

ETHERNET-CAN ШЛЮЗ ECG01/ECG02

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Версия 1.0b



ЗАО «Сервотехника», 2011

Документ описывает предназначение устройств ECG01/ECG02, руководство по подключению и настройке и обновлению программного обеспечения устройства.

ЗАО «Сервотехника» не возлагает на себя обязанность оповещать пользователей СПС о появлении обновлений комплекта документации и программного обеспечения. Пожалуйста, следите за новостями на сайте компании www.servotechnica.ru.

Оглавление

Комплект поставки	3
Подключение	3
Разъем RJ45	4
Разъем DB9-M – Интерфейс CAN	4
Настройка параметров	7
Настройка сетевых параметров	7
Обновление программного обеспечения пульта	8
Протокол взаимодействия	9
Габаритные размеры	12

Комплект поставки

1. Ethernet-CAN шлюз ECG01 или Ethernet-CAN шлюз с встроенным пультом управления ECG02.
2. Блок питания.
3. Компакт диск с документацией и программным обеспечением.

Подключение

Ethernet-CAN шлюз (устройство) предназначен для подключения настольных и промышленных компьютеров, а также ноутбуков к промышленной шине CAN посредством интерфейса Ethernet по протоколу TCP/IP.

На Рис. 1 представлен пример подключения Ethernet-CAN шлюза.

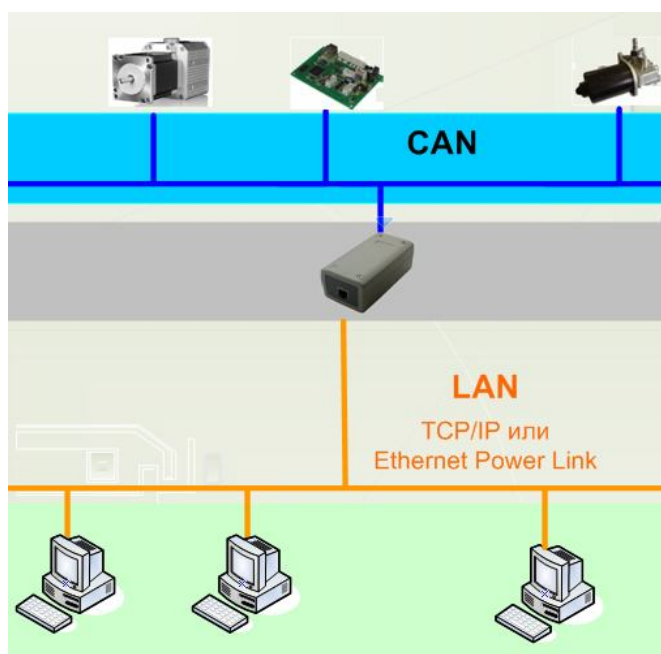


Рис. 1. Общая архитектура сети с использованием Ethernet-CAN шлюза.

Ethernet-CAN шлюз опционально поставляется с встроенным пультом управления (номенклатурный номер ECG02). Руководство пользователя на встроенный пульт управления, приведено в документе «СТ1_112 Руководство пользователя ПУ СПС.pdf».

На Рис. 2 приведено наименование разъемов шлюза.



Рис. 2. Разъемы шлюза.

Разъем RJ45

Разъем RJ45 предназначен для подключения к локальной шине Ethernet.

Прямой порядок обжима кабеля для подключения к концентратору или маршрутизатору, приведен на Рис. 3.

1		бело-оранжевый	бело-оранжевый		1
2		оранжевый	оранжевый		2
3		бело-зелёный	бело-зелёный		3
4		синий	синий		4
5		бело-синий	бело-синий		5
6		зелёный	зелёный		6
7		бело-коричневый	бело-коричневый		7
8		коричневый	коричневый		8

Рис. 3. Порядок обжима витой пары для прямого подключения.

