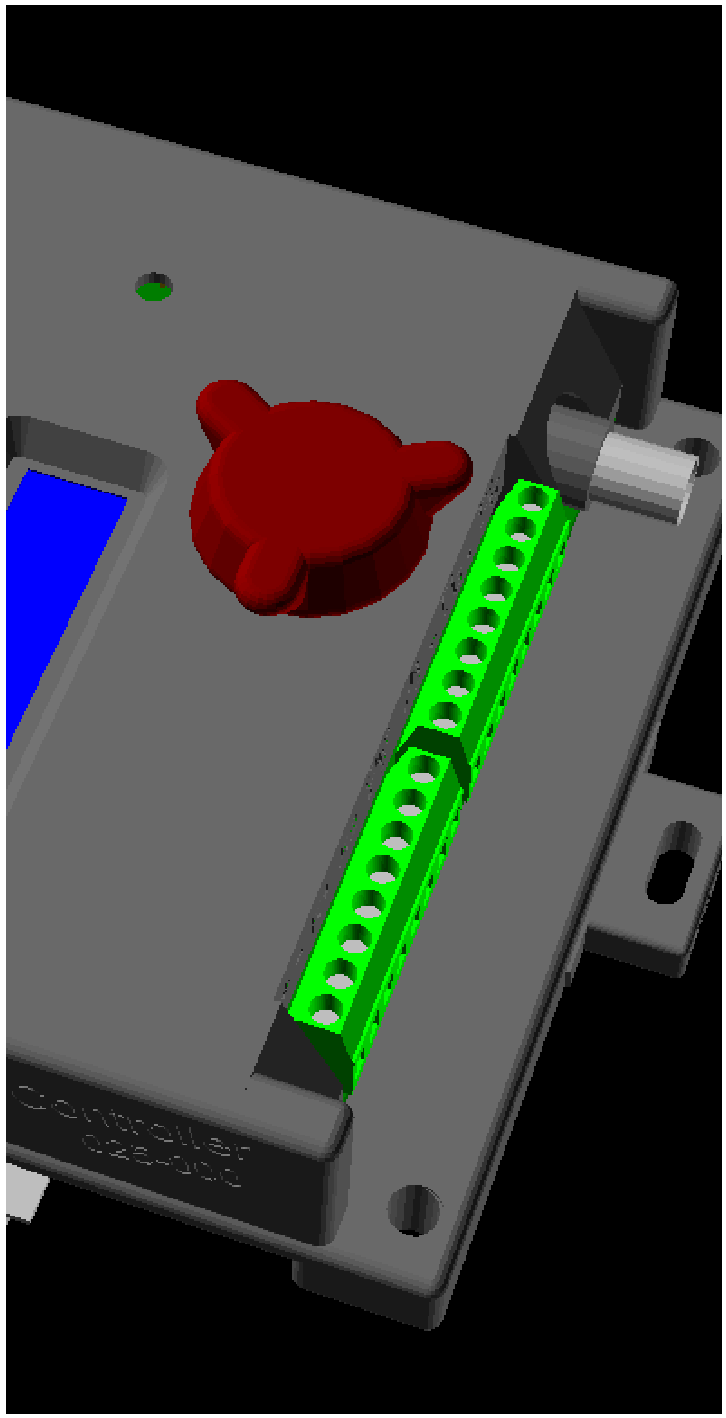
Symbrosia Controller
Hardware Manual

The Open Source
Aquaculture Controller



Andrew Cooper

Biosystems Engineer

Symbrosia Inc.

Table of Contents

1 Introduction

The Symbrosia controller was developed for in-house use as a lower cost alternative to the PLC based and commercial off-the-shelf systems we use for algae tank and bioreactor control. In addition to controlling the tanks there were a few other needs, places where a little remote control and monitoring is useful. With the availability of WiFi processors like the ESP32 automating these functions was a practical option.

Thus the Symbrosia Controller was born… An ESP32 wedded to a circuit board that contains the most commonly needed interfaces. A water quality sensor amplifier, some temperature inputs, a couple analog inputs, along with some relays and transistor outputs for controlling external devices like valves and pumps.

The available commercial controllers are expensive and tend to provide “closed garden” systems where it becomes difficult to retrieve data outside of dedicated applications and integrate the equipment into a wider system. While a number of similar open solutions do exist, none of the devices we have seen are fully developed into a robust, industrial solution. The Symbrosia controller offers a number of key advantages…

- Designed from the ground up as a network appliance with full WiFi access.
- A standard industrial ModbusTCP interface, a well established interface supported by a vast array of software and solutions available in the market.
- A simple and effective user interface allowing access to key settings.
- A well thought out and flexible control arrangement allowing automated control of pH, temperature, or other parameters. Four software controllers are available that will take any sensor reading and take action with controllable setpoints.
- One fully isolated water quality amplifier (for pH, ORP, dissolved oxygen, etc.), two Pt100 temperature inputs, two analog inputs, two relay outputs, and two digital outputs are available. The two analog inputs may be used with 0-5V, 0-10V, or 4-20mA inputs, these can also be used as simple digital inputs.
- Good calibration solutions in place for measured values. Analog inputs can be two point calibrated with offset and gain.
- Semi-automated pH calibration built into the software for use with 7ph, and 4pH or 10pH reference solutions. Very simple temperature offset calibration with the front control dial.
- A complete solution ready for deployment in real applications. Use it as designed and published, or change the hardware or code to suit your own needs.

Communication is via 2.4GHz WiFi, thus no cabling beyond power, or perhaps your sensors is required, a very real concern when salt water is involved and locations are scattered across a large facility. 2.4GHz offers much increased range for applications that do not require the high bandwidth of higher frequencies.

For remote monitoring a ModbusTCP interface is implemented. This allows easy integration with custom software or standard industrial SCADA infrastructure. All SCADA software can interface with Modbus devices. While most programming languages like Python have ModbusTCP libraries available to allow connections with a few lines of code.

There are many options for designing and building a device like this. The philosophy used in the design was carefully considered…

- The controller is designed to be constructed and maintained by anyone with modest electronics knowledge and skills. Unless changing features is desired, no programming is required to set up and use the controller.
- Complex solutions or specialized parts were avoided. Rather parts that are commonly available, inexpensive, and likely to remain available for some time to come were chosen for the design.
- An open source programming platform is used so that development tools are available at no cost, specifically the Arduino platform.
- An open source schematic capture PCB CAD package was used, specifically KiCad. The CAD files may be used or edited at no cost beyond learning to use the software.

An example of the commonly available part philosophy can be found in the display. There was great temptation to use a newer OLED color graphical display that would be both fun and attractive. Rather a mundane 16x2 LCD character display was chosen. The LDC display is very standard, inexpensive, robust in outdoor settings, available from multiple vendors, and is likely to remain available for a decade or two. Still, a version based on an OLED display may exist someday just for the fun of it.

Another good example of this is in using a development module for the ESP32 rather than laying out the ESP32 and support electronics directly onto the main PCB. While much of the design is surface mount, larger packages were used, this allows the entire PCB to be assembled by hand with no more specialized equipment than a decent soldering iron.

Likewise using the Arduino programming platform provides an approachable simplicity that is easily learned. A huge array of support exists on the web with a robust development community. Libraries to support Arduino are available from many sources that allow the use of a wide array of sensors, chips, or ready to use modules.

With the controller proving its usefulness across our algae production facility a further decision was made… To release the controller to the community as an open source device. Symbrosia is publishing the complete plans… Schematic, BOM, PCB, 3D models, code, and user manual.

There is nothing exotic or particularly special here. The Symbrosia Controller is the result of taking a number of parts and solutions that are available in the community and weaving them into a well thought out and robust solution. The controller has been a good solution for us, maybe you can use it as well.

The Symbrosia Team

2 License

The Symbrosia Controller has been released under the terms of the GNU General Public License version 3 or GPLv3. Full text of the license can be found in the documentation package or at https://www.gnu.org/licenses/gpl-3.0.en.html.

The primary points of this license are that…

- Symbrosia Inc owns and maintains the copyright to the Symbrosia Controller documentation and software. Instead of using those rights to restrict users, we use them to ensure that every user has freedom.
- Anyone may freely use, build, or modify the Symbrosia Controller for their own use, either privately or within a commercial venture.

The GNUv3 public license upholds the ideals of open source. These ideals are clearly stated in the GNUv3 documentation and we at Symbrosia support them.

- The freedom to use the design for any purpose,
- The freedom to change the design to suit your needs,
- The freedom to share the design with your friends and neighbors, and
- The freedom to share the changes you make.

You are reminded that use of the Symbrosia Controller represents consent to the licensing terms embodied in GPLv3.

2.1 Software and Libraries

The Arduino framework is used in compiling device firmware and is released under the GPL. The terms of this license are clear and require Information on this license can be found at https://support.arduino.cc/hc/en-us/articles/4415094490770-Licensing-for-products-based-on-Arduino.

The various libraries used in compiling the Symbrosia Controller firmware feature other software licenses. These are permissive and compatible with the GPL license of the Arduino system. A complete listing is shown below…

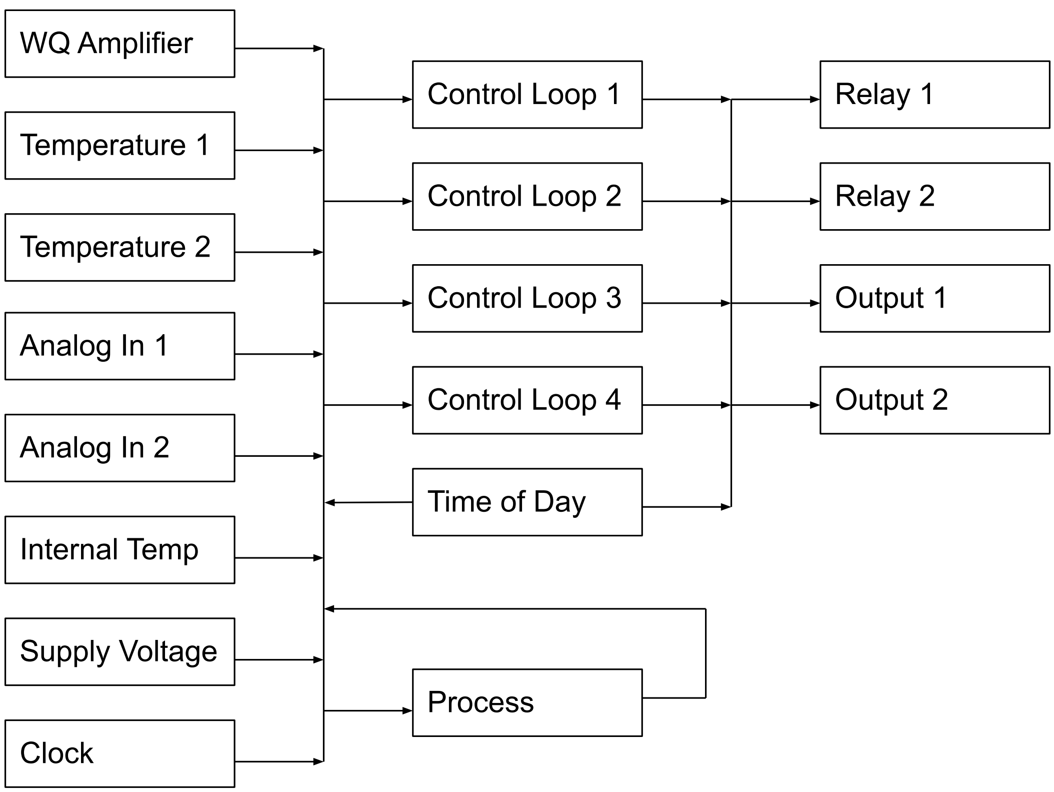
- ModbusTCP by Alexander Emelianov is licensed under the permissive 3-Clause BSD License, information can be found at https://opensource.org/license/bsd-3-clause.
- ESP32Time by Felix Biego is licensed under the permissive MIT license, information can be found at https://en.wikipedia.org/wiki/MIT_License.
- ESP Rotary by Lennart Hennings under the MIT license, information can be found at https://en.wikipedia.org/wiki/MIT_License.
- Button 2 by Lennart Hennings under the MIT license, information can be found at https://en.wikipedia.org/wiki/MIT_License.
- Adafruit Neopixel from Adafruit provided under the GNU Lesser General Public License as published by the Free Software Foundation, information at https://www.gnu.org/licenses/lgpl-3.0.en.html
- MCP_ADC by Rob Tillart released under the MIT license, information can be found at https://en.wikipedia.org/wiki/MIT_License.

3 Overview

The basic logic of the controller may be a bit daunting at first, but is really quite simple. An array of inputs are available, along with four outputs that can control devices outside the controller, notably valves, lights, or solenoids.

Four control loops allow decisions to be made on the inputs with a setpoint, hysteresis, alarms, etc. Any control loop can use any input and set any output. These are simple on-off controllers with hysteresis, sometimes called bang-bang controllers.

Setup of the controller is done by configuring the sensors, then connecting these sensors to an output through a control loop to accomplish the desired control. All configurations are by means of Modbus registers, indeed the status of all inputs, control loop settings, and the state of all outputs are available to remote software in Modbus registers.



For example, temperature control of a tank may be accomplished by setting control loop 1 to use temperature 1 as an input, comparing to the desired setpoint, then switching relay 1 on to activate a heater.

While simple on-off control is implemented onboard, complex control solutions can be implemented using remote software. In this mode the SymbCtrl can provide input data via one or more of the sensors while outputs may be directly commanded via the remote software.

3.1 Inputs

The Symbrosia controller has five analog inputs used to connect sensors to the unit.

A water quality sensor amplifier is used for pH, ORP, dissolved oxygen, or similar sensors that provide a small voltage (usually less than +/- half a volt). This amplifier is completely galvanically isolated to eliminate the effects of stray voltages, particularly an issue in salt water media.

Connections for two Pt100 temperature sensors is included due to the importance of temperature in most aquaculture applications. Pt100 sensors were chosen as they are widely available and offer accurate temperature readings with excellent long term stability. A three wire connection is used to remove the effects of long cables

Provision has been made to average readings of any two sensors together or select one or the other on a priority basis.

Two additional general purpose analog inputs allow connection of 0 to 5V, 0 to 10V, or 4 to 20mA sensors. A jumper location is used on the PCB to convert these inputs from voltage to current inputs. These inputs can also be used as digital inputs with a 5V threshold if desired.

Supply voltage and internal temperature are also available to be monitored or used by any of the control loops. Monitoring supply voltage may be particularly useful for solar/battery operation.

All sensor inputs may be monitored via software across the network through Modbus registers. This allows data logging of any aspect of controller operation.

3.2 Outputs

Control of devices such as valves, pumps, or heaters can be done using one of the four outputs. Two relays and two low-side transistor switches.

Two of these outputs are normally open relay contacts that can switch modest AC and DC loads up to 5A @ 120Vac. These output contacts are completely isolated from the internal controller circuitry thus can be used to switch remote devices without concern for ground loops.

Two low side transistor driver outputs allow control of external relays, indicators, warning lamps, or similar equipment. These outputs are low-side open collector switches, they connect the output terminal to controller ground when active. Loads up to 1A and 40Vdc can be controlled by connecting the positive lead of the load to any voltage up to 40V, then connecting the negative side of the device to the controller output terminal.

3.3 Auxiliary Functions

A processed analog value may be created from any two other analog values, summing, averaging, or selecting the minimum or maximum value. The result of this processed value can then be used by any of the control loops..

A Time of Day function allows turning on or off any output, or enabling any of the control loop outputs, thus allowing control for part of the day.

A counter is available to count the number of times any output is activated, likewise a timer is available to time how long any output is active. The counter and timer have proven quite useful for tracking CO_2 or water usage and other experimental needs.

3.4 Configuration

Due to the choice of ModbusTCP as the network control protocol configuration is handled through the Modbus register model. Each setting, sensor value, and status bit is a Modbus register. This has the power of providing a very industry standard interface. This also makes remote access to practically every sensor reading, control state, and output quite simple.

The register model can be slightly cryptic as the correct values must be loaded in the correct registers. Using a setup utility such as the supplied SyView Python script allows configuration without having to look up each of the registers.

In a commercial device this complexity would be hidden from the user, wrapped behind a computer program or a cell phone app. The symbrosia controller is an open device where access is provided to all of the normally behind-the-scenes stuff. You have your choice… Use SyView to configure the controller quickly, or get into the works and do it yourself with few limitations.

