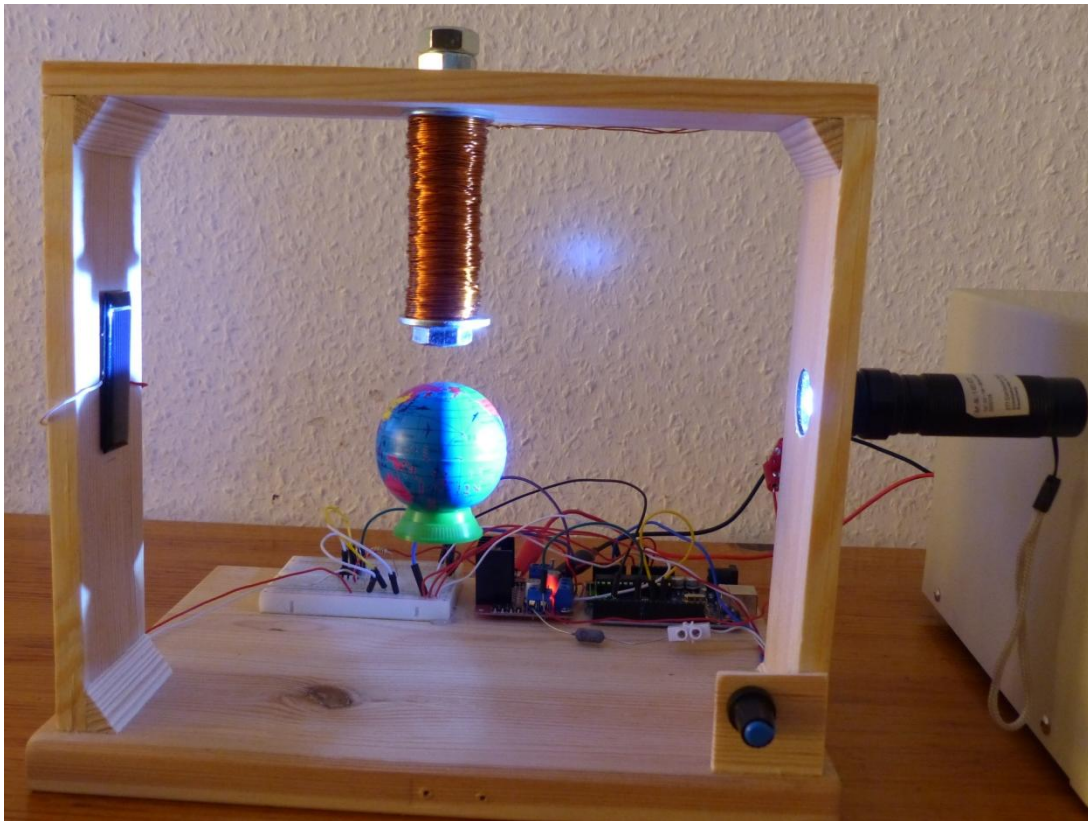


# Die schwebende Weltkugel

arduino, PID-Regelung, Solarzelle als Sensor



Hier schwebt eine Blechkugel, kein Neodym-Magnet !

Der Sensor für die Höhe der Kugel ist eine Solarzelle, auf der die Kugel einen Schatten wirft.

Wesentliche Bauteile:

- Solarpanel (0.5V, 100mA, 53mm x 18mm)
- Elektromagnet: Kupferlackdraht mit 0.5mm Durchmesser (mein Magnet: 70m Draht, entspricht etwa 125g Draht), Schraube M10 x 90 mit 2 M10-Muttern
- Netzgerät (12V,2A), gut stabilisiert
- Verstärker für Magnet: FET IRLZ14PBF (+ Freilaufdiode) (das ist neu: im Video mit Motortreiber L298N)
- Glühlämpchen mit Linse: Taschenlampen-Spitzlinse (2.5V, 300mA) und bei 12V-Versorgung ein 330Ohm-Vorwiderstand (5W) (sehr wichtig: gut stabilisierte, sehr glatte Spannung für die Lampe) oder: Taschenlampe mit Linse (LED, 1W) (oder: 12V-Halogenlämpchen ohne Linse, oder Lämpchen mit Linse (4,5V/0,3A)).
- Spitzer-Weltkugel (Durchmesser etwa 5cm) oder z.B. Pulmoll-Dose
- Operationsverstärker: MCP6021-I/P (2,5V bis 5,5v Rail2Rail I/O OPV, DIP-8 IC)

Mit Hilfe des Operationsverstärkers MCP6021-I/P und zwei Widerständen (invertierender Verstärker) wird eine Spannung generiert, die zur Kurzschlussstromstärke des Solarpanels weitgehend linear ist. Der Operationsverstärker wird mit  $V_{CC}$  an +5V des arduino angeschlossen. Die Kurzschlussstromstärke des Solarpanels und die beschienene Fläche des Panels sind ebenfalls weitgehend linear zueinander. Da die Weltkugel einen Schatten auf das Solarpanel wirft, hat man somit einen Sensor für die aktuelle Höhe der Weltkugel.

