



User Guide

MS-0401E

4-way HDBaseT / IP Gateway



Contents

Introduction	03
Key Features	03
Connectivity Overview (front & back)	04
Single Input to Multiple Outputs	05
Multiple Inputs to Single Output	05
Cascading	06
Cabling for HDBaseT	06
Power	06
Control Tools	07
UART RS232 User Interface	08
Web-based User Interface	11
JSON Interface	12
Firmware Upgrade	13
Web Interface	14
Specifications & Package Contents	19
Example Schematics	20
Appendix	22

Notices

- This MSolutions product contains electrical components that may be damaged by electrical spikes, surges, electric shock, lightning strikes, etc. Use of surge protection is highly recommended in order to protect and extend the life of your equipment.
- The transmission distances of HDMI over CAT cables are measured using TE CONNECTIVITY 1427071-6 EIA/TIA-568-B termination (T568B) of cables is recommended for optimal performance. To minimise interference of unshielded twisted pairs in the CAT5e/6 cable, do not run the HDBaseT / CAT5e/6/6a cabling with or in close parallel proximity to mains power cables.
- Do not substitute or use any other power supply other than the enclosed unit, or an MSolutions approved replacement. Do not disassemble either the Transmitter or Receiver units for any reason. Doing so will void the manufacturer's warranty.
- MSolutions reserves the right to change the specifications of this unit without prior notice. As a result of this, physical representations or graphical elements contained within this user guide may not be accurate.

Introduction

The MSolutions 4-way HDBaseT / IP Gateway is a small form-factor unit with 5 x HDBaseT / IP ports, offering an efficient HDBaseT / IP switching platform.

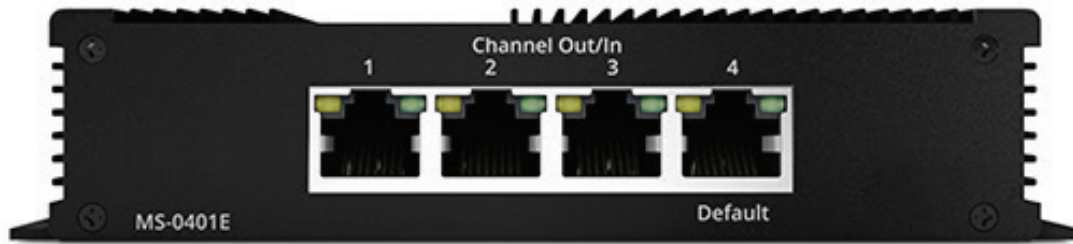
The unit is designed to offer two differing types of HDBaseT / IP switching, allowing for up to 4 x HDBaseT source devices (Transmitters) or IP Hosts to be linked through the MSolutions Gateway to a single HDBaseT / IP output, or for a single HDBaseT / IP input to be independently switched to the 4x HDBaseT / IP outputs.

The unit can be used with any manufacturers compatible HDBaseT transmitter and receiver equipment to provide extra connectivity to larger distribution systems, or for separation of multiple units via LAN, to provide a simple, yet cost-effective switching solution.

Key Features

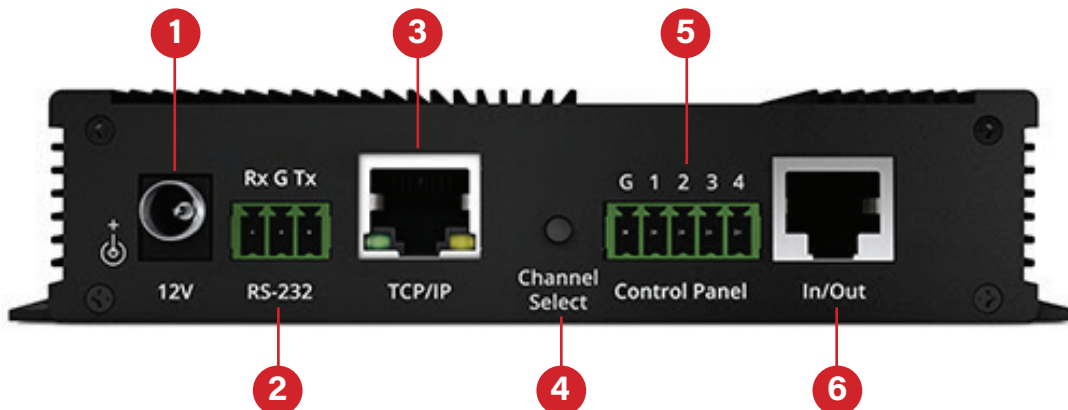
- Inline switching of HDBaseT Transmitters or Receivers: 1 x input to 4 x selectable outputs, or 4 x independent inputs to 1 x output
- Supports multiple control interfaces, including built-in web UI, UART RS232, 4 buttons keypad, and JSON interface.
- Supports multiple protocols, including TCP/IP, UDP, HTTP, and MQTT.
- Low power consumption.
- Easy configuration and management through a web-based user interface.
- Supports firmware upgrade using UART interface.
- Can be used for dynamic routing of LAN connections (outside of HDBaseT spec)
- Interoperability with all HDBaseT manufacturers equipment
- Distances of manufacturers HDBaseT equipment maintained with switch inline
- Support for all HDBaseT specifications up to HDBaseT Spec 3.0 including video, audio, audio return, control and Ethernet (based on spec of individual TX or RX attached)
- Transparent to HDCP content being sent via HDBaseT - pass-through is based on the specification of the HDBaseT transmitter and receiver modules attached
- Supports pass-through of all HDBaseT power formats: PoE, PoH, PoC up to 100W
- Internal galvanic separation of inputs / outputs to ensure disconnect when ports are not in use

Connectivity Overview - front



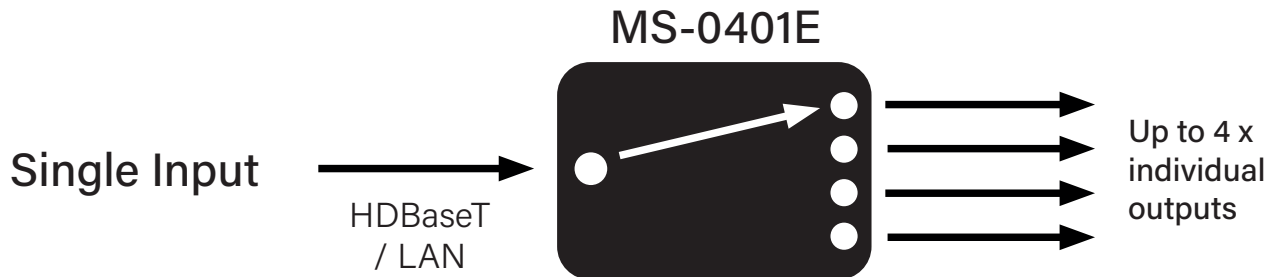
- HDBaseT input / output ports (x 4) - connect to CAT cable to transmitter/s or receiver/s

Connectivity Overview - back



1. Power connector – connect to supplied 12V 2A power supply
2. 3-pin Phoenix connector
3. TCP/IP
4. Channel Select
5. Control – 5-pin Phoenix connector
6. HDBaseT Input / Output – connect to CAT cable to transmitter or receiver

Single Input to Multiple Outputs



The MS-0401E can be used to transfer the signal from one point of transmission to one of the 4 x outputs. When there is no power to the MS-0401E, port 1 is automatically selected as a static pass-through.

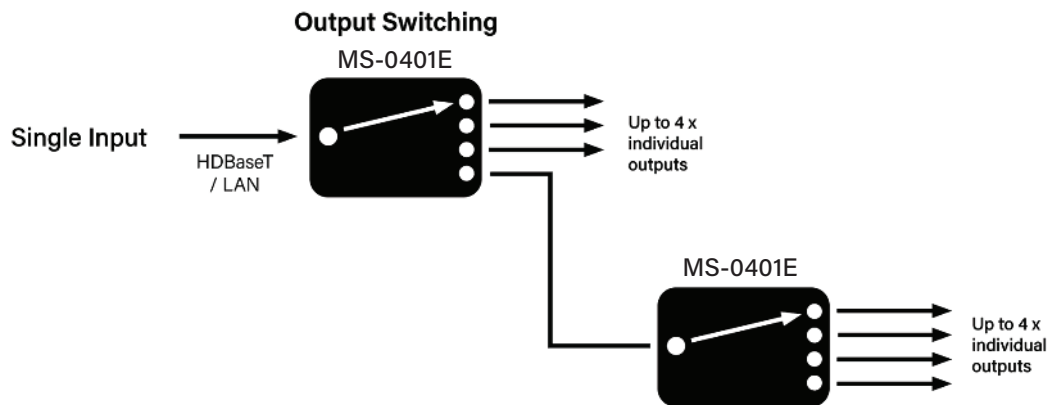
Please note: the input is only ever routed to one output in this configuration. The unit does not act as a splitter connecting all outputs of the input simultaneously.

Multiple Inputs to Single Output



The MS-0401E can also be used to route any of up to 4 x input signals to the output.

Cascading



Networks and HDBaseT connectivity can be expanded using multiple MS-0401E units inline between point of transmission to receive. Adding in a subsequent MS-0401E unit will provide connectivity to up to 7 x displays from a single source. Cascading can also work for extra inputs.

Cabling for HDBaseT

It is important that the interconnecting CAT cable between the MSolutions HDBaseT switch and the HDBaseT / LAN equipment is terminated using the correct RJ45 pin configuration. The link CAT cable must be a 'straight' (pin-to-pin) CAT cable and it is advised that this is wired to the T568B wiring standard as this format is less prone to EMI (Electro-Magnetic Interference).

When installing CAT cables it is advised that the best possible CAT cable quality possible is used. HDMI distribution products will only work if used with CAT5e standard cable or above. MSolutions recommends using a CAT6 (or higher) cable for installations, especially when running over longer distances, in areas of high EMI, or with 4K signal distribution.

