

L^AT_EX
HSLU Elektrotechnik Master
Template, Grundlagen, Tipps, Vorlagen

Stefanie Schmidiger

MASTER OF SCIENCE IN ENGINEERING

Vertiefungsmodul I

Advisor: Erich Styger

Experte: Der Experte

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche verwendeten Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Horw, 10.01.2017

Stefanie Schmidiger

Versionen

Version 0 Vorabzug

01.09.14 Stefanie Schmidiger

Vorwort

This work is being done for Aeroscout GmbH, a company that specialized in development of drones. With unmanned vehicles, there are always on-board and off-board components. Data transmission between those components is of vital importance. Depending on the distance between on-board and off-board components, different data transmission technologies have to be used.

In this project, a hardware has been designed where multiple data inputs and outputs and multiple transmitters can be connected to a serial switch. The designed hardware features an SD card with a configuration file where data routing can be configured.

Data from connected devices will be collected and put into a data package with header, checksum, time stamp and other information. The package is then sent out via the configured transmitter. The corresponding second serial switch hardware receives this package, extracts and checks the payload, sends it out to the corresponding device and sends an acknowledge back to the package sender.

When data transmission over one transmission technology fails, the configuration file lets the user select the order of back up transmitters to be used. Data priority can also be configured because reliability of data transmission is extremely important with information such as exact location of the drone but not as important with information such as state of charge of the battery.

The serial switch hardware designed in the scope of this project features four serial RS232 connections where input and output devices can be connected that process or generate data. There are also four RS232 connectors where transmitters can be connected to send or receive data packages. The routing between data generating devices and transmitters to use can be done in a .ini file saved on an SD card.

There are two SPI to UART converters that act as the interface between the four devices connected and the micro controller respectively the four transmitters and the micro controller.

In a first version of the project, a Teensy 3.1 development board has been used as a micro controller unit. The software was written in the Arduino IDE with the provided Arduino libraries. As the project requirements became more complex, the limit of only a serial interface available as a debugging tool became more challenging. In the end, the first version of the software ran with more than ten tasks and an overhaul of the complex structure was necessary.

For this reason, an adapter board has been designed so the existing hardware could be used with the more powerful Teensy 3.5. This adapter board features a SWD hardware debugging interface that was ready to use after removing a single component on the Teensy 3.5 development board.

The Teensy 3.5 was then configured to run with FreeRTOS. Task scheduler and queues provided by this operating system have been used to develop software that extracts data from received packages to output them on the configured interface or generates packages from received data bytes to send them out over the configured transmitter. The concept of acknowledges has also been applied so package loss can be detected and lost packages can be resent.

The software concept implemented is easy to understand, maintainable and expandable. Even though the functionality of the finished project remains the same as in the first version with Teensy 3.1 and Arduino, a refactoring has been necessary. Now further improvements and extra functionalities can be implemented more easily.

Kurzfassung

Hier wird der gesamte Text der Kurzfassung eingefügt.

Inhaltsverzeichnis

1 Task Description	1
2 Starting Situation	3
2.1 Existing Hardware	3
2.1.1 Serial Interfaces	4
2.1.2 RS232 to UART Converter	4
2.1.3 USB Interface	5
2.1.4 SPI to UART Converter	5
2.1.5 Teensy 3.1 Development Board	5
2.1.6 Power Supply	5
2.2 Software	5
3 Hardware	11
3.1 To-Do List for next version of UAV serial switch	11
4 Software	13
4.1 Analysis of old Software	13
4.2 New Software	13
5 Hyperref	15
5.1 Backref	15
5.2 Autoref	15
6 Literaturverweise	17
6.1 Bibliography und Zotero	17
Anhang A Anhangstruktur	19
A.1 Unterkapitel im Anhang	19
A.1.1 Tieferes Kapitel	19
Literaturverzeichnis	21
Bezeichnungen	23

1 Task Description

This project has been done for the company Aeroscout. Aeroscout specialized in the development of drones for various needs.

With unmanned aerial vehicles, the communication between on-board and off-board devices is essential and a reliable connection for data transmission is necessary. While the drone is within sight of the control device, data can be transmitted over a wireless connection. With increasing distances, other means of transmission have to be selected such as GPRS or even satellite.

So far, the switching between different transmission technologies could not be handled automatically. The data stream was directly connected to a modem and transmitted to the corresponding receiver with no way to switch to an other transmission technology in case of data transmission failure. A visualization of this set up can be seen from Bild 1.1

