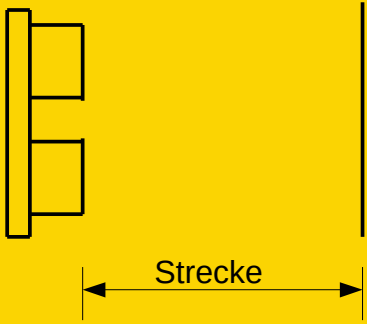


Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Arduino Nano
Teil 2.5: Ultraschall

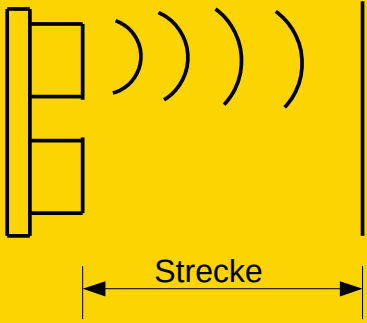
Ultraschall- Sensor



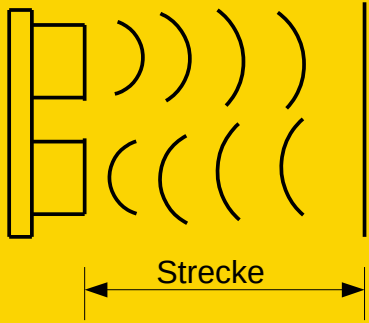
Ultraschall-Sensor



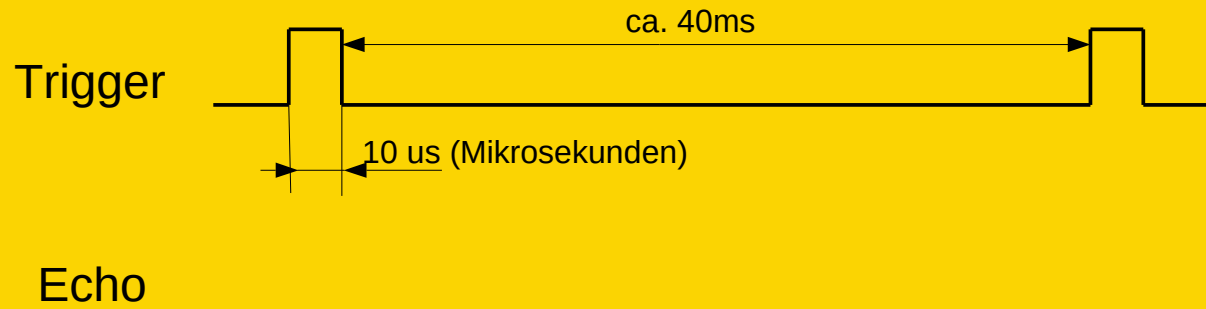
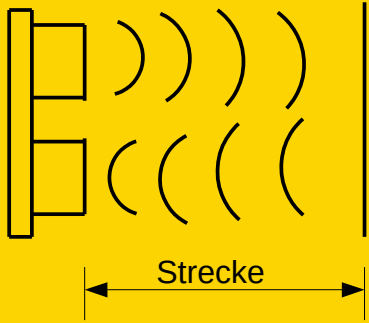
Ultraschall-Sensor



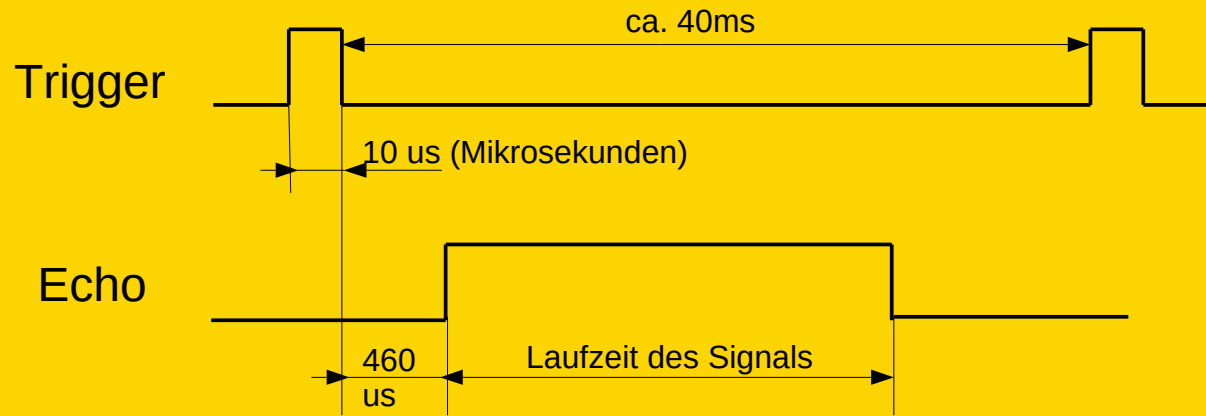
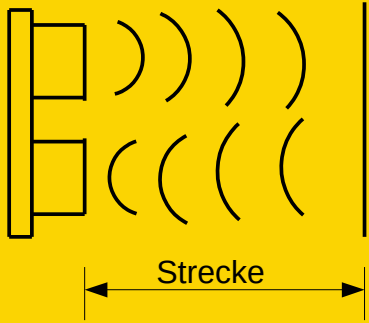
Ultraschall-Sensor



