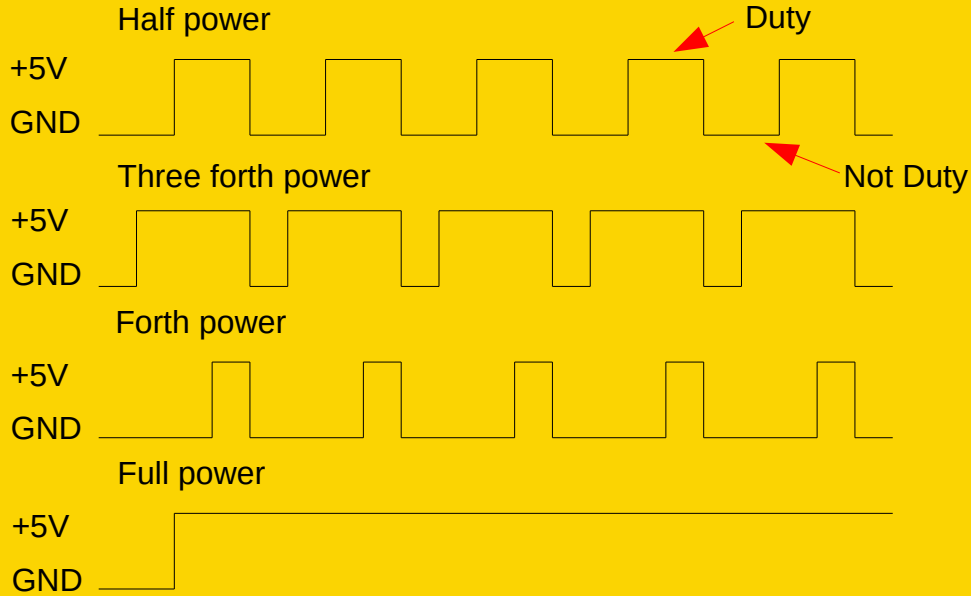


**Montage und Programmierung
eines Roboters für
den Hessen SolarCup
Disziplin: SolaRobot
Teil 2.6: Motor**

Von Charlotte und Andreas

PWM: Pulse Width Modulation

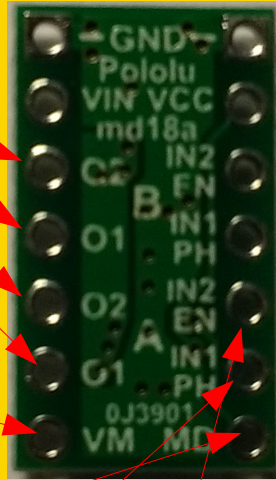


```
Board wählen

07_Motor.ino Drive_Functions.h Init.h Interrupt_Service_Routines.h USART_Functions.h

13 #define PWMA OCR0A //Overflow compare register A
14 #define PWMB OCR0B //Overflow compare register B
15
16 void Forward (unsigned char, unsigned char); //Direction with velocity A and B,for drivin
17 void Backward (unsigned char, unsigned char); //Direction with velocity A and B,for drivin
18 void Turn_right (unsigned char); //Rotate on the spot right, same velocity of the wheels
19 void Turn_left (unsigned char); //Rotate on the spot left, same velocity of the wheels
20 void Stop (void); //Stop all Motors
21
22 //Motors forward
23 void Forward (unsigned char velocity_A, unsigned char velocity_B) {
24     PWMA = velocity_A; //Pulswidth modulation, MotorA
25     PWMB = velocity_B; //MotorB
26     PORTB &= ~(1 << PORTB1); //MotorA
27     PORTB &= ~(1 << PORTB0); //MotorB, PORTB 0bxxxxxx00
28 }
29
30 //Motors backward
31 void Backward (unsigned char velocity_A, unsigned char velocity_B) {
32     PWMA = velocity_A; //Pulswidth modulation, MotorA
33     PWMB = velocity_B; //MotorB
34     PORTB |= (1 << PORTB1); //MotorA
35     PORTB |= (1 << PORTB0); //MotorB, PORTB 0bxxxxxx11
36 }
37
38 //Turn right on the spot
39 void Turn_left (unsigned char velocity) {
40     PWMA = velocity; //Pulswidth modulation, MotorA
41     PWMB = velocity; //MotorB
42     PORTB |= (1 << PORTB1); //MotorA
43     PORTB &= ~(1 << PORTB0); //MotorB, PORTB 0bxxxxxx10
44 }
45
46 //Turn left on the spot
47 void Turn_right (unsigned char velocity) {
48     PWMA = velocity; //Pulswidth modulation, MotorA
49     PWMB = velocity; //MotorB
50     PORTB &= ~(1 << PORTB1); //MotorA
51     PORTB |= (1 << PORTB0); //MotorB, PORTB 0bxxxxxx01
52 }
```

