

# Руководство пользователя УРТК

Владимир Ермаков

14 мая 2010 г.

# Оглавление

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Мехатронные модули</b>	<b>3</b>
2.1	Модуль линейного перемещения . . . . .	4
<b>3</b>	<b>Интерфейс последовательного порта</b>	<b>5</b>
3.1	Стек протоколов . . . . .	5
3.1.1	Командный уровень . . . . .	5
3.1.2	Транспортный уровень . . . . .	5
3.1.3	Физический уровень . . . . .	6
3.2	Команды . . . . .	7
3.2.1	Echo . . . . .	7
3.2.2	Error . . . . .	7
3.2.3	Version . . . . .	8
3.2.4	Get sensors status . . . . .	8
3.2.5	Set duty cycle (PWM) . . . . .	8
3.2.6	Set drive . . . . .	9
3.2.7	Config set encoder . . . . .	9
3.3	URTC Serial Link Interface Protocol . . . . .	10
3.3.1	Общая структура пакета . . . . .	10
3.3.2	FUNCTION . . . . .	11
3.3.3	SIZE . . . . .	11
3.3.4	DATA ... . . . .	11
3.3.5	CRC . . . . .	11

# Глава 1

## Введение

//TODO//

## Глава 2

# Мехатронные модули

Манипуляторы различных кинематических схем, транспортер деталей и поворотный стол изготовлены на базе унифицированных мехатронных модулей следующих типов:

- мехатронный модуль линейного перемещения;
- мехатронный модуль линейного перемещения с захватным устройством;
- мехатронный модуль линейного перемещения с захватным и фрезерным устройствами и модулем ориентации;
- мехатронный модуль вращения.

Габаритные размеры мехатронных модулей, не более:

- модуль линейного перемещения (укороченное исполнение, степени X, Z) /dots/  $450 \times 100 \times 60$  мм;
- модуль линейного перемещения (удлиненное исполнение, степень Y, транспортер деталей) /dots/  $550 \times 100 \times 60$  мм;
- модуль линейного перемещения с захватным устройством (удлиненное исполнение, степень Y) /dots/  $550 \times 210 \times 60$  мм;
- модуль вращения /dots/  $210 \times 210 \times 140$  мм.

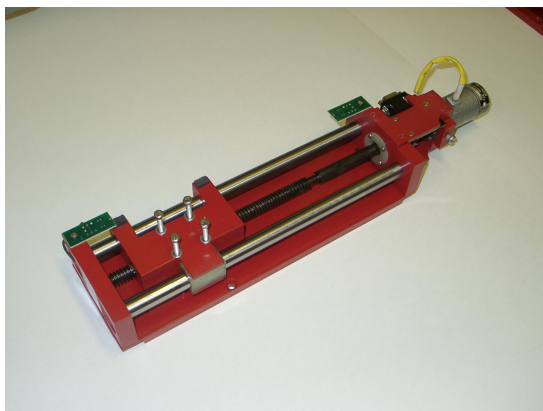


Рис. 2.1: Общий вид модуля линейного перемещения

## 2.1 Модуль линейного перемещения

Внешний вид модуля приведен на рисунке 2.1.

В качестве двигателя в ММ линейного перемещения использован двигатель постоянного тока МН-145А, имеющий следующие основные технические характеристики:

- питание от источника постоянного тока с напряжением  $27 \pm 2.7$  В;
- температурный интервал работы от  $-50^{\circ}\text{C}$  до  $+55^{\circ}\text{C}$ ;
- потребляемый ток, не более:
  - при холостом ходе /dots/ 0.4 А,
  - при номинальной нагрузке 1.69 кг/см /dots/ 0.6 А,
  - при максимальной нагрузке 2.25 кг/см /dots/ 0.7 А,
  - при максимальной нагрузке 2.25 кг/см /dots/ 0.7 А;

**ОСТОРОЖНО!** Во время работы двигателя температура его корпуса может достигать  $60 [U+F0B0] ^{\circ}\text{C}$ . В процессе работы с ММ пользователь должен контролировать температуру двигателей и при их нагреве сверх указанной температуры отключать питание двигателей не менее чем на 15 мин.

