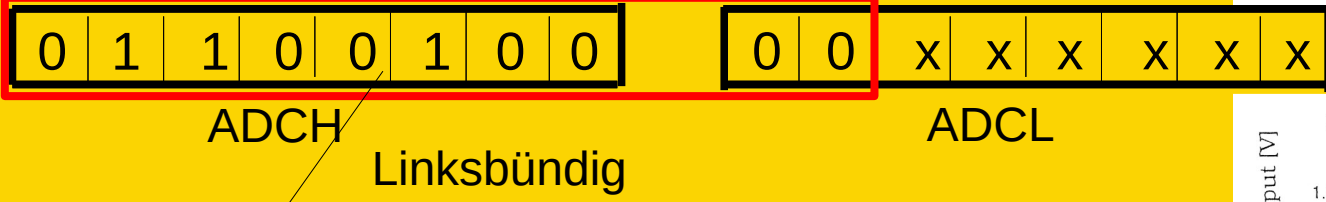


Montage und Programmierung  
eines Roboters für  
ROBOCUP JUNIOR RESCUE  
mit Elegoo Car Kit  
Teil 2.9: Analog Digital Converter

Von Charlotte und Andreas

Referenzspannung +5V  
 5V = 5000 mV (Millivolt)  
 5000 mV / 1024 ~ 5mV  
 2V = 2000 mV  
 2000 mV / 5mV = 400  
 ADCH = 100

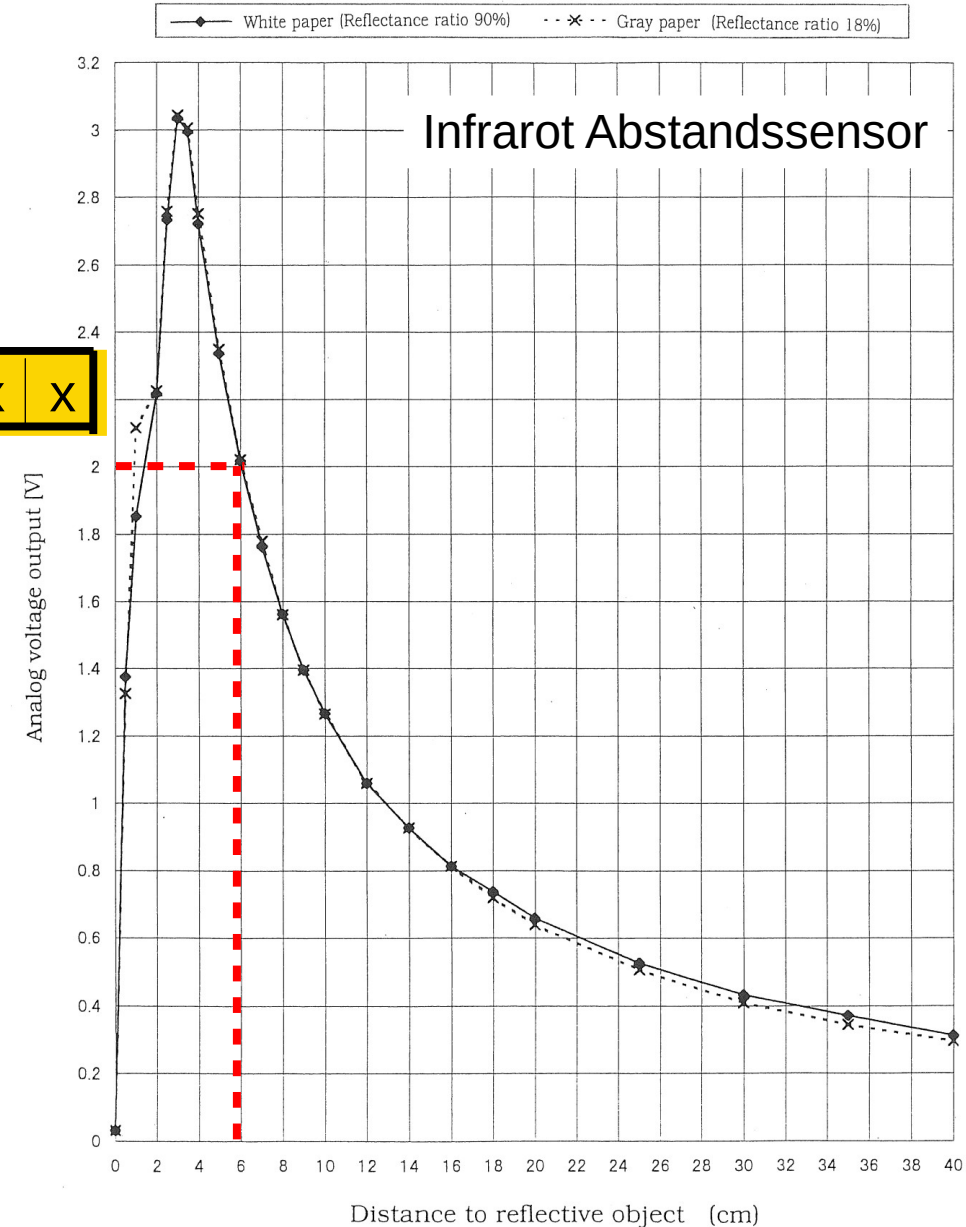
HEX	190	
DEC	400	400
OCT	620	
BIN	0d01 1001 0000	10 Bits



