Supplementary Materials

S1 Schematic circuit diagram

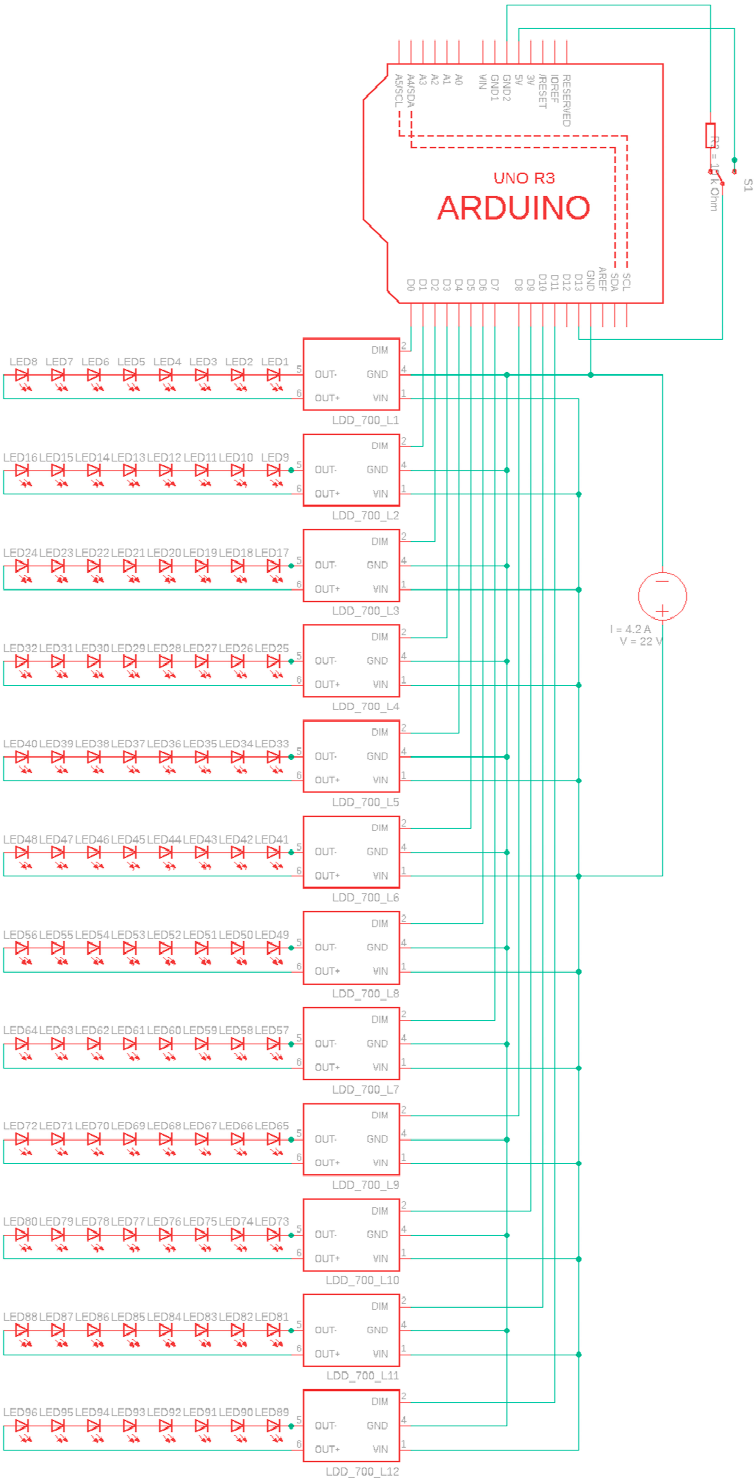

Figure 1. Schematic Control of LEDs using an Arduino Uno.

S2 Example code to control LED rows

This is an example of how the LED driver Meanwell 700 L can be controlled using an Arduino UNO. The program can be copied directly into the free of charge Arduino software, which is available online. LED control can also be programmed as an array, however this does not allow individual frequency control.

