

CANDrv

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	CANPAGEHandler Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Constructor & Destructor Documentation . . . . .	5
3.1.2.1	CANPAGEHandler() . . . . .	5
3.1.2.2	~CANPAGEHandler() . . . . .	6
3.2	mob_settings Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Member Data Documentation . . . . .	6
3.2.2.1	can_id . . . . .	6
3.2.2.2	can_msk . . . . .	7
3.2.2.3	data . . . . .	7
3.2.2.4	dlc . . . . .	7
3.2.2.5	ide . . . . .	7
3.3	MobConfigElement Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Member Data Documentation . . . . .	8
3.3.2.1	additionalData . . . . .	8
3.3.2.2	f . . . . .	8
3.3.2.3	ms . . . . .	8
3.3.2.4	op . . . . .	8
3.3.2.5	timestamp . . . . .	8

<b>4 File Documentation</b>	<b>9</b>
4.1 CANDrv.cpp File Reference	9
4.1.1 Detailed Description	10
4.1.2 Function Documentation	10
4.1.2.1 CANDrv_ClearAll_MOB()	10
4.1.2.2 CANDrv_FRMMan_Get_Msg() [1/2]	10
4.1.2.3 CANDrv_FRMMan_Get_Msg() [2/2]	10
4.1.2.4 CANDrv_FRMMan_Init()	11
4.1.2.5 CANDrv_FRMMan_Send_Msg()	12
4.1.2.6 CANDrv_Init()	12
4.1.2.7 CANDrv_Set_bt()	12
4.1.2.8 CLEAR_RXOK()	14
4.1.2.9 getMOBsetup()	14
4.1.2.10 ISR()	14
4.1.2.11 receiveData_generic()	14
4.1.2.12 receiveData_generic_restart()	15
4.1.2.13 setupMOB()	15
4.1.3 Variable Documentation	15
4.1.3.1 __internal_CAN_Config	15
4.2 CANDrv.h File Reference	15
4.2.1 Detailed Description	16
4.2.2 Enumeration Type Documentation	17
4.2.2.1 CanBaudrate	17
4.2.2.2 mob_operation	17
4.2.2.3 mob_purpose	17
4.2.3 Function Documentation	17
4.2.3.1 CANDrv_ClearAll_MOB()	17
4.2.3.2 CANDrv_FRMMan_Get_Msg()	18
4.2.3.3 CANDrv_FRMMan_Init()	18
4.2.3.4 CANDrv_FRMMan_Send_Msg()	19
4.2.3.5 CANDrv_Init()	19
4.2.3.6 getMOBsetup()	20
4.2.3.7 receiveData_generic()	20
4.2.3.8 receiveData_generic_restart()	20
4.2.3.9 setMOB_Operation()	20

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">CANPAGEHandler</a>	
Class for safe handling of the CANPAGE register . . . . .	5
<a href="#">mob_settings</a>	
Settings for a MOB can_id will be output if can_msk not fully set. can_msk is the bit masking for the acceptance Filter of the MOB The Member are described as Follows: RX TX DESCRIPTION . . . . .	6
<a href="#">MobConfigElement</a>	
Configuration object for the FRMMan Settings . . . . .	7



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">CANDrv.cpp</a>	
CANDrv cpp file . . . . .	9
<a href="#">CANDrv.h</a>	
CANDrv header File . . . . .	15





## Chapter 3

# Class Documentation

### 3.1 CANPAGEHandler Class Reference

Class for safe handling of the CANPAGE register.

```
#include <CANDrv.h>
```

#### Public Member Functions

- [CANPAGEHandler](#) (uint8\_t mob\_nr)  
*set CANPAGE to the supplied NR*
- [~CANPAGEHandler](#) ()  
*recovers the previously saved CANPAGE*

#### 3.1.1 Detailed Description

Class for safe handling of the CANPAGE register.

