

mbed-lib

1.0

Generated by Doxygen 1.7.5.1

Sun Nov 27 2011 14:45:42

Contents

1	Module Index	1
1.1	Modules	1
2	File Index	3
2.1	File List	3
3	Module Documentation	5
3.1	CAN baudrates	5
3.1.1	Detailed Description	5
3.2	MBED_CAN0 buffers	6
3.2.1	Detailed Description	6
3.3	MBED_CAN1 buffers	7
3.3.1	Detailed Description	7
3.4	Initialization flags	8
3.4.1	Detailed Description	8
3.4.2	Define Documentation	8
3.4.2.1	INIT_CAN0	8
3.4.2.2	INIT_CAN1	8
3.4.2.3	INIT_GPI	8
3.4.2.4	INIT_GPO	8
3.4.2.5	INIT_LED	9
3.4.2.6	INIT_MUSB	9
3.4.2.7	INIT_SERIAL0	9
3.4.2.8	INIT_SERIAL1	9
3.4.2.9	INIT_SERIAL2	9
3.4.2.10	INIT_TIMER0	9

3.4.2.11	INIT_TIMER1	9
3.4.2.12	INIT_TIMER2	9
3.4.2.13	INIT_TIMER3	9
3.5	Status flags	10
3.5.1	Detailed Description	10
3.5.2	Define Documentation	10
3.5.2.1	MBED_CAN0_INIT	10
3.5.2.2	MBED_CAN1_INIT	10
3.5.2.3	MBED_GPI_INIT	10
3.5.2.4	MBED_GPO_INIT	10
3.5.2.5	MBED_LED_INIT	10
3.5.2.6	MBED_MUSB_INIT	11
3.5.2.7	MBED_SERIAL0_INIT	11
3.5.2.8	MBED_SERIAL1_INIT	11
3.5.2.9	MBED_SERIAL2_INIT	11
3.5.2.10	MBED_SYS_INIT	11
3.5.2.11	MBED_TIMER0_INIT	11
3.5.2.12	MBED_TIMER1_INIT	11
3.5.2.13	MBED_TIMER2_INIT	11
3.5.2.14	MBED_TIMER3_INIT	11
3.6	General purpose I/O pins	12
3.6.1	Detailed Description	12
3.6.2	Define Documentation	12
3.6.2.1	MBED_GPIO10	12
3.6.2.2	MBED_GPIO11	12
3.6.2.3	MBED_GPIO12	13
3.6.2.4	MBED_GPIO13	13
3.6.2.5	MBED_GPIO14	13
3.6.2.6	MBED_GPIO15	13
3.6.2.7	MBED_GPIO16	13
3.6.2.8	MBED_GPIO17	13
3.6.2.9	MBED_GPIO18	13
3.6.2.10	MBED_GPIO19	13
3.6.2.11	MBED_GPIO20	13

3.6.2.12	MBED_GPIO21	13
3.6.2.13	MBED_GPIO22	14
3.6.2.14	MBED_GPIO23	14
3.6.2.15	MBED_GPIO24	14
3.6.2.16	MBED_GPIO25	14
3.6.2.17	MBED_GPIO26	14
3.6.2.18	MBED_GPIO27	14
3.6.2.19	MBED_GPIO28	14
3.6.2.20	MBED_GPIO29	14
3.6.2.21	MBED_GPIO30	14
3.6.2.22	MBED_GPIO5	14
3.6.2.23	MBED_GPIO6	15
3.6.2.24	MBED_GPIO7	15
3.6.2.25	MBED_GPIO8	15
3.6.2.26	MBED_GPIO9	15
3.6.2.27	MBED_GPIOX	15
3.7	Port configuration	16
3.7.1	Detailed Description	16
3.7.2	Define Documentation	16
3.7.2.1	MBED_GPIO_P0	16
3.7.2.2	MBED_GPIO_P1	16
3.7.2.3	MBED_GPIO_P2	16
3.8	Device IDs	17
3.8.1	Detailed Description	17
3.8.2	Define Documentation	17
3.8.2.1	MBED_CAN0	17
3.8.2.2	MBED_CAN1	17
3.8.2.3	MBED_SERIAL0	17
3.8.2.4	MBED_SERIAL1	17
3.8.2.5	MBED_SERIAL2	17
3.8.2.6	MBED_TIMER0	17
3.8.2.7	MBED_TIMER1	18
3.8.2.8	MBED_TIMER2	18
3.8.2.9	MBED_TIMER3	18

