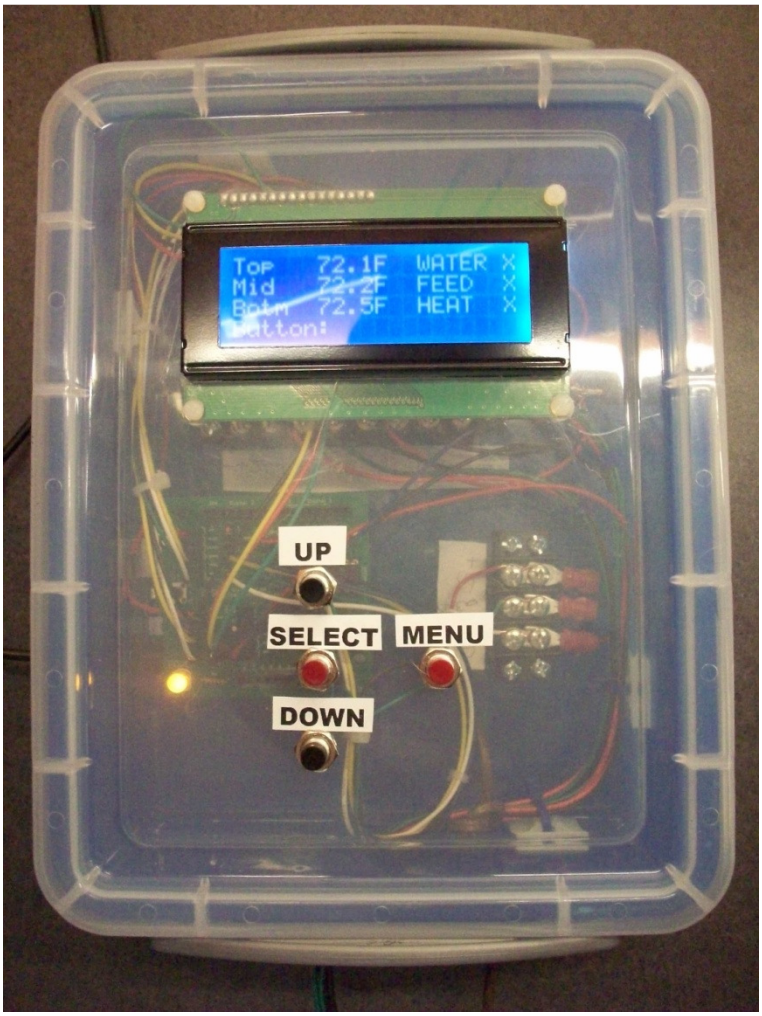
Open Source Controller

Instruction Manual

Release 1.1

Bob Glicksman

3/10/2012



Instructions for construction, installation and use of an Open Source Controller based upon an Arduino microcontroller. The Open Source Controller has been specifically designed to control the Curbie Open Source Still, but may be used in other applications, such as feedstock cooker and fermenter.

License and Disclaimer

The Open Source Controller, including but not limited to the items listed below, is released under a Creative Commons license Attribution-ShareAlike 3.0:

- Open Source Controller hardware, including the box design, Test Board, and the Shield Board.
- Open Source Controller software, including Controller Test Software, all libraries and the Arduino development environment software.
- Open Source Controller documentation, including this manual, "read_me" files, box cover template, and parts list.

To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

THE OPEN SOURCE CONTROLLER HARDWARE, SOFTWARE AND DOCUMENTATION IS PROVIDED TO YOU "AS IS," AND THE AUTHORS AND CREATORS MAKE NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR INFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGARDLESS OF THE FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILITY MAY BE ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

The Open Source Controller is intended for hobbyist applications and for "do it yourself" assembly and use. The user assumes all responsibility and liability for any damages, injuries or consequences resulting from the construction, implementation and use of this product or any part thereof.

Table of Contents

Contents

Acknowledgements

The author would like to acknowledge the hard work and excellent advice supplied by Curbie and by Thurmond Moore in the functional and physical design of this Controller. The author would like to thank Thurmond Moore for performing a test build of the Controller from the instructions in this Manual, for correcting errors that he found in the instructions, and for making excellent suggestions for the improvements in process and parts that are described in Appendix C of this Manual.

1 Introduction.

1.1 Function and Purpose.

The Open Source Controller (Controller) is a software programmable controller that has specifically been designed to control the Curbie Open Source Still, although it can be used for other purposes as well. The Controller is based on the Arduino Uno microcontroller board (http://arduino.cc/en/Main/ArduinoBoardUno). The Controller is supplied with two software packages: (1) a controller test software package useful for testing the assembled Controller as well as a template for custom software applications; and (2) software specifically developed to run the Curbie Open Source Still. The latter software package is described in the documentation for the Curbie Open Source Still construction and operation.

The Controller has a 4 line, 20 character-per-line liquid crystal display (LCD) and a set of navigation pushbuttons for navigating through setup and selection menus displayed on the LCD. Inputs to the Controller are temperature probes based upon the Maxim / Dallas Semiconductor DS18B20 chip (http://www.maxim-ic.com/datasheet/index.mvp/id/2812), which is an inexpensive but highly accurate temperature sensor. The Controller's "Electronic Module" can accept additional digital and analog inputs; however, these instructions do not include wiring to anything other than the one-wire bus that connects the temperature probes.

The Controller provides 7 external digital outputs for controlling devices such as relays, solenoids, heaters and motors. The Controller outputs are 5 volt logic-level signals intended to control a variety of high power driver modules that are mounted externally to the Controller box. The Controller is specifically intended to control alcohol production and distillation equipment, which may require that high power switching modules and devices be enclosed in explosion proof electrical housings and not within a plastic Controller box.

The Controller also has an internal audible alert module that is used by the software to alert an operator when conditions for manual intervention exist.

> ***Note**: The Controller is housed in a plastic box that is available at Office Depot stores and on-line. Some of the photographs in this manual are for a box that is only available in retail stores and not on-line. The parts list (Appendix A) specifies a slightly different box that is available both on-line and in retail stores. These two boxes are the same volume and differ only slightly in size and color. The box listed in the parts list is clear all over, whereas the retail-only box is blue with a clear cover. The dimensional differences between these boxes is very slight and do not impact any instructions in this manual.*

1.2 Tools and Equipment.

> ***Note***: *This manual contains complete instructions to assemble and test the Controller from parts that can be purchased in local stores or over the Internet. It is strongly recommended that the reader familiarize him/herself with this manual prior to starting a project to build a Controller.*

This manual contains complete instructions to assemble and test the Controller from parts that can be purchased over the Internet. It is assumed that the user has available a basic set of woodworking hand tools, including:

- Screwdrivers (straight and Phillips)
- Hammer
- Chisels
- Small files
- ¼" Hand drill (electric or manual) and set of metal and wood bits.
- Assortment of C-clamps and bench vise (or other means to hold material in place while drilling and cutting)
- Box knife and/or plastic cutter knife, and/or a nibbling tool.
- Electrical soldering and assembly tools (see section 4.1 and chapter 5 for further details).

It is recommended that the user read through this manual and review the assembly instructions prior to starting any of assembly work, in order to ensure that the necessary tools and materials are on hand.

This manual is part of a complete assembly package for the Controller. The entire package consists of the following items, in addition to this manual:

- Controller software: test software, driver and libraries that are needed to run and test the Controller. These are bundled in a compressed archive (zip) file.
- Parts List: available as an Excel spreadsheet, sorted by sub-assembly and by vendor.
- Eagle CAD Files: the CAD files for the custom Shield Board, released under open source licensing so that others can make improvements and beneficial alterations to the design.

Make sure that you have all materials, tools, software files and parts before starting assembly of the Controller.

***Note**: A complete documentation package is posted at:*
http://www.liquidsunenergy.com/controller/controller.html

In the future, this package may also be posted in the FILES section of the Alcohol Fuels Yahoo chat group (http://tech.groups.yahoo.com/group/alcoholfuel) and on other Web sites as well.

1.3 Organization of this Manual.

This manual provides instruction to:

- Connect the completed Controller to external devices (chapter 2).
- Install the software in the completed Controller and test that it is working properly (chapter 3).
- Build the Controller from scratch, using parts that can be purchased over the Internet and a custom built Shield Board (chapter 4).
- Build the Shield Board, using parts that can be purchased over the Internet (chapter 5).
- A complete list of parts and ancillary materials (Appendix A).

At this time, it is necessary for the user to assemble the Controller from scratch. In so doing, it may be helpful to read the manual in chapter order, but to assemble the Controller in reverse chapter order, i.e.:

- Order and receive all parts and materials (Appendix A).
- Assemble the Shield Board, which is part of the Electronic Module (chapter 5).
- Assemble the Controller (chapter 4).
- Install the software development environment and load the software to the Controller (chapter 3).
- Test the Controller with the test software (chapter 3).

It is possible that, in the future, some parts or even the entire Controller may be available for purchase as assembled and tested units. In this event, and if the user elects to purchase pre-assembled units, the chapters of this manual related to assembly of that unit may be skipped.

1.4 Overview Tour of the Controller.

Figure 1-1 depicts the internal components of the Controller. The Controller's cover contains a 4 line x 20 character per line liquid crystal display (LCD) that connects to the Shield Board that is atop the Electronic Module. An Arduino Uno microcontroller board (figure 1-2) resides underneath the Shield Board and the two are interconnected via the Uno's header sockets that are provided specifically for this purpose.

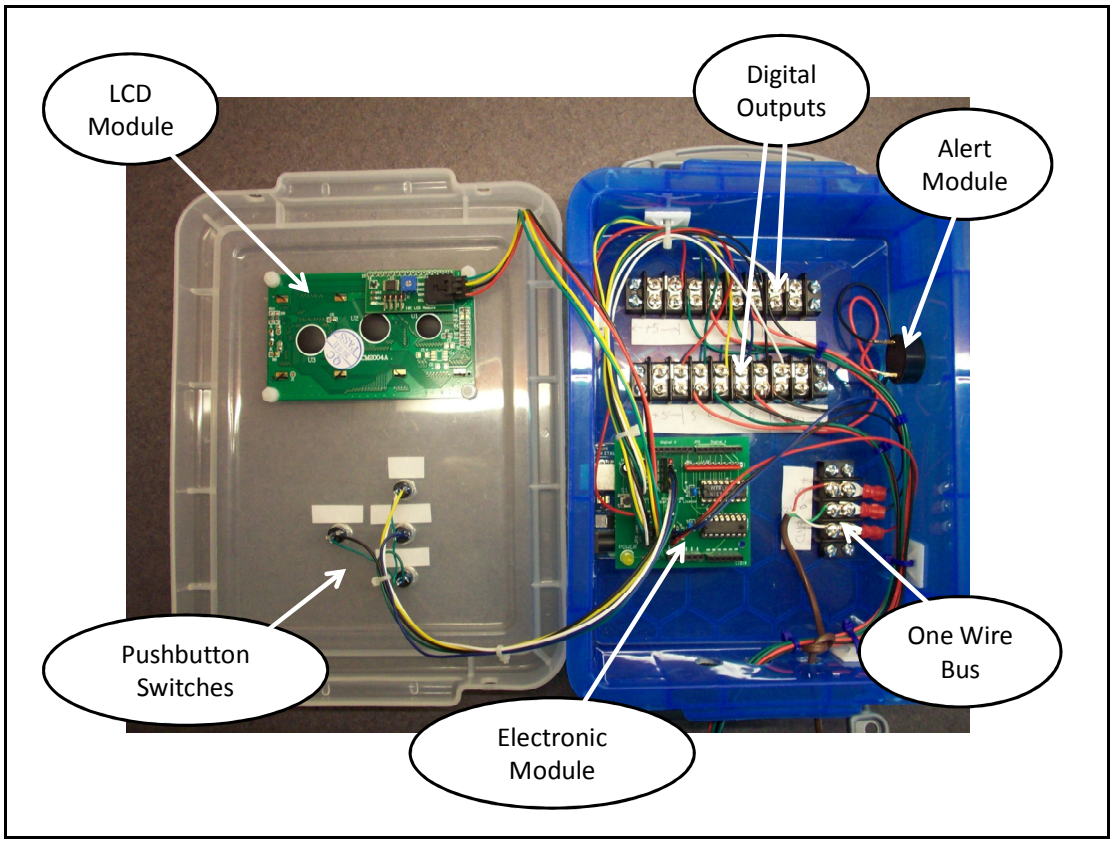


Figure 1-1. Controller internal parts.

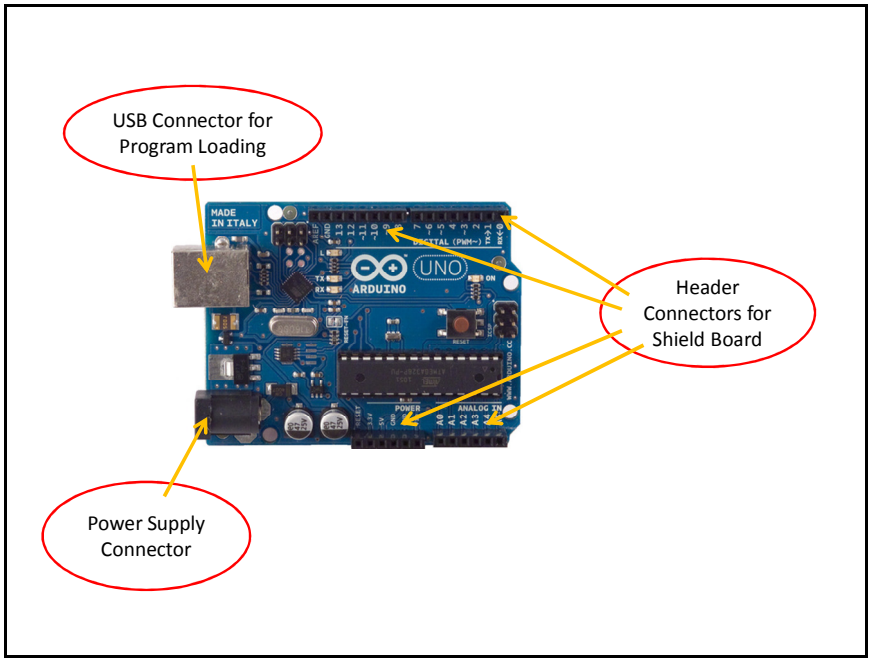


Figure 1-2. Arduino Uno Microcontroller Board.

The Controller's cover also contains 4 pushbutton switches for navigation and user input (figure 1-3):

- MENU: the MENU button is used by the microcontroller's software to display menu choices for the user.
- UP / DOWN: the UP and DOWN buttons are used by the microcontroller's software to allow the user to navigate through menu selections displayed on the LCD display.
- SELECT: the SELECT button is used by the microcontroller's software to allow the user to enter and execute the current menu selection.

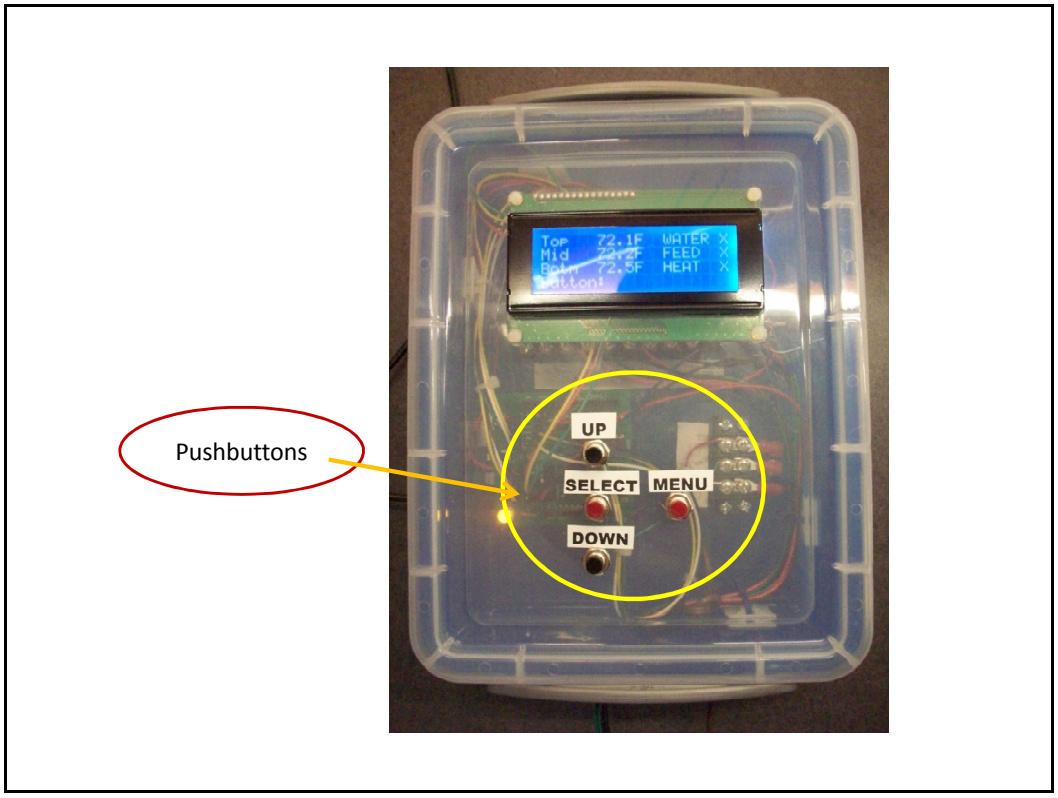


Figure 1-3. Pushbuttons.

The Controller is powered through the power supply connector on the Uno board (figure 1-2). A plug-in modular "wall wart" power supply (e.g. http://www.adafruit.com/products/63) is recommended. Alternatively, a 9 to 12 volt battery may be used for short duration power or backup power. The Controller can also be powered through the Uno's USB connection, which is additionally used for loading software into the Uno microcontroller. Software is uploaded to the Uno from a host PC running the Windows, Mac OSX, or Linux operating systems. The USB power must be a regulated 5 volt supply.

2 Connecting the Controller to the Outside World.

2.1 Physical Connection.

The inside of the controller box is shown in figure 2-1. Terminal blocks are provided for wiring the Controller to the outside world.

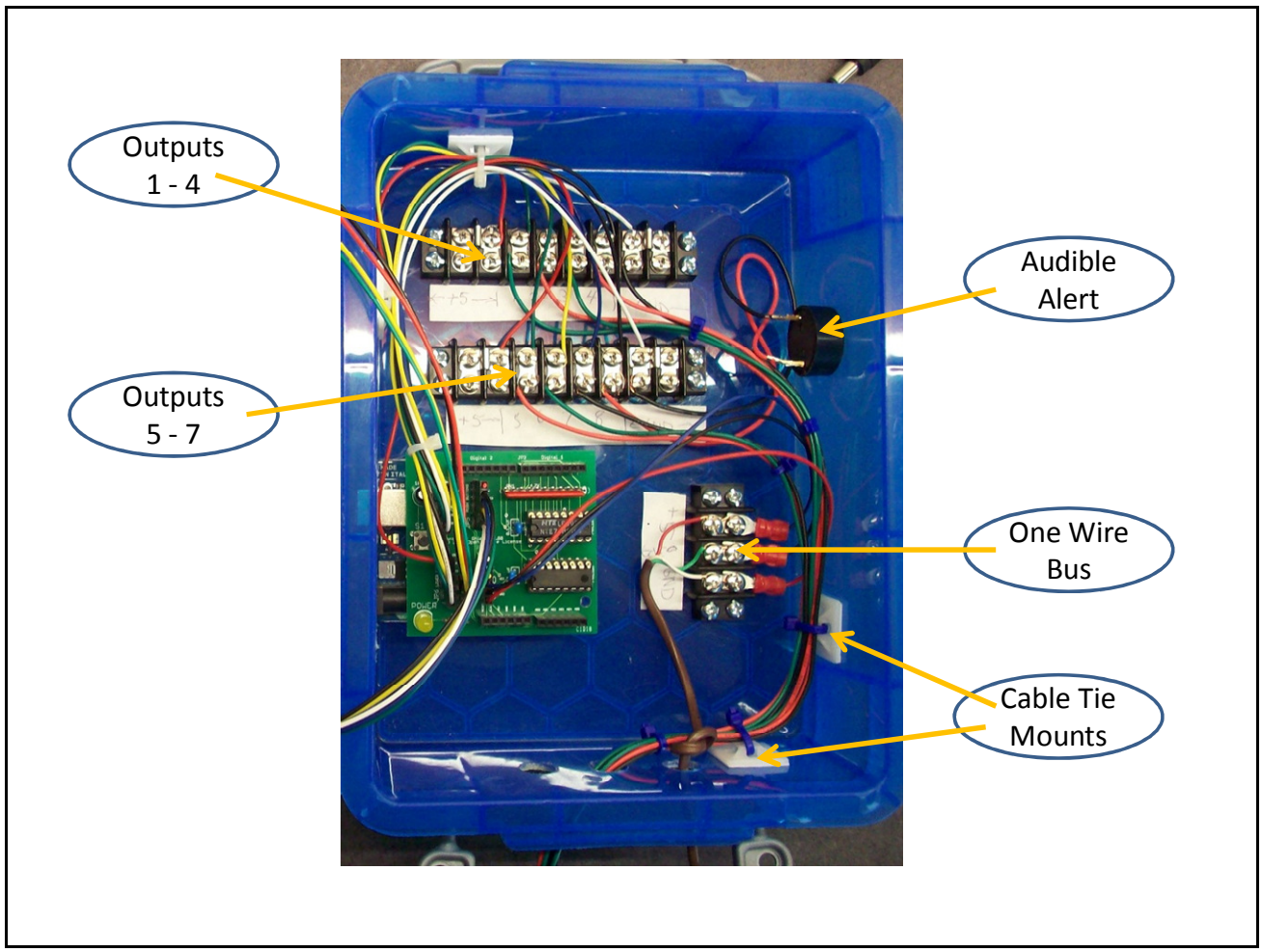


Figure 2-1. Terminal Blocks for External Connections.

A three position terminal block located in the lower right hand side of the box and is used to provide the "one-wire bus" connection to the temperature probes. The one-wire bus actually consists of three wires: +5 volts, ground, and the bus itself.

Two, 8-position terminal blocks are located in the upper part of the box, as shown in figure 2-1. The upper terminal block has two connections for +5 volts (leftmost two connections), two connections for ground (rightmost two connections) and four connections for the signals, with output #1 on the left, followed by output #2, output #3, and output #4. The lower terminal block has two connections for +5 volts (leftmost two connections), two connections for ground

(rightmost two connections) and four connections for the signals, with output #5 on the left, followed by output #6, and output #7. The last output signal, #8, is used to wire up the audible alert module that is cemented on the side of the box.

> ***Suggestion:*** *It is suggested, but not required, to use stick-on cable tie mounts to dress and secure wires coming off of the terminal blocks and exiting the Controller box, as shown in figure 2–1. The cable tie mounts keep the wiring neat inside of the box and provide strain relief to the wires exiting the holes in the bottom of the Controller box. Sources for cable tie mounts and cable ties are included in the Controller parts list (Appendix A).*

Figure 2-2 shows a close-up of the one-wire bus terminal block. Connector screws are labeled for +5 volts, the bus wire (OWB), and ground (GND). All three of these signals must be brought out to all one-wire devices. The Curbie Open Source Still uses three temperature probes. Each probe contains a DS18B20 one-wire temperature sensor chip. The three probes are connected as stubs off of a bus that contains these three signals. Section 2.2, below, describes the characteristics of the one-wire bus and the wiring of the temperature probes.

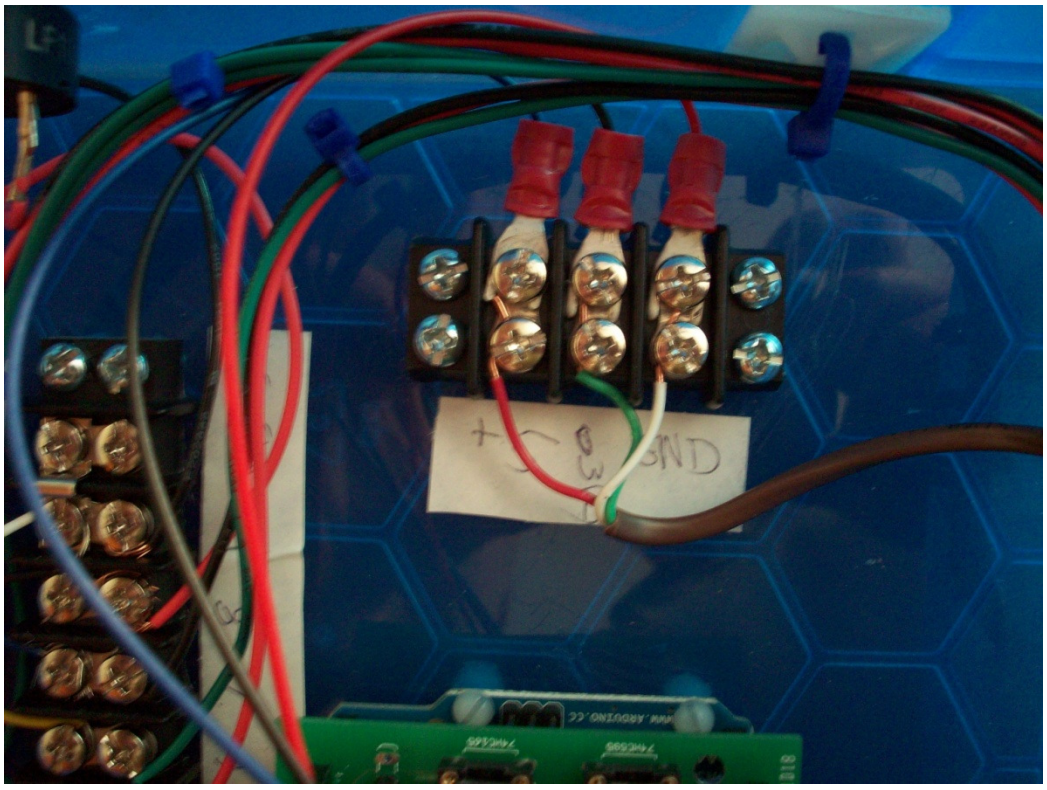


Figure 2-2 One-Wire Bus Terminal Block Detail.

Figure 2-3 shows a close-up of the lower and upper digital output terminal blocks. The terminal blocks have been configured so that each output can be paired with either a +5 volt power or a Ground. An output that is paired with a +5 volt power is intended to be driven “active-low”,

while an output that is paired with Ground is intended to be driven "active-high". "Active-low" and "active-high" driving is described further in section 2.3, below.

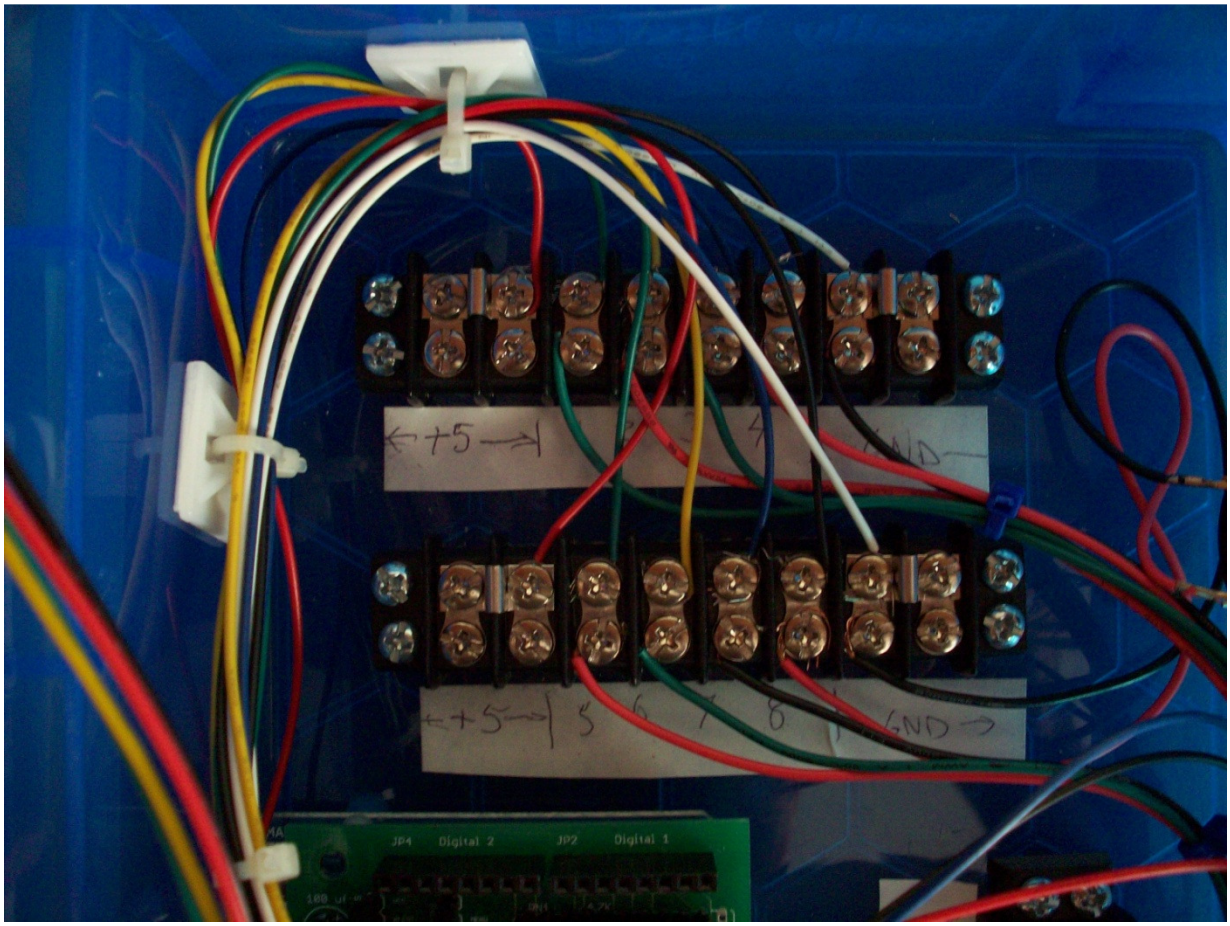


Figure 2-3 Output Terminal Block Detail.