## Глава 3

# Интерфейс последовательного порта

### 3.1 Стек протоколов

Задача передачи команд от ПК к УРПК и прием ответа разделена на три уровня. При чем каждый уровень может взаимодействовать только с соседними. Образуется так называемый стек протоколов [3].

Таблица 3.1: Стек протоколов

Командный уровень
Транспортный уровень (USLIP)
Физический уровень (RS-232)

#### 3.1.1 Командный уровень

Взаимодействует с программным обеспечением пользователя сверху и транспортным уровнем снизу. Здесь определены все возможные команды блоку управления.

#### 3.1.2 Транспортный уровень

Обеспечивает разбивку на пакеты данных, контроль верности данных. Взаимодействует с командным уровнем сверху и физическим снизу. Используется протокол USLIP.

### 3.1.3 Физический уровень

Взаимодействует с транспортным уровнем сверху и физической линией передачи данных снизу. Преобразует поток байт от транспортного уровня в физический сигнал и наоборот. Важно заметить, что этот уровень реализуется двумя частями: системным программным обеспечением (операционной системой) и контроллером последовательной шины. Операционная система предоставляет FIFO буфер на запись и чтение шины, но фактически он работает прозрачно и незаметно для транспортного уровня.

## 3.2 Команды

Команды состоят из номера функции и ее данных. Все возможные функции приведены в таблице 3.2.

Таблица 3.2: Функции

№	Название
0	Echo
1	Version
10	Get sensors status
20	Set duty cycle (PWM)
21	Set drive
30	Config set encoder
255	Error

### 3.2.1 Echo

Отправляет обратно принятые данные. Служит для проверки качества соединения.

### 3.2.2 Error

Сообщение об ошибке. Отправляется блоком управления в случае возникновения ошибки, но сам БУ ошибки не принимает. Поля ответа ошибки приведены в таблице 3.3. Список кодов ошибок приведен в таблице 3.4.

Таблица 3.3: Поля ответа ошибки

Function code (u8)	Error code (u8)
--------------------	-----------------

Таблица 3.4: Стандартные коды ошибок

№	Название
1	Ошибочное количество данных
2	Нет такого канала



### 3.2.3 Version

Получить версию микропрограммного обеспечения БУ. Отправляет ASCII строку вида: "URTC hw:2.0 fw:0.4.1". Что означает: версия аппаратного обеспечения – 2.0, версия микропрограммного обеспечения – 0.4.1.

### 3.2.4 Get sensors status

Запрос наборов состояний сенсоров канала(ов). В качестве данных передается набор номеров каналов. В ответе будет переданы состояния каналов в указанном порядке, состоящие из битовой маски состояний входов и положения энкодера (таблица 3.5). Порядок бит маски состояний приведен в таблице 3.6.

Таблица 3.5: Поля ответа сенсоров

Sensor state (u8)	Encoder state (s16)
-------------------	---------------------

Таблица 3.6: Биты состояния сенсоров

.	.	.	.	Start (D)	End (C)	Encoder1 (B)	Encoder0 (A)
---	---	---	---	-----------	---------	--------------	--------------

### 3.2.5 Set duty cycle (PWM)

Устанавливает скважность и направление перемещения мехатронного модуля. Функции передается набор записей канал-скважность (таблица 3.7). Всего каналов – 6.

Таблица 3.7: Биты записи канал-скважность

$P_{sign}$	$C_2$	$C_1$	$C_0$	.	.	$P_9$	$P_8$	$P_7$	$P_6$	$P_5$	$P_4$	$P_3$	$P_2$	$P_1$	$P_0$
------------	-------	-------	-------	---	---	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Пример функции упаковки:

```
unsigned short
pack_pwm(unsigned char channel, signed short pwm)
{
    return (pwm & 0x83ff)|((channel & 7) << 12);
}
```

### 3.2.6 Set drive

Устанавливает состояние исполнительного устройств(а). Функции передается набор канал-состояние (таблица 3.8). Всего каналов – 3

Таблица 3.8: Биты записи канал-состояние

.	.	$C_1$	$C_0$	.	.	.	$S_0$
---	---	-------	-------	---	---	---	-------

Пример функции упаковки:

unsigned char

**pack\_drive**(unsigned char channel, unsigned char state)

```
{
    return (state & 1)|((channel & 4) << 4)
}
```

### 3.2.7 Config set encoder

Устанавливает значение энкодера. Функции передается набор канал-положение (таблица 3.9). Всего каналов – 3

Таблица 3.9: Биты записи канал-положение

Channel (u8)	Encoder state (s16)
--------------	---------------------

### 3.3 URTC Serial Link Interface Protocol

Протокол USLIP является логическим уровнем интерфейса управления оборудованием с помощью асинхронного последовательного канала. Физический уровень интерфейса определен стандартом RS-232. Протокол позволяет производить обмен пакетами данных (data frames) длиной до 254 байт.

