

altimeter

Készítette Doxygen 1.8.4

Sat Nov 29 2014 15:18:14

Tartalomjegyzék

1. Adatszerkezet-mutató	1
1.1. Adatszerkezetek	1
2. Fájlmutató	3
2.1. Fájllista	3
3. Adatszerkezetek dokumentációja	5
3.1. log_rec_ext_s struktúráreferencia	5
3.1.1. Részletes leírás	5
3.1.2. Adatmezők dokumentációja	5
3.1.2.1. ff_addr	5
3.1.2.2. record_to_write	5
4. Fájlok dokumentációja	7
4.1. /HDD/GIT/variometer/Altimeter/src/Descriptors.c fájlreferencia	7
4.1.1. Függvények dokumentációja	8
4.1.1.1. CALLBACK_USB_GetDescriptor	8
4.1.2. Változók dokumentációja	8
4.1.2.1. ConfigurationDescriptor	8
4.1.2.2. DeviceDescriptor	8
4.1.2.3. LanguageString	8
4.1.2.4. LanguageStringPtr	8
4.1.2.5. ManufacturerString	8
4.1.2.6. ManufacturerStringPtr	9
4.1.2.7. ProductString	9
4.1.2.8. ProductStringPtr	9
4.2. /HDD/GIT/variometer/Altimeter/src/eprom.c fájlreferencia	9
4.2.1. Típusdefiníciók dokumentációja	10
4.2.1.1. log_rec_ext_t	10
4.2.2. Függvények dokumentációja	10
4.2.2.1. ee_get_first_free_address	10
4.2.2.2. ee_read_log_rec	11

4.2.2.3.	ee_write_log_rec	11
4.3.	/HDD/GIT/variometer/Altimeter/src/HP03.c fájlreferencia	12
4.3.1.	Függvények dokumentációja	12
4.3.1.1.	HP03_getPressure	13
4.3.1.2.	HP03_getTemperature	14
4.3.1.3.	HP03_pressureSeaLevelFromAltitude	15
4.3.1.4.	HP03_pressureToAltitude	16
4.3.1.5.	HP03_readCoeffs	17
4.3.1.6.	HP03_reset	17
4.4.	/HDD/GIT/variometer/Altimeter/src/itoa.c fájlreferencia	17
4.4.1.	Függvények dokumentációja	18
4.4.1.1.	itoa	18
4.5.	/HDD/GIT/variometer/Altimeter/src/Kalman.c fájlreferencia	19
4.5.1.	Függvények dokumentációja	20
4.5.1.1.	init_kalman	20
4.5.1.2.	kalman	21
4.6.	/HDD/GIT/variometer/Altimeter/src/LCD.c fájlreferencia	21
4.6.1.	Függvények dokumentációja	22
4.6.1.1.	LCD_clear	22
4.6.1.2.	LCD_init	23
4.6.1.3.	LCD_writeAlt	24
4.6.1.4.	LCD_writeDate	25
4.6.1.5.	LCD_writePress	25
4.6.1.6.	LCD_writeSpeed	26
4.6.1.7.	LCD_writeTemp	26
4.6.1.8.	LCD_writeTime	27
4.6.1.9.	LCD_writeUSB	27
4.6.1.10.	LCD_writeUSB_delete	28
4.7.	/HDD/GIT/variometer/Altimeter/src/logger.c fájlreferencia	28
4.7.1.	Függvények dokumentációja	29
4.7.1.1.	logger_deleteLog	29
4.7.1.2.	logger_init	29
4.7.1.3.	logger_logThis	30
4.7.1.4.	logger_readFromEE	30
4.7.1.5.	logger_writeToEE	30
4.8.	/HDD/GIT/variometer/Altimeter/src/main.c fájlreferencia	31
4.8.1.	Függvények dokumentációja	32
4.8.1.1.	main	32
4.8.2.	Változók dokumentációja	33
4.8.2.1.	actualDate	33

4.8.2.2.	actualTime	33
4.8.2.3.	calculatedSeaLevelPressure	33
4.9.	/HDD/GIT/variometer/Altimeter/src/mymath.c fájlreferencia	34
4.9.1.	Függvények dokumentációja	34
4.9.1.1.	myPow	34
4.10.	/HDD/GIT/variometer/Altimeter/src/periph.c fájlreferencia	34
4.10.1.	Függvények dokumentációja	35
4.10.1.1.	pwm3c0cb	35
4.10.1.2.	pwm3pcb	35
4.10.2.	Változók dokumentációja	35
4.10.2.1.	i2ccfg	35
4.10.2.2.	pwmcfg	36
4.10.2.3.	spicfg	36
4.11.	/HDD/GIT/variometer/Altimeter/src/RTC_r2051.c fájlreferencia	36
4.11.1.	Függvények dokumentációja	37
4.11.1.1.	RTC_getDate	37
4.11.1.2.	RTC_getTime	37
4.11.1.3.	RTC_init	38
4.11.1.4.	RTC_setDate	38
4.11.1.5.	RTC_setTime	38
4.12.	/HDD/GIT/variometer/Altimeter/src/tasks.c fájlreferencia	38
4.12.1.	Függvények dokumentációja	39
4.12.1.1.	Thread1	39
4.12.1.2.	Thread2	40
4.12.1.3.	Thread3	40
4.12.1.4.	Thread4	41
4.12.1.5.	Thread5	42
4.12.2.	Változók dokumentációja	42
4.12.2.1.	actualDate	42
4.12.2.2.	actualTime	42
4.12.2.3.	calculatedSeaLevelPressure	42
4.13.	/HDD/GIT/variometer/Altimeter/src/VirtualSerial.c fájlreferencia	43
4.13.1.	Függvények dokumentációja	43
4.13.1.1.	EVENT_USB_Device_ConfigurationChanged	43
4.13.1.2.	EVENT_USB_Device_ControlRequest	43
4.13.1.3.	VS_echoCharacter	43
4.13.1.4.	VS_setupHardware	44
4.13.1.5.	VS_USBdataHandling	44
4.13.2.	Változók dokumentációja	44
4.13.2.1.	VirtualSerial_CDC_Interface	44

Tárgymutató	46
--------------------	-----------

1. fejezet

Adatszerkezet-mutató

1.1. Adatszerkezetek

Az összes adatszerkezet listája rövid leírásokkal:

log_rec_ext_s	5
--------------------------------	---

2. fejezet

Fájlmutató

2.1. Fájllista

Az összes fájl listája rövid leírásokkal:

/HDD/GIT/variometer/Altimeter/src/ Descriptors.c	7
/HDD/GIT/variometer/Altimeter/src/ eeeprom.c	9
/HDD/GIT/variometer/Altimeter/src/ HP03.c	12
/HDD/GIT/variometer/Altimeter/src/ itoa.c	17
/HDD/GIT/variometer/Altimeter/src/ Kalman.c	19
/HDD/GIT/variometer/Altimeter/src/ LCD.c	21
/HDD/GIT/variometer/Altimeter/src/ logger.c	28
/HDD/GIT/variometer/Altimeter/src/ main.c	31
/HDD/GIT/variometer/Altimeter/src/ mymath.c	34
/HDD/GIT/variometer/Altimeter/src/ periph.c	34
/HDD/GIT/variometer/Altimeter/src/ RTC_r2051.c	36
/HDD/GIT/variometer/Altimeter/src/ tasks.c	38
/HDD/GIT/variometer/Altimeter/src/ VirtualSerial.c	43

3. fejezet

Adatszerkezetek dokumentációja

3.1. log_rec_ext_s struktúrareferencia

Adatmezők

- uint16_t **ff_addr**
- log_rec_t **record_to_write**

3.1.1. Részletes leírás

Definíció a(z) eeprom.c fájl 13. sorában.

3.1.2. Adatmezők dokumentációja

3.1.2.1. uint16_t ff_addr

Definíció a(z) eeprom.c fájl 14. sorában.

