

Application Note: AN10049 How to control port output data rates with timers

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to control port output data rates with timers.

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 control port output data rates with timers

The rate at which data is output to a port can be controlled by using one of the xCORE's pool of timer resources. This example uses a timer to toggle the output of port toggle_port every 500ms.

```
#define DELAY 5000000 //(50,000,000 * 10ns = 0.5s)
unsigned state = 1, time;
timer t;
t :> time;
while(1) {
    // Output the value of variable 'state' to port 'toggle_port'
    toggle_port <: state;
    // Increment the variable 'time' by the value 'DELAY'
    time += DELAY;
    // Use the timer 't' to wait until this time delay is reached.
    t when timerafter(time) :> void;
    // Toggle the value of variable 'state' and repeat the process again.
    state = !state;
}
```

The ability to control output on a port can also be achieved using the ports time operator @. Note however that the ports time operator is 16-bit whereas the processor timer resource is 32-bit. See example "How to perform timed output on a port" for further information



Copyright © 2016, All Rights Reserved.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") 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.