

Описание примера работы контроллеров серий Delta AS/AH/ES3 с смарт камерой на примере чтения DMC кодов «AS228T_DMC_reading»

В данном примере по сигналу с датчика смарт камера считывает группу Data Matrix кодов (DMC), а контроллер типа AS228T-A их принимает при помощи TCP сокет в режиме клиента. Далее контроллером проверяется факт прихода кодов от камеры, количество принятых кодов путём подсчёта разделителей, а также можно оценить правильность GTIN. В случае правильных кодов они отправляются через другой TCP сокет контроллера в режиме клиента на ПК. Для работы примера в контроллер необходимо загрузить сам проект из ISPSoft, а также аппаратные настройки через конфигуратор ISPSoft -> HWCONFIG.

В качестве источника кодов используются утверждённые российскими стандартами Data Matrix коды следующего вида:

4	стр.30, раздел V,п.45		DataMatrix	01	GTIN	21	Эриин.ном	ASII29	93	Ключ проверки	Итого	
5	841 от 31.05.2021	Postanovlenie Pravitelstva Rossijskoj	Вода	Длина	2	14	2	13	1	2	4	38



Данные коды считываются в виде ASCII кодов следующего вида:

```
0104610053147452215s:sKbUFh<fdM 93dGVz;  
0104610053147452215.PZ3cR,R2WQY 93dGVz;  
0104610053147452215bTlAXHLqNlhg 93dGVz;  
0104610053147452215BCPINp/e8Hi293dGVz;  
0104610053147452215GBTZaHslinQ:93dGVz
```

(на месте пропуска имеется ввиду символ ASCII29, см. структуру выше).

Помимо кодов камера выдаёт стартовый и стоповый символы, а также добавляются разделители кодов. В данном примере символы хранятся в следующих локальных переменных (POU DataProcessing):

```
START_SYMBOL:= 16#000A; // Стартовый символ пакета с кодами (символ LF)  
SEPARATOR := 16#003B; // разделитель кодов - точка с запятой  
END_SYMBOL:= 16#000D; // Символ конца пакета с кодами (символ CR)  
CODE_LENGTH:=38; // Стандартное количество символов в коде
```

Соответственно камера также настроена на данные символы (начало, разделитель, конец). Обработка кодов строится по следующей последовательности действий:

1. При подаче питания или команды RUN контроллер открывает свои TCP сокет 1 и 2 оба в режиме клиента и пробует установить соединение с камерой и ПК. Команда на открытие сокетов находится в **POU CONTROL**, а процедуры в **POU TCP_SOCKET** и **TCP_SOCKET2**. Введена небольшая задержка на ожидание инициализации сокетов камеры и ПК. В случае удачного соединения включаются флаги **CONNECTED_IN_CLIENT_MODE** и **CONNECTED_IN_CLIENT_MODE2**, которые находятся в таблице глобальных переменных.
2. При появлении предмета с DMC кодами срабатывает оптический датчик, который подаёт сигнал на вход контроллера **X0.0** (глобальная переменная **SENSOR**).
3. Контроллер в **POU STROB** по прерыванию выхватывает передний фронт сигнала от датчика и включает свой выход **Y0.0** (глобальная переменная **T_CAMERA**), который подключен к дискретному входу смарт камеры
4. По физическому сигналу от контроллера камера осуществляет чтение кодов и отправляет в режиме сервера TCP пакет контроллеру
5. Контроллер после срабатывания датчика выжидает заданное количество времени (глобальная переменная **DELAY_FOR_CODE_READING**). Если за это время от камеры не пришёл TCP пакет данных, то считается, что камера вообще не отреагировала на триггер и не выдала пакет. В этом случае увеличивается на 1 счётчик **NO_CODES_RECEIVED** и включается на заданное время (таймер **T3**) выход контроллера **Y0.5** . Если TCP пакет от камеры пришёл, т.е. входной буфер (глобальная переменная массив **TCP_1_RECEIVE_BUFFER**) не пуст, то запускается процедура оценки содержимого пакета данных от смарт камеры
6. Процедура оценки пакета данных находится в **POU DataProcessing**. Сначала запускается процедура разбиения принятого массива байтов в младшие байты слов регистров контроллера с перестановкой старшего и младшего байтов с целью восстановления исходного порядка символов DMC кода. Данная процедура нужна ввиду того, что в контроллере нет типа данных **BYTE**, а только слово **WORD**, т.е. два байта. Следовательно, чтобы получить последовательность байтов, удобную для дальнейшей обработки, приходится каждый символ кода поместить в младший байт слова. Для данной процедуры используется локальная переменная **ARRAY_BYTES**, которая является массивом структур типа **BYTES**
7. Далее идёт подсчёт разделителей кодов, по числу которых можно сделать вывод о количестве прочитанных камерой DMC кодов. Заданное количество кодов устанавливается в глобальной переменной **DMC_CODES_AMOUNT_SV**
8. По результатам подсчёта выставляется либо глобальный флаг **GOOD_FLAG** (принятое число кодов соответствует заданному) либо **BAD_FLAG** (не соответствует)
9. Если выставлен глобальный флаг **GTIN_CHECK** , то осуществляется проверка правильность **GTIN** (это та часть кода, которая отвечает за тип товара, см. структуру кода выше) по первому коду. Эталонный **GTIN** задаётся в локальной переменной **GTIN_ETALON** , которая является массивом структур типа **BYTES**
10. По итогам проверки выставляется либо глобальный флаг **GTIN_GOOD** (соответствует) либо **GTIN_BAD** (не соответствует)
11. По итогам оценки кода, в **POU CONTROL** осуществляется увеличение счётчиков и выставление физических выходов контроллера:
 - a. **GOOD_CODES_COUNTER**
 - b. **BAD_CODES_COUNTER**
 - c. **NO_CODES_RECEIVED**

- d. **Y0.2** (получены все коды)
 - e. **Y0.7** (коды получены, но не все)
 - f. **Y0.5** (вообще от камеры ничего не получено)
 - g. **Y0.3** (GTIN не правильный)
12. Если коды признаны правильными по показателю количество и GTIN (если включена проверка), то они попадают в буфер отправки второго TCP сокета, который стоит в режиме клиента, и отправляются на ПК (в режиме сервера). Во всех других случаях коды не пересылаются.
13. Обнуляются все буферы и массивы. После этого система готова к принятию новых кодов.

Также, осуществляется подсчёт суммы всех счётчиков:

// Сумма всех счётчиков

```
ALL_COUNTERS_SUM := GOOD_CODES_COUNTER + BAD_CODES_COUNTER +  
NO_CODES_RECEIVED;
```

Сумма всех счётчиков должно быть равна сумме полученных сигналов от датчика, которая хранится в локальной переменной **STROB_CNT (POU STROB)**.

Программа запускается сама при подаче команды RUN на контроллер. Если камера и ПК открыли свои сокеты, то работа начнётся автоматически.

Но для корректной работы необходимо внести в программу следующие данные, которые будут в Вашей системе:

SEPARATOR, GTIN_ETALON и DMC_CODES_AMOUNT_SV.

Задать IP адреса камеры и ПК, а также номера портов контроллера, камеры и ПК:

В POU TCP_SOCKET

```
IP_CAMERA_1[0]:=16#C0A8; IP_CAMERA_1[1]:=16#010A; // IP address of the  
CAMERA 1 (192.168.1.10)  
CAMERA_1_PORT:=1201; // remote port 1201 (camera)  
PLC_PORT:=1200; // local port 1200 (PLC)
```

В POU TCP_SOCKET2

```
IP_PC[0]:=16#C0A8; IP_PC[1]:=16#0128; // IP address of the PC (192.168.1.40)  
PC_PORT:=1211; // remote port 1211 (PC)  
PLC_PORT:=1210; // local port 1210 (PLC)
```

