

Интерфейсные модули

GT-52xx Руководство пользователя



ИСТОРИЯ ИЗМЕНЕНИЯ ДОКУМЕНТА				
ВЕР	СТРАНИЦА	ПРИМЕЧАНИЕ	ДАТА	РЕДАКТОР
1.00	Создание документа		31.07.20	CH,Hong
1.01	17, 19 -38	Обновление информации	01.12.20	CH,Hong
1.02	39 -42	Обновление информации об изменении размера «окна» приёма / передачи с помощью XML файла-описателя	09.02.21	CH,Hong
1.02R		Перевод на русский язык	21.07.21	IV,Maevskiy
1.02R2		Добавлен раздел по работе с интерфейсными модулями через библиотеку CBL	14.03.22	IV,Maevskiy

Оглавление	
1. Важные примечания	5
1.1. Инструкция по безопасности	6
1.1.1. Символьные обозначения	6
1.1.2. Примечания по безопасности	6
1.1.3. Сертификация	6
2. Список модулей	7
3. Спецификация	8
3.1. GT-5211	8
3.1.1. Схема подключения	8
3.1.2. Индикаторы	9
3.1.3. Индикатор состояния канала	9
3.2. GT-5212	10
3.2.1. Схема подключения	10
3.2.2. Индикаторы	11
3.2.3. Индикатор состояния канала	11
3.3. GT-5221	12
3.3.1. Схема подключения	12
3.3.2. Индикаторы	13
3.3.3. Индикатор состояния канала	13
3.4. GT-5231	14
3.4.1. Схема подключения	14
3.4.2. Индикаторы	15
3.4.3. Индикатор состояния канала	15
3.4. GT-5232	16
3.4.1. Схема подключения	16
3.4.2. Индикаторы	17
3.4.3. Индикатор состояния канала	17
4. Общая эксплуатационная спецификация	18
5. Технические характеристики	19
6. Конфигурирование и примеры работы с модулями	21
6.1. Соотношение данных модулей в таблице отображения	21
6.2. Таблица параметров модуля	24
6.3. Примеры работы с модулями	26
6.3.1. Пример передачи данных	26

6.3.2. Пример приёма данных.....	28
6.3.3. Пример передачи и приёма данных.....	31
6.3.4. Пример передачи данных с использованием функции обработки данных ...	33
6.3.5. Пример проверки переполнения буфера приёма.....	36
7. Изменение размера «окна» приёма / передачи в файлах-описателях (XML).....	38
7.1. Утилита IOGuide Pro	38
7.2. Среда разработки CODESYS	41
8. Пример работы с интерфейсными модулями через библиотеку CBL (Modbus RTU, CODESYS).....	42
9. Габариты	59
9.1. GT-52xx (RTB)	59
10. Монтаж.....	60
10.1. Монтаж и демонтаж модулей.....	60
10.2. RTB (Съёмный клеммный блок).....	61
11. Описание контактов шины G-Bus.....	62

1. Важные примечания

Полупроводниковое оборудование имеет эксплуатационные характеристики, отличные от электромеханического.

Указания по безопасности в случаях применения, установки и технического обслуживания полупроводниковых устройств управления описывают некоторые важные различия между полупроводниковым оборудованием и проводными электромеханическими устройствами.

Из-за этих различий, а также из-за большого разнообразия применений полупроводникового оборудования, все лица, ответственные за применение этого оборудования, должны убедиться, что каждое предполагаемое применение данного оборудования является приемлемым.

Ни при каких обстоятельствах CREVIS не несет ответственности за прямой или косвенный ущерб, возникший в результате использования или применения этого оборудования.

Примеры и диаграммы в этом руководстве приведены исключительно в иллюстративных целях. Из-за множества факторов и требований, связанных с каким-либо конкретным применением, CREVIS не может нести ответственность за фактическое использование, основанное на примерах и схемах.

Предупреждение!

Несоблюдение инструкций может привести к травмам, повреждению оборудования или взрыву.

Не подключайте модули и провода при включенном питании системы. В противном случае это может вызвать электрическую дугу, которая может привести к неожиданным и потенциально опасным воздействиям полевых устройств. При электрической дуге возникает опасность взрыва в опасных зонах. Убедитесь, что область подключения безопасна, или отключите питание системы надлежащим образом перед подключением модулей.

Не прикасайтесь к клеммным колодкам или модулям ввода-вывода во время работы системы. В противном случае это может привести к поражению электрическим током или неисправности устройства.

Держитесь подальше от странных металлических предметов, не связанных с устройством, электромонтажные работы должны контролироваться инженером-электриком. В противном случае это может привести к возгоранию, поражению электрическим током или неисправности устройства.

Осторожно!

Несоблюдение инструкций может привести к травмам, повреждению оборудования или взрыву. Пожалуйста, следуйте инструкциям ниже.

Перед подключением проверьте номинальное напряжение и клеммную колодку. Избегайте мест с температурой более 50 °C. Избегайте попадания прямых солнечных лучей.

Избегайте мест с влажностью более 85%.

Не размещайте модули рядом с легковоспламеняющимися материалами. В противном случае это может вызвать пожар.



Не допускайте прямого приближения к ним какой-либо вибрации.

Внимательно ознакомьтесь со спецификациями модулей, убедитесь, что входные и выходные подключения выполнены в соответствии с этими спецификациями. Для подключения используйте стандартные кабели.


Используйте модули в среде со степенью загрязнения 2.

1.1. Инструкция по безопасности

1.1.1. Символьные обозначения

<p>DANGER</p> 	<p>Определяет информацию о методах или обстоятельствах, которые могут вызвать взрыв в опасной среде, что может привести к травмам, смерти, материальному ущербу или экономическим потерям</p>
<p>IMPORTANT</p>	<p>Определяет информацию, которая имеет решающее значение для успешного применения и понимания продукта</p>
<p>ATTENTION</p> 	<p>Определяет информацию о методах или обстоятельствах, которые могут привести к травмам, материальному ущербу или экономическим потерям.</p> <p>Данный символ поможет вам идентифицировать опасность, избежать её или распознать последствия</p>

1.1.2. Примечания по безопасности

<p>DANGER</p> 	<p>Модули оснащены электронными компонентами, которые могут быть разрушены электростатическим разрядом. При обращении с модулями убедитесь, что окружающая среда (люди, рабочее место и упаковка) хорошо заземлены. Не прикасайтесь к проводящим компонентам, выводам шины G-Bus.</p>
--	---

1.1.3. Сертификация

c-UL-us UL Listed Industrial Control Equipment – сертификация для США и Канады (UL File E235505)

CE Certificate - EN 61000-6-2; Устойчивость к электромагнитным помехам EN 61000-6-4;

Электромагнитная эмиссия

Reach, RoHS (EU, CHINA)

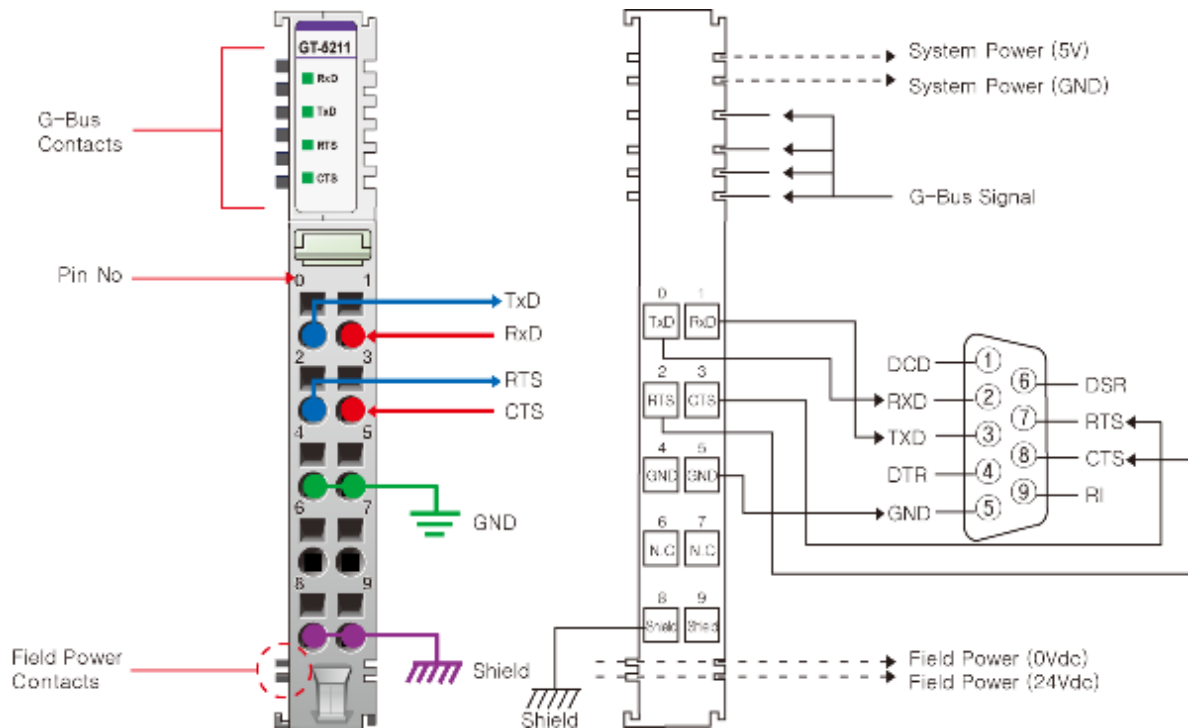
2. Список модулей

Модуль	Описание	ID
GT-5211	Интерфейсный модуль, RS-232, 1 канал, 10 RTB	5211
GT-5212	Интерфейсный модуль, RS-232, 2 канала, 10 RTB	5212
GT-5221	Интерфейсный модуль, RS-422, 1 канал, 10 RTB	5221
GT-5231	Интерфейсный модуль, RS-485, 1 канал, 10 RTB	5231
GT-5232	Интерфейсный модуль, RS-485, 2 канала, 10 RTB	5232

3. Спецификация

3.1. GT-5211

3.1.1. Схема подключения



Контакт	Описание сигнала	Описание сигнала	Контакт
0	RS-232. Контакт TxD	RS-232. Контакт RxD	1
2	RS-232. Контакт RTS	RS-232. Контакт CTS	3
4	RS-232. Контакт GND	RS-232. Контакт GND	5
6	Не используется	Не используется	7
8	Заземление	Заземление	9

3.1.2. Индикаторы



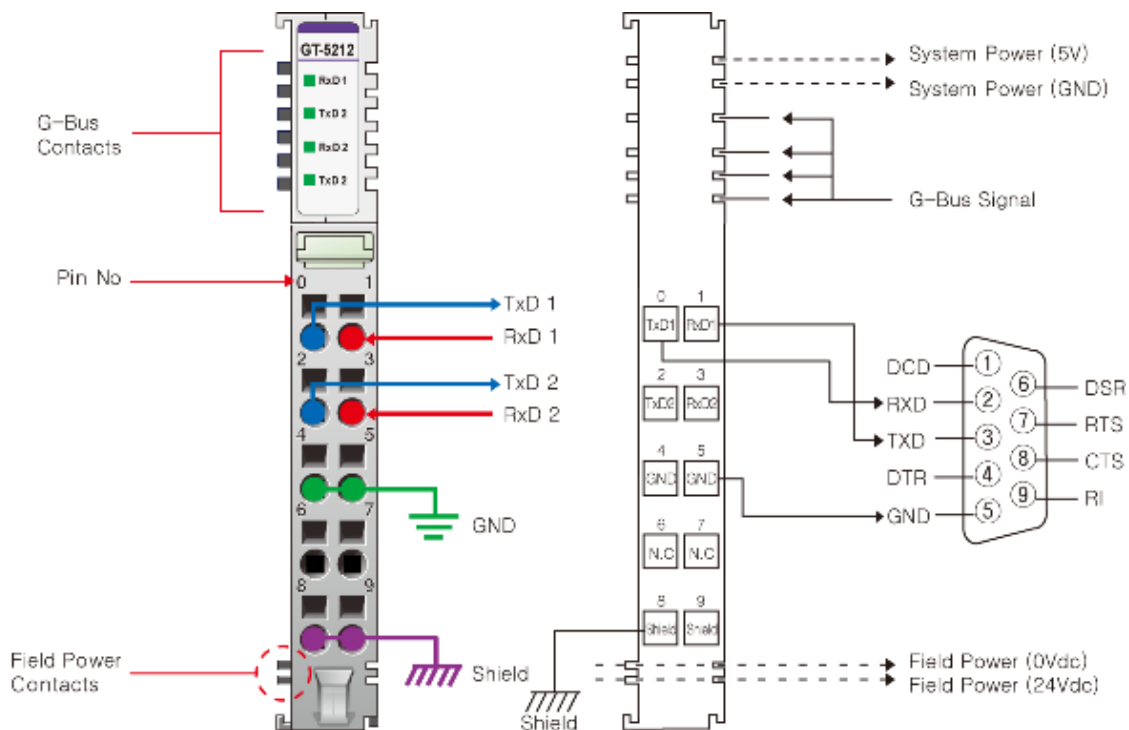
№	Функция / Описание	Цвет
RxD	Индикатор приёма данных	Зелёный
TxD	Индикатор передачи данных	Зелёный
RTS	Готовность передачи данных	Зелёный
CTS	Разрешение передачи данных	Зелёный

