

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

## Variables

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

## Detailed Description

Source file that allows communication between ground station and CubeSat.

=====

Copyright (c) 2017-2018, Texas Instruments Incorporated All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Texas Instruments Incorporated nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**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 CubeSat-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