

Praktikum 4

Jetzt geht es los! In diesem Praktikum sollen die beiden vorderen Phototransistoren benutzt werden, um den ASURO am „Abgrund“ in Form einer schwarzen Linie anhalten zu lassen.

Für dieses Manöver müssen wir die Phototransistoren des ASURO verwenden, deren Signale mittels Analog-Digitalwandlung (AD/C) eingelesen werden. Da der AD-Wandler erst noch in der Vorlesung behandelt wird, hier das Template für die Initialisierung:

```
; A/D Conversion
ldi R16, (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1)
out ADCSRA, R16
```

Der linke Phototransistor wird durch folgende Einstellung des Analogmultiplexers vor dem AD-Wandler als Quelle der Wandlung ausgewählt:

```
ldi R16, (1 << REFS0) | (1 << MUX0) | (1 << MUX1) | (1 << ADLAR);
out ADMUX, R16
```

Der rechte Phototransistor dann entsprechend:

```
ldi R16, (1 << REFS0) | (1 << MUX1) | (1 << ADLAR);
out ADMUX, R16
```

Die Wandlung wird durch setzen des AD-Start-Conversion bits gestartet:

```
sbi ADCSRA, ADSC ; start conversion
```

Ist die Wandlung abgeschlossen, kann das in dem ADIF - Flag im ADCSRA-Port gesehen werden. Durch Schreiben einer 1 auf dieses Flag wird es wieder gelöscht.

Der gewandelte Wert kann in das Register R16 mit folgendem Befehl geladen werden:

```
in R16, ADCH
```

Danach müsste das ADIF - Flag durch schreiben einer 1 auf dieses Bit gelöscht werden und mit

„sbi ADCSRA, ADSC“ eine neue Wandlung gestartet werden...

Weiterhin muss die PWM - Generierung genutzt werden. Die Initialisierung geht wie folgt:

```
; for PWM (8-Bit PWM) on OC1A & OC1B
ldi R16, (1 << WGM10) | (1 << COM1A1) | (1 << COM1B1)
out TCCR1A, R16
ldi R16, (1 << CS11)
out TCCR1B, R16
```

Die Geschwindigkeit kann gesetzt werden, indem in die Register R2 (für linkes Rad) und R3 (für rechtes Rad) mit einem Wert zwischen 255 (max.

Geschwindigkeit) und 0 (steht still) beschrieben werden und die folgende Funktion aufgerufen wird:

Motorspeed:

```
;in R2 is value for left wheel, in R3 value for right wheel
;(assigned before call)
Push R16
```

```
clr R16
out OCR1AH, R16
out OCR1AL, R2
out OCR1BL, R3
pop R16
RET
```

Vorbereitung

Um die Schwellwerte für eine Detektion der schwarzen Linie richtig einstellen zu können, ist die Kenntnis der vom ADC gewandelten Werte sehr vorteilhaft. Deshalb sollen diese Werte über den UART auf den Hyperterminal des PC geschrieben werden.

Schreiben Sie sich eine Funktion, die einen in R16 stehenden Wert in entsprechende ASCII-Zeichen wandelt und mit Hilfe des UART überträgt.

Machen Sie sich mit der Funktionsweise der zu benutzenden Phototransistoren vertraut!

Wie sollte sich das zu messende Spannungssignal verhalten, wenn der Transistor hellere Gebiete beobachtet?

Durchführung

Die Fahrtrichtung wird über eine Motorbrücke gesteuert. Schreiben Sie zwei Makros, die die entsprechenden Ports setzen:

```
.macro MotorDir_FWD_FWD ; beide Motoren vorwärts
    ; left side: it is assumed that FWD is if T3 and T2 are conducting.
    ; right side: it is assumed that FWD is if T5 and T8 are conducting.
```

```
.macro MotorDir_RWD_RWD ; beide Motoren rückwärts.
```

Mit Hilfe der von Ihnen ausgearbeiteten Makros und Funktionen soll ein Programm erstellt werden, das den ASURO nach vorne fahren lässt, bis er an eine schwarze Linie kommt. An der schwarzen Linie soll er anhalten.

Lassen Sie sich zuerst die gewandelten Werte über die IR-Schnittstelle auf dem PC-Terminal anzeigen (ASURO aufgebockt, ein Blatt Papier mit einer schwarzen Linie vor den Sensor schieben und wieder wegnehmen)! Dafür benutzen Sie Ihre Routinen aus dem letzten Praktikum sowie aus der Vorbereitung. Erst nachdem Sie sich von der Funktionsfähigkeit Ihres Programmes überzeugt haben, lassen Sie den Roboter „von der Leine“ und probieren es aus. Hält der Roboter an der schwarzen Linie an, haben Sie das Ziel des Praktikums erreicht.

Zusatzaufgabe:

Statt anzuhalten, fahren Sie ein Stück zurück, drehen um ca. 90° und versuchen, ob es in einer anderen Richtung weitergeht! Der ASURO würde dann in einem durch eine schwarze Linie abgegrenzten Gebiet „frei“ herumfahren.