

# Klasse „Schokidrucker“

**Verwendung:** Mit der Klasse können Objekte vom Typ `Schokidrucker` erstellt werden. Über ein solches Objekt, lässt sich dann der Schokodrucker einfach und bequem steuern.

**Einbindung:** Kopiere die `Schokidrucker.hpp`-Datei in den Ordner, in welchem auch die `.ino`-Datei liegt. Binde dann die Datei in deinen Programmcode ein, indem du ganz oben an den Anfang schreibst: `#include "Schokidrucker.hpp"`

**Objekt erstellen:** Denke dir einen netten Namen für den Schokodrucker aus (z.B. `Horst`) und erstelle mit folgendem Code das Objekt `Horst` vom Typ `Schokidrucker` :

```
Schokidrucker Horst(x,y,p1,p2,p3,p4,pixel,p5,p6,p7);
```

Was da in die Klammern muss, wird gleich erklärt.

**Auf Methoden zugreifen:**

Auf die Methoden von `Horst` (z.B. `MakeLine(x,y)`) wird wie folgt

zugegriffen: `Horst.MakeLine(3,7);`

## Methoden der Klasse:

Parameter beim erstellen,

Der Konstruktor setzt sich wie folgt zusammen:

```
Schokidrucker Horst(1,2,3,4,5,6,7,8,9,10);
```

1 = X-Wert der Startposition

2 = Y-Wert der Startposition

3 = Pinnummer für den X-Stepper

4 = Pinnummer für den Y-Stepper

5 = Pinnummer für die X-Richtung

6 = Pinnummer für die Y-Richtung

7 = Umrechnungsrate der Pixel und Steps (bei uns meistens 106, da das genau 1cm sind)

8 = Pinnummer für die Spritze in Richtung „Hoch“

9 = Pinnummer für die Spritze in Richtung „Runter“

10 = Pinnummer für die Geschwindigkeit der Spritze

## Wichtige Befehle:

void Goto(int x, int y)

→ Der Drucker fährt zur Position x,y. Dabei fährt er erst komplett in X-Richtung und dann in Y-Richtung.

void MakeLine(float x, float y)

→ erzeugt eine stufenlose Linie von der aktuellen Position zu den angegebenen Koordinaten.

void Print(float x[], float y[], int laenge, int anzahl)

→ Der Drucker fährt hintereinander die in den Arrays x und y beschriebenen Koordinaten an und druckt dabei. In laenge wird die Länge der Arrays angegeben und in anzahl, wie oft die Form abgefahren werden soll.

void SchokoLos()

→ Beginnt Schokolade zu drucken.

void SchokoStop()

→ Stoppt den Schokoladen Druck durch 2,5 sekündiges Hochziehen.

void Stop()

→ Stoppt die komplette Anlage.

void FullStop()

→ Stoppt die komplette Anlage und zieht für 2,5 Sekunden den Spritzenkolben hoch.

void End()

→ Beendet alles und stoppt den Loop.

void StartingPosition()

→ Fährt den Drucker auf seine Startposition.

void SpritzeStop()

→ Hält die Spritze an.

void SpritzeHoch(int v)

→ Zieht die Spritze mit der Geschwindigkeit v (0-128) nach oben.

void SpritzeRunter(int v)

→ Drückt die Spritze mit der Geschwindigkeit v (0-128) nach unten.

void SpritzeMove()

→ aktiviert die Spritze mit den vorher festgelegten Eigenschaften.

### **Geter und Seter Methoden:**

int GetX()

→ gibt die aktuelle X-Position zurück.

int GetY()

→ gibt die aktuelle Y-Position zurück.

int GetStartX()

→ gibt den X-Wert der Startposition zurück.

int GetStartY()

→ gibt den Y-Wert der Startposition zurück.

void SetStartX(int x)

→ Setzt den X-Wert der Startposition auf x.

void SetStartY(int y)

→ Setzt den Y-Wert der Startposition auf x.

void StartingPosition(int x, int y) → Setzt die Startposition und aktuelle Position auf x und y.

void Calibrate(int x, int y) → Setzt die aktuelle Position auf x und y.

void SetDelay(int d)

→ Setzt den Delay der Stepper.

int GetDelay()

→ Gibt den Delay der Stepper zurück.

int GetXPin()

→ Gibt den Pin des X-Steppers zurück.

int GetYPin()

→ Gibt den Pin des Y-Steppers zurück.

int GetXRPin()

→ Gibt den Pin der X-Richtung zurück.

int GetYRPin()

→ Gibt den Pin der Y-Richtung zurück.

void SetXPin(int pin)

→ Setzt den Pin des X-Steppers auf pin.

void SetYPin(int pin)

→ Setzt den Pin des Y-Steppers auf pin.

void SetXRPin(int pin)

→ Setzt den Pin für die X-Richtung auf pin.

void SetYRPin(int pin)

→ Setzt den Pin für die Y-Richtung auf pin.

void SetXRichtung(int r)

→ Setzt die X-Richtung: negativ = -1, positiv = 1.

void SetYRichtung(int r)

→ Setzt die Y-Richtung: negativ = -1, positiv = 1.

int GetXRichtung()

→ Gibt die X-Richtung zurück: -1 = negativ, 1 = positiv.

int GetYRichtung()

→ Gibt die Y-Richtung zurück: -1 = negativ, 1 = positiv.

int GetPixel()

→ Gibt die Umrechnungsrate der Steps und Pixel aus:  
106 => 1 Pixelbreite = 1cm.

void SetPixel(int p) → Setzt die Umrechnungsrate der Steps und Pixel:  
106 => 1 Pixelbreite = 1cm.

int SpritzeGetSpeed() → gibt die Geschwindigkeit der Spritze aus.  
void SpritzeSetSpeed(int v) → Setzt die Geschwindigkeit der Spritze auf v  
(0-128).

int SpritzeGetRichtung() → Gibt die Richtung der Spritze aus:  
-1 = Hoch, 1 = Runter, 0 = keine Richtung.  
void SpritzeSetRichtung(int r) → Setzt die Richtung der Spritze:  
-1 = Hoch, 1 = Runter, 0 = keine Richtung.

int SpritzeGetPinHoch() → Gibt den Pin der Spritze für „Hoch“ zurück.  
void SpritzeSetPinHoch(int pin) → Setzt den Pin der Spritze für „Hoch“.

int SpritzeGetPinRunter(); → Gibt den Pin der Spritze für „Runter“ zurück.  
void SpritzeSetPinRunter(int pin); → Setzt den Pin der Spritze für „Runter“.

int GetDruck() → Setzt die Spritzengeschwindigkeit während des Drucks.  
void SetDruck(int d) → Setzt die Spritzengeschwindigkeit für den Druck.