For HDBaseT and HDMI testing capabilities, please refer to the MSolutions MS-TestPro to prove CAT cables can be used for video signal distribution.

Power

The MS-0401E allows for all HDBaseT and PoE power schemes to pass-through the unit the same as if a single CAT cable were to be connected between the transmission equipment were connected to the remote equipment. The MS-0401E does not use the PoE, or power contained within the signal for its own switching or power processes.

Control Tools

Four MS-0401E Interfaces

1 **UART RS232**
for firmware upgrade and debugging

2 **Web-based UI**
DHCP UI for configuration and management

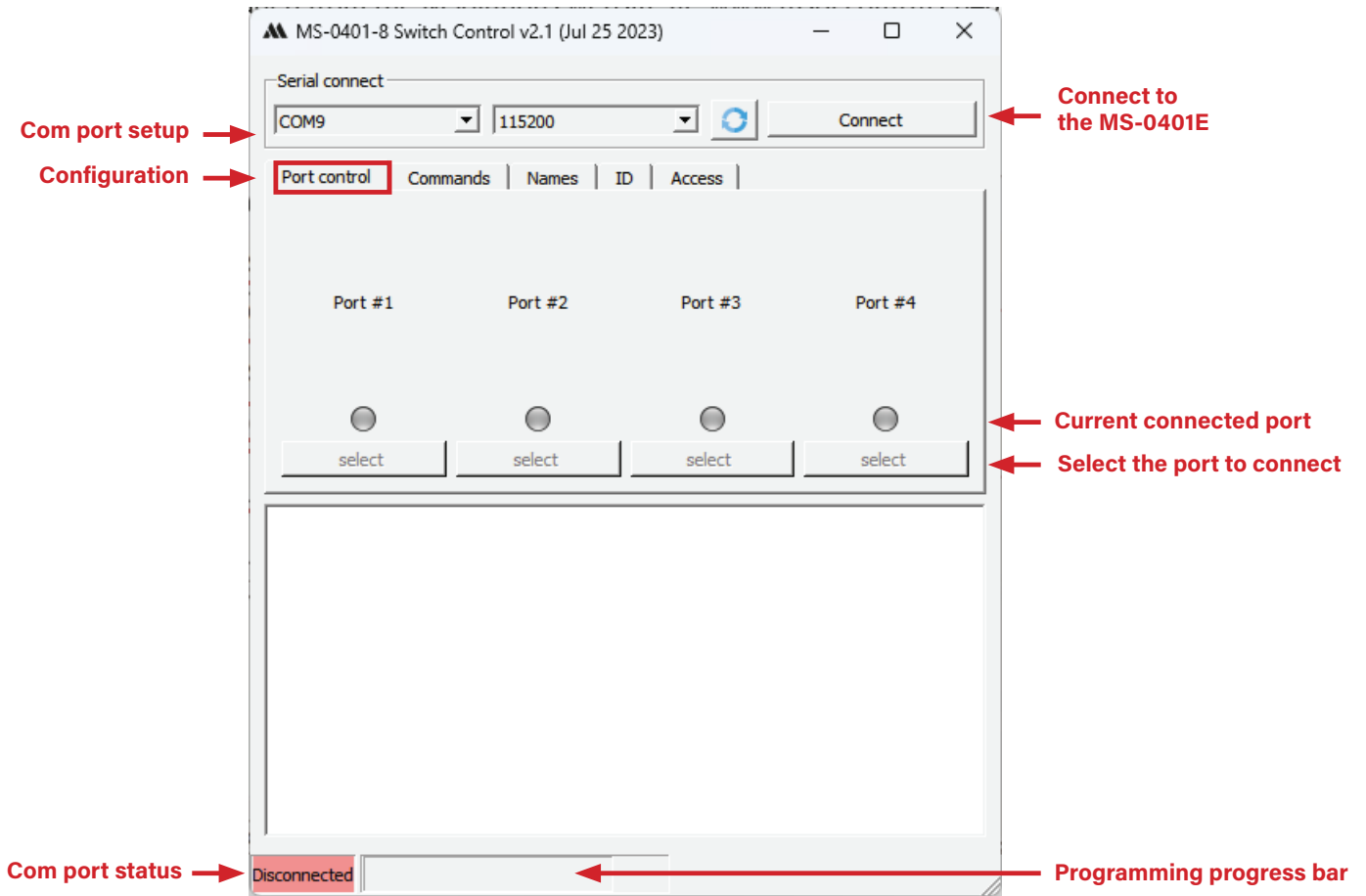
3 **JSON Interface**
for integration with other applications

4 **Keypad**

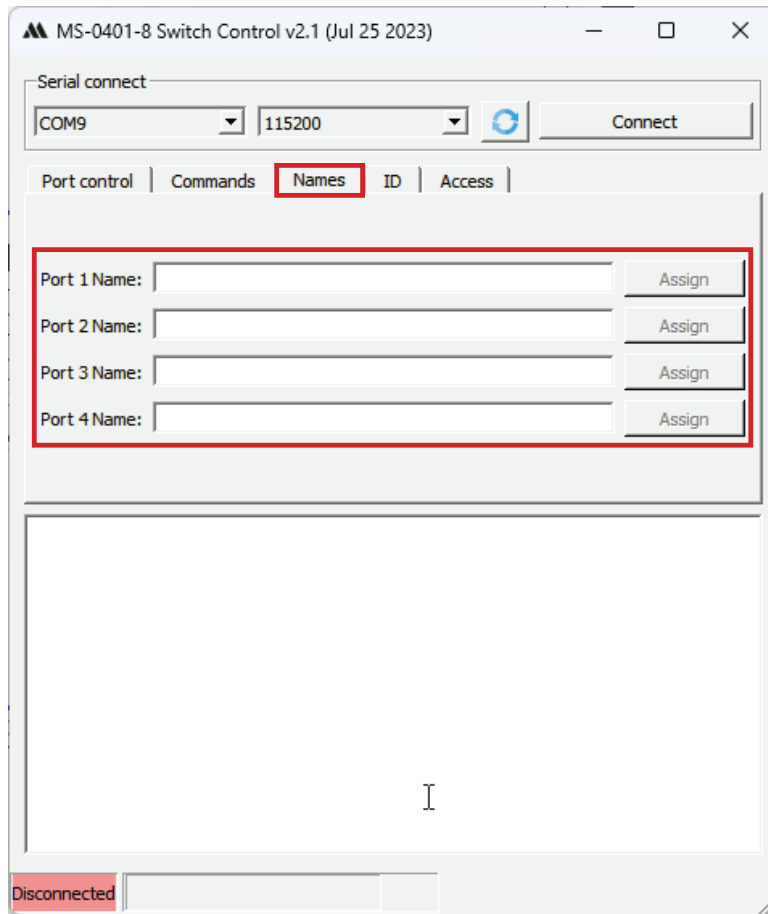
1. UART RS232 User Interface for firmware upgrade and debugging

MSolutions provides an RS232 API interface so that third party controllers can control the MS0401E. The app can be downloaded from the MSolutions website at: www.msol.com/ms-0401e

MS-0401E Application:

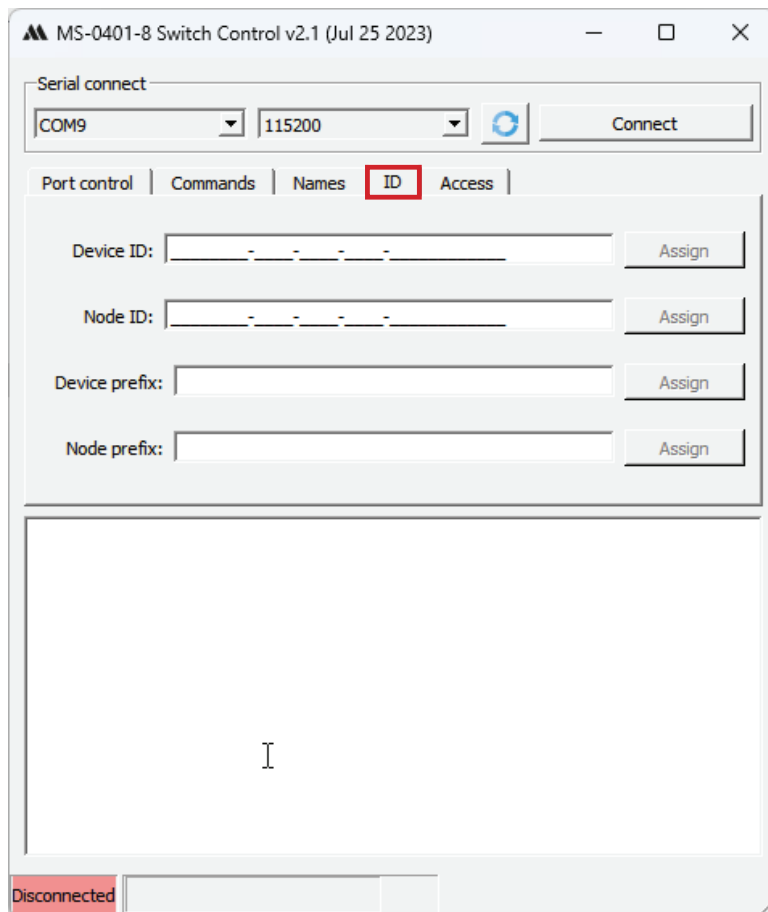


Port labeling



← Assign port name

IPMX interface configuration



← IPMX interface Conduct IP interface

Third Party Controller:

Switch RS232 Command List

COMMAND	DESCRIPTION
RR	Reply: current state: "R1", "R2" ... etc.
01	Set new switch state (no reply)
02	
03	
04	
ip	Reply: network settings and state
er	Reset network settings to default Reply: "Settings have been reset to default"
b	Reply: firmware version
bt	Start bootloader mode

