Technical Description

Green Water System

The Green Water Control application is a real-time control application for intermittent delivery of accurate dosages of any fluid substance. It features a SCADA-like interface. It can control up to three separate dosage loops, delivering up to 3 different substances to the same or different tanks. The user controls the rate of dosage delivery by telling the program how many seconds each solenoid is open for. The program is designed to work with the National Control Devices ProXR command set. The ProXR is a PLC. The program monitors the activity of the PLC and provides close supervision of the PLC. Timing is software-controlled. The program has limited fault-tolerance with a fail-but-safe operation mode and real-time logging of errors. In addition, the software automatically attempts recovery after failure, and thus can automatically recover from most failure states.

Feeder System

The Feeder controls delivery of live feeds to tanks. The Feeder GWC application is based on the PLC interface used in the Green Water System. The feeder program is in essence a peristaltic pump multiplexer (one pump & one feed to many tanks) and scheduler. Both the Greenwater system and the Feeder use the same software (class) to interface with the PLC. The difference is that the Feeder's fail-safe mode is not designed for error recovery. Instead, over-feeding is prevented by shutting the pump and the valves. This action ensures feed is available the next morning and that no single tank gets overfed. One could describe this as a fail-and-shut-down system. The feeder program is also less precise: Feeding times are measured in up to half-minute increments.

Similarities between the programs

Both programs require an Assignment file and a timing file. The assignment file is written in a common language while the timing file is specific to each program. Assignment files are exchangeable between the two programs. Both programs feature a SCADA-like interface and the time each valve is open can be changed by editing then saving the timings of the tanks. A log file is generated for each run. Error recovery is different for each of the two programs but the log files stores similar information about the errors and any recovery attempt. I have tested both programs and I have found them to be exact within 500ms or so. Although in theory the Window's Executive can preempt the program's execution and screw up the timings, we have not encountered this problem. Previous versions of the program had that vulnerability, but I was able to eliminate it in this second revision which we have been using for two years.

General Instructions

Valid for both the Greening and the Feeder program

How to change relay assignments

In the board, relays are organized in relay banks of eights. For a board with 16 relays, there are 2 banks (1 and 2) each with 8 relays. Each board is different, so you need to look at the board, but usually they are numbered from the bottom to the top.

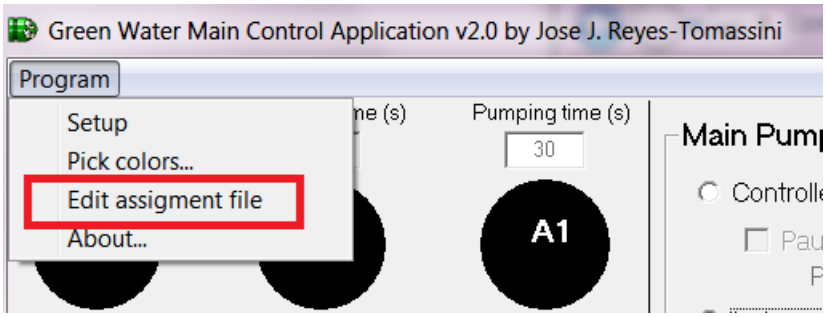
By contrast, the program numbers the relays not by banks but by a logical numbering system (e.g. 1 thru 24). In the program, relay number 1 is the first relay on the first bank, while relay 9 would be the first relay on the second bank. Thus, logical relay 16 would be physical relay 8 on bank 2.

