

Следите за обновлениями документации!

Последнюю версию данной инструкции можно скачать по [ссылке](#)

Архив данных к инструкции: [https://static.dwin.pro/share/guides\\_ru/DWIN\\_DGUS\\_ModBus.zip](https://static.dwin.pro/share/guides_ru/DWIN_DGUS_ModBus.zip)

## Порты UART, RS-485 и ModBus RTU

### История изменений

вер	дата	изменения	автор
1	30.09.2022	создание документа, описан режим работы ModBus slave	Мирослав Кириллин
2	06.10.2022	правки и добавлено описание режима работы ModBus master и примеры из серии “быстрый старт”	^

### Оглавление

[История изменений](#)

[Оглавление](#)

[Как объединены интерфейсы и порты?](#)

[ModBus](#)

[Общая информация](#)

[Если нужно внести изменения в параметры работы Modbus](#)

[Настройка modbus через конфигурационный файл ядра “DWINOS\\_\\*.bin”.](#)

[Для режима master/ведущий добавление modbus команд в файл дампа памяти VP\(RAM\) “22\\*.bin”.](#)

[Быстрый старт](#)

[ModBus режим slave/ведомый](#)

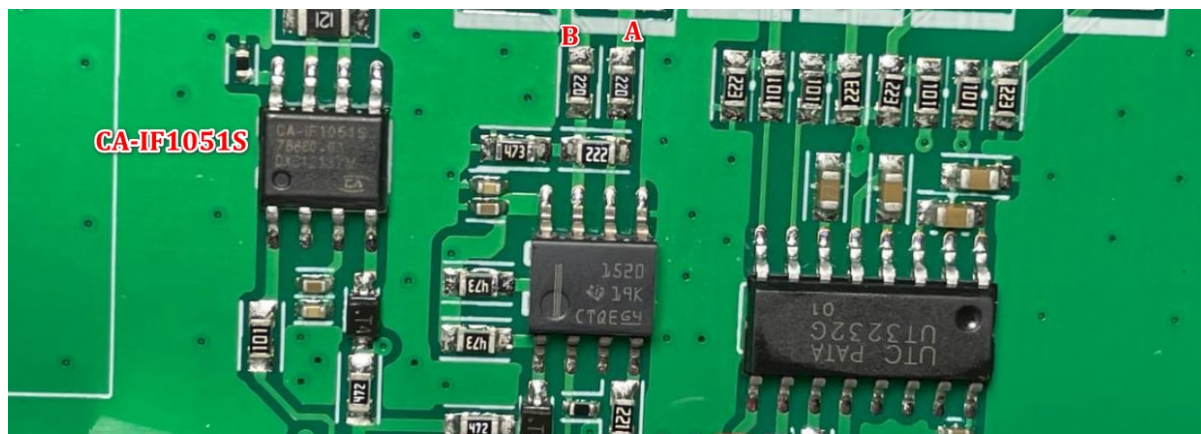
[ModBus режим master/ведущий](#)

[Утилита для копирования modbus инструкций в файл 22\\*.bin](#)

## Как объединены интерфейсы и порты?

Следует понимать что микроконтроллер T5L\* имеет только **шесть** UART выходов и **один** CAN.

Все остальные интерфейсы(RS-232, RS-485 и др) реализуются за счёт установки на плате микросхем преобразователей интерфейсов: UART<->RS-232, UART<->RS-485 и т.д.



Кстати, CAN тоже требует внешней микросхемы приемопередатчика к физической шине.

На каком порту какой интерфейс подключен - можно узнать из **datasheet** на **соответствующую модель дисплея**.

Примерно в разделе *2.3 Serial interface parameters*

串口模式 Mode	UART2: RS232 UART4: RS232(OS 配置后才能使用 Only available after OS configuration) UART5: RS485(OS 配置后才能使用 Only available after OS configuration)
--------------	--

Пример для дисплея промышленного класса в пластиковом корпусе.

**Линейка продукции для настенного монтажа**(“термостаты” и “контроллеры”) часто имеет RS-485 на порту UART5.

### Линейка продукции коммерческого класса

не оснащается преобразователем RS-485, однако часто оснащается RS-232 который подключен к порту UART2. При этом режим работы можно настраивать перемычкой, которая находится возле микросхемы преобразователя RS-232.

То есть преобразователь RS-232 можно:

отключить(если замкнуть перемычку) - на выходе RX2/TX2 будет TTL.

включить(если разомкнуть перемычку) - на выходе RX2/TX2 будет RS-232. <- так с завода

# ModBus

## Общая информация

Структура кадра(пакета) данных в протоколе ModBus:

[Slave ID] [Код функции] [Данные] [CRC]

[Slave ID] - адрес *ведомого* устройства с которым происходит взаимодействие от *ведущего*;

[Код функции] - команда определяющая действия,

команды поддерживаемые протоколом ModBus и дисплеем:

- 1 (0x01) — чтение значений из нескольких регистров флагов (Read Coil Status).
- 2 (0x02) — чтение значений из нескольких дискретных входов (Read Discrete Inputs).
- 3 (0x03) — чтение значений из нескольких регистров хранения (Read Holding Registers).
- 4 (0x04) — чтение значений из нескольких регистров ввода (Read Input Registers).
- 5 (0x05) — запись значения одного флага (Force Single Coil).
- 6 (0x06) — запись значения в один регистр хранения (Preset Single Register).
- 7 (0x07) — чтение сигналов состояния (Read Exception Status).
- 15 (0x0F) — запись значений в несколько регистров флагов (Force Multiple Coils).
- 16 (0x10) — запись значений в несколько регистров хранения (Preset Multiple Registers).

[Данные] - передаваемые данные;

[CRC] - контрольная сумма для проверки целостности доставленного пакета.

Больше информации по [modbus в wiki](https://modbus.wiki)

**В микроконтроллере T5L протокол ModBus** реализован программно и использует порты UART4 или UART5.

А также таймер T1 и *seventh page of the register*(~седьмую страницу регистров).

ModBus может работать только на одном из портов: UART4 или UART5.

❗ Режим работы, порт и скорость modbus можно настроить через загрузку конфигурационного файла ядра “**DWINOS\_\*.bin**”, не “T5L\*.cfg”.

Скорость порта которая указывается в **T5L\*.cfg** влияет только на UART2 и **не влияет на modbus**.

Конфигурационный файл ядра “DWINOS\_\*.bin” для самых популярных режимов modbus можно скачать и использовать не настраивая. [Файлы есть в общем архиве](#).

Имя	Дата изменения	Тип
DWIN_T5L_9600_UART4_MODBUS_Master	07.07.2021 15:59	Папка с файлами
DWIN_T5L_9600_UART4_MODBUS_Slave	07.07.2021 12:45	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Slave	07.07.2021 12:48	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Slave	07.07.2021 12:51	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Slave	30.09.2022 14:33	Папка с файлами

ℹ Файл \*.bin загружается в дисплей также как и остальные - через флеш карту и папку DWIN\_SET или через загрузчик по UART2.

## Внесение изменений в параметры работы Modbus

Через конфигурационный файл ядра "DWINOS\_\*.bin".

Данный файл можно отредактировать через HEX редактор(например мощный и [свободный HxD](#)).  
 Ниже приведены основные байты файла.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
00001060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00001070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00001080	26	04	23	00	26	04	2D	00	27	00	04	21	03	0A	5A	A5	&.#.&.-.'...ZΓ
00001090	03	0C	E0	00	03	0E	FF	04	03	10	05	FF	03	12	01	50	...а...я...я...Р
000010A0	03	14	00	E0	03	00	00	88	01	0A	00	06	00	00	00	00	...а...€.....
000010B0	25	00	00	00	02	0A	07	01	06	0A	00	01	03	00	00	88	%.....€
000010C0	01	02	01	01	23	02	52	03	02	02	00	01	01	02	00	01	± 7

На примере для режима slave/ведомого.

1. **E0 00** - 0xE000 - Начальный адрес(слово - т.е. должно быть кратно 2) области памяти VP с которого начинают располагаться modbus инструкции.

2. **FF** - Максимальное количество инструкций в файле 22\*.bin, 0xFF = 255(DEC).

**04** - Режим slave/ведомый, RTU формат.

Возможные варианты:

0x00 = Сконфигурирован как мастер, RTU формат.

