

DWIN T5L* архитектура/структура

Чем отличаются ядра OS, GUI, DWINOS, T5L51.bin?

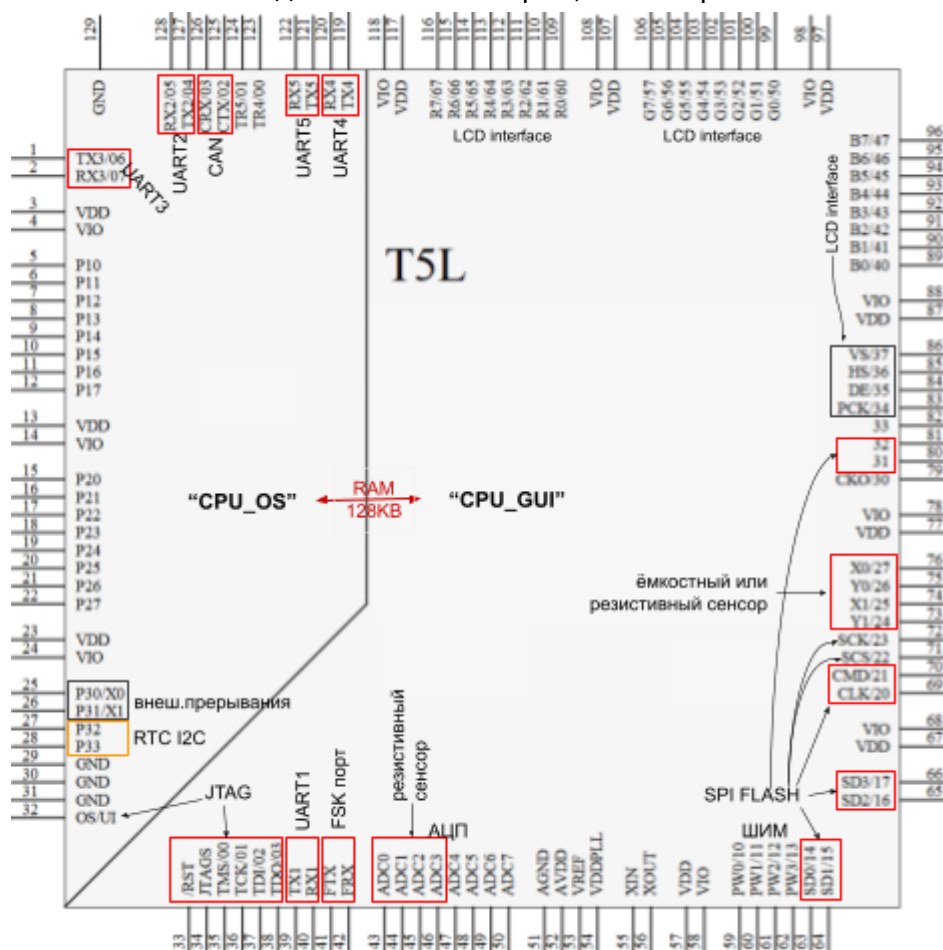
Вспомним устройство микроконтроллера серии **T5L*** вкратце,

Два ядра на базе архитектуры 8051(да-да, это не опечатка) работающие на частотах до 200-250 МГц(и это тоже).

Ядро на схеме справа(**"CPU_GUI"**) имеет UART1, выходы для подключения SPI флеш памяти и множество выводов для работы с LCD матрицей и сенсором/тачскрином, в общем предназначено для работы с GUI(обработкой графических данных и их отображения).

Ядро на схеме слева(**"CPU_OS"**) имеет множество портов ввода-вывода, CAN и 4 шт UART(2, 3, 4, 5). Предназначено для коммуникации контроллера с внешним миром: работой с внешним микроконтроллером, или для подключения интерфейсных чипов RS-485, RS-232 и пр.

Между собой ядра общаются через общую RAM/оперативную память размером 256 КБ, ~~однако в DGUS2 используется только 128 КБ~~. В DGUS к данной памяти обращаются через VP и SP.



В ядре **"CPU_GUI"** загружается закрытая прошивка DWIN **T5L_UI*.BIN**. Прошивка считывает с внешней флеш памяти различный контент, в основном это изображения, в RAM памяти настраивает части изображений и выводит итоговый массив на LCD(ЖК матрицу).

