

Energy Harvesting Bicycle Computer



**ZÜRCHER HOCHSCHULE
FÜR ANGEWANDTE WISSENSCHAFTEN**

INSTITUTE OF EMBEDDED SYSTEMS

Autoren Katrin Bächli,
 Manuel König

Hauptbetreuer Prof. Dr. Marcel Meli

Nebenbetreuer Herr Dario Dündar

Datum 3. Juni 2016

Inhaltsverzeichnis

1

Einleitung

In einer vernetzten Welt senden Geräte Daten über ihren Zustand oder der, ihrer Umgebung. Diese Technologie wird für ein Fahrradfahrer nutzbar gemacht. Das Handy soll die aktuelle Geschwindigkeit, die Höhe über dem Meer, die Temperatur und die Luftfeuchtigkeit während der Fahrt empfangen.

Diese Idee ist nicht neu. Erhältlich sind batteriebetriebene Modelle, die Daten auf einem Display anzeigen. Das Neue an dieser Arbeit ist, dass die Energie aus der Fahrradumdrehung geerntet wird und dass der User sein eigenes Handy für das Anzeigen der Daten nutzen kann.

1.1 Ausgangslage

Als Inspiration dienten zwei batteriebetriebene Modelle der Hersteller SIGMA SPORT und POLAR. SIGMA SPORT bietet Geräte mit eigenem Display und Sensoren an. Auf dem Display erscheint neben der Geschwindigkeit, die Daten der Sensoren, die GPS-Ortung und den aktuellen Ladestand der Batterie. Der Hersteller POLAR stellt ein Gerät her, welches die Fahrt über GPS aufzeichnet und wichtige Informationen zur Trainingsverbesserung liefert. Als Nachteil bewerteten wir, dass ein (verdrahtetes) Display gebraucht wird.

Während der Arbeit wurden wir auf den Hersteller Reelight aufmerksam. Reelight gewinnt über Wirbelströme Energie und schafft es bei seinem Produkt City Supreme genügend Energie für eine LED-Lampe zu erzeugen. Da auf der Webseite keine Dokumentation des Funktionsprinzips erhältlich ist, ergaben eigene Untersuchungen, dass sich im Innern der Lampe „etwas“ bewegt. Unserer Meinung nach ist dies ein Magnet, der so gelagert ist, dass er sich drehen kann. Der an der Felge vorbeiziehende Magnet erzeugt ein Wirbelstrom auf der Felge. Dieser wirkt auf den Magneten im Licht und der Magnet im Licht beginnt sich zu drehen. Befindet sich neben dem sich drehen-

den Magneten eine Spule, so wird genügend Spannung für das Betreiben einer LED induziert. Diese Harvesting-Methode beurteilen wir als interessant für eine zukünftige Arbeit. Der Nachteil dieser Methode ist, dass das Erzeugen eines Wirbelstroms auf den Felgen bremsend wirkt. Bei dieser Bachelorarbeit war die Harvesting-Methode bereits vorgegeben, da sie auf einer vorangehenden Projektarbeit basiert.

Als Grundlage dient der Aufbau aus der Machbarkeitsstudie „Bicycle computer and sensoric powered with harvested energy“ (?). In dieser Projektarbeit wurde der Beweis erbracht, dass durch Bewegungsinduktion genug Energie erzeugt werden kann, um die Geschwindigkeit des Fahrrads per Bluetooth Low Energy zu übermitteln. Der Aufbau funktioniert nach vorangehendem Laden der Kondensatoren zuverlässig bei 20 km/h. Das Ziel dieser Arbeit besteht aus einer verbesserten Energiegewinnung, einem besseren Verbrauchsmanagement und einer ansprechenden Applikation. Konkret soll ein attraktives Produkt ohne Aufladen der Kondensatoren für eine Geschwindigkeit bei 10 km/h entstehen. Diesen Prototypen wird in dieser Arbeit kurz Bicycle Computer genannt.

1.2 Definition der Aufgabenstellung

Durch die offiziellen Ausschreibung der Bachelorarbeit an der ZHAW ist der Inhalt der Bachelorarbeit vorgegeben (siehe Anhang ??). Das Ziel der Arbeit ist, aus dem Aufbau einer Machbarkeitsstudie einem Prototypen eines batterielosen Fahrradcomputers zu entwickeln. In den ersten Sitzungen zusammen mit Prof. Dr. Marcel Meli und Herr Dario Dündar wurde die Aufgabenstellung auf folgende Punkte konkretisiert:

1. Inbetriebnahme des Prototypen, Einlesen in die vorangegangene Projektarbeit und Beschäftigung mit der Materie, sind die Hauptpunkte des ersten Schrittes.
2. Die bestehende Hardware muss verkleinert und überarbeitet werden. Dafür wird ein neues PCB entworfen, welches verschiedene vorhandene Platinen vereint.
3. Initialisierung der Bluetooth-Schnittstelle muss auf dem Android-Endgerät und der Hardware vorgenommen werden. Eine erste Bluetooth-Kommunikation zwischen der Hardware und der Applikationen ist implementiert.
4. Das bestehende Energiemanagement soll auf die Anwendung eines Fahrradcomputers optimiert werden.
5. Die Benutzeroberfläche der Android-Applikation soll benutzerfreundlich und optisch ansprechend gestaltet werden.
6. Die erfassten Messwerte der Geschwindigkeit und der aktuellen Höhe sollen über Bluetooth übermittelt werden.
7. Die erfassten Daten sollen gespeichert und nur dann übertragen werden, wenn die nötige Energie vorhanden ist.

8. Per GPS soll die aktuelle Position ermittelt, sowie die bereits abgefahrene Route erfasst werden. Alles soll auf einer Karte veranschaulicht werden.
9. Die Beschleunigung, Luftfeuchtigkeit und Temperatur sollen ebenfalls erfasst und über Bluetooth übermittelt werden.
10. Das Energiemanagement soll für verschiedene Geschwindigkeiten optimiert werden.

Bei der Festlegung der Arbeitsschritte half die Vision eines innovativen Fahrradcomputers. Die Miniaturisierung (Punkt 2) ist für die Attraktivität des Produkts entscheidend. Die Möglichkeit, auf dem eigenen Handy die Daten zu sehen, beurteilen wir als attraktiv. Weil das Handy wichtig ist, wurde Punkt 5. zu den wesentlichen Aufgaben genommen. Die App soll ansprechend und einfach für die Benutzerin oder den Benutzer sein. Das verbesserte Energy Management (Punkt 4) hat mit dem Ziel zu tun, dass der Fahrradcomputer bereits bei 10 km/h die Geschwindigkeit ausgeben soll (Im Kapitel Inbetriebnahme wird beschrieben, dass das Vorgängermodell erst ab 20 km/h funktioniert). So gelten für die Bachelorarbeit die Punkte 1.) bis 6.) als Minimalanforderungen, während die Punkte 6.) bis 10) dynamisch und in Abhängigkeit des Projektfortschritts einbauen lassen. Aus den definierten Anforderungen entstand der auf der CD abgelegte Projektplan.

1.3 Übersicht der Aufgabenblöcke

Um den Überblick der zu erledigenden Punkte zu behalten, werden die Aufgaben in Arbeitsblöcke (siehe Abbildung ??) geteilt. Die gepunkteten Blöcke sind optional, die voll umrandeten entsprechen dem Minimum. Die Projektplanung wurde so aufgebaut, dass bei Meilenstein 1, das Layout gezeichnet ist, bei Meilenstein 2 die Kommunikation zur App besteht, bei Meilenstein 3 die überarbeitete Version des Prototyps gezeigt wird und bis dahin das Minimum erreicht ist. Welche optionalen Ziele realisiert werden, wird im Meilenstein 3 definiert.

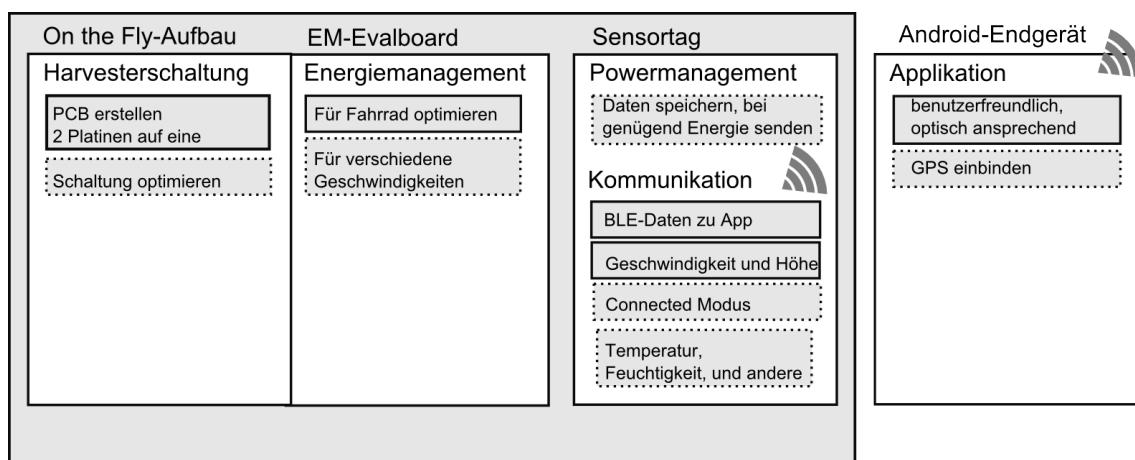


Abbildung 1.1: Arbeitsblöcke

2

Theoretische Grundlagen

Der Bicycle Computer basiert auf Energy Harvesting. Was Energie ernten bedeutet und welche Art von Energy Harvesting in dieser Arbeit angewandt wird, wird im ersten Unterkapitel ?? beschrieben. Im folgenden Unterkapitel ?? geht es um Ansätze zum Sammeln und Weiterleiten von Energie. Da die in Unterkapitel ?? gewonnene Energie im μW -Bereich liegt, ist zuerst ein Sammeln der Energie notwendig, sodass Leistungen im mW -Bereich zur Verfügung stehen. Das nächste Unterkapitel ?? befasst sich mit dem notwendigen Power Management. Denn die Energie soll nicht sofort verbraucht werden. Power Management regelt, wie schnell und wie viel Energie aufs Mal verbraucht werden soll. Als letzte Stufe in der Umsetzung ist eine energiearme Kommunikation notwendig. Da bietet die Bluetooth Low Energie Technologie ideale Voraussetzungen. Das Protokoll und die Technologie werden im letzten Grundlagenteil ?? vorgestellt.

2.1 Energy Harvesting

„Mit Energy Harvesting ... wird die Gewinnung von elektrischer Energie in kleinen Mengen aus dem Umfeld elektronischer Geräte für deren Betrieb bezeichnet.“?. Als erstes werden Methoden zur Energiegewinnung vorgestellt (??) und danach die im Bicycle Computer verwendete Harvesting Art genauer beschrieben (??). Als letztes wird der Unterschied zwischen den Harvestingmethoden festgehalten. Denn diese Unterschiede werden in der Implementation des Bicycle Computers wichtig.

Abstand zu
klein

2.1.1 Energy Harvesting Methoden

Bekannte Methoden sind die Solarzelle, die aus der Energie der Sonnenstrahlen Strom erzeugt, die Thermogeneratoren (TEG), die aus Umgebungswärme Energie gewinnen, passive RFID-Tags, die aus der elektromagnetischen Strahlung Energie gewinnen und

der piezoelektrische Effekt, der mechanischen Druck in elektrische Spannung umwandelt. Da der im Prototyp verwendete Energy Management-Chip ? für die Energieoptimierung von Solarzellen oder von Thermogeneratoren spezialisiert ist, werden diese zwei Methoden vorgestellt.

a. Energy Harvesting mit einer Solarzelle

Bei der Umwandlung von Elektromagnetischen Wellen (Licht) in Strom wird eine spezielle Eigenschaft des Siliziums genutzt: Führt man Silizium Energie zu, entstehen freie Ladungsträger, bzw. Elektronen und Löcher. Um aus diesen Ladungen einen elektrischen Strom zu erzeugen, ist es nötig, die erzeugten freien Ladungsträger in unterschiedliche Richtungen zu lenken; dies geschieht durch ein internes elektrisches Feld, welches durch einen p-n-Übergang erzeugt werden kann. Auf der einen Seite sammelt sich positive, auf der anderen Seite negative Ladung an. Werden diese verbunden, entsteht ein Strom (?). Diese Harvestermethode produziert ein Gleichstrom. Größen- und materialabhängig kann Energie im kW-Bereich gesammelt werden.

b. Energy Harvesting mit einem TEG

TEG steht für Thermoelectric Generator und bezeichnet eine Konstruktion, die aus einem Temperaturunterschied elektrische Spannung erzeugt. Erzeugt wird die Spannung am Ende zweier metallischer Leiter aus unterschiedlichem Material, die an einem Ende verbunden sind (?). Diese Harvestermethode produziert eine Gleichspannung. Die produzierte Spannung ist vergleichsweise klein und bewegt sich im Bereich einiger $10 \mu\text{V}$ pro 1°C Temperaturdifferenz.

2.1.2 Energy Harvesting über Bewegungsinduktion

Beim Bicycle Computer wird Energie über Bewegungsinduktion gewonnen. Die Funktionsweise ist in der Machbarkeitsstudie beschrieben ? S.8.:

Befindet sich eine Spule in einem *dynamischen „Magnetfeld“*, wird in der Spule eine Spannung induziert. Dies sieht man in der Formel (??).

$$U_{ind} = -\frac{d}{dt} \int A dB \quad (2.1)$$

Der magnetische Fluss B durch die Fläche einer Spule A ist gleich dem magnetischen Fluss ϕ . Hat die Spule mehrere Wicklungen N , so verstärkt sich der magnetische Fluss proportional.

$$\frac{d}{dt} \int A dB = \phi \cdot N \quad (2.2)$$

Verläuft der magnetische Fluss ϕ senkrecht zur Fläche der Spule A kann das Integral durch eine Multiplikation ersetzt werden (siehe Formel).

$$\frac{d}{dt} \int A \perp dB = \frac{d}{dt} \int \phi \cdot N = B \cdot A \cdot N \quad (2.3)$$

In diesem Fall berechnet sich die induzierte Spannung in einer Spule vereinfacht mit

$$U_{ind} = -N \cdot A \cdot B \quad (2.4)$$

Das dynamische Magnetfeld wird durch das Bewegen, oder im Fall eines Fahrrads einem Vorbeiziehen, eines Magneten an einer fix verankerten Spule erzeugt. Die produzierte Spannung hängt von folgenden vier Faktoren ab:

1. die eingeschlossene Fläche A der Spule
2. die magnetische Flussdichte des Magneten B
3. die Anzahl Windungen N der Spule und
4. die Bewegungsgeschwindigkeit v des Magneten, welche Einfluss auf dt hat

Diese Harvestermethode produziert einen Wechselstrom. Ein Gleichrichter und einen Kondensator zur Glättung der Rippelspannung ist nach der Energiegewinnung notwendig. Die Leistung der produzierten Spannung geht vom μW -Bereich bis zu für die Industrie optimierten Anlagen mit Leistung MW-Bereich wie z. B. durch Drehstrom-Generatoren.

2.1.3 Unterschiede Harvestermethoden

Unterschiede bestehen in der Art, in der die Energie zur Verfügung steht. Die nachfolgenden zwei Unterpunkte zeigen, dass zwei Differenzen auf: Punkt a verweist auf die unterschiedliche Art, in der die Energie erhältlich ist: entspricht dies einer gleichmässigen Energie, analog zum DC, oder liefert die Quelle Energie in Form von Wechselstrom? Folgt der Strom konstant oder in Pulsen? Die aufwendige Art der Energieerntung wird für den Prototypen relevant. Der zweite Unterschied unter den Harvestermethoden ist die Leistungskurve. An welcher Stelle zwischen Kurzschluss und liegt das Leistungsmaximum? Die unterschiedlichen Leistungsmaxima der einzelnen Harvestermethoden werden unter Punkt b zusammengefasst. Beide Unterschiede werden für die Entwicklung des Prototypen über Bewegungsinduktion relevant.

a. Gleichmässige Energie versus gepulster Energie

Messprotokolle

Die Solarzelle und ein TEG liefern Gleichstrom bzw. -spannung (siehe Abbildung ??). Dadurch ist keine Gleichrichterschaltung und Glättung notwendig sind. Die durch Bewegungsinduktion gewonnene Energie ist eine Wechselspannung (siehe Abbildung ??, aus Messprotokoll xxxx). Im Fall des Bicycle Computers ist diese gleichzeitig gepulst. Die Energie ist somit nicht konstant da, sondern nur in Zeitintervallen. Dadurch ergeben sich zwei Probleme bei der Energiegewinnung: Durch den Gleichrichter geht Energie verloren und durch die Pulsform entsteht, trotz Signalglättung über einen Kondensator, eine Rippelspannung (siehe Abbildung ??, aus Messprotokoll xxxx).

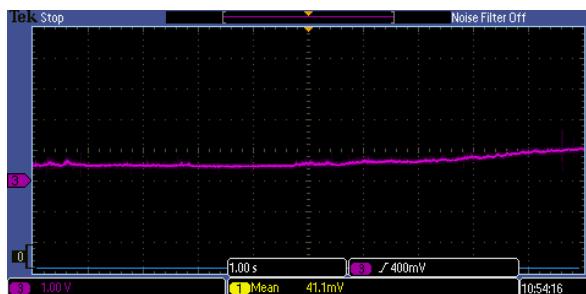


Abbildung 2.1: Gleichspannung am Ausgang eines TEG

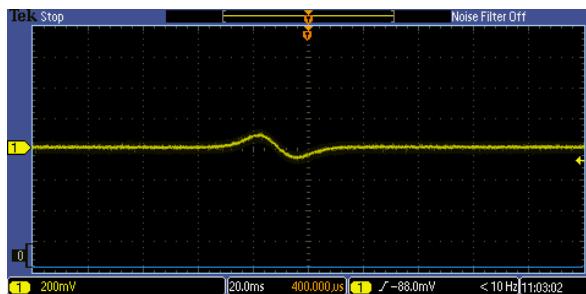


Abbildung 2.2: Wechselspannung als Energieform der Bewegungsinduktion

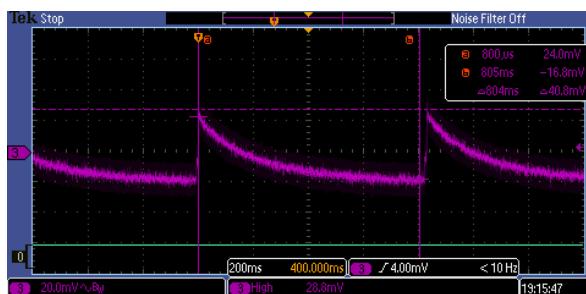


Abbildung 2.3: Rippelspannung aufgrund der gepulsten Eingangsenergie

b. Konstanter Maximum Power Point zu dynamischem

Die drei Harvester unterscheiden sich in ihrer Leistungskurve. Das Leistungsmaximum, der Maximum Power Point (MPP), liegt auf der Skala von Kurzschluss bis Leerlauf proportional an unterschiedlichen Stellen. Bei einem TEG liegt das MPP in der Mitte dieser Skala. Die MPPT-Ratio beträgt 50 %. Bei der Solarzelle liegt das Leistungsmaximum auf der Skala bei ca. 80 % der maximalen Spannung. Die MPP-Ratio ist 80 %. Bei der Bewegungsinduktion existiert kein fixe MPP-Ratio. Wie bei der Spule, wandert das Leistungsmaximum aufgrund mehrerer Indikatoren (wie Geschwindigkeit des Magneten durch die Spule, Abstand von Magnet und Spule) auf der Skala hin und her.

