Technologisches Gewerbemuseum

Systemtechnik-Maturaausarbeitung

Industrielle Informationstechnologie

Autor:
Burkhard Hampl

1. Juni 2016

Inhaltsverzeichnis

1 Abstract

Internet of Things ist ein neuer Trend, der immer mehr in unserem Alltag integriert wird. Der Kühlschrank, der selbst fehlendes Essen nachbestellt, ist nicht mehr ein Wunschdenken. Das selbst steuernde Licht, Klimaanlage oder Rollläden des ganzen Hauses ist schon Wirklichkeit und das all das und noch viel mehr mit einer einfachen Touch-Geste von überall, nicht mal von zuhause, mit dem Smartphone, gesteuert werden kann, ist auch schon möglich. Dazu benötigt es aber die Technologien, damit Sensorwerte übertragen, ausgewertet und dann die Geräte mit den gewonnen Daten gesteuert werden können. Die Kommunikation der einzelnen Komponenten ist auch ein sehr wichtiger Faktor, damit solche Steuerungen auch funktionieren können, dabei dürfen Themen wie Sicherheit und Privatsphäre aber nicht vergessen werden.

2 Cloud Computing und Internet of Things

Das Internet der Dinge ist heutzutage in aller Munde. Es kommen stetig neue Geräte auf den Markt und es gibt andauernd neue Entwicklungen.

2.1 Grundlagen Internet of Things

Das Internet der Dinge (auf englisch Internet of Things (kurz IoT)) ist der Ansatz, alle (technischen) Geräte miteinander zu verbinden bzw. vernetzen. Vor allem das jedes "Ding" seinen eigenen Internetzugriff hat spielt eine große Rolle. Damit können Informationen aus der Ferne abgerufen und analysiert werden und es ist möglich über diese Geräte auch Dinge fern zu steuern. Beispiele für vernetzte "Dinge" sind Maschinen, Fahrzeuge, Haushaltstechnik, ergo Beleuchtung, Rollläden, Fernseher, PCs, Kühlschränke, Warenverpackungen, alle möglichen Arten von Sensoren, Kameras, Wecker, Armbanduhren. Das "konventionelle" (alte) Internet mit "nur" Server und PCs, wird oft als "Internet der Computer" (auf englisch Internet of Computers) bezeichnet, dabei befinden sich beide im gleichen Netzwert (dem Internet). [1][2]

2.1.1 IoT aus technischer Sicht

Es gibt ein paar Funktionen, welche aus den Technikentwicklungen einstanden sind:

- **Kommunikation und Kooperation:** Es gibt die Möglichkeit für die Objekte Daten mit dem Internet oder untereinander auszutauschen bzw. gemeinsam auf Ressourcen/Dienste zuzugreifen. Es gibt zahlreiche richtfunkbasierte Technologien wie "Global System for Mobile Communications" (kurz GSM) oder "Universal Mobile Telecommunications System" (kurz UMTS), "Radio-frequency identification" (kurz RFID), Bluetooth, "Near Field Communication" (kurz NFC), ZigBee und "Wireless Local Area Network" (kurz WLAN) oder Wi-Fi (die Zertifizierung von WLAN). Es gibt auch diverse Weiterentwicklungen der einzelnen Technologien, vor allem im Bereich der Wireless Personal Area Networks (WPAN).

- **Adressierbarkeit:** Um ein Objekt zu finden und ansprechen zu können, benötigt es Discovery-, Lookup- oder Namensdienste.

- **Identifikation:** Jedes "Ding", egal ob passiv oder aktiv, braucht eine eindeutige Identifikation, mittels der die Objekte voneinander unterschieden werden können. Dafür hat jede Schnittstelle eine eindeutige Nummer. Auch Strichcodes oder RFID-Tags, welche alle Passive sind, haben eine derartige eindeutige Nummer. Passive Technologien brauchen einen sogenannte "Mediator", damit sie ausgelesen werden können. Dieser ist mit dem Netzwerk verbunden und gibt die Informationen weiter.

- **Sensorik:** Die Dinge bzw. Objekte messen ihre Umgebung und entweder geben die gesammelte Informationen weiter oder reagieren auf diese.

- **Effektorik:** Mit Effektoren haben die Objekte die Möglichkeit auf ihre Umwelt und Umgebung Einfluss zu nehmen, dadurch ist es möglich über die Ferne Dinge zu steuern.

- **Eingebettete Informationsverarbeitung:** Die Objekte können auch "Smart" sein, d.h. sie haben eine Prozessor oder Mikrocontroller und Speicherplatz eingebaut und können so Informationen verarbeiten und interpretieren.

- **Lokalisierung:** Die Objekte wissen wo ihr physischer Standort ist oder Objekte können sie finden bzw. lokalisieren. Auch hier gibt es mehrere Technologie die eingesetzt werden können z.B. "Global Positioning System" (kurz. GPS), das Mobilfunknetz, Ultraschallzeitmessungen, "Ultra-Wide Band" (kurz UWB), Funkbaken (z.B. benachbarte WLAN-Basisstationen oder RFID-Lesegeräte mit bekannten Koordinaten) und optische Technologien.

- **Benutzungsschnittstelle:** Die smarten Objekte haben die Möglichkeit mit Menschen, auf direkter oder indirekt, etwa via Smartphone, kommunizieren. Hierfür gibt es auch innovative Interaktionsparadigmen wie z.B. "tangible user interfaces" oder biegbare Displays, aber auch Sprach-, Bild- und Gestenerkennung.

Oft werden aber für konkrete Anwendungszwecke nur Teile dieser Funktionen benötigt, da manche dieser Funktionen auch z.B. einen relativ großen Energieverbrauch haben oder sie einfach nicht benötigt werden. [1][2]

2.1.2 Technische Herausforderungen

Auch wenn es so viele Einsatzzwecke und Szenarien gibt, es gibt ein paar Dinge, die zu Beachten oder noch (technische) Herausforderungen darstellen:

- **Skalierbarkeit:** Es gibt eine viel größere Gesamtmenge an IoT-Geräten, als es klassische Geräte im Internet der Computer gibt. Die Herausforderung ist für sowohl kleine als auch im globalen Umfeld eine effiziente Lösung für Kommunikation und Service bereit zu stellen.

- **"Arrive and operate":** Es soll für den Nutzer kein Masterstudium benötigen, die verschiedenen Alltagsgegenstände zu Konfigurieren und für seine Situation anzupassen. Hier ist eher die Devise "Plug & Play" zur Anwendung zu bringen.

