1 TABLE OF CONTENTS

2 DESCRIPTION AND KEY SPECIFICATIONS

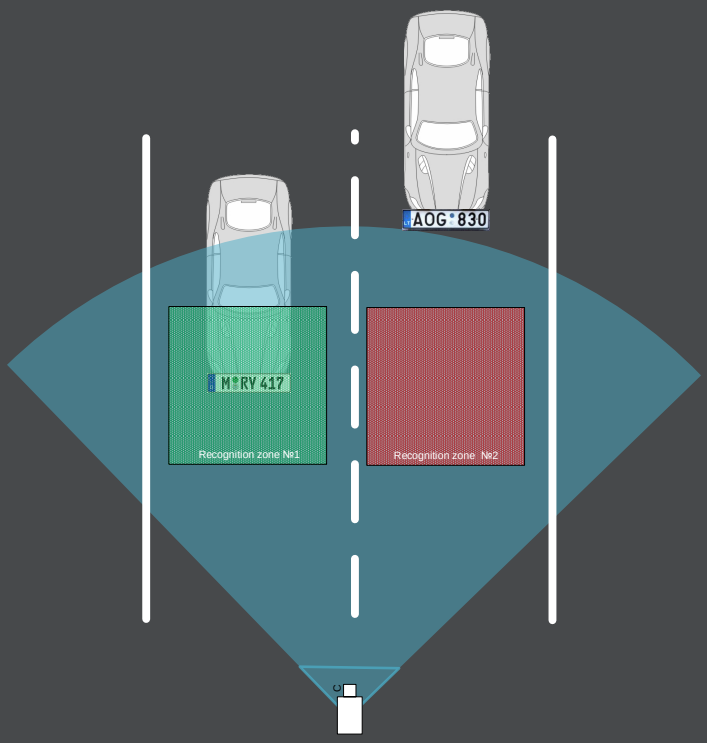
NumberOK software hardware complex is designed to recognize license plate numbers and control executive units.

TABLE 1 KEY SPECIFICATIONS

Name	Parameters
OS	Windows
Security system	• Electronic keys • Guardant dongles
Number of connected video streams	1,2,4
Number of recognition zones in a single video stream	Up to 4
Video stream rate	Up to 8 Mbit/s
Video resolution	Up to 3 MP
Recognition rate	10-100 ms
Type of recognized license plate numbers	Countries of EU, CIS, Israel
Recognition accuracy	>95%
Vehicle speed	Up to 200 km/h
Number of executive units (gates, barriers), which can be controlled	Up to 4
Number of alarm sensors (photocells, induction loops)	Up to 8

3 SCOPE OF APPLICATION

3.1 VEHICLE CONTROL SYSTEMS



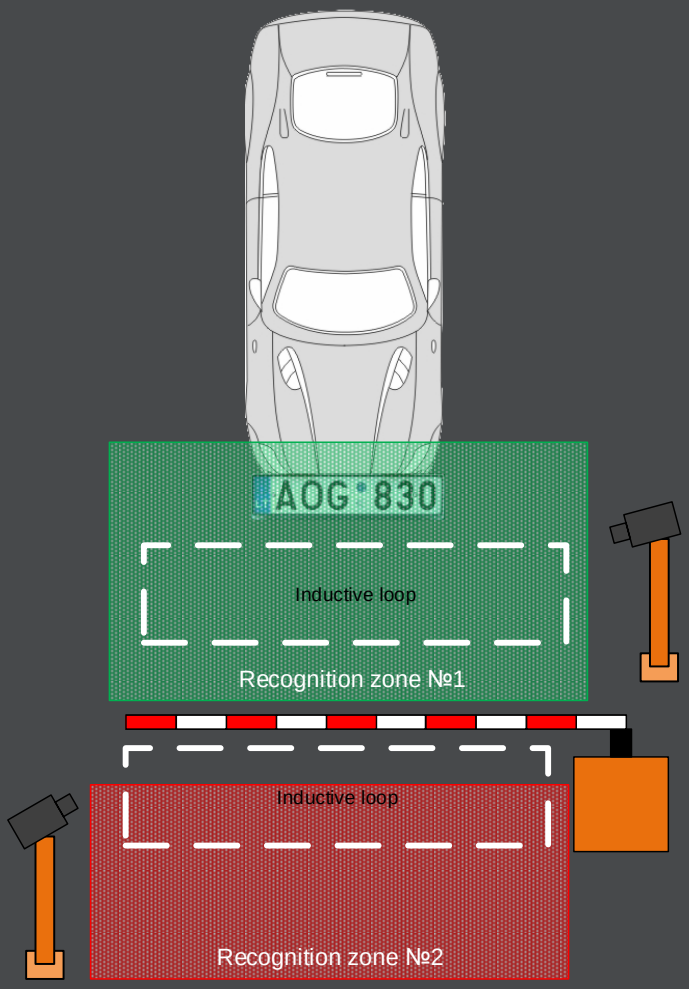
PICTURE 1

TRAFFIC CONTROL

Main objectives:

- Recognize number plates and store them in the database
- Determine the direction of vehicle
- Search database for certain vehicle (stolen, debtors)
- Display obtained data in the external systems
- Generate reports for any period of time:
  - Detailed
    - Date and time
    - License plate number
    - Direction of travel
    - Channel, recognition zone
    - Snapshot of vehicle
  - General – number of vehicles which passed through preset recognition zone in specified direction

3.2 CHECKPOINTS. PARKING AREAS



PICTURE 2. CHECKPOINT

Main objectives:

- Control vehicle access to the territory
- Detect entry of the vehicle and its direction
- Control barriers (gates) and obtain data from photo sensors (induction loops)
- Determine duration of stay at the area
- Generate reports:
  - Detailed (*see basic mode*) +
    - Duration of stay at the area
  - General (for a period of time)
    - Number of vehicles that entered /left the area or passed through the selected checkpoint
    - Number of vehicles at the area
    - Overall number of vehicles/hours

4 RECOMMENDATIONS ON SYSTEM INSTALLATION

4.1 CONNECTION OF VIDEO SOURCES AND ENHANCEMENT OF RECOGNITION ACCURACY

4.1.1 VIDEO SOURCES

NumberOk receives and decodes video streams in the following formats:

- Live 555 library
  - RTSP stream
- FFMPEG library
  - AV_CODEC_ID_H264
  - AV_CODEC_ID_MPEG4
  - AV_CODEC_ID_HEVC

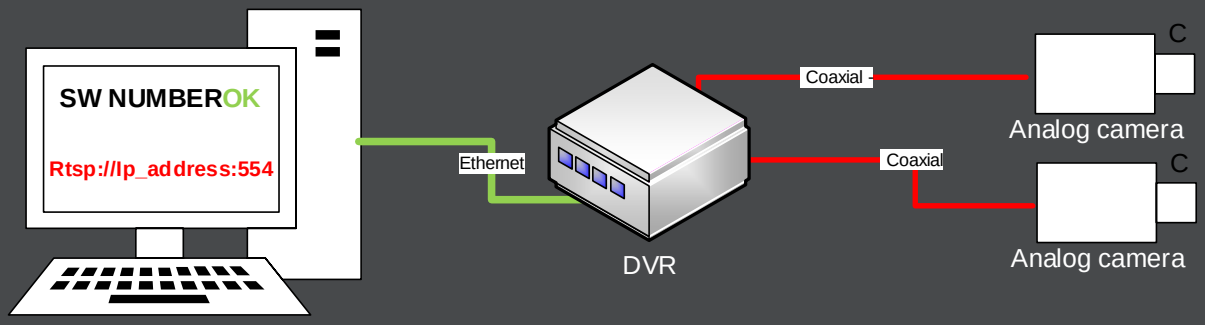
1. Source 1 - IP cameras supporting RTSP
2. Source 2 – analogue cameras connected via DVRs:
   a. DVR Hikvision
   b. DVR Dahua
   c. DVR TVT
   d. DVR NOVUS
   e. DVR NUUO
   f. CBH Line
   g. Any DVR supporting RTSP
3. Source 3 – any video source supporting RTSP.
4. Video files stored in the computer where NumberOk is installed

- CODEC: H264
- Container: AVI, MP4, TS

For more detailed information on connecting video sources to NumberOk refer to the section Connection of video sources

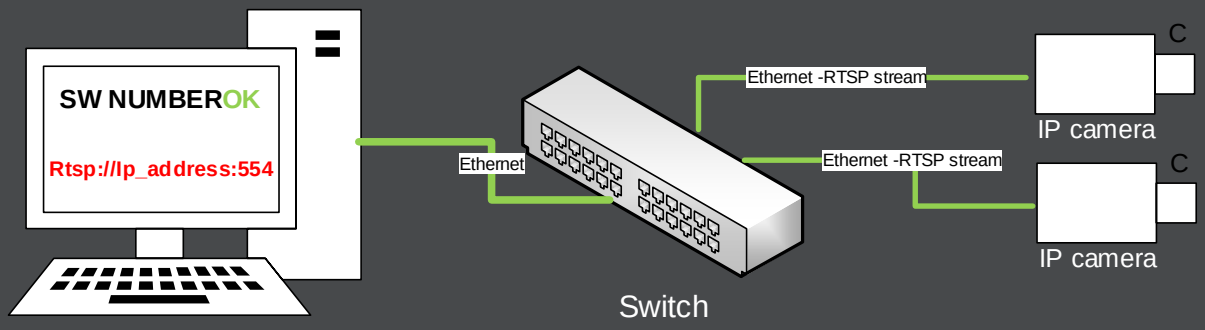
4.1.2 CONNECTION DIAGRAMS

4.1.2.1 CONNECTION DIAGRAM FOR ANALOGUE CAMERA AND DVR



PICTURE 3. DIAGRAM OF CONNECTION ANALOGUE CAMERAS

4.1.2.2 CONNECTION DIAGRAM FOR IP CAMERAS



PICTURE 4. CONNECTION DIAGRAM FOR IP CAMERAS

4.1.3 REQUIREMENTS FOR HIGH-QUALITY LICENSE PLATE RECOGNITION

4.1.3.1 SIZE OF LISENCE PLATE NUMBER

PICTURE 5. REQUIREMENTS FOR RECOGNITION -SIZE OF NUMBER PLATE

- Minimum size of number plate in the frame - 130 pixels
- Maximum size of number plate in the frame - 500 pixels
- Recommended size of number plate in the frame – 150-200 pixels

The parameters are provided regardless of camera resolution: from D1 (704x576) to 3 MP(2048x1536).

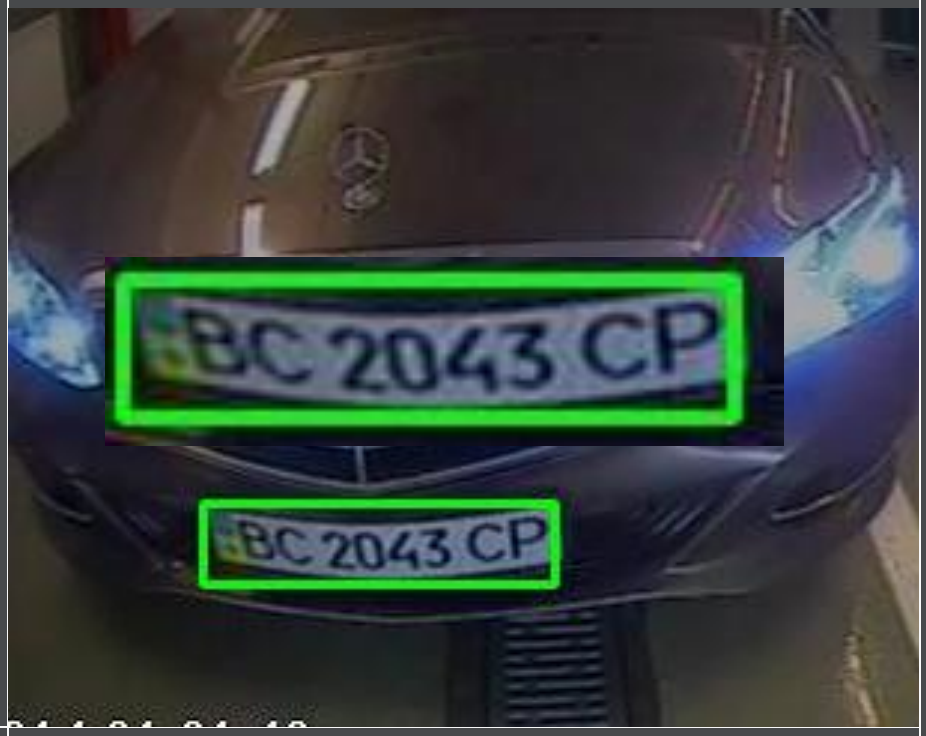
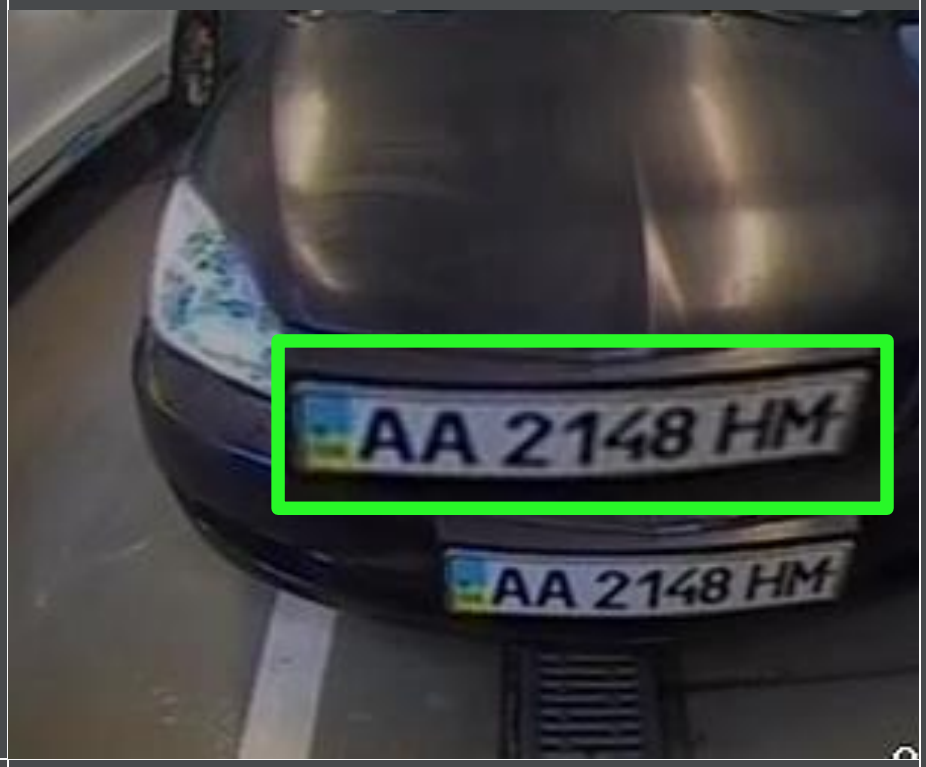
4.1.3.2 IMAGE QUALITY

The image of license plate number should be crisp, contrast and easy-to-read.

TABLE 2 ASSESMENT OF IMAGE QUALITY FOR LICENSE PLATE RECOGNITION

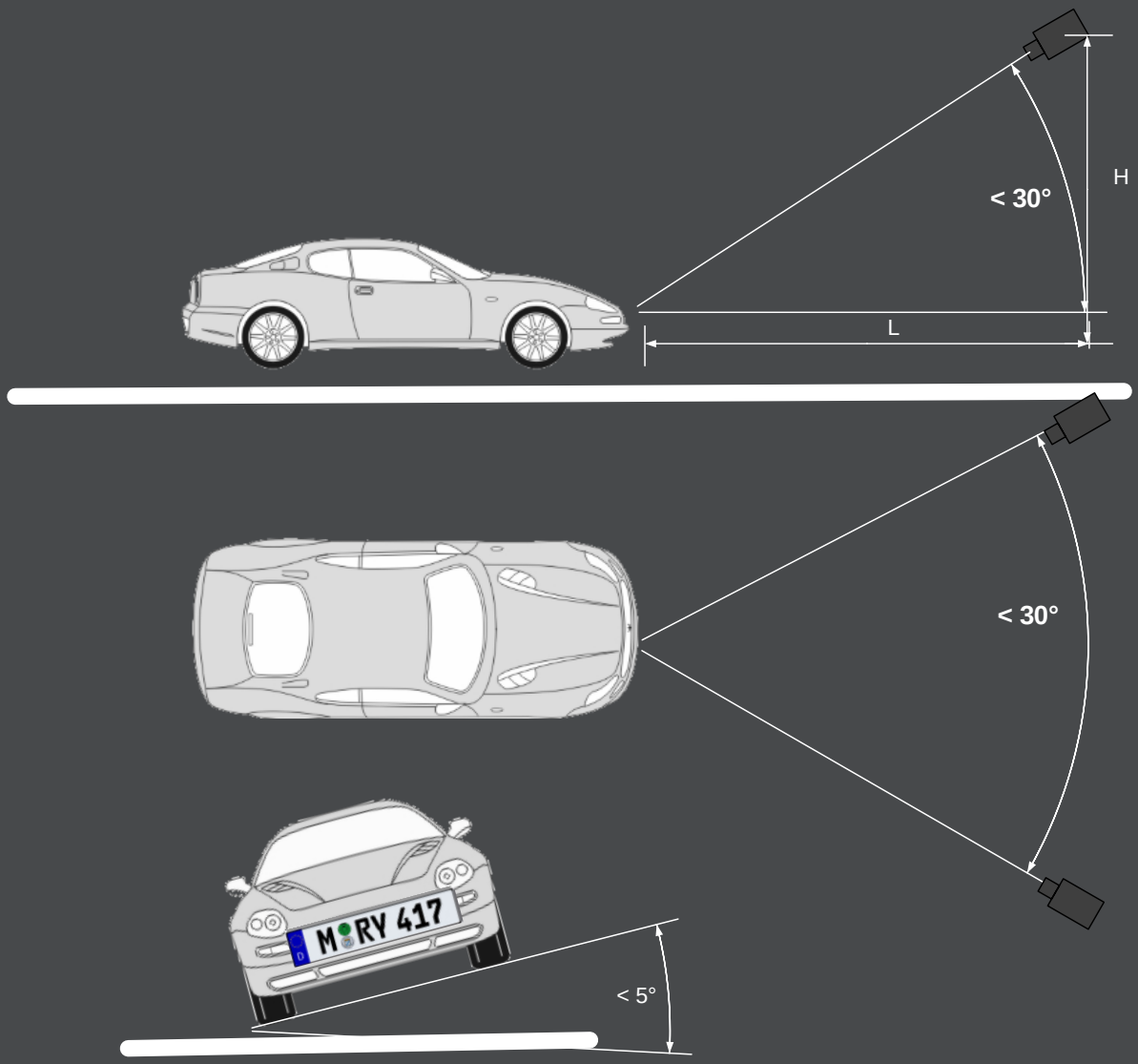
GREAT License plate number is clear and easy-to-read	
BAD Rear license plate number is partly covered by another car and there is a possibility of erroneous determination of the last symbol. Nonetheless the image is clear. The front license plate number is not decipherable. Camera lens provides shallow depth of field and cannot deliver clear image.	

BAD The image is not clear due to the low frame rate or wrongly set shutter speed. The faster travelling speed of the vehicle is, the lower this parameter should be. Available values:1/250, 1/500,1/1000	
BAD The image is spoiled by the headlights.	

