



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

www.procerius.com

The IO Geek™ System

Hardware and Software



By Procerius, Inc.

412 851-5100

sales@procerius.com

support@procerius.com

<http://www.iogeek.com>



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

Table of Contents

Introduction.....	3
Intended Use	3
IOGeek Hardware.....	3
PC Requirements	3
Software.....	3
Options.....	3
Getting Started.....	3
Hooking up the hardware.....	4
Board layout – identifying the board.....	4
FIGURE 1 – Board layout Revision 1 and Revision 2.....	4
Board layout – connections	5
Example Connections – Rev 1 Diagram 1.....	6
Example Connections – Rev 2 Diagram 2.....	7
Example Connections – Rev 2 Diagram 3.....	8
Power Packs.....	9
Rev 1 board 12 VAC	9
Rev 2 board 800 mA power pack 12 VDC for board and low power solenoids.....	9
Rev 2 board 1500 mA power pack 12 VDC for board and most 12 VDC solenoids.....	9
Grounding.....	10
Installing the software.....	10
Basic Wiring	10
Hardware References - Setting up Your Display	12
Programming sequences	12
Scripting Language.....	12
Setting outputs	12
Interlocks	12
Reading Switch Inputs.....	13
Playing Sounds	13
Configuring the HMI / GUI Screen.....	14
Structuring Your System and Sequences.....	15
Trouble Shooting	17
Outputs turn off unexpectedly	17
Can't communicate.....	17
Precautions.....	17
Wiring – System Design.....	19
Water.....	19



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

Introduction

Thank you for your interest in the IOGeek system. This system is based on technology from Procerius, Inc. for industrial controls and automation. This manual will describe some sample uses for the board but you are not limited by the examples in this manual. The use of this system requires knowledge of basic principles of electrical wiring and general safety principles in electrical systems. The user is fully responsible for all aspects of safety including design, testing, and human factors when using the IOGeek system, see precautions below.

Intended Use

This system is intended only for control of low voltage systems, specifically 12 VAC landscape lighting systems that are properly protected using ground fault indication (GFI) wiring, kept dry, and no exposed wiring. **IMPORTANT:** See precautions below for important safety information.

IOGeek Hardware

The hardware is a specially designed circuit board for sensing inputs and controlling outputs. Each board that you hook to your computer requires one serial port. The hardware may be mounted remotely, away from the computer.

PC Requirements

This software runs on the Microsoft Windows operating system platform. A minimum system requirement is 800 MHz, 128 MB RAM, 1024x768 graphics. Windows 2000, XP, or higher.

Software

The software is designed for sequencing the outputs based on easy to write programs and based on input readings and user commands. Up to 16 IOGeek circuit boards can run from one PC, as long as you have 16 serial ports. The software is designed to provide multitasking and has a script based programming language so that a single computer can easily be programmed to automate the entire landscape.

The software is a limited license of our automation control suite. The VLCH2 IOGeek card does not require a full license of the software yet you have access to all of the features of the full registered version for the IO on this card.

Options

This system is expandable to include other types of IO, see www.procerius.com for more information.

Getting Started

To get started, connect the hardware, install the software, power up and have fun! The steps are outlined below.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

www.procerius.com

Board layout – connections

There are three connectors, J2 is a serial port connection to your PC, Terminal Strip TS1 is for the power source input and relay switch closures; Terminal Strip TS2 has the connections for the four inputs and 4 solenoid driver outputs.

J2:

To hook up the hardware, connect the PC to the card using a straight through Male to Female DB9 to DB9 serial cable. The equivalent Radio Shack part number is: 26-117. The serial cable you use should not exceed 100 feet.

Revision 1 board connector TS1	Revision 2 board connector TS1
Pin 1: Relay contact SW 1, 10 Amps maximum	Pin 1: Relay contact SW 1, 10 Amps maximum
Pin 2: Relay contact SW 1, 10 Amps maximum	Pin 2: Relay contact SW 1, 10 Amps maximum
Pin 3: Relay contact SW 2, 10 Amps maximum	Pin 3: Relay contact SW 2, 10 Amps maximum
Pin 4: Relay contact SW 2, 10 Amps maximum	Pin 4: Relay contact SW 2, 10 Amps maximum
Pin 5: Relay contact SW 3, 10 Amps maximum	Pin 5: Relay contact SW 3, 10 Amps maximum
Pin 6: Relay contact SW 3, 10 Amps maximum	Pin 6: Relay contact SW 3, 10 Amps maximum
Pin 7: Relay contact SW 4, 10 Amps maximum	Pin 7: Relay contact SW 4, 10 Amps maximum
Pin 8: Relay contact SW 4, 10 Amps maximum	Pin 8: Relay contact SW 4, 10 Amps maximum
Pin 9: Power 10 to 12 VAC	Pin 9: GND
Pin 10: Power 10 to 12 VAC (Bottom right of figure 1)	Pin 10: GND
	Pin 11: GND
	Pin 12: +12 VDC Power, 800 mA min.