This class is initialized with the mob\_nr to switch to. The current CANPAGE is saved and will be restored after destruction of the context

#### 3.1.2 Constructor & Destructor Documentation

##### 3.1.2.1 CANPAGEHandler()

```
CANPAGEHandler::CANPAGEHandler (  
    uint8_t mob_nr )
```

set CANPAGE to the supplied NR

**Parameters**

<i>mob</i> ↔ _nr	the Number of the MOB to switch to
---------------------	------------------------------------

**3.1.2.2 ~CANPAGEHandler()**

```
CANPAGEHandler::~CANPAGEHandler ( )
```

recovers the previously saved CANPAGE

The documentation for this class was generated from the following files:

- [CANDrv.h](#)
- [CANDrv.cpp](#)

**3.2 mob\_settings Struct Reference**

Settings for a MOB can\_id will be output if can\_msk not fully set.  
can\_msk is the bit masking for the acceptance Filter of the MOB  
The Member are described as Follows:  
RX TX DESCRIPTION.

```
#include <CANDrv.h>
```

**Public Attributes**

- uint16\_t [can\\_id](#)
- uint16\_t [can\\_msk](#)
- uint8\_t [ide](#)
- uint8\_t [dlc](#)
- uint8\_t \* [data](#)

**3.2.1 Detailed Description**

Settings for a MOB can\_id will be output if can\_msk not fully set.  
can\_msk is the bit masking for the acceptance Filter of the MOB  
The Member are described as Follows:  
RX TX DESCRIPTION.

**3.2.2 Member Data Documentation****3.2.2.1 can\_id**

```
uint16_t mob_settings::can_id
```

IO IN ID of the CAN Message

#### 3.2.2.2 can\_msk

```
uint16_t mob_settings::can_msk
```

IN XX Mask of the CAN ID for RX

#### 3.2.2.3 data

```
uint8_t* mob_settings::data
```

OUT IN Pointer to the Data array

#### 3.2.2.4 dlc

```
uint8_t mob_settings::dlc
```

IO IN Data Length of the Message

#### 3.2.2.5 ide

```
uint8_t mob_settings::ide
```

IN IN Extended Message Format

The documentation for this struct was generated from the following file:

- [CANDrv.h](#)

## 3.3 MobConfigElement Struct Reference

configuration object for the FRMMan Settings

```
#include <CANDrv.h>
```

### Public Attributes

- [mob\\_purpose](#) op
- [mob\\_settings](#) ms
- [void\(\\* f\)\(uint8\\_t\)](#)
- [uint32\\_t](#) timestamp
- [void \\*](#) [additionalData](#)

### 3.3.1 Detailed Description

configuration object for the FRMMan Settings

### 3.3.2 Member Data Documentation

#### 3.3.2.1 `additionalData`

`void* MobConfigElement::additionalData`

additional Data for usage in an custom ISR

#### 3.3.2.2 `f`

`void (* MobConfigElement::f) (uint8_t)`

ISR routine

#### 3.3.2.3 `ms`

`mob_settings MobConfigElement::ms`

settings struct

#### 3.3.2.4 `op`

`mob_purpose MobConfigElement::op`

purpose of the MOB

#### 3.3.2.5 `timestamp`

`uint32_t MobConfigElement::timestamp`

OUT: timestamp of the Last Interrupt

The documentation for this struct was generated from the following file:

- [CANDrv.h](#)

## Chapter 4

# File Documentation

### 4.1 CANDrv.cpp File Reference

CANDrv.cpp file.

```
#include <avr/io.h>
#include "CANDrv.h"
#include "Arduino.h"
```

#### Functions

- `uint8_t CANDrv_Set_bt (CanBaudrate baudrate)`  
*set the Timing Parameters for the CAN.*
- `void CLEAR_RXOK ()`  
*Clears the interrupt FLAG.*
- `void CANDrv_ClearAll_MOB (void)`  
*advanced Function not needed for normal operation*
- `uint8_t CANDrv_Init (CanBaudrate baudrate)`  
*Initialize the CAN Driver.*
- `void getMOBsetup (mob_settings *ms)`  
*advanced Function not needed for normal operation*
- `void setupMOB (mob_settings *ms)`  
*advanced Function not needed for normal operation*
- `void receiveData_generic (uint8_t mob_NR)`  
*generic Receive function for an Interrupt Driven MOB.*
- `void receiveData_generic_restart (uint8_t mob_NR)`  
*generic Receive function for an Interrupt Driven MOB.*
- `uint8_t CANDrv_FRMMan_Init (MobConfigElement *CAN_Config)`  
*Initialize the FRMMan.*
- `uint8_t CANDrv_FRMMan_Send_Msg (uint8_t index)`  
*sends a configured Message*
- `uint8_t CANDrv_FRMMan_Get_Msg (uint8_t index, mob_settings *ms, uint8_t iteration=0)`  
*get a received Can Message.*
- `uint8_t CANDrv_FRMMan_Get_Msg (uint8_t index, mob_settings *ms)`  
*get a received Can Message.*
- `ISR (CAN_INT_vect)`

