EE SENIOR DESIGN

High Level Design

Team AutoBev

Liz Clark, Lori Garcia, Alex Macomber, Mark Pomerenke

11/9/2010

Table of Contents

1 Introduction

In an environment where profit is dependent upon efficiency of product delivery, it is evident that a major source of missed revenue comes from lengthy wait times. A place where this problem is often seen is at overcrowded drinking establishments. Here, the drawn out process of ordering a drink, followed by waiting for completion of a transaction leads to much wasted time.

With the technology available in our current society, it is apparent that waiting times can and should be reduced. Much of the time associated with this process can be eliminated if the initial stage is automated. Through the utilization of a computer system that provides a means to order, pay for, and possibly pour a beverage, without dependence on a restaurant employee, the time between request and delivery of a final product can be greatly reduced and streamlined.

The AutoBev system will be beneficial to any establishment that wishes to decrease wait times, increase accuracy of orders, and furthermore, reduce workload on employees.

2 Problem Statement and Proposed Solution

The efficiency of most drinking establishments today is less than spectacular. Patrons can spend the majority of their night waiting to be noticed by a bartender. Once noticed, the transaction between patron and server can last for several crucial minutes that could be spent on serving other customers.

In order for a transaction to be completed, a patron must first be recognized as the next waiting customer by the bartender. The patron then explains his or her order to the server who then proceeds to make the specified drink. After the drink is served to the customer, the bartender tells him or her the price of the order and the customer in turn pays the bartender by the preferred method of payment. Regardless of payment choice, the payment transaction is very time consuming. Bartenders must enter the order into a touch-screen, swipe a card or open the register, print a receipt and have the patron sign the receipt.

The entire process of ordering a drink at a bar can be both time consuming and a frustrating experience for the server and customer. If a bar was able to serve drinks more frequently then it would increase its revenue. Another problem with the current system is that patrons tend to be served out of order which in turn decreases customer satisfaction.

The proposed solution for a more efficient drink ordering and payment system at a bar is to have an ordering and payment interface that allows for customers to place their own orders and serve themselves if their desired beverage is not something that needs to be made at the bar. In other words the machine would act as an automated bartender. The proposed machine would delay payments on tabs, completing the transaction upon closing of the bar. This will make it so that customers do not have to

leave their card at the bar, but instead will be able to swipe a magnetic stripe card with their information at the machine when an order is placed. This will be done via a card scanner integrated to interact with the automated bar tender. The automated bar tender will allow for customers to place their desired order. Upon placement of an order, the machine will allow the customer to pour their desired beverage into their cup if it is not a drink that needs to be prepared by the bartender. However, if the drink is something that must be prepared by a bartender, the order will be digitally sent to a display behind the bar where the bartender may view the list of drinks to be made.

3 System Description and Block Diagram

The AutoBev system will increase the efficiency of an establishment by effectively replacing the drink ordering process. AutoBev will have a user-friendly interface to select a beverage and allow the user to pour any selected beverage from a keg, or add the drink order to a queue to be prepared by a bartender. The system will personalize the ordering experience by storing user information, preferences, and history into a database which will be on display throughout the ordering experience. The system will also be accessed by the bartender to move through the drink queue, and indicate when drinks are ready. A system usage flow chart is included in Figure 3.1.

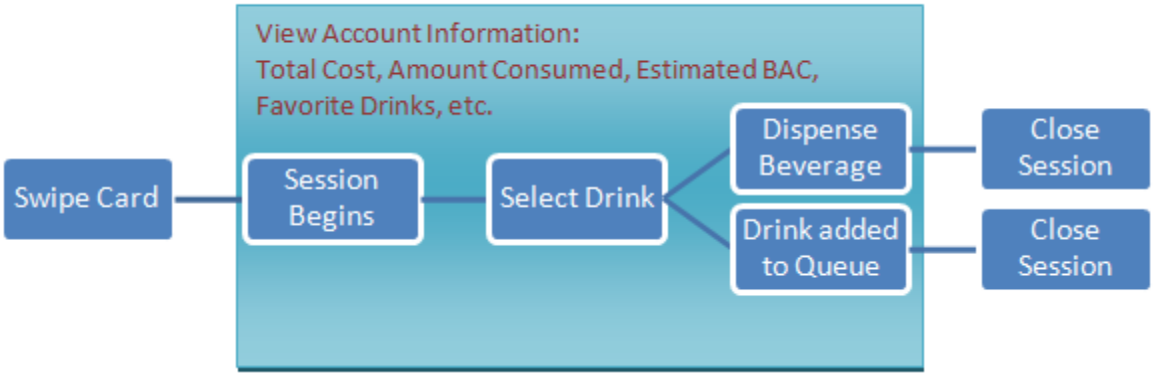

Figure3.1. System Usage Process

The AutoBev system will require the use of a magnetic stripe reader at the front-end to bring up an account. If the account exists, a personalized session will begin and the user will use a touch screen interface to select and order drinks. The interface will be running in parallel with the drink queue program on the same computer and will simply be separated by changing the display settings on the monitor. Touch screen operation is the equivalent to clicking a mouse, which will allow the bartender to have exclusive use of a mouse. On the whole the user will interface with a touch screen and a manual tap.

The computer will store and load user account information and will update particular values for each values. The database will hold static values of name and card

numbers, as well as dynamic values such as fluid consumed from kegs, most popular drinks, estimated BAC, bill balance, and a record of drinks ordered.

The computer will also communicate to a microcontroller which will control all functions associated with a keg. It will take temperature and flow measurements from the keg as well as enable/disable tap ability. The microcontroller will report back to the computer. The entire system can be viewed in Figure 3.2.

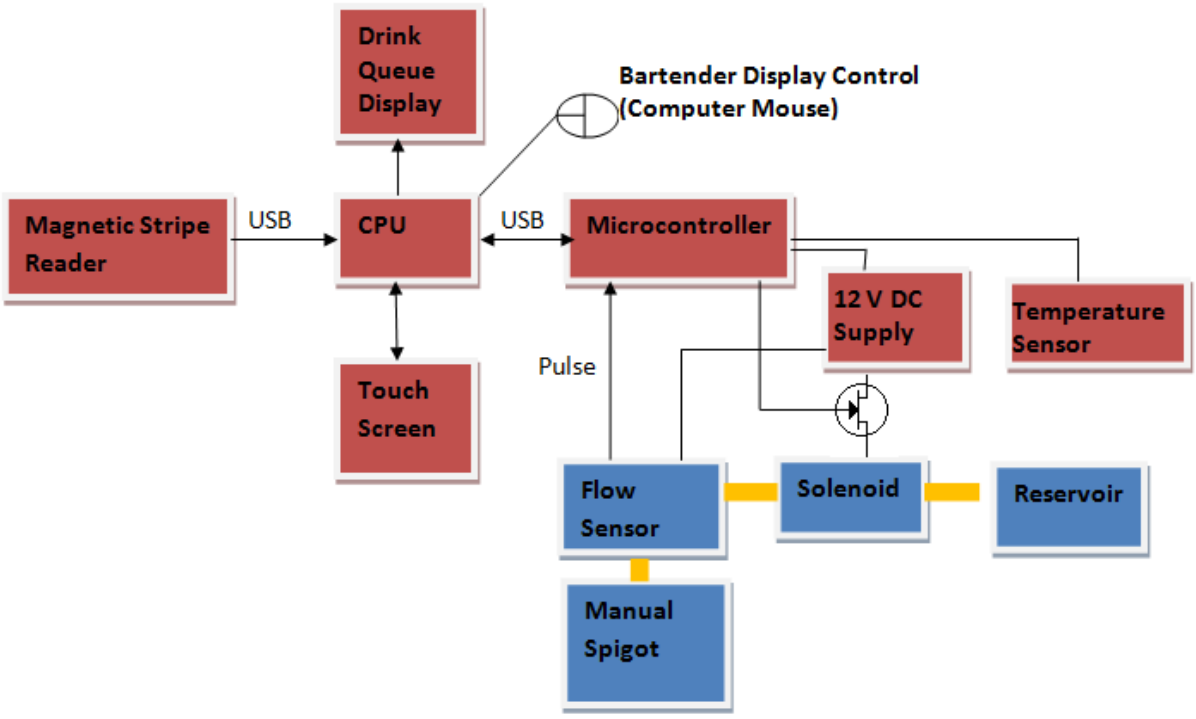

Figure 3.2. Block Diagram of Complete System

4 System Requirements

4.1 Overall System:

Overall System Requirements	
General	Must be capable of receiving and processing drink orders digitally Must use simple user commands for navigation (single click) Must have a layer of protection between CPU and other sensitive electronics and users/beverages
Size	Must fit onto a floor area no larger than 2'x2' (not including any keg)
Power	Must be powered by wall plug in 120V AC
Compatibility	Must be able to connect to a standard keg Must be expandable to include more drinks Must be in English Must be in compliance with current health standards

4.2 Subsystem and Interface Requirements:

Magnetic Card Reader	
General	Must be able to scan credit cards and send information to the PC
Size	Must not be bigger than 6"x6"
Power	Must be powered through USB interface
PC Software	Must be able to interface to a PC through USB Must be able to emulate a USB Human Interface Device (HID) keyboard Must turn on and off with PC
PC Computer Program/ Information Storage	
General	Must determine credit card scan has occurred, extract account number and verify validity
Implementation	Must run on a standard PC which is connected to a magnetic stripe reader
Connection	Will issue a signal to the micro-controller via a USB connection, indicating permission to pour

Database	Must calculate cost of purchase and properly update account balance of current user
User Interface	
General	Must allow for touch screen capability
Power	Screen must be powered by 120 V AC
PC Software	Must allow user to login and out of a session Must allow user to place multiple order Must allow user to navigate forwards and backwards through algorithm Will utilize tabbed browsing
Compatibility	Monitor must have DVI capability Software must be able to run on a windows PC
Drink Queue Display	
PC Software	Must run simultaneously with user interface
Compatibility	Must have same compatibility as user interface
Arduino Microcontroller	
General	Must determine when receiving pulse signal from flow sensor Must count and store number of pulses sent by flow sensor Will toggle solenoid valve
Power	Powered by 12 V supply from wall converter
Compatibility	Will receive signal from computer Will receive signal from temperature sensor
Beverage Dispensing and Sensing	
General	Must have keg to pour drink from with appropriate keg components (spigot, keg adapter, pressure valve, tubing, and gas tank). Must have a solenoid valve to only allow beverage to pour through if already paid for. Must have a flow sensor to keep track of how much beverage has been poured. Must have a temperature sensor to measure the temperature of the contents of the keg.
Power	Must be able to draw power from wall.

Size	All sizes must be consistent with size of keg and keg components.
Flow Sensing	Must be consistent with size of tubing for keg. Must be able to keep track of how much liquid is going through the tubing. Must be able to communicate the flow sensed to the microcontroller.
Temperature Sensing	Must be consistent with size of keg (should not be a problem). Must be able to measure the temperature of the contents of the keg. Must be able to communicate temperature sensed to microcontroller.
Valve Permission Solenoid	Must be originally closed and able to open upon request from microcontroller. Must be consistent with size of keg tubing/tap.
Microcontroller Software	Must use a reasonable amount of program memory Must be able to communicate with solenoid(open/close it). Must be able to get input from flow sensor periodically. Must be able to get input from temperature sensor periodically.

4.3 Future Enhancement Requirements

Future Enhancement Requirements	
Online Database	System should be capable of uploading the data onto an online database. This information will be accessed by the individual users who will have accounts registered with the website. Data will show drink purchase information such as time, quantity, type and expense.
Chard Credit Cards	Must be able to charge the credit card of the patron. This will be an upgrade from the current proposed version that simply stores the credit card information in a local database without charging the account.
Multiple Taps	Must be able to pour more than one type of beverages
Full Automation	Must be able to dispense the beverages without human interaction Must be able to place glass under tap and dispense.
Interfacing with Mobile Device	Must be able to use an iPhone application to complete beverage order and queue the order Send text message when drink is ready
Serve Mixed Drinks	Must be able to properly mix and serve “mixed drinks”

5 High Level Design Decisions

5.1 Card Scanner

The magnetic card reader is needed to identity the user of the system and to charge his or her credit card. The reader must be able to scan tracks 1 or 2 of ISO 7810 cards. The scanner must be a “plug and play” device that simply connects to a USB port of a PC and emulates keyboard inputs without any additional software. When a card is scanned the information of the card, such as account number and expiration date, is read into the USB port as if the keyboard were typing it in.

Many card readers meet the requirements for this design. The MagTek Magstripe Mini Swipe Reader costs $46.69 from provantage.com, part number 21040145. The MagTek Magstripe is 1.2”x3.9”x1.3”, reads tracks 1-3, has an LED indicator and can work as both an HID and keyboard emulator. The Magstripe can read cards in either direction and can read a card that is moving between 7.62 - 152.4 cm/sec.

5.2 Information Storage

A local database storage system will act to properly determine the occurrence of a valid inquiry to make a drink purchase, and to use customer inputs and selections to properly update user account information. Users will be preloaded into the AutoBev system by a master user.

The program associated with storage will work in conjunction with a graphic user interface and must:

- determine occurrence of credit card scan
- extract primary account number from input of magnetic card reader
- verify validity of account number
- send signal to microcontroller indicating permission to pour
- receive signal back from microcontroller indicating what was ordered
- update database with charge
- store statistics

In order to implement the account verification and revision software, a program will be implemented in the C++ programming language. The input to the program will be the output from the magnetic card reader subsystem (described in section 5.3). The input will be interpreted as a string of numbers.

The data that is read from the card reader will be taken from ‘track 2’ on the credit card. The first byte of data on the track is a one byte start sentinel. This byte will need to be discarded by the software. Bytes two through nineteen correspond to the personal identification number. Thus, these bytes will be stored. The program will then query an existing database to verify that the account exists. If the account is not found, no further action is taken by the program. However, if the account is successfully verified the, program will send a signal to the microcontroller giving permission to complete a drink

order.

The program will then activate an interactive graphical user interface on an LCD touch screen on which the customer will indicate if they will be ordering a drink from the bar, or be using the dispenser. If the user chooses a mixed drink, the program will sort through a list of mixed drinks and determine the cost of the drink. It will then relocate the user's account in the database and update the balance in the account. If the user chooses the option to use the dispenser, the program will wait to receive an additional input from the microcontroller based on the flow sensor, which will indicate how much of the beverage was dispensed. Using this input, the program will calculate the cost of the poured beverage based on volume dispensed. Following this, the program will again access the database, locate the correct account, and update the user's balance.

5.3 User Interface

The User interface will be a simple touch screen monitor connected to a learning center computer which will run a program that uses the entire screen and will have simple block clicking instructions for the user. Potential touch screen monitors are easy to find and the touch of the screen is the equivalent to a click of the mouse so there will be no added difficulty in programming the GUI. The touch screen monitor does not need to be large and could potentially be as small as 7", however, smaller screens are typically designed for specific applications and may be much harder to interface with. Although cost may be more, using a larger touch screen designed as a computer monitor will make this project much easier by simply allowing us to display on two screens just through changing computer display settings. Also larger monitors are more common, and for prototyping purposes we can find a cheap refurbished monitor.

Potential monitors range up to $300 and are:

- HP Compaq L2105tm - 21.5" (refurbished)
- 15" Planar Touch screen Multi-Media LCD Monitor No Base PT1503NT

The actual user interface will be programmed in C++ and will be a simple to navigate with the user clicking on blocks and tabs to move from broad options to further specific a drink selection. The user will enter a session upon login and will be able to move through the program front to back and back to front to specify drinks, and will click an "add to order" button when the drink is fully defined. This will show price, ask the user to confirm, and then allow the user to order more drinks. Should the user select a beverage on tap they will have free reign to use the tap and will be charged by volume, they will indicate on the touch screen when they are done pouring. The user session will end when a log out button is selected.

5.4 Drink Queue Display and Bartender Interface

The drink queue display will be attached to the same program and computer running the User interface. It will also respond to mouse click which will be accessed by the bartender. The Queue will simply be split onto a separate screen by changing computer display settings to split the display on two screens and dragging the queue onto one

screen and the user interface onto the other touch screen. The Bartender queue will display ordered drinks, and allow the bartender to simply delete entries once the drink is made.

5.5 Microcontroller

The microcontroller will be responsible for allowing a user to pour a beverage, for determining how much of the beverage was dispensed, and for communicating the volume dispensed to a computer application.

The microcontroller will receive a signal from an external computer application signifying a request for a beverage. This signal will be sent via a USB communication port. Once this request has been recognized by the microcontroller, it must then output a voltage which with initiate a relay, eventually charging and opening the solenoid valve.

Once the solenoid value has opened, liquid will begin to flow through the flow sensor and the flow sensor will start to send pulses to the microcontroller. This will be done by connecting the pulse output of the flow sensor to an input pin on the microcontroller. When this pin senses a signal, a routine will be entered in which the number of pulses are counted. The routine will increment a count value every time a pulse is sent. The microcontroller will determine that a pour is complete when it has not received a pulse from the flow sensor in an interval of time greater than one second. When this happens, the final count value will be sent back to the computer via the USB connection.

The microcontroller that will be used to achieve the above functionality will be the Arduino Duemilanove. This microcontroller has 14 input/output pins which will be attached to the output from the flow sensor and the input to a relay which will open the solenoid valve. It also gives the option of being powered via a USB connected to a computer, or through an external power supply (AC-to-DC adapter or battery). Additionally, this board enables serial communication via the USB from the Arduino to the computer, a functionality needed for determining the amount of volume dispensed. As a final benefit, the board is capable of I2C communication and supports an SPI interface.

5.6 Beverage Dispensing and Sensing

After a drink has been ordered, this feature will allow for drinks that do not need to be made by a bartender to be instantly available. Customers should be able to place their cup under a tap and manually pour the drink. The mechanical system to control flow will be created to connect to a keg of liquid. In order to keep a constant pressure in the keg, a gas tank with a constant pressure valve will be needed. One of the team members already has a kegerator and the rigging. He is allowing the team to use this on the condition that we purchase our own tubing. Thus, the needed spigot, keg adapter, pressure valve, and gas tank are already available. The solenoid valve will be attached to the tap and will be actuated by a digital output from the microcontroller. The microcontroller will signal the solenoid to open if a customer has placed their order and

is manually trying to pour their drink. A flow sensor will be needed in the flow path to send a signal to the microcontroller which will specify how much drink has been poured for pricing purposes. Additionally, we will have a temperature sensor to measure the temperature of the contents of the keg.

5.6.1 Flow Sensing

Finding an appropriate flow sensor was difficult. Many of the sensors found online cost too much and had components that were not necessary for our project. We ended up finding two types that were in a decent price range. These were the Vision 2000 and Swiss Flow meters. The Vision 2000 model number 46510-164-2F66 operates on 5V, 8 mA, and has a high sensitivity of 6900 pulses/liter. The Swiss Flow SF800 runs on 5 to 24 V, 12 to 24 MA, and has a high sensitivity of 6100 pulses/liter. The two meters were similar and we chose the SwissFlow SF800. This flow sensor sends the microcontroller an electrical pulse every time approximately 0.164mL passes through it. By counting the pulses we can calculate the volume and flow rate of the beer passing through the tube.

5.6.2 Temperature Sensing

The temperature sensor will be responsible for measuring the temperature of the contents of the keg. The temperature sensor that we will be using is the LM335A analog temperature sensor. The LM335A is a very easy-to-use analog temperature sensor. The LM335A works like a Zener diode with a breakdown voltage proportional to absolute temperature at 10mV/K. By hooking up a resistor from 5V and GND, the LM335A will output an analog voltage of 2.98V (298 Kelvin is 25C or room temperature). The output of the sensor is linear, and when calibrated at 25°C the LM335A has typically less than 1°C error over a 100°C temperature range. The sensor can operate continuously from −40°C to 100°C. Calibration of the sensor requires a potentiometer connected across the sensor, with the wiper of the potentiometer connected to the adjustment pin of the LM335.

5.6.3 Valve Permission Solenoid

The solenoids looked at were the ASCO subminiature solenoid valves series 8256. These solenoids could operate at supply voltages varying from 12V-240V DC or 120V-480V AC. We decided to use the 12V DC solenoid because 12V could be attained from a wall wart. 12V is also easier to switch with a simple n-MOS device. An AC powered solenoid would require a relay and would be more difficult to switch on with solid state devices, so we decided not to use one. Our solenoid is going to be switched on by a power n-MOS device that will have its gate tied to a general purpose I/O pin on the microcontroller. We chose the MOSFET IRF510 Transistor. The n-MOS device allows the low-current, low-voltage I/O pin to turn the 575mA, 12V solenoid on and off. The solenoid is normally closed and will open for an amount of time specified by a customer who is attempting to pour their drink.

Because the solenoid to be used runs on 12V DC a 12 V DC wall wart is going to be needed to have sufficient DC voltage for the solenoid. We found a 12 V DC – l A wall adapter power supply online that was good in price and would meet these requirements.

