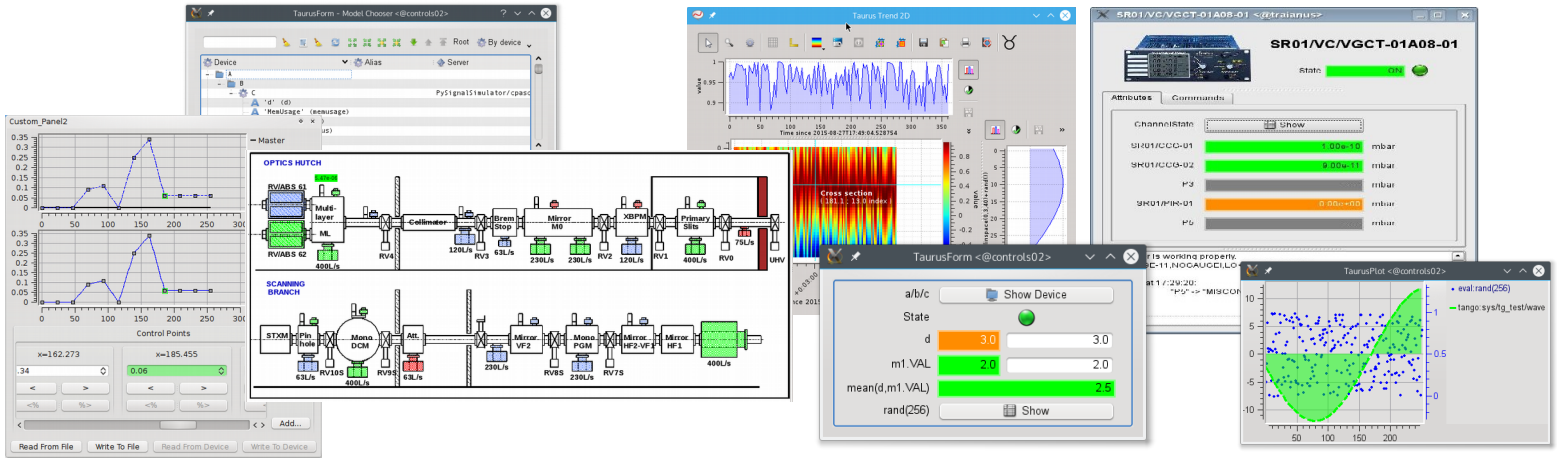
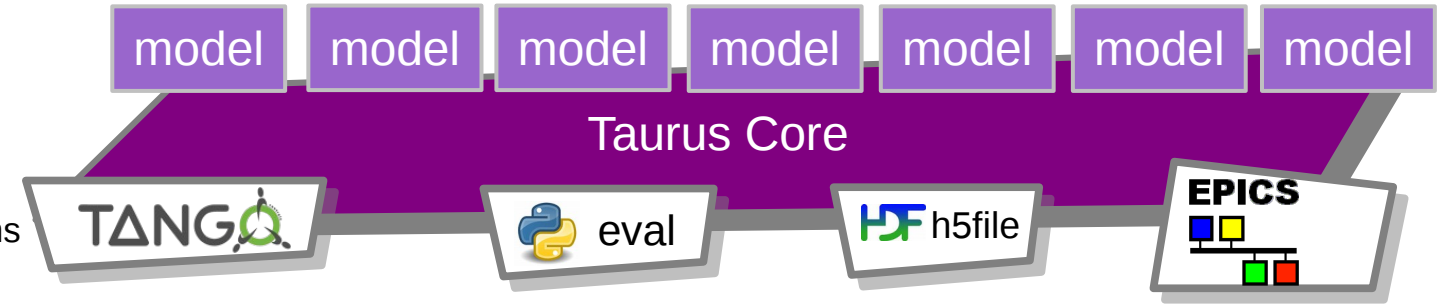
Structure of Taurus

