GroundStation_nortos.c File Reference

Source file that allows communication between ground station and CubeSat. More...

```
#include <GroundStation_nortos.h>
#include <stdio.h>
#include <stdlib.h>
#include <ti/drivers/pin/PINCC26XX.h>
#include <ti/drivers/Power.h>
#include <ti/drivers/rf/RF.h>
#include <ti/drivers/timer/GPTimerCC26XX.h>
#include <ti/display/Display.h>
#include <ti/devices/DeviceFamily.h>
#include <ti/drivers/UART.h>
#include <unistd.h>
#include "Board.h"
#include "smartrf_settings/smartrf_settings.h"
#include "easylink/EasyLink.h"
```

Functions

void	**displaySetup** () Setup display driver. More...
void	**uartDisplaySetup** () Setup UART display driver. More...
void	**timerSetup** () Setup GP timer. More...
void	**timerStart** () Start the timer. More...
void	**timerEnd** () Stop the timer and close the driver This also displays the timer value in milliseconds. More...
bool	**inputErrors** (char userEntry, **Entry_Type** entryType) Handle errors in user entry. More...
void	**uartWriteSimple** (UART_Handle uartHandle, char *string, int stringSize) Perform UART write with error checking. More...
void	**uartReadSimple** (int status) Perform UART read with error checking. More...
char	**userInput** (char *message, int messageSize) Display start-up prompts to user. More...
void	**userEntryCompile** () Process the data entered by the user. More...
void	**rfPacketSetup** (uint8_t commandNumber, uint8_t sleepTime, uint8_t femtoAddress) Setup packet to be sent to CubeSat. More...

bool	**isPacketCorrect** (EasyLink_RxPacket *rxp, EasyLink_TxPacket *txp) Check ack packet from CubeSat is correct. More...
void	**femtosatStatusDisplay** (uint8_t femtoRssi, uint8_t statusByte) Display the status of the femtosat to the user via UART display. More...
void	**cubeSatTx** () Transmit command(s) to CubeSat. More...
bool	**cubeSatAckRx** () Switch to receive mode for ack from CubeSat. More...
void	**dataRx** (bool ackFlag) Switch to receive mode for data from CubeSat. More...
void *	**mainThread** (void *arg0)

Variables

EasyLink_Params	**easyLinkParams**
uint32_t	**absTime**
EasyLink_TxPacket	**txPacket** = {{0}, 0, 0, {0}}
EasyLink_RxPacket	**rxPacket** = {{0}, 0, 0, 0, 0, {0}}
EasyLink_Status	**result**
uint8_t	**femtoAddr**
bool	**receptionFlag**
static PIN_Handle	**pinHandle**
static PIN_State	**pinState**
static Display_Handle	**display**
static GPTimerCC26XX_Handle	**timerHandle**
static UART_Handle	**uartHandle**
static UART_Params	**uartParams**
static GPTimerCC26XX_Params	**timerParams**
static GPTimerCC26XX_Value	**timerValue**
PIN_Config	**pinTable** []

Detailed Description

Source file that allows communication between ground station and CubeSat.

==

Author

Aymen Benylles

Date

30/12/2020

Function Documentation

◆ cubeSatAckRx()

```
bool cubeSatAckRx ( )
```

Switch to receive mode for ack from CubeSat.

Returns

ackFlag

◆ cubeSatTx()

```
void cubeSatTx ( )
```

Transmit command(s) to CubeSat.

Returns

none

◆ dataRx()

```
void dataRx ( bool  ackFlag )
```

Switch to receive mode for data from CubeSat.

Returns

none

◆ displaySetup()

```
void displaySetup ( )
```

Setup display driver.

Returns

none

◆ femtosatStatusDisplay()

```
void femtosatStatusDisplay ( uint8_t  femtoRssi,
                             uint8_t  statusByte
                           )
```

Display the status of the femtosat to the user via UART display.

Parameters

femtoRssi	RSSI value for CubeSay-femtosat link
statusByte	Byte containing status of femtosat reception
femtoAddr	Address of femtosat

Returns

none

◆ inputErrors()

```
bool inputErrors ( char        userEntry,
                   Entry_Type  entryType
                 )
```

Handle errors in user entry.

Parameters

userEntry Character entered by user

entryType Type of data entered by user

Returns

userErrorFlag Flag raised when user makes an invalid input

◆ isPacketCorrect()

```
bool isPacketCorrect ( EasyLink_RxPacket *  rxp,
                       EasyLink_TxPacket *  txp
                     )
```

Check ack packet from CubeSat is correct.

Parameters

***rxp** received packet

***txp** transmitted packet

Returns

status

◆ rfPacketSetup()

```
void rfPacketSetup ( uint8_t  commandNumber,
                     uint8_t  sleepTime,
                     uint8_t  femtoAddress
                   )
```

Setup packet to be sent to CubeSat.

Returns

none

◆ timerEnd()

void timerEnd ( )

Stop the timer and close the driver This also displays the timer value in milliseconds.

Returns

none

◆ timerSetup()

void timerSetup ( )

Setup GP timer.

Returns

none

◆ timerStart()

void timerStart ( )

Start the timer.

Returns

none

◆ uartDisplaySetup()

void uartDisplaySetup ( )

Setup UART display driver.

Returns

none

◆ uartReadSimple()

```
void uartReadSimple ( int  status )
```

Perform UART read with error checking.

Parameters

> **status** Status of read operation

Returns

> none

uartWriteSimple()

```
void uartWriteSimple ( UART_Handle  uartHandle,
                       char *       string,
                       int          stringSize
                     )
```

Perform UART write with error checking.

Returns

> none

userEntryCompile()

```
void userEntryCompile (  )
```

Process the data entered by the user.

Returns

> none

userInput()

```
char userInput ( char *  message,
                 int     messageSize
               )
```

Display start-up prompts to user.

Parameters

***message**	Prompt to be displayed to user
messageSize	Size of the prompt

Returns

userEntry Character entered by user

Variable Documentation

◆ absTime

uint32_t absTime

Variable the RF core uses to time commands

◆ femtoAddr

uint8_t femtoAddr

Address of femtosatellite message is being sent to

◆ pinTable

PIN_Config pinTable[]

Initial value:

```
= {
    Board_PIN_LED1 | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,
    Board_PIN_LED2 | PIN_GPIO_OUTPUT_EN | PIN_GPIO_LOW | PIN_PUSHPULL | PIN_DRVSTR_MAX,
    PIN_TERMINATE
}
```

◆ receptionFlag

```
bool receptionFlag
```

Flag raised if femtosatellite has received commands

◆ result

```
EasyLink_Status result
```

Status of each RF command

◆ rxPacket

```
EasyLink_RxPacket rxPacket = {{0}, 0, 0, 0, 0, {0}}
```

Packet received from CubeSat

◆ txPacket

```
EasyLink_TxPacket txPacket = {{0}, 0, 0, {0}}
```

Packet transmitted to CubeSat

Generated by doxygen 1.8.20