SyView also allows the saving of a controller's entire configuration to a file that can then be downloaded to other units. Thus a standard setup may be easily used for multiple units.

3.5 Power

The controller can operate on any DC voltage from 9Vdc to 36Vdc. Reverse polarity protection is provided. For supply voltages above 12V a switching regulator module is recommended in place of the linear regulator U2, see the bill of materials for more information.

4 Construction

The controller should cost about $70 in parts to construct if produced in small volume, with parts purchased for around a dozen controllers at a time. If you include the cost of a small weatherproof enclosure, power supply, and various hardware, the total cost should be around $250 per installation plus the cost of any sensors or valves you might need.

4.1 Purchasing Parts

A wide array of online sources are available to purchase your electronics parts from, this can be rather confusing to a first time buyer. In the US the largest suppliers that sell small quantities are Digikey, Newark, Allied Electronics, and Mouser. In Europe Farnell is the largest small quantity catalog seller. All of these sites offer excellent website catalogs, easy ordering, and extensive in-house inventories that ship quickly.

4.1.1 Digikey

The largest electronics distributor that sells small volume orders in the US is DigiKey. During development of the Symbrosia controller most parts were ordered from DigiKey.

Many electronic parts are subject to steep volume discounts as more parts are ordered, this is particularly true with electronic suppliers like Digikey. These discounts begin to apply as 10 or 50 of each part are ordered. Thus it may be much cheaper to order 100 or more of the low cost parts like resistors and diodes.

Be aware that the electronics part marketplace can change quickly and that for most of the components used in the Symbrosia controller there are many possible substitute parts. Parts like resistors and capacitors, and even some of the IC’s, can often be easily substituted in case of trouble ordering the specified parts found in the published BOM.

4.1.2 eBay Parts

Several of the key parts are available on eBay, particularly the displays and ESP32 modules. Quality is of course always suspect on parts sourced from some eBay vendors. If you are knowledgeable about what is being purchased you can save some money, but not really that much if ordering in modest quantities to take advantage of volume discounts. Otherwise, it is probably best to source parts from a reputable electronics vendor like DigiKey or Mouser.

4.1.3 PCB

Having custom PCB’s manufactured is surprisingly inexpensive and quite easy. You simply need the Gerber files that define the PCB patterns, these are provided as part of the open source documentation. We have purchased these boards from Seeed, JLC PC, and OSH Park with no quality issues.

When ordering a few additional bits of data will be required. These are listed in the table below.

PCB Size:	120mm x 92mm
PCB Thickness:	1.6mm (0.062”)
Layers:	2
Material:	FR-4
Copper:	1oz
Finish:	HASL (hot air leveled solder) or ENIG (nickel and gold plating)
Minimum drill size:	0.3mm
Spacing:	6/6mil
Silk Screen:	Top and bottom
PCB color:	Pick something fun

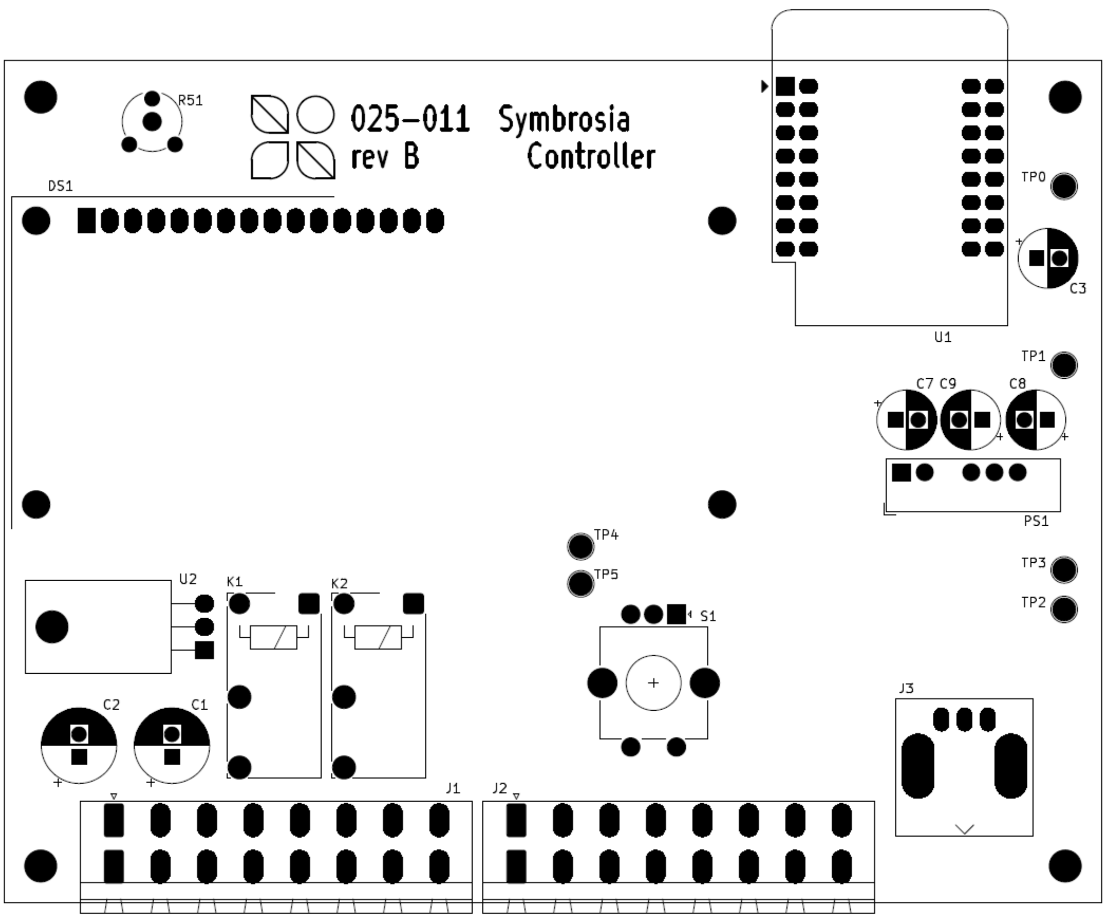
4.1.4 Partially Assembled PCBA

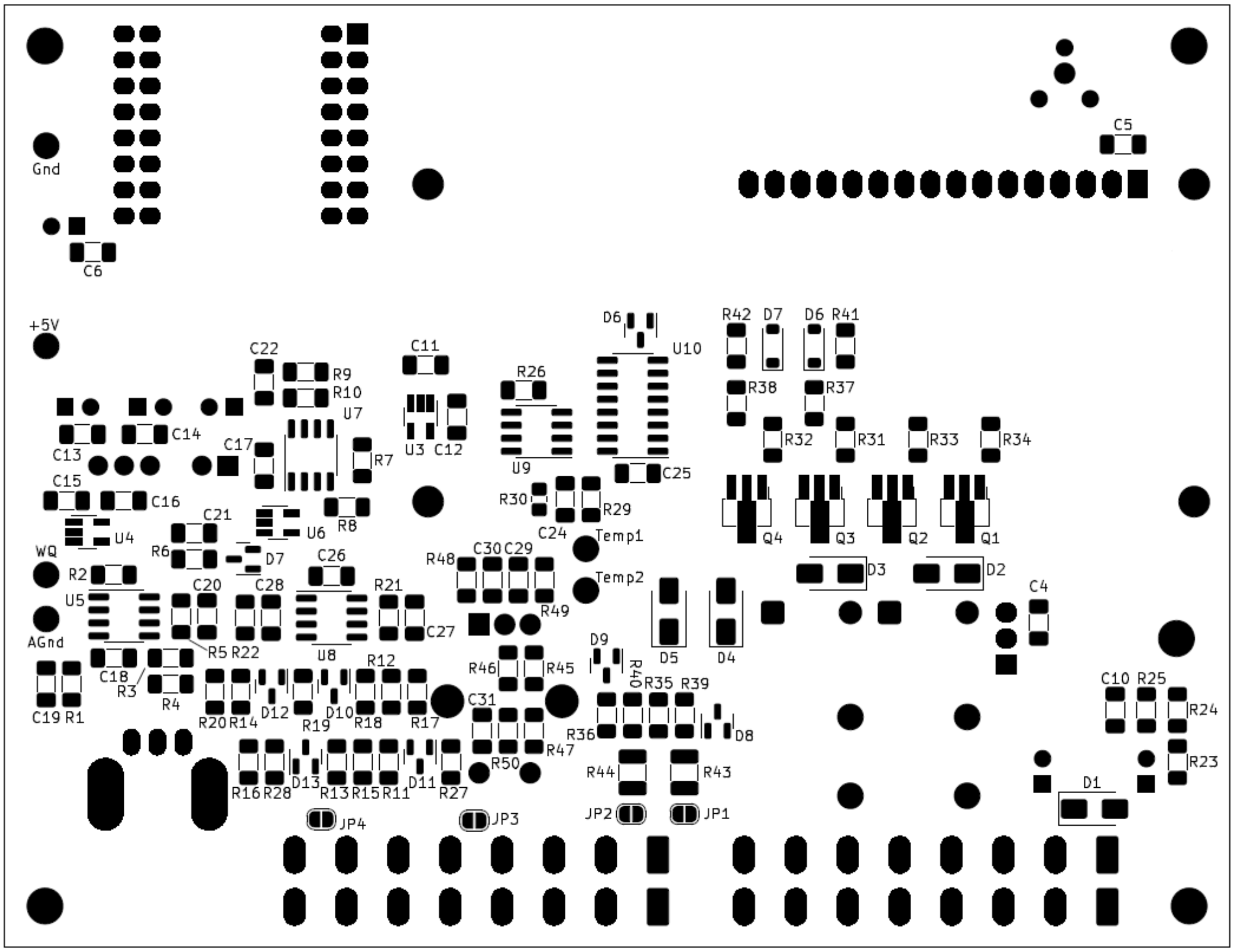
It is also possible to order mostly assembled PCBA from a board house. Symbrosia has been ordering boards with all of the surface mount parts in place from JLCPCB for less than $20 each. This vastly simplifies ordering and reduces the labor needed to build controllers. All that is required is to solder the few through hole parts needed for the version desired.

4.2 PCB Assembly

The only equipment necessary to assemble the Symbrosia controller is a decent soldering station and the usual small hand tools such as tweezers. Some experience in soldering electronic parts is recommended, but it is not too difficult to learn as you go. A good 10x magnifier is also useful for inspecting your soldering.

If in any doubt of your ability to perform this soldering simply purchase an extra PCB or two and some extra components and use these for practice. There are plenty of good websites or YouTube videos that can assist you in learning these useful skills.





4.2.1 SMD Parts

The controller's compact design is a result of using surface mount devices (SMD) for the resistors, capacitors, diodes, and IC's. These parts are also cheaper, and often more readily available than equivalent through-hole components. A number of the IC's used are only available in surface mount.

To allow for easier SMD soldering only the larger SMD parts such as 1206 and SOIC packages were chosen for the design. With a little practice these parts are easily hand soldered.

4.2.2 Connector Mounting

The connectors J1, J2, and the BNC connector J3 can be mounted on either side of the circuit board to accommodate different mounting solutions for the PCB. To use the SymbCtrl DIN case the connectors should be mounted on the PCB front as shown in the above assembly drawings. In a panel mount solution, such as some of the Serpac cabinet frames included with the design package, connectors will need to be mounted on the reverse of the PCB to provide good access to these connections. Reverse mounting does have the effect of reversing the pin order of the terminal blocks J1 and J2, while having no effect on the BNC connector.

Care should be taken to determine if you need the connectors front or reverse mounted for your situation. They can be de-soldered and moved, but doing so is quite annoying and risks damage to the circuit board.

4.3 Assembly Instructions

Soldering the PCB assembly is easiest in the order specified, generally placing the smaller SMD parts first, with the largest parts such as the LCD display and the rotary encoder being the last parts soldered.

1. Locate and solder the SMD resistors, capacitors, and diodes according to the BOM and parts layout
2. Solder the transistors Q1 to Q4
3. Locate and solder the U3 to U6 integrated circuits
4. Place and solder the electrolytic capacitors C1 to C3 and C7 to C9.
5. Solder the terminal blocks J1 and J2
6. Solder the header for the ESP-32 U1
7. Attach the regulator U1 with a heatsink using a small dab of heatsink paste between the regulator and the heatsink. Once mounted, solder and trim the regulator leads.
8. Solder the capacitors C1 and C2
9. Solder the header for the display DS1
10. Solder the BNC connector J3.
11. Solder the relays K1 and K2
12. Solder the rotary encoder S1 to the PCB.
13. Carefully inspect the entire PCB with magnification for good solder joints, any missing components, etc.
14. Clean any excess flux from the PCB with isopropyl alcohol and a small brush such as a toothbrush.
15. Attach the display with the M2.5 standoffs and screws and insure the display is level to the PCB before soldering the display header
16. Do not insert the ESP-32 until after initial power up testing

4.4 Inspection and Testing

It is recommended that the ESP-32 be programmed and tested (see section 3 on programming) before installation into the PCB.

4.5 Conformal Coating

After the PCB has been constructed and fully tested, an optional, but recommended step is conformal coating to protect the PCB from moisture and salt when used in the damp environments typical of aquaculture.

A number of different conformal coatings are available for use on printed circuit boards, generally acrylic, urethane, or silicone based products. As silicone provides better protection against salt spray and moisture it is recommended for the Symbrosia controller. Products like ACL Staticide 8695 or TechSpray Fine-L-Kote SR should provide good protection for the PCB.

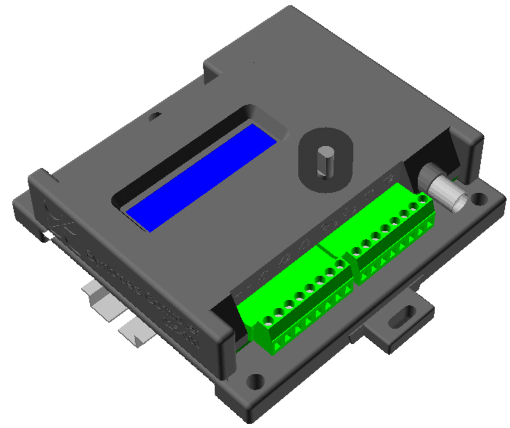
A budget alternative to a commercial conformal coating material is quick drying enamel, specifically inexpensive clear fingernail polish as can be purchased at drugstores and some supermarkets. This can usually be purchased for less than $5 a bottle, with one bottle providing enough for several PCB’s.

Conformal coating does make repair or modification of the PCB more difficult as the coating must be removed from the rework site. For this reason coating should only be applied once the PCB is fully tested and configured.

The coating should be applied to the bottom side of the PCB to cover the bulk of the SMD components and wiring. Care should be taken to avoid applying the coating to the LCD display and the connectors. No need to coat the entire PCB, just the exposed components and traces.

4.6 3D Printed Case

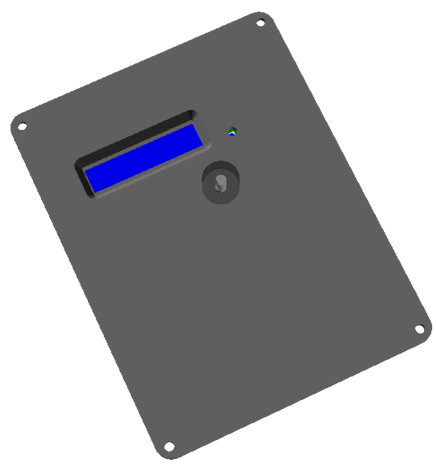
In the spirit of DIY the case is designed to be 3D printed.



No specialized printer is needed, rather a standard commercial FDM 3D printer using PLA, PETG, ABS, or ASA material will serve nicely. Both the STL and OpenSCAD source files are provided as part of the open source release. You will have to slice the design in your favorite slicer software for your particular printer.

Most of these parts can be printed using a smaller printer with a 200mm x 200mm build plate. Some of the supplied files require the use of a larger 250mm x 250mm build volume.

These cases are not waterproof, it is recommended that the controller be mounted in a weatherproof cabinet for outdoor applications. This cabinet can also house the power supply and any sensor electronics in a protected space. DIN rail mounting of the controller and terminal blacks to provide rugged connection points is also recommended. Fiberglass or polycarbonate enclosures are recommended as metal enclosures will inhibit WiFi signals.



