

microSRCP



Photo by Nicholas Zambetti, www.arduino.cc

Modelleisenbahnsteuerung auf Basis von
standarddisierten Microcontrollerboards

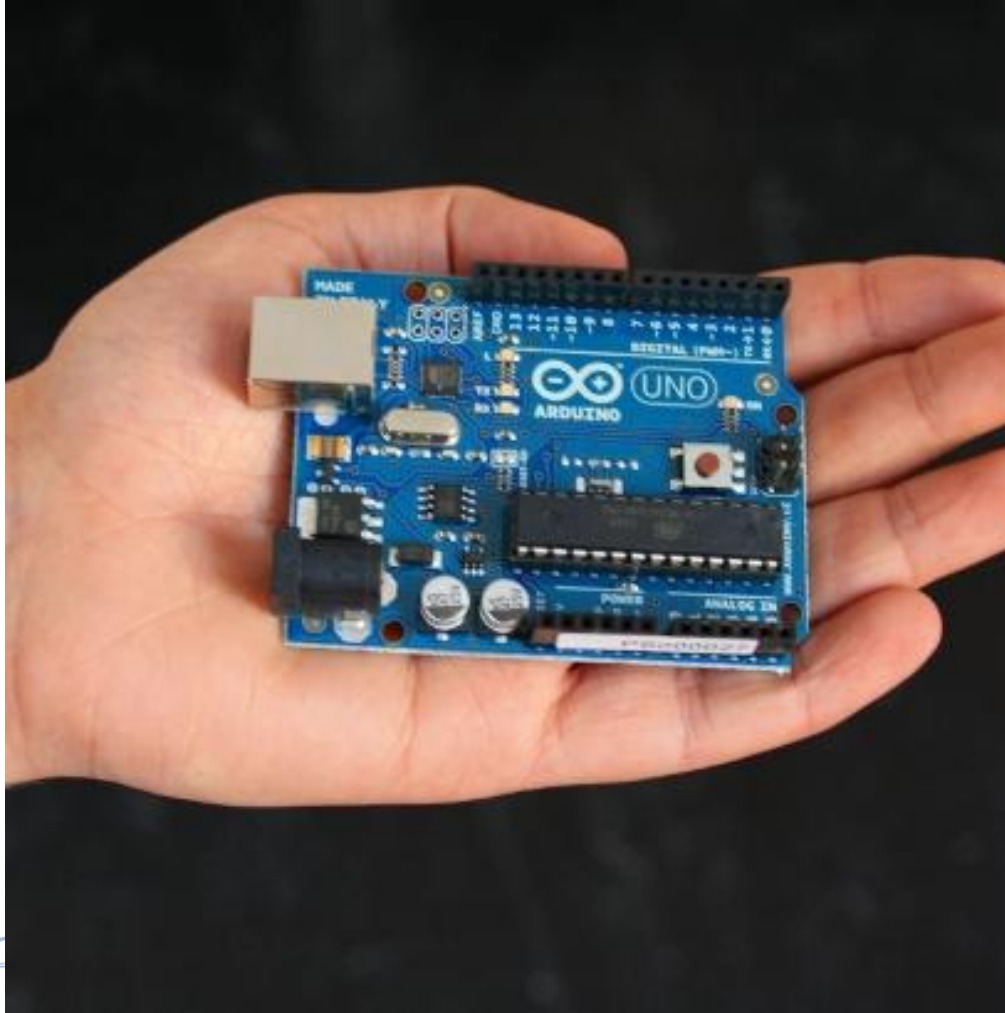
<http://developer.berlios.de/projects/microsrcp/>

Agenda



- Einleitung
- Die Komponenten
- Erste Schritte
- Die MicroSRCP Steuerung
- Anwendungsbeispiele

Arduino - OpenSource Hardware

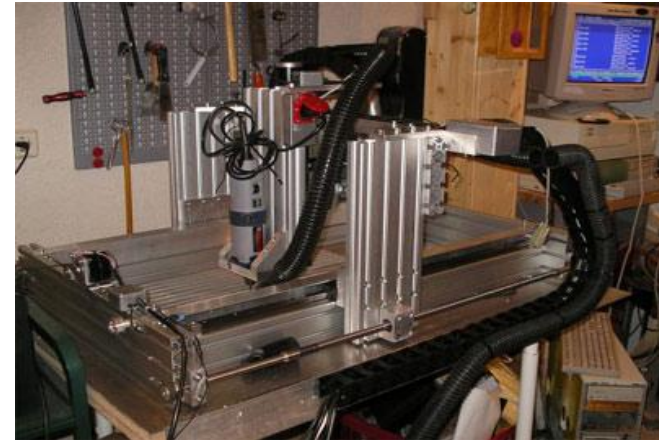


www.arduino.cc

Arduino - Einsatzgebiete



In Kombination mit Wii Remote zur RC Steuerung



Steuerung von CNC Fräsen und 3D Plotter



Fahrtrichtungsanzeige für Radfahrer



USB Host, zur Steuerung von USB Geräten

Arduino Entwicklungsumgebung



```
WebServer | Arduino 0022
File Edit Sketch Tools Help
WebServer
/*
  Web Server

  A simple web server that shows the value of the analog input pins.
  using an Arduino Wiznet Ethernet shield.

  Circuit:
  * Ethernet shield attached to pins 10, 11, 12, 13
  * Analog inputs attached to pins A0 through A5 (optional)

  created 18 Dec 2009
  by David A. Mellis
  modified 4 Sep 2010
  by Tom Igoe

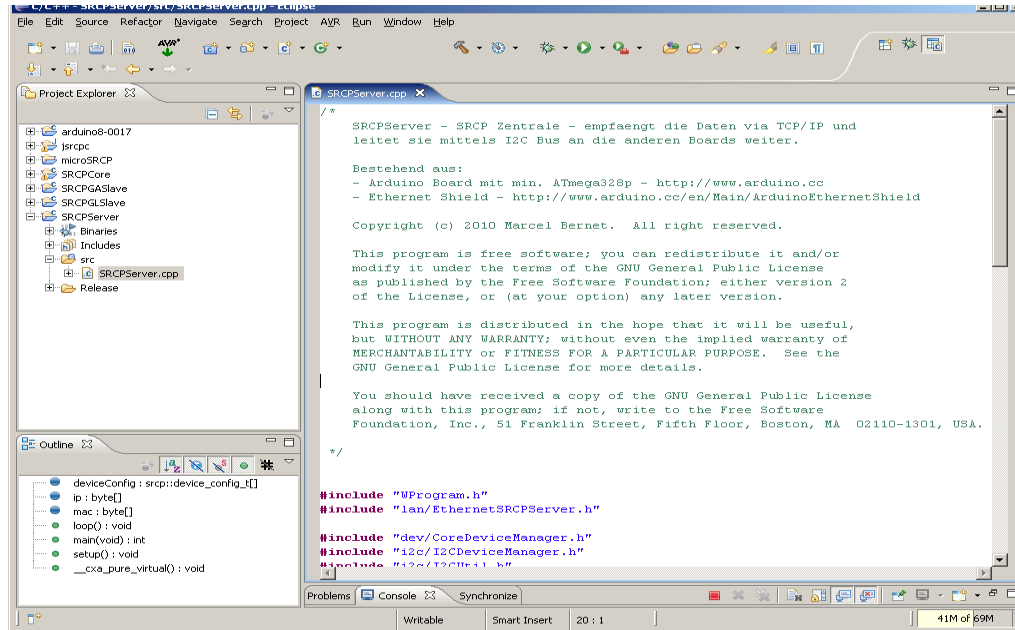
  */

#include <SPI.h>
#include <Ethernet.h>
```



- Vergleichbar mit <http://www.swisseduc.ch/informatik/karatojava/javakara/>
- *"Nach drei Wochen Informatik-Unterricht in der Jahrgangsstufe 11 mit Kara und dem Einstieg in JavaKara sind meine Schüler ebenso wie ich von Kara begeistert."* - Horst Gierhardt

Entwicklungsumgebung Eclipse



- Alternativ kann Eclipse (www.eclipse.org) verwendet werden.
- Die Installation ist nicht ganz einfach und hier (<http://www.arduino.cc/playground/Code/Eclipse>) beschrieben.
- Auch sind die Arduino Sourcen als Eclipse Projekt abzulegen (bei mir arduino8-0017, mit der Library libarduino8.a)
- Die benötigten microSRCP Eclipse Projekte sind im Repository.