## Variables

- [MobConfigElement](#) \* [\\_\\_internal\\_CAN\\_Config](#)

### 4.1.1 Detailed Description

CANDrv.cpp file.

This File contains the Implementation of the CANDrv and the FRMMan

### 4.1.2 Function Documentation

#### 4.1.2.1 CANDrv\_ClearAll\_MOB()

```
CANDrv_ClearAll_MOB (
    void )
```

advanced Function not needed for normal operation

This function clear all MOB registers. This is used in INI.

#### 4.1.2.2 CANDrv\_FRMMan\_Get\_Msg() [1/2]

```
CANDrv_FRMMan_Get_Msg (
    uint8_t index,
    mob_settings * ms,
    uint8_t iteration = 0 )
```

get a received Can Message.

This function reads a received Can Message from the internal Buffer.  
It also retries to read if there were any Changes, because of an Interrupt, receiving a new Message.  
In this case the read operation is retried.

!ONLY USED INTERNALLY!

#### Parameters

<i>index</i>	the index of the mob to read
<i>ms</i>	OUTPUT Pointer to a message settings object, where the data is written.
<i>iteration</i>	Helper argument to limit the number of tries

#### Returns

1 on success. 0 in case of an error.

#### 4.1.2.3 CANDrv\_FRMMan\_Get\_Msg() [2/2]

```
CANDrv_FRMMan_Get_Msg (
```

```
uint8_t index,
mob_settings * ms )
```

get a received Can Message.

This function reads a received Can Message from the internal Buffer.

It also retries to read if there were any Changes, because of an Interrupt, receiving a new Message.

In this case the read operation is retried.

#### Parameters

<i>index</i>	the index of the mob to read
<i>ms</i>	OUTPUT Pointer to a message settings object, where the data is written.

#### Returns

1 on success. 0 in case of an error.

#### 4.1.2.4 CANDrv\_FRMMan\_Init()

```
CANDrv_FRMMan_Init (
    MobConfigElement * CAN_Config )
```

Initialize the FRMMan.

This function initializes the FRMMan with the given CAN\_Config.

Example configuration:

```
//Make sure all these variables are global an not in a scoped context, like a funtion.
uint8_t data_130[8];
uint8_t data_7e8[8];
uint8_t data_7e0[8];
MobConfigElement CAN_Config[] =
{
    {TX_DATA_SW_DRIVEN, {0x7e0, 0x000, 0, 8, (uint8_t*)&data_7e0}}, //This Messsage can be sent
    with CANDrv_FRMMan_Send_Msg(0);
    {RX_DATA_INTERRUPT_DRIVEN, {0x130, 0x3FF, 0, 8, (uint8_t*)&data_130}, &
    receiveData_generic_restart, 0, 0}, //This Message will be received with an
    Interrupt. It will be directly enabled again
    {RX_DATA_INTERRUPT_DRIVEN, {0x7e8, 0x3FF, 0, 8, (uint8_t*)&data_7e8}, &
    receiveData_generic, 0, 0}, //This Message will be received with an Interrupt. It will be
    only received once!
    {UNUSED}, //unused MOBs
    {UNUSED},
    {UNUSED}
};

void setup() {
    data_7e0[0] = 0x02; //Initialize the Message to send.
    data_7e0[1] = 0x01;
    data_7e0[2] = 0x05;
    data_7e0[3] = 0x33;
    data_7e0[4] = 0x44;
    data_7e0[5] = 0x55;
    data_7e0[6] = 0x66;
    data_7e0[7] = 0x77;
    CANDrv_Init(CAN_500k); //init CANDrv
    CANDrv_FRMMan_Init(CAN_Config); //init FRMMan
    sei(); //enable interrupts. This is neccessary for the Interrupt driven receives
}
```

**Parameters**

<i>CAN_Config</i>	The CAN_Config to use. For examples see funtion description.
-------------------	--

**Returns**

1 on success. 0 in case of an error.