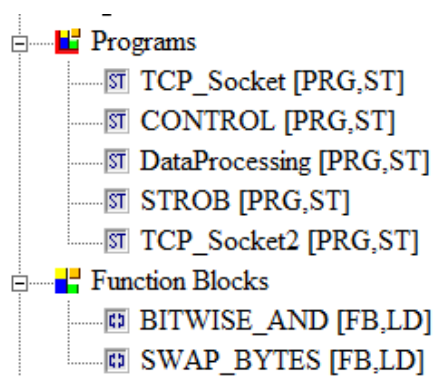
Также, можно скорректировать уставки таймеров удержания выходов (см. **POU CONTROL**), и задержки на чтение кодов (переменная **DELAY_FOR_CODE_READING**). Кроме того, для уменьшения времени отработки циклов оценки кодов очень имеет смысл уменьшить размер буферов приёма и отправки сокетов, а также размер массива структур типа BYTES, и счётчики циклов **Pointer1** и **Pointer3** в **POU DataProcessing** до реального количества байтов, которое посылает камера.

В примере стоит достаточно большой размер в 512 слов, т.е. 1024 байта. Если у Вас реально принимается 2 кода по 38 байт, то размер достаточно сделать следующий:

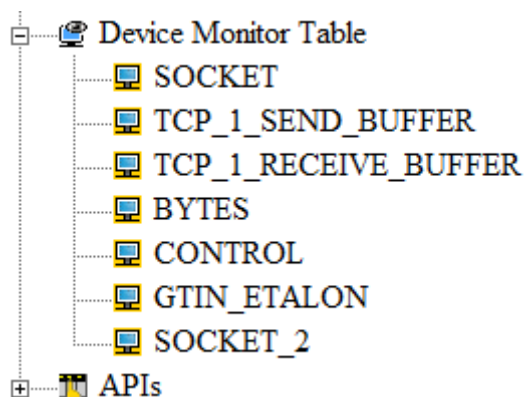
START_SYMBOL + 38 + SEPARATOR + 38 + END_SYMBOL = 79 байтов

Т.е. размер буферов, массивов и счётчиков циклов достаточно сделать 40 слов (80 байтов). Это существенно ускорит обработку кодов.

Для удобства работы программа разбита на структурные единицы (POU) в разделах Programs и Function Blocks:



А для целей работы в онлайн созданы таблицы мониторинга:



Основной таблицей является **CONTROL**, из которой можно управлять программой в онлайн режиме.

CONTROL					
	Object	Identifiers	Device Name	Status	Data Type
►	GlobalVar	DELAY_FOR_CODE_READING	D20014	<input type="text"/>	WORD
	GlobalVar	DMC_CODES_AMOUNT_SV	D20006	<input type="text"/>	WORD
	GlobalVar	STROB_CNT	D20026	<input type="text"/>	DWORD
	CONTROL	ALL_COUNTERS_SUM		<input type="text"/>	DWORD
	GlobalVar	GOOD_CODES_COUNTER	D20010	<input type="text"/>	DWORD
	GlobalVar	BAD_CODES_COUNTER	D20012	<input type="text"/>	DWORD
	GlobalVar	NO_CODES_RECEIVED	D20024	<input type="text"/>	DWORD
	GlobalVar	GTIN_CHECK	M32	<input type="text"/>	BOOL
	GlobalVar	GTIN_GOOD_COUNTER	D20020	<input type="text"/>	DWORD
	GlobalVar	GTIN_BAD_COUNTER	D20022	<input type="text"/>	DWORD
	GlobalVar	GTIN_GOOD	M10	<input type="text"/>	BOOL
	GlobalVar	GTIN_BAD	M11	<input type="text"/>	BOOL
	GlobalVar	CONNECTED_IN_CLIENT_MODE		<input type="text"/>	BOOL
	TCP_Socket	TCP_1_SOCKET_OPEN		<input type="text"/>	BOOL
	TCP_Socket	TCP_1_SOCKET_CLOSE		<input type="text"/>	BOOL
	TCP_Socket	CONNECTION_CLOSED		<input type="text"/>	BOOL
	GlobalVar	CONNECTED_IN_CLIENT_MODE2		<input type="text"/>	BOOL
	TCP_Socket2	TCP_2_SOCKET_OPEN	M25	<input type="text"/>	BOOL
	TCP_Socket2	TCP_2_SOCKET_CLOSE	M24	<input type="text"/>	BOOL
	TCP_Socket2	CONNECTION2_CLOSED	M14	<input type="text"/>	BOOL
	CONTROL	T_COUNTERS_RST		<input type="text"/>	BOOL

