

altimeter

Készítette Doxygen 1.8.6

Thu Apr 2 2015 13:01:01

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. src/Descriptors.c fájlreferencia	7
4.1.1. Függvények dokumentációja	7
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. 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	10

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

4.8.2.	Változók dokumentációja	31
4.8.2.1.	actualDate	31
4.8.2.2.	actualTime	32
4.8.2.3.	binSem_T5	32
4.8.2.4.	calculatedSeaLevelPressure	32
4.8.2.5.	canT5Run	32
4.9.	src/mymath.c fájlreferencia	32
4.9.1.	Függvények dokumentációja	32
4.9.1.1.	myPow	32
4.10.	src/periph.c fájlreferencia	33
4.10.1.	Függvények dokumentációja	33
4.10.1.1.	pwm3c0cb	33
4.10.1.2.	pwm3pcb	34
4.10.2.	Változók dokumentációja	34
4.10.2.1.	i2ccfg	34
4.10.2.2.	pwmcfg	34
4.10.2.3.	spicfg	34
4.11.	src/RTC_r2051.c fájlreferencia	34
4.11.1.	Függvények dokumentációja	35
4.11.1.1.	RTC_getDate	35
4.11.1.2.	RTC_getTime	36
4.11.1.3.	RTC_init	36
4.11.1.4.	RTC_setDate	36
4.11.1.5.	RTC_setTime	36
4.12.	src/tasks.c fájlreferencia	37
4.12.1.	Függvények dokumentációja	38
4.12.1.1.	Thread1	38
4.12.1.2.	Thread2	38
4.12.1.3.	Thread3	39
4.12.1.4.	Thread4	39
4.12.1.5.	Thread5	40
4.12.2.	Változók dokumentációja	41
4.12.2.1.	actualDate	41
4.12.2.2.	actualTime	41
4.12.2.3.	binSem_T5	41
4.12.2.4.	calculatedSeaLevelPressure	41
4.12.2.5.	canT5Run	41
4.12.2.6.	VirtualSerial_CDC_Interface	41
4.13.	src/VirtualSerial.c fájlreferencia	41
4.13.1.	Függvények dokumentációja	42

4.13.1.1. EVENT_USB_Device_ConfigurationChanged	42
4.13.1.2. EVENT_USB_Device_ControlRequest	42
4.13.1.3. VS_echoCharacter	42
4.13.1.4. VS_setupHardware	43
4.13.1.5. VS_USBdataHandling	43
4.13.2. Változók dokumentációja	43
4.13.2.1. binSem_T5	43
4.13.2.2. VirtualSerial_CDC_Interface	43
Tárgymutató	45

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:

src/ Descriptors.c	7
src/ eeeprom.c	9
src/ HP03.c	11
src/ itoa.c	16
src/ Kalman.c	17
src/ LCD.c	19
src/ logger.c	26
src/ main.c	30
src/ mymath.c	32
src/ periph.c	33
src/ RTC_r2051.c	34
src/ tasks.c	37
src/ VirtualSerial.c	41

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:

- src/**eeprom.c**

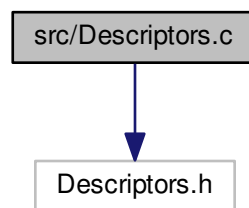
4. fejezet

Fájlok dokumentációja

4.1. 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('l'),
    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.

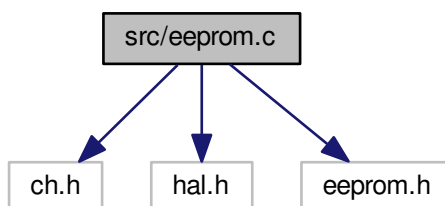
4.1.2.8. USB_Descriptor_String_t* ProductStringPtr = (USB_Descriptor_String_t *) ProductString

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

4.2. src/EEPROM.c fájlreferencia

```
#include "ch.h"
#include "hal.h"
#include "eeprom.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 144. 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) eeprom.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

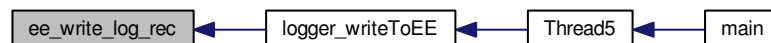
in	<i>record</i>	A beírandó rekord.
in	<i>address</i>	EEPROM cím, ahová a rekord kerül.

Visszatérési érték

0 - sikeres, < 0 - sikertelen.

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

Here is the caller graph for this function:

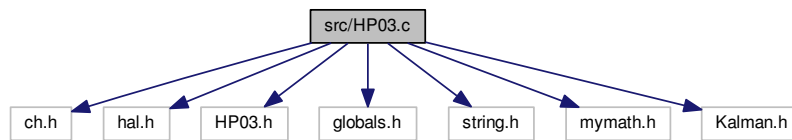
**4.3. src/HP03.c fájlreferencia**

```

#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 koefficiensek értékeit.
- **int HP03_getPressure** (HP03_meas_t *constsIn_pressureOut, bool withKalman)
Kiolvassa a légnyomás szenzorból a nyomásértéket, majd a koefficiensek segítségével kiszámítja a pontos értéket.
- **int HP03_getTemperature** (HP03_meas_t *result)
Kiolvassa a légnyomás szenzorból a hőmérsékletet, majd a koefficiensek segítségével kiszámítja a pontos értéket.

