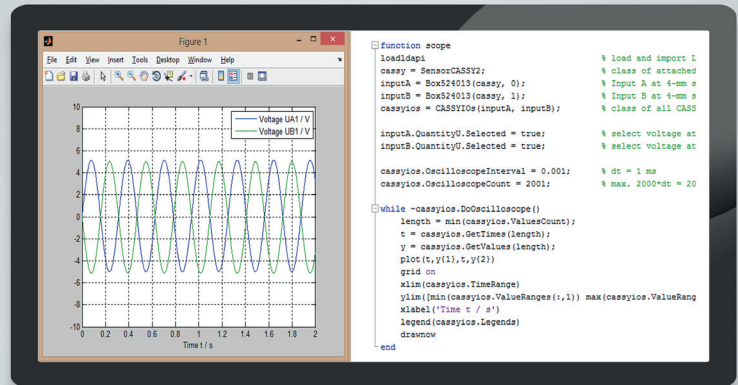
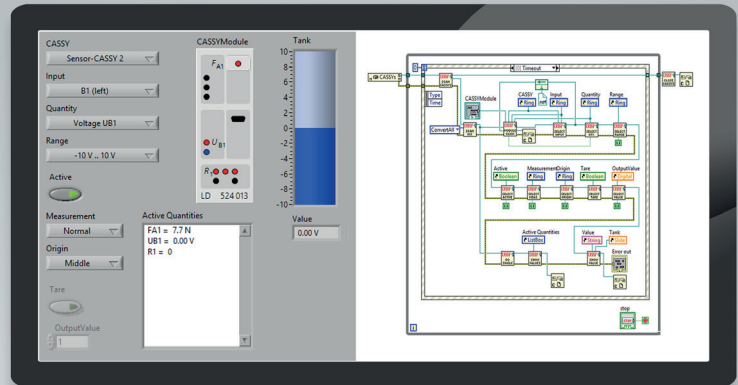
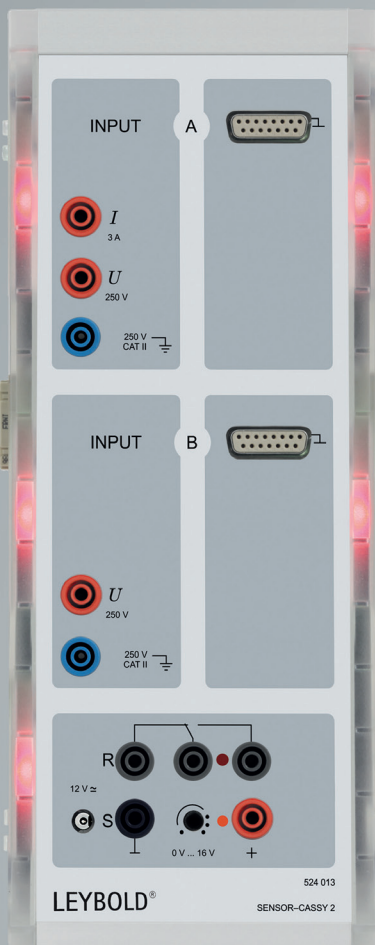


# LEYBOLD®

## CASSY

### SDK, LABVIEW AND MATLAB DRIVERS

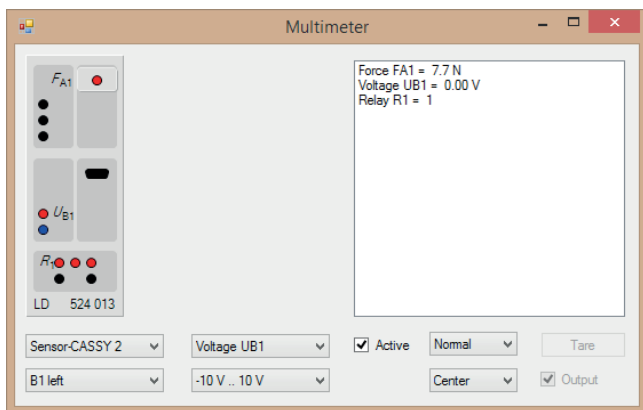


- CASSY SDK for Windows and Linux
- LabVIEW driver for Windows
- MATLAB driver for Windows

# CASSY SDK

LEYBOLD offers software developers free solutions for various development environments:

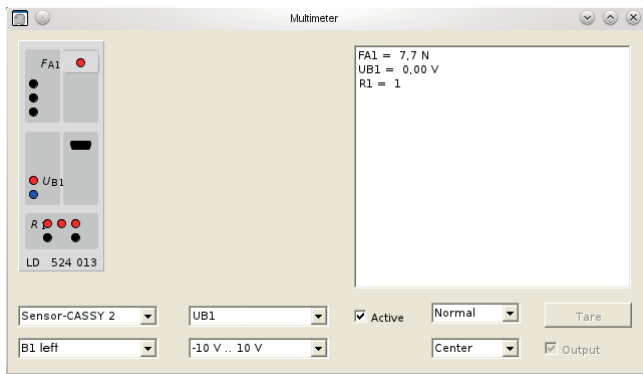
- CASSY SDK for Windows and Linux
- LabVIEW driver for Windows
- MATLAB driver for Windows



The CASSY Software Development Kit includes all classes that you need in order to access all CASSYs, all CASSY sensors and the X-ray apparatus under Windows or Linux. All you need under Windows is a free version of Visual Studio Express, and under Linux you can get started with MonoDevelop. More than 15 examples are provided, making the first steps easy.

Free download at:

<http://www.ld-didactic.com/software/cassysdk.zip>



The ready-to-use multimeter example supports all CASSYs and most sensors and sensor boxes, including all measurement parameters and measurement ranges under Windows as well as under Linux.

# LEYBOLD®

A dual-channel oscilloscope with initialisation and measurement display requires 20 lines. Just a few additional lines are needed for trigger and pre-trigger.

```
using (CASSY cassy = new SensorCASSY2()) // class of attached CASSY
using (CASSYIOs cassyios = new CASSYIOs()) // class of all CASSYIOs
{
    cassyios.AutoDefaults = true; // change oscilloscope defaults automatically
    cassy.Open();

    BoxS24013 inputA = new BoxS24013(cassy, 0); // Input A at 4-mm sockets
    BoxS24013 inputB = new BoxS24013(cassy, 1); // Input B at 4-mm sockets
    cassyios.Add(inputA);
    cassyios.Add(inputB);
    cassyios.Scan(ScanMode.ScanOnly); // sets all SensorBoxValidis

    CASSYQuantity quantityUA = inputA.QuantityU;
    CASSYQuantity quantityUB = inputB.QuantityU;
    quantityUA.Selected = true; // select voltage at input A
    quantityUB.Selected = true; // select voltage at input B

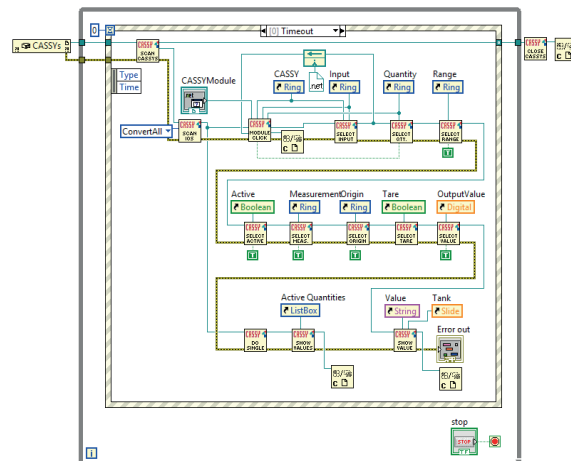
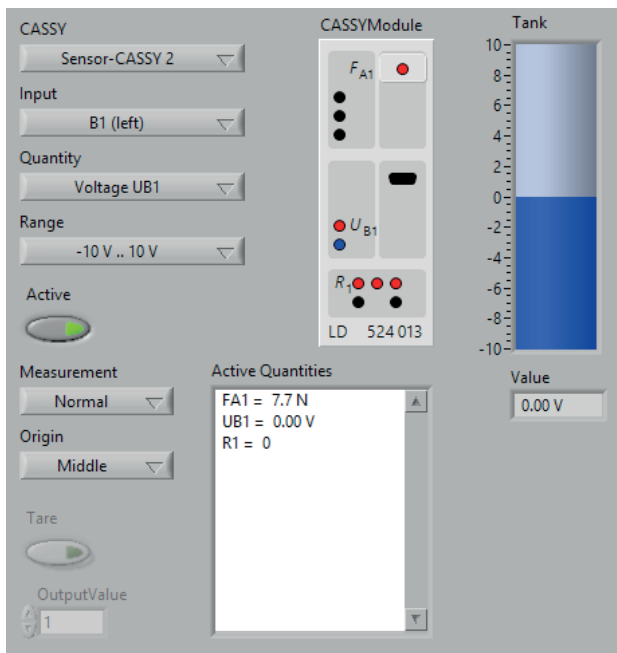
    cassyios.OscilloscopeInterval = 0.001; // dt = 1 ms
    cassyios.OscilloscopeCount = 201; // max. 200*dt = 200 ms

    int i = 0;
    Console.WriteLine("Waiting for trigger signal.");
    while (!cassyios.DoOscilloscope()) // start the oscilloscope and output the values
        for (; i < Math.Min(quantityUA.Values.Count, quantityUB.Values.Count); i++)
            Console.WriteLine("t={0} s\tUA={1} V\tUB={2} V",
                |cassyios.Times[i], quantityUA.Values[i], quantityUB.Values[i]);
}
```

