

Observação diária da rotina dos monitores responsáveis pelos Laboratórios do Departamento de Engenharia da UERJ.

Problema:

Os monitores precisam:

- abrir/fechar as dependências dos laboratório;
- ligar/desligar o barramento elétrico dos Laboratórios (disjuntores dos circuitos elétricos situados dentro dos laboratórios); e
- ligar/desligar aparelhos condicionadores de ar, ventiladores e projetores de vídeo.

Essas tarefas são realizadas constantemente no dia a dia, dependendo do uso pelos professores e alunos. Para execução da tarefa os monitores precisam se dirigir até os laboratórios, levando-os a interromper a execução de outras atividades/tarefas, e a ter que se dividir para atender a constante demanda acadêmica.

Proposta

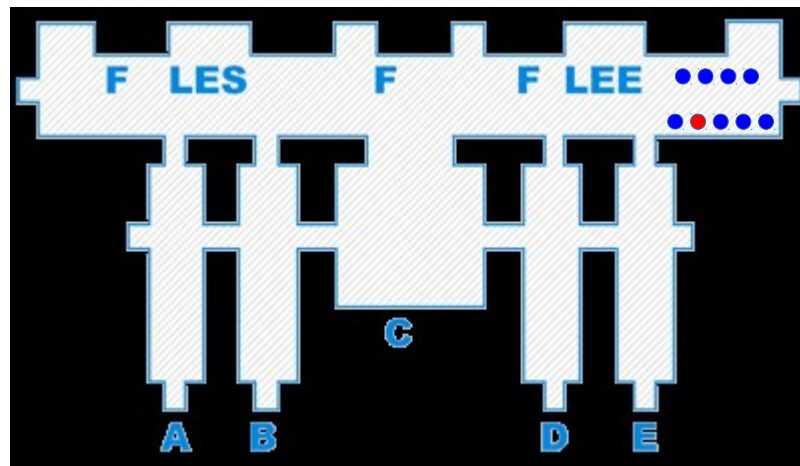
Solução:

Disponibilizar a abertura do laboratório, além da energização de seu barramento elétrico, de forma remota. Sendo utilizado uma rede de **RF with use arduino**, tendo sua administração centralizada na sala do monitores.

Vantagens de uso do arduino comparado a utilização das Redes de Computadores tradicionais:

