Envirion

Version 0.1

Generated by Doxygen 1.8.16

Chapter 1

Back-end

The back-end needed to store the data for further use over time that the module sent over LoRaWAN.
This is needed since the data stored on The Things Network isn't stored longer then 7 days
if an extra application is included named Storage data.

Chapter 2

Documentation

2.1 Github Repository:

The repository of the main controller
https://github.com/projectRise

The repository of the local back-end
https://github.com/projectRise/back-end

The repository of the front-end
https://github.com/projectRise/front-end

2.2 Jira:

The jira page of the project
https://projektrise.atlassian.net

2.3 Project Dokumentation:

Project speficifation.
https://docs.google.com/document/d/1qDZcl1V8Ov-OLbcYm44G_peAaIZKt5wkov6uv6↩
Ui07Q/edit?usp=sharing

Pin layout of project.
https://docs.google.com/spreadsheets/d/16jWboL5m84bEegnQy6BUv7_CsHupfp_3E↩
N4pvGDylac/edit?usp=sharing

Chapter 3

front-end

testing

Chapter 4

projectRise

The one and only

Chapter 5

Namespace Index

5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 6

Data Structure Index

6.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 7

File Index

7.1 File List

Here is a list of all files with brief descriptions:

Chapter 8

Namespace Documentation

8.1 config_device Namespace Reference

Functions

- def **receiveResponse** (sp)
- def **sendCommand** (sp, cmd, ∗args)
- def **main** ()

Variables

- string **CMD_INIT** = '$$$'
- string **CMD_OK** = 'OK'
- string **CMD_FAIL** = 'FAIL'
- string **CMD_UNKNOWN** = 'UNKNOWN'
- string **CMD_ABORT** = 'ABORT'
- string **CMD_DT** = 'DT'
- **options**
- **dest**
- **type**
- **str**
- **help**
- **metavar**
- **int**
- **required**
- **False**
- **default**
- **float**

8.1.1 Function Documentation

8.1.1.1 main()

```
def config_device.main ( )
```

8.1.1.2 receiveResponse()

```
def config_device.receiveResponse (
              sp )
```

8.1.1.3 sendCommand()

```
def config_device.sendCommand (
              sp,
              cmd,
             * args )
```

8.1.2 Variable Documentation

8.1.2.1 CMD_ABORT

```
string config_device.CMD_ABORT = 'ABORT'
```

8.1.2.2 CMD_DT

```
string config_device.CMD_DT = 'DT'
```

8.1.2.3 CMD_FAIL

```
string config_device.CMD_FAIL = 'FAIL'
```

8.1.2.4 CMD_INIT

```
string config_device.CMD_INIT = '$$$'
```

8.1.2.5 CMD_OK

```
string config_device.CMD_OK = 'OK'
```

8.1.2.6 CMD_UNKNOWN

```
string config_device.CMD_UNKNOWN = 'UNKNOWN'
```

8.1.2.7 default

```
config_device.default
```

8.1.2.8 dest

```
config_device.dest
```

8.1.2.9 False

```
config_device.False
```

8.1.2.10 float

```
config_device.float
```

8.1.2.11 help

```
config_device.help
```

8.1.2.12 int

```
config_device.int
```

8.1.2.13 metavar

```
config_device.metavar
```

8.1.2.14 options

```
config_device.options
```

Initial value:

```
1 =  argparse.ArgumentParser(
2         description="",
3         epilog=""
4     )
```

8.1.2.15 required

```
config_device.required
```

8.1.2.16 str

```
config_device.str
```

8.1.2.17 type

```
config_device.type
```

Chapter 9

Data Structure Documentation

9.1 CommandHandler Class Reference

`#include <CommandHandler.hpp>`

Public Member Functions

- bool **Receive** (void)
- template<class T >
  size_t **Send** (const T &msg)
- template<class T >
  size_t **SendLine** (const T &msg)
- void **Flush** (void)
- **CommandHandler** (Stream &source, char *buffer, size_t size, **cmdhndlr_t** callback)

9.1.1 Constructor & Destructor Documentation

9.1.1.1 CommandHandler()

```
CommandHandler::CommandHandler (
            Stream & source,
            char * buffer,
            size_t size,
             cmdhndlr_t callback )
```

