

CAIO BUSSADORI ORLANDINI

ADAPTADOR ISOLADO DE COMUNICAÇÃO SERIAL

Trabalho de Conclusão de Curso apresentado à
Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em
Eletrônica

ORIENTADOR: Prof. Me. Edson Gesualdo

São Carlos
2011

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

Orlandini, Caio Bussadori

O71a Adaptador isolado de comunicação serial / Caio Bussadori Orlandini ; orientador
Edson Gesualdo -- São Carlos, 2011.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola
de Engenharia de São Carlos da Universidade
de São Paulo, 2011.

FOLHA DE APROVAÇÃO

Nome: Caio Bussadori Orlandini

Título: "Adaptador Isolado de Comunicação Serial"

Trabalho de Conclusão de Curso defendido e aprovado
em 06/06/2011,

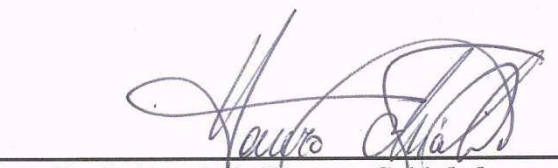
com NOTA 10,0 (dez, zero), pela comissão julgadora:



Profa. Dra. Maria Stela Veludo de Paiva - EESC/USP



Profa. Assistente Luiza Maria Romeiro Coda - EESC/USP



Prof. Associado Homero Schiabel
Coordenador da CoC-Engenharia Elétrica
EESC/USP

DEDICATÓRIA

Para meus pais, Celene e Claudinei, meu avô, Aristides, e minha namorada, Natália, que me apoiaram incondicionalmente desde o começo de minha caminhada até a minha graduação em Engenharia Elétrica.

AGRADECIMENTOS

A todos os meus familiares por me apoiarem nos momentos fáceis e difíceis.

A todos os professores e funcionários da USP – São Carlos pelos ensinamentos que levarei durante toda a vida.

Ao professor Edson Gesualdo por sua orientação e incentivo deste trabalho.

Aos meus amigos de faculdade pelo companheirismo e pelos ótimos momentos compartilhados.

Aos meus amigos e companheiros de trabalho Mário, Bárbara, Alessandro, Rosimar e Yuri pelo apoio e imensa cooperação neste trabalho.

Por fim, agradeço a todos que contribuíram de alguma forma para a concretização deste projeto.

SUMÁRIO

LISTA DE FIGURAS	ii
LISTA DE TABELAS.....	iii
RESUMO	iv
ABSTRACT	v
1. Introdução.....	1
1.1 Comunicação de Dados.....	1
1.2 Padrões de Comunicação.....	3
1.2.1 RS232.....	3
1.2.2 RS485.....	6
1.2.3 USB	8
1.2.4 TTL	10
2. Desenvolvimento	13
2.1 Fonte de Alimentação	13
2.1.1 Funcionamento da Fonte	18
2.2 Transceptores.....	19
2.3 Isolamento Galvânico.....	22
2.4 Lógica de Transmissão/Recepção e Sinais de Controle.....	25
3. Resultados e Discussões.....	31
3.1 Protótipo	31
3.2 Fonte de Alimentação	32
3.3 Transceptores.....	33
3.4 Isolamento Galvânico.....	37
3.5 Geração de RTS para o RS232	41
4. Teste em Aplicação Real	42
5. Conclusões.....	44
6. Referências Bibliográficas.....	44
Apêndice A	48
Apêndice B	49

LISTA DE FIGURAS

Figura 1 – Transmissão RS232.....	4
Figura 2 – Conectores DB.....	5
Figura 3 - Comprimento do cabo x Taxa de Transmissão para o RS485.....	7
Figura 4 – Conectores USB série A	9
Figura 5 – Conectores USB série B	10
Figura 6 – Comparação TTL x RS232	11
Figura 7 – Conectores Mini-DIN de 6 pinos	11
Figura 8 – Formas de onda de uma fonte chaveada.....	13
Figura 9 – Esquemático do transformador	14
Figura 10 – Viper 22.....	15
Figura 11 – Acoplador Óptico TLP181.....	16
Figura 12 – Aplicação padrão do regulador <i>shunt</i> TL431.....	17
Figura 13 – FT232R.....	20
Figura 14 - MAX232.....	21
Figura 15 – ST485.....	22
Figura 16 – Isolamento galvânico dos sinais providos dos transceptores.....	23
Figura 17 – Isolamento galvânico dos sinais providos do barramento.....	24
Figura 18 – Isolamento galvânico do sinal de RTS do dispositivo TTL	24
Figura 19 – 7408.....	25
Figura 20 – Lógica de Transmissão/Recepção	26
Figura 21 – Circuito de acionamento do LED	26
Figura 22 – LM555	27
Figura 23 – Circuito de geração do sinal de RTS para o RS232.....	27
Figura 24 – 7432.....	28
Figura 25 – Circuito de geração de controle do ST485.....	29
Figura 26 – Placa de circuito impresso do Adaptador Isolado de Comunicação Serial	31
Figura 27 – Adaptador Isolado de Comunicação Serial pronto.	32
Figura 28 – Formas de onda das fontes de alimentação.....	32
Figura 29 – Teste do transceptor FT232	33
Figura 30 – Mensagens trocadas para o teste do transceptor FT232	34
Figura 31 – Conversão de níveis do MAX232 sentido barramento-dispositivo	35
Figura 32 – Conversão de níveis do MAX232 sentido dispositivo-barramento.....	35
Figura 33 – Conversão de níveis do ST485 sentido barramento-dispositivo	36
Figura 34 – Conversão de níveis do ST485 sentido dispositivo-barramento	36
Figura 35 – Comparação no acoplador óptico do sinal recebido pelo dispositivo USB.....	37
Figura 36 – Comparação no acoplador óptico do sinal transmitido pelo dispositivo USB.....	38
Figura 37 – Comparação no acoplador óptico do sinal recebido pelo dispositivo RS232.....	38
Figura 38 - Comparação no acoplador óptico do sinal transmitido pelo dispositivo RS232 ..	39
Figura 39 – Comparação no acoplador óptico do sinal recebido pelo dispositivo RS485.....	39
Figura 40 – Comparação no acoplador óptico do sinal transmitido pelo dispositivo RS485..	40
Figura 41 – Comparação no acoplador óptico do sinal de RTS do dispositivo TTL.....	40
Figura 42 – Transmissão RS232 e sinal gerado de RTS.....	41

Figura 43 – Diagrama de blocos do sistema de Teste em Aplicação Real.....	42
Figura 44 – Montagem do sistema de Teste em Aplicação Real	43

LISTA DE TABELAS

Tabela 1 – Pinagem dos conectores DB	5
Tabela 2 – Conexões na configuração Null Modem	6
Tabela 3 – Pinagem do conector Mini-DIN	12
Tabela 4 – Comportamento do sinal de controle do ST485	30

RESUMO

Nos últimos anos, o crescimento em diversidade de equipamentos eletrônicos e a evolução da tecnologia fizeram crescer a variedade dos padrões de comunicação e meios de conexão através dos quais podemos realizar troca de dados entre terminais. Para equipamentos domésticos, o padrão USB vem mostrando-se cada vez mais dominante em relação a padrões mais antigos, mas está longe de ser o único. Em aplicações industriais, diversas configurações diferentes de comunicação serial são utilizadas de acordo com as características necessárias para cada aplicação; podemos citar quatro entre as mais encontradas atualmente: a comunicação nos padrões RS232, RS485, USB e a comunicação com níveis TTL. Com isso, surge o problema da incompatibilidade entre dispositivos que utilizam diferentes configurações para comunicação e troca de dados. Para se superar esta dificuldade foi desenvolvido o Adaptador Isolado de Comunicação Serial, o qual permite a comunicação entre quaisquer dispositivos que utilizem qualquer uma das configurações citadas acima; por exemplo, podemos comandar um micro controlador que utiliza comunicação com níveis TTL através de um computador que utiliza padrão RS232 e, ao mesmo tempo, monitorar todas as informações trocadas através de um terminal com padrão USB. Outra aplicação possível para este adaptador é na área educacional, podendo ser utilizado em forma de *kit* para estudo e familiarização dos estudantes com estes diferentes padrões de comunicação.

Palavras chaves: Adaptador de Comunicação Serial; RS232; RS485; TTL; USB.

ABSTRACT

In recent years, growth in diversity and the evolution of electronics technology have increased the variety of patterns of communication and connection means through which we can perform data exchange between terminals. For domestic equipment, the standard USB is proving to be increasingly dominant in relation to older standards, but is far from unique. In industrial applications, several different configurations of serial communication are used in accordance with the characteristics needed for each application, we can mention among them the four most commonly found today: communication standards RS232, RS485, USB and communication with TTL levels. With that comes the problem of incompatibility between devices that use different settings for communication and data exchange. To overcome this difficulty was developed the Isolated Serial Communication Adapter, which allows communication between any devices that use any of the settings mentioned above; for example, it is possible to operate a micro controller that uses TTL levels through a computer that uses standard RS232 and at the same time, monitor any information exchanged via a standard USB terminal. Another possible application for this adapter is the educational area, where it can be used in a form of a kit for the students to study and familiarize with these different communication patterns.

Keywords: Universal Communications Adapter; RS232; RS485; TTL; USB.

1. Introdução

Com a evolução da eletrônica, foram surgindo gradativamente diversos tipos de equipamentos e dispositivos eletrônicos, cada qual reunindo as características necessárias específicas para sua aplicação. Uma das características que mais variou ao longo do tempo é a forma de comunicação.

Nos dias de hoje, uma das formas mais comuns de comunicação utilizada tanto domesticamente quanto na indústria é a comunicação serial, porém existem diversos padrões de comunicação serial utilizados no dia-a-dia, seja pela especificidade da aplicação ou pela evolução da tecnologia. Com isso surge o problema da incompatibilidade entre esses padrões, tornando assim complicada a comunicação entre dispositivos com padrões diferentes. Para superar este problema foi desenvolvido o Adaptador Isolado de Comunicação Serial, que permite a comunicação entre quaisquer dispositivos que utilizem qualquer uma das configurações de comunicação serial contempladas pelo projeto: RS232, RS485, TTL e USB.

Ao longo deste capítulo será dada uma breve introdução sobre comunicação de dados e serão detalhadas as configurações de comunicação serial acima citadas.

1.1 Comunicação de Dados

A Comunicação de Dados trata da comunicação entre diferentes dispositivos através de um meio comum de comunicação, tendo como principal função fornecer a maior taxa de transmissão possível, com o menor ruído e potência possíveis.

a) Canais de Comunicação

Um canal de comunicação é um caminho sobre o qual a informação trafega; no projeto em questão, os canais de comunicação são constituídos por fios que unem os dispositivos.

O canal *simplex* é o canal cuja direção de transmissão de dados é sempre a mesma.

O canal *half-duplex* é o canal no qual a direção pode ser revertida, ou seja, a transmissão de dados pode se dar nas duas direções, mas nunca simultaneamente.

O canal *full-duplex* é o canal no qual a transmissão de dados pode se dar em ambas as direções, simultaneamente. Pode-se considerar *full-duplex* também o conjunto de dois canais *simplex*, um para transmissão e um para recepção de dados (ZUFFO, 1981).

A distância que um sinal percorre em um canal de comunicação pode variar desde alguns milímetros, até algumas centenas de metros. Para distâncias pequenas, o dado pode ser transmitido diretamente. No entanto, freqüentemente os dados devem ser enviados para fora dos circuitos que constituem o computador, podendo ter enormes distâncias envolvidas no processo e com isso surgem os problemas da distorção do sinal ao longo dos condutores longos e do ruído externo, incorporado ao longo da linha de transmissão; portanto cada padrão de comunicação é utilizado de acordo com a necessidade de aplicação envolvida como, por exemplo, usa-se o RS485 para aplicações de longa distância devido a sua maior imunidade ao ruído, fato que será detalhado mais adiante. Com isso, pode-se constatar que a variedade de padrões pode ser grande, dependendo da aplicação que se pretende dar ao dispositivo.

b) Comunicação Serial

Ao processo de se enviar dados através de um canal de comunicação, um *bit* por vez, seqüencialmente, dá-se o nome de comunicação serial. Ou seja, os *bytes* a serem transmitidos são “divididos” em *bits* que são individualmente transmitidos um a um. Na outra ponta do canal de comunicação, o receptor conta os *bits* que chegam e quando formam os 8, os *bytes* são remontados, reconstituindo-se assim os *bytes* originais.

As portas seriais dependem de um *chip* controlador para funcionar corretamente. Este *chip* transforma em serial a saída paralela do barramento do computador, possibilitando assim a transferência via porta serial (CAMPBELL, 1990).

Um fator importante na comunicação serial é a taxa de transferência (*Baud Rate*); ela se refere ao número de sinais elétricos transmitidos por unidade de tempo. No padrão RS232, por exemplo, ocorre uma transição de sinal por *bit*, portanto a taxa de transferência (*Baud Rate*) e a taxa de *bit* (*Bit Rate*) são as mesmas; ou seja, uma taxa de 9600 *bauds* corresponde a uma transferência de 9600 *bits* por segundo pelo canal (ZANCO, 2008).

É fundamental que o circuito receptor dos dados saiba o momento exato para ler os *bits* desse canal, saiba exatamente quando um pacote começa, quando termina e quanto tempo decorre entre os *bits*; quando essa temporização for precisa, temos a sincronia entre transmissão e recepção, tornando assim possível a correta transferência de dados. Falha nessa sincronia pode causar perda ou leitura incorreta dos dados.

Existem dois tipos de sistemas que utilizam técnicas diferentes para garantir essa sincronia, os sistemas assíncronos e os síncronos.

Nos sistemas assíncronos, um único canal é o meio pelo qual a informação trafega, sendo assim necessária uma configuração prévia de taxa de transferência tanto no

transmissor, quanto no receptor; essa configuração é dada através de um *clock* interno similar nos dois dispositivos. De forma geral, os dados são enviados em pequenos pacotes de 10 ou 11 *bits*, dos quais 8 (um *byte*) constituem a mensagem. Em estado de repouso, o canal está sempre com nível lógico '1', o primeiro *bit* enviado (*start bit*) sempre tem nível lógico "0" e sinaliza que os 8 *bits* seguintes constituem a mensagem; após o *start bit*, o *clock* é ativado e a transmissão se dá na taxa pré-definida; após os 8 *bits*, a transmissão é concluída com o *bit* de parada (*stop bit*). A cada nova transmissão de pacote, o "*start bit*" inicia a sincronização, portanto pode-se ter uma pausa longa entre pacotes que mesmo assim teremos a sincronização garantida.

Nos sistemas síncronos, canais separados são usados para transmitir dados e informação de tempo; no canal de temporização, pulsos de *clock* são transmitidos para o receptor que lê o canal de dados e armazena o valor do *bit* encontrado naquele momento. O receptor somente lê novamente o canal de dados após a chegada de um novo pulso de *clock*. Os pulsos de *clock* e os dados são enviados pelo transmissor. Sendo assim, o receptor só efetuará a leitura quando o transmissor sinalizar, proporcionando assim a garantia da sincronização.

1.2 Padrões de Comunicação

Dentre as diversas configurações de comunicação serial, quatro das mais comumente utilizados no dia-a-dia são contempladas neste trabalho e serão detalhadas neste item.

1.2.1 RS232

RS232 nada mais é do que uma padronização de uma interface comum para troca serial de dados binários entre um DTE (terminal de dados) e um DCE (comunicador de dados), criada nos anos 60 por um comitê conhecido atualmente como EIA (Electronic Industries Association) (CAMPBELL, 1990). Naquela época, a comunicação se dava através da troca de dados digitais entre um computador central e terminais remotos de computador ou diretamente entre dois terminais que eram conectados através de linhas telefônicas, sendo assim necessário um modem de cada lado da linha sendo usado como decodificador. A partir dessas idéias e da necessidade de simplificação das formas de comunicação, surgiu o padrão RS232, que especifica as funções do sinal, temporizações, níveis de tensão e as devidas conexões mecânicas.