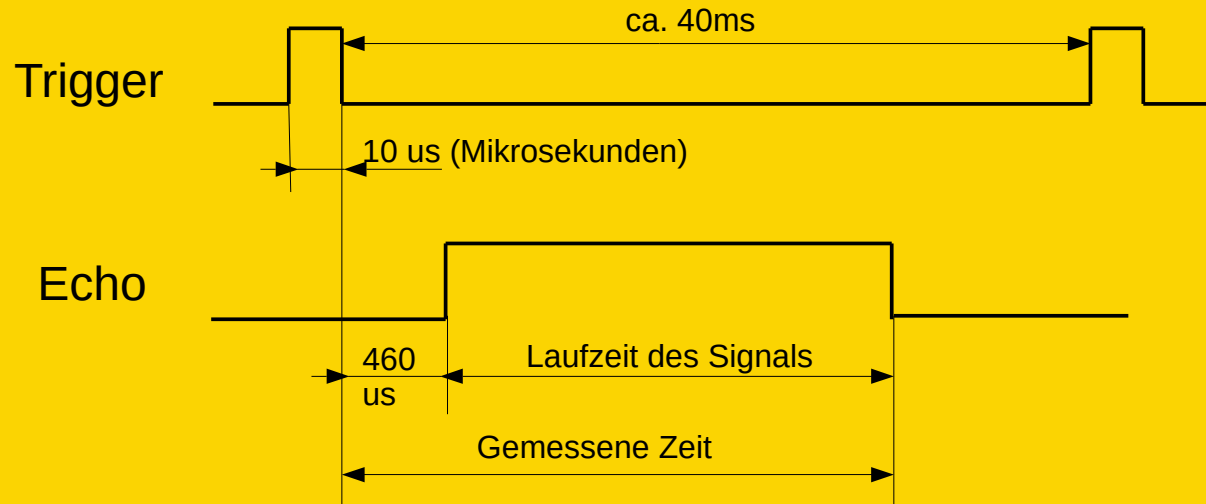
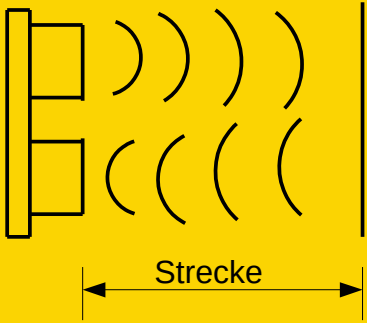
Ultraschall-Sensor



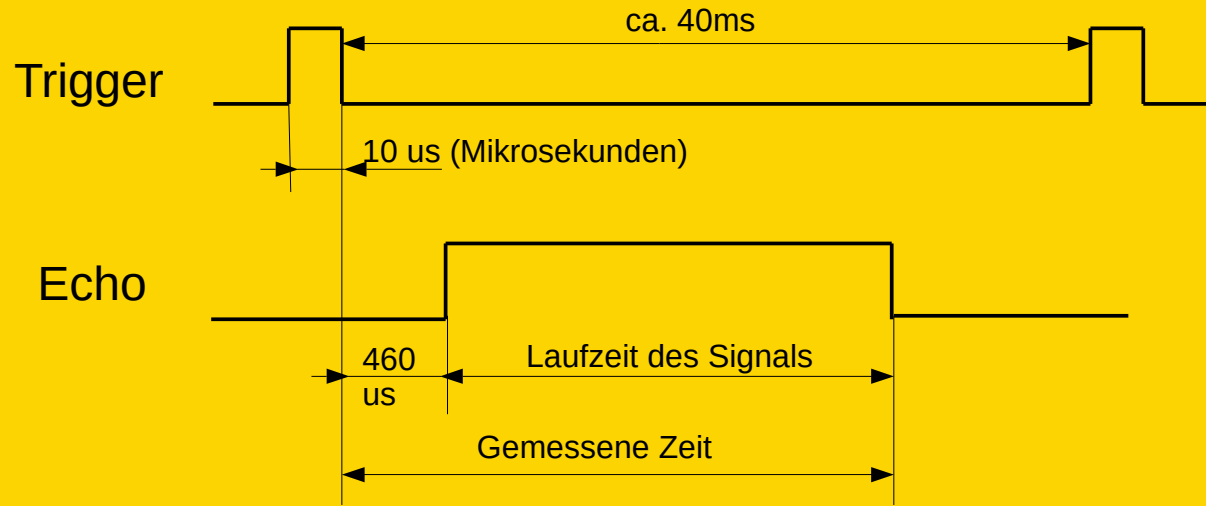
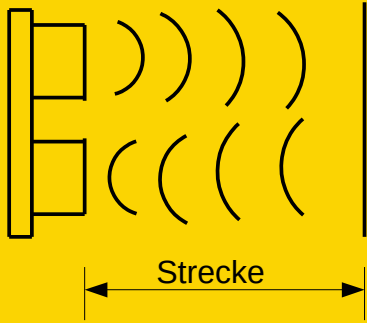
Ultraschall-Sensor



Ultraschall-Sensor

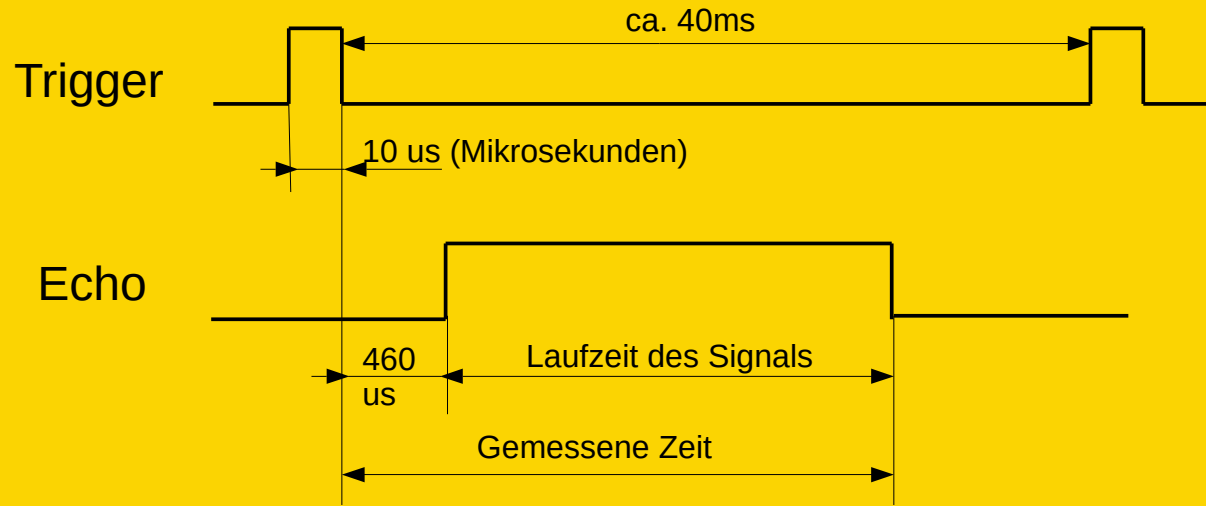
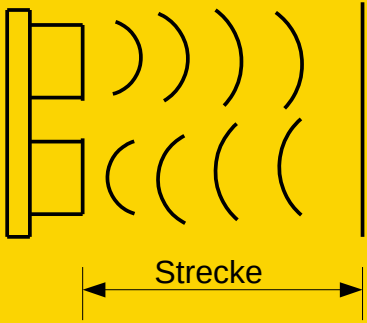


