

ARDUINO - Wald und Wiesen-Anwendungen

[Software](#) -> [ARDUINO - Wald und Wiesen-Anwendungen](#)

gaulois

#1/13 Verfasst am: 03 Okt 2018 12:45 Titel: ARDUINO - Wald und Wiesen-Anwendungen

Hallo zusammen,
um das Thema abzurunden: Es gibt ja auch noch die Wald und Wiesen-Anwendungen, wo nix kompliziertes eingelesen werden muss, um was anderes kompliziertes zu steuern, z. B. eine "einfache" Lichtsteuerung, auch wenn diese durchaus komplizierte Züge annehmen kann.

Für meine erste Häuserzeile am Sassenstein habe ich z. B. eine Lichtfunktion geschrieben, die extern über zwei Schalter in zwei Testmodul und im Tag- und Nachtrhythmus extern gesteuert werden kann, aber intern auch einen Zeitablauf hat, der ermöglicht, dass neben den nach zufällig langen Zeitintervallen zufällig eingeschalteten Lampen in verschiedenen Räumen auch Ablaufsteuerungen verschiedener Vorgänge logisch eingesetzt werden können (Wolli B. haut mitten in der Nacht aufs Schlagzeug und alle Nachbarn werden wach) oder Fernseher und ähnliches durch Flackern dargestellt werden können.

Hierzu poste ich einfach mal den Steuerungscode, der sicher modifiziert auch für andere Anwendungen eingesetzt werden kann (die anderen Häuserzeilen wollen ja irgendwann auch mal beleuchtet werden und da gibt es bestimmt auch andere Abläufe, für die man aber das Grundgerüst nehmen kann:

```
const char TagNacht = A6;
const char forceButton = A7;    // soll alle Lampen zum Leuchten bringen (force)

const int Moba = 19; // Opa mit Enkel an der Modellbahn           Haus 1 links oben
const int Jugend = 0; // das Pärchen, das von ihrer Mutter gestört wird, Haus 1 rechts oben
const int Schreibtisch = 7; // Schreibtischtäter,                Haus 1 links unten
const int Kueche1 = 1; // der Mann, der auf sein Essen wartet,    Haus 1 rechts unten (darunter)
const int BettPaerchen = 15; // Pärchen kommt ins Schlafzimmer, Haus 2 links oben
const int Schaeetze = 2; // der Mann mit den Truhen und dem Gold, Haus 2 rechts oben
const int Kueche2 = 12; // Frau in Küche mit Holzofen,           Haus 2 links unten
const int Radio = 13; // Mann in der Stube mit Röhrenradio,       Haus 2 rechts unten
const int Wolli = 8; // Schlagzeuger Wolli Boecker,              Haus 3 links oben
const int Betthocker = 18; // Wollis direkter Nachbar entnervt auf Bett Haus 3 mittig oben
const int Frisieren = 17; // Frau beim Frisieren,                Haus 3 rechts oben
const int Fensterfrau = 3; // Die Frau am Fenster,               Haus 3 links unten
const int Bibliothek = 9; // Mann mit Buch, Frau sauer,          Haus 3 mittig unten
const int Kueche3 = 14; // Frau in Küche,                         Haus 3 rechts unten
const int Tanzpaar = 4; // tanzende Frauen,                      Haus 4 links oben
const int Fensterputz = 10; // Frau am Fenster,                  Haus 4 mittig oben
const int Schatz = 5; // alter Mann mit Stock und einer Truhe,    Haus 4 rechts oben
const int Kueche4 = 16; // Frau in Küche,                         Haus 4 links unten
const int Skat = 6; // Skatrunde,                                 Haus 4 mittig unten
const int Vorrat = 11; // Frau in der Vorratskammer,              Haus 4 rechts unten

int forceLength = 5000;    // Leuchtdauer im force-Modus - 1 Zyklus
int forceDelayAus = 200;    // Delay beim Ausschalten im force-Modus
int forceMode; // > 128 für force-Modus, <128> 128 = Nacht, < 128 = Tag (keine Aktivität, Licht ausschalten)

char writeValue; // HIGH oder LOW

void setup()
{
  pinMode(forceButton, INPUT);
  for (int i=1; i<14; i++) {pinMode(i-1, OUTPUT); }
  for (int i=1; i<7; i++) {pinMode(ledAnalog[i-1], OUTPUT); }
  randomSeed(millis());
```

```

UhrZeit = 1;
Durchlauf = 1;
WolliHatGespielt = false;
}

void loop() {

// UMBAU - STRATEGIE
//
// ReadInput
// defineParams
// if forceModeAndNight
// elseif forceModeAndDay
// elseif nonForceMode and Night
//   if Wolli / if Ereignis - man könnte Wolli auch als Input nehmen,
//   dann müsste die Stromversorgung über den Knopfdrücker und eine externe Schaltung funktionieren,
//   der die anderen Fenster steuert,
//   dann hätte man ein forceWolli, aber man hätte erst mal keinen Zufallsgenerator
//   Zufallsgenerator für Wolli könnte über ein anderes Fenster erfolgen (dann ein Fenster weniger? Oder Verzögerung?
//   Oder Wechsel von An auf Aus?)
// else (nonForceMode and Day)
//

// for (int i=1; i<100; i++)
// {
//   forceMode = analogRead(forceButton);
//   TagNachtVal = analogRead(TagNacht);
// }
// if (forceMode < 128) // forceButton nicht gedrückt - normaler Modus
// {

// *****
// TESTBEREICH - EINGABE FESTER WERTE ZUM AUSPROBIEREN
//
//   randNr = random(14, 17);
//   randNr = random(led1, led4);
//
//   randOnOff = 1;
//   TagNachtVal = 255;
//   writeValue = HIGH;
//   randNr = random(3, 13);
//
//   digitalWrite(13, digitalRead(TagNacht));
//   digitalWrite(12, writeValue);
// ENDE DES TESTBEREICHS
// *****

  randOnOff = random(1, 4);
  randNr = random (ledDigital[0], ledMaxPinAnalog);
  if (TagNachtVal <128> 128 and (UhrZeit <AbendDaemmerung>= MorgenDaemmerung) and (randOnOff == 1 or
  randOnOff == 2) and (UhrZeit <SonnenAufGang> EINSCHALTEN erlauben
  {
    writeValue = HIGH;
  }

  if (randNr == ledDigital[3] or randNr == ledDigital[5] or randNr == ledDigital[7] or randNr == ledDigital[9] or randNr ==
  ledDigital[10] or randNr == ledDigital[11])
    // ein PWM-fähiger Pin
    {
      if (ledOnOff[randNr] == 0 and writeValue == HIGH) // der Pin ist ausgeschaltet und soll eingeschaltet werden =>
      fade in
      {
        for(int fadeValue = 0; fadeValue <fadeMax> ausschalten oder den nächsten Pin einschalten

