

Diplomarbeit: netcon

Lipp, Pietryka

Zusammenfassung

Abstract

Danksagung

Vorwort

Sie wollen Umweltgrößen an mehreren Standorten (aus der Ferne) überwachen und einfache Anlagen steuern? An den Standorten ist lediglich ein gemeinsames Ethernet-Netzwerk verfügbar und für den Aufbau eines eigenen Netzes fehlt das Budget?

Im Rahmen dieser Diplomarbeit, wurde mit **netcon** ein quelloffenes, flexibles Mess- und Steuersystem auf Ethernet-Basis geschaffen. Dabei wurde darauf Rücksicht genommen, erfahrene Endanwender, Unternehmen und Entwickler gleichermaßen zu bedienen. Je nach Anwendungsfall und Vorkenntnissen sollten Sie in der Lage sein ihr eigenes Mess- und Steuersystem aufzubauen.

Im ersten Kapitel folgt ein Überblick über die **allgemeine Konzeption** - Systemvoraussetzungen, Aufbau, Schnittstellen und grundsätzliche Funktionsweise. Ein zweites Kapitel gibt eine Einführung in **grundlegende Begriffe**, die für ein tieferes Verständnis der Entwicklungen erforderlich sind. Danach folgt das große Kapitel, **Hardware**, das den Aufbau und die Funktionsweise der Mess- und Steuermodule behandelt, sowie in die genaue Verwendung der Schnittstellen und Protokolle auf der Hardwareseite einführt. Das letzte Kapitel, **Software** beschreibt die Verwaltungsschicht und dessen Interfaces für die Anzeige und Steuerung der Module.

Inhaltsverzeichnis

1	Überblick [Lipp]	7
1.1	Zielgruppen	7
1.2	Anwendungsbeispiele	8
1.3	Systemvoraussetzungen	9
1.4	Systemaufbau	10
1.4.1	Module	10
1.4.2	Software	12
1.5	Funktionsweise	14
2	Grundlagen [Pietryka]	15
3	Hardware [Pietryka]	16
3.1	Auswahl des Ethernet Controllers	16
3.2	ENC28J60 Beschaltung	16
3.3	ENC28J60 Treibersoftware	17
4	Software [Lipp]	17
4.1	Aufgaben	17
4.2	Java	17
4.3	Aufbau und Funktionsweise	18
4.4	Installation	18
4.4.1	JRE	18
4.4.2	Webserver	18

1 Überblick [Lipp]

Netcon ist zum einen ein Mess- und Steuersystem zur Einbindung in ein bestehendes Ethernet-Netzwerk, zum anderen aber auch das Ziel flexible Werkzeuge für die Erstellung eines solchen Systems bereitzustellen. Dabei stehen alle Entwicklungen unter OpenSource.

Da das System als Übertragungsmedium die Ethernet-Schnittstelle mit der TCP/IP-Protokollschicht verwendet, ist Echtzeitverhalten nicht garantiert. Dadurch ist es nur für unzeitkritische Aufgaben geeignet.

1.1 Zielgruppen

Erfahrene Endanwender, genauso Entwickler sollten mit netcon in der Lage sein, ein Mess- und Steuersystem zu realisieren. Es wurden hardware- und softwareseitig einfache Schnittstellen geschaffen um je nach Wunsch und vorherrschenden Kenntnissen eigene Anwendungen zu erstellen.

Der Anwender kann sich entscheiden, entweder entwickelt er auf Basis der Spezifikationen die netzwerkfähigen Module selbst, oder aber er verwendet die im Rahmen dieser Diplomarbeit gewählten Mikrocontroller-Systeme. Dazu stellt netcon die entwickelte Firmware zur Verfügung. Weiters besteht für netzwerktechnisch unerfahrene Entwickler die Möglichkeit, ihre Module netzwerkfähig zu machen.

Auch auf der Softwareseite stehen mehrere Wege offen. Entwickler können eigene Applikationen über die Schnittstelle der Verwaltungsschicht aufsetzen, oder aber auch die entwickelte Weboberfläche zur Anzeige und Steuerung der Module verwenden.