Ele funciona da seguinte maneira: como qualquer comunicação serial, os *bits* são enviados um a um, sequencialmente, e normalmente com o *bit* menos significativo (LSB) primeiro. A configuração mais utilizada é a do tipo “*start-stop* assíncrono”, portanto, por se tratar de comunicação assíncrona, cabe ao transmissor e ao receptor efetuarem o controle de tempo para saber o início e fim de cada *byte*, com isso tem-se que a velocidade de comunicação deve ser configurada previamente nos dois dispositivos.

O padrão RS232 utiliza normalmente dois sinais para controle de fluxo via *hardware*, o RTS (*ready to send*) e o CTS (*clear to send*); quando o transmissor irá iniciar um envio ele sinaliza através do pino RTS. O receptor, percebendo esse sinal, se prepara para receber o dado e quando está pronto, ele sinaliza através do pino CTS. Somente após receber o sinal CTS, o transmissor inicia a transmissão.

Para cada *byte*, temos como sinalizadores o *bit* de início (*start bit*) e o *bit* de parada (*stop bit*). O transmissor envia o *bit* de início, aguarda um tempo, e envia os próximos 8 *bits* mais o *bit* de parada, todos com a mesma duração. O receptor, após a primeira mudança de nível lógico provocada pelo *start bit* (“1” para “0”), sabe que uma seqüência de *bits* chegará. Como ele já conhece a velocidade de transmissão, ele só precisa aguardar o tempo de transmissão e efetuar a leitura dos *bits* até receber o *bit* de parada; quando a recepção se encerra ele fica aguardando o próximo *start bit* para nova recepção.

No padrão RS232, o nível lógico “0” corresponde a uma tensão entre 3V e 25V e o nível lógico “1” corresponde a uma tensão entre -3V e -25V, valores entre -3V e 3V são indefinidos e precisam ser evitados (ZUFFO, 1981). Esses níveis de tensão não são diretamente compatíveis com outros padrões de comunicação, sendo portando necessário o uso de conversores. Vale ressaltar que o padrão RS232 na sua configuração usual é do tipo *full-duplex*, com um fio para recepção de dados e outro para transmissão de dados.

A figura 1 mostra como a transmissão de um *byte* no RS232 ocorre:

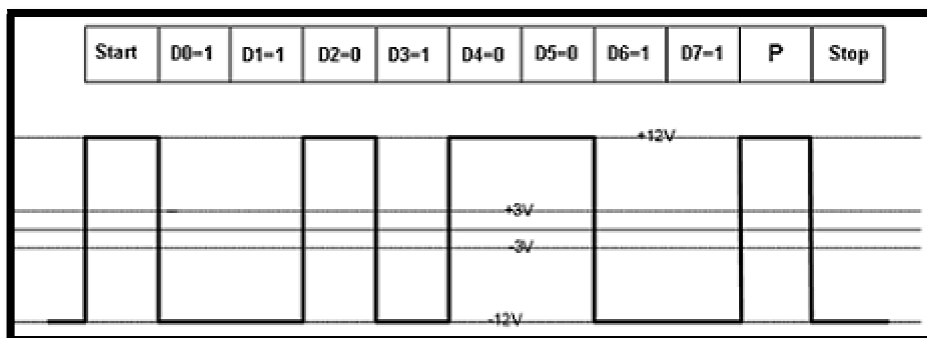


Figura 1 – Transmissão RS232
(ELETRONICA.ORG, 2008)

Tem-se, portanto, o *start bit* seguido dos 8 *bits* de dados mais o *stop bit*.

Os conectores normalmente utilizados no padrão RS232 são o DB9 e o DB25 que podem ser vistos na figura 2:

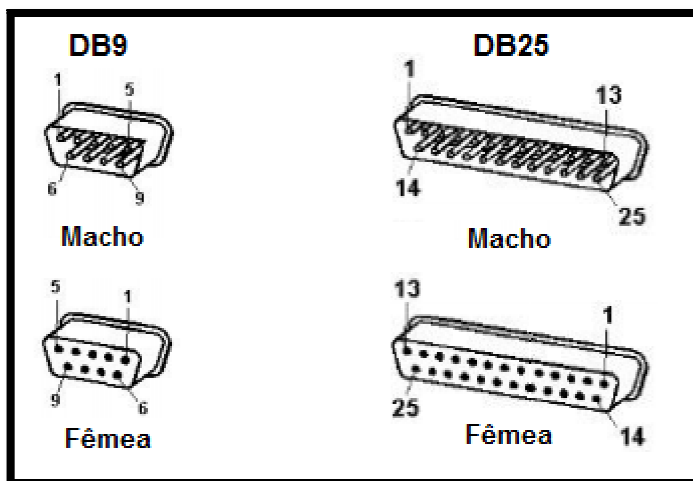


Figura 2 – Conectores DB
(LR, 2011)

Eles obedecem à seguinte pinagem:

Tabela 1 – Pinagem dos conectores DB
(LR, 2011)

Função RS232	Pinos DB9	Pinos DB25	Direção de Fluxo	
			Computador (DTE)	Modem (DCE)
CD	1	8	entrada	saída
RXD	2	3	entrada	saída
TXD	3	2	saída	entrada
DTR	4	20	saída	entrada
GND	5	7	----	----
DSR	6	6	entrada	saída
RTS	7	4	saída	entrada
CTS	8	5	entrada	saída
RI	9	22	entrada	saída

Onde,

GND = Terra

TXD ou TX = Transmissão de dados

RXD ou RX = Recepção de dados

DTR = Terminal de dados pronto

DSR = Conjunto de dados pronto

RTS = Pronto para enviar os dados

CTS = Envie os dados

CD = Portadora detectada

RI = Indicador de telefone tocando

Neste projeto, será utilizado o conector DB9, na configuração 3 pinos Null Modem, que permite a comunicação entre dois DTE:

Tabela 2 – Conexões na configuração Null Modem
(LR, 2011)

Conexão	DB9<->DB9
TXD<->RXD	3<->2
RXD<->TXD	2<->3
GND<->GND	5<->5

1.2.2 RS485

Um dos principais problemas do padrão RS232 é a sua baixa imunidade ao ruído. O padrão RS485 foi criado na década de 80 e, comparado ao RS232, é bastante robusto e vem sendo muito utilizado na indústria para comunicação multiponto.

O padrão RS485 é administrado pela Telecommunication Industry Association(TIA) , responsável pelo setor de comunicação da Electronic Industries Alliance (EIA); por isso o padrão RS485 é também conhecido atualmente como EIA-485.

Conforme descrito anteriormente, no RS232 os sinais são representados por níveis de tensão referentes a um terra, com um fio para transmissão e um fio para recepção de dados.

Devido a essa necessidade de um terra comum, uma das limitações é a possibilidade de interferências externas no sinal, portanto, fatores como comprimento do cabo utilizado e sua resistência são possíveis fatores geradores de ruído. O padrão RS485 utiliza um princípio diferente, ele é do tipo *half-duplex* no qual apenas dois fios são utilizados; quando o fio A for positivo e o fio B for negativo, tem-se nível lógico “1” e quando o inverso ocorre, tem-se nível lógico “0”. Portanto, o nível lógico é determinado pela diferença de tensão entre os fios, a isso se dá o nome de modo de operação diferencial.

Umas das vantagens desse modo de operação é sua robustez a ruídos e interferências, pois, quando um ruído é introduzido na linha, ele é induzido nos dois fios de modo que a diferença dessa interferência entre os fios A e B tende a ser quase nula, com isso o alcance do cabo pode chegar a aproximadamente 1200 metros enquanto o padrão RS232 tem, em sua máxima taxa de comunicação, a distância máxima de aproximadamente 15 metros. Quanto maior a distância a ser percorrida pelos dados menor será a taxa de transmissão; para distâncias curtas podemos ter para o padrão RS485 uma taxa de transmissão de 10Mbps (BLACK, 1988).

A figura 3 mostra a relação entre o comprimento do cabo e a taxa de de transmissão para o RS485:

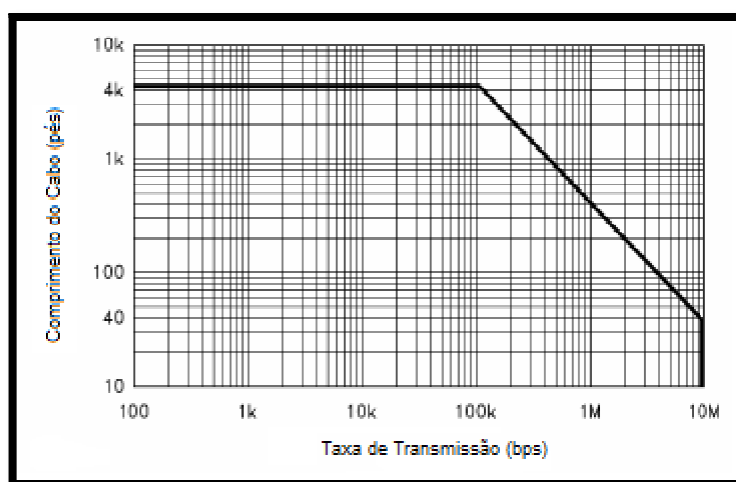


Figura 3 - Comprimento do cabo x Taxa de Transmissão para o RS485
(NATIONAL SEMICONDUCTORS, 1996)

O padrão RS485 não especifica um tipo de conector, portanto é possível a utilização de diversos modelos. Neste projeto foi utilizado um conector do tipo BR7, da fabricante Metaltex.

1.2.3 USB

O USB (*Universal Serial Bus*) surgiu através de uma parceria entre várias empresas de alta tecnologia, dentre as quais se destacam Microsoft, Apple, Hewlett-Packard, NEC, Intel e Agere.

História das Versões:

USB 0.7: Lançado em novembro de 1994.

USB 0.8: Lançado em dezembro de 1994.

USB 0.9: Lançado em abril de 1995.

USB 0.99: Lançado em agosto de 1995.

USB 1.0: Lançado em janeiro de 1996.

USB 1.1: Lançado em setembro de 1998.

USB 2.0: Lançado em abril de 2000.

USB 3.0: Lançado em setembro de 2009.

Quando a versão 1.0 foi lançada, ela possuía velocidade de 1,5Mbps (*low-speed*). O lançamento do padrão USB 1.1 em 1998, que possuía velocidades de 1,5Mbps a 12Mbps (*full-speed*), trouxe uma série de vantagens em relação ao antecessor pois, devido a uma interface única, deixou mais simples a tarefa de conectar diversos tipos de periféricos ao computador.

Após isso iniciou-se o processo de desenvolvimento de uma versão que pudesse alcançar velocidades muito superiores, culminando no ano 2000 com o lançamento da versão 2.0 (*high-speed*) que, apesar de compatível com as versões anteriores, continha alterações de topologia e possuía velocidade de tráfego de dados chegando ao limite de 480Mbps (480 milhões de *bits* por segundos), sendo assim muito superior ao seu antecessor (TORRES, 2011).

O *host* USB comunica-se com os demais dispositivos através do seu controlador, que pode ser encontrado na própria estrutura base da placa-mãe do computador ou pode ser adicionada num dos *slots* do barramento PCI.

É de responsabilidade do *host* detectar a inclusão e remoção de dispositivos, monitorar os sinais do barramento, gerenciar o fluxo de dados e fornecer alimentação aos dispositivos conectados quando necessário.

As primeiras versões USB utilizavam os Controladores *Host UHCI* (*Universal Host Controller Interface*) e *OHCI* (*Open Host Controller Interface*). A versão UHCI transfere parte do processamento do protocolo para o *software* (*driver*), deixando assim mais simples o controlador eletrônico; a versão OHCI faz o contrário, transferindo a maior parte do esforço para o controlador eletrônico, simplificando o controlador lógico (*driver*). O USB 2.0 utiliza o

Controlador *Host* EHCI (*Enhanced Host Controller Interface*), que é uma mescla das duas outras versões anteriores.

É através do protocolo que um ou mais dispositivos conversam e trocam informação com o *host*. No *host*, o protocolo está disponível em forma de *drivers* de arquivos, enquanto no dispositivo o protocolo pode ser encontrado dentro de um microcontrolador específico. Existem algumas versões de microcontroladores PIC que já possuem internamente o protocolo; para outras, ele deve ser previamente gravado. Outra forma possível é através de um *chipset* que carrega em si todas as funcionalidades do protocolo, poupando o microcontrolador do trabalho pesado.

Nas Portas Seriais e Paralelas é possível se comunicar com um dispositivo através dos sinais elétricos e um programa básico. No USB isso só é possível graças ao protocolo, sem ele não há comunicação; portanto, pode-se dizer que o protocolo é parte fundamental no USB.

Existem duas séries de conectores USB, a série A (figura 4) e a série B (figura 5):

Série A fêmea: encontrado no *host* (PC).

Série A macho: encontrado na extremidade do cabo USB; deve ser conectado ao *host* (PC).

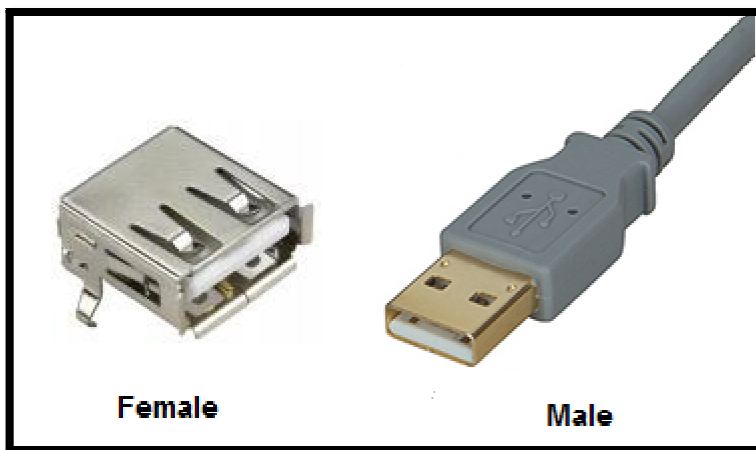


Figura 4 – Conectores USB série A
(NINGBO HI-SUN, 2011)

Série B fêmea: encontrado nos dispositivos do cliente tais como impressoras e câmeras digitais.

Série B macho: encontrado na extremidade do cabo USB; devendo ser conectado a um dispositivo.

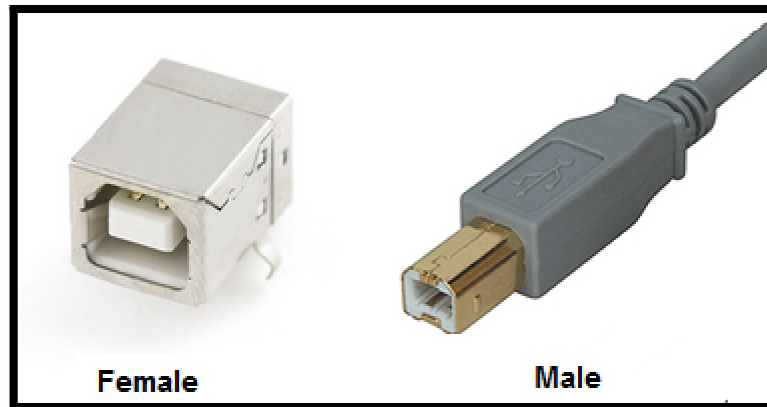


Figura 5 – Conectores USB série B
(NINGBO HI-SUN, 2011 e ELECTRONICS DIY, 2011)

O cabo USB é composto por quatro fios e uma malha para eliminação de ruídos. Os fios são:

- 1) VBus : fornece a alimentação (+5V) de periféricos quando necessário (*Bus-powered*).
- 2) D- : utilizado para transferência de dados (*data-*).
- 3) D+ : utilizado para transferência de dados (*data+*).
- 4) Gnd :terra

Todos os sinais de dados trafegam apenas pelos fios D+ e D- usando a codificação NRZI (*No Return to Zero Inverted*). O *bit* “1” é codificado através de uma transição entre os níveis de maior para menor voltagem e no sentido contrário, enquanto o *bit* “0” é codificado quando não há transição.