Eventuell muss für den Widerstand 5k $\Omega$  ein anderer Widerstandswert gewählt werden: Bei Vollaussleuchtung der Solarzelle, also ohne Schwebobjekt, sollte die Spannung von A0 gegenüber GND etwas weniger als 5V betragen. Es sollte keinesfalls eine Spannung über 5V an den Analog-Eingang gegeben werden.

Weitere Infos auf youtube:

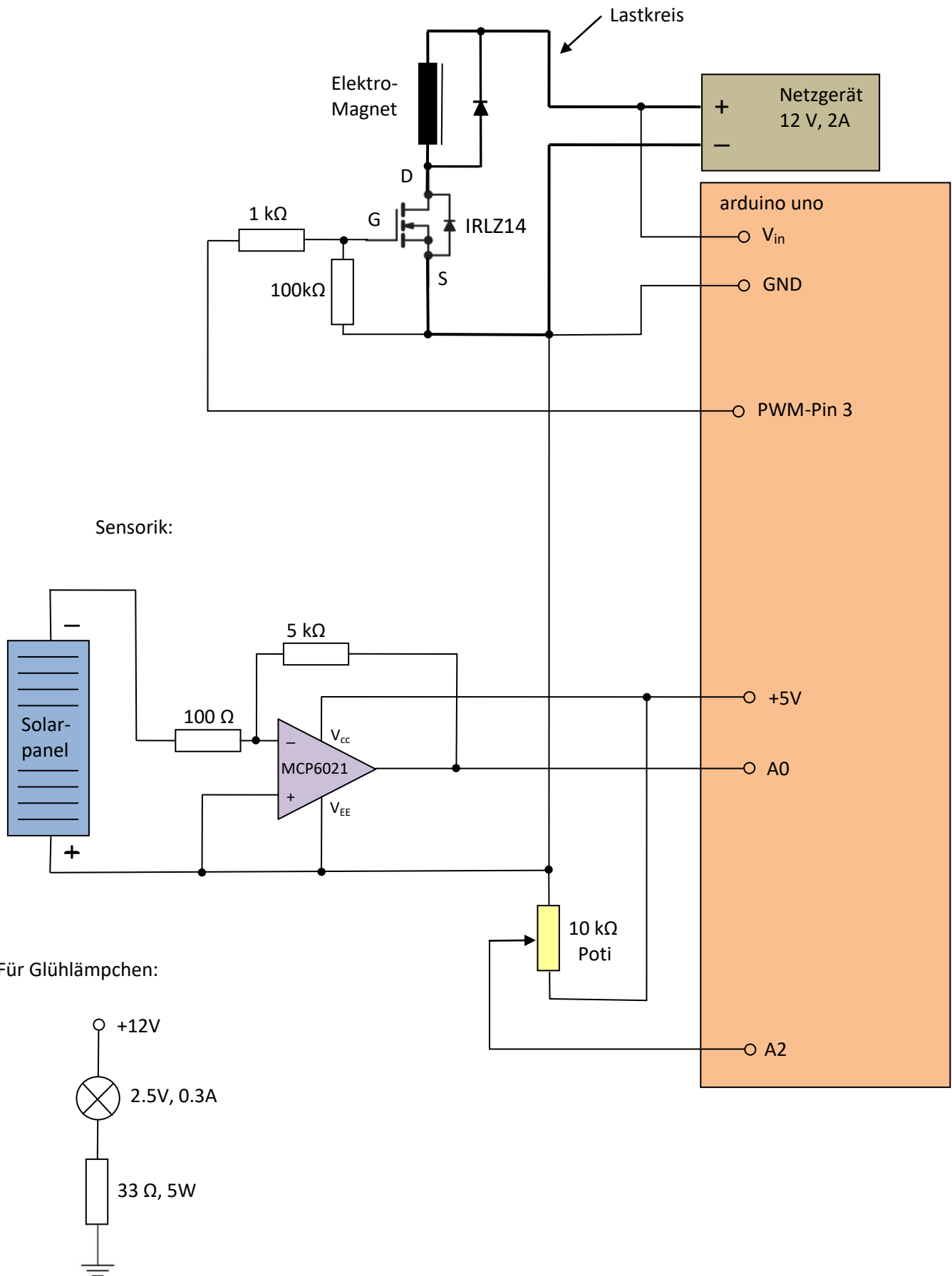
<https://www.youtube.com/watch?v=noMdS1ZenPQ&t=2s>

