

Introducció:

L'objectiu d'aquesta pràctica és conèixer la UART (port sèrie) del micro i introduir-nos en la seva programació. Hem dividit aquest tema (les comunicacions sèrie) en tres pràctiques per tal d'anar assolint en successives etapes els diferents conceptes. Aquesta pràctica n'és la primera.

Els dispositius de la família MCS'51 implementen en el seu maquinari una UART (Universal Asynchronous Receiver Transmitter) que permet efectuar comunicacions sèrie síncrones o bé asíncrones sobre diferents 'protocols' (RS-232, RS-485...). En el nostre cas explotarem un 'sub-conjunt' de la norma RS-232. Consulteu el 'datasheet' del 8051 (o equivalent) per als detalls.

Aprofitant el fet que, per desenvolupar, tenim connectada la placa DS-552 al PC a través d'un port sèrie i que, sobre aquest últim, corre una aplicació de tipus 'terminal' (Procom, Hyper Terminal...) veurem en la pantalla del PC tot allò que arriba per la línia sèrie. Tanmateix, totes les tecles premudes del PC seran enviades per aquesta línia. Tota aquesta informació (alfanumèrica) 'viaja' seguint una codificació 'numèrica', el codi ASCII (American Standart Code for Interchange of Information) que ha esdevingut un codi molt extès.

Cal recordar que hi ha alguns valors d'aquest codi que 'no tenen representació' (ex. l'escape) però sí una 'funció' o 'sentit' i que hi ha 'tecles' (ex. el retorn) que no corresponen a un únic valor ASCII sinó a dos (0Dh i 0Ah, 'retorn de carro' i 'avanç de línia'). Annexat a aquest document hi ha una taula ASCII.

Enunciat de la pràctica.

- 1) Feu un programa senzill (no més de deu línies de codi, en ensamblador) que envia repetidament un caràcter per la línia sèrie. Aproveu-vos del fet que la línia sèrie ja és inicialitzada (pel carregador) i no toqueu cap configuració (timers ni interrupcions).

Guieu-vos amb la subrutina 'PutChar' de la nota d'aplicació AN440 de Philips en la que es descriu un carregador de codi (pràcticament idèntic al que hi ha en el nostre sistema de desenvolupament). El resultat de la execució d'aquest programa en el laboratori ha de ser que, un cop carregat, us van apareixent a la pantalla del terminal un munt de caràcters idèntics (fins que atureu la placa). Amb això "garantim" que la línia sèrie funciona en un sentit (del micro cap al PC).

- 2) Partint del codi de la pràctica 2 (tecles i LEDs), modifiqueu-lo (afegir el necessari) per tal de que, a més d'encendre els LEDs, enviï els caràcters '1', '2', '3' i '4' (valors del 31h al 34h) per la línia sèrie cada cop que s'encén el LED corresponent.

El funcionament d'aquest programa, sobre la placa, ha de ser idèntic al de la pràctica anterior i, a la pantalla del PC, us apareixeran una sèrie de números corresponents a la seqüència d'enceses de LEDs. En aquest cas, tampoc cal inicialitzar el port sèrie però cal tenir en compte que, si el vostre codi fa servir un timer, podeu entrar en conflicte amb la configuració que fa el carregador.

Si es dona aquest cas, canvieu de timer (el "zero" pel "un", o al revés) i no feu 'escriptures absolutes' sobre els registres de control (ex. 'MOV TMOD,#20h'). Respecteu els valors existents amb codi del tipus 'ANL TMOD,#0Fh i ORL TMOD,#20h' a l'hora de configurar els timers.

3) Modifiqueu el codi de l'apartat anterior (un cop funcioni correctament) per tal de llegir les comandes de la línia sèrie.

En aquest cas, a més de consultar l'estat de les tecles, es mirarà SI han arribat els caràcters '1', '2', '3' ó '4' per la línia sèrie (resultat de pitjar les tecles del PC) i tindrà el mateix efecte que pitjar les tecles de la placa. Podeu basar-vos en la subrutina 'GetChar' tenint en compte que, un cop cridada, atura l'execució del programa fins que no ha arribat un caràcter. Això pot fer que es deixi d'atendre les tecles, actueu en conseqüència (modifiqueu-la per que retorni '0' en cas que no ha arribat caràcter).

Aquesta pràctica no us ha de portar més feina de la que feu en una sessió (dues hores). En cas contrari, us manca assolir els objectius de les pràctiques anteriors.