- **Interoperabilität:** Verschiedenste "Dinge" von verschiedensten Herstellern, müssen miteinander kommunizieren und interagieren können. Um eine möglichst heterogene Umgebung zu schaffen, benötigt es Prinzipien und Standards, welche umgesetzt werden müssen. Vor allem die Adressierung der Objekte ist ein wichtiges Thema.

- **Discovery:** Die Umgebungen der Dienste und "Dinge" verändert sich ständig. Es müssen vor allem alle Gräte eine gleiche Beschreibung der Funktionalitäten bereitstellen, damit sie eingeordnet und richtig verwendet werden können.

- **Softwarekomplexität:** Zum einem gibt es die eingebetteten System, welche nur begrenzte Ressourcen zur Verfügung haben und zum anderen gibt es die großen Dienst im Internet, welche möglichst viele Leistungen zur Verfügung stellen sollen.

- **Datenvolumen:** Bestimmte Einsatzzwecke verursachen recht große Menge an Daten, bei anderen gibt es fast keine Kommunikation.

- **Intelligente Dateninterpretation:** Das Interpretieren der Daten, welche von "Dingen" gemessen werde, sollten für den lokalen Kontext angepasst werden, sodass sie für den Nutzer abgestimmt sind. Aus diesen angepassten Daten dann globale Schlüsse zu ziehen, ist recht komplex.

- **Sicherheit und Privatsphärenschutz:** Die Vernetzung von so vielen Objekten, ruft gewisse Bedenken bezüglich deren Sicherheit und Privatsphäre hervor. Es müssen mindestens die gängigen Sicherheits- und Schutzaspekten, wie z.B. Verschlüsselung und Authentisierung, umgesetzt werden. Die Möglichkeit den Zugriff auf gewisse Dienst einzuschenken sollte auch umgesetzt werden.
- **Fehlertoleranz:** Durch die raschen Veränderungen in der Welt der Dinge, benötigt es Redundanz auf vielen Ebenen und automatische Anpassung an die veränderten Bedingungen.
- **Energieversorgung:** Dadurch das es derzeit nur einen langsamen Fortschritt im Bereich der Energiespeicherung (Batterietechnik) gibt, aber die Geräte energieautark handeln müssen, muss mit Low-Power-Prozessoren und energiesparenden Kommunikationsmöglichkeiten nachgeholfen werden.
- **Interaktion und Nahbereichskommunikation:** Oft wird nur auf kurze Strecken von wenigen Zentimetern kommuniziert, dafür benötigt es nur wenig Energie, die Adressierung ist recht einfach und es kann schwer mitgehört werden. Zwei gute Beispiele für so eine Kommunikation ist NFC und RFID.
- **Funkbasierte Kommunikation:** Gängige Funktechnologien wie GSM, UMTS, WLAN oder Bluethooth, sind recht Energieintensiv, vor allem bei schlechten Empfang. Deswegen gibt es neu Entwicklungen wie ZigBee, welche deutlich Energiesparender sind.

[1][2]

2.2 Projektumfeld

Projektumfeld: “Loxone”
Die Themen Smart-Home und Internet of Things sind sehr miteinander verbunden. Die Überwachung der mobilen Robotik kann sehr einfach mit smarten Sensoren umgesetzt werden. Es empfehlt sich, auf den Energieverbrauch der Geräte zu achten, der Einsatz von sparsamen Funktechnologien und Prozessoren ist recht wichtig.

3 Automatisierung, Regelung und Steuerung

In der Praxis werden oft verschiedene Arten der Kommunikation eingesetzt. Bus-Technologien erfreuen sich über eine große Beliebtheit, vor allem weil sie sehr einfach erweiterbar sind und nicht viel Verkabelung benötigen und deswegen kostengünstig sind. Manchmal möchte man aber "nur" Motoren oder dergleichen ansteuern, dafür benötigt man Methoden wie PWM.

3.1 Controller Area Network

Das Controller Area Network (kurz CAN, oft CAN-Bus) ist ein Bus-System, welches gleichberechtigte Komponenten (Knoten, Node) über zwei Drähte mit einander verbindet. Bosch entwickelte das CAN-Protokoll 1983 als Bus in Kraftfahrzeugen. Dadurch das CAN eine hohe Störsicherheit, geringe Kosten und Echtzeitfähigkeit hat, wird es auch in Textilmaschinen, Aufzugsteuerungen und Landmaschinen eingesetzt. Um die Weiterentwicklung des CAN-Protokoll kümmert sich "Can in Automation" (kurz CiA). Dadurch, dass ein Bit auf zwei Leitungen gleichzeitig mit einer gegensinnigen Potezialänderung (differentiellen Signal) gesendet wird, wird die elektrische Störsicherheit erreicht. Für die Steckverbindungen hat sich ein 9 poliger Sub-D Stecker durchgesetzt. Für die Übertragung wird mindestens ein 3 poliges Kable benötigt, an welches "CAN-High", "CAN-Low" – invertiertes und nicht invertiertes Signal – und "Ground" gelegt wird. CAN kann bis zu einer Übertragungsgeschwindigkeit von 1 Mbit/s übertragen, denn alle CAN-Knoten müssen die Nachrichten gleichzeitig verarbeiten können. [3]

3.1.1 Prinzip des Datenaustausches im Controller Area Network

Die Datenübertragung von CAN erfolgt in Form von sogenannte Botschaftsrahmen, auch Frames genannt. Beim Übertragen von Nachrichten (z.B. Drehzahl oder Motortemperatur), wird kein konkreter Knoten des Netzwerks adressiert, sonder die Nachrichten bekommen eine eindeutigen Identifier. Dieser Identifier legt die Inhaltskennzeichnung und Priorität der Nachricht fest. Nach korrektem Empfangen der Nachricht, stellt jeder Station anhand des Identifiers fest, ob die Nachricht für einen relevant ist oder nicht. Mit der inhaltsbezogenen Adressierung ist es möglich, sehr einfach Stationen zum Netzwerk hinzuzufügen. Dadurch ergibt sich auch Multicast, die Nachricht kann gleichzeitig mehrere Stationen betreffen. [3]

Kollisionsprüfung Dadurch, dass der CAN-Bus kein "Master-Slave-Bus" ist, kann jeder Teilnehmer ohne besondere Aufforderung Nachrichten verschicken. Dabei kann es aber zu Kollisionen (wie bei Ethernet (Stichwort CSMA/CD (Carrier Sense Multiple Access/Collision Detection))) kommen, was von der Hardware mittels erneut Sendern gelöst wird. So eine Kollision wird durch vergleichen der gesendeten und empfangenen Nachricht auf der Sender-Seite erkannt. [3]