1.2 Anwendungsbeispiele

Netcon sieht in seiner Spezifikation mehrere Typen von Mess- und Steuermodulen vor. Folgende Liste zeigt Anwendungen, die unter anderem mit diesem System verwirklicht werden können:

- Spannungsmessung
- Temperaturmessung
- Zeitmessung
- Ein/Aus-Schaltung

1.3 Systemvoraussetzungen

Das netcon Mess- und Steuersystem wurde für den Einsatz in einem Ethernet-Netzwerk konzipiert. Dieses muss zumindest über folgende Komponenten verfügen:

- Anschlussmöglichkeiten für die Module (Router/Switch/WLAN)
- DHCP-Server für die IP-Adressvergabe
- **Server** - Javafähige Betriebsumgebung für die Verwaltungsschnittstelle
z.B. PC, Embedded System
- **Client** - Steuer- und Anzeigegerät z.B. Smartphone, Computer

Zusätzlich wird gegebenenfalls ein PHP-fähiger Webserver benötigt, um die bereits entwickelte Website verwenden zu können. Die genauen Anforderungen an die Softwareumgebung, sowie die Einrichtung einer Java Runtime Environment (JRE) und eines Webserver sind im Kapitel Software nachzulesen.

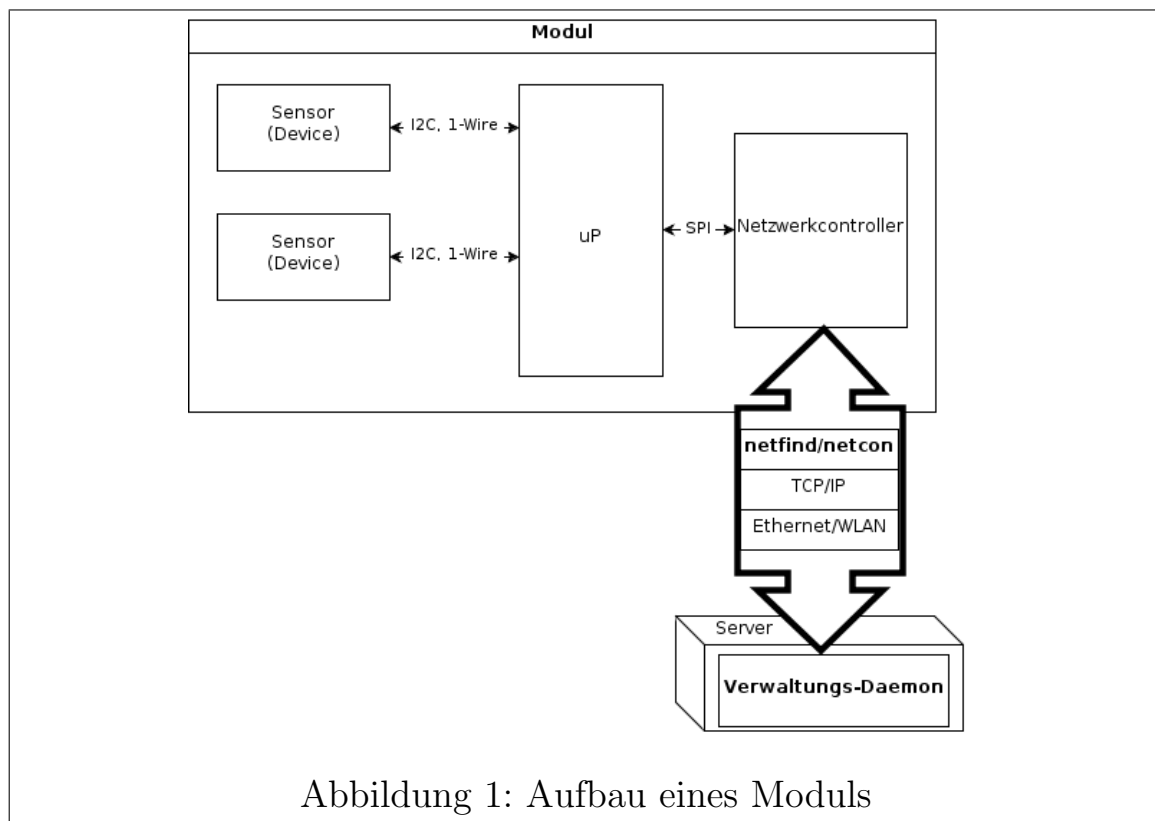
Und nicht zu vergessen sind die wichtigsten Komponenten, die netzwerkfähigen Mess- und Steuermodule. Diese können, wie bereits erwähnt, nach den netcon-Protokollen selbst entwickelt, oder aber auch nach Anleitung erstellt werden. Dazu mehr im nächsten Abschnitt.

