

Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Arduino Nano
Teil 2.4: Black 'n White

Von Charlotte und Andreas

Black_and_White.cpp

Black_and_White.cpp | C:\Kassel_Uni\RoboCup\Tutorials\Atmel\Arduino Nano\Lernprogramme\05 Black and White\05_Line_Sensor\Black_and_White.cpp

```

10 * 3 INT1 PD3 Encoder In
11 * 4 Digit PD4 US-Trig Out
12 * 5 OC0B PD5 PWM-B Out
13 * 6 OC0A PD6 PWM-A Out
14 * 7 PCINT23 PD7 Color-1 In -> DDRD = 0b01110000
15 * 8 Digit PB0 Motor-A Out
16 * 9 Digit PB1 Motor-B Out
17 *10 Digit PB2 In
18 *11 Digit PB3 Incl-F In Inclination front
19 *12 Digit PB4 Incl-B In Inclination back
20 *13 LED PB5 LED Out -> DDRB = 0b00100011
21 *A0 Ana PC0 Dist-F In Distance front
22 *A1 Ana PC1 Dist-B In Distance back
23 *A2 Ana PC2 Bl-Wh-L In Black/White Sensor Left
24 *A3 Ana PC3 Bl-Wh-R In Black/White Sensor Right
25 *A4 Ana PC4 In
26 *A5 Ana PC5 In -> DDRC = 0b00100000
27 *A6 Ana ADC6 In
28 *A7 Ana ADC7 In
29 */

```

100 %

```

61 //Main routine
62 int main(void) {
63     Init(); //Initialize the MC
64     USART_Init(MYUBRR); //Transmission/reception initialize
65     sei(); //Enable all interrupts
66     //cli(); //Clear all interrupts
67     while (1) {
68         // Transmit_String("Text: May be as long as you want, but lasts and lasts and...! \n");
69         Data_Visualizer(); //Define the values to be displayed and transmit them
70         //Status Line Detector
71         BlacknWhite_left = PINC & 0b00000100; //State of PORTC 2 pin, if the pin is HI, 4 will be displayed
72         BlacknWhite_right = PINC & 0b00001000; //PortC 3 pin, display 8
73     }
74 }

```

100 %

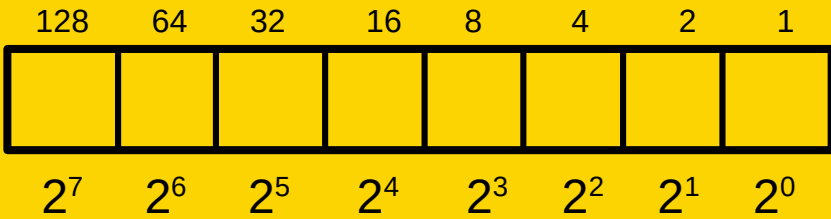
Output Error List VA Find References Results

Linien-Sensor:

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

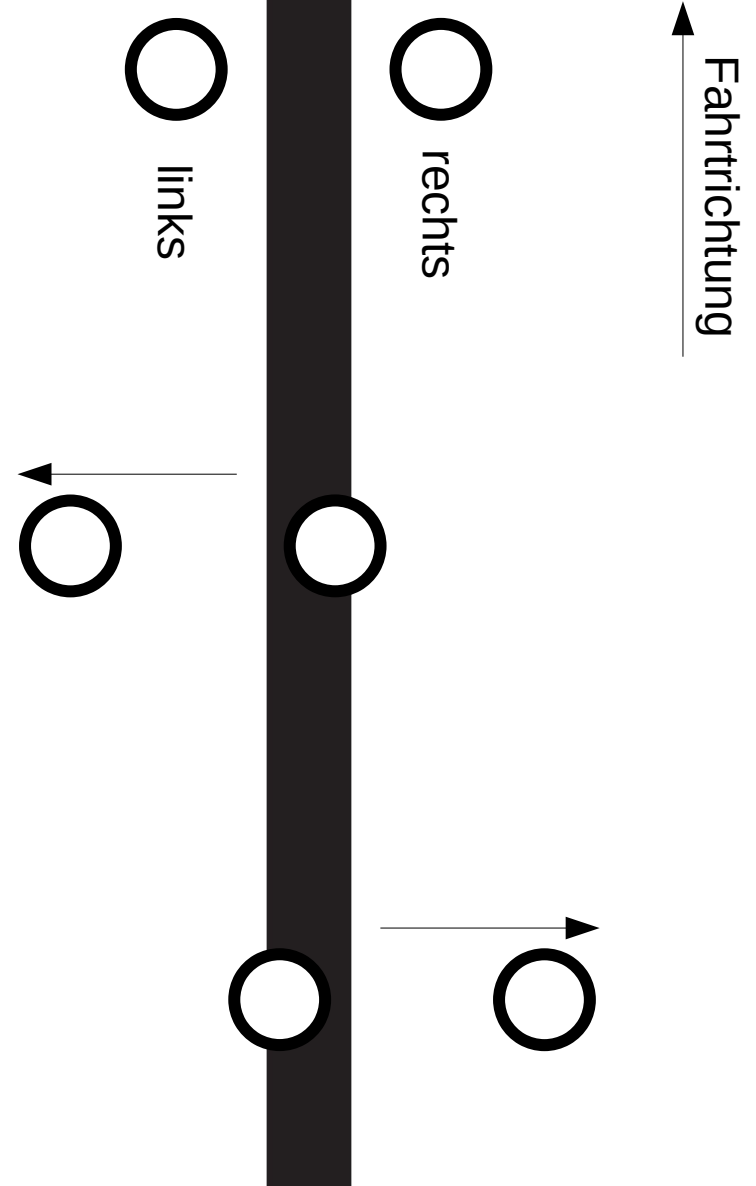
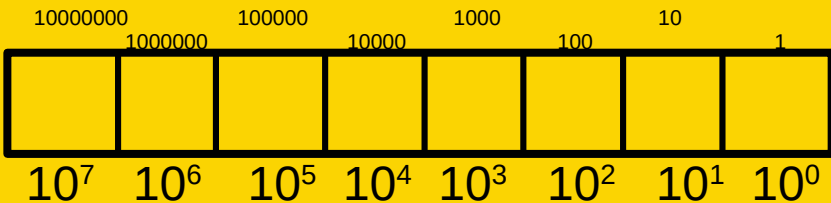
Linker Sensor Bit 2 von PORT C
Rechter Sensor Bit 3 von PORT C

Stellenwerte der Bits



Bit 2 hat den Stellenwert 4
Bit 3 hat den Stellenwert 8

Stellenwerte von Dezimalzahlen

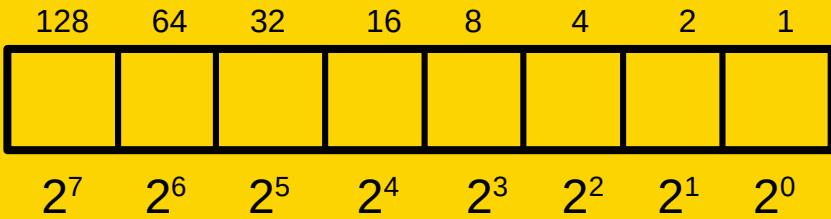


Linien-Sensor:

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

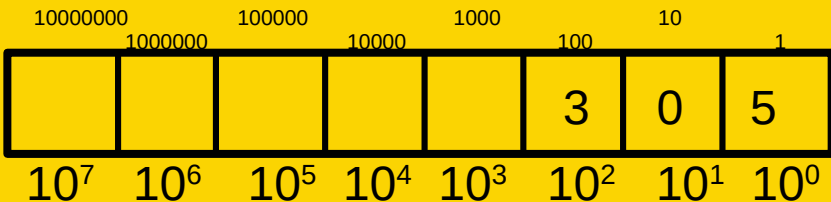
Linker Sensor Bit 2 von PORT C
Rechter Sensor Bit 3 von PORT C

