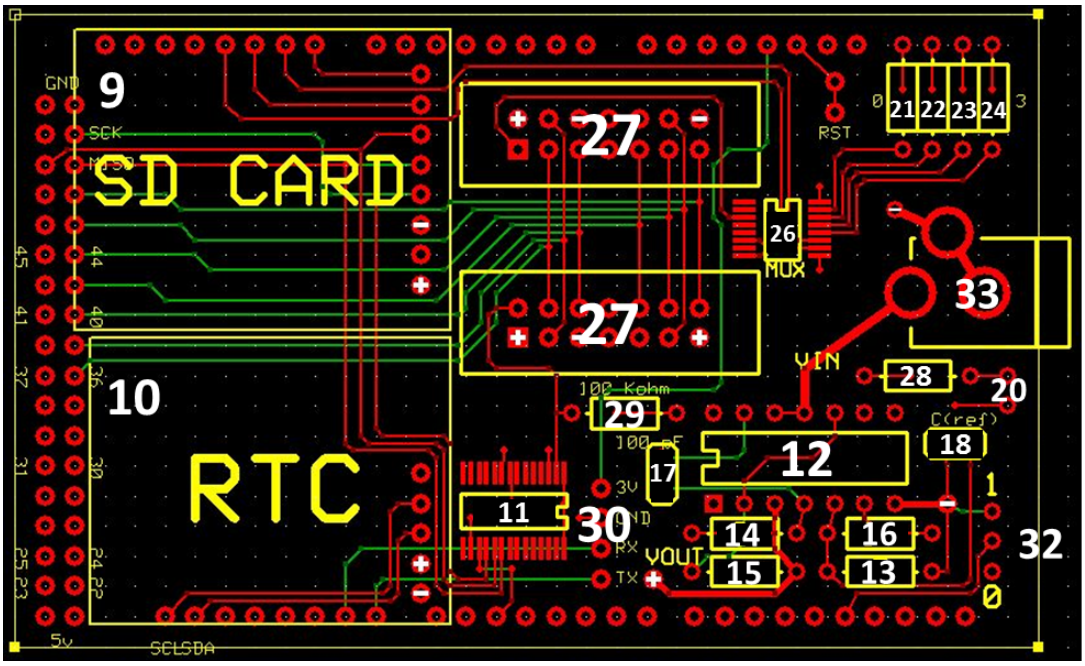
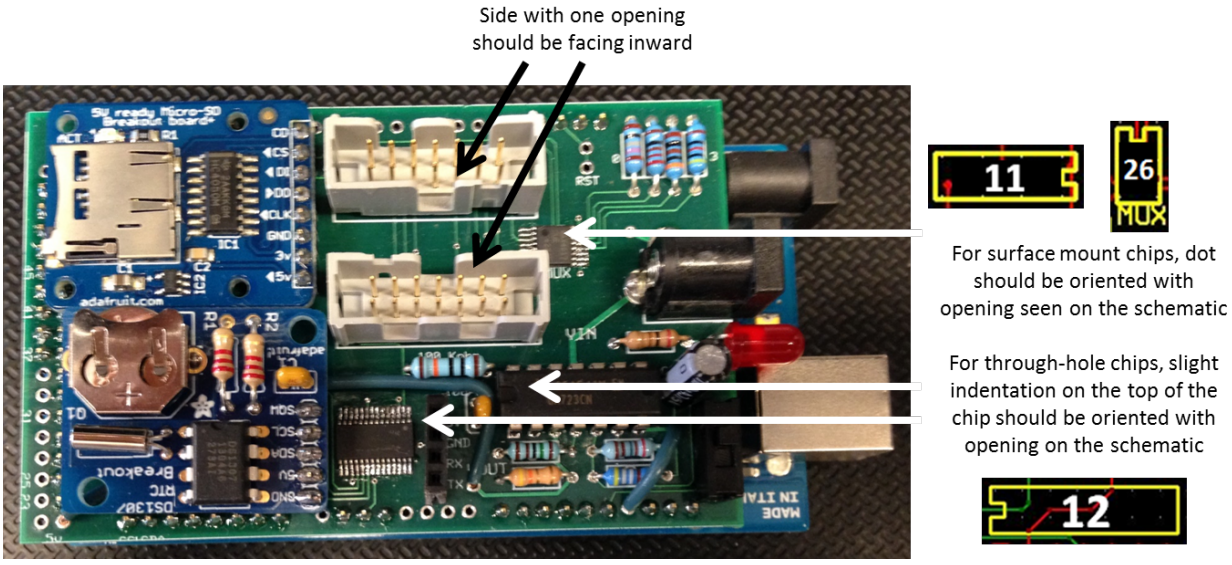
I. Assembling electronics

The Arduino Mega Shield:

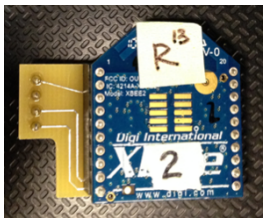
The following diagram shows the placement of each component (identified by the number in the parts list) on the PCB for the Arduino Mega Shield:



Note: For parts 11, 12, 26, and 27, pay close attention to the orientation on the boards.



