

**Montage und Programmierung  
eines Roboters für  
ROBOCUP JUNIOR RESCUE  
mit Elegoo Car Kit  
Teil 2.4: Line Sensor**

*Von Charlotte und Andreas*

```
Line_Sensor.cpp Init.h USART_Functions.h  
C:\Users\User\Desktop\Tutorials\Atmel\Software\Elegoo_04_Line_Sensor\Elegoo_04_Line_Sensor\Line_Sensor.cpp  
8 * 1 TX PD1 TX In  
9 * 2 INT0 PD2 US-Echo In  
10 * 3 Digit PD3 Servo Out  
11 * 4 Digit PD4 Line-M In  
12 * 5 OC0B PD5 PWM-A Out  
13 * 6 OC0B PD6 PWM-B Out  
14 * 7 PCINT23 PD7 Color-1 In -> DDRD = 0b01101000  
15 * 8 Digit PB0 Motor-A Out  
16 * 9 Digit PB1 Motor-B Out  
17 *10 Digit PB2 Line-L In  
18 *11 PCINT3 PB3 Color-2 In  
19 *12 Digit PB4 Infr. In Not needed  
20 *13 LED PB5 LED Out -> DDRB = 0b00100011  
21 *A0 Ana PC0 Line-RA In  
22 *A1 Ana PC1 Line-LA In  
23 *A2 Ana PC2 Dist-1 In  
24 *A3 Ana PC3 Dist-2 In  
25 *A4 Digit PC4 Line-R In  
26 *A5 Digit PC5 US-Trig Out -> DDRC = 0b00100000
```

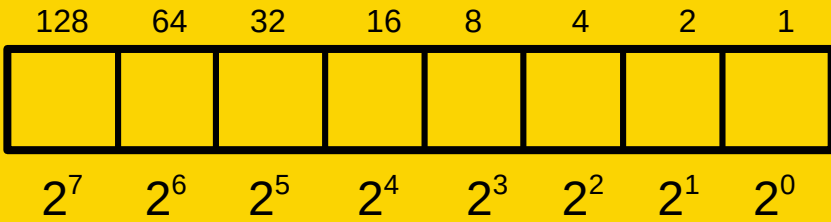
```
100 %  
65 //Main routine  
66 int main(void) {  
67     Init(); //Initialize the MC  
68     USART_Init(MYUBRR); //Transmission/reception initialize  
69     sei(); //Enable all interrupts  
70     cli(); //Clear all interrupts  
71     while (1) {  
72         Transmit_String("Text: May be as long as you want, but lasts and lasts and...! \n");  
73         Data_Visualizer(); //Define the values to be displayed and transmit them  
74         //Status Line Detector  
75         Line_digital_right = PINC & 0b00010000; //State of PORTC 4 pin, if the pin is HI, 16 will be displayed  
76         Line_digital_middle = PINB & 0b00000100; //PORTB 2 pin, display 4  
77         Line_digital_left = PIND & 0b00010000; //PortD 4 pin, display 16  
78     }  
79 }
```

# Linien-Sensor:

Liniensensor über weiß,  
entsprechendes Bit High (= 1)  
Über schwarz Bit Low (= 0)