9.1.2 Member Function Documentation

9.1.2.1 Flush()

```
void CommandHandler::Flush (
            void  )  [inline]
```

9.1.2.2 Receive()

```
bool CommandHandler::Receive (
            void  )
```

9.1.2.3 Send()

```
template<class T >
size_t CommandHandler::Send (
            const T & msg )  [inline]
```

9.1.2.4 SendLine()

```
template<class T >
size_t CommandHandler::SendLine (
            const T & msg )  [inline]
```

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

- inc/ **CommandHandler.hpp**
- src/ **CommandHandler.cpp**

9.2 LightTracker Class Reference

```
#include <LightTracker.hpp>
```

Public Member Functions

- int **GetHorizontalPosition** (void)
- int **GetVerticalPosition** (void)
- bool **Poll** (void)
- void **Begin** (void)
- **LightTracker** (uint8_t servoHPin, uint8_t servoVPin, uint8_t prULPin, uint8_t prURPin, uint8_t prDLPin, uint8_t prDRPin, int activeLightLevel=250, int tolerance=50, int angleHMargin=15, int angleVMargin=15)

9.2.1 Constructor & Destructor Documentation

9.2.1.1 LightTracker()

```
LightTracker::LightTracker (
            uint8_t servoHPin,
            uint8_t servoVPin,
            uint8_t prULPin,
            uint8_t prURPin,
            uint8_t prDLPin,
            uint8_t prDRPin,
            int activeLightLevel = 250,
            int tolerance = 50,
            int angleHMargin = 15,
            int angleVMargin = 15 )
```

9.2.2 Member Function Documentation

9.2.2.1 Begin()

```
void LightTracker::Begin (
            void  )
```

9.2.2.2 GetHorizontalPosition()

```
int LightTracker::GetHorizontalPosition (
            void  )
```

9.2.2.3 GetVerticalPosition()

```
int LightTracker::GetVerticalPosition (
            void  )
```

9.2.2.4 Poll()

```
bool LightTracker::Poll (
            void  )
```

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

- inc/ **LightTracker.hpp**
- src/ **LightTracker.cpp**

9.3 WeatherShield Class Reference

```
#include <WeatherShield.hpp>
```

Public Member Functions

- float **GetTemperature** (void)
- float **GetHumidity** (void)
- float **GetPressure** (void)
- float **GetLightLevel** (void) const
- float **GetBatteryLevel** (void) const
- bool **Begin** (void)
- **WeatherShield** (uint8_t vrefPin, uint8_t lightSensorPin, uint8_t batterySensorPin, float vrefVoltage=3.3f)

9.3.1 Constructor & Destructor Documentation

9.3.1.1 WeatherShield()

```
WeatherShield::WeatherShield (
            uint8_t vrefPin,
            uint8_t lightSensorPin,
            uint8_t batterySensorPin,
            float vrefVoltage = 3.3f )
```

9.3.2 Member Function Documentation

9.3.2.1 Begin()

```
bool WeatherShield::Begin (
            void  )
```

9.3.2.2 GetBatteryLevel()

```
float WeatherShield::GetBatteryLevel (
            void  ) const
```

9.3.2.3 GetHumidity()

```
float WeatherShield::GetHumidity (
            void  )
```

9.3.2.4 GetLightLevel()

```
float WeatherShield::GetLightLevel (
            void  ) const
```

9.3.2.5 GetPressure()

```
float WeatherShield::GetPressure (
            void  )
```

9.3.2.6 GetTemperature()

```
float WeatherShield::GetTemperature (
            void  )
```

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

- inc/ **WeatherShield.hpp**
- src/ **WeatherShield.cpp**

Chapter 10

File Documentation

10.1 back-end/config.h File Reference

Macros

- #define **key** "Authorization: key ttn-account-v2.x8mM_bEkA43bvHjAPvKQmjt0d9YNOpSEV1pghpWx2vU"

10.1.1 Macro Definition Documentation

10.1.1.1 key

```
#define key "Authorization:  key ttn-account-v2.x8mM_bEkA43bvHjAPvKQmjt0d9YNOpSEV1pghpWx2vU"
```

10.2 config.h File Reference

Macros

- #define **APPLY_CONFIG**
- #define **CONFIG_NSKEY** { 0x23, 0xFD, 0xCF, 0x5D, 0x83, 0xFF, 0x5D, 0x15, 0xC5, 0x6A, 0x76, 0x8E, 0xC0, 0x77, 0xE4, 0x4D }
- #define **CONFIG_ASKEY** { 0x33, 0xA0, 0x0A, 0x93, 0xAC, 0x6D, 0x81, 0xAF, 0xDE, 0xBE, 0xF5, 0x66, 0x0E, 0x90, 0xC0, 0xEF }
- #define **CONFIG_DEVADDR** 0x260115AF
- #define **CONFIG_NETID** 0x00000001
- #define **CONFIG_CHANNEL** 1

