My Project

Generated by Doxygen 1.9.7

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Namespace Documentation

3.1 Network_Hub_FORDOXY Namespace Reference

Functions

- parse_verify_data (data_bytes)

  Parse and return Zigbee data for ¨Verify¨ command from an interlock device.
- parse_validate_data (data_bytes)

  Parse and return Zigbee data for ¨Validate¨ command from an interlock device.
- parse_zigbee_data (data_bytes)

  Parse and return header and body data from RS232 interface.
- parse_data (data_bytes)

  Parse and return USB log data.
- watchdog_timeout ()

  A function to handle watchdog timer timeout.
- start_watchdog_timer ()

  Starts the watchdog timer with a 60-second timeout.
- reset_watchdog_timer ()

  Resets the watchdog timer.
- post_to_url (url, data)

  Posts JSON data to the specified URL.
- main ()

  The main function for setting up the XBee device, reading data, and initiating parsing.

3.1.1 Function Documentation

3.1.1.1 main()

```
Network_Hub_FORDOXY.main ( )
```

The main function for setting up the XBee device, reading data, and initiating parsing.

3.1.1.2 parse_data()

```
Network_Hub_FORDOXY.parse_data (
                data_bytes )
```

Parse and return USB log data.

Parameters

data_bytes	Byte array of raw data.

Returns

Dictionary of parsed data.

3.1.1.3 parse_validate_data()

```
Network_Hub_FORDOXY.parse_validate_data (
              data_bytes )
```

Parse and return Zigbee data for ¨Validate¨ command from an interlock device.

Parameters

data_bytes	Byte array of raw data.

Returns

Dictionary of parsed data.

3.1.1.4 parse_verify_data()

```
Network_Hub_FORDOXY.parse_verify_data (
              data_bytes )
```

Parse and return Zigbee data for ¨Verify¨ command from an interlock device.

Parameters

data_bytes	Byte array of raw data.

Returns

Dictionary of parsed data.

3.1.1.5 parse_zigbee_data()

```
Network_Hub_FORDOXY.parse_zigbee_data (
              data_bytes )
```

Parse and return header and body data from RS232 interface.

Parameters

data_bytes	Byte array of raw data.

Returns

Dictionary of parsed data.

3.1.1.6 post_to_url()

```
Network_Hub_FORDOXY.post_to_url (
              url,
              data )
```

Posts JSON data to the specified URL.

Parameters

url	The URL to post data to.
data	The data to be posted.

3.1.1.7 reset_watchdog_timer()

```
Network_Hub_FORDOXY.reset_watchdog_timer ( )
```

Resets the watchdog timer.

3.1.1.8 start_watchdog_timer()

```
Network_Hub_FORDOXY.start_watchdog_timer ( )
```

Starts the watchdog timer with a 60-second timeout.

3.1.1.9 watchdog_timeout()

```
Network_Hub_FORDOXY.watchdog_timeout ( )
```

A function to handle watchdog timer timeout.

Resets state and logs timeout event.

3.2 py_xbee Namespace Reference

This module provides functions to parse and send data from XBee devices and RS232 interfaces.

3.2.1 Detailed Description

This module provides functions to parse and send data from XBee devices and RS232 interfaces.

The main functionalities include data parsing from Zigbee, USB log, and RS232, as well as sending the parsed data to a specified URL. It also implements a watchdog timer for RS232 data reception.

3.3 SQM_emulator_demo_rev3_FORDOXY Namespace Reference

Functions

- wait_for_serial_message (ser)
- send_serial_response (ser, response_message)

Variables

- str serial_port = ¨COM6¨

  Set the serial port and baud rate.
- int baud_rate = 9600

  Open the serial port.
- timeout

  Open the serial port.
- int rate_count = 0
- int thickness_count = 0
- bool timer_started = False

  Start a timer if it hasn't been started.
- rate = round(random.uniform(2.0, 10.0), 2)

  Generate random values for rate and thickness.
- thickness = round(random.uniform(2.0, 10.0), 2)
- received_message = wait_for_serial_message(ser)

  Wait for a message and process the received message.
- str response_message = ¨!%A1(118)(135)¨

  Emulate and respond to the U message.
- start_time = time.time()
- end_time = time.time()

  Record the end time and display elapsed time.
- elapsed_time = end_time - start_time

3.3.1 Function Documentation

3.3.1.1 send_serial_response()

```
SQM_emulator_demo_rev3_FORDOXY.send_serial_response (
              ser,
              response_message )
```

```
@brief Send a response message over the serial port.

@param ser: The serial port object.
@type ser: Serial
@param response_message: The response message to send.
@type response_message: str
```

3.3.1.2 wait_for_serial_message()

```
SQM_emulator_demo_rev3_FORDOXY.wait_for_serial_message (
              ser )
```

```
@brief Wait for a message from the serial port and return it.

@param ser: The serial port object.
@type ser: Serial
@return: The received message.
@rtype: str
```

3.3.2 Variable Documentation

3.3.2.1 baud_rate

```
SQM_emulator_demo_rev3_FORDOXY.baud_rate = 9600
```

Open the serial port.