С помощью перекрестного кабеля устройство может быть подключено непосредственно к сетевой карте персонального компьютера. Порядок обжима перекрёстного кабеля дан на Рис. 4.

1		бело-оранжевый	бело-зелёный		1
2		оранжевый	зелёный		2
3		бело-зелёный	бело-оранжевый		3
4		синий	синий		4
5		бело-синий	бело-синий		5
6		зелёный	оранжевый		6
7		бело-коричневый	бело-коричневый		7
8		коричневый	коричневый		8

Рис. 4. Порядок обжима витой пары для перекрестного подключения.

Разъем DB9-M – Интерфейс CAN

CAN – (Controller Area Network) последовательная шина коллективного доступа, специально разработана для обеспечения взаимодействия промышленных управляющих контроллеров. Физический уровень интерфейса совместим со стандартом ISO 11898. Приёмопередатчик учитывает особенности работы систем промышленной автоматики и робототехники, в частности, систем управления электроприводом. Приёмопередатчик обеспечивает:

- совместимость со стандартом ISO 11898;
- скорость обмена до 1000 КБит/сек;
- до 120-и активных узлов физического сегмента сети;
- «горячее» подключение к работающей сети;
- отсутствие помех и переходных процессов при включении или выключении данного устройства для других работающих устройств;
- гарантированную работу в условиях синфазных помех амплитудой до $\pm 25\text{В}$;
- поддерживает работоспособность при обрыве общего опорного провода или одной из линий дифференциальной пары;
- термальную защиту и защиту от статического электричества до 16 КВ.

Для передачи сигналов интерфейса используется одна витая пара проводников + опорный общий провод. Рекомендуется использовать экранированную витую пару, причём опорный общий провод и экран должны иметь в кабеле отдельные жилы.

Состав и описание сигналов интерфейса CAN приведены в Табл. 1.

Табл. 1. Разъём интерфейса CAN.

Номер контакта	Наименование сигнала	Описание
1	-	Не используется
2	CANL	Сигнал шины CAN, низкий уровень в доминантном состоянии. Сигналы CANL и CANH образуют дифференциальную пару сигналов.
3	-	Не используется
4	-	Не используется
5	-	Не используется
6	GND	Общий провод устройства.
7	CANH	Сигнал шины CAN, высокий уровень в доминантном состоянии. Сигналы CANL и CANH образуют дифференциальную пару сигналов.
8	-	Не используется
9	-	Не используется

На Рис. 5 приведён пример использования интерфейса CAN для соединения трех устройств.

