mbed-lib

1.0

Generated by Doxygen 1.7.5.1

Sun Dec 18 2011 14:41:50

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Module Documentation

3.1 CAN baudrates

3.1.1 Detailed Description

Baudrates for the CAN interfaces. These are set via setCANBaudrate().

3.2 MBED_CAN0 buffers

3.2.1 Detailed Description

Receive and transmit buffers for the MBED_CAN0 interface.

3.3 MBED_CAN1 buffers

3.3.1 Detailed Description

Receive and transmit buffers for the MBED_CAN0 interface.

3.4 GPIO configuration

3.4.1 Detailed Description

Input and output configurations for General Purpose I/O. These are set via setGPInputCfg() and setGPOutputCfg().

3.5 Initalization flags

Defines

- #define INIT_LED (1 << 1)
- #define INIT_MUSB (1 << 2)
- #define INIT_SERIAL0 (1 << 3)
- #define INIT_SERIAL1 (1 << 4)
- #define INIT_SERIAL2 (1 << 5)
- #define INIT_CAN0 (1 << 6)
- #define INIT_CAN1 (1 << 7)
- #define INIT_TIMER0 (1 << 8)
- #define INIT_TIMER1 (1 << 9)
- #define INIT_TIMER2 (1 << 10)
- #define INIT_TIMER3 (1 << 11)
- #define INIT_GPI (1 << 12)
- #define INIT_GPO (1 << 13)

3.5.1 Detailed Description

These flags indicate which functionality should be initialized.

See also

initSys()

3.5.2 Define Documentation

3.5.2.1 #define INIT_CAN0 (1 << 6)

CAN on DIP 9/10 (CAN1).

3.5.2.2 #define INIT_CAN1 (1 << 7)

CAN on DIP 29/30 (CAN2).

3.5.2.3 #define INIT_GPI (1 << 12)

Initialize general purpose input ports.

3.5.2.4 #define INIT_GPO (1 << 13)

Initialize general purpose output ports.

3.5.2.5 #define INIT_LED (1 << 1)

Initialize the four blue LEDs.

3.5.2.6 #define INIT_MUSB (1 << 2)

Serial on mini USB-B port (UART0).

3.5.2.7 #define INIT_SERIAL0 (1 << 3)

Serial on DIP 13/14 (UART1).

3.5.2.8 #define INIT_SERIAL1 (1 << 4)

Serial on DIP 27/28 (UART2).

3.5.2.9 #define INIT_SERIAL2 (1 << 5)

Serial on DIP 9/10 (UART3).

3.5.2.10 #define INIT_TIMER0 (1 << 8)

Initialize TIMER0.

3.5.2.11 #define INIT_TIMER1 (1 << 9)

Initialize TIMER1.

3.5.2.12 #define INIT_TIMER2 (1 << 10)

Initialize TIMER2.

3.5.2.13 #define INIT_TIMER3 (1 << 11)

Initialize TIMER3.

3.6 Status flags

Defines

- #define MBED_SYS_INIT (1 << 0)
- #define MBED_LED_INIT (1 << 1)
- #define MBED_MUSB_INIT (1 << 2)
- #define MBED_SERIAL0_INIT (1 << 3)
- #define MBED_SERIAL1_INIT (1 << 4)
- #define MBED_SERIAL2_INIT (1 << 5)
- #define MBED_CAN0_INIT (1 << 6)
- #define MBED_CAN1_INIT (1 << 7)
- #define MBED_TIMER0_INIT (1 << 8)
- #define MBED_TIMER1_INIT (1 << 9)
- #define MBED_TIMER2_INIT (1 << 10)
- #define MBED_TIMER3_INIT (1 << 11)
- #define MBED_GPI_INIT (1 << 12)
- #define MBED_GPO_INIT (1 << 13)

3.6.1 Detailed Description

These flags indicate which functionality has been successfully initialized.

3.6.2 Define Documentation

3.6.2.1 #define MBED_CAN0_INIT (1 << 6)

CAN1 initialized.

3.6.2.2 #define MBED_CAN1_INIT (1 << 7)

CAN2 initialized.

3.6.2.3 #define MBED_GPI_INIT (1 << 12)

General purpose inputs initialized.

3.6.2.4 #define MBED_GPO_INIT (1 << 13)

General purpose outputs initialized.

3.6.2.5 #define MBED_LED_INIT (1 << 1)

The four LEDs are initialized.

3.6.2.6 #define MBED_MUSB_INIT (1 << 2)

Serial communication functionality on the mini USB-B port initialized.

