

Djames Suhanko – Linux e Programação

Assuntos técnicos voltados a Linux e programação.

Eletrônica digital com Arduino – Redes RS485 com MAX487

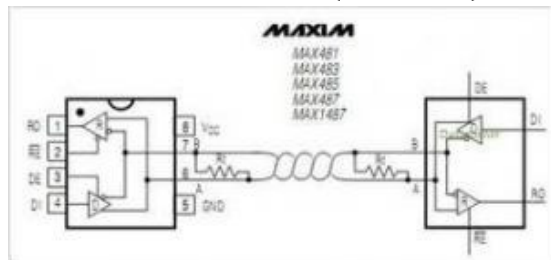
Publicado em 11/07/2012 | [Deixe um comentário](#)



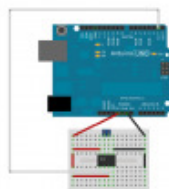
O RS485 é um padrão de rede serial para distâncias de até 1200M. Como descrito [nessa referência](#), um dos principais problemas do RS232 é o ruído na linha de sinal, que usa o zero como referência. Com RS485 não há esse problema, pois há uma tolerância grande prevista no protocolo. Os sinais de RS485 são flutuantes e transmitidos sobre as linhas de sinal + e -. Muitos outros detalhes importantes podem ser vistos no link de referência (em inglês), mas outra informação importante é que a limitação de dispositivos interconectados é de 32.

Não encontrei nenhum tutorial de RS485 e foi um parto chegar a uma simples conexão. Para não perder essa primeira etapa, vou escrever o procedimento do 'hello world' e depois complemento com o protocolo que vou escrever.

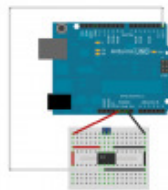
No datasheet você encontrará (entre outras) essa imagem:



Tem que ler bastante e filosofar a respeito para chegar à conclusão das conexões, mas enfim, para a primeira fase desse post fiz um emissor e um receptor. O esquema no Arduino ficou assim para o sender:



E assim para o receiver:



Perceba que a mudança está em basicamente trocar os fios da esquerda de baixo para cima no CI e de TX para RX no Arduino. As correspondências dos pinos (do 1 ao 8, lembrando que se conta em 'U' da esquerda para a direita):

(RO) Receiver out – utilizado no receptor (utilizar o RX do Arduino aqui – pino 0 no Uno)

(RE) Receiver enable – Utilizado com o receptor (LOW ativa a recepção de dados. Colocá-lo em HIGH desabilita a recepção)

(DE) Driver enable (quando em HIGH, utilizado para sinalizar o envio de dados)

(DI) Driver in (transmissor, conectado ao TX do Arduino – porta 1 para o Uno)

(GND) Ground (adivinha?)

(A) Pino A ao pino A do outro CI MAX487 (CPA1036 no exemplo)

(B) Pino B ao pino B do outro CI MAX487 (CPA1036 no exemplo)

(VCC) Energizar em 5V

Então forcei uma mensagem para simular o tráfego. Ainda escreverei um protocolo completo de comunicação RS485, só não tenho certeza se utilizando apenas 1 CI por ponto.

No sender o código ficou assim:

```
int i;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop()
{
  Serial.print("<1-M=00:1d:ba:92:cf:94,H=08:29:33,D=11.07.2012,1=472,2=744,3=646,4=456,5=511,F=60,T=774,E>");
  delay(1000);
}
```

E no receiver, um pouco mais elaborado:

```
char incoming[100] = {' '};
String buffer;
int iniPosStarts = 0;
int finPosStarts = 0;
int iniPosEnds = 0;
int finPosEnds = 0;
```

```

int serialMsgLen = 90;
String strIni,strFim;
int i = 0;

void setup()
{
  Serial.begin(9600);
}

void serialEvent(){
  Serial.readBytes(incoming,serialMsgLen);

  buffer = incoming;
  for (i=0;i<buffer.length();i++){
    if (incoming[i] == '<'){
      iniPosStarts = i;
    }
  }
  iniPosEnds = i;

  if (iniPosStarts > 0){
    for (i=0;i<buffer.length();i++){
      if (incoming[i] == '>'){
        finPosEnds = i;
        break;
      }
    }
  }

  strIni = buffer.substring(iniPosStarts+1,iniPosEnds);
  strFim = buffer.substring(finPosStarts,finPosEnds);

  buffer = strIni;
  buffer += strFim;
}

Serial.println(buffer);
}

void loop()
{
}

```

Reparou em uma coisa muito legal? Não tem nada dentro do loop. O que acontece é que reimplentei o método serialEvent, de forma que ao chegar qualquer dado na serial, ele faz o tratamento e assim não preciso de código em loop.

A saída é aquela string do sender, eliminando os sinais de maior e menor.

Esse é mais um material de colaboração com o software livre promovido pela [LightComm](http://LightComm.com). Quando os produtos que eu

estou trabalhando ficarem prontos, escreverei um artigo de apresentação.

A noite atualizo o post com um vídeo do funcionamento e posteriormente atualizarei o post se fizer ida e volta no mesmo CI.

Ainda veremos outros tipos de rede, fiquem atentos aos próximos posts. Sugiro inclusive que você clique aí em ‘Seguir’ (não deixando de compartilhar nas redes sociais de sua preferência)!

Share this:

2

Esta entrada foi publicada em [Linux](#) e marcada com a tag [arduino](#), [max487](#), [rs485](#). Adicione o [link permanente](#) aos seus favoritos.

Tema: Coraline por Automattic. Blog no WordPress.com.