3.1.2.2. log_rec_t record_to_write

Definíció a(z) eeprom.c fájl 15. sorában.

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- /HDD/GIT/variometer/Altimeter/src/**eeprom.c**

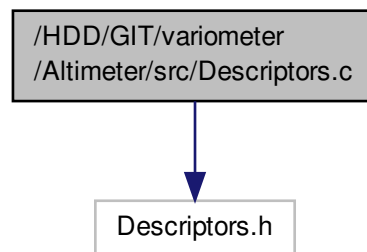
4. fejezet

Fájlok dokumentációja

4.1. /HDD/GIT/variometer/Altimeter/src/Descriptors.c fájlreferencia

```
#include "Descriptors.h"
```

A Descriptors.c definíciós fájl függési gráfja:



Függvények

- `uint16_t CALLBACK_USB_GetDescriptor (uint8_t corenum, const uint16_t wValue, const uint8_t wIndex, const void **const DescriptorAddress)`

Változók

- `USB_Descriptor_Device_t` **DeviceDescriptor**
- `USB_Descriptor_Configuration_t` **ConfigurationDescriptor**
- `uint8_t` **LanguageString []**
- `USB_Descriptor_String_t * LanguageStringPtr = (USB_Descriptor_String_t *) LanguageString`
- `uint8_t` **ManufacturerString []**
- `USB_Descriptor_String_t * ManufacturerStringPtr = (USB_Descriptor_String_t *) ManufacturerString`
- `uint8_t` **ProductString []**
- `USB_Descriptor_String_t * ProductStringPtr = (USB_Descriptor_String_t *) ProductString`

4.1.1. Függvények dokumentációja

4.1.1.1. uint16_t CALLBACK_USB_GetDescriptor (uint8_t corenum, const uint16_t wValue, const uint8_t wIndex, const void **const DescriptorAddress)

Definíció a(z) Descriptors.c fájl 250. sorában.

4.1.2. Változók dokumentációja

4.1.2.1. USB_Descriptor_Configuration_t ConfigurationDescriptor

Definíció a(z) Descriptors.c fájl 85. sorában.

4.1.2.2. USB_Descriptor_Device_t DeviceDescriptor

Kezdő érték:

```
= {
    .Header = { .Size = sizeof(USB_Descriptor_Device_t), .Type = DTYPE_Device },

    .USBSpecification = VERSION_BCD(01.10),
    .Class = CDC_CSCP_CDCClass,
    .SubClass = CDC_CSCP_NoSpecificSubclass,
    .Protocol = CDC_CSCP_NoSpecificProtocol,

    .Endpoint0Size = FIXED_CONTROL_ENDPOINT_SIZE,

    .VendorID = 0x1fc9,
    .ProductID = 0x0083,
    .ReleaseNumber = VERSION_BCD(00.01),

    .ManufacturerStrIndex = 0x01,
    .ProductStrIndex = 0x02,
    .SerialNumStrIndex = USE_INTERNAL_SERIAL,

    .NumberOfConfigurations = FIXED_NUM_CONFIGURATIONS
}
```

Definíció a(z) Descriptors.c fájl 59. sorában.

4.1.2.3. uint8_t LanguageString[]

Kezdő érték:

```
= {
    USB_STRING_LEN(1),
    DTYPE_String,
    WBVAL(LANGUAGE_ID_ENG),
}
```

Definíció a(z) Descriptors.c fájl 188. sorában.

4.1.2.4. USB_Descriptor_String_t* LanguageStringPtr = (USB_Descriptor_String_t *) LanguageString

Definíció a(z) Descriptors.c fájl 193. sorában.

4.1.2.5. uint8_t ManufacturerString[]

Kezdő érték:

```
= {
    USB_STRING_LEN(3),
    DTYPE_String,
    WBVAL('N'),
    WBVAL('X'),
    WBVAL('P'),
}
```

Definíció a(z) Descriptors.c fájl 199. sorában.

4.1.2.6. USB_Descriptor_String_t* ManufacturerStringPtr = (USB_Descriptor_String_t *) ManufacturerString

Definíció a(z) Descriptors.c fájl 206. sorában.

4.1.2.7. uint8_t ProductString[]

Kezdő érték:

```
= {
    USB_STRING_LEN(18),
    DTYPE_String,
    WBVAL('A'),
    WBVAL('1'),
    WBVAL('t'),
    WBVAL('i'),
    WBVAL('m'),
    WBVAL('e'),
    WBVAL('t'),
    WBVAL('e'),
    WBVAL('r'),
    WBVAL(' '),
    WBVAL('C'),
    WBVAL('D'),
    WBVAL('C'),
    WBVAL(' '),
    WBVAL('C'),
    WBVAL('o'),
    WBVAL('n'),
    WBVAL('n'),
}
```

Definíció a(z) Descriptors.c fájl 212. sorában.

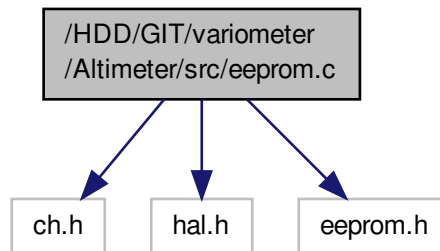
4.1.2.8. USB_Descriptor_String_t* ProductStringPtr = (USB_Descriptor_String_t *) ProductString

Definíció a(z) Descriptors.c fájl 234. sorában.

4.2. /HDD/GIT/variometer/Altimeter/src/eprom.c fájlreferencia

```
#include "ch.h"
#include "hal.h"
#include "eprom.h"
```

Az eeprom.c definíciós fájl függési gráfja:



Adatszerkezetek

- struct **log_rec_ext_s**

Típusdefiníciók

- typedef struct **log_rec_ext_s** **log_rec_ext_t**

Függvények

- int **ee_write_log_rec** (log_rec_t *record, uint16_t *address)
Beír egy rekordot az EEPROM-ba az adott címtől kezdődően.
- uint16_t **ee_read_log_rec** (log_rec_t *record, uint16_t from_addr, uint16_t num_of_rec)
Kiolvas adott számú rekordot az EEPROM-ból az adott címtől kezdődően.
- uint16_t **ee_get_first_free_address** ()
Visszaadja az első szabad hely címét.

4.2.1. Típusdefiníciók dokumentációja

4.2.1.1. typedef struct log_rec_ext_s log_rec_ext_t

4.2.2. Függvények dokumentációja

4.2.2.1. uint16_t ee_get_first_free_address ()

Visszaadja az első szabad hely címét.

Visszatérési érték

Az első szabad hely címe az EEPROM-ban.

Definíció a(z) eeprom.c fájl 140. sorában.

Here is the caller graph for this function:



4.2.2.2. uint16_t ee_read_log_rec (log_rec_t * record, uint16_t from_addr, uint16_t num_of_rec)

Kiolvas adott számú rekordot az EEPROM-ból az adott címtől kezdődően.

Paraméterek

out	<i>record</i>	Rekordokból álló tömb, melyet a függvény tölt fel.
in	<i>from_addr</i>	EEPROM cím, ahonnan a kiolvasás kezdődik.
in	<i>num_of_rec</i>	A kiolvasandó rekordok száma.

Visszatérési érték

A következő kiolvasandó rekord címe.

Definíció a(z) eprom.c fájl 99. sorában.

Here is the caller graph for this function:



4.2.2.3. int ee_write_log_rec (log_rec_t * record, uint16_t * address)

Beír egy rekordot az EEPROM-ba az adott címtől kezdődően.

Paraméterek

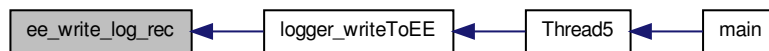
in	<i>record</i>	A beírandó rekord.
in	<i>address</i>	EEPROM cím, ahová a rekord kerül.

Visszatérési érték

0 - sikeres, < 0 - sikertelen.