```

```

{
  if (writeValue == LOW and ledOnOff[randNr] == 1) // ausschalten
  {
    for(int fadeValue = fadeMax; fadeValue >0; fadeValue -=fadeStep)
    {
      analogWrite(randNr, fadeValue);
      delay(fadeDelay);
      UhrZeit = UhrZeit + fadeDelay;
    }
    ledOnOff[randNr] = 0;
  }
  else // es soll eingeschaltet werden, aber die LED ist schon an => nächste LED einschalten, wirklich nur
  einschalten, nicht ausschalten
  {
    if (writeValue == HIGH)
    {
      randNr = randNr++;
      if (randNr > ledMaxPinAnalog) {randNr = ledDigital[0];}
      digitalWrite(randNr, writeValue);
    }
  }
}
else // kein PWM-fähiger Pin
{
  if (writeValue == HIGH and ledOnOff[randNr] == 1) // die LED, die eingeschaltet werden soll, ist schon an =>
  nächste LED einschalten mit Überlauf
  {
    randNr = randNr++;
    if (randNr > ledMaxPinAnalog) {randNr = ledDigital[0];}
    digitalWrite(randNr, writeValue);
    if (UhrZeit <AbendDaemmerung> MorgenDaemmerung) // morgens wird schnell ausgeschaltet
    {
      randDelay = random((Minute + Durchlauf) * minMinutenSchnell, (Minute + Durchlauf) * maxMinutenSchnell);
    }
    else // abends wird eher langsam ausgeschaltet
    {
      randDelay = random((Minute + Durchlauf) * minMinutenLangsam, (Minute + Durchlauf) * maxMinutenLangsam);
    }
  }
  //
}
}
// randDelay = random(1000, 10000);
delay(randDelay);

if (Durchlauf <= maxDurchlauf) // feste Anzahl Schaltzyklen, unabhängig vom externen Signal
{
  Durchlauf = Durchlauf++;
}
else // Ende der erlaubten Schaltzyklen ist erreicht
{
  Durchlauf = 1;
  WolliHatGespielt = false;
  for(int i=ledDigital[0]; i<ledMaxPinAnalog+1; i++) // alles ausschalten mit Delay
  {
    digitalWrite(i, LOW);
    ledOnOff[i-3]=0;
    delay (random((Minute + Durchlauf) * minMinutenSchnell, (Minute + Durchlauf) * maxMinutenLangsam));
  }
}

if (UhrZeit <= SonnenAufGang) // es ist noch Nacht (gemäß fester Uhrzeit), es können noch Lampen eingeschaltet
werden

```

```

{
  UhrZeit = UhrZeit + randDelay;
}
else // gemäß fester Uhrzeit ist jetzt SonnenAufGang und die Lampen werden ausgeschaltet
{
  UhrZeit = 1;
  WollHatGespielt = false;
  for(int i=ledDigital[0]+1; i<ledMaxPinAnalog+1; i++) // alles ausschalten mit Delay
  {
    digitalWrite(i, LOW);
    ledOnOff[i-1]=0;
    delay (random((Minute + Durchlauf) * minMinutenSchnell, (Minute + Durchlauf) * maxMinutenSchnell));
  }
}
}
}

```

else // forceButton gedrückt - alle Lampen an! - Testmodus

```

{
  if (TagNachtVal < 128) // kein Nachtmodus-Steuersignal von Extern (und forcebutton)
  {
    // TESTROUTINE FÜR EINFACHES LAUFLICHT

```

```

    for (int i=1; i<22; i++)
    {
      digitalWrite(i-1, HIGH);
      delay(forceDelayAus);
    }
    delay (forceDelayAus);
    for (int i=1; i<22>0; i--)
    {
      digitalWrite(i-1, HIGH);
      delay(forceDelayAus);
    }
    delay (forceDelayAus);
    for (int i=21; i>0; i--)
    {
      digitalWrite(i-1, LOW);
      delay(forceDelayAus);
    }
    delay (forceDelayAus);

```

```

  }
  else // externes Nachtmodus-Steuersignal gesetzt (und forcebutton)
  {

```

```

    for(int i=ledDigital[0]; i<ledMaxPinAnalog+1; i++)
    {
      digitalWrite(i, HIGH);
      ledOnOff[i-3]=1;
    }
    delay (forceLength);
    for(int i=ledDigital[0]; i<ledMaxPinAnalog+1; i++)
    {
      digitalWrite(i, LOW);
      ledOnOff[i-3]=0;
    }
  }

```

```

  UhrZeit = 1;
  Durchlauf = 1;
  WollHatGespielt = false;
}

```

//// TESTROUTINE FUER LED-TEST - VOLLTEST LAUFLICHT DIGITALPINS

```

// for (int i=3; i<14; i++)
// {
//   digitalWrite(i, HIGH);
//   delay (200);

```

```

// }
//
// for (int i=3; i<14; i++)
// {
//   digitalWrite(i, LOW);
//   delay (200);
// }
//// TESTROUTINE FUER LED-TEST - VOLLTEST LAUFLICHT ANALOGPINS
// for (int i= 1; i<9; i++)
// {
//   digitalWrite(ledAnalog[i-1], HIGH);
//   delay (200);
// }
// for (int i= 1; i<9; i++)
// {
//   digitalWrite(ledAnalog[i-1], LOW);
//   delay (200);
// }
}

```

Ich hoffe, ich habe die richtige Version erwischt, ist schon ein paar Tage her, dass ich die geschrieben habe.

gaulois

#2/13 Verfasst am: 03 Okt 2018 17:39 Titel: Sassensteiner Lichtspiele 01

... scheinbar nicht 😊 die letzte Version war Version 7.4, die dann auch auf die nächste Häuserreihe modifiziert übertragen werden sollte. Hier der Code, der besser sein sollte als der oben. Leider kann ich ihn nicht als attachment posten, weil die Endung .ino in diesem Forum verboten ist...

```

// *****
//
// JAHRESZEITABHÄNGIG SOLLTE MORGENS GAR KEIN LICHT ANGEHEN (VOR DER ARBEIT IST ES IM
// SOMMER HELL, IM WINTER DUNKEL
//
// DIESE STEUERUNG IST AUF HERBST ODER FRÜHLING AUSGELEGT. IM SOMMER WÄRE DAS ANDERS
//
// Variante mit fester Zeiteinstellung, eine WOCHENTAGSTEUERUNG wäre die nächste Erweiterung KONZEPT:
// Montags: Späteres Licht-Einschalten morgens, früheres Licht-Ausschalten Abends
// Dienstags - Donnerstags normal
// Freitags: Späteres Licht-Ausschalten Abends
// Samstags: Späteres Licht-Einschalten bei einzelnen Personen, die samstags nicht arbeiten, späteres
// Licht-Ausschalten
// Sonntags: Späteres Licht-Einschalten bei allen, abends früheres Licht-Ausschalten (für den Montag), da spielt Wolli
// nicht!
//
// Variante mit Auslesen des Status der Pins und bei Einschalten bereits eingeschalteter LED nächste einschalten...
// BUGFIX: Ausschalt routine nur im non-force Modus, bei force-Taste bleiben alle LED an für 5 sec, dann wird erneut
// der Modus gelesen
// BUGFIX: nach Force-Modus werden trotz Tages-Pegel am Nachtschalter PWM-LEDs eingeschaltet, fixed
// BUGFIX: nach Force-Modus kein zweiter Force-Modus und kein Nachtmodus möglich, wenn Ausschalten im
// forceModus mit delay, sonst ja ...
//
// A5 und A6 können nicht als Ausgänge genutzt werden, aber als Eingänge, daher erfolgt die Tag-Nacht-Steuerung
// über A6
// und die Force-Steuerung über A7
// dafür einen Ausgang an D2 (bisher Eingang für die Tag-Nacht-Steuerung) und dann vielleicht über D0 und D1 auch
// Ausgänge
// das wären dann 20 Ausgänge und zwei Eingänge! PERFEKT!!! wenn möglich, D0 und D1 sind auch die serielle
// Schnittstelle (erledigt)
//
// Variante mit langsamerer Ausschaltzeit als Einschaltzeit (eigentlich muss die Einschaltzeit abends kurz sein, die
// Ausschaltzeit morgens auch (Dämmerung))
// die Ausschaltzeit gegen Mitternacht und die Einschaltzeit nach Mitternacht müssten länger sein, weil mehr vom Zufall