If you are having a hard time figuring out the logical relay number, just use the following formula:

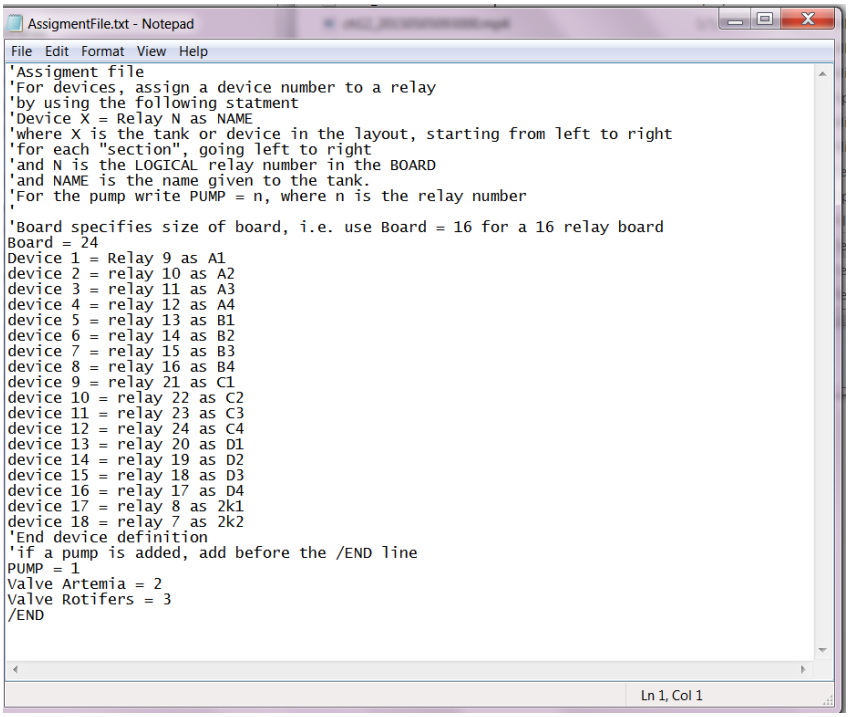
(BANK-1)*8 + Physical Number = Logical Relay number

When assigning relays to different solenoid valves, one only needs to know which relay the valve (or pump) is connected to.

You can edit assignments by going into **Program** menu and selecting **Edit assignment file**.



The program will open an instance of Notepad with the assignment file:



The assignment file is read by the program *at the beginning of each session* so <u>any changes will not take place until you exit the program and re-open it again</u>.

The assignment file is a kind of macro language for the program. Statements are NOT case-sensitive. The valid statements are:

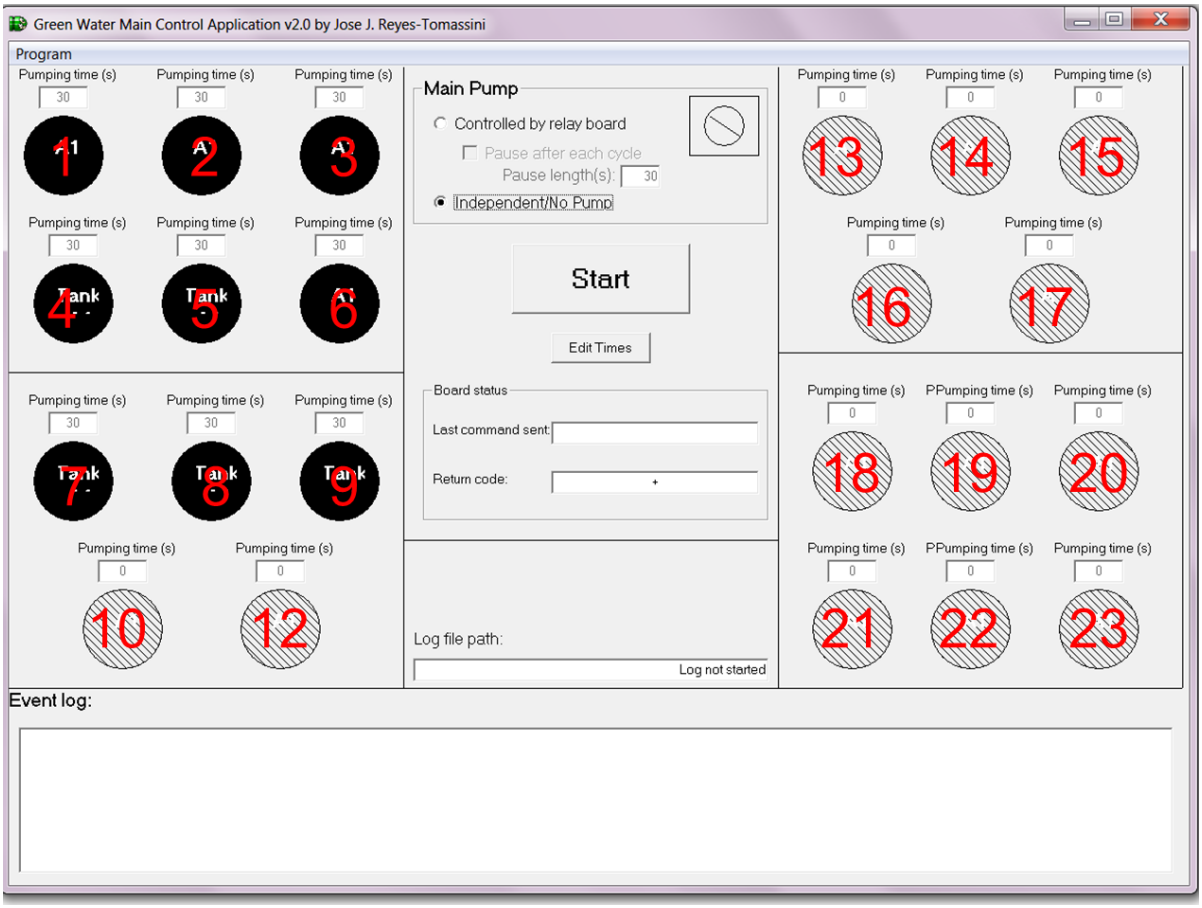
Statement	Function	Required
`	Remark. Anything in the line is ignored. This is used to write comments on the file	
`Board = N`	Sets the maximum number of relays to N. You should match this to the board model you have.	X
`Device N = Relay R as` *SSSS*	Assigns the device N in the program to the relay R (logical number) in the board and sets the name of the device to the string *SSSS*. See the section **Device Numbering** below.	
`Pump = R`	Assigns the pump to relay R. You do not need to have a pump assigned for the Green Water system.	X[1]
`Valve Artemia = R`	Only used by the Feeder. Assigns a "feeding" Artemia valve to relay R (logical number). Used when feeding more than one feed type.	
`Valve Rotifer = R`	Only used by the Feeder. Assigns a "feeding" Rotifer valve to relay R (logical number). When this feature is not used the rotifer valve should go to a valid relay NOT connected to any valve.	X[1]
`/END`	Place at end of file	X

[1]Needs to be in the assignment file for the FEEDER. This statement is ignored by the GW.

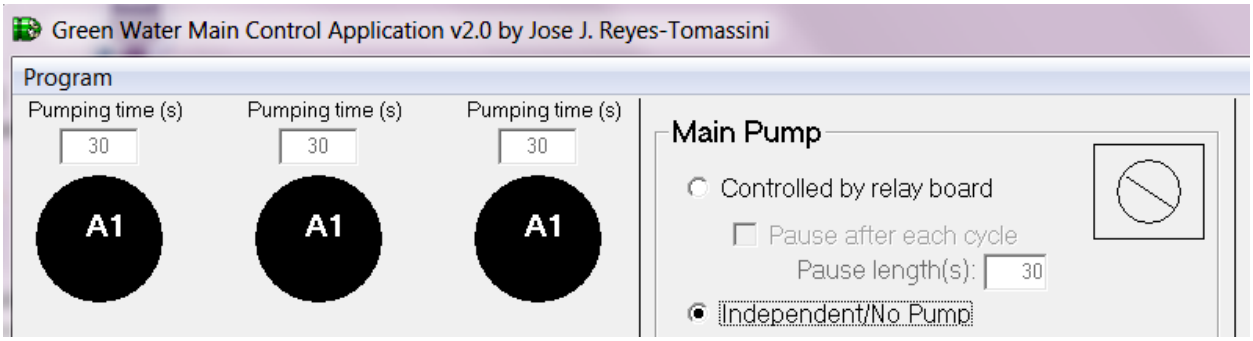
Device Numbering

The program's assignment file uses a device numbering scheme that assigns a "tank" to a relay. This ordering of the devices corresponds to what you see on the program's main screen. The general rule is that DEVICE 1 is on the leftmost and uppermost corner of the screen, and the last device is the one on the rightmost and bottommost corner of the screen.