3.6.2.7 #define MBED_SERIAL0_INIT (1 << 3)

UART1 initialized.

3.6.2.8 #define MBED_SERIAL1_INIT (1 << 4)

UART2 initialized.

3.6.2.9 #define MBED_SERIAL2_INIT (1 << 5)

UART3 initialized.

3.6.2.10 #define MBED_SYS_INIT (1 << 0)

System initialization complete.

3.6.2.11 #define MBED_TIMER0_INIT (1 << 8)

TIMER0 initialized.

3.6.2.12 #define MBED_TIMER1_INIT (1 << 9)

TIMER1 initialized.

3.6.2.13 #define MBED_TIMER2_INIT (1 << 10)

TIMER2 initialized.

3.6.2.14 #define MBED_TIMER3_INIT (1 << 11)

TIMER3 initialized.

3.7 General purpose I/O pins

Defines

- #define MBED_GPIOX (1 << 0)
- #define MBED_GPIO5 (1 << 1)
- #define MBED_GPIO6 (1 << 2)
- #define MBED_GPIO7 (1 << 3)
- #define MBED_GPIO8 (1 << 4)
- #define MBED_GPIO9 (1 << 5)
- #define MBED_GPIO10 (1 << 6)
- #define MBED_GPIO11 (1 << 7)
- #define MBED_GPIO12 (1 << 8)
- #define MBED_GPIO13 (1 << 9)
- #define MBED_GPIO14 (1 << 10)
- #define MBED_GPIO15 (1 << 11)
- #define MBED_GPIO16 (1 << 12)
- #define MBED_GPIO17 (1 << 13)
- #define MBED_GPIO18 (1 << 14)
- #define MBED_GPIO19 (1 << 15)
- #define MBED_GPIO20 (1 << 16)
- #define MBED_GPIO21 (1 << 17)
- #define MBED_GPIO22 (1 << 18)
- #define MBED_GPIO23 (1 << 19)
- #define MBED_GPIO24 (1 << 20)
- #define MBED_GPIO25 (1 << 21)
- #define MBED_GPIO26 (1 << 22)
- #define MBED_GPIO27 (1 << 23)
- #define MBED_GPIO28 (1 << 24)
- #define MBED_GPIO29 (1 << 25)
- #define MBED_GPIO30 (1 << 26)

3.7.1 Detailed Description

Enumeration of general purpose I/O pins.

3.7.2 Define Documentation

3.7.2.1 #define MBED_GPIO10 (1 << 6)

DIP 10.

3.7.2.2 #define MBED_GPIO11 (1 << 7)

DIP 11.

3.7.2.3 #define MBED_GPIO12 (1 << 8)

DIP 12.

3.7.2.4 #define MBED_GPIO13 (1 << 9)

DIP 13.

3.7.2.5 #define MBED_GPIO14 (1 << 10)

DIP 14.

3.7.2.6 #define MBED_GPIO15 (1 << 11)

DIP 15.

3.7.2.7 #define MBED_GPIO16 (1 << 12)

DIP 16.

3.7.2.8 #define MBED_GPIO17 (1 << 13)

DIP 17.

3.7.2.9 #define MBED_GPIO18 (1 << 14)

DIP 18.

3.7.2.10 #define MBED_GPIO19 (1 << 15)

DIP 19.

3.7.2.11 #define MBED_GPIO20 (1 << 16)

DIP 20.

3.7.2.12 #define MBED_GPIO21 (1 << 17)

DIP 21.

3.7.2.13 #define MBED_GPIO22 (1 << 18)

DIP 22.

3.7.2.14 #define MBED_GPIO23 (1 << 19)

DIP 23.

3.7.2.15 #define MBED_GPIO24 (1 << 20)

DIP 24.

3.7.2.16 #define MBED_GPIO25 (1 << 21)

DIP 25.

3.7.2.17 #define MBED_GPIO26 (1 << 22)

DIP 26.

3.7.2.18 #define MBED_GPIO27 (1 << 23)

DIP 27.

3.7.2.19 #define MBED_GPIO28 (1 << 24)

DIP 28.

3.7.2.20 #define MBED_GPIO29 (1 << 25)

DIP 29.

3.7.2.21 #define MBED_GPIO30 (1 << 26)

DIP 30.

3.7.2.22 #define MBED_GPIO5 (1 << 1)

DIP 5.

3.7.2.23 #define MBED_GPIO6 (1 << 2)

DIP 6.