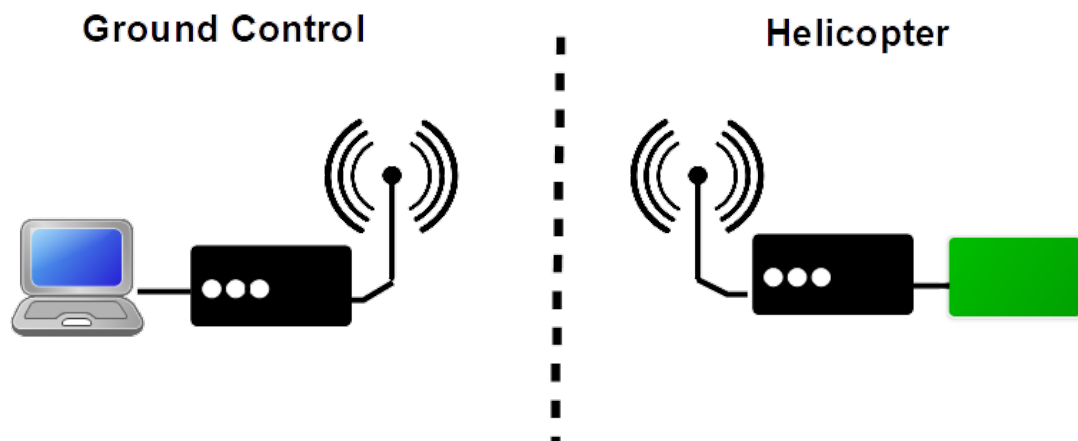


Bild 1.1: Previous system setup for data transmission

The aim of this project is to provide a solution that provides the needed flexibility. The finished product should act as a serial switch with multiple input/output interfaces for connecting devices and sensors and multiple interfaces for connecting transmitters. When one transmission technology fails to successfully transmit data, an other technology can be chosen for the next send attempt. Also, multiple sensors or input streams should be able to send out data over the same wireless connection. A visualization of this set up can be seen in Bild 1.2

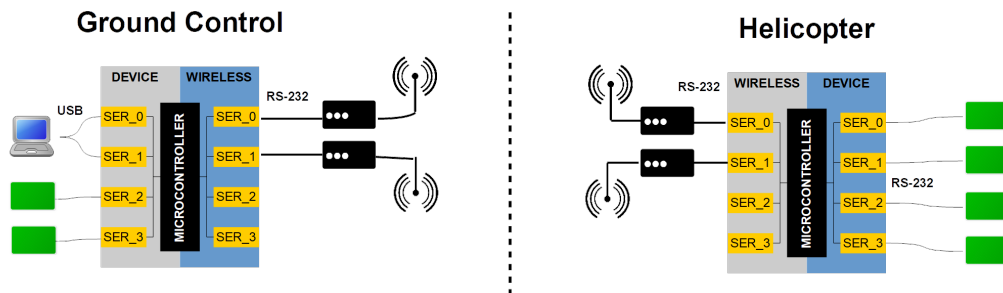


Bild 1.2: New system setup for data transmission

There are various kinds of information exchanged between drone and control device such as state of charge of the battery, exact location of the drone, control commandos etc. Some information such as the exact location of the drone should be prioritized over battery status information when data transmission becomes unreliable. The finished product should therefore take data priority into account. Encryption should be configurable individually for each interface in case sensitive data is exchanged over a connection.

The finished product should have a debugging interface such as SWD and a shell/command line interface. During run time, the software should log system data and any other relevant information to a file saved on an SD card. The SD card should also contain a configuration file so the behavior of the hardware can be changed easily.

A detailed description of all the requirements can be taken from the appendix Aufgabenstellung.

Link
zur
Auf-
gaben-
stel-
lung
im Ap-
pendix

2 Starting Situation

It was not necessary to start from scratch for this project.

In the beginning of 2017, Andreas Albisser has already started with an implementation and provided a first solution.

He developed a hardware that was used as the interface between input/output data and modem for wireless transmission. He chose the Teensy 3.1 development board as a micro controller and worked with the Arduino IDE and Arduino libraries.

There are various problems still with his work which lead to this follow up project to improve the overall functionality.

More details about the work Andras Albisser has done can be taken from this chapter.

2.1 Existing Hardware

The hardware developed by Andreas Albisser has a total of eight interfaces where peripheral devices can be connected. Four connections are for control units, sensors or any other devices that process or generate data to be transmitted. On the other side, there are four connections where modems can be connected to allow different ways of transmission. An overview can be seen in Bild 2.1.

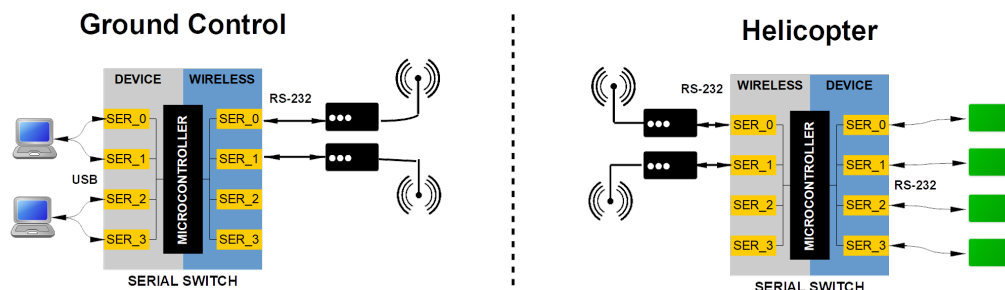


Bild 2.1: Hardware overview

Each interface accessible to the user is bidirectional which allows for more flexibility.

From now on, the side where data generating and processing devices can be connected will be referred to as the device side and the side where modems can be connected will be referred to as the wireless side.

On both device side and wireless side, periphery can be connected to the four UART serial interfaces. On device side, the user can choose between a UART interface and a USB mini interface individually for each interface with jumpers. When selecting the USB mini interface, one USB hub acts as a dual COM interface, allowing two serial COM ports to open up to simulate two serial interfaces.

The serial interfaces are not connected to the Teensy 3.1 development board directly. There is a SPI to UART converter that acts as a hardware buffer between serial input/output and microcontroller. All serial connections work on RS232 level which is $\pm 12V$. Because the SPI to UART converter is not RS232 level compatible, a voltage regulator is used between the serial interface accessible to the user

and the SPI to UART converter.

