

```

int main (void){
char User_response;
char start_time[8], digits[8], display_mode;
unsigned char input_mode;
char keypress;

setup_HW_Arduino_IO;

if(switch_3_down){eeprom_write_byte((uint8_t*)(0x02),255);          //Set the EEPROM location 2 to its default value
Serial.write("EEPROM reset\r\n");
while(switch_3_down);SW_reset;}

if (switch_1_up)                                                    //Push SW1 to avoid user prompt
{User_prompt_A;}
while (switch_1_down);
User_instructions;
input_mode = eeprom_read_byte((uint8_t*)0x02);                    //Read mode. Default value is 255

switch (input_mode){
case 255:                                                            //If EEPROM contains 255 initiate the clock
Serial.write("\r\nSend time: hours mins secs\
(24Hr clock)");

clear_display;
for (int m = 0; m < 8; m++)
{
keypress = waitforkeypress_A();                                    //Obtain time from KBD

if((m == 2) || (m == 5))
{digits[0] = ' ';save_to_eeprom;
shift_display_left; m += 1;}

digits[0] = keypress;
save_to_eeprom;
I2C_Tx_8_byte_array(digits);
if (m < 7){shift_display_left;}}

eeprom_write_byte((uint8_t*)(0x02),0);                            //Update mode and save in EEPROM location 2

Serial.write("\r\nPress SW1 & Power cycle!:\
Can now use 5V USB charger if required");

break;                                                            //Exit and send Start clock command

case 0:                                                            //If EEPROM location 2 contains 0 start clock immediately
eeprom_write_byte((uint8_t*)(0x02),255);                          //Restore the EEPROM location 2 to its default value
break;

default: eeprom_write_byte((uint8_t*)(0x02),255);                //If EEPROM ever gets corrupted reset it to 255 (0b11111111)
wdt_enable(WDTO_15MS); while(1);break;}                          //Exit and start clock

for (int m = 0; m < 8; m++)
{start_time[m] =
eeprom_read_byte((uint8_t*)(m+3));}

I2C_Tx_OS_timer(AT_clock_mode, start_time);                      //Send Start clock command (AT clock mode is 7)
display_mode = 0;

while(1){

switch (display_mode){                                           //Beware of switch bounce

case 0: if(switch_2_down){Timer_T0_10mS_delay_x_m(50); display_mode = 'A';}
if(switch_1_down){display_mode = 'B';I2C_Tx_Clock_command(hide_clock);while(switch_1_down);}
if(switch_3_down){display_mode = 'C';I2C_Tx_Clock_command(pause_clock);while(switch_3_down);}break;

case 'A': if(switch_2_up)display_mode = 1; else {while(switch_2_down); display_mode = 2;} break;
case 'B':if(switch_1_down){display_mode = 0;I2C_Tx_Clock_command(display_clock);while(switch_1_down);}break;
case 'C':if(switch_3_down){display_mode = 0; I2C_Tx_Clock_command(display_clock);while(switch_3_down);}break;

case 1: if (switch_3_down){I2C_Tx_Clock_command(increment_seconds);Timer_T0_10mS_delay_x_m(20);}
if (switch_1_down){I2C_Tx_Clock_command(increment_minutes);Timer_T0_10mS_delay_x_m(20);}
if(switch_2_down){ while (switch_2_down);display_mode = 0; }break;

case 2: if (switch_3_down){I2C_Tx_Clock_command(decrement_seconds);Timer_T0_10mS_delay_x_m(20);}
if (switch_1_down){I2C_Tx_Clock_command(decrement_minutes);Timer_T0_10mS_delay_x_m(20);}
if(switch_2_down){while (switch_2_down);display_mode = 0; }break;}}}

void timer_utoa(char n){                                          //converts an unsigned number to a string of digits
if (n>=10)
{charL = n%10;
charH = (n-charL)/10;
charL = charL + '0';
charH = charH + '0';}
else
{charH = '0'; charL = n + '0';}}

```

