

Dokumentation der Arduino Photokommunikation

von Tom Arlt, Florian Körwer, Florian Meyer, Henning Müller

Hochschule Niederrhein FB03, der 03.01.2021

Links

- [GitHub link](#)
- [IDE3 link](#)

Einleitung

Unsere Aufgabe war es als Gruppe für das Erstsemesterprojekt, im folgenden "ESP" genannt, zwei Arduinos mithilfe von Leds und Photoresistoren miteinander kommunizieren zu lassen und unsere Arbeit hiermit zu dokumentieren.

Die notwendigen Bauteile wurden uns hierfür pro Person von der Hochschule zur Verfügung gestellt:

- 1x Arduino Uno
- 2x LM392 mit Photoresistor und Potenziometer
- 1x Breadboard
- 2x 330 Ohm Widerstände
- verschied. LEDs
- verschied. Kabel für das Breadboard
- ~~1x Potenziometer~~
- ~~1x Ultraschallsensor~~

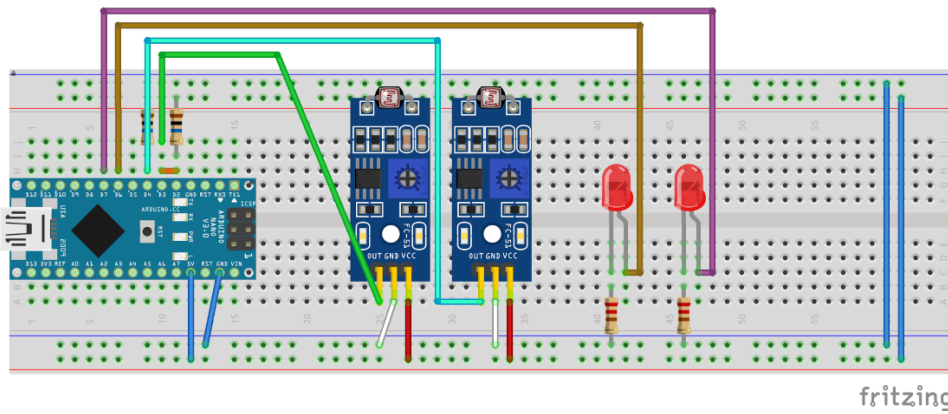
Da der Wissensstand innerhalb der Gruppe zum Thema Arduino sehr verschieden war, wurde nur zeitgleich und immer gemeinsam gearbeitet. Die Teilnehmer, welche bereits einiges an Erfahrung mit sich brachten, fungierten eher als Lehrperson, die den anderen das Nötige Wissen überschaubar vermittelt haben, sodass alle dann gemeinsam die Aufgabe bearbeiten konnten.

Umsetzung

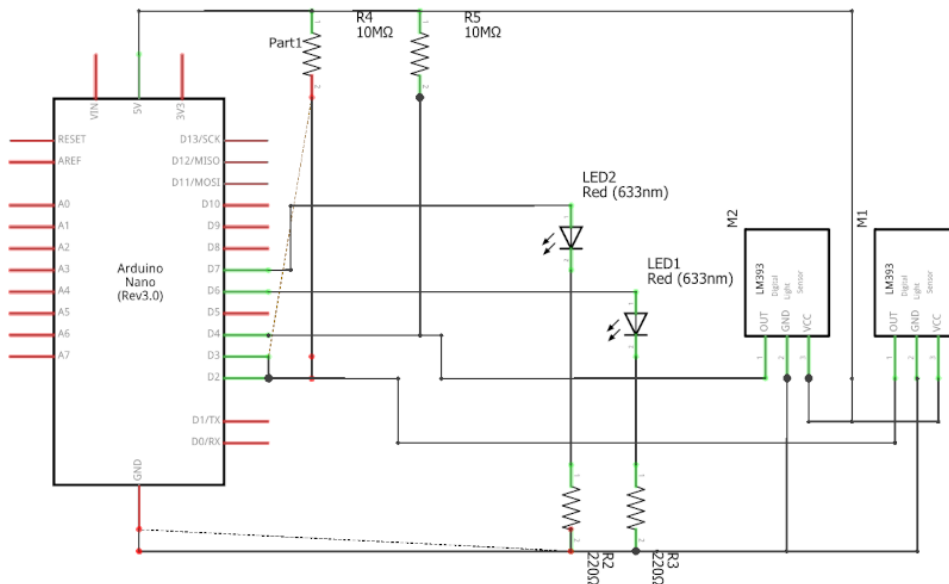
Die Kommunikation der beiden Arduinos basiert auf bereits existierenden und in jedem Microchip hardwareimplementierten Protokoll namens **I2C**. Bei diesem Protokoll ist in der Regel ein "Master" deklariert und mehrere "Slaves" welche alle gemeinsam an zwei Datenleitungen angeschlossen sind: "Data" und "Clock". Immer wenn die Spannung auf "Clock" von 0V auf 3.3/5V ändert wird auf "Data" der Zustand (Bit) abgelesen, bis sich letztendlich, nach einigen Takten ein Byte (8 Bit) ergeben. Das Standardprotokoll arbeitet eigentlich mit Adressen um zu differenzieren an wen und von wem die Nachricht gesendet wurde.