Details about the components used on this hardware can be taken from the following section. A block diagram of the on-board hardware components can be taken from Bild 2.2.

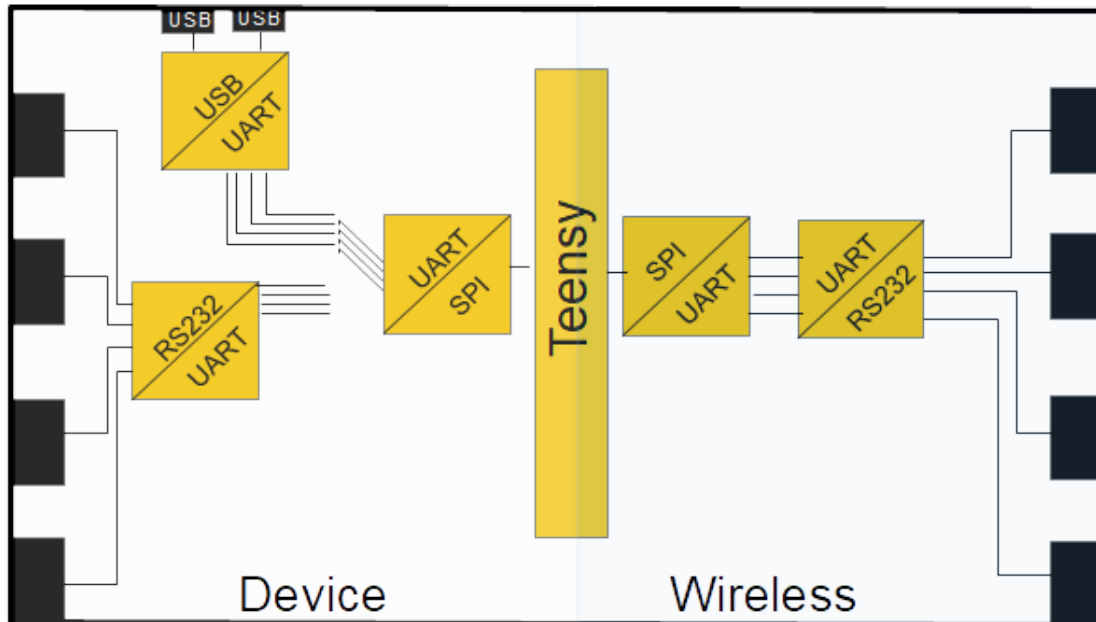


Bild 2.2: Hardware details

2.1.1 Serial Interfaces

There are a total of eight UART serial connections accessible to the user, four on device side and four on wireless side.

The baud rate for each serial connection can be configured individually.

UART is an asynchronous serial interface which means that there is no shared clock line between the two components. Both sides need to be configured with the same baud rate so they can communicate correctly.

A UART interface requires three wires: two unidirectional data lines (RX and TX) and a ground connection.

2.1.2 RS232 to UART Converter

The serial interfaces accessible to the user work on RS232 level. Just behind the serial interface, there is a level shifter that converts the RS232 level to TTL (5V).

This level shifter is bypassed on the device side in case the USB serial connection is used instead of the RS232 serial interface.

2.1.3 USB Interface

On device side, the user can chose whether data is provided via USB or via RS232 serial connection. A jumper is used to switch between RS232 input and USB input.

In case when the USB input is selected, each USB hub acts as a dual serial COM port which means that when connecting the hardware to a computer, there will be two COM ports available per USB connection.

The on board USB to UART converter acts as an interface between USB hub and SPI to UART converter.

2.1.4 SPI to UART Converter

UART is an asynchronous serial interface which requires three connections: ground and two unidirectional data lines. If the teensy was to communicate to each serial port directly, it would require eight of those UART interfaces (which would add up to 16 data lines). To facilitate communication to the serial interfaces, a SPI to UART converter was selected as an intermediate interface.

There are two SPI to UART converters on board, one for the four device serial connections and one for the four wireless serial connections. SPI is a synchronous master-slave communication interface where the unidirectional data lines are shared amongst all participants. The only individual line between master and slave is the Slave Select line that determines, which slave is allowed to communicate to the master at a time.

Those converters are used as hardware buffers and can store up to 128 bytes.

2.1.5 Teensy 3.1 Development Board

Andreas Albisser used a Teensy 3.1 as a micro controller.

The Teensy development boards are breadboard compatible USB development boards. They are small, low-priced and yet equipped with a powerful ARM processor.

The Teensy development boards all come with a pre-flashed bootloader to enable programming over USB. They use a less powerful processor as an interface to the developer to enable the use of Arduino libraries and the Arduino IDE.

2.1.6 Power Supply

The hardware needs 5V as a power supply. This can be achieved by using any of the USB connections or via a dedicated power connector located on the board.

2.2 Software

The software written by Andreas Albisser provided a good basis and reference for the software developed in the scope of this project.

