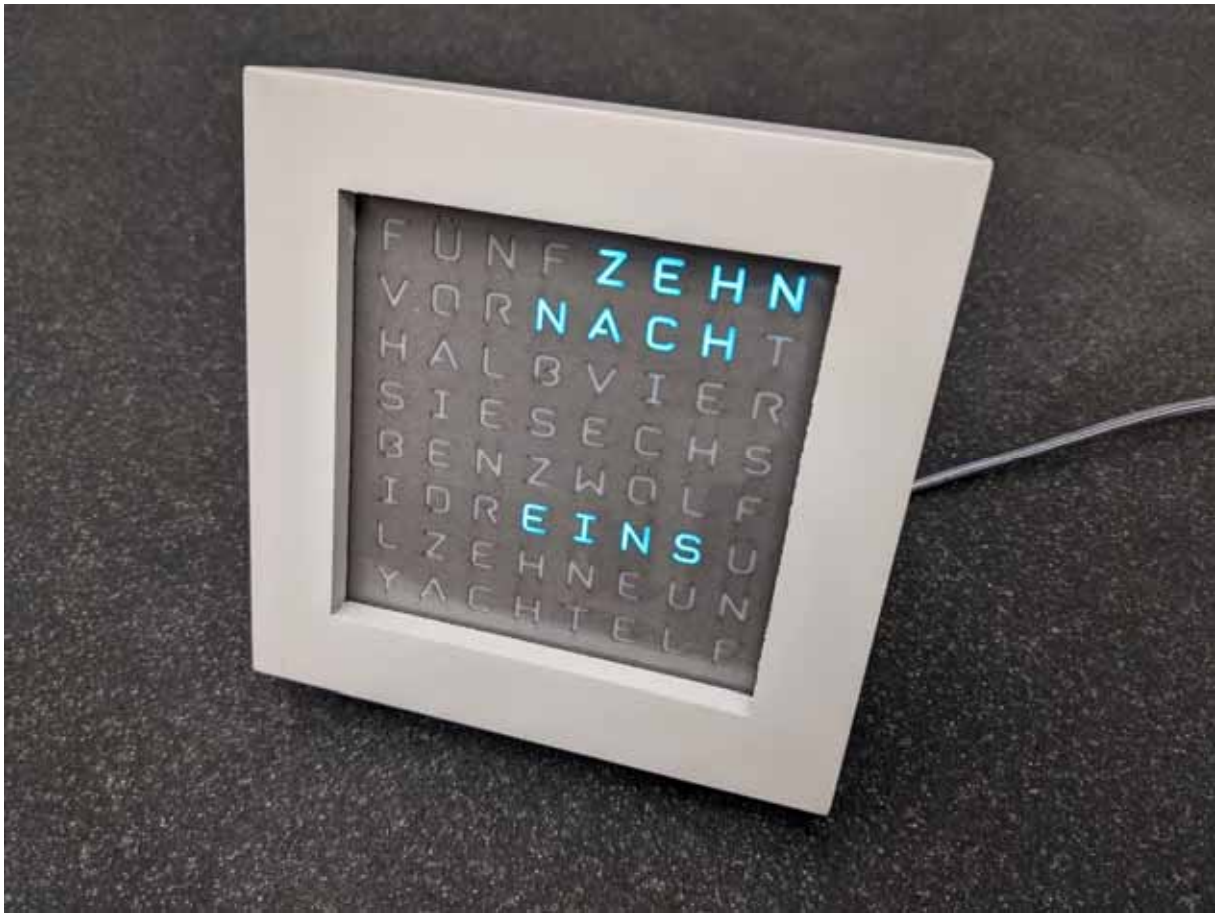
Klartextuhr – Projektdokumentation

Vorwort:
Während des ersten Corona-Lockdowns habe ich mich der Arduino-Programmierung angenähert. Mein erstes großes Projekt war eine Klartextuhr die mich täglich erfreut, aber auch sehr viel Arbeit gemacht hat. Ich wollte noch eine Uhr für den Schreibtisch haben, die günstig und schnell zu fertigen ist.



Stückliste:

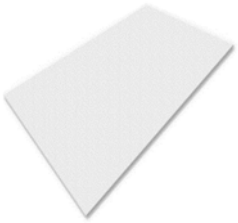
1. D1 mini ESP8266 Mikrocontroller

Der Mikrokontroller hat WLAN onboard und bezieht die Uhrzeit über das Internet. Folglich sind keine Bedientasten zum Stellen der Uhr notwendig.