Start of code:

```
// This is an example code to control 12 rows of LED with an Arduino UNO microcontroller - program is started with a 5 seconds delay after pressing a simple push button

#define LED1 2 //initializing LED rows
#define LED2 3
#define LED3 4
#define LED4 5
#define LED5 6
#define LED6 7
#define LED7 8
#define LED8 9
#define LED9 10
#define LED10 11
#define LED11 12
#define LED12 13
#define BUTTON 14
#define FREQUENCY1_ON 300000L // Defining an abritrary amount of different Frequencies in which row should be pulsed FREQUENCY1_ON being the on time, FREQUENCY1_OFF being the off time
#define FREQUENCY1_OFF 300000L // time is defined in milliseconds, minimum time is 1 ms when using described setup
#define FREQUENCY2_ON 3000L
#define FREQUENCY2_OFF 0L
#define FREQUENCY3_ON 3000L
#define FREQUENCY3_OFF 444L
#define DELAY 30000L // Define Delay between each LED row
#define RUNTIME1 60000L //Overall runtime per LED row in Millisecs 1 min= 60000 ms 2 min 120000, 3 min 180000, 4 min 240000, 5 min 300000, 6 min 360000, 7 min 420000 etc..
#define RUNTIME2 120000L
#define RUNTIME3 180000L
#define RUNTIME4 240000L
#define RUNTIME5 300000L
#define RUNTIME6 0L
#define ON 1
```

```
#define OFF 0

long int timestamp1 = 0;                    //define a timestamp for each LED row, so each row can be started at a certain time. Here rows are started 30 seconds after preceding row
long int timestamp2 = timestamp1 + DELAY;
long int timestamp3 = timestamp2 + DELAY;
long int timestamp4 = timestamp3 + DELAY;
long int timestamp5 = timestamp4 + DELAY;
long int timestamp6 = timestamp5 + DELAY;
long int timestamp7 = timestamp6 + DELAY;
long int timestamp8 = timestamp7 + DELAY;
long int timestamp9 = timestamp8 + DELAY;
long int timestamp10 = timestamp9 + DELAY;
long long int timestamp11 = timestamp10 + DELAY;
long long int timestamp12 = timestamp11 + DELAY;
long int button_start = 0;

long long int zeit = 0;   //define variables for LED states (defines if LED is on or off)
int ledstate1 = OFF;
int ledstate2 = OFF;
int ledstate3 = OFF;
int ledstate4 = OFF;
int ledstate5 = OFF;
int ledstate6 = OFF;
int ledstate7 = OFF;
int ledstate8 = OFF;
int ledstate9 = OFF;
int ledstate10 = OFF;
int ledstate11 = OFF;
int ledstate12 = OFF;

int running_led1 = 1;   //define LED states switching
int running_led2 = 1;
int running_led3 = 1;
int running_led4 = 1;
int running_led5 = 1;
int running_led6 = 1;
int running_led7 = 1;
int running_led8 = 1;
```

```
int running_led9 = 1;
int running_led10 = 1;
int running_led11 = 1;
int running_led12 = 1;

void setup() {
    pinMode(LED1, OUTPUT); //Declare LED Pins as OUTPUT
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    pinMode(LED5, OUTPUT);
    pinMode(LED6, OUTPUT);
    pinMode(LED7, OUTPUT);
    pinMode(LED8, OUTPUT);
    pinMode(LED9, OUTPUT);
    pinMode(LED10, OUTPUT);
    pinMode(LED11, OUTPUT);
    pinMode(LED12, OUTPUT);
    pinMode(BUTTON, INPUT); //Declare Button Pin as INPUT
}

void loop() {
     if (digitalRead(BUTTON) == HIGH) {   // if button is pressed, start
program and initilize LEDs)
        running_led1 = 1;
        running_led2 = 1;
        running_led3 = 1;
        running_led4 = 1;
        running_led5 = 1;
        running_led6 = 1;
        running_led7 = 1;
        running_led8 = 1;
        running_led9 = 1;
        running_led10 = 1;
        running_led11 = 1;
        running_led12 = 1;

        ledstate1 = OFF;
        ledstate2 = OFF;
        ledstate3 = OFF;
        ledstate4 = OFF;
```

```
        ledstate5 = OFF;
        ledstate6 = OFF;
        ledstate7 = OFF;
        ledstate8 = OFF;
        ledstate9 = OFF;
        ledstate10 = OFF;
        ledstate11 = OFF;
        ledstate12 = OFF;

         timestamp1 = 0;                    // LED row 2 is switched on 30
seconds after LED row 1 - defined by varible DELAY
        timestamp2 = timestamp1 + DELAY;
        timestamp3 = timestamp2 + DELAY;
        timestamp4 = timestamp3 + DELAY;
        timestamp5 = timestamp4 + DELAY;
        timestamp6 = timestamp5 + DELAY;
        timestamp7 = timestamp6 + DELAY;
        timestamp8 = timestamp7 + DELAY;
        timestamp9 = timestamp8 + DELAY;
        timestamp10 = timestamp9 + DELAY;