4.3.1. Függvények dokumentációja

4.3.1.1. int HP03_getPressure (HP03_meas_t * constsIn_pressureOut, bool withKalman)

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

Paraméterek

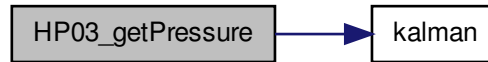
in, out	constsIn_pressureOut	Itt keletkezik az eredmény.
in	withKalman	Használja-e a Kálmán szűrőt.

Visszatérési érték

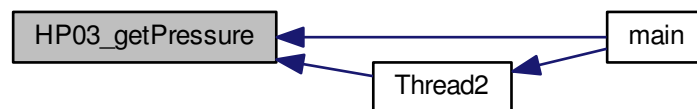
0, ha sikeres.

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

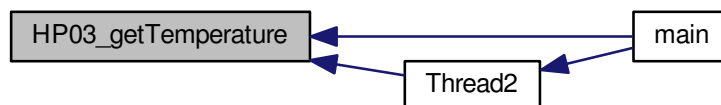
out	result	Itt keletkezik az eredmény.
-----	--------	-----------------------------

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

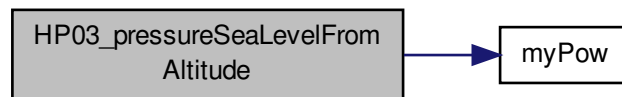
in	<i>altitude</i>	Magasság méterben.
in	<i>measuredPress-Temp</i>	Mért légnyomás és hőmérséklet.

Visszatérési érték

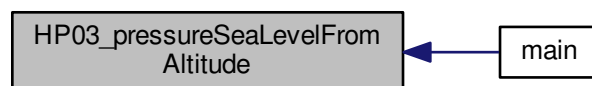
Tengerszintre átszámított légnyomás hPa-ban.

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

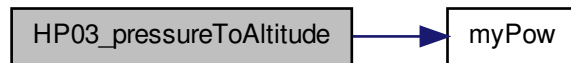
in	<i>seaLevel</i>	Tengerszintre átszámított légnyomás hPa-ban.
in	<i>measuredPress-Temp</i>	Mért légnyomás és hőmérséklet.

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 koeficiensek 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.

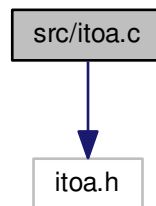
Here is the caller graph for this function:



4.4. 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 egész 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 egész karakter tömbbé konvertál.

Paraméterek

in	<i>value</i>	Az átalakítandó egész szám.
out	<i>buffer</i>	Itt keletkezik az eredmény.

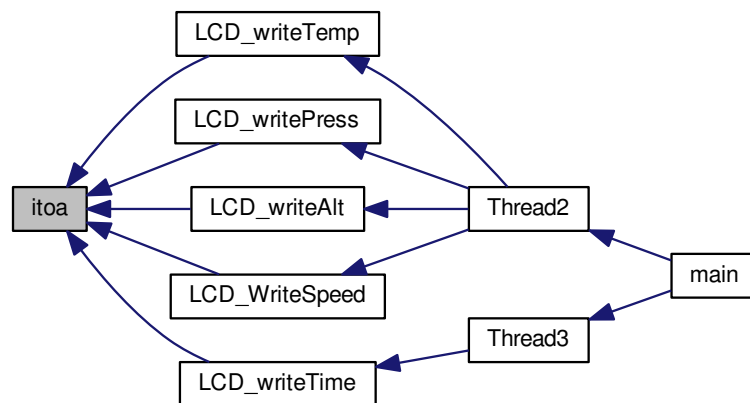
in	<i>base</i>	Alap (2; 10; 16 lehet).
in	<i>decimals</i>	Tizedesek száma.
in	<i>expectedLength</i>	Kívánt karakterhosszúság.
in	<i>padding_char</i>	Kitöltő karakter.
in	<i>prefix</i>	Előtag, mely a karakterlánc elejéhez fűződik.
in	<i>suffix</i>	Utótag, mely a karakterlánc végéhez fűződik.

Visszatérési érték

Mutató az eredmény bufferre.

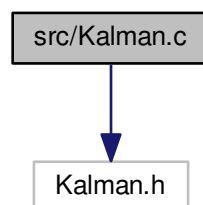
Definíció a(z) itoa.c fájl 24. sorában.

