# STEVAL UART COMMUNICATION

Generated by Doxygen 1.8.13

# Contents

# Chapter 1

# Class Index

## 1.1 Class List

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

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 3

# Class Documentation

## 3.1 Frame Struct Reference

Uart frame obejct.

`#include <uartSteval.h>`

### Public Member Functions

- Frame ()

  *Default construct a new Frame object.*
- Frame (int motorid, FRAME_CODES framecode, STEVAL_REGISTERS registerId, STEVAL_REGISTER↩ S_LEN payload_len, int data)

  *Construct a new Frame object.*
- int getCommad ()

  *Get the Commad object as int.*
- int reverse (int b, int l)

  *Bitwise inversion of a number.*

### Public Attributes

- int **motorId**
- FRAME_CODES **frameCode**
- STEVAL_REGISTERS **reg**
- STEVAL_REGISTERS_LEN **regLen**
- int **data**
- int **CRC**

### 3.1.1 Detailed Description

Uart frame obejct.

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 Frame()

```
Frame::Frame (
            int motorid,
            FRAME_CODES framecode,
            STEVAL_REGISTERS registerId,
            STEVAL_REGISTERS_LEN payload_len,
            int data )  [inline]
```

Construct a new Frame object.

**Parameters**

| | |
|---|---|
| *motorid* | Motor id |
| *framecode* | Frame code from FRAME_CODES enum |
| *registerId* | Motor control register id |
| *payload_len* | Motor control register payload length |
| *data* | Data to send, if no data put NO_DATA |

### 3.1.3 Member Function Documentation

#### 3.1.3.1 getCommad()

```
int Frame::getCommad ( )  [inline]
```

Get the Commad object as int.

**Returns**

int Command as single int

#### 3.1.3.2 reverse()

```
int Frame::reverse (
            int b,
            int l )  [inline]
```

Bitwise inversion of a number.

**Parameters**

| | |
|---|---|
| *b* | int number to reverse |
| *l* | Bitwise lenght of number |

**Returns**

int inverst number

The documentation for this struct was generated from the following file:

- uartSteval.h

## 3.2 UART Struct Reference

Uart connection object.

`#include <uartSteval.h>`

### Public Attributes

- int **baud**
- char * **device**

### 3.2.1 Detailed Description

Uart connection object.

The documentation for this struct was generated from the following file:

- uartSteval.h

# Chapter 4

# File Documentation

## 4.1 registers.h File Reference

This graph shows which files directly or indirectly include this file:

## 4.2 uartSteval.cpp File Reference

```
#include <ctype.h>
#include <stdio.h>
#include <wiringPi.h>
#include <iostream>
#include <unistd.h>
#include <wiringSerial.h>
#include "uartSteval.h"
```

Include dependency graph for uartSteval.cpp:

### Functions

- UART_STATUS receive (Frame *cmd, int connection)

  *Receive uart frame.*
- UART_STATUS **sendData** (int con, int data, int l, bool cmd_show=false)
- UART_STATUS send (Frame cmd, UART uart, bool cmd_show=false)

  *Send data in frame.*
- UART_STATUS StartMotor (int motorId, UART uart)

  *Execute start motor command.*
- UART_STATUS StopMotor (int motorId, UART uart)

  *Execute stop motor command.*
- UART_STATUS SetMotorRefSpeed (int ref, int motorId, UART uart)

  *Set the motor ramp final speed registry.*
- UART_STATUS SetRegistry (STEVAL_REGISTERS reg, STEVAL_REGISTERS_LEN regL, int motorId, UART uart, int data)

  *Set the registry value.*
- UART_STATUS GetRegistry (STEVAL_REGISTERS reg, STEVAL_REGISTERS_LEN regL, int motorId, UART uart)

  *Get the registry value.*

### 4.2.1 Detailed Description

Created on: Nov 27, 2020 Author: Jakub Walkowski

### 4.2.2 Function Documentation

#### 4.2.2.1 GetRegistry()

```
UART_STATUS GetRegistry (
            STEVAL_REGISTERS reg,
            STEVAL_REGISTERS_LEN regL,
            int motorId,
            UART uart )
```

Get the registry value.

**Parameters**

| | |
|---|---|
| *reg* | Motor control register id |
| *regL* | Montrol register payload length |
| *motorId* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Getting resoult

#### 4.2.2.2 receive()

```
UART_STATUS receive (
            Frame * cmd,
            int connection )
```

Receive uart frame.

**Parameters**

| | |
|---|---|
| *cmd* | Received frame |
| *connection* | Uart connection id |

**Returns**

UART_STATUS Receiving status

#### 4.2.2.3 send()

```
UART_STATUS send (
            Frame cmd,
            UART uart,
            bool cmd_show = false )
```

Send data in frame.

**Parameters**

| | |
|---|---|
| *con* | Uart conneciton id |
| *data* | Data to send |
| *l* | Control register payload length |
| *cmd_show* | Optional param, if true show sending fame in console |

**Returns**

UART_STATUS Sending resoult

#### 4.2.2.4 SetMotorRefSpeed()

```
UART_STATUS SetMotorRefSpeed (
            int ref,
            int motorId,
            UART uart )
```

Set the motor ramp final speed registry.