Taurus Qt Widgets



Taurus Core

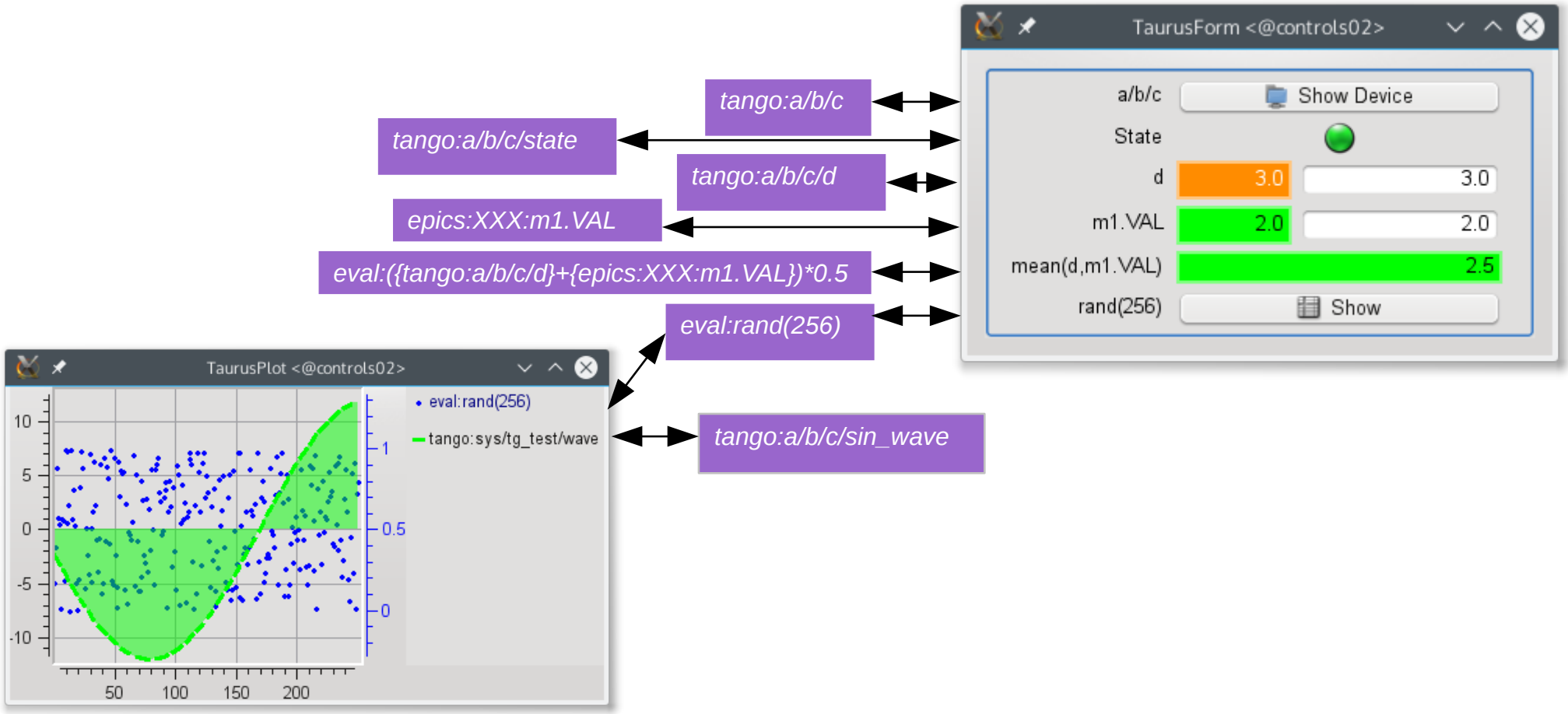
Model Objects



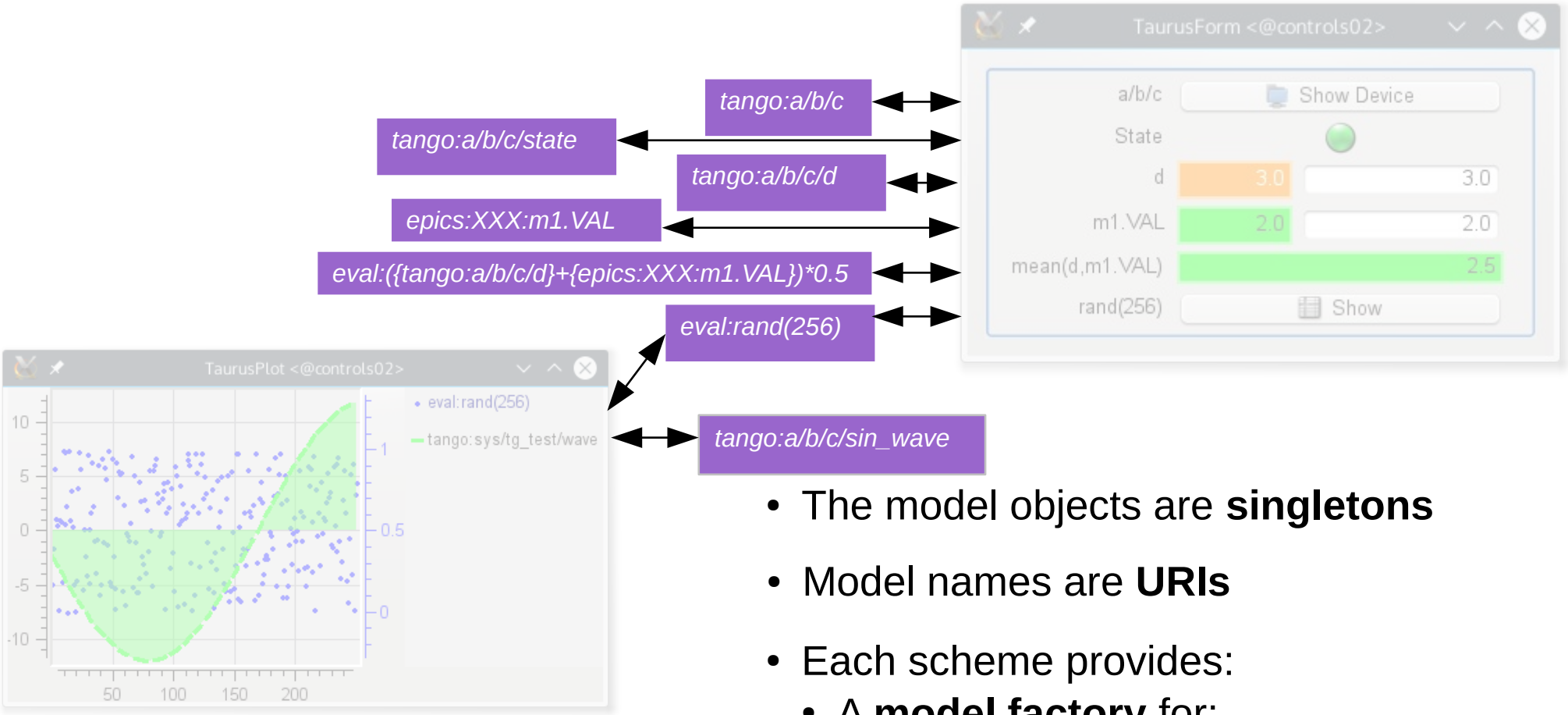
Scheme plugins

Data Sources

Model-View-controller

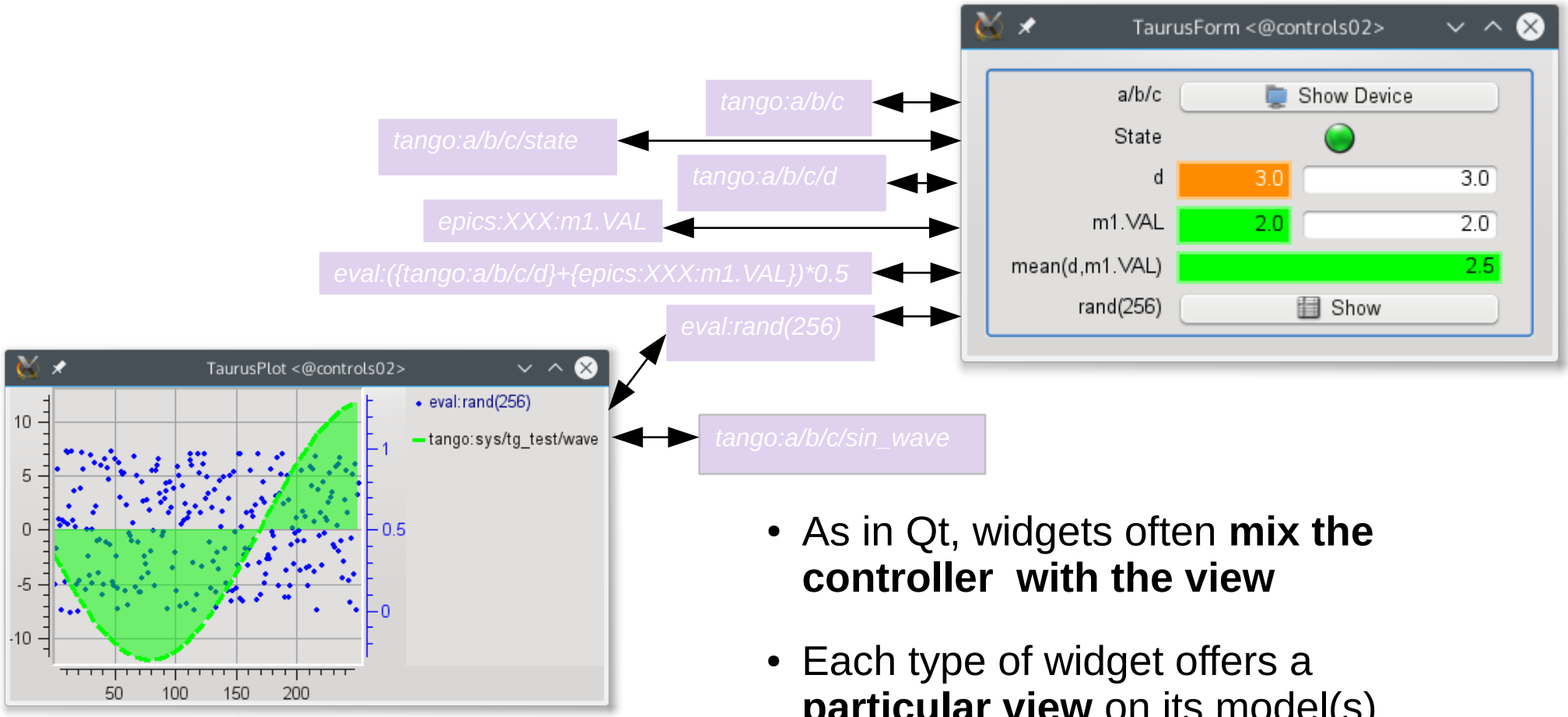


Model-View-controller



- The model objects are **singletons**
- Model names are **URIs**
- Each scheme provides:
  - A **model factory** for:
    - Authority
    - Device
    - Attribute
  - **Model name validators**

Model-View-controller



- As in Qt, widgets often **mix the controller with the view**
- Each type of widget offers a **particular view** on its model(s)
- All functionality is enabled by just **attaching** the widget to a model (i.e. providing its URI)