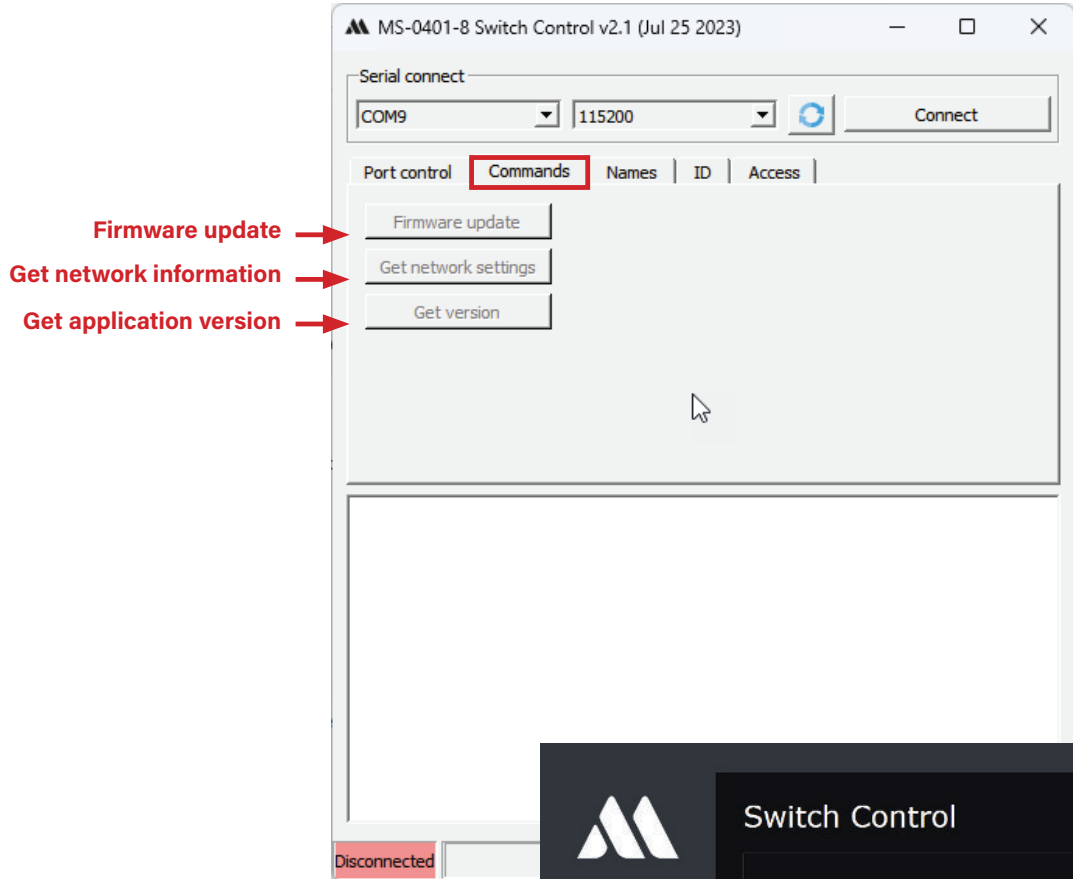
Note: Don't append CR, LF to the command.

Examples for Switching Ports

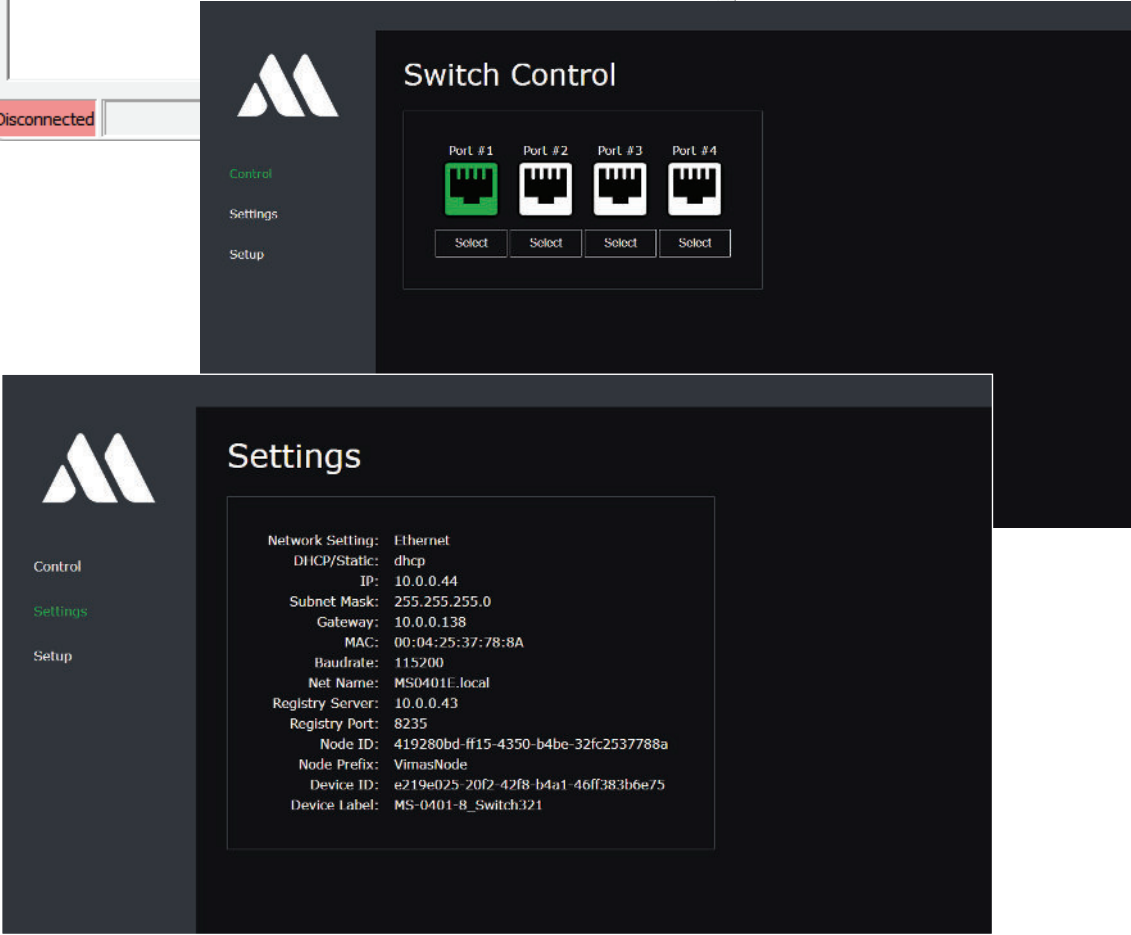
FUNCTION	COMMAND
Switch to port 1	4F 31 0A 52 52 0A
Switch to port 2	4F 32 0A 52 52 0A
Switch to port 3	4F 33 0A 52 52 0A
Switch to port 4	4F 34 0A 52 52 0A

2. Web-based UI DHCP UI for configuration and management

To get to web-based screen, open Chrome and type in the IP address of the unit. You can find the IP address by following these five steps:



1. Connect the MS-0401E Gateway to a PC using a UART-to-USB cable.
2. Open the Switch control App.
3. Power on the Gateway.
4. Select the appropriate COM port and press connect.
5. Press "Get network settings" to identify the unit IP.



3. JSON Interface for integration with other applications

Compatible with popular development environments, such as Arduino, Raspberry Pi, and Python. MSolutions provides a JSON command API to control the system via a third party control system.

TCP IP Commands:

JSON COMMAND	RESPONSE	DESCRIPTION
<pre>{ select:1 }</pre>	No response Request report	Command to select port 1
<pre>{ select:1 }</pre>	<pre>{ "dhcp": true, "IP": "10.0.0.34", "Mask": "255.255.255.0", "Gateway": "0.0.0.0", "MAC": "00:04:AA:AA:AA:AA", "Baudrate": 19200, "CurrentPort": 1 }</pre>	Request report

Python Example Code:

SELECT PORT

```
import socket
import time

conn = socket.socket()
conn.connect(("10.0.0.61",9760))

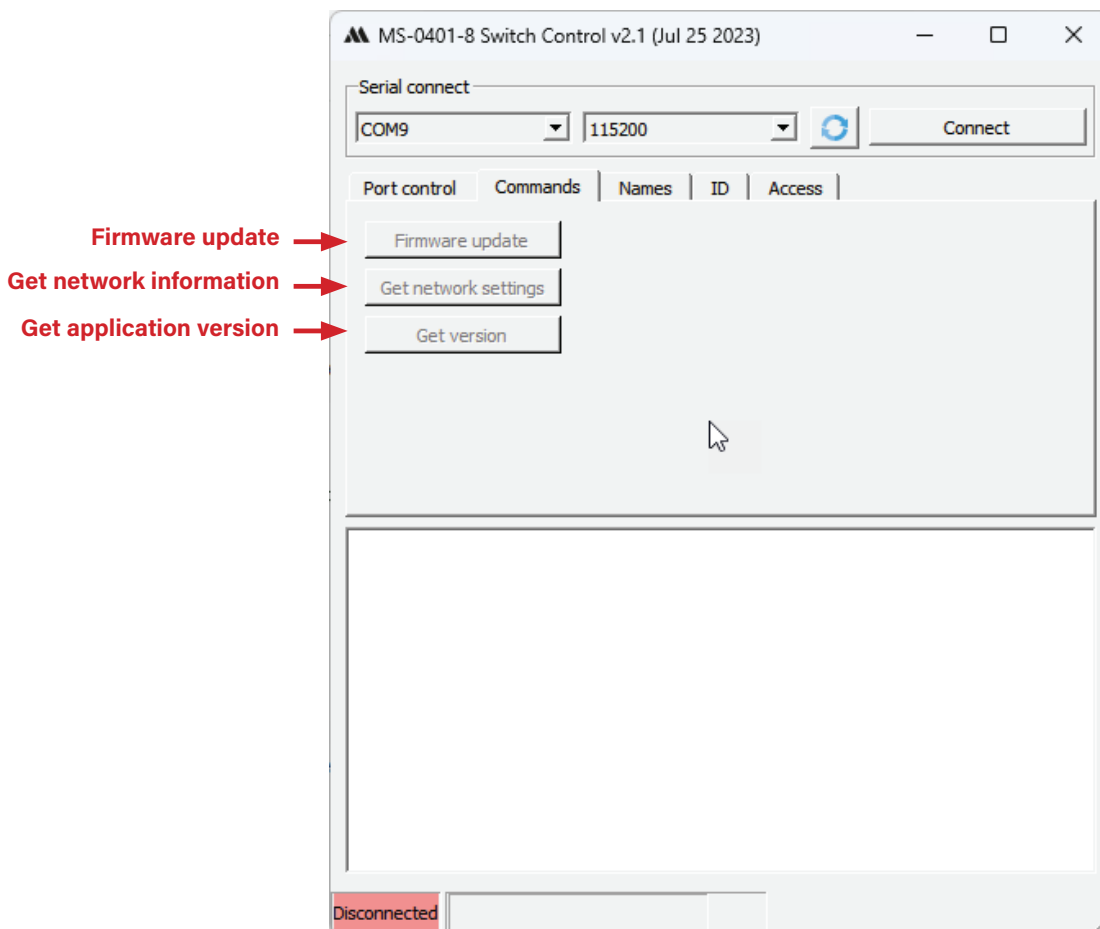
i = 1
while(True):
    i = i+1
    if i>4:
        i=1
    a = '{"select":'+str(i)+'}'
    print(a + '\n')
    conn.send(a.encode('utf-8'))
    time.sleep(0.1)
```

