UART

Versão 2015

RESUMO

O objetivo desta experiência é projetar e implementar circuitos digitais para comunicação de dados serial (transmissão e recepção de dados) com um terminal de dados, utilizando a norma EIA-RS-232C e o código ASCII (*American Standard Code for Information Interchange*).

A parte prática consiste no projeto e na implementação de um circuito digital que troca (envia e recebe) dados (caracteres em código ASCII) de um terminal serial usando uma placa de desenvolvimento FPGA.

OBJETIVOS

Após a conclusão desta experiência, os seguintes tópicos devem ser conhecidos pelos alunos:

- *UART;*
- *Comunicação serial assíncrona (RS-232C);*
- *Conversores de nível de tensão;*
- *Projeto com FPGA.*

1. INTRODUÇÃO TEÓRICA

1.1. Comunicação Serial Assíncrona

Uma comunicação é chamada de **serial** quando o envio dos códigos dos caracteres se processa sobre uma única linha, onde os bits enviados são encadeados um por vez, e numa fila. Essa comunicação é **assíncrona** quando não exige o sincronismo dos relógios entre o receptor e o transmissor. Cada caractere carrega seus próprios sinais de sincronismo.

1.2. Normas EIA-RS-232[1]

As normas EIA (*Electronic Industries Alliance*) são adotadas pelos fabricantes de equipamentos para eliminar possíveis discrepâncias na interligação dos mesmos, e auxiliar o usuário na escolha e instalação dos equipamentos. Em particular, a norma EIA-RS-232C[2] tem por objetivo padronizar um método de interconexão entre terminais e canais de comunicação de dados, quando os mesmos são fornecidos por fabricantes distintos. Ela define um modo de troca de sinais de controle (protocolo) e de sinais de dados serializados entre o terminal e o canal de comunicação de dados.

O canal de transmissão de dados inclui os conversores de sinais e a linha de transmissão; os conversores são utilizados para compatibilizar os níveis dos sinais recebidos e transmitidos para a linha de transmissão com a norma em questão; as normas elétricas destes sinais são descritas mais adiante.

O terminal de dados, entretanto, deve providenciar a serialização dos dados, para que o projeto do canal de transmissão seja independente do comprimento e da codificação dos mesmos. A Figura 1.1 mostra os sinais típicos em uma interligação entre dois terminais por meio de um canal de transmissão de dados, utilizando os critérios apresentados por esta norma.

[1] O padrão EIA RS-232 também é conhecido como EIA 232 ou, mais recentemente, TIA 232. TIA é uma abreviatura de "*Telecommunications Industry Association*".

[2] A revisão C do padrão EIA RS-232 é conhecido como RS-232C e foi elaborada em 1969. Atualmente, a última revisão é o ANSI/EIA/TIA 232 F de 1997.

a) Sinais de Intercomunicação

Neste item são descritas as funções dos principais sinais de intercomunicação padronizados pela norma EIA-RS-232C.

- FRAME GROUND: corresponde ao fio terra dos equipamentos. É ligado à carcaça metálica do equipamento e nas partes condutoras do mesmo, expostas ao operador, de modo a evitar diferenças de potencial entre o equipamento e o operador;
- SIGNAL GROUND: estabelece a referência elétrica (terra) para todos os sinais de intercomunicação, exceto para o FRAME GROUND;
- TRANSMITTED DATA: corresponde ao dado serializado, gerado pelo terminal de dados;
- RECEIVED DATA: corresponde ao dado serializado recebido do canal de comunicação de dados;
- REQUEST TO SEND: gerado pelo terminal de dados, informa aos conversores de sinais a ele conectados, que o terminal deseja transmitir dados.
- CLEAR TO SEND: gerado nos circuitos dos conversores de sinais, é utilizado para indicar que os circuitos estão prontos para enviar dados.

Todos estes sinais e os possíveis opcionais devem obedecer às características elétricas relacionadas no item 1.2.

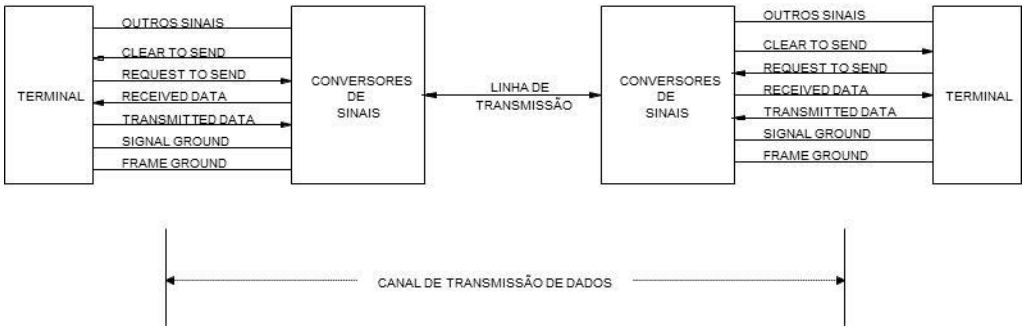

Figura 1.1 – Interligação entre dois Terminais de Dados.

b) Características Elétricas dos Sinais

O ponto de interconexão entre o terminal de dados e o canal de transmissão deve ser feito por meio de conectores, onde o conector fêmea deve estar associado ao canal de transmissão de dados e fixado próximo do terminal de dados. O conector macho, associado ao terminal de dados, deve ser acoplado a um cabo, que deve ser o menor possível (menor que 15 metros).

A norma ainda específica que a amplitude de nenhum sinal deve exceder 50V em relação aos sinais SIGNAL GROUND ou FRAME GROUND e que a máxima corrente de curto-circuito entre dois sinais quaisquer, inclusive os terras, não deve exceder 1,5 A.

Os circuitos que geram os sinais de interligação devem ser projetados de forma a não apresentarem nenhum dano no caso desses sinais sofrerem algum curto-circuito com os terras presentes no equipamento ou permanecerem em aberto. Também deve ser possível seu funcionamento contínuo nas condições máximas especificadas anteriormente.