The basic functionality provided by his software was the transmission of data packages on wireless

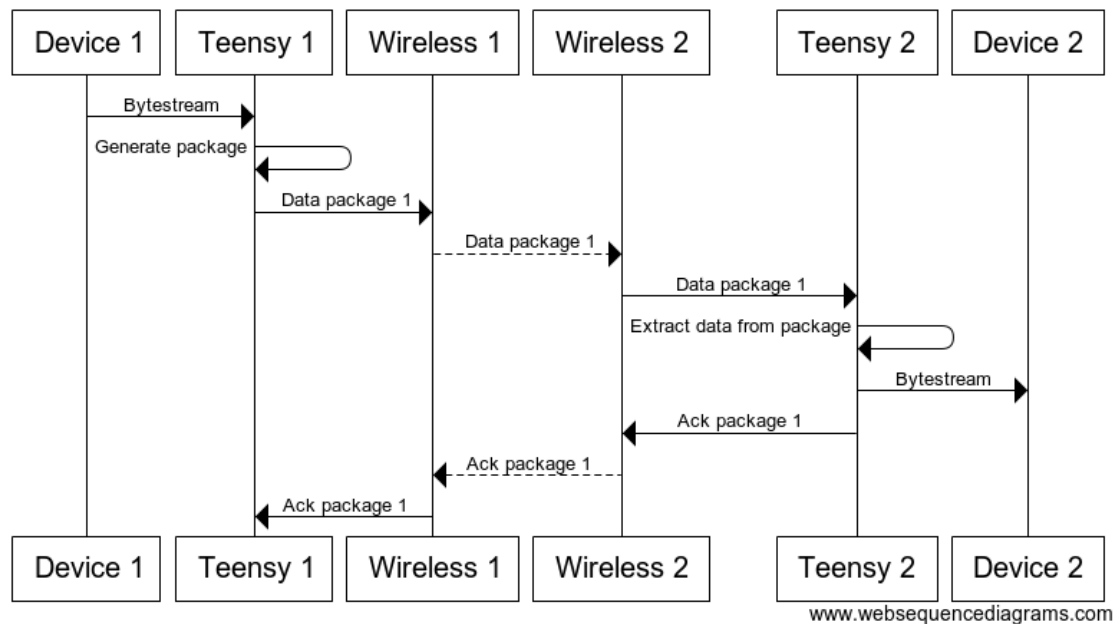


Bild 2.3: Successful package transmission

side.

The Teensy would frequently poll the SPI to UART hardware buffer for received data. In case the SPI to UART converter had data in its buffer, the Teensy would read the data in a second SPI command. The read data would then be wrapped in a package with header which contained CRC, timestamp and other information and sent out on the wireless side.

The corresponding second hardware would receive this package on its wireless side, extract the payload from it and send the extracted payload out on its device side.

To ensure successful transmission of packages, the concept of acknowledges was applied in the software where the receiver replies with an acknowledge to a successful package reception. A sequence diagram of a successful package transmission can be found in Bild 2.3.

The maximum number of payload bytes per package can be configured in the software, just like the maximum amount of time the application should wait for a package to fill up until it will be sent anyway.

In case the package transmission was unsuccessful, either if the package got lost or corrupted, the receiving hardware will not send an acknowledge back. The application that sent the package will wait for a configurable amount of time before trying to send the same package again. Details can be found in figure Bild 2.4.

The maximum time to wait for an acknowledge before resending the same package can be configured in the software. The maximum number of resends per package can be configured for each wireless connection.

All configuration possibilities of Andreas Albissers software can be taken from the Tabelle 2.1:

Configuration parameter	Possible values	Description
BAUD_RATES_WIRELESS_CONN	9600, 38400, 57600, 115200	Baud rate to use on wireless side, configurable per connection. Example: 9600, 38400, 57600, 115200 would be used for wireless connection 0, 38400 baud for wireless connection 1 etc.

BAUD_RATES_DEVICE_CONN	9600, 38400, 57600, 115200	Baud rate to use on device side, configurable. Example: 9600, 38400, 57600, 115200 would mean 9600 baud for device connection 0, 38400 baud for device connection 1, 57600 baud for device connection 2, 115200 baud for device connection 3.
PRIO_WIRELESS_CONN_DEV_X	0, 1, 2, 3, 4	This parameter determines over which wireless connection a stream of a device will possibly be sent out. Connection 0 will not be used. 1: Highest priority, data will be sent out over this connection first. 2: Second highest priority, data will be tried to send out over this connection should the first priority connection fail. 3: Third highest priority, data will be sent out over this connection if the first two connections fail. 4: Lowest priority for data transmission. Example: 1, 2, 3, 4 would mean that in data being sent out over wireless connection 0, data will be sent out over wireless connection 1 in case of failure of connection 0. Connections 2 and 3 would not be used. Replace the X in the parameter name with 0, 1, 2 or 3.
SEND_CNT_WIRELESS_CONN_DEV_X	0...255	Determines how many times a package should be sent out over a wireless connection before moving on to the next lower priority wireless connection. Example: 5, 4, 3, 2 would result in the package being sent out over connection 0 five times and four times over wireless connection 1. Replace the X in the parameter name with PRIO_WIRELESS_CONN_DEV_X, then the number of resends per connection. Replace the X in the parameter name with 0, 1, 2 or 3.
RESEND_DELAY_WIRELESS_CONN_DEV_X	0...255	Determines how many milliseconds the software will wait for an acknowledge per wireless connection before resending the package again. Example: 10, 0, 0, 0 would mean that the software will wait for an acknowledge for 10ms when sending data out via wireless connection 0 before attempting to send it out via wireless connection 1. Replace the X in the parameter name with PRIO_WIRELESS_CONN_DEV_X, then the delay of the resend behaviour. Replace the X in the parameter name with 0, 1, 2 or 3.
MAX_THROUGHPUT_WIRELESS_CONN	0...4294967295	Limit of the maximum data throughput in bytes per second for a wireless connection. If two devices use the same wireless connection but the same priority but the maximum throughput is different, the lower priority device will be redirected to the next lower priority or discarded (if no other connection with the next lower priority or discarded (if no other wireless connection with lowest priority already exists). Example: 10000, 10000 means that wireless connection 0 and 1 have a maximum throughput of 10000 bytes per second.
USUAL_PACKET_SIZE_DEVICE_CONN	0...512	Maximum number of payload bytes per wireless connection. If the known payload, the PACKAGE_GEN_MAX_PACKET_SIZE parameter always determines the payload size. Example: 128 would result in a maximum payload of 128 bytes. If the payload is unknown maximum payload size for wireless connection 0, 1, 2, 3.
PACKAGE_GEN_MAX_TIMEOUT	0...255	Maximum time (in milliseconds) that the software will wait for a package to fill up before sending it out over a wireless connection. USUAL_PACKET_SIZE_DEVICE_CONN determines the size of a package. Example: 50, 50 would mean that data being sent out after a maximum wait time of 50ms.