The assignments of outputs 1 – 7 for the Curbie Open Source Still are as follows:

- Output #1: Heater power to the reboiler.
- Output #2: Water feed pump or valve control.
- Output #3: Beer feed pump or valve control.
- Outputs #4 - #6 are currently spare; reserved for future still improvements.
- Output #7: Scram alarm.

As previously stated, output #8 is internally connected to the audible alert module.

The nominal arrangement of the output terminal blocks is to have two terminals for +5 and two terminals for ground on each terminal block. These can easily be changed by changing the arrangement of the terminal block jumpers, which are depicted in figure 2-4.

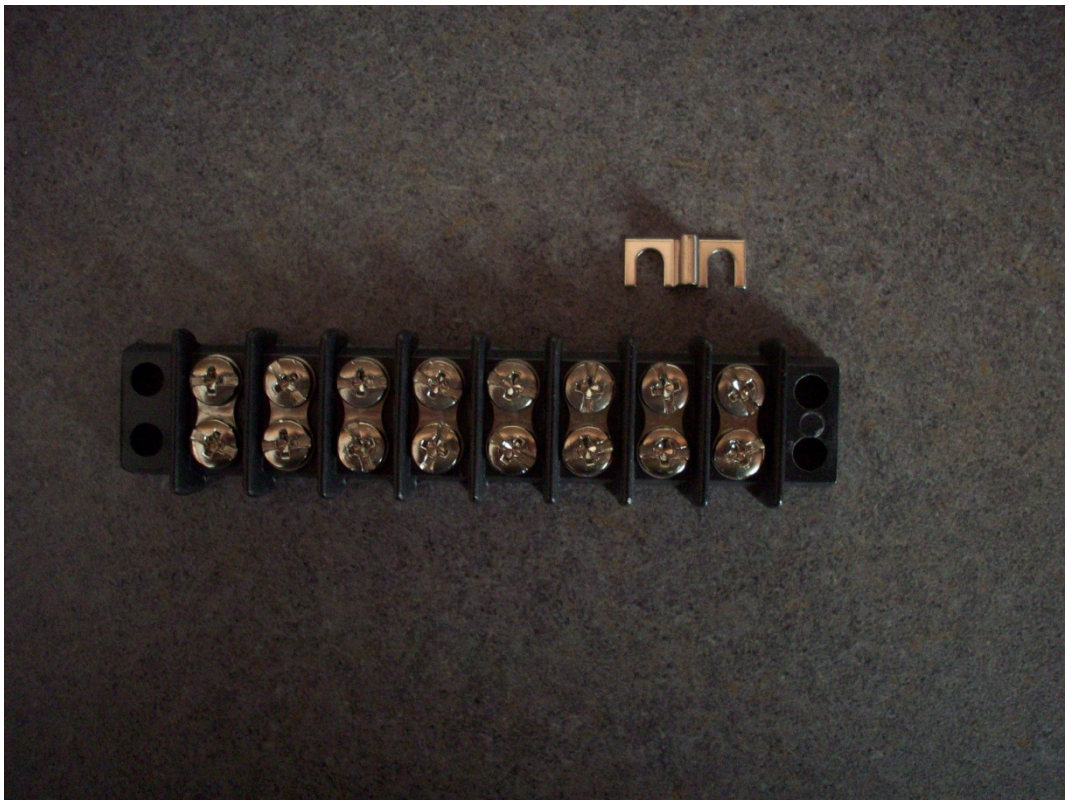


Figure 2-4 terminal block jumper.

Terminal blocks allow a variety of wire types and gauges to be used to connect external devices to the Controller. Any wire gauge #18 - #28 can be used, either solid or stranded wire. #22 AWG stranded wire is recommended, but any wire that meets these specifications will do. The wires carry no more than 5 volts and no more than 20 mA of current each. A solid wire can be connected directly to a screw terminal. A stranded wire can also be connected directly to a screw terminal; however, it is recommended that the wire either be tinned to make it more solid, or (better still) that a spade lug be crimped to a stranded wire using lugs and an electrician's tool as depicted in figure 2-4. A spade lug makes an excellent connection to a screw terminal that will not rotate and break off when the screw is tightened. Spade lugs are color-coded by wire gauge – use the red ones for #22 AWG stranded wire.

Figure 2-4. Spade Lug and Electrician's Tool

2.2 One-Wire Bus.

The Maxim / Dallas Semiconductor "one-wire bus" is used to connect the temperature probes to the Controller. This bus actually consists of three wires: +5 volts (Vcc), ground (Gnd), and the bus itself (OWB). Every one-wire device connects to these three bus lines. The Controller has been designed and tested with three DS18B20-based temperature probes, but there is essentially no limit on the number of temperature probes and/or other one-wire bus devices that can be connected to this bus.

The one-wire bus is internally (to the Controller) terminated with a 4.7K ohm pull-up resistor. This resistor resides on the Shield Board of the Electronic Module inside of the Controller. Devices are "daisy-chained" on the one-wire bus cable. The maximum length of the bus cable is 10 meters (25 feet). Each device coming off of the bus (except for the furthest one) is a "stub" and the maximum length of a stub wire is 3 meters (7.5 feet). Figure 2-5 depicts the one-wire bus connection to three temperature probes. The Curbie Open Source Still documentation describes, in detail, the physical connection of the temperature probes to the main bus, and the main bus routing from the terminal block in the Controller, up the still column.

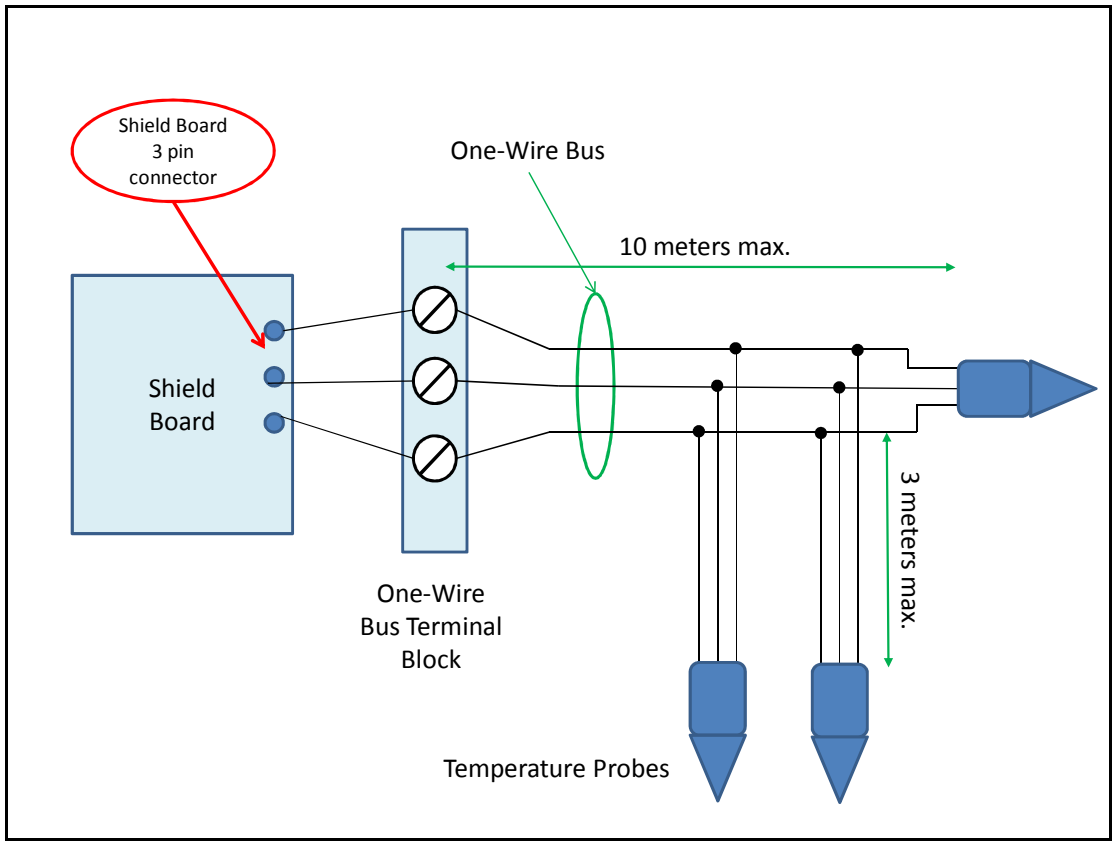


Figure 2-5. One-Wire Bus Temperature Probe Connection.

Full specifications for the Maxim / Dallas Semiconductor one-wire bus can be found at: http://www.maxim-ic.com/app-notes/index.mvp/id/148.

The Controller implementation does not use the "parasitic power" mode of the one-wire bus, since this mode is not compatible with the DS18B20 temperature sensor chips. A DS18B20 data sheet can be found at: http://www.maxim-ic.com/datasheet/index.mvp/id/2812.

2.3 Digital Outputs.

A total of 7 digital outputs are provided by the Controller. These digital outputs are CMOS logic level signals. When the signal bit is on, the output is +5 volts. When the signal bit is off, the output is Ground.

The digital outputs are supplied by a 74HC595 latched shift register. A data sheet for this device can be found at: http://www.jameco.com/Jameco/Products/ProdDS/46105.pdf.

Each output line can source or sink 20 mA of current. However, the total of all 7 outputs should not exceed 70 mA. Thus, if all 7 outputs are used, each output should source or sink no more than 10 mA each.

The digital outputs are meant to send control signals to various types of power driver devices. Samples of these devices are:

- **4 channel DC output board**: http://arduino-direct.com/sunshop/index.php?l=product_detail&p=60. Suitable for driving DC loads, specifically 12 volt DC motors used in the Curbie Open Source Still.
- **AC "Power Switch Tail"**: http://powerswitchtail.com/default.aspx. Easy and safe way to control a 110 or 220 VAC load, such as the reboiler heater for the Curbie Open Source Still.
- **Industrial DC Output Module**: http://www.alliedelec.com/search/productdetail.aspx?SKU=70133552. Industrial quality module for driving DC loads.
- **Industrial AC Output Module**: http://www.alliedelec.com/search/productdetail.aspx?SKU=70133541. Industrial quality module for driving AC loads.
- **Optically Isolated Relay Board**: http://b2cqshop.com/products/195-2-channel-relay-module-5v-for-arduino-pic-arm-avr-dsp.aspx. Inexpensive board with 2 optically isolated electro-mechanical relays that can drive either DC or AC loads.

The digital outputs from the Controller are not designed to drive high current loads, nor are they designed to drive long lengths of cable. A maximum of 10 feet of cable should be used between the Controller output terminal block and one of the aforementioned modules. Shielded cable is not required and is not desirable due to the added capacitance. It is desirable, although not required, to run each digital output as a twisted pair, paired with either +5 volts or Ground, depending upon active-high or active-low control (these terms are described further below).

The power driver devices listed above all use optically isolated (opto-isolated) connections between the Controller Electronic Module power and the AC or DC power for the device being controlled (solenoid, motor, heater, etc.). Opto-isolation is not strictly required but is highly recommended.

Figure 2-6 depicts the principle of opto-isolation. An opto-isolator is a device that contains a light emitting diode (LED) and a photo-sensitive device (e.g. photo-transistor or photo-triac), both enclosed in a light-tight package. The LED is driven by the Controller and the photo-sensitive device switches power to the device being controlled. There is no electrical connection between the Controller power and the controlled device power, which helps ensure that accidental short circuits, arching, and other problems do not damage the sensitive Controller electronics.

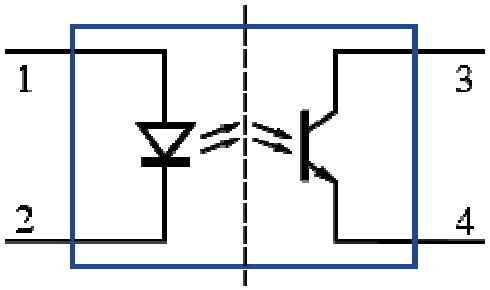


Figure 2-6. Opto-Isolator

From the Controller standpoint, driving an opto-isolator is exactly like driving an LED indicator. The Controller supplies a voltage to a series connection of a current limiting resistor and an LED and then completes the electrical circuit to ground. When the LED is on, the opto-isolated device is energized (on) and when the LED is off, the opto-isolated device is de-energized (off).

There are two ways for the Controller Electronic Module to power an opto-isolated device: "active-high" and "active-low". Figure 2-7 shows the active-high connection from the Controller.

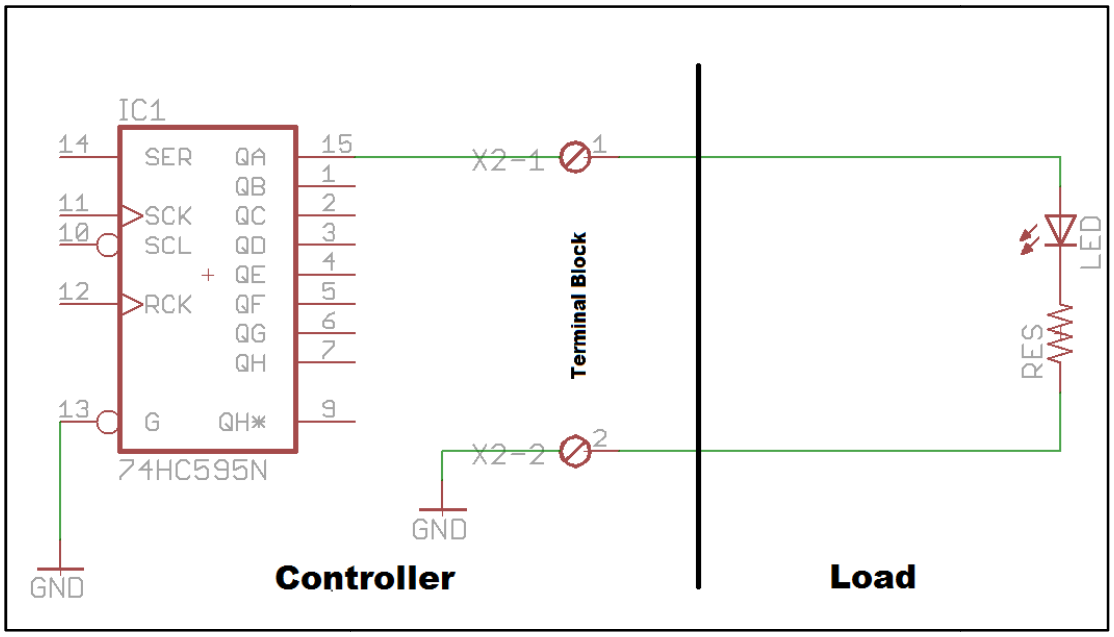


Figure 2-7. Active-High Connection to Power Driver.

In this configuration, the 74HC595 shift register supplies a control signal out to a screw terminal of the digital output terminal block. The opto-isolated power device, containing an LED and current limiting resistor on its control terminals, has its positive side connected to the shift register output bit and its negative side connected to Controller ground. When the control bit is a "1" (on), the output signal supplies +5 volts to the opto-isolated input, which turns on the LED

and thus activates the device connected to the power driver's load side (the load side is not shown on the figure). When the control bit is a "0" (off), the output signal supplies 0 volts to the opto-isolated input, which turns off the LED and thus de-activates the device connected to the power driver's load side.

Another way to drive an opto-isolated power driver is "active-low", depicted in figure 2-8.

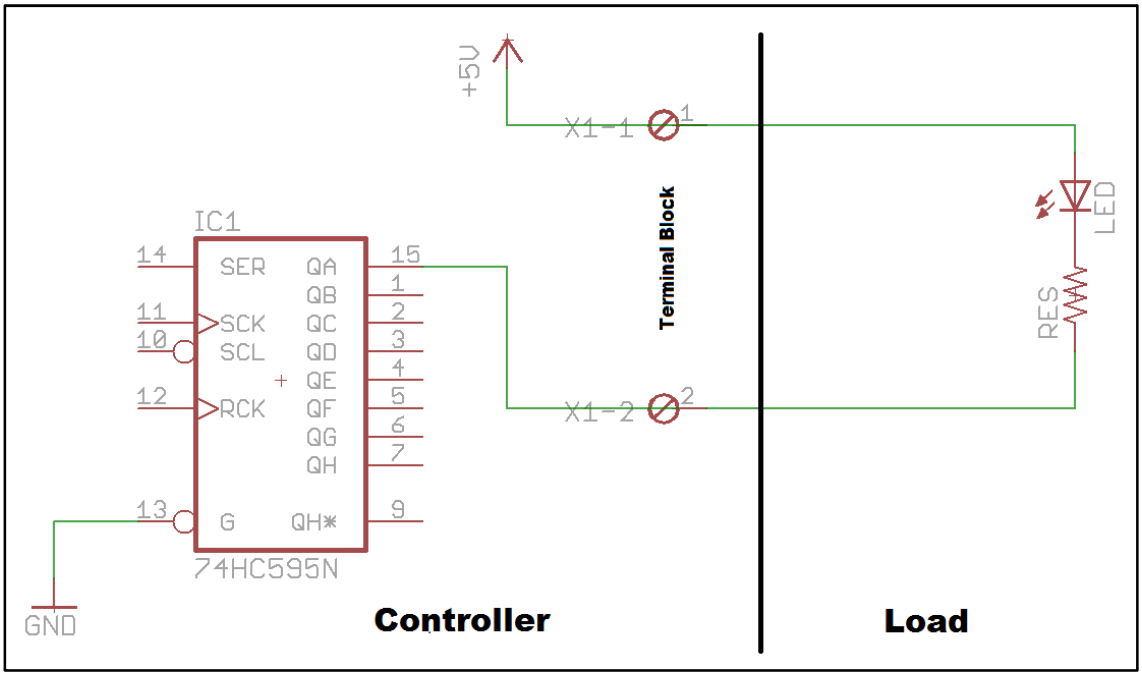


Figure 2-8. Active-Low Connection to Power Driver.

In this configuration, the 74HC595 shift register supplies a control signal out to a screw terminal of the digital output terminal block. The opto-isolated power device, containing an LED and current limiting resistor on its control terminals, has its negative side connected to the shift register output bit and its positive side connected to Controller +5 volt power. When the control bit is a "0" (on), the output signal supplies ground (0 volts) to the opto-isolated input, which completes the circuit thereby turning on the LED and activating the device connected to the power driver's load side (the load side is not shown on the figure). When the control bit is a "1" (off), the output signal supplies +5 volts to the opto-isolated input, which turns off the LED (zero voltage across it) and thus de-activates the device connected to the power driver's load side.

The shift register devices on the Controller's Electronic Module are equally capable of either "sourcing" current at +5 volts or "sinking" current at ground, and therefore it does not matter whether active-high or active-low activation is used. Whenever possible, it is recommended to use the active-high configuration, simply because it does not expose the Controller +5 volt power supply to inadvertent short circuiting. However, some types of multiple channel power driver modules "bus" one side of all LED/resistor combinations to a common connection, thus dictating

that active-high or active-low MUST be used. To this end, the Controller Test Software and the Curbie Open Source Still software both provide a setting for each of the Electronic Module digital outputs to be active-high or active-low. This permits a wide variety of such power driver modules to be used with the Controller.

3 Installing the Software.

The Electronic Module of the Controller contains an Arduino Uno microcontroller board. This board contains a programmable microcontroller (computer chip) that executes software loaded into its on-board flash memory. In order for the Controller to perform some specific function, software must be loaded into the Uno board's microcontroller memory. This process is called "uploading" the software, and is explained below. Once software is loaded onto the Uno board, it resides in a flash memory, which is "non-volatile". Non-volatile simply means that the flash memory retains its contents even when the power is removed (this is the type of memory used in USB memory sticks and micro-SD cards). Therefore, once the software is loaded into the Controller, it never needs to be uploaded again (unless the Controller functionality is to be changed and/or updated).

> ***Note**: If you can find someone to upload the proper program into your Controller Uno board, you can skip the rest of this chapter. However, it is strongly advised that you learn about the Uno development environment and install the development software so that you can install released updates and even make your own patches and modifications to the software.*

3.1 Software overview.

In order to upload software to your Controller's Uno board, you need to install the "Arduino integrated development environment" (Arduino IDE) on a PC. There are versions of this environment for PCs running the Windows (XP through version 7), Macintosh OSX, and Linux operating systems. Once the Arduino development environment is loaded onto your PC, you need to install the special libraries that are used by the Controller design into the development environment, and then load in the software that runs the Controller. You then use the Arduino development environment to "compile" the software, i.e. translate the software and its library calls into a form that can be executed by the Arduino Uno board, and then "upload" the compiled software to the Uno board. This all sounds a lot more formidable than it really is, since the software is already developed and tested. The sub-sections of this chapter will lead you through the steps, as follows:

- Step 1: Install the Arduino development environment on your PC.
- Step 2: Add the custom software libraries that the Controller uses to the installed Arduino development environment.
- Step 3: Install the software that operates the Controller. Note: these instructions install the Controller Test Software that is used to test your controller for proper operation. Instructions for installing the software that runs the Curbie Open Source Still are the same as these; only the software package is different.
- Step 4: Compile, Verify and upload the Test Software (or Curbie Open Source Still software) to your Controller (Uno board) using a USB cable.

- Step 5: Use the Test Software to verify correct operation of the fully assembled Controller.

The Arduino development environment can run on PCs running various versions of the Windows, Macintosh and Linux operating systems. Screen shots used in this chapter are from a PC running Windows 7. The screens for other operating systems are similar.

Figure 3-1 shows a completed Controller connected to a host PC (Windows laptop) via a USB cable. This is how the software gets uploaded to the Arduino Uno board inside the Controller. Note that the USB bus can power the Controller, so connecting an external power source to the Controller's power connector is not required when it is connected to a host computer via USB. Note too that software can be loaded into the Uno board when it is inside a fully assembled Controller, but you can also connect a bare Uno board right out of the shipping package to your computer (using USB) and upload the software to it. The Uno stores its software in flash memory. Flash memory is non-volatile, meaning that the content of the memory is preserved even when power is removed. Once software is loaded into the Uno, it need not be loaded again, unless you want to upload new or revised software.

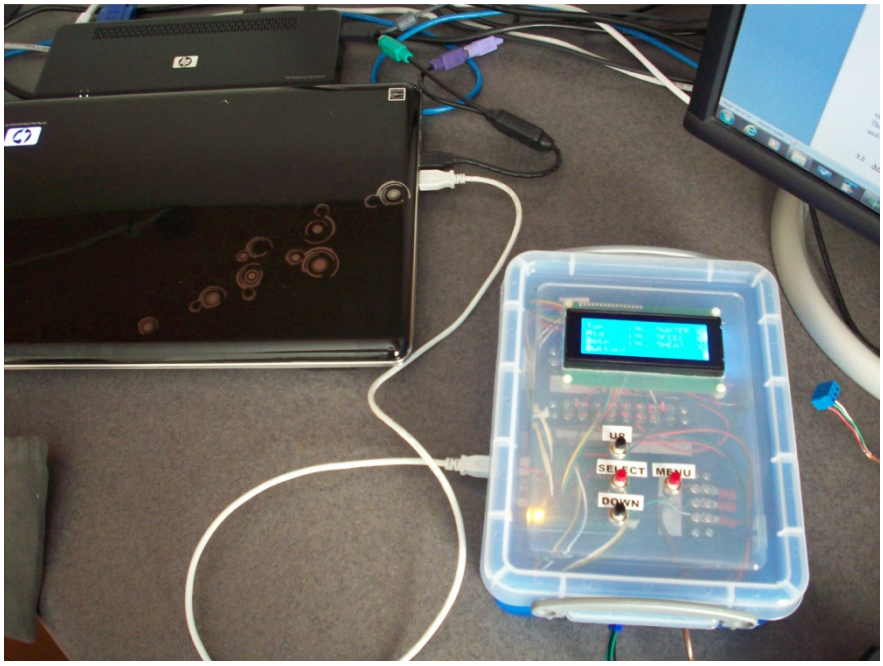


Figure 3-1. Controller connected to PC via USB.

3.2 Arduino Development Software Installation.

You download the Arduino development environment from the Arduino website: http://arduino.cc/en/Main/Software. When on this web page, scroll down to "Previous IDE Versions" and locate "Arduino 0022". Click on the operating system for your PC: Windows, Mac OS X, Linux (32 bit), or Linux (64 bit). This will start a download of a compressed archive

containing all of the Arduino software and libraries. Follow the directions to uncompress the software and install it on your PC (directions will differ somewhat depending upon the operating system of your PC). In so doing, note the directory (folder) where the Arduino software is located. You will need this information later to add the custom libraries to the Arduino development environment. You will find detailed instructions for this step at: http://arduino.cc/en/Guide. Click on your operating system for detailed instructions and follow them carefully.

> ***Note**: The Controller software (both Test Software and Curbie Open Source Still control software) provided with this project has been developed under Arduino development environment 0022, which is not the latest Arduino release package. You must take care to use the version 0022, as this is the only version that has been tested and is known to be working. The new release version "Arduino 1.0", specifically, is known NOT to work with the libraries that you will install in the next section. These libraries (some of which come from third party vendors) will need to be updated for Arduino 1.0 compatibility and such updates are not available at this time.*

During installation of the Arduino development environment, you may be prompted to select the type of Arduino board that you are using. Your Arduino board is the "Uno" (release 3, if they ask for this). This will ensure that the correct device driver is installed on your PC so that it can communicate with the Uno board. You may also be prompted to select the serial port (virtual COM port for Windows) to use for the Uno board that is plugged into a USB port on your computer, per the installation instructions. (The default for Windows will be the next unused COM port number).

> ***Note**: If you are using a PC with the Windows operating system, the Arduino installation instructions specifically say that the driver installation for the Uno will fail the first time. This is normal. Keep following the instructions. If you have an Uno R2 board, the installation will succeed as per the instructions. If you have an Uno R3 board and you get an "unknown device" error when you subsequently try and plug your Uno board into a USB port on your computer, you will need to copy a newer version of the driver into your development environment. The instructions for doing this are provided below.*

Special driver installation instructions for Windows PCs: The latest version of the Arduino Uno board is Revision 3 (REV3 or R3). In some instances, Windows cannot associate the Uno driver supplied with the Arduino IDE version 0022 release with the Uno R3 board. If, after following the installation instructions carefully, you still get an "unknown device" error from Windows when you plug your Uno board into a USB port on your Windows PC, you will need to install the latest Uno driver into the Arduino Development Environment, as described below. This

issue has been reported on some Windows computers only. Macintosh and Linux operating systems do not need a special driver installation.

If you have a Windows computer and cannot get your Uno R3 board to communicate with your Windows PC, use the "Arduino_UNO_REV3.inf" driver file that is supplied in the zip archive for the "Open Source Controller Release Package" that this manual was contained in. Go to the location on your Windows PC where you downloaded this zip archive to, open the archive by double clicking on its icon and you will see a regular folder with the same name. Double click on the folder to open it and look for the "Drivers" folder. Double click on the Drivers folder and you will see the file: "Arduino_UNO_REV3.inf". Copy this file to the clipboard (Ctrl-C, or Edit| Copy or Organize|Copy, depending upon your Windows version). Next, go the location on your PC where you downloaded the "arduino-0022" package. Double click to open the compressed folder, double click on the regular folder by the same name to open it, and then locate the "drivers" folder there. Double click to open the "drivers" folder and paste the "Arduino_UNO_REV3.inf" file into the "drivers" folder (Ctrl-V, Edit| Paste or Organize|Paste, depending upon your Windows version). Try connecting your Uno again, following the same instructions for Arduino installation as failed before. Now Windows should find the new driver and installation is complete.

<u>All users resume here</u>:

After installation is complete, you will have a shortcut to the Arduino software on your desktop. It will look something like figure 3-2, below:

Figure 3-2. Arduino Software shortcut on the desktop.

If you click (double click) on this shortcut, it will launch the Arduino development environment, showing something similar to figure 3-3.

Figure 3-3. Arduino development environment opening screen.

If you see the update dialog as in figure 3-3 above, click “No”.

One more thing that you ought to do is to locate your “sketch folder”, as shown in figure 3-4. In Arduino-speak, programs that you write are called “sketches” and they are kept in your sketch folder. You will be copying the Test Software and the Curbie Open Source Still software into your sketch folder, so you should know where your sketch folder is.