Ultraschall-Sensor



Schallgeschwindigkeit
 $343\text{m/s} = 1236\text{km/h}$
ca. 3s für 1km
Blitz - Donner

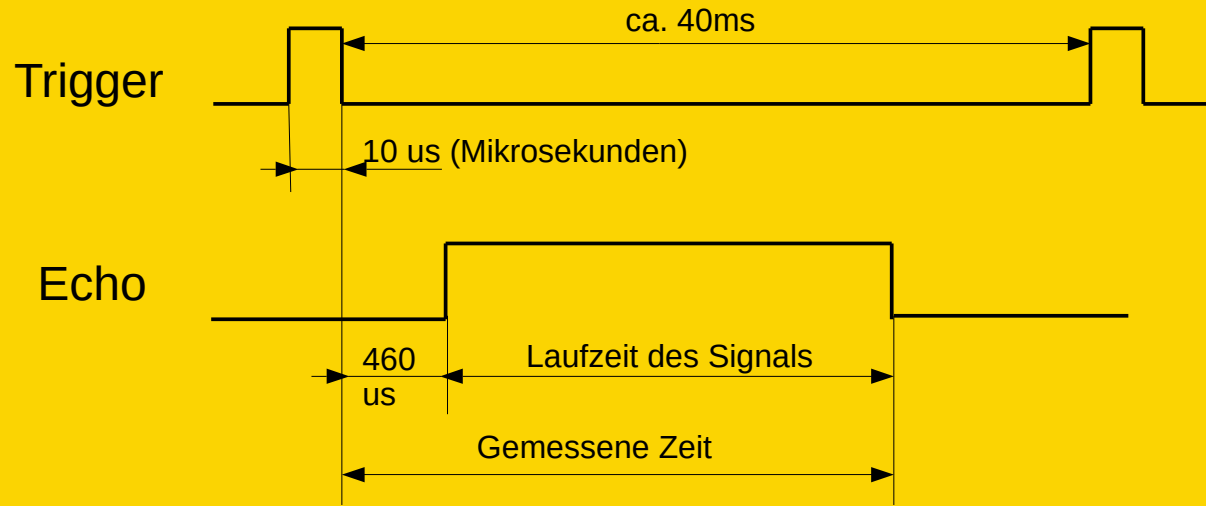
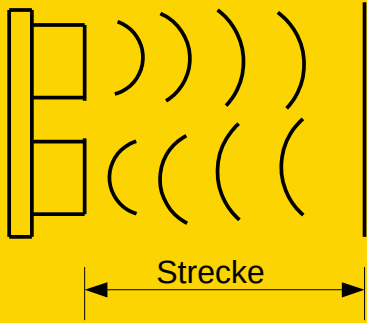
Ultraschall-Sensor



Schallgeschwindigkeit
 $343\text{m/s} = 1236\text{km/h}$
ca. 3s für 1km
Blitz - Donner

Rechenbeispiel:
Strecke = 10cm → 0,1m

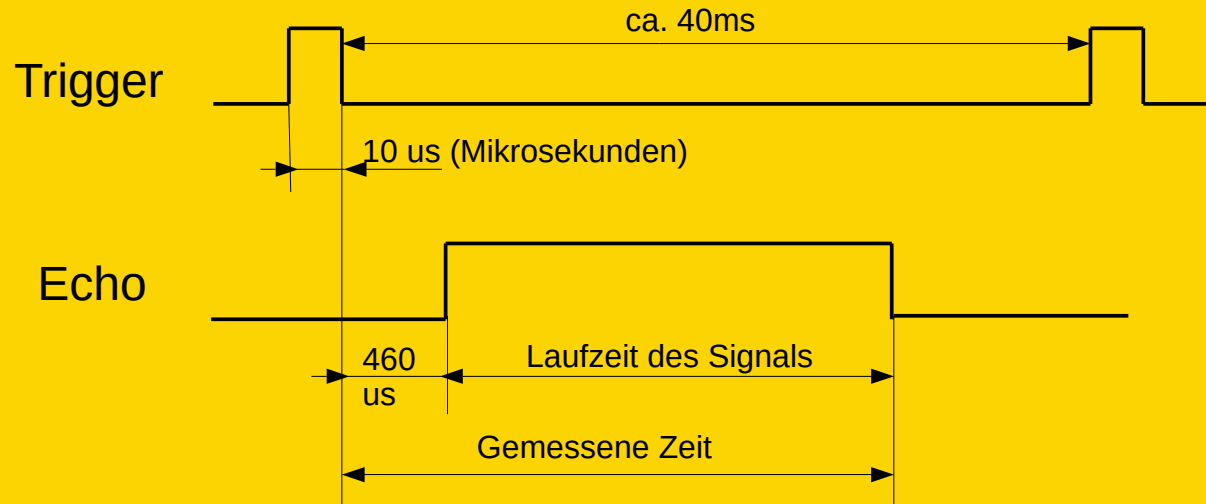
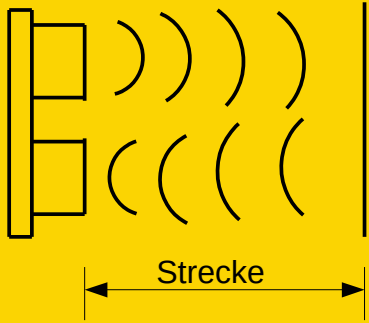
Ultraschall-Sensor



Schallgeschwindigkeit
343m/s = 1236km/h
ca. 3s für 1km
Blitz - Donner

Rechenbeispiel:
Strecke = 10cm → 0,1m
 $0,1\text{m} / 343\text{m/s} = 290\mu\text{s} \rightarrow 29\mu\text{s/cm}$

