

Описание дисплея от Siemens S65

Автор **Christian Kranz**, октябрь 2005

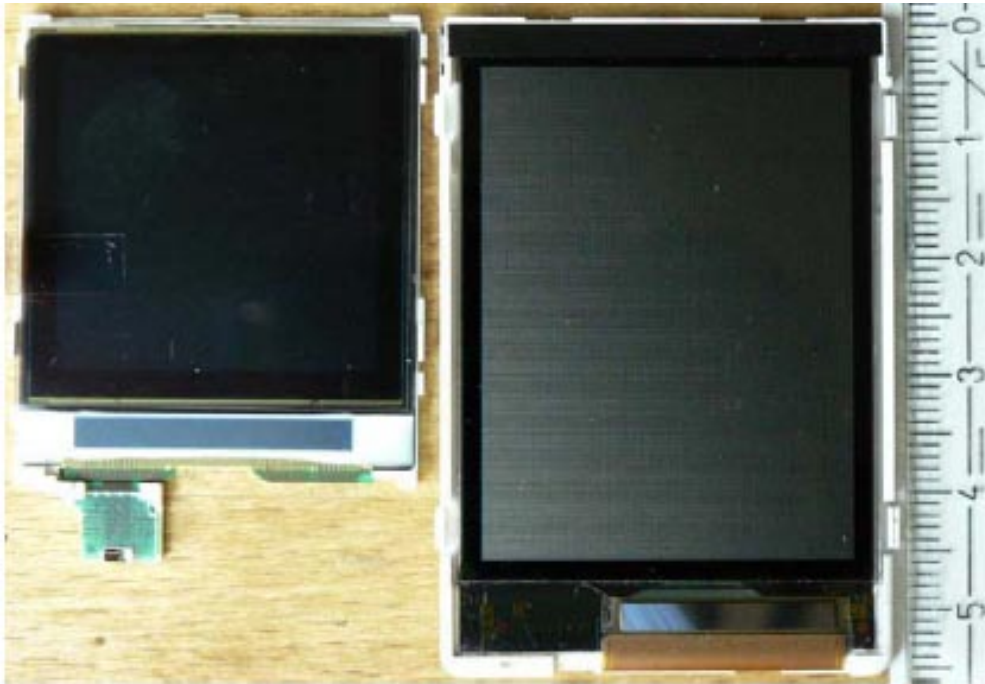
(http://www.superkranz.de/christian/S65_Display/DisplayIndex.html)

Перевод EfremKos & Ведущий специалист (v0.5 / 25.12.2007)

(<http://vrtp.ru/index.php?showtopic=1120&st>)

Предисловие:

Инженеры компании Siemens использовали для мобил S65, M65, CX65 и SK65, 16-битный цветной TFT дисплей, с разрешением 132x176 пикселей. В сервисных центрах и интернет магазинах он стоит менее 20 €. На фотке, дисплей от Nokia 6100 выглядит неприметно.



Теперь опишу, как использовать этот дисплей в собственных проектах.

Фактически это подключение к Atmel AVR процессору, из описания понятно как подключать его к другим устройствам.

Информация:

Известны три различных типа Дисплея. На фотке ниже две версии. К счастью они совместимы с точки зрения программного обеспечения и аппаратных средств. Третий дисплей мне неизвестен.



Железо:

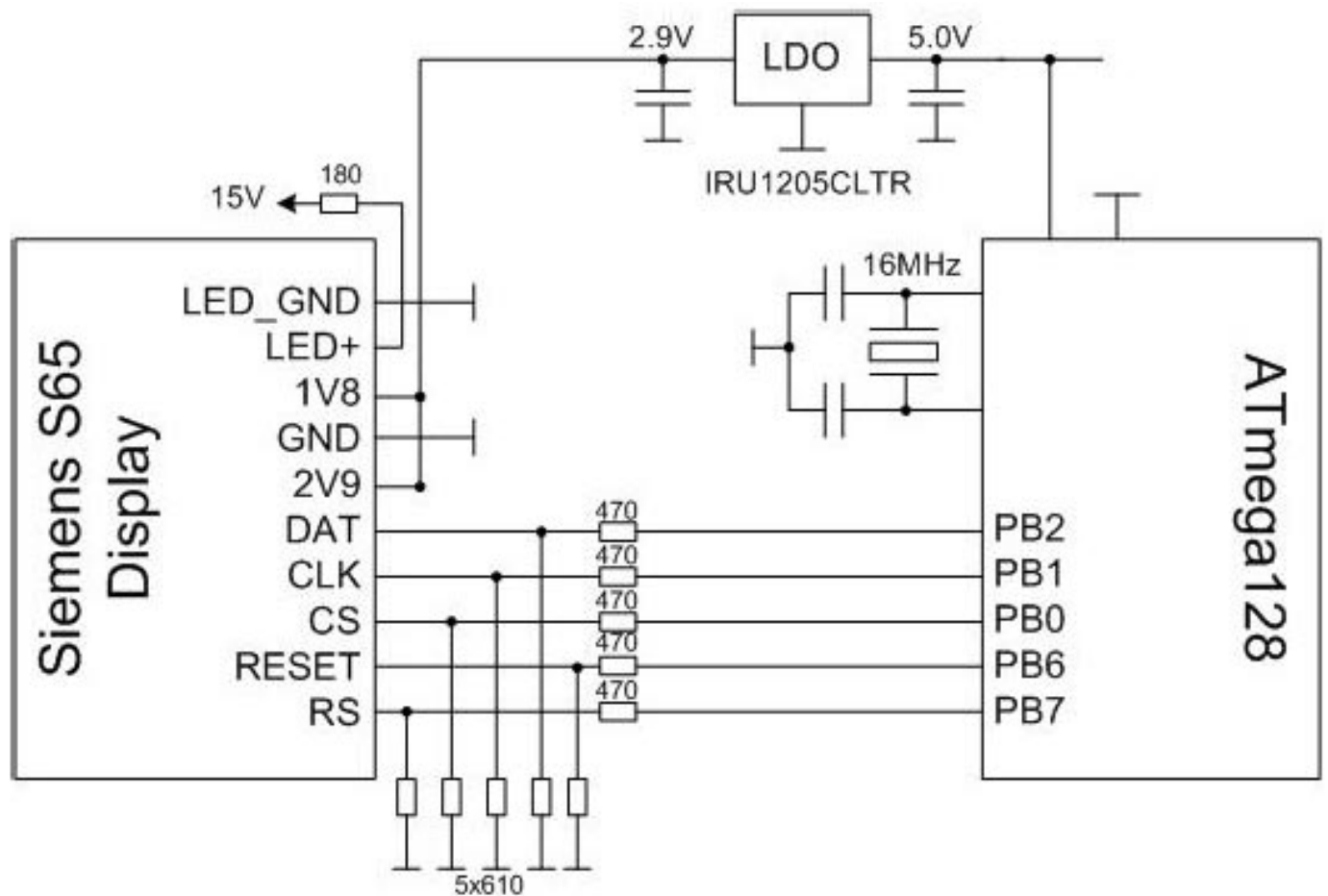
Цоколёвка:



Схема:

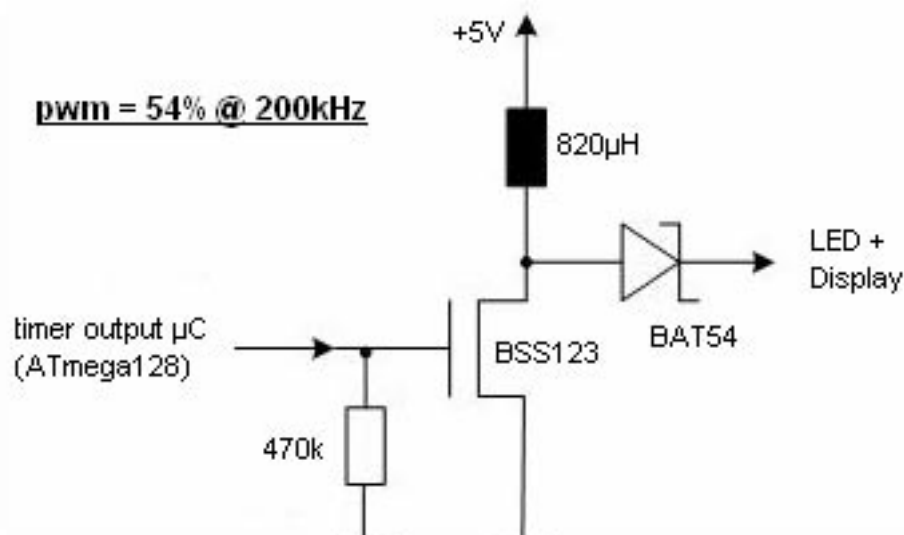
Тестовая схема будет дальше. На оба входа питания подаётся напряжение 2.9V, через понижающий стабилизатор. Я использовал IRU1205CLTR, потому что в нем есть функция вкл\выкл питания, которая мне позже потребуется. Тактовая частота Atmeg'i 16MHz.

