Arbeitstitel: Eingliederung einer zusätzlichen Hardwarekomponetene in ein bestehendes Hexacopter System zur Leistungssteigerung

Malte Markus Breitenbach, Autor B, Autor C

***Zusammenfassung*— Hierhin kommt eine kurze (5-6 Sätze) Zusammenfassung der Arbeit. In diesem Fall beschreibt das Dokument die LaTeX-Vorlage für die Erstellung der Ausarbeitungen eines Projektseminars.**

***Abstract*— This is the english translation of your „Zusammenfassung “.**

I. Einführung

Ein Hexacopter stellt mit seinen zahlreichen Mess- und Stellgrößen und dynamischem Verhalten eine zugleich komplexe als auch herausfordernde Regelaufgabe dar.

Der digital ausgeführte Regler als solcher erfordert somit ein hohes Maß an Rechenleistung. Es ist daher erforderlich eine ausreichend leistungsfähige Rechenhardware zur Verfügung zu haben, was durch den im Firefly zur Verfügung stehenden High-Level-Prozessor nur bedingt gegeben ist. Sowohl der kleine Programmspeicher als auch die fehlende Rechengeschwindigkeit insbesondere bei Gleitkommaoperationen schränken den Regelungstechniker bei der Lösungsfindung stark ein. So muss beispielsweise durch den limitierten Programmspeicher auf lange Codesequenzen verzichtet werden und durch die fehlende Rechengeschwindigkeit zeitintensive Rechenoperatoren nur sparsam Anwendung finden. Abhilfe schaffen soll der in jeder Hinsicht performantere Microcontroller „Nucleo STM32F767ZI“(im Folgenden als Nucleo abgekürzt), der dem System als weiterer Rechenprozessor hinzugefügt werden soll. Um den neuen Prozessor sinnvoll in die Berechnungen miteinzubeziehen muss jedoch zunächst eine Schnittstelle zwischen dem zu leistungsschwachen High-Level-Prozessor und dem Nucleo hergestellt werden. Dabei gelten besondere Anforderungen an die Schnittstelle wie hohe Transferraten, ein hohes Maß an Robustheit als auch Echtzeitfähigkeit. Nur wenn diese Anforderungen auch ausreichend erfüllt werden, können später die Berechnungen praktisch auf dem Nucleo durchgeführt werden und der Hexacopter entsprechend geregelt werden.

Somit soll durch diese Arbeit eine solide Grundlage geschaffen werden, um darauf aufbauend eine Regelung für den Hexacopter entwerfen zu können.

II. Grundlagen

A. Systembeschreigung

Subsection text.

Diese Arbeit wurde von M.Sc. Raúl Acuña Godoy, Dipl.-Ing. Dinu Mihailescu-Stoica unterstützt.

1) UAV: Text
Text

2) Nucleo: STM32F767 Nucleo-144

Wie oben erwähnt, die Sensorfusion Algorithmus wird implementiert und überprüft in hinzugefügte Hochleistungsprozessor – „Nucleo STM32F767ZIT6“, die von STMicroelectronics entwickelt wird und wegen hoher Leistung heutzutage weit benutzt wird. Mikrocontroller enthaltet Prozessor und zugleich auch Peripheriefunktion, d.h. nicht nur Arbeits- und Programmspeicher sondern auch komplexe Peripheriefunktionen wie z.B. PWM-Ausgänge, CAN-, USB-, I2C, SPI (angewendet in unserem Protokoll) usw. werden komplett und kompakt auf demselben Chip integriert, was eine kostengünstige und flexible Möglichkeit für Benutzer bietet, neue Konzepte auszuprobieren und Prototypen zu erstellen.

Dank der kompletten Entwicklungswerkzeugen erleichtert man sehr die Entwicklungsarbeit. Einerseits, mit zahlreiche verpackte Bibliotheksfunktionen ermöglicht man schnelle Entwicklung verstärkt sehr die Lesbarkeit der Code, was ein wichtiger Grund die Popularität des STM32-Familien ist. Andererseits, mit GUI-Werkzeug wie CubeMX darf man einfach und sicher das System wie z.B. Konfiguration von dem ganzen Uhrbaum initialisieren.

