

Sistemas Embarcados CSW41

Matheus Mattos

LAB 4

1. Objetivo

O objetivo deste laboratório é de se projetar no IAR um programa que receba os dados pela porta serial UART do processador da Tiva, processar os dados e enviá-los pela mesma porta serial. A comunicação será feita com um emulador de terminal (TeraTerm ou similar) executando no PC. O processamento de dados consiste em converter os caracteres maiúsculos (A-Z) nos correspondentes minúsculos (a-z) sem alterar os demais bytes recebidos. A taxa de comunicação deve ser de 115200 bps no formato 8N1.

2. Planejamento das fases do processo de desenvolvimento

O planejamento se deu por meio de estudos iniciais da atividade a ser desenvolvida, do funcionamento da plataforma com as funções UART, das funções disponíveis na TivaWare para comunicação da porta serial UART e um estudo sobre o funcionamento do Tera Term.

3. Definição do problema a ser resolvido

O principal problema dessa atividade, que trabalha com comunicação por porta serial por meio da UART, foi compreender a utilização das suas funções e da sua biblioteca e suas pinagens utilizada além da integração com a Tiva.

4. Estudo da plataforma de HW e SW

Foi necessário fazer um estudo das funções UART para a placa Tiva, compreender como utilizar e gerenciar as suas interrupções e como fazer ela receber e enviar dados (Rx,Tx) pela mesma porta serial. Além de relembrar o funcionamento do algoritmo de enfileiramento FIFO e como configurar o baud rate, foi necessário estudar o funcionamento do Tera Term que foi utilizado para fazer a comunicação com a placa Tiva.

5. Projeto Design da solução

O problema consiste, basicamente, em transformar caracteres maiúsculos em minúsculos, sem alterar os outros bytes que forem recebidos. Para que isso pudesse ser resolvido, se fez necessário a utilização de um terminal, no caso o Tera Term, para que pudessem haver entradas externas do usuário para verificar o correto funcionamento do programa. Resumidamente, o programa inicia, verifica o buffer e se ele foi alterado, além de verificar se é um caractere ASCII válido, caso sim, transforma-se em minúsculo. Se ele for um caractere minúsculo, ele se mantém e é reimpresso. Para isso, foram inicializados os periféricos. Depois foi utilizado o protocolo de comunicação UART, um receptor e transmissor assíncrono universal. Ele é utilizado para transferir dados serialmente entre dois dispositivos em aplicativos de sistemas embarcados. Apenas dois pinos são usados para transferir e receber dados, como um pino de transmissão (Tx) e um pino do receptor (Rx). Outros dois pinos são apenas

pinos de fonte de alimentação, como aterramento e Vcc (geralmente uma fonte de alimentação do dispositivo). Utilizando o manual da placa, podemos verificar na tabela abaixo na qual o GPIO PortA 0 possui a função de receptor (U0Rx) e o GPIO PortA 1 possui a função de transmissor (U0Tx):

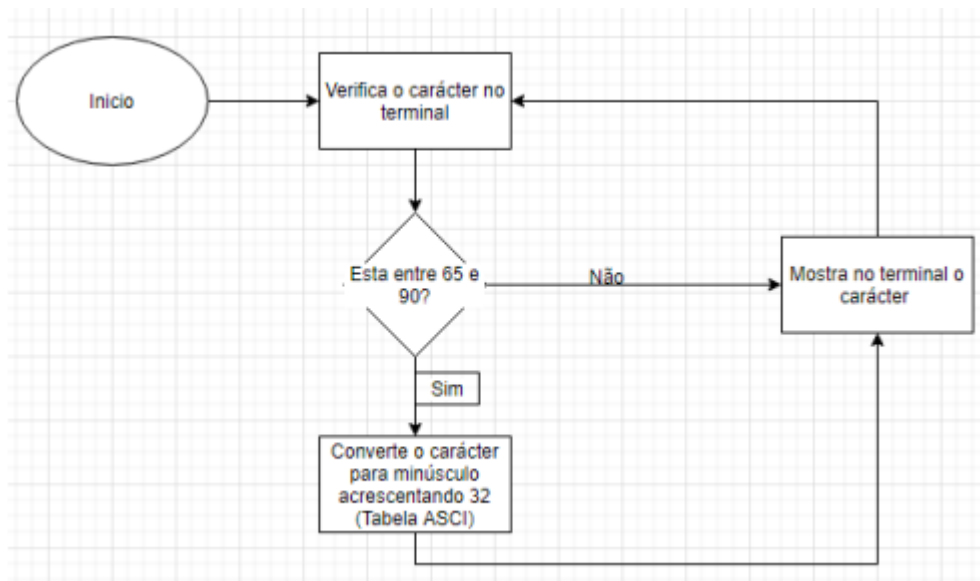
IO	Pin	Analog or Special Function ^a	Digital Function (GPIOCTL PMCx Bit Field Encoding) ^b											
			1	2	3	4	5	6	7	8	11	13	14	15
PA0	33	-	U0Rx	I2C9SCL	T0CCP0	-	-	-	CAN0Rx	-	-	-	-	-
PA1	34	-	U0Tx	I2C9SDA	T0CCP1	-	-	-	CAN0Tx	-	-	-	-	-