Os sinais nos circuitos de interligação podem estar nas condições ON ou MARK, quando a tensão nesses sinais for mais negativa que -3 V em relação à terra de sinal, e OFF ou SPACE, quando a tensão nesses sinais for maior que +3V em relação à terra de sinal. Geralmente, deve-se usar a nomenclatura MARK/SPACE para os sinais TRANSMITTED DATA e RECEIVED DATA e ON/OFF para os demais.

Os circuitos de recepção de sinais devem ser projetados de modo a serem acionados apenas por tensão ficando, portanto, insensíveis a parâmetros como Tempos de Subida, Tempos de Descida, existência de *overshoots* ou *undershoots* e etc.

A impedância de entrada nos circuitos de recepção deve ter uma resistência, em regime DC, maior que 1000 Ω, tensão em circuito aberto menor que 2 V e capacitância menor que 2500 pF, medidas nos pinos do conector de acoplamento.

A seguir, apresentam-se as características elétricas que os sinais de interligação devem apresentar:

a) a tensão dos sinais deve ser pelo menos ± 3V em relação à terra de sinais e não exceder ± 25V, também com respeito à terra de sinais;

b) a forma de onda dos sinais deve ser aproximadamente retangular;

c) para se evitar indução de ruídos nos circuitos de interligação, não devem ser utilizados *drives* indutivos.

Esta técnica de transmissão de sinal digital por nível de tensão é denominada de "Loop de tensão".

A transmissão digital também pode ser efetuada por níveis de corrente. Esta técnica é denominada de "*loop* de corrente". Os valores padrões são:

- MARK ou ON: 60 mA (ou 20mA);
- SPACE ou OFF: 0 mA.

A comunicação por *loop* de corrente é mais imune a ruído que a comunicação por níveis de tensão, e suas interfaces transmissoras e receptoras são da mesma complexidade.

c) Interface Mecânica

Como qualquer outro padrão, o RS-232C é uma referência para projetistas de equipamentos. Assim define entre outras coisas as interfaces mecânicas ou conectores. A Figura 1.2 ilustra alguns conectores padrão existentes.

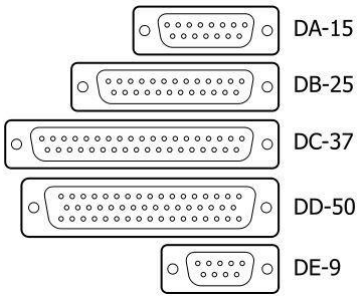

Figura 1.2 – Alguns conectores – interface mecânica.

A Tabela I apresenta a designação dos sinais do RS-232C aos pinos dos conectores DB-25 e DE-9.

Tabela I – Designação de alguns sinais do RS-232C aos pinos de conectores.

Sinal	DB-25	DE-9
Common Ground (CG)	7	5
Transmitted Data (TxD)	2	3
Received Data (RxD)	3	2
Data Terminal Ready (DTR)	20	4
Data Set Ready (DSR)	6	6
Request To Send (RTS)	4	7
Clear To Send (CTS)	5	8
Carrier Detect (DCD)	8	1
Ring Indicator (RI)	22	9

d) UART

Um UART (*Universal Asynchronous Receiver/Transmitter* ou transmissor/receptor assíncrono universal) é um dispositivo responsável pela comunicação de dados (paralelos) em um meio de transmissão serial. Sua principal função é converter dados entre as formas paralela e serial. Os UARTs normalmente são usados com os padrões de comunicação serial, como RS-232C, RS-422 e RS-485. Nos computadores pessoais, as placas-mãe incorporam um circuito integrado UART para comunicação pelas portas seriais. Um exemplo de UART é o National Semiconductor PC16650D, que alcança taxas de até 1,5M bauds.

De uma maneira geral, uma implementação de UART pode ser dividida em quatro blocos, a saber: transmissão, recepção, geração de *baud-rate* e lógica de interface (figura 1.3).

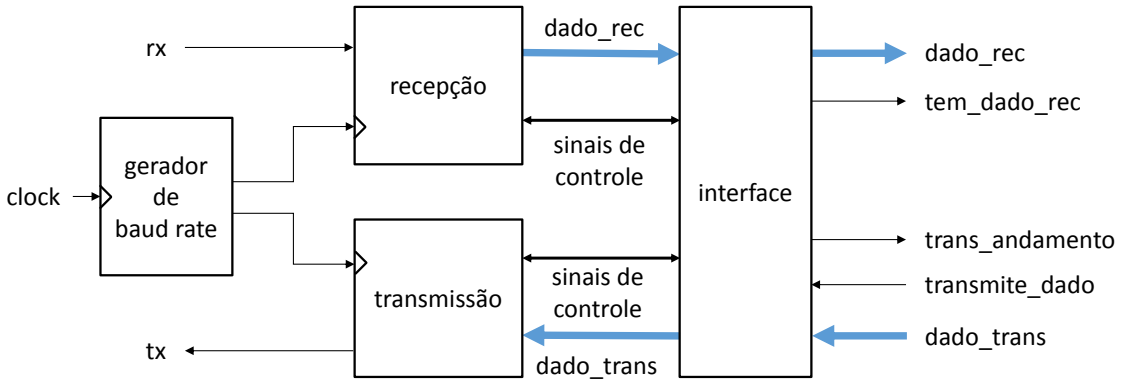

Figura 1.3 – Diagrama de blocos de uma implementação de UART.

Os blocos de **transmissão** e **recepção** são responsáveis pela troca de dados entre os equipamentos, com a comunicação de dados seriais. O bloco **gerador de *baud rate*** é responsável pela geração de um sinal de controle de temporização para os blocos de transmissão e recepção. Por exemplo, para o projeto do circuito de transmissão serial assíncrona, gera um sinal de 1200 Hz para uma transmissão configurada para 1200 *bauds*. O bloco de **interface** faz a comunicação com outro sistema digital através dos sinais de dados paralelos *dado_trans* e *dado_rec*. Os sinais de controle da interface são:

- *tem_dado_rec*: indica a presença de um dado recebido pelo circuito que deve ser registrado;
- *trans_andamento*: indica que uma transmissão está em andamento; assim, uma nova transmissão não pode ser realizada até que a atual termine;
- *transmite_dado*: submete uma nova transmissão para o UART.

