How to write to a file during execution

version 1.1.1

scope Example. This code is provided as example code for a user to base

their code on.

description How to write to a file during execution

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.

The xTIMEcomposer tools can handle system calls on behalf of the target application. This handling is enabled by default.

A file can be opened for writing as follows:

An open file can be written using the _write system call

```
if (_write(fd, "hello there!", 13) != 13) {
  printstrln("Error: _write failed");
  exit(1);
}
```

We can then close an open file using the _close system call

```
if (_close(fd) != 0) {
   printstrln("Error: _close failed.");
   exit(1);
}
return 0;
}
```

Compile the above code. When the resulting executable is run, a file called test.txt containing 'hello world!' is written to the current working directory.

Note: In this case we are using the raw system call functions directory as we are working in XC. However, if working in C then it might be advisable to the the C std library equivalents: *fopen*, *fwrite* and *fclose*.

Publication Date: 2013/11/15

REV A

XMOS © 2013, All Rights Reserved

Xmos Ltd. is the owner or licensee of the information in this document 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.