3.9	CANopen objects	19
3.9.1	Detailed Description	19
3.9.2	Define Documentation	19
3.9.2.1	CAN_COB_EMCY	19
3.9.2.2	CAN_COB_ERROR_CONTROL	19
3.9.2.3	CAN_COB_NMT	19
3.9.2.4	CAN_COB_RPDO1	19
3.9.2.5	CAN_COB_RPDO2	20
3.9.2.6	CAN_COB_RPDO3	20
3.9.2.7	CAN_COB_RPDO4	20
3.9.2.8	CAN_COB_RSDO	20
3.9.2.9	CAN_COB_SYNC	20
3.9.2.10	CAN_COB_TIME	20
3.9.2.11	CAN_COB_TPDO1	20
3.9.2.12	CAN_COB_TPDO2	20
3.9.2.13	CAN_COB_TPDO3	20
3.9.2.14	CAN_COB_TPDO4	20
3.9.2.15	CAN_COB_TSDO	21
3.10	Network management commands	22
3.10.1	Detailed Description	22
3.10.2	Define Documentation	22
3.10.2.1	CAN_NMT_PREOP	22
3.10.2.2	CAN_NMT_RESET	22
3.10.2.3	CAN_NMT_RSCOMM	22
3.10.2.4	CAN_NMT_START	22
3.10.2.5	CAN_NMT_STOP	22
4	File Documentation	23
4.1	mbed.h File Reference	23
4.1.1	Detailed Description	25
4.1.2	Define Documentation	25
4.1.2.1	MBED_H	25
4.1.3	Function Documentation	25
4.1.3.1	initSys	25

4.1.4	Variable Documentation	26
4.1.4.1	mbedStatus	26
4.2	mbed_can.h File Reference	26
4.2.1	Detailed Description	27
4.2.2	Define Documentation	27
4.2.2.1	CAN_BUFSIZE	27
4.2.2.2	MBED_CAN_H	27
4.2.3	Function Documentation	27
4.2.3.1	CAN_IRQHandler	27
4.2.3.2	CANopenRecv	28
4.2.3.3	CANopenSend	28
4.2.3.4	flushCANRxBuffer	29
4.2.3.5	flushCANTxBuffer	29
4.2.3.6	initCAN	29
4.2.3.7	setCANBaudrate	30

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

CAN baudrates	5
MBED_CAN0 buffers	6
MBED_CAN1 buffers	7
Initialization flags	8
Status flags	10
General purpose I/O pins	12
Port configuration	16
Device IDs	17
CANopen objects	19
Network management commands	22

Chapter 2

File Index

2.1 File List

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

mbed.h	In this file are all device definitions for the mbed platform	23
mbed_can.h	Various definitions for the CANopen implementation	26
mbed_gpio.h	??
mbed_led.h	??
mbed_musb.h	??
mbed_serial.h	??
mbed_timer.h	??

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 Initialization 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)`

3.4.1 Detailed Description

These flags indicate which functionality should be initialized.

See also

[initSys\(\)](#)

3.4.2 Define Documentation

3.4.2.1 `#define INIT_CAN0 (1 << 6)`

CAN on DIP 9/10 (CAN1).

3.4.2.2 `#define INIT_CAN1 (1 << 7)`

CAN on DIP 29/30 (CAN2).

3.4.2.3 `#define INIT_GPI (1 << 12)`

Initialize general purpose input ports.

3.4.2.4 `#define INIT_GPO (1 << 13)`

Initialize general purpose output ports.

3.4.2.5 #define INIT_LED (1 << 1)

Initialize the four blue LEDs.

3.4.2.6 #define INIT_MUSB (1 << 2)

Serial on mini USB-B port (UART0).

3.4.2.7 #define INIT_SERIAL0 (1 << 3)

Serial on DIP 13/14 (UART1).

3.4.2.8 #define INIT_SERIAL1 (1 << 4)

Serial on DIP 27/28 (UART2).

3.4.2.9 #define INIT_SERIAL2 (1 << 5)

Serial on DIP 9/10 (UART3).

3.4.2.10 #define INIT_TIMER0 (1 << 8)

Initialize TIMER0.

3.4.2.11 #define INIT_TIMER1 (1 << 9)

Initialize TIMER1.

3.4.2.12 #define INIT_TIMER2 (1 << 10)

Initialize TIMER2.