<https://www.youtube.com/watch?v=iQHHRfEgOEw>

Wesentliche Quelle: Kein Panik vor Regelungstechnik, Springer-Verlag

Foto: Weltkugel-Spitzer vom Moses-Verlag

# Schaltplan:



sketch für arduino uno:

```
// Projekt: Schwebende Weltkugel

// Objekt: Weltkugel-Spitzer aus Blech
// Elektromagnet am 17.3.2020 selbst gewickelt (Drahtdurchmesser 0,5mm, 100g Draht)
// Sensor: kleines Solarmodul (0,5V,100mA), Messgroesse: Kurzschlussstromstärke
// Gluehlaempchen mit Linse: Taschenlampen-Spitzlinse (2.5V,300mA)
// Gluehlaempchen an 12V anschliessen mit Drahtwiderstand: 33 Ohm, 5W
// Verstaerkung mit FET IRLZ14PBF, 100kOhm Widerstand zwischen Gate und Source
//                                     1kOhm Widerstand zwischen Gate und PWM-Pin 3
// mit schnellem PWM, auf Pin3 (31300 Hz)
// Netzgeraet: 12,0 Volt
// Schwebe-Abstand bis maximal 14mm
// Sample-Dauer: 2,5ms (also 400 PID-Berechnungen pro Sekunde)
// Serielle Ausgabe mit Baudrate 115200, an Plotter oder Monitor
// Alternativ: PWM-Pin 5 (980 Hz), ohne schnellem PWM
// 10. Maerz 2021
// by max

float kp = 1; // 0.5
float ki = 0.01; // 0.01
float kd = 15; // 10; 25
int mitte = 100; // 145 fuer analogWrite

float height;
float error, lastErr, errSum, dErr, output, setpoint, sp1, sp2, sp3;
float errSum_neu, kd_normal;
unsigned long loop_timer; // fuer Festlegung der Loop-Dauer
int incomingByte = 0; // Veraendern des D-Anteils
int height_without; // ohne Objekt, Solarmodul voll beleuchtet
int w1,w2,w3; // eventuell zur Mittelung der Hoehe

void setup(){
  Serial.begin(115200);
  TCCR2B = TCCR2B & 0b11111000 | 0x01; // erhoert PWM-Frequenz von Pin 3 auf
                                     // 31300 Hz (statt 490 Hz)
  pinMode(3, OUTPUT); // PWM für Spule, an H-Bruecke L298N
  pinMode(6, OUTPUT); // Zusatz-GND, fuer Pin7-Steckbruecke
  pinMode(7, INPUT_PULLUP); // fuer Poti-Betrieb ohne PID
  pinMode(9, OUTPUT); // Zusatz-GND, fuer Pin10-Steckbruecke
  pinMode(10, INPUT_PULLUP); // fuer Einschalten des Integral-Anteils
  pinMode(11, OUTPUT); // Zusatz-GND, fuer Pin12-Steckbruecke
  pinMode(12, INPUT_PULLUP); // fuer Ausschalten des Einlesens des Setpoints
  delay(3000);
  height_without = 1023-analogRead(A0); // "Hoehe" ohne Objekt
  setpoint = 450;
  kd_normal = kd;
  loop_timer = micros() + 2500; // ==> waere 400 Hz
}

void loop(){
  if (Serial.available() > 0) { // Nachjustieren des D-Anteils
    incomingByte = Serial.read();
    if (incomingByte == 'f') kd=kd+0.5; // rechts von "D"-Taste
    if (incomingByte == 's') kd=kd-0.5; // links von "D"-Taste
    if (incomingByte == 'n') kd=kd_normal; // fuer normal
  }
  if (digitalRead(12) == 1) setpoint = readSetpoint();
  height = 1023-analogRead(A0);
  // w1 = w2; // alternativ: zur Mittelung der letzten drei Messungen
  // w2 = w3;
  // w3 = analogRead(A0);
  //height = 1023-(w1+w2+w3)/3.0;
  //height = 0.6*height+(1-0.6)*(1023-analogRead(A0));

  //PID controller calculations:
  error = height - setpoint; // Differenz zwischen gemessener Hoehe und Wunschhoehe
  errSum_neu = errSum + ki*error; // I-Anteil
  if ((-errSum_neu+mitte>=0)&&(-errSum_neu+mitte<=255)) errSum = errSum_neu; // errSum begrenzen
  if (digitalRead(10) == 1) errSum = 0; // I-Anteil nur, wenn Pin10 auf Low gesetzt ist
  dErr = error - lastErr;

  output = kp * error + kd * dErr + errSum; // Output der PID-Berechnung
  //output = 0.6*output+(1-0.6)*(kp * error + kd * dErr + errSum);

  if (output > mitte) output = mitte; // begrenzen
  else if(output < (mitte-255)) output = mitte-255;
```

```

lastErr = error;

if (height < 360) output = mitte; // (height_without+50) Elektromagnet aus,
// wenn kein Objekt im Lichtweg
if (height > 750) output = mitte; // // Elektromagnet aus,
//wenn Solarmodul ganz dunkel

if (digitalRead(7)==0){ // Poti-Betrieb ohne PID, wenn Pin7 auf GND
  output = map(analogRead(A2),0,1023,mitte,mitte-255);
  setpoint = height;
  errSum = 0;
}

//output = 0;
analogWrite(3, -output+mitte); // aktuelle "Staerke" des Elektromagneten

Serial.print(height); // Hoehe
Serial.print(",");
Serial.print(setpoint); // Wunschhoehe
Serial.print(",");
Serial.print(-output+mitte+200); // "Staerke" des Elektromagneten, um 200
//verschoben, fuer bessere Sichtbarkeit im Plotter
Serial.print(",");
Serial.println(0); // für Plotter, als untere Grenze
//Serial.println((loop_timer-micros())/10); // Wie viel Zeit bleibt im Loop übrig?
while(loop_timer > micros()); // Warten, bis 2,5 Millisekunden vorbei sind
loop_timer += 2500;
}

float readSetpoint(){
  sp1 = sp2;
  sp2 = sp3;
  sp3 = analogRead(A2);
  return (sp1+sp2+sp3)/3.0; // Mittelung der letzten drei Messungen oder " /3 "
}