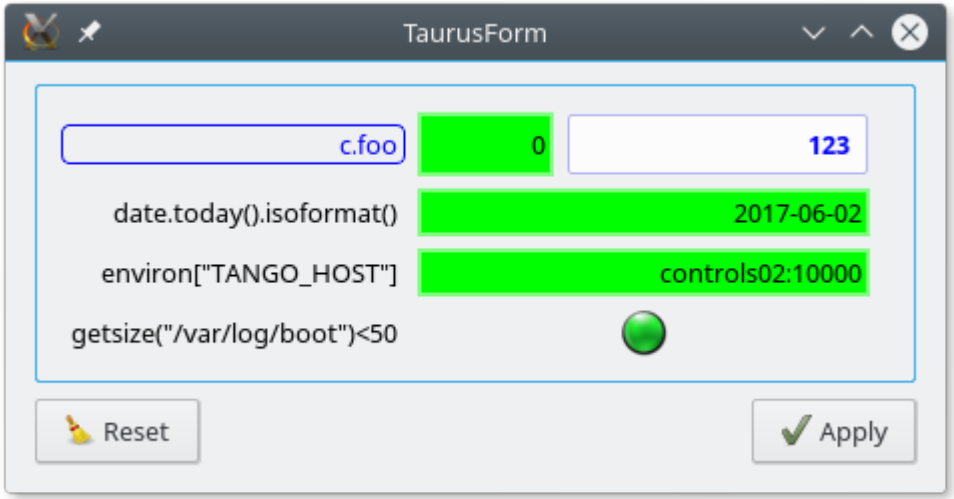
Eval scheme

- Allows mathematical evaluations
- Use **any module or class** as a **custom evaluator**
- Support writable eval attributes


```
$> taurusform 'eval:@c=mymod.MyClass()/c.foo'                    \
              'eval:@datetime.*/date.today().isoformat()'        \
              'eval:@os.*/environ["TANGO_HOST"]'                 \
              'eval:@os.path.*/getsize("/var/log/boot")<50'
```