A power jack will be needed to connect the wall wart to the microcontroller board. The wall wart used has a 5.5x2.1mm center-positive barrel connector. Thus, we found a 2.1mm DC Barrel Power Jack PCB Mount to serve as the connection between the wall wart and board.

6 Open Questions

6.1 Programming Language for GUI and main program

The different programming languages that will be used have yet to be determined. The Arduino uses its own language similar to C++. The main program must be able to create a professional looking GUI while also handling a large database and communicating with the Arduino.

Possible languages that for the GUI will include Python with PyQT; C++ with QT; C++/C# (with Microsoft Visual Studio libraries); C++ with C, assembly and native Win32 API calls; JAVA with standard Swing GUI libraries. The language can either handle the database internally or communicate with a database management system (DBMS) such as MySQL, PHP, or XML.

The overall structure of the software that controls the system is currently undetermined. The main program must be able to communicate with the Arduino, provide a GUI, input numbers from the keyboard and store data to a database.

6.2 Microcontroller

A fairly simple microcontroller, such as an Arduino, would be a cheap usable solution for the AutoBev system. Is it required that we design our own microcontroller board?

6.3 Ticketing System

When a patron orders a drink his or her order is queued on the bartender's list of drinks to make. The method by which the patron claims his or her drink has yet to be determined. Possible techniques are printing a ticket with the type of drink ordered printed on it or assigning a number to each order and displaying the number currently being served (similar to a deli) on the wall.

7 Major Component Costs

Table 7.1 – Table of Major Components with Cost

Component	Cost
USB Universal Magnetic Stripe Credit/Debit Card Bidirectional Track-2 Swipe Reader	46.69
HP Compaq L2105tm - 21.5" (refurbished)	125.00
SwissFlow SF800	25.00
12 V Power Supply	15.00
3/8" Tubing (5 feet)	5.00
ASCO Subminiature Solenoid Valve	50.00
nmos transistor	1.99
Arduino Duemilanove	25.00
LM335A analog temperature sensor	1.50
Bourns 10K Pot	1.40
Total:	**296.58**

Provided Components	
Breadboard kit	provided
Learning Center PC	provided
Refrigerator and Resevoir Rigging	loaned for testing

The components listed without price are all accessible free of charge, and for prototyping purposes are not costs we need to worry. However, they will be considered in the future for total system costs.

8 Conclusions

We acknowledge that throughout the completion of this project, many obstacles will be encountered. The implementation of several software subsystems that all must communicate seamlessly with one another will require extensive research and a good deal of learning. In addition to the software challenge, the construction of the unit to dispense a beverage will require creative placement of each electrical component.

If our group succeeds, we expect that our project will help to increase the efficiency of drinking establishments through the integration of the ordering and payment process, into a computer program. It could also have applications at private parties as an accounting system to keep track of drinking information and allow access to authorized users.

References:

1. http://www.ascovalve.ca/catalog35pdf/8256_NSFR1.pdf (solenoid)
2. http://bestofferbuy.com/USB-Universal-Magnetic-Stripe-Credit/…
3. http://shop.wildcatcpu.com/Dell-ST2010-20-Monitor-Dell-ST2010.htm
4. http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove
5. https://www.eio.com/p-18850-national-semiconductor-lm335a-temperature-sensor.aspx
6. https://www.eio.com/p-900-bourns-rj26fw103-10k-potentiometer.aspx
7. "Magstripe Swipe Card Readers SureSwipe USB MSR Track-1/2/3 – Black". Provantage.com. Date Accessed: 11/7/2010
8. http://www.ascovalve.ca/catalog35pdf/8256_NSFR1.pdf (solenoid)
9. http://www.remag.ch/documents/bulletin_vision_2000_en.pdf (Remag Flow Sensor)
10. http://www.swissflow.com/en/SF800/applications/food_and_beverage (Swiss Flow Sensor)
11. http://www.radioshack.com/product/index.jsp?productId=2062618 (NMOS)
12. http://www.cutedigi.com/product_info.php?products_id=4184&osCsid=6aa2e283bbd8e15d895c844 b36ef7057 (Wall Wart)
13. http://www.cutedigi.com/product_info.php?products_id=4348 (Power Jack)