How to use the XSCOPE to capture logical core information

version	1.0.0
scope	Example. This code is provided as example code for a user to base their code on.
description	How to use the XSCOPE to capture logical core information
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 provide a simple demonstration of using the functionality in XSCOPE to capture additional information about logical core state when tracing events. A user can add data about the id, program counter and node of a logical core generating an event. This allows a user to use register a single event which the XSCOPE system will split into multiple logical core events when processed on the host machine.

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.

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

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

The xscope_probe_data_cpu() function is used to send the contents of user variable current_state to XSCOPE probe id 0 for logging and add the additional information associated with the logical core to the data transmitted.

```
xscope_probe_cpu_data(0, value);
```

Using the XSCOPE constructor which gets called automatically by the XSCOPE system register 2 probes using event type XSCOPE_CONTINUOUS which will collect data of type XSCOPE_UINT. These probes will be used from multiple logical cores in parallel

```
void xscope_user_init(void)
{
  xscope_register(2,
                    XSCOPE_CONTINUOUS, "Continuous Value 1", XSCOPE_UINT, "
                      ↪ Value",
                    XSCOPE_CONTINUOUS, "Continuous Value 2", XSCOPE_UINT, "
                      ↪ Value");
}
```

XMOS®