Arduino und Google

Why Google Choosing Arduino Matters



```
AndroidAccessory acc("Google, Inc.", "DemoKit",  
    "DemoKit Arduino Board", "1.0",  
    "http://www.android.com",  
    "0000000012345678");
```

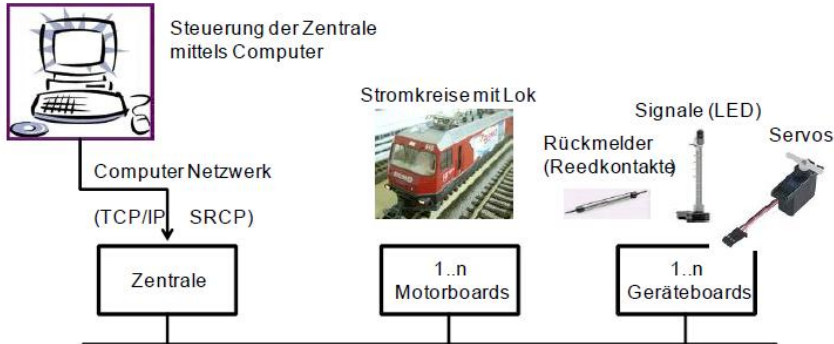
```
void setup() {  
    Serial.begin(115200); Serial.print("\r\nStart");  
    acc.powerOn();  
}  
void loop() {  
    byte msg[3];  
    if (acc.isConnected()) {  
        Serial.print("Accessory connected. ");  
        int len = acc.read(msg, sizeof(msg), 1);  
        Serial.print("Message length: ");  
        Serial.println(len, DEC);  
    }  
    delay(100);  
}
```

Quelle: <http://blog.makezine.com/archive/2011/05/why-google-choosing-arduino-matters-and-the-end-of-made-for-ipod-tm.html>

Arduino + Modelleisenbahn =



Übersicht



- Steuern
- Hardware
- Gestalten
- Sammeln

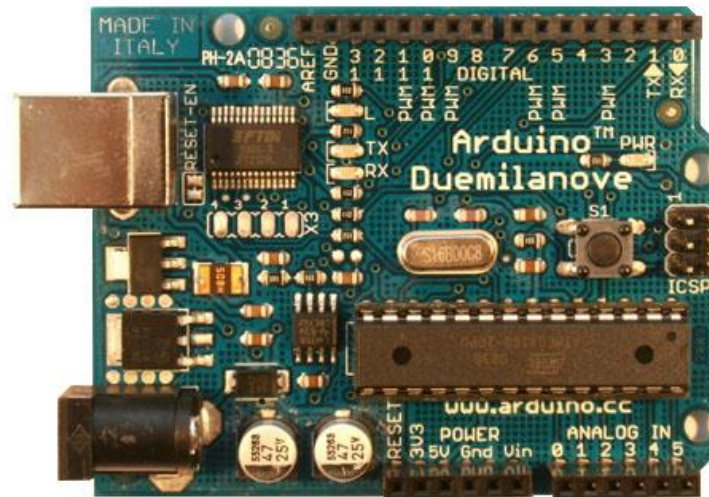


Agenda



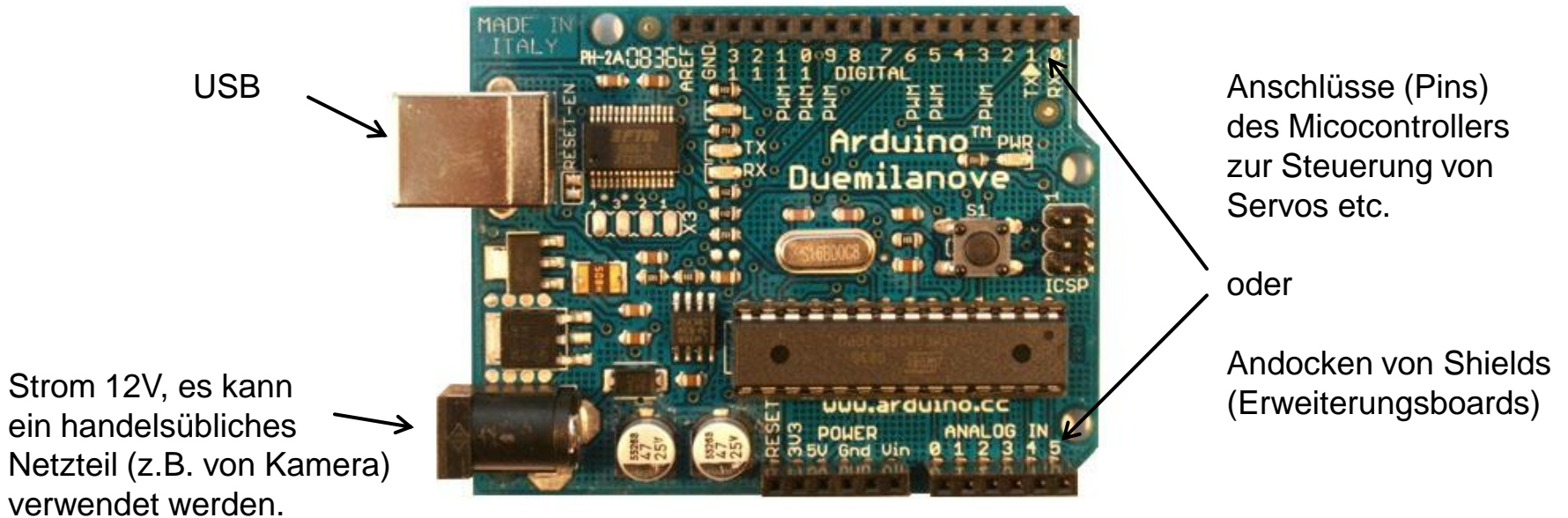
- Einleitung
- **Die Komponenten**
- Erste Schritte
- Die MicroSRCP Steuerung
- Anwendungsbeispiele

Das Hauptboard (1)



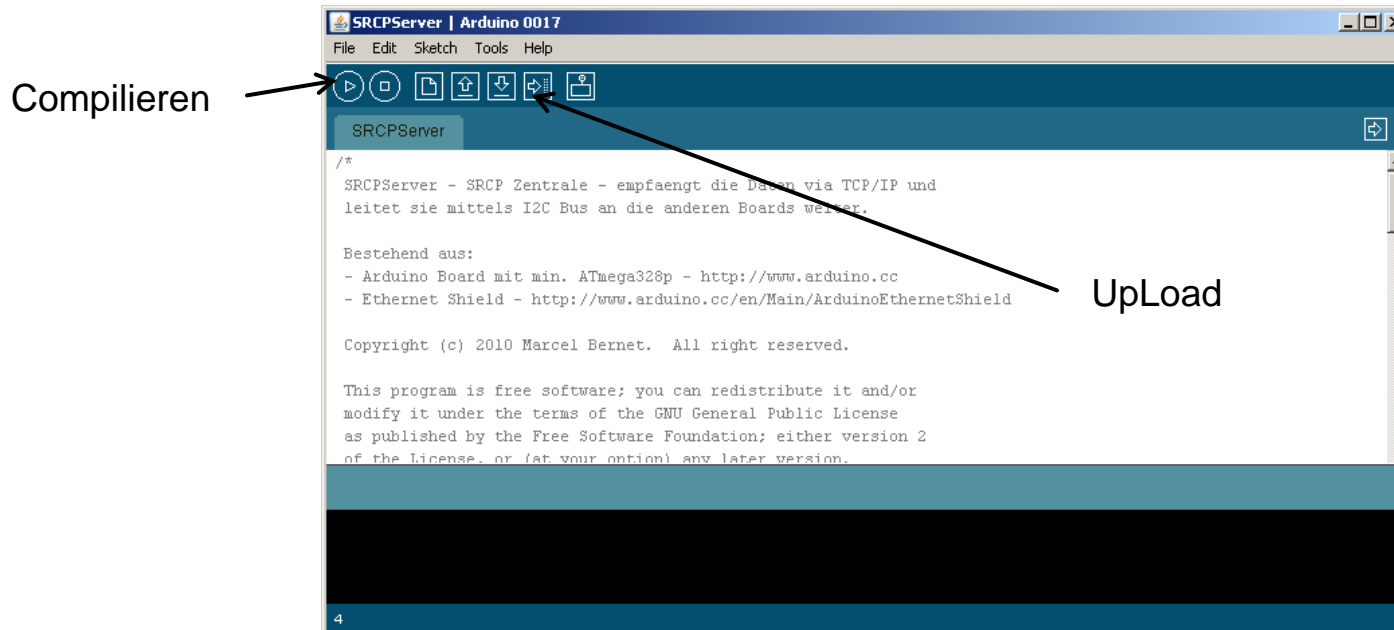
- Basis aller Boards bildet die Arduino Plattform -> www.arduino.cc
- Dabei handelt es sich um ein Microcontroller Board auf dem Erweiterungsboards, sogenannte Shields, aufgesteckt werden können -> <http://www.arduino.cc/en/Main/ArduinoShields>
- Boards können gekauft oder selbst hergestellt werden, die Informationen dazu sind frei erhältlich.

Das Hauptboard (2)



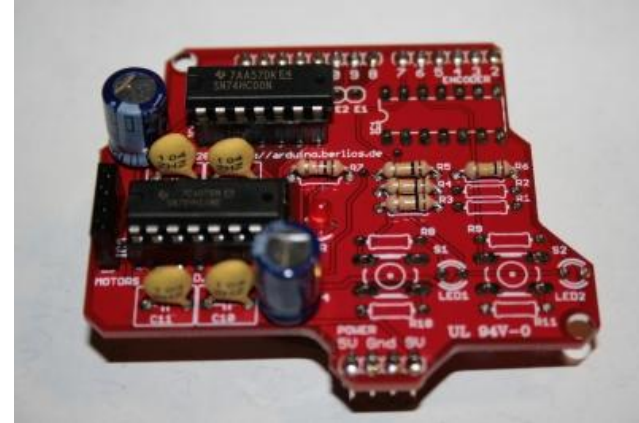
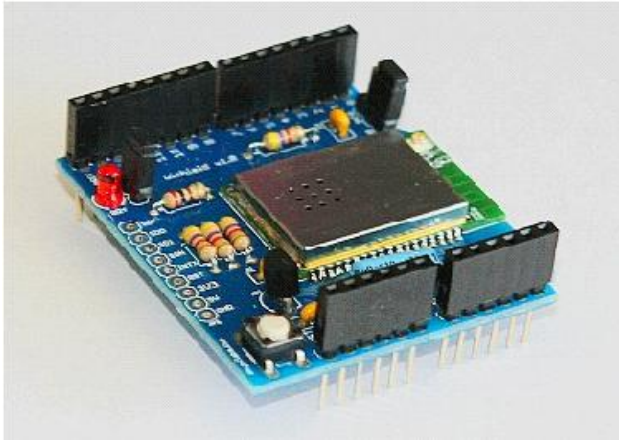
- Der grosse Chip ist der Microcontroller, über den USB Anschluss werden Sketche (Programme) geladen, eine 12V Stromanschluss sorgt für genügend Power.
- Alle Beine (Pins) des Microcontrollers sind nach aussen geführt und auf dem Board als Digital 0 – 13 und Analog 0 – 5 ersichtlich.
- Die Anschlüsse liefern genügend Strom um Servos, Lichtsignale mit LED direkt anzusteuern. Umgekehrt können an den Anschlüssen Momentkontakte abgefragt werden.

