# Notes on CAN bus

## Preliminary list of message IDs and masks

#define ID\_brakes 3

#define ID\_dashboard 6

#define ID\_steeringWheel 15

#define ID\_power 12

#define ID\_hallSensor 9 //checked

#define ID\_lightsFront 81 //not really in use

#define ID\_lightsBack 91 //not really in use

#define MASK\_MOTOR\_MODULE 1 //checked

#define MASK\_BACK\_LIGHTS 2 //checked

#define MASK\_FRONT\_MODULE 4 //checked

#define MASK\_FRONT\_LIGHTS 4 //checked

## #define MASK\_GPS 12//checked

## Pitfalls

* Beware of *sending the message itself* and not it’s pointer.
* Remember to *set global interrupts* before going to sleep when receiving messages.
* Remember to *clear global interrupts* before transmitting a message.
* If the data received is 0, *check the length* of the message sent.

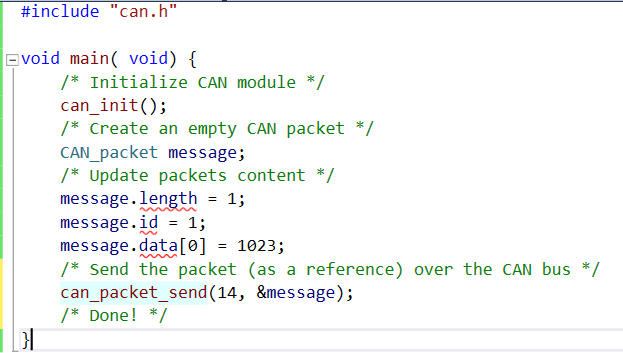
## prepare\_rx(char mob, unsigned id, unsigned idmask, CAN\_cbf callback)

Parameter *mob* (message object) is used to denote which CAN bus buffer to use when looking for the message particular message. Parameter *id* is used to identify the messages on the bus that are of interest for the particular module. The range is typically 0..2048. The complete list of IDs used is in the “can.h” header file. Parameter *idmask* is used to filter unwanted messages allowing only a single or a range of messages to pass. See example below. Parameter *callback* is a pointer to callback function when a message with “allowed” id is found.

Example of message filtering:

* *id* = 0b1100, *idmask* = 0b1000. Allowed message ids: 0b1xxx. That is 0b1000, 0b1001, 0b1010, 0b1011, 0b1100, 0b1101, 0b1110, and 0b1111. In decimal: 8..15.
* *id* = 0b1100, *idmask* = 0b1111. Allowed message ids: 0b1100. In decimal: 12.
* *Id* = 0b1100, *idmask* = 0. Allowed message ids: all.

## How to send a CAN message



## How to receive a CAN message

