

Steuerkennlinie Haswing Protruar

Eingangsgröße: V_{Power} ; Ganzzahlenwert „int16“; $\mathbb{D} = \mathbb{Z} \wedge [-16384, 16384]$

Ausgangsgröße: V_{PWM} ; Ganzzahlenwert „uint8“; $\mathbb{D} = \mathbb{Z} \wedge [0, 255]$

Mathematische Funktion

$$V_{PWM}(V_{Power}) = \lfloor m \cdot V_{Power} + n \rfloor$$

$$V_{PWM}(V_{Power}) = \left\lfloor \frac{m_f}{2^{d_f}} V_{Power} + n \right\rfloor$$

$$V_{PWM}(V_{Power}) = \lfloor m_f \cdot V_{Power} \cdot 2^{-d_f} + n \rfloor$$

mit

m – Faktor

m_f – Faktor Divisor

d_f – Faktor 2^n Dividend

n – Offset

Kennlinien Segmentierung

- Segment 1: Vorwärtsfahrt; $\mathbb{D}_1 = \mathbb{Z} \wedge [112, 235]$
- Segment 2: Stillstand; $\mathbb{D}_2 = \mathbb{Z} \wedge [69, 111]$
- Segment 3: Rückwärtsfahrt $\mathbb{D}_3 = \mathbb{Z} \wedge [0, 68]$

Segment 1 Funktion:

$$m_1 = 7.53784 \cdot 10^{-3} \approx \frac{494}{2^{16}}$$

$$n_1 = 112$$

$$V_{PWM}(V_{Power}) = \lfloor 7.53784 \cdot 10^{-3} \cdot V_{Power} + 112 \rfloor$$

$$V_{PWM}(V_{Power}) = \lfloor 494 \cdot V_{Power} \cdot 2^{-16} + 112 \rfloor$$

Segment 2 Funktion:

$$m_2 = 335.94 \cdot 10^{-3} \approx \frac{344}{2^{10}}$$

$$n_2 = 91$$

$$V_{PWM}(V_{Power}) = \lfloor 335.94 \cdot 10^{-3} \cdot V_{Power} + 91 \rfloor$$

$$V_{PWM}(V_{Power}) = \lfloor 344 \cdot V_{Power} \cdot 2^{-10} + 91 \rfloor$$

Segment 3 Funktion:

$$m_3 = 2.6932 \cdot 10^{-3} \approx \frac{353}{2^{17}}$$

$$n_3 = 69$$

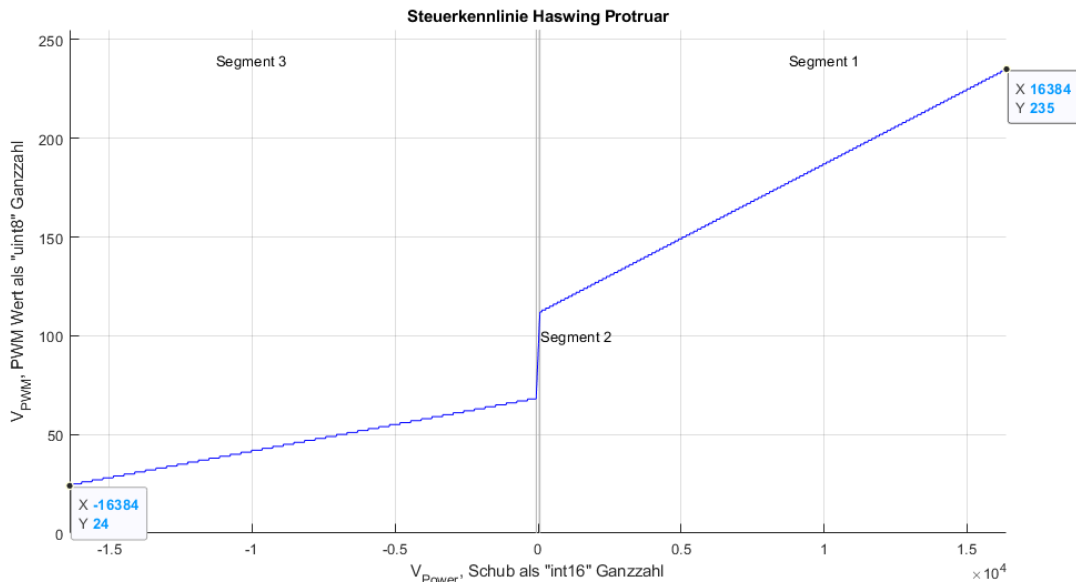
$$V_{PWM}(V_{Power}) = \lfloor 2.6932 \cdot 10^{-3} \cdot V_{Power} + 69 \rfloor$$

$$V_{PWM}(V_{Power}) = \lfloor 353 \cdot V_{Power} \cdot 2^{-17} + 69 \rfloor$$

Segmentfunktion:

$$V_{PWM}(V_{Power}) = \begin{cases} \left\lfloor \frac{494}{2^{16}} \cdot V_{Power} + 112 \right\rfloor, & \text{if } V_{Power} > 64 \\ \left\lfloor \frac{344}{2^{10}} \cdot V_{Power} + 91 \right\rfloor, & \text{if } V_{Power} \geq 64 \wedge V_{Power} \leq 64 \\ \left\lfloor \frac{344}{2^{10}} \cdot V_{Power} + 69 \right\rfloor, & \text{if } V_{Power} < 64 \end{cases}$$

Funktionsgraph



Beispiel C-Code

```
#include <stdint.h>
// Matrix mit den Parametern für die PWM-Kennlinie
// mzmnnMatrix[][]:={mul 0, shift right 0, add 0}, Segment 0
//                  {mul 1, shift right 1, add 1}, Segment 1
//                  {mul 2, shift right 2, add 2}, Segment 2
//                  {mul 3, shift right 3, add 3}, Segment 3
//                  };
int24_t const mzmnnMatrix[4][3] = {{0,0,92}, {494,16,112}, {344,10,91}, {353,17,69}};
int16_t iPower;
int24_t iPWPow24; // 24 Bit oder 32 Bit Zwischenwert
uint8_t uSegment;

iPower = get_Value(leftPower);

// Schublimitierung
iPower = MIN(iPower, 16384);
iPower = MAX(iPower, -16384);

iPWPow24 = (int24_t)(iPower);

// Berechne für den Motor über die Kennlinie den 8-bit PWM-Wert als
// Segmentweise Funktion
// Auswahl des Segmentes
if(iPower > 64)
    uSegment = 1;
else if((iPower <= 64)&&(iPower >= -64))
    uSegment = 2;
else if((iPower < -64))
```

```
        uSegment = 3;
else
    uSegment = 0;

// Ganzzahl Funktionswertberechnung
iPWMPower24 *= mzmnnMatrix[uSegment][0];
iPWMPower24 >>= mzmnnMatrix[uSegment][1];
iPWMPower24 += mzmnnMatrix[uSegment][2];

iPower = MIN(iPower, 255);
iPower = MAX(iPower, 0);

// Schreibe den PWM Wert in den Timer
TimerCounterBx_set_PWM_Value(leftEngine, (uint8_t)iPower);
```