

Application Note: AN10042

How to use overlays and libflash in the same application

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use overlays and libflash in the same application.

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 use overlays and libflash in the same application

When the flash overlay runtime is initialized it takes ownership of the xCORE ports used to access the SPI flash device. This prevents them being used for any other purpose (e.g. reading and writing data using libflash).

If necessary an application can ask the flash overlay runtime to transfer ownership of SPI ports back to the application:

```
f1_SPIPorts * movable spi_ports = overlay_flash_claim_ports();
```

If the overlay runtime is in the middle of loading an overlay the call to `overlay_flash_claim_ports` will block until it has finished loading the overlay.

Once the application has ownership of the SPI ports it can use them to read and write data from the flash using libflash:

```
f1_connect(*spi_ports);  
// Access flash here...  
f1_disconnect();
```

While the application has ownership of the ports a call to function in an overlay that is not loaded will block. Call `overlay_flash_return_ports` to give ownership of the ports back to the overlay runtime:

```
overlay_flash_return_ports(move(spi_ports));
```