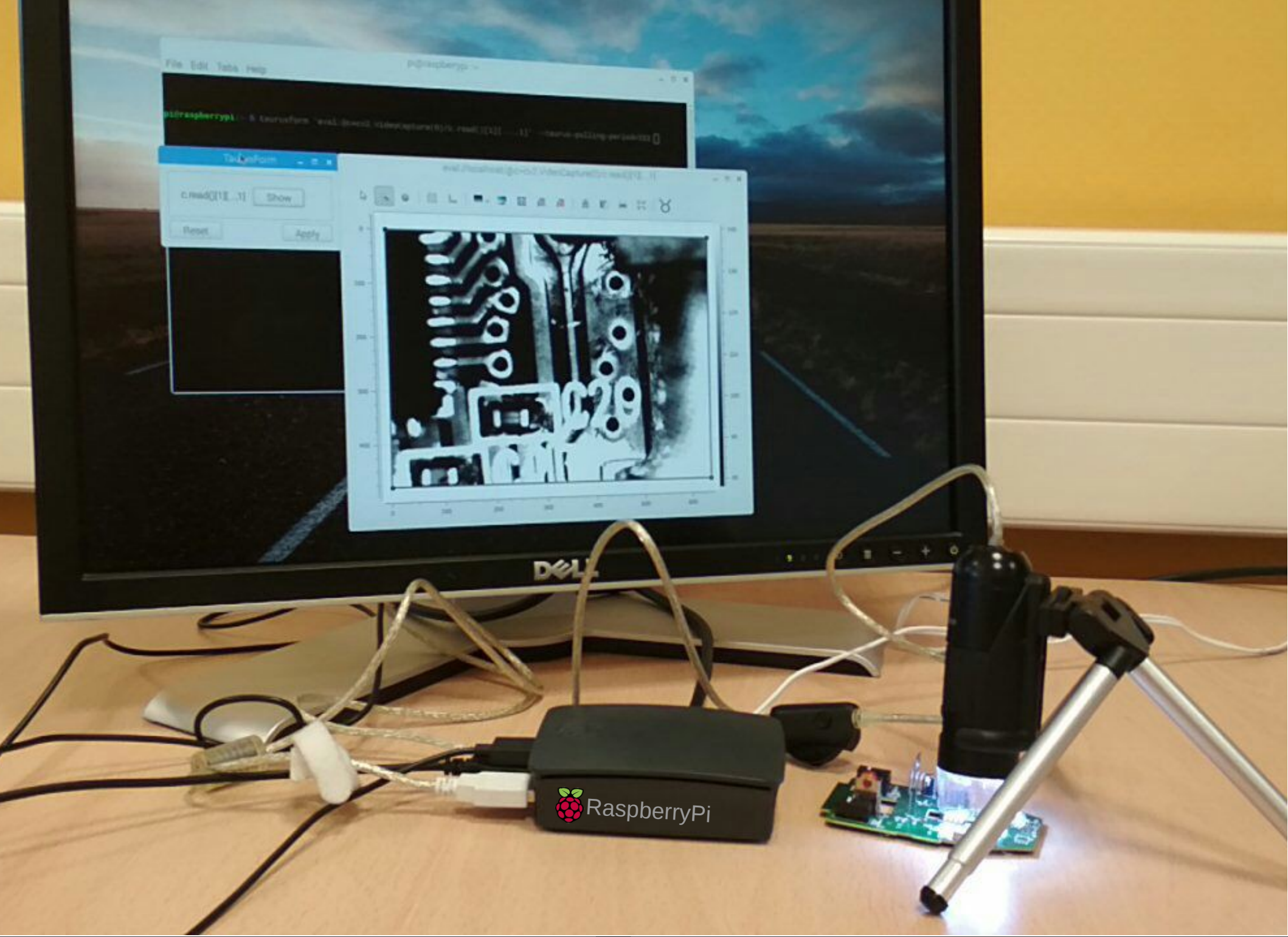
mymod.py

```
class MyClass(object):

    _foo = 0

    def get_foo(self):
        return self._foo

    def set_foo(self, value):
        self._foo = value

    foo = property(get_foo, set_foo)
```

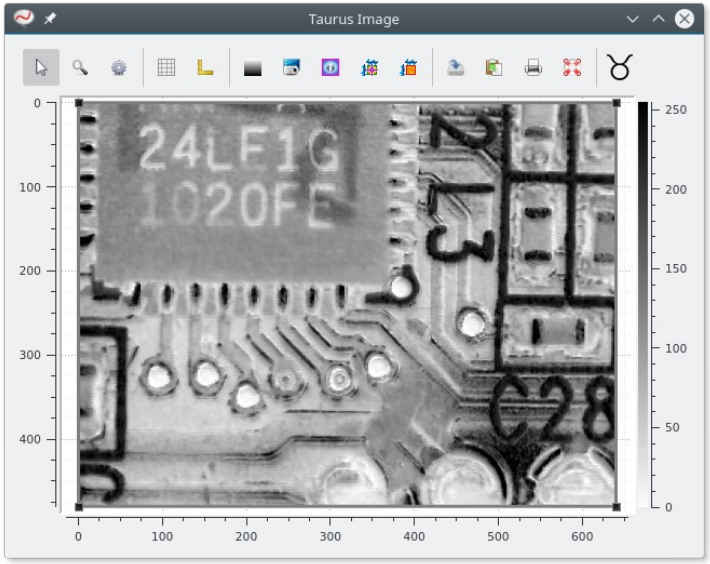


MORE INFO

docs: http://www.taurus-scada.org/devel/api/taurus/core/evaluation.html
mymod example: taurus.core.evaluation.test.res.mymod


File Edit Tabs Help
Show
Reset
Apply
RaspberryPi

Eval scheme: Capturing images from a webcam




```
$> taurusimage 'eval:@c=cv2.VideoCapture(0)/c.read()[1][...,1]'
```