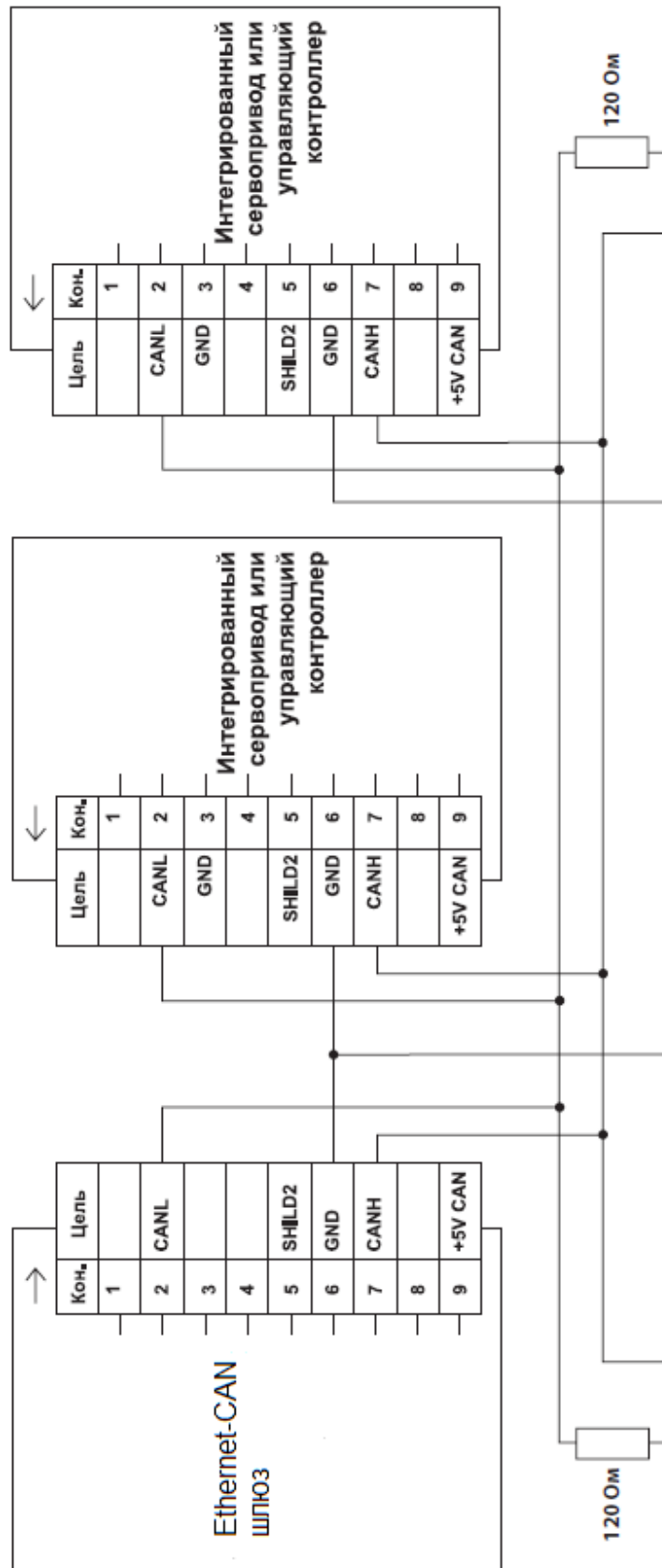


Рис. 5. Пример соединения двух устройств с интерфейсом CAN.

При объединении нескольких устройств используют параллельное их подключение с применением шинной топологии. Соединение устройств посредством дифференциальной пары сигналов требует подключения пассивных терминаторов (резисторов) номиналом 120 Ом / 0.125Вт ±5%. Резисторы терминаторов размещают непосредственно в разъёмах конечных точек шины.

Стандарт интерфейса не определяет конкретные характеристики соединительных проводов и кабелей. Мы рекомендуем применять медную витую пару категории 3 или категории 5. Такие кабели применяются в телефонии и сетях Ethernet. Отличной помехоустойчивостью обладают экранированные варианты таких кабелей. При этом для сигналов CANL/CANH используют одну скрученную пару проводов, а для соединения общего опорного провода, – другую. Ниже даны варианты подключения сигнальных и опорной линий с их распределением по парам проводов.

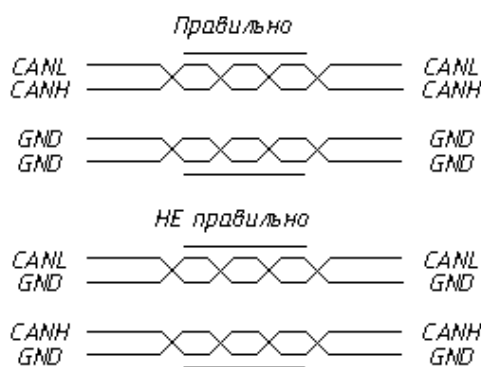


Рис. 6. Использование витой пары.

