Items

learn_ir_code

Name	Value Type	Method	Default value	Description
learn_ir_code	int	setter	0	When setting the item use a value > 0. Usually, 1. Setting this item will trigger the learning mode on the IR blaster. The learning mode is available for 30s (default on device. Currently we have no way of controlling this). The plugin polls every ~2 seconds to check if the device has learned a new code. When a new code is found, it is broadcasted to the mobile clients and the polling mechanism is stopped. During the 30 seconds the learn mode is active and until it detects a code or stops the item will have the value **1**, after which it is reverted back to **0**. This can be used to 'know' that the learning mode is active.

Example of item, learn_ir_code item example:

```
{
  "_id": "601d2686123e335cf9d19529",
  "deviceId": "60182a30123e334f5a897b3e",
  "hasGetter": false,
  "hasSetter": true,
  "name": "learn_ir_code",
  "show": true,
  "value": 0,
  "valueFormatted": "0",
  "valueType": "int"
}
```

Example of starting learning mode:

```
  {
      "method": "hub.item.value.set",
      "id": "_AUTO_429096",
      "params": {
          "_id": "601d3751123e33600adabca0",
          "value": 1
      }
  }
```

Example of UI broadcast from plugin:

```
  {
      "id": "ui_broadcast",
      "msg_id": "602bccfe123e3319727c9e1d",
      "msg_subclass": "hub.extensions.plugin.ui_broadcast",
      "result": {
          "code":
  "JgBYAAABKZURFBEUEhMSFBEUERQSExMSEjgROBI4ETgSOBE4EjgROBI4EDkSOBEUEhMRFBE
  VERQRFBITEhMSOBE4EjgROBI4EQAFHwABKEsRAAxWAAEoSxIADQU=",
          "plugin": "broadlink",
          "type": "IR"
      }
  }
```

send_ir_code

Name	Value Type	Method	Default value	Description
send_ir_code	string	setter	""	The IR code is expected to be a base64 string of the binary IR code itself. It can come from third party sources and as long as it is compatible it can be blasted. Setting this item does not result in a UI broadcast (except maybe if errors are found)

Example of item, send_ir_code item example:

```
{
"_id": "601d2686123e335cf9d1952a",
"deviceId": "60182a30123e334f5a897b3e",
"hasGetter": false,
"hasSetter": true,
"name": "send_ir_code",
"show": true,
"value": "",
"valueFormatted": "",
"valueType": "string"
}
```

Example of sending IR code:

```
  {
      "method": "hub.item.value.set",
      "id": "_AUTO_456495",
      "params": {
          "_id": "601d3a48123e33600adabca3",
          "value":
  "JgBMABQQFg8VEBQQFw4VEBQRFg8UNRY0FDUVNBY0FDUVNBczFTUUERMRFjQUEBYPFRATERc
  OFTQXMxUQFjQUNRU0FjQUAAUWAAEmSRYADQU="
      }
  }
```

Errors for send_ir_code

```
{
    "id": "ui_broadcast",
    "msg_id": "602bccfe123e3319727c9e1d",
    "msg_subclass": "hub.extensions.plugin.ui_broadcast",
    "result": {
        "code": 1,
        "data":"invalid_code",
        "message":"Expected format: base64",
        "plugin": "broadlink"
    }
}
```

```
{
    "id": "ui_broadcast",
    "msg_id": "602bccfe123e3319727c9e1d",
    "msg_subclass": "hub.extensions.plugin.ui_broadcast",
    "result": {
        "code": 1,
        "data":"invalid_code",
        "message": "Invalid code",
        "plugin": "broadlink"
    }
}
```

```
{
    "id": "ui_broadcast",
    "msg_id": "602bccfe123e3319727c9e1d",
    "msg_subclass": "hub.extensions.plugin.ui_broadcast",
    "result": {
        "code": 2,
        "data":"send_ir_code_error",
        "message": "An error occurred when transmitting the IR code.",
        "plugin": "broadlink"
    }
}
```

sweep_rf

Name	Value Type	Method	Default value	Description
sweep_rf	int	setter	0	There are devices that operate either on 315 MHz or 433 MHz. So this will help the device 'detect' what frequency your remote is using before you can learn any RF codes. Set this to 1 in order to start sweeping for frequencies. Set it to 0 to cancel sweeping. If the Broadlink device has locked onto a frequency a UI broadcast will be sent (see below) and this item will automatically switch back to 0. This will switch automatically to 0 if the device has not locked onto a frequency and no UI broadcast will be sent in this case. Usual flow is: 1. Set 'sweep_rf' to 1 2. Keep your RF enabled remote's button pressed for 3 to 6 seconds. Repeat if nothing changes. 3. If the device gives up, the plugin will detect this and abort automatically, setting 'sweep_rf' to 0 4. If the device locks onto a frequency the plugin will broadcast the below "sweep_rf success broadcast" message and will set 'sweep_rf' to 0 5. If the user chooses to cancel sweeping, set 'sweep_rf' to 0

sweep_rf success broadcast

```
{
    "id": "ui_broadcast",
    "msg_id": "6077fe1a123e3312853978f3",
    "msg_subclass": "hub.extensions.plugin.ui_broadcast",
    "result": {
        "found": true,
        "plugin": "broadlink",
        "type": "sweep"
    }
}
```

learn_rf_code

IMPORTANT NOTE: For learning an RF code you must first do a frequency swipe at least once per each device whose buttons you want to learn. Otherwise RF learning won't start even if the plugin says it did. (We have no way of distinguishing this from a plugin).

