****
Proj_1B_LED_display   Driving the leds in a slightly different way. The display cycles round and round rather than repeating endlessly.

SOME OF THE THINGS ABOUT C THAT IT INTRODUCES:

1. The char variable: A char variable is stored in a location in data memory that holds only 8 bits of data rather than 16 used for integer variables. A 'char' variable can hold 2^8 = 256 different values i.e. numbers from 0 to 255. A 'signed char' can hold numbers from -128 to +127 (unsigned is the default option for chars (in WINAVR only!).

2. Binary numbers: Numbers are saved in data memory as ones and zeros. For example 1 is saved as 1, 2 as 10, 4 as 100, 8 as 1000, 16 as 10000. These will be considered in more detail later.

3. Hex numbers: These will also be considered in more detail later. Note however that the hex number 0x8000 is 1000 0000 0000 0000 in binary.

4. Some logic: i.e. the 'or' and 'and' functions and the shift '<<' or '>>' operators.

5. The '|' (or) function. For example 10001010 | 10101000 = 10101010
The short hand notation 'a |= b' which means that 'a' is set equal to 'a | b'.

6. The 'if-else' statements: Consider the statements
if (m<=5){PORT_1 |= (PORT_1 << 1); m += 1;} else PORT_1 = PORT_1 << 1;
The statements '{PORT_1 |= (PORT_1 << 1); m += 1;}' are only executed if 'm' is less than 6
Otherwise the statement 'PORT_1 = PORT_1 << 1' is executed.

7. The '&' (and) function 11001010 & 10101000 = 10001000 Consider the statement

if (PORT_1 & 0x8000) overflow=1;
It is used here to test the most significant bit (MSB) of PORT_1.

Note: the 'for-loop' is omitted, instead the value of 'PORT_1' is tested every time that the 'while-loop' is executed. After 15 repetitions the least significant bit of Port_1 is reset to 1.

MORE ABOUT THE PROJECT FIRM WARE (FW)

Project subroutine 'waitforkeypress()': This does just does what it says. It can be use here in place of the 'T0' delay by anyone interested in exploring the operation of the logic. Repeatedly press any key (AK) to see the display develop in slow time.

Note: 'waitforkeypress();' brings program execution to a halt awaiting user input from the keyboard. This is not really good practice. There are alternatives which make use of the watch dog timer.

MORE ON THE LOGIC

Note: a|b and a&b operate on the bits defined by -a- and -b-
00001010 | 10101000 = 10101010 and 11101010 & 10100011 = 10100010

If PORT_1 starts of as 00000111 then the statement PORT_1 = PORT_1 << 1 changes it to 00001110
and the statement PORT_1 |= (PORT_1 << 1) changes it to 00000111 | 00001110 which equals 00001111

if PORT_1 = 1xxxxxxxxxxxxxxx then "overflow" is set to save the LH 1 which will otherwise be lost at the next shift left. It is then placed in the most RH location. Note x is either 0 or 1; LH is left hand, RH is right hand.