Mikrokontrolle STM32767 in unsere Nucleo besitzt ARM Cortex-M7 Kern, der im maximal 216 MHZ arbeiten könnte. Zudem erhält er 2 MByte Flash-Speicher, 512Kbytes SRAM und 144 Pins, während der Stromverbrauch aber sehr gering – c.a. 250mA (Alle Peripheriegeräte aktiviert wird. TODO BibReference reference datasheet P124!!!) ist. Und Debugger/Programmierer-Werkzeug ST-LinkV2 ist schon implementiert. Ausreichende Peripheriegeräte werden implementiert, z.B. insgesamt vier I2C Schnittstellen sechs SPI Schnittstellen. SPI3, die von uns benutzt wird, könnte maximal als 25 Mbit/s Kommunikation Geschwindigkeit erreichen. Diese Geschwindigkeit ist schon viel größer als die Geschwindigkeit, welche HLP erreichen könnte.

3) Sensor Data:

Unsere Daten zur Protokoll und Sensorfusion basiert aus zwei Teilen:die integriert in UAV Sensoren, äußere Sensoren. Firefly besitzt wie normale UAV das Gyro, Kompass, GPS. etc. Sie sind genug für normale Regelung in normaler Situationen aber nicht gut wie z.B. Indoor-Situation. Außerdem ist Druckmesser nicht sehr präzise in Höhe Bestimmung. Um besser zu positionieren und Attitüde zu schätzen, wir werden noch wie z.B. Optischer Fluss, Kamera, usw. hinzufügen. Die innere Sensoren von UAV sind im folgende:

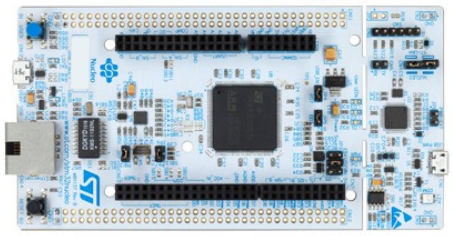

Bild 1. Nucleo

Bild 2. äußere Sensoren

- Gyro: Die zeige die lineare Beschleunigung und Winkelgeschwindigkeit.
- Kompass: Es messt das Magnetfeld der Erde.
- GPS: Durch GPS wird z.B. globale Position, Geschwindigkeit und Zeit.
- Pressuresensor: Die Höhe könnte im Natur damit bestimmen.

Die äußere Sensoren von UAV sind im folgende:

- Optischer Fluss: GPS könnte nicht innerhalb des Gebäudes arbeiten, während optische Fluss ist eine gute Ersetzung. Wir benutzen PX4FLOW, die erfolgreich als Position System GPS ersetzt hat. (Ref)
- LIDAR-Lite v3. Für Indoor-Höhe-Bestimmung drinnen. Es ist viel schneller als Ultraschall und robust, zuverlässig.

Außerdem werden noch die anderen Sensoren spät hinzugefügt. z.B. Kamera, Microsoft Kinect, usw. um besser die verschiedenen Situationen anzupassen. Die Kapazität unseres Protokolls ist noch ausreichend für die Hinzufügung.

B. Serial Peripheral Interface (SPI)

Das Serial Peripheral Interface (kurz SPI) ist ein im Jahr 1987 von Susan C. Hill Et al., damals bei dem Halbleiterhersteller Motorola (heute NXP Semiconductors), entwickeltes Bus-System und stellt einen „lockeren “Standard für einen synchronen seriellen Datenbus (Synchrones Serial Port) dar, mit den digitalen Schaltungen nach dem Master-Slave-Prinzip miteinander verbunden werden können(BibReference.)

SPI benutzt nicht Adresse zum Slave Auswahl aber einfach einen Stift als „Chip Select “. SPI ist ein wirklich vollduplexfähiger Bus und hat hohe Kommunikationsgeschwindigkeit im Vergleich andere häufige angewendete serielle Bus wie USART, I2C. Die typische Verbindung von SPI ist wie Bild 3 gezeigt(Sternverbindung). Auch ist es gezeigt, dass