BAD The camera lens caused geometric distortion of the image.	
BAD Geometric distortion of the image is due to the large installation angle of the camera in vertical plane.	

4.2 RECOMMENDATIONS ON SELECTION AND INSTALLATION OF CAMERAS

4.3 INSTALLATION ANGLE AND DISTANCE TO LICENSE PLATE NUMBER

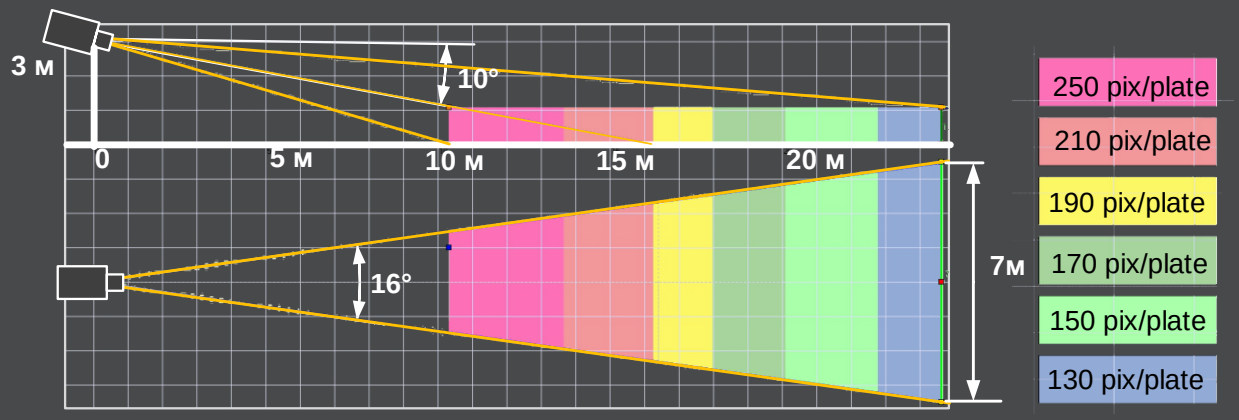


PICTURE 7. REQUIREMENTS FOR CAMERA POSITION

- Installation angle for vertical mounting – up to 30°
- Installation angle for horizontal mounting – up to 30°
- Distance to the vehicle L can vary provided all the requirements are met, namely installation angles and size of number plate.
- Vertical distance from the camera to the ground plane H can vary, provided installation angle in vertical plane is less than 30°
- Roll angle – no more than +-5°

4.3.1 CAMERA SELECTION

The picture below illustrates the installation example for NumberOK system.



Matrix – 1/32";Rseolution 1920*1080; focal length – 16мм

PICTURE 8 EXAMPLE OF SYSTEM INSTALLATION

Requirements: detect license plate numbers within the 10-m long and 3-m wide area. (See Picture 12 – the area is indicted with red rectangle)

Steps:

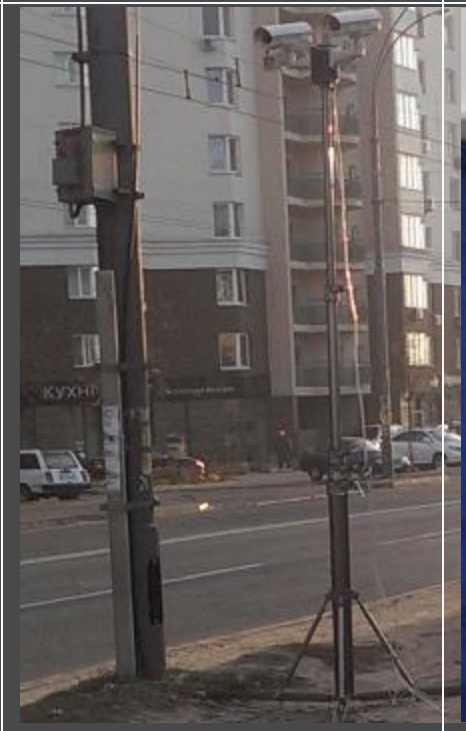
1. Select installation point of the camera, taking into account technical conditions at the site. In the present case installation height of the camera is 3 m and the distance to recognition zone is 14 m with camera focused at the centre of the zone. Therefore we obtain tilt angle of 10° in vertical plane (with accordance to the requirement for recognition (see paragraph 4.2)
2. Select lens. Focal length should be adjusted in a way to provide maximum angle of view for the camera. In the present case the focal length of the lens we use is 16 mm.
3. Select camera resolution. In our case minimum resolution is FULL HD , taking into account the requirements for the size of number plate to be at least 130 pixel (see paragraph 4.1.3.1)

Conclusion: According to the results obtained, we can choose IP camera with 1920x1080 resolution or analogue camera HD CVI with the same resolution.

We have determined one possible installation point of the camera. The similar calculation can be performed for all available points in order to choose the most suitable.

The table below illustrates two examples of camera installation.

TABLE 2 STANDARD EXAMPLES OF CAMERA MOUNTING

	Example 1	Example 2
Camera type	IP 	Analogue 
Resolution	1920*1080	704*576
Lens	Fixed 6 mm	Varifocal 6 m
Specific characteristics of the installation		
Installation height	3m	1m
Field of view		
Width of recognition zone	12m	4 m

4.4 CONNECTION OF EXECUTIVE UNITS

4.4.1 CONTROL MODULES

NumberOk supports two control modules:

- Barbos control module
- ICP CON PEN-7060 control module, supporting protocol ModBUS.

Two Barbos units and three ICP CON devices can be connected to NumberOK.

Control modules allow you to connect up to 4 barriers and receive data from 4 pairs of alarm sensors (induction loops or photo sensors).

4.4.1.1 BARBOS CONTROL MODULE

PICTURE 9. BARBOS CONTROL MODULE

Barbos is designed to control executive units via 4 channels of relay outputs, as well as receive information on the condition of other devices by means of the 4 channels of alarm outputs.

Barbos is controlled by NumberOK via *USB* interface. The device is easy to install, operate and maintain. It is available in plastic housing and can be mounted without external box. All connectors are removable and furnished with explanatory symbols which make for an easy connection process.

TABLE 3. KEY SPECIFICATIONS

Alarm inputs	
Number of channels	4
Actuation time of alarm outputs	100 ms
Optical isolator	yes
Isolation voltage rating	1 kV
Relay outputs	
Number of channels	4
Isolation voltage rating	1 kV
General	
Power supply	5V DC (from USB)
Power consumption	no more than 500 mA
Overall dimensions (Wx D x H)	134 x 89 x 34 mm
Operating temperature range	from -20 to 50°C
Operating humidity	Up to 90%

4.4.1.2 ICP DAS PET-7060 CONTROL MODULE

PICTURE 10. ICP CON PET 7060 CONTROL MODULE

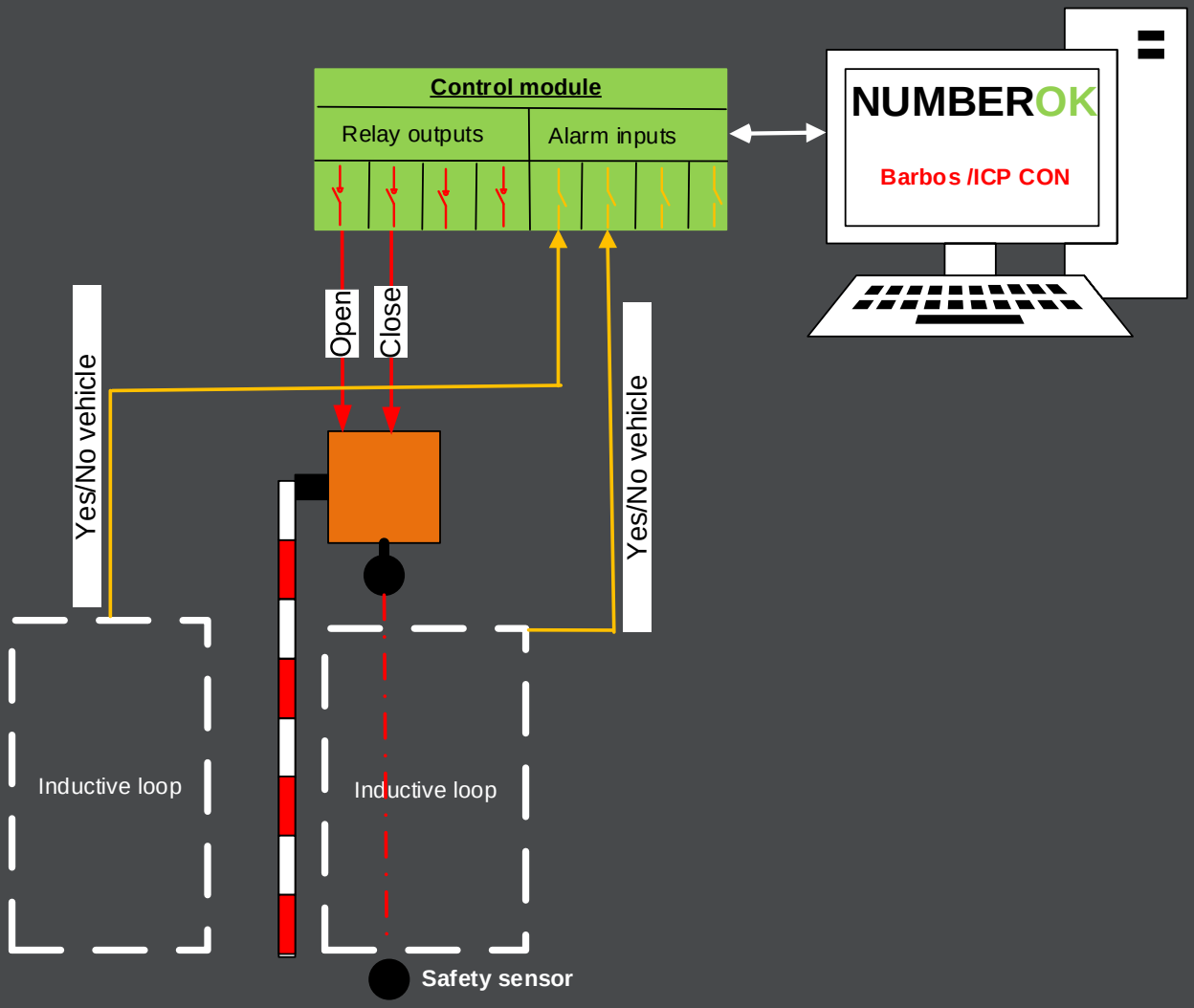
ET-7060 is a module for 6-channel isolated discrete input with 32-bit counter and 6-channel relay output with integrated web-server. The device is designed for storage and control of data, and configured to support Modbus/TCP providing remote input/output system via Ethernet.

TABLE 4. KEY SPECIFICATIONS

Manufacturer	
Name:	ICP DAS
Network interfaces	
Port10/100 Mbit/s:	1
Port connectors 10/100 Mbit/s:	RJ-45
Industrial interfaces/protocols	
Support Modbus TCP:	Slave
Support Modbus UDP:	Slave
Discrete input	
Overall number of channels for discrete input	6

Type:	Contact with external power supply
Receiver/Source:	Receiver, Source
Logic 1 for contact with external power supply:	10 ... 50 V
Logic 0 for contact with external power supply:	0 ... 4 V
Counter:	500 Hz, 32 bit
Discrete output	
Overall number of channels for discrete output:	6
Type:	Electromechanical relay
Relay with normally open contacts (A):	6
Maximum switching current for relay with contacts (A):	5A with 250V AC, 5A with 30V AC
Switching time:	6 ms
Input voltage DC:	10 ... 30 V
Power consumption:	2.9 W
Protection at power supply input:	Protected against polarity reversal
Operating conditions	
Operating temperature range:	-25 ... 75 °C
Operating humidity:	10 ... 90 % (non-condensing)

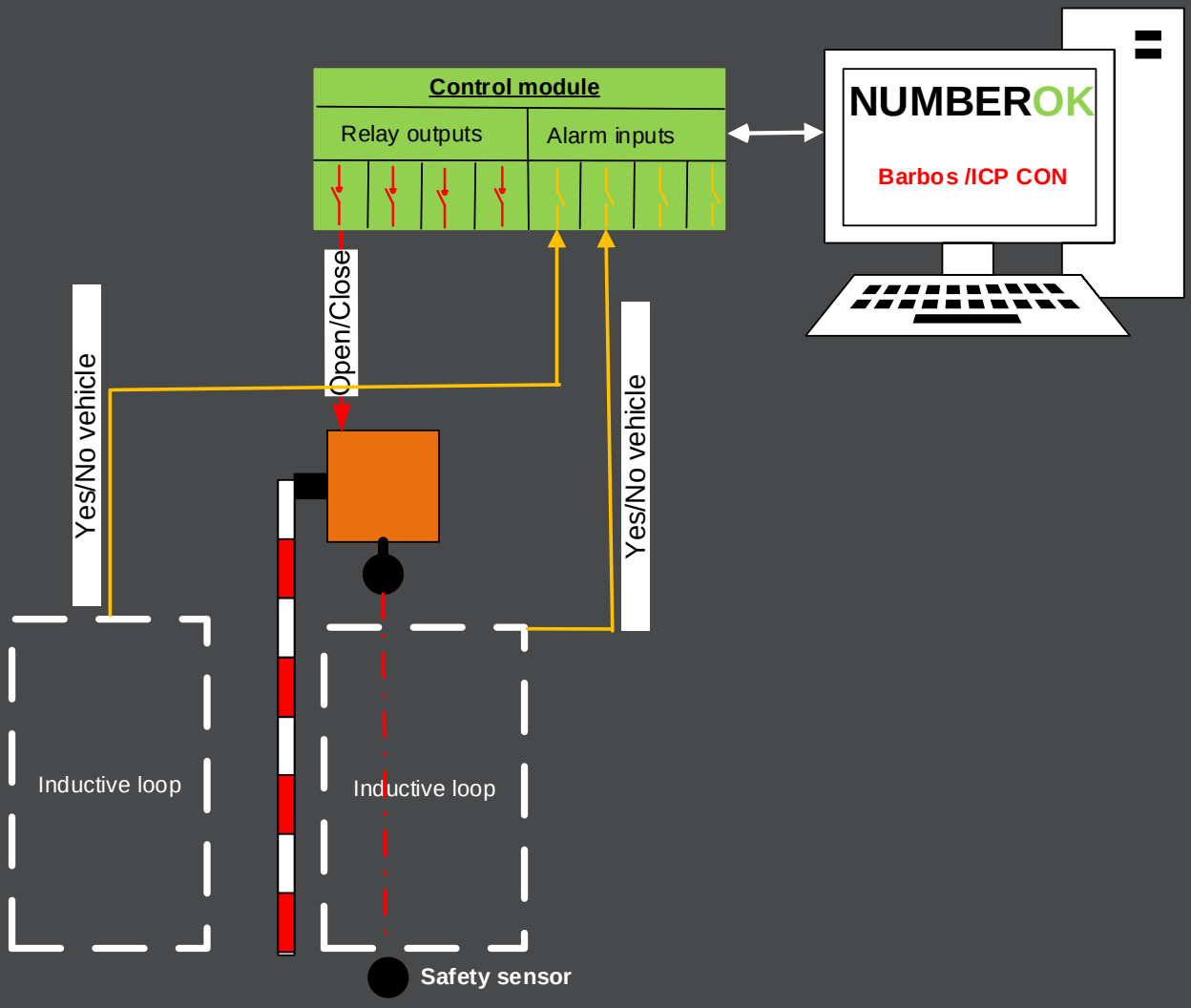
4.4.2 BARRIER CONTROL IN AUTOMATIC MODE (BASIC)



PICTURE1. MULTI-USER OPERATING MODE

Output relays of control module are connected to barrier controller in this mode. In this case two relays are used: the first one transmits the command «open», and the second one - «close». Relays of control module can me connected in parallel with barrier operating buttons. The control unit of barrier and control module can work simultaneously without mutual interference. This operating mode is the most preferred and secured. Control module can be connected to induction loops or normally open photo sensors NC. It is required in case the user wants to detect entry of the vehicle by alarm sensors. Use the link available to see functionality algorithm

4.4.3 BARRIER CONTROL IN DISCRETE MODE



PICTURE 12. DISCRETE OPERATING MODE

In this mode output relays of control module are connected to barrier controller. One relay is involved in the process. When the relay is closed for the first time the signal is given to the control unit to open executive mechanism. When the relay is closed for the second time the control unit performs closing of the barrier. Relay of control module can me connected in parallel with barrier operation button. Simultaneous operation of barrier control panel and control module is possible but unwanted. NumberOK does not track the commands of control unit and cannot determine position of executive mechanism (opened or closed). This operating mode is not secure. Control module can be connected with induction loops or normally open photo sensors NC. It is required in case the user wants to detect entry of the vehicle by alarm sensors. Use the link available to see functionality algorithm

4.5 COMPUTER REQUIREMENTS

TABLE 5. SELECTION OF PROCESSOR

Channel	1	2	4
Intel processor	**intel i3 4330**	**Intel i5 4440**	**intel i7 4770K**
RAM	4 Gb	4 Gb	8 Gb
HDD	50 GB	150 GB	300 GB
Operating system	Windows XP/7/2013/8/		

5 SOFTWARE

5.1 TYPES OF SOFTWARE

The software is classified by the following parameters:

1. By number of channels
    a. One channel
    b. Two channels
    c. Four channels
2. By regions
    a. CIS countries
    b. EU countries + Israel
3. By duration
    a. Full version – unlimited time.
    b. Demo version – 30 days

5.2 OPERATION MODES

NumberOK provides the following operation modes:

I. Recognition (basic mode).
Basic functions:
- License plate recognition
- Determination of vehicle direction
- Groups of vehicles
- Reports
- Transfer of recognized number plates into external applications

Basic parameters for number plate transfer

- License plate number
- License plate description -the field can be blank if license plate number is not available in the database
- Group – the field can be blank if license plate number is not available in the database
- Vehicle direction
    - Attempt of entry
    - Entry
    - Attempt of departure
    - Departure
    - Undetermined
- Technical information
    - Size of license plate number in pixel
    - Time of recognition
    - Accuracy of license plate recognition
    - Number of recognized license plates
    - Country indicator

- The country where license plate number is registered

II. Checkpoint (access control)
Basic functions:

Basic parameters for license plate transfer

- License plate number
- License plate description – the field can be blank if license plate number is not available in the database
- Group – the field can be blank if license plate number is not available in the database Vehicle direction
- Vehicle direction
  - In one stage
    - Entry
    - Departure
  - In two stages (with confirmation of entry and departure)
    - Attempt of entry
    - Entry
    - Attempt of departure
    - Departure
- Technical information
  - Size of license plate number in pixel
  - Time of recognition
  - Accuracy of license plate recognition
  - Number of recognized license plates
  - Country indicator
- The country where license plate number is registered
- Access permit
  - Allowed
  - Allowed by date
  - Allowed by duration of stay at the area
  - Allowed by number of entry records
- For more detailed information on checkpoint settings refer to paragraph.7.1.

