

DWIN T5L* архитектура/структура

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

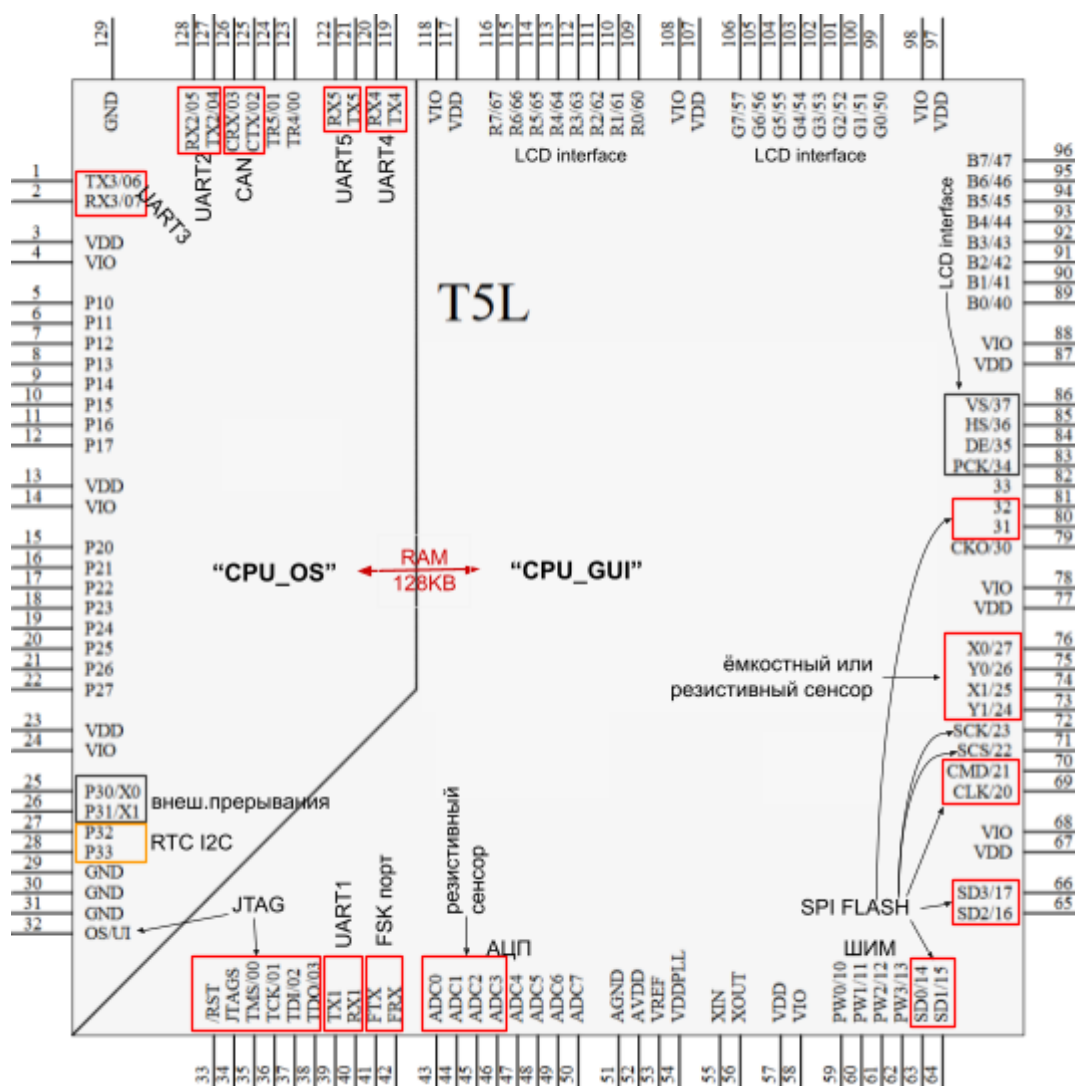
Вспомним устройство микроконтроллера серии **T5L*** вкратце,

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, выводы 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. Адреса 0x0000-0xFFFF с ячейками по 2 байта/1 слово. Взаимодействие между ядрами происходит передачей команд в области памяти 0x0000-0x1000, остальная область памяти свободна для пользователя. В DGUS к данной памяти обращаются через переменные VP и SP.



В ядре “CPU_GUI” загружается закрытая прошивка DWIN **T5L_UI_*.BIN**. Прошивка считывает с внешней SPI флеш памяти различный контент(изображения(jpeg), шрифты, звуки и пр.) затем в собственной(не общей) RAM памяти настраивает части изображений и выводит итоговый массив на LCD(ЖК матрицу). У этого ядра есть аппаратный JPEG декодер.

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

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

В ядре “CPU_OS” может быть любой код: например пользовательский код написанный на Си-8051 в Keil(это сторонний компилятор) или на ассемблере(8051), 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_*.BIN**(также закрытая и не имеет исходного кода).

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_*.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*

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_*.BIN** и **DWINOS*.BIN**. Соответственно реализацию работы UART2, modbus rtu и др. пользователю нужно программировать в Си код самостоятельно.

Программирование кода позволяет пользователю:

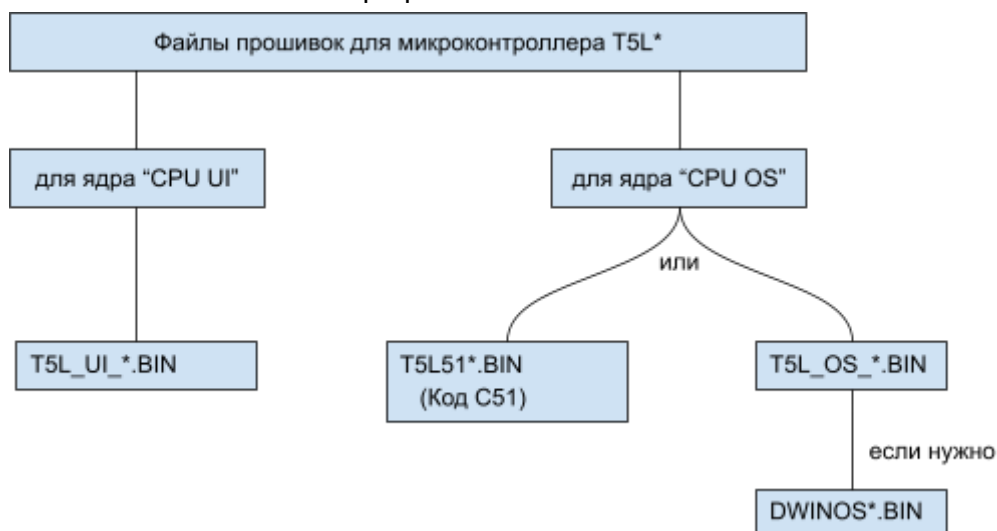
- заложить в дисплей бизнес-логику приложения,
- отказаться от внешнего контроллера,
- использовать порты ввода-вывода например подключив дублирующие кнопки или реле,
- использовать АЦП и ШИМ преобразователи
- и др.

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.~~

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*.

Графическое пояснение.



Подытожим:

В ядро “CPU_GUI” загружается неизменяемая 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_*.BIN - код DWIN, он может иметь подпрограмму DWIN OS

DWIN OS(**DWINOS*.BIN**) - это код, который выполняется в T5L_OS_*.BIN.

Нужно выбрать: C51(свой код) или OS(код DWIN). Они не могут существовать одновременно.

Взаимодействие происходит через общую RAM память и команды в диапазоне адресов 0x0000-0x1000.

При загрузке файла T5L51*.BIN(свой код) в микросхему T5L, он удалит код T5L_OS и DWINOS.

Чтобы восстановить заводскую программу T5L_OS, вы должны сначала загрузить T5L_OS, например T5L_OS_*.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	Программный код DWIN OS, код должен начинаться с адреса 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	Программный код, разработанный пользователями для платформы 8051.

Память

Внешняя SPI Flash

Дисплеи для хранения медиа данных(шрифты, jpeg изображения, звуки) используют внешнюю SPI флеш память.

У большинства дисплеев устанавливается микросхема spi nor flash 16 МБ, однако есть дисплеи с 8МБ.

У некоторых моделей дисплеев на плате можно встретить площадки под установку микросхем дополнительной памяти(SOP-8 pin to pin). То есть размер памяти можно расширить впаяв микросхемы самостоятельно или при заказе партии. На практике необходимость в этом встречается очень редко так как 16 МБ и даже 8 МБ памяти достаточно при грамотном проектировании приложения.

Важно понимать принцип блочной структуры хранения данных во внешней памяти.

Программа UI считывает данные из SPI Flash памяти блоками по 256 КБ(это виртуальное деление).

То есть независимо от того какой установлен суммарный размер памяти: 8 МБ, 16 МБ, 24 МБ или др.

Память всегда делится на блоки по 256 КБ. Порядковый номер каждого блока в англо. документации именуется ID.

При 16 МБ нам доступно $16 \cdot 1024 / 256 = 64$ блока(ID: 0 - 63).

При 8 МБ нам доступно $8 \cdot 1024 / 256 = 32$ блока(ID: 0 - 31).

Если допаяна память, например, до 48 МБ, то нам доступно $48 \cdot 1024 / 256 = 192$ блока(ID: 0 - 191).

Очень важно уместить все файлы в данную область и не допустить наложения одних блоков на другие.

При загрузке/прошивке в флеш память попадают файлы которые имеют в начале своего наименования цифру - индекс или ID нужного блока.

Локальный диск (D:) > DWIN.PRO > DWIN_SET

Имя	Тип	Размер
0_DWIN_ASC.HZK	Файл "HZK"	3 011 КБ
12_HZK.BIN	Файл "BIN"	173 КБ
13_TouchFile.bin	Файл "BIN"	5 КБ
14_ShowFile.bin	Файл "BIN"	22 КБ
22_Config.bin	Файл "BIN"	129 КБ
24_GBK24_宋体.HZK	Файл "HZK"	1 684 КБ
32.icl	Библиотека знач...	1 960 КБ
50.icl	Библиотека знач...	266 КБ
54.icl	Библиотека знач...	16 КБ
56.icl	Библиотека знач...	32 КБ
58.icl	Библиотека знач...	43 КБ
60.icl	Библиотека знач...	329 КБ

Важно не допускать наложения файлов в памяти так как при загрузке крайний загруженный файл стирает блоки предыдущего.

Для этого нужно проводить простой расчёт.

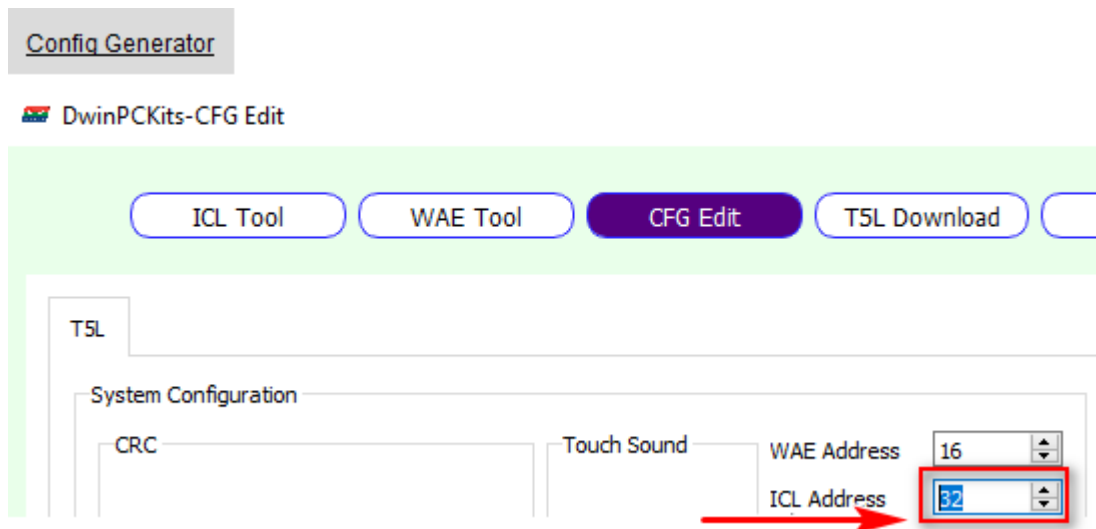
Например посчитаем максимальный размер самого популярного файла: 32.icl.

Следующий за ним файл помещается в блок с ID 50. Следовательно: $50 - 32 = 18 \cdot 256 \text{ КБ} = 4608 \text{ КБ}$ - максимальный размер для файла 32.icl, в данном случае его размер 1960 КБ - в порядке.