DELAY_DISMISS_OLD_PACK_PER_DEV	0...255	Maximum time (in milliseconds) an old package is kept before being discarded. Example: 5, 5, 5, 5 results in old packages being discarded 5ms after the next package is available for sending. Example: 5, 5, 5, 5 results in old packages being discarded 5ms after the next package is available for sending. Example: 5, 5, 5, 5 results in old packages being discarded 5ms after the next package is available for sending.
SEND_ACK_PER_WIRELESS_CONN	0, 1	Acknowledges turned on/off for each wireless connection. Example: 1, 1, 0, 0 results in acknowledges being expected for wireless connection 0 and 1 but not over wireless connection 2 and 3.
USE_CTS_PER_WIRELESS_CONN	0, 1	Hardware flow control turned on/off for each wireless connection. Example: 1, 1, 0, 0 results in hardware flow control being turned on for wireless connection 0 and 1 only.

Tabelle 2.1: Configuration parameters of the old software

Talk about what the teensy was able to do in June 2017 and about the configuration possibilities.

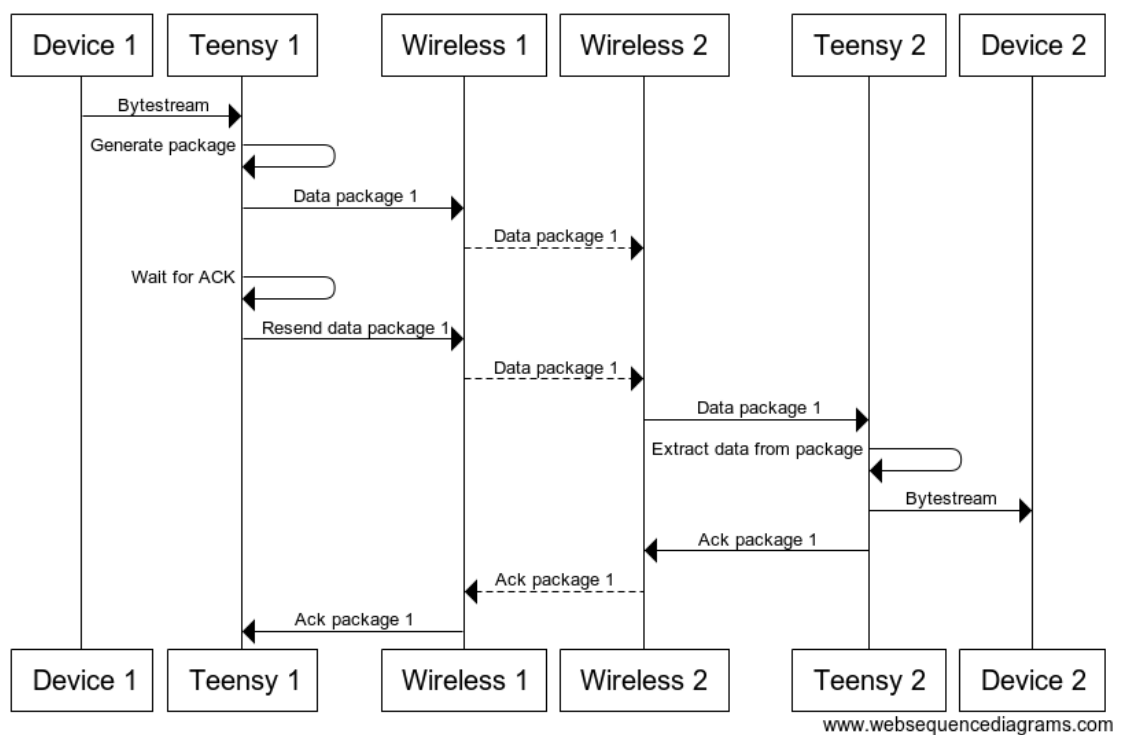


Bild 2.4: Unsuccessful package transmission

3 Hardware

Talk about teensy adapter board.

3.1 To-Do List for next version of UAV serial switch

List all the changes that are needed on the UAV base board

HW Flow control can't be done without changing HW because lines are RS232 level!