3.1.3. Индикатор состояния канала

Статус	Индикатор	Описание
Нормальная работа	Индикатор не горит	Сигнал не передаётся / не приходит
Нормальная работа	Индикатор горит зелёным	Сигнал передаётся / приходит

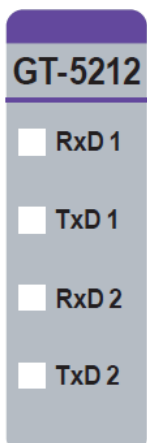
3.2. GT-5212

3.2.1. Схема подключения



Контакт	Описание сигнала	Описание сигнала	Контакт
0	RS-232. Канал 0. Контакт Tx/D	RS-232. Канал 0. Контакт Rx/D	1
2	RS-232. Канал 1. Контакт Tx/D	RS-232. Канал 1. Контакт Rx/D	3
4	RS-232. Общий контакт GND	RS-232. Общий контакт GND	5
6	Не используется	Не используется	7
8	Заземление	Заземление	9

3.2.2. Индикаторы



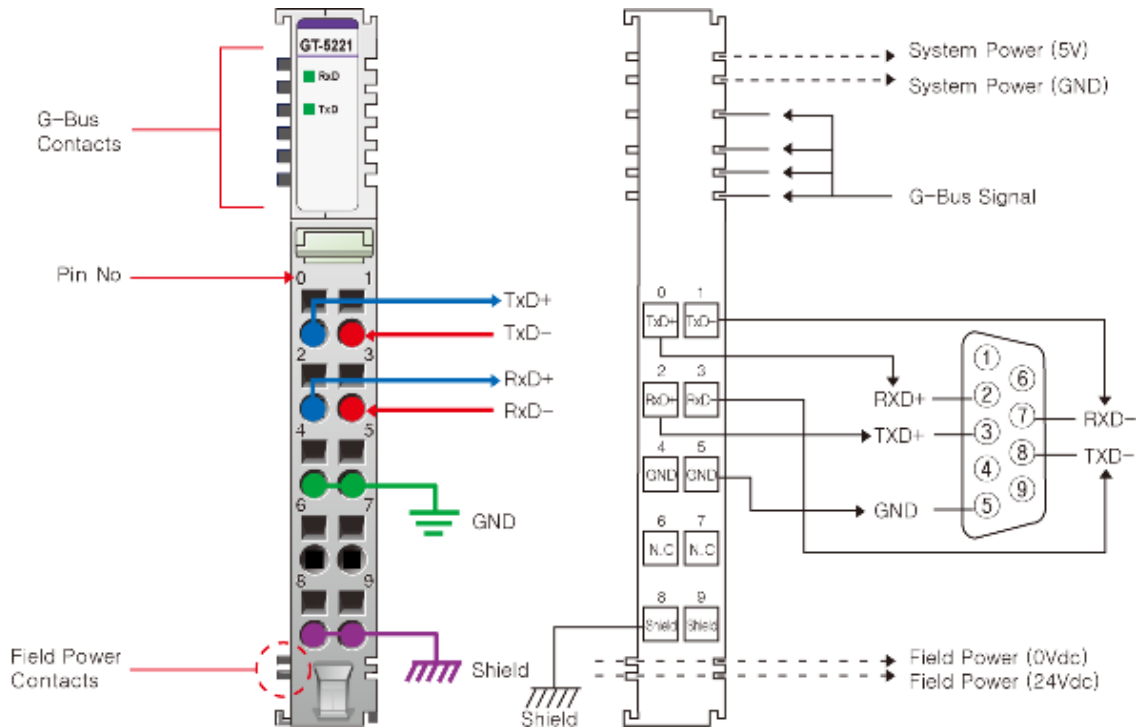
№	Функция / Описание	Цвет
RxD1	Индикатор приёма данных. Канал 0	Зелёный
TxD1	Индикатор передачи данных. Канал 0	Зелёный
RxD2	Индикатор приёма данных. Канал 1	Зелёный
TxD2	Индикатор передачи данных. Канал 1	Зелёный

3.2.3. Индикатор состояния канала

Статус	Индикатор	Описание
Нормальная работа	Индикатор не горит	Сигнал не передаётся / не приходит
Нормальная работа	Индикатор горит зелёным	Сигнал передаётся / приходит

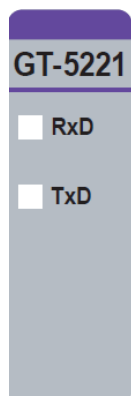
3.3. GT-5221

3.3.1. Схема подключения



Контакт	Описание сигнала	Описание сигнала	Контакт
0	RS-422. Контакт TxD+	RS-422. Контакт TxD-	1
2	RS-422. Контакт RxD+	RS-422. Контакт RxD-S	3
4	RS-422. Контакт GND	RS-422. Контакт GND	5
6	Не используется	Не используется	7
8	Заземление	Заземление	9

3.3.2. Индикаторы



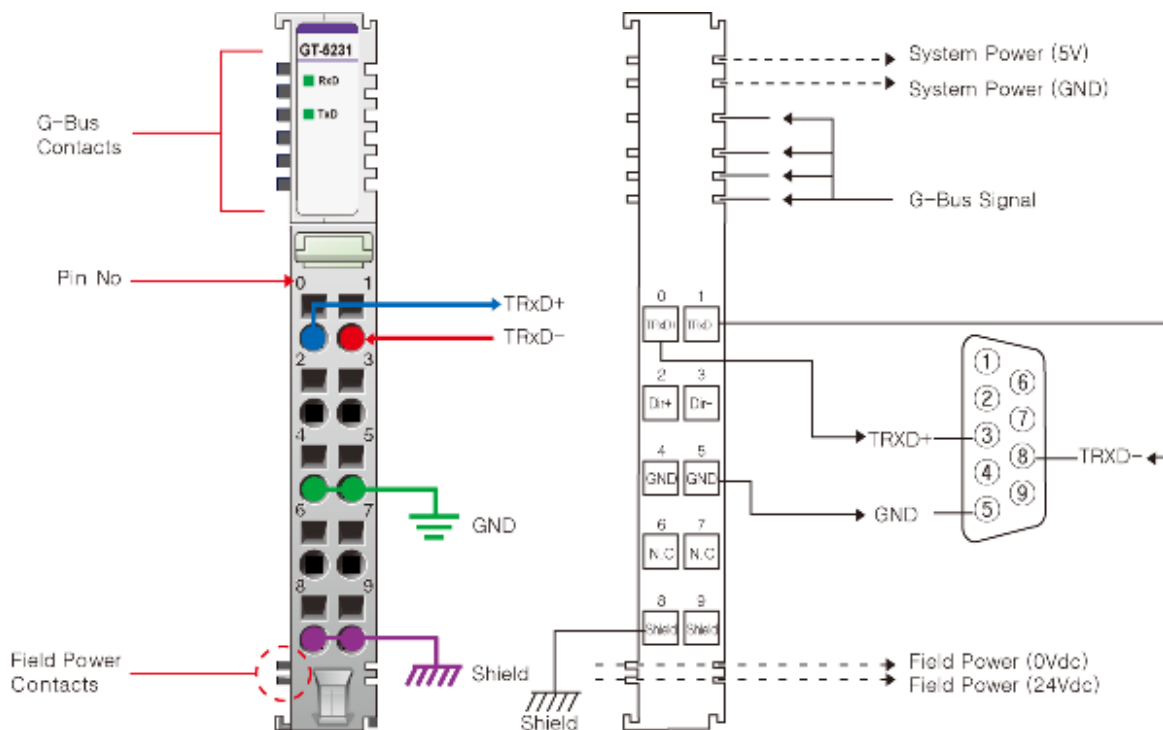
№	Функция / Описание	Цвет
RxD	Индикатор приёма данных	Зелёный
TxD	Индикатор передачи данных	Зелёный

3.3.3. Индикатор состояния канала

Статус	Индикатор	Описание
Нормальная работа	Индикатор не горит	Сигнал не передаётся / не приходит
Нормальная работа	Индикатор горит зелёным	Сигнал передаётся / приходит

3.4. GT-5231

3.4.1. Схема подключения



Контакт	Описание сигнала	Описание сигнала	Контакт
0	RS-485. Контакт TRxD+	RS-485. Контакт TRxD-	1
2	RS-485. Контакт Dir+	RS-485. Контакт Dir-	3
4	RS-485. Контакт GND	RS-485. Контакт GND	5
6	Не используется	Не используется	7
8	Заземление	Заземление	9

3.4.2. Индикаторы



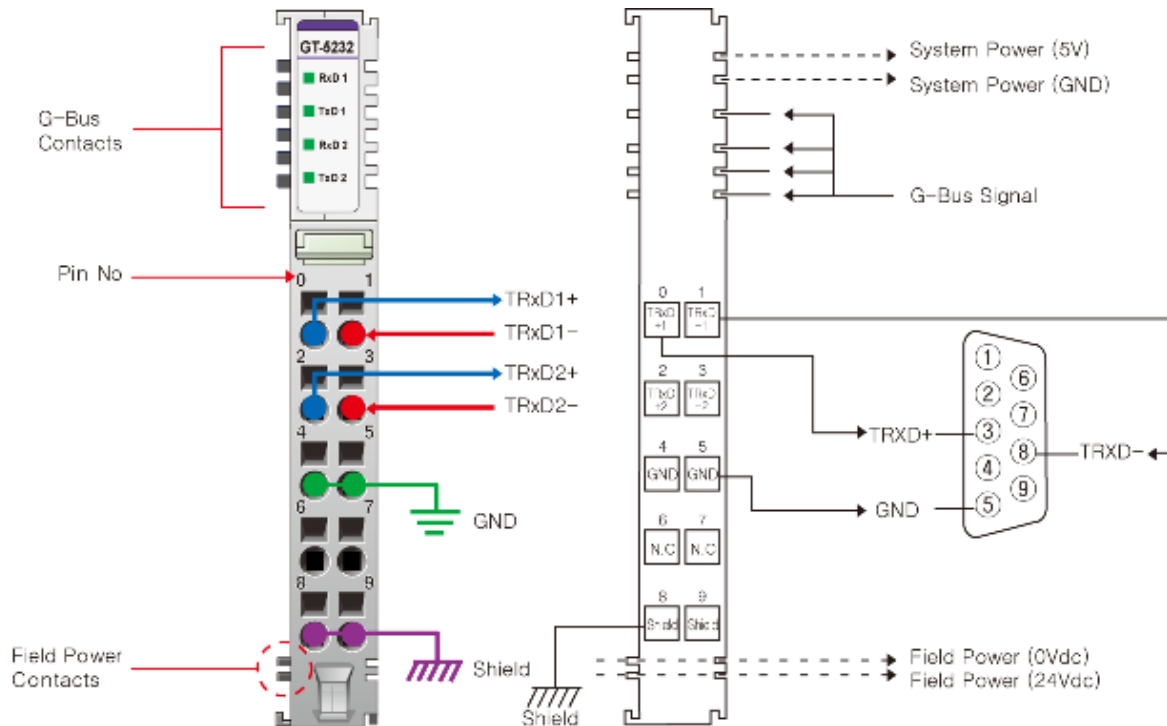
№	Функция / Описание	Цвет
RxD	Индикатор приёма данных	Зелёный
TxD	Индикатор передачи данных	Зелёный

3.4.3. Индикатор состояния канала

Статус	Индикатор	Описание
Нормальная работа	Индикатор не горит	Сигнал не передаётся / не приходит
Нормальная работа	Индикатор горит зелёным	Сигнал передаётся / приходит

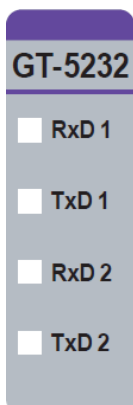
3.5. GT-5232

3.5.1. Схема подключения



Контакт	Описание сигнала	Описание сигнала	Контакт
0	RS-485. Канал 0. Контакт TRxD+	RS-485. Канал 0. Контакт TRxD-	1
2	RS-485. Канал 1. Контакт TRxD+	RS-485. Канал 1. Контакт TRxD-	3
4	RS-485. Общий контакт GND	RS-485. Общий контакт GND	5
6	Не используется	Не используется	7
8	Заземление	Заземление	9

3.5.2. Индикаторы



№	Функция / Описание	Цвет
RxD 1	Канал 0. Индикатор приёма данных	Зелёный
TxD 1	Канал 0. Индикатор передачи данных	Зелёный
RxD 2	Канал 1. Индикатор приёма данных	Зелёный
TxD 2	Канал 1. Индикатор передачи данных	Зелёный

3.5.3. Индикатор состояния канала

Статус	Индикатор	Описание
Нормальная работа	Индикатор не горит	Сигнал не передаётся / не приходит
Нормальная работа	Индикатор горит зелёным	Сигнал передаётся / приходит

4. Общая эксплуатационная спецификация

Эксплуатационная спецификация	
Температура эксплуатации	-40 °C ~ 70 °C
Температура эксплуатации (UL)	-20 °C ~ 60 °C
Температура хранения	-40 °C ~ 85 °C
Относительная влажность	5% ~ 90% без образования конденсата
Монтаж	DIN-рейка
Общая спецификация	
Ударопрочность	IEC 60068-2-27: 2008/15g, 11 мс
Устойчивость к вибрации	На основании IEC 60068-2-6 DNVGL-CG-0039: класс вибрации B, 4g
Электромагнитная эмиссия	EN61000-6-4/All: 2011
Устойчивость к электромагнитным помехам	EN 61000-6-2: 2005
Место установки	Возможна вертикальная и горизонтальная установка
Сертификаты	CE, UL, FCC