Também existe uma porta UART0 que trabalha por meio de uma porta USB, que é conectada ao computador. De modo a utilizar essas funções, as configurações e funções que devemos utilizar são encontradas no manual do TivaWare. Na interrupção UART0 se verificam os caracteres recebidos na FIFO, e caso seu valor seja entre 65 e 90 ele é um caractere maiúsculo, seguindo a tabela ASCII, somando 32 ao seu valor, pois as letras minúsculas são representadas entre 97 e 122. Se ele for transformado, é retransmitido em letra minúscula, e caso já esteja assim, é só enviado novamente. Podemos verificar esses caracteres na tabela abaixo:

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

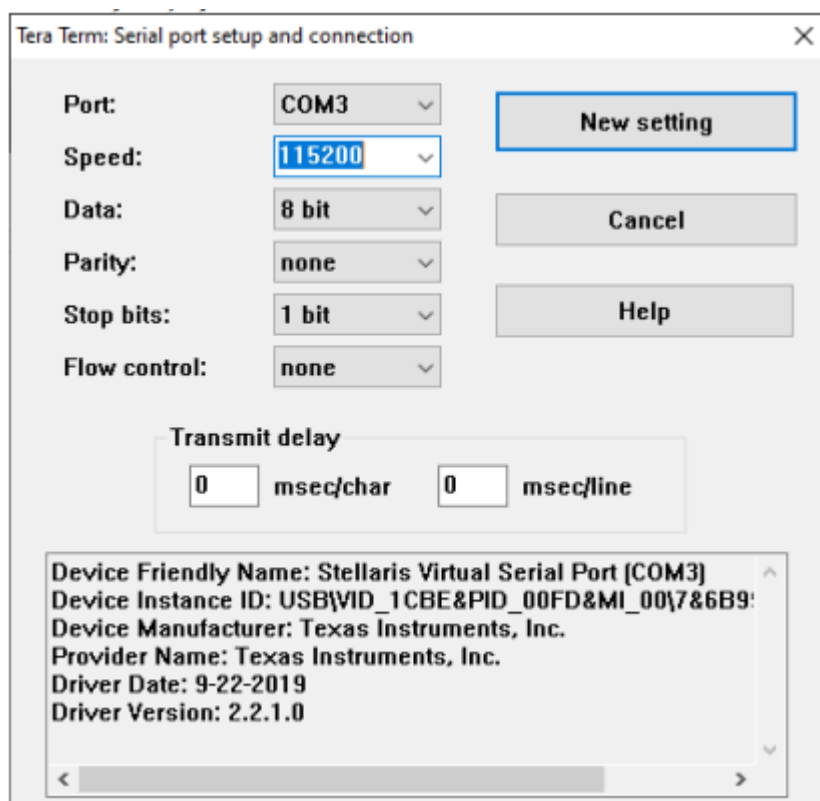
Também foi necessário setar o baud rate em 115200 bps, tanto no IAR quanto nos emuladores, para que seu funcionamento ocorresse corretamente.

Fluxograma da solução desenvolvida:



6. Configuração do projeto na IDE (IAR)

Para este projeto foi utilizado as mesmas configurações do Lab2 na IDE, as únicas alterações que foi necessário fazer foi no Tera Term, onde foi necessário configurar a placa para o baud rate estipulado no começo do projeto.



7. Teste e depuração.

O teste foi feito através do Tera Term onde foi digitados diversos caracteres no terminal entre maiúsculo e minúsculo, conforme era digitado uma letra maiúscula era

transformado em minúscula e reimpresso no terminal, já para letra minúscula não era feito nenhuma transformada somente era impresso novamente no terminal.