Firmware Upgrade

Procedure for Firmware Upgrade Using UART Interface:

To upgrade the firmware using the UART interface, follow the steps below:

1. Connect the MS-0401E Gateway to a PC using a UART-to-USB cable.
2. Open the Switch control App.
3. Power on the Gateway.
4. Select the appropriate COM port and press Connect.
5. Press Firmware Update and select the upgrade bin file.

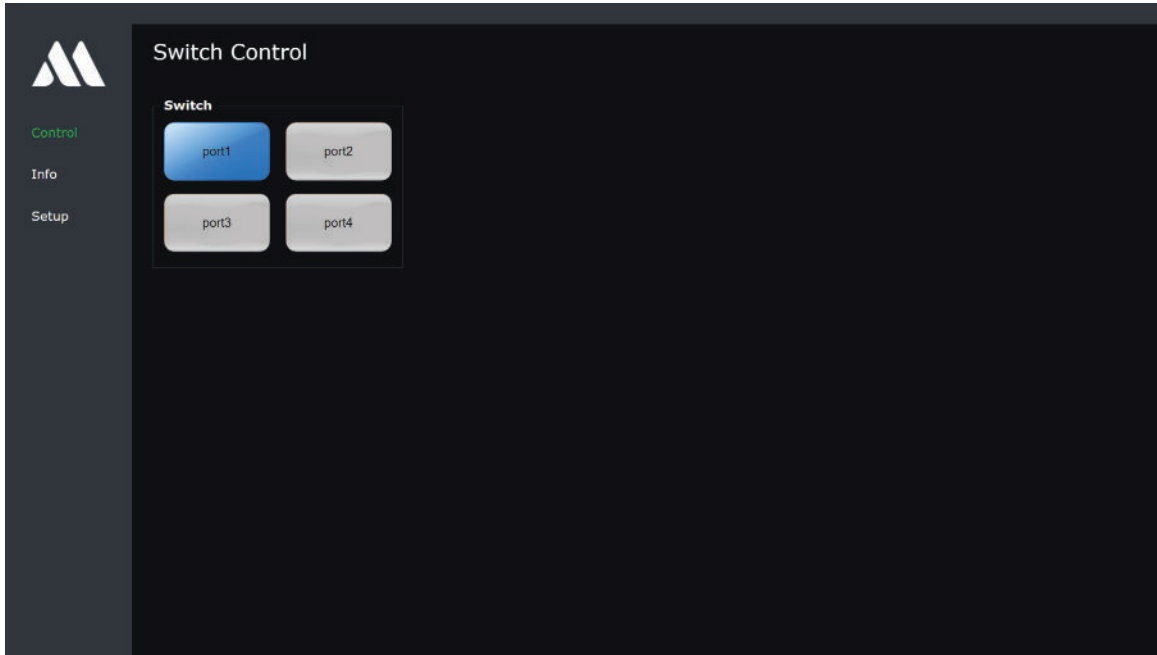


At that point download will start automatically. Please take no further action until the firmware upgrade process is complete, which may take about 15 seconds. The Gateway will automatically restart after the firmware upgrade process is complete.

Note: Ensure that the firmware file is compatible with the MS-0401E Gateway before upgrading the firmware. Incorrect firmware files can cause the device to malfunction.

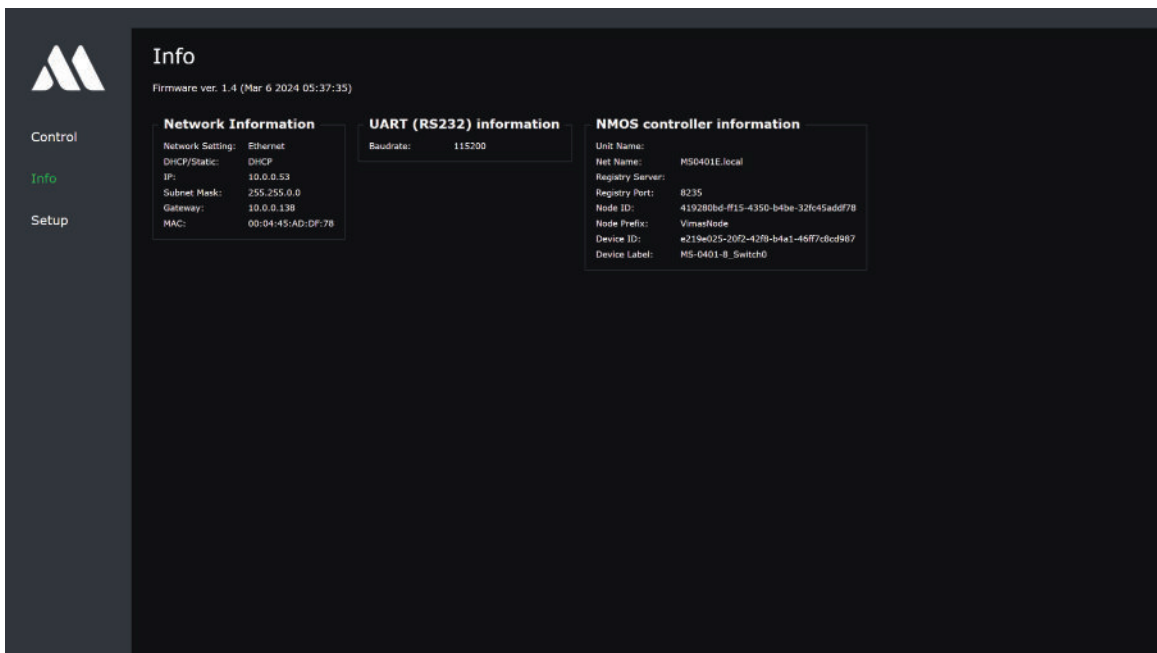
Web Interface

To access the web interface, simply open a browser and input the IP address of the MS-0401E. The first tab when accessing the web interface is the Switch Control page. On this page, the user can control the switching functions of the MS-0401E.



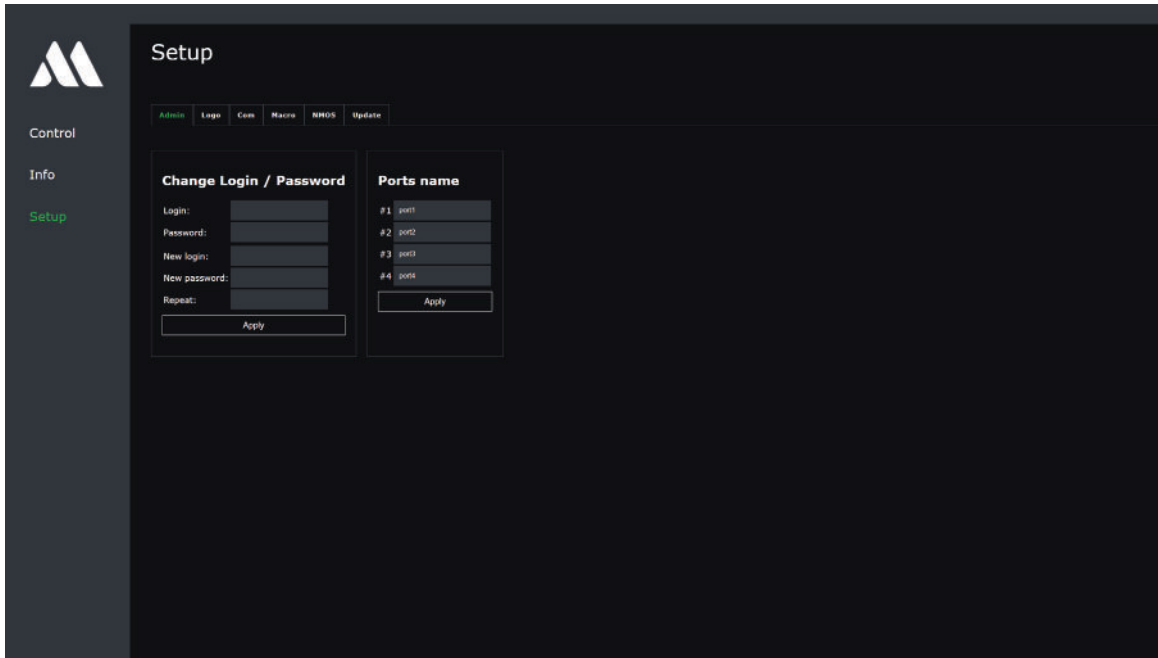
The 2nd tab is the Info page:

This page displays all the relevant information on the device, such as: IP address, FW version, name, etc.



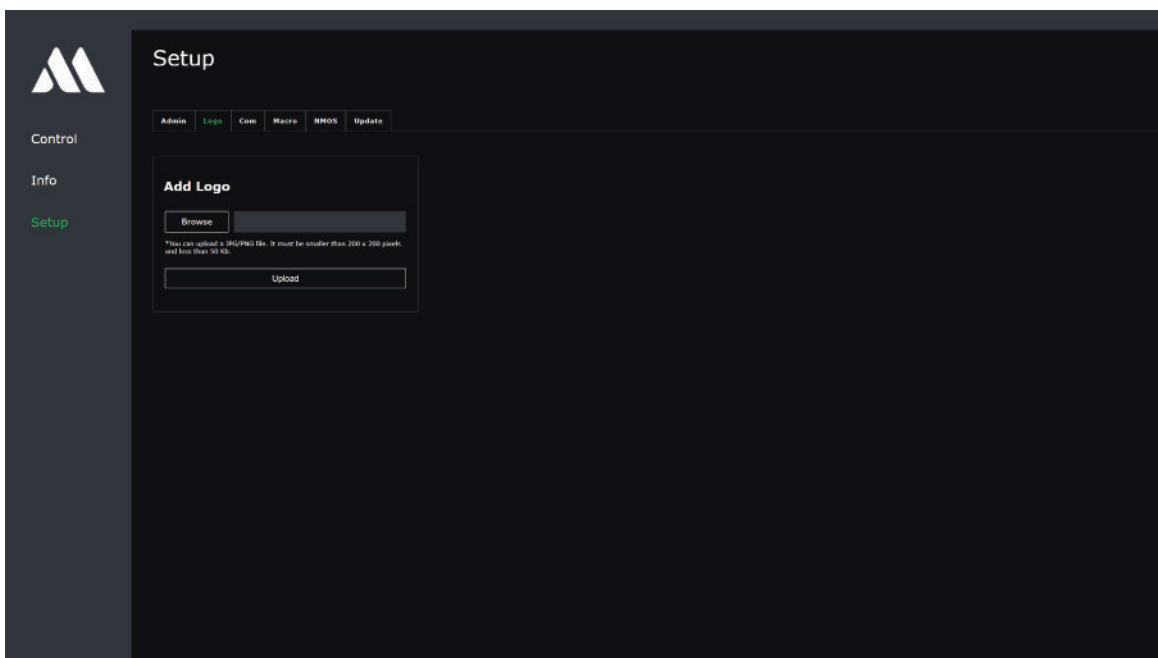
In the 3rd tab are the Setup pages:

The first subtab is the Admin tab. The user can set a new login user and password on this tab. Additionally, the user can change the switching button's names to suit the solution's implementation.



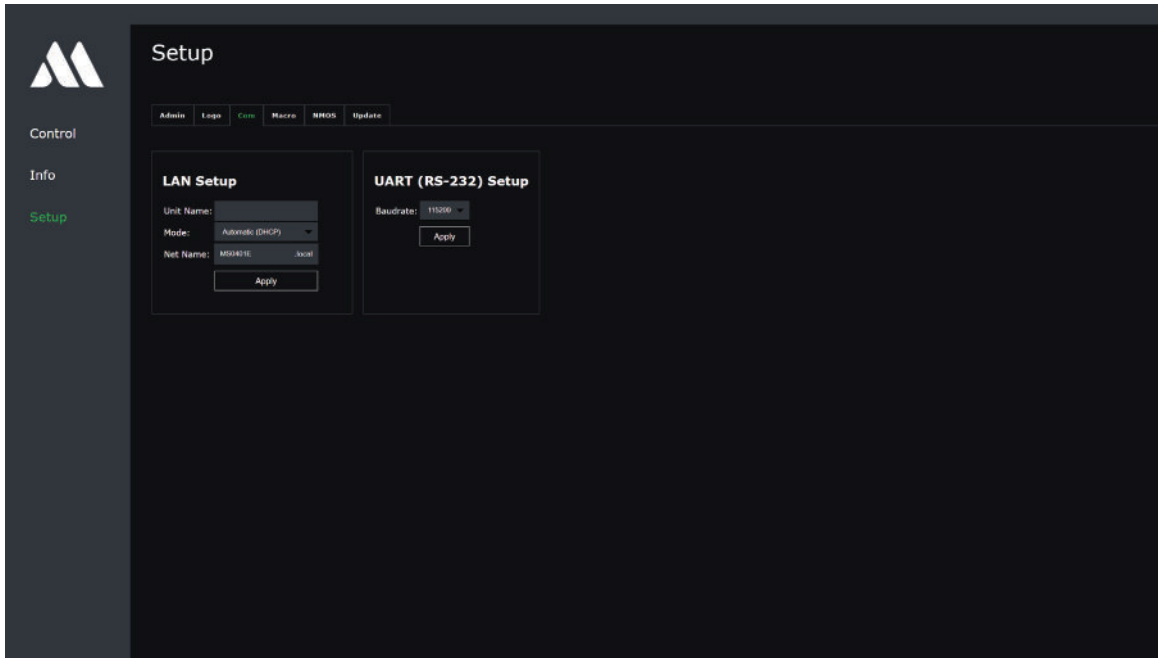
The next subtab is the Logo tab:

On this tab, the user can upload their logo. It needs to be up to 200x200 pixels and smaller than 50Kb.



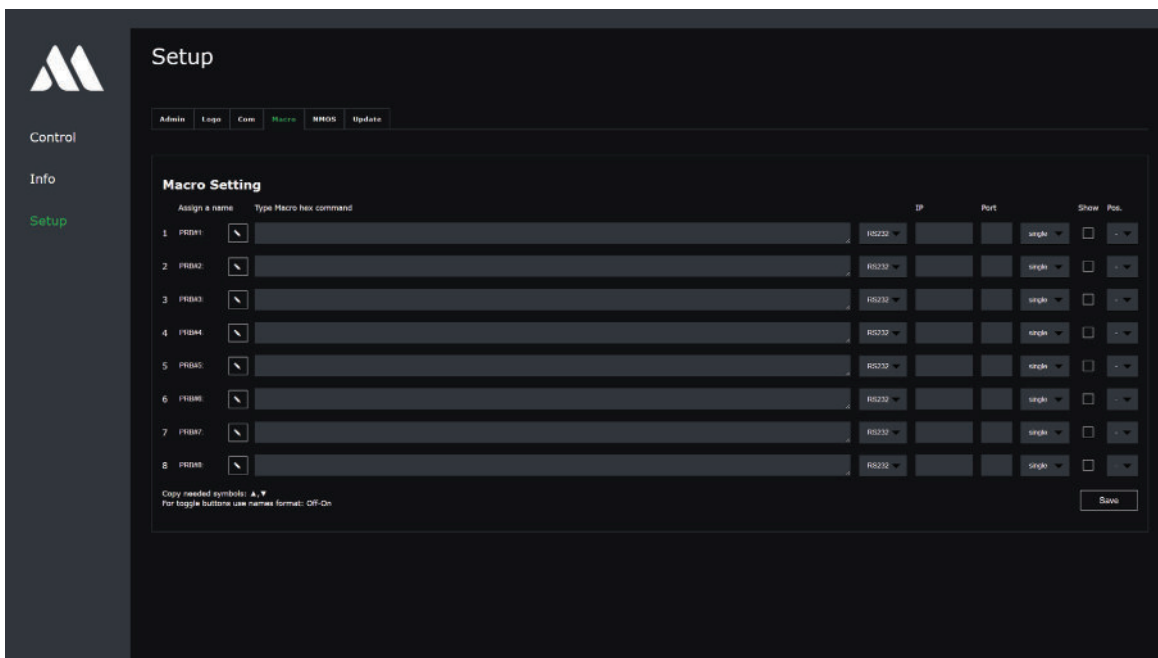
The Com subtab:

On this subtab, the user can change the communication parameters of the MS-0401E, both IP and serial.



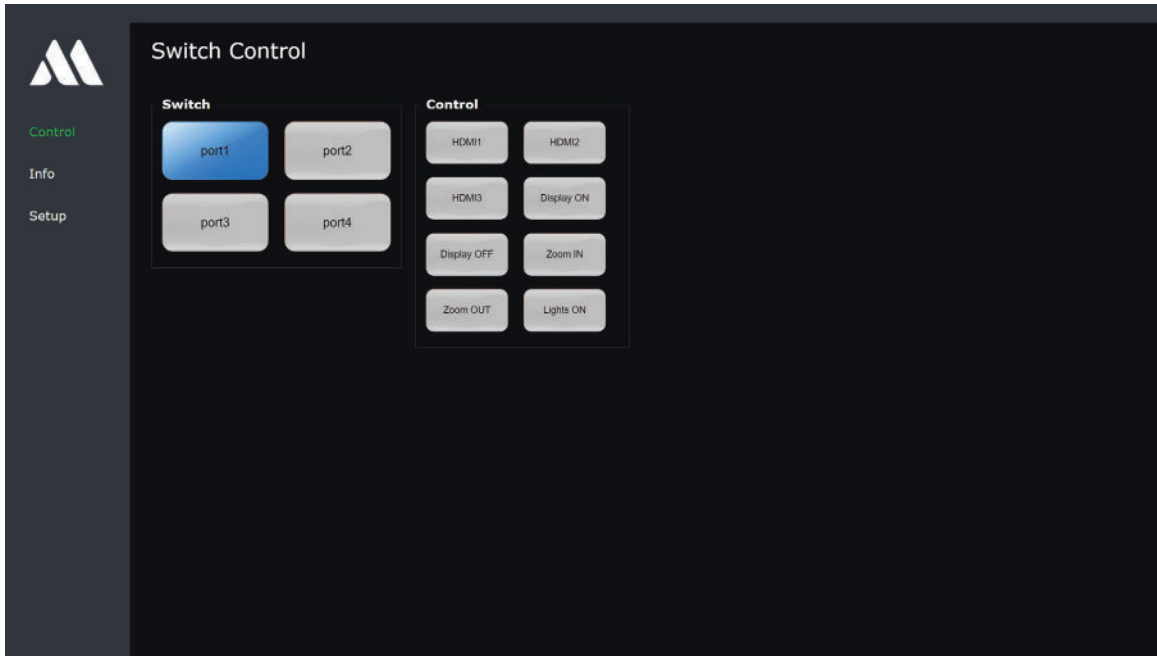
The Macro subtab:

This subtab contains all the parameters to set the IP control for any 3rd party devices. Here the user can create commands for any 3rd party device with an open API for IP control. Simply input the hex command, IP address, and logical port and assign a button to the command. After enabling the button ("Show"), the buttons will appear on the main page under control.