Below are the device numbers for the Green Water system superimposed on a screencap of the program:



For example, assume you start with a series of unassigned tanks:

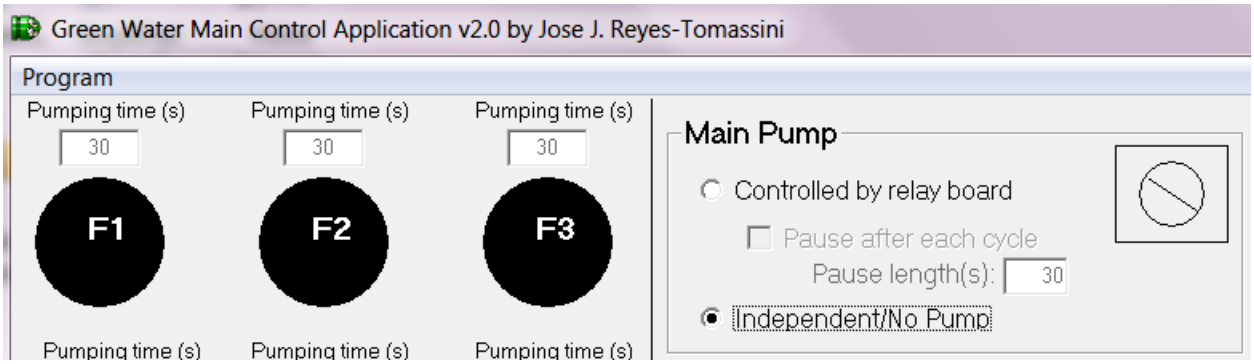


To change the tank names to F1, F2 and F3, you would write the following assignment statements. Note the use of the **/END** statement as the last line in the file.

```
Board = 16
'This only labels the first 3.  Note that the /END is crucial for this to work!
Device 1 = Relay 2 as F1
Device 2 = Relay 5 as F2
Device 3 = Relay 8 as F3
/END
```

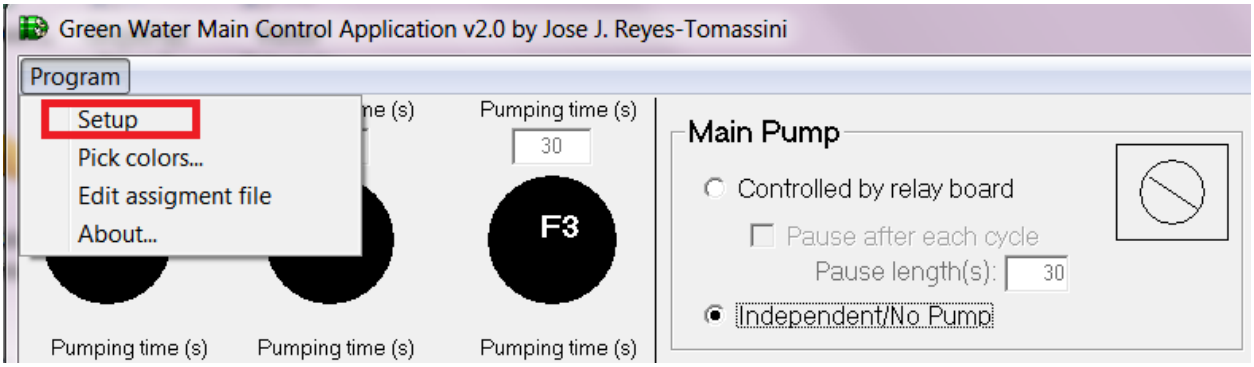
Save the file, quit the program and exit it. Make sure that you have directed the program to the appropriate assignment file (see **Setup**).