3.3.2.2 elapsed_time

```
SQM_emulator_demo_rev3_FORDOXY.elapsed_time = end_time - start_time
```

3.3.2.3 end_time

```
SQM_emulator_demo_rev3_FORDOXY.end_time = time.time()
```

Record the end time and display elapsed time.

3.3.2.4 rate

```
SQM_emulator_demo_rev3_FORDOXY.rate = round(random.uniform(2.0, 10.0), 2)
```

Generate random values for rate and thickness.

3.3.2.5 rate_count

```
int SQM_emulator_demo_rev3_FORDOXY.rate_count = 0
```

3.3.2.6 received_message

```
SQM_emulator_demo_rev3_FORDOXY.received_message = wait_for_serial_message(ser)
```

Wait for a message and process the received message.

3.3.2.7 response_message

```
str SQM_emulator_demo_rev3_FORDOXY.response_message = ¨!%A1(118)(135)¨
```

Emulate and respond to the U message.

Emulate and respond to the A message.

3.3.2.8 serial_port

```
SQM_emulator_demo_rev3_FORDOXY.serial_port = ¨COM6¨
```

Set the serial port and baud rate.

Open the serial port.

3.3.2.9 start_time

```
SQM_emulator_demo_rev3_FORDOXY.start_time = time.time()
```

3.3.2.10 thickness

```
SQM_emulator_demo_rev3_FORDOXY.thickness = round(random.uniform(2.0, 10.0), 2)
```

3.3.2.11 thickness_count

```
int SQM_emulator_demo_rev3_FORDOXY.thickness_count = 0
```

3.3.2.12 timeout

```
SQM_emulator_demo_rev3_FORDOXY.timeout
```

Open the serial port.

3.3.2.13 timer_started

```
bool SQM_emulator_demo_rev3_FORDOXY.timer_started = False
```

Start a timer if it hasn't been started.

Chapter 4

File Documentation

4.1 Interlock_Code_FORDOXY.c File Reference

```
#include <Adafruit_RA8875.h>
#include <SPI.h>
#include ¨Adafruit_GFX.h¨
#include <Keypad.h>
#include <MFRC522.h>
#include <stdio.h>
#include <stdint.h>
#include <SoftwareSerial.h>
#include <Arduino.h>
```

Macros

- #define RA8875_CS 10
- #define RA8875_RESET 9
- #define RA8875_INT 3
- #define RST_PIN 7
- #define SS_PIN 8

Functions

- MFRC522 mfrc522 (SS_PIN, RST_PIN)
- void setup ()
- void loop ()
- void XbeeSend (const byte ∗Data, const size_t DataLength)
- void XbeeCredentials (const uint32_t userID, const char ∗pin, const byte time)
- void XbeeLogout ()
- void XbeeRFID (const byte ∗RFID, const byte time)
- byte XbeeReceive ()
- void XbeeExtend (const byte time)
- void decomposeInt32 (uint32_t input_int, byte bytes[4])
- void decomposeInt64 (uint64_t input_int, byte bytes[8])
- byte calculateChecksum (const byte ∗Frame, size_t FrameLength)
- void drawLoggedInScreen (String u, String x1)
- void drawUserIDScreen ()
- void drawUserIDScreenUpdating (String x)
- void drawPinScreen (String y)
- void drawDurationScreen (String z)
- byte TimeToByte (const String &str)
- String removeWhitespace (String str)

Variables

- int timeoutResponse = 100
- String userID = ""
- String pin = ""
- String loginDuration = ""
- String RFIDtrans = ""
- int state = 1
- String key = ""
- String key1 = ""
- const byte ROWS = 4
- const byte COLS = 3
- char hexaKeys [ROWS][COLS]
- byte rowPins [ROWS] = {31, 36, 35, 33}
- byte colPins [COLS] = {32, 30, 34}
- Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS)
- Adafruit_RA8875 tft = Adafruit_RA8875(RA8875_CS, RA8875_RESET)