1.2.4 TTL

A família TTL opera no intervalo de 0 a +5V (TAUB, 1984). No padrão RS232 o nível lógico “0” é equivalente a uma tensão entre +3V e +25V e o nível lógico “1” é equivalente a uma tensão entre -25V e -3 V. Nos níveis TTL o nível lógico “0” é equivalente a uma tensão entre 0V e 0,8V e o nível lógico “1” é equivalente a uma tensão de entre +2,4 V e +5V. Assim como no RS232, o TTL tem como estado inicial nível alto, a transmissão se inicia com um *bit* “0”, o *start bit*. Logo após vem os 8 *bits* que compõem a mensagem e finaliza-se com um *bit* “1”, o *stop bit*.

A figura 6 mostra uma comparação entre transmissão no RS232 e TTL:

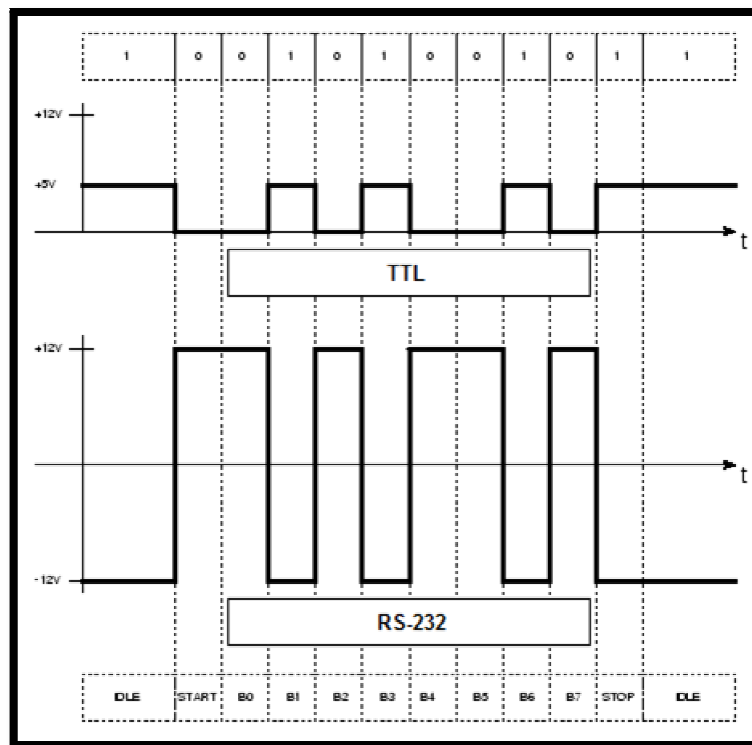


Figura 6 – Comparação TTL x RS232
(ARNE ROBOTICS, 2005)

A comunicação com níveis TTL pode ser utilizada com diversos conectores. Neste projeto foi utilizado o conector Mini DIN de 6 pinos (figura 7):

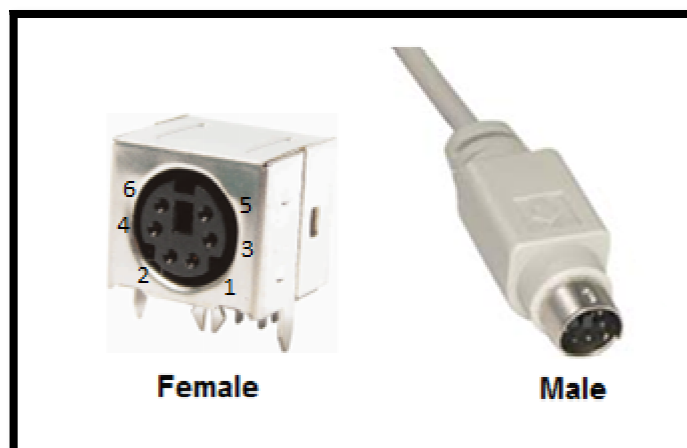


Figura 7 – Conectores Mini-DIN de 6 pinos
(METALTEX, 2011)

A tabela 3 mostra a pinagem utilizada com o conector Mini-DIN de 6 pinos:

Tabela 3 – Pinagem do conector Mini-DIN

Pino	Função
1	Terra
2	RTS (sinaliza a transmissão de dados)
3	Transmissão de dados
4	Recepção de dados
5	Não utilizado
6	Não utilizado

2. Desenvolvimento

O projeto do Adaptador Isolado de Comunicação Serial integra quatro partes principais: a fonte, os transceptores, o isolamento galvânico e a lógica de transmissão e recepção, incluindo os sinais de controle. Cada uma das etapas do desenvolvimento será detalhada ao longo deste capítulo.

2.1 Fonte de Alimentação

O projeto da fonte de alimentação é de extrema importância, pois ela será responsável por fornecer a tensão correta de funcionamento para todo o restante do circuito. Caso os níveis de tensão fornecidos pela fonte não sejam os projetados, pode-se ter um funcionamento incorreto dos componentes ou até mesmo danificá-los, causando danos maiores ao equipamento. Neste projeto, optou-se por uma fonte chaveada com duas saídas isoladas.

Optou-se pela fonte chaveada pelo fato de ser mais leve, menor, mais barata e de maior eficiência quando comparada a uma equivalente linear. Como pretendia-se deixar o projeto o mais compacto possível, uma fonte linear necessitaria de um transformador muito grande e pesado, inviabilizando o seu uso. Como a fonte chaveada trabalha em altas frequências, ela pode ser geradora de ruídos caso não seja projetada cuidadosamente.

A figura 8 mostra as formas de onda encontradas em um circuito de fonte chaveada:

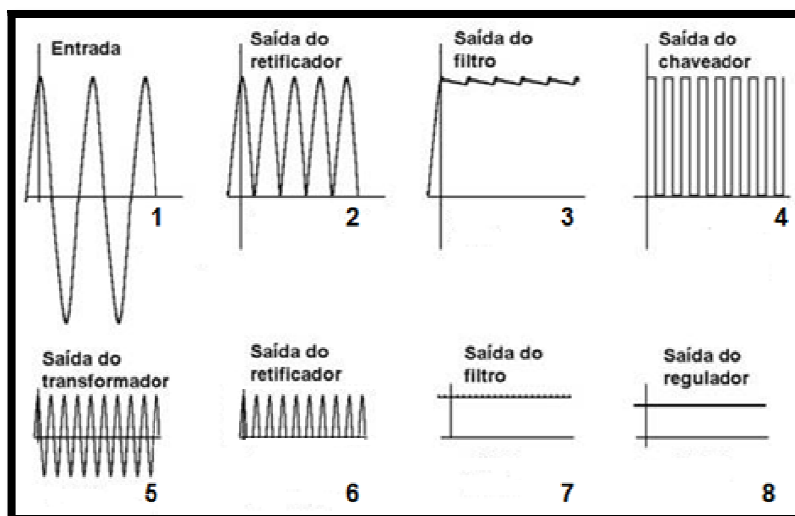


Figura 8 – Formas de onda de uma fonte chaveada
(VASCONCELOS, 2005)

Em “1” tem-se a tensão alternada vinda da rede, 127 V ou 220 V, com freqüência de 60Hz. O sinal da rede primeiramente passa por um retificador de onda completa e fica conforme mostrado em “2”, após isso é filtrado por um capacitor e o sinal resultante é mostrado em “3”. Com o auxílio de um elemento chaveador, o sinal é amostrado em alta freqüência e fica conforme mostrado em “4”; é este o sinal que entra no primário do transformador.

No secundário do transformador tem-se como saída o sinal visto em “5”, que tem valor reduzido e alta freqüência (freqüência do chaveador). Após isso o sinal é novamente retificado e o sinal resultante pode ser visto em “6”; este sinal é novamente filtrado conforme visto em “7” e por último, se necessário, passa por um regulador e o sinal resultante é mostrado em “8”. Desta forma temos completa a conversão de tensão alternada para contínua.

Para o projeto da fonte, foram utilizados os seguintes componentes principais:

a) Transformador

Para este projeto optou-se por um transformador de ferrite de alta freqüência produzido pela MGS Transformadores. A escolha deste tipo de transformador justifica-se pelo fato que será utilizado como elemento chaveador o componente Viper22, que chaveia em freqüência fixa de 60kHz e este transformador suporta freqüências de até 100kHz, com baixa perda e alta eficiência. A relação de espiras foi escolhida de modo que, tanto para 127VAC quanto 220VAC de tensão de entrada, a tensão de saída dos secundários seja sempre superior a 5V, de forma que ela possa ser regulada posteriormente. A figura 9 mostra o esquemático e a pinagem do transformador:

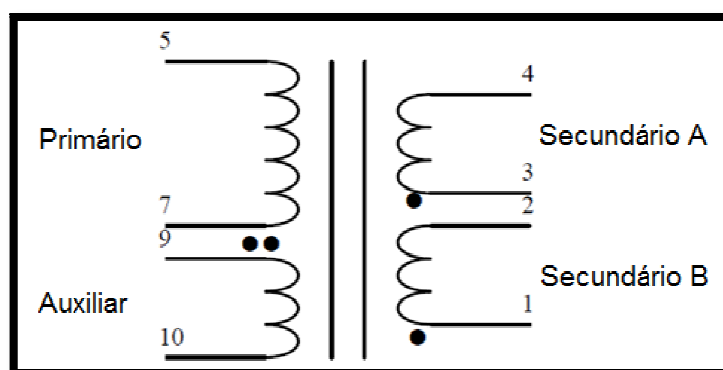


Figura 9 – Esquemático do transformador

Relação de espiras:

Pino 1 ao 2 - 16 espiras

Pino 3 ao 4 - 16 espiras

Pino 5 ao 7 - 248 espiras

Pino 9 ao 10 - 78 espiras

b) Viper 22

Como elemento chaveador, foi utilizado o circuito integrado Viper 22, da fabricante ST Microelectronics. Trata-se de um chaveador com frequência fixa de 60 kHz; a largura de pulso do chaveamento pode ser regulado com base na corrente que entra em seu pino de realimentação, com isso pode-se fazer o controle da tensão no secundário com o auxílio do circuito integrado TL431 e do opto-acoplador TLP181. Este controle será detalhado no item 2.1.1.

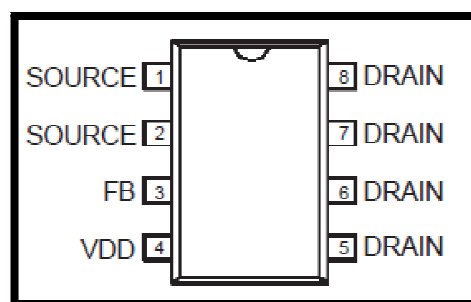


Figura 10 – Viper 22
(VIPER22 Datasheet, 2002)

A figura 10 mostra a configuração dos pinos do Viper 22. O pino VDD permite a alimentação do circuito como um todo e também promove uma corrente inicial de funcionamento do Viper através de uma fonte de corrente interna. Ele possui um comparador interno por histerese que compara o nível de tensão em VDD com níveis internos pré estabelecidos e nos fornece dois patamares de saída, VDD *on* e VDD *off*.

VDD *on* – Nível de tensão (tipicamente 14,5V) no qual o dispositivo começa o chaveamento e desabilita a fonte de corrente interna.

VDD *off* – Nível de tensão (tipicamente 8 V) no qual o dispositivo desliga o chaveamento e habilita a fonte de corrente interna.

O pino SOURCE está ligado à fonte do MOSFET de potência interno e também é o terra de referência do circuito.

O pino DRAIN está ligado ao dreno do MOSFET de potência interno e também é usado pela fonte de corrente interna para carregar o capacitor externo C13, responsável pela tensão inicial de funcionamento do Viper.

O pino FB é o pino de realimentação do circuito. Ele é responsável por indicar ao Viper 22 se a regulação de tensão no secundário está boa ou não, atuando no controle do circuito.

c) Acoplador Óptico

Acopladores ópticos são utilizados para transferir uma informação elétrica entre dois circuitos distintos através de luz, sem contato elétrico entre eles, promovendo assim o isolamento galvânico dos mesmos.

Neste projeto foi utilizado o acoplador óptico TLP181 (figura 11) do fabricante Toshiba. Ele é constituído por um LED e um foto-transistor. Possui tempo de resposta em torno de 2 μ s, tensão de condução direta do LED de 1,15V, corrente típica no LED quando em condução de 20mA e corrente do coletor do transistor de 1mA, corrente esta que é fundamental no controle da fonte que será detalhado no item 2.1.1.

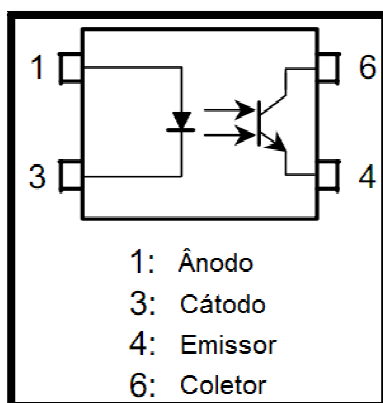


Figura 11 – Acoplador Óptico TLP181
(TLP181 Datasheet, 2002)

d) Regulador Shunt TL431

O regulador *shunt* TL431, do fabricante Fairchild Semiconductors, é um regulador de três terminais, com referência fixa em 2,5V. Uma aplicação padrão pode ser vista abaixo na figura 12 :

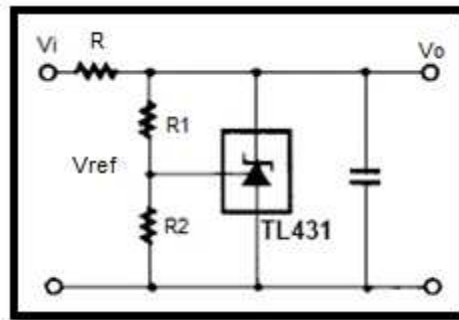


Figura 12 – Aplicação padrão do regulador *shunt* TL431
(TL431 Datasheet, 2003)

A tensão de saída V_o pode ser calculada através da equação 1:

$$V_o = \left(1 + \frac{R_1}{R_2}\right) \cdot V_{ref} \quad (1)$$

Em sua aplicação padrão, conforme a figura 12, caso a tensão em seu pino de referência (V_{ref}) for superior a 2,5V, o TL431 drena mais corrente, aumentando também a corrente que circula pelo resistor R. Como consequência do aumento de corrente, temos também o aumento da queda de tensão sobre R e com isso a tensão V_{ref} abaixa para 2,5V novamente. Caso a tensão em V_{ref} seja menor que 2,5V, ele opera de forma oposta, o TL431 drena menos corrente, abaixando assim a corrente que circula pelo resistor R. Como consequência da diminuição de corrente, temos também a diminuição da queda de tensão sobre R e com isso a tensão V_{ref} aumenta para 2,5V novamente.

No projeto da fonte do Adaptador Isolado de Comunicação Serial, o TL431 atua de forma semelhante, mas sem o uso do resistor R. Seu funcionamento no projeto será detalhado no item 2.1.1.

2.1.1 Funcionamento da Fonte

O esquemático da fonte de alimentação pode ser visto no Apêndice A.