When the program is run again, you will see the new assignments:

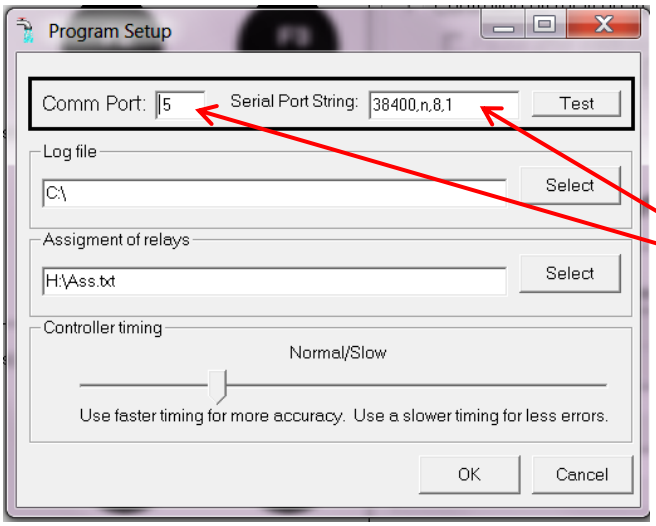


Setup

Both the feeder and the greenwater program have a Setup window. To access the program setup, go to the **Program** menu and click on **Setup**.



The Setup window will be displayed.



This information is important when configuring the program for the first time in another computer!

This window allows you to change the Comm port, comm port settings, etc. You can also test the communication with the board by clicking on the **Test** button.

Please note that the **Serial Port String** by the **Test** button contains the *baud rate*, *parity*, *data bits*, and *stop bits* (respectively) for the serial port. If you run the program in a different computer but with the same card, make sure to copy this string to the new computer's GW or Feeder program. Also, make sure that if you configure a new computer, you set the correct communications port (**Comm Port**). The communication port will be different in different computers. Consult IT to figure out which communication port you need to use.

If you press the **Test** button, the program will attempt to establish successful communication with the board. If it fails to communicate, it will begin to cycle through increasing baud rates until it reaches one that works.

Note that once a test cycle begins, you need to wait for the cycle to complete. You can use **Test** if you are not sure which baud rate to use for the card. The baud rate is card-specific. Refer to the card instructions for more information on how to change the baud rate in the card. This usually involves moving jumpers or DIP switches.

You can select the directory and filename of the LOG FILE. During each communication session between the computer and the board, a log file is generated. There is an important difference between what the GreenWater program records and what the Feeder records on the log file.

GreenWater: Reports the ERRORS that occur in a session. Any abnormal behavior of the relays or the board is recorded. If no errors occur, then the log file is empty except for the time stamp corresponding to the beginning and the end of the session.
Feeder: Reports both errors and general session information such as the commands sent to the board, etc.

You can also select the Assignment file to be used for assigning the relays to the valves. Note that if you get the error that there is no valid assignment file, you must go to the **Setup** menu and select an assignment file. If you don't have a valid assignment file in your computer, you must create one and then point to where it is located by going into the **Setup** menu and choosing the location of the file.

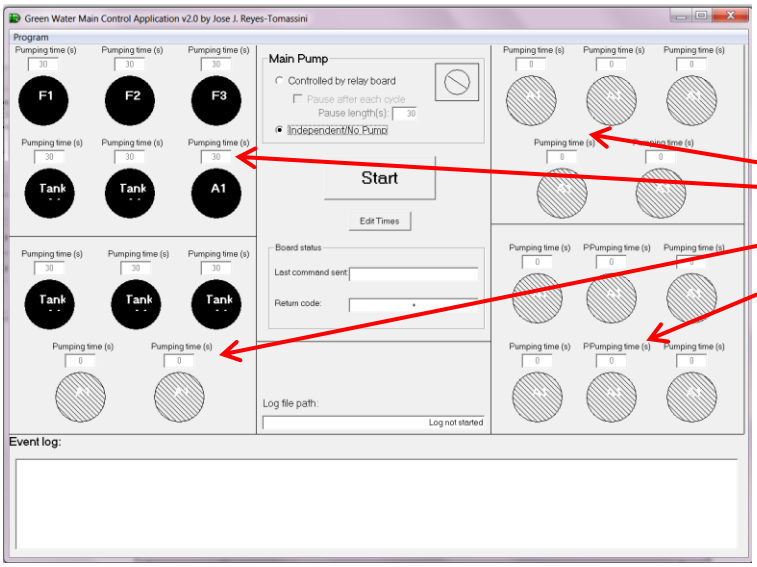
Controller Timing

You can change the amount of time the computer waits between sending a command to the relay board and receiving a response. I would say leave this alone unless there are problems specific to the configuration of the board (i.e. a new computer or a new board). Otherwise, it is better to not change it from what I have it set.

Setting up more than one timing loop

(Green Water System ONLY)

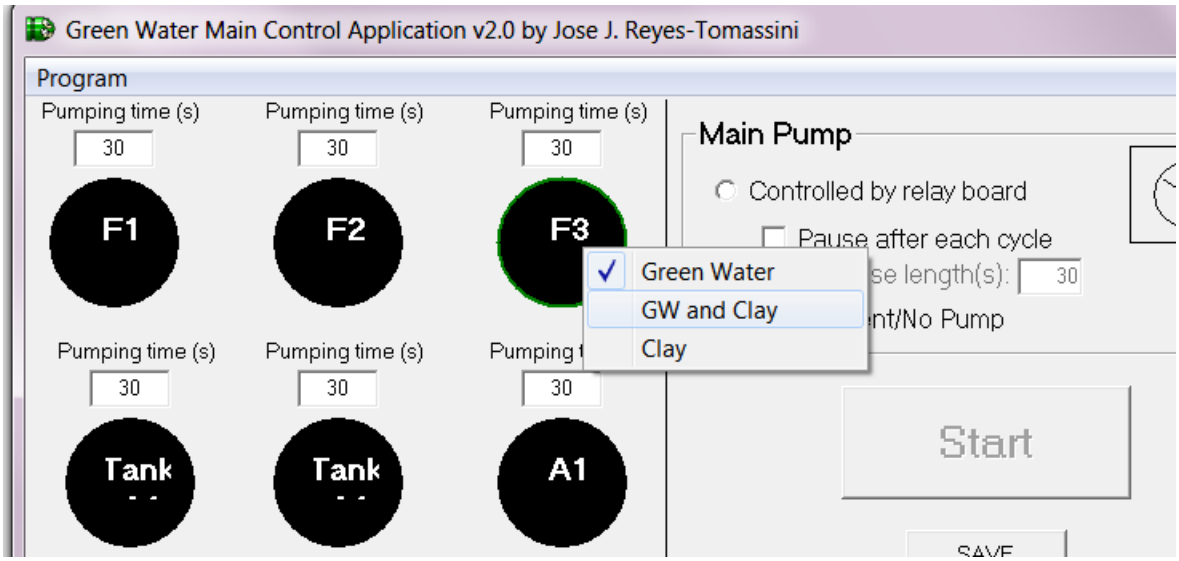
The program has the capacity to control more than one pump loop (e.g. green water and clay). The loops can include any number of tanks. However, I would suggest they are set according to the lay out in the screen. In any event, <u>the screen layout does not correspond to the loops</u> unless you configure the tanks within that layout. Instead, all the tanks are set for the green water loop by default. What does this mean? See below:



By default, this are not assigned to different timing loops! You need to assign each tank INDIVIDUALLY to a timing loop…

