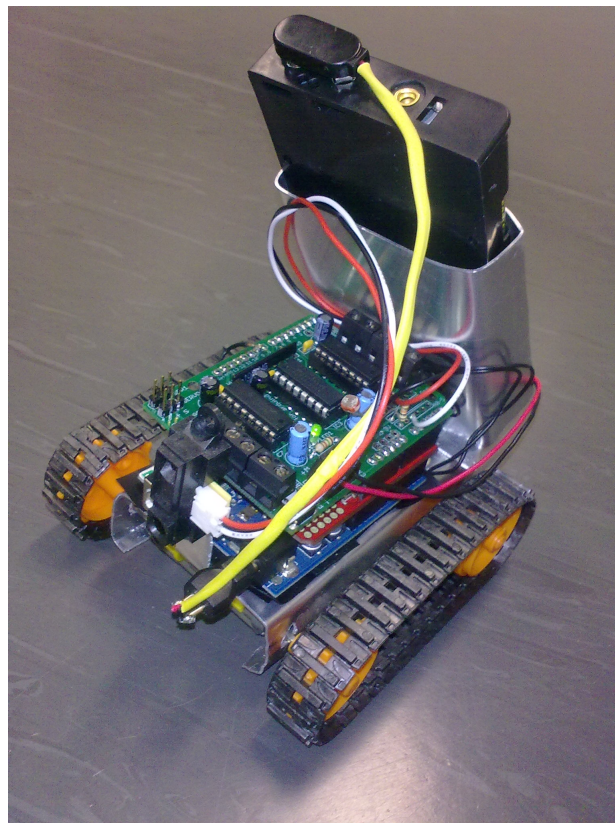


ROBOiNO Projekt

erstellt von: Stephan Bergemann
Matrikelnummer: 521034
Studiengang: Angewandte Informatik
Fachbereich: Wirtschaftswissenschaften II
Hochschule: Hochschule für Technik und Wirtschaft Berlin
letzte Änderung: 23. Juli 2010



Inhaltsverzeichnis

1 Idee	2
2 Vorüberlegungen	2
2.1 nötige Hardware	2
2.2 zu entwickelnde Software	3
2.3 Kommunikationsprotokoll	3
3 Umsetzung	4
3.1 eigene Shields löten und ätzen	4
3.2 Herausforderungen	6
3.2.1 WiFly GSX WLAN	6
3.2.2 Ethernet Library	6
3.2.3 Motoransteuerung	7
3.2.4 Roboter Chassis	8
4 geschaffene Lösung	8

1 Idee

Die ursprüngliche Idee zum Projekt war ein Spielkonzept, welches 2 Roboter vorsah, sowie eine Fernbedienung für jeden. Mittels dieser sollte der Roboter je Spielpartei möglichst schnell nah an eine Wand gefahren werden. Über einen Lichtimpuls von einem Roboter zum anderen sollte bei diesem die Steuerung invertiert werden.

Aufgrund der hohen Komplexität eines solchen Szenarios beschränkte ich mich bei meinem Projekt daher zunächst auf die Steuerung eines Roboters per WLAN und Ethernet.

2 Vorüberlegungen

2.1 nötige Hardware

Um einen Roboter per WLAN zu steuern, braucht es für den Roboter und dessen Fernbedienung bestimmte Hardware. So ist der Roboter mit Motoren, einem Antrieb und der WLAN-Technologie auszurüsten und die Fernbedienung mit einer Eingabemöglichkeit, sowie einem netzwerkanschluss - hier muss nicht unbedingt WLAN her - ein normales LAN-Kabel reicht aus. Um dem Benutzer eine Rückmeldung über den aktuellen Status zu geben, wurden zwei verschiedene Szenarien erdacht: per 16x2 LCD-Display, oder aber über ein Webinterface.

Da für die Fernbedienung bereits eine Netzwerkanbindung vorhanden sein würde, entschied ich mich schließlich für letztere Variante mit dem Webinterface.