1.4 Systemaufbau

Im folgenden sind die zwei grundlegenden netcon-Komponenten inkl. ihrer Schnittstellen beschrieben. Je nachdem wie netcon genutzt werden soll, wird auf weitere Kapitel verwiesen.

1.4.1 Module

Die **Module** sind Hardware, die über Ethernet und TCP/IP erreichbar, abfragbar und bedienbar sind.

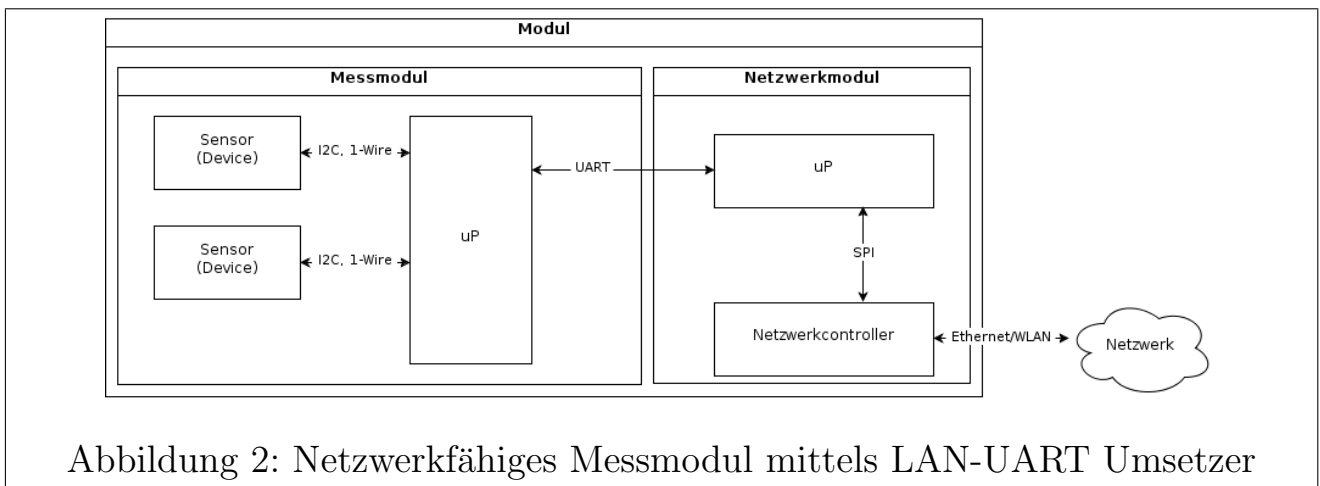


Sie vereinen alle benötigten Komponenten - Mess- und Netzwerkeinheit - auf einer Platine (siehe Abb. 1). Hier erfolgt die Übertragung zwischen den Sensoren und dem Mikroprozessor (uP) meist über Schnittstellen, wie I2C oder 1-Wire, während der Netzwerkcontroller per SPI mit dem uP kommuniziert. Über TCP und mit den beiden Protokollen *netfind* und *netcon* erfolgt die Abfrage und Steuerung durch den plattformunabhängigen Verwaltungs-Deamon *netcond*. Näheres dazu im nächsten Abschnitt.

Messmodule umfassen beispielsweise Sensoren für Temperatur, Luftdruck, oder aber auch Aktoren, wie z.B. Relais oder Displays. Jeder dieser Sensoren bzw. Aktoren wird von netcon als **Device** bezeichnet und kann mit seiner ID abgefragt/gesteuert werden.

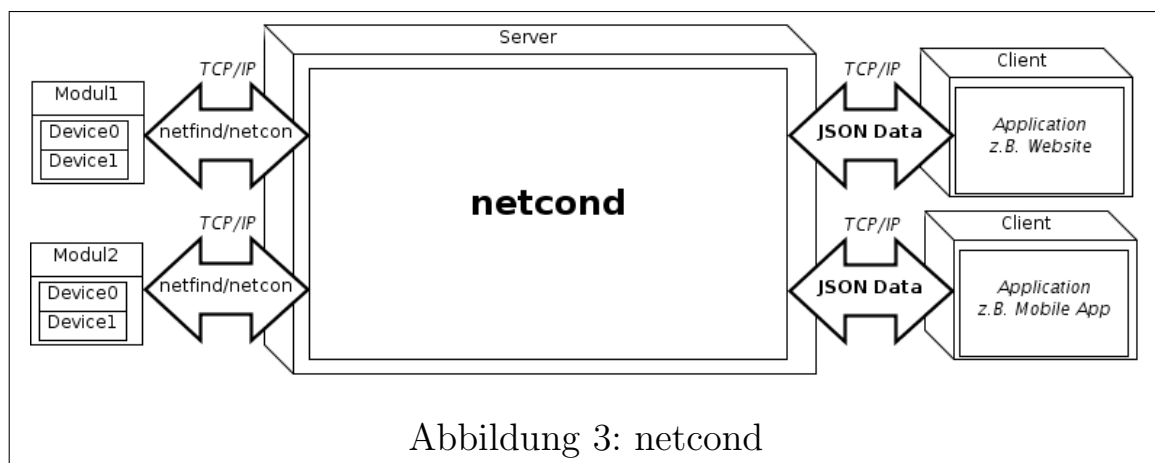
Um den Modulen automatisch eine IP-Adresse zuweisen zu können und damit Konfigurationsarbeit zu ersparen, sollten diese auch das DHCP-Protokoll unterstützen.

Es bestehen grundsätzlich drei Möglichkeiten zur Erstellung von Modulen. Wenn Sie alle erforderlichen Kenntnisse besitzen, um die gesamte Entwicklung selbst zu übernehmen, informieren Sie sich im Kapitel Hardware über den Aufbau der netcon-Protokolle. Sind Sie in der Lage einfache Mess-/Steuermodule ohne Netzwerkfähigkeit zu erstellen, verbinden Sie doch ein zusätzliches Netzwerkmodul (siehe Abb. 2). Diese Möglichkeit erfordert lediglich die Implementierung der Seriellen Schnittstelle (UART). Genauso können die im Rahmen dieser Diplomarbeit konzipierten Module mit ihrer Firmware für die Erstellung eigener Module herangezogen werden. Egal welche Wahl Sie treffen, das Kapitel Hardware unterstützt Sie in allen drei Fällen.