Хотя оба варианта подключения будут вполне работоспособны, нижний вариант будет обладать меньшей пропускной способностью в условиях помех, особенно в том случае, когда используется экранированный кабель. В первом варианте подключения внешние помехи наводит на сигнальные линии синфазное напряжение, поскольку обе сигнальные линии входят в одну витую пару. Такие помехи успешно подавляются дифференциальным приёмником интерфейса CAN и мало влияют на качество связи. Во втором случае сигнальные линии используют разные пары проводов, находятся в удалении друг от друга и наведённое помехами напряжение может иметь значительную дифференциальную составляющую на приёмниках интерфейса. Дифференциальный приёмник не имеет возможности отфильтровать эту составляющую, и помеха может исказить принятую информацию. В результате искажённые кадры будут браковаться, что снизит пропускную способность интерфейса. Кроме того, во втором варианте создаётся дополнительный канал проникновения дифференциальных помех за счёт протекания зашумлённых токов утечек питания через опорные проводники в каждой паре. Экранирование, в этом случае, не даёт никакого результата.

В текущей версии устройства поддерживается скорость только 1 Мбит/сек.

Настройка параметров

Настройка сетевых параметров устройства

Заводские сетевые параметры:

IP адрес устройства: 192.168.2.25

Маска подсети:	255.255.255.0
IP адрес шлюза узла сети:	192.168.2.1
MAC адрес:	159.219.124.212.19.0

Настройка сетевых параметров производят с помощью терминальной программы `sps_boot_switcher`.

Для задания текущего MAC адреса в командной строке вводят:

```
> sps_boot_switcher.exe -setmac MAC0. MAC1. MAC2. MAC3. MAC4. MAC5
```

Для задания текущих настроек IP сети вводят команду:

```
> sps_boot_switcher.exe -setip IP0.IP1.IP2.IP3 M1.M2.M3.M4 G0.G1.G2.G3
```

Параметры команды позволяет установить IP адрес устройства, маску подсети и IP адрес шлюза сегмента сети, соответственно.

Новые настройки вступают в силу после перезапуска устройства.

Обновление программного обеспечения устройства

Обновление встроенного программного обеспечения осуществляется с помощью терминальной программы `sps_boot_switcher`.

Ethernet-CAN шлюз (устройство) начинает работу с передачи управления начальному загрузчику. Загрузчик проверяет, требуется ли перевести шлюз в режим обновления прикладного программного обеспечения. Переход в режим программирования может быть осуществлен двумя способами, - либо путем удержания клавиши F1 в момент включения (для варианта шлюза с встроенным пультом управления), либо путем обмена UDP сообщения в сегменте сети Ethernet (для обоих вариантов исполнения устройства).

Последовательность обновления 1:

1. Нажимают и удерживают клавишу F1 и включают устройство.
2. Ожидают появление надписи boot на индикаторе.
3. На любом компьютере, подключённом к тому же сегменту сети Ethernet, запускают обозреватель Интернет (например, Internet explorer).
4. В строке адреса обозревателя вводят команду: `https://192.168.2.25`, где последние числа – IP адрес устройства.
5. В окне запроса выбирают имя файла обновления (*.bin) и нажимают ввод.

Последовательность обновления следующая 2:

1. В командной строке компьютера, подключённого к тому же сегменту сети Ethernet, что и устройство вводят: `> sps_boot_switcher.exe -reload`
2. Включают устройство.
3. Программа автоматически переведет устройство в режим программирования, запустит обозреватель Интернет и подключится к WEB серверу устройства.
4. В окне запроса выбирают имя файла обновления (*.bin) и нажимают ввод.

Протокол взаимодействия

Процесс взаимодействия с устройством приведен на Рис. 7.