HEX	64	100
DEC	100	
OCT	144	
BIN	0110 0100	

```

35 //ADC settings
36 ADMUX = 0b01100001; //AVCC with external capacitor at AREF pin, left adjusted, A1-Channel
37 ADCSRA = 0b11101110; //1enable ADC,1start conversion,1auto trigger,0int. flag,1int. enable,110pr
38 //DIDR0 = 0b00001111; //
  
```



# Interrupt\_Service\_Routines.h

```
91 //Timer 1 Overflow Interrupt
92 //ISR(TIMER1_OVF_vect) { //Prescaler TMR1 1:8 = Interrupt every 32 ms
93     US_Time = 255; //If there is no obstacle
94     //LED_TOGGLE; //for testing
95 }
96
97 //Interrupt on completion of Analog_to_Digital_Conversion
98 //ISR(ADC_vect) {
99     Analogue_value[Channel] = ADCH; //HI byte of the resolution
100     Channel++; //Channels of the ADC
101     if (Channel == 4) Channel = 0; //We only need 4 channels of 6 of the
102     ADMUX = 0b01100000 | Channel; //Bit 3:0 choose the channel
103 }
104
105 #endif /* INTERRUPT_SERVICE_ROUTINES_H */
```

# Init.h

```
25 TCCR0B = 0b00000010; //00doesn't matter,00unimpl.,0PWM,011TMR0-Pres1:8
26 TIMSK0 = 0b00000000; //Timer 0 no Interrupt
27 //Timer1 Ultrasonic
28 TCCR1A = 0b00000000; //Normal-Mode
29 TCCR1B = 0b00000010; //Timer 1 Prescaler 1:8, normal-Mode
30 TIMSK1 = 0b00000001; //Interrupt at overflow, appr. 32ms
31 //Timer2 Time divider
32 TCCR2A = 0b00000000; //0000Normal port operation,00unimpl.,00WGM2[1:0]
33 TCCR2B = 0b00000010; //0WGM22,010Prescaler 1:8
34 TIMSK2 = 0b00000001; //Interrupt at TMR2-OVFL, 128us
35 //ADC settings
36 ADMUX = 0b01100001; //AVCC with external capacitor at AREF pin, left adjusted, A
37 ADCSRA = 0b11101110; //1enable ADC,1start conversion,1auto trigger,0int. flag,1in
38 //DIDR0 = 0b00001111; //
39 }
40
41 #endif /* INIT_H */
```

10\_ADC - AtmelStudio

File Edit View VAssistX ASF Project Build Debug Tools Window Help

Debug Debug Browser

ADC.cpp Init.h Interrupt\_Service\_Routines.h Drive\_Functions.h USART\_Functions.h

```
42 #define PWM_B OCR0B //Overflow compare register B
43 //define hundred 0 //First digit of a 3 digit literal
44 //define ten 1 //Second digit
45 //define one 2 //Third digit
46 #define Line_analogue_right Analogue_value[0] //Right line sensor
47 #define Line_analogue_left Analogue_value[1] //Left line sensor
48 #define Distance_1 Analogue_value[2] //UV distance sensor
49 #define Distance_2 Analogue_value[3] //UV distance sensor
50
51 //Macros
52 #define LED_ON PORTB |= (1 << PORTB5) //Push PORTB pin 5 high (LED, pin 13 "Uno")
53 #define LED_OFF PORTB &= ~(1 << PORTB5) //~ Not Operator
54 //define LED_TOGGLE PINB |= (1<<PINB5) //Toggle Arduino Uno pin 13
55
56
57 //Variables
```

100 %

```
123 /*****
124 //Other functions
125
126 //Data beeing transmitted to the Data Visualizer
127 void Data_Visualizer (void) {
128     Data[0] = Line_digital_right;
129     Data[1] = Line_digital_middle;
130     Data[2] = Line_digital_left;
131     Data[3] = Line_analogue_right;
132     Data[4] = Line_analogue_left;
133     // Data[5] = Color_Value_1;
134     // Data[6] = Color_Value_2;
135     Data[7] = Distance_1;
136     Data[8] = Distance_2;
137     Data[9] = US_Time;
138 }
139
140 /*****/
```

100 %

Output Error List VA Find References Results

Ready

# Line

## Aufgabe:

Stellt euren Roboter auf eine Kachel mit Linie. Lest die Werte ab die, bei weißem Untergrund, angezeigt werden und tragt sie in ein Diagramm ein. Schiebt dann den Roboter langsam **quer** über den Strich und stellt den **analogen** Wert fest, bei dem der digitale Sensor von Eins auf Null springt. (Diesen Vorgang ruhig mehrmals durchführen, damit ihr ein Gefühl für die Messgenauigkeit bekommt!) Dann schiebt den Roboter weiter, bis ihr die dunkelste Stelle gefunden habt. Wiederholt die Messung bei unterschiedlichen Stellungen des weißen Drehknopfs am Liniensensor.