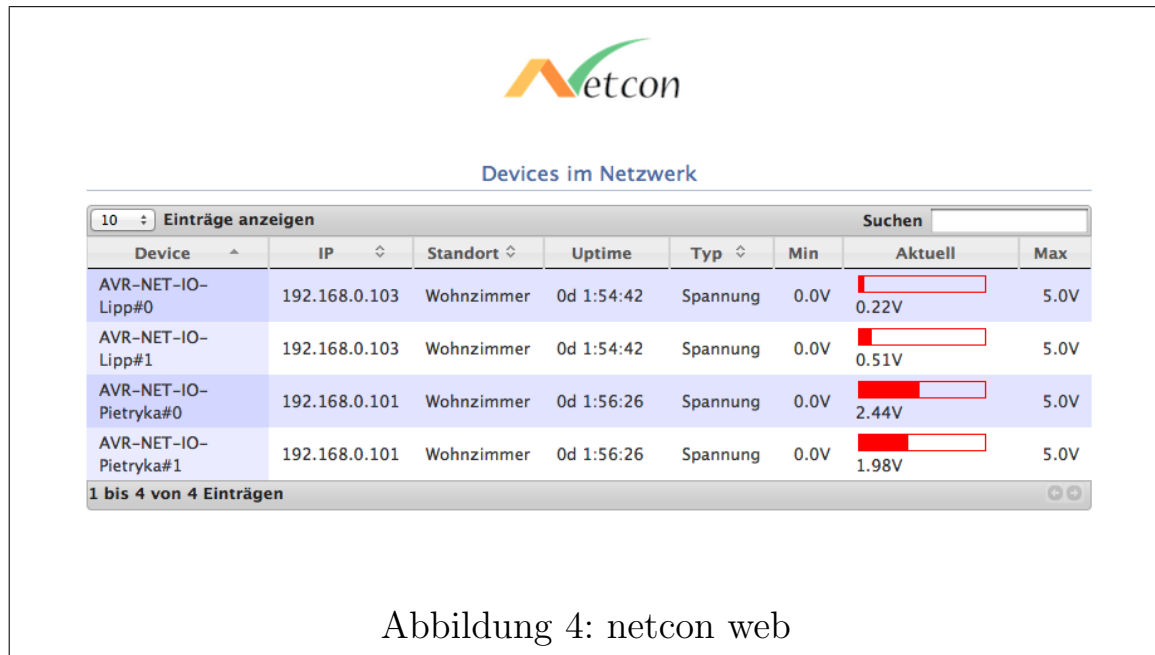


1.4.2 Software

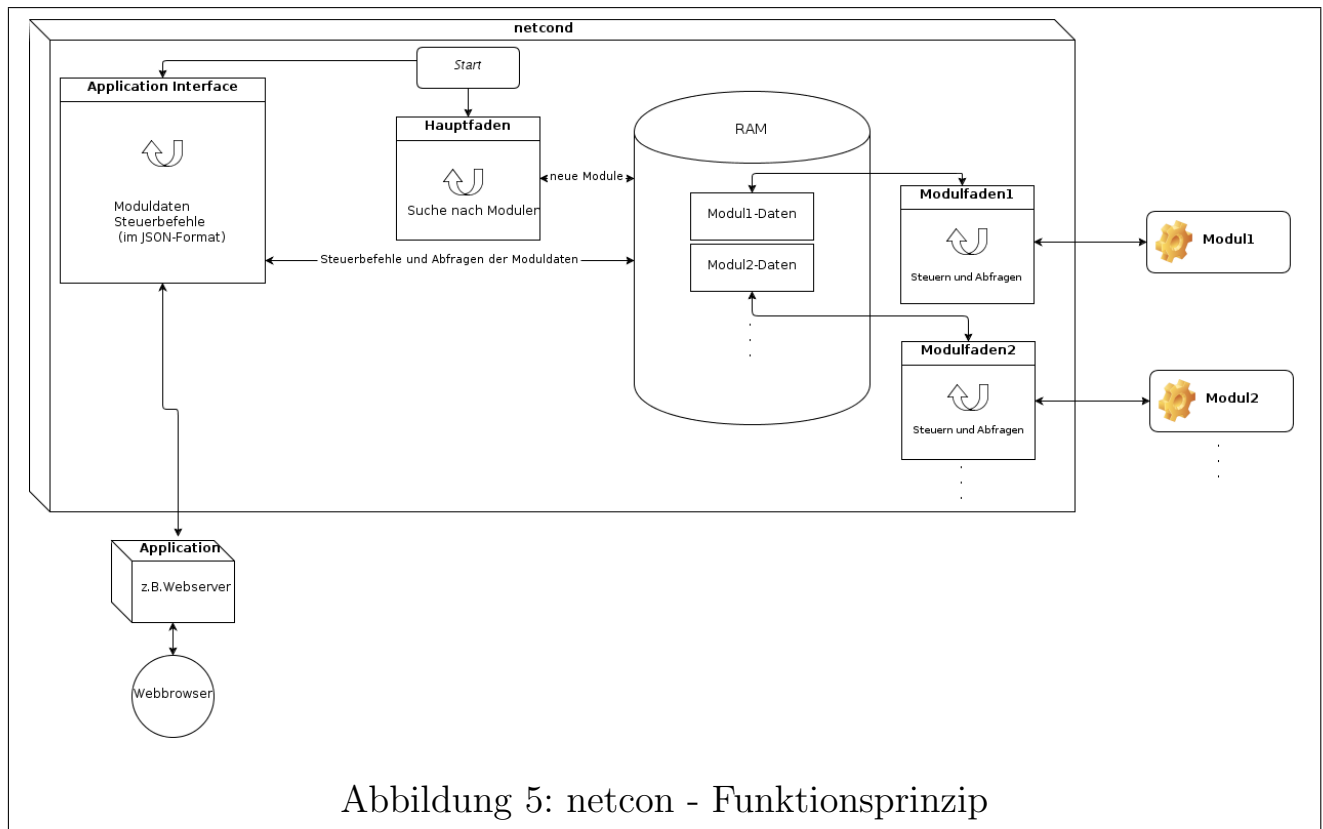
Die Verwaltungsschnittstelle **netcond** ist eine in Java geschriebene Hintergrundanwendung (Daemon), die sich um die Verwaltung der Module kümmert. Wie in Abb. 3 erkennbar können über TCP die Moduldaten abgefragt und Steuernachrichten gesendet werden. Der Datenaustausch erfolgt im *JSON-Format*. Die Anwendung kann beispielsweise eine Website auf einem Webserver oder ein Smartphone-App sein.