Stellenwerte der Bits

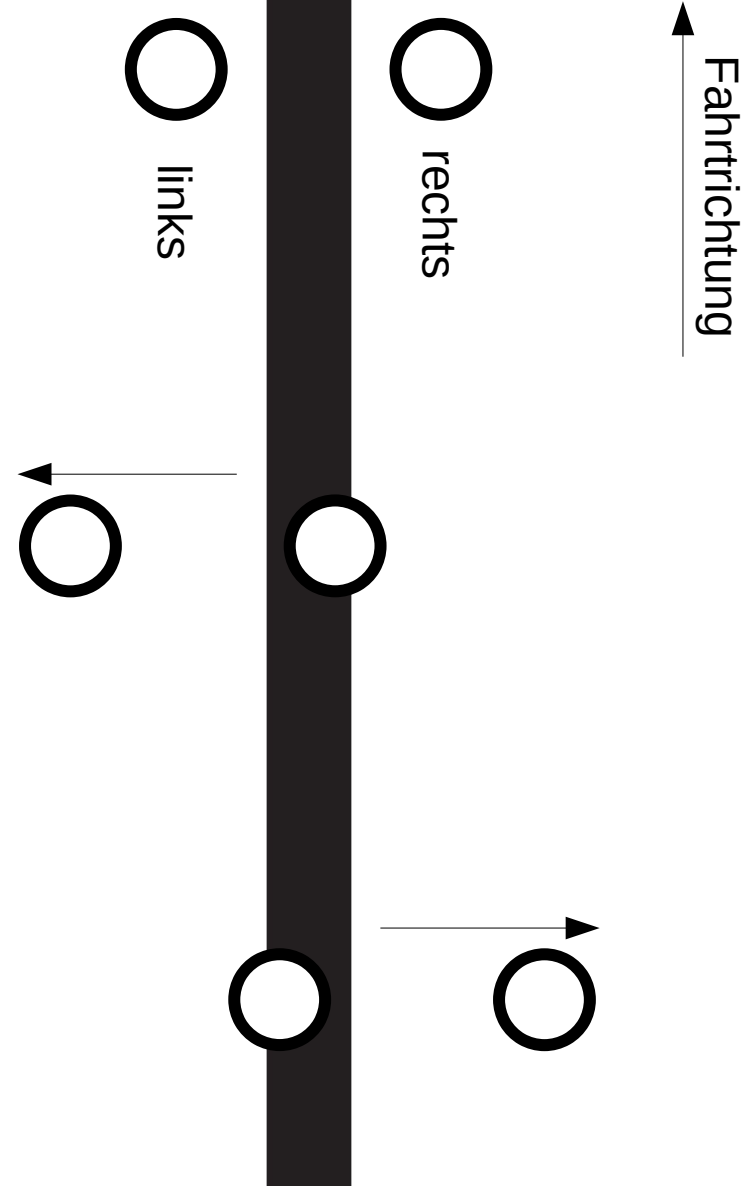


Bit 2 hat den Stellenwert 4
Bit 3 hat den Stellenwert 8

Stellenwerte von Dezimalzahlen



3 Hunderter
0 Zehner
5 Einer



Linien-Sensor:

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

Linker Sensor Bit 2 von PORT C
Rechter Sensor Bit 3 von PORT C

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

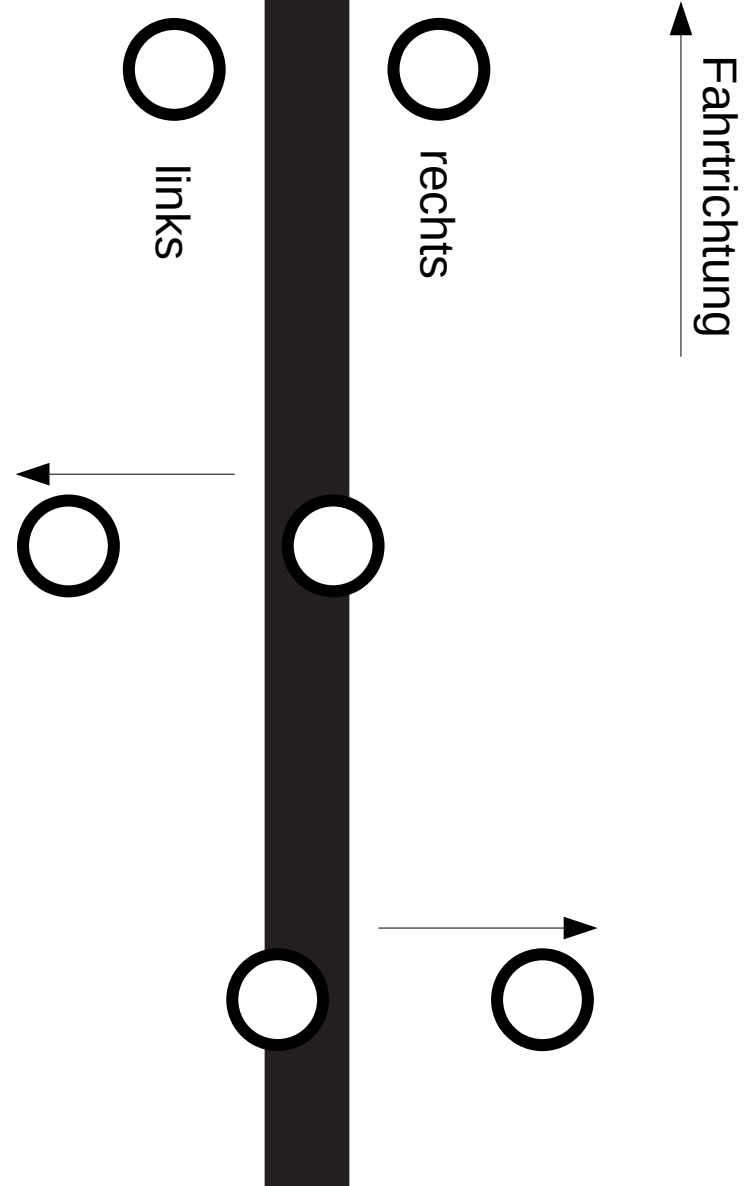
Bit 2 hat den Stellenwert 4
Bit 3 hat den Stellenwert 8