0x01 = Сконфигурирован как мастер, ASCII формат (в настоящее время не поддерживается).

0x02 = Сконфигурирован как мастер, RTU формат, ведомый отвечает ?часами? в данных.

0x03 = Сконфигурирован как мастер, ASCII формат, ведомый отвечает ?часами? в данных (в настоящее время не поддерживается).

0x04 = Сконфигурирован как ведомый, RTU формат.

0x05 = Сконфигурирован как ведомый, ASCII формат (в настоящее время не поддерживается).

3. **05** - Количество повторений отправки команды.

**FF** - время задержки между командами 0xFF ~ 255 мс.

4. **01** - идентификатор дисплея в режиме ведомого. ID Slave: от 0x01 до 0x7F (1 - 127).

**40** - Параметры COM порта.

для UART4: 0x40 = 8N1 (по умолчанию), 0x41 = 8E1, 0x42 = 8O1, 0x43 = 8N2.

для UART5: 0x50 = 8N1 (по умолчанию), 0x51 = 8E1, 0x52 = 8O1, 0x53 = 8N2.

5. **00 E0** - Делитель скорости порта (UART4, UART5) для Modbus

0x00E0 = 224(DEC), 25804800 / 224 = 115 200 Baud.



Добавление modbus инструкций в VP(RAM) **“22\*.bin”** для режима *master/ведущий*.

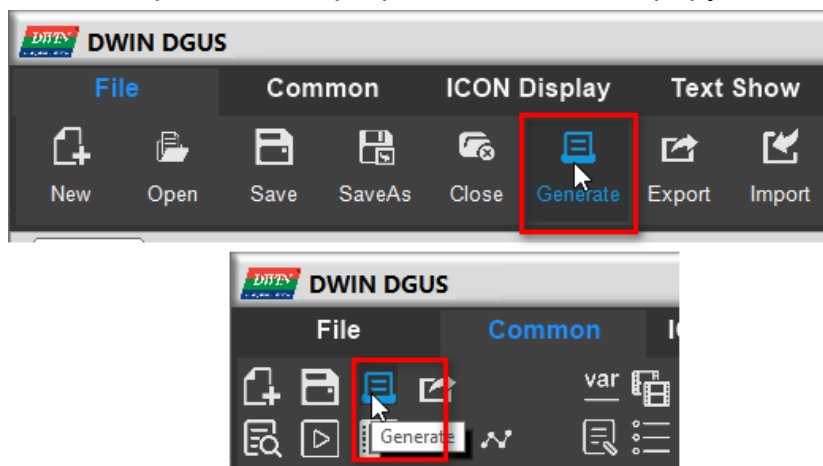
Актуально только для режима master/ведущий.

Вспомним что в область памяти VP(RAM) информация попадает:

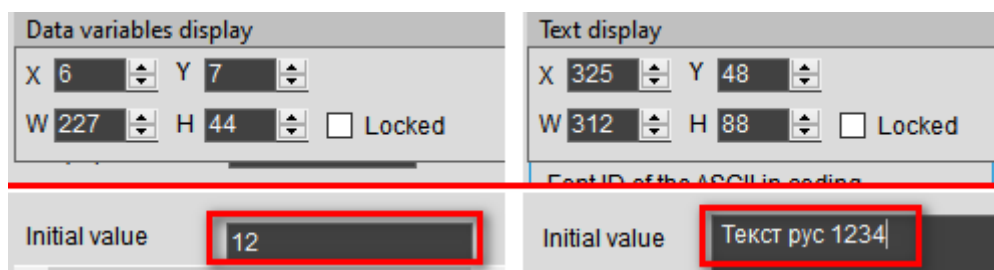
- либо в процессе работы дисплея(за счёт ввода данных и командами через UART порты),
- либо из файла дампа памяти **22\*.bin**, который содержит массив данных(значений переменных), который выгружается в область памяти VP(RAM) при включении/перезагрузки дисплейного модуля.

Подробнее:

При генерации файлов дизайн-прошивки в программе DGUS генерируется также файл 22.bin,



файл 22.bin содержит **начальные значения переменных**, которые задаются в различных инструментах дизайна:



Инструкции добавляются начиная с адреса VP=0xE000, в файле 22\*.bin это адрес 0x1C000.

Открывать файл 22\*.bin нужно в HEX редакторе([например HxD](#)), затем вносим/копируем нужные нам инструкции начиная с адреса 0x1C000.

Добавлять инструкции в файл 22\*.bin нужно после каждой генерации. Так как генерация удаляет старый файл.

Также добавить команды можно динамически через UART(в область VP) или используя переменные VP с адреса 0xE000.

Структура инструкции modbus master: (пример инструкции для файла 22\*.bin)

5A 01 03 06 02 01 00 00 00 00 14 00 00 00 00

D0: 5A - стартовый байт команды;

D1: 01 - Идентификационный номер ведомого устройства;

D2: 03 - команда (может быть 1,2,3,4,5,6,10);

D3: 06 - длина данных(количество слов);

D4: 02 - время ожидания приема(2 ~ 255);

D5: 01 - Режим отправки выполнения команды триггера (0-4)

0x00 = Безусловное выполнение.

0x01 = Выполнить на странице, указанной в D6.D7.

0x02 = Выполнить, когда значение переменной VP указанной в D6.D7, не равно нулю, значение переменной будет очищено после выполнения инструкции.

0x03 = Для команды 0x06, когда значение переменной, на которую указывает D8.D9, не равно нулю, оно будет отправлено автоматически.

После завершения связи очистите значение переменной, указанной в D8.D9.

0x04 = Для команд 0x05, 0x06; 0x10 он будет отправлен автоматически при изменении указанной переменной D8.D9.

D6, D7: 00 00

при D5 = 0, настройка не требуется.

при D5 = 1, D6 D7 содержит номер страницы (0x0002).

при D5 = 2, D6 D7 ключевой адрес, запускающий отправку.

при D5 = 3, настройка не требуется.

при D5 = 4, настройка не требуется.

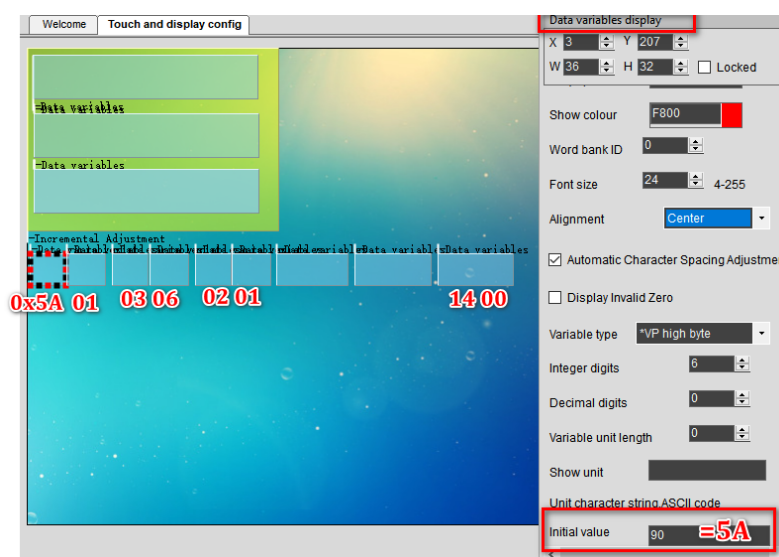
D8, D9: 00 00 - Эта инструкция используется для управления адресом исходной переменной или начальным адресом (0x1000) данных, размещенных на экране DGUS.

D10, D11: 14 00 - таблица адресов ведомого устройства (0x0001), связанная с данной инструкцией.

D12, D13, D14, D15: Не используются.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
0001BFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C000	5A	01	03	06	02	01	00	00	00	00	14	00	00	00	00	00	Z.....
0001C010	5A	01	03	01	30	01	00	00	50	04	00	04	00	00	00	00	Z...0...P.....
0001C020	5A	01	03	01	30	01	00	00	50	02	00	02	00	00	00	00	Z...0...P.....
0001C030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Пример прямой записи в 22\*.bin



Пример как можно добавить запись в 22\*.bin через интерфейс.