3.2 I^2C

I^2C (steht für Inter-Integrated Circuit) ist ein serielle Bus, der für die Kommunikation zwischen ICs (Integrated Circuits (auf deutsch Integrierter Schaltkreise)) verwendet wird. Es besteht aus zwei Drähten, einer Daten und Taktleitung. Philips entwickelte den Bus Anfang der 80er Jahre und wird von manchen Herstellern aus Lizenzgründen TWI (Two Wire Interface) genannt. Bei einem I^2C gibt es mindestens einen Master und bis zu 127 Slaves, sollte es mehrere Master geben, dass wird er als "Multi-Master-Bus" bezeichnet. Die maximale Übertragungsrate im "standart mode" ist 100 kbit/s, im "fast mode" 400 bit/s und im "high-speed mode" 3,4 MBit/s. I^2C ist zwar dazu ausgelegt, nur auf der Leiterplatine

Sensoren miteinander zu verbinden, mit einer entsprechend langsamen Geschwindigkeit, können auch mehrere Meter überwunden werden. [4]

3.2.1 Adressierung

Jeder Slave auf einem Bus, muss eine eindeutige individuelle Adresse besitzen. Es gibt auch einen Broadcastkanal, über den alle Salves gleichzeitig angesprochen werden können. Die Kommunikation geht immer von dem/einem Master aus, Slaves können nie selbständig mit der Datenübertagung beginnen. Die Kommunikation startet mit der übernahmen des Busses durch eine Master, der dann die (7-bit bzw. 10-bit) Adresse des Salve ausgibt. Danach teilt der Master dem Salve mit, ob er Daten senden oder empfangen will und dann werden erst die eigentlichen Daten übertragen. Erst nach Abschluss erfolgreicher Übertragung gibt der Master den Bus wieder frei. Sollten mehrere Master vorhanden sein, stellt ein logisches Protokoll, dass es keine Störungen untereinander gibt. Für Broadcast-Operationen gibt es ein "Time-Slot-System", welches sicherstellt, dass immer nur ein Slave gleichzeitig antwortet. [4]

3.3 PWM

Pulsweitenmodulation (auf englisch Pulse Width Modulation (kurz PWM)) ist eine Möglichkeit größere Lasten, wie beispielsweise Motoren, anzusteuern. Dabei werden Impulse zwar mit voller Spannung aber mit variabler Breite an die Last angelegt. Das Rechtecksignal hat eine konstante Frequenz, wird aber mit einem bestimmten Testverhältnis moduliert. Die Charakteristik einer PWM ist also ihre Frequenz und ihre Testverhältnise (auf englisch Duty Cycle). Das bietet den Vorteil, weniger Last zu verbrauchen, da nicht andauernd eine Eingangsspannung anliegt und bei Mikrocontrollern gibt es oft spezielle PWM-Ausgänge. [5][6]

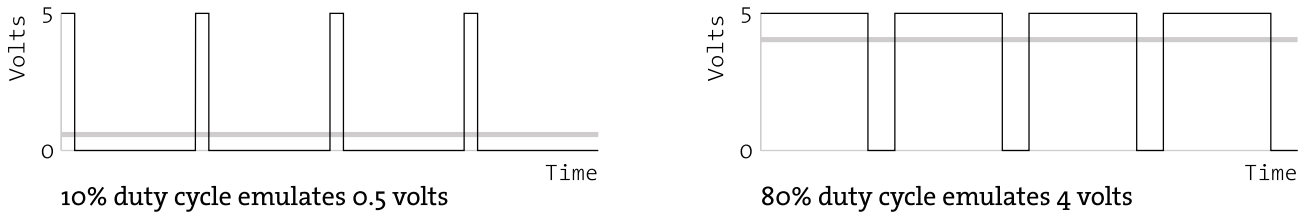

Abbildung 1: Verschiedene Testverhältnise [7]

In der Abbildung 1 sieht man zwei verschiedene Testverhältnise bzw. Pulsweiten, welche in verschiedenen emulierten Volt Anzahlen resultieren. Oft wird eine Pulsweite von 255 Schritten (8 Bit) oder mehr eingesetzt. In der Abbildung kann man auch erkenne, dass die eigentliche Grundfrequenz sich nicht ändert, sonder nur die Ein- und Ausschaltzeiten. Es ist möglich die Modulationsfrequenz zu variieren, in der Praxis findet man oft Frequenzbereich zwischen mehreren hundert Hertz (z.B. Bremslichter bei Autos) bis in den Bereich von 100 kHz (Schaltregler). Außerdem ist es auch möglich mit Pulsweitenmodulation Leuchtdiode (kurz LED (steht für light-emitting diode)) zu dimmen. [5][6]

3.4 Projektumfeld

Projektumfeld: "Loxone"
Auch in Smart-Homes benötigt man eine Art der Kommunikation, dabei kann auf Bus-Technologien

zurückgegriffen werden. Es gibt verschiedene Busse, über die Kommuniziert werden kann, meistens kommt es immer auf den genauen Einsatzzweck drauf an. Auch ist es möglich Autos mit IoT zu verbinden, in dem beispielsweise Sensoren an den internen CAN-Bus geschlossen werden. Für die Steuerung von Leuchtmittel oder Motoren benötigt man z.B. PWM.

4 Security, Safety, Availability

Sicherheit ist überall in der Informatik ein wichtiges Thema, so auch bei Sensor-Netzwerken oder Smart-Homes. Verschlüsselung ist wichtiger den je und sollte (fast) überall eingesetzt werden. Der Algorithmus AES wird sehr weitreichend verwendet und wird derzeit als sicher angesehen. Dabei gibt es ein paar Probleme was Hardwarezugriff und Verschlüsselung und was Schlüsselaustausch angeht.

4.1 Grundlagen von AES

AES (Advanced Encryption Standard) ist ein symmetrischer Block-Chiffre, mit einer Blocklänge von 128 Bit, welcher den vorherigen Standard DES (Data Encryption Standard), der nur einen 56-Bit langen Schlüssel hat, ersetzt und wurde vom National Institute of Standards and Technology (NIST) standardisiert. Ein weiterer Unterschied ist der, dass AES eine flexible Block- und Schlüssellänge besitzt, welche für die Blocklänge standardmäßig 128 Bit und für die Schlüssellänge 128, 192 und 256 Bit beträgt. Die Rundenanzahl hängt von der Schlüssellänge ab und beträgt im Normalfall 10 Runden bei 128 Bit, 12 Runden bei 192 Bit und 14 Runden bei 256 Bit. Eine XOR-Verknüpfung vom Rundenschlüssel und Klartext wird vor der erste Runde durchgeführt. AES hat den Vorteil, sehr schnell und von viel Hardware und Software unterstützt zu sein. Es gibt viele Hersteller, die erweiterte Befehlssatz für AES implementiert haben und die Performance dieser steigern konnten, dadurch muss dann weniger auf der Software-Ebene implementiert werden und die Performance ist höher. [8][9]

4.1.1 Funktionsweise von AES

Bei AES werden die Ergebnisse und Texte als Bytes in den sogenannten States (4x4-Matrizen) gespeichert. Diese Matrizen werden Spaltenweise, von rechts oben nach links unten, gefüllt und die Suche erfolgt mittels der Indizes. Die 128 Bit der Transformationsfunktion werde in 16 Byte geteilt. AES besitzt eine Rundenfunktion, welche aus SubByte, ShiftRow, MaxColumn und AddRoundKey bestehen. [8]

4.2 Probleme von AES Hardware-Implementationen

Ein Problem der Hardware-Implementationen, kann die fehlerhafte oder schlechte Implementation sein. Dadurch kann es vorkommen, dass dies Implementierungen für gewisse Attacken angreifbar sind und so vor allem das ganze System unsicher machen. Denn es können kein sicheren Systeme auf unsichere Hardware oder Softwarekomponenten aufgebaut werden. Auch gibt es Betriebsarten wie z.B. den Electronic Codebook (ECB), welche, wenn verwendet, die Verschlüsselung unsicherer macht. Solche Algorithmen oder Modi werden oft zur “Optimierung” oder Verschnellerung eingesetzt, wobei man da oft Abstriche beim Punkt Sicherheit machen muss.
Oft besteht das Problem, dass ein potentieller Angreifer physischen Zugriff auf die Hardware hat und so können alle möglichen Szenarien durchspielen werden. Ein Beispiel für so eine Attacke wäre “Fault Attacks”. Dabei wird mit Beeinflussung der Hardware-Komponenten die Umgebung verändert und so Fehler in den Berechnungen erzeugen. Dabei gibt es mehrere Möglichkeiten einzuwirken: [9]

Veränderung der Stromversorgung Dabei wird die Spannung des Mikrocontrollers so manipuliert bzw. verändert, dass Fehlinterpretationen oder das überspringen von Anweisungen auftreten kann. [9]

Temperatur Hardware muss innerhalb ihrem Arbeitstemperatur-Bereich operieren, damit die richtige Funktionsweise garantiert werden kann. Wenn diese absichtlich verlassen wird, kann es zu Hardwarefehlern wie beispielsweise beim Memory kommen, sodass nicht mehr von ihm gelesen werden kann. [9]

Weißes licht Photoelektrische Effekte verursachen Empfindlichkeiten im Bezug auf weißes Licht bei Mikrocontrollern. [9]

Laser Mit ähnliche Effekten wie bei weißen Licht können Laser sogar auf bestimmte Bereiche der Chips gerichtet werden. [9]

Durch solche Einwirkungen können die Anzahl an Runden verkürzen oder Daten manipuliert werden. Gegenmaßnahmen für "Fault Attacks" können Sensoren, wie Lichtsensoren, Frequenz-Detektoren oder Stromversorgungs-Detektoren, welche bei Unregelmäßigkeiten anschlagen oder Hardware-Redundanz sein. [9]

4.3 Key Pre-distribution

Damit Blockverschlüsselung wie AES funktionieren kann, benötigt es "pre-distributed" Schlüssel. Schlüssel werden benötigt, damit überhaupt eine Kommunikation zwischen verschiedenen Geräten zustande kommen kann. Es kommt zwar immer auf den Anwendungszweck drauf an, welche Algorithmus zum ausliefern von "pre-shared" Schlüssel verwenden soll, doch gibt es ein paar Dinge, die immer können sollte. So ein Schüssel-Verteil-Schema sollte vor allem die Identität der Nodes prüfen könne, Skalierbar und Flexibel, was neue Nodes hinzufügen anbelangt, sein. Es sollte auch möglich sein, nachdem z.B. ein Node versucht hat das Netzwerk zu infizieren, ausgeschlossen zu werden. [9]

Simple Key Pre-distribution Die einfachst Variante der Schlüssel "pre-distribution", ist das "pre-deployment", dabei wird auf jeden Node $n - 1$ Schlüssel ausgeliefert, welche für die paar weise Verschlüsselung benötigt wird. So eine Schema macht natürlich nur im kleine Rahmen Sinn, denn für jeden neuen Knoten, muss auf jeden anderen Knoten sein Schlüssel ausgeliefert werden und bei vielen Schlüssel, ist der Speicherverbrauch recht hoch. Als Verbesserung dieses Schemas kann ein "Key Distribution Center" (KDC) verwendet werden, über den für jede Session der jeweilige Schlüssel, welcher für die Kommunikation benötigt wird, geholt wird. [9]

Random Key Pre-distribution Eine andere Möglichkeit ist jedem Node nur einen zufälligen Anteil an Schlüssel zu übergeben und so die Schlüsselmenge zu vermindern. Die Wahrscheinlichkeit liegt recht hoch, dass beide einen gleichen Schlüssel haben, wenn nicht müssen sie sich an einen dritt Knoten wenden und sich von dem einen Schlüssel hohlen. [9]

Hierarchical Group Key Management Schemes Des Weiteren besteht die Möglichkeit einer hierarchischen Anordnung. Dabei teilen sich Gruppen den gleichen Schlüssel, welche aus deren Mitgliedern berechnet wird. Die hierarchische Anordnung basiert auf Rechenpower oder anderen Faktoren wie Zugriffsrechten und jeder Zweig hat die Schlüssel seiner Kinder. Nachteil einer solchen Struktur ist immer, dass je weiter die Nodes an der Wurzel dran sind, desto attraktiver sind sie als Angriffsziel. [9]

ECC Based Key Management Aufgrund der Inflexibilität der vorher erwehrten Schemen, wurde ein neue Verfahren namens "Elliptic Curve Cryptography" (ECC) entwickelt, welche auf Problemen von Logarithmen basiert. Dabei kennen die Knoten nur die Schlüssel ihrer Nachbarn (peer-to-peer) und es kann bestimmte Verbindungs-Nodes geben, welche für die Verbindung der Nodes zuständig sind. [9]

4.4 Projektumfeld

Projektumfeld: "UN Bojen-Netz"
Bei einem Sensor-Netzwerk, sollte auch auf Verschlüsselung gesetzt werden, damit vor allem die Integrität gegeben ist. Es soll nicht möglich sein, Bojen abzufangen und so zu manipulieren, das Daten abgefangen und verändert werden können. Sollte es möglich sein Bojen hinzuzufügen, dann kann sich natürlich auch eine "feindliche" Bojen einhängen und Unsinn getrieben werden. Nach der Erkennung soll diese Knoten natürlich nicht mehr kommunizieren können und aus dem Netzwerk entfernt werden.

