```
/*Proj_1C_LED_display

Testing your reaction time
*************************************************************/


#include "Proj_1C_header_file.h"


volatile unsigned int PORT_1, mask;     //Variables used by both the main routine and also by the ISR


int main (void){
setup_HW;
config_sw3_for_PCI;                     //Enable PCI interrupt on switch_3

mask = 0xFFFF;                          //0xFFFF = 0b1111111111111111 indicating that none of the leds have yet been shot down
sei();                                  //Enable all interrupts
while(mask){                            //Exit the "while-loop" as soon as mask gets set to zero
PORT_1=1;                               //Initialise display to 0000 0000 0000 0001

for(int m = 1; m < 17; m++){            //Repeat "for-loop" 16 times
I2C_Tx_2_integers
(PORT_1 & mask, (~mask) ^ PORT_1);      //LOGIC: "Dead" leds are transferred to the bottom row

Timer_T0_10mS_delay_x_m(10);            //Program execution spends most time waiting here, so this is where the interrupt almost always occurs
PORT_1 = (PORT_1 << 1);                 //Move on to next display location
}}I2C_Tx_2_integers(0, 0xFFFF);         //When all leds are dead illuminate all the bottom leds and then
Timer_T0_10mS_delay_x_m(100);           //pause for 1 sec before starting all over again.
SW_reset;}


/**************ISR Routine executed every time that switch 3 is  operated************************************/
ISR(PCINT0_vect) {                      //This ISR momentarily interrupts the main routine
if(switch_3_up)return;                  //It notes which LED has just been shot down and
mask &= ~PORT_1;}                       //writes zero at its location in the "mask" register
```