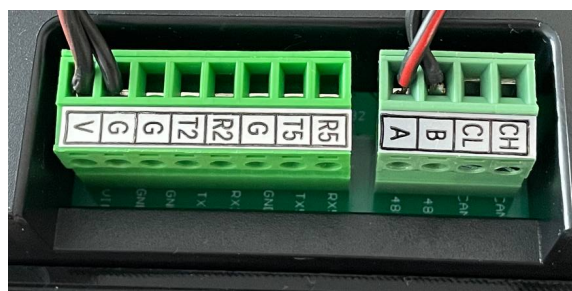
## Быстрый старт

С одной стороны в качестве master/ведущего и slave/ведомого будет эмулятор для ПК: [ПО Modbus Poll\(master\) и Modbus Slave](#).

С другой стороны будет [дисплейный HMI модуль DMG10600T070\\_A5WTC](#),  
Посмотрим на надписи на шелкографии дисплея: RS-485 находится на UART4



Больше информации есть в datasheet: на внешних Phoenix разъёмах выведены 4 порта:

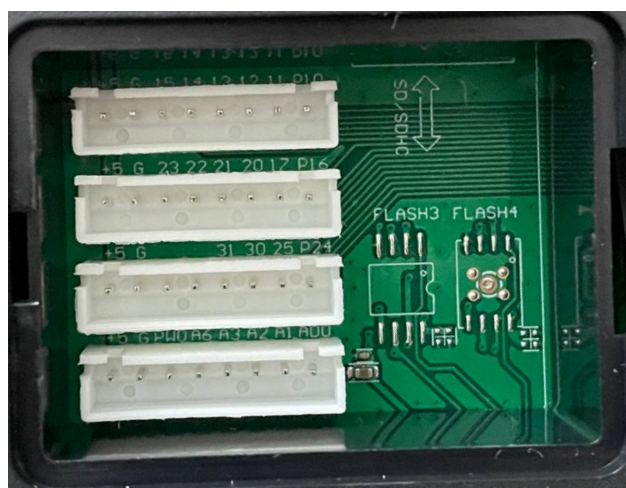


**UART2:** RS232 (на шелкографии платы подписаны как RX2/TX2, на разъёме R2/T2)

**UART4:** RS485(на шелкографии платы подписаны как 485, на разъёме A/B)

**UART5:** RS232 (на шелкографии платы подписаны как RX5/TX5, на разъёме R5/T5)  
CAN\*1

на внутренних PH2.0 разъёмах выведены ещё 16 GPIO и 4 порта TTL:  
UART3, UART5, UART6, UART7.



**i** Modbus не обязательно использовать на RS-485, он может работать на RS-232, UART-TTL и др.  
В зависимости от того как подключен UART4 и UART5.

Дисплей от внешнего разъёма RS-485/UART4  
подключен к USB ПК через [преобразователь](#):



## ModBus режим slave/ведомый

Modbus будет работать на скорости 115200, адрес дисплея slave/ведомого = 1.

Выбираем соответствующий файл:

Имя	Дата изменения	Тип
DWIN_T5L_9600_UART4_MODBUS_Master	07.07.2021 15:59	Папка с файлами
DWIN_T5L_9600_UART4_MODBUS_Slave	07.07.2021 12:45	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Slave	07.07.2021 12:48	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Slave	07.07.2021 12:51	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Slave	30.09.2022 14:33	Папка с файлами

Имя	Дата изменения	Тип
DWIN_T5L_115200__UART4_ModBus.asm	07.07.2021 12:51	Файл "ASM"
DWINOS_DWIN_T5L_115200__UART4_ModBus.bin	07.07.2021 12:51	Файл "BIN"
VAR_DWIN_T5L_115200__UART4_ModBus.Lst	07.07.2021 12:51	Файл "LST"

Графический интерфейс возьмём из проекта */GUI\_DGUS\_slave/DWIN\_SET/*

В папку DWIN\_SET скопирован файл конфигурации ядра  
*DWINOS\_DWIN\_T5L\_115200\_\_UART4\_ModBus.bin*  
 режим ModBus RTU, 115200 бод, slave ID = 1

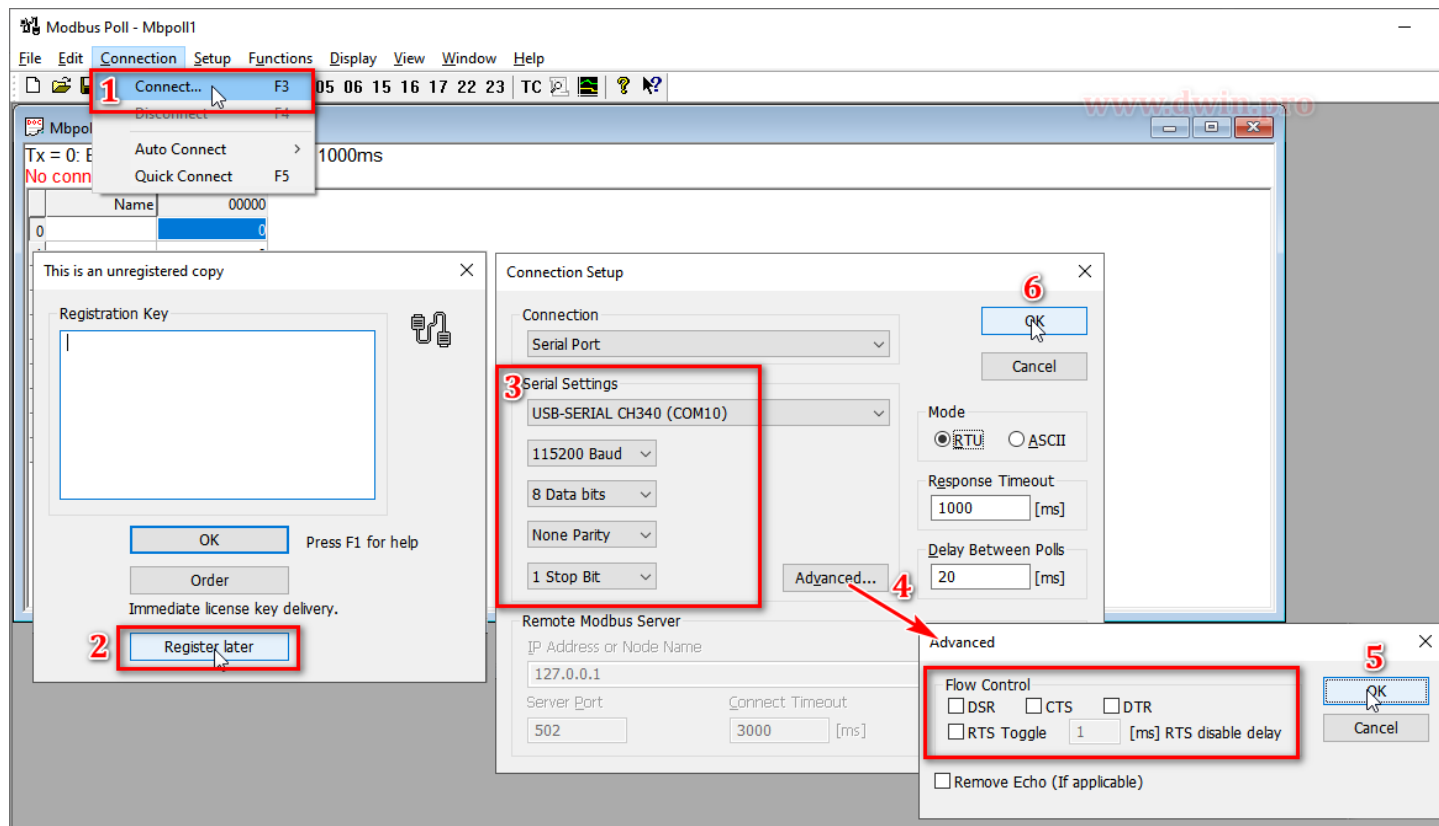
Если дисплей имеет RS-485 на порту UART5, а не UART4, то скопируйте файл из соответствующей папки

Имя	Дата изменения	Тип
DWIN_T5L_9600_UART4_MODBUS_Master	07.07.2021 15:59	Папка с файлами
DWIN_T5L_9600_UART4_MODBUS_Slave	07.07.2021 12:45	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Slave	07.07.2021 12:48	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Slave	07.07.2021 12:51	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Slave	30.09.2022 14:33	Папка с файлами

