



Guide for configuring and working with the ZKTeco PUSH SDK integration module

ACFA PSIM 1.11

Last update 16/07/2026

Table of Contents

1 Introduction into the Guide for configuring and working with the ZKTeco PUSH SDK integration module	3
1.1 Purpose of the document.....	3
1.2 General information about the ZKTeco PUSH SDK integration module.....	3
2 Supported hardware and licensing of the ZKTeco PUSH SDK integration module.....	4
3 Configuring the ZKTeco PUSH SDK integration module.....	5
3.1 Configuring the parent object of the ZKTeco PUSH SDK integration module.....	5
3.2 Configuring the ZKTeco PUSH SDK controller.....	6
3.3 Configuring the ZKTeco PUSH SDK door	6
4 Working with the ZKTeco PUSH SDK integration module	9
4.1 General information about working with the ZKTeco PUSH SDK integration module	9
4.2 Managing the ZKTeco PUSH SDK parent object.....	9
4.3 Managing the ZKTeco PUSH SDK controller.....	9
4.4 Managing the ZKTeco PUSH SDK door	10
4.5 Example of a configured macro when working with the ZKTeco PUSH SDK integration module.....	12

1 Introduction into the Guide for configuring and working with the ZKTeco PUSH SDK integration module

On the page:

- [Purpose of the document](#)
- [General information about the ZKTeco PUSH SDK integration module](#)

1.1 Purpose of the document

The *Guide for configuring and working with the ZKTeco PUSH SDK integration module* is a reference and information manual and is intended for configuration specialists and operators of the *ZKTeco PUSH SDK* integration module. This module is a part of the access control system (ACS) implemented on the basis of *ACFA PSIM*.

The Guide has the following information:

1. General information about the *ZKTeco PUSH SDK* integration module.
2. Configuring the *ZKTeco PUSH SDK* integration module.
3. Working with the *ZKTeco PUSH SDK* integration module.

1.2 General information about the ZKTeco PUSH SDK integration module

The *ZKTeco PUSH SDK* integration module is a part of the ACS implemented on the basis of *ACFA PSIM* and is used to provide interaction between the *ZKTeco PUSH SDK* ACS and *ACFA PSIM* (configuration, monitoring, control).

Note

For detailed information about the *ZKTeco PUSH SDK* ACS, see the official reference documentation for this system (manufacturer ZKTeco).
For the correct operation, you must install *ZKTeco SDK*.

Before configuring the *ZKTeco PUSH SDK* integration module, you must do the following:

1. Install the *ZKTeco PUSH SDK* hardware on the protected facility (see the *ZKTeco PUSH SDK* official reference documentation).
2. Connect the *ZKTeco PUSH SDK* hardware to the *Axxon PSIM* Server (see the *ZKTeco PUSH SDK* official reference documentation).

2 Supported hardware and licensing of the ZKTeco PUSH SDK integration module

Manufacturer	ZKTeco Address: ZK Building, Wuhe Road, Gangtou, Bantian, Buji Town, Longgang District, Shenzhen, China Phone: +86 755-33985019 Fax: 12 39 89 901 Website: https://www.zkteco.com/en/ Email: service@zkteco.com
Type of integration	SDK
Hardware connection	Ethernet

Supported hardware

All *ZKTeco* access control devices that work via the PUSH SDK protocol (SpeedFace, ProFace, ELITE PASS terminals, and Inbio Pro, C2-260, and so on controllers) are integrated.

Module licensing

Per each door (**ACFARreader**). Door can contain one or two readers depending on the model of an access controller. For more information, see the technical specifications of the corresponding controller.