Here is the caller graph for this function:

**4.5. 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

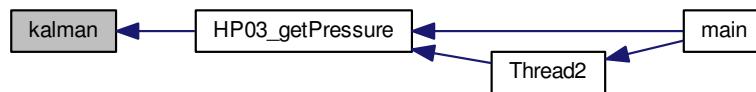
< Kálmán erősítés számítása

< Korrekció

Utolsó érték frissítése

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

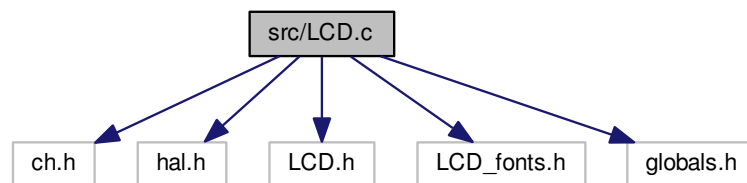
Here is the caller graph for this function:



4.6. 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_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.

- void **LCD_writeLOG** ()

LOG felirat megjelenítése a kijelzőn.

- void **LCD_writeLOG_delete** ()

LOG felirat törlése a kijelzőről.

- void **LCD_WriteSpeed** (int prevAlt, int actAlt, int deltaTime)

A függőleges sebesség megjelenítése a kijelzőn.

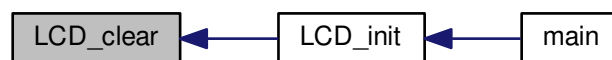
4.6.1. Függvények dokumentációja

4.6.1.1. void LCD_clear ()

Törli a kijelző tartalmát.

Definíció a(z) LCD.c fájl 148. 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 169. 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

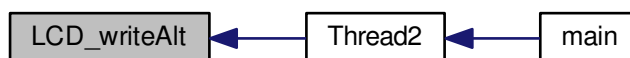
in	<i>altToWrite</i>	Magasság érték.
----	-------------------	-----------------

Definíció a(z) LCD.c fájl 250. 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 267. sorában.

4.6.1.5. void LCD_writeLOG ()

LOG felirat megjelenítése a kijelzőn.

Definíció a(z) LCD.c fájl 333. sorában.

Here is the caller graph for this function:



4.6.1.6. void LCD_writeLOG_delete ()

LOG felirat törlése a kijelzőről.

Definíció a(z) LCD.c fájl 346. sorában.

Here is the caller graph for this function:



4.6.1.7. void LCD_writePress (HP03_meas_t measuredByHP03)

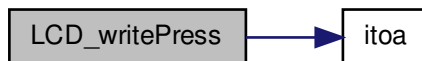
Légnyomás megjelenítése a kijelzőn.

Paraméterek

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

Definíció a(z) LCD.c fájl 225. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.8. void LCD_WriteSpeed (int *prevAlt*, int *actAlt*, int *deltaTime*)

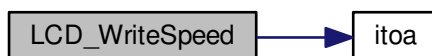
A függőleges sebesség megjelenítése a kijelzőn.

Paraméterek

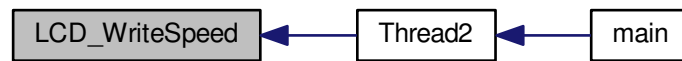
in	<i>prevAlt</i>	Az előző magasság értéke.
in	<i>actAlt</i>	Az aktuális magasság értéke.
in	<i>deltaTime</i>	A két mérés közt eltelt idő.

Definíció a(z) LCD.c fájl 362. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.9. void LCD_writeTemp (HP03_meas_t measuredByHP03)

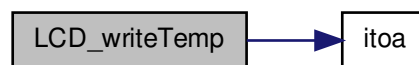
Hőmérséklet megjelenítése a kijelzőn.

Paraméterek

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

Definíció a(z) LCD.c fájl 207. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:



4.6.1.10. void LCD_writeTime (RTC_time_t timeToWrite)

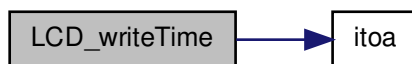
Idő megjelenítése a kijelzőn.

Paraméterek

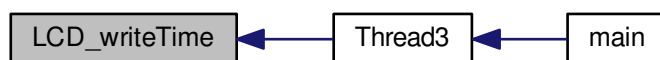
in	<i>timeToWrite</i>	Idő érték.
----	--------------------	------------

Definíció a(z) LCD.c fájl 277. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

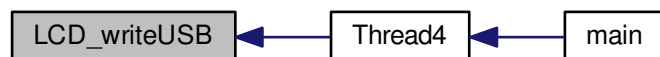


4.6.1.11. void LCD_writeUSB ()

USB felirat megjelenítése a kijelzőn.

Definíció a(z) LCD.c fájl 302. sorában.

Here is the caller graph for this function:



4.6.1.12. void LCD_writeUSB_delete ()

USB felirat törlése a kijelzőről.

Definíció a(z) LCD.c fájl 320. sorában.

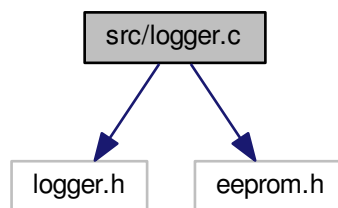
Here is the caller graph for this function:



4.7. 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.