5. Технические характеристики

Параметры	GT-5211	GT-5212	GT-5221	GT-5231	GT-5232
Технические характеристики					
Интерфейс	RS-232		RS-422	RS-485	
Каналы передачи	TxD, RxD, Дуплексный режим			TxD, RxD, Полудуплексный режим	
Скорость обмена	1200 ~ 115200 бод				
Количество бит данных	8				
Контроль чётности	Нет (None, по умолчанию), Нечёт (Odd), Чёт (Even)				
Количество стоповых бит	1 бит (по умолчанию), 2 бита				
Контроль потока	RTS, CTS	Не используется			
Искажение бит	< 1.6%		-		
Тип кабеля	Рекомендуется использовать экранированный кабель				
Длина кабеля	Максимум 15 м			Витая пара 1 км	
Напряжение сигнала низкого уровня	-18 ~ 3 В		-		
Напряжение сигнала высокого уровня	3 ~ 18 В		-		
«Окна» приёма /передачи данных	14 байт (по умолчанию) данных. Максимум 61 байт	14 байт (по умолчанию) данных. Максимум 61 байт	14 байт (по умолчанию) данных. Максимум 61 байт		14 байт (по умолчанию) данных. Максимум 61 байт
	Размер «окон» приёма / передачи задаётся. Максимум 63 байта	Размер «окон» приёма / передачи задаётся. Максимум 62 байта	Размер «окон» приёма / передачи задаётся. Максимум 63 байта		Размер «окон» приёма / передачи задаётся. Максимум 62 байта
	1 байт слова статуса, 1 байт слова управления	2 байта слова статуса, 2 байта слова управления	1 байт слова статуса, 1 байт слова управления		2 байта слова статуса, 2 байта слова управления
Буфер приёма (RxD)	1024 байт				
Буфер передачи (TxD)	1024 байт				
Входное сопротивление	-			120 Ом	

Объём таблицы отображения (входные данные)	16 байт (по умолчанию). Максимум 63 байта	16 байт (по умолчанию). Максимум 62 байта	16 байт (по умолчанию). Максимум 63 байта	16 байт (по умолчанию). Максимум 62 байта
Объём таблицы отображения (выходные данные)	16 байт (по умолчанию). Максимум 63 байта	16 байт (по умолчанию). Максимум 62 байта	16 байт (по умолчанию). Максимум 63 байта	16 байт (по умолчанию). Максимум 62 байта
Общая спецификация				
Рассеяние мощности	Максимум 85 мА (5.0 В DC)			
Изоляция	Ввод/вывод к адаптеру: есть изоляция Полевое питание: не используется			
Полевое питание (UL)	Напряжение питания: номинальное 24 В (DC), класс 2			
Полевое питание	Не используется, Полевое питание передается на следующий модуль расширения			
Тип проводников	Кабель ввода/вывода Макс. 2.0 мм ² (AWG 14)			
Крутящий момент	0.8 Нм			
Масса	57 г			
Размер модуля	12 мм x 99 мм x 70 мм			
Условия эксплуатации	Обратитесь к «Общая эксплуатационная спецификация»			

6. Конфигурирование и примеры работы с модулями

6.1. Соотношение данных модулей в таблице отображения

Таблица отображения для одноканальных интерфейсных модулей (GT-5211, GT-5221, GT-5231) имеет вид:

Входные данные модуля		Выходные данные модуля	
Байт 0	Слово статуса	Байт 0	Слово управления
Байт 1	Длина принимаемых данных	Байт 1	Длина передаваемых данных
Байт 2	Принимаемые данные (Байт 0)	Байт 2	Передаваемые данные (Байт 0)
Байт 3	Принимаемые данные (Байт 1)	Байт 3	Передаваемые данные (Байт 1)
Байт 4	Принимаемые данные (Байт 2)	Байт 4	Передаваемые данные (Байт 2)
Байт 5	Принимаемые данные (Байт 3)	Байт 5	Передаваемые данные (Байт 3)
Байт 6	Принимаемые данные (Байт 4)	Байт 6	Передаваемые данные (Байт 4)
Байт 7	Принимаемые данные (Байт 5)	Байт 7	Передаваемые данные (Байт 5)
Байт 8	Принимаемые данные (Байт 6)	Байт 8	Передаваемые данные (Байт 6)
Байт 9	Принимаемые данные (Байт 7)	Байт 9	Передаваемые данные (Байт 7)
Байт 10	Принимаемые данные (Байт 8)	Байт 10	Передаваемые данные (Байт 8)
Байт 11	Принимаемые данные (Байт 9)	Байт 11	Передаваемые данные (Байт 9)
Байт 12	Принимаемые данные (Байт 10)	Байт 12	Передаваемые данные (Байт 10)
Байт 13	Принимаемые данные (Байт 11)	Байт 13	Передаваемые данные (Байт 11)
Байт 14	Принимаемые данные (Байт 12)	Байт 14	Передаваемые данные (Байт 12)
Байт 15	Принимаемые данные (Байт 13)	Байт 15	Передаваемые данные (Байт 13)
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----
Байт 62	Принимаемые данные (Байт 60)	Байт 62	Передаваемые данные (Байт 60)

Объём таблицы отображения (принимаемых / передаваемых данных) может быть изменён с помощью соответствующего параметра в таблице параметров модуля. Максимальный объём – 63 байта.

Таблица отображения для двухканальных интерфейсных модулей (GT-5212, GT-5232) имеет вид:

Входные данные модуля		Выходные данные модуля	
Канал 0			
Байт 0	Слово статуса	Байт 0	Слово управления
Байт 1	Длина принимаемых данных	Байт 1	Длина передаваемых данных
Байт 2	Принимаемые данные (Байт 0)	Байт 2	Передаваемые данные (Байт 0)
Байт 3	Принимаемые данные (Байт 1)	Байт 3	Передаваемые данные (Байт 1)
Байт 4	Принимаемые данные (Байт 2)	Байт 4	Передаваемые данные (Байт 2)
Байт 5	Принимаемые данные (Байт 3)	Байт 5	Передаваемые данные (Байт 3)
Байт 6	Принимаемые данные (Байт 4)	Байт 6	Передаваемые данные (Байт 4)
Байт 7	Принимаемые данные (Байт 5)	Байт 7	Передаваемые данные (Байт 5)
Канал 1			
Байт 8	Слово статуса	Байт 8	Слово управления
Байт 9	Длина принимаемых данных	Байт 9	Длина передаваемых данных
Байт 10	Принимаемые данные (Байт 0)	Байт 10	Передаваемые данные (Байт 0)
Байт 11	Принимаемые данные (Байт 1)	Байт 11	Передаваемые данные (Байт 1)
Байт 12	Принимаемые данные (Байт 2)	Байт 12	Передаваемые данные (Байт 2)
Байт 13	Принимаемые данные (Байт 3)	Байт 13	Передаваемые данные (Байт 3)
Байт 14	Принимаемые данные (Байт 4)	Байт 14	Передаваемые данные (Байт 4)
Байт 15	Принимаемые данные (Байт 5)	Байт 15	Передаваемые данные (Байт 5)

Объём таблицы отображения (принимаемых / передаваемых данных) может быть изменён с помощью соответствующего параметра в таблице параметров модуля. Максимальный объём – 63 байта. При объёме таблицы отображения, равном 62 байта данные канала 0 будут занимать байт 0 – 31, данные канала 1 – байт 32 – 62.

Слово статуса для всех интерфейсных модулей имеет следующий набор битов:

№ Бита	Обозначение	Описание
0	IA	Подтверждение инициализации
1	TA	Подтверждение передачи
2	RR	Запрос принят
3	RBO	Буфер приёма (RxD) переполнен
4	RE	Буфер приёма (RxD) не пустой
5	FRA	Подтверждение очистки буфера на приём
6	FTA	Подтверждение очистки буфера на передачу
7	TPA	Подтверждение отправки обработанных данных

Слово управления для всех интерфейсных модулей имеет следующий набор битов:

№ Бита	Обозначение	Описание
0	IR	Запрос инициализации
1	TR	Запрос передачи
2	RA	Подтверждение приёма
3	-	-
4	-	-
5	FR	Запрос очистки буфера на приём
6	FT	Запрос очистки буфера на передачу
7	TPR	Запрос отправки обработанных данных

6.2. Таблица параметров модуля

Таблица параметров для модуля GT-5211 имеет вид:

Объем таблицы параметров модуля: 4 байта

Параметры модуля

№ Бита	7	6	5	4	3	2	1	0
Байт 0	Обработка данных 1	Стоп бит 1	Чётность 1 (Код)		Скорость обмена 1 (Код)			
Байт 1	Объём таблицы отображения модуля («Окон» приёма / передачи)						Управление потоком	
Байт 2	Не используется							
Байт 3	Не используется							

Таблица параметров для одноканальных интерфейсных модулей (GT-5221, GT-5231) имеет вид:

Объем таблицы параметров модуля: 4 байта

Параметры модуля

№ Бита	7	6	5	4	3	2	1	0
Байт 0	Обработка данных 1	Стоп бит 1	Чётность 1 (Код)		Скорость обмена 1 (Код)			
Байт 1	Объём таблицы отображения модуля («Окон» приёма / передачи)							
Байт 2	Не используется							
Байт 3	Не используется							

Таблица параметров для двухканальных интерфейсных модулей (GT-5212, GT-5232) имеет вид:

Объем таблицы параметров модуля: 4 байта

Параметры модуля

№ Бита	7	6	5	4	3	2	1	0
Байт 0	Обработка данных 1	Стоп бит 1	Чётность 1 (Код)		Скорость обмена 1 (Код)			
Байт 1	Объём таблицы отображения модуля («Окон» приёма / передачи)							
Байт 2	Обработка данных 2	Стоп бит 2	Чётность 2 (Код)		Скорость обмена 2 (Код)			
Байт 3	Не используется							

В таблицах параметров модулей:

- Бит **«Обработка данных 1 / 2»** - режим передачи данных из «окна» передачи в буфер передачи перед пересылкой непосредственно в канал для каналов 0 / 1, соответственно (**0** – отключен, **1** - включен). Данный бит устанавливается для возможности пересылки в канал посылки объемом, превышающим размер «окна» передачи. Например, необходимо переслать посылку объемом 100 байт при размере окна 62 байта. Для этого сначала первые 62 байта посылки помещаются в «окно» передачи, далее это «окно» пересылается в буфер передачи. После повторяется аналогичная процедура для оставшихся 38 байт посылки. В конце вся посылка (100 байт) из буфера передачи отправляется в канал.

- Бит **«Стоп бит 1 / 2»** - определяет количество стоповых бит для канала 0 / 1, соответственно (**0** – 1 бит, **1** – 2 бита);

- Параметр **«Чётность 1 / 2 (Код)»** - определяет режим проверки чётности для канала 0 / 1, соответственно (**0** – Нет проверки / None, **1** – Нечёт / Odd, **2** – Чёт / Even);

- Параметр **«Скорость обмена 1 / 2 (Код)»** - определяет скорость обмена для канала 0 / 1, соответственно (**0** – 115200 бод, **1** – 1200 бод, **2** – 2400 бод, **3** – 4800 бод, **4** – 9600 бод, **5** – 19200 бод, **6** – 38400 бод, **7** – 57600 бод);

- Параметр **«Управление потоком»** для модуля GT-5211 определяет режим управления потоком (**0** – RTS/CTS отключены, **1** - RTS включен, **2** – CTS включен, **3** – RTS/CTS включены);

- Параметр **«Объем таблицы отображения модуля («Окон» приёма / передачи)»** определяет соответствующий объем. Для минимизации времени пересылок данных из «окон» приёма / передачи в буферы приёма / передачи рекомендуется задавать как можно больший объем таблицы.

* При изменении параметров для их установки **модуль должен быть перезагружен.**

6.3. Примеры работы с модулями

6.3.1. Пример передачи данных

Необходимо последовательно передать следующий набор данных:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z (26 байт)

Размер «окон» приёма / передачи – 16 байт.