Пять сигналов интерфейса между дисплеем и МК понижены резисторами до рабочего напряжения дисплея, 2.9V. Поскольку сигнал будет идти с частотой 8MHz, сопротивление подтягивающих резисторов должно быть низким, чтобы получить устойчивый сигнал. Это хорошая идея, обойтись резисторами, дабы сэкономить питание, специально для сигнала RESET, потому что его уровень должен быть высоким, пока дисплей работает. С другой стороны можно использовать более высокое сопротивление на ресете, потому что скорость ему не важна.



Подсветка:

Максимальное напряжение подсветки 10.4V при 20mA. Значит необходим относительно высокий источник напряжения. Напряжением можно управлять при помощи широтно-импульсной модуляции с МК. Не забудьте ограничить $I_{пит}$, пока МК не выдаёт ШИМ. Если вы откроете NPN транзистор на долго, то будет коротыш. Подсветке требуется длительность импульса 54 % при частоте 200kHz, чтобы поддерживать необходимое напряжение при токе 20mA. Сглаживающий конденсатор для подсветки уже есть на плате дисплея, его достаточно.



Команды и данные:

Данные или команды поступают от МК, указывает сигнал RS.

Если сигнал RS высокий, драйвер дисплея ожидает команды, если сигнал RS низкий - то принимает данные.

RS=1 команды
RS=0 данные

Команды инициализации:

Дисплей, инициализируется в три этапа. Чтобы включить его, необходима правильная последовательность команд, для дальнейшего управления драйвером дисплея. Это необходимо из-за большого кол-ва пикселей дисплея, относительно высокое напряжение используется для управления экраном. Для работы жидких кристаллов требуется напряжение порядка 10 - 20 вольт. А напруга в таких широких диапазонах зависит от температуры окр.среды. Но это уже заботы контроллера, он сам будет регулировать напряжение.

Первая из двух используемых инициализаций:

INIT1:

0xFDFD, 0xFDFD
0xEF00
0xEE04, 0x1B04
0xFEFE, 0xFEFE
0xEF90, 0x4A04, 0x7F3F, 0xEE04, 0x4306

Так вот для того чтобы получить это напряжение надо программно включить внутренний dc-dc преобразователь и подождать 7 миллисекунд пока он запустится. Это значение будет использоваться во всех последовательностях.

После запуска дисплей может работать.

Семь оригинальных команд инициализации от дисплея S65 к счастью могут быть уменьшены до двух. Кроме того мы не нуждаемся в очень длинных командах, которые можно подать на наш Siemens.

INIT2:

0xEF90, 0x0983, 0x0800, 0x0BAF, 0x0A00, 0x0500, 0x0600, 0x0700
0xEF00
0xEE0C
0xEF90, 0x0080
0xEFB0, 0x4902
0xEF00
0x7F01, 0xE181
0xE202

После инициализации дисплей готов к работе. Рекомендуется очистка экрана, потому что дисплей показывает случайное содержимое после первой подачи питания.

Команда для выключения:

Для нормального отключения дисплею требуется специальная команда, чтобы очистить шум в памяти драйвера. Иначе жизнь дисплея сократится. В настоящее время инициализация отключения не известна. Поэтому мы можем только перезагрузить дисплей и отключить питание.

Запись полного экрана и его части:

После инициализации дисплей готов к работе. Обычно первое действие - очистка дисплея. Очистить его можно следующей командой:

MEMWR:

0xEF90, 0x05**OR**, 0x06**YS**, 0x07**XS**

После чего можно записать данные в каждый 16-битный пиксель дисплея.

Параметры **XS** и **YS** определяют координаты вывода первой точки в дисплей. дальнейшая запись будет инкрементировать XS на 1.

Если нужно заполнить только часть экрана, то данные записывают, далее может быть останов, и команда MEMWR сбрасывает записываемые данные на эти координаты.