**4.1.2.5 CANDrv\_FRMMan\_Send\_Msg()**

```
CANDrv_FRMMan_Send_Msg (  
    uint8_t index )
```

sends a configured Message

This function sends a predefined Message.

**Parameters**

<i>index</i>	the index of the Message to send
--------------	----------------------------------

**Returns**

1 on success. 0 in case of an error.

**4.1.2.6 CANDrv\_Init()**

```
uint8_t CANDrv_Init (  
    CanBaudrate baudrate )
```

Initialize the CAN Driver.

**Parameters**

<i>baudrate</i>	the baudrate to set
-----------------	---------------------

**Returns**

1 on success. 0 in case of an error.

**4.1.2.7 CANDrv\_Set\_bt()**

```
uint8_t CANDrv_Set_bt (  
    CanBaudrate baudrate ) [inline]
```

set the Timing Parameters for the CAN.



the Values of the Registers CANBT1-3 are also defined in this file.

!ONLY USED INTERNALLY!

**Parameters**

<i>baudrate</i>	the baudrate to set.
-----------------	----------------------

**Returns**

1 on success. 0 in case of an error.

**4.1.2.8 CLEAR\_RXOK()**

```
void CLEAR_RXOK ( ) [inline]
```

Clears the interrupt FLAG.

**4.1.2.9 getMOBsetup()**

```
void getMOBsetup (
    mob_settings * ms )
```

advanced Function not needed for normal operation

This function reads the Settings from the currently selected MOB. To select a MOB see [CANPAGEHandler](#)

**See also**

[CANPAGEHandler](#)

**Parameters**

<i>ms</i>	pointer to the MOB settings
-----------	-----------------------------

**4.1.2.10 ISR()**

```
ISR (
    CAN_INT_vect )
```

**4.1.2.11 receiveData\_generic()**

```
receiveData_generic (
    uint8_t mob_NR )
```

generic Receive funtion for an Interrupt Driven MOB.

This function only receives this Message once!

## Parameters

<i>mob_NR</i>	the MOB Number given by the ISR
---------------	---------------------------------

## 4.1.2.12 receiveData\_generic\_restart()

```
receiveData_generic_restart (
    uint8_t mob_NR )
```

generic Receive funtion for an Interrupt Driven MOB.

This function will activate the Interrupt again.

## Parameters

<i>mob_NR</i>	the MOB Number given by the ISR
---------------	---------------------------------

## 4.1.2.13 setupMOB()

```
setupMOB (
    mob_settings * ms )
```

advanced Function not needed for normal operation

This function writes the given Settings to the currently selected MOB. To select a MOB see [CANPAGEHandler](#)

## See also

[CANPAGEHandler](#)

## Parameters

<i>ms</i>	pointer to the MOB settings
-----------	-----------------------------

## 4.1.3 Variable Documentation

## 4.1.3.1 \_\_internal\_CAN\_Config

```
MobConfigElement* __internal_CAN_Config
```

## 4.2 CANDrv.h File Reference

CANDrv header File.

## Classes

- struct [mob\\_settings](#)  
*Settings for a MOB can\_id will be output if can\_msk not fully set.  
can\_msk is the bit masking for the acceptance Filter of the MOB  
The Member are described as Follows:  
RX TX DESCRIPTION.*
- struct [MobConfigElement](#)  
*configuration object for the FRMMan Settings*
- class [CANPAGEHandler](#)  
*Class for safe handling of the CANPAGE register.*

