BioSens

1.0

Generated by Doxygen 1.9.1

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Block Class Reference

`#include <Block.h>`

Public Member Functions

- template<size_t N>
  void init (Data(&_bloques)[N], unsigned int _offset=0)
- void initAllData (void)
- ~Block (void)
- Data getDataObject (uint8_t _index)
- Data getDataObject (String _dataName)
- unsigned int getBlockOffset (void)
- void setBlockOffset (unsigned int _offset)

2.1.1 Detailed Description

Block class. Class for joint data management.
This class is in charge of managing a data set (Data.h) integrated in itself, so that we can more easily access certain functionalities of the data structure to be created.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 ~Block()

```
Block::~Block (
            void  )
```

Destroy the instance.

2.1.3 Member Function Documentation

2.1.3.1 getBlockOffset()

```
unsigned int Block::getBlockOffset (
            void  )
```

Get the whole block offset on EEPROM.

Returns

The block offset on EEPROM.

2.1.3.2 getDataObject() [1/2]

```
Data Block::getDataObject (
            String _dataName )
```

Get a Data object on block according to its name.

Parameters

_dataName	The name of the Data to search.

Returns

The Data object requested.

2.1.3.3 getDataObject() [2/2]

```
Data Block::getDataObject (
            uint8_t _index )
```

Get the indexed Data object.

Parameters

_index	The Data index on block.

Returns

The Data object requested.

2.1.3.4 init()

```
template<size_t N>
void Block::init (
            Data(&) _bloques[N],
            unsigned int _offset = 0 )  [inline]
```

Initialize the Block charging the Data on it.

Parameters

_bloques	The Data structure on an array.
_offset	The whole block offset.

Returns

The Data object requested.

2.1.3.5 initAllData()

```
void Block::initAllData (
            void  )
```

Initialize all the stored Data in the EEPROM.

initAllData is in charge of execute the init() method on every Data saved on the block.
This will initialize the Data in EEPROM. Only for first time writing Data.

Returns

Nothing

2.1.3.6 setBlockOffset()

```
void Block::setBlockOffset (
            unsigned int _offset )
```

Set the whole block offset on EEPROM.

Parameters

_offset	- The new block offset on EEPROM.

Returns

Nothing.

The documentation for this class was generated from the following files:

- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/Block/src/Block.h
- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/Block/src/Block.cpp

2.2 Data Class Reference

`#include <Data.h>`

Public Member Functions

- Data (int16_t _offset=-1, String _data="")
- Data (String _name, int16_t _offset=-1, String _data="")
- void init (void)
- String getData (void)
- String getStaticData (void)
- String getName (void)
- unsigned int getLength (void)
- int16_t getOffset (void)
- int16_t getEndOfData (void)
- void setOffset (int16_t _offset)
- void setData (String _data)

2.2.1 Detailed Description

Data class. Class for save and read data in EEPROM.
This class is in charge of create and manage a single data on the RTC EEPROM.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 Data() [1/2]

```
Data::Data (
            int16_t _offset = -1,
            String _data = "" )
```

Constructor to the Block charging the data on it.

Parameters

_offset	The data offset on the EEPROM.
_data	The data as a String.

2.2.2.2 Data() [2/2]

```
Data::Data (
            String _name,
            int16_t _offset = -1,
            String _data = "" )
```

Constructor to the Block charging the data on it.

Parameters

_name	The data name for search it in a Block using this Data object.
_offset	The data offset on the EEPROM.
_data	The data as a String.

2.2.3 Member Function Documentation

2.2.3.1 getData()

```
String Data::getData (
            void  )
```

Get the data reading on the EEPROM starting from the indicated offset to the end of the data.

Returns

The data saved on EEPROM as a String.

2.2.3.2 getEndOfData()

```
int16_t Data::getEndOfData (
            void  )
```

Get the static end of data byte in EEPROM

Returns

The data's end of data Byte.

