```c
#include "Proj_5E_header_file.h"

volatile char T1_ovf_flag = 0;                               //requires volatile label
                                                            //even though "T1_ovf_flag" is used in "main"
int main (void){

setup_HW;
User_prompt;
User_instructions;

setup_and_enable_PCI;
disable_pci_on_sw1_and_sw2;                                  //pci on sw1 & 2 not required
{char digit_num=0; for (int m = 0; m < 8; m++)               //initialise display by
{I2C_Tx_any_segment('d', digit_num++);}}                     //illuminating seg "d" on each digit

while(switch_1_up && switch_2_up);                           //wait for a keypress
I2C_Tx_any_segment_clear_all();                              //clear display
if(switch_2_down){I2C_Tx_Clock_command(one100ms_mode);}      //sw_1 selects 100ms clock
if(switch_1_down){I2C_Tx_Clock_command(ten_ms_mode);}        //sw_2 selects 10ms clock
Timer_T0_10mS_delay_x_m(5);                                  //50ms delay for switch bounce
while(switch_1_down || switch_2_down);                       //wait for switch release

TIMSK1 |= (1 << TOIE1); sei();                               //Enable timer 1 interrupt

while(1){                                                    //Infinite loop, requires sw2 interrupt to exit
while(switch_1_up && switch_2_up);                           //wait for a keypress

if(switch_1_down)
{I2C_Tx_Clock_command(store_time);                           //save time and pause clock display
while (switch_1_down);                                       //wait for key release
I2C_Tx_Clock_command(display_current_time);}                 //clock display continues as normal

Timer_T1_sub_with_interrupt(T1_delay_50ms);                  //50mS pause: avoids switch bounce
while(T1_ovf_flag == 0);T1_ovf_flag = 0;                     //T1_ovf_flag is reset by T1 ISR after a 50ms delay

if(switch_2_down)                                            //read stored times at 200ms intervals
{I2C_Tx_Clock_command(display_stored_times);                 //if sw1 is held down
Timer_T0_10mS_delay_x_m(20);}
if(switch_1_down && switch_2_down)                           //press sw2 while sw1` is still held down to
{I2C_Tx_Clock_command(display_current_time);                 //continue clock display
while(switch_1_down || switch_2_down);} }}                   //wait until both switches have been released


ISR(TIMER1_OVF_vect) {TCCR1B = 0; T1_ovf_flag = 1;}          //stop timer 1 and set overflow flag

ISR(PCINT0_vect) {if(switch_3_up){return;}
I2C_Tx_Clock_command(AT_exit_stop_watch);
SW_reset;}
```