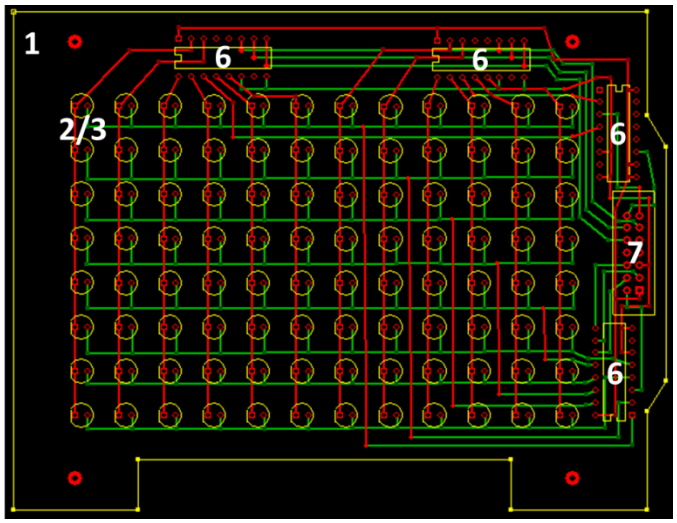
After soldering on all the components to the shield, solder on header in the external holes so that the shield can be placed in the Arduino. Both the barrel jack on the Arduino and the barrel jack on the shield should be on the same side (see picture above).



The XBee Shield will fit into the port labeled 30 on the first diagram. The XBee should face forward in the shield (the front end of the XBee, with the R13 label, should be on the side with the four pins). The four pins should be placed in the port labeled 30 with the front end (labeled R13) facing out from the center of the board.

The LED boards:

The following diagram shows the placement of each component (identified by the number in the parts list) on the PCB for the Arduino Mega Shield:

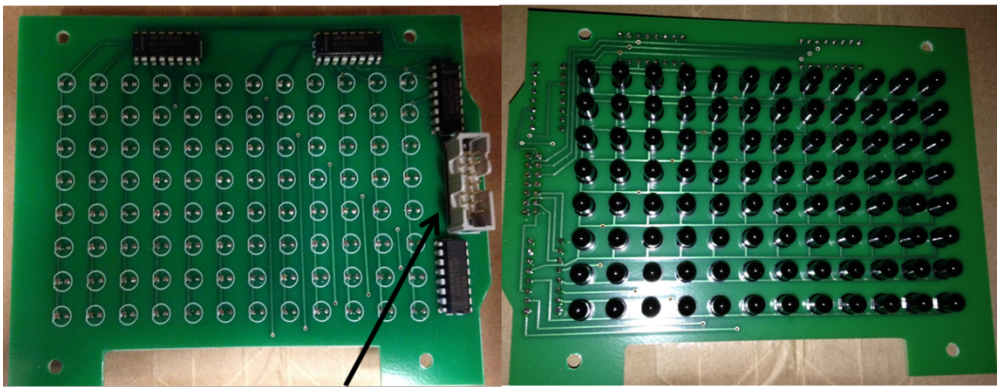


With the PCB in this orientation, the multiplexers (part 6) will always be placed facing outward, toward the user. As before, with through-hole chips, the slight indentation on the top of the chip should be oriented with the opening on the schematic.

Pay careful attention to the orientation of the LEDs with respect to the PCB – the emitters are placed on the front of the board and the detectors are placed on the back of the board.

Detectors:

Front **Back**

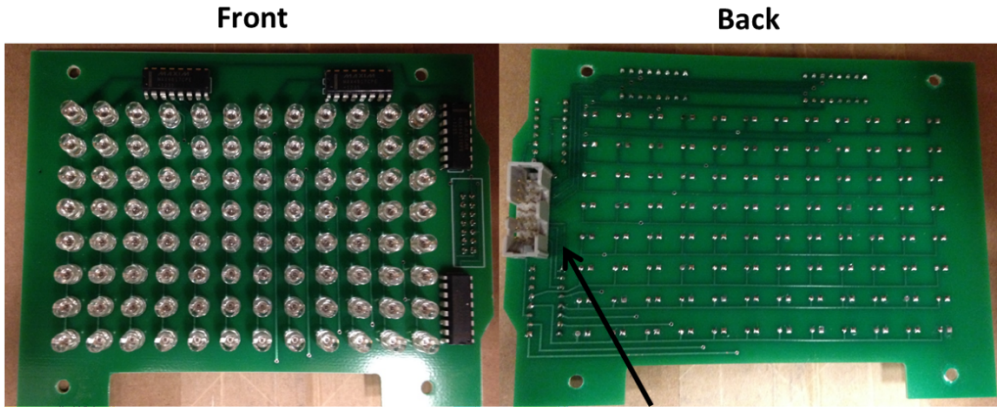


Multiplexers are facing outward.

