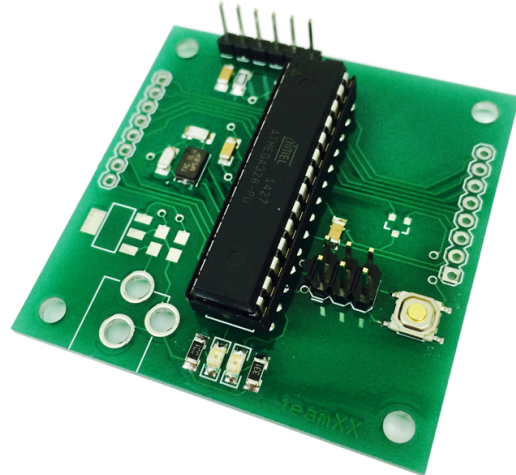


## 18-549 Embedded System Design: Lab 2 Assembly and Bring up

**Assigned: 02/10**

**Due: 02/20 AoE**

The purpose of this lab is to take the PCB that you design in Lab 1 and assemble and test the core components. It is important that you strategically bring up boards such that if a core component fails you can easily isolate and debug potential problems.



### Hand-in Procedure:

- This Lab is to be done by **your group**. There is no paper hand-in, you simply need to get a member of the course staff to sign-off on your board. Demonstration of a complete board will include showing the output of a simple “hello world” program on a serial terminal.

### Grading:

- Your submissions will be graded on the following criteria:

Criteria	Grade	Score
Team board assembled, boot loader works, LEDs work, terminal output visible	A	10
TA reference design assembled, boot loader works, LEDs work and terminal output is visible (assuming team board fails)	B	8
Any board assembled but not all components work	D	5
None of the above	F	0

## Assess your Kit

Make sure you have the following in your parts kit:

Part	Qty
PCB	$\geq 1$
6x1 Header	1
3x2 Header	1
DC Power Jack	1
ATmega328p	1
28-pin Chip Holder	1
Button	1
5V Voltage Regulator (5209) <i>(Ask TA if you need 3.3V or other)</i>	1
16 MHz crystal (TXC 16.0) <i>(Ask TA if you operate at 3.3V or need 8 MHz)</i>	1
12 pF Capacitor (Almost white) <i>(Ask TA if you operate at 3.3V or need 8 MHz)</i>	2
0.1 $\mu$ F Capacitor (Light Brown)	4
22 $\mu$ F Capacitor (Dark Brown-Red)	1
330 Resistor (331)	1
10K Resistor (1002)	2
Green LED (red dot in the center, green marks on cathode side)	1
Red LED (black dot in the center, green marks on cathode side)	1



16.0 MHz  
Crystal



12pF  
Cap



0.1 $\mu$ F  
Cap



22 $\mu$ F  
Cap



330  
Res



10K  
Res



Green  
LED



Red  
LED

## Assemble Your Board

*Step 1.* Solder the holder for the microcontroller and insert the ATmega328p. Then solder the 3x2 ISP header, the 6x1 programming header, and the 10K reset pull-up resistor, and the 6-pin programming header.

Make sure to check your board layout file before soldering the holder and plugging in the ATmega328p. Ensure that the tab marking on the chip matches the one in the layout, so that you know that you have the correct orientation. When soldering any multi-pin component, start with just one pin and adjust the placement if needed, then solder a pin on the opposite side for stability, then do the rest of the pins.

*Step 2.* If you have an AVR ISP programmer, use it to check your board before proceeding (it should give you double-green lights). If not, get one of the TAs to check your board before proceeding.

*Step 3.* Solder the 16.0MHz crystal oscillator and the two 12pF capacitors (use these values though you had used different values in your schematic). Also, solder the two 0.1uF bypass capacitors. The orientation of the crystal and the capacitors does not matter in this case. If you want to operate at 3.3V or lower, talk to a TA since you may need to use a slower clock (8 MHz with 18pF).

*Step 4.* Solder the two LEDs and the two 330Ω resistors. The orientation of the LEDs is tricky, consult a datasheet if you are unsure (ask a TA if you are still doubtful). Then solder the reset button and the 0.1uF reset capacitor.

*Step 5.* Install avrdude and minicom

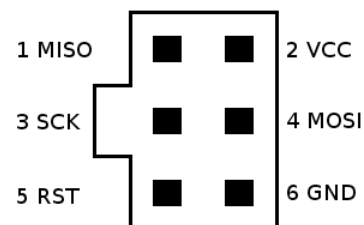
**\$ sudo apt-get install avrdude**

**\$ sudo apt-get install minicom**

**WARNING! Your board would be powered from this point onwards. Be careful where you place it! It might short if placed on metallic surfaces (like on your laptop)**

Step 6. Upload bootloader

Use the AVR mkII programmer to upload a bootloader. The pinout for the programmer is shown on the right (Red wire corresponds to pin 6 on the programmer header). Use the FTDI serial cable to supply power during this process as the ISP does not supply power. Make the appropriate connections.



Run once to check for any errors in connection:

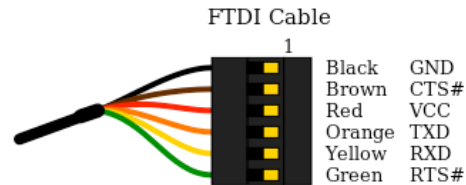
```
$ sudo avrdude -c avrispmkII -P usb -p m328p
```

If all goes well, your programmer should show a green light. (Check the manual for avrdude to understand what these commands mean).

Extract the provided 549\_lab2 tarball. In the `arduino_bootloader_328p` directory, run the following to burn the bootloader:

```
$ sudo make atmega328_isp
```

Step 7. Your board should be ready for programming and applications can now be downloaded using the FTDI cable. Use the pinout on the right to connect the cable in the correct direction (verify using your eagle file).



Connecting it the wrong way could burn some of your hardware.

Program the application file in the `uart_test_328p` directory using the following command:

```
$ sudo avrdude -b 57600 -F -e -c arduino -P [port location] -p atmega328p -U flash:w:[hex file path]
```

(To find your port location, use `$ dmesg |tail` before and after connecting the FTDI cable)

Step 8. Look at minicom for output

To test the application open minicom:

```
$ sudo minicom -s
```

Make sure to setup the serial port properly in the settings menu

Serial Device: [your device (port) location]

Bits: 115200 8N1

HW Flow: Yes

SW Flow: No

Type something & it should show up in minicom. Use Ctrl-A + Q to quit minicom and Ctrl-A + O to get back to the settings menu

Step 9. Show your TA to get signed off on lab 2