Zur Verdeutlichung der Unterschiede sind für jedes Leistungsverhalten eine Abbildung angefügt. Das TEG hat unabhängig von der gewonnenen Energie und der Temperatur das Leistungsmaximum immer bei 50 %. Die Abbildung ?? zeigt, dieses unabhängige Verhalten.

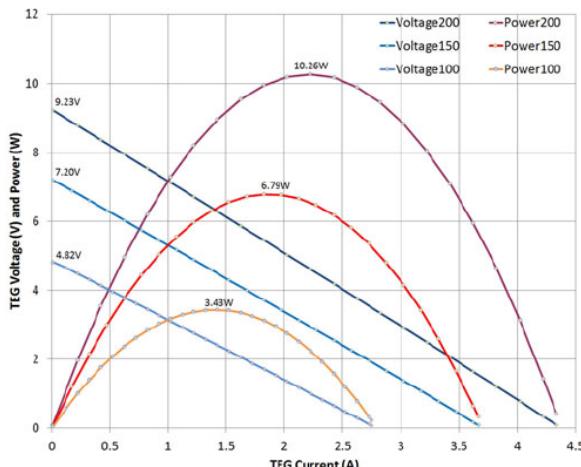


Abbildung 2.4: MPP TEG (?)

Abbildung ?? zeigt, dass das Leistungsmaximum bei der Solarzelle unabhängig von der zur Verfügung stehenden Energie immer bei 80 % liegt.

Die Stelle des Leistungsmaximums wandert bei einer Spule und somit bei der Bewegungsinduktion auf der Skala. Exemplarisch sind drei MPPT-Ratios einer Spule in der Abbildung ?? abgebildet. In dieser Abbildung zeigt sich der Einfluss des Abstands der Spule vom Magnetfeld auf die Stelle der maximalen Leistung. Diese Abbildung wurde ausgewählt, weil beim Ausmessen des Harvesters der Abstand des Magneten als einer der Einflüsse festgestellt wurde. Im Kapitel ?? Resultat sind die ausgemessenen Leistungskurven des Prototypen in der Graphik ?? abgebildet. Die Leistungskurven wurden für verschiedene Geschwindigkeiten aufgenommen und es zeigte sich, dass bei der Harvesterschaltung analog zur Spule, die MPPT-Ratio wandert. Für den theoretischen Teil wurde die MPPT-Kurven des Harvesters stark geglättet und vereinfacht (siehe unten).

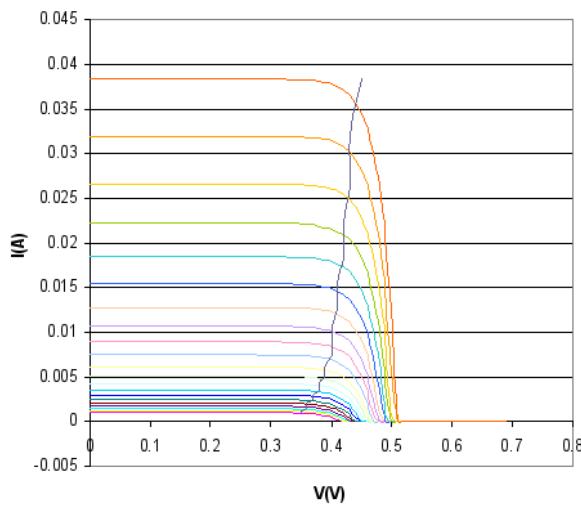


Abbildung 2.5: MPP Solarzelle (?)

stehende Abbildung ??). So ist der Effekt der Verschiebung des Leistungsmaximums besser ersichtlich wird. Es lässt sich grob über die MPPT-Ratio des Bicycle Computers sagen, dass sie sich zwischen 35 - 75 % bewegt.

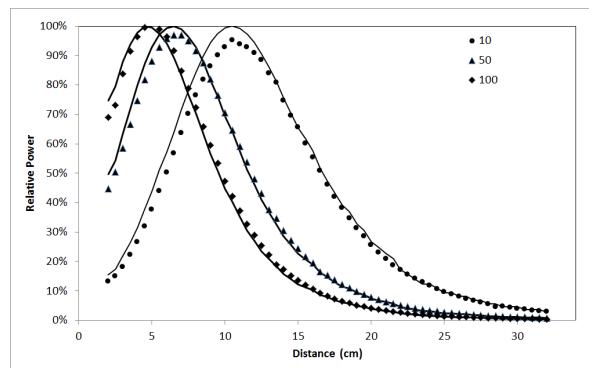


Abbildung 2.6: MPP Spule (?)

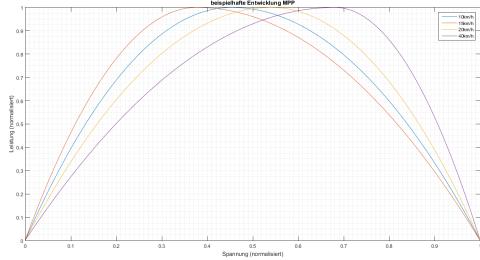


Abbildung 2.7: MPP Harvester (?)

2.2 Energy Management

Der Harvester des Bicycle Computer erntet eine gepulste Energie im μJ -Bereich. Um diese für eine Applikation zu verwenden, müssen die geringen Energieportionen summiert werden. Sind Energimengen im mJ -Bereich verfügbar, kann die Energie kontrolliert freigegeben werden.

Energy Management bezeichnet das der Energie in Speichern, das Regeln des Inputs, damit die maximale Leistung aus der Quelle bezogen werden kann, das Aufwärtswandeln von Spannung oder Strom auf den geforderten Wert und die kontrollierte Freigabe.

Überprüfen
mJ notwendig
sind, oder ob
 μJ genügen

In der Bachelorarbeit ist das Verwenden des Chip EM8500 vorgegeben. Der EM8500 ist ein Power Management Controller für den Low Power-Bereich. Das Datenblatt des EM8500 ist der CD beigelegt. Als erstes wird das kontrollierte Energiespeichern anhand dieses Chips erklärt. Danach folgt die Umsetzung des Maximum Power Point Trackings (MPPT) und eine kurze Erklärung der Wirkung des Boosters auf das Energy Managements. Zuletzt wird auf das freischalten von Ausgängen eingegangen, da dies für das Verwenden der Energie die wesentliche Schnittstelle ist.

2.2.1 Kontrollierte Energiespeicherung

Bei einer Low Power Harvesting Applikation ist wesentlich, dass vor der Verwendung der Energie durch einen Microkontroller, genug Energie in Speichern gesammelt wurde. Dies ist in der Abbildung ?? dargestellt. Sie zeigt, dass für die Freigabe der Energie an eine Applikation, das Signal der Speisung der Applikation wird als VSUP bezeichnet, erst nach dem Erreichen eines gewissen Speicherzustands erfolgt. Die Höhe des Speicherwertes kann im Register `v_bat_min_low` eingetragen werden. Die Ladespannung des sogenannten Primärspeichers, dem Short Time Storage (STS), ist mit VSTS in der Abbildung ?? abgebildet.

Im EM8500 wird dies folgendermassen umgesetzt (Abbildung ??): Erreicht der Primärspeichers STS den Schwellwert `v_bat_min_low`, wird VSUP mit der eingestellten Spannung gespiesen. Die Applikation sollte nicht alle Energie verbrauchen, sodass sich der Speicher



Abbildung 2.8: Grundprinzip Applikationsspeisung

weiter lädt. VSUP folgt der Speicherspannung. Verbraucht die Applikation viel Energie, fallen VSUP und VSTS parallel. Speiste der Harvester viel Energie, steigt bei beiden die Spannung an. Unterschreitet VSTS/VSUP den Schwellwert von $v_{bat_min_low}$, so wird die Speisung der Applikation gestoppt.

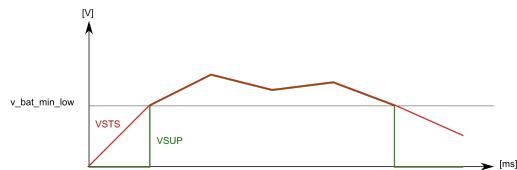


Abbildung 2.9: Applikationsspeisung EM8500

Der Primärspeicher STS ist für die kurzfristige Speisung der Applikation verantwortlich. So bedeutet STS Short Time Storage. Für das langfristige, sichere Ausführen braucht das System ein Long Term Storage (LTS). Seine Aufgabe ist, Reserveenergie aufzubauen. Diese überbrückt die Energieengpässe, wenn der Harvester zu wenig Energie liefert. VLTS wird geladen, wenn der Schwellwert bei $v_{appl_max_los}$ ist (siehe Abbildung ??).

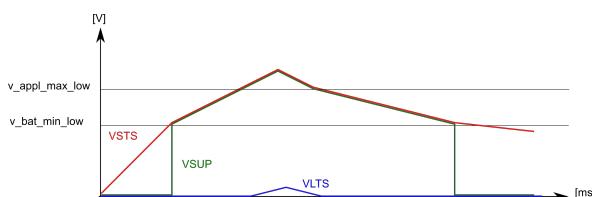


Abbildung 2.10: Sichereres Betreiben durch Long Term Storage

Im Datenblatt des EM8500 (siehe ?) sind weitere Feineinstellungen beschrieben und drei Application Notes helfen bei der Berechnung der Schwellwerte für ein sicheres betreiben. Die Dateien sind auf der CD abgelegt.

Grundsätzlich ist zur Berechnung der Speicher und den Schwellwerten zu sagen, dass der erste Schwellwert ($v_{bat_min_lo}$), bei dem die Speisung der Applikation beginnt,

genug Energie für die Initialisierung der Applikation gesammelt haben muss. Zudem muss das Abschalten von VSUP vermieden werden. Denn ein Neustart braucht aufgrund der Initialisierung viel Energie und ist ein unnötiger Kraftakt in einem Low Power System. In den Beispielkonfigurationen des Herstellers (? S. 5 - 8) sieht man, dass in deren Überlegungen VSUP nicht abgeschaltet wird. Der Hersteller geht davon aus, dass sogar bei dem Freischalten von VSUP die Spannung am STS nicht aufgrund der Last der Applikation fällt, sondern sich weiterhin auflädt.

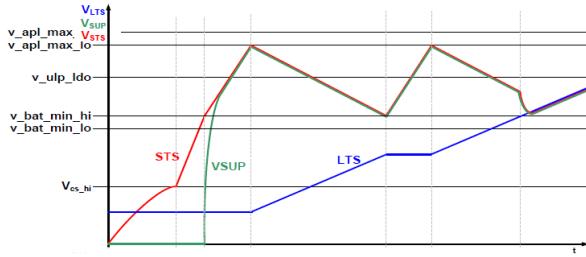


Abbildung 2.11: Konzept Hersteller (?)

2.2.2 Regelung des optimalen Leistungsbezugs

Wichtigster Punkt in der Energieoptimierung ist, das Maximum aus der produzierten Energie weiterzuverwenden. Aus diesem Grund wird vor Inbetriebnahme eine Leistungskurve des Harvesters erstellt. Wie in Unterkapitel ?? beschrieben, unterscheidet sich der Maximum Power Point (MPP) unter den Harvestern stark.

EM8500 versucht die Quelle stets in der Nähe dieses Optimums zu betreiben. Dies geschieht über eine Innenwiderstand-Regelung, sodass die Eingangsleistung möglichst dem MPP entspricht. Wie schnell die aktuelle Leistung überprüft wird, ist einstellbar. Der EM8500 besitzt eine Auflösung von 37 mV. Die Abbildung ?? zeigt das periodische Messen des Spannungswertes des Harvesters. Da die Kurzschlussmessung für das Messen des Stromwerts eine Spannungsspitzen verursacht, (gut zu sehen in der Abbildung bei höherer Spannung), sollte die Leistungsüberprüfung nicht zu oft geschehen. Spannungsspitzen bedeuten Energieverlust, weshalb die Energiemessung nur so oft als absolut notwendig erfolgen soll. In der Abbildung ?? beträgt die Periode 8 s.

2.2.3 Booster definiert Spannung

Direkt mit dem MPPT-Kontroller ist der Booster (siehe Blockdiagramm im Anhang ??). Die Aufgabe des Boosters ist es, das interne Spannungsniveau (VREG) zu heben. Der Booster arbeitet ab einer Eingangsspannung von 0.3 V. Danach regelt er in Schritten von 0.3 V. Der Booster geht von einer Gleichspannung eines TEG oder

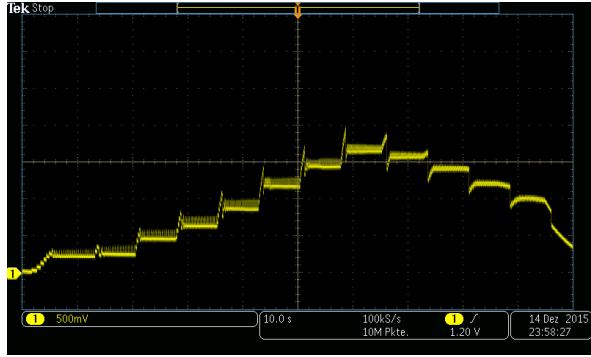


Abbildung 2.12: Leistungsmessung des Harvesters

einer Solarzelle aus. Für diese zwei Anwendungen ist der EM8500 konzipiert (siehe ?, Abschnitt Description, S.1).

2.2.4 Energiezustand kennen und In- und Ausgänge schalten

Da der Primärspeicher STS vom Boosterausgang $V_{Booster}$ gespiesen wird, entspricht dessen Spannung dem des Boosterausgangs. $E_{Applikation}$ bezeichnet die minimale Energie, die die Applikation braucht, also mindestens die Initialisierung der Applikation.

$$E_{Applikation} = C_{STS} \times \frac{1}{2} (V_{Booster})^2 \quad (2.5)$$

Da sich der Kondensator C_{STS} bei der Freigabe der Energie über V_SUP nicht auf 0 entlädt, sondern die minimale Applikationsspannung erhalten bleibt, gilt für die Berechnung des Kondensators C_{STS} nur die Energiedifferenz als verwendbar. Es ist die Differenz zwischen den zwei eingestellten Primärspeicher-Schwellwerten v_bat_min_hi_dis (ab da wird V_SUP gespiesen) und v_bat_min_low (ab da hört die Speisung von V_SUP auf). Die zwei Schwellwerte werden in der Gleichung vereinfacht als STS_HIGH und STS_LOW bezeichnet.

$$E_{Applikation} = E_{STS_HIGH} - E_{STS_LOW} \quad (2.6)$$

$$E_{Applikation} = C_{STS} \times \frac{1}{2} (V_{STS_HIGH})^2 - C_{STS} \times \frac{1}{2} (V_{STS_LOW})^2 \quad (2.7)$$

Kennt man die notwendige Energie, so kann die Grösse des Kondensator C_{STS} aus der notwendigen Energie berechnet werden:

$$C_{STS} = \frac{E_{Applikation}}{(V_{STS_HIGH})^2 - (V_{STS_LOW})^2} \quad (2.8)$$

Umgekehrt argumentiert, kann man aus der Höhe der Applikationsspannung (V_{SUP} bzw. V_{STS_LOW}), des bekannten Energiekonsums $E_{Applikation}$ den notwendigen Schwellwert für den Primärspeicher berechnen:

$$(V_{STS_HIGH})^2 - (V_{STS_LOW})^2 = \frac{2 \times E_{Applikation}}{C_{STS}} \quad (2.9)$$

$$(v_bat_min_low)^2 = \frac{2 \times E_{Applikation}}{C_{STS}} + (V_{STS_LOW})^2 \quad (2.10)$$

Neben VSUP kann der EM8500 drei weitere Ausgänge freischalten: VAUX[0] bis VAUX[2] (siehe Abbildung ??). Vor allem aber kann per I2C oder SPI der aktuelle Spannungspegel der Regelung (VREG), die Speicher (VDD_STS und VDD_LTS) und des Harvestereingangs (VDD_HRV) abgefragt werden. So kennt die Applikation jederzeit den aktuellen Energiezustand der gesammelten Energie.

EM8500 stellt zwei digitale Überwachungssignale zur Verfügung:
Der Ausgang HRV_LOW ist auf logisch '0', wenn die Eingangsspannung vom Harvester grösser als 0.3 V ist. Fällt diese darunter, geht HRV_LOW auf logisch '1'. Der Ausgang BAT_LOW zeigt die Zeitdauer an, in der nur STS die Applikation speist:

- BAT_LOW = '0'
Nicht genügend Energie zur Speisung der Applikation. VSUP ist ausgeschaltet.
- BAT_LOW = '1'
Genügend Energie zur Speisung der Applikation.
VSUP ist eingeschaltet.
- BAT_LOW = '0'
Genügend Energie zur Speisung von LTS.
VSUP ist eingeschaltet.
Der Zustand entspricht nicht mehr BAT_LOW.

Mit den zwei digitalen Signalen kann der Energiezustand grob abgebildet werden.

2.3 Power Management

Die Aufgabe des EM8500-Chip ist es, Energie zu sammeln und kontrolliert frei zu geben. Die Aufgabe des nachfolgenden Microkontrollers ist es, die freigegebenen Energiepor-

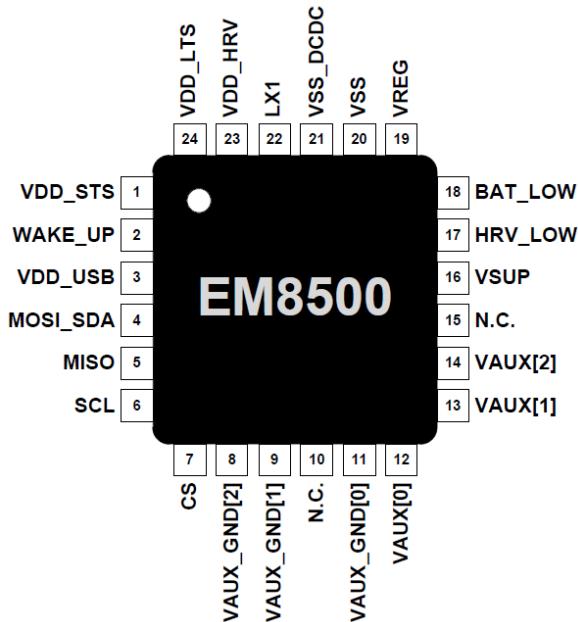


Abbildung 2.13: In- und Outputs EM8500 (?, p.11)

tionen optimal zu verwenden. Das bedeutet, möglichst wenig Energie bei der Datenverarbeitung zu benötigen. Dies wird durch Abstellen aller unnötigen Microkontroller-Bereiche erreicht und einem zusätzlichen Schlafen während allen Warteprozessen.

In diesem Kapitel werden drei Konzepte zum Umsetzen eines Low Power Systems vorgestellt. Das Hauptthema ist das Schlafen zwischen allen Prozessen. Dies wird im ersten Unterkapitel beschrieben. Das Schlafen bedingt ein Aufwecken aufgrund von Ereignissen. Dadurch ergibt sich eine Interrupt Driven Applikation. Diese wird im zweiten Unterkapitel erklärt. Als letztes dient ein Design Aspekt: Durch das Einbauen einer State Machine über alle laufenden Interrupts, ist es nachfolgenden Entwicklerinnen und Entwicklern einfacher, den Code und die gegenseitigen Beeinflussungen zu verstehen. Dies wird im Unterkapitel beschrieben.