Полный экран начинается с **XS = 0** и **YS = 0**. В этом случае повторять координаты не обязательно. Достаточно писать данные снова и снова. Память дисплея организована (?) согласно картине на правой стороне. Всего байтов 132x176x2, в hex формате это 0xB580 пикселей. Память заполняется сначала в X, а потом в Y - direction. Параметр **OR** определяет ориентацию экрана.

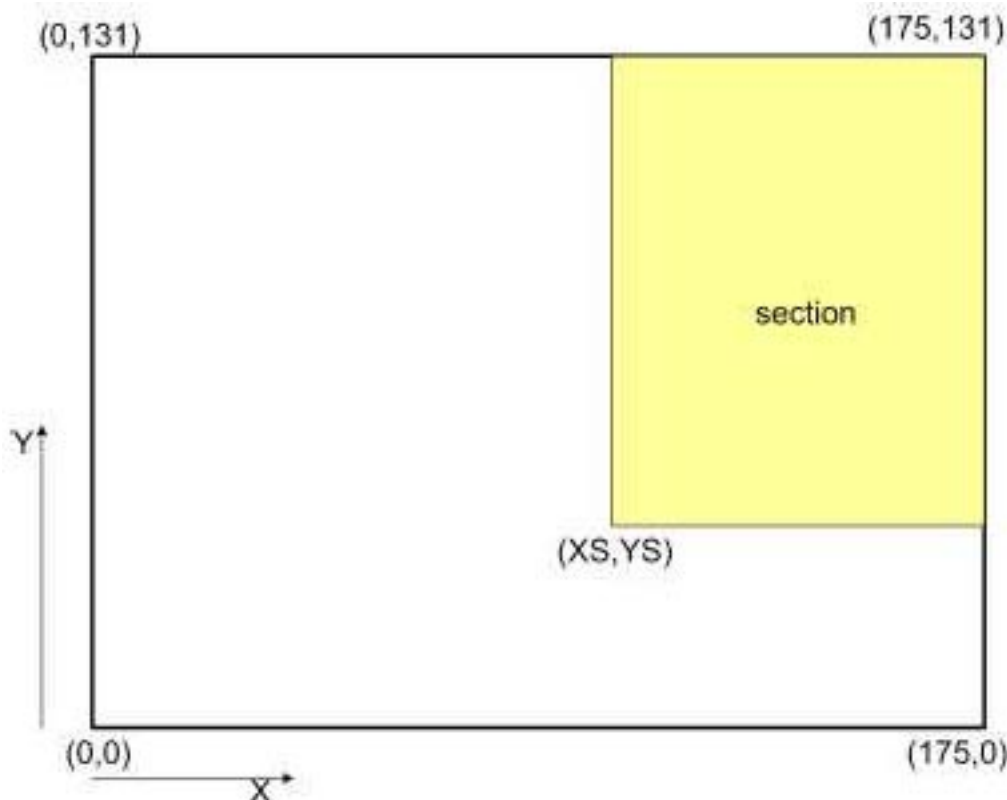
OR = 0x04 устанавливает **горизонтальную ориентацию изображения**, показанного на картинке ниже.

OR = 0x00 поворачивает экран **на 90 градусов**, вертикальная ориентация.

Частичное использование экрана.

Для **OR = 0x04** данные пишутся в память x-direction, что означает вывод инфы линия за линией в x-координате с лева на право, то есть от X малого до большего X.

Для **OR = 0x00** заполняется память в Y-координате, от **YS** до **131**, колонна за колонной.