2.2 zu entwickelnde Software

Die Software für die beiden Geräte wird grundverschieden sein. Die Fernbedienung wird dabei als Server für sowohl das Webinterface, als auch den Roboter dienen, während der Roboter ein Client ist, der sich bei der Fernbedienung anmeldet

2.3 Kommunikationsprotokoll

Zum Einsatz kommt hier ein simples Adressierungsprotokoll - ein Paket besteht dabei immer aus 2 Byte: 1 Byte Adresse, 1 Byte Wert. Dadurch ist eine sehr schnelle bidirektionale Kommunikation möglich. Wenn beispielsweise die Fernbedienung dem Roboter sagen möchte, er solle Motor 1 mit einer bestimmten Geschwindigkeit drehen, so sähe das Paket wohl so aus, dass zunächst die Motor-Adresse gesendet wird (welche in einer eigenen Header-Datei definiert sein wird) und anschließend den Wert der Geschwindigkeit

3 Umsetzung

3.1 eigene Shields löten und ätzen

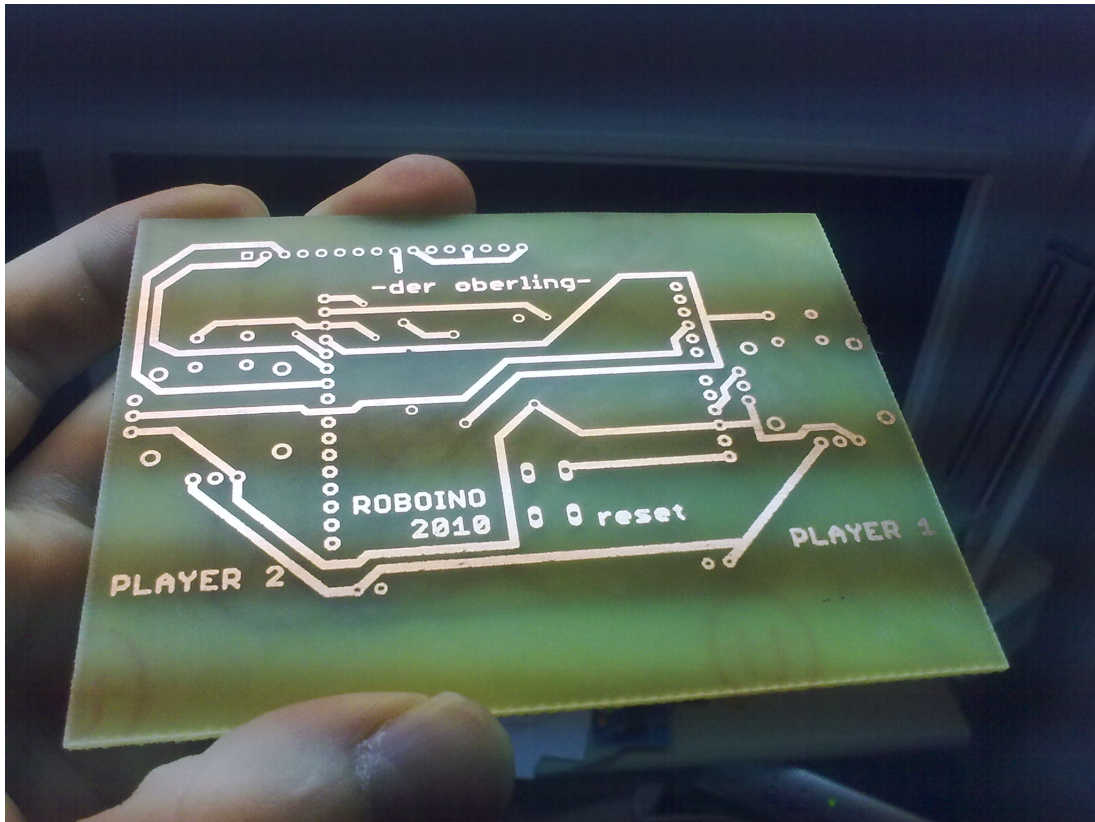


Abbildung 1: geätzte Rohfassung des Controller-Shields

Da bereits Erfahrung mit dem Ätzen und Löten von Platinen gesammelt wurde, lag es nah, auch diese Fertigkeiten in diesem Projekt unter Beweis zu stellen.