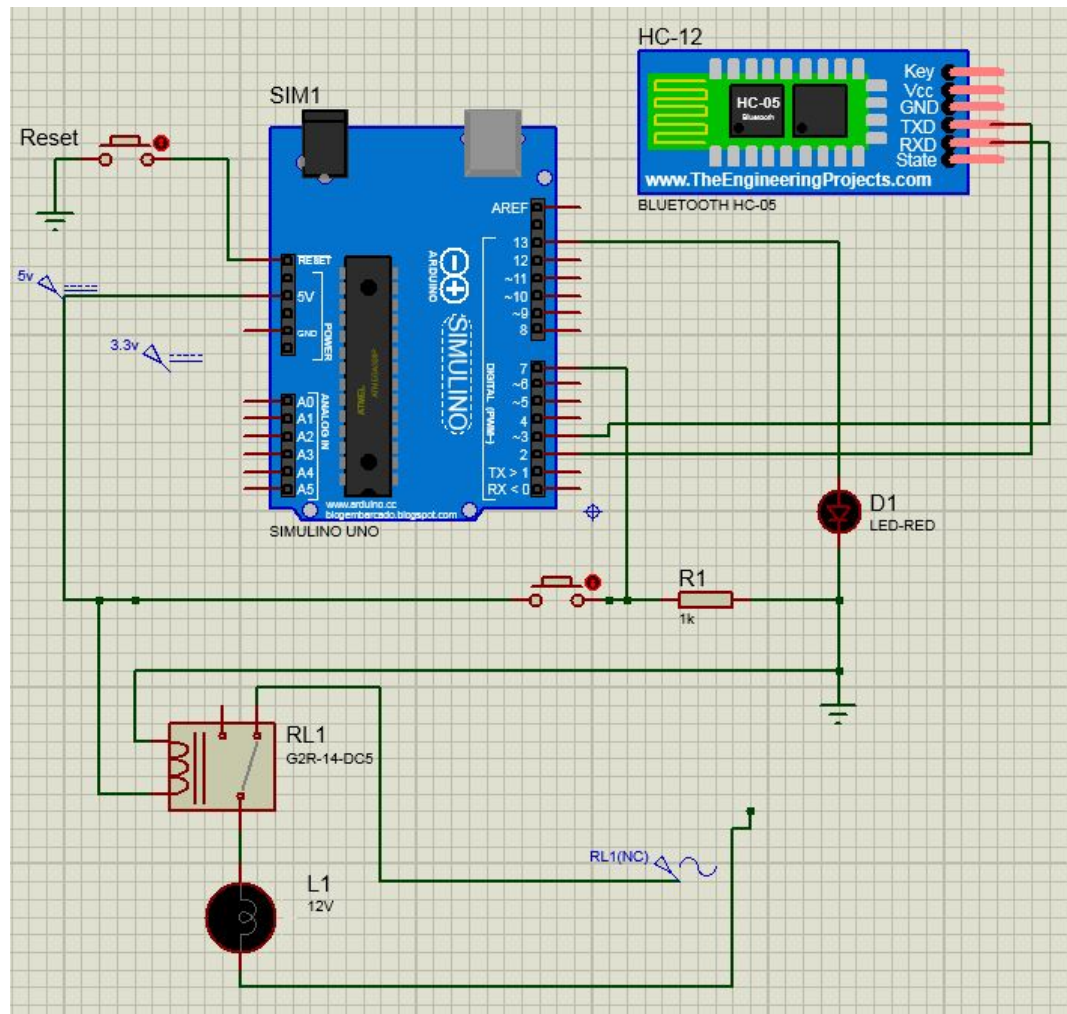
- **802.3 ou 802.11**
 - custo com cabeamento
 - requer dispositivos de Rede (switchs e roteadores)
 - uso de rede de gerência, mais gasto
 - uso de rede corporativa, compromete nível de segurança
- **RF with arduino**
 - baixo custo
 - baixa manutenção
 - sem custo com cabeamento ou dispositivos de Rede

Proposta



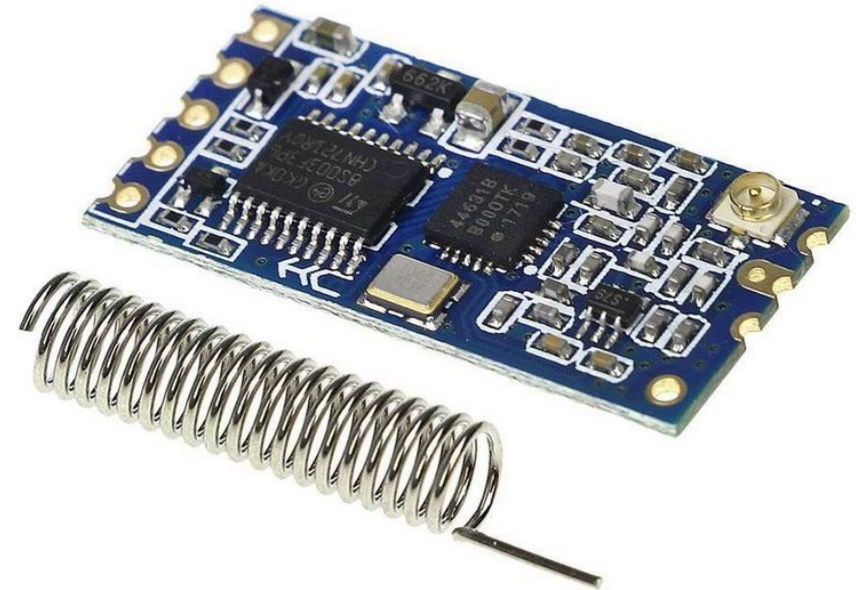
- Sala de monitores
- Laboratórios

Proposta



Proposta

- Tensão: 3.3V à 5V
- Frequência: 433.4 a 473.0MHz
- (até 100 canais de comunicação)
- Número de endereços: 256 (Padrão 000)
- Número de canais: 100 (Padrão 001)
- Configuração: Comandos AT
- Comunicação: RS232
- Potência de transmissão máx.: 100mw
- BaudRate: 9600 (Padrão)
- Distância de comunicação padrão: 600m (ambiente aberto), máx. 1000m ajustável;
- Dimensões (sem antena): 27,5 x 13,5 x 1,8mm.



Código

```
/*
 * Fonte de Estudo:
 * Projeto Curto Circuito - Módulo HC12 - Conectando dois Arduinos (Emissor)
 * http://hausofpayne.weebly.com/hc12teletype.html
 *
 * observacoes:
 * a) porta serial possui um buffer de 64bytes;
 * b) a funcao read() le retirando o byte do buffer, caso queira mante-lo usar funcao peek()
 * c) available() retorna o numero de bytes no buffer
 */

#include <SoftwareSerial.h> // Inclui Biblioteca
#define buttonPin 7 // pino do button
#define ledPin 13 // pino do LED
#define lamp 8 // pino do LED
SoftwareSerial HC12(2, 3); // Simula RX e TX em outras portas
|
char input;
String msgTX = "";
String msgRX = "";

int incomingByte;

void setup()
{
    pinMode(lamp, OUTPUT);
    pinMode(ledPin, OUTPUT); // Declara que ledPin é uma saída
    pinMode(buttonPin, INPUT); // Declara que buttonPin é uma entrada
    HC12.begin(9600); // Taxa de transferência do HC12
    Serial.begin(9600); // Taxa de transferência da Serial
    digitalWrite(lamp, HIGH);
}
```

Código

```
void loop()
{
    // **** bloco de envio ****
    if (Serial.available() > 0)    // se tiver byte no buffer da serial satisfaz condicao
    {
        input = Serial.read(); // realiza leitura do buffer serial, conforme já retira o byte
        if (input != '\r' && input != '\n') // retirada dos bytes de recuo de carro e nova linha
        {
            msgTX += char(input); // monta a string dos caracteres colocados no buffer da serial
        }
    }
    else
    {
        delay(200);
        if(msgTX.length() != 0)    // se tiver msg transfere pelo modulo HC-12
        {
            for (int i = 0; i < msgTX.length(); i++)
            {
                HC12.write(msgTX.charAt(i)); // transfere pelo modulo HC-12 byte a byte
            }
            Serial.println(msgTX); // no fim da TX imprime o conteúdo transmitido
            msgTX = ""; // limpa a string com conteúdo transmitido
        }
    }

    // **** bloco de recepcao ****
    while (HC12.available() > 0)
    {
        delay(5); // gera um atraso antes realizar leitura do buffer
        incomingByte = HC12.read(); // leitura do que chega pelo RX no modulo HC-12
        if (incomingByte != '\r' && incomingByte != '\n') // retirada dos bytes de recuo de carro e nova linha
        {
            msgRX = msgRX + char(incomingByte); // monta string por meio da recepcao de caracter e caractere
        }
    }
}
```

Código

```
if(msgRX.length() != 0) // se tiver msg imprime na serial
{
    Serial.println(msgRX);

    // Se msg recebida for igual a:
    if (msgRX.endsWith("LED") || msgRX.endsWith("2"))
    {
        digitalWrite(ledPin, HIGH); // acende o LED
    }
    else if (msgRX.endsWith("led"))
    {
        digitalWrite(ledPin, LOW); // apaga o LED
    }
    else if (msgRX.endsWith("LAMPADA"))
    {
        digitalWrite(lamp, LOW); // apaga o LED
    }
    else if (msgRX.endsWith("lampada"))
    {
        digitalWrite(lamp, HIGH); // apaga o LED
    }
    msgRX = "";
}

int buttonState = digitalRead(buttonPin); // Armazena o estado lógico do botão
if (buttonState == 1) // Se o botão for pressionado
{
    HC12.write('2'); // TX pelo modulo HC-12
}
}
```

Referências

Referências:

- [1] MAPA UERJ - <http://www.eng.uerj.br/admin/espaco.php>
- [2] HC-12 - <https://www.curtocircuito.com.br/modulo-radio-frequencia-433mhz-hc12.html>
- [3] Código Base - <http://hausofpayne.weebly.com/hc12teletype.html>

Dúvidas