Vor der technischen Beschreibung der Konzepte in den drei Unterkapiteln wird kurz auf die verwendete Hardware eingegangen. In der Bachelorarbeit war als Microkontroller das Simple Link Sensortag von Texas Instrument vorgegeben. Der Grund für dieses Board ist, dass das Sensortag drei Anforderungen auf einem Board vereint:

- Ein Cortex M3 dient als Haupt-Microkontroller und ist aufgrund seiner hohen, und somit schnellen, Rechenleistung und seiner Low Power-Fähigkeiten für eine Harvester-Anwendung wie der Bicycle Computer geeignet.
- Auf dem Board ist ein zweiter Cortex M0 für die Wireless-Anbindung angegeschlossen. Dieser Prozessor kann von der Benutzerin oder dem Benutzer nicht programmiert werden, da die Schnittstelle zur Low Power Datenkommunikation fix aufgesetzt ist. Neben Bluetooth Low Energy kann auch Zigbee verwendet werden.

- Auf dem Board sind 10 Sensoren angebunden.

Die Funktionsblöcke des Sensortags befinden sich im Anhang ??.

2.3.1 Einbauen von Schlaufmodi

(Low Power Microcontroller können Gebiete des Prozessors oder von Peripherieelementen temporär ausschalten. Das System befindet sich im Standby Modus. Nur die für die Applikation unabdingbaren Aktivitäten laufen mit niederstem Takt weiter. Über Interrupts können einzelne Bereich aufgeweckt werden, die ihre Aktionen ausführen und danach geht das System wieder in den Standby Modus.)

Dass Prozessoren nach längerer Zeit ohne externen Input in den Schlafmodus gehen, ist Usus (Allgemeingut, bekannt). Bei einer Low Power Applikation geht der Prozessor jedoch nach kleinsten Ausführungsblöcken direkt wieder schlafen. So gehört zu jedem Aufwecken einer Peripherie, der Parallel-Schlafmodus, bis dass die Peripherie gestartet ist. (Dies gilt auch für die Sensoren.) In diesem Unterkapitel werden zwei Umsetzungen des Sleep-Moduses konzeptionell erklärt.

a. Schlafen zwischen Ausführungen

Die Abbildung ?? zeigt das Grundprinzip. Das Programm besteht aus verschiedenen Aktionsblöcken. Diese dauern unterschiedlich lang und verbrauchen unterschiedlich viel Energie. Zwischen den Aktionen ist eine frei wählbare Wartezeit einbaubar ($\Delta t_0 - t_3$). Die graue Markierung in jedem Aktionsblock sind die Initialisierungen vor jeder Aktion.

Während der Schlafenszeit sind alle Peripherien abgeschaltet und im Prozessor (hier als Bsp. Cortex M3) wird nur die Konfigurationen im Flash regelmässig „refreshed“. (Zu den unabdingbaren Aktivitäten eines laufenden Microkontrollers gehört das Refreshen (Neuladen) der Register mit den Systemeinstellungen. Diese Refreshing-Peaks sieht man im Standby Modus.) Dies ist in der Abbildung ?? an den grünen Spannungsspitzen zu sehen. Ohne „refreshen“ des Speichers, gehen die Konfigurationen verloren und vor jeder Aktion muss das System neu komplett initialisiert werden.

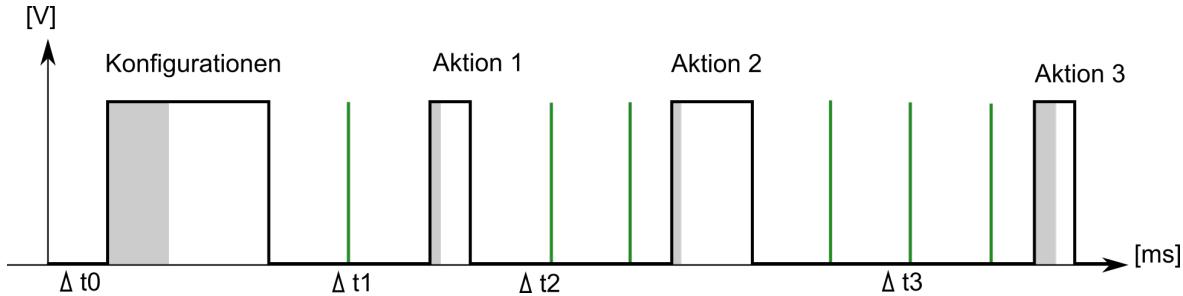


Abbildung 2.14: Schlafen zwischen Ausführungen

b. Schlafen innerhalb einer Aktion

Bei einer Applikation im μ - oder mW-Bereich wird bei jedem Warteprozess, wie z.B. die Zeit, die der Sensor zum Aufwachen braucht, in den Sleep-Mode gegangen. Innerhalb des Codes dominieren die Aufwach- und Abstelleinstellungen. Für jede Aktion, wird nur die PowerDomain dieser Funktionalität eingeschaltet und nach ausführen der Aktion wieder abgeschaltet. Die Abbildung ?? zeigt dieses Prinzip mit UML dargestellt.

2.3.2 Interrupt Driven Application

Zum Schließen gehört auch ein Aufwachen. Dies ist ein nicht-triviales Problem, lässt sich mit Interrupts lösen. In diesem Unterkapitel werden zwei Konzepte einer Interrupt Driven Applikation erklärt: fixes Aufwachen aufgrund interner Interrupts und asynchrones Aufwachen aufgrund externer Events.

a. Aufwachen durch interne Interrupts

Ein System kann intern seine Signale auswerten und aufgrund kombinatorischen Logik, dem erreichen eines Schwellwerten oder dem Ablauen eines Timers aufwachen. Solche Wakeups sind fix und unabhängig von äusseren Einflüssen. Ein System mit internen Interrupts ist determinierbar. Das heisst, der Empfang interner Interrupts ist präzis (nur 1 malig und kein Prellen) und kann über Prioritäten gut geregelt werden.

b. Aufwachen durch externe Events

Der Prozessor, oder Teile davon, können auch aufgrund äusserer Impulse aufwachen. Die Verarbeitung des Interrupts ist dieselbe, nur weiss man nicht, wann das Ereignis auftritt. Die Gefahr, dass zwei Interrupts zur selben Zeit eintreffen oder eine Quelle mehrere Interrupts sendet, ist gegeben. Das Löschen der eingegangenen Interrupts und das Prüfen, ob ein Event nicht zu oft verarbeitet wird, muss bewerkstelligt werden.

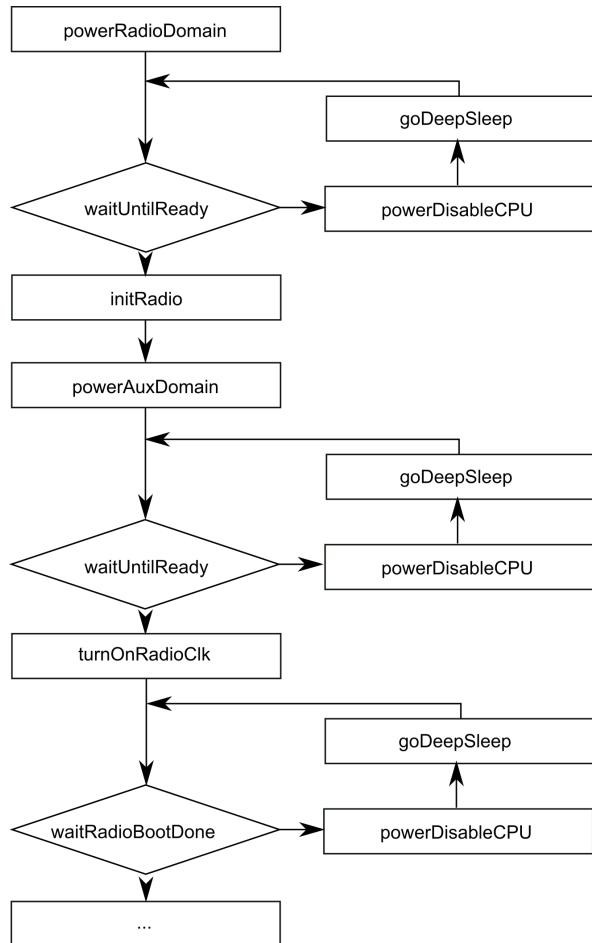


Abbildung 2.15: Schlafen innerhalb des Codes

Löst eine Quelle Interrupts über längere Zeit aus, kann dies das System absorbieren und schlimmstenfalls den Systemablauf aus dem Rhythmus bringen.

2.3.3 State Machine für definierte Abläufe

Low Power Applikationen enthalten viele Interrupts. Jede grössere Tätigkeit braucht das Aktivieren mehrerer Schnittstellen, die alle aufgeweckt und aufeinander abgestimmt werden. Da die Codeausführung nicht sequentiell verläuft, sondern Ausführungen in den Interrupt-Handlern stehen, ist ein Überblick der Abhängigkeiten nicht einfach ersichtlich. Um unerwünschte Effekte zu vermeiden, kann eine State Machine implementiert werden. Diese stellt sicher, dass nur aufgrund von gewissen Signalen, eine Aktion ausgeführt wird. Alle anderen Effekte werden ignoriert. Die Abbildung ?? zeigt die definierten Abhängigkeiten durch eine State Machine.

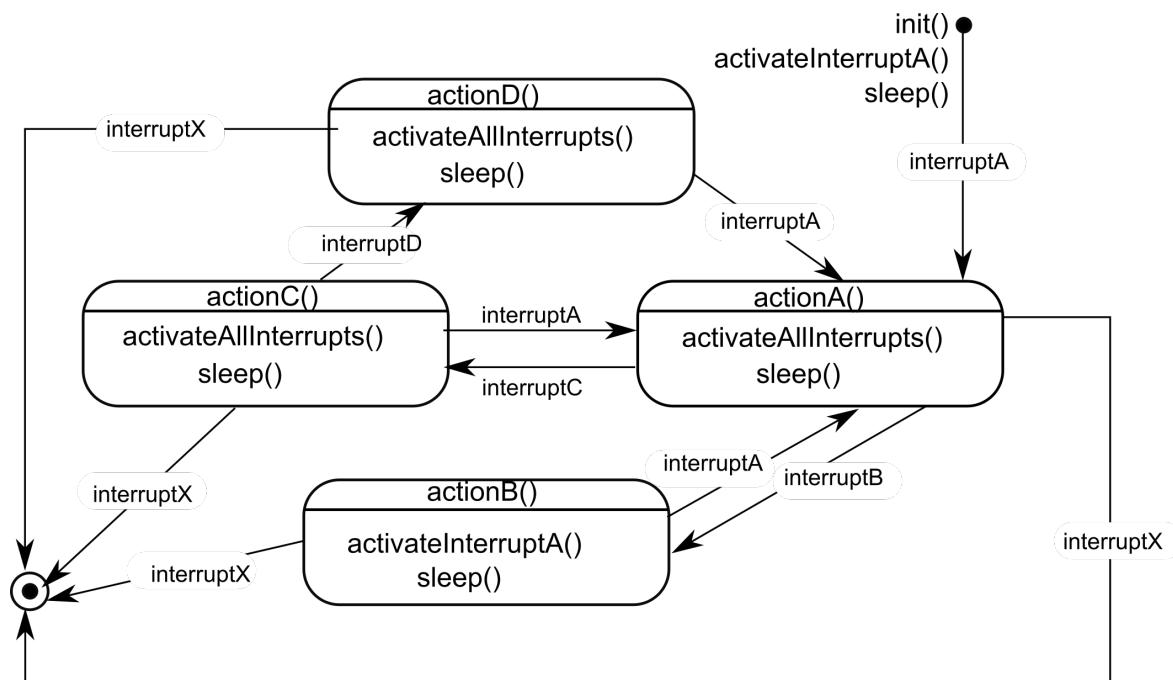


Abbildung 2.16: Struktur durch State Machine

2.4 Bluetooth Low Energy

Bluetooth Low Energy (BLE) bezeichnet eine Funktechnik, welche es ermöglicht, Daten zwischen Geräten auszutauschen. Der Vorteil von Bluetooth Low Energy ist der niedrigere Energieverbrauch im Gegensatz zum traditionellen Bluetooth. Das Bluetooth Low Energy Protokoll gehört zum Bluetooth Core Specification Version 4.0, wo auch das Classic Bluetooth Protokoll und das Bluetooth High Speed Protokoll enthalten sind. Die Bluetooth Core Specification Version 4.0 ist besser unter dem Namen Bluetooth Smart bekannt und wurde im Juli 2010 veröffentlicht (?).

2.4.1 BLE im Vergleich zu Bluetooth

Bluetooth Low Energy verwendet das gleiche Frequenzband wie das traditionelle Bluetooth, jedoch sind nur 40 Kanäle à 2 MHz verfügbar, anstatt 79 Kanäle à 1 MHz beim traditionellen Bluetooth. Ausserdem verbraucht BLE, wie der Name bereits indiziert, weniger Energie als andere Übertragungsmedien. So sendet BLE mit maximal 10 mW, was einer Reichweite von ca. 10 Metern entspricht im Gegensatz zu Klasse 1 Bluetooth-Geräten, welche mit 100 mW eine Reichweite von rund 100 Metern erreichen. Ein der BLE-Technik ist, dass die Bauteile für eine BLE-Kommunikation relativ günstig sind und damit die Geräte ebenfalls günstiger hergestellt werden können (? , Abschnitt Bluetooth Range).

2.4.2 Advertising und Connected Mode

BLE wird vor allem für batterielose Sensoren verwendet, welche die Energie aus der Umwelt beziehen. Diese Sensoren arbeiten meist als Beacon, was bedeutet, dass sie Daten senden, ohne eine aktive Verbindung mit einem Gerät aufzubauen oder nur eine Verbindung auf Anfrage eingehen, diese jedoch nach kurzer Zeit wieder beenden. Dieser Modus nennt sich Advertising Mode, was vom Englischen advertisement stammt, es soll aussagen, dass der Beacon eine Werbung aussendet und diese nicht auf eine spezielle Person zugeschnitten ist, sondern an die breite Masse gesendet wird.

Eine aktive Verbindung ist bei den meisten Sensoranwendungen auch nicht notwendig, da die Daten einfach gesendet werden können und das empfangende Gerät entscheidet was mit den vorliegenden Daten gemacht wird, wenn das Gerät mehr Informationen benötigt kann eine Verbindung aufgebaut werden. Trotzdem kann mit BLE eine aktive Verbindung eingerichtet werden, jedoch verbraucht eine aktive Verbindung mehr Energie, da Daten gesendet und empfangen werden müssen. Das bedeutet der Sensor kann nicht in einen Standby- Modus gehen, in welchem weniger Energie verbraucht wird, da auf ankommende Daten gewartet wird (?).

2.4.3 BLE Pakete

Der Aufbau eines BLE Pakets ist überschaubar. Als erstes wird ein Preamble, bestehend aus abwechselnden 1 und 0, womit der Empfänger sich auf die richtige Frequenz synchronisieren kann. Diese Preamble wird auf dafür verwendet die Verstärkung des Empfängers einzustellen, dies kann sehr wichtig sein bei Signale, welche von einer grösseren Distanz versendet werden, da eine falsche Verstärkung des Signals in Fehlern resultieren kann.

Anschliessend wird die Access Address verschickt, anhand dieser Adresse kann der Empfänger die Nachricht einem ganz bestimmten Sender zuordnen und somit entschei-

den, ob die Daten vom richtigen Sender kommen oder ob es eventuell nur Störungen waren, welche zufälligerweise eine Preamble dargestellt haben.

Der Header enthält Informationen zum Aufbau der Daten, welche folgen. Es gibt sieben verschiedene Arten von Aufbauten der Daten.

- ADV_IND – general advertising indication
- ADV_DIRECT_IND – direct connection indication
- ADV_NONCONN_INC – nonconnectable indication
- ADV_SCAN_IND – scannable indication
- SCAN_REQ – active scanning request
- SCAN_RSP – active scanning response
- CONNECT_REQ – connection request

Nachfolgen wird die Length eingereiht, welche Informationen über die Anzahl Bytes der Daten enthält. Es wird unterschieden zwischen der Länge eines Advertising Pakets und eines Data Pakets. Die Länge eines Advertising Pakets wird mit sechs Bits dargestellt, welche die Werte von 6 – 37 einnehmen können, wo ein Data Paket nur mit fünf Bits arbeitet, welche die Werte 0 – 31 einnehmen können.

Anschliessend werden die Nutzdaten übertragen, welche je nach gewählter Art, einen anderen Aufbau aufweisen. Es können zwischen 0 bis 296 Bits, also 0 bis 37 Bytes übertragen werden.

Abgeschlossen wird ein Paket mit dem CRC, welcher die Checksumme der Nachricht enthält. Die Checksumme wird über den Header, Length und die Nutzdaten gebildet (? , Kapitel 7.2).

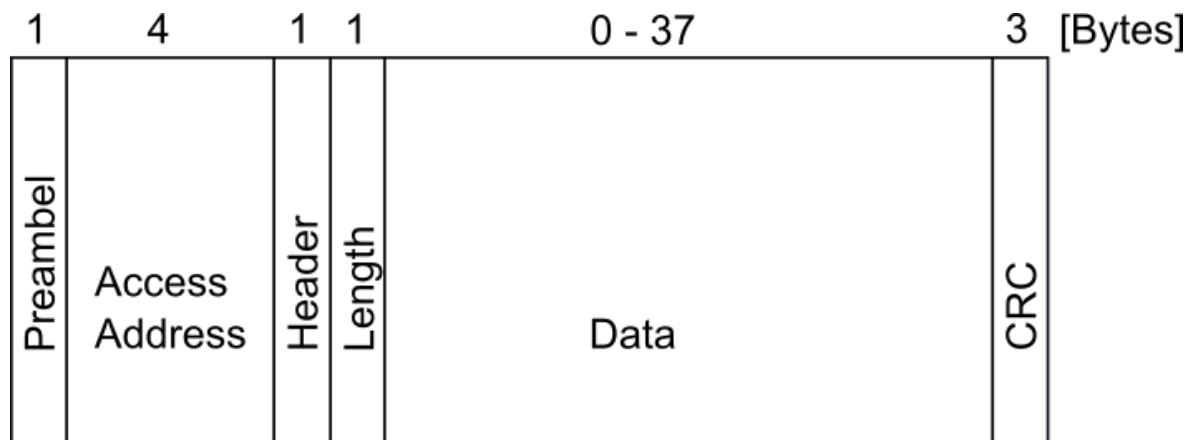


Abbildung 2.17: BLE Paketstruktur

3

Vorgehen

Das Ziel ist die Entwicklung eines Prototypen aus dem bestehenden Modell der Projektarbeit von ?. Die Vorgehensschritte sind in der untenstehenden Auflistung abgebildet. Sie bilden die Struktur dieses Kapitels. Die Definition der konkreten Schritte geschah in Absprache mit Prof. Dr. M. Meli und dem Research Assistenten Herr Dario Dünar vom Institut of Embedded Systems (InES). Auf der CD finden sich die Sitzungsprotokolle über den aktuellen Entwicklungsstand, die offenen Fragen und die Entscheidungen.

Liste der Arbeitschritte

1. Inbetriebnahme Machbarkeitsstudie
2. Hardware entwickeln
3. Inbetriebnahme Prototyp
4. Energy und Power Management
5. Entwickeln einer BLE-Applikation

Die sprachliche Unterscheidung von Energy - und Power Management entspricht der Unterscheidung zweier „Management-Teilen“ im Prototypen: Das Endprodukt regelt an zwei Stellen auf unterschiedliche Art die zur Verfügung stehende Energie. Der Begriff Energy Management wird in dieser Arbeit für das Sammeln und Weiterleiten von Energie über eine Hardwareimplementation gebraucht. Der Begriff Power Management wird für die Softwareimplementation gebraucht. Diese regelt, dass die zur Verfügung gestellte Energie nicht sofort verbraucht wird. Die sprachliche Trennung ist künstlich, denn in der Umsetzung spielen Hard- und Softwareregelung Hand in Hand. Die sprachliche Unterscheidung dient der Lesbarkeit und bezeichnet keinen physikalischen Unterschied.

