Create and use custom devices in ZEN Blue using the Arduino microcontroller

- How to leverage the full potential of ZEN Blue and 3rd party image processing (or hardware) in your image processing and analysis workflows (and image acquisition)

Usual choices for development of home-made custom devices

- Write a new software from scratch
- Labview
- Matlab (ScanImage)
- Micro-Manager

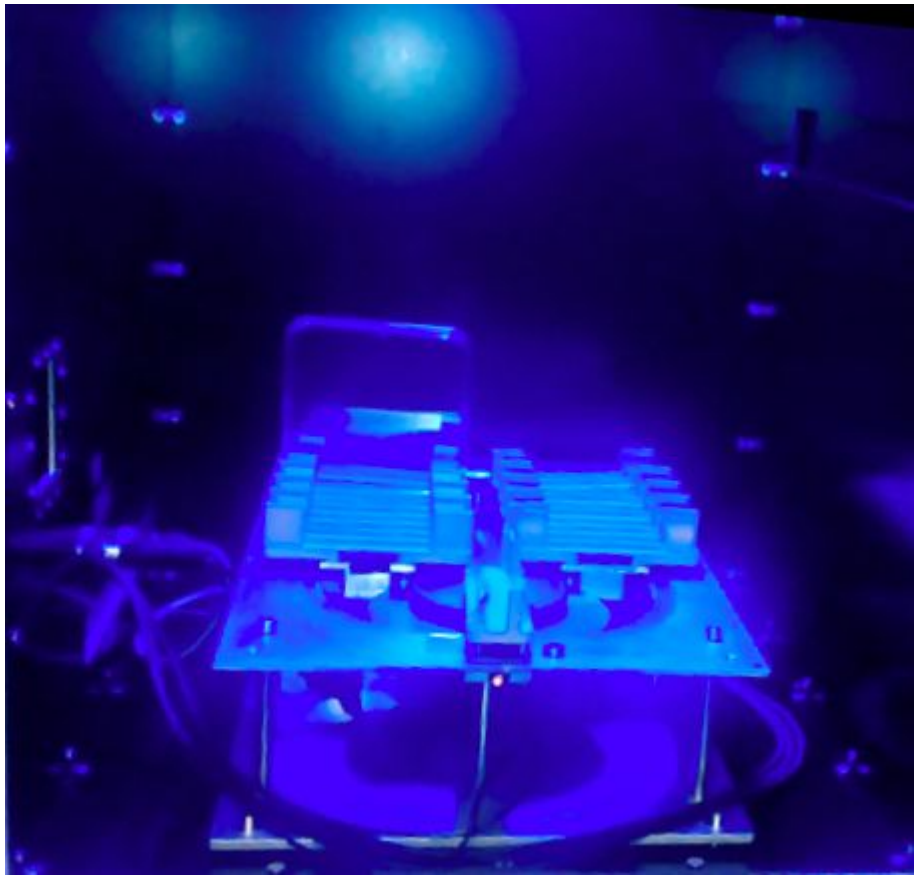
- Proprietary software (eg ZEN)

Lumalum, a root imaging robot

with Hélène Javot and Michel Philibert (CEA)

µManager

Open Source Microscopy Software



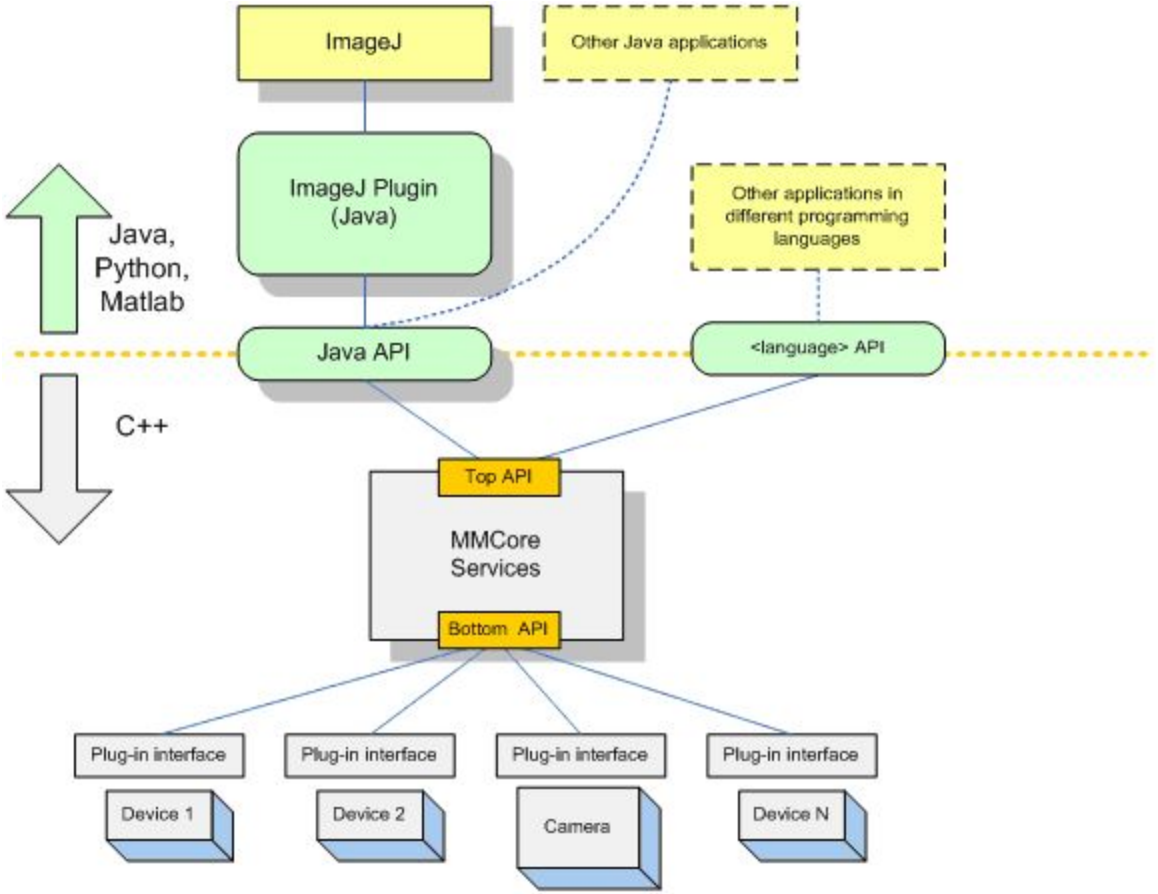
java com layer
mm bsh
mm plugin
ij macros
...