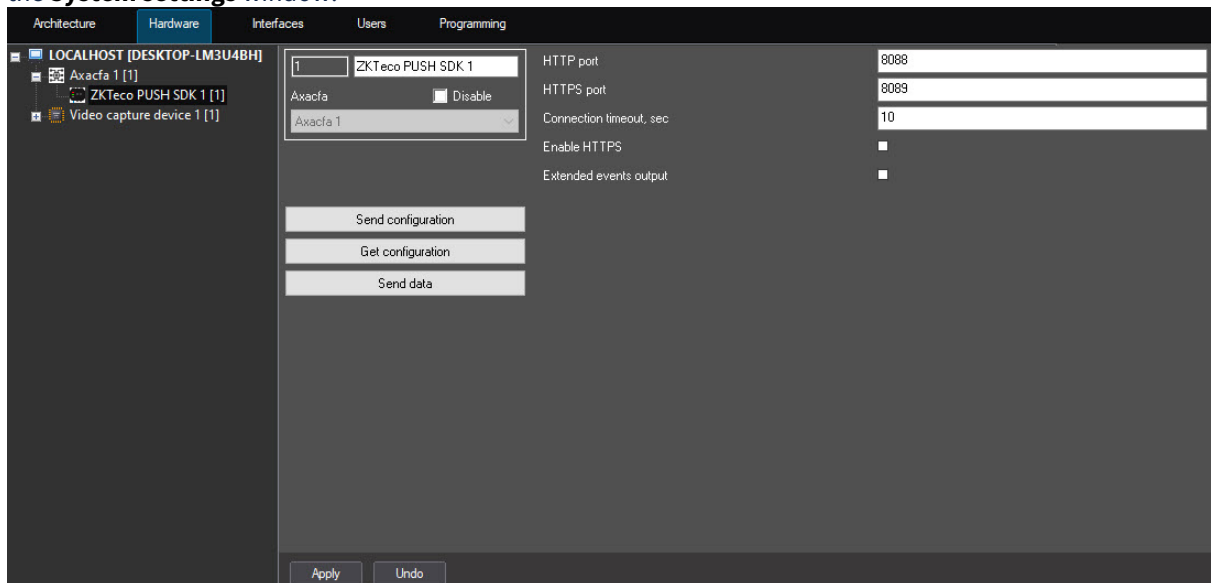
3 Configuring the ZKTeco PUSH SDK integration module

3.1 Configuring the parent object of the ZKTeco PUSH SDK integration module

To work with the *ZKTeco PUSH SDK* integration module, you must install and configure the *AxACFA* feature. For more details, see [Connecting and configuring the AxACFA feature](#).

To configure the *ZKTeco PUSH SDK* parent object, do the following:

1. Create the **ZKTeco PUSH SDK** parent object on the basis of the **Axacfa** object on the **Hardware** tab of the **System settings** window.

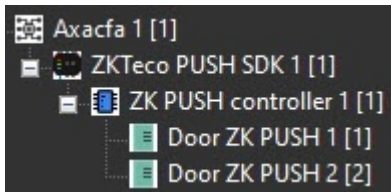


2. In the **HTTP port** field, enter the number of the HTTP port to connect to the *ZKTeco* SDK, **8088** by default.
3. If you need to use the HTTPS protocol, set the **Enable HTTPS** checkbox, and in the **HTTPS port** field, enter the number of the HTTPS port for connection to the *ZKTeco* SDK, **8089** by default.
4. In the **Connection timeout, sec** field, enter the time period in seconds after which the connection is considered lost and the reconnection of the *ZKTeco* device to *ACFA PSIM* occurs, **10** by default.
5. By default, the **Extended events output** checkbox is clear. So one event is generated when the *ZKTeco PUSH SDK* user is updated. More detailed information is written to the log file. Set the checkbox if you want to receive detailed events about each user update for each *ZKTeco PUSH SDK* device.
6. To save the connection settings, click the **Apply** button.
7. Wait until the event of requesting the connection from the required *ZKTeco* device appears in the event viewer since, in this integration, *ZKTeco* devices connect to *ACFA PSIM*.