Name	Value Type	Method	Default value	Description
learn_rf_code	int	setter	0	When setting the item use a value > 0. Usually, 1. Setting this item will trigger the RF learning mode on the Broadlink device. The learning mode is available for 30s (default on device. Currently we have no way of controlling this). The plugin polls every ~2.5 seconds to check if the device has learned a new code. When a new code is found, it is broadcasted to the mobile clients and the polling mechanism is stopped. During the 30 seconds the learn mode is active and until it detects a code or stops the item will have the value **1**, after which it is reverted back to **0**. This can be used to 'know' that the learning mode is active.

Example of item, learn_ir_code item example:

```
{
"_id": "601d2686123e335cf9d19529",
"deviceId": "60182a30123e334f5a897b3e",
"hasGetter": false,
"hasSetter": true,
"name": "learn_rf_code",
"show": true,
"value": 0,
"valueFormatted": "0",
"valueType": "int"
}
```

Example of starting RF learning mode:

```
{
    "method": "hub.item.value.set",
    "id": "_AUTO_429096",
    "params": {
        "_id": "601d3751123e33600adabca0",
        "value": 1
    }
}
```

Example of UI broadcast from plugin:

```
{
    "id": "ui_broadcast",
    "msg_id": "602bccfe123e3319727c9e1d",
    "msg_subclass": "hub.extensions.plugin.ui_broadcast",
    "result": {
        "code":
"sgB6AAUEBQQFBAUEBAQFBAUEBQQFBAQECggFBAUECQgFBAUECQQFBAUEBQgJBAUEBQgFBAU
EBQQJBAUEBQQEBAUEBQkEBAkFBAQFBAUEBQQECQkEBQQFBAQJBQQJCAUECQQFCQQFBAQFBAU
ECQQFBAUIBQQJCAYECgQFAAXcAAAAAAAAAAAAAAAAAAA=",
        "plugin": "broadlink",
        "type": "RF"
    }
}
```

send_rf_code

Name	Value Type	Method	Default value	Description
send_rf_code	string	setter	""	The RF code is expected to be a base64 string of the binary RF code itself. It can come from third party sources and as long as it is compatible it can be blasted. Setting this item does not result in a UI broadcast (except maybe if errors are found)

Example of item, send_ir_code item example:

{

"_id": "601d2686123e335cf9d1952a",

"deviceId": "60182a30123e334f5a897b3e",

"hasGetter": false,

"hasSetter": true,

"name": "send_rf_code",

"show": true,

"value": "",

"valueFormatted": "",

"valueType": "string"

}

Example of sending IR code:

```
{
    "method": "hub.item.value.set",
    "id": "_AUTO_456495",
    "params": {
        "_id": "601d3a48123e33600adabca3",
        "value":
"sgB6AAUEBQQFBAUEBAQFBAUEBQQFBAQECggFBAUECQgFBAUECQQFBAUEBQgJBAUEBQgFBAU
EBQQJBAUEBQQEBAUEBQkEBAkFBAQFBAUEBQQECQkEBQQFBAQJBQQJCAUECQQFCQQFBAQFBAU
ECQQFBAUIBQQJCAYECgQFAAXcAAAAAAAAAAAAAAAAAA="
    }
}
```

Errors for send_rf_code

All errors from "send_ir_code" above and additionally:

```
{
    "id": "ui_broadcast",
    "msg_id": "602bccfe123e3319727c9e1d",
    "msg_subclass": "hub.extensions.plugin.ui_broadcast",
    "result": {
        "code": 3,
        "data":"send_rf_code_error",
        "message": "An error occurred when transmitting the RF code.",
        "plugin": "broadlink"
    }
}
```

Device Settings

Setting	Description	Example

lock_unlock	Enables the user to lock or unlock the IR blaster.
	
	When a device is locked the plugin cannot obtain an authentication key and thus cannot communicate further with the device.
	
	The device status is also updated to "locked" when using this setting to lock the device.
	
	The device will continue to be operational if it is locked from our app (and previously it was unlocked) since at that point we've already authenticated with the device.
		`{`
		` "_id": "60509c54123e331726f45893",`
		` "description": {`
		` "text": "When the device is locked, other users in the same WLAN will be unable to find the device for enhanced security."`
		` },`
		` "deviceId": "60509c54123e331726f45890",`
		` "hasSetter": true,`
		` "label": {`
		` "text": "Lock device"`
		` },`
		` "status": "synced",`
		` "value": false,`
		` "valueType": "bool"`
		`}`