Ultraschall-Sensor



Schallgeschwindigkeit
343m/s = 1236km/h
ca. 3s für 1km
Blitz - Donner

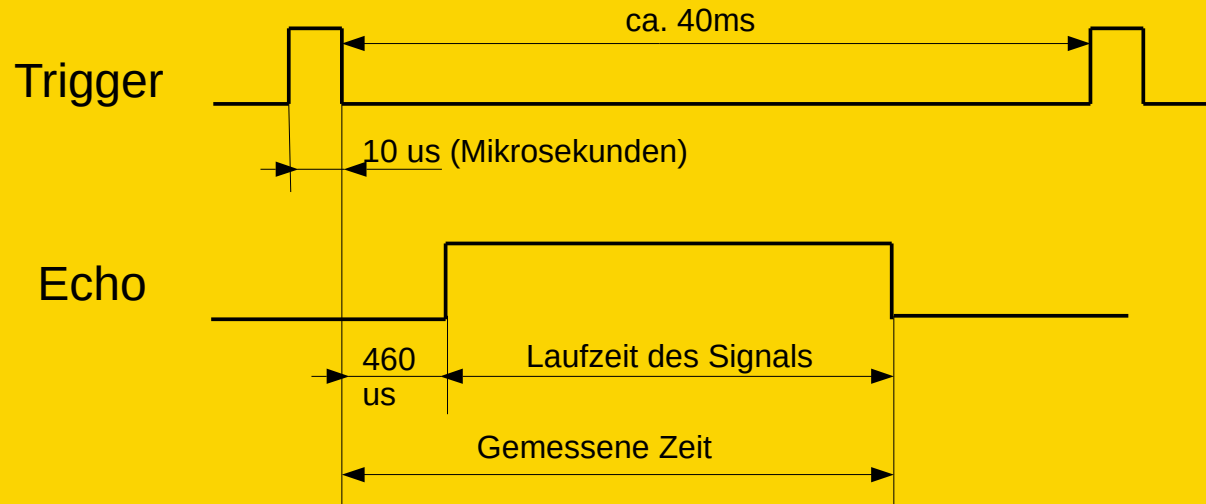
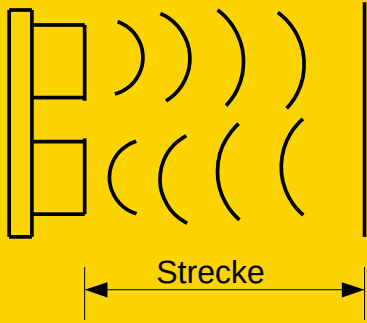
Rechenbeispiel:

Strecke = 10cm → 0,1m

$0,1\text{m} / 343\text{m/s} = 290\text{us} \rightarrow 29\text{us/cm}$

$290\text{us} * 2 = 580\text{us}$ (läuft 2mal die Strecke)

Ultraschall-Sensor



Schallgeschwindigkeit
343m/s = 1236km/h
ca. 3s für 1km
Blitz - Donner

Rechenbeispiel:

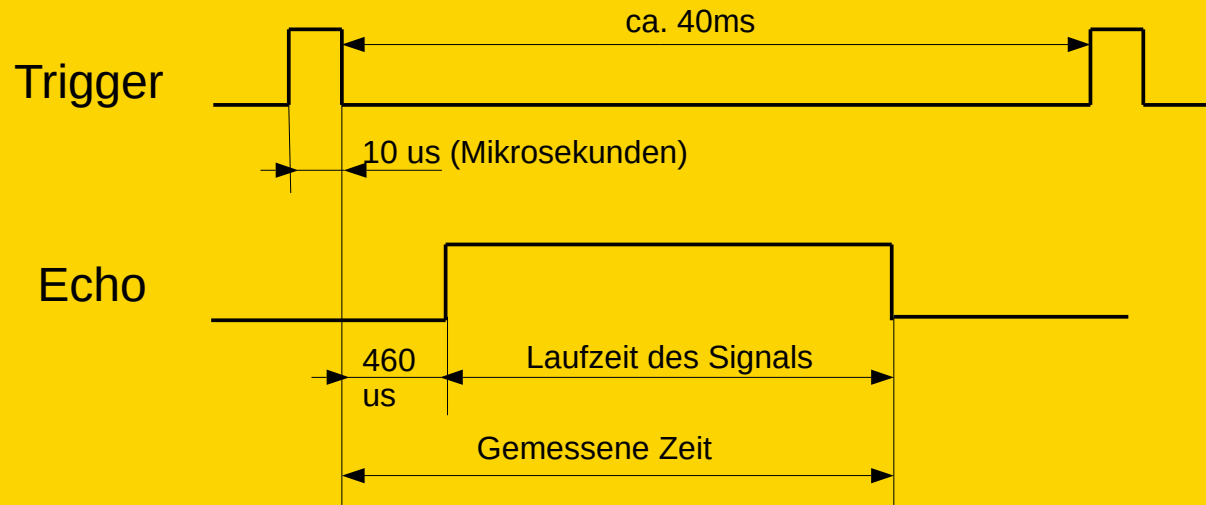
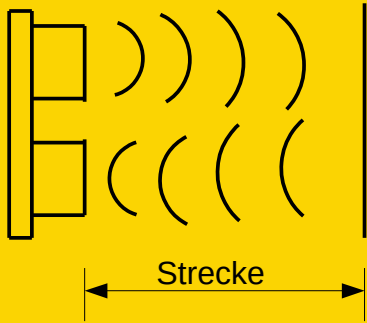
Strecke = 10cm → 0,1m

$0,1\text{m} / 343\text{m/s} = 290\text{us} \rightarrow 29\text{us/cm}$

$290\text{us} * 2 = 580\text{us}$ (läuft 2mal die Strecke)