## Enumerations

- enum [CanBaudrate](#) { [CAN\\_500k](#), [CAN\\_800k](#) }  
*Baudrate Enumerator for the Baudrate.*
- enum [mob\\_operation](#) { [TX\\_DATA](#) = 0x01, [RX\\_DATA](#) = 0x02, [DISABLED](#) = 0x00 }  
*MOB Operation Enumerator for the operation while manually using a MOB.*
- enum [mob\\_purpose](#) {  
[TX\\_DATA\\_SW\\_DRIVEN](#) = 0x10, [RX\\_DATA\\_SW\\_DRIVEN](#) = 0x20, [MULTIPURPOSE](#) = 0x30, [RX\\_DATA\\_INTERRUPT\\_DRIVEN](#) = 0x62,  
[RX\\_DATA\\_INTERRUPT\\_DRIVEN\\_INACTIVE](#) = 0x60, [TX\\_DATA\\_INTERRUPT\\_ACTIVE](#) = 0x50, [UNUSED](#) = 0x00 }  
*MOB Purpose Enumerator for the Purpose of the MOB when configuring the FRMMan.*

## Functions

- [uint8\\_t CANDrv\\_Init](#) ([CanBaudrate](#) baudrate)  
*Initialize the CAN Driver.*
- [uint8\\_t CANDrv\\_FRMMan\\_Get\\_Msg](#) ([uint8\\_t](#) index, [mob\\_settings](#) \*ms)  
*get a received Can Message.*
- [uint8\\_t CANDrv\\_FRMMan\\_Send\\_Msg](#) ([uint8\\_t](#) index)  
*sends a configured Message*
- [uint8\\_t CANDrv\\_FRMMan\\_Init](#) ([MobConfigElement](#) \*CAN\_Config)  
*Initialize the FRMMan.*
- void [receiveData\\_generic](#) ([uint8\\_t](#) mob\_NR)  
*generic Receive funtion for an Interrupt Driven MOB.*
- void [receiveData\\_generic\\_restart](#) ([uint8\\_t](#) mob\_NR)  
*generic Receive funtion for an Interrupt Driven MOB.*
- void [CANDrv\\_ClearAll\\_MOB](#) (void)  
*advanced Function not needed for normal operation*
- void [getMOBsetup](#) ([mob\\_settings](#) \*ms)  
*advanced Function not needed for normal operation*
- void [setMOB\\_Operation](#) ([mob\\_operation](#) mo)  
*set the Operation of the current MOB*

### 4.2.1 Detailed Description

CANDrv header File.

This File describes the Interface of the CANDrv and the FRMMan

## 4.2.2 Enumeration Type Documentation

### 4.2.2.1 CanBaudrate

enum `CanBaudrate`

Baudrate Enumerator for the Baudrate.

Enumerator

CAN_500k	500 Kbaud
CAN_800k	800 Kbaud

### 4.2.2.2 mob\_operation

enum `mob_operation`

MOB Operation Enumerator for the operation while manually using a MOB.

Enumerator

TX_DATA	Transmit DATA
RX_DATA	Receive DATA
DISABLED	DISABLED

### 4.2.2.3 mob\_purpose

enum `mob_purpose`

MOB Purpose Enumerator for the Purpose of the MOB when configuring the FRMMan.

Enumerator

TX_DATA_SW_DRIVEN	TX triggered by SW
RX_DATA_SW_DRIVEN	RX triggered by SW
MULTIPURPOSE	RX and TX triggered by SW
RX_DATA_INTERRUPT_DRIVEN	RX triggered by Interrupt. Automatically activated at Startup
RX_DATA_INTERRUPT_DRIVEN_INACTIVE	RX triggered by Interrupt. Automatically deactivated at Startup
TX_DATA_INTERRUPT_ACTIVE	TX triggered by SW. Will cause an Interrupt on completion
UNUSED	MOB is unused

## 4.2.3 Function Documentation

### 4.2.3.1 CANDrv\_ClearAll\_MOB()

```
void CANDrv_ClearAll_MOB (
    void )
```

advanced Function not needed for normal operation

This function clear all MOB registers. This is used in INI.

#### 4.2.3.2 CANDrv\_FRMMan\_Get\_Msg()

```
uint8_t CANDrv_FRMMan_Get_Msg (
    uint8_t index,
    mob_settings * ms )
```

get a received Can Message.

This function reads a received Can Message from the internal Buffer.  
It also retries to read if there were any Changes, because of an Interrupt, receiving a new Message.  
In this case the read operation is retried.

##### Parameters

<i>index</i>	the index of the mob to read
<i>ms</i>	OUTPUT Pointer to a message settings object, where the data is written.

