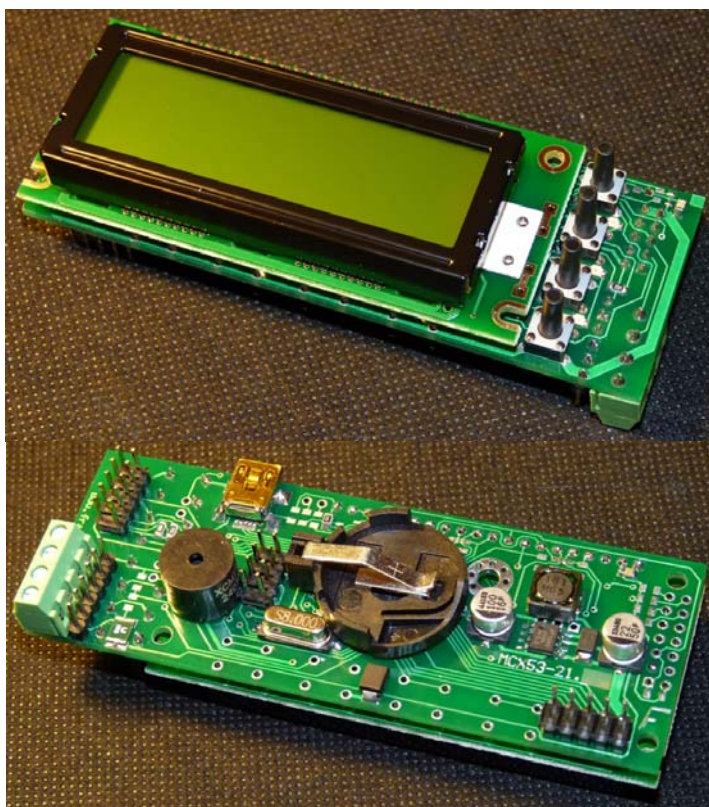


# Терминальный модуль МСХ53-21.21



**Фирма Фрактал**  
Москва Зеленоград  
[www.fractal.com.ru](http://www.fractal.com.ru)  
[fractal@aha.ru](mailto:fractal@aha.ru)  
(495) 978-12-86

## **Список добавлений и исправлений документа.**

### **5.02.2014 v1.07**

Начиная с версии 3-02e Fractal-BASIC-Cortex, для операторов PWM стал доступен TIM5.  
При работе с ШИМ в этом модуле это самый удобный канал таймера. См. таблицу функций выводов.

### **22.03.2013 v1.06**

Исправлено указание о требуемом размере массива при работе с использованием интерпретатора .  
Требуется объявление массива на 10 элементов, а не 6.

### **18.09.2012 v1.04**

Добавлено замечание о необходимой паузе между поступающими командами. В зависимости от типа команды пауза может составить 0...20мс .

### **21.04.2012 v1.03**

Добавлен демо режим. Раздел 9.

### **04.04.2012 v1.02**

Введена поддержка канала SPI.

Добавлены в таблицу разъема X2 номера линий при инициализации их как линий общего назначения.

### **20.02.2012 v1.01**

Создание документа.

### 1. Назначение.

Модуль предназначен для отображения алфавитно-цифровой и графической информации и ввода информации с кнопок. Может использоваться как законченное терминальное устройство или как управляющее(ведущее) устройство.

Модуль включает в себя графический ЖКИ 122x32 с подсветкой, 4 кнопки, малогабаритный динамик, часы реального времени с литиевой батареей, модуль питания, 12 универсальных линий ввода-вывода.

Модуль поддерживает работу с интерфейсами USB, RS485, I2C, SPI, MicroLan .

Основой MCX53-21.21 является микроконтроллер ARM7 Cortex-M3 STMicroelectronics STM32F103, который обеспечивает работу с последовательными каналами и работу с ЖКИ, кнопками, звуком и линиями ввода-вывода.

Применение этого функционально насыщенного и содержащего большие объемы памяти микроконтроллера, позволило предустанавливать в модуль MCX53-21.21 однокристалльный интерпретатор Fractal-BASIC-Cortex. Интерпретатор дает пользователю большие возможности простого использования ресурсов модуля.

При работе с ЖКИ, для вывода символов обеспечиваются эмуляция символьного режима 20 символов на 4 строки, символы двойной высоты/ширины в разных режимах вывода.

Для графики реализован вывод точек/линий/прямоугольников в разных режимах вывода.

Яркость подсветки и контрастность ЖКИ программно регулируются , что создает дополнительные удобства при отображении информации.

Питание +5V+-5% или + 8...30V.

### Краткие технические характеристики:

- ✓ Микроконтроллер STM32F103RET6 / STM32F103RGT6
  - Быстродействие 72 MHz / 1.25 DMIPS/MHz (Dhrystone 2.1)
  - Flash 512 Кбайт / 1024 Кбайт
  - RAM 64 Кбайт / 96 Кбайт
  - Отладочный разъем Serial wire debug (SWD)
- ✓ Встроенный однокристалльный Fractal-BASIC-Cortex
- ✓ Интерфейсы без развязки - USB, RS485, I2C/SPI, MicroLan
- ✓ Предустановленные параметры RS485 - 115200 bps, 8бит/без четности/1стоп
- ✓ Скорость работы I2C - 400кГц
- ✓ ЖКИ - WG12232E
- ✓ Разрешение ЖКИ - 122x32
- ✓ Подсветка - да
- ✓ Кнопки - 4
- ✓ Универсальные линии ввода-вывода -12
  - АЦП 12 бит - 4(8)
  - ЦАП 12 бит - 2
  - ШИМ 16 бит - 4
  - Счетчик / энкодер - 1
  - линия MicroLan - 4(8)
- ✓ Часы реального времени с литиевым элементом CR2032
- ✓ Светодиодная индикация – Работа, прием, передача
- ✓ Питание +5V+-5% или + 8...30V
- ✓ Габаритные размеры 99x36x25 мм

## 2. Состав и функционирование.

MCX53-21.21 можно условно разделить на следующие составные части:

- Микроконтроллер с резонаторами и стабилизатором;
- группа интерфейсов USB, RS485, I2C(SPI), (MicroLan);
- графический ЖКИ WG12232E с подсветкой;
- кнопки;
- звук;
- линии ввода-вывода
- отладочный разъем;
- разъем межмодульного соединения;
- элемент питания для RTC;
- импульсный стабилизатор;
- светодиодная индикация.

### Микроконтроллер с резонаторами и стабилизатором

В модуле применен микроконтроллер ARM7 Cortex-M3 STM32F103RET6(GET6).

Для тактирования основного генератора используется кварц 8.000МГц. Из этой частоты микроконтроллер синтезирует нужные тактовые частоты, вплоть до 72 МГц. В нем так же имеется прецизионный RC-тактовый генератор на 8 МГц. При необходимости можно им воспользоваться.

Для тактирования часов реального времени используется отдельный кварц на 32768Гц.

Для питания микроконтроллера использован отдельный стабилизатор 3.3В, входным питанием которого служит +5В.

В микроконтроллере имеются встроенный температурный датчик и источник опорного напряжения 1.2В. Они доступны пользователю из Fractal-BASIC-Cortex.

### Группа интерфейсов

Для работы с USB на контроллере установлен mini-USB-B разъем.

USB рассматривается, в основном, как терминальный канал для занесения BASIC-программ, занесения параметров, приема накопленных данных, обновления версий интерпретатора и отладки.

При подключении к PC MCX53-21 будет виден как VCP -> виртуальный COM –порт (предварительно необходимо установить соответствующий драйвер).

Драйвер VCP под Windows доступен на нашем сайте или на сайте STMicroelectronics.

Для Linux специальный драйвер VCP не требуется, он должен быть просто разрешен в ядре, но обычно он разрешен к использованию по умолчанию.

Параметры канала по умолчанию : скорость 115200 бод, 8 бит, 1 стоп бит, без четности, аппаратный контроль.

Канал RS485 выведен на две группы разъемов: на разъем короткого межмодульного соединения и винтовые клеммы. Контроллер может работать и в качестве master и в качестве slave устройства в сети MODBUS.

Параметры канала по умолчанию : адрес MODBUS =2, скорость 115200 бод, 8 бит, 1 стоп бит, без четности.

Список поддерживаемых команд MODBUS приведен в описании интерпретатора Fractal-BASIC-Cortex.

В Fractal-BASIC-Cortex этот канал фигурирует как канал 0(UART1).

В контроллере установлены 3 джампера, которые в случае необходимости подключают подтяжки и терминатор.

Канал I2C выведен на короткого межмодульного соединения. Используемые линии могут работать и в режиме канала SPI, но по умолчанию линии инициализированы как I2C.

Канал I2C может работать и в режиме slave и в режиме multi-master. Мастер режимы обеспечены командами встроенного интерпретатора. При их помощи пользователь может получить простой доступ к подключаемым I2C-slave устройствам(датчикам, ЦАП, АЦП...).

В режиме slave I2C канал позволяет получить доступ ко всей пользовательской RAM. Мастер устройство может практически полностью управлять модулем чтение/запись RAM.

Доступ к RAM осуществляется наиболее часто употребляемым способом, аналогично принятому в модулях MCU4:

- первый байт содержит slave-адрес и признак записи / чтения;
- второй байт при записи расценивается как адрес байта в RAM(адрес слова – word – адрес);
- чтение производится применением совмещенного цикла запись-чтение, где сначала указывается адрес.

Поскольку word-адрес имеет размер всего один байт, для доступа к большому объему RAM сделана возможность записи смещения к этому адресу путем записи величины смещения в специальный страничный регистр(указатель).

Причем используется два разных указателя – отдельно для записи и отдельно для чтения. Это позволяет при необходимости «развести» буфера чтения и записи в разные области.

После сброса оба указателя равны нулю, что означает - выбранные страницы записи и чтения совпадают и расположены в самом начале пользовательской области.

Word-адрес указателя для буфера записи = 0.

Word-адрес указателя для буфера чтения = 4.

Для доступа к управляющим регистрам(указателям) используется отдельный slave-адрес. Другими словами модуль отвечает на два разных slave - адреса.

Заводские установки – для доступа к данным slave - адрес 0x02, для доступа к управляющим регистрам 0x04.

Пользователь может изменить эти адреса (адреса могут быть только четные!) :

Для установления адреса доступа к данным нужно вызвать оператор I2C#256,(0) = NewAdr

Для установления адреса доступа к регистрам нужно вызвать оператор I2C#258,(0) = NewAdr

Канал SPI выведен на короткого межмодульного соединения. Используемые линии после сброса проинициализированы как канал I2C. Для перевода их в режим SPI, достаточно обратиться к оператору SPI. Канал программно эмулируется, но пользователь при работе из Fractal-BASIC не заметит разницы с другими модулями, где использован аппаратный узел, в том числе и по скорости. Частота тактирования SCK при обмене в пределах 1-2 МГц.

#### Графический ЖКИ WG12232E с подсветкой

В модуле устанавливается графический ЖКИ WG12232E или аналогичный с SED1520 - совместимым контроллером разрешением 122 \* 32 точки. Индикатор подключен по параллельному каналу.

Поддерживается и вывод на ЖКИ прямо из интерпретатора(например оператором PRINT), и обработка команд приходящих по интерфейсам RS485 и I2C (полностью аналогично другим нашим терминальным модулям).

Обеспечивается вывод символов в символьном режиме(по знакоместам), в графическом режиме(в любой точке экрана). Доступны символы двойной ширины / высоты (+).

Можно выводить точки, линии, прямоугольники, заранее сохраненные графические фрагменты.

Все это при полном наборе режимов вывода: перо белое/черное, забой, наложение, инверсия и т.д.

Для символьного вывода из PRINT нужно указать адрес #16.

Доступ к графическим функциям осуществляется соответствующими операторами BASIC.

Так же пользователь может программно устанавливать яркость подсветки и требуемую контрастность.

#### Кнопки

Установлено 4 кнопки. По интерфейсам RS485 и I2C доступен регистр 8 бит с их состоянием после устранения дребезга.

#### Звук

На модуле установлен малогабаритный звуковой излучатель. Из BASIC или по интерфейсам можно задавать тон и длительность сигнала.

#### Отладочный разъем.

!!! обратите внимание, что попытка стирания битов конфигурации или записи Flash памяти микроконтроллера при использовании отладочных средств или через Boot-loader STMicroelectronics приведет к полному стиранию Flash и потере не только Fractal-BASIC-Cortex, но и Boot-loader Fractal. Восстановление будет возможно только у нас.

Разъем – джамперное поле установлен недалеко от разъема USB.

Разъем скомпонован их 4-х джамперов. Джамперы расположены так чтобы минимальным числом контактов удобно использовать функции сброса, задания BOOT-режима, работы с SWD –отладчиком.

Для запуска Boot-loader Fractal и обновления версии Fractal-BASIC-Cortex необходимо перед подачей питания замкнуть J3. Сама процедура обновления описана ниже. SWD отладчик подключается к J4 и J1.

Для повышения надежности нормальная работа модуля выбрана когда отсутствуют все джамперы.

Для сброса нужно кратковременно замкнуть J1.

Для выбора Boot-loader STMicroelectronics J2 должен быть замкнут в момент сброса.

#### Разъем короткого межмодульного соединения.

На этот разъем выведены: сигналы локальных шин I2C / SPI , сигналы дублирующие клеммы гальванически не развязанного канала RS485 и питание +5B & GND. SPI реализован программно.

При необходимости, 4 линии отведенные под I2C/SPI могут быть использованы как линии общего назначения.

#### Элемент питания для RTC

Для питания часов реального времени и backup регистров в отсутствие основного питания используется литиевый элемент. Держатель элемента CR2032 расположен над микроконтроллером. Элемент устанавливается плюсом вверх – на контакте держателя есть обозначение «+».

**Импульсный стабилизатор(питание узла).**

Для питания МСХ53-21 требуется постоянное напряжение в диапазоне 8...30В. В зависимости от партии контроллеров могут устанавливаться разные типы импульсных стабилизаторов с идентичными параметрами. На выходе стабилизатора получается +5В. Это основное напряжение используемое большинством узлов контроллера.

**На модуле установлено три светодиода.** Два красный и желтый - это прием / передача по внешним интерфейсам UART1, UART3, USB.

Оставшийся зеленый указывает на что микроконтроллер начал выполнение программы.

**3. Назначение выводов разъемов**

Разъем X1 (USB)		
Номер вывода	Имя цепи	Описание
1	U_BUS	Питание USB +5В
2	D-	DATA -
3	D+	DATA +
4	NC	не подключен
5	GND	Земля 0В

Разъем короткого межмодульного соединения X2			
Номер вывода	Имя цепи	Описание	BASIC PIN,LAN
1	+5B	Питание +5В	
2	+5B	Питание +5В	
3	X+	RS485 X+ канал межмодульного соединения (X4)	
4	SCL	Резервный канал I2C (SPI -> SCK)	12
5	X-	RS485 X- канал межмодульного соединения (X5)	
6	SDA	Резервный канал I2C (SPI -> SDI)	13
7	CS	Резервный канал SPI	15
8	SDO	Резервный канал SPI	14
9	GND	общий	
10	GND	общий	

RS485		
Номер	Имя цепи	Описание
X4	X+	RS485 X+ или CANH
X5	X-	RS485 X- или CANL
J5	Term	замкнут – терминатор подключен; по умолчанию разомкнут
J6	X+ к +5В	замкнут -> X+ подтянут к +5В через 1кОм; по умолчанию замкнут
J7	X- к GND	замкнут -> X- подтянут к GND через 1кОм; по умолчанию замкнут

Питание		
Номер	Имя цепи	Описание
X6	+8...+30В	напряжение питания +8...+30В
X7	GND	общий

Отладочный разъем / Джамперы (группа отладки)		
Номер	Имя цепи	Описание
J1	RESET	замкнут – сигнал RESET активен; по умолчанию разомкнут; контакт 1 -> RESET ; контакт 2 -> GND
J2	BOOT0	замкнут – сигнал BOOT0 = 1; по умолчанию разомкнут (BOOT0 = 0) ; контакт 1 -> BOOT0; контакт 2 -> +3.3В
J3	BOOT1	замкнут – сигнал BOOT1 = 1; по умолчанию разомкнут (BOOT1 = 0) ; контакт 1 -> +3.3В ; контакт 2 -> BOOT1
J4	SWD	Контакты для подключения SWD -отладчика; контакт 1 -> SWC ; контакт 2 -> SWD

Комбинации джамперов режимов старта		
BOOT1	BOOT0	Режим
0	0	Старт Fractal-BASIC-Cortex
0	1	Запускается заводской Boot-loader STMicroelectronics
1	0	Запускается Boot-loader Fractal



<b>Разъем линий ввода-вывода X3</b>						
<b>Номер вывода</b>	<b>Имя цепи</b>	<b>Имя функции</b>	<b>тип</b>	<b>Описание возможных функций</b>	<b>BASIC PIN,LAN,ADC</b>	<b>BASIC PWM</b>
1	+5B	+5B	POW	Питание +5В		
2	IO0	<b>PA0</b> ADC0 TIM5_CH1	I/O I-AN I/O	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC123_IN0; Таймер 2 линия TIM5_CH1_ETR	0	51
3	IO1	<b>PA1</b> ADC1 TIM5_CH2	I/O I-AN I/O	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC123_IN1; Таймер 2 линия TIM5_CH2	1	52
4	IO2	<b>PA2</b> ADC2 TIM5_CH3	I/O I-AN I/O	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC123_IN2; Таймер 2 линия TIM5_CH3	2	53
5	IO3	<b>PA3</b> ADC3 TIM5_CH4	I/O I-AN I/O	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC123_IN3; Таймер 2 линия TIM5_CH4	3	54
6	IO4	<b>PA4</b> ADC4 DAC1	I/O I-AN O-AN	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC12_IN4; Выход ЦАП DAC1	4	
7	IO5	<b>PA5</b> ADC5 DAC2	I/O I-AN O-AN	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC12_IN5; Выход ЦАП DAC2	5	
8	IO6	<b>PA6</b> ADC6 TIM3_CH1	I/O I-AN I/O	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC12_IN6; Таймер 3 линия TIM3_CH1	6	31
9	IO7	<b>PA7</b> ADC7 TIM3_CH2	I/O I-AN I/O	Цифровой <b>вход/выход</b> порт PA; Вход АЦП ADC12_IN7; Таймер 3 линия TIM3_CH2	7	32
10	GND	<b>GND</b>	GND	Общий 0В		

POW - линии питания;

I/O - вход / выход;

I-AN - аналоговый вход;

O-AN - аналоговый выход.

Та или иная функция линии автоматически выбирается при использовании конкретных операторов / функций Fractal-BASIC-Cortex или путем записи в соответствующие регистры специальных функций необходимых установок.

По умолчанию все линии работают в режиме **входов** портов ввода-вывода.

Функции «по умолчанию» выделены жирным шрифтом. В таблицах представлены не все возможные функции линий ввода-вывода.

!!! все логические линии этого разъема не толерантны к напряжению 5В, максимально возможный потенциал на них относительно земли +3.3В.

Питающее напряжение +5В выведено на разъем только из соображений подключения модулей расширения и их питания.

! в исполнении .21 на линиях IO0-3 установлены резисторы подтяжки 2.2 кОм к +3.3В.

Это полезно при подключении датчиков DS18B20, "сухой контакт " и т.д.

Линии IO4-7 не содержат внешних подтяжек, что позволяет использовать их в режиме с высоким входным сопротивлением, что удобно при использовании АЦП / ЦАП. При необходимости на них могут быть программно включены внутренние подтяжки.



#### **4. Подключение к компьютеру**

Перед первым подключением модуля к системам с операционной системой Windows необходимо установить драйвер виртуального COM-порта STMicroelectronics. Для систем с Linux установка специального драйвера не требуется.

Драйвер VCP для Windows доступен на CD-Fractal, на нашем сайте или на сайте STMicroelectronics.

После этого через кабель USB A-miniB подключить модуль к компьютеру. Операционная система найдет и установит новое устройство. В системе появится VCP -> виртуальный COM порт. В дальнейшем при настройке терминала для работы с модулем, необходимо будет указать этот порт.

Если необходимо, получившийся номер COM порта можно изменить, зайдя в диспетчер устройств и выбрав этот канал. Там в разделе дополнительных установок можно выбрать свободный канал.

При последующих подключениях модуля будет появляться COM порт с этим же номером.

Если будет подключен другой контроллер, то в системе появится другой COM порт.

Параметры канала : скорость 115200 бод, 8 бит, 1 стоп бит, без четности, аппаратный контроль.

!обратите внимание – при запуске Boot-loader Fractal для обновления версии Fractal-BASIC-Cortex или других целей, номер COM порта скорее всего будет иным чем при подключении в режиме Fractal-BASIC-Cortex.

После подачи питания на модуль он на ЖКИ будет выведена информация о исполнении модуля и версии Fractal-BASIC-Cortex. При необходимости эта функция может быть отключена однократной подачей команды EEPROM(0x206)=1 .

#### **5. Программирование**

##### Работа с Fractal-BASIC-Cortex.

В модуле предустановлен однокристалльный интерпретатор Fractal-BASIC-Cortex.

Пользователь может вносить и редактировать программу в режиме командной строки в BASIC-Terminal или другой терминальной программе позволяющей работать непосредственно с COM –портами.

Или загружать заранее подготовленный файл с текстом BASIC –программы при помощи BASIC-Terminal.

! обратите внимание - BASIC-Terminal необходимо запускать после того как MCX53-20 подключен к компьютеру и виртуальный COM-порт системой найден, иначе в BASIC-Terminal соответствующий COM порт будет не виден.

Программа загруженная в модуль может быть запущена директивой RUN.

При подаче питания на модуль происходит автоматический запуск программы занесенной ранее.

Описание Fractal-BASIC-Cortex программирование приведено в отдельном документе.

##### Работа без использования Fractal-BASIC-Cortex.

Пользователь может использовать модуль ступев Fractal-BASIC-Cortex и занеся собственный код.

В этом случае восстановление Fractal-BASIC-Cortex возможно только у нас, процедура бесплатная для устройств изготовленных нами.

Занести собственный код можно через линии SWD -отладчика.

*Программирование через SWD* –отладчик рассмотрим на примере программатора – отладчика ST-Link от ST Microelectronics:

- взять однорядный разъем с семью разноцветными проводами идущими в комплекте ST-Link;
- убрать заглушку на одном из свободных контактов;
- подсоединить разъем к ST-Link так чтобы на первый контакт ST-Link попал красный провод;
- соединить красный провод (ST-LINK function ->Target VCC) с J2 контакт 2;
- соединить оранжевый провод (ST-LINK function ->SW CLK) с J3 контакт 1;
- соединить желтый провод (ST-LINK function ->SW IO) с J3 контакт 2;
- соединить зеленый провод (ST-LINK function ->GROUND) с J1 контакт 2;
- маркировка нанесенная на разъемах не совпадает с перечисленной, т.к. мы используем свой способ

подключения;

- подсоединить ST-Link к компьютеру через USB;
- подать питание на MCX53-32, например через еще один кабель USB;
- запустить ST-Link Utility или среду программирования.

Так, например, среда от IAR поддерживает работу с ST-Link.

#### **6. Обновление Fractal-BASIC-Cortex пользователем**

!!! Внимание! Попытка стирания битов конфигурации или записи Flash памяти микроконтроллера при использовании отладочных средств или через Boot-loader STMicroelectronics приведет к полному стиранию Flash и потере не только Fractal-BASIC-Cortex, но и Boot-loader Fractal. Восстановление будет возможно только у нас.

Во всех продуктах ООО Фрактал на Cortex-M3 пользователь имеет возможность самостоятельно обновить или заменить приложение поставляемое ООО Фрактал.

Для получения обновлений необходимо прислать на адрес [fractal@aha.ru](mailto:fractal@aha.ru) запрос с просьбой включить Ваш адрес в список рассылки обновлений. В письме необходимо указать серийный номер продукта(ов) и e-mail на который будут посылаться обновления. Каждый модуль имеет уникальный номер. Как узнать этот номер описано ниже.

Каждая версия обновлений привязана к конкретному экземпляру модуля, версия обновления для модуля с другим серийным номером не загрузится в модуль. Поэтому если у Вас не один модуль, то обновления придут на каждый заявленный модуль с пометкой серийного номера.

При изготовлении контроллеров MCX53-21.21 в них заносится Boot-loader Fractal (не путать с Boot-loader STMicroelectronics).

Это позволяет пользователю самостоятельно полностью обновить приложение поставляемое ООО Фрактал. В данном случае это Fractal-BASIC-Cortex.

Обновление может быть произведено через USB.

Есть два способа обновления Fractal-BASIC-Cortex.

**Первый и самый простой – воспользоваться утилитой AppSend.exe.** Эта утилита поставляется в составе архива приходящего очередного обновления. Скопируйте содержание архива в отдельную папку и запустите эту утилиту.

Она самостоятельно обновит подключаемые модули, записав в них, соответствующий их серийному номеру, файл.

Если Ваши модули имеют версию Boot-Loader не ниже 1-07, то для обновления не обязательно замыкать J3 перед включением модуля.

**Второй способ – обновление непосредственно через Boot-loader Fractal.** Для запуска Boot-loader необходимо перед включением модуля установить J3. Или из Fractal-BASIC-Cortex подать директиву REBOOT.

Далее, если модуль будет подключен через USB, то желтый светодиод начнет часто мигать (зеленый – питание всегда горит постоянно). Если USB не подключен и на модуль будет подано питание через кросс и подключен RS485, то желтый светодиод будет гореть постоянно.

Если это так, то модуль находится в режиме Boot-loader Fractal. Теперь можно запустить терминал для выполнения некоторых команд или если этого не требуется, приступить к загрузке обновления.

#### Набор команд загрузчика

При работе загрузчик воспринимает символы, передаваемые в устройство через виртуальный COM-порт как команды. Не распознанные символы просто игнорируются.

#### Версия загрузчика

ENTER. В ответ на нее выводится строка с версией загрузчика. Не путайте с версией Fractal-BASIC-Cortex.

#### Серийный номер устройства.

'N' (на верхнем регистре) + enter. В ответ на нее выдается строка с серийным номером устройства, в кавычках. Пример: "05D4 FF32 3334 5630 4322 3257 8A"

#### Имя файла приложения

'A' (на верхнем регистре) + enter. В ответ на нее выдается строка с именем файла приложения, которое было загружено. Если приложение отсутствует или искажено, то выдается соответствующее сообщение об ошибке. Все сообщения об ошибке сопровождаются включением на 1с красного светодиода.

#### Дата и время файла приложения

'D' (на верхнем регистре) + enter. В ответ на нее выдается строка с датой и временем создания файла приложения, которое было загружено. Если приложение отсутствует или искажено, то выдается соответствующее сообщение об ошибке.

#### Команда стирания BASIC программы

'K' (на верхнем регистре) + enter. Эта команда необходима для удаления BASIC программы которая, например, ввела модуль в неопределенное состояние.

#### Команда запуска Fractal-BASIC-Cortex

'R' (на верхнем регистре) + enter. Эта команда применяется для запуска BASIC после обновления без сброса питания.

#### Загрузка обновления

Для загрузки обновления надо определить номер COM порта в системе под которым определился Boot-loader Fractal.

Этот номер возможно будет отличаться от номера при работе Fractal-BASIC-Cortex.

Номер можно узнать через Диспетчер устройств при подключенном модуле в режиме Boot-loader Fractal.

Если этот номер больше десяти, то надо изменить его, выбрав свободный в диапазоне 1...10.

Далее в командной строке WINDOWS (!!! программы использующие этот COM-порт должны быть закрыты!) набрать команду:

```
copy/b xxx.out COM1:
```

указав правильный номер COM-порта, название конкретного файла обновления с путем к нему.

На экране появится надпись:

```
Скопировано файлов: 1.
```

При этом коротко вспыхнет красный светодиод и непродолжительное время оба светодиода – красный и желтый будут мерцать, после чего красный погаснет, а желтый перейдет в режим соответствующий каналу загрузки (USB).

Для загрузки обновления из под Linux:

```
cp xxx.out /dev/ttyS0
```

На этом загрузка приложения закончена. Надо выключить модуль, снять J3. Запуск произойдет с обновленной версией. Можно не перебрасывать питание, а подать команду запуска приложения 'R'.

### **7. Система команд при доступе в модуль через RS485(MODBUS) и I2C.**

Управление модулем можно осуществлять через RS485 или I2C записью в буфер команд, расположенный по адресам RAM 0x2000 - 0x202F.

Запись пакета команд с параметрами или сообщениями должна осуществляться в одном слитном обмене, начиная с адреса RAM 0x2000. Не допускается запись пакета не с нулевого адреса, иначе команда может быть не выполнена или выполнена неправильно.

Первый байт всегда команда. В зависимости от типа команды количество обязательных и необязательных параметров или длина сообщений могут быть разными.

После подачи команды, модуль ее дешифрует и исполняет. На это уходит некоторое время, поэтому команды должны поступать с некоторой небольшой паузой зависящей от типа исполняемой команды. Дольше выполняются команды обрабатывающие большие участки экрана - это очистки и заполнения.

Максимальное время паузы - не более 20мс.

Далее перечислены команды и их параметры, необязательные параметры (в круглых скобках) могут быть опущены. Команды перечислены в порядке их значимости.

#### Команда вывода символьной строки

<0x80 ... 0xFF> <символы 1 ... 20шт.>

Любое значение первого байта от 0x80 до 0xFF воспринимается как команда вывода строки.

В теле команды может содержаться информация о позиции на экране с которой начнется вывод полученной строки:

0x80...0x94 -> позиции в строке №0(верхняя строка) с 0-го по 19-е знакоместо;

0xA0...0xB4 -> позиции в строке №1 с 0-го по 19-е знакоместо;

0xC0...0xD4 -> позиции в строке №2 с 0-го по 19-е знакоместо;

0xE0...0xF4 -> позиции в строке №3 с 0-го по 19-е знакоместо.

Это удобно при использовании управляющего модуля с Fractal-BASIC. При этом можно использовать оператор **PRINT#** в связке с оператором установки позиции курсора **CSR**.

При этом надо не забывать, что PRINT# при выводе сам формирует первый байт именно в виде описываемой команды вывода символьной строки, добавляя к заранее указанной позиции курсора в операторе CSR +0x80.

Так, значения задаваемые в операторе CSR эквивалентные приведенным выше будут выглядеть так:

0...19 -> позиции в строке №0(верхняя строка) с 0-го по 19-е знакоместо;

32...51 -> позиции в строке №1 с 0-го по 19-е знакоместо;

64...83 -> позиции в строке №2 с 0-го по 19-е знакоместо;

96...115 -> позиции в строке №3 с 0-го по 19-е знакоместо.

Если в качестве команды выбрано значение 0xFF, то модуль будет выводить сообщение с текущей позиции курсора, сохраненной внутри модуля. Другими словами эта команда не изменяет позицию курсора. Это удобно при использовании второго способа форматирования вывода, а именно использования команды установки курсора(см.ниже). Т.е. если в Fractal-BASIC набрать CSR= 127, то пока не будет задан новый CSR, все PRINT# будут выводить символы с текущей позиции без предварительной установки курсора.

**Обратите внимание** при выводе символов в графическом режиме координата вывода задается отдельным графическим курсором, указание позиции в теле команды игнорируется.

Буфер для приема пакета имеет размер 24 байта. Это позволяет принять команду + 20 символов(т.е. одну строку) + 2 служебных символа(например LF и CR).

Все символы с кодом меньше 0x20(пробел), кроме упомянутых LF и CR, игнорируются.

LF - код 0x0A «перевод строки» переводит курсор на следующую строку без изменения номера позиции без всяких сдвигов экрана. Если этот код пришел, когда курсор стоял в последней 3-й строке, то осуществляется перевод на строку №0.

Код CR – 0x0D «возврат каретки» устанавливает курсор в начало текущей строки.

Соответственно комбинация этих символов переводит курсор в начало следующей строки.

Если при выводе сообщения достигнут конец строки, то он автоматически продолжится на следующей или если это была последняя строка – с начала экрана.

#### Команда установка курсора, его типа и режима вывода

<0x02> < X > < Y > (< режим >)

При работе в режиме эмуляции алфавитно-цифрового режима(далее по тексту – символьного) это номер знакоместа соответственно в строке и ряду. Счет ведется начиная с 0-ой позиции 0-ой строки(верхней).

При работе в графическом режиме(далее по тексту – спецсимволов) это координаты левого нижнего угла выводимого сообщения.

Оба режима(символьный и спецсимвольный) работают одновременно и прозрачно, регистры курсора у каждого свои. Просто для правильной работы этой команды есть программный переключатель на нужный комплект регистров в зависимости от контекста параметра <режим> или, в случае его отсутствия, прошлого значения этого параметра.

Необязательный параметр <режим> устанавливает режимы вывода и вид курсора.

Ниже описана отдельная команда управления режимом. Действие этого параметра там полностью идентичное.

Здесь опишем команды управления курсором, остальные ниже.

0x0C -> курсор выключить;

0x0E -> курсор мигающее подчеркивание;

0x0F -> курсор мигающее знакоместо.

При выводе спецсимволов отрисовка курсора не поддерживается, координата графического курсора сдвигается только вправо на новую позицию вплоть до конца экрана.

!Включение курсора значительно замедляет скорость вывода символов.

#### Команда установка режима вывода

<0x01> < режим > (< параметр >)

0x01 -> установить режим вывода с забоем вверх, в случае вывода символов рисуется и символ и его фон поверх бывшего изображения;

0x02 -> режим вывода наложением. т.е. фон остается прежним, рисует только перо;

0x03 -> режим вывода «исключающий или»;

0x04 -> режим вывода инверсный (перо белое);

0x05 -> режим вывода прямой (перо черное);

0x06 -> режим вывода спецсимволов нормального размера, символы как при обычном выводе, но координата задается графическим курсором;

0x07 -> режим вывода спецсимволов широкий -:- ;

0x08 -> режим вывода спецсимволов высокий -:- ;

0x09 -> режим вывода спецсимволов широкий + высокий -:- ;

0x0A -> режим вывода обычных символов по знакоместам;

0x0C -> курсор выключить;

0x0E -> курсор включить мигающее подчеркивание;

0x0F -> курсор включить мигающее знакоместо;

0x10 -> управление контрастностью, значение в параметре, автоматически сохраняется в EEPROM;

0x00 -> ничего не изменять.

После включения питания или сброса, модуль устанавливает следующие параметры:

вывод обычных символов по знакоместам;

режим вывода с забоем вверх;

режим вывода прямой (перо черное);

координаты символьного и графического курсоров обнуляются;

курсor – выключен.

#### Команда очистки экрана и инициализации модуля

<0x03> (< параметр >)

Команда без параметра или с параметром равным 0x00 производит очистку экрана(заполнение нулями) и сбрасывает символьный курсор.

При параметрах:

0x01 -> производит заполнение экрана единицами и сбрасывает символьный курсор;

0x02 -> инвертирует экран и сбрасывает символьный курсор;

0x55 -> общий сброс модуля, начальное состояние параметров указано выше.

Команда управления яркостью подсветки

<0x04> < яркость > (<контрастность>)

Яркость и контрастность могут принимать значения 0...100.

Команда управления звуком

<0x05> < тон > < длительность >

Значение < тон > в пределах от 0x01 до 0xFF. Меньшее значение соответствует более высокому тону.

Длительность - это число умноженное на 10мс.

Команда вывод точки

<0x06> < X > < Y >

Начало координат – левый нижний угол.

Значения < X > - 0x00 ...0x7F.

Значения < Y > - 0x00 ...0x1F.

Точка выводится в соответствии с выбранным режимом графического вывода.

Значения обрезаются по «&» в соответствии с допустимым диапазоном.

Команда вывод линии

<0x07> < X > < Y > < XX > < YY >

Начало координат – левый нижний угол.

Значения < X >, < XX > - 0x00 ...0x7F.

Значения < Y >, < YY > - 0x00 ...0x1F.

Линия выводится в соответствии с выбранным режимом графического вывода.

Значения обрезаются по «&» в соответствии с допустимым диапазоном.

Линия нулевой длины рисуется как точка.

Команда вывод прямоугольника

<0x08> < X > < Y > < XX > < YY >

Начало координат – левый нижний угол.

Значения < X >, < XX > - 0x00 ...0x7F.

Значения < Y >, < YY > - 0x00 ...0x1F.

Прямоугольник выводится в соответствии с выбранным режимом графического вывода.

Значения обрезаются по «&» в соответствии с допустимым диапазоном.

Прямоугольник нулевого размера рисуется как точка.

Работа с кнопками модуля

Ячейка с состоянием кнопок расположена в 39-й ячейке буфера команд. Нажатие соответствует нулю в соответствующем бите.

## 8. Доступ из BASIC.

Как уже отмечалось, модуль может осуществлять вывод на ЖКИ по командам поступающим по интерфейсам и из BASIC. Если планируется использование интерпретатора, то в начале BASIC программы необходимо зарезервировать место под буфер команд поступающих от интерфейсов.

Необходимо до всех объявлений, объявить массив из 10 элементов(40 байт), название массива не имеет значения :

```
DIM lcd(10)
```

Управление контрастностью:

При первом включении и при обновлении версии интерпретатора, пользователю, возможно, придется отредактировать значение контрастности ЖКИ. Это достаточно сделать один раз, значение запоминается в энергонезависимой памяти.

Для управления контрастностью применяется оператор **LCD(110) = ...**

Допустимые значения : 0...100

! Обратите внимание, питание модуля от внешнего источника +8...30В и от USB может давать разную контрастность и яркость подсветки. Это вызвано тем что при питании только от USB внутреннее питающее напряжение ниже оптимального +5В(на работоспособности модуля это не отражается).

Вывод символов:

Вывод символов осуществляется операторами **PRINT#**, **PHB#**, **PHW#** в устройство с адресом 16 (0x10).



Позиция с которой начинается вывод задается оператором **CSR**.

Так, **CSR** = ... :

0...19 -> позиции в строке №0(верхняя строка) с 0-го по 19-е знакоместо;

32...51 -> позиции в строке №1 с 0-го по 19-е знакоместо;

64...83 -> позиции в строке №2 с 0-го по 19-е знакоместо;

96...115 -> позиции в строке №3 с 0-го по 19-е знакоместо.