Канал конфигурируется следующим образом:

- число бит в посылке – 8
- количество стоп-бит – 1
- бит четности – нет
- скорость обмена – 300 ... 115200 бод
- использование линий управления модемом – произвольное

Основа протокола USLIP – UNIX™ Serial Link Interface Protocol [1] и WAKE [2]. Протокол бинарный. Для передачи служебной информации зарезервировано четыре кода:

- END = 0300<sub>8</sub> = C0<sub>16</sub>
- ESC = 0333<sub>8</sub> = DB<sub>16</sub>
- ESC\_END = 0334<sub>8</sub> = DC<sub>16</sub>
- ESC\_ESC = 0335<sub>8</sub> = DD<sub>16</sub>

Посылка начинается старт-стоповым байтом END, и в теле посылки значение END встречаться не должно. Это достигается за счет замены END на экранирующую последовательность ESC ESC\_END и ESC – на ESC ESC\_ESC. Такая замена называется byte stuffing.

#### 3.3.1 Общая структура пакета

Первым передается старт-стоповый байт END, за ним номер функции, количество байт данных и завершает посылку контрольная сумма CRC.

Таблица 3.10: Поля пакета

END	FUNCTION	SIZE	DATA ...	CRC
-----	----------	------	----------	-----

### 3.3.2 FUNCTION

Номер функции которой отправлены данные. Нежелательно использовать номера равные END и ESC, так как в таком случае придется производить замену.

### 3.3.3 SIZE

Количество полезных байт данных может принимать значения от 0 до 254 байт.

Не учитывает служебные байты пакета END, FUNCTION, SIZE и CRC. В результате замены фактическая длина пакета может возрасти, но это не учитывается.

Если передаваемая команда не имеет параметров, то передается SIZE = 0 и байты данных опускаются.

Если возникает необходимость передать значение SIZE, равное END или ESC, то производится замена, то есть передача экранирующей последовательности. Однако при таких больших значениях SIZE длина пакета столь велика, что его удлинение еще на один байт практически незаметно.

### 3.3.4 DATA ...

Байты данных, количество которых определяется значением SIZE. При SIZE = 0 байты данных отсутствуют.

При передаче значения байта равного END или ESC он заменяется экранирующей последовательностей.

### 3.3.5 CRC

Байт контрольной суммы CRC-8. Контрольная сумма CRC-8 рассчитывается перед операцией замены для всего пакета, начиная с байта END и заканчивая последним байтом данных.

Для расчета контрольной суммы используется полином

$$CRC = X_8 + X_5 + X_4 + 1$$

Значение CRC перед вычислением инициализируется числом DE<sub>16</sub>.

При передаче значения байта контрольной суммы END и ESC заменяются экранирующей последовательностей.

Пример функции подсчета:

```
#define CRC_INIT      0xDE
```

```
static void docrc8(unsigned char b, unsigned char *crc)
{
    for (int i=0; i<8; i++) {
        if (((b ^ *crc) & 1) != 0)
            *crc = ((*crc ^ 0x18) >> 1) | 0x80;
        else
            *crc = (*crc >> 1) & ~0x80;
        b = b >> 1;
    }
}
```

# Литература

- [1] Romkey, J.L. Nonstandard for transmission of IP datagrams over serial lines: SLIP. — RFC 1055 (Standard). — 1988. — июль. <http://www.ietf.org/rfc/rfc1055.txt>.
- [2] Ридико, Л.И. Спецификация протокола WAKE. <http://сахара.ru/lib/wake/>.
- [3] Филимонов, Александр. Построение мультисервисных сетей Ethernet / Александр Филимонов. — bhv, 2007.