Шаг 1

Записать первые 14 байт в «окно» передачи, задать длину передаваемых данных 14 байт

Инвертировать флаг запроса передачи TR (TR≠TA)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	0	0	0
	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	1	0
Длина перед. данных	байт)							
Перед. данные 0	'A' (0x41: ASCII код)							
Перед. данные 1	'B' (0x42)							
Перед. данные 2	'C' (0x43)							
Перед. данные 3	'D' (0x44)							
Перед. данные 4	'E' (0x45)							
Перед. данные 5	'F' (0x46)							
Перед. данные 6	'G' (0x47)							
Перед. данные 7	'H' (0x48)							
Перед. данные 8	'I' (0x49)							
Перед. данные 9	'J' (0x4A)							
Перед. данные 10	'K' (0x4B)							
Перед. данные 11	'L' (0x4C)							
Перед. данные 12	'M' (0x4D)							
Перед. данные 13	'N' (0x4E)							

Шаг 2

Проверить бит подтверждения передачи TA (TR=TA, значит передача завершена)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	0	1	0

Шаг 3

Записать оставшиеся 12 байт в «окно» передачи, задать длину передаваемых данных 12 байт

Инвертировать флаг запроса передачи TR (TR≠TA)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	0	0
Длина перед. данных	12 (12 байт)							
Перед. данные 0	'O' (0x4F:ASCII код)							
Перед. данные 1	'P' (0x50)							
Перед. данные 2	'Q' (0x51)							
Перед. данные 3	'R' (0x52)							
Перед. данные 4	'S' (0x53)							
Перед. данные 5	'T' (0x54)							
Перед. данные 6	'U' (0x55)							
Перед. данные 7	'V' (0x56)							
Перед. данные 8	'W' (0x57)							
Перед. данные 9	'X' (0x58)							
Перед. данные 10	'Y' (0x59)							
Перед. данные 11	'Z' (0x5A)							
Перед. данные 12	0x00							
Перед. данные 13	0x00							

Шаг 4

Проверить бит подтверждения передачи TA (TR=TA, значит передача завершена)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	0	0	0

6.3.2. Пример приёма данных

Необходимо принять следующий набор данных:

“Company:CREVIS G-Series“ (22 байта)

Размер «окон» приёма / передачи – 16 байт.

Шаг 0

RR = RA, т.е. вся последняя посылка принята успешно. При этом бит наличия новой посылки RE = 1.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	1	0	0	0	0

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	0	0

Шаг 1

Инвертировать бит подтверждения приёма RA (RA≠RR).

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	1	0	0

Шаг 2

Проверить состояние бита подтверждения приёма (RA=RR, значит приём первой части данных завершен). При этом в байт длины принимаемых данных запишется количество принятых данных (учитывая размер «окна» приёма, 14 байт), а в байты принимаемых данных запишутся первые 14 байт посылки, т.е. **“Company:CREVIS”**

Состояние бита RE=1 означает, что в буфере приёма ещё остались данные.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	1	0	1	0	0
Длина принятых данных	'14 (14 байт)							
Прин. данные 0	'C' (0x43:ASCII код)							
Прин. данные 1	'o' (0x6F)							
Прин. данные 2	'm' (0x6D)							
Прин. данные 3	'p' (0x70)							
Прин. данные 4	'a' (0x61)							

Прин. данные 5	'n' (0x6E)
Прин. данные 6	'y' (0x79)
Прин. данные 7	':' (0x3A)
Прин. данные 8	'C' (0x43)
Прин. данные 9	'R' (0x52)
Прин. данные 10	'E' (0x45)
Прин. данные 11	'V' (0x56)
Прин. данные 12	'I' (0x49)
Прин. данные 13	'S' (0x53)

Шаг 3

Для того, чтобы переместить оставшиеся данные из буфера приёма в байты «окна» приёма, необходимо ещё раз инвертировать бит подтверждения приёма RA (RA≠RR).

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	0	0

Шаг 4

По аналогии с шагов 2, проверить состояние бита подтверждения приёма (RA=RR, значит приём первой части данных завершен). При этом в байт длины принимаемых данных запишется количество принятых данных (оставшиеся 8 байт), а в байты принимаемых данных запишутся последние 8 байт посылки, т.е. "G-Series"

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	0	0	0
Длина принятых данных	(8 байт)							
Прин. данные 0	'G' (0x47:ASCII код)							
Прин. данные 1	'-' (0x2D)							
Прин. данные 2	'S' (0x53)							
Прин. данные 3	'e' (0x65)							
Прин. данные 4	'r' (0x72)							
Прин. данные 5	'i' (0x69)							
Прин. данные 6	'e' (0x65)							
Прин. данные 7	's' (0x73)							
Прин. данные 8	0x00							
Прин. данные 9	0x00							

Прин. данные 10	0x00
Прин. данные 11	0x00
Прин. данные 12	0x00
Прин. данные 13	0x00

6.3.3. Пример передачи и приёма данных

Необходимо принять из канала и передать в канал следующий набор данных: "CREVIS" (6 байт)

Размер «окон» приёма / передачи – 16 байт.

Шаг 1 (Передача данных)

Записать желаемую посылку в «окно» передачи, задать длину передаваемых данных 6 байт

Инvertировать флаг запроса передачи TR (TR≠TA)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	1	0
Длина перед. данных	06							
Перед. данные 0	'C' (0x43:ASCII код)							
Перед. данные 1	'R' (0x52)							
Перед. данные 2	'E' (0x45)							
Перед. данные 3	'V' (0x56)							
Перед. данные 4	'I' (0x49)							
Перед. данные 5	'S' (0x53)							
Перед. данные 6	0x00							
Перед. данные 7	0x00							
Перед. данные 8	0x00							
Перед. данные 9	0x00							
Перед. данные 10	0x00							
Перед. данные 11	0x00							
Перед. данные 12	0x00							
Перед. данные 13	0x00							

Шаг 2

Проверить бит подтверждения передачи TA (TR=TA, значит передача завершена)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	0	1	0

Шаг 3 (Приём данных)

RR = RA, т.е. вся последняя посылка принята успешно. При этом бит наличия новой посылки RE = 1.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	1	0	0	0	0

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	0	0

Шаг 4

Инвертировать бит подтверждения приёма RA (RA≠RR).

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	1	0	0

Шаг 5

Проверить состояние бита подтверждения приёма (RA=RR, значит приём данных завершен, RE=0, значит буфер приёма пустой). При этом в байт длины принимаемых данных запишется количество принятых данных (6 байт), а в байты принимаемых данных запишутся 6 байт посылки, т.е. "CREVIS"

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	1	0	0
Длина принятых данных	0 6							
Прин. данные 0	'C' (0x43:ASCII код)							
Прин. данные 1	'R' (0x52)							
Прин. данные 2	'E' (0x45)							
Прин. данные 3	'V' (0x56)							
Прин. данные 4	'I' (0x49)							
Прин. данные 5	'S' (0x53)							
Прин. данные 6	0x00							
Прин. данные 7	0x00							
Прин. данные 8	0x00							
Прин. данные 9	0x00							
Прин. данные 10	0x00							
Прин. данные 11	0x00							
Прин. данные 12	0x00							
Прин. данные 13	0x00							

6.3.4. Пример передачи данных с использованием функции обработки данных

Необходимо передать следующий набор данных:

CREVIS (6 байт)

Размер «окон» приёма / передачи – 16 байт.

Функция обработки данных подразумевает, что посылки из «окна» приёма / передачи изначально направляется в буфер передачи. Для включения этой функции необходимо установить в 1 соответствующий бит (Обработка данных 1 / 2) в таблице параметров модуля.

В примере ниже для иллюстрации работы данной функции необходимые для передачи данные будут записываться в буфер передачи последовательно за 2 передачи.

Шаг 1

Записать первые 3 байта в «окно» передачи, задать длину передаваемых данных 3 байта

Инвертировать флаг запроса передачи TR (TR≠TA)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса 0	TPA	FTA	FRA	RE	RBO	RR	TA	IA
	0	0	0	0	0	0	0	0
	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	1	0
Длина перед. данных	0 3							
Перед. данные 0	'C' (0x43: ASCII код)							
Перед. данные 1	'R' (0x52)							
Перед. данные 2	'E' (0x45)							
Перед. данные 3	0x00							
Перед. данные 4	0x00							
Перед. данные 5	0x00							
Перед. данные 6	0x00							
Перед. данные 7	0x00							
Перед. данные 8	0x00							
Перед. данные 9	0x00							
Перед. данные 10	0x00							
Перед. данные 11	0x00							
Перед. данные 12	0x00							
Перед. данные 13	0x00							

Шаг 2

Проверить бит подтверждения передачи TA (TR=TA, значит передача завершена)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса	TPA	FTA	FRA	RE	RBO	RR	TA	IA
0	0	0	0	0	0	0	1	0

После этого буфер передачи будет выглядеть следующим образом:

Байты данных	Буфер передачи
Перед. данные 0	'C' (0x43: ASCII код)
Перед. данные 1	'R' (0x52)
Перед. данные 2	'E' (0x45)
Перед. данные 3	0x00
Перед. данные 4	0x00
Перед. данные 5	0x00
...	0x00

Шаг 3

Записать оставшиеся 3 байта в «окно» передачи, задать длину передаваемых данных 3 байта

Инвертировать флаг запроса передачи TR (TR≠TA)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления 0	TPR	FT	FR	----	----	RA	TR	IR
	0	0	0	0	0	0	0	0
Длина перед. данных	0 3							
Перед. данные 0	'V' (0x56 ASCII код)							
Перед. данные 1	'I' (0x49)							
Перед. данные 2	'S' (0x53)							
Перед. данные 3	0x00							
Перед. данные 4	0x00							
Перед. данные 5	0x00							
Перед. данные 6	0x00							
Перед. данные 7	0x00							
Перед. данные 8	0x00							
Перед. данные 9	0x00							
Перед. данные 10	0x00							
Перед. данные 11	0x00							
Перед. данные 12	0x00							
Перед. данные 13	0x00							

Шаг 4

Проверить бит подтверждения передачи ТА (TR=ТА, значит передача завершена)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса	TPA	FTA	FRA	RE	RBO	RR	ТА	IA
0	0	0	0	0	0	0	0	0

После этого буфер передачи будет выглядеть следующим образом:

Байты данных	Буфер передачи
Перед. данные 0	'C' (0x43: ASCII код)
Перед. данные 1	'R' (0x52)
Перед. данные 2	'E' (0x45)
Перед. данные 3	'V' (0x56: ASCII код)
Перед. данные 4	'I' (0x49)
Перед. данные 5	'S' (0x53)
Перед. данные 6	0x00
Перед. данные 7	0x00
...	0x00

Шаг 5

Инvertировать флаг запроса передачи обработанных данных TPR (TPR≠TPA). При этом посылка из буфера передачи будет отправлена непосредственно в канал. При этом данные в буфере передачи будут очищены.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления	TPR	FT	FR	----	----	RA	TR	IR
0	1	0	0	0	0	1	0	0

Шаг 6

Проверить бит подтверждения передачи обработанных данных TPA (TPR=TPA, значит передача из буфера передачи в канал завершена)

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса	TPA	FTA	FRA	RE	RBO	RR	ТА	IA
0	1	0	0	0	0	0	0	0

6.3.5. Пример проверки переполнения буфера приёма

Предположим, что другое устройство отправляет на интерфейсный модуль GT-52xx посылку длины, превышающей размер буфера приёма, например, 1025 байт. В этом случае, так как размер буфера приёма равен 1024 байт, все данные после 1024 байта будут перезаписываться с начала, т.е.:

Другое устройство		GT-52xx	
Байты данных	Данные на передачу	Байты данных	Буфер приёма
Перед. данные 1	0x01	Прин. данные1	0
Перед. данные 2	0x02	Прин. данные2	0x02
Перед. данные 3	0x03	Прин. данные3	0x03
Перед. данные 4	0x04	Прин. данные4	0x04
Перед. данные 5	0x05	Прин. данные5	0x05
Перед. данные 6	0x06	Прин. данные6	0x06
.	.	.	
.	.	.	
.	.	.	
Перед. данные 1020	0x01	Прин. данные1019	0x10
Перед. данные 1021	0x02	Прин. данные1020	0x01
Перед. данные 1022	0x03	Прин. данные1021	0x02
Перед. данные 1023	0x04	Прин. данные1022	0x03
Перед. данные 1024	0x05	Прин. данные1023	0x04
Перед. данные 1025	0	Прин. данные1024	0x05

При приёме такой посылки последовательность действий с битами слова управления и слова статуса будет следующая:

Шаг 0

RR = RA, т.е. вся последняя посылка принята успешно. При этом бит наличия новой посылки RE = 1.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса	TPA	FTA	FRA	RE	RBO	RR	TA	IA
0	0	0	0	1	0	0	0	0

Шаг 1

Инвертировать бит подтверждения приёма RA (RA≠RR).

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления	TPR	FT	FR	---	---	RA	TR	IR
0	0	0	0	0	0	1	0	0

Шаг 2

Проверить состояние бита подтверждения приёма (RA=RR, значит приём данных завершен). При этом состояние бита RE=1 будет означать, что в буфере приёма ещё остались данные, а бит RBO=1 означать переполнение буфера приёма.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово статуса	TPA	FTA	FRA	RE	RBO	RR	TA	IA
0	0	0	0	1	1	1	0	0

Для очистки переполненного буфера приёма необходимо установить бит запроса инициализации IR=1.

	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1	Бит 0
Слово управления	TPR	FT	FR	----	----	RA	TR	IR
0	0	0	0	0	0	0	0	1

После этого все байты данного буфера будут очищены.