Motortreiber



Motorausgänge

IN2(B) In1(B)

Steuer-
spannungs-
eingänge

Motorspannung

Mode	PH(A)	EN (A)	Motor
High	X	Low	Brake Low
High	Low	PWM	Forward/Brake
High	High	PWM	Backward/Brake

X = Don't care
High = +5V
Low = GND

07_Motor - Drive_Functions.h | Arduino IDE 2.3.4

Datei Bearbeiten Sketch Werkzeuge Hilfe

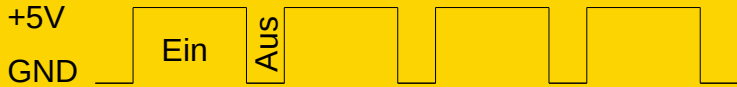
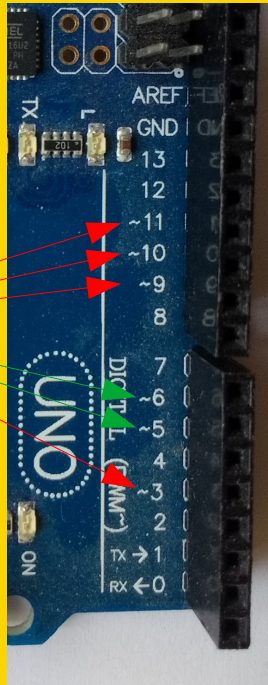
Board wählen

07_Motor.ino Drive_Functions.h Init.h Interrupt_Service_Routines.h USART_Functions.h

```

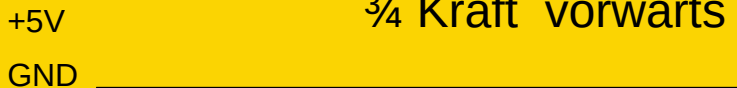
13 #define PWMA OCR0A //Overflow compare register A
14 #define PWMB OCR0B //Overflow compare register B
15
16 void Forward (unsigned char, unsigned char); //Direction with velocity A and B,for drivin
17 void Backward (unsigned char, unsigned char); //Direction with velocity A and B,for drivin
18 void Turn_right (unsigned char); //Rotate on the spot right, same velocity of the wheels
19 void Turn_left (unsigned char); //Rotate on the spot left, same velocity of the wheels
20 void Stop (void); //Stop all Motors
21
22 //Motors forward
23 void Forward (unsigned char velocity_A, unsigned char velocity_B) {
24     PWMA = velocity_A; //Pulswidth modulation, MotorA
25     PWMB = velocity_B; //MotorB
26     PORTB &= ~(1 << PORTB1); //MotorA
27     PORTB &= ~(1 << PORTB0); //MotorB, PORTB 0bxxxxxx00
28 }
29
30 //Motors backward
31 void Backward (unsigned char velocity_A, unsigned char velocity_B) {
32     PWMA = velocity_A; //Pulswidth modulation, MotorA
33     PWMB = velocity_B; //MotorB
34     PORTB |= (1 << PORTB1); //MotorA
35     PORTB |= (1 << PORTB0); //MotorB, PORTB 0bxxxxxxx11
36 }
37
38 //Turn right on the spot
39 void Turn_left (unsigned char velocity) {
40     PWMA = velocity; //Pulswidth modulation, MotorA
41     PWMB = velocity; //MotorB
42     PORTB |= (1 << PORTB1); //MotorA
43     PORTB &= ~(1 << PORTB0); //MotorB, PORTB 0bxxxxxxx10
44 }
45
46 //Turn left on the spot
47 void Turn_right (unsigned char velocity) {
48     PWMA = velocity; //Pulswidth modulation, MotorA
49     PWMB = velocity; //MotorB
50     PORTB &= ~(1 << PORTB1); //MotorA
51     PORTB |= (1 << PORTB0); //MotorB, PORTB 0bxxxxxxx01
52 }
    
```