**TABLE 1-2
COMMON ALPHANUMERIC CODES**

ASCII Symbol	HEX Code for 7-Bit ASCII	EBCDIC Symbol	HEX Code for EBCDIC	ASCII Symbol	HEX Code for 7-Bit ASCII	EBCDIC Symbol	HEX Code for EBCDIC	ASCII Symbol	HEX Code for 7-Bit ASCII	EBCDIC Symbol	HEX Code for EBCDIC
NUL	00	NUL	00	*	2A	*	5C	T	54	T	E3
SOH	01	SOH	01	+	2B	+	4E	U	55	U	E4
STX	02	STX	02	.	2C	.	6B	V	56	V	E5
ETX	03	ETX	03	-	2D	-	60	W	57	W	E6
EOT	04	EOT	03	.	2E	.	4B	X	58	X	E7
ENQ	05	ENQ	2D	/	2F	/	61	Y	59	Y	E8
ACK	06	ACK	2E	0	30	0	F0	Z	5A	Z	E9
BEL	07	BEL	2F	1	31	1	F1		5B		AD
BS	08	BS	16	2	32	2	F2	λ	5C	NL	15
HT	09	HT	05	3	33	3	F3		5D		DD
LF	0A	LF	25	4	34	4	F4	.	5E		5F
VT	0B	VT	0B	5	35	5	F5	—	5F	—	6D
FF	0C	FF	0C	6	36	6	F6	.	60	RES	14
CR	0D	CR	0D	7	37	7	F7	a	61	a	81
SO	0E	SO	0E	8	38	8	F8	b	62	b	82
S1	0F	S1	0F	9	39	9	F9	c	63	c	83
DLE	10	DLE	10	:	3A	:	7A	d	64	d	84
DC1	11	DC1	11	:	3B	:	5E	e	65	e	85
DC2	12	DC2	12	—	3C	—	4C	f	66	f	86
DC3	13	DC3	13	=	3D	=	7E	g	67	g	87
DC4	14	DC4	35	\	3E	\	6E	h	68	h	88
NAK	15	NAK	3D	?	3F	?	6F	i	69	i	89
SYN	16	SYN	32	@	40	@	7C	j	6A	j	91
ETB	17	EOB	26	A	41	A	C1	k	6B	k	92
CAN	18	CAN	18	B	42	B	C2	l	6C	l	93
EM	19	EM	19	C	43	C	C3	m	6D	m	94
SUB	1A	SUB	3F	D	44	D	C4	n	6E	n	95
ESC	1B	BYP	24	E	45	E	C5	o	6F	o	96
FS	1C	FLS	1C	F	46	F	C6	p	70	p	97
GS	1D	GS	1D	G	47	G	C7	q	71	q	98
RS	1E	RDS	1E	H	48	H	C8	r	72	r	99
US	1F	US	1F	I	49	I	C9	s	73	s	A2
SP	20	SP	40	J	4A	J	D1	t	74	t	A3
!	21	!	5A	K	4B	K	D2	u	75	u	A4
"	22	"	7F	L	4C	L	D3	v	76	v	A5
#	23	#	7B	M	4D	M	D4	w	77	w	A6
\$	24	\$	5B	N	4E	N	D5	x	78	x	A7
%	25	%	6C	O	4F	O	D6	y	79	y	A8
&	26	&	50	P	50	P	D7	z	7A	z	A9
'	27	'	7D	Q	51	Q	D8	{	7B	{	8B
(28	(4D	R	52	R	D9		7C		4F
)	29)	5D	S	53	S	E2	}	7D	}	9B
								-	7E	-	4A
								DEL	7F	DEL	07

**TABLE 1-3
DEFINITIONS OF CONTROL CHARACTERS**

NULL	Null	DC1	Direct control 1
SOH	Start of heading	DC2	Direct control 2
STX	Start text	DC3	Direct control 3
ETX	End text	DC4	Direct control 4
EOT	End of transmission	NAK	Negative acknowledge
ENQ	Enquiry	SYN	Synchronous idle
ACK	Acknowledge	ETB	End transmission block
BEL	BS	CAN	Cancel
BS	Backspace	EM	End of medium
HT	Horizontal tab	SUB	Substitute
LF	Line feed	ESC	Escape
VT	Vertical tab	FS	Form separator
FF	Form feed	GS	Group separator
CR	Carriage return	RS	Record separator
SO	Shift out	US	Unit separator
SI	Shift in		
DLE	Data link escape		