Example of the event of requesting the connection from the *ZKTeco* device in the event viewer:

Event viewer 1 [~5]						☐ Show filters	Clear
co...	Source	Event	Region	Add info	Card	Date and time	
0	ZKTeco PUSH SDK 1	Connected				10/4/2024 2:38:48 PM	
0	ZKTeco PUSH SDK 1	Connected				10/4/2024 2:38:52 PM	
0	ZKTeco PUSH SDK 1	Disconnected				10/4/2024 2:38:52 PM	
0	Display 1	Activated		VANURIN		10/4/2024 2:39:08 PM	
0	ZKTeco PUSH SDK 1	System message		ZK Push device SpeedFace-VSL(QR) has requested registration SN: CMWN213560007 /Address : 10.0.11.125		10/4/2024 2:39:29 PM	

8. After the event of requesting the connection appears, you can download the configuration and build the hardware tree. For this, click the **Get configuration** button.
As a result, the hardware tree is built:



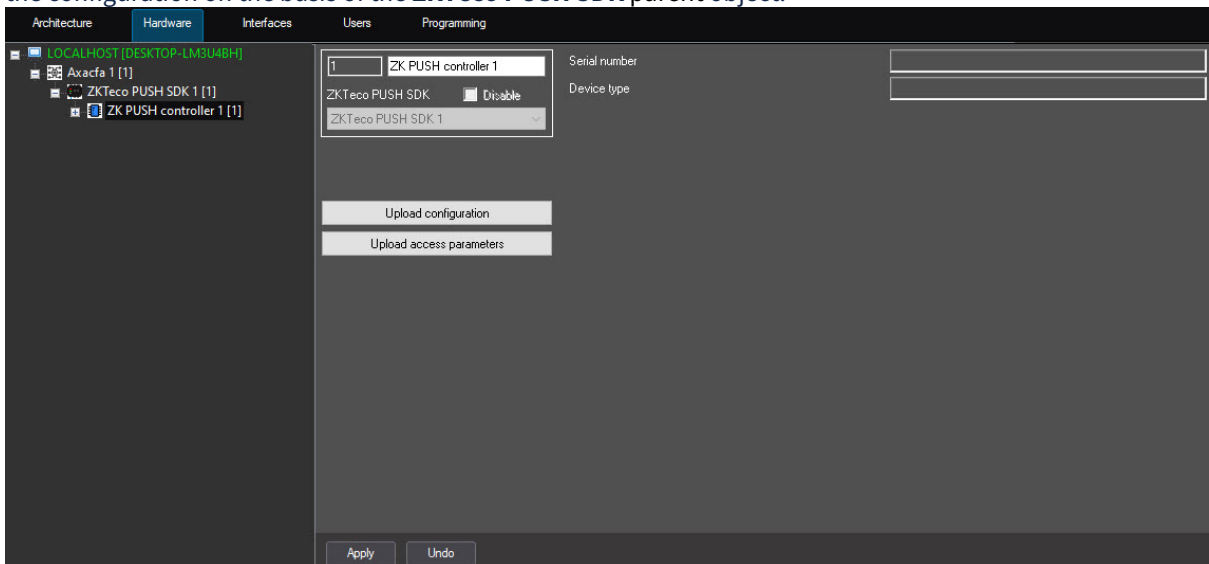
9. Click the **Send configuration** button to send the configuration to all *ZKTeco PUSH SDK* devices.
10. Click the **Send data** button to write the user access parameters from the *Access Manager* to all *ZKTeco PUSH SDK* devices.

Configuring the *ZKTeco PUSH SDK* parent object is complete.

3.2 Configuring the ZKTeco PUSH SDK controller

To configure the *ZKTeco PUSH SDK* controller, do the following:

1. Go to the settings panel of the **ZK PUSH controller** object that is created automatically after you download the configuration on the basis of the **ZKTeco PUSH SDK** parent object.



