JonnyMac's Prop-1 Programming Attack!

By Jon McPhalen

Published by Irish Ayes Productions
Burbank, CA

Editor:

ISBN: xxx-x-xxx-xxxxx-x (paperback)

McPhalen, Jon, 1962-

Dedicated to my friend, Cliff Osmond. I'm a better actor and writer for you having been my life; more important, though, is that I'm a better man. I miss you.

Origins

It all started for me back in the Spring of 1994. Ads for the new BASIC Stamp 1 (BS1) microcontroller module had been running in *Nuts & Volts* magazine for a few months. I mostly ignored them, though, thinking they were probably more hype than truth; I didn't believe that anyone could squeeze so many features into such a small microcontroller.

Finally, I found myself facing a forced "vacation" to recover after a surgery. I knew I was going to be off work for a couple weeks, and didn't relish the thought of spending that time in bed or on my couch. I called Parallax and ordered a BASIC Stamp 1 starter kit. It arrived a few days later.

That night, after dinner, I cracked open the box and loaded the BASIC Stamp programming editor into my IBM PS/2 Model 30 (*with a whopping 20-megabyte hard drive!*). As an avid electronics hobbyist for most of my life, my bench is well stocked with resistors, capacitors, LEDs, etc. – I had parts enough to have a lot of fun. And I did. So much so that the next time I checked my watch it was nearly 10AM – the fun I had with the BS1 kept me going through night and I didn't notice! As I was already late for work I decided to call in sick because it doesn't look good to call in tired. The surgery took place a few weeks later and my recover was spent in front of my computer, not on my couch.

I have programmed the BASIC Stamp 1 microcontroller in various forms nearly every day since. Along the way I've developed a lot of tricks and I intend to share them.

Back to the time-line... about 10 years after getting started with the BASIC Stamp 1, I was spending part of my Christmas vacation surfing the Internet for information on prop building. During the search I came across a frighteningly-named, yet family-friendly event: Death Fest. This was a weekend of seminars and demonstrations by and for Halloween enthusiasts. One of the presenters was giving a session on the BASIC Stamp 1 and happened to mention me (by my former name, Jon Williams) in his materials. By then I had been a long-time writer for *Nuts & Volts* and was well known for my BASIC Stamp programming skills.

I contacted Joe and asked him if there was anything I could do. He suggested that I look over his materials for glaring errors. In 2004 I was an Applications Engineer for Parallax (the company that created the BASIC Stamp), working from my home in Dallas, TX. I scheduled one of my trips to the home office in California to coincide with Death Fest so I could attend.

My boss, Ken, told a colleague, John Barrowman, about the event and John – also a Halloween enthusiast – asked me if he could go, too. Of course! – the more the merrier. We loaded up the car with Parallax products to demonstrate and give away, then headed south to the event.

The Birth of the Prop-1 and EFX-TEK

To say that we had a great time would be a serious understatement. John and I didn't know each other very well before that trip. What we learned is that we're only a couple months apart in age (he's older), both served the United States military (John was in the Navy, I was in the Air Force), we both love to play guitar and sing (John plays better than I), and that we enjoy acting (John enjoys occasional community theater while I'm a member of the Screen Actors Guild and have been in plays, commercials, on television, and in feature films).

Death Fest was a blast. While chatting and working with other attendees we noticed that many were cobbling together custom controllers from a simple BASIC Stamp 1 board and a ULN2803A driver chip. It was a good idea that made sense, but not for individuals to be hand-soldering. After all, their desire was to build and animate props and decorations, not to build electronic circuits.

On the drive back to Rocklin, John and I talked over ideas and agreed on a circuit. By the end of the next day, John, a master PCB layout engineer, had moved us from schematic to the Rev. A Prop-1 controller. It is named the Prop-1 because its primary audience is prop control and effects, and it was designed around the BASIC Stamp 1 microcontroller.

John sent files to a PCB fab house, ordered parts, and on the Friday following our first visit to Death Fest we had working Prop-1 controllers to send out to our new friends.

Finding a bit of a market niche for entertainment-oriented electronics, we formed a little product group inside Parallax called Parallax EFX. A year later we were given the opportunity to take the new products we created and start our own company to focus on that customer base; that company is EFX-TEK.

The Prop-1 continues to be the flagship product for EFX-TEK. It gets used in industry, museums, amusement parks of all sizes, in film and television props, and, yes, in lots of props and displays created by Halloween enthusiasts and professionals. It's inexpensive, it's reliable, and with a little effort it can be programmed to do most of the things one needs for prop control.

If I do my job well, by the end of this book you'll be programming the Prop-1 to do the things you want and need it to do. Yes, it will take a bit of work on your part. I promise that your efforts will be rewarded. Ready? All right, then, let's have some fun!