III. Car wash
IV. Detection of vehicle entry/exit

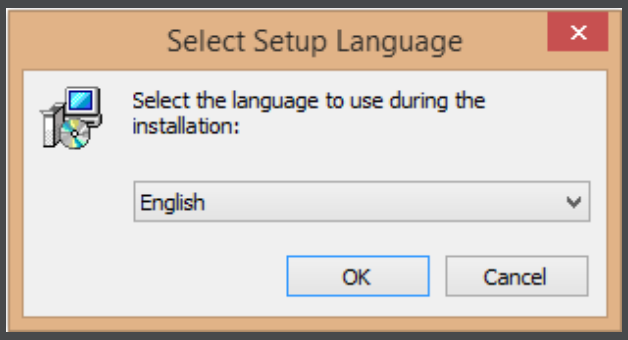
5.3 INSTALLATION AND REGISTRATION

5.3.1 USB KEY GUARDANT (WITHOUT INTERNET CONNECTION)

Installation requires Nomerok_xxx|-guardant which can be downloaded from the official web-site Number-ok.com. To make software work properly you should use the guardant dongle key.

1. Run the installer of NumberOK software numberok_xxx_guardant

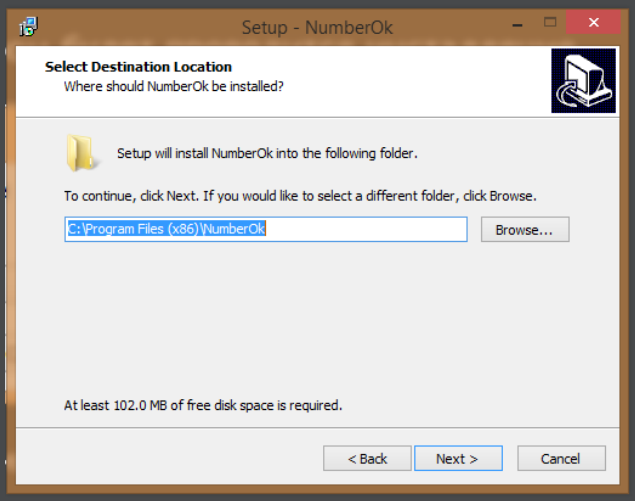
2. Select setup language



PICTURE 13. SOFTWARE INSTALLATION - GUARDANT KEY - SETUP LANGUAGE

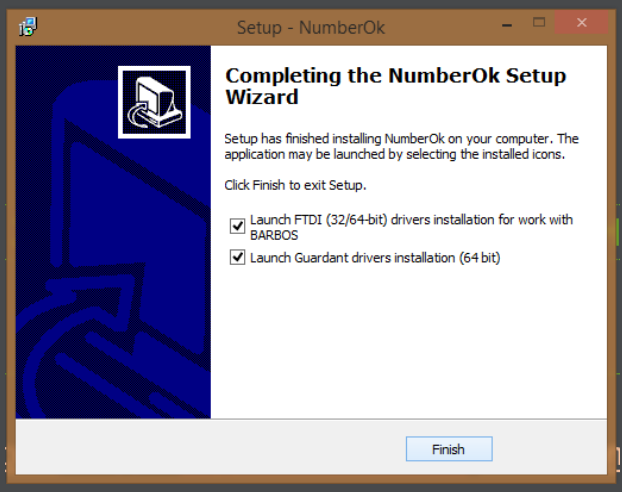
3. Follow the directions on the screen to complete installation

4. Select the disk for software installation and folders to store the screenshots and database.



PICTURE 14. SOFTWARE INSTALLATION - GUARDANT KEY - DISK OF INSTALLATION

5. It is required to check the field Launch Guardant drivers installation. If Barbos module is used please check the field Launch FTDI.

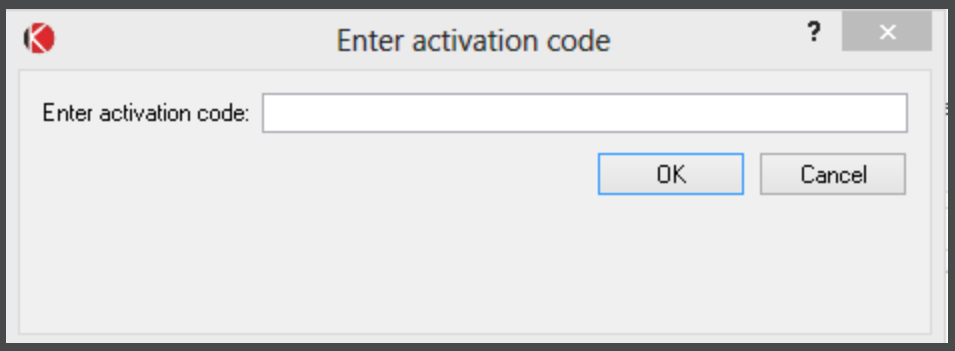


PICTURE 15. SOFTWARE INSTALLATION - GUARDANT KEY - DRIVER SETUP

6. After installing the software and drivers, plug guardant key into USB port.
If installation has been completed successfully LED indicator on the key will light up and remain illuminated. If the error has occurred when installing drivers, LED indicator will start flashing and software will not run. To remove this error refer to paragraph 7.2

7. Start NumberOK.

5.3.2 ELECTRONIC KEYS (WITH INTERNET CONNECTION)



PICTURE 16 DIALOG BOX FOR ACTIVATION KEY

After purchasing the license for NumberOK or downloading demo version of the software, the license activation key will be sent to you via e-mail. You can enter the license activation key in the dialog box shown in Picture 20 in order to activate the license package. When entering the activation key computer must be connected to the Internet. Installation of demo version requires permanent Internet connection. When full-featured version is used, computer must be connected to the Internet only during installation.

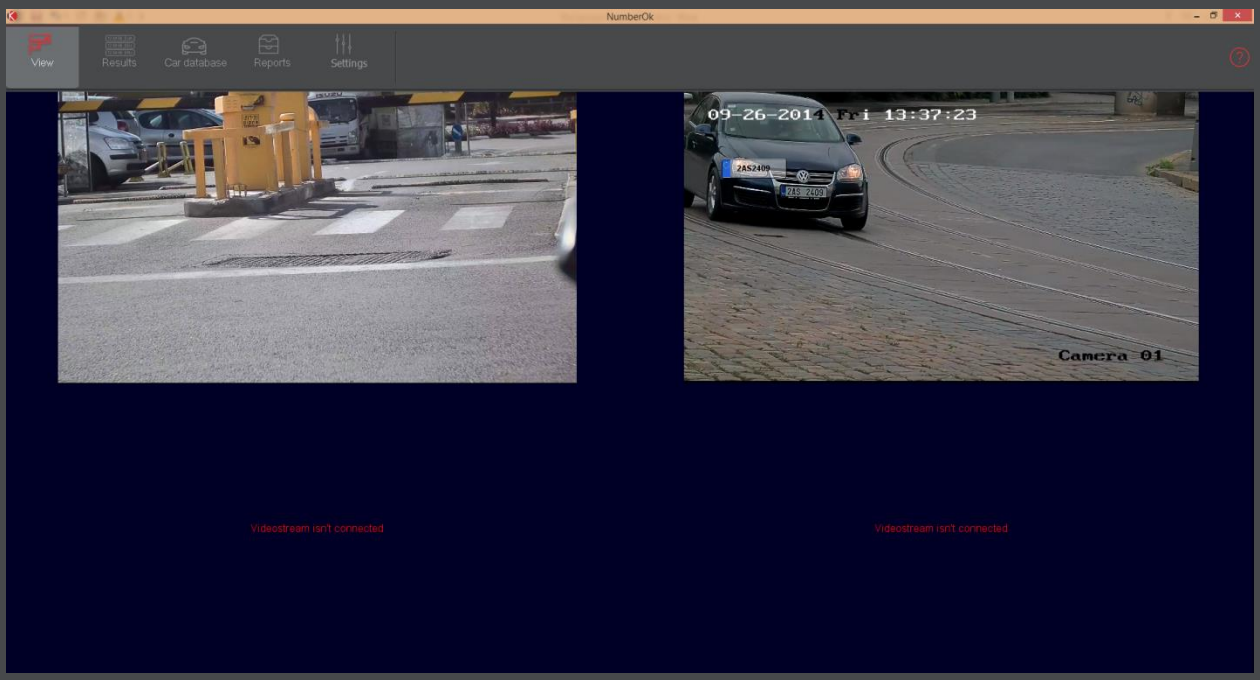
5.4 DATABASE

NumberOK supports two databases:

- **SQLIte** (basic) is intended to work with one user (one NumberOK software)
  - 200 000 entries
  - 20 000 car database

- **Firebird** (optional) is designed to work with multiple users. Multiple distributed NumberOK systems can use single data base. The data can be modified (for instance, database of allowed vehicles) and exported into external applications (for instance, data export to 1C)
  - 1 000 000 entries
  - 100 000 car database

5.5 LIVE VIEW TAB



PICTURE 17. LIVE VIEW TAB

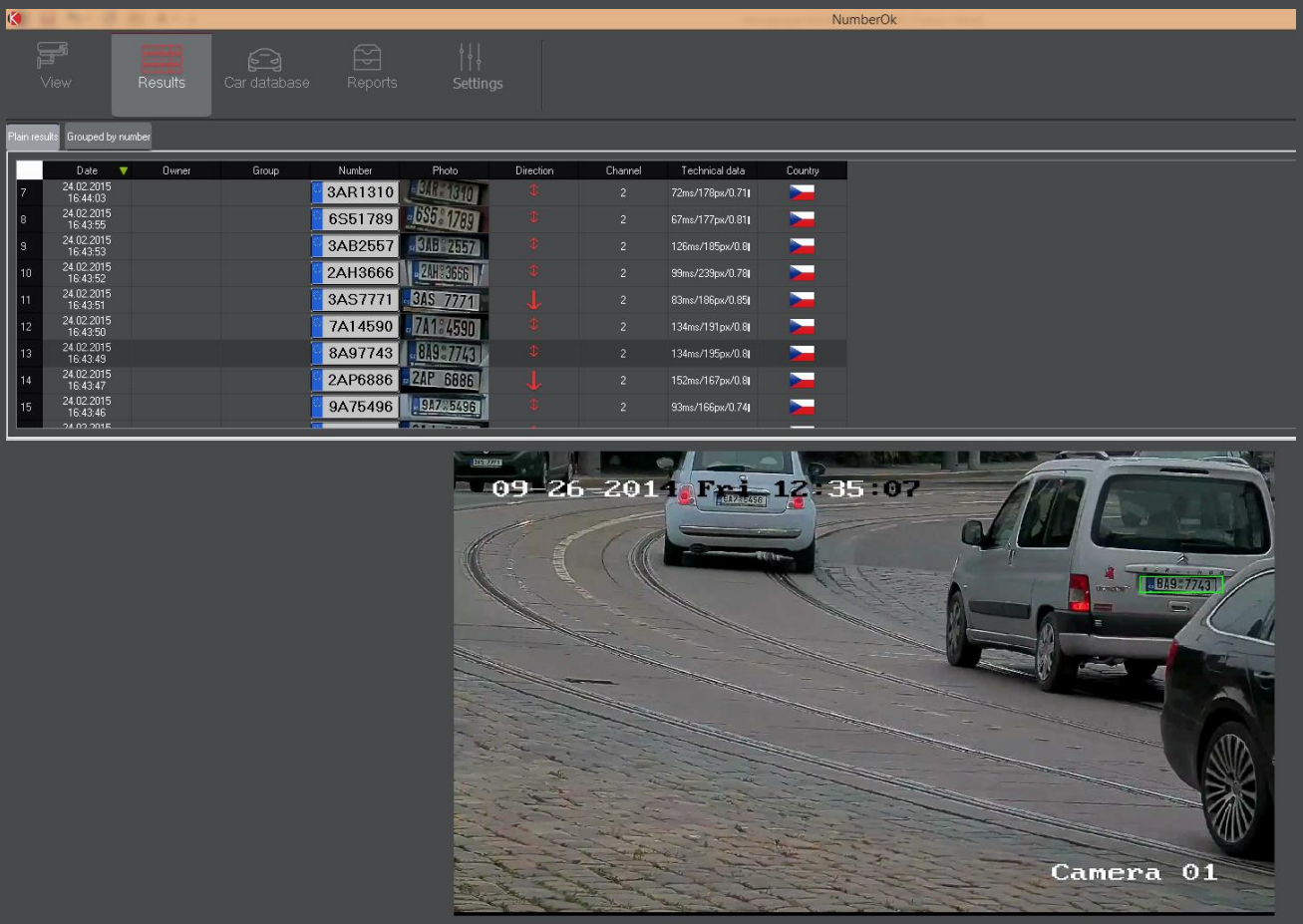
Live View tab shows real-time video stream from selected camera. In this tab can be displayed one, two or four video streams, depending on the type of purchased license.

- The section above the image of detected license plate Recognized license plate number with description of the number plate or without the description in case information on the number plate is not available in the database.
- Information on denied or allowed access to this vehicle. Available only in Checkpoint mode.
- Group to which the car belongs. If number plate cannot be found in any group, this field won't be displayed.
- Duration of stay at the area. Available only in Checkpoint mode when vehicle direction is detected as "Attempt of entry" or "Departure". Double-clicking on this field will take you to Results tab

PICTURE 18. PARAMETERS OF RECOGNIZED NUMBER PLATE

5.6 RESULTS TAB

5.6.1 SORTING BY DATE

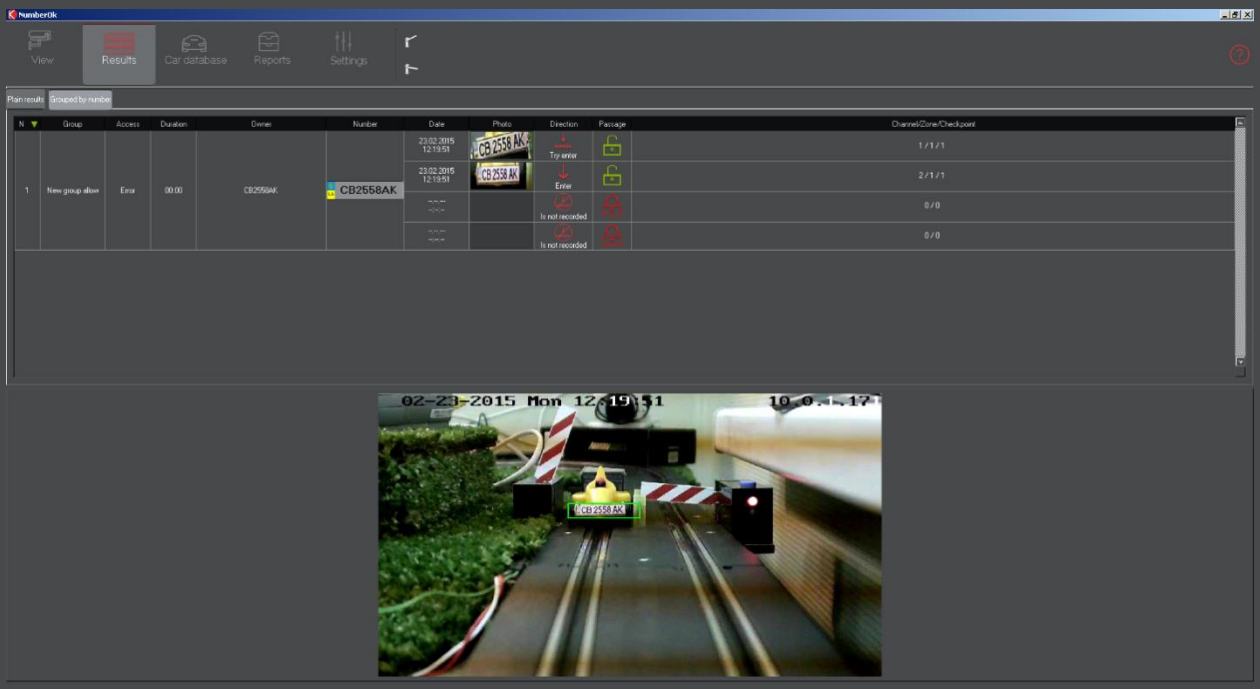


PICTURE 19 RESULTS TAB - SORTING BY DATE

This tab is available for all operation modes. The tab displays the last 100 recognition events and snapshots of vehicles.

Double-clicking on the number plate will save this number into the database and will take you to *Vehicle database - Cars* tab.

5.6.2 SORTING BY EVENTS



PICTURE 20 RESULTS TAB - SORTING BY EVENTS

This tab is available only for Checkpoint mode. Here all the events are grouped by entry/departure, by license plate number. Maximum number of recognition events is 100.

By double-clicking on the license plate number you can add it to the database and go to the *Vehicle database - cars* tab.

5.7 VEHICLE DATABASE TAB

5.7.1 CARS

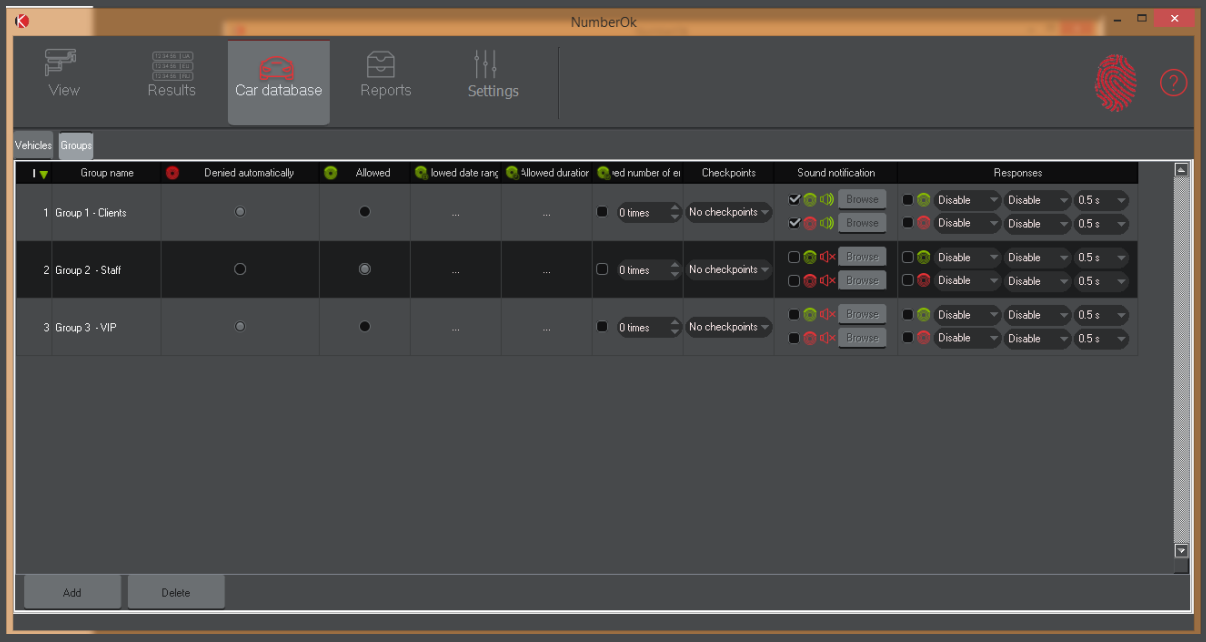
PICTURE 21 - VEHICLE DATABASE TAB - CARS

Under this tab user can enter license plate number and add the car to the preset group. It is essential to select country and enter the symbols of license plate number according to the template of that country. NumberOK verifies that the symbols are entered correctly and in case the number is incorrect the last will be highlighted in red.

The user can add and delete cars by means of corresponding buttons if he has the appropriate permissions.

The list of cars can be exported into Excel. The format of the document is illustrated in Appendix

5.7.2 GROUPS



PICTURE 22 VEHICLE DATABASE TAB - GROUPS

Under this tab the group of cars can be created, edited and deleted. Settings for access control can be configured in Checkpoint mode. The tab provides configuration of responses to recognition of specified license plate number.