3.1 Inbetriebnahme Machbarkeitsstudie

Ziel der Inbetriebnahme des Aufbaus der vorangehenden Arbeit ? ist es, zu definieren, welche Funktionalitäten verbessert werden sollen. Zur Orientierung werden im ersten Unterkapitel ?? die Funktionsblöcke und deren Aufgaben festgehalten. Danach wird das Verhalten der Vorgängermodells in Unterkapitel ?? ausgemessen. Aus der Analyse entsteht die in Unterkapitel ?? aufgelistete erste Optimierungsliste. Als letztes folgt eine Vertiefung in das auffällige Verhalten des Eingangssignals in Unterkapitel ???. Dieser Exkurs hat Ursache in der Begegnung mit Ives Théoduloz von EM MicroElectronic. Er entwickelte den in dieser Arbeit verwendete EM8500-Chip mit und wies uns auf den auffälligen Signalverlauf des Harvester-Eingangssignals (siehe Abbildung ??) hin. Dieses Signal wurde deshalb bei der Inbetriebnahme eigens getestet, was im letzten Unterkapitel dokumentiert ist.

3.1.1 Funktionsblöcke

Der Bicycle Computer besteht aus vier Funktionsblöcken, die in der Abbildung ?? dargestellt sind. Der erste Funktionsblock ist der Harvester (siehe 1) in der Abbildung ???. Die Aufgaben des Harvesters ist es Energie zu Ernten und dem nächsten Funktionsblock (Nummer 2) in der Abbildung ?? zur Verfügung zu stellen. Der zweite Funktionsblock wird als Energy Management-Teil in der Arbeit bezeichnet. Die Aufgabe des zweiten Blocks ist es, Energie zu Sammeln und kontrolliert an die Verbrauchsstelle freizuschalten. Detaillierte Informationen finden sich in den Theoretischen Grundlagen im Unterkapitel ???. Der dritte Funktionsblock ist der Ort, an dem die Energie verbraucht wird. In dieser Arbeit dient die Energie dem Betreiben von Sensoren und dem versenden derer Daten. Dies wird auf dem Sensortag (siehe Anhang ??) umgesetzt. Der Grund dafür wird in der Einleitung des Kapitels dargelegt. Der letzte Funktionsblock bezeichnet das Ziel, das Erhalten von Sensordaten in einer Applikation.

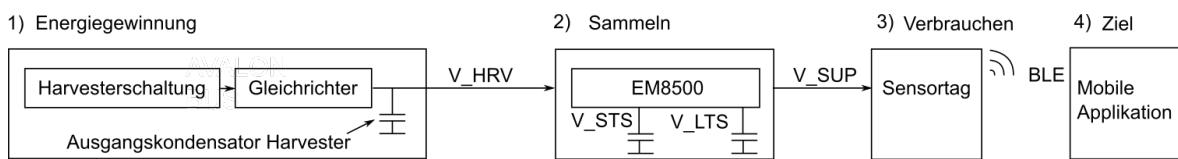


Abbildung 3.1: Funktionsblöcke Bicycle Computer

Den Funktionsblöcke sind Spannungsbezeichnungen sowie Kondensatoren beigefügt. Dies, weil bei der Beschreibung des Verhaltens des Vorgängermodells, diese Spannungslevel für die Funktionsbeurteilung wichtig werden. Die nachfolgende Legende beschreibt die Beschriftung näher.

Legende Abbildung ??

Bezeichnung	Beschreibung
V_HRV	Ausgangsspannung Harvesterquelle, Eingangsspannung Energy Management
V_STS	Spannung am STS-Kondensator (Primärspeicher)
V_LTS	Spannung am LTS-Kondensator (Sekundärspeicher)
V_SUP	Ausgangsspannung Energy Management, Eingangsspannung Sensortag
BLE	Senden der Daten per Bluetooth Low Energy (siehe ??)

3.1.2 Verhalten des Vorgängermodells

Die Inbetriebnahme bestätigte das in der Dokumentation ?? beschriebene Verhalten. Die Abbildung ?? zeigt den zeitlichen Verlauf der Energiestände zwischen den Funktionsblöcken (siehe Abbildung ?? und an den Speicherelementen. Die Legende zur Abbildung ?? erklärt die Signale.

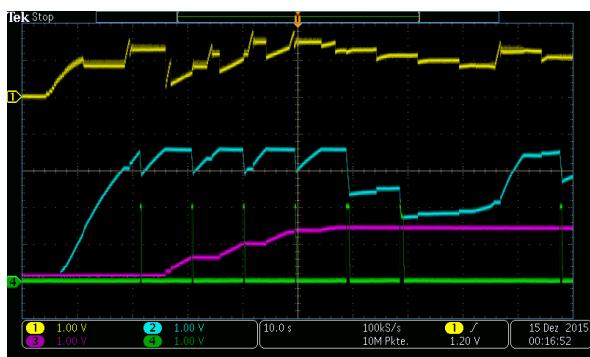


Abbildung 3.2: Spannungswerte Modell der Machbarkeitsstudie

Legende Abbildung ??

Channel	Farbe	Beschreibung
CH1	gelb	Spannungsverlauf V_HRV
CH2	blau	Spannungsverlauf am STS-Kondensator
CH3	violet	Spannungsverlauf am LTS-Kondensator
CH4	grün	Ausgangsspannung nach Energy Management Eingangsspannung Sensortag

Im Folgenden werden die einzelnen Spannungsverläufe chronologisch der Kanalnummer nach analysiert. Kanal 1 spiegelt die Spannung am Harvesterausgang wieder (V_HRV).

Gemäss Theorie ?? bzw. gemäss Datenblatt des EM8500 ist der EM8500-Chip für ein DC-Signal ausgelegt. Er geht von einem regelmässigen Eingangssignal aus und regelt die Spannung auf den MPPT. Diese Regelung sollte wie in Abbildung ?? aussehen. Das reale Signal entspricht nicht diesem Verhalten. Die Regelung ist abrupt und überspringt mehrere Spannungslevel. Der Ursache für die schlechte Regelung soll nachgegangen werden.

Kanal 2, blau, gibt den Spannungsverlauf am Hauptspeicher, dem Primärspeicher, der in der im Datenblatt von EM8500 STS heisst, wieder. Das Energy Management des Vorgängermodells setzt den Schwellwert für den Primärspeicher (STS) mit 3.6 V hoch (Details zu den Schwellwerten im Kapitel Theoretische Grundlagen ??). Der hohe Wert erklärt durch das Ziel, genug Energie für ein konstantes Paketversenden zu haben. Dies gelingt für fünf BLE-Pakete. Danach reicht die Energie nicht mehr aus. V_SUP, grünes Signal, wird nicht mehr gespiesen. Nach 30 s ohne Pakete senden ist wieder genug Energie gespeichert, für weitere 5 Datenpakete.

In der Auswertung viel uns auf, dass dieser Signalverlauf nur bei einer Geschwindigkeit gegen 70 km/h möglich ist. Die Speicherkapazität von $470 \mu\text{F}$ und ein Schwellwert der Spannung von 3.6 V ist mit normaler Geschwindigkeit (15 km/h) auch nach 30 min nicht zu erreichen. Fährt man gegen 70 km/h so erhält man die in der Abbildung gezeigte Ladezeit von rund 25 s. Eine exakte Geschwindigkeitsmessung ist zu Beginn der Arbeit nicht möglich. Das Rad wird von Hand gedreht und mit einem Metronom wird die Umdrehungsgeschwindigkeit vorgegeben. Bei einem Radumfang von 2.04 m und mit drei Umdrehungen pro Sekunde, erreicht man eine Geschwindigkeit von xxx km/h. Da diese Messmethode über längere Zeit nicht sehr genau ist, bestand eine der Aufgabe nach der Inbetriebnahme im Organisieren eines Messaufbaus. Dieser professionellere Messaufbau wird im Anhang ?? beschrieben.

Kanal 3, das pinkige Signal, zeigt die Spannung am Long Time Speicher. Die Inbetriebnahme zeigt, dass sich der LTS lädt. Es erstaunt jedoch, dass seine geerntete Energie nicht verwendet wird. Der Spannungswert von LTS geht nie herunter. Die Vermutung ist, dass der eingestellte Schwellwert für den Bezug von Energie von LTS ?? nicht stimmt.

Kanal 4, das grüne Signal, zeigt, die Speisung des Sensortags. In Modell der Projektarbeit steuert der Microkontroller des Sensortags den Verbrauch. Alle 10 s wacht das System auf, bezieht Energie vom EM8500-Ausgang für das Senden eines Paketes und geht dann wieder schlafen. Das Aufwachintervall ist fix.

3.1.3 Optimierungsliste

Aus den Messungen der Inbetriebnahme konkretisierten sich die generellen Aufgaben, die in der ersten Liste der Arbeitsschritte zu Beginn des Vorgehens beschrieben wurden. Folgende vier Punkte sollen durch den Prototypen verbessert werden:

- Der Verlauf des Harvester-Eingangs wechselt abrupt. Der Eingang soll besser geregelt werden.
- Das Laden des Primärspeichers von $470 \mu\text{F}$ in 25 s benötigt es eine Geschwindigkeit von mehr als 60 km/h. Die Harvesterschaltung soll so weiterentwickelt werden, dass bei 10 km/h genug Energie zum Senden von BLE-Paketen besteht.
- Das zweite Speicherelement, der LTS, entlädt sich nicht. Dadurch kann seine Energie nicht verwendet werden. Die Schwellwerte am EM8500 und ev. die Kondensatorenwerte sollen so angepasst werden, dass sich der zweite Kondensator entlädt
- Die Energie wird statisch nach einem fixen Zeitintervall von 10 s genutzt. Das Zeitintervall soll der Geschwindigkeit angepasst werden. Bei höherer Geschwindigkeit soll das Intervall kürzer werden.

Im Resultatteil Kapitel ?? werden die Fortschritte in diesen vier Punkten ausgewiesen.

3.1.4 Vertiefung in auffälliges Verhalten des Harvestereingangs

Text aus Einleitung: Für Argumentation. (Als letztes folgt eine Vertiefung in das auffällige Verhalten des Eingangssignals in Unterkapitel ??). Dieser Exkurs hat Ursache in der Begegnung mit Ives Théoduloz von EM MicroElectronic. Er entwickelte den in dieser Arbeit verwendete EM8500-Chip mit und wies uns auf den auffälligen Signalverlauf des Harvester-Eingangssingals (siehe Abbildung ??) hin. Dieses Signal wurde deshalb bei der Inbetriebnahme eigens getestet, was im

?????? folgte.) Hier ist nun dieser erwähnte Text.

Vorschlag Katrin: Die auffällige Regelung des Harvesterinputs wurd Ives Théoduloz gezeigt.

Gemäss Ives Théoduloz sollten Kondensatoren der Harvesterschaltung im Bereich von $4.7 \mu\text{F}$ liegen, sodass die Energiemanagementschaltung ordnungsgemäss funktioniert.

In der Machbarkeitsstudie ist nach dem Gleichrichter ein Kondensator von $470 \mu\text{F}$ nachgeschaltet. Dieser glättet die Spannungspulse nach dem Gleichrichter zu einer DC-ähnlichen Spannung mit Rippeln.

Aus diesem Grund wird die Rippelspannung am Ausgang der Harvesterschaltung mit kleineren Kondensatoren gemessen. Das Messprotokoll befindet sich im Anhang.

a. Ausmessen der Auswirkung des Ausgangskondensators

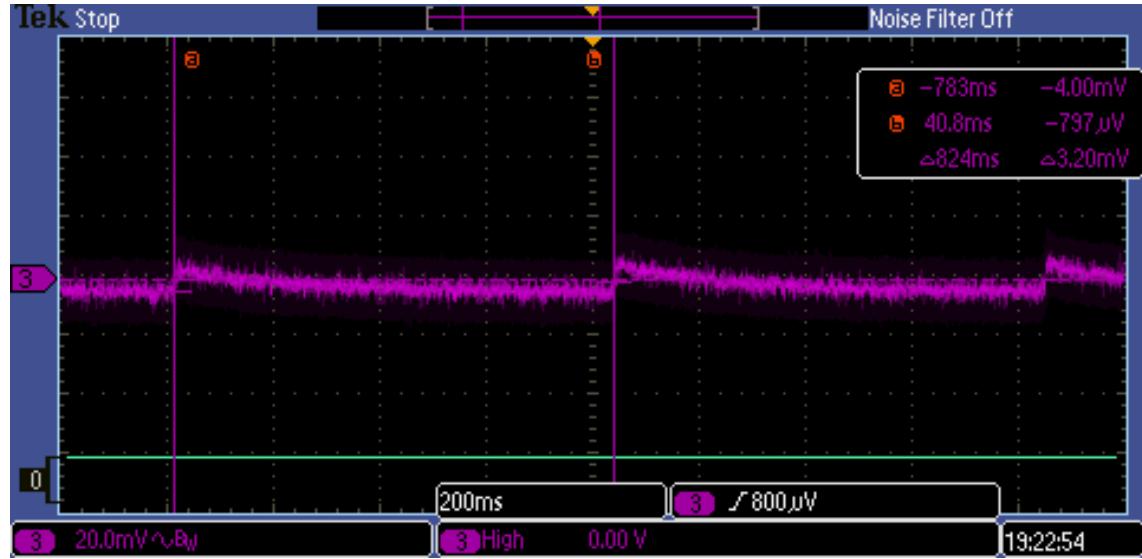


Abbildung 3.3: Rippelspannung bei Glättung mit $470 \mu\text{F}$ Kondensator

Mit einem Kondensator von $470 \mu\text{F}$ wird die Ausgangsspannung der Harvesterspannung fast rippelfrei. Die Rippelspannung beträgt 3.2 mV (siehe Abbildung ??).

Messaufbau

In der gegebenen Harvesterschaltung wird am Kondensator die Spannung mit einem Kathodenstrahlloszilloskop (KO) gemessen. Ausgehend vom bestehenden Kondensator ($470 \mu\text{F}$), werden danach Elektrolytkondensatoren (Elko) mit den Werten $100 \mu\text{FF}$, $47 \mu\text{F}$ und $10 \mu\text{F}$ gemessen.

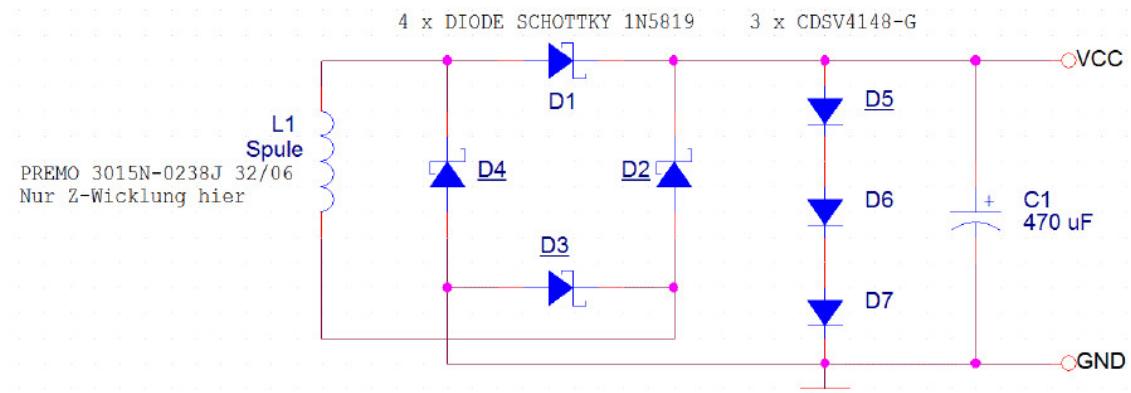


Abbildung 3.4: Messschaltung

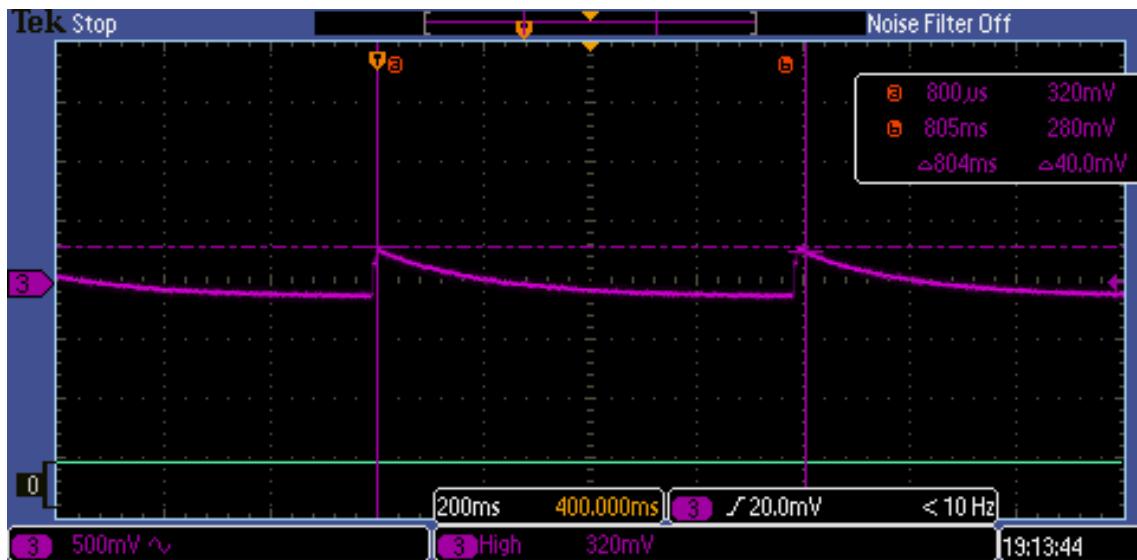


Abbildung 3.5: Rippelspannung mit 10 uF Kondensator

Resultat

Die Rippelspannung erhöht sich wie erwartet. V_{pp} beträgt bei 100 uF xx mV, bei 47 uF 28.8 mV (siehe Abbildung ??) und bei 10 uF 320 mV (Abbildung ??).

Nach Rücksprache mit einem Entwickler des EM8500-Chips wird der zu hohe Kondensator vor dem Harvester-Eingang als Ursache vermutet. Laut Datenblatt sollte dieser 4.7 μ F, damit der Booster mit eingebautem MPPT ideal regeln kann.