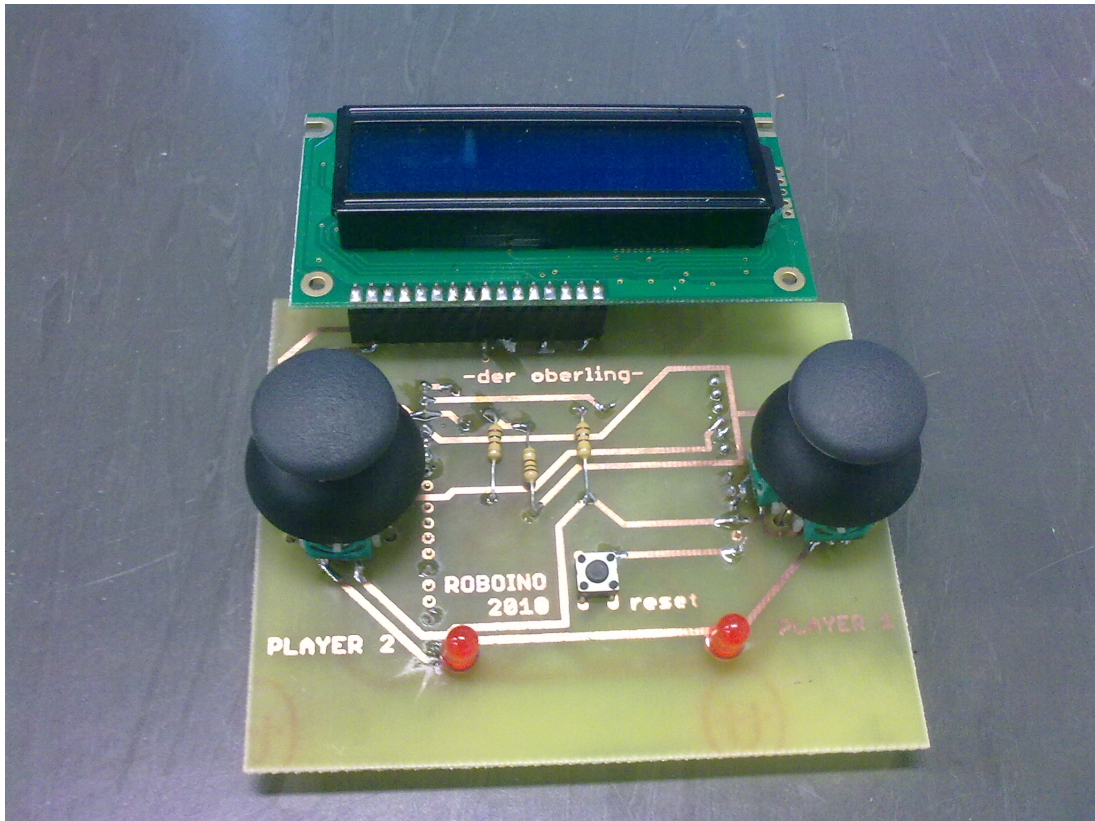


Abbildung 2: endgültige leider fehlerhafte Fassung des Controller-Shields

So habe ich mir für das Projekt ein doppelseitiges Shield für die Fernbedienung geätzt, welches allerdings letztlich doch nicht verwandt wird, da hier ein Layoutfehler vorlag und für eine weitere Variante nicht die Zeit blieb.

3.2 Herausforderungen

3.2.1 WiFly GSX WLAN

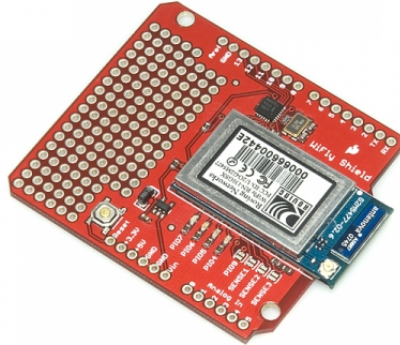


Abbildung 3: WLAN-Shield für Arduino

Mit diesem Modul konnte schon im Rahmen der Bachelorarbeit Erfahrung gesammelt werden und somit hielt sich die Einarbeitungszeit in Grenzen.

