

Laboratory Assignment #2:
“ASCII to Morse Player”

Due Monday, October 14, 2002

In this lab you will learn to use the AVR’s serial communications interface (the UART) and the Timer1 output-captures to generate tones. Your goal is to take characters from the serial port and play them on the speaker as their Morse code equivalent.

The International Morse Code is made up of a series of “dots” and “dashes,” which are represented by long and short tones. A dash is three times as long as a dot. The silence between dots and dashes within a letter is the same as a dot. The silence between letters is the same as a dash. The silence between words is the same as two dashes. Code speed is expressed in words per minute, and the calculation of words per minute is based upon how many times the word “PARIS” can be sent in one minute at a given code speed.

The Morse code representations of the letters, digits, and punctuation are shown below.

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	-----	;	-----		

There are a variety of ways you can choose to encode the Morse descriptions of the characters. For example, you could create a table indexed by the ASCII value, containing two bytes per character. The first byte would contain the number of elements, and the second byte could contain a bitwise representation of the character, with a 1 for a dash and a 0 for a dot. You may find the AVR instruction **LPM** and the assembler directive **.DB** helpful for reading data and placing data in program (FLASH) memory.

Your program should have a means of setting the code speed directly in Words Per Minute. Setting of the code speed can be done at compile-time, and can remain constant during execution. The program should accept characters from the serial port (9600bps, 8bits data, no stop bits) into a buffer of at least 100 characters. If the buffers overflow, it may drop further characters until space becomes available in the buffer by outputting characters. You should completely ignore (produce no output for) characters not defined in the table above. As an intermediate stage of program development, you may wish to simply output your morse code to an LED before attempting to use the Timer to create beeps. Beeps should have a reasonable tonality to them; shoot for about a 2kHz tone. Speakers are available in the lab.

Submit: Your .asm file, and a readme.txt explaining your code in reasonable detail (a few paragraphs). Zip these files and send them to the TA by the due date.