

## PKT Sourcecode, Hinweise zur Softwarestruktur und Bearbeitung

### Wichtiger Hinweis zu Thema Umlaute im Sourcecode und PKT

Der Sourcecode muss in der Entwicklungsumgebung zur Zeit zwingend mit der Codepage ISO-8859-1 bearbeitet und gespeichert werden.

Grund:

normalerweise wird heute UTF-8 als Codesatz verwendet, das Problem bei UTF-8 ist aber das Umlaute als 2 Byte Werte gespeichert und verarbeitet werden, bei ISO-8859-1 aber als 1 byte Wert. Umlaute werden im Sourcecode in der Regel nur für Kommentare verwendet, das ist nicht das Problem. Beim PKT werden Umlaute aber auch als Bestandteil von Strings im LCD-Display benötigt. Die LCD-Routinen im PKT arbeiten nur mit 1 Byte Zeichen, bei UTF-8 werden aber dann bei Umlauten 2Byte Zeichen an das LCD übergeben, damit werden bei Umlauten Hyroglyphen angezeigt. Für UTF-8 müssten die LCD-Routinen in der PKT Software umgebaut werden. Das ist zur Zeit nicht geplant! Wer sich dazu berufen fühlt möge es tun, Beispiele für Stringbehandlung in Verbindung mit UTF-8 findet man im Arduino Umfeld.

### Beschreibung für die Anpassung der PKT-Software an neue FC Versionen

Grundsätzlicher Aufbau der Parameterstruktur des MK:

Die Struktur des MK-Parameterdatensatz ist in der **FC-Software eeprom.h** definiert. Dort bezeichnet als '**paramset\_t**', dieser Parametersatz besteht aus 'char' Feldern die unterschiedlich verwendet werden, meistens als Bytefeld, einige als Bitfeld (8-Bit) und als String. Das erste Char ist die **Revisionsnummer** definiert aus **EEPARAM\_REVISION**. Die Revisionsnummer ist die Referenz die PKT verwendet um unterschiedliche Versionen zu unterscheiden. Eine neue FC-Software hat nicht zwangsläufig eine neue Parameter Revisionsnummer zur Folge.

Für die Parameterbearbeitung im PKT gibt es eine Unterscheidung bei den Bytefeldern und den Bitfeldern. Bei den Bitfeldern muss im PKT noch die Versionnummer der Parametersatzes mitbetrachtet werden..

Jede neue Revisionsnummer hat zur Folge das im PKT in der **paramset.c** eine neue Parameter Const eingefügt werden muss ( `paramRevItem_t const paramset_???[] PROGMEM =` )

Diese Const entspricht exakt dem **paramset\_t** aus der FC-Software.

Die neue Version muss in der **paramset.c** auch in die Mappingtabelle eingetragen werden: ( `paramsetRevMap_t paramsetRevMap[] =` )

In der paramset.c existiert noch eine Konfigurationstabelle ( `paramSubItem_t const paramSubItem[] PROGMEM =` )

in der bestimmte Bit und Bytefelder den Revisionsnummern zugewiesen werden.

Hier werden neue Bit Zuordnungen eingetragen mit dem Verweis von welcher bis zu welcher FC Version die Gültigkeit ist.

Beispiel: 0,0 = für alle Versionen

0,109 = bis Version 109

110,0 = ab Version 110

Es kommt vor das Bits von einer Version zu nächsten undefiniert werden und neue Bedeutung haben.