~~Если это режим DGUS(а не TA), то именно она реализует все функции отрисовки.~~

Стандартными средствами разработки данную прошивку нельзя как-то модифицировать, дополнять, изменять. С ней можно только взаимодействовать через 256 КБ RAM памяти. По этой-же причине недоступна **прямая** работа с UART1, SPI, ШИМ-ом и АЦП.

Однако к данной периферии можно получить доступ общаясь с ядром через память, но это имеет ряд ограничений, в первую очередь временных.

В ядре “CPU_OS” может быть любой код: например пользовательский код написанный на Си в Keil, скомпилированный файл имеет название **T5L51*.BIN(C51)**.

Так как лишь малая часть пользователей готовы к программированию, компания DWIN предоставляет свою прошивку для данного ядра: **T5L_OS_DGUS2_V*.BIN**(также не имеет исходного кода - закрытая).

Данная прошивка отвечает за параметры работы портов UART с 2 по 5(в т.ч. настройки RS-485, CAN и пр.), а также настройки параметров modbus rtu и других функций.

Прошивка **T5L_OS_DGUS2_V*.BIN** при необходимости поддерживает загрузку подпрограмм которые имеют название **DWINOS*.BIN**

Подпрограмма DWIN OS(файл **DWINOS*.BIN**) создаётся

либо в результате компиляции ассемблерного кода в программе *DWIN OSBuild 3.0.2.6*

либо в результате компиляции Си кода в программе *DWIN C Compiler 1.1.0.7*

При использовании языка Си в компиляторе *DWIN C Compiler* не нужно прописывать заголовочные файлы. Библиотеки необходимые для общего языка Си - компилятор импортирует сам, а также собственные библиотеки DWIN: 1Lab, 2Lab. Если вы используете язык Си для программирования, вам необходимо загрузить эти две библиотеки на экран с помощью SD-карты, а затем вы можете использовать последовательный порт 2 для прямой загрузки подпрограммы.

Типичный **пример** подпрограммы **DWINOS*.BIN** - файл который конфигурирует параметры работы modbus: *DWINOS_DWIN_T5L_115200__UART5_ModBus.bin*

Пользовательский код T5L51*.BIN(C51) (из Keil C под процессор 8051) для ядра “CPU_OS”.

При загрузке пользовательского кода **T5L51*.BIN(C51)** удаляются прошивки **T5L_OS_DGUS2_V*.BIN** и **DWINOS*.BIN**

Соответственно реализацию работы UART2, modbus rtu и др. пользователю нужно проводить через Си код. Программирование кода позволяет пользователю:

- заложить в дисплей бизнес-логику приложения,
- отказаться от внешнего контроллера,
- использовать порты ввода-вывода например подключив дублирующие кнопки или реле,
- использовать АЦП и ШИМ преобразователи
- и др.

Однако это требует высокую квалификацию пользователя и дополнительные человеко часы для изучения платформы и написания кода.

~~Для эмуляции необходимо установить интерфейсный драйвер Keil для DWIN T5L, но качество эмуляции “оставляет желать лучшего”.~~

Примечание: шестнадцатеричный файл, скомпилированный программой Keil, необходимо преобразовать в файл T5L51*.BIN(C51), например с помощью программы *srec_cat.exe* или *DWIN DownLoadFor8051 V1.4*.

Подытожим:

В ядро “CPU_GUI” загружается неизменная прошивка **T5L_UI*.BIN**

В ядро “CPU_OS” загружается:

либо T5L51*.BIN(C51) - код пользователя, который выполняется непосредственно на 8051, без платформы DWIN OS.

либо T5L_OS_DGUS2_V*.BIN - код DWIN, он может иметь подпрограмму DWIN OS

DWIN OS(**DWINOS*.BIN**) - это код, который выполняется на виртуальной машине OS T5L DWIN.

Нужно выбрать: C51 или OS. Они не могут существовать одновременно.

При загрузке T5L51*.BIN в микросхему T5L, он удалит ядро T5L_OS и DWINOS.

Чтобы восстановить заводскую программу T5L_OS, вы должны сначала загрузить T5L_OS, например T5L_OS_DGUS2_V*.BIN, а затем загрузить DWINOS*.BIN, при необходимости.

CPU_OS отвечает за связь UART2~UART5.

Тип файла	Название	Описание
Программный файл	T5L_UI_*.BIN T5L_OS_*.BIN	Базовые прошивки ядер от DWIN
DWIN OS	DWINOS*.BIN	Программа DWIN OS, код должен начинаться с 0x1000. Логически, наиболее верное название - <i>подпрограмма конфигурирования OS</i> . Например файл <i>DWINOS_DWIN_T5L_115200__UART5_ModBus.bin</i> конфигурирует параметры работы modbus.
OS CPU 8051	T5L51*.BIN	Приложения, разработанные пользователями для платформы 8051.