A tensão alternada da rede, com frequência 60Hz, é retificada na ponte formada por D2, D3, D6 e D7 e filtrada pelo capacitor C9. A tensão resultante, já filtrada, é chaveada pelo Viper 22 com frequência de 60kHz. Esta tensão chaveada é vista na entrada do primário do transformador e induz nos outros enrolamentos do transformador uma tensão de alta frequência. O enrolamento entre os pinos 1 e 2 do transformador faz parte da malha de controle do sistema como um todo. A tensão neste enrolamento é retificada por D5 e depois filtrada; o valor desta tensão é dado pelo controle envolvendo basicamente os componentes TL431, TLP181 e Viper 22. Como a referência do TL431 é fixa em 2,5V e como se quer uma fonte de 5V, foi feito um divisor de tensão com os resistores R37 e R41, de valores iguais. Logo 5V_PC pode ser calculado conforme a equação 2:

$$5V_{PC} = \left(1 + \frac{10k}{10K}\right) \cdot 2,5 = 5V \quad (2)$$

Caso a tensão em 5V_PC seja maior que 5V, a tensão no pino de referência do TL431 será maior que 2,5V. Logo, ele drena mais corrente, acendendo com mais intensidade o LED do acoplador óptico TLP181; vale ressaltar que R35 é utilizado para limitar a corrente que passa no LED e o resistor R36 juntamente com o capacitor C11 são utilizados para dar maior estabilidade ao mesmo. Conforme aumenta a intensidade luminosa do LED, a corrente que circula pelo foto-transistor aumenta, esta corrente é a mesma que entra no pino de realimentação do Viper e sinaliza que ele deve fornecer menos energia ao primário do transformador alterando o ciclo de trabalho da tensão aplicada ao primário, ou seja, diminuindo a largura de pulso da onda quadrada e conseqüentemente diminuindo a tensão média no primário do transformador; essa técnica é conhecida por PWM (Modulação por Largura de Pulso). Com isso as tensões nos outros enrolamentos abaixam, tendendo a voltar para 5V em 5V_PC e conseqüentemente em 5V_ Board pois o enrolamento entre os pinos 1 e 2 do transformador e o enrolamento entre os pinos 3 e 4 do transformador possuem o mesmo número de espiras.

Caso a tensão no pino de referência do TL431 seja menor que 2,5V o oposto ocorre; o TL431 drena menos corrente, diminuindo a intensidade do LED do acoplador óptico e conseqüentemente diminuindo a corrente que entra no pino de realimentação do Viper, indicando que ele deve fornecer mais energia ao primário, aumentando a largura de pulso da onda quadrada e conseqüentemente aumentando a tensão média no primário do

transformador. Com isso, a tensão no secundário sobe e temos novamente os 5V desejados em 5V_PC e 5V_Board.

2.2 Transceptores

Os transceptores, ou *transceivers*, são responsáveis pela conversão dos diferentes padrões de comunicação serial para o nível utilizado em todo o circuito, 5V para nível alto e 0V para nível baixo, ou seja, nível TTL. Com isso, o dispositivo TTL conectado ao adaptador de comunicação serial não necessita de conversão de níveis para os seus sinais, portanto não necessita de um transceptor.

Neste item serão detalhados todos os transceptores utilizados no circuito. O esquemático completo de como eles são utilizados no projeto encontra-se no Apêndice B.

a) FT232R

O FT232R (figura 13), da fabricante Future Technology Devices International (FTDI), é uma interface USB para serial UART. Trata-se de um circuito integrado complexo, com muitas aplicações e um nível elevado de integração de funcionalidades.

Este dispositivo possui uma EEPROM interna de 1024 *bits* com cinco pinos de E/S que podem ser configurados para realizarem funções diversas.

Possui um gerador interno de *clock* que elimina a necessidade de cristais externos. Todo o protocolo USB está na memória interna do dispositivo, eliminando a necessidade de programação de *firmware*. Possui também internamente as resistências de terminação do barramento USB e um circuito para controle do sinal de *reset*, além de diversos outros recursos.

Em termos de velocidade, é totalmente compatível com a velocidade máxima do USB 2.0, com taxas que variam de 300 *Baud* a 3M *Baud*.

Cada *chip* possui um número de identificação único, gravado durante a fabricação do mesmo e que não pode ser alterado.

Neste projeto, o FT232R será utilizado na conversão de USB para serial UART, com níveis TTL utilizados no barramento.

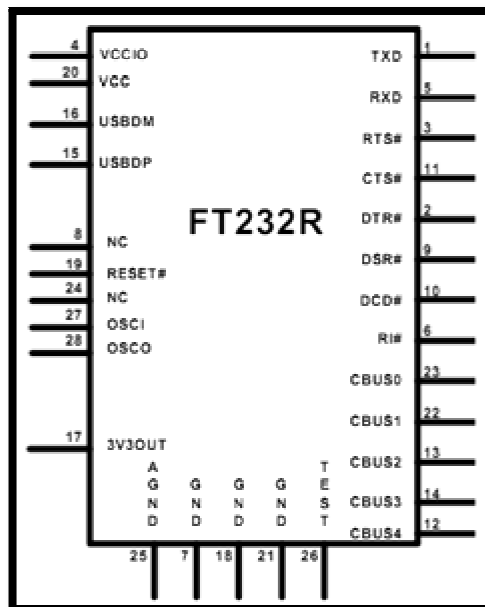


Figura 13 – FT232R
(FT232R Datasheet, 2005)

Foram utilizados os seguintes pinos para transmissão de dados: TxD, RxD, CBUS2, USBDM, USBDP.

Os pinos TxD e RxD trabalham nos níveis TTL. O pino TxD é responsável pela transmissão de dados para o barramento e o pino RxD é responsável pela recepção de dados do barramento.

Os pinos USBDM e USBDP trabalham no padrão USB. Funcionam como par diferencial de comunicação e são responsáveis pela transmissão e recepção de dados do USB.

O pino CBUS2 é configurado pela EPROM interna e neste projeto ele é utilizado em sua configuração padrão, ou seja, como TXDEN (*Transmission Data Enable*) que funciona como um sinal de controle para a interface RS485. Por padrão o pino TXDEN é ativado um período de *bit* anterior ao *start bit* e é desativado no mesmo instante do *stop bit*.

Os demais pinos utilizados são referentes à alimentação e configuração padrão do circuito integrado FT232.

b) MAX232

O MAX232 (figura 14) é um circuito integrado do fabricante MAXIM que converte os sinais Rx, Tx, CTS e RTS do padrão RS232 para níveis TTL e vice-versa. Ele consegue fornecer os níveis de tensão do padrão RS232 a partir de uma simples fonte de alimentação de 5V, com o auxílio de capacitores.

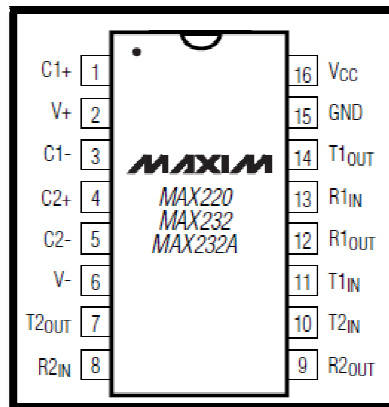


Figura 14 - MAX232
(MAX232 Datasheet, 2000)

Neste projeto, para o tráfego de dados, foram utilizados os pinos R1IN, R1OUT, T1IN e T1OUT. Os pinos R1IN e T1OUT trabalham no padrão RS232 e os pinos R1OUT e T1IN trabalham nos níveis TTL.

O pino R1IN é responsável pela recepção de dados do dispositivo padrão RS232; esses dados têm seus níveis convertidos para TTL pelo MAX232 e são transmitidos ao barramento pelo pino R1OUT.

O pino T1IN é responsável pela recepção de dados do barramento; esses dados têm seus níveis convertidos para o padrão RS232 pelo MAX232 e são transmitidos ao dispositivo pelo pino T1OUT.

Os demais pinos são referentes à alimentação do circuito e aos capacitores externos que auxiliam na conversão dos níveis de tensão.

c) ST485

O ST485 (figura 15) é um transceptor utilizado para controle da transmissão e recepção de dados de um dispositivo padrão RS485.

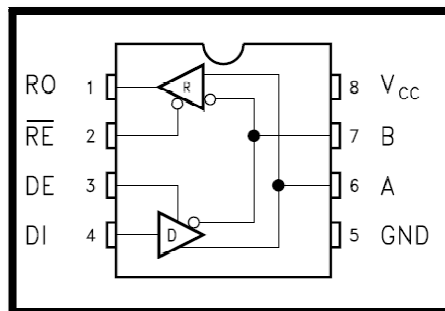


Figura 15 – ST485
(ST485 Datasheet, 2002)

O pino DI é responsável pela recepção dos dados vindos do barramento que serão enviados ao dispositivo padrão RS485. O pino RO é responsável pela transmissão para o barramento dos dados provindos do dispositivo. Os pinos A e B são responsáveis pela transmissão e recepção de dados do dispositivo padrão RS485. Pelo fato do padrão RS485 ser *half-duplex* e os pinos A e B funcionarem em conjunto como um par diferencial de comunicação, o dispositivo nunca transmite e recebe dados simultaneamente. Para controle desse fluxo de dados utilizamos os pinos complementares DE e RE; quando DE está em nível lógico “1” o pino RE está em nível lógico “0” e vice-versa, habilitando assim somente a transmissão ou a recepção de dados no circuito integrado ST485.

2.3 Isolamento Galvânico

Como forma de proteção dos equipamentos utilizados com o adaptador de comunicação serial e de usuários do mesmo, optou-se por fazer um circuito isolado galvanicamente entre o TTL, utilizado no barramento, e os demais padrões, prevenindo assim que possíveis curto-circuitos se alastrem colocando em risco a segurança de pessoas e dos demais equipamentos conectados.

Para este isolamento utilizamos novamente o acoplador óptico TLP181 da Toshiba que está especificado no item 2.1.

Conforme o exemplo da figura 16, os sinais de transmissão de dados provenientes dos transceptores, que estão todos referenciados à fonte 5V_PC, estão inicialmente em nível lógico “1”, colocando em condução o transistor BC547, acendendo o LED do acoplador óptico e conseqüentemente colocando o foto-transistor, que está referenciado à fonte 5V_Board, em condução. Com isso tem-se replicado o nível lógico “1” do sinal de transmissão de dados.

Quando um nível lógico “0” é transmitido, o transistor BC547 pára de conduzir, o LED do acoplador óptico se apaga e com isso temos replicado na saída o nível lógico “0” do sinal

de transmissão de dados. Temos com isso, de forma isolada, a reprodução total dos sinais dos transceptores no barramento.

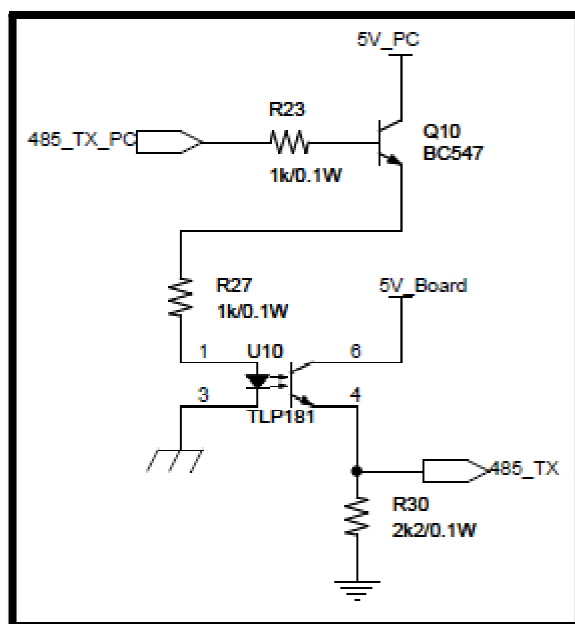


Figura 16 – Isolamento galvânico dos sinais providos dos transceptores

De forma semelhante, conforme a figura 17, os sinais de recepção de dados provenientes do barramento, que estão todos referenciados à fonte 5V_Board, estão inicialmente em nível lógico “1”, colocando em condução o transistor BC547, acendendo o LED do acoplador óptico e conseqüentemente colocando o foto-transistor, que está referenciado á fonte 5V_PC, em condução. Com isso temos replicado o nível lógico “1” do sinal de recepção de dados.

Quando um nível lógico “0” é recebido, o transistor BC547 pára de conduzir, o LED do acoplador óptico se apaga e com isso temos replicado na saída o nível lógico “0” do sinal de recepção de dados. Temos com isso, de forma isolada, a reprodução total dos sinais do barramento nos transceptores.

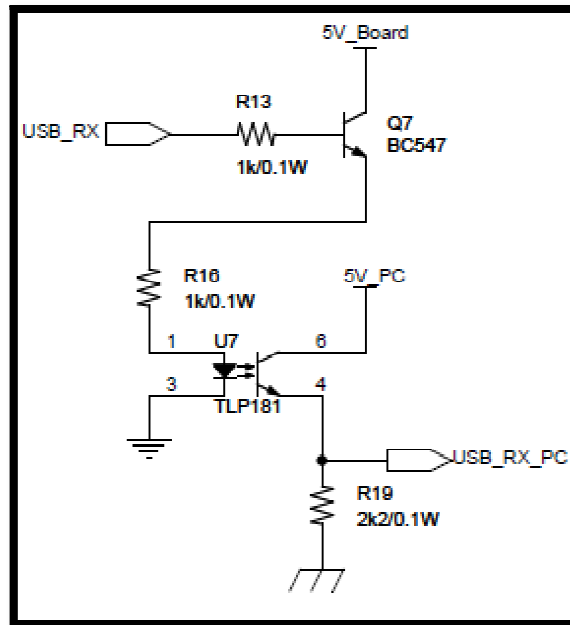


Figura 17 – Isolamento galvânico dos sinais providos do barramento

Vale ressaltar que para a comunicação nos níveis TTL os sinais de transmissão e recepção de dados não necessitam ser convertidos, pois são sempre utilizados com relação à fonte 5V Board. Portanto, conforme a figura 18 apenas o sinal de RTS é convertido pois no circuito de geração do controle para o ST485, que será detalhado no item 2.4, ele é utilizado referenciado à fonte 5V_PC.

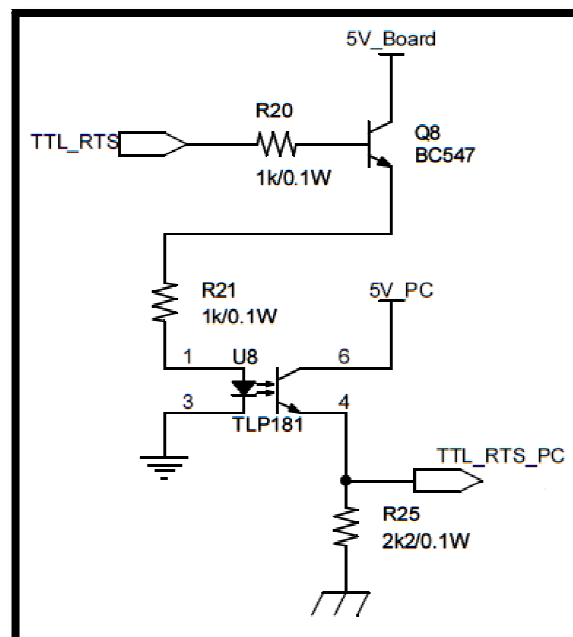


Figura 18 – Isolamento galvânico do sinal de RTS do dispositivo TTL

2.4 Lógica de Transmissão/Recepção e Sinais de Controle

a) Multiplexador RX e TX

Esta parte do projeto é responsável pela lógica de transmissão e recepção dos dados entre barramento e transceptores. Para construir esta lógica foram utilizados dois circuitos integrados 7408 (figura 19), que possui quatro portas lógicas do tipo AND de duas entradas cada.