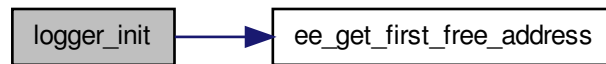
Definíció a(z) logger.c fájl 107. sorában.

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

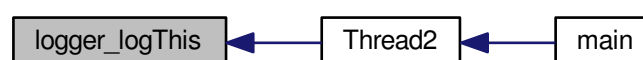
in	rec_to_log	A naplózandó rekord.
----	------------	----------------------

Visszatérési érték

0 - normál működés, 1 - túlcsordulás, 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	buffer	Mutató az eredménybufferre.
in	size_in_rec	Rekordok száma.

Visszatérési érték

A következő kiolvasható rekord címe.

Definíció a(z) logger.c fájl 98. 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:



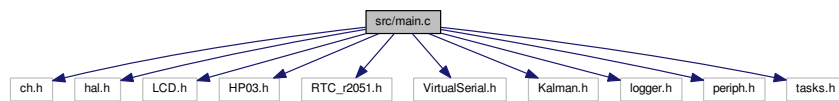
Here is the caller graph for this function:



4.8. 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)

A szoftver belépési pontja.

Változók

- volatile RTC_time_t **actualTime**
- volatile RTC_date_t **actualDate**
- volatile float **calculatedSeaLevelPressure** = 0
- BinarySemaphore **binSem_T5**
- volatile bool_t **canT5Run** = FALSE

4.8.1. Függvények dokumentációja

4.8.1.1. int main (void)

A szoftver belépési pontja.

Rendszer-inicializálás.

- HAL inicializálás (SoC és kártya specifikus)
- Kernel inicializálás, a **main()** (o. 30) függvényből szól 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.

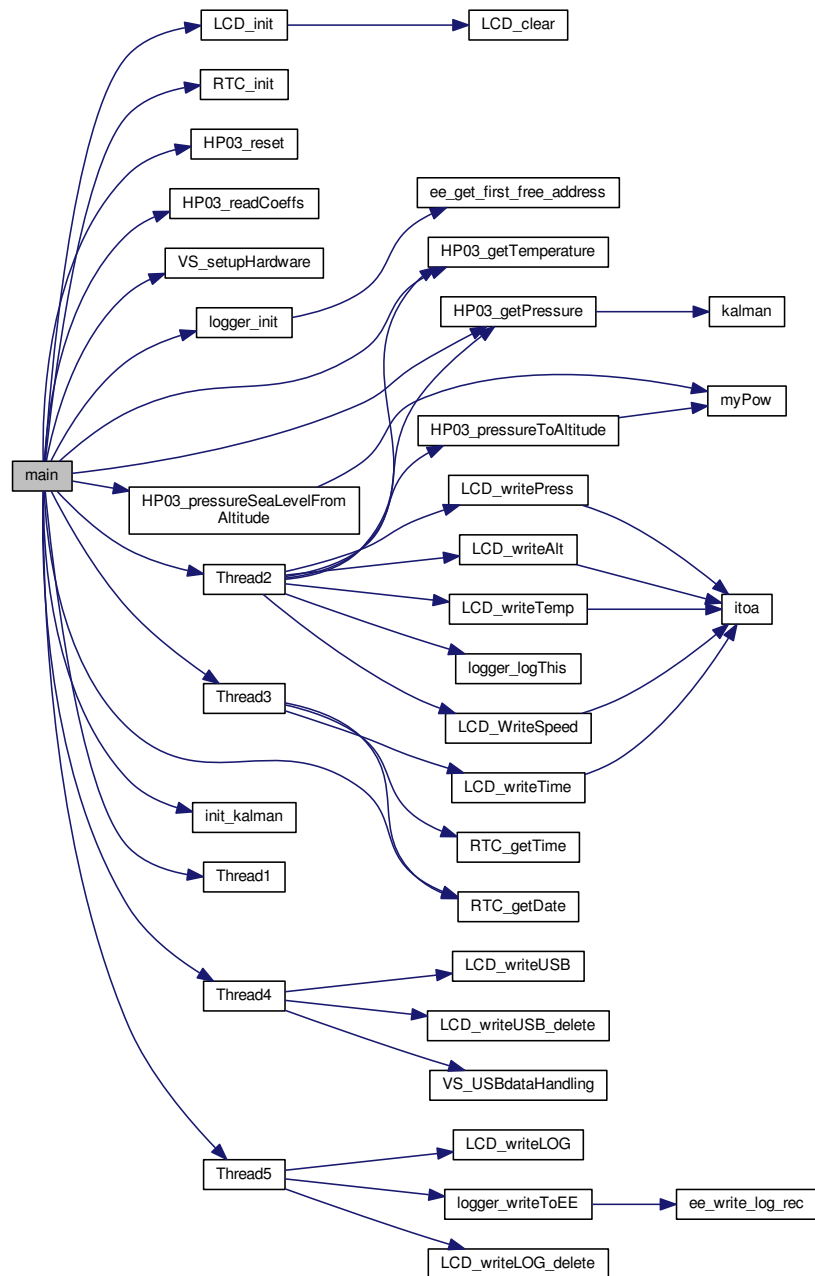
Hangszóró tesztelése (PWM).