Getting to Know the Prop-1 Controller

New users will often ask, "*So what is a Prop-1 controller and how is it different from Brand X's Model Y?*"

In most of the markets it serves, the Prop-1 does stand out because it is one of the few products that is truly programmable. When I use the term *programmable*, I mean via standard programming practices, not pressing one button and then banging a sequence of output states into the device with another button. These *bit-banger* style products (like the EFX-TEK EZ-8) are not controllers in my mind, they are programmable *sequencers*.

What's the difference? A programmable controller like the Prop-1 has the ability to make decisions based on internal and external conditions, decisions which can change its behavior while operating; a sequencer cannot – it does the same thing every time it runs. In a word, *sequencers are boring* (okay, that was three words). Yes, they have their place in simple props, but they are still boring. And here's the rub: a controller can be programmed to behave like a sequencer; a sequencer cannot be changed to behave like a controller.

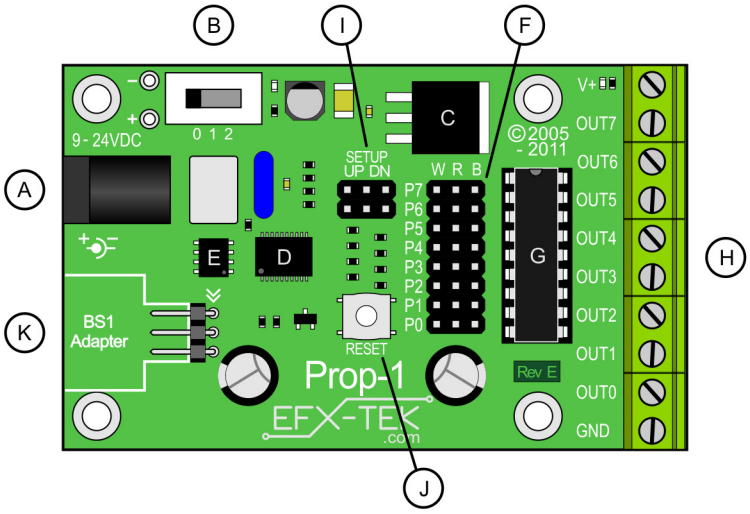
Fair warning: I live and work in Hollywood. We are a diverse and highly opinionated group of artists, technicians, and engineers. Through the course of this text I will at times express opinions that you may not agree with. If I've offended anyone who loves bit-banging sequencers, now is your opportunity to get past it. I designed the circuit and firmware for the EFX-TEK EZ-8. It's the best bit-banger available, but I still think it's less exciting than a Prop-1 for what I do. Programmable controllers rule!

The Prop-1 is programmed in a variant of the popular BASIC language. Here's the good news: BASIC is an acronym for **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. Note the use of the word, 'Beginner.' BASIC was specifically designed to be easy to learn and use. If you're 30ish or older, you may have learned BASIC in school. If you didn't then, yes, for you it means learning a new language to use the Prop-1. That said, there is more good news: the BASIC language used by the Prop-1 (called PBASIC 1.0) has a dictionary of 31 words. That's it; 31. Human conversational speech is typically composed of 2000 or more words. So, you see, this is not like trying to learn French for a summer vacation in Paris; it is far easier. You just have to fetch your own croissants.

You will need a computer to program the Prop-1, and that computer must be capable of running some flavor of the Windows operating system (XP or higher). The software for programming the Prop-1 is free, but only available for Windows. If you're using a Mac or running some variant of Linux, the programming editor – to my knowledge – runs fine in an emulator (a friend works for a huge SoCal amusement park and does his Prop-1 programming on a Mac).

Programming the Prop-1 is what the bulk of this book will focus on. Still, you're dealing with an embedded device and you control the connections. With that, let me introduce the Prop-1 controller and its major components.

Prop-1 Tour



This image is of the Prop-1, Revision E PCB. It has a few updates over earlier revisions, but the major features of the Prop-1 have not changed since John's first PCB design created the day after Death Fest in 2005.

A) Power Jack

This is the power input jack; it is called a 2.1mm *barrel connector* (the pin diameter is 2.1mm, the barrel diameter is 5.5mm). The input voltage to the Prop-1 will usually be 12 or 24 volts DC, depending on the devices connected to it. The Prop-1 will work down to 9 volts for simple LED projects (e.g., with a 9-volt battery), but you shouldn't use anything lower, and do not apply a voltage above 24 volts. The critical requirements are that *the input voltage is DC*, and that the plug is *center-positive*. If you're using a power supply that did not come from EFX-TEK, please check its specifications very carefully.