PWM Ausgänge
beim Uno besser zu sehen



Pin 5

$\frac{3}{4}$ Kraft vorwärts

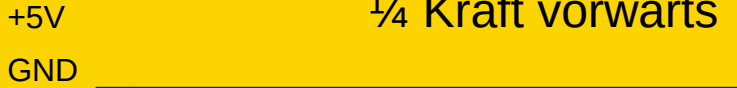


Port B 0



Pin 6

$\frac{1}{4}$ Kraft vorwärts



Port B 1

```

07_Motor - Drive_Functions.h | Arduino IDE 2.3.4
Datei Bearbeiten Sketch Werkzeuge Hilfe

Board wählen

07_Motor.ino Drive_Functions.h Init.h Interrupt_Service_Routines.h USART_Functions.h

13 #define PWMA OCR0A //Overflow compare register A
14 #define PWMB OCR0B //Overflow compare register B
15
16 void Forward (unsigned char, unsigned char); //Direction with velocity A and B,for drivin
17 void Backward (unsigned char, unsigned char); //Direction with velocity A and B,for drivin
18 void Turn_right (unsigned char); //Rotate on the spot right, same velocity of the wheels
19 void Turn_left (unsigned char); //Rotate on the spot left, same velocity of the wheels
20 void Stop (void); //Stop all Motors
21
22 //Motors forward
23 void Forward (unsigned char velocity_A, unsigned char velocity_B) {
24     PWMA = velocity_A; //Pulwidth modulation, MotorA
25     PWMB = velocity_B; //MotorB
26     PORTB &= ~(1 << PORTB1); //MotorA
27     PORTB &= ~(1 << PORTB0); //MotorB, PORTB 0bxxxxxx00
28 }
29
30 //Motors backward
31 void Backward (unsigned char velocity_A, unsigned char velocity_B) {
32     PWMA = velocity_A; //Pulwidth modulation, MotorA
33     PWMB = velocity_B; //MotorB
34     PORTB |= (1 << PORTB1); //MotorA
35     PORTB |= (1 << PORTB0); //MotorB, PORTB 0bxxxxxx11
36 }
37
38 //Turn right on the spot
39 void Turn_left (unsigned char velocity) {
40     PWMA = velocity; //Pulwidth modulation, MotorA
41     PWMB = velocity; //MotorB
42     PORTB |= (1 << PORTB1); //MotorA
43     PORTB &= ~(1 << PORTB0); //MotorB, PORTB 0bxxxxxx10
44 }
45
46 //Turn left on the spot
47 void Turn_right (unsigned char velocity) {
48     PWMA = velocity; //Pulwidth modulation, MotorA
49     PWMB = velocity; //MotorB
50     PORTB &= ~(1 << PORTB1); //MotorA
51     PORTB |= (1 << PORTB0); //MotorB, PORTB 0bxxxxxx01
52 }
    
```

Aufgaben:

1. Lasst den Roboter mit halber Kraft fahren und dann mit voller Kraft.

Was müsst ihr dafür ändern?

2. Lasst den Roboter längere Fahrten und kürzere unternehmen.

Was müsst ihr dafür ändern?

3. Lasst den Roboter eine Links- oder Rechtskurve fahren.

Welche Funktion braucht ihr dafür?

Kommentiert die anderen aus!

