

# PROJEKTARBEIT

im Studiengang BEL4

Lehrveranstaltung Embedded Systems Software Design

## CNC-Machine

Ausgeführt von: Alexander Rössler  
Johannes Wimmer

Matrikelnummer: 1110254020  
1110254003

Koorosh Pirghaibi

1110254002

Begutachter: Dr. Martin Horauer

Wien, 24. April 2013

# Inhaltsverzeichnis

<b>1</b>	<b>Projektbeschreibung</b>	<b>1</b>
1.1	Ziel . . . . .	1
1.2	Funktionsweise . . . . .	1
<b>2</b>	<b>Umsetzung</b>	<b>2</b>
2.1	Infrarot Sender und Empfänger . . . . .	2
2.2	WLAN . . . . .	2
2.3	Software . . . . .	2
2.4	Desktop-Anwendung . . . . .	2
2.5	Schaltung . . . . .	3
2.6	Gehäuse . . . . .	4
<b>3</b>	<b>Kommunikation</b>	<b>5</b>
3.1	Kommunikation zwischen dem PC und dem Microcontroller . . . . .	5
3.2	Kommunikation zwischen dem Microcontroller und der Fräsmaschine . . . . .	5
<b>4</b>	<b>Zukunft</b>	<b>6</b>
4.1	Software . . . . .	6
4.2	Hardware . . . . .	6
	<b>Abkürzungsverzeichnis</b>	<b>7</b>

# 1 Projektbeschreibung

## 1.1 Ziel

Ziel des Projekts ist es ein Gerät zu entwickeln welches Infrarot-Signale von Fernbedienungen aufzeichnen und wiedergeben kann. Das Gerät soll per WLAN über Computer oder Handy bedieht werden können.

## 1.2 Funktionsweise

Handelsübliche Infrarot-Fernbedienungen senden digital codierte Signale per Sende-Diode aus welche am Empfänger wieder decodiert werden. Die Signale werden in 30-40kHz PWM Pulsen gesendet und am Empfänger gefiltert. Die meist verbreitete Frequenz liegt bei 38kHz, deshalb haben wir den Empfänger auch so gewählt. Da viele verschiedene Codierungsformate verbreitet sind, entschieden wir uns dafür die Signale nicht zu decodieren sondern "analog" aufzunehmen und wiederzugeben, somit können beliebige Geräte ferngesteuert werden ohne die Codierung zu kennen (auch etwaige zukünftige Codierungen sind somit steuerbar).

## 2 Umsetzung

### 2.1 Infrarot Sender und Empfänger

Als Empfänger verwenden wir einen TSOP. Dieser nimmt uns die Filterung und Entstörung der IR-Signale ab, am Ausgang wird Logisch 1 oder 0 ausgegeben. Die TSOP Module haben eine sehr hohe Empfindlichkeit und werden vor allem in hochwertigeren Produkten verwendet.

Als Sender werden herkömmliche 950nm IR-Dioden verwendet. Diese sind auf Puls-Betrieb ausgelegt und erfordern somit sehr hohe Ströme (200mA-1A). Da die Stromversorgung über USB erfolgen soll und dieser maximal für 500mA ausgelegt ist, brauchen wir einen Kondensator um kurzzeitig den benötigten Strom bereitzustellen. Um die Diode zu Schalten wurden verschiedene Varianten probiert. Zuerst versuchten wir dies mit einer FET zu realisieren, aus uns nicht erklärlichen Gründen funktionierte dies nicht konsistent. Am Ende erwies sich eine Darlington-Schaltung als am brauchbarsten (da der  $\mu C$  nur wenige mA Ausgangstrom liefert ist eine direkte Ansteuerung des Leistungstransistors nicht möglich).

### 2.2 WLAN

Für die Anknüpfung an WLAN Netzwerke haben wir ein WLAN-Modul von Roving Networks gewählt. Die Ansteuerung erfolgt über UART.

### 2.3 Software

Die Software wurde strukturiert aufgebaut. Die Hardware Interfaces werden jeweils per Library angesteuert welche in einen Systemunabhängigen und einen Treiberpart besteht. Umgesetzt wurden dabei Libraries für Timer, UART, PWM, LEDs, Button, GPIO, Pincon, IAP und WiFly Modul. Die Entwicklung der Libraries stellte somit einen großen Part der Entwicklung der gesamten Anwendung dar.

Die Aufzeichnung der Signale erfolgt über eine Kombination aus GPIO-Interrupt und Timer, da die Timer-interne Capture Compare Schnittstelle nicht funktionieren wollte. Die Wiedergabe der Daten erfolgt über eine Kombination aus PWM und Timer, mit dem Timer wird das PWM-Signal aus- und eingeschalten.

Um Einstellungen im Mikrocontroller internen Flash-Speicher speichern zu können muss auch eine IAP-Library implementiert werden.

Die Interaktion mit dem Benutzer erfolgt zum einen über die Software am Computer und zum Anderen über die Buttons am Gerät. Zum Beispiel ist mit diesen ein Wechseln zwischen Adhoc und Infrastructure-Modus möglich.