On later Prop-1 boards you'll see a silver rectangle just to the right of the power jack. This is a self-resetting fuse that is designed to protect the Prop-1 from bad power input, or too much power demanded from the terminal blocks. Please understand that no fuse is perfect and there is no guarantee that this will protect your Prop-1 from damage.

Again, check your power supply. DC only (*AC will do damage!*), 12 or 24 volts.

A final note: a power supply is analogous to a hose. A hose provides a stream of water, a power supply provides a stream of electricity. A garden hose is still a hose, but does not have the same capacity as a fire hose. Likewise, you will come across power supplies that do not have the capacity required by your circuit. This capacity is measured in *Amperes* (Amps). The Prop-1 Starter Kit comes with a 1A power supply; this is adequate for most props and displays.

Where I have seen beginners get caught out is when they buy a small – *usually very cheap* – power supply that does not have the capacity to handle current consumed by the external devices the want to operate. The problem tends to be hidden during the programming phase as the Prop-1 requires very little current. The ugly surprise rears its head when the circuit goes live and the Prop-1 attempts to power a relay, valve, or servo. This scenario can become incredibly frustrating.

Something often seen with an inadequate power supply is that the program starts to run, gets to some section, and then restarts for no apparent reason. There is a reason: the output demand exceeded the capacity of the power supply and it went into a *brown-out* condition; this is where the power supply will reduce its output voltage to keep up with an excessive current demand. When the voltage into to the Prop-1 falls to an unsafe level the brown-out detection circuit will cause it to reset.

Checklist for power-supply:

- ✔ 12 (preferred) or 24 volts DC?
- ✔ 2.1mm, center-positive barrel connector?
- ✔ 1A or greater (depending on your circuit)?

B) Power Switch

The power switch on the Prop-1 has three positions 0) off, 1) logic on, and 2) logic and outputs on. Position 0 is obvious: no power to anything. Position 1 connects the incoming power to the on-board 5v regulator (C) which provides 5 volts to the circuit. In this position the BS1 processor (D), EEPROM (E) and the TTL IO header (F) are active. In position 2 power is applied to the regulator (just as in position 1) and is also routed to the V+ terminal for powering external devices like valves and relays.

In position 1 a small LED near the power switch will light. In position 2, the power switch LED and second LED, adjacent to the V+ terminal, will light.

C) 5-Volt Regulator

The Prop-1 processor and circuitry operates at TTL (transistor transistor logic) levels which means it is designed to run at 5 volts DC. If you applied 12 or 24 volts directly to this circuity you would probably hear a loud *crack!* – and then see and smell a puff of acrid blue smoke. Yes, when a component fails badly it does let out blue smoke; it smells horrible, too.

The large component with three leads is a voltage regulator. This device takes the incoming voltage and regulates it down to 5 volts for the circuit. What happens with the rest of the voltage? It gets converted to heat. You will notice that this device becomes a bit warm when running, but it should never get so hot that you can't keep a finger on it – if that happens you may have a big problem brewing. That said, this regulator is very stout and has the ability to protect itself from over-voltage and over-temperature situations.

The output of the regulator runs the Prop-1 circuitry and is provided to the R pins to the TTL IO header (F). This allows the connection of 5v devices like PIRs (input sensor) and servos (output device). Early versions of the Prop-1 could supply about 1A at 5v; newer (Rev E and later) are capable of supplying up to 3A.

> **Tip:** If you're using 24v power supply the only LED that lights is for V+ when the power switch is in position 2, you have a bad or unregulated supply. What you're witnessing is the regulator protecting itself from over-voltage by shutting down – this is why the power LED (which uses 5 volts) does not light. The V+ LED uses the incoming power and will light even when the regulator is shut down. The regulator will shut down when the input goes above 29 volts; this can happen with unregulated power supplies, especially with the small load imposed by the Prop-1 circuitry.

D) BASIC Stamp 1 (BS1) Processor

This chip is the 'brain' of the Prop-1. It is a Microchip PIC16C56A microcontroller which has been specially programmed with the Parallax PBASIC 1.0 interpreter. That is to say that these chips are only available from Parallax – a blank PIC16C56A will not run your Prop-1 programs.

E) EEPROM

The EEPROM (Electrically Erasable Programmable Read-Only Memory) is where your program is stored when downloading. It's important to understand a couple things: 1) Your program will stay in the EEPROM until you replace it with another program; it will not be lost, no matter how long power has been removed. This type of memory is called

non-volatile. 2) What gets downloaded to the EEPROM is not the PBASIC source code you will have typed into the programming editor. The EEPROM is very small – only 256 bytes (¼K!). What is stored in the EEPROM is a compiled and compressed version of the program called *tokens*.