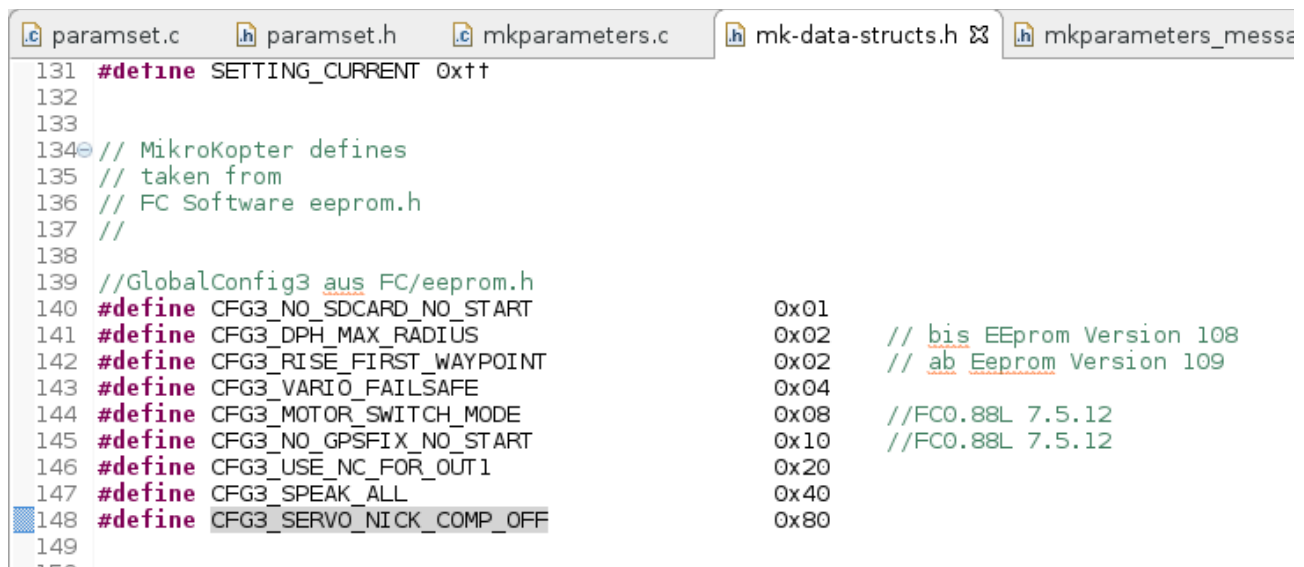
1. Differenz der FC eeprom.h prüfen und neue Parameter finden.
2. neue Parameter zum Editieren in MikroKoptertool lokalisieren.

PKT-Sourcecode anpassen.

Beispiel FC 2.09 zu FC 2.16

EEProm Revision der FC prüfen ob sich die Version erhöht hat, in diesem Beispiel hat sie sich auf 109 erhöht

Neuer Parameter `CFG3_RISE_FIRST_WAYPOINT` in eeprom.h der FC  
Eintragen in **mk-data-structs.h**



```
paramset.c  paramset.h  mkparameters.c  mk-data-structs.h  mkparameters_messa

131 #define SETTING_CURRENT 0x1f
132
133
134 // MikroKopter defines
135 // taken from
136 // FC Software eeprom.h
137 //
138
139 //GlobalConfig3 aus FC/eeprom.h
140 #define CFG3_NO_SD_CARD_NO_START 0x01
141 #define CFG3_DPH_MAX_RADIUS 0x02 // bis EEprom Version 108
142 #define CFG3_RISE_FIRST_WAYPOINT 0x02 // ab EEprom Version 109
143 #define CFG3_VARIO_FAILSAFE 0x04
144 #define CFG3_MOTOR_SWITCH_MODE 0x08 //FC0.88L 7.5.12
145 #define CFG3_NO_GPSFIX_NO_START 0x10 //FC0.88L 7.5.12
146 #define CFG3_USE_NC_FOR_OUT1 0x20
147 #define CFG3_SPEAK_ALL 0x40
148 #define CFG3_SERVO_NICK_COMP_OFF 0x80
149
150
```

in **paramset.c** neue Tabelle für neue Eepromversion 109 mit copy&paste anlegen und anpassen.

```
paramset.c  paramset.h  mkparameters.c  mk-data-structs.h  mkparameters_messages.h
1900 //-----
1901 // FC-Parameter Revision: 109
1902 //
1903 // ab FC-Version: ab 2.16 bis ?
1904 // gefunden in: 2.16
1905 //
1906 // STRUKTUR-DIFF zu 108:
1907 //
1908 //
1909 //
1910 //
1911 //-----
1912 //-----
1913 paramRevItem_t const paramset_109[] PROGMEM =
1914 {
1915     { param_Revision, 1, }, // unsigned char Revision;
1916     { param_Kanalbelegung, 12, }, // unsigned char char Kanalbelegung[12]; // GAS[0], GIER[1], NICK[2], R
1917     { param_GlobalConfig, 1, }, // unsigned char char GlobalConfig; // 0x01=Höhenregler aktiv, 0x0
1918     { param_Hoehe_MinGas, 1, }, // unsigned char char Hoehe_MinGas; // Wert : 0-100
1919     { param_Luftdruck_D, 1, }, // unsigned char char Luftdruck_D; // Wert : 0-250
1920     { param_HoeheChannel, 1, }, // unsigned char char HoeheChannel; // Wert : 0-32
1921     { param_Hoehe_P, 1, }, // unsigned char char Hoehe_P; // Wert : 0-32
1922     { param_Hoehe_Verstaerkung, 1, }, // unsigned char char Hoehe_Verstaerkung; // Wert : 0-50
1923     { param_Hoehe_ACC_Wirkung, 1, }, // unsigned char char Hoehe_ACC_Wirkung; // Wert : 0-250
1924     { param_Hoehe_HoverBand, 1, }, // unsigned char char Hoehe_HoverBand; // Wert : 0-250
1925     { param_Hoehe_TiltCompensation, 1, }, // unsigned char char Hoehe_TiltCompensation; // Wert : 0-250
1926     { param_Hoehe_StickNeutralPoint, 1, }, // unsigned char char Hoehe_StickNeutralPoint; // Wert : 0-250
1927     { param_Stick_D, 1, }, // unsigned char char Stick_D; // Wert : 1-6