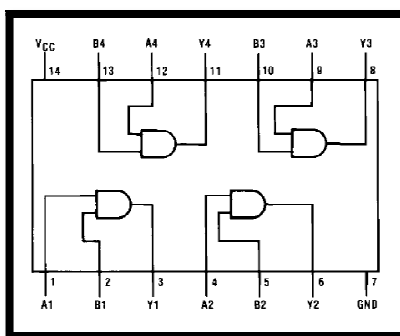


Figura 19 – 7408
(7408 Datasheet, 1998)

O princípio de operação do 7408 é uma operação lógica cuja saída terá nível lógico “1” se e somente se as duas entradas também possuírem nível lógico “1”, ou seja, $Y=AB$.

O funcionamento da lógica de transmissão/recepção se dá explicada a seguir.

Conforme o exemplo da figura 20, o canal de recepção de dados de um dispositivo está na saída da lógica envolvendo as portas AND. Os sinais de transmissão provenientes dos outros dispositivos estão na entrada da lógica envolvendo as portas AND; como inicialmente os sinais de transmissão estão em nível lógico “1”, quando um *start bit* seguido pelos demais *bits* é enviado por qualquer um dos dispositivos, este sinal é replicado na saída das portas AND. Portanto, um dispositivo recebe os dados provenientes das transmissões de todos os outros dispositivos, menos da transmissão dele próprio. Este sinal recebido é novamente convertido e referenciado à 5V_PC e é enviado ao pino de recepção do respectivo transceptor; com exceção do TTL pois, conforme citado no item 2.2, ele não necessita de transceptor e sua transmissão e recepção são sempre referenciados à fonte 5V_Board.

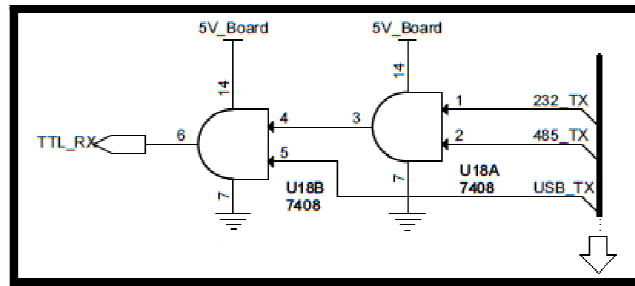


Figura 20 – Lógica de Transmissão/Recepção

Para melhor sinalização da transmissão de dados, optou-se por acender um LED toda vez que um dado é transmitido, como mostra a figura 21:

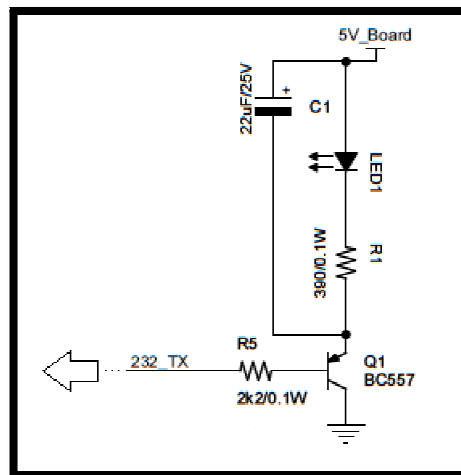


Figura 21 – Circuito de acionamento do LED

Toda vez que um *bit* "0" está no canal de transmissão, fato que somente ocorre quando um dado é transmitido, o transistor BC557 entra em condução e uma corrente circula acendendo o LED. Foi utilizado um capacitor para segurar o LED aceso por um período maior, sendo assim possível notar claramente que a transmissão está sendo efetuada.

b) Geração de RTS para o RS232

Como o sinal de RTS provindo do RS232 não tem como padrão operar em qualquer aplicação e como é necessário um sinal de controle que sinalize sempre que o dispositivo envia um dado, foi criado um novo sinal de RTS que fica ativo sempre que um dado é transmitido. Para isto foi utilizado como elemento principal o circuito integrado LM555.

O LM555 (figura 22) utilizado no projeto, fabricado pela Fairchild Semiconductor, é um circuito integrado extremamente versátil, utilizado em aplicações como temporizador ou oscilador.

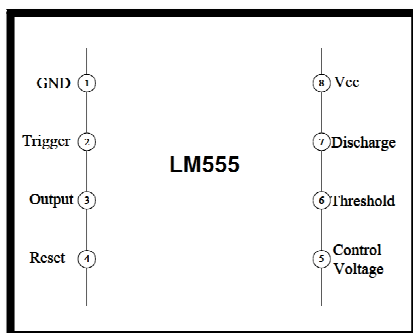


Figura 22 – LM555
(LM555 Datasheet, 2002)

Neste projeto, o circuito integrado LM555 será responsável por gerar um sinal toda vez que o dispositivo padrão RS232 transmite um dado.

Quando um *start bit*, que tem sempre nível lógico “0”, é enviado pelo dispositivo RS232, temos nível lógico “0” no pino 232_TX_PC provindo do MAX232. Conforme a figura 23, este sinal entra no *trigger* do LM555 e em sua saída temos um pulso de duração suficiente para que o *byte* seja enviado por completo.

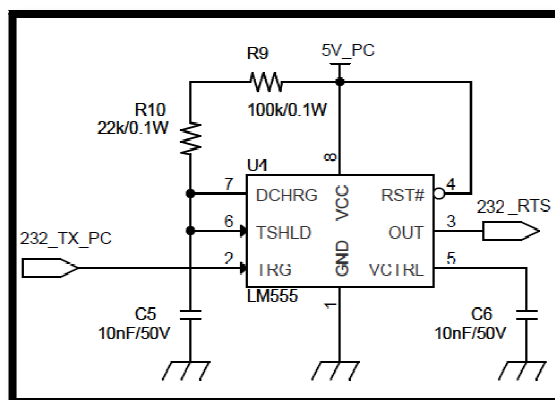


Figura 23 – Circuito de geração do sinal de RTS para o RS232

O período do temporizador pode ser calculado através da equação 3:

$$T = 1,1 \cdot (R9 + R10) \cdot C5 = 1,34ms \quad (3)$$

Com isso, é gerado um pulso de RTS do dispositivo RS232 com duração de 1,34ms.

A duração da transmissão de um *byte* com velocidade padrão de 9600 *bauds* pode ser calculada pela através da equação 4:

$$T2 = \frac{1}{9600} \times 10 = 1,04ms \quad (4)$$

Portanto, o sinal gerado fica em nível lógico alto durante tempo suficiente para que o *byte* seja transmitido por completo.

c) Geração do controle para o ST485

O padrão RS485 trabalha com um mesmo par diferencial de comunicação tanto para transmissão quanto para recepção de dados, portanto, ele apenas transmite os dados aos outros dispositivos ou apenas recebe os dados dos mesmos. Para este controle de fluxo, conforme citado no item 3.2, o transceptor ST485 utiliza-se de dois pinos complementares DE e RE. Quando o sinal de RTS, que indica transmissão de dados, proveniente de qualquer um dos outros dispositivos fica em nível lógico “1”, o transceptor ST485 habilita somente a recepção de dados providos dos outros dispositivos; quando nenhum outro dispositivo está transmitindo, o sinal de controle do transceptor está em nível lógico “0” e a transmissão dos dados do dispositivo RS485 para os demais é permitida.

Para a lógica de criação deste sinal de controle no ST485, utilizamos o circuito integrado 7432 (figura 24), que possui quatro portas lógicas do tipo OR de duas entradas cada.

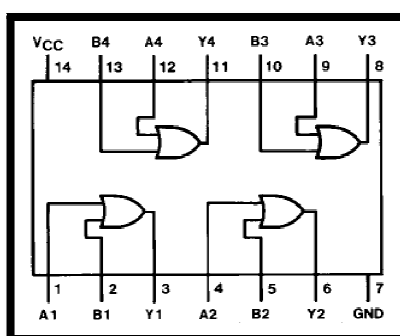


Figura 24 – 7432
(7432 Datasheet, 1986)

O princípio de operação do 7432 é uma operação lógica cuja saída terá nível lógico “0” se e somente se as duas entradas também possuírem nível lógico “0”, ou seja, $Y=A+B$.

O circuito de geração do controle para o ST485 como um todo é mostrado na figura 25:

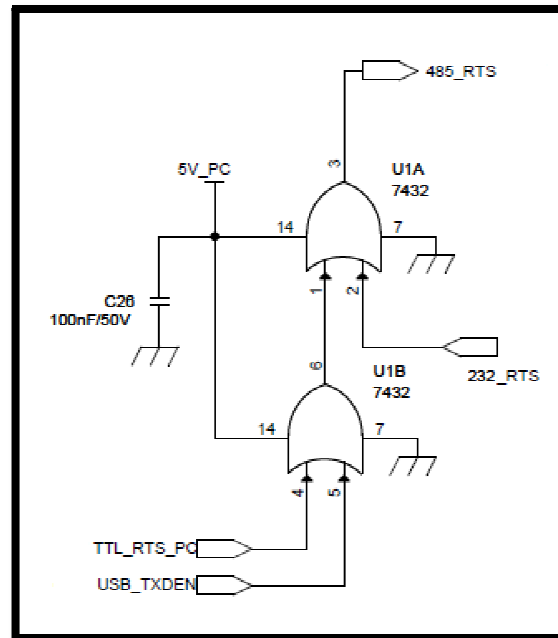


Figura 25 – Circuito de geração de controle do ST485

O funcionamento do circuito é bem simples; quando o dispositivo USB transmite, o sinal USB_TXDEN, que funciona como um sinal de RTS do USB, fica em nível lógico “1”; quando o dispositivo TTL transmite, o sinal TTL_RTS_PC, que funciona como um sinal de RTS do TTL, fica em nível lógico “1” e quando o dispositivo RS232 transmite, o sinal proveniente da saída do oscilador LM555, que funciona como um sinal de RTS do RS232, também fica em nível lógico “1”. Estes sinais entram nas portas OR do 7432 e na saída (pino 3 do 7432) temos um sinal controle do ST485, o 485_RTS, cujo comportamento pode ser visto na tabela 4.

Tabela 4 – Comportamento do sinal de controle do ST485

RTS USB	RTS TTL	RTS RS232	Controle ST485
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Com isso, a transmissão de dados do dispositivo RS485 é habilitada pelo transceptor ST485 se e somente se nenhum dos outros dispositivos estiver transmitindo dados.

3. Resultados e Discussões

Os resultados obtidos com a construção do protótipo, testes e simulações foram extremamente positivos, funcionando totalmente de acordo com o esperado. Neste capítulo será mostrado o protótipo construído e serão detalhados os resultados encontrados em todas as partes do mesmo.

3.1 Protótipo

Para auxílio na confecção do protótipo utilizamos o *software* OrCAD Capture e OrCAD PCB Editor, ambos da fabricante Cadence. O *software* OrCAD Capture foi utilizado na confecção do esquema elétrico completo do projeto e o *software* OrCAD PCB Editor foi utilizado para gerar o *layout* da placa de circuito impresso confeccionada.

A placa de circuito impresso pronta pode ser vista abaixo, na figura 26.

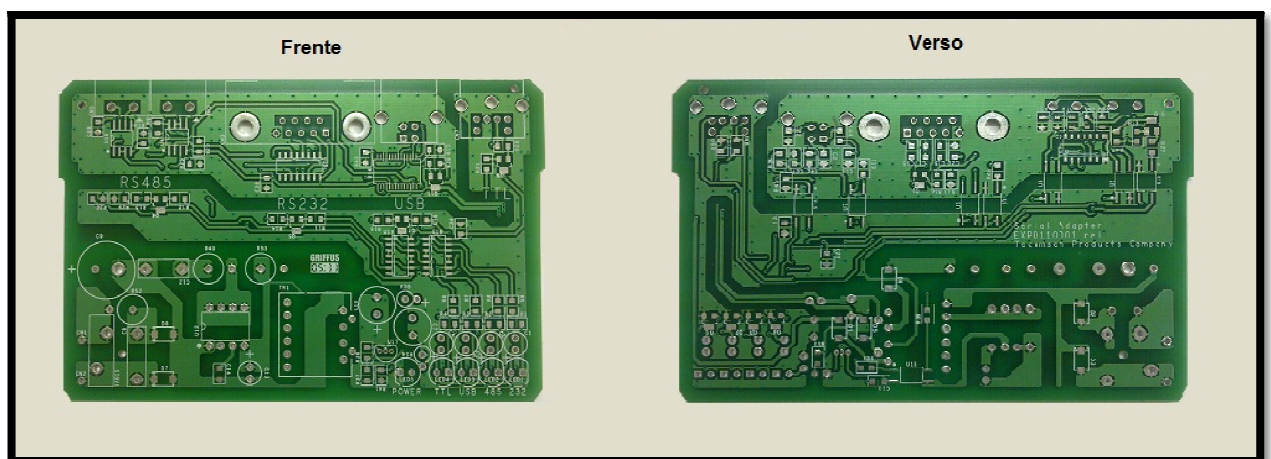


Figura 26 – Placa de circuito impresso do Adaptador Isolado de Comunicação Serial

O protótipo do adaptador de comunicação serial pronto pode ser visto abaixo, na figura 27:



Figura 27 – Adaptador Isolado de Comunicação Serial pronto.

3.2 Fonte de Alimentação

A fonte de alimentação foi testada tanto para tensão alternada de entrada de 127V quanto de 220V e os resultados obtidos foram os mesmos: em 5V_PC foi obtido o valor de 5,27 V e em 5V_Board foi obtido o valor de 5,29 V, em ambas situações o ripple encontrado é praticamente nulo. Os valores permaneceram estáveis em todas as situações de comunicação.

As formas de onda obtidas em 5V_PC e em 5V_Board podem ser vistas na figura 28:

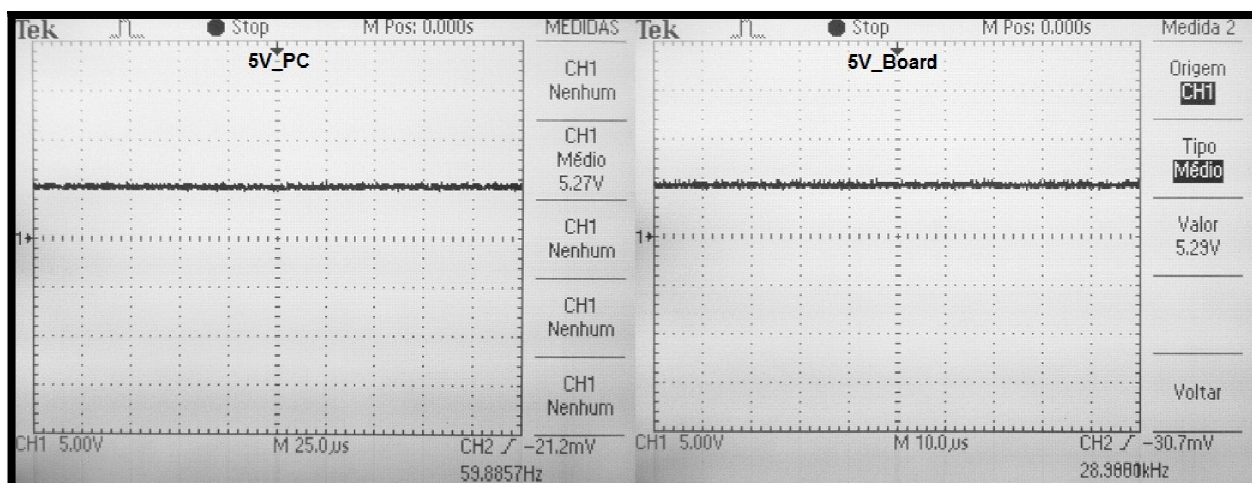


Figura 28 – Formas de onda das fontes de alimentação

Com isso tem-se garantida a correta alimentação do circuito durante todas as situações de funcionamento do adaptador de comunicação.