Normalmente, o bloco de interface deve conter registradores para armazenar os dados enquanto as operações de transmissão e recepção não são concluídas. O número de registradores varia de acordo com a capacidade de dados do projeto.

Os sinais *tx* e *rx* devem ser conectados com conversores de níveis de tensão para serem convertidos para tensões compatíveis com a comunicação serial (veja seção 1.6).

1.3. Transmissão Serial Assíncrona

A comunicação com o terminal é realizada com protocolo assíncrono como mostrado na Figura 1.4, com notação MARK (1) e SPACE (0). Em estado de repouso, o canal de comunicação apresenta o sinal MARK. O início da transmissão é sinalizado pelo START BIT, através de um sinal SPACE. Em seguida, são transmitidos os bits de dados, o bit de paridade e os STOP BITs.

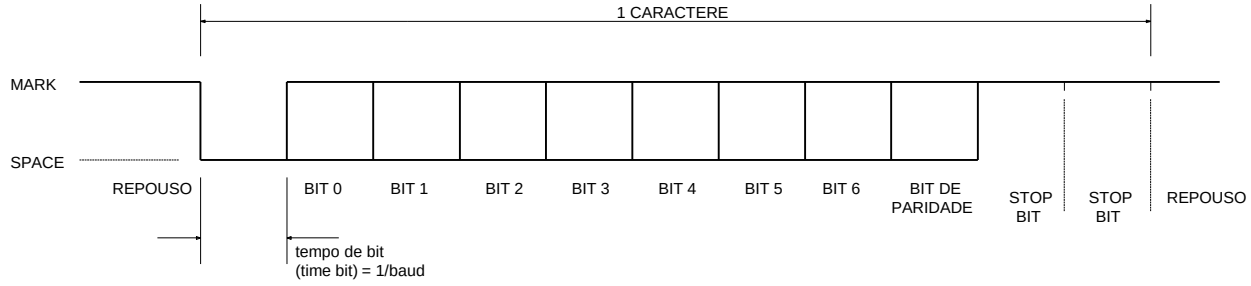

Figura 1.4 – Protocolo Serial Assíncrono (MARK e SPACE).

A velocidade de comunicação é definida em *bauds*, que, em um sistema de transmissão de dados binários, representam o número de *bits* transmitidos por segundo.

Os níveis de tensão utilizados tipicamente para os níveis MARK e SPACE são -12V e +12V, respectivamente. A Figura 1.5 mostra como o caractere '5', cujo código em ASCII é dado pela palavra $(0110101)_2$, é enviado para um terminal.

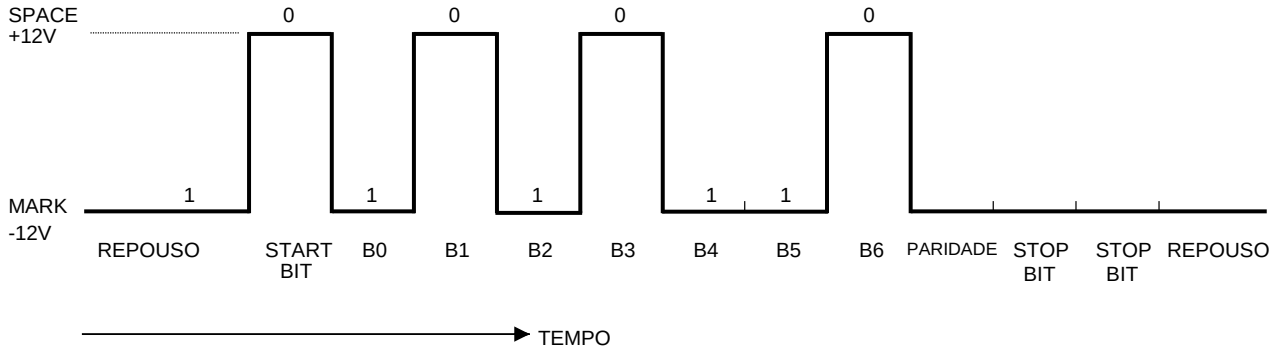

Figura 1.5 – Formato da saída serial da palavra $(0110101)_2$.

O circuito de transmissão deve incluir um registrador de deslocamento com entrada paralela e saída serial. Um aspecto importante no projeto de um transmissor serial assíncrono diz respeito ao sincronismo do *clock* interno do circuito e a transmissão dos bits pelo canal serial: cada bit transmitido deve seguir um intervalo de tempo específico, ou seja, um bit deve ser transmitido a cada 1/baud segundos. Ao término da transmissão, o canal serial deve retornar para o repouso, ou seja, para o sinal MARK.

1.4. Recepção Serial Assíncrona

Os sinais enviados pelo terminal têm a mesma forma dos recebidos pelo terminal. Portanto, o circuito de recepção de dados deve detectar o pulso do START BIT e, a partir desse ponto, por meio de um registrador de deslocamento, copiar o dado em código ASCII e o bit de paridade.

Como o início da operação do terminal certamente é assíncrono com relação ao relógio do circuito de recepção, deve-se tentar sincronizar, o melhor possível, a entrada de dados detectada pelo pulso do START BIT. O circuito de recepção deve tentar amostrar o dado de entrada no meio do intervalo do bit, ou seja, $\frac{1}{2}\times\frac{1}{\text{baud}}$ após o início da transmissão do bit.

Uma sugestão para realizar esse sincronismo é utilizar um sinal de frequência algumas vezes mais alta que aquela necessária no registrador de deslocamento, para gerar o *clock* desse registrador. Este método é conhecido como sobre-amostragem ou *oversampling*.

O diagrama de blocos do circuito sugerido é apresentado na Figura 1.6; os diagramas de sinais dessa figura são para um caso em que a frequência do oscilador é 8 vezes maior que a frequência do *clock* do registrador de deslocamento.