7. Изменение размера «окна» приёма / передачи в файлах-описателях модулей (XML)

7.1. Утилита I/O Guide Pro

Файлы-описатели располагаются в папке Products/IO и имеют следующий вид:

- для одноканальных интерфейсных модулей

```
<IOData InputLength="16" OutputLength="16">
  <Input OneChSize="8">
    <Ref TextId="T_Status_Data_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Length_Data_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_01" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_02" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_03" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_04" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_05" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_06" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_07" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_08" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_09" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_10" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_11" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_12" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_13" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    Write <Ref TextId="T_Byte_InData_XX" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
  </Input>
  <Output OneChSize="8">
    <Ref TextId="T_Ctrl_Data_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Length_Data_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_01" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_02" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_03" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_04" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_05" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_06" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_07" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_08" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_09" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_10" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_11" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_12" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_13" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    Write <Ref TextId="T_Byte_OutData_XX" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
  </Output>
</IOData>
```

- для двухканальных интерфейсных модулей

```
<IOData InputLength="16" OutputLength="16">
  <Input OneChSize="8">
    <Ref TextId="T_Status_Data_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Length_Data_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_01" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_02" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_03" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_04" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_05" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    Write <Ref TextId="T_Byte_InData_XX" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/> 1Ch Area
    <Ref TextId="T_Status_Data_01" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Length_Data_01" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_00" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_01" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_02" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_03" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_04" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_InData_05" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    Write <Ref TextId="T_Byte_InData_XX" ChannelType="BI" DataType="ByteArea" Length="1" Unit="" Visible="true"/> 2Ch Area
  </Input>
  <Output OneChSize="8">
    <Ref TextId="T_Ctrl_Data_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Length_Data_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_01" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_02" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_03" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_04" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_05" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    Write <Ref TextId="T_Byte_OutData_XX" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/> 1Ch Area
    <Ref TextId="T_Ctrl_Data_01" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Length_Data_01" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_00" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_01" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_02" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_03" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_04" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    <Ref TextId="T_Byte_OutData_05" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/>
    Write <Ref TextId="T_Byte_OutData_XX" ChannelType="BO" DataType="ByteArea" Length="1" Unit="" Visible="true"/> 2Ch Area
  </Output>
</IOData>
```

Шаг 1

Изменить объём данных в «окнах» приёма / передачи - первая строчка.

```
<IOData InputLength="16" OutputLength="16">
```

Например, для «окна» размером 30 байт эта строчка должна выглядеть, как:

```
<IOData InputLength="30" OutputLength="30">
```

Шаг 2

Добавить необходимое количество строк с описаниями дополнительных байт данных.

Например, для «окна» размером 18 байт (т.е. на 2 байта данных больше, чем по умолчанию), необходимо добавить:

- для одноканальных интерфейсных модулей

После строк описания принимаемых данных:

```
<Ref TextId="T_Byte_InData_14" ChannelType="BI" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
<Ref TextId="T_Byte_InData_15" ChannelType="BI" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
```

После строк описания передаваемых данных:

```
<Ref TextId="T_Byte_OutData_14" ChannelType="BO" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
<Ref TextId="T_Byte_OutData_15" ChannelType="BO" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
```

- для двухканальных интерфейсных модулей

После строк описания принимаемых данных Канала 0:

```
<Ref TextId="T_Byte_InData_06" ChannelType="BI" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
```

После строк описания принимаемых данных Канала 1:

```
<Ref TextId="T_Byte_InData_06" ChannelType="BI" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
```

После строк описания передаваемых данных Канала 0:

```
<Ref TextId="T_Byte_OutData_06" ChannelType="BO" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
```

После строк описания передаваемых данных Канала 1:

```
<Ref TextId="T_Byte_OutData_06" ChannelType="BO" DataType="ByteArea" Length="1" Unit=""
Visible="true"/>
```


7.2. Среда разработки CODESYS

Файлы-описатели интерфейсных модулей для CODESYS содержат следующие строки:

```
<Parameter ParameterId="1000" type="localTypes:ARRAY [0..15] OF TBit1Byte">
  <Attributes channel="input" download="true" functional="false" offlineaccess="read" onlineaccess="read" />
  <Default>0</Default>
  <Name name="local:in0">IN</Name>
</Parameter>

<Parameter ParameterId="2000" type="localTypes:ARRAY [0..15] OF TBit1Byte">
  <Attributes channel="output" download="true" functional="false" offlineaccess="readwrite" onlineaccess="readwrite" />
  <Default>0</Default>
  <Name name="local:out0">OUT</Name>
</Parameter>
```

Для изменения размера «окна» приёма / передачи необходимо изменить значения в квадратных скобках. Например, для «окна» размером 30 байт необходимо «[0..15]» изменить на «[0..29]». После этого измененный файл-описатель должен быть заново загружен в CODESYS.

8. Пример работы с интерфейсными модулями через библиотеку CBL (Modbus RTU, CODESYS)

В примере ниже описывается приём и передача данных через интерфейсный модуль GT-5232 (RS-485, 2 канала) по протоколу Modbus RTU с помощью функциональных блоков библиотеки примеров CBL.

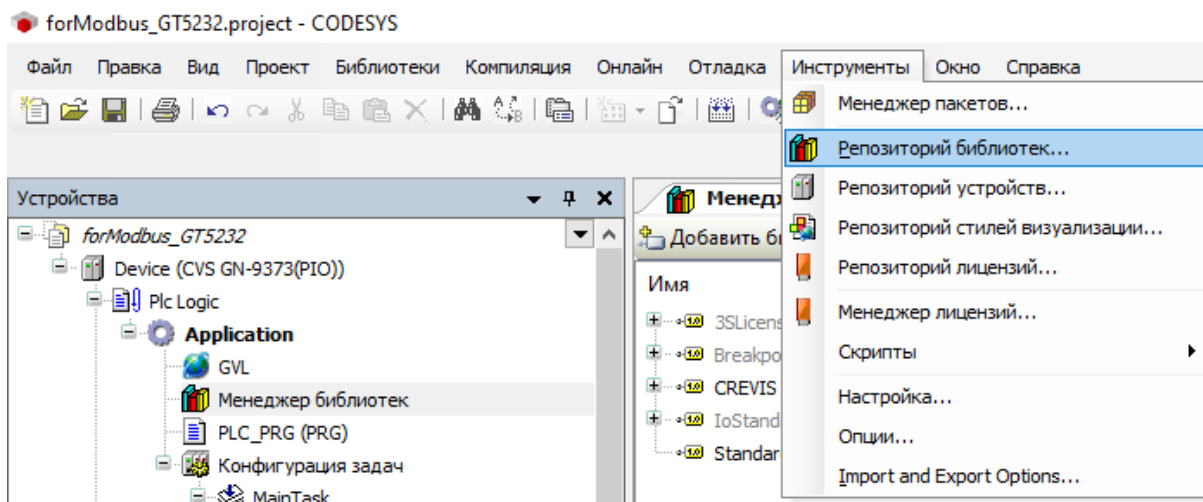
В данном примере используется программируемый адаптер GN-9373 и интерфейсный модуль GT-5232 в качестве Modbus RTU Master и программируемый логический контроллер REGUL R200 CU 00 041 (Производитель «Прософт-Системы») в качестве Modbus RTU Slave.



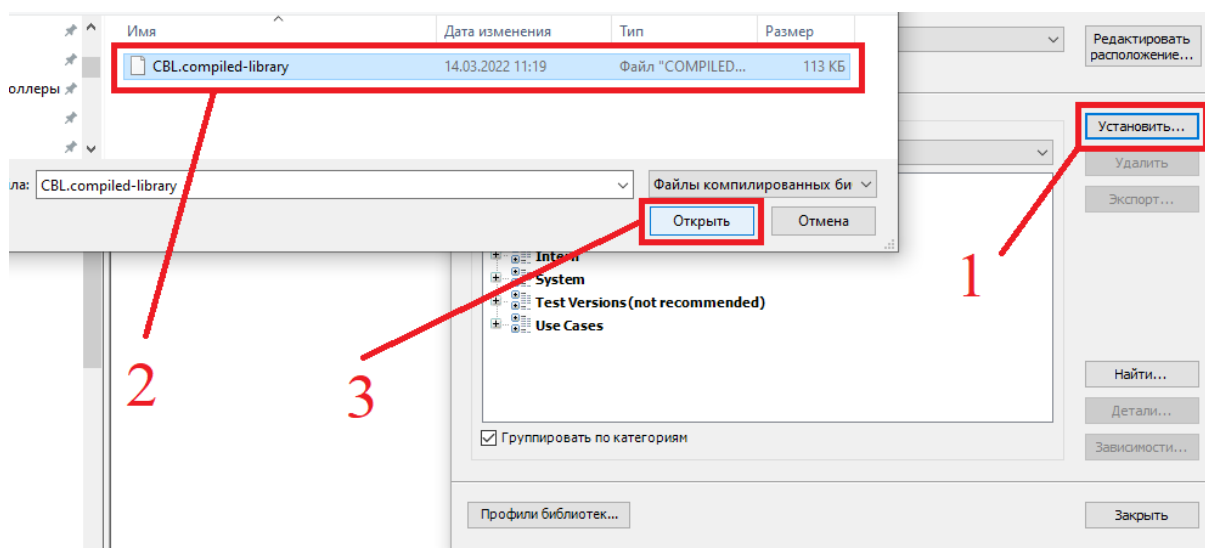
Примечание. .csv файл для автоматического соотнесения переменных, а также демонстрируемый пример предоставляются по запросу.

Для приёма и передачи данных по Modbus RTU предлагается следующий алгоритм:

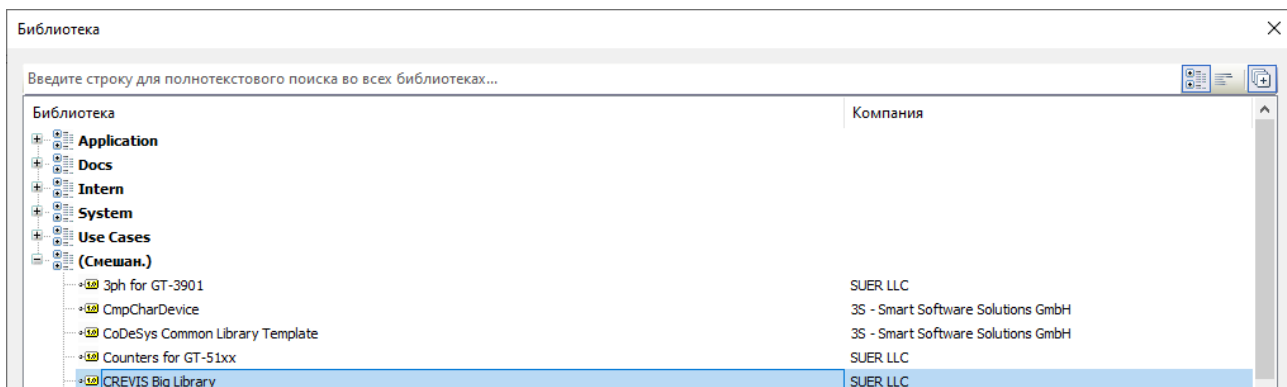
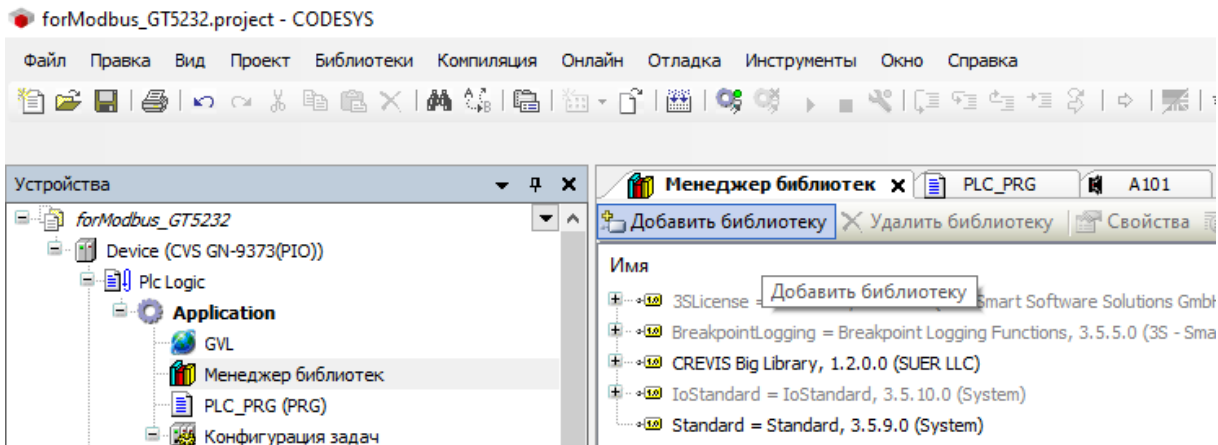
- 1) В проекте адаптера GN-9373 в CODESYS в меню «Инструменты» - «Репозиторий библиотек» добавить библиотеку примеров CBL в общий репозиторий библиотек среды разработки;



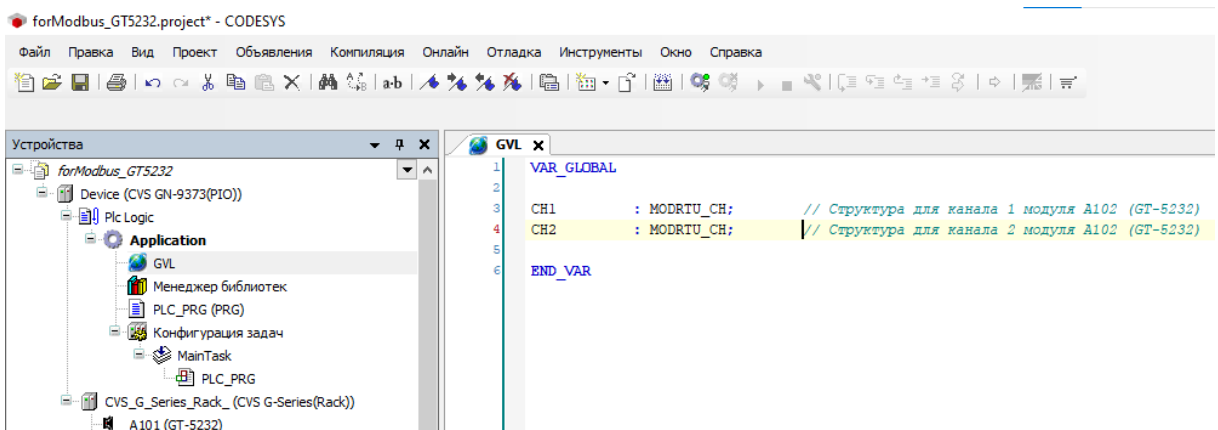
Для этого в данном меню предлагается нажать кнопку «Установить», выбрать файл компилированной библиотеки и нажать кнопку «Открыть».