### 2.4 Desktop-Anwendung

Die Desktop Anwendung wurde mit Qt/C++ geschrieben und ist in 2 Teile gesplittet: Einen GUI-Part und einer Library welche die Kommunikation zum  $\mu C$  übernimmt. Der GUI-Part dieht vor allem Testzwecken, hierbei wurde unter anderem die Simulation einer echten Fernbedienungen umgesetzt.

Die Library kommuniziert mit dem  $\mu\text{C}$  per Textbefehlen, somit wäre eine Steuerung auch ohne Library ohne Probleme möglich. Signale können gesendet und aufgezeichnet werden, wahlweise per WLAN oder serieller Verbindung. Das Senden von Daten über die Serielle Schnittstelle ist Aufgrund eines Bugs in der Bibliothek für die Serielle Schnittstelle nicht möglich.

## 2.5 Schaltung

Aufbau auf Lochrasterplatine:

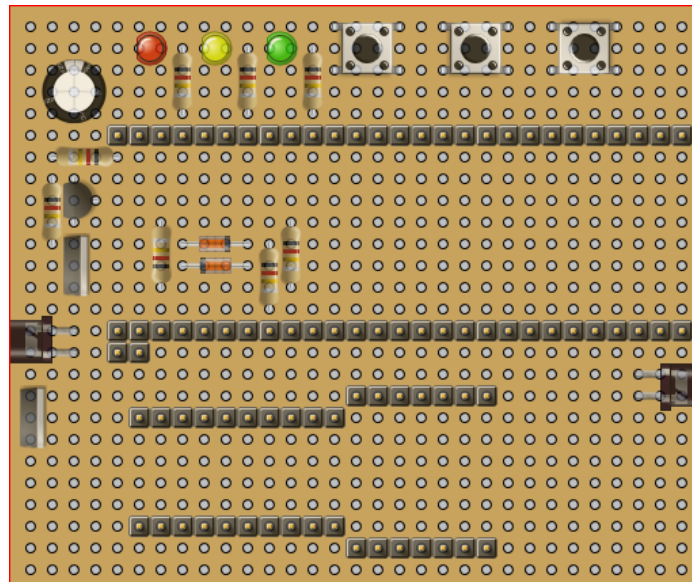


Abbildung 2.1: Vorderseite

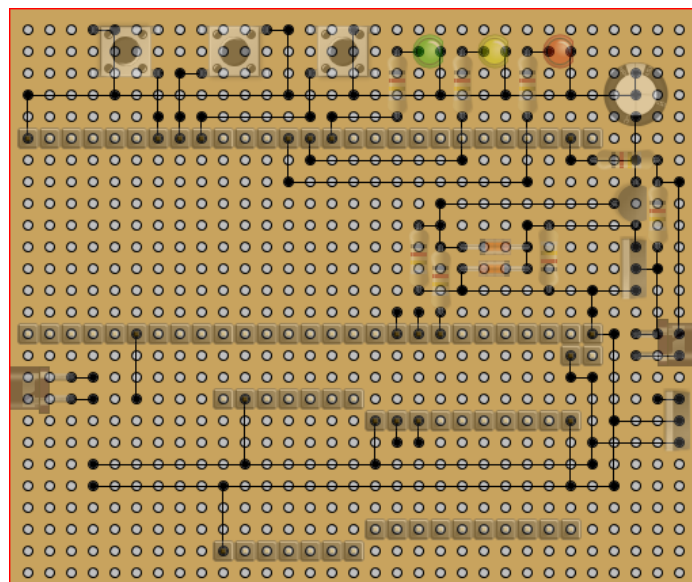


Abbildung 2.2: Rückseite

## **2.6 Gehäuse**

Aufgrund eines vorhandenen 3D-Druckers konnte auch ein Gehäuse für den Prototypen umgesetzt werden.

# 3 Kommunikation

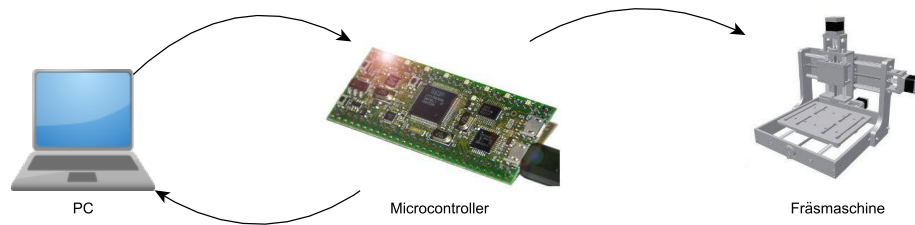


Abbildung 3.1: Kommunikation

## 3.1 Kommunikation zwischen dem PC und dem Microcontroller

## 3.2 Kommunikation zwischen dem Microcontroller und der Fräsmaschine

## **4 Zukunft**

### **4.1 Software**

Eine anwenderfreundlichere Bedienoberfläche ist zwingend notwendig. Weiters sollte eine Anwendung für die mobile Benutzung entwickelt werden. Auch ein Command-Line Client wäre sinnvoll.

### **4.2 Hardware**

Eine Erweiterung mit Funkmodulen für das 433kHz und das 844kHz Band wäre denkbar um auch die zahlreich vorhandenen Geräte aus diesen Bereich steuern zu können. Nach dem Motto: "One remote to control them all"



# Abkürzungsverzeichnis

PWM	Pulsweitenmodulation
OPV	Operationsverstärker