Em t = 0, o diferenciador liga o *flip-flop* que habilita o contador a começar a contar (admitindo-se que esse contador seja sensível à borda de subida do sinal OSC). Temos duas possibilidades de temporização entre o dado de entrada e o sinal do oscilador do circuito de recepção:

- No **caso A**, vemos que, logo após o contador ser habilitado, ocorre uma borda de subida do sinal "OSC A". Quando o contador atinge o valor 4 (metade do número de vezes que o clock é superior à frequência de comunicação), o sinal CLOCK A muda de estado, indo para o nível lógico 1 e amostrando o sinal DADOS (supondo-se também que o registrador de deslocamento é acionado pela borda de subida do sinal OSC).
- No **caso B**, por outro lado, o sinal OSC B passa por uma borda de subida e, em seguida, temos o instante t = 0 e o contador é habilitado. Portanto, o contador só começará a contar na próxima borda de subida de OSC B. Quando ele atingir o valor 4, o sinal CLOCK B passa para nível lógico 1 e o dado é amostrado no registrador de deslocamento.

Vemos, portanto, que para o caso em que $F_{OSC} = 8\times f_{CLOCK}$ o dado será amostrado o mais rápido em $\frac{1}{8}\times T_{CLOCK}$ antes do meio do intervalo do dado, e o mais lento bem no meio do dado. É fácil concluir que o intervalo de amostragem ΔA (veja ainda Figura 1.6) ficará sempre entre $\frac{1}{2}\times T_{CLOCK}$-$T_{OSC}$ e T_{CLOCK}; quanto menor T_{OSC}, maiores são as chances de o dado ser amostrado mais perto de $\frac{1}{2}\times T_{CLOCK}$ (meio do bit); e isso é bom porque é em $T_{CLOCK}/2$ que o dado tem as maiores probabilidades de estar estável.

Veja também que o terminal só garante que o dado esteja estável de $\frac{1}{4}\times T_{CLOCK}$ a $\frac{3}{4}\times T_{CLOCK}$; portanto $\Delta A<\frac{1}{4}\times T_{CLOCK}$ (ou seja, $T_{OSC} < \frac{1}{4}\times T_{CLOCK}$).

O fim de transmissão de um caractere ASCII também deve ser decidido pelo circuito do receptor de dados. Essa detecção é feita utilizando-se um contador que deve contar o número de pulsos do sinal de *clock* do registrador de deslocamento correspondente ao caractere e gerar um sinal quando a recepção do mesmo chegar ao fim.

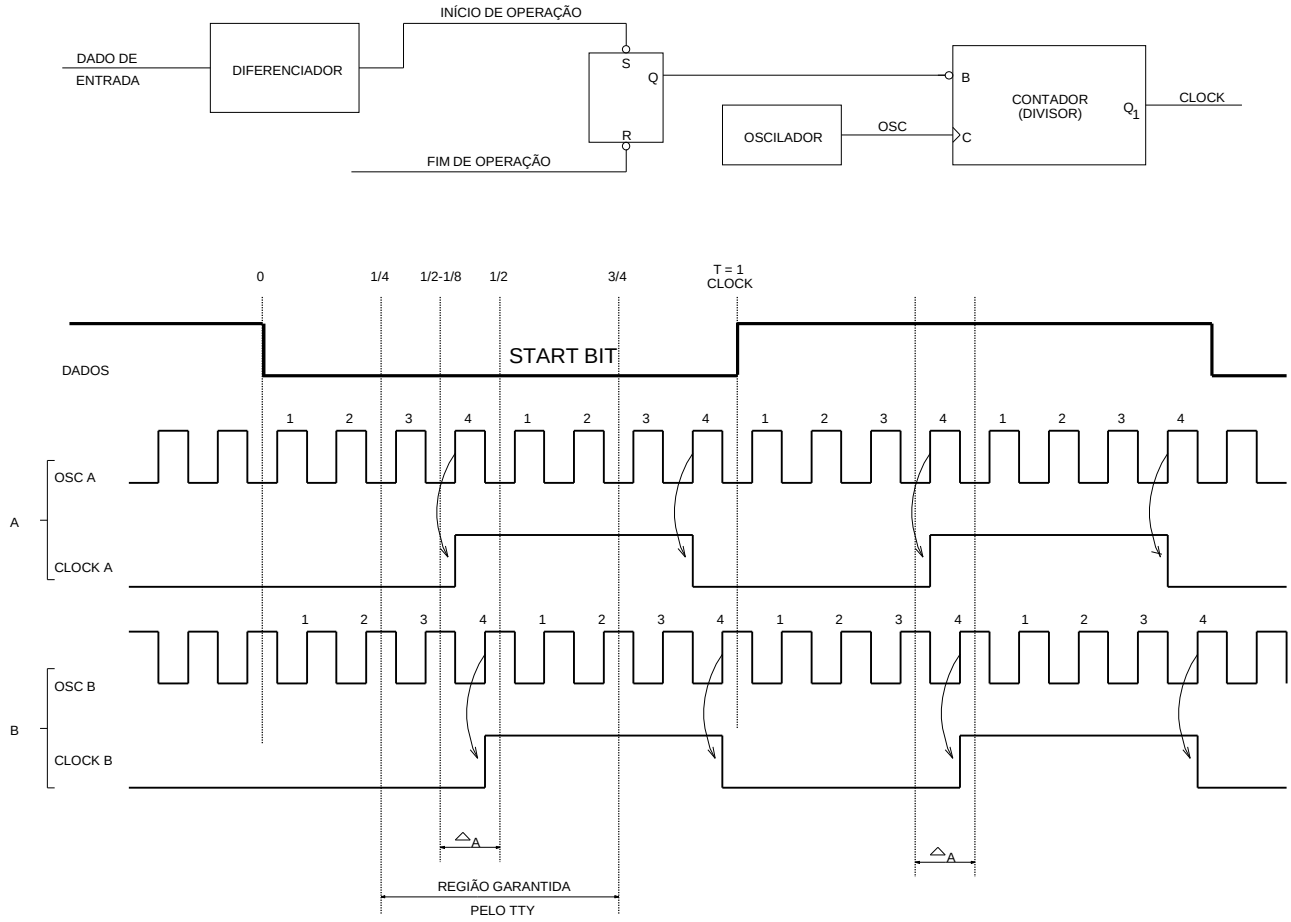

Figura 1.6 – Circuito de Recepção: Diagrama de blocos e Diagrama de sinais.

