

# UAV Serial Switch

## 1 Funktionalität

Der Serial Switch besitzt grundsätzlich zwei Seiten – die Device Seite sowie die Wireless Seite. Auf Device Seite befindet sich im Falle der Bodenstation die Laptops oder im Falle des Fluggerätes die verschiedenen Komponenten, zu denen eine serielle Verbindung besteht. Auf Wireless Seite befinden sich jeweils die Funkmodule. Sowohl auf Device- als auch auf Wireless-Seite sind jeweils vier Verbindungen vorhanden, welche SER\_0 bis SER\_3 benannt sind. In Abbildung 1 sieht man die Übersicht über die involvierte Hardware in einem typischen Use Case.

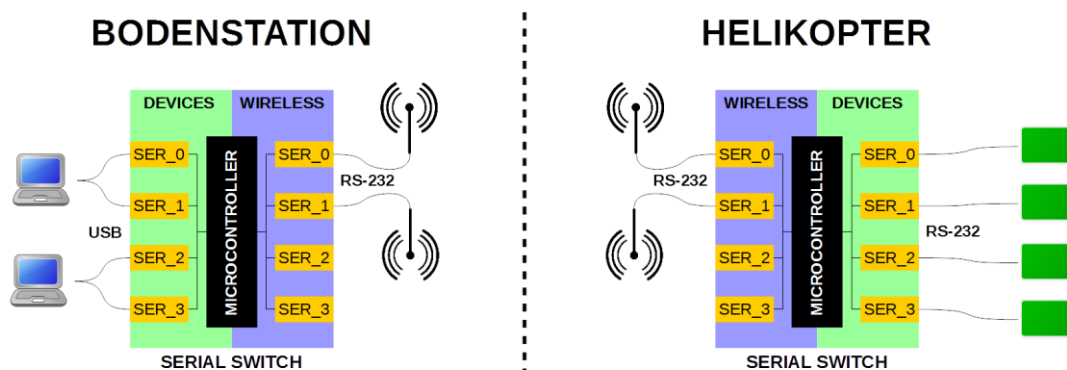


Abbildung 1: Übersicht Hardware, typischer Use Case

Sowohl bei der Bodenstation als auch im Helikopter befindet sich dieselbe Hardware. Auf Device-Seite kann auf dem Serial Switch mit Jumpers gewählt werden, ob die Daten per RS-232 oder USB CDC abgegriffen werden.

Beim Serial Switch muss jeweils unter anderem Konfiguriert werden, welche Serielle Verbindung von der Device Seite mit welcher Priorität über welchen Funkkanal übertragen werden soll. Dies ist in Kapitel 2 genauer erläutert.

## 2 Konfiguration und Priorisierung

| Konfigurationsparameter   | Beschreibung  | Wertebereich/Beispiel  |
|---|---|--|
| <b>BAUD_RATES_WIRELESS_CONN</b><br>sowie<br><b>BAUD_RATES_DEVICE_CONN</b> | Konfiguration der Baudraten der einzelnen seriellen Verbindungen.   | {SER_0, SER_1, SER_2, SER_3}<br>Es wird jeweils direkt die Baudrate eingefüllt. Momentan werden die Baudraten 9600, 38400, 57600 sowie 115200 unterstützt.<br><i>Beispiel:</i> {9600, 38400, 57600, 115200}  |
| <b>PRIO_WIRELESS_CONN_DEV_X</b><br><br>(X = 0-3)                          | Priorität einer Wireless Verbindung aus Sicht einer Device Verbindung. Dieser Konfigurationsparameter existiert für jedes Device und gibt an, mit welcher Priorität welche Wireless Verbindung genutzt werden soll. Er besteht entsprechend aus vier Werten, wobei der Wert Null bedeutet, dass die entsprechende Verbindung nicht genutzt werden soll. Wichtig: Die Prioritäten müssen aufeinanderfolgend sein, wenn also nur die Prioritäten 1 und 3 vorhanden sind, wird die Priorität 3 nie erreicht. | {SER_0, SER_1, SER_2, SER_3}<br>Wertebereich: 0-4<br>0: Verbindung nicht verwenden<br>1: Höchste Priorität<br>2: Zweite Priorität usw.<br>Ist bei zwei Verbindungen dieselbe Priorität vorhanden, so werden die Daten über beide Verbindungen verschickt.<br><i>Beispiel:</i> {1, 2, 0, 0}<br>Hier wird die Wireless Verbindung SER_0 mit höchster Priorität verwendet, SER_1 mit der zweiten Priorität. SER_2 und SER_3 werden nicht verwendet. |
| <b>SEND_CNT_WIRELESS_CONN_DEV_X</b><br><br>(X = 0-3)                      | Definiert die Anzahl Sendeversuche pro Wireless-Verbindung, die durchgeführt werden, bevor auf eine Verbindung mit niedriger Priorität ausgewichen wird oder aber die Daten verworfen werden, wenn keine Verbindung mit niedrigerer Priorität vorhanden ist.  | {SER_0, SER_1, SER_2, SER_3}<br>Wertebereich: 0-255<br>Definiert die Anzahl Versuche pro Wireless Verbindung, die durchgeführt werden, die Daten über diese Verbindung zu senden.<br><i>Beispiel:</i> {5, 10, 0, 0}<br>Bei der Verbindung SER_0 werden 5 Versuche durchgeführt, die Daten zu senden, bei SER_1 10 versuche sowie bei den Verbindungen SER_2 und SER_3 keine Versuche.  |
| <b>RESEND_DELAY_WIRELESS_CONN_DEV_X</b>                                   | Definiert pro Device und Wireless Connection die Zeit in ms, die gewartet wird, bis das Paket erneut gesendet wird falls in dieser Zeit keine Empfangsbestätigung empfangen wird. Hinweis: Ein zu kurzes Resend-Delay sollte vermieden werden, da ansonsten Pakete mehrfach gesendet werden bevor das Acknowledge überhaupt empfangen wird.   | {SER_0, SER_1, SER_2, SER_3}<br>Wertebereich: 0-255<br><br><i>Beispiel:</i><br>RESEND_DELAY_WIRELESS_CONN_DEV_0 {10, 20, 20, 20}<br>Beim Device 0 wird bei der Wireless Verbindung 0 10ms gewartet, bis erneut ein Paket gesendet wird, bei den anderen Verbindungen jeweils 20ms.   |
| <b>MAX_THROUGHPUT_WIRELESS_CONN</b>                                       | Beschränkung des maximalen Durchsatzes pro Wireless Verbindung in bytes/s. Haben mehrere Devices bei der gleichen Wireless Verbindung dieselbe Priorität (PRIO_WIRELESS_CONN_DEV_X) und reicht der Durchsatz aufgrund davon nicht aus, so werden die Daten des Devices mit der niedrigeren Devicepriorität auf eine andere Verbindung umgeleitet oder aber verworfen (abhängig von den Parametern PRIO_WIRELESS_CONN_DEV_X und SEND_CNT_WIRELESS_CONN_DEV_X)  | {SER_0, SER_1, SER_2, SER_3}<br>Wertebereich: 0-(2 <sup>32</sup> -1) bytes/s<br><br><i>Beispiel:</i> {1000, 3000, 0, 0}<br>SER_0: Maximal 1000 bytes/s<br>SER_1: Maximal 3000 bytes/s<br>SER_2: Kein Funkmodul vorhanden<br>SER_3: Kein Funkmodul vorhanden  |
| <b>USUAL_PACKET_SIZE_DEVICE_CONN</b>                                      | Konfiguration der üblichen Paketgrösse auf Device Seite falls bekannt. In der Software wird dann maximal PACKAGE_GEN_MAX_TIMEOUT gewartet, ganze Devicepakete zu sammeln und in Wirelesspakete zu verpacken. Ist dieser Wert 0, so mit einem Intervall von 1ms die vorhandenen Daten in ein Paket verpackt und verschickt.  | {SER_0, SER_1, SER_2, SER_3}<br>Übliche Paketgrösse in bytes pro Deviceverbindung falls bekannt oder 0 falls unbekannt.<br>Wertebereich: 0-512 bytes<br><br><i>Beispiel:</i> {256, 0, 0, 0}<br>SER_0: Übliche Paketgrösse 256 bytes<br>SER_1-3: Übliche Paketgrösse unbekannt  |
| <b>PACKAGE_GEN_MAX_TIMEOUT</b>  | Maximale Zeit, die gewartet wird, zwischen dem Zeitpunkt von dem Daten vom Device   | {SER_0, SER_1, SER_2, SER_3}<br>Maximales Timeout pro Device in ms.  |

|                                       |  |   |
|---------------------------------------|--|---|
|                                       | vorhanden sind bis die Daten versendet werden ohne <i>USUAL_PACKET_SIZE_DEVICE_CONN</i> erreicht wurde.  | <i>Beispiel: {3, 3, 3, 3}</i><br>Maximales Timeout 3ms für alle Devices.  |
| <i>DELAY_DISMISS_OLD_PACK_PER_DEV</i> | Legt fest, wie lange ein Paket im Buffer behalten wird, während bereits ein neues Paket vom selben Device darauf wartet, gesendet zu werden.   | <i>{SER_0, SER_1, SER_2, SER_3}</i><br>Zeit in ms, während der versucht wird, ein altes Paket erneut zu senden, während bereits ein neues Paket vom selben Device vorhanden ist.<br><br><i>Beispiel: {5, 10, 5, 5}</i><br>Beim Device <i>SER_1</i> wird 10ms gewartet, bei den anderen Devices 5ms. |
| <i>SEND_ACK_PER_WIRELESS_CONN</i>     | Parameter um zu konfigurieren, bei welchen Wireless Verbindungen bei Empfang eines Datenpaketes ein Acknowledge zurückgesendet werden soll.  | <i>{SER_0, SER_1, SER_2, SER_3}</i><br>1 für senden von Acknowledge, bei 0 wird kein Acknowledge gesendet.<br><br><i>Beispiel: {1, 0, 1, 1}</i><br>Bei allen Verbindungen mit Ausnahme von <i>SER_1</i> werden Acknowledges zurückgesendet.   |
| <i>USE_CTS_PER_WIRELESS_CONN</i>      | Parameter zum Ein-/Ausschalten der Berücksichtigung des CTS-Signals (Hardware Flow Control) auf Wireless-Seite (ist das Signal 0, so werden Daten gesendet, wenn 1 wird gewartet => Input auf Serial Switch Hardware).<br>Hinweis: Wenn dies eingeschaltet wird muss zwingend auch die entsprechende Hardware-Verbindung zum Wireless Transmitter gemacht werden, da ansonsten gar keine Daten mehr gesendet werden (es ist ein schwacher interner Pull Up Widerstand dieses Signals vorhanden). | <i>{SER_0, SER_1, SER_2, SER_3}</i><br>1 für Berücksichtigung des CTS-Signals, 0 wenn nicht.<br><br><i>Beispiel: {1, 0, 0, 0}</i><br>Bei der Verbindung <i>SER_0</i> wird das CTS-Signal berücksichtigt, bei allen anderen nicht.   |

**Tabelle 1: Auflistung und Beschreibung der verschiedenen Konfigurationsparameter**

In Tabelle 1 sind die verschiedenen Konfigurationsparameter aufgelistet und beschrieben. In der Firmware finden sich diese im File *serialSwitch\_Config.h*, momentan werden diese Parameter statisch zur Compilezeit konfiguriert.

Neben der in der Tabelle beschriebene *PRIO\_WIRELESS\_CONN\_DEV\_X* existiert auch noch die Devicepriorität, welche höher gewichtet wird. Die Devicepriorität ist fix abhängig vom verwendeten seriellen Anschluss des Devices. So wird dem Anschluss *SER\_0* die höchste Priorität zugeordnet, *SER\_1* die zweithöchste usw. Entsteht also beispielsweise ein Engpass beim Datendurchsatz, so werden die Daten des Devices mit der höheren Devicepriorität bevorzugt.

Nicht unterschätzt werden sollte das Timing, in welchem Abstand die Sendeveruche (wenn Parameter *SEND\_CNT\_WIRELESS\_CONN\_DEV\_X > 1*) vorgenommen werden, siehe Parameter *RESEND\_DELAY\_PER\_DEVICE\_MS*. Ist dieser Parameter zu klein gewählt, so werden mehrere Pakete gesendet, bevor überhaupt die Empfangsbestätigung empfangen wird, was zu unnötig hohem Traffic über die Wireless Verbindung führen kann. Wie gross dieser Parameter sein sollte hängt unter anderem von der verwendeten Funkverbindung und der Baudrate ab.

Ein anderer wichtiger Parameter bezüglich des Timings ist *PACKAGE\_GEN\_MAX\_TIMEOUT*. Damit wird definiert, wie lange maximal bei einem Device auf Daten gewartet wird, bis ein Paket daraus generiert und danach verschickt wird. Ist dies zu kurz gewählt, so kann dies die Anzahl Pakete bei der gleichen zu übertragenden Anzahl Bytes von Device zu Device erhöhen. Dadurch wird unnötig viel Traffic auf der Wireless Verbindung generiert. Bei einer langen Zeitdauer wiederum werden die

Daten entsprechend länger zwischengespeichert, wodurch die Verzögerung auf dem Kommunikationspfad zunimmt.

### 3 Beschreibung Wireless Kommunikation

Eine grobe Übersicht über die Hardware wurde bereits in Kapitel 1 gegeben. Die Frage ist nun natürlich, wie die in Kapitel 2 beschriebenen Funktionalitäten in der Software umgesetzt werden. Grundsätzlich geht es darum, dass die Daten unverfälscht vom Device der Bodenstation unverfälscht zum Device des Helikopters übertragen werden zu können und umgekehrt. Dazu erhalten die Daten der Devices für die Wireless Strecke einen Header gemäss Tabelle 2. Der Erhalt der Daten wird bestätigt, indem die Gegenseite eine Empfangsbestätigung zurücksendet.

| Name Feld            | Beschreibung   | Länge [bytes] |
|----------------------|--|---------------|
| <i>PACK_START</i>    | Escape-Character (\e in C, 0x1B ASCII). Bezeichnet den Beginn eines Paketes.   | 1             |
| <i>PACK_TYPE</i>     | Markierung des Pakettyps:<br>- 0x01: Datenpaket<br>- 0x02: Empfangsbestätigung für Paket   | 1             |
| <i>DEV_NUM</i>       | Nummer des Devices, welchem dieses Paket zugeordnet ist (0 bis 3)  | 1             |
| <i>SESSION_NR</i>    | Session Nummer. Werden Pakete mit einem älterem Zeitstempel empfangen, als vorher bereits empfangen wurden, werden diese verworfen – ausser wenn die Session Nummer gewechselt hat. Die Session Nummer wird jeweils beim Boot Up zugewiesen (Zufallszahl).   | 1             |
| <i>SYS_TIME</i>      | Systemzeit in ms. Wird auch zur eindeutigen Identifikation eines Paketes verwendet.  | 4             |
| <i>PAYLOAD_SIZE</i>  | Grösse des Payloades in bytes.   | 2             |
| <i>CRC8_HEADER</i>   | 8 bit Checksumme über den Header zur Verifikation der Gültigkeit des Inhaltes.   | 1             |
| <i>PAYLOAD</i>       | Der Inhalt des Payloads ist abhängig vom Pakettyp:<br>- <i>PACK_TYPE</i> 0x01 (Datenpaket): In diesem Fall beinhaltet der Payload die Daten, welche zwischen dem Device der Bodenstation sowie dem Device des Helikopters ausgetauscht werden sollen.<br>- <i>PACK_TYPE</i> 0x02 (Empfangsbestätigung Paket): Beinhaltet genau eine <i>SYS_TIME</i> des Paketes, dessen Empfang bestätigt werden soll. | n             |
| <i>CRC16_PAYLOAD</i> | 16 bit Checksumme über den Payload.  | 2             |

Bei allen Feldern mit Ausnahme von *PACK\_START* wird, falls der Escape-Character vorkommt (\e), dieser durch "\e" ersetzt (ASCII: 0x22 0x1B 0x22). Dies wird auf Empfangsseite entsprechend wieder rückgängig gemacht. Somit wird verhindert, dass sich in den Daten per Zufall ein gültiger Header zusammensetzt. (Ein allfälliges "\e" in den Daten wird entsprechend durch ein ""\e"" ersetzt usw.)

**Tabelle 2: Beschreibung eines Paketes zur Kommunikation auf Wireless Seite**

Der Fluss der Daten innerhalb eines Serial Switches ist in Abbildung 2 aufgezeichnet. Die Funktion der verschiedenen Softwarekomponenten wird nachfolgend kurz erläutert:

- **HW Buffer Interface:** Liest und schreibt die Daten von und zu den Hardware Buffern unter Berücksichtigung des Parameters *MAX\_THROUGHPUT\_WIRELESS\_CONN* auf Wireless Seite. Die Daten werden von den entsprechenden Queues gelesen respektive geschrieben.
- **Package Generator:** Generiert einerseits Datenpakete gemäss Tabelle 2 aus den Daten, die von den Devices gelesen wurden unter Berücksichtigung der Parameter

*USUAL\_PACKET\_SIZE\_DEVICE\_CONN* sowie *PACKAGE\_GEN\_MAX\_TIMEOUT*. Andererseits werden Pakete zur Empfangsbestätigung generiert.

- **Wireless Ack Send Handler:** Liest die generierten Acknowledge Pakete und schreibt diese zum HW Buffer Interface. Empfangsbestätigungspakete werden einmalig gesendet.
- **Wireless Data Send Handler:** Liest die generierten Datenpakete ein, diese werden unter Berücksichtigung der Parameter *PRIO\_WIRELESS\_CONN\_DEV\_X* an die konfigurierte Wireless Verbindung oder Verbindungen gesendet. Danach wird das Paket noch im Speicher behalten, bis die Empfangsbestätigung eintrifft. Falls diese nicht eintrifft, wird das Paket entsprechend dem Parameter *SEND\_CNT\_WIRELESS\_CONN\_DEV\_X* nochmals gesendet oder aber verworfen.
- **Wireless Package Extractor:** Liest die empfangenen Wirelessdaten ein und überprüft anhand der Checksummen, ob die Daten fehlerfrei sind. Sind sie fehlerfrei, so werden – je nach Pakettyp, siehe Tabelle 2 – die Daten via dem HW Buffer Interface an die Devices weitergeleitet. Bei erfolgreich empfangenen Datenpaketen wird dem Package Generator mitgeteilt, ein Empfangsbestätigungspaket zu generieren. Erfolgreich empfangene Empfangsbestätigungen werden dem Wireless Send Handler weitergeleitet.

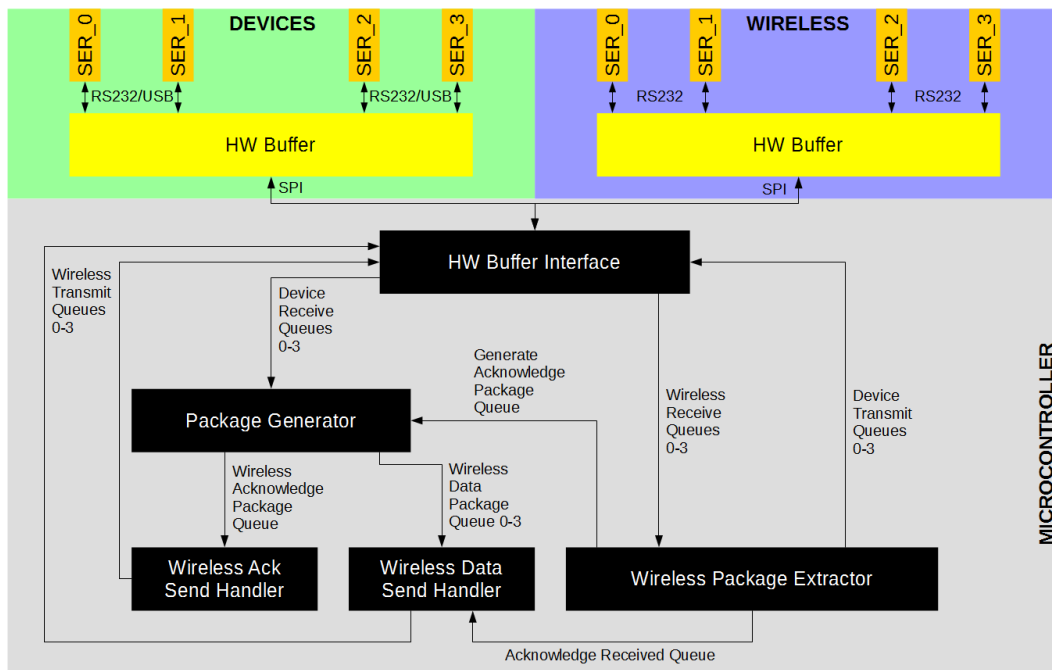


Abbildung 2: Detaillierte Systemübersicht eines einzelnen Serial Switches inklusive Softwarekomponenten