

Software plan

```

xk32
├── xk32_clock.h
├── xk32_error.h
├── xk32_gpio.h
├── xk32_module.h
├── xk32_task.h
├── xk32_term.h
└── xk32_timer.h
src
├── drv
├── atsame70q20
│   ├── afec.c
│   ├── nvic.c
│   ├── pio.c
│   ├── pmc.c
│   └── nwm.c

```

```

    }
    s_term_msg_list_out[TERM_MSG_ERRBUF].next = &s_term_msg_list_out;
    s_term_msg_out.msg_first = &s_term_msg_list_out;
    s_term_msg_out.msg_last = &s_term_msg_list_out[0x01u];
    s_term_msg_out.msg_first->msg = "";
    s_term_msg_out.b_overflow = NO;

    pself->is_init = bTRUE;
}

void term_run(xk32_term *pself){
    if(pself->is_init == bTRUE){

        /*init when freq_in is ok*/
        pself->is_driver_init = uart_is_init();
        if(pself->is_driver_init == bFALSE){

```

```

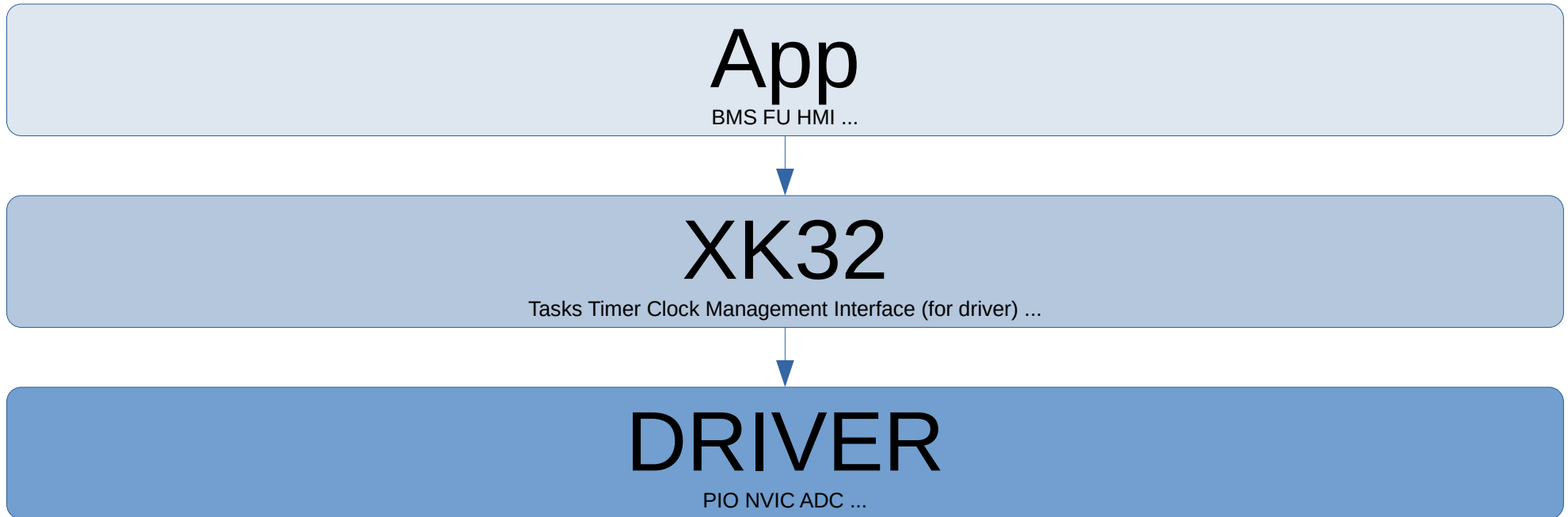
typedef void (*task_function)(void)

typedef struct xk32_task_function{
    uint8_t registered;
    uint8_t enable;
    uint16_t unique_id;
    char *name;
    uint8_t is_running;
    task_function pfct;
    void *pPara;
}xk32_task_function;

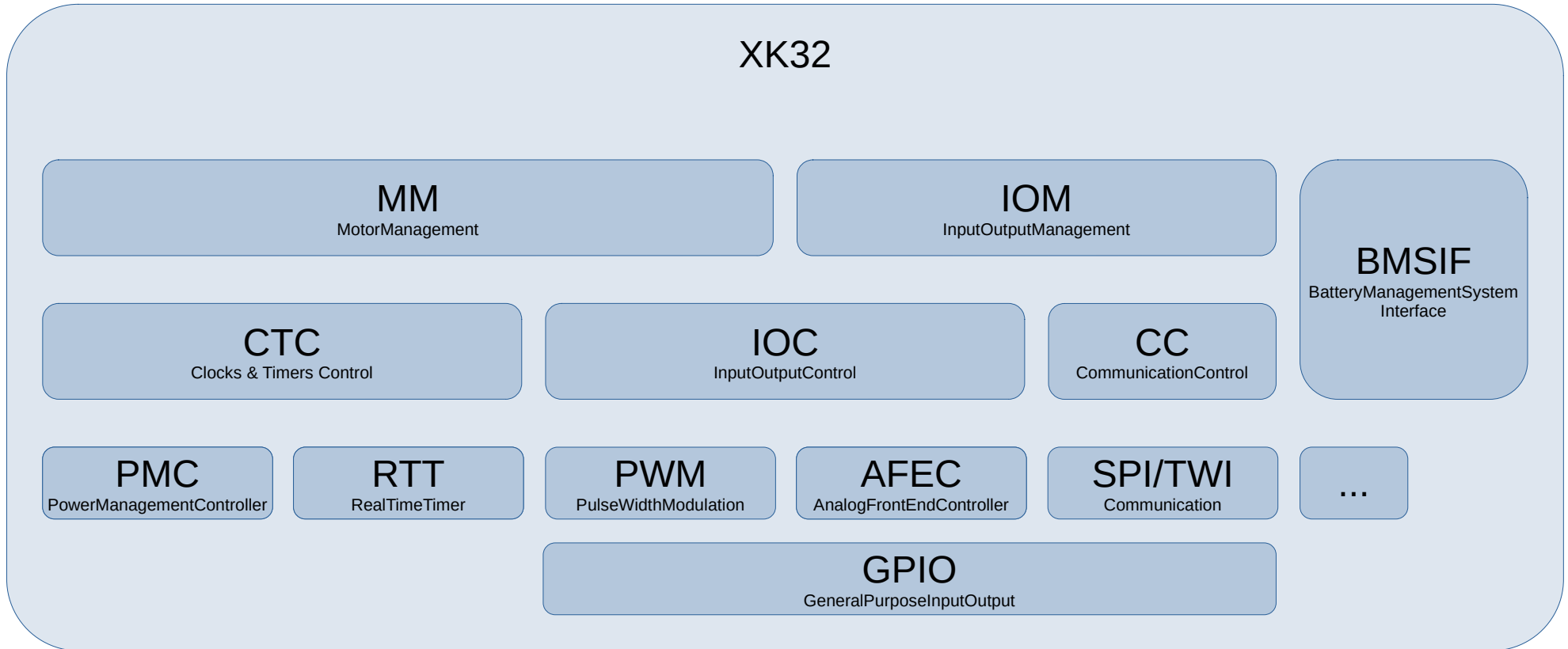
typedef struct xk32_task{
    uint16_t ctr;
    uint32_t interval_ms;
    uint32_t is_interval_over;