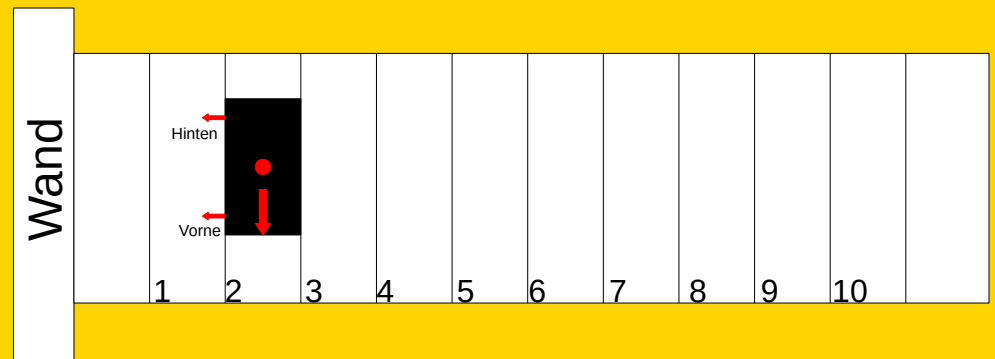
	digital links	digital rechts	analog links	analog rechts
weiß				
Wechsel bei unterschiedlichen Potistellungen				
schwarz				

# Maze

## Aufgabe:

Stellt euren Roboter neben eine Wand, auf ein weißes Blatt Papier. Das habt ihr präpariert mit Linien mit einem Abstand von 1 cm. Tragt in ein Diagramm ein, welche Werte ihr für welche Abstände des Roboters (äußere Kante) erhaltet. Gibt es Unterschiede zwischen vorne und hinten? Wie groß?

	1	2	3	4	5	6	7	8	9
Vorne									
Hinten									



## Line

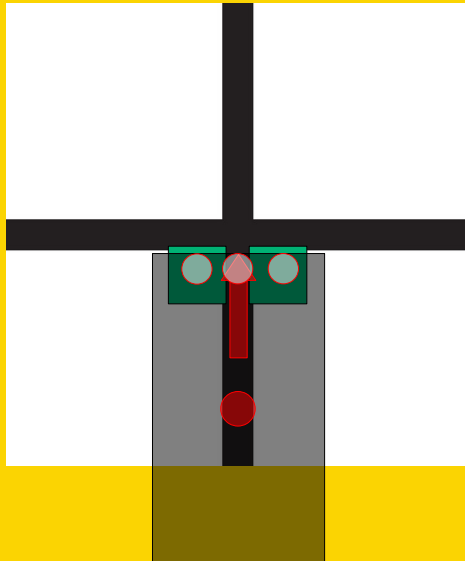
### Aufgabe:

Stellt euren Roboter genau mittig über die Linie an einer Kreuzung mit grünem Punkt.

Versucht zu ermitteln, welche Werte ihr **analog** für den grünen Punkt definieren könnt.

Schiebt dafür den Roboter ein wenig hin und her.

Was zeigen die **digitalen** Linien-Sensoren bei grün? Verändert sich das Verhalten bei unterschiedlichen Potistellungen?



## Maze

### Aufgabe:

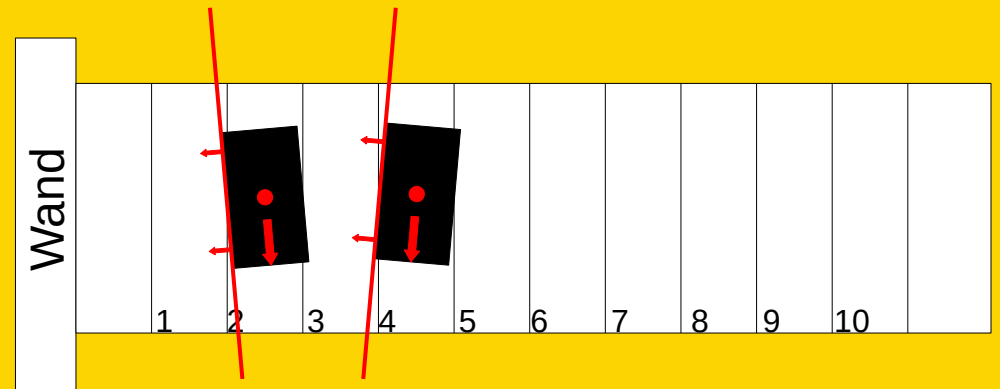
Stellt euren Roboter neben eine Wand.

Verdreht den Roboter um ca.  $5^\circ$ .  
(Benutzt ein Geodreieck um den Winkel auf Papier zu zeichnen.)

Wie groß ist der Unterschied zwischen vorne und hinten?

Wiederholt die Messung bei unterschiedlichen Abständen des Roboters zur Wand.

Was könnt ihr an den Werten ablesen?



Montage und Programmierung  
eines Roboters für  
**ROBOCUP JUNIOR RESCUE**  
mit Elegoo Car Kit  
Teil 2.10: Drive Analogue

Von Charlotte und Andreas