Aus den Beobachtungen ergaben sich folgende Aufgaben für die Entwicklung des Prototypen:

1. Die Harvesterschaltung funktioniert nicht optimal. Die Auswirkung des zu hohen Kondensator vor Harvestereingang soll getestet werden
2. Die Schaltung soll für eine Geschwindigkeit von 10 km/h ausgelegt werden
3. Die Konfigurationen beim EM8500 sollen überarbeitet werden, sodass LTS genutzt wird
4. Das Senden der Pakte soll der Geschwindigkeit angepasst werden

Punkte 1 und 2 haben Auswirkung auf das Layout, Punkte 3 und 4 auf die

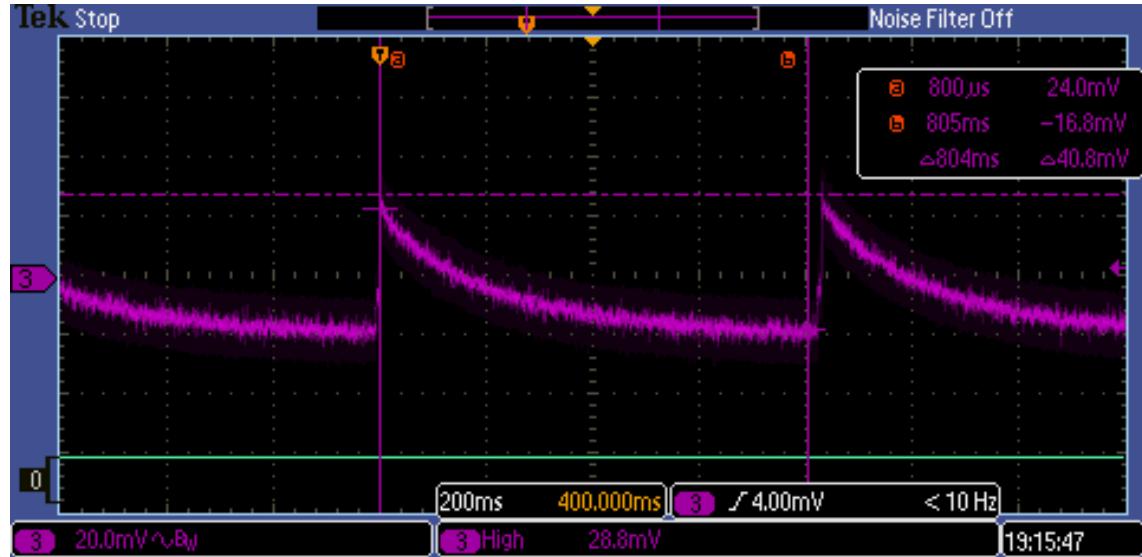


Abbildung 3.6: Rippelspannung mit 47 uFKondensator

3.2 Hardware entwickeln

Notiz: Ziel dieses Artikels gemäss Inbetriebnahme: - Genug Energie für BLE bei 10 km/h

3.2.1 Schema

3.2.2 Bauteildefinition und Optimierung

3.2.3 Layout

— alter Text — Ein wichtiger Punkt der Arbeit ist die Miniaturisierung der bestehenden Hardware, das heisst der Aufbau aus der Machbarkeitsstudie soll auf eine Leiterplatte gebracht werden. Die Leiterplatte hat einige Vorgaben, welche im besten Fall alle eingehalten werden sollen.

1. Die Leiterplatte soll nicht oder nur geringfügig grösser sein als das TI-SensorTag.
2. Alle Netze sollen mit Testpunkten ausgestattet werden.
3. Alle Anschlüsse vom TI-SensorTag sollen auf der Leiterplatte mit Testpunkten ausgestattet werden.
4. Alle Testpunkte vom TI-SensorTag sollen in einem Raster von 2.54 mm angeordnet werden, damit ein Stecker kontaktiert werden kann.

3.2.4 Das Schema (oder der Stromlaufplan)

Als erstes musste ein Schema, auch Stromlaufplan genannt, gezeichnet werden. Das Schema wurde Blockweise erfasst, als erstes wurde die Harvesterschaltung erfasst. Das Schema wurde aus der Machbarkeitsstudie entnommen. Die Funktionsweise der Harvesterschaltung kann wieder in mehrere Teile unterteilt werden.

1. Die Spule: Gewinnt Energie aus dem vorbei schnellenden Magneten.
2. Der Gleichrichter: Erzeugt positive Pulse aus der induzierten Spannung.
3. Der Limiter: Limitiert die Spannung auf eine fixe Spannung.
4. Der Ausgangskondensator: Glättet die positiven Pulse aus dem Gleichrichter.

Der nächste Block ist der EM8500-Chip mit seinen Stützkondensatoren. Das Schema wurde aus dem Datenblatt entnommen. Die Energiespeicher, welche in dieser Arbeit mittels Elektrolytkondensatoren dargestellt werden, sind einige der wichtigsten Elemente. Die Speicherelemente werden nicht auf der Leiterplatte Platz finden, da die meisten Elektrolytkondensatoren zu hoch sind und der Platz zwischen den Leiterplatten sehr gering ist. Die Umlauferfassung wird mit einem Reed-Switch ermöglicht. Der Reedswitch ist einer der kleinsten Blöcke im Schema. Der Block Interface enthält die Verbindung zum TI-SensorTag, ein Stecker realisiert dieses Interface. Der Stecker ist bereits vom TI-SensorTag vorgegeben, es handelt sich um einen Stecker, welcher sein eigenes Gegenstück darstellt.

3.2.5 Optimierung der Harvesterschaltung

Nach dem Erfassen des Schemas wurde die Optimierung der Hardware angegangen. Die beste Optimierungsmöglichkeit und auch der kritischste Block ist die Harvesterschaltung, hier wird die Energie für die restliche Schaltung gewonnen. In mehreren Schritten wurden die einzelnen Teile der Schaltung analysiert und versucht zu optimieren.

a. Optimierung der Spule

Die Spule gewinnt die Energie aus dem vorbei schnellenden Magneten, hier kann die gewonnene Energie beeinflusst werden. Eine gute Spule kann mehr Energie aus dem bewegten Magneten gewinnen, wichtig ist die Induktivität L und die Fläche A, welche die Spule hat. Eine Vorgabe war dass die Spule von der Grösse nicht merklich verändert wird, ausser man würde eine kleinere Spule finden, welche mehr Energie gewinnt. Eine Spule mit ähnlicher Fläche bzw. Grösse wurde gefunden, welche eine höhere Induktivität besitzt. Die Spule von Würth Elektronik ist sehr vielversprechend, denn die gleiche Fläche mit höherer Induktivität bedeutet mehr Energiegewinn aus dem Magneten.

b. Optimierung des Gleichrichters

Der Gleichrichter aus dem Aufbau der Machbarkeitsstudie besteht aus vier Dioden vom Typ 1N5819, diese Dioden sind nicht für eine LowPower-Anwendung ausgelegt. Ausserdem könnte ein Gleichrichter gefunden werden, welcher in einem Gehäuse ausgeliefert wird. Wichtig ist dass der Leckstrom so gering wie möglich ist und die Schwellenspannung ebenfalls möglichst klein bleibt.

c. Optimierung des Limiter

Der Limiter ist eine Spannungsbegrenzung, da die nachfolgende Schaltung nicht mit einer zu hohen Spannung betrieben werden darf. Dieser Schaltungsteil ist sehr kritisch, denn er darf nicht zu viel Energie verlieren, muss aber trotzdem die Spannung immer begrenzen. Die Spannung darf einen Pegel von 2 V nicht überschreiten, da ansonsten der EM-8500-Chip droht zerstört zu werden. .

er Schluss-
t von
sprotokoll
fügen

d. Optimierung des Ausgangskondensators

Der Ausgangskondensator muss möglichst niedrig gehalten werden, gemäss Aussage von Yves, da ansonsten der EM8500-Chip Mühe hat den Eingang zu regeln. Trotzdem darf der Ausgangskondensator nicht zu klein dimensioniert werden, da ansonsten die Rippelspannung am Ausgang der Harvestersschaltung zu hoch ist und der EM8500-Chip ebenfalls nicht mehr richtig regeln kann.

3.2.6 Bauteildefinition

Nachdem das Schema gezeichnet wurde und die Schaltung optimiert wurde, mussten die Bauteile noch definiert werden. Es mussten die Footprints, sowie die Hersteller, Herstellerbezeichnungen, Lieferant und Lieferantenartikelnummer hinterlegt werden. Einige Footprints waren bereits in den Bibliotheken vorhanden, welche wir von Lukas erhalten haben. Fehlende Footprints wurden ergänzt, wie zum Beispiel der Footprint der Spule.

3.2.7 Das Layout

a. Positionierung

Die Positionierung der Bauteile auf der Leiterplatte ist sehr wichtig, da hier schon unnötige Leiterbahnen gespart werden können bzw. die Länge von gewissen Leiterbahnen können extrem verkürzt werden. Wichtig ist, dass die Stützkondensatoren bei dem EM8500-Chip so nah wie möglich am Chip platziert werden, damit die Spannungen am Chip so konstant gehalten werden können, wie nur möglich. Weiter sollte die Harvesterschaltung ebenfalls sehr eng beieinander platziert werden, um zu verhindern, dass durch lange Stromlaufwege bereits Leistung verloren geht. Problematisch ist, dass die Spule auf der unteren Seite der Leiterplatte platziert werden muss, somit wird die Schaltung ein auf zwei Layer aufgeteilt. Eine grosse Herausforderung ist die Positionierung der Testpunkte, um das Interface zum TI-SensorTag zu realisieren. Dadurch wird ein grosser Platz für die korrekte Positionierung der Testpunkte eingenommen.

b. Gestaltung der Leiterbahnen

Wann immer möglich wurden die Leiterbahnen, welche zu der Harvesterschaltung gehören, mit 20 Mil gezogen, um eine möglichst verlustfreie Leistungsübertragung zu gewährleisten. Alle anderen nicht leistungskritischen Leiterbahnen wurden mit einer Leiterbahnbreite von 10 Mil platziert, um nicht mehr Platz in Anspruch zu nehmen als nötig.

c. Ergebnis

Das Ergebnis ist eine Leiterplatte, welche alle gewünschten Spezifikationen erfüllt und somit kann die Leiterplatte auch für ein Praktikum verwendet werden. Die Leiterplatte ist mit sehr vielen Testpunkten ausgestattet, sowie die Möglichkeit für Strommessungen.

3.3 Inbetriebnahme des Prototypen

Ziel des Kapitels: Ausmessen und sehen, dass zu wenig Energie. (Weglassen. Testteil).
2. Magnete nach 180 % (Bild), Reellight bringt idee für 2 Magnete hintereinander, besser Spule verwenden

— alter Text Der entwickelte Prototyp wurde intensiv ausgemessen (siehe Messprotokolle XXXX.) Es werden 3 Messstellen unterschieden siehe Abbildung ???. In den

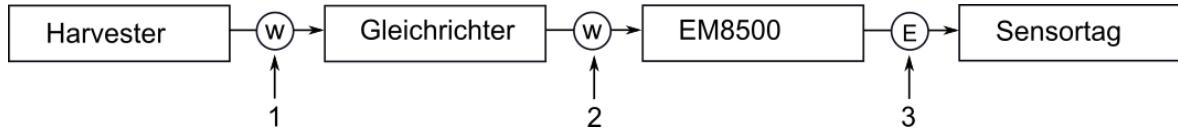


Abbildung 3.7: Messstellen am Prototypen

folgenden Unterkapiteln werden die Resultate und die darauf folgenden Entwicklungsschritte kurz beschrieben:

3.3.1 Testen der Harvesterschaltung

- Zu wenig Energie - Zwei Magnete - stärkere Spule

3.3.2 Ausmessen der Energie vor und nach dem Gleichrichter

3.3.3 Energiemessungen nach dem EMBoard

3.4 Energy Management

Die Primäraufgabe des Energy Managements ist es, dass die zur Verfügung stehende Energie bei 10 km/h genügt, zum BLE-Pakete zu versenden. Die sekundäre Aufgabe ist es, dass man erreicht, dass sich der Long Time Storage entweder nicht lädt oder aber sich auch entlädt. Ein Laden des LTS ohne Energieabgabe ist Energieverschwendug. Die zwei gestellten Aufgaben sollen durch möglichst optimale Speichergrößen und intelligente Schwellwerte erreicht werden.

Damit man die Kondensatorenwerte berechnen kann, braucht es Energiedaten. Deshalb stehen im ersten Unterkapitel ?? Messergebnisse, danach folgt im zweiten Unterkapitel ?? die Dimensionierung der Kondensatoren und dann das Berechnen der Schwellwerte in Unterkapitel ???. Als letzter Punkt werden aufgrund der Ladewerte der Kondensatoren Energiezustände definiert. Dies, weil die letzte offene Aufgabe der Optimierungsliste ???: Das Ersetzen des fixen Paketversendens nach 10 s dem Energiezustand angepasst werden soll. Die Vorgehensweise zur Erfüllung dieses Punktes ist im letzten Unterkapitel ?? beschrieben.

3.4.1 Energiemessungen

Die Entwicklung wird konstant begleitet durch Energiemessungen. Sei dies durch Leistungsmessungen bei der Hardware (Harvester und EM8500-Chip) oder sei dies als Energiemessung der Software (Sensortag). Für die Energiemessungen konnte der Power Analyser von Agilent Electronic N6705B mit der Serienummer MY50000795 gebraucht werden. Dieses mächtige Messgerät misst gleichzeitig Strom, Spannung und somit die Leistung im Zeitverlauf. Mühsame Annäherungen aus eigenen KO-Messungen sind somit nicht mehr notwendig. In diesem Unterkapitel werden die Resultate vieler Messungen in ihrer zeitlichen Reihenfolge zusammengefasst. Die Detailaufnahmen zu allen Messungen finden sich in der CD im Ordner Messungen in der Datei Energiemessungen.pdf.

Energiemes...
Sensortag p...
generieren

a. Erste Energiemessungen

Um einen Anhaltspunkt über den Energieverbrauch des Sensortags zu erhalten, wurde 3 BLE-Pakete im Advertising Mode per Knopfdruck gesendet (siehe Abbildung ??). Dies entspricht der Messung zur Sensortagversion V0. Nach der Weiterentwicklung des Programms, folgte die Messung V1, in der zusätzlich die GPIO ausgelesen werden. Dies ist für das Einlesen der Reed Switch-Impulse unumgänglich und für das Einlesen der Energy States vom 8500 sinnvoll. Die Version V2, erster Versuch mehr Sleep-Time einzubauen, scheiterete am Zusammenspiel des Timings der Radio- und GPIO-Interrupts. Das Neuaufsetzen des Programms half, sodass die Version V3 power-optimiert Geschwindigkeitspaket sendet. Die untenstehende Tabell stellt die ersten Energieresultate dar. Unterschieden wird zwischen dem Energieverbrauch für die Initialisierung und der Energie zum Senden von 3 BLE Paketen. Die Diskussion über Energieoptimierungen und die Deutung der Resultate finden sich in den Sitzungsprotokollen vom xxxx - xxxx.

Messresultate nach Sensortag-Versionen

Version	Datum	Energie Init	Energie Senden
V0	10.3.16	unbekannt	33 μ J
V1	16.3.16	87 μ J	32 μ J
V3	22.4.16	40 μ J	29 μ J

Bei diesen Messungen werden die Sensoren noch nicht ausgelesen. Es zeigt sich, dass das Auslesen der Sensoren über I2C und das Warten, bis dass die Sensoren aufgestartet

sind, viel mehr Energie verbraucht, als erwartet. Der Energieverbrauch des Auslesens nur eines Sensors (erste Priorität hat laut Aufgabenstellung der Drucksensor) ist ohne Optimierungsmassnahmen so gross, dass kein Senden der Daten mehr möglich ist. Die Speisung der Applikation bricht sofort zusammen (siehe Abbildung ??).

Aus diesen Grund wurde die Power-Optimierung innerhalb des Programms zur zentralen Herausforderung in dieser Arbeit. Details darüber sind im nächsten Unterkapitel ?? Power Management zusammengefasst. Hier die Auflistung der Energiewerte von Version V4. Bei dieser Version werden die Sensoren ausgelesen und dazwischen geht das Sensortag in den Schlafmodus, damit sich der Primärkondensator wieder laden kann. Erst nachdem alle Sensoren ausgelesen sind, wird das BLE-Paket versendet.

b. Sensortag

Der Energieverbrauch des Sensortags wird 4-mal gemessen. Einmal mit der Minimalversion: Nur Pakete mit Geschwindigkeitsdaten senden. Dann diese Version mit den drei Sensoren, die einer nach dem anderen ausgelesen werden. Weil dazwischen geschlafen wird, damit das System wieder genug Energie gespeichert hat, dauert diese Version xxx s. Von jedem Sensor wird der eigene Energieverbrauch ermittelt, um einen Vergleich unter ihnen herstellen zu können.

b..1 Nur Geschwindigkeit: Simple Harvester

b..2 Alle drei Sensoren nacheinander: Advanced Harvester

b..3 Nur Drucksensor

b..4 Nur Temperatursensor

b..5 Nur Feuchtigkeitssensor

3.4.2 Energiekalkulation

Der Energieverbrauch hängt von der Anzahl ausgelesener Sensoren ab (siehe ??). Für die Bestückung der Kondensatoren wird vom schlechtesten Fall, einer Geschwindigkeit von 10 km/h ausgegangen. Die Grundlagen zur Energiekalkulation $E_{HRV}^- \geq E_{BLE}^-$ sind im Theorieteil ?? abgebildet. Hier werden die Kondensatoren berechnet.

$$C_{STS} = \frac{E_{Applikation}}{(V_{STS_HIGH})^2 - (V_{STS_LOW})^2} \quad (3.1)$$

$$C_{STS} = \frac{E_{Applikation}}{(3.6V)^2 - (2.1V)^2} \quad (3.2)$$

$$C_{STS} = \frac{E_{Applikation}}{(3.6V)^2 - (2.1V)^2} \quad (3.3)$$

$$C_{STS} = \frac{E_{Applikation}}{(3.6V)^2 - (2.1V)^2} \quad (3.4)$$

Abschluss
Übergang

a. MPP einstellen

Das Ziel der Entwicklung ist, dass bei einer Geschwindigkeit von 10 km/h genug Energie zum Senden eines BLE-Pakets zur Verfügung steht. Während der Entwicklung des Harvesters wurde die Leistungskurve öfters aufgenommen (siehe Anhang ??, xxx,yyy, zz). Wie im Theorienteil erklärt ?? unterscheidet sich das Leistungsmaximum nach Geschwindigkeit. Weil das Ziel ein funktionstüchtiger Prototyp bei 10 km/h ist, bezieht sich das Einstellen nur auf diese Wert.

MPP-Ratio bei 10 km/h gemäss Messprotokollen

Datum	Leistungmaximum	MPP-Ratio
30. März	12 µW	43.23 %
xx	xx µW	xx %
resultat	21.87 µW	24.87 %

Werte

Es zeigt sich, dass das Leistungsmaximum unterhalb von 50 % liegt. Bedauerlich ist, dass durch die Energieoptimierung sich die Stelle des Leistungsmaximums Richtung Leerlauf verschiebt. In der Umsetzung mit dem EM8500-Chip ergibt sich das Problem, dass nur Konfigurationen von MPPT-Ratios von 50 - 80 % erlaubt sind (siehe Tabelle unten). Die Einstellungen der tieferen MPP-Ratio sind zudem größer. Dies, weil der EM8500-Chip für Harvester vom Typ TEG (mit einem MPP konstant bei 50 %) und

Solarzelle konzipiert ist und die MPPT-Ratio für diese zwei Anwendungen genau in diesem Range liegen. In unserem Fall ist dieser vorgegebene Range nicht ideal. Da beim energiekritischen Zustand bei 10 km/h die Ratio deutlich unter 50 %. Die Vorgänger wählten in ihren Einstellungen eine MPPT-Ratio von 88 %. Die Vermutung liegt nahe, dass keine Leistungskurve des Harvesters im Voraus aufgenommen wurde.

