Design Competition 2014 Workshop 6: Design and Mill a Printed Circuit Board

Task:

1. Design **a printed circuit board** in EAGLE, containing a PIC32MX150F128B, DMV885 hbridge, Nokia 5110 LCD, and whatever else your robot might need.
2. Use the **Roland MDX-15** to mill the PCB. Solder the components to the PCB.
3. Download and install MPLABX and XC32 on your computer, make a new project for the PC32MX150F128B and compile the sample code. Use a PICKIT3 to program your PIC32.

Download: (see email attachments for now)

1. DC14Workshop4.zip, containing **DC2014_PIC32MX150board.sch, DC2014PIC32MX150board.brd**, DC14LineSensor.sch, DC14LineSensor.brd, DC14PIC32MX150_main.c, PIC32MX150_pins.c, and PIC32MX150_pins.h.

Milling a PCB

- A **Printed Circuit Board** is a substrate (fiberglass or pressboard) with copper cladding (one side, two sides, or more) where the copper is **routed** into the conductive **traces** that match a **circuit schematic**.
- Components can be soldered into holes in the board (the same components you would use in a breadboard) or soldered to the surface of the board with no holes (surface mount components, which do not have long leads). Surface mounting saves space, but can be more difficult to solder and route. Many modern components are not manufactured in DIP (breadboardable) style anymore, and are only available in SMD (surface mount) packages.
- We use EAGLE to design circuit schematics and PCBs (free but limited size in v6, any size with our v5 license). Altium is a professional grade software package, and KiCAD is a nice freeware package. Some manufactures, Advanced Circuits and ExpressPCB, have good free software, but require you to purchase the board from them.
- Obviously the ‘P’ in PCB comes from the manufacturing process a real board house uses to make a board. Professional boards are great, but cost at least $20 with shipping, and can take 3 weeks to be delivered. We will use a CNC mill to mill our boards, which takes around 30 minutes, but at the moment the boards are only one layer, the races have to be further apart, which makes the design a little trickier.

The DC14 template

- Open DC2014_PIC32MX150board.sch and .brd.
- In the .sch, note:
    1. We are using the SOIC package of the PIC32MX150F128B – surface mount, but pretty easy to solder.
    2. There is no USB connection, so there is no bootloader on the chip. You will need a PICKIT3 to program the board.
    3. Because you cannot send data back to the computer, there is a Nokia 5110 LCD to display data (https://www.sparkfun.com/products/10168)
    4. Power is supplied by a 1 cell, 2200mAHr rechargeable LiPo battery (http://www.all-battery.com/Li-Ion186503.7VRechargeableBatterymodulewithPCBandBareLeads-30027.aspx)
    5. Two motors can be controlled with a DRV8835 h-bridge (http://www.pololu.com/product/2135)
    6. The rest of the pins of the PIC32 are broken out so you can add more circuitry.
    7. The PIC32MX150F128B has reassignable pins (in code you can change which pin can do PWM, etc). Some functionalities cannot be moved (analog input, some communication pins). Take a look at the sample code (especially PIC32MX150pins.c) to see which pin can do what.
        - This PIC has 5 PWM capable pins (OC1-5). That means you can have 5 motors with controllable speeds (2 drive motors and 3 servos, 4 drive motors and 1 servo, 5 servos and not speed controlled dive motors). Of course you can control drive motors without PWM, but servos require PWM (unless you want to write some sophisticated pin-toggling code).
- In the .brd, note:

1. The board is 3” x 2”, single-sided.
2. Some parts are thru-hole, some are SMD. Resistors, capacitors and LEDs are size 1206.
3. Because the board is single sided, it can be tricky to route some traces, like power and ground. Using jumpers (wires that you will add later) or 0Ω resistors to cross wires is recommended (within reason, you don’t want everything to have a jumper).
4. Keep your traces at least 0.012” wide, and 0.012” apart, for manufacturability. The mill uses a ~0.25mm bit to cut out your traces, and a 1/32” endmill to make holes.
5. Surface mount parts go on the top surface, thru-hole parts go on the bottom (for solderability). Keep that in mind when you design where wires and switches and things will go in relation to your robot.

Using the Roland MDX-15

- The Roland mill uses .rml code as instructions to cut material. There is not direct conversion from .brd to .rml.
  1. Hit the ‘layers’ button in EAGLE, and click ‘none’ and ‘apply’. Select the top layer, pads lay, and vias layer, and click ‘apply’. You should see only red and green on the screen. This is the top copper that will remain after milling. Make sure the traces are not crossing / are too close / other errors.
  2. Go to File->Export and select .png. Check the ‘monochrome’ checkbox. Set the DPI to 1500. Save the file.
  3. If your board needs to have holes drilled, go into layers and turn off top, and export again. Open the holes .png file, and fill in the black areas with white, leaving only black holes.
  4. Put the .png files onto a flash drive and copy them over to the laptop attached to the mill. The laptop name is ‘millme’ and the password is ‘millme’. The operating system is Ubuntu, and the code is written in Python.
  5. From the command line, type ‘sudo fab’, and enter ‘millme’ as the password as prompted (if you do not use sudo, the program will not be able to access the USB port to talk to the mill).
  6. Select the input file as .png and the output as .rml.
  7. Select your .png file, and make sure the dimensions look reasonable.
  8. Select ‘mill traces (1/64)’ from the top (ignore the fact that is says 1/64). Enter your bit size, commonly the 1/64” endmill (0.39mm) or 0.25mm (for the V-bit).
  9. Hit the ‘make path’ button, and make sure every trace is surrounded by a mill path. If not, you can undersize the mill diameter, but note that that will cut everything small.
  10. At this point, put some double stick tape on the back of a blank PCB. Stick it down to the acrylic plate. Turn on the mill, and hit the view button. This will bring the mill to the (0,0) location.
  11. In the software, use the move button to move the head to the bottom left corner of the board. Use the down button to bring the tip just above the board. Use the hex key to loosen the bit, and gently place it on top of the board, then retighten the bit. This process zeros the mill.
  12. In the software press ‘send file’, and in the next window click ‘begin milling’.
  13. When the process is done, use the up button to clear the bit from the board. Open the .png file with the holes, and make the path using the ‘mill board (1/32)’ option. Change the bit to the 1/32” endmill, and rezero the board using the same method as before, with the same (x,y) offset you used previously.
  14. Set the drill depth to 1.2mm, and mill the board.
  15. When done, use the up button to clear the bit from the board, and remove the board. You may need to lightly sand it with 200 grit paper, and make sure there are no tiny copper burs shorting the traces.

Soldering and testing

- If the board is a new design, solder a few parts, then test, then add more parts and test. Don’t solder the entire thing at once, because if the board doesn’t work you won’t know where it is failing.
- Solder the voltage regulator and power things first, and check to make sure it works, and that there are no shorts. Then do the PIC and sensors and actuators.
- Soldering thru-hole parts is easy. Stick the part through the bottom (non-copper) side, bend the legs a little, and solder them to the traces. Soldering from the top can be difficult, depending on how much the part covers the traces.

- Soldering surface mount parts is a little harder because you need to align the part while soldering, so you need three hands. You can cheat and pre-apply solder to one pad, then solder that pin first, and it will hold the chip in place while you do the others.
- Soldering anything to ground is harder than any other pin because the ground acts as a giant heat sink. You will have to hold the iron on the ground longer than a normal pin to solder it.

Programming the PIC32MX150F128B

- Download and install MPLABX IDE and XC32 from https://www.microchip.com/pagehandler/en-us/family/mplabx/
- XC32 is the compiler, the program that takes you code and makes it usable by the PIC. The IDE is a nice text editor and project organizer.
- Make a new project in the IDE for the PIC32MX150F128B, and select the PICKIT as the programmer, and the version of XC32 you downloaded as the compiler. This will make a new folder in the MPLABX folder. Put the .c and .h files in the new folder. In the IDE, right click on the headers folder and add the .h files. Right click on the source folder and add the .c files. Now you can edit the code, and compile.
- Use the button with the green arrow to program the board with the PICKIT3. In project settings, you can set the PICKIT3 to power the board while it is plugged in. Or you can use the battery, but don't do both.