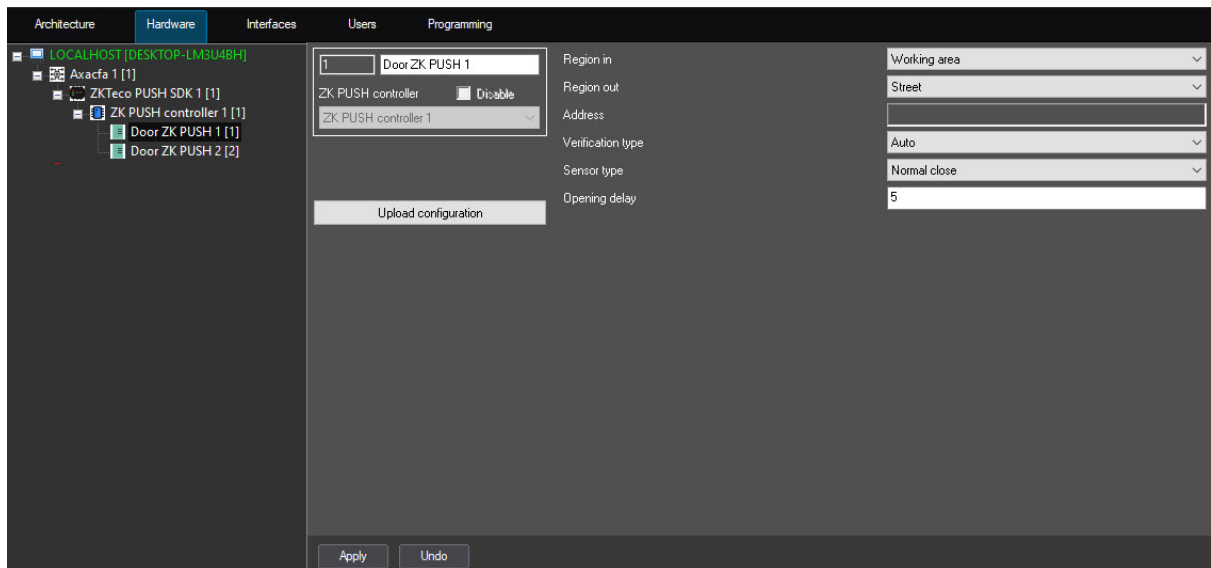
2. Click the **Upload configuration** button to upload the configuration into this *ZKTeco PUSH SDK* controller.
3. Click the **Upload access parameters** button to upload user access parameters into this *ZKTeco PUSH SDK* controller.
4. Click the **Apply** button to save the settings.

Configuration of the *ZKTeco PUSH SDK* controller is complete.

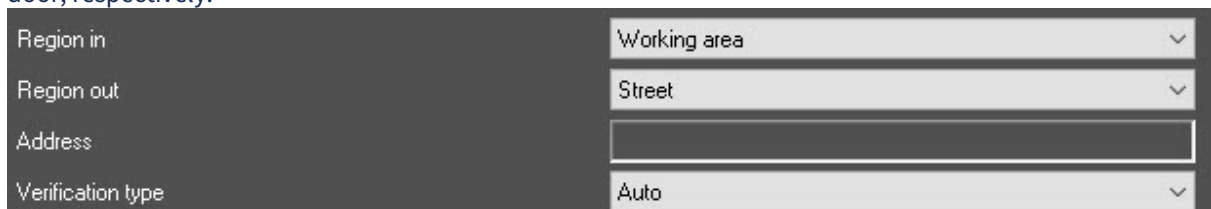
3.3 Configuring the ZKTeco PUSH SDK door

To configure the *ZKTeco PUSH SDK* door, do the following:

1. Go to the settings panel of the **Door ZK PUSH** object that is created automatically after you download the configuration on the basis of the **ZK PUSH controller** object.



2. In the **Region in** and **Region out** drop-down lists, select the regions on the exit and entry side through the door, respectively.

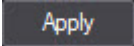
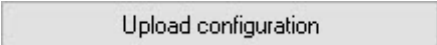


3. From the **Verification type** drop-down list, select the required type of user verification:

- a. **Auto** (default),
- b. **Fingerprint**,
- c. **User ID**,
- d. **Password**,
- e. **Card**,
- f. **Fingerprint or password**,
- g. **Fingerprint or card**,
- h. **Card or password**,
- i. **User ID + fingerprint**,
- j. **Fingerprint + password**,
- k. **Card + fingerprint**,
- l. **Card + password**,
- m. **Fingerprint + password + card**,
- n. **User ID + fingerprint + password**,
- o. **User ID + fingerprint or card + fingerprint**,
- p. **Face**,
- q. **Face + fingerprint**,
- r. **Face + password**,
- s. **Face + card**,
- t. **Face + fingerprint + card**,
- u. **Face + fingerprint + password**,
- v. **Finger vein**,
- w. **Finger vein + password**,
- x. **Finger vein + card**,
- y. **Finger vein + password + card**,
- z. **Other**.

