Mobiles Hardware-Praktikum 2017

Jan Burchard, Tobias Schubert, Bernd Becker

Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Freiburg im Breisgau

Übungsblatt 2 (24 Punkte)

Abgabe: 17.05.2017 bis 17:00 Uhr

Ziel:

Mit Hilfe des Microcontrollers werden die Motoren und Ultraschall-Sensoren des S-Trike gesteuert.

Prolog:

Die Motoren des Roboters werden über eine H-Brücke angesteuert. Diese erlaubt es die Drehrichtung der Motoren frei zu wählen. Zur Steuerung der H-Brücke werden zwei Signale pro Motor benötigt. Diese sind an der Buchse "Motors" auf die Experimentierplatine geleitet. Die genaue Ansteuerung der Motoren können Sie der Dokumentation der Experimentierplatine auf der Website entnehmen.

Die Ultraschallsensoren werden über jeweils eine Leitung angesprochen. Auf dieser Leitung wird zunächst der *trigger* vom Microcontroller an den Sensor gesendet. Der Sensor antwortet dann auf der gleichen Leitung nach einer Verzögerung mit einem Puls. Die Länge des Pulses korrespondiert zu der Entfernung des nächsten Objektes zum Sensor.

Da die Hardware des Arduino nicht sehr leistungsfähig ist, sind die Größen der Variablen anders definiert als auf einem normalen PC (ein "int" hat z.B. nur 16 Bit). Um Verwirrung zu vermeiden empfehlen wir Ihnen Typenbezeichner mit Größenangabe zu verwenden (also z.B. int8_t, int16_t und int32_t bzw. uint8_t, ...). "float" und "double" haben die übliche Größe (32 bzw. 64 Bit), sollten aber zur Verbesserung der Performance nur verwendet werden wenn sie wirklich benötigt werden.

Aufgabe 1 (3 Punkte): setMotorSpeed, digitalWrite

Erstellen Sie ein neues Arduino Programm. Definieren Sie die Pins 10 und 11 als Ausgang und verbinden Sie sie mit den Motor-Signalen A1 und A2.
Fügen Sie eine neue Funktion "setMotorSpeed" zu Ihrem Programm hinzu. Diese soll als Parameter einen Wahrheitswert (boolean) "forward" erhalten. Implementieren Sie dann die folgende Funktionalität in der Funktion: Hat "forward" den Wert $true$ soll der Motor sich vorwärts drehen, sonst rückwärts. Erzeugen Sie dazu die entsprechenden Steuersignale für den Motor.

Aufgabe 2 (2 Punkte) setMotorSpeed, analogWrite

Damit der Motor sich dreht muss jeweils eines der beiden Steuersignale $HIGH$ (1) sein. Um die Geschwindigkeit des Motors zu regulieren können Sie dieses Signal schnell zwischen 1 und 0 umschalten. Die Länge der 1-Phase bestimmt dann die Geschwindigkeit des Motors. Dieses Verfahren ist als Pulsweitenmodulation (PWM) bekannt.
Auf dem Arduino steht ihnen die Funktion "analogWrite(pin, value)" zur Verfügung. Diese erzeugt ein PWM-Signal zwischen immer aus ($value = 0$) und immer an ($value = 255$).
Erweitern Sie die Funktion "setMotorSpeed" um einen weiteren Eingang "spd" (uint8_t) und implementieren Sie die Funktion so, dass die Rotationsgeschwindigkeit vom Signal "spd" abhängt.

Ihre Funktion sollte den Motor mit gegebener Geschwindigkeit in beide Richtungen rotieren lassen können.

Aufgabe 3 (2 Punkte): driveForward

Verbinden Sie Motor B mit Pins 6 und 9 des Arduino und erweitern Sie die "setMotorSpeed" Funktion um einen weiteren Eingang um auszuwählen welcher Motor angesprochen werden soll. Da "analogWrite" nicht alle Pins unterstützt können nicht die Pins 12 und 13 verwendet werden.
Fügen Sie eine neue Funktion "driveForward' zu Ihrem Programm hinzu. Diese erhält als Parameter eine Zeit in Millisekunden und eine Geschwindigkeit. Die Funktion soll den Roboter für die gegebene Zeit gerade aus fahren lassen und dann stoppen.
Sie können den "delay(time)" Befehl verwenden um die entsprechende Verzögerung zu erreichen. Natürlich können Sie außerdem den die gerade definierte "setMotorSpeed" Funktion wiederverwenden.

Aufgabe 4 (6 Punkte): driveCurve

Fügen Sie eine neue Funktion "driveCurve' zu Ihrem Programm hinzu. Diese erhält zusätzlich zu den Parametern von "driveForward" einen weiteren für die Stärke der Kurve (zwischen -90 und +90). Implementieren Sie die Funktionalität so dass die Stärke den Kurvenradius bestimmt: Bei einer Stärke von $\pm$ 90 soll der Roboter sich auf der Stelle drehen, ist die Stärke 0 soll er geradeaus fahren. Dazwischen soll die Kurvenstärke linear steigen bzw. sinken.
Implementieren Sie diese Funktionalität indem Sie je nach Drehrichtung immer einen Motor mit der gegebenen Geschwindigkeit spd rotieren lassen und die Geschwindigkeit des anderen abhängig von der Stärke ändern (siehe Abbildung 1).