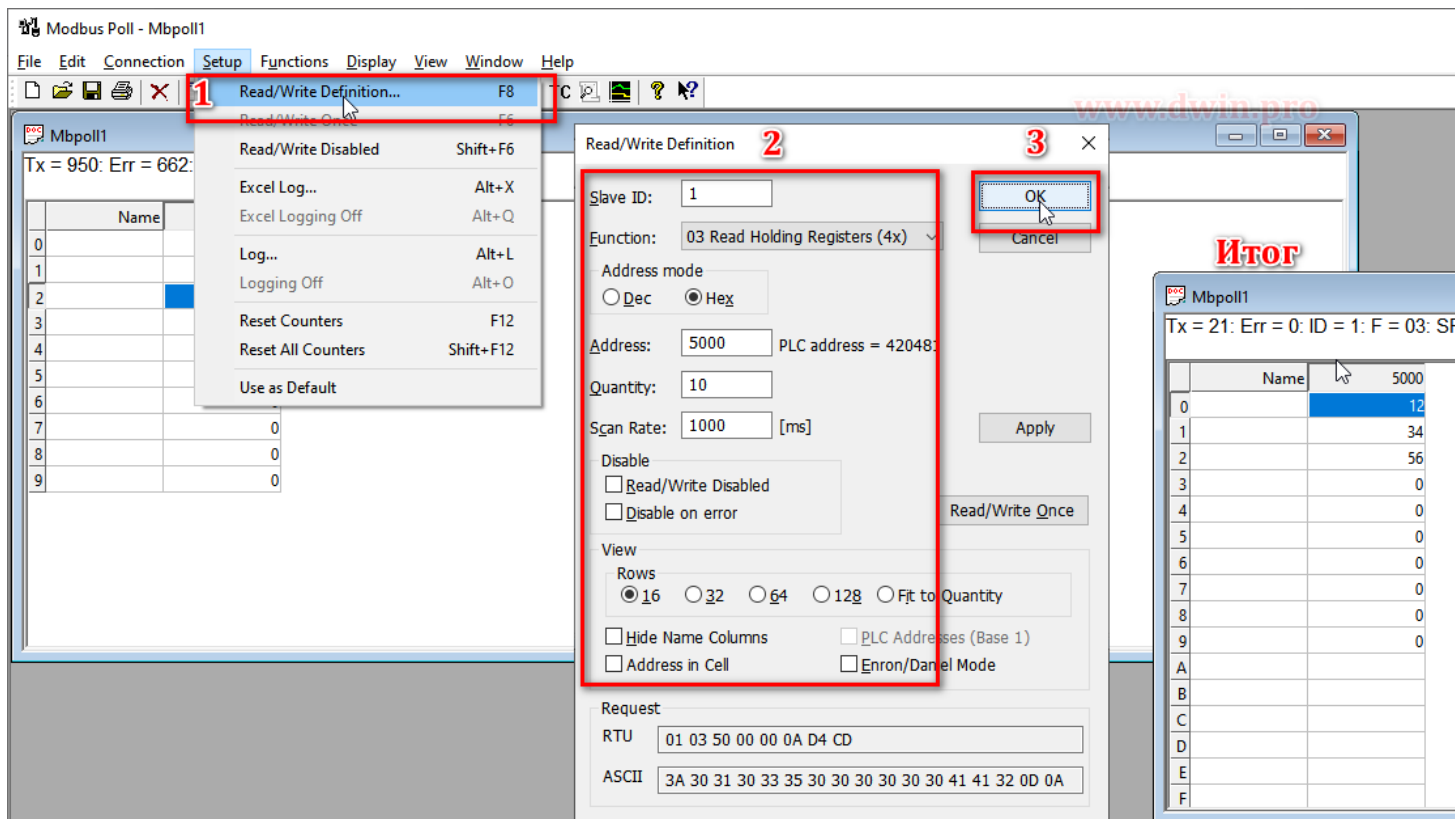
файл *DWINOS\_DWIN\_T5L\_115200\_\_UART5\_ModBus.bin*



## Modbus Poll: Настройки подключения

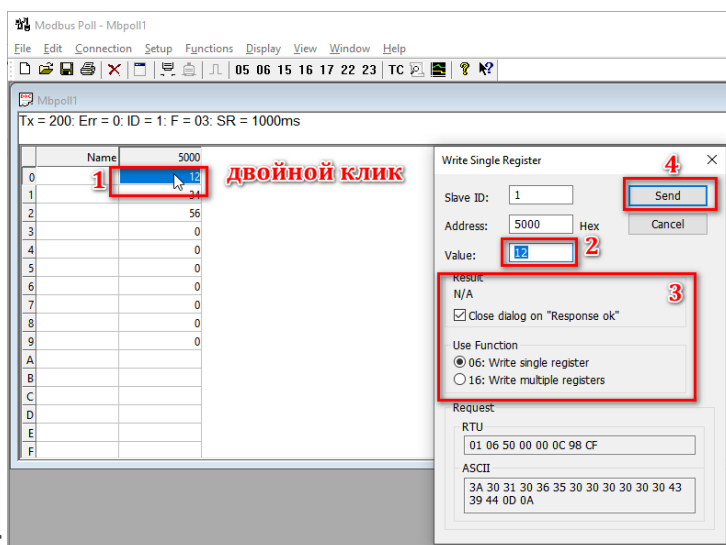


## Modbus Poll: определение команд чтения/записи

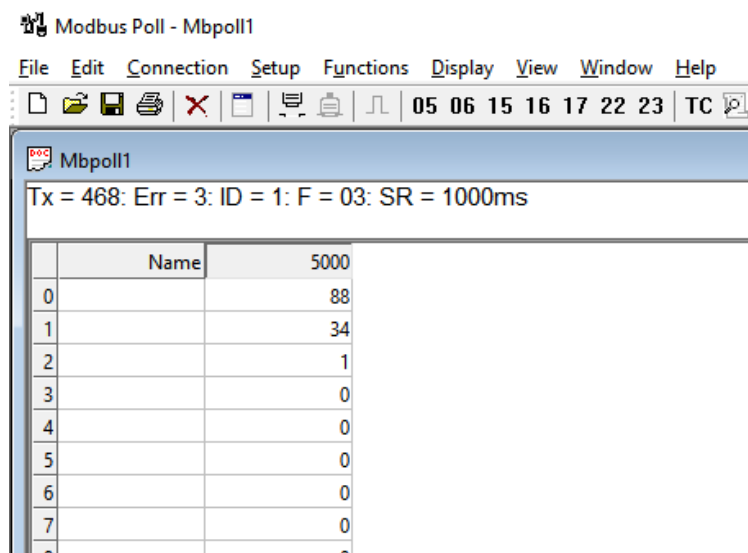


## Modbus Poll: меняем данные

До:



После:





## ModBus режим master/ведущий

Modbus будет работать на скорости 115200.

Выбираем соответствующий файл:

Имя	Дата изменения	Тип
DWIN_T5L_9600_UART4_MODBUS_Master	07.07.2021 15:59	Папка с файлами
DWIN_T5L_9600_UART4_MODBUS_Slave	07.07.2021 12:45	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Slave	07.07.2021 12:48	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Slave	07.07.2021 12:51	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Slave	30.09.2022 14:33	Папка с файлами

Имя	Дата изменения	Тип
DWIN_T5L_115200__UART4_ModBus.asm	07.07.2021 12:51	Файл "ASM"
DWINOS_DWIN_T5L_115200__UART4_ModBus.bin	07.07.2021 12:51	Файл "BIN"
VAR_DWIN_T5L_115200__UART4_ModBus.lst	07.07.2021 12:51	Файл "LST"

Графический интерфейс возьмём из проекта */GUI\_DGUS\_master/DWIN\_SET/*

В папку DWIN\_SET скопирован файл конфигурации ядра  
*DWINOS\_DWIN\_T5L\_115200\_\_UART4\_ModBus.bin*  
режим ModBus RTU, 115200 бод.

