

```
int main (void){
setup_HW_Arduino_IO;
set_up_PCI;
enable_PCI;

disable_PCI_on_sw1_and_sw3;
{char digit_num=0; for (int m = 0; m < 8; m++)
{I2C_Tx_any_segment('d', digit_num++);}}

while(switch_1_up && switch_3_up);
I2C_Tx_any_segment_clear_all();
if(switch_1_down){I2C_Tx_Clock_command(one100ms_mode);}
if(switch_3_down){I2C_Tx_Clock_command(ten_ms_mode);}
Timer_T0_10mS_delay_x_m(5);
while(switch_1_down || switch_3_down);

TIMSK1 |= (1 << TOIE1); sei();

while(1){
while(switch_1_up && switch_3_up);

if(switch_3_down)
{I2C_Tx_Clock_command(store_time);
while (switch_3_down);
I2C_Tx_Clock_command(display_current_time);}

Timer_T1_sub_with_interrupt(T1_delay_50ms);
while(T1_ovf_flag == 0);T1_ovf_flag = 0;

if(switch_1_down)
{I2C_Tx_Clock_command(display_stored_times);
Timer_T0_10mS_delay_x_m(20);}
if(switch_1_down && switch_3_down)
{I2C_Tx_Clock_command(display_current_time);
while(switch_1_down || switch_3_down);}}

//pci on sw1 & 3 not required
//initialise display by
//illuminating seg "d" on each digit

//wait for a keypress
//clear display
//sw_1 selects 100ms clock
//sw_3 selects 10ms clock
//50ms delay for switch bounce
//wait for switch release

//Enable timer 1 interrupt

//Infinite loop, requires sw2 interrupt to exit
//wait for a keypress

//save time and pause clock display
//wait for key release
//clock display continues as normal

//50mS pause: avoids switch bounce
//T1_ovf_flag is reset by T1 ISR after a 50ms delay

//read stored times at 200ms intervals
//if sw1 is held down

//press sw3 while sw1` is still held down to
//continue clock display
//wait until both switches have been released
```