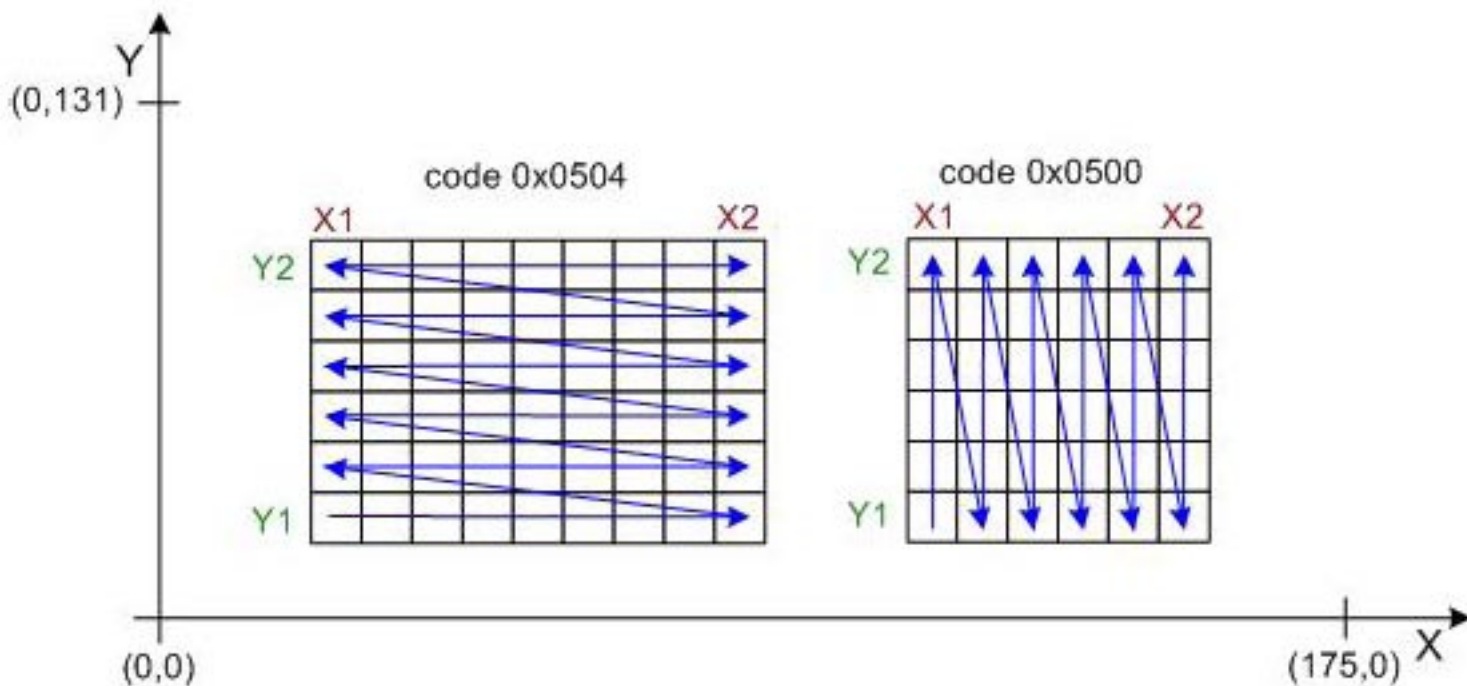


Настройка цвета пикселя:

Дисплей работает в -5-6-5- (-R-G-B-) режиме цветопередачи. Это означает, что в потоке данных первые 5 бит устанавливают красный цвет, следующие 6 - зеленый, и последние 5 синий. (0xFFFF белый, 0x0000 черный)

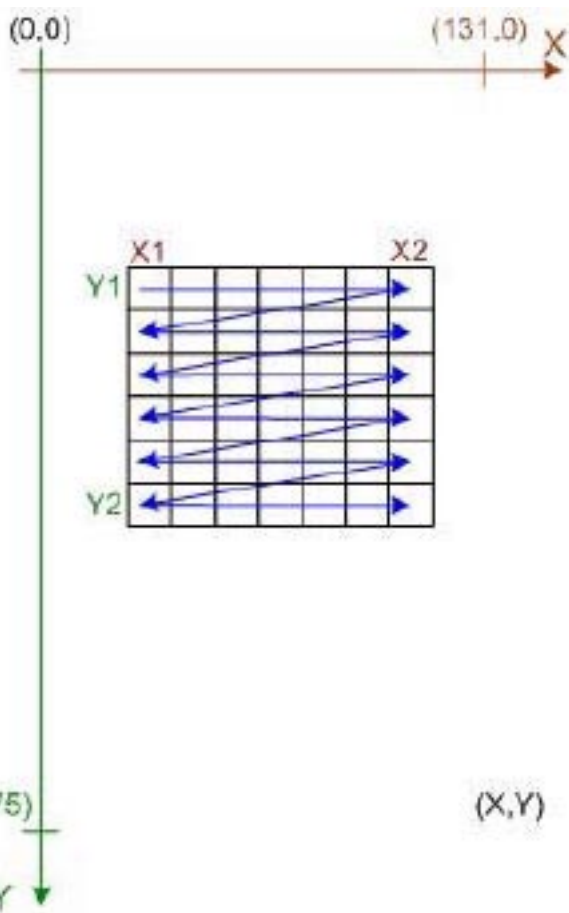
Частичная запись в дисплей:

Память дисплея имеет матричную структуру с X- и Y осями координат. В каждой матричной координате хранится одно значение (пиксель) 16-разрядного цвета. Усовершенствованные методы подобно PET(интерполяционная адресация) не поддерживаются.