Информация с кит.форума не вошедшая в статью в формате вопросы и ответы:

В: Можно ли запрограммировать UART1 программированием C51(keil C)?

О: UART1 управляется основным процессором графического интерфейса пользователя и пользователи не могут переписать его через C51. Вы можете использовать другой последовательный порт.

Обратите внимание, что скорость передачи данных для других последовательных портов, за исключением UART 2(настраивается в .CFG файле), должна быть настроена в C51(keil C).

В: Связь между графическим интерфейсом T5L и ОС только через регистр переменных 128K?

О: Да, два ядра работают независимо и могут совместно использовать 128 килобайт RAM памяти(регистры переменных) для взаимодействия с данными.

Ядро GUI не может быть настроено/изменено и использовано, можно понять, что система DWIN закрыта, но через ядро OS можно самостоятельно взаимодействовать с ядром GUI.

В: Программа Tetris представляет собой программу 51 одночипового микрокомпьютера, которая написана на Keil C. Если ее скопировать, ядро исходной сборки ОС будет удалено, поэтому протокол DGUS больше не будет поддерживаться в это время.

О: Чтобы восстановить скомпилированную программу ОС, сначала нужно загрузить программы ядра T5L, такие как T5L_OS_DGUS2_V10.BIN и T5L_UI_DGUS2_V30.BIN, а затем загрузить DWINOS*.BIN. Способ загрузки - положить ядро в папку DWIN_SET отдельно, а затем использовать SD-карту, чтобы скопировать его в экран.

В: Может ли система OS напрямую обрабатывать сенсорную информацию экрана?

О: Может, есть соответствующие системные переменные операции, состояние сенсорного экрана и координаты могут быть обнаружены.

В: Каково назначение использование 50 мс? Снижение энергопотребления? (есть конкретные данные?)

О: Это просто изменяет частоту обновления пользовательского интерфейса чипа T5L.

*речь про T5L_UI_DGUS2_V41_50mSDGUS cycle_20210713

В/О: Если экран черный и не может прочитать карту, вам нужно сначала стереть программу записи, а затем записать базовую программу. Обычное обновление не требует стирания, просто используйте SD-карту для загрузки последней версии ядра.

В/О: вам не нужно использовать изображения PNG, просто используйте изображения BMP. С изображениями BMP не должно быть проблем. PNG может не работать с некоторыми программами.

*речь про корректное создание .ICL файла.

В: Подскажите, пожалуйста, как сделать обновление CAN онлайн

Продукт требует, чтобы SD-карта не могла быть обновлена, и я хочу отправить команды CAN через главный компьютер для обновления. Что я должен делать?

О: Риск обновления CAN относительно высок. В принципе, его следует использовать с осторожностью, потому что .icl библиотеки изображения, библиотека текстовых шрифтов могут составлять более десяти МБ, а время загрузки велико, в то время как SD-карта является параллельной загрузкой SPI, расстояние до MCU короткое, а скорость загрузки более чем в 40 раз выше, чем у CAN.

Когда объем загружаемых данных относительно велик, обновление CAN может быть только 8 байт каждый раз. Во время процесса обновления может быть потеря кадров, поэтому риск обновления относительно высок. Большой, и программа относительно сложная. Как только экран дисплея включает слишком много кода, сложность дизайна программа будет увеличена. Кроме того, логику нелегко контролировать, и легко потерять данные. Или непредсказуемые ошибки, приводящие к аномальным явлениям.

Обслуживание непростое, поэтому мы рекомендуем клиентам использовать SD-карту для загрузки, которая является стабильной, надежной и быстрой. Если SD-карта действительно неудобна, вы можете попросить компанию DWIN, чтобы настроить оборудование и спроектировать SD-карту скрытым образом и использовать клеммную колодку для ее вывода. Вам необходимо загрузить и подключить SD-карту к небольшой плате, что может быть лучше.

CAN рекомендуется только для связи, а не для проектов модернизации. Рекомендуется в первый раз загрузить большой файловый шрифт с SD-карты, а часто используемые файлы дисплея (небольшие файлы, такие как файлы дисплея 13 14 и сенсорные файлы) можно обновить онлайн с помощью программного обеспечения для настройки.