Háttérvilágítás bekapcsolása.

Szálak létrehozása.

Definíció a(z) main.c fájl 25. 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. `BinarySemaphore binSem_T5`

Definíció a(z) `main.c` fájl 19. sorában.

4.8.2.4. `volatile float calculatedSeaLevelPressure = 0`

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

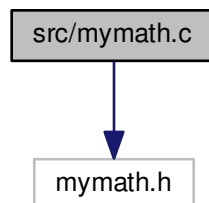
4.8.2.5. `volatile bool_t canT5Run = FALSE`

Definíció a(z) `main.c` fájl 20. sorában.

4.9. `src/mymath.c` fájlreferencia

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

A `mymath.c` definíciós fájl függési gráfja:



Függvények

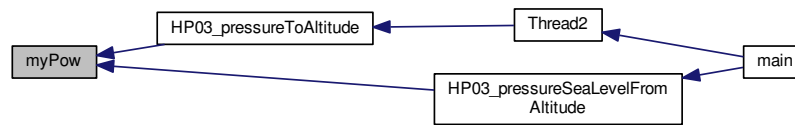
- float **myPow** (float *a*, float *b*)

4.9.1. Függvények dokumentációja

4.9.1.1. float myPow (float *a*, float *b*)

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

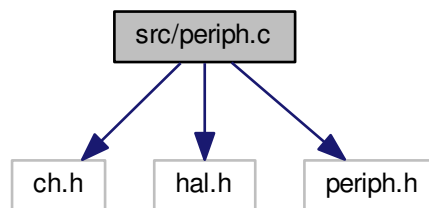
Here is the caller graph for this function:



4.10. 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)
Callback függvény.
- void **pwm3c0cb** (PWMDriver *pwmp)

Változók

- SPIConfig **spicfg**
SPI konfiguráció (1MHz, CPHA=0, CPOL=0).
- I2CConfig **i2ccfg**
I2C konfiguráció (400kHz).
- PWMConfig **pwmcfg**
PWM konfiguráció.

4.10.1. Függvények dokumentációja

4.10.1.1. void pwm3c0cb (PWMDriver * pwmp)

Definíció a(z) `periph.c` fájl 53. sorában.

4.10.1.2. void pwm3pcb (PWMDriver * pwmp)

Callback függvény.

Definíció a(z) periph.c fájl 47. sorában.

4.10.2. Változók dokumentációja

4.10.2.1. I2CConfig i2ccfg

Kezdő érték:

```
= {
    I2C_FAST_MODE_PLUS,
    48
}
```

I2C konfiguráció (400kHz).

