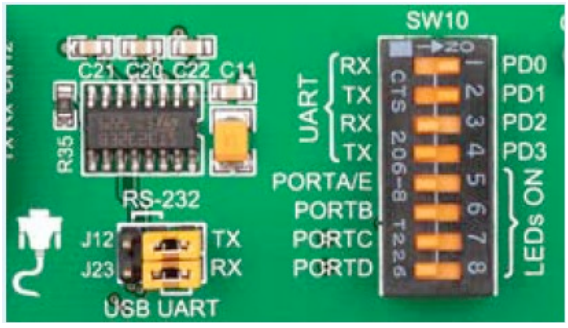
COMP/ELEC 317

Lab 3 - PWM & UART

In this lab, you are going to experiment with using the PWM resources of the AVR processor to generate a beep noise from the buzzer on your AVR card and to use the UART resources to drive the number and duration of these beeps so that you can generate Morse code from typed letters.

1. First, you need to configure your AVR card so that you can use its "USB RS232" connector with additional USB cable (see Preliminary work-8). In order to configure:
    a. Set J23 and J12 jumpers in the USB-UART position
    b. Enable (turn ON) line 1 and 2 on the switch SW10



    c. Keep the lines 5,6,7,8 ON for the same switch (SW10)
2. Find the sample codes, USB driver and the terminal emulation program under the F-drive share directory. When you connect USB serial communication cable if the system does not install necessary drives, use the CDM20814_Setup.exe to setup USB drivers.
3. Download the code **uart_int.asm** from the course web site (**lab3.zip**). Assemble and run it. You will need to load a terminal program running on the PC in order to communicate with this program. You can find two terminal programs in the lab3 package. You can use whichever one you choose. You need to set the appropriate RS232 parameters and select the correct COM port. Make sure that in AVRFLASH programming Internal RC Oscillator and Device Frequency are set to the correct clock frequency.
    a. Describe how **uart_int.asm** works?

4. Get the code **sinew.asm** from the web site. Assemble and use it. Check if it actually works by connecting the PWM output to the buzzer (you will see cylindrical shaped black component, remove its seal and set its jumper J21 in PD4 position or if you desire you can change it to PB1 position in order to drive buzzer from PB1 pin).
    a. Describe how this code works.
    b. What code changes do you need to do in order to;
        i. double the frequency of the output?
        ii. make it one half?

5. Combine the code in part 1 with the code in part 2 such that the resulting code not only lights the LEDs according to the typed character but also beeps the Morse code corresponding to the typed letter as per the following table:

A	·-	K	-·-	U	··-	5	·····	:	---···
B	-···	L	·-··	V	···-	6	-····	/	-··-·
C	-·-·	M	--	W	·--	7	--···	-	-····-
D	-··	N	-·	X	-··-	8	---··	'	·----·
E	·	O	---	Y	-·--	9	----·	()	-·--·-
F	··-·	P	·--·	Z	--··	0	-----	_	··--·-
G	--·	Q	--·-	1	·----	,	--·---		
H	····	R	·-·	2	··---	.	·-·-·-		
I	··	S	···	3	···--	?	··--··		
J	·---	T	-	4	····-	;	-·-·-		

Your code should ignore all the other characters not listed here and case should not make a difference. Choose an appropriate duration for the "dot". The duration of a "dash" should be three times that of a "dot". The silence between dots and dashes in a letter is as long as a dot. The silence between letters in a word is as long as a dash. And, the silence between words is as long as two dashes. An ASCII table is provided below.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

Preliminary work:

6. Read Timer/Counters and UART sections of the datasheet of your microprocessor.
7. Build an ASM chart for your design. Submit your design before the lab.
8. You can borrow additional USB cable from our Lab technician Selim Ölçer.

Procedure:

1. Create a new folder, **last_name1-last_name2-lab3**, in your home directory
2. Do your programming on the source code. Assemble.
3. Answer all the questions in parts (1-4) at the top of your source code under comment lines. Include your preliminary ASM chart design with any final updates.
4. Compress your entire **last_name1-last_name2-lab3** project folder, and submit the compressed folder to ku.blackboard.com web page.
5. Write up a lab report including flow charts of your implemented algorithms and a discussion of challenges and learning outcomes. Submit your report separately to ku.blackboard.com web page.

Each group should work independently! Copying or idea exchanges are not allowed!

Good luck!