```

```

abhängig, nicht von der Dämmerung
// das wurde berücksichtigt
//
// UMBAU auf Funktionsaufrufe statt Spaghetti-Code - Version 7.0 mit allen bisherigen features und dem ZWANG ZUM
SPIELEN für WOLLI (Wolli-Mania)
// Einbau einer "Fernsehfunktion" bei der in jedem Durchlauf der Fernseher in der Helligkeit angepasst wird
// BUGFIX: Fernseher ändert sich nur, wenn auch ein anderes Fenster sich ändert ... sollte öfter sein, also
extra-Routinen einbauen ohne andere Fenster
// BUGFIX: SendeSchluss stimmt nicht, Fernseher-Erkennung in PWM-Routine stimmt nicht, Fernseher-Ausklammern in
ForceDown-Routine stimmt nicht.
// BUG: SONNTAGS kann auch tagsüber der Fernseher bei Opa Krämer laufen...
//
// Umbau auf CallFernseher, Erweiterung auf noch nie angewesene Fenster, die beim Forcedown für 200 - 400 ms
eingeschaltet werden.
// BUG: aus dem Tagmodus kommt das Programm nicht mehr raus...
//
// *****

const char TagNacht = A6;
const char forceButton = A7;      // soll alle Lampen zum Leuchten bringen (force)

const int Moba = 19; // Opa mit Enkel an der Modellbahn           Haus 1 links oben
const int Jugend = 0; // das Pärchen, das von ihrer Mutter gestört wird, Haus 1 rechts oben
const int Schreibtisch = 7; // Schreibtischtäter,                Haus 1 links unten PWM
const int Kueche1 = 1; // der Mann, der auf sein Essen wartet,    Haus 1 rechts unten (darunter)
const int BettPaerchen = 15; // Pärchen kommt ins Schlafzimmer, Haus 2 links oben
const int Schaetze = 2; // der Mann mit den Truhen und dem Gold, Haus 2 rechts oben
const int Kueche2 = 12; // Frau in Küche mit Holzofen,           Haus 2 links unten
const int Radio = 13; // Mann in der Stube mit Röhrenradio,       Haus 2 rechts unten
const int Wolli = 8; // Schlagzeuger Wolli Boecker,              Haus 3 links oben
const int Betthocker = 18; // Wollis direkter Nachbar entnervt auf Bett Haus 3 mittig oben
const int Frisieren = 17; // Frau beim Frisieren,                Haus 3 rechts oben
const int Fensterfrau = 3; // Die Frau am Fenster,               Haus 3 links unten PWM
const int Bibliothek = 9; // Mann mit Buch, Frau sauer,           Haus 3 mittig unten PWM
const int Kueche3 = 14; // Frau in Küche,                         Haus 3 rechts unten
const int Tanzpaar = 4; // tanzende Frauen,                      Haus 4 links oben
const int Fensterputz = 10; // Frau am Fenster,                  Haus 4 mittig oben PWM
const int Schatz = 5; // alter Mann mit Stock und einer Truhe,    Haus 4 rechts oben PWM
const int Kueche4 = 16; // Frau in Küche,                        Haus 4 links unten
const int Skat = 11; // Skatrunde,                                Haus 4 mittig unten PWM
const int Vorrat = 6; // Frau in der Vorratskammer,              Haus 4 rechts unten

//int ledDigital[14]={Jugend, Kueche1, Schaetze, Fensterfrau, Tanzpaar, Schatz, Skat, Schreibtisch, Wolli,
//
// Bibliothek, Fensterputz, Vorrat, Kueche2, Radio};
//int ledAnalog[6]={Kueche3, BettPaerchen, Kueche4, Frisieren, Betthocker, Moba};

const int Fernseher[6] = {Schatz, Fensterfrau, Fensterputz, Schreibtisch, Bibliothek, Skat}; // alle PWM-Pins eben
//const boolean FernseherAktiv[6] = {true, true, true, true, true, true}; //legt fest, wo ein Fernseher stehen darf
const boolean FernseherAktiv[6] = {true, true, true, true, true, true}; //legt fest, wo ein Fernseher stehen darf
const int FernsehEinstellung[6] = {4, 2, 1, 2, 3, 2}; //Helligkeitsfaktoren der Fernseher
int FernsehProgramm[6] = {0, 1, 2, 3, 4, 5}; //Zuordnung der Fernseher zu einem Programm
const int ProgrammMin = 30;
const int ProgrammMax = 64;
//boolean gleichesProgramm = false;
int Programm[6] = {ProgrammMax, ProgrammMax, ProgrammMin, ProgrammMin, ProgrammMin, ProgrammMin};
int ProgrammLaenge[6] = {1500, 2000, 1200, 3000, 1000, 2500};

//const int Fernseher1 = Schatz; // da soll der Fernseher stehen Haus 4 rechts oben PWM - FERNSEHER
//const int Fernseher2 = Fensterfrau; // da steht noch ein Fernseher Haus 3 links unten PWM
//const int Fernseher2 = Fensterputz; // da steht noch ein Fernseher Haus 4 mittig oben PWM
//const int Fernseher2 = Schreibtisch; // da steht noch ein Fernseher Haus 1 links unten PWM

int forceDelay = 5000; // Leuchtdauer im force-Modus - 1 Zyklus
int forceDelayAus = 200; // Delay beim Ausschalten im force-Modus
int forceLevel = 128; // > 128 für force-Modus, < 128 für normalen Betriebsmodus