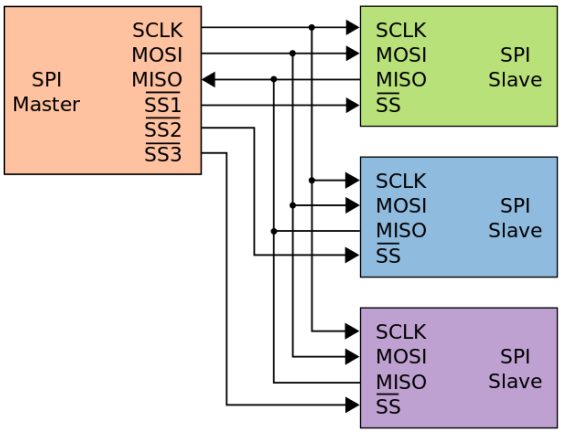

Bild 3. SPI-Verbindung durch Kaskadierung der Slaves

jede Master-Slave Kommunikation im General vier Leitungen angefordert. Sind jeweils:

- SCLK (Serial Clock) auch SCK, wird vom Master zur Synchronisation ausgegeben
- MOSI (Master Output, Slave Input) oder SIMO (Slave Input, Master Output)
- MISO (Master Input, Slave Output) oder SOMI (Slave Output, Master Input)
- $\overline{SS}$ (Slave Select), oder $\overline{CS}$ (Chip Select).

Der Protokollablauf ist wie Bild 4 gezeigt. Legt Master mit der Leitung „Slave Select “fest, mit welchem Slave er kommunizieren will, ist der jeweilige Slave aktiv. Es wird ein Wort vom Master zum Slave und gleichzeitig ein anderes Wort vom Slave zum Master transportiert.

Die Polarität von Clock und Datenübertragung sind nicht bei Motorola definiert. Sie werden durch „CPHA“, der Clock Idle definiert, und „CPOL“, der definiert, in welche Flanken der Chip Daten übernommen, festgelegt.

Mit jeder Taktperiode wird ein Bit übertragen. Beim üblichen Bytetransfer sind also acht Taktperioden für eine vollständige Übertragung nötig. Master verschiebe durch MOSI ein Bit aus dem Register und übernimmt durch MISO ein Bit ins diese Register. Slave benutzt aber MISO und MOSI. Eine vollständige Übertragung ist den Austausch zwischen den Registern von Master und Slave. Es können auch mehrere Worte hintereinander übertragen werden. Eine Übertragung ist beendet, wenn das Slave-Select-Signal endgültig auf High gesetzt wird.

C. Serial Peripheral Interface (SPI)

Subsubsection text.

D. Protokolltheorie

Um eine Kommunikation zwischen den zuvor erwähnten Microcontrollern zu ermöglichen, bedarf es eines gut geplaneten sowie gut umgesetzen Kommunikationsprotokolls.

Ein Kommunikationsprotokol ist nach [1] nichts anderes als eine Verhaltenskonvention zwischen zwei Kommunikationspartern (oft auch als Instanzen bezeichnet). Durch diese

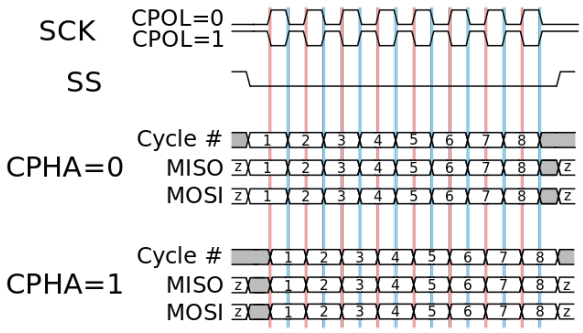

Bild 4.

Verhaltenskonvention wird einerseits der zeitliche Ablauf der Kommunikationsaktionen zwischen den Partnern definiert. Andererseits wird auch die Form der zu übermittelnden Nachrichten eindeutig festgelegt. Diese Nachrichten werden auch als PDUs (protocol data units) bezeichnet und ihre Struktur muss bei beiden Kommunikationspartnern bekannt sein, damit sie identisch interpretiert werden. Damit ein Protokoll erfolgreich zwischen zwei Partnern ausgeführt werden kann, müssen besagte Partner über bestimmte Protokollfunktionen verfügen. Dazu zählt bspw. die PDU-Codierung oder Fehlerkontrolle.

Einen wichtiger Begriff in der Protokolltheorie stellen die sogenannten Schichten dar, durch die sich die Funktionalität eines Protokolls veranschaulichen lässt. Eine Schicht umfasst dabei vereinfacht ausgedrückt alle Teilfunktionalitäten, die zur Erfüllung einer bestimmten Zielstellung herangezogen werden. Während der Protokollausführung interagieren besagte Schichten miteinander, wobei jede Schicht (N) die Funktionalitäten der nächsttieferen Schicht (N-1) verwendet. Dabei ist für die höhere Schicht irrelevant, wie die benötigte Funktionalität genau realisiert wird. Als Beispiel für eine Schicht wäre die *physikalische Bitübertragungsschicht* als unterste Schicht oder die darüberliegende Datensicherungsschicht anzusehen. Auf beide Schichten wird später noch genauer eingegangen. Die nachfolgende Abbildung veranschaulicht die Generierung einer PDU über mehrere Schichten.

III. Zusammenfassung

Hier die wichtigsten Ergebnisse der Arbeit in 5-10 Sätzen zusammenfassen. Dies sollte keine Wiederholung des Abstracts oder der Einführung sein. Insbesondere kann hier ein Ausblick auf zukünftige Arbeiten gegeben werden.

Anhang I
Optionaler Titel

Anhang eins.

Anhang II

Anhang zwei.

Anhang III
Richtlinien für das Verfassen wissenschaftlicher Arbeiten

Im Folgenden werden einige wichtige Richtlinien zusammengefasst. Die Aufzählung ist allerdings nicht erschöpfend.

- Klare Darstellung, was der Eigenanteil ist und was schon vorhanden war.
- Vorsicht vor Plagiaten: vollständige Quellenangaben, auch bei Bildern. Es sollte immer klar ersichtlich sein, was der Eigenanteil ist und was aus Quellen entnommen wurde.
- Bilder nicht 1:1 aus Quellen kopieren.
- Diskussion der Ergebnisse (Simulationen, Messungen, Rechnungen): Wurde das Ergebnis so erwartet? Wenn nein, was sind mögliche Gründe?
- Autoren: Als Autor sollte jede Person in Betracht gezogen werden, die wesentlich zur Arbeit beigetragen hat (siehe auch die Empfehlungen der DFG diesbezüglich, vgl. [2]). Alle Personen mit kleinerem Beitrag (fachliche Hinweise, Beteiligung an Datensammlung etc.) können in der Danksagung oder einer Fußnote erwähnt werden.
- Formeln in den Satz einbetten und alle Variablen bei der ersten Verwendung im Text einführen. Beispiel: Für die Temperatur ergibt sich damit

$$T(h) = Kh^2,$$

  sie hängt quadratisch von der Höhe h ab.

Anhang IV
Hinweise zur Notation

- Abkürzungen bei der ersten Verwendung erklären, z.B.: “DFG (Deutsche Forschungsgemeinschaft)”.
- Formelzeichen konsistent benennen, nicht zwischen den Abschnitten umbenennen. Formelzeichen kursiv schreiben, z.B. Variable a.
- Auf korrekte Dimensionen und Einheiten achten. Für Einheiten das SI-System verwenden, z.B. das LaTeX-Paket *units* oder *SIunits*.
- Zahlen: Im Deutschen Komma als Dezimaltrennzeichen, im Englischen Punkt.
- Tabellen haben Überschriften, Diagramme haben Unterschriften.
- Diagramme: Achsenbeschriftungen hinreichend groß (insbesondere die Zahlen).
- Diagrammunterschriften sollen im Wesentlichen ausreichen, um das Diagramm zu verstehen.
- Indizes werden *kursiv* gesetzt, wenn sie die Bedeutung von Variablen haben, ansonsten **normal**. Beispiele: $V_k,\ k = 1, 2, \ldots$ und V_{input}.

Danksagung

Wenn ihr jemanden danken wollt, der Euch bei der Arbeit besonders unterstützt hat (Korrekturlesen, fachliche Hinweise,...), dann ist hier der dafür vorgesehene Platz.

LITERATURVERZEICHNIS

[1] H. König, *Protocol Engineering*, 1st ed. Wiesbaden, Deutschland: Teubner, 2003.
[2] Deutsche Forschungsgemeinschaft, *Vorschläge zur Sicherung guter wissenschaftlicher Praxis*, Denkschrift, Weinheim: Wiley-VCH, 1998.

Autor A Biographie Autor A.

Autor B Biographie Autor B.

Autor C Biographie Autor C.