In order to facilitate weatherproof installation 3D models for mounting the controller into two different size Serpac waterproof enclosures are provided as part of the document package, the Serpac model I142 and I342 cases. These allow mounting of the controllers into these cases providing a neat, weather tight solution for outdoor use.

If another model of waterproof case is desired take a look at the supplied designs for use as a starting point.

In the 3D source files you will find models for the SymbCtrl PCB and some of the usual accessories like gas valves and flow gauges. Use of these models will make designing far easier.

4.6.1 Material

The original SymbCtrl cases were printed in PLA. This material has served well in some environments without trouble. For situations where the case will be exposed to higher temperatures or more UV light a high temperature material such as ABS, PET-G, or ASA is recommended. We have had PLA parts soften and distort when in enclosures outside in the hot Hawaiian tropical sunlight and now use ABS for all parts.

4.6.2 3D Printer Settings

You will need to slice the STL files with settings for your own printer, in most cases your default settings will serve well.

- Material: Standard ABS, PET-G, or ASA (avoid PLA for outdoor applications)
- Nozzle: 0.4mm
- Layer height: 0.2mm
- Support: Some parts require support, see documentation
- Infill: 20%
- Walls: 6
- Top and Bottom Layers: 6

It is recommended to print the parts with a very standard 0.4mm nozzle, the default nozzle supplied for most printers, modest infill and thick walls. If you are skilled in 3D printing you may of course experiment with other settings to your taste, but these are the settings used successfully to this point.

A number of heat set brass inserts are used to create the threaded holes for PCB mounting screws and the case screws. These may take a little practice to learn to set them properly.

5 Programming

Editing and programming the ESP32 S2 controller requires an integrated development environment (IDE). To make this simple all firmware for the controller was developed with the widely available and free Arduino IDE 2. This is a well supported IDE with a vast amount of community support and use instructions available on the net.

In addition to the Arduino IDE several other options exist that support the Arduino framework. This includes the popular Platform IO environment that can be run with a number of other IDE's on any of the common operating systems. The Symbrosia controller has been programmed without issues on the Atom IDE using Platform IO. Generally the only change needed to the firmware is changing the extension of one file, SymbCtrl.ino is changed to .cpp to fit the usual C++ standard..

The instructions below cover using the Arduino IDE to program the ESP32 microcontrollers.

5.1 The Arduino Programming Environment

The Arduino IDE can be downloaded from the official Arduino website at https://www.arduino.cc/en/software, either the classic version or the newer IDE 2 will work properly. We suggest installing and using the IDE 2 as it has become the standard.

Once installed you can familiarize yourself with the IDE if you are not already familiar with it using the training material provided by Arduino at https://docs.arduino.cc/software/ide/.

5.2 Loading the Libraries

A number of libraries are required to build the Symbrosia controller sketch. All of these libraries are available in the official Arduino channel and can be installed through the library manager in the IDE.

Select the library manager on the sidebar and search for each of the libraries listed below. Be aware that there can be several libraries with similar names, ensure the version below is used. The revision should be the same or later than the revision number shown…

- LiquidCrystal by Arduino and Adafruit, v1.0.7
- ESP32Time, ESP32 RTC time support by Felix Biego v2.0.6
- ModbusTCP, Modbus TCP support by Alexander Emelianov v4.0.0
- ESP Rotary, rotary encoder support by Lennart Hennigs v1.4.2
- Button2, pushbutton support by Lennart Hennigs v1.6.5
- Adafruit Neopixel, support for the RGB neopixel LED v1.10.4 found on the S3 mini
- The MCP3021 AtoD is handled by MCP3X21 by Pavel Slama v1.0.1
- The MCP3208 AtoD is handled by MCP_ADC by Rob Tillaart v0.5.0

The Symbrosia controller firmware also required the following libraries, these should be installed by default with a standard Arduino IDE install…

- Arduino
- WiFi, v1.2.7
- Wire, v2.0.0
- SPI, v2.0.0
- EEPROM, v2.0.0

5.3 Downloading the Source Code

Current source code for the Symbrosia Controller can be found in our GitHub repository at https://github.com/symbrosia-co/Symbrosia-Controller.

Place the unpacked sketch folder in the Arduino project directory, on a Windows machine this is usually C:\Users\<username>\Documents\Arduino, the sketch will then appear in the Arduino IDE.

5.4 Set the Module Type

Currently two processor modules are supported, these are the Lolin ESP32 S2 Mini and the Lolin ESP32 S3 Mini. It is necessary to select one of these before compiling and downloading the code to the module.

Defining the module type is done by commenting out the module not being used and ensuring the correct module is not commented. Note lines 24 and 25 in the figure below define the module being used. The double slash is used to comment the line, thus the example below is set to use the ESP32 S3.

```
21  ---------------------------------------------------------------------------*/
22  #pragma once
23
24  //#define hardwareS2Mini 2
25  #define hardwareS3Mini 3
26
27  #ifdef hardwareS2Mini
28  #include "hardwareSymbCtrlS2Mini.h"
29  #endif
30
31  #ifdef hardwareS3Mini
32  #include "hardwareSymbCtrlS3Mini.h"
33  #endif
34
35  // unit ID
36  #define modelNumber   2
37  #define serialNumber  220
38  #define firmMajor     2
39  #define firmMinor     4
40  #define modelNameStr  "SymbCtrl Mk2"
41  #define timeZone      -10
42
43  // alarm points
44  #define alarmVSupLow  8  // low supply voltage alarm point
```

5.5 Set the Unit Serial Number

The serial number is found in the globals.h file. Assign a serial number and edit the file to place this serial number into the unit upon programming. Note line number 37 in the figure above where the serial number is set to 220, edit line 37 as needed.

5.6 First Power-Up

The unit can run on just the 5V power from the programming USB cable. Of course some outputs such as the relays will not operate, but the display, sensors, and input encoder will. This allows most programming and testing of the unit with only USB power.

6 Installation

6.1 Enclosure

As mentioned above a weather tight enclosure is recommended. We have used the 12” x 10” Serpac I342HL cabinet for many of our installations. This case is available for order from home centers like Home Depot for $85 (free shipping with in store pickup). The case is large enough to easily protect the controller, a 12V power supply, and all connection points for sensors. The case is all plastic and survives the salt water environment on our ponds. If there are no valves, relays or other equipment to be added a smaller case may suffice, the Serpac I152HL is a 10” x 8” case that can be purchased for about $40.

All entries into the cabinet are protected by cable glands that seal neatly around the cables while keeping it easy to change sensors when needed.

The controller may be mounted to a DIN rail or bolted into the rear panel. We generally use the 3D printed panel inserts that fit these Serpac cases. A few DIN rail mount terminal blocks allow easy and rugged connections to be made in the field, though do increase the time needed to build each control cabinet.

6.2 Power

A 12Vdc power supply is recommended for operation, connected to terminals 1 and 2 of P1 as shown in the connection diagrams. As long as power is applied the LCD backlight will be on, this doubles as the power indicator. The controller is protected against reverse polarity on the supply.

A 2A power supply is recommended, potentially larger if additional power is needed for accessories such as valves and pumps. The unit itself uses less than 250mA without anything connected. This supply may be DIN rail mounted beside the controller or closed frame supplies inside an enclosure. It is recommended that the power supply be purchased from a reputable vendor as a great deal of questionable or even downright dangerous supplies exist on sellers like eBay. These supplies may scrimp on safety features and isolation, a possibly dangerous issue around water. Good supplies can be found on Digikey and the other distributors for about the same price as less reputable sources. Please refer to the bill of materials.

If larger relays, lights, or contactors are required, or any other accessories that require more power a larger supply may be purchased. If the 12V is not needed for anything other than the controller itself only 250mA or 3W is required.

Upon applying power there is a startup sequence. The unit will attempt to log into a wireless network, retrieve network time, and begin readings of the various sensors. A startup lockout of about 30 seconds prevents any inadvertent operation of attached devices (valves, pumps, etc.) until after readings of the various sensors are stable.

6.3 Solar and Battery Power

Solar and battery powered systems are supported through several features. The controller will operate properly on an input range of 9-15Vdc, this covers the recommended range for operating deep cycle lead acid batteries and the 3 cell lithium battery packs designed for similar use. Supply voltage is measured and can be read out on the ModBus connection allowing battery performance and state of charge information to be remotely monitored.

Addition of a current monitor or hall effect current sensor to the battery circuit would allow measurement of the charging currents via one of the analog inputs, thus providing a complete solution.

6.4 Connecting Sensors

6.4.1 WQ Sensor

The WQ sensor input supports a wide range of typical aquaculture sensors including pH, ORP, dissolved oxygen, etc. These sensors generally provide a voltage output of around ±0.5Vdc. The software will then convert this reading into the units selected.

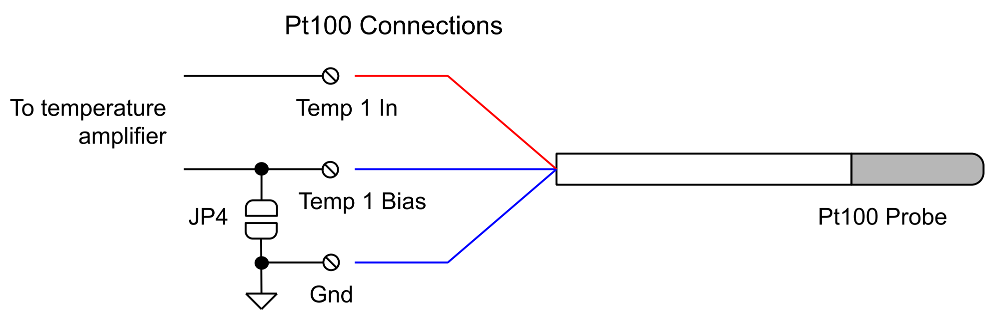
Most of these probes traditionally use a BNC connector which can be directly connected to the controller. If a probe with another connector is to be connected, such as SMA, inexpensive adapters are available to convert to BNC.

If sensors with 4-20mA or 0-10v outputs are to be used, these should be connected to analog inputs 1 or 2.

A software assisted calibration is provided for pH sensors allowing quick calibration with just a few button presses. See the pH calibration instructions for more information.

6.4.2 Temperature Sensors

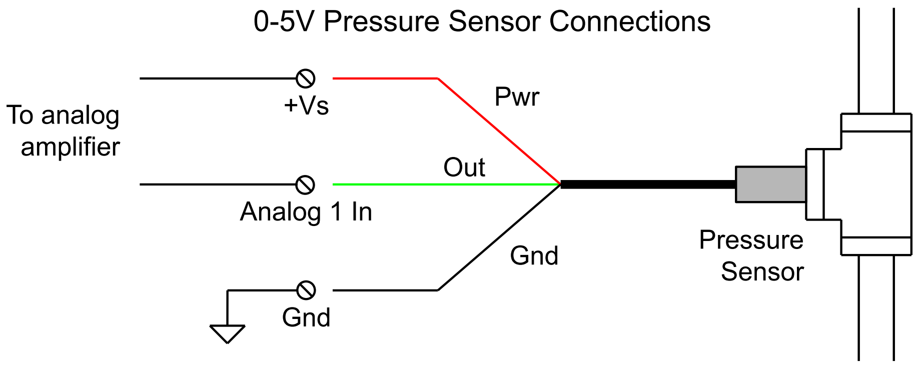
The symbrosia controller uses standard Pt100 temperature sensors. Pt100 sensors are connected to one of the two dedicated temperature sensor inputs. The readings of these two sensors may be used individually or averaged to provide redundant control.



The Symbrosia controller supports use of three terminal connections for Pt100 sensors, using this arrangement the resistance of long cables can be corrected. If a two wire Pt100 sensor is used the sensor bias and sensor ground terminals must be connected with a jumper. This can be either a wire jumper placed in the terminal block, or using a solder jumper (JP4 and JP5) provided on the circuit board.

6.4.3 0-5V and 0-10v Analog Sensors

An extraordinarily wide range of sensors are available that provide either a 0-5V or 0-10V output. Sensors for pressure, humidity, weight, pH, flow, solar intensity, or just about anything you can think of are available and can be connected using the analog inputs. Offset and gain scaling in software allow conversion of the reading into engineering units.

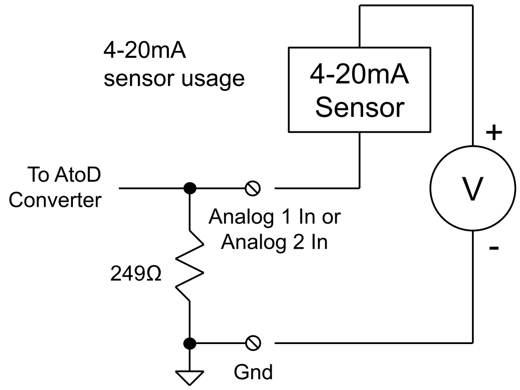


For example, a 100PSI pressure sensor with a 0 to 5V output is desired where 5V represents 100PSI. Connecting this sensor to Analog Input 1 (as shown) and entering a gain of 20 in Analog 1 Gain, will convert the number to PSI. Additionally, setting Analog Input 1 Units, to 16 will display all readings with the units of PSI.

6.4.4 4-20mA Analog Sensors

The controller supports 4-20mA sensors by use of a 249Ω resistor found on the PCB that converts the current signal into a voltage signal for the controller to measure. The resistor must be connected into the circuit to use 4-20mA signals, this is accomplished by soldering across the jumper JP1 or JP2 of the PCB as needed.

The controller will need to be the last connection in the 4-20mA chain as the other side of the resistor returns to controller ground.



The 4-20mA sensor supply voltage will need to be at least 5v plus the voltage needed by the sensor as found in the sensor data sheet. For example… If using a 12V supply for the controller, this supply may be used as long as the sensor will operate properly with a minimum of 7V. Since sensors requiring 5V are quite common the 12V controller supply could be used to power the sensors on shorter cable runs.

6.5 Connecting Outputs

Some care and understanding is useful when connecting the output channels. Pay attention to the maximum ratings to prevent damage to the controller.

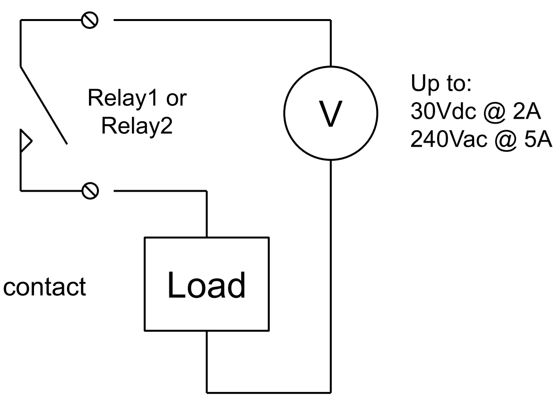
It is important to use wire rated for the voltages and currents in your system. The terminal blocks will accommodate up to 14AWG (2.5mm^2) wire. Please consider that heavy wires of 16 or 14AWG will be difficult to work with while wires smaller than 22AWG are easily damaged in this sort of installation. In general a good quality hookup wire of about 18AWG (0.75mm^2) with 300V rated insulation will accommodate any of the controller connections.

6.5.1 Relay Outputs

Two normally open relay contacts are provided labeled Relay 1 and Relay 2 and can be used for switching mid-sized loads such as valves and pumps up to a few amps.

The relay outputs are capable of switching up to 240Vac @ 5A or up to 30vdc @ 2A. Larger devices will require the use of a larger external relay or contactor to control.

The internal relay contacts are completely isolated from the remainder of the circuitry. There are no concerns for polarity or ground loops when using the relays.

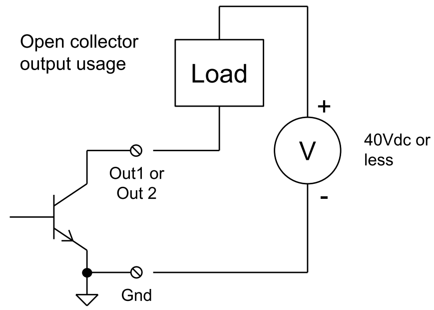


Relay contact usage

Note that the current rating for DC switching is much smaller than AC. Switching DC is potentially more damaging to the relay contacts at higher currents due to sustained electrical arcing as the contact is opened.