3.3 Transceptores

Nesta seção, como forma de demonstração de funcionamento dos transceptores, serão mostradas as formas de ondas tanto na recepção quanto na transmissão dos dispositivos padrões RS232 e RS485. As formas de onda referentes aos dados enviados e recebidos pelo dispositivo USB não são de fácil visualização, pois o USB consiste de uma estrutura complexa, com diversas informações sendo enviadas ao mesmo tempo, tais como informações de protocolo. Portanto, para demonstração do funcionamento do transceptor FT232R, foi efetuado um teste auxiliar que será detalhado no item “a” a seguir.

a) FT232R

Para demonstração do funcionamento do transceptor FT232R foi feita uma comunicação entre dois *notebooks* utilizando o Adaptador Isolado de Comunicação Serial, um deles conectado na porta padrão RS232 e outro na porta padrão USB. Com o auxílio do *software* Acess Port v1.34 da empresa BrotherSoft, foram enviadas mensagens pelo *notebook* conectado na porta padrão RS232, verificada a reprodução no *notebook* conectado na porta padrão USB e vice-versa.

A figura 29 ilustra o teste realizado:

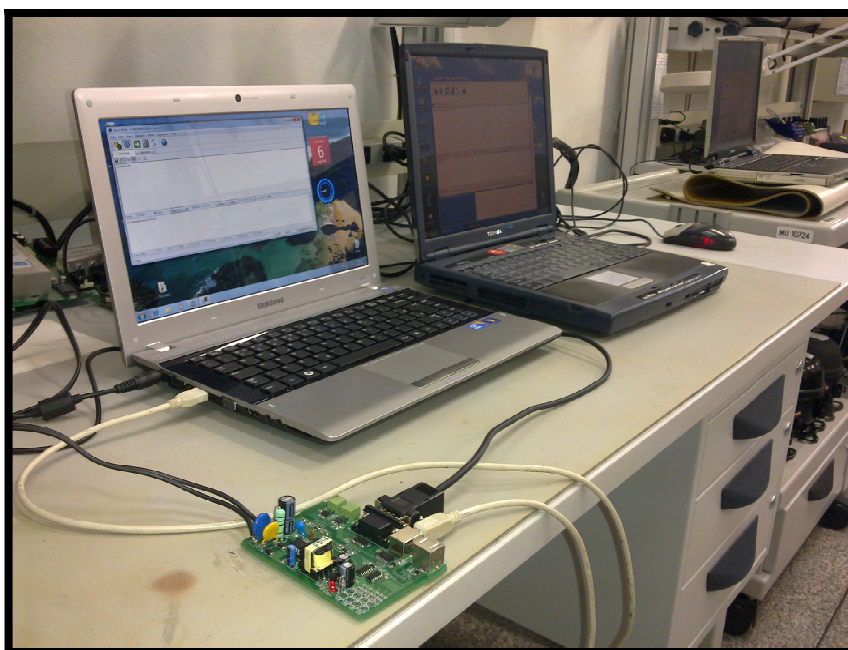


Figura 29 – Teste do transceptor FT232

Na figura 30 pode-se ver as mensagens sendo corretamente reproduzidas entre os *notebooks*, temos portanto o correto funcionamento do transceptor, sem perda de nenhum caractere e conseqüentemente de nenhum *byte* enviado.

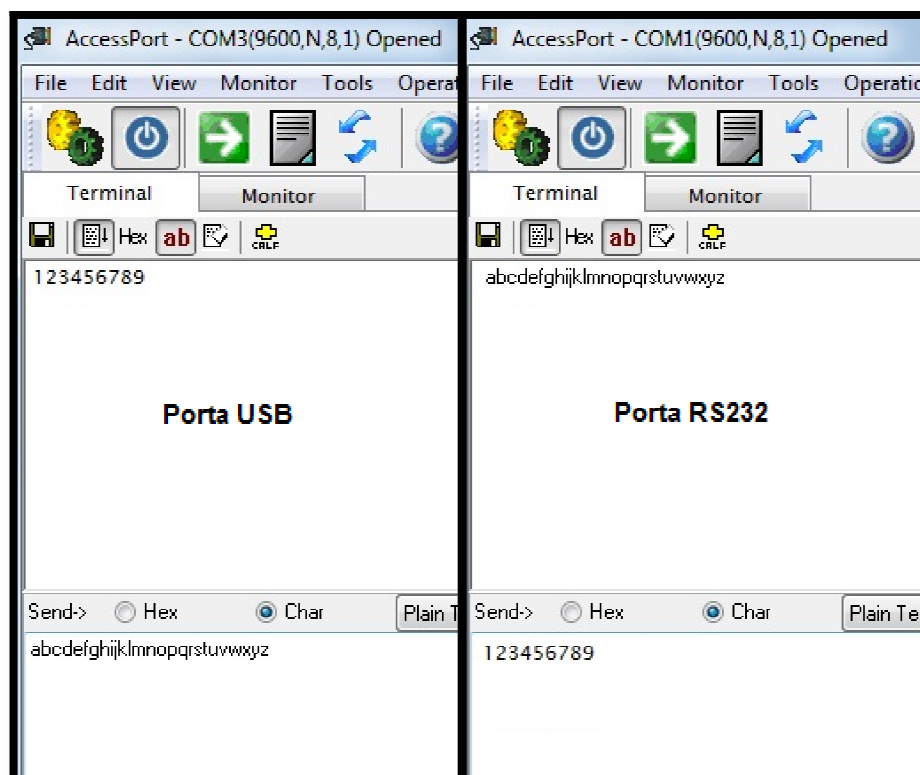


Figura 30 – Mensagens trocadas para o teste do transceptor FT232

b) MAX232

Como forma de melhor ilustrar o funcionamento do transceptor MAX232, com o auxílio de um *notebook*, primeiramente foi enviado o caractere “a” para o dispositivo padrão RS232 e, na figura 31, pode-se ver as formas de onda nos pinos 11 (níveis TTL) e 14 (padrão RS232) do transceptor.

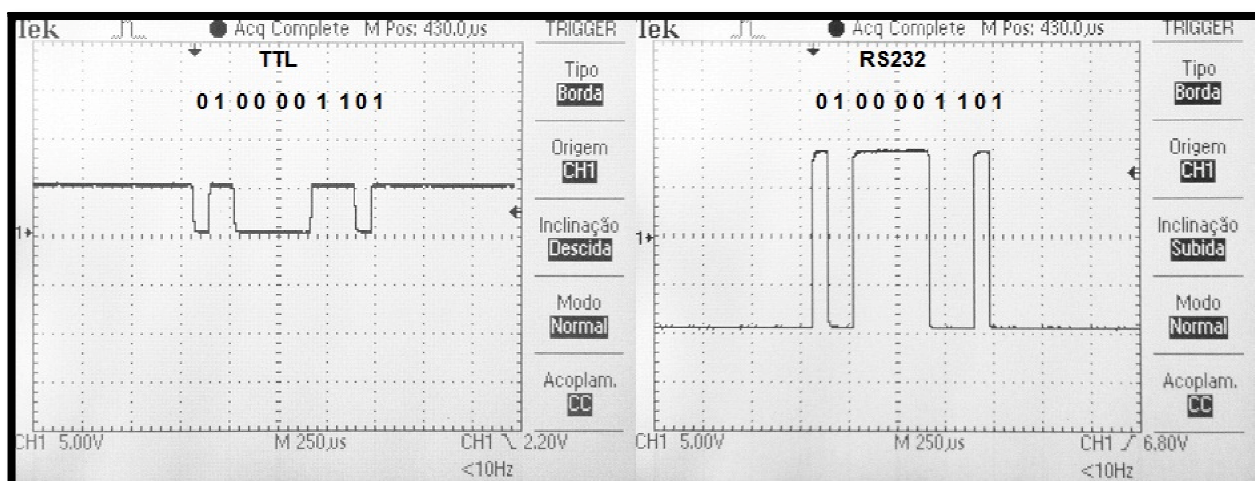


Figura 31 – Conversão de níveis do MAX232 sentido barramento-dispositivo

Após isso, foi enviado o mesmo caractere “a” pelo dispositivo padrão RS232 e, na figura 32, pode-se ver as formas de onda nos pinos 13 (padrão RS232) e 12 (níveis TTL) do transceptor.

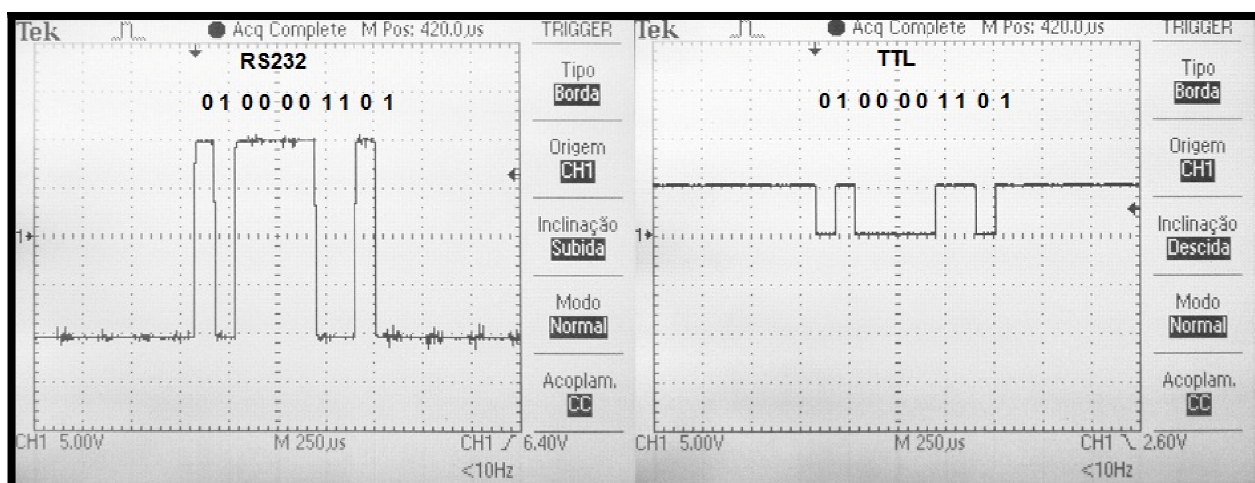


Figura 32 – Conversão de níveis do MAX232 sentido dispositivo-barramento

Pode-se notar que as pequenas distorções nos sinais causadas por interferências e pela resposta lenta do acoplador óptico não descaracterizaram os *bits*, desta forma pode-se verificar a correta conversão de níveis feita pelo transceptor MAX232 e a transmissão do caractere “a” (01100001 em binário), do *bit* menos significativo ao mais significativo, precedido pelo *start bit* (*bit* “0”) e sucedido pelo *stop bit* (*bit* “1”).

c) ST485

Para melhor ilustrar o funcionamento do transceptor ST485, com o auxílio de um *notebook*, primeiramente foi enviado o caractere “b” para o dispositivo padrão RS485 e, na figura 33, pode-se ver as formas de onda nos pinos 4 (níveis TTL) e no par diferencial composto pelos pinos 6 e 7 (padrão RS485) do transceptor.

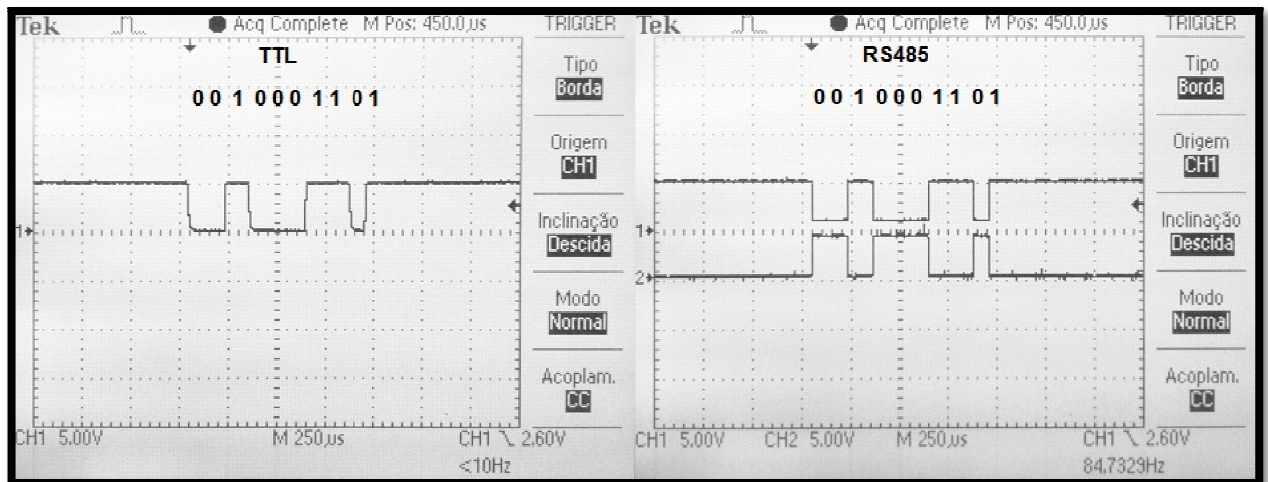


Figura 33 – Conversão de níveis do ST485 sentido barramento-dispositivo

Após isso, foi enviado o caractere “C” pelo dispositivo padrão RS485 ; para tal envio utilizou-se um adaptador RS485-RS232 e um *notebook*.

Na figura 34 pode-se ver as formas de onda no par diferencial composto pelos pinos 6 e 7 (padrão RS485) e no pino 1 (níveis TTL) do transceptor).

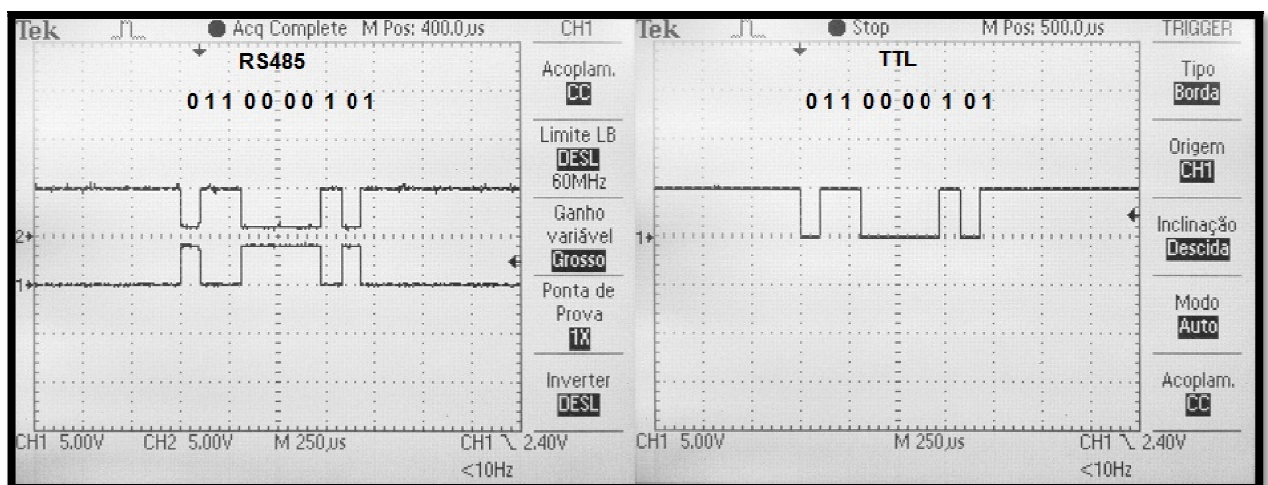


Figura 34 – Conversão de níveis do ST485 sentido dispositivo-barramento

Portanto, na análise dos resultados do transceptor ST485, pode-se notar que as pequenas distorções nos sinais causadas por interferências e pela resposta lenta do acoplador óptico não descaracterizaram os *bits*, desta forma pode-se verificar a correta conversão de níveis feita pelo transceptor e a correta transmissão dos caracteres “b” (01100010 em binário) e “C” (01000011 em binário) do *bit* menos significativo ao mais significativo, precedido pelo *start bit* (*bit* “0”) e sucedido pelo *stop bit* (*bit* “1”).