Important Note: The BASIC Stamp programming editor can compile and compress your PBASIC program into tokens for downloading to the Prop-1. What it cannot do is convert compiled and compressed tokens back into PBASIC source code. There is an old programmer's saying: *Save early, save often*. And make back-ups of important programs. If you lose the listing for a program in a Prop-1 you must recreate it from memory or reverse-engineer the program based on the Prop-1's behavior.

F) TTL IO Header

This 3x8 male header provides direct connections to the BS1 processor IO (input/output) pins (W), as well as 5v regulated power (R), and system ground (B). The processor is protected from short circuits with 100-ohm resistors, as these pins are capable of being inputs (e.g., a PIR sensor) or outputs (e.g., controlling an LED or servo). Pin behavior – input or output – is determined by the program. This is another aspect that separates true controllers from sequencers: you can define inputs and outputs as needed by the device you're writing the program for.

Note: The WRB marking above the TTL IO header corresponds with the color code (White-Red-Black) of many hobby servo connections, as well as the female-to-female extension wires sold by EFX-TEK (#805-00035 and #805-00175)

G) ULN2803A Low-Side Driver

While the BS1 processor is quite capable, it cannot provide enough electrical energy to power devices like valves and relays – connecting a device like this directly to the processor (via the TTL IO header) would damage the chip, so don't do it! Inline between the processor and the output terminal block (H) is the ULN2803A, a low-side driver.

'Low-side' means that it switches the ground side of the circuit. This can be confusing, at first, to those with a little electrical experience that are expecting the output terminals, OUT7-OUT0, to read 12v or 24v when active. They don't because they connect to ground when active. When not active the outputs will 'float' because the ULN2803A is an *open-collector* device (we'll cover open-collector in more detail later). A floating output acts like a broken wire: it's not connected to anything.

H) Output Terminal Block

These terminals provide outputs for V+ (power to the external devices), OUT7-OUT0 (outputs for the processor, depending on the program), and the power supply ground. Note that these are outputs only.

When connecting external devices like valves and relays, the V+ terminal is common to all; this is why it is switched (killing the V+ terminal output by moving the power switch back to position 1 allows you to test a program without activating the attached devices). The other side of the device will connect to the corresponding OUTx terminal. Assuming the power switch is in position 2 and the BS1 processor commands the output to be active, current will flow from the V+ terminal, through the device, through the ULN2803A to ground.

> For those that want to check an output with a multimeter when troubleshooting, you must remember to put the red lead (+) on the V+ terminal, and the black lead (-) on the OUTx terminal you're checking. If the program has made the IO pin an output and turned it on, then your meter should read about 12 or 24 volts, depending on your power supply.

I) Setup Jumpers for P6 and P7

A floating output is not a problem, but a floating IO pin, when used as an input, can create unexpected results – false positives. If you examine the Prop-1 circuit you'll find that the ULN2803A acts like a pull-down on all the IO pins, keeping all inputs in a low (0) state until a 5v signal is applied to the corresponding pin on the TTL header.

Some sensors, though, provide an open-collector output which means it switches from floating to ground when active. The Prop-1 can use these kinds of sensors, but requires some intervention; specifically, the input pin used must be held in a high (1) state when the sensor is not active.

You can do this on P6 or P7 with the following steps:

- ✔ Move the SETUP jumper to the UP position
- ✔ Clip the corresponding input pin on the ULN2803A
  – this removes the pull-down influence of the ULN's input circuitry

Don't worry about the gnarly details right now – I will go into lots of detail with examples later. The point is that the Prop-1 can have inputs P6 and P7 configured to accommodate open-collector devices, and this allows the use of many industrial sensors.

J) Program Reset Button

On power-up or a brown-out condition, circuitry in the Prop-1 will cause the processor and program to be reset. This clears all the temporary memory (RAM), sets the IO pins to inputs, and then starts the program from the beginning. If you want to re-start a program during testing there is no need to cycle power or download again; simply press the reset button and the program will start over.

K) Programming Connection

The Prop-1 is programmed through a TTL serial connection. If you're using an older computer with a 9-pin serial cable, the BS1 Serial Adapter (#27111) will convert the RS-232 levels from the computer to TTL levels which are required by the BS1 processor. Standard RS-232 serial ports are nearly non-existent on newer computers, which means you would use a USB-to-Serial Adapter (#28030) inline between computer USB port and the BS1 Serial Adapter

A new adapter from EFX-TEK, the Prop-1 USB Adapter (#28101), replaces the USB-to-Serial Adapter and the BS1 Serial Adapter with a single, easy-to-use module. In addition to allowing for USB connectivity, the Prop-1 USB Adapter has LEDs that indicate information flow between the computer and the Prop-1.