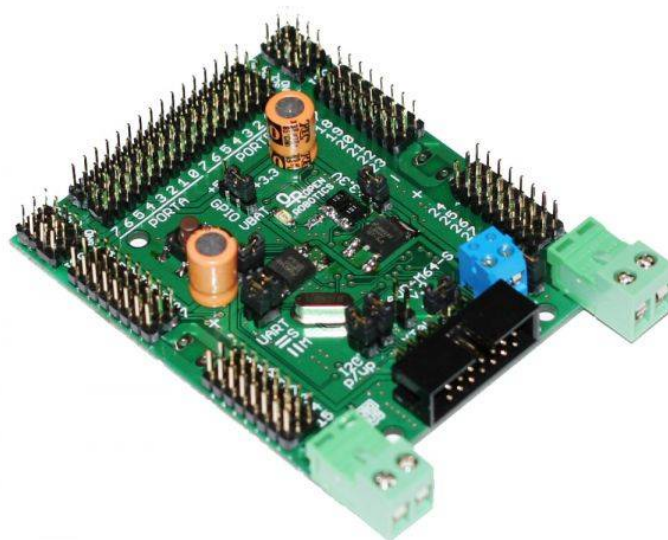


Руководство пользователя
OR-AVR-M128-S

Open Robotics team

12.10.2010

Последняя версия документа доступна по адресу:
<http://roboforum.ru/wiki/OR-AVR-M128-S>



Содержание

| | |
|---|----|
| 1. Описание и основные характеристики..... | 3 |
| 2. Расположение разъемов на плате контроллера..... | 4 |
| 3. Питание контроллера..... | 5 |
| Питание с использованием встроенных стабилизаторов напряжения..... | 5 |
| Питание с использованием внешних стабилизаторов напряжения..... | 5 |
| 4. Разъем RoboBus..... | 6 |
| 5. Порты I2C..... | 7 |
| 6. Порты GPIO..... | 8 |
| Управление питанием портов..... | 9 |
| АЦП..... | 9 |
| 7. UART интерфейс..... | 10 |
| 8. Порты RoboServo (сервоприводы)..... | 11 |
| 10. Габаритные размеры и крепежные отверстия..... | 12 |
| 11. Загрузка прошивки через программатор..... | 12 |
| 12. Загрузка прошивки через bootloader..... | 13 |
| 13. Использование прошивки ORFA..... | 14 |
| Загрузка обновленной прошивки ORFA..... | 14 |
| Работа с ORFA при использовании ПК или ноутбука..... | 14 |
| Формат обмена данными шлюза UART<=>I2C..... | 15 |
| Команды настройки шлюза UART<=>I2C..... | 16 |
| Работа с I2C-устройствами через шлюз (порты RoboI2C и RoboBus)..... | 16 |
| Драйвер низкоскоростного цифрового ввода/вывода (порты GPIO)..... | 17 |
| Драйвер АЦП (порты GPIO)..... | 18 |
| Драйвер управления сервоприводами (порты RoboServo)..... | 19 |
| Драйвер SPI-интерфейса (порт RoboBus)..... | 19 |
| Драйвер интроспекции..... | 20 |

1. Описание и основные характеристики

Универсальный контроллер OR-AVR-M128-S с расширенной поддержкой сервоприводов (далее - контроллер) предназначен для управления устройствами мобильного робота. Контроллер имеет в своем составе все необходимое (за исключением драйвера двигателей) для управления распространенными электронными устройствами, применяемыми в робототехнике, такими как сервоприводы, ИК-дальномеры, контактные и бесконтактные бамперы, датчики линии, различные устройства с интерфейсами SPI и I²C.

В составе контроллера имеются стабилизаторы напряжения на 5 и 3.3 В, которые могут быть использованы для питания внешних устройств.

Основой контроллера является микроконтроллер (МК) ATmega128L, выпускаемый компанией Atmel <http://www.atmel.com/>.

Таблица 1. Основные характеристики контроллера OR-AVR-M128-S

| | |
|---|---|
| Микроконтроллер: | ATmega128L |
| Тактовая частота: | 7,3728 МГц |
| Объем памяти программ (Flash): | 128 КиБ |
| Объем ОЗУ (RAM): | 4 КиБ |
| Объем EEPROM: | 4 КиБ |
| Габаритные размеры: | 66(д)х79(ш)х16(в) мм |
| Логические уровни: | 3,3 В |
| Уровень питания контроллера: | 6 В |
| Максимальный ток, потребляемый контроллером: | 20 мА |
| Максимальный ток внешней нагрузки по линии 5 В: | до 0,8 А (подробнее см. в разделе "Питание контроллера") |
| Максимальный ток внешней нагрузки по линии 3,3 В: | до 0,8 А (подробнее см. в разделе "Питание контроллера") |
| Количество портов сервоприводов: | 32 |
| Количество выводов общего назначения: | 16 |
| в том числе каналов АЦП: | 8 |
| Цифровые интерфейсы: | UART, SPI, I ² C |
| Стандарты разъемов: | RoboBus http://www.roboforum.ru/wiki/RoboBUS RoboGPIO http://www.roboforum.ru/wiki/RoboGPIO RoboI2C http://www.roboforum.ru/wiki/RoboI2C RoboServo http://www.roboforum.ru/wiki/RoboServo |

2. Расположение разъемов на плате контроллера

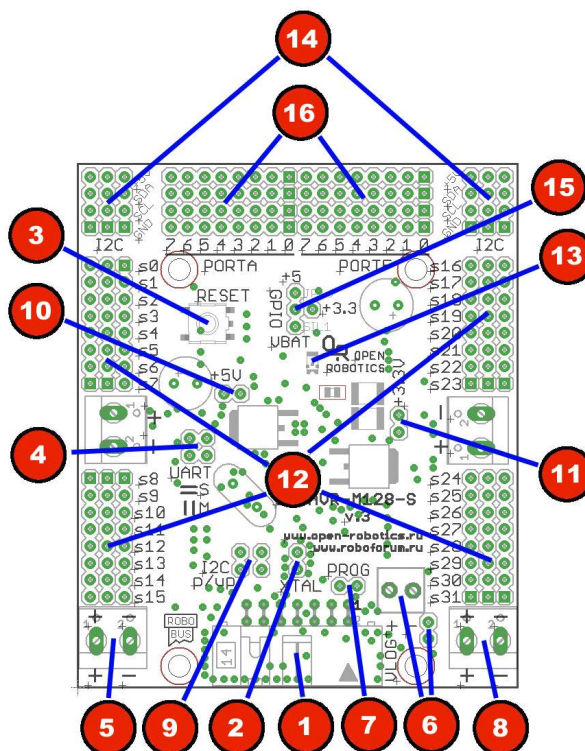


Рис. 1. Схема расположения разъемов и элементов управления на плате контроллера OR-AVR-M128-S

1. Разъем RoboBus
2. Джемпер включения внешнего тактового сигнала
3. Кнопка “Сброс”
4. Джемперы выбора режима работы UART
5. Разъем силового питания сервоприводов 0-15
6. Разъем питания логики и джемпер единого питания логики и силовой части (VLOG);
7. Джемпер отключения режима программирования
8. Разъем силового питания сервоприводов 16-31
9. Джемперы подтяжки шины I²C
10. Джемпер включения встроенного линейного стабилизатора 5,0 В
11. Джемпер включения встроенного линейного стабилизатора 3,3 В
12. Порты RoboServo сервоприводов 0-31
13. Светодиод индикации питания логики
14. Разъемы RoboI2C
15. Джемперы выбора уровня питания 2 линии портов GPIO
16. Порты GPIO

3. Питание контроллера

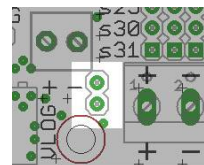
Питание контроллера может осуществляться от источника постоянного тока напряжением 6 вольт с использованием встроенных стабилизаторов напряжения, или от стабилизированного источника постоянного тока напряжением 3,3 вольт.



Всегда проверяйте полярность при подключении источника питания! Неверная полярность может привести к выходу из строя контроллера и подключенных к нему устройств.

Питание с использованием встроенных стабилизаторов напряжения

В этом режиме, для питания используются встроенные стабилизаторы напряжения 5 В и 3,3 В. Эти напряжения выдаются на разъем RoboBus (см. раздел “Разъем RoboBus” далее) и разъемы подключения внешних устройств, а напряжение 3,3 В, кроме того, используется для питания микроконтроллера. Для включения этого режима, необходимо установить переключки +5V и +3V3, и подать внешнее питание 6 вольт через разъем питания логики. Можно подать питание на этот разъем с силового питания 16-31 сервоприводов установив джампер VLOG (см. рисунок справа).



При использовании встроенных стабилизаторов, допустимая нагрузка по линиям +5В и +3,3 В зависит от напряжения внешнего источника питания. Зависимость приведена в таблице:

Таблица 2. Зависимость допустимого тока от напряжения внешнего питания

| Напряжение внешнего источника питания, В | Допустимый ток по линии «3,3 В», А | Допустимый ток по линии «5 В», А |
|--|------------------------------------|----------------------------------|
| 6 | 0,8 | 0,8 |
| 9 | 0,35 | 0,5 |
| 12 | 0,25 | 0,3 |
| 16 | 0,16 | 0,18 |

Питание с использованием внешних стабилизаторов напряжения

Этот режим питания целесообразно использовать, когда мощности встроенных стабилизаторов недостаточно для питания внешних устройств. В этом случае, для питания используются внешние стабилизаторы напряжения 5 В и 3,3 В.



Если питание осуществляется от внешних стабилизаторов, убедитесь, что переключки +5V и +3V3 сняты, иначе возможен выход из строя контроллера и внешнего стабилизатора.

Для включения этого режима, необходимо снять переключки +5V и +3V3, и подать стабилизированные напряжения +5 вольт и +3,3 вольта через соответствующие контакты разъёма RoboBus (см. раздел “Разъем RoboBus” далее). Напряжение +5 вольт необходимо, только если оно требуется для питания внешних устройств, для самого контроллера нужно только +3,3 вольта.

4. Разъем RoboBus

Контроллер оснащён разъемом шины RoboBus. RoboBus - открытый стандарт системной шины, предназначенной для взаимодействия электронных модулей в робототехнике. Подробная информация о стандарте RoboBus - на странице <http://www.roboforum.ru/wiki/RoboBus>

На разъем RoboBus выведены интерфейсы I2C, UART, SPI, 3 вывода общего назначения и питающие напряжения +5 В и +3,3 В. Разъем может использоваться не только для взаимодействия модулей, но и для программирования контроллера (см. одноименный раздел).

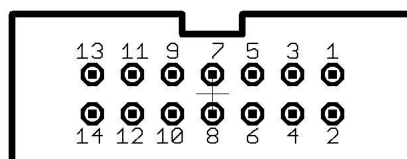


Рис. 2: Схема расположения контактов разъема RoboBus



Нумерация показана для разъема “папа”, установленного на печатную плату, вид с торца платы.



Контакты разъема RoboBus подключены к выводам микроконтроллера через ограничительные резисторы сопротивлением 150 Ом.

Таблица 3. Назначение контактов разъема RoboBus

| № | Название | Интерфейс | Вывод МК | Примечание |
|----|-----------|------------------|----------------|---|
| 1 | GND | | | Земля |
| 2 | 3,3V | | | Питание +3,3 В |
| 3 | RESET | | RESET | Ввод сброса |
| 4 | MOSI | SPI | MOSI (PORTB.5) | Используется для программирования МК |
| 5 | MISO | SPI | MISO (PORTB.6) | Используется для программирования МК |
| 6 | SCK | SPI | SCK (PORTB.7) | Используется для программирования МК |
| 7 | GP2/XTAL1 | | PORTD.6 | Вход\выход общего назначения или вход тактового сигнала |
| 8 | GP1 | | PORTD.7 | Вход\выход общего назначения |
| 9 | GP0 | | PORTB.4 | Вход\выход общего назначения |
| 10 | SDA | I ² C | SDA (PORTC.1) | |
| 11 | SCL | I ² C | SDA (PORTC.0) | |
| 12 | TXD | UART | | UART передача данных (от контроллера) |
| 13 | RXD | UART | | UART прием данных (в контроллер) |
| 14 | 5V | | | |

5. Порты I²C

Помимо разъёма RoboBus, устройства с интерфейсом I²C можно подключать к специальным разъёмам RoboI2C. Это позволяет использовать кабель меньшей ширины, что удобно для подключения подвижных устройств, например вращающийся ультразвуковой сонар.

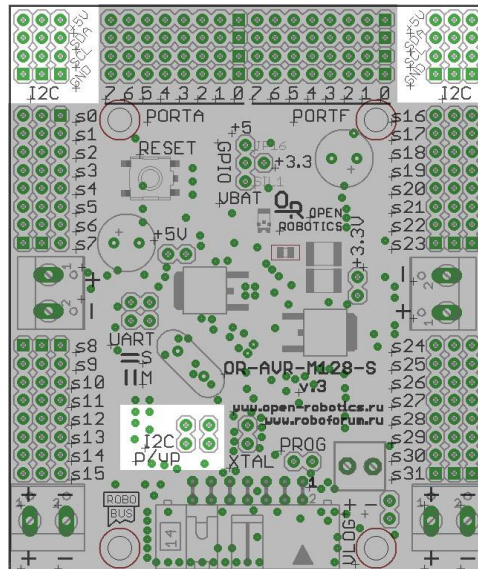


Рис. 3: Разъёмы RoboI2C и джамперы подтяжки линий I²C к напряжению питания

В соответствии со стандартом I²C, сигнальные линии должны быть подтянуты к напряжению питания через резисторы. На плате контроллера имеются 2 джампера, обозначенные I2C p/up. Для использования I²C, эти джамперы следует установить. Если используется несколько контроллеров, соединённых шиной RoboBus, джамперы следует установить только на одном контроллере.

На контроллере имеются 6 разъёмов RoboI2C. Назначение контактов разъёма приведено в таблице. Нумерация контактов - снизу вверх.

Таблица 4. Назначение контактов разъёма RoboI2C

| № | Название |
|---|----------|
| 1 | GND |
| 2 | SCL |
| 3 | SDA |
| 4 | +5V |

6. Порты GPIO

Контроллер имеет 16 выводов общего назначения (GPIO), сгруппированных в 2 порта, в соответствии с названиями портов микроконтроллера, к которым они подключены: А и F. Обозначения портов и номера контактов нанесены на печатную плату. Контакты разъемов портов GPIO расположены в соответствии со стандартом RoboGPIO <http://roboforum.ru/wiki/RoboGPIO>.



Рис. 4: Порты GPIO

Контакты разъемов RoboGPIO нумеруются снизу вверх от 1 до 4. Каждому выводу порта соответствует 4 контакта разъема: сигнал, выбираемое напряжение питания (+5В, +3,3В, или не стабилизированное питание внешнего источника), земля, и фиксированное напряжение +5В.

Контакты на разъемах расположены таким образом, что позволяют напрямую подключать стандартные модельные рулевые машинки (серво, servo, RC servo). При этом контакт разъема серво, по которому подается управляющий сигнал, должен быть подключен к контакту 1 разъема RoboGPIO. Но для этого контроллера подключать модельные сервоприводы в эти порты смысле не имеет, для этого есть специализированные порты RoboServo.



Контакты разъема RoboGPIO подключены к выводам микроконтроллера через ограничительные резисторы сопротивлением 150 Ом.

Управление питанием портов

На контакт 4 разъемов RoboGPIO подается напряжение +5В, либо от встроенного стабилизатора, либо от внешнего.

На контакт 2 разъемов RoboGPIO можно дополнительно подать напряжение питания: 5В, 3,3В или напряжение внешнего источника питания. Выбор напряжения производится не отдельно для каждого вывода порта, а для всех сразу. Для выбора напряжения питания предназначена группа контактов, обозначенная GPIO. С её помощью выбирается напряжение 2 линии сразу для обоих портов А и F.



Рис. 5: Джемпер выбора питания портов

Контакты выбора напряжения расположены в виде буквы Т. Выбор осуществляется установкой джемпера между центральным и одним из крайних контактов. Положение Vbat обычно используют при подключении серво. В этом положении, на контакты 2 соответствующей группы портов подается не стабилизированное напряжение от внешнего источника питания.

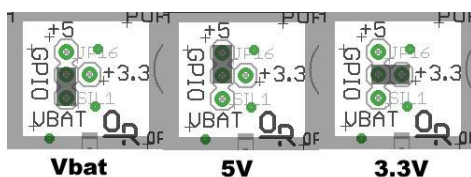


Рис. 6: Варианты питания портов А и F

АЦП

Сигнальные линии порта F, помимо цифрового ввода-вывода, могут использоваться как входы аналогово-цифрового преобразователя (АЦП). Это позволяет подключать датчики, выдающие информацию в аналоговом виде, например переменные резисторы, фоторезисторы, дальномеры Sharp, акселерометры, и т. д.

7. UART интерфейс

UART (Universal Asynchronous Receiver-Transmitter, универсальный асинхронный приёмопередатчик) - один из самых распространённых последовательных интерфейсов связи. Контроллер OR-AVR-M128-D позволяет использовать UART на скоростях обмена до 115200 бит/с.

Для связи через UART используются контакты 12(TXD) и 13(RXD) шины RoboBus. В контроллере предусмотрены 2 варианта подключения UART. В первом варианте, контроллер считается ведущим (Master), передавая данные по линии TXD и принимая - по линии RXD. Второй вариант - подчинённый (Slave), контроллер принимает данные по линии TXD и передаёт - по линии RXD. Для выбора режима предназначена группа джамперов, обозначенная UART.

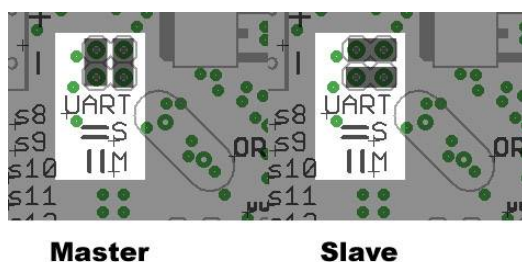


Рис. 7: Конфигурация UART

8. Порты RoboServo (сервоприводы)

Контроллер может управлять до 32 модельными сервоприводами (другие названия - рулевые машинки, RC-servo, servo, серво). Для подключения сервоприводов на контроллере имеется 32 разъёма стандарта RoboServo (<http://roboforum.ru/wiki/RoboServo>). Контакты разъема нумеруются от центра контроллера к левому или правому краю платы от 1 до 3:

Таблица 5: Назначение контактов разъёма RoboServo

| № | Название | Используемый на сервоприводах цвет | Примечание |
|---|----------|------------------------------------|--|
| 1 | PWM | Белый / Желтый | Цифровой сигнал управления сервоприводом |
| 2 | VBAT | Красный | “+” напряжения силового питания |
| 3 | GND | Черный / Коричневый | Земля |

К контроллеру можно подключать любые сервоприводы управляемые сигналом ШИМ с частотой 50Гц и скважностью 500-2500мкс, контроллер обеспечивает точность выдачи скважности порядка 1мкс. Уровень ШИМ сигнала выдаваемого контроллером — 5,0 В.



Обратите внимание — питание сервоприводов 0-15 и сервоприводов 16-31 подаётся через различные разъемы силового питания, подробнее смотрите в разделе «Расположение разъемов на плате контроллера».

На рисунке приведена схема подключения сервоприводов к разъемам в зависимости от расцветки использованной их производителем для обозначения контактов разъема:

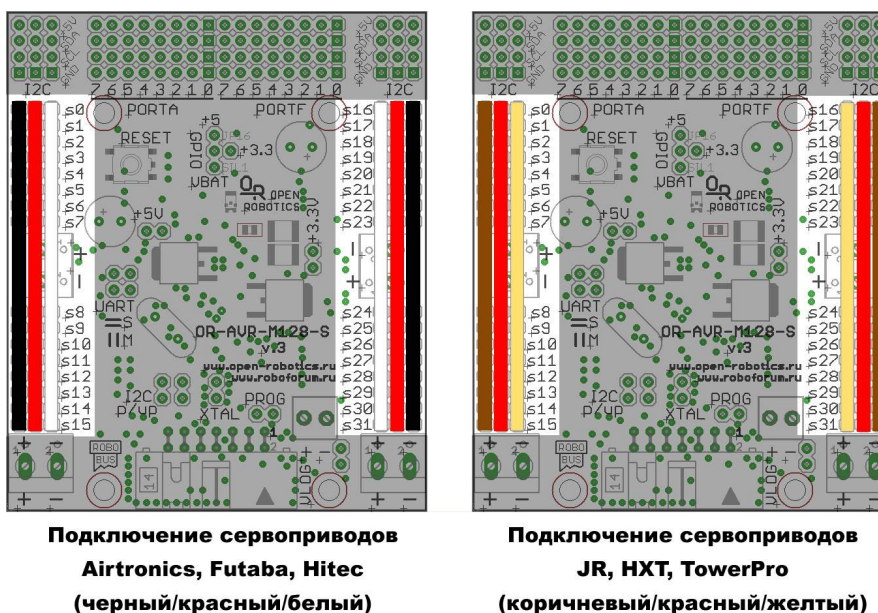


Рис. 8: Схема подключения сервоприводов к контроллеру

10. Габаритные размеры и крепежные отверстия

Для крепления на работе в печатной плате контроллера имеются 4 отверстия под M3 диаметром 3,2 мм. На рисунке размеры указаны в миллиметрах [дюймах].

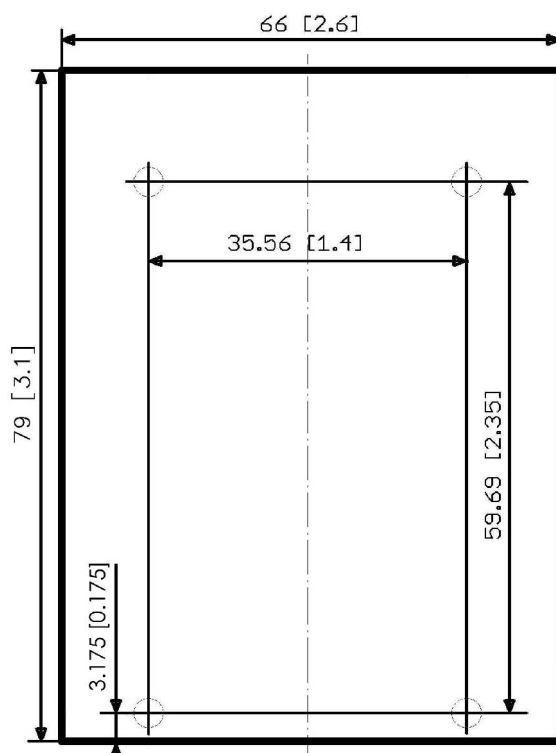


Рис. 9: Габаритные размеры и крепёжные отверстия

11. Загрузка прошивки через программатор

Для загрузки прошивки обычным программатором необходимо снять джампер отключения от программирования PROG, подключить программатор к контроллеру через разъем RoboBus, подать питание на контроллер (если программатор не обеспечивает его питание) и загрузить прошивку. Подробный порядок действий по прошивке контроллера смотрите в инструкции к программатору.

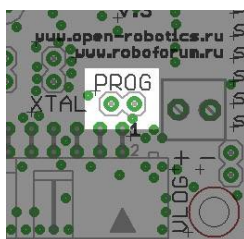


Рис. 10: Джампер отключения от программирования

12. Загрузка прошивки через bootloader

Возможно обновление прошивки без использования программатора, для этого вам потребуется UART соединение ПК с вашим модулем. Например, можно использовать Bluetooth-адаптер OR-BT*, либо OR-USB-UART адаптер (подробней об установлении UART-соединения и где посмотреть соответствующий ему номер COM-порта смотрите в инструкции к соответствующему модулю).

Для обновления прошивки на стороне ПК вам потребуется программа avrdude, её можно бесплатно скачать в интернете отдельно. Кроме того её часто поставляют в рамках различных пакетов для программирования, например, WinAVR.

Порядок обновления прошивки:

1. Установите UART-соединение ПК с модулем;
2. Выполните на ПК команду:

```
"<avrdude_location>\avrdude.exe" -p m128 -c avr911 -P <com_port> -b <bitrate> -U  
flash:w:"<hex_file_with_location>":i -U signature:r:con:r -F -v
```

Например:

```
"C:\WinAVR-20080610\bin\avrdude.exe" -p m128 -c avr911 -P com4 -b 115200 -U  
flash:w:"C:\Robot\or-ssc32\orc32.hex":i -U signature:r:con:r -F -v
```

3. При возникновении ошибки повторите процедуру с пункта 1.

13. Использование прошивки ORFA

Для управления контроллером через UART-порт (например, с ПК, ноутбука, КПК или мобильного телефона) разработана специальная прошивка, одна из последних версий которой доступна для скачивания на странице контроллера: <http://roboforum.ru/wiki/OR-AVR-M128-S>. Прошивка принимает команды через UART-порт, выполняет их на контроллере и выдаёт в этот же порт результат выполнения.

Кроме того в ORFA заложена возможность соединения нескольких ORFA-контроллеров по шине RoboBus (I²C), при этом всеми ими можно будет управлять через единый UART-порт, что полезно при создании сложных конструкций, в которых требуется подключение значительного количества периферии.

Прошивка позволяет работать с:

1. Портами GPIO в режиме АЦП (например, читать информацию с ИК-дальномеров Sharp);
2. Портами GPIO в цифровом режиме (низкоскоростной цифровой вход / выход);
3. Портами RoboServo в режиме управления сервоприводами (servo, серво, RC-servo);
4. SPI и I²C-устройствами;

С точки зрения внутреннего устройства прошивка состоит из шлюза UART \Leftrightarrow I²C и набора драйверов, обрабатывающих команды, обращенные непосредственно к контроллеру. Шлюз принимает команды поступающие через UART, команды настройки шлюза обрабатывает сам, команды направленные к внешним I²C-устройствам перенаправляет на шину I²C, а команды направленные к адресу I²C, закрепленному (специальной командой L настройки шлюза, см. ниже) за самим контроллером — на выполнение соответствующему драйверу.

Загрузка обновленной прошивки ORFA

1. Скачайте последнюю версию прошивки по адресу <http://roboforum.ru/wiki/OR-AVR-M128-S>;
2. Загрузите в контроллер (см. инструкцию к программатору и предыдущий раздел);

Работа с ORFA при использовании ПК или ноутбука

На ПК или ноутбуке UART-порту соответствует понятие «последовательный COM порт». Для работы с таким портом может быть использована как собственная программа, так и любой терминал для работы с COM-портом, например, HyperTerminal встроенный в Microsoft Windows (вызывается через меню Пуск \ Стандартные \ Связь \ HyperTerminal). Об использовании для установки соединения BT-адаптера OR-BT* или модуля USB \Leftrightarrow UART-моста читайте в инструкции к соответствующему модулю.

Общие принципы построения системы команд ORFA

Кроме прямых специализированных простых команд все основные запросы к контроллеру унифицированы с протоколом I²C. Это позволяет однообразно обращаться к контроллеру и любым другим I²C-устройствам подключенным к нему. Все команды текстовые (для управления контроллера с ПК в самом простом случае будет достаточно любого терминала). Перевод строки означает окончание команды и передачу её на исполнение.



Все команды должны оканчиваться переводом строки, который означает окончание команды и передачу её на исполнение!

Работа с GPIO-портами ввода-вывода (простые команды)

Каждая линия GPIO контроллера может работать как цифровой вход или выход, кроме того линии порта F могут работать в режиме АЦП.



***ВНИМАНИЕ!!!** Чтение линий АЦП происходит в автоматическом режиме, поэтому после любых изменений, относящихся к АЦП, делайте паузу в 10мс перед чтением значений.*

Таблица 7. Формат команд для работы с GPIO портами

| Команда | Запрос | Ответ | Примечание |
|--|----------------------------|----------------------------|--|
| Установить режим работы линии порта | PinMode<Port><Line>=<Mode> | PinMode<Port><Line>=<Mode> | Port=(A F) Line=(0..7) Mode=(In Out ADC) <ul style="list-style-type: none"> In – режим «цифровой вход» Out – режим «цифр. выход» ADC – режим «АЦП» Пример команды: PinModeF0=ADC Ответ на неё: PinModeF0=ADC |
| Прочитать значение на входе линии порта | PinGet<Port><Line> | <Port><Line>:<Value> | Port=(A F) Line=(0..7) Value=(0 1) для In/Out, число в формате V.VV для АЦП Пример команды: PinGetF0 Ответ на неё: F0:3.30 |
| Установить значение на выходе линии порта | PinSet<Port><Line>=<Value> | <Port><Line>=<Value> | Port=(A F) Line=(0..7) Value=(0 1) для In/Out Пример команды: PinSetF1=1 Ответ на неё: F1=1 |
| * для АЦП команда запрещена | | | |
| Выбор точности работы АЦП: 8 или 10 бит | AdcBits=<Bits> | AdcBits=<Bits> | Bits=(8 10) Пример команды: AdcBits=8 Ответ на неё: AdcBits=8 |
| Выбрать источник опорн. напряжения для АЦП | AdcRef=<Ref> | AdcRef=<Ref> | Ref=(Int Ext AVCC) <ul style="list-style-type: none"> In – режим «цифровой вход» Out – режим «цифр. выход» ADC – режим «АЦП» Пример команды: AdcRef=AVCC Ответ на неё: AdcRef=AVCC |

Работа с модельными (RC-Servo) сервоприводами (простые команды)



Модельные сервоприводы (RC-servo) управляются прямоугольными сигналами частотой 50Гц, уровнем 3-6 В и скважностью (временем, которое на линии высокое напряжение) от 1 до 2мс. Поэтому не зная характеристик конкретного подключенного сервопривода часто приходится говорить о положении сервопривода, указывая лишь скважность управляющего сигнала. Например, 1200мкс. Это означает, что если сервопривод поворачивается на угол $+75^\circ$ при управляющих сигналах 1мс и 2мс соответственно, тогда 1200мкс соответствует примерно $-75^\circ + 2 \cdot 75^\circ \cdot (1200 - 1000) / (2000 - 1000) = -45^\circ$.



Номера сервоприводов подписаны на печатной плате, рядом с разъемами RoboServo.

Таблица 8. Формат команд для работы с сервоприводами

| Команда | Запрос | Ответ | Примечание |
|--|---|---------|---|
| Задать положения сервоприводов | (#<num> P<pos> S<maxspeed>)+ [T<time> | | num=0..31 – номер сервопривода pos=500..2500 – положение в мкс maxspeed=0..10000 – максимальная скорость сервопривода в мкс/сек time – время за которое нужно привести сервоприводы в заданное положение Нельзя одновременно использовать S и T модификаторы, если S будут использованы, тогда контроллер рассчитает время за которое можно выполнить это движение и за это время все сервоприводы будут двигаться, то есть придут в конечное положение ровно к концу этого периода. Пример команды: #0 P1300 #3 P1700 T4000 |
| Проверить, завершено ли выполнение всех команд | Q | (+ .) | «+» если завершено, «.» если еще нет Пример команды: Q Ответ на неё: + |

Формат обмена данными шлюза UART<=>I²C

Работа с драйверами с помощью I²C запросов организована через концепцию регистров, часто применяемую в I²C-устройствах — ячеек памяти, в которые можно записывать значения и из которых можно их читать. Не дожидаясь конца обработки команды, которую мы передали. Это позволяет разгрузить шину во время обработки команды для работы с другими устройствами.

При работе с I²C-командами все числа передаются побайтно в шестнадцатиричном формате (например, десять будет выглядеть как «0A»), в остальных случаях работаем в десятичном режиме.



Все команды должны оканчиваться переводом строки, который означает окончание команды и передачу её на исполнение!

Таблица 9. Формат обмена данными шлюза UART<=>I²C

| Команда | Запрос | Ответ | Примечание |
|--|---------------------------------|-----------------------|--|
| Получить версию протокола ORFA | V | V1.0 | |
| Сбросить порт I ² C | X | X | |
| Установить I ² C-адрес локального контроллера | L<addr> | L<addr> | addr — uint8 (&0xfe) |
| Установить скорость обмена по I ² C в КГц | C<freq> | C<freq> | freq — uint16 (100 или 400), КГц |
| Прочитать регистр | R<addr><reg>[<len>] | SWASR<rdata>P | addr — uint8 reg — uint8 len — uint8, по умолчанию 1 rdata — массив uint8 |
| Записать в регистр | W<addr><reg><data> | SWA (A) +P | addr — uint8 reg — uint8 data — массив uint8 A = Ack |
| Выполнить I ² C запрос | S<adr+w><data> S<adr+r><len> | SW (A) + SR<rdata> | adr+w — uint8 (=adr & 0xfe) adr+r — uint8 (=adr 0x01) len — uint8 data — массив uint8 rdata — массив uint8 |

Работа с драйверами с помощью I²C запросов организована через концепцию регистров, часто применяемую в I²C-устройствах — ячеек памяти, в которые можно записывать значения и из которых можно их читать. Не дожидаясь конца обработки команды, которую мы передали. Это позволяет разгрузить шину во время обработки команды для работы с другими устройствами.

Для удобной работы с регистрами предусмотрены специальные команды их чтения и записи.

Команды настройки шлюза UART \Leftrightarrow I²C

Для работы непосредственно со шлюзом используется 4 команды: V, X, L<addr> и C<freq>. Примеры использования этих команд приведен ниже (здесь и далее знаками «<» и «>» будет обозначаться направление передачи по UART – в контроллер и от него, соответственно).

Пример отправки запроса версии протокола и получения ответа:

```
< v
> v1.0
```

Пример отправки команды установки I2C-адреса контроллера в 0x20 и получения ответа:

```
< L20
> L20
```

Пример отправки команды установки скорости работы по I2C шине в 100КГц и получения ответа:

```
< C0064
> C0064
```

Пример отправки команды сброса I2C шины и получения ответа:

```
< x
> x
```

Работа с I²C-устройствами через шлюз (порты RoboI2C и RoboBus)

Шлюз UART \Leftrightarrow I²C позволяет работать с любым I²C устройством используя запросы вида:

```
(S<adr+w><data>|S<adr+r><len>)+P
```

На каждый переданный устройству байт будет в ответ получен символ A, а в ответ на каждый запрос нескольких байт будут получены эти несколько байт. Формат ответа на такой запрос:

```
(SW(A)+|SR<rdata>)+P
```

Драйвер низкоскоростного цифрового ввода/вывода (порты GPIO)

Драйвер позволяющий использовать GPIO порты в режимах цифрового логического входа или цифрового логического выхода. Количество управляемых портов N=4 (по количеству выводов МК подключенных к GPIO-портам). Количество регистров, используемых драйвером N*2. Далее будем считать, что {rr} - первый регистр драйвера, а {rr+n} - (n+1)-й регистр драйвера.

Первые N регистров, это чтение\запись значения, оставшиеся N регистров - чтение\запись режима порта. Каждой из 8 линий порта соответствует 1 бит. При работе в режиме входа запись в значение порта включает (1) или выключает (0) подтягивающие резисторы контроллера.

Таблица 10. Порядок работы

| Действие | Регистр* | Отправляемая в регистр или читаемая из него последовательность байт |
|---|--|--|
| Установить режим работы линий ввода\вывода порта | Порт А - 0x06 Порт В - 0x07 Порт С - 0x08 Порт D - 0x09 | <p>XX - устанавливаемая маска, какие линии GPIO порта включить на вход (0), а какие на выход (1)</p> <p>Пример* задания 2-ой и 3-ей линиям порта PORTB режима работы "цифровой вход", а остальным битам этого порта режим "цифровой выход":</p> <pre><S40 07 F3 P >SWAAP</pre> <p><i>Примечание*: Отдавая такую команду, считаем, что ранее командой L40 установили шлюзу локальный адрес 0x40.</i></p> |
| Установить уровень на выходе цифрового порта или включить/выключить подтягивающий резистор на входе цифрового порта | Порт А - 0x02 Порт В - 0x03 Порт С - 0x04 Порт D - 0x05 | <p>XX - устанавливаемая маска, на какие линии GPIO порта подать высокий логический уровень или включить подтягивающий резистор (1), а на какие подать низкий логический уровень или выключить подтягивающий резистор (0)</p> <p>Пример* Установить порт A0 контроллера в логический выход и выставить на нём значение 1:</p> <pre><S40 06 01 S40 02 01 P >SWAASWAAP</pre> <p><i>Примечание*: Отдавая такую команду, считаем, что ранее командой L40 установили шлюзу локальный адрес 0x40.</i></p> |
| Прочитать уровень на входе цифрового порта | Порт А - 0x02 Порт В - 0x03 Порт С - 0x04 Порт D - 0x05 | <p>XX - читаемая маска, на каких линиях GPIO порта какой логический уровень сейчас присутствует или какой логический уровень мы выставили.</p> <p>Пример* Установить весь порт А контроллера в логический вход и прочитать его значение:</p> <pre><S40 06 00 S40 02 S41 01 P >SWAASWASR01P</pre> <p><i>Примечание*: Отдавая такую команду, считаем, что ранее командой L40 установили шлюзу локальный адрес 0x40.</i></p> |

Примечание* Указаны номера регистров действующей на момент написания этого текста версии прошивки ORFA. Чтобы не было опасности изменения своего ПО при смене номеров этих регистров можно использовать драйвер инстроспекции (см. далее).

Драйвер АЦП (порты GPIO)

Драйвер позволяющий управлять блоком Аналого-Цифрового Преобразователя (далее АЦП). Пока поддерживается только 8 входов АЦП (весь порт F). Количество регистров, используемых драйвером - 2. Далее будем считать, что {сс} - первый регистр драйвера, {mm} - второй регистр.

В действующей на момент написания этого текста версии прошивки ORFA, эти регистры имеют номера 0x0C и 0x0D соответственно. Чтобы не было опасности изменения своего ПО при смене номеров этих регистров можно использовать драйвер интроспекции (см. далее).

Первый регистр - конфигурационный, в него пишутся 2 байта всегда:

1. Настройки АЦП (используются младшие 3 бита):
 - биты 0..1:
 - ◆ 00 - External reference
 - ◆ 01 - AVCC
 - ◆ 10,11 - Internal reference
 - бит 2 - режим точности АЦП:
 - ◆ 0 - 8-битный;
 - ◆ 1 - 10-битный;
2. Маска включаемых в АЦП каналов (8 бит - соотв. 8 линиям порта A);

Второй регистр для чтения результатов - из него читаются 8 байт, если режим 8-битный или 16 байт, если режим 10-битный. Можно читать меньше, тогда будем знать состояние первых нескольких линий.



ВНИМАНИЕ!!! Чтение линий АЦП происходит в автоматическом режиме, поэтому после любых изменений, относящихся к АЦП, делайте паузу в 10мс перед чтением значений.

Пример 1*. Включение PORTF.2 в 8-битный режим АЦП с внутренним опорным напряжением.

```
<S 40 0C 03 04 P  
>SWAAAP
```

S40 0C 03 04 - отправим в устройство 0x40 номер регистра 0C и два байта конфигурации 03 – режим внутреннего источника опорного напряжения, 8 бит, 0x04 — включить АЦП только по 2 линии порта F, P - конец запроса. Ответ: SW - начали запись в устройство, AAA - подтверждения записи 3 байт в устройство, P - конец ответа.

Пример 2*. Читаем 0-3 каналы АЦП.

```
<S 40 0D S 41 04 P  
>SWASRFFD7D7D6P
```

S40 0D - отправим в устройство 0x40 номер регистра 0D, после чего S 41 04 – прочитаем теперь из устройства четыре байта значений из АЦП, P - конец запроса. Ответ: SW - начали запись в устройство, A - подтверждения записи 1 байта в устройство, SR<aa><bb><cc><dd> - возвращенные значения на входах АЦП, P - конец ответа.

Примечание*: Отдавая такую команду, считаем, что ранее командой L40 установили шлюзу локальный адрес 0x40.

Драйвер управления сервоприводами (порты RoboServo)



Модельные сервоприводы (RC-servo) управляются прямоугольными сигналами частотой 50Гц, уровнем 3-6 В и скважностью (временем, которое на линии высокое напряжение) от 1 до 2мс. Поэтому не зная характеристик конкретного подключенного сервопривода часто приходится говорить о положении сервопривода, указывая лишь скважность управляющего сигнала. Например, 1200мкс. Это означает, что если сервопривод поворачивается на угол $+75^\circ$ при управляющих сигналах 1мс и 2мс соответственно, тогда 1200мкс соответствует примерно $-75^\circ + 2 \cdot 75^\circ \cdot (1200 - 1000) / (2000 - 1000) = -45^\circ$.

Для контроллера драйвер управления сервоприводами имеет два регистра, через первый осуществляется настройка, на какие порты подключены сервоприводы (пока не используется), а через второй определяется положение каждого из включенных сервоприводов.



Номера сервоприводов подписаны на печатной плате, рядом с разъемами RoboServo.

В действующей на момент написания этого текста версии прошивки ORFA, эти регистры имеют номера 0x0E и 0x0F соответственно. Чтобы не было опасности изменения своего ПО при смене номеров этих регистров можно использовать драйвер инспекции (см. далее).

Таблица 11. Порядок работы

| Действие | Регистр | Отправляемая в регистр последовательность байт |
|---|---------|---|
| Определить, на какие порты подключены сервоприводы* | 0x0E | <i>Команда не используется в данной версии драйвера.</i> |
| Установить положение сервоприводов* | 0x0F | <p>[{aa} {bb}{cc}]* — установить положения сервоприводов (сервопривод {aa} в положение 0x{bb}{cc} мкс).</p> <p>Пример* установки положения 2-го сервопривода в 1200мкс (0x04B0), а 3-го сервопривода в положение 1700мкс (0x06A4): <code><S40 0F 02 04B0 03 06A4 P</code> <code>>SWAAAP</code></p> <p><i>Примечание*:</i> Отдавая такую команду, считаем, что ранее командой L40 установили шлюзу локальный адрес 0x40. А также включили 2 и 3 сервопривод, например, командой из вышеуказанного примера.</p> |

Примечание* В текущей версии драйвера нет возможности отключить какой-либо порт RoboServo.

Драйвер SPI-интерфейса (порт RoboBus)

Приносим извинения, описание драйвера пока не готово.

Драйвер интроспекции

Специальный драйвер, всегда закрепленный за регистром 0x00. Позволяет узнать какие вообще драйверы присутствуют в устройстве, какой версии эти драйвера и какие регистры им назначены.

Типы запросов (регистр, данные...):

- 00 00 — запросить количество драйверов, после чего нужно прочитать 1 байт;
- 00 nn — запросить информацию о драйвере nn, после чего нужно прочитать 6 байт;

Таблица 12. Структура информации в ответе на запрос 00 nn

| Байт | Название | Комментарий |
|------|---------------------------------|--|
| 0..1 | UID | Уникальный идентификатор драйвера. Может быть получен у команды проекта «Open Robotics». Таблица уже выданных идентификаторов: http://roboforum.ru/wiki/ORFA/UID . Для личного использования зарезервирован диапазон адресов с первым байтом 0xFF. |
| 2..3 | Major Version, Minor Version | Старший и младший номер версии. |
| 4 | Start Register | Номер начального регистра драйвера. |
| 5 | Count of registers | Количество регистров драйвера. То есть все регистры драйвера это интервал: (Start Register)..(Start Register + Count of registers - 1) |

Пример 1*. Определяем сколько драйверов в устройстве

```
<S40 00 00 S41 01 P  
>SWAASR06P
```

S40 00 00 - отправим в устройство 0x40 номер регистра 00 и тип запроса 00 - получить количество драйверов, S41 01 - получить ответ 1 байт, P - конец запроса. Ответ: SW - начали запись в устройство, AA - подтверждения записи 2 байт в устройство, SR - начали получать ответ от устройства, 06 - получили 1 байт ответа (количество драйверов), P - конец ответа.

Пример 2*. Определяем информацию о драйвере №2

```
<S40 00 02 S41 06 P  
>SWAASR003101001002P
```

S40 00 02 - отправим в устройство 0x40 номер регистра 00 и тип запроса 02 - получить информацию о драйвере 0x02, S41 01 - получить ответ 6 байт, P - конец запроса. Ответ: SW - начали запись в устройство, AA - подтверждения записи 2 байт в устройство, SR - начали получать ответ от устройства, 003101001002 - получили 6 байт ответа (информация о драйвере - 0x0031 - идентификатор тип драйвера, 0x01.0x00 - версия драйвера, 0x10 - номер первого регистра драйвера, 0x02 - количество регистров драйвера), P - конец ответа.

Примечание*: Отдавая такую команду, считаем, что ранее командой L40 установили шлюзу локальный адрес 0x40.