

Описание примера работы контроллера AS320T с камерой чтения кодов VIS100

В данном примере камера VIS100 считывает группу кодов, а контроллер AS320T их принимает в режиме TCP или UDP сокета и проверяет количество принятых кодов путём подсчёта разделителей (запятые).

Используются одинаковые ДМС коды следующего вида:



Код считывается набором ASCII символов: df98674ygk6940furp7802365

В настройках камеры в качестве стартового символа выбран «-», разделитель «,», стоповый символ «.».

В приёмный буфер сокета будет попадать пакет следующего вида (отличие только в количестве кодов):

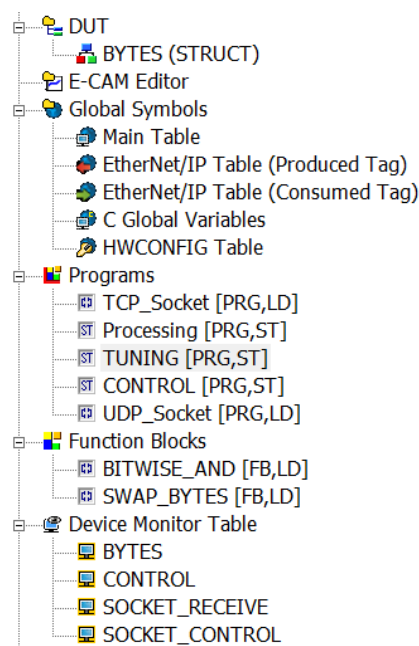
df98674yggk6940furp7802365,df98674yggk6940furp7802365,df98674yggk6940furp7802365,df98
674yggk6940furp7802365,df98674yggk6940furp7802365,df98674yggk6940furp7802365,df98674y
ggk6940furp7802365,df98674yggk6940furp7802365,df98674yggk6940furp7802365,df98674yggk69
40furp7802365,df98674yggk6940furp7802365,df98674yggk6940furp7802365,df98674yggk6940fur
p7802365,df98674yggk6940furp7802365,df98674yggk6940furp7802365,df98674yggk6940furp780
2365,df98674yggk6940furp7802365,df98674yggk6940furp7802365.

Контроллер обрабатывает пакет в следующей последовательности:

1. Камера постоянно по таймеру осуществляет сканирование своей области обзора. При появлении предмета с кодами камера осуществляет их чтение и по шине Ethernet выдаёт пакет в формате TCP/UDP сокета.
2. Контроллер по сигналу с датчика о появлении предмета с кодами осуществляет обработку пакета, который находится у него в приёмном буфере сокета (в буфер пакет попадает автоматически, сразу как камера его отправляет).
3. В контроллере регистры 16 бит, следовательно в каждом регистре будет находиться по 2 ASCII символа. Для дальнейшей обработки они разбиваются на отдельные регистры. Т.е. чтобы в одном регистре лежал один ASCII символ.

4. Далее весь массив символов прокручивается в цикле с подсчётом запятых, т.е. разделителей (по количеству разделителей можно судить о том, сколько камерой прочитано кодов).
5. Полученное количество разделителей минус 1 (т.к. после последнего символа нет запятой) сравнивается с заданным значением.
6. Исходя из результата сравнения делается вывод о том, было ли чтение кодов удачным или неудачным.
7. Увеличивается соответствующий счётчик удачных или неудачных попыток чтения.
8. Включается соответствующий выход контроллера с удержанием на 100 мс.
9. Осуществляются сброс буфера и флагов.

Программа контроллера состоит из 5 POU (Programs), 2-х функциональных блоков (Function Blocks), 1 структуры BYTES и для удобства работы созданы 4 таблицы мониторинга (Monitor Table):



Функционал программных единиц (POU) следующий:

CONTROL – общее управление процессом открытия/закрытия сокета, чтения и анализа кодов, подсчёта удачных/неудачных попыток чтения, включения/выключения выходов контроллера

Processing – Содержит процедуры: обработки кодов, принятых в приёмный буфер сокета; раскладки байтов по регистрам; подсчёта разделителей; сравнения с заданным количеством; очистка буфера и сброса флагов.

TCP_Socket – процедуры конфигурирования, открытия и закрытия TCP сокета

UDP_Socket – процедуры конфигурирования, открытия и закрытия UDP сокета

TUNING – процедура заполнения приёмного буфера сокета в целях отладки. Используется при отсутствии камеры.

Основные переменные, используемые для управления процессом:

Для удобства находятся в глобальных переменных (Main Table).

SOCKET_RECEIVE	Буфер приёма сокета Сюда поступают коды от камеры в исходном виде Т.е. два ASCII символа в одном регистре
SENSOR	Сигнал с датчика появления предмета с кодами для считывания
TUNING	Включение режима отладки
FILL_TCP_BUFFER	Автозаполнение входного буфера сокета с целью имитации пакета от камеры
BAD_CODES_SV	Количество попыток чтения кодов
GOOD_FLAG	Внутренний флаг удачного чтения кодов
BAD_FLAG	Внутренний флаг неудачного чтения кодов
GOOD_CODES_COUNTER	Итоговый счётчик удачных попыток чтения кодов
BAD_CODES_COUNTER	Итоговый счётчик неудачных попыток чтения кодов
SEPARATOR	Разделитель кодов Задаётся как ASCII символ в HEX
QR_CODES_AMOUNT_SV	Заданное количество кодов которое нужно проверять путём подсчёта разделителей

Переменная **SENSOR** активируется входом **X0.4**.

(Для начала работы нужно задать значения переменных **QR_CODES_AMOUNT_SV** и **BAD_CODES_SV**)

Для удобства работы данные переменные собраны в таблице мониторинга **CONTROL** :

CONTROL					
	Object	Identifiers	Device Name	Status	Data Type
	Processing	GOOD_COUNTER	D20002	<input type="text"/>	DWORD
	Processing	BAD_COUNTER	D20004	<input type="text"/>	DWORD
	Отладка	QR_CODES_AMOUNT_SV_Test	D20007	<input type="text"/>	WORD
	GlobalVar	TUNING	M22	<input type="text"/>	BOOL
	GlobalVar	GOOD_CODES_COUNTER	D20010	<input type="text"/>	DWORD
	GlobalVar	BAD_CODES_COUNTER	D20012	<input type="text"/>	DWORD
	GlobalVar	BAD_CODES_SV	D20008	<input type="text"/>	WORD
	GlobalVar	QR_CODES_AMOUNT_SV	D20006	<input type="text"/>	WORD

Переменная **QR_CODES_AMOUNT_SV_Test** в режиме отладки (**TUNING**) задаёт количество кодов, которые будут сгенерированы самим контроллером. Используется в случае отсутствия камеры для изучения работы алгоритмов в программе.

Переменные управления сокетом собраны в таблице мониторинга **SOCKET_CONTROL** :
(являются локальными переменными POU CONTROL)

SOCKET_CONTROL					
	Object	Identifiers	Device Name	Status	Data Type
	CONTROL	START_TCP	M20	<input type="text"/>	BOOL
	CONTROL	STOP_TCP	M21	<input type="text"/>	BOOL
	CONTROL	TCP1_OK		<input type="text"/>	BOOL
	CONTROL	TCP1_ERROR		<input type="text"/>	BOOL
	CONTROL	START_UDP2	M18	<input type="text"/>	BOOL
	CONTROL	STOP_UDP	M19	<input type="text"/>	BOOL
	CONTROL	UDP2_OK		<input type="text"/>	BOOL
	CONTROL	UDP2_ERROR		<input type="text"/>	BOOL

Приёмный буфер сокета выведен в таблицу мониторинга **SOCKET_RECEIVE** :

Object	Identifiers	Device Name	Data Type	Value (16bits)	Radix
GlobalVar	SOCKET_RECEIVE[0]	D2000	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[1]	D2001	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[2]	D2002	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[3]	D2003	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[4]	D2004	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[5]	D2005	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[6]	D2006	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[7]	D2007	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[8]	D2008	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[9]	D2009	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[10]	D2010	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[11]	D2011	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[12]	D2012	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[13]	D2013	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[14]	D2014	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[15]	D2015	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[16]	D2016	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[17]	D2017	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[18]	D2018	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[19]	D2019	WORD		ASCII
GlobalVar	SOCKET_RECEIVE[20]	D2020	WORD		ASCII

Буфер, разложенный на отдельные символы по регистрам выведен в таблицу **BYTES** :

Object	Identifiers	Device Name	Data Type	Value (16bits)	Radix
Processing	Pointer1		WORD		Signed Decimal
Processing	ARRAY_BYTES[0] HIGHBYTE	D4000	WORD		ASCII
Processing	ARRAY_BYTES[0] LOWBYTE	D4001	WORD		ASCII
Processing	ARRAY_BYTES[1] HIGHBYTE	D4002	WORD		ASCII
Processing	ARRAY_BYTES[1] LOWBYTE	D4003	WORD		ASCII
Processing	ARRAY_BYTES[2] HIGHBYTE	D4004	WORD		ASCII
Processing	ARRAY_BYTES[2] LOWBYTE	D4005	WORD		ASCII
Processing	ARRAY_BYTES[3] HIGHBYTE	D4006	WORD		ASCII
Processing	ARRAY_BYTES[3] LOWBYTE	D4007	WORD		ASCII
Processing	ARRAY_BYTES[4] HIGHBYTE	D4008	WORD		ASCII
Processing	ARRAY_BYTES[4] LOWBYTE	D4009	WORD		ASCII
Processing	ARRAY_BYTES[5] HIGHBYTE	D4010	WORD		ASCII
Processing	ARRAY_BYTES[5] LOWBYTE	D4011	WORD		ASCII
Processing	ARRAY_BYTES[6] HIGHBYTE	D4012	WORD		ASCII
Processing	ARRAY_BYTES[6] LOWBYTE	D4013	WORD		ASCII
Processing	ARRAY_BYTES[7] HIGHBYTE	D4014	WORD		ASCII
Processing	ARRAY_BYTES[7] LOWBYTE	D4015	WORD		ASCII
Processing	ARRAY_BYTES[8] HIGHBYTE	D4016	WORD		ASCII
Processing	ARRAY_BYTES[8] LOWBYTE	D4017	WORD		ASCII
Processing	ARRAY_BYTES[9] HIGHBYTE	D4018	WORD		ASCII
Processing	ARRAY_BYTES[9] LOWBYTE	D4019	WORD		ASCII
Processing	ARRAY_BYTES[10] HIGHBYTE	D4020	WORD		ASCII
Processing	ARRAY_BYTES[10] LOWBYTE	D4021	WORD		ASCII
Processing	ARRAY_BYTES[11] HIGHBYTE	D4022	WORD		ASCII
Processing	ARRAY_BYTES[11] LOWBYTE	D4023	WORD		ASCII

Структура **BYTES** содержит массив из двух регистров. В программе объявлен массив данных структур **ARRAY_BYTES**. Её задача принять в каждый регистр по одному ASCII символу. Далее весь массив структур прокручивается в цикле на предмет поиска запятых.

Функциональный блок **SWAP_BYTES** осуществляет перестановку байтов в слове. На вход подаётся исходное слово, а на выходе будет слово с поменяными местами старшим и младшим байтами.

Функциональный блок **BITWISE_AND** осуществляет побитовую операцию «И» с двумя входными регистрами и на выход подаётся результат операции.

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

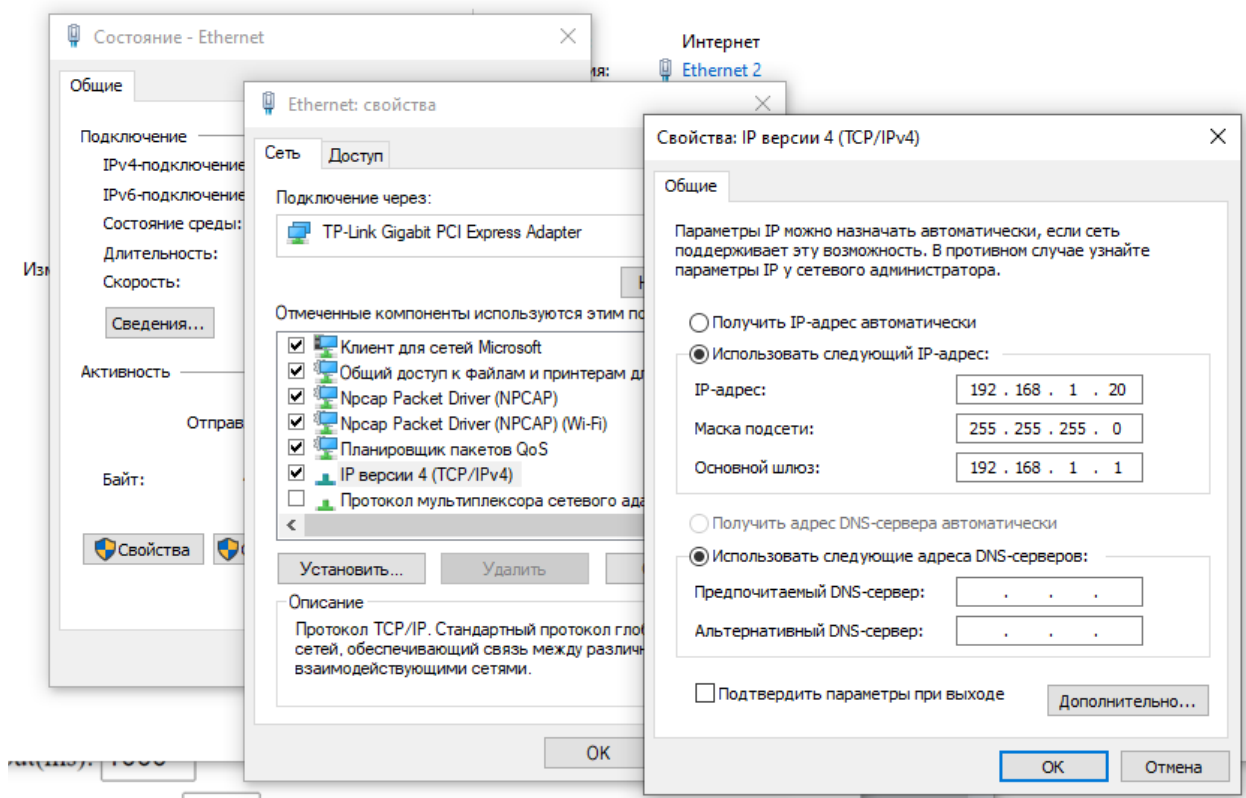
Save

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:

Просмотр основных сведений о сети и настройка подключений

Просмотр активных сетей



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 сохранения настроек необходимо нажимать:



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

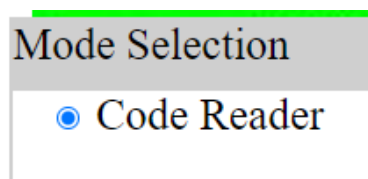
Header: 45 Ascii(DEC)

Seperator: 44 Ascii(DEC)

EOF: 46 Ascii(DEC)

В режиме TCP сокета порт контроллера 1200, порт камеры 1201.
Для UDP сокет порт контроллера 1300, порт камеры 1301.

Режим камеры **Mode**:



Данные настройки можно импортировать из файла:

settings.json

Внимание!!!

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

В противном случае камера выйдет из строя от перегрева.