3.7.2.24 #define MBED_GPIO7 (1 << 3)

DIP 7.

3.7.2.25 #define MBED_GPIO8 (1 << 4)

DIP 8.

3.7.2.26 #define MBED_GPIO9 (1 << 5)

DIP 9.

3.7.2.27 #define MBED_GPIOX (1 << 0)

Not configured.

3.8 Port configuration

Defines

- #define MBED_GPIO_P0 (MBED_GPIO5 | MBED_GPIO6 | MBED_GPIO7 | MBED_GPIO8 | MBED_GPIO9 | MBED_GPIO10 | MBED_GPIO11 | MBED_GPIO12 | MBED_GPIO13 | MBED_GPIO14 | MBED_GPIO15 | MBED_GPIO16 | MBED_GPIO17 | MBED_GPIO18 | MBED_GPIO27 | MBED_GPIO28 | MBED_GPIO29 | MBED_GPIO30)
- #define MBED_GPIO_P1 (MBED_GPIO19 | MBED_GPIO20)
- #define MBED_GPIO_P2 (MBED_GPIO21 | MBED_GPIO22 | MBED_GPIO23 | MBED_GPIO24 | MBED_GPIO25 | MBED_GPIO26)

3.8.1 Detailed Description

Configuration of GPIO pins on the different ports.

3.8.2 Define Documentation

3.8.2.1 #define MBED_GPIO_P0 (MBED_GPIO5 | MBED_GPIO6 | MBED_GPIO7 | MBED_GPIO8 | MBED_GPIO9 | MBED_GPIO10 | MBED_GPIO11 | MBED_GPIO12 | MBED_GPIO13 | MBED_GPIO14 | MBED_GPIO15 | MBED_GPIO16 | MBED_GPIO17 | MBED_GPIO18 | MBED_GPIO27 | MBED_GPIO28 | MBED_GPIO29 | MBED_GPIO30)

GPIO on port 0.

3.8.2.2 #define MBED_GPIO_P1 (MBED_GPIO19 | MBED_GPIO20)

GPIO on port 1.

3.8.2.3 #define MBED_GPIO_P2 (MBED_GPIO21 | MBED_GPIO22 | MBED_GPIO23 | MBED_GPIO24 | MBED_GPIO25 | MBED_GPIO26)

GPIO on port 2.

3.9 Device IDs

Defines

- #define MBED_SERIAL0 0
- #define MBED_SERIAL1 1
- #define MBED_SERIAL2 2
- #define MBED_CAN0 0
- #define MBED_CAN1 1
- #define MBED_TIMER0 0
- #define MBED_TIMER1 1
- #define MBED_TIMER2 2
- #define MBED_TIMER3 3

3.9.1 Detailed Description

ID enumeration of various devices.

3.9.2 Define Documentation

3.9.2.1 #define MBED_CAN0 0

CAN1 on DIP 9/10.

3.9.2.2 #define MBED_CAN1 1

CAN2 on DIP 29/30.

3.9.2.3 #define MBED_SERIAL0 0

UART1 on DIP 13/14.

3.9.2.4 #define MBED_SERIAL1 1

UART2 on DIP 27/28.

3.9.2.5 #define MBED_SERIAL2 2

UART3 on DIP 9/10.

3.9.2.6 #define MBED_TIMER0 0

TIMER0.

3.9.2.7 #define MBED_TIMER1 1

TIMER1.

3.9.2.8 #define MBED_TIMER2 2

TIMER2.

3.9.2.9 #define MBED_TIMER3 3

TIMER3.

3.10 CANopen objects

Defines

- #define CAN_COB_NMT (0 << 7)
- #define CAN_COB_SYNC (1 << 7)
- #define CAN_COB_TIME (2 << 7)
- #define CAN_COB_EMCY (1 << 7)
- #define CAN_COB_TPDO1 (3 << 7)
- #define CAN_COB_RPDO1 (4 << 7)
- #define CAN_COB_TPDO2 (5 << 7)
- #define CAN_COB_RPDO2 (6 << 7)
- #define CAN_COB_TPDO3 (7 << 7)
- #define CAN_COB_RPDO3 (8 << 7)
- #define CAN_COB_TPDO4 (9 << 7)
- #define CAN_COB_RPDO4 (10 << 7)
- #define CAN_COB_TSDO (11 << 7)
- #define CAN_COB_RSDO (12 << 7)
- #define CAN_COB_ERROR_CONTROL (14 << 7)

3.10.1 Detailed Description

These define numeric identifiers for a set of CANopen object types.

