



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT



Lehrstuhl für Technische Elektronik

Prof. Dr.-Ing. Dr.-Ing. habil. Robert Weigel

Prof. Dr.-Ing. Georg Fischer

Bachelorarbeit

im Studiengang

„Elektrotechnik, Elektronik und Informationstechnik (EEI)“

von

Christof Pfannenmüller

zum Thema

Aufbau und Inbetriebnahme einer mobilen Basisstation für feldstärkebasierte Lokalisierung

Betreuer: Dipl.-Ing. Felix Pflaum

Beginn: 25.04.2016

Abgabe: 26.09.2016

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.

Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 26. September 2016

Christof Pfannenmüller

Kurzfassung

Ziel der vorliegenden Arbeit war

Zur Lokalisierung von mobilen Sensorknoten, welche in einem Sub-GHz-Frequenzbereich von 868 MHz arbeiten, sollte eine energieeffiziente Art der Ortung umgesetzt werden. Dazu sollte mithilfe einer auf Feldstärke basierten Ortung die genaue Position der Sensorknoten festgestellt werden. Um dies zu ermöglichen wurde eine Basisstation mit sechs Transceiver-ICs der Bauart TDA5340 entworfen. Die Steuerung übernahm ein Mikrocontroller der Baureihe XMC4500. Diese Basisstation sollte dabei ein Auslesen der empfangen Daten sowohl über den USB-Standard, als auch über Ethernet als zweite Kommunikationsschnittstelle ermöglichen. Die dabei verwendete Hardware basierte zum Großteil auf Bauteilen des Herstellers Infineon. Die Platine der Basisstation wurde mit Altium Designer entwickelt und umgesetzt. Dabei wurde die Verbindung zwischen der Steuereinheit und den Transceivern mit dem SPI-Protokoll umgesetzt. Die einzelnen Sende-/Empfangseinheiten wurden dabei gleichmäßig in alle Raumrichtungen zeigend angeordnet, sodass auch die Abstahlung der Antennen über alle Raumrichtungen gleichförmig verteilt ist. Die Funksegmente der Platine wurden so gestaltet, dass diese bei Bedarf abgetrennt und mit einer Kabelverbindung weiter voneinander entfernt werden konnten. Die Peripherie der verwendeten TDA5340 Transceiver generierte im Programmablauf nach einer Kommunikation mit dem Sensor ein Interrupt Signal. Dies erlaubte dem Mikrocontroller die Daten der einzelnen Empfangseinheiten auszulesen, zu speichern und zu einem späteren Zeitpunkt weiterzuleiten. Beim Messen der Feldstärke wurde ausgenutzt, dass die vorliegende Feldstärke bereits durch den TDA5340-Empfänger zur Verfügung gestellt wurde.

Abstract

add abstract

Abkürzungsverzeichnis

PCB	Printed Circuit Boards
EDA	Electronic Design Automation
SPI	Serial Peripheral Interface
DRC	Design-Rule-Check
FIFO	First In – First Out
SMA	Sub-Miniature-A
LQFP	Low Profile Quad Flat Package
BGA	Ball Grid Array
JTAG	Joint Test Action Group
TVS	Transient Voltage Suppressor
LDO	Low Drop-Out
SOT	Small Outline Transistor
IC	Integrierter Schaltkreis
SMD	Surface Mounted Device
STEP	Standard for the Exchange of Product Model Data
CAD	Computer-aided design
NC	Numerical Control
IDE	integrated development environment
GUI	Graphical User Interface
SDK	Software development kit
CPU	Central Processing Unit

USIC Universal Serial Interface Channel

ETH Ethernet MAC

USB Universal Serial Bus

GPIO General Purpose Input/Output

Inhaltsverzeichnis

Einleitung

1.1 Motivation

1.2 Zieldefinition

1.3 Projektmanagement

Als Versionskontrolle für das Projekt wurde Github eingesetzt.

Platinenaufbau

2.1 Vorüberlegungen

Um Sicherzustellen, dass alle Antennen gleichmäßig in die sechs vorgegebenen Raumrichtungen abstrahlen, sollte bereits die Platine symmetrisch aufgebaut werden. Dazu wurden zuerst das Layout der sechs identischen Transceiver-Einheiten mit dem TDA5340 Baustein und den Antennen erstellt und anschließend gleichmäßig um die weiten für die Schaltung notwendigen funktionellen Segmente angeordnet.

