

Open Source Roboter Plattform

Lukas Reichwein
Yves Ehrlich
Nick Gnoevoj

University of Applied Science Fulda — 5. Februar 2020

Inhaltsverzeichnis

1	Vorwort	2
2	SPI	3
3	Funksteuerung	4
3.1	RF24	4
4	Arduino Libraries	5
5	Joystick Integration	6
6	Motorsteuerung	7
7	Thermosensor	8
8	Ultraschallsensor	9
9	Beispiele für Spezielle LaTeX Strukturen	10

1 Vorwort

Motivation Eine Plattform bieten ist etwas was momentan sehr stark im Trend liegt, sei es im Software oder im Hardware Bereich. Im Softwarebereich zeigt sich dies meist durch opensource Bibliotheken, welche möglichst variable einsetzbaren Code für jeden frei zugänglich machen.

Ein solches Projekt war auch von einem der Projektmitglieder (Yves Ehrlich) als Privates Projekt geplant und so kam die Überlegung dies innerhalb des Modules Embedded Networking zu wählen.

Basis des Projektes Als Basis des Projektes dient einer schon bereits von Yves Ehrlich angefertigter Arduino Nano Shield samt Code, [1] welcher als Fernsteuerung verwendet wird.

Ziel des Projektes Ziel des Projektes ist eine ferngesteuerte, Opensource basierende Roboterplattform.

2 SPI

3 Funksteuerung

Wie schon zuvor erwähnt wird für die Basis der Funksteuerung das Arduino Shield verwendet, welches mit einem RF24 Chip erweitert wurde.

3.1 RF24

Die Opensoucre Bibliothek RF24 [?] diene als Codebasis für die Funksteuerung. Da diese Bibliothek bei korrekter Verwendung genau auf die Kommunikation zwischen zwei nRF24L01 Chips abgestimmt ist. Zur Verwendung der Bibliothek muss sie nur inkludiert und Instanziiert werden dabei werden die Beiden Pins CE und CSN für das Hardware-SPI konfiguriert.

```
RF24 initialisieren

#include <RF24.h>
RF24 radio(A2, A3); // CE, CSN
```

Damit sind bereits Sämtliche Konfigurationen für die Hardware-SPI Kommunikation zwischen Arduino nano und dem nRF24L01 erledigt. Kommunizieren zwischen zwei dieser Setups wird dann durch die Funktionen read und writre, jedoch muss vor dem Start einer Kommunikation die Methode begin() aufgerufen werden.

```
RF24 initialisieren

//An der Sender Seite
radio.begin();
radio.write(&payload, sizeof(payload));

//An der Empfaenger Seite
radio.begin();
if (radio.available()) {
    radio.read(&payload, sizeof(payload));
    //Payload weiter verarbeiten.
}
```

4 Arduino Libraries

5 Joystick Integration

6 Motorsteuerung

7 Thermosensor

8 Ultraschallsensor

9 Beispiele für Spezielle LaTeX Strukturen



Info: benutze den Info block um wichtige Informationen hervorzuheben.

Algorithm 1: FastTwoSum

Input: (a, b) , two floating-point numbers

Result: (c, d) , such that $a + b = c + d$

if $|b| > |a|$ **then**

 | exchange a and b ;

end

$c \leftarrow a + b$;

$z \leftarrow c - a$;

$d \leftarrow b - z$;

return (c, d) ;

hello.py

```
#!/usr/bin/python

import sys
sys.stdout.write("Hello World!\n")
```

Command Line

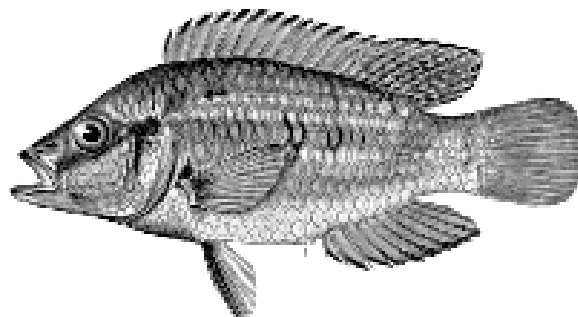
```
$ chmod +x hello.py
```

```
$ ./hello.py
```

```
Hello World!S
```



Notice: Warnungen könnten auch nützlich sein, immerhin braucht der RF24 3.3V und nicht 5V



Literatur

[1] Yves Ehrlich. <https://gitlab.informatik.hs-fulda.de/fdai5253/nanogame>. Repository Nano Game.