Part 7: Side with one opening should be facing outward.

LEDs should be placed with the flat side facing to the left and should be flush with the PCB.

Emitters:

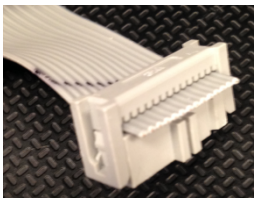


Multiplexers are facing outward.

LEDs should be placed with the flat side facing the left and should be flush with the PCB.

Part 7: Side with one opening should be facing outward.

Assembling the ribbon cables:

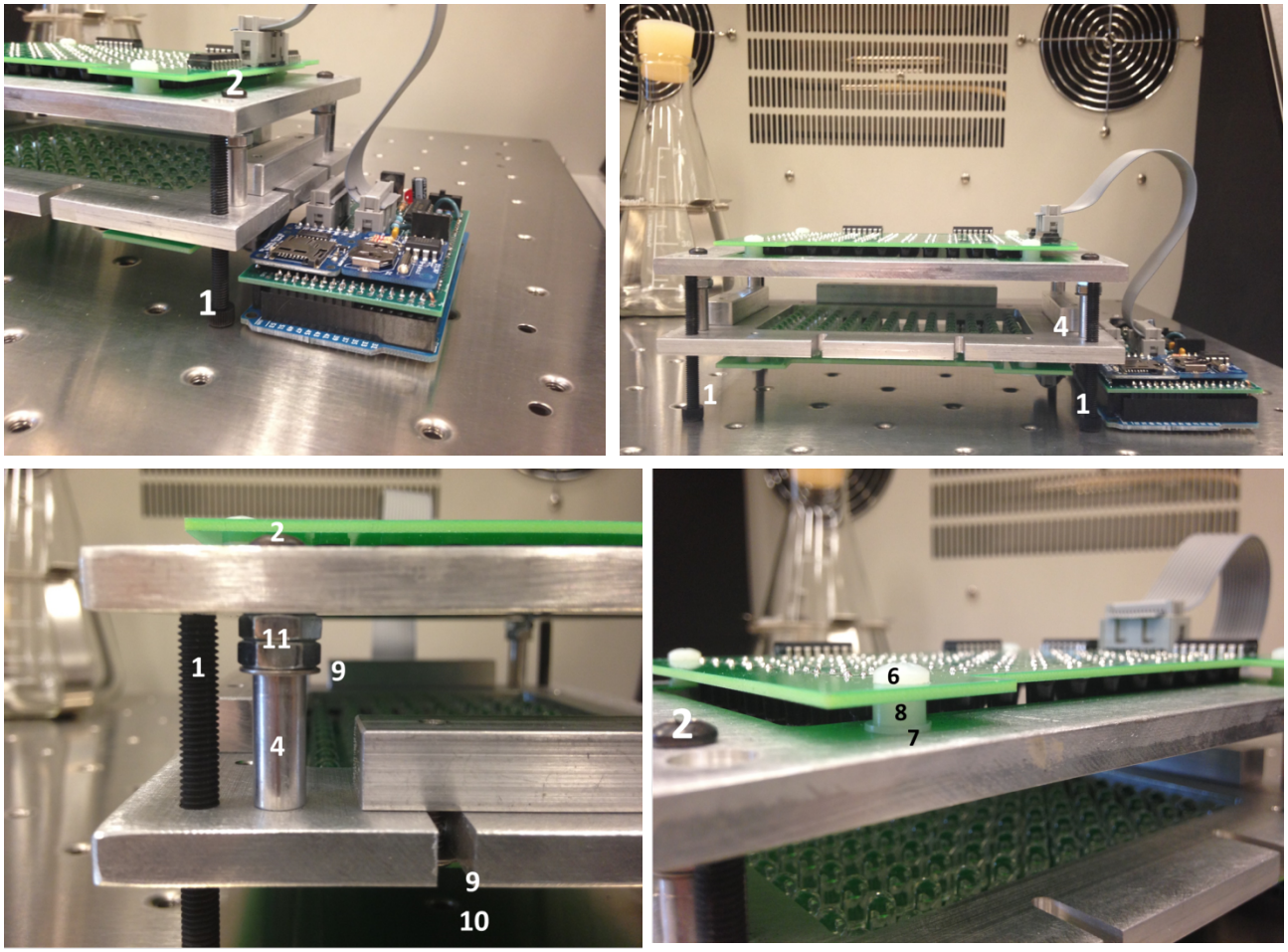


You will need two ribbon cables to connect the emitter and detector PCBs to the Arduino. For the ribbon cable connecting the emitter PCB and the Arduino, part 4 should be facing outward for both ends (see to the left). For the ribbon cable, connecting the detector PCB and the Arduino, part 4 of the ribbon cables should be facing inward for both ends.