2.2 Layoutprogramm Altium Designer

Bei dem Entwicklungswerkzeug „Altium Designer“ des Entwicklers Altium Limited handelt es sich um ein System zum Entwurf von gedruckten Schaltungen oder **PCB!s** (**PCB!s**). Ein solches Programm wird auch als **EDA!s** (**EDA!s**) oder ECAD für electronic **CAD! (CAD!)** bezeichnet, da es den Entwickler bei der Umsetzung der Anforderungen in einen Schaltplan unterstützen soll. Wie viele andere **EDA!-Programme** ist auch Altium Designer so aufgebaut, dass sich der Entwickler zuerst mit dem allgemeinen symbolisierten Schaltplan befassen kann und erst zu einem späteren Zeitpunkt die tatsächliche Anordnung der Bauteile auf dem **PCB!-Substrat** festgelegt wird. Somit können zuerst im Schematic Editor die Funktionen der Schaltung umgesetzt werden. Dazu werden die verwendeten Bauteile aus zuvor angelegten Bibliotheken verwendet oder es werden bestehende Bibliotheken verwendet, die etwa vom Hersteller der Bauteile zur Verfügung gestellt werden. Altium selbst bietet hierfür auch diverse Möglichkeiten an und stellt Bauteile nach Hersteller und Art geordnet bereit. In den Bibliotheken sind alle im weiteren Verlauf benötigten Informationen über die einzelnen Bauteile enthalten. So liegen dort etwa entsprechenden Abbildungen für das Bauteil im Schaltplan vor. In den sogenannten „Footprints“ zu jedem Bauteil, welche ebenfalls in den Bibliotheken enthalten sind, wurde zuvor die, für das physikalische Gehäuse, notwendigen Abmessungen und Löt pads festgelegt. Da es Bauteile, wie den verwendeten Mikrocontroller, in verschiedenen Gehäusen geben kann, besteht somit auch die Möglichkeit hier verschiedene Footprints

zu wählen. Da viele Gehäuse herstellerübergreifend genormt sind, konnten teilweise bestehende Footprints genutzt werden oder diese mehrfach verwendet werden.

Altium Designer ist dabei in drei Teilbereiche unterteilt: im „Board Planning Mode“ liegt der Fokus auf dem Anordnen der einzelnen Bauteile und Komponenten auf der Leiterplatte, außerdem wird in diesem Bereich die Form und Ausmaße der Leiterplatte festgelegt. Im 2D-Modus des **PCB!**-Editor lassen sich anschließend die aus der Definition im Schaltplan ergebenden elektrischen Verbindungen örtlich auf den verschiedenen Kupferebenen (Layern) anordnen. Die Hauptarbeit findet also in diesem Teil des **PCB!**-Editors statt. Der 3D-Modus dient anschließend zur Evaluation des Designs und zur Anpassung an Gehäuse oder andere Komponenten.

Wegen der Größe des Projekts wurde zur besseren Übersicht ein so genanntes „Multi-Sheet-Design“ erstellt. Dadurch war es möglich die verschiedenen funktionellen Blöcke der Basistation auf getrennte Blätter des Schaltplans zu verteilen. Der Mikrocontroller, die Spannungsversorgung und der Transceiver sowie die für eine Netzwerkkommunikation notwendigen Bauteile wurden dabei auf getrennten Seiten angeordnet und die elektrischen Verbindungen erstellt. Da der Transceiver und die entsprechende Peripherie sechs mal genau identisch verwendet wurde und auch auf dem **PCB!**-Substrat mehrfach mit Leiterbahnen verbunden und angeordnet werden musste, wurde hierfür ein so genanntes „Multi-Channel“-Design gewählt. In Altium Designer können mit diesem Feature identische Schaltungsteile einmal angeordnet und mit Leiterbahnen verbunden werden und dieses Design auf alle anderen entsprechenden Schaltungsteile angewendet werden. Somit muss das aufwändige Anordnen der Bauteile und die Führung der Leiterbahnen nur bei einem der Kanäle durchgeführt werden. Mit Hilfe eines übergeordneten Sheet-Symbols wurde der Schaltplan des Transceivers in den Schaltplan des Mikrocontroller eingefügt und diesem somit untergeordnet. Über Ports welche zum Schaltplansymbol hinzugefügt werden, lassen sich elektrische Verbindungen zwischen Netzen innerhalb des Schaltplans für Transceiver und Mikrocontroller erstellen. Dabei wird im Transceiver-Schaltplan ein Port und hinzugefügt, der mit elektrischen Netzen verbunden werden kann. Auf dem Schaltplan-Symbol im Mikrocontroller-Schaltplan wird ein entsprechender gleichnamiger Port erstellt, der mit Netzen am Mikrocontroller verbunden werden kann. Da die drei Verbindungen der **SPI!** (**SPI!**)-Kommunikation jeweils aus einem Netz bestehen und etwa alle MISO-Leitungen an dem selben Pin des XMC4500 und den selben Anschluss bei allen TDAs angebunden sind konnte hierfür ein einfacher Port verwendet werden. Alle anderen Anschlüsse, wie etwa die Auswahlleitung für die **SPI!**-Verbindung, welche für jede der sechs verschiedenen Transceiver-Einheiten mit einem anderen Anschluss des Mikrocontroller verbunden sein mussten mussten deswegen mit dem Repeat-Kommando erstellt werden. So wird im untergeordneten Schaltplan, in diesem Fall dem des TDA, der Port beliebig benannt, etwa als „NCS“. Der auf dem Schaltplansymbol erstellte Port wird dagegen in „Repeat(NCS)“ umbenannt. Globale Netze wie die Versorgungsspannung von 3,3 Volt oder die Masse müssen dabei nicht als Port hinzugefügt werden. Altium Designer stellt deren Verbindungen automatisch her. Wird nun noch der Name des Schaltplansymbols nach dem folgenden Muster angepasst, wird Altium Designer beim Kompilieren des Projektes ein Multi-Channel-Design erstellen, die Kanäle wie angegeben durchnummerieren und den Kanal im **PCB!**-Editor entsprechend mehrfach erstellen. Dazu sollte der Name die Struktur „Repeat(<Name>,<Startnummer> ,<Endnummer>)“