2.2.3.3 getLength()

```
unsigned int Data::getLength (
            void  )
```

Get the length of data in EEPROM on bytes.

Returns

The data Length.

2.2.3.4 getName()

```
String Data::getName (
            void  )
```

Get the name of the data. It's not in the EEPROM.

Returns

The data's name.

2.2.3.5 getOffset()

```
int16_t Data::getOffset (
            void  )
```

Get the offset byte of data in EEPROM.

Returns

The data's byte offset.

2.2.3.6 getStaticData()

```
String Data::getStaticData (
            void  )
```

Get the static data setted on Constructor, not the EEPROM saved data.

Returns

The static data.

2.2.3.7 init()

```
void Data::init (
            void  )
```

Initialize the data in the EEPROM. Saves the data in the eeprom using the offset selected only if there is a data defined.

Returns

Nothing.

2.2.3.8 setData()

```
void Data::setData (
            String _data )
```

Change the data writing in EEPROM.

Parameters

_data	The new data to write in EEPROM.

Returns

Nothing.

2.2.3.9 setOffset()

```
void Data::setOffset (
            int16_t _offset )
```

Set the offset byte of data in EEPROM.

Parameters

_offset	The new data offset on EEPROM.

Returns

Nothing.

The documentation for this class was generated from the following files:

- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ Data/src/Data.h
- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ Data/src/Data.cpp

2.3 Environment Class Reference

`#include <Environment.h>`

Public Member Functions

- Environment (Block *_B, RTCuse *_rtc, int8_t _pinGeneral=-1, int8_t _pinTermica=-1, int8_t _pinGrupos=-1)
- int8_t getGeneralPin (void)
- int8_t getTermicaPin (void)
- int8_t getGruposPin (void)
- Data getPhone (uint8_t _index)
- String getPhoneString (uint8_t _index)
- Data getAlertTime (void)
- int getAlertTimeInt (void)
- void setGeneralPin (int8_t _pinGeneral)
- void setTermicaPin (int8_t _pinTermica)
- void setGruposPin (int8_t _pinGrupos)
- String runMenuOption (menuOptions _option, String _arg="")

Static Public Member Functions

- static int getOptionID (menuOptions _option)
- static menuOptions getOptionByID (int _id)
- static menuOptions keywordToOption (String _keyword)
- static bool isAPhoneNumber (String _number)

2.3.1 Detailed Description

Environment class. Class for managin all the most important software actionss.
This class is in charge of the simple manage of the project environment and provide simplicity and clarity to the main code.

2.3.2 Constructor & Destructor Documentation

2.3.2.1 Environment()

```
Environment::Environment (
            Block * _B,
            RTCuse * _rtc,
            int8_t _pinGeneral = -1,
            int8_t _pinTermica = -1,
            int8_t _pinGrupos = -1 )  [inline]
```

Environment constructor.

Parameters

_B	The block addres for its use in the environment.
_rtc	The block addres for its EEPROM use in the environment.
_pinGeneral	Pin for General current verification.
_pinTermica	Pin for Termica current verification.
_pinGrupos	Pin for Grupos current verification.

2.3.3 Member Function Documentation

2.3.3.1 getAlertTime()

```
Data Environment::getAlertTime (
            void  )
```

Get the alerts time Data object.

Returns

Alert time Data object.

2.3.3.2 getAlertTimeInt()

```
int Environment::getAlertTimeInt (
            void  )
```

Get the alerts time configured.

Returns

Alert time on minutes as an int.

2.3.3.3 getGeneralPin()

```
int8_t Environment::getGeneralPin (
            void  )
```

Get the General configured pin.

Returns

The pin of General Current.

2.3.3.4 getGruposPin()

```
int8_t Environment::getGruposPin (
            void  )
```

Get the Grupos configured pin.

Returns

The pin of Generator sets Current.

2.3.3.5 getOptionByID()

```
menuOptions Environment::getOptionByID (
            int _id )  [static]
```

Get the option enum type through its ID or index.