```

Die Tabelle enthält die komplette Parameterstruktur die zum MK gesendet bzw. Empfangen wird. Im konkreten Beispiel hat sich nur ein Bit in **GlobalConfig3**; geändert, deshalb ändert sich in der Tabelle im Grunde nichts, ausser der Tabellennamen auf „paramset\_109“.

Anschliessend die Mappingtabelle um den neue Parametersatz ergänzen.

```
paramset.c  paramset.h  mkparameters.c  mk-data-structs.h  mkparameters_messages.h
2342 // Mappingtabelle zwischen FC-Revison und Paramset-Tabelle
2343 //
2344 // Anmerkung bzgl. Groesse:
2345 // Angenommen man wuerde die Rev 97 bis 90 weglassen, dann
2346 // wuerde man ca. 1 KByte sparen
2347 //-----
2348 paramsetRevMap_t paramsetRevMap[] =
2349 {
2350     { 111, paramset_111, }, // 2.19
2351     { 110, paramset_110, }, // 2.18
2352     { 109, paramset_109, }, // 2.16
2353     { 108, paramset_108, }, // 2.14a
2354     { 107, paramset_107, }, // 2.13a
2355     { 106, paramset_106, }, // 2.11a
2356     { 105, paramset_105, }, // 2.09j
2357     { 104, paramset_104, }, // 2.09d
2358     { 103, paramset_103, }, // 2.07f
2359     { 102, paramset_102, }, // 2.05g (bis min. 2.06b)
2360     { 101, paramset_101, }, // 2.05f (eingeschraenkte Unterstuetzung - ) nur in einer einzigen Betaversion vorhanden!)
2361     { 100, paramset_100, }, // 2.05e (Anmerkung OG 06.04.2014: ggf. spaeter loeschen um Platz zu sparen da diese Version n
2362 // Rev. 99: nicht vorhanden; ignorieren
2363     { 98, paramset_098, }, // 2.03d
2364     { 97, paramset_097, }, // 2.02b
2365     { 96, paramset_095, }, // ??? (keine Struktur-Aenderung zu 095)
2366     { 95, paramset_095, }, // 0.90L, 2.00a, 2.00e
2367     { 94, paramset_094, }, // eingeschraenkte Unterstuetzung; 0.91a
2368     { 93, paramset_093, }, // eingeschraenkte Unterstuetzung; 0.90e, 0.90g, 0.90j (keine Struktur-Aenderung zu 093)
2369     { 92, paramset_093, }, // eingeschraenkte Unterstuetzung; 0.90d
2370     { 91, paramset_091, }, // eingeschraenkte Unterstuetzung; 0.88m, 0.88n (keine Struktur-Aenderung zu 091)
2371     { 90, paramset_091, }, // eingeschraenkte Unterstuetzung; 0.88e
2372
2373     { MAPEOF, 0, } // END OF LIST - MUST BE THE LAST!!
2374 };