```

General Software Structure



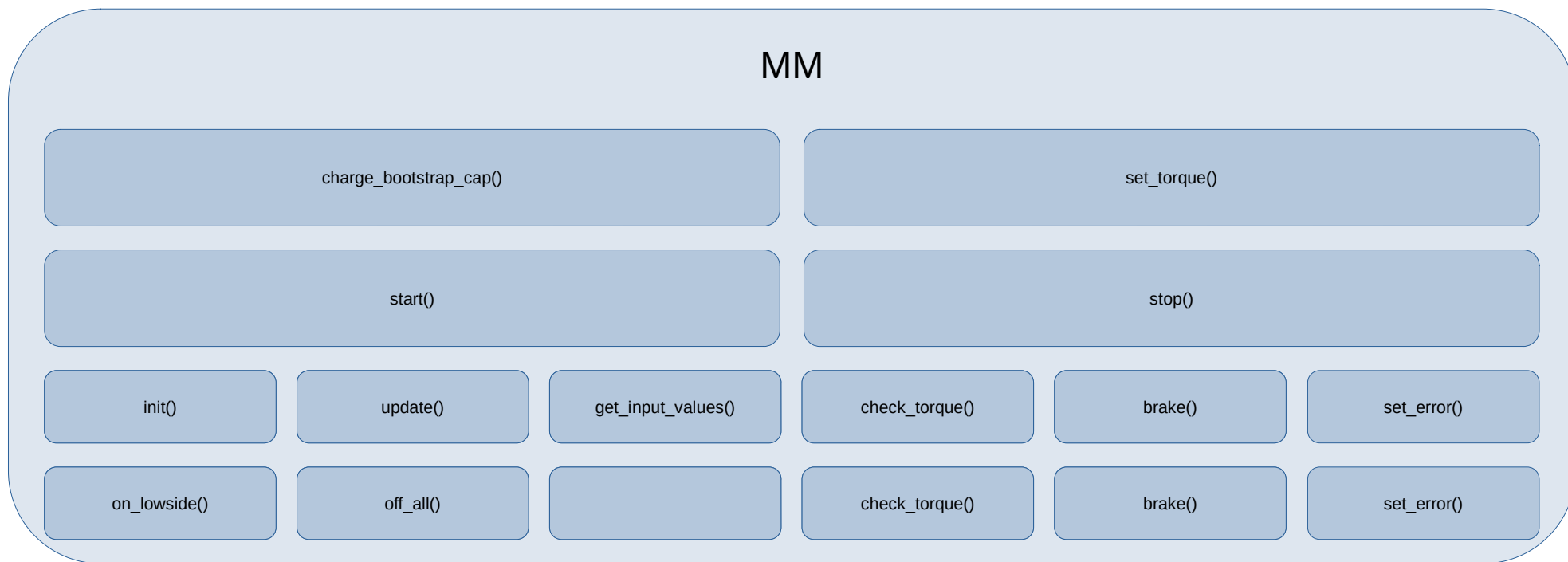
XK32 Module



Software design rule

- Every module has at least one struct where all default values are defined for its module
- Module specific struct is the first argument in every function
- Module can have more different structs
- Kind of object orientated programming

MotorManagement MM



InputOutputController IOC

IOC

set_io()

get_io()

change_io()

check_io()

init()

update()

get_input_values()

set_output_values()

set_periph_io()

set_error()

XK32 Standard Module

Public functions of a standard module

```
module_init(struct module *pself);
```

```
module_run(struct module *pself);
```

```
module_subinit(struct module *pself);
```

```
ret module_set/get(struct module *pself);
```

...

XK32 Standard Driver

Public functions of a standard driver

```
driver_init(struct module *pself);
```

```
driver_run(struct module *pself);
```

```
driver_subinit(struct module *pself);
```

```
ret driver_set/get(struct module *pself);
```

```
driver_error_msg(struct module *pself);
```

...

PMC Driver Functions

PMC

```
pmc_init(struct module *pself);
```

```
pmc_run(struct module *pself);
```

```
pmc_main_clk_select(struct module *pself);
```

```
ret pmc_main_clk_freq(struct module *pself);
```

...

TODO