Entwicklungsumgebung Arduino



- Die Entwicklungsumgebung wird benötigt um die Steuerungssoftware, sogenannte Sketches, zu compilieren und in die Boards zu laden (= UpLoad)
- Die Installation ist Ausführlich unter den folgenden Links beschrieben:
 - <http://arduino.cc/en/Main/Software>, <http://www.freeduino.de/books>

Shields



- Mittels Erweiterungsboards, sogenannten Shields, kann die Funktionalität des Board erweitert werden.
- Diese werden einfach in die Buchsenleiste des Arduino Board gesteckt und mittels entsprechender Software angesteuert.
- Es existieren eine Vielzahl von Shields, u.a. für
 - Netzwerkanbindung mittels RJ45 / Ethernet
 - Ansteuerung von Motoren (Lokomotiven).
 - Ein Shield zum Abspielen von Sounddateien z.B. Für Bahnhofsdurchsagen
 - Ein Prototypenshield für eigene Entwicklungen.

Übung 1



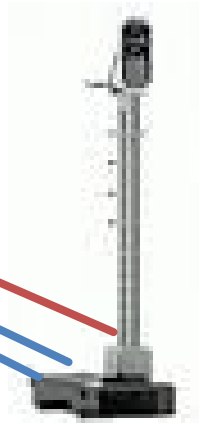
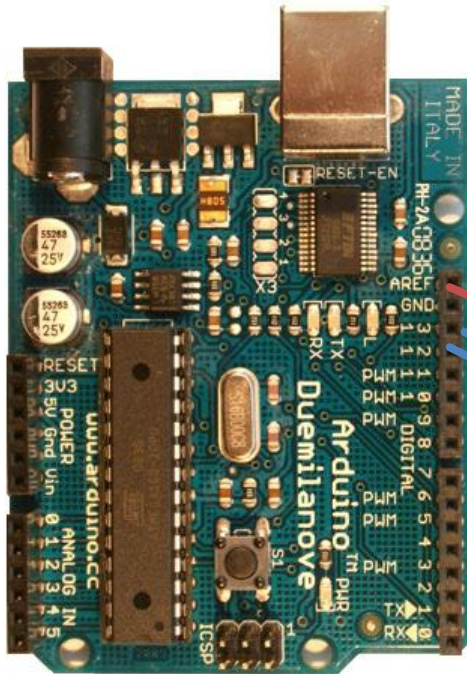
- Installieren Sie die benötigte Software (Entwicklungsumgebung, Treiber)
- Schliessen Sie das Arduino Board an Ihren Computer an.
- Wählen Sie unter File -> Examples -> Basic den Blink Sketch aus.
- Nach Compilieren und Upload, sollte das Led (auf jedem Board vorhanden) an Pin 13 Blinken.

Agenda



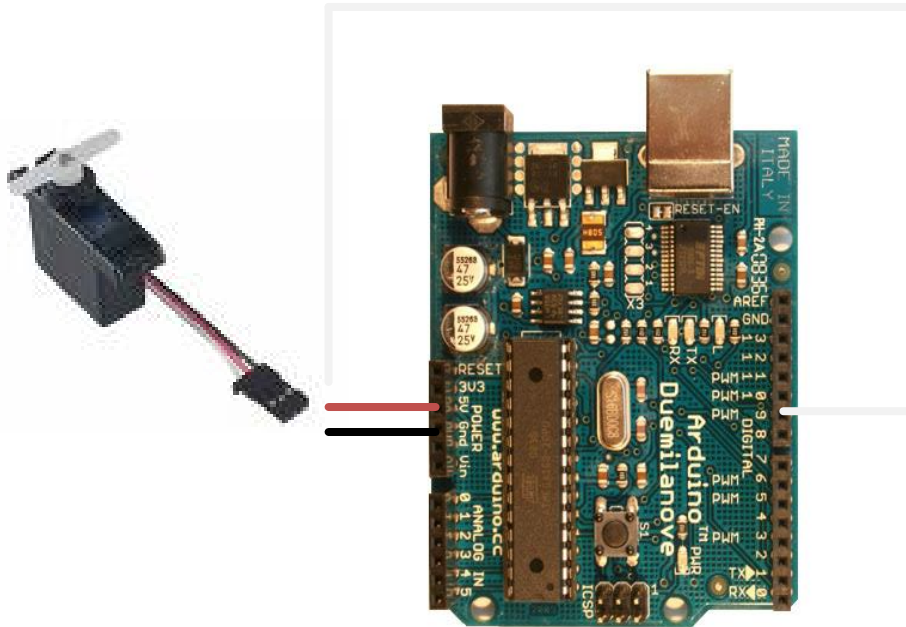
- Einleitung
- Die Komponenten
- **Erste Schritte**
- Die MicroSRCP Steuerung
- Anwendungsbeispiele

Ansteuerung von Lichtsignalen



```
int rot = 13; // LED Lichtsignal
int gruen = 12;
// Die setup() Methode, wird einmal beim
// Start des Sketches durchlaufen
void setup()
{
    // verwende die nachfolgenden Pins als Output:
    pinMode(rot, OUTPUT);
    pinMode(gruen, OUTPUT);
}
// Die loop() Methode wird immer und immer aufgerufen,
// solange das Board am Strom angeschlossen ist.
void loop()
{
    digitalWrite(gruen, HIGH); // freie Fahrt
    digitalWrite(rot, LOW);
    delay(1000); // 1ne Sekunde warten
    digitalWrite(gruen, LOW); // Stop
    digitalWrite(rot, HIGH);
    delay(1000); // 1ne Sekunde warten
}
```

Schalten von Weichen



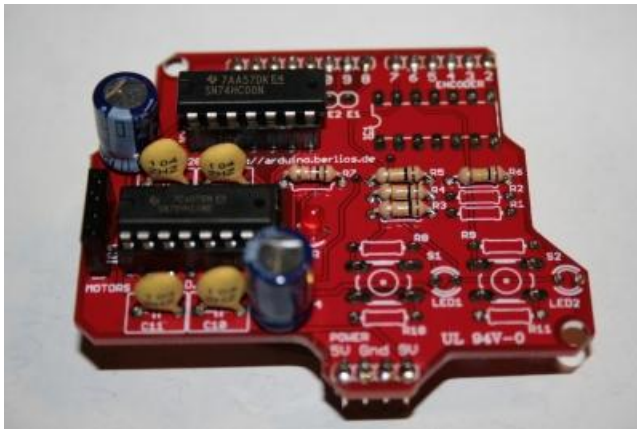
```
// Sweep
#include <Servo.h>
Servo myservo; // Erstellt ein Servo Object zur Steuerung des Servos
// Es können max. 8 Servo angeschlossen werden (Atmega128 = 48)
int pos = 0; // Variable zur Speicherung der akt. Pos. des Servos
void setup()
{
    myservo.attach(9); // Servo dem Pin 9 zuweisen.
}
void loop()
{
    for(pos = 0; pos < 180; pos += 1)
        // Servo vom linken zum rechten Anschlag
        {
            // in Schritten von 1 schalten
            myservo.write(pos);
            delay(15); // 15 Millissekunden warten
        }
    for(pos = 180; pos >= 1; pos -= 1)
        // Servo vom rechten zum linken Anschlag
        {
            // in Schritten von 1 schalten
            myservo.write(pos);
            delay(15);
        }
}
```