```

In der **paramset.c** muss der neue Bit- Parameter in der Konfigurationstabelle der paramSubID's (Bit- und Bytefelder) eingetragen werden inkl. Der Information für welchen FC Versionbereich er verwendet wird. Das bezieht sich auf die Eeprom Version der jeweiligen FC-Software

```
paramset.c x h *paramset.h c *mkparameters.c h mk-data-structs.h h *mkparameters_messages.h
```

```
2397
2398 //-----+ von_Rev
2399 //
2400 //
2401 +- paramSubID +- gehoert zu? +- Bit/Byte? +- Bit/Byte der paramID +- bis_
2402 | | | | |
2403 /---|-----|-----|-----|-----|
2404 paramSubItem_t const paramSubItem[] PROGMEM =
2405 {
2406   { param_ServoCompInvert_SERVO_NICK_INV , param_ServoCompInvert, SUBTYPE_BIT, SERVO_NICK_INV , 0, 0},
2407   { param_ServoCompInvert_SERVO_ROLL_INV , param_ServoCompInvert, SUBTYPE_BIT, SERVO_ROLL_INV , 0, 0},
2408   { param_ServoCompInvert_SERVO_RELATIVE , param_ServoCompInvert, SUBTYPE_BIT, SERVO_RELATIVE , 0, 0},
2409
2410   { param_ExtraConfig_HeightLimit , param_ExtraConfig, SUBTYPE_BIT, CFG2_HEIGHT_LIMIT , 0, 0},
2411   { param_ExtraConfig_HeightVario , param_ExtraConfig, SUBTYPE_BITN, CFG2_HEIGHT_LIMIT , 0, 0},
2412   { param_ExtraConfig_VarioBeep , param_ExtraConfig, SUBTYPE_BIT, CFG2_VARIO_BEEP , 0, 0},
2413   { param_ExtraConfig_SensitiveRc , param_ExtraConfig, SUBTYPE_BIT, CFG_SENSITIVE_RC , 0, 0},
2414   { param_ExtraConfig_33vReference , param_ExtraConfig, SUBTYPE_BIT, CFG_3V_REFERENCE , 0, 0},
2415   { param_ExtraConfig_NoRcOffBeeeping , param_ExtraConfig, SUBTYPE_BIT, CFG_NO_RC_OFF_BEEPING , 0, 0},
2416   { param_ExtraConfig_GpsAid , param_ExtraConfig, SUBTYPE_BIT, CFG_GPS_AID , 0, 0},
2417   { param_ExtraConfig_LearnableCarefree , param_ExtraConfig, SUBTYPE_BIT, CFG_LEARNABLE_CAREFREE , 0, 0},
2418   { param_ExtraConfig_IgnoreMagErrAtStartup , param_ExtraConfig, SUBTYPE_BIT, CFG_IGNORE_MAG_ERR_AT_STARTUP , 0, 0},
2419
2420   { param_BitConfig_LoopOben , param_BitConfig, SUBTYPE_BIT, CFG_LOOP_OBEN , 0, 0},
2421   { param_BitConfig_LoopUnten , param_BitConfig, SUBTYPE_BIT, CFG_LOOP_UNTEN , 0, 0},
2422   { param_BitConfig_LoopLinks , param_BitConfig, SUBTYPE_BIT, CFG_LOOP_LINKS , 0, 0},
2423   { param_BitConfig_LoopRechts , param_BitConfig, SUBTYPE_BIT, CFG_LOOP_RECHTS , 0, 0},
2424   { param_BitConfig_MotorBlink1 , param_BitConfig, SUBTYPE_BIT, CFG_MOTOR_BLINK1 , 0, 0},
2425   { param_BitConfig_MotorOffLed1 , param_BitConfig, SUBTYPE_BIT, CFG_MOTOR_OFF_LED1 , 0, 101},
2426   { param_BitConfig_MotorOffLed2 , param_BitConfig, SUBTYPE_BIT, CFG_MOTOR_OFF_LED2 , 0, 101},
2427   { param_BitConfig_MotorBlink2 , param_BitConfig, SUBTYPE_BIT, CFG_MOTOR_BLINK2 , 0, 0},
2428
2429   { param_GlobalConfig3_NoSdCardNoStart , param_GlobalConfig3, SUBTYPE_BIT, CFG3_NO_SD_CARD_NO_START , 0, 0},
2430   { param_GlobalConfig3_DphMaxRadius , param_GlobalConfig3, SUBTYPE_BIT, CFG3_DPH_MAX_RADIUS , 0, 100},
2431   { param_GlobalConfig3_RiseFirstWP , param_GlobalConfig3, SUBTYPE_BIT, CFG3_RISE_FIRST_WAYPOINT , 109, 0},
2432   { param_GlobalConfig3_VarioExitSafe , param_GlobalConfig3, SUBTYPE_BIT, CFG3_VARIO_EXIT_SAFE , 0, 0}
```