- 2) В проекте адаптера GN-9373 в CODESYS в элементе «Менеджер библиотек» необходимо добавить библиотеку CBL в проект. Для этого предлагается нажать кнопку «Добавить библиотеку», выбрать библиотеку «CREVIS Big Library» и нажать кнопку «ОК»;



- 3) В программе создать структуры для каналов интерфейсного модуля;

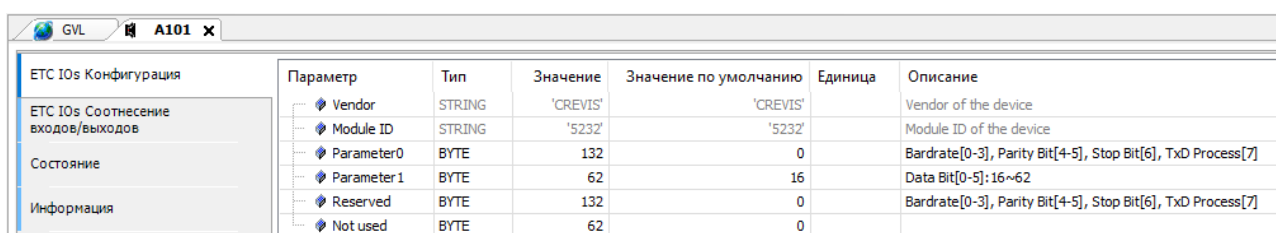


- 4) В настройках интерфейсного модуля в меню «Конфигурация» задать необходимые значения для параметров обмена каналов (Скорость обмена, Чётность и т.д.) в соответствии с п.б.2. и размер массивов для управления «окнами» приёма и передачи. В примере ниже слово параметров обмена для обоих каналов равно **132**: Скорость обмена **9600** бод, Без контроля чётности (**None**), 1 стоповый бит.

Примечание 1. Для корректной работы функциональных блоков библиотеки бит «Обработка данных 1 / 2» (7й бит) **должен** быть установлен в **1**.

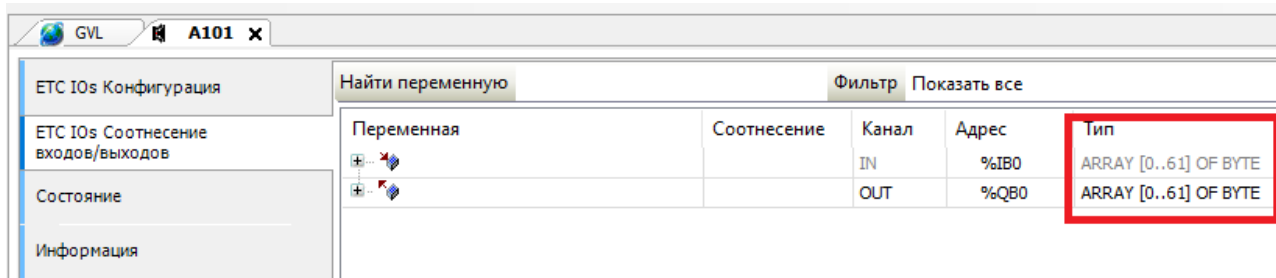
Примечание 2. Для корректной работы функциональных блоков библиотеки размер массивов для управления «окнами» приёма и передачи должен быть равен **62** для двухканальных модулей и **31** для одноканальных модулей.

Примечание 3. При изменении параметров модуль должен быть перезагружен по питанию.



Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Vendor	STRING	'CREVIS'	'CREVIS'		Vendor of the device
Module ID	STRING	'5232'	'5232'		Module ID of the device
Parameter0	BYTE	132	0		Bardrate[0-3], Parity Bit[4-5], Stop Bit[6], TxD Process[7]
Parameter1	BYTE	62	16		Data Bit[0-5]: 16~62
Reserved	BYTE	132	0		Bardrate[0-3], Parity Bit[4-5], Stop Bit[6], TxD Process[7]
Not used	BYTE	62	0		

- 5) В настройках интерфейсного модуля в меню «Соотнесение входов/выходов» необходимо убедиться, что массивы IN и OUT, используемые для управления «окнами» приёма/передачи имеют размерность [0..61], как показано на рисунке ниже.



Переменная	Соотнесение	Канал	Адрес	Тип
		IN	%IB0	ARRAY [0..61] OF BYTE
		OUT	%QB0	ARRAY [0..61] OF BYTE

Далее необходимо соотнести элементы данных массивов с полями созданных ранее структур в следующем порядке:

- IN[0] / IN[31] – слова состояния для канала 1 и канала 2, соответственно;
- IN[1] / IN[32] – длина принятых данных для канала 1 и канала 2, соответственно;

Переменная	Соотнесение	Канал	Адрес	Тип
		IN	%IB0	ARRAY
		IN[0]	%IB0	BYTE
Application.CH1.State.IA		BIT0	%IX0-0	BOOL
Application.CH1.State.TA		BIT1	%IX0-1	BOOL
Application.CH1.State.RR		BIT2	%IX0-2	BOOL
Application.CH1.State.RBO		BIT3	%IX0-3	BOOL
Application.CH1.State.RE		BIT4	%IX0-4	BOOL
Application.CH1.State.FRA		BIT5	%IX0-5	BOOL
Application.CH1.State.FTA		BIT6	%IX0-6	BOOL
Application.CH1.State.TRA		BIT7	%IX0-7	BOOL
Application.CH1.RxLength		IN[1]	%IB1	BYTE

- IN[2] – IN[30], IN[33] – IN[61] – байты принимаемых данных (29 байт) для канала 1 и канала 2, соответственно;

Переменная	Соотнесение	Канал	Адрес	Тип
		IN	%IB0	ARRAY [0,
		IN[0]	%IB0	BYTE
Application.CH1.RxLength		IN[1]	%IB1	BYTE
Application.CH1.RxData0		IN[2]	%IB2	BYTE
Application.CH1.RxData1		IN[3]	%IB3	BYTE
Application.CH1.RxData2		IN[4]	%IB4	BYTE
Application.CH1.RxData3		IN[5]	%IB5	BYTE
Application.CH1.RxData4		IN[6]	%IB6	BYTE
Application.CH1.RxData5		IN[7]	%IB7	BYTE
Application.CH1.RxData6		IN[8]	%IB8	BYTE

- OUT[0] / OUT[31] – слова управления для канала 1 и канала 2, соответственно;
- OUT[1] / OUT[32] – длина передаваемых данных для канала 1 и канала 2, соответственно;

Переменная	Соотнесение	Канал	Адрес	Тип
		IN	%IB0	ARRAY
		OUT	%QB0	ARRAY
		OUT[0]	%QB0	BYTE
Application.CH1.Control.IR		BIT0	%QX0.0	BOOL
Application.CH1.Control.TR		BIT1	%QX0.1	BOOL
Application.CH1.Control.RA		BIT2	%QX0.2	BOOL
Application.CH1.Control.RES1		BIT3	%QX0.3	BOOL
Application.CH1.Control.RES2		BIT4	%QX0.4	BOOL
Application.CH1.Control.FR		BIT5	%QX0.5	BOOL
Application.CH1.Control.FT		BIT6	%QX0.6	BOOL
Application.CH1.Control.TPR		BIT7	%QX0.7	BOOL
Application.CH1.TxLength		OUT[1]	%QB1	BYTE

- OUT[2] – OUT[30], OUT[33] – OUT[61] – байты передаваемых данных (29 байт) для канала 1 и канала 2, соответственно.

Переменная	Соотнесение	Канал	Адрес	Тип
		IN	%IB0	ARRAY
		OUT	%QB0	ARRAY
		OUT[0]	%QB0	BYTE
Application.CH1.TxLength		OUT[1]	%QB1	BYTE
Application.CH1.TxData0		OUT[2]	%QB2	BYTE
Application.CH1.TxData1		OUT[3]	%QB3	BYTE
Application.CH1.TxData2		OUT[4]	%QB4	BYTE
Application.CH1.TxData3		OUT[5]	%QB5	BYTE
Application.CH1.TxData4		OUT[6]	%QB6	BYTE
Application.CH1.TxData5		OUT[7]	%QB7	BYTE
Application.CH1.TxData6		OUT[8]	%QB8	BYTE

Примечание. Соотнесение может быть произведено автоматически с помощью функций импорта/экспорта с использованием .csv файла.

- 6) В программе контроллера создать экземпляр функционального блока *MODRTU_REQ* из библиотеки СBL, массивы для принимаемых/передаваемых по Modbus RTU данных, а также вспомогательные переменные для формирования импульса на запуск приёма/передачи (в примере ниже это булевая переменная *bool1* для формирования одиночного импульса и таймер *TON_1s* для формирования периодических импульсов);

```

PLC_PRG x
1  PROGRAM PLC_PRG
2  VAR
3
4  bool1   : BOOL;           // Бит для формирования импульса (одиночный опрос)
5
6  TON_1s  : TON := (PT := TIME#1SOMS); // Таймер для формирования периодических импульсов (периодически)
7
8  ModReq_FB : MODRTU_REQ := (usiUnitID := 1, ModbusCommand := (FunctionCode := 3, uiReadLen := 1));
9  //ModCont_FB : MODRTU_CONT := (SlavesNumber := 2);
10
11 RecvData : ARRAY [0..100] OF WORD;           // Массив с принимаемыми по Modbus
12 SendData : ARRAY [0..100] OF WORD := [14, 4(0), 1794, 95(0)]; // Массив с отправляемыми по Modbus
13
14
15 // Реализация формирования периодических импульсов
16
17 TON_1s (IN := NOT(TON_1s.Q));
18
19 // Вызов ФБ для обычного опроса
20
21 ModReq_FB (pChannel := ADR(CH1), xExecute := bool1, pSendData := ADR(SendData[0]), pRecvData := ADR(RecvData));
22
23 // Вызов ФБ для периодического опроса
24
25 //ModCont_FB (pChannel := ADR(CH1), xExecute := TON_1s.Q, SlavesDescription := ADR(Slaves1), pwData := ADR(RecvData));
26
27 // Сброс бита для формирования одиночного импульса
28
29 bool1 := FALSE;

```

Далее в коде программы вызвать созданный экземпляр функционального блока *MODRTU_REQ*. В примере выше в данный ФБ передаётся указатель на структуру канала 1 (так как Modbus RTU Slave подключен к этому каналу), булевая переменная *bool1* для выполнения команды по её положительному фронту и указатели на массивы для принятых/передаваемых данных. Значение таймаута выполнения команды оставлено по умолчанию (1 секунда), идентификатор (адрес) Modbus устройства и параметры Modbus команды задаются в области инициализации.

Полный список входных/выходных параметров данного функционального блока приведен в таблицах ниже:

Таблица 1. Входные параметры ФБ MODRTU_REQ

Параметр	Тип данных	Описание
pChannel	POINTER TO MODRTU_CH	Указатель на структуру канала модуля GT-523x
xExecute	BOOL	Выполнение функции (по положительному фронту)
xAbort	BOOL	Прерывание выполнения функции (по положительному фронту)
usiUnitID	UINT	Идентификатор (адрес) Modbus устройства
ModbusCommand	Modbus_Command	Структура с данными Modbus команды
pRecvData	POINTER TO WORD	Указатель на буфер с данными на запись
pSendData	POINTER TO WORD	Указатель на буфер с данными чтения
tTimeout	TIME	Таймаут ожидания ответа устройства

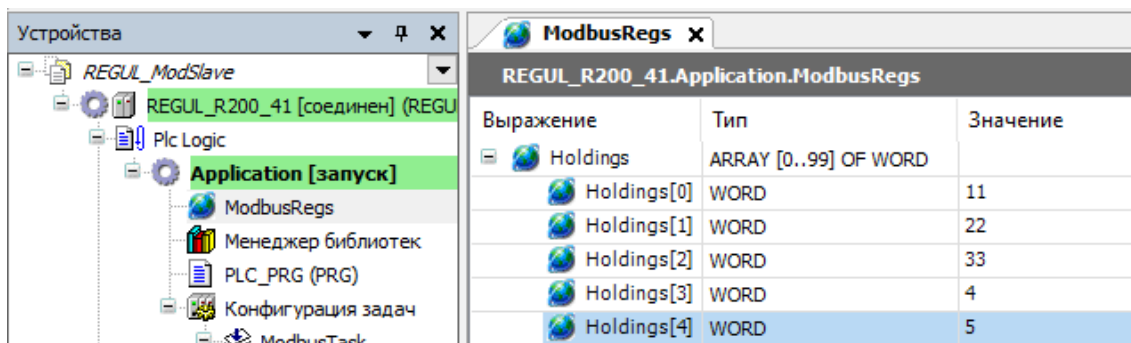
Таблица 2. Выходные параметры ФБ MODRTU_REQ

Параметр	Тип данных	Описание
xDone	BOOL	Выполнение команды завершилось успешно
xError	BOOL	Ошибка выполнения команды
Timeout_Flag	BOOL	Таймаут выполнения
CRC_Err_Flag	BOOL	Ошибка CRC
State	STRING	Состояния выполнения команды

Примечание. В приведенном примере программа выполняется в циклической задаче MainTask с периодом вызова 20 мс.