MPPT-Ratio Einstellungen EM8500

Registerwert	MPPT-Ratio
0x00	50 %
0x01	60 %
0x02	67 %
0x03	71 %
0x04	75 %
0x05	78 %
0x06	80 %
0x07	82 %
0x08	83 %
0x09	85 %
0x0A	86 %
0x0B	87 %
0x0C	88 %

Im ersten Versucht wird das Register auf das Minimum (50 %) eingestellt (siehe Sitzungsprotokoll YYYY). Die Messung ergab, dass bei 10 km/h das EM-Board nicht gespiesen wird. So luden sich weder der STS noch der LTS auf. Der Grund ist, dass der optimale Leistungsbezug bei 50 % liegt, das Leistungsmaximum aber später. So entstand beim Regeln auf den optimalen Leistungsbezug der Umstand, dass dort nur 0.2 V produziert werden. Dieser Spannungswert ist zu tief, der EM-Chip beginnt nicht zu arbeiten. Betreibt man den Harvester nicht am optimalen Punkt, sondern bei einer MPPT-Ratio von 60 %, so wird eine Eingangsspannung von 0.3 V produziert und EM8500 beginnt zu arbeiten.

3.4.3 Einstellen der Schwellwerte

Das Konzept der Schwellwerte von VSTS und VLTS sind im Theorieteil bei der Erklärung des Speicherkonzepts des EM8500-Chips im Unterkapitel ?? beschrieben. Das Ziel ist, dass bei 10 km/h sich LTS entlädt. Die Berechnung der Schwellwerte stammt aus dem Theorieteil ??.

Aus der Vorgängerarbeit sind in der Tabelle unten notierte Konfigurationswerte überliefert.

Bei der Inbetriebnahme fiel der hohe Schwellwert von v_bat_min_hi_dis auf (3.6 V), ab der erst die Applikation gespiesen wird. Den hohen Wert beruht auf dem Versuch, möglichst viel Energie vor dem Datensenden zu sammeln. Ein hoher Schnellwert verzögert die Zeit, bis zum ersten Datensenden.

Konfiguration Vorgängermodell

Register Wert

v_bat_max_hi	4.2 V
v_bat_max_lo	4.1 V
v_bat_min_hi_dis	3.6 V
v_bat_min_hi_con	2.1 V
v_bat_min_lo	1.8 V
v_appl_max_hi	3.8 V
v_appl_max_lo	3.7 V
mppt_ratio	88 %

Nach der Energiemessungen der Version V3 des Sensortags (siehe ??), in dem die Geschwindigkeit per BLE gesendet werden kann, entstanden folgende Schwellwerte:

Erste Konfiguration aufgrund Geschwindigkeitsmessung

Register Wert

v_bat_max_hi	4.8 V
v_bat_max_lo	4.7 V
v_bat_min_hi_dis	2.6 V (siehe Berechnungen unten)
v_bat_min_hi_con	2.1 V
v_bat_min_lo	2.0 V (vorgegeben von Sensortag)
v_appl_max_hi	3.8 V
v_appl_max_lo	3.7 V
mppt_ratio	50 % (aus MPP-Kurve Harvester)

Die Grundlage bildete die Gleichung $E_{Applikation} = E_{STS_HIGH} - E_{STS_LOW}$, wobei $E_{Applikation}$ aus der Messung von V3 mit total $93.2 \mu\text{J}$ für die Initialisierung und das Senden der Daten gebraucht werden, und V_{SUP} durch das Sensortag mit 2.0 vorgegeben ist. Laut Datenblatt des EM8500 (?, S. 9, Abschnitt Operating Conditions) soll STS zwischen $10 - 100 \mu\text{F}$ gross sein. Typisch ist $47 \mu\text{F}$. Um auf der sicheren Seite zu sein, denn der Harvester gibt zu diesem Zeitpunkt noch nicht genug Energie ab, wurde als Berechnungsgrundlage C_STS von $100 \mu\text{F}$ genommen. So ergab sich folgender minimaler oberere Schwellwert:

$$(v_{bat_min_low})^2 = \frac{2 \times E_{Applikation}}{C_{STS}} + (V_{STS_LOW})^2 \quad (3.5)$$

$$(v_{bat_min_low})^2 = \frac{2 \times 93.2, \mu J}{100 \mu F} + (2.0)^2 \quad (3.6)$$

$$v_{bat_min_low} = \sqrt{\frac{186.4}{100} + 4} = \sqrt{5.864} = 2.42V \quad (3.7)$$

Alternativ kann man den laut Datenblatt vorgeschlagenen Kondensatorenwert von 47 μF . Dann ergibt sich als oberer Schwellwert $v_{bat_min_low} = \sqrt{\frac{186.4}{47} + 4} = \sqrt{7.97} = 2.82, V$. Als erstes wird der Schwellwert auf 2.6 V gesetzt. Wichtigster Wert in der Konfiguration ist jedoch der korrekte MPPT-Wert. Er wird auf das Minimum (50 %) gesetzt. Das Maximum des Ladezustands wird auf den maximal erlaubten Wert gesetzt: $v_{bat_max} = 4.8 V$ bzw. $4.7 V$. Denn solang Energie geerntet werden kann, soll sie gespeichert werden. v_{appl_max} spielt in unserer Anwendung keine wichtige Rolle. Der Wert wird von den Vorgängern übernommen.

Das Zusammensetzen der Last (Sensortag) mit der konfigurierten Hardware erstaunte (siehe Messprotokoll xxx und Sitzungsprotokoll yyy). Ab 25 km/h funktionierte die Schaltung gut. Doch darunter reagiert der EM8500-Chip nicht. Weder STS noch LTS konnten sich laden. Nachmessungen am Eingang vom EM8500-Chip ergaben, dass bei 10 km/h und einer MPPT-Ratio von 50 % eine Spannung von 0.2 V anliegt. Diese ist zu tief, um den Booster zu starten. Erst bei höherer Geschwindigkeit wird eine Eingangsspannung von mehr als 0.3 V erreicht, und das System beginnt zu funktionieren. Aus diesem Grund wird die MPPT-Ratio zukünftig auf den zweitiefsten Wert (60 %) eingestellt.

Ein unerwartetes Verhalten zeigt sich bei der Implementation des Auslesen des ersten Sensors, der Drucksensor. Der BLE-Sniffer empfängt die Druckdaten korrekt, doch beim Anschliessen des Sensortags an den Harvester, riss die Spannung sofort zusammen. Innert kürzester Zeit haben sich entladiet sich der STS wie parallel zu ihm der LTS und unterschreitet der STS den Schwellwert von V_{SUP} von 2 V, so stellt die Speisung der Applikation ab (siehe Abbildung ?? im Unterkapitel ??). Das Ausmessen des Energieverbrauchs bringt Klarheit: Die I2C-Kommunikation und das Aufstarten des Sensors brauchen xxx-mal mehr Energie, als das Berechnen der Geschwindigkeit aus den Reed-Switch Impulsen (siehe Abbildung yyy im Resultat-Teil).

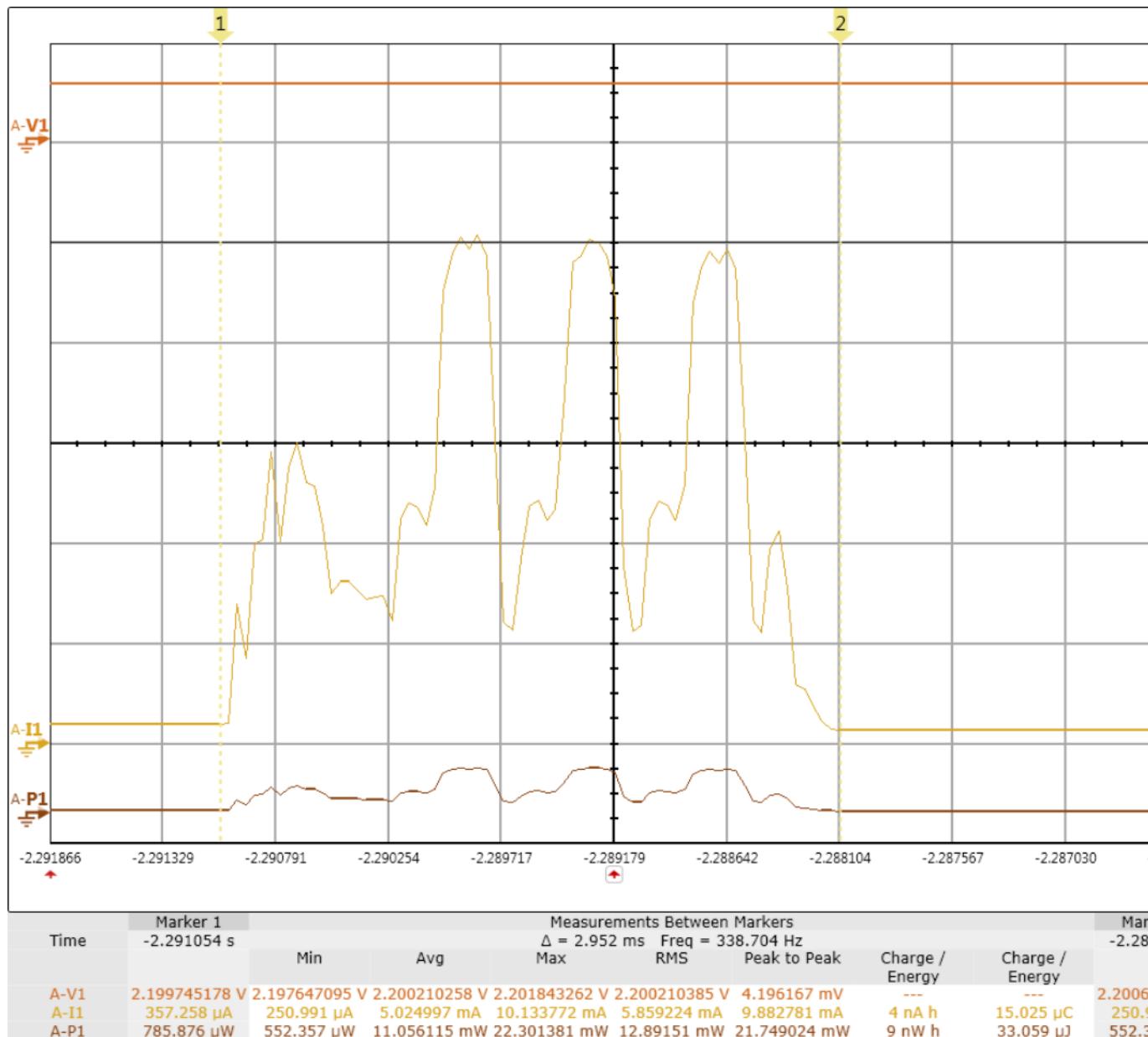


Abbildung 3.8: Minimalster Energieverbrauch: 3 BLE Pakete über Advertising Mode senden

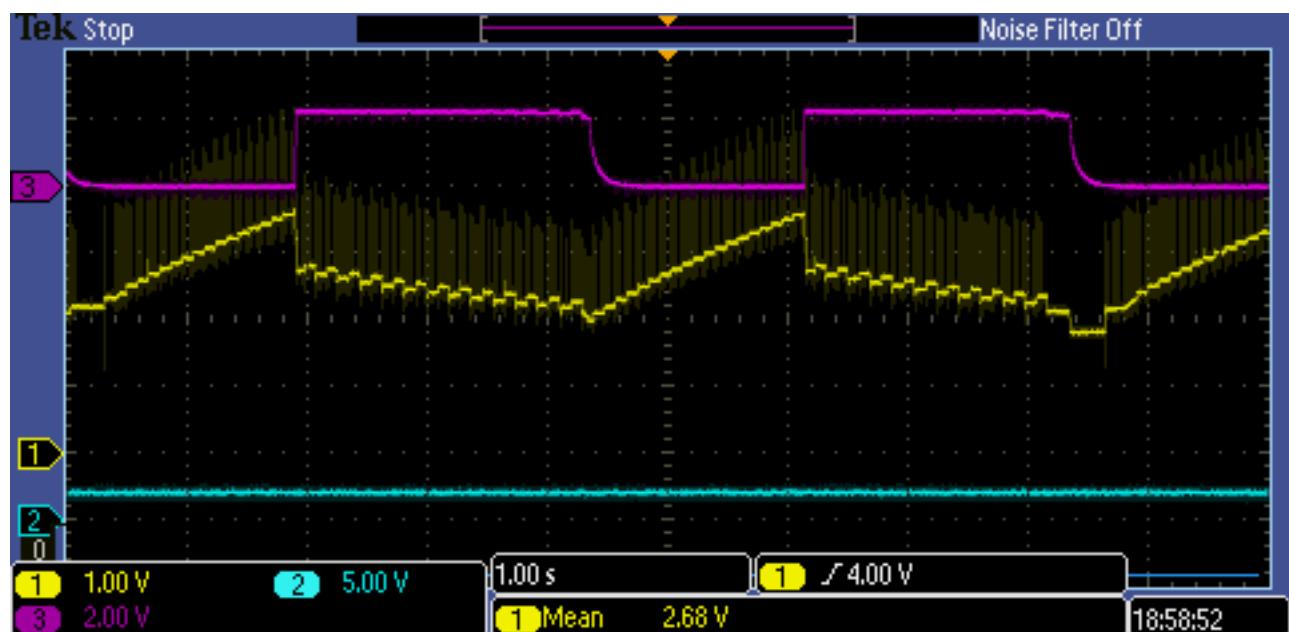


Abbildung 3.9: Auslesen der Sensoren reisst Energie zusammen

4

Resultate

Im Kapitel ?? ist die Umsetzung ausführlich erklärt. In diesem Kapitel geht es um die relevanten Resultate. Diese teilen sich in die bekannten Kategorien Harvester, Energy - und Power Management sowie die BLE-Applikation. Die Resultate jeder Kategorie werden in einem Unterkapitel beschrieben.

4.1 Harvesterschaltung

4.1.1 Der Print

Das erste konkrete Resultat ist der fertige Print. In der Abbildung ?? ist die Vorderseite abgebildet. Darauf befindet sich im Wesentlichen der Header als Verbindungsstecker zum Sensortag, die vier Gleichrichterdioden, der Limiter, die Spule und der EM8500. In der Abbildung ?? befindet sich der Reed-Switch und die dreilagige Premo-Spule, mit der die Induktion erzeugt wird.

bessere Be-
schreibung

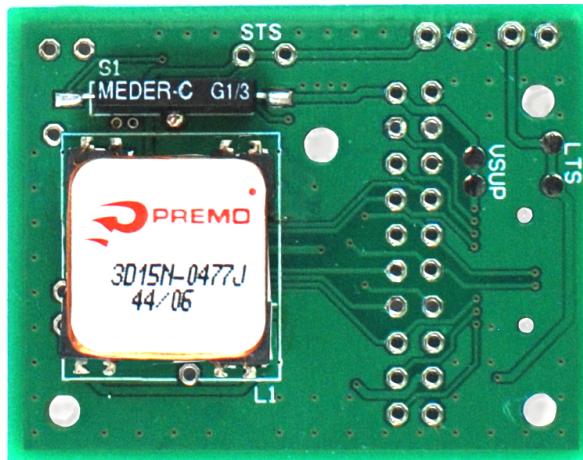


Abbildung 4.1: Print Vorderseite

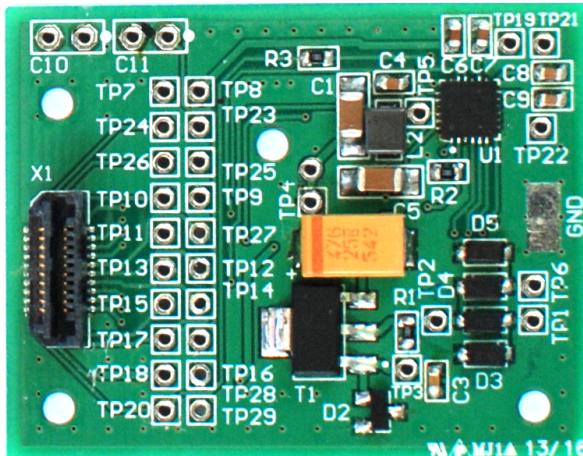


Abbildung 4.2: Print Rückseite

4.1.2 Leistung am Harvesterausgang

Wie erwartet (siehe theoretische Grundlagen ??) ändert sich bei einer Hardware mit Spule das Leistungsmaximum. Die Abbildung ?? zeigt die reale MPP-Kurve des Prototypen. Bei 10 km/h liegt das Leistungsmaximum bei xxx mW, bei 15 km/h bei xxx mW, bei 20 km/h bei xxx mW und bei 40 km/h bei xxx mW. Die detaillierten Angaben finden sich im Messprotokoll xxxx .

Tabelle Leistung Harvesterausgang

Geschwindigkeit	Leistung Harvester
10 km/h	21.87 μ W
15 km/h	57.19 μ W
20 km/h	114.67 μ W

40 km/h \dot{c} 416.29 μ W

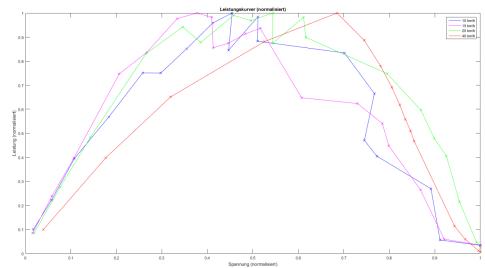


Abbildung 4.3: Leistungskurve Harvesterausgang (normalisiert)

4.1.3 Verhalten des Harvesterausgangs

Die Abbildung ?? zeigt, das reelle Verhalten des Harvesters bei Belastung mit dem EM-Chip. Wie in beschrieben ?? regelt der Chip EM8500 den Eingang, bzw. den Ausgang des Harvesters, um den MPP zu erreichen. Die Abbildung ?? zeigt den Verlauf der geregelten Spannung am Eingang des EM-Chips über längere Zeit. Weitere Messungen zum reelen Verhalten am EM8500-Chip-Eingang können den Messprotokoll entnommen werden.

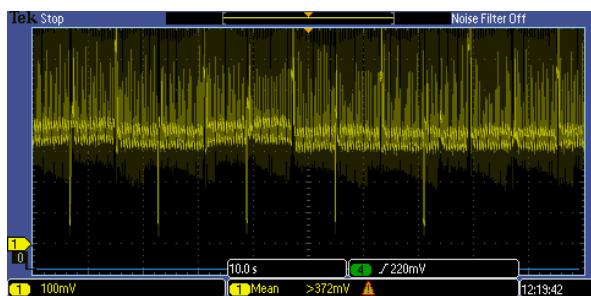


Abbildung 4.4: Spannung VCC beim Harvesterausgang bei 15 km/h

4.1.4 Energie am EM-Chipausgang

Die Abbildung ?? zeigt die berechnete Energie, die am Ausgang des EM-Chip abgegeben wird in Bezug zur Geschwindigkeit. Die Berechnung, wie aus dem Puls über die Zeit, bis dass VSUP wieder eingeschalten wird, ist im Messprotokoll yyy dokumentiert. Die gewonnene Energie, die dem SensorTag zur Verfügung steht ist:

Tabelle Leistung und Wirkungsgrad

Geschwindigkeit	Leistung EM8500_out
10 km/h	5.44 μ W
15 km/h	20.91 μ W
20 km/h	41.39 μ W
40 km/h	170.75 μ W

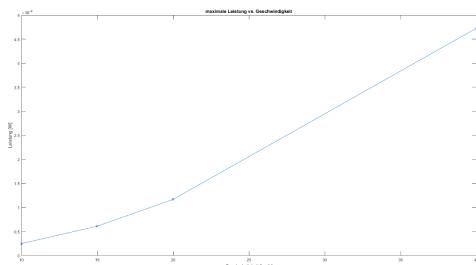


Abbildung 4.5: Maximale Leistung vs. Geschwindigkeit

4.1.5 Wirkungsgrad des Prototypen

Aus den zwei Leistungsmessungen wird graphisch die Energiegewinnung im Vergleich dargestellt (Abbildung ??). Die Werte beziehen sich auf die gemittelte Leistung über 10 s. Es ist zu sehen, dass die Energiegewinnung bei der Harvesterschaltung mit der Geschwindigkeit deutlich schneller zunimmt, als die Energie am Ausgang des EM-Chips.