aufweißen. Anschließend kann die Anordnung und das Routing des ersten Kanals erfolgen. Da für jeden Kanal ein eigener so genannter „Room“ erstellt wird, lassen sich, nach dem erstellen der Leiterbahnen des ersten Kanals, die Anordnungen der Bauteile und Leiterbahnen mit dem „Copy Room Format“-Befehl auf alle anderen Kanäle erweitern. Ein Room bezeichnet dabei eine Gruppe an Elementen und einen Bereich in dem diese angeordnet sind und erlaubt das Anordnen aller enthaltenden Elemente auf der Platine.

Altium Designer nummeriert automatisch die verwendeten Bauelemente durch um eine genaue Identifikation eines Bauteiles zu erlauben und erstellt automatisch einen Aufdruck neben dem Lötspfad mit dem Bauteilnamen in der entsprechenden Ebene. Ein solcher Bestückungsdruck zur leichteren Anordnung der Bauteile auf der fertig entwickelten Platine wurde nur auf der Hauptplatine erstellt. Auf den Teilplatinen für die Transceiver wurde dieser aus Platzgründen weggelassen. Um sicherzustellen, dass die sich aus der Bauteilanordnung ergebenden Pads und die Leiterbahnen auch fertigbar sind stellt Altium Designer zwei **DRC!**s (**DRC!**s) bereit. Im ersten Live-**DRC!** werden „Violations“ also Bauteile mit Verstößen durch ein farbiges Overlay markiert. Im folgenden kompletten Test zeigt Altium alle weiteren Verstöße an. Als Verstoß gelten etwa Leiterbahnen unterschiedlicher Netze die sich berühren und zu einem Kurzschluss führen würden oder auch Verletzungen der festgelegten Abstandsregeln (Clearance). Alle Regeln für Violations können im „PCB Rules Editor“ eingestellt werden und so an die Möglichkeiten des **PCB!**-Herstellers angepasst werden. Da die Leiterplatte für die Basisstation von Multi Circuit Boards Ltd. hergestellt wurde, konnten die auf der Webseite dieses Herstellers angegebenen Designregeln für den **DRC!** übernommen werden. Da Altium im **DRC!** sich überschneidende und falsch verbundene Leiterbahnen angezeigt werden, muss kein extra „Layout vs. Schematic“-Test durchgeführt werden, da durch den **DRC!** Unterschiede zwischen Schaltplan und dem **PCB!**-Layout bereits angezeigt würden.

2.3 verwendete Hardware

2.3.1 TDA5340

Der verwendete Transceiver TDA5340 wird von Infineon Technologies AG entwickelt und vertrieben. Er ist Teil der SmartLEWISTM Produktfamilie die energiesparende Lösungen für Funkanwendungen im Frequenzspektrum unterhalb von einem Gigahertz bietet. Der Transceiver kommuniziert mit seinem Host über das **SPI!**-Protokoll, der Mikrocontroller ist in diesem Fall sternförmig mit den einzelnen TDA-Bausteinen verbunden, die als Slaves fungieren. Die Daten werden auf drei gemeinsamen Leitungen übertragen, eine vierte Leitung dient dem XMC zur Auswahl des gewünschten Slaves für die Kommunikation. Diese „not Chip select“-Leitung (NCS) arbeitet active-low, sodass der jeweilige TDA5340 eine Interaktion akzeptiert sobald diese vom XMC-Baustein auf Erdpotential gezogen wird. Von den drei eigentlichen Datenleitungen fungiert eine als reiner Ausgang des Masters bzw. Dateneingang des TDA (MOSI), eine zweite als Eingang des Masters (MISO) und die dritte als ein vom XMC getriebenes Clock-Signal. Bei dem auf MISO und MOSI anliegenden Signal handelt es sich um ein unipolar kodierte non-return-to-zero Signal, welches einer logischen 0 bei Erdpotential entspricht. Der TDA unterstützt acht verschiedene Instruktionen, die es erlauben entweder einzelne Register des Bausteins

zu lesen bzw. zu schreiben, auf mehrere hintereinander folgende Register oder auf die beiden Puffer des Bausteins zuzugreifen. In den beiden Puffern, die als **FIFO! (FIFO!)**-Strukturen aufgebaut sind, werden die vom TDA erkannten und demodulierten bzw. die auf Übertragung wartenden Signalpakete zwischengespeichert. Diese Zwischenspeicherung soll den Mikrocontroller entlasten, so können entsprechende Datenpakete dem TDA5340 mitgeteilt werden und dieser übernimmt selbsttätig eine korrekte Modulation und Übertragung mit den eingestellten Parametern.