2. WS2812-LED-Matrix 80 x 80 mm

Auf der RGB-LED-Matrix sind 64 adressierbare LEDs verdrahtet. Es werden nur drei Anschlussdrähte benötigt.

3. PVC-hart Leuchtkastenfolie, transluzent 100 x 100 x 0,3 mm



4. lasergeschnittene Front
5. lasergeschnittene Abschattung

Fotokarton 300g/m²

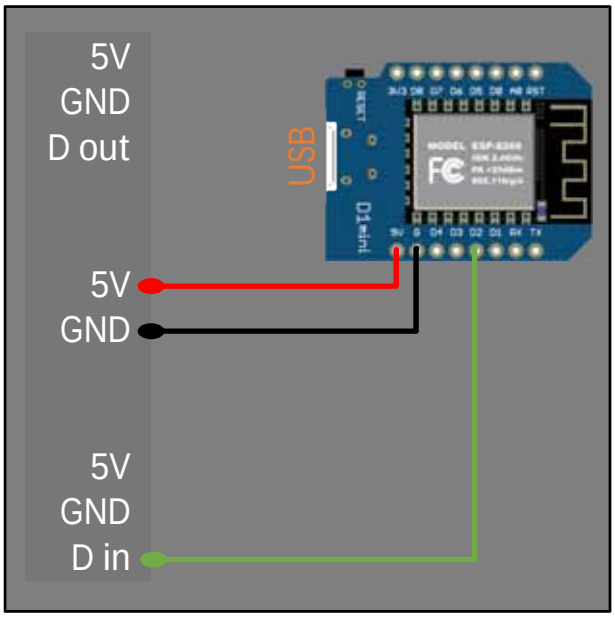
6. Gehäuse / Bilderrahmen 120 x 120 x 33 mm

7. Handyladegerät (5V) mit A Buchse
8. USB-Kabel, A Stecker auf Micro B Stecker
9. Schaumstoff 80 x 80 x 10 mm

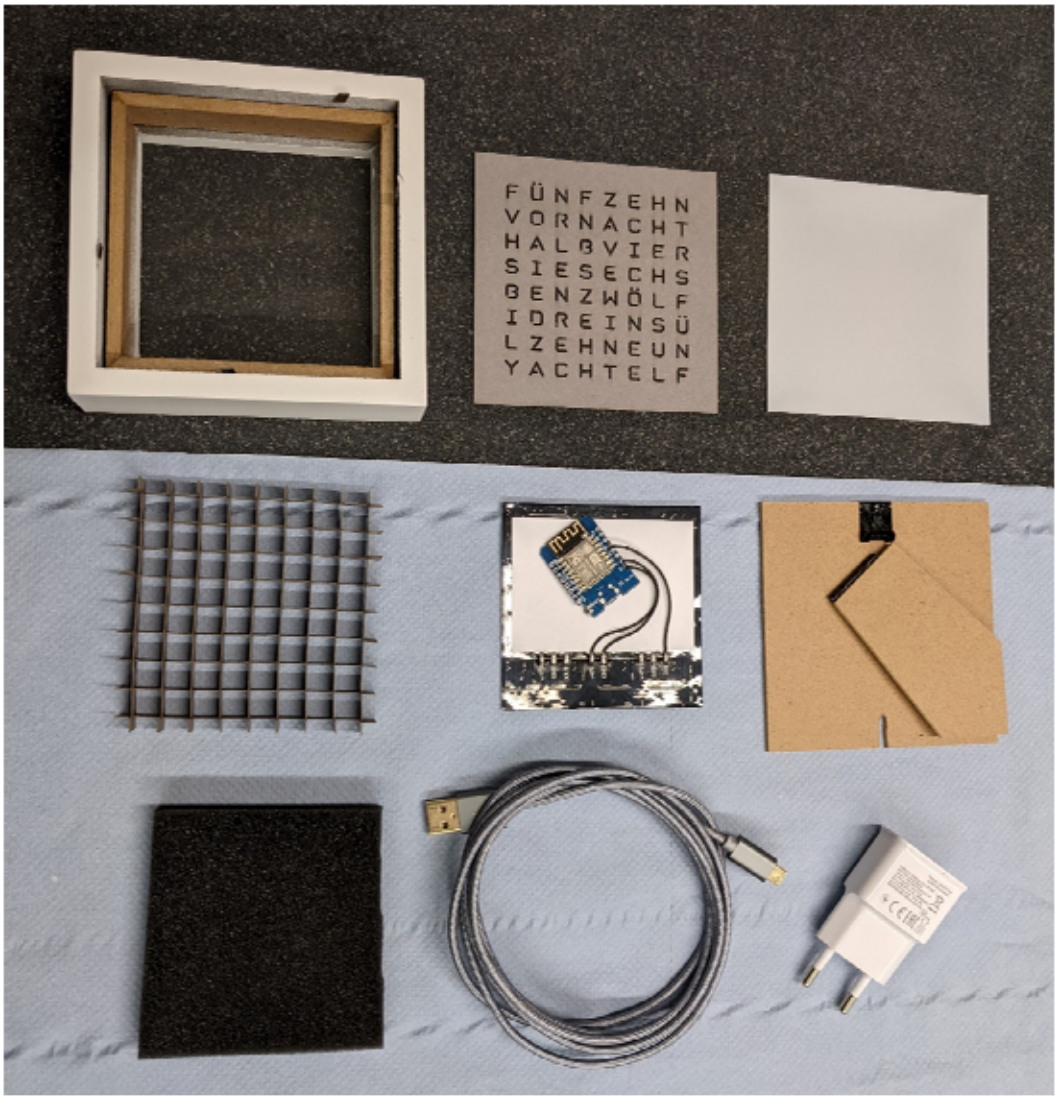
Zuordnung der Buchstaben zu den LED-Adressen:

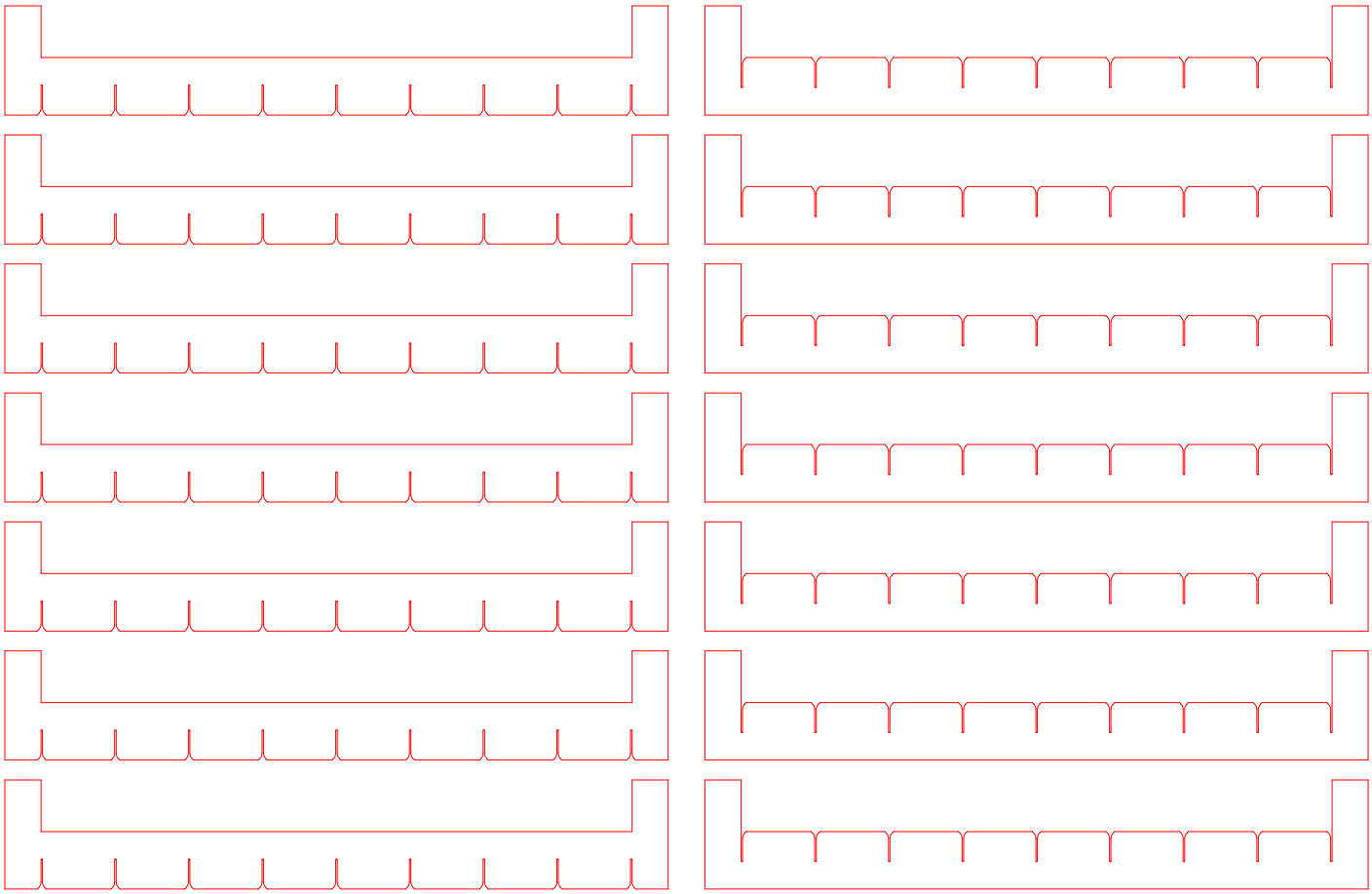
F *63*	**Ü** *62*	**N** *61*	**F** *60*	**Z** *59*	**E** *58*	**H** *57*	**N** *56*
V *48*	**O** *49*	**R** *50*	**N** *51*	**A** *52*	**C** *53*	**H** *54*	**T** *55*
H *47*	**A** *46*	**L** *45*	**B** *44*	**V** *43*	**I** *42*	**E** *41*	**R** *40*
S *32*	**I** *33*	**E** *34*	**S** *35*	**E** *36*	**C** *37*	**H** *38*	**S** *39*
B *31*	**E** *30*	**N** *29*	**Z** *28*	**W** *27*	**Ö** *26*	**L** *25*	**F** *24*
I *16*	**D** *17*	**R** *18*	**E** *19*	**I** *20*	**N** *21*	**S** *22*	**Ü** *23*
L *15*	**Z** *14*	**E** *13*	**H** *12*	**N** *11*	**E** *10*	**U** *9*	**N** *8*
Y *0*	**A** *1*	**C** *2*	**H** *3*	**T** *4*	**E** *5*	**L** *6*	**F** *7*

