



Figure 12.3 HPL, HAL, and HIL timer components on the micaz mote.

The two chips, and hence the two storage HPLs have one significant similarity: both chips are accessed via an SPI bus. As a result, each platform must provide glue code that connects the platform-independent (but chip-dependent) HPL code to a platform-specific SPI bus implementation. For the ST M25P, this connection is specified by the platform-specific `HplStm25pSpiC` component, and `Stm25pSpiC` contains platform-independent code with low-level erase, read, write, and CRC computation operations. For the AT45DB041B, each platform must implement the `HplAt45dbC` component which provides low-level erase, read, write, CRC computation, status, and cache-management operations.

HAA structures vary significantly across devices. For instance, Figure 12.3 summarizes TinyOS's support for timers on the micaz platform. At the lowest layer, four HPL components provide access to the four ATmega128 timers. Two of these timers (0 and 2) are 8-bit, and two (1 and 3) are 16-bit. Furthermore, timer 0 supports an "asynchronous" mode where it is clocked by an external (usually 32 768 Hz) crystal, while the other timers are clocked from the same source as the microcontroller, an external 7.37 MHz crystal on the micaz. On the mica, timer 0 is dedicated to implementing TinyOS's timer subsystem, `HilTimerMilliC`:

```
configuration HilTimerMilliC {
    provides interface Init;
    provides interface Timer<TMilli> as TimerMilli[uint8_t num];
    provides interface LocalTime<TMilli>;
}
```

The user-level timer components, `TimerMilliC` (and `LocalTimeMilliC`) are simple portable wrappers over `HilTimerMilliC` (see Section 9.2.1). At the HAL level, the micaz provides 32-bit counters and alarms using  $1/32768$  s and  $\mu$ s time units. The Alarm and Counter interfaces provided by these components are hardware independent, as are the component names (`Alarm32khz32C`, etc.). However, these components need not be provided by all platforms, and the number of available alarms – three in the case of the