

\*\*\*\*\*

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).
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  $10001010 \mid 10101000 = 10101010$   
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. See Proj\_1B1.

Note: 'waitforkeypress();' brings program execution to a halt awaiting user input from the keyboard. This is not really good practice. Alternatives will be considered in Proj\_2 which deals with the watch dog timer.