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}
    }
}
```

PWM konfiguráció.

Definíció a(z) periph.c fájl 34. sorában.

4.10.2.3. SPIConfig spicfg

Kezdő érték:

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

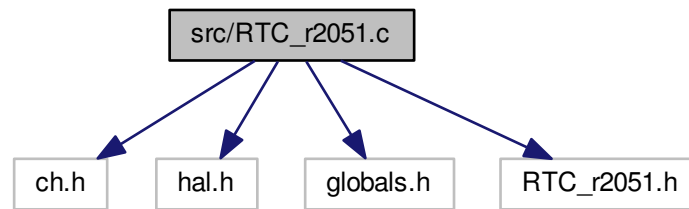
SPI konfiguráció (1MHz, CPHA=0, CPOL=0).

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

4.11. 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 ()

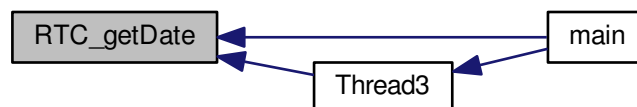
A dátum kiolvasása az RTC áramkörből.

Visszatérési érték

Az aktuális dátum.

Definíció a(z) RTC_r2051.c fájl 141. sorában.

Here is the caller graph for this function:



4.11.1.2. `RTC_time_t RTC_getTime ()`

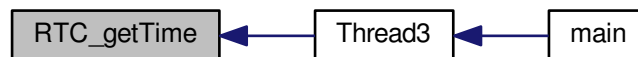
Az idő kiolvasása az RTC áramkörből.

Visszatérési érték

Az aktuális idő.

Definíció a(z) `RTC_r2051.c` fájl 103. 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

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

Definíció a(z) `RTC_r2051.c` fájl 75. sorában.

4.11.1.5. `void RTC_setTime (RTC_time_t timeToBeSet)`

Beállítja az időt az RTC áramkörben.

Paraméterek

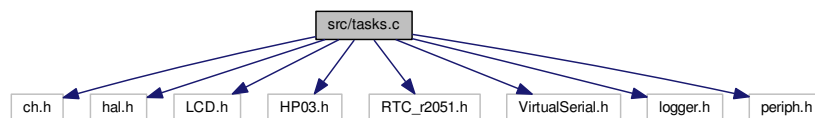
in	timeToBeSet	A beállítandó idő.
----	-------------	--------------------

Definíció a(z) RTC_r2051.c fájl 49. sorában.

4.12. 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**
- BinarySemaphore **binSem_T5**
- bool_t **canT5Run**
- USB_ClassInfo_CDC_Device_t **VirtualSerial_CDC_Interface**
LPCUSBlib CDC Class driver interfész konfiguráció és állapot-információ.

4.12.1. Függvények dokumentációja

4.12.1.1. `msg_t Thread1 (void * arg)`

"HeartBeat" LED villogtató szál.

Definíció a(z) `tasks.c` fájl 27. sorában.

Here is the caller graph for this function:

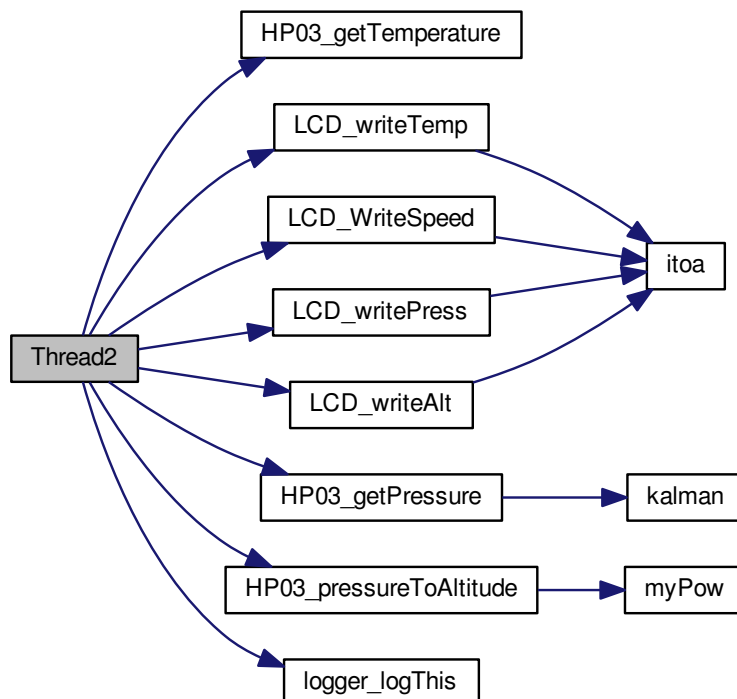


4.12.1.2. `msg_t Thread2 (void * arg)`

Nyomás és hőmérséklet kiolvasó szál.

Definíció a(z) `tasks.c` fájl 44. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

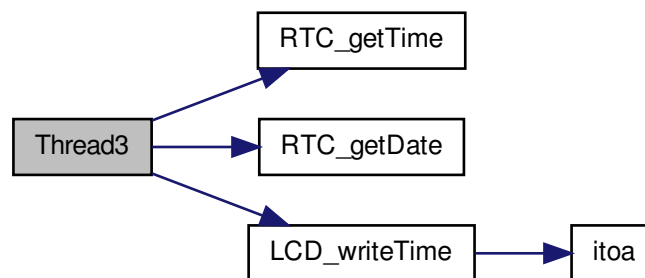


4.12.1.3. msg_t Thread3 (void * arg)

Dátum és idő kezelő szál.

Definíció a(z) tasks.c fájl 113. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

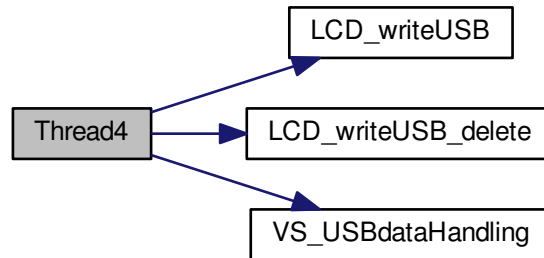


4.12.1.4. msg_t Thread4 (void * arg)

USB kezelő szál.

Definíció a(z) tasks.c fájl 136. sorában.

A függvény hívási gráfja:



Here is the caller graph for this function:

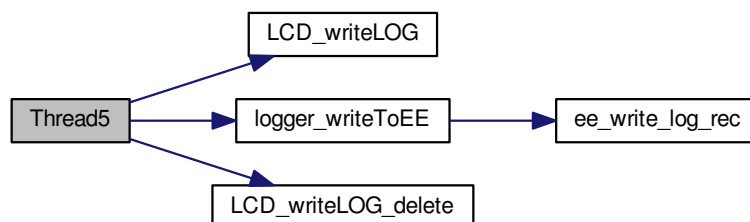


4.12.1.5. msg_t Thread5 (void * arg)

EEPROM író szál (ringbuffer -> EEPROM).

Definíció a(z) tasks.c fájl 164. 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. BinarySemaphore binSem_T5

Definíció a(z) main.c fájl 19. sorában.

4.12.2.4. volatile float calculatedSeaLevelPressure

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

4.12.2.5. bool_t canT5Run

Definíció a(z) main.c fájl 20. sorában.

4.12.2.6. USB_ClassInfo_CDC_Device_t VirtualSerial_CDC_Interface

LPCUSBLib CDC Class driver interfész konfiguráció és állapot-információ.

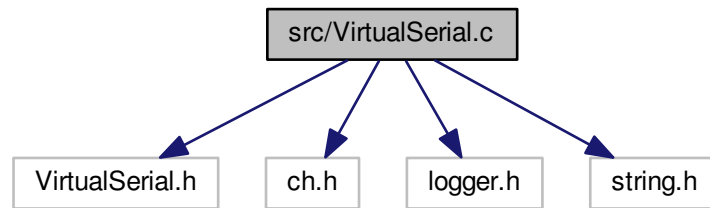
Ez a struktúra adódik át minden CDC Class driver függvénynek.

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

4.13. 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)
A hardver és a chip perifériáinak beállítása.
- void **VS_echoCharacter** (void)
- void **EVENT_USB_Device_ConfigurationChanged** (void)
Eseménykezelő az USB könyvtár "Configuration Changed" eseményéhez.
- void **EVENT_USB_Device_ControlRequest** (void)
Eseménykezelő az USB könyvtár "Control Request reception" eseményéhez.
- void **VS_USBdataHandling** (void)
Az USB-n keresztül érkező adatokat kezeli le.

Változók

- BinarySemaphore **binSem_T5**
- USB_ClassInfo_CDC_Device_t **VirtualSerial_CDC_Interface**
LPCUSBLib CDC Class driver interfész konfiguráció és állapot-információ.

4.13.1. Függvények dokumentációja

4.13.1.1. void EVENT_USB_Device_ConfigurationChanged (void)

Eseménykezelő az USB könyvtár "Configuration Changed" eseményéhez.

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

4.13.1.2. void EVENT_USB_Device_ControlRequest (void)

Eseménykezelő az USB könyvtár "Control Request reception" eseményéhez.

Definíció a(z) VirtualSerial.c fájl 118. sorában.

4.13.1.3. void VS_echoCharacter (void)

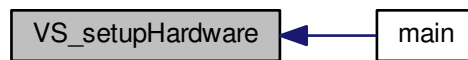
Definíció a(z) VirtualSerial.c fájl 61. sorában.

4.13.1.4. void VS_setupHardware (void)

A hardver és a chip perifériáinak beállítása.

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

Here is the caller graph for this function:



4.13.1.5. void VS_USBdataHandling (void)

Az USB-n keresztül érkező adatokat kezeli le.

Definíció a(z) VirtualSerial.c fájl 125. sorában.

Here is the caller graph for this function:



4.13.2. Változók dokumentációja

4.13.2.1. BinarySemaphore binSem_T5

Definíció a(z) main.c fájl 19. sorában.

4.13.2.2. 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
        }
    }
}
```