4.1.1 Macro Definition Documentation

4.1.1.1 RA8875_CS

```
#define RA8875_CS 10
```

4.1.1.2 RA8875_INT

```
#define RA8875_INT 3
```

4.1.1.3 RA8875_RESET

```
#define RA8875_RESET 9
```

4.1.1.4 RST_PIN

```
#define RST_PIN 7
```

4.1.1.5 SS_PIN

```
#define SS_PIN 8
```

4.1.2 Function Documentation

4.1.2.1 calculateChecksum()

```
byte calculateChecksum (
             const byte * Frame,
             size_t FrameLength )
```

4.1.2.2 decomposeInt32()

```
void decomposeInt32 (
            uint32_t input_int,
            byte bytes[4] )
```

4.1.2.3 decomposeInt64()

```
void decomposeInt64 (
            uint64_t input_int,
            byte bytes[8] )
```

4.1.2.4 drawDurationScreen()

```
void drawDurationScreen (
            String z )
```

4.1.2.5 drawLoggedInScreen()

```
void drawLoggedInScreen (
            String u,
            String x1 )
```

4.1.2.6 drawPinScreen()

```
void drawPinScreen (
            String y )
```

4.1.2.7 drawUserIDScreen()

```
void drawUserIDScreen ( )
```

4.1.2.8 drawUserIDScreenUpdating()

```
void drawUserIDScreenUpdating (
            String x )
```

4.1.2.9 loop()

```
void loop ( )
```

4.1.2.10 mfrc522()

```
MFRC522 mfrc522 (
             SS_PIN ,
             RST_PIN  )
```

4.1.2.11 removeWhitespace()

```
String removeWhitespace (
             String str )
```

4.1.2.12 setup()

```
void setup ( )
```

4.1.2.13 TimeToByte()

```
byte TimeToByte (
             const String & str )
```

4.1.2.14 XbeeCredentials()

```
void XbeeCredentials (
             const uint32_t userID,
             const char * pin,
             const byte time )
```

4.1.2.15 XbeeExtend()

```
void XbeeExtend (
             const byte time )
```

4.1.2.16 XbeeLogout()

```
void XbeeLogout ( )
```

4.1.2.17 XbeeReceive()

```
byte XbeeReceive ( )
```

4.1.2.18 XbeeRFID()

```
void XbeeRFID (
             const byte * RFID,
             const byte time )
```

4.1.2.19 XbeeSend()

```
void XbeeSend (
             const byte * Data,
             const size_t DataLength )
```

< Digi Xbee Start Delimeter

4.1.3 Variable Documentation

4.1.3.1 colPins

```
byte colPins[COLS] = {32, 30, 34}
```

4.1.3.2 COLS

```
const byte COLS = 3
```

4.1.3.3 customKeypad

```
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS)
```

4.1.3.4 hexaKeys

```
char hexaKeys[ROWS][COLS]
```

Initial value:

```
= {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
}
```

4.1.3.5 key

```
String key = ""
```

4.1.3.6 key1

```
String key1 = ""
```

4.1.3.7 loginDuration

```
String loginDuration = ""
```

4.1.3.8 pin

```
String pin = ""
```

4.1.3.9 RFIDtrans

```
String RFIDtrans = ""
```

4.1.3.10 rowPins

```
byte rowPins[ROWS] = {31, 36, 35, 33}
```

4.1.3.11 ROWS

```
const byte ROWS = 4
```

4.1.3.12 state

```
int state = 1
```

4.1.3.13 tft

```
Adafruit_RA8875 tft = Adafruit_RA8875(RA8875_CS, RA8875_RESET)
```

4.1.3.14 timeoutResponse

```
int timeoutResponse = 100
```

4.1.3.15 userID

```
String userID = ""
```

4.2 Network_Hub_FORDOXY.py File Reference

Namespaces

- namespace Network_Hub_FORDOXY
- namespace py_xbee

    This module provides functions to parse and send data from XBee devices and RS232 interfaces.