На рисунке показан принцип вывода изображения:

Ниже используются команды для частичной записи в память. координаты начала (X1, Y1) и конца (X2, Y2) должны быть определены. X2 должен быть больше или равняться X1, Y2 должен быть больше или равняться Y1.



PMEMWRX: (X-direction)

0xEF90, 0x0504, 0x08X1, 0x09X2, 0x0AY1, 0x0BY2

PMEMWRY: (Y-direction)

0xEF90, 0x0500, 0x08X1, 0x09X2, 0x0AY1, 0x0BY2

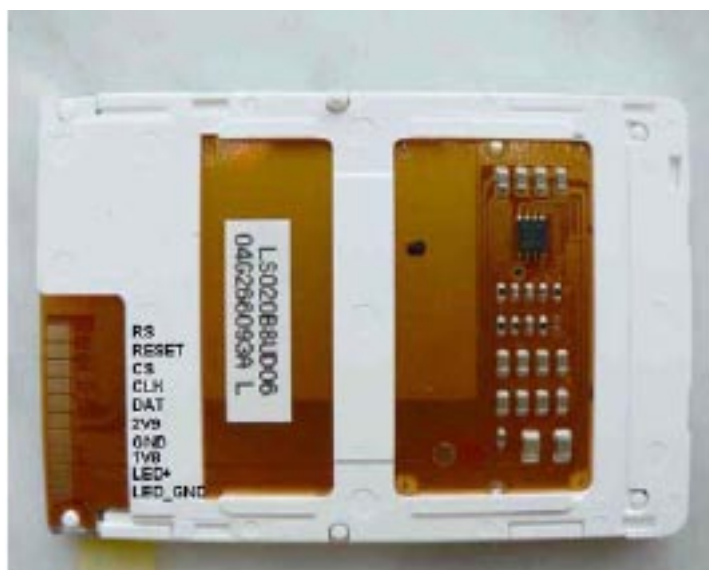
Данные в память дисплея пишутся в X или Y направлении. Другие режимы доступа к памяти не поддерживаются.

Запись в Y-направлении используется если дисплей повернут на 90 градусов. Если X-и Y-значения в команде PMEMWRY поменять местами, то получится система координат показанная на левом рисунке. В этом случае битовые массивы идут от левого верхнего угла до до нижнего правого.

Изучение S65 дисплея с нуля :

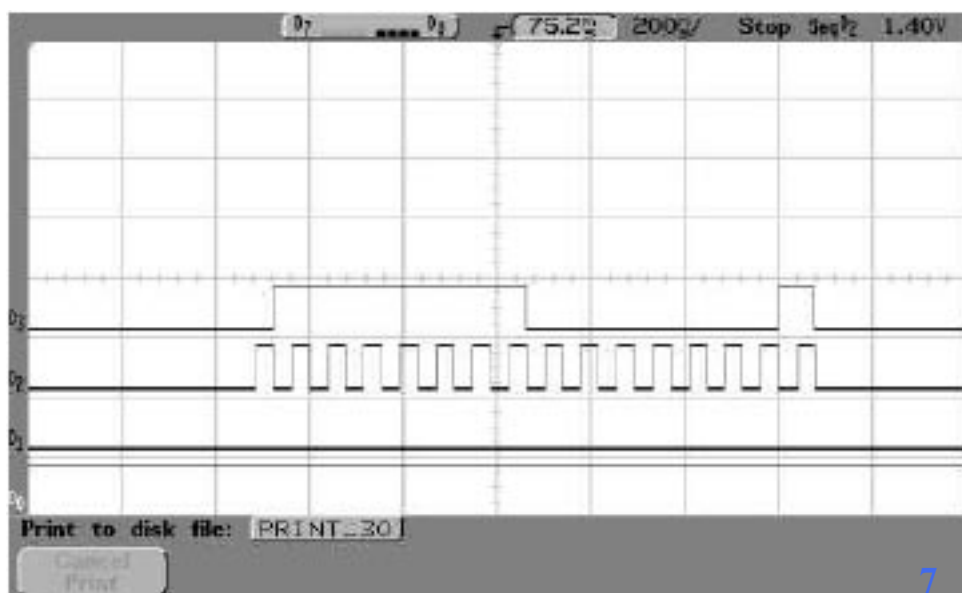
Дисплей от Siemens S65 - специально разработан для телефонов Siemens/BenQ. Поэтому для него очень сложно достать data sheets. Изготовлен он Seiko-Epson, но гад не отвечает на известные команды дисплеев Epson. Из-за этого я решил заново разработать интерфейс дисплея, используя мобильный телефон S65.