```

# Weitere Varianten:

Alternativ zu PWM und FET funktioniert auch Folgendes ähnlich gut:

## DAC und Konstantstromquelle

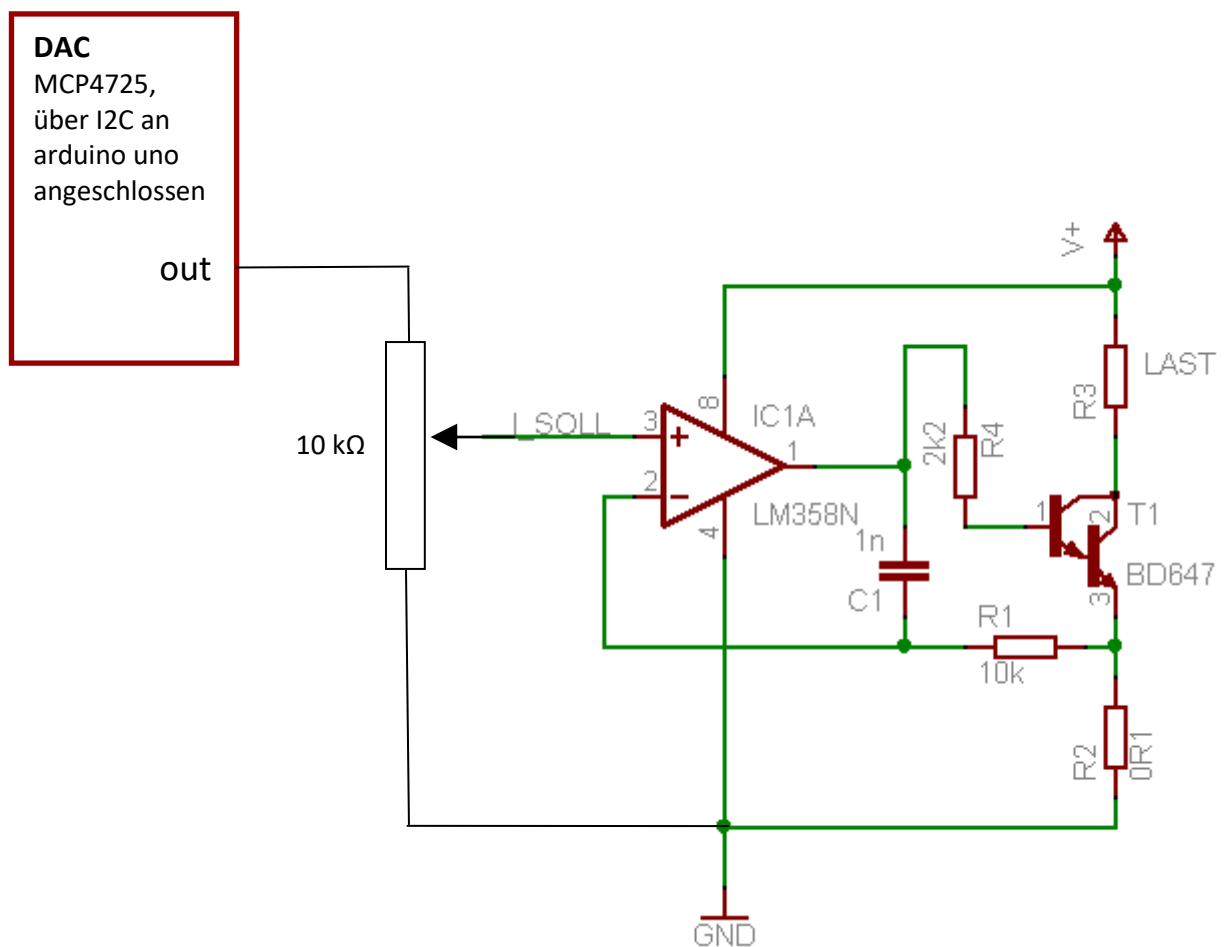
Statt PWM wird hier ein Digital-Analog-Wandler (MCP4725) verwendet.