Если дисплей имеет RS-485 на порту UART5, а не UART4, то скопируйте файл из соответствующей папки

Имя	Дата изменения	Тип
DWIN_T5L_9600_UART4_MODBUS_Master	07.07.2021 15:59	Папка с файлами
DWIN_T5L_9600_UART4_MODBUS_Slave	07.07.2021 12:45	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_9600_UART5_MODBUS_Slave	07.07.2021 12:48	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART4_MODBUS_Slave	07.07.2021 12:51	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Master	07.07.2021 16:00	Папка с файлами
DWIN_T5L_115200_UART5_MODBUS_Slave	30.09.2022 14:33	Папка с файлами

*файл DWINOS\_DWIN\_T5L\_115200\_\_UART5\_ModBus.bin*

В качестве ведомого устройства будет использоваться эмулятор Modbus Slave.

Запишем 4 инструкции,

2 инструкции на считывание данных с ведомого устройства и их запись в ячейки с VP адресами 5000 и 5001.

И 2 инструкции на запись данных в ведомое устройство из VP ячеек 5002 и 5003.

## Инструкции modbus master:

1. 5A 01 03 01 30 01 00 00 50 00 00 02 - чтение ячейки 0002 из Slave устройства(01) и запись в дисплей в ячейку 5000.
2. 5A 01 03 01 30 01 00 00 50 01 00 04 - чтение ячейки 0004 из Slave устройства(01) и запись в дисплей в ячейку 5001.
3. 5A 01 06 01 30 04 00 00 50 02 00 01 - автоматическая отправка данных в ведомое устройство при изменении переменной в ячейки 0x5002, отправка происходит в ячейку памяти 0001 Slave устройства с адресом 0x01.

Данные команды ^^ запомним через **визуальный редактор DGUS**.

4. 5A 01 06 01 30 04 00 00 50 03 00 00 - автоматическая отправка данных в ведомое устройство при изменении переменной в ячейке 0x5003, отправка происходит в ячейку памяти 0000 Slave устройства с адресом 0x01.

А эту команду запишем через **hex редактор HxD**.

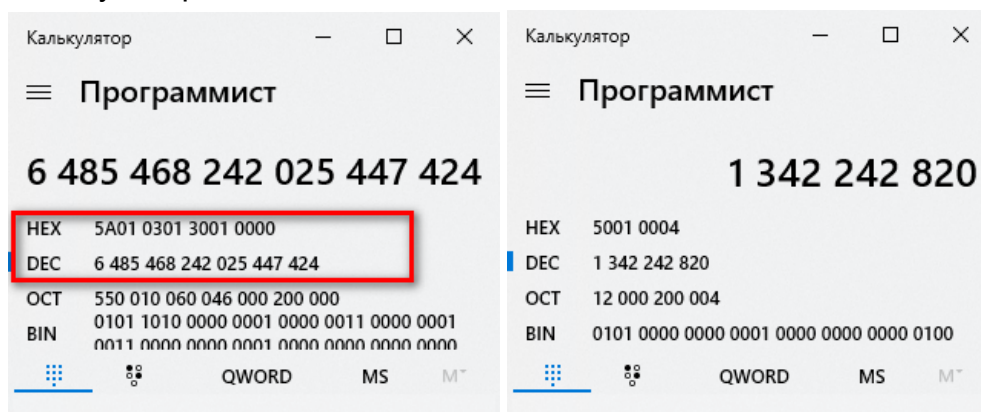
**Пример** на второй инструкции: 5A 01 03 01 30 01 00 00 50 01 00 04.

Инструкция состоит из 12 байт. Её можно разделить на

*Extra Long Int*(8 байт): 5A 01 03 01 30 01 00 00 и

*Long Int*(4 байта): 50 01 00 04.

Так как в переменные DGUS Data variables можно записать только десятичные числа, то переведём HEX в DEC через калькулятор методом ctrl+c ctrl+v.



В итоге:

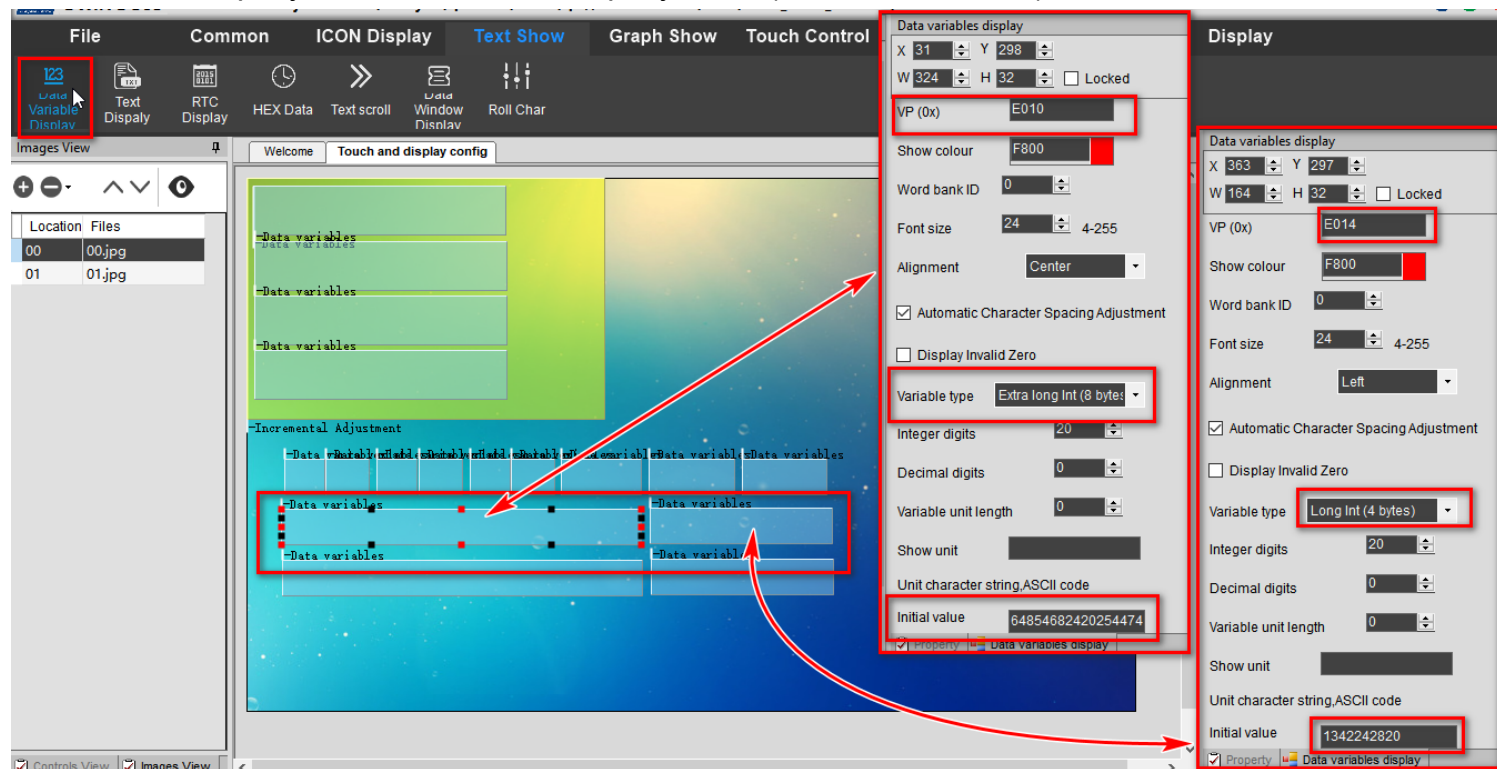
*Extra Long Int*(8 байт): 5A 01 03 01 30 01 00 00 = 6 485 468 242 025 447 424