Simplifying MM device interaction

μManager
Open Source Microscopy Software

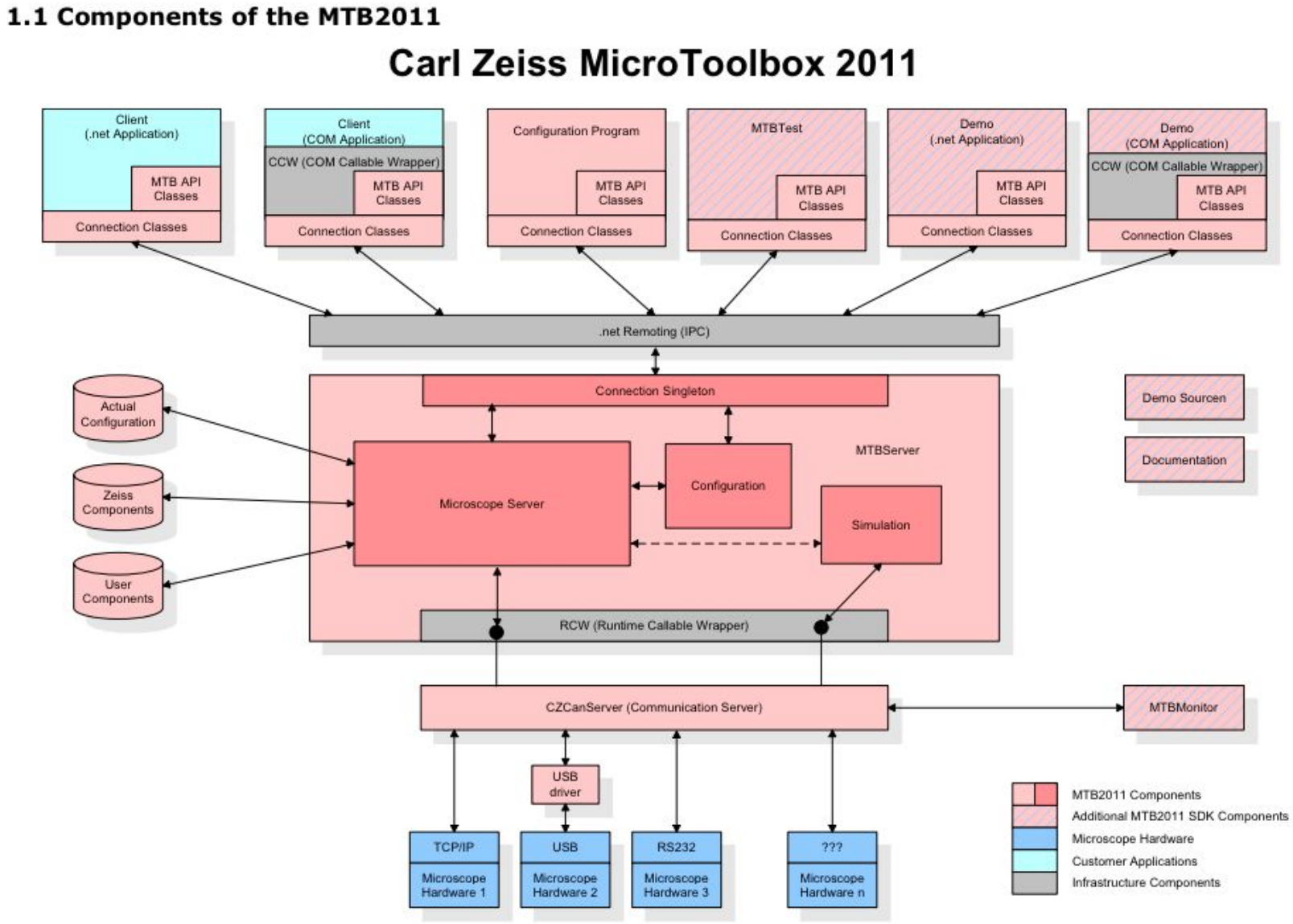
```
run("MM MacroExtensions");

for ( i = 0 ; i < 50 ; i = i+2 ) {
    Ext.moveRelativeXYZ( 0 , 0 , i );
    Ext.snap( );
    }

run ("Images to Stack");
run ("EDF Easy ", "quality='0' topology='1'");
```

MM design model



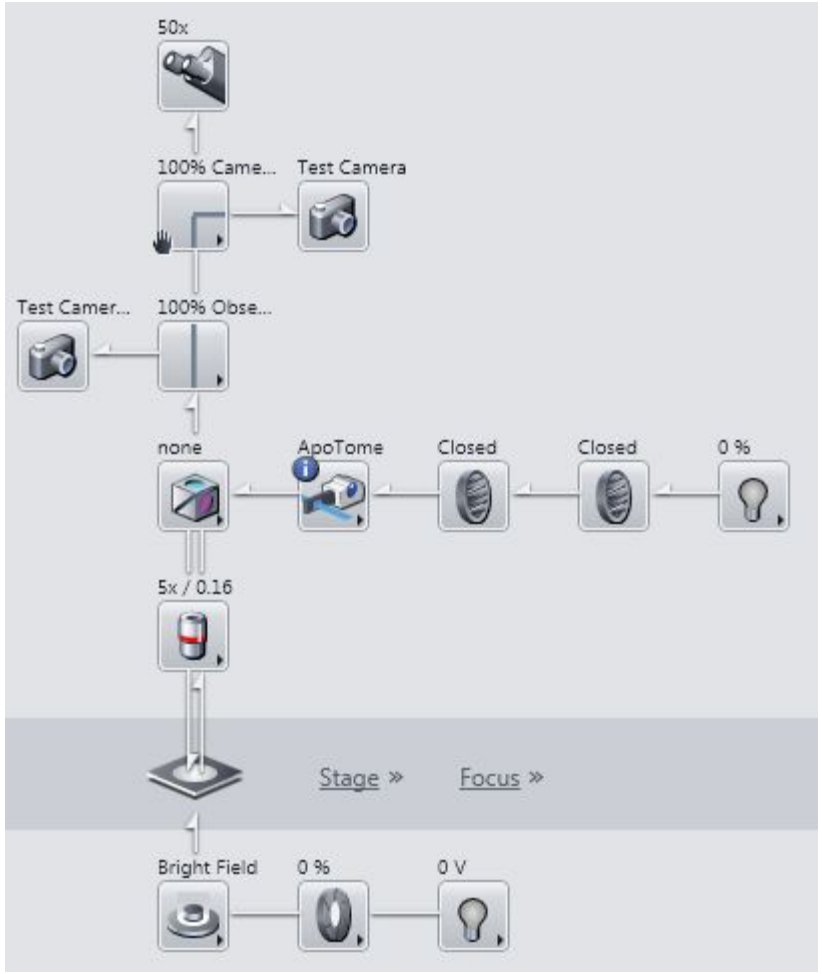
https://micro-manager.org/wiki/Micro-Manager%20Project%20Overview

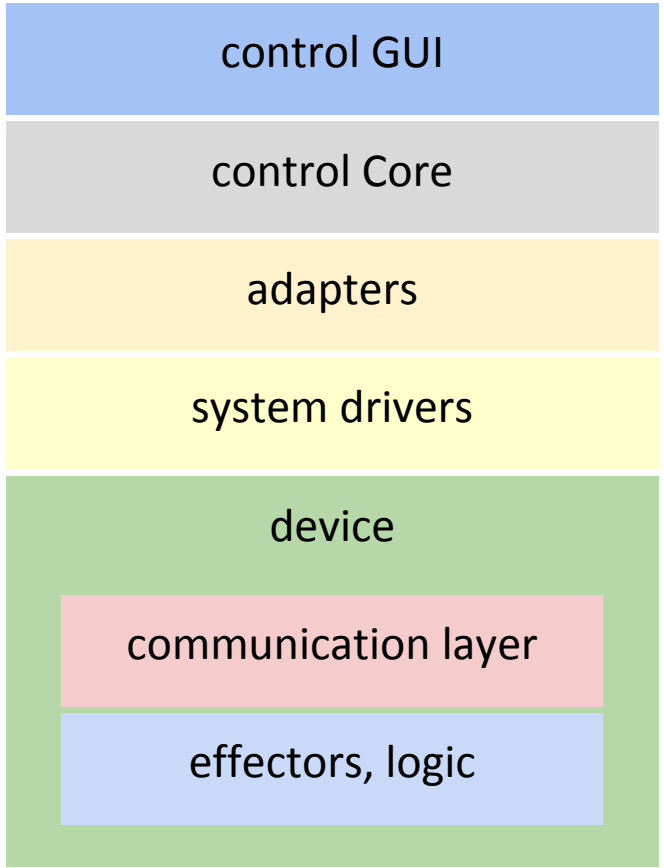
Zeiss MTB design model



https://www.zeiss.de/content/dam/Microscopy/Downloads/MicroToolbox/MTB2011/MTB2011_How_to_start.pdf

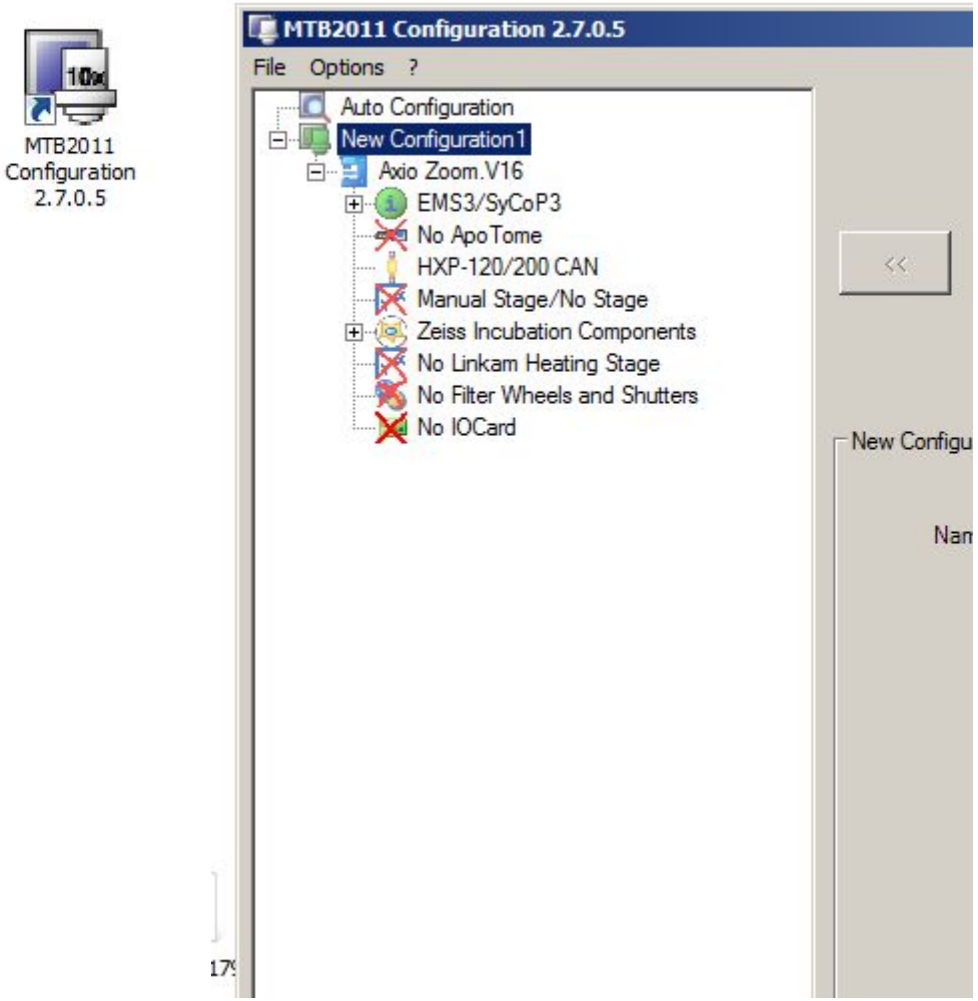
Elements of device interaction





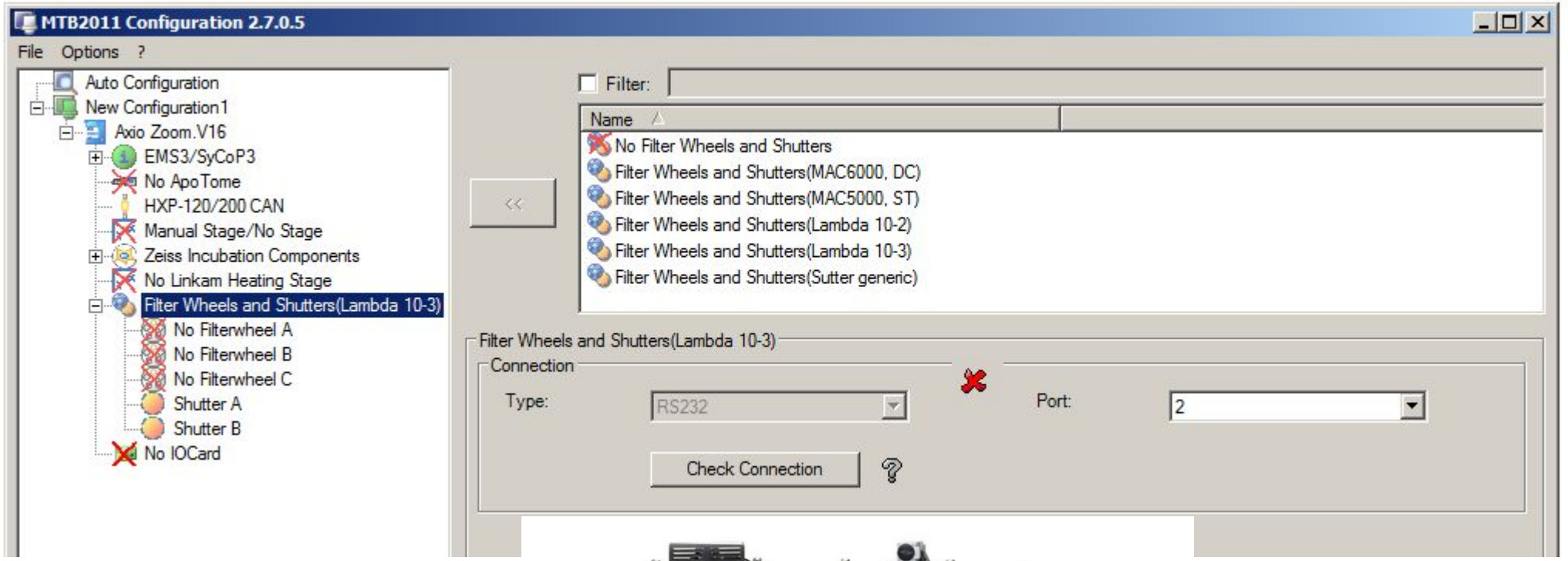
Target: Axiozoom under Zen Blue
Ability to control external light from Zen

The Zeiss *.mtb device model & Micro Tool Box Configuration



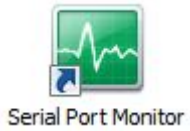

```
▼<MTB2004Export Version="2.7.0.5">
 ▼<MTBConfiguration Name="New Configuration1" MyChildId="0">
  ▼<Devices>
   ▶<Device Class="MTBCtrlDevice,MTBKernel" Name="Wega.Stand" External="false"
    PortType="None" PortNo="1" PortBaudAddr="57600" RTSCTS="false" CanNode="25"
    MyChildId="1">...</Device>
   ▶<Device Class="MTBCtrlElectronicModuleDeviceREO,MTBKernelREO"
    Name="Wega_Gateway_V16.EMS3" External="false" PortType="USB" PortNo="4111"
    PortBaudAddr="" RTSCTS="false" CANConnection="" CanNode="40" MyChildId="11600">...
    </Device>
   ▼<Device Class="MTBCtrlDeviceSycop3,MTBKernelWEGA" Name="Wega.Main" External="false"
    PortType="USB" PortNo="4111" PortBaudAddr="" RTSCTS="false" CanNode="25" MyChildId="0">
     <SimulationMode>true</SimulationMode>
     <SimulationClass>ZEISS.MTB.Simulation.MTBSimDeviceWegaMain</SimulationClass>
     <TechReport>ReportWegaMain</TechReport>
    ▼<Components>
     ▼<Component Class="MTBCtrlMicroscopeManagerSycop3,MTBKernelREO"
      Name="REO.MicroscopeManager" MTBId="MTBMicroscopeManager" CANId="00"
      MyChildId="11024">
       <LightManagerModes>Off</LightManagerModes>
      </Component>
     </Components>
    </Device>
   ▶<Device Class="MTBCtrlOpticsDeviceWEGA,MTBKernelWEGA" Name="Wega.OpticsDevice_mot"
    External="false" PortType="USB" PortNo="4111" PortBaudAddr="" RTSCTS="false"
    CanNode="32" MyChildId="0">...</Device>
   ▶<Device Class="MTBCtrlFocusDeviceREO,MTBKernelREO" Name="Wega.FocusDevice_mot"
    External="false" PortType="USB" PortNo="4111" PortBaudAddr="" RTSCTS="false"
    CanNode="36" CANConnection="" MyChildId="11500">...</Device>
   ▶<Device Class="MTBCtrlOpticsDeviceREO,MTBKernelREO" Name="Wega.FLTube_mot"
    External="false" PortType="USB" PortNo="4111" PortBaudAddr="" RTSCTS="false"
    CanNode="102" CANConnection="" MyChildId="11502">...</Device>
    <Device Active="false" Name="ApoTome.none" MyChildId="11604"/>
   ▶<Device Class="MTBCtrlCan29HXP120Device,MTBKernelCan29Changers" Name="HXP120.Device"
    External="false" PortType="USB" PortNo="4099" PortBaudAddr="" RTSCTS="false"
    CANConnection="" CanNode="54" MyChildId="11601">...</Device>
    <Device Active="false" Name="Stage.none" MyChildId="11602"/>
    <Device Active="false" Name="Incubation.AtmoTemp_none" MyChildId="14000"/>
    <Device Active="false" Name="Incubation.FlowTemp_none" MyChildId="14001"/>
    <Device Active="false" Name="Linkam.none" DisplayName="No Linkam Heating Stage"
    DisplayName-de="Kein Linkam Heiztisch" MyChildId="9914"/>
    <Device Active="false" Name="IOCard.NoIOCard" DisplayName="No IOCard" DisplayName-
    de="Keine IO-Karte" External="true" PortType="RS232" PortNo="2" PortBaudAddr="9600"
    RTSCTS="true" MyChildId="25100"/>
    <Device Active="false" Name="NoFilterWheelsShutters" DisplayName="No Filter Wheels and
    Shutters" DisplayName-de="Keine Filterräder und Verschlüsse" External="true"
    PortType="RS232" PortNo="2" PortBaudAddr="9600" RTSCTS="true" MyChildId="30050"/>
  </Devices>
 </MTBConfiguration>
</MTB2004Export>
```

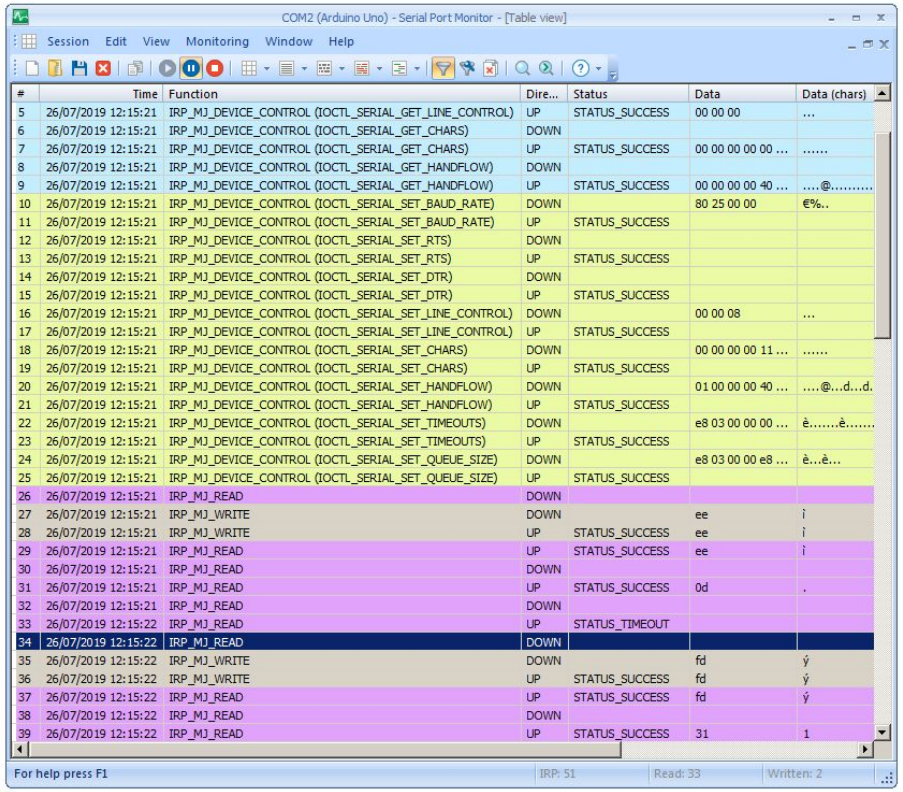
Chosen device : Sutter Lambda 10-3



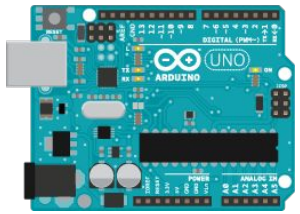
Lambda 10-3
Optical Filter Changer

Emulating the Sutter protocol





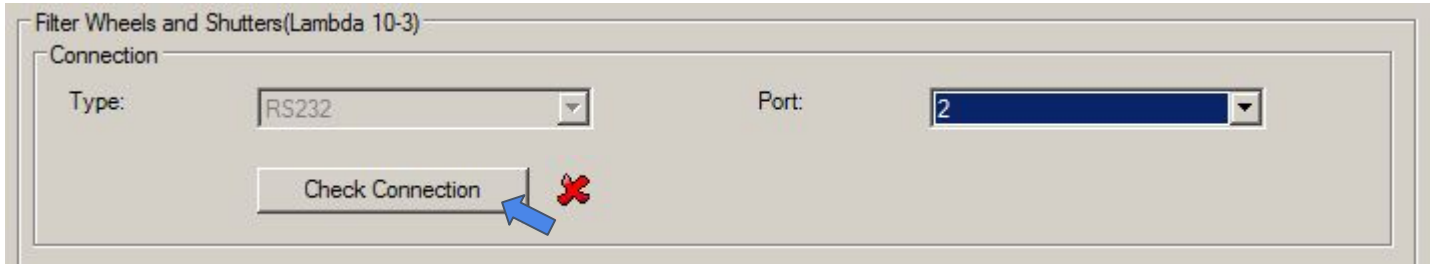
#	Time	Function	Dire...	Status	Data	Data (chars)
5	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_LINE_CONTROL)	UP	STATUS_SUCCESS	00 00 00	...
6	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_CHARS)	DOWN			
7	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_CHARS)	UP	STATUS_SUCCESS	00 00 00 00 00 ...	.....
8	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_HANDFLOW)	DOWN			
9	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_HANDFLOW)	UP	STATUS_SUCCESS	00 00 00 00 40 ...	....@.....
10	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_BAUD_RATE)	DOWN		80 25 00 00	€%..
11	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_BAUD_RATE)	UP	STATUS_SUCCESS		
12	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_RTS)	DOWN			
13	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_RTS)	UP	STATUS_SUCCESS		
14	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_DTR)	DOWN			
15	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_DTR)	UP	STATUS_SUCCESS		
16	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_LINE_CONTROL)	DOWN		00 00 08	...
17	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_LINE_CONTROL)	UP	STATUS_SUCCESS		
18	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_CHARS)	DOWN		00 00 00 00 11 ...	.....
19	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_CHARS)	UP	STATUS_SUCCESS		
20	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_HANDFLOW)	DOWN		01 00 00 00 40 ...	....@...d...d.
21	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_HANDFLOW)	UP	STATUS_SUCCESS		
22	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_TIMEOUTS)	DOWN		e8 03 00 00 00 ...	è.....è.....
23	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_TIMEOUTS)	UP	STATUS_SUCCESS		
24	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_QUEUE_SIZE)	DOWN		e8 03 00 00 e8 ...	è...è...
25	26/07/2019 12:15:21	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_QUEUE_SIZE)	UP	STATUS_SUCCESS		
26	26/07/2019 12:15:21	IRP_MJ_READ	DOWN			
27	26/07/2019 12:15:21	IRP_MJ_WRITE	DOWN		ee	î
28	26/07/2019 12:15:21	IRP_MJ_WRITE	UP	STATUS_SUCCESS	ee	î
29	26/07/2019 12:15:21	IRP_MJ_READ	UP	STATUS_SUCCESS	ee	î
30	26/07/2019 12:15:21	IRP_MJ_READ	DOWN			
31	26/07/2019 12:15:21	IRP_MJ_READ	UP	STATUS_SUCCESS	0d	.
32	26/07/2019 12:15:21	IRP_MJ_READ	DOWN			
33	26/07/2019 12:15:22	IRP_MJ_READ	UP	STATUS_TIMEOUT		
34	26/07/2019 12:15:22	IRP_MJ_READ	DOWN			
35	26/07/2019 12:15:22	IRP_MJ_WRITE	DOWN		fd	ý
36	26/07/2019 12:15:22	IRP_MJ_WRITE	UP	STATUS_SUCCESS	fd	ý
37	26/07/2019 12:15:22	IRP_MJ_READ	UP	STATUS_SUCCESS	fd	ý
38	26/07/2019 12:15:22	IRP_MJ_READ	DOWN			
39	26/07/2019 12:15:22	IRP_MJ_READ	UP	STATUS_SUCCESS	31	1



ZEN <-> RS232 <-> LB10-3

https://www.sutter.com/manuals/LB10-3_OpMan.pdf
https://www.arduino.cc/
https://www.eltima.com/products/serial-port-monitor/

MTB connection test



102	26/07/2019 12:23:46	IRP_MJ_CREATE	DOWN			C:\Program File...			COM2
103	26/07/2019 12:23:46	IRP_MJ_CREATE	UP	STATUS_SUCCESS		C:\Program File...			COM2
104	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_BAUD_RATE)	DOWN						COM2
105	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_BAUD_RATE)	UP	STATUS_SUCCESS	80 25 00 00	€%..	4		COM2
106	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_LINE_CONTROL)	DOWN						COM2
107	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_LINE_CONTROL)	UP	STATUS_SUCCESS	00 00 08	...	3		COM2
108	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_CHARS)	DOWN						COM2
109	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_CHARS)	UP	STATUS_SUCCESS	00 00 00 00 11 ...	.....	6		COM2
110	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_HANDFLOW)	DOWN						COM2
111	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_HANDFLOW)	UP	STATUS_SUCCESS	01 00 00 00 40 ...	....@...d...d...	16		COM2
112	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_BAUD_RATE)	DOWN		80 25 00 00	€%..	4		COM2
113	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_BAUD_RATE)	UP	STATUS_SUCCESS					COM2
114	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_RTS)	DOWN						COM2
115	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_RTS)	UP	STATUS_SUCCESS					COM2
116	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_DTR)	DOWN						COM2
117	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_DTR)	UP	STATUS_SUCCESS					COM2
118	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_LINE_CONTROL)	DOWN		00 00 08	...	3		COM2
119	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_LINE_CONTROL)	UP	STATUS_SUCCESS					COM2
120	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_CHARS)	DOWN		00 00 00 00 11 ...	.....	6		COM2
121	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_CHARS)	UP	STATUS_SUCCESS					COM2
122	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_HANDFLOW)	DOWN		01 00 00 00 40 ...	....@...d...d...	16		COM2
123	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_HANDFLOW)	UP	STATUS_SUCCESS					COM2
124	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_TIMEOUTS)	DOWN		e8 03 00 00 00 ...	è.....è.....	20		COM2
125	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_TIMEOUTS)	UP	STATUS_SUCCESS					COM2
126	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_QUEUE_SIZE)	DOWN		e8 03 00 00 e8 ...	è...è...	8		COM2
127	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_QUEUE_SIZE)	UP	STATUS_SUCCESS					COM2
128	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
129	26/07/2019 12:23:46	IRP_MJ_WRITE	UP	STATUS_SUCCESS	ee	î	1		COM2
130	26/07/2019 12:23:46	IRP_MJ_WRITE	UP	STATUS_SUCCESS	ee	î	1		COM2

Protocol standard : not standard

Transfer to On Line	238 EE 11101110	Puts controller on-line

5.6.2 Command Transmission Protocol

The Lambda 10-3 does not use any of the standard protocols commonly used for serial line or USB communications between computers or between a computer and a peripheral device. The Lambda 10-3 controller, however, does generate a primitive form of protocol of which control software running on the remote computer can, and should, make use. This protocol consists of two main components: "confirmation command echo" and "command completion indicator". This command transmission protocol is used in the same manner for RS-232 serial port, USB, and parallel port connections.

5.6.2.1 Confirmation Command Echo

When the host computer sends a command to the Lambda 10-3, each byte received is immediately echoed back to the host computer. This echoing back of each sent byte is a confirmation that the byte has been received and will be acted upon shortly. A short period after the last byte of a command has been echoed back, the Lambda 10-3 sends a confirmation byte (described next) that indicates the operation associated with the command completed.

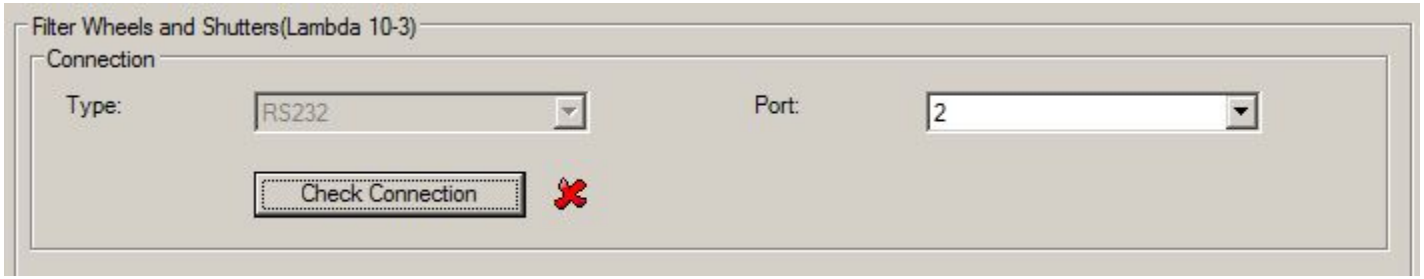
5.6.2.2 Command Completion Indicator

When the Lambda 10-3 completes the operation associated with the command it has just received, it transmits back to the host computer a byte value of 13 decimal (0D hexadecimal, 00001101 binary). This byte value corresponds to an ASCII carriage return (often abbreviated as "CR").

ZEN_SUTTERLAMBDA10_3.ino §

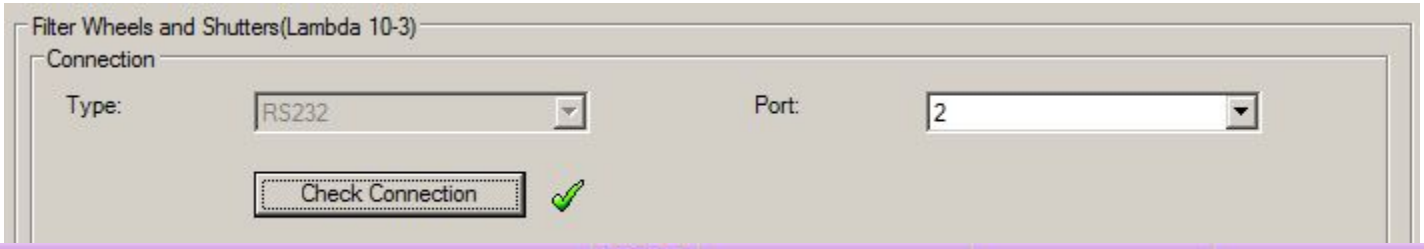
```
byte cmd ;

void setup() {
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  digitalWrite(13,LOW);
}

void loop()
{
  if (Serial.available()) {
    cmd = Serial.read();
    processCommand(cmd);
  }
}

void processCommand(byte s) {
  if (s==0xee) {
   Serial.write(s); // echo ONLINE cmd
   Serial.write(0x0d);
  }
}
```

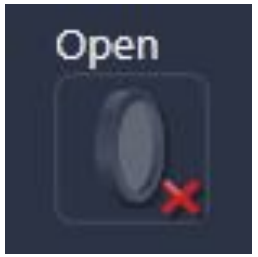
Configuration command **0xFD** : simple echo not enough



127	26/07/2019 12:23:46	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_SET_QUEUE_SIZE)	UP	STATUS_SUCCESS					COM2
128	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
129	26/07/2019 12:23:46	IRP_MJ_WRITE	DOWN		ee	ì	1	1	COM2
130	26/07/2019 12:23:46	IRP_MJ_WRITE	UP	STATUS_SUCCESS	ee	ì	1		COM2
131	26/07/2019 12:23:46	IRP_MJ_READ	UP	STATUS_SUCCESS	ee	ì	1		COM2
132	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
133	26/07/2019 12:23:46	IRP_MJ_READ	UP	STATUS_SUCCESS	0d	.	1		COM2
134	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
135	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_TIMEOUT					COM2
136	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
137	26/07/2019 12:23:47	IRP_MJ_WRITE	DOWN		fd	ý	1	1	COM2
138	26/07/2019 12:23:47	IRP_MJ_WRITE	UP	STATUS_SUCCESS	fd	ý	1		COM2
139	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	fd	ý	1		COM2
140	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
141	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	0d	.	1		COM2
142	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
143	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_TIMEOUT					COM2
144	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2

Just the model (**10-3**) is enough for MTB test check, but not for ZEN ! x





128	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
129	26/07/2019 12:23:46	IRP_MJ_WRITE	DOWN		ee	ì	1	1	COM2
130	26/07/2019 12:23:46	IRP_MJ_WRITE	UP	STATUS_SUCCESS	ee	ì	1		COM2
131	26/07/2019 12:23:46	IRP_MJ_READ	UP	STATUS_SUCCESS	ee	ì	1		COM2
132	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
133	26/07/2019 12:23:46	IRP_MJ_READ	UP	STATUS_SUCCESS	0d	.	1		COM2
134	26/07/2019 12:23:46	IRP_MJ_READ	DOWN					1	COM2
135	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_TIMEOUT					COM2
136	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
137	26/07/2019 12:23:47	IRP_MJ_WRITE	DOWN		fd	ý	1	1	COM2
138	26/07/2019 12:23:47	IRP_MJ_WRITE	UP	STATUS_SUCCESS	fd	ý	1		COM2
139	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	fd	ý	1		COM2
140	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
141	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	31	1	1		COM2
142	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
143	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	30	0	1		COM2
144	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
145	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	2d	-	1		COM2
146	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2
147	26/07/2019 12:23:47	IRP_MJ_READ	UP	STATUS_SUCCESS	33	3	1		COM2
148	26/07/2019 12:23:47	IRP_MJ_READ	DOWN					1	COM2

Fully functional reply to configuration command **0xFD**

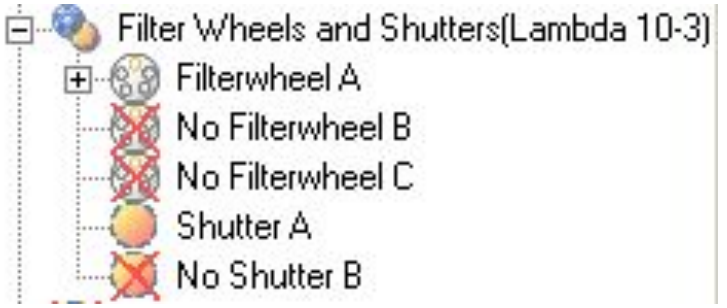
5.4.9 Get Controller Type and Configuration

This command is used to obtain information about the controller as to its model and configuration. The following table shows the type of information returned when issuing this command.

Table 5-8. "Get Controller Type and Configuration" command return codes and data.

Total Num. Bytes	Description			
	Category	Num. Bytes	Possible Values	
			ASCII String	Meaning
29	Controller Type	4	10-3	Lambda 10-3
	Filter Wheel A Type	5	WA-25	25mm
			WA-32	32mm
			WA-HS	High Speed
			WA-BD	Belt Driver
			WA-NC	Not Connected
			WA-ER	Error
	Filter Wheel B Type	5	WB-25	25mm
			WB-32	32mm
			WB-HS	High Speed
			WB-BD	Belt Driver
			WB-NC	Not Connected
			WB-ER	Error
	Filter Wheel C Type	5	WB-25	25mm
			WB-32	32mm
			WB-HS	High Speed
			WB-BD	Belt Driver
			WB-NC	Not Connected
			WB-ER	Error
	Shutter A Type	5	SA-IQ	SmartShutter
			SA-VS	Vincent Shutter
	Shutter B Type	5	SB-IQ	SmartShutter
			SA-VS	Vincent Shutter

```
void processCommand(byte s) {
  if (s == 0xee) {
    Serial.write(s); // echo ONLINE cmd
    Serial.write(0x0d);
  }  else if (s == 0xfd) {
    Serial.write(s); // echo CONFIG cmd
    Serial.print("10-3WA-BDWB-NCWC-NCSA-VSSB-VS");
    Serial.write(0x0d);
```



Full reply to the status command **0xCC**

5.4.2 Status

The Status command is used to return information about the filter wheel state (if installed) and the state of one or two shutters (if, and as, installed). The following table describes the type of data and information returned by the Status command.

Table 5-7. Status command return codes and data.

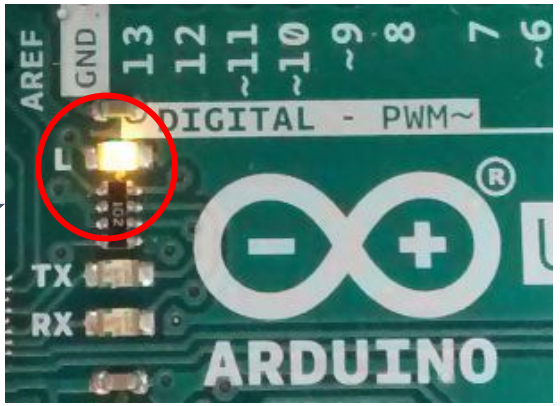
Order	Num. of bytes	Category	Sub Category	Value (Decimal, hexadecimal, & binary)	Description
1	1	**Command echo**		204 CC 11001100	The Status command byte code echoed back.
2	1	**Filter Wheel A status**	**Wheel, Speed and Position**	Encoded command byte: 0 – 121 00 – 79 00000000 – 01111001 *	Filter Wheel (Bit 7): 0
					Speed (Bits 6, 5, & 4): 0 – 7.
					Position (Bits 3, 2, 1, & 0): 0 – 9
5	1	**Shutter A open/-closed state**	**Open**	170 AA 10101010	Shutter A is in the open state.
			Open state is conditional	171 AB 10101011	The open state of Shutter A is conditional upon the movement of the Filter Wheel A.
			Closed	172 AC 10101100	Shutter A is in the closed state.

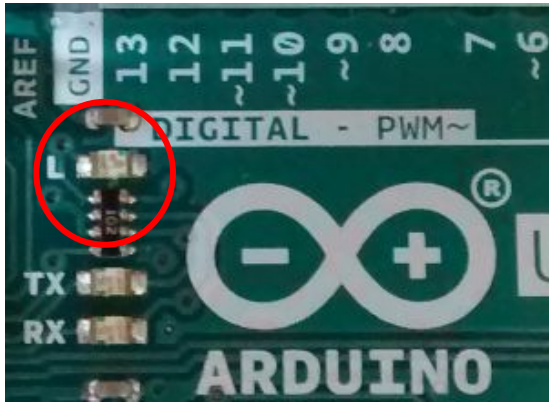
```
void processCommand(byte s) {
  if (s == 0xee) {
    Serial.write(s); // echo ONLINE cmd
    Serial.write(0x0d);
  }  else if (s == 0xfd) {
    Serial.write(s); // echo CONFIG cmd
    Serial.print("10-3WA-BDWB-NCWC-NCSA-VSSB-VS");
    Serial.write(0x0d);
    Serial.flush();
  } else if (s == 0xcc) {
    Serial.write(s);       // echo STATUS cmd
    Serial.write(0x00);    // FWA
    Serial.write(0x80);    // FWB
    Serial.write(0xfc);    // FWC
    Serial.write(0x00);
    Serial.write(0xac);    // SA closed
    Serial.write(0xbc);    // SB closed
    Serial.write(0xdc);    // SA SmartShutter Fast mode
    Serial.write(0x01);
    Serial.write(0xdc);    // SB SmartShutter Fast mode
    Serial.write(0x02);
    Serial.write(0x0d);    // confirmation
    Serial.flush();
```

Functional Shutter

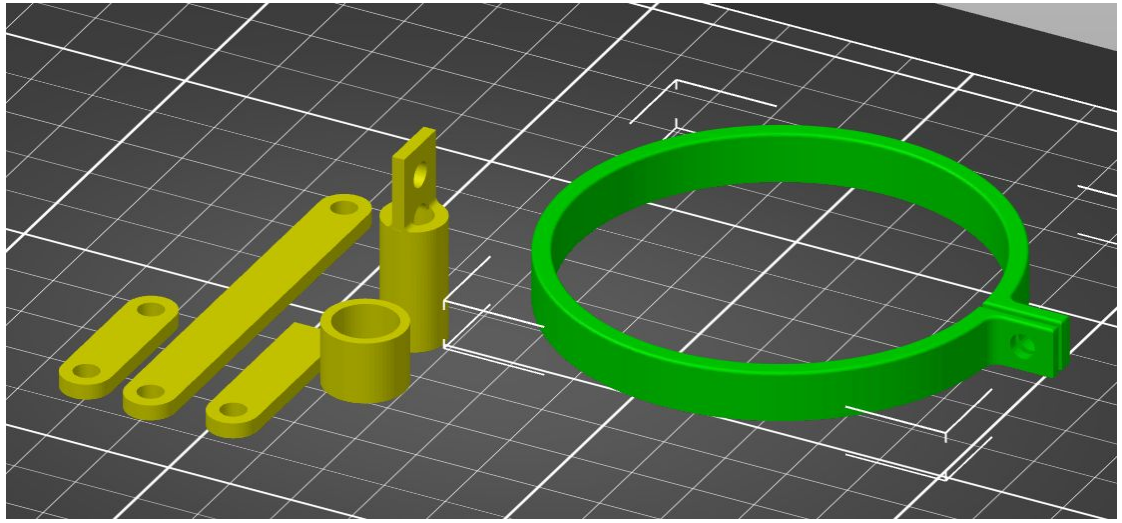
DOWN		aa	ª	1	1	COM
UP	STATUS_SUCCESS	aa	ª	1		COM
UP	STATUS_SUCCESS	aa	ª	1		COM
DOWN					1	COM
UP	STATUS_SUCCESS	0d	.	1		COM
DOWN					1	COM
UP	STATUS_TIMEOUT					COM
DOWN					1	COM
DOWN		ac	¬	1	1	COM
UP	STATUS_SUCCESS	ac	¬	1		COM
UP	STATUS_SUCCESS	ac	¬	1		COM
DOWN					1	COM
UP	STATUS_SUCCESS	0d	.	1		COM

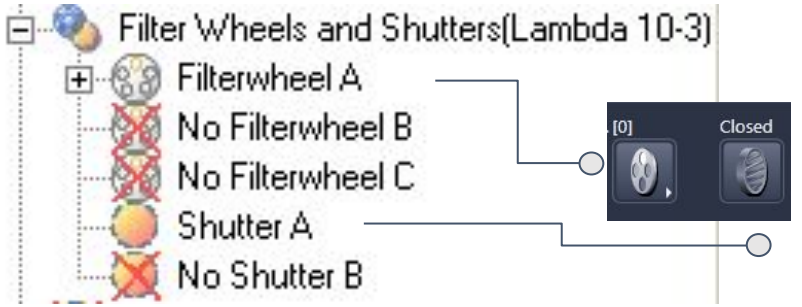
```
} else if (s==0xaa) {
 Serial.write(s);        // echo SA OPEN
 Serial.write(0x0d);
 digitalWrite(13,HIGH);
} else if (s==0xac) {
 Serial.write(s);        // echo SA CLOSE
 Serial.write(0x0d);
 digitalWrite(13,LOW);
} else {
 Serial.write(s); // echo any other case
 Serial.write(0x0d);
}
```