##### Returns

1 on success. 0 in case of an error.

#### 4.2.3.3 CANDrv\_FRMMan\_Init()

```
uint8_t CANDrv_FRMMan_Init (
    MobConfigElement * CAN_Config )
```

Initialize the FRMMan.

This function initializes the FRMMan with the given CAN\_Config.

Example configuration:

```
//Make sure all these variables are global an not in a scoped context, like a funtion.
uint8_t data_130[8];
uint8_t data_7e8[8];
uint8_t data_7e0[8];
MobConfigElement CAN_Config[] =
{
    {TX_DATA_SW_DRIVEN, {0x7e0, 0x000, 0, 8, (uint8_t*)&data_7e0}}, //This Messsage can be sent
    with CANDrv_FRMMan_Send_Msg(0);
    {RX_DATA_INTERRUPT_DRIVEN, {0x130, 0x3FF, 0, 8, (uint8_t*)&data_130}, &
    receiveData_generic_restart, 0, 0}, //This Message will be received with an
    Interrupt. It will be directly enabled again
    {RX_DATA_INTERRUPT_DRIVEN, {0x7e8, 0x3FF, 0, 8, (uint8_t*)&data_7e8}, &
    receiveData_generic, 0, 0}, //This Message will be received with an Interrupt. It will be
    only received once!
    {UNUSED}, //unused MOBs
    {UNUSED},
    {UNUSED}
};

void setup() {
    data_7e0[0] = 0x02; //Initialize the Message to send.
    data_7e0[1] = 0x01;
```

```

data_7e0[2] = 0x05;
data_7e0[3] = 0x33;
data_7e0[4] = 0x44;
data_7e0[5] = 0x55;
data_7e0[6] = 0x66;
data_7e0[7] = 0x77;
CANDrv_Init(CAN_500k); //init CANDrv
CANDrv_FRMMan_Init(CAN_Config); //init FRMMan
sei(); //enable interrupts. This is neccessary for the Interrupt driven receives
}

```

**Parameters**

<i>CAN_Config</i>	The CAN_Config to use. For examples see funtion description.
-------------------	--

**Returns**

1 on success. 0 in case of an error.

**4.2.3.4 CANDrv\_FRMMan\_Send\_Msg()**

```

uint8_t CANDrv_FRMMan_Send_Msg (
    uint8_t index )

```

sends a configured Message

This function sends a predefined Message.

**Parameters**

<i>index</i>	the index of the Message to send
--------------	----------------------------------

**Returns**

1 on success. 0 in case of an error.

**4.2.3.5 CANDrv\_Init()**

```

uint8_t CANDrv_Init (
    CanBaudrate baudrate )

```

Initialize the CAN Driver.

**Parameters**

<i>baudrate</i>	the baudrate to set
-----------------	---------------------

**Returns**

1 on success. 0 in case of an error.

#### 4.2.3.6 getMOBsetup()

```
void getMOBsetup (
    mob_settings * ms )
```

advanced Function not needed for normal operation

This function reads the Settings from the currently selected MOB. To select a MOB see [CANPAGEHandler](#)

See also

[CANPAGEHandler](#)

##### Parameters

<i>ms</i>	pointer to the MOB settings
-----------	-----------------------------

#### 4.2.3.7 receiveData\_generic()

```
void receiveData_generic (
    uint8_t mob_NR )
```

generic Receive funtion for an Interrupt Driven MOB.

This function only receives this Message once!

##### Parameters

<i>mob_NR</i>	the MOB Number given by the ISR
---------------	---------------------------------

#### 4.2.3.8 receiveData\_generic\_restart()

```
void receiveData_generic_restart (
    uint8_t mob_NR )
```

generic Receive funtion for an Interrupt Driven MOB.

This function will activate the Interrupt again.

