****

Proj_1D_Key_press_interrupt

ANOTHER VARIATION ON Proj_1A This time not all sets of vertical LEDs are illuminated in any given sweep.

DISCOVERING A BIT MORE ABOUT THE ATMEGA HW

1. The Serial Port (UART): This is also known as the Universal Asynchronous Receiver Transmitter or just Receiver Transmitter (Rx/Tx). It enables communication with the PC. Because PCs require a USB connection a USB bridge is included on the UNO PCB. It provides an interface between the Atmega serial port and PC USB port.

2. HW register UCSR0B: This is known as the UART Control and Status Register B. Bit 7 of this register is RXCIE0 (Rx complete interrupt enable). User code must set this bit to 1 if an interrupt is required when a character is received. Note also the registers used in waitforkeypress() and receiveChar().

3. ISR(USART_RX_vect){} This is the ISR that responds to iterrupts generated using keypresses made at the PC keyboard.

INTRODUCING SOME NEW C TOPICS

1. The switch block: This includes a switch statement and the break instructions that cause program execution to exit the switch block.

2. Instructions cli(); used in combination with sei(); to temporarily disable interrupts.

3. The statement row = row%2;: This gives the remainder of dividing the variable row by 2, it only takes on the values 0 and 1. (A%2 is spoken as A modulo 2).

4. The Statement while(A); (See local version of waitforkeypress()). This halts program execution while A is non-zero. Similarly statement while(!(A)); halts program execution while A is zero. A can be changed by an interrupt or by the hardware when for example the UART receives a character.

PROGRAM OPERATION

After uploading the sketch open a terminal program and click on connect (57600 Baud, no parity or handshaking, 8 data bits and 1 stop bit). Then press 1,2,3,4,5,6,7 or x.

Before uploading a sketch again remember to disconnect the terminal program.

The main process that of controlling the display is interrupted for a brief instant by a keypress. This calls an ISR that affects the main process, changing the number of leds that are skipped every time that the display increments.

WHAT REALLY MATTERS HERE

The Rx interrupt (the keypress interrupt) and setting it up, subroutine ISR(USART_RX_vect){}, and the switch statement block.

POSSIBLY for LATTER REVIEW

The division and logic might possibly be worth revisiting latter on but really they have just been included in an attempt to do something different.