The cars are divided into the following groups:

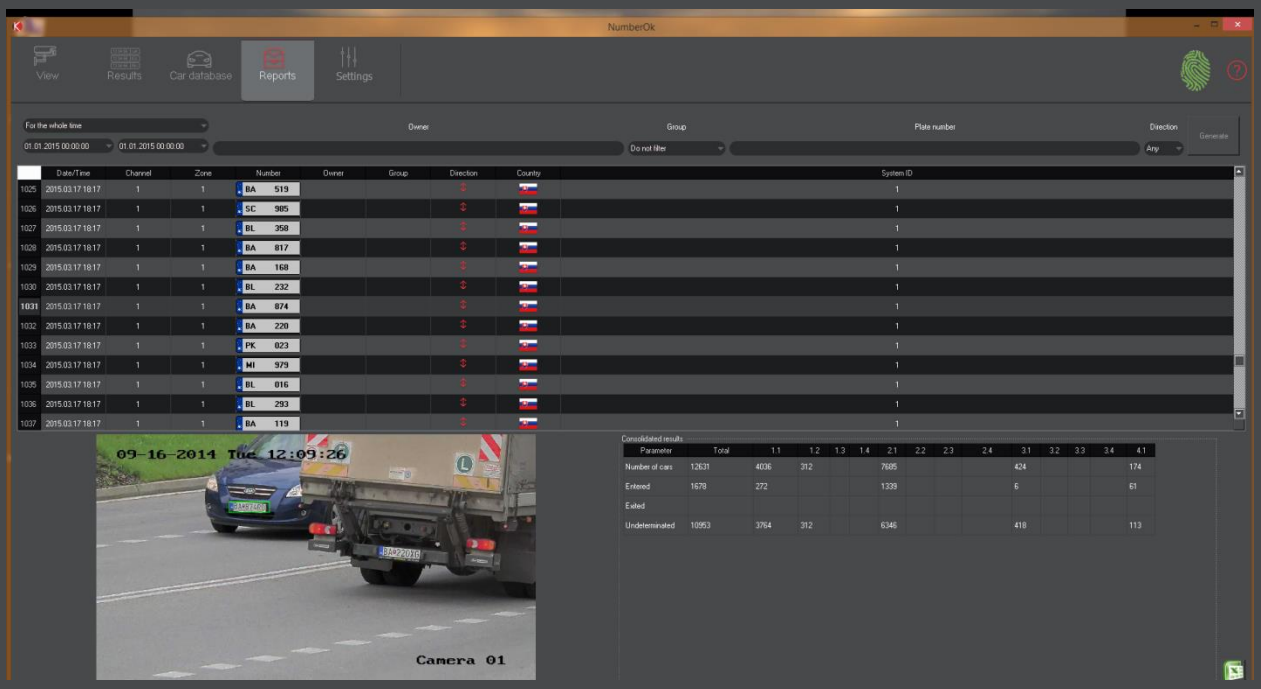
- The license plate numbers available in the database
- The license plate numbers not available in the database. Those license plate numbers can be divided into various groups.

TABLE 6 TYPES OF ACCESS CONTROL FOR VEHICLE GROUPS

Icon	Name	Description
	Allowed	For all vehicle groups the entry is always allowed
	Denied	For all vehicle groups the entry is always forbidden
	Restricted by time	For all vehicle groups the entry is allowed within specified time frame
	Restricted by number of entries	For all vehicle groups the entry is allowed in case the number of entries is not exceeded (counting by every group)
	Restricted by duration of stay at the area	For all vehicle groups the entry is allowed in case the duration of their stay is not exceeded

Name	Description
Audio buzz	When the license plate number of this group is recognized the playback of selected file will be generated
Response by control module	When the license plate number of this group is recognized selected relay of control module will be opened/closed

5.8 REPORTS TAB



PICTURE 235 REPORTS TAB

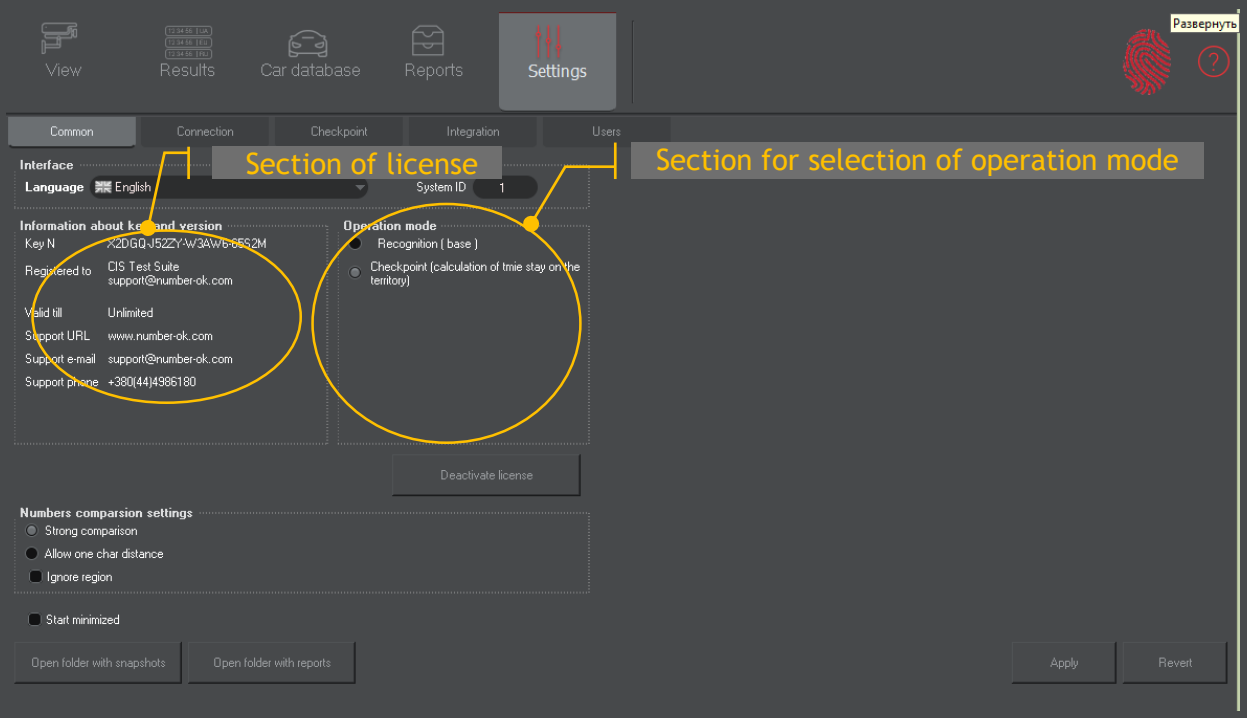
Under this tab the user can generate reports of two types:

- General reports – all recognition events, generated by the following filters:
  - By time period
  - By group
  - By checkpoint (for checkpoint mode)
  - By channels and recognition zones
  - By license plate number or part of license plate number
  - By direction of travel of the vehicle
  - By description
- Consolidated reports
  - Recognition mode. Total amount of passed cars, grouped by channels/areas and direction of travel
  - Checkpoint mode. Total amount of passed cars, grouped by direction of travel and type of access.

All general reports can be exported into Excel format.

5.9 SETTINGS TAB

5.9.1 GENERAL SETTINGS



PICTURE 24 SETTINGS-GENERAL

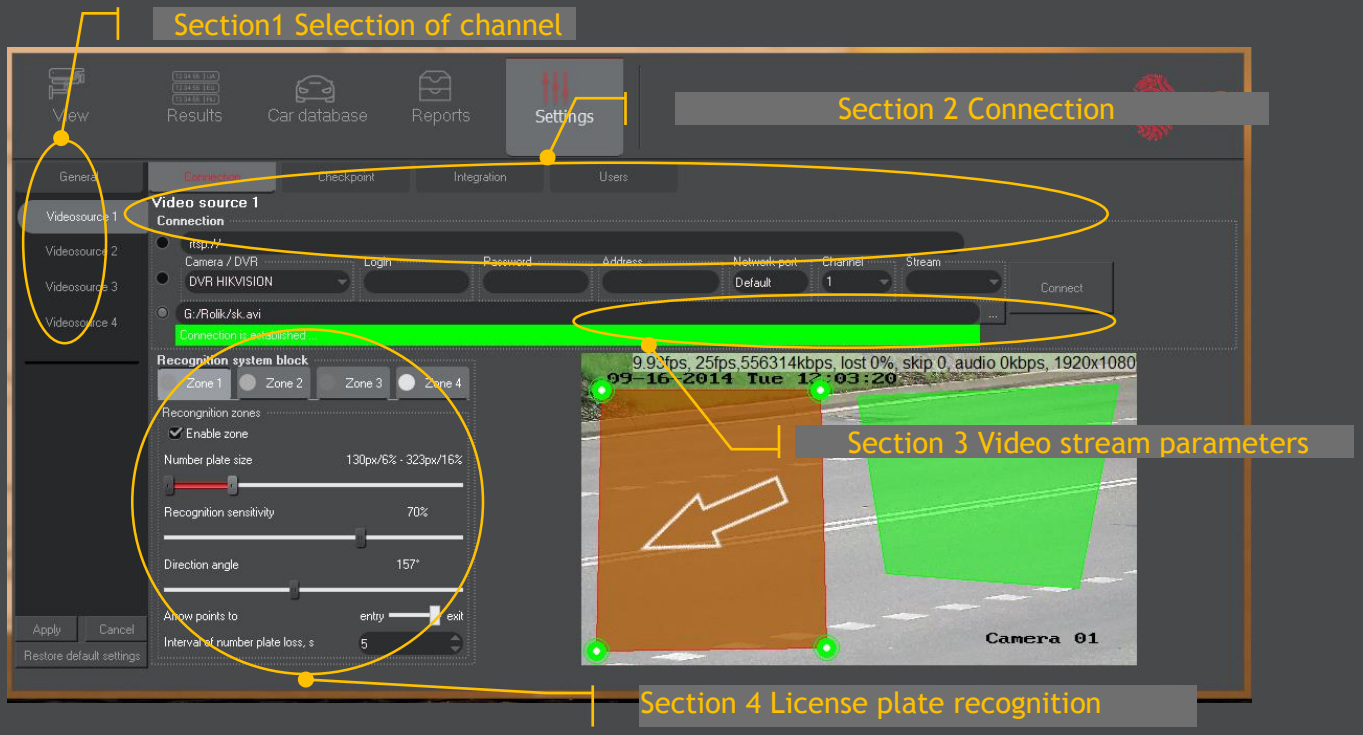
This tab is divided into three main sections:

1. License section, where all available data on license are displayed
    a. License No
    b. Customer
    c. Information on technical support
    d. Deactivate license
    e. Buy full-featured version of the product (when demo version is used)

2. Operation mode section(see paragraph 5.2):
    - Recognition mode (basic)
    - Checkpoint mode

3. General settings
    a. Customization of comparison with reference data base
    b. Templates/countries for license plate recognition
    c. Work in в трее.

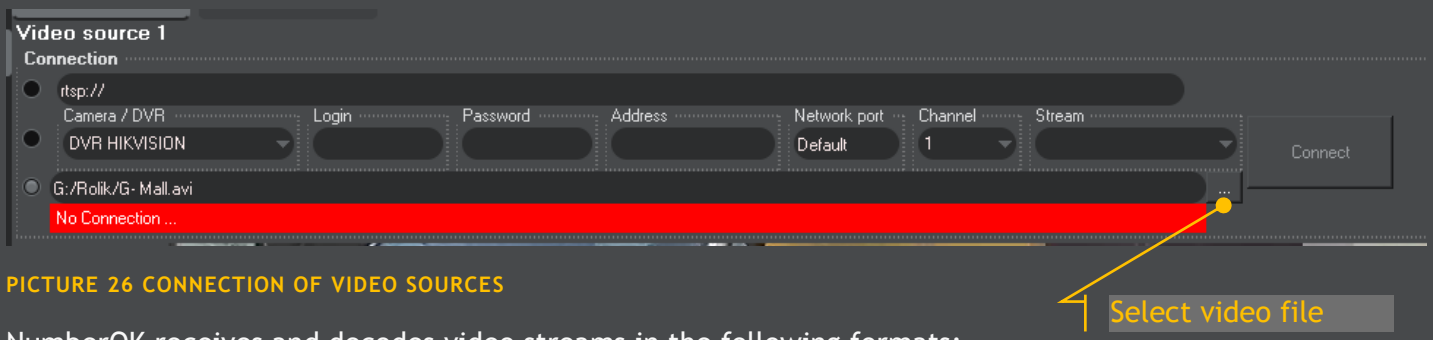
5.9.2 CONNECTION



PICTURE 25 CONNECTION

- Section1 – Selection of video channel
  - Depending on the type of purchased license one, two or four channels can be available for the user. Video channel is taken to mean the camera (file) which can be connected to NumberOk.
- Section 2 – Connection of video sources (see paragraph 5.3.3 connection of video sources)
- Section 3 – Video stream parameters
  - F_{recogn} – Real-time FPS (frames per second) transferred to recognition module. This is a theoretical parameter which depends on computer's CPU and ideally responds to $F_{video} = F_{recogn}$
  - F_{video} – Number of frames per second delivered by video source.
  - Lost – rate of data loss from video sources.
  - Skip – rate of missed (not captured) frames. When computer's CPU is insufficient the frames can be skipped (video signal will be unstable)
  - Resolution – resolution of video source.
- Section 4 –License plate recognition block (see paragraph 5.3.4 Recognition block set up)

5.9.3 CONNECTION OF VIDEO SOURCES



PICTURE 26 CONNECTION OF VIDEO SOURCES

NumberOK receives and decodes video streams in the following formats:

- Live 555 library
  - MPEG_DECODER_TYPE_H264
  - MPEG_DECODER_TYPE_MPEG4
  - MPEG_DECODER_TYPE_JPEG
  - MPEG_DECODER_TYPE_MXPEG
- FFMPEG library
  - AV_CODEC_ID_H264
  - AV_CODEC_ID_MPEG4
  - AV_CODEC_ID_HEVC

NumberOK provides 3 types of connection:

1. Uniform connection via RTSP protocol (refer to example). About 99% IP cameras and most of DVRs support this protocol.
   For example, we provide connection to Hikvision IP camera:
   Rtsp://admin:12345@IP_address:554/, where

   Admin – username
   12345 – user password
   IP_address – camera address
   554 – RTSP port

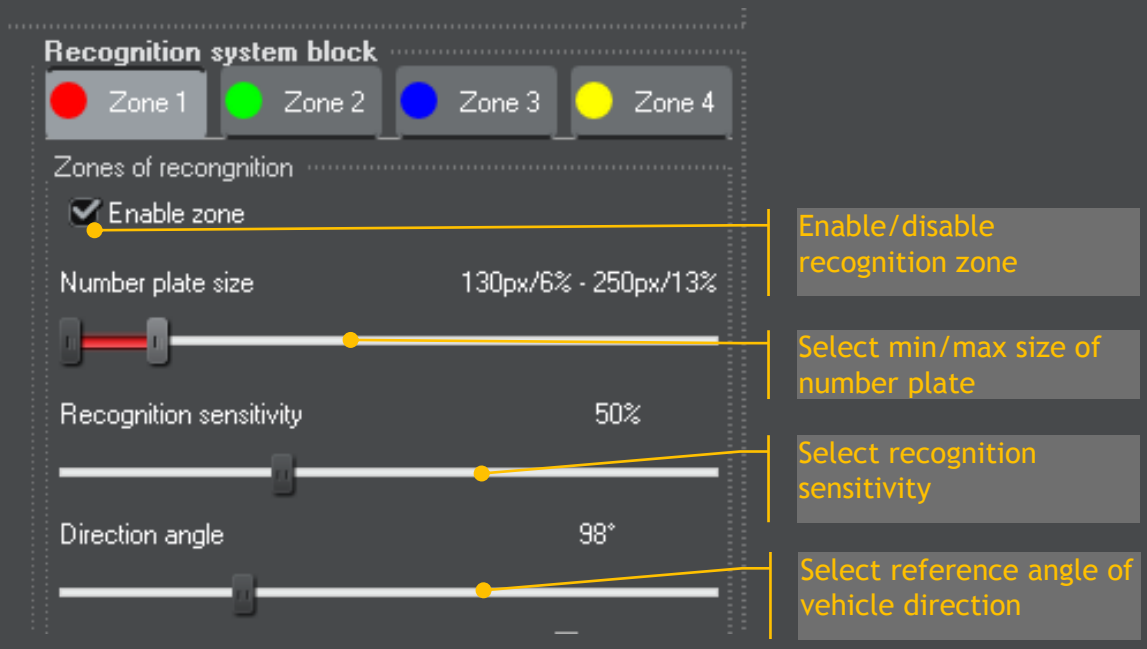
2. CAMERA/DVR – these video sources are connected via SDK, provided by manufacturers of these cameras or DVRs. NumberOK supports the following brands:
   - DVR Hikvision
   - DVR Dahua
   - DVR NUVUS B series
   - DVR LINIA
   - DVR TVT
   - DVR Praxis
   - DVR NUUO
   - DVR Partizan

3. Text fields

4.
    - Login – username
    - Password – user password
    - Address – IP address
    - Network port – port
    - Channel –number of channel (camera)
    - Stream – number of video stream (main/ secondary,1/2/3 …)
5. Video files

In order to quickly start application and test its functionality, user may watch a demo-video, which is available at C:\Program Files (x86) \FF\nomerOK\test_video_eu

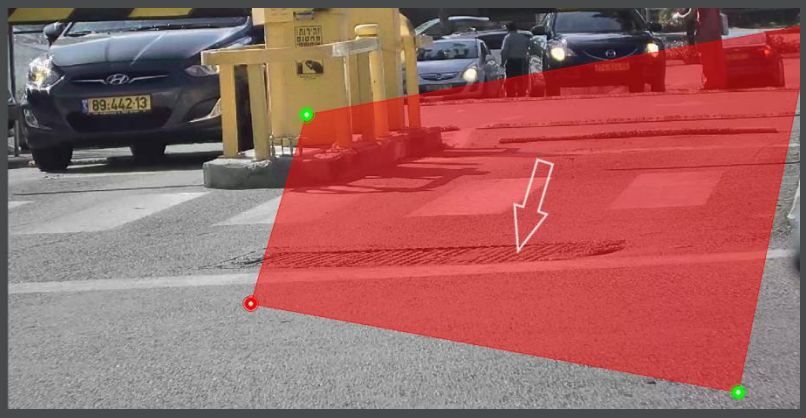
5.9.3.1 RECOGNITION BLOCK SET UP



PICTURE 276 SETTINGS OF RECOGNITION BLOCK

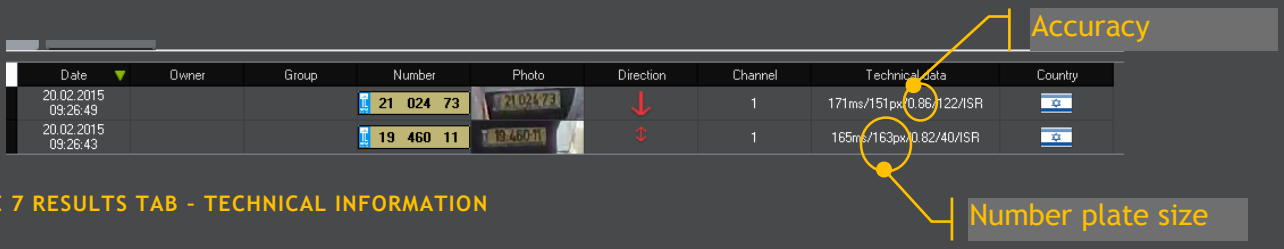
The following settings are available for the user:

1. **Recognition zone** is the area where license plate numbers will be detected. The user can draw the area for recognition zone and move its boundaries using the mouse. You can configure up to 4 recognition zones which can be enabled or disabled on request. The first zone is enabled by default and occupies 90% of the frame.