Verdrahtung:

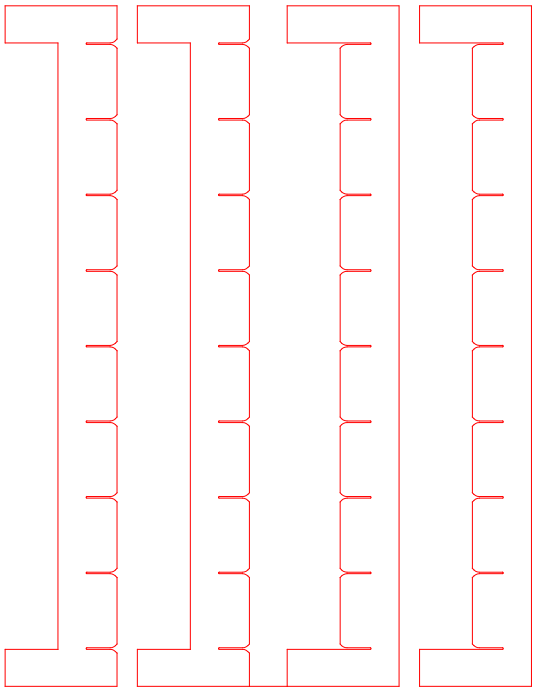


Zusammenbau:





FÜNFZEHN
VORNACHT
HALBVIER
SIESECHS
BENZWÖLF
IDREINSÜ
LZEHNEUN
YACHTELF



Code:

```
/* Wordclock mit 8x8-Matrix, Zeitinfo übers Wlan
  aufgesetzt auf NTP_Example.ino
  D1 mini mit Matrix WS2812 Neopixel
  Board LOLIN(WEMOS) D1 R2 & mini
  Boardverwaltung esp8266 Version 3.0.2
  Arduino IDE 1.8.15                    */

// Intro Wifi und NTP
#include <ESP8266WiFi.h>
#include <time.h>
const char* ssid      = "WL_name";     // z.B. FritzBox4711;
const char* password  = "password";    // z.B. Clock_007;
const char* NTP_SERVER = "ptbtime1.ptb.de";
const char* TZ_INFO    = "CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00";

// enter your time zone (https://remotemonitoringsystems.ca/time-zone-abbreviations.php)
tm timeinfo;
time_t now;
long unsigned lastNTPtime;
unsigned long lastEntryTime;

// Intro Zeitberechnungberechnung und Korrektur
long time_sec = 0; // Uhrzeit in Sekunden
long Stunde_korr  = 0;
long Minute_korr  = 0;
long Sekunde_korr = 0;

// Intro Wordclock
byte wch = 0; // Wordclockstunden
byte wcm = 0; // Wordclockminuten
byte old_wcm = 0; // 5min Intervall
int pause = 50;

// Intro Neopixel
byte rt = 0; // Helligkeit rot
byte gn = 0; // Helligkeit grün
byte bl = 0; // Helligkeit blau
#include <Adafruit_NeoPixel.h>  // Version 1.8.5
#define PIN 4 //PIN D2 GPIO-4; Daten für WS2812 LEDs bzw. NeoPixel
#define NUMPIXELS 64 // Anzahl der LEDs 8x8 ohne Opferpixel(Pegelshifter)
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(115200);
  pixels.begin(); // Initialisierung der NeoPixel
  off();
  alle(rt, gn, bl);
  Serial.println("\n\nNTP Time Test\n");
  WiFi.begin(ssid, password);

  int counter = 0;
  while (WiFi.status() != WL_CONNECTED) {
    delay(200);
    if (++counter > 100) ESP.restart();
    Serial.print ( "." );
    rot();                      // blinkendes Y
```

```
    antenne(rt, gn, bl);
    delay(250);
    off();
    antenne(rt, gn, bl);
    //alle(rt, gn, bl);
    delay(250);
  }
  Serial.println("\n\nWiFi connected\n\n");

  configTime(0, 0, NTP_SERVER);

  setenv("TZ", TZ_INFO, 1);

  if (getNTPtime(10)) {  // wait up to 10sec to sync
  } else {
    Serial.println("Time not set");
    ESP.restart();
  }
  lastNTPtime = time(&now);
  lastEntryTime = millis();
}

void loop() {
  getNTPtime(10);
  KorrTime(timeinfo);
  DisplayWC();
}

bool getNTPtime(int sec) {
  {
    uint32_t start = millis();
    do {
      time(&now);
      localtime_r(&now, &timeinfo);
      delay(10);
    } while (((millis() - start) <= (1000 * sec)) && (timeinfo.tm_year < (2016 - 1900)));
    if (timeinfo.tm_year <= (2016 - 1900)) return false;  // the NTP call was not successful
    char time_output[30];
    //strftime(time_output, 30, "%a  %d-%m-%y %T", localtime(&now));
    strftime(time_output, 30, "%d-%m-%y %T", localtime(&now));
    Serial.println(time_output);
    delay(1000);
  }
  return true;
}

// Korrektur der Uhrzeit ********************************************
/* Wordclocks lösen die Zeit nur auf fünf Minuten auf.
   Der maximale Fehler liegt demzufolge bei maximal fünf Minuten.
   Um diesen Fehler zu minimieren müssen diese Uhren um zweieinhalb Minuten "vor" gehen (150 Sekunden)*/
void KorrTime(tm localTime) {
  time_sec = ((localTime.tm_hour * 60 * 60) + (localTime.tm_min * 60) + (localTime.tm_sec) + 150);

  Stunde_korr = (time_sec / 3600) % 24; // modulo %
  Minute_korr = (time_sec / 60) % 60;
  Sekunde_korr = time_sec % 60;

} // Ende ************************************************************
```

```
// Ansteuerung der Textfelder ****************************************
void DisplayWC() {

  wcm = Minute_korr / 5; // Wordclockminuten (Minuten durch fuenf)
  wch = Stunde_korr;

  //      ---> Aktualisierung nur alle 5 Minuten

  if (old_wcm != wcm) {
    old_wcm = wcm;
    off();
    alle(rt, gn, bl);

    textfarbe();

    // Ansteuerung der Textfelder

    switch (wcm)

    { case  0:
        break;

      case  1: fuenf_praefix(rt, gn, bl);
        nach(rt, gn, bl);
        break;

      case  2: zehn_praefix(rt, gn, bl);
        nach(rt, gn, bl);
        break;

      case  3: fuenf_praefix(rt, gn, bl);
        zehn_praefix(rt, gn, bl);
        nach(rt, gn, bl);
        break;

      case  4: zehn_praefix(rt, gn, bl);
        vor(rt, gn, bl);
        halb(rt, gn, bl);
        break;

      case  5: fuenf_praefix(rt, gn, bl);
        vor(rt, gn, bl);