3.4 Isolamento Galvânico

Para teste de funcionamento do circuito de isolamento galvânico, foram comparados os sinais de transmissão e recepção dos dispositivos padrões USB, RS232 e RS485, bem como o sinal de RTS do dispositivo que utiliza níveis TTL, referenciados às duas fontes diferentes, 5V_Board e 5V_PC.

a) USB

Na figura 35 pode-se ver a comparação do sinal recebido do barramento pelo dispositivo USB antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_Board e à fonte 5V_PC respectivamente.

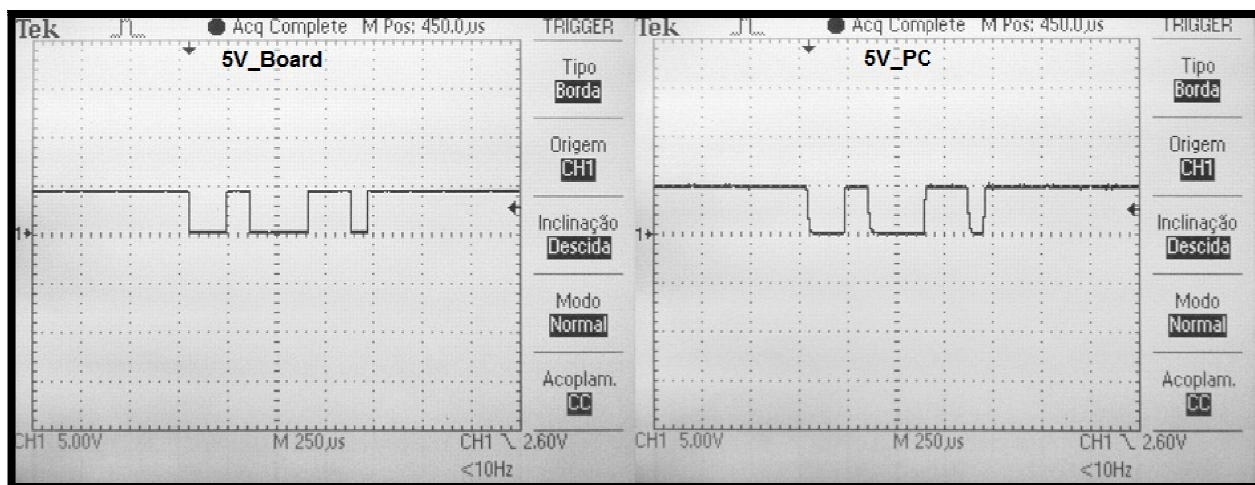


Figura 35 – Comparação no acoplador óptico do sinal recebido pelo dispositivo USB

Na figura 36 pode-se ver a comparação do sinal transmitido pelo dispositivo USB para o barramento antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_PC e à fonte 5V_Board respectivamente.

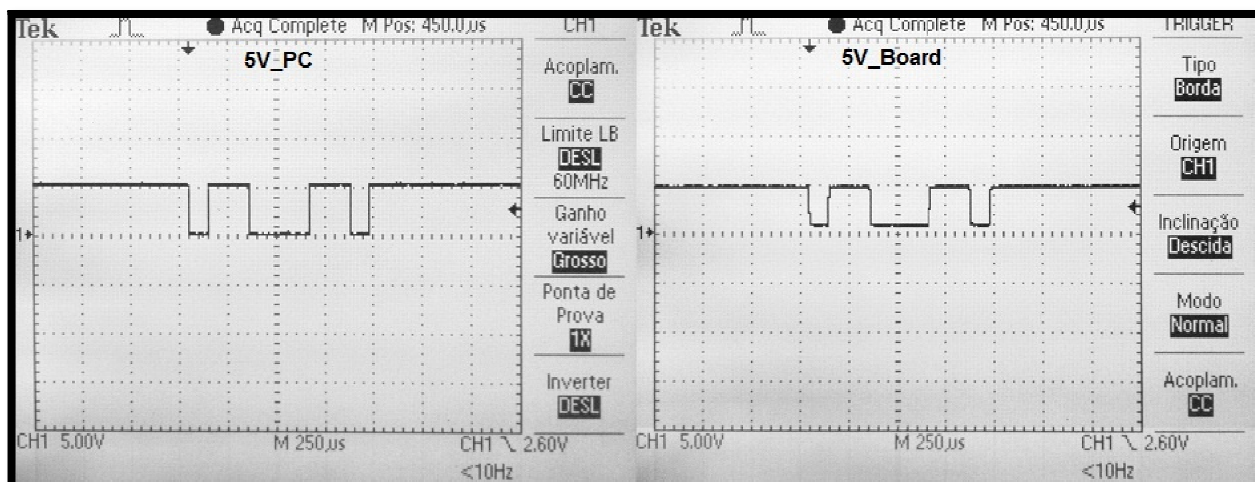


Figura 36 – Comparação no acoplador óptico do sinal transmitido pelo dispositivo USB

b) RS232

Na figura 37 pode-se ver a comparação do sinal recebido do barramento pelo dispositivo RS232 antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_Board e à fonte 5V_PC respectivamente.

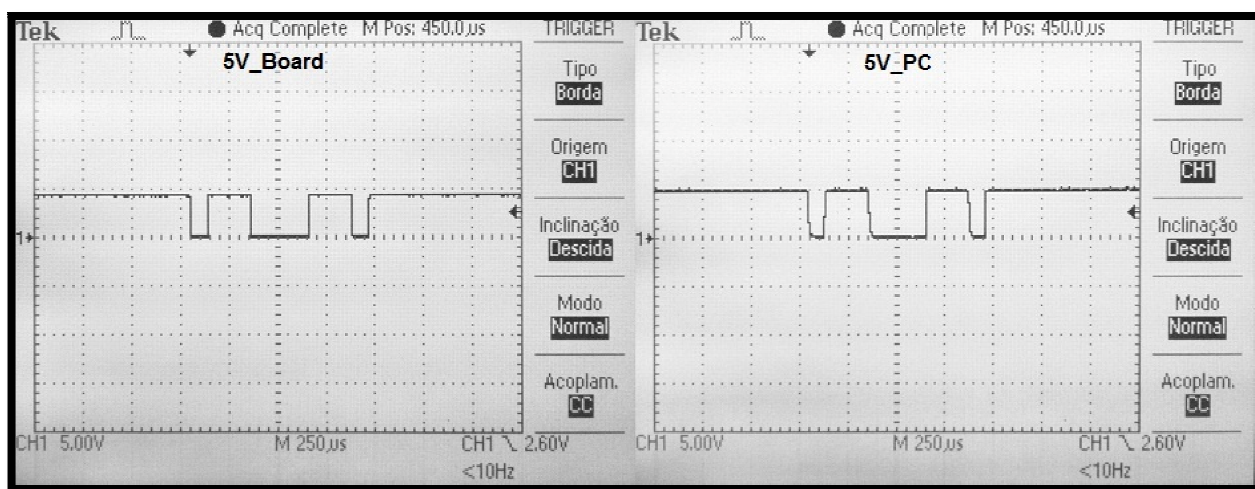


Figura 37 – Comparação no acoplador óptico do sinal recebido pelo dispositivo RS232

Na figura 38 pode-se ver a comparação do sinal transmitido pelo dispositivo RS232 para o barramento antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_PC e à fonte 5V_Board respectivamente.

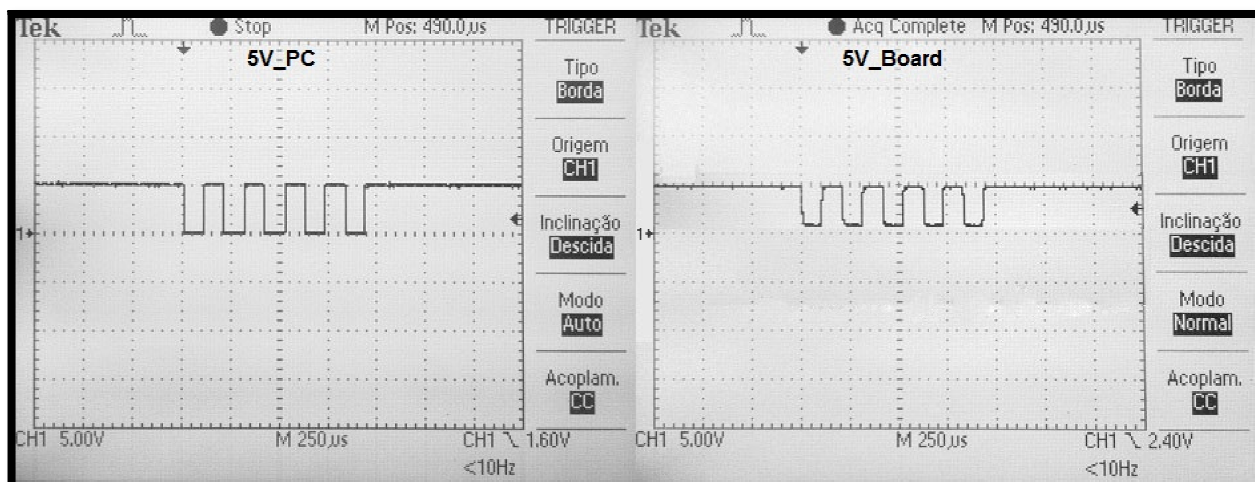


Figura 38 - Comparação no acoplador óptico do sinal transmitido pelo dispositivo RS232

c) RS485

Na figura 39 pode-se ver a comparação do sinal recebido do barramento pelo dispositivo RS485 antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_Board e à fonte 5V_PC respectivamente.

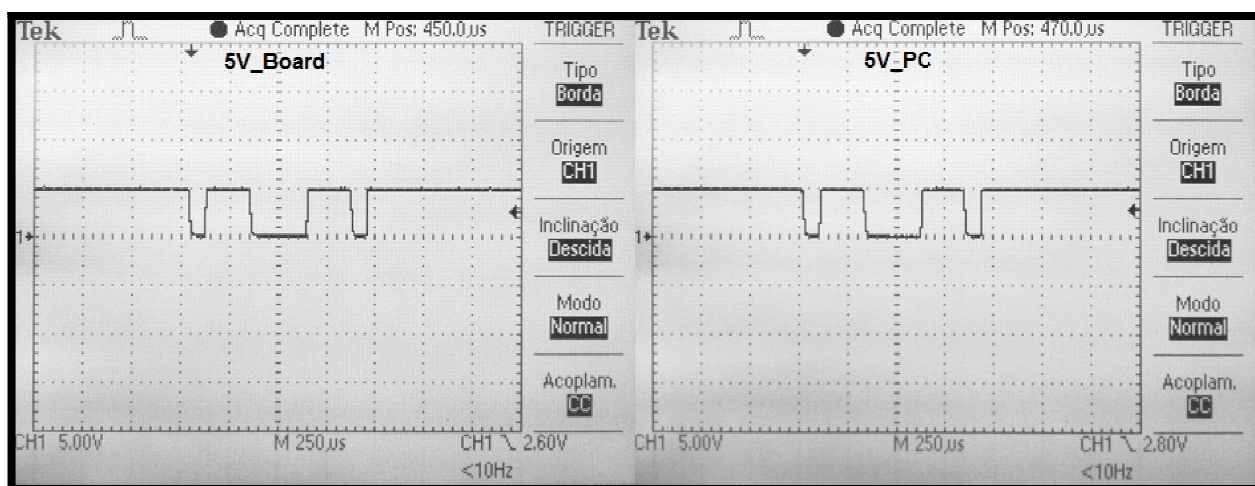


Figura 39 – Comparação no acoplador óptico do sinal recebido pelo dispositivo RS485

Na figura 40 pode-se ver a comparação do sinal transmitido pelo dispositivo RS485 para o barramento antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_PC e à fonte 5V_Board respectivamente.

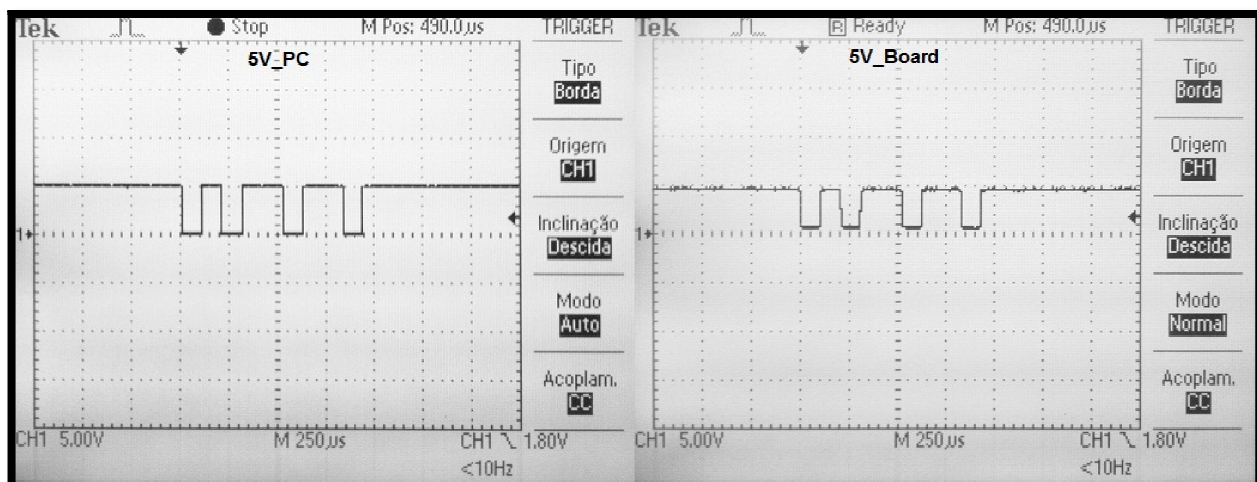


Figura 40 – Comparação no acoplador óptico do sinal transmitido pelo dispositivo RS485

d) TTL

Conforme o item 2.3, o TTL não necessita de isolamento para os sinais de transmissão e recepção de dados, portanto, a análise do isolamento é feita somente com o sinal de RTS.

Na figura 41 pode-se ver a comparação do sinal de RTS antes e depois do acoplador óptico, ou seja, referenciado à fonte 5V_Board e à fonte 5V_PC respectivamente.