6.5.2 Digital Outputs

The digital outputs are sinking outputs rated to 40Vdc @ 1A. These can be used to power indicator lamps, additional relays, or remote inputs on other equipment. A catch diode is present to protect against the voltage surges typical in switching relay coils or other inductive loads.



Be aware that large loads on these inputs may cause voltage differences that affect analog readings if care is not taken with the wiring. Larger loads should use the relay outputs or an external relay. Sensors should be grounded directly to the PCB terminal blocks rather than an external grounding point using J2 positions 5 and 8.

A notable advantage to the transistor output over a relay is the lack of mechanical parts to wear out. If the load is continually switching on and off, such as a flashing warning lamp, a transistor output is likely the better choice rather than one of the relay outputs.

7 Setup & Calibration

Initial setup can be performed once the unit is connected to the local network.

7.1 Initial Setup

Upon first startup the processor will see that the non-volatile EEPROM memory is empty, the processor will load the factory default values into EEPROM. After this all settings may be kept in EEPROM to be loaded at startup. After changing settings it is important to save these settings to EEPROM using the on-screen option or via remote software.

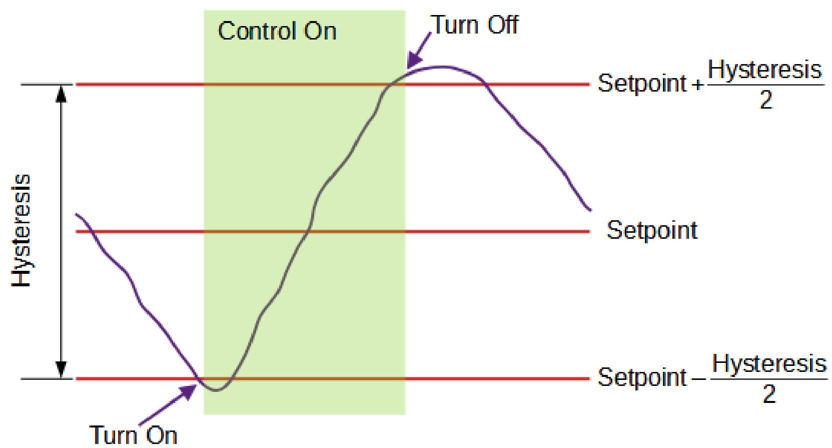
7.2 Control Loops

The SymCtrl is set up with four identical control loops. Each loop can take any of the analog inputs, compare the reading to the setpoints, and activate any of the outputs as needed. Each loop will control a single output. With the exception of the ToD controller (see below) outputs cannot be shared.

These loops are set up as simple bang-bang control with hysteresis. The average value desired is selected as the setpoint. The minimum and maximum allowable excursion from the setpoint is set as the hysteresis.

For example, if controlling pH with CO_2 sparging, the desired average pH is 7.0, this is selected as the setpoint. Setting hysteresis to 1.0 will cause the CO_2 valve to open at 7.5 and close at 6.5, keeping the average at 7.0.

As seen in the example below the control activates when the value falls below the setpoint minus half the hysteresis. This control is then turned off when the value exceeds the setpoint plus half the hysteresis.



In the example above the control activated on the low setpoint and deactivated on the high setpoint. In the Symbrosia controller this can be reversed with the control high/low parameter. Setting of this parameter depends on whether the controlled parameter is expected to increase or decrease when the control is active.

This would be typical of a temperature control turning on a heater where the temperature rises when the control is on. In SymbCtrl the control high/low parameter should be set to false for heating. For a cooling system this is reversed as the temperature goes down when activated. For SymbCtrl the high/low control would be set to true for cooling so that the control output activates at the high setpoint.

Properly set hysteresis ensures stable operation of the loop. Setting a hysteresis too small will cause noisy operation or chatter (quick on-off cycling). Rapid operation can be very energy inefficient for fans, pumps, compressors, or other equipment or even damage some types of equipment. The hysteresis setting should be significantly larger than any noise or fluctuations in the reading.

7.2.1 Control Loop Programming

7.2.2 Control Loop Parameters

The following parameters are available for all four control loops…

- **Control Input** The loop may use any analog input as the control input. Selecting no input will disable the control loop.
- **Control Output** The control may use any of the discrete outputs to effect control (relays or digital outputs). Selecting no output will allow the loop to operate and generate alarms, but no other control will be performed.
- **Loop Enabled** A true value will enable the control loop.
- **Setpoint** The average value desired in the controlled process.
- **Hysteresis** Difference between the high and low control thresholds.
- **Control High/Low** Determines if the control will activate at the low or high setpoint.
- **Alarm Low Value** The value at which a high alarm will be triggered.
- **Alarm High Value** The value at which a low alarm will be triggered.
- **Minimum Reading** The lowest reading encountered since the last reset of the minimum and maximum values.
- **Minimum Reading** The lowest reading encountered since the last reset of the minimum and maximum values.
- **Min/Max Reset** Clears the min/max values.

The following status outputs allow monitoring of the loop condition…

- **Control Active** is a read only parameter indicating if the conditions have been met to activate the control output. This indication does not show the effects of a time of day overlay on the actual output, if condition of the output is desired it should be queried directly.
- **Alarm** An alarm condition exists
- **Alarm High** A high alarm condition exists
- **Alarm Low** A low alarm condition exists

7.3 Time of Day Controller

The ToD controller allows functions to be turned off during part of the day or night. A time on and time off are selected, then any output channels are selected. The output will be on or off during the time specified. Since the controller has access to network time these functions should be quite accurate and the time need not be checked or adjusted over the months like mechanical or stand alone electrical timers.

The ToD controller can also overlay any of the output channels being used for other functions. Simply select the same output as controlled by one of the other control loops and the output will be disabled when the ToD controller is off. When the ToD controller is on the output will operate under control of the designated control loop.

The ToD controller has only a few registers required for setup. The start and stop times, in hours and minutes are set, then up to four outputs are selected. Unused outputs should be set to none or zero.

For example if control loop 1 is using relay 1 to control a CO_2 valve for pH control, overlaying the ToD controller by selecting relay 1 as an output can be used to disable the valve at night.

7.4 Software Counter

A counter is provided that will count events on one of the digital inputs or outputs. The counter is a 32-bit integer capable of counting up to 4,294,967,295 events. Any digital I/O channel can be selected as the source, the counter can be programmed to automatically reset every minute, hour, or day, or not reset at all.

It must be emphasized that the software counter is for low frequency events only, it will miss high frequency events. Events that occur at frequencies faster than 4Hz are not likely to be accurately counted. Likewise events with short pulse widths, narrower than 0.25 second are likely to be missed.

7.5 Software Timer

A timer is provided that will time events on one of the digital inputs or outputs. The timer is a 32-bit integer capable of counting up to 4,294,967,295 seconds. Any digital I/O channel can be selected as the source, the timer can be programmed to automatically reset every minute, hour, or day, or not reset at all.

It must be emphasized that the software timer is for low frequency events, events that last one second or longer, with a low sample rate it will miss fast events. This is suitable for monitoring pump run time or valve actuation and similar slow processes.

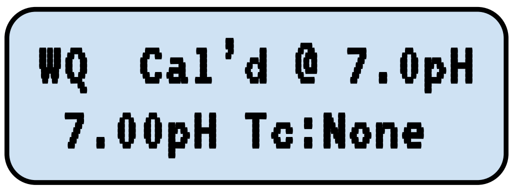
If the flow rate is known this timer can be used to measure the amount of gas or water used in a system. Simply multiply the timer result by the flow rate to calculate the amount used.

7.6 pH Calibration

PH calibration is accomplished with reference solutions of 7.0 for the zero offset and either 4.0pH or 10.0pH to set the gain. The controller will automatically recognize which solution is being used and set the correct calibration value. As long as the water quality sensor is configured for pH the controller may calibrate the sensor using the following procedure…

1. Advance to the WQ sensor screen and verify the probe is configured as a pH probe, the reading displayed should show pH as the units.
2. Long press the control wheel for at least two seconds to begin calibration, the message “Confirm?” will appear.
3. Long press the control wheel again to confirm calibration, the message “Calibrate!” should appear.

4. Place the probe into the 7.0pH reference solution and wait for the reading to stabilize.
5. Long press the control wheel to set the calibration offset. Once the reading is complete the controller will show success on the screen with the message “Cal’d @ 7pH” and the reading should be 7.0pH.
6. Long press the control wheel for at least two seconds to continue calibration, the message “Confirm?” will appear again.
7. Long press the control wheel again to confirm calibration, the message “Calibrate!” should appear.
8. Rinse the probe in clean water then place into either 4pH or 10pH reference solution, wait for the reading to settle and long press the button once again, the controller will take a reading to set the calibration gain.
9. Once the reading is complete the controller will show success on the screen with the message “Cal’d @ 4pH” or “Cal’d @ 10pH”, press the wheel once to return to the WQ sensor screen.
10. Verify a correct calibration by placing the pH probe into the different reference solutions and observing the displayed value. Recall that readings will take at least ten seconds to stabilize.



The electronics used for the pH probe are designed for very low long term drift, with a quality pH probe calibration should not be necessary again for some time. Calibration should be performed any time a pH probe is changed.

7.7 Temperature Calibration

Calibration of the temperature probes is done by accessing the correct calibration screen and adjusting the value to the correct reading. If you wish to know the correction factor needed to achieve this corrected reading it can be read at the Modbus address found in the table.

Because of the good linearity and repeatability of Pt100 probes no gain adjustment is usually needed. It is advised that offset adjustments be performed near the normal temperature range that will be used in operation (rather than using boiling or freezing as a calibration reference). In this manner any resulting errors will be quite small in practice.

Our normal approach is to calibrate probes to match a reference at about 20 to 25°C, as our ponds and bioreactors are normally kept near this temperature. This reference is simply a known good laboratory thermometer, preferably one that has been calibrated against an NIST standard. Good calibrated thermometers, such as the Cole-Parmer Traceable Lollipop, are available for less then $40.

7.7.1 Method One

This method should be used when high accuracy is required.

Place the temperature probe to be calibrated and a reference thermometer together in a container of water that has been allowed to stabilize at room temperature for at least a few hours. A decent sized container is better, at least a liter. We prefer to use an insulated thermos or hydro-flask that has been sitting overnight to stabilize. Perform calibration as shown below.

7.7.2 Method Two

This method can be used in place on the equipment, such as when a sensor is replaced.

Install the controller and the temperature probe into the culture vessel or bioreactor that will be used. Use a calibrated reference thermometer to measure the water temperature in the culture vessel and adjust the temperature reading to match as described below.

7.7.3 Calibration

With the temperature sensor in place and a known reading on a calibrated thermometer.

1. Wait for the readings on both the controller and calibrated thermometer to stabilize. The two readings should be close to each other, if the difference is large (more than 10°C) there may be something wrong with the sensor or unit.
2. Using the control knob select the temperature sensor screen.
3. Press the control knob for at least two seconds and release, the Temp 1 reading should begin to flash.
4. Adjust the reading by rotating the control knob until the reading on the controller matches the reading on the thermometer. The reading is adjusted in 0.1 degree steps.
5. Quickly press the knob again, the Temp 2 reading will begin to flash, adjust if needed or skip if Temp 2 is not used.
6. Quickly press the knob one more time to exit the calibration settings and save the new calibration constants to flash memory.

If the unit is left in calibration mode for an extended time it will automatically exit and return to the main status screen. In this case the new calibration values ARE NOT saved to flash and will not be present after a reset or power cycle.

7.8 Processed Reading

The processed reading is a method to combine two floating point readings into a single value for use by a control loop. Consider the case of a large tank with two temperature sensors, the reading processor would allow these two temperatures to be averaged before using the result for temperature control. A number of processes are available.

7.9 Logic Gate

The logic gate allows a logic operation to be applied to one or two boolean IO channels.

For example, a flashing alarm lamp is needed. The logic gate could be configured to take the desired alarm state, AND this value together with the Flasher register, and the result used to drive digital output 1 which is wired to the lamp.

8 Local Operation

Local control is through a series of screens on the front panel LCD display. Each screen is identified through a title on the upper left corner of the display.

The control knob can be rotated to scroll screens or values, The knob can also be pressed to begin adjustments. There are two click types used in entry, a short click which is brief, less than one second. Also used is a long click where the control knob must be depressed for at least two seconds.

A separate set of screens allows entry of many of the most common parameters using the control knob. The setting screens are entered on the setting screen found in the main set of screens by pressing the control knob for at least two seconds. If left in a setting screen the controller with timeout and go back to the main status screen after a couple minutes.

Keep in mind that only a limited subset of parameters can be adjusted using the front panel, those that are most likely to be needed by the operator in the field. Complete configuration of all parameters is most easily done through the remote interface.

8.1 Screens

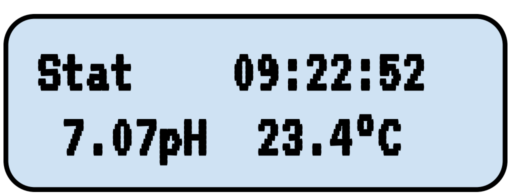
The LCD display can be cycled through a number of screens to display some (by no means all) of the operating parameters. The name of each screen appears in the upper left corner of most screens.

A few of these screens allow setting of some of the key parameters or calibration while full control is available only via a remote connection.

Advancing through the screens is done by either pressing the control wheel, or by rotating the control wheel back and forth. The list is wrap-around, going past the last screen brings you to the first, this works in both directions, thus that last few screens are often quicker to access by going backwards. Those settings that can be changed are generally accessed by a long press, longer than two seconds, see the sections below for specific details.

8.1.1 Status

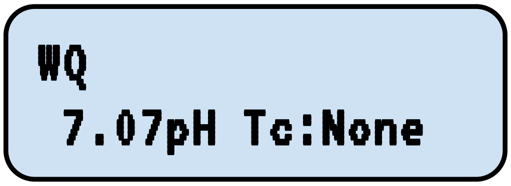
The basic status screen with time and two selectable values. The display will return to the status screen after five minutes of inactivity. The upper left will display the word Stat unless a name has been assigned to the unit in which case the first seven characters of the controller name will be displayed.



The two sensor readings can be customized with any two readings of choice being displayed using Modbus registers Status Display 1 and Status Display 2. This gives an immediate view of the most critical values in a given system.

8.1.2 WQ Sensor

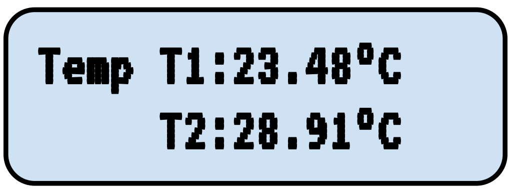
The WQ screen displays calibrated reading from the water quality sensor and the source for use in temperature compensation in selected.



If the sensor is set for pH a long press will start calibration. See the [instructions for pH calibration] in the setup section.

8.1.3 Temps

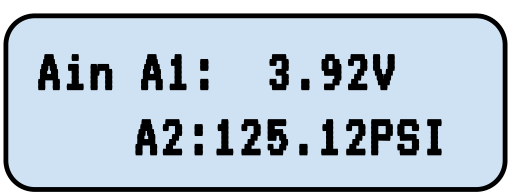
The Temp screen displays readings from the two temperature sensors. Either Centigrade or Fahrenheit may be selected via Modbus settings.



Simple offset calibration may be started with a long press, two seconds or more. Temperature sensor one is then calibrated by rotating the wheel to adjust the reading in 0.1 degree steps to the correct value. A click of the button then allows adjustment of temperature sensor 2, with one more click used to exit calibration. The new calibration offset is written to flash memory upon the last click

8.1.4 Analog

Readings from the two analog inputs are displayed. These readings may be converted from voltage or current to whatever engineering units are needed by use of the offset and gain values set via the Modbus interface. Likewise the units displayed may be selected and set via software.



8.1.5 Processed Reading

The result of the processed reading and which inputs are used is displayed. The processed reading is configured via software.

```
Proc    23.48°C
A:Temp1 B:Temp2
```

8.1.6 Control

Status of each of the control loop is shown on a series of three screens for each loop, a total of twelve screens. The first shows the loop name, current reading of the selected input, and loop status (Disabled/Active/Inactive). On this screen the user may enable or disable the control loop with a long press of the control knob.

```
Ctrl1
 7.01pH Active
```

The second screen displays the input and output assignments for the control loop.

```
Ctrl1  In:WQAmp
      Out:Rly1
```

The third screen displays the setpoint and hysteresis. On this screen the setpoint may be adjusted with a long press of the knob, at which point the S: will flash indicating adjustment. Rotating the control knob will then adjust the setting, while clicking once more to accept the new value. The new setpoint is written to flash memory upon the last click.

```
Ctrl1 S:   7.0pH
      H:   1.0pH
```

8.1.7 Logic

Shows the status of the logic gate, the two selected inputs, the logic function, and result.

```
Logic AND   Low
A:Ctrl1 B:Flash
```

8.1.8 Outputs

Displays the status of the four control outputs; relays 1 and 2, outputs 1 and 2.

```
Out 1:On D1:Off
 Rly2:Off D2:Off
```

8.1.9 Time of Day

Displays the Time of Day range setting, current time, and active status. A long click on this screen will enable or disable the time-of-day function.

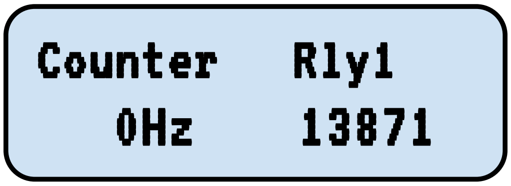
```
TofD 06:00-18:00
09:55:57 Active
```

A second screen displays which outputs are controlled by the ToD function.

```
TofD Rly1  None
     None None
```

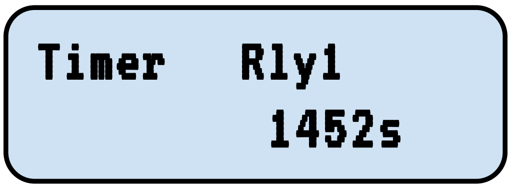
8.1.10 Counter

Displays the event counter input and current count. A long 2 second click will reset the counter to zero.



8.1.11 Timer

Displays the event timer input and current time in seconds. A long click will reset the timer to zero.

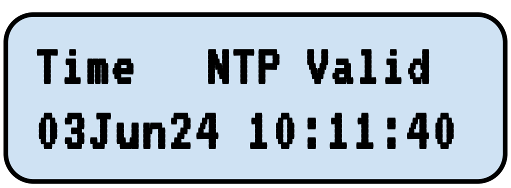


8.1.12 Modbus

Displays current Modbus server status. Any remote queries will be indicated by the Active/Inactive indication at the top.

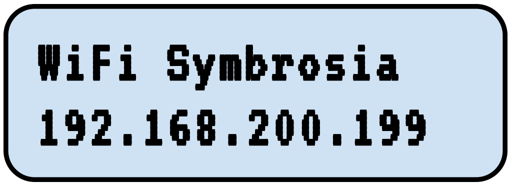
8.1.13 Time

Displays the current date and time along with the result of the last NTP sync attempt. A long press of the knob will trigger an NTP update attempt.

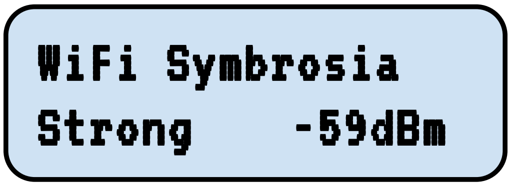


8.1.14 WiFi

The first WiFi screen displays the currently connected WiFi access point SSID along with the assigned IP address.



A second WiFi screen displays the signal strength. This screen will be skipped if there is no network connection.



If the unit fails to connect to the local network the WiFi screen will simply display “No network!” In this case the following procedure is used to get the unit online. This can also be used to shift the unit from one network to another or to simply enter a new password if the password has been changed.

1. Power on the unit and wait for the connection attempt to complete
2. Use the control knob to select the WiFi status screen, this screen will either display the connection status or “No network!”
3. Press and hold the control knob for at least two seconds, the unit will begin a scan for local networks.
4. After about 10-20 seconds the unit will display “Select…” and a list of network names.
5. Use the control knob to select the desired WiFi network and click the control knob once.
6. Enter the WiFi password one character at a time using the control knob with a short click used to select each character in turn. When selecting the last character use a long press to complete password entry.

   The characters are available in the following order and the list will roll around to the start after the last character, a backspace is available after the last character, or before the first.

   ABCDEFGHIJKLMNOPQRSTUVWXYZ
   abcdefghijklmnopqrstuvwxyz0123456789
   $@^`,|%;.()/{}:?[]=-+_#!*<backspace>

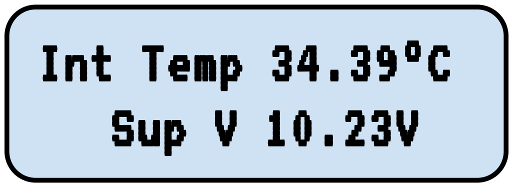
   Once the network and password have been entered the unit returns to normal operation. If you wish to attempt a new connection using the new credentials simply reset the unit using a two second press on the unit data screen.

A single click will accept the selection and begin entry of the password. Scroll through the alphabet in the order shown above to select each character with a single click. On the last character use a long click of two seconds to enter the new password and save it to flash memory. A backspace is available at the end of the characters to correct any mistakes.

Note that after successful entry of the WiFi credentials the unit will jump to the Unit Data screen. A second long press of the control knob here will reset the controller and attempt to use the newly entered WiFi setup.

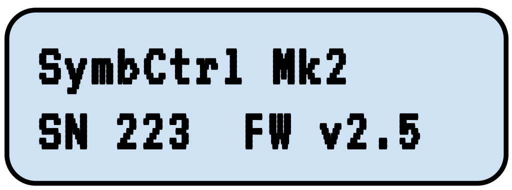
8.1.15 Internal Status

Internal values are displayed here including internal temperature as measured on the PCB and the supply voltage.



8.1.16 Unit Data

Basic data on the controller is displayed, this includes model, serial number, and firmware revision. A long press of the control knob on this screen will reset the unit.



9 Configuration and Remote Operation

The controller uses ModbusTCP protocol for configuration and remote monitoring and operation. This protocol is robust, widely supported, and is easily integrated into any SCADA software.

9.1 SyView

To allow easy setup and configuration of the controller a Python script is provided. SyView gives a graphical user interface to the unit and allows configuration of all functions as well as downloading and uploading configuration files. This allows the user to avoid all of the mucking about in Modbus registers.

SyView may also be used as example code allowing a more advanced user to write their own scripts and setting up more complex automation and logging.

9.2 Modbus Address Map

Traditional Modbus uses specific ranges for different types of data (Holding registers, coils, etc.). Each register has an address and reading or writing to a register requires little more than knowing the address for the information needed.

The Modbus specification makes no distinction in which register is to be used for what function, this is left entirely to the device designer. Thus a user will need a list of the registers and contents. SymbCtrl uses 400 holding registers and 70 coil registers.

9.3 Holding Registers

In Modbus holding registers hold a 16 bit integer value, thus a binary number from 0 to 65535 may be represented.

To store other forms of data a little creativity is needed. Negative numbers can be represented by using the first bit as a sign, leaving 15-bits for the number, thus a signed integer can be anything from -32768 to +32767.

A larger integer number may be stored by using two holding registers together, thus a 32-bit unsigned integer used to store a number from 0 to 4,294,967,296. Likewise a floating point number may be stored in two holding registers, usually as a IEEE-754 format 32-bit floating point number with bits set aside as a sign, an exponent, and a mantissa.

Strings can also be stored, each 16-bit Modbus register can store two 8-bit ASCII characters, thus a sixteen character string can be packed into eight holding registers. The SymbCtrl uses all of these formats, and the user will need to know which format is in use if writing code for interfacing with the controller.

9.3.1 Status Code

Address	0	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The status register holds a series of bits, each with the status of a specific problem or alarm, this bit is set if the issue exists. Many of these can be queried separately, see the sections on those items for more information. Thus a value of zero indicates that no problems are present in the controller.

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Control Loop Alarm	Control 4 Alarm	Control 3 Alarm	Control 2 Alarm	Control 1 Alarm	Analog 2 Invalid	Analog 1 Invalid	Temp 2 Invalid	Temp 1 Invalid	WQ Input Invalid	Network Time Invalid	WiFi Status	Internal Temperature High	Supply Voltage High	Supply Voltage Low	Startup

9.3.2 Model Number

Address	1	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

A model number of the controller, currently all Symbrosia controllers are model #2. This may be used in the future to allow software checks and the use of other models with other register maps in the same software.

9.3.3 Serial Number

Address	2	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The unique serial number of the unit, this is programmed into the unit during manufacture.

9.3.4 Firmware Revision

Address	3	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The firmware revision is stored as two eight bit bytes, the high byte is the major revision, the low bye is the minor revision. Thus firmware 2.6 will be 0206h or 518 in decimal.

9.3.5 Heartbeat Input

Address	4	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The heartbeat is a simple way to check communication and if code is running in the controller. A number written to Heartbeat will be copied to Heartbeat Output.

9.3.6 Heartbeat Output

Address	5	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The heartbeat is a simple way to check communication and if code is running in the controller. A number written to Heartbeat Input will be copied to Heartbeat Output.

9.3.7 Status Display 1

Address	8	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

This register allows the user to select which data is displayed on the status screen lower left area. Any IO channel may be displayed, see the list in IO Channels for the value needed.

9.3.8 Status Display 2

Address	9	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

This register allows the user to select which data is displayed on the status screen lower right area. Any IO channel may be displayed, see the list in IO Channels for the value needed.

9.3.9 Time Zone

Address	10	**Registers**	1
Format	Signed Integer	**Mode**	Read and Write

Time zone allows the correct time to be calculated using the universal time returned by NTP when the controller fetches time from the network. This is a signed integer to allow time zone ahead of or behind universal time.

US/Canada EST	-5	Hawaii	-10	Germany	1
US/Canada CST	-6	Alaska	-9	Japan/Korea	9
US/Canada MST	-7	England	0	China	8
US/Canada PST	-8	France	1	Eastern Australia	10

As most aquaculture applications prefer to remain synchronized with daylight, the controller makes no provision for daylight savings time.

9.3.10 RTC Year

Address	11	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The four digit calendar year from the real time clock (RTC) set using network time.

9.3.11 RTC Month

Address	12	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The four digit calendar year from the real time clock (RTC) set using network time. Note that January is month zero, the range is 0 to 11.

9.3.12 RTC Day

Address	13	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The calendar day from the real time clock (RTC) set using network time.

9.3.13 RTC Hour

Address	14	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The current hour from the real time clock (RTC) set using network time.

9.3.14 RTC Minute

Address	15	**Registers**	1

Format	Unsigned Integer	**Mode**	Read Only

The current minute from the real time clock (RTC) set using network time.

9.3.15 RTC Second

Address	16	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

The current second from the real time clock (RTC) set using network time.

9.3.16 WQ Sensor Reading

Address	20	**Registers**	2
Format	Floating point	**Mode**	Read Only

The current reading from the WQ Sensor Amplifier. This reading is converted to voltage in the range -0.5 to 0.5Vdc. If units of pH are selected the number is automatically converted to pH using the Nernst constant of 0.0592V/pH. Any other conversion will require scaling with the WQ Offset Register and WQ Gain Register. If the reading is invalid, likely due to a disconnected probe, WQ Valid will be false.

9.3.17 Temperature Sensor 1 Reading

Address	22	**Registers**	2
Format	Floating point	**Mode**	Read Only

The current reading from temperature amplifier 1. This reading is converted to °C or °F depending on the selected units. The amplifier and digital converter are configured for the range -35°C to 135°C, readings outside this range will be reported as invalid.

9.3.18 Temperature Sensor 2 Reading

Address	24	**Registers**	2
Format	Floating point	**Mode**	Read Only

The current reading from temperature amplifier 2. This reading is converted to °C or °F depending on the selected units. The amplifier and digital converter are configured for the range -35°C to 135°C, readings outside this range will be reported as invalid.

9.3.19 Analog Input 1

Address	26	**Registers**	2
Format	Floating point	**Mode**	Read Only

The reading from Analog Amplifier 1. The valid range is 0V to 10.8V, readings higher than 10.8V will be reported as invalid.

The default reading is in volts, this will automatically be converted to mV or mA if these units are selected in the Analog Input 1 Units register. Conversion to other engineering units can be done through use of the Analog Input 1 Gain and Analog Input 1 Offset registers.

9.3.20 Analog Input 2

Address	28	**Registers**	2
Format	Floating point	**Mode**	Read Only

The reading from analog amplifier 2. The valid range is 0V to 10.8V, readings higher than 10.8V will be reported as invalid.

The default reading is in volts, this will automatically be converted to mV or mA if these units are selected in the Analog Input 2 Units register. Conversion to other engineering units can be done through use of the Analog Input 2 Gain and Analog Input 2 Offset registers.

9.3.21 Internal Temperature

Address	30	**Registers**	2
Format	Floating point	**Mode**	Read Only

The current internal temperature in either °C or °F, units can be set using the Internal Temperature Units register, see IO Channels Units Table for the needed values. Readings below -20°C or above 120°C are flagged as invalid using the Internal Temperature Valid register.

9.3.22 Supply Voltage

Address	32	**Registers**	2
Format	Floating point	**Mode**	Read Only

The current supply voltage in volts.

9.3.23 Processed Reading

Address	34	**Registers**	2
Format	Floating point	**Mode**	Read Only

The result of the processed reading, the units will be inherited from the processed reading input A. See Processed Reading for more information on this function.

9.3.24 WQ Sensor Units

Address	36	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to set the desired units for the water quality amplifier reading, this is normally in volts, millivolts, or pH. The unit value is taken from the IO Channels Units Table.

9.3.25 Temperature 1 Units

Address	37	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to set the desired units for temperature sensor 1, either °C or °F. The unit value is found in the IO Channels Units Table.

9.3.26 Temperature 2 Units

Address	38	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to set the desired units for temperature sensor 2, either °C or °F. The unit value is found in the IO Channels Units Table.

9.3.27 Analog Input 1 Units

Address	39	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to set the desired units for analog input 1. The native reading here is either volts or mA, conversion will be done automatically for the selected units. The reading can be converted to any other engineering unit using analog 1 gain and offset. The desired unit value is found in the IO Channels Units Table.

9.3.28 Analog Input 2 Units

Address	40	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to set the desired units for analog input 2. The native reading here is either volts or mA, conversion will be done automatically for the selected units. The reading can be converted to any other engineering unit using analog 2 gain and offset. The desired unit value is found in the IO Channels Units Table.

9.3.29 Internal Temperature Units

Address	41	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to set the desired units for internal temperature reading, a thermistor mounted in the center of the controller PCB. This should be either °C or °F, the unit value is found in the IO Channels Units Table.

9.3.30 Supply Voltage Units

Address	42	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

Reflects the units for the supply voltage, this is fixed as volts and is provided for convenience in writing software.

9.3.31 Processed Reading Units

Address	43	**Registers**	1
Format	Unsigned Integer	**Mode**	Read Only

Reflects the units for the processed reading, the units will be inherited from input A of the processed reading.

9.3.32 pH Temperature Compensation

Address	44	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Select a temperature source for use in pH temperature compensation, this is usually temperature sensor 1 or temperature sensor 2. The desired unit value is found in the IO Channels Units Table.

9.3.33 WQ Sensor Offset

Address	50	**Registers**	2
Format	Float	**Mode**	Read and Write

An offset added to the WQ sensor reading to allow slope and offset calibration of the value. This can be set directly through this register, or using the built in pH calibration procedure. Note that WQ gain and offset are in the selected units, set the units before calibration.

9.3.34 Temperature Sensor 1 Offset

Address	52	**Registers**	2
Format	Float	**Mode**	Read and Write

An offset added to the Temperature Sensor 1 Reading in order to implement slope and offset calibration of the reading. Note that temperature gain and offset are in °C.

9.3.35 Temperature Sensor 2 Offset

Address	54	**Registers**	2
Format	Float	**Mode**	Read and Write

An offset added to the Temperature Sensor 2 Reading in order to implement slope and offset calibration of the reading. Note that temperature gain and offset are in °C.

9.3.36 Analog Input 1 Offset

Address	56	**Registers**	2
Format	Float	**Mode**	Read and Write

An offset added to the Analog Input 1 Reading in order to implement slope and offset calibration of the reading. This may also be used to allow conversion of the reading into the desired engineering units. Note that gain and offset are in the selected units, set the units before calibration.

9.3.37 Analog Input 2 Offset

Address	58	**Registers**	2
Format	Float	**Mode**	Read and Write

An offset added to the Analog Input 2 Reading in order to implement slope and offset calibration of the reading. This may also be used to allow conversion of the reading into the desired engineering units. Note that gain and offset are in the selected units, set the units before calibration.

9.3.38 WQ Sensor Gain

Address	60	**Registers**	2
Format	Float	**Mode**	Read and Write

A gain factor multiplied with the WQ sensor reading to allow slope and offset calibration of the value. This can be set directly through this register, or using the built in pH calibration procedure. Note that WQ gain and offset are in the selected units, set the units before calibration.

9.3.39 Temperature Sensor 1 Gain

Address	62	**Registers**	2
Format	Float	**Mode**	Read and Write

A gain factor multiplied with the Temperature Sensor 1 Reading in order to implement slope and offset calibration of the reading. Note that temperature gain and offset are in °C.

9.3.40 Temperature Sensor 2 Gain

Address	64	**Registers**	2
Format	Float	**Mode**	Read and Write

A gain factor multiplied with the Temperature Sensor 2 Reading in order to implement slope and offset calibration of the reading. Note that temperature gain and offset are in °C.

9.3.41 Analog Input 1 Gain

Address	66	**Registers**	2
Format	Float	**Mode**	Read and Write

A gain factor multiplied with the Analog Input 1 Reading in order to implement slope and offset calibration of the reading. This may also be used to allow conversion of the reading into the desired engineering units. Note that gain and offset are in the selected units, set the units before calibration.

9.3.42 Analog Input 2 Gain

Address	68	**Registers**	2
Format	Float	**Mode**	Read and Write

A gain factor multiplied with the Analog Input 2 Reading in order to implement slope and offset calibration of the reading. This may also be used to allow conversion of the reading into the desired engineering units. Note that gain and offset are in the selected units, set the units before calibration.

9.3.43 Control 1 to 4 Input Channel

Addresses	70, 86, 102, 118	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to specify the input channel for control loop 1, may be any analog value. See the list in IO Channels for the value needed.

9.3.44 Control 1 to 4 Output Channels

Addresses	71, 87, 103, 119	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to specify the output channel for control loop 1, may be any output channel. See the list in IO Channels for the value needed.

9.3.45 Control 1 to 4 Setpoints

Addresses	72, 88, 104, 120	**Registers**	2
Format	Float	**Mode**	Read and Write

Used to specify the setpoint for the control loop..

9.3.46 Control 1 to 4 Hysteresis

Addresses	74, 90, 106, 122	**Registers**	2
Format	Float	**Mode**	Read and Write

Used to specify the hysteresis for the control loop.

9.3.47 Control 1 to 4 Alarm Low

Addresses	76, 92, 108, 124	**Registers**	2
Format	Float	**Mode**	Read and Write

Used to specify the low alarm point for the control loop, if the input value is below this value a low alarm will occur.

9.3.48 Control 1 to 4 Alarm High

Addresses	78, 94, 110, 126	**Registers**	2
Format	Float	**Mode**	Read and Write

Used to specify the high alarm point for the control loop, if the input value is above this value a high alarm will occur.

9.3.49 Control 1 to 4 Enable Source

Addresses	80, 96, 112, 128	**Registers**	1
Format	Integer	**Mode**	Read and Write

Used to specify an external enable source for the control loop. This source, if true, will enable the control loop in the absence of the regular control enable register. May be any boolean IO channel, see the IO Channel List for the correct value.

9.3.50 Control 1 to 4 Minimum On and Off Time

Addresses	81, 97, 113, 129	**Registers**	1
Format	Integer	**Mode**	Read and Write

Used to specify a minimum on and off time for the control loop in seconds. If set to anything other than zero this will force the control loop result to remain on or off for a minimum the specified time.

This is needed for devices like refrigeration compressors that should not be started or stopped rapidly, once on it will stay on for the specified time, once off it will stay off for at least the specified time.

9.3.51 Control 1 to 4 Minimum Value

Addresses	82, 98, 114, 130	**Registers**	2
Format	Float	**Mode**	Read and Write

A register that records the minimum value experienced by the input reading. Useful for control loop tuning and checks, the lowest value experienced can be retrieved. This can be reset by setting the control loop min/max reset.

9.3.52 Control 1 to 4 Maximum Value

Addresses	84, 100, 116, 132	**Registers**	2
Format	Float	**Mode**	Read and Write

A register that records the minimum value experienced by the input reading. Useful for control loop tuning and checks, the lowest value experienced can be retrieved. This can be reset by setting the control loop min/max reset.

9.3.53 Logic Input A

Address	136	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The registers specifies input A to the logic gate function, this may be any digital IO channel as listed in the IO Channels Table.

9.3.54 Logic Input B

Address	137	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The registers specifies input B to the logic gate function, this may be any digital IO channel as listed in the IO Channels Table.

9.3.55 Logic Function

Address	138	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The register is used to specify the logic function to be used by the logic gate function.

0	NOT - Output is the inverse of A
1	AND - Output is the logical and of A and B
2	NAND - Output is the inverse logical AND of A and B
3	OR - Output is the logical OR and A and B
4	NOR - Output is the inverse logical OR of A and B
5	XOR - Output is the logical exclusive OR of A and B
6	NXOR - Output is the inverse logical exclusive OR of A and B
7	Echo - Output is the same as A, can be used to move the value of one digital value to another channel

9.3.56 Logic Gate Output Channel

Address	139	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

Used to specify the output channel for he logic gate, may be any output channel. See the list in IO Channels for the value needed.

9.3.57 Time of Day On Time Hours

Address	140	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The start time hours of the Time of Day function, must be in the range 0 to 23.

9.3.58 Time of Day On Time Minutes

Address	141	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The start time minutes of the Time of Day function, must be in the range 0 to 59.

9.3.59 Time of Day Off Time Hours

Address	142	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The end time hours of the Time of Day function, must be in the range 0 to 23.

9.3.60 Time of Day Off Time Minutes

Address	143	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The end time minutes of the Time of Day function, must be in the range 0 to 59.

9.3.61 Time of Day Output 1, 2, 3, and 4

Addresses	144, 145, 146, 147	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The output channels to be controlled by the Time of Day function, up to four outputs may be controlled. The IO channels are found in the IO Channels Table and may be specified in any order.

9.3.62 Counter Source

Address	150	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The channel to be monitored by the Software Counter. The IO channel number found in the IO Channels Table.

9.3.63 Counter

Address	151	**Registers**	2
Format	Long Integer	**Mode**	Read Only

The counter value representing the number of times the input channel has been activated.

9.3.64 Counter Reset Interval

Address	153	**Registers**	1
Format	Long Integer	**Mode**	Read and Write

The counter reset interval, this may be never, reset each minute, reset each hour, or reset each day.

0	Never reset	**2**	Reset each hour
1	Reset each minute	**3**	Reset each day

9.3.65 Timer Source

Address	154	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The channel to be monitored by the Software Timer. The IO channel number found in the IO Channels Table.

9.3.66 Timer

Address	155	**Registers**	2
Format	Long Integer	**Mode**	Read Only

The timer value representing the time in seconds the input channel has been active.

9.3.67 Timer Reset Interval

Address	157	**Registers**	1
Format	Long Integer	**Mode**	Read and Write

The timer reset interval, this may be never, reset each minute, reset each hour, or reset each day.

0	Never reset	**2**	Reset each hour
1	Reset each minute	**3**	Reset each day

9.3.68 Process Input A

Address	166	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The channel to be used as the A input of the Processed Reading function, this may be any analog IO channel as listed in the IO Channels Table.

9.3.69 Process Input B

Address	167	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The channel to be used as the B input of the Processed Reading function, this may be any analog IO channel as listed in the IO Channels Table.

9.3.70 Process

Address	168	**Registers**	1
Format	Unsigned Integer	**Mode**	Read and Write

The specific process to be applied to the input channels as found in the table below.

0	Average - The output will be the statistical average of the inputs
1	Minimum - The output will be the lesser of the two inputs
2	Maximum - The output will be the larger of the two inputs
3	Sum - The output will be the sum of the two inputs A + B
4	Difference - The output will be the difference of the two inputs A - B
5	Priority - The output will be A as long as a is valid, if A is invalid the output will be B

.

9.3.71 Model Name

Address	170	**Registers**	8
Format	String	**Mode**	Read Only

Model name in string format, currently 'SymbCtrl Mk2'.

9.3.72 Controller Name

Address	178	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for the controller, 16 characters available, the first eight characters are shown on the status screen.

9.3.73 WQ Sensor Name

Address	186	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for the WQ sensor, with 16 characters available.

9.3.74 Temperature Sensor 1 Name

Address	194	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for temperature sensor 1, with 16 characters available.

9.3.75 Temperature Sensor 2 Name

Address	202	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for temperature sensor 2, with 16 characters available.

9.3.76 Analog Input 1 Name

Address	210	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for analog input 1, with 16 characters available.

9.3.77 Analog Input 2 Name

Address	218	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for analog input 2, with 16 characters available.

9.3.78 Relay 1 Name

Address	226	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for relay 1, with 16 characters available.

9.3.79 Relay 2 Name

Address	234	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for relay 2, with 16 characters available.

9.3.80 Digital Output 1 Name

Address	242	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for digital output 1, with 16 characters available.

9.3.81 Digital Output 2 Name

Address	250	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for digital output 2, with 16 characters available.

9.3.82 Control 1 Name

Address	258	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for control loop 1, with 16 characters available.

9.3.83 Control 2 Name

Address	266	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for control loop 2, with 16 characters available.

9.3.84 Control 3 Name

Address	274	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for control loop 3, with 16 characters available.

9.3.85 Control 4 Name

Address	282	**Registers**	8
Format	String	**Mode**	Read and Write

User assigned name for control loop 4, with 16 characters available.

9.4 IO Channels

Any register that is used to select an IO channel for input or output uses an integer value to select the desired channel. Please refer to the table here for these values…

	Channel	Format
0	No IO Channel	
1	WQ Amplifier	Float
2	Temperature 1	Float
3	Temperature 2	Float
4	Analog Input 1	Float
5	Analog Input 2	Float
6	Internal Temperature	Float
7	Supply Voltage	Float
8	Processed Reading	Float
9	Digital Input 1	Boolean
10	Digital Input 2	Boolean
11	Output Relay 1	Boolean
12	Output Relay 2	Boolean
13	Digital Output 1	Boolean
14	Digital Output 2	Boolean
15	Virtual IO 1	Boolean
16	Virtual IO 2	Boolean
17	Control 1 Out	Boolean
18	Control 2 Out	Boolean
19	Control 3 Out	Boolean
20	Control 4 Out	Boolean
21	Control Alarm	Boolean
22	Control 1 Alarm	Boolean
23	Control 2 Alarm	Boolean
24	Control 3 Alarm	Boolean
25	Control 4 Alarm	Boolean
26	Control 1 Alarm Low	Boolean
27	Control 2 Alarm Low	Boolean
28	Control 3 Alarm Low	Boolean
29	Control 4 Alarm Low	Boolean
30	Control 1 Alarm High	Boolean
31	Control 2 Alarm High	Boolean
32	Control 3 Alarm High	Boolean
33	Control 4 Alarm High	Boolean
34	Flasher	Boolean
35	RTC Days	Integer
36	RTC Hours	Integer
37	RTC Minutes	Integer
38	RTC Seconds	Integer

9.5 IO Channels Units Table

Units for any analog IO channel are selected by entering the integer value found below into the units register for the desired channel.

0	No Units	**9**	m	**18**	%
1	°C	**10**	ml	**19**	ppm
2	°F	**11**	l	**20**	Ω
3	pH	**12**	g	**21**	day
4	mV	**13**	kg	**22**	hour
5	V	**14**	lbs	**23**	min
6	mA	**15**	kPa	**24**	sec
7	A	**16**	PSI		
8	mm	**17**	Hz		

9.6 Coil Registers

In Modbus a coil register stores a single bit, thus it represents true or false. Of course this may also represent on or off for a relay coil, yes or no, alarm, or any other true or false data.

9.6.1 Status

Address	0	**Mode**	Read Only

Status of the controller, true if status good, false if a fault exists.

9.6.2 NTP Time Valid

Address	1	**Mode**	Read Only

True if network time is valid, this will go false if no NTP request has succeeded in the last two hours. Time based controls will be disabled after 48 hours without a valid NTP request.

9.6.3 Startup

Address	2	**Mode**	Read Only

True if the controller is in a startup delay, no controls will be enabled until 30 seconds after controller startup has elapsed.

9.6.4 Save Settings

Address	3	**Mode**	Write Only

Writing a true to Save Settings will save all current settings to the EEPROM memory, any settings not saved will be lost on restart of the controller. This register will automatically clear once saved.

9.6.5 Clear Settings

Address	4	**Mode**	Write Only

Writing a true to Clear Settings will clear all settings in the controller except the WiFi credentials, restoring them to the default values. This register will automatically clear once the settings are cleared.

Be aware that this DOES NOT clear the settings in EEPROM, to completely clear the controller the Save Settings register must then be used to save the default settings to EEPROM. Conversely if the clear settings register is accidentally set the controller settings can be restored from EEPROM memory by power cycling the controller.

9.6.6 Midnight Save

Address	5	**Mode**	Read and Write

If the Midnight Save register is set the unit will automatically save all settings each midnight.

9.6.7 Midnight Reset

Address	6	**Mode**	Read and Write

If the Midnight Reset register is set the unit will automatically reboot each midnight.

9.6.8 Silence Alarms

Address	7	**Mode**	Read and Write

Setting the Silence Alarms register will prevent control loop alarms from occurring. Note that this does not block other faults like NTP failure or supply undervoltage, only the control loop alarms.

9.6.9 Digital Input 1

Address	8	**Mode**	Read Only

The state of Digital Input 1, this will be true if the voltage on the terminal is above 5V. Recall that Digital Input 1 shares the input terminal with Analog Input 1.

9.6.10 Digital Input 2

Address	9	**Mode**	Read Only

The state of Digital Input 2, this will be true if the voltage on the terminal is above 5V. Recall that Digital Input 2 shares the input terminal with Analog Input 2.

9.6.11 Relay 1 Status

Address	10	**Mode**	Read Only

The state of Relay 1, this will be true if the relay is energized and the contacts closed.

9.6.12 Relay 2 Status

Address	11	**Mode**	Read Only

The state of Relay 2, this will be true if the relay is energized and the contacts closed.

9.6.13 Digital Output 1 Status

Address	12	**Mode**	Read Only

The state of Digital Output 1, this will be true if the output transistor is on and the output pulled to ground.

9.6.14 Digital Output 2 Status

Address	13	**Mode**	Read Only

The state of Digital Output 2, this will be true if the output transistor is on and the output pulled to ground.

9.6.15 Relay 1 Request

Address	14	**Mode**	Read and Write

Writing true to the Relay 1 Request will force Relay 1 to actuate closing the contacts. This will override any other Relay 1 command. Writing a false to this register will allow other sources to command the relay.

9.6.16 Relay 2 Request

Address	15	**Mode**	Read and Write

Writing true to the Relay 2 Request will force Relay 1 to actuate closing the contacts. This will override any other Relay 2 command. Writing a false to this register will allow other sources to command the relay.

9.6.17 Digital Output 1 Request

Address	16	**Mode**	Read and Write

Writing true to the Digital Output 1 Request will force Output 1 to activate, pulling the terminal to ground. Writing a false to this register will allow other sources to command the output.

9.6.18 Digital Output 2 Request

Address	18	**Mode**	Read and Write

Writing true to the Digital Output 2 Request will force Output 2 to activate, pulling the terminal to ground. Writing a false to this register will allow other sources to command the output.

9.6.19 Virtual State 1

Address	18	**Mode**	Read and Write

A register that may be used for input or output for any controller function that reads or writes to a digital IO channel. Virtual State 1 can be used as a control enable source, a control result, or a logic gate input or output, etc.

9.6.20 Virtual State 2

Address	19	**Mode**	Read and Write

A register that may be used for input or output for any controller function that reads or writes to a digital IO channel. Virtual State 2 can be used as a control enable source, a control result, or a logic gate input or output, etc.