        halb(rt, gn, bl);
        break;

      case  6: halb(rt, gn, bl);
        break;

      case  7: fuenf_praefix(rt, gn, bl);
        nach(rt, gn, bl);
        halb(rt, gn, bl);
        break;

      case  8: zehn_praefix(rt, gn, bl);
        nach(rt, gn, bl);
        halb(rt, gn, bl);
        break;
```

```
  case  9: fuenf_praefix(rt, gn, bl);
    zehn_praefix(rt, gn, bl);
    vor(rt, gn, bl);
    break;

  case 10: zehn_praefix(rt, gn, bl);
    vor(rt, gn, bl);
    break;

  case 11: fuenf_praefix(rt, gn, bl);
    vor(rt, gn, bl);
    break;
}

if (wcm >= 4) {
  wch = wch + 1;
}  // Bezug zur Folgestunde

if (wch > 12) {
  wch = wch - 12; // Umrechnung 24h/12h
}

switch (wch)

{ case  0: if (wcm == 0) {
      rot();
      nacht(rt, gn, bl);

      textfarbe();
    }
    else {
      zwoelf (rt, gn, bl);
    }
    break;

  case  1: eins(rt, gn, bl);
    break;

  case  2: zwei(rt, gn, bl);
    break;

  case  3: drei(rt, gn, bl);
    break;

  case  4: vier(rt, gn, bl);
    break;

  case  5: fuenf(rt, gn, bl);
    break;

  case  6: sechs(rt, gn, bl);
    break;

  case  7: sieben(rt, gn, bl);
    break;

  case  8: acht(rt, gn, bl);
```

```
      break;

    case  9: neun(rt, gn, bl);
      break;

    case 10: zehn(rt, gn, bl);
      break;

    case 11: elf(rt, gn, bl);
      break;

    case 12: zwoelf(rt, gn, bl);
      break;
   }

   Serial.println("");
 }
}
// Ende ***************************************************************

// Farben *************************************************************
void textfarbe()  {
  rt =   0, gn =  30, bl =  20; // 25 entspricht ca 1mA
}
void rot()    {
  rt =  50, gn =   0, bl =   0;
}
void gelb()   {
  rt =  25, gn =  25, bl =   0;
}
void gruen()  {
  rt =   0, gn =  50, bl =   0;
}
void cyan()   {
  rt =   0, gn =  25, bl =  25;
}
void blau()   {
  rt =   0, gn =   0, bl =  50;
}
void magenta(){
  rt =  25, gn =   0, bl =  25;
}
void weiss()  {
  rt =  17, gn =  17, bl =  17;
}
void off()    {
  rt =   0, gn =   0, bl =   0;
} // Ende ***************************************************************

// Ansteuerung des Feldes "FÜNF" (präfix) *****************************
void fuenf_praefix (byte rt , byte gn, byte bl)
{
  for (int i = 63 ; i >= 60; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
```

```
  Serial.print("FÜNF ");
} // Ende ************************************************************

// Ansteuerung des Feldes "ZEHN" (präfix) ****************************
void zehn_praefix (byte rt , byte gn, byte bl)
{
  for (int i = 59 ; i >= 56; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("ZEHN ");
} // Ende ************************************************************

// Ansteuerung des Feldes "VOR" **************************************
void vor (byte rt , byte gn, byte bl)
{
  for (int i = 48 ; i <= 50; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("VOR ");
} // Ende ************************************************************

// Ansteuerung des Feldes "NACH" *************************************
void nach (byte rt , byte gn, byte bl)
{
  for (int i = 51 ; i <= 54; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("NACH ");
} // Ende ************************************************************

// Ansteuerung des Feldes "HALB" *************************************
void halb (byte rt , byte gn, byte bl)
{
  for (int i = 47 ; i >= 44; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("HALB ");
} // Ende ************************************************************

// Ansteuerung des Feldes "EINS" *************************************
void eins (byte rt , byte gn, byte bl)
{
  for (int i = 19 ; i <= 22; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
```

```
   delay(pause);
  }
  Serial.println("EINS");
} // Ende ************************************************************

// Ansteuerung des Feldes "ZWEI" ***************************************
void zwei (byte rt , byte gn, byte bl)
{
  for (int i = 28 ; i >= 27; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  for (int i = 19 ; i <= 20; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("ZWEI");
} // Ende ************************************************************

// Ansteuerung des Feldes "DREI" ***************************************
void drei (byte rt , byte gn, byte bl)
{
  for (int i = 17 ; i <= 20; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("DREI");