4 Software

4.1 Analysis of old Software

Pro and cons about old SW

4.2 New Software

Talk about new SW concept

To-Do List for new SW

Talk about the things that have not yet been implemented Package numbering instead of waiting for ACK

5 Hyperref

Ein mit der vorliegenden Vorlage erstelltes L^AT_EX-Dokument enthält zahlreiche Hyperref-Links. Dies bedeutet, dass sämtliche Verweise im PDF auch direkt als Link fungieren. Klickt man den Verweis an, landet man auf der entsprechenden Seite. Hyperref-Links funktionieren für:

- Bilder
- Tabellen
- Überschriften
- Gleichungen
- Codeblöcke
- Quellenverweise

5.1 Backref

In den Quellenverweisen ist jeweils die Information enthalten, auf welchen Seiten auf diese Quelle verwiesen wird (Zitiert auf Seite ...). Auch diese Verweise funktionieren wiederum als Links im PDF.

5.2 Autoref

Bei Verweisen, welche mit der Funktion „\autoref“, bzw. „\aref“ oder „\autoeqref“ formuliert werden, wird beispielsweise der Begriff „Bild“ automatisch dazugeschrieben. Dabei fungiert zudem im PDF nicht nur die Zahl als Link, sondern auch der ganze dazugehörige Begriff (z.B. ?? anstatt nur ??).das funktioniert natürlich auch mit:

??

??

Kapitel 5

Abschnitt 5.1

??

Im Fall von Anhängen wird der „\aref{ }“-Befehl (*a* für appendix oder Anhang) benötigt:

Anhang A

Anhang A.1

Bei Formeln kann der Befehl „\autoeqref“ angewendet werden:

Gleichung (??)

Tipp

- Labels mit einem Kürzel beginnen, das Auskunft darüber gibt, auf welche Art von Textbaustein es verweist (z.B. „picBeispiel“ für Bild, „tabBeispiel“ für Tabelle, „eqBeispiel“ für Gleichung, „refBeispiel“ für Überschriften).

6 Literaturverweise

6.1 Bibliography und Zotero

Die Einträge im Bibliography-File können mit Zotero erstellt werden. Wenn die entsprechende Literatur dort bereits eingetragen ist, kann sie einfach per Drag-and-Drop in das BibLaTeX-Literaturfile gezogen werden. Als Alternative kann per Rechtsklick auf die Datei über den Befehl “ausgewählten Eintrag exportieren” ein neues BibLaTeX-File mit dem Eintrag erstellt werden. Dies funktioniert auch, wenn mehrere Dateien angewählt sind.

Bei MSE-Berichten sind sämtliche Literaturstellen in der Zotero-Datenbank abzulegen. Zum Eintragen der benötigten Attribute (Titel, Autor etc.) kann Tabelle 6.1 konsultiert werden. Folgende sind Punkte zu beachten:

- **Bevor man bei Zotero eine Literaturstelle hinzufügt, ist zu prüfen, ob diese bereits existiert.** Allfällig bemerkte doppelte Einträge werden fusioniert.
- Der Name der heraufgeladenen PDF-Datei soll dem Schema „Jahr - Autor - Titel“ folgen. Also zum Beispiel „2009 - Seelhofer - Ebener Spannungszustand im Betonbau.pdf“. Bei MSE-Dokumenten schreibt man zusätzlich die das Modul dazu, also beispielsweise: „2013 - Stenz - VM2 - Kontinuierliche Spannungsfeldmodelle.pdf“.
- Beim Eintrag einer Literaturstelle in Zotero ist unter „Datum“ immer nur das Jahr einzutragen, Ausnahme: Zeitschriftenartikel (dort wenn vorhanden den Monat auch berücksichtigen).
- Bei Vertiefungsmodulen ist unter „Art des Berichtes“ der Eintrag „Bericht Vertiefungsmodul 2“ zu machen. Der Zusatz „Bericht“ wird im Hinblick auf die Zitierung in \LaTeX der Verständlichkeit halber benötigt.
- Beim Literaturtyp „Bericht“ werden in Zotero „Seiten“ (von-bis) und nicht die „Anzahl der Seiten“ verlangt. Meistens soll im Literaturverweis aber „123 S.“ (Seitenanzahl) und nicht „S. 123-127“ (gewisse Seiten eines Dokuments) stehen. Die erste Darstellung kann erzwungen werden, wenn in Zotero im Feld „Seiten“ der Eintrag „123 S.“ und nicht nur „123“ gemacht wird. Letzterer Eintrag würde zur meist unerwünschten Darstellung „S. 123“ im Literaturverzeichnis führen.
- Um in \LaTeX auf eine aus Zotero exportierte Literaturstelle zu verweisen, wird im Argument des `\cite`-Befehl folgendes Muster verlangt: „Autor“_ „1.Wort des Titels“ _ „Jahr“ . Beispiel: Auf „Ebener Spannungszustand im Betonbau“ von Seelhofer (2009) wird mit `\cite{seelhofer_ ebener_ 2009}` zitiert.
- Achtung: In Zotero zusätzlich eingegebene Informationen (übrige, unbenutzte Felder) können unter Umständen auch in \LaTeX im Literaturverzeichnis erscheinen (z.B. wenn bei einem Buch der ISBN eingegeben wird, wird dieser am Ende des Verweises im Literaturverzeichnis aufgeführt).
- Die Argumente „@keywords“ und „@file“ in BibLaTeX-Literaturdatenbanken entstehen automatisch beim Export aus Zotero und haben keinen Einfluss auf den Output im Literaturverzeichnis. Sie können also in der Datenbank belassen werden.
- Bei Zeitschriftenartikeln muss bei Verweisen keine Seitenangabe gemacht werden, z.B. [10]. In allen anderen Fällen muss die Seitenzahl, von der die Information aus der Quelle entnommen wurde, angegeben werden, z.B. [11, S. 34] mit `\cite[S. 34]{seelhofer_ ebener_ 2009}`