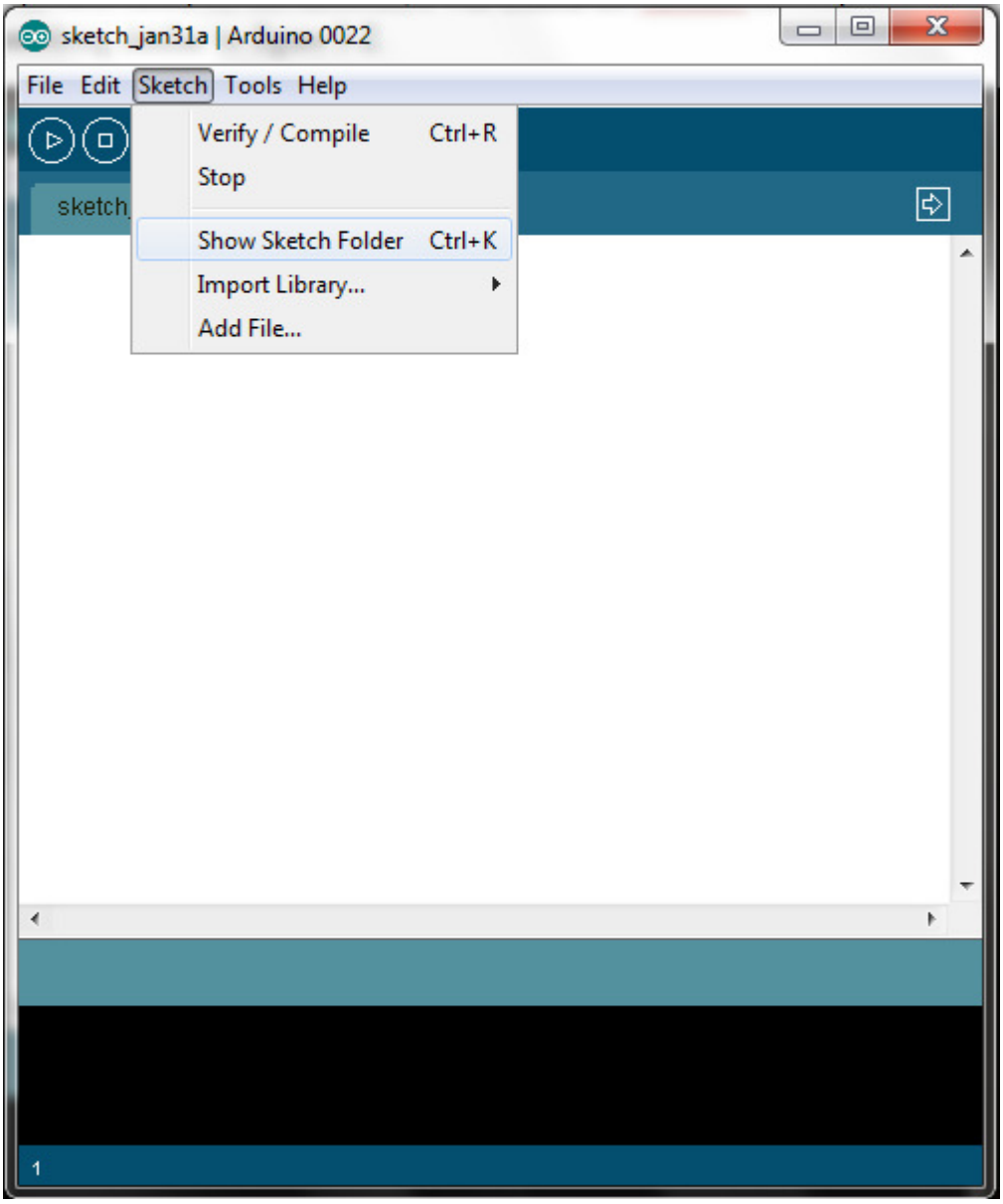


Figure 3-4. Locating the Sketch Folder.

Move your mouse cursor over the "Sketch" toolbar heading as shown in figure 3-4. Click, then move the mouse cursor down to "Show Sketch Folder Ctrl+K" and click. This will open the sketch folder. Make a note of the path to your sketch folder on the window that opens.

Note: *During the Arduino development environment installation, you have the option to put your sketchbook folder anywhere you that want. Just make a note of where you place it, or use this method to locate it later. You will need to load the Test Software and the Curbie Open Source Still software into your sketch folder in order to Compile/Verify and Upload it to your Controller's Uno board.*

3.3 Software Library Installation.

This section contains instructions to load additional "libraries" to your Arduino development environment. These libraries contain software that allows the Controller's Electronic Module to communicate with the LCD display module, the one-wire bus and temperature probe chips. The Test Software and the Curbie Open Source Still software will not compile correctly unless these libraries are added to your arduino-0022 development environment that you installed in the previous section.

The first step is to locate your uncompressed arduino-0022 development environment folder. In the previous section you were instructed to note where this software was installed. If you didn't do this, you will need to find a folder called "arduino-0022" on your PC. On a Windows PC, it is usually installed in a "Program Files" or "Program Files (x86)" folder on your C: drive by default, but you had the option to install it anywhere you wanted to (I installed mine in a "Bob" folder – see screenshot figure 3-5).

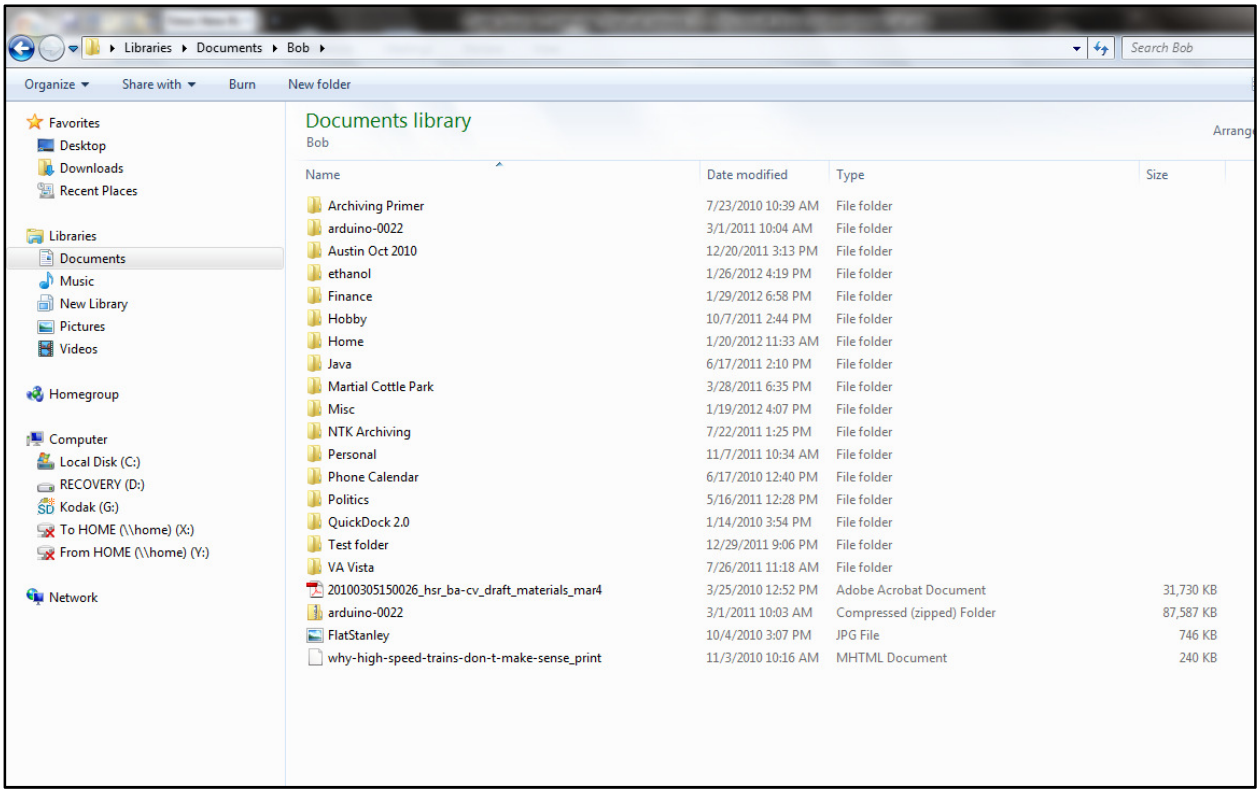


Figure 3-5. Folder containing the Arduino 0022 Folder.

Once you have located your arduino-0022 folder, double click to open it. You will see something similar to figure 3-6.

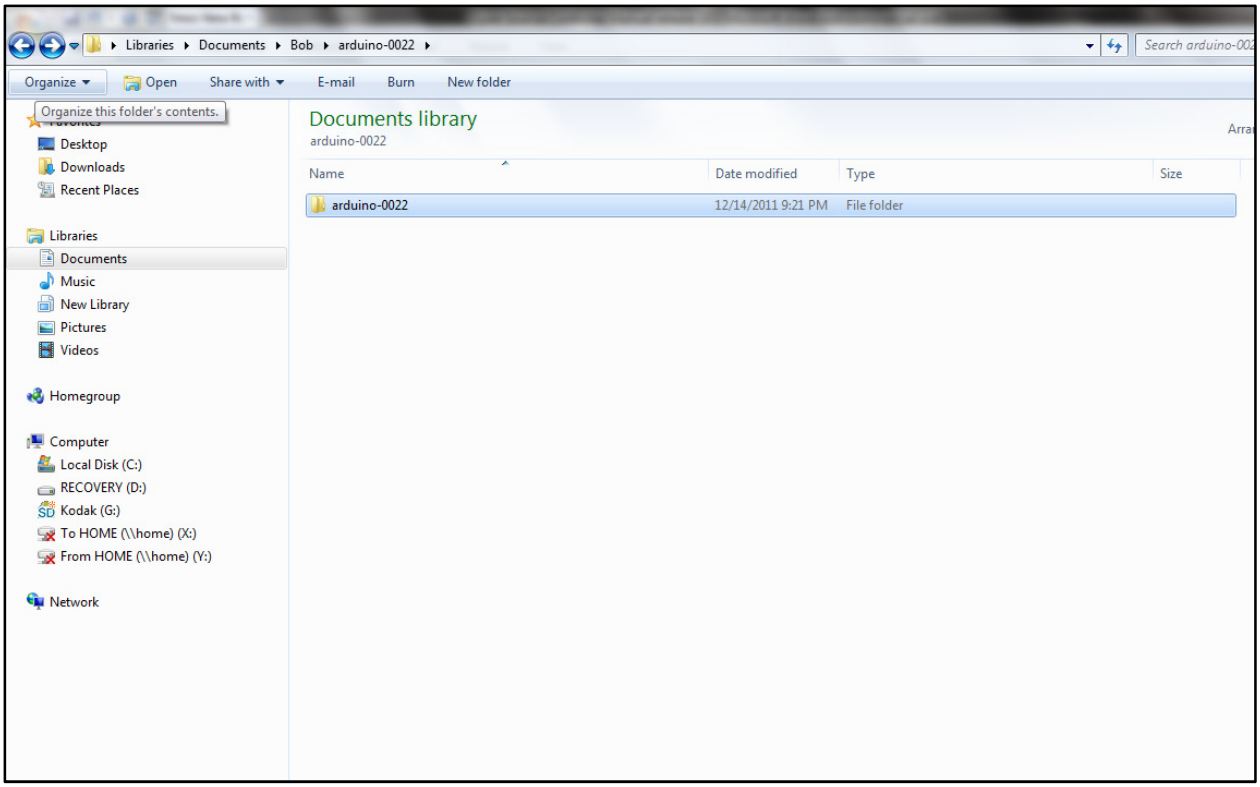


Figure 3-6. Opening the Arduino 0022 Folder.

Note from figure 3-6 that the arduino-0022 folder may contain another folder called arduino-0022. So proceed to double click on this folder to open it to show something similar to figure 3-7:

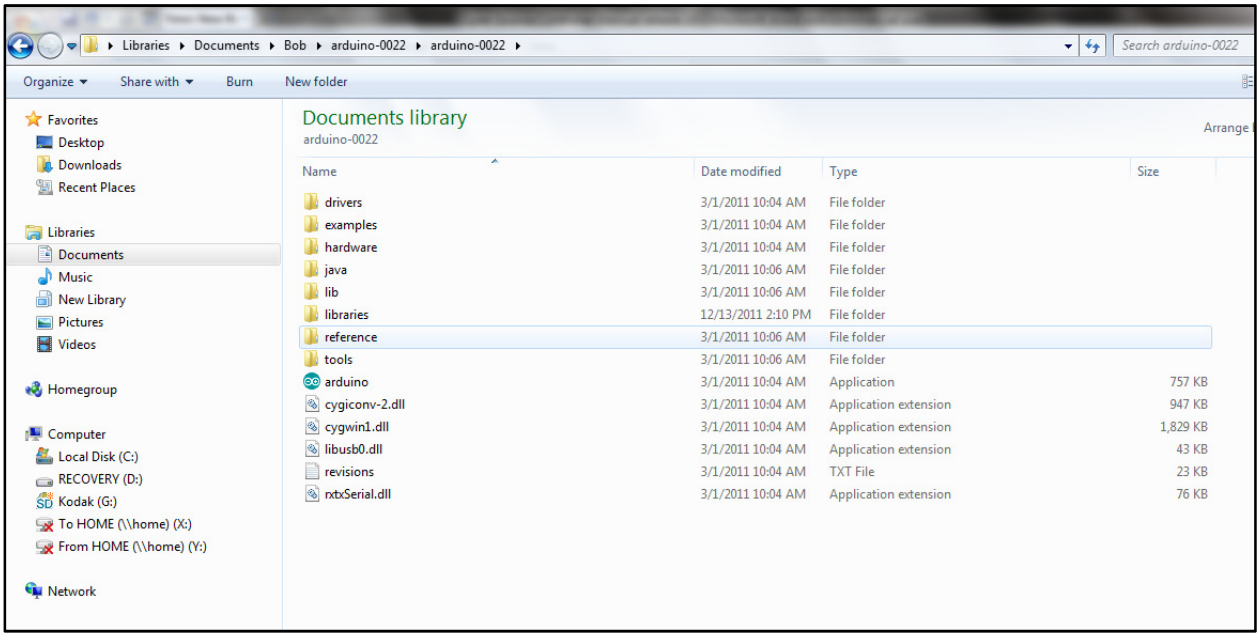


Figure 3-7. Contents of the Arduino 0022 Folder.

Note that there is a folder called “libraries” in the arduino-0022 folder. Now double click to open the libraries folder and you will see a list of the libraries similar to figure 3-8.

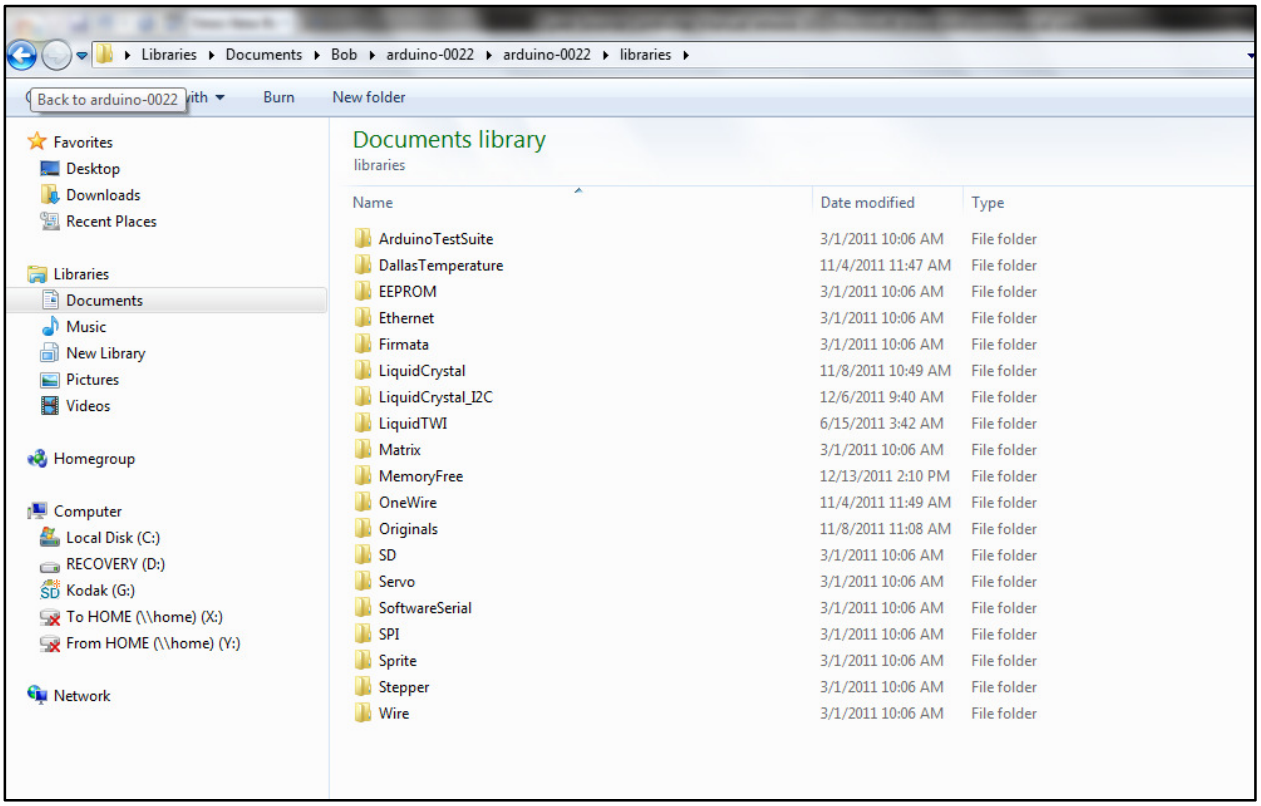


Figure 3-8. Contents of the Libraries Folder.

At this point, you should have a list of library folders similar to folders shown in figure 3-8 EXCEPT that the following libraries (folders) will NOT be present on your system:

- Dallas Temperature
- LiquidCrystal_I2C
- LiquidTWI
- OneWire

These are the libraries that you must install. You install the libraries as follows:

- Download the Controller Software archive folder from the website, if you have not already done so. Download this archive to a temporary folder on your PC, and note where this temporary folder is.
- Uncompress the Controller Software archive folder. On a Windows PC, you only need to double click on the compressed folder icon to do so. You will see an uncompressed

folder icon with the same name. Double click on this folder and find a folder called "Libraries". Double click to open it.

- You will see these folders on your PC screen:
  - Dallas Temperature
  - LiquidCrystal_I2C
  - LiquidTWI
  - OneWire
- Copy and paste each of these folders from this temporary archive folder to the "libraries" folder that you displayed in figure 3-8. Note: do not open these library folders! Just copy them intact to the arduino-0022 "libraries" folder so that your display is similar to figure 3-8. Make sure that you have copied all four of these library folders to your arduino-0022 "libraries" folder by checking the updated contents per figure 3-8. If the PC display shows that these library folders have been added, you have successfully completed this step.

3.4 Controller Test Software.

You are almost done! The next thing that you need to do is to copy the Test Software into your sketchbook. The Test Software is software that is supplied with the Controller for the purposes of testing the completed Controller hardware and software installation.

Going back to the temporary folder on your PC where you previously downloaded the Controller Software archive folder from the website, open this folder and find the folder called "Software". Double click to open this folder and find the folder called: "still_control_test_11". Copy and paste this folder into your Sketchbook folder.

> ***Note**: You copy and paste the folder. Do not open the folder and copy/paste the file inside the folder. The Arduino development environment wants to know where the folder is; it will find the relevant program file to open when you Compile/Verify.*

Recall from section 3.2 (figure 3-4) that you were instructed to note the location of your Sketchbook folder during the Arduino development environment installation. Don't worry if you didn't do so at that time; go back now to section 3.2 and follow the instructions to find out where your Sketchbook folder is.

Next, re-open the Arduino development environment by double clicking on the shortcut that is on your desktop, as shown previously in figures 3-2 and 3-3. Click "No" if you get the upgrade notice as shown in figure 3-3.

Now click on "File" in the menu, and then move the mouse cursor to "Sketchbook", as depicted in figure 3-9. Note: you won't have all of the files that I have. It is only important that your sketchbook contains the entry "still_control_test_11". Scroll down to this listing, and then click

the mouse to select it. Another Arduino development environment window will open with lots of code displayed in it, similar to figure 3-10.

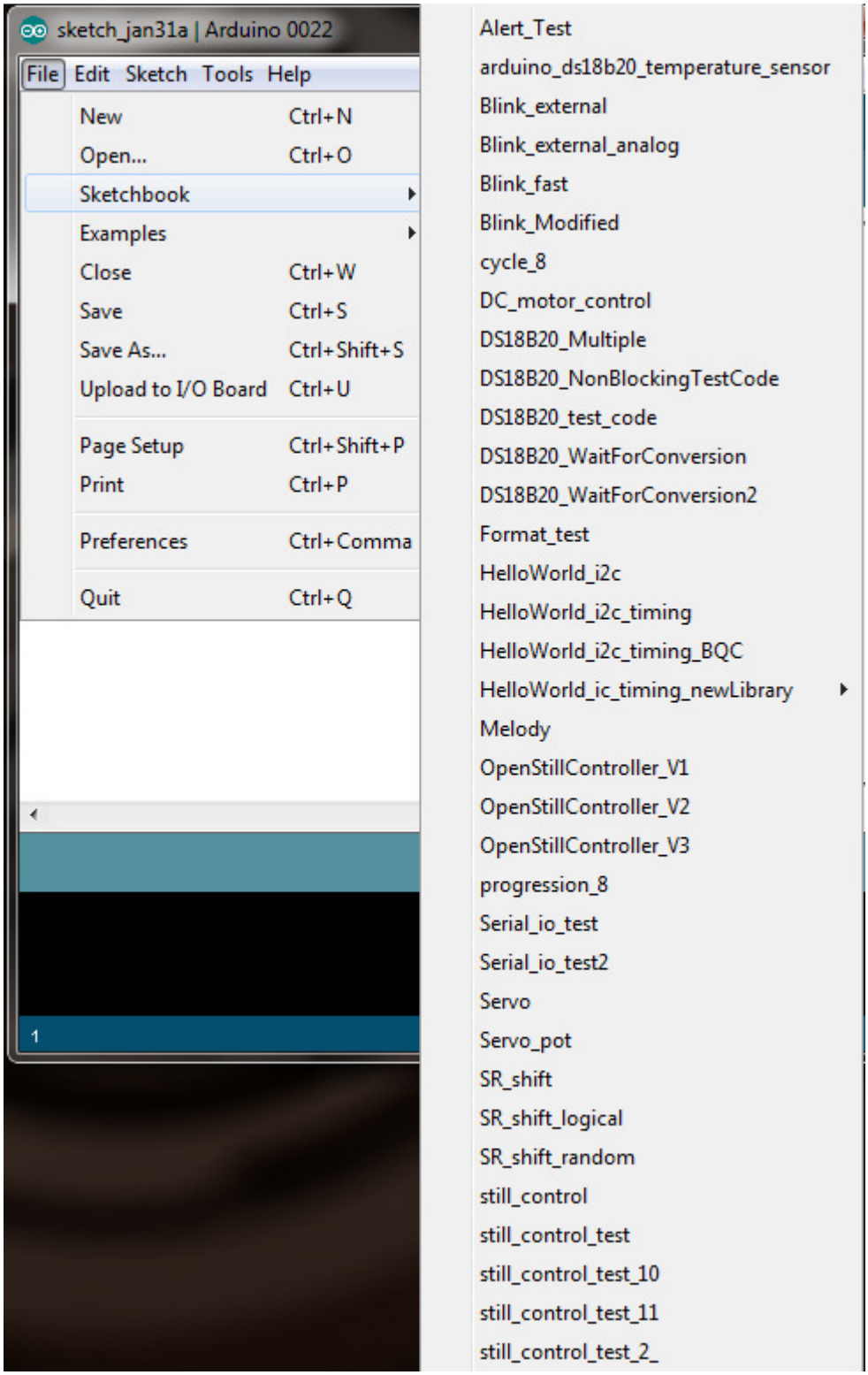


Figure 3-9. Opening the Sketchbook.

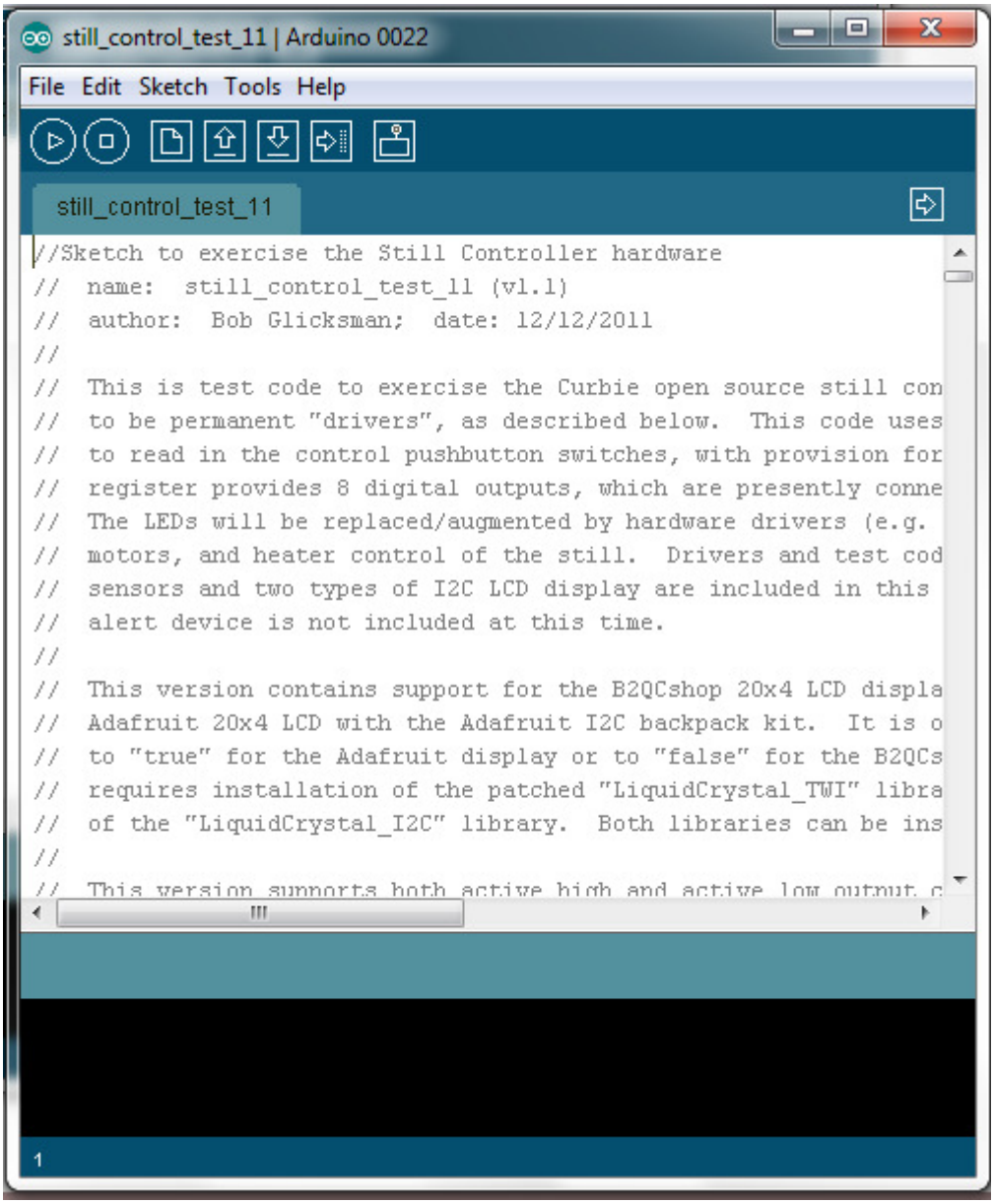


Figure 3-10. Test Software Window.

Almost there! Now, click your mouse on "Sketch" on the menu, then on "Verify / Compile Ctrl+R", as depicted in figure 3-11. Look down at the blue bar over the black area near the bottom of the window. I should say "Compiling …". The program may take 10 seconds or more to compile, then it should show "Done compiling." in the blue bar with a file size dialog in the black area, as shown in figure 3-12.

> ***Note**: if you get a bunch of red error messages in the black area, you have not installed everything correctly. Go back and double check that you have followed the instructions precisely. If you can't find and correct your error, please DO NOT HESITATE to ask for help via a posting on the Yahoo Alcohol Fuel chat group!*

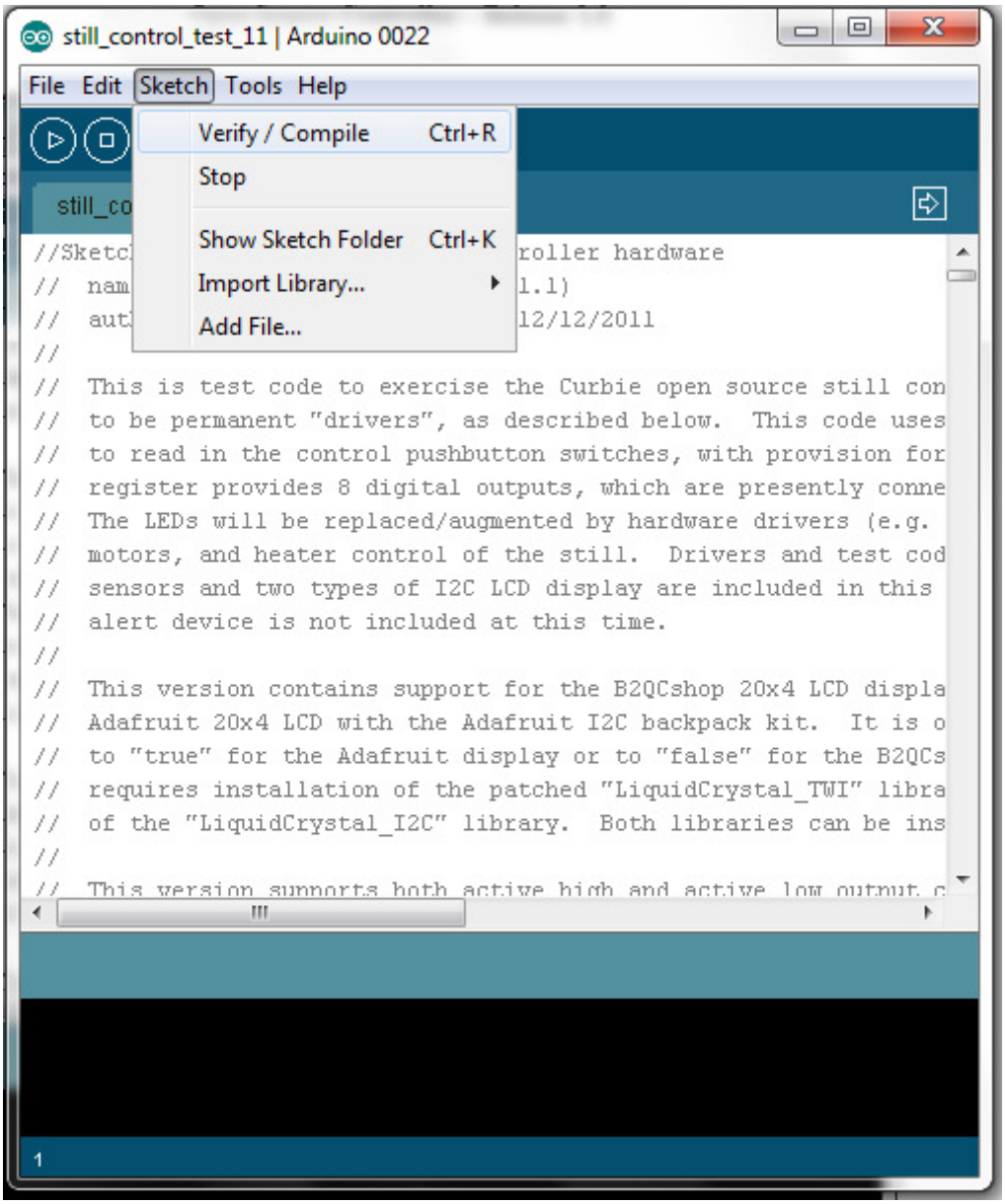


Figure 3-11. Selecting “Verify/Compile”.

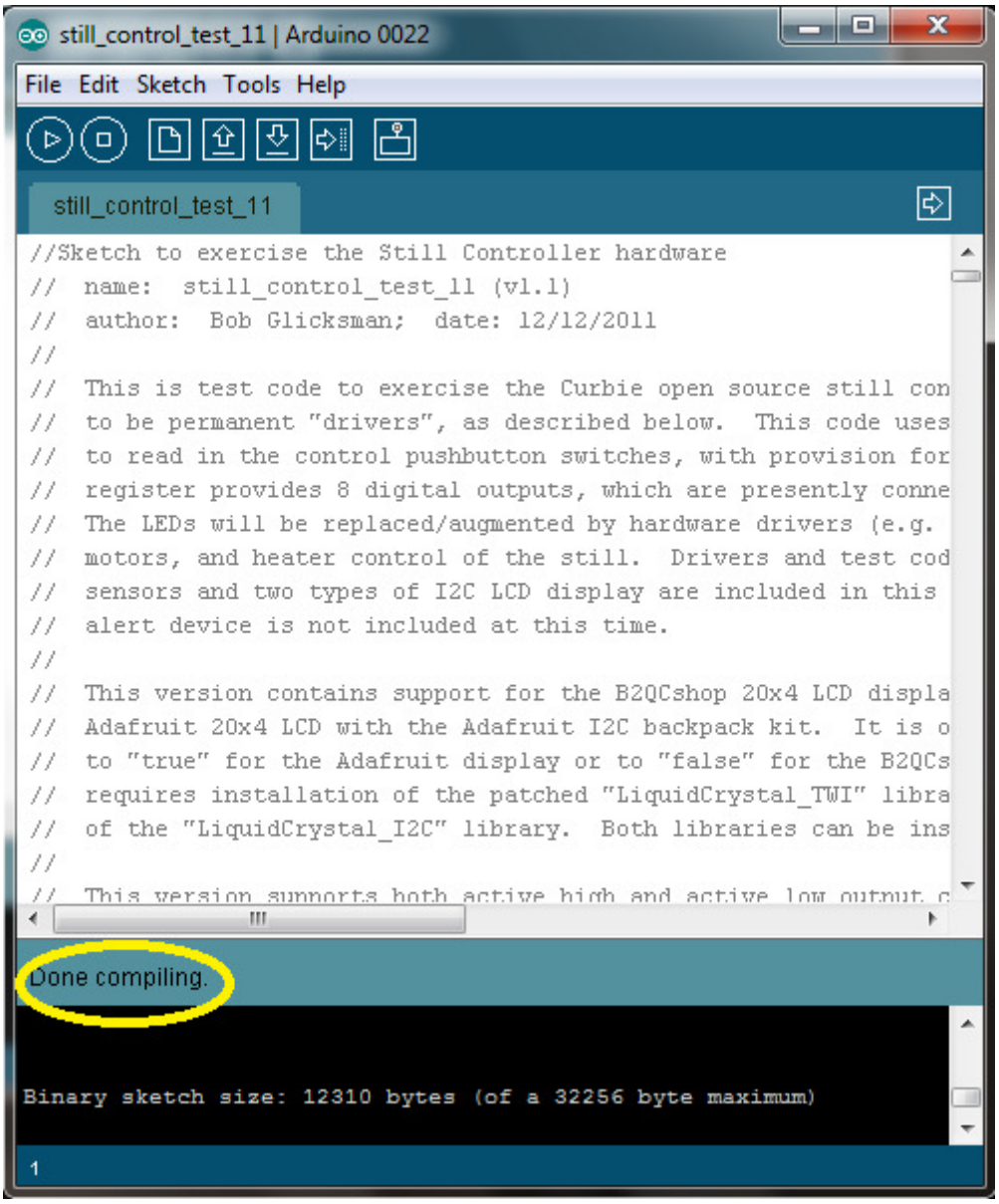


Figure 3-12. Successful Compilation.

If you have gotten to a screen similar to figure 3-12 without any compilation error messages, you can now relax. The hard part is over!

3.5 Uploading Software to the Controller.

Software is uploaded from your PC to the Arduino Uno board via a USB cable (A/B cable). A suitable cable is included in the parts list – see Appendix A. It does not matter if your Uno board is mounted in a completed controller or just taken out of its original package. Either way, plug the USB cable into a USB port on your computer and the other end into the USB connector on the Uno. Your PC should indicate a correct USB connection. If it prompts you for a driver, then you did not configure the "COM" port on your PC correctly during the Arduino development

environment installation. If this is the case, return to the instructions in section 3.2. Note the special instructions for Windows users in section 3.2. You may need to install a newer driver. See the instructions in section 3.2 for details on installing the driver that is included in your Open Source Controller release package (the zip archive that you extracted this manual from).

On a Windows computer, you can check the "COM" port assignment by clicking on "Tools" and then placing the mouse over the drop down menu "Serial Port" to see which COM port is selected for the upload, as depicted in figure 3-13.

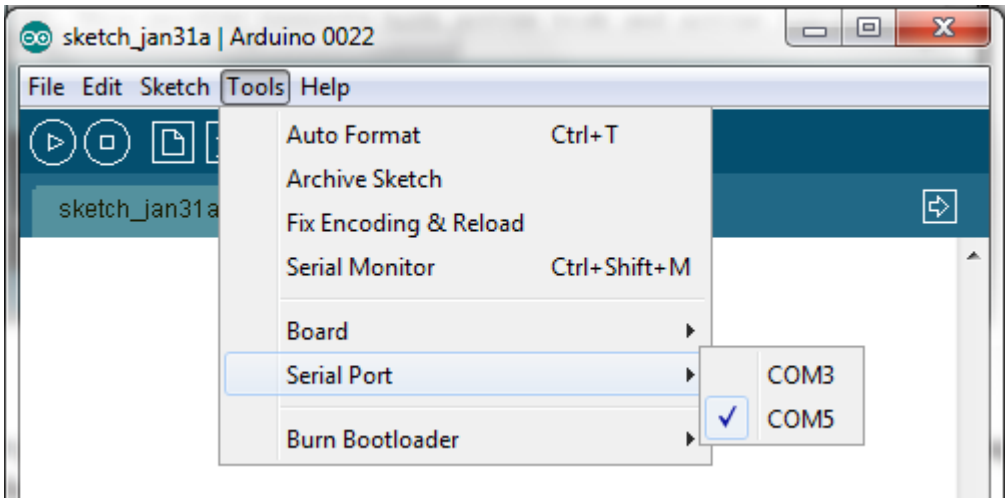


Figure 3-13. Checking the Uno COM Port Assignment.

***Note**: Windows PCs use a "virtual COM port" over the USB serial port to communicate with the Arduino Uno board. Every Uno board that you have will require its own COM port assignment. Non-Windows PCs will have their own way of designating the driver for the serial connection over the USB cable. Refer to the documentation for Arduino installation for your operating system on the Arduino website.*

Normally, there will be no issue and your PC will recognize the Uno board as soon as the USB cable is connected at both ends. Note that the Uno can be powered by the USB-supplied power, so it is only necessary to plug in the USB cable to both power and establish a serial connection to the Uno board.

Using the Arduino development environment window, after successfully compiling the Test program, click on the upload button as shown in figure 3-14.

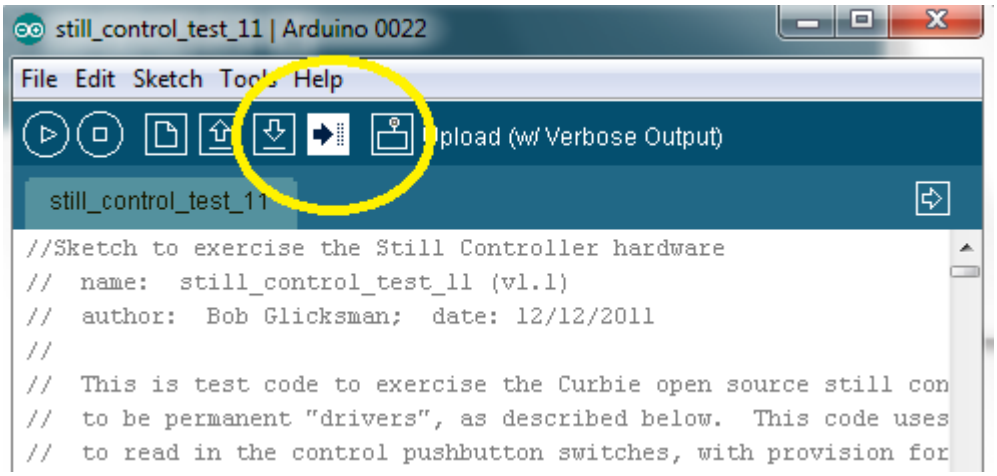


Figure 3-14. Selecting the Upload Feature.

The blue bar toward the bottom of the Arduino development environment window will say "Uploading to I/O Board …" until the transfer of the software is complete. Note: this may take 20 seconds or more; be patient. When the upload is complete, you will see this as indicated in figure 3-15.

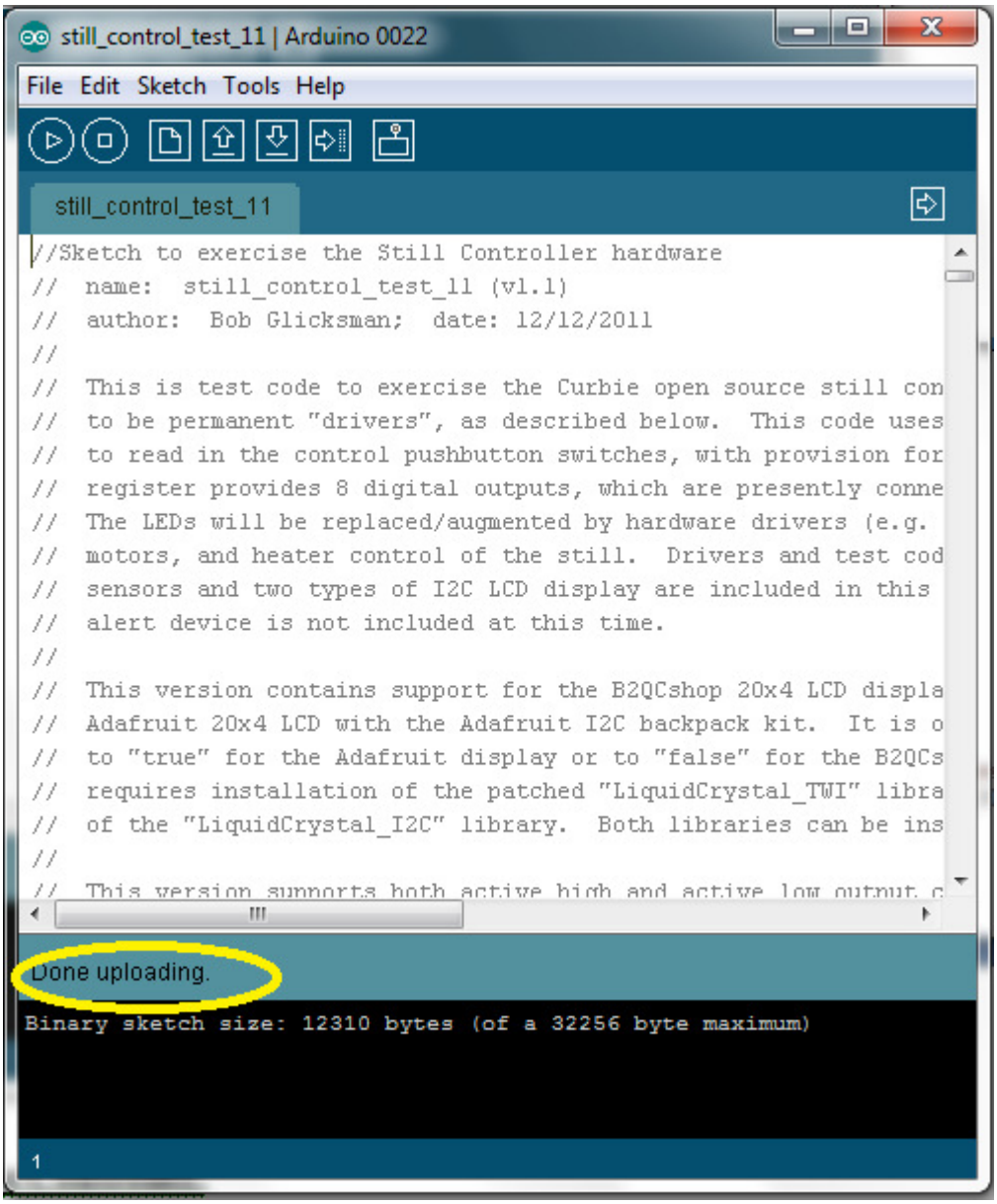


Figure 3-15. Upload Complete.

Completion of uploading also resets the Uno board and begins execution of the program. If your Uno board is installed in a completed Controller, the Test Program will be running and you can use it to test out the Controller as described in section 3.6, below.

If you encounter an error or problem in uploading the software, go back over this chapter to ensure that you have installed everything correctly and did not skip any steps. There is also helpful troubleshooting information on the Arduino website at: http://arduino.cc/en/Guide/Troubleshooting#toc1.

3.6 Testing Using the Controller Test Software.

This section assumes that you have:

- Completed assembling the Shield Board using the instructions in Chapter 5 of this manual.
- Completed the Controller using the instructions in Chapter 4 of this manual, including installing the Shield Board on top of the Arduino Uno board.
- Assembled and connected a "Test Board" breadboard, as described in Section 4.4 of this manual.
- Completed installing the software and uploading the Controller Test Program to the Uno board in the Controller according to the instructions in sections 3.1 through 3.5 of this manual.

If you have performed all of these steps correctly, the LCD display should be lit up and displaying the test screen, as shown in figure 3-16.

Figure 3-16. Test Software Power-Up Display

If you don't see any display, turn the little pot on the back of the display – it controls the contrast and the contrast may have been set too low from the vendor. If you see a display that looks like two rows of little rectangles and nothing readable, open the Controller box and press the "Reset" button on the Shield Board. This should reset the display after a cold power up.

Note from figure 3-16 that you should have three temperature readings and they should all be close to each other, as the temperature sensor chips on the Test Board are right next to each other. The temperature reading won't be exactly what is shown in figure 3-16; they should read out your ambient room temperature. You can put your finger on top of one of the temperature sensor chips and see one of the temperature readings increase.

> ***Note**: There is no relationship between the position of the temperature sensor chips on the Test Board and which one reads out as "Top", "Mid", and "Botm" in the LCD Display. The order in which the chips read out is a function of their internal unique identifier number. Instructions for installation of temperature probes in the Curbie Open Source Still provide a means to identify which probe goes on the top, the middle and the bottom of the still column prior to mounting the probes.*

Other things to note about the initial LCD display for the Test Software are as follows:

- There is no button displayed for "Button:"
- There is an "X" in the last column by the words "WATER", "FEED", and "HEAT".
- No LEDs are lit on the Test Board.
- The alert is not sounding (it may sound briefly during power up, and then stop when the display of figure 3-16 shows).

In order to fully test the Controller, each pushbutton is pressed in turn and the following observations are made on the Test Board and the LCD:

- Press "MENU". Observe:
  - The "Button:" line displays [MENU]
  - The alert sounds
  - LED 1 lights up.
  - There is an "X" in the last column by the words "WATER", "FEED", and "HEAT".
- Press "MENU" again. Observe:
  - The "Button:" line still displays [MENU]
  - The alert silences (thankfully!)
  - LED 1 turns off (all LEDs off).
  - There is an "X" in the last column by the words "WATER", "FEED", and "HEAT".
- Press "DOWN". Observe:
  - The "Button:" line displays [DOWN]
  - The alert does not sound
  - LEDs 4 and 5 light up.

    - There is an "X" in the last column by the words "WATER", "FEED", but "HEAT" has a "spinning" figure in the last column.
- Press "DOWN" again. Observe:
    - The "Button:" line still displays [DOWN]
    - The alert does not sound
    - LEDs 4 and 5 go out (all LEDs off).
    - There is an "X" in the last column by the words "WATER", "FEED", and "HEAT".
- Press "SELECT". Observe:
    - The "Button:" line displays [CENTER]
    - The alert does not sound
    - LEDs 2 and 7 light up.
    - There is an "X" in the last column by the words "WATER", "HEAT", but "FEED" has a "spinning" figure in the last column.
- Press "SELECT" again. Observe:
    - The "Button:" line still displays [CENTER]
    - The alert does not sound
    - LEDs 2 and 7 go out (all LEDs off).
    - There is an "X" in the last column by the words "WATER", "FEED", and "HEAT".
- Press "UP". Observe:
    - The "Button:" line displays [UP]
    - The alert does not sound
    - LEDs 3 and 6 light up.
    - There is an "X" in the last column by the words "FEED", "HEAT", but "WATER" has a "spinning" figure in the last column.
- Press "UP" again. Observe:
    - The "Button:" line still displays [UP]
    - The alert does not sound
    - LEDs 3 and 6 go out (all LEDs off).
    - There is an "X" in the last column by the words "WATER", "FEED", and "HEAT".

This completes the testing of all of the functionality of the Controller. In general, the pushbuttons cause the following actions to occur:

- When a button is pressed, its name is displayed on the LCD in square brackets next to "Button:". The name persists as many times as the button is pressed and only changes when another button is pressed or the power is cycled.
- When a button is pressed, a combination of LEDs/Alert is activated. The buttons act as a toggle on these – first press turns them on, next press turns them off:

- MENU: LED 1 and Alert tone toggles, no “spinner” in the last column of the LCD.
- DOWN: LEDs 4 and 5 and a “spinner” next to HEAT toggles.
- SELECT: LEDs 2 and 7 and a “spinner” next to FEED toggles.
- UP: LEDs 3 and 6 and a “spinner” next to WATER toggles.

The buttons may be depressed in any combination and will act according to these rules.

4 Building the Controller.

This chapter contains instructions for building the Controller from parts in the parts list, Appendix A. You will need an assembled Shield Board. Instructions for assembling the Shield Board are in Chapter 5 of this manual.

4.1 Controller Overview, Major Components and Tools.

Figure 4-1 shows the major components that need to be mounted and integrated to make the Controller.

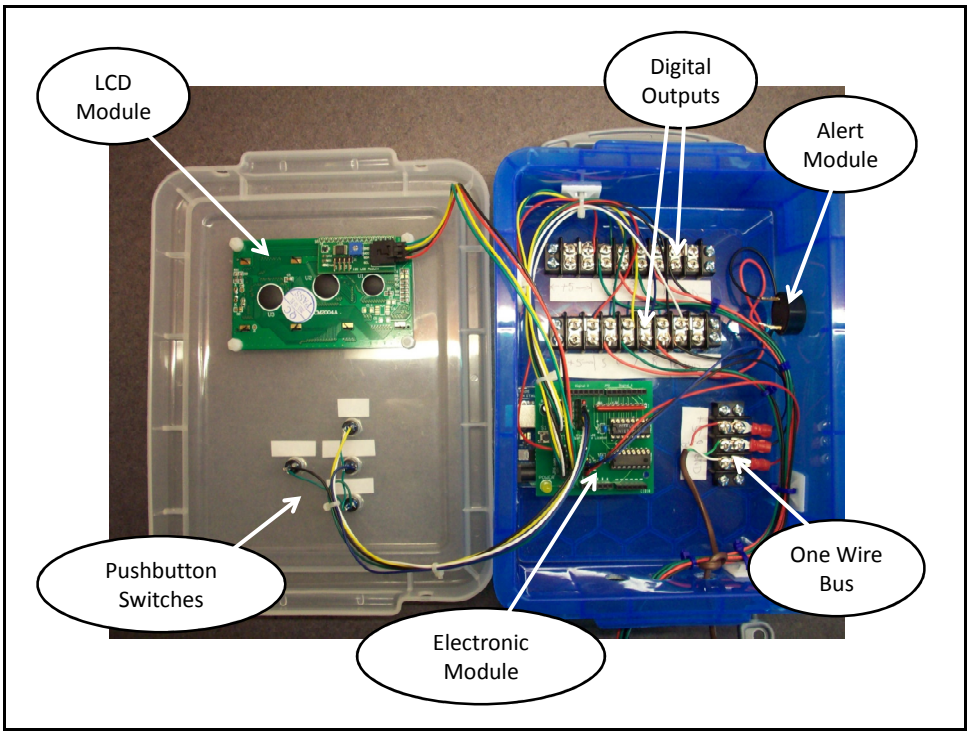


Figure 4-1. Major Components of the Controller.

The major components are:

- **Controller Power Supply** (not shown). http://www.adafruit.com/products/63
- **Controller box**. http://www.officedepot.com/a/products/452333/Really-Useful-Boxes-Plastic-Storage-Box/#firstTab
- **LCD Module**. http://b2cqshop.com/products/192-arduino-i2c-serial-2004-lcd-module-20x4-iic-twi-lcd.aspx
- **LCD Module Connecting Cable**. http://b2cqshop.com/products/193-2-pcs-cables-for-iic-serial-lcd-module-i2c-for-arduino-diy.aspx

- **Pushbutton Switches, red** (2 ea).
  http://www.taydaelectronics.com/electromechanical/switches-key-pad/push-button/pb-11d02-push-button-panel-mount-spst-no-pb-11d02-th1r00.html

- **Pushbutton Switches, black** (2 ea).
  http://www.taydaelectronics.com/electromechanical/switches-key-pad/push-button/pb-11d02-push-button-panel-mount-spst-no-pb-11d02-th1-00.html

- **Audible Alert Module**. http://www.taydaelectronics.com/piezo-electronic-tone-buzzer-alarm-1-5-28v-pcb.html

- **Electronic Module**:
  - Arduino Uno microcontroller board.
    http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_2151486_-1
  - Shield Board (piggyback on the Uno). Chapter 5

- **Eight connection terminal block** (2 ea) for Digital Outputs.
  http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=231036&catalogId=10001&freeText=231036&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView

- **Three connection terminal block** for One-Wire Bus.
  http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=230981&catalogId=10001&freeText=230981&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView

- **Six conductor interconnect cables**. http://b2cqshop.com/products/170-6-pcs-6-pin-wire-jumpers-female-to-female-for-arduino.aspx

The parts list (Appendix A) also contains nylon and metal mounting hardware, cable ties and other basic hardware needed to construct the Controller.

A summary of the steps needed to construct the Controller is:

- Make cutouts and holes for the LCD and pushbutton switches on the box cover and mount and wire these components.
- Make cutouts and holes in the box body for the Electronic Module, terminal blocks, and audible alert module and mount and wire these components.
- (optional): construct a Test Board to test the completed Controller and wire it to the Controller.

These steps are detailed in the sections that follow.

***Note**: many of the photographs shown in this chapter use a slightly different box than the one specified on the parts list. The differences are very minor and do not impact the instructions in this chapter.*

The tools and equipment that you will need in order to follow these instructions are basic home woodworking tools and electrical tools, including (but not limited to) the following:

Basic Home/Woodworking Tools:

- Bench vise or similar means to hold material while cutting and drilling.
- Basic screwdrivers (straight and Phillips) and hammer.
- Electric or manual drill (1/4" or larger) and metal and/or wood drill bits.
- Small round and flat metal files.
- Box cutter, X-acto knife, or specialized plastic cutting knife.
- Misc pieces of 2x4 wood scraps with clean face for cutting and drilling support.
- Assorted C-clamps.
- Blue Painters Tape (1 roll).

Basic Electrical Tools:

- Soldering iron with small tip, electrical solder (#22 AWG), solder wick and/or solder pullet (de-soldering tool).
- Diagonal cutters.
- Long nose pliers.
- Magnifying glass.
- Wire stripping tool with pre-made stripping dies for AWG #20 – #26 wire.
- Optional – recommended: inexpensive multi-meter.
- Optional - recommended: electrician's tool and #22 spade lugs (figure 2-4 of chapter 2).

A typical set of basic electrical tools and multi-meter are shown in figure 4-2. It is best that the soldering iron be 25 watts and have a removable tip (e.g. Weller). A soldering stand, as shown in figure 4-2 is also recommended. A fine tip and thin solder is necessary for the electrical work required of this chapter and of chapter 5. Figure 4-3 shows a close-up of a fine soldering tip and fine (#22 AWG) solder. It is also important to have electrical tape and handy to have a small hobby vise for fine work, as well as cable ties and cable tie stick on mounts, as shown in figure 4-4. Lastly, a small magnifying glass is very useful for inspecting detailed work – see example in figure 4-5. If you have never soldered electronics, you can find a good tutorial here: http://www.sparkfun.com/tutorials/213

Figure 4-2. Basic Electrical Tools

Figure 4-3. Fine Tip Soldering Iron and Solder.

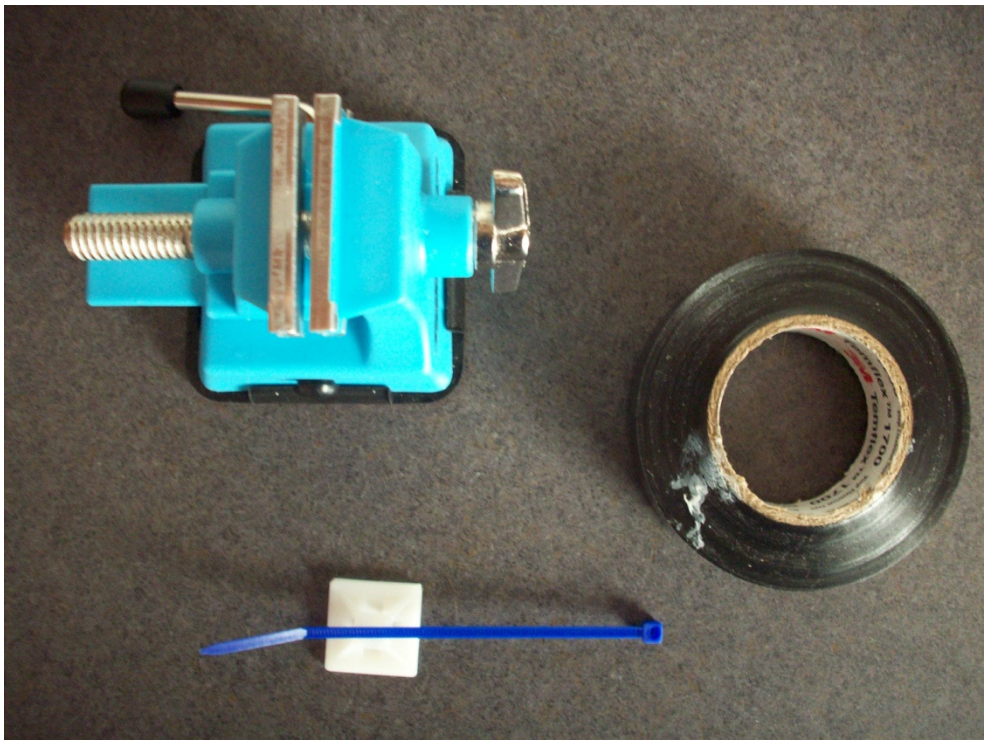


Figure 4-3. Electrical Accessories.

Figure 4-5. Handy Magnifying Glass for Work Inspection.

4.2 Controller Cover Panel.

Figure 4-6 shows the cover of the Controller box as it comes from the vendor. It is necessary to cut out a rectangular hole for the LCD and mounting holes for the LCD and for the pushbutton switches.

Figure 4-6. Blank Controller Box Cover.

4.2.1 Preparing to Mount the LCD.

The first step will be cutting out the rectangular hole for the LCD display.

> ***Note**: Cutting out the rectangular hole for mounting the LCD display will be, far and away, the most difficult and tedious step of this entire process! Have patience, and you will be rewarded something that you can be proud of! You should also refer to Appendix C for an alternative way to make this cutout, courtesy of Thurmond Moore.*

Figure 4-7 contains a template for cutting and drilling the Controller box cover. Figure 4-7 is not to scale. A 1:1 scaled version can be found in the file "ControllerCaseLayout – to scale.pdf", which is part of the release package for the Controller, in the "Hardware" folder.

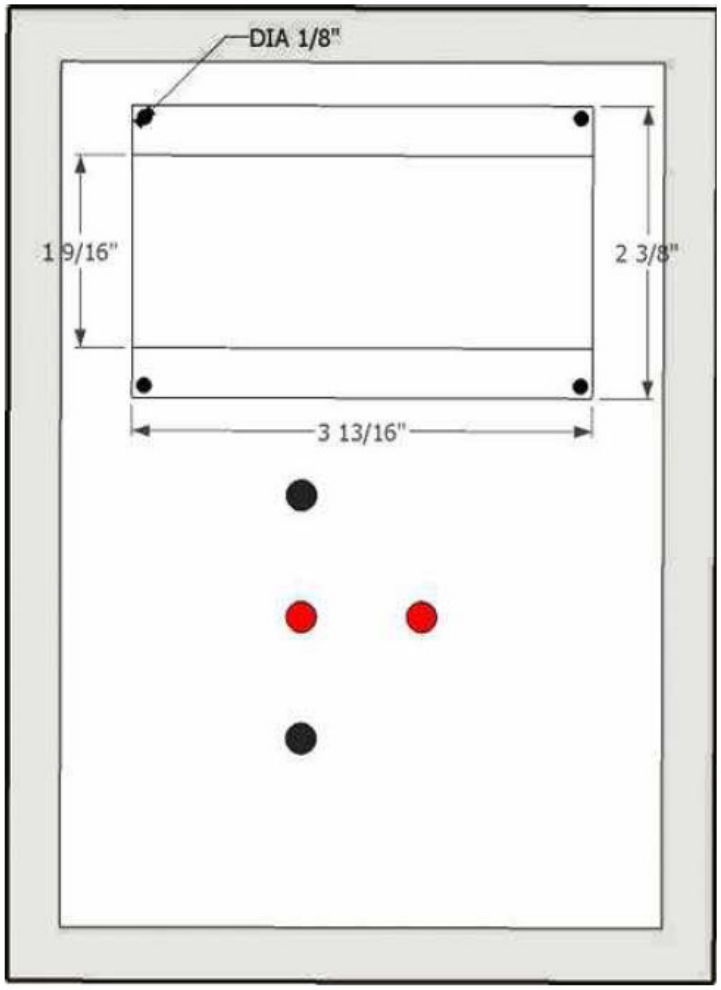


Figure 4-7. Cover Cutting and Drilling Template.

The rectangular cutout (1-9/16” x 3-3/16”) needs to be hand cut with a sharp box knife, X-acto knife or (better still) a specialized plastic cutting knife. In order to get the cut lines and drill hole centers onto the plastic cover, use blue painter’s tape, as depicted in figure 4-8. Blue painter’s tape will adhere very well to the plastic cover but can easily be removed without leaving a residue.

Figure 4-8. Using Blue Painter's Tape to Mark Cut Out Lines.

The cutout lines can be measured based upon figure 4-7, can be carefully transferred from the 1:1 template with carbon paper and a straight edge, or can be marked with a carefully cut out piece of metal or plastic. I used a steel ruler to carefully measure the cutout corners and draw sharp lines for the rectangular cutout.

Once you have marked the rectangular cutout on the painter's tape, cut yourself some clean pieces of 2x4 wood and firmly clamp the box cover down on the wood. The wood is necessary because cutting and drilling the soft, pliable plastic requires that it be adhered to a solid, firm backing. Next, clamp the whole assembly into a bench vise or similar means of holding it in place. Take a straight metal edge and line it carefully up with one of the sides of the rectangular cutout. Cut along the guide from corner to corner with your knife. Finish cutting one edge all the way through before moving the guide for another edge.

***Important Note**: The plastic cover of the box is VERY TOUGH (a virtue for all but this step!). You will need to draw the knife across one edge at least 40 times, with significant pressure on the knife, in order to cut all the way through the plastic. Please have a lot of patience with this step! Take care and take your time and you will rewarded with a very clean and professional looking cutout. Hurry up and you will end up with something that works OK but you will not be proud of how it looks!*

Notes:

- It may help if you drill small holes at each of the 4 corners of the rectangular cutout. These will help prevent your drawing the cutting knife past the corners of the rectangle and scratching the cover surface. Small holes will not appear unsightly when the LCD is mounted through the cutout.
- When cutting the straight lines of the rectangle, you MUST use a very clean, straight metal edge as a guide. I used a steel ruler, but you can use the backside of an old hacksaw blade as an alternative. I held the steel guide edge down with one hand while drawing the cutting knife with the other. Curbie is experimenting with first fashioning a sheet metal rectangle of the exact size and bolting this sheet metal guide down to the plastic and wood underlayment with wood screws to hold it fast in place. This technique should work best, assuming that you have the ability to make the rectangular sheet metal template in the first place.
- Always make knife cuts in the plastic starting at a corner and drawing the knife past the middle of the line, but never as far as the opposite corner. Then reverse the cut from the opposite corner to the middle of the line. This technique helps solve two problems: (1) it prevents the plastic shavings from clogging up the cut, and (2) it prevents you from going past the opposite corner and scratching the plastic outside the rectangular cutout space (very easy to do – be careful).
- If you use an X-acto knife, please take care not to put so much pressure on the knife as to break off the blade. <u>You will be tempted to put a lot of pressure on the knife as the plastic is very tough</u>!

Once you have completed the rectangular cut, you have finished the hardest part of the Controller assembly. Now it is time to mark and drill the four mounting holes for the LCD. The LCD will mount from the back of the box cover with the display part through the rectangular hole that you just cut. However, the mounting holes are symmetrical. Therefore, you can mark the mounting holes by placing the LCD through the top of the cover to mark the holes on the painter's tape.

Unclamp the cover and wood backing that you used to cut the rectangular LCD hole. Place the LCD module face down on the top of the cover with the LCD display protruding through the rectangular cutout and the LCD electronic board flush against the box cover with the painter's tape still in place. Now you can use a pencil to mark the LCD module mounting hole locations on the painters tape. Once you have done this, remove the LCD module, re-clamp the wood backing to the box cover, and clamp the whole assembly back in the bench vise. Now drill a 7/64" diameter hole at each of the four marked locations (1/8" will work as well). As with any other material, use a center punch to indent the center of the hole and use a sharp drill bit to drill with.

Once the mounting holes are drilled, remove the clamps and wood backing and insert the LCD in the cover from the back so that the LCD display pokes all the way through and check the alignment of the mounting holes. Do not mount the LCD in the cover at this time, because you need to drill the holes for the pushbutton switches first.

4.2.2 Preparing to Mount the Pushbutton Switches.

Figure 4-7 depicts the locations for the four pushbutton switches. Refer back to Figure 1-3 for what the finished product will look like. The buttons are in two columns equidistant from the centerline of the box cover. The left column will have three of the buttons, from top to bottom:

- UP (black button).
- SELECT or CENTER (red button).
- DOWN (black button).

The right column has the MENU button (red), directly across from the SELECT button. Of course, you may choose to alter this arrangement in any way that works ergonomically for you.

The button arrangement in figure 4-7 has the left column 1-7/8" from the left edge of the raised portion inside of the box cover, and the right column 1-7/8" from the right left edge of the raised portion of the box cover. The DOWN button is about 1-1/2" up from the bottom edge of the raised portion of the box cover, the SELECT button is about 1" up from the DOWN button and the UP button is about 1" up from the SELECT button. Again, you may alter this arrangement in any way that best suits your needs.

The portion of the box cover where the pushbuttons are to be mounted should be covered with blue painters tape and the pushbutton centers measured and marked with a pencil. Mount the plastic box cover firmly to the 2x4 wood pieces that you used when making the LCD cutout and clamp your assembly in a bench vise. Use a center punch to indent the center of each pushbutton hole and then drill 17/64" holes for each of the pushbuttons. Note: 17/64" is an odd size. If you don't have this drill bit size, you can use ¼", which will be slightly undersized. Then file out the holes until you can just get the screw body of a pushbutton through the hole.

> ***Note**: It is always best to drill a small pilot hole first, and then the 17/64" hole. Use a small drill for the pilot hole, e.g. 1/64".*

Unclamp your work, remove the painter's tape, clean off any plastic shavings carefully with an X-acto knife or file and wipe any residue off of the box cover with a clean rag. You are ready to mount and wire the box cover components.

4.2.3 Mounting and Wiring the LCD.

Place the LCD module through the rectangular cutout in the box cover from the back of the cover so that the LCD electronic board is in back of the cover with the LCD display module poking through the front of the cover. The rightside-up position of the LCD is where the little connector board on the back of the LCD module is at the top (see figure 4-9).

Place a #4-40 x ½" nylon screw through the cover and place a nylon #4 nut on the back. Tighten the nut. I prefer to use double nuts to keep things tightly in place, but you can also use captive nuts or locktight glue if you prefer. You will probably never have to replace the LCD once it is mounted in the cover, unless you mounted it upside down (check the instructions above!).

> ***Note**: It is always best to use nylon hardware when mounting electronic modules, in order to avoid any possibility of short circuiting something with metal screws and nuts. The LCD display module is well designed with decent "keepout" clearances between the mounting holes and any electrical traces on the board. Nevertheless, use nylon mounting hardware to be sure!*

Figure 4-9 shows the LCD mounted in the box cover with the LCD module connecting cable attached. The LCD module connecting cable is: http://b2cqshop.com/products/193-2-pcs-cables-for-iic-serial-lcd-module-i2c-for-arduino-diy.aspx and is also found in the parts list (Appendix A).

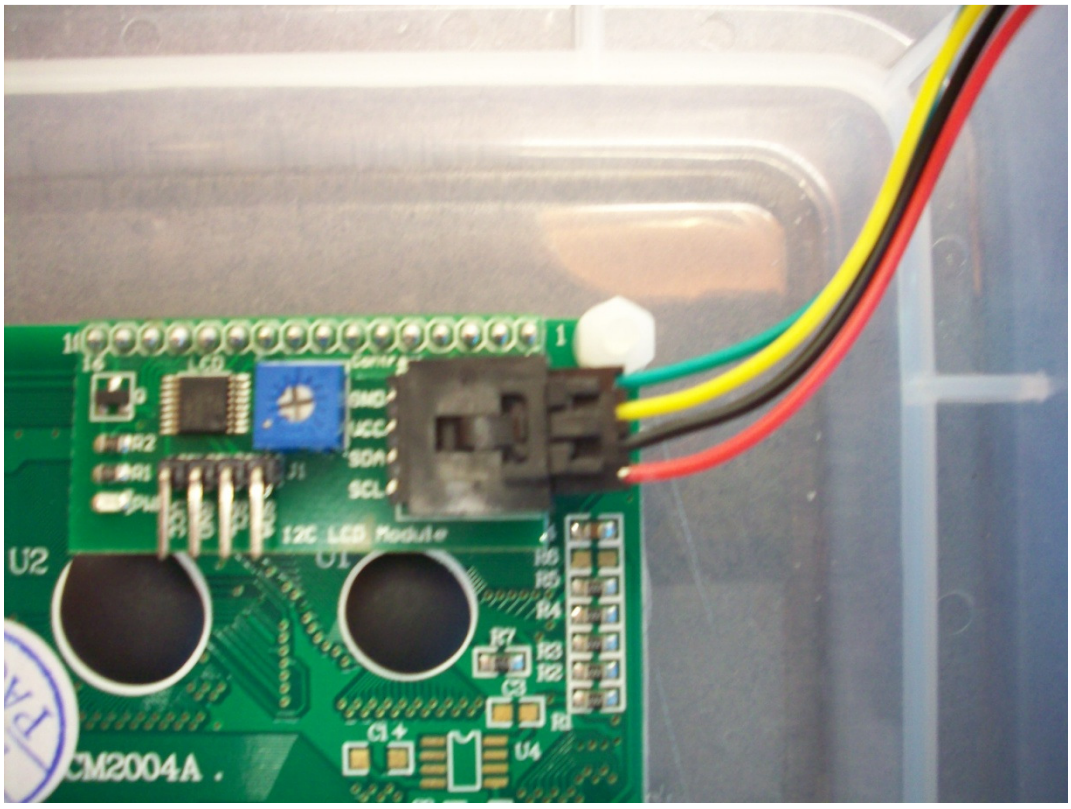


Figure 4-9. LCD Module and Connecting Cable.

Note very carefully which end of the cable has the red wire on the outside pin. This is the end of the cable the plugs into the LCD module plastic connector, as shown in figure 4-9. The cable is not 1:1 – the other end of the cable interchanges the red and black wires so that the black wire is on the outside. **It is very important to plug the right end of the cable into the right place**! The LCD cable is polarized (only plugs in one way) and snaps into place when fully inserted. If you need to remove the cable from the LCD module at any time, you need to depress the plastic tab in the center of the cable and then gently pull on the cable connector body to remove it from the mating connector on the LCD module.

4.2.4 Mounting and Wiring the Pushbutton Switches.

Remove the screw and washer from the threaded portion of each of 4 pushbutton switches (2 red, 2 black). Push the threaded end of each pushbutton through the hole in the box cover for that button and fasten with the washer and nut on the cover top. Once all four pushbutton switches are mounted in the cover, you are ready to wire them up. You will need your soldering iron (heated up), solder, #28 AWG solid wire, wire stripper, diagonal cutter and long nose pliers to wire up the pushbuttons. You will also need one 6-conductor interconnect cable: http://b2cqshop.com/products/170-6-pcs-6-pin-wire-jumpers-female-to-female-for-arduino.aspx.

Figure 4-10 is a schematic diagram of the wiring of the pushbuttons.

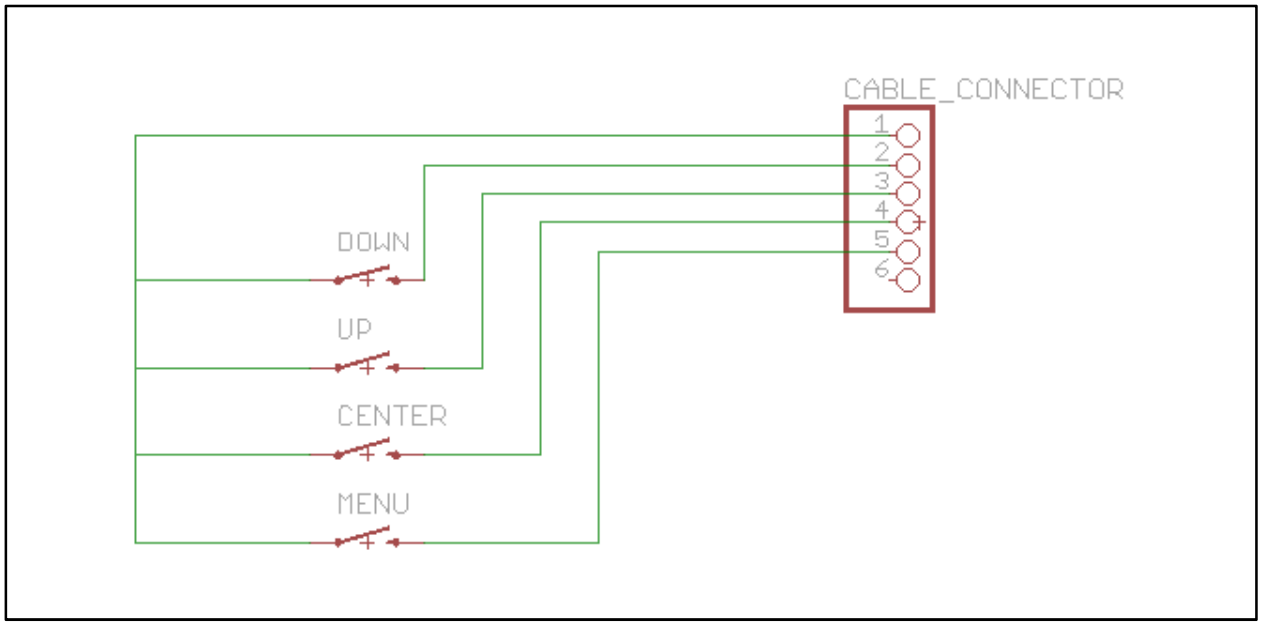


Figure 4-10. Pushbutton Switch Wiring Schematic.

Note from figure 4-10 that one end of all pushbutton switches are wired together to a common pin on the cable. It does not matter which of the two connectors on each pushbutton switch that you use for this; just choose one of them. Figure 4-11 depicts the construction of this wiring.

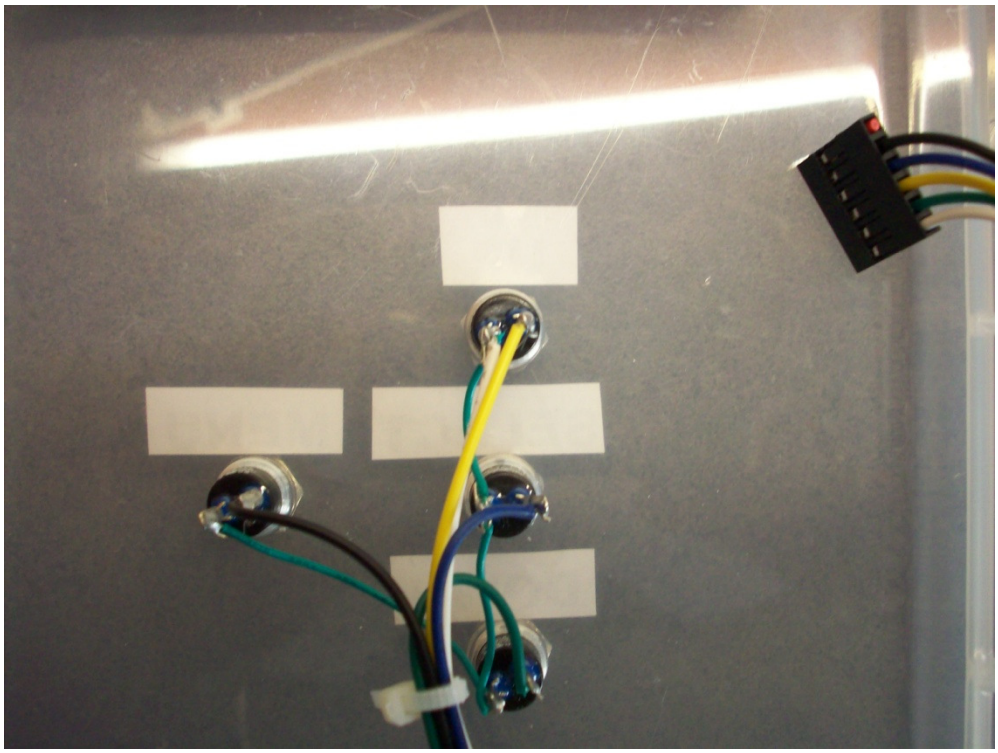


Figure 4-11. Pushbutton Switch Wiring Photo.

Cut and strip lengths of #28 AWG solid wire to interconnect each of the pushbutton switches to the next switch in turn in any order that seems neat and easy. Loop the bare end of each wire through the hole in the connector on the pushbutton. If you start with the MENU button, you can solder this connection and loop the other end of the wire through a connector on the DOWN button, but do not solder this end yet. Pass one end of another length of stripped #28 AWG solid wire through the same connector hole on the DOWN button as the last wire and now solder this connection. Loop the other end of this wire through one of the connector terminals on the SELECT button but do not solder. Take another length of stripped #28 AWG solid wire and loop it through the same connector hole on the SELECT button as the last wire and now solder this connection. Loop the other end of this wire through one of the connector terminals on the UP button but do not solder at this time.

Take one of your 6 conductor interconnect cables. It looks like figure 4-12. Using your diagonal cutters, cut off one of the connectors by clipping the wires entering the connector body. Now cut off the red wire from the other connector, right where it enters the connector body, so that it the connector end of the cable looks like the picture in figure 4-11. Now you have a cable with one connector at one end, 5 wires total in the cable, and five single wires at the other end of the cable. Carefully strip each of the loose wires with your wire stripper and tin the stranded bare

wires. Now place the **white** wire through the pushbutton switch connector hole where the last unsoldered #28 AWG solid wire is. Solder this connection.

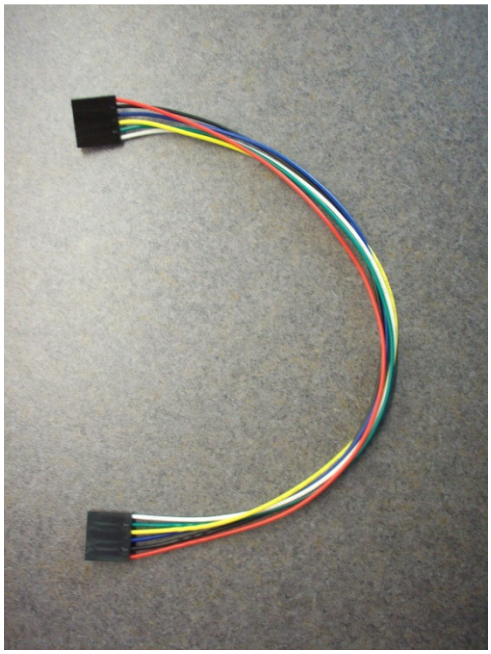


Figure 4-12. Six Conductor Interconnect Cable.

Solder the remaining wires to the empty connectors on each of the pushbuttons, as follows:

- Black wire to MENU pushbutton.
- Yellow wire to UP pushbutton.
- Blue wire to SELECT pushbutton.
- Green wire to DOWN pushbutton.

Double check your work against the schematic (figure 4-10) and the picture in figure 4-11. When you have verified correctness, you can dress up the cable that you just made with cable ties to keep the cable wires together neatly.

4.3 Controller Box.

The Controller cover is complete. Next comes the assembly of the components inside the box. These components, shown in figure 4-13, are:

- Electronic Module, consisting of an Arduino Uno microcontroller with the Shield Board piggybacked on top. The Electronic Module is located on the lower left hand side of the box.
- Audible Alert module, mounted to the right hand side of the box.

- Digital Output terminal blocks – two 8-connection terminal blocks mounted near the top of the box. Alternative terminal strips suggested by Thurmond Moore are described in Appendix C.
- One Wire Bus terminal block – one 3-connection terminal block mounted opposite the Electronic Module on the right hand side of the box. Alternative terminal strips suggested by Thurmond Moore are described in Appendix C.

The placement of components inside the box was chosen for convenience. You are free to locate them anywhere else that you wish as there is plenty of room inside the box.

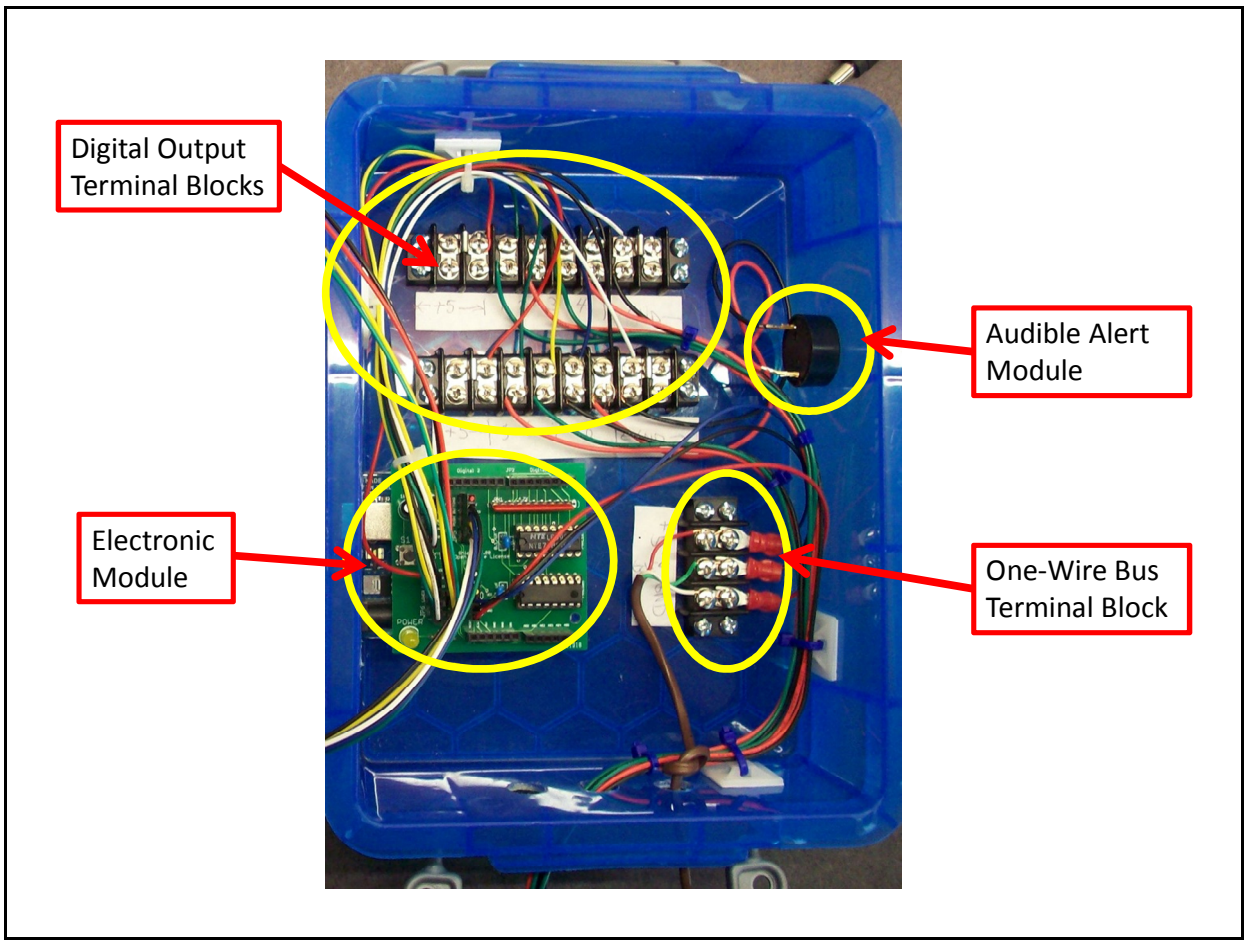


Figure 4-13. Components Mounted inside the Controller Box.

Figure 4-14 is a blow-up photo of the inside of the Controller box when completely assembled.

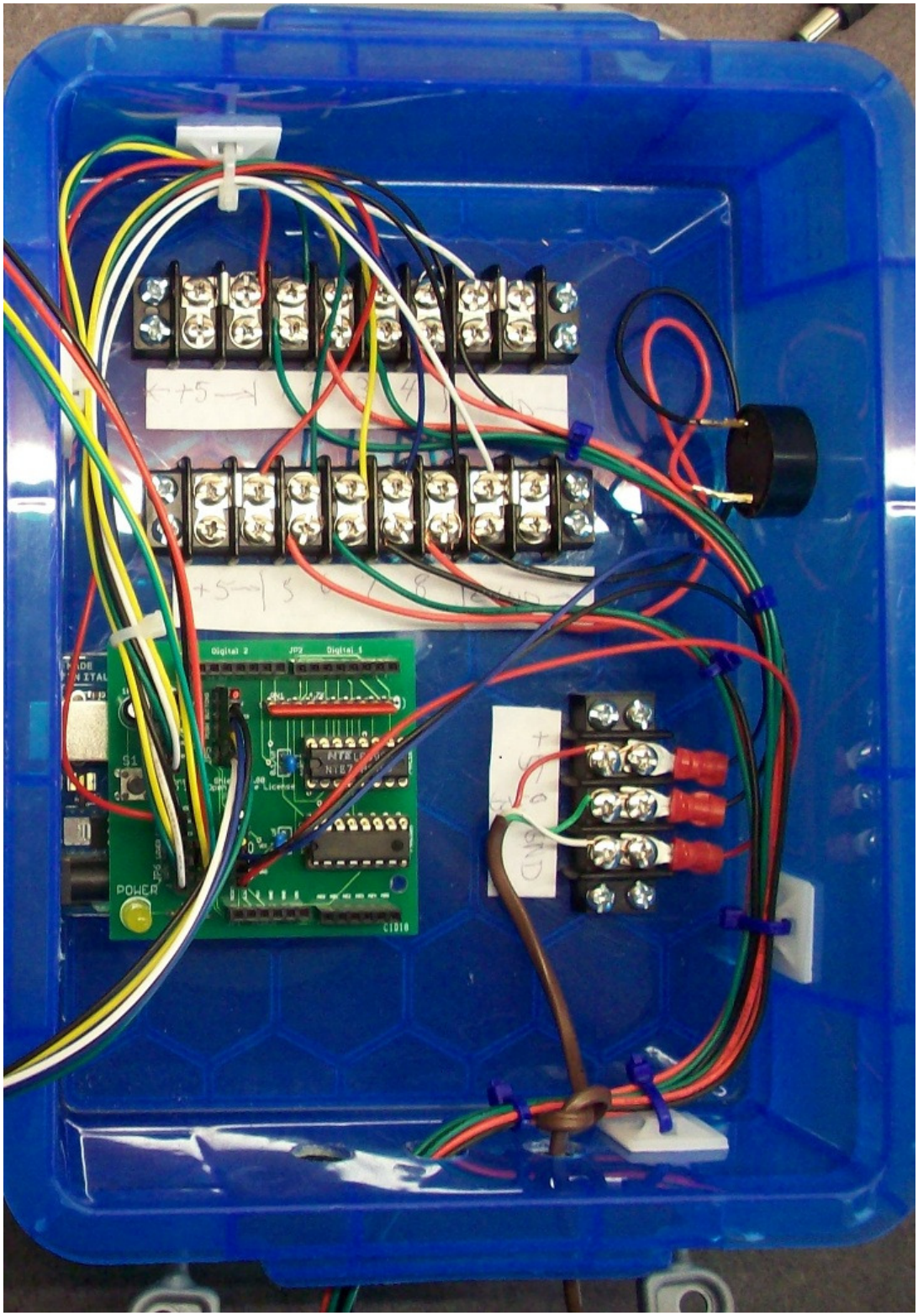


Figure 4-14. Detailed Photo of Inside of the Controller Box.

4.3.1 Preparing to Mount the Electronic Module.

The Controller's Electronic Module consists of an Arduino Uno microcontroller board and a custom made Shield Board that mounts on top of the Uno. The Shield Board simply plugs into the header connectors on the Uno and there are no steps needed to prepare the Controller box for the Shield Board. Preparation is necessary to mount the Uno board into the box.

Figure 4-15 shows the Arduino Uno board.

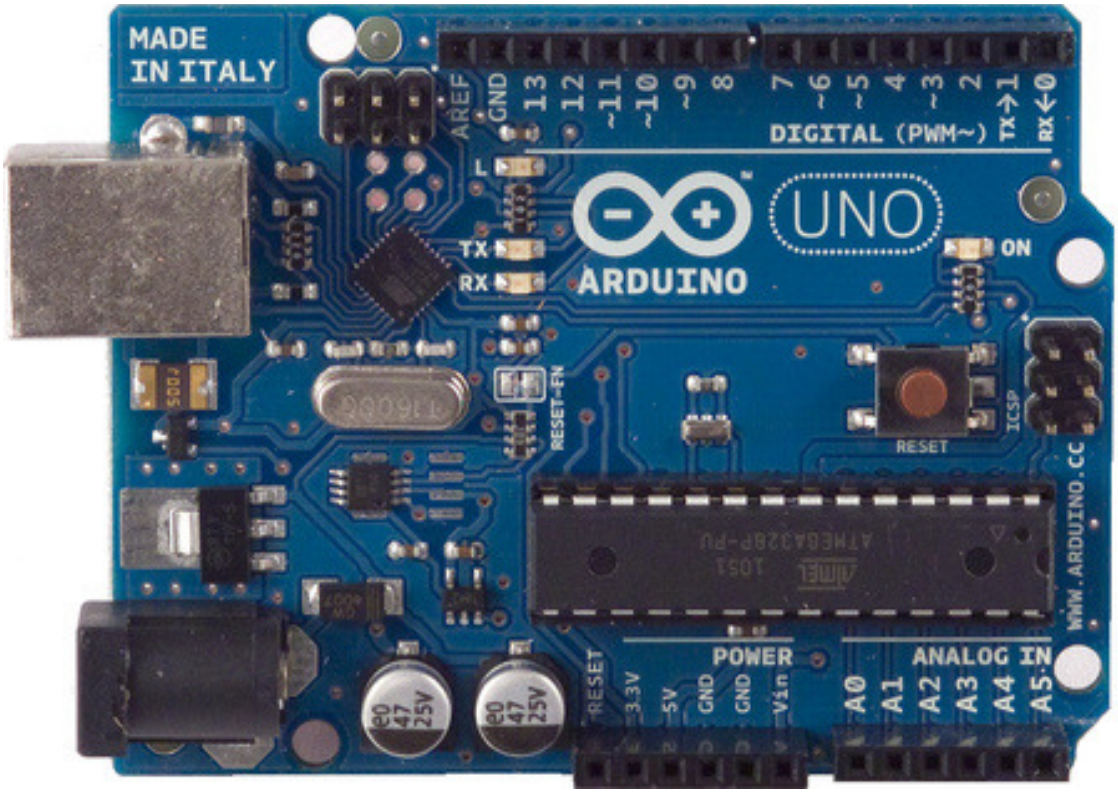


Figure 4-15. Arduino Uno Board.

Notice, on figure 4-15, that the Uno has 4 mounting holes accepting size #4-40 screws. It also has two connectors on the front – a USB connector and a power connector. These connectors protrude past the front of the board and must be accessible from outside the Controller box.

When you first unpack your Uno board from its box, you will notice that the package contains four little rubber stick-on feet. I recommend that you attach these feet to the underside of the Uno board. In so doing, take care to find locations to stick on the rubber feet that are not obstructed by pins of the circuitry soldered onto the Uno board. The feet should be stuck to places where they sit flat. In so doing, the Uno will sit flat and level when placed face up inside the Controller box.

Note the location where the Uno board mounts from figure 4-14, above. The exact location in the Controller box is not critical. Just place it far enough up from the bottom of the box so that it

is above the box's side reinforcements. Cover the area on the inside bottom of the Controller box where the Uno will go with blue painter's tape. Sit the Uno board in the box at this location with the USB connector flush against the inside of the box. Try and square up the Uno board with the side of the box such that after holes are made for the two connectors to protrude through the box, the front edge of the Uno board itself will be flush with the side of the box.

The next step will be to mark the locations to drill holes in the side of the box for the two Uno board connectors to protrude though. The goal is to end up with something like figure 4-16.

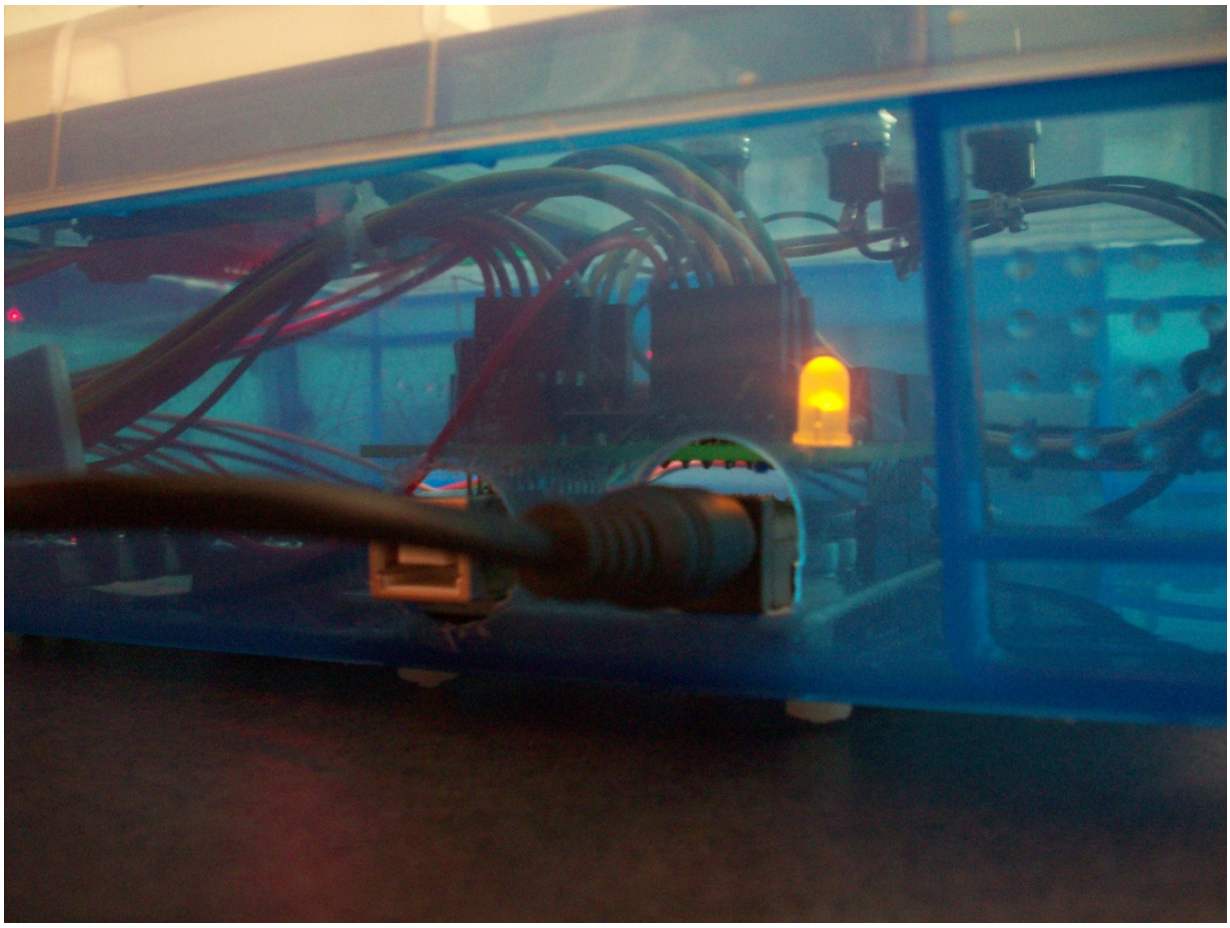


Figure 4-16. Holes for Power and USB Connections to the Uno Board.

With the Uno board USB connector flush against the side of the box and the board square with the box, take a grease pencil or laundry marker pen and mark the outline of the USB and Power connectors on the outside of the box. The box is transparent (actually clear and not even tinted blue, as in the figure), so seeing the connector outlines is very easy.

Remove the Uno from the box so that you have an empty box. Measure and mark the center of the connector outlines that you drew. Clamp the box to some pieces of 2x4 so that there is a firm wood backing behind the cutout markings that you made, and clamp the whole assembly into a bench vise or other means of supporting your work. Center punch the center of the two cutouts

and drill a small pilot hole in each. Now you can drill out holes for the connectors to protrude through. You have several choices here and all will work:

- Drill round holes large enough for the connectors to fit through comfortably. Something like a 5/8" wood drill will do nicely.
- Drill ¼" holes and use small round and flat files to file out the profile of each connector; again so that the connectors fit through comfortably when the Uno board is flush with the inside side of the box.
- Some combination of the above (figure 4-16 has a combination for the power connector).

Unclamp your work, clean the box off with a rag, and check the alignment of everything by putting the Uno board into the box with the board edge firmly and squarely against the inside of the box. File out one or both holes as needed if the fit isn't right. Clean off the box after filing.

Once the fit of the Uno board is right, line the Uno board up carefully and mark the 4 mounting holes of the Uno board on the painter's tape below. Take some care in doing this, as the little rubber feet have the Uno board raised a little off of the bottom of the box.

After marking the 4 mounting holes for the Uno, remove the Uno board, clamp the box to some 2x4 wood pieces for drilling and clamp the whole assembly into a bench vise so that you can drill through the painter's tape, the box bottom and into the wood backing below. Center punch the 4 holes and drill mounting holes with a 7/64" diameter bit at each of the four marked locations (1/8" will work as well). Unclamp your work and double check the placement of the Uno board and alignment of all holes, but do mount the Uno board to the box at this time. You can remove the painter's tape and clean the bottom of the box at this time if you wish.

4.3.2 Preparing to Mount the Terminal Blocks.

Next comes preparation to mount the three terminal blocks inside the Controller box. The position of the terminal blocks is shown in figure 4-14. The position of the blocks is not critical and use figure 4-14 as a general guide. Leave yourself enough space to mount wires to the top and bottom screws of each terminal block. Keep the two 8-connection terminal blocks close to the left edge of the box so that the pre-made cables used later to make connections to the Electronic Module will reach.

Cover the areas of the inside bottom of the box where the terminal blocks will go with blue painter's tape. Place each terminal in its chosen location and mark the mounting holes with a pencil. Remove the terminal blocks, clamp the box to some 2x4 wood scraps for drilling and clamp the assembly in a bench vise to hold the work for drilling. Center punch the mounting holes where you marked them and drill small pilot holes for each. Then drill out the holes with a 5/32" drill for #6-32 screws. After drilling, unclamp the assembly and double heck the alignment of all holes. The remove the painter's tape and clean the inside of the box with a clean rag.

Now drill three holes, approximately ¾" diameter, in the middle section of the bottommost side of the box. None of these holes are critical, as to either size or location. These holes are for wiring from the terminal blocks to probes and power drivers that are outside of the Controller box.

4.3.3 Mounting the Audible Alert Module.

Figure 4-17 shows the audible alert module that must be fastened to the right inside of the Controller box. The module is found at: http://www.taydaelectronics.com/piezo-electronic-tone-buzzer-alarm-1-5-28v-pcb.html.

Figure 4-17. Audible Alert Module.

The location of the alert module is not critical. Use figures 4-13 and 4-14 as a general guide.

> ***Note**: the Audible Alert modules in the parts list are PC mount modules and must be glued to the side of the box and have connecting wires soldered to them. An alternative, box ear mounting version with wire leads is being evaluated and, if suitable, may be substituted for this module. These instructions are for the module shown in figure 4-17.*

The audible alert module must be glued to the inside of the Controller box with a plastic superglue. Prior to gluing, a hole must be drilled to let the sound out of the center hole in the module. Line up the module on the inside of the box where you wish to locate it and mark the center of the module on the outside of the box with a grease pencil or laundry marker. Remove the module, clamp 2x4 wood scraps to the inside of the box where you will drill, and clamp the whole assembly into a bench vise to hold the work for drilling. Center punch at your mark, drill a small pilot hole and then drill it out using a 7/32" bit (up to a ¼" bit will work; the hole must be larger than the hole in the center of the alert module and leave sufficient space for gluing the module).

The next step will be to glue the module to the inside of the box using a plastic superglue. I used Devcon "Plastic Welder" adhesive, part number S-220/22045. You will find something similar in any good hardware store.

Un-assemble the work and clean the inside of the box with a clean rag. Make sure that the location where the module is to be mounted is clean. Cleaning the area with alcohol is recommended, but not essential. Now, place some dabs of plastic superglue on the face of the module outside the center hole, carefully line up the center of the module with the hole that you just drilled and press the module down to the side of the box. Hold in place for the time recommended by the superglue manufacturer (something like 2 minutes) for the glue to set. Then place the box with the module glued into it face down on the side where the module is and leave in place to fully cure the superglue (perhaps 12 hours, per the manufacturer instructions).

Once the glue has set, you need to connect a wire to each pin. Use #22 AWG stranded wire. You can solder a wire to each pin and cover the connection with heat shrinkable tubing or electrical tape. If you have some female "sub-D" crimp pins and a crimping tool, you can crimp a bare pin on one end of each wire and the pin will fit snugly on the module connector pin without the need to solder. The wires need to be long enough to reach the right hand terminals of the lower 8-connection terminal board. Make the wires longer for now and cut and strip the free ends when it is time to actually make the connection to the terminal block screw terminals.

Note that the alert module's terminal pins are marked "+" and "-". It is a good idea (but not required) to use a different color wire for each, e.g. red for "+" and black for "-", so that you don't connect the Audible Alert Module backwards later.

4.3.4 Mounting the Controller Box Components.

The Arduino Uno board is mounted using #4-40 screws, ½" long. Place the board over the mounting holes that you drilled per section 4.3.1, above, insert the screws through the Uno board and down through the mounting holes in the box and tighten a #4-40 nylon nut at the other end and gently tighten the nut. I recommend double nuts to keep things from loosening up, but you can use locking nuts or locktight glue as alternates.

The three terminal blocks are mounted using #6-32 machine screws, ½" long, and matching #6-32 nuts. Tighten gently. There are 4 screws per terminal. Use double nuts, lock-washers, or locktight glue to hold the nuts in place long term.

Take some terminal board jumpers and jumper the end two screws at the left end of one of the 8 connection terminal boards and take another jumper and do the same with the right two screw connectors. Do the same thing with the other terminal block. The jumpered terminal blocks will look like figure 4-18.

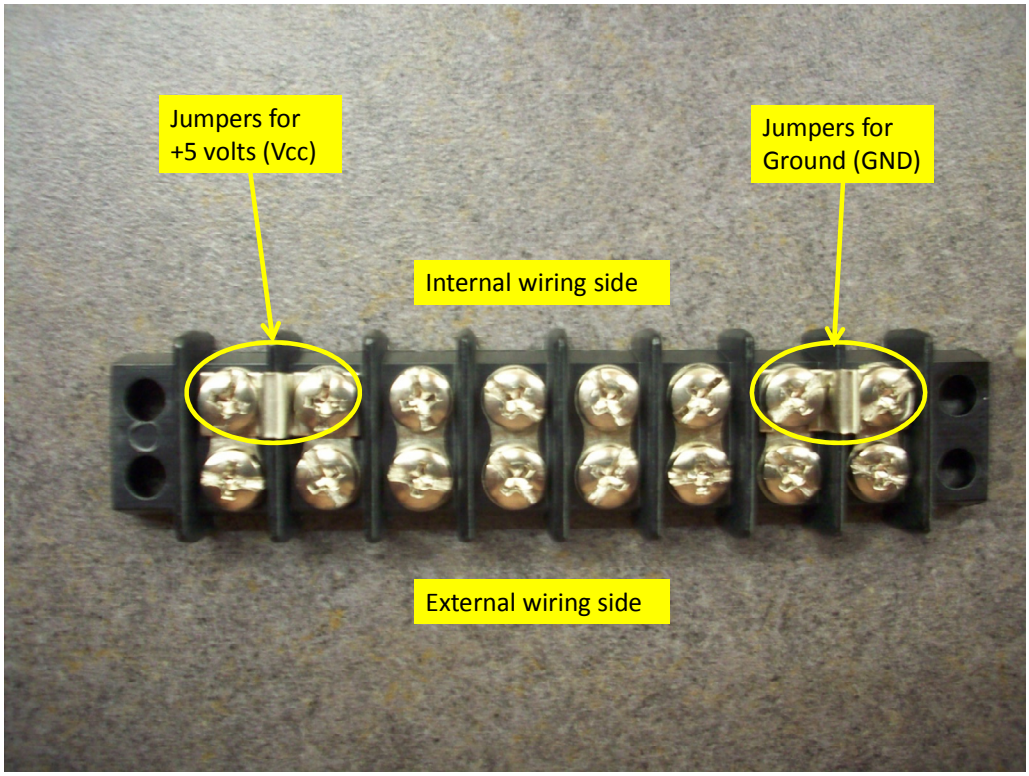


Figure 4-18. Jumpering the Terminal Blocks.

Now place some white marking tape down in front of each terminal board and label the lower screw terminals on all three terminal blocks, as shown in figure 4-14. The topmost terminal block is labeled (left to right):

| +5 | +5 | #1 | #2 | #3 | #4 | GND | GND |

The lower 8 connection terminal block is labeled:

| +5 | +5 | #5 | #6 | #7 | AA | GND | GND |

The three connection terminal block is labeled (top to bottom):

| +5 | OWB | GND |

These labels are to help you remember where to connect the external wiring out from the Controller box.

Lastly, align an assembled Shield Board over the matching header connectors on the Arduino Uno and gently make sure that every Shield Board pin is set into the matching header connector pin on the Uno board. Once the Shield Board is properly aligned, gently press down on the Shield Board to fully it seat it into the Uno headers.

***Note**: The pins on the underside of the Shield Board are longer than necessary and will not insert fully into the Uno header connectors. This is normal. Do not force the Shield Board into the Uno headers. Apply gentle pressure until you feel that the Shield Board is fully seated. As such, it will make excellent electrical connections and hold the assembly together firmly.*

4.3.5 Wiring the Controller Box.

Internal wiring of all components in the Controller box is via 0.1" (2.54 mm) spacing "header" connectors to matching headers on the Shield Board. The Shield Board has the male (bare) pins and the cabling from the LCD display, pushbutton switches, and terminal blocks all have matching female connectors. You already have the connectors for the LCD display and the Pushbutton switches from construction of the Controller box cover. You will make connectors for the three terminal blocks using the 6 connection jumper cables: http://b2cqshop.com/products/170-6-pcs-6-pin-wire-jumpers-female-to-female-for-arduino.aspx. (shown in figure 4-12). Header pins are inexpensive and easy to use but they do have one drawback: they are NOT polarized. This means that there is nothing mechanical to prevent you from inserting the cable end onto the male pins backwards. Please take care to observe the color coding on the cable wires to be sure that you connect them up correctly.

To help familiarize you with the Shield Board connection points and orientation, figure 4-19 is a close-up view of a bare Shield Board.

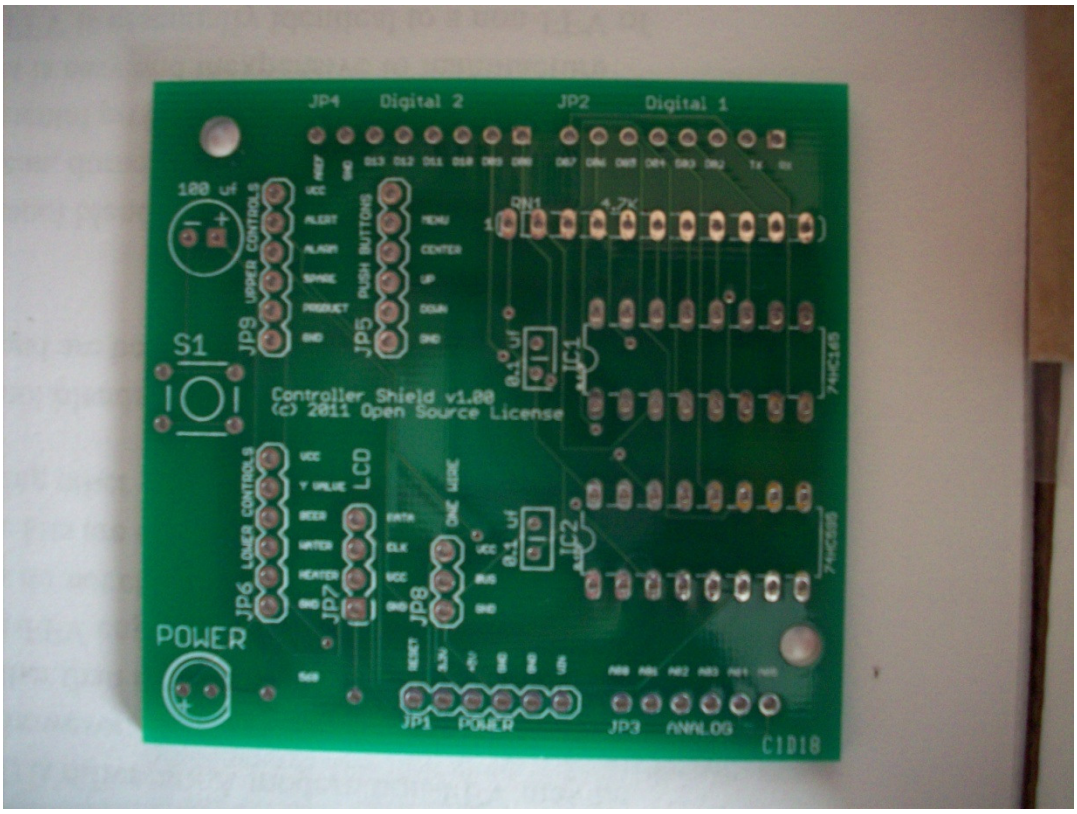


Figure 4-19. Bare Shield Board.

Referring to figure 4-19, you will see two 6-pin header connectors near the front of the board (the side where the "POWER" LED is; left side of figure 4-19). JP6 provides the lower digital output signals (#1 - #4) and JP9 provides the upper digital output signals (#5 - #7, and #8 which is for the Audible Alert). Each of these connectors has pin #1 (lowest on the figure) connected to ground and pin #6 (highest on the figure) connected to Vcc (+5 volt power). The middle pins are the digital outputs in numerical order, lowest to highest on the figure.

JP7 is to the right of JP6 and is a 4 pin connector wired for the LCD. JP8 is a 3 pin connector to the right of JP7 and is for the One-Wire Bus. The lowest of the three pins on JP8 in the figure is ground, the middle pin is the bus wire itself and the top pin is Vcc (+5 volts).

JP5 is to the right of JP9, at the top of figure 4-19. It is a 6 pin header but only 5 of the pins are wired. From the bottom (as per the figure) up, the pins connect to ground, DOWN pushbutton, UP pushbutton, CENTER pushbutton, and MENU pushbutton. If you assembled the cover correctly in step 4.2.4, above, then the cable connector from the pushbuttons is correct and ready to plug into the assembled Shield Board.

You will be wiring up the box per figure 4-14. The following steps are suggested:

- Take one of the 6 conductor interconnect cables and cut off the connector at one end. Take the other end (the one with the connector still on it) and plug it onto the JP6

connector pins of the Shield Board so that the white wire is at the bottom per figure 4-19 (white wire to Ground) and the red wire is at top (red wire to Vcc).

- Take another of the 6 conductor interconnect cables and cut off the connector at one end. Take the other end (the one with the connector still on it) and plug it onto the JP9 connector pins of the Shield Board so that the white wire is at the bottom per figure 4-19 (white wire to Ground) and the red wire is at top (red wire to Vcc).
- Take the 4 wire cable from the LCD display and plug it into the LCD connector in the Shield Board, JP7. The green wire is at the bottom, then the yellow wire is next up, then the red wire is next up, then the black wire is topmost. This is a good time to recall that this special LCD cable interchanges wires within the cable, so in addition to making sure that you plug the cable into JP7 with the correct polarity, ensure that the correct end is on the LCD (red, black, yellow, green) and the correct end is on the Shield Board (green, yellow, red, black).
- Take the 5 wire cable from the pushbutton switches (6 pin connector with one wire clipped off) and plug this cable into JP5 on the Shield Board. When viewed with the orientation of figure 4-19, the lowest wire is white, followed by green, yellow, blue and black. The red wire has been cut off when assembling the box cover (per section 4.2.4, above).
- Take the loose wires from the cable on JP6 and connect them to the upper row of screws on the upper terminal block screw connectors as follows:
    - Red wire to the second screw terminal from the left (+5 volts)
    - Green wire to the next screw terminal to the right of red (Digital Output #1)
    - Yellow wire to the next screw terminal to the right of green (Digital Output #2)
    - Blue wire to the next screw terminal to the right of yellow (Digital Output #3)
    - Black wire to the next screw terminal to the right of blue (Digital Output #4)
    - White wire to the next screw terminal to the right of black (Ground)

    Take care when stripping the fine wires. Either tin the wires or, better still, use spade lugs to connect the wires to the screw terminals.

- Take the loose wires from the cable on JP9 and connect them to the upper row of screws on the lower terminal block screw connectors as follows:
    - Red wire to the second screw terminal from the left (+5 volts)
    - Green wire to the next screw terminal to the right of red (Digital Output #5)
    - Yellow wire to the next screw terminal to the right of green (Digital Output #6)
    - Blue wire to the next screw terminal to the right of yellow (Digital Output #7)
    - Black wire to the next screw terminal to the right of blue (Digital Output #8)
    - White wire to the next screw terminal to the right of black (Ground)

    Take care when stripping the fine wires. Either tin the wires or, better still, use spade lugs to connect the wires to the screw terminals.

- Connect the wire from the "-" side of the Audible Alert module to the rightmost screw terminal on the lower terminal block (Ground). Connect the wire from the "+"side of the Audible Alert module to the Digital Output #8 screw terminal on the lower terminal block (Ground). This connects the Audible Alert module to the Electronic Module using "active-high" control, as described in section 2.3 of this manual.
- The last connection is JP8, the One-Wire Bus to the three screw terminal block to the right of the Electronic Module. I took another of the 6 conductor interconnect cables and cut off one end. I then cut off the white, green and yellow wires from the other end of the cable, leaving a 6 wire connector with only three wires on it, as shown in figure 4-20, below. Make sure to connect this cable connector to the three pins of JP8 as shown in figure 4-20, i.e. with the three remaining wires carrying these three signals. Connect the three wires to the rightmost column of screws on the three position terminal block as follows:
    - Red wire to the lowest terminal of the three screw terminal block (Ground).
    - Black wire to the middle terminal of the three screw terminal block (One-Wire Bus signal).
    - Blue wire to the uppermost terminal of the three screw terminal block (+5 volts).