```

```

const int ledMaxPinAnalog = 20; // maximal 20 Ausgaenge, davon 15 - 20 als A0 bis A5
const int ledMinPinDigital = 0;
int ledOnOff[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // Zustand der Pins
int ledWasOn[20]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // Pins die schon mal an waren

//const int fadeMax = 255; // MAXIMALE HELLIGKEIT AN PINS 3, 5, 7, 9, 10, 11
const int fadeStep = 1; // EIN- und AUSBLENDE-GESCHWINDIGKEIT als Stufe
const int fadeMax[6] = {127, 63, 31, 63, 255, 127}; // Helligkeit, wenn der Fernseher aus ist (Modus 1)

//int Verzoegerung = 15; // brauchbarer Wert
//int Verzoegerung = 20; // brauchbarer Wert
//int Verzoegerung = 17; // brauchbarer Wert
int Verzoegerung = 13;

const int Minute = 8; // Grundstock für den Verzögerungsfaktor mit den Durchläufen, müsste 22,5 sein... Wir
sind als 1/3 zu schnell für 15Minuten-Tage, wir haben 5Minuten-Tage
const int minMinutenSchnell = 10 * Minute; // Multiplikator für die kleinste Verzögerung zwischen zwei loops beim
Einschalten der Lampen, das müssten 225 ms sein, sind 230
const int maxMinutenSchnell = 30 * Minute; // Multiplikator für die größte Verzögerung zwischen zwei Loops beim
Einschalten der Lampen
const int minMinutenLangsam = 3 * minMinutenSchnell; // Multiplikator für die kleinste Verzögerung zwischen zwei loops
beim Ausschalten der Lampen
const int maxMinutenLangsam = 3 * maxMinutenSchnell; // Multiplikator für die größte Verzögerung zwischen zwei loops
beim Ausschalten der Lampen
const int AbendDaemmerung = 2400 * minMinutenSchnell; // Aufschaltzeit der LED in Millisekunden
const int SonnenAufGang = 6100 * minMinutenSchnell; // Beginn der Helligkeit, Ende der Nacht das müssten dann
450.000 ms sein sind 460.000
const int MorgenDaemmerung = SonnenAufGang - (2500 * minMinutenSchnell); // Ausschaltzeit der LED in
Millisekunden
const int fadeDelay = 12; // Delay für PWM De- oder Increment
int UhrZeit; // Summe der tatsächlich vergangenen Millisekunden
int Durchlauf; // Zähler für die Durchläufe
int Wochentag; // von 1 bis 7
const int SonnenUnterGang = 2 * SonnenAufGang; // Bei der Equinox ist das so, das müssten 900.000 ms sein, sind
920.000 (15 Minuten und 20 Sekunden)
const int AbendsFrueherAus[7] =
{240*minMinutenSchnell,120*minMinutenSchnell,120*minMinutenSchnell,120*minMinutenSchnell,0,0,360*minMinutenS
chnell}; // Faktor, um den das Licht abends früher aus geht
const int MorgensSpaeterAn[7] = {0,0,0,0,0,120*minMinutenSchnell,240*minMinutenSchnell}; // Faktor, um den das
Licht morgens später an geht
int FernsehEinschaltZeit = SonnenUnterGang; // wann der Fernseher[1] eingeschaltet wurde
const int FernsehLaufZeitMin = 20 * maxMinutenLangsam; // wie lange der Fernseher mindestens laufen muss
const int FernsehLaufZeitMax = 50 * maxMinutenLangsam; // wie lange der Fernseher maximal laufen darf
const int SendeBeginn = 1400 * minMinutenSchnell; // SONNTAGS WAR DER ABER FRÜHER ... bzw. kein
SendeSchluss vom letzten Tag ...
const int SendeSchluss = MorgenDaemmerung ; // früher gab es einen SendeSchluss
boolean warGeradeNacht; // zum Vergleich, ob sich das Datum geändert hat
boolean WolliHatGespielt; // Variable zur Steuerung von Wolli B., damit er nur einmal pro Nacht spielt

void setup()
{
  pinMode(forceButton, INPUT);
  for (int i=1; i<22; i++) {pinMode(i-1, OUTPUT); digitalWrite(i-1, LOW);}
  randomSeed(millis());
  UhrZeit = 1;
  Durchlauf = 1;
  WolliHatGespielt = false;
  Wochentag = 1;
  warGeradeNacht = true;
  delay(random(10 * minMinutenSchnell, 20 * maxMinutenLangsam));
  for (int i=1; i<7; i++)
  {
    FernsehProgramm[i] = random(1, i); // Programmzuordnung zu Fernsehern neu gestalten
  }
}

```

```

void loop() {
  ledOnOff[Fernseher[0]] = SendeZeit(SendeBeginn, SendeSchluss, UhrZeit, Fernseher[0], ledOnOff[Fernseher[0]]);
  // SENDEZEIT - PRÜFEN
  for (int i=2; i<7> 2 and not forceMode())
  {
    callFernseher(1, 100, 300);
    for (int i = 1; i <random> (MorgenDaemmerung - AbendsFrueherAus[Wochentag])) // KEINE SENDEZEIT
und Fernseher an
  {
    forceDown();
    for (int i=1; i<7; i++)
    {
      ledOnOff[Fernseher[i-1]] = 0;
    }
    callFernseher(1, 100, 300);
  }
  if (forceMode())
  {
    for (int i=1; i<7; i++)
    {
      ledOnOff[Fernseher[i-1]] = 2;
    }
    if (isNight())
    {
      sequenceLight();
    }
    else
    {
      allLightsOn();
    }
    UhrZeit = 1;
    Durchlauf = 1;
    WolliHatGespielt = false;
  }
  else // nonForceMode
  {
    if (Wochentag == 7 or Wochentag == 1)
    {
      WolliHatGespielt = true; // Sonntags und Montags spielt Wolli nämlich nicht
    }
    if (Wochentag == 5)
    {
      WolliHatGespielt = false; // Freitags spielt er dafür ums öfter, jedenfalls prinzipiell ...
    }
    if (isNight() and not warGeradeNacht)
    {
      Wochentag = newDay(Wochentag); // Datumswechsel
    }
    // Serial.println(Wochentag);
  }
  if (isNight())
  {
    // if (UhrZeit <MorgenDaemmerung> 20) {Verzoegerung = Verzoegerung - 8;}
    // if ((UhrZeit <AbendDaemmerung> MorgenDaemmerung + MorgensSpaeterAn[Wochentag-1] and not EinAus))
    // { // DIESE VERZÖGERUNG MACHT ÄRGER!!!
    //   Verzoegerung = random((Minute + Durchlauf) * minMinutenSchnell, (Minute + Durchlauf) * maxMinutenSchnell);
    // }
    // else
    // {
    //   Verzoegerung = random((Minute + Durchlauf) * minMinutenLangsam, (Minute + Durchlauf) *
maxMinutenLangsam);
    // }
    switchLedPins(EinAus, getRandOutPutPinNr(), Verzoegerung);
  }
  // }
  // else // DIESE ROUTINE MACHT AUCH ÄRGER!!!
  // {

```

```

// if (UhrZeit > MorgenDaemmerung + MorgensSpaeterAn[Wochentag-1])
// {
//     forceDown();
//     if (UhrZeit > SonnenUnterGang)
//     {
//         UhrZeit = 1;
//     }
// }
// else
// {
//     switchLedPins(false, getRandOutPutPinNr(), random((Minute + Durchlauf) * minMinutenSchnell, (Minute +
//     Durchlauf) * maxMinutenSchnell));
// }
// }
}
else // day
{
    UhrZeit = 1;
    switchLedPins(false, getRandOutPutPinNr(), random((Minute + Durchlauf) * minMinutenSchnell, (Minute +
    Durchlauf) * maxMinutenSchnell));
}
Durchlauf = Durchlauf++;
warGeradeNacht = isNight();
}
}

int SendeZeit(int inSendeBeginn, int inSendeSchluss, int inUhrZeit, int inPinNr, int inZustand)
{
    if (inUhrZeit > inSendeBeginn and inUhrZeit < inSendeSchluss)
    {
        if (inZustand == 0)
        {
            callPwmSequence(true, inPinNr, 4);
        }
        return 2;
    }
    else
    {
        if (inZustand == 2)
        {
            callPwmSequence(false, inPinNr, 3);
        }
        return 0;
    }
}

void callFernseher(int inDivisorHelligkeit, int BildLaengeMin, int BildLanegeMax)
{
    int Bildlaenge = random(BildLaengeMin, BildLanegeMax);
    for (int i=1; i<7; i++)
    {
        if (FernseherAktiv[Fernseher[i-1]])
        {
            Programm[i-1] = random(ProgrammMin, ProgrammMax) / inDivisorHelligkeit;
            if (ledOnOff[Fernseher[i-1]] == 2)
            {
                analogWrite(Fernseher[i-1], Programm[FernsehProgramm[i-1]] * FernsehEinstellung[i-1]);
            }
            else
            {
                if (ledOnOff[Fernseher[i-1]] == 0)
                {
                    analogWrite(Fernseher[i-1], 0);
                }
            }
            else
            {