Soll die Anwendung zur Anzeige und Steuerung der Module selbst entwickelt werden, führt das Kapitel Software in die Verwendung der Softwareschnittstelle ein. Sonst kann die bereits entwickelte Website **netcon web** (siehe Abb. 4) verwendet werden. Dazu ist zusätzlich zur JRE ein http-Webserver mit PHP-Unterstützung erforderlich. Deren Installation und Konfiguration ist im Kapitel Software erklärt.



1.5 Funktionsweise



Der Java-Daemon netcond sucht nach dem Start alle paar Sekunden das Netzwerk nach verbundenen Modulen ab und hält diese in einer Liste. Für jedes dieser Module wird ein neuer Programmfaden erzeugt, der ständig Messdaten abfragt und diese speichert, sowie die Steuerung der Module auf Anfrage übernimmt. Zusätzlich startet der Daemon einen weiteren Subprozess, die Schnittstelle, über die eine Anwendung - in diesem Fall ein Webserver - die Daten abfragen und Steuerinformationen übermitteln kann. Verbindet sich ein Webbrowser zu diesem Webserver, weißt ein PHP-Script den Daemon an, die aktuellen Daten zu übermitteln. Diese werden dann auf der Website angezeigt. Genau umgekehrt können auch Steuerdaten übermittelt werden z.B. bei Drücken eines Buttons. Dieses Prinzip ist noch einmal in Abb. 5 grafisch verdeutlicht.

2 Grundlagen [Pietryka]

3 Hardware [Pietryka]

3.1 Auswahl des Ethernet Controllers

Damit ein Mikrocontroller über das Ethernet kommunizieren kann, wird eine entsprechende Hardware benötigt, der sogenannte Ethernet Controller. Ein Ethernet Controller übernimmt dabei die Aufgaben der OSI-Schichten 1(Physical) und 2(Data-Link). Der Controller benötigt zudem einen entsprechend großen Empfangspuffer, um mindestens einen vollwertigen Ethernet-Frame(1542 Byte) aufzunehmen zu können. Dabei standen für 8-Bit Mikrocontroller vorerst zwei verschiedene Bausteine zur Auswahl, einmal der CP2200 von SiLabs, und einmal der ENC28J60 von Microchip. Beide Controller haben, was die Netzkommunikation angeht, so ziemlich die selben Features, der gravierende Unterschied liegt jedoch in der Ansteuerung dieser. Der CP2200 wurde von SiLabs, wie es scheint, nur für die Verwendung mit einem Mikrocontroller vom Typ 8051 entwickelt, die Ansteuerung erfolgt deshalb über einen parallelen Adress-/Datenbus wodurch man mindestens 16 Leitungen und Pins am Mikrocontroller benötigt. Beim ENC28J60 erfolgt die Kommunikation über den SPI-Bus, daher benötigt man nur vier Leitungen(MOSI, MISO, SCK, CS), dadurch hat auch der Netzwerkcontroller selber nur 28 Pins und ist auch im "bastlerfreundlichen" DIP-Gehäuse zu bekommen. Ein anderer Faktor für die Auswahl des ENC28J60 war das Vorhandensein einer günstigen Entwicklungsplatine, es gibt bei Pollin den AVR-NET-IO Bausatz, dieser kostet nur 20 € und enthält alle für die Netzwerkprogrammierung benötigten Komponenten(ATmega32, ENC28J60, RJ-45 Buchse).