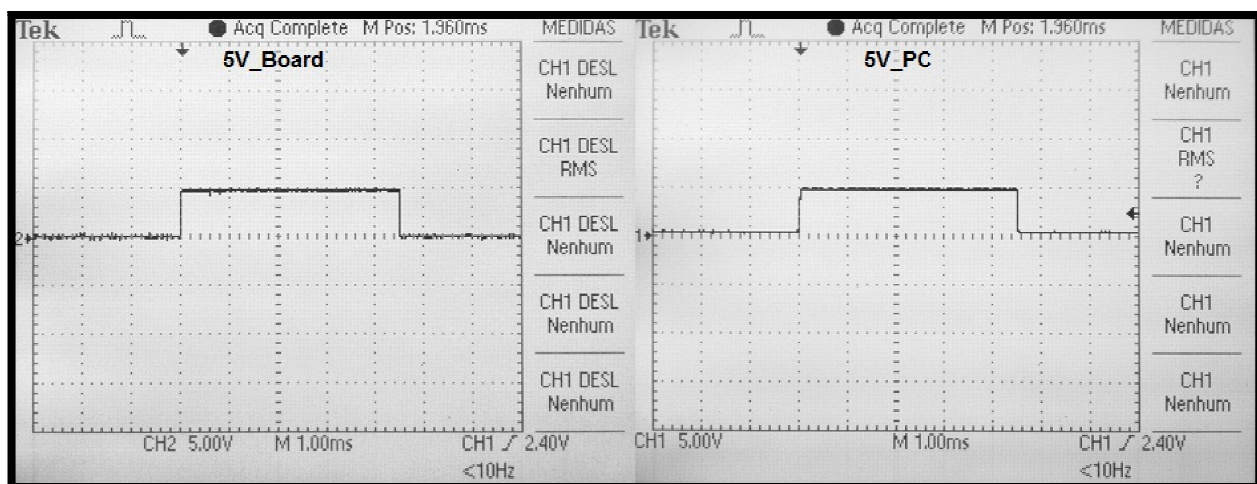


Figura 41 – Comparação no acoplador óptico do sinal de RTS do dispositivo TTL

Pode-se notar uma leve distorção entre os sinais comparados, fato que ocorre devido ao tempo de resposta do acoplador óptico não ser rápido o suficiente para evitar a distorção dos *bits*. Porém, pode-se ver que esta distorção não é suficientemente grande para

descaracterização dos *bits*, não afetando assim a comunicação entre os dispositivos de diferentes padrões encontrados no projeto.

3.5 Geração de RTS para o RS232

Para testar o funcionamento da geração do sinal de RTS do dispositivo RS232, foi comparado o sinal de transmissão de um dado pelo dispositivo (232_TX_PC) e o sinal de RTS gerado na saída do oscilador LM555.

Na figura 42 pode-se notar a correta geração do sinal de RTS, mudando para nível “1” junto com o *start bit* do sinal de transmissão e perdurando tempo suficientemente grande para que o *byte* seja transmitido.

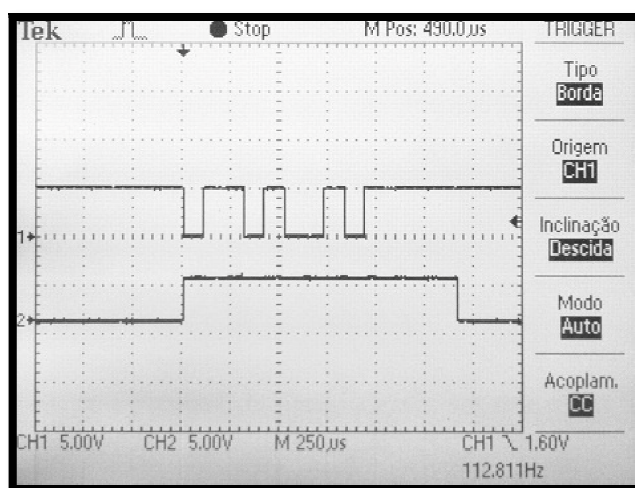


Figura 42 – Transmissão RS232 e sinal gerado de RTS

4. Teste em Aplicação Real

Para teste do funcionamento do adaptador de comunicação serial em uma aplicação real foi utilizada uma unidade controladora fabricada pela empresa Tecumseh do Brasil chamada T-Control ®, uma giga de teste utilizada para explicitar os comandos da controladora, um *software* próprio da controladora e dois *notebooks*.

O teste consiste em enviar comandos para a controladora através de um *notebook* com o *software* da controladora instalado e verificar a reprodução deste comando na giga de teste e via comunicação serial do segundo *notebook*. O adaptador de comunicação serial foi conectado na porta padrão RS232 de um *notebook* e, com o auxílio do *software* da controladora, comandos foram enviados ao adaptador de comunicação serial, convertidos para os níveis TTL e enviados até a controladora, que reproduzia estes comandos na giga de teste. Um segundo *notebook* foi conectado na porta USB do adaptador de comunicação serial para verificação dos comandos enviados.

Um diagrama de blocos do teste pode ser visto abaixo, na figura 43:

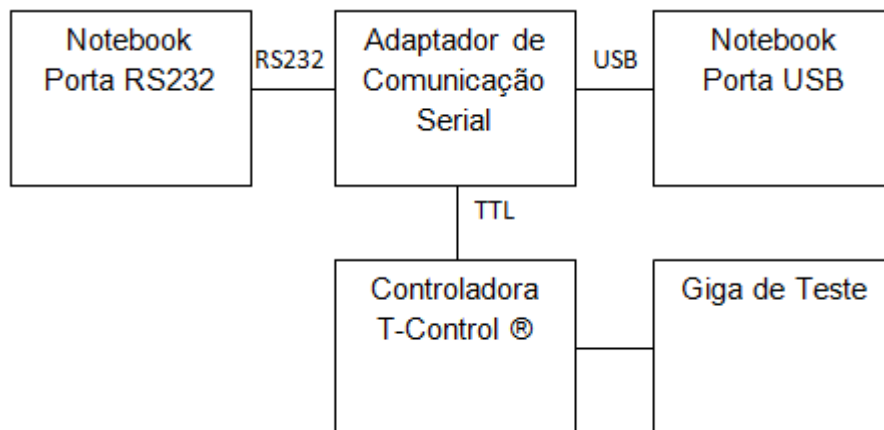


Figura 43 – Diagrama de blocos do sistema de Teste em Aplicação Real

Uma foto do teste pode ser vista abaixo, na figura 44:



Figura 44 – Montagem do sistema de Teste em Aplicação Real

Com este teste, que envolveu três dos quatro padrões de comunicação serial contemplados pelo adaptador de comunicação, pode-se constatar a funcionalidade do mesmo em uma aplicação prática, comandando uma controladora e verificando com um dispositivo externo se os comandos e respostas estão sendo corretamente interpretados.

5. Conclusões

A partir da construção do protótipo e da realização dos testes, concluímos que para aplicações que não necessitam de grandes taxas de transmissão, o Adaptador Isolado de Comunicação Serial comporta-se perfeitamente na conversão dos padrões abordados. Caso aplicações mais específicas necessitem de mais velocidade na transmissão e recepção de dados, componentes mais rápidos podem ser utilizados no projeto, tornando-o totalmente adaptável.

O Adaptador Isolado de Comunicação Serial pode ser utilizado tanto em aplicações domésticas com uma simples conversão USB – RS232 quanto em aplicações industriais, centros de pesquisas e desenvolvimento, entre outros. Baseado nos atuais conversores de padrões encontrados no mercado, o projeto permite grande economia pois, o custo total do adaptador sai por menos de R\$70,00, podendo assim ter valor de venda próximo a um simples adaptador USB-TTL, vendido por cerca de R\$90,00 em diversas lojas na *internet*; além disso substitui seis conversores duplos por um quádruplo, com mesma eficiência e com a vantagem do isolamento galvânico das comunicações envolvendo os níveis TTL, além do fato de permitir a interação entre três ou mais dispositivos de padrões diferentes.

Com isso pode-se dizer que trata-se de um produto extremamente viável para ser colocado no mercado, aliando uma vasta gama de aplicações com um preço relativamente baixo.

Uma futura aplicação que merece ser destacada é a possibilidade do uso do Adaptador Isolado de Comunicação Serial em aplicações didáticas. Uma nova placa poderia ser construída com componentes mais caros, porém próprios para comunicação de dados em alta velocidade, conseguindo assim uma menor deformação das formas de onda ao longo do circuito. Um *kit* didático poderia ser montado, com pontos disponíveis aos estudantes para aquisição das formas de onda dos diversos padrões de comunicação serial envolvidos no projeto, fazendo assim o aluno familiarizar-se com os mesmos, entender suas diferenças e aplicações específicas.

Por fim pode-se dizer que o projeto atendeu seus objetivos, com uma placa funcional, de baixo custo e com diversas possibilidades de aplicações.

6. Referências Bibliográficas

7408 Datasheet. Fairchild Semiconductors: Folha de dados. 1998. Disponível em:
<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-34/DSA-678818.html>>.
Acesso em: 22 de abr. 2011.

7432 Datasheet. Fairchild Semiconductors: Folha de dados. 1986. Disponível em:
<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-11/DSA-204062.html>>.
Acesso em: 22 de abr. 2011.

ARNE ROBOTICS. Conversor RS232/TTL duplo. 2005. Disponível em:
<http://www.arnerobotics.com.br/eletronica/conversor_RS232_ttl_duplo.htm>.
Acesso em : 24 de mar. 2011.

BLACK, U. *Physical Levels Interfaces and Protocols*. Washington D.C.: Computer Society Press, 1988. p. 100.

CAMPBELL, J. RS-232 : Técnicas de Interface. Rio de Janeiro: EBRAS, 1990. p. 7-28.

ELECTRONICS DIY. FT232RL Breakout Board. 2011. Disponível em: <http://electronics-diy.com/FT232RL_USB_to_Serial_Adapter_for_PIC_AVR_ATMEGA_ARDUINO_MCUs.php>. 2011. Acesso em : 22 de mar. 2011.

ELETRONICA.ORG. Padrão Serial RS232. 2008. Disponível em:
<<http://www2.eletronica.org/artigos/eletronica-digital/padrao-serial-RS232>>. Acesso em: 13 de mar. 2011.

FT232R Datasheet. FTDI: Folha de dados. 2005. Disponível em:
<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-29/DSA-563628.html>>.
Acesso em : 12 de abr. 2011.

LM555 Datasheet. Fairchild Semiconductors: Folha de dados. 2002. Disponível em:
<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-5/DSA-83938.html>>. Acesso em: 19 de mar. 2011.

LR. Conectores DB9 e DB25 para comunicação serial RS232. 2011. Disponível em:
<<http://www.lri.com.br/images/downloads/RS232.pdf>>. Acesso em : 13 de mar. 2011.

MAX232 Datasheet. MAXIM: Folha de dados. 2000. Disponível em:

<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-18/DSA-345942.html>>.

Acesso em: 20 de abr. 2011.

METALTEX. Conectores mini-DIN. 2011. Disponível em:

<<http://www.metaltex.com.br/downloads/MDF.pdf>>. Acesso em: 24 de mar. 2011.

NATIONAL SEMICONDUCTOR. Ten Ways to Bulletproof RS485 Interfaces. 1996.

Disponível em: <<http://www.national.com/an/AN/AN-1057.pdf>>. Acesso em: 12 de fev. 2011.

NINGBO HI-SUN. Conectores USB. 2011. Disponível em:

<<http://www.nbhisun.net/products/usb.html>> . Acesso em: 22 de mar. 2011.

ST485 Datasheet. STMicroelectronics: Folha de dados. 2002. Disponível em:

<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-31/DSA-606266.html>>.

Acesso em : 20 de abr. 2011.

TAUB, H. *Circuitos digitais e microprocessadores*. São Paulo: McGraw-Hill do Brasil, 1984. p. 78.

TL431 Datasheet. Fairchild Semiconductors: Folha de dados. 2003. Disponível em:

<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-11/DSA-209957.html>>.

Acesso em: 12 de abr. 2011.

TLP181 Datasheet. Toshiba: Folha de dados. 2002. Disponível em:

<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-4/DSA-79837.html>>.

Acesso em: 4 de abr. 2011.

TORRES, G. *Hardware: Curso Completo*. Rio de Janeiro: Axcel, 2001. p. 347.

VASCONCELOS, L. Fontes de alimentação, reguladores, potência. 2005. Disponível em:

<http://blogs.forumpcs.com.br/laercio_vasconcelos/2005/05/16/fontes-de-alimentacao-reguladores-potencia>. Acesso em: 3 de abr. 2011.

VIPER22 DATASHEET. STMicroelectronics: Folha de dados. 2002. Disponível em:

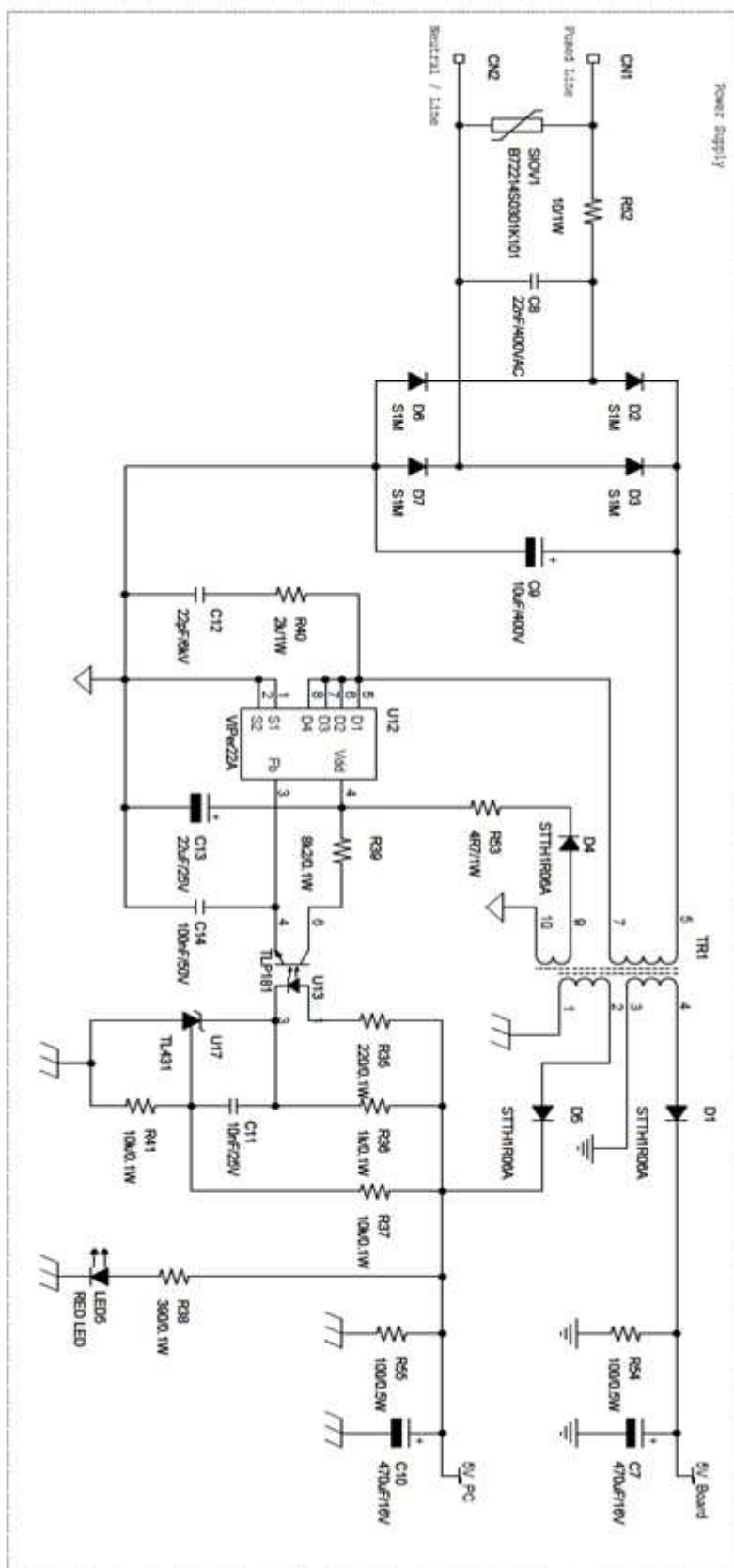
<<http://www.datasheetarchive.com/pdf-datasheets/Datasheets-31/DSA-607110.html>>.

Acesso em: 4 de abr. 2011.

ZANCO, W.S. *Microcontroladores PIC: Técnicas de Hardware e Software* para projetos de circuitos eletrônicos. São Paulo: Érica, 2008. p. 253.

ZUFFO, J. A. *Microprocessadores: dutos de sistema, técnicas de interface e sistemas de comunicação de dados*. São Paulo: Edgard Blücher, 1981. p. 344.

Esquemático da fonte de alimentação.



Apêndice B

Esquemático dos transceptores.