No caso em que se usa um *clock* com uma frequência muito maior que a taxa de transmissão do terminal, é preciso o projeto de uma unidade de controle que possa manter esta contagem de períodos do *clock* para fazer a amostragem do sinal de dados. Não basta um contador binário simples para realizar a divisão de frequência ou contagem dos pulsos. Por exemplo, na placa **Altera DE2**, se for adotado o *clock* interno de 27 MHz, seria necessário um contador que realizasse uma divisão por 22500 para obter uma frequência de 1200 Hz. A unidade de controle deve incorporar em seu diagrama de estados uma contagem interna para garantir a correta amostragem do sinal serial o mais próximo possível do meio do pulso.

Uma forma alternativa é adotar uma frequência de 27MHz ou 50 MHz para o relógio da Unidade de Controle e contar com o auxílio do módulo Gerador de *Baud Rate*. Este módulo gera sinais de amostragem dos dados de entrada (*enable ticks*) na temporização correta. Ou seja, a Unidade de Controle aguarda a chegada destes sinais de controle e, quando for o momento correto da amostragem, o circuito de recepção recebe um pulso com duração de um período de *clock* para que o bit de dado seja armazenado.

O projeto do circuito de recepção serial assíncrona pode ser desenvolvido com base em um diagrama ASM (figura 1.7), conforme apresentado em (Chu, 2008).

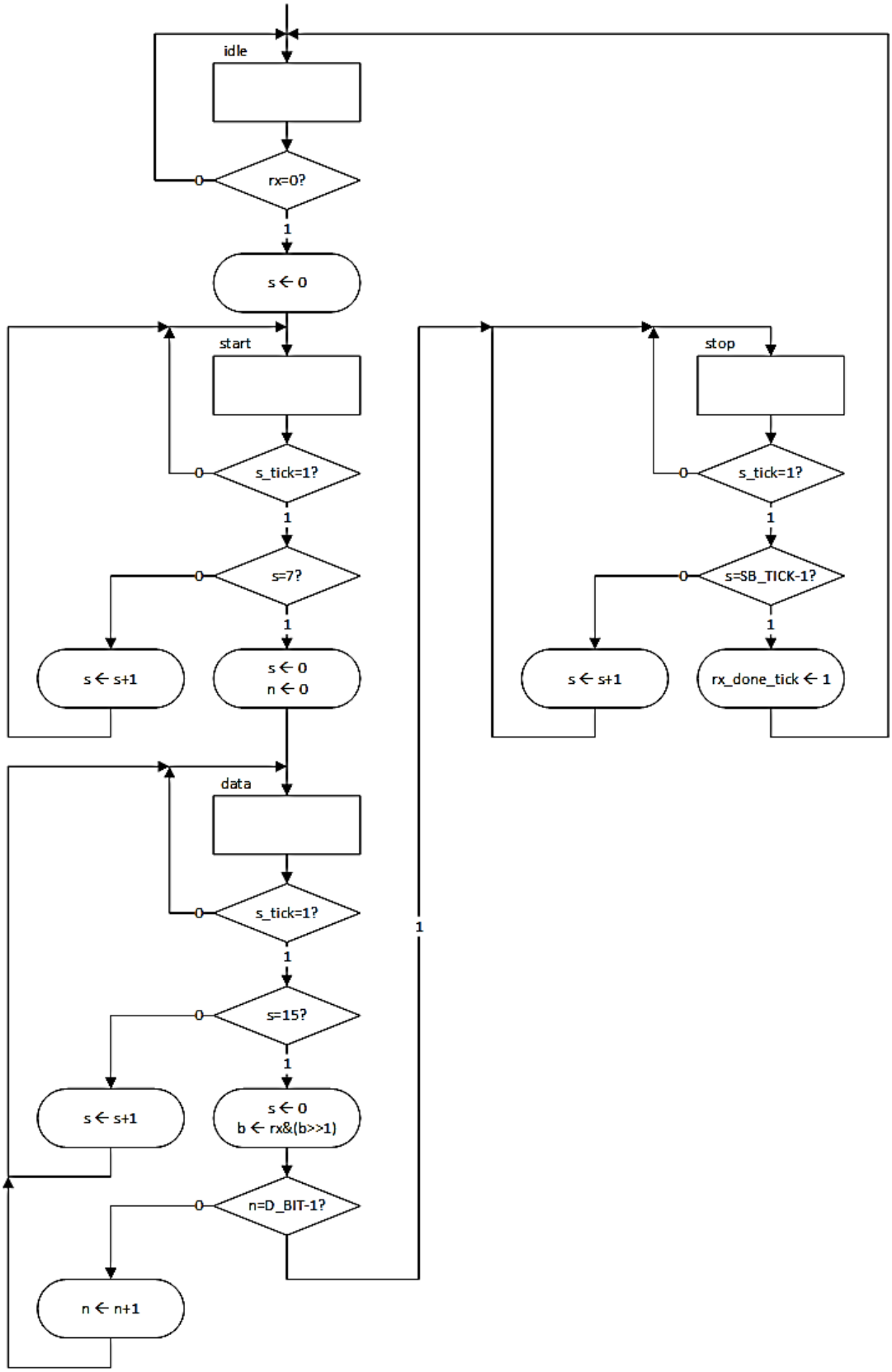

Figura 1.7 – Diagrama ASM do circuito de recepção serial da UART.
(fonte: adaptado de Chu, 2008).

1.5. Código ASCII

O código ASCII é um código padronizado, utilizado para a representação dos caracteres alfanuméricos na área de computação digital. Possui 7 bits de informação e, eventualmente, um bit de paridade; com esses 7 bits são codificadas as letras maiúsculas e minúsculas, números decimais, sinais de pontuação e caracteres de controle. A Tabela II apresenta o código ASCII completo.

Tabela II – Tabela com Código ASCII.