5 Authentication, Authorization, Accounting

Die Identifikation einer Person ist meistens sehr wichtig. Es soll z.B. nicht möglich sein, dass jemand einfach so Geld von einem fremden Konto abheben kann oder ein Buch ohne ausgeliehen zu werden aus eine Bibliothek mitgenommen werden kann.

5.1 RFID

"Radio-frequency identification" (RFID) ist eine Technologie, ähnlich wie Barcodes, zur Identifikation von Objekten, Personen und Tieren. Diese besteht aus zwei Teilen, einem "Transponder" (auch RFID-Tag oder RFID-Etikette genannt) und einem RFID-Lesegerät (auch Lesegerät oder RFID-Reader genannt). Der Transponder hat eine eindeutige kodierte Nummer, welche durch Radiowellen vom Lesegerät gelesen werden kann. Der Transponder ist am zu identifizierten Objekt befestigt, das Lesegerät selber kann mobil geführt oder fest befestigt sein. Die kodierte Nummer kann sobald gelesen werde, sobald sich der Transponder in Lesereichweite befindet, danach kann die Nummer angezeigt oder verarbeitet werden. Durch die Verwendung von Radiowellen gegen über optischen Erkennungen ergeben sich unter anderen folgende Vorteile: [10]

1. Es ist möglich mehrere Transponder im Lesefeld selektiv zu lesen
2. Diese können variable Daten besitzen und vom Lesegerät empfangen und verarbeite bzw. gespeichert werden
3. Es besteht die Möglichkeit, die Transponder doch Nichtmetalle hindurch zu lesen, d.h. der Transponder muss für den Leser nicht sichtbar sein

[10]
Durch diese Vorteile, ist RFID den Barcodes weit überlegen und wird z.B. oft für Auslieh-Systeme bei Bibliotheken verwendet. [10]

5.2 Smartcard

Smartcard (auf deutsch Chipkarte) sind Plastikkarten, welche zum Informationsaustausch oder zur Identifiziert genutzt werden. Das gebräuchlichste Kartenformat, welches heutzutage eingesetzt wird ist das in ISO 7811 beschriebene ID-1 Format. Chipkarte besitzen, im Gegensatz zu seinem Vorgänger der Magnetstreifenkarten, einen eingebetteten Chip, welcher für die Speicherung von Daten verwendet werden kann. Der große Vorteil zu anderen Kartenarten, ist das die Informationen auf diesem Chip gegen unberechtigtes Auslesen oder Manipulation stark geschützt sind. So ein Schutz wird meist durch kryptografische Methoden umgesetzt und Mikroprozessorkarten sind zusätzlich in der Lage Daten logisch zu verarbeiten. Die meisten Chipkarten haben eine Chip mit 8 oder 6 Kontakteinrichtungen über die mit einem Lesegerät kommuniziert wird. [11]
Neben den vorher erwähnten kontaktbehafteten Karten, bei denen es ein direkte elektrische Verbindung zwischen dem Chip und dem Terminal zur Kommunikation geben muss, gibt es auch noch kontaktlose Karten, welche kein Problem mit statischer Elektrizität oder Oxidation haben. Diese Karten kommunizieren mit dem Terminal "drahtlos" und werden mittels induktiver Kopplung mit Strom versorgt. Des Weiteren gibt es keine Probleme mit der Orientierung bei solchen Karten, denn diese müsse nicht in ein Terminal, welche dadurch verschmutzt werden können, eingesteckt werden. Der große Nachteil von kontaktlose Karten ist, dass Personen, welche solche Karten mit sich tragen, ohne ihr Wissen erfasst werden können. [11]

5.2.1 Sicherheitstechniken

Smartcards sind vor allem im Bereich der IT-Sicherheit vertreten, das liegt daran, dass sie effizient gegen Missbrauch schützen können, wenn sie richtig implementiert sind. Auch sind sie und eventuelle Terminals, welche zum auslesen verwendet werden, billig in der Anschaffung und Verwendung. Die Sicherheit einer Chipkarte, basiert auf mehreren Teilbereichen: [11]

Benutzeridentifizierung Mittels dieser soll sichergestellt werden. das die Karte nur von befugten Personen benützt werden kann. Dies geschieht sehr oft durch eine Eingebe eines PINs (Personal Identification Number, eine Geheimzahl). Beispiele für so eine Abfrage wäre der PIN, der zur entsprungen einer SIM-Karte bei Mobiltelefon erforderlich ist oder der PIN, welcher jedes mal bei Geldabheben beim Bankomat eingegeben werden muss. Wenn man z.B. bei Mobiltelefonen diese 4-stellige Geheimnummer zu oft (üblicherweise 3 mal) falsch eingibt, dann wird die Karte gesperrt und ein PIN mit deutlich mehr Stellen (oft PUC (Personal Unblocking Key) oder SuperPIN genannt) wird abgefragt. Mit einer Zeitverzögerung verbunden macht ein systematische Ausprobieren (Brute-Force-Methode genannt) aller Kombinationen als Angriffsszenario der Karte recht wenig Sinn. [11]

Authentisierung des Kartenterminals Ein weiterer wichtigen Aspekt für die Verhinderung von Missbrauch spielt die Authentisierung des Kartenterminals. Um nicht Opfer von z.B. falschen Bankomaten zu werden, muss schon beim Design von Anwendung darauf geachtet werden, dass theoretisch der gesamte Datenverkehr zwischen Terminal und Karte abgehört oder manipuliert werden kann. [11]

5.3 Card Terminals

Damit Smartcards auch funktionieren, muss es ein Gegenstück geben, welches die Informationen der Karte lesen kann, die Karten-Terminals. Allgemein wird zwischen zwei Terminals unterschieden, die speziell für eine Anwendung gebaut worden sind und solche, die im Computerhandel erhältlich sind und viele Kartentypen unterstützen. Spezielle Karten, werden so konstruiert, dass einige Funktionen für alle Kartenleser nutzbar und die sicherheitsrelevanten Funktionen nur von Spezialterminals benutzbar sind. [11]

5.3.1 Sicherheitssysteme

Bei solchen Spezialterminals soll verhindert werden, dass manipulierte Karten in ein Terminal eingeführt werden können, dazu wird ein sogenannter “Shutter” eingesetzt. Dabei verschließt sich die Öffnung und die Karte wird tief in das Innere des Terminals befördert. Um die Echtheit einer Karte zu prüfen, werden zahlreiche Methoden wie Überprüfung von physikalischen Eigenschaften (z.B. Stromverbrauch, Wärmeentwicklung oder Masse), Seriennummern oder sie wird Extemsituaionen ausgesetzt. [11]