Aus den zwei Leistungsmessungen wurde der Wirkungsgrad bei den gemessenen Geschwindigkeiten berechnet (siehe Tabelle unten). Der Wirkungsgrad ist bei tiefen Geschwindigkeiten sehr schlecht (24.87 %). Dies ist eine der zwei Ursache, dass trotz genug geernteter Energie, das SensorTag für das Senden der Sensorwerte bei tiefer Geschwindigkeiten nicht wie erwartet genug Energie hat. Mit mehr Geschwindigkeit, erhöht sich

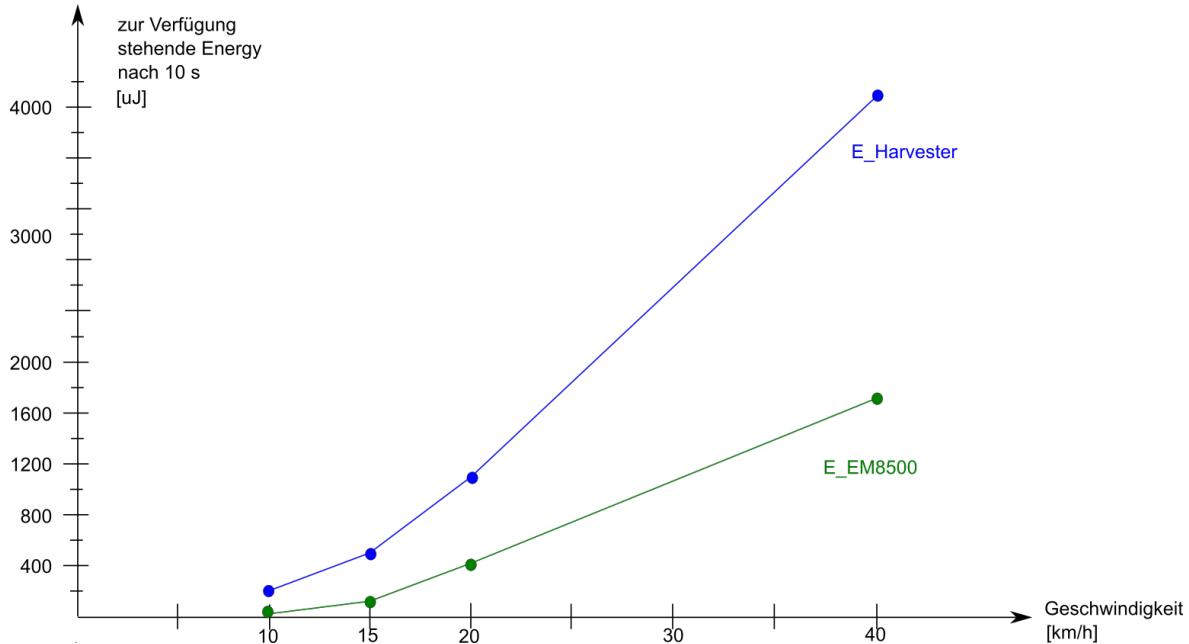


Abbildung 4.6: Energiegewinn Zusammengefasst nach Stelle in der Schaltung

nicht nur die geerntete Leistung rapide, sondern auch der Wirkungsgrad. Der Wirkungsgrad des EM8500 innerhalb des entwickelten Prototypen liegt bei 40 km/h bei 41.01 %.

Tabelle Leistung und Wirkungsgrad

Geschwindigkeit	Leistung Harvester	Leistung EM8500_out	Wirkungsgrad
10 km/h	21.87 μ W	5.44 μ W	24.87 %
15 km/h	57.19 μ W	20.91 μ W	36.56 %
20 km/h	114.67 μ W	41.39 μ W	36.09 %
40 km/h	416.29 μ W	170.75 μ W	41.01 %

4.2 Energy Management

Das Ziel beim Energy Management ist, dass die vom Long Time Storage gespeicherte Energie für das Versenden der BLE-Pakete genutzt wird. Dies ist (siehe Abbildung ??) möglich. Die gesammelte Energie beider Kondensatoren wird für das Sender der BLE-Pakete verwendet. Die benutzten Kondensatorenwerte sind:

$$STS = 100 \mu F$$

$$\text{LTS} = 2200 \mu\text{F}$$

Die Werte ergaben sich aus folgender Energiekalkulation

$$W = \frac{1}{2} \cdot 100\mu\text{F} \cdot (3.6 - 2.2V)^2 = xxxxxuW \quad (4.1)$$

$$W = \frac{1}{2} \cdot 2.2m\text{F} \cdot (3.6 - 2.2V)^2 = xxxxxmW \quad (4.2)$$

Dadurch sammelt STS in einer Periode von xx ms eine Energie in der Größenordnung von yyy mJ. LTS lädt in der Periode von zz ms zusätzlich eine Energie in der Größenordnung von yyy mJ. Die angereicherte Menge von tt mJ genügen, um ein xxx-Paket zu versenden.

finale Zeiten
eintragen.
Energie darf
eintragen

besseres Bild



Abbildung 4.7: LTS liefert Energie für die Arbeitspakete

Gelungen ist das Laden und Entladen beider Kondensatoren durch folgende EM-Schwellwerte:

correct spacing
in table

Tabelle Finale Konfiguration Schwellwerte

Register	Spannungswert
v_bat_max_hi	3.9 V
v_bat_max_lo	3.8 V
v_bat_min_hi_dis	3.6 V
v_bat_min_hi_con	2.2 V
v_bat_min_lo	2.0 V
v_appl_max_hi	3.8 V
v_appl_max_lo	3.7 V

mi ni low
d höher

4.3 Powermanagement

Lange schien es so, dass die Sensoren, aufgrund des grossen Verbrauchs der I2C-Kommunikation und der langen Aufstartzeit der Sensoren, nicht per BLE bei tiefer Geschwindigkeit gesendet werden können. Eine Lösung wurde möglich, weil Aufgabenblöcke in Unterblöcke mit Schlafenzeiten eingebaut wurden. Die untenstehende Tabelle zeigt die gefundene Lösung:

Tabelle Energieverbrauch Aufgabenblöcke

Funktion	Energieverbrauch
Auslesen Reed Switch und Berechnen Geschwindigkeit	xx mW
Starten und Auslesen Drucksensor	xx mW
Starten und Auslesen Temperatursensor	xx mW
Starten und Auslesen Feuchtigkeitssensor	xx mW
Auslesen Statusregister per SPI	xx mW

Die Lösung ist das Aufsplitten von Aufgabenblöcken. Bei genug Energie, wird nicht ein Sensor gestartet, ausgelesen und seine Daten versendet, sondern der Sensor wird gestartet, dann geht das System schlafen, bis dass LTS voll ist. Dann folgt der nächste Aufgabe: die Werte auslesen. Es wird wieder geschlafen und beim dritten Mal genügend Energie wird das Paket versendet. Das Konzept des Schlaffens zwischen Aufgabenblöcken ist im theoretischen Teil im Unterkapitel ?? im Abschnitt ?? beschrieben . Die Abbildung ?? zeigt, das Aufteilen der Arbeitsschritte: Zuerst folgt die Initialisierung des Sensors, dann folgt das Auslesen der Daten und zum Schluss das Senden der Werte.

Bild Arbeits
aufteilung



Abbildung 4.8: Aufteilen in drei Schritte bis zum Senden der Daten

Das Ergebnis ist in der Abbildung ?? graphisch dargestellt. Es zeigt sich, dass das Senden von Sensordaten xx mal mehr Energie verbrauchen, als das Senden der Geschwindigkeit. Die schlechteste Bilanz hat das Auslesen des Status Registers aus dem EM8500-Chip. In diesem Byte ist hinterlegt, welche Schwellwerte des EM8500 überschritten wurden und dient dadurch auf einfache Art dem Abbilden des aktuellen Energiezustands (siehe Unterkapitel ??).

Wert eintra
Tabelle und
Text



Abbildung 4.9: LTS liefert Energie für die Arbeitspakete

Kombiniert man den Energieverbrauch mit der zur Verfügung stehenden Energie am Ausgang nach dem EM8500-Chips, können (siehe Abbildung ??) folgende Schlussfolgerungen gezogen werden:

auere An-
ten nach
messungen

1. Das Übermitteln der Geschwindigkeit ist ab 10 km/h möglich
2. Das Übermitteln aller Sensordaten (gestaffelt) ist ab xxx km/h möglich
3. Das Übermitteln aller Sensordaten braucht bei einer Geschwindigkeit von 10 km/h xxxx s
4. Das Übermitteln aller Sensordaten braucht bei 40 km/h xxx s

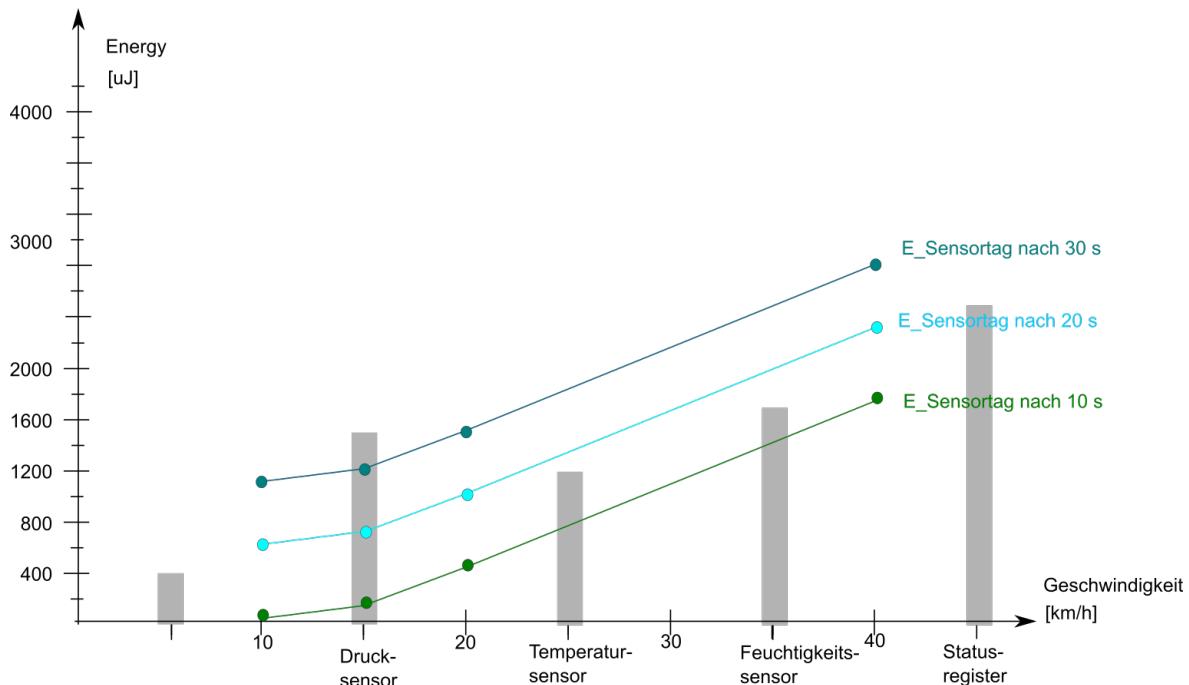


Abbildung 4.10: Energieverbrauch gemäss Verarbeitungsaufwand für CPU

4.4 Ergebnisse BLE-Applikation

Der Aufbau des Sensortags zusammen mit dem selbst entwickelten Board wird „Sensor“ genannt. Dies, weil aus der Sicht einer Benutzerin oder eines Benutzers keine detaillierte Hardware besteht, sondern „ein Sensor“. Die Vereinfachung dient der Benutzerfreundlichkeit. Die Android-Applikation ist bewusst einfach aufgebaut, um die Benutzerin bzw. den Benutzer nicht zu verwirren.

Der animierte Tachometer stellt die aktuelle Geschwindigkeit schnell und übersichtlich dar. Weitergehende Funktionen sind durch prägnante Namen selbsterklärend und der User sollte keine Mühe haben, die App ohne Lesen einer Anleitung zu verstehen.

4.4.1 Applikationsstruktur

Beim Öffnen der Applikation wird geprüft, ob Bluetooth aktiviert ist (siehe Abbildung ??). Sollte Bluetooth nicht aktiviert sein, wird der User gefragt, ob Bluetooth aktiviert werden darf. Sollte der User die Aktivierung ablehnen schliesst sich die Applikation sofort.

Wird eine Verbund zu Bluetooth erlaubt, erscheint der Startbildschirm. Auf diesem befindet sich zentral der Tachometer (siehe Abbildung ??). Dieser zeigt Geschwindigkeiten von 0 – 90 km/h mit einer animierten Tachonadel an. Unterhalb des Tachometers werden die einzelnen Sensordaten angezeigt und im untersten Teil des Startbildschirms befinden sich zwei Buttons, über die man Einstellungen vornehmen kann. Die Kontextmenüs zu diesen Einstellungen werden in den nächsten zwei Abbildungen ?? und ?? ersichtlich.

Wählt man auf dem Startbildschirm „Sensor wählen“, erscheint ein neuer Bildschirm. Auf diesem erscheinen nur die aktiven Bluetooth Geräte mit dem implementierten Prototypen-Filter. Der Bildschirm bildet die Sensortagadresse ab. Jedes Sensortag hat eine eigene Adresse und bei mehreren Prototypen im Raum, kann das entsprechende Gerät ausgewählt werden.

ev. besser:
Sensorboard
wählen

Auf dem Startbildschirm befindet sich auch ein Konfigurationsknopf. In das Untermenü gelangt man, in dem man den Button Einheiten + Einstellungen auswählt.

Text: warum
user das Se
nor nennt. U
es dehalb so
heisst und
nicht Board

Die Applikation stellt reiche Konfigurationsmöglichkeiten zur Verfügung (siehe Abbildung ??). Zu jedem Sensor gehört ein Drop Down-Element, in dem die Einheiten ausgewählt werden können. Neben der Einheitenauswahl verfügt die App über die Möglichkeit, den Radumfang anzupassen und die Temperatur zu kalibrieren.

Die Radgrösse, sprich der Radumfang, muss einstellbar sein, da die interne Geschwindigkeitsberechnung von diesem abhängt. Da die Radgrösse nicht normiert ist, konnte

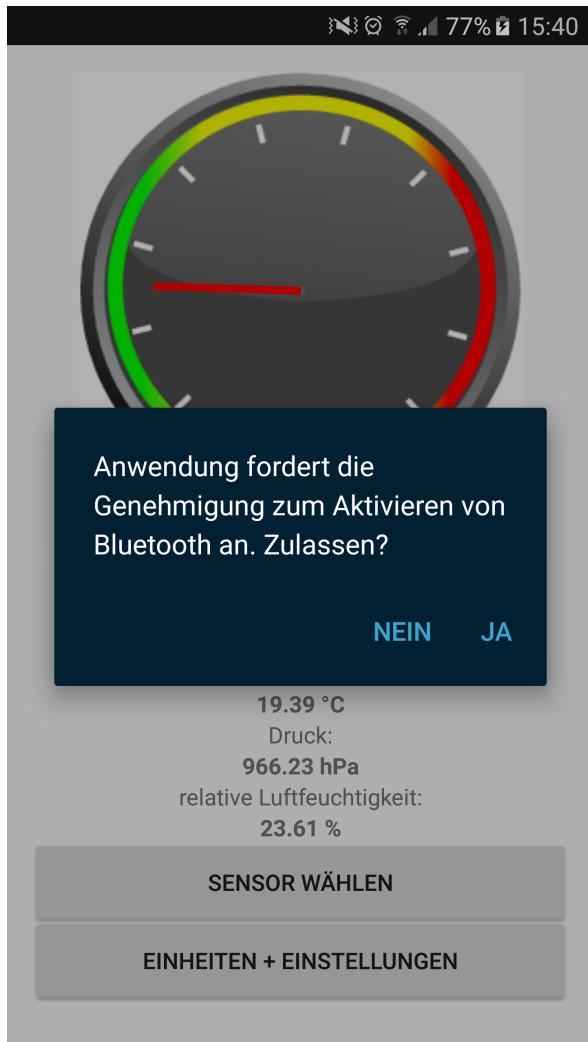


Abbildung 4.11: Bluetooth Permission

kein Drop Down-Element eingebaut werden. Die Zahl wird über die Tastatur in das Feld eingegeben.

Bei der Temperaturanzeige ist eine Kalibrierung eingebaut. Dies, weil alle drei Temperatursensoren auf dem Sensortag einheitlich zu hohe Werte ausgeben. Der zu hohe Temperaturwert liegt am Aufbau des Prototypen. Sensortag und Print liegen nahe aufeinander und es entsteht Wärme im Zwischenraum. Der Benutzer oder die Benutzerin kann den korrekten Wert im Kalibrationsfeld eingeben. Die nächste Temperatur, die empfangen wird, wird mit dem eingetragenen Wert verglichen und ein Offset wird eingestellt. Wird keine Kalibration eingetragen, wird die Temperatur nicht kalibriert und der Offset bleibt unverändert.

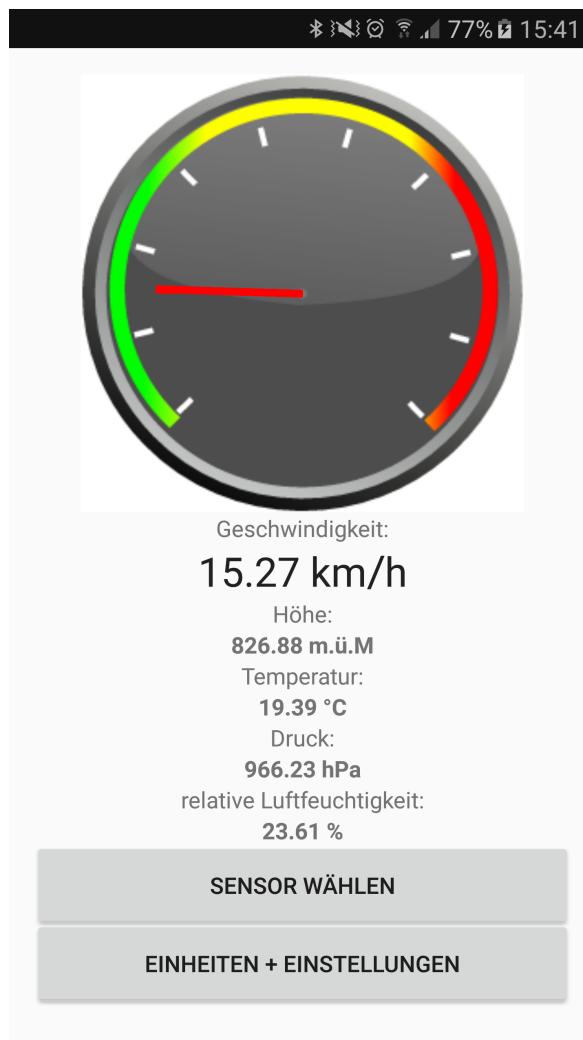


Abbildung 4.12: Startbildschirm der Applikation

Auswählbare Einheiten

Temperatur	Geschwindigkeit	Druck
Celsius (C)	Kilometer pro Stun de (km/h)	Pascals (Pa)
Fahrenheit (F)	Miles per hour (mp h)	bar (bar)
Kelvin (K)		Atmosphäre (atm)
		Pound-Force per square inch (psi)
		Millimeter Quecksilber (mmHG)

Die vorgenommenen Einstellungen werden über den Button „Speichern“ gesichert. Danach werden die eingelesenen Sensordaten in den ausgewählten Einheiten dargestellt.