To use the relays, when you set Relay0 to 1 in the software, Pin 1 and Pin 2 on TS1 will be shorted together just like a light switch. When Relay0 is set to a 0 pins 1 and 2 will not be connected and the circuit will be open.

Revision 1 board connector TS1	Revision 2 board connector TS1
Pin 1: Input 1, short to ground (pin 10) to activate	Pin 1: Input 1, short to ground (pin 10) to activate
Pin 2: Input 2, short to ground (pin 10) to activate	Pin 2: Input 2, short to ground (pin 10) to activate
Pin 3: Input 3, short to ground (pin 10) to activate	Pin 3: Input 3, short to ground (pin 10) to activate
Pin 4: Input 4, short to ground (pin 10) to activate	Pin 4: Input 4, short to ground (pin 10) to activate
Pin 5: SolenoidOut0, Open Drain (Pulls to gnd) 250 mA *	Pin 5: SolenoidOut0, Open Drain (Pulls to gnd) 350 mA *
Pin 6: SolenoidOut1, Open Drain (Pulls to gnd) 250 mA *	Pin 6: SolenoidOut1, Open Drain (Pulls to gnd) 350 mA *
Pin 7: SolenoidOut2, Open Drain (Pulls to gnd) 250 mA *	Pin 7: SolenoidOut2, Open Drain (Pulls to gnd) 350 mA *
Pin 8: SolenoidOut3, Open Drain (Pulls to gnd) 250 mA *	Pin 8: SolenoidOut3, Open Drain (Pulls to gnd) 350 mA *
Pin 9: Power Output, do not use.	Pin 9: +12 VDC Out
Pin 10: Gnd	Pin 10: +12 VDC Out
	Pin 11: GND
	Pin 12: GND



PROCERIUS, INC.

SYSTEMS & SOFTWARE

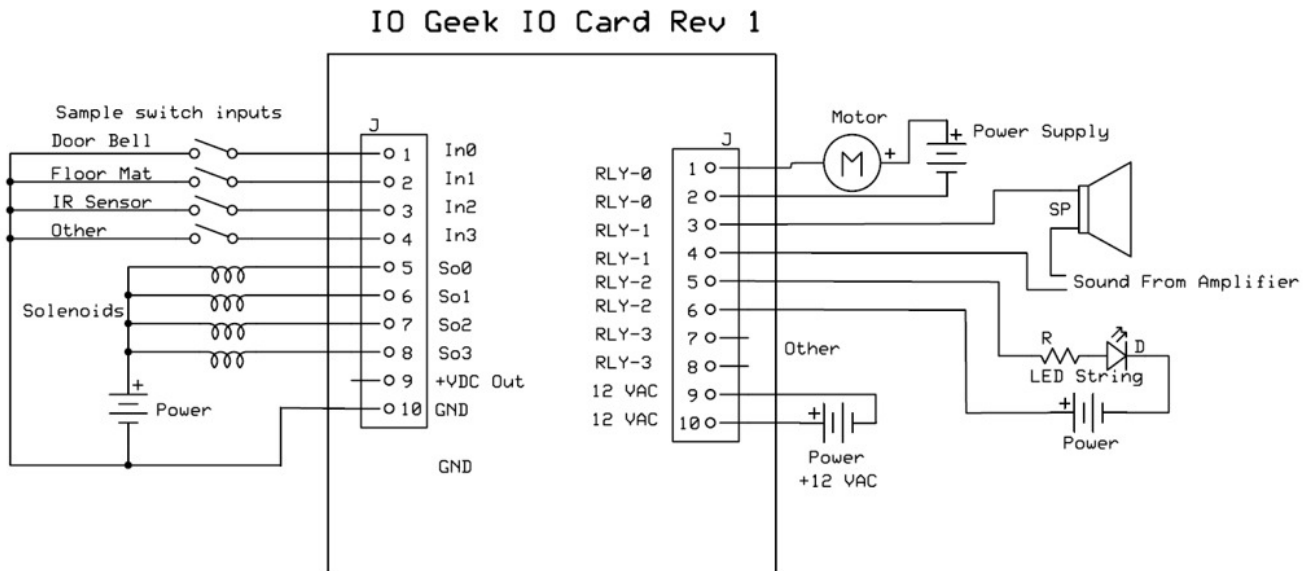
PO Box 156
Bethel Park, PA 15102
(800) 936-5280

www.procerius.com

- Solenoid output drivers, when on the driver will pull to ground; otherwise the connection will be open. This solenoid driver is designed to control pneumatic actuator valves. In general, you supply the power using a separate power supply for the solenoid. Most solenoids for this purpose are 24 VDC but depending on the devices you use, you will need to ensure that it is designed properly. If you have low current 12 VDC solenoids, no external power supply is required to drive your solenoids; you can use the power from the Revision 2 card TS2 Pins 9 and 10.

Example Connections – Rev 1 Diagram 1

IOGeek Revision 1 board (yellow board).



The rev 1 board requires the use of the provided power pack for board power. Use a separate power pack for external loads, such as solenoids – as shown above.



PROCERIUS, INC.

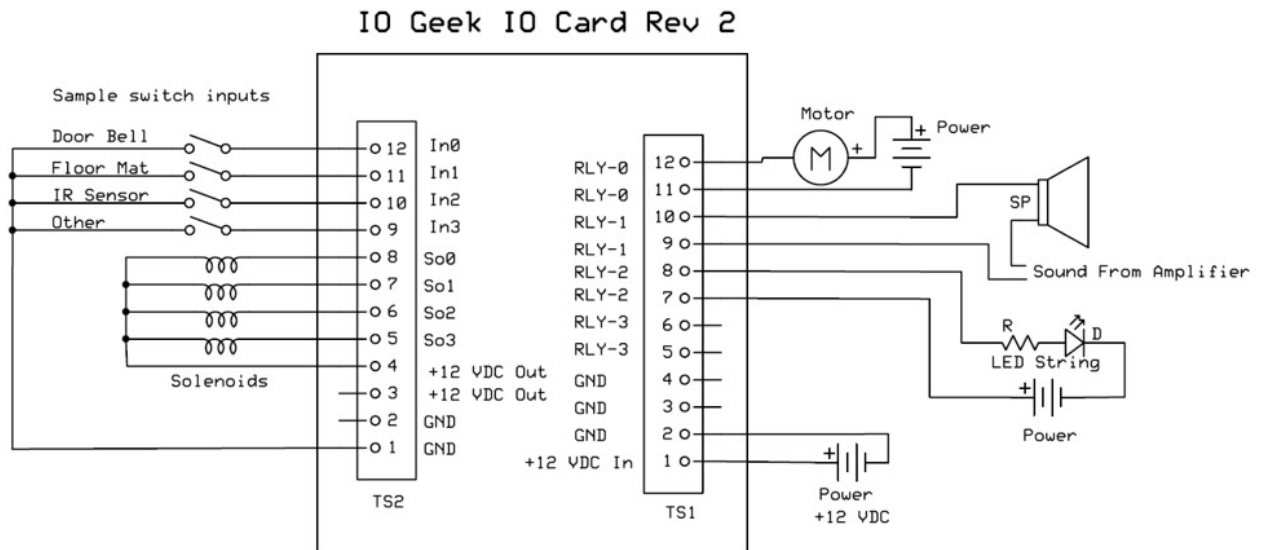
SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

www.procerius.com

Example Connections – Rev 2 Diagram 2

IOGeek Revision 2 Board (green board) with a single power supply for solenoids and the IOGeek



The revision 2 board can power 12 volt solenoids directly, if the +12 VDC power pack has sufficient power to power both the board and the solenoids.



PROCERIUS, INC.

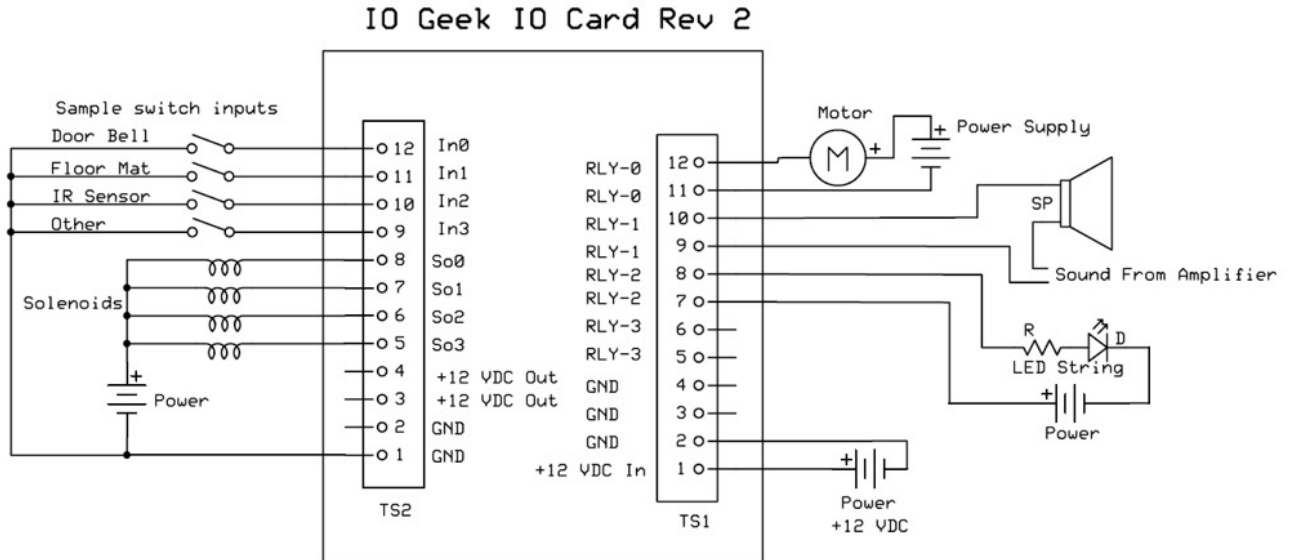
SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

www.procerius.com

Example Connections – Rev 2 Diagram 3

IOGeek Revision 2 Board (green board) using separate power supply for solenoids



If you require an external power source for solenoids use the above connection diagram.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

Power Packs

This section applies only to power packs supplied by Procerius, Inc.

Rev 1 board 12 VAC

The power pack provided with the yellow revision one boards have three wires, Red, Yellow and Green. The Red and yellow wires are the 12 VAC connections. The green wire is ground.

Rev 2 board 800 mA power pack 12 VDC for board and low power solenoids



The 800 mA power pack is for the Rev 2 IOGeek and up to 500 mA of external 12 VDC hardware.

Using this power power pack you can drive four Minimatic pressure valves (60 mA ea) or two Maximatic valves (200 mA ea).

IMPORTANT: When hooking up the power supply the polarity of the wires is very important. ON THIS power pack the **wire lead with the white stripe** should be inserted into TS1 +12VDC, and the other lead is the ground and should be connected to TS1 - GND.

Rev 2 board 1500 mA power pack 12 VDC for board and most 12 VDC solenoids



The 1500 mA power pack is sufficient to drive the Rev 2 IOGeek and up to 1000 mA of external 12 VDC hardware.

IMPORTANT: When hooking up the power supply the polarity of the wires is very important. ON THIS power pack **the wire lead with the printing** should be inserted into TS1 +12VDC, the wire lead with the ribs is the Ground (GND) and should be connected to TS1 - GND.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156

Bethel Park, PA 15102

(800) 936-5280

Grounding

While not always necessary, in some situations it will be required to connect the ground of the IOGeek to earth ground. Earth ground is the center lug on a three prong outlet – the big round one. To be safe consult an electrician or electrical expert. It should only be necessary to ground the IOGeek if you are working with higher voltages or if you are in a very (electrically) noisy environment.

Installing the software

The software can be downloaded from <http://www.iogeek.com/downloads> you will be provided the username and password with your purchase.

Download the software, setup.exe and run that on your PC.

The software provided comes configured to work with a single IOGeek circuit card. You may, however need to change the com port, which is by default set to com 1. To change the comm. port, in the TRACER WINDOW (bottom right) click on Devices->View. Click on serial ports and change the comm. port to the correct comm. port number under the “Serial ports” box at the bottom of the screen. Then click on the VLCH2 and change the comm. port to the correct number. Right click and choose Reload devices and IO.

To determine the “correct” comm. port, open the device manager by right clicking on my computer on the desktop. Choose Manage then Manage Devices, then look for Ports. Expand Ports and you will see the list of serial ports numbers. If you want to test the IOGeek Card with a communications program like HyperTerminal, the baud rate is 115,200, 8N1. When you press the enter key you should get the following or similar response:

Procerius, Inc. www.procerius.com, VLCH2(TM) 1.0 Hello.

The rest of the communications protocol is unpublished; if you would like to use other software to communicate with the board, please contact us for the protocols.

Basic Wiring

Some example wiring is shown below. The power supply section is on the IOGeek board and is provided for reference only. The example connections demonstrate how to connect to control IO devices. You must verify that the electrical design that you put together is safe and that the components are compatible with each other.

The relay contacts are computer controlled switches that can close circuits up to 10 amps, it is recommended that you do not exceed 5 amps on the switch inputs, per switch contact. One great use for the relay contacts is switching on and off a speaker. The relay contacts merely need to be placed in



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156

Bethel Park, PA 15102

(800) 936-5280

series with the speaker. (<http://www.allaboutcircuits.com> for useful information on circuitry, search for series circuit or series switch.)

The solenoid outputs can be amplified using a standard solid state relay, if it is required. The solenoid drivers can drive many pneumatic solenoids without any amplification (or buffering). The outputs are thermally protected for overload and can drive up to 350 mA each output.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156

Bethel Park, PA 15102

(800) 936-5280

Hardware References - Setting up Your Display

Proposing display configurations and explaining how to setup and wire the electrical aspects of your display is beyond the scope of this document.

There are lots of display ideas on the internet and while we cannot vouch for what you might find example wiring can be discovered by searching the internet with the following search phrases as examples to get you started:

- wiring projects
- Monsterlist of Halloween Projects

Programming sequences

The programming sequences are what bring your display alive!

Scripting Language

The scripting language is a powerful programming language that enables you to control the IOGeek board. see: Help->Tutorials->IOScript->Writing A Script. There are several scripts that are delivered with the IOGeek hardware as well. You can access them through the following menus on the tracer window: IO->Edit Scripts->Select File from the drop down. In order to use any changes that you make you must save and compile your changes from the menu on the editor screen.

Setting outputs

There are a couple of useful ways to set outputs. The first way is to use the IO Screen and set an output directly. This is great for setup and debugging. When the program is running there is a tracer window in the bottom right hand of the computer monitor. On the tracer window choose IO->View->IO Points. Locate the signal, like VLCH2_1.Relay0, and set the output to 1 by pressing “Toggle” (careful! If the output is connected and energized, the output will trigger!)

Another more convenient way to set an output is to place an object on the HMI/GUI screen and attach the output signal to the object as a “Control Signal”. Please see *Configuring the HMI / GUI* below for an example.

Interlocks

Interlocks prevent a condition like “both relays on at the same time” from occurring. An interlock will block conditions from damaging your equipment, when used properly. Please refer to the movies in the help menu in the tracer window for instructions on how to setup interlocks, see: Help->Tutorials->IOPoints->Interlocks.

Interlocks are an advance topic any generally are only required in advanced systems.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

Reading Switch Inputs

Switch inputs can be read using script code or on the IO Points screen. To test your inputs you can temporarily add an object on the HMI / GUI screen and attached it to the IO Point. Please see *Configuring the HMI / GUI* below for an example. To view it on the IO Screen choose IO->View->IO Pints.

Playing Sounds

Sounds can be played through your PC speakers at your command or during automated sequencing. In each case, you must write a little script code, See *Scripting Language*, above.

Steps:

- 1> Record your sound file in a format that can be played on your computer in Windows Media player. Sound files in this format should work in this program. If you experience problems and the error message is related to the sound format use PCM 11 KHz, that is a standard format that is known to work. If you need a program to reformat, you can resave in Windows sound recorder in the accessories menu or use <http://www.goldwave.com/> for more advance functions.
- 2> Copy the file into the installation directory, usually c:\Program Files\Procerius, Inc\VacuLab\
- 3> Write your script file and run the script.

For example:

```
Routine MySound () {  
    PlaySound ( 1, boo.wav, 0 );    // this plays "boo.wav" on the first channel  
                                    // the script does not wait for the sound to complete  
                                    // the third parameter is the repeat flag, 0 is no repeat  
}
```

Note: each channel is an independent sound player; each channel plays in both the left and right speakers. All channels can play at once. You may not play the same sound file in different channels at the same time. If that is required copy the file so that each channel can use its own unique file name.

Here are some more examples:

```
Routine MySound2 () {  
    PlaySound ( 1, "fx-creak.wav", 1 ); // play on channel 1 and repeat forever  
    Dwell ( 10 );  
    PlaySound ( 1, "Stop", 0 ); // Stops immediately  
    Dwell ( 5 );  
    PlaySound ( 1, "fx-creak.wav", 1 );  
    Dwell ( 10 );  
    PlaySound ( 1, "StopRepeat", 0 ); // Stops after completing the current file  
}
```



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156

Bethel Park, PA 15102

(800) 936-5280

After you write the script you must save and compile, also you may need to add a button on the GUI screen.

There are 5 channels, 1 to 5. Five separate files can be played at the same time through the speakers of your PC.

To run the script MySound2 in the above example, just to test it: on the TRACER WINDOW (bottom right) choose IO->DebugScripts Click on IOScript Routines. Locate MySound2 (if it's not in the list you must edit the script, save and compile; IO->Edit Scripts...) right click and choose Run Selected.

Some more IOScript commands, note there is on-line help in the program documenting all commands:

```
PlayVideo ( "FileName.wmv", 0, 100, 100, 300, 300 );  
// plays video: parameters are FileName, repeat/no repeat, Top, Left, Width, Height  
// this is great for using a projector as a second monitor.  
// You can set the "Left" parameter to be in a second monitor with a black background for a desk top.  
// A dual monitor might have a desktop area of 2048x768.  
// 0 to 1023 is in the first monitor, 1024 to 2047 (for the Left parameter) is in the second monitor.  
// for instance PlayVideo ( "FileName.wmv", 0, 0, 1024, 1024, 768 );  
// would play full screen in the second monitor.
```

```
Random ( Temp10, 5, 60 ); // generates a random integer between 5 and 60 and stuffs it into Temp10  
Dwell ( Temp10 ); // Dwell for Temp10 (random) seconds
```

```
Command ( "c:\Halloween\cmd1.exe", "sendplc", "A", 1, "OFF" );
```

Once you are happy with it you can attach the script to an object on the HMI / GUI screen by using the design mode. See the next section.

Configuring the HMI / GUI Screen

The human machine interface (HMI) and graphical user interface (GUI) is configured by entering the design mode. The screen starts automatically when the IOGeek PC Software is started. Enter Design mode by clicking Utilities->Design. Right click in the design area and add an object. You can right click most objects to edit the properties but the help menu will be a good guide for editing objects.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

www.procerius.com

Structuring Your System and Sequences

Planning, that is the way to be successful. Decide what you want to do and then count the number of inputs and outputs required. Here are a few examples:

Description	Relay Outputs	Solenoid Outputs	Inputs
Cauldron with fan, light, fog switch, fog power. Fan and light together on SSR on solenoid outputs, fog power and switch are separate relay outputs.	2	1	
Skeleton with moving jaw, LED eyes, moving hand, moving head, speaker for sound. Jaw is fitted with solenoid, LED eyes are on a relay output, hand is a solenoid, moving head is on a solenoid, and speaker is a relay output.	2	4	
Flying ghost with LED eyes and speaker for sound. Assumes a single motor that is energized to stimulate motion.	1	1	
Floor mat			1
Infrared detector			1
Total	5	6	2
This display requires two IOGeek cards with:	8	8	8

The next step is to decide how to control your display. Generally speaking it is a good idea to make a single Routine for each display item. The cauldron would be one or more Routine, the Skeleton would be one or more Routines, and the Flying Ghost would be one Routine.

The reason for making separate routines is that each can run separately and at the same time. This is called multitasking. Also by doing this you can set triggers to start the subroutines so they start based on the electrical inputs or by pressing a button on the HMI program and each will run separately. It is worth taking a little time to study this example to understand the reasoning behind setting up the Routines as shown. After the system is built the timing and setup of the routines will be fine tuned based on the reaction of the system when it is in operation.

```

Routine Cauldron () {
    CauldronRunning = 0.0; // make sure it doesn't run all by itself
    FogPower = 1.0; // in this example the power will be on all the time.
    WaitFor ( 240, FloorMat == 1.0 || CauldronRunning == 1.0 ) {
        // Code here will execute as soon as the FloorMat signal is set to one OR
        // the dummy variable CauldronRunning is set to one by the user
        // left blank on purpose.
    } else {
        // code here will execute if it times out, after 240 seconds.
    }
    PlaySound ( 1, "bubbles.wav", 1 ); // play bubbles over and over, channel 1, repeat = 1
    FogSwitch = 1.0; // start the smoke
    Dwell ( 20.0 ); // let the smoke build up a little
    CauldronLight = 1.0;
}

```



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

```
CauldronFan = 1.0;
Dwell ( 60 ); // let it go for 60 seconds
FogSwitch = 0.0; // stop the smoke
Dwell ( 30 );
CauldronLight = 0.0;
CauldronFan = 0.0;
PlaySound ( 1, "StopRepeat", 0 ); // stop after the bubble.wav is done.
CauldronRunning = 0.0;
Return(); // this line restarts this subroutine so it will run over and over and over.
}
```

```
RoutineFlyGhost() {
    GhostRunning = 0.0; // make sure it doesn't run all by itself
    GhostPower = 0.0; // in this example the power will be off all the time.
    WaitFor ( 240, GhostRunning == 1.0 ) {
        // Code here will execute as soon as the GhostRunning is set to 1,0 by the user
        // left blank on purpose.
    } else {
        // code here will execute if it times out.
    }
    PlaySound ( 1, "whisper.wav", 1 ); // play whisper over and over, channel 1, repeat = 1
    GhostPower = 1.0;
    Dwell ( 20.0 ); // Run for a while
    PlaySound ( 1, "StopRepeat", 0 ); // stop after the bubble.wav is done.
    Dwell ( 5.0 ); // be sure sound has ended
    GhostRunning = 0.0; // make sure it doesn't run all by itself
    GhostPower = 0.0; // in this example the power will be on all the time.
    Return(); // this line restarts this subroutine so it will run over and over and over.
}
```

```
Routine Skeleton ( ) {
    SkeleRunning = 0.0; // make sure it doesn't run all by itself
    SkeleJaw = 0.0;
    SkeleHead = 0.0;
    WaitFor ( 240, SkeleRunning == 1.0 ) {
    } else {
    }
    PlaySound ( 1, "scream.wav", 0 ); // play scream over and over, channel 1, repeat = 1
    Spawn DoSkeleJaw();
    Spawn DoSkeleHand();
    Spawn DoSkeleHead(); // each of these routines starts simultaneously
                        // using a "Call" would cause the program to wait
    PlaySound ( 1, "StopRepeat", 0 ); // stop after the scream.wav is done.
    Dwell ( 5.0 ); // be sure sound has ended
    SkeleRunning = 0.0; // make sure it doesn't run all by itself
    Return(); // this line restarts this subroutine so it will run over and over and over.
}
```

```
Routine DoSkeleJaw() {
    SkeleJaw = 1.0;
    Dwell ( 5 );
    SkeleJaw = 0.0;
    Dwell ( 5 );
    SkeleJaw = 1.0;
    Dwell ( 5 );
    SkeleJaw = 0.0;
}
```

```
Routine DoSkeleHand ( ) {
    SkeleHand = 1.0;
    Dwell ( 5 );
}
```



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156
Bethel Park, PA 15102
(800) 936-5280

```
SkeleHand = 0.0;  
Dwell ( 5 );  
SkeleHand = 1.0;  
Dwell ( 5 );  
SkeleHand = 0.0;  
  
}  
  
Routine DoSkeleHead () {  
    SkeleHead = 1.0;  
    Dwell ( 5 );  
    SkeleHead = 0.0;  
    Dwell ( 5 );  
    SkeleHead = 1.0;  
    Dwell ( 5 );  
    SkeleHead = 0.0;  
}
```

Trouble Shooting

Outputs turn off unexpectedly

The outputs automatically turn off when the PC is disconnected or communication stops. This is a safety feature that ensures that if the computer crashes the outputs will not continue to drive.

Can't communicate

When the IOGeek is properly connected to your computer and the software is installed the green LED D3 will flash. If it is dark, the PC is not communicating.

