****

Proj_2B1_watch_dog_timer

SNOW STORM DISPLAY: Uses the watch dog timer to prevent the display from getting jammed in a way that sometimes affected earlier programs

PROVIDES MORE ON

1. Random number generation: Project subroutine PRN_16bit_GEN (0) uses a number stored in EEPROM to generate another one which it saves in the same EEPROM location. In this way repeated calls to PRN_16bit_GEN (0) generate a sequence of random numbers.

Note:

a. EEPROM memory is not affected by POR or a WD timeout.

b. PRN_16bit_GEN (Num) generates a random number based on Num and it is up to the project to increment Num in some way.

2. The Watch Dog Timer Once this has been set running it will reset the program after a predetermined time interval (250mS in this case) unless it is first reset itself using command wdr().

OPERATION

Pressing sw_2 calls subroutine ISR(PCINT2_vect) which contain the statement while(1); This halts program flow because there are no active interrupts at this point and therefore the watch dog timer cannot be reset and will time out.

Note

1. Interrupts are automatically disabled when program execution enters an ISR. They can however be re-enabled using sei() in which case one ISR can be interrupted by another interrupt.

2. The display pauses but continues in sequence, because the random number generator uses the EEPROM for its data storage.

3. In practice the watch dog delay is made as short as possible without interfering with normal program operation so that any pause is insignificant.

4. The switch is used to allow us to test the operation of the WDT. Normally we only want it to timeout when a real glitch in the system causes the program to crash.

IT ALSO INTRODUCES

1. Project subroutine I2C_Tx(). This transmits data to the PCB 111000_1 Atmega 328 over the I2C bus.

2. Pointers: Look at the ISR: The following statement: I2C_Tx(1, 'Q', &Dimmer_control); is used to call subroutine I2C_Tx();

The -&- before the variable Dimmer_control means that the subroutine I2C_Tx() expects the calling routine to provide the data memory location of the variable (i.e. its address). It uses and possibly modifies whatever data it finds in that memory location.

In this case Proj_1B1 defines a memory location and calls it "Dimmer_control". The subroutine call hands the address of Dimmer_control to the subroutine rather than the actual value.

3. Subroutine void I2C_Tx(char num_bytes, char mode, char* s){}.

This is the subroutine being called by the statement I2C_Tx(1, 'Q', &Dimmer_control);

It provides memory locations num_bytes and mode for the 1 and Q. The -*- signifies that it does not provide memory for variable -s- but expects the calling routine to provide it.

Note: Q is the mini-OS mode that toggles the display brightness.

The pointer is used because I2C_Tx() is also used to send arrays of data (to be considered later).