

mbed-lib

1.0

Generated by Doxygen 1.7.5.1

Fri Dec 9 2011 15:33:54



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Module Documentation</b>	<b>5</b>
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	GPIO configuration . . . . .	8
3.4.1	Detailed Description . . . . .	8
3.5	Initialization flags . . . . .	9
3.5.1	Detailed Description . . . . .	9
3.5.2	Define Documentation . . . . .	9
3.5.2.1	INIT_CAN0 . . . . .	9
3.5.2.2	INIT_CAN1 . . . . .	9
3.5.2.3	INIT_GPI . . . . .	9
3.5.2.4	INIT_GPO . . . . .	9
3.5.2.5	INIT_LED . . . . .	10
3.5.2.6	INIT_MUSB . . . . .	10
3.5.2.7	INIT_SERIAL0 . . . . .	10
3.5.2.8	INIT_SERIAL1 . . . . .	10

3.5.2.9	INIT_SERIAL2	10
3.5.2.10	INIT_TIMER0	10
3.5.2.11	INIT_TIMER1	10
3.5.2.12	INIT_TIMER2	10
3.5.2.13	INIT_TIMER3	10
3.6	Status flags	11
3.6.1	Detailed Description	11
3.6.2	Define Documentation	11
3.6.2.1	MBED_CAN0_INIT	11
3.6.2.2	MBED_CAN1_INIT	11
3.6.2.3	MBED_GPI_INIT	11
3.6.2.4	MBED_GPO_INIT	11
3.6.2.5	MBED_LED_INIT	11
3.6.2.6	MBED_MUSB_INIT	12
3.6.2.7	MBED_SERIAL0_INIT	12
3.6.2.8	MBED_SERIAL1_INIT	12
3.6.2.9	MBED_SERIAL2_INIT	12
3.6.2.10	MBED_SYS_INIT	12
3.6.2.11	MBED_TIMER0_INIT	12
3.6.2.12	MBED_TIMER1_INIT	12
3.6.2.13	MBED_TIMER2_INIT	12
3.6.2.14	MBED_TIMER3_INIT	12
3.7	General purpose I/O pins	13
3.7.1	Detailed Description	13
3.7.2	Define Documentation	13
3.7.2.1	MBED_GPIO10	13
3.7.2.2	MBED_GPIO11	13
3.7.2.3	MBED_GPIO12	14
3.7.2.4	MBED_GPIO13	14
3.7.2.5	MBED_GPIO14	14
3.7.2.6	MBED_GPIO15	14
3.7.2.7	MBED_GPIO16	14
3.7.2.8	MBED_GPIO17	14
3.7.2.9	MBED_GPIO18	14

3.7.2.10	MBED_GPIO19	14
3.7.2.11	MBED_GPIO20	14
3.7.2.12	MBED_GPIO21	14
3.7.2.13	MBED_GPIO22	15
3.7.2.14	MBED_GPIO23	15
3.7.2.15	MBED_GPIO24	15
3.7.2.16	MBED_GPIO25	15
3.7.2.17	MBED_GPIO26	15
3.7.2.18	MBED_GPIO27	15
3.7.2.19	MBED_GPIO28	15
3.7.2.20	MBED_GPIO29	15
3.7.2.21	MBED_GPIO30	15
3.7.2.22	MBED_GPIO5	15
3.7.2.23	MBED_GPIO6	16
3.7.2.24	MBED_GPIO7	16
3.7.2.25	MBED_GPIO8	16
3.7.2.26	MBED_GPIO9	16
3.7.2.27	MBED_GPIOX	16
3.8	Port configuration	17
3.8.1	Detailed Description	17
3.8.2	Define Documentation	17
3.8.2.1	MBED_GPIO_P0	17
3.8.2.2	MBED_GPIO_P1	17
3.8.2.3	MBED_GPIO_P2	17
3.9	Device IDs	18
3.9.1	Detailed Description	18
3.9.2	Define Documentation	18
3.9.2.1	MBED_CAN0	18
3.9.2.2	MBED_CAN1	18
3.9.2.3	MBED_SERIAL0	18
3.9.2.4	MBED_SERIAL1	18
3.9.2.5	MBED_SERIAL2	18
3.9.2.6	MBED_TIMER0	18
3.9.2.7	MBED_TIMER1	19

3.9.2.8	MBED_TIMER2	19
3.9.2.9	MBED_TIMER3	19
3.10	CANopen objects	20
3.10.1	Detailed Description	20
3.10.2	Define Documentation	20
3.10.2.1	CAN_COB_EMCY	20
3.10.2.2	CAN_COB_ERROR_CONTROL	20
3.10.2.3	CAN_COB_NMT	20
3.10.2.4	CAN_COB_RPDO1	20
3.10.2.5	CAN_COB_RPDO2	21
3.10.2.6	CAN_COB_RPDO3	21
3.10.2.7	CAN_COB_RPDO4	21
3.10.2.8	CAN_COB_RSDO	21
3.10.2.9	CAN_COB_SYNC	21
3.10.2.10	CAN_COB_TIME	21
3.10.2.11	CAN_COB_TPDO1	21
3.10.2.12	CAN_COB_TPDO2	21
3.10.2.13	CAN_COB_TPDO3	21
3.10.2.14	CAN_COB_TPDO4	21
3.10.2.15	CAN_COB_TSDO	22
3.11	Network management commands	23
3.11.1	Detailed Description	23
3.11.2	Define Documentation	23
3.11.2.1	CAN_NMT_PREOP	23
3.11.2.2	CAN_NMT_RESET	23
3.11.2.3	CAN_NMT_RSCOMM	23
3.11.2.4	CAN_NMT_START	23
3.11.2.5	CAN_NMT_STOP	23
3.12	GPIO pin definitions	24
3.12.1	Detailed Description	24
3.12.2	Define Documentation	24
3.12.2.1	GPIO10	24
3.12.2.2	GPIO11	24
3.12.2.3	GPIO12	25