Ansteuerung von Lokomotiven

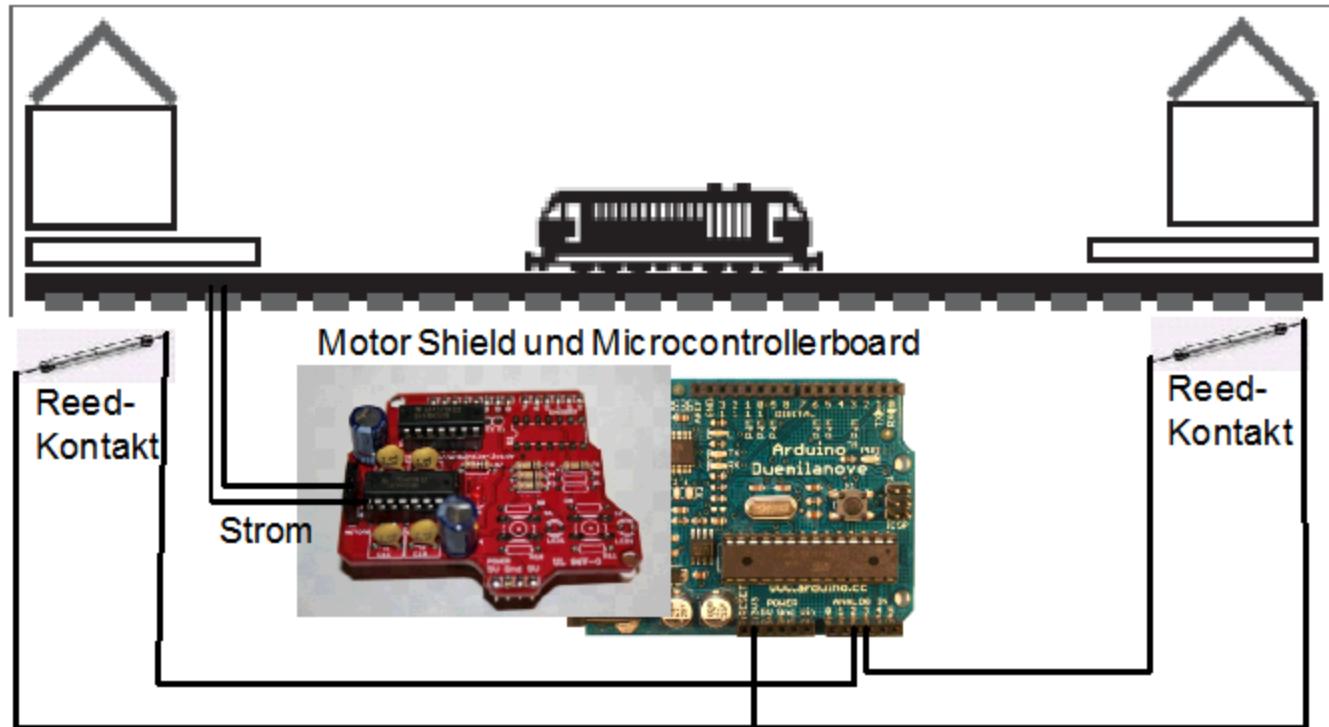


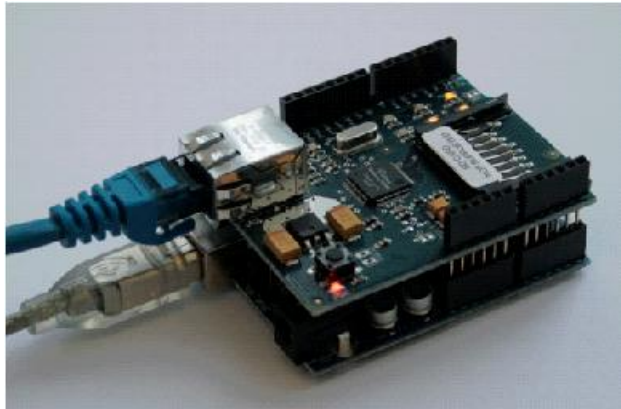
```
// Motor Shield test// by NKC Electronics// Test Motor B
int dirbpin = 12; // Fahrriichtung für Treiber B, Pin 12
int speedbpin = 9; // Geschwindigkeit für Treiber B, Pin 9
int speed = 200;
int dir = 0;
```

```
void setup()
{
    pinMode(dirbpin, OUTPUT);
}
void loop()
{
    digitalWrite(dirbpin, dir); // Fahrriichtung setzen
    analogWrite(speedbpin, speed); // Geschwindigkeit setzen
    dir = ((dir == 0) ? 1 : 0); // Fahrriichtung ändern
    delay(3000); // 3 Sekunden warten
}
```



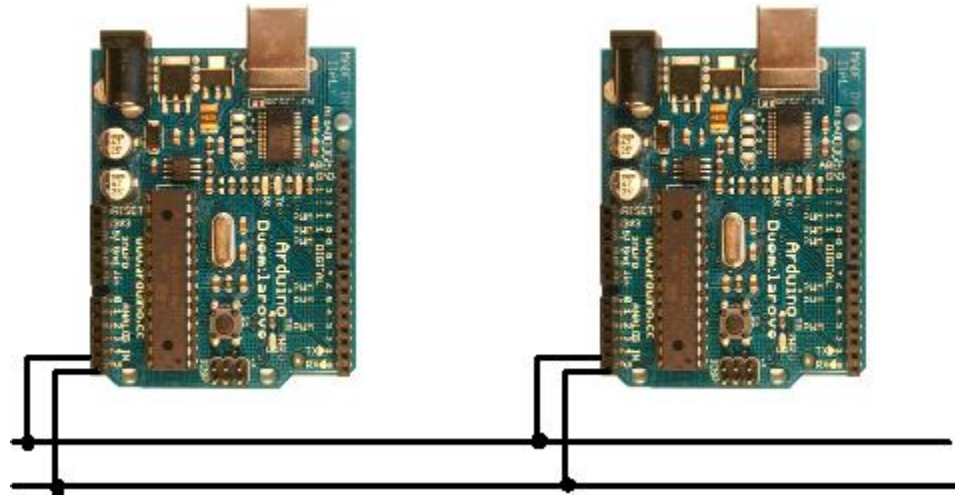
Rückmelder / Sensoren





- Mittels dem Webserver Sketch (unter File -> Examples -> Ethernet) kann die Funktionalität des Ethernetshields getestet werden. Dazu muss im Sketch die IP-Adresse (Variable ip) auf eine Adresse innerhalb des verwendeten Netzwerkes gesetzt werden.

Mehrere Boards verbinden – der I2C Bus



- Die Vernetzung der Boards wird mittels dem I2C (auch TWI genannt) Bus erreicht. Dazu muss lediglich eine gerade Verbindung von Pin 4 und Pin 5 der jeweiligen Boards hergestellt werden, z.B. mit Drahtbrücken.
- Mittels den Sketchen `master_reader` und `slave_sender` kann die Verbindung getestet werden. Bei einer korrekten Verbindung wird im Seriellen Monitor des Boards mit dem Sketch `master_reader` das Wort „hello“ ausgegeben.

Übung 2



- Suchen Sie sich eines der vorherigen Beispiele aus und implementieren Sie dieses.
- Die detaillierten Beschreibungen zu den Beispielen finden Sie in http://prdownload.berlios.de/microsrcp/MicroSRCP_T0_91.pdf

Agenda

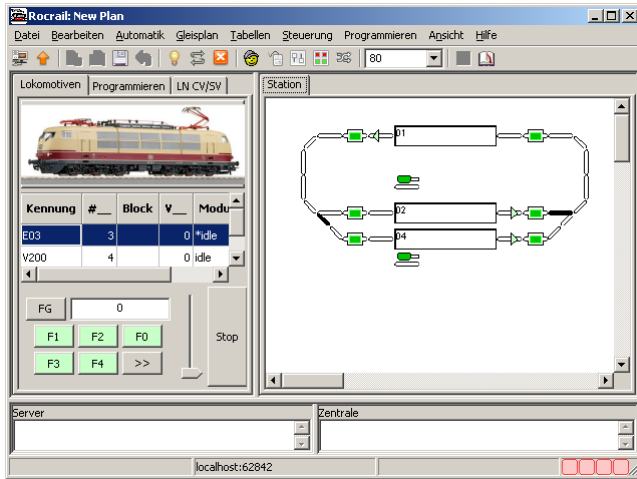


- Einleitung
- Die Komponenten
- Erste Schritte
- **Die MicroSRCP Steuerung**
- Anwendungsbeispiele

Die MicroSRCP Steuerung



PC Software RocRail



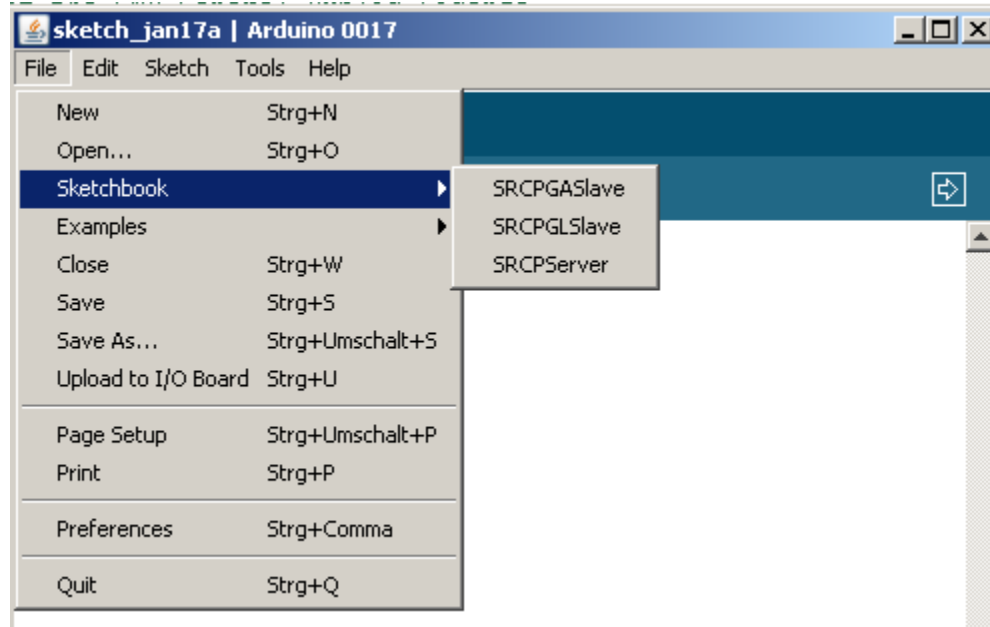
- Die MicroSRCP Steuerung baut auf den vorgängig vorgestellten Komponenten auf.
- Der Unterschied besteht darin, dass nicht mehr selber programmiert werden muss, sondern auf vorgefertigte Sketches zurückgegriffen werden kann. In Kombination mit einem Board übernimmt jedes Sketch eine bestimmte Aufgabe. Die Verbindung zwischen den Boards wird mittels dem I2C Bus hergestellt.
- Die Verbindung Boards mit dem Computer wird mittels dem SRCP-Protokoll hergestellt.