PICTURE 28 ADJUSTMENT OF RECOGNITION ZONE

2. **Number plate size** – user may choose minimum and maximum size of license plate number in pixels. NumberOK will search the number plates in that specified size range. Minimum size of number plate is 130 pixels, maximum – 1000 pixels. Default size is 130-300 pixels. The size of number plate can be determined by measuring screenshot from the camera. You can also refer to technical information on recognized numbers in the Results tab using wide range, and after that set more narrow range.



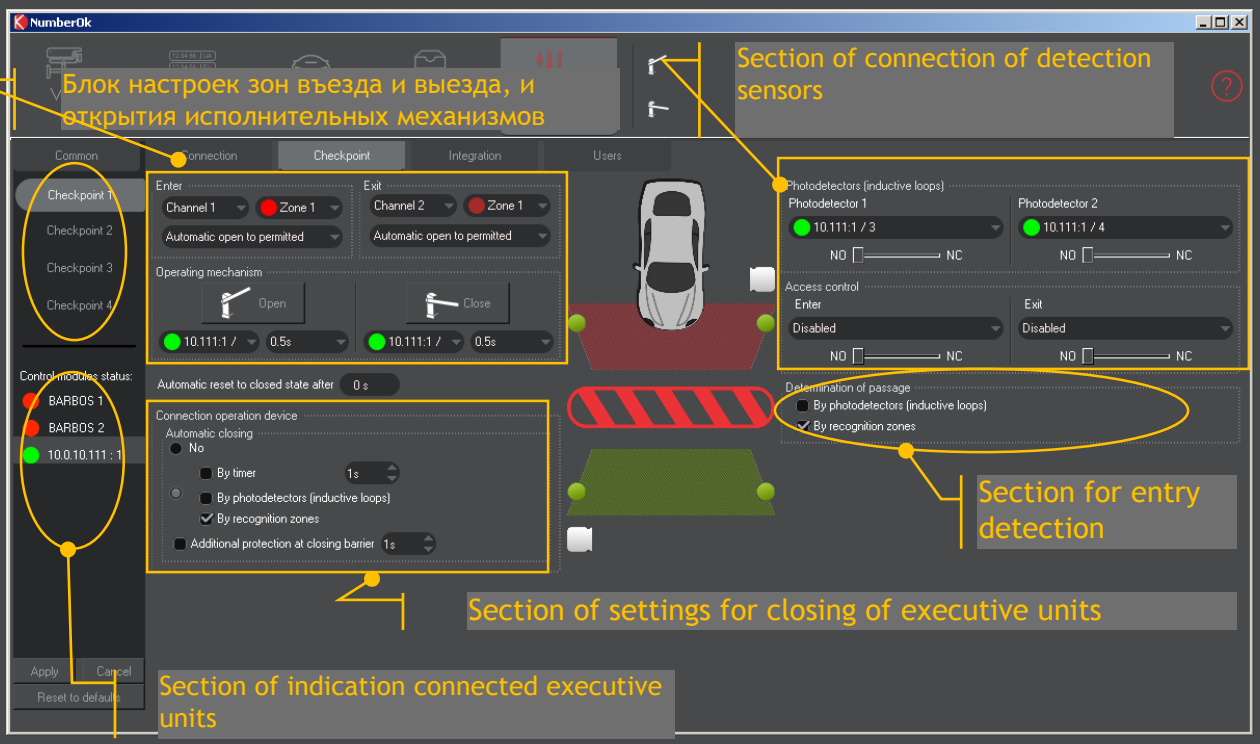
PICTURE 7 RESULTS TAB - TECHNICAL INFORMATION

3. Recognition sensitivity (recognition accuracy). Every license plate has its own recognition accuracy rate. The higher image quality is the higher is recognition accuracy. Accordingly, false number plates have lower accuracy rate. To eliminate false number plate recognition it is essential to select recognition sensitivity that enables detection of more real number plates. The value of recognition accuracy depends on the quality of video stream. By default recognition sensitivity is set to 70% which is applicable for most cases. The real values of recognized number plates are available in the section of technical information.

4. Reference angle of vehicle direction. Proper operation of algorithm for detection of vehicle direction by movement of number plate in the frame requires a thorough understanding of vehicle travelling directions. Said parameter determines reference travelling direction. The value of this parameter is within 0 -359°, in 1-degree increments. Use slider bar or scroll wheel of your mouse to adjust the value. As you can see from Picture 25 the arrow points at Entry. If the vehicle direction is fully in line with the direction of arrow, the system will detect *Entry*. If travelling direction is opposite to the direction of arrow the system will detect *Departure*.

PICTURE 29 REFERENCE VEHICLE DIRECTION

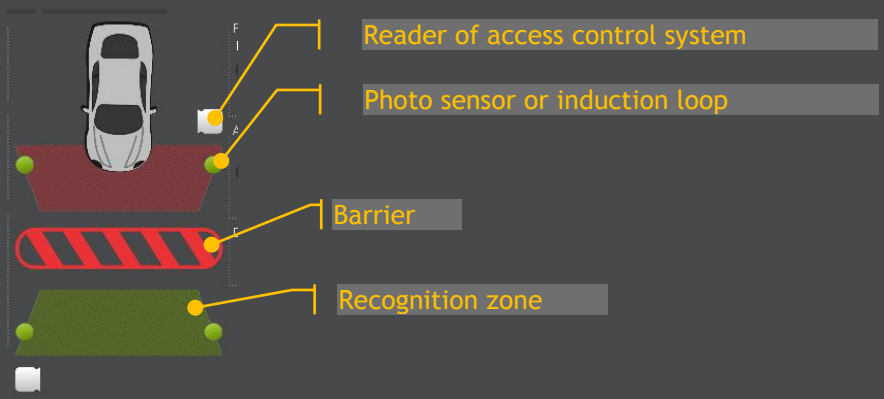
5.9.4 CHECKPOINT



PICTURE 30 CHECKPOINT TAB

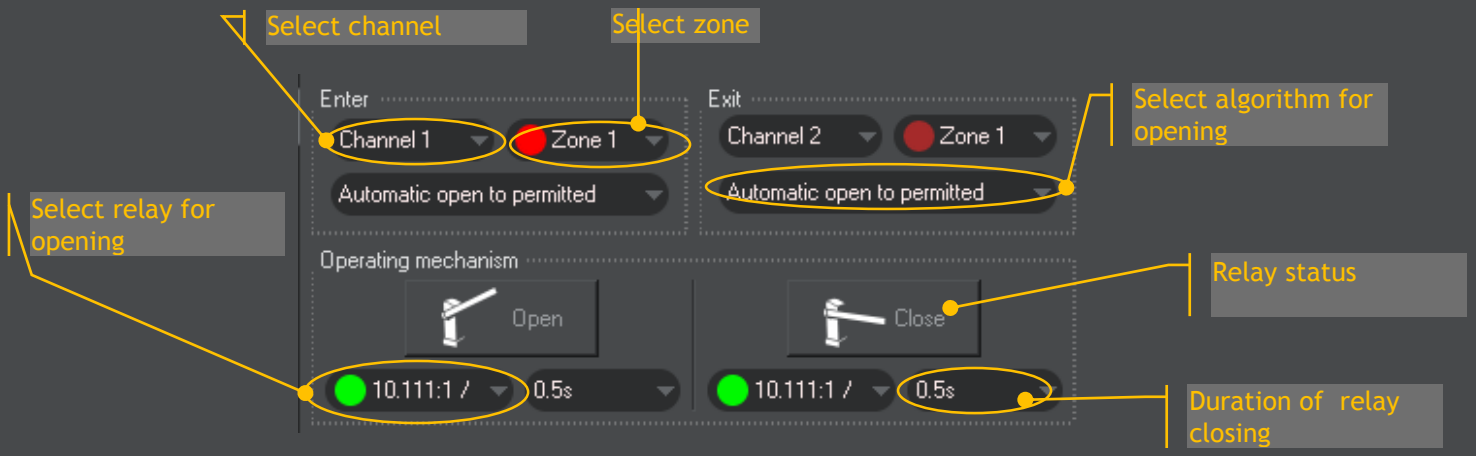
Checkpoint tab is available only in checkpoint mode. The tab is intended for configuration of checkpoint and includes the following settings:

- Configuration of entry/departure areas
- Connection of executive units
- Configuration of algorithms for opening/closing of executive units
- Connection of algorithms for detection of entry



PICTURE 31 SCHEMATIC DIAGRAM OF CHECKPOINT

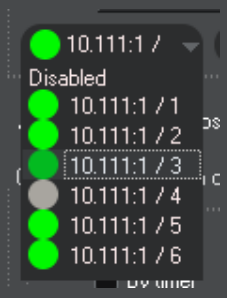
5.9.4.1 CONFIGURATION OF ENTRY/DEPARTURE ZONES AND OPENING/CLOSING OF EXECUTIVE UNITS



PICTURE 32 SECTION FOR CONFIGURATION OF ENTRY/DEPARTURE AREAS AND OPENING EXECUTIVE UNITS

- Assign recognition zones for entry and/or departure areas. Firstly, you should select video channel preset for that zone, than select the zone itself. Selected zone will be unavailable in other settings. Only enabled zones can be selected. For two-directional checkpoints two recognition zones are required (for entrance and exit). In this case detection of entry by adjacent recognition zones will be activated. For one-directional checkpoints only one recognition zone is required.

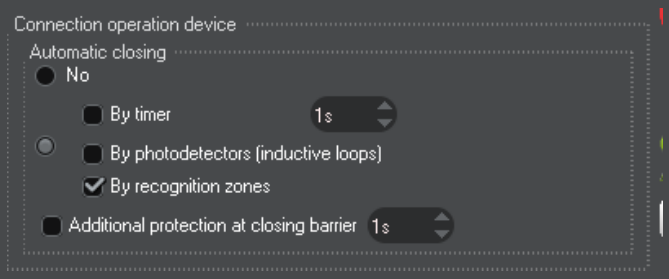
  When recognition zone is selected it will appears highlighted in the specified color under checkpoint tab.



PICTURE 33 RELAY SELECTION

- Connect output relays of control module, responsible for opening and closing (see diagram of connection of executive units) and configure duration of closing.
- Select algorithm for opening:
  - Do not open automatically
  - Open for all
  - Open for the vehicles from the white list

5.9.4.2 CONFIGURATION OF CLOSING OF EXECUTIVE UNITS



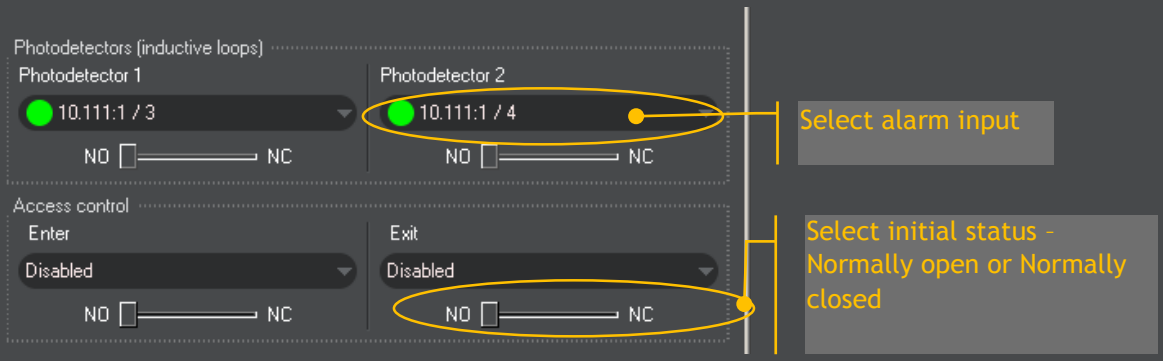
PICTURE 34 CONFIGURATION OF BARRIER CLOSING

The following algorithms of closing are available:

- Do not close automatically
- Close by timer. Range of values – 5-60 s
- Close by detection sensors – closing after detector is triggered in the right direction
- Closing by recognition zones

The system provides algorithm for safe closing. Closing of barrier will be cancelled (when command for opening is issued) in case at least one detection sensor is triggered at the moment of closing.

5.9.4.3 CONNECTION OF DETECTION SENSORS

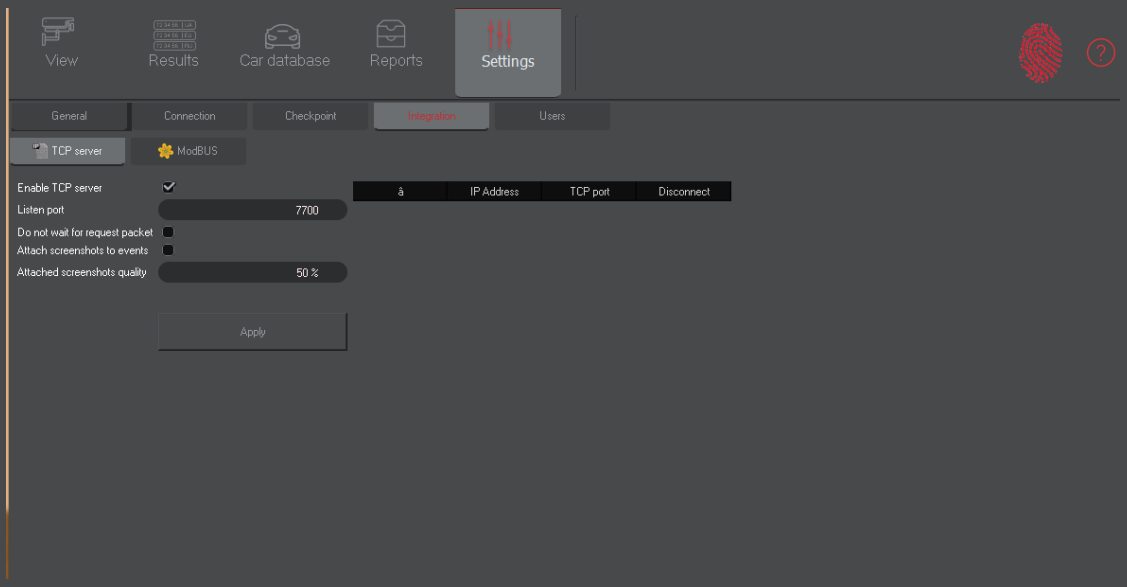


PICTURE 35 CONNECTION OF DETECTION SENSORS

- Several pairs of detectors are available for connection. Detectors provide the detection of entry and ensure closing of the barrier after the vehicle passed through. Photo sensors or induction loops can serve as the detection sensors. When the vehicle crosses photo sensor or stands within induction loop, the event will be highlighted in red on the screen. In case there is no any vehicle – the event will be highlighted in green. When algorithm for detection of vehicle direction by detectors is triggered (the car passed through), two indicators on detectors will be illuminated in the direction the vehicle movement.
- It is also possible to connect alarm events from controller of access system via contacts in order to receive confirmation that the vehicle can access territory by card. If this alarm input is triggered, the event will be highlighted in green on the screen. See connection diagram for executive units

5.9.5 INTEGRATION

5.9.5.1 TCP SERVER

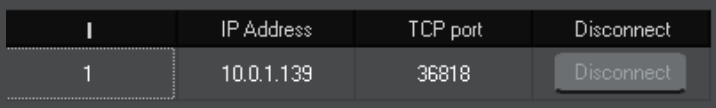


PICTURE 36 SETTINGS RAB – INTEGRATION

Under this tab integration with external applications via TCP protocol can be performed. NumberOK acts as server, to which external applications can be connected, and sends all recognition events at the selected port. NumberOK supports up to 4 connections. The systems may have preliminary connection (after connection is established preliminary «request-response» is sent from the systems and only after that event is transferred) or not (the recognition events are sent immediately after the connection is established).

Apart from text messages, you can send snapshots of the vehicle in selected quality.

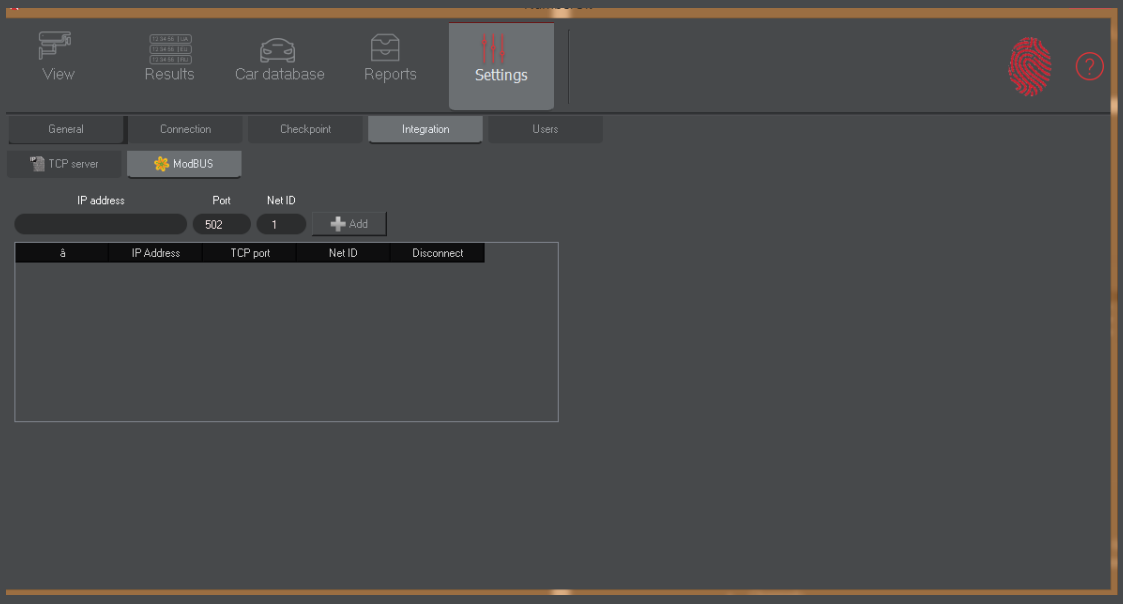
For detailed information on event transfer protocol and typical examples refer to appendix.



PICTURE 37 SETTINGS TAB-INTEGRATION-APPLICATION CONNECTION

Here IP address and port of connected application is displayed. You can disable this connection.