3.10.2 Define Documentation

3.10.2.1 #define CAN_COB_EMCY (1 << 7)

Emergency.

3.10.2.2 #define CAN_COB_ERROR_CONTROL (14 << 7)

Error control.

3.10.2.3 #define CAN_COB_NMT (0 << 7)

Network management.

3.10.2.4 #define CAN_COB_RPDO1 (4 << 7)

PDO1 receive.

3.10.2.5 #define CAN_COB_RPDO2 (6 << 7)

PDO2 receive.

3.10.2.6 #define CAN_COB_RPDO3 (8 << 7)

PDO3 receive.

3.10.2.7 #define CAN_COB_RPDO4 (10 << 7)

PDO4 receive.

3.10.2.8 #define CAN_COB_RSDO (12 << 7)

SDO receive.

3.10.2.9 #define CAN_COB_SYNC (1 << 7)

Synchronization.

3.10.2.10 #define CAN_COB_TIME (2 << 7)

Timestamp.

3.10.2.11 #define CAN_COB_TPDO1 (3 << 7)

PDO1 transmit.

3.10.2.12 #define CAN_COB_TPDO2 (5 << 7)

PDO2 transmit.

3.10.2.13 #define CAN_COB_TPDO3 (7 << 7)

PDO3 transmit.

3.10.2.14 #define CAN_COB_TPDO4 (9 << 7)

PDO4 transmit.

3.10.2.15 #define CAN_COB_TSDO (11 << 7)

SDO transmit.

3.11 Network management commands

Defines

- #define CAN_NMT_START 0x01
- #define CAN_NMT_STOP 0x02
- #define CAN_NMT_PREOP 0x80
- #define CAN_NMT_RESET 0x81
- #define CAN_NMT_RSCOMM 0x82

3.11.1 Detailed Description

CANOpen network management commands are used for node control.

3.11.2 Define Documentation

3.11.2.1 #define CAN_NMT_PREOP 0x80

Enter pre-operational state.

3.11.2.2 #define CAN_NMT_RESET 0x81

Reset node.

3.11.2.3 #define CAN_NMT_RSCOMM 0x82

Reset node communication.

3.11.2.4 #define CAN_NMT_START 0x01

Start node.

3.11.2.5 #define CAN_NMT_STOP 0x02

Stop node.

3.12 GPIO pin definitions

Defines

- #define GPIO5 9
- #define GPIO6 8
- #define GPIO7 7
- #define GPIO8 6
- #define GPIO9 0
- #define GPIO10 1
- #define GPIO11 18
- #define GPIO12 17
- #define GPIO13 15
- #define GPIO14 16
- #define GPIO15 23
- #define GPIO16 24
- #define GPIO17 25
- #define GPIO18 26
- #define GPIO19 30
- #define GPIO20 31
- #define GPIO21 5
- #define GPIO22 4
- #define GPIO23 3
- #define GPIO24 2
- #define GPIO25 1
- #define GPIO26 0
- #define GPIO27 11
- #define GPIO28 10
- #define GPIO29 5
- #define GPIO30 4

3.12.1 Detailed Description

These define how the DIP connections of the MBED devices are connected.

3.12.2 Define Documentation

3.12.2.1 #define GPIO10 1

DIP 10 is on position 1 of port 0.

3.12.2.2 #define GPIO11 18

DIP 11 is on position 18 of port 0.

3.12.2.3 #define GPIO12 17

DIP 12 is on position 17 of port 0.

3.12.2.4 #define GPIO13 15

DIP 13 is on position 15 of port 0.

3.12.2.5 #define GPIO14 16

DIP 14 is on position 16 of port 0.

3.12.2.6 #define GPIO15 23

DIP 15 is on position 23 of port 0.

3.12.2.7 #define GPIO16 24

DIP 16 is on position 24 of port 0.

3.12.2.8 #define GPIO17 25

DIP 17 is on position 25 of port 0.

3.12.2.9 #define GPIO18 26

DIP 18 is on position 26 of port 0.

3.12.2.10 #define GPIO19 30

DIP 19 is on position 30 of port 1.

3.12.2.11 #define GPIO20 31

DIP 20 is on position 31 of port 1.

3.12.2.12 #define GPIO21 5

DIP 21 is on position 5 of port 2.

3.12.2.13 #define GPIO22 4

DIP 22 is on position 4 of port 2.

3.12.2.14 #define GPIO23 3

DIP 23 is on position 3 of port 2.