Definíció a(z) eprom.c fájl 35. sorában.

Here is the caller graph for this function:



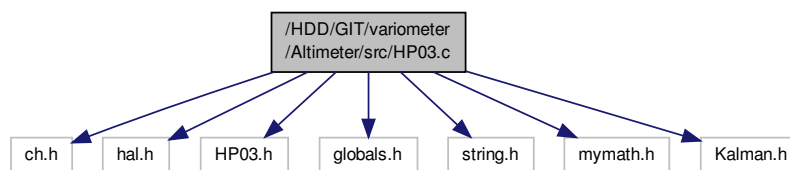
4.3. /HDD/GIT/variometer/Altimeter/src/HP03.c fájlreferencia

```

#include "ch.h"
#include "hal.h"
#include "HP03.h"
#include "globals.h"
#include <string.h>
#include "mymath.h"
#include "Kalman.h"

```

A HP03.c definíciós fájl függési gráfja:



Függvények

- **int HP03_pressureToAltitude** (float seaLevel, HP03_meas_t measuredPressTemp)
Kiszámítja a magasságot (méterben) a hPa-ban megadott légnyomás, tengerszintre átszámított légnyomás és a °C-ban megadott hőmérséklet alapján.
- **float HP03_pressureSeaLevelFromAltitude** (float altitude, HP03_meas_t measuredPressTemp)
Kiszámítja a tengerszintre átszámított légnyomást hPa-ban az aktuális méterben megadott magasság, hPa-ban megadott légnyomás és a °C-ban megadott hőmérséklet alapján.
- **void HP03_reset** ()
Alaphelyzetbe állítja a légnyomás szenzort az XCLR kivezetésének alacsony szintre történő állításával.
- **void HP03_readCoeffs** ()
Kiolvassa a szenzorból a pontos hőmérséklet és légnyomás kiszámításához szükséges koeficiens értékeit.
- **int HP03_getPressure** (HP03_meas_t *constsIn_pressureOut, bool withKalman)
Kiolvassa a légnyomás szenzorból a nyomásértéket, majd a koeficiens segítségével kiszámítja a pontos értéket.
- **int HP03_getTemperature** (HP03_meas_t *result)
Kiolvassa a légnyomás szenzorból a hőmérsékletet, majd a koeficiens segítségével kiszámítja a pontos értéket.

4.3.1. Függvények dokumentációja

4.3.1.1. `int HP03_getPressure (HP03_meas_t * constln_pressureOut, bool withKalman)`

Kiolvassa a légnyomás szenzorból a nyomásértéket, majd a koefficiensek segítségével kiszámítja a pontos értéket.

Paraméterek

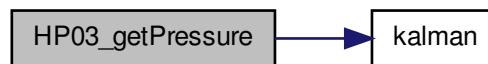
in, out	<i>constsIn_ - pressureOut</i>	Itt keletkezik az eredmény.
in	<i>withKalman</i>	Használja-e a Kálmán szűrőt.

Visszatérési érték

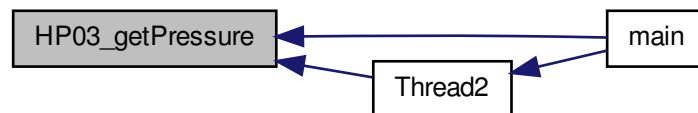
0, ha sikeres.

Definíció a(z) HP03.c fájl 150. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.3.1.2. int HP03_getTemperature (HP03_meas_t * result)

Kiolvassa a légnyomás szenzorból a hőmérsékletet, majd a koefficiensek segítségével kiszámítja a pontos értéket.

Paraméterek

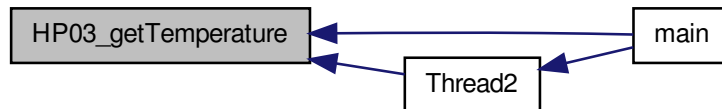
out	<i>result</i>	Itt keletkezik az eredmény.
-----	---------------	-----------------------------

Visszatérési érték

0, ha sikeres.

Definíció a(z) HP03.c fájl 215. sorában.

Here is the caller graph for this function:



4.3.1.3. float HP03_pressureSeaLevelFromAltitude (float *altitude*, HP03_meas_t *measuredPressTemp*)

Kiszámítja a tengerszintre átszámított légnyomást hPa-ban az aktuális méterben megadott magasság, hPa-ban megadott légnyomás és a °C-ban megadott hőmérséklet alapján.

Paraméterek

in	<i>altitude</i>	Magasság méterben.
in	<i>measuredPress-Temp</i>	Mért légnyomás és hőmérséklet.

Visszatérési érték

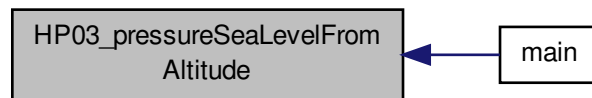
Tengerszintre átszámított légnyomás hPa-ban.

Definíció a(z) HP03.c fájl 71. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.3.1.4. int HP03_pressureToAltitude (float *seaLevel*, HP03_meas_t *measuredPressTemp*)

Kiszámítja a magasságot (méterben) a hPa-ban megadott légnyomás, tengerszintre átszámított légnyomás és a °C-ban megadott hőmérséklet alapján.

Paraméterek

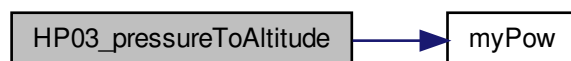
in	<i>seaLevel</i>	Tengerszintre átszámított légnyomás hPa-ban.
in	<i>measuredPressTemp</i>	Mért légnyomás és hőmérséklet.

Visszatérési érték

Magasság méterben.

Definíció a(z) HP03.c fájl 40. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.3.1.5. void HP03_readCoeffs ()

Kiolvassa a szenzorból a pontos hőmérséklet és légnyomás kiszámításához szükséges koefficiensek értékeit.

Visszatérési érték

HP03_coeff változó.

Definíció a(z) HP03.c fájl 116. sorában.

Here is the caller graph for this function:



4.3.1.6. void HP03_reset ()

Alaphelyzetbe állítja a légnyomás szenzort az XCLR kivezetésének alacsony szintre történő állításával.

Definíció a(z) HP03.c fájl 101. sorában.

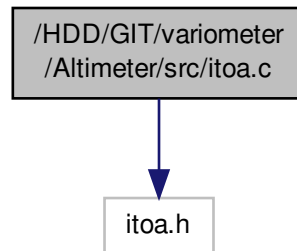
Here is the caller graph for this function:



4.4. /HDD/GIT/variometer/Altimeter/src/itoa.c fájlreferencia

```
#include "itoa.h"
```

Az itoa.c definíciós fájl függési gráfja:



Függvények

- `char * itoa (int value, char *buffer, int base, int decimals, int expectedLength, char padding_char, char *prefix, char *suffix)`

Előjeles egészet karakter tömbbé konvertál.

4.4.1. Függvények dokumentációja

4.4.1.1. `char* itoa (int value, char * buffer, int base, int decimals, int expectedLength, char padding_char, char * prefix, char * suffix)`

Előjeles egészet karakter tömbbé konvertál.

Paraméterek

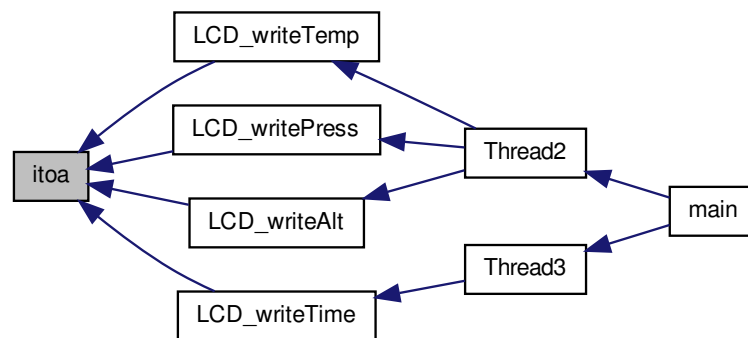
in	<i>value</i>	Az átalakítandó egész szám.
out	<i>buffer</i>	Itt keletkezik az eredmény.
in	<i>base</i>	Alap (2; 10; 16 lehet).
in	<i>decimals</i>	Tizedesek száma.
in	<i>expectedLength</i>	Kívánt karakterhosszúság.
in	<i>padding_char</i>	Kitöltő karakter.
in	<i>prefix</i>	Előtag, mely a karakterlánc elejéhez fűződik.
in	<i>suffix</i>	Utótag, mely a karakterlánc végéhez fűződik.