Der TDA5340 kann sowohl mit einer Spannungsversorgungsspannung von 5V als auch bei 3,3V arbeiten. Da aber der XMC nur bei letzterer betrieben werden kann, wurde der TDA-Baustein und die externe Beschaltung einfachheitshalber auch auf 3,3V ausgelegt.

Um zu einem späterem Zeitpunkt eine größere Entfernung zwischen den einzelnen Antennen, und somit auch den jeweiligen Transceivern zu erlauben, wurde eine Sollbruchstelle vorgesehen. Dadurch könnten die gesamte Baugruppe von der Mutterplatine entfernt werden, was unter Umständen notwendig gewesen wäre um größere Unterschiede in der Signalstärke an den einzelnen Transceivern zu erhalten, und somit eine bessere Auflösung in der Ortung zu erlauben. Dazu wurden Anschlussleisten im Rastermaß 2,54mm an beiden Seiten der Sollbruchstelle vorgesehen. Die Verbindung der Transceiver-Einheiten mit der Hauptplatine wurde über diese Sollbruchstelle mit Leiterbahnen gewährleistet. Nach dem Abtrennen der TDA-Teilplatine, an der durch Bohrungen vorgesehenen Bruchstelle, wäre die elektrische Verbindung durch Kabel sichergestellt worden. Da es sich bei den zu übertragenden ausschließlich um digitale Signale handelt hätte dies unproblematisch mit ungeschirmten Flachbandkabeln ermöglicht werden können. Ein solches Vergrößern des Abstandes der Antennen war jedoch nicht notwendig. Neben der Versorgungsspannung, Masse und den vier für die **SPI!**-Kommunikation notwendigen Signalen wurden noch die drei multifunktionalen Digitalausgänge und der power-on reset-Pin dem XMC4500 über die Buchsenleisten zur Verfügung gestellt.

Die Antenne wurde am oberen Ende jeder TDA5340-Teilplatine vorgesehen. Als Anschluss für die Antenne wurde hier eine Koaxialbuchse in **SMA! (SMA!)**-Ausführung verwendet, welche auf 50Ω angepasst ist. Durch den Koaxialsteckverbinder konnte sichergestellt werden, dass alle notwendigen Frequenzen auch korrekt und ungedämpft passieren können. Das Anpassnetzwerk zwischen dem integrierten Transceiver und der verwendeten SMA-Buchse diente der Leistungsanpassung zwischen den Pins des TDA5340 und der 50Ω-Koaxialbuchse. Der Aufbau des Anpassnetzwerkes basiert auf einem von Stefan Erhard erstellten Schaltplan für eine Aufsteckplatine für das Evaluationsboard „XMC 2Go“ von Infineon. Durch die Verwendung von hoch abgestimmten Spulen und Kondensatoren mit Toleranzen von nur $\pm 0,05\text{pF}$ bei einem Nennwert von 2,5pF wurde die korrekte Anpassung sichergestellt.

Zur Verbesserung der Hochfrequenzeigenschaften wurden die nach dem Freiräume zwischen den Leiterbahnen mit einer Kupferfläche gefüllt, die mit dem Masseanschluss kontaktiert war. Durch die Verwendung von Vias, vor allem im Bereich des Anpassnetzwerkes, sollte eine niederohmige Verbindung zwischen den beiden Masseflächen auf der Ober- bzw. Unterseite der Platine erreicht werden. Daneben dienten diese, auf Nullpotential liegenden Vias, jedoch vor allem der Abschirmung der Pfade für die HF-Signale gegen mögliche Einkoppelungen aus der Umgebung, welche ankommende Funksignale stören könnten.

Obwohl der TDA5340 einen eingebauten Zwischenfrequenz-Filter hat, der über eine umschaltbare Bandweite verfügt, wurde ein externer Keramikfilter verwendet. Der TDA stellt dafür zwei Pins bereit, zwischen denen ein solcher Filter mit einer Frequenz von 10,7 MHz angeschlossen werden kann. Ohne einen hier extern angeschlossenen Filter würde der TDA als einfacher heterodyner Mischer direkt auf die Zwischenfrequenz 274 kHz heruntermischen. Bei Verwendung eines externen Keramik oder auch eines LC π -Filters kann das ankommende HF-Signal jedoch in zwei Stufen gefiltert werden, ehe es in das Basisband demoduliert wird, was zu einer höheren Signalqualität führt