  Take care when stripping the fine wires. Either tin the wires or, better still, use spade lugs to connect the wires to the screw terminals.

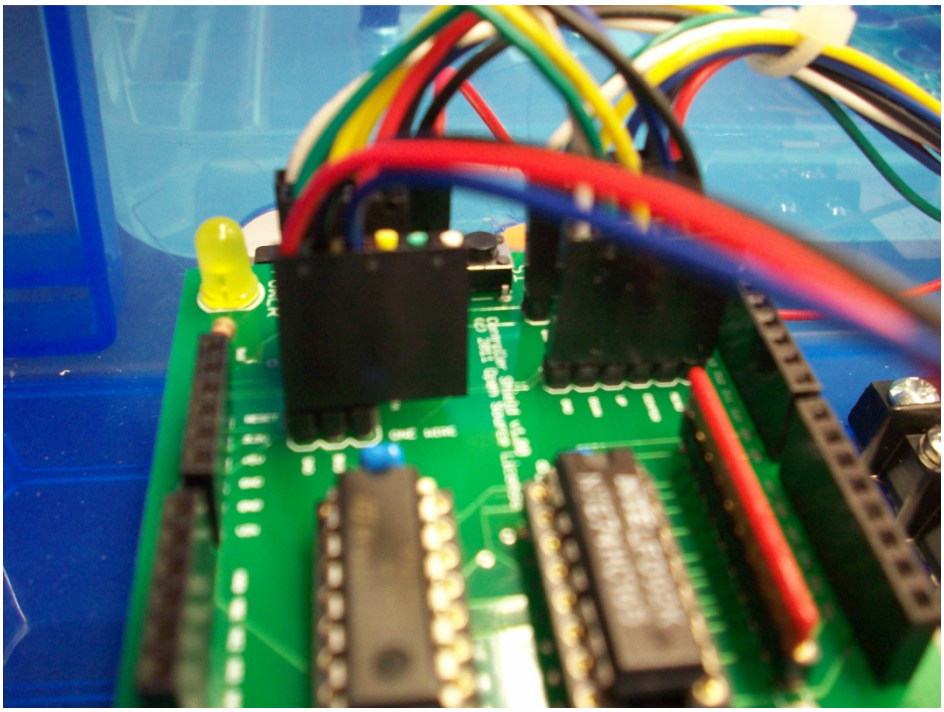


Figure 4-20. One-Wire Bus internal Cable.

Now, place the cover, face up, on the Controller box taking care to keep the cables inside the box. Snap the handles on the top and bottom of the box to the box cover to fasten the box and cover together.

4.4 Test Board.

The Test Board is not part of the Controller and its construction is optional. The Test Board is a simple device to test the One-Wire Bus and 7 Digital Outputs from the Controller when running the Test program, which is described in section 3.6, above. You can construct the Test Board according to instructions in this section and then load the Controller Test Software into your completed Controller per chapter 3 of this manual.

The Test Board is constructed using a solderless breadboard, as shown in figure 4-21.

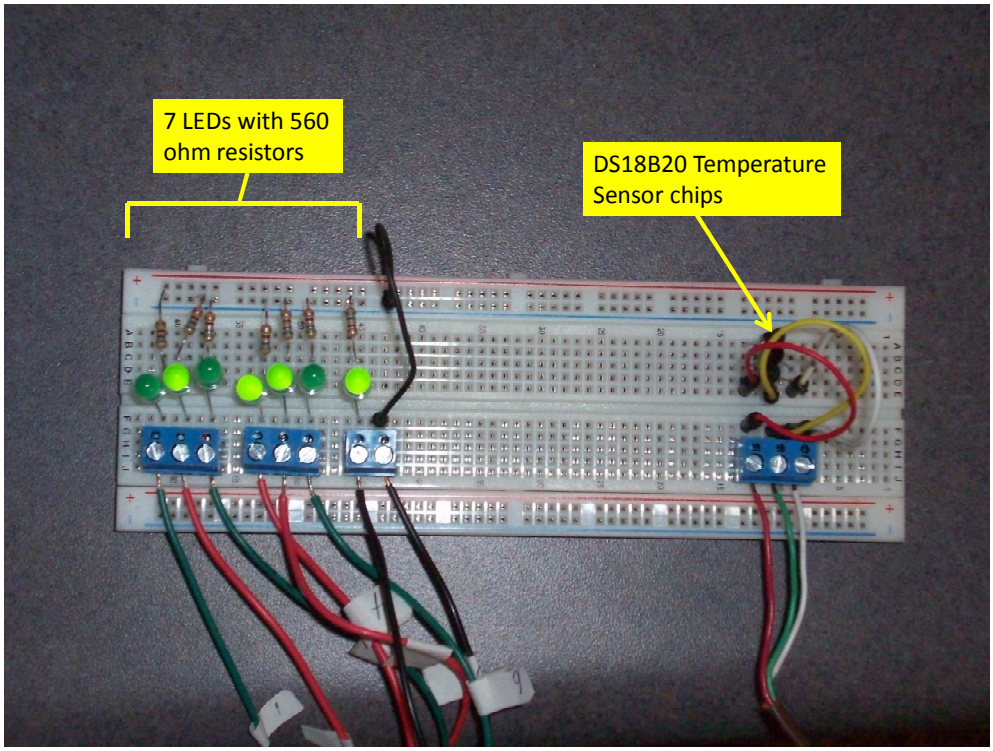


Figure 4-21. Test Board Detail

Figure 4-22 contains the schematic for wiring the test Board.

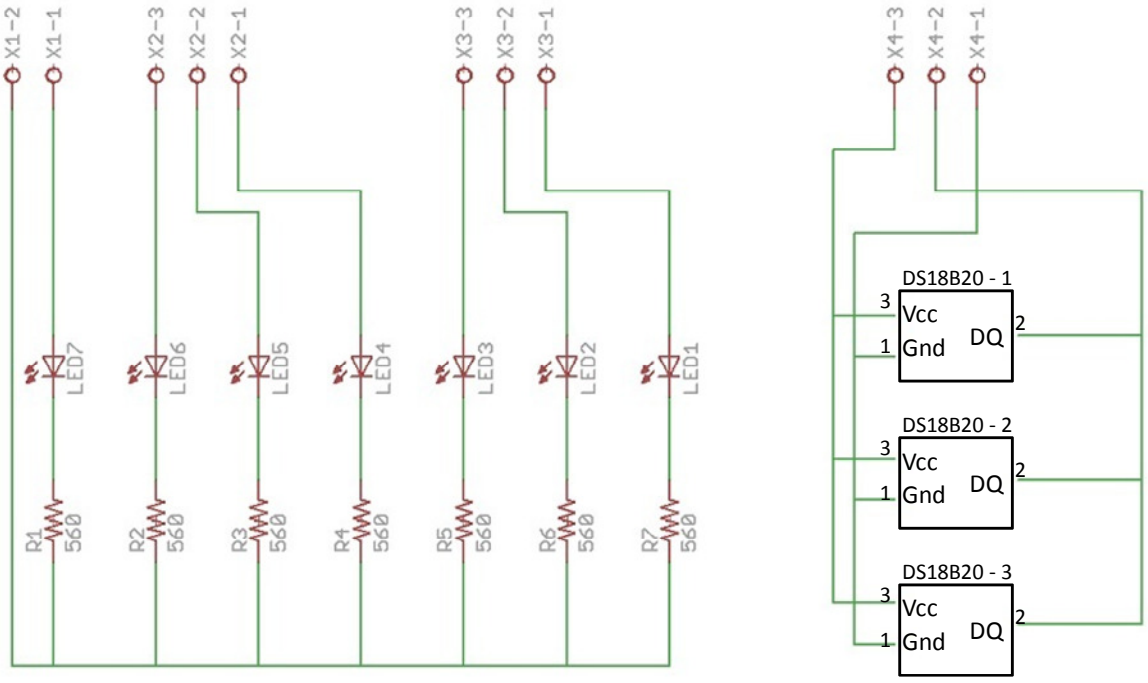


Figure 4-22. Test Board Schematic.

The Parts list (Appendix A) contains the materials needed for the Test Board. As a breadboard used only for testing, it can be constructed in any way that is inexpensive, quick and simple. I purchased some 0.2" center (5.08 mm) printed circuit mount screw terminals to make it easy to connect wires from the Controller's terminal blocks to the Test Board, as shown in figure 4-21.

The solderless breadboard contains rows of interconnected holes into which you can stick component leads, printed circuit board connectors, and jumper wires. Rows A through E are interconnected in each column of the board, as are rows F-J. Since the one-wire bus devices, such as the DS18B20 chips, all connect to the same three wires of the bus, you need only to line up the three chips on any three columns of the breadboard to interconnect all +5 volts (Vcc) pins, all Bus (DQ) pins, and all Ground (Gnd) pins from all three chips. I used rows A, B and C for the chips. I put a three connection PC terminal block in similar rows in the F-J section of the breadboard and used jumper wires to connect each terminal to the proper column for +5 volts, One-Wire Bus, and Ground for the DS18B20 temperature sensor chips. The pin-out for the

DS18B20 chips is shown in figure 4-23. Pin #1 is Ground, pin #2 is the One-Wire Bus and pin #3 is +5 volts (Vcc).

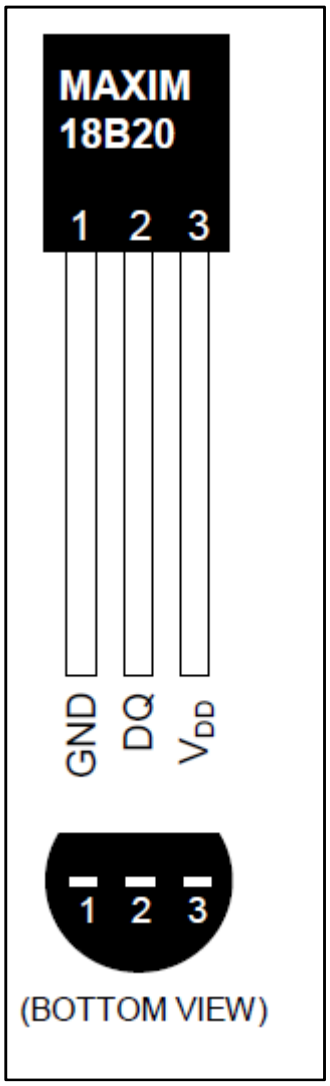


Figure 4-23. DS18B20 Pinout.

Back on the other end of the breadboard, I placed two 3-screw PC terminal blocks and one 2-screw PC terminal block, as shown in figure 4-21. I placed the positive end of an LED (the end with the longest lead) in row F of the column for each of the 6 screw terminals of the 3-terminal connectors and the leftmost terminal of the 2-teminal connector, making 7 LEDs in all. I plugged the other lead of each LED to row E of the column where the positive lead was placed. I plugged one end of a 560 ohm resistor into row C of each LED column and the other end of each resistor into the "-" Bus of the breadboard above row A. All such "-" row pins are already interconnected. I used a jumper to connect the last (rightmost) of the 2-terminal connector from row F of its column to the uppermost "-" bus to provide the common ground connection back to the Controller, as shown in figures 4-21 and 4-22.

Lastly, I used some #22 AWG stranded wire and connected from the Digital Output terminal blocks of the Controller to the LED terminal blocks on the breadboard, and from the One-Wire Bus terminal block of the Controller to the One-Wire bus block of the breadboard. Digital Output #1 is connected to the leftmost LED, Digital Output #2 to the next to the leftmost LED, etc, for all 7 LEDs. One of the Ground connections on the Controller's Digital Output Terminal Blocks is connected to the breadboard Ground terminal connection. This wires all seven LEDs in the "active-high" configuration, as described in section 2.3. Active-low can be used as well; just be sure to change the binary constant in the Controller Test Software to the correct polarity for the Controller digital outputs. Please make sure to wire this up correctly – it helps to label each wire coming out of the Controller, as shown in figure 4-21.

I used a three-wire thermostat wiring cable to connect the One-Wire Bus Terminal Block connections on the Controller to the corresponding connections on the breadboard. This was convenient, but the solid wire is stiff and the PC terminal connector on the breadboard tends to pop out. Therefore, #22 AWG stranded wires are recommended here as well.

As always, it is best to tin the wires going to terminal block screw connectors and to use spade lugs for the Controller terminal bocks, so that tightening the screws don't twist off the bare end of the wire.

You run the wires out of the Controller box through one or more of the three holes that you drilled in the bottom side of the Controller box. I suggest dressing up the wires with cable ties and stick-on cable tie mounts, as shown in figure 4-14. This keeps things neat inside the Controller box.

Make sure that all cables are connected properly to the Electronic Module of the Controller and close and latch the cover to the Controller box. You use the Arduino Uno USB connector to load in the software and test your completed Controller per the directions for software in Chapter 3.

5 Shield Board.

The Shield Board is part of the Electronic Module of the Controller. The Shield Board must be assembled by soldering the components onto the custom printed circuit board that you can purchase at:
http://www.batchpcb.com/product_info.php?products_id=75513&check=a69606452bf49d346372d5d822638f62

A complete parts list for the Shield Board is in the parts list, Appendix A.

5.1 Shield Board Overview.

The Shield Board expands the Arduino Uno microcontroller's I/O pins for the Controller. It provides pull-up and termination resistors for the pushbutton switch interface and for the One-Wire Bus. It provides male pin header connectors for a neat connection of all Controller parts and terminal blocks into the Electronic Module. It also provides a power on LED indicator and miniature "reset" pushbutton, as these functions on the Uno board are covered up when the Shield Board is in place. Appendix B contains a schematic of the Shield Board, for those readers who are interested. Eagle CAD files for the Shield Board are included in the release package for the Controller.

Figure 5-1 is a photo of the bare Shield Board and figure 5-2 is a photo of the Shield board after it is fully assembled and piggybacked onto the Uno board to form the Controller's Electronic Module.

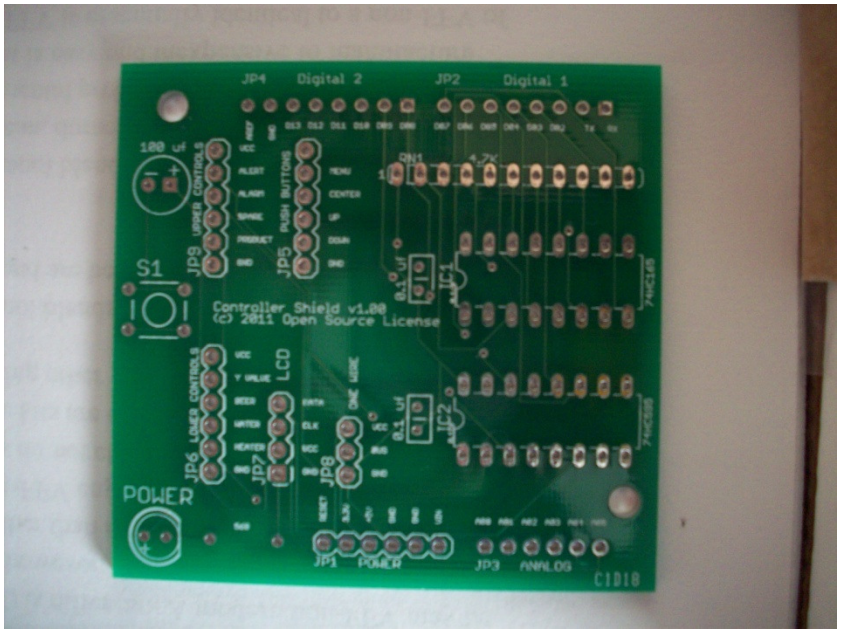


Figure 5-1. Bare Shield Board, As Purchased.

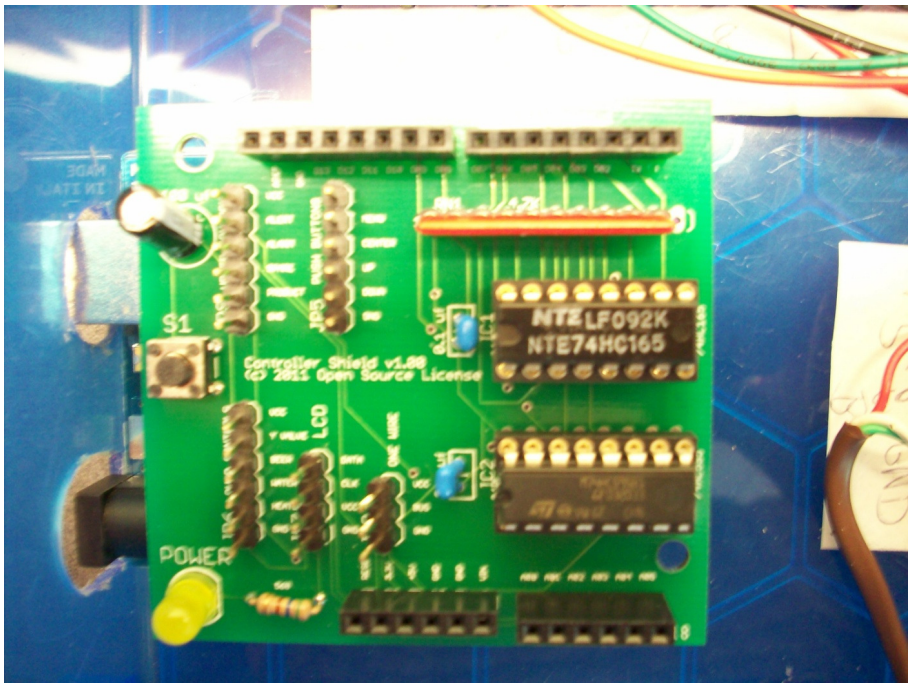


Figure 5-2. Shield Board, Fully Assembled.

***Note**: The printed circuit board for the Shield Board is a first generation design, specifically for the Controller project. It has a few minor defects, none of which interfere with its assembly or function. The silkscreen outline for three of the connectors is on the bottom of the board, vs. the top of the board, but the silkscreen for the text is all on the correct side. Additionally, there are 11 holes for the single in-line socket that holds the resistor network, but the resistor network and its SIP socket are only 10 pin devices. The 11th pin (rear of the board) has no connections and is left blank, as described in the assembly instructions.*

5.2 Tools and Materials.

In order to assemble the Shield Board, you will need the basic electrical tools described in section 4.1, above and figures 4-2, 4-3, 4-4, and 4-5. Here is a summary of the tools required:

Basic Electrical Tools:

- Soldering iron with small tip, electrical solder (#22 AWG), solder wick and/or solder pullet (de-soldering tool).
- Diagonal cutters.
- Long nose pliers.
- Magnifying glass.
- Optional – recommended: inexpensive multi-meter.

The soldering iron should be limited to 25 watts and have a removable tip (e.g. Weller). A soldering stand, as shown in figure 4-2 is also recommended. A fine tip and thin solder is necessary for the electrical work required to assemble the Shield Board. Figure 4-3 shows a close-up of a fine soldering tip and fine (#22 AWG) solder. A small magnifying glass is very useful for inspecting detailed work – see example in figure 4-5.

You will need a means to hold your work when soldering. An inexpensive hobby vise, such as shown on figure 5-3, will suffice. If you can find an apparatus that clips your work into place and has a magnifier to look thorough when soldering, so much the better.

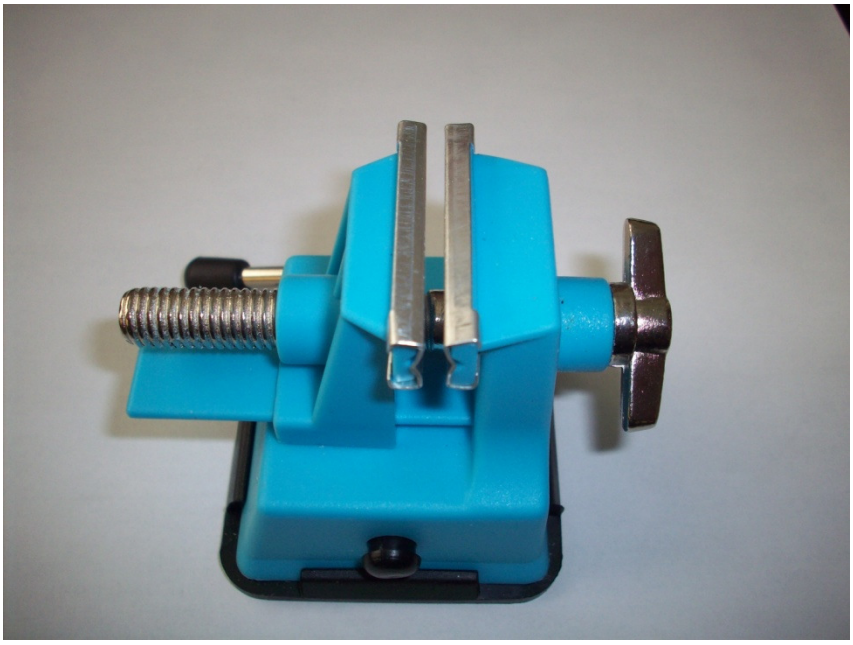


Figure 5-3. Hobby Vise.

5.3 Shield Board Assembly.

The top (silkscreen side) of the blank Shield Board (figure 5-1) gives the locations (and polarity, if required) of all components on the board. The following components are polarized and require special attention to their orientation when soldering or socketing them:

- LED: the long lead on the LED is the positive "+" side.
- Electrolytic capacitor (100uf): the long lead is the positive "+" side; the negative "-" side is marked with a stripe with minus signs (bars) on it for the negative side. The capacitors that you buy may vary in appearance but all have some notation for marking the polarity.
- Single in-line (SIP) resistor network: pin #1 is marked with a black bar. Pin #1 goes toward the front (LED side) of the board, as indicated on the silkscreen. The resistor network plugs into a single in-line (SIP) socket that is soldered to the Shield Board. The socket is not polarized.

- Integrated circuits (74HC165 and 74HC595): Pin #1 goes toward the front (LED side, and the side where the small 0.1 uf capacitors are) of the board. There will be an indentation at the top of the chip or a little dimple indented by pin #1, depending upon the manufacturer. The integrated circuits plug into sockets that are soldered to the Shield Board. The sockets are not polarized, albeit some sockets have an indentation or other indication at the top (the side where pin #1 is). Although a socket can be soldered in either way and will work fine, it is best to line it up properly so that it is obvious which way the integrated circuit chip plugs in.

Assembly of the Shield Board requires soldering. Good electrical soldering skills are required. Electrical soldering is different than soldering plumbing. If you have never soldered together electronics, you can find a good tutorial here: http://www.sparkfun.com/tutorials/213

The sockets, and in particular the pin headers that piggyback to the Arduino Uno are the hardest part of the assembly because they must be held onto the board upside down and be soldered flush to the board and with the pins straight up and down (at right angles to the plane of the board). Even if you have a vice to hold the board still while soldering, you still need three hands to do this correctly: one to hold the socket in place, one to hold the soldering iron and one to apply the solder to the pin/solder pad on the board. Since you probably only have two hands like I do, you need some way to hold the connector in place. I just used some electrical tape, as illustrated in figure 5-4. Make sure that the tape adheres to the socket and then place the socket down on the board in the correct orientation and make sure that the tape adheres to the board and holds the socket fully in place and fully perpendicular to the board. Now you can turn the board over so that the soldering side (bottom) is exposed and place the board in or on a vice or other place so that there is no pressure on the connector to be soldered, as illustrated in figure 5-5. Now, solder one end pin to the board, then solder the opposite end pin to the board. Now there is mechanical support to hold the connector in place. Proceed to solder the rest of the pins to the board. Double check that you didn't skip any! It is always a good idea to use your magnifying glass to check your solder connections after soldering one of these connectors. Check that there is enough solder so that the pin and the solder pad on the board are connected. Check that there isn't too much solder and a short circuit between two adjacent pins or a pin and an adjacent via on the board. Normally, the solder flows and adheres to the plated pads on the board and does not flow over to another pin or location.

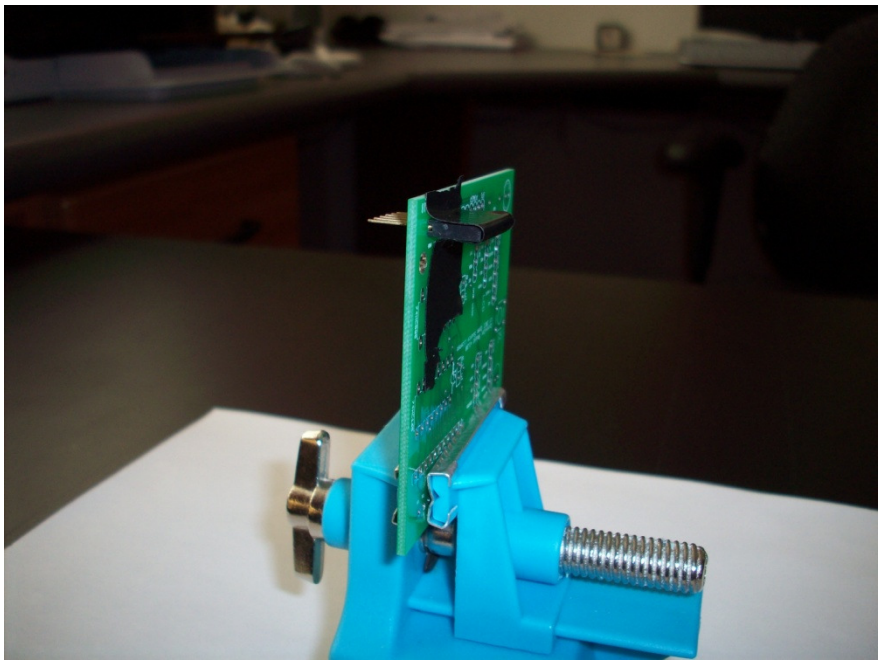


Figure 5-4. Using Electrical Tape to Hold Connectors in Place.

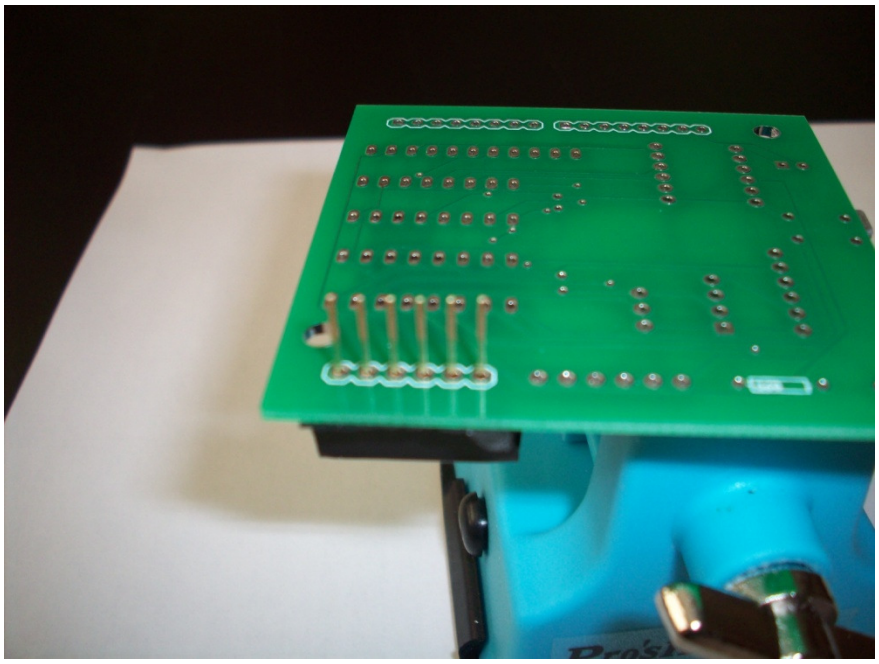


Figure 5-5. Connector Pins Ready to Solder.

You can solder the components to the board in any order that you like but here is my suggested order:

- First solder the 4 Arduino header connectors (JP1, JP2, JP3, and JP4). Be very careful with these as they must be perpendicular to the board. This isn't super critical, but the connectors can get themselves soldered in at an angle that is too great to mate with the Uno. Double check mating with an Uno board before proceeding to assemble the rest of the board. If you do solder in a connector badly, you will most likely have to cut the connector body off with your diagonal cutters and use your solder wick or de-soldering tool to remove one pin at the time. You will then need a new connector, of course!
- Next solder the male headers to JP5, JP6, JP7, JP8 and JP9. If you bought what is listed in the parts list (Appendix A), you will have long (e.g. 40 pin) header strips. Just break off the number of pins that you need for the connector (easily done with your long nose pliers). The short end of the connector gets soldered to the board; the long pins are the connector itself. Again, these connectors need to be held in place with electrical tape and aligned pretty much vertically (it is not as critical as the Arduino header connectors, though). I recommend soldering the connectors in the follow order: JP8, JP5, JP7, JP9, JP6.
- Next, solder the 10 pin single in-line (SIP) connector for the resistor network to the board where it says "RN1". Note again that there are 11 holes for this connector (a bug). You must solder the connector in the ten forward holes starting where pin 1 is marked (see figure 5-6).