Visszatérési érték

Mutató az eredmény bufferre.

Definíció a(z) itoa.c fájl 28. sorában.

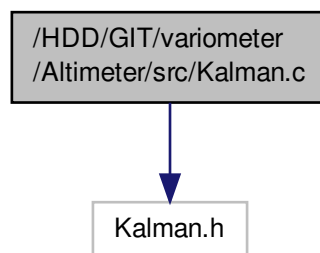
Here is the caller graph for this function:



4.5. /HDD/GIT/variometer/Altimeter/src/Kalman.c fájlreferencia

```
#include "Kalman.h"
```

A Kalman.c definíciós fájl függési gráfja:



Függvények

- void **init_kalman** (int z_measured)
A Kálmán szűrő inicializálása.
- int **kalman** (int z_measured)
Elvégzi a szűrést (jóslás, erősítés, korrekció).

4.5.1. Függvények dokumentációja

4.5.1.1. void init_kalman (int *z_measured*)

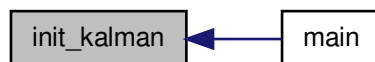
A Kálmán szűrő inicializálása.

Paraméterek

in	<i>z_measured</i>	A mért érték.
----	-------------------	---------------

Definíció a(z) Kalman.c fájl 23. sorában.

Here is the caller graph for this function:

4.5.1.2. int kalman (int *z_measured*)

Elvégzi a szűrést (jóslás, erősítés, korrekció).

Paraméterek

in	<i>z_measured</i>	A mért érték.
----	-------------------	---------------

Visszatérési érték

Számított (jósolt) érték.

< Jóslás

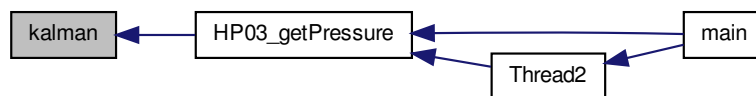
< Kálmán erősítés számítása

< Korrekció

Utolsó érték frissítése

Definíció a(z) Kalman.c fájl 33. sorában.

Here is the caller graph for this function:

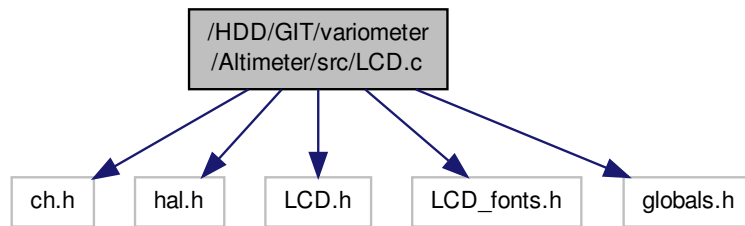


4.6. /HDD/GIT/variometer/Altimeter/src/LCD.c fájlreferencia

```

#include "ch.h"
#include "hal.h"
#include "LCD.h"
#include "LCD_fonts.h"
#include "globals.h"
  
```

A LCD.c definíciós fájl függési gráfja:



Függvények

- void **LCD_clear** ()
Törli a kijelző tartalmát.
- void **LCD_init** ()
A kijelző inicializálását végzi.
- void **LCD_writeTemp** (HP03_meas_t measuredByHP03)
Hőmérséklet megjelenítése a kijelzőn.
- void **LCD_writePress** (HP03_meas_t measuredByHP03)
Légnyomás megjelenítése a kijelzőn.
- void **LCD_writeAlt** (int altToWrite)
Magasság megjelenítése a kijelzőn.
- void **LCD_writeSpeed** (int speedToWrite)
Függőleges sebesség megjelenítése a kijelzőn.
- void **LCD_writeDate** (RTC_date_t dateToWrite)
Dátum megjelenítése a kijelzőn.
- void **LCD_writeTime** (RTC_time_t timeToWrite)
Idő megjelenítése a kijelzőn.
- void **LCD_writeUSB** ()
USB felirat megjelenítése a kijelzőn.
- void **LCD_writeUSB_delete** ()
USB felirat törlése a kijelzőről.

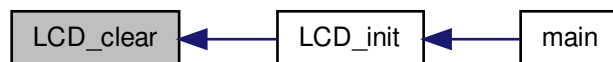
4.6.1. Függvények dokumentációja

4.6.1.1. void LCD_clear ()

Törli a kijelző tartalmát.

Definíció a(z) LCD.c fájl 147. sorában.

Here is the caller graph for this function:



4.6.1.2. void LCD_init ()

A kijelző inicializálását végzi.

< PowerON, ExtCommandSet - 0x21

< Internal HV-gen x3 - 0x09

< Set Vop

< Bias n=2

< Temperature coeff 2

< StandartCommandSet - 0x20

< normal mode, display non-inverted

Definíció a(z) LCD.c fájl 168. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.3. void LCD_writeAlt (int *altToWrite*)

Magasság megjelenítése a kijelzőn.

Paraméterek

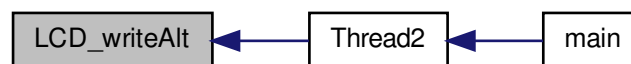
<i>in</i>	<i>altToWrite</i>	Magasság érték.
-----------	-------------------	-----------------

Definíció a(z) LCD.c fájl 245. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.4. void LCD_writeDate (RTC_date_t dateToWrite)

Dátum megjelenítése a kijelzőn.

Paraméterek

<i>in</i>	<i>dateToWrite</i>	Dátum érték.
-----------	--------------------	--------------

Definíció a(z) LCD.c fájl 271. sorában.

4.6.1.5. void LCD_writePress (HP03_meas_t measuredByHP03)

Légnyomás megjelenítése a kijelzőn.

Paraméterek

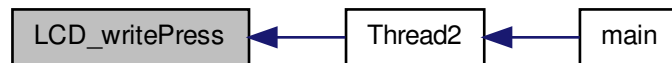
<i>in</i>	<i>measuredByHP03</i>	Nyomás érték.
-----------	-----------------------	---------------

Definíció a(z) LCD.c fájl 220. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.6. void LCD_writeSpeed (int *speedToWrite*)

Függőleges sebesség megjelenítése a kijelzőn.

Paraméterek

in	<i>speedToWrite</i>	Sebesség érték.
----	---------------------	-----------------

Definíció a(z) LCD.c fájl 262. sorában.

4.6.1.7. void LCD_writeTemp (HP03_meas_t *measuredByHP03*)

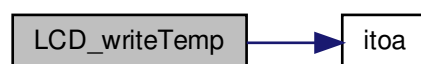
Hőmérséklet megjelenítése a kijelzőn.

Paraméterek

in	<i>measuredByHP03</i>	Hőmérséklet érték.
----	-----------------------	--------------------

Definíció a(z) LCD.c fájl 206. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.8. void LCD_writeTime (RTC_time_t timeToWrite)

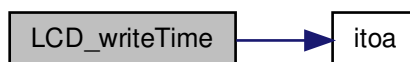
Idő megjelenítése a kijelzőn.

Paraméterek

in	<i>timeToWrite</i>	Idő érték.
----	--------------------	------------

Definíció a(z) LCD.c fájl 281. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

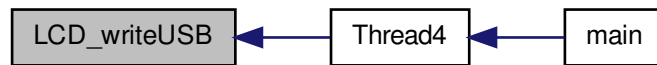


4.6.1.9. void LCD_writeUSB ()

USB felirat megjelenítése a kijelzőn.