Elektromagnet: Last R3,  $V+ = 12V$

Der Poti wird so eingestellt, dass wenn das Potential bei out +5V ist, die maximale gewünschte Stromstärke durch den Elektromagneten fließt.

Der Widerstand R2 arbeitet als shunt und regelt mit Hilfe des LM358 die Stromstärke durch die Spule.

Der Darlington-Transistor BD647 braucht einen guten Kühlkörper!!!



Quelle für das Grüne: <https://www.mikrocontroller.net/articles/Konstantstromquelle>

### Hier der sketch zur DAC-Variante:

```
// Projekt: Schwebende Weltkugel
// Objekt: Weltkugel-Spitzer aus Blech
// Elektromagnet am 17.3.2020 selbst gewickelt (Drahtdurchmesser 0,5mm, 100g Draht)
// Sensor: kleines Solarmodul (0,5V,100mA), Kurzschlussstromstärke
// 1W-LED-Taschenlampe
// mit DAC MCP4725, der Konstantstromquelle steuert
// Konstantstromquelle (OPV LM358 und Darlington-Transistor BD647 und shunt 0,12 Ohm)
// siehe: https://www.mikrocontroller.net/articles/Konstantstromquelle
// Netzgeraet: 12,0 Volt
// Loop-Dauer: 5 Millisekunden -> 200 Hz
// Schwebe-Abstand bis maximal 14mm
// Serielle Ausgabe mit Baudrate 115200, an Plotter oder Monitor
// Zuerst LED einschalten, dann Netzgeraet
// 10. April 2020
```

```

// by max

#include <Wire.h>

float kp = 10;    // 10
float ki = 0.1; // 0.1
float kd = 60;   // 60
int mitte =1400; // 1400 fuer setDAC

float height;
float error, lastErr, errSum, dErr, output, setpoint, sp1, sp2, sp3;
float errSum_neu, kd_normal;
unsigned long loop_timer;
int incomingByte = 0 ; // Veraendern des D-Anteils
int height_without; // ohne Objekt, Solarmodul voll beleuchtet

void setup(){
  Serial.begin(115200);
  TCCR2B = TCCR2B & 0b11111000 | 0x01; // erhoeht PWM-Frequenz von Pin 3 auf 31300 Hz
  //(statt 490 Hz)
  TWBR = 12;
  setDAC(0);
  pinMode(3, OUTPUT); // PWM für Spule, an H-Bruecke L298N
  pinMode(6, OUTPUT); // Zusatz-GND, fuer Pin7-Steckbruecke
  pinMode(7, INPUT_PULLUP); // fuer Poti-Betrieb ohne PID
  pinMode(9, OUTPUT); // Zusatz-GND, fuer Pin10-Steckbruecke
  pinMode(10, INPUT_PULLUP); // fuer Einschalten des Integral-Anteils
  pinMode(11, OUTPUT); // Zusatz-GND, fuer Pin12-Steckbruecke
  pinMode(12, INPUT_PULLUP); // fuer Ausschalten des Einlesens des Setpoints
  delay(100);
  height_without = 0.0*height+(1-0.0)*(1023-analogRead(A0));
  setpoint = 510; //
  kd_normal = kd;
  loop_timer = micros() + 5000;
}

void loop(){
  if (Serial.available() > 0) { // Nachjustieren des D-Anteils
    incomingByte = Serial.read();
    if (incomingByte == 'f') kd=kd+0.5; // rechts von "D"-Taste
    if (incomingByte == 's') kd=kd-0.5; // links von "D"-Taste
    if (incomingByte == 'n') kd=kd_normal; // fuer normal
