Программирование микроконтроллеров STM32

Режим пониженного энергопотребления

МФТИ, 2019

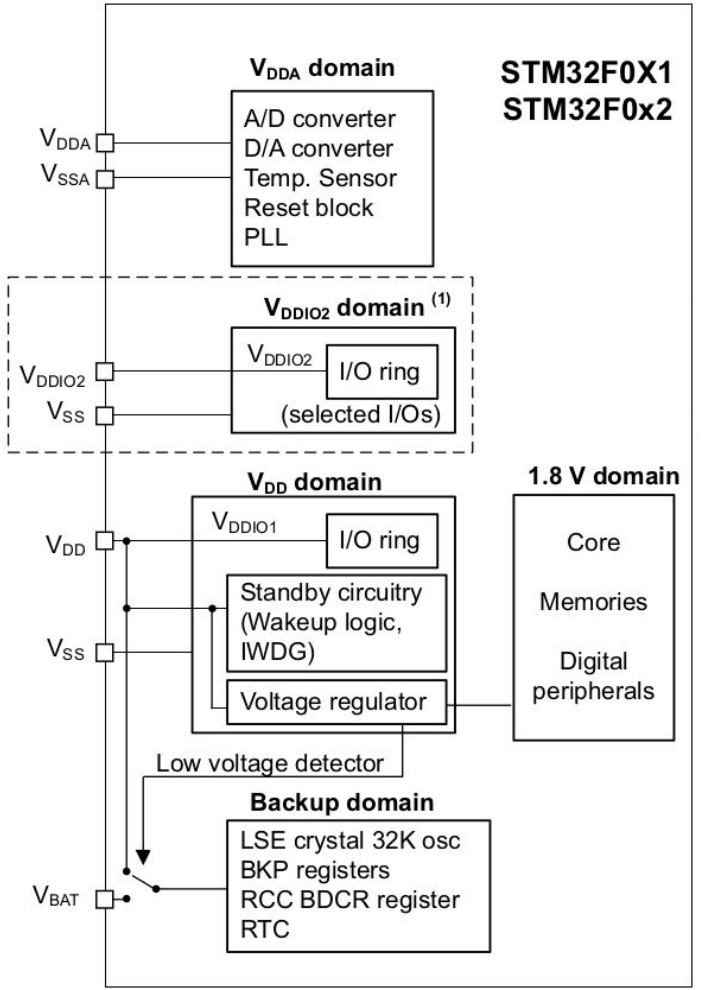
Режим пониженного энергопотребления

3 режима пониженного энергопотребления:

- Sleep mode (отключено только тактирование ядра)
- Stop mode (вся периферия в VCORE выключена, PLL, HSI, HSI14 и HSE остановлены, контент SRAM и регистров сохранен)
- Standby mode (1.8 периферия полностью обесточена, регулятор питания и осцилляторы выключены, контент SRAM и регистров не сохранен, кроме Backup

Помимо этого:

- Потребление может быть снижено за счет понижения частоты тактирования и полного отключения шин
- Необходимо переводить неиспользуемые порты ввода-вывода в режим аналогового входа перед переходом в режим пониженного потребления (для low и stop режимов)



Регулятор питания. Режимы работы

- Run mode - питание 1.8В полностью включено
- Stop mode - режим пониженного энергопотребления (для сохранения контента SRAM и регистров)
- Standby mode - регулятор полностью выключен

PDDS и **LPDS** биты в **PWR_CR** устанавливаются с помощью LL_PWR_SetPowerMode (mode):

- LL_PWR_MODE_STOP_MAINREGU
- LL_PWR_MODE_STOP_LPREGU
- LL_PWR_MODE_STANDBY

Пониженное энергопотребление. Вход и выход

Режим	Подготовка	Вход	Выход
Sleep	LL_LPM_EnableSleep() LL_LPM_EnableSleepOnExit [Sleep-now/Sleep-on-exit], LL_LPM_EnableEventOnPend() [для включения поддержки wakeup событий]	__WFI()	Любое прерывание
		__WFE()	Любое событие
Stop	LL_LPM_EnableDeepSleep() LL_PWR_SetPowerMode() Все ожидающие прерывания должны быть сброшены, Тактирование лучше на внутреннее	__WFI() & __WFE()	EXTI прерывание или событие, Прерывание периферии в режиме wakeup
Standby	LL_LPM_EnableDeepSleep(), LL_PWR_SetPowerMode(), LL_PWR_ClearFlag_WU()	__WFI() & __WFE()	WKUP, будильник, сброс (reset), IWDG

Репозиторий

https://github.com/edosedgar/stm32f0_ARM