

# A Beginner's Guide to Haunting with Stamp Microcontrollers

Joe Stone

[joe@wickedstone.com](mailto:joe@wickedstone.com)

<http://www.wickedstone.com>

With Special Guests

**Parallax** Application Engineers

John Williams

John Barrowman

<http://www.parallax.com>

# Haunt Controllers

Single-Input, Single-Output, Single-Step

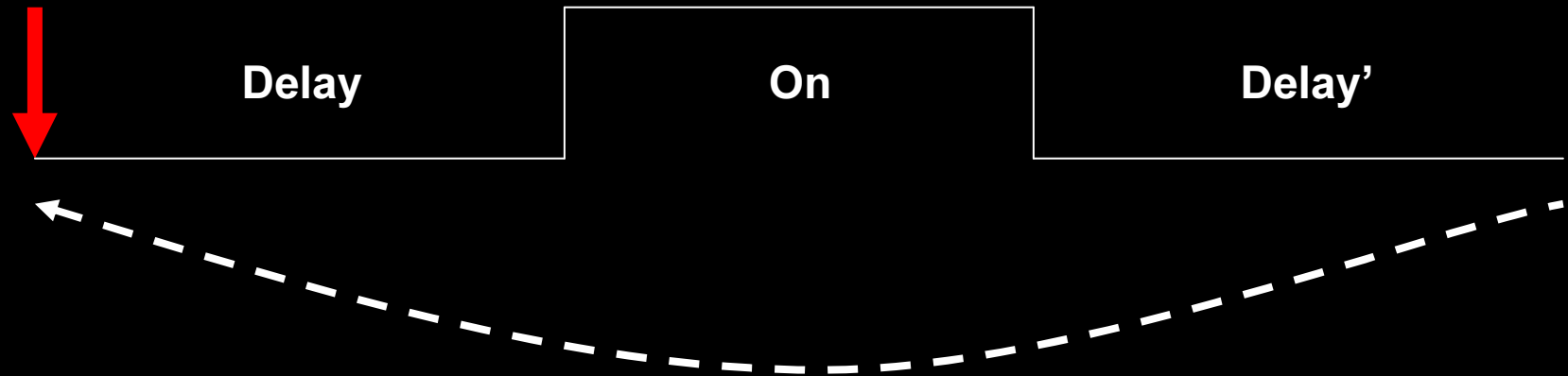
Single-Input, Single-Output, Multi-Step

Single-Input, Multi-Output, Multi-Step

Multi-Input, Multi-Output, Multi-Step

# Haunt Controllers (cont)

Single-Input, Single-Output, Single-Step



e.g., Pop-Up

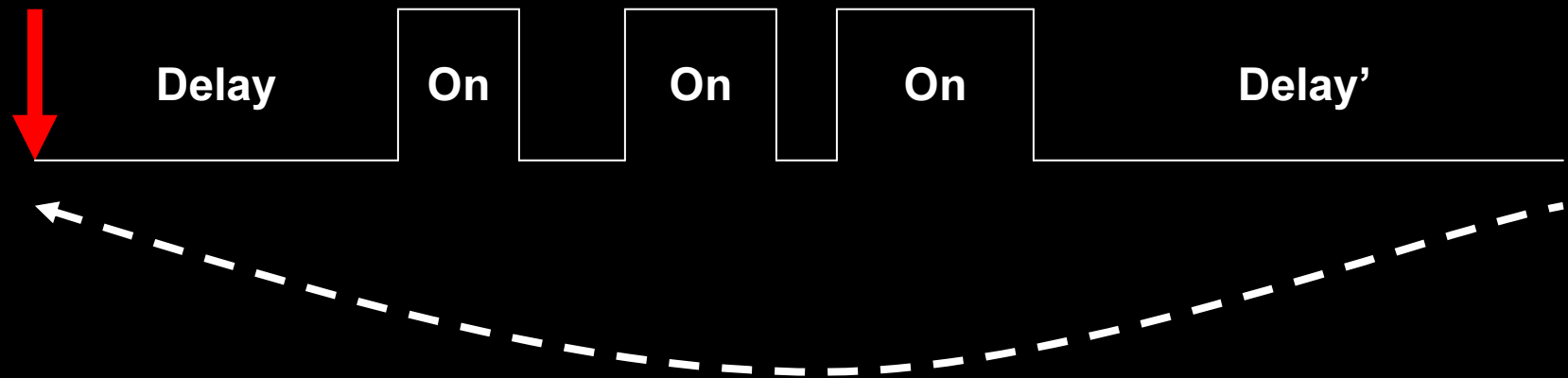
# Haunt Controllers (cont)

**Terror by Design Universal Dual Timer II**  
**UDT II \$52.95, UDT II IR \$77.95**



# Haunt Controllers (cont)

Single-Input, Single-Output, Multi-Step

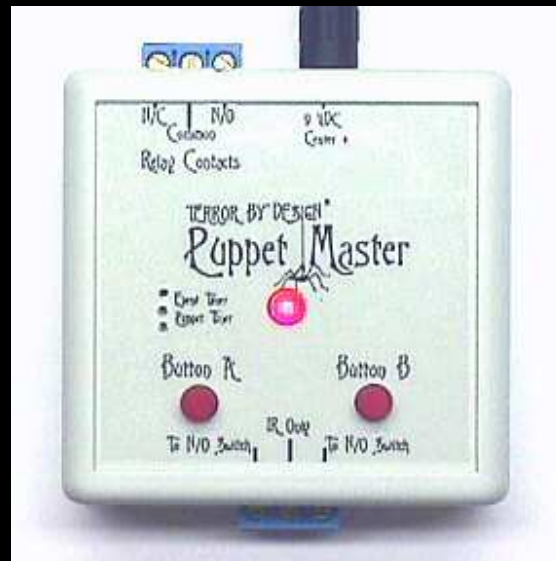


e.g., Electric Chair

# Haunt Controllers (cont)

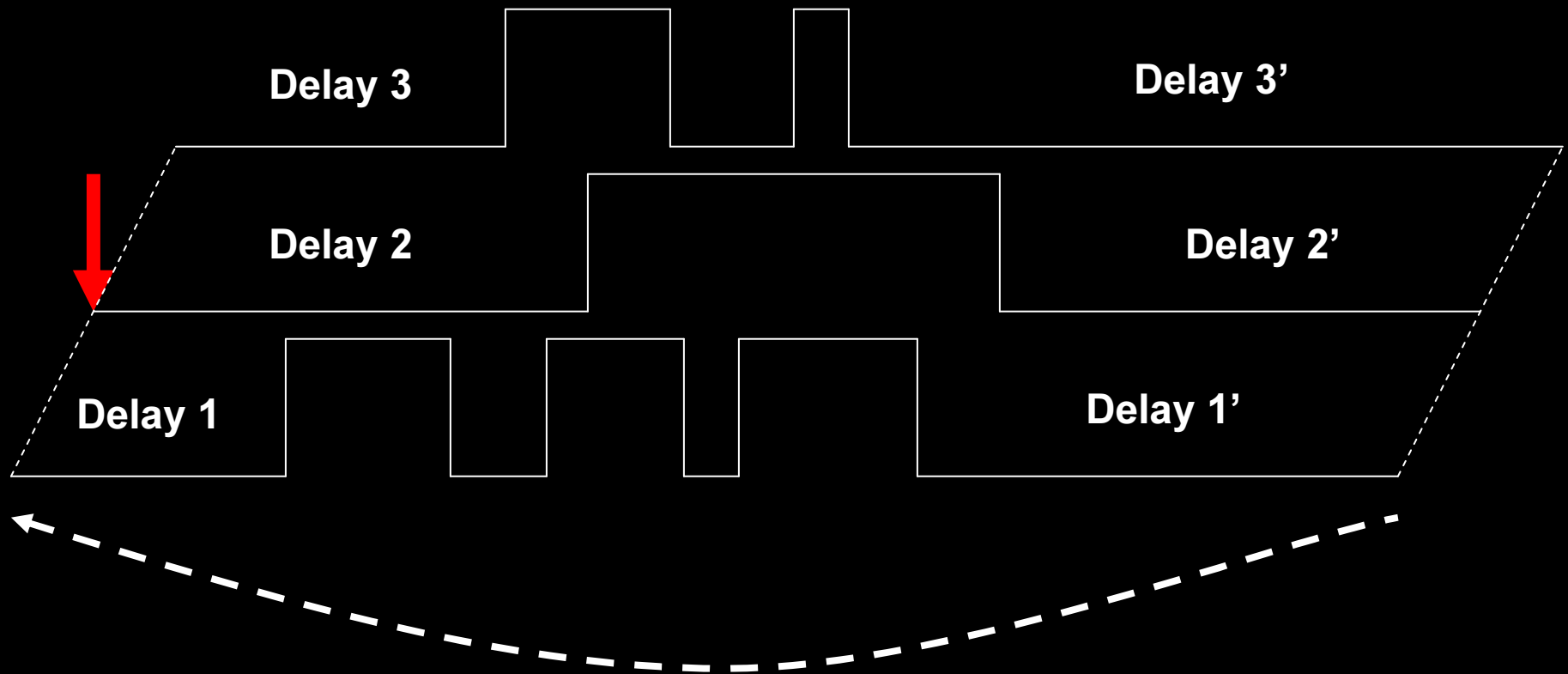
## Terror by Design Puppet Master

Puppet Master \$49.95, Puppet Master IR \$74.95



# Haunt Controllers (cont)

Single-Input, Multi-Output, Multi-Step

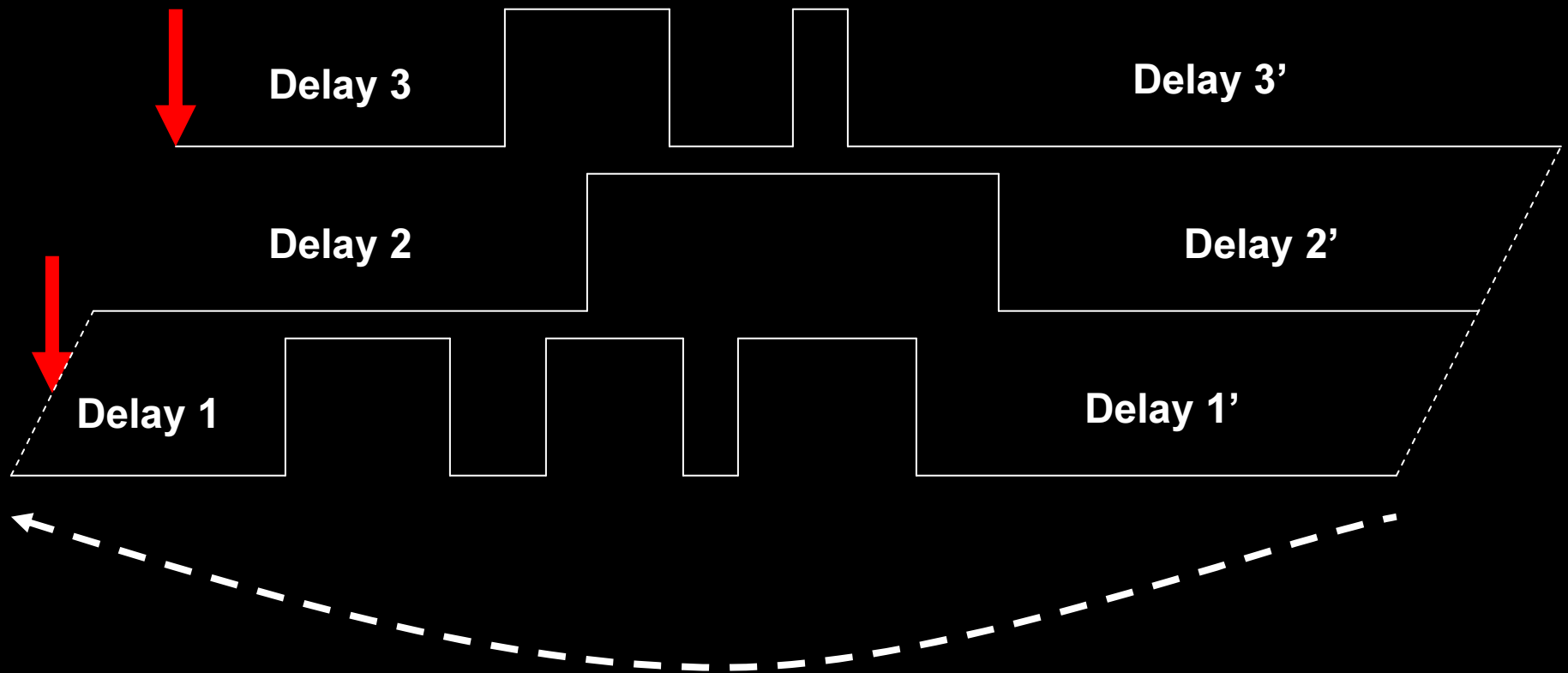


e.g., Electric Chair, Fog, Sound, ...



# Haunt Controllers (cont)

Multi-Input, Multi-Output, Multi-Step



# Haunt Controllers (cont)

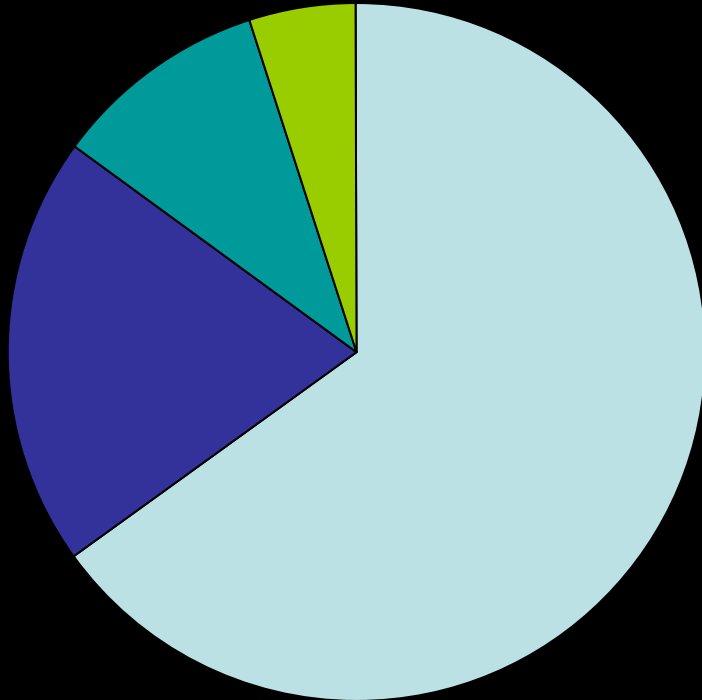
## **Gilderfluke MiniBrick8**

**BR-MiniBRK8, \$150.00**

**PC•MACs Free, MACs (Real-Time) License \$550**



# Haunt Controllers (cont)



- Single-Input, Single-Output, Single-Step
- Single-Input, Single-Output, Multi-Step
- Single-Input, Multi-Output, Multi-Step
- Multi-Input, Multi-Output, Multi-Step

## What is a Stamp Microcontroller?

**A BASIC Stamp microcontroller is a single-board computer that runs the Parallax PBASIC language interpreter in its microcontroller.**

## What is a Stamp Microcontroller? (cont)

**The developer's code is stored in an EEPROM, which can also be used for data storage.**

## What is a Stamp Microcontroller? (cont)

**The PBASIC language has easy-to-use commands for basic I/O, like turning devices on or off, interfacing with sensors, etc.**

# What is a Stamp Microcontroller? (cont)

**Parallax  
Beginner's  
All-Purpose  
Symbolic  
Instruction  
Code**

# What is a Stamp Microcontroller? (cont)

**P**arallax

**B**eginner's

**A**ll-Purpose

**S**ymbolic

**I**nstruction

**C**ode



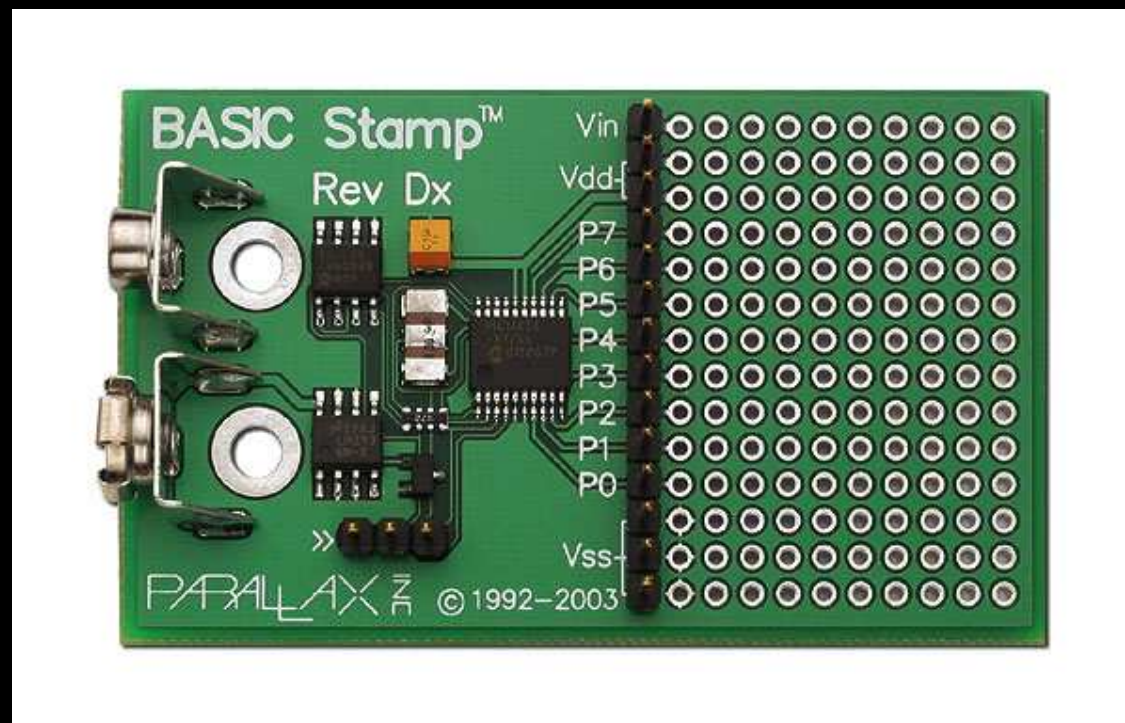
# BASIC Stamp 1 vs BASIC Stamp 2

	<u>BS1</u>	<u>BS2</u>
Speed (MHz)	4 MHz	20 Mhz
Speed (instructions per second)	2,000	4,000
Inputs/Outputs	8	16
RAM Variables (bytes)	16	32
EEPROM (bytes)	256	2K
Program Length (lines of code)	80 – 100	500
PC Port (Programming)	Parallel	Serial

# BASIC Stamp 1

## Parallax BASIC Stamp 1 Rev. Dx

Parallax Stock#: 27100, \$34.00



# BASIC Stamp 1 (cont)

## Lynxmotion First Step BASIC Stamp 1 Microcontroller

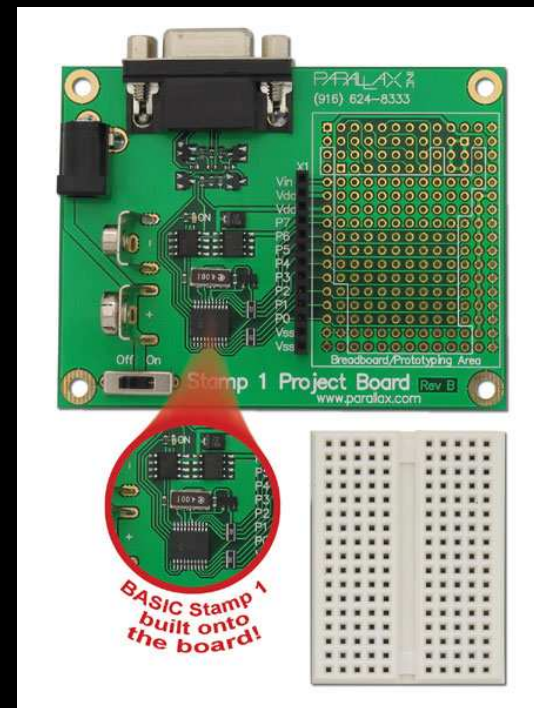
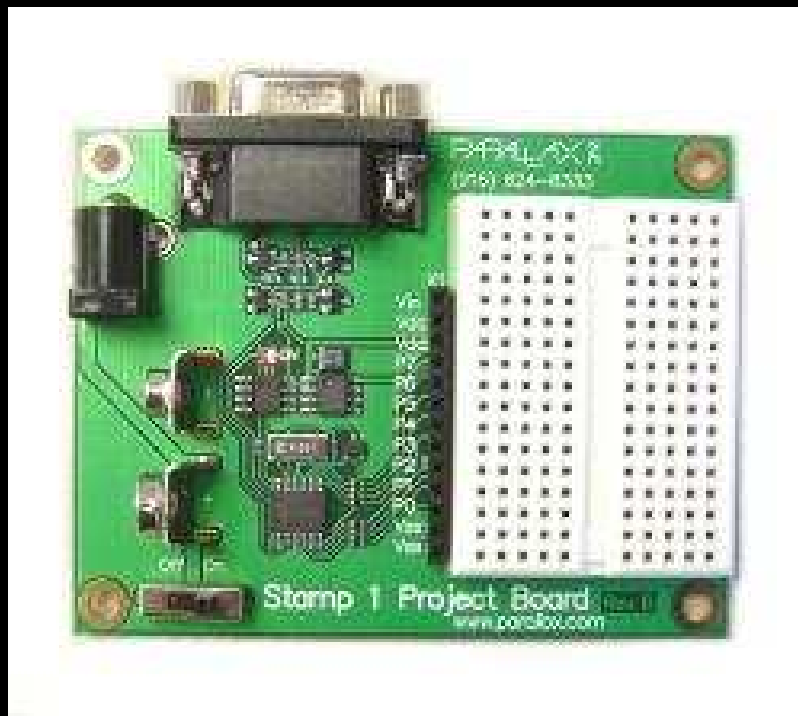


Discontinued

# BASIC Stamp 1 (cont)

## Parallax BASIC Stamp 1 Project Board

Parallax Stock#: 27112, \$29.00

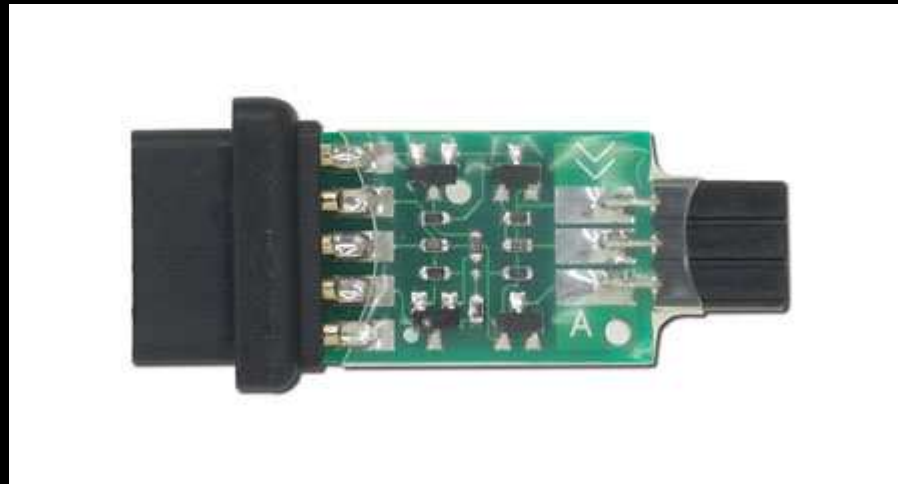


# Programming Interface

## Parallax BASIC Stamp 1 Serial Adapter

Parallax Stock#: 27111, \$4.95

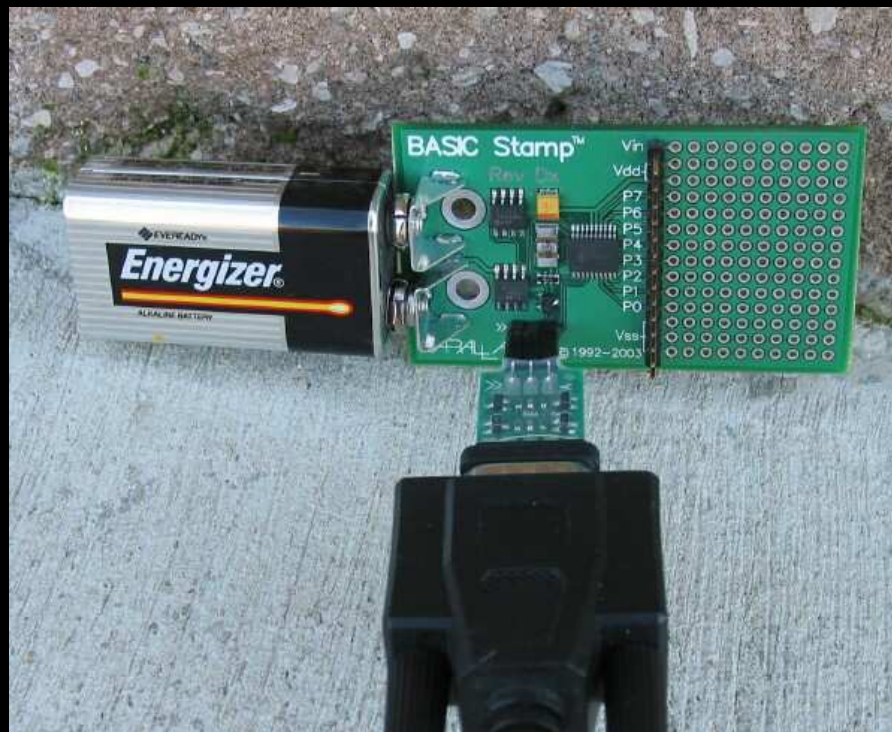
Allows “traditional” BS1 (Lynxmotion First Step, Parallax Rev. DX, ...) to be programmed via a serial port.



# Programming Interface (cont)

## Parallax BASIC Stamp 1 Serial Adapter

Parallax Stock#: 27111, \$4.95



<http://www.parallax.com>

# Stamp Editor (cont)

**Parallax BASIC Stamp 1 Editor for DOS**

Parallel Port Interface

Windows 95/98

**Obsolete**

# Stamp Editor (cont)

## **Parallax BASIC Stamp Windows Editor Version 2.1**

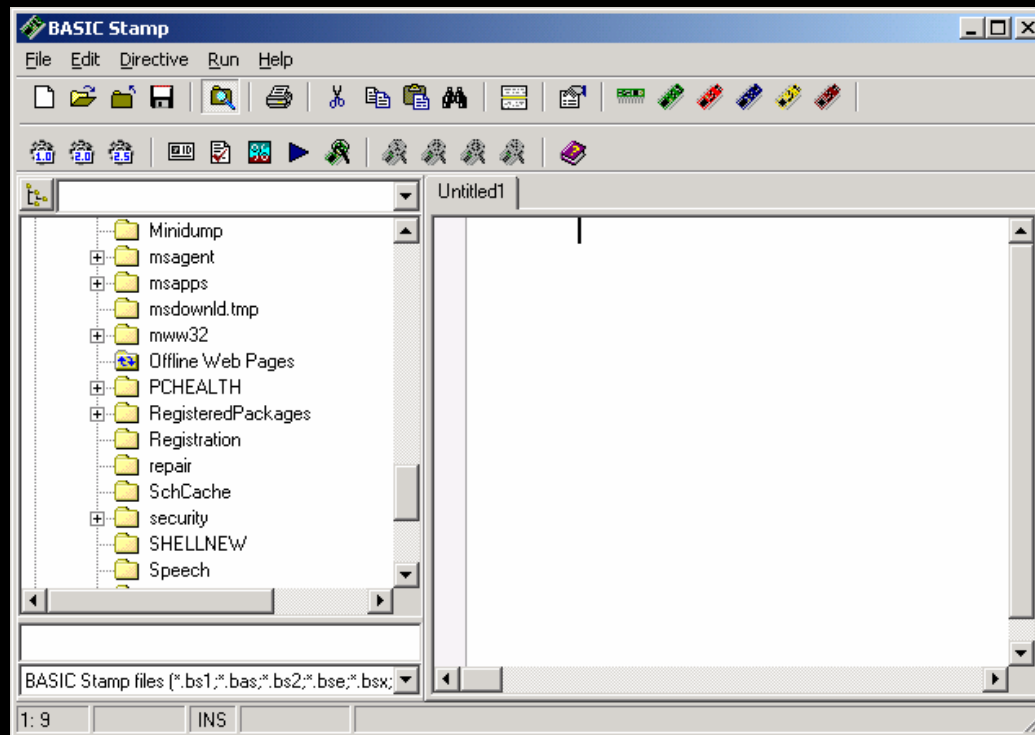
Serial Port Interface

Windows 98/ME/NT4.0/2K/XP



# Stamp Editor (cont)

## Parallax BASIC Stamp Windows Editor Version 2.1



# Stamp Programming using PBASIC

**Parallax BASIC Stamp Manual Version 2.1**

**It's free!**

**Defines hundreds of commands of which we  
will consider eleven (11).**

# BS1 Memory Organization

Word Name	Byte Name	Bit Name	Special Notes
PORT	PINS	PIN0 – PIN7	I/O pins; bit-addressable.
	DIRS	DIR0 – DIR7	I/O pins direction; bit-addressable.
W0	B0	BIT0 – BIT7	Bit-addressable.
	B1	BIT8 – BIT15	Bit-addressable.
W1	B2		
	B3		
W2	B4		
	B5		
W3	B6		
	B7		
W4	B8		
	B9		
W5	B10		
	B11		
W6	B12		
	B13		

# Stamp Programming using PBASIC

## SYMBOL Command

The SYMBOL command is used to “declare” a variable Name. A variable name may correspond to a RAM RegisterName,

**SYMBOL Name = RegisterName**

or to a ConstantValue,

**SYMBOL Name = ConstantValue**

# Stamp Programming using PBASIC

## IF..THEN Command

The IF..THEN command evaluates the Condition and if the Condition is true (non-zero), goes to the point in the program marked by Address,

**IF Condition THEN Address**

where,

Condition is a statement that can be evaluated as true (non-zero) or false (zero).

Address is a label that specifies where to go in the event that Condition is true.

# Stamp Programming using PBASIC

## FOR..NEXT Command

The FOR..NEXT command creates a repeating loop that executes the program lines between FOR and NEXT, incrementing or decrementing the Counter according to the StepValue until the value of the Counter variable passes the EndValue.

```
FOR Counter = StartValue TO EndValue {STEP {-} StepValue}  
NEXT {Counter}
```

where,

Counter is a variable that counts the number of iterations through the FOR..NEXT loop. StartValue and EndValue are variables or constants with a range of 0 - 65,535 that specifies the initial and final value of the Counter, respectively.

StepValue is an optional variable or constant with a range of 0 - 65,535 by which the Counter is incremented or decremented with each iteration through the FOR..NEXT loop. If the StartValue is greater than the EndValue, the StepValue is preceded with a minus sign (-).

# Stamp Programming using PBASIC

## OUTPUT Command

The OUTPUT command sets the specified pin to output mode,

**OUTPUT Pin**

where,

Pin is a variable or constant with a range of 0 - 7.

# Stamp Programming using PBASIC

## LOW Command

The LOW command sets the specified pin to output mode and sets the output of the specified pin low (zero volts),

### LOW Pin

where,

Pin is a variable or constant with a range of 0 - 7.



# Stamp Programming using PBASIC

## HIGH Command

The HIGH command sets the specified pin to output mode and sets the output of the specified pin high (five volts),

**HIGH Pin**

where,

Pin is a variable or constant with a range of 0 - 7.

# Stamp Programming using PBASIC

## INPUT Command

The INPUT command makes the specified pin an input pin,

**INPUT Pin**

where,

Pin is a variable or constant with a range of 0 - 7.

# Stamp Programming using PBASIC

## PAUSE Command

The PAUSE command halts the execution of the program for the specified Period,

**PAUSE Period**

where,

Period is a variable or constant and is specified in milliseconds with a range of 0 - 65,535.

# Stamp Programming using PBASIC

## GOTO Command

The GOTO command goes to the point in the program specified by Address,

**GOTO Address**

where,

Address is a label that specifies where to go.

# Stamp Programming using PBASIC

## PULSOUT Command

The PULSOUT command generates a pulse on Pin with a width of Period,

**PULSOUT Pin, Period**

where,

Pin is a variable or constant with a range of 0 - 7.

Period is a variable or constant and is specified in 10  $\mu$ sec units with a range of 0 - 65,535.

# Stamp Programming using PBASIC

## SEROUT Command

The SEROUT command transmits asynchronous serial data (e.g., RS-232 data),,

**SEROUT Pin, BaudMode, ( {#} OutputData)**

where,

**Pin** is a variable or constant with a range of 0 – 7 that specifies the output pin through which the serial data will be transmitted.

**BaudMode** is a variable or constant with a range of 0 – 15 that specifies serial timing and configuration (e.g., 4 = N2400 = 2400 Baud inverted, always driven).

**OutputData** is a list of variables, constants, expressions and formatters that tells SEROUT how to format outgoing data.

# Stamp Programming using PBASIC

## Example: Flashing LED

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```



# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: Flashing LED (cont)

```
SYMBOL led = 0
OUTPUT led

start:
HIGH led
PAUSE 250 ' Delay 1/4 sec
LOW led
PAUSE 250 ' Delay 1/4 sec
GOTO start
```

# Stamp Programming using PBASIC

## Example: LED Chaser

```
DIRS = %11111111
SYMBOL pin = B2

start:
FOR pin = 0 TO 7
    HIGH pin
    PAUSE 100 ' Delay 0.1 sec
    LOW pin
    PAUSE 100 ' Delay 0.1 sec
NEXT pin
GOTO start
```

# Stamp Programming using PBASIC

## Example: LED Chaser (cont)

```
DIRS = %11111111
SYMBOL pin = B2

start:
FOR pin = 0 TO 7
    HIGH pin
    PAUSE 100 ' Delay 0.1 sec
    LOW pin
    PAUSE 100 ' Delay 0.1 sec
NEXT pin
GOTO start
```



# Stamp Programming using PBASIC

## Example: LED Chaser (cont)

```
DIRS = %11111111
SYMBOL pin = B2

start:
FOR pin = 0 TO 7
    HIGH pin
    PAUSE 100 ' Delay 0.1 sec
    LOW pin
    PAUSE 100 ' Delay 0.1 sec
NEXT pin
GOTO start
```

# Stamp Programming using PBASIC

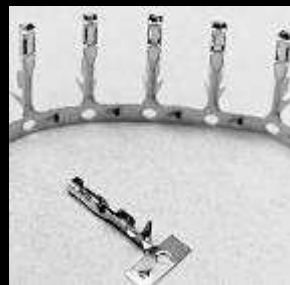
## Example: LED Chaser (cont)

```
DIRS = %11111111
SYMBOL pin = B2

start:
FOR pin = 0 TO 7
    HIGH pin
    PAUSE 100 ' Delay 0.1 sec
    LOW pin
    PAUSE 100 ' Delay 0.1 sec
NEXT pin
GOTO start
```

