Manual

of

XK-SDK

BY XANDAR KARDIAN

XANDAR KARDIAN

· Release notes

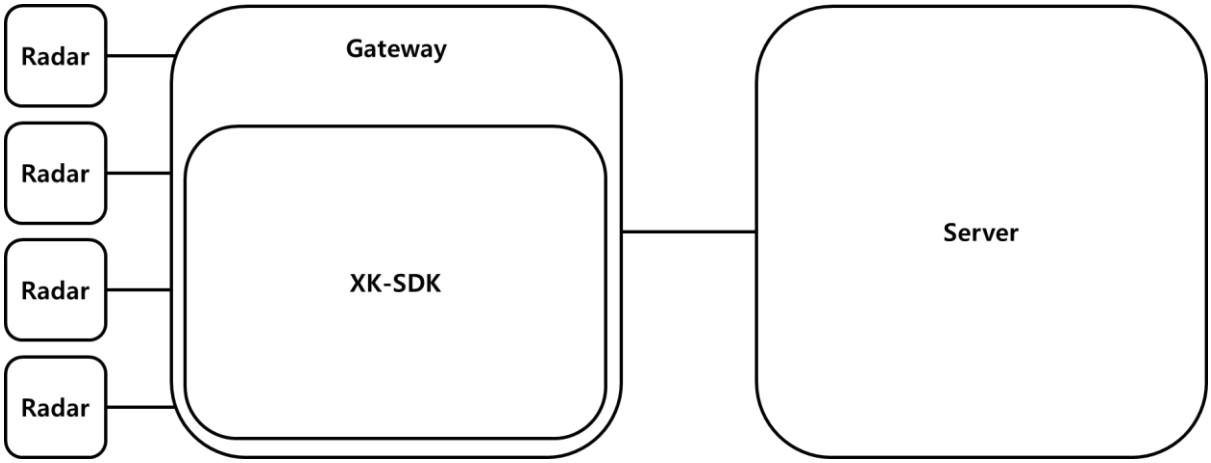
Date	Release	Edit	Note	
19-04-02	0.1	Jerry	Created.	
19-07-05	0.2	Jerry	Changed Picture.	
19-08-08	1.0	Jerry	Updated about XK-SDK 2.0.0.	

XANDAR KARDIAN

1. Introduction

XK-SDK is XK-Gateway software that sends and receives data between radar and server.

You can make changes as you like. It uses UART serial communication interface between Radar and Gateway, and Gateway connects to various servers or client devices using HTTP, HTTPS communication. There are various options for running the Gateway device. However, it provides the ability to change settings in an easy way.



2. Requirements

This document is for XK-SDK 2.0.0 or later software.

3. Build Environment

3.1. Extract SDK

1) Open the 'terminal' on the XK-Gateway.
2) Download latest XK-SDK.
3) Enter the download path.

   $ cd *[DOWNLOAD_PATH]*

   ex) cd /home/pi/Download

4) Extract XK-SDK.

   $ tar -xzvf *XK_SDK.tar.gz* /home/pi

   ex) tar -xzvf XK_SDK_V_2_0_0.tar.gz /home/pi

5) Change permission.

   $ sudo chmod 777 -R *XK_SDK_[LATEST_VERSION]*

   ex) sudo chmod 777 -R /home/pi/XK_SDK_V_2_0_0

3.2. Install SDK

1) Enter the latest version SDK project path

   $ cd *XK_SDK_[LATEST_VERSION]*

   ex) cd /home/pi/XK_SDK_V_2_0_0

2) Make and install

   $ make clean

   $ make

   $ make install

4. Run

4.1. How to Run

```
$ sudo xksdk [-option]
```

Note: The options are explained in later chapters.

If necessary, you can view descript with `sudo xksdk -h`.

4.2. How to Kill

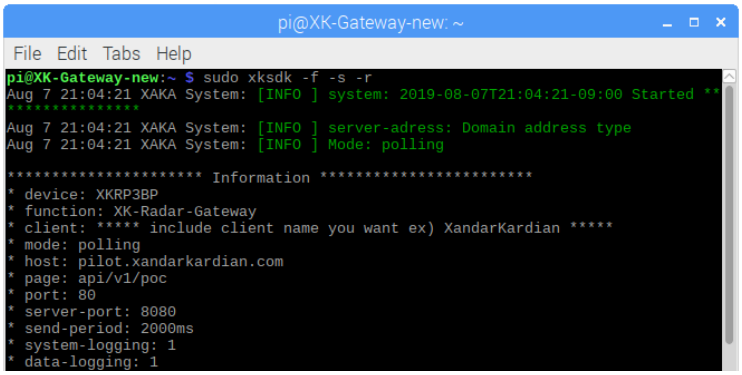
4.2.1. Running in the Background

```
$ sudo killall -2 xksdk
```

Note: If it is killed abnormally, it cannot be executed again.