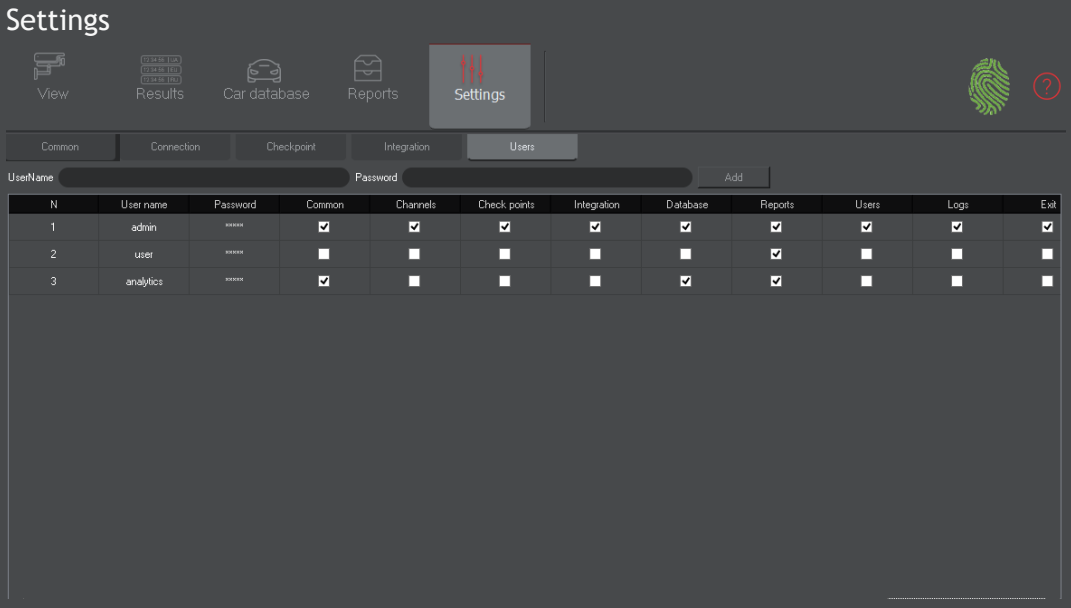
5.9.5.2 CONNECTION OF NETWORK CONTROL MODULE ICP CPN VIA MODBUS PROTOCOL



PICTURE 38 INTEGRATION- MODBUS SETTINGS

In order to connect control module enter its IP address and port in the corresponding text fields and press *Add* button. In case connection is completed successfully the unit will be added to the table. The system provides connection of up to 3 units.

5.9.6 USERS



PICTURE 39 SETTINGS TAB - USERS

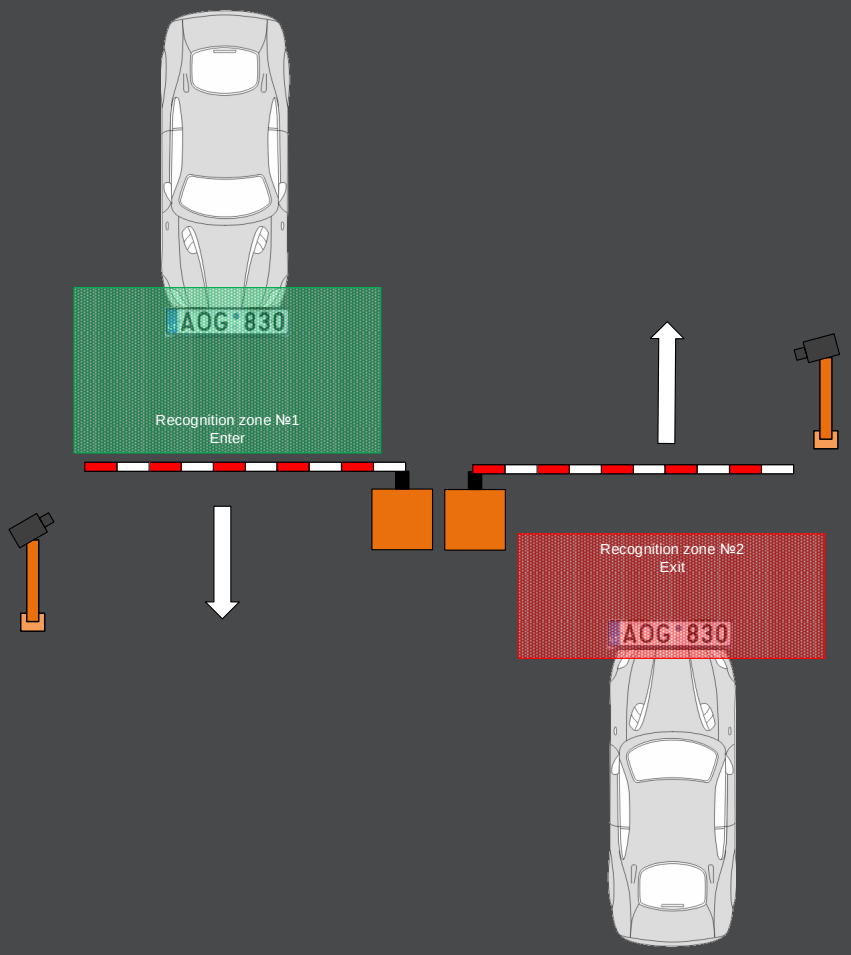
This tab is intended for management of user access rights for the functions of the system. Access restrictions can be applied to the following functions:

- General
- Channels
- Checkpoints
- Integration
- Database
- Reports
- Users
- Logs
- Password

6 SYMBOLS, TERMS AND ALGORITHMS

6.1 TYPES OF CHECKPOINTS

6.1.1 ONE-DIRECTIONAL CHECKPOINTS WITHOUT CONFIRMATION OF ENTRY



PICTURE 40 TWO ONE-DIRECTIONAL CHECKPOINTS WITHOUT CONFIRMATION OF ENTRY

Checkpoint is considered one-directional when vehicles move in one direction (enter or exit), therefore such checkpoint requires one recognition zone to control entrance and exit.

Algorithm for calculation of vehicle direction is used to determine travelling direction of the vehicle by movement of license plate number in the frame. In case algorithm cannot determine vehicle direction the last will be configured by recognition zone.

Available values::

Entry ↓ ↑ Departure

Barrier opens when the license plate is recognized, and closes by timer or manually. It is impossible to monitor violators (in case vehicle from black list passed through).

Available types of entry:

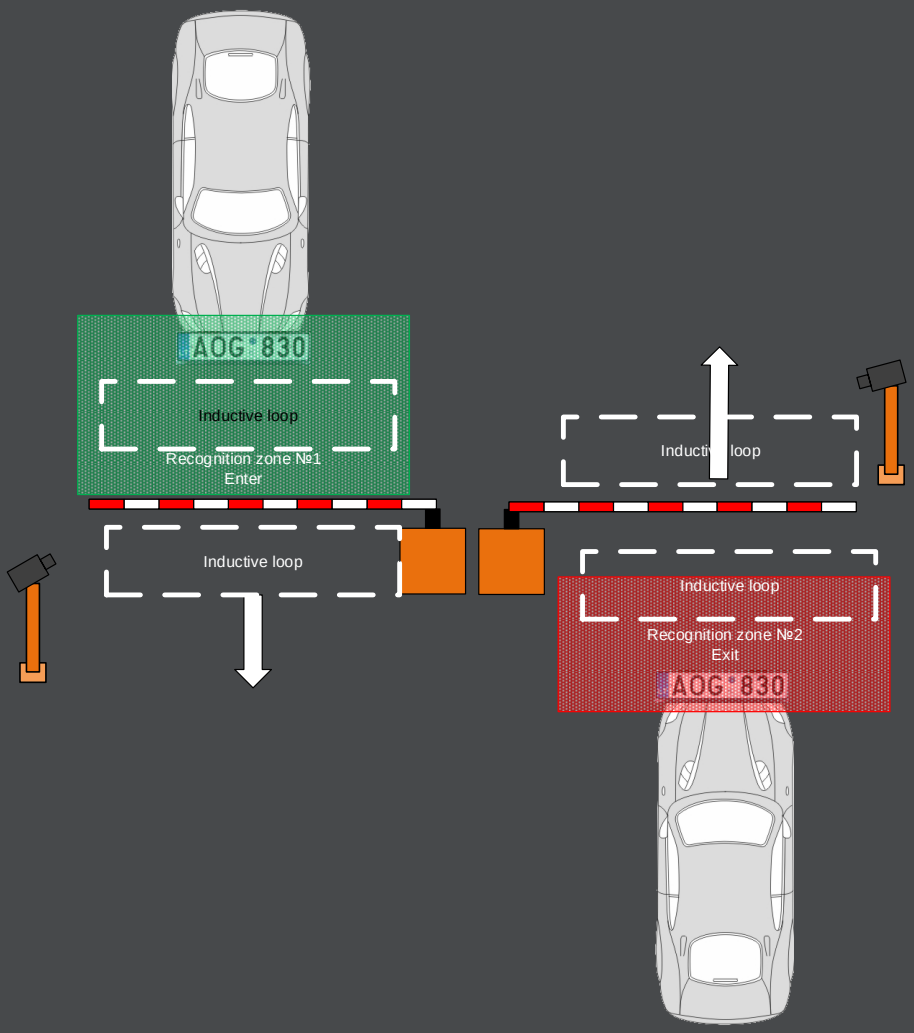
Acces automatically denied

Acces automatically allowed

Opened by user

Opened by card

6.1.2 ONE-DIRECTIONAL CHECKPOINTS WITH CONFIRMATION OF ENTRY



PICTURE 41 TWO ONE-DIRECTIONAL CHECKPOINTS WITH CONFIRMATION OF ENTRY

Checkpoint is considered one-directional when vehicles move in one direction (enter or exit), therefore such checkpoint requires one recognition zone to control entrance and exit.

To confirm entry you must connect alarm sensors (photo sensors and induction loops).

For detection of vehicle direction all algorithms are used: algorithm for calculation of vehicle direction by movement of license plate in the frame, by adjacent recognition zones and by triggering of alarm sensors.

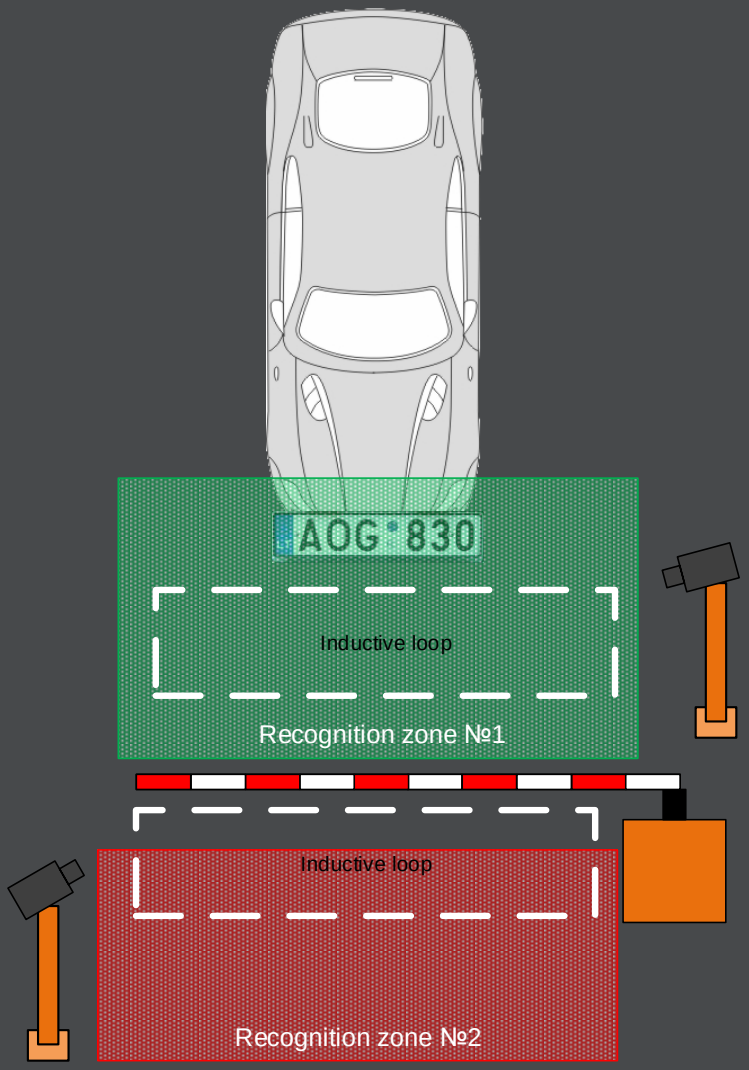
TABLE 74 ONE-DIRECTIONAL CHECKPOINTS -POSSIBLE OPTIOINS OF VEHICLE DIRECTION

Icon	Name	Description
	Attempt of entry	The event occurs when the vehicle is detected in the recognition zone of Entrance and algorithm for detection of vehicle direction by movement of license plate in the frame generates status *Entry* or *Undetermined*.
	Entry	The event occurs in the following cases: 1. Standard situation: the vehicle passes through the checkpoint in the right direction when related event *Attempt of entry* occurs and alarm sensors at the entrance are triggered. Alarm sensors must be triggered within the time frame preset for *Time of number plate loss*. 2. Nonstandard situation: the vehicle passes through the second checkpoint configured only for departure. License plate is recognized in the recognition zone of *Exit* and algorithm for detection of vehicle direction by movement of license plate generates *Departure* status.
	Attempt of departure	The event occurs when the vehicle enters recognition zone of *Exit* and algorithm for detection of vehicle direction by movement of license plate generates status *Departure* or Not *Undetermined*. This is a standard situation.
	Departure	The event occurs in the following cases: 1. Standard situation: the vehicle passes through the checkpoint in the right direction when related event *Attempt of departure* occurs and alarm sensors in the direction of exit are triggered. Alarm sensors must be triggered within the time frame preset for *Time of number plate loss* 2. Nonstandard situation: the vehicle passes through the second checkpoint configured only for entry. License plate is recognized in the recognition zone of *Entrance* and algorithm for detection of vehicle direction generates *Departure* status.
	Attempt of entry is not detected	The event occurs only when *Entry, Departure and Attempt of departure* were detected, but *Attempt for entry* wasn't captured. That can happen only if the recognition error occurred in the *Entrance* zone or false sensor alarm was triggered.
	Attempt of departure is not detected	The event occurs only in case *Entry* or *Re-Entry* events were detected, but *Attempt for departure* was not captured. That can happen if recognition failure occurred in the *Exit* zone.
	Entry is not detected	The event occurs only in case *Departure* or *Re-Entry* is detected, but *Entry* wasn't captured. That can happen if recognition failure occurred or false sensor alarm was triggered.
	Departure is not detected	The event occurs when only *Re-Entry* is detected. That can happen if recognition failure occurred or false sensor alarm was triggered.

Table 8 ONE-DIRECTIONAL CHECKPOINTS - POSSIBLE OPTIONS OF TRAVELLING DIRECTION

Icon	Name	Description
	Opened automatically	The command to open executive mechanisms (barriers or gates) is issued automatically.
	Closed	The command to open executive mechanisms (barriers or gates) is not issued.
	Opened on user's command	The command to open executive mechanisms (barriers or gates) is issued via control button pressed by user.
	Opened by card	The command to open executive mechanisms (barriers or gates) is issued after it is received at alarm input of control module
	Successful entry	The event occurs only if entry with confirmation is configured. Vehicle with allowed access has successfully entered the area.
	Violator	The event occurs only when entry with confirmation is configured. Vehicle with denied access enters the area or nonstandard situation occurs (the vehicle passes through the wrong checkpoint).

6.1.3 TWO-DIRECTIONAL CHECKPOINTS WITH CONFIRMATION OF ENTRY



PICTURE 42 TWO-DIRECTIONAL CHECKPOINTS WITH CONFIRMATION OF ENTRY

The checkpoint through which the vehicles can enter and leave out is considered two-directional; therefore such checkpoint requires two recognition zones to control entrance and exit.

To confirm entry you must connect alarm sensors (photo sensors and induction loops) or use adjacent recognition zones.

For detection of vehicle direction all algorithms are used: algorithm for calculation of vehicle direction by movement of license plate in the frame, by adjacent recognition zones and by triggering of alarm sensors.

TABLE 95 TWO-DIRECTIONAL CHECKPOINTS –POSSIBLE OPTIONS OF TRAVELLING DIRECTION

Icon	Name	Description
	Attempt of entry	The event occurs when the vehicle is detected in the recognition zone of *Entrance* and algorithm for detection of vehicle direction by movement of license plate in the frame generates status *Entry* or *Undetermined*. Standard event.
	Entry	The event occurs when the vehicle is captured in the recognition zone of *Entrance* and algorithm for detection of vehicle direction by movement of license plate in the frame generates status *Entry* or *Undetermined*; or alarm sensors are triggered at the *Entrance*.
	Attempt of departure	The event occurs when the vehicle is captured in the recognition zone of *Exit* and algorithm for detection of vehicle direction by movement of license plate in the frame generates status *Departure* or *Undetermined*.
	Departure	The event occurs when the vehicle is captured in the recognition zone of *Exit* and algorithm for detection of vehicle direction by movement of license plate generates status *Departure* or *Undetermined* or alarm sensors are triggered at the Exit.
	Attempt of entry is not detected	The event occurs when only *Entry, Departure and Attempt of departure* were detected, but *Attempt of entry* wasn't captured. That can only happen if the recognition error occurred in the *Entrance* zone or in case false sensor alarm was triggered.
	Attempt of departure is not detected	The event occurs only in case *Entry* or *Re-Entry* events were detected, but *Attempt of departure* was not captured. That can happen if recognition failure occurred in the *Exit* area.
	Entry is not detected	The event occurs only in case *Departure* or *Re-Entry* is detected, but Entry was not captured. That can happen if recognition failure occurred or false sensor alarm was triggered.
	Departure is not detected	The event occurs when *Re-Entry* is detected, but *Departure* was not captured. That can happen if recognition failure occurred or false sensor alarm was triggered.

TABLE 10 TWO-DIRECTIONAL CHECKPOINTS - POSSIBLE ACCESS OPTIONS

Icon	Name	Description
	Opened automatically	The command to open executive mechanisms (barriers or gates) is issued automatically.
	Closed	The command to open executive mechanisms (barriers or gates) is not issued.
	Opened on user's command	The command to open executive mechanisms (barriers or gates) is issued via control button pressed by user.
	Opened by card	The command to open executive mechanisms (barriers or gates) is issued after it is received at alarm input of control module
	Successful entry	The event occurs only if entry with confirmation is configured. Vehicle with allowed access has successfully entered the area.
	Violator	The event occurs only when entry with confirmation is configured. Vehicle with denied access enters the area or nonstandard situation occurs (the vehicle passes through the wrong checkpoint).

