Developer Information

Blackmagic PTZ Control

Camera Control using ATEM, PTZ Control over SDI, VISCA commands, Controlling Pan, Tilt and Zoom with Blackmagic 3G-SDI Arduino Shield, Controlling your Arduino

April 2018
Contents

Blackmagic PTZ Control

Camera Control using ATEM ........................................ 3
PTZ Control over SDI ............................................. 3
VISCA commands .................................................. 4
Controlling Pan, Tilt and Zoom with Blackmagic 3G-SDI Arduino Shield ........................................ 5
Controlling your Arduino ........................................... 6
Camera Control using ATEM

VISCA

All ATEM switchers with a remote port support VISCA camera control via RS-422. VISCA commands are defined by controlling the cameras via ATEM external hardware panels, such as ATEM 1 M/E Advanced Panel and ATEM Broadcast Panels.

Refer to the ATEM Production Studio Switchers and ATEM Television Studio Switchers manuals for more information.

PTZ Control over SDI

ATEM

ATEM external hardware panels, such as ATEM 1 M/E Advanced Panel and ATEM Broadcast Panels can control PTZ camera heads via your switcher’s SDI program return output. By connecting the program return feed from your switcher to a Blackmagic Micro Studio Camera, then connecting the SDI output from the camera’s expansion cable to your PTZ head, you can control the head via the SDI signal.

For more information on PTZ control using a Blackmagic Micro Studio Camera refer to the Blackmagic Studio Cameras manual. This manual can be downloaded from the Blackmagic Design support center at www.blackmagicdesign.com/support

Blackmagic 3G-SDI Arduino Shield

Blackmagic Micro Studio Camera 4K supports PTZ output in the form of VISCA commands, which can be sent to a compatible motorized head. By using a Blackmagic 3G-SDI Arduino Shield, you can send pan, tilt and zoom commands over SDI to your Blackmagic Micro Studio Camera 4K. Your camera will then translate these SDI camera control protocol commands into the VISCA protocol, and send them to a compatible motorized head via the 9-pin connector on the expansion cable labelled ‘PTZ control’.

This means that you can use one SDI cable in a live production environment, to send camera control commands to remotely control any setting in the camera, as well as send PTZ commands to a compatible motorized head to control pan and tilt. The pan and tilt commands will be sent by your Blackmagic Micro Studio Camera 4K to the motorized head, whereas lens related commands such as iris, focus and zoom commands will be sent to the active lens that is connected to the camera.
The commands that the Micro Studio Camera 4K can accept over SDI are:

- Lens Zoom
- Lens Focus
- Lens Iris
- Pan Tilt
- Memory Set
- Memory Recall
- Memory Reset

These commands are referenced in the ‘Blackmagic SDI Control Protocol’ in the ‘Blackmagic Camera Control’ developer information document which can be downloaded at www.blackmagicdesign.com/developer/

Most PTZ heads support the setting and recalling of their positions but it is a good idea to check which commands are supported by each PTZ head manufacturer.

The commands that are output through the ‘PTZ control’ connector in the form of VISCA commands are:

- CAM_Memory
- Pan-tiltDrive

### VISCA commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pan-tiltDrive</strong></td>
<td></td>
</tr>
<tr>
<td>Up</td>
<td>8x 01 06 01 VV WW 03 01 FF</td>
</tr>
<tr>
<td>Down</td>
<td>8x 01 06 01 VV WW 03 02 FF</td>
</tr>
<tr>
<td>Left</td>
<td>8x 01 06 01 VV WW 01 03 FF</td>
</tr>
<tr>
<td>Right</td>
<td>8x 01 06 01 VV WW 02 03 FF</td>
</tr>
<tr>
<td>UpLeft</td>
<td>8x 01 06 01 VV WW 01 01 FF</td>
</tr>
<tr>
<td>UpRight</td>
<td>8x 01 06 01 VV WW 02 01 FF</td>
</tr>
<tr>
<td>DownLeft</td>
<td>8x 01 06 01 VV WW 01 02 FF</td>
</tr>
<tr>
<td>DownRight</td>
<td>8x 01 06 01 VV WW 02 02 FF</td>
</tr>
<tr>
<td>Stop</td>
<td>8x 01 06 01 VV WW 03 03 FF</td>
</tr>
<tr>
<td>AbsolutePosition</td>
<td>8x 01 06 02 VV WW 0Y 0Y 0Y 0Z 0Z 0Z FF</td>
</tr>
<tr>
<td>RelativePosition</td>
<td>8x 01 06 03 VV WW 0Y 0Y 0Y 0Z 0Z 0Z FF</td>
</tr>
<tr>
<td>Home</td>
<td>0Y 0Y 0Y 0Y 0Z 0Z 0Z 0Z FF</td>
</tr>
<tr>
<td>Reset</td>
<td>8x 01 06 05 FF</td>
</tr>
</tbody>
</table>