3.2.2 Ethernet Library



Abbildung 4: Ethernetshield für Arduino

Diese Library ist offenbar vor allem für Anwendungen mit einmalig verbundenen Clients gedacht - es gibt keine Möglichkeit, sich eine Liste von verbundenen Clients, oder einen bestimmten aus dieser Liste zu holen. Die einzige Möglichkeit ist das direkte Abspeichern verbundener Clients in eigenen Pointer-Variablen zur Laufzeit.

3.2.3 Motoransteuerung

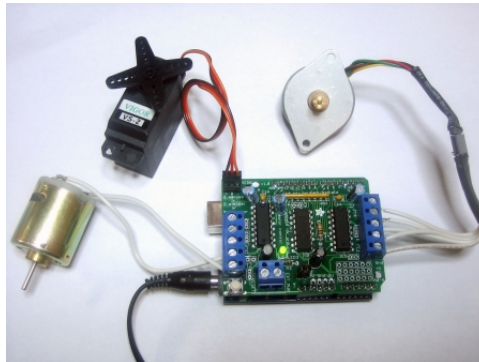


Abbildung 5: Adafruit Motorshield

Obwohl zur Motoransteuerung ein vorgefertigtes Shield (vgl. Abbildung 5 verwendet wurde, bereitete die genügende Energieversorgung der Motoren, sowie die Ansteuerung der Motoren, sodass sie kontinuierlich laufen einige Kopfzerbrechen. Zunächst funktionierte immer nur ein Motor, dann waren die Batterien zu schwach, schließlich musste statt dem einmaligen Befehl loszulaufen in jedem loop jedem der beiden Motoren ein Befehl gegeben werden, sich erneut zu drehen.



Abbildung 6: Dual Getriebe Motorbox

Das verwendete Getriebemotorbox mit zwei DC-Motoren (vgl. Abbildung 6) ließ sich nach Anleitung einfach zusammenbauen. Allerdings bereitete die Tatsache, dass auf unterschiedlichen Webseiten unterschiedliche Spannungspegel aufgeführt waren weitere Probleme bei der Motoransteuerung (letztlich sollen es pro Motor ca. 3 Volt sein).

3.2.4 Roboter Chassis

Da noch keinerlei Erfahrungen im Bereich des Bauens von Modellen gesammelt wurden, musste ich mich hier auch zunächst in sehr viele Dinge einarbeiten. Schließlich wurde das Chassis des Roboters aus einer Aluplatte geformt und verschraubt.