Die elektrischen Verbindungen zwischen den beiden Eingängen des Low Noise Amplifier und dem Anpassnetzwerk wurden mit dem „Differential Pair Routing“-Feature von Altium Designer erstellt. Durch ein im Schaltplan auf die positive und die negative elektrischen Verbindung zwischen dem TDA5340 und dem Anpassnetzwerk wird das Leitungspaar als differentiell markiert. Anschließend kann mit dem interaktiven „Differential Pair Routing“ einer der beiden Leitungen begonnen werden. Altium Designer wird dabei selbstständig versuchen die zweite Leiterbahn des Paares so anzuordnen, dass die beiden Leiterbahnen symmetrisch und parallel zueinander liegen. Durch die Verwendung des „Differential Pair Routing“ versucht Altium Designer auch die Länge der beiden Verbindungen anzugleichen. Durch die Anpassungen der differentiellen Leiterbahnen wird eine gleichmäßige Übertragung sichergestellt. Da diese Leitungen hochfrequente Signale führen, ist eine genaue Anpassung notwendig.

Um Einkopplungen auf die Pfade für hochfrequente Signale zu vermeiden wurde das für den Transceiver benötigte Quarz möglichst weit vom Sende- bzw. von den Empfangsanschlüssen des TDA angeordnet. Aus diesem Grund wurde der für den Oszillator benötigte Quarz mit einer Frequenz $f_{Crystal} = 21,948717$ MHz zwischen dem **IC!** (**IC!**) und vorgesehenen Stiftleiste platziert. Die Frequenz des benötigten Quarzes ergibt sich aus dem Zusammenhang

$$f_{Crystal} = f_{IF2} * 80 = \frac{f_{IF1}}{39} * 80 = \frac{10,7MHz}{39} * 80 = 21,948717MHz \quad (2.1)$$

wobei die Zwischenfrequenz der ersten Stufe durch die internen funktionalen Blöcke des TDA und den Keramikfilter vorgegeben ist. Die weiteren Faktoren ergeben sich aus dem Aufbau des Empfängers und werden von Infineon bereitgestellt **TDA-UserManual**

2.3.2 XMC4500

Die Hauptsteuerung der Basisstation übernimmt ein Mikrocontroller der Bauart XMC4500, welcher aus der Mikrocontroller-Familie XMC4000 von Infineon stammt. Diese Baureihe stellt energieeffiziente **IC!**s bereit welche für industrielle Steuerungen und „Sense & Control“ optimiert sind. Der XMC4500 basiert auf einem Kern Cortex™-M4 des britischen Herstellers ARM™. Daneben ist der Mikrocontroller auf die von Infineon selbst entwickelte Entwicklungsumgebung DAVE™ angepasst. Im speziellen Anwendungsfall kommt die Variante des XMC mit 144 Pins und einem Flash-Speicher von 1024 Kilobit zum Einsatz. Der Chip ist dabei in ein **LQFP!** (**LQFP!**)-Gehäuse verbaut. Durch die Wahl dieses Gehäuses konnte die elektrische Verbindung mit der Platine relativ leicht durch löten erreicht werden. Im Gegensatz zum ebenfalls erhältlichen **BGA!** (**BGA!**)-Gehäuse des XMC sind in diesem alle Kontakte direkt erreichbar und können verlötet werden.

Zur Verteilung der entstehenden Abwärme wurde auch in diesem Bereich der Platine frei gebliebener Abschnitte zwischen den Leiterbahnen mit geerdeten Kupferflächen gefüllt. Durch teilweise auch mehrfache Durchkontaktierungen wurde sowohl eine saubere Kontaktierung der Flächen durchgeführt um Flächen schwimmenden Potentials zu vermeiden. Durch die Vias wurde aber auch die thermische Leitfähigkeit zwischen den beiden Kupferlagen erhöht und somit die Abgabe entstehender Wärme von den Bauteilen verbessert. Beim verwendeten **LQFP!**-Gehäuse des XMC4500 liegt die Rückseite des Halbleiters offen und ist nicht im Gehäuse verschlossen. Im Bereich unter der offenliegenden Rückseite des Chips ist deswegen zur Wärmeableitung ein Feld von 6x6 Vias vorgesehen. Dieser Aufbau dient dazu die Temperatur des Chips (junction temperature) auf den maximal erlaubten Wert $T_J = 150^\circ\text{C}$ zu beschränken.