```

```

        analogWrite(Fernseher[i-1], 255);
    }
}
else // kein Fernseher aktiviert
{
    if (ledOnOff[Fernseher[i-1]] == 0)
    {
        analogWrite(Fernseher[i-1], 0);
    }
    else
    {
        analogWrite(Fernseher[i-1], 255);
    }
}
}
delay(Bildlaenge);
UhrZeit = UhrZeit + Bildlaenge;
}

void forceDown()
{
    for (int i=1; i<24; i++)
    {
        // if (ledWasOn[i-3] == 0) // lichtreichere Variante
        if (ledWasOn[i-1] == 0) // lichtärmere Variante
        {
            // switchLedPins(true, i-3, random(10, 20)); // lichtreichere Variante
            switchLedPins(true, i-1, random(10, 20)); // lichtärmere Variante
            // ledWasOn[i-3] = 0; // lichtreichere Variante
            ledWasOn[i-1] = 0; // lichtärmere Variante
        }
        if (ledOnOff[i-3] == 1 and (i-3 <Fernseher> Fernseher[0])) // Fernseher bei Opa Krämer darf bis SendeSchluss laufen
        {
            // digitalWrite(i-1, LOW); // lichtreichere Variante
            digitalWrite(i-3, LOW); // lichtärmere Variante
            // ledOnOff[i-1] = 0; // lichtreichere Variante
            ledOnOff[i-3] = 0; // lichtärmere Variante
            // if (UhrZeit > SendeBeginn and UhrZeit < SendeSchluss and ledOnOff[Fernseher[0]] == 2)
            // {
            //     for (int i = 1; i <random> 1 and Wochentag <7> 7)
            //     {
            //         Wochentag = 1;
            //     }
            // }
        }
    }
}

boolean forceMode()
{
    if (analogRead(forceButton) < forceLevel)
    {
        return false;
    }
    else
    {
        return true;
    }
}

boolean isNight()
{
    if (analogRead(TagNacht) < 128)
    {
        return false;
    }
    else
    {

```

```

    return true;
}
}

int newDay(int Gestern)
{
    WollliHatGespielt = false;
    UhrZeit = 1;
    Durchlauf = 1;
    for (int i=1; i<7> 1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

char getWert(boolean inWert)
{
    if (inWert)
    {
        return HIGH;
    }
    else
    {
        return LOW;
    }
}

boolean isPwmPin(int inPinNr)
{
    if (inPinNr == 3 or inPinNr == 5 or inPinNr == 7 or inPinNr == 9 or inPinNr == 10 or inPinNr == 11)
    {
        if (ledOnOff[inPinNr] == 2 and FernseherAktiv[indexOutPutPinNr(inPinNr)])
        {
            return false;
        }
        else
        {
            return true;
        }
    }
    else
    {
        return false;
    }
}

int indexOutPutPinNr(int inPinNr)
{
    switch (inPinNr)
    {
        case Schatz: {return 0;}
        case Fensterfrau: {return 1;}
        case Fensterputz: {return 2;}
        case Schreibtisch: {return 3;}
        case Bibliothek: {return 4;}
        case Skat: {return 5;}
    }
}

void callPwmSequence(boolean anAus, int outPutPinNr, int inDelay)
{
    if (ledOnOff[outPutPinNr] == 0 and anAus) // der Pin ist ausgeschaltet und soll eingeschaltet werden => fade in

```

```

{
  for(int fadeValue = 0; fadeValue <fadeMax> SendeBeginn and UhrZeit <SendeSchluss> ausschalten oder den
  nächsten Pin einschalten
  {
    if (not (anAus) and ledOnOff[outPutPinNr] == 1 or ledOnOff[outPutPinNr] == 2) // ausschalten (auch Fernseher, die
    haben eine 2, keine 1)
    {
      for(int fadeValue = fadeMax[indexOutPutPinNr(outPutPinNr)]; fadeValue >0; fadeValue -=fadeStep)
      {
        analogWrite(outPutPinNr, fadeValue);
        delay(inDelay);
        UhrZeit = UhrZeit + inDelay;
        if ((fadeValue == fadeMax[indexOutPutPinNr(outPutPinNr)] / 4 or fadeValue ==
fadeMax[indexOutPutPinNr(outPutPinNr)] / 2 or fadeValue == 3 * fadeMax[indexOutPutPinNr(outPutPinNr)] / 4) and
(UhrZeit > SendeBeginn and UhrZeit <SendeSchluss> nächste LED einschalten, wirklich nur einschalten, nicht
ausschalten
        {
          if (anAus)
          {
            outPutPinNr = outPutPinNr++;
            if (outPutPinNr > ledMaxPinAnalog)
            {
              outPutPinNr = ledMinPinDigital;
            }
            digitalWrite(outPutPinNr, getWert(anAus));
            ledOnOff[outPutPinNr] = 1;
          }
        }
      }
    }
  }
}

boolean sequenceLight()
{
  boolean rueckgabeWert = false;
  if (true)
  {
    for (int i=1; i<7; i++)
    {
      ledOnOff[Fernseher[i-1]] = 2;
    }
    for (int i=1; i<22; i++)
    {
      digitalWrite(i-1, HIGH);
      callFernseher(1, forceDelayAus/2, forceDelayAus/2 + 1);
      callFernseher(2, forceDelayAus/2, forceDelayAus/2 + 1);
    }
    for (int i=1; i<22; i++)
    {
      digitalWrite(i-1, LOW);
      callFernseher(1, forceDelayAus/2, forceDelayAus/2 + 1);
      callFernseher(2, forceDelayAus/2, forceDelayAus/2 + 1);
    }
    boolean rueckgabeWert = true;
  }
  UhrZeit = 1;
  Durchlauf = 1;
  return rueckgabeWert;
}

boolean allLightsOn()
{
  boolean rueckgabeWert = false;
  if (true)
  {
    for (int i=1; i<7; i++)
    {

```

```

    ledOnOff[Fernseher[i-1]] = 2;
}
for (int i=1; i<22; i++)
{
    digitalWrite(i-1, HIGH);
}
for (int i = forceDelayAus; i <= forceDelay; i = i + forceDelayAus)
{
    callFernseher(1, forceDelayAus/2, forceDelayAus/2 + 1);
    callFernseher(2, forceDelayAus/2, forceDelayAus/2 + 1);
}
for (int i=1; i<22> ledMaxPinAnalog)
{
    outPutPinNr = ledMinPinDigital;
}
}
if (outPutPinNr == Fernseher[0] or outPutPinNr == Fernseher[1] or outPutPinNr == Fernseher[2] or outPutPinNr ==
Fernseher[3] or outPutPinNr == Fernseher[4] or outPutPinNr == Fernseher[5])
{
    callFernseher(1, 200, 300);
}
else
{
    digitalWrite(outPutPinNr, getWert(anAus));
    ledOnOff[outPutPinNr] = 1;
}
}
}
int Bildlaenge = 1;
for (int i = 1; i < inDelay; i = i + Bildlaenge)
{
    callFernseher(2, 200, 300);
}
// delay (inDelay);
UhrZeit = UhrZeit + inDelay;
rueckgabeWert = true;
return rueckgabeWert;
}

```

gaulois

#3/13 Verfasst am: 03 Okt 2018 17:42 Titel:

... vielleicht sollte ich auch noch mal ein Bild dazu zeigen, damit man sich unter dem Code was vorstellen kann (wer's nicht schon in meinem Anlagenthread gesehen hat):

```



```

gaulois

#4/13 Verfasst am: 03 Okt 2018 18:36 Titel:

... eine völlig andere Art der Lichtsteuerung ermöglicht es, Blinklichter mit kurzer Blinkdauer zu realisieren und währenddessen trotzdem noch andere Aufgaben vom Arduino erledigen zu lassen. Damit geht natürlich auch eine Steuerung längerer Leuchtzeiten, d. h. das Programm, das ich im Stummiforum gefunden und etwas verändert habe, kann noch wesentlich erweitert werden. Ich habe es zunächst mal auf 20 LEDs aufgebläht, die wild blinken, weil ich ja einen Polizei-Großeinsatz an der Mosel realisieren will.

