Application Note: AN10018

A flashing LEDs example

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows a flashing LEDs example.

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 A flashing LEDs example

The simplest flashing LED program loops forever alternating between driving a port high and driving it low. The <: operator drives a value on a port. In between outputs you can make the task pause with the delay_milliseconds function. There are several delay_ functions defined in timer.h.

```
void flashing_led_task1(port p, int delay_in_ms) {
  while (1) {
    p <: 0;
    delay_milliseconds(delay_in_ms);
    p <: 1;
    delay_milliseconds(delay_in_ms);
  }
}
```

The simple example above will block the logical core while waiting between port outputs. This is quite inefficient. To allow other computation to occur in between outputs you need to use timer events. This requires declaring a variable of timer type and then using a select statement which reacts to events on this timer.

```
[[combinable]]
void flashing_led_task2(port p, int delay_in_ms) {
  timer tmr;
  unsigned t;
  // Convert delay from ms to 100Mhz timer ticks
  const int delay_ticks = delay_in_ms * 100000;
  // The value we are going to output
  unsigned val = 0;

  // read the initial timer value
  tmr :> t;
  while (1) {
    select {
    // This case will event when the timer moves past (t + delay_ticks) i.e
    // delay_ticks after when we took the timestamp t
    case tmr when timerafter(t + delay_ticks) :> void:
      p <: val;
      val = ~val;
      // set up the next event
      t += delay_ticks;
      break;
    }
  }
}
```

Note that this function has been marked as [[combinable]]. This means it can share a logical core with other combinable functions that will handle other events in between the port outputs of this flashing LED task.