# Arduino, Zigbee and Embedded Development

A helpful blog designed to get beginners up and running with Arduino, Zigbee and general embedded development.

Sunday, 25 April 2010

## Sleeping Arduino - Part 1

### Overview

In this series of blogs I will be describing how to put an Arduino Diecimila into sleep mode, thus reducing the power consumption of the device, and detail several mechanisms for waking the Arduino. This can be quite useful (even necessary) when you are powering your Arduino via a battery and/or solar panel.

Four mechanisms for waking the Arduino from sleep will be covered:

- **via an external interrupt**. The Diecimila will wake up only when an external interrupt occurs.
- **via the UART** (USB serial interface). The Diecimila will remain asleep until data is received over the serial interface.
- **via an internal timer**. The Diecimila will periodically be woken up from sleep via Timer1, carry out an action and go back to sleep.
- **via the watchdog timer**. The Diecimila will periodically wake up from sleep via the Watchdog timer, carry out an action and go back to sleep. Note using the Watchdog for this provides the longest sleep time and lowest power consumption (see here).

Note: There are various versions of Arduino available (see the I/O Board section of here), we will be using the Arduino Diecimila with an ATmega168 micro-controller. The source code provided below may run on other versions of Arduino, but the power consumption values will most likely be different.

This tutorial assumes the user is able to run the Arduino environment, create sketches and run them on the USB based Arduino Diecimila. For further information please see the official Getting Started with Arduino web page.

### Arduino Power Consumption

There are several devices on the Arduino Diecimila that consume battery power, including:

- ATmega168 micro-controller
- FT232RL USB UART
- The power regulator

The Arduino Diecimila I have uses about 35mAmps during normal operation and in power-down sleep mode about 15mAmps. There isn't a huge difference here, the main problem is that the power regulator draws 10mAmps, irrespective of the sleep state the Arduino is in. The ATmega168 micro-controller draws about 0.05 mAmps when in power-down sleep mode and 20mAmps during normal operation.

A cheap 9 Volt alkaline battery will have a typical capacity of 565 mAh. If used to power our Arduino Diecimila, it will last for about

- **16 hours** (565mAh/35mA) under normal operation
- up to almost **38 hours** (565mAh/15mA) in power-down sleep mode, depending on how often and for how long the Arduino is woken from sleep mode.

If you require your battery to last longer, you would have to consider the following options:

- Use a battery with a bigger capacity, e.g. 9 Volt Lithium batteries have a typical capacity of 1200mAh, or you could use a battery pack, made up of several batteries.
- Use a solar panel with/without a battery pack. If a battery pack is used, it can be charged by the solar panel during daylight hours.

About Me

**Donal Morrissey**

Senior embedded software engineer and team leader with experience in the telecommunications and automotive industries. Android application developer during my spare time. This includes the continued development of 'My Migraine Log', which is an application for tracking a user's migraine attack history. And a similar app called 'My Headache Log'. Both are currently on sale on the Android market. Specialties o Embedded software development o Android application development o SCRUM o Real-Time Operating Systems o Linux software development (embedded & desktop) o Microchip PIC16/PIC24 firmware development o Bootloaders, including fault tolerant remote update of application software o Embedded RF system software o Automotive embedded software o CAN Gateways & networks o Hardware interfacing (I2C, SPI, CAN, Etc) o Zigbee o C, C++, JAVA, Perl, Python o QT o Team Leadership o Recruitment

View my complete profile

Site Map

Site Map

- Use the ATmega168 micro-controller stand-alone outside of the Arduino, see here.

Also note that the power consumption will be influenced by any external circuit that is connected to the Arduino, e.g. if you are interfacing with a bread board or using an Arduino shield.

## Arduino Sleep Modes

The ATmega168 micro-controller in our Arduino Diecimila supports several modes of sleep:

- SLEEP_MODE_IDLE - the least power savings
- SLEEP_MODE_ADC
- SLEEP_MODE_PWR_SAVE
- SLEEP_MODE_STANDBY
- SLEEP_MODE_PWR_DOWN - the most power savings

The more power saving the sleep mode provides, the less functionality is active.
E.g. in **Power-Down** sleep mode, only the external interrupt and watch dog timer (WDT) are active, in **Idle** sleep mode the UART, timers, ADC, etc are all active, just the CPU and Flash clocks are disabled. See Section 9 of the ATmega168 datasheet for more information.