In **paramset.h** muss für den neuen Parameter eine neue **ParameterID** vergeben werden. Die ID's dürfen nur einmal verwendet werden. Neue Parameter-ID's sortiert man am besten logisch an die passende Stelle ein und passt die ID's an.

```

paramset.c  paramset.h  mkparameters.c  mk-data-structs.h  mkparameters_messages.h
300
301 #define param_GlobalConfig3_NoSdCardNoStart 220 // SUBTYPE_BIT - mk-data-structs.h: CFG3_NO_SD_CARD_NO_START 0x01
302 #define param_GlobalConfig3_DphMaxRadius 221 // SUBTYPE_BIT - mk-data-structs.h: CFG3_DPH_MAX_RADIUS 0x02
303 #define param_GlobalConfig3_RiseFirstWP 222 // SUBTYPE_BIT - mk-data-structs.h: CFG3_RISE_FIRST_WAYPOINT 0x02
304 #define param_GlobalConfig3_VarioFailSafe 223 // SUBTYPE_BIT - mk-data-structs.h: CFG3_VARIO_FAILSAFE 0x04
305 #define param_GlobalConfig3_MotorSwitchMode 224 // SUBTYPE_BIT - mk-data-structs.h: CFG3_MOTOR_SWITCH_MODE 0x08
306 #define param_GlobalConfig3_NoGpsFixNoStart 225 // SUBTYPE_BIT - mk-data-structs.h: CFG3_NO_GPSFIX_NO_START 0x10
307 #define param_GlobalConfig3_UseNcForOut1 226 // SUBTYPE_BIT - mk-data-structs.h: CFG3_USE_NC_FOR_OUT1 0x20
308 #define param_GlobalConfig3_SpeakAll 227 // SUBTYPE_BIT - mk-data-structs.h: CFG3_SPEAK_ALL 0x40
309 #define param_GlobalConfig3_ServoNickCompOff 228 // SUBTYPE_BIT - mk-data-structs.h: CFG3_SERVO_NICK_COMP_OFF 0x80
310
311 #define param_GlobalConfig_HoehenRegelung 230 // SUBTYPE_BIT - mk-data-structs.h: CFG_HOEHENREGELUNG 0x01
312 #define param_GlobalConfig_HoehenSchalter 231 // SUBTYPE_BIT - mk-data-structs.h: CFG_HOEHEN_SCHALTER 0x02
313 #define param_GlobalConfig_HeadingHold 232 // SUBTYPE_BIT - mk-data-structs.h: CFG_HEADING_HOLD 0x04
314 #define param_GlobalConfig_KompassAktiv 233 // SUBTYPE_BIT - mk-data-structs.h: CFG_KOMPASS_AKTIV 0x08
315 #define param_GlobalConfig_KompassFix 234 // SUBTYPE_BIT - mk-data-structs.h: CFG_KOMPASS_FIX 0x10
316 #define param_GlobalConfig_GpsAktiv 235 // SUBTYPE_BIT - mk-data-structs.h: CFG_GPS_AKTIV 0x20
317 #define param_GlobalConfig_AchsenkopplungAktiv 236 // SUBTYPE_BIT - mk-data-structs.h: CFG_ACHSENKOPPLUNG_AKTIV 0x40
318 #define param_GlobalConfig_Landoutside_Flyzone 237 // SUBTYPE_BIT - mk-data-structs.h: CFG_LAND_OUTSIDE_FLYZONE 0x40
319 #define param_GlobalConfig_DrehratenBegrenzer 238 // SUBTYPE_BIT - mk-data-structs.h: CFG_DREHRATEN_BEGRENZER 0x80
320 #define param_ServoNickFailSave 239 // SUBTYPE_BYTE - mk-data-structs.h: ServoFS_Pos[0]
321 #define param_ServoRollFailSave 240 // SUBTYPE_BYTE - mk-data-structs.h: ServoFS_Pos[1]
322 #define param_Servo3FailSave 241 // SUBTYPE_BYTE - mk-data-structs.h: ServoFS_Pos[2]
323 #define param_Servo4FailSave 242 // SUBTYPE_BYTE - mk-data-structs.h: ServoFS_Pos[3]
324 #define param_Servo5FailSave 243 // SUBTYPE_BYTE - mk-data-structs.h: ServoFS_Pos[4]
325
326