Definíció a(z) LCD.c fájl 306. sorában.

Here is the caller graph for this function:



4.6.1.10. void LCD_writeUSB_delete ()

USB felirat törlése a kijelzőről.

Definíció a(z) LCD.c fájl 319. sorában.

Here is the caller graph for this function:

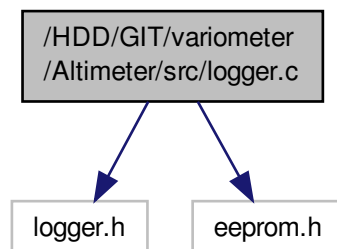


4.7. /HDD/GIT/variometer/Altimeter/src/logger.c fájlreferencia

```
#include "logger.h"
```

```
#include "eeprom.h"
```

A logger.c definíciós fájl függési gráfja:



Függvények

- void **logger_init** ()
A naplózó alrendszer inicializálása.
- int **logger_logThis** (log_rec_t *rec_to_log)
A paraméterként kapott rekordot berakja a ringbufferbe, ahonnan később ki fog íródni az EEPROM-ba.
- int **logger_writeToEE** ()
A ringbufferből kiír egy rekordot az EEPROM-ba.
- uint16_t **logger_readFromEE** (log_rec_t *buffer, uint16_t size_in_rec)
Adott számú rekord kikérése az EEPROM-ból.
- int **logger_deleteLog** ()
Kitörli a naplóállományt az EEPROM-ból.

4.7.1. Függvények dokumentációja

4.7.1.1. int logger_deleteLog ()

Kitörli a naplóállományt az EEPROM-ból.

(Még nincs implementálva.)

Visszatérési érték

0.

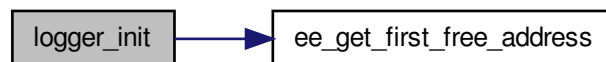
Definíció a(z) logger.c fájl 107. sorában.

4.7.1.2. void logger_init ()

A naplózó alrendszer inicializálása.

Definíció a(z) logger.c fájl 18. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.7.1.3. int logger_logThis (log_rec_t * rec_to_log)

A paraméterként kapott rekordot berakja a ringbufferbe, ahonnan később ki fog íródni az EEPROM-ba.

Paraméterek

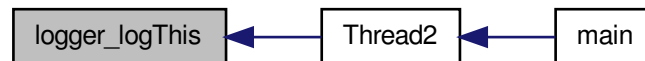
in	<i>rec_to_log</i>	A naplózandó rekord.
----	-------------------	----------------------

Visszatérési érték

0 - normál működés, 1 - túlsordulás, a legrégebbi elem felülíródott.

Definíció a(z) logger.c fájl 37. sorában.

Here is the caller graph for this function:



4.7.1.4. uint16_t logger_readFromEE (log_rec_t * buffer, uint16_t size_in_rec)

Adott számú rekord kikérése az EEPROM-ból.

Paraméterek

out	<i>buffer</i>	Mutató az eredménybufferre.
in	<i>size_in_rec</i>	Rekordok száma.

Visszatérési érték

A következő kiolvasható rekord címe.

Definíció a(z) logger.c fájl 96. sorában.

A függvény hívási gráfja:



4.7.1.5. int logger_writeToEE ()

A ringbufferből kiír egy rekordot az EEPROM-ba.

Ha az írás sikeres volt, törli az elemet a bufferből.

Visszatérési érték

A ringbufferben maradt elemek száma.

Definíció a(z) logger.c fájl 69. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

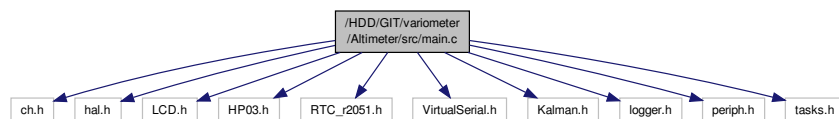
**4.8. /HDD/GIT/variometer/Altimeter/src/main.c fájlreferencia**

```

#include "ch.h"
#include "hal.h"
#include "LCD.h"
#include "HP03.h"
#include "RTC_r2051.h"
#include "VirtualSerial.h"
#include "Kalman.h"
#include "logger.h"
#include "periph.h"
#include "tasks.h"

```

A main.c definíciós fájl függési gráfja:

**Függvények**

- int **main** (void)

Az alkalmazás belépési pontja.

Változók

- volatile RTC_time_t **actualTime**
- volatile RTC_date_t **actualDate**
- volatile float **calculatedSeaLevelPressure** = 0

4.8.1. Függvények dokumentációja

4.8.1.1. int main (void)

Az alkalmazás belépési pontja.

Rendszer-inicializálás.

- HAL inicializálás (SoC és kártya specifikus)
- Kernel inicializálás, a **main()** (o. 32) függvényből szál lesz és az RTOS elindul.

Az SPI és az I2C aktiválása.

A perifériák és modulok inicializálása.

A tengerszintre átszámított légnyomás kiszámíttatása. Az első kiolvasás a szenzorból nem biztos, hogy helyes, ezért két kiolvasás szükséges.

Aktuális dátum kiolvasása az RTC-ből.

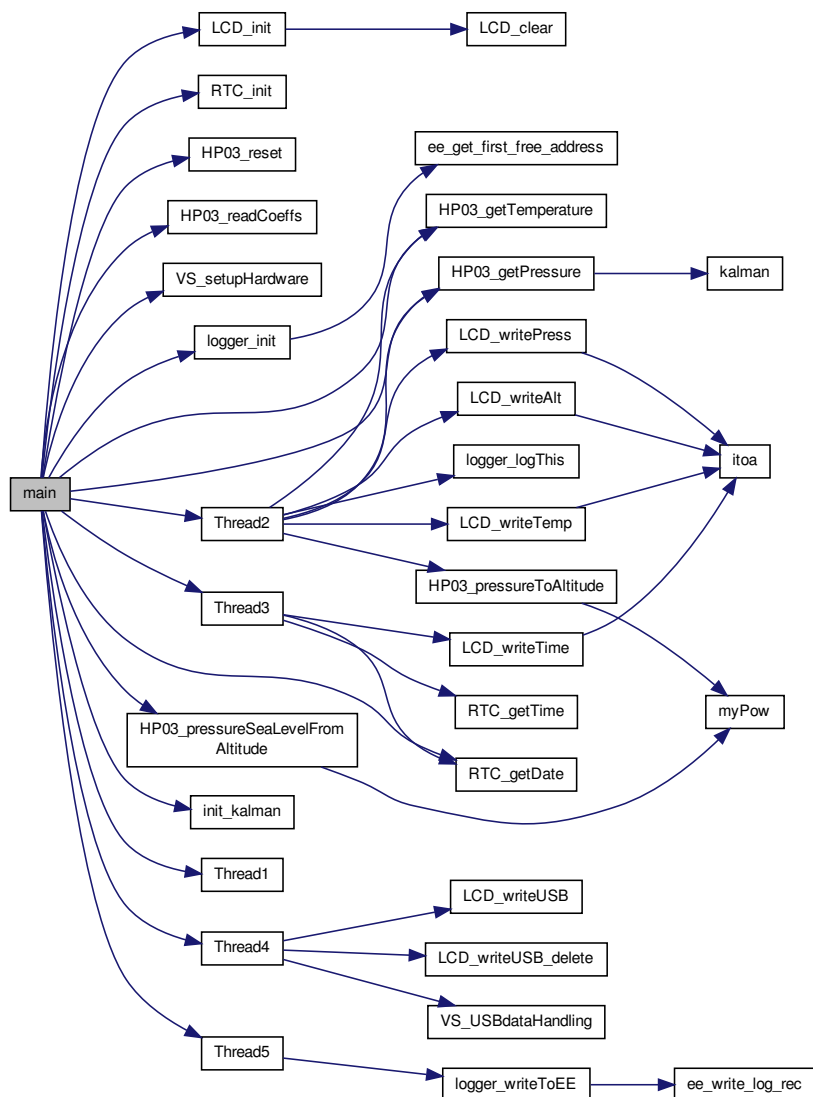
Hangszóró tesztelése (PWM).