Das SRCP Protokoll



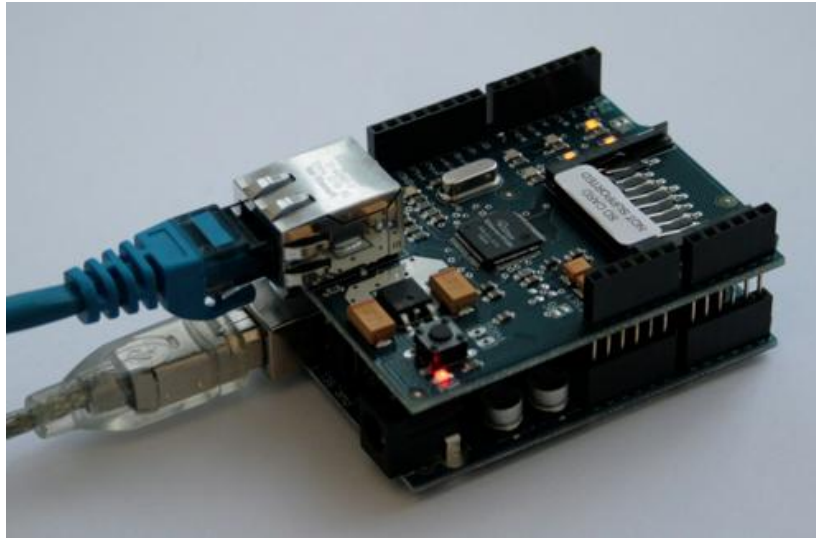
- **Das SRCP Protokoll ist u.a. unterteilt in folgende Gerätegruppen**
 - GL Generic Loco – Lokomotiven
 - GA Generic Accessoire - Zubehör z.B. Signal, Weiche
 - FB Feed Back – Rückmelder, Sensor
 - SM Service Mode – zum setzen von Konfigurationsvariablen (CV's).
 - POWER Energieversorgung
 - SERVER SRCP Server
 - SESSION SRCP Clientsession
- **Jedes Board dient zur Steuerung eines oder max. zwei Gerätegruppen.**
- **Es existieren im Moment folgende MicroSRCP Boards bzw. Sketches:**
 - SRCPServer – implementiert den eigentlichen SRCP Server, dient als Schnittstelle zwischen Computer und dem I2C Bus.
 - SRCPGASlave – dient zur Steuerung von Zubehör (GA) und zur Abfrage von Rückmeldern (FB)
 - SRCPWaveSlave – spielt Sounddateien ab. Jede Sounddatei entspricht einem GA Gerät.
 - SRCPGLSlave – dient zur Steuerung von analogen Lokomotiven (GL)
 - SRCPDCCSlave – erzeugt ein DCC Signal womit digitale Lokomotiven (GL), Weichendecoder (GA) angesprochen werden können (auch Booster genannt)

Die Sketches



- Pro Type eines Boardes existiert ein Sketch, u.a.:
 - SRCPGASlave – Steuerung für Servos, Lichtsignale, Rückmelder
 - SRCPGLSlave – Steuerung für 2 analoge Stromkreise
 - SRCPWaveSlave – Wiedergabe von Sounds
 - SRCPDCCSlave – DCC Booster
 - SRCPServer – SRCP Zentrale
- Weitere Informationen stehen im Sourcecode.

Die Zentrale

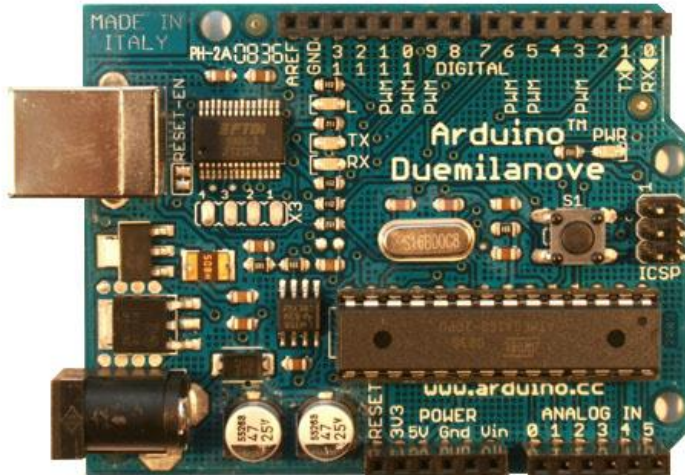


- Die Zentrale besteht aus einem Arduino mit Ethernet Shield.
- Darauf wird der Sketch SRCPServer geladen.
- Strom (5V!), TCP/IP und I2C Kabel anschließen fertig!
- Details zum Board
 - <http://arduino.cc/en/Main/ArduinoEthernetShield>

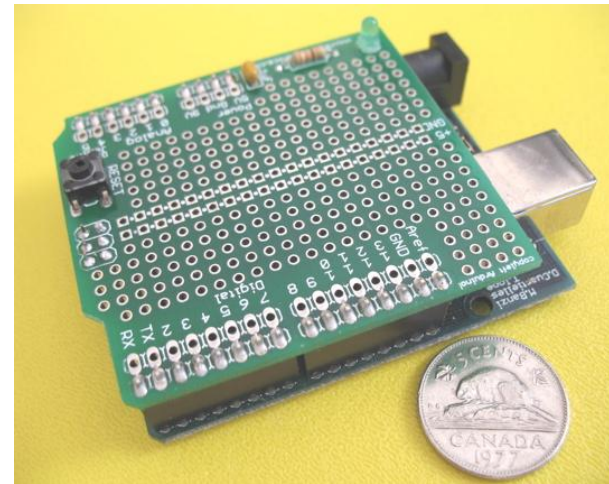
Steuerung von Servos, Lichtsignalen, Rückmelder (1)



Ein Arduino D. oder kompatibles Board



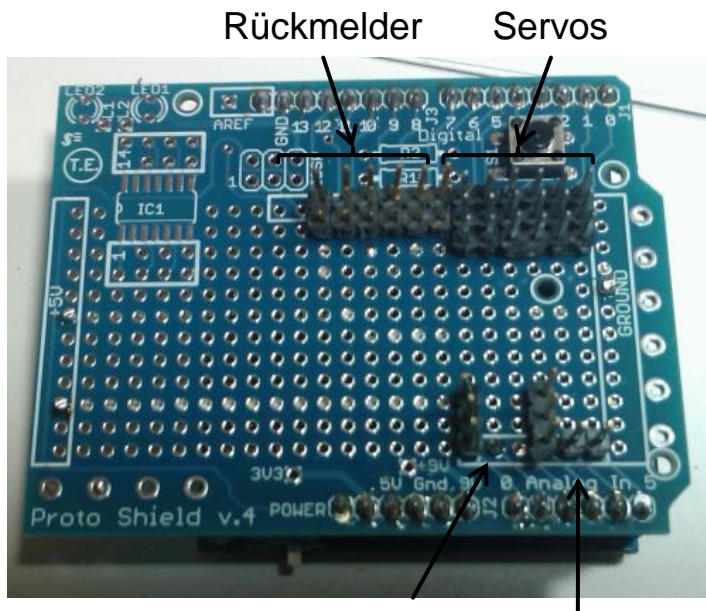
Arduino Proto Shield als Ausgangsplatine



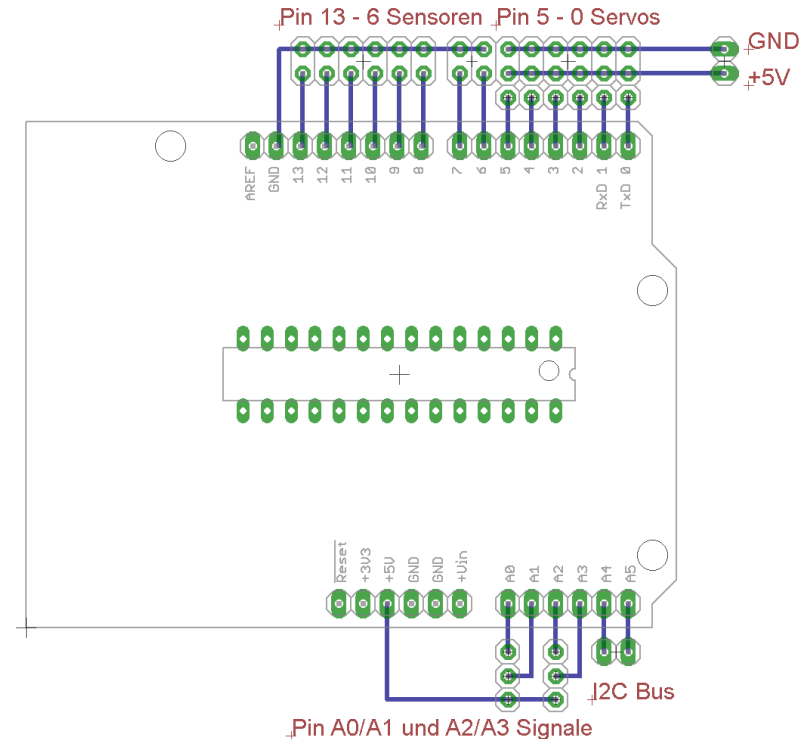
- Hier gibt es leider ein fertiges Board aber mit ein paar Lötarbeiten ist dieses einfach herzustellen.
- Der Sketch SRCPGASlave ist ausgelegt auf 6 Servos, 2 Lichtsignale und 7 (8) Rückmelder. Um diese anzuschliessen verwende ich einfache Crimp Stecker. Der Layout ist auf dem folgenden Slide ersichtlich.
- Details zum Board
 - <http://www.arduino.cc/playground/Tutorials/Protoshield>

Steuerung von Servos, Lichtsignalen, Rückmelder (2)

Mein Board



Lichtsignale und I2C Bus



- Die Belegung der Pins ist wie folgt:
 - Digital 0 – 5 für Servos (mit eigener Stromquelle versorgen!)
 - Digital 6 – 12 (13 ist leider von einer LED belegt) für Rückmelder
 - Analog 0 – 4 für Lichtsignale
- **ACHTUNG:** Relais, Lichtsignale mit Glühlampen nicht direkt anschliessen, zerstört den Microcontroller!