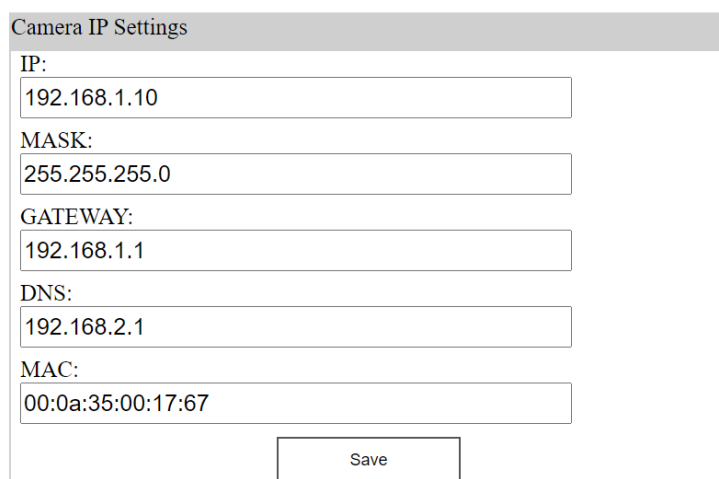
В данном примере использовалась камера Delta VIS-100. Но можно использовать любую смарт камеру, которая выдаёт коды стандартным TCP пакетом.

Камера и контроллер соединены по сети Ethernet.

Сетевые настройки контроллера:

IP - 192.168.1.5
MASK - 255.255.255.0
GATE - 192.168.1.1

Сетевые параметры камеры



Camera IP Settings

IP:
192.168.1.10

MASK:
255.255.255.0

GATEWAY:
192.168.1.1

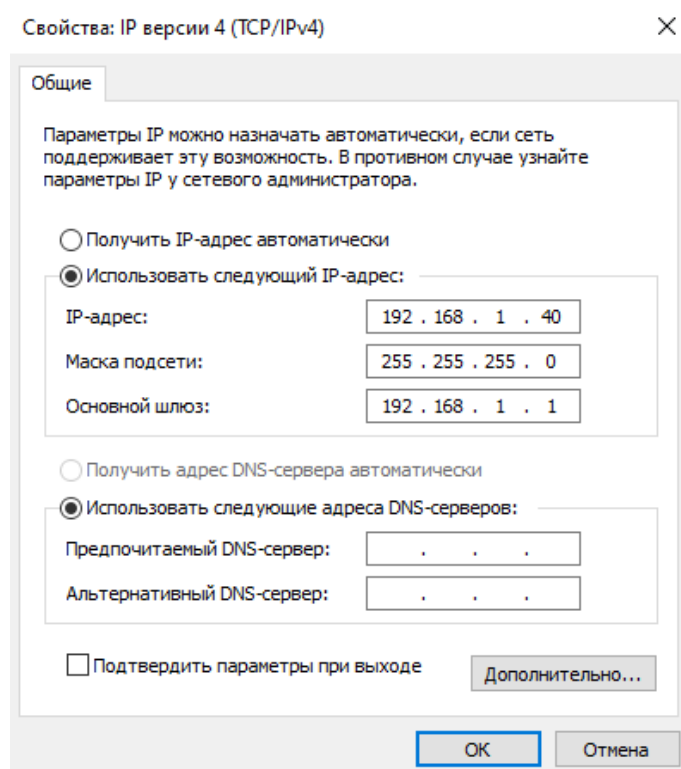
DNS:
192.168.2.1

MAC:
00:0a:35:00:17:67

Save

При смене IP адреса камере необходимо передёрнуть питание!

