XMOS

Application Note: AN10123

How to write to a file during execution

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to write to a file during execution.

Required tools and libraries

This application note is based on the following components:

- xTIMEcomposer Tools - Version 14.0.0

Required hardware

Programming how-tos are generally not specific to any particular hardware and can usually run on all XMOS devices. See the contents of the note for full details.

1 How to write to a file during execution

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:

```
int fd = _open("test.txt", O_WRONLY | O_CREAT | O_TRUNC, S_IREAD | S_IWRITE);
if (fd == -1) {
  printstrln("Error: _open failed");
  exit(1);
}
```

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*.