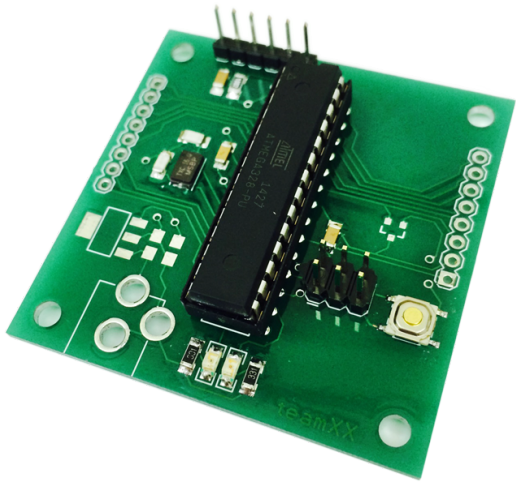
# 18-549 Embedded System Design: Lab 2 Assembly and Bring up

**Assigned: 02/20**
**Due: 02/26 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 | >= 1 |
| 6x1 Header | 1 |
| 3x2 Header | 1 |
| DC Power Jack | 1 |
| ATmega328p | 1 |
| 28-pin Chip Holder | 1 |
| Button | 1 |
| 5V Voltage Regulator (5209) *(Ask TA if you need 3.3V or other)* | 1 |
| 16 MHz crystal *(Ask TA if you operate at 3.3V or need 8 MHz)* | 1 |
| 18 pF Capacitor (for crystal) *(Ask TA if you operate at 3.3V or need 8 MHz)* | 2 |
| 0.1 uF Capacitor | 3 |
| 10 uF Capacitor | 2 |
| 330 Resistor (331) | 2 |
| 10K Resistor (1002) | 1 |
| 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 |

## 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 18pF capacitors. 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 330 Ω 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.

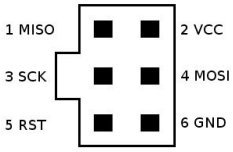
*Step 5.* Install avrdude, the avr toolchain, and minicom

```
$ sudo apt-get install avrdude avr-libc 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 Atmel ICE ISP 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:

**$ avrdude -c atmelice_isp -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).

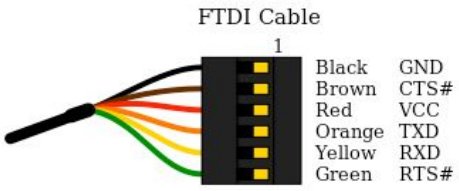
Clone the provided 549_lab2 project from the embedded.andrew.cmu.edu GitLab.

**$ git clone git@embedded.andrew.cmu.edu:549-s17-ta/lab2.git**

In the *arduino_bootloader* directory, run the following command to burn the bootloader:
**$ make atmega328_isp**

If you are using the 8MHz design, run `make atmega328_pro8_isp` instead.

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* directory using the following command:

**$ make program PRGM_PORT=***[port_location]*

To find your *port_location*, check `sudo dmesg | tail` before and after connecting the FTDI cable. The port location should be something like /dev/ttyUSB0 or /dev/tty.usbserial-AMA017Y3B

If you are using the 8MHz design, run `make program AVR_FREQ=8000000L PRGM_PORT=[port_location]` instead.

Step 8. Look at minicom for output

To test the application open minicom:
**$ minicom -s**

Make sure to setup the serial port properly in the settings menu
Serial Device: [*port_location*]
Bits: 115200 8N1
HW Flow: No
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

## Extra Notes

For further information about programming your board, please checkout the README.md*s in the lab2 repository OR the repository web front.
https://embedded.andrew.cmu.edu:5443/549-s17-ta/lab2