Um die Ausgabe von aktuellen Systemzuständen zu ermöglichen wurden sieben Status-LEDs an freien Ausgänge des XMC4500 angeschlossen. Diese ermöglichten in active-low Ansteuerung eine Anzeige verschiedener im Mikrocontroller ablaufender Prozesse. Für vier der verwendeten Leuchtdioden wurde grün als Farbe gewählt, für die drei weiteren rot. Zur Vereinfachung eines Resets der Hardware wurde ein entsprechender Taster vorgesehen, mit dem der entsprechenden **PORST**-Pin des Mikrocontroller auf das 0V Potential gezogen wird und somit die Hardware zurückgesetzt wird. Die Programmierung des Mikrocontrollers erfolgt über das **JTAG!** (**JTAG!**)-Interface über welches auch das debuggen möglich ist. Der XMC4500 stellt dafür ein **JTAG!**-Modul bereit welches mit der in IEEE1149.1 festgelegten Standards übereinstimmt. Verwendet wird hierfür die achpolige Variante des Debug-Steckers bei dem der Platine vom **JTAG!**-Adapter Versorgungsspannung und Masse sowie Signale für Reset, Systemtakt und die Steuerleitung übergeben wird.

Für die Kommunikation des Mikrocontroller mit einem Computer wird die im XMC bereitgestellte Peripherie genutzt. Dabei arbeitet die bereitgestellte Hardware nach den Spezifikationen für USB 2.0 und den „On-The-Go“-Spezifikationen der Version 1.3. Zur Verbindung mit einem anderen Gerät wurde deshalb eine kombinierte Micro-USB-Buchse verwendet welche sowohl für Typ A oder Typ B Stecker geeignet ist. Um sowohl den Mikrocontroller als auch einen an die Basisstation angeschlossenen Computer gegen Fehlerströme über die USB-Leitung zu schützen wurden die Datenleitungen mit so genannten **TVS!** (**TVS!**)-Dioden, welche gegen Masse geklemmt sind geschützt. Sowohl positive als auch negative Spannungsspitzen werden dadurch gegen Masse kurzgeschlossen, was zum Schutz des XMC bzw. des angeschlossenen Computer führt. Um eine Verpolung bei Stromversorgung über die USB-Buchse, und somit eine Zerstörung, zu vermeiden wurde eine Schottky-Diode im Strompfad zum Spannungsregler vorgesehen. Diese soll einen Stromfluss im Verpolungsfall unterbinden. Da der Mikrocontroller für die Kommunikation über das USB-Interface die aktuelle Busspannung auf der USB-Leitung benötigt, muss der extra dafür vorgesehene Pin des XMC direkt und ohne schützende Schottky-Diode mit der 5V Leitung der USB-Buchse verbunden werden.

2.3.3 Ethernet

Die Ethernetschnittstelle der Basisstation basiert auf dem RelaxKit von Infineon. Genau wie im Evaluations Board des Herstellers Infineon wurde der Ethernet-Controller

KSZ8031RNL von Mircel Inc. verwendet. Dieser stellt alle wichtigen Peripherien selbst zur Verfügung und muss somit nur noch durch ein Quarz und diverse Kapazitäten und Induktivitäten an den Versorgungsleitungen ergänzt werden. Da die im Controller verbaute Stufe zur Interruptgenerierung nur über einen schwachen Pull-Up Widerstand verfügt, musste ein externer Widerstand von $1k\Omega$ verbaut werden. Am Reset-Eingang wurde ebenfalls ein Pull-Up Widerstand verbaut. Dieser wurde um zwei Dioden sowie einen Kondensator zu der im Datenblatt empfohlenen Verschaltung erweitert. So kann sichergestellt werden, das sowohl beim Anlegen einer Spannung an das Gesamtsystem, als auch bei einem Reset des Ethernetbausteins durch den steuernden Mikrocontroller alle Spannungen im sicheren Bereich liegen und die Funktion gewährleistet ist. Die dreizehn zum XMC4500 notwendigen Verbindungen wurden zur besseren Übersicht im Schaltplan in einem Signal-Kabelbaum zusammengefasst. Wegen der Gefahr von Rissen in Lötstellen wurde im Bereich um den Netzwerkstecker die Anordnung von Bauteilen vermieden. Da der KSZ8031RNL nicht lieferbar war und die anfallende Datenmenge nur von geringem Umfang ist, wurde der Controller und die entsprechende Netzwerkbuchse von Würth Electronics zunächst nicht bestückt. Somit wurde eine Verwendung des Ethernet-Controllers auch in der Software des XMC-Mikrocontroller nicht umgesetzt. Da jedoch ein entsprechendes Softwareprojekt für das RelaxKit von Infineon zur Verfügung gestellt wird