Literaturtyp	Typ Zotero	Typ \LaTeX	Attribute															
			Titel title	Autor author	Nr. Bericht number	Art Bericht type	Ort location	Institution institution	Seiten pages	Anz. Seiten pagetotal	Datum year	Verlag publisher	Name Konf. eventtitle	Band volume	Ausgabe issue /number	Publikation journaltitle		
Bericht [5]	Bericht	report	X	X	Nr. 75	Bericht	X	X	000 S.		Jahr							
Buch [17]	Buch	book	X	X					000		Jahr	X						
Dissertation [11]	Dissertation	thesis	X	X			X	X	000		Jahr							
Diskussionsbericht [6]	Bericht	report	X	X	Nr. 124	Diskussionsbericht	X	X	000 S.		Jahr							
Konferenz-Paper, -bericht [14]	Konferenz-Paper	inproceedings	X	X					00-00		Jahr		X	X				
MSE Master-Thesis [1]	Bericht	report	X	X		Master-Thesis	X	X	000 S.		Jahr							
MSE Bericht VM1, VM2 [2]	Bericht	report	X	X		Bericht Vertiefungsmodul 1	X	X	000 S.		Jahr							
Norm [4] [8] [13]	Bericht	report	X				X	X	000 S.		Jahr							
Norm Dokumentation [12]	Bericht	report	X				X	X	000 S.		Jahr							
Anleitung / Manual [15]	Bericht	report	X	X		Anleitung (o.ä.)	X		000 S.		Jahr							
Versuchsbericht [3] [9]	Bericht	report	X	X		Versuchsbericht	X	X	000 S.		Jahr							
Vorlesungsskript [7]	Manuskript	report	X	X		Vorlesungsskript	X	X	000 S.		Jahr							
Zeitschriftenartikel [10] [16]	Zeitschriftenart.	article	X	X					00-00		Monat.Jahr			V. 00 oder 00	No. 00 oder 00	X		

Tabelle 6.1: Für die Literaturverweise benötigte Informationen beim Heraufladen auf Zotero und Zitieren in \LaTeX

Anhang A Anhangstruktur

Hier sollte man am besten jegliche Teile über den `\include`-Befehl importieren. Die Überschriften werden genau gleich wie beim Hauptteil des Berichts über die Befehle `\chapter`, `\section`, `\subsection` und `\subsubsection` eingefügt. Die Layoutstruktur ist analog zu den normalen Kapiteln:

A.1 Unterkapitel im Anhang

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

A.1.1 Tieferes Kapitel

Noch tieferes Kapitel

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Literaturverzeichnis

- [1] Amsler, M., *Bemessung von Platten - Modelle und Beispiele*, Master-Thesis, Horw: Hochschule Luzern - Technik & Architektur, Kompetenzzentrum Konstruktiver Ingenieurbau, 2013, 105 S. (Zitiert auf S. 18).
- [2] Amsler, M., *Verstärkung von bestehenden Betontragwerken mit Aufbeton*, VM1, Horw: Hochschule Luzern - Technik & Architektur, Kompetenzzentrum Konstruktiver Ingenieurbau, 2012, 74 S. (Zitiert auf S. 18).
- [3] Amsler, M. und Thoma, K., *Durchstanzversuch mit Aufbeton - Versuch VAI*, Versuchsbericht, Horw: Hochschule Luzern - Technik & Architektur, Kompetenzzentrum Konstruktiver Ingenieurbau, 2013, 70 S. (Zitiert auf S. 18).
- [4] *Eurocode 2: Bemessung und Konstruktion von Stahlbeton- und Spannbetontragwerken - Teil 1-1: Allgemeine Bemessungsregeln und Regeln für den Hochbau*, Lausanne: Europäisches Komitee für Normung, 2010, 246 S. (Zitiert auf S. 18).
- [5] Grob, J., *Ermüdung von Stahlbeton- und Spannbetontragwerken*, Bericht Nr. 75, Zürich: IBK, 1977, 58 S. (Zitiert auf S. 18).
- [6] Haller, P., *Schwinden und Kriechen von Mörtel und Beton*, Diskussionsbericht Nr. 124, Zürich: Eidgenössische Materialprüfungs- und Versuchsanstalt, 1940, S. 39, (Zitiert auf S. 18).
- [7] Menn, C., *Langzeit-Vorgänge - Der Einfluss von Kriechen und Schwinden auf den Verformungs- und Spannungszustand von Stahl-Beton-Tragwerken*, Vorlesungsskript, Zürich: ETH Zürich, 1977, 69 S. (Zitiert auf S. 18).
- [8] *Model Code 2010 - Final draft, Volume 1*, fib Bulletin No. 65, Lausanne: Fédération Internationale du Béton, 2010, 311 S. (Zitiert auf S. 18).
- [9] Muttoni, A., Schwarz, J. und Thürlimann, B., *Bemessen und Konstruieren von Stahlbetontragwerken mit Spannungsfeldern*, Vorlesungsskript, Zürich: ETH Zürich, 1988, 122 S. (Zitiert auf S. 18).
- [10] Rüschi, H., "Researches Toward a General Flexural Theory for Structural Concrete", in: *Journal of the American Concrete Institute* 57 (No. 7 Juli 1960), S. 28, (Zitiert auf S. 17, 18).
- [11] Seelhofer, H., "Ebener Spannungszustand im Betonbau Grundlagen und Anwendungen", Diss., Zürich: ETH Zürich, 2009, 247 S., (Zitiert auf S. 17, 18).
- [12] *SIA Dokumentation D 0192, Betonbau, Bemessungsbeispiele zur Norm SIA 262*, Zürich: Schweizerischer Ingenieur- und Architektenverein, 2004, 156 S. (Zitiert auf S. 18).
- [13] *SIA Norm 262, Betonbau*, Zürich: Schweizerischer Ingenieur- und Architektenverein, 2013, 102 S. (Zitiert auf S. 18).
- [14] Szépe, F., "Bemessung der Eisenbahnbrücken in Stahlbeton mit Rücksicht auf die Einschränkung der Rissbildung", in: *IABSE*, Bd. Vol. 5, 1956, S. 843–857, (Zitiert auf S. 18).
- [15] Teschl, S., *Matlab - Eine Einführung*, Anleitung, Wien, 2001, 44 S. (Zitiert auf S. 18).
- [16] Trost, H., "Auswirkungen des Superpositionsprinzips auf Kriech- und Relaxationsprobleme bei Beton und Spannbeton", in: *Beton und Stahlbetonbau* 10 (1967), S. 230–238, 261–269, (Zitiert auf S. 18).
- [17] Wehnert, M., *Ein Beitrag für drainierten und undrainierten Analyse in der Geotechnik*, Eigenverlag des Instituts für Geotechnik, 2006, 167 S., (Zitiert auf S. 18).

Bezeichnungen

Lateinische Grossbuchstaben

A_c	Fläche eines Betonquerschnitts
B	Belastungsgrad
B_{cr}	Belastungsgrad bei Erreichen des Risslastniveaus
E_c	Elastizitätsmodul von Beton
M	Moment
P	Pol auf dem Mohrschen Kreis der Verzerrungen
P	Einzellast
P_F	Pol auf dem Mohrschen Kreis der aufgetragenen Spannungen
Q	Last, Belastung
RH	Luftfeuchtigkeit

Lateinische Kleinbuchstaben

a_s	längenbezogener Bewehrungsquerschnitt
c_u, c_o	Bewehrungsüberdeckung unten und oben
c_{cIij}	Ungerissene Betonsteifigkeitsmatrix
c_{cIIij}	Gerissene Betonsteifigkeitsmatrix
n_x, n_y, n_{xy}	Plattenschnittkräfte: Längenbezogene Normalkräfte
q_x, q_y, q_z	Flächenlasten
s	Beiwert Abbindegeschwindigkeit
s_{rm}	diagonaler Rissabstand
t_s	Zeitpunkt des Schwindbeginns
u	Umfang des Betonquerschnitts
x, y, z	Kartesische Koordinaten

Griechische Grossbuchstaben

$\Delta\sigma_{ci}$	Tensor Änderung der Betonspannungen
---------------------	-------------------------------------

Griechische Kleinbuchstaben

α	Faktor Abbindegeschwindigkeit, Drehwinkel Transformation
----------	--

Bezeichnungen

$\epsilon_{cs}, \epsilon_{csi}$	Schwinddehnung bzw. Schwinddehnungstensor des Betons
$\epsilon_{cs,\infty}$	Endschwindmass
ρ_x, ρ_y	geometrischer Bewehrungsgehalt in x -Richtung bzw. in y -Richtung
φ	Kriechzahl

Sonderzeichen

$\varnothing_x, \varnothing_y$	Stabdurchmesser der Bewehrung in x -Richtung bzw. in y -Richtung
∂	Differenz bei der partiellen Ableitung
∞	unendlich

Abkürzungen

CMM	Gerissenes Scheibenmodell
Emat	Steifigkeitsmatrix (Jakobimatrix)
GH	Modell für gerissene Hauptrichtungen
LE	Modell für linearelastisches Verhalten
MC	Model Code

Lebenslauf

Personalien

Name	Peter Muster
Adresse	Bahnhofstrasse 1 6004 Luzern
Geburtsdatum	01.01.1989
Heimatort	6004 Luzern
Zivilstand	ledig


Ausbildung

August 1996 - Juli 2005	Primar- und Sekundarschule, Dallenwil
August 2005 - Juli 2009	Lehre als Bauzeichner mit technischer Berufsmaturität Biegebruch GmbH, Luzern
September 2009 - Juli 2012	Bauingenieurstudium Bachelor of Science Hochschule Luzern - Technik & Architektur, Horw
September 2013 - Februar 2016	Bauingenieurstudium Master of Science Vertiefung im Konstruktiven Ingenieurbau Hochschule Luzern - Technik & Architektur, Horw

Berufliche Tätigkeit

Juli 2010 - August 2010	Bauzeichner bei Schubversagen AG, Luzern
Juli 2011 - August 2011	Hilfsassistent Abteilung Bautechnik, Hochschule Luzern - Technik & Architektur, Horw
Dezember 2012 - September 2015	Assistent Abteilung Bautechnik, Hochschule Luzern - Technik & Architektur, Horw

Liste der noch zu erledigenden Punkte

 Link zur Aufgabenstellung im Appendix	2
---	---