```

Im Beispiel: **#define param\_GlobalConfig3\_RiseFirstWP 222 //**

Anzeigetexte in **mkparameters\_messages.h** eintragen

```

paramset.c  paramset.h  mkparameters.c  mk-data-structs.h  mkparameters_messages.h
606 static const char param_Kanalbelegung_Poti1_de[] PROGMEM = "Poti 1";
607 static const char param_Kanalbelegung_Poti2_de[] PROGMEM = "Poti 2";
608 static const char param_Kanalbelegung_Poti3_de[] PROGMEM = "Poti 3";
609 static const char param_Kanalbelegung_Poti4_de[] PROGMEM = "Poti 4";
610 static const char param_Kanalbelegung_Poti5_de[] PROGMEM = "Poti 5";
611 static const char param_Kanalbelegung_Poti6_de[] PROGMEM = "Poti 6";
612 static const char param_Kanalbelegung_Poti7_de[] PROGMEM = "Poti 7";
613 static const char param_Kanalbelegung_Poti8_de[] PROGMEM = "Poti 8";
614 #define param_Kanalbelegung_Poti1_en param_Kanalbelegung_Poti1_de
615 #define param_Kanalbelegung_Poti2_en param_Kanalbelegung_Poti2_de
616 #define param_Kanalbelegung_Poti3_en param_Kanalbelegung_Poti3_de
617 #define param_Kanalbelegung_Poti4_en param_Kanalbelegung_Poti4_de
618 #define param_Kanalbelegung_Poti5_en param_Kanalbelegung_Poti5_de
619 #define param_Kanalbelegung_Poti6_en param_Kanalbelegung_Poti6_de
620 #define param_Kanalbelegung_Poti7_en param_Kanalbelegung_Poti7_de
621 #define param_Kanalbelegung_Poti8_en param_Kanalbelegung_Poti8_de
622
623 static const char param_CompassOffset_DisableDeclCalc_de[] PROGMEM = "Disable Decl. Calc";
624 #define param_CompassOffset_DisableDeclCalc_en param_CompassOffset_DisableDeclCalc_de
625
626
627 static const char param_NaviMaxFlyingRange_de[] PROGMEM = "Max. Flugradius [10m]";
628 static const char param_NaviMaxFlyingRange_en[] PROGMEM = "Max. Flightradius [10m]";
629
630 static const char param_NaviDescendRange_de[] PROGMEM = "Fail SinkRadius [10m]";
631 #define param_NaviDescendRange_en param_NaviDescendRange_de
632
633 static const char param_GlobalConfig3_RiseFirstWP_de[] PROGMEM = "Höhe vor 1.WP";
634 static const char param_GlobalConfig3_RiseFirstWP_en[] PROGMEM = "Rise before 1. WP";
635