5.3.2 Probleme von Termianls

Bei Terminals gibt es oft Probleme mit der Toleranz für Fehler, welche zugelassen werden müssen. Optische Sensoren sind an ihre Arbeitstemperatur und Arbeitsluftfeuchtigkeit angewiesen, damit sie zuverlässig arbeiten können. Es ist auch möglich, die Shuttertechnik zu umgehen, es gibt z.B Anleitungen, wie man den Shutter offenhält und beispielsweise mit einen Draht spezielle Sensoren austickst. [11]

5.4 Projektumfeld

Projektumfeld: "Magna-Konzern"
Die Integration eines anderen Unternehmen ins eigene bring meistens gewisse Herausforderungen mit. Da dadurch das eigene Unternehmen wächst, kann es zu größeren sicherheitstechnischen Herausforderungen kommen. Ein großer Konzer kann sich überlegen, statt jedem neuen Mitarbeiter Schlüssel zu übergeben, elektrische Schlösser mit Smartcards umzusetzen, was auf längere Sicht viel Arbeits- und Verwaltungs-Aufwand einsparen würde.

6 Desaster Recovery

6.1 Projektumfeld

Projektumfeld: “UN Bojen-Netz”

7 Algorithmen und Protokolle

Es gibt viele Möglichkeiten mit Peripherie oder anderen Systemen zu kommunizieren. Weit verbreitete Schnittstellen bzw. Protokolle sind USART, SPI und I^2C.

7.1 USART

"Universal Synchronous and Asynchronous Serial Receiver and Transmitter" (USART) ist eine serielle Schnittstelle, welche zur synchronen und asynchronen Datenübertragung eines Mikrocontrollers verwendet wird. Dabei wird das byteweise Senden und Empfangen von seriellen Daten ermöglicht. Damit gleichzeitig gesendet und empfangen werden kann, besitzt der USART einen Sender (Transmitter) und Empfänger (Receiver), welche unabhängig von einander arbeiten können. USART ist eine Erweiterung von "Universal Asynchronous serial Receiver and Transmitter" (UART) und bietet, wie man am Namen schon erraten kann, zusätzlich eine synchrone Datenübertragung. Das Erzeugen einer Bautrate (Übertragungsrate), welche für die Kommunikation benötigt wird, geschieht durch den USART. Nichtsdestotrotz, das über die Register auf dem Mikrocontroller zahlreiche Baudraten eingestellt werden können, haben sich gewisse Baudraten als Standard etabliert. Beispielsweise 300, 2400, 9600 und 38400 Baud. U(S)ART wird oft in Verbindung mit der RS232 Schnittstelle, wofür sehr oft eine Baudrate von 9600 Boud verwendet wird, verwendet, wofür es aber eine zusätzlichen einen IC, welcher die Kommunikation umwandelt, benötigt wird. U(S)ART besteht aus 2 Daten-Leitungen und GND: [12][13][14]

- TxD (Tranceive Data = Daten Senden)
- RxD (Receive Data = Daten Empfangen)

[12][13][14]
Es gibt viele Einsatzzwecke für U(S)ART: [12][13][14]

- Kommunikation mit dem PC
- Debugging
- Kommunikation unter Roboter-On-Board-(Sub)Systemen
- Datenübertragung per Infrarot
- Bootloader

[12][13][14]
Für die Kommunikation am PC benötigt es einen Terminalemulator welcher vom angeschlossenen Port (dem COM-Port) die Daten auslesen kann. [12][13][14]

7.2 SPI

SPI (Serial Peripheral Interface) ist ein externer serieller Mikrocontroller Bus der von Motorola ins Leben gerufen werde. Dieser verwendet prinzipiell 4 Leitungen und ist nicht so stark standardisiert wie z.B. I^2C (siehe 3.2). Er wird in erster Linie für die synchrone serielle Kommunikation von Hostprozessor und Peripheriebausteinen benutzt, wobei auch mehrere Prozessoren miteinander verbunden werden können. Die 4 Leitungen sind: [15][16]

- Eine Signal-Clock SCLK, welcher vom Bus-Master an alle Slaves versendet wird. Alle SPI-Signale sind mit dieser Clock synchronisiert.
- Ein "Slave-Select-Signal" für jeden Slave, SSn, wird vom Master zum Auswählen des Slaves verwendet.
- Eine Datenleitung vom Master zu den Slaves, MOSI (Master Out-Slave In) genannt.
- Eine Datenleitung vom den Slaves zum Master, MISO (Master In-Slave Out) genannt.

[15][16]

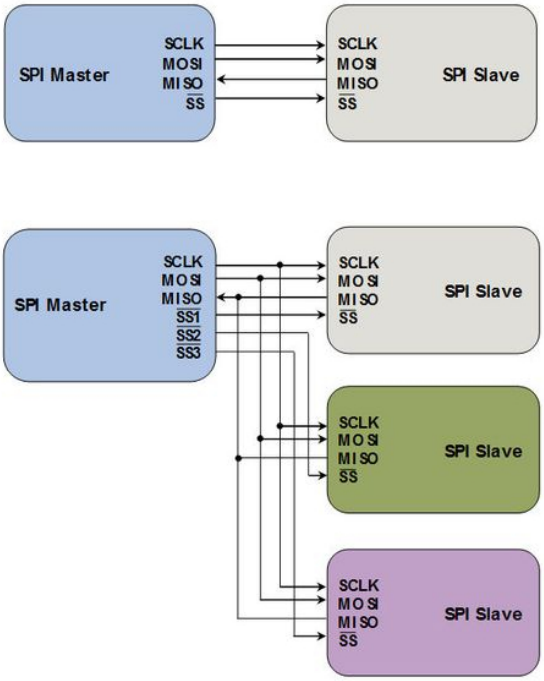

Abbildung 2: Zwei SPI-Bus Topologien. Die obere Abbildung zeigt einen SPI-Master, der mit eine Slave (Point-to-Point Topologie) verbunden ist. Die untere Abbildung zeit eine SPI-Master, der mit mehreren Slaves verbunden ist. [17]

SPI ist ein "Single-Master-Kommunikations-Protokoll", d.h. es initialisiert ein zentrales Gerät alle Kommunikationen zu den Slaves. Wenn der SPI-Master eine Datenverbindung zu oder von einem Slave starten will, dann setzt er die entsprechende "SS-Leitung" auf "Low" und aktiviert das Clock-Signal. Es gibt vier Kommunikations-Modi, welche verschiedene "SLCK-idle-Level" definiert. SPI wird beispielsweise im Meßtechnikbereich und Audiobereich eingesetzt und ist vor allem bei "high speed" Datentransfer vor I^2C (siehe 3.2) vorzuziehen, denn SPI kann bis Geschwindigkeiten über 10 Mbps übertragen. [15][16]