See also

menuOptions

Parameters

↩ _↩ *id*	Option ID or index.

Returns

The enum type related to the numeric ID option.

2.3.3.6 getOptionID()

```
int Environment::getOptionID (
            menuOptions _option )  [static]
```

Get the option ID through its enum type.

See also

menuOptions

Parameters

_option	Option enum type to convert to ID or index.

Returns

A numeric ID or index for the option selected.

2.3.3.7 getPhone()

```
Data Environment::getPhone (
            uint8_t _index )
```

Get the indexed phone number.

Parameters

_index	Index of Phone groove on EEPROM. (0-2)

Returns

The Data object of the phone requested.

2.3.3.8 getPhoneString()

```
String Environment::getPhoneString (
            uint8_t _index )
```

Get the indexed phone number as a String.

Parameters

_index	Index of Phone groove on EEPROM. (0-2)

Returns

The requested phone number as a String.

2.3.3.9 getTermicaPin()

```
int8_t Environment::getTermicaPin (
            void  )
```

Get the Termica configured pin.

Returns

The pin of Termica Current.

2.3.3.10 isAPhoneNumber()

```
bool Environment::isAPhoneNumber (
            String _number )  [static]
```

Verify if a String is a phone number or not.

Parameters

_number	String to analize.

Return values

true	Is a phone number.
false	Is not a phone number.

2.3.3.11 keywordToOption()

```
menuOptions Environment::keywordToOption (
            String _keyword )  [static]
```

Get the option enum type through its keyword.

See also

menuOptions

Parameters

_keyword	Keyword choosen for refer to the option.

Returns

The enum type related to the keyword option.

2.3.3.12 runMenuOption()

```
String Environment::runMenuOption (
            menuOptions _option,
            String _arg = "" )
```

Run the option requested.

See also

menuOtions

Parameters

_option	Option to run (menuOptions).
_arg	Extra arguments if required.

Returns

A String with the response to the action performed.

2.3.3.13 setGeneralPin()

```
void Environment::setGeneralPin (
            int8_t _pinGeneral )
```

Set the General pin.

Parameters

_pinGeneral	New Pin for General.

Returns

Nothing.

2.3.3.14 setGruposPin()

```
void Environment::setGruposPin (
            int8_t _pinGrupos )
```

Set the Grupos pin.

Parameters

_pinGeneral	New Pin for Grupos.

Returns

Nothing.

2.3.3.15 setTermicaPin()

```
void Environment::setTermicaPin (
            int8_t _pinTermica )
```

Set the Termica pin.

Parameters

_pinGeneral	New Pin for Termica.

Returns

Nothing.

The documentation for this class was generated from the following files:

- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ Environment/src/Environment.h
- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ Environment/src/Environment.cpp

2.4 LCDuse Class Reference

```
#include <LCDuse.h>
```

Public Member Functions

- void init (byte _address=0x27, uint8_t _columns=16, uint8_t _rows=2)
- void print (String _text1, String _text2="", unsigned short int inicPos1=0, unsigned short int inicPos2=0)
- void print (String _text1, Data _data2, unsigned short int inicPos1=0, unsigned short int inicPos2=0)
- void print (Data _data1, String _text2="", unsigned short int inicPos1=0, unsigned short int inicPos2=0)
- void print (Data _data1, Data _data2, unsigned short int inicPos1=0, unsigned short int inicPos2=0)

2.4.1 Detailed Description

LCDuse class. Class for LCD management.
This class is in charge of setup and use the LCD Screen with I2C connections.

2.4.2 Member Function Documentation

2.4.2.1 init()

```
void LCDuse::init (
            byte _address = 0x27,
            uint8_t _columns = 16,
            uint8_t _rows = 2 )
```

Initialize the LCD setup and reset it for its correctly use.

Parameters

_address	The I2C LCD address.
_columns	Number of LCD char columns.
_rows	Number of LCD char rows.

Returns

Nothing.

