

# CubeSat\_nortos.c File Reference

---

Source file for CubeSat. [More...](#)

```
#include <CubeSat_nortos.h>
#include <stdlib.h>
#include <stdbool.h>
#include <stdint.h>
#include <ti/drivers/rf/RF.h>
#include <ti/drivers/PIN.h>
#include <ti/drivers/Power.h>
#include <ti/drivers/SD.h>
#include <ti/display/Display.h>
#include <ti/devices/DeviceFamily.h>
#include "Board.h"
#include "smartrf_settings/smartrf_settings.h"
#include "easylink/EasyLink.h"
```

## Macros

---

```
#define PAYLOAD_LENGTH 30
#define GROUND_ADDRESS 0xFF
#define RX_TIMEOUT 500
#define BUFFSIZE 1024
#define STARTINGSECTOR 0
#define BYTESPERKILOBYTE 1024
#define WRITEENABLE 1
```

---

## Functions

---

void	<b>driverSetup</b> ()	Setup LED and display drivers. <a href="#">More...</a>
bool	<b>sdSetup</b> (int8_t rssi, uint8_t errorCode)	Function to initialise the (micro)SD card driver. <a href="#">More...</a>
bool	<b>sdWrite</b> (SD_Handle <b>sdHandle</b> , int_fast8_t <b>result</b> , bool sdFailure)	Write to (micro)SD card and check operation. <a href="#">More...</a>
void	<b>commandRx</b> ()	Enter RX mode to get commands from ground station. <a href="#">More...</a>
void	<b>groundStationAckTx</b> ()	Echo the packet from the ground station (ack) <a href="#">More...</a>
void	<b>commandTx</b> ()	Perform TX command to send commands to femtosat. <a href="#">More...</a>
void	<b>femtosatAckRx</b> ()	Enter RX mode to receive ack from femtosat. <a href="#">More...</a>

---

void	<b>dataTx</b> ()
Enter TX mode and send data to ground station. <a href="#">More...</a>	
bool	<b>isPacketCorrect</b> (EasyLink_RxPacket *rxp, EasyLink_TxPacket *txp)
Check that received packet is the same as transmitted packet. <a href="#">More...</a>	
void *	<b>mainThread</b> (void *arg0)

## Variables

static PIN_Handle	<b>pinHandle</b>
static PIN_State	<b>pinState</b>
static Display_Handle	<b>display</b>
SD_Handle	<b>sdHandle</b>
unsigned char	<b>sdPacket</b> [BUFSIZE]
unsigned char	<b>cpyBuff</b> [BUFSIZE]
EasyLink_Params	<b>easyLinkParams</b>
EasyLink_RxPacket	<b>rxPacket</b> = {{0}, 0, 0, 0, 0, {0}}
EasyLink_TxPacket	<b>txPacket</b> = {{0}, 0, 0, {0}}
uint32_t	<b>absTime</b>
EasyLink_Status	<b>result</b>
static bool	<b>bBlockTransmit</b> = false
PIN_Config	<b>pinTable</b> []

## Detailed Description

Source file for 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**

19/12/2020

---

## Macro Definition Documentation

---

### ◆ BUFFSIZE

```
#define BUFFSIZE 1024
```

Buffer size for microSD operations

### ◆ BYTESPERKILOBYTE

```
#define BYTESPERKILOBYTE 1024
```

Bytes in a kilobyte

### ◆ GROUND\_ADDRESS

```
#define GROUND_ADDRESS 0xFF
```

Ground station address for address filtering

### ◆ PAYLOAD\_LENGTH

```
#define PAYLOAD_LENGTH 30
```

Length of payload in bytes

### ◆ RX\_TIMEOUT

```
#define RX_TIMEOUT 500
```

RX command times out after 500ms

## ◆ STARTINGSECTOR

```
#define STARTINGSECTOR 0
```

Starting sector to write to/read from

## ◆ WRITEENABLE

```
#define WRITEENABLE 1
```

Enable/disable writing to microSD without file system

## Function Documentation

---

### ◆ commandRx()

```
void commandRx ( )
```

Enter RX mode to get commands from ground station.

#### Returns

none

### ◆ commandTx()

```
void commandTx ( )
```

Perform TX command to send commands to femtosat.

#### Returns

none

### ◆ dataTx()

```
void dataTx ( )
```

Enter TX mode and send data to ground station.

**Returns**

none

**◆ driverSetup()**

```
void driverSetup ( )
```

Setup LED and display drivers.

**Returns**

none

**◆ femtosatAckRx()**

```
void femtosatAckRx ( )
```

Enter RX mode to receive ack from femtosat.

**Returns**

none

**◆ groundStationAckTx()**

```
void groundStationAckTx ( )
```

Echo the packet from the ground station (ack)

**Returns**

none

**◆ isPacketCorrect()**

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

Check that received packet is the same as transmitted packet.

#### Parameters

**\*rxp** Pointer to RX packet.

**\*txp** Pointer to TX packet.

#### Returns

status Flag pulled low when RX and TX packets are not the same

### ◆ sdSetup()

```
bool sdSetup ( int8_t rssi,  
              uint8_t errorCode  
              )
```

Function to initialise the (micro)SD card driver.

#### Parameters

**rssi** RSSI value of RF link.

**errorCode** Error code of operation.

#### Returns

sdOpFlag Flag raised when microSD op fails.

### ◆ sdWrite()

```
bool sdWrite ( SD_Handle sdHandle,
               int_fast8_t result,
               bool      sdFailure
             )
```

Write to (micro)SD card and check operation.

#### Parameters

**sdHandle** SD driver handle

**result** Result of SD driver initialisation

#### Returns

sdFlag Flag raised in the event of microSD failure.

## Variable Documentation

### ◆ absTime

uint32\_t absTime

Used by RF core to time RF commands

### ◆ bBlockTransmit

bool bBlockTransmit = false

static

Flag raised when TX command is to be blocked

### ◆ cpyBuff

unsigned char cpyBuff[BUFFSIZE]

Buffer used to check the success of microSD operation

### ◆ display

Display\_Handle display

static

Handle for UART display driver

◆ easyLinkParams

EasyLink\_Params easyLinkParams

Use this to initialise EasyLink parameters to their default values

◆ pinHandle

PIN\_Handle pinHandle

static

Handle for pin driver

◆ pinState

PIN\_State pinState

static

Used for GPIO commands

◆ 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
}
```

◆ result

EasyLink\_Status result

Status of RF command



## ◆ sdHandle

SD\_Handle sdHandle

Handle for SD driver

## ◆ sdPacket

unsigned char sdPacket[**BUFFSIZE**]

Packet to be stored in microSD card

Generated by doxygen 1.8.20