10.2.1 Macro Definition Documentation

10.2.1.1 APPLY_CONFIG

```
#define APPLY_CONFIG
```

10.2.1.2 CONFIG_ASKEY

```
#define CONFIG_ASKEY { 0x33, 0xA0, 0x0A, 0x93, 0xAC, 0x6D, 0x81, 0xAF, 0xDE, 0xBE, 0xF5, 0x66,
0x0E, 0x90, 0xC0, 0xEF }
```

10.2.1.3 CONFIG_CHANNEL

```
#define CONFIG_CHANNEL 1
```

10.2.1.4 CONFIG_DEVADDR

```
#define CONFIG_DEVADDR 0x260115AF
```

10.2.1.5 CONFIG_NETID

```
#define CONFIG_NETID 0x00000001
```

10.2.1.6 CONFIG_NSKEY

```
#define CONFIG_NSKEY { 0x23, 0xFD, 0xCF, 0x5D, 0x83, 0xFF, 0x5D, 0x15, 0xC5, 0x6A, 0x76, 0x8E,
0xC0, 0x77, 0xE4, 0x4D }
```

10.3 back-end/README.md File Reference

10.4 front-end/README.md File Reference

10.5 README.md File Reference

10.6 config_device.py File Reference

Namespaces

- **config_device**

Functions

- def **config_device.receiveResponse** (sp)
- def **config_device.sendCommand** (sp, cmd, ∗args)
- def **config_device.main** ()

Variables

- string **config_device.CMD_INIT** = '$$$'
- string **config_device.CMD_OK** = 'OK'
- string **config_device.CMD_FAIL** = 'FAIL'
- string **config_device.CMD_UNKNOWN** = 'UNKNOWN'
- string **config_device.CMD_ABORT** = 'ABORT'
- string **config_device.CMD_DT** = 'DT'
- **config_device.options**
- **config_device.dest**
- **config_device.type**
- **config_device.str**
- **config_device.help**
- **config_device.metavar**
- **config_device.int**
- **config_device.required**
- **config_device.False**
- **config_device.default**
- **config_device.float**

10.7 documentation.md File Reference

10.8 inc/CommandHandler.hpp File Reference

```
#include <string.h>
#include <stddef.h>
#include <Stream.h>
```

Data Structures

- class **CommandHandler**

Typedefs

- typedef bool(∗ **cmdhndlr_t**) (**CommandHandler** ∗, const char ∗, const char ∗)

10.8.1 Detailed Description

CommandHandler.hpp (p. 27) Class for a more flexible command-like communication between two devices via serial

Authors

albrdev (albrdev@gmail.com)

Date

2020-03-02

10.8.2 Typedef Documentation

10.8.2.1 cmdhndlr_t

```
typedef bool(* cmdhndlr_t) ( CommandHandler *, const char *, const char *)
```

10.9 inc/debug.hpp File Reference

```
#include <Arduino.h>
```

Macros

- #define **NOP** ((void)0)
- #define **DebugPrint**(...) **NOP**
- #define **DebugPrintLine**(...) **NOP**
- #define **DebugWrite**(...) **NOP**
- #define **DebugFlush**() **NOP**

10.9.1 Macro Definition Documentation

10.9.1.1 DebugFlush

```
#define DebugFlush( )  NOP
```

10.9.1.2 DebugPrint

```
#define DebugPrint(
               ... )    NOP
```

10.9.1.3 DebugPrintLine

```
#define DebugPrintLine(
               ... )    NOP
```

10.9.1.4 DebugWrite

```
#define DebugWrite(
               ... )    NOP
```

10.9.1.5 NOP

```
#define NOP ((void)0)
```

10.10 inc/LightTracker.hpp File Reference

```
#include <stdint.h>
#include <Servo.h>
```

Data Structures

- class **LightTracker**

10.10.1 Detailed Description

LightTracker.hpp (p. 29) Light source tracker using photoresistors with vertical/horizontal servomotors.