4. From the **Sensor type** drop-down list, select the required type of door sensor:
 - a. **None** (default)—door sensor isn't used.
 - b. **Normal open**—door is always open when the power supply is on.
 - c. **Normal close**—door is always closed when the power supply is on.

Sensor type	Normal close
Opening delay	5

5. In the **Opening delay** field, specify in seconds the maximum allowable time of delay for closing the door after user verification is confirmed. An alarm is generated when the door is open longer than the specified time. The default value is **0**.
6. Click the **Apply**  button to save the settings.
7. Click the **Upload configuration**  button to upload this configuration to the corresponding controller.

Configuration of the ZKTeco *PUSH SDK* door is complete.

4 Working with the ZKTeco PUSH SDK integration module

4.1 General information about working with the ZKTeco PUSH SDK integration module

The following interface objects are used to work with the *ZKTeco PUSH SDK* integration module:

1. **Map.**
2. **Event Viewer.**
3. **Access Manager.**

For the information on configuring these interface objects, see the *Axxon PSIM Administrator's Guide* and [Guide for configuring and working with the Access Manager module](#).

For the information on working with these interface objects, see the *Axxon PSIM Operator's Guide*.

4.2 Managing the ZKTeco PUSH SDK parent object

You can manage the *ZKTeco PUSH SDK* parent object in the **Map** window using the menu of the **ZKTeco PUSH SDK** object.

Command to manage the *ZKTeco PUSH SDK* parent object:

- **Sync time across devices**—write the time of the *ACFA PSIM* Server to all *ZKTeco PUSH SDK* devices.

The *ZKTeco PUSH SDK* parent object can have the following states:

	Connected
	Disconnected
	Unknown

4.3 Managing the ZKTeco PUSH SDK controller

You can manage the *ZKTeco PUSH SDK* controller in the **Map** window using the menu of the **ZK PUSH controller** object.

Commands to manage the *ZKTeco PUSH SDK* controller are described in the table:

Menu command	Function
Cancel all alarms	Cancel all alarms on the devices connected to the controller, set them to normal state
Get IP address	Get IP address of the controller

Menu command	Function
Sync date and time	Write the date and time of the <i>ACFA PSIM</i> server to all devices connected to the controller

The *ZKTeco PUSH SDK* controller can have the following states:

	Connected
	Disconnected
	Unknown

4.4 Managing the ZKTeco PUSH SDK door

You can manage the *ZKTeco PUSH SDK* door in the **Map** window using the menu of the **Door ZK PUSH** object.

Commands to manage the *ZKTeco PUSH SDK* door are described in the table:

Menu command	Function
Handle alarm	Handle door alarm, set it to <i>normal state</i>
Open	Open the door
Close	Close the door
Open permanently	Open the door for an unlimited time

The *ZKTeco PUSH SDK* door can have the following states:

	Disconnected
	Open
	Closed
	Alarm
	Accidental door opening alarm
	Tamper alarm
	Duress password alarm
	Duress fingerprint alarm
	Door sensor timeout alarm
	Main power failure alarm
	Battery power failure alarm
	Reader disassembly alarm
	No sensor
	Unknown

4.5 Example of a configured macro when working with the ZKTeco PUSH SDK integration module

- ✓ Creating and using macros
- Examples of macros

When working with the *ZKTeco PUSH SDK* integration module, you can configure a macro that will trigger when an event is received from the *ZKTeco PUSH SDK* device.

Example of a configured macro:

The screenshot shows the configuration window for a macro named "Macro 1". The interface includes a "Disable" checkbox, a "Fast call" dropdown set to "None", and a "Response sending delay (s)" input field. The "Icon type" is set to "Macro 1".

Settings

State: Standard (dropdown) [Local] [Hidden]

Events

Type	Number	Name	Event
Door ZK PUSH	1	Door ZK PUSH 1	Access Denied

Parameters

Name	Value
------	-------

Actions

Type	Number	Name	Action
Door ZK PUSH	1	Door ZK PUSH 1	Close

Parameters

Name	Value
------	-------

Buttons: Apply, Undo