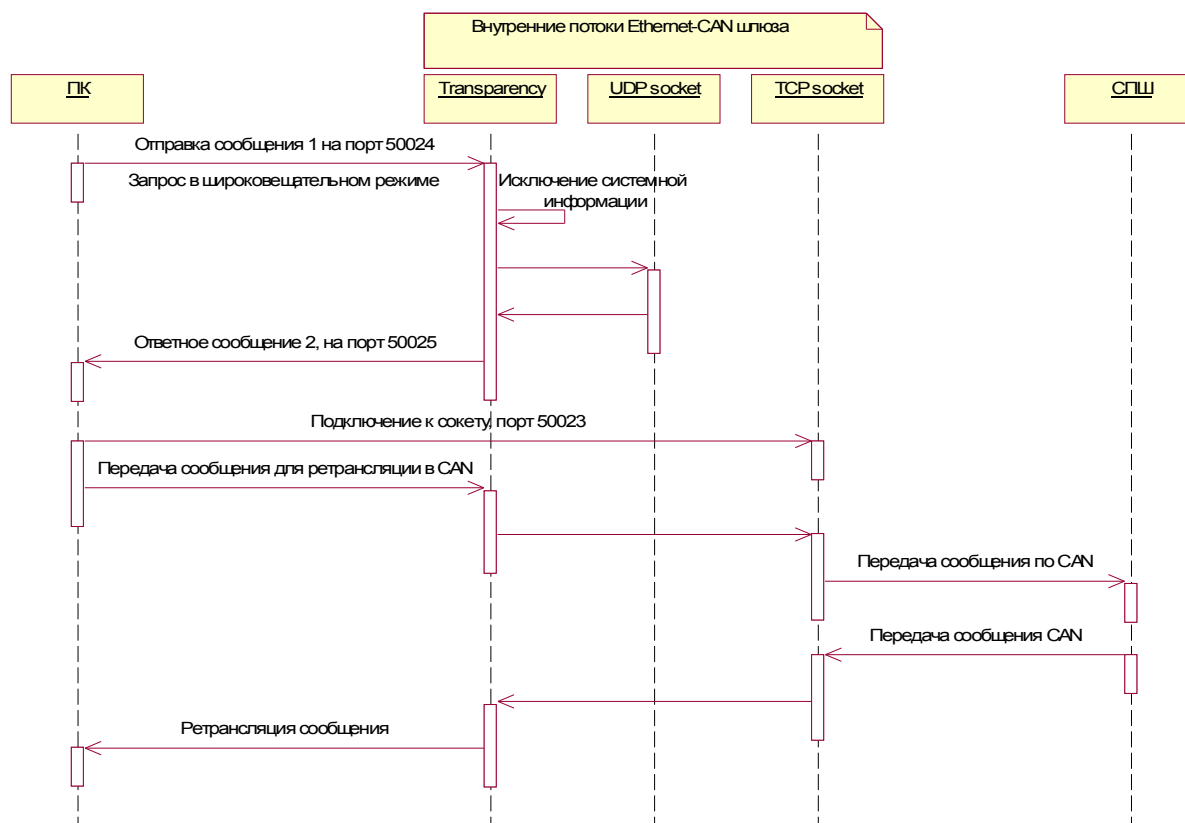


Рис. 7. Последовательность работы с устройством.

При обмене между устройством и ПК используется механизм прозрачности (Transparency), который обрамляет каждое сообщение маркерами и соответственно исключает данные маркеры из тела самого сообщения. Механизм позволяет выделять сообщения в потоке передаваемых данных.

Чтобы определить все доступные устройства в сети, осуществляется их поиск с помощью широковещательного сообщения. Устройство открывает UDP сокет, порт 50024.

Все подключенные Ethernet-CAN шлюзы отвечают на данное сообщение, направляя ответ в порт 50025.

Прикладная программа ПК выполняет подключение к сокету TCP порт 50023, открытое на устройстве.

Далее обмен между прикладной программой ПК и устройством осуществляется в соответствии с протоколом Ethernet – CAN ретрансляции.

Механизм прозрачности

Служебные символы механизма прозрачности.

Служебный символ	Назначение
0x01	Маркер начала сообщения
0x03	Маркер конца сообщения
0x1A	Маркер исключения системных символов

Механизм обрамляет прикладное сообщение маркерами, так называемое прямое преобразование. При этом механизм исключает их появление в теле самого сообщения. Для этого вводится дополнительный маркер исключения 0x1A. При обнаружении служебного символа в теле сообщения протокол заменяет его на два символа 0x1A 0x40+символ. На приемной стороне механизм делает обратные преобразование, исключает маркеры 0x01 и 0x03, а при обнаружении символа 0x1A в теле сообщения исключает его и вычитает из следующего за ним символа 0x40.