Bits				b_6	0	0	0	0	1	1	1	1
				b_5	0	0	1	1	0	0	1	1
				b_4	0	1	0	1	0	1	0	1
b_3	b_2	b_1	b_0									
0	0	0	0		NULL	DLE	SP	0	@	P	’	p
0	0	0	1		SOH	DC1	!	1	A	Q	a	q
0	0	1	0		STX	DC2	“	2	B	R	b	r
0	0	1	1		ETX	DC3	#	3	C	S	c	s
0	1	0	0		EDT	DC4	$	4	D	T	d	t
0	1	0	1		ENQ	NAK	%	5	E	U	e	u
0	1	1	0		ACK	SYN	&	6	F	V	f	v
0	1	1	1		BEL	ETE	`	7	G	W	g	w
1	0	0	0		BS	CAN	(	8	H	X	h	x
1	0	0	1		HT	EM	)	9	I	Y	i	y
1	0	1	0		LF	SUB	*	:	J	Z	j	z
1	0	1	1		VT	ESC	+	;	K	[	k	{
1	1	0	0		FF	FS	,	<	L	\	l	:
1	1	0	1		CR	GS	-	=	M	]	m	}
1	1	1	0		S0	RS	.	>	N	^	n	~
1	1	1	1		S1	US	/	?	O	_	o	DEL

Palavra	b_6	b_5	b_4	b_3	b_2	b_1	b_0

O código ASCII foi desenvolvido para representar textos para computadores ou equipamentos de comunicação e seu primeiro padrão foi publicado em 1963. Mais recentemente, com a difusão do uso de computadores, novos padrões foram desenvolvidos para incorporar novos caracteres e símbolos. Uma evolução do código ASCII é o código UTF-8 (*UCS Transformation Format* – 8-bit).

1.6. Conversão de Níveis de Tensão

Como os padrões de nível de tensão para circuitos digitais (TTL ou CMOS) e para o RS-232C são diferentes, é necessário o uso de circuitos especializados para conversão de níveis de tensão. Por exemplo, o bit 1, que em um circuito digital tem um nível de tensão típico da ordem de +5V, deve ser convertido para um sinal MARK que tem tipicamente um nível de tensão de -12V. Da mesma forma, o bit 0 (tensão de 0V) deve ser convertido para o sinal SPACE (tensão +12V).

Vários componentes estão disponíveis no mercado para realizar a conversão de níveis de tensão. Por exemplo, temos o par 1488/1489 e o MAX232.

O componente 1488 é responsável pela conversão de níveis de tensão TTL para RS-232 e o 1489, de RS-232 para TTL. A figura 1.8 apresenta as pinagens destes componentes.

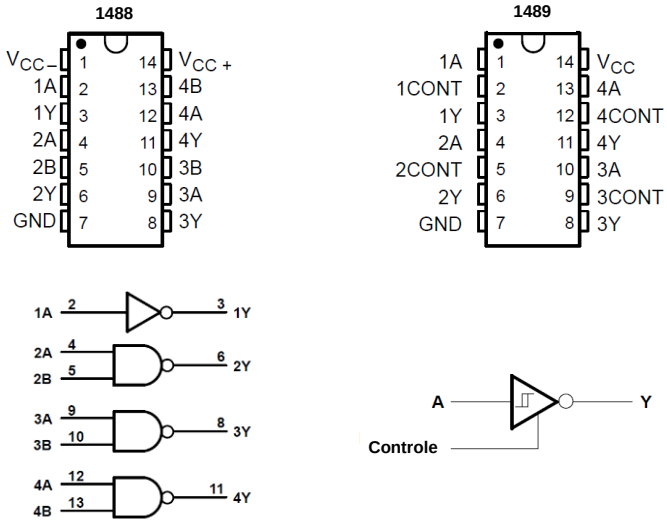

Figura 1.8 – Pinagens e esquemas lógicos dos conversores de tensão 1488 e 1489.

Convém observar que o conversor 1488 tem como pinos de alimentação: VCC+, VCC- e GND (tipicamente, +12V, -12V e 0V, respectivamente). Já o conversor 1489 tem os pinos comuns de alimentação: VCC (tipicamente +5V) e GND (0V).

O circuito integrado MAX232 contém circuitos de conversão de tensão para ambos os sentidos (TTL para RS232C e RS232C para TTL) em um único encapsulamento. A figura 1.9 mostra a pinagem deste CI.

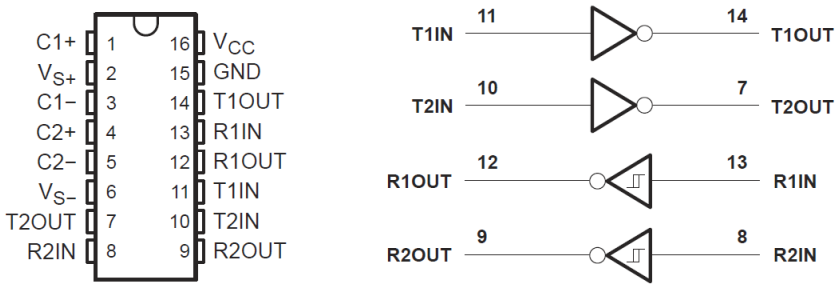

Figura 1.9 – Pinagem e esquema lógico do MAX232.

2. PARTE EXPERIMENTAL

A parte experimental envolve o desenvolvimento de um circuito digital para a troca de dados de um terminal serial, usando a placa de desenvolvimento FPGA da Altera. O circuito a ser projetado tem semelhança em seu funcionamento com os circuitos integrados UART usados em computadores pessoais e outros dispositivos.

2.1. Atividades Pré-Laboratório

a) As atividades pré-laboratório desta experiência consistem em projetar, implementar e documentar um circuito digital que realize a transmissão e recepção serial assíncrona de dados para um terminal serial, que estará disponível no laboratório. A figura 2.1 apresenta a interface de comunicação da placa FPGA e o terminal serial. Use o diagrama da figura 1.3 para o projeto.