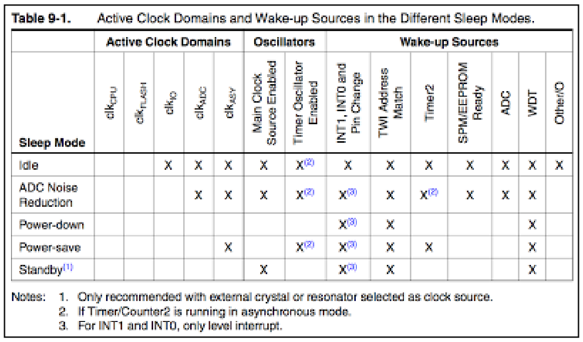

**Table 9-1.** Active Clock Domains and Wake-up Sources in the Different Sleep Modes.

| | Active Clock Domains | | | | | Oscillators | | Wake-up Sources | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Sleep Mode** | $clk_{CPU}$ | $clk_{FLASH}$ | $clk_{IO}$ | $clk_{ADC}$ | $clk_{ASY}$ | Main Clock Source Enabled | Timer Oscillator Enabled | INT1, INT0 and Pin Change | TWI Address Match | Timer2 | SPM/EEPROM Ready | ADC | WDT | Other/O |
| Idle | | | X | X | X | X | X[2] | X | X | X | X | X | X | X |
| ADC Noise Reduction | | | | X | X | X | X[2] | X[3] | X | X[2] | X | X | X | |
| Power-down | | | | | | | | X[3] | X | | | | X | |
| Power-save | | | | | X | | X[2] | X[3] | X | X | | | X | |
| Standby[1] | | | | | | X | | X[3] | X | | | | X | |

Notes: 1. Only recommended with external crystal or resonator selected as clock source.
2. If Timer/Counter2 is running in asynchronous mode.
3. For INT1 and INT0, only level interrupt.

When the micro-controller is entered into sleep mode by your code, the execution of code will pause at that point. In order to resume execution of your code, the micro-controller must then be woken from sleep mode by one of it's internal hardware modules, e.g. timer expiring, external interrupt, WDT, etc.

There are several Arduino library functions used to control sleep mode. They are:

- set_sleep_mode(mode) - Configures the Atmega168 for the specified sleep mode (see above for supported sleep modes);
- sleep_enable() - Enables the sleep mode to be entered;
- sleep_mode() - Enters the sleep mode. Before this is called, the appropriate mechanism for waking the microcontroller must have been set up;
- sleep_disable() - Disables the sleep mode;

The following is the basic code needed to put the Arduino into a sleep mode:

```
void enterSleep(void)
{
 set_sleep_mode(A_SLEEP_MODE);
 sleep_enable();
 sleep_mode();
 /** The program will continue from here. **/
 /* First thing to do is disable sleep. */
 sleep_disable();
}
```

Note that once the mechanism to wake the device from sleep mode has occurred, execution of code will continue, starting at the next code statement after the sleep_mode() function call.

## Waking the Arduino

The following entries cover the various approaches available to bringing the Arduino out of a sleep mode:

- Part 2 Wake Up Via An External Interrupt
- Part 3 Wake Up Via the UART
- Part 4 Wake Up Via Internal Timer
- Part 5 Wake Up Via The Watchdog Timer

If you would like to use an internal timer to wake the Arduino, and don't know which one to use, have a look at the table I've put together here.

## Resources

http://interface.khm.de/index.php/lab/experiments/sleep_watchdog_battery/
http://rubenlaguna.com/wp/2008/10/15/arduino-sleep-mode-waking-up-when-receiving-data-on-the-usart/

Posted by Donal Morrissey at 11:47

## 21 comments:

**yekorzh** 19 October 2010 at 20:40

Hi, great article!