*Long Int*(4 байта): 50 01 00 04 = 1 342 242 820

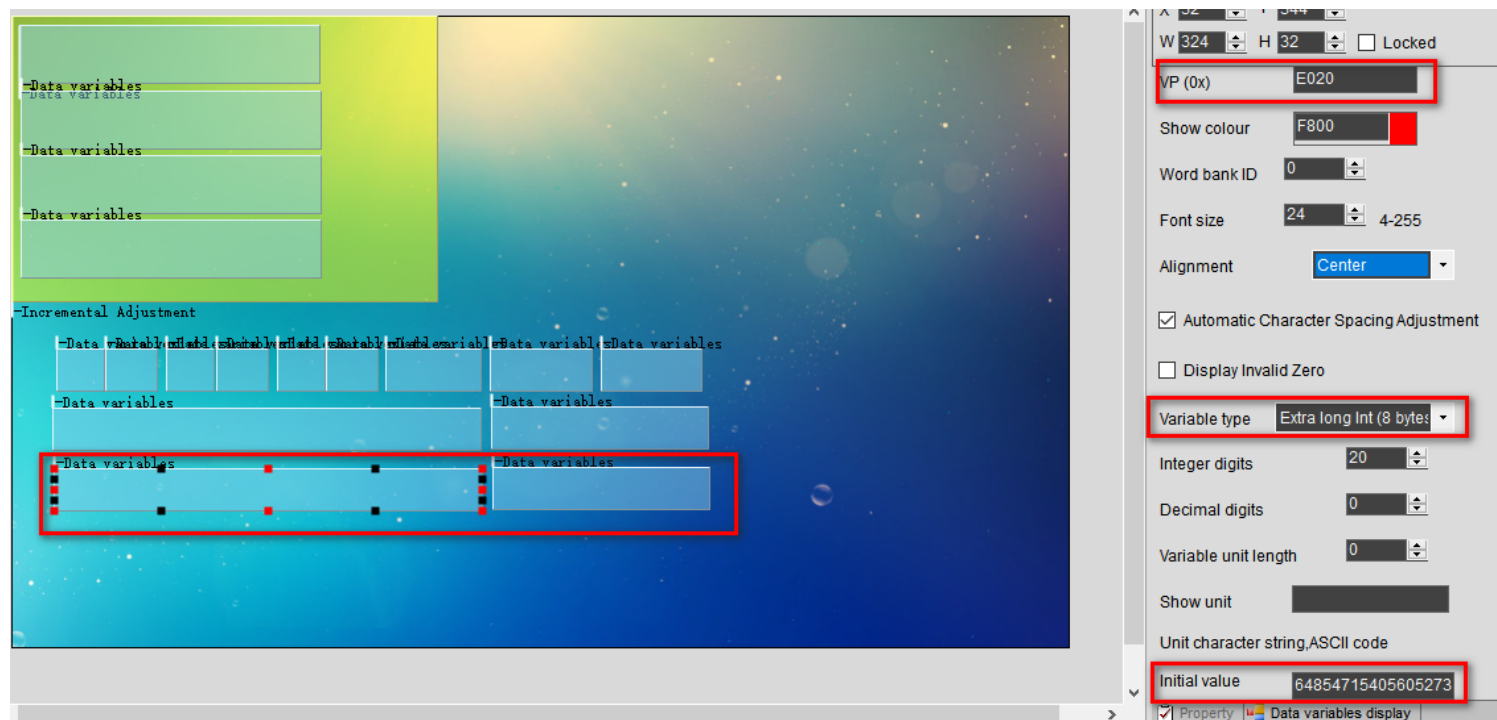
 при копировании в DGUS нужно удалить пробелы.

Добавим Data variables в DGUS.

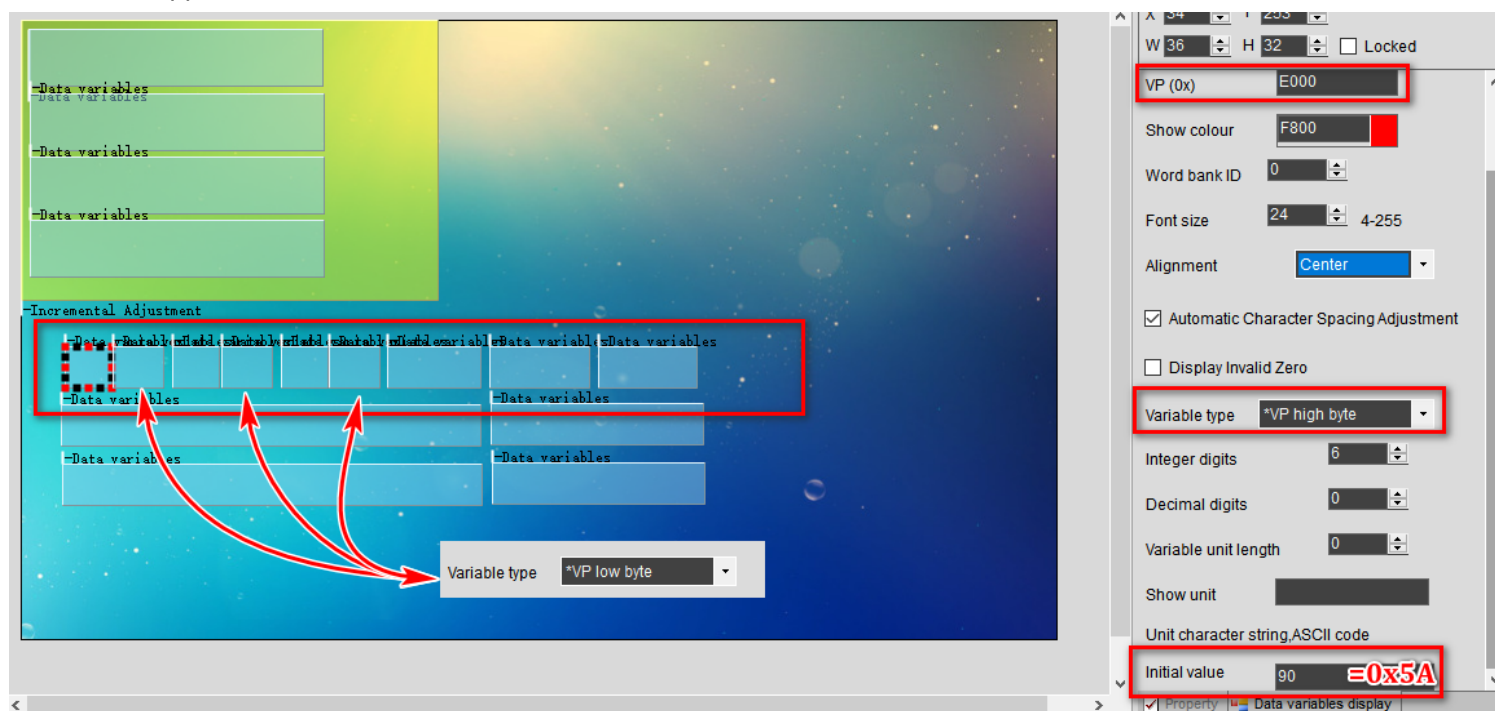
8 байт по VP адресу E010, 4 байта по адресу E014 (смещение 8 байт)



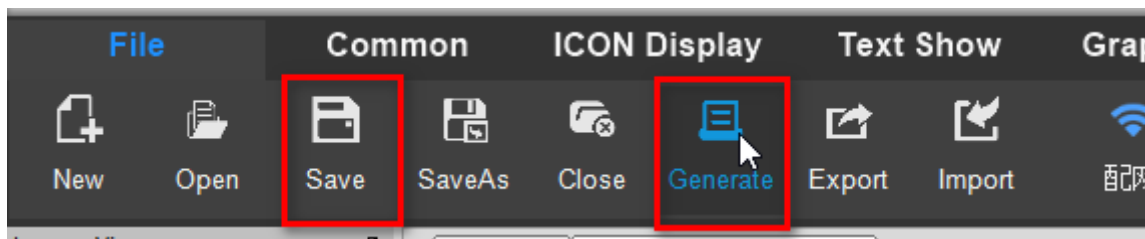
Повторим тоже самое для 3ей инструкции помня про смещение VP



Для первой инструкции инструменты *Data variables* разбивают инструкцию по байтам чтобы можно было детальнее понять как это работает. Главное помнить что в Data variables мы указываем значения в десятичной системе счисления.



После завершения генерируем файлы.



Заходим в `/GUI_DGUS_master/DWIN_SET/`  
И видим обновлённые файлы

Имя	Дата изменения	Тип
00.jpg	30.09.2022 12:37	Файл "JPG"
01.jpg	30.09.2022 12:37	Файл "JPG"
13TouchFile.bin	05.10.2022 23:27	Файл "BIN"
14ShowFile.bin	05.10.2022 23:27	Файл "BIN"
22_Config.bin	05.10.2022 23:27	Файл "BIN"
32.icl	30.09.2022 13:01	Библиотека знач.
T5LCFG.CFG	02.07.2022 0:34	Файл "CFG"

Открываем/перетаскиваем 22\*.bin в hex редактор(например HxD) и сразу переходим по адресу 0x1C000.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
0001BFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C000	5A	01	03	01	30	01	00	00	50	00	00	02	00	00	00	00	Z...0...P.....
0001C010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C020	5A	01	03	01	30	01	00	00	50	01	00	04	00	00	00	00	Z...0...P.....
0001C030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C040	5A	01	06	01	30	04	00	00	50	02	00	01	00	00	00	00	Z...0...P.....
0001C050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Видим 3 инструкции добавленные через визуальный редактор.