**Parameters**

| | |
|---|---|
| *ref* | Value to set |
| *motor↩ Id* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Setting resoult

#### 4.2.2.5 SetRegistry()

```
UART_STATUS SetRegistry (
            STEVAL_REGISTERS reg,
```

```
            STEVAL_REGISTERS_LEN regL,
            int motorId,
            UART uart,
            int data )
```

Set the registry value.

**Parameters**

| | |
|---|---|
| *reg* | Motor control register id |
| *regL* | Montrol register payload length |
| *motor↩ Id* | Motor id |
| *uart* | Uart object |
| *data* | Value to set |

**Returns**

UART_STATUS Setting resoult

#### 4.2.2.6 StartMotor()

```
UART_STATUS StartMotor (
            int motorId,
            UART uart )
```

Execute start motor command.

**Parameters**

| | |
|---|---|
| *motor↩ Id* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Starting motor resoult

#### 4.2.2.7 StopMotor()

```
UART_STATUS StopMotor (
            int motorId,
            UART uart )
```

Execute stop motor command.

**Parameters**

| | |
|---|---|
| *motor↩ Id* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Stoping motor resoult

## 4.3 uartSteval.h File Reference

```
#include <climits>
#include "registers.h"
```

Include dependency graph for uartSteval.h: This graph shows which files directly or indirectly include this file:

### Classes

- struct UART

  *Uart connection object.*
- struct Frame

  *Uart frame obejct.*

### Macros

- #define **NO_DATA** INT_MAX

### Functions

- UART_STATUS receive (Frame *cmd, int connection)

  *Receive uart frame.*
- UART_STATUS send (Frame cmd, UART uart, bool cmd_show=false)

  *Send data in frame.*
- UART_STATUS StartMotor (int motorId, UART uart)

  *Execute start motor command.*
- UART_STATUS StopMotor (int motorId, UART uart)

  *Execute stop motor command.*
- UART_STATUS SetMotorRefSpeed (int ref, int motorId, UART uart)

  *Set the motor ramp final speed registry.*
- UART_STATUS SetRegistry (STEVAL_REGISTERS reg, STEVAL_REGISTERS_LEN regL, int motorId, UART uart, int data)

  *Set the registry value.*
- UART_STATUS GetRegistry (STEVAL_REGISTERS reg, STEVAL_REGISTERS_LEN regL, int motorId, UART uart)

  *Get the registry value.*

### 4.3.1 Detailed Description

Created on: Nov 27, 2020 Author: walko

### 4.3.2 Function Documentation

#### 4.3.2.1 GetRegistry()

```
UART_STATUS GetRegistry (
            STEVAL_REGISTERS reg,
            STEVAL_REGISTERS_LEN regL,
            int motorId,
            UART uart )
```

Get the registry value.

**Parameters**

| | |
|---|---|
| *reg* | Motor control register id |
| *regL* | Montrol register payload length |
| *motorId* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Getting resoult

#### 4.3.2.2 receive()

```
UART_STATUS receive (
            Frame * cmd,
            int connection )
```

Receive uart frame.

**Parameters**

| | |
|---|---|
| *cmd* | Received frame |
| *connection* | Uart connection id |

**Returns**

UART_STATUS Receiving status

#### 4.3.2.3 send()

```
UART_STATUS send (
            Frame cmd,
            UART uart,
            bool cmd_show = false )
```

Send data in frame.

**Parameters**

| | |
|---|---|
| *con* | Uart conneciton id |
| *data* | Data to send |
| *l* | Control register payload length |
| *cmd_show* | Optional param, if true show sending fame in console |

**Returns**

UART_STATUS Sending resoult

#### 4.3.2.4 SetMotorRefSpeed()

```
UART_STATUS SetMotorRefSpeed (
            int ref,
            int motorId,
            UART uart )
```

Set the motor ramp final speed registry.

**Parameters**

| | |
|---|---|
| *ref* | Value to set |
| *motor↩ Id* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Setting resoult

#### 4.3.2.5 SetRegistry()

```
UART_STATUS SetRegistry (
            STEVAL_REGISTERS reg,
```

```
            STEVAL_REGISTERS_LEN regL,
            int motorId,
            UART uart,
            int data )
```

Set the registry value.

**Parameters**

| | |
|---|---|
| *reg* | Motor control register id |
| *regL* | Montrol register payload length |
| *motor↩ Id* | Motor id |
| *uart* | Uart object |
| *data* | Value to set |

**Returns**

UART_STATUS Setting resoult

#### 4.3.2.6 StartMotor()

```
UART_STATUS StartMotor (
            int motorId,
            UART uart )
```

Execute start motor command.

**Parameters**

| | |
|---|---|
| *motor↩ Id* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Starting motor resoult

#### 4.3.2.7 StopMotor()

```
UART_STATUS StopMotor (
            int motorId,
            UART uart )
```

Execute stop motor command.

**Parameters**

| | |
|---|---|
| *motor↩ Id* | Motor id |
| *uart* | Uart object |

**Returns**

UART_STATUS Stoping motor resoult

# Index