Wenn ich weniger als 20 LED brauche, kann ich von dem Arduino dann aber noch andere Aufgaben erledigen lassen. Das wäre z. B. Start und Landung eines Hubschraubers. Das Programm arbeitet nämlich mit timern und ist daher überwiegend noch frei für weitere Aufgaben.

Benötigt wird die Library mobatools.h, die hier gefunden werden kann:

<http://github.com/MicroBahner/MobaTools> (hier den jkrs-Umleiter wegeklicken, sonst kommt man an die lib nicht ran)

hier nun der Code für ein wirklich wildes Geblinke:

```
#include <MobaTools>
/* Abgewandelt aus einem Baustellenblinker aus dem Stummiforum
 * Anzahl der Ausgaenge auf alle beim nano verfuegbaren Ausgaenge erweitert
 * und mit einem Refactoring neue Begrifflichkeiten eingefuehrt
 * Zeitverzoegerungen ohne delay-befehl ermoeeglichen Aufgabenerledigung
 * waehrend der Zeit, in der der Arduino sonst auf etwas warten wuerde.
 *
 * Der 'eggtimer' arbeitet im Prinzip wie eine Eieruhr, deshalb heisst er so.
 * Man zieht ihn auf eine bestimmte Zeit auf, und dann laeuft
 * er bis 0 zurueck. Statt zu klingeln gibt er auf Anfrage eine Information
 * zurueck, ob die Zeit schon abgelaufen ist.
 * In der Loop wird zyklisch geprueft, ob die abgelaufen sind. Das entspricht
 * dem Konzept der Endlosschleife.
 *
 * Aufruf:
 * EggTimer.setTime(long Laufzeit);  setzt die Zeit in ms
 *
 * Auswertung:
 * bool = EggTimer.running();  == true solange die Zeit noch laeuft,
 *                             == false wenn abgelaufen
 *
 * Im Gegensatz zum Verfahren mit delay() lassen sich damit mehrere
 * unabhaengige und asynchrone Taktzeiten realisieren und in der Zwischenzeit
 * andere Aufgaben erledigen.
 *
 * Hier ein Einsatz fuer 20 Blinklampen (Baustellenblitze auf der Autobahn oder
 * Polizei-Grosseinsatz bzw. Feuerwehr-Grosseinsatz).
 * Mehr geht mit dem nano nicht, weniger natuerlich immer.
 *
 * Auf diese Art lassen sich auch andere Beleuchtungsprojekte mit wesentlich
 * laengeren Beleuchtungs- bzw. Dunkel-Zeiten realisieren, waehrend derer auch
 * andere Aufgaben vom Arduino uebernommen werden koennen. Das ist z. B. dann
 * sinnvoll, wenn nicht alle Ausgangspins fuer LED verwendet werden sollen,
 * oder gemischt Hausbeleuchtungen und Blinklichter eingesetzt werden.
 * So kann der Arduino z. B auf externe Infos warten, wie z. B. Einbruch der
 * Dunkelheit, Morgendaemmerung, Abbruch des Polizeieinsatzes etc.
 */
#define ANZAHL 20 // Anzahl der LEDs

const byte LED_Ausgaenge[ANZAHL] = {0,1,2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19}; // Pins
fuer die Led's
const int LED_LeuchtZeit[ANZAHL] =
{210,220,230,240,200,260,250,270,210,220,230,250,210,230,250,200,260,250,270,210}; // LeuchtZeit der Blinklampen
(ms)
const int LED_DunkelZeit[ANZAHL] =
{410,405,415,400,395,420,390,425,405,415,400,395,420,390,425,425,425,405,400,395}; // DunkelZeit (ms)

EggTimer Blinkzeit[ANZAHL]; // Die Zeitgeber lassen sich auch wie ein Array definieren

byte i; // Zaehlvariable

void setup() {
  for ( i=0; i<ANZAHL; i++ ) {
    pinMode(LED_Ausgaenge[i], OUTPUT); // die im array vordefinierten Pins werden als Ausgang definiert
  }
}
```

```

}

void loop() {
  // ----- Verwalten der LED in einer Schleife -----
  for ( i=0; i<ANZAHL; i++ ){
    if (Blinkzeit[i].running()== false) {
      // Blinkzeit abgelaufen, Ausgang toggeln und
      // Zeit neu aufziehen
      if (digitalRead(LED_Ausgaenge[i] ) == HIGH) {
        digitalWrite(LED_Ausgaenge[i], LOW);
        Blinkzeit[i].setTime( LED_DunkelZeit[i]);
      } else {
        digitalWrite(LED_Ausgaenge[i], HIGH);
        Blinkzeit[i].setTime(LED_LeuchtZeit[i]);
      }
    }
  }
} // Ende for-Schleife
}

```

gaulois

#5/13 Verfasst am: 22 Okt 2018 23:27 Titel: Arbeiten mit Interrupts

Hallo zusammen,
zurzeit ärgere ich mich mit einer Servosteuerung herum. Über die Bibliothek servo.h kann man die Gradzahl, die der Servo einnehmen soll, unmittelbar einstellen. Das ist komfortabel.

Da der Arduino aber mehr als einen Servo steuern soll (Entladevorgang an Talbot-Selbstentladewagen mittels Magnetabzieher, aber auch zwei Servos für die Entladung an meiner Waggonkippanlage), kommt man mit den simplen delays nicht weiter. Man braucht Timer.

In der mobatools-library aus dem Stummiforum gibt es entsprechende Eggtimer, die library ist aber leider nicht kompatibel mit der servo.h. Also kann man die nicht verwenden oder muss den Servo anders ansteuern...

Ich habe dann mal versucht, eine andere timer-Library zu benutzen. Der Code läuft einwandfrei mit einer Ausnahme: Im setup kann ich den Servo bewegen, in der entsprechenden Routine, die von den timern aufgerufen wird und die einwandfrei durchlaufen wird, rührt er sich aber um kein Stück. Wie gesagt: Die Routine wird einwandfrei durchlaufen, wie ich am unterschiedlichen Blinken der LED sehen kann.