3.4.2.13 #define INIT_TIMER3 (1 << 11)

Initialize TIMER3.

3.5 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)`

3.5.1 Detailed Description

These flags indicate which functionality has been successfully initialized.

3.5.2 Define Documentation

3.5.2.1 `#define MBED_CAN0_INIT (1 << 6)`

CAN1 initialized.

3.5.2.2 `#define MBED_CAN1_INIT (1 << 7)`

CAN2 initialized.

3.5.2.3 `#define MBED_GPI_INIT (1 << 12)`

General purpose inputs initialized.

3.5.2.4 `#define MBED_GPO_INIT (1 << 13)`

General purpose outputs initialized.

3.5.2.5 `#define MBED_LED_INIT (1 << 1)`

The four LEDs are initialized.

3.5.2.6 #define MBED_MUSB_INIT (1 << 2)

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

3.5.2.7 #define MBED_SERIAL0_INIT (1 << 3)

UART1 initialized.

3.5.2.8 #define MBED_SERIAL1_INIT (1 << 4)

UART2 initialized.

3.5.2.9 #define MBED_SERIAL2_INIT (1 << 5)

UART3 initialized.

3.5.2.10 #define MBED_SYS_INIT (1 << 0)

System initialization complete.

3.5.2.11 #define MBED_TIMER0_INIT (1 << 8)

TIMER0 initialized.

3.5.2.12 #define MBED_TIMER1_INIT (1 << 9)

TIMER1 initialized.

3.5.2.13 #define MBED_TIMER2_INIT (1 << 10)

TIMER2 initialized.

3.5.2.14 #define MBED_TIMER3_INIT (1 << 11)

TIMER3 initialized.

3.6 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)`

3.6.1 Detailed Description

Enumeration of general purpose I/O pins.

3.6.2 Define Documentation

3.6.2.1 `#define MBED_GPIO10 (1 << 6)`

DIP 10.

3.6.2.2 `#define MBED_GPIO11 (1 << 7)`

DIP 11.

3.6.2.3 `#define MBED_GPIO12 (1 << 8)`

DIP 12.

3.6.2.4 `#define MBED_GPIO13 (1 << 9)`

DIP 13.

3.6.2.5 `#define MBED_GPIO14 (1 << 10)`

DIP 14.

3.6.2.6 `#define MBED_GPIO15 (1 << 11)`

DIP 15.

3.6.2.7 `#define MBED_GPIO16 (1 << 12)`

DIP 16.

3.6.2.8 `#define MBED_GPIO17 (1 << 13)`

DIP 17.

3.6.2.9 `#define MBED_GPIO18 (1 << 14)`

DIP 18.

3.6.2.10 `#define MBED_GPIO19 (1 << 15)`

DIP 19.

3.6.2.11 `#define MBED_GPIO20 (1 << 16)`

DIP 20.

3.6.2.12 `#define MBED_GPIO21 (1 << 17)`

DIP 21.

3.6.2.13 `#define MBED_GPIO22 (1 << 18)`

DIP 22.

3.6.2.14 `#define MBED_GPIO23 (1 << 19)`

DIP 23.

3.6.2.15 `#define MBED_GPIO24 (1 << 20)`

DIP 24.

3.6.2.16 `#define MBED_GPIO25 (1 << 21)`

DIP 25.

3.6.2.17 `#define MBED_GPIO26 (1 << 22)`

DIP 26.

3.6.2.18 `#define MBED_GPIO27 (1 << 23)`

DIP 27.

3.6.2.19 `#define MBED_GPIO28 (1 << 24)`

DIP 28.

3.6.2.20 `#define MBED_GPIO29 (1 << 25)`

DIP 29.

3.6.2.21 `#define MBED_GPIO30 (1 << 26)`

DIP 30.

3.6.2.22 `#define MBED_GPIO5 (1 << 1)`

DIP 5.

3.6.2.23 `#define MBED_GPIO6 (1 << 2)`

DIP 6.

3.6.2.24 `#define MBED_GPIO7 (1 << 3)`

DIP 7.

3.6.2.25 `#define MBED_GPIO8 (1 << 4)`

DIP 8.

3.6.2.26 `#define MBED_GPIO9 (1 << 5)`

DIP 9.

3.6.2.27 `#define MBED_GPIOX (1 << 0)`

Not configured.

3.7 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)

3.7.1 Detailed Description

Configuration of GPIO pins on the different ports.

