*Projects and Stuff*

Beer Pong Sense

Project Log

Contents

[2012/04/21 3](#_Toc323472808)

[2012/04/22 3](#_Toc323472809)

[2012/04/24 3](#_Toc323472810)

[2012/04/25 3](#_Toc323472811)

[2012/04/28 4](#_Toc323472812)

[2012/04/29 4](#_Toc323472813)

# 2012/04/21

Restarted work on the project after a long hiatus (military deployments make engineering difficult).

# 2012/04/22

Began working BOM based on parametric information from Atmel and my past QTouch experience. BOM will focus on parts to be used in both the test board and final design.

BPS needs a better name. Going to have to brainstorm. Beer Pong Sense works well as a code name of sorts, like Microsoft's Longhorn, but it doesn't have much of a ring to it.

# 2012/04/24

Today began fleshing out the BOM based on known needs. For instance, I know I will require the following basic parts:

* **Battery Controller (currently looking for a rechargeable Li-Ion battery at around 5V and 2000mAh)**
* **AVR Microcontroller capable of QMatrix/QTouch and with a good amount of memory and peripherals for hacking**
* **Display drivers for the 47+ RBG LEDs used in this project**
* **A Voltage Regulator that can source enough current for all of the LEDs and other components**

I used the parametric search tools on Digikey to start finding components that would meet my needs. For instance, if I assume each LED (3x47=141 LEDs due to 3 LEDs per RGB) is 10mA, that’s 141\*0.01= 1.41A – A lot of current. Now in reality, we won’t allow all the LEDs to be on at once, and we’ll use PWN or other methods to adjust their brightness, so the end result will be much less than 1.41Amps, but we’ll still be using a lot of current.

Once these parts have been identified and I’ve gone through the datasheets to verify that everything meets my needs and will be compatible, I’ll move on toward starting the basic schematics in KiCad. I prefer KiCad over Cadsoft Eagle for several reasons, including the fact that KiCad is open source, and it doesn’t limit users to a specific board size.

# 2012/04/25

The plan right now is to use the Atmel AVR ATMEGA164/324/644 Series Microcontrollers along with Atmel’s QTouch Library (utilizing QMatrix technology) to sense the liquid in cups placed on the board, and then use three AS1107WL display drivers to light up LEDs corresponding to the location of each cup.

There are two possible configurations for using these display drivers

1. Each AS1107WL will control one color of all RGB LEDs

Pro: Should be pretty simple to program.

Con: Each RGB LED is driven from 3 different display drivers (one driver for each of the 3 LEDs in an RGB LED). This will likely greatly increase the complexity of PCB routing.

1. Utilize 6 of the 8 segments on each AS1107WL, controlling 2 sets of 8 RGB LEDs with each display driver.

Pro: Would likely make PCB routing much simpler, since each RGB LED is controlled from only one display driver. Likely pretty easy to program.

Con: Would have to add additional external pots to each drive line to adjust the current for the different colors of LEDs, increasing cost. Leaves 2 segments of each driver unutilized (though this could also prove to be a pro).

I didn’t realize Atmel Studio 6 Beta is now out. Once I start coding, I may use this, since (from what I’m reading) it already has the Touch libraries integrated. May make things easier, though Beta can be a bit scary. Still, I’m not in production with this project yet, so maybe taking the leap and trying Atmel Studio 6 is the way to go. I’m concerned, though, about the fact that version 6, like version 5, doesn’t support Linux. That makes it difficult for a large portion of the Open Hardware community to work on or adapt this project ☹.

# 2012/04/28

I’ve been communicating with various battery manufacturers in China and elsewhere via Alibaba. I’m working on getting quotes for Li-Ion and Li-Pol batteries. The specifications I’m aiming for:

Voltage: 6-7.2V

Capacity: 1500-2500mAh

Power Capability: around 300mA-500mA (or very roughly, about 0.2C)

Thickness: < 7mm (or roughly 0.27”)

So far the most promising quote has come from Shenzhen Puchuangyuan Technology Co. Ltd, who have answered all of my questions, and promise the ability to provide Li-Pol batteries thin enough to meet the needs of this project. They’ve also sent datasheets for each, though I have a couple questions I’m waiting for answers on. Samples will allow me to make real-world tests, rather than hoping that a similar batter will act the same in production, but the expedited shipping from China is expensive!

A couple other companies have also offered quotes, and I’m following up with them to find the best battery for my needs at a reasonable price.

# 2012/04/29

One interesting thing you’ll learn if you buy batteries from the manufacturer is that you can buy them with or without a charge management PCB. In the quotes I’ve received so far, the difference in cost with the PCB vice without it comes to about $1.50 or so. Looking at the charge management ICs on the market, in bulk, quality chips themselves would cost about $0.80 to $1.00. When you take into account the cost to manufacture the boards, but the additional components (only a few cents), and assemble the board, doing it yourself will cost around $2.50-$3.50 total per board.

Having gone through this pricing exercise, I think I’ll stick with the manufacturer’s board. If I trust the battery, I should be able to trust the board, and ideally, the manufacturer should know best which chip to use with their particular batteries.

I’ve received another battery quote with good details (from Zhejiang Nuociss New Energy Technology Co.,Ltd.). The cost per battery is about a dollar more than the quote from Shenzhen Puchuangyuan Technology Co. Ltd, but the cycle life is significantly higher, which I think should be well worth the cost. I’m building a simple spreadsheet as quotes come in, so that I can see more easily which offer is the best overall. I think it’s important to consider price, but to remember that price alone should not be of higher importance than overall quality.

I sometimes wonder if others develop things the same way as I do. I generally think up the project and begin to build it in my mind, using scratch paper to work out more complex problems and design the high-level schematics using black-boxes when I don’t know all the details yet). Once I’ve got all of the major components of the project figured out, I begin sourcing them. Then, once I’m confident there’s enough of each part on the market to manufacture effectively, I input the schematics in KiCad (or sometimes Cadsoft Eagle). From there, I begin to fill in the basic parts like resistors, determining the best part and values. And once that’s all don, it’s on to the PCB. I’ll back-edit as needed if there are changes to the design, but this can be exceedingly difficult if it’s a complex board, or ig there are specific routing requirements, like in this project. The traces to the capacitive sensors must follow very specific rules in order to obtain the best sensitivity and noise immunity. You can read more about these requirements in Atmel’s [QTAN0079: Buttons, Sliders and Wheels Touch Sensor Design Guide](http://www.atmel.com/products/touchsolutions/bsw/buttons_10.aspx?tab=documents), which covers all sorts of information about PCB and sensor layouts for Atmel’s QTouch and QMatrix technologies.