AT90CAN128/64/32

CAN Library

AVR GCC Platforms:

WinAVR-20060421

or

WinAVR-20070122

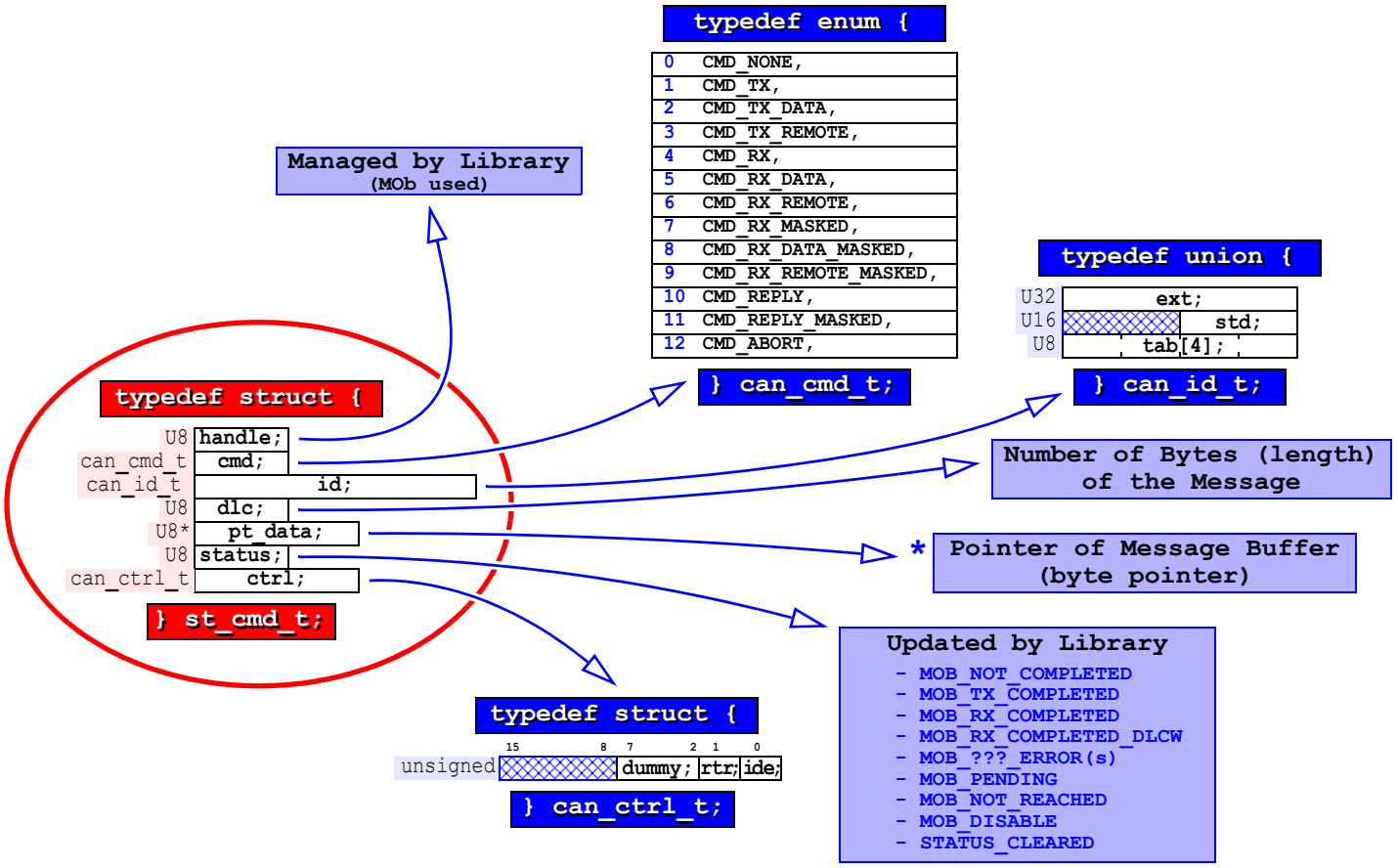
CAN Library

1 - THREE MAIN ELEMENTS (c.f. “can_lib.h”)

1. **“*st_cmd_t*”** structure (CAN message descriptor),
2. **“*U8 can_cmd (st_cmd_t *)*”** function (to perform a command -action- on the CAN bus),
3. **“*U8 can_get_status (st_cmd_t *)*”** function (to know if the command is performed or not).

1.1 - “*st_cmd_t*” Structure

Figure 1. CAN Message Descriptor Structure



1.2 - “*U8 can_cmd (st_cmd_t *)*” Function

- This function needs as argument a pointer on a structure of ***“st_cmd_t”*** type, a CAN message descriptor.
- This function looks for a MOb free and configures it with data given by the CAN message descriptor.
- This function updates the status of the CAN message descriptor and returns the rating of its action:
  -> CAN_CMD_ACCEPTED or **CAN_CMD_REFUSED**.

1.3 - "*U8 can_get_status (st_cmd_t *)*" Function

- This function needs as argument a pointer on a structure of ***"st_cmf_t"*** type, a CAN message descriptor.
- With the ***"handle"*** information of the CAN message descriptor, this function returns a status about current action of the MOb:
  -> CAN_STATUS_COMPLETED, **CAN_STATUS_NOT_COMPLETED** or **CAN_STATUS_ERROR**.

2. Exemple of Using

2.1 - Simple

```
//.......
U8 buffer[];
st_cmd_t message;
//.......
message.pt_data = &buffer[0];
message.cmd = CMD_RX;
while(can_cmd(&message) != CAN_CMD_ACCEPTED);
while(can_get_status(&message) == CAN_STATUS_NOT_COMPLETED);
//.......
```

2.2 - Full

```
//.......
U8 c_status;
U8 buffer[];
st_cmd_t message;
//.......
message.pt_data = &buffer[0];
message.cmd = CMD_RX;
//.......
if(can_cmd(&message) == CAN_CMD_REFUSED)
{
    //.......
}
else
{
    c_status = can_get_status(&message);
    switch (c_status)
    {
        case CAN_STATUS_COMPLETED:
            //.......
            break;
        case CAN_STATUS_NOT_COMPLETED:
            //.......
            break;
        case CAN_STATUS_ERROR:
            //.......
            break;
        default:
            //.......
        break;
    }
}
```

3 - OTHER FUNCTION(S)

3.1 - "*U8 can_init (U8 mode)*" Function

- This function performs a full initialization of the CAN controller. This function also accepts an automatic recognition of the Baud Rate on the CAN bus (c.f. ***"#define CAN_BAUDRATE ..."*** in ***"config.h"*** file).
- The first time in the program, this function performs needs ***"0x00"*** as argument. If the automatic recognition of the Baud Rate is used and some fails are detected, new calling(s) to this function may be necessary (the function will try other values than those already tested), it will need "0x01" as argument.
- Especially when automatic recognition of the Baud Rate is used, this function returns ***"0x00"*** when **none** Baud Rate has been found, else ***"0x01"*** is returned.

Example:

```
void main (void)
{
U8 temp;

temp = 0;
//.......
while(1)
{
    temp = can_init(temp);
    if (temp != 0)
    {
        //.......
        your_program(); // If too many errors, output from "your_program()"
    }
    else
    {
        printf("No synchro, exit.\r\n"); break;
    }
}
while(1); // Error loop
```

Notes: