

Техническое описание платы сопряжения **ТФТ-Компаньон**
2D-ускоритель для TFT с интерфейсом SPI
Версия 1, вариант 4

СОДЕРЖАНИЕ

1	Назначение	3
2	Описание работы	3
3	Соединители.....	4
4	Характеристики	5
5	Связь с пользовательской системой	6
6	Особенности работы DVI-версии.....	7
7	Система команд.....	8
7.1	Формат пакета	9
7.2	Формат 16-битного цвета пиксела	9
7.3	Хранение данных во флэш-памяти	10
7.4	Код 00h. Чтение статуса/данных	10
7.5	Код A5h. Сброс процессора	10
7.6	Код 01h. Установка видимой видеостраницы	11
7.7	Код 02h. Установка активной видеостраницы	11
7.8	Код 03h. Поточная запись видеостраницы	11
7.9	Код 05h. Поточное чтение видеостраницы	12
7.10	Код 04h. Установка яркости подсветки	12
7.11	Код 06h. Чтение яркости подсветки	12
7.12	Код 0Ah. Очистка экрана	12
7.13	Код 0Bh. Чтение разрешения экрана.....	13
7.14	Код 0Ch. Установка границ отсечки	13
7.15	Код 10h. Вывод пиксела	13
7.16	Код 11h. Чтение цвета пиксела	13
7.17	Код 12h. Вывод прямой сплошной линии	14
7.18	Код 13h. Вывод прямой пунктирной линии	14
7.19	Код 15h. Вывод прямоугольника.....	14
7.20	Код 16h. Вывод заштрихованного прямоугольника.....	15
7.21	Код 17h. Вывод эллипса	15
7.22	Код 18h. Вывод заштрихованного эллипса	15
7.23	Код 19h. Штриховка произвольной области	15
7.24	Код 20h. Копирование прямоугольной области	16
7.25	Код 21h. Копирование области в буфер обмена	16
7.26	Код 22h. Вставка области из буфера обмена	16
7.27	Код 30h. Вывод изображения из флэш-памяти.....	16
7.28	Код 31h. Запись изображения во флэш-память.....	17
7.29	Код 32h. Удаление изображения из флэш-памяти.....	17
7.30	Код 33h. Прием изображения на экран	18
7.31	Код 34h. Чтение параметров изображения	18
7.32	Код 38h. Установка состояния и типа курсора	18
7.33	Код 39h. Установка координаты курсора	19
7.34	Код 3Ah. Установка координаты вывода текста.....	19
7.35	Код 3Bh. Чтение координаты текстового курсора	19

7.36	Код 3Ch. Чтение координаты вывода текста	19
7.37	Код 40h. Выбор шрифта	20
7.38	Код 41h. Запись шрифта во флэш-память	20
7.39	Код 42h. Удаление шрифта из флэш-памяти	21
7.40	Код 48h. Вывод символа	21
7.41	Код 49h. Вывод строки символов.....	22
7.42	Код 4Ah. Установка текущего цвета текста	22
7.43	Код 4Bh. Чтение длины строки символов	22
7.44	Код 50h. Чтение состояния тач-панели.....	23
7.45	Код 58h. Запись скрипта во флэш-память	23
7.46	Код 5Ah. Выполнение скрипта из флэш-памяти	25
7.47	Код 5Bh. Удаление скрипта из флэш-памяти.....	25
7.48	Код 60h. Изменение тактовой частоты процессора.....	25
7.49	Код 61h. Установка значения контраста.....	26
7.50	Код 62h. Чтение значения контраста	26
7.51	Код 63h. Переход в режим пониженного энергопотребления	26
7.52	Код 64h. Выход из режима пониженного энергопотребления	27
7.53	Код 68h. Изменение разрешения TFT-монитора (для DVI-версии)	27
8	Временные диаграммы	28
9	Эскиз печатной платы	28
10	Рекомендации по включению	31
11	Изменения	32
12	Ссылки.....	33

Плата сопряжения ТФТ-Компаньон

- ✓ Позволяет подключать TFT-модуль к любому микроконтроллеру по SPI интерфейсу;
- ✓ Имеет полнофункциональный набор команд вывода графических примитивов;
- ✓ Позволяет многократно сократить вычислительные затраты на обслуживание графического интерфейса пользователя;
- ✓ Кардинально упрощает применение TFT-модулей в пользовательской системе. Добавить цветной интерфейс в изделие – не сложнее, чем поставить микросхему флэш-памяти с интерфейсом SPI.

1 Назначение

ТФТ-Компаньон (ТФТ-К) предназначен для подключения TFT-модуля с цифровым интерфейсом к произвольному микроконтроллеру, имеющему интерфейс SPI. Список поддерживаемых модулей можно узнать на сайте [1]. Вывод на экран может производиться как попиксельно (16 бит на пиксел), так и при помощи команд вывода графических примитивов (линий, закрашенных областей, текста и т.д.). ТФТ-К имеет встроенную флэш-память, в которой пользователь может хранить изображения, шрифты и наборы команд (скрипты). Поддерживается работа с тач-панелью и текстовым курсором.

2 Описание работы

ТФТ-Компаньон состоит из двух плат - основной процессорной платы и дочерней платы, служащей для подключения к конкретному TFT-модулю. Структура ТФТ-К показана на рис. 1.

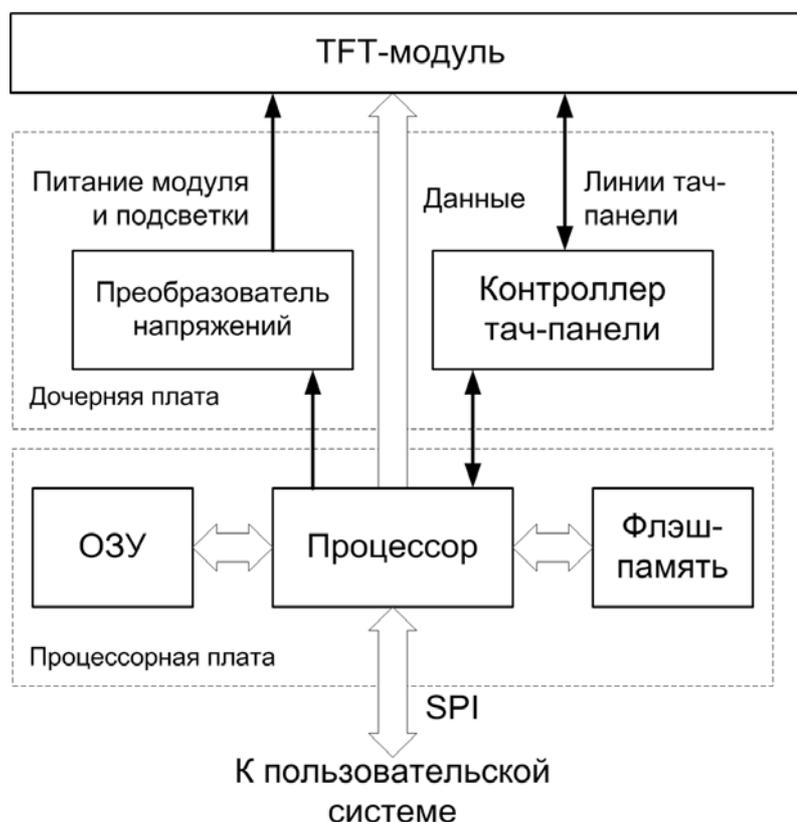


Рис. 1. Структура ТФТ-Компаньона

Процессорная плата неизменна для всех типов TFT-модулей, разрешением от 320x240 до 1280x1024. Процессор, установленный на ней, принимает и обрабатывает команды от пользовательской системы по интерфейсу SPI и обеспечивает регенерацию изображения на экране из ОЗУ, размещенного на плате. ОЗУ содержит 4 (2 для DVI-версии) видеостраницы.

Процессор имеет входной буфер типа FIFO объемом 10 кБайт, в который принимаются команды/данные переданные по SPI пользовательской системой. Благодаря этому пользовательская система может передавать блоки команд до 10 кБайт, не ожидая выполнения каждой команды в последовательности.

Флэш-память, размещенная на процессорной плате, позволяет пользователю хранить изображения, шрифты и скрипты. Это позволяет максимально разгрузить пользовательскую систему, и высвободить вычислительные ресурсы.

Для каждого типа модулей создается своя дочерняя плата.

Она может содержать:

- Преобразователи питающих напряжений;
- Схемы генерации VCOM и напряжений гамма-коррекции;
- DC/DC преобразователь для питания LED-подсветки с программной регулировкой яркости;
- Контроллер тач-панели и разъем для ее подключения;
- Разъем для подключения TFT-модуля.

Процессорная плата жестко припаивается к дочерней плате при монтаже. Две платы поставляются только комплектом.

3 Соединители

TFT-Компаньон имеет следующие разъемы:

1. Разъем для связи с пользовательской системой (IDC-10MR);
2. Разъем для подключения TFT-модуля или TFT-монитора (DVI-D);
3. Разъем для подключения тач-панели (не для всех TFT-модулей);
4. Разъем для подключения подсветки (в версии для LVDS интерфейса);
5. Разъем для питания DC/DC преобразователя подсветки (в версии для LVDS интерфейса) (WF-2MR).

Назначение выводов разъема связи с пользовательской системой Таблица 1

Номер	Название	Назначение
1	#RST	Сброс
2	DO	Выходные данные SPI
3	GND	Общая точка
4	SCLK	Тактовый сигнал SPI
5	GND	Общая точка
6	DI	Входные данные SPI
7	#CS	Сигнал выбора TFT-К
8	GND	Общая точка
9	Vdd	+3.3 V питание
10	Vdd	+3.3 V питание

Назначение выводов разъема питания DC/DC преобразователя подсветки (в версии для LVDS интерфейса)

Таблица 2

Номер	Название	Назначение
1	V+	Питание
2	GND	Общая точка

Назначение выводов разъема тач-панели, LED-подсветки и TFT-модуля можно найти в документации на соответствующий модуль.

4 Характеристики

Максимально допустимые значения:

1. Напряжение питания платы, В-0.3...+3.6
2. Напряжение на логических входах, В-0.3...+3.6
3. Рабочая температура, град. С-40...+85
4. Температура хранения, град. С-50...+85

Рекомендуемые условия работы:

1. Напряжение питания платы, В+3.0...+3.3
2. Напряжение на логических входах3.3 В LVTTTL
3. Допустимые пульсации напряжения питания, не более100 мВ
4. Напряжение питания преобразователя подсветки LVDS-версии, В +4.0...+13.0

Потребление при напряжении питания 3.3 В:

1. Типовое потребление платы TFT-Компаньона при разрешении 640x480, мА при скорости ядра

- 0:60
- 1:70
- 2:100
- 3:130

2. Типовое потребление DC/DC преобразователей подсветки и питающих напряжений при максимуме яркости, мА (для справки):

- для модуля RH320240-006 (3.5")210
- для модуля AM640480 (5.7")610
- для модуля MTF-T070 (7.0")630
- для модуля MI1040GT-1 (10.4")600@5 В

3. Потребление в режиме пониженного энергопотребления с подключенным TFT-модулем или TFT-монитором, мА:

- для модуля RH320240-006 (3.5")20
- для модуля AM640480 (5.7")130
- для модуля MTF-T070 (7.0")90
- для DVI-версии20

4. Потребления платы TFT-Компаньон DVI при подключенном TFT-мониторе с разрешением 1280x1024, мА:240

5. Типовое потребление платы TFT-Компаньона с интерфейсом LVDS при скорости ядра 2 и подключенном TFT-модуле, мА:

- для модуля MI1040GT-1(10.4", 800x600)330 мА

Прочие характеристики:

1. Количество видеостраниц:
для разрешений 320x240, 640x480, 800x480, 800x6004
для DVI-версии2
2. Максимальное количество цветов65536
3. Объем доступной пользователю флэш-памяти1830 кБайт

5 Связь с пользовательской системой

ТФТ-Компаньон обменивается информацией с пользовательской системой посредством последовательного интерфейса SPI. SPI-порт процессора сконфигурирован как Slave в режиме 0 (CPHA=CPOL=0). Данные передаются по 8 бит, старшим битом вперед (MSB first). При передаче чисел разрядностью более одного байта младший байт передается первым.

Интерфейс имеет следующие линии:

- DI – линия входных данных;
- DO – линия выходных данных;
- SCLK – вход тактового сигнала защелкивания данных;
- #CS – вход выбора микросхемы. Активный уровень – низкий;
- #RST – вход сброса. Активный уровень – низкий.

После подачи питания на ТФТ-Компаньон обмен по SPI не должен осуществляться в течение 1600 мс, до окончания инициализации SPI-порта процессора.

После снятия сигнала сброса #RST обмен по SPI не должен осуществляться в течение 500 мс, до окончания инициализации SPI-порта процессора.

Структура блока обработки команд показана на рис. 2.



Рис. 2. Структура блока обработки команд

Когда в SPI-порт приходит очередной байт команд/данных, он записывается в FIFO-буфер объемом 10 кБайт. Команды из этого буфера обрабатываются ядром обработки команд в порядке поступления. Если команда должна вернуть данные, то эти данные записываются в FIFO буфер выходных данных, из которого поступают в SPI-порт.

Все байты фиксированной части команды должны поступить в буфер в течение 0.2 секунды. Если этого не происходит, срабатывает таймаут, и выполнение команды прерывается. Для переменной части команды – между приемом двух последовательных байт данных должно проходить не более 2 сек. Также и при передаче – если при передаче данных выходной буфер полон дольше, чем 2 сек, то выполнение команды прерывается.

Поскольку при обмене по SPI происходит одновременно и прием и передача, при приеме очередного байта в выходной буфер SPI порта записывается либо байт статуса, либо данные из выходного FIFO-буфера, если он не пуст. Последовательность данных из выходного буфера всегда начинается с байта FFh, что позволяет отличить ее от байта статуса.

Байт статуса имеет следующую структуру:

7	6	5	4	3	2	1	0
-	-	PD	WD	-	BF	BSY	IDL

Бит IDL – Idle – равен 1 когда входной буфер пуст, и ядро обработки команд находится в состоянии ожидания команды;

Бит BSY – Busy – равен 1 когда ядро обработки команд занято выполнением команды;

Бит BF – Buffer Full – равен 1 когда входной буфер полон.

Бит WD – Wait Data – равен 1 когда ядро обработки команд ожидает поступления данных для выполнения команды. Бит не выставляется в команде PutString.

Бит PD – Power Down – равен 1 когда TFT-компаньон находится в режиме пониженного энергопотребления. Когда этот бит равен 1, команды не выполняются за исключением команд опроса тач-панели (50h) и входа из режима пониженного энергопотребления (64h).

TFT-Компаньон имеет довольно большой буфер на входе – 10 кБайт, поэтому в большинстве случаев ожидать, пока он выполнит все команды - не надо. Команды будут выполняться по очереди из буфера.

Достаточно проверять флаг переполнения буфера BF перед передачей команды, или флаг IDL перед передачей всего экрана, если заведомо известно, что набор команд по его отображению умещается во входной буфер.

6 Особенности работы DVI-версии

TFT-Компаньон DVI позволяет подключить к любому микроконтроллеру TFT-мониторы с интерфейсом DVI-D, а также телевизоры с интерфейсом HDMI (через кабель DVI-HDMI). Это дает возможность отображать информацию на экранах с большой диагональю с минимальными затратами.

После включения питания TFT-Компаньон начинает регенерацию экрана с разрешением 640x480, которое поддерживают все мониторы. После этого из монитора через DDC2B считываются его параметры, определяется основное разрешение монитора.

Если монитор не подключен, или канал связи не соответствует стандарту DDC2B, т.е. если параметры не удастся считать, TFT-Компаньон ожидает подключения монитора, и не выполняет команды, передаваемые по SPI.

Если основное разрешение монитора превышает по количеству пикселей разрешение 1280x1024, то TFT-Компаньон устанавливает максимальное поддерживаемое разрешение для полученных пропорций экрана данного монитора.

Пропорции экрана	Разрешение
16:10	1440x900
4:3	1024x768
5:4	1280x1024
16:9	1366x768

После того, как параметры считаны и разрешение установлено, TFT-Компаньон переходит к обработке команд полученных от пользовательской системы по SPI.

Разрешение экрана может быть изменено командой 68h (см. п. 7.53 на стр. 27).

Информацию по DVI-версии TFT-Компаньона можно найти на соответствующей страничке на сайте [3].

7 Система команд

ТФТ-Компаньон имеет полнофункциональную систему команд, достаточную для построения графического интерфейса пользователя.

Краткое описание набора команд представлено в табл. 3.

Набор команд ТФТ-Компаньона

Таблица 3

Код команды	Название функции	Описание
00h	GetState()	Чтение текущего статуса/данных.
A5h	ResetTFT()	Сброс процессора, установка начальных значений
01h	SetVisualPage(p)	Установка номера видимой видеостраницы
02h	SetActivePage(p)	Установка номера изменяемой видеостраницы
03h	SendDataFlow(...)	Поточная запись видеостраницы
04h	SetBacklight(BI)	Устанавливает яркость подсветки от 0 до 255
05h	GetDataFlow()	Поточное чтение видеостраницы
06h	GetBacklight()	Возвращает значение яркости подсветки
0Ah	ClrScr()	Очистка экрана
0Bh	GetScreenSize()	Возвращает размер экрана по осям X и Y
0Ch	SetClipRect(X1,Y1,X2,Y2)	Устанавливает режим отсечки к заданной области
10h	PutPixel(X,Y,C)	Выводит пиксел цветом C с координатами X, Y
11h	GetPixel(X,Y)	Возвращает цвет пиксела с заданными координатами
12h	Line(X1,Y1,X2,Y2,C)	Выводит линию цветом C между указанными координатами
13h	LinePattern(X1,Y1,X2,Y2,C,P)	Выводит линию цветом C между указанными координатами с однобайтовым шаблоном P
15h	Rectangle(X1,Y1,X2,Y2,C)	Выводит прямоугольник цветом C с указанными координатами
16h	FillRect(X1,Y1,X2,Y2,C,BC)	Выводит прямоугольник цветом C с указанными координатами, заполняет его цветом BC
17h	Ellipse(Xc,Yc,Xr,Yr,C)	Выводит эллипс с координатами центра Xc, Yc, радиусом Xr,Yr, цветом C
18h	FillEllipse(Xc,Yc,Xr,Yr,BC)	Выводит эллипс с координатами центра Xc, Yc, радиусом Xr,Yr, и заполняет его цветом BC. Без рамки.
19h	FloodFill(X,Y,C,BC)	Заполняет область, ограниченную цветом C, заливкой цветом BC, начиная с координаты X, Y
20h	CopyRect(Xo1,Yo1,Xo2,Yo2,Po,Xn,Yn,Pn)	Копирует прямоугольник с координатами Xo, Yo на странице Po в прямоугольник с координатами Xn, Yn на странице Pn
21h	CopyToClipboard(X,Y,W,H)	Копирует в буфер обмена область с левым верхним углом X,Y, длиной W, высотой H
22h	PasteFromClipboard(X,Y)	Копирует в активную страницу с координаты X,Y данные из буфера обмена
30h	DrawImage(N,X,Y)	Вывести изображение из флэш с номером N с координатами левого верхнего угла X,Y
31h	SaveImage(N,W,H,T,S,...)	Записать во флэш изображение типа T, размером S байт, высотой H и шириной W под номером N. Возвращает номер изображения, или FFFFh если нет места
32h	DelImage(N)	Удалить из флэш изображение с номером N
33h	LoadBitmap(X,Y,W,H,...)	Вывести на экран изображение по координате X,Y, шириной W, высотой H без записи во флэш
34h	GetImagePar(N)	Возвращает высоту и ширину изображения с номером N
38h	SetCursorState(S,T)	Изменить состояние курсора S - вкл/выкл, T – тип курсора
39h	CursorXY(X,Y)	Установить координату текстового курсора. X и Y – произвольные координаты

3Ah	GotoXY(X,Y)	Установить координату вывода текста. X и Y – произвольные координаты
3Bh	GetCursorXY()	Возвращает текущую координату текстового курсора
3Ch	GetTextXY()	Возвращает текущую координату вывода текста
40h	SetFont(N)	Выбрать текущим шрифт N
41h	SaveFont(N,T,S,W,H, ,...)	Записать во флэш шрифт в запись N, типом T (моноширинный или в формате Microchip GUI) размером S байт, с размером символа WxH пикселей. Возвращает номер шрифта, или FFFFh если нет места
42h	DelFont(N)	Удалить шрифт с номером N
48h	Putchar(c)	Выводит символ текущим шрифтом с текущей координаты вывода текста
49h	PutString(...)	Выводит строку до получения байта 0 текущим шрифтом с текущей координаты вывода текста
4Ah	SetCharColor(C)	Устанавливает текущий цвет текста
4Bh	GetStringLen(...)	Возвращает длину строки, выведенной текущим шрифтом, в пикселях
50h	GetTouchStat()	Возвращает координату касания touch screen. Числа от 1 до 255. Если нет касания – возвращает 0.
58h	RecordScript(N,S,...)	Начало записи набора команд (скрипта) во флэш. N – номер записи. S – размер записи
5Ah	PlayScript(N,X,Y)	Выполнить из флэш скрипт номер N с координаты X, Y
5Bh	DelScript(N)	Удалить из флэш скрипт номер N
60h	SetSpeedGrade(S)	Изменить тактовую частоту процессора. S – номер тактовой частоты (0..3)
61h	SetContrast(c)	Установить значение контраста с (0..255)
62h	GetContrast()	Возвращает текущее значение контраста
63h	PowerDown()	Переводит TFT-Компаньон в режим пониженного энергопотребления
64h	PowerUp()	Выводит TFT-Компаньон из режима пониженного энергопотребления
68h	DVISetRes()	Устанавливает разрешение TFT-монитора (только для DVI-версии)

7.1 Формат пакета

Все команды TFT-Компаньона (кроме 00h) имеют унифицированный формат пакета:

Номер байта	0	1	2	...	N+2	N+3	N+4	...
Значение	AAh	IC	DF0	...	DFN	IC	DA0	...

AAh – первый байт пакета;

IC – код команды;

DF0...DFN – фиксированные (обязательные) данные для команды;

DA0... - данные переменной длины для команды.

Первым байтом пакета всегда передается байт AAh. За ним следует IC – код команды. За кодом команды передается последовательность данных фиксированной длины, обязательных для данной команды DF0...DFN. Завершает фиксированную часть байт, дублирующий код команды. Далее передаются данные переменной длины, если они нужны для выполнения команды. Количество байт данных в переменной части обычно задается в фиксированной части команды.

При передаче всех команд фиксированной длины проверять готовность входного FIFO-буфера принять новую команду (BF=0) нужно только однажды – перед передачей команды. В процессе передачи команды фиксированной длины анализировать бит BF не обязательно.

7.2 Формат 16-битного цвета пиксела

Цвет пиксела в TFT-Компаньоне задается в стандартном 16-битном формате:

Номер бита	15	14	...	10	9	...	5	4	...	0
Значение	IN	B4	...	B0	G4	...	G0	R4	...	R0

Цветовые составляющие R (красный), G (зеленый) и B (синий) кодируются пятью битами. Старший 15-й бит (IN) задает дополнительную градацию интенсивности, которая добавляется младшим битом ко всем цветовым составляющим. В большинстве применений не требуется использовать бит IN, поэтому его обычно делают равным 1.

Пример макроса на языке C для формирования значения цвета из составляющих:

```
#define COLOR(R,G,B) (0x8000|((unsigned int)B<<10)|((unsigned int)G<<5)|R)
```

7.3 Хранение данных во флэш-памяти

На плате ТФТ-Компаньона установлена флэш-память, позволяющая сохранять пользовательские изображения, шрифты и наборы команд (скрипты). Наличие такой памяти позволяет сократить количество данных, передаваемых от пользовательской системы в процессе вывода графики на экран и ускорить вывод.

Флэш-память имеет объем 1830 кБайт. Объект, хранящийся во флэш, идентифицируется номером от 0 до 1319. Таким образом, во флэш может быть записано до 1320 объектов. Объекты нумеруются пользователем при записи произвольным образом.

При записи объекту выделяется непрерывная область флэш, соответствующая размеру объекта. Если под записываемым номером ранее был записан другой объект, место, занимаемое этим объектом, освобождается после записи нового объекта.

Объект с номером 0 имеет особое значение – при рестарте системы процессор считывает его. Если объект 0 – изображение, то оно выводится автоматически на экран. Если объект 0 – скрипт, то он автоматически выполняется. Это избавляет пользовательскую систему от необходимости инициализации экрана при запуске системы.

На плате ТФТ-Компаньона имеется переключатель для защиты флэш-памяти от изменения (см п. 9). При разработке конечного изделия и в производстве – когда нужно изменять данные во флэш, эта переключатель разомкнута и запись во флэш-память разрешена. Перед отправкой изделия конечному пользователю переключатель защиты от записи может быть замкнута, что исключит возможность порчи содержимого флэш-памяти в процессе эксплуатации изделия.

7.4 Код 00h. Чтение статуса/данных

Команда чтения статуса/данных передается без заголовка пакета (AAh).

Передача по SPI кода 00h не инициирует никаких действий ядра выполнения команд, но позволяет считать байт статуса или данные от предыдущей команды, если они уже поступили в выходной FIFO-буфер.

Следует отметить, что команда 00h при приеме так же как и все прочие помещается во входной FIFO-буфер. Поэтому проверять готовность буфера принять новую команду (BF=0) при помощи этой команды можно только между передачей двух команд.

7.5 Код A5h. Сброс процессора

Формат команды:

Номер байта	0	1	2	3	4	5	6
Значение	AAh	A5h	FFh	AAh	00h	55h	A5h

Команда A5h выполняет программный сброс процессора. После передачи этой команды требуется пауза 500 мс, необходимая для перезагрузки процессора и инициализации SPI-порта, перед передачей следующей команды. При выполнении команды сброса содержимое входного FIFO-буфера уничтожается.

Команда сброса процессора не может выполняться из скрипта.

7.6 Код 01h. Установка видимой видеостраницы

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	01h	PN	01h

Команда устанавливает видимой видеостраницу с номером PN. Во избежание фликера перед переключением страницы процессор дожидается окончания регенерации текущей страницы, поэтому выполнение команды может занять до 20 мс.

Команда установки видимой видеостраницы может выполняться из скрипта.

7.7 Код 02h. Установка активной видеостраницы

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	02h	PN	02h

Команда устанавливает активной видеостраницу с номером PN. Все последующие команды вывода, не содержащие в параметрах номер видеостраницы, будут выполняться на видеостранице PN.

Команда установки активной видеостраницы может выполняться из скрипта.

7.8 Код 03h. Поточная запись видеостраницы

Формат команды:

Номер байта	0	1	2	3	4	...	N*2+3	N*2+4
Значение	AAh	03h	03h	D0.0	D0.1	...	DN.0	DN.1

Команда поточной записи видеостраницы позволяет передать всю страницу целиком. Вслед за двойным кодом команды последовательно передаются цвета пикселей, начиная с координаты 0,0, построчно. Всего с командой передается $X_Res * Y_Res * 2$ байтов данных, где X_Res и Y_Res – разрешение по горизонтали и вертикали соответственно.

Команда поточной записи выполняется без учета установленных границ отсечки.

Команда поточной записи не может выполняться из скрипта.

7.9 Код 05h. Поточное чтение видеостраницы

Формат команды:

Номер байта	0	1	2
Значение	AAh	05h	05h

Команда поточного чтения видеостраницы позволяет считать всю страницу целиком. Вслед за двойным кодом команды последовательно считываются цвета пикселей, начиная с координаты 0,0, построчно. Всего команда передает $X_Res * Y_Res * 2$ байтов данных, где X_Res и Y_Res – разрешение по горизонтали и вертикали соответственно. Байты данных предваряются байтом FFh, позволяющим отделить данные от байта статуса.

Команда поточного чтения выполняется без учета установленных границ отсечки.

Команда поточного чтения не может выполняться из скрипта.

7.10 Код 04h. Установка яркости подсветки

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	04h	BL	04h

По команде установки яркости подсветки новое значение подсветки BL (от 0 до 255) передается в DC/DC преобразователь питания подсветки. Ток подсветки прямо пропорционален этому значению. При значении 0 подсветка выключена.

Команда установки яркости подсветки может выполняться из скрипта.

7.11 Код 06h. Чтение яркости подсветки

Формат команды:

Номер байта	0	1	2
Значение	AAh	06h	06h

По команде чтения яркости подсветки возвращается установленное значение яркости (один байт). Байт данных предваряется байтом FFh, позволяющим отделить данные от байта статуса.

Команда чтения яркости подсветки не может выполняться из скрипта.

7.12 Код 0Ah. Очистка экрана

Формат команды:

Номер байта	0	1	2
Значение	AAh	0Ah	0Ah

По команде очистки экрана область активной страницы в пределах установленных границ отсечки заполняется цветом 0 (черный).

Команда очистки экрана может выполняться из скрипта.

7.13 Код 0Bh. Чтение разрешения экрана

Формат команды:

Номер байта	0	1	2
Значение	AAh	0Bh	0Bh

По команде чтения разрешения экрана возвращается количество точек по горизонтали и вертикали, на которые рассчитан данный экземпляр TFT-Компаньона.

Формат ответа:

Номер байта	0	1	2	3	4
Значение	FFh	XR.0	XR.1	YR.0	YR.1

XR – разрешение по горизонтали, YR – разрешение по вертикали. Двухбайтовые числа передаются младшим байтом вперед.

Команда чтения разрешения экрана не может выполняться из скрипта.

7.14 Код 0Ch. Установка границ отсечки

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9	10
Значение	AAh	0Ch	X1.0	X1.1	Y1.0	Y1.1	X2.0	X2.1	Y2.0	Y2.1	0Ch

Команда установки границ отсечки устанавливает разрешенную область для вывода графических объектов. (X1,Y1) – координаты левого верхнего угла разрешенной области, (X2,Y2) – правого нижнего.

После сброса процессора разрешенная область установлена на весь экран.

Команда установки границ отсечки может выполняться из скрипта. Но при выполнении скрипта с координаты, отличной от (0,0) границы отсечки устанавливаются относительно координаты выполнения скрипта, и не сохраняются после окончания его выполнения.

7.15 Код 10h. Вывод пиксела

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8
Значение	AAh	10h	X.0	X.1	Y.0	Y.1	C.0	C.1	10h

Команда выводит пиксел цветом C с координатой (X,Y) на активную страницу с учетом границы отсечки.

Команда вывода пиксела может выполняться из скрипта.

7.16 Код 11h. Чтение цвета пиксела

Формат команды:

Номер байта	0	1	2	3	4	5	6
Значение	AAh	11h	X.0	X.1	Y.0	Y.1	11h

Команда возвращает цвет пиксела с координатой (X,Y) на активной странице. Байты данных предваряются байтом FFh, позволяющим отделить данные от байта статуса.

Команда чтения цвета пиксела не может выполняться из скрипта.

7.17 Код 12h. Вывод прямой сплошной линии

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	12h	X1.0	X1.1	Y1.0	Y1.1	X2.0	X2.1	Y2.0	Y2.1
Номер байта	10	11	12							
Значение	C.0	C.1	12h							

Команда выводит прямую сплошную линию между точками (X1,Y1) и (X2,Y2) активной страницы цветом C с учетом границ отсечки.

Команда вывода прямой линии может выполняться из скрипта.

7.18 Код 13h. Вывод прямой пунктирной линии

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	13h	X1.0	X1.1	Y1.0	Y1.1	X2.0	X2.1	Y2.0	Y2.1
Номер байта	10	11	12	13						
Значение	C.0	C.1	P	13h						

Команда выводит прямую пунктирную линию между точками (X1,Y1) и (X2,Y2) активной страницы цветом C с учетом границ отсечки. При выводе используется шаблон P. Использование шаблона, задаваемого пользователем, позволяет получить максимальную гибкость при выводе пунктирной линии.

Команда вывода прямой пунктирной линии может выполняться из скрипта.

7.19 Код 15h. Вывод прямоугольника

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	15h	X1.0	X1.1	Y1.0	Y1.1	X2.0	X2.1	Y2.0	Y2.1
Номер байта	10	11	12							
Значение	C.0	C.1	15h							

Команда выводит прямоугольник с координатой левого верхнего угла (X1,Y1) и правого нижнего (X2,Y2) активной страницы цветом C с учетом границ отсечки.

Команда вывода прямоугольника может выполняться из скрипта.

7.20 Код 16h. Вывод заштрихованного прямоугольника

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	16h	X1.0	X1.1	Y1.0	Y1.1	X2.0	X2.1	Y2.0	Y2.1
Номер байта	10	11	12	13	14					
Значение	C.0	C.1	BC.0	BC.1	16h					

Команда выводит прямоугольник с координатой левого верхнего угла (X1,Y1) и правого нижнего (X2,Y2) активной страницы цветом С с учетом границ отсечки. Внутренняя область прямоугольника закрашивается цветом ВС.

Команда вывода заштрихованного прямоугольника может выполняться из скрипта.

7.21 Код 17h. Вывод эллипса

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	17h	XC.0	XC.1	YC.0	YC.1	XR.0	XR.1	YR.0	YR.1
Номер байта	10	11	12							
Значение	C.0	C.1	17h							

Команда выводит эллипс с координатами центра (XC,YC), радиусом по горизонтали XR, по вертикали YR, цветом С на активную страницу с учетом границ отсечки.

Команда вывода эллипса может выполняться из скрипта.

7.22 Код 18h. Вывод заштрихованного эллипса

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	18h	XC.0	XC.1	YC.0	YC.1	XR.0	XR.1	YR.0	YR.1
Номер байта	10	11	12							
Значение	BC.0	BC.1	18h							

Команда выводит заштрихованный эллипс с координатами центра (XC,YC), радиусом по горизонтали XR, по вертикали YR, цветом штриховки ВС на активную страницу с учетом границ отсечки. Эллипс выводится одним цветом, без рамки.

Команда вывода заштрихованного эллипса может выполняться из скрипта.

7.23 Код 19h. Штриховка произвольной области

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9	10
Значение	AAh	19h	X.0	X.1	Y.0	Y.1	C.0	C.1	BC.0	BC.1	19h

Команда штриховки произвольной области закрашивает цветом ВС область, ограниченную цветом С начиная с координаты (X,Y). Выполняется на активной странице, с учетом границ отсечки.

Команда штриховки произвольной области может выполняться из скрипта.

7.24 Код 20h. Копирование прямоугольной области

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	20h	Xo1.0	Xo1.1	Yo1.0	Yo1.1	Xo2.0	Xo2.1	Yo2.0	Yo2.1
Номер байта	10	11	12	13	14	15	16			
Значение	Po	Xn.0	Xn.1	Yn.0	Yn.1	Pn	20h			

Команда копирует прямоугольник с координатами (Xo1,Yo1)-(Xo2,Yo2) видеостраницы Po в прямоугольник с координатами левого верхнего угла (Xn,Yn) на странице Pn.

Данная команда позволяет реализовать скроллинг изображения в любом направлении.

Команда копирования прямоугольной области может выполняться из скрипта.

7.25 Код 21h. Копирование области в буфер обмена

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9	10
Значение	AAh	21h	X.0	X.1	Y.0	Y.1	W.0	W.1	H.0	H.1	21h

Команда копирует область активной страницы с координатами левого верхнего угла (X,Y) шириной W и высотой H в буфер обмена.

Команда копирования области в буфер обмена может выполняться из скрипта.

7.26 Код 22h. Вставка области из буфера обмена

Формат команды:

Номер байта	0	1	2	3	4	5	6
Значение	AAh	22h	X.0	X.1	Y.0	Y.1	22h

Команда вставляет область из буфера обмена в активную страницу с координатами левого верхнего угла (X,Y).

Команда копирования области в буфер обмена может выполняться из скрипта.

7.27 Код 30h. Вывод изображения из флэш-памяти

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8
Значение	AAh	30h	N.0	N.1	X.0	X.1	Y.0	Y.1	30h

Команда выводит изображение с номером N из флэш на активную видеостраницу с координатой левого верхнего угла (X,Y). Если объект с номером N не записан во флэш, или он не является изображением, то команда игнорируется. Если запись - это JPEG файл неподдерживаемого формата, то команда игнорируется.

Команда вывода изображения из флэш-памяти может выполняться из скрипта.

7.28 Код 31h. Запись изображения во флэш-память

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9	10
Значение	AAh	31h	N.0	N.1	W.0	W.1	H.0	H.1	T	S.0	S.1
Номер байта	11	12	13	14	...	i*2+13	i*2+14				
Значение	S.2	31h	C0.0	C0.1	...	Ci.0	Ci.1				

Команда записывает во флэш-память под номером N пользовательское изображение с типом изображения T.

Изображение может быть представлено либо в виде набора пикселей (T=0), передаваемых начиная с верхнего левого угла построчно, либо в виде JPEG-файла (T=7). W – ширина изображения, H – высота изображения. Трехбайтное значение S показывает размер JPEG-файла.

Декодер JPEG-файлов имеет следующие ограничения:

1. Размер JPEG-файла: до 400 кБайт;
2. Цветность: RGB 24 бита;
3. Не поддерживается прогрессивный формат.

Получить файлы подходящего формата можно, к примеру, при помощи бесплатной программы IrfanView [2].

При приеме команды сначала принимается фиксированная часть пакета, и проверяется наличие места во флэш для записи этого изображения. Если места нет – возвращается три байта FFh, и выполнение команды прерывается. При получении признака отсутствия места передача данных должна быть прекращена.

Если место под изображение найдено, изображение начинает записываться во флэш. По окончании записи, если она была успешна, отсылается признак ответа FFh и два байта номера объекта N.

Если при приеме изображения между байтами данных прошло более 0.2 с, то команда прерывается по таймауту, и возвращается три байта FFh.

Если перемычка защиты флэш-памяти от записи замкнута, то возвращается три байта FFh.

Команда записи изображения во флэш-память не может выполняться из скрипта.

7.29 Код 32h. Удаление изображения из флэш-памяти

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	32h	N.0	N.1	32h

Команда удаляет изображение с номером N из флэш. Если объект с номером N не записан во флэш, то команда игнорируется.

Команда удаления изображения из флэш-памяти не может выполняться из скрипта.

7.30 Код 33h. Прием изображения на экран

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	33h	X.0	X.1	Y.0	Y.1	W.0	W.1	H.0	H.1
Номер байта	10	11	12	...	i*2+11	i*2+12				
Значение	33h	C0.0	C0.1	...	Ci.0	Ci.1				

Команда выводит на активную страницу пользовательское изображение шириной W и высотой H. (X,Y) – координата левого верхнего угла. Команда приема изображения подобна команде поточной записи видеостраницы (код 03h), но в отличие от нее позволяет вывести изображение любого размера начиная с любой координаты.

Сначала принимается фиксированная часть пакета, вычисляется количество данных изображения (W*H*2 байт). Затем принимаемые данные из переменной части пакета записываются в активную видеостраницу.

Если при приеме изображения между байтами данных прошло более 0.2 с, то команда прерывается по таймауту.

Команда приема изображения на экран не может выполняться из скрипта.

7.31 Код 34h. Чтение параметров изображения

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	34h	N.0	N.1	34h

Команда возвращает ширину W и высоту H изображения, записанного во флэш-памяти под номером N.

Формат ответа:

Номер байта	0	1	2	3	4
Значение	FFh	W.0	W.1	H.0	H.1

Если записи под номером N не существует, или она содержит не изображение, то возвращается W=FFFFh и H=FFFFh.

Команда чтения параметров изображения не может выполняться из скрипта.

7.32 Код 38h. Установка состояния и типа курсора

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	38h	S	T	38h

Команда позволяет включить (S=FFh) или выключить (S=0) текстовый курсор. При этом указывается тип курсора. Предусмотрены следующие типы:

- 00h: ctInsBar – вертикальная линия в один пиксел шириной;

- 01h: ctOverwrite – прямоугольник размером в знакоместо;
- 02h: ctUnderline – горизонтальная линия подчеркивания толщиной в два пикселя.

Текстовый курсор отображается на видимой странице, в позиции установленной командой CursorXY (код 39h). Он мигает с частотой примерно 1 Гц. Перемещение курсора, его выключение при выводе графики на видимую страницу, при переключении страниц, необходимо выполнять пользовательской системе при помощи команд 38h и 39h.

Команда установки состояния и типа курсора не может выполняться из скрипта.

7.33 Код 39h. Установка координаты курсора

Формат команды:

Номер байта	0	1	2	3	4	5	6
Значение	AAh	39h	X.0	X.1	Y.0	Y.1	39h

Команда устанавливает координаты (X,Y) левого верхнего угла знакоместа, в котором будет отображаться текстовый курсор.

Команда установки координаты курсора может выполняться из скрипта.

7.34 Код 3Ah. Установка координаты вывода текста

Формат команды:

Номер байта	0	1	2	3	4	5	6
Значение	AAh	3Ah	X.0	X.1	Y.0	Y.1	3Ah

Команда устанавливает координаты (X,Y) левого верхнего угла знакоместа, в которое будет выведен следующий символ при помощи команд PutChar или PutString.

Команда установки координаты вывода текста может выполняться из скрипта.

7.35 Код 3Bh. Чтение координаты текстового курсора

Формат команды:

Номер байта	0	1	2
Значение	AAh	3Bh	3Bh

По команде чтения текущей координаты текстового курсора возвращаются его координаты по горизонтали (X) и вертикали (Y).

Формат ответа:

Номер байта	0	1	2	3	4
Значение	FFh	X.0	X.1	Y.0	Y.1

Двухбайтовые числа передаются младшим байтом вперед.

Команда чтения координаты текстового курсора не может выполняться из скрипта.

7.36 Код 3Ch. Чтение координаты вывода текста

Формат команды:

Номер байта	0	1	2
Значение	AAh	3Ch	3Ch

По команде чтения текущей координаты вывода текста возвращаются координаты по горизонтали (X) и вертикали (Y) левого верхнего угла следующего символа.

Формат ответа:

Номер байта	0	1	2	3	4
Значение	FFh	X.0	X.1	Y.0	Y.1

Двухбайтовые числа передаются младшим байтом вперед.

Команда чтения координаты вывода текста не может выполняться из скрипта.

7.37 Код 40h. Выбор шрифта

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	40h	N.0	N.1	40h

Команда выбирает для вывода символов шрифт, записанный во флэш-памяти под номером N. Шрифт может быть либо пропорциональным моноширинным, либо с символами переменной ширины в формате Microchip GUI (см. п. 7.38).

ТФТ-Компаньон имеет несколько встроенных шрифтов, которые размещаются во внутренней памяти процессора и благодаря этому выводятся быстрее остальных. Встроенные шрифты выбираются указанием следующих номеров в поле N:

- 2000h: Шрифт размером 8x14 пикселей;
- 2001h: Шрифт размером 8x8 пикселей.

Команда выбора шрифта может выполняться из скрипта.

7.38 Код 41h. Запись шрифта во флэш-память

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	9
Значение	AAh	41h	N.0	N.1	T	S.0	S.1	S.2	W	H
Номер байта	10	11	...	i+11						
Значение	41h	D0	...	Di						

Команда записывает во флэш-память под номером N пользовательский шрифт типа T. Для пропорциональных моноширинных шрифтов T=0, для шрифтов в формате Microchip GUI T=3.

После фиксированной части команды посылаются данные о шрифте в количестве S байт. Символ пропорционального шрифта имеет ширину W и высоту H. Максимально допустимый размер символа пропорционального шрифта 32x32 пиксела. Ширина символа W для шрифтов переменной ширины используется при выводе курсора и как ширина пробела (символ 20h).

Данные для пропорциональных шрифтов имеют следующий формат (на примере символа «2» шрифта 12x18):

11	10	9	8	7	6	5	4	3	2	1	0	
												0000h
												00F0h
												0108h
												0204h
												0004h
												0004h
												0008h
												0010h
												0020h
												0040h
												0080h
												0100h
												0204h
												03FCh
												0000
												0000
												0000
												0000

Первыми передаются данные о верхней строке, числа более одного байта передаются младшими байтами вперед. Таким образом, для приведенного символа поток данных будет следующим: 00h, 00h, F0h, 00h, 08h, 01h, 04h, 02h, 04h, 00h, 04h, 00h, 08h, 00h, 10h, 00h, 20h, 00h, 40h, 00h, 80h, 00h, 00h, 01h, 04h, 02h, FCh, 03h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h.

Шрифты в формате Microchip GUI имеют переменную ширину символа. Их можно получить при помощи конвертора из любого TrueType шрифта. При этом задается размер символа, и диапазон используемых символов. Полученный двоичный (*.bin) файл записывается во флэш ТФТ-компаньона командой 41h, и в дальнейшем этот шрифт может использоваться наравне с пропорциональными моноширинными шрифтами. Конвертор можно скачать со странички ТФТ-Компаньона [1].

Команда записи шрифта во флэш-память не может выполняться из скрипта.

7.39 Код 42h. Удаление шрифта из флэш-памяти

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	42h	N.0	N.1	42h

Команда удаляет шрифт с номером N из флэш. Если объект с номером N не записан во флэш, то команда игнорируется.

Команда удаления шрифта из флэш-памяти не может выполняться из скрипта.

7.40 Код 48h. Вывод символа

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	48h	C	48h

Команда выводит символ *C* на активную страницу. Символ выводится текущим шрифтом, текущим цветом, в текущую позицию вывода текста, установленную командой с кодом 3Ah (GotoXY). После выполнения команды позиция вывода текста автоматически сдвигается на один символ.

Команда вывода символа может выполняться из скрипта.

7.41 Код 49h. Вывод строки символов

Формат команды:

Номер байта	0	1	2	3	4	i+3	i+4
Значение	AAh	49h	49h	C0	...	Ci	0

Команда выводит строку символов (C0...Ci) на активную страницу. Строка выводится текущим шрифтом, текущим цветом, в текущую позицию вывода текста, установленную командой с кодом 3Ah (GotoXY). После выполнения команды позиция вывода текста автоматически сдвигается на длину строки.

Строка должна оканчиваться нулевым байтом. Длина строки – не более 254 символов.

Команда вывода строки символов может выполняться из скрипта.

7.42 Код 4Ah. Установка текущего цвета текста

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	4Ah	C.0	C.1	4Ah

Команда устанавливает текущий цвет текста равным *C*.

Команда установки цвета текста может выполняться из скрипта.

7.43 Код 4Bh. Чтение длины строки символов

Формат команды:

Номер байта	0	1	2	3	4	i+3	i+4
Значение	AAh	4Bh	4Bh	C0	...	Ci	0

Команда принимает строку символов (C0...Ci) и подсчитывает длину в пикселах, которая получится при выводе строки на экран текущим шрифтом. Полученная величина *L* отсылается в ответе.

Строка должна оканчиваться нулевым байтом. Длина строки – не более 254 символов.

Формат ответа:

Номер байта	0	1	2
Значение	FFh	L.0	L.1

Двухбайтовые числа передаются младшим байтом вперед.

Команда чтения длины строки символов не может выполняться из скрипта.

7.44 Код 50h. Чтение состояния тач-панели

Формат команды:

Номер байта	0	1	2
Значение	AAh	50h	50h

Команда считывает состояние тач-панели из контроллера панели, и передает его пользовательской системе.

Формат ответ:

Номер байта	0	1	2
Значение	FFh	X	Y

X и Y – координаты нажатия тач-панели по горизонтали и вертикали в диапазоне от 1 до 255. Если X и Y равны нулю – нет нажатия.

Опрос тач-панели занимает 1 мс. При опросе не производится подавление дребезга, поскольку оно может занимать значительное время. Поэтому обработку дребезга при нажатии пользовательской системе нужно производить самостоятельно.

Команда чтения состояния тач-панели не может выполняться из скрипта.

7.45 Код 58h. Запись скрипта во флэш-память

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8	...	i+8
Значение	AAh	58h	N.0	N.1	S.0	S.1	S.2	58h	C0	...	Ci

Команда записывает набор команд во флэш с номером N размером S байт. Размер скрипта ограничен 100 кБайтами.

Скрипт представляет собой последовательность команд, по формату соответствующих командам, передаваемым по SPI-интерфейсу. Но не все команды могут выполняться из флэш. Возможность выполнения команды из скрипта указана в табл. 4.

Возможность выполнения команд из скрипта

Таблица 4

Код команды	Название функции	Допустима в скрипте
00h	GetState()	-
A5h	ResetTFT()	-
01h	SetVisualPage(p)	+
02h	SetActivePage(p)	+*
03h	SendDataFlow(...)	-
04h	SetBacklight(BI)	+
05h	GetDataFlow()	-
06h	GetBacklight()	-
0Ah	ClrScr()	+
0Bh	GetScreenSize()	-
0Ch	SetClipRect(X1, Y1, X2, Y2)	+*
10h	PutPixel(X, Y, C)	+

11h	GetPixel(X,Y)	-
12h	Line(X1,Y1, X2,Y2,C)	+
13h	LinePattern(X1,Y1, X2,Y2,C,P)	+
15h	Rectangle(X1,Y1, X2,Y2,C)	+
16h	FillRect(X1,Y1, X2,Y2,C,BC)	+
17h	Ellipse(Xc,Yc, Xr,Yr,C)	+
18h	FillEllipse(Xc,Yc, Xr,Yr,BC)	+
19h	FloodFill(X,Y,C,BC)	+
20h	CopyRect(Xo1,Yo1, Xo2,Yo2,Po,Xn,Yn, Pn)	+
21h	CopyToClipboard(X, Y,W,H)	+
22h	PasteFromClipboard(X,Y)	+
30h	DrawImage(N,X,Y)	+
31h	SaveImage(N,W,H, T, S...)	-
32h	DelImage(N)	-
33h	LoadBitmap(X,Y,W, H, ...)	-
34h	GetImagePar(N)	-
38h	SetCursorState(S,T)	-
39h	CursorXY(X,Y)	+
3Ah	GotoXY(X,Y)	+
3Bh	GetCursorXY	-
3Ch	GetTextXY	-
40h	SetFont(N)	+
41h	SaveFont(N,T,S,W,H , ...)	-
42h	DelFont(N)	-
48h	Putchar(c)	+
49h	PutString(...)	+
4Ah	SetCharColor(C)	+
4Bh	GetStringLen(...)	-
50h	GetTouchStat()	-
58h	RecordScript(N,S,...)	-
5Ah	PlayScript(N,X,Y)	+**
5Bh	DelScript(N)	-
60h	SetSpeedGrade(S)	+
61h	SetContrast(c)	-
62h	GetContrast()	-
63h	PowerDown()	-
64h	PowerUp()	-
68h	DVISetRes()	-

* - при выполнении скрипта начиная с координат, отличных от (0,0), выполнение команд установки активной страницы и установки границ отсечки вызовет неверное выполнение скрипта.

** - допускается вложение скриптов (т.е. запуск скрипта из скрипта) с глубиной вложения не более четырех уровней вложенности.

Следует проявлять особую осторожность при записи скрипта в ячейку с номером 0. Этот скрипт выполняется автоматически после сброса процессора.

Команда записи скрипта не может выполняться из скрипта.

7.46 Код 5Ah. Выполнение скрипта из флэш-памяти

Формат команды:

Номер байта	0	1	2	3	4	5	6	7	8
Значение	AAh	5Ah	N.0	N.1	X.0	X.1	Y.0	Y.1	5Ah

Команда выполняет набор команд – скрипт, с номером N из флэш на активной видеостранице относительно координаты (X,Y). Команды проверяются на соответствие табл. 3 и выполняются последовательно.

Команда выполнения скрипта может выполняться из скрипта со степенью вложенности не более четырех уровней.

7.47 Код 5Bh. Удаление скрипта из флэш-памяти

Формат команды:

Номер байта	0	1	2	3	4
Значение	AAh	5Bh	N.0	N.1	5Bh

Команда удаляет скрипт с номером N из флэш. Если объект с номером N не записан во флэш, то команда игнорируется.

Команда удаления скрипта из флэш-памяти не может выполняться из скрипта.

7.48 Код 60h. Изменение тактовой частоты процессора

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	60h	S	60h

Команда устанавливает тактовую частоту процессора и ОЗУ ТФТ-Компаньона. Это непосредственно влияет на скорость приема и выполнения команд, но при этом увеличение быстродействия ведет к увеличению потребления.

Параметр S может принимать значения от 0 до 3. Необходимость установки той или иной скорости зависит от количества выводимой графики и допустимого потребления. Для обслуживания дисплея с разрешением 320x240 обычно достаточно скорости 0, для DVI-версии скорость фиксирована и не изменяется. При рестарте ТФТ-Компаньона процессор выставляет рекомендуемую для данного разрешения скорость работы.

При смене тактовой частоты процессор в течение 5-10 мс не может принимать команды по интерфейсу SPI. Поэтому после передачи команды установки частоты следует выдержать паузу в передаче, в противном случае возможны сбои в приеме последующих команд.

Команда установки скорости процессора может выполняться из скрипта.

7.49 Код 61h. Установка значения контраста

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	61h	C	61h

Команда устанавливает значение контраста для TFT-модуля. Параметр C может принимать значения от 0 до 255. Напряжение контраста прямо пропорционально C. Напряжение контраста требуется регулировать нечасто – при смене TFT-модуля. Делать это следует с осторожностью, так как при определенных значениях контраста изображение не будет видно. Установленное значение сохраняется в энергонезависимой памяти преобразователя напряжений, и автоматически устанавливается при рестарте системы.

Команда установки значения контраста в настоящее время не используется.

Команда установки контраста не может выполняться из скрипта.

7.50 Код 62h. Чтение значения контраста

Формат команды:

Номер байта	0	1	2
Значение	AAh	62h	62h

Команда возвращает установленное значение контраста для TFT-модуля в диапазоне от 0 до 255. Сначала передается признак ответа – байт FFh, а затем – однобайтовое значение контраста.

Команда чтения значения контраста не может выполняться из скрипта.

7.51 Код 63h. Переход в режим пониженного энергопотребления

Формат команды:

Номер байта	0	1	2
Значение	AAh	63h	63h

Команда переводит TFT-Компаньон в режим пониженного энергопотребления, при этом:

- ✓ Отключается подсветка TFT-модуля;
- ✓ Снижается тактовая частота процессора и контроллера тач-скрина;
- ✓ Все микросхемы переводятся в ждущий режим.

В режиме пониженного энергопотребления обрабатываются только две команды: опрос тач-панели GetTouchStat (50h) и выход из режима пониженного энергопотребления PowerUp (64h).

Следует иметь в виду, что в режиме пониженного энергопотребления скорость обработки байтов, получаемых по SPI, снижена примерно в 10 раз по сравнению с нормальной работой с градацией скорости 0.

Длительность выполнения команды перехода в режим пониженного энергопотребления 1 мс.

Команда перехода в режим пониженного энергопотребления не может выполняться из скрипта.

7.52 Код 64h. Выход из режима пониженного энергопотребления

Формат команды:

Номер байта	0	1	2
Значение	AAh	64h	64h

Команда выводит TFT-Компаньон из режима пониженного энергопотребления, при этом:

- ✓ Все микросхемы выводятся из ждущего режима;
- ✓ Восстанавливается тактовая частота процессора и контроллера тач-скрина;
- ✓ Восстанавливается подсветка TFT-модуля.

Тактовая частота процессора и яркость подсветки TFT-модуля восстанавливается до значений, которые были установлены перед переходом в режим пониженного энергопотребления.

Содержимое ВидеоОЗУ в режиме пониженного энергопотребления сохраняется.

Время выхода из режима пониженного энергопотребления - до 5 мс. В это время в приеме команд по интерфейсу SPI могут возникать ошибки, поэтому не следует передавать команды или данные сразу за командой PowerUp.

Команда выхода из режима пониженного энергопотребления не может выполняться из скрипта.

7.53 Код 68h. Изменение разрешения TFT-монитора (для DVI-версии)

Формат команды:

Номер байта	0	1	2	3
Значение	AAh	68h	R	68h

Команда изменяет разрешение TFT-монитора на одно из стандартных разрешений. Параметр R может принимать следующие значения:

Значение R	Разрешение
0	640x480
1	800x480
2	800x600
3	1024x768
4	1366x768
5	1440x900
6	1280x1024
7	Основное для монитора

Основное для монитора разрешение (R=7) – разрешение, параметры которого считаны по DDC2B из монитора.

Команда изменения разрешения не может выполняться из скрипта.

8 Временные диаграммы

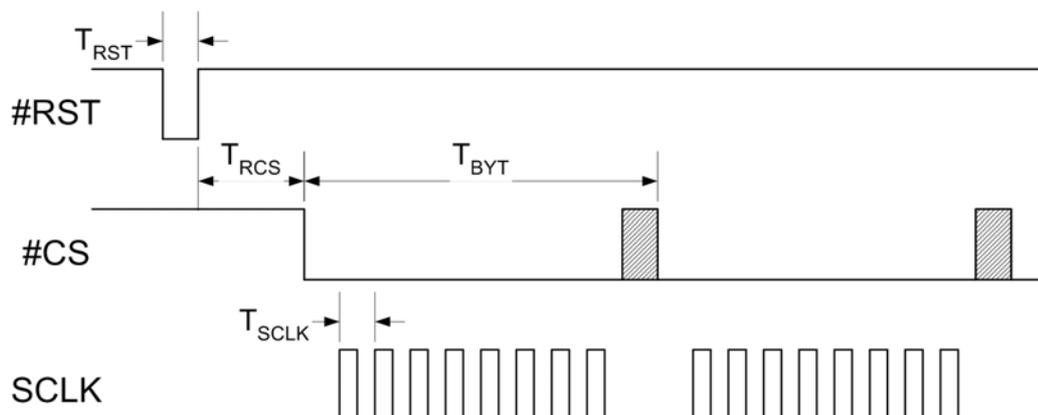


Рис. 3. Временная диаграмма сброса и передачи данных

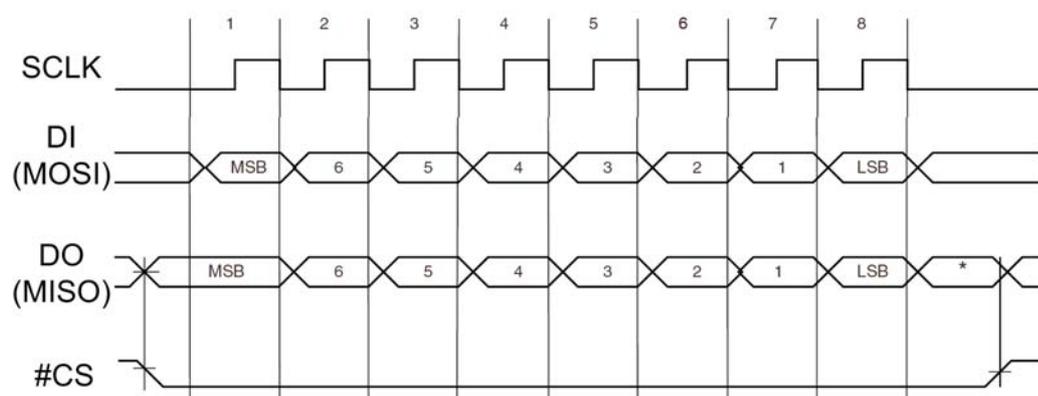


Рис. 4. Временная диаграмма передачи байта по SPI

SPI интерфейс работает в режиме 0 (CPOL=CPHA=0), при этом данные выставляются по срезу SCLK, а защелкиваются – по фронту.

Временные параметры указаны в табл. 5.

Таблица 5

Параметр	Скорость				Ед. изм.
	0	1	2	3	
T_{RST}, min	10				мкс
T_{RCS}, min	500				мс
T_{BYT}, min	6.68	3.34	1.67	1.37	мкс
T_{SCLK}, min	50	25	25	20	нс

9 Эскиз печатной платы

Эскиз печатной платы ТФТ-Компаньона для 3.5-дюймовых дисплеев приведен на рис. 5, для 5.7 и 7-дюймовых дисплеев – на рис. 6, для DVI-мониторов – на рис. 7.

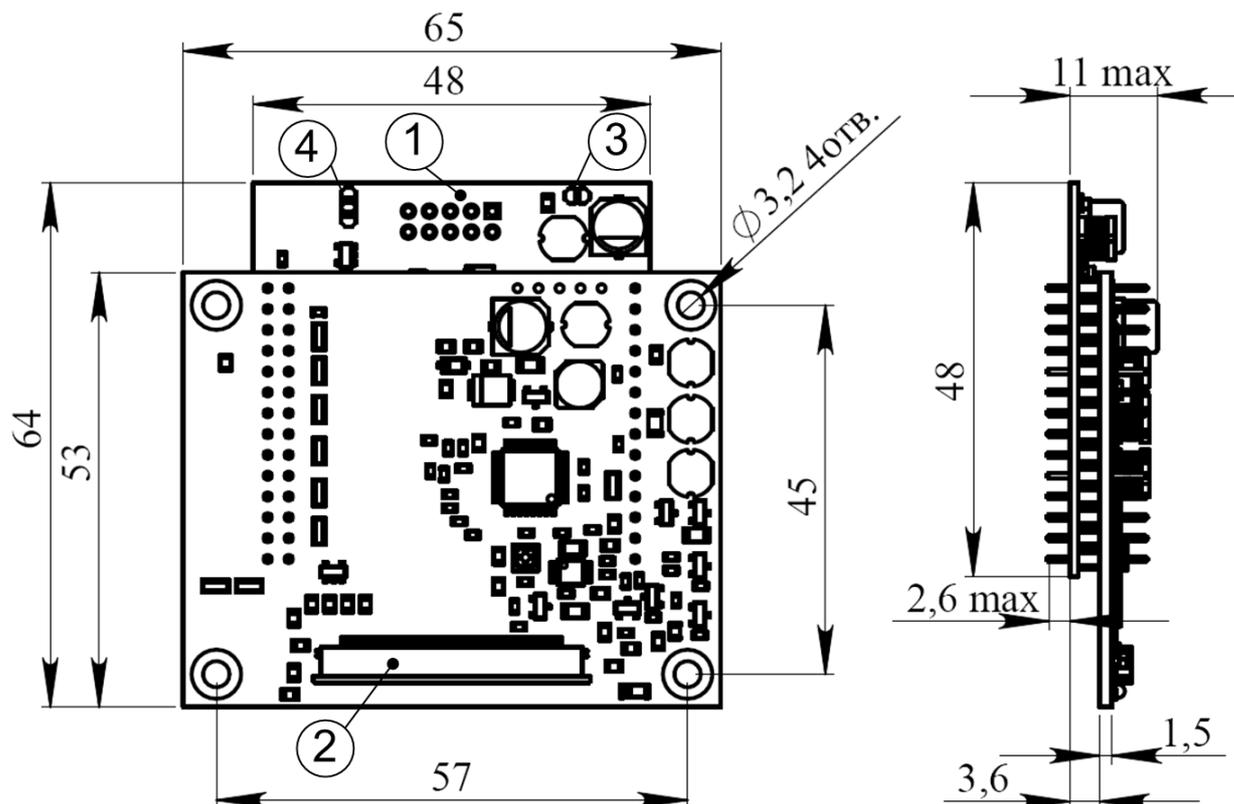


Рис. 5. Печатная плата для TFT-модулей с диагональю 3.5 дюйма

На рисунке: 1 – 10-контактный двухрядный разъем связи с пользовательской системой. Шаг выводов 2.54 мм. 2 – 54-контактный разъем для подключения TFT-модуля. 3 – перемычка для защиты флэш-памяти от изменения. Замкнута – запись запрещена. 4 – Перемычки J1 и J2 на рис. 9.

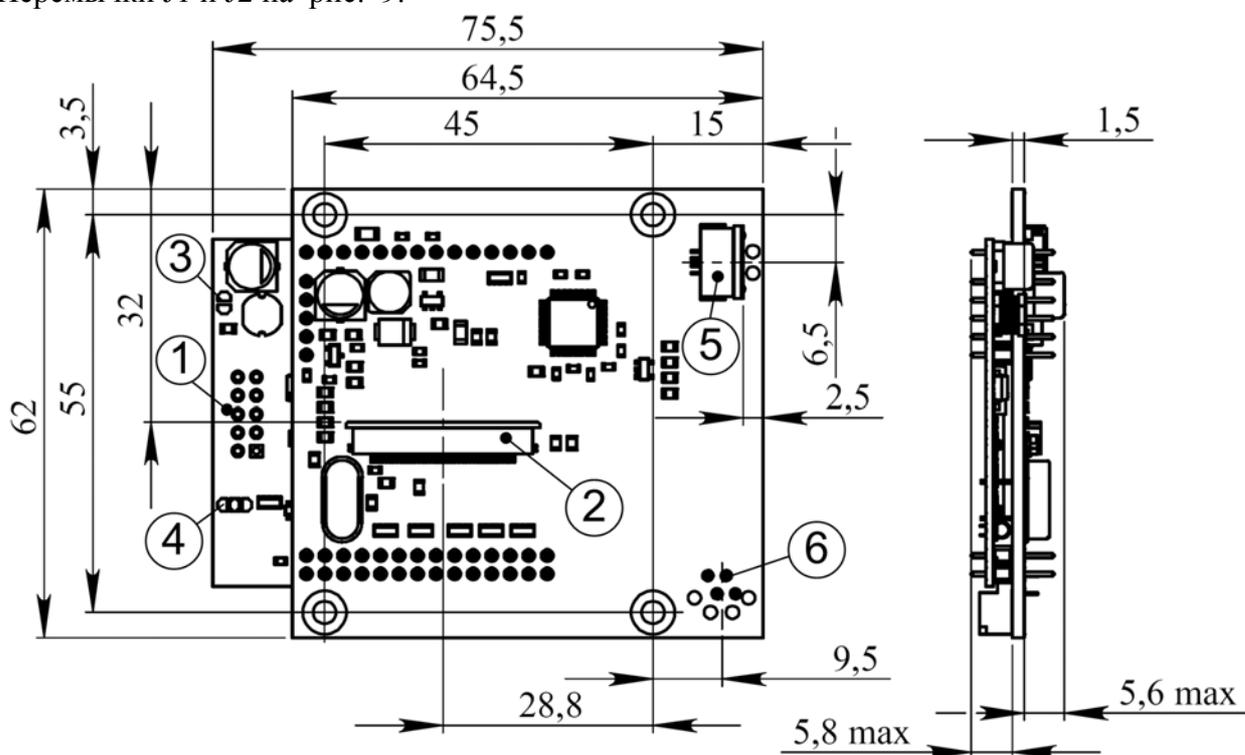


Рис. 6. Печатная плата для TFT-модулей с диагональю 5.7 и 7 дюймов

На рисунке: 1 – 10-контактный двухрядный разъем связи с пользовательской системой. Шаг выводов 2.54 мм. 2 – 40-контактный разъем для подключения TFT-модуля. 3 – перемычка для защиты флэш-памяти от изменения. Замкнута – запись запрещена. 4 – Перемычки J1 и J2 на рис. 9. 5 – разъем для подключения тач-панели 7” TFT-модуля. 6 – разъем для подключения тач-панели 5.7” TFT-модуля.

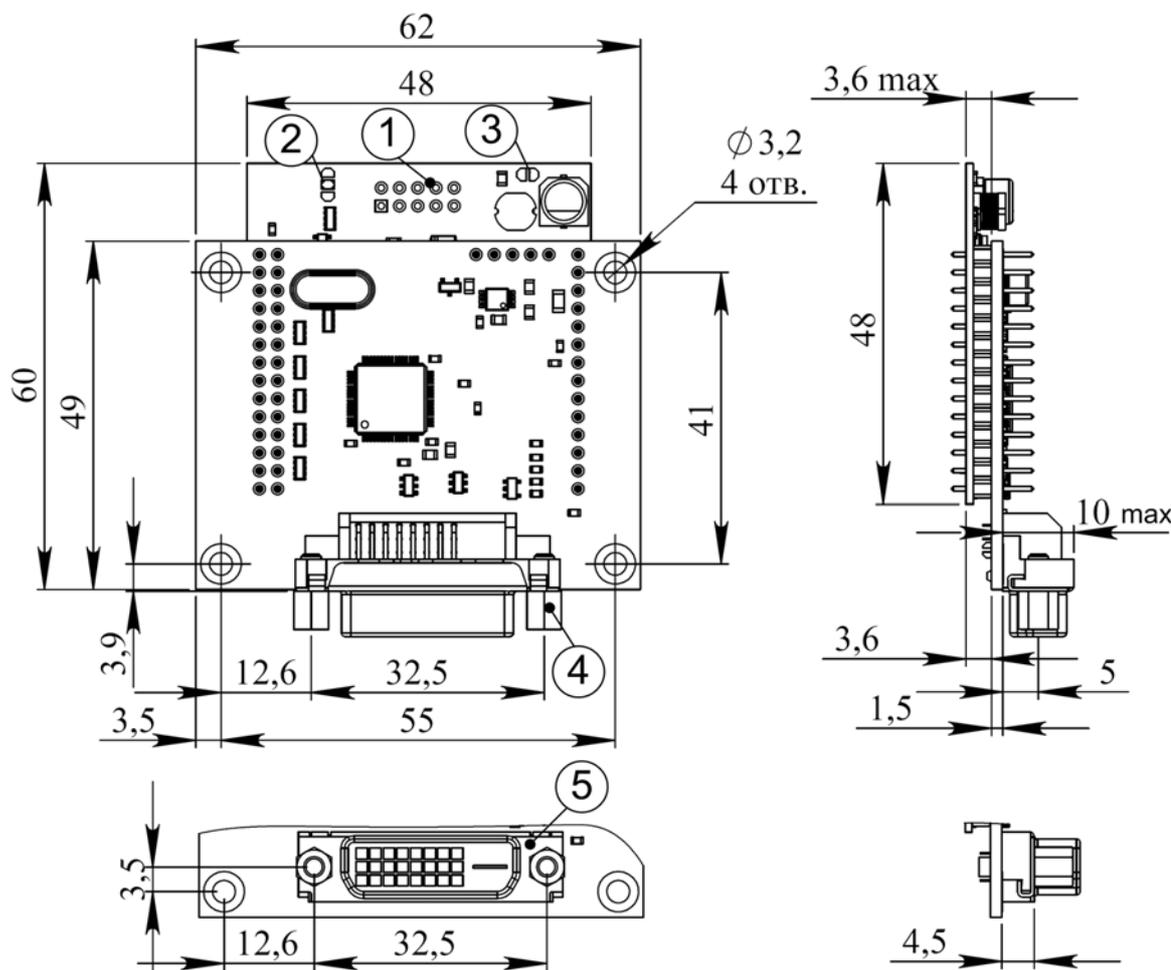


Рис. 7. Печатная плата TFT-Компаньона DVI

На рисунке: 1 – 10-контактный двухрядный разъем связи с пользовательской системой. Шаг выводов 2.54 мм. 2 – Перемычки J1 и J2 на рис. 9. 3 – перемычка для защиты флэш-памяти от изменения. Замкнута – запись запрещена. 4 – Горизонтальный разъем DVI-D для подключения к монитору. 5 – Вертикальный разъем DVI-D для подключения к монитору (опция).

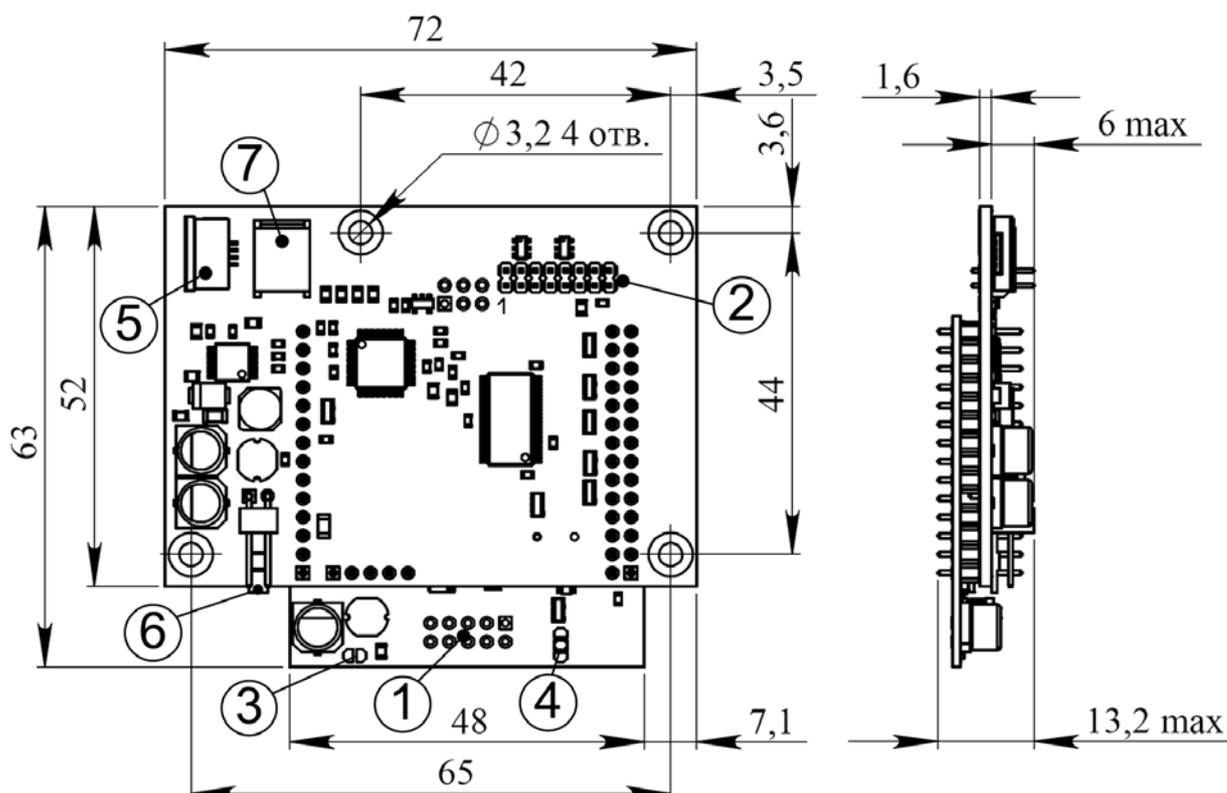


Рис. 8. Печатная плата для TFT-модулей с интерфейсом LVDS

На рисунке: 1 – 10-контактный двухрядный разъем связи с пользовательской системой. Шаг выводов 2.54 мм. 2 – Разъем для подключения TFT-модуля. 3 – перемычка для защиты флэш-памяти от изменения. Замкнута – запись запрещена. 4 – Перемычки J1 и J2 на рис. 9. 5 – разъем для подключения тач-панели TFT-модуля. 6 – разъем для подключения питания DC/DC преобразователя LED-подсветки. 7 – разъем для подключения LED-подсветки TFT-модуля.

10 Рекомендации по включению

Пример подключения TFT-компаньона представлен на рис. 9.

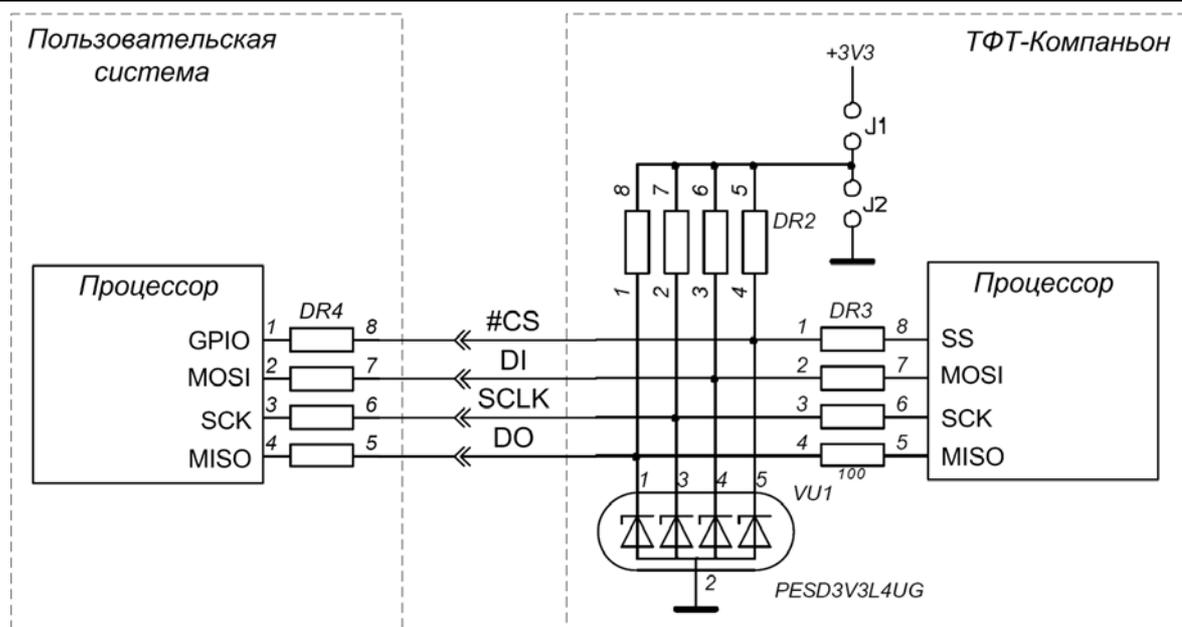


Рис. 9. Схема входа ТФТ-Компаньона. Пример включения.

Используя наборы резисторов DR2 и DR4 с замкнутой перемычкой J2 можно реализовать делитель напряжения, и использовать для управления 5-вольтовые сигналы.

Резисторы DR2 и перемычка J1 или J2 позволяют повысить помехоустойчивость, а при необходимости и согласовать линию. Благодаря этому шлейф между пользовательской системой и ТФТ-Компаньоном можно сделать достаточно длинным. По умолчанию DR2 не устанавливается на плату.

При использовании короткого шлейфа и 3.3 В уровней в пользовательской системе, резисторы DR2 и DR4 не нужны.

11 Изменения

27.01.08 Начальная версия

14.05.08 Изменены номера выводов входного разъема (табл. 1)

27.05.08 Дополнены данные для версий 5.7" и 7"

05.06.08 Изменен формат команды SaveBitmap (31h). Команды SaveBitmap (31h), DelBitmap (32h), DrawBitmap (30h) и GetBitmapPar (34h) переименованы соответственно в SaveImage, Dellmage, DrawImage и GetImagePar.

24.06.08 Добавлена работа с JPEG файлами.

Удален параметр S из команды установки шрифта 40h.

Добавлено описание работы со шрифтами в формате Microchip GUI.

06.10.08 Добавлены детали описания команды SetSpeedGrade (60h).

23.10.08 Добавлен режим пониженного энергопотребления, команды PowerDown (63h) и PowerUp (64h).

Добавлен бит PD в байт состояния.

В характеристиках добавлены данные по потреблению в режиме пониженного энергопотребления.

Добавлены команды GetCursorXY (3Bh), GetTextXY (3Ch), GetStringLen (4Bh).

Добавлено время выполнения команды опроса тач-панели GetTouchStat (50h).

16.02.09 Добавлено описание DVI-версии, функции DVISetRes (68h) и чертеж модификации для 5.7" и 7" модулей.

13.07.09 Добавлена LVDS-версия TFT-Компаньона и характеристики для TFT-модуля MI1040GT-1.

12 Ссылки

1. Страничка TFT-Компаньона: www.schemov.com/tftcomp/
2. Программа просмотра/преобразования графических файлов IrfanView
www.irfanview.com
3. Страничка TFT-Компаньона DVI: www.schemov.com/tftcdvi/