6.2 ALGORITHMS

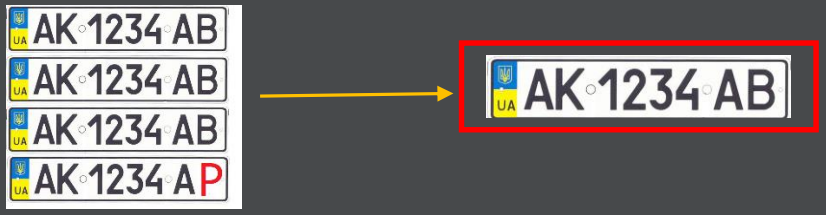
6.2.1 ALGORITHM FOR LICENSE PLATE RECOGNITION

- Connection to video source
- Decoding video stream frame by frame
- Detection of license plate numbers

- Image adjustment

- Segmentation

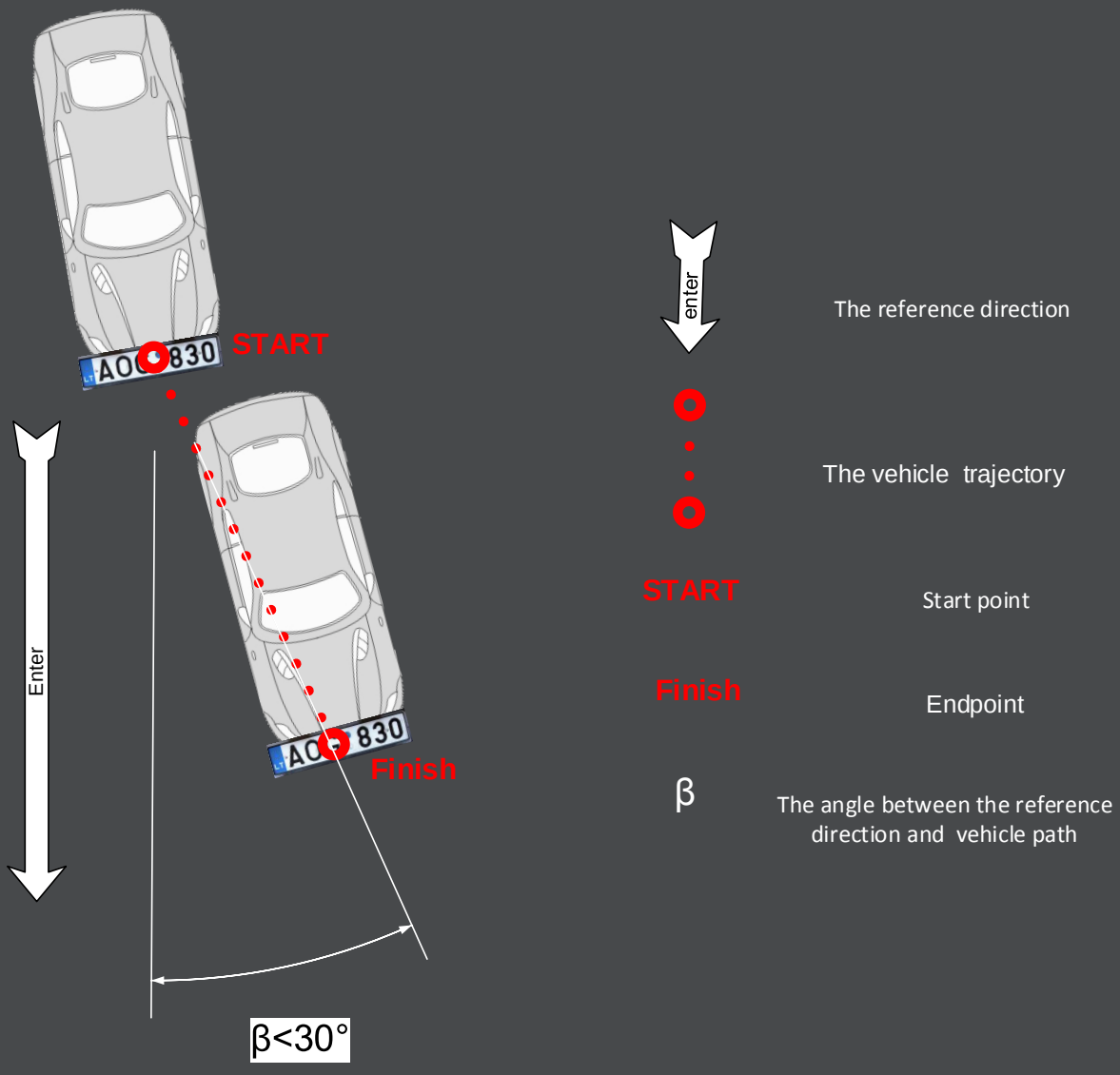
- Identification of symbols
- Reprocessing of obtained data- linking of number plates



6.2.2 ALGORITHM FOR DETECTION OF VEHICLE DIRECTION BY MOVEMENT OF NUMBER PLATE

To detect travelling direction the following requirement should be fulfilled:

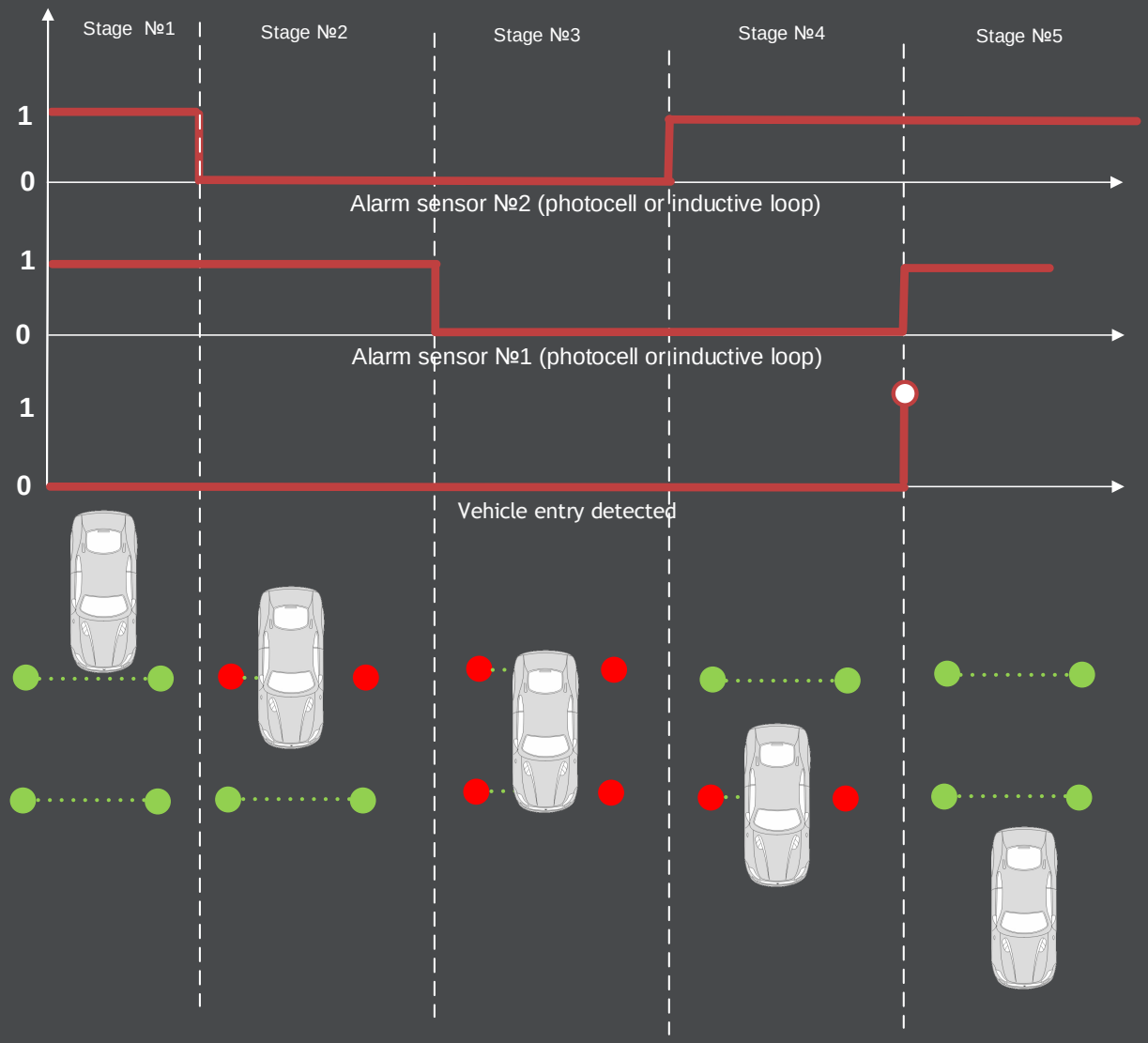
- The course angle of the vehicle should be configured at Settings - Connection - Recognition block
- License plate number must be recognized at least twice, with different coordinates
- The movement pattern of the vehicle must not differ from course angle by more than ± 30%



PICTURE 43 . ALGORITHM FOR DETECTION OF VEHICLE DIRECTION BY MOVEMENT OF NUMBER PLATE IN THE FRAME

6.2.3 ALGORITHM FOR DETECTION OF ENTRY

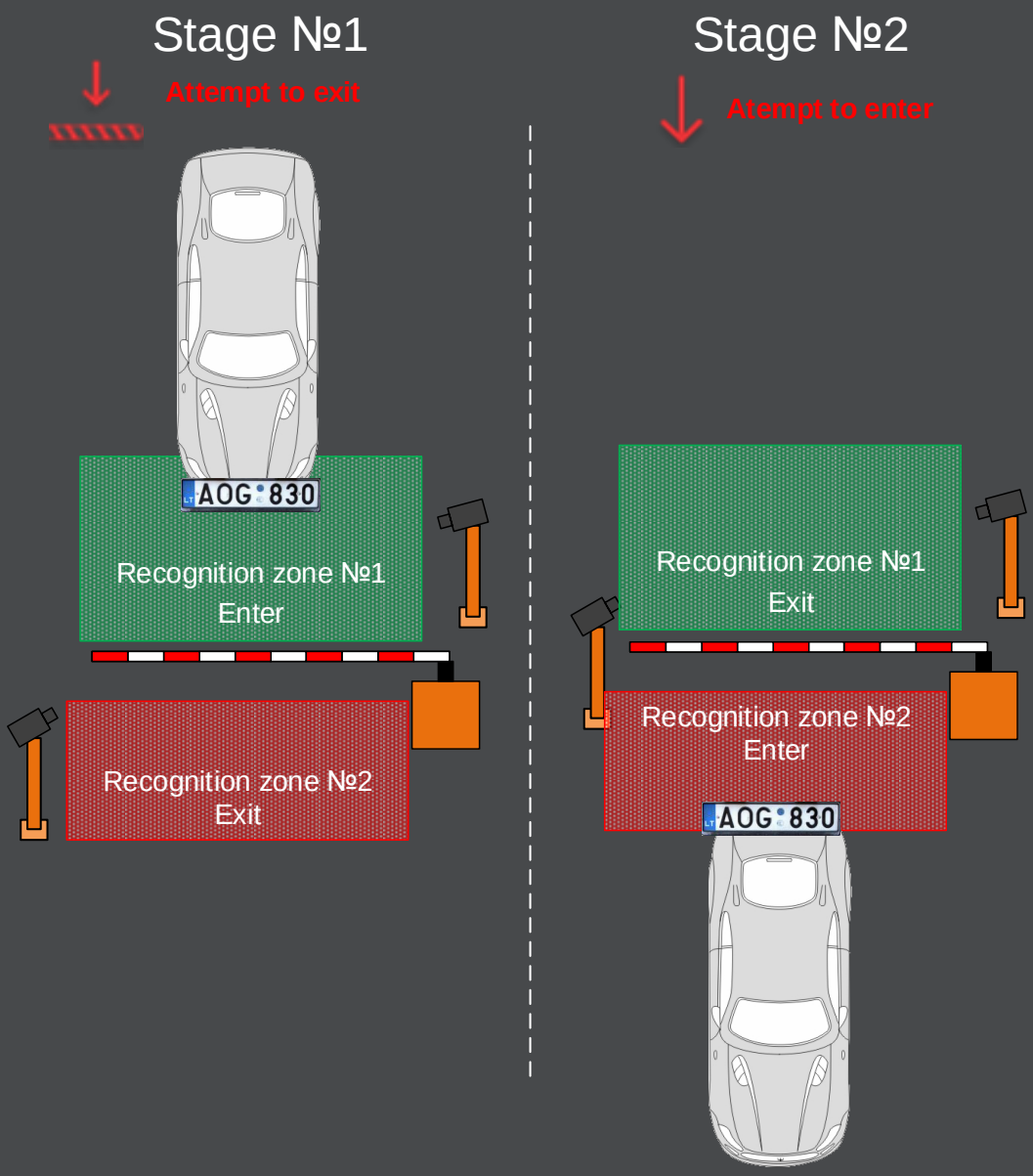
6.2.3.1 BY ALARM SENSORS (PHOTO SENSORS OR INDUCTION LOOPS)



PICTURE 44 OPERATION DIAGRAM OF ALGORITHM - DETECTION OF ENTRY BY ALARM SENSORS

1. Stage 1 - both sensors are normally closed (or open) - there are not any cars
2. Stage 2 - the first sensor is open (or closing) - the car is in the area of the first sensor
3. Stage 3 - both sensors are open (or closed) - the car is in the area of both sensors
   Stage 4 - the first sensor is closed (or open) - the car has passed the first sensor and currently passing the second sensor
4. Stage 5 - the second is closed (or opened) - the car has passed both sensors.
5. When all the five stages are completed the system generates event of vehicle entry detected by alarm sensors. If the first triggered sensor is located at the entrance the event of entry will be generated; if the sensor is installed at the exit the event of departure will be generated. See the connection diagram of executive units and connection settings in the system

6.2.3.2 BY ADJACENT RECOGNITION ZONES



PICTURE 45 OPERATION DIAGRAM OF ALGORITHM FOR DETECTION OF ENTRY BY RECOGNITION ZONES

1. Stage 1 – recognition of license plate in the recognition zone of Entry (*Attempt of entry* event)

2. Stage 2 - recognition of the same license plate number within the time of number plate loss in the recognition zone of Exit (*Entry event*). See the connection diagram in the system

3. The listed stages in the opposite order of travelling direction will generate events *Attempt of departure*, *Departure*.

6.2.4 ALGORITHM FOR CALCULATON OF THE VEHICLE STAY

Duration of the vehicle stay at the area is calculated using the formula shown below:

$T_d = T_{departure} - T_{entry}$, where

$T_{departure}$ - time of detection of vehicle entry

T_{entry} - time of detection of vehicle departure

6.3 SYMBOLS

TABLE 11 SYMBOLS - TRAVELLING DIRECTION OF THE VEHICLE

Icon	Name	Description
	Attempt of entry	The event occurs only in Checkpoint mode (see operation modes) when entry with confirmation is configured(see algorithm for detection of entry).That means that the vehicle is standing in front of checkpoint and attempts to enter, but confirmation of this event is nor received.
	Entry	The event may occur in any operation mode (see algorithm for detection of vehicle direction)
	Attempt of departure	The event occurs only in Checkpoint mode (see operation modes) when entry with confirmation is configured(see algorithm for detection of entry). That means that the vehicle is standing in front of the checkpoint and attempts to leave, but confirmation of this event is not received.
	Departure	The event may occur in any operation mode (see algorithm for detection of vehicle direction)
	Direction is undetermined	Algorithm could not determine vehicle direction. That can happen when single recognition event occurs therefore it is impossible to determine offset value of number plate. (see algorithm for detection of vehicle direction)
	Attempt of entry is not detected	The event occurs when only Entry, Departure and Attempt of departure were detected, but Attempt for entry wasn't captured. That can only happen if the recognition error occurred in the *Entrance* area.
	Attempt of departure is not detected	The event occurs only in case Entry or Re-Entry event was detected, but *Attempt of departure* was not captured. That can happen if recognition failure occurred in the *Exit* area.
	Entry is not detected	The event occurs only in case *Departure* or *Re-Entry* is detected, but entry wasn't captured. That can happen if recognition failure occurred or false sensor alarm was triggered.
	Departure is not detected	The event occurs when *Re-Entry* is detected, but *Departure* was not captured. That can happen if recognition failure occurred or false sensor alarm was triggered.

The following settings impact the above parameters

- Course angle in Settings – Connection – Block for number plate recognition
- Recognition zone settings in the Checkpoint tab
- Alarm sensor settings in the Checkpoint tab

TABLE 12 SYMBOLS -ENTRY

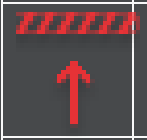
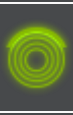
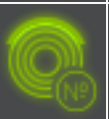
Icon	Name	Description
	Opened automatically	The command to open executive mechanisms (barriers or gates) is issued automatically.
	Closed	The command to open executive mechanisms (barriers or gates) is not issued
	Opened by user	The command to open executive mechanisms (barriers or gates) is issued via control button pressed by user.
	Opened by card	The command to open executive mechanisms (barriers or gates) is issued after the one is received at alarm input of control module
	Successful entry	The event occurs only if entry with confirmation is configured. (see algorithm for detection of entry). Vehicle with allowed access has successfully entered the area.
	Violator	The event occurs only if entry with confirmation is configured. (see algorithm for detection of entry). Vehicle with denied access has entered the area.

TABLE 13 SIMBOLS - ACCESS

Icon	Name	Description
	Allowed	Access allowed always
	Denied	Access denied always
	Allowed by time and date	Access allowed within preset time frame
	Allowed by number of entries	Access allowed if the number of entries is not exceeded
	Allowed by duration of stay at the area	Access allowed in case the duration of stay is not exceeded

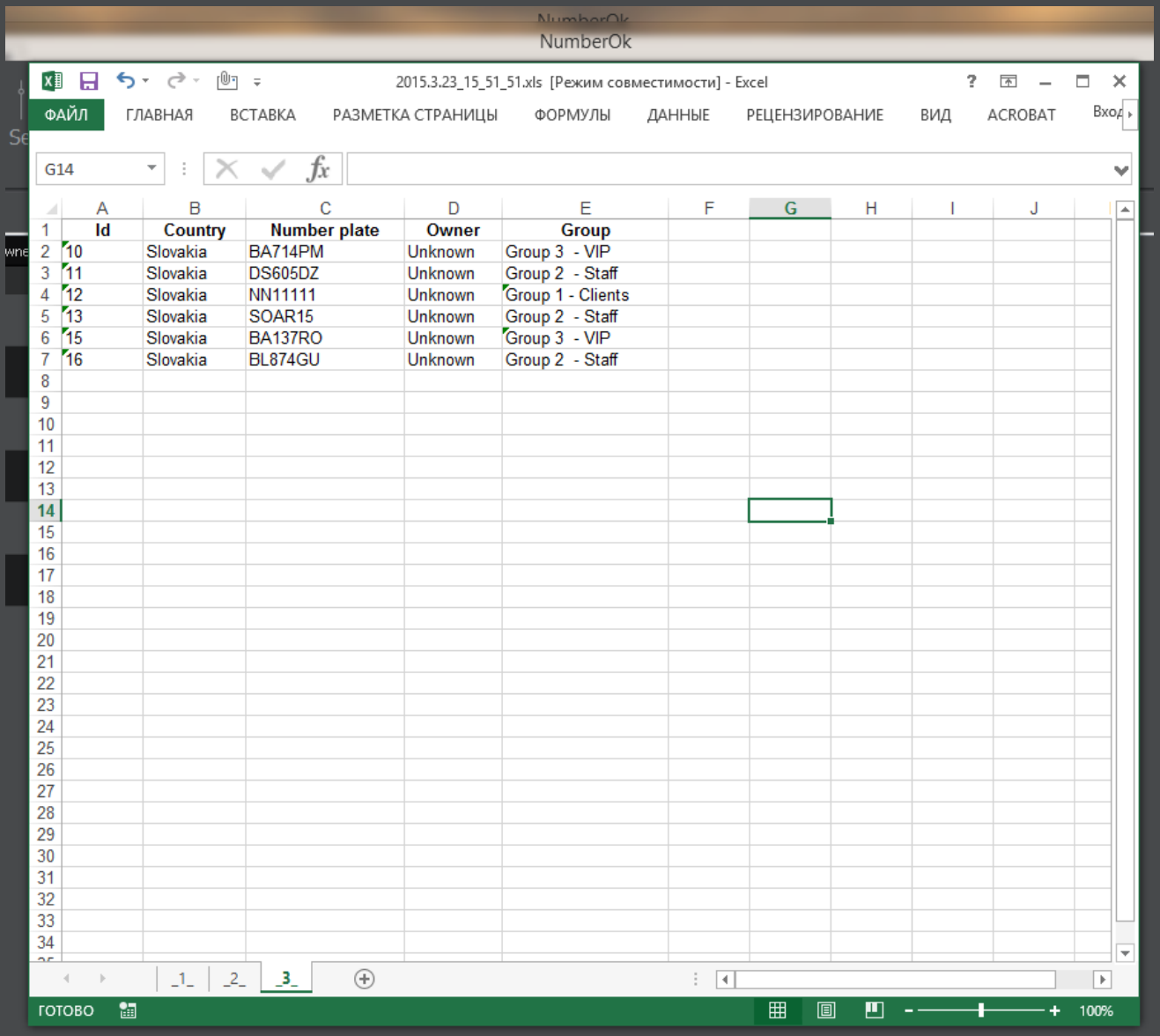
7 APPENDIX

7.1 TYPICAL RTSP LINES OF VARIOUS MANUFACTURERS

Manufacturer	Connection line
AXIS	rtsp:// LOGIN:PASSWORD@<camera ip address>/mpeg4/media.amp
Bosch	rtsp:// LOGIN:PASSWORD@<camera ip address>/rtsp_tunnel
D-Link	rtsp:// LOGIN:PASSWORD@<camera ip address>/play1.sdp
Hikvision	rtsp:// LOGIN:PASSWORD@<camera ip address>:554/h264
MicroDigital	rtsp:// LOGIN:PASSWORD@<camera ip address>/cam0_0
Panasonic	rtsp:// LOGIN:PASSWORD@<camera ip address>/MediaInput/mpeg4
Samsung	rtsp:// LOGIN:PASSWORD@<camera ip address>/h264/media.smp
Sony	rtsp:// LOGIN:PASSWORD@<camera ip address>/media/video1
TP-Link	rtsp:// LOGIN:PASSWORD@<camera ip address>/video.mp4
Vivotek	rtsp:// LOGIN:PASSWORD@<camera ip address>/live.sdp
Novus	rtsp:// LOGIN:PASSWORD@<camera ip address>:554/h264
Partizan	rtsp:// LOGIN:PASSWORD@<camera ip address>: PORT /user= LOGIN & PASSWORD =&channel=1&stream=0.sdp
TVT	rtsp:// LOGIN:PASSWORD@<camera ip address>:554/h264
Avigilion	rtsp:// LOGIN:PASSWORD@<camera ip address>
Cisco	rtsp:// LOGIN:PASSWORD@<camera ip address>/img/media.sav
D-Link	rtsp:// LOGIN:PASSWORD@ip_adx/play1.sdp
GeoVision	rtsp:// LOGIN:PASSWORD@ip_adx:8554/CH001.sdp
Honeywell	rtsp:// LOGIN:PASSWORD@ip_adx/h264
LG	rtsp:// LOGIN:PASSWORD@<camera ip address>/
Toshiba	rtsp:// LOGIN:PASSWORD@<camera ip address>/live.sdp

Typical examples for majority of manufacturers are available at avtonomerok.com(www.number – ok.com) under the tab About product→ Video sources.