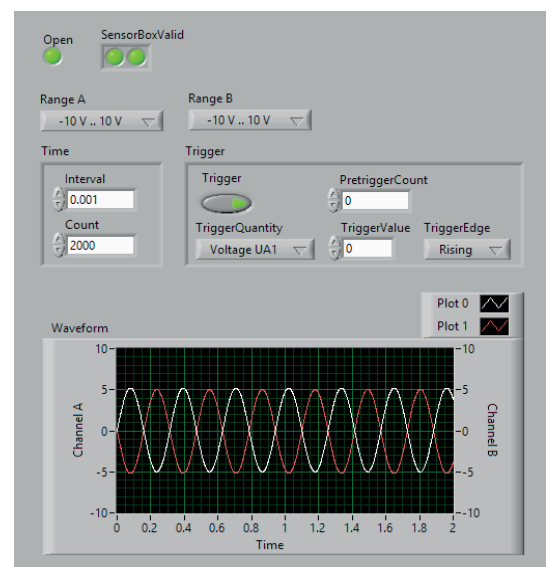
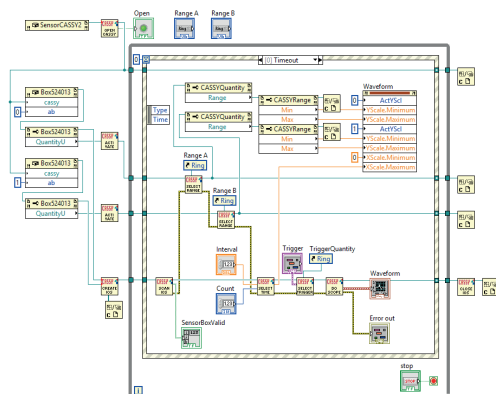
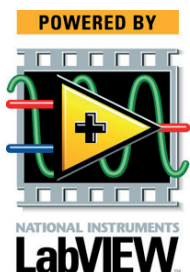
# LABVIEW DRIVER

The LabVIEW driver for CASSY and the X-ray apparatus provides the fully functionality of both instrument families under LabVIEW 2011 for Windows or higher. It extends the classes of CASSY SDK by over 20 virtual instruments which provide access to the functions of CASSYs, sensors and sensor boxes via the graphic programming interface and demonstrates this on the basis of many examples, e.g. multimeter, oscilloscope, function generator, MCA and X-ray apparatus.

Free download at:  
<http://www.ld-didactic.com/software/cassylabview.zip>



The ready-to-use multimeter example supports all CASSYs and most sensors and sensor boxes, including all measurement parameters and measurement ranges.



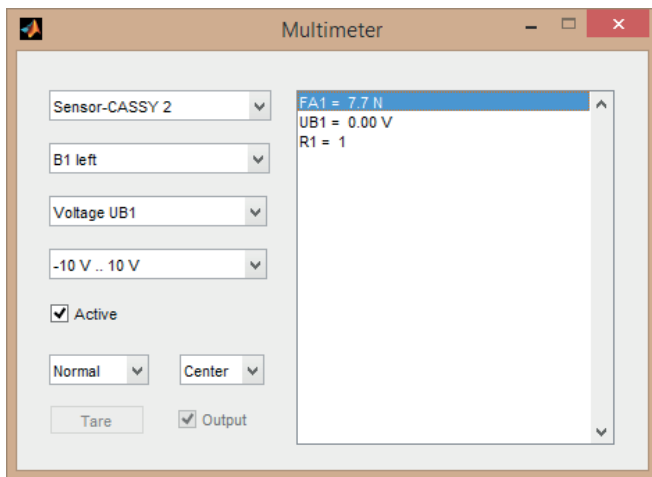
A complete dual-channel oscilloscope with adjustment facility for both measurement ranges and for measurement time, trigger and pre-trigger.