For these experiments, metal frames were used to hold the electrical boards in place around a 96-well plate. The technical drawings for these plates are available along with the PCBs for the electrical components. Below are some images of our assembled device.

For assembling the aluminum frame, the part numbers in the following images refer to the parts in the Parts_List.xslx document under the tab labeled Aluminum Frame. Some notes on assembly:

- We recommend using at least one spacer (4), one washer (9) and two nuts (11) to space the distance between the emitter and detector boards.
- The washer (9) and screw (10) are used to secure the side rails to the metal frames to keep the plates in place. The positioning of these rails should be to line up each well of a 96 well plate with a corresponding LED pair.
- The nylon screws (5/6), washers (7), and spacer (8) are used to attach both the emitter and detector boards to the aluminum frames.



II. Loading code onto the Arduino

For an introduction to Arduino: Beginning Arduino by Michael McRoberts

All code for this project is available at: github.com/BonnieDougherty/PlateReader.

For all Arduino code, you must download the five libraries (LMP, Maxim, Plate, Radio, and Reporter) and save them in the libraries folder in the Arduino documents folder. Additionally, you must download the following libraries:

- OneWire (http://playground.arduino.cc/Learning/OneWire)
- RTClib (https://github.com/adafruit/RTClib)
- SDfat (https://code.google.com/p/sdfatlib/)
- XBee (https://code.google.com/p/xbee-arduino/)

Before uploading any code to the Arduino, upload the EEPROM_default script. Ensure that the entire script runs by monitoring it through the serial monitor.

For running a plate reader through Python, upload the file named WorkingXBeeCommunicationNoSD to the Arduino. This file will allow you to run multiple plate readers at the same time with help from an external server. All communication will be done through the XBee so be sure to have the XBee running with the proper settings.

For running a plate reader without Python (only using the Arduino), upload the file named PlateReaderWithSD to the Arduino. This file will allow you to run experiments without the aid of an external server. The files saved to the SD card will require additional processing before they can be analyzed.

III. Setting up a growth experiment

For these experiments, two aluminum frames were used to hold the emitter and detector PCBs as well as ensure that the wells in the 96 well plates aligned with the PCBs. There should be at least 1 ¾ inches between the emitter and detector PCBs to ensure that the LEDs can detect growth in your samples.

Attach the ribbon cables to their respective PCBs. The ribbon cable connecting the emitter PCB to the Arduino connects to the bottom port of the Arduino when the Arduino is oriented with the barrel jack to the right. The ribbon cable connecting the detector PCB to the Arduino connects to the bottom port of the Arduino when the Arduino is oriented with the barrel jack to the right.

If you are running the plate reader through an external server, ensure that the XBees (parts 34 and 35) are up and running. The Python XBee library requires that the coordinator be in API Enable mode 2. The Arduino library requires that each router also be in API Enable mode 2. A good resource for setting up your XBee is Building Wireless Sensor Networks by Robert Faludi.

If you are running the plate reader solely through the Arduino, ensure that the microSD card is in place before powering up the Arduino. Once you plug the Arduino in (part 19) using the barrel jack on the Arduino shield, watch to make sure the SD card initializes. You should see the red light by the SD card flash. The plate reader will start readings when the power switch (part 32) is in the ON position (down when the Arduino is oriented with the barrel jack to the right). Before plugging in the Arduino, ensure that the switch is in the off position. Default readings are 10 flashes per well and the entire plate is read every minute. To stop readings, simply turn the switch (part 32) to OFF.

IV. Running a growth experiment in Python (via external server)

Running the plate readers via an external server will allow you to run multiple plate readers at one time and change the reading parameters, specifically the time delay between reads, for your plate readers. Before running an experiment, double check the file destination address

where your data will be saved (line 149 in CollectData.py) and change if necessary. Add in the addresses of your Xbee routers to the top of the CollectData.py script. The script will identify the first XBee address as 1, the second as 2, etc. for the XBee ID. To run a growth experiment, run the Python script CollectData.py. Once started, the script will prompt you for file names for each plate reader in your experiment; the plate readers are identified by their XBee ID. The script will then ask for the read delay between plate reads in seconds followed by the number of reads that you want to collect. Once the experiment has begun, the script will print a message every time data is received.

V. Processing Data

Under the ProcessingData folder in GitHub, there are multiple files available for process the raw data from either the plate reader SD card or the text files from your server.