Übung 3



- Laden Sie den Sketch der Zentrale und steuern diese mittels der Seriellen Console.
- Die detaillierten Beschreibungen zu den Beispielen finden Sie in http://prdownload.berlios.de/microsrcp/MicroSRCP_T0_91.pdf

Agenda



- Einleitung
- Die Komponenten
- Erste Schritte
- Die MicroSRCP Steuerung
- **Anwendungsbeispiele**

Steuerung einer analogen Modelleisenbahnanlage



Lokomotiven | Programmieren | LN CV/SV

Kennung	#	Block	V	Modus
Stromkreis 1	3		0	
Stromkreis 2	4	BlockB	0	idle

FG: 0

F1 F2 F0 F3 F4 >>

Stop

Stromkreis 1

Stromkreis 2

BlockB Stromkreis 2

BlockA

Server: Zentrale

(0,0) localhost:62842

Übung 4



- Installieren Sie RocRail, bauen Sie einen SRCPServer mit SRCPGASlave (optional) und SRCPGLSlave auf.
- Steuern Sie Anlage mittels RocRail.
- Die detaillierten Beschreibungen zu den Beispielen finden Sie in http://prdownload.berlios.de/microsrcp/MicroSRCP_T0_91.pdf

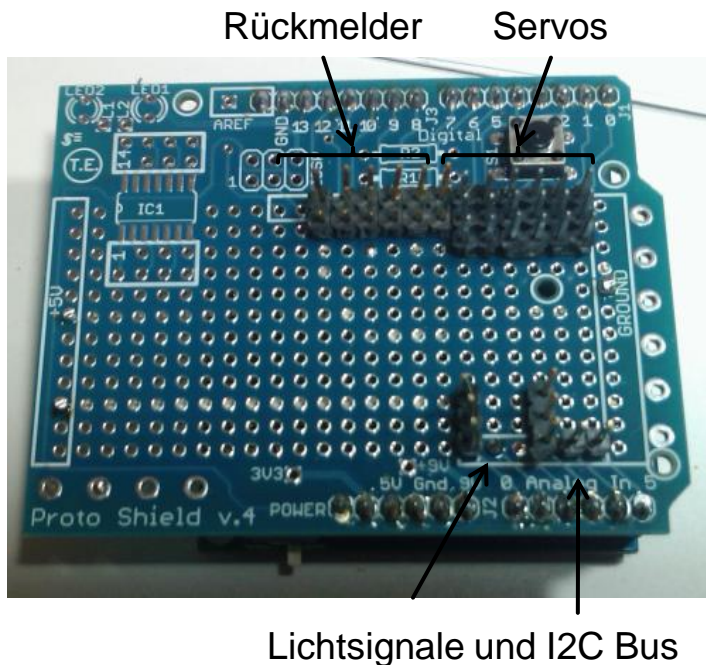
Ein Wort zum Schluss



- Es handelt sich hier um ein sehr junges Projekt, dass von einer Person (mir) erstellt wurde.
- Bitte nur Fragen zum microSRCP Projekt im Forum stellen, meine Zeit ist sehr begrenzt und die Antwort kann dauern!
- Für alle anderen Fragen rund um Arduino gibt ein ausgezeichnetes Mehrsprachiges Forum unter:
 - <http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl>

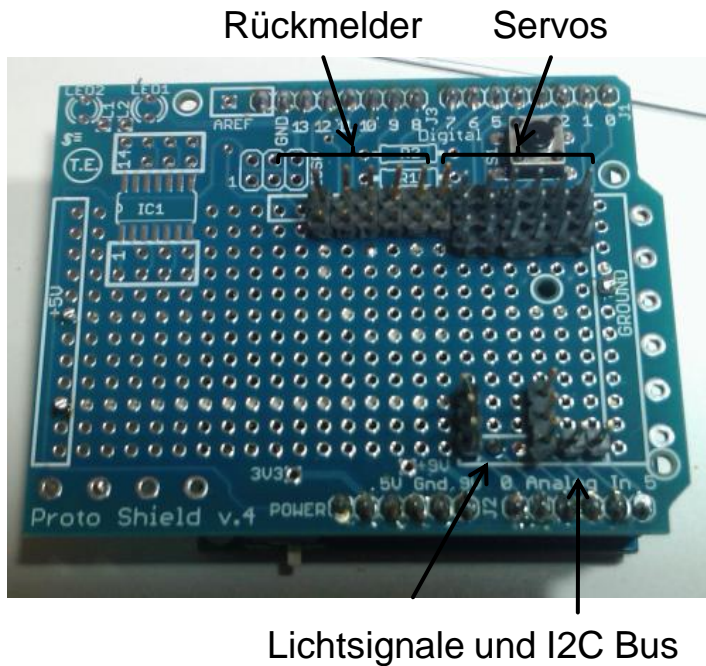
Übungen

Lichtsignal (Led's)



- Hardware
 - 1 x Arduino
 - 1 x Arduino Prototyp Platine
 - 1 x Lichtsignal (oder 2x Led)
- Anschlüsse
 - Lichtsignal an Pin 14 und 15 (Analog 0 und 1), gemeinsamer Leiter an +5V
- Sketch
 - Aufbauend auf Beispiel Blink

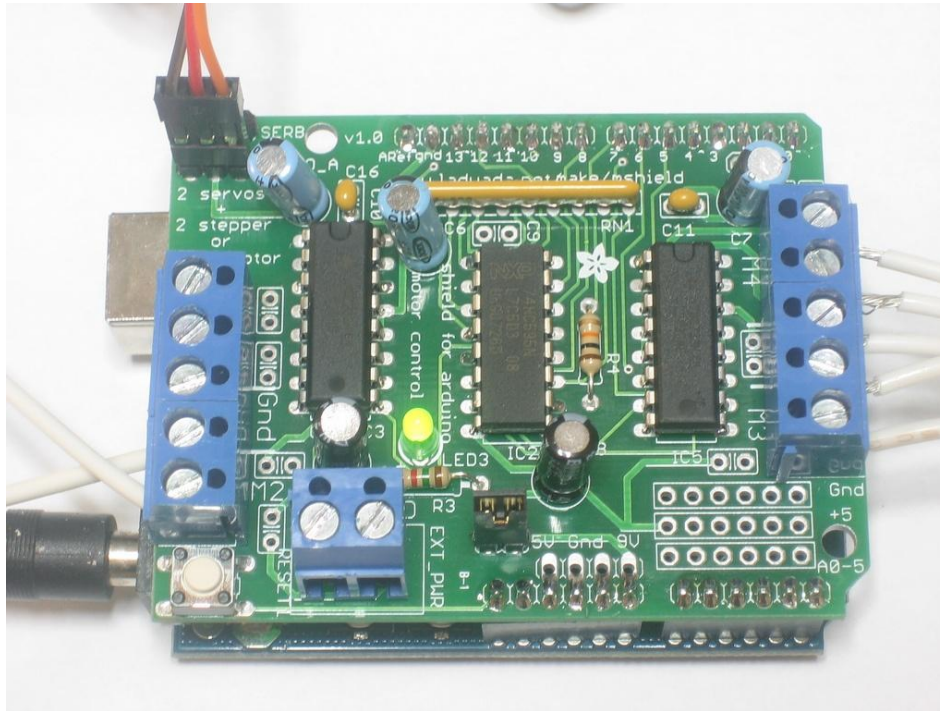
Servo (Prototyp Platine)



- Hardware
 - 1 x Arduino
 - 1 x Arduino Prototyp Platine
 - 1 x Servo
- Anschlüsse
 - Servo an Pin 2 - 5 (weisses Kabel W = gegen Pin)
- Sketch
 - Sweep



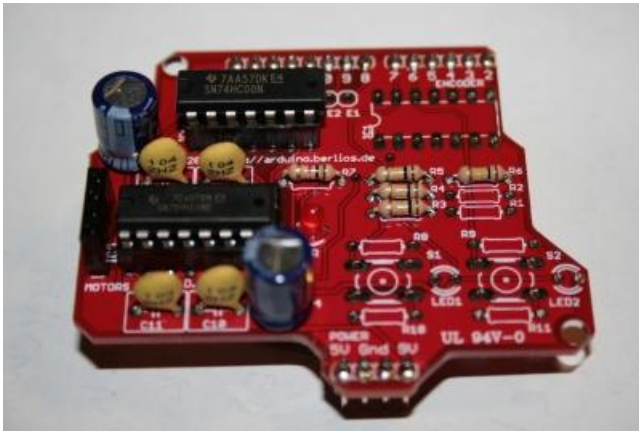
Servo (Motor Shield)



- Hardware
 - 1 x Arduino
 - 1 x Arduino Motor Shield
 - 1 x Servo
- Anschlüsse
 - Servo an Pin 9 oder 10
(weisses Kabel W = an Pin S)
- Sketch
 - Sweep



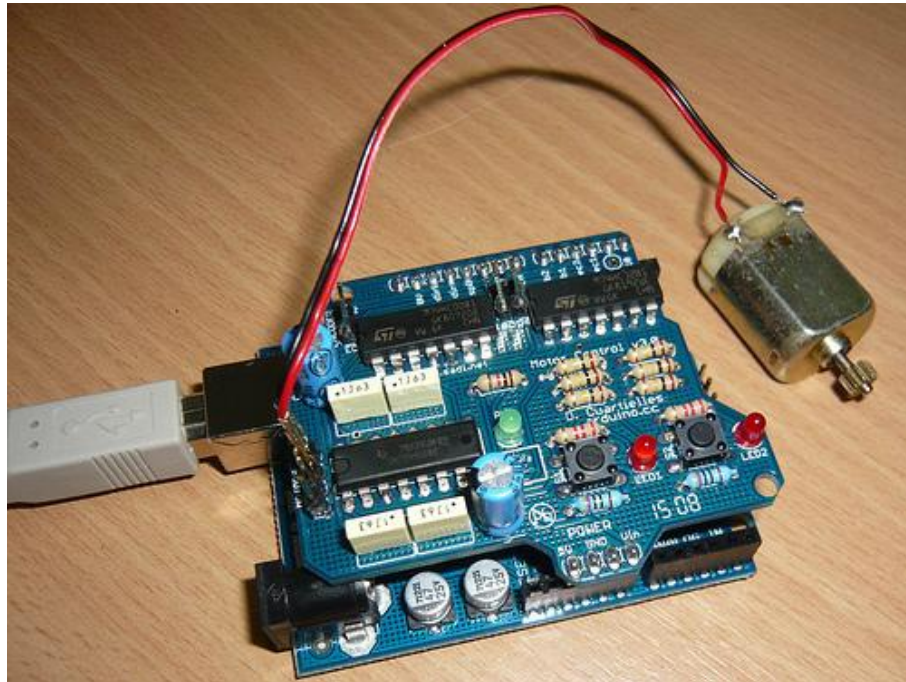
DC Motoren (Lokomotive)