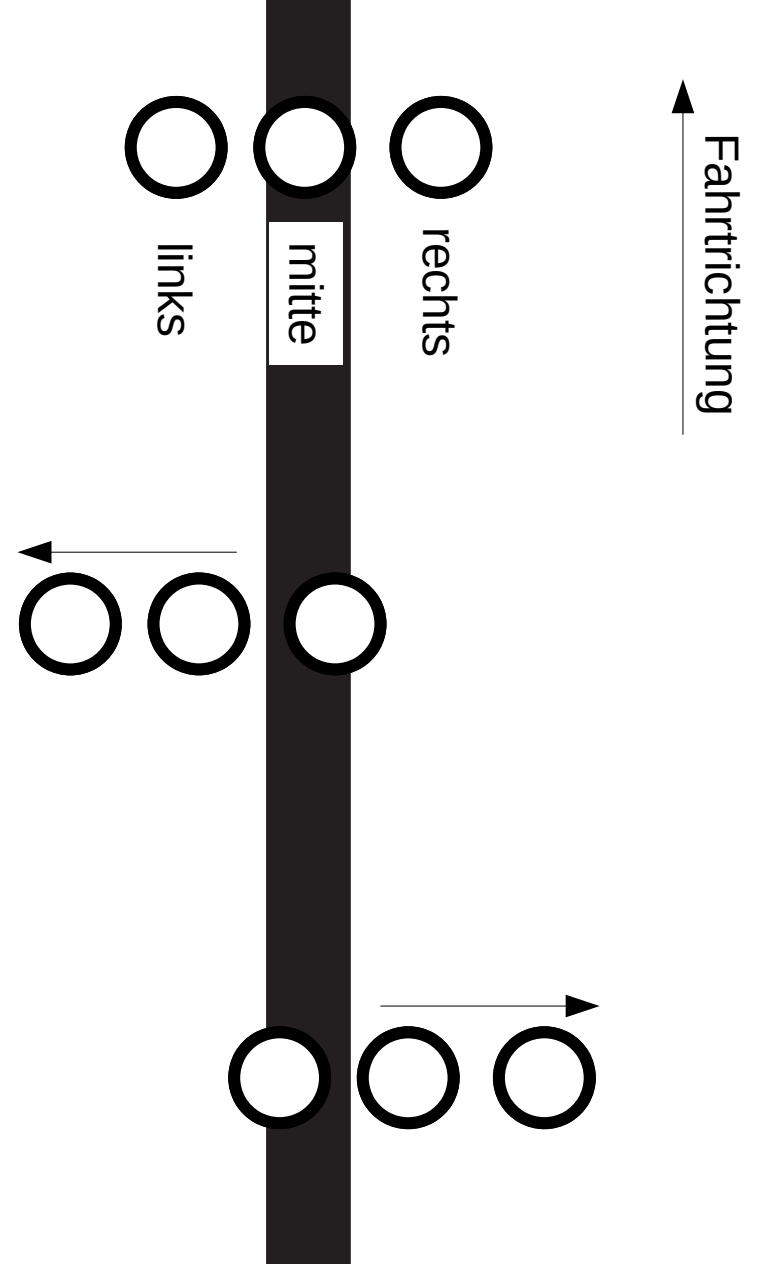
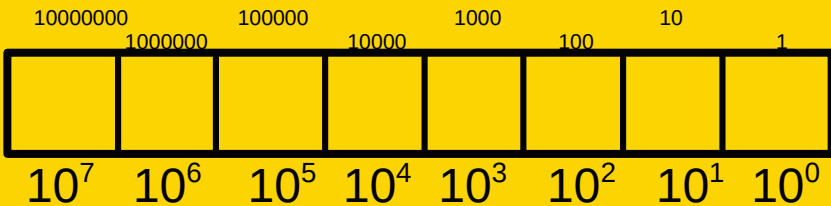
Rechter Sensor      Bit 4 von PORT C  
Mitte      Sensor      Bit 2 von PORT B  
Linker      Sensor      Bit 4 von PORT D

## Stellenwerte der Bits



Bit 4 hat den Stellenwert 16  
Bit 2 hat den Stellenwert 4

## Stellenwerte von Dezimalzahlen

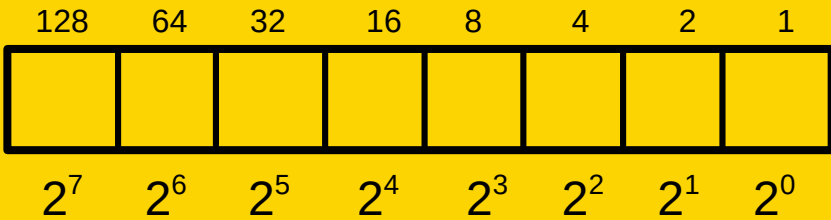


# Linien-Sensor:

Liniensensor über weiß,  
entsprechendes Bit High (= 1)  
Über schwarz Bit Low (= 0)

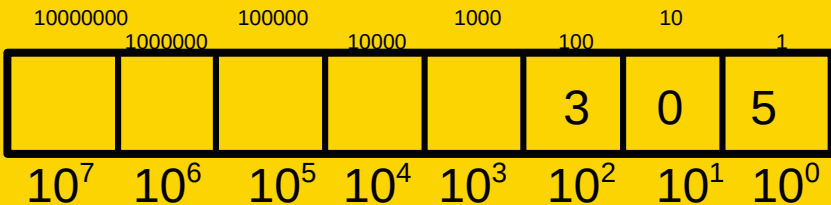
Rechter Sensor      Bit 4 von PORT C  
Mitte      Sensor      Bit 2 von PORT B  
Linker      Sensor      Bit 4 von PORT D