LPCUSBLib CDC Class driver interfész konfiguráció és állapot-információ.

Ez a struktúra adódik át minden CDC Class driver függvénynek.

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

Tárgymutató

actualDate	log_rec_ext_t, 10
main.c, 31	
tasks.c, 41	
actualTime	ff_addr
main.c, 31	log_rec_ext_s, 5
tasks.c, 41	
binSem_T5	HP03.c
main.c, 32	HP03_getPressure, 12
tasks.c, 41	HP03_getTemperature, 13
VirtualSerial.c, 43	HP03_pressureSeaLevelFromAltitude, 13
	HP03_pressureToAltitude, 14
	HP03_readCoeffs, 15
	HP03_reset, 15
	HP03_getPressure
	HP03.c, 12
	HP03_getTemperature
	HP03.c, 13
	HP03_pressureSeaLevelFromAltitude
	HP03.c, 13
	HP03_pressureToAltitude
	HP03.c, 14
	HP03_readCoeffs
	HP03.c, 15
	HP03_reset
	HP03.c, 15
	i2ccfg
	periph.c, 34
	init_kalman
	Kalman.c, 18
	itoa
	itoa.c, 16
	itoa.c
	itoa, 16
	kalman
	Kalman.c, 18
	Kalman.c
	init_kalman, 18
	kalman, 18
	LCD.c
	LCD_WriteSpeed, 23
	LCD_clear, 20
	LCD_init, 20
	LCD_writeAlt, 21
	LCD_writeDate, 21
	LCD_writeLOG, 22
	LCD_writeLOG_delete, 22
	LCD_writePress, 22
	LCD_writeTemp, 24
CALLBACK_USB_GetDescriptor	
Descriptors.c, 7	
calculatedSeaLevelPressure	
main.c, 32	
tasks.c, 41	
canT5Run	
main.c, 32	
tasks.c, 41	
ConfigurationDescriptor	
Descriptors.c, 8	
Descriptors.c	
CALLBACK_USB_GetDescriptor, 7	
ConfigurationDescriptor, 8	
DeviceDescriptor, 8	
LanguageString, 8	
LanguageStringPtr, 8	
ManufacturerString, 8	
ManufacturerStringPtr, 9	
ProductString, 9	
ProductStringPtr, 9	
DeviceDescriptor	
Descriptors.c, 8	
EVENT_USB_Device_ConfigurationChanged	
VirtualSerial.c, 42	
EVENT_USB_Device_ControlRequest	
VirtualSerial.c, 42	
ee_get_first_free_address	
eeprom.c, 10	
ee_read_log_rec	
eeprom.c, 10	
ee_write_log_rec	
eeprom.c, 11	
eeprom.c	
ee_get_first_free_address, 10	
ee_read_log_rec, 10	
ee_write_log_rec, 11	