} // Ende ************************************************************

// Ansteuerung des Feldes "VIER" ***************************************
void vier (byte rt , byte gn, byte bl)
{
  for (int i = 43 ; i >= 40; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("VIER");
} // Ende ************************************************************

// Ansteuerung des Feldes "FÜNF" ***************************************
void fuenf (byte rt , byte gn, byte bl)
{
  for (int i = 24 ; i >= 23; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  for (int i = 8 ; i >= 7; i--)
  {
```

```
   pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
   pixels.show(); // Durchführen der Pixel-Ansteuerung
   delay(pause);
  }
  Serial.print("FÜNF");
} // Ende ************************************************************

// Ansteuerung des Feldes "SECHS" ************************************
void sechs (byte rt , byte gn, byte bl)
{
  for (int i = 35 ; i <= 39; i++)
  {
   pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
   pixels.show(); // Durchführen der Pixel-Ansteuerung
   delay(pause);
  }
  Serial.print("SECHS");
} // Ende ************************************************************

// Ansteuerung des Feldes "SIEBEN" ***********************************
void sieben (byte rt , byte gn, byte bl)
{
  for (int i = 32 ; i <= 34; i++)
  {
   pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
   pixels.show(); // Durchführen der Pixel-Ansteuerung
   delay(pause);
  }
  for (int i = 31 ; i >= 29; i--)
  {
   pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
   pixels.show(); // Durchführen der Pixel-Ansteuerung
   delay(pause);
  }
  Serial.print("SIEBEN");
} // Ende ************************************************************

// Ansteuerung des Feldes "ACHT" *************************************
void acht (byte rt , byte gn, byte bl)
{
  for (int i = 1 ; i <= 4; i++)
  {
   pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
   pixels.show(); // Durchführen der Pixel-Ansteuerung
   delay(pause);
  }
  Serial.print("ACHT");
} // Ende ************************************************************

// Ansteuerung des Feldes "NEUN" *************************************
void neun (byte rt , byte gn, byte bl)
{
  for (int i = 11 ; i >= 8; i--)
  {
   pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
   pixels.show(); // Durchführen der Pixel-Ansteuerung
   delay(pause);
  }
```

```
  Serial.print("NEUN");
} // Ende **************************************************************

// Ansteuerung des Feldes "ZEHN" ***************************************
void zehn (byte rt , byte gn, byte bl)
{
  for (int i = 14 ; i >= 11; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("ZEHN");
} // Ende **************************************************************

// Ansteuerung des Feldes "ELF" ****************************************
void elf (byte rt , byte gn, byte bl)
{
  for (int i = 5 ; i <= 7; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("ELF");
} // Ende **************************************************************

// Ansteuerung des Feldes "ZWÖLF" **************************************
void zwoelf (byte rt , byte gn, byte bl)
{
  for (int i = 28 ; i >= 24; i--)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("ZWÖLF");
} // Ende **************************************************************

// Ansteuerung des Feldes "NACHT" **************************************
void nacht (byte rt , byte gn, byte bl)
{
  for (int i = 51 ; i <= 55; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
    delay(pause);
  }
  Serial.print("NACHT");
} // Ende **************************************************************

// Ansteuerung des Feldes "ANTENNE" ************************************
void antenne (byte rt , byte gn, byte bl)
{
  pixels.setPixelColor(0, pixels.Color(rt, gn, bl));
  pixels.show(); // Durchführen der Pixel-Ansteuerung

} // Ende **************************************************************
```

```
// Ansteuerung aller Pixel ******************************************
void alle (byte rt , byte gn, byte bl)
{
  for (int i = 0; i <= 63; i++)
  {
    pixels.setPixelColor(i, pixels.Color(rt, gn, bl));
    pixels.show(); // Durchführen der Pixel-Ansteuerung
  }
} // Ende ************************************************************
```