$580\text{us} + 460\text{us} = 1040\text{us}$

Ultraschall-Sensor

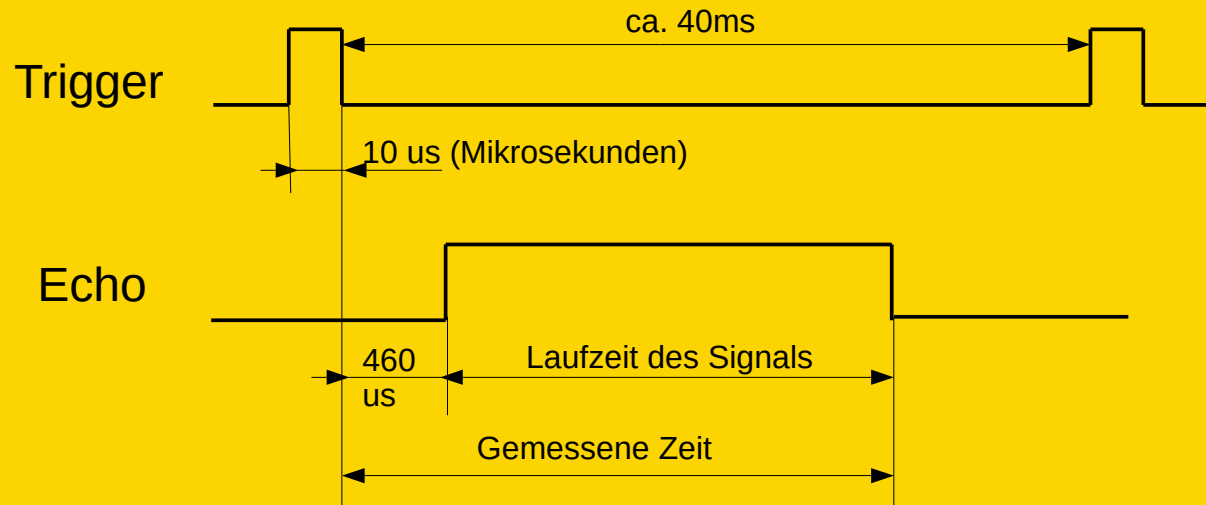
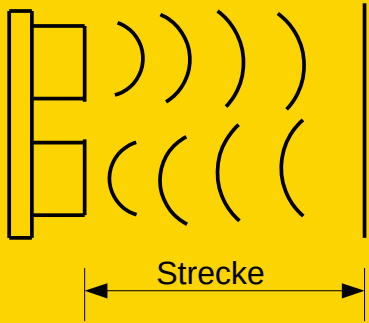


Schallgeschwindigkeit
 $343\text{m/s} = 1236\text{km/h}$
ca. 3s für 1km
Blitz - Donner

Rechenbeispiel:
Strecke = 10cm → 0,1m
 $0,1\text{m} / 343\text{m/s} = 290\mu\text{s} \rightarrow 29\mu\text{s/cm}$
 $290\mu\text{s} * 2 = 580\mu\text{s}$ (läuft 2mal die Strecke)
 $580\mu\text{s} + 460\mu\text{s} = 1040\mu\text{s}$

Taktfrequenz **16MHz**
Timer1-Prescaler 1:8

Ultraschall-Sensor



Schallgeschwindigkeit
343m/s = 1236km/h
ca. 3s für 1km
Blitz - Donner

Rechenbeispiel:

Strecke = 10cm → 0,1m

0,1m / 343m/s = 290us → 29us/cm

290us * 2 = 580us (läuft 2mal die Strecke)

580us + 460us = 1040us

Taktfrequenz **16MHz**

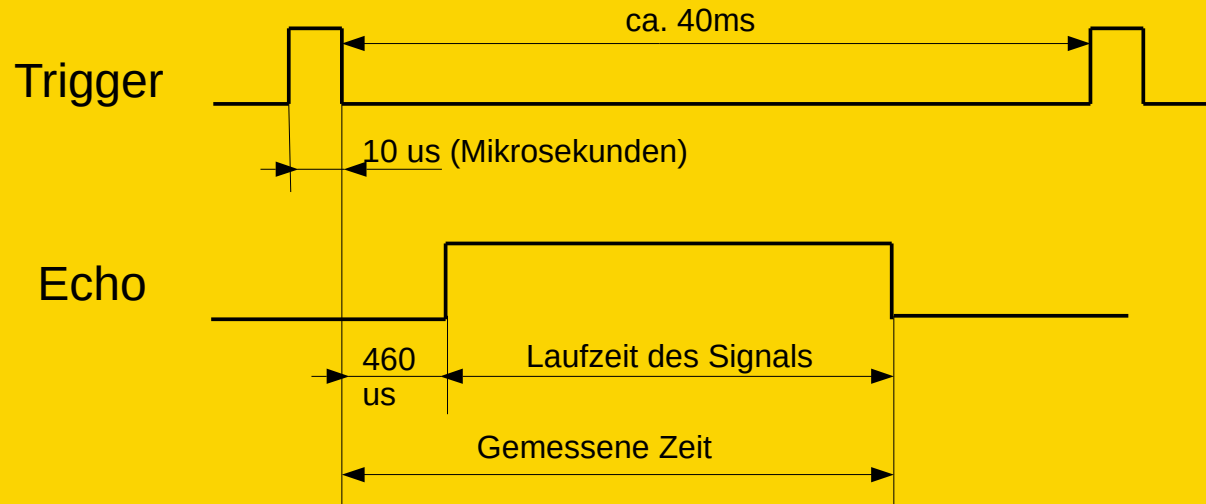
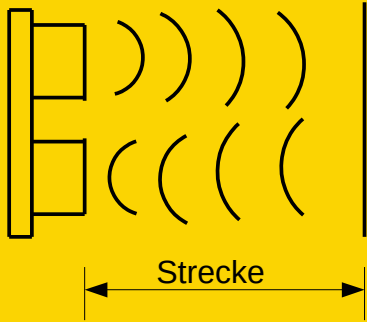
Timer1-Prescaler 1:8, alle 500us um 1 erhöht

1040 * 2 = 2080_{dez} → 00001000 00100000_{bin}

Die oberen 8 Bits → 8

HEX	8
DEC	8
OCT	10
BIN	<u>1000</u>

Ultraschall-Sensor



Schallgeschwindigkeit
 $343\text{m/s} = 1236\text{km/h}$
 ca. 3s für 1km
 Blitz - Donner

Rechenbeispiel:

Strecke = 10cm → 0,1m

$0,1\text{m} / 343\text{m/s} = 290\text{us} \rightarrow 29\text{us/cm}$

$290\text{us} * 2 = 580\text{us}$ (läuft 2mal die Strecke)

$580\text{us} + 460\text{us} = 1040\text{us}$

20cm: $580\text{us} * 2 + 460\text{us} = 1620 * 2 = 3240 \rightarrow$ obere 8 Bits → 12

30cm: $290\text{us} * 3 = 870\text{us} * 2 + 460\text{us} = 2200 * 2 = 4400 \rightarrow$ obere 8 Bits → 17

Taktfrequenz **16MHz**

Timer1-Prescaler 1:8, alle 500us um 1 erhöht

$1040 * 2 = 2080_{\text{dez}} \rightarrow 00001000\ 00100000_{\text{bin}}$

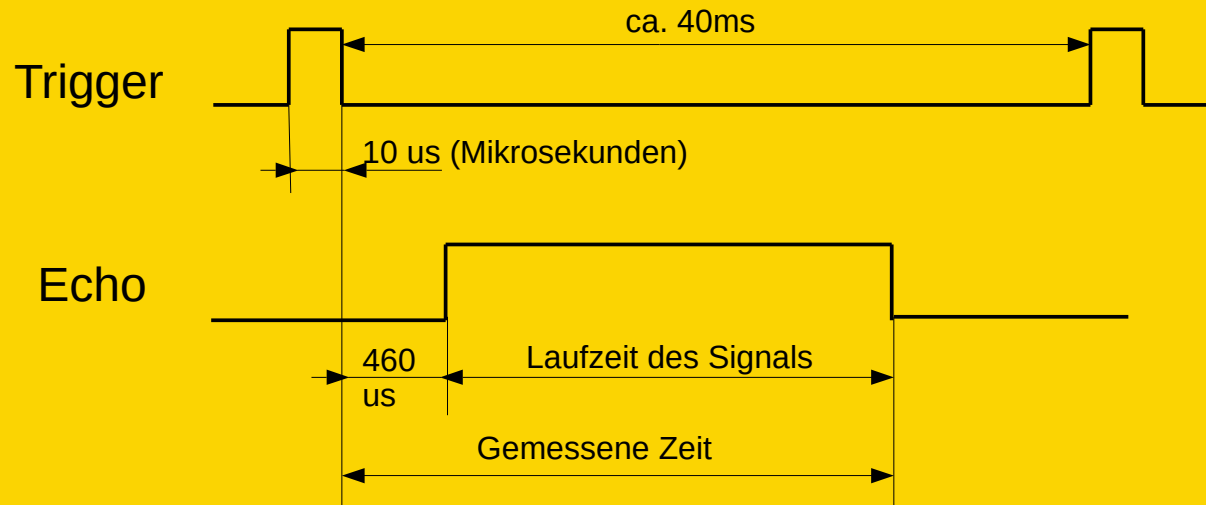
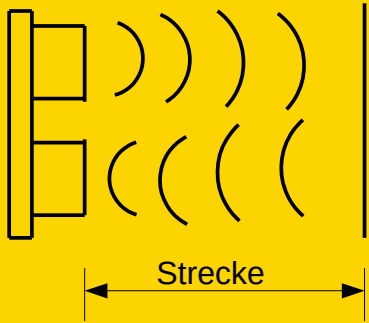
Die oberen 8 Bit → 8

Bei 10cm wird 8 angezeigt

Bei 20cm wird 12 angezeigt

Bei 30cm wird 17 angezeigt

Ultraschall-Sensor



Schallgeschwindigkeit
 $343\text{m/s} = 1236\text{km/h}$
 ca. 3s für 1km
 Blitz - Donner

Rechenbeispiel:

Strecke = 10cm \rightarrow 0,1m

$0,1\text{m} / 343\text{m/s} = 290\text{us} \rightarrow 29\text{us/cm}$

$290\text{us} * 2 = 580\text{us}$ (läuft 2mal die Strecke)

