

Application Note: AN10019

How to load overlays from a SPI flash

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to load overlays from a SPI flash.

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 load overlays from a SPI flash

An overlay is a block of code and data that is loaded on demand at runtime. Each overlay has a predetermined region of memory that it is copied to called an overlay region. Several overlays may be associated with the same overlay region, but only one of these overlay can be loaded at any one time.

Overlays reduce the amount of memory needed to run your application since it is no longer necessary to reserve space for all your code and data - instead the tools only need to reserve space for the largest overlay that can be loaded into each overlay region.

Specify the `-foverlay=flash` option when building your application to link against the flash overlay runtime. To use this runtime the application must be booted from flash. The runtime loads overlays from the same flash device or flash devices used for boot.

Include the following file:

```
#include <overlay_flash.h>
```

Declare a movable pointer to the SPI ports.

```
f1_SPIPorts spi_ports = {  
    PORT_SPI_MISO,  
    PORT_SPI_SS,  
    PORT_SPI_CLK,  
    PORT_SPI_MOSI,  
    XSI_CLKBLK_1  
};  
  
f1_SPIPorts * movable spi_ports_ptr = &spi_ports;
```

Call `overlay_flash_init` to initialize the flash overlay runtime:

```
overlay_flash_init(move(spi_ports_ptr), 100, 8);
```

The SPI clock frequency in MHz is specified as a ratio using the last two arguments of `overlay_flash_init`. In this case the clock frequency is set to 12.5 MHz (i.e. $100 / 8$).