

How to use XSCOPE in real-time mode

| | |
|-------------|--|
| version | 1.1.0 |
| scope | Example. This code is provided as example code for a user to base their code on. |
| description | How to use XSCOPE in real-time mode |
| boards | Unless otherwise specified, this example runs on the SliceKIT Core Board, but can easily be run on any XMOS device by using a different XN file. |

XSCOPE is fully supported on hardware platforms which provide an XMOS link between the target device and the XSYS development connector.

View the document (XSCOPE overview (see [XM-000957-PC](#))) for further information on tracing data from XMOS applications.

This example provides a simple demonstration of using the XSCOPE continuous event type for data logging from within an XMOS application. The continuous event type can be used to capture and log the value of specific variables within an application to allow debugging. XSCOPE can operate in both a post mortem and a real-time mode where data is streamed back to the host machine and displayed as the application is running. In this example we look at using this mode.

This example assumes you are familiar with creating a run configuration and enabling the associated XSCOPE options in that run configuration in xTIMEcomposer Studio or using the command line tools. For this example the user needs to select the real-time option in the XSCOPE configuration.

In order to use XSCOPE the correct header file must be included in the application

```
#include <xscope.h>
```

The `xscope_probe_data()` function is used to send the contents of user variable values to XSCOPE probe id 0 for real time display. In this example the program loops and continuously outputs events to the XSCOPE system for display in real time

```
while (1) {
    for (i = 0; i < 63; i++) {
        wait(wait_time);
        xscope_probe_data(0, sin_values[i]);
        xscope_probe_data(1, cos_values[i]);
        xscope_probe_data(2, tan_values[i]);
    }
}
```

Using the XSCOPE constructor which gets called automatically by the XSCOPE system register 3 probes using event type XSCOPE_CONTINUOUS which will collect data of type XSCOPE_INT

```
void xscope_user_init(void)
{
    xscope_register(3,
                   XSCOPE_CONTINUOUS, "Sin Value", XSCOPE_INT, "Value",
                   XSCOPE_CONTINUOUS, "Cos Value", XSCOPE_INT, "Value",
                   XSCOPE_CONTINUOUS, "Tan Value", XSCOPE_INT, "Value");
}
```



Copyright © 2013, All Rights Reserved.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.