## Stellenwerte der Bits

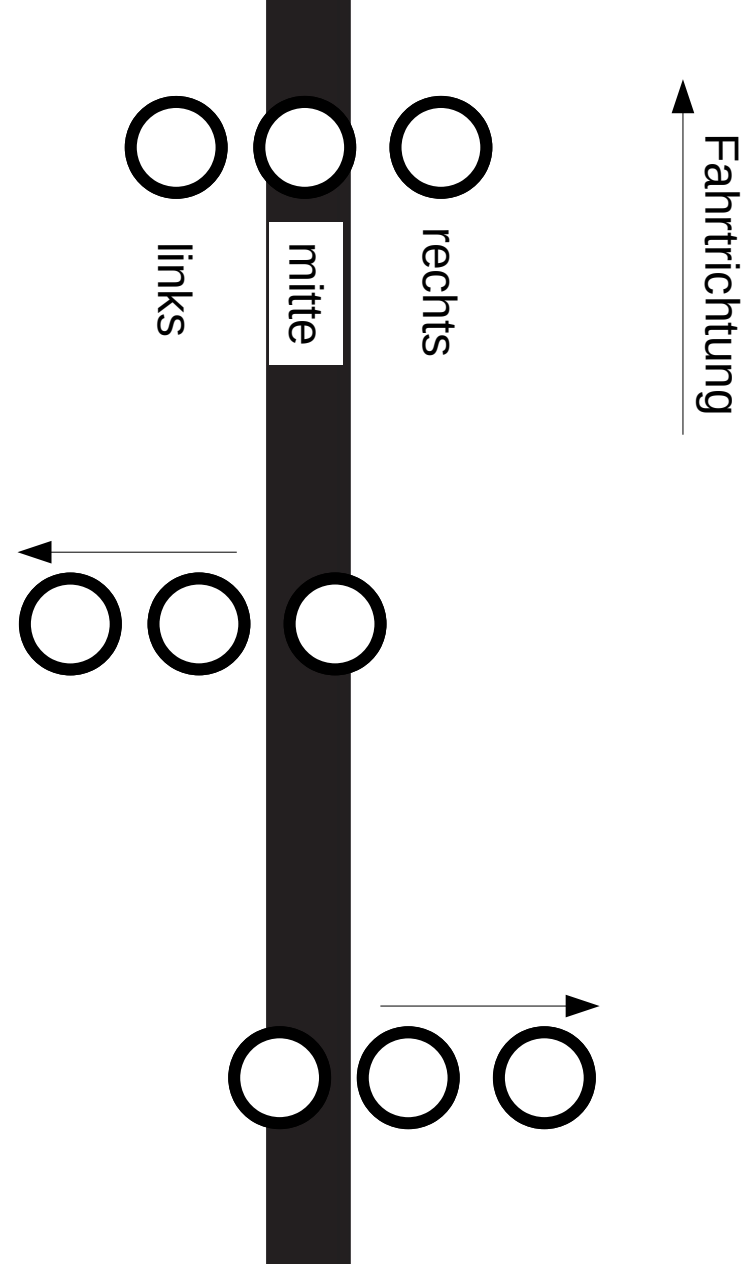


Bit 4 hat den Stellenwert 16  
Bit 2 hat den Stellenwert 4

## Stellenwerte von Dezimalzahlen



3 Hunderter  
0 Zehner  
5 Einer



# Linien-Sensor:

Liniensensor über weiß,  
entsprechendes Bit High (= 1)  
Über schwarz Bit Low (= 0)

Rechter Sensor      Bit 4 von PORT C  
Mitte      Sensor      Bit 2 von PORT B  
Linker      Sensor      Bit 4 von PORT D

## Stellenwerte der Bits

128	64	32	16	8	4	2	1
				1	0	1	1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

