

Драйвер CAN BUS для актуатора на STM32

Драйвер зроблено за класичною схемою в одному потоці RunLoop одного процесу з циклічними буферами протокола Can

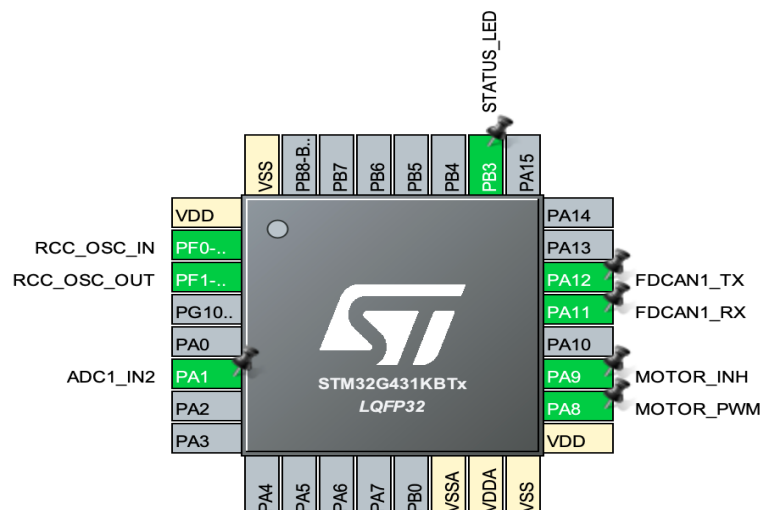
Драйвер складається з бізнеслогіки, драйверів мережі, двигуна та налаштувань MCU

Дісклеймер:

Принципова схема зображена не якісно, деякі назви пінів не зчиталися, тому мають бути перевірені та регенеровані в CubeMX за необхідністю. Задіяна конфігурація вказана на малюнку нижче.

З міркувань безпеки під час першого включення не рекомендовано під'єднувати навантаження у вигляді реально двигуна із-за можливих помилок при формування таймерів сигналу ШИМ. Має бути перевірено за допомогою осцилографа для виявлення помилок в налаштування таймеру.

Програмний комплекс не перевірявся на реальному процесорі, тільки в емуляторі ReNode



Як запустити:

Розробка велася в MacOS M1 у VCODE, тому на інших машинах треба буде перегенерувати проект під поточне IDE та OS у CubeMX та збілдити в IDE

Опис каталогів та модулів

КАТАЛОГИ:

- /

can_driver_g4.ioc	Налаштування CubeMX
CMake*.*	Файли для зборки проекта

- cmake – каталог зборщика CMake

- _docs

T3 на драйвер.md	
gpio.png	Налаштування GPIO
readme.pdf	Цей документ

- _emu

can_driver_g4.ioc	Налаштування CubeMX
emulation_res.txt	Результати емуляції

- Core/Inc/Src

automate.*	Код та заголовки пакувальника, распаковщика пакетів Автоматично згенерований
can.*	Драйвер шини CAN
common.*	Допоміжний код та константи
controller.*	Бізнеслогіка
left_break.*	Драйвер мотору та потенціометру
main.*	Ініціалізація та Runloop редагувати з обережністю, може перезаписуватися
stm32g4xx_hal_conf.*	Бібліотеки HAL під проект
stm32g4xx_it.*	

- Drivers – драйвери для проекту