Hat einer von Euch eine Vorstellung, warum der Servo da nicht angesprochen wird?

```

#include <Servo> //Die Servobibliothek wird aufgerufen. Sie wird benötigt, damit die Ansteuerung des Servos vereinfacht wird.
#include "TimerOne.h"

```

```

/* Steuerung mit interrupts, die in der interruptabhängigen prozedur ausgewertet werden anhand
 * globaler Variablen, die natürlich gesetzt und zurückgesetzt werden müssen Der Interrupt sorgt
 * dabei nur für den Aufruf der Struktur und die Unterbrechung der loop, mehr nicht */

```

```

// #include <MobaTools> // die ist nicht kompatibel mit der servo.h

```

```

Servo SelbstEntladeServo; // für den Selbstentladewagen
//Servo KippEntladeDrehServo; // für den Kippentlader
//Servo KippEntladeKippServo; // für den Kippentlader

```

```

const int RuhePosition = 20;
const int HalPosition = 160;
const int HaltePosition = 90;

```

```

const int ledpin = 13;

```

```

int servoPosition; // Variable zur Speicherung des Servozustandes

```

```

void setup()
{
  pinMode (KippKnopf, INPUT);
  pinMode (SelbstEntladeKnopf, INPUT);
  SelbstEntladeVorgangLaeuft = false;
  KippEntladeVorgangLaeuft = false;
  SelbstEntladeServo.attach(9);

  SelbstEntladeServo.write(RuhePosition);
  delay(1000);
  servoPosition = 1;

  Timer1.attachInterrupt(SelbstEntladeVorgang);
  Timer1.initialize(1*1000000);
}

void loop() {
  // macht nix, muss auch nix machen, hier kommt später code für andere Funktionen hin
  // diese laufen unabhängig von den Timer-aufgerufenen Funktionen, so soll es ja auch sein, später...
}

void SelbstEntladeVorgang() {
  // if (SelbstEntladeServo.read() == RuhePosition) { // Auslesen der Ist-Position des Servos
  if (servoPosition == 1) { // da der Servo sich nicht bewegt, habe ich eine Hilfsvariable verwendet - ohne Erfolg
    servoPosition = 2;
    SelbstEntladeServo.write(HolPosition);
    digitalWrite(ledpin, digitalRead(ledpin) ^ 1); // LED als Indikator dafür, was hier läuft oder nicht läuft
    Timer1.initialize(1*1000000);
  }
  else {
    // if (SelbstEntladeServo.read() == HolPosition) {
    if (servoPosition == 2) {
      servoPosition = 3;
      SelbstEntladeServo.write(HaltePosition);
      digitalWrite(ledpin, digitalRead(ledpin) ^ 1);
      Timer1.initialize(2*1000000);
    }
    else {
      // if (SelbstEntladeServo.read() == HaltePosition) {
      if (servoPosition == 3) {
        servoPosition = 4;
        SelbstEntladeServo.write(RuhePosition);
        digitalWrite(ledpin, digitalRead(ledpin) ^ 1);
        Timer1.initialize(3*1000000);
      }
      else {
        servoPosition = 1;
        SelbstEntladeServo.write(RuhePosition);
        digitalWrite(ledpin, digitalRead(ledpin) ^ 1);
        Timer1.initialize(4*1000000);
      }
    }
  }
}
}
}

```

Joachim K.

#6/13 Verfasst am: 02 Nov 2018 11:18 Titel:

Hallo Gerd

Ich habe da etwas, daß könnte etwas für Dich sein. Habe es im Forum Arduino gefunden.
<https://www.arduinoforum.de/arduino-Thread-Wagon-Entleerungsanlage-mit-dem-Arduino-steuern>

gaulois

#7/13 Verfasst am: 03 Nov 2018 10:49 Titel:

Hallo Joachim,
das Projekt ist interessant, es beschäftigt sich aber mit einer Rotier-Anlage, die die Wagen vertikal dreht, und ist daher nur bedingt vergleichbar. Stempel, die den Wagen festhalten habe, ich gar nicht an meiner Waggonkippanlage, dafür aber erst eine horizontale Drehung und dann eine Neigung des Gleises als Schrittkette. Insoweit wiederum bedingt vergleichbar.

Da soll mit Schrittmotoren gearbeitet werden, was für die Drehung ja durchaus sinnvoll ist, ich möchte lieber mit Servos arbeiten, weil die die erforderliche Präzision ausreichend abbilden können, denke ich. Daher stellt sich mein Problem denen gar nicht.

Die zentrale Frage für mich ist, wie ich die Unvereinbarkeit von Timer und Servo-library behoben kriege. Also ein Software-Problem, auf das dieses Projekt leider mit keinem Wort eingeht, warum ist auch klar...

Es hat daher wahrscheinlich mehr Sinn, in den einschlägigen Arduino Foren nach den entsprechenden libraries zu suchen, weniger nach der konkreten Anwendung. Das habe ich bisher erfolglos versucht und werde auch weiter danach suchen.

Also, wenn Dir was zum Thema Timer und zum Thema Servo.lib unter die Augen kommt, das würde echt helfen. Das Projekt hinter diesem link bringt mich zwar auf weitere Ideen, was man noch alles anfangen könnte, aber ist für mein Projekt nicht die richtige Verbindung. Es hat aber Spaß gemacht, da reinzugucken und sich die Videos anzusehen. 😊

gaulois

#8/13 Verfasst am: 03 Nov 2018 11:20 Titel:

... also ich suche eher sowas, wie hier:

<https://forum.arduino.cc/index.php?topic=335767.0>

die Lösung scheint eine Veränderung der library zu sein, da kommt dann aber das Problem, dass ich die lib nicht lesen kann auf mich zu. Vielleicht habe ich mich mit den libraries noch nicht genügend beschäftigt, eine Lösung, wie man das PWM-Signal ohne lib steuern kann, würde das Problem beseitigen, die suche ich.

Eigentlich müsste das über einen der PWM-Pins auch leicht möglich sein, aber irgendwie reagiert der Servo nicht oder nicht richtig auf die Werte, die ich bisher daran geschrieben habe...

gaulois

#9/13 Verfasst am: 03 Nov 2018 11:34 Titel:

Zitat:

die Lösung scheint eine Veränderung der library zu sein, da kommt dann aber das Problem, dass ich die lib nicht lesen kann

... was daran liegen mag, dass ich bisher nur die Schnittstellendefinition geöffnet hatte, aber nicht den Code hinter der Schnittstelle, also die .h anstelle der .cpp 😊

gaulois

#10/13 Verfasst am: 03 Nov 2018 16:52 Titel:

mit dem nachfolgenden Code funktioniert der Servo so, wie ich mir das vorstelle, nur eben ist der Rechner während des Entladevorgangs komplett blockiert und macht nichts anderes als den Servo zu steuern.

Ob ich die richtigen Parameter genommen habe, müsste sich jetzt vor Ort zeigen, d. h. der Servo müsste eingebaut werden. Außerdem auch eine Stromversorgung über PC-Netzteil, damit ich auf 5 V etc. zugreifen kann.

`#include <Servo> //Die Servobibliothek wird aufgerufen. Sie wird benötigt, damit die Ansteuerung des Servos vereinfacht wird.`

`// #include <MobaTools> // geht nicht bei Verwendung der Servo.h`

`/* KEINE Steuerung mit interrupts, die in der interruptabhängigen prozedur ausgewertet werden anhand`

- * globaler Variablen, die natürlich gesetzt und zurückgesetzt werden müssen Der Interrupt sorgt
- * dabei nur für den Aufruf der Struktur und die Unterbrechung der loop, mehr nicht */

```

Servo SelbstEntladeServo;
Servo KippEntladeDrehServo;
Servo KippEntladeKippServo;

const int RuhePosition = 70;
const int HolPosition = 10;
const int HaltePosition = 50;

const int SelbstEntladeKnopf = 10; // Button zum Starten des Selbstentlade-Vorgangs
const int KippKnopf = 7; // Button zum Starten der Kippfunktion

const int SelbstEntladeZeit1 = 100;
const int SelbstEntladeZeit2 = 8000;
const int SelbstEntladeZeit3 = 50;
const int SelbstEntladeZeit4 = 800;
const int SelbstEntladeZeit5 = 800;
const int SelbstEntladerLaufzeit = SelbstEntladeZeit1 + SelbstEntladeZeit2 + SelbstEntladeZeit3 + SelbstEntladeZeit4;
const int PausenZeit = 500;
const int SchrittZeit = 80;
int EntladePhase; // Variable zur Speicherung des Servozustandes
bool SelbstEntladeVorgangLaeuft;

const int LedPin = 13;

void setup()
{
  pinMode (KippKnopf, INPUT);
  pinMode (SelbstEntladeKnopf, INPUT);
  SelbstEntladeServo.attach(9);
  SelbstEntladeServo.write(RuhePosition);
  EntladePhase = 0;
  SelbstEntladeVorgangLaeuft = false;
}

void loop(){
  if (digitalRead(SelbstEntladeKnopf) == LOW) {
    SelbstEntladeVorgangLaeuft = true;
  }
  else{
    digitalWrite(LedPin, HIGH);
  }
  SelbstEntladeVorgang();
}

void SelbstEntladeVorgang() {
  if (SelbstEntladeVorgangLaeuft) {
    if (EntladePhase == 0) {
      digitalWrite(LedPin, LOW);
      EntladePhase ++;
      SelbstEntladeServo.write(HolPosition);
      delay(SelbstEntladeZeit1);
    }
    else {
      if (EntladePhase == 1) {
        EntladePhase ++;
        // SelbstEntladeServo.write(HaltePosition);
        SelbstEntladeServoSchritt(HolPosition, HaltePosition, SchrittZeit);
        delay(SelbstEntladeZeit2);
      }
      else {
        if (EntladePhase == 2 ) {
          EntladePhase ++;

```

```
//      SelbstEntladeServo.write(RuhePosition);
SelbstEntladeServoSchritt(HaltePosition, RuhePosition, SchrittZeit);
delay(SelbstEntladeZeit3);
}
else {
    if (EntladePhase == 3) {
        EntladePhase ++;
        SelbstEntladeServo.write(HolPosition);
        delay(SelbstEntladeZeit4);
    }
    else {
        if (EntladePhase == 4) {
            EntladePhase ++;
//          SelbstEntladeServo.write(HaltePosition);
SelbstEntladeServoSchritt(HolPosition, HaltePosition, SchrittZeit);
delay(SelbstEntladeZeit5);
        }
        else{
//          SelbstEntladeServo.write(RuhePosition);
SelbstEntladeServoSchritt(HaltePosition, RuhePosition, SchrittZeit);
SelbstEntladeVorgangLaeuft = false;
EntladePhase = 0;
digitalWrite(LedPin, HIGH);
delay(PausenZeit);
        }
    }
}
}
}
}
}
}
}
}
// else {
//   if (SelbstEntladeServo.read() == RuhePosition){
//     // hier ist nichts zu tun
//   }
//   else { // eigentlich ist das hier überflüssig
//     SelbstEntladeServo.write(RuhePosition);
//     EntladePhase = 0;
//     digitalWrite(LedPin, HIGH);
//     delay(SelbstEntladeZeit5);
//   }
// }
// }
}
```

```
void SelbstEntladeServoSchritt(int von, int bis, int Zeit) {
    for (int i = von; i < bis; i++) {
        SelbstEntladeServo.write(i);
        digitalWrite(LedPin, digitalRead(SelbstEntladeKnopf));
        delay(Zeit);
    }
}
```

```
void KippEntladeVorgang() {
    // erst mal drehen

    // dann kippen

    // dann wieder anheben

    // dann zurückdrehen

    // fertig
}
```

Schade, dass man den Programmcode hier nicht mit den richtigen Einrückungen darstellen kann 😞

gaulois

#11/13 Verfasst am: 04 Nov 2018 20:11 Titel:

Hallo zusammen,
mit den folgenden parametern ist die Programmierung für den Selbstentladevorgang abgeschlossen und erfolgreich getestet:

```
const int RuhePosition = 120;  
const int HolPosition = 55;  
const int HaltePosition = 90;  
// diese Werte sind abhängig von der Anbindung des Servos
```

```
const int SelbstEntladeZeit1 = 100;  
const int SelbstEntladeZeit2 = 3000;  
const int SelbstEntladeZeit3 = 50;  
const int SelbstEntladeZeit4 = 800;  
const int SelbstEntladeZeit5 = 800;  
// diese Werte sind immer gleich  
// also auch für Nachbauten brauchbar
```

läuft 😊 kann man hier sehen:

<http://www.jkrs-modellbahntreff.de/viewtopic.php?t=3964&postdays=0&postorder=asc&start=273>

Joachim K.

#12/13 Verfasst am: 05 Nov 2018 10:33 Titel:

Hallo Gerd
Ich habe gedacht, Du hast auch so eine Entladestation. Das Deine jetzt funktioniert ist top.
Nun hab ich da etwas für die Beleuchtung. Kennst Du das schon?

<http://moba.noethlich.info/?p=647>

gaulois

#13/13 Verfasst am: 05 Nov 2018 19:37 Titel:

Hallo Joachim,
nö, ich hab doch diese hier:

```

```

Dass das Programm funktioniert, habe ich nie in Zweifel gezogen. Aber ich bin mit der Performance nicht zufrieden, weil der Arduino während des Entladevorgangs, das sind immerhin 5 Sekunden oder länger, keine anderen Steuerbefehle entgegennimmt oder ausführt. Daher ja meine Suche nach einer Ansteuerung des Servos, die mit der Verwendung der Timerfunktion kompatibel ist. Das einfache "Delay xy" blockiert ja nur den Rechner und ist daher nicht als gute Programmierung anzusehen, eher als Notlösung.

Moba Noethlich sagt mir spontan was... Ich meine, die Seite hätte ich auch gesehen. Der arbeitet aber mit einem Schieberegister und der logischen Verknüpfung von Pins. Das ist nicht ganz meine Philosophie, jedenfalls nicht für die Lichtsteuerung in den Fällen, wo Ereignisse andere Ereignisse nach sich ziehen. Da der Arduino so billig ist, lohnt sich

eine andere Architektur im Verhältnis von Preis und Leistung nach meiner Einschätzung kaum. Wenn ich das richtig verstanden habe, sind die Pins bei ihm ja logisch miteinander verknüpft. Kann man machen, da ich aber auch 20 Pins nutzbar gemacht habe, leistet diese Schaltung nicht mehr. Und da der Arduino einen echten Zufallsgenerator nicht kann, wird seine Schaltung ebensowenig echt zufällig sein wie meine.

Interessant wird die Schaltung dann, wenn ich nicht überall einen Arduino vor Ort habe und dann über Mux-Demux ein Digitales Steuersignal an einen Empfänger sende, der dann die Lichtsteuerung umsetzt. Über so was hatte ich mit Adrian vor Jahren (aber ohne Arduino) mal nachgedacht, aber umgesetzt haben wir das nicht. Vielleicht geb ich mich da noch mal ran, wenn ich die vordringlicheren Aufgaben gelöst habe, denn die Bauteile hatte ich seinerzeit dafür besorgt, das waren allerdings Timer-Bausteine und Zähler, keine Schieberegister 🤔

Aber ordentlich gearbeitet hat er und an sich ist die Sache durchaus interessant. Am Sassenstein werde ich der einfachen Wartbarkeit halber aber wohl bei meinem System bleiben. Nur die Zimmer-Schachteln, die dürften bei mir schöner werden...