Dies ist eine Zusammenstellung der Stationsaufgaben, die am 18. Mai 2016 Lehrkräften in den Räumen der Kieler Forschungswerkstatt zur Einarbeitung in die Arduino-Programmierung angeboten worden sind. Alle Stationen waren mit Notebooks, den entsprechen Programmen und der benötigten Hardware ausgestattet.

1. Bluetooth (vgl. http://mint-unt.de/bluetooth.html)

a. Verbindung zwischen zwei Arduinos:

Beide Arduino-Boards müssen mit Bluetooth-Modulen ausgestattet sein. Diese stellen bei entsprechender Programmierung eine drahtlose Verbindung zwischen den RX und TX Pins auf den Boards her. Ein Board muss dabei als „Master“ und das andere als „Slave“ definiert sein, damit die Kommunikation klappt.

b. Verbindung zwischen Arduino und einem Bluetooth-Gerät (Tablet, Smartphone):

Tablet oder Smartphone fungieren als Master, die Befehle zum Bluetooth-Modul am Arduino übermitteln.

In einer ersten Annäherung soll versucht werden eine LED an- und auszuschalten. Für Android Tablets und Smartphones gibt es kostenfreie Apps, die das Bluetooth-Modul ansteuern können. Als Alternative kann man ein solches Programm auch selbst schreiben. Am ehesten gelingt das mit dem „MIT App Inventor 2“. Eine Anleitung gibt es hier, die fertige App LED_BLUETOOTH.apk für Android-Phones hier.

Das nachfolgende Arduino-Programm passt zur kostenfreie Android App „ArduDroid“.

```
int ledPin = 13;                // Für Pin 13 die Bezeichnung „ledPin“ zulassen
int state = 0;                  // Dieser Befehl ermöglicht variable Eingaben

void setup() {
pinMode(ledPin, OUTPUT); // ledPin als Ausgang festlegen
digitalWrite(ledPin, LOW);  // Die am ledPin angeschlossene LED ausschalten
Serial.begin(9600);             // Starten des seriellen Monitors, um Eingaben sehen zu können
}
void loop() {
if (Serial.available() > 0) {   // Wenn eine Null oder ein größerer Wert größer gesendet wird …
state = Serial.read();          //… soll er gelesen werden
}
if (state == '0') {             // Kommt vom Eingabegerät eine “0” …
digitalWrite(ledPin, LOW);  // … wird der ledPin auf Minus gesetzt (LED aus)
}
if (state == '1') {             // Kommt vom Eingabegerät eine “1” …
digitalWrite(ledPin, HIGH); // … wird der ledPin auf Plus gesetzt (LED an)
}
```

Aufgabe:

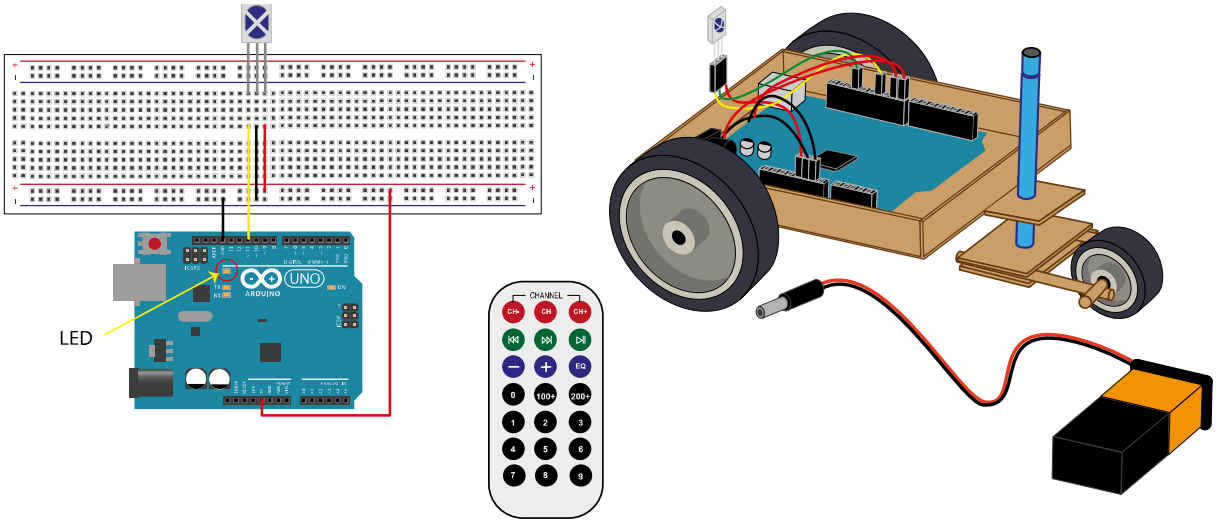
„ArduDroid“ kann mehr als nur eine Zahl senden. Verändern Sie das Arduino-Programm so, dass zwei LEDs angesteuert werden können.