        timestamp11 = timestamp10 + DELAY;
        timestamp12 = timestamp11 + DELAY;
        delay(5000);
        button_start = millis();

    }

    zeit = millis() - button_start; // set time to the time when button has
been pressed


 // LED1 is off and is switched on after frequency off time is over, can
only be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp1 > FREQUENCY1_OFF && ledstate1 == OFF &&
running_led1) {
        digitalWrite(LED1, HIGH);
        timestamp1 = zeit;
        ledstate1 = ON;
    }

    // LED1 is on and is switched off, after frequency on time ist over
```

```
    if (zeit - timestamp1 > FREQUENCY1_ON && ledstate1 == ON && running_led1)
{
        digitalWrite(LED1, LOW);
        timestamp1 = zeit;
        ledstate1 = OFF;
    }
   // when RUNTIME is over turn LED off and - by setting variable running_led
to 0 - stop if loops that turn LEDs on and off
    if (zeit > RUNTIME1) {
        running_led1 = 0;
        digitalWrite(LED1, LOW);
    }


// LED2
          if (zeit - timestamp2 > FREQUENCY2_OFF && ledstate2 == OFF &&
running_led2) {
       digitalWrite(LED2, HIGH);
        timestamp2 = zeit;
        ledstate2 = ON;
    }

   if (zeit - timestamp2 > FREQUENCY2_ON && ledstate2 == ON && running_led2)
{
        digitalWrite(LED2, LOW);
        timestamp2 = zeit;
        ledstate2 = OFF;
    }

    if (zeit > (RUNTIME2 + DELAY)) {
        running_led2 = 0;
        digitalWrite(LED2, LOW);
    }

// LED3
      if (zeit - timestamp3 > FREQUENCY1_OFF && ledstate3 == OFF &&
running_led3) {
        digitalWrite(LED3, HIGH);
        timestamp3 = zeit;
        ledstate3 = ON;
   }
```

```
    if (zeit - timestamp3 > FREQUENCY1_ON && ledstate3 == ON && running_led3)
{
        digitalWrite(LED3, LOW);
        timestamp3 = zeit;
        ledstate3 = OFF;
    }

   if (zeit > (RUNTIME3 + 2*DELAY)) {
        running_led3 = 0;
        digitalWrite(LED3, LOW);
   }

// LED4
        // LED4 is off and is switched on after frequency off time is over,
can only be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp4 > FREQUENCY1_OFF && ledstate4 == OFF &&
running_led4) {
        digitalWrite(LED4, HIGH);
        timestamp4 = zeit;
        ledstate4 = ON;
    }

    // LED4 is on and is switched off, after frequency on time ist over
    if (zeit - timestamp4 > FREQUENCY1_ON && ledstate4 == ON && running_led4)
{
        digitalWrite(LED4, LOW);
        timestamp4 = zeit;
        ledstate4 = OFF;
    }

   // when RUNTIME is over turn LED off and - by setting variable running_led
to 0 - stop if loops that turn LEDs on and off
    if (zeit > RUNTIME4 + 3*DELAY) {
        running_led4 = 0;
        digitalWrite(LED4, LOW);
        }

 // LED5
        // LED5 is off and is switched on after frequency off time is over,
can only be truned on if was OFF before AND RUNTIME is not reached
```

```
      if (zeit - timestamp5 > FREQUENCY3_OFF && ledstate5 == OFF &&
running_led5) {
        digitalWrite(LED5, HIGH);
        timestamp5 = zeit;
        ledstate5 = ON;
    }

    // LED5 is on and is switched off, after frequency on time ist over
    if (zeit - timestamp5 > FREQUENCY3_ON && ledstate5 == ON && running_led5)
{
        digitalWrite(LED5, LOW);
        timestamp5 = zeit;
        ledstate5 = OFF;
    }

    if (zeit > RUNTIME5 + 4*DELAY) {// when RUNTIME is over turn LED off and
- by setting variable running_led to 0 - stop if loops that turn LEDs on and
off
        running_led5 = 0;
        digitalWrite(LED5, LOW);
        }

// LED6
        // LED6 is off and is switched on after frequency off time is over,
can only be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp6 > FREQUENCY1_OFF && ledstate6 == OFF &&
running_led6) {
        digitalWrite(LED6, HIGH);
        timestamp6 = zeit;
        ledstate6 = ON;
    }

    // LED6 is on and is switched off, after frequency on time ist over
    if (zeit - timestamp6 > FREQUENCY1_ON && ledstate6 == ON && running_led6)
{
        digitalWrite(LED6, LOW);
        timestamp6 = zeit;
        ledstate6 = OFF;
    }
```

```
    if (zeit > RUNTIME1 + 5*DELAY) {// when RUNTIME is over turn LED off and
- by setting variable running_led to 0 - stop if loops that turn LEDs on and
off