3.7.2 Define Documentation

3.7.2.1 `#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.

3.7.2.2 `#define MBED_GPIO_P1` (MBED_GPIO19 | MBED_GPIO20)

GPIO on port 1.

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

GPIO on port 2.

3.8 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`

3.8.1 Detailed Description

ID enumeration of various devices.

3.8.2 Define Documentation

3.8.2.1 `#define MBED_CAN0 0`

CAN1 on DIP 9/10.

3.8.2.2 `#define MBED_CAN1 1`

CAN2 on DIP 29/30.

3.8.2.3 `#define MBED_SERIAL0 0`

UART1 on DIP 13/14.

3.8.2.4 `#define MBED_SERIAL1 1`

UART2 on DIP 27/28.

3.8.2.5 `#define MBED_SERIAL2 2`

UART3 on DIP 9/10.

3.8.2.6 `#define MBED_TIMER0 0`

TIMER0.

3.8.2.7 `#define MBED_TIMER1 1`

TIMER1.

3.8.2.8 `#define MBED_TIMER2 2`

TIMER2.

3.8.2.9 `#define MBED_TIMER3 3`

TIMER3.

3.9 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)

3.9.1 Detailed Description

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

3.9.2 Define Documentation

3.9.2.1 #define `CAN_COB_EMCY` (1 << 7)

Emergency.

3.9.2.2 #define `CAN_COB_ERROR_CONTROL` (14 << 7)

Error control.

3.9.2.3 #define `CAN_COB_NMT` (0 << 7)

Network management.

3.9.2.4 #define `CAN_COB_RPDO1` (4 << 7)

PDO1 receive.

3.9.2.5 #define CAN_COB_RPDO2 (6 << 7)

PDO2 receive.

3.9.2.6 #define CAN_COB_RPDO3 (8 << 7)

PDO3 receive.

3.9.2.7 #define CAN_COB_RPDO4 (10 << 7)

PDO4 receive.

3.9.2.8 #define CAN_COB_RSDO (12 << 7)

SDO receive.

3.9.2.9 #define CAN_COB_SYNC (1 << 7)

Synchronization.

3.9.2.10 #define CAN_COB_TIME (2 << 7)

Timestamp.

3.9.2.11 #define CAN_COB_TPDO1 (3 << 7)

PDO1 transmit.

3.9.2.12 #define CAN_COB_TPDO2 (5 << 7)

PDO2 transmit.

3.9.2.13 #define CAN_COB_TPDO3 (7 << 7)

PDO3 transmit.

3.9.2.14 #define CAN_COB_TPDO4 (9 << 7)

PDO4 transmit.

3.9.2.15 #define CAN_COB_TSDO (11 << 7)

SDO transmit.

3.10 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`

3.10.1 Detailed Description

CANOpen network management commands are used for node control.

3.10.2 Define Documentation

3.10.2.1 `#define CAN_NMT_PREOP 0x80`

Enter pre-operational state.

3.10.2.2 `#define CAN_NMT_RESET 0x81`

Reset node.

3.10.2.3 `#define CAN_NMT_RSCOMM 0x82`

Reset node communication.

3.10.2.4 `#define CAN_NMT_START 0x01`

Start node.

3.10.2.5 `#define CAN_NMT_STOP 0x02`

Stop node.

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. [Initialization flags](#) are used to indicate which peripherals should be initialized upon 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 convenience, 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

<i>flags</i>	Functionality to be initialized, indicated by setting corresponding bits.
--------------	---

See also

[Initialization 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 not 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 available message from the CAN receive buffer, if available.

Parameters

<i>portNo</i>	CAN interface to read from. Should be MBED_CAN0 or MBED_CAN1.
<i>nodeID</i>	Address of variable to store the CAN device ID that sent the message.
<i>cobType</i>	Address of variable to store the COB type of the message.
<i>data</i>	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 available.

Parameters

<i>portNo</i>	CAN interface to send on. Should be MBED_CAN0 or MBED_CAN1.
<i>nodeID</i>	ID of the CAN device the message should be sent to.
<i>cobType</i>	COB type of the message.
<i>len</i>	Message length.
<i>data</i>	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

<i>portNo</i>	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

<i>portNo</i>	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.

Initializes the CAN interface. This function is called from [initSys\(\)](#), only call this function if you know what you are doing.

Parameters

<i>portNo</i>	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

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