- 7) Загрузить проект с разработанной программой в контроллер.

В ПЛК REGUL R200 для проверки создано четыре массива для четырёх областей Modbus памяти. У массива Holdings для регистров 0-4 заданы значения, отличные от нуля.

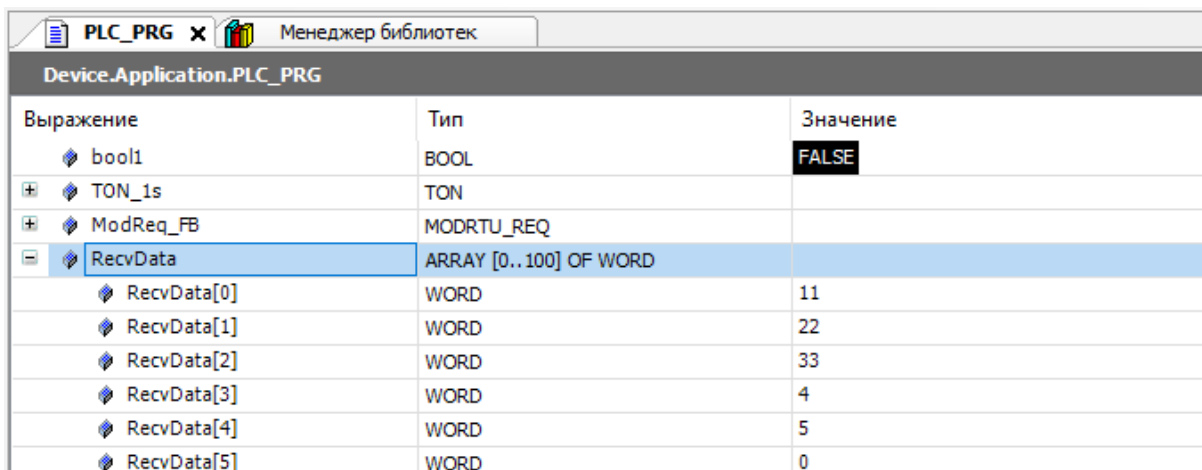


Выражение	Тип	Значение
Holdings	ARRAY [0..99] OF WORD	
Holdings[0]	WORD	11
Holdings[1]	WORD	22
Holdings[2]	WORD	33
Holdings[3]	WORD	4
Holdings[4]	WORD	5

- 8) Во входной структуре ModbusCommand задать значения для чтения пяти регистров типа Holding Registers (код функции 3), начиная с нулевого.

Выражение	Тип	Значение
ModbusCommand	Modbus_Command	
FunctionCode	UINT	3
uiReadOffset	UINT	0
uiReadLen	UINT	5

- 9) Установить бит выполнения программы bool1 в TRUE. После этого произойдет однократное выполнение заданной команды и массив для принимаемых данных будет иметь вид:



Выражение	Тип	Значение
bool1	BOOL	FALSE
TON_1s	TON	
ModReq_FB	MODRTU_REQ	
RecvData	ARRAY [0..100] OF WORD	
RecvData[0]	WORD	11
RecvData[1]	WORD	22
RecvData[2]	WORD	33
RecvData[3]	WORD	4
RecvData[4]	WORD	5
RecvData[5]	WORD	0

- 10) Для варианта использования интерфейсного модуля, когда на канал присоединяется несколько Modbus устройств, предлагается использовать функциональный блок *MODRTU_CONT*. Для этого в программе контроллера создать экземпляр функционального блока *MODRTU_CONT* из библиотеки CBL, массивы для принимаемых/передаваемых по Modbus RTU данных, вспомогательные переменные для формирования импульса на запуск приёма/передачи, а также массив со структурами для описания Modbus устройств;

```

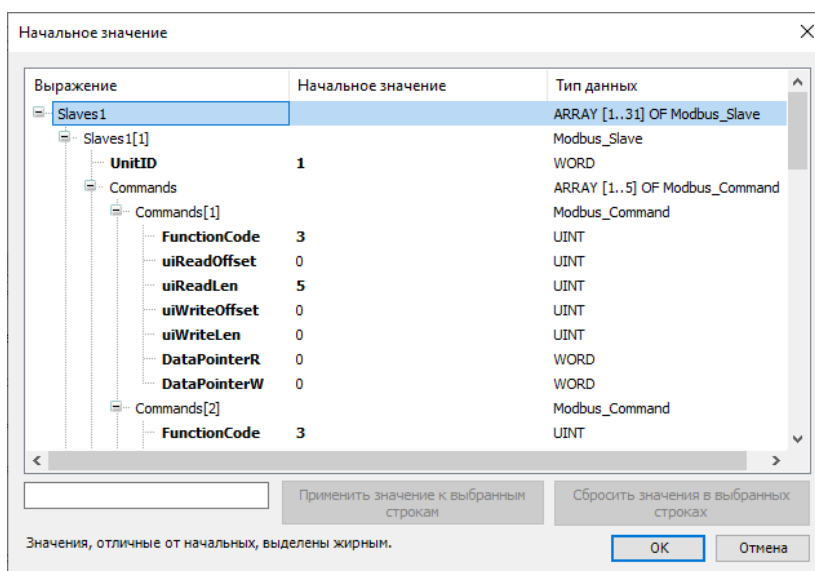
PLC_PRG x Менеджер библиотек
1 PROGRAM PLC_PRG
2 VAR
3
4 bool1 : BOOL; // Бит для формирования импульса (одиночный опрос)
5
6 TON_1s : TON := (PT := TIME#1S0MS); // Таймер для формирования периодических импульсов (периодический опрос,
7
8 //ModReq_FB : MODRTU_REQ := (usiUnitID := 1, ModbusCommand := (FunctionCode := 3, uiReadLen := 1)); // Экземпляр
9 ModCont_FB : MODRTU_CONT := (SlavesNumber := 2); // Экземпляр
10
11 RecvData : ARRAY [0..100] OF WORD; // Массив с принимаемыми по Modbus данными
12 SendData : ARRAY [0..100] OF WORD := [14, 4(0), 1794, 95(0)]; // Массив с отправляемыми по Modbus данными
13
14 // Описание Modbus слейвов для последовательного опроса
15
16 Slaves1 : ARRAY [1..31] OF Modbus_Slave := [(UnitID := 1, Commands := [(FunctionCode := 3, uiReadLen := 5),
17
18 END_VAR

1 // Реализация формирования периодических импульсов
2
3 TON_1s (IN := NOT(TON_1s.Q));
4
5 // Вызов ФБ для обычного опроса
6
7 //ModReq_FB (pChannel := ADR(CH1), xExecute := bool1, pSendData := ADR(SendData[0]), pRecvData := ADR(RecvData[0]));
8
9 // Вызов ФБ для периодического опроса
10
11 ModCont_FB (pChannel := ADR(CH1), xExecute := TON_1s.Q, SlavesDescription := ADR(Slaves1), pwDataRead := ADR(RecvData));
12
13 // Сброс бита для формирования одиночного импульса
14
15 bool1 := FALSE;

```

В структуре описания Modbus устройства задаётся его идентификатор (адрес), а также массив с пятью Modbus командами для этого устройства, которые будут выполняться последовательно. Если необходимо меньшее количество команд (например, две), то предлагается оставить ненужные структуры с командами незаполненными, эти команды будут пропущены и не будут участвовать в общем опросе. Если необходимо большее количество команд, то предлагается соответствующим образом изменить код функционального блока.

Для каждой конкретной команды, помимо указания кода функции, смещения, количества регистров и т.д., также указываются смещения (в элементах массива) для принимаемых/передаваемых данных общих массивов. Пример указания смещений см. далее.



Примечание. Данный функциональный блок при обнаружении потери связи с Modbus устройством (например, три последовательных ошибки передачи, уставка количества ошибок является входным параметром ФБ) отключает данное устройство от общего обмена на определённое количество циклов опроса (данное количество является входным параметром ФБ).

Полный список входных параметров данного функционального блока приведен в таблицах ниже:

Таблица 3. Входные параметры ФБ MODRTU_CONT

Параметр	Тип данных	Описание
pChannel	POINTER TO MODRTU_CH	Указатель на структуру канала модуля GT-523x
xExecute	BOOL	Выполнение функции (по положительному фронту)
SlavesNumber	BYTE	Количество Modbus устройств
SlavesDescription	POINTER TO ARRAY [1..31] OF Modbus_Slave	Указатель на описание Modbus устройств
pwDataRead	POINTER TO WORD	Указатель на буфер с данными чтения
pwDataWrite	POINTER TO WORD	Указатель на буфер с данными на запись
ErrorCountMax	WORD	Количество ошибок, при котором Modbus устройство отключается от общего опроса (по умолчанию, 3 ошибки подряд)
ErrorCycles	WORD	Количество циклов опроса, на которое устройство отключается от общего опроса (по умолчанию, 100 циклов)
tTimeout	TIME	Таймаут ожидания ответа устройства

11) Загрузить проект с разработанной программой в контроллер.

В массиве с описанием Modbus устройств в данном примере задано два устройства: у устройства с идентификатором **1** задано 5 команд:

- Считать **5** Holding Registers, начиная с **нулевого**, и поместить результат в массив с принимаемыми данными, начиная с **нулевого** элемента:

Начальное значение	
Выражение	На
Slaves1	
Slaves1[1]	
UnitID	1
Commands	
Commands[1]	
FunctionCode	3
uiReadOffset	0
uiReadLen	5
uiWriteOffset	0
uiWriteLen	0
DataPointerR	0
DataPointerW	0
Commands[2]	

- Считать **5** Holding Registers, начиная с **пятого**, и поместить результат в массив с принимаемыми данными, начиная с **пятого** элемента:

Начальное значение	
Выражение	На
Slaves1	
Slaves1[1]	
UnitID	1
Commands	
Commands[1]	
Commands[2]	
FunctionCode	3
uiReadOffset	5
uiReadLen	5
uiWriteOffset	0
uiWriteLen	0
DataPointerR	5
DataPointerW	0
Commands[3]	

- Записать в **1** Holding Registers, начиная с **девятого**, значение из массива с передаваемыми данными, начиная с **нулевого** элемента:

Начальное значение

Выражение	На
Slaves1	
Slaves1[1]	
UnitID	1
Commands	
Commands[1]	
Commands[2]	
Commands[3]	
FunctionCode	6
uiReadOffset	0
uiReadLen	0
uiWriteOffset	9
uiWriteLen	1
DataPointerR	0
DataPointerW	0

- Записать в **10** Coils, начиная с **пятого**, значения из массива с передаваемыми данными, начиная с **пятого** элемента:

Начальное значение

Выражение	На
UnitID	1
Commands	
Commands[1]	
Commands[2]	
Commands[3]	
Commands[4]	
FunctionCode	15
uiReadOffset	0
uiReadLen	0
uiWriteOffset	5
uiWriteLen	10
DataPointerR	0
DataPointerW	5

- Записать в **5** Holding Registers, начиная с **двадцатого**, значения из массива с передаваемыми данными, начиная с **десятого** элемента:

Начальное значение	
Выражение	Начальное значение
Commands	
Commands[1]	
Commands[2]	
Commands[3]	
Commands[4]	
Commands[5]	
FunctionCode	16
uiReadOffset	0
uiReadLen	0
uiWriteOffset	20
uiWriteLen	5
DataPointerR	0
DataPointerW	10

У устройства с идентификатором **2** (физически отсутствует) задана одна команда, аналогичная первой команде первого устройства.