Freduino Motor control shield KIT for Arduino

- Hardware
 - 1 x Arduino **mit Netzteil!!!**
 - 1 x Motorentreiber
 - DC Motor (z.B. Lokomotive)
- Anschlüsse
 - Motor 1: Pin 9 – Geschwindigkeit, Pin 12 – Fahrrichtung, an Stecker rechts
- Sketch
 - Siehe Folie 20

DC Motoren (Lokomotive)



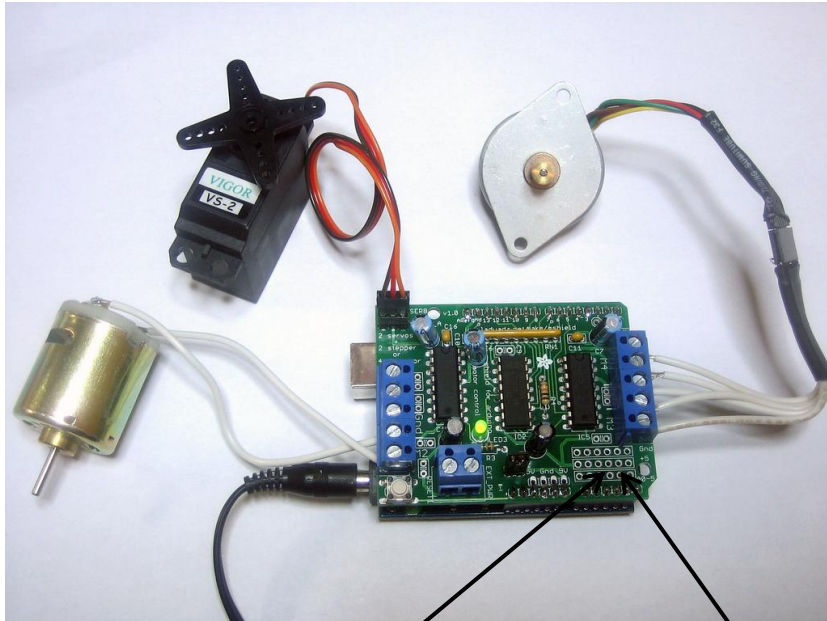
Motor Control v1.1

- Hardware
 - 1 x Arduino **mit Netzteil!!!**
 - 1 x Motorentreiber
 - DC Motor (z.B. Lokomotive)
- Anschlüsse
 - Motor 1: Pin 10 –
Geschwindigkeit, Pin 12 –
Fahrrichtung, an Stecker links
- Sketch
 - Siehe Folie 20

DC Motoren (Lokomotive)



<http://www.ladyada.net/make/mshield/index.html>

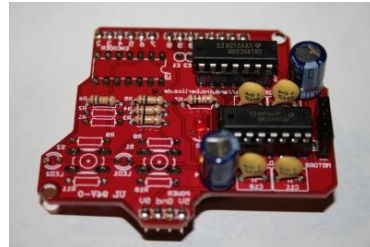
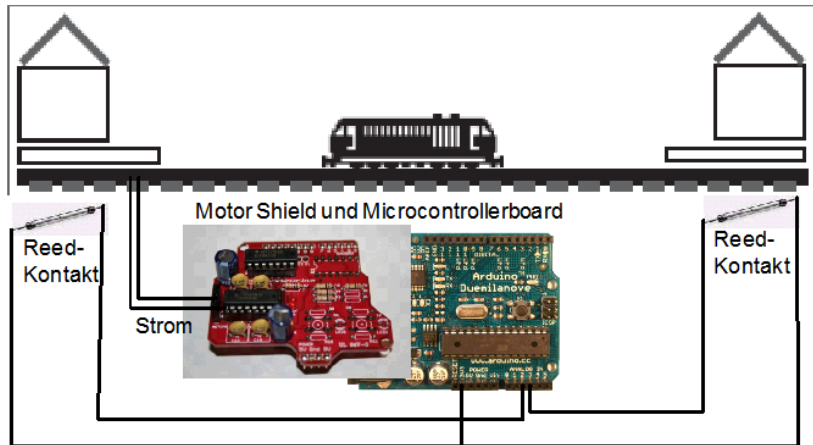


Sensoren (Pin 14 - 17)

I2C Bus

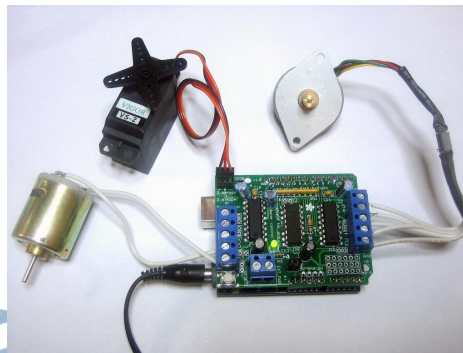
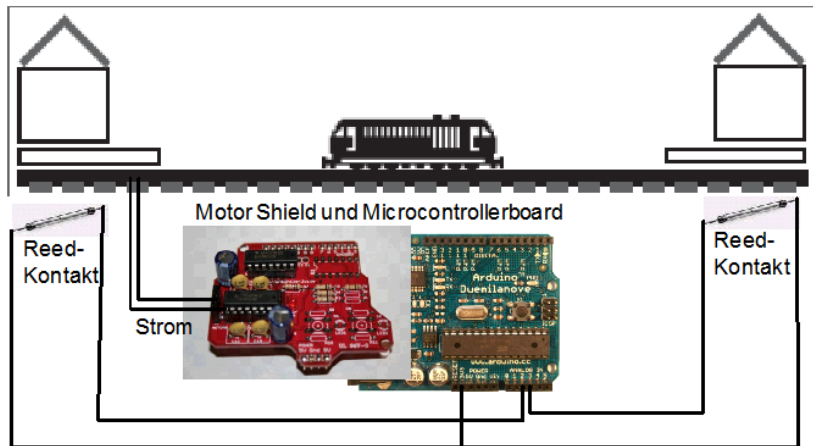
- Hardware
 - 1 x Arduino **mit Netzteil!!!**
 - 1 x Motorentreiber
 - DC Motor (z.B. Lokomotive)
- Anschlüsse
 - Servo's an Pin 9 und 10
 - Motoren ansteuern via AF_Motor Library
 - Sensoren Pin 14 – 17
 - I2C Bus
- Sketch
 - <http://www.ladyada.net/make/mshield/use.html>

Sensoren

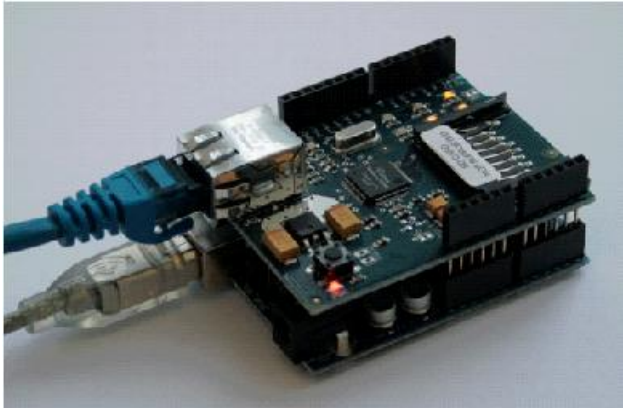


- Hardware
 - 1 x Arduino **mit Netzteil!!!**
 - 1 x Motorentreiber
 - 1 x Prototyp Platine
 - DC Motor (z.B. Lokomotive)
- Anschlüsse
 - Motor 1: Pin 9 – Geschwindigkeit, Pin 12 – Fahrriichtung, an Stecker rechts
 - Sensoren Pin 6 und 7
- Sketch
 - Siehe Textdokument, Seite 14

Sensoren



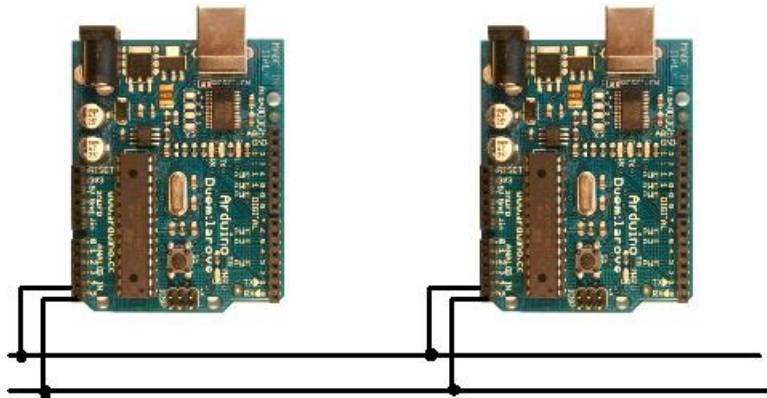
- Hardware
 - 1 x Arduino mit **Netzteil!!!**
 - 1 x Motorentreiber
 - DC Motor (z.B. Lokomotive)
- Anschlüsse
 - Servo's an Pin 9 und 10
 - Motoren ansteuern via AF_Motor Library
 - Sensoren Pin 14 – 17
 - I2C Bus
- Sketch



- Hardware
 - 1 x Arduino oder Seeduino Mega
 - 1 x Ethernet Shield
 - Netzwerkkabel
- Anschlüsse
 - RJ45-Anschluss
- Sketch
 - WebServer
- **ACHTUNG: für Seeduino Board auf Arduino Mega (1280) umstellen**



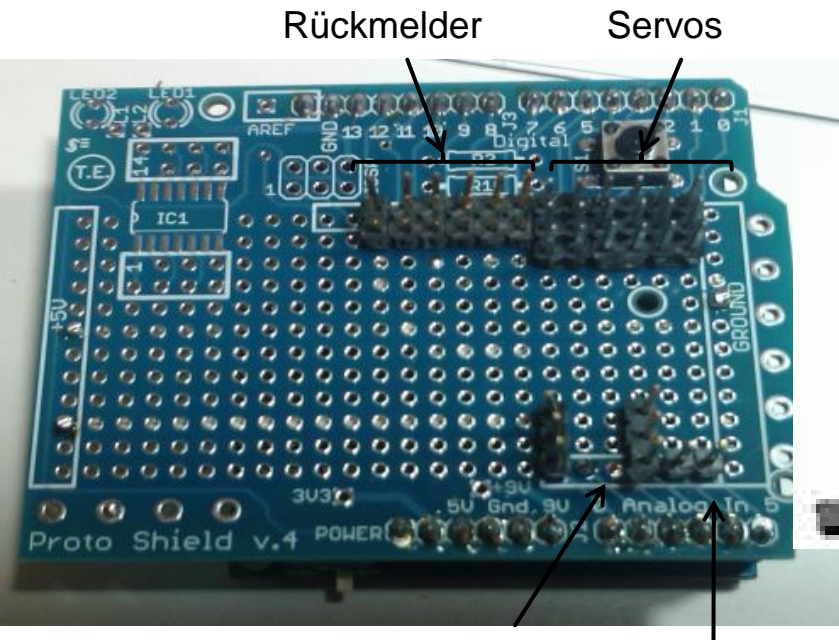
I2C Bus



- Hardware
 - 2 x Arduino oder Seeeduno Mega
 - 2adriges Kabel
- Anschlüsse
 - I2C Bus (SCL und SDA)
- Sketch
 - master_reader und slave_sender und umgekehrt
 - Ausgabe in Console von Master
- **ACHTUNG: für Seeduino Board auf Arduino Mega (1280) umstellen**

Lösungen

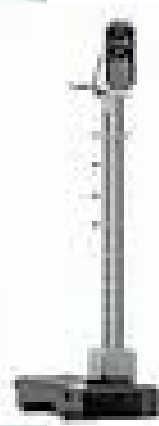
Lichtsignal (Lösung)



Rückmelder

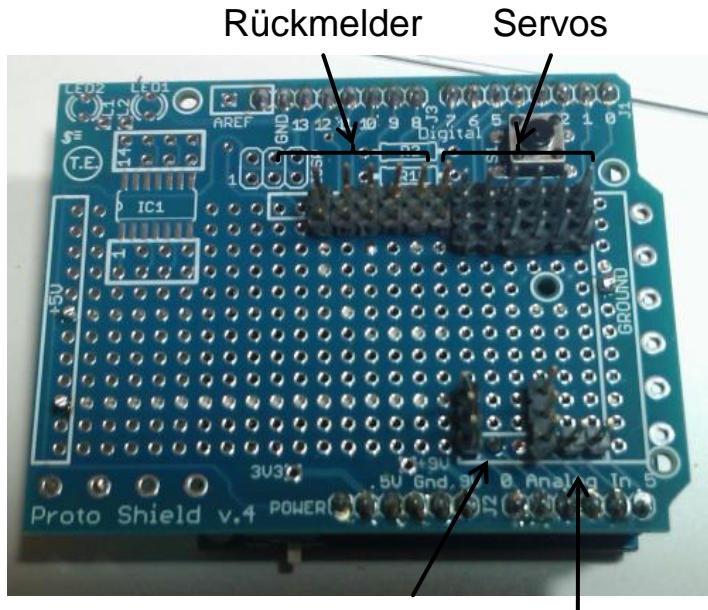
Servos

Lichtsignale und I2C Bus



```
/*  
  Blink  
  Turns on an LED on for one second, then off for one  
  second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(14, OUTPUT);  
  pinMode(15, OUTPUT);  
}  
  
void loop() {  
  
  digitalWrite(14, HIGH); // set the LED on  
  digitalWrite(15, LOW); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(14, LOW); // set the LED off  
  digitalWrite(15, HIGH); // set the LED off  
  delay(1000);           // wait for a second  
}
```

Servo (Lösung)



Lichtsignale und I2C Bus



```
#include <Servo.h>

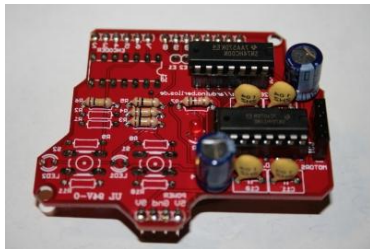
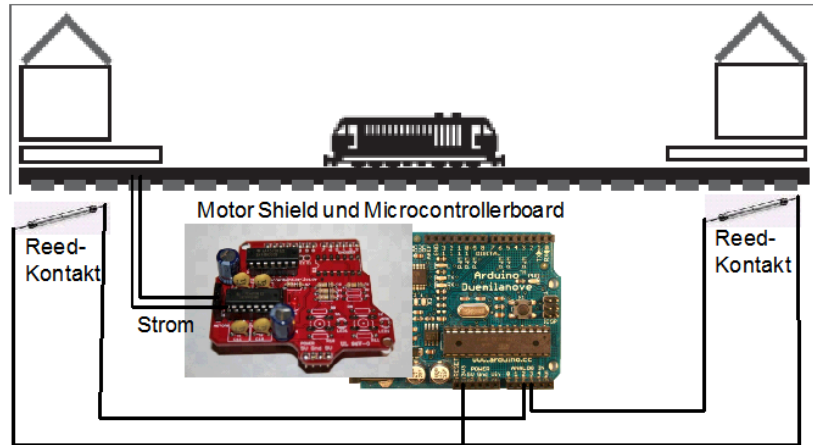
Servo myservo; // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0;   // variable to store the servo position

void setup()
{
  myservo.attach(5); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180 degrees
  {
    myservo.write(pos);           // tell servo to go to position in variable 'pos'
    delay(15);                    // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos >= 1; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);           // tell servo to go to position in variable 'pos'
    delay(15);                    // waits 15ms for the servo to reach the position
  }
}
```

Sensoren (Lösung)



```
int dirbpin = 12; // Fahrrichtung für Treiber B, Pin 12
int speedbpin = 9; // Geschwindigkeit für Treiber B, Pin 9

int sensorA = 6; // Analog Pin 2 - Sensor Haltepunkt A
int sensorB = 7; // Analog Pin 3 - Sensor Haltepunkt B
int speed = 120;

void setup()
{
  pinMode(dirbpin, OUTPUT); // steuert die Fahrrichtung
  pinMode( sensorA, INPUT ); // Sensoren A und B als Input
  pinMode( sensorB, INPUT );
  digitalWrite( sensorA , HIGH ); // Interne Widerstände setzen
  digitalWrite( sensorB, HIGH );
}

void loop()
{
  digitalWrite(dirbpin, 0); // Vorwaerts
  analogWrite(speedbpin, speed); // Geschwindigkeit
  while ( 1 ) // Loop bis SensorA aktiviert wird
  {
    if ( digitalRead( sensorA) == 0 ) // Sensor ist gegen GND geschaltet
      break;
    delay( 10 );
  }
  analogWrite(speedbpin, 0); // Stop
  delay(1000); // 1 Sekunden Wartezeit
  digitalWrite(dirbpin, 1); // Rueckwaerts
  analogWrite(speedbpin, speed); // Geschwindigkeit
  while ( 1 ) // Loop bis SensorA aktiviert wird
  {
    if ( digitalRead( sensorB) == 0 ) // Sensor ist gegen GND geschaltet
      break;
    delay( 10 );
  }
  analogWrite(speedbpin, 0); // Stop
  delay(1000); // 1 Sekunden Wartezeit
}
```

Sensoren (Lösung)

```
#include "AFMotor.h"
```

```
int sensorA = 14; // Analog Pin 0 - Sensor Haltepunkt A  
int sensorB = 15; // Analog Pin 1 - Sensor Haltepunkt B  
AF_DCMotor motor( 1 ); //
```

```
void setup()
```

```
{  
  pinMode( sensorA, INPUT ); // Sensoren A und B als Input  
  pinMode( sensorB, INPUT );  
  digitalWrite( sensorA, HIGH ); // Interne Widerstände setzen  
  digitalWrite( sensorB, HIGH );  
}
```

```
void loop()
```

```
{  
  motor.run( FORWARD );  
  motor.setSpeed( 120 );  
  while ( 1 ) // Loop bis SensorA aktiviert wird  
  {  
    if ( digitalRead( sensorA ) == 0 ) // Sensor ist gegen GND geschaltet  
      break;  
    delay( 10 );  
  }  
  motor.run( RELEASE );  
  delay( 1000 ); // 10 Sekunden Wartezeit  
  
  motor.run( BACKWARD );  
  motor.setSpeed( 120 );  
  
  while ( 1 ) // Loop bis SensorA aktiviert wird  
  {  
    if ( digitalRead( sensorB ) == 0 ) // Sensor ist gegen GND geschaltet  
      break;  
    delay( 10 );  
  }  
  motor.run( RELEASE );  
  delay( 1000 ); // 10 Sekunden Wartezeit  
}
```