4. Lasst den Roboter mit halber Kraft ein Quadrat mit der Kantenlänge 20cm fahren.

Welche Funktionen braucht ihr dafür?



```
07_Motor | Arduino IDE 2.3.4
Datei Bearbeiten Sketch Werkzeuge Hilfe

Arduino Nano

07_Motor.ino Drive_Functions.h Init.h Interrupt_Service_Routines.h USART_Functions.h

80 | Stop();
81 | }
82 |
83 | //Main routine
84 void loop(void) {
85 | sei(); //Enable all interrupts
86 | //cli(); //Clear all interrupts
87 | /**Testschleife*****
88 | //Mögliche Fahrbefehle:
89 | //Forward(V,V), Backward(V,V), Turn_left(V), Turn_right(V), Stop()
90 | //V = Geschwindigkeit (Velocity)
91 | while (0) { //Testschleife: ein (1), aus (0)
92 | | Forward(128, 128);
93 | }
94 | /**Ende-Testschleife*****
95 | _delay_ms(1000);
96 | while (US_Time_L < 10) { //Bleibt solange stehen, bis die Bahn frei ist
97 | Stop();
98 | }
99 | while (1) { //Endlosschleife
100 | | Data_Visualizer(); //Define the values to be displayed and transmit them
101 | | _delay_us(1);
102 | | Forward(192,192);
103 | | _delay_ms(500);
104 | | Backward(192,192);
105 | | _delay_ms(500);
106 | | Turn_left(192);
107 | | _delay_ms(500);
108 | | Turn_right(192);
109 | | _delay_ms(500);
110 | | Stop();
111 | | _delay_ms(1000);
112 | }
113 | }
114 | /*****

Ausgabe
Der Sketch verwendet 1530 Bytes (4%) des Programmspeicherplatzes. Das Maximum sind 30720 Bytes.
Globale Variablen verwenden 43 Bytes (2%) des dynamischen Speichers, 2005 Bytes für lokale Vari
```

Aufgabe:

Kombiniert den Ultraschallsensor mit dem Fahren des Roboters.

Lasst ihn auf ein Hindernis zufahren.

Ca. 20cm vor dem Hindernis stoppen.

Eine halbe Sekunde zurück fahren.

Und wieder auf das Hindernis zufahren.

Hilfe:

Benutzt das Grundprogramm „Motor“ und modifiziert es.

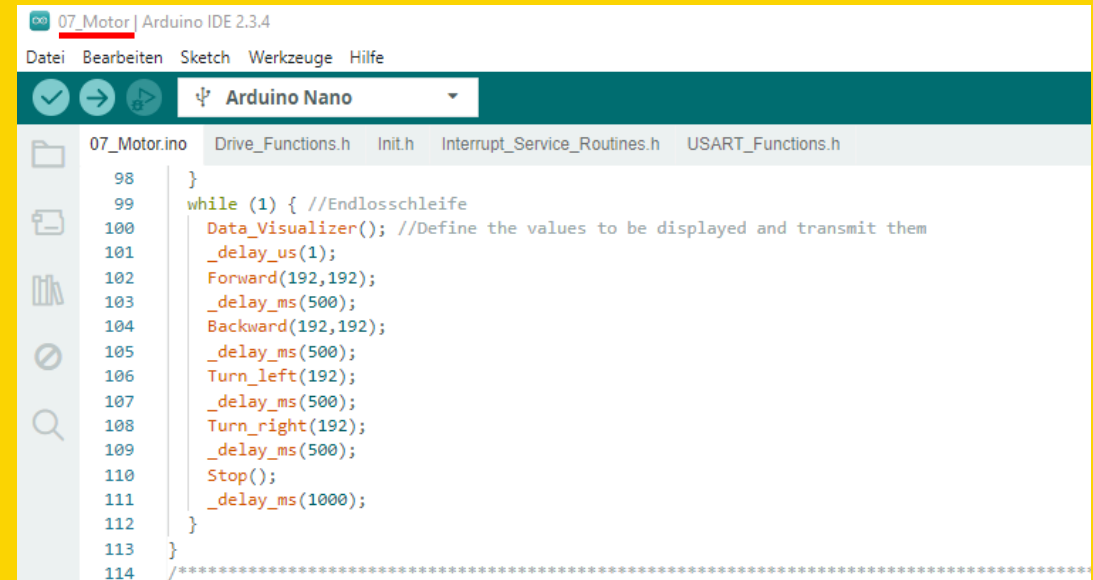
Ihr müsst den Wert von US_Time_L abfragen.