        running_led6 = 0;
        digitalWrite(LED6, LOW);
    delay(10);
    }

// LED7 is off and is switched on after frequency off time is over, can only
be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp7 > FREQUENCY1_OFF && ledstate7 == OFF &&
running_led7) {
        digitalWrite(LED7, HIGH);
        timestamp7 = zeit;
        ledstate7 = ON;
    }

// LED7 is on and is switched off, after frequency on time ist over
    if (zeit - timestamp7 > FREQUENCY1_ON && ledstate7 == ON && running_led7)
{
        digitalWrite(LED7, LOW);
        timestamp7 = zeit;
        ledstate7 = OFF;
    }
    // when RUNTIME is over turn LED off and - by setting variable running_led
to 0 - stop if loops that turn LEDs on and off
    if (zeit > RUNTIME2 + 6*DELAY) {
        running_led7 = 0;
        digitalWrite(LED7, LOW);
    }


// LED8
         if (zeit - timestamp8 > FREQUENCY1_OFF && ledstate8 == OFF &&
running_led8) {
       digitalWrite(LED8, HIGH);
        timestamp8 = zeit;
        ledstate8 = ON;
    }
```

```
   if (zeit - timestamp8 > FREQUENCY1_ON && ledstate8 == ON && running_led8)
{
        digitalWrite(LED8, LOW);
        timestamp8 = zeit;
        ledstate8 = OFF;
    }

    if (zeit > (RUNTIME3 + 7*DELAY)) {
        running_led8 = 0;
        digitalWrite(LED8, LOW);
    }

// LED9
      if (zeit - timestamp9 > FREQUENCY1_OFF && ledstate9 == OFF &&
running_led9) {
        digitalWrite(LED9, HIGH);
        timestamp9 = zeit;
        ledstate9 = ON;
   }

    if (zeit - timestamp9 > FREQUENCY1_ON && ledstate9 == ON && running_led9)
{
        digitalWrite(LED9, LOW);
        timestamp9 = zeit;
       ledstate9 = OFF;
    }

   if (zeit > (RUNTIME4 + 8*DELAY)) {
        running_led9 = 0;
        digitalWrite(LED9, LOW);
   }

// LED10
        // LED10 is off and is switched on after frequency off time is over,
can only be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp10 > FREQUENCY1_OFF && ledstate10 == OFF &&
running_led10) {
        digitalWrite(LED10, HIGH);
        timestamp10 = zeit;
        ledstate10 = ON;
    }
```

```
    // LED10 is on and is switched off, after frequency on time ist over
      if (zeit - timestamp10 > FREQUENCY1_ON && ledstate10 == ON &&
running_led10) {
        digitalWrite(LED10, LOW);
        timestamp10 = zeit;
        ledstate10 = OFF;
    }

   // when RUNTIME is over turn LED off and - by setting variable running_led
to 0 - stop if loops that turn LEDs on and off
    if (zeit > RUNTIME5 + 9*DELAY) {
        running_led10 = 0;
        digitalWrite(LED10, LOW);
        }

 // LED11
        // LED11 is off and is switched on after frequency off time is over,
can only be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp11 > FREQUENCY1_OFF && ledstate11 == OFF &&
running_led11) {
        digitalWrite(LED11, HIGH);
        timestamp11 = zeit;
        ledstate11 = ON;
    }

    // LED11 is on and is switched off, after frequency on time ist over
      if (zeit - timestamp11 > FREQUENCY1_ON && ledstate11 == ON &&
running_led11) {
        digitalWrite(LED11, LOW);
        timestamp11 = zeit;
        ledstate11 = OFF;
    }

    if (zeit > RUNTIME6 + 10*DELAY) {// when RUNTIME is over turn LED off
and - by setting variable running_led to 0 - stop if loops that turn LEDs on
and off
        running_led11 = 0;
        digitalWrite(LED11, LOW);
        }
```

```
// LED12
        // LED12 is off and is switched on after frequency off time is over,
can only be truned on if was OFF before AND RUNTIME is not reached
      if (zeit - timestamp12 > FREQUENCY1_OFF && ledstate12 == OFF &&
running_led12) {
        digitalWrite(LED12, HIGH);
        timestamp12 = zeit;
        ledstate12 = ON;
    }

    // LED12 is on and is switched off, after frequency on time ist over
      if (zeit - timestamp12 > FREQUENCY1_ON && ledstate12 == ON &&
running_led12) {
        digitalWrite(LED12, LOW);
        timestamp12 = zeit;
        ledstate12 = OFF;
    }

    if (zeit > RUNTIME6 + 11*DELAY) {// when RUNTIME is over turn LED off
and - by setting variable running_led to 0 - stop if loops that turn LEDs on
and off
        running_led12 = 0;
        digitalWrite(LED12, LOW);

    }
     delay(10);
}
```

End of Code