7.2 EXPORT OF DATABASE INTO EXCEL



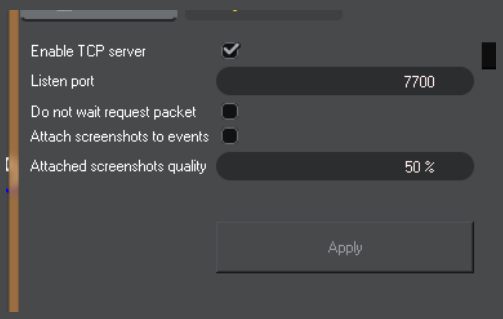
	A	B	C	D	E
1	**Id**	**Country**	**Number plate**	**Owner**	**Group**
2	10	Slovakia	BA714PM	Unknown	Group 3 - VIP
3	11	Slovakia	DS605DZ	Unknown	Group 2 - Staff
4	12	Slovakia	NN11111	Unknown	Group 1 - Clients
5	13	Slovakia	SOAR15	Unknown	Group 2 - Staff
6	15	Slovakia	BA137RO	Unknown	Group 3 - VIP
7	16	Slovakia	BL874GU	Unknown	Group 2 - Staff

7.3 INTEGRATION VIA TCP PROTOCOL – OVERVIEW AND TYPICAL EXAMPLES

General

Configuration file, log files, database are available at **%ALLUSERSPROFILE%\FF\NumberOk3**.

To enable TCP server and obtain snapshots of recognized license plates NumberOk provides configuration parameter



To configure quality of images change configuration parameter NetServer_Attached Screenshots Quality (range of value is from 10 to 100, the default value is 50). **Note: Number of queue events is limited to 1024. If you have slow or loaded network we do not recommend attaching images to events.**

NumberOk waits for connection on TCP port 7700.
Maximum incoming connections - 4 (four).

Constants

```
#define MAX_MESSAGE_SIZE = ( 1 * 1024 * 1024 + sizeof( PacketHeader ) )
#define PACKET_TYPE_StartEventsTransmit          0x19
#define PACKET_TYPE_Event                                0x21
```

Structure

Packet header:

```
typedefstructPacketHeader
{
  UINT32   pType;              // packet type
  UINT32   len;                        // packet length
} PacketHeader, *PPacketHeader;
```

Event Packet:

```
typedefstructEventPacket {
  PacketHeader  header;                              // packet header
  unsigned char data[MAX_MESSAGE_SIZE];     // packet data
} EventPacket, * PEventPacket;
```

Packet structure

PacketHeader	Param1=Value1	0x05	ParamN=ValueN	0x00	0x00	Binarydata

Procedure

1. Connect host NumberOk to TCP port 7700 (check settings of firewall and IP filter if connection is not established). NumberOk will wait for subscription to packet data during 1 second. If subscribe packet is not received, NumberOk will disable connection.
2. Subscribe to events:
   ```
   PacketHeaderpacket;
   packet.len   = 0;
   packet.pType = PACKET_TYPE_StartEventsTransmit;
   // send subscribe packet
   // start receive events
   ```
3. NumberOk will start sending events (recognized events and channel status) to connected client.
4. Packet data (event text, string and binary data) consist of multiple pairs like “Param=Value”. Pairs divided with ASCII char 0x05. There is no ASCII char 0x05 after the last pair “Param=Value”.
5. If you set “NetServer_AttachScreenshotsToEvents” to 1 then after the last pair there are two chars ASCII 0x00 and attached binary data (JPG image). In this case special parameterss are added to event text: "BinaryDataCount" - count of binary data blocks, "BinaryData_%n_Size" - size of n-th data block. It is possible to send several binary

blocks with one packet. Binary blocks are not delimited with each other. Binary data are attached AFTER TEXT PART.

6. When you receive packet: packet header type (*pType*== PACKET_TYPE_Event), packet length (*len*) - length of transmitted packet. **Note: you can receive a part of event.**
7. If NumberOk detects error during data transfer, the system will disable the connection.

Event fields for LPR event:

"State"	“Detected” or “Updated”
"RuntimeID"	Internal
"MovementType"	“Undefined”, “IN”, “OUT”, “TryIN”, “TryOUT”
"MovementTypePrevious"	Previous movement (see “MovementType”)
"LastMovementDetector"	int value [0..5]
"LastMovementDetectorText"	One of: "Undefined" - undetermined, "By movement in frame", "By single zone", "By connected zones", "By photo sensors", "By detection of entry"
"IsUpdated"	if car is updated (coordinates) then "1", otherwise the field is blank
"IsTextUpdated"	if car is updated (number) then "1", else field absent.
"Text"	cars’ number
"Class"	always "Status"
"Type"	always "IAService.LPR"
"ZoneIndex"	int value [0..3], zone index
"ChannelIndex"	int value [0..1], channel index
"PlateStandart"	internal
"PlateTTL"	Duration of storage of number plate in the system
"KPP"	int value [0..3], checkpoint index
"FrameTimeStampMs"	frame timestamp in milliseconds
"ProcessTimeMs"	How much time did recognition take
"DetectedTimestampMS"	detection timestamp, milliseconds
"LastTimestampMS"	last detection timestamp, milliseconds
"PlateConfidence"	Plate recognition confidence
"CoordsX"	Plate’s coordinates in frame, absolute values
"CoordsY"	
"CoordsWidth"	
"CoordsHeight"	
"SymbolsCount"	int value, how many symbols are recognized
"SymbolThreshold"	configuration parameter: threshold for symbol, float
"SymbolConfidence_%d"	float value. %d - [1..SymbolsCount]. Accuracy of symbols.
"isVariableLength"	int value [0,1], configuration parameter.
"EtalonId"	ID from table ‘etalon’
"Owner"	Field ‘description’ of table ‘etalon’
"PassMode"	Field ‘passMode’’ of table ‘etalon’
"PassIntervalStart"	Field ‘iBeginMs’ of table ‘etalon’
"PassIntervalEnd"	Field ‘iEndMs’’ of table ‘etalon’

Event fields for channel status:

"Class"	always "Status"
"Type"	always "VideoSignal";
"Channel"	int value, [0,1]. Channel number from zero.
"State"	One of: "Present", “Lost”. Channel state.
"IntState"	integer state: 1 == "Present", 0 == “Lost”

NumberOk’s Log messages

NumberOk obtains information about connected clients and sends data to log file with log level 0xFF:

Client opens connection:
EventsSource: socket: HEX_SOCKET, new connection from a.b.c.d:port
HEX_SOCKET - socket hex value returned by accept();
a.b.c.d - IP address of remote client;
port - remote client port.

Client closes connection:
EventsSource: socket HEX_SOCKET, close due NdNrNcUaRT. Clients count: N
HEX_SOCKET - socket hex value returned by accept()..
NdNrNcUaRT:
(Nd) Net down
(Nr) Net reset
(Nc) Not connected (send() error)
(U) Host unreachable
(a) Connection aborted (by remote client)
(R) Connection reset (by remote client)
(T) Timeout
Clients count: N - count of current clients

Information about successful data transfer:
EventsSource: socket HEX_SOCKET, sent xxxxxxxx bytes. Ok.

Example linux TCP client for NumberOk

```
/*
 Example client for NumberOk
*/
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>
#include <sys/errno.h>
#include <sys/poll.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
```

```
#include <sys/time.h>

#define SERVER "10.0.1.168"
#define PORT   7700

#define PACKET_TYPE_StartEventsTransmit              0x19
#define PACKET_TYPE_Event                            0x21

#define CONNECT_TIMEOUT_S 2

#define POLL_TIMEOUT_US  200000 // 0.2s

#define MIN(X,Y) ( (X) < (Y) ? (X) : (Y) )

typedefstructPacketHeader
{
   int32_tpType;
   int32_tlen;
} PacketHeader, *PPacketHeader;

// 1 Mb
#define MAX_MESSAGE_SIZE ( 10 * 1024 * 1024 + sizeof( PacketHeader ) )

typedefstruct Packet {
   PacketHeader  header;
   unsigned char data[MAX_MESSAGE_SIZE];
} Packet, * PPacket;

Packet staticPacket;

// Set socket options
voidSetSocketOptions( intfd )
{
   if ( fd == -1 ) return;
   int res = -1;
   longfl = fcntl( fd, F_GETFL, 0 );
   if ( fl< 0 )
   {
        printf( "failed to get F_GETFL to socket %d | errno : %d\n", fd, errno );
   }
   else
   {
       if ( fcntl( fd,F_SETFL, fl | O_NONBLOCK ) < 0 )
       {
            printf( "failed to get F_SETFL to socket %d | errno : %d\n", fd, errno );
       }
   }
```

```
    structtimeval to = { CONNECT_TIMEOUT_S, 0 };
    res = setsockopt( fd, SOL_SOCKET, SO_SNDTIMEO, &to, sizeof( to ) );

    int flag = 1;
    if ( setsockopt( fd, IPPROTO_TCP, TCP_NODELAY, &flag, sizeof(int)) < 0 )
    {
        printf( "failed to set TCP_NODELAY to socket %d | errno : %d\n", fd, errno );
    }

    flag = 1;
    if ( setsockopt( fd, SOL_SOCKET, SO_KEEPALIVE, &flag, sizeof(int) ) < 0 )
    {
        printf( "failed to set SO_KEEPALIVE to socket %d | errno : %d\n", fd, errno );
    }

    flag = 5; // time to send keepalive after last data packet
    if ( setsockopt( fd, SOL_TCP, TCP_KEEPIDLE, &flag, sizeof(int) ) < 0 )
    {
        printf( "failed to set TCP_KEEPIDLE to socket %d | errno : %d\n", fd, errno );
    }

    flag = 3; // failed keepalive probes before connection would be marked as dead
    if ( setsockopt( fd, SOL_TCP, TCP_KEEPCNT, &flag,sizeof(int) ) < 0 )
    {
        printf( "failed to set TCP_KEEPCNT to socket %d | errno : %d\n", fd, errno );
    }

    flag = 3; // interval between probes
    if ( setsockopt( fd, SOL_TCP,TCP_KEEPINTVL, &flag, sizeof(int) ) < 0 )
    {
        printf( "failed to set TCP_KEEPINTVL to socket %d | errno : %d\n", fd, errno );
    }
}

intSelectSocket( intfd, unsigned inttimeoutUs )
{
    if ( fd<= 0 )
    {
       return -1;
    }

    structtimevaltimet;
    memset(&timet, 0, sizeof( timet ) );

    timet.tv_sec = timeoutUs / 1000 / 1000;
    timet.tv_usec= timeoutUs  - (timet.tv_sec * 1000 * 1000);
    int res = 0;
```

```
    unsignedinttimeoutMs = timeoutUs< 1000 ? 1 :timeoutUs / 1000;

    while (1)
    {
        structpollfdpollfd;
        memset(&pollfd, 0, sizeof( pollfd ) );
        pollfd.fd = fd;
        pollfd.events = POLLIN | POLLPRI;

        inteventsCount = poll( &pollfd, 1, timeoutMs );
        if ( eventsCount == 1 )
        {
            if ( ( pollfd.revents& ( POLLERR | POLLHUP | POLLNVAL | POLLPRI ) ) != 0x00 )
            {
                res = -1;
            }
            else if ( ( ( pollfd.revents& ( POLLIN ) ) != 0x00 ) )
            {
                res = 1;
            }
        }
        else
        {
            res = 0;
        }
        break;
     }
     return res;
}

int main( intargc, char ** argv )
{
    structsockaddr_innomerokaddr;
    intfd = socket( AF_INET, SOCK_STREAM, IPPROTO_TCP );
    int res;
    // create socket
    if ( fd> 0 )
    {
        SetSocketOptions(fd );

        // set address:port
        memset(&nomerokaddr, 0, sizeof( nomerokaddr ) );
        nomerokaddr.sin_family      = AF_INET;
        nomerokaddr.sin_addr.s_addr = inet_addr( SERVER );
        nomerokaddr.sin_port        = htons( PORT );

        socklen_tlen = sizeof( nomerokaddr );
```

```
        // switch to blocking mode until connect
        longfl = fcntl( fd, F_GETFL, 0 );
        if ( fl>= 0 )
        {
            if ( fcntl( fd,F_SETFL, fl& ( ~O_NONBLOCK ) ) < 0 )
            {
                printf( "failed to set F_SETFL | errno : %d", errno );
            }
        }
        else
        {
            printf( "failed to get F_GETFL | errno : %d\n", errno );
        }
        // connect
        res = connect( fd,(structsockaddr*)&nomerokaddr, len );
        if ( res == 0 )
        {
            // switch to non blocking mode
            fl = fcntl( fd, F_GETFL, 0 );
            if ( fl>= 0 )
            {
                if ( fcntl( fd, F_SETFL, fl | O_NONBLOCK ) < 0 )
                {
                    printf( "failed to set F_SETFL | errno : %d", errno );
                }
            }
            else
            {
                printf( "failed to get F_GETFL | errno : %d\n", errno );
            }
            // subscribe to events
            PacketHeader packet;
            packet.len   = 0;
            packet.pType = PACKET_TYPE_StartEventsTransmit;
            // send subscribe packet
            res = write( fd, (char *)&packet, sizeof( packet ) );
            if ( res == sizeof( PacketHeader ) )
            {
                res = 0;
                                        intreceivd = 0;
//                                      FILE * f = fopen( "./dump.bin", "ab" );
                        intjpg_count = 0;
                while ( 1 )
                {
                    // wait for event
                    res = SelectSocket( fd, POLL_TIMEOUT_US );
                    if ( res == -1 )
                    {
```

```
                    break;
                }
                else if ( res == 1 )
                {
                    // receive data
                    res = recv( fd, (char*)&staticPacket + receivd, MAX_MESSAGE_SIZE +
sizeof( PacketHeader ) - receivd, MSG_DONTWAIT );
                    if ( res > 0 )
                    {
//                        fwrite( (char*)&staticPacket + receivd, res, 1, f );
//                        fflush( f );
                        receivd += res;
                        printf( "received %d from %d\n", receivd, staticPacket.header.len + sizeof(
PacketHeader ) );
                        // is this event?
                        if ( staticPacket.header.pType ==
PACKET_TYPE_Event&&staticPacket.header.len> 0 )
                        {
                            // we have all data here ?
                            if ( staticPacket.header.len<= ( receivd - sizeof( PacketHeader ) ) )
                            {
                                        printf( "process received %d from %d\n", receivd,
staticPacket.header.len + sizeof( PacketHeader ) );
                                char * stringData = (char *)staticPacket.data;
                                intstringDataLen = strlen( stringData );
                                char * strEnd = stringData + stringDataLen;
                                // print to stdout
                                const char * delim = "\05";
                                char * token = strtok( stringData, delim );
                                char * saveptr = 0;
                                                        while ( token != 0 && token
<strEnd )

                                                        {
//                                                          printf( "%s\n", token );
                                                          if ( strstr( token,

"BinaryData_1_Size" ) != 0  )

                                                            {
                                                    printf( "%s ======== ", token );
                                                    token += strlen(

"BinaryData_1_Size=" );

                                                    saveptr = token;
                                                    while ( *token  != *delim&& *token

!= 0x00  )

                                                    {
                                                        ++token;
                                                    }
                                                    *token = 0x00;
                                                    intjpgLen = atoi( saveptr );
```

```
                                                        *token = 0x05;
                                                                char * jpgDta =
stringData + stringDataLen + 2;

        charjpgFileName[128];

        snprintf(jpgFileName, 127, "./%d.jpg", jpg_count );
                                                        printf( " %d %s\n", jpgLen,
jpgFileName );
                                                                ++jpg_count;
                                                                FILE * f =
fopen(jpgFileName, "wb" );
                                                                fwrite(jpgDta,
jpgLen, 1, f );
                                                                fclose( f );
                                                          }
                                                          token = strtok( 0, delim );
                                                      }

                        // copy the rest of other packet if present
                        int diff = receivd - ( staticPacket.header.len + sizeof( PacketHeader )
);
                        printf( "diff %d, strlen %d\n==========================\n", diff,
stringDataLen );
                        if ( diff > 0 )
                        {
                            memcpy( (char *)&staticPacket, stringData +
staticPacket.header.len, diff );
                            receivd = diff;
                            memset( (char *)&staticPacket + receivd, 0x00,
MAX_MESSAGE_SIZE - receivd - 1 );
                        }
                        else
                        {
                            receivd = 0;
                        }
                                                            }
                    }
                }
                else
                {
                    if ( res == 0 || ( ( res < 0 ) && ( errno == EFAULT ) ) )
                    {
                        if ( errno == EINPROGRESS )
                        {
                            sleep( 1 );
                            continue;
                        } // else break it
```

```
                        } // EINPROGRESS
                        // error
                        break;
                    }
                }
            }
        }
        shutdown(fd, SHUT_RDWR );
    }
    else
    {
        printf( "connect(...) failed | errno : %d | %s\n", errno, strerror( errno ) );
    }
    close(fd );
  }
  else
  {
    printf( "socket(...) failed | errno : %d | %s\n", errno, strerror( errno ) );
  }
}
```