Для настроек камеры необходимо соединиться с ней через любой браузер, предварительно настроив сетевую карту ПК (ноутбука) на такую же сеть:



Свойства: IP версии 4 (TCP/IPv4)

Общие

Параметры IP можно назначать автоматически, если сеть поддерживает эту возможность. В противном случае узнайте параметры IP у сетевого администратора.

☐ Получить IP-адрес автоматически

☒ Использовать следующий IP-адрес:

IP-адрес: 192 . 168 . 1 . 40

Маска подсети: 255 . 255 . 255 . 0

Основной шлюз: 192 . 168 . 1 . 1

☐ Получить адрес DNS-сервера автоматически

☒ Использовать следующие адреса DNS-серверов:

Предпочитаемый DNS-сервер: . . .

Альтернативный DNS-сервер: . . .

☐ Подтвердить параметры при выходе

Дополнительно...

ОК Отмена

Для входа в меню камеры необходимо зайти во вкладку Login:



и ввести:

admin
admin

Login:

Username:

admin

Password:

.....

Login

Далее выполнить настройку камеры для условий данного примера, т.е. статическая картинка с кодами при обычном освещении (большая выдержка и усиление). Для наглядности работы программы таймер 1000 мс.

☒ Boot Autorun

Name: VIS100

Output Image: Yes Every: 1 Process

Result Logic AND

Crop image: Auto

Size: 1280x960

Exposure: 5500

Gain: 40

Retry(times): 0

Compare Result(times): 0

Trigger: Continuous

Software Trigger Cycle(ms): 1000

Light: None

Для вступления изменений в силу необходимо перезапустить камеру на Главной странице (Main):



Количество и тип кодов выбирается во вкладке **Application** :

Main

Application

☐ Filter

☐ Barcode

☒ DMC

☐ QR

☐ Micro QR

Configuration

Camera

Comm.

ROI: Complete ▾

Num. of Code: 9

Timeout(ms): 1000

☐ Code Length: 3

☐ Code Content: 123,***

☐ Code Quality: ISO/IEC 155415 ▾ Overall ▾

3

Настройки связи **Comm**:

TCP Client Send Image:

Send image: Disable ▾

Comm. Setting:

TCP Server ▾ Always ▾

Port: 1201

Header: 10 Ascii(DEC)

Seperator: 59 Ascii(DEC)

EOF: 13 Ascii(DEC)

В режиме TCP сокета порт контроллера 1200, порт камеры 1201.

Режим камеры **Mode**:

Mode Selection

☒ Code Reader

Внимание!!!

При работе камеры в режиме чтения кодов её необходимо установить на алюминиевый радиатор!

В противном случае камера выйдет из строя от перегрева.

Для симуляции работы камеры и сервера на ПК можно воспользоваться бесплатной прилагаемой к данному примеру утилитой **TCP_UDP_Sockets_Builder_setup.exe**

В разделе Local IP указывается IP адрес сетевой карточки ПК и порт как бы камеры 1201. Далее нужно нажать «Create Socket», а затем «Listen». Утилита откроет сокет в режиме сервера и будет ждать приглашения от контроллера (клиента) установить соединение. После установления соединения в поле IP должен появиться адрес контроллера 192.168.1.5 и его порт 1200.

Далее взяв образцы кодов из прилагаемого к примеру файла **codes_for_sending.txt**, отправлять на контроллер коды из поля **Send data**.

Таким же образом запустив второй сеанс данной утилиты можно симитировать ПК в режиме сервера, принимая от контроллера коды в поле **Receive data**. (IP адрес будет такой же, а порт 1211. Порт контроллера 1210).