---

3.12.2.4	GPIO13	25
3.12.2.5	GPIO14	25
3.12.2.6	GPIO15	25
3.12.2.7	GPIO16	25
3.12.2.8	GPIO17	25
3.12.2.9	GPIO18	25
3.12.2.10	GPIO19	25
3.12.2.11	GPIO20	25
3.12.2.12	GPIO21	25
3.12.2.13	GPIO22	26
3.12.2.14	GPIO23	26
3.12.2.15	GPIO24	26
3.12.2.16	GPIO25	26
3.12.2.17	GPIO26	26
3.12.2.18	GPIO27	26
3.12.2.19	GPIO28	26
3.12.2.20	GPIO29	26
3.12.2.21	GPIO30	26
3.12.2.22	GPIO5	26
3.12.2.23	GPIO6	27
3.12.2.24	GPIO7	27
3.12.2.25	GPIO8	27
3.12.2.26	GPIO9	27
3.13	GPIO levels	28
3.13.1	Detailed Description	28
3.13.2	Define Documentation	28
3.13.2.1	GP_ERR	28
3.13.2.2	GP_HIGH	28
3.13.2.3	GP_LOW	28
<b>4</b>	<b>File Documentation</b>	<b>29</b>
4.1	mbed.h File Reference	29
4.1.1	Detailed Description	31
4.1.2	Define Documentation	31

---

4.1.2.1	MBED_H	31
4.1.3	Function Documentation	31
4.1.3.1	initSys	31
4.1.4	Variable Documentation	32
4.1.4.1	mbedStatus	32
4.2	mbed_can.h File Reference	32
4.2.1	Detailed Description	33
4.2.2	Define Documentation	33
4.2.2.1	CAN_BUFSIZE	33
4.2.2.2	MBED_CAN_H	33
4.2.3	Function Documentation	33
4.2.3.1	CAN_IRQHandler	33
4.2.3.2	CANopenRecv	34
4.2.3.3	CANopenSend	34
4.2.3.4	flushCANRxBuffer	35
4.2.3.5	flushCANTxBuffer	35
4.2.3.6	initCAN	35
4.2.3.7	setCANBaudrate	36
4.3	mbed_gpio.h File Reference	36
4.3.1	Detailed Description	37
4.3.2	Define Documentation	37
4.3.2.1	MBED_GPIO_H	37
4.3.3	Function Documentation	38
4.3.3.1	checkConflicts	38
4.3.3.2	GPIOReadVal	38
4.3.3.3	GPIOSetVal	38
4.3.3.4	initGPInputs	39
4.3.3.5	initGPIO	39
4.3.3.6	initGPOutputs	39
4.3.3.7	setGPInputCfg	40
4.3.3.8	setGPOutputCfg	40



# 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
GPIO configuration . . . . .	8
Initialization flags . . . . .	9
Status flags . . . . .	11
General purpose I/O pins . . . . .	13
Port configuration . . . . .	17
Device IDs . . . . .	18
CANopen objects . . . . .	20
Network management commands . . . . .	23
GPIO pin definitions . . . . .	24
GPIO levels . . . . .	28



# Chapter 2

## File Index

### 2.1 File List

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

<a href="#">mbed.h</a>	In this file are all device definitions for the mbed platform . . . . .	29
<a href="#">mbed_can.h</a>	Various definitions for the CANopen implementation . . . . .	32
<a href="#">mbed_gpio.h</a>	Various definition for General Purpose Input/Output (GPIO) . . . . .	36
<b>mbed_led.h</b>	. . . . .	??
<b>mbed_musb.h</b>	. . . . .	??
<b>mbed_serial.h</b>	. . . . .	??
<b>mbed_timer.h</b>	. . . . .	??



## 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 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)`

### 3.5.1 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`

DIP 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. [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.

## 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.*
- 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 initialized 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

<i>pin</i>	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\(\)](#)

#### 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

<i>pin</i>	Pin to be set, should be MBED_GPIOx.
<i>val</i>	Value to set, should be GP_LOW or GP_HIGH.

See also

[General purpose I/O pins](#)  
[GPIO levels](#)  
[GPIOReadVal\(\)](#)

#### 4.3.3.4 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\(\)](#)

#### 4.3.3.5 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

<i>mode</i>	Indicates what is to be initialized (inputs or outputs). Should be INIT_GPI or INIT_GPO.
<i>gpioConfig</i>	Pin configuration (see <a href="#">GPIO configuration</a> ).

See also

[setGPInputCfg\(\)](#)  
[setGPOutputCfg\(\)](#)  
[initGPInputs\(\)](#)  
[initGPOutputs\(\)](#)

#### 4.3.3.6 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\(\)](#)

#### 4.3.3.7 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

<i>config</i>	Pins to be configured as inputs, indicated by setting the corresponding bits.
---------------	---

See also

[General purpose I/O pins](#)

#### 4.3.3.8 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

<i>config</i>	Pins to be configured as outputs, indicating by setting the corresponding bits.
---------------	---

See also

[General purpose I/O pins](#)