4 geschaffene Lösung

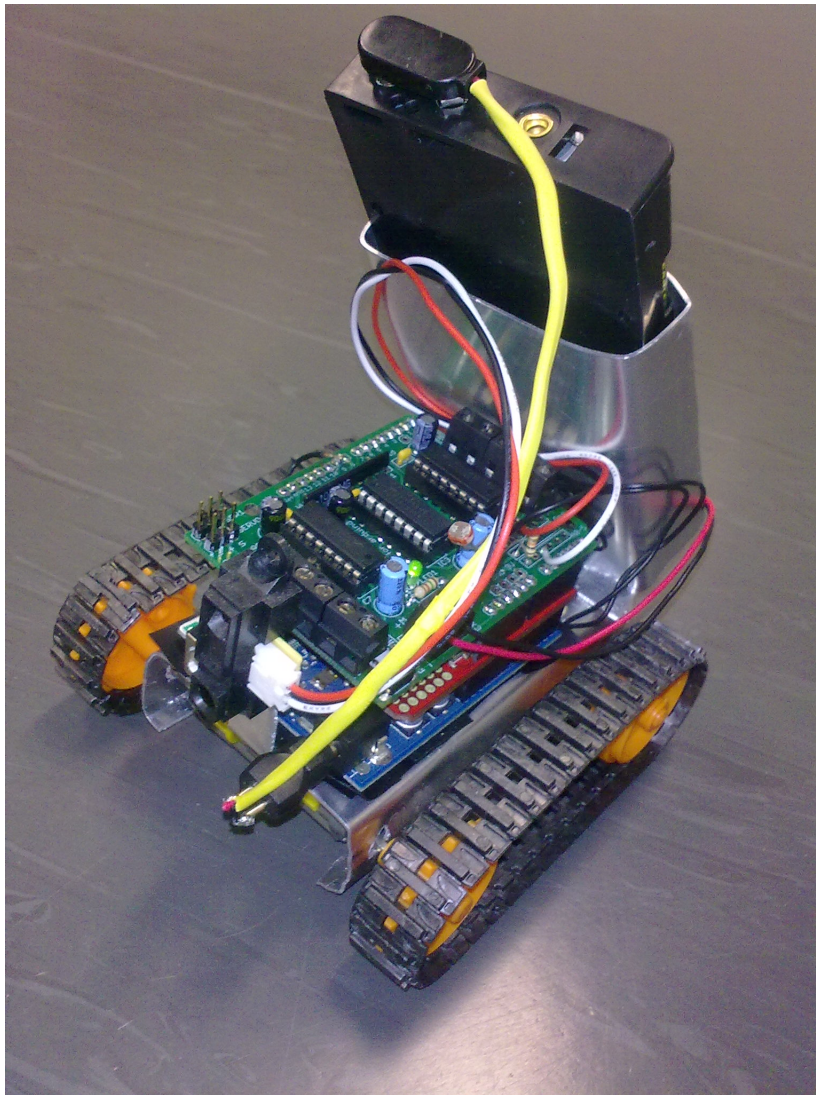


Abbildung 7: WLAN-Roboter

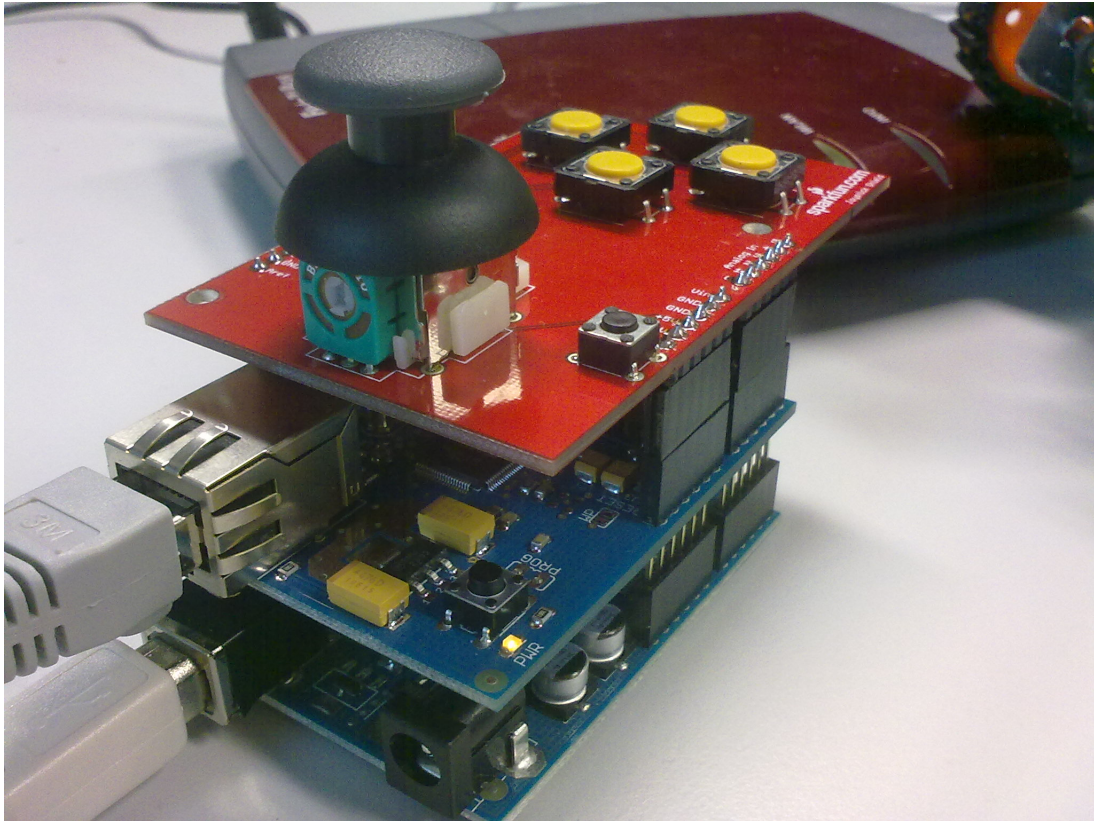


Abbildung 8: Fernbedienung

Die Fernbedienung des Roboters über WLAN funktioniert sehr gut - die hohe Datenübertragungsrate von WLAN im Vergleich zu