Falls der größer ist als „20cm“, soll der

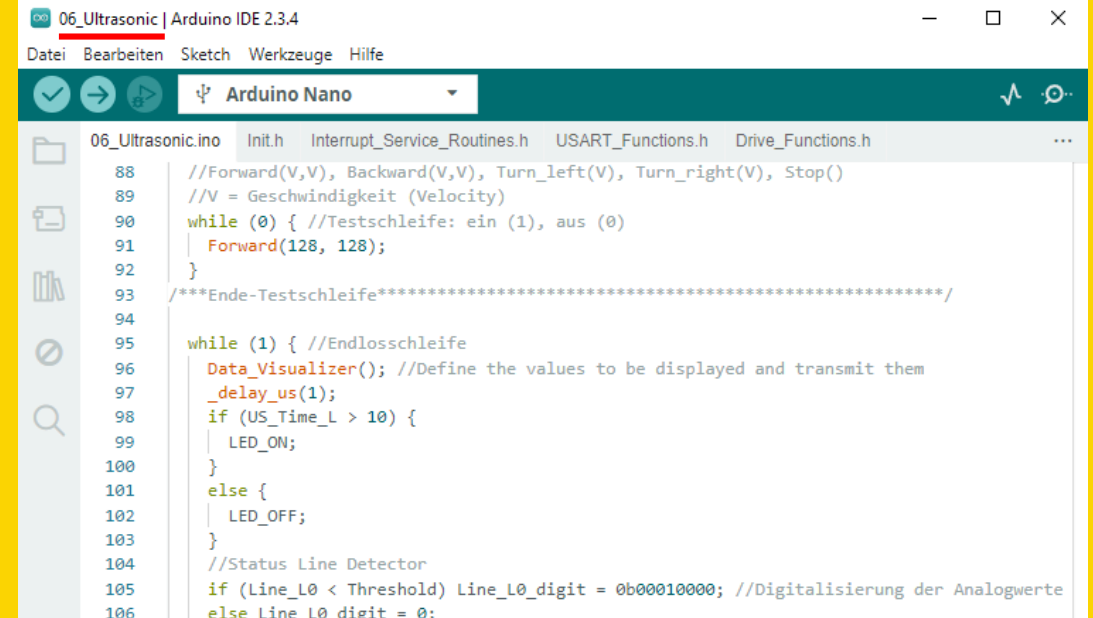
Roboter vorwärts fahren.

Sonst soll er 500 Millisekunden rückwärts

fahren.



```
07_Motor.ino | Arduino IDE 2.3.4
Datei Bearbeiten Sketch Werkzeuge Hilfe
Arduino Nano
07_Motor.ino Drive_Functions.h Init.h Interrupt_Service_Routines.h USART_Functions.h
98 }
99 while (1) { //Endlosschleife
100   Data_Visualizer(); //Define the values to be displayed and transmit them
101   _delay_us(1);
102   Forward(192,192);
103   _delay_ms(500);
104   Backward(192,192);
105   _delay_ms(500);
106   Turn_left(192);
107   _delay_ms(500);
108   Turn_right(192);
109   _delay_ms(500);
110   Stop();
111   _delay_ms(1000);
112 }
113 }
114 /*****
```



```
06_Ultrasonic.ino | Arduino IDE 2.3.4
Datei Bearbeiten Sketch Werkzeuge Hilfe
Arduino Nano
06_Ultrasonic.ino Init.h Interrupt_Service_Routines.h USART_Functions.h Drive_Functions.h ...
88 //Forward(V,V), Backward(V,V), Turn_left(V), Turn_right(V), Stop()
89 //V = Geschwindigkeit (Velocity)
90 while (0) { //Testschleife: ein (1), aus (0)
91   Forward(128, 128);
92 }
93 /****Ende-Testschleife*****/
94
95 while (1) { //Endlosschleife
96   Data_Visualizer(); //Define the values to be displayed and transmit them
97   _delay_us(1);
98   if (US_Time_L > 10) {
99     LED_ON;
100   }
101   else {
102     LED_OFF;
103   }
104   //Status Line Detector
105   if (Line_L0 < Threshold) Line_L0_digit = 0b00010000; //Digitalisierung der Analogwerte
106   else Line_L0_digit = 0;
```

Montage und Programmierung
eines Roboters für
den Hessen SolarCup
Disziplin: SolaRobot
Teil 2.7: Line Follower

Von Charlotte und Andreas