3.12.2.15 #define GPIO24 2

DIP 24 is on position 2 of port 2.

3.12.2.16 #define GPIO25 1

DIP 25 is on position 1 of port 2.

3.12.2.17 #define GPIO26 0

DIP 26 is on position 0 of port 2.

3.12.2.18 #define GPIO27 11

DIP 27 is on position 11 of port 0.

3.12.2.19 #define GPIO28 10

DIP 28 is on position 10 of port 0.

3.12.2.20 #define GPIO29 5

DIP 29 is on position 5 of port 0.

3.12.2.21 #define GPIO30 4

Dlp 30 is on position 4 of port 0.

3.12.2.22 #define GPIO5 9

DIP 5 is on position 9 of port 0.

3.12.2.23 #define GPIO6 8

DIP 6 is on position 8 of port 0.

3.12.2.24 #define GPIO7 7

DIP 7 is on position 7 of port 0.

3.12.2.25 #define GPIO8 6

DIP 8 is on position 6 of port 0.

3.12.2.26 #define GPIO9 0

DIP 9 is on position 0 of port 0.

3.13 GPIO levels

Defines

- #define GP_LOW 0
- #define GP_HIGH 1
- #define GP_ERR 2

3.13.1 Detailed Description

Set and read values for GPIO ports.

See also

GPIOSetVal()
GPIOReadVal()

3.13.2 Define Documentation

3.13.2.1 #define GP_ERR 2

Indicates that the pin is not configured.

3.13.2.2 #define GP_HIGH 1

High level (+3.3V)/

3.13.2.3 #define GP_LOW 0

Low level (ground).

Chapter 4

File Documentation

4.1 mbed.h File Reference

In this file are all device definitions for the mbed platform.

```
#include "LPC17xx.h" #include "mbed_can.h" #include "mbed-
_gpio.h"   #include "mbed_led.h"   #include "mbed_musb.h" ×
#include "mbed_serial.h" #include "mbed_timer.h"
```

Defines

- #define MBED_H
- #define INIT_LED (1 << 1)
- #define INIT_MUSB (1 << 2)
- #define INIT_SERIAL0 (1 << 3)
- #define INIT_SERIAL1 (1 << 4)
- #define INIT_SERIAL2 (1 << 5)
- #define INIT_CAN0 (1 << 6)
- #define INIT_CAN1 (1 << 7)
- #define INIT_TIMER0 (1 << 8)
- #define INIT_TIMER1 (1 << 9)
- #define INIT_TIMER2 (1 << 10)
- #define INIT_TIMER3 (1 << 11)
- #define INIT_GPI (1 << 12)
- #define INIT_GPO (1 << 13)
- #define MBED_SYS_INIT (1 << 0)
- #define MBED_LED_INIT (1 << 1)
- #define MBED_MUSB_INIT (1 << 2)
- #define MBED_SERIAL0_INIT (1 << 3)
- #define MBED_SERIAL1_INIT (1 << 4)
- #define MBED_SERIAL2_INIT (1 << 5)
- #define MBED_CAN0_INIT (1 << 6)