Authors

albrdev (albrdev@gmail.com)

Date

2020-02-27

10.11 inc/lora.hpp File Reference

Functions

- void **setupLoRa** (void)

10.11.1 Function Documentation

10.11.1.1 setupLoRa()

```
void setupLoRa (
            void  )
```

10.12 inc/misc.hpp File Reference

```
#include "CommandHandler.hpp"
```

Macros

- #define **CMD_INIT** "$$$"
- #define **CMD_OK** "OK"
- #define **CMD_FAIL** "FAIL"
- #define **CMD_UNKNOWN** "UNKNOWN"
- #define **CMD_ABORT** "ABORT"
- #define **CMD_SD** "SD"
- #define **CMD_DT** "DT"

Functions

- bool **handleCommand** (**CommandHandler** ∗self, const char ∗cmd, const char ∗args)

10.12.1 Macro Definition Documentation

10.12.1.1 CMD_ABORT

```
#define CMD_ABORT "ABORT"
```

10.12.1.2 CMD_DT

```
#define CMD_DT "DT"
```

10.12.1.3 CMD_FAIL

```
#define CMD_FAIL "FAIL"
```

10.12.1.4 CMD_INIT

```
#define CMD_INIT "$$$"
```

10.12.1.5 CMD_OK

```
#define CMD_OK "OK"
```

10.12.1.6 CMD_SD

```
#define CMD_SD "SD"
```

10.12.1.7 CMD_UNKNOWN

```
#define CMD_UNKNOWN "UNKNOWN"
```

10.12.2 Function Documentation

10.12.2.1 handleCommand()

```
bool handleCommand (
             CommandHandler * self,
            const char * cmd,
            const char * args )
```

10.13 inc/types.h File Reference

```
#include <stdint.h>
```

Typedefs

- typedef float **single_t**

Functions

- struct **__attribute__** ((packed))

Variables

- **header_t**
- **sensordata_t**
- **sensordata2_t**
- **collection_t**

10.13.1 Typedef Documentation

10.13.1.1 single_t

```
typedef float  single_t
```

10.13.2 Function Documentation

10.13.2.1 __attribute__()

```
struct __attribute__ (
             (packed)  )
```

10.13.3 Variable Documentation

10.13.3.1 collection_t

```
collection_t
```

10.13.3.2 header_t

```
header_t
```

10.13.3.3 sensordata2_t

```
sensordata2_t
```

10.13.3.4 sensordata_t

```
sensordata_t
```

10.14 inc/WeatherShield.hpp File Reference

```
#include <stdint.h>
#include <Wire.h>
#include "SparkFunMPL3115A2.h"
#include "SparkFun_Si7021_Breakout_Library.h"
```

Data Structures

- class **WeatherShield**

10.14.1 Detailed Description

WeatherShield.hpp (p. 33) Wrapper class for SparkFun's weather shield

Authors

albrdev (albrdev@gmail.com)

Date

2020-03-02

10.15 projectRise.cpp File Reference

```
#include <stdint.h>
#include <SPI.h>
#include <SdFat.h>
#include <RTClib.h>
#include <Sleep_n0m1.h>
#include <lmic.h>
#include <hal/hal.h>
#include "types.h"
#include "lora.hpp"
#include "WeatherShield.hpp"
#include "LightTracker.hpp"
#include "CommandHandler.hpp"
#include "misc.hpp"
#include "debug.hpp"
#include "sdios.h"
#include <Wire.h>
#include <Adafruit_BMP280.h>
```

Macros

- #define **USE_SDIO** 0
- #define **ARRCNT**(x) (sizeof((x)) / sizeof(∗(x)))
- #define **STRLEN**(x) (**ARRCNT**((x)) - 1U)
- #define **ANALOG_MAX** ((1 << 10) - 1)
- #define **SD_ENABLE**
- #define **LIGHT_TRACKER_ENABLE**
- #define **RTC_ENABLE**
- #define **BMP_ENABLE**
- #define **CONFIG_MISO_PIN** 24
- #define **CONFIG_MOSI_PIN** 23
- #define **CONFIG_SCK_PIN** 25
- #define **CONFIG_CS_PIN** 22
- #define **LOGDIR** "weather"
- #define **LOGFILE_TEXT** "data.log"
- #define **LOGFILE_BINARY** "data.dat"
- #define **TEST_READOFFSET** 0U
- #define **TEST_ELEMENTCOUNT** 2U
- #define **TEST_READCOUNT** 5U
- #define **TEST_TOTALCOUNT** ((**TEST_ELEMENTCOUNT**) ∗ (**TEST_READCOUNT**))