Háttérvilágítás bekapcsolása.

Szálak létrehozása.

Definíció a(z) main.c fájl 23. sorában.

A függvény hívási gráfja:



4.8.2. Változók dokumentációja

4.8.2.1. volatile RTC_date_t actualDate

Definíció a(z) main.c fájl 17. sorában.

4.8.2.2. volatile RTC_time_t actualTime

Definíció a(z) main.c fájl 16. sorában.

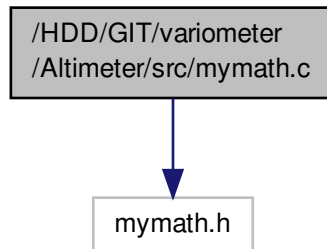
4.8.2.3. volatile float calculatedSeaLevelPressure = 0

Definíció a(z) main.c fájl 18. sorában.

4.9. /HDD/GIT/variometer/Altimeter/src/mymath.c fájlreferencia

```
#include "mymath.h"
```

A mymath.c definíciós fájl függési gráfja:



Függvények

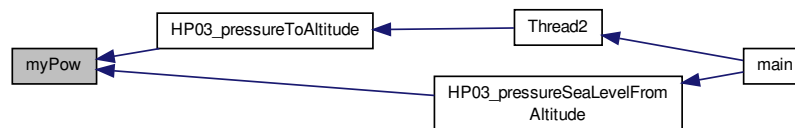
- float **myPow** (float a, float b)

4.9.1. Függvények dokumentációja

4.9.1.1. float myPow (float a, float b)

Definíció a(z) mymath.c fájl 70. sorában.

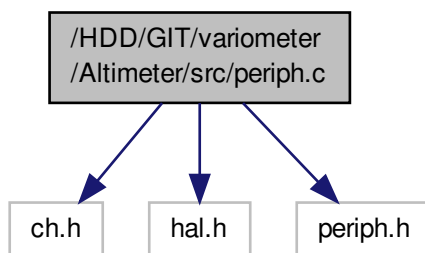
Here is the caller graph for this function:



4.10. /HDD/GIT/variometer/Altimeter/src/periph.c fájlreferencia

```
#include "ch.h"
#include "hal.h"
#include "periph.h"
```


A periph.c definíciós fájl függési gráfja:



Függvények

- void **pwm3pcb** (PWMDriver *pwmp)
- void **pwm3c0cb** (PWMDriver *pwmp)

Változók

- SPIConfig **spicfg**
- I2CConfig **i2ccfg**
- PWMConfig **pwmcfg**

4.10.1. Függvények dokumentációja

4.10.1.1. void pwm3c0cb (PWMDriver * *pwmp*)

Definíció a(z) periph.c fájl 50. sorában.

4.10.1.2. void pwm3pcb (PWMDriver * *pwmp*)

Definíció a(z) periph.c fájl 44. sorában.

4.10.2. Változók dokumentációja

4.10.2.1. I2CConfig i2ccfg

Kezdő érték:

```

= {
    I2C_STANDARD_MODE,
    48
}
  
```

Definíció a(z) periph.c fájl 26. sorában.

4.10.2.2. PWMConfig pwmcfg

Kezdő érték:

```
= {
    100000,
    100,
    pwm3pcb,
    {
        {PWM_OUTPUT_ACTIVE_LOW, pwm3c0cb},
        {PWM_OUTPUT_ACTIVE_LOW, NULL}
    }
}
```

Definíció a(z) periph.c fájl 31. sorában.

4.10.2.3. SPIConfig spicfg

Kezdő érték:

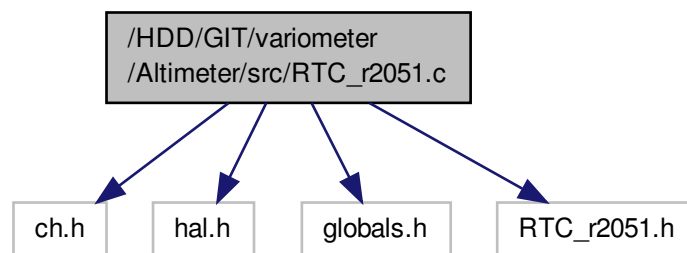
```
= {
    NULL,
    GPIO0,
    GPIO0_LCD_SEL,
    CR0_DSS8BIT | CR0_FRFSPI | CR0_CLOCKRATE(0),
    48
}
```

Definíció a(z) periph.c fájl 15. sorában.

4.11. /HDD/GIT/variometer/Altimeter/src/RTC_r2051.c fájlreferencia

```
#include "ch.h"
#include "hal.h"
#include "globals.h"
#include "RTC_r2051.h"
```

A RTC_r2051.c definíciós fájl függési gráfja:



Függvények

- void **RTC_init** ()
A Real Time Clock áramkör inicializálása.
- void **RTC_setTime** (RTC_time_t timeToBeSet)

Beállítja az időt az RTC áramkörben.

- void **RTC_setDate** (RTC_date_t dateToBeSet)

Beállítja a dátumot az RTC áramkörben.

- RTC_time_t **RTC_getTime** ()

Az idő kiolvasása az RTC áramkörből.

- RTC_date_t **RTC_getDate** ()

A dátum kiolvasása az RTC áramkörből.

4.11.1. Függvények dokumentációja

4.11.1.1. RTC_date_t RTC_getDate ()

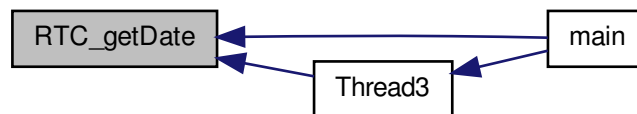
A dátum kiolvasása az RTC áramkörből.

Visszatérési érték

Az aktuális dátum.

Definíció a(z) RTC_r2051.c fájl 123. sorában.

Here is the caller graph for this function:



4.11.1.2. RTC_time_t RTC_getTime ()

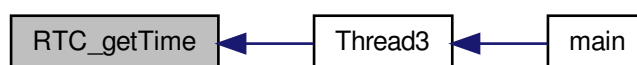
Az idő kiolvasása az RTC áramkörből.

Visszatérési érték

Az aktuális idő.

Definíció a(z) RTC_r2051.c fájl 85. sorában.

Here is the caller graph for this function:

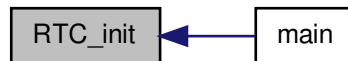


4.11.1.3. void RTC_init ()

A Real Time Clock áramkör inicializálása.

Definíció a(z) RTC_r2051.c fájl 17. sorában.

Here is the caller graph for this function:



4.11.1.4. void RTC_setDate (RTC_date_t dateToBeSet)

Beállítja a dátumot az RTC áramkörben.

Paraméterek

in	<i>dateToBeSet</i>	A beállítandó dátum.
----	--------------------	----------------------

Definíció a(z) RTC_r2051.c fájl 68. sorában.

4.11.1.5. void RTC_setTime (RTC_time_t timeToBeSet)

Beállítja az időt az RTC áramkörben.

Paraméterek

in	<i>timeToBeSet</i>	A beállítandó idő.
----	--------------------	--------------------

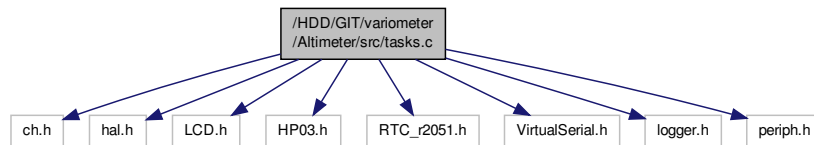
Definíció a(z) RTC_r2051.c fájl 42. sorában.

