

Application Note: AN10122

How to read from a file during execution

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to read from 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 read from 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 reading as follows:

```
int fd = _open("test.txt", O_RDONLY, 0);
if (fd == -1) {
    printstrln("Error: _open failed");
    exit(1);
}
```

An open file can be read using the `_read` system call

```
_read(fd, readBuffer, BUFFER_SIZE);
```

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, then create a file in the current directory named 'test.txt'. When the resulting executable is run, the first 16 bytes of the file will be read and the contents displayed on the console.

Note: In this case the raw system call functions directory is used as you are working in XC. However, if working in C then it might be advisable to use the C std library equivalents: `fopen`, `fread` and `fclose`.