Нетрудно догадаться что в Siemens применили последовательный трёхпроводной интерфейс SPI с дополнительным выбором чипа, сигнала выбора команд/данных (RS) и сигналом RESET. Тактовая частота интерфейса - 13MHz. Напряжения сигнала - 2.9V для драйвера, и 1.8V для цифрового интерфейса. Кроме того есть два контакта для питания четырех белых светодиодов. LED'S включены последовательно, и по их питанию стоит сглаживающий конденсатор. Последовательное включение гарантирует ровное освещение. Плохо что для питания LEDs требуется большое напряжение , 10.4V 20mA. Но глядя на наше железо, вы увидите, что не трудно произвести ШИМ, с помощью МК. Цоколёвка дисплея слева.



Сигналы меж дисплеем и мобилой я увидел, с помощью 16 канального цифрового осциллографа HP Agilent. (к сожалению логического анализатора под рукой не оказалось). Сигналы были сняты в 16-битном формате как показано на скриншоте. Бит данных запирается с фронтом тактового сигнала контроллера дисплея.

RS - выбор, команды или данные
RESET - сброс
CS - выбор кристалла (для нескольких SPI устройств)
CLK - тактовой сигнал
DAT - линия данных
2V9 - питания драйвера
GND - масса
1V8 - питание линии интерфейса
LED+ - питание подсветки
LED_GND - минус подсветки



Последовательность для запуска:

Данная последовательность команд инициализирует дисплей.

Sequence 1: (time 0ms)

0xFDFD, 0xFDFD

Sequence 2: (time 68ms после старта)

0xEF00, 0xEE04, 0x1B04, 0xFEFE, 0xFEFE
0xEF90, 0x4A04, 0x7F3F, 0xEE04, 0x4306

Sequence 3: (time 75ms после старта)

0xEF90, 0x0983, 0x0800, 0x0BAF, 0x0A00, 0x0500, 0x0600, 0x0700
0xEF00
0xEE0C
0xEF90, 0x0080,
0xEFB0, 0x4902
0xEF00
0x7F01, 0xE181

Sequence 4: (time 182ms после старта)

0xE202

Sequence 5: (time 297ms после старта)

0xE276

Sequence 6: (time 401ms после старта)

0xE183

Sequence 7: (time 460ms после старта)

0x8001

Кроме того выяснилось, что Siemens всегда обновляет экран сразу. Значит, если часть экрана изменена, то обновляется вся память дисплея. Потому что Siemens использует графический драйвер s1d13732. Этот драйвер имеет память, она и есть память дисплея, и пишет туда данные, в нашем случае с внешнего МК, который выдаёт только команды.

Програмисты Siemens используют следующую последовательность для заполнения дисплея.

Полное заполнение экрана:

0xEF90, 0x0500, 0x0600, 0x0700

После этого принимаются данные дисплея (с низким RS). Запись данных происходит каждые 28.59ms, что соответствует $132 \times 176 \times 16 = 371712$ бит при частоте 13MHz.

Отключение дисплея:

Команда выключения не измерена с моим оборудованием. Не было условий для снятия стабильного лог при отключении. Доступ к подсветке возможен, но память моего осцила мала, чтобы поймать команду. К сожалению подсветка вырубается сразу после дисплея. Посторавшись можно уловить, но лог так и не снят . Пока единственный алгоритм выключения дисплея - ресет, и отключение питания.

П.С.

С замеченными ошибками обращайтесь в аську 440261243, или форум. Исправлю.
<http://vrtp.ru/index.php?showtopic=1120&st>