Lab 6h Introduction to PCB Design

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, Third edition, by Jonathan W. Valvano, published by Cengage, copyright © 2010.

Goals
- PCB Layout,
- Systems level approach to embedded system design,
  - Mechanical considerations,
  - Available of parts (second source),
  - Cost considerations,
  - Power considerations,
- Design for Test.

Review
- Data sheets for the MC9S12C32 microcontroller (review layout for 80 pin QFP),
  http://www.freescale.com/
  Click Documentation (under Get Support)
  Click Data Sheets
  Enter MC9S12C32MFUE25 into Search field, hit Go
  Select and download the datasheet (the file is called MC9S12C128V1.pdf)
- http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2010/PCBArtistLibraryTutorial.pdf,
- http://users.ece.utexas.edu/~valvano/Datasheets/TechArtsKitMOD912C32.pdf
- http://users.ece.utexas.edu/~valvano/Datasheets/AD9S12DP512R1.pdf
- Design Process in Section 1.3 of the book

Starter files **EE345L** library for PCB Artist, Lab6h_artistStarter.sch, Lab6h_artistStarter.pcb **Lab8BOM.xls**

Background

The objective of this lab is to perform a PCB layout for an embedded system. In particular, you are asked to create a PCB layout for one of the previous three labs: Lab 3f Alarm Clock, Lab 4h Stepper Motor Controller, or Lab 5h Music Player. In Labs 8, 9 and 11, you will design, build and test a real PCB, but in this lab, you will just create the design with a cardboard mockup without actually buying or soldering the PCB. Considering the design cycle presented in Section 1.3 of the book, one starts by analyzing a problem, and then one generates a **requirements document** describing what the system must do. The output of the analysis phase is a list of specifications and constraints. One can consider background section of the EE445L lab assignments as a requirements document. The Preparation and Procedure sections of lab assignments address the design and implementation phases of the project (e.g., data flow graphs and call graphs are design steps). The lab assignment and your solution to it, which includes the initial design, an implementation on a breadboard system, and hardware/software testing, represents one pass through the design cycle in Figure 6.1.

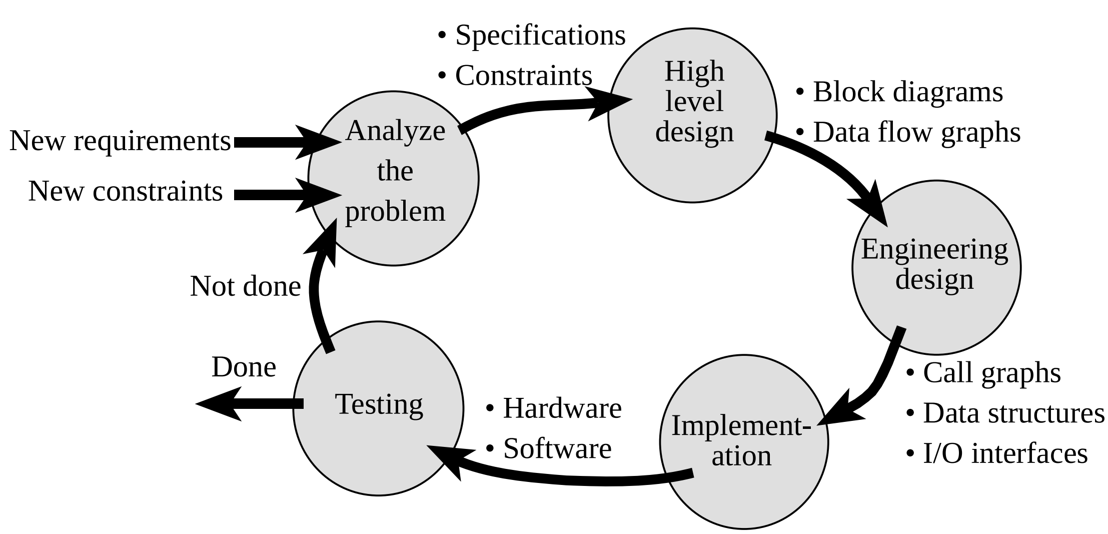

Figure 6.1. Product development cycle.

The following is one possible outline of a **Requirements Document**. IEEE publishes a number of templates that can be used to define a project (IEEE STD 830-1998). A requirements document states what the system will do. It does not state how the system will do it. The main purpose of a requirements document is to serve as an agreement between you and your clients describing what the system will do. This agreement can become a legally binding contract. Write the document so that it is easy to read and understand by others. It should be unambiguous, complete, verifiable, and modifiable.

1. Overview
- **1.1. Objectives:** Why are we doing this project? What is the purpose?
- **1.2. Process:** How will the project be developed?
- **1.3. Roles and Responsibilities:** Who will do what? Who are the clients?
- **1.4. Interactions with Existing Systems:** How will it fit in?
- **1.5. Terminology:** Define terms used in the document.
- **1.6. Security:** How will intellectual property be managed?

2. Function Description
- **2.1. Functionality:** What will the system do precisely?
- **2.2. Scope:** List the phases and what will be delivered in each phase.
- **2.3. Prototypes:** How will intermediate progress be demonstrated?
- **2.4. Performance:** Define the measures and describe how they will be determined.
- **2.5. Usability:** Describe the interfaces. Be quantitative if possible.
- **2.6. Safety:** Explain any safety requirements and how they will be measured.

3. Deliverables
- **3.1. Reports:** How will the system be described?
- **3.2. Audits:** How will the clients evaluate progress?
- **3.3. Outcomes:** What are the deliverables? How do we know when the system is done?

As you know, commercial products are not manufactured using solder-less breadboards like the ones we use in EE319K and EE445L. Implementing the embedded system on a PCB will improve maintainability, testability and reliability. It will also reduce the size, weight and cost of the system. In this lab, we will also consider power. The following requirements are added to the design

The system will be controlled by the 80-pin QFP MC9S12C32MFUE25
All electronic components must be soldered to a single PCB board.
You must hand place and hand route the PCB (you can not use *Autoplace* or *Autoroute* features)
The system must fit neatly into a box (your choice of the box)
The system will be battery powered, run for 24 hours, and the battery must also be in the box.

There are no specific desires in this lab to minimize size, weight, cost or power. On the other hand, you will be asked to estimate these parameters in your system. There are three considerations when designing a power efficient system. First, we need to make a **Power Budget**. This means we determine the maximum amount of power that can be supported by the power source. Second, we need to identify and analyze required system tasks. This means we determine the tasks that need to be performed by the system and estimate the power required to perform these tasks. It is important to consider timing of the tasks and synchronization to I/O events. For example, executing about once a second (e.g., smoke detector checking for fires) will require less power than executing exactly once a second (digital watch). One way the TI MSP430 family achieves very low power is it can run with no crystal oscillator, and the rate at which the microcontroller runs can only be approximately set. For example, the 40-pin MSP430F2274 running at 1 MHz requires only 400 μA of supply current, whereas an equivalent 9S12 requires about 35 mA. The third consideration is to develop a **power strategy**. Having a strategy for power helps the power needed to perform the required system tasks to fit within the power budget. A **Power Budget** is a first order calculation that gives you a ballpark figure of the "total average current" supported by your power source. From the system specifications we are given how long the system must operate without replacing batteries, $\mathbf{t_{life}}$ in hours ($\mathbf{t_{life}}$ is 24 hours in this lab). From the battery datasheet we determine the storage capacity of the battery, **E** in mA-hours. Be aware, however, that for many batteries the storage capacity depends also on current, time and temperature. The power budget is

Average Current must be less than $\mathbf{E/t_{life}}$

The system has functionality to handle the clock, reset, power, RS232 serial, and BDM interface. In particular, it has a 6-pin BDM header (BDMconnector[1]), an on/off switch (BOXMOUNTEDBUTTON[1] PowerJumper[1] or Jumper2[1]), a crystal (XTAL[1]), a voltage regulator (e.g., LM2931T-5 or LM7805CT[1]), and a reset switch. The high/low value of PE7 at the time of reset affects which clock circuit the 9S12 uses (making it low is the correct approach). PE1 and PE0 can cause interrupts (making them high will prevent unwanted interrupts). There are two links in the **Review** section above that show complete 9S12 circuit diagrams. We expect most EE445L students to power their system with 5V, but you are free to design the system at any voltage that works.

Preparation (do this before your lab period)

1. Choose for which of the 3 previous labs you want to design your PCB. Decide which microcontroller you want to use. You are free to use any microcontroller you want, however we expect most EE445L students will choose the MC9S12C32MFUE25 in an 80-pin QFP package. If you want to choose a different microcontroller than this, please get approval from your TA. You will find four 9S12 chips in the EE345L library for PCB Artist. The specific part number most students will use is the MC9S12C32MFUE25[1]. As an option for Labs 8, 9 and 11, we will have some MC9S12C32MFUE25 parts to distribute to students. You will begin by downloading the Lab6h_artistStarter.sch and Lab6h_artistStarter.pcb files, changing the file names, leaving the **Lab6**, but adding your names, e.g., Lab6_Valvano_Bard. The SCH and PCB files are connected, so they need the same name, but different extensions. Next, you make a copy of the SCH file for your lab solution. Then, you delete the existing 9S12DP512 component (H1_DP512[1]) and all the nets that connect the 9S12 to your circuit (essentially editing the copy of your lab solution SCH). Next, you copy the circuit from your lab solution into the renamed Lab6 starter SCH file. Pay careful attention that all the +5V connections are one net, and all the ground connections are one net. You must reconnect the I/O pins to reestablish the functionality of the lab. You may add other hardware beyond what was needed for the lab if you choose, but it is not necessary to add more I/O functionality.

2. As part of Labs 3, 4, and 5, you measured the supply current required to run your system. Choose a battery that will power your system for at least 24 hours. For the stepper motor system, you can run your system for 4 hours instead of 24 if you want. For example, if your system draws 50 mA, then a 1200 mAh battery will operate 24 hours between charges. Go to a battery manufacturer (like http://www.tenergybattery.com/) and select a Li-ion or Li-poly battery pack to run your system. You will need to know its dimensions, weight, cost, and energy storage capacity. To get the price, you can search a retailer (like http://www.batteryspace.com http://www.batteryjunction.com) or a search engine like Google. It is not necessary, but just for fun, look at Figure 1 of the data sheet for the MAX1873; it shows a charging circuit that would allow you to charge your Li-ion battery. If you were to place this circuit on your PCB, then you could simply plug your system into a charging station without needing to disconnect the battery from the box. Lab 6 does not require an on-board charging circuit.

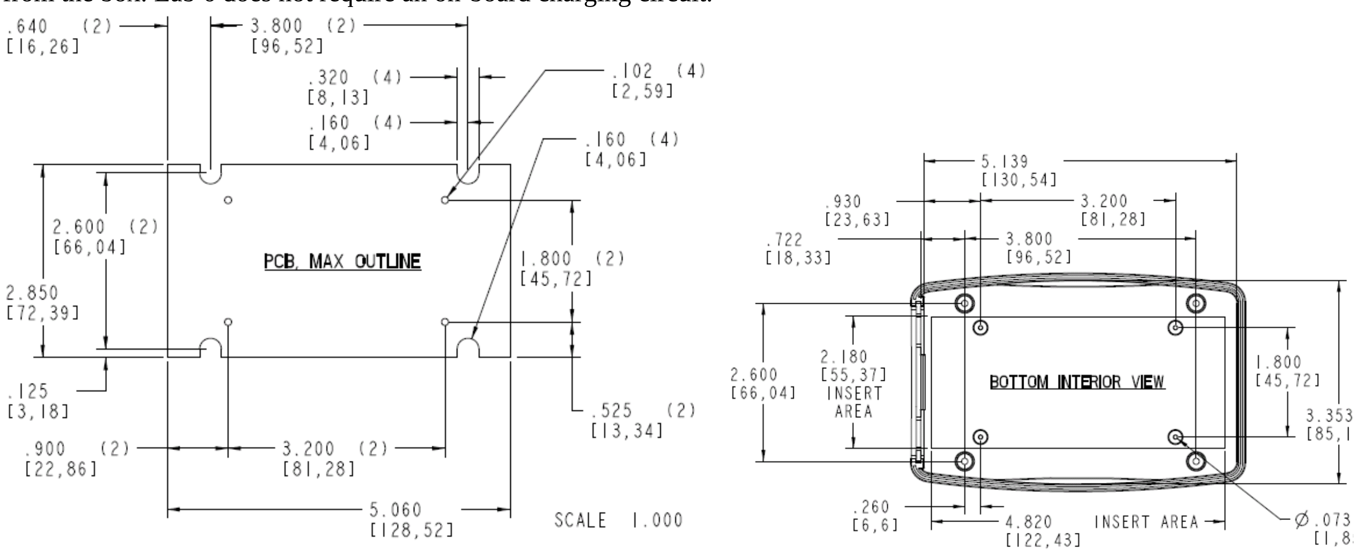

Figure 6.2. Mechanical drawings for the Pactec XP enclosure.

[1] Name of the component as defined in the EE345L library

3. Decide on the PCB size you need to hold all the components. Choose a box in which to place the system. Go to a box manufacturer (like http://www.pactecenclosures.com/ or http://www.serpac.com/) and select a box for your system. There are lots of Pactec and Serpac boxes in Valvano's office if you want to see what they look like. You will need to know the box dimensions, weight, and cost. The starter files are configured for a 5in by 5in PCB, to fit into a cheap $1 surplus box. You will want to choose a smaller, prettier box. There is no specific requirement in this lab to minimize cost or size. A bigger box will handle a bigger PCB, making layout simpler. To get the price, you can search a parts distributor (like www.jameco.com www.mouse.com www.newark.com or http://www.digikey.com/) or a parts search engine like http://octopart.com/. In particular, you will need layout dimensions for placing a PCB board into the box, like Figure 6.2. Remember to leave room for the battery. All systems have operator buttons, which can be placed either on the box with cables onto the PCB (for $2 each), or soldered directly on the PCB (for no additional cost), but positioned so the buttons protrude outside the box. If you are building Lab 3 alarm clock, then the LCD needs to protrude out the top of the box and be plugged into a 14-pin header (HD44780[1]) on the PCB without needing a cable (using the Samtec BCS-114-L-S-TE header for which you should have a free sample). You can place a small 32-Ω speaker in the box. If you are building the Lab 4 stepper controller, then the 7-pin header to the motor is added to the PCB and the motor can be placed outside the box (StepperMotor[1]). If you are building the Lab 5 music player, then a stereo jack (HeadphoneJack[1]) is soldered directly on the PCB, but positioned so the jack protrudes outside the box. You were given a headphone jack in EE319K.

Procedure (do this during your lab period)

1. If you were asked to make 1000 copies of the system, think about how you would test each system during manufacturing. The BDM pod allows you to download and run software on the 9S12, but add test points to allow electrical testing of each system built. The RS232 port will allow you to run the serial monitor debugger. In particular, specify a test procedure for your system. Be as specific as possible, but limit this test procedure to one page. For high-volume production, the test procedure is automated and performed robotically, but for this part you can have steps like "connect a scope to test point V1" or "connect a 16-channel logic analyzer to 14-pin header J3". You do not have to write actual software, but you should describe what the software does. E.g., "Run 9S12 software that generates a 1 kHz sine wave on the DAC".

2. Make a list of components required for the system. To get started, you can create a **Bill of Materials** for the PCB itself by executing *Output-> Component List*. Copy/paste this list into the **Lab8BOM.xls** starter file and add other components like the battery, box, LCD, speaker, motor etc. Add a column called estimated current. Collect pdf files of the data sheets for all components except resistors, capacitors, connectors, stepper motor, speaker, and crystal (do not print the data sheets, just collect the pdf files). For each component, specify quantity, package type, cost and estimated supply current in the spreadsheet. To get the price, you can search a parts distributor (like www.jameco.com www.mouse.com www.newark.com or http://www.digikey.com/) or a parts search engine like http://octopart.com/. It doesn't matter whether you use quantity 1 pricing or quantity 1000 prices, as long as you document what you did. Make a note of any device that has a single source (for example, Freescale is the only manufacturer to produce the 9S12). Compare the total estimated current of the individual components with the actual measured current from lab. For every cable/connector your system needs, add $2 to the cost. Estimate the total cost of the system. Calculate the total weight of the device as the battery, LCD (26.1g), box, plus the weight of your TechArts 9S12DP512 board (35.5g). For resistors, specify wattage (1/4 or 1/6 watt) and tolerance (5% carbon or 1% metal film). For capacitors, specify type as 5% C0G ceramic, 10% X7R ceramic, 20% Z5U ceramic, or 10% tantalum. For more information on resistors and capacitors, see

http://users.ece.utexas.edu/~valvano/Datasheets/CarbonFilmResistors.pdf
http://users.ece.utexas.edu/~valvano/Datasheets/CarbonFilmresistors2.pdf
http://users.ece.utexas.edu/~valvano/Datasheets/CapacitorC0G.pdf
and http://users.ece.utexas.edu/~valvano/Datasheets/TantalumCap.pdf

3. There needs to be at least one component you use that you have created from scratch using the Library editor within PCB Artist. Your TA has some actual chips for which you will build the component. When you are done, give the chip back to the TA, so we can use it again next semester. Please make a printout of the component in the Schematic Symbols editor, PCB Symbols editor and the Components editor for your report. You should create a new library into which to place your new part(s), so the information is not lost if EE345L or PCBArtist updates their libraries. Your does not need to utilize the part, but you will have to demonstrate your footprint is accurate.

4. Within the SCH editor, execute *Output->Unconnected Pins Report*. Many of the 9S12 pins will be not connected, but go through this list to spot any forgotten connections.

5. A net describes the interconnection pattern on the PCB. You need to label all your nets with descriptive names. I.e., there shall be no net names that begin with **N0**. Within the SCH editor, select the *GoTo* tab on the *Interaction Bar* and view the **Net** list. For each net, give it a descriptive name, and for each important net activate *Display Net Name*. Then one by one click on the Net name in the *GoTo* window on the *Interaction Bar* and verify the proper connections as highlighted in the schematic window. A common mistake people make is to inadvertently merge two nets, or split one net into two separate nets. Another common mistake people make is to place two lines close to each other so they look like they are connected on the computer screen, but they are not.

6. Have your SCH file checked by your TA or instructor before you start the PCB layout. There is no particular need to minimize size, cost or weight. So, except for the 9S12, it is ok to use the larger footprint through-hole parts. Within PCBArtist, execute *Tools->SchematicPCB* and select *ForwardDesignChanges*. Place the fixed objects first (e.g., drill holes for mounting in the box, headphone jack, LCD, and switches soldered to the PCB.) The next step is to place all remaining components inside the PCB area. Give careful thought when placing component to minimize trace lengths. Put parts next to each other that connect to each other. If possible, position polarized parts (i.e. diodes, and electrolytic caps) with the positive leads all having the same orientation. Many of the components use a square pad to mark the positive leads of these components. Doing a good job here will make laying the traces much easier. You can not use *AutoPlace* or *AutoRoute* for this lab. You will save a lot of time by leaving generous space between ICs for traces. Frequently the beginner runs out of room when routing traces. Leave 0.350" to 0.500" between ICs, for large ICs allow even more. Once all components are placed, print out the PCB top layers. Using a ruler make sure the large components (LCD, switches, and connectors) will fit in the box.

7. Once you are sure everything fits, you will route the nets. Execute *Tools->SchematicPCB* and select *Consistency Check*. Execute update components if necessary. I like to route power and ground first. Avoid loops in the signals, because loops can pickup EM field noise. As you get more experience (after your 4th PCB design) you can use ground planes. However in this class, place ground and power paths in the shape of a capital E. Most of the signals in this class can be 7 to 10 mils wide. For the stepper motor, use 20 to 30 mil wide signals for power, ground and signals carrying motor current. Whenever possible increase the power and ground paths as will fit. For this introductory class I recommend that you do NOT cover the unused area with a ground plane or a power plane, as shown in Figure 6.3. Although ground planes are great for low noise or high frequency circuits, they do make it very hard to see what is connected to what. Later on in Lab 11, ground planes will also complicate the soldering process because of the thermal load of the plane (it will be hard to get it hot enough to melt solder). Here are some guidelines.

Make sure the Snap to *Grid* mode is active (experiment with different settings of the snap)
Add Top Silk labels for your initials, your TA's initials, the date, and the purpose of the board,
Place all though-hole components on the top side (surface mount components can go on either side),
If possible align all chips in the same direction,
Configure the board so that all though-hole soldering occurs on the bottom side,
Add Top Silk labeling to assist in construction and debugging,
Add test points at strategic points to assist in debugging,
 Either by placing two holes 0.1 in apart then soldering a U wire into it,
 or by making a 0.090 in pad with 0.043 in hole then solder a test point into the one hole
Each IC should have a bypass capacitor, placed as close to the chip as possible,
All components need labels (e.g., U1 R1 C1 J1 etc.), shown both on the board and the circuit diagram,
Avoid 90-degree turns, convert them to two 45 degree turns,
One way to make it all fit is to go left-right on one side and up-down on the other side

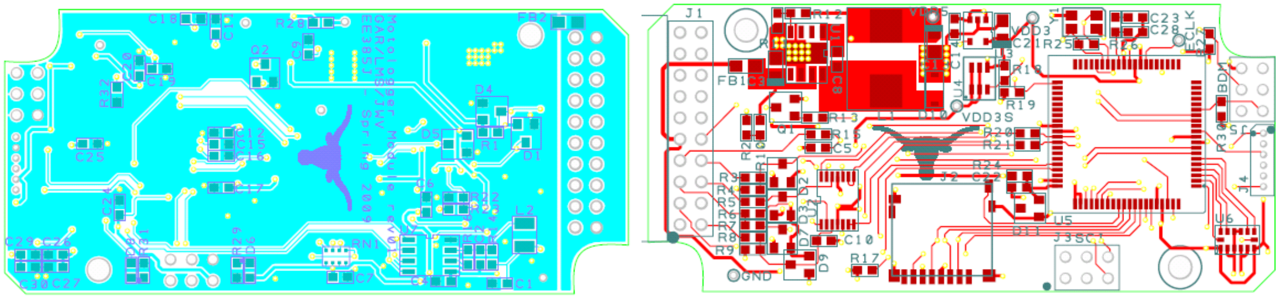

Figure 6.3. Bottom and top PCB layout for a 9S12 system.

Traces that carry significant current should be wider than signal traces. In order to determine trace width, one needs to know the maximum current, the length of the trace, the thickness (1 oz/ft^2), ambient temperature, and the maximum allowable temperature rise in the trace. For traces much longer than they are wide, and currents less than 35 amps, you can use the following calculators based on the IPC-2221 standard (formerly IPC-D-275). The second link can be used to calculate sizes of vias.

http://www.circuitcalculator.com/wordpress/2006/01/31/pcb-trace-width-calculator/
http://circuitcalculator.com/wordpress/2006/03/12/pcb-via-calculator/

Table 6.1 gives rough guidelines of how wide to make a trace for a given amount of current that trace will carry.

Length	**Temperature Rise**	**Current**	**Thickness**	**Resistance**	**Trace Width**
5"	1 C	100 mA	1 oz/ft^2	1 Ω	2 mil
5"	1 C	200 mA	1 oz/ft^2	0.47 Ω	5 mil
5"	1 C	500 mA	1 oz/ft^2	0.13 Ω	20 mil
5"	1 C	1 A	1 oz/ft^2	0.05 Ω	50 mil
5"	1 C	2 A	1 oz/ft^2	0.02 Ω	120 mil

Table 6.1. Minimum trace width for various current levels

8. When you think you are done, execute from the *Output* menu *Design Status Report* and a *Net Completion Report*. Fix any errors. Execute a *Tools->DesignRuleCheck*. Lastly, you should be able to go through the process of ordering the board described at
http://users.ece.utexas.edu/~valvano/EE345L/Labs/Fall2010/PCBOrderProcess.pdf
up to the point where Advanced Circuits asks for a credit card. Please do not give them a credit card or order any boards. Make five printouts

A PCB showing the part you created in the library (put your actual chip on the paper to see if it fits)
Just top copper to be judged for layout style
Just bottom copper to be judged for layout style
Top copper and top silk, glue to top of cardboard
Bottom copper and bottom silk, glue to bottom of cardboard (print mirrored).

Cut out a piece of cardboard or wood about the thickness of a PCB board and glue the last two printouts to it.

9. Make two mechanical drawings of your system. Please include both top and side views. Your drawings can be hand-drawn. They must be drawn to scale, and please include the scale. For this type of system a 1-1 scale makes sense, but you can have other scales if you want. Explain the physical configuration of your major components, including the box, switches, jacks, speaker, connectors, PCB, and battery (as appropriate for your system).

Deliverables (exact components of the lab report)

A) Objectives (1/2 page maximum)
B) Hardware Design
One page description of the battery (printout from the web) (Preparation 2)
One page description of the box (like Figure 6.2, Preparation 3)

Three pages showing the new component you created (Procedure 3), and an example PCB using it
Two mechanical drawings (Procedure 9)
Final circuit diagram of the embedded system, SCH file
Top copper printout of the PCB layout
Bottom copper printout of the PCB layout
Cardboard mockup of the PCB layout

C) Software Design none

D) Measurement Data

Bill of Materials (quantity, package type, cost, and supply current) (Procedure 2)
Explain how you chose the battery (Preparation 2)

E) Analysis and Discussion (1/2 page maximum)

Explain the testing procedure you would suggest for the system (Procedure 1)
Explain any differences between estimated current (Procedure 2) and actual measured current measured when doing the lab.

Checkout (show this to the TA)

You execute the commands listed in Procedures 4, 5, 6, 7, and 8 to verify project completion. You will be given a part number and asked to search for its data sheet and find the cheapest place to buy one. You should be prepared to discuss alternative approaches and be able to justify your solution.

Create a zip file and give it to your TA, to be graded for style at a later time, include

Your SCH and PCB files
Downloaded pdf files of data sheets and/or specifications, created by the distributer or manufacturer
Battery
Box
All components attached your PCB except resistors, capacitors, connectors, and crystal

A good reference for PCB design is http://focus.ti.com/lit/ml/slup230/slup230.pdf