Добавим 4-ую инструкцию просто скопировав её 5A 01 06 01 30 04 00 00 50 03 00 00  
 Устанавливаем курсор в начало следующей строки и вставляем инструкцию,

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
0001BFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C000	5A	01	03	01	30	01	00	00	50	00	00	02	00	00	00	00	Z...0...P.....
0001C010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C020	5A	01	03	01	30	01	00	00	50	01	00	04	00	00	00	00	Z...0...P.....
0001C030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C040	5A	01	06	01	30	04	00	00	50	02	00	01	00	00	00	00	Z...0...P.....
0001C050	5A	01	06	01	30	04	00	00	50	03	00	00	00	00	00	00	Z...0...P...0...
0001C060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

затем нажимаем на дискету(ctrl+s) для сохранения данных, должно получиться так:

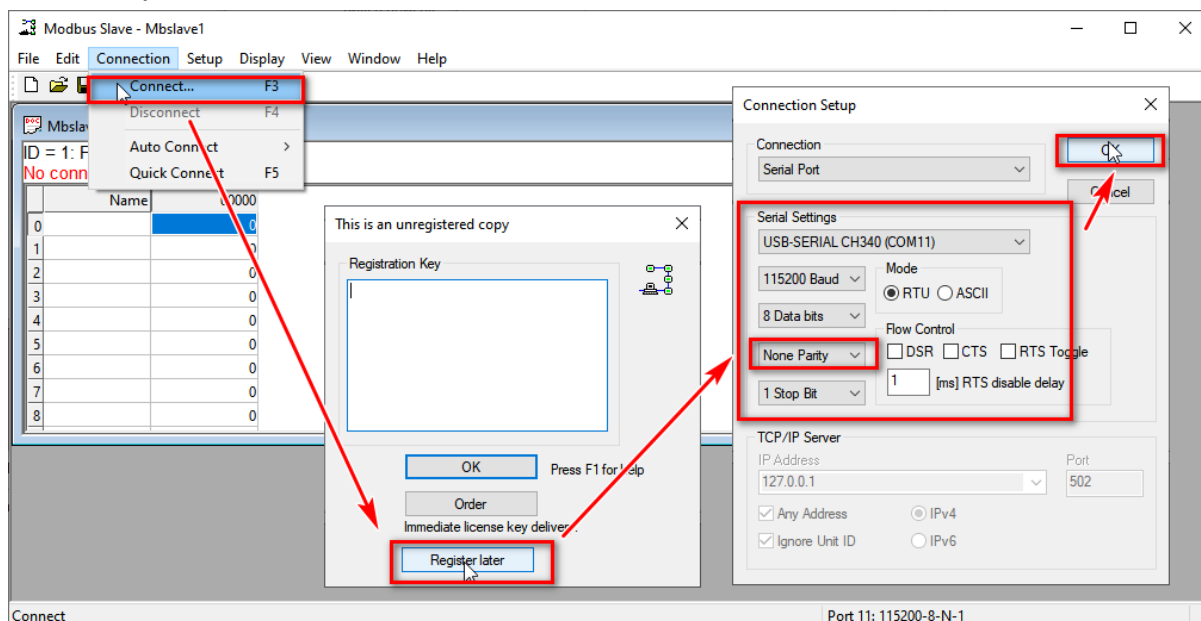
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
0001BFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C000	5A	01	03	01	30	01	00	00	50	00	00	02	00	00	00	00	Z...0...P.....
0001C010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C020	5A	01	03	01	30	01	00	00	50	01	00	04	00	00	00	00	Z...0...P.....
0001C030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C040	5A	01	06	01	30	04	00	00	50	02	00	01	00	00	00	00	Z...0...P.....
0001C050	5A	01	06	01	30	04	00	00	50	03	00	00	00	00	00	00	Z...0...P...0...
0001C060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Следует помнить что при повторной DGUS генерации копирование инструкций нужно повторить.  
 Выше приведены два примера. Можете использовать любой из них.

Итоговые файлы загружаем в дисплей. (файл .bak можно удалить)

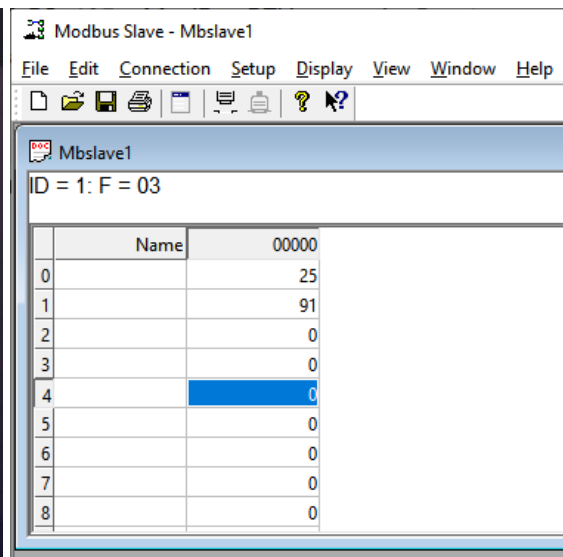
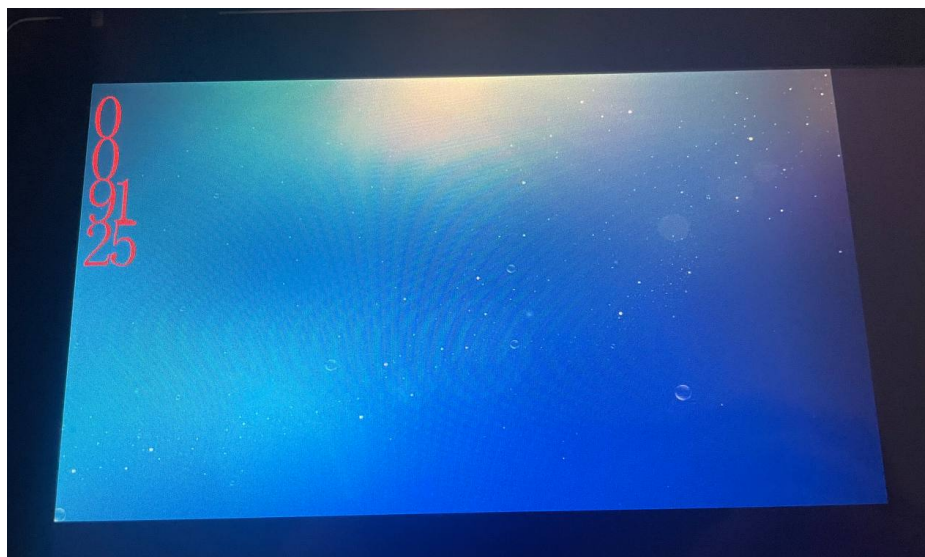
Имя	Дата изменения	Тип
00.jpg	30.09.2022 12:37	Файл "JPG"
01.jpg	30.09.2022 12:37	Файл "JPG"
13TouchFile.bin	05.10.2022 23:27	Файл "BIN"
14ShowFile.bin	05.10.2022 23:27	Файл "BIN"
22_Config.bin	05.10.2022 23:31	Файл "BIN"
22_Config.bin.bak	05.10.2022 23:27	Файл "BAK"
32.icl	30.09.2022 13:01	Библиотека знач...
T5LCFG.CFG	02.07.2022 0:34	Файл "CFG"