2.3.4 Spannungsversorgung

Die Bereitstellung der notwendigen Spannung sollte wahlweise über den zur Datenerfassung angeschlossenen Computer oder über ein externes Netzteil erfolgen. Zum Anschluss eines externen Netzteils wurden Lötanschlüsse für eine Steckerleiste im Rastermaß 2,54mm vorgesehen. Genau wie bei der Stromversorgung über die USB-Buchse wurde auch hier eine Schottky-Diode zum Verpolungsschutz der Schaltung integriert. Ausgelegt ist die Basisstation für ein Gleichspannungsnetzteil mit 5V Ausgangsspannung, durch den Aufbau mit den beiden verwendeten Schottky-Dioden und die mögliche Eingangsspannung des nachfolgenden Reglers wäre jedoch eine angeschlossene 6V-Stromquelle (bei vernachlässigtem Spannungsabfall an der Diode) unproblematisch. Wegen der bereits erwähnten notwendigen Versorgungsspannung von 3,3V für den XMC4500 und die Transceiver wurde diese mit einem **LDO!** (**LDO!**)-Regler angepasst. Dieser verwendete Spannungsregler der Bauart MCP1826S von Microchip Technology Inc. sollte die Eingangsspannung auf das gewünschte Niveau herunter regeln. Als Gehäusotyp wurde das dreibeinige **SOT!** (**SOT!**)-Package gewählt, da keine Variante des **LDO!** mit einstellbarer Ausgangsspannung und somit keine Variante des **IC!**s mit mehr Anschlusspins benötigt wurde. Statt des **LDO!** von Microchip war zunächst ein gleichwertiger Spannungsregler von Infineon, der IFX1117MEV33, vorgesehen. Die beiden **LDO!**s unterscheiden sich in der elektrischen Belegung der Kühlfahne des **SOT!**-223 beim Spannungsregler von Infineon ist diese mit dem 3,3V Output kontaktiert, beim verwendeten **LDO!** mit Ground. Da es sich bei dem Spannungsregler um ein **SMD!** (**SMD!**)-Bauteil handelt ist die Verwendung von Kühlkörpern schwer möglich. Die Abführung der im Spannungsregler erzeugten Verlustwärme erfolgt deshalb üblicherweise über das Anlöten der Kühlfahne an eine Kupferfläche. So wird die erzeugte Wärme gespreizt und kann gut an die Umgebung abgegeben werden. Bei Verwendung des zuerst eingeplanten IFX1117MEV33 wäre somit eine Kupferfläche auf 3,3V Potential

zum Kühlen notwendig, welche elektrisch isoliert sein müsste. Wegen des XMC4500 und den anderen Bauteilen wäre somit nur eine Kupferfläche mit Abmessungen von etwa 15mm auf 16mm möglich, da die elektrischen Verbindungen bestehender Bauteile des Bereiches nicht unterbrochen werden sollten. Durch die Verwendung des entsprechenden Bauteils von Microchip konnte auf eine abgetrennte Kupferinsel verzichtet werden und somit die bereits erwähnte GND-Kupferfläche um den Mikrocontroller als gemeinsame Masse- und Kühlfäche verwendet werden. Wegen der vorderseitigen Bauteilbestückung war die verfügbare Kupferfläche auf der Platinenrückseite größer. Um dies beim Ableiten der Wärme vom Bauteil zu nutzen wurden vor allem im Bereich um die Kühlfinne des **SOT!**-223 Gehäuses Durchkontaktierungen angebracht. Diese parallelen „Thermal Vias“ konnten als Wärmepfad zur unteren Kupferfläche dienen. Außerdem wurde der **JTAG!**-Stecker des XMC absichtlich im Bereich neben dem Spannungsregler angebracht. Durch die große Oberfläche stellt auch dieser eine gute Wärmesenke dar.

2.4 Generierte Dokumente

Altium Designer kann aus den erstellten **PCB!**-Daten die für die weitere Verarbeitung benötigten Dateien generieren. Im dazu vorgesehene Output-Job-Manager können entsprechende Outputs gewählt werden und einem Output-Container zugeordnet werden. Dies ist vor allem zum Erstellen gewünschter Dateistrukturen bei größeren Projekten notwendig um dies übersichtlich zu halten. Im der vorliegenden Arbeit wurde dies jedoch nur bedingt benötigt. Für die Bestückung der Basisstation wurde zunächst eine „Bill of materials“ also eine Materialliste exportiert mit deren Hilfe die entsprechenden Bestellnummern des Lieferanten Digi-Key herausgesucht und sortiert werden konnten. Mithilfe der ebenfalls generierten „Assembly Drawings“ war Ausgabe aller Bauteile und deren Platzierung auf der Platine möglich. Ebenfalls wurden in diesem Menü die strukturierten Schaltpläne ausgegeben. Zur dreidimensionalen Visualisierung wurde die fertige Platine mit den in den Footprints enthaltenen Bauteilabmessungen und Formen als 3D-Modell im **STEP! (STEP!)**-Format ausgegeben. Mithilfe des **CAD!**-Programms „SolidWorks“ wurde daraus ein Gehäuse mit Deckel für die Basisstation erstellt. Diese wurde anschließend mit einem 3D-Drucker ausgedruckt und dient dazu die Platine zu schützen. Die Fertig der Platine durch den Hersteller erfolgte durch erstellte Gerber Daten, die ebenfalls aus dem Output-Job-Manager generiert werden. Dabei wird für jedes Layer des **PCB!**-Editors eine Gerberdatei erstellt in der die Geometrie der entsprechenden Lage angegeben ist. Jede Lage entspricht in der Herstellung dabei einem Fertigungsschritt. Um Bohrungen in der Platine zu setzen werden zusätzliche „NC Drill-Files“ also Daten für die **NC! (NC!)** der automatischen Maschinen zum setzen von Bohrungen. Diese enthalten den Bohrdurchmesser, die Art der Bohrung sowie den Ort auf der Platine und müssen zusätzlich aus Altium Designer exportiert werden.