$580\text{us} + 460\text{us} = 1040\text{us}$

Taktfrequenz **16MHz**

Timer1-Prescaler 1:8, alle 500us um 1 erhöht

$1040 * 2 = 2080_{\text{dez}} \rightarrow 00001000\ 00100000_{\text{bin}}$

Die oberen 8 Bit \rightarrow 8

Bei 10cm wird 8 angezeigt

Bei 20cm wird 12 angezeigt

Bei 30cm wird 17 angezeigt

Bei 40cm wird ?? angezeigt

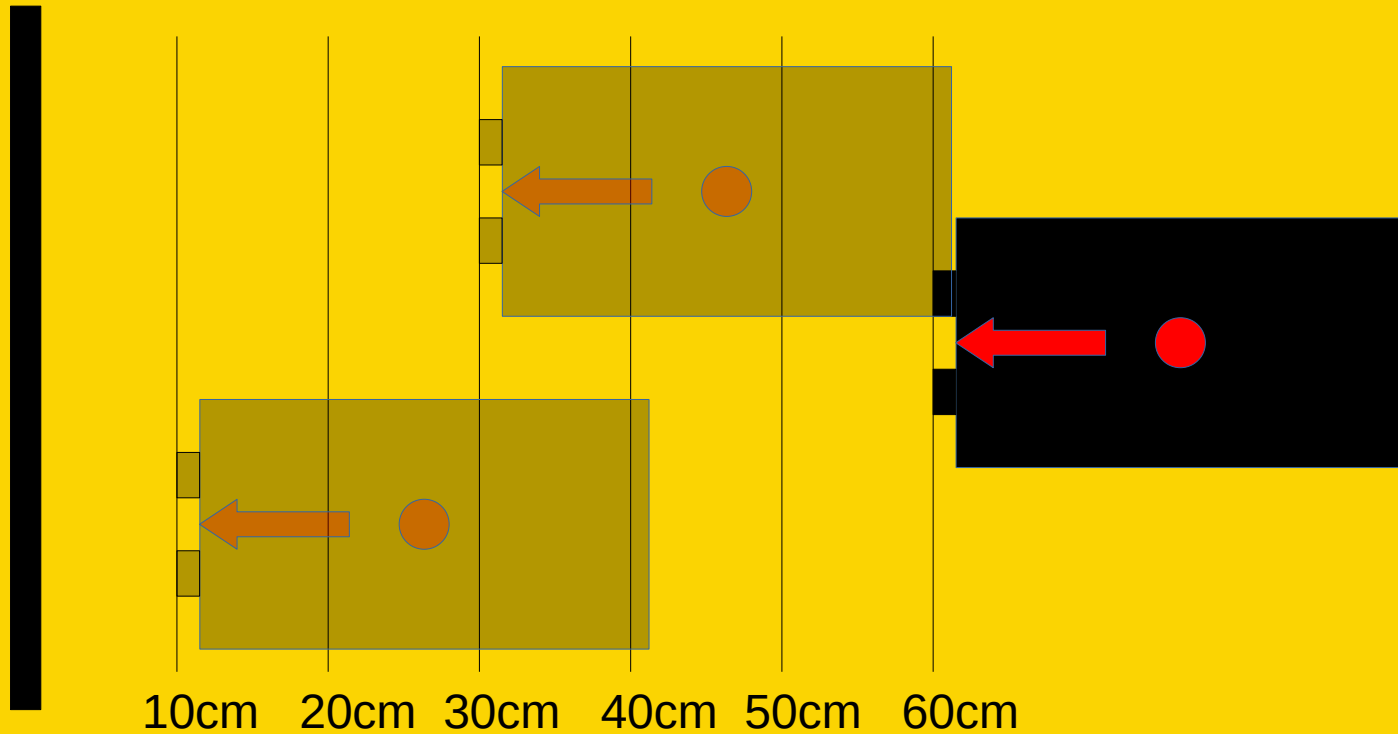
Bei 50cm wird ?? angezeigt

Bei 60cm wird ?? angezeigt

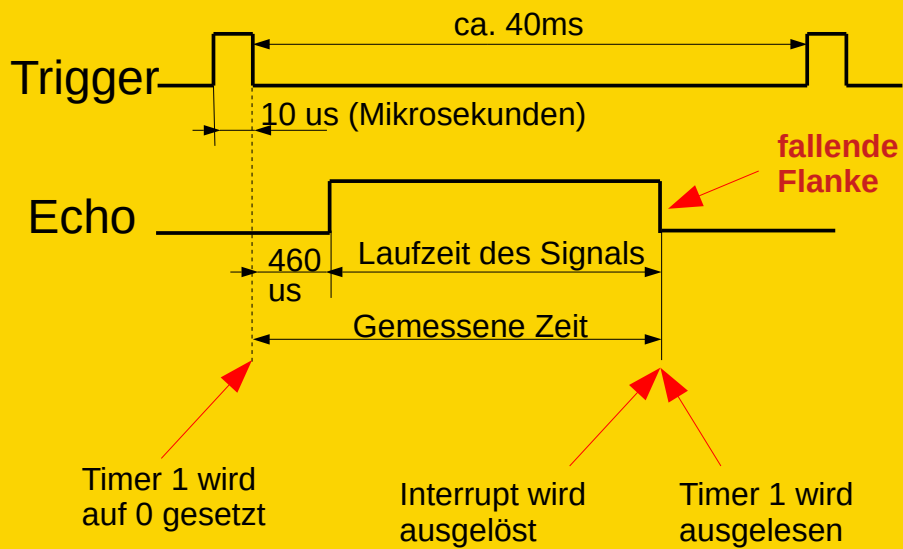
Aufgabe: Ausrechnen, welche Zahl angezeigt wird bei 40, 50 und 60cm.

Dann mit Hilfe des Sensors nachprüfen.

Ultraschall-Sensor



Abstand[cm]	US Wert
10	
20	
30	
40	
50	
60	



Advanced Mode Quick Launch (Ctrl+Q)

File Edit View VAssistX ASF Project Build Debug Tools Window Help

Debug Debug Browser

Hex Arduino

UltraSonic.cpp Init.h USART_Functions.h Interrupt_Service_Routines.h

Interrupt_Service_Routines.t C:\Users\User\Desktop\Tutorials\Atmel\Software\Elegoo_05_US\Elegoo_05_US\Interrupt_Service

```

17 //To time several works
18 ISR(TIMER2_OVF_vect) { //Prescaler TMR2 1:8 = Interrupt every 128 us
19     //Toggeling the LED, just to see, if the MC is working
20     T2ck++; //Counter TMR2 OVF
21     if(T2ck == 2000) { //Every 250ms
22         T2ck = 0; //Don't forget!!
23         // LED_TOGGLE; //See above
24     }
25     //Ultrasonic distance
26     Counter_US++; //256 * 128us = 32768us
27     if (!Counter_US) { //Approx. every 32ms
28         //LED_TOGGLE; //for testing
29         PORTD |= 0b00010000; //Trigger US, HI:
30         _delay_us(10); //10 microseconds
31         PORTD &= 0b11101111; //Trigger US, LO)
32         //First you have to write TCNT1H, before writing TCNT1L!!!
33         TCNT1H = 0; //Timer 1 set to zero...
34         TCNT1L = 0; //...to quantify the distance of the obstacle
35         //LED_ON; //Tests
36     }
37 }
38
39 //Interrupt on falling edge at RD0, US sensor
40
41 //Interrupt settings
42 //External
43 //Internal
44
45 ISR(INT0_vect) {
46     //LED_ON; //Test
47     char junk; //Only to read TCNT1L
48     junk = TCNT1L; //We have to read the low Byte first, only then we could read TCNT1H, but we do
49     US_Time = TCNT1H; //runtime of the US signal, HI part
50     TCNT1H = 0; //Reset TMR1
51     TCNT1L = 0; //Reset
52     //16MHz Prescaler 1:8 -> 2MHz TMR1 Counter
53     //LED_OFF; //Test
54 }
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

DDRC = 0b00100000; //00n.n.,1US-Trig,0Line-R,00D
 DDRD = 0b01101000; //0Color-1,11PWMA&B,0Line-M,1
 //Interrupt settings
 //External
 EICRA = 0b00001010; //INT0, INT1, falling edge
 EIMSK = 0b00000001; //Enable INT0, disable INT1
 //Internal

Output Error List

Ready Ln 14 Col 1 Ch 1 INS

if:

Falls Bedingung erfüllt ist, mache etwas

else:

Sonst mache etwas anderes

Syntax:

```
if(Bedingung) {  
    Anweisungen  
}
```

```
else {  
    Andere Anweisungen  
}
```

Bedingung ist ein Wert.

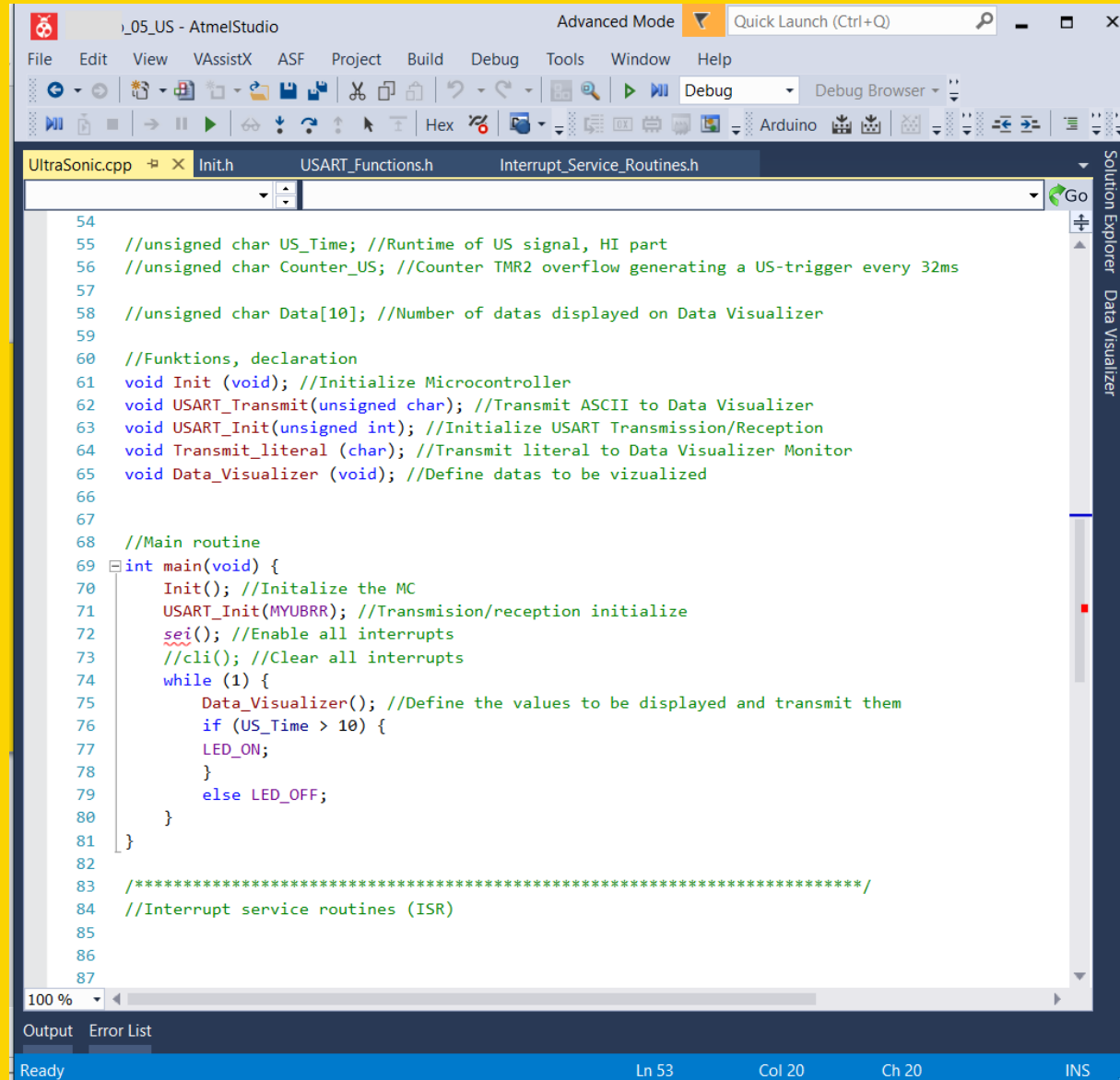
Ist die Aussage falsch ist er Null.

Ist die Aussage richtig ist er Eins.

Die Bedingung ist erfüllt, wenn sie

NICHT Null

ist!



The screenshot shows the Atmel Studio IDE interface. The main window displays the source code for 'UltraSonic.cpp'. The code includes comments and function declarations for initialization, transmission, and data visualization. The main function 'main' is highlighted, showing a while loop that calls 'Data_Visualizer()' and checks a condition 'US_Time > 10' to toggle an LED. The IDE interface includes a menu bar, a toolbar, and a status bar at the bottom showing 'Ready', 'Ln 53', 'Col 20', 'Ch 20', and 'INS'.

```
54  
55 //unsigned char US_Time; //Runtime of US signal, HI part  
56 //unsigned char Counter_US; //Counter TMR2 overflow generating a US-trigger every 32ms  
57  
58 //unsigned char Data[10]; //Number of datas displayed on Data Visualizer  
59  
60 //Funktions, declaration  
61 void Init(void); //Initialize Microcontroller  
62 void USART_Transmit(unsigned char); //Transmit ASCII to Data Visualizer  
63 void USART_Init(unsigned int); //Initialize USART Transmission/Reception  
64 void Transmit_literal(char); //Transmit literal to Data Visualizer Monitor  
65 void Data_Visualizer(void); //Define datas to be vizualized  
66  
67  
68 //Main routine  
69 int main(void) {  
70     Init(); //Initalize the MC  
71     USART_Init(MYUBRR); //Transmission/reception initialize  
72     sei(); //Enable all interrupts  
73     //cli(); //Clear all interrupts  
74     while (1) {  
75         Data_Visualizer(); //Define the values to be displayed and transmit them  
76         if (US_Time > 10) {  
77             LED_ON;  
78         }  
79         else LED_OFF;  
80     }  
81 }  
82  
83 /*****  
84 //Interrupt service routines (ISR)  
85  
86  
87
```

Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Elegoo Car Kit
Teil 2.6: Motor