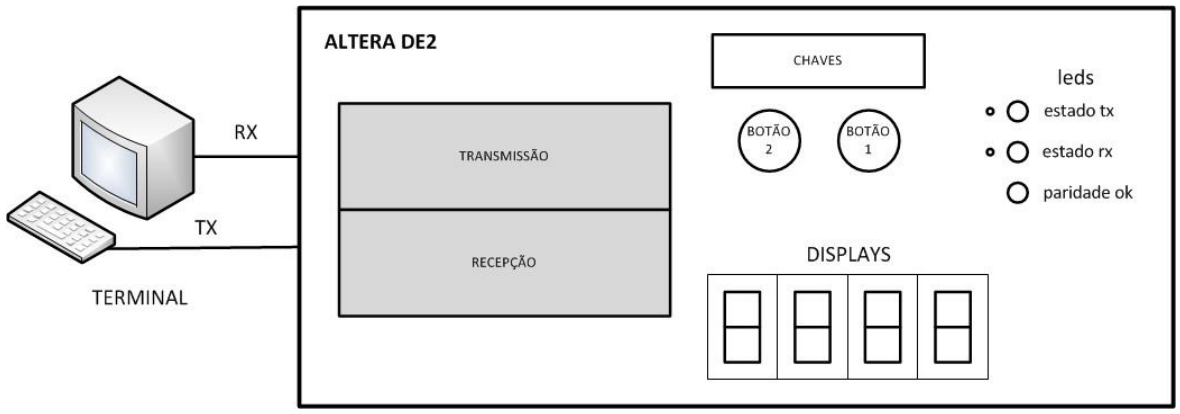

Figura 2.1 – Diagrama de Blocos do Circuito do UART.

O circuito executa duas operações principais: a transmissão de dados (caracteres ASCII) especificados em 7 chaves e a recepção de dados digitados no terminal a serem apresentados em displays de 7 segmentos. A **transmissão** se inicia, após a definição dos dados nas chaves, ao pressionar-se um botão de **partida** (botão 1). A **recepção** recebe os caracteres ASCII que vem da linha serial e apresenta em displays de 7 segmentos. Como pode ser visto na figura 2.1, os dois últimos caracteres digitados devem ser armazenados internamente para serem mostrados. O estado da transmissão e recepção de dados devem ser sinalizados através de *leds* e saídas digitais (GPIO).

Um sinal de **reset** (botão 2) está disponível para reiniciar o circuito e zerar os registradores internos que armazenam dados de transmissão e de recepção.

b) A documentação do projeto deve apresentar uma descrição detalhada do funcionamento do circuito e também os componentes escolhidos (decisões de projeto). Planeje os casos de testes devem ser executados para assegurar o correto funcionamento do circuito. Acrescente cartas de tempo dos circuitos de transmissão e de recepção serial no planejamento.

c) **Transmissão Serial**. O terminal serial que será utilizado no laboratório deve ser operado por tensão, isto é, com pulsos de tensão entre +12V e -12V. É necessário usar um conversor de nível para a interface com a linha serial (circuito integrado MAX232 da placa Altera DE2). A Figura 2.2 mostra o diagrama de blocos do circuito de transmissão a ser projetado.

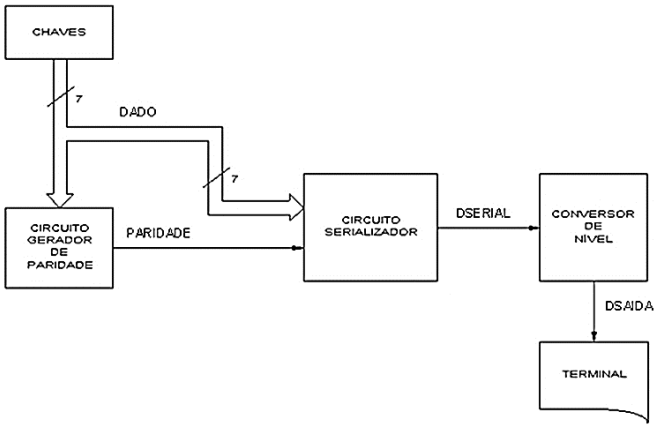

Figura 2.2 – Diagrama de Blocos do Circuito de Transmissão de Dados.

d) **Recepção Serial**. O segundo módulo a ser projetado é um circuito de recepção de dados enviados a partir do terminal serial, completando o estudo do funcionamento dos UARTs. O circuito a ser projetado deve detectar o acionamento de uma tecla no teclado do terminal. Os dois últimos caracteres recebidos devem ter seus respectivos códigos ASCII apresentados em "*displays*" hexadecimais (D3 e D2 indicam o último caractere, e D1 e D0, o último). Além disso, deve testar o bit de paridade e indicar, por meio de um LED (L0), que ocorreu um erro de transmissão. A comunicação deve ser projetada e configurada com velocidade de transmissão de 110 *bauds*, 7 bits de dados, paridade ímpar e 2 *stop bits*. Use uma taxa de sobre-amostragem de 8 ou 16 vezes. A figura 2.3 mostra um diagrama de blocos do circuito de recepção.

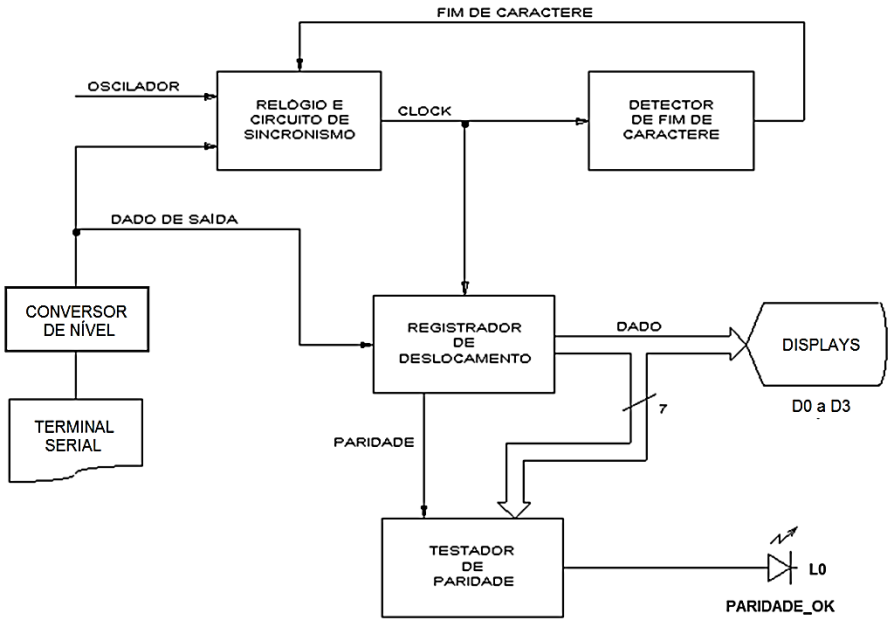

Figura 2.3 – Diagrama de Blocos do Circuito de Recepção de Dados.

e) **DICA**: Estude a documentação da placa de desenvolvimento FPGA Altera DE2 para localizar os componentes de suporte a comunicação serial (circuito *transceiver* MAX232 e conector DSUB para comunicação RS-232C). A figura 2.4 ilustra o diagrama esquemático dos componentes relativos à interface serial da placa DE2 da Altera.