- #define MBED_CAN1_INIT (1 << 7)
- #define MBED_TIMER0_INIT (1 << 8)
- #define MBED_TIMER1_INIT (1 << 9)
- #define MBED_TIMER2_INIT (1 << 10)
- #define MBED_TIMER3_INIT (1 << 11)
- #define MBED_GPI_INIT (1 << 12)
- #define MBED_GPO_INIT (1 << 13)
- #define MBED_GPIOX (1 << 0)
- #define MBED_GPIO5 (1 << 1)
- #define MBED_GPIO6 (1 << 2)
- #define MBED_GPIO7 (1 << 3)
- #define MBED_GPIO8 (1 << 4)
- #define MBED_GPIO9 (1 << 5)
- #define MBED_GPIO10 (1 << 6)
- #define MBED_GPIO11 (1 << 7)
- #define MBED_GPIO12 (1 << 8)
- #define MBED_GPIO13 (1 << 9)
- #define MBED_GPIO14 (1 << 10)
- #define MBED_GPIO15 (1 << 11)
- #define MBED_GPIO16 (1 << 12)
- #define MBED_GPIO17 (1 << 13)
- #define MBED_GPIO18 (1 << 14)
- #define MBED_GPIO19 (1 << 15)
- #define MBED_GPIO20 (1 << 16)
- #define MBED_GPIO21 (1 << 17)
- #define MBED_GPIO22 (1 << 18)
- #define MBED_GPIO23 (1 << 19)
- #define MBED_GPIO24 (1 << 20)
- #define MBED_GPIO25 (1 << 21)
- #define MBED_GPIO26 (1 << 22)
- #define MBED_GPIO27 (1 << 23)
- #define MBED_GPIO28 (1 << 24)
- #define MBED_GPIO29 (1 << 25)
- #define MBED_GPIO30 (1 << 26)
- #define MBED_GPIO_P0 (MBED_GPIO5 | MBED_GPIO6 | MBED_GPIO7 | MBED_GPIO8 | MBED_GPIO9 | MBED_GPIO10 | MBED_GPIO11 | MBED_GPIO12 | MBED_GPIO13 | MBED_GPIO14 | MBED_GPIO15 | MBED_GPIO16 | MBED_GPIO17 | MBED_GPIO18 | MBED_GPIO27 | MBED_GPIO28 | MBED_GPIO29 | MBED_GPIO30)
- #define MBED_GPIO_P1 (MBED_GPIO19 | MBED_GPIO20)
- #define MBED_GPIO_P2 (MBED_GPIO21 | MBED_GPIO22 | MBED_GPIO23 | MBED_GPIO24 | MBED_GPIO25 | MBED_GPIO26)
- #define MBED_SERIAL0 0
- #define MBED_SERIAL1 1
- #define MBED_SERIAL2 2
- #define MBED_CAN0 0
- #define MBED_CAN1 1

- #define MBED_TIMER0 0
- #define MBED_TIMER1 1
- #define MBED_TIMER2 2
- #define MBED_TIMER3 3

Functions

- void initSys (uint32_t)

  System initialization function.

Variables

- uint32_t mbedStatus

4.1.1 Detailed Description

In this file are all device definitions for the mbed platform. Via these defines, all hardware peripherals are identified and initialized. Initalization flags are used to indicate which peripherals should be initialized upons startup. Status flags set in mbedStatus then indicate which peripherals have been successfully initialized.

A number of device IDs is defined to identify various devices.

All mbed header files are included here for convience, so that including this header file is sufficient.

4.1.2 Define Documentation

4.1.2.1 #define MBED_H

Header guard.

4.1.3 Function Documentation

4.1.3.1 void initSys (uint32_t *flags*)

System initialization function.

This function initializes the LPC1768 clock registers and any other functionality that is required.

Parameters

flags	Functionality to be initialized, indicated by setting corresponding bits.

See also

Initalization flags

4.1.4 Variable Documentation

4.1.4.1 uint32_t mbedStatus

Global status register.

See also

Status flags

4.2 mbed_can.h File Reference

Various definitions for the CANopen implementation.

```
#include "LPC17xx.h"   #include "lpc17xx_can.h"   #include
"lpc17xx_pinsel.h"
```

Defines

- #define MBED_CAN_H
- #define CAN_COB_NMT (0 << 7)
- #define CAN_COB_SYNC (1 << 7)
- #define CAN_COB_TIME (2 << 7)
- #define CAN_COB_EMCY (1 << 7)
- #define CAN_COB_TPDO1 (3 << 7)
- #define CAN_COB_RPDO1 (4 << 7)
- #define CAN_COB_TPDO2 (5 << 7)
- #define CAN_COB_RPDO2 (6 << 7)
- #define CAN_COB_TPDO3 (7 << 7)
- #define CAN_COB_RPDO3 (8 << 7)
- #define CAN_COB_TPDO4 (9 << 7)
- #define CAN_COB_RPDO4 (10 << 7)
- #define CAN_COB_TSDO (11 << 7)
- #define CAN_COB_RSDO (12 << 7)
- #define CAN_COB_ERROR_CONTROL (14 << 7)
- #define CAN_NMT_START 0x01
- #define CAN_NMT_STOP 0x02
- #define CAN_NMT_PREOP 0x80
- #define CAN_NMT_RESET 0x81
- #define CAN_NMT_RSCOMM 0x82
- #define CAN_BUFSIZE 256

Functions

- void setCANBaudrate (uint8_t, uint32_t)

  Baudrate configuration function.
- void initCAN (uint8_t)

  CAN initialization function.
- uint8_t CANopenRecv (uint8_t, uint8_t *, uint32_t *, uint8_t *)

  Receive a CAN message via CANopen protocol.
- uint8_t CANopenSend (uint8_t, uint8_t, uint32_t, uint8_t, uint8_t *)

  Send a CAN message via CANopen protocol.
- void flushCANRxBuffer (uint8_t)

  Flush CAN receive buffers.
- void flushCANTxBuffer (uint8_t)

  Flush CAN transmit buffers.
- void CAN_IRQHandler (void)

  CAN IRQ handler.