Пример преобразования.

Исходное сообщение имеет вид: 0x20 0x31 0x01 0xAB 0x1A 0x05

Исходное сообщение имеет два системных символа, которые должны быть исключены. Результирующее сообщение, которое передается по сети, выглядит следующим образом:

0x01 0x20 0x31 0x1A 0x41 0xAB 0x1A 0x5A 0x05 0x03

Все преобразования, сделанные механизмом, выделены красным цветом.

Листинг процедуры прямого преобразования механизма прозрачности:

```

/*****
TranspareByte
in_byte          Входной байт данных
out_bytes        Указатель на массив выходных данных
RETURN:
    Количество значимых байт
*****/
int TranspareByte(unsigned char in_byte, unsigned char* out_bytes)
{
    unsigned int t;
    if((in_byte == 0x01)|| (in_byte == 0x03)|| (in_byte == 0x1A)) {
        out_bytes[0] = 0x1A;
        out_bytes[1]=in_byte+0x40;
        return 2;
    }
    out_bytes[0] = in_byte;
    return 1;
}

```

Листинг процедуры обратного преобразования механизма прозрачности:

```

/*****
RetranspareByte
in_byte          Входной байт данных
out_bytes        Указатель на выходные данные
RETURN:
    Информацию преобразования
*****/
enum RET_CODE {NONE_BYTE_RECEIVED, ONE_BYTE_RECEIVED, PACKET_RECEIVED};
struct ControlTranspare {
    unsigned char Type;
    unsigned char Ctrl;
};
RET_CODE RetranspareByte(unsigned char in_byte, unsigned char *out_byte)
{
    RET_CODE ret = NONE_BYTE_RECEIVED;
    static struct ControlTranspare ReTranspareStruct;
}

```

```

switch(ReTranspareStruct.Type)
{
case RxBEGIN:
    if(in_byte == 0x01) {
        ReTranspareStruct.Type = RxDATA;
        ReTranspareStruct.Ctrl = 0;
    }
    break;
case RxDATA:
    switch(in_byte){
    case 0x03:
        ReTranspareStruct.Type = RxBEGIN;
        ret = PACKET_RECEIVED;//Пакет принят
        break;
    case 0x01:
        ReTranspareStruct.Type = RxBEGIN;
        break;
    case 0x1A:
        //Из следующего байта данных нужно будет вычесть 0x40
        ReTranspareStruct.Ctrl = 1;
        break;
    default:
        if(ReTranspareStruct.Ctrl == 1) {
            *out_byte = in_byte - 0x40;
            ReTranspareStruct.Ctrl = 0;
        }
        Else {
            *out_byte = in_byte;
            ret = ONE_BYTE_RECEIVED;
        }
    }
    break;
}
return ret;
}

```

Протокол Ethernet – CAN ретрансляции

Структура сообщения приведена в Табл. 2. Формат протокола Ethernet-CAN ретрансляции.

Табл. 2. Формат протокола Ethernet-CAN ретрансляции.

Поле	Команда	Тело сообщения
Размер, байт	1	От 0 до 64

Ниже приведены команды протокола.

Табл. 3. Описание команд протокола Ethernet-CAN ретрансляции.

Команда	Описание
0x05	Запрос значения параметра Ethernet-CAN шлюза
0x06	Передача значения параметра Ethernet-CAN шлюза
0x16	Команда ретрансляции в CAN
0x17	Команда ретрансляции из CAN

Команды разделены на две группы. Первая группа предназначена для взаимодействия непосредственно с устройством, вторая – для осуществления обмена по шине CAN.

Команда 0x05 – Запрос значения параметра Ethernet-CAN шлюза

Номер байта	0	1
Значение	Группа параметров	Номер параметра

В данной версии устройства поддерживается одна группа параметров 0x09 – группа параметров шлюза.