If you want to run it again, run it with the ***-f*** option.

4.2.2. Common run

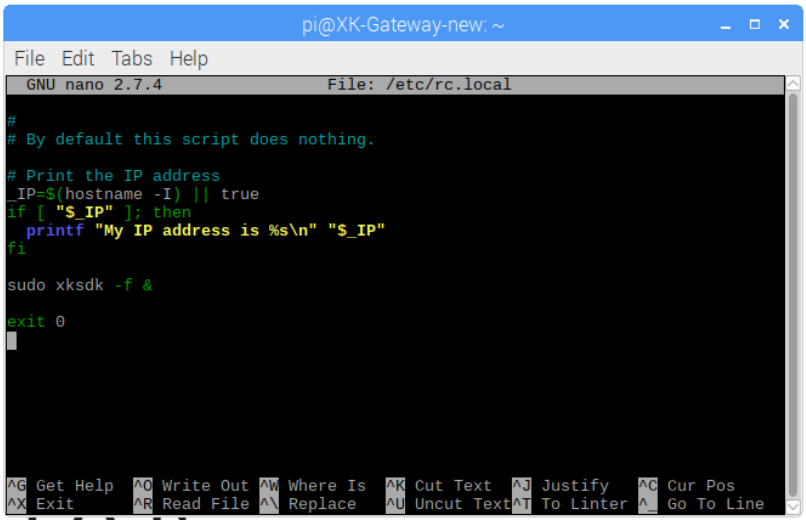


Enter key **[ctrl] + [c]**

4.2.3. Startup Program Registration

```
$ sudo nano /etc/rc.local
```

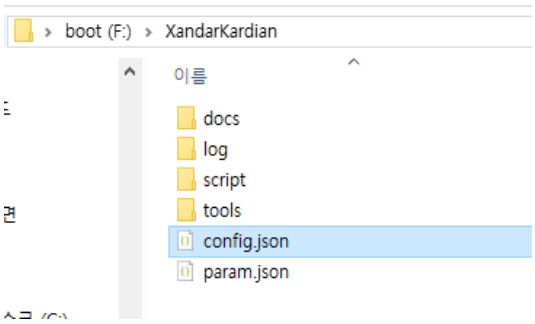
Insert "**sudo xksdk -f &**" as shown below



5. Configuration

5.1. SDK Configuration

1) **Power off and remove** micro 5 pin cable.
2) **Remove** the Micro SD Card
3) **Connect** Micro SD Card to PC via USB adapter
4) **Open** path "**boot/Xandarkardian**" inside USB drive
5) **Open** file "**config.json**"



5.2. Configuration List

Hierarchy	Object	Type	Range	Description
info	**name**	String		- SDK name
	version	String		- SDK version information
	function	String		- What does it do
	excutable	String		- How to run

XANDAR KARDIAN

	client	String		- Name sent to the server to indicate the owner of the gateway device. ex) "XandarKardian" "Google" "GRASP"
info/ api	**name**	String		- API name to communicate with the server
	version	String		- API version to communicate with the server
Info/ description	**detail**	String		- Description about the XK-SDK
	author	String		- Authors who contributed to the SDK
config	**authorization**	String	"none" or real key	- The key value assigned when signing up at XandarKardian's web server. To identify gateway. ex) "gk_d0202d93d0d29293d9" "none"
	device	String	"XKRP3BP"	- Gateway device hardware name
	mode	String	"polling" ~ "event"	- Change the XKGW-API Mode - Polling mode: Gateway device sends data to server periodically. - Event mode: Getting Data when user want it. Note: Refer the XKGW-API documentation for details.
	send-period	Integer	0 ~ 2147483647	- How often to send data to the server in polling mode - Unit is [ms]. ex) 2000 -> 2 sec
	system-log	Integer	0, 1	- Turn on / off system logging. - The system log file is in "/var/log/xk/sys". - Created with *.xkl extension.
	data-log	Integer	0, 1	- Turn on / off data logging. - The data log file is in "/var/log/xk". - Created with *.xkl extension. - Save data received from radar, separated by commas. - Save 1 frame per second.
	auto-reboot	Integer	0, 1	- Turn on / off data logging. - The data log data is in "/var/log/xk". - Created with *.xkl extension.