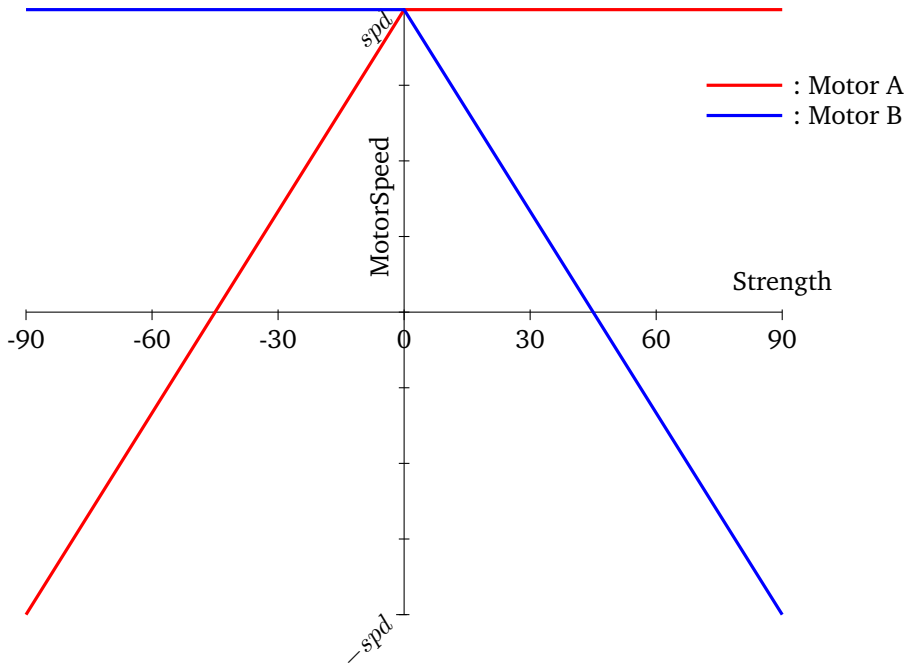


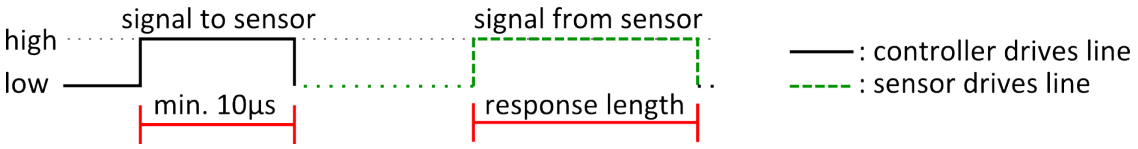
Abbildung 1: Die Geschwindigkeit der Motoren abhängig von der Stärke der Kurve. Negative Geschwindigkeiten bedeuten, dass der Motor sich rückwärts drehen soll.

Aufgabe 5 (4 Punkte): make shapes

Verbinden Sie die Taster wie auf Blatt 1 mit dem Analog-Pin A0. Je nach gedrückter Taste soll Ihr Roboter jeweils ein anderes Muster fahren (also z.B. Kreis, Sechseck, Stern, ...). Programmieren Sie 4 verschiedene Muster für die 4 äußeren Tasten. Die mittlere Taste wird in Aufgabe 7 mit einem Programm belegt

Aufgabe 6 (5 Punkte): measureDistance

Verbinden Sie den Ultraschall-Sensor US1 mit Pin 12 des Arduino. Fügen Sie eine neue Funktion "meassureDistance" zu ihrem Programm hinzu. Diese soll die gemessene Distanz zu einem Hindernis in cm zurück geben (oder -1 wenn kein Hindernis erkannt wurde). Als Parameter soll die Funktion den Ultraschall-Sensor-Pin einlesen.
Um den Sensor zu aktivieren muss zunächst der Sensor-Pin als Ausgang definieren und einen Trigger Impuls gesendet werden (*LOW*, dann mindestens 10 μs *HIGH*, dann wieder *LOW*). Danach muss der Pin als Eingang definiert werden. Der Sensor hält jetzt die Leitung für eine bestimmte Zeit auf *LOW* und sendet dann ebenfalls einen *HIGH*-Puls. Messen Sie die Länge dieses Pulses indem Sie über die "micros()"-Funktion[1] des Arduino die aktuelle Systemzeit (in μs) beim Start und Ende des Pulses in einer Variable speichern (uint32_t).



Wenn Sie nach 30 ms (= 30000 μs) keinen Puls vom Sender erhalten haben wurde keine Objekt detektiert. Ansonsten können Sie die Entfernung zum detektierten Objekt (in cm) berechnen indem Sie die gemessene Zeit durch 58 dividieren.
Zum Testen Ihrer Funktion verbinden Sie das LCD wie folgt und geben Sie die vom Ultraschall-Sensor gemessene Entfernung regelmäßig auf dem Display aus:
R/S : Pin 2, R/W : nicht verbunden, E : Pin 3, DB4 : Pin 4, DB5 : Pin 5, DB6 : Pin 7, DB7 : Pin 8

Aufgabe 7 (2 Punkte) don't crash into walls

Wird die mittlere Taste (S3) gedrückt soll der Roboter mit Hilfe des Ultraschall-Sensor fahren ohne ein Hindernis zu berühren. Wird ein Hindernis erkannt soll die Fahrtrichtung geändert werden.

Abgabe:

Archivieren Sie Ihr Programm in einer ZIP-Datei und laden Sie diese im Übungsportal hoch. Ihre Abgabe sollte 1 Programm mit einer Lösung für jede Aufgabe enthalten. Bewertet werden Ihre Implementationen von Aufgabe 1,2, 3, 4, 5, 6 und 7. Überprüfen Sie die Archiv-Datei auf Vollständigkeit.

[1] Der Zähler der micros() Funktion hat nur 32 Bit und läuft daher nach etwa 71 Minuten über. Dies brauchen Sie in Ihrer Implementierung nicht zu beachten.