# Stamp Interfacing

## Connector Kit



# 50 VDC via Darlington Sink Driver

## Darlington Sink Driver

Digi-Key Part #: ULN2803AP-ND, \$0.72



# 50 VDC via Darlington Sink Driver (cont)

## **Darlington Sink Driver**

Digi-Key Part #: ULN2803AP-ND, \$0.72

**Eight Channels**

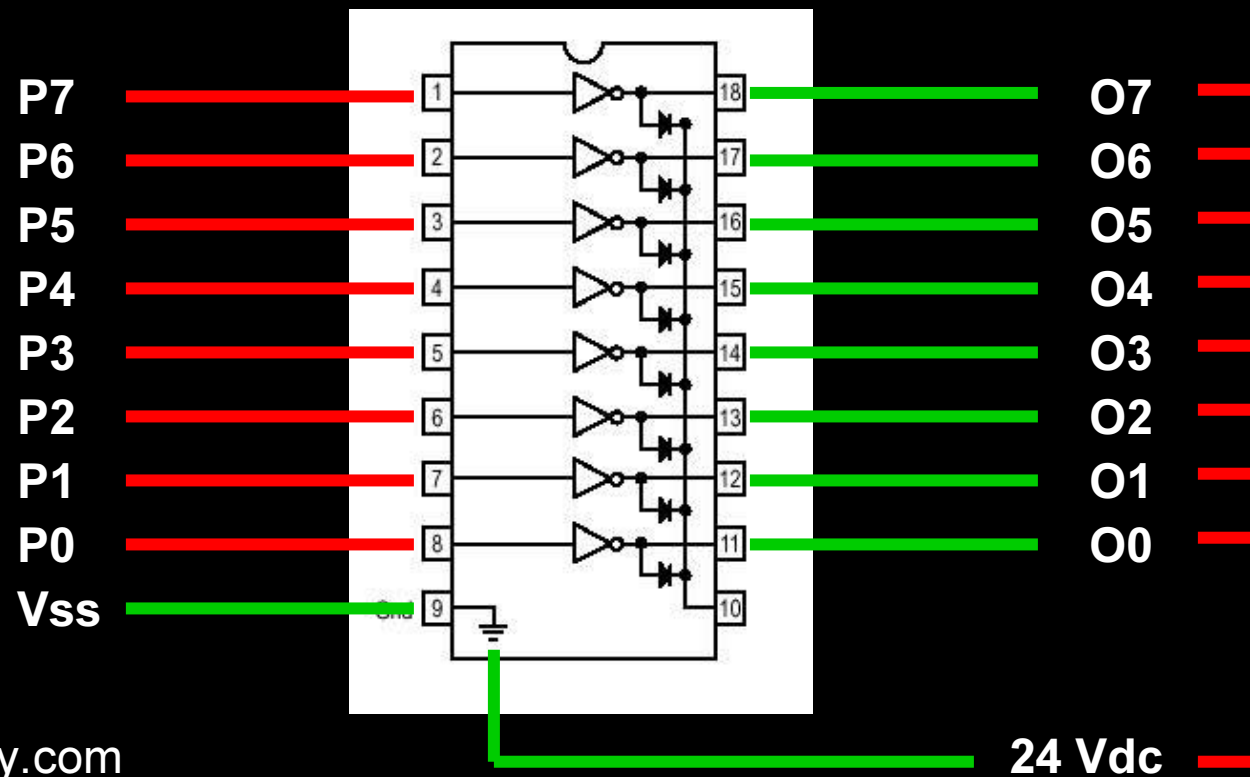
**Output Current: 500 mADC**

**Output Voltage: Up to 50 VDC**

# 50 VDC via Darlington Sink Driver (cont)

## Darlington Sink Driver

Digi-Key Part #: ULN2803AP-ND, \$0.72

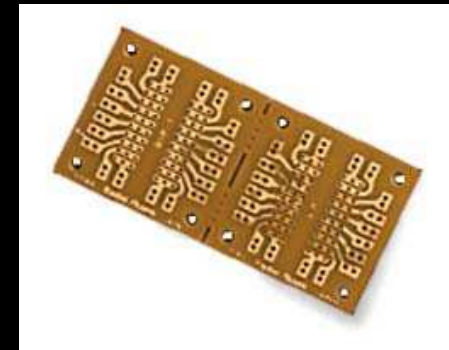


# 50 VDC via Darlington Sink Driver (cont)

## Darlington Sink Driver

### Dual General-Purpose IC PC Board

Radio Shack Catalog #: 276-159, \$2.29



**2 - 8-Pin Header Posts + 1 Pin**

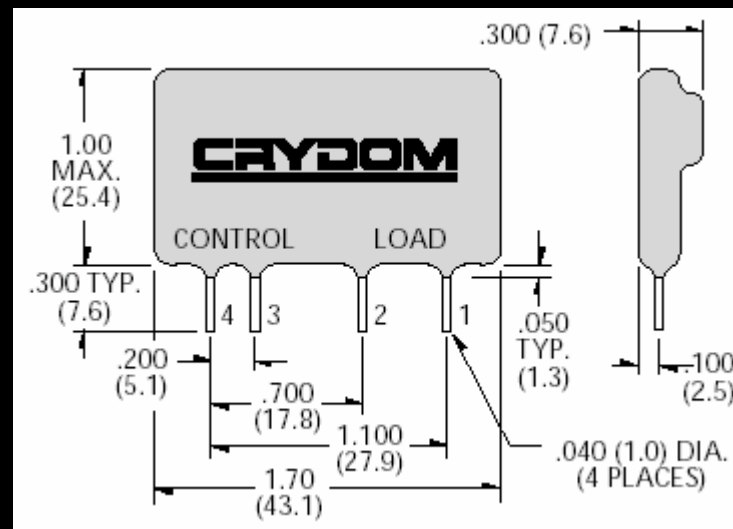
**20-Pin Retention Contact**

Radio Shack Catalog #: 276-1991, \$0.69

# 110 VAC via Solid-State Relay

## Crydom CX240D5 Solid-State Relay

Digi-Key Part #: CC1063-ND, \$11.78





# 110 VAC via Solid-State Relay (cont)

## **Crydom CX240D5 Solid-State Relay**

Digi-Key Part #: CC1063-ND, \$11.78

**One Channel**

**Input Voltage: 3 – 15 VDC**

**Input Current: 15 mA**

**Output Current: 5 A**

**Output Voltage: 12 – 280 VAC**

# 110 VAC via Mechanical Relay

## Chauvet SR-8 Relay Pack

Chauvet Part #: SR-8, \$37.00

(used in conjunction with Darlington sink driver and 12 Vdc)



# 110 VAC via Mechanical Relay (cont)

## Chauvet SR-8 Relay Pack

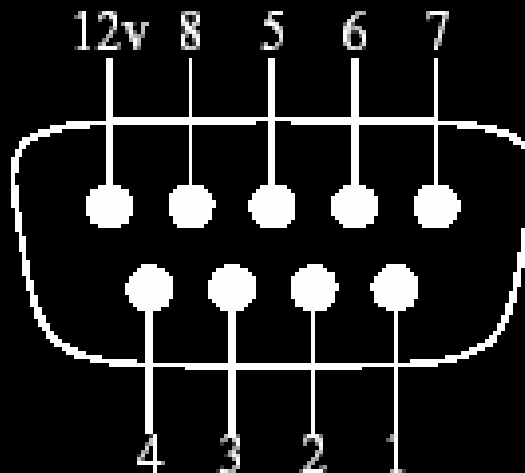
**12 Vdc - Yellow**

**8 - Orange**

**7 - Black**

**6 - Brown**

**5 - Red**



**1 - Green**

**2 - Blue**

**3 - Purple**

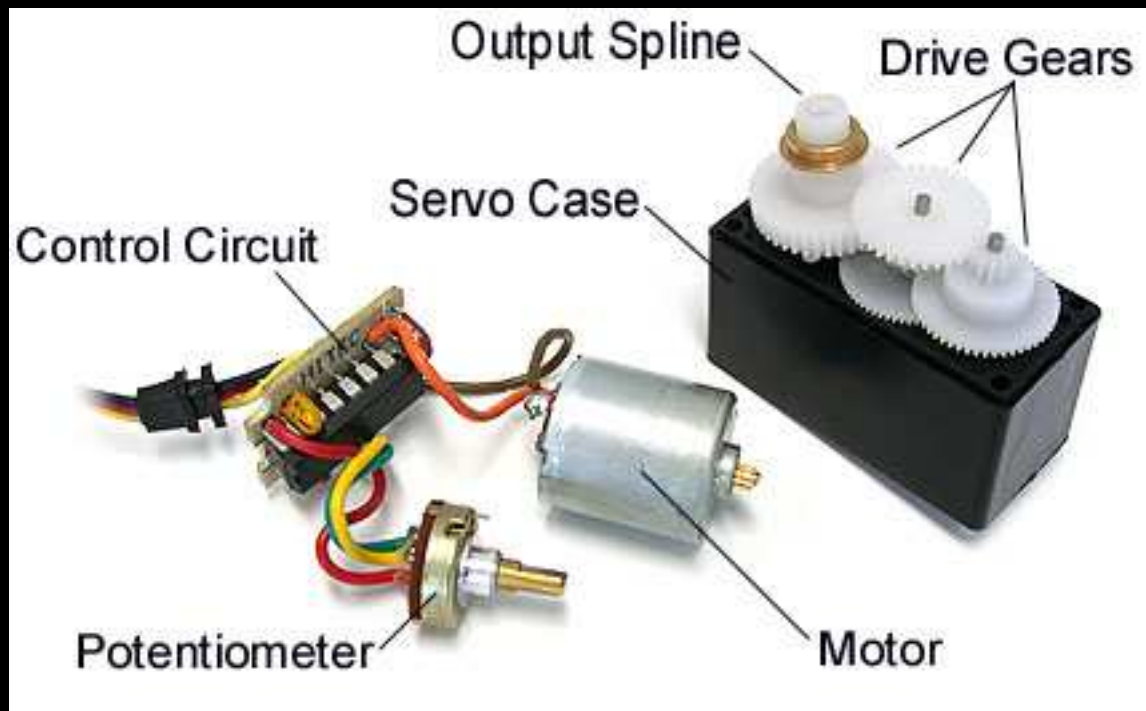
**4 - Gray**

# Servo Motors



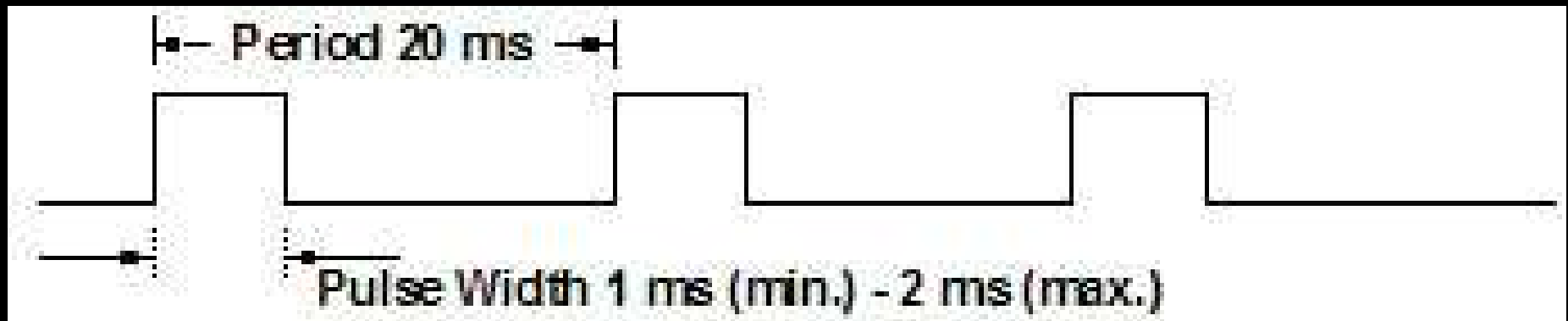
# Servo Motors (cont)

## Internals



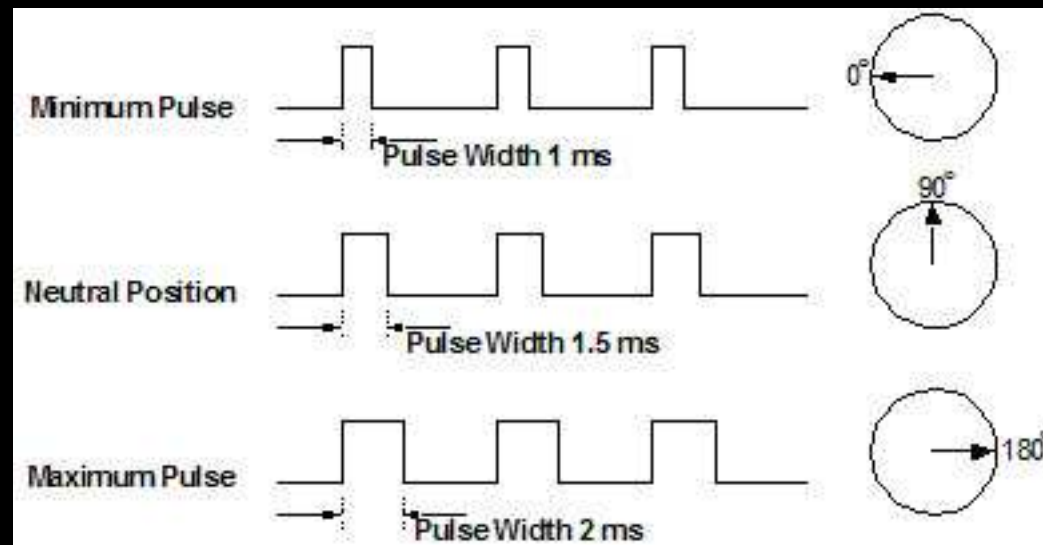
# Servo Motors (cont)

## Theory



# Servo Motors (cont)

## Theory



# Servo Motors (cont)

## Direct Servo Control

```
SYMBOL servo = 0
OUTPUT servo
SYMBOL position = B2          \ 100 x 10 µsec = 1000 µsec = 1 msec
                              \ 200 x 10 µsec = 2000 µsec = 2 msec

start:
FOR position = 100 TO 200 STEP 2  \ 1 to 2 msec
  PULSOUT servo, position
  PAUSE 10
NEXT position
FOR position = 200 TO 100 STEP -2 \ 2 to 1 msec
  PULSOUT servo, position
  PAUSE 10
NEXT position
GOTO start
```



# Servo Motors (cont)

## Direct Servo Control

```
SYMBOL servo = 0
OUTPUT servo
SYMBOL position = B2          \ 100 x 10 µsec = 1000 µsec = 1 msec
                               \ 200 x 10 µsec = 2000 µsec = 2 msec

start:
FOR position = 100 TO 200 STEP 2  \ 1 to 2 msec
  PULSOUT servo, position
  PAUSE 10
NEXT position
FOR position = 200 TO 100 STEP -2 \ 2 to 1 msec
  PULSOUT servo, position
  PAUSE 10
NEXT position
GOTO start
```

# Servo Motors (cont)

## Direct Servo Control

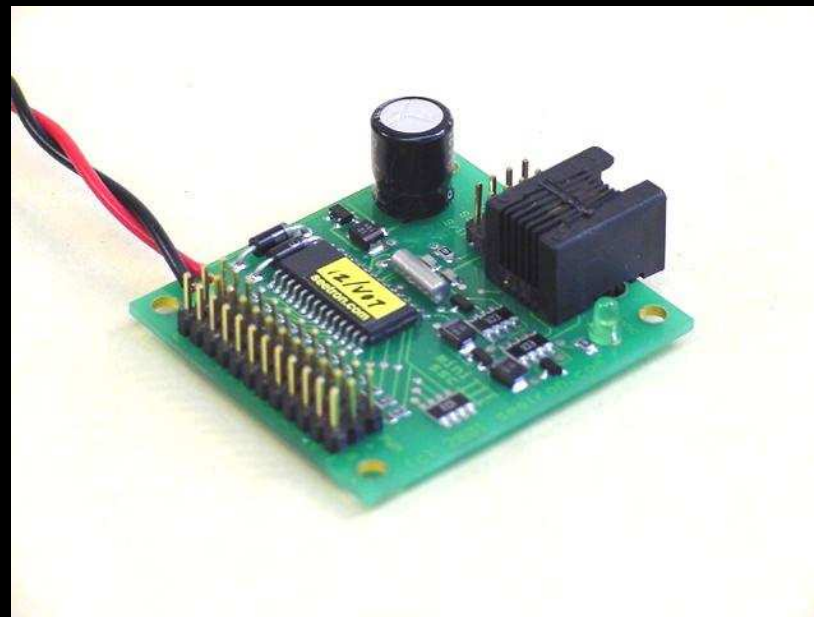
```
SYMBOL servo = 0
OUTPUT servo
SYMBOL position = B2           \ 100 x 10 µsec = 1000 µsec = 1 msec
                                \ 200 x 10 µsec = 2000 µsec = 2 msec

start:
FOR position = 100 TO 200 STEP 2 \ 1 to 2 msec
  PULSOUT servo, position
  PAUSE 10
NEXT position
FOR position = 200 TO 100 STEP -2 \ 2 to 1 msec
  PULSOUT servo, position
  PAUSE 10
NEXT position
GOTO start
```

# Servo Motors (cont)

Dedicated Servo Controller

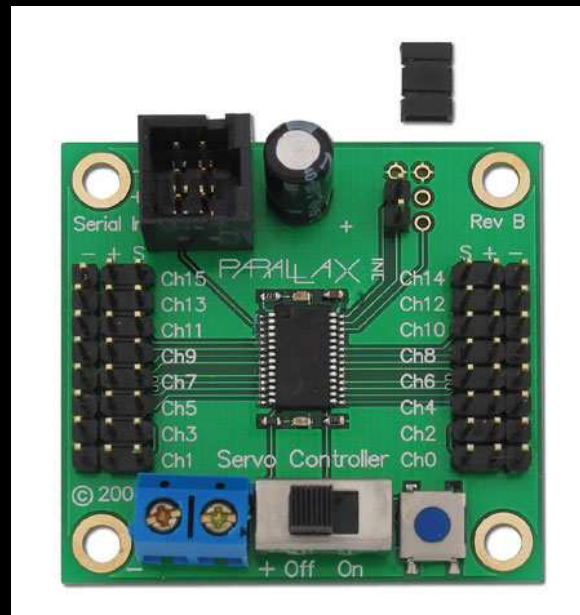
**Seetron SSC-12**



# Servo Motors (cont)

Dedicated Servo Controller