Functions

- Network_Hub_FORDOXY.parse_verify_data (data_bytes)

  Parse and return Zigbee data for ¨Verify¨ command from an interlock device.
- Network_Hub_FORDOXY.parse_validate_data (data_bytes)

  Parse and return Zigbee data for ¨Validate¨ command from an interlock device.
- Network_Hub_FORDOXY.parse_zigbee_data (data_bytes)

  Parse and return header and body data from RS232 interface.
- Network_Hub_FORDOXY.parse_data (data_bytes)

  Parse and return USB log data.
- Network_Hub_FORDOXY.watchdog_timeout ()

  A function to handle watchdog timer timeout.
- Network_Hub_FORDOXY.start_watchdog_timer ()

  Starts the watchdog timer with a 60-second timeout.
- Network_Hub_FORDOXY.reset_watchdog_timer ()

  Resets the watchdog timer.
- Network_Hub_FORDOXY.post_to_url (url, data)

  Posts JSON data to the specified URL.
- Network_Hub_FORDOXY.main ()

  The main function for setting up the XBee device, reading data, and initiating parsing.

4.3 RS232_Luke_Final_rev1_FORDOXY.c File Reference

```
#include <stdio.h>
#include <SoftwareSerial.h>
#include <Arduino.h>
#include <SD.h>
#include <SPI.h>
```

Functions

- void setup ()

  Function to set up the baud rate for Serial port (RS232) and for the Serial1 port (Zigbee) Sets up a timeout for inactive responses Initialises the SD card for writing.
- void loop ()

  Main loop function.
- String readSerialResponse ()

  Function to read a string from the serial buffer.
- void extractResponseA (const String &receivedPacket, float &density, float &tooling, float &zRatio)

  Function to handle the extraction of the density, tooling, and zratio values from the received string from SQM The received string structure is: !0AFILM1____1.01_120__1.213_33.380__0.211_0_1(79)(59), where the first three values are the desired values The function returns them as floats.
- char extractResponseU (const String &receivedPacket)

  Function to handle the extraction of the response from command U The received string structure is: !A1(118)(135) where the value of U is the 4th character (either 1 or 0)
- float extractResponseL (const String &receivedPacket)

  Function to handle the extraction of the rate from the returned string from command L The received string structure is: !∗A_8.20_(91)(100) where the rate value is the float after the first underscore.
- float extractResponseN (const String &receivedPacket)

Function to handle the extraction of the thickness from the returned string from command N The received string structure is: !+A_2.00_(74)(111) where the thickness value is the float after the first underscore.

- void writeToSDCardFloat (float value)

  Function to write a float value to the SD card.
- void deleteAllTextEntries ()

  Function that deletes every entry of the text file after all the data has been sent to zigbee for that process.
- void sendToZigbee ()

  Function to send data to Zigbee.
- byte calculateChecksum (const byte ∗Frame, size_t FrameLength)

  Function to calculate the checksum of a frame.

Variables

- const int baudRate = 9600

  RS232 baud rate of 9600, this can be updated to 19200 for use with SQM-160.
- const int pollInterval = 1000

  Log interval in milliseconds.
- const int timeoutResponse = 100

  Wait time for serial response.
- const String messageA = ¨!%A1?(46)(149)¨

  A message.
- const String messageL = ¨!%L1?(133)(123)¨

  L message.
- const String messageN = ¨!$N1(93)(81)¨

  N message.
- const String messageU = ¨!$U?(91)(84)¨

  U message.
- char responseU
- char UTracker = '0'

  Initialise responseU to 0 and a flag for the U tracker.
- bool logActive = false

  Flag for if a log is active.
- unsigned long lastPollTime = 0

  Initialise the last poll time to 0.

4.3.1 Function Documentation

4.3.1.1 calculateChecksum()

```
byte calculateChecksum (
            const byte * Frame,
            size_t FrameLength )
```

Function to calculate the checksum of a frame.

Parameters

Frame	Pointer to the frame
FrameLength	Length of the frame

Returns

The calculated checksum

4.3.1.2 deleteAllTextEntries()

```
void deleteAllTextEntries ( )
```

Function that deletes every entry of the text file after all the data has been sent to zigbee for that process.

4.3.1.3 extractResponseA()

```
void extractResponseA (
             const String & receivedPacket,
             float & density,
             float & tooling,
             float & zRatio )
```

Function to handle the extraction of the density, tooling, and zratio values from the received string from SQM The received string structure is: !0AFILM1____1.01_120__1.213_33.380__0.211_0_1(79)(59), where the first three values are the desired values The function returns them as floats.

Parameters

receivedPacket	The received string
density	Output parameter for density
tooling	Output parameter for tooling
zRatio	Output parameter for zRatio

4.3.1.4 extractResponseL()

```
float extractResponseL (
             const String & receivedPacket )
```

Function to handle the extraction of the rate from the returned string from command L The received string structure is: !∗A_8.20_(91)(100) where the rate value is the float after the first underscore.