I'm wondering if you had a chance to finish internal and watchdog timer sections? It will make this series of posts as a best arduino power saving tutorial available on the Internet.

Reply

**Trystan Lea** 15 February 2011 at 06:21

Thanks for this article, very clear

Reply

**piniflopa** 23 July 2011 at 23:37

Many thanks for so clarifiyng article!

Reply

**Donal Morrissey** 24 July 2011 at 08:42

No problem. I must get around to finishing the series....

Reply

**Avishaan** 10 September 2011 at 21:34

Oh man, this would have been perfect if you had time to finish explaining it via the watchdog or internal timer. Now gotta get back on google, oh well i guess.

Reply

**rmurrish** 2 November 2011 at 12:26

Please finish this series. This is the clearest example in the universe on how to sleep the arduino. Complete the series and the Arduino world will thank you.

Reply

**Donal Morrissey** 2 November 2011 at 13:11

@rmurrish Yes, I really need to complete this! I see what I can do over the next few weeks.

Reply

**DoktorJ** 8 November 2011 at 15:21

Great article, though I'm definitely looking forward to the internal interrupt part. I really need this for a current project and I can find no useful information on the topic.

Reply

**Alan** 16 November 2011 at 05:45

Yea, it's really a good job already done! Like everybody, I'm searching for this way to wake my arduino up from internal way.
Your completion of these series will be great for so many of us ! Thank you very much for your work Donal! Great !

Reply

**Donal Morrissey** 26 November 2011 at 11:25

Hi All,
I've just completed "Part 4: Wake Up Via Internal Timer". Part 5 is on the way.
Donal

Reply

**Dat** 7 August 2012 at 01:48

Thank you very much for your articles. It is very helpful for my project.

Reply

**Tyrant lover** 22 November 2012 at 19:44

Can you please tell me if I can set the Arduino to gather and save data while in sleep mode?

Reply

Replies

**Donal Morrissey** 23 November 2012 at 04:58

What type of data do you want gather and dave?

**Reply**

**ambuj saxena** 22 May 2013 at 23:46

can we awake mcu from spi mode?? if yes so how could i do this

Reply

**Faye** 9 April 2014 at 08:32

Found this so useful for my current project, thank you!

Reply

**luo XW** 10 June 2014 at 11:01

I did a solar tracker using arduino. but the board draw too much power. so I want put arduino in sleep mode when sunlight is not strong. thanks for your BLOG.
I am a liitle disappointed that sleep mode only save 20mA. if I put arduino in sleep for 12hr/day. then I could save 240mAhr/day.

Reply

**BALA KRISHNA JULURI** 14 November 2014 at 13:01

Very useful and clear. Thanks!

Reply

**Atrius** 1 March 2015 at 16:43

I am using the Capacitive Sensing Library to create capacitive touch pins (as per http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense), and am wondering what levels of sleep, if any, they would be able to bring the Arduino out of.

Any help would be greatly appreciated.

Reply

**Atrius** 1 March 2015 at 16:43

I am using the Capacitive Sensing Library to create capacitive touch pins (as per http://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense), and am wondering what levels of sleep, if any, they would be able to bring the Arduino out of.

Reply

**johnleehooker** 17 May 2016 at 15:42

how can i extend sleep time a day or a month? i want to change sleep time depends on the analog input.for example if analog input is 200 (assume 0-1023) then want to sleep mc 1 day(24 hours)or if it is 1000 sleep 10 minutes.Any suggestion?

Reply

**Donal Morrissey** 18 May 2016 at 04:42

Dear John,<br>
Using just the arduino, I'm not aware of any mechanism to extend the sleep time beyond the max of 8 seconds provided by the Watchdog (see part 5 of the tutorial above).
Using the Watchdog: each time you wake up, you could read the analog input, and then decide what to do.
Donal

Reply

```
Enter your comment...
```

**Comment as:** Select profil ▼

Publish Preview

Newer Post Home Older Post

Simple theme. Powered by Blogger.