| **CAM_Memory**     |                         |
| Reset              | 8x 01 04 3F 00 0p FF     |
| Set                | 8x 01 04 3F 01 0p FF     |
| Recall             | 8x 01 04 3F 02 0p FF     |

| VV:                | Pan speed 01 to 18      |
| WW:                | Tilt speed 01 to 17     |
| YYYY:              | Pan position F725 to 08DB (center 0000) |
| ZZZZ:              | Tilt position FE70 to 04B0 (image flip: OFF) (center 0000) |
|                   | Tilt position FB50 to 0190 (image flip: ON) (center 0000) |

| p:                 | Memory number (=0 to 5) |
|                    | Corresponds to 1 to 6 on the remote commander. |

Compatible motorized heads include the following:

- KXWell KT-PH180BMD
- PTZOptics PT-Broadcaster
- RUSHWORKS PTX Model 1
Controlling Pan, Tilt and Zoom with Blackmagic 3G-SDI Arduino Shield

Using the Blackmagic 3G-SDI Arduino Shield with an Arduino board, a joystick and a switch, you can control a PTZ head via Blackmagic Micro Studio Camera 4K.

Connecting your Blackmagic Micro Studio Camera 4K to the Blackmagic Design 3G-SDI Shield

1. Connect the Blackmagic Design 3G-SDI Shield to an Arduino board.
2. Connect the custom shield to the Arduino board.
3. Attach the SDI output connector from the shield to the SDI input on your Blackmagic Micro Studio Camera 4K and set the camera as camera number 1.
4. Connect the joystick and button to the shield.

The joystick is mapped as follows:
- X axis adjusts the PTZ head’s pan.
- Y axis adjusts the PTZ head’s tilt.
- Pressing the joystick button tells the PTZ head to store the current X, Y position in memory.
- Pressing the switch recalls the stored position.

**NOTE** The ATEM SDK supports the Blackmagic SDI Camera Control Protocol, and is an alternative to using a Blackmagic 3G-SDI Arduino Shield for control.

Refer to the ATEM Switchers SDK manual for more information. The ATEM Switchers SDK manual can be downloaded at www.blackmagicdesign.com/support.
Controlling your Arduino

The following sketch demonstrates a simple example of using a joystick and button with an Arduino board and the Blackmagic 3G-SDI Arduino Shield, to control a PTZ head via a Blackmagic Micro Studio Camera 4K.
Controlling your Arduino

```c
// Check if joystick has been pressed
if (getJoystickPressed(joystickPinX) == true) {
    // Setup memory values
    R // Reset memory
    B // First slide
}
setCameraControl::writeCameraControl(
    // Destination: Camera 1
    // Category: Internal device
    // Port: IS_Pan Tilt Speed
    // Operation: Set Abs Value,
    pwmValues // Values
);
}

// Read the joystick value as a (-1024:1024) scaled value
setJoystickValue = analogRead(joystickPinY);
setJulieJoystickValue = joystickValue / (1024.0 / 1024.0);

// Determine value close to zero by zero, so that when the joystick is
// centered it reports zero even if it is slightly mis-aligned
if (setJulieJoystickValue < 0.0) {
    setJulieJoystickValue = 0.0;
}
return setJulieJoystickValue;

// Detects double-pressed edge (i.e. pressed and released) of a button
bool previousLevel = !digitalRead(digitalPin);
bool currentLevel = digitalRead(digitalPin);

return previousLevel != currentLevel;

// Reads a digital pin and filters it, returning the stable button position
int pressed = digitalRead(digitalPin);
unsigned long currentTime = millis();

// If the button is rapidly changing (bouncing) during a press, keep
// recording the last stable time value.
if (pressed == maintainedLevel) {
    maintainedLevel = digitalPin;
    maintainedLevel = (millis() - currentTime) / pressed;
    // Once the button has been stable for
    if (currentTime > lastStableButtonTime[digitalPin]) {
        stableButtonTime[digitalPin] = currentTime;
        stableButtonTime[digitalPin] = pressed;
    }
}
return stableButtonTime[digitalPin];
```

7

Controlling your Arduino