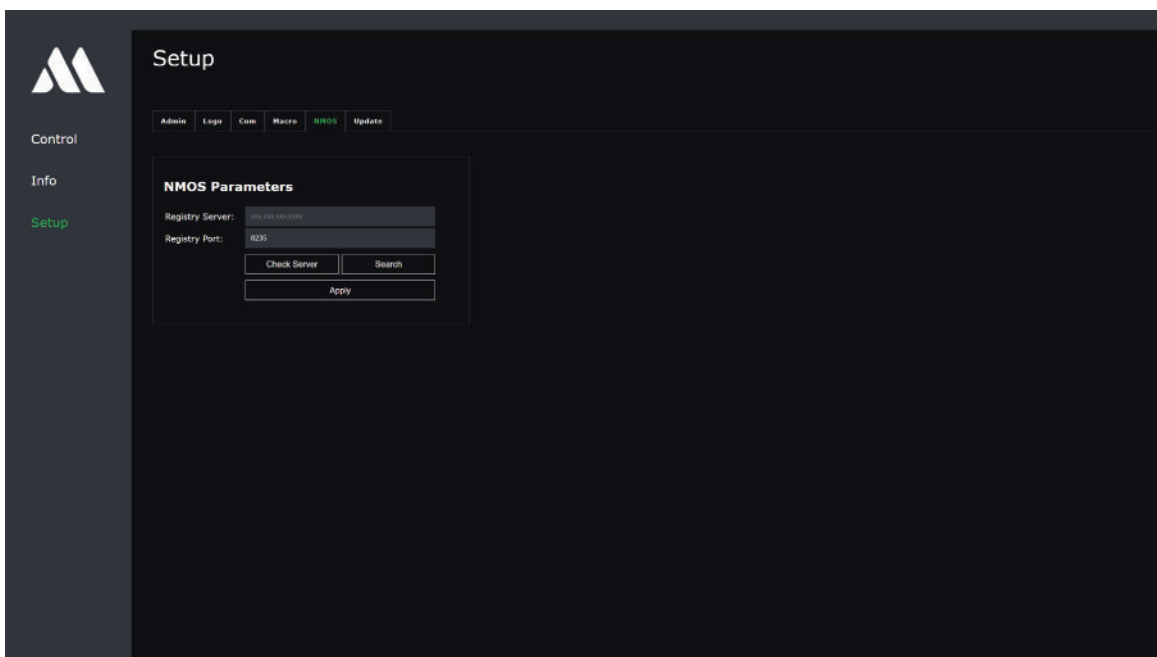
Switch Control:

After clicking save, the main display will show the buttons created.



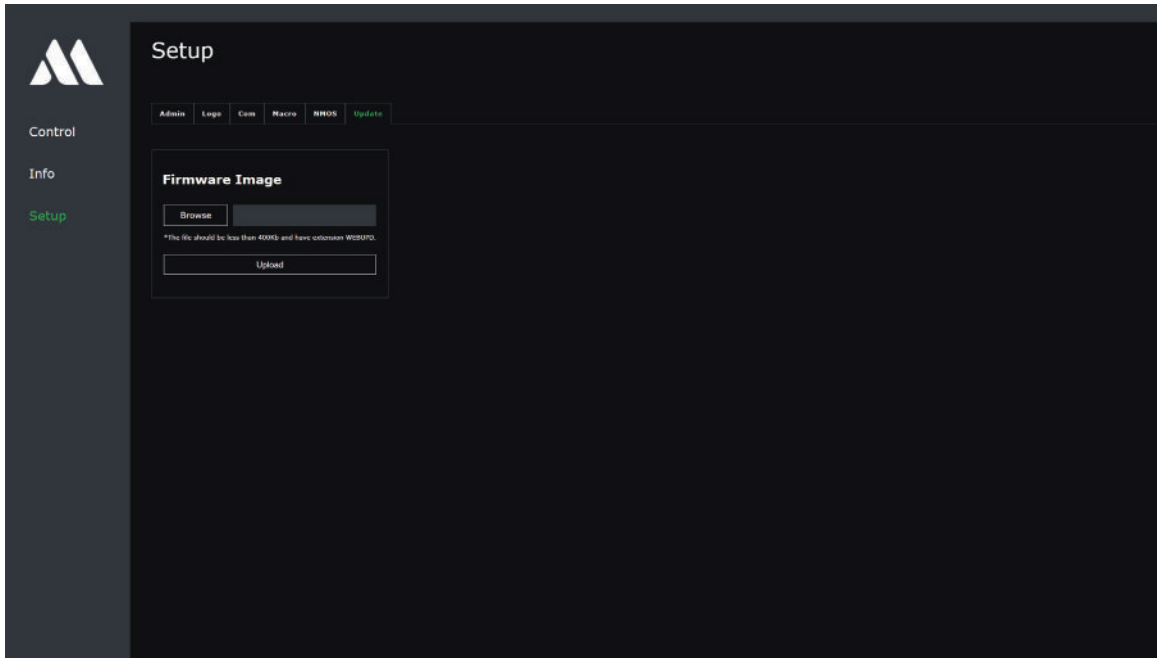
The NMOS subtab:

On this subtab, the user can integrate the NMOS server.



Update subtab:

On this subtab, the user can update the FW of the MS-0401E.



Specifications

- RJ45 connectivity (HDBaseT or LAN): 5x RJ45 - 1:4 or 4:1
- RS-232 connectivity: 1 x 3-pin Phoenix serial - serial to USB cable included
- Power supply: 12V 2A DC
- Power consumption: 7.5W
- Individual unit dimensions (W x D x H): 85 x 56 x 25mm
- Individual unit weight: 0.2Kg
- Operating temperature: 32°F to 104°F (0°C to 40°C)
- Storage temperature: -4°F to 140°F (-20°C to 60°C)
- Operating humidity: 10-90% non condensing

Package Contents

- 1 x MS-0401E Switch
- 1 x 12V 2A power supply with US, UK & EU territory clips
- 1 x 3-pin Phoenix connector
- 1 x 5-pin Phoenix connector

Extender Compatibility

All MSolutions extenders are tested and compatible with the 0401E.
These include the following:

USB-C/USB3.0 Compatible Extenders:

MS-6U1CP, MS-6U1C, MS-6U41A

HDBaseT™ KVM Extenders:

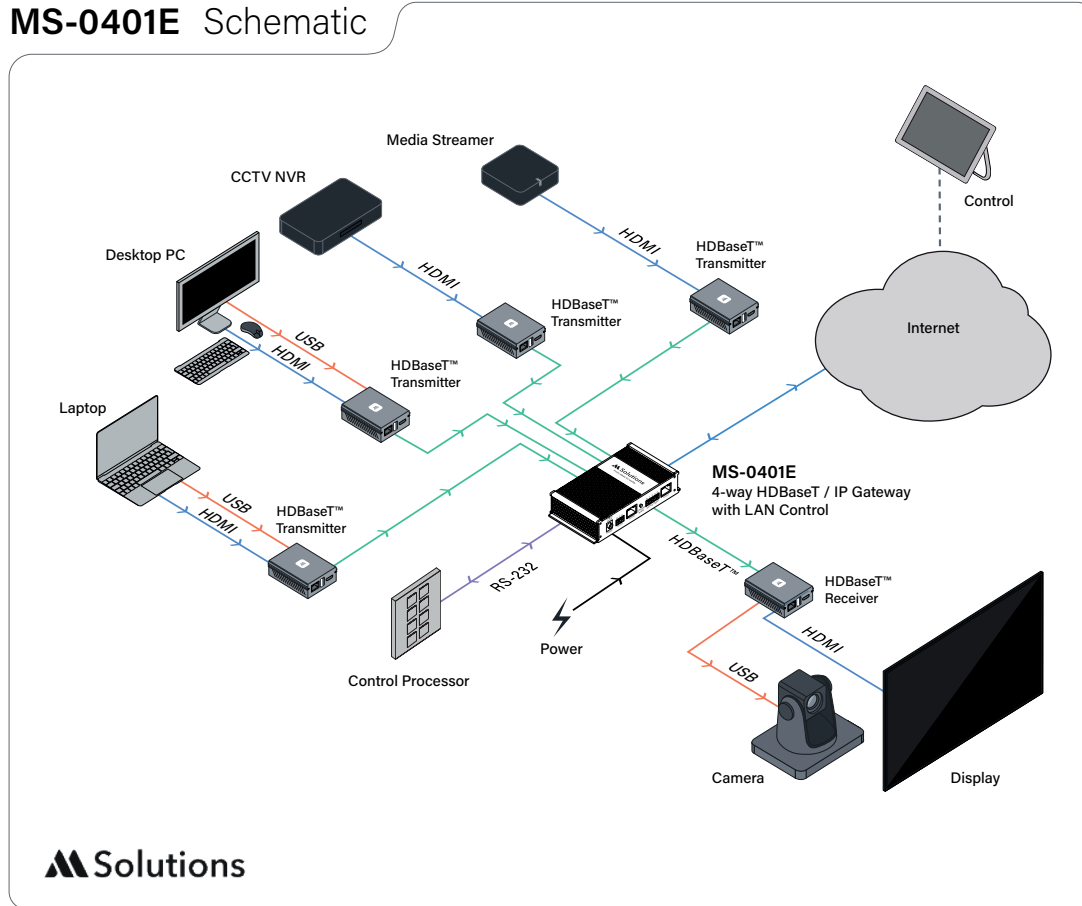
MS-210U2R, MS-310U2R, MS-210U4P

HDBaseT™ Extenders:

MS-310U1R XR, MS-3RU, MS-210U6S, MS-070PRI, MS-100PRI, MS-070SP, MS-210SP, MS-3R

Example Schematics

MS-0401E Schematic



MS-0401E
Front



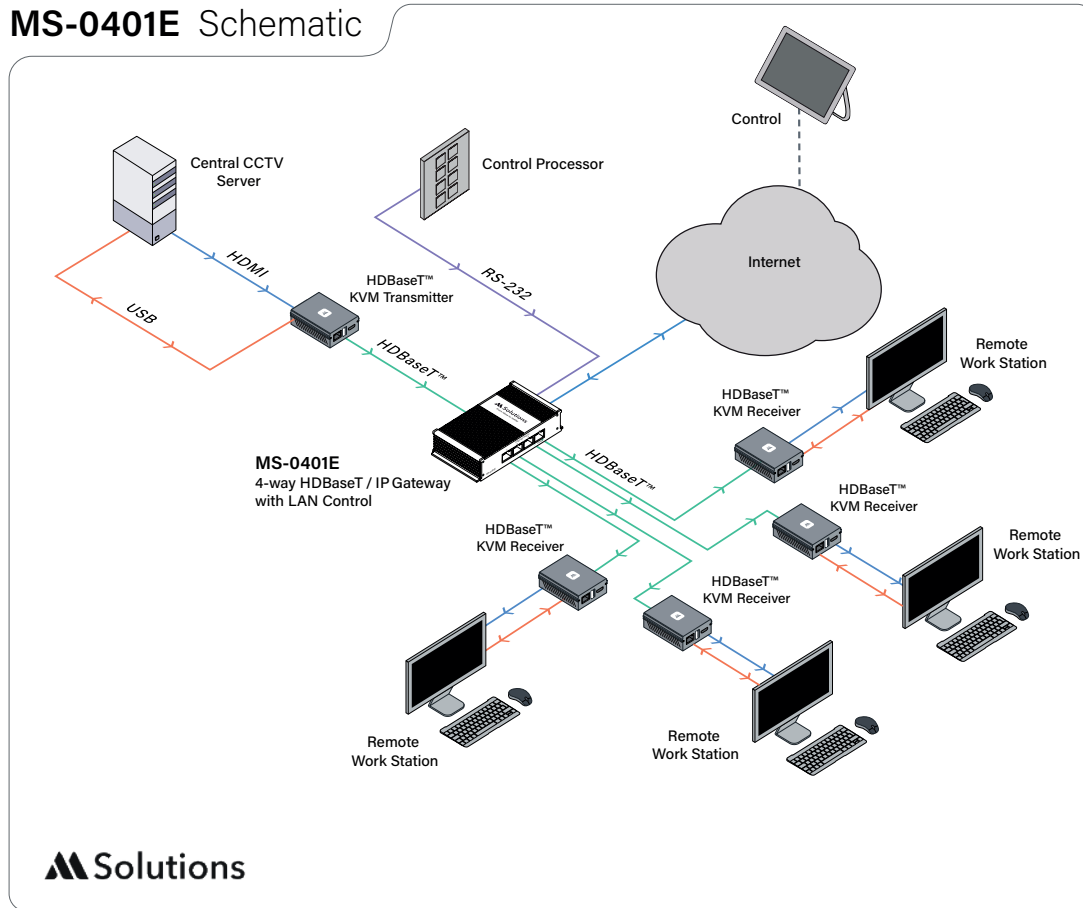
MS-0401E
Back



- USB —
- HDMI —
- HDBaseT™ —
- RS-232 —

Example Schematics

MS-0401E Schematic



MS-0401E
Front



MS-0401E
Back



Appendix: Full Set of Comprehensive Commands

1 MSolutions Switch RS232 command list

1.1 Command Format

Each Command must be terminated by CR Symbol ('\n' in C Notation).

1.2 Answer Format

Answers can be in Human Readable Format and in Machine Decoding Format. The example of Human Readable format is (green is command, blue is answer):

```
SetLogin test
OK
```

Another example of Human Readable Answer is:

```
bt
BootLoader>
```

Some messages may be sent by firmware without any commands just for debug purposes:

```
NodeRegistered with result 201
DeviceRegistered with result 201
```

Answers for Machine Decoding process, have strong format for be extracted from the stream and decoded.

Each Machine Decode Answer begins from "##" sequence. Then it has Message Type Symbol (described in chapter 1.4). The data follows the header and has sequence "@#" as a terminator. Additionally, the line is terminated by CR Symbol.

As a result, the Regular Expression for search formatted message is: "##.+?@#\n"

1.3 Commands List

Command	Description
RR	Reply: current active port GetPortNames ##1Cat's port@# ##2Dog's port@# ##3Owl's Port@# ##4My Port@#
O1	Set new switch state (no reply)
O2	
O3	
O4	
ip	Reply: network settings and state ip dhcp ip:10.0.0.44 MAC:00:04:25:37:78:8A mask:255.255.255.0 gw:10.0.0.138 Link: 1

Command	Description
er sure	Reset settings to default (erase EEPROM and reset CPU). er sure Settings have been reset to default
reset sure	Force System Reset reset sure TCP/IP Stack: Initialization Started TCP/IP Stack: Initialization Ended - success 1 0:formatted
bb	Reply: firmware version bb FW:1.2 HW:1.1
bt	Jump to bootloader mode bt BootLoader>
GetPortNames	Request assigned names of Switch's ports GetPortNames ##1port0@# ##2port1@# ##3port2@# ##4port3@#
SetPortName<N> <Val>	Set name of selected Port (number is 0-Based) SetPortName0 Cat's Port port name changed GetPortNames ##1 Cat's Port@# ##2port1@# ##3port2@# ##4port3@#
GetLogin	Get Administrator's Login GetLogin ##YMSolutions@#
SetLogin <Val>	Set Administrator's Login SetLogin test OK GetLogin ##Ytest@#
GetPassword	Get Administrator's Password (example is similar with "GetLogin")
SetPassword <Val>	Set Administrator's Password (example is similar with "SetLogin")
getNodeId	Get NMOS Node's ID (example is similar with "GetLogin")
SetNodeId <Val>	Set NMOS Node's ID (example is similar with "SetLogin")
GetDevId	Get NMOS Device's ID (example is similar with "GetLogin")
SetDevId <Val>	Set NMOS Device's ID (example is similar with "SetLogin")
getNodePrefix	Get NMOS Node's Prefix (example is similar with "GetLogin")
SetNodePrefix <Val>	Set NMOS Node's Prefix (example is similar with "SetLogin")
GetDevPrefix	Get NMOS Device's Prefix (example is similar with "GetLogin")
SetDevPrefix <Val>	Set NMOS Device's Prefix (example is similar with "SetLogin")

Command	Description
GetAll	Request all settings except login/password (for minimal security reasons) <pre>GetAll ##1 Cat's Port@# ##2port1@# ##3port2@# ##4port3@# ##5419280bd-ff15-4350-b4be-32fc2537788a@# ##6e219e025-20f2-42f8-b4a1-46ff383b6e75@# ##7VimasNode@# ##8MS-0401-8_Switch0@#</pre>

1.4 Answers List for Machine Decode

Type Symbol	Description	Example
A	Port 1 is activated	RR ##A@#
B	Port 2 is activated	RR ##B@#
C	Port 3 is activated	RR ##C@#
D	Port 4 is activated	RR ##D@#
Y	Administrator's Login	GetLogin ##Ytest@#
Z	Administrator's Password	GetPassword ##Zpassed@#
1	Name of Port 1	GetPortNames
2	Name of Port 2	##1Cat's port@#
3	Name of Port 3	##2Dog's port@#
4	Name of Port 4	##3Owl's Port@# ##4My Port@#
5	NMOS Node ID	GetNodeId ##5419280bd-ff15-4350-b4be-32fc2537788a@#
6	NMOS Device ID	GetDevId ##6e219e025-20f2-42f8-b4a1-46ff383b6e75@#
7	NMOS Node Prefix	GetNodePrefix ##7VimasNode@#
8	NMOS Device Prefix	GetDevPrefix ##8MS-0401-8_Switch0@#

2 TCP IP commands

2.1 Command List

JSON command example	Response	Description
<pre>{ "action": "select", "port": 1 }</pre>	<pre>{ "response": "select", "port": 1 }</pre>	Command to select port 1
<pre>{ "action": "report" }</pre>	<pre>{ "response": "report", "dhcp": true, "IP": "10.0.0.34", "Mask": "255.255.255.0", "Gateway": "0.0.0.0", "MAC": "00:04:AA:AA:AA:AA", "Baudrate": 19200, "CurrentPort": 1 }</pre>	Request report
<pre>{ "action": "setpassword", "clogin": "MSolutions", "cpassword": "MSolutions", "nlogin": "test", "npassword": "passed" }</pre>	<pre>{ "response": "setpassword", "errcode": <N> }</pre>	Set New Administrator Login and password Error Codes: 0 – No Error 1 – Current login/password mismatched 2 – Missed any parameter
<pre>{ "action": "setportname" "port1": "AnyValue" "port2": "AnyValue" "port3": "AnyValue" "port4": "AnyValue" }</pre>	<pre>{ "response": "setportname" }</pre>	Port Names are optional
<pre>{ "action": "setuartparams" "baudrate": <val> }</pre>	<pre>{ "response": "setuartparams" }</pre>	Possibly values: 9600, 19200, 38400, 57600. 115200 Any other value will means as 115200
<pre>{ "action": "getnmosparams" }</pre>	<pre>{ "response": "getnmosparams", "server": "...", "port": "...", "node_prefix": "...", "node_id": "...", "device_prefix": "...", "device_id": "...", }</pre>	Report NMOS Settings

JSON command example	Response	Description
<pre>{ "action": "setnmosparams", "server": "<Address>", "port": <Val> }</pre>	<pre>{ "response": "setnmosparams" }</pre>	Set MNOS Settings System Reset is required!
<pre>{ "action": "systemreset" }</pre>	No Response	Force system reset. Connection will be lost!
<pre>{ "action": "setnetparams", "dhcp": <0 1>, "ip": "<Address>", "mask": "<Address>", "gateway": "<Address>", }</pre>	<pre>{ "response": "setnetparams" }</pre>	Set Network Parameters If dhcp is 1, then all other parameters are not needed Connection may be lost that is why System reset will be automatically forced
<pre>{ "action": "setnetname", "name": "any name", }</pre>	<pre>{ "response": "setnetname" }</pre>	Set mdns ID of this device. System Reset is required!