9.6.21 WQ Sensor Valid

Address	20	**Mode**	Read Only

A true value in this register indicates that a valid reading is available from the WQ sensor, that the reading is in a normal range. A false value generally indicates a disconnected or failed WQ sensor.

9.6.22 Temperature Sensor 1 Valid

Address	21	**Mode**	Read Only

A true value in this register indicates that a valid reading is available from Temperature Sensor 1, that the reading is in a normal range. A false value generally indicates a disconnected or failed sensor.

9.6.23 Temperature Sensor 2 Valid

Address	22	**Mode**	Read Only

A true value in this register indicates that a valid reading is available from Temperature Sensor 2, that the reading is in a normal range. A false value generally indicates a disconnected or failed sensor.

9.6.24 Analog Input 1 Valid

Address	23	**Mode**	Read Only

A true value in this register indicates that a valid reading is available from Analog Input 1, that the reading is in a normal range. A false value generally indicates an input voltage above 10.8V. Note that the A/D converter cannot report values below zero volts, thus low readings are not reported as invalid.

9.6.25 Analog Input 2 Valid

Address	24	**Mode**	Read Only

A true value in this register indicates that a valid reading is available from Analog Input 2, that the reading is in a normal range. A false value generally indicates an input voltage above 10.8V. Note that the A/D converter cannot report values below zero volts, thus low readings are not reported as invalid.

9.6.26 Internal Temperature Valid

Address	25	**Mode**	Read Only

A true value in this register indicates that a valid reading is available from the Internal Temperature Sensor, that the reading is in the range -20°C to 120°C (-4°F to 248°F).

9.6.27 Supply Voltage Valid

Address	26	**Mode**	Read Only

A true value in this register indicates that a valid reading of the supply voltage is available. This is always true and included for convenience in programming.

9.6.28 Processed Reading Valid

Address	27	**Mode**	Read Only

Indicates that a valid result is available for the Processed Reading. This will be valid as long as at least one of the two inputs has a valid reading, this allows the controller to continue operating using the remaining input despite the failure of one of the inputs being processed (a control alarm will still be asserted due to an invalid input).

9.6.29 Logic Gate Result

Address	28	**Mode**	Read Only

Result of the Logic Gate function.

9.6.30 Flasher

Address	29	**Mode**	Read Only

A 1Hz 50% duty cycle signal is available in the Flasher register. This can be used with the logic gate to flash a warning light or sounder.

9.6.31 Control Loop 1, 2, 3, and 4 Enable

Addresses	30, 31, 32, 33	**Mode**	Read and Write

A true value in the Control Loop Enable will enable the corresponding control loop. This enable will override the external enable.

9.6.32 Control Loop 1, 2, 3, and 4 High

Addresses	34, 35, 36, 37	**Mode**	Read and Write

This register selects control on low or control on high. When true the corresponding control loop will be active when the controlled parameter is above the setpoint. When false the control loop will become active when the controlled parameter is below the setpoint.

9.6.33 Control Loop 1, 2, 3, and 4 Active

Addresses	38, 39, 40, 41	**Mode**	Read Only

A true value indicates that the corresponding control loop is active. If active and connected to an output, that output will activate.

9.6.34 Control Alarm

Address	42	**Mode**	Read Only

This register indicates that a control alarm is active, a master alarm. Any one of the four control loops has either a high or low alarm.

9.6.35 Control Loop 1, 2, 3, and 4 Alarm

Addresses	43, 44, 45, 46	**Mode**	Read Only

A true value indicates that the corresponding control loop is asserting an alarm, either high or low.

9.6.36 Control Loop 1, 2, 3, and 4 Alarm Low

Addresses	47, 48, 49, 50	**Mode**	Read Only

A true value indicates that the corresponding control loop is asserting a low alarm, that the controlled parameter is less than the Alarm Low setpoint.

9.6.37 Control Loop 1, 2, 3, and 4 Alarm High

Addresses	47, 48, 49, 50	**Mode**	Read Only

A true value indicates that the corresponding control loop is asserting a high alarm, that the controlled parameter is greater than the Alarm High setpoint.

9.6.38 Reset Control Loop 1, 2, 3, and 4 Min/Max Reset

Addresses	47, 48, 49, 50	**Mode**	Write Only

Writing a true to this register resets the control loop recorded maximum and minimum values, this will set these registers to the current value. The register will automatically clear once this is done.

9.6.39 Software Counter Enable

Address	59	**Mode**	Read and Write

If this register is true and an input channel is set this will allow the Software Counter to count actuations of the channel.

9.6.40 Software Timer Enable

Address	60	**Mode**	Read and Write

If this register is true and an input channel is set this will allow the Software Timer to time actuation of the channel.

9.6.41 Time of Day Enable

Address	61	**Mode**	Read and Write

If this register is true the Time of Day function will be active.

9.6.42 Time of Day Active

Address	62	**Mode**	Read Only

If this register is true the Time of Day function is active

9.6.43 Reset Counter

Address	63	**Mode**	Write Only

Writing a true to this register will clear the Software Counter. This register will clear automatically once the counter is cleared.

9.6.44 Reset Timer

Address	64	**Mode**	Write Only

Writing a true to this register will clear the Software Timer. This register will clear automatically once the counter is cleared.

9.6.45 Control Loop 1, 2, 3, and 4 One-Shot

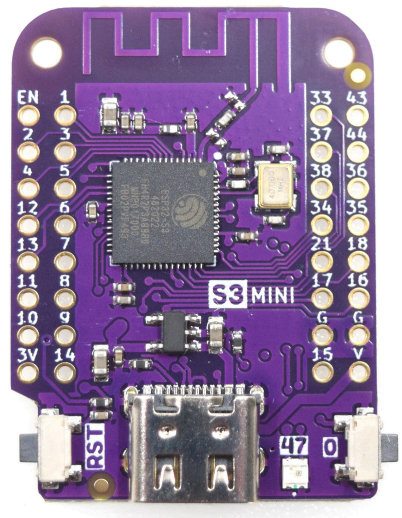
Addresses	66, 67, 68, 69	**Mode**	Read and Write

If true the corresponding control loop enable will operate in one-shot mode, the loop will only activate once. De-asserting the enable and re-asserting it will clear the one shot and allow it to operate once more.

10 The Electronics

Discussed here are design details of the circuitry on the SymbCtrl PCB. This is of interest to anyone delving into the design or contemplating modifications of the circuitry. Otherwise this section is not needed for anyone simply using the Symbosia Controller.

10.1 Controller



A WiFi centric processor, the Espressif ESP32-S2 or ESP32-S3 is used to control the unit. This processor is mounted to a Wemos S2 Mini or Wemos S3 Mini development board.

Use of the development board allows these high density surface mount components to be assembled without specialized equipment. The module can also be easily removed for testing, reprogramming, or repair.

These modules are available from the usual distributors for less than $10.

10.2 Power Supply

A simple linear regulator supplies internal power to the controller. Based on the ubiquitous LM7805 the regulator can accept from 8 to 24Vdc and remain in regulation. A blocking diode is in place to prevent damage in the case of reversed power supply connections.

As this is a linear regulator power dissipation is an issue. Thus higher input voltages will result in more heating of the regulator. Input voltages above 15Vdc are not recommended as the regulator will become quite warm. The normal current draw by the controller PCB is less than 275mA.

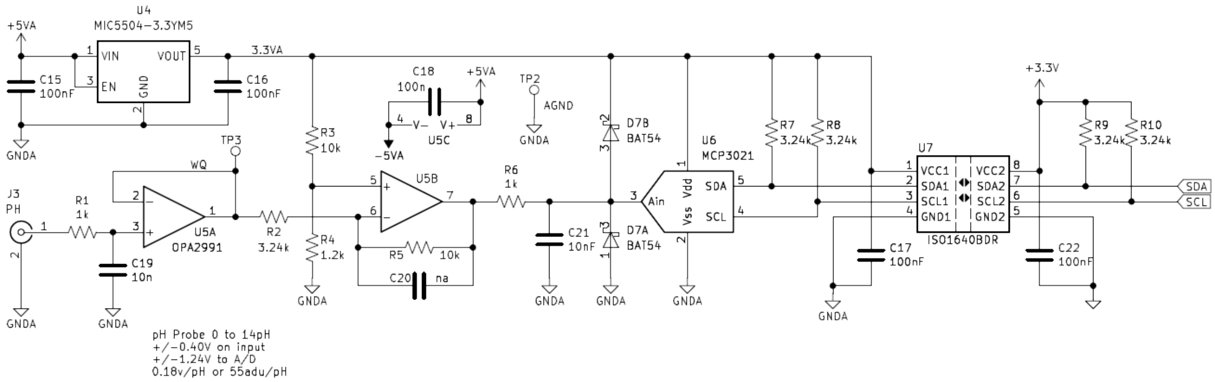
A number of switching regulator replacements for the 7805 are available, such as the CUI model VXO7805-500 that fit in the same PCB location. This replacement would allow operation on up to 36Vdc power supplies with minimal power dissipation.

The 8-15Vdc range allows reliable operation from battery and/or solar powered sources.

10.3 WQ Amplifier

The WQ amplifier will accept outputs from common pH, ORP, or similar water quality sensors with no modification. Adapting each type of sensor will simply require setting the units in software, then adjustment of the gain and offset parameters provided for calibration.

The typical sensor used is pH. Standard pH probes offer a convenient voltage output of -400mV to 400mV over the 1-14pH range. The ideal pH probe will present 0V with a balanced solution of 7pH, about -400mV at 1pH and +400mV at 14pH varying slightly with temperature.



As most commercial pH sensors use BNC style connectors this connector was chosen for the design. Other connectors can be easily adapted to BNC with inexpensive adapters.

The amplifier is designed as two stages, the first stage is simply a buffer amplifier to allow probes with high output impedances to be connected. The second stage performs three functions; gain, filtering, and applying an offset to shift the bipolar probe voltage to the 0-3.3Vdc input range of the MCP3021 10-bit ADC.

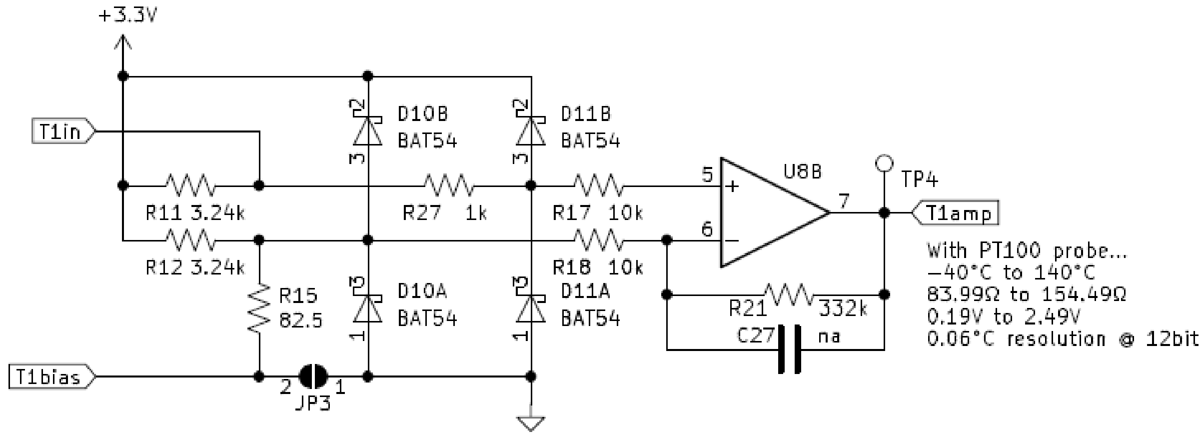
A pair of schottky diodes clamp the amplifier output keeping the voltage supplied to the ADC within the safe range of 0-3V as otherwise the output could be anywhere within -5 to 5V range of the amplifier, particularly when the probe is disconnected.

The entire circuit is electrically isolated from the remainder of the controller. Power is provided by an isolated DC-DC converter while the I2C communication signals are isolated using a ISO1640 isolated driver. A separate ground plane sits under the circuit on the PCB. Thus allowing contact between the exposed ground on the metal shell of the BNC connector and some other conductor should not cause harm.

10.4 Pt100 Amplifier

As temperature is such a critical function in aquaculture applications the Symbrosia controller commits two amplifiers to specifically measure temperature. The common Pt100 sensor was chosen in place of other types of temperature sensors for its availability, modest cost, and accuracy. An extraordinarily wide range of Pt100 sensors are available for almost any application.

The temperature inputs are fed to a pair of amplifiers designed to use Pt100 temperature sensors. The input biasing and gains are selected to support readings from -40°C to 120°C, temperatures beyond this will result in invalid readings and temperature control being disabled.



The amplifier is an OPA2991 op-amp, these are rail to rail, low drift amplifiers suitable for instrumentation applications like this. The inputs are arranged as voltage bridges that provide large impedances on the exposed output pins making misconnection or voltage transients less of an issue.

The supply voltage for the OPA2991 op-amp is connected to 0 and 3.3V, this keeps the output voltage within the safe voltage range for the ESP-32.

A solder jumper is provided that connects one branch of the bridge to ground. This allows the use of two wire Pt100 sensors without using a jumper in the terminal block to make this connection..

Pt100 Resistance vs. Temperature

Temperature (°C)	Resistance (Ω)	Voltage (V)	Temperature (°C)	Resistance (Ω)	Voltage (V)
-40	84.27	0.244	50	119.40	1.929
-30	88.22	0.435	60	123.24	2.111
-20	92.16	0.626	70	127.08	2.292
-10	96.09	0.815	80	130.90	2.472
0	100.00	1.003	90	134.71	2.651
10	103.90	1.190	100	138.51	2.829
20	107.79	1.376	110	142.29	3.006
30	111.67	1.561	120	146.07	3.183
40	115.54	1.745			

Given the circuit design each amplifier could be reconfigured to support other uses if necessary simply by changing part values. Specifically, common semiconductor temperature sensors such as the LMT87 temperature sensor IC could be directly connected if R10, R11, R17, R14, and C10 are removed from the design, and R20 changed to a zero-ohm jumper. The change in voltages then adjusted using the software gain and offset values to recalibrate the sensor.

10.5 Analog and Digital Inputs

Two analog inputs are available that can be used as 0-10Vdc or 4-20mA inputs. These inputs are protected against modest reverse or over voltage events with limiting resistors and transient protection diodes. Care should still be taken to avoid voltages outside of the correct input range or damage may occur.

A 249Ω resistor allows measurement of 4-20mA signals, converting 20mA to approximately 5Vdc. The jumpers JP1 or JP2 must be shorted across with a small bead of solder to use the 4-20mA inputs. Setting the analog units to mA will use the appropriate conversion factors to convert the reading to current.

The inputs are always configured to be analog inputs. Use as a digital input is accomplished by simply thresholding the analog reading in software and making this value available as a digital value. Values above approximately 5V are considered high, while 0 to 5V is considered low.

10.6 Internal Temperature

Internal temperature of the unit is available from a thermistor (R38) mounted to the PCB. This provides readings from about -20 to 120°C.