anderen Funktechnologien ermöglicht eine sehr hoch auflösende Steuerung des Roboters. Über ein Webinterface können die verschiedenen Sensoren des Roboters mit ihren Werten ausgelesen werden.

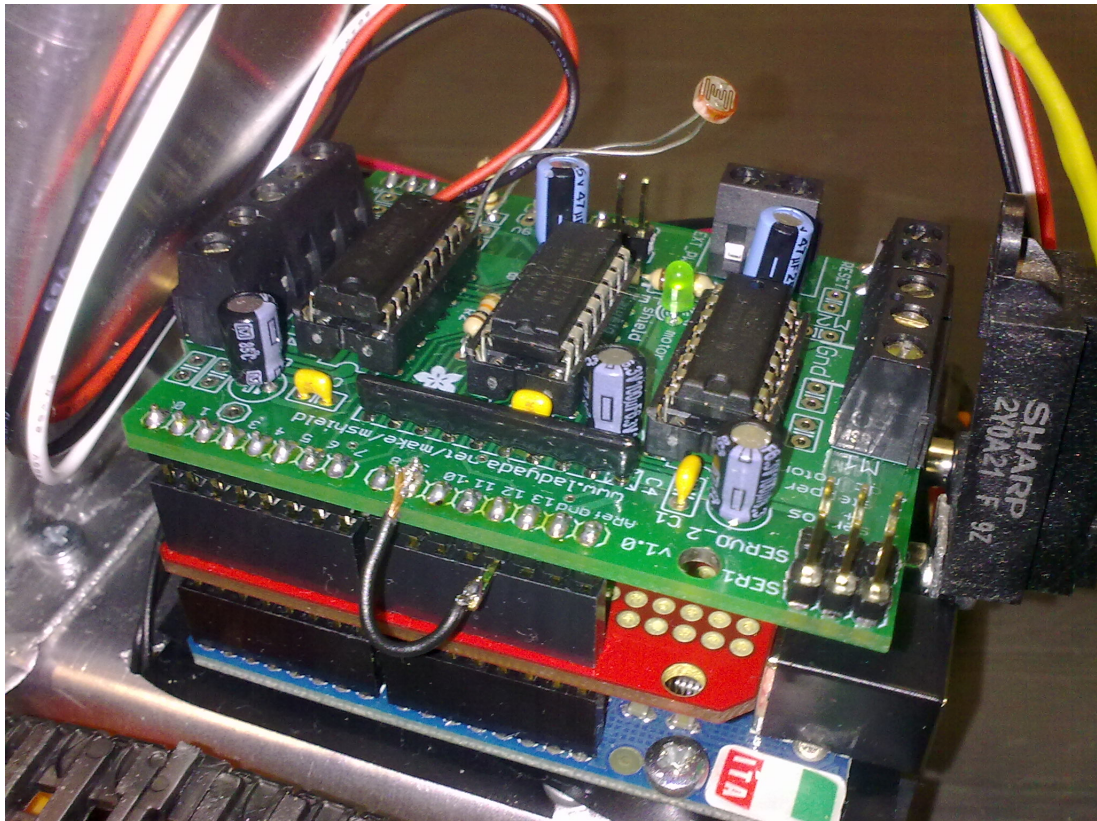


Abbildung 9: Shields und Sensoren auf dem Roboter