Precautions

This system is provided as is and is not guaranteed or warranted for any particular purpose. The user assumes all liability from their application of the product and the manufacturer assumes no liability whatever resulting from the use or inability to use the system.

It is YOUR responsibility to make sure that the system that you assemble is safe. Design considerations and for safety are your responsibility. For your consideration the following is included but should not be considered complete:

ELECTRONICS SAFETY GUIDELINES

1. **NO HORSEPLAY.**
2. **WORK ON DE-ENERGIZED CIRCUITS ONLY.** As far as possible, remove power before handling a circuit in any way.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156

Bethel Park, PA 15102

(800) 936-5280

3. **RINGS, WATCHES, AND NECKLACES** made of metal can be hazardous when they touch energized circuits. Use care and prevent accidents. Also, tie back medium length and long hair when working moving equipment.
4. **HIGH-VOLTAGE SAFETY.** This system is not intended for high voltages (Greater than 15 VAC power input and control of signals greater than 24 VDC.) If you must work with voltages higher than this, be sure you are qualified and prepared to take all responsibility for its use in this fashion.
5. **HIGH-CURRENT SAFETY.** Even at low voltage, circuits can generate appreciable heat, and cause burns. Know what you're doing. If you don't know approximately how much current will flow in a circuit, don't apply power until a more experienced person has checked it.
6. **LEAD HAZARD.** Many electronic components are pre-coated with solder that contains lead. **DO NOT EAT, DRINK, OR SMOKE** after handling electronic components until you have washed your hands. *This is doubly important when you have been soldering.* Make it a practice to wash your hands before and after working with electronics materials.
7. **NO FOOD OR DRINKS.** Don't eat or drink while working on electronics projects.
8. **BASIC EYE PROTECTION** (eyeglasses or goggles) is required for **SOLDERING**, use of **WIRE CUTTERS** (which can result in flying fragments), and all use of **POWER TOOLS** other than the power screwdriver.
9. **GOGGLES** (over your glasses if you wear glasses) are required to protect your eyes whenever you are setting up electronics. If there is a wiring error, parts can explode with high velocity.
10. **PROPER HANDLING OF COMPONENTS:** Integrated circuits are easily damaged by static electricity. To protect them from damage, avoid touching the pins any more than necessary; touch the frame of a piece of equipment before touching anything inside it; and in dry weather, make sure anti-static spray is applied to the carpet and chairs at least once a month. Also, turn off power before inserting components into a circuit.
11. **SAFE USE OF TOOLS:** Use pliers, wrenches, and screwdrivers that fit. It's much easier to get hurt, and to damage the equipment you're working on, when you use the wrong size. The right size Phillips (+) screwdriver is often surprisingly large; always use the largest one that fits.
12. **NO EXPOSED WIRES** When you complete you project and before powering the system be sure there are no exposed wires that can electrocute anyone or otherwise pose a safety hazard.



PROCERIUS, INC.

SYSTEMS & SOFTWARE

PO Box 156

Bethel Park, PA 15102

(800) 936-5280

13. **KEEP IT CLEAN:** A clean work environment can help you avoid many unnecessary accidents. Keep your work area clean and well organized. Pick up any trash that finds its way to the floor.

14. **THESE ARE JUST GUIDELINES:** Be sure you understand all aspects of your connections.

FIRE EXTINGUISHER: keep one handy.

IDEAS AND QUESTIONS TO ASK YOURSELF:

There is a safe way to do all jobs; ask yourself these questions:

What are the worst possible things that could go wrong?

How will I deal with them?

Am I sure there is no exposed wiring?

Am I sure the system is on a GFI?

Can water get into the system and short it out?

Am I sure the system will be safe no matter what programming I do in the computer?

DO NOT power the system until you have answers to these questions.

Wiring – System Design

Always use ground fault indication (GFI) protection on your system and make sure you have proper system grounding. There are numerous references on wiring safety and we recommend that you review them in detail until you are absolutely certain that you have safe wiring, before turning on the power. A complete guide to wiring is beyond the scope of this product, if you are not certain of what you are doing please discontinue use of the device / system.

The outputs automatically turn off when the PC is disconnected or communication stops. This is a safety feature that ensures that if the computer stops working the outputs will not continue to drive the circuits. The default, safe, and de-energized state is the state when zeros are written to the outputs or when the card is first powered up.

Water

The IOGeek system is not compatible with water. Keep the IOGeek and any wiring that you do dry.