4.2.1 Detailed Description

Various definitions for the CANopen implementation. This file defines a basic number of objects and messages for the CANopen implementation provided by mbed-lib, and it provides function definitions.

4.2.2 Define Documentation

4.2.2.1 #define CAN_BUFSIZE 256

CAN buffer size.

See also

MBED_CAN0 buffers
MBED_CAN1 buffers

4.2.2.2 #define MBED_CAN_H

Header guard.

4.2.3 Function Documentation

4.2.3.1 void CAN_IRQHandler (void)

CAN IRQ handler.

This routine is called when either of the CAN devices raises an interrupt. Do no call this routine yourself.

4.2.3.2 uint8_t CANopenRecv (uint8_t *portNo*, uint8_t * *nodeID*, uint32_t * *cobType*, uint8_t * *data*)

Receive a CAN message via CANopen protocol.

This routine reads an availabe message from the CAN receive buffer, if available.

Parameters

portNo	CAN interface to read from. Should be MBED_CAN0 or MBED_CAN1.
nodeID	Address of variable to store the CAN device ID that sent the message.
cobType	Address of variable to store the COB type of the message.
data	Pointer to data storage.

Returns

0 if a message was successfully read, or an integer larger than 0 when no message was available.

See also

CANopenSend()
CANopen objects

4.2.3.3 uint8_t CANopenSend (uint8_t *portNo*, uint8_t *nodeID*, uint32_t *cobType*, uint8_t *len*, uint8_t * *data*)

Send a CAN message via CANopen protocol.

This routine sends a message via the CAN interface. If the message could not be sent (because the internal transmit buffers are full), the message is buffered and sent when when of the internal transmit buffers becomes availble.

Parameters

portNo	CAN interface to send on. Should be MBED_CAN0 or MBED_CAN1.
nodeID	ID of the CAN device the message should be sent to.
cobType	COB type of the message.
len	Message length.
data	Data to be sent. The total number of bytes that will be sent depends on the provided message length, but will never be more than eight bytes.

Returns

0 if a message was successfully sent or buffered, or an integer larger than 0 when both sending and buffering failed.

See also

CANopenRecv()
CANopen objects

4.2.3.4 void flushCANRxBuffer (uint8_t *portNo*)

Flush CAN receive buffers.

This routine flushes the CAN receive buffer by discarding all messages.

Parameters

portNo	CAN interface of which the buffer should be flushed. Should be MBED_CAN0 or MBED_CAN1.

See also

flushCANTxBuffer()

4.2.3.5 void flushCANTxBuffer (uint8_t *portNo*)

Flush CAN transmit buffers.

This routine flushes the CAN transmit buffer by discarding all messages.

Parameters

portNo	CAN interface of which the buffer should be flushed. Should be MBED_CAN0 or MBED_CAN1.

See also

flushCANRxBuffer()

4.2.3.6 void initCAN (uint8_t *portNo*)

CAN initialization function.

Initalizes the CAN interface. This function is called from initSys(), only call this function if you know what you are doing.

Parameters

portNo	CAN interface to be initialized. Should be MBED_CAN0 or MBED_CAN1.

See also

initSys()

4.2.3.7 void setCANBaudrate (uint8_t *portNo*, uint32_t *baudrate*)

Baudrate configuration function.

This function sets the baudrates for the MBED_CAN0 and MBED_CAN1 devices. Call this function before calling initSys().

Parameters

portNo	CAN interface to set the baudrate for. Should be MBED_CAN0 or MBED_CAN1.
baudrate	Desired baudrate, in bits per second.