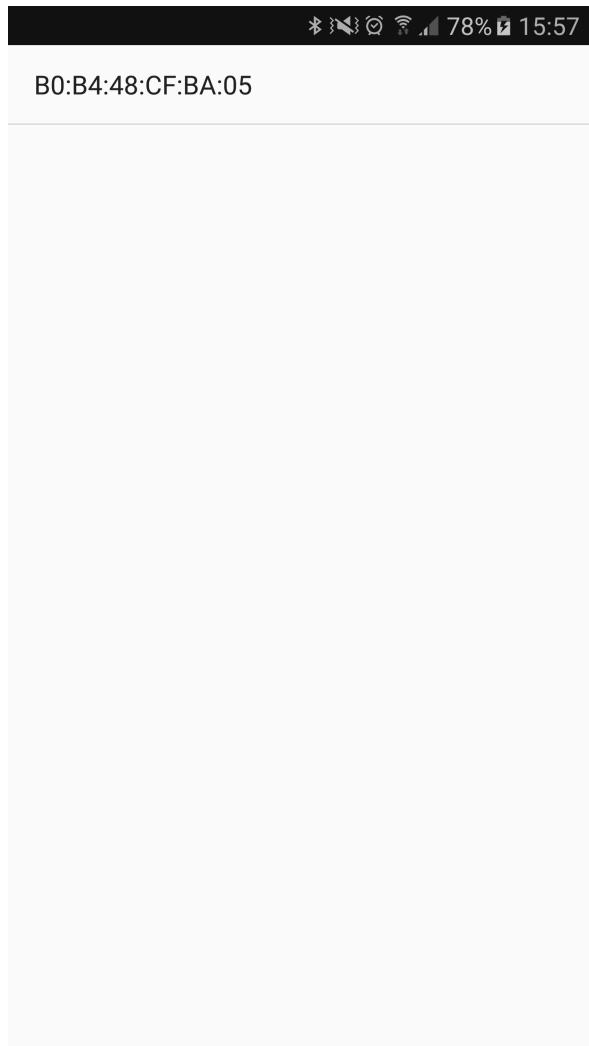


Abbildung 4.13: Sensortagauswahl

4.4.2 Paketverlust

Sensortag sendet im Advertising Mode 3 Pakete pro Sendevorgang. Gemessen wird der Paketverlust bei einer Geschwindigkeit von 20 km/h (siehe untenstehende Tabelle und).

sprotokoll
ähnen

eelle refe-
zieren

Paketverlust BLE-Applikation

Messperiode	Gesendete Pakete	Empfangene Pakete	Paketverlust
1 min	38	35	7.9 %
1 min	37	31	16.2 %
2 min	75	64	14.7 %
2 min	77	65	15.6 %
5 min	193	168	13.0 %

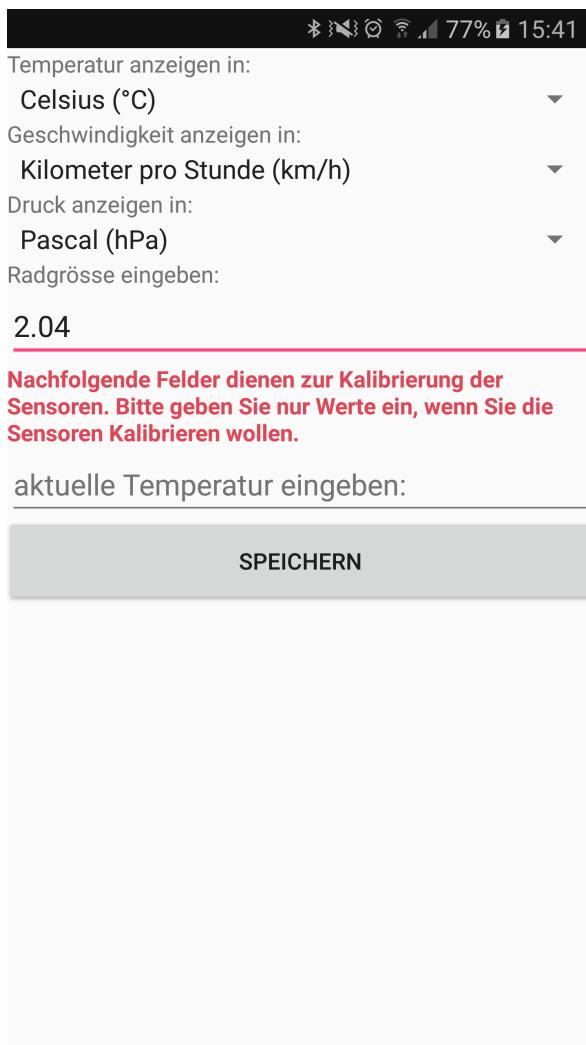


Abbildung 4.14: Einheiten und Einstellungen

5 min	190	158	16.8 %
10 min	345	301	12.8 %
10 min	349	302	13.5 %

Da der Advertising Mode keine sicher Verbindung ist, ist ein Paketverlust von rund 15 % zu erwarten.

4.4.3 Korrektheit der Daten

Die Korrektheit der empfangenen Daten auf der Applikation wird mit einem visuellen Vergleich der Datenpakete im BLE-Sniffer von TI verglichen (siehe Abbildung ??). Die Daten im Sniffer entsprechen exakt den Daten, welche die Applikation empfängt. Der Inhalt der Daten wird somit unverfälscht übermittelt.

elles Snif-
Bild

Packet	Time (us)	Channel	Access Address	Adv PDU Type	Adv PDU Header	Type TxAdd RxAdd PDU-Length	AdvA	ScanRepData	CRC
14	51456412	0x25	0x00FF0004	ADV_SCAD_BSF	4	0 0 0 15	0x3C2D874F430	0B 09 4B 65 79 66 0E 42 84 44 42 87	0x1F742D
15	+12998936	0x25	0x00FF0004	Adv PDU Type	Adv PDU Header	Type TxAdd RxAdd PDU-Length	AdvA	AdvData	CRC
15	+12998936	0x25	0x00FF0004	ADV_SCAD_BSF	4	0 0 0 15	0x3C2D874F430	02 01 06 07 02 03 18 02 18 04 18	0x1BCFB5
16	+12998673	0x25	0x00FF0004	Adv PDU Type	Adv PDU Header	Type TxAdd RxAdd PDU-Length	ScanA	AdvA	CRC
16	+12998673	0x25	0x00FF0004	ADV_SCAD_BSF	4	1 0 0 12	0x48A3980148E7B	0x3C2D874F430	0x398A45
17	+3271200	0x25	0x00FF0004	Adv PDU Type	Adv PDU Header	Type TxAdd RxAdd PDU-Length	AdvA	ScanRepData	CRC
17	+3271200	0x25	0x00FF0004	ADV_SCAD_BSF	4	0 0 0 15	0x3C2D874F430	0B 09 4B 65 79 66 0E 42 84 44 42 87	0x1F742D

Abbildung 4.15: Snifferpaket Prototyp

eo zeiger
(CD)

5

Diskussion

Bespricht die erzielten Ergebnisse bezüglich ihrer Erwartbarkeit, Aussagekraft und Relevanz

Interpretation und Validierung der Resultate

Rückblick auf Aufgabenstellung, erreicht bzw. nicht erreicht Legt dar, wie an die Resultate (konkret vom Industriepartner oder weiteren Forschungsarbeiten; allgemein) angeschlossen werden kann; legt dar, welche Chancen die Resultate bieten

Es ist machbar. Stellen zum Optimieren definieren

Reflexion Katrin:

EM8500 I2C geht nicht richtig. Kein RTOS bebrauchen Code von Ines chaotisch Super Unterstützung.

Reflexion Manu:

Dank:

Ives The.... Leiterplattenhersteller

bei App: we
mehrere uns
rer Geräte i
Raum, dann
sie voneinan
der untersch
den

6

Verzeichnisse

6.1 Glossar und Abkürzungen

Clock Domain

Ein Bereich der Hardware, der mit demselben Takt läuft.

GPIO

General Purpose Input and Output bezeichnet die externe Schnittstelle zum Sensortag. Alle Zustände, die das EM8500 sendet, gelangen über die GPIO zum SensorTag. Für den Bicycle Computer ist der Reed Switch-Eingang der wichtigste, da aus diesem die Geschwindigkeit berechnet werden kann.

InES

Ist der Institutsname eines Instituts der ZHAW. InES steht für Institut of Embedded Systems.

Power Domain

Basiert auf der Fähigkeit eines Prozessor Speisungsgebiete zur Verfügung zu stellen. Der Prozessor teilt seine Funktionalitäten in Gebiete ein, die separat ein- und ausgeschalten werden können.

MPP

Maximum Power Point (MPP) bezeichnet in einer Leistungskurve den höchsten Punkt, also das Leistungsmaximum.

MPPT

Versucht ein System, einen Input stets auf das Leistungsmaximum zu regeln, spricht man von Maximum Power Point Tracking. Tracking steht für Einfangen.

MPP-Ratio

Bezeichnet die Auswertung des MPP auf Spannungsachse. Liegt das Leistungmaximum beim Kurzschluss, so ist die MPP-Ratio bei 0 %, liegt sie bei Leerlauf, dann liegt die MPP-Ratio bei 100 %. Üblicherweise liegt die MPPT-Ratio dazwischen.

MPPT-Ratio

Einstellungswert eines Registers im EM8500-Chip. Durch den Wert gibt man vor, auf welchen MPP das System sich einstellen (tracken) soll.

RTC

Der Real Time Clock kann im Cortex M3 bei ausgeschaltener CPU weiterlaufen.

State Machine

Heisst korrekt Finite State Machine und bezeichnet eine Konzept, bei dem aufgrund einer Kombination von Eingangssignalen, sich das System in einem bestimmten Zustand befindet. In jedem Zustand sind nur gewisse Inputs zulässig, ansonsten verbleibt das System in diesem Zustand. Folgt ein korrekter Input, wechselt das System in den entsprechenden Zustand.

UML

Die Unified Modeling Language (UML) ist ein Quasistandard, wie Prozesse abgebildet werden können. Die Sprache definiert Formen, aufgrund deren man weiß, ob es sich um eine Initialisierung, eine Entscheidung oder um eine Verarbeitung, etc. handelt.

6.2 Abbildungsverzeichnis

6.3 Tabellenverzeichnis

Anhang A

Ausschreibung Bachelorarbeit

Bachelorarbeit 2016 – FS: BA16_mema_1

Titel

Energy harvesting powered bicycle computer

Beschreibung

Im Rahmen dieser Bachelorarbeit soll ein batterieloses Geschwindigkeitsmessgerät für Fahrräder entwickelt werden. Das Messgerät wird an der Gabel des Fahrrades befestigt und misst die Anzahl Durchgänge eines an den Speichen befestigten Magneten. Der Tachometer gewinnt die gesamte benötigte Energie aus den Magnetdurchläufen. Die gemessenen Daten werden per Bluetooth an einen Fahrradcomputer (in diesem Fall ein Mobiltelefon) gesendet.

Der Machbarkeitsbeweis wurde durch die vorangegangene Projektarbeit „Bicycle computer and sensoric powered with harvested energy“ (PA15_mema_1) bereits erbracht. Der hier entwickelte Fahrradcomputer wird sich durch ein intelligentes Energieverwaltungssystem vom Prototypen abheben. Dies bedeutet, der Fahrradcomputer soll wissen, wie viel Energie bereits gespeichert wurde und er soll prognostizieren, wie viel Energie in näherer Zukunft geerntet werden kann. Aufgrund dieser und möglicherweise weiteren Informationen berechnet der Fahrradcomputer den optimalen Zeitpunkt und Zeitabstand, um die gemessenen Geschwindigkeitsdaten zu versenden. Sobald genug Energie zur Verfügung steht, beginnt der Fahrradcomputer ausserdem Werte wie der Luftdruck (Höhe), die Temperatur und die Luftfeuchtigkeit zu messen. Optional kann auch ein Bewegungssensor in Betrieb genommen werden.

Es soll ausserdem eine Smartphone App entwickelt werden, welche die gemessenen Geschwindigkeitsdaten, Sensordaten und aktuelle Werte der Energieverwaltung empfängt sowie darstellt. Optional kann ein Kommunikationskanal vom Smartphone zum Fahrradcomputer implementiert werden, um Parameter wie Radumfang oder Sicherheitseinstellungen auszutauschen. Selbstverständlich müsste der Fahrradcomputer auch dann noch ohne Batterie auskommen.

Firmware: 40%
Hardware: 30%
App: 30%

Anhang B

Blockdiagramm EM8500

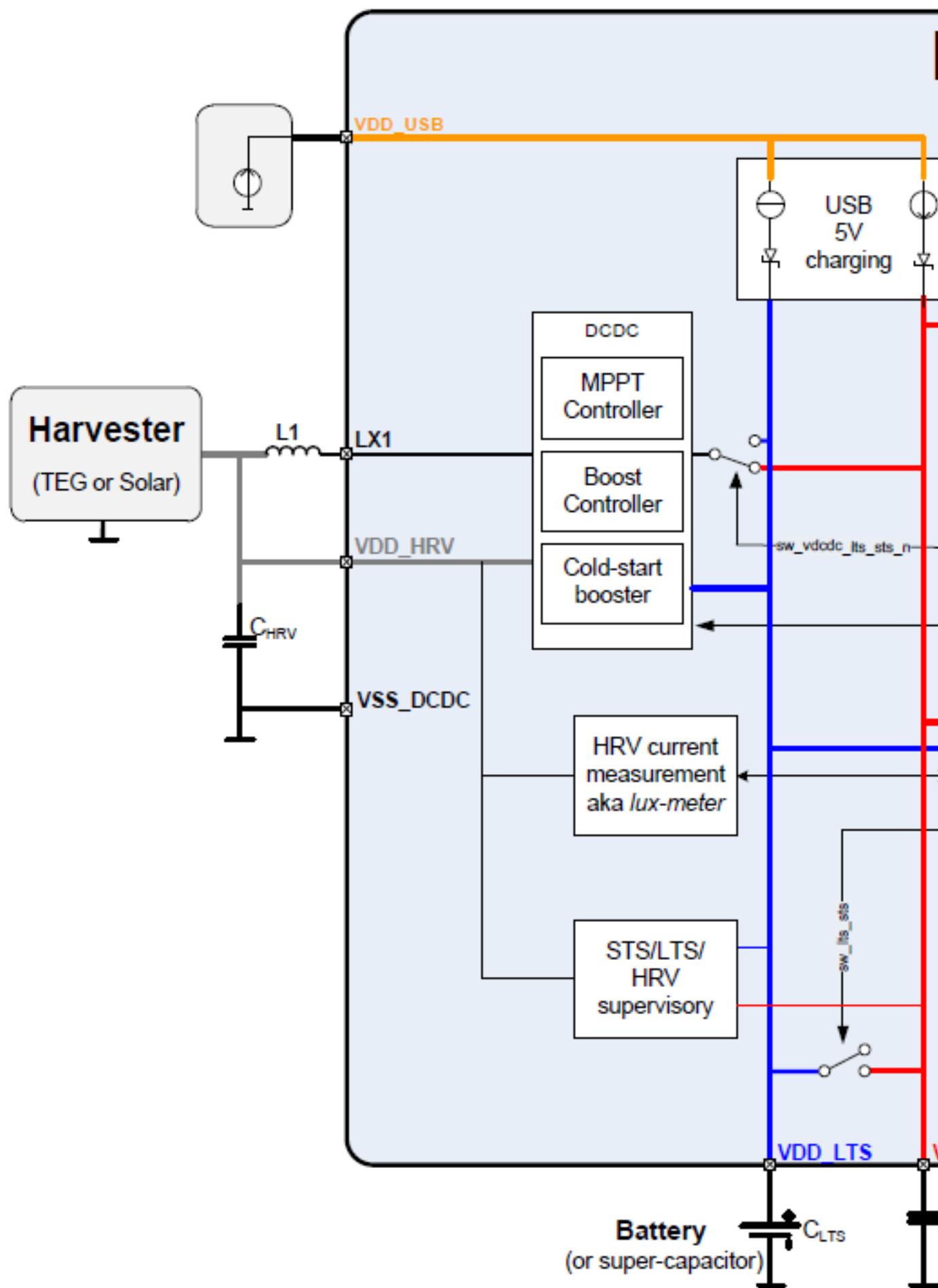


Abbildung B.1: Blockschema Sensortag

Anhang C

Funktionsblöcke Sensortag von Texas Instrument

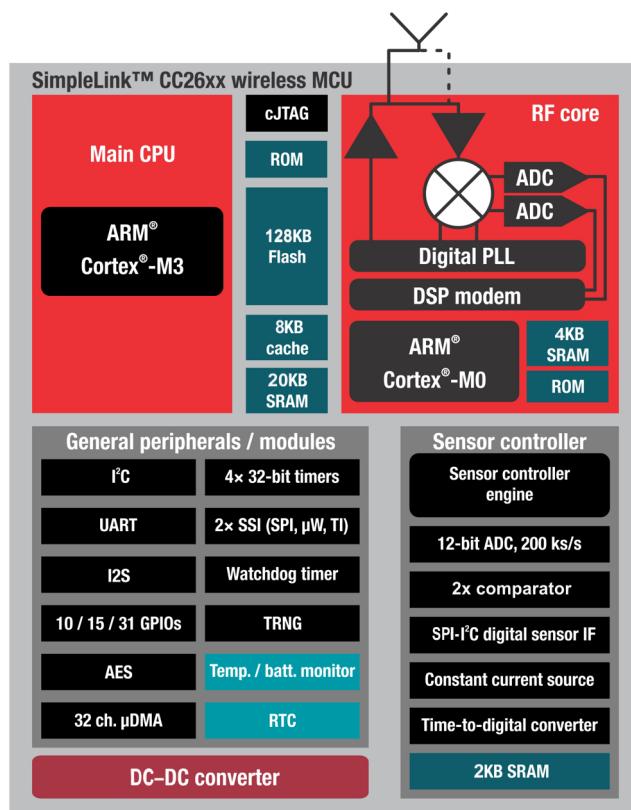


Abbildung C.1: Blockschema Sensortag aus ?, S.3

Anhang D

Messaufbau

Um reproduzierbare Werte zu erhalten, wurde eine Radimitation mit einem elektrischen Motor aufgestellt. Die Radimitation besitzt nur eine Speicheimitation. Dies ist eine Alustange, mit mehreren Löchern für das Anbringen des Magneten nach gewünschtem Abstand.

Bild.

Bild einfügen

Der Radimitation ist ein Tachometer angehängt, der die aktuelle Geschwindigkeit anzeigt.

Im Verlauf der Entwicklung wurde entschieden, dass zwei Magnete im Abstand von 180 Grad angebracht werden. Dies, damit genug Energie zur Verfügung steht (siehe Messprotokoll xxxxx).

Der Grundabstand ist 25 cm, was einen Radumfang von 2.04 m ergibt. Die ersten Messungen, siehe Anhang ?? beziehen sich auf diesen Radumfang. Die Messungen xxx beziehen sich auf einen Radumfang von yyyy.

Grad Zeichen
einbauen

Abstand un
Radumfang
kontrollieren

Anhang E

Übersicht Messprotokolle

Aufstellung von Manu

Messprotokol
Liste