Табл. 4. Номера параметров группы 0x09 – группа параметров шлюза.

Номер параметра	Кол-во байт	Описание
0x00	1	Режим работы устройства. 0 – Неизвестный режим. 1 – Режим СПС. 2 - Режим Ethernet-CAN шлюза.

Устройство Ethernet-CAN шлюза основано на универсальном контроллере, которое может входит в различные устройства. Данный контроллер, например, входит в состав сервопривода СПС на базе синхронного двигателя. В этом режиме протокол взаимодействия имеет другой формат. Протокол Ethernet-CAN ретрансляции относится только к устройствам, у которых параметр 0x00 имеет значение 2.

Команда 0x06 – Запрос значения параметра Ethernet-CAN шлюза

Номер байта	1	2	3-6
Значение	Группа параметров	Номер параметра	Значение

Значение параметра зависит от параметра и может составлять от 1 до 4 байт. Количество байт параметра указано в Табл. 4.

Команда 0x16/0x17 – Команда ретрансляции в/из CAN

Номер байта	1-2	3-9
Значение	идентификатор сообщения CAN	8 байт данных сообщения CAN

В текущей ревизии устройство поддерживает только стандартные сообщения CAN с идентификатором 11 бит. Сообщение может содержать от 0 до 8 байт данных.

Пример формирования сообщения в прикладной программе ПК для передачи устройству:

```
msg[0] = 0x16;  
msg[1] = CAN.ID;  
msg[2] = CAN.ID>>8;  
msg[3] = CAN.D0;  
msg[4] = CAN.D1;  
msg[5] = CAN.D2;  
msg[6] = CAN.D3;  
msg[7] = CAN.D4;  
msg[8] = CAN.D5;  
msg[9] = CAN.D6;  
msg[10] = CAN.D7;
```

Взаимодействие с приводами серии СПШ

Программное обеспечение Мотомастер©, начиная с версии 1.8, а также библиотека связи MotoDLL2, поддерживают Ethernet-CAN шлюз на сетевом уровне. Это означает, что все подключенные приводы к Ethernet-CAN шлюзам будут находиться в списке доступных для подключения устройств, по аналогии с устройствами, подключенные к компьютеру через интерфейс USB.

Программное обеспечение Мотомастер© предназначено для настройки и программирования приводов серии СПШ. Подробности о возможностях программы и порядке работы с ней

приведены в документе «Руководство пользователя сервопривода серии СПШ v31.pdf».

Габаритные размеры

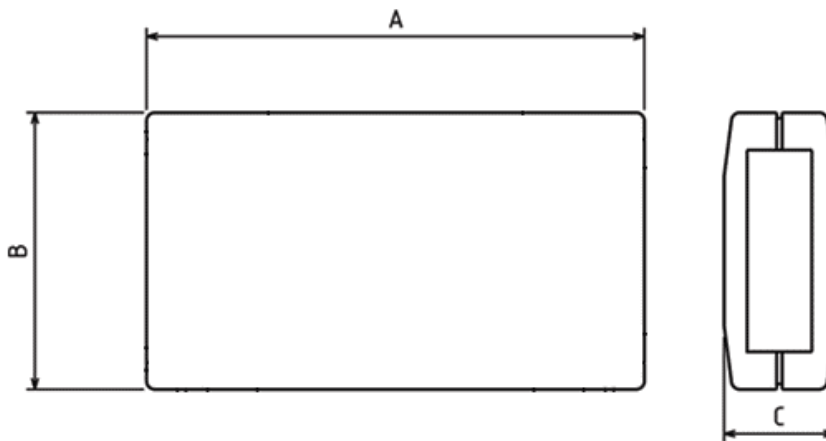


Табл. 5. Таблица габаритных размеров.

Наименование	A, мм	B, мм	C, мм
ECG01	120	60	40
ECG02	150	80	45