7.3 I^2C

I^2C (siehe 3.2) wird oft als Gegensatz zu SPI (siehe 7.2) eingesetzt. Es bietet den Vorteil, weniger Leitungen zu benötigen und "eleganter" als SPI zu sein. [15]

7.4 Projektumfeld

Projektumfeld: "Loxone"
Es gibt viele Möglichkeiten Sensoren oder andere Peripherie an eine Mikrocontroller anzuschließen. U(S)ART eignet sich eher weniger um Sensoren mit einem kleine Mikrocontroller zu verbinden, damit dieser Daten an eine zentrale Einheit weiterleiten kann. Für einen solche Einsatzzweck würde sich eher ein Bus wie SPI oder I^2C eignen. Dabei kommt es dann auf den genaue Einsatzzweck drauf an, welcher von den beiden vorzuziehen ist.

8 Konsistenz und Datenhaltung

Bei jeder Datenverbindung sollte es eine Fehlerüberprüfung geben, damit die Konsistenz der gesendeten Daten auch gewährleiste werden kann. Der CAN-Bus hat sogar einige Möglichkeiten Fehler zu erkennen und darauf zu reagieren.

8.1 Fehlererkennung beim CAN-Bus

Das CAN-Protokoll (siehe 3.1) hat die Möglichkeit selber Fehler zu erkennen und sie anzeigen, was für eine hohe Datenkonsistenz im CAN-Netzwerk sorgen soll. Dafür sind im CAN-Protokoll drei Mechanismen auf der Nachrichtenebene implementiert: [3][18]

8.1.1 Cyclic Redundancy Check

Der "Cyclic Redundancy Check" (CRC) sichert die Informationen des Rahmens, das durch das Hinzufügen von redundanten Prüfbits erreicht wird. Der Empfänger berechnet diese Prüfbits aus den empfangenen Bits neu und vergleicht diese mit den empfangenen Prüfbits. Sollte etwas nicht übereinstimmen, dann ist ein CRC-Fehler aufgetreten. [3][18]

8.1.2 Frame-Check

Beim "Frame-Check" (auch message frame check) wird die Struktur des übertragenen Rahmens überprüft. Dabei werden Bitfelder und die Rahmenlänge mit dem vorgegebenen festen Format verglichen. Sollte etwas nicht übereinstimmen, dann ist ein Formatfehler aufgetreten. [3][18]

8.1.3 ACK-Fehler

Jeder Empfänger quittiert das empfangen eines Rahmens/Frames durch ein positives Acknowledgements (ACK). Sollte der Sender keine Acknowledgements bekommen (ACK-Fehler), so deutet das auf einen möglichen Übertragungsfehler der nur vom Empfänger erkannt worden ist, das eine Verfälschung des ACK-Feldes aufgetreten ist oder das die Nachricht nicht vom Empfänger erhalten worden ist. [3][18]

Des Weiteren sind im CAN-Protokoll zwei Mechanismen zur Fehlererkennung auf der Bitebene implementiert: [3][18]

8.1.4 Monitoring

Der Sender beobachtet beim Senden den Pegel des Busses. Sollte er dabei einen Unterschied zwischen gesendetem und empfangenen Bits erkennen, dann weiß er es liegt ein Fehler vor. Damit können alle globalen Fehler und am Sender auftretende Bitfehler sicher erkannt werden. [3][18]

8.1.5 Bit-stuffing

Beim Bit-stuffing (auch Stuff Check genannt), wird auf der Bitebene die Codierung der Einzelbits überprüft. Dafür verwendet das CAN-Protokoll die NRZ-Codierung (Non-Return-to Zero Codierung), welche eine maximale Effizienz bei der Bitcodierung gewährleisten soll. Dabei wird vom Sender nach jedem fünften aufeinanderfolgenden gleichen Bit ein Suff-Bit mit komplementärem Wert gesendet, welches vom Empfänger dann automatisch wieder entfernt wird. Das Bit-Stuffing soll einerseits zur Synchronisation und andererseits zur Erkennung von z.B. Leitungsstörungen verwendet werden.

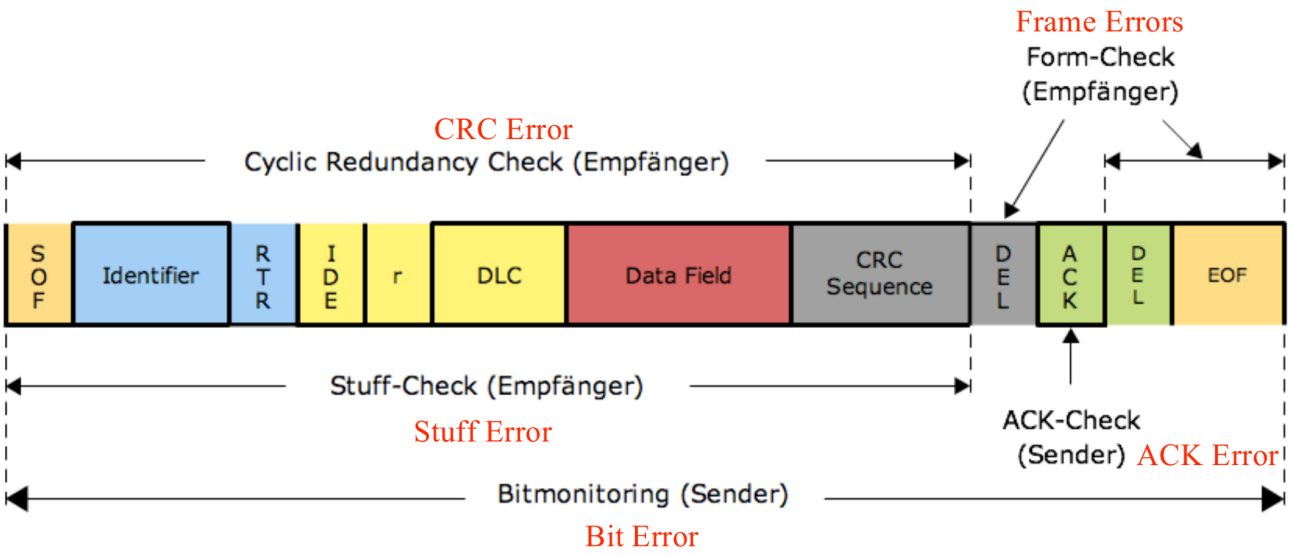

Abbildung 3: Bereiche der Fehlererkennung in Daten-/Remoteframe [18]

