

\*\*\*\*\*

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    The LED dimmer.    A test routine for the display brightness control.  
Uses an Interrupt on key press to toggle the brightness control.

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: It makes the following subroutine call  
         I2C\_Tx(1, 'Q', &Dimmer\_control);

The "&" before the variable "Dimmer\_control" means that the subroutine "I2C\_Tx" expects the calling routine to provide the data memory location (i.e. its address) and uses and possibly modifies whatever data it finds in that memory location.

In this case Proj\_2B1 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).