## Modbus Slave: настройка подключения

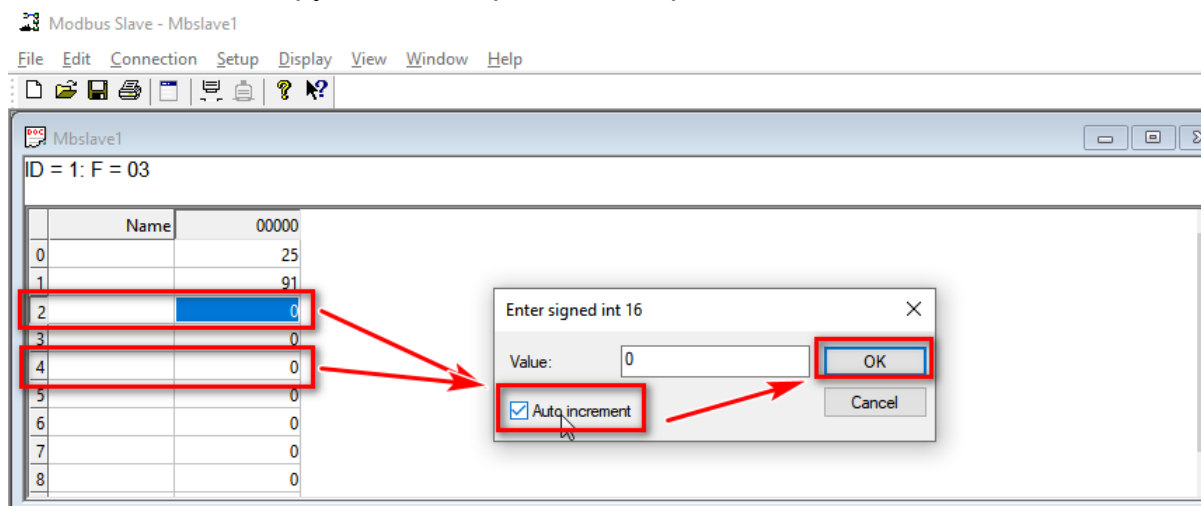


## Проверка:

Нажимаем на 3 и 4 числа для их изменения, наблюдаем что в *Modbus Slave* произошли изменения.

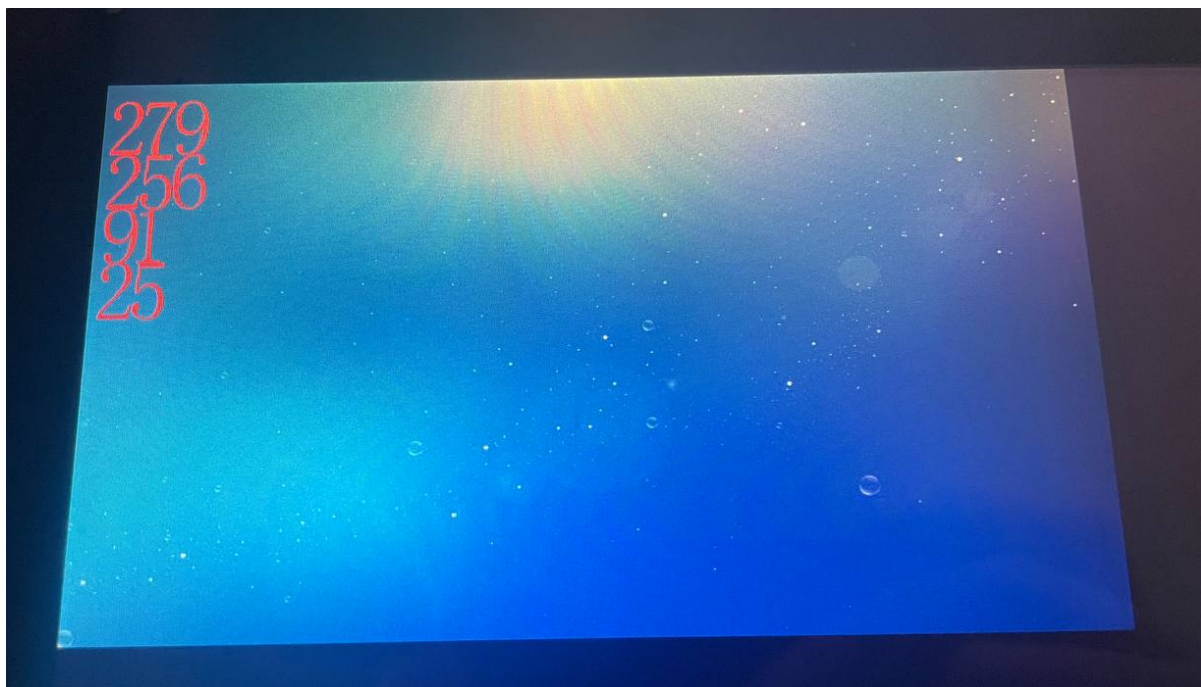


Далее в *Modbus Slave* активируем автоинкремент в переменных 0002 и 0004





Проверяем результат на дисплее:



### Утилита для копирования modbus инструкций в файл 22\*.bin

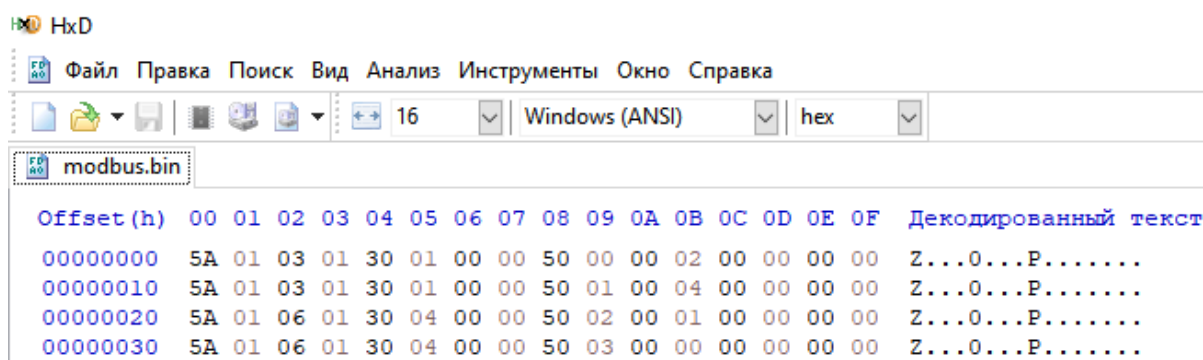
Открытие файла 22\*.bin в hex редакторе после каждой генерации - дело утомительное.

В этом может помочь готовый скрипт **“копировать в 22.vbs”** который находится в общем архиве.

Его нужно скопировать в папку *DWIN\_SET*

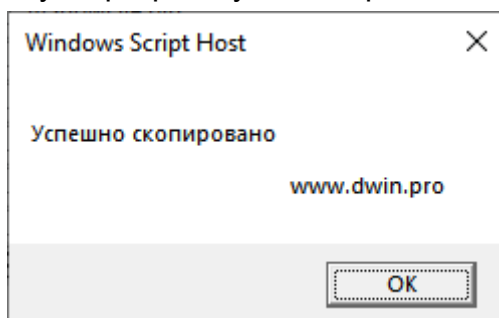
Скрипт копирует содержимое *modbus.bin* в файл *22\_Config.bin* с адреса 0x1C000

Для использования скрипта нужно создать в папке *DWIN\_SET* файл *modbus.bin* и скопировать в него нужные проекту modbus инструкции(через hex редактор).



Затем нажмите кнопку генерации в DGUS, DGUS сгенерирует файл *22\_Config.bin*

Следом запустите скрипт как обычную программу. После работы скрипта он сообщит о завершении



Проверим его работу.

**До запуска:**

22\_Config.bin

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст
0001BFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0001C070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

**После запуска:**

22_Config.bin																		
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Декодированный текст	
0001BFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	
0001C000	5A	01	03	01	30	01	00	00	50	00	00	02	00	00	00	00	Z...0...P.....	
0001C010	5A	01	03	01	30	01	00	00	50	01	00	04	00	00	00	00	Z...0...P.....	
0001C020	5A	01	06	01	30	04	00	00	50	02	00	01	00	00	00	00	Z...0...P.....	
0001C030	5A	01	06	01	30	04	00	00	50	03	00	00	00	00	00	00	Z...0...P.....	
0001C040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	



Теги для поиска:

HMI панель,  
modbus,  
ПЛК,  
PLC,  
конфигурация портов dwin,  
rs485,  
dwin порты,  
DGUS,  
dwin rs485,  
HMI дисплей,  
modbus dwin,  
панель оператора,  
uart tft,  
arduino,

#diy #сенсорная панель #modbus #ПЛК #PLC #dwin #dgus