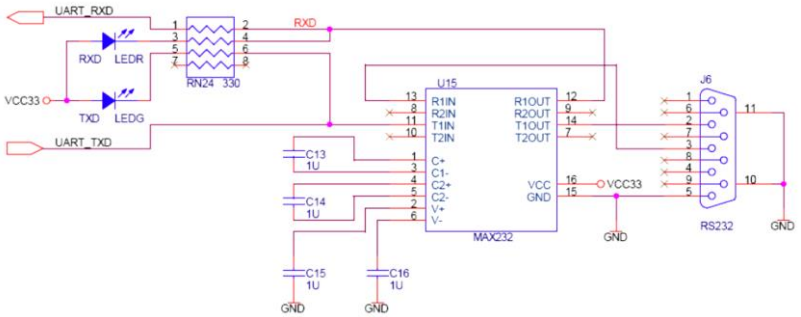

Figura 2.4 – Suporte a comunicação serial da placa DE2 da Altera (fonte: Altera 2008).

f) Anote a identificação dos pinos relacionados à comunicação serial na placa Altera DE2.

Nome do sinal	Pino	Descrição
UART_RXD		*UART Receiver*
UART_TXD		*UART Transmitter*

DICA: o sinal DSERIAL do projeto deve ser designado para o pino do sinal UART_TXD.

2.2. Implementação do Projeto

Neste item vamos implementar o projeto do circuito de transmissão serial na placa DE2.

g) Sintetizar o circuito do UART para a placa Altera DE2. Usar a seguinte designação de pinos:

- DADOS TX: chaves SW0 a SW6
- PARTIDA: botão KEY2
- RESET: botão KEY1
- DISPLAYS: displays HEX0 a HEX3
- PARIDADE_OK: led LEDR[0]
- LEDs STATUS: leds LEDG[0] e LEDG[1]
- GPIO: pinos GPIO_0[0] e GPIO_0[1]

DICA: lembrem-se que os botões na placa DE2 são ativos em baixo. O projeto deve levar isto em consideração. Use a tabela de designação de pinos da placa DE2.

h) Testar o circuito de transmissão e recepção para vários valores de taxa de transmissão e dados seriais.

i) Mostre o funcionamento dos sinais de estado de transmissão e de recepção usando o osciloscópio.

j) Analise o funcionamento do circuito com a variação da taxa de comunicação do terminal serial.

2.3. Desafio

k) Uma **modificação** será especificada pelo professor. Implemente esta modificação. Atualize a documentação do circuito.

l) Teste o circuito modificado na placa Altera DE2. Anote resultados obtidos.

2.4. Atividades Pós-Laboratório

m) Após o término das atividades experimentais, responda as perguntas abaixo.

1. *Há alguma limitação de funcionamento do circuito projetado? Que intervalo de velocidades de comunicação serial são possíveis no projeto elaborado?*
2. *O módulo de transmissão serial sofre alguma influência do módulo de recepção? Se sim, qual é esta influência? Explique.*
3. *Os dados de teste (caracteres a serem mostrados no circuito) influem nos testes efetuados? Caso afirmativo explique por que.*
4. *É possível a transmissão e a recepção de dados operem em frequências diferentes? Como isto poderia ser realizado?*

CONFIGURAÇÃO DA COMUNICAÇÃO SERIAL COM O PC

Caso seja usado um computador tipo PC para emular um terminal serial, conecte o cabo serial na porta serial e utilize o software de comunicação serial *HyperTerminal* do Windows (ou outro compatível). Crie uma nova conexão usando os parâmetros abaixo (configurações de porta), como na figura 2.3:

- Conexão: COMx (verifique porta serial)
- Bits por segundo: 110
- Bits de dados: 7
- Paridade: ímpar
- Bits de parada: 2
- Controle de fluxo: Nenhum

Se for necessário mudar algum parâmetro, deve-se criar uma nova conexão.

Figura 2.3 – Exemplo de configuração do HyperTerminal.

Para testar a conexão, faça um curto-circuito com os pinos TX e RX do cabo serial e veja se o que for digitado é ecoado no terminal.

3. BIBLIOGRAFIA

- ALTERA. **Altera DE2 Development and Education Board User Manual**. 2008.
- CHU, P.P. **FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 version.** Wiley, 2008.
- FREGNI, E.; SARAIVA, A. M. **Engenharia do Projeto Lógico Digital: Conceitos e Prática**. Editora Edgard Blücher, 1995.
- HELD, G. **Understanding Data Communications**. 6th edition, New Riders, 1999.
- SIGNETICS. **TTL Logic Data Manual**. Signetics, 1982.
- Fairchild Optoelectronics. Manuais de components. Palo Alto, CA.
- CCITT - Fifth Plenary Assembly. Green Book. Vol. VIII, Geneve, December 1972
- Electronic Industries Association. **Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Date Interchange EIA-RS-232-C**, Washington, August 1969.

4. EQUIPAMENTOS NECESSÁRIOS

- 1 osciloscópio digital.
- 1 terminal serial ou computador com interface serial e software de comunicação.
- 1 computador compatível com IBM-PC com software Altera Quartus II.
- 1 placa de desenvolvimento FPGA DE2 da Altera com o dispositivo Altera Cyclone II EP2C35F672C6.

Histórico de Revisões

E.S.G. e F.N.A/2001 – revisão
E.T.M./2004 – revisão
E.T.M./2005 – revisão
E.T.M./2008 – revisão
E.T.M./2011 – revisão
E.T.M./2012 – revisão
E.T.M./2013 – adaptação e revisão
E.T.M./2014 – revisão
E.T.M./2015 – revisão