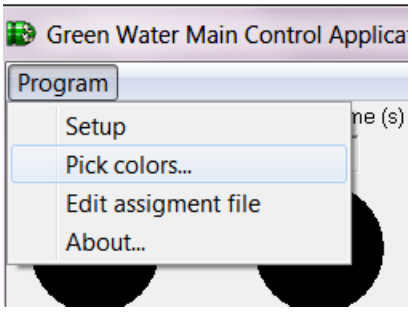
To assign a tank to a timing loop:

1) Click on the **Edit Time** button located in the central area of the main screen (below **Start**)
2) Move the mouse cursor over the tank you would like to add to the loop
3) Right-click (that is, use the "other" button to click over the tank) while the cursor is over the tank's ID
4) Select the loop you would like this tank to be in. There are three loops:
    1. Green Water (default)
    2. GW+Clay
    3. Clay
5) When you are done with the changes, click the **SAVE** button (where **Edit Time** used to be!)

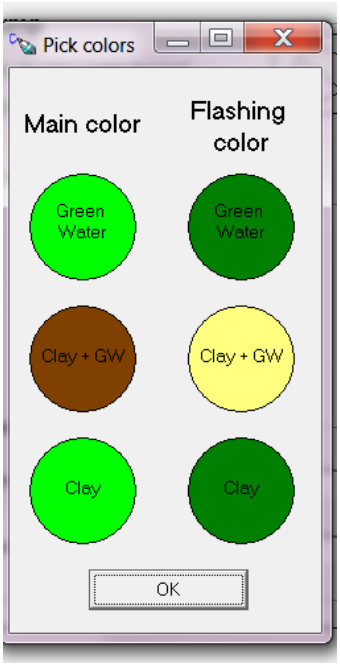


Each loop is identified in the layout with a color. You will notice that once you select a loop, the tank's border will change color to indicate which loop it corresponds to. When the program runs an instance of the timing loop, the tank will be identified with that color.

You can change the loop's colors by going to the **Program** menu and clicking on **Pick colors**.



This will bring the **Pick colors** window which is pretty self-explanatory. Click on any of the tank icons to change the color of the corresponding loop/state. When you are done, click **OK**.



Note: <u>The assignment of tanks to loops is stored on the Timing file</u>. Thus, if you haven't created a timing file (which the program will let you know on startup), all the tanks will be set to the default loop (Green Water).

Known Issues and Program Idiosyncrasies

*Each time the program starts an instance, it runs loop 1 while it places the other loops on "hold" (if there is more than one loop). This is done to prevent the relays from closing all at once which we thought at one point, could send a high-voltage jolt to the board. We no longer think this is the case but the feature is there…

*You will occasionally see an error that says a relay didn't turn on or off. If it happens always on the first relay or the last relay in the loop, it is probably nothing and should be ignored. This is something I never got around fixing. It has something to do with the "verify" function and what it expects at the end of a loop… Was thinking of fixing it this year!

*The computer sends a "verify" request to the board every so often. This is controlled by the timing option on the **Setup** window. The board keeps an internal table of the status of the relays. The program also keeps track of the status of the relays. If the table in the program's memory and the table in the board do not match, the program flags this as an error. This will happen when the board's power is cut (e.g. during an outage) or when the board's information gets scrambled with any kind of surge.

*if the "verify" function returns an error (e.g. the table's don't match), the computer will request the board to reset all the relays and turn them to the correct status. These actions will be shown in the log.

*If the above fails, the computer will issue a command to the board that "resets" the communications, flushes its buffer, etc. If that fails, the communication port itself is reset and a new communication port is established.

*When the above fails, the program will halt, but that has never happened unless the communication cable itself is damaged or the computer is damaged.

*There is no provision for the computer to restart the program if the computer restarts. For this reason, I have avoided hooking the computer up to the internet (as IT could in theory push an update that can cause the computer to crash). We could provide this future (start after a reset)…

*The highest chance of encountering fatal errors is when you set up a new computer or try to change the board. I suggest you let me know beforehand and I can provide assistance in installing the program and setting it in a new system… Other than that, I think between IT and what Doug and Andy can do, you can probably figure it out yourselves… It is not that complicated.