4.12. /HDD/GIT/variometer/Altimeter/src/tasks.c fájlreferencia

```

#include "ch.h"
#include "hal.h"
#include "LCD.h"
#include "HP03.h"
#include "RTC_r2051.h"
#include "VirtualSerial.h"
#include "logger.h"
#include "periph.h"
  
```

A tasks.c definíciós fájl függési gráfja:



Függvények

- `msg_t Thread1 (void *arg)`
"HeartBeat" LED villogtató szál.
- `msg_t Thread2 (void *arg)`
Nyomás és hőmérséklet kiolvasó szál.
- `msg_t Thread3 (void *arg)`
Dátum és idő kezelő szál.
- `msg_t Thread4 (void *arg)`
USB kezelő szál.
- `msg_t Thread5 (void *arg)`
EEPROM író szál (ringbuffer -> EEPROM)

Változók

- volatile RTC_time_t **actualTime**
- volatile RTC_date_t **actualDate**
- volatile float **calculatedSeaLevelPressure**

4.12.1. Függvények dokumentációja

4.12.1.1. `msg_t Thread1 (void * arg)`

"HeartBeat" LED villogtató szál.

(Az idő msec)

Definíció a(z) tasks.c fájl 26. sorában.

Here is the caller graph for this function:

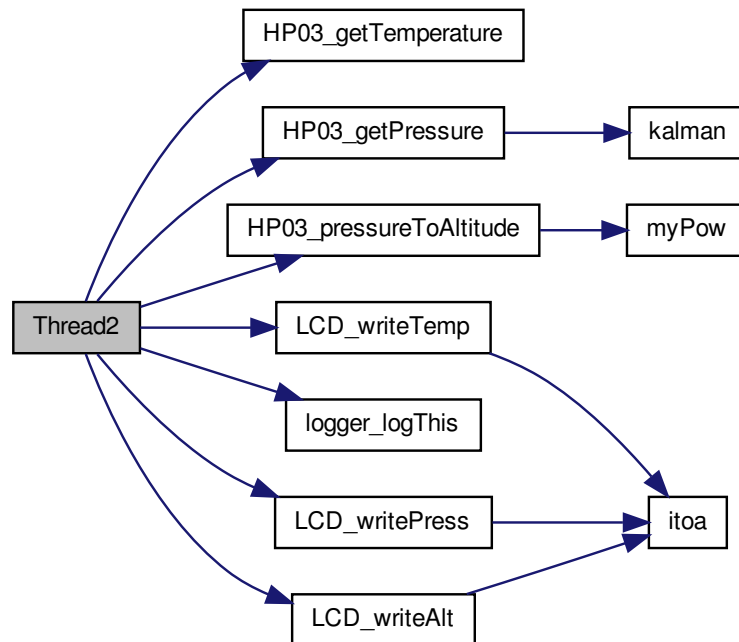


4.12.1.2. `msg_t Thread2 (void * arg)`

Nyomás és hőmérséklet kiolvasó szál.

Definíció a(z) `tasks.c` fájl 43. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

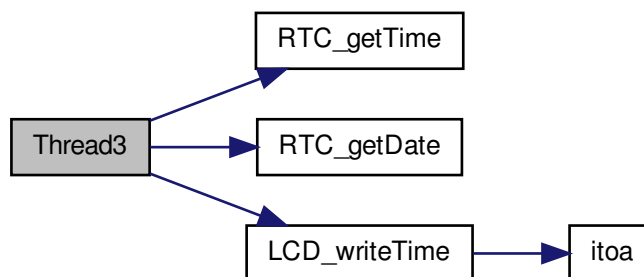


4.12.1.3. `msg_t Thread3 (void * arg)`

Dátum és idő kezelő szál.

Definíció a(z) `tasks.c` fájl 99. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

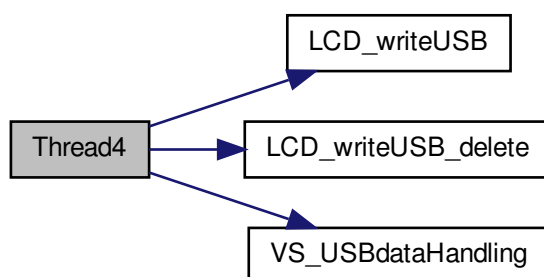


4.12.1.4. msg_t Thread4 (void * arg)

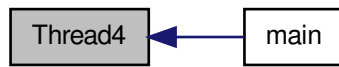
USB kezelő szál.

Definíció a(z) tasks.c fájl 122. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

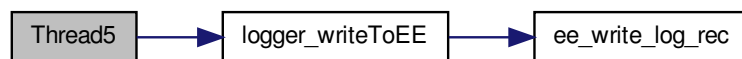


4.12.1.5. msg_t Thread5 (void * arg)

EEPROM író szál (ringbuffer -> EEPROM)

Definíció a(z) tasks.c fájl 149. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.12.2. Változók dokumentációja

4.12.2.1. volatile RTC_date_t actualDate

Definíció a(z) main.c fájl 17. sorában.

4.12.2.2. volatile RTC_time_t actualTime

Definíció a(z) main.c fájl 16. sorában.

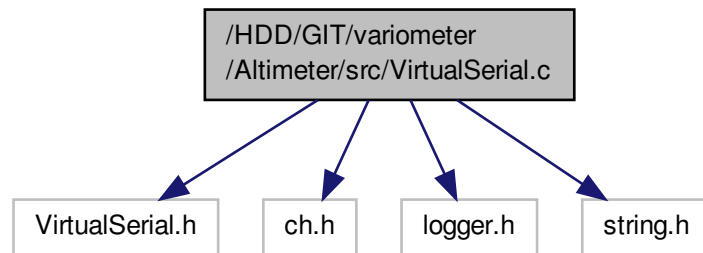
4.12.2.3. volatile float calculatedSeaLevelPressure

Definíció a(z) main.c fájl 18. sorában.

4.13. /HDD/GIT/variometer/Altimeter/src/VirtualSerial.c fájlreferencia

```
#include "VirtualSerial.h"  
#include "ch.h"  
#include "logger.h"  
#include <string.h>
```

A VirtualSerial.c definíciós fájl függési gráfja:



Függvények

- void **VS_setupHardware** (void)
- void **VS_echoCharacter** (void)
- void **EVENT_USB_Device_ConfigurationChanged** (void)
- void **EVENT_USB_Device_ControlRequest** (void)
- void **VS_USBdataHandling** (void)

Változók

- USB_ClassInfo_CDC_Device_t **VirtualSerial_CDC_Interface**

4.13.1. Függvények dokumentációja

4.13.1.1. void EVENT_USB_Device_ConfigurationChanged (void)

Definíció a(z) VirtualSerial.c fájl 103. sorában.

4.13.1.2. void EVENT_USB_Device_ControlRequest (void)

Definíció a(z) VirtualSerial.c fájl 114. sorában.

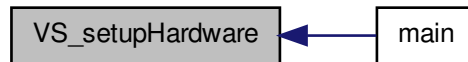
4.13.1.3. void VS_echoCharacter (void)

Definíció a(z) VirtualSerial.c fájl 57. sorában.

4.13.1.4. void VS_setupHardware (void)

Definíció a(z) VirtualSerial.c fájl 46. sorában.

Here is the caller graph for this function:



4.13.1.5. void VS_USBdataHandling (void)

Definíció a(z) VirtualSerial.c fájl 120. sorában.