##### Parameters

<i>mob_NR</i>	the MOB Number given by the ISR
---------------	---------------------------------

#### 4.2.3.9 setMOB\_Operation()

```
setMOB_Operation (
    mob_operation mo ) [inline]
```



set the Operation of the current MOB

This function sets the Operation of the currently selected MOB To select a MOB see [CANPAGEHandler](#)

See also

[CANPAGEHandler](#)

#### Parameters

<i>mo</i>	the operation mode to set the MOB to
-----------	--------------------------------------



# Index

- \_\_internal\_CAN\_Config
    - CANDrv.cpp, [15](#)
  - ~CANPAGEHandler
    - CANPAGEHandler, [6](#)
- additionalData
  - MobConfigElement, [8](#)
- CANDrv.cpp, [9](#)
  - \_\_internal\_CAN\_Config, [15](#)
  - CANDrv\_ClearAll\_MOB, [10](#)
  - CANDrv\_FRMMan\_Get\_Msg, [10](#)
  - CANDrv\_FRMMan\_Init, [11](#)
  - CANDrv\_FRMMan\_Send\_Msg, [12](#)
  - CANDrv\_Init, [12](#)
  - CANDrv\_Set\_bt, [12](#)
  - CLEAR\_RXOK, [14](#)
  - getMOBsetup, [14](#)
  - ISR, [14](#)
  - receiveData\_generic, [14](#)
  - receiveData\_generic\_restart, [15](#)
  - setupMOB, [15](#)
- CANDrv.h, [15](#)
  - CANDrv\_ClearAll\_MOB, [17](#)
  - CANDrv\_FRMMan\_Get\_Msg, [18](#)
  - CANDrv\_FRMMan\_Init, [18](#)
  - CANDrv\_FRMMan\_Send\_Msg, [19](#)
  - CANDrv\_Init, [19](#)
  - CanBaudrate, [17](#)
  - getMOBsetup, [19](#)
  - mob\_operation, [17](#)
  - mob\_purpose, [17](#)
  - receiveData\_generic, [20](#)
  - receiveData\_generic\_restart, [20](#)
  - setMOB\_Operation, [20](#)
- CANDrv\_ClearAll\_MOB
  - CANDrv.cpp, [10](#)
  - CANDrv.h, [17](#)
- CANDrv\_FRMMan\_Get\_Msg
  - CANDrv.cpp, [10](#)
  - CANDrv.h, [18](#)
- CANDrv\_FRMMan\_Init
  - CANDrv.cpp, [11](#)
  - CANDrv.h, [18](#)
- CANDrv\_FRMMan\_Send\_Msg
  - CANDrv.cpp, [12](#)
  - CANDrv.h, [19](#)
- CANDrv\_Init
  - CANDrv.cpp, [12](#)
  - CANDrv.h, [19](#)
- CANDrv\_Set\_bt
  - CANDrv.cpp, [12](#)
- CANPAGEHandler, [5](#)
  - ~CANPAGEHandler, [6](#)
  - CANPAGEHandler, [5](#)
- CLEAR\_RXOK
  - CANDrv.cpp, [14](#)
- can\_id
  - mob\_settings, [6](#)
- can\_msk
  - mob\_settings, [6](#)
- CanBaudrate
  - CANDrv.h, [17](#)
- data
  - mob\_settings, [7](#)
- dlc
  - mob\_settings, [7](#)
- f
  - MobConfigElement, [8](#)
- getMOBsetup
  - CANDrv.cpp, [14](#)
  - CANDrv.h, [19](#)
- ISR
  - CANDrv.cpp, [14](#)
- ide
  - mob\_settings, [7](#)
- mob\_operation
  - CANDrv.h, [17](#)
- mob\_purpose
  - CANDrv.h, [17](#)
- mob\_settings, [6](#)
  - can\_id, [6](#)
  - can\_msk, [6](#)
  - data, [7](#)
  - dlc, [7](#)
  - ide, [7](#)
- MobConfigElement, [7](#)
  - additionalData, [8](#)
  - f, [8](#)
  - ms, [8](#)
  - op, [8](#)
  - timestamp, [8](#)
- ms
  - MobConfigElement, [8](#)
- op

- MobConfigElement, [8](#)
- receiveData\_generic
  - CANDrv.cpp, [14](#)
  - CANDrv.h, [20](#)
- receiveData\_generic\_restart
  - CANDrv.cpp, [15](#)
  - CANDrv.h, [20](#)
- setMOB\_Operation
  - CANDrv.h, [20](#)
- setupMOB
  - CANDrv.cpp, [15](#)
- timestamp
  - MobConfigElement, [8](#)