Unser Protokoll übernimmt dabei nur das Prinzip von "Data" und "Clock", somit kann ein Empfangsmodul nur mit einem Sendermodul verbunden werden und vice versa.

Hier einmal die Schematics in der Arduino Schematic Software [fritzing](#)



fritzing



fritzing

In dieser Schematic wurde der zum Arduino Uno funktionsidentische Arduino nano verwendet und die Grafik übersichtlicher und kompakter zu gestalten. Beide Arduino-versionen besitzen denselben Prozessor (ATmega328P) und dieselben Pins. Die einzigen Unterschiede beider sind der Formfaktor und die Funktion des Nanos, dass er direkt auf ein Breadboard gesteckt werden kann. Dargestellt wurde hier ein einzelner Arduino, der in der Lage ist sowohl zu Empfangen als auch zu senden, wenn das passend aufgebaute Gegenstück existiert.

Verwendung

(Übersetzt und aufbereitet aus [Readme.md](#)) Um unser Projekt zu verwenden, müssen zwei Schaltungen aufgebaut sein, wobei mindestens zwei LEDs und zwei Empfänger aufgebaut sein müssen um zumindest die Einseitige Kommunikation testen zu können. Zu beachten ist, dass jedes LED-LM392 Paar zu allen anderen Paaren abgeschirmt sein sollte, da es sonst zu Interferenzen kommen kann. Zudem muss das Potenziometer des LM392 eingestellt werden nur die dazugehörige eingeschaltete LED zu erkennen und nicht das Umgebungslicht. Bei der Erstellung der LED-LM392 Paare ist auf die Unterscheidung zwischen "Data" und "Clock" zu achten:

```

Arduino 1      <-> Arduino 2
sender_clock    <-> reciever_clock
sender_data     <-> reciever_data
  
```

```
reciever_clock  <-> sender_clock  
reciever_data   <-> sender_data
```

Ist das Programm einmal auf beiden Arduinos hochgeladen, ist die Arduino IDE (o.ä.) nicht mehr notwendig. Nun kann über einen Seriellen Monitor, wie zum Beispiel der Integrierte der [Arduino IDE](#) oder [Putty](#) über den Arduino kommuniziert werden. In diesen Tools muss noch der Port, über den der Arduino angeschlossen ist, angegeben werden, sowie die Baud Rate des Seriellen Monitor, welche in [config.hpp](#) eingesehen und verändert werden kann, der Standard ist hier [9600](#).

Link zum Aktuellen Programmcode: [Github](#), [IDE3](#)

Link zum aufbereiteten/alternativen Code (entwickelt von Tom Arlt): [Github](#), [IDE3](#)

Reflexion

Das Entwickeln der Strategie für die Kommunikation war sehr einfach und stand sofort nach dem Lesen der Aufgabenstellung. Auch das Entwickeln eines grundsätzlichen Gerüsts war straight forward. Das Debuggen sowie ein ordentlicher Aufbau der Schaltung war dann die größere Herausforderung. Da durch ständig wechselnde Lichtverhältnisse der Photoresistor nicht immer wie gewünscht funktionierte, verzögerte sich das Programm-Debuggen ungemein. Erst nach vielen Stunden rumgrübelns, rumschrauben und ausprobieren, stellte sich als einzige zuverlässige Methode eine eigene Blackbox für jedes LED-LM392 Paar heraus.

Zusammenfassung