External wakeup

Demonstrates the usage of SAMD chip to furtherly reduce the power usage of Tian board. This method can be applied to any board with companion chips which expose a method (via direct pin interrupt or via a command) to enter and exit standby.

Hardware Required

- Arduino Tian

Code

```
/*
TianStandby

This sketch demonstrates the usage of SAMD chip to furtherly reduce the power usage of Tian
board. This method can be applied to any board with companion chips which expose a method
(via direct pin interrupt or via a command) to enter and exit standby.
Sleep modes allow a significant drop in the power usage of a board while it does nothing waiting for an event to happen. Battery powered application can ta

In this sketch, the internal RTC of SAMD chip will wake up the processor every 20 seconds.
Before going to sleep, the SAMD chip tells the MIPS CPU to standby too.
Please note that, if the processor is sleeping, a new sketch can't be uploaded. To overcome this, manually reset the board (usually with a single or double

This example code is in the public domain.
*/

#include <ArduinoLowPower.h>

const int MIPS_PIN 32

void MIPS_PM(bool sleep) {
 pinMode(MIPS_PIN, OUTPUT);
 digitalWrite(MIPS_PIN, sleep ? LOW: HIGH);
}

void setup() {
 pinMode(LED_BUILTIN, OUTPUT);
LowPower.companionLowPowerCallback(MIPS_PM);
 // Uncomment this function if you wish to attach function dummy when RTC wakes up the chip
LowPower.attachInterruptWakeup(RTC_ALARM_WAKEUP, onWakeup, CHANGE);
}

void loop() {
 digitalWrite(LED_BUILTIN, HIGH);
 delay(500);
 digitalWrite(LED_BUILTIN, LOW);
 delay(500);
 // Triggers a 2000 ms sleep (the device will be woken up only by the registered wakeup sources and by internal RTC)
 // The power consumption of the chip will drop consistently
 // Send sleep command to MIPS CPU and then go to sleep
LowPower.companionSleep();
LowPower.sleep(20000);
}

void onWakeup() {
 // This function will be called once on device wakeup
 // You can do some little operations here (like changing variables which will be used in the loop)
 // Remember to avoid calling delay() and long running functions since this functions executes in interrupt context

 // Wakeup the companion chip, too
LowPower.companionWakeup();
}
```