4.3 mbed_gpio.h File Reference

Various definition for General Purpose Input/Output (GPIO).

```
#include "LPC17xx.h" #include "lpc17xx_gpio.h"
```

Defines

- #define MBED_GPIO_H
- #define GPIO5 9
- #define GPIO6 8
- #define GPIO7 7
- #define GPIO8 6
- #define GPIO9 0
- #define GPIO10 1
- #define GPIO11 18
- #define GPIO12 17
- #define GPIO13 15
- #define GPIO14 16
- #define GPIO15 23
- #define GPIO16 24
- #define GPIO17 25
- #define GPIO18 26
- #define GPIO19 30
- #define GPIO20 31
- #define GPIO21 5
- #define GPIO22 4
- #define GPIO23 3
- #define GPIO24 2

- #define GPIO25 1
- #define GPIO26 0
- #define GPIO27 11
- #define GPIO28 10
- #define GPIO29 5
- #define GPIO30 4
- #define GP_LOW 0
- #define GP_HIGH 1
- #define GP_ERR 2

Functions

- void setGPInputCfg (uint32_t)

  General Purpose Input Configuration.
- void setGPOutputCfg (uint32_t)

  General Purpose Output Configuration.
- void initGPInputs (void)

  General Purpose Input initialization function.
- void initGPOutputs (void)

  General Purpose Output initialization function.
- void checkConflicts (void)

  Conflict resolver.
- void initGPIO (uint32_t, uint32_t)

  GPIO initialization function.
- void GPIOSetVal (uint32_t, uint8_t)

  Set value for general purpose outputs.
- void GPIOToggle (uint32_t)

  Toggle value for general purpose outputs.
- uint8_t GPIOReadVal (uint32_t)

  Read value from general purpose inputs.

4.3.1 Detailed Description

Various definition for General Purpose Input/Output (GPIO). In this file are definitions for all pins on the MBED device. They can be used as either input or output by adding them to the corresponding configuration.

4.3.2 Define Documentation

4.3.2.1 #define MBED_GPIO_H

Header guard.

4.3.3 Function Documentation

4.3.3.1 void checkConflicts (void)

Conflict resolver.

This routine checks if the GPIO configuration conflicts with any initlialized peripherals. For any conflict, it unsets the pins in the GPIO configuration.

See also

Device IDs

4.3.3.2 uint8_t GPIOReadVal (uint32_t *pin*)

Read value from general purpose inputs.

This routine reads the output value from the indicated GP pin.

Parameters

pin	Pin to be read, should be MBED_GPIOx.

Returns

GP_LOW or GP_HIGH if the pin is correctly configured, or GP_ERR if the pin was not properly configured.

See also

General purpose I/O pins
GPIO levels
GPIOSetVal()
GPIOToggle()

4.3.3.3 void GPIOSetVal (uint32_t *pin*, uint8_t *val*)

Set value for general purpose outputs.

This routine sets the output value for the indicated GP pin.

Parameters

pin	Pin to be set, should be MBED_GPIOx.
val	Value to set, should be GP_LOW or GP_HIGH.

See also

General purpose I/O pins
GPIO levels
GPIOToggle()
GPIOReadVal()

4.3.3.4 void GPIOToggle (uint32_t *pin*)

Toggle value for general purpose outputs.

This routine toggles the output value for the indicated GP pin.

Parameters

pin	Pin to be toggled, should be MBED_GPIOx.

See also

General purpose I/O pins
GPIO levels
GPIOSetVal()
GPIOReadVal()

4.3.3.5 void initGPInputs (void)

General Purpose Input initialization function.

This routine initializes the pins set by setGPInputCfg() as general purpose inputs, unless they conflict with any other previously initialized peripherals.

See also

initGPOutputs()
checkConflicts()

4.3.3.6 void initGPIO (uint32_t *mode*, uint32_t *gpioConfig*)

GPIO initialization function.

This routine configures the pins according to the GPIO configuration. It is called from initGPInputs() and initGPOutputs().

Parameters

mode	Indicates what is to be initialized (inputs or outputs). Should be INIT_GPI or INIT_GPO.
gpioConfig	Pin configuration (see GPIO configuration).

See also

setGPInputCfg()
setGPOutputCfg()
initGPInputs()
initGPOutputs()

4.3.3.7 void initGPOutputs (void)

General Purpose Output initialization function.

This routine initializes the pins set by setGPOutputCfg() as general purpose outputs, unless they conflict with any other previously initialized peripherals.

See also

initGPInputs()
checkConflicts()

4.3.3.8 void setGPInputCfg (uint32_t *config*)

General Purpose Input Configuration.

This function defines which pins should be configured as general purpose inputs by setting the GPIO configuration.

Parameters

config	Pins to be configured as inputs, indicated by setting the corresponding bits.

See also

General purpose I/O pins

4.3.3.9 void setGPOutputCfg (uint32_t *config*)

General Purpose Output Configuration.

This function defines which pins should be configured as general purpose outputs by setting the GPIO configuration.

Parameters

config	Pins to be configured as outputs, indicating by setting the corresponding bits.

See also

General purpose I/O pins