Parameters

receivedPacket	The received string

Returns

The extracted rate value

4.3.1.5 extractResponseN()

```
float extractResponseN (
             const String & receivedPacket )
```

Function to handle the extraction of the thickness from the returned string from command N The received string structure is: !+A_2.00_(74)(111) where the thickness value is the float after the first underscore.

Parameters

receivedPacket	The received string

Returns

The extracted thickness value

4.3.1.6 extractResponseU()

```
char extractResponseU (
             const String & receivedPacket )
```

Function to handle the extraction of the response from command U The received string structure is: !A1(118)(135) where the value of U is the 4th character (either 1 or 0)

Parameters

receivedPacket	The received string

Returns

The extracted response character

4.3.1.7 loop()

```
void loop ( )
```

Main loop function.

4.3.1.8 readSerialResponse()

```
String readSerialResponse ( )
```

Function to read a string from the serial buffer.

Returns

The read string

4.3.1.9 sendToZigbee()

```
void sendToZigbee ( )
```

Function to send data to Zigbee.

4.3.1.10 setup()

```
void setup ( )
```

Function to set up the baud rate for Serial port (RS232) and for the Serial1 port (Zigbee) Sets up a timeout for inactive responses Initialises the SD card for writing.

4.3.1.11 writeToSDCardFloat()

```
void writeToSDCardFloat (
             float value )
```

Function to write a float value to the SD card.

Parameters

value	The float value to write

4.3.2 Variable Documentation

4.3.2.1 baudRate

```
const int baudRate = 9600
```

RS232 baud rate of 9600, this can be updated to 19200 for use with SQM-160.

4.3.2.2 lastPollTime

```
unsigned long lastPollTime = 0
```

Initialise the last poll time to 0.

4.3.2.3 logActive

```
bool logActive = false
```

Flag for if a log is active.

4.3.2.4 messageA

```
const String messageA = ¨!%A1?(46)(149)¨
```

A message.

4.3.2.5 messageL

```
const String messageL = ¨!%L1?(133)(123)¨
```

L message.

4.3.2.6 messageN

```
const String messageN = ¨!$N1(93)(81)¨
```

N message.

4.3.2.7 messageU

```
const String messageU = ¨!$U?(91)(84)¨
```

U message.

4.3.2.8 pollInterval

```
const int pollInterval = 1000
```

Log interval in milliseconds.

4.3.2.9 responseU

```
char responseU
```

4.3.2.10 timeoutResponse

```
const int timeoutResponse = 100
```

Wait time for serial response.

4.3.2.11 UTracker

```
char UTracker = '0'
```

Initialise responseU to 0 and a flag for the U tracker.

4.4 SQM_emulator_demo_rev3_FORDOXY.py File Reference

Namespaces

- namespace SQM_emulator_demo_rev3_FORDOXY

Functions

- SQM_emulator_demo_rev3_FORDOXY.wait_for_serial_message (ser)
- SQM_emulator_demo_rev3_FORDOXY.send_serial_response (ser, response_message)

Variables

- str SQM_emulator_demo_rev3_FORDOXY.serial_port = ¨COM6¨

  Set the serial port and baud rate.
- int SQM_emulator_demo_rev3_FORDOXY.baud_rate = 9600

  Open the serial port.
- SQM_emulator_demo_rev3_FORDOXY.timeout

  Open the serial port.
- int SQM_emulator_demo_rev3_FORDOXY.rate_count = 0
- int SQM_emulator_demo_rev3_FORDOXY.thickness_count = 0
- bool SQM_emulator_demo_rev3_FORDOXY.timer_started = False

  Start a timer if it hasn't been started.
- SQM_emulator_demo_rev3_FORDOXY.rate = round(random.uniform(2.0, 10.0), 2)

  Generate random values for rate and thickness.
- SQM_emulator_demo_rev3_FORDOXY.thickness = round(random.uniform(2.0, 10.0), 2)
- SQM_emulator_demo_rev3_FORDOXY.received_message = wait_for_serial_message(ser)

  Wait for a message and process the received message.
- str SQM_emulator_demo_rev3_FORDOXY.response_message = ¨!%A1(118)(135)¨

  Emulate and respond to the U message.
- SQM_emulator_demo_rev3_FORDOXY.start_time = time.time()
- SQM_emulator_demo_rev3_FORDOXY.end_time = time.time()

  Record the end time and display elapsed time.
- SQM_emulator_demo_rev3_FORDOXY.elapsed_time = end_time - start_time

4.5 USB_PowerShell_Analysis_FORDOXY.ps1 File Reference

Index