Here is the caller graph for this function:



4.13.2. Változók dokumentációja

4.13.2.1. USB_ClassInfo_CDC_Device_t VirtualSerial_CDC_Interface

Kezdő érték:

```

= {
    .Config = {
        .ControlInterfaceNumber      = 0,

        .DataINEndpointNumber        = CDC_TX_EPNUM,
        .DataINEndpointSize          = CDC_TXRX_EPSIZE,
        .DataINEndpointDoubleBank    = false,

        .DataOUTEndpointNumber       = CDC_RX_EPNUM,
        .DataOUTEndpointSize         = CDC_TXRX_EPSIZE,
        .DataOUTEndpointDoubleBank   = false,

        .NotificationEndpointNumber  = CDC_NOTIFICATION_EPNUM,
        .NotificationEndpointSize    = CDC_NOTIFICATION_EPSIZE,
        .NotificationEndpointDoubleBank = false,
        .PortNumber                  = 0
    },

    .State = {
        .LineEncoding = {
            .BaudRateBPS          = 115200,
            .CharFormat            = CDC_LINEENCODING_OneStopBit,
            .ParityType            = CDC_PARITY_None,
            .DataBits              = 8
        }
    }
}
  
```

Definíció a(z) VirtualSerial.c fájl 10. sorában.

Tárgymutató

/HDD/GIT/variometer/Altimeter/src/Descriptors.c, 7	ee_write_log_rec
/HDD/GIT/variometer/Altimeter/src/HP03.c, 12	eeprom.c, 11
/HDD/GIT/variometer/Altimeter/src/Kalman.c, 19	eeprom.c
/HDD/GIT/variometer/Altimeter/src/LCD.c, 21	ee_get_first_free_address, 10
/HDD/GIT/variometer/Altimeter/src/RTC_r2051.c, 36	ee_read_log_rec, 11
/HDD/GIT/variometer/Altimeter/src/VirtualSerial.c, 43	ee_write_log_rec, 11
/HDD/GIT/variometer/Altimeter/src/eeprom.c, 9	log_rec_ext_t, 10
/HDD/GIT/variometer/Altimeter/src/itoa.c, 17	ff_addr
/HDD/GIT/variometer/Altimeter/src/logger.c, 28	log_rec_ext_s, 5
/HDD/GIT/variometer/Altimeter/src/main.c, 31	HP03.c
/HDD/GIT/variometer/Altimeter/src/mymath.c, 34	HP03_getPressure, 12
/HDD/GIT/variometer/Altimeter/src/periph.c, 34	HP03_getTemperature, 14
/HDD/GIT/variometer/Altimeter/src/tasks.c, 38	HP03_pressureSeaLevelFromAltitude, 15
actualDate	HP03_pressureToAltitude, 16
main.c, 33	HP03_readCoeffs, 16
tasks.c, 42	HP03_reset, 17
actualTime	HP03_getPressure
main.c, 33	HP03.c, 12
tasks.c, 42	HP03_getTemperature
CALLBACK_USB_GetDescriptor	HP03.c, 14
Descriptors.c, 8	HP03_pressureSeaLevelFromAltitude
calculatedSeaLevelPressure	HP03.c, 15
main.c, 33	HP03_pressureToAltitude
tasks.c, 42	HP03.c, 16
ConfigurationDescriptor	HP03_readCoeffs
Descriptors.c, 8	HP03.c, 16
Descriptors.c	HP03_reset
CALLBACK_USB_GetDescriptor, 8	HP03.c, 17
ConfigurationDescriptor, 8	i2ccfg
DeviceDescriptor, 8	periph.c, 35
LanguageString, 8	init_kalman
LanguageStringPtr, 8	Kalman.c, 20
ManufacturerString, 8	itoa
ManufacturerStringPtr, 9	itoa.c, 18
ProductString, 9	itoa.c
ProductStringPtr, 9	itoa, 18
DeviceDescriptor	kalman
Descriptors.c, 8	Kalman.c, 21
EVENT_USB_Device_ConfigurationChanged	Kalman.c
VirtualSerial.c, 43	init_kalman, 20
EVENT_USB_Device_ControlRequest	kalman, 21
VirtualSerial.c, 43	LCD.c
ee_get_first_free_address	LCD_clear, 22
eeprom.c, 10	LCD_init, 23
ee_read_log_rec	LCD_writeAlt, 23
eeprom.c, 11	

- LCD_writeDate, 25
- LCD_writePress, 25
- LCD_writeSpeed, 26
- LCD_writeTemp, 26
- LCD_writeTime, 27
- LCD_writeUSB, 27
- LCD_writeUSB_delete, 28
- LCD_clear
 - LCD.c, 22
- LCD_init
 - LCD.c, 23
- LCD_writeAlt
 - LCD.c, 23
- LCD_writeDate
 - LCD.c, 25
- LCD_writePress
 - LCD.c, 25
- LCD_writeSpeed
 - LCD.c, 26
- LCD_writeTemp
 - LCD.c, 26
- LCD_writeTime
 - LCD.c, 27
- LCD_writeUSB
 - LCD.c, 27
- LCD_writeUSB_delete
 - LCD.c, 28
- LanguageString
 - Descriptors.c, 8
- LanguageStringPtr
 - Descriptors.c, 8
- log_rec_ext_s, 5
 - ff_addr, 5
 - record_to_write, 5
- log_rec_ext_t
 - eeeprom.c, 10
- logger.c
 - logger_deleteLog, 29
 - logger_init, 29
 - logger_logThis, 29
 - logger_readFromEE, 30
 - logger_writeToEE, 30
- logger_deleteLog
 - logger.c, 29
- logger_init
 - logger.c, 29
- logger_logThis
 - logger.c, 29
- logger_readFromEE
 - logger.c, 30
- logger_writeToEE
 - logger.c, 30
- main
 - main.c, 32
- main.c
 - actualDate, 33
 - actualTime, 33
 - calculatedSeaLevelPressure, 33
 - main, 32
 - ManufacturerString
 - Descriptors.c, 8
 - ManufacturerStringPtr
 - Descriptors.c, 9
 - myPow
 - mymath.c, 34
 - mymath.c
 - myPow, 34
 - periph.c
 - i2ccfg, 35
 - pwm3c0cb, 35
 - pwm3pcb, 35
 - pwmcfg, 35
 - spicfg, 36
 - ProductString
 - Descriptors.c, 9
 - ProductStringPtr
 - Descriptors.c, 9
 - pwm3c0cb
 - periph.c, 35
 - pwm3pcb
 - periph.c, 35
 - pwmcfg
 - periph.c, 35
 - RTC_getDate
 - RTC_r2051.c, 37
 - RTC_getTime
 - RTC_r2051.c, 37
 - RTC_init
 - RTC_r2051.c, 37
 - RTC_r2051.c
 - RTC_getDate, 37
 - RTC_getTime, 37
 - RTC_init, 37
 - RTC_setDate, 38
 - RTC_setTime, 38
 - RTC_setDate
 - RTC_r2051.c, 38
 - RTC_setTime
 - RTC_r2051.c, 38
 - record_to_write
 - log_rec_ext_s, 5
 - spicfg
 - periph.c, 36
 - tasks.c
 - actualDate, 42
 - actualTime, 42
 - calculatedSeaLevelPressure, 42
 - Thread1, 39
 - Thread2, 39
 - Thread3, 40
 - Thread4, 41
 - Thread5, 42
 - Thread1

- tasks.c, 39
- Thread2
 - tasks.c, 39
- Thread3
 - tasks.c, 40
- Thread4
 - tasks.c, 41
- Thread5
 - tasks.c, 42
- VS_USBdataHandling
 - VirtualSerial.c, 44
- VS_echoCharacter
 - VirtualSerial.c, 43
- VS_setupHardware
 - VirtualSerial.c, 43
- VirtualSerial.c
 - EVENT_USB_Device_ConfigurationChanged, 43
 - EVENT_USB_Device_ControlRequest, 43
 - VS_USBdataHandling, 44
 - VS_echoCharacter, 43
 - VS_setupHardware, 43
 - VirtualSerial_CDC_Interface, 44
- VirtualSerial_CDC_Interface
 - VirtualSerial.c, 44