Wird ein oder mehrere Fehler von mindestens einem Knoten, mit Hilfe der oben beschriebenen Mechanismen bzw. Methoden, entdeckt, so wird die gerade laufende Übertragung durch das Senden eines "Error Flags" abgebrochen. In Folge wird die Annahme der fehlerhafte Botschaft durch die Empfänger abgebrochen und somit die globale Datenkonsistenz sichergestellt. Danach beginnt der Sender der fehlerhaften Nachricht automatisch die Nachricht erneut zu senden, das wird "Automatic Repeat Request" genannt. Sollte ein Fehler mehrmals hintereinander auftreten, dann wird automatisch dieser Knoten abgeschaltet und so aus dem Netzwerk geworfen. [3][18]

8.2 Paritätsprüfung

Die Paritätsprüfung (parity check (PAR)) ist eine einfaches und häufig angewendetes Verfahren zur Erkennung von Datenübertragungsfehlern bei der Übertragung vom Datenblöcken. Dabei werden Datenblöcke als Bit-Folge übermittelt und die Anzahl der 1-Bits gezählt. Es kann entweder längs (longitudinal redundancy check (LRC)) oder vertikal (vertical redundancy check (VRC)) gezählt werden. Ferner wird außerdem zwischen geraden (even) und ungeraden (odd) Parität unterschieden. Die Paritätsprüfung kann aber nur eine ungerade Anzahl an Datenübertragungsfehlern erkennen. [19]

8.2.1 Beispiele:

1	2	3	4	5	6	7	**parity bit**
1	0	0	0	0	0	1	**0**

Tabelle 1: A (ASCII) bei gerader Parität [19]

1	2	3	4	5	6	7	**parity bit**
1	0	0	0	0	0	1	**1**

Tabelle 2: A (ASCII) bei ungerader Parität [19]

A	1	0	0	0	0	0	1
S	1	0	1	0	0	1	1
C	1	0	0	0	0	1	1
I	1	0	0	1	0	0	1
I	1	0	0	1	0	0	1
parity	**1**	**0**	**1**	**0**	**0**	**0**	**1**

Tabelle 3: Datenblock “ASCII” (ASCII) bei gerader Parität und VRC [19]

8.3 Projektumfeld

Projektumfeld: “Loxone”
Es gibt viele Möglichkeiten Fehler zu erkenne und die Konsitenz von Datenübertragungen zu wahren. Der CAN-Bus hat viele Mechanismen, damit die Daten auch richtig bei den anderen Kommunikationsteilnehmern auch ankommt und vor allem auch wirklich richtig ist. Bei einer Implementation, die auf einer solchen Technologie aufbaut, muss auf solche Sachen dann nicht mehr acht gegeben werden und man kann sich sicher sein, das die Informationen auch Konsistent sind.

9 Conclusio

Die Vormarsch von Internet of Thing und Smart Home ist nicht mehr aufzuhalten. Doch muss dabei einiges Beachtet werden, damit alles reibungslos funktioniert und die verschiedenen System auch miteinander kommunizieren können. Es gibt Bereiche, in denen was falsch gemacht und wo etwas vergessen werden kann. Aber vieles was hinter solchen Technologien steckt ist sehr ausgereift und macht auch Sinn. Doch das wichtige überhaupt ist, dass es einen einheitlichen (offenen) Standard gibt, welchen alle verwenden und von allen Akzeptiert wird. Oft dauert das eigene Zeit doch im Endeffekt profitierten alle von so einer Situation.

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[1] Volker Andelfinger, Till Hänisch, “Grundlagen: Das Internet der Dinge,” in *Internet der Dinge*, Springer, 2015.

[2] Christian Flörkemeier, Friedemann Mattern, “Vom Internet der Computer zum Internet der Dinge,” tech. rep., ETH Zürich, 2010.

[3] “CAN Bus Grundlagen.” `http://www.me-systeme.de/canbus.html`, 2015.

[4] “I^{2}C - Mikrocontroller.net.” `http://www.mikrocontroller.net/articles/I%C2%B2C`, 2016.

[5] “Pulsweitenmodulation - Mikrocontroller.net.” `http://www.mikrocontroller.net/articles/Pulsweitenmodulation`, 2016.

[6] “Pulsweitenmodulation - RN-Wissen.de.” `http://rn-wissen.de/wiki/index.php/Pulsweitenmodulation`, 2014.

[7] Ben Fry, Casey Reas, “Processing: A Programming Handbook for Visual Designers, Second Edition.” `https://processing.org/tutorials/electronics/imgs/fg39-12.svg`, 2014.

[8] Philipp Adler, Adin Karic, “Sicherheit.” 2015.

[9] Erwin *, “Encryption and Key Exchange in Wireless Sensor Networks,” Seminararbeit, Technische Universität Braunschweig, 2013.

[10] Christian Kern, *RFID für Bibliotheken*. Springer, 2011.

[11] Sven Tantau, “Grundlagen und Anwendungsgebiete von Chipkarten,” Seminararbeit, Universität Bonn, 2002.

[12] RN-Wissen.de, “UART - RN-Wissen.de.” `http://rn-wissen.de/wiki/index.php?title=UART`, 2011.

[13] Herbert Bernstein, *Mikrocontroller: Grundlagen der Hard- und Software der Mikrocontroller ATtiny2313, ATtiny26 und ATmega32*. Springer, 2015.

[14] mikrocontroller.net, "USART." http://www.mikrocontroller.net/mc-project/Pages/AVR/USART/usart.html, 2011.

[15] byteparadigm.com, "Introduction to I^2C and SPI protocols - Byte Paradigm - Speed up embedded system verification." http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/?/article/AA-00255/22/Introduction-to-SPI-and-IC-protocols.html, 2016.

[16] MikroComputerTechnik PAULundSCHERER, "mct.de: SPI - Serial Peripheral Interface." http://www.mct.de/faq/spi.html, 2006.

[17] byteparadigm.com, "Introduction to I^2C and SPI protocols - Byte Paradigm - Speed up embedded system verification." https://s-media-cache-ak0.pinimg.com/736x/d1/1d/f3/d11df3e934973460d51899a74b8b7b17.jpg, 2016.

[18] I. Bernsdorf, T. Vogt, L. Stojanovic, S. Szabadi, "CAN-Bus," Ausarbeitung, Beuth Hochschule für Technik Berlin, 2009.

[19] Karlheinz Korbmacher, Lydia Korbmacher, "Paritätsprüfung." http://www.luk-korbmacher.de/Schule/Orga/par.htm, 1996.