Если задать **CSR** = **127**, то модуль будет выводить сообщение с текущей позиции курсора, сохраненной внутри модуля.

**Обратите внимание** при выводе символов в графическом режиме координата вывода задается отдельным графическим курсором, значение **CSR** игнорируется.

Если при выводе сообщения достигнут конец строки, то он автоматически продолжится на следующей или если это была последняя строка – с начала экрана.

**Напоминание:** если необходимо продолжение вывода в текущей строке следующими операторами **PRINT#**, **RNB#**, **RHW#**, то в конце оператора необходимо поставить запятую.

Пример : `PRINT#16, "T = ", t`

#### Работа с графикой:

Оператор **POINT** выводит точку с координатами X и Y.

Пример: `POINT 107, 23 ;` вывод точки с координатами x=107, y=23

Оператор **LINE** выводит линию с координатами от прошлого значения **POINT** или **LINE** до X и Y.

Пример: `LINE 67, 14 ;` вывод линии в точку x=67, y=14

Оператор **RECT** выводит прямоугольник с координатами от прошлого значения **POINT** или **LINE** до X и Y.

Пример: `RECT 49, 30 ;` вывод прямоугольника в точку x=49, y=30

Оператор **COLOR** управляет яркостью подсветки ЖКИ. Диапазон 0...100.

Пример: `COLOR 50 ;` включить подсветку 50%

#### Работа со звуком:

Оператор **BEEP** обеспечивает подачу звукового сигнала на модулях где установлен динамик.

Первый параметр – частота в герцах, второй – длительность в секундах 0...9.999с. После обработки этого оператора интерпретатор сразу переходит к обработке следующего, т.е. воспроизведение звука не приостанавливает работу программы. Для случаев, когда требуется ожидание завершения звука, например проигрывание мелодии, длительность необходимо указывать на 10 больше. Если в качестве частоты указан 0, то будет проиграна пауза нужной длительности.

Пример «Былененький он зел»:

10 beep 1335,10.5

20 beep 1000,10.5

30 beep 1335,10.5

40 beep 1000,10.5

50 beep 1335,10.5

60 beep 1260,10.25

70 beep 0,10.25

80 beep 1260,11

#### Управление режимами вывода:

Для управления режимом вывода применяется оператор **LCD(111)** = ...

**0x01** -> установить режим вывода с забоем поверх, в случае вывода символов рисуется и символ и его фон поверх бывшего изображения;

**0x02** -> режим вывода наложением. т.е. фон остается прежним, рисует только перо;

**0x03** -> режим вывода «исключающий или»;

**0x04** -> режим вывода инверсный (перо белое);

**0x05** -> режим вывода прямой (перо черное);

**0x06** -> режим вывода спецсимволов нормального размера, символы как при обычном выводе, но координата задается графическим курсором;

**0x07** -> режим вывода спецсимволов широкий -:- ;

**0x08** -> режим вывода спецсимволов высокий -:- ;

**0x09** -> режим вывода спецсимволов широкий + высокий -:- ;

**0x0A** -> режим вывода обычных символов по знакоместам;

**0x0C** -> курсор выключить;

0x0E -> курсор включить мигающее подчеркивание;

0x0F -> курсор включить мигающее знакоместо;

0x00 -> ничего не изменять.

После включения питания или сброса, модуль устанавливает следующие параметры:

вывод обычных символов по знакоместам;

режим вывода с забоем поперх;

режим вывода прямой (перо черное);

координаты символьного и графического курсоров обнуляются;

курсор – выключен.

#### Команда очистки экрана

Для очистки/заполнения/инверсии применяется оператор **LCD(130) = ...**

0x00 -> производит заполнение экрана нулями и сбрасывает символьный курсор;

0x01 -> производит заполнение экрана единицами и сбрасывает символьный курсор;

0x02 -> инвертирует экран и сбрасывает символьный курсор;

#### Команда установка курсора, его типа и режима вывода

Для управления курсором применяется оператор **LCD(120) = x , y, <rej>**

При работе в режиме эмуляции алфавитно-цифрового режима(далее по тексту – символьного) это номер знакоместа соответственно в строке и ряду. Счет ведется начиная с 0-ой позиции 0-ой строки(верхней).

При работе в графическом режиме(далее по тексту – спецсимволов) это координаты левого нижнего угла выводимого сообщения.

Оба режима(символьный и спецсимвольный) работают одновременно и прозрачно, регистры курсора у каждого свои. Просто для правильной работы этой команды есть программный переключатель на нужный комплект регистров в зависимости от контекста параметра <режим> или, в случае его отсутствия, прошлого значения этого параметра.

Необязательный параметр <режим> устанавливает режимы вывода и вид курсора.

Ниже описана отдельная команда управления режимом. Действие этого параметра там полностью идентичное.

Здесь опишем команды управления курсором, остальные ниже.

0x0C -> курсор выключить;

0x0E -> курсор мигающее подчеркивание;

0x0F -> курсор мигающее знакоместо.

При выводе спецсимволов отрисовка курсора не поддерживается, координата графического курсора сдвигается только вправо на новую позицию вплоть до конца экрана.

!Включение курсора значительно замедляет скорость вывода символов.

#### Работа с кнопками:

Для работы с кнопками используются операторы PIN.

Номера линий 8 - 11.

### **9. Демо режим.**

Если при подаче питающего напряжения нажать и удерживать кнопку К11, то запустится на исполнение демо режим. В нем демонстрируются некоторые возможности модуля.

При удерживании кнопки К12 запускается режим демонстрации времени и даты.

Для сброса демо режимов нужно снять питающее напряжение и снова его подать.