2. **Bootswettbewerb** (vgl. http://mint-unt.de/bootswettbewerb.html)

Das Ziel des Wettbewerbs ist, das Boot so zu programmieren, dass es – ohne anzustoßen – möglichst dicht an den gegenüberliegenden Beckenrand fährt, eine Kurve macht und in die Gegenrichtung zurückfährt. Hierzu gibt es unterschiedliche Lösungen. Ein möglicher Programmieransatz ist auf dem Stations-Notebook zu finden. Die Programmierung führt aber noch nicht zum Ziel und muss korrigiert werden.

Es kann auch sein, dass die Bootsmotoren nicht genügend gleichmäßig laufen, sodass der Befehl „analogWrite(10, X);“ hilfreich sein kann. Die erste Zahl in der Klammer bezieht sich auf den Pin, an dem der Motor angeschlossen ist, das „X“ steht für eine Zahl zwischen 0 (minimale Leistung) und 255 (maximale Leistung) ist. Dieser Befehl funktioniert nur mit den Pins 3,5,6,9,10 und 11. Diese besonderen Pins sind auf dem Board mit einem Wellensymbol kenntlich gemacht.

3. **Steuerung eines Robot-Fahrzeugs mit einer Infrarot-Fernbedienung** (vgl.: http://mint-unt.de/robino-iii.html)



Auf dem Stations-Notebook und dem Arduino befindet sich ein Programm, das es möglich macht mit einer Fernbedienung und einen IR-Empfänger die auf dem Board neben Pin 13 befindliche LED an- und auszuschalten. Startet man den seriellen Monitor und drückt einen Knopf an der Fernbedienung, wird der dazugehörige Zahlencode angezeigt.

```
#include <IRremote.h>                 // Einbinden der Bibliothek für den IR-Empfänger
int RECV_PIN = 11;                    // Pin 11 (= RECV_PIN) wird als Eingang definiert
IRrecv irrecv(RECV_PIN);              // Die vom IR-Empfänger an RECV_PIN gesendete
                                      //Daten als „irrecv“ speichern
decode_results results;               // Daten decodieren und als „results“ speichern

void setup(){
pinMode(13, OUTPUT);                  // Pin 13 als Steuerleitung freigeben
Serial.begin(9600);                   // Seriellen Monitor starten
irrecv.enableIRIn();                  //IR-Empfänger initialisieren
}
void loop() {
  if (irrecv.decode(&results)) {      // Wenn Daten empfangen werden …
```

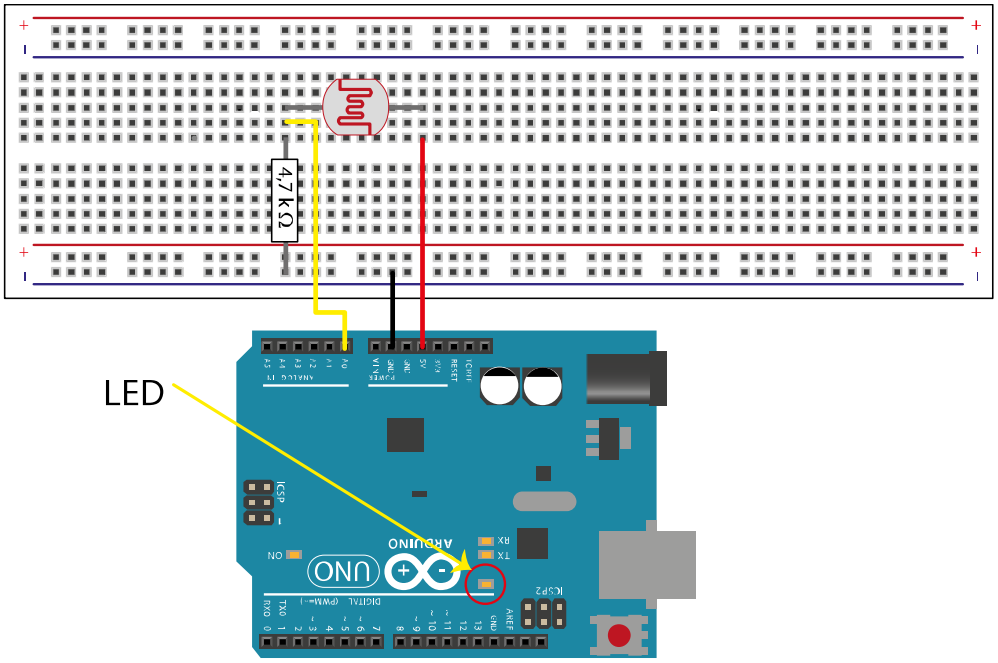
```
    Serial.println(results.value, DEC);      // … gib sie in dezimaler Schreibweise im Monitor aus
    if (results.value == 16724175)           // Wenn der Wert 16724175 empfangen wird …
      {
        digitalWrite(13, HIGH);              // … wird Pin 13 zum Pluspol
      }
     if (results.value == 16718055 )         // Wenn der Wert 16718055 empfangen wird …
     {
       digitalWrite(13, LOW);                // … wird Pin 13 zum Minuspol
     }
     irrecv.resume();                        // Bereitschaft zum Empfangen weiterer Daten
  }
}
```

Aufgabe:
Modifizieren Sie das Programm auf dem Arduino so, dass mit dem Board das an der Station vorgehaltene Fahrzeug nach links und rechts gelenkt und angehalten werden kann.