2.4.2.2 print() [1/4]

```
void LCDuse::print (
            Data _data1,
            Data _data2,
            unsigned short int inicPos1 = 0,
            unsigned short int inicPos2 = 0 )
```

Print on the screen (Only for 2 rows LCD device).

Parameters

_data1	Data object to print in first row.
_data2	Data object to print in second row.
inicPos1	Column start position in the first row.
inicPos2	Column start position in the second row.

Returns

Nothing.

2.4.2.3 print() [2/4]

```
void LCDuse::print (
            Data _data1,
            String _text2 = "",
            unsigned short int inicPos1 = 0,
            unsigned short int inicPos2 = 0 )
```

Print on the screen (Only for 2 rows LCD device).

Parameters

_data1	Data object to print in first row.
_text2	Text to print in second row.
inicPos1	Column start position in the first row.
inicPos2	Column start position in the second row.

Returns

Nothing.

2.4.2.4 print() [3/4]

```
void LCDuse::print (
            String _text1,
            Data _data2,
            unsigned short int inicPos1 = 0,
            unsigned short int inicPos2 = 0 )
```

Print on the screen (Only for 2 rows LCD device).

Parameters

_text1	Text to print in first row.
_data2	Data object to print in second row.
inicPos1	Column start position in the first row.
inicPos2	Column start position in the second row.

Returns

Nothing.

2.4.2.5 print() [4/4]

```
void LCDuse::print (
            String _text1,
```

```
String _text2 = "",
unsigned short int inicPos1 = 0,
unsigned short int inicPos2 = 0 )
```

Print on the screen (Only for 2 rows LCD device).

Parameters

_text1	Text to print in first row.
_text2	Text to print in second row.
inicPos1	Column start position in the first row.
inicPos2	Column start position in the second row.

Returns

Nothing.

The documentation for this class was generated from the following files:

- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ LCDuse/src/LCDuse.h
- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ LCDuse/src/LCDuse.cpp

2.5 RTCuse Class Reference

```
#include <RTCuse.h>
```

Public Member Functions

- void saveInEEPROM (uint16_t _offset, char _dato)
- void saveInEEPROM (uint16_t _offset, String _dato)
- char readInEEPROM (uint16_t _offset)
- String readInEEPROM (uint16_t _offset, unsigned int _length)
- void adjustDate (void)
- DateTime getDateTime (void)
- String getDate (void)

2.5.1 Detailed Description

RTCuse class. Class for RTC time and EEPROM management.
This class is in charge of managing the RTC EEPROM and provide the time and date using the same module.

2.5.2 Member Function Documentation

2.5.2.1 adjustDate()

```
void RTCuse::adjustDate (
            void  )
```

Adjust the Date when the program is uploaded from the computer.

Returns

Nothing.

2.5.2.2 getDate()

```
String RTCuse::getDate (
            void  )
```

Get the actual date as a String.

Returns

Parsed String who indicates the date and time DD/MM/YYYY - hh:mm:ss.

2.5.2.3 getDateTime()

```
DateTime RTCuse::getDateTime (
            void  )
```

Get the actual date and hour in a DateTime object.

Returns

DateTime object with the actual date and hour.

2.5.2.4 readInEEPROM() [1/2]

```
char RTCuse::readInEEPROM (
            uint16_t _offset )
```

Read a character in EEPROM.

Parameters

_offset	The byte where is the character that we want to read.

Returns

Character on _offset byte in EEPROM.

2.5.2.5 readInEEPROM() [2/2]

```
String RTCuse::readInEEPROM (
            uint16_t _offset,
            unsigned int _length )
```

Read a String in EEPROM.

Parameters

_offset	The initial byte where is the String that we want to read.
_length	The length of the data we want to read.

Returns

String from _offset byte in EEPROM of size _length.

2.5.2.6 saveInEEPROM() [1/2]

```
void RTCuse::saveInEEPROM (
            uint16_t _offset,
            char _dato )
```

Save a character in the _offset byte in EEPROM.

