

STICK-KU1/KE1 Remote Protocol

There are two ways to control the STICK, the first method uses a quick triggering message and the second more powerful method uses a button simulation.

1 - Quick Triggering

If you wish to trigger a specific scene without waiting for an answer from the STICK, use the datagram described below. All the datagrams must be sent via UDP protocol on port 2430.

UDP Datagram to trigger STICK's scenarios (12 Bytes) :

| Field | Name | Size | Description |
|-------|---------|---------|--|
| 1 | ID[8] | 8 bytes | Array of 8 characters. Value must be "Stick_U1" |
| 2 | OpCode | 2 bytes | Operation code. Value must be 6 |
| 3 | Address | 2 bytes | Scene number from 1 to 200. (8 buttons x 25 pages) |

Examples:

For triggering scene number 2 (Page A, button 2):

ASCII: **Stick_U1 6 002** → HEX: **53 74 69 63 6b 5f 55 31 06 00 02 00**

For triggering scene 20 (Page C, button 4):

ASCII: **Stick_U1 6 020** → HEX: **53 74 69 63 6b 5f 55 31 06 00 14 00**

Note that each byte is inverted because of the endianness:

20 in decimal → 14 in hexadecimal → 00 14 as a Byte → 14 00 in the datagram.

2 - Button Simulation

UDP Datagram to trigger STICK's buttons (13 Bytes).

| Field | Name | Size | Description |
|-------|--------------|---------------|--|
| 1 | ID[8] | 8 bytes | Array of 8 characters. Value must be "Stick_U1" |
| 2 | OpCode | 2 bytes | Operation code. Value must be 101 |
| 3 | SceneButton | Unsigned char | Scene number 1 → 1 Scene number 2 → 2 Scene number 3 → 4 Scene number 4 → 8 Scene number 5 → 16 Scene number 6 → 32 Scene number 7 → 64 Scene number 8 → 128 Value 0 (0x00) represents an empty value. |
| 4 | OtherButton | Unsigned char | Other buttons : Page Down → 1 Page Up → 2 Select → 4 Blackout → 8 |
| 5 | SliderButton | Unsigned char | Represents the slider's value from 0 to 100. Value 255 (0xff) represents an empty value. |

→ To build the **SceneButton**, you just need to understand that each bit of the char is linked to a button. For example:

If you want to simulate button 1, **SceneButton** will be:

1 (in binary 00000001, in hexa 01)

If you want to simulate button 2, **SceneButton** will be:

2 (in binary 00000010, in hexa 02)

If you want to simulate button 5, **SceneButton** will be:

16 (in binary 00010000, in hexa 10)

If you want to simulate button 8, **SceneButton** will be:

128 (in binary 10000000, in hexa 80)

→ To build **OtherButton**, it's the same. For example:

PageDown is 1 (0x01)

PageUp is 2 (0x02)

Select is 4 (0x04)

BlackOut is 8 (0x08)

→ The last char **SliderButton** is just the slider value between 0 and 100.

Example in C:

```
int i = 85;
```

```
Byte b = (Byte) i;
```

Below are a few examples:

Empty datagram:

ASCII: **STICK_U1 101 000 000 000** → HEX: **53 74 69 63 6B 5F 55 31 65 00 00 00 FF**

Triggering scene 5 datagram:

ASCII: **STICK_U1 101 016 000 000** → HEX: **53 74 69 63 6B 5F 55 31 65 00 10 00 FF**

Pressing Select button datagram:

ASCII: **STICK_U1 101 000 004 000** → HEX: **53 74 69 63 6B 5F 55 31 65 00 00 04 FF**

Changing slider value to 85 datagram:

ASCII: **STICK_U1 101 000 000 085** → HEX: **53 74 69 63 6B 5F 55 31 65 00 00 00 55**

All datagrams must be sent to the stick via UDP on port 2430.

Finally, you must send an empty datagram after each "button pressed" datagram so the stick understands that the "virtual finger has been lifted up from it. This allows you to simulate the multi-touch capabilities of the STICK in addition to stopping and pausing scenes. For example :

Activate audio triggering (Select + Button 7):

ASCII: **STICK_U1 101 064 004 000** → HEX: **53 74 69 63 6b 5f 55 31 65 00 40 04 ff**

Pressing buttons 5 and 8 simultaneously :

16 | 128 (in binary 10010000, in hexa 0x90)

ASCII: **STICK_U1 101 144 000 000** → HEX: **53 74 69 63 6b 5f 55 31 65 00 90 00 ff**

Stop Scene 8 :

ASCII: **STICK_U1 101 128 000 000** → HEX: **53 74 69 63 6B 5F 55 31 65 00 80 00 ff**

Wait 4 seconds

ASCII: **STICK_U1 101 000 000 000** → HEX: **53 74 69 63 6B 5F 55 31 65 00 00 00 ff**

STICK-KU1/KE1 Remote Protocol Feedback

Understanding STICK KE1/KU1 feedback messages

Each time the stick is touched or accessed remotely it broadcasts its new LED state on port 2430 using the UDP protocol.

This datagram is always 14bytes long.

Stick_U1(8bytes) Op Code(2bytes) Mode(2bytes) State(2bytes)

Op Code(2bytes) → Allows you to know which LEDs are affected by the datagram (scenes, pages, mode or slider).

Mode(2bytes) → For some very specific modes. Not useful in most cases.

State(2bytes) → LED state or slider value [0...1000].

SCENE BUTTONS (Op Code = 0x67):

Example: Button 1 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 01 0

Example: Button 2 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 02 0

Example: Button 3 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 04 0

Example: Button 4 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 08 0

Example: Button 5 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 10 0

Example: Button 6 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 20 0

Example: Button 7 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 40 0

Example: Button 8 is ON:

HEX: 53 74 69 63 6b 5f 55 31 67 0 0 0 80 0

The State is a byte mask. 0xff 0x0 means " all scene LEDs ON "

PAGE BUTTONS (Op Code = 0x68):

Example: Page 1 (A):

HEX: 53 74 69 63 6b 5f 55 31 68 0 0 0 01 0

Example: Page 2 (B):

HEX: 53 74 69 63 6b 5f 55 31 68 0 0 0 02 0

Example: Page 25:

HEX: 53 74 69 63 6b 5f 55 31 68 0 0 0 19 0

MODE BUTTON (Op Code = 0x69):

Example: Mode Dimmer:

HEX: 53 74 69 63 6b 5f 55 31 69 0 0 0 0 0

Example: Mode Speed:

HEX: 53 74 69 63 6b 5f 55 31 69 0 0 0 01 0

Example: Mode Color:

HEX: 53 74 69 63 6b 5f 55 31 69 0 0 0 02 0

SLIDER (Op Code = 0x6a):

The slider value is from 0 to 1000.

Example: Slider at 0:

HEX: 53 74 69 63 6b 5f 55 31 6a 0 01/0x02 0 0 0

Example: Slider at 50%:

HEX: 53 74 69 63 6b 5f 55 31 6a 0 01/0x02 0 f4 01

$0xf4 + (256 * 0x01) = 254 + (256 * 1) = 500$

BLACKOUT BUTTON(Op Code = 0x66/0x6b):

Example: BLACKOUT enabled:

HEX: 53 74 69 63 6b 5f 55 31 6b 0 0 0 1 0

Example: ALL LEDs OFF:

HEX: 53 74 69 63 6b 5f 55 31 6b 0 0 0 0 0

NOTE: Although this is an official release of the protocol, all the commands described are subject to change without notice. Therefore please check with us before performing a firmware upgrade.