  }
  if (digitalRead(12) == 1) setpoint = readSetpoint();
  height = 0.0*height+(1-0.0)*(1023-analogRead(A0));
  //PID controller calculations:
  error = height - setpoint; // Differenz zwischen Wunschhoehe und gemessener Hoehe
  errSum_neu = errSum + ki*error; // I-Anteil
  if ((-errSum_neu+mitte>=0)&&(-errSum_neu+mitte<=4095)) errSum = errSum_neu; // errSum begrenzen
  if (digitalRead(10) == 1) errSum = 0; // I-Anteil nur, wenn Pin10 auf Low gesetzt
  dErr = error - lastErr;

  output = 0.4*output+(1-0.4)*(kp * error + kd * dErr + errSum); // Output der PID-Berechnung

  if (output > mitte) output = mitte;
  else if(output < (mitte-4095)) output = mitte-4095;
  lastErr = error;

  if (height < height_without+50) output = mitte; // Elektromagnet aus, wenn kein Objekt
  //im Lichtweg
  if (height > 1000) output = mitte; // // Elektromagnet aus, wenn Solarmodul ganz dunkel

  if (digitalRead(7)==0){ // Poti-Betrieb ohne PID, wenn Pin7 auf GND
    output = map(analogRead(A2),0,1023,mitte,mitte-4095);
    setpoint = height;
    errSum = 0;
  }

  //output = 0;
  setDAC(-output+mitte); // aktuelle "Staerke" des Elektromagneten

  Serial.print(height); // Hoehe
  Serial.print(",");
  Serial.print(setpoint); // Wunschhoehe
  Serial.print(",");
  Serial.print(-output+mitte); // "Staerke" des Elektromagneten
  Serial.print(",");
  Serial.println(0); // für Plotter, als untere Grenze

```

```

//Serial.println((loop_timer-micros())/10); // Wie viel Zeit bleibt im Loop übrig?
while(loop_timer > micros()); // Warten, bis 5 Millisekunden um sind
loop_timer += 5000;
}

float readSetpoint(){
  sp1 = sp2;
  sp2 = sp3;
  sp3 = analogRead(A2);
  return (sp1+sp2+sp3)/3; // Mittelung der letzten drei Messungen
}

void setDAC(int x) { // 0..4095
  Wire.beginTransaction(0x60);
  Wire.write(0x40);
  Wire.write(x / 16);
  Wire.write((x % 16) << 4);
  Wire.endTransmission();
}

```

### Weitere Variante:

PWM-Ausgang am arduino, dann Filterung durch einen RC-Tiefpass, dann diese analoge Spannung an obige Konstantstromquelle geben.

Dies habe ich ausprobiert. Ich weiß nicht, ob das eine Standard-Lösung ist.

Es funktioniert, aber nicht so gut, wie die beiden ersten Lösungen. Das müsste man sehr verbessern.

Ich habe dafür den ersten sketch verwendet:

Für die PWM-Frequenz 31300 Hz am Pin3 habe ich gewählt:  $R_5 = 5k\Omega$ ,  $C_2 = 0,1\mu F$  (unveränderter sketch).

Für die PWM-Frequenz 490 Hz am Pin3 habe ich gewählt:  $R_5 = 5k\Omega$ ,  $C_2 = 4,7\mu F$

(Die Zeile `TCCR2B = TCCR2B & 0b11111000 | 0x01;` auskommentieren.

Für diesen Fall habe ich die loop-Dauer auf 5ms erhöht.)

