```c
#include "Proj_2C1_header_file.h"

char display_bkp[7];                                    //One element to backup each segment letter
char Dimmer_control;

int main (void){
char segment=0, digit_num=0, seg_counter = 0,direction = 0;

setup_HW;
wdt_enable(WDTO_2S);                                    //WDT prevents display from being completed in either direction


UCSR0B |= (1 << RXCIE0);                                //Set Interrupt on key press (for test purposes only)
sei();                                                  //Global enable interrupt
Dimmer_control = 1;


while(1){                                               //Generate pattern
while(seg_counter < 56){                                //There are 56 segments in total
segment = (PRN_16bit_GEN (0)%7) + 'a';
digit_num = (PRN_16bit_GEN (0)%8);
                                                        //Continue statements skip back to the top of the while-loop
                                                        //This is to ensure segments are not turned-off before
                                                        //all have been turned on.

if ((!(direction)) && (display_bkp[segment - 'a'] & (1 << digit_num))) continue;
if ((direction) && (!(display_bkp[segment - 'a'] & (1 << digit_num)))) continue;


I2C_Tx_any_segment(segment, digit_num);                 //Update display
backup_the_display(segment, digit_num);                 //keep backup up to date
Timer_T0_10mS_delay_x_m(5);  wdr();                     //delay and reset watch dog
seg_counter += 1;}

direction ^= 1;                                         //Toggle the direction_counter value
seg_counter = 0;

Timer_T0_10mS_delay_x_m(100);}}                         //Just pause before toggling leds off one at a time


/*************************************************************************/
void backup_the_display(char segment, char digit_num){
display_bkp[segment - 'a'] = display_bkp[segment - 'a'] ^ (1 << digit_num);}

/*************************************************************************/
ISR(USART_RX_vect){
receiveChar();
I2C_Tx(1, 'Q', &Dimmer_control);}


/*Local version of subroutine "I2C_Tx()"

void I2C_Tx_local(char num_bytes, char mode, char* s){
waiting_for_I2C_master;                                 //Turn on I2C slave and await call from master
send_byte_with_Ack(num_bytes);                          //send data byte, request acknowledgement
send_byte_with_Ack(mode);
for (int m = 0;  m < num_bytes; m++){
if (m==num_bytes-1){send_byte_with_Nack(s[m]);}         //Last byte, no ackowledgement needed
else {send_byte_with_Ack(s[m]);}}
TWCR = (1 << TWINT);}                                   //Clear interrupt and close I2C slave*/
```