2.2 Examples

2.2.1 Select Port Example

```
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

i = 1
while(True):
    i = i+1
    if i>4:
        i=1
    a = '{\n\t"action": "select", \n\t"port": '+str(i)+'\n}'
    print(a + '\n')
    conn.send(a.encode('utf-8'))

    conn.settimeout(2);
    data = conn.recv(1024)
    stringdata = data.decode('utf-8')
    print(stringdata)

    time.sleep(0.1)
```

```
{
    "action": "select",
    "port": 1
}

{
    "response": "select",
    "port": 1
}

{
    "action": "select",
    "port": 2
}

{
    "response": "select",
    "port": 2
}
```

2.2.2 Read Parameters Example

```
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a = '{\n\t"action": "report"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))
conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)
```

```
{
    "action": "report"
}

{
    "response": "report"
    "dhcp": true,
    "IP": "10.0.0.44",
    "Mask": "255.255.255.0",
    "Gateway": "10.0.0.138",
    "MAC": "00:04:25:37:78:8A",
    "Baudrate": 115200,
    "CurrentPort": 1
}
```

2.2.3 Change Password Example

```
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a =
'{"\n\t"action": "setpassword", \n\t"clogin": "test", \n\t"cpassword": "passed", \n\t"nlogin": "MSolutions", \n\t"npassword": "MSolutions"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))
conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)
```

```
{
    "action": "setpassword",
    "clogin": "test",
    "cpassword": "passed",
    "nlogin": "MSolutions",
    "npassword": "MSolutions"
}

{
    "response": "setpassword",
    "errcode": 0
}
```

2.2.4 Change Multiu Port Names Example

```
import socket
import time
```

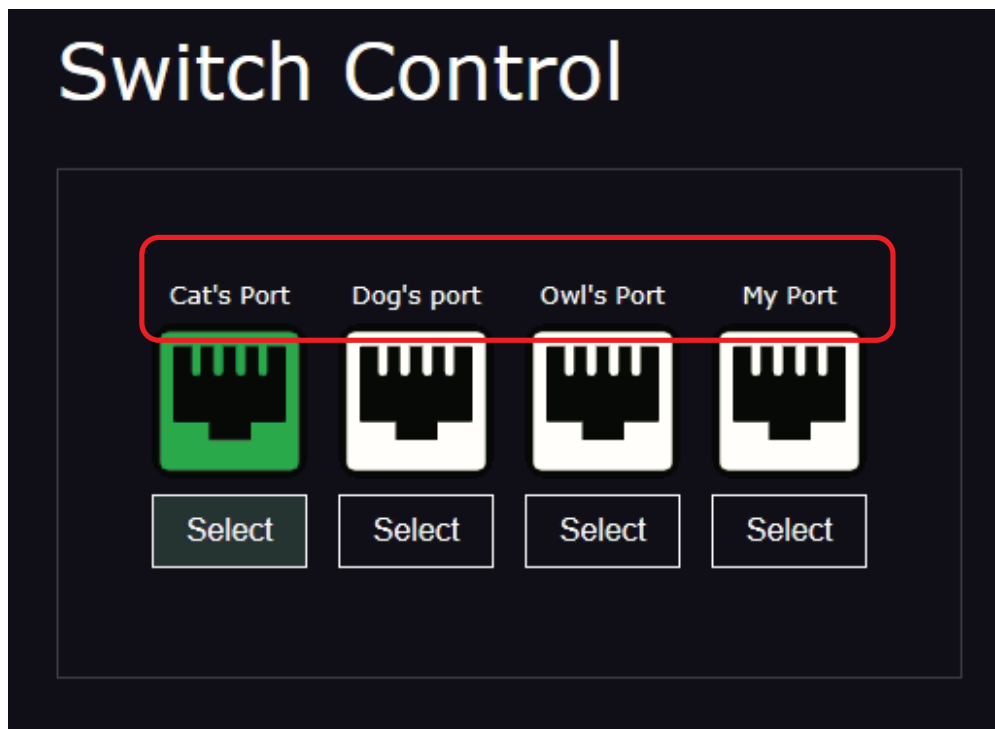
```

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a = '{\n\t"action": "setportname", \n\t"port1": "Cat\'s
Port", \n\t"port2": "Dog\'s port", \n\t"port3": "Owl\'s
Port", \n\t"port4": "My Port"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))

conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)

```



2.2.5 Change Single Port Name Example

```

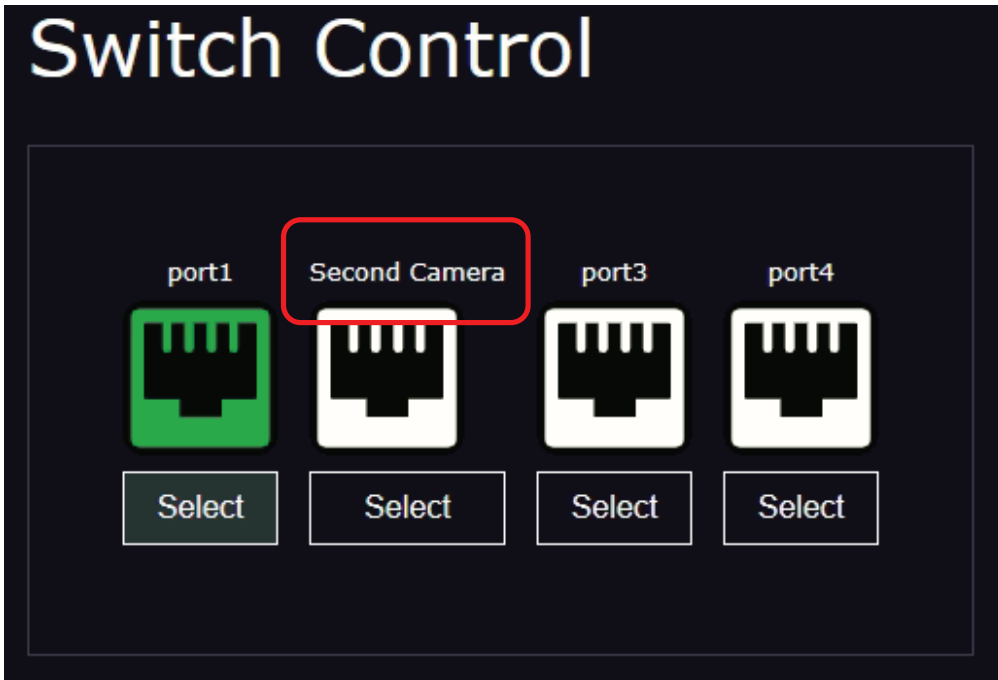
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a = '{\n\t"action": "setportname", \n\t"port2": "Second Camera"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))

conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)

```



2.2.6 Set Baud Rate Example

```
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a = '{\n\t"action": "setuartparams", \n\t"baudrate": 9600\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))

conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)
```

2.2.7 Get NMOS Parameters Example

```
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a = '{\n\t"action": "getnmosparams"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))
conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)
```

```
{
    "action": "getnmosparams"
}

{
    "response": "getnmosparams",
    "server": "10.0.0.43",
    "port": 8235,
    "node_prefix": "VimasNode",
    "node_id": "419280bd-ff15-4350-b4be-32fc2537788a",
    "device_prefix": "MS-0401-8_Switch321",
    "device_id": "e219e025-20f2-42f8-b4a1-46ff383b6e75"
}
```

2.2.8 Set NMOS Parameters example

```
import socket
import time

conn = socket.socket()
conn.connect(("MS0401E.local", 9760))

a =
'{"\n\t"action": "setnmosparams", \n\t"server": "10.0.0.43", \n\t"port": 8235\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))
conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)

a = '{"\n\t"action": "systemreset"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))
time.sleep(0.5)
```

```
{
    "action": "setnmosparams",
    "server": "10.0.0.43",
    "port": 8235
}

{
    "response": "setnmosparams",
}

{
    "action": "systemreset"
}
```

2.2.9 Set Net Name (MDNS Name) Example

```
import socket
import time

conn = socket.socket()
conn.connect(("mswitch.local", 9760))

a = '{\n\t"action": "setnetname", \n\t"name": "MS0401E.local"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))

conn.settimeout(2);
data = conn.recv(1024)
stringdata = data.decode('utf-8')
print(stringdata)

a = '{\n\t"action": "systemreset"\n}'
print(a + '\n')
conn.send(a.encode('utf-8'))
time.sleep(0.5)
```

```
{
  "action": "setnetname",
  "name": "MS0401E.local"
}

{
  "response": "setnetname",
}
{
  "action": "systemreset"
}
```



www.m4sol.com