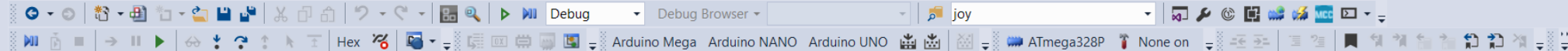
Stellenwerte von Dezimalzahlen

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





Black_and_White.cpp* [X]

Black_and_White.cpp C:\Kassel_Uni\RoboCup\Tutorials\Atmel\Arduino Nano\Lernprogramme\05 Black and White\05_Black_and_White\Black_and_W

```

60
61 //Main routine
62 int main(void) {
63     Init(); //Initialize the MC
64     USART_Init(MYUBRR); //Transmission/reception initialize
65     sei(); //Enable all interrupts
66     //cli(); //Clear all interrupts
67     while (1) {
68         // Transmit_String("Text: May be as long as you want, but lasts and lasts and...! \n");
69         Data_Visualizer(); //Define the values to be displayed and transmit them
70         //Status Line Detector
71         BlacknWhite_left = PINC & 0b00000100; //State of PORTC 2 pin, if the pin is HI, 4 will be displayed
72         BlacknWhite_right = PINC & 0b00001000; //PortC 3 pin, display 8
73     }
74 }

```

100 %

```

75 /*****
76 //Other functions
77
78 //Datas to transmit to the Data Visualizer
79 void Data_Visualizer (void) {
80     Data[0] = z++; //Same as z = z + 1. Just to see how "fast" the transmission is
81     Data[1] = (unsigned char) (T2ck >> 8); //Casted to char, HI byte of T2ck
82     Data[2] = (unsigned char) T2ck; //Casted to char, LO byte of T2ck
83     Data[3] = BlacknWhite_left; //Fourth value at the console
84     Data[4] = BlacknWhite_right; //Fifth
85     for (unsigned char i = 0; i < 5; i++) { //0 to 4
86         Transmit_literal(Data[i]); //Three digit literal unsigned
87     }
88     USART_Transmit('\n'); //New line
89     //USART_Transmit(10); //ASCII for "line feed", LF
90     //USART_Transmit('\r'); //Cariage return (CR) or 13
91 }
92
93 /*****

```

100 %

Output Error List

Ready

Data Visualizer

Configuration

DGI Control Panel

Serial Port Control Panel

USB-SERIAL CH340 (COM10) Disconnect

DTR RTS

Open Terminal

Autodetect protocols

Baud rate: 76800 Parity: None Stop bits: 1 bit

Terminal 0

226	3	170	4	8
227	3	191	4	8
228	3	212	4	8
229	3	234	4	8
230	3	255	4	8
231	4	20	4	8
232	4	42	4	8
233	4	63	4	8
234	4	84	4	8
235	4	106	4	8
236	4	127	4	8
237	4	148	4	8
238	4	170	4	8
239	4	191	4	8
240	4	212	4	8
241	4	234	4	8
242				

Clear Add \r\n Hexadecimal Values Show Timestamp

Automatically Scroll to End

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

Von Charlotte und Andreas