Массив для передаваемых данных имеет следующий вид:

Начальное значение	
Выражение	Начальное значение
SendData	
SendData[0]	14
SendData[1]	0
SendData[2]	1794
SendData[3]	0
SendData[4]	1
SendData[5]	2
SendData[6]	3
SendData[7]	4
SendData[8]	5

В ПЛК REGUL R200 массив с данными Holding Registers имеет вид:

Выражение	Тип	Значение
Holdings	ARRAY [0..99] OF WORD	
Holdings[0]	WORD	11
Holdings[1]	WORD	22
Holdings[2]	WORD	33
Holdings[3]	WORD	4
Holdings[4]	WORD	5
Holdings[5]	WORD	122
Holdings[6]	WORD	123
Holdings[7]	WORD	124
Holdings[8]	WORD	125
Holdings[9]	WORD	126

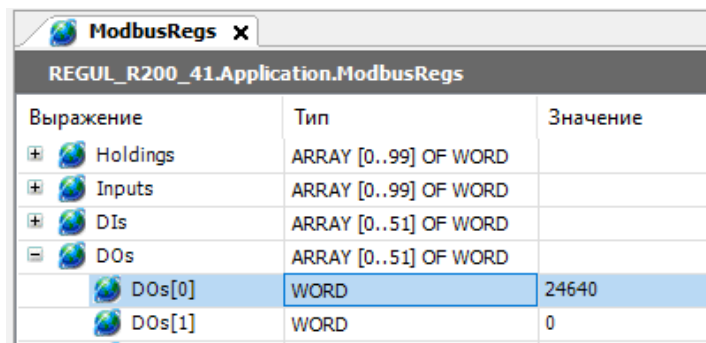
После загрузки программы и первого цикла опроса массив для принимаемых данных имеет вид (результат выполнения команд 1,2 и 3):

Выражение	Тип	Значение
bool1	BOOL	FALSE
TON_1s	TON	
ModCont_FB	MODRTU_CONT	
RecvData	ARRAY [0..100] OF WORD	
RecvData[0]	WORD	11
RecvData[1]	WORD	22
RecvData[2]	WORD	33
RecvData[3]	WORD	4
RecvData[4]	WORD	5
RecvData[5]	WORD	122
RecvData[6]	WORD	123
RecvData[7]	WORD	124
RecvData[8]	WORD	125
RecvData[9]	WORD	14

Массив с данными Holding Registers на ПЛК REGUL R200 имеет вид (результат выполнения команд 3 и 5):

Holdings[19]	WORD	0
Holdings[20]	WORD	1
Holdings[21]	WORD	2
Holdings[22]	WORD	3
Holdings[23]	WORD	4
Holdings[24]	WORD	5

Массив с данными Coils на ПЛК REGUL R200 имеет вид (результат выполнения команды 4):

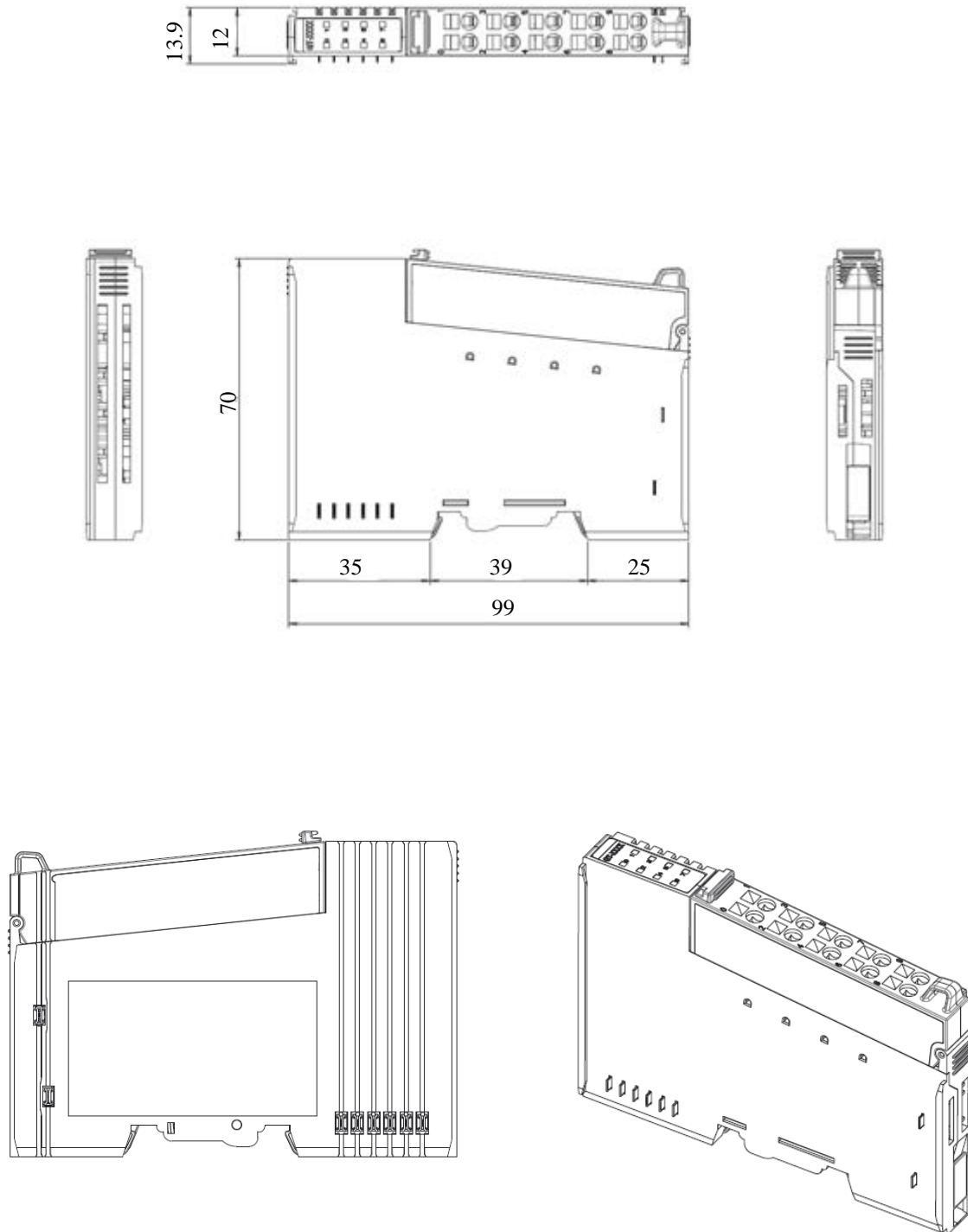


The screenshot shows a window titled "ModbusRegs" with a sub-header "REGUL_R200_41.Application.ModbusRegs". It contains a table with three columns: "Выражение" (Expression), "Тип" (Type), and "Значение" (Value). The table lists several data points, with "DOs[0]" and "DOs[1]" expanded to show their individual values.

Выражение	Тип	Значение
Holdings	ARRAY [0..99] OF WORD	
Inputs	ARRAY [0..99] OF WORD	
DIs	ARRAY [0..51] OF WORD	
DOs	ARRAY [0..51] OF WORD	
DOs[0]	WORD	24640
DOs[1]	WORD	0

9. Габариты

9.1. GT-52xx (RTV)



10. Монтаж

Осторожно!

Горячая поверхность!

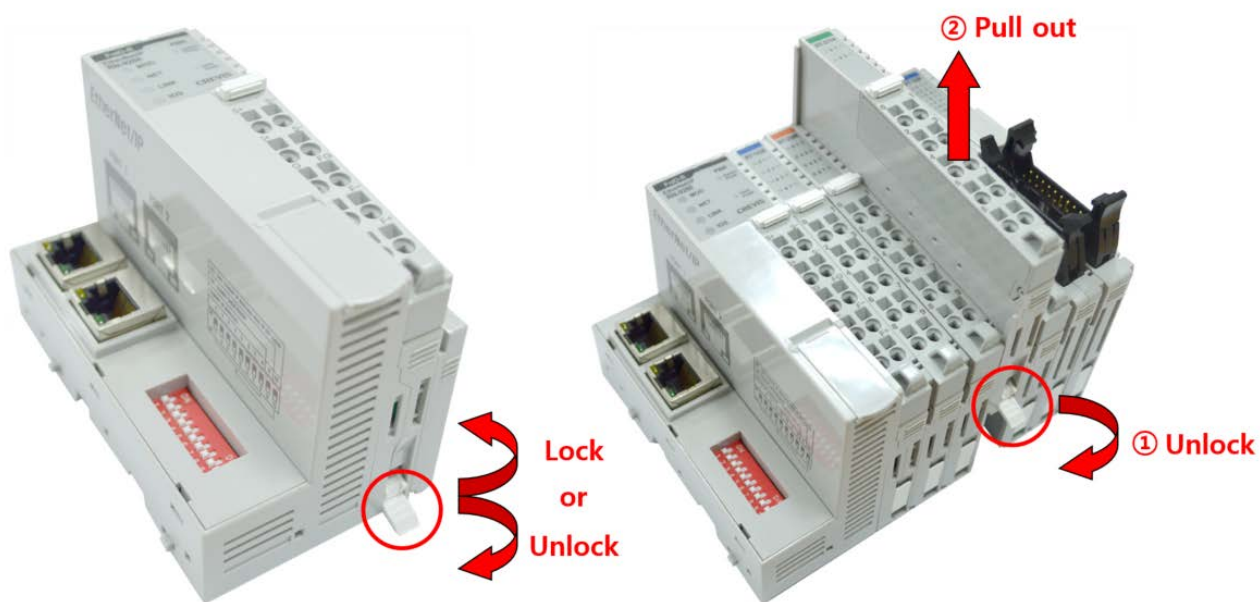
Во время работы поверхность корпуса может нагреваться. Если устройство эксплуатировалось при высоких температурах окружающей среды, дайте ему остыть, прежде чем прикасаться к нему.

Предупреждение!

Работы с приборами производить только в обесточенном состоянии!

Работа с устройствами под напряжением может привести к их повреждению, поэтому перед работой отключайте питание на устройствах.

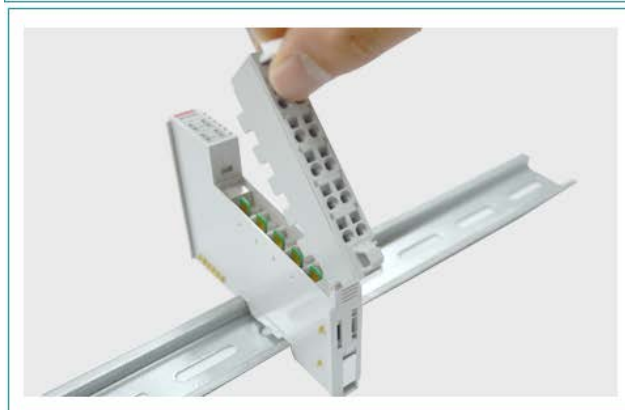
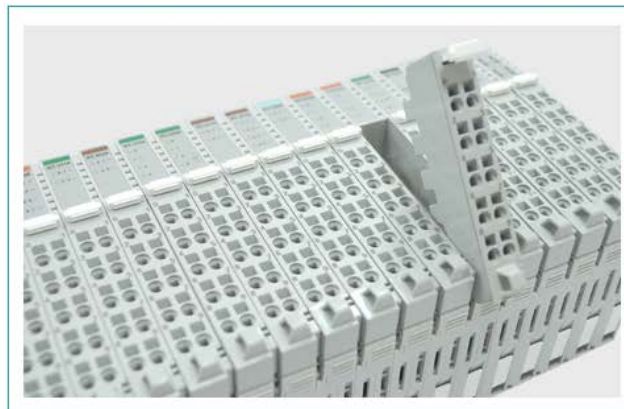
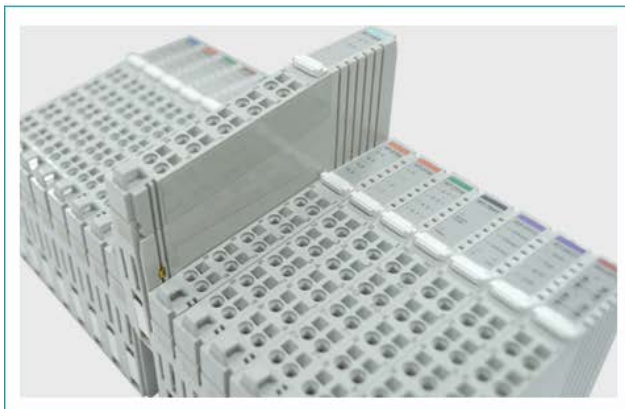
10.1. Монтаж и демонтаж модулей



Как показано на рисунке выше (слева), для фиксации модуля серии G его следует закрепить на DIN-рейке фиксирующими защёлками. Для этого откиньте верхнюю часть фиксирующей защёлки.

Чтобы вытащить модуль серии G, откройте фиксирующую защёлку, как показано на рисунке выше (справа).

10.2. RTB (Съёмный клеммный блок)

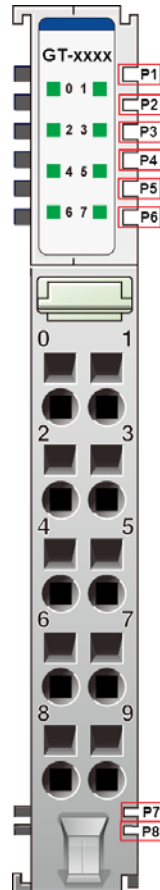


Для удобства монтажа вся клеммная колодка может быть снята, как это показано на рисунке выше.

На RTB в верхней части колодки есть фиксирующий рычажок для её лёгкого снятия.

11. Описание контактов шины G-Bus

Обмен данными между адаптерами серии G (FnIO и PIO) и модулями расширения, а также системное/полевое питание осуществляется через внутреннюю шину G-Bus. Данная шина состоит из 8 контактов (P1 - P8):



№	Описание
P1	Системное питание (VCC)
P2	Системное питание (GND)
P3	GBUS TX +
P4	GBUS TX -
P5	GBUS RX +
P6	GBUS RX -
P7	Полевое питание (GND)
P8	Полевое питание (VCC)

DANGER



Не прикасайтесь к контактам шины G-Bus, чтобы избежать воздействия помех и повреждений устройства от ESD шума.