Software

3.1 DAVE Entwicklungsumgebung

Das Programm DAVE™ (Digital Application Virtual Engineer) wird von Infineon Technologies AG entwickelt. Sie basiert auf der Entwicklungsumgebung oder **IDE!** (**IDE!**) „eclipse“ die von der Eclipse Foundation entwickelt wird. Eine **IDE!** beschreibt dabei allgemein ein Programm zur Softwareentwicklung, welches die einzelnen dazu notwendigen Tools gesammelt zur Verfügung stellt. Dies sind vor allem der Compiler, der Linker, und der Debugger auf die im folgenden noch eingegangen werden soll DAVE™ greift bei der Programmierung von Mikrocontrollern der XMC-Serie auf die so genannten XMC Libraries zurück die von Infineon ebenfalls zur Verfügung gestellt werden. Auf diese soll ebenfalls im weiteren Verlauf eingegangen werden. Ein weiteres Feature in der **IDE!** sind die sogenannten DAVE™ APPs. Mit diesen soll es die Programmierung des Mikrocontrollers durch ein **GUI!** (**GUI!**) ermöglicht werden. Dazu werden für mögliche von der Hardware zu verrichtende Teilaufgaben APPs von Infineon bereitgestellt. Durch das Einfügen der entsprechenden APPs in das Projekt können diese angepasst und miteinander grafisch verschalten werden. So wird der spätere Programmablauf im Mikrocontroller und dessen Aufgaben festgelegt. Nachdem vom Programmierer nun noch die Pins ebenfalls grafisch den Aufgaben zugeordnet werden, generiert DAVE™ den Programmcode mit den in den Apps enthaltenen Informationen. Mithilfe des DAVE™ **SDK!** (**SDK!**) können nicht nur die Parameter der APPs beim programmieren, sondern auch diese selbst grundlegend angepasst werden und das entwickeln eigener APPs ist möglich. Im Verlauf dieser Arbeit wurden DAVE™ APPs jedoch nur in einem bereits existierenden Softwareprojekt für ein RelaxKit genutzt, mit welchem Signale zum Testen der Empfänger an die Basisstation gesendet wurden. In der Basisstation selbst wurden die APPs jedoch nicht benutzt.

3.2 verwendete Peripherie des XMC4500

Der XMC4500-Baustein enthält diverse funktionelle Blöcke die mit einer Bus Matrix an die ARM Cortex M4 **CPU!** (**CPU!**) angebunden sind. Dieser Aufbau soll den Prozessor

entlasten und im Programmablauf Ressourcen freihalten für andere Operationen. Für die Basisstation waren vor allem der **USIC!** (**USIC!**), der **USB!** (**USB!**) sowie der **ETH!** (**ETH!**) die bedeutende Peripherie. Von besonderer Bedeutung sind jedoch die **GPIO!**s (**GPIO!**s).

3.2.1 GPIO

Die **GPIO!**s werden im so genannten PORTS-Modul der XMC-Architektur gesteuert. In dieser lassen sich die Treiberstufen für die entsprechenden Pins des Mikrocontroller regeln. Dieses Modul ist ebenfalls über die Peripheriebrücke PBA1 an die Bus Matrix und somit den Cortex M4 Kern angebunden.

3.2.2 USIC

3.2.3 ERU

3.2.4 USB

3.3 verwendete Bibliotheken

3.3.1 XMC Libraries (XMC Lib)

3.3.2 SPI Library

3.3.3 TDA5340 Library

3.3.4 Virtueller COM Port

3.4 Programmablauf

3.4.1 Konfiguration der Funkmodule

3.4.2 interruptbasierte Datenerfassung

3.4.3 Weiterleitung der erfassten Daten

wie viel speicher braucht man aaktuell? speichergröße uint immer da 32bit mikrocontroller

Feldtest

4.1 Aufbau

4.2 Durchführung

4.3 Ergebnisse und Auswertung

Zusammenfassung und Ausblick

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

EINFÜGEN: Layout Stefan Erhard Bilder Altium 3D Modelle

6.1 Seriennummern

Alle TDA5340 verfügen über eine eingebaute Seriennummer, welche ausgelesen werden kann. Die Seriennummern der verwendeten TDA5340 sind in der Tabelle aufgeführt.

TDA	Seriennummer
TDA1	33020236
TDA2	11727080
TDA3	11545236
TDA4	11728870
TDA5	11550773
TDA6	33026263

Tab. 6.1: Seriennummern der im Projekt verwendeten TDA5340

6.2 3D-Daten

6.2.1 Gehäuse

Literatur