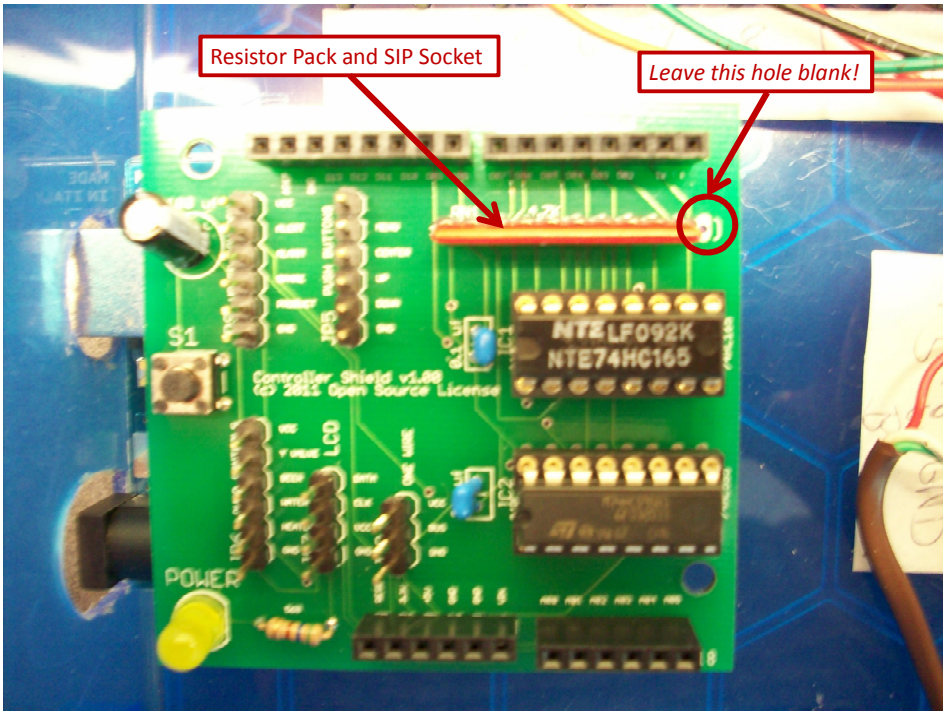


Figure 5-6. Resistor Network and SIP Socket Placement.

- Next, solder the two 16-pin dual in-line (DIP) sockets to the board where indicated (IC1 and IC2). Now you are done with all connectors and electrical tape to hold them in place for soldering!
- Solder in the LED where the board says "POWER". Observe the polarity – long lead is "+". When soldering leaded components, put the leads through the board and bend them over on the solder side to hold the component in place while you solder. Clip off the lead ends at the top of the solder bump after soldering and inspecting your solder joint.
- Solder the pushbutton switch in "S1". Note that the switch has 4 leads – two are connected together inside of the switch for one side and the other two for the other side. The switch should only fit into the holes in the board two ways, and either orientation will work.
- Solder the 100 uf electrolytic capacitor to the board where marked. Observe the polarity!
- Solder the 560 ohm resistor to the board, near the LED, where the "560" is marked. Resistors are not polarized and can go in either way.
- Solder the two 0.1 uf capacitors where indicated. These devices are not polarized and can go in either way.
- Plug the 10 pin resistor network into the 10 pin SIP socket with pin #1 to the front of the board (where the LED and pushbutton are). Pin #1 is marked with a black stripe.
- Plug the two integrated circuits into the 16 pin DIP sockets. Make sure that you put the right one into the right socket as marked on the board. IC1 is the 74HC165 and IC2 is the 74HC595. Make sure that pin #1 is toward the front of the board (LED and pushbuttons side). The orientation of the chip is marked with some sort of dimple or indentation in the chip at the side where pin #1 is. You may have to bend the leads on the chip inward a little to get them aligned with the holes on the socket. You can do this by placing the chip on its side on a flat surface and bending the chip package from the middle. Then bend the pins on the other side of the chip the same way.

***Note**: These integrated circuits use CMOS technology and are very sensitive to electrostatic discharge. They come in a piece of conductive foam to keep static electricity from building up, which can damage the chip. Take care when handling the chips to insert them. There is usually no issue with your handling them briefly. However, if you are in a low humidity place where you get static shocks when you touch doorknobs, you should ground yourself before handling the chips. A grounding wrist strap, available at Radio Shacks and other electronics stores, is recommended in this instance.*

5.4 Attaching the Shield Board to the Arduino Uno.

Inspect your Shield Board after assembly to make sure that you:

- Put the correct parts in the correct place.

- Put all parts in with the correct polarity and correct orientation.
- Did not create shorts with your soldering.
- Bent the leads of the integrated circuits so that all leads are firmly inserted straight into their socket pins.

After verifying your work with a visual inspection, you can insert the Shield Board into your Uno board. At this point, this is just a test of how well things mate up. You will need to separate the boards in order to mount your Uno into the Controller box, per section 4.3.4 of this manual.

Align the fully assembled Shield Board over the matching header connectors on the Arduino Uno and gently make sure that every Shield Board pin is set into the matching header connector pin on the Uno board. Once the Shield Board is properly aligned, gently press down on the Shield Board to fully it seat it into the Uno headers.

> ***Note**: The pins on the underside of the Shield Board are longer than necessary and will not insert fully into the Uno header connectors. This is normal. Do not force the Shield Board into the Uno headers. Apply gentle pressure until you feel that the Shield Board is fully seated. As such, it will make excellent electrical connections and hold the assembly together firmly.*

Construction of the Shield Board is now complete. You will test the Shield Board on the fully assembled Controller using the Test Board of section 4.4 of this manual and the test software that is installed and operated as described in Chapter 3 of this document.

Appendix A. Parts List.

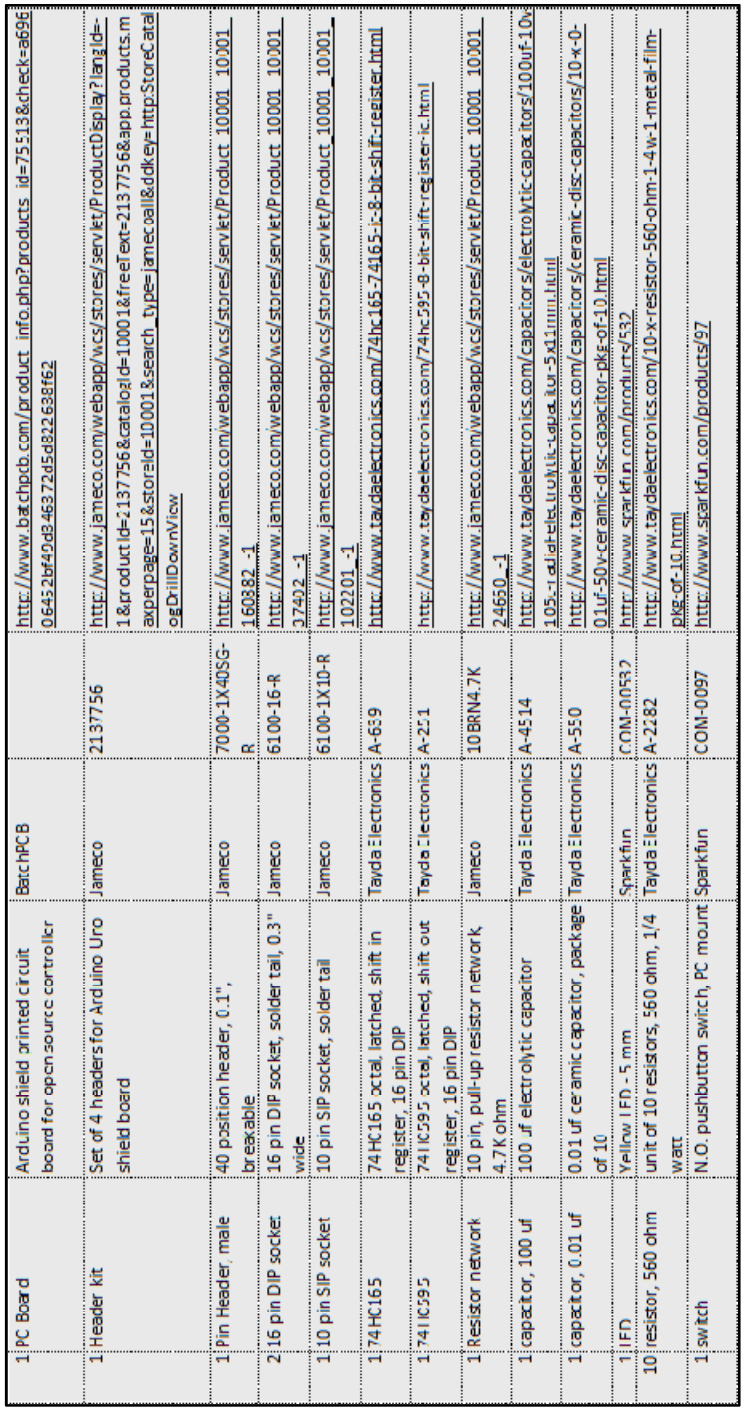
An Excel spreadsheet with the complete parts list, sorted by vendor as well as by subsystem, is included with the release package. The following tables are extracted from the Excel file.

Table A-1 lists the parts for the Controller Box (Chapter 4):

1	Box	Plastic box, 1.75L, 3"H x 7"W x 9-1/2"D, Clear	Office Depot	–	http://www.officedepot.com/a/products/452333/Really-Useful-Boxes-Plastic-Storage-Box/#/firstTab
1	Arduino Uno R3	Microcontroller	Jameco	A000066	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_2151186_-1
1	LCD Module	20x4 character LCD module with I2C interface	B2CQShop		http://b2cqshop.com/products/192-arduino-i2c-serial-2004-lcd-module-20x4-lictwi-lcd.aspx
2	Pushbutton	Pushbutton Switch - red	Tayda Electronics	A-5051	http://www.taydaelectronics.com/electromechanical/switches-key-pad/push-button/pp-11d02-push-button-panel-mount-sast-no-pb-11d02-th1n00.html
2	Pushbutton	Pushbutton Switch - black	Tayda Electronics	A-5050	http://www.taydaelectronics.com/electromechanical/switches-key-pad/push-button/pp-11d02-push-button-panel-mount-sast-no-pb-11d02-th1-00.html
1	Power Supply	9 VDC, 1000 ma switching power supply	Adafruit	63	http://www.adafruit.com/products/63
1	Audible Alert	Piezo electric tone buzzer	Tayda Electronics	A-5019	http://www.taydaelectronics.com/piezo-electronic-tone-buzzer-alarm-1-5-28v-pcb.html
2	Terminal Block	Terminal Block - 8 position	Jameco	231036	http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=231036&catalogId=10001&freeText=231036&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView
1	Terminal Block	Terminal Block - 3 position	Jameco	230981	http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=230981&catalogId=10001&freeText=230981&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView
5	Jumper	Terminal Block Jumper	Jameco	659382	http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=659382&catalogId=10001&freeText=659382&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView
1	Nylon Screw	Bag of #4-40 Nylon screw, 1/2"	Fry's Electronics	3579564	http://www.frys.com/product/3579564?site=sr:SEARCH:MAIN_RS:T_PG
3	Nylon Nut	#4-40 Nylon nut, 2 pack	Home Depot	594613	http://www.homedepot.com/h_d1/N-5yc1v/R-202210238/h_d2/ProductDisplay?langId=-1&storeId=10051&catalogId=10053
12	Machine Screw	#6-32 machine screw, 1/2"	Home Depot	749347	http://www.homedepot.com/h_d1/N-5yc1v/R-202209804/h_d2/ProductDisplay?langId=-1&storeId=10051&catalogId=10053

Table A-1. Controller Box Parts.

Table A-2 lists the parts for the Shield Board (Chapter 5):



1	PC Board	Arduino shield printed circuit board for open source controller	BatchPCB		http://www.batchpcb.com/product_info.php?products_id=7513&check=a696b452bf49d3463724d5d822638f62
1	Header kit	Set of 4 headers for Arduino Uno shield board	Jameco	2137756	http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=2137756&catalogId=10001&freeText=2137756&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView
1	Pin Header, male	40 position header, 0.1", breakable	Jameco	7000-1X40SG-R	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10301_160382_-1
2	16 pin DIP socket	16 pin DIP socket, solder tail, 0.3" wide	Jameco	6100-16-R	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10301_37402_-1
1	10 pin SIP socket	10 pin SIP socket, solder tail	Jameco	6100-1X10-R	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10301_102201_-1
1	74HC165	74HC165 octal, latched, shift in register, 16 pin DIP	Tayda Electronics	A-639	http://www.taydaelectronics.com/74hc165-74165-ic-8-bit-shift-register.html
1	74HC595	74HC595 octal, latched, shift out register, 16 pin DIP	Tayda Electronics	A-211	http://www.taydaelectronics.com/74hc595-8-bit-shift-register-ic.html
1	Resistor network	10 pin, pull-up resistor network, 4.7K ohm	Jameco	10BRN4.7K	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10301_24650_-1
1	capacitor, 100 uf	100 uf electrolytic capacitor	Tayda Electronics	A-4514	http://www.taydaelectronics.com/capacitors/electrolytic-capacitors/100uf-10v-105-radial-electrolytic-capacitor-5x11mm.html
1	capacitor, 0.01 uf	0.01 uf ceramic capacitor, package of 10	Tayda Electronics	A-550	http://www.taydaelectronics.com/capacitors/ceramic-disc-capacitors/10-x-0-01uf-50v-ceramic-disc-capacitor-pkg-of-10.html
1	LED	Yellow LED - 5 mm	Sparkfun	COM-00532	http://www.sparkfun.com/products/532
10	resistor, 560 ohm	unit of 10 resistors, 560 ohm, 1/4 watt	Tayda Electronics	A-2282	http://www.taydaelectronics.com/10-x-resistor-560-ohm-1-4w-1-metal-film-pkg-of-10.html
1	switch	N.O. pushbutton switch, PC mount	Sparkfun	COM-0097	http://www.sparkfun.com/products/97

Table A-2. Shield Board Parts.

Table A-3 lists the parts for the Test Board:

1	Breadboard	Full sized solderless breadboard	Adafruit	239	https://www.adafruit.com/products/239
3	Screw Terminal, 3 pos	Connector terminal block, 3 position, 5.08 mm, solder straight through	Jameco	152355	http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=152355&catalogId=10001&freeText=152355&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView
1	Screw Terminal, 2 pos	Connector terminal block, 3 position, 5.08 mm, solder straight through	Jameco	152347	http://www.jameco.com/webapp/wcs/stores/servlet/ProductDisplay?langId=-1&productId=152347&catalogId=10001&freeText=152347&app.products.maxperpage=15&storeId=10001&search_type=jamecoall&ddkey=http:StoreCatalogDrillDownView
7	LED	Yellow LED - 5 mm	Sparkfun	COM-00532	http://www.sparkfun.com/products/532
0	resistor, 560 ohm	unit of 10 resistors, 560 ohm, 1/4 watt	Tayda Electronics	A-2282	http://www.taydaelectronics.com/10-x-resistor-560-ohm-1-4w-1-metal-film-pkg-of-10.html
1	Jumper Wires	Bag of 10, 6" M/M jumpers	Sparkfun Electronics	PRT-08431	http://www.sparkfun.com/products/8431
3	DS18B20	One-wire bus temperature sensor	Tayda Electronics	A-072	http://www.taydaelectronics.com/ic-integrated-circuits/temperature-sensors/ds18b20-1-wire-digital-temperature-sensor-ic-dallas.html

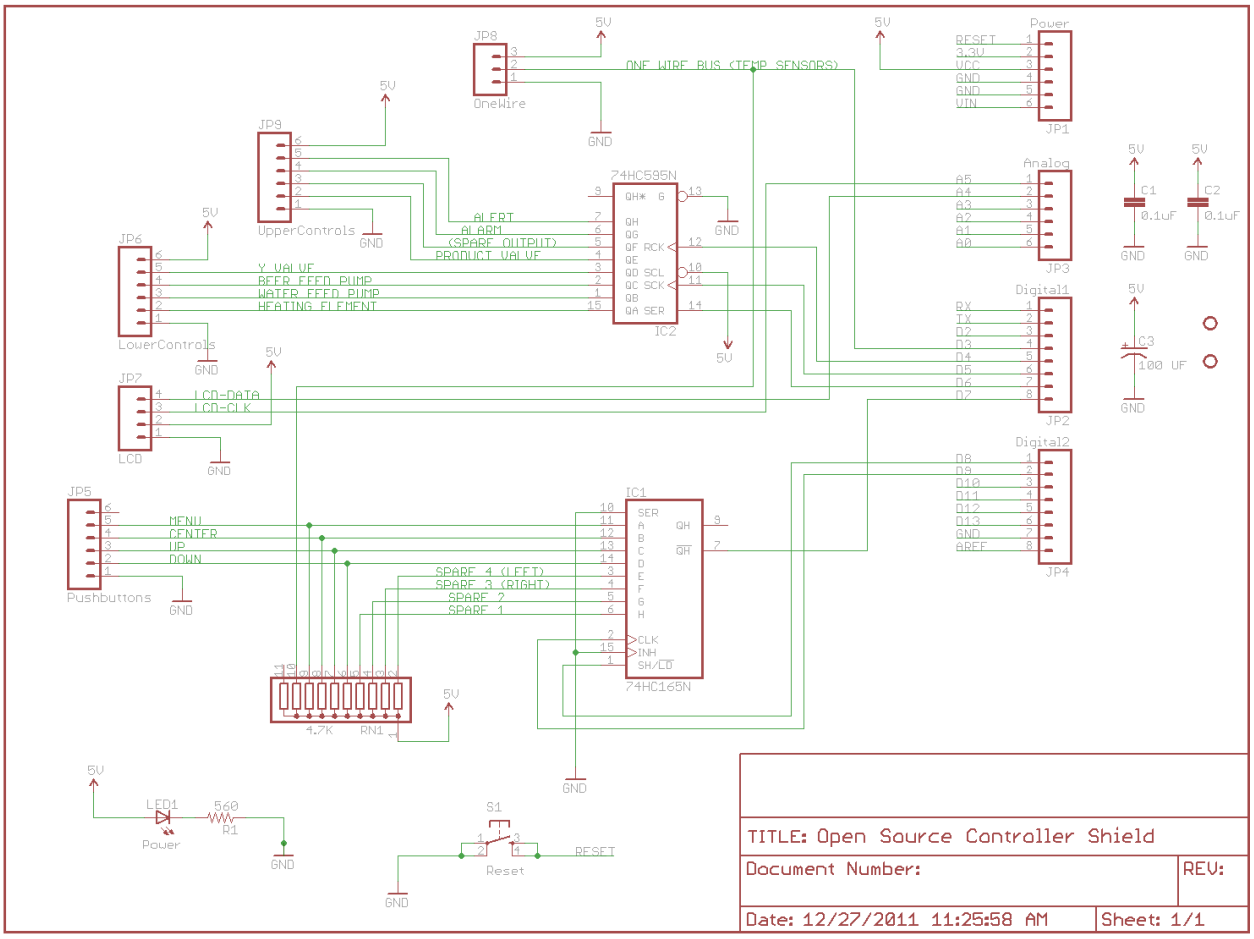
Table A-3. Test Board Parts.

Table A-4 list misc. parts and materials that you should have on hand:

1 USB Cable	USB Cable, A-B, 3 ft.	Adafruit	62	http://www.adafruit.com/products/62
1 Hookup Wire	22 AWG stranded, 25' ea. Red, green and black	Radio Sack	278-1224	http://www.radioshack.com/product/index.jsp?productId=2049745
1 Solder	8 oz roll of rosin core solder	Radio Shack	64-009	http://www.radioshack.com/product/index.jsp?productId=2062715
100 Cable Ties	Bag of cable ties	Jameco	70561	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_70561_-1?CID=PDF
15 Tie Mounts	Nylon cable tie mount, 3/4 inch, adhesive	Jameco	182942	http://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_182942_-1

Table A-4. Misc. Parts and Materials.

Appendix B. Shield Board Schematic.



Appendix C. Improvements from Thurmond Moore.

Thurmond Moore performed a test build of the Controller using the instructions in this manual. In so doing, he made a number of improvements that make construction and interconnection simpler. In summary, these are:

1. Using a nibbling tool (vs. a plastic cutting knife) to make the cut out for the LCD display. From section 4.2.1.
2. Using "Euro-style" pinch-down terminal blocks in lieu of the rotary screw terminal blocks. From section 4.3.4.

You may wish to substitute these improvements for the instructions in the referenced sections of this Manual.

Nibbling Tool:

Thurmond suggests using a sheet metal nibbling tool to make the LCD display cutout of section 4.2.1, vs. scribing the outline with a plastic cutting knife. He has used this technique to make the cutout in approximately 20 minutes. A suitable nibbling tool can be purchased in a hardware store or on-line at:

http://www.amazon.com/Parts-Express-Nickel-Plated Nibbling/dp/B0002KRACO/ref=dp_cp_ob_e_title_1

Figure C-1 is a photo of the nibbling tool cutting though the plastic of the Controller box cover.

Figure C-1. Nibbling tool positioned to nibble the plastic.

You can see from figure C-1 that the fit is tight, but it works. Drill a hole somewhere inside the rectangular cutout so that the nibbling tool head fits through. Then nibble to an edge, rotate the tool to nibble along the edge, and continue to nibble out the rectangular cutout. Thurmond supplied the photo in figure C-2 for the finished cutout:

Figure C-2. Finished LCD Display Cutout.

Careful use of the tool will result in a very good fit and very professional looking finished product, as shown in the photo in figure C-3.

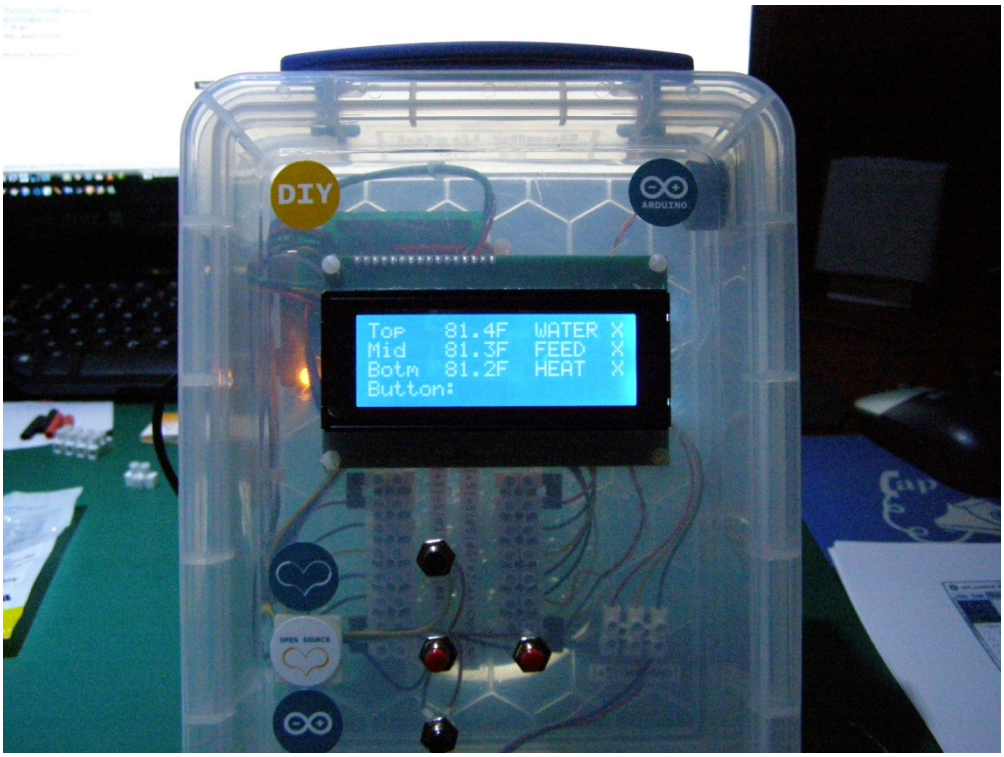


Figure C-3. Thurmond's Finished Controller.

Alternate Terminal Strips:

It was noted in section 2-1 of this Manual that the screw type terminal blocks tend to torque a thin gauge wire when tightening the screw and may result in breaking off the stripped end of the wire. Crimping a spade lug onto each wire was a recommended solution. This solution works very well, but Thurmond suggests an alternative solution that substitutes a pinch-down type terminal trip for these terminal blocks. Figure C-4 shows this alternative type of terminal strip.

Figure C-4. Pinch-down "Euro" style terminal strips and jumpers.

You can purchase these terminal strips and jumpers, on-line, at:

- ALTECH is the smaller strip MFG PN HEWPR/12 Allied Electronics PN 70077187 http://www.alliedelec.com/search/productdetail.aspx?SKU=70077187 ; Price $2.70
- Jumpers MFG PN HCL8-2 Allied Electronics PN 70077359 http://www.alliedelec.com/search/productdetail.aspx?SKU=70077359 ; Price $ .59
- WECO is the larger strip. MFG PN 324-HDS/12 Allied Electronics PN 70211849 http://www.alliedelec.com/search/productdetail.aspx?SKU=70211849 ; Price $3.39

The terminal strips come in 12-connection strips and can be broken into 8 connection and 4 connection pieces or just used as is. The terminal strips can be screw mounted to the bottom of the Controller box, or can be affixed to the Controller box using double sided mounting tape, such as 3M 4011.

Note: the Audible Alert module can successfully mounted using this same double sided tape, in lieu of gluing.

Figure C-5 is a photo of the inside of Thurmond's Controller, using these pinch-down terminal blocks and double sided tape mounting techniques.

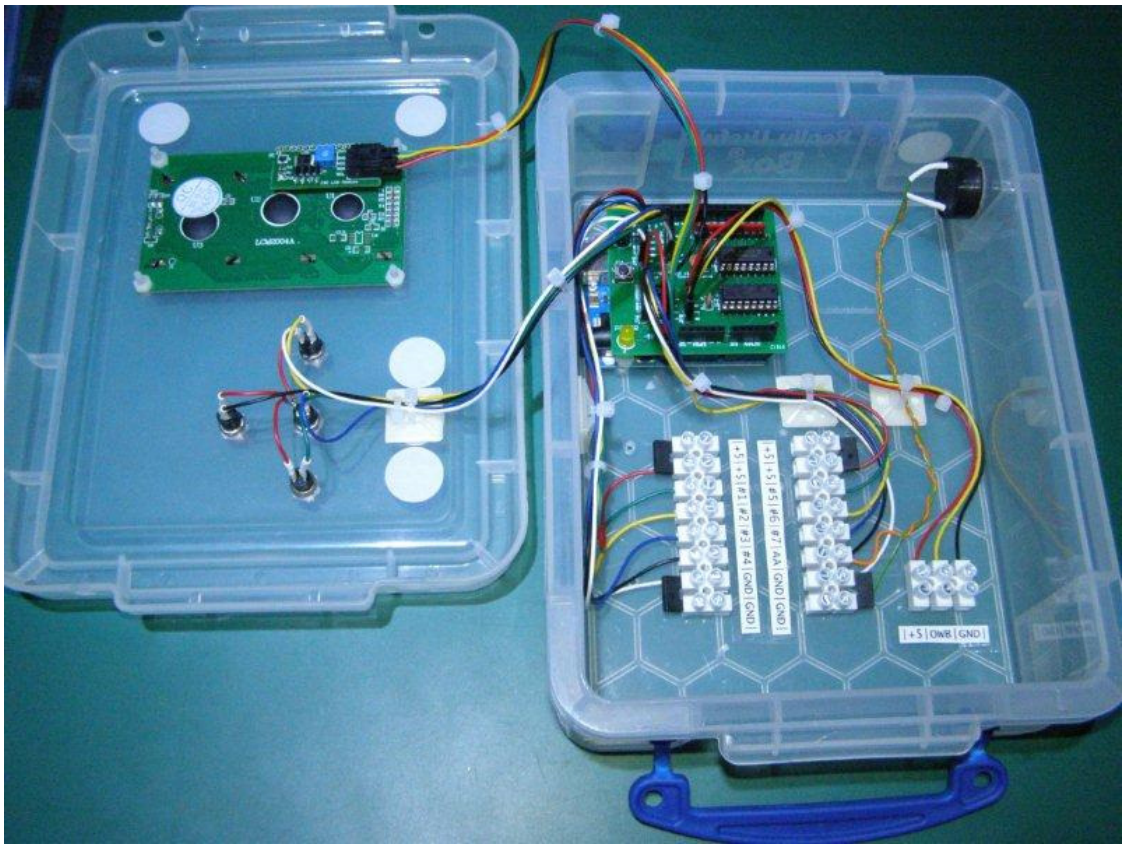


Figure C-5. Alternate Terminal Strips in the Controller.

Note from figure C-5 that Thurmond elected to mount the Electronic Module higher up in the Controller box so that the terminal strips are closer to wiring exit holes in the bottom of the box. This also means that the power connection/USB connection is higher up on the side of the box. Either the arrangement in section 4 of this manual or Thurmond's arrangement shown here will work well.