Functions

- bool **getSensorValues** (**collection_t** &result)
- void **saveToFile** (void)
- bool **writeTextFile** (const char ∗const filepath, const **collection_t** &content)
- bool **writeBinaryFile** (const char ∗const filepath, const **collection_t** &content)
- bool **readBinaryFile** (const char ∗const filepath, **collection_t** ∗const result, size_t ∗const resultCount, const size_t count, const size_t index=0U)
- void **printSensorValues** (const **collection_t** &content)
- void **testReadFromFile** (void)
- bool **setupSensors** (void)
- void **setup** (void)
- void **loop** (void)

Variables

- SdFatSoftSpi< **CONFIG_MISO_PIN**, **CONFIG_MOSI_PIN**, **CONFIG_SCK_PIN** > **sd**
- unsigned long **sleepDuration** = 60UL * 1000UL
- RTC_DS3231 **rtc**
- **LightTracker** **lightTracker** (44, 45, A8, A9, A10, A11, 0.25 * **ANALOG_MAX**, 0.05 * **ANALOG_MAX**)
- Adafruit_BMP280 **bmp**
- char **commandBuffer** [16+1]
- **CommandHandler** **commandHandler** (Serial, **commandBuffer**, sizeof(**commandBuffer**), **handle↩ Command**)
- unsigned long **nextUpdate** = 0UL
- unsigned int **counter** = 0U
- **collection_t** **measurementBuffer**

10.15.1 Detailed Description

projectRise.cpp (p. 34) Main code of the energy harvesting/weather station project

Authors

albrdev (albrdev@gmail.com), Ziiny (sebastian.cobert@gmail.com)

Date

2020-02-06

10.15.2 Macro Definition Documentation

10.15.2.1 ANALOG_MAX

```
#define ANALOG_MAX ((1 << 10) - 1)
```

10.15.2.2 ARRCNT

```
#define ARRCNT(
             x ) (sizeof((x)) / sizeof(*(x)))
```

10.15.2.3 BMP_ENABLE

```
#define BMP_ENABLE
```

10.15.2.4 CONFIG_CS_PIN

```
#define CONFIG_CS_PIN 22
```

10.15.2.5 CONFIG_MISO_PIN

```
#define CONFIG_MISO_PIN 24
```

10.15.2.6 CONFIG_MOSI_PIN

```
#define CONFIG_MOSI_PIN 23
```

10.15.2.7 CONFIG_SCK_PIN

```
#define CONFIG_SCK_PIN 25
```

10.15.2.8 LIGHT_TRACKER_ENABLE

```
#define LIGHT_TRACKER_ENABLE
```

10.15.2.9 LOGDIR

```
#define LOGDIR "weather"
```

10.15.2.10 LOGFILE_BINARY

```
#define LOGFILE_BINARY "data.dat"
```

10.15.2.11 LOGFILE_TEXT

```
#define LOGFILE_TEXT "data.log"
```

10.15.2.12 RTC_ENABLE

```
#define RTC_ENABLE
```

10.15.2.13 SD_ENABLE

```
#define SD_ENABLE
```

10.15.2.14 STRLEN

```
#define STRLEN(
              x ) ( ARRCNT((x)) - 1U)
```

10.15.2.15 TEST_ELEMENTCOUNT

```
#define TEST_ELEMENTCOUNT 2U
```

10.15.2.16 TEST_READCOUNT

```
#define TEST_READCOUNT 5U
```

10.15.2.17 TEST_READOFFSET

```
#define TEST_READOFFSET 0U
```

10.15.2.18 TEST_TOTALCOUNT

```
#define TEST_TOTALCOUNT (( TEST_ELEMENTCOUNT) * ( TEST_READCOUNT))
```

10.15.2.19 USE_SDIO

```
#define USE_SDIO 0
```

10.15.3 Function Documentation

10.15.3.1 getSensorValues()

```
bool getSensorValues (
              collection_t & result )
```

10.15.3.2 loop()

```
void loop (
             void  )
```

10.15.3.3 printSensorValues()

```
void printSensorValues (
             const  collection_t & content )
```

10.15.3.4 readBinaryFile()

```
bool readBinaryFile (
             const char *const filepath,
              collection_t *const result,
             size_t *const resultCount,
             const size_t count,
             const size_t index = 0U )
```

10.15.3.5 saveToFile()

```
void saveToFile (
             void  )
```

10.15.3.6 setup()

```
void setup (
             void  )
```

10.15.3.7 setupSensors()

```
bool setupSensors (
            void  )
```

10.15.3.8 testReadFromFile()

```
void testReadFromFile (
            void  )
```

