Supported Timers

luni64 edited this page on Jul 26, 2023 · 10 revisions

The following table shows available timers for the Teensy boards. Entries show m x c where m is the number of modules and c is the number of channels per module. Modules with entries in parentheses are not yet implemented. Modules for entries with dashes are not available for the board. Details can be found in the corresponding processor datasheets

Timer Module		Width	T-MM
TMR1...TMR4	QUAD Timer	16bit	4x4
GPT1...GPT2	General Purpose Timer	32bit	2x1
PIT1...PITn	Periodic Timer	32bit	1x4
FTM1...FTMn	Flex Timer	16bit	-
RTC	Real Time	n.a.	1x1

Pages 9

Home
Basic Usage
Supported Timers
Avoid PWM timer clashes
Callbacks
Error Handling
Configuration
Examples

Clone this wiki locally

https://github.com/luni64/Teensy

	Clock		
TCK	Tick-Timer	32bit	1x20
TCK64	64 bit Tick-Timer	64bit	1x20
TCK_RTC	64 bit Tick-Timer	64bit	1x20

GPT - General Purpose Timer

The general purpose timer module (GPT) is a 32bit timer module with one timer channel. The controller of the T4.0 boards (IMXRT1062) implements two of those timer modules (GPT1 and GPT2). Currently the GPT modules are not used by the Teensyduino core libraries.

```
// allocate two timers using GPT1
PeriodicTimer t1(GPT1);
PeriodicTimer t2(GPT2);
```

TMR aka QUAD Timer

The QUAD modules (TMR) are 16 bit timer modules with 4 timer channels per module. Teensy 4.0 controller has four TMR modules (TMR1 ... TMR4). The Teensyduino core uses TMR1-TMR3 for generating PWM signals. Using one of those will disable the PMW capability of the corresponding pins.

```
PeriodicTimer t1(TMR1); // first
OneShotTimer  t2(TMR1); // next free c
PeriodicTimer t3(TMR3); // first free
...
```

PIT - Periodic Timer

The PIT timer modules differ between the T3.x and the T4.x boards.

Teensy 4.x

The processor of the T4.x and Teensy MicroMod boards has one PIT module with four 32bit channels. All four channels share the same interrupt.

Usage:

```
PeriodicTimer t1(PIT), t2(PIT); /
OneShotTimer  t3(PIT);          // ini
...
```

Note: The PIT module is clocked with either 24MHz (default) or FBUS (usually 150MHz). You can configure the used clock as described in the configuration chapter.

Teensy 3.x

There are 4 dedicated PIT modules, each of them has one 32bit channel with its own interrupt and runs independently of the other channels. **The T3.x PITs are not yet implemented**. The stock IntervalTimer also uses the PIT modules and can be used instead.

FTM - Flex Timer Module

The FTM modules are only available for the T3.x boards. The number of modules available depends on the board type. Each module has a various number of 16bit timer channels.

Usage:

```
PeriodicTimer t1(FTM0), t2(FTM0);  F
OneShotTimer  t3(FTM1);            // F
...
```

RTC - Real Time Clock

The RTC module uses the built in real time clock to generate periodic interrupts.The interrupt periods are limited to 1s, 0.5s, 250ms, 125ms, ..., 30.51µs. The interrupts are synchronized with the clock. E.g. the 1s interrupt is invoked exactly at the second transitions of the clock.

```
...
PeriodicTimer t1(RTC);

void setup()
{
    pinMode(LED_BUILTIN, OUTPUT);
    t1.begin([] { digitalToggleFast(LE
}
...
```

TCK - Tick Timer

The tick timer is a 32bit software timer. Instead of using one of the built in hardware timer modules it relies on calling a `tick()` function as often as possible. The `tick()` function checks the cycle counter to determine if the callback function needs to be called. In the default configuration, calling `tick()` is automatically handled by TeensyTimerTool in the `yield()` function. Thus, you can use the tick timer in exactly the same way as the hardware timers. You can configure the way how the `tick()` function is called in the config file. See here for the corresponding description.

Usage:

```
PeriodicTimer t1(TCK), t2(TCK);
OneShotTimer  t3(TCK);

void setup()
{
    t1.begin(callback1, 200);
    t2.begin(callback2, 20'000'000);
    t3.begin(callback3);
}
```

Advantages:

- Since the tick timers don't need any hardware resources, you can allocate as many timers you need. However, to allow static memory allocation TeensyTimerTool currently limits the total number of tick timers (TCK, TCK64, TCK_RTC) to 20 (that will be settable later).
- Due to the weak coupling of peripheral to the ARM core the hardware timers of the Teensy 4.0 are not very efficient. (e.g. https://forum.pjrc.com/threads/57959-Teensy-4-IntervalTimer-Max-Speed). Given the very fast ARM core of the T4.0 controller the tick timers can be more effective and faster than the hardware timers on this board.

Caveats: The tick timers only work if the TeensyTimerTool can call the *tick* function with a high frequency. This means, if you can not avoid long blocking tasks the tick timers might not work for you. (Please note: `delay()` or other functions calling `yield()` in the background are perfectly fine to use.)

TCK64 - 64bit Tick Timer

The TCK64 timer is a 64bit software timer. It works exactly like the normal TCK timers but extends the internally used counter variable to 64bit. Thus, periods of more than one hour (depending on the clock frequency) are possible.

TCK_RTC - 64bit Real Time Clock Tick Timer

The TCK64_RTC timer is a 64bit software timer which uses the RTC module of the T4x boards as time base. Thus, the attached callback function will be invoked in sync with the RTC which might be useful in real time applications. Usage is identical to the other TCK timers.

TeensyTimerTool - Generic Interface to Teensy Timers