

```

/*
* 01_06_Mixmaster_2016_Waage_Programm
* Datum @ 16 Nov, 2016
* Programm zum Takten der Gläterschiene und zum Ausheben
* der Gläser mittels Hubeinheit ELKE 17/10
*/

#include <Stepper.h> // Inkludiert Headerdatei
Stepper.h

// Definiert Schritte des Motors
#define STEPS 200 // 360° / 1,8 = 200

// Anlegen eines Schrittmotors und Anlegen der Anschlusspins
Stepper stepper(STEPS, 8, 11, 12, 13);

// Definieren von Variablen zum Kommunizieren mit dem zweiten Genuino 101
byte Eingang_Cocktail = 3; // Pin 3 als Eingang_Cocktail
definieren (signalisiert "Cocktail mixen")
byte Eingang_Glas = 4; // Pin 4 als Eingang_Glas
definieren (signalisiert "Glas voll")

// Anlegen von Variablen der Hubeinheit
byte Eingang_Startimpuls = 5; // Pin 5 als Eingang_Startimpuls
definieren
byte Eingang_E1 = 6; // Pin 6 als Eingang_E1
definieren

// Anlegen einer Variablen des Endschalters
byte Eingang_Last = 7; // Pin 7 als Eingang_Last
definieren

// Definieren einer Schrittvariablen zum Hochzählen
int schritte = 0; // Zuweisung den Wert 0 der
Vaariablen schritte

// Definiert eine Variable um Referenzposition abzufragen
bool ref_OK;

// the previous reading from the analog input
int previous = 0;

void step(int steps)
{
    digitalWrite(9, HIGH); // Pin 9 auf high 1 schreiben
    digitalWrite(10, HIGH); // Pin 10 auf high 1 schreiben
    stepper.step(steps);
    digitalWrite(9, LOW); // Pin 9 auf low 0 schreiben

```

```

digitalWrite(10, LOW); // Pin 10 auf low 0 schreiben
}

void setup()
{
  // Definieren der Schrittgeschwindigkeit des Motors
  pinMode(9, OUTPUT); // Pin 9 als Ausgang definieren
  pinMode(10, OUTPUT); // Pin 10 als Ausgang definieren
  digitalWrite(9, LOW); // Pin 9 auf low 0 schreiben
  digitalWrite(10, LOW); // Pin 10 auf low 0 schreiben
  stepper.setSpeed(30); // Schrittgeschwindigkeit in
RPMs

  pinMode(Eingang_Startimpuls, OUTPUT); // Eingang_Startimpuls als
Ausgang definieren
  pinMode(Eingang_E1, OUTPUT); // Eingang_E1 als Ausgang
definieren
  pinMode(Eingang_Last, INPUT); // Eingang_Last als Eingang
definieren
  pinMode(Eingang_Glas, INPUT); // Eingang_Glas als Eingang
definieren
  pinMode(Eingang_Cocktail, INPUT); // Eingang_Cocktail als Eingang
definieren

  // Initialzustand Startimpuls und E1 auf 0V
  digitalWrite(Eingang_Startimpuls, HIGH); // Eingang_Startimpuls auf high
1 (0V an Elke)setzen
  digitalWrite(Eingang_E1, HIGH); // Eingang_E2 auf high 1 (0V an
Elke)setzen
}

void loop()
{
  // Definiert die Referenzfahrt
  if(ref_OK == false) // Dauerhafte Referenzfahrt
verhindern
  {
    referenzfahrt(); // Funktionsaufruf
referenzfahrt()
    delay(10000); // Wartezeit 10s zum Befüllen
der Glashalterung
  }

  if(cocktailReady() == true) // Signal "Cocktail mixen" =
high 1
  {

    // Eingabe der Schrittzahl zum Takten der Aushebeposition
    while(schritte != 500) // Schrittzähler ungleich 500

```

```

    {
        step(1); // Definiert 1 Schritt zum
Fahren bis 500 Schritte erreicht
        schritte++; // Schritte hoch Zählen
    }
    delay(1000); // Wartezeit 1000ms zur
Sicherung der Position

    // Definiert die Bedingung zum Ausheben des Glases
    if (schritte == 500) // Freigabe zum Ausfahren
(Motor an Aushebebelegung)
    {
        ausfahrenElke(); // Funktionsaufruf
ausfahrenElke()
    }

    while(digitalRead(Eingang_Glas) != 1) // Bleibt so lange in der
Schleife bis Signal "Glas voll" = high 1
    { // Sperrfunktion (verhindert zu
frühes Einfahren)
        ;
    }

    // Definiert die Bedingung zum Absenken des Glases
    einfahrenElke(); // Funktionsaufruf
einfahrenElke()
    delay(1000); // Wartezeit 1000ms um
Einfahren sicher zu stellen

    // Definiert Bedingung um in Befüllposition zu fahren
    step(500); // Definiert 500 Schritte um zu
Entnahmeposition zu fahren
    delay(10000); // Wartezeit 10s um Glas
entnehmen zu können
    }
}

// Definieren der Funktion referenzfahrt()
void referenzfahrt()
{
    while (digitalRead(Eingang_Last) == 1) // Signal Eingang_Last = high 1
(Endschalter nicht belegt)
    {
        step(1); // Definiert 1 Schritt zum
Fahren bis Endschalter belegt
    }
    ref_OK = true; // Signal Referenz erreicht =
high 1
}

```

```

// Definieren der Funktion cocktailReady()
bool cocktailReady()
{
    if(digitalRead(Eingang_Cocktail == 1)) // Signal "Cocktail mixen" =
high 1
    {
        return true; // Signal Cocktail soll
zubereitet werden
    }
    else // Signal "Cocktail mixen" =
low 0
    {
        return false; // Signal Cocktail soll nicht
zubereitet werden
    }
}

// Definieren der Funktion ausfahrenElke()
void ausfahrenElke()
{
    // Ausfahren Hubeinheit (Startimpuls = 24V , E1 = 24V)
    digitalWrite(Eingang_E1, LOW); // Eingang_E1 auf low (24V an
Elke) setzen
    digitalWrite(Eingang_Startimpuls, LOW); // Eingang_Startimpuls auf low
0 (24V an Elke) setzen
    delay (1000); // Wartezeit 1000 ms

    // Startimpuls auf 0V setzen um neue Startflanke zu ermöglichen
    digitalWrite(Eingang_Startimpuls, HIGH); // Eingang_Startimpuls auf high
1 (0V an Elke) setzen
    delay (50); // Wartezeit 50ms wegen
Startflanke
}

// Definieren der Funktion einfahrenElke()
void einfahrenElke()
{
    // Einfahren Hubeinheit (Startimpuls = 24V , E1 = 0V)
    digitalWrite(Eingang_E1, HIGH); // Eingang_E2 auf high 1 (0V an
Elke) setzen
    digitalWrite(Eingang_Startimpuls, LOW); // Eingang_Startimpuls auf low
0 (24V an Elke) setzen
    delay (1000); // Wartezeit 1000 ms

    // Startimpuls auf 0V setzen um neue Startflanke zu ermöglichen
    digitalWrite(Eingang_Startimpuls, HIGH); // Eingang_Startimpuls auf high
1 (0V an Elke) setzen
    delay (50); // Wartezeit 50ms wegen

```

Startflanke

}