# MATLAB DRIVER

The MATLAB driver for CASSY and the X-ray apparatus provides the full functionality of both instrument families under MATLAB for Windows. It enables rapid implementation of measurement and control tasks, as well as the usual calculation capabilities of MATLAB, also for CASSYs and the X-ray apparatus. More than 15 examples are provided, making the first steps easy.

Free download at:

<http://www.ld-didactic.com/software/cassymatlab.zip>



```
function multimeter
loadldapi % load and import LD.Api.dll
cassys = CASSYs; % find and open all CASSYs

m = multimetergui;
set(m,'Name','Multimeter');
handles = guidata(m);
handles.Multimeter = MultimeterHelper(cassys, handles);
guidata(m, handles); % tell multimeter figure about multimeter class

while ishghandle(m)
cassys.Scan(ScanMode.ConvertAll); % scan for new CASSYs and/or new sensors
cassys.CASSYIOs.DoSingleMeasurement(); % measure current values
handles.Multimeter.ListValues % show current values
drawnow
pause(0.1)
end
```

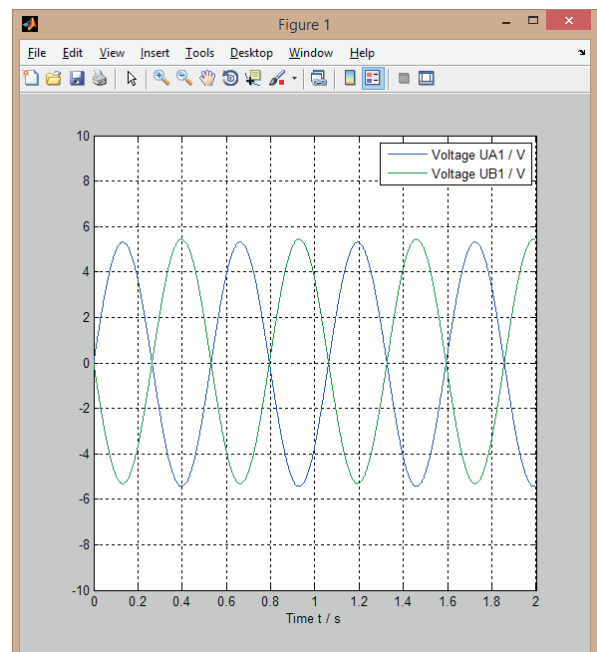
The ready-to-use multimeter example supports all CASSYs and most sensors and sensor boxes, including all measurement parameters and measurement ranges.

```
function scope
loadldapi % load and import LD.Api.dll
cassy = SensorCASSY2; % class of attached CASSY
inputA = Box524013(cassy, 0); % Input A at 4-mm sockets
inputB = Box524013(cassy, 1); % Input B at 4-mm sockets
cassys = CASSYIOs(inputA, inputB); % class of all CASSYIOs

inputA.QuantityU.Selected = true; % select voltage at input A
inputB.QuantityU.Selected = true; % select voltage at input B

cassys.OscilloscopeInterval = 0.001; % dt = 1 ms
cassys.OscilloscopeCount = 2001; % max. 2000*dt = 2000 ms

while ~cassys.DoOscilloscope()
length = min(cassys.ValuesCount);
t = cassys.GetTimes(length);
y = cassys.GetValues(length);
plot(t,y(1),t,y(2))
grid on
xlim(cassys.TimeRange)
ylim([min(cassys.ValueRanges(:,1)) max(cassys.ValueRanges(:,2))])
xlabel('Time t / s')
legend(cassys.Legends)
drawnow
end
```



A dual-channel oscilloscope with initialisation and real-time display can be quickly implemented with 20 clearly organised lines of program code. Just a few additional lines are needed for trigger and pre-trigger.