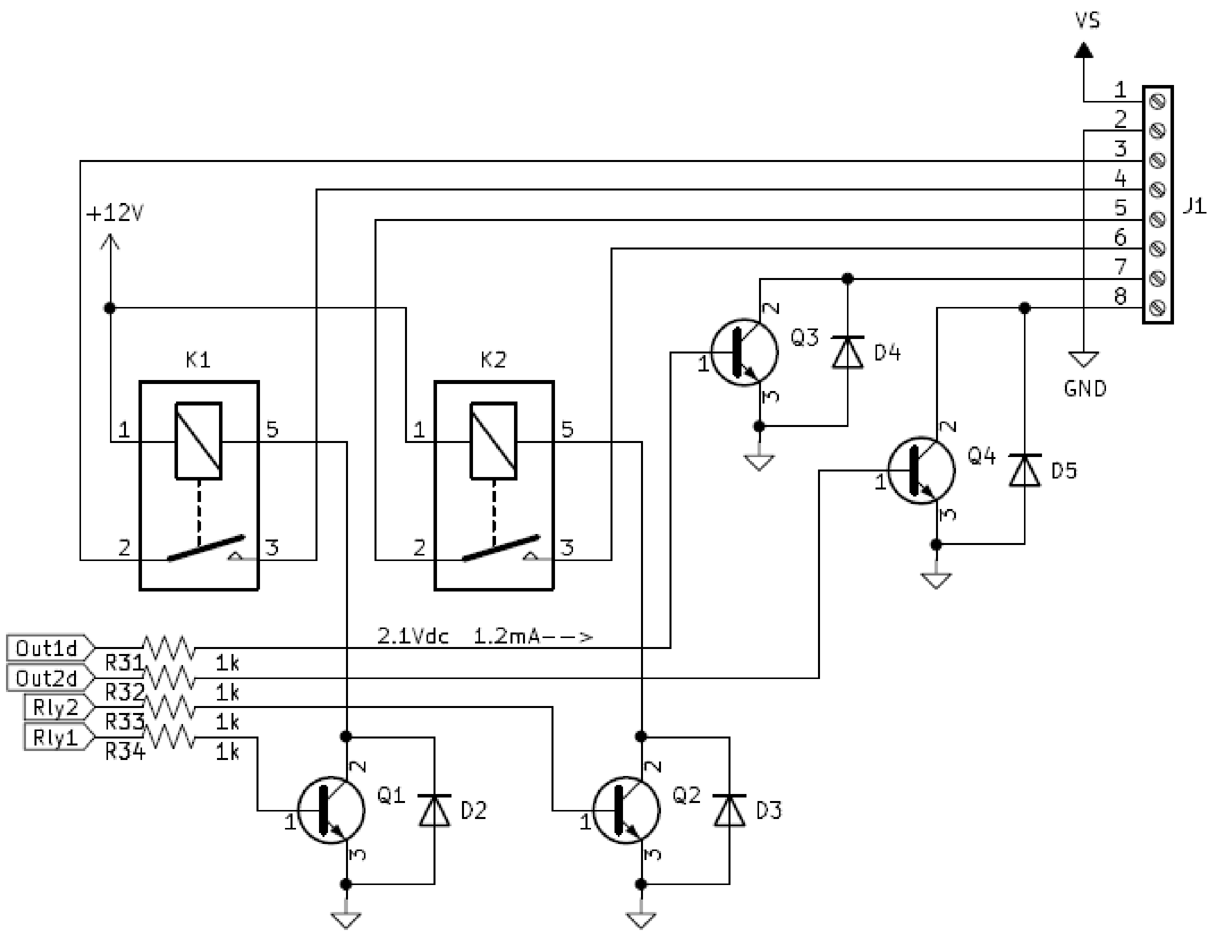
10.7 Supply Voltage Monitor

A voltage divider allows measurement of the supply voltage as connected to the J1 connector and before the voltage regulator.

10.8 Relay Outputs

Two relay outputs are provided to switch light loads such as valves or pumps.

The contacts on these are uncommitted and fully isolated from the controller circuitry. They may be connected in any fashion to voltages up to 250Vac and switch loads as much as 5A.



10.9 Digital Outputs

Two open collector transistor outputs provide much the same functionality as the relay outputs without the relay. These are suited to switching external loads such as small valves, indicator lights, or external relays and contactors.

These are commonly referred to as “open collector” or “low-side switches” where one side of the load is connected to power while the switch provides the ground connection when active. The load power supply may use any supply voltage lower than the voltage rating of the transistor, in this case about 45V using the specified BCX54 transistors. Thus 12Vdc, 24Vdc, or 36Vdc devices may be switched.

Since these are transistor outputs with no mechanical components to wear out these outputs are quite suited for functions that turn off and on repeatedly. For example, a flashing indicator lamp that displays an alarm status should be connected to one of the transistor outputs rather than a relay output.

10.10 Display

A common 16x2 LCD character display. This display was chosen for its low cost and availability, as well as its proven ruggedness in outdoor applications. These displays have been available for decades, and should be for decades more.

This display is available in a range of colors with light on dark, or dark on light characters if you prefer. There are also OLED and fluorescent displays that use the same interface and could also be used in place of the LCD version.

Using the eleven pins set aside for the display it would be perfectly possible to adapt a wide range of other displays to the design including full graphic displays, support is available in the Arduino libraries for a number of options. This would require a new version of the userCtrl.cpp code.

10.11 Input Encoder

For local user input a rotary encoder is used, a single component to allow a capable user interface. This device may be rotated to scroll a value up or down, and can be pressed to accept the entry. To complement this encoder a daily large knob was designed that can be easily spun and pressed.

11 Networking

The Symbrosia Controller was designed from the start as a network appliance or IoT device. The first versions used an ESP8266, though this was soon replaced by the current ESP32 S3 network controller. Much of the useful functionality comes with this, but some of the concerns as well.

11.1 DHCP and IP Addresses

All access to a particular controller is done by IP address. DHCP servers typically assign the next available IP address to the controller at login to the network. This can result in different IP addresses being assigned each time a controller reboots resulting in lost connections for any control software. A fixed IP address is required if accessing the controller from a remote computer.

Most network gateways offer a feature to permanently set aside and assign a fixed IP address to a particular device. This is done by fixing the IP address to the unique hardware MAC of the device. This does require administrative access to the network gateway and DHCP server, see your network administrator or refer to the manual for your network gateway to understand how to do this.

11.2 Private Network

A substantial number of controllers and other internet connected hardware on your company network can begin to consume bandwidth and IP addresses. If a large number of controllers are to be deployed it is helpful to create a separate network for these devices. A separate WiFi access point with its own gateway and DHCP server is advisable, this creates a private network for your aquaculture cultivation area. The control server running your support software or SCADA software should also be connected to this private net.

Such a network adds an additional layer of security to your aquaculture controllers preventing possible issues, preventing outside access to these devices that often have little or no network security. Placing the controllers on a private network with a tightly configured gateway allowing only outgoing connections (such as NTP requests or posting of data) can solve some of these concerns

Keeping access to the private network tightly controlled should prevent most access to critical cultivation equipment.

Selection of a modern gateway that supports virtual private networks (VPN) allows secure remote access to this cultivation network from another location outside of the network.

11.3 Security

One issue with ModbusTCP is its very open communication protocol. Modbus was originally developed for hardwired serial interfaces where remote access simply was not a concern. While this makes it easy to deploy and maintain it is completely open to malicious access.

A secure version of the ModbusTCP protocol is available for the ESP32 but not currently implemented in the controller. This would add a nice layer of protection for any enterprise applications, this may also create a large stumbling block for students or other users with modest skill sets. As such, it has been decided, for now, not to implement secure access for the Modbus interface.

More information at https://github.com/emelianov/modbus-esp8266

11.4 WiFi

The ESP32 network modules in the Symbrosia Controller use the 2.4GHz WiFi band to communicate. While the 2.4GHz band does not offer the higher bandwidth of the 5GHz band, it does offer better range. High bandwidth is not a concern for this application, reliable service is.

The ESP32 controllers use WPA2 encryption for access to the wireless network. WiFi passwords may be written to the controllers but may not be read from the controllers.

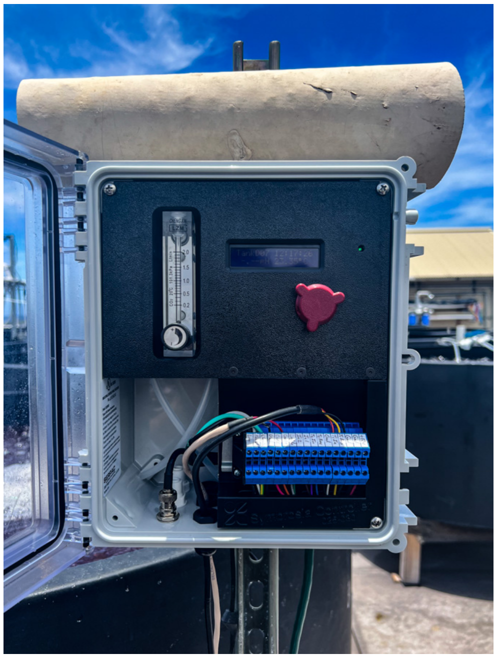
It is recommended that a good quality outdoor wireless access point with external antennas be used to provide network coverage to the cultivation areas. The typical outdoor operation of aquaculture can be reliably covered with such a system.

It is also noted that large tanks of water are very, very good at blocking WiFi signals. Both WiFi access points and the controllers should be located above the surrounding aquaculture tanks with a clear line of sight between. Locating the controller above water level also serves to keep backflow from coming through CO2 lines or other tubing and getting into valves. (Ask us how we know this;)

12 Examples

Three example setups are fully documented and ready for use. Schematics, bills of material, and setup files are provided, look for the example folders to locate the complete documentation.

12.1 A Bioreactor Controller



The primary purpose for which the controller is intended is a pond, tank, or bioreactor controller for algae or fish cultivation. Both pH and temperature will be measured and controlled. Additionally pH control will be shut off at night to reduce CO_2 consumption.

This example uses two control loops, one for pH and another for cooling. For pH control a CO_2 valve will sparge gas into the bioreactor or tank to lower the pH. The CO_2 valve will be on Digital Out 1, while a cooling valve will be connected to relay 1.

This setup is well tested and in use at Symbrosia with dozens of controllers in operation.

12.2 A CO2 Tapping Point Monitor

Monitoring the CO_2 supply can be critical to operational success in an algae grow operation. The example here is a simple sensor monitor used to detect a low CO_2 supply condition.

CO_2 is normally compressed to a liquid form in tanks or cylinders. This allows a large amount of gas to be stored in a relatively small volume. This also means that simple pressure monitoring will not allow estimation of the amount remaining in the vessel as the pressure will remain relatively constant, varying only somewhat with temperature, until all liquid CO_2 in the vessel has evaporated and the vessel is near empty.

In most operations, short periods of time without CO_2 will not substantially harm the cultures. An alarm generated when the vessel is nearly out will suffice. A simple pressure transducer with a 0-10v or 4-20mA output, such as the Automation Direct SPT25-10-1500D would provide a simple solution.

If CO_2 supply is more critical a scale or load cell may be mounted underneath the CO2 cylinder and an analog signal supplied to the controller for monitoring. This will allow direct measurement of the amount of CO2 remaining in the vessel by weight as well as daily usage information. Many commercial scales can provide an analog output, usually 0-10v that may be connected to the controller.

In this case the SCADA software can be used to monitor the alarm and relay a warning to operators. Many SCADA packages have options to generate e-mail or phone text alerts to selected staff.

For this example it is assumed that short periods without CO_2 will not harm the operation, thus only the low pressure distribution piping is monitored. If the pressure falls below 10PSI an alarm is generated. To accomplish this a 0-50psi pressure gauge with a 0-10v output will be connected to analog input 1. A single control output is used to activate a "change cylinder" light at the tapping station. Control software reading the pressure remotely can also be used to send some form of alert such as an SMS text message or email to staff to request service.

12.3 Weather Station

Accurately measuring precipitation around open culture ponds is critical to maintaining water quality or salinity. In this example a tipping bucket rain gauge is connected to one of the digital inputs. Attaching the counter to this input provides a direct reading of precipitation.

These tipping bucket gauges use a simple switch contact to indicate each tip of the double bucket underneath the funnel. Each tip of the bucket, or each switch transition is equal to a fixed amount of precipitation, usually 0.01 inch per tip.

Using the internal software counter, reset each day, allows a simple measurement that can be used for pond management.

13 Appendixes

13.1 Specifications

The controller is designed to provide for the most common aquaculture needs, in PH monitoring and controls and temperature monitoring and control. Many of these could be modified for different voltage ranges or applications with simple circuit board changes. Shown here are the specifications for base design as specified in the BOM and published software.

Power:

- Typical: 12vdc @ <275mA
- Allowable: 9-15vdc (9-35vdc with U2 switching regulator)
- Polarity: Protected to 40v

Inputs:

- 1 x standard water quality probe with BNC connector
- 2 x Pt100 temperature sensors, three wire
- 2 x 0-10Vdc or 4-20mA analog inputs (shared inputs with digital inputs 1 and 2)

Outputs:

- 2 x isolated relay outputs, up to 30Vdc @ 2A or 240Vac @ 5A
- 2 x open collector digital outputs 40Vdc @ 1A

Control Loops

- 4 control loops, each loop may take any analog input and control any output

Communications

- Remote: 2.4GHz WiFi in station mode
- WiFi: 802.11 b/g/n
- Bit rate: 802.11n up to 150 Mbps
- RF Power: up to 21dBm

Physical

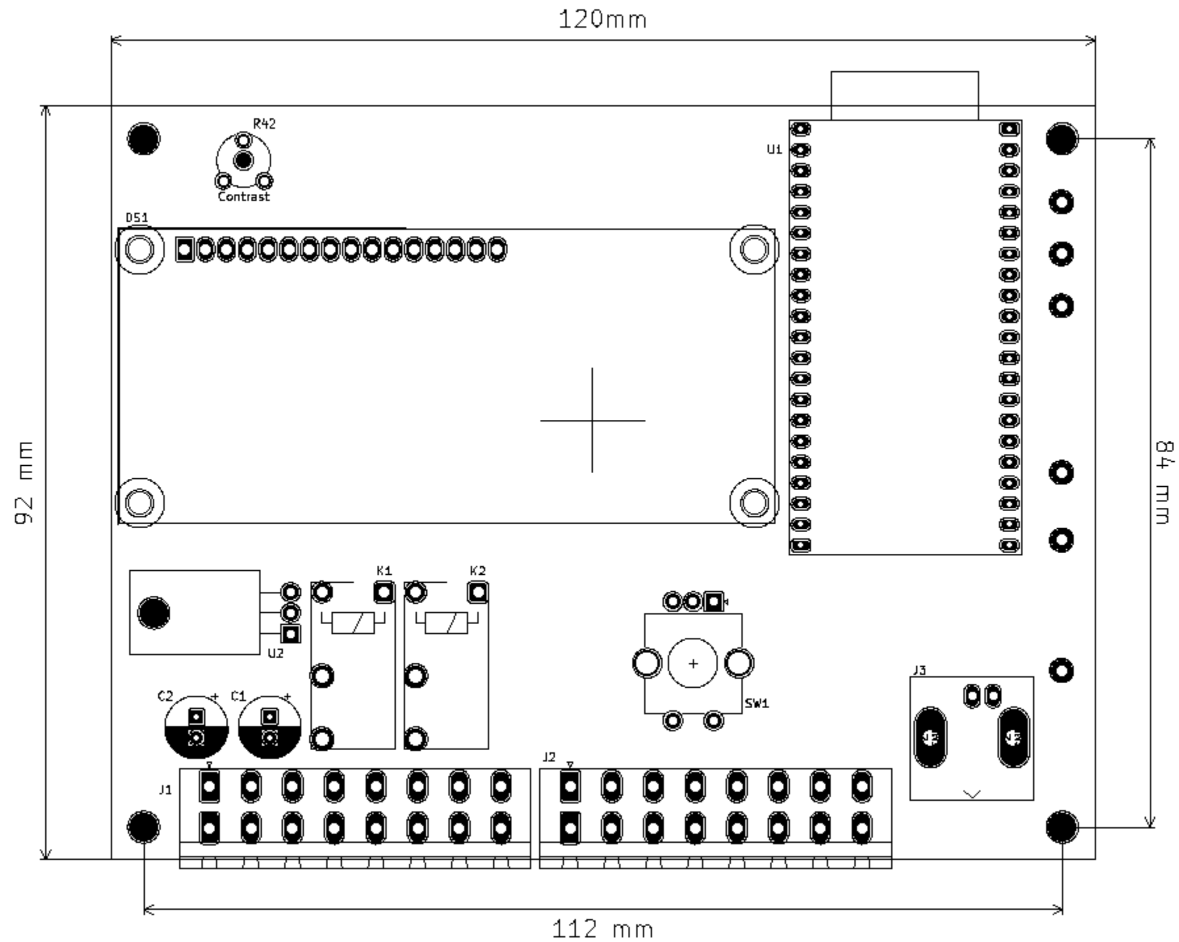
- Dimensions: 139 x 134 x 37mm (5.47 x 5.27 x 1.45in)
- Weight: ??g (??oz)

Environmental

- Operating temperature: -20 to 40°C (display may become difficult to view at extremes)
- Storage Temperature: -20 to 60°C
- Humidity: 0 to 90% non-condensing

13.2 Mechanical

The PCB measures 120 x 92mm and is 1.6mm thick. All components on the front besides the encoder shaft are less than 16mm high if the parts specified in the BOM are used.



13.3 Schematic