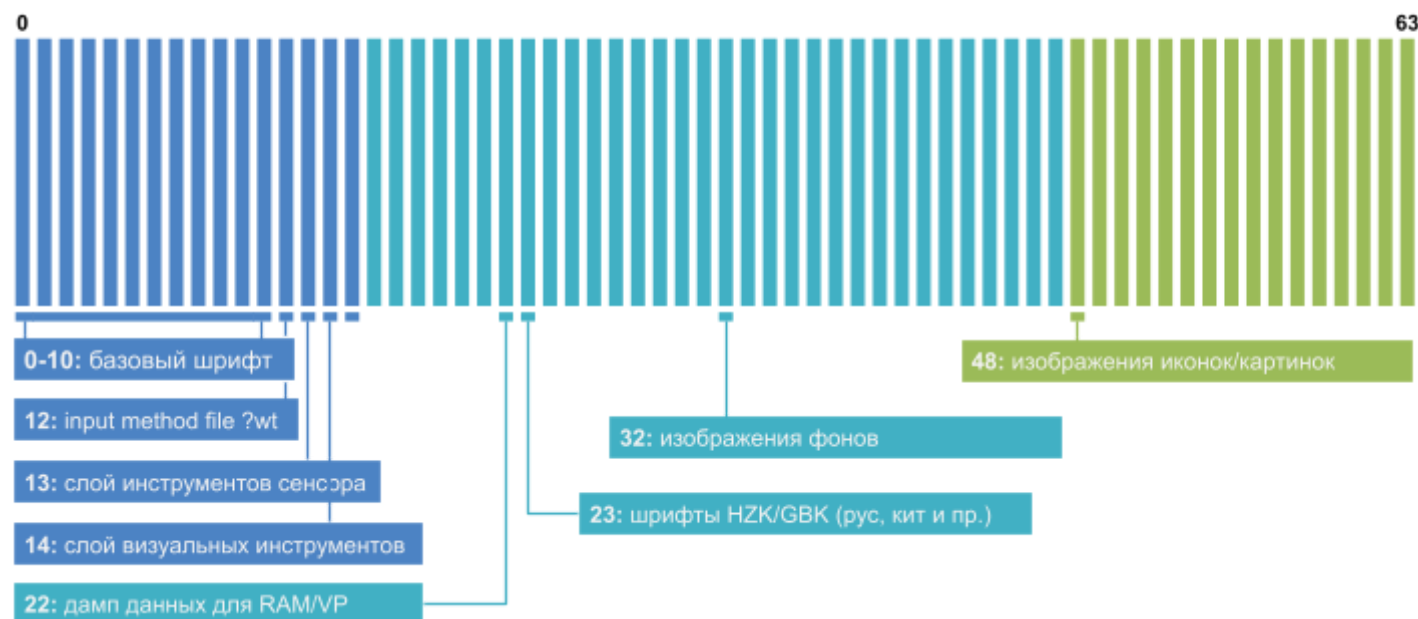
Если вдруг размер превышен, то нужно **либо** сдвинуть следующий за ним файл вправо. То есть 50 переименовать например в 53(+3) и остальные также(+3).

Либо сам файл 32.icl сдвинуть влево(если есть куда) и переименовать например в 23.icl(если бы не было файла 24.hzk).

При перемещении 32.icl на новое место обязательно зафиксируйте это в *T5LCFG.CFG*
В противном случае при старте экрана вы увидите чёрный экран вместо вашей картинки.



От DWIN есть рекомендация размещать файлы в следующем порядке.
Однако жёстко определены сектора только для файлов 12, 13, 14 и 22.
Остальные файлы можно размещать по собственному усмотрению.



Общая RAM

Два ядра микроконтроллера T5L* взаимодействуют между собой через общую RAM/оперативную память размером 256 КБ. В DGUS используется только 128 КБ. Адресация ячеек 0x0 - 0xFFFF, размер ячейки 2 байта(1 слово). В DGUS к данной памяти обращаются через переменные VP и SP. Ранее это описано.

Адресация общей RAM памяти(внутри программы)

! Могут быть неточности.

Адресация общей памяти находится на 16 битной шине и в DGUS используются адреса 0x00:0000-0x00:7FFF.

Например, две переменных DGUS II, 0x1000 и 0x1001, соответствуют адресу 0x0800 памяти переменных OS 8051 DGUS, где D3 соответствует старшему байту переменной 0x1000, а D2 соответствует младшему байту переменной 0x1000; D1 соответствует переменному старшему байту 0x1001, а D0 соответствует переменному младшему байту 0x1001.

(2) Адреса 0x00:8000-0x00:FFFF в настоящее время не используются системой DGUS II и могут использоваться пользователями.

(3) 0xFF:0060-0xFF:006B адрес, конфигурация и буфер приемопередатчика интерфейса CAN. Доступ к памяти переменных DGUS использует интерфейс регистров SFR в следующих случаях:

Всего 256 КБайт, но разложены в ячейки по 2 байта(1 слово). Так что имеем диапазон адресов 0-0x3FFFF

Адресация памяти

~~Машинное слово~~ в одном ядре 16 бит, в другом 32 бита

Ядро CPU_UI организовано в соответствии с 32-битным адресом, а ядро CPU_OS имеет 8 битную адресацию.

две памяти переменных DGUS II,

0x1000 и 0x1001,

соответствуют адресу 0x0800 памяти переменных OS 8051 DGUS, где

D3 соответствует старшему байту переменной 0x1000, а **D2** соответствует младшему байту переменной 0x1000;

D1 соответствует переменному старшему байту 0x1001, а **D0** соответствует переменному младшему байту 0x1001.

(2) Адреса 0x00:8000-0x00:FFFF в настоящее время не используются системой DGUS II и могут быть определены пользователями как переменная память.

От автора:

Текст основан на личном понимании архитектуры микроконтроллера T5L и приложения DGUS основываясь на доступной технической документации и сообщениях с [официального китайского форума DWIN](#).

Если у читателя есть большее понимание - прошу рассказать мне об этом:

tg [@stoumk](#) или mail mir@dwin.pro

История изменений

вер	дата	изменения	автор
1	ноябрь 2022	создание документа	Мирослав Кириллин
2	08.10.2023	правки, корректировки формулировок	^

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 коде 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 по CAN шине

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.