4. **Ein automatischer Dämmerungsschalter** (vgl. http://mint-unt.de/lichtsteuerung.html)
Je nach einfallendem Licht ändert ein Fotowiderstand seinen Widerstandswert: Bei wenig Licht ist der Widerstand größer, bei viel Licht geringer. Diese Eigenschaft wird genutzt, um eine Beleuchtungsanlage, hier eine LED, entsprechend den herrschenden Lichtverhältnissen ein- oder auszuschalten.

```
int Eingang = A0;                        // Pin A0 als Eingang festlegen
int LED = 13;                            // Die im Board verbaute LED verwenden
int sensorWert = 0;                      // Sensorwerte beginnen bei Null

void setup()
{
 Serial.begin(9600);                     // Seriellen Monitor starten
 pinMode (LED, OUTPUT);                  // Die eingebaute LED ansprechbar machen
}

void loop()
{
 sensorWert = analogRead(Eingang);       // Den Wert am Pin A0 auslesen
 Serial.print("Sensorwert = ");          // Den Begriff „Sensorwert =“ im seriellen
                                         // Monitor ausgeben und …
 Serial.println(sensorWert);             // … dahinter den eigentlichen Wert anzeigen

 if (sensorWert > 512 )                  // Wenn der Wert größer als 512 ist …
 {
  digitalWrite(LED, HIGH);               // … wird die LED angeschaltet
 }

 else                                    // Im anderen Fall …
```

```
 {
  digitalWrite(LED, LOW);                    // … wird die LED ausgeschaltet
 }

 delay (50);                                 // 50 Millisekunden warten bis zur neuen Abfrage

}
```

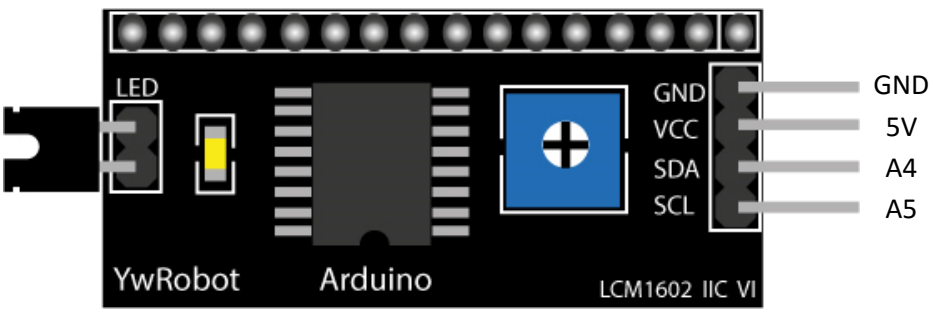


Aufgabe:

a. Schatten Sie den Fotowiderstand mehr und weniger ab und überprüfen Sie die sich verändernden Werte im seriellen Monitor. Stellen Sie den Schalter so ein, dass er bei zunehmender Dunkelheit die LED anschaltet.

b. Verändern Sie die Empfindlichkeit des Schalters so, dass er erst bei erheblicher Dunkelheit reagiert.

5. Ein LCD-Display mit I²C-Interface programmieren (vgl. http://mint-unt.de/lcd-display-1.html)



Um das Interface programmieren zu können, müssen Bibliotheken eingebunden werden.

```
#include <Wire.h>                           // Bibliothek 1
```

```
#include <LiquidCrystal_I2C_AvrI2C.h>      // Bibliothek 2
LiquidCrystal_I2C_AvrI2C lcd(0x27,16,2);   // Details des Displays festlegen: Zeichensatz,
                                           //Anzahl der Zeichen pro Zeile, Zahl der Zeilen
void setup(){
  lcd.begin();                             // Display initialisieren
  lcd.backlight();                         // Displayhintergrund beleuchten
  lcd.clear();                             // Displayinhalt löschen
  lcd.setCursor(2,0);                      // Cursor an Position 2 in Zeile 0
  lcd.print("Beispieltext");               // Text Zeile 0
  lcd.setCursor(0,1);                      // Cursor an den Anfang Zeile 1
  lcd.print("1234567 ABCDEFG");            // Text Zeile 1
}
void loop(){
}
```

Aufgabe:

Verändern sie Position und Inhalt der Texte auf dem LCD-Display.