

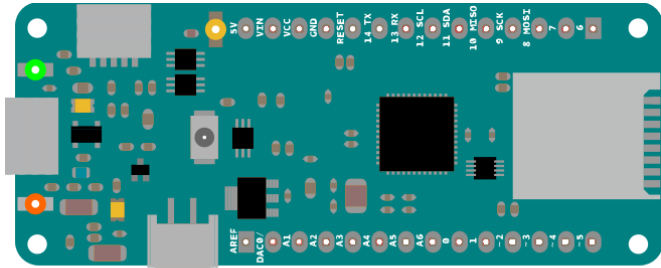
This example for a SAMD21 based board allows demonstrates the usage of Internal Interrupts to wakeup a chip in sleep mode. In this example, the internal RTC will wake up the processor every 2 seconds. Please note that, if the processor is sleeping, a new sketch can't be uploaded. To overcome this, manually reset the board (usually with a single or double tap to the RESET button).

## Hardware Required

- SAMD21 based board (i.e MKR1000, MKRZero, Zero, Tian)

## Circuit

There is no circuit for this example.



fritzing

## Code

```
/*  
TimedWakeup
```

This sketch demonstrates the usage of Internal Interrupts to wakeup a chip in sleep mode. 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 c

In this sketch, the internal RTC will wake up the processor every 2 seconds. 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 c

```
This example code is in the public domain.  
*/
```

```
#include <ArduinoLowPower.h>
```

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
  // Uncomment this function if you wish to attach function dummy when RTC wakes up the chip  
  // LowPower.attachInterruptWakeup(RTC_ALARM_WAKEUP, dummy, 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  
  LowPower.sleep(2000);  
}  
  
void dummy() {  
  // 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  
}
```