

ACHTUNG Zeilennummern: Diesen File NICHT in die Arduino-IDE einlesen!

```
1  /*
2  SteuerSketch für INFINEON-Experimentier-Platine mit BTN8982TA
3  * erzeugt das PWM-Signal für die Platine
4  * Bedienung: zwei Schalter für Vorwärts/Rückwärts ON-OFF-ON
5  und für Schneller/Langamer (ON)-OFF-(ON)
6  * die HALT-Funktion fährt den Motor schnell herunter
7  * steuert Timer2 für PWM-Ausgabe auf Pins 3 und 11
8  * nach einem Beispielsketch des RobotPower-Teams  http://robotpower.com
9  * Version 10/07/2015  (C) Peter Bickel  http://Puydorat.fr
10 * dieses Programm ist freie Software
11 -----*/
12
13 /**** Einstellungen *****/
14 byte mode = 1;          // Timer-Einstellung (siehe unten)
15 char h_delay = 5;      // HALT:      { Zähl-
16 char s_delay = 30;     // SCHNELLER: { Geschwindig-
17 char l_delay = 30;     // LANGSAM:  { keitdigkeit
18 int countMin = 40;     // minimale Geschwindigkeit
19 int countMax = 255;    // maximale Geschwindigkeit
20 /*****/
21
22 // Vorwärts/Rückwärts-Teil
23 char VR_VPin = 5;      // Pin 5 = vorwärts  { Schalter-
24 char VR_RPin = 6;      // Pin 6 = rückwärts { Anschlüsse
25 char VR_V, VR_R;
26 char ENPin = 7;       // Pin 7 = inhibit --> Ein-/Ausschalten des
Boards
27
28 // Schneller/Langamer-Teil
29 int count;             // Zähler für PWM-Vergältnis
30 char SL_SPin = 2;     // Pin 2 = schneller  { Schalter-
31 char SL_LPin = 4;     // Pin 4 = langsamer  { Anschlüsse
32 char SL_S, SL_L;
33 void decrementCount (); // Fkt: zählt count auf countMin hinunter
34
35 // PWM-Teil
36 char PWMPin1 = 3;     // { Pins von
37 char PWMPin2 = 11;    // { Timer2
38
39 // -----
40 void setup() {
41 // Grenzwerte: Delay-Min = 5
42 if (h_delay < 5) h_delay = 5;
43 if (s_delay < 5) h_delay = 5;
44 if (l_delay < 5) h_delay = 5;
45
46 // Schneller/Langamer-Teil
47 pinMode (SL_SPin,INPUT_PULLUP); // InputPins: Pullup-Widerstände
einschalten
48 pinMode (SL_LPin,INPUT_PULLUP);
49
50 // Vorwärts/Rückwärts-Teil
51 pinMode (VR_VPin,INPUT_PULLUP); // InputPins: Pullup-Widerstände
einschalten
52 pinMode (VR_RPin,INPUT_PULLUP);
53
54 // PWM-Teil
55 pinMode(PWMPin1, OUTPUT); // OutputPins für Timer2
56 pinMode(PWMPin2, OUTPUT);
57 TCCR2B = TCCR2B & 0b11111000 | mode; // PWM-Frequenz einstellen
58 }
59
```

```

60 // -----
61 void loop() {
62 // Schneller/Langsamere-Teil
63   SL_S = digitalRead (SL_SPin); // Schneller-Pin lesen
64   SL_L = digitalRead (SL_LPin); // Langsamere-Pin lesen
65   analogWrite(PWMPin2, 0); // Gegenpin Null setzen
66
67   if (SL_S==LOW && count<countMax) { // aufwärts zählen = schneller
68     if (count == 0) count = countMin;
69     count++;
70     analogWrite (PWMPin1, count); // PWM-duty setzen
71     delay (s_delay); // warten
72   }
73
74   else if (SL_L==LOW && count > countMin) { // abwärtszählen = langsamer
75     count--;
76     if (count == countMin) count = 0;
77     analogWrite (PWMPin1, count); // PWM-duty setzen
78     delay (l_delay); // warten
79   }
80
81   // Vor/Rück-Teil
82   VR_V = digitalRead (VR_VPin); // vorwärts-Pin lesen
83   VR_R = digitalRead (VR_RPin); // rückwärts-Pin lesen
84   if (VR_V == LOW) { // vorwärts
85     PWMPin1 = 11;
86     PWMPin2 = 3;
87   }
88   else if (VR_R == LOW) { // rückwärts
89     PWMPin1 = 3;
90     PWMPin2 = 11;
91   }
92   else if (VR_V == HIGH && VR_R == HIGH) { // Schalter in Mittelstellung:
aus
93     decrementCount (); // zurückzählen auf countMin
94   }
95 }
96
97 // -----
98 // Zähler zurückfahren auf countMin
99 void decrementCount () {
100   while (count>countMin) { // Zähler nach countMin
zurückzählen
101     count--;
102     analogWrite (PWMPin1, count);
103     delay (h_delay); // warten
104   }
105   analogWrite (PWMPin1, 0); // PWM ausschalten
106 }
107
108 /*
109 Kommentar zur Timer-Programmierung
110 Basisfrequenz = 31'250Hz (abhängig vom Prozessortakt!)
111
112 mode divisor Frequenz
113 -----
114 1 1 31250
115 2 8 3906.25
116 3 32 976.5625
117 4 64 488.28125
118 5 128 244.140625
119 6 256 122.0703125
120 7 1024 30.517578125
121 -----*/

```