	rID-type	String	"r", "s"	- In the JSON data structure sent to the server,
				Object name type to distinguish radar data.
				ex) "r": included radar ID
				...
				r31001: {
				}
				...

				ex) “s”: included serial number
				…
				000068000001: {
				}
				…

	data-type	String	“v”, “a”	- In the JSON data structure sent to the server,
				Object name type to distinguish parameter number.
				ex) “v”: v type parameter number
				r31001: {
				“v0”: 0.00,
				“v1”: 51.00,
				“v2”: -1.00,
				“v3”: 5.77
				}
				
				ex) “a”: a type parameter number (v0 = a3)
				r31001: {
				“a3”: 0.00,
				“a4”: 51.00,
				“a5”: -1.00,
				“a6”: 5.77
				}

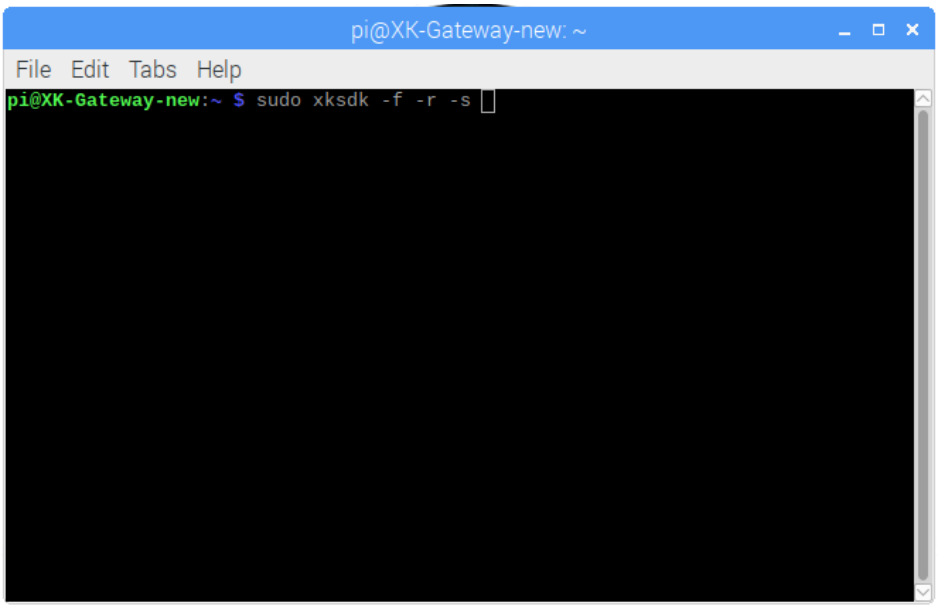
config/ endpoint	**host**	String		- Server domain or IP address to transfer data. ex) “xandarkardian.com” “192.168.0.200” Note: Remove special characters or “http://”, “https://”
	page	String		- Server page to transfer data. ex) “sensor/test” “api/v1/poc” Note: Remove first “/”
	port	Integer	0 ~ 65535	- Server port number to transfer data. ex) 80 443
config/ server	**port**	Integer	0 ~ 65535	- Port number for the client connecting to the gate when in event mode ex) 8080 3009

6. Run options

6.1. Execution Option

```
$ sudo xksdk [-option]
```

Option	Description
-h	Print optional command.
-f	This SDK does not allow duplicate execution. However, this option is used to terminate the existing running process and force it to run.
-d	Print data from the radar to the terminal without sending or receiving.
-s	Run with printing JSON message to be sent to the server.
-r	Run with printing response message from the server.
-R	Run with printing free memory size.
-u	Run with printing CPU usage.
-d	Print installed XK-SDK's version.