3.2 ENC28J60 Beschaltung

Die Aussenbeschaltung benötigt neben einigen Standardbauelementen auch einige 1% Widerstände und einen 1:1 Übertrager, jedoch gib es RJ-45 Buchsen in denen bereits der Übertrager, sowie die LEDs bereits eingebaut sind.

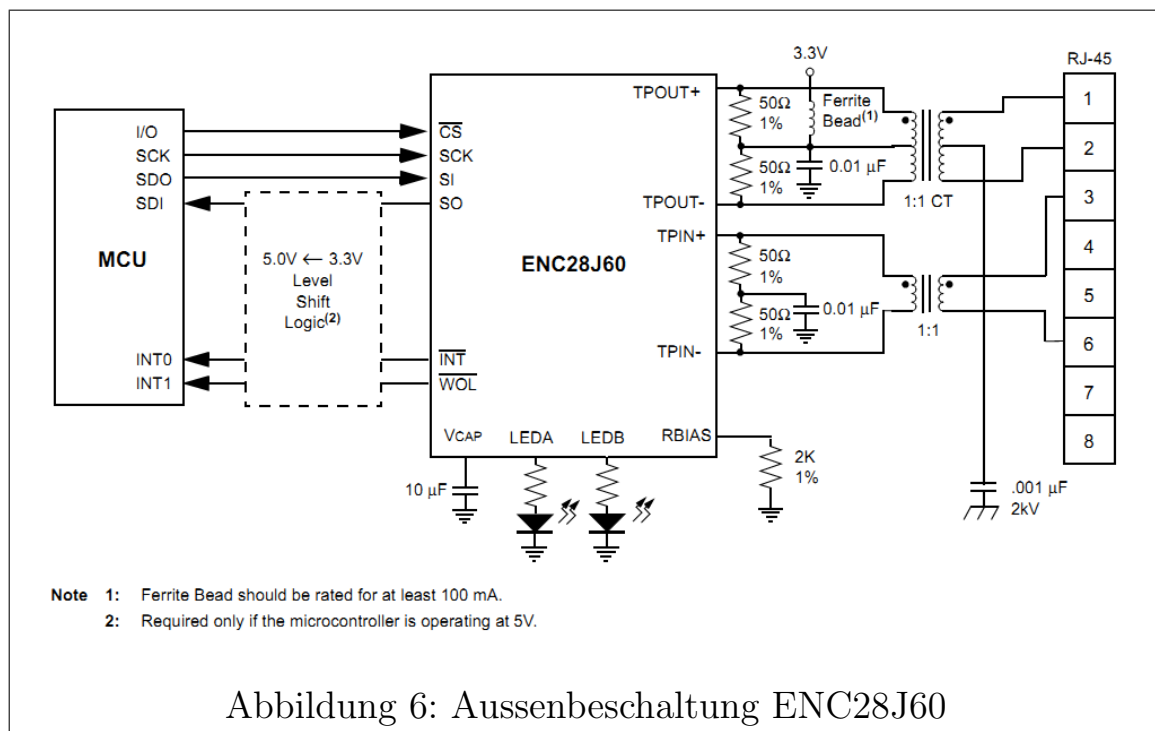


Abbildung 6: Aussenbeschaltung ENC28J60

3.3 ENC28J60 Treibersoftware

4 Software [Lipp]

4.1 Aufgaben

was sind die grundsätzlichen aufgaben der Software, wofür ist sie da?

4.2 Java

warum java?

4.3 Aufbau und Funktionsweise

Aufbau der Software Programmabläufe (Flussdiagramme) Genaue Beschreibung der Schnittstellen

4.4 Installation

4.4.1 JRE

4.4.2 Webserver