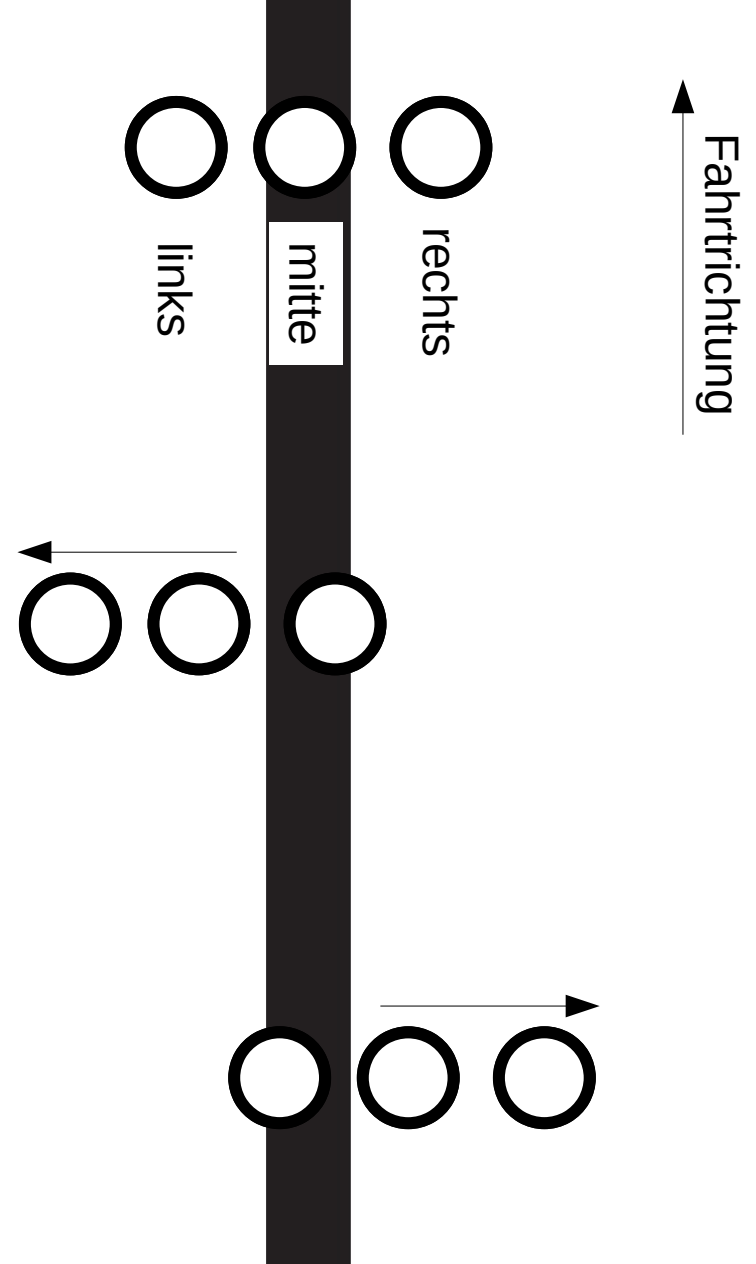
Bit 4 hat den Stellenwert 16  
Bit 2 hat den Stellenwert 4

## Stellenwerte von Dezimalzahlen

10000000	1000000	100000	10000	1000	100	10	1
					3	0	5
$10^7$	$10^6$	$10^5$	$10^4$	$10^3$	$10^2$	$10^1$	$10^0$

1 Achter  
0 Vierer  
1 Zweier  
1 Einer

3 Hunderter  
0 Zehner  
5 Einer



```

Line_Sensor.cpp Init.h USART_Functions.h
63
64
65 //Main routine
66 int main(void) {
67     Init(); //Initialize the MC
68     USART_Init(MYUBRR); //Transmission/reception initialize
69     sei(); //Enable all interrupts
70     cli(); //Clear all interrupts
71     while (1) {
72         // Transmit_String("Text: May be as long as you want, but lasts and lasts and...! \n");
73         Data_Visualizer(); //Define the values to be displayed and transmit them
74         //Status Line Detector
75         Line_digital_right = PINC & 0b00010000; //State of PORTC 4 pin, if the pin is HI, 16 will be displayed
76         Line_digital_middle = PINB & 0b00000100; //PORTB 2 pin, display 4
77         Line_digital_left = PIND & 0b00010000; //PortD 4 pin, display 16
78     }
79 }
80 /*****
81 //Other functions
82
83 //Datas to transmit to the Data Visualizer
84 void Data_Visualizer (void) {
85     Data[0] = Line_digital_right; //First value at the console
86     Data[1] = Line_digital_middle; //
87     Data[2] = Line_digital_left; //
88     Data[3] = z++; //Same as z = z + 1. Just to see how "fast" the transmission is
89     for (unsigned char i = 0; i < 4; i++) {
90         Transmit_literal(Data[i]); //Three digit literal unsigned
91     }
92     USART_Transmit('\n'); //New line
93     //USART_Transmit(10); //ASCII for "line feed", LF
94     //USART_Transmit('\n'); //Carriage return (CR) or 13
95 }
96
97 /*****

```

**Data Visualizer**

**DGI Control Panel**

**Serial Port Control Panel**

Arduino Uno ( Disconnect)

DTR  RTS

Open Terminal

Autodetect protocols

Baud rate: 76800 Parity: None

**Terminal 1**

0	4	16	49
0	4	16	50
0	4	16	51
0	4	16	52
0	4	16	53
0	4	16	54
0	4	16	55
0	4	16	56
0	4	16	57
0	4	16	58
0	4	16	59
0	4	16	60
0	4	16	61
0	4	16	62
0	4	16	63
0	4	16	64

Clear  Add \r\n  Hexadecimal Values

Show Timestamp  Automatically Scroll to End

Montage und Programmierung  
eines Roboters für  
**ROBOCUP JUNIOR RESCUE**  
mit Elegoo Car Kit  
Teil 2.5: Ultraschall

Von Charlotte und Andreas