10.15.3.9 writeBinaryFile()

```
bool writeBinaryFile (
            const char *const filepath,
            const  collection_t & content )
```

10.15.3.10 writeTextFile()

```
bool writeTextFile (
            const char *const filepath,
            const  collection_t & content )
```

10.15.4 Variable Documentation

10.15.4.1 bmp

```
Adafruit_BMP280 bmp
```

10.15.4.2 commandBuffer

```
char commandBuffer[16+1]
```

10.15.4.3 commandHandler

```
CommandHandler commandHandler(Serial, commandBuffer, sizeof( commandBuffer), handleCommand)
```

10.15.4.4 counter

```
unsigned int counter = 0U
```

10.15.4.5 lightTracker

```
LightTracker lightTracker(44, 45, A8, A9, A10, A11, 0.25 * ANALOG_MAX, 0.05 * ANALOG_MAX)
```

10.15.4.6 measurementBuffer

```
collection_t measurementBuffer
```

10.15.4.7 nextUpdate

```
unsigned long nextUpdate = 0UL
```

10.15.4.8 rtc

```
RTC_DS3231 rtc
```

10.15.4.9 sd

```
SdFatSoftSpi< CONFIG_MISO_PIN, CONFIG_MOSI_PIN, CONFIG_SCK_PIN> sd
```

10.15.4.10 sleepDuration

```
unsigned long sleepDuration = 60UL * 1000UL
```

10.16 src/CommandHandler.cpp File Reference

```
#include "CommandHandler.hpp"
```

10.17 src/LightTracker.cpp File Reference

```
#include "LightTracker.hpp"
#include <Arduino.h>
```

Macros

- #define **SERVO_ANGLE_MIN** 0
- #define **SERVO_ANGLE_MAX** 180

10.17.1 Macro Definition Documentation

10.17.1.1 SERVO_ANGLE_MAX

```
#define SERVO_ANGLE_MAX 180
```

10.17.1.2 SERVO_ANGLE_MIN

```
#define SERVO_ANGLE_MIN 0
```

10.18 src/lora.cpp File Reference

```
#include "lora.hpp"
#include <lmic.h>
#include <hal/hal.h>
#include "types.h"
#include "debug.hpp"
#include "config.h"
```

Macros

- #define **CONFIG_INTERVAL** 60UL

Functions

- void **os_getArtEui** (u1_t ∗buf)
- void **os_getDevEui** (u1_t ∗buf)
- void **os_getDevKey** (u1_t ∗buf)
- void **onEvent** (ev_t ev)
- void **setupLoRa** (void)

Variables

- const lmic_pinmap **lmic_pins**
- **collection_t measurementBuffer**

10.18.1 Macro Definition Documentation

10.18.1.1 CONFIG_INTERVAL

```
#define CONFIG_INTERVAL 60UL
```

10.18.2 Function Documentation

10.18.2.1 onEvent()

```
void onEvent (
            ev_t ev )
```

10.18.2.2 os_getArtEui()

```
void os_getArtEui (
            u1_t * buf )
```

10.18.2.3 os_getDevEui()

```
void os_getDevEui (
            u1_t * buf )
```

10.18.2.4 os_getDevKey()

```
void os_getDevKey (
            u1_t * buf )
```

10.18.2.5 setupLoRa()

```
void setupLoRa (
            void  )
```

10.18.3 Variable Documentation

10.18.3.1 lmic_pins

```
const lmic_pinmap lmic_pins
```

Initial value:

```
=
{
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = { 2, 6, 7 }
}
```

10.18.3.2 measurementBuffer

```
collection_t measurementBuffer
```

10.19 src/misc.cpp File Reference

```
#include "misc.hpp"
#include <errno.h>
#include <RTClib.h>
```

Functions

- bool **handleCommand** (**CommandHandler** *self, const char *cmd, const char *args)

Variables

- unsigned long **sleepDuration**
- RTC_DS3231 **rtc**

10.19.1 Function Documentation

10.19.1.1 handleCommand()

```
bool handleCommand (
             CommandHandler * self,
            const char * cmd,
            const char * args )
```

10.19.2 Variable Documentation

10.19.2.1 rtc

```
RTC_DS3231 rtc
```

10.19.2.2 sleepDuration

```
unsigned long sleepDuration
```

10.20 src/WeatherShield.cpp File Reference

```
#include "WeatherShield.hpp"
#include <Arduino.h>
```

Index