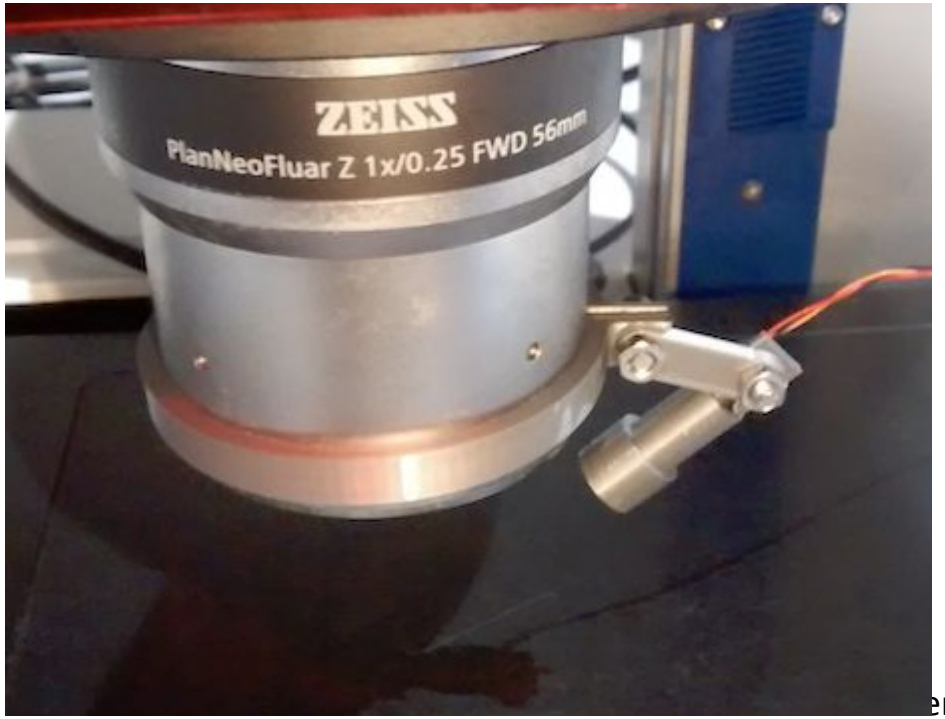




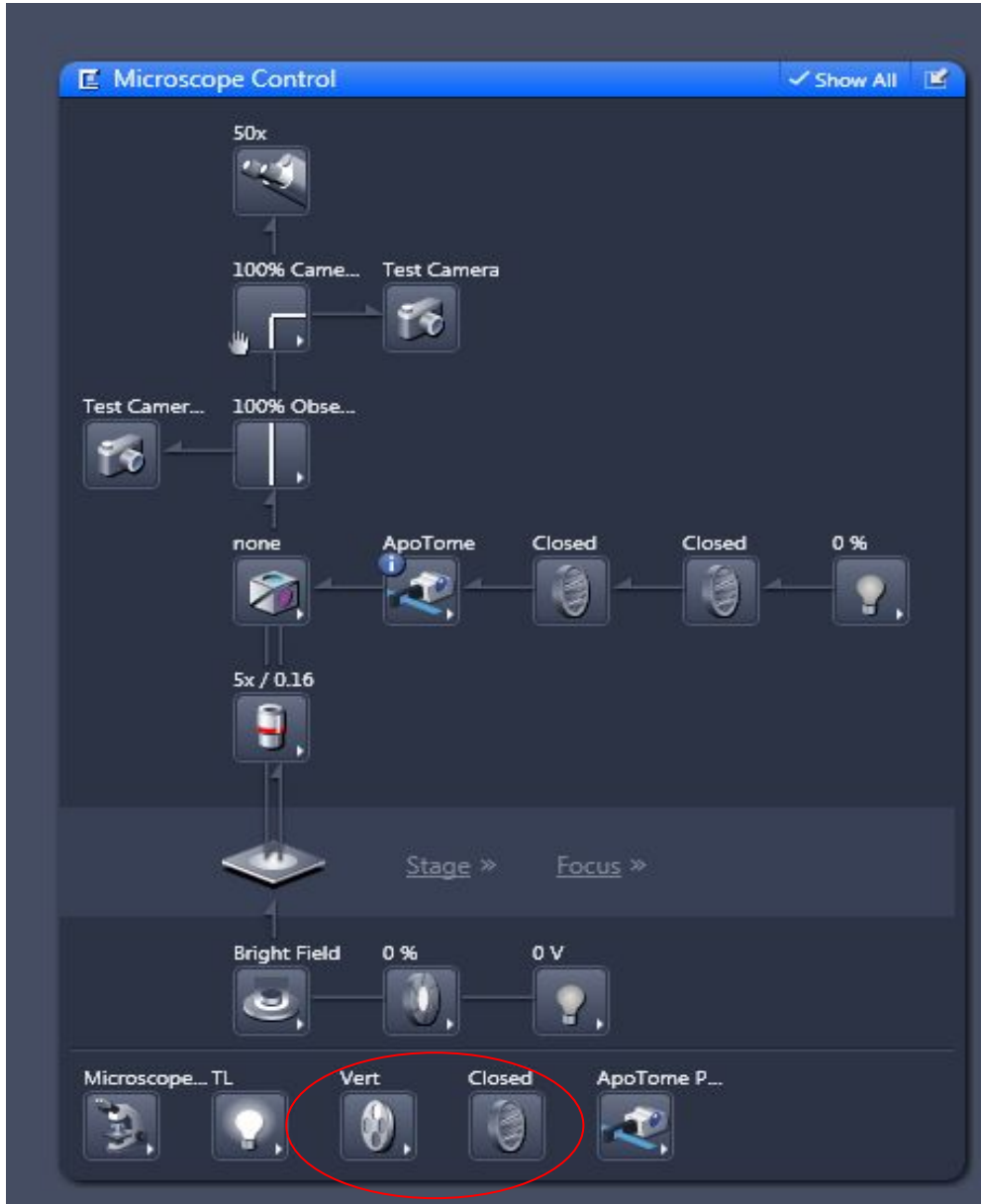
Construction of the objective support

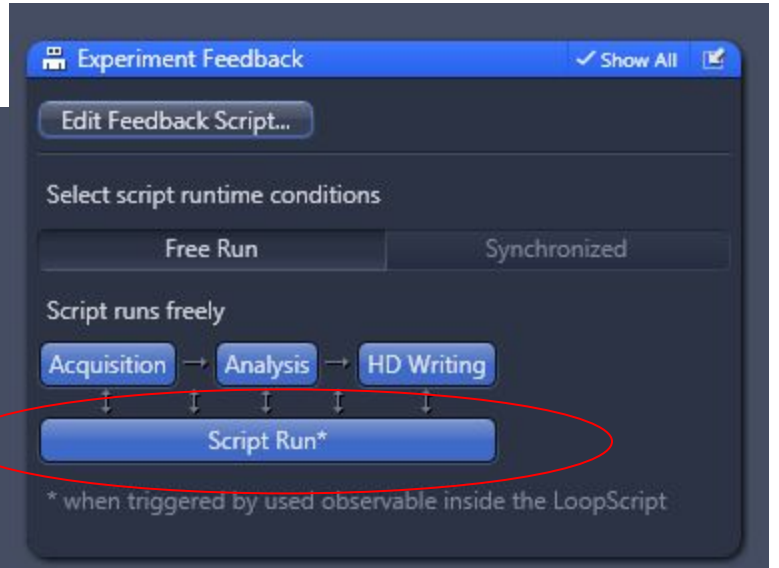


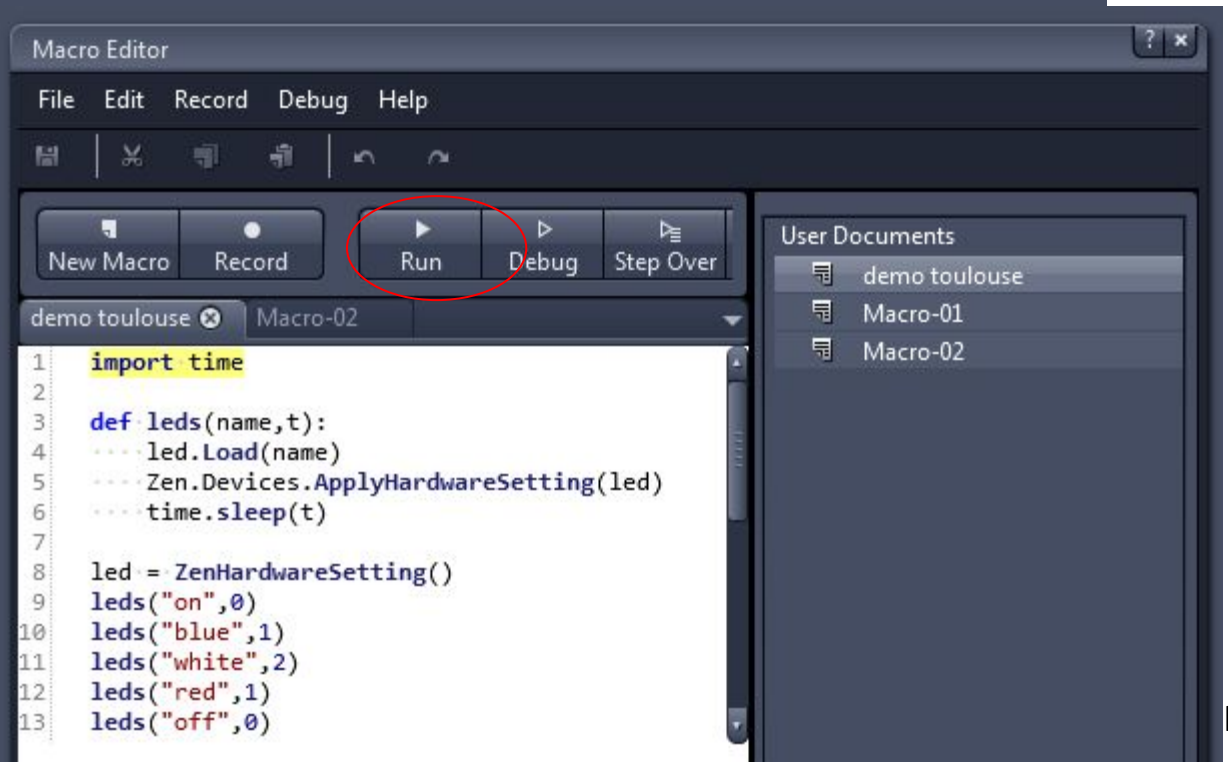




Levels of Zen Blue 2.3 integration







burg

Link to this presentation:

https://docs.google.com/presentation/d/1BlDqrpJWmzF-OCsNnKX5SfcnYt5UuXP2ksjkCaLP0ZQ/edit?usp=sharing

Link to the github repository with example code for emulated devices:

https://github.com/mutterer/MM_CustomArduino