## Pre-Build Testing

First, test each pad for connectivity using a multimeter according to the device schematic in the build area. If there are any failed connections, patch this connection and continue testing. After testing every pad and correcting any errors in connectivity, confirm that all components are present before stuffing. If any components are missing, take note of the missing components so that replacements can be procured.

## Soldering

### Through-hole

For through-hole components, dip the leads of the component in flux to act as a glue as well as helping the solder to flow. Tack one lead of the component to hold it in place before soldering the other leads of the component. After soldering all other leads of the through-hole component, go back and solder the tacked lead.

### Surface Mount

For surface mount components, preheat the board to two hundred degrees and spread flux on the pad. Add a small amount of solder to the pads to help with connection using a heat gun or soldering iron. Using a set of tweezers, place the component in the desired orientation and heat both pads with the heat gun. Place the component on the heated pads and remove the heat gun and allow to cool.

### PIC Microcontroller

For the microcontroller, preheat the board to two hundred degrees and spread flux on the pads. Place the microcontroller in the desired configuration, making sure that all legs are aligned with their respective pads. After you have ensured correct alignment, apply heat using a heat gun to solder the microcontroller to the board. After 30 seconds of heat and making sure that all sides of the microcontroller are evenly heated, remove the heat gun while still holding the microcontroller in place for another 15 seconds. Check and make sure that the microcontroller is soldered by lightly nudging it with the tweezers.

## 

## Correcting Board Errors

There are several known errors on the board. Fortunately, these may be corrected by soldering copper wire between the points that are missing connections. Hardwire the following pads together:

* R6 (right side) to VO (on the LCD pinout)
* RD8 pin on the microcontroller to H2 pinout RD8.
* Pin number 37 on the microcontroller to SCL on H2
* Pin number 38 on the microcontroller to SDA on H2
* Pin number 63 on the microcontroller to LED pin number 9 (Only for Deepak’s board)
* Solder joints SJ1 to Vraw
* Solder joints SJ3, SJ5, and SJ6 to 3.3 V rail

## Correcting Board Errors

There may be disconnected pads on the board. These may be corrected by soldering copper wire between the disconnected pads. Determine errors by testing the pads against the pads they should be connected to according to the schematic.

## Post-Solder Testing

Test each pin as was done in Pre-Build Testing. Make sure to test connection by touching the pins if any are connected and not just the pads. If any components have not made sufficient contact, reapply or reheat the solder on the components to ensure a solid connection. If any pins of the microcontroller have not made a good connection, apply large amounts of flux to the area and apply extra solder to the unconnected pin, making sure to not bridge any of the pins across the microcontroller. After all pins have been checked for connectivity and connected, load the ledflash program, modified as necessary to check for the correct configuration of your board. If LED LED3 begins to flash, you have successfully soldered your board.

Stuffing, soldering, and partial-build testing procedure

T1:

1. Lab components were received and hence soldered onto the board following the schematics and parts datasheets.
2. Connection were ensured using multimeter. If any discontinuity found, were fixed immediately.
3. Wrote weather module on python and tested using testbench provided.
4. Downloaded all the software required for the labs throughout the semester. All the set up were made and scons were fixed to build files based on our file system.

T2:

1. Any further schematic errors were fixed. In the meantime, component selection and justification documents were prepared.
2. Partial build and test were done. Build and test documentation were done for several components assigned in the task manual.
3. Header file for all the hardware components were written. This file managed and fully utilized components like LEDs, SWs etc.
4. Wrote traffic signal controller specification based on the assignment provided in task manual.

T3:

1. Created team’s Bitbucket repository. Also, started populating repo with the lab contents.
2. Wrote ESOS services configuration files, task support files, and sample UI application. All the functionalities were implemented and demoed to TA.
3. All the files and folders were pushed into the team’s repository.

T4:

1. Populated potentiometer, LM60, and ADC part was onto the board. Temperature sensor LM60 was solder to IC1 pad. All the connections were ensured using schematics, datasheets, and multimeter.
2. Sensor service files were created that had generic code to utilize sensor reading and hardware-specific code development.
3. Wrote multiple sample ESOS application that exercised the newly created sensor service file.

T5:

1. LCD module by Newhaven was added onto the board and associated components (few resistors and capacitors).
2. Since the new hardware was added, initialization and configuration files for that hardware and the ESOS service file for that specific hardware were created.
3. Then, display synchronization functions were implemented. Finally, sample application was created that includes character module, Nibble-wide module, and temperature display module.
4. Not to mention, everything was updated into the team’s repository subsequently. All the functionality as assigned in task manual was demoed to TA.

T6:

1. DAC (U4) and voltage reference (IC2) is added in this lab and also EEPROM and External thermometer (DS1632) are used in breadboard.
2. Soldered SJ1 to Vraw, SJ3, SJ5, SJ6 to 3V3 rail, DAC chip select pin to RF1.
3. Used external breadboard to connect EEPROM and DS1631 through I2C protocol into the board. Set up were made such that SDA and SCL for both ICs were connected in parallel.
4. Connect VDD and GND for each chip. And also connect A0, A1, A2 pin to GND for EEPROM.
5. Wrote SPI and DAC functionality codes
6. Hex file produced after build using scons was uploaded into the board and demoed to TA.

T7:

1. Add flex on the pads, then place CAN female connector into the board. Once placed, solder the pins.
2. Use datasheet and schematic to make sure orientation of the CAN transreceiver (U3). Add flex on the pads and in the pins. Place the IC into the pad and solder one pin at a time making sure there is no bridge formation.
3. Use schematic to ensure the proper resistance of the resistor R8. Add flux and solder it on the board.
4. After soldering all components, use multimeter to ensure everything is work as good as expected. Then hex file can be loaded into the board and test the functionality.

T8:

1. No new components were added during this lab.
2. Wrote codes for task as assigned in manual and demoed to TA.
3. Utilized CAN, I2C, SPI, DAC components of the board to implement the task.