Parameters

_offset	The byte where will be stored the character on EEPROM.
_dato	The character to save in EEPROM.

Returns

Nothing.

2.5.2.7 saveInEEPROM() [2/2]

```
void RTCuse::saveInEEPROM (
            uint16_t _offset,
            String _dato )
```

Save a String in EEPROM starting on the _offset byte.

Parameters

_offset	The byte where will be stored the String on EEPROM.
_dato	The String to save in EEPROM.

Returns

Nothing.

The documentation for this class was generated from the following files:

- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/RTCuse/src/RTCuse.h
- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/RTCuse/src/RTCuse.cpp

2.6 SMSend Class Reference

`#include <SMSend.h>`

Public Member Functions

- SMSend (uint8_t _pinRX, uint8_t _pinTX, int _baudRate)
- void sendSMS (String _phoneNumber, String _message)
- String receiveSMS ()
- bool isSMS (String _message)
- String getMessageSMS (String _message)
- String getMessagePhone (String _message)
- String getMessageDate (String _message)
- String getMessageInfo (datos_mensaje _option, String _message)

2.6.1 Detailed Description

SMSend class. Class for sending and receiving SMS with SIM800L.
This class is in charge of send and receive SMS messages and access its information.

2.6.2 Constructor & Destructor Documentation

2.6.2.1 SMSend()

```
SMSend::SMSend (
            uint8_t _pinRX,
            uint8_t _pinTX,
            int _baudRate )
```

SMSend object constructor for Arduino UNO, Leonardo and more.

Parameters

_pinRX	RX pin for new Software Serial.
_pinTX	TX pin for new Software Serial.
_baudRate	The baud rate of the communication with SIM800L.

Returns

Nothing.

2.6.3 Member Function Documentation

2.6.3.1 getMessageDate()

```
String SMSend::getMessageDate (
            String _message )
```

Get the Date when the message was sent using the whole String received from receiveSMS() and parsing it.

See also

receiveSMS()

Parameters

_message	The message received.

Returns

The Date when the message was sent.

2.6.3.2 getMessageInfo()

```
String SMSend::getMessageInfo (
            datos_mensaje _option,
            String _message )
```

Get the message information we want using the datos_mensaje enum types and parcing the String received from receiveSMS().

See also

receiveSMS()

datos_mensaje

Parameters

_option	The information we want to extract.
_message	The message received.

Returns

The message information requested.

2.6.3.3 getMessagePhone()

```
String SMSend::getMessagePhone (
            String _message )
```

Get the phone who sent a message using the whole String received from receiveSMS() and parsing it.

See also

receiveSMS()

Parameters

_message	The message received.

Returns

Phone who sent the message.

2.6.3.4 getMessageSMS()

```
String SMSend::getMessageSMS (
            String _message )
```

Get the message itself using the whole String received from receiveSMS() and parsing it.

See also

receiveSMS()

Parameters

_message	The message received.

Returns

Only the message.

2.6.3.5 isSMS()

```
bool SMSend::isSMS (
            String _message )
```

Verify if the message received is a SMS or not.

Parameters

_message	The message received.

Return values

true	The message is a valid SMS.
false	The message is not a valid SMS.

2.6.3.6 receiveSMS()

```
String SMSend::receiveSMS ( )
```

Verify if a message was received and get it.

Returns

The message with its information (without parsing).

2.6.3.7 sendSMS()

```
void SMSend::sendSMS (
            String _phoneNumber,
            String _message )
```

Send an SMS to a phone number.

Parameters

_phoneNumber	The phone number as +ZZxxxxxxxxx. Example: +541199999999
_message	The message to send.

Returns

Nothing.

The documentation for this class was generated from the following files:

- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ SMSend/src/SMSend.h
- D:/Users/nicov/Desktop/UNSAM/CIDI/Automatizacion de Laboratorios/Software/Librerias Terminadas/↩ SMSend/src/SMSend.cpp

Index