XANDAR KARDIAN

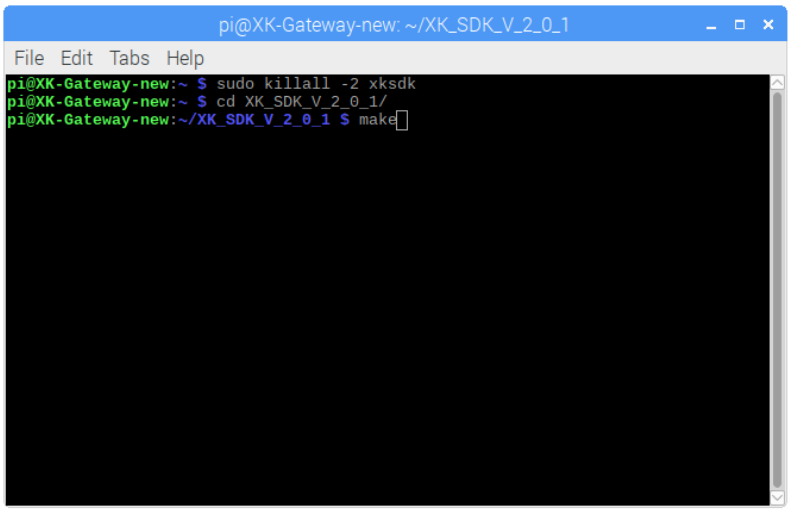
7. Make

Describes how to install the SDK for the first time or install a new version of the SDK. It build a project based on 'Makefile' in the project PATH. It can install the environment and create path, config and parameter files.

7.1. Project build

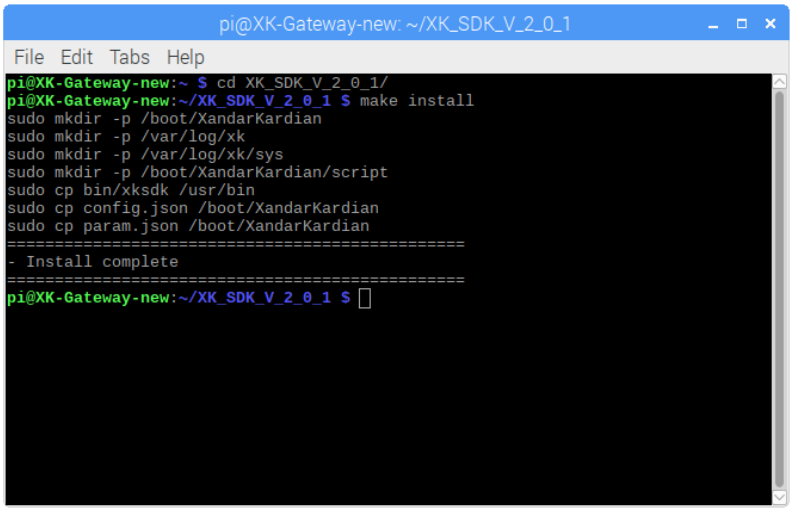
```
$ sudo killall -2 xksdk
$ cd [latest PROJECT_PATH]
$ make
```

Note: the contents of 'Makefile' may be different for each version, use the correct version's 'Makefile'. Ignore the error message for kill



7.2. Install

```
$ make install
```



7.3. Config

Modify 'config.json', 'param.json' while it is already installed and apply it to the gateway with the following command.

```
$ make config
```

7.4. Clean

```
$ make clean
```

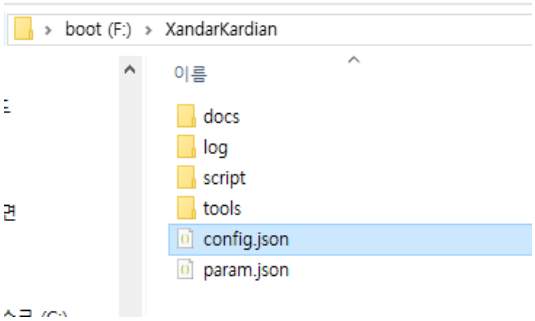
8. Parameter Size

The gateway receives the output of the algorithm from the radar. It then sends the data to the server, which can send all or part of the data received from the radar. This is because data consumption can be controlled.

8.1. Parameter configuration

Follow **5.1 SDK Configuration** 1)~5)

6) **Open** file "**param.json**" with editor program like Notepad.



7) **Change** the **parameter number** you want to receive
object ID is **application number (refer application)**
Array is a **parameter number** that can be **received**


```
{
  "1": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41],
  "2": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "3": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42],
  "4": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "5": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "6": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "7": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "8": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "9": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "10": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "11": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40],
  "12": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]
}
```

ex) Zone counting radar, only receive 1, 3, 5, 10, 20 parameters
modify like "3": [1, 3, 5, 10, 20],

8) Save.

9. Application

9.1. App number by Application

appnum	appname	appcode	Example radar ID
10	PERS		10001
20	in/out counting		20001
30	Zone		30001
31	Zone-Presence		31001
40	Presence		40001
41	Presence Vital		41001
42	Dwell time		42001
43	L3_presence		43001
50	WM Fall detection		50001
60	Foot		60001
70	Presence & Object		70001
71	Skimmer		71001
80	RHRBR		80001