LCD_writeTime, 24
 LCD_writeUSB, 25
 LCD_writeUSB_delete, 25
 LCD_WriteSpeed
 LCD.c, 23
 LCD_clear
 LCD.c, 20
 LCD_init
 LCD.c, 20
 LCD_writeAlt
 LCD.c, 21
 LCD_writeDate
 LCD.c, 21
 LCD_writeLOG
 LCD.c, 22
 LCD_writeLOG_delete
 LCD.c, 22
 LCD_writePress
 LCD.c, 22
 LCD_writeTemp
 LCD.c, 24
 LCD_writeTime
 LCD.c, 24
 LCD_writeUSB
 LCD.c, 25
 LCD_writeUSB_delete
 LCD.c, 25
 LanguageString
 Descriptors.c, 8
 LanguageStringPtr
 Descriptors.c, 8
 log_rec_ext_s, 5
 ff_addr, 5
 record_to_write, 5
 log_rec_ext_t
 eeprom.c, 10
 logger.c
 logger_deleteLog, 26
 logger_init, 26
 logger_logThis, 27
 logger_readFromEE, 27
 logger_writeToEE, 29
 logger_deleteLog
 logger.c, 26
 logger_init
 logger.c, 26
 logger_logThis
 logger.c, 27
 logger_readFromEE
 logger.c, 27
 logger_writeToEE
 logger.c, 29
 main
 main.c, 30
 main.c
 actualDate, 31
 actualTime, 31
 binSem_T5, 32
 calculatedSeaLevelPressure, 32
 canT5Run, 32
 main, 30
 ManufacturerString
 Descriptors.c, 8
 ManufacturerStringPtr
 Descriptors.c, 9
 myPow
 mymath.c, 32
 mymath.c
 myPow, 32
 periph.c
 i2ccfg, 34
 pwm3c0cb, 33
 pwm3pcb, 33
 pwmcfg, 34
 spicfg, 34
 ProductString
 Descriptors.c, 9
 ProductStringPtr
 Descriptors.c, 9
 pwm3c0cb
 periph.c, 33
 pwm3pcb
 periph.c, 33
 pwmcfg
 periph.c, 34
 RTC_getDate
 RTC_r2051.c, 35
 RTC_getTime
 RTC_r2051.c, 35
 RTC_init
 RTC_r2051.c, 36
 RTC_r2051.c
 RTC_getDate, 35
 RTC_getTime, 35
 RTC_init, 36
 RTC_setDate, 36
 RTC_setTime, 36
 RTC_setDate
 RTC_r2051.c, 36
 RTC_setTime
 RTC_r2051.c, 36
 record_to_write
 log_rec_ext_s, 5
 spicfg
 periph.c, 34
 src/Descriptors.c, 7
 src/HP03.c, 11
 src/Kalman.c, 17
 src/LCD.c, 19
 src/RTC_r2051.c, 34
 src/VirtualSerial.c, 41
 src/eeprom.c, 9
 src/itoa.c, 16
 src/logger.c, 26

- src/main.c, 30
- src/mymath.c, 32
- src/periph.c, 33
- src/tasks.c, 37
- tasks.c
 - actualDate, 41
 - actualTime, 41
 - binSem_T5, 41
 - calculatedSeaLevelPressure, 41
 - canT5Run, 41
 - Thread1, 38
 - Thread2, 38
 - Thread3, 39
 - Thread4, 39
 - Thread5, 40
 - VirtualSerial_CDC_Interface, 41
- Thread1
 - tasks.c, 38
- Thread2
 - tasks.c, 38
- Thread3
 - tasks.c, 39
- Thread4
 - tasks.c, 39
- Thread5
 - tasks.c, 40
- VS_USBdataHandling
 - VirtualSerial.c, 43
- VS_echoCharacter
 - VirtualSerial.c, 42
- VS_setupHardware
 - VirtualSerial.c, 42
- VirtualSerial.c
 - binSem_T5, 43
 - EVENT_USB_Device_ConfigurationChanged, 42
 - EVENT_USB_Device_ControlRequest, 42
 - VS_USBdataHandling, 43
 - VS_echoCharacter, 42
 - VS_setupHardware, 42
 - VirtualSerial_CDC_Interface, 43
- VirtualSerial_CDC_Interface
 - tasks.c, 41
 - VirtualSerial.c, 43