```

In **mkparameter.c** muss jetzt der neue Menüpunkt in der **paramEditItemTable** eingefügt werden.

```
paramset.c  paramset.h  mkparameters.c  mk-data-structs.h  mkparameters_messages.h

391
392
393
394 //*****
395 //-----
396 // ZUR INFO
397 //-----
398 typedef struct
399 {
400     unsigned char paramID;           // paramID aus paramset.h
401     void (*editfunc)(unsigned char paramID, uint8_t cmd); // Edit-Funktion - z.B. editGeneric(); cmd = MKPARAM_EDIT oder MKP
402     const char *format;              // Parameter: String (PROGMEM) (vorallem fuer editGeneric() )
403     unsigned char min;               // Parameter: min (P1)
404     unsigned char max;               // Parameter: max (P2)
405     const char *title_de;            // Text in PROGMEM - Menutext und Beschreibung im Edit-Screen
406     const char *title_en;            // Text in PROGMEM
407 } paramEditItem_t;
408 //*****
409
410 //-----
411 // HINWEIS!
412 // Die unten stehende Aufgliederung in die verschiedenen Menue-Bereiche ist nur zur ORIENTIERUNG!
413 // Jede paramID wird nur EINMAL deklariert - die Zuordnung in die Menues erfolgt in den
414 // nachfolgenden ID_MENU_xyz_Items Strukturen!
415 //-----
416 paramEditItem_t const paramEditItemTable[] PROGMEM =
417 {
418     //-----
419     // Menue: ID_MENU_KANAELE (Kanäle)
420     //-----
421     { param_Kanalbelegung_Gas, &editGeneric, GENERIC_Channel, 0, 0, param_Kanalbelegung_Gas_de,
422     { param_Kanalbelegung_Gear, &editGeneric, GENERIC_Channel, 0, 0, param_Kanalbelegung_Gear_de,
423     { param_Kanalbelegung_Mix, &editGeneric, GENERIC_Channel, 0, 0, param_Kanalbelegung_Mix_de,

543 //-----
544 // Menue: ID_MENU_NAVICTRL (NaviCtrl)
545 //-----
546 { param_GlobalConfig_GpsAktiv, &editGeneric, GENERIC_NoYes, 0, 0, param_GlobalConfig_GpsAktiv_de,
547 { param_NaviGpsModeChannel, &editGeneric, GENERIC_DisChSerWpeFreeCHPH, 0, 0, param_NaviGpsModeChannel_de,
548 { param_NaviGpsGain, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsGain_de,
549 { param_NaviStickThreshold, &editGeneric, GENERIC_Value, 0, 247, param_NaviStickThreshold_de,
550 { param_NaviGpsMinSat, &editGeneric, GENERIC_Value, 0, 247, param_NaviGpsMinSat_de,
551 { param_NaviGpsP, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsP_de,
552 { param_NaviGpsI, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsI_de,
553 { param_NaviGpsD, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsD_de,
554 { param_NaviGpsPLimit, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsPLimit_de,
555 { param_NaviGpsILimit, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsILimit_de,
556 { param_NaviGpsDLimit, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsDLimit_de,
557 { param_NaviGpsA, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviGpsA_de,
558 { param_NaviWindCorrection, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviWindCorrection_de,
559 { param_NaviAccCompensation, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviAccCompensation_de,
560 { param_NaviMaxFlyingRange, &editGeneric, GENERIC_Value, 0, 250, param_NaviMaxFlyingRange_de,
561 { param_NaviOperatingRadius, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviOperatingRadius_de,
562 { param_NaviAngleLimitation, &editGeneric, GENERIC_ValuePoti, 0, 247, param_NaviAngleLimitation_de,
563 { param_NaviPH_LoginTime, &editGeneric, GENERIC_Value, 0, 247, param_NaviPH_LoginTime_de,
564 { param_ExtraConfig_GpsAid, &editGeneric, GENERIC_NoYes, 0, 0, param_ExtraConfig_GpsAid_de,
565 { param_GlobalConfig3_DphMaxRadius, &editGeneric, GENERIC_NoYes, 0, 0, param_GlobalConfig3_DphMaxRadiu
566 { param_GlobalConfig3_RiseFirstWP, &editGeneric, GENERIC_NoYes, 0, 0, param_GlobalConfig3_RiseFirstWP
567 { param_ComingHomeAltitude, &editGeneric, GENERIC_Value, 0, 247, param_ComingHomeAltitude_de,
568 { param_ComingHomeOrientation, &editGeneric, GENERIC_CH_Orientation, 0, 3, param_ComingHomeOrientation_de,
569 { param_SingleWpControlChannel, &editGeneric, GENERIC_InaChSerWpeMMM, 0, 0, param_SingleWpControlChannel_de
570 { param_MenuKeyChannel, &editGeneric, GENERIC_InaChSerWpeMMM, 0, 0, param_MenuKeyChannel_de,
571 { param_SingleWpSpeed, &editGeneric, GENERIC_ValuePoti, 0, 247, param_SingleWpSpeed_de,
572 { param_NaviDescendRange, &editGeneric, GENERIC_Value, 0, 250, param_NaviDescendRange_de,
573
```

**param\_GlobalConfig3\_RiseFirstWP** wird hier entsprechend eingefügt mit den Eigenschaften des Editfeldes, min, max etc.



in **mkparamters.c** die Menüzuordnung anpassen, im Beispiel

**param\_GlobalConfig3\_RiseFirstWP**

im **ID\_MENU\_NAVICTRL\_Items** einfügen. In dieser Reihenfolge werden die Einträge im PKT

beim Parameter

ändern

angezeigt.

```
paramset.c paramset.h mkparameters.c mk-data-structs.h mkparameters_m
677
678
679 //#####
680 // # Menue's und Menuezuordnung
681 //#####
682
683 //-----
684 // Menue ID's: Parameters Hauptmenue
685 //-----
686 #define ID_MENU_FAVORITEN 1 // fuer spaeter reserviert....
687 #define ID_MENU_KANAEL 2
688 #define ID_MENU_KONFIGURATION 3
689 #define ID_MENU_STICK 4
690 #define ID_MENU_LOOPING 5
691 #define ID_MENU_HOEHE 6
692 #define ID_MENU_KAMERA 7
693 #define ID_MENU_CAMCTRL 8
694 #define ID_MENU_NAVICTRL 9
695 #define ID_MENU_AUSGAENGE 10
696 #define ID_MENU_VERSCHIEDENES 11
697 #define ID_MENU_GYRO 12
698 #define ID_MENU_BENUTZER 13
699 #define ID_MENU_ACHSKOPPLUNG 14
700 #define ID_MENU_MIXERSETUP 15 // nicht verwendet / unterstuetzt
701 #define ID_MENU_EASYSETUP 16
702
703 #define ID_MENU_TEST 66 // TEST / DEBUG
704
705
706 //-----
707 // Zuordnungen von paramID's zu den jeweiligen Menue's
708 //
709 // Eintrag "SEPARATOR" - damit wird eine Trennlinie
710 // im Menue dargestellt
```

```
paramset.c paramset.h mkparameters.c mk-data-structs.h
880 //-----
881 // Menue: ID_MENU_NAVICTRL
882 //-----
883 unsigned char const ID_MENU_NAVICTRL_Items[] =
884 {
885     param_GlobalConfig_GpsAktiv,
886     param_NaviGpsModeChannel,
887     param_NaviGpsGain,
888     param_NaviStickThreshold,
889     param_NaviGpsMinSat,
890     param_NaviGpsP,
891     param_NaviGpsPLimit,
892     param_NaviGpsI,
893     param_NaviGpsILimit,
894     param_NaviGpsD,
895     param_NaviGpsDLimit,
896     param_NaviGpsA,
897     SEPARATOR,
898     param_NaviWindCorrection,
899     param_NaviAccCompensation,
900     param_NaviMaxFlyingRange,
901     param_NaviOperatingRadius,
902     param_NaviAngleLimitation,
903     param_NaviPH_LoginTime,
904     param_ExtraConfig_GpsAid,
905     param_GlobalConfig3_DphMaxRadius,
906     SEPARATOR,
907     param_ComingHomeAltitude,
908     param_ComingHomeOrientation,
909     SEPARATOR,
910     param_GlobalConfig3_RiseFirstWP,
911     SEPARATOR,
912     param_SingleWpControlChannel,
913     param_MenuKeyChannel,
914     param_SingleWpSpeed,
915     SEPARATOR,
916     param_NaviDescendRange,
917
918     EOF // *** MUST BE THE LAST! ***
919 };
920
```

Zum Schluss PKT Verisonnummer in der **main.h** hochzählen.

```
paramset.c paramset.h mkparameters.c mk-data-structs.h messages.c mkparameters_messages.h main.h 33
314 /** - add: define USE_PCFASTCONNECT
315 /** - add: define USE_FONT_BIG
316 /** - add: define USE_XPRINTF_LONG
317 /** - chg: Anordnung der Feature- & Debug-defines um einfacher via Copy&Paste
318 /** die Einstellungen notieren bzw. posten zu koennen
319 /**
320 /** 03.04.2013 OG
321 /** - chg: define 'analognames' zu define 'USE_MKDEBUGDATA'
322 /** - add: USE_OSD... defines
323 /** #####
324
325 #ifndef _MAIN_H
326 #define _MAIN_H
327 #include <inttypes.h>
328 #include <stdbool.h>
329 // Softwareversion des PKT
330
331 #define PKTSWVersion "3.89b" // PKT Version
332
333
```

Bei kleinen Änderungen z.B. Fehlerbeseitigung am Ende der Version die Buchstaben hochzählen.  
Bei grösseren Änderungen ( neue FC Version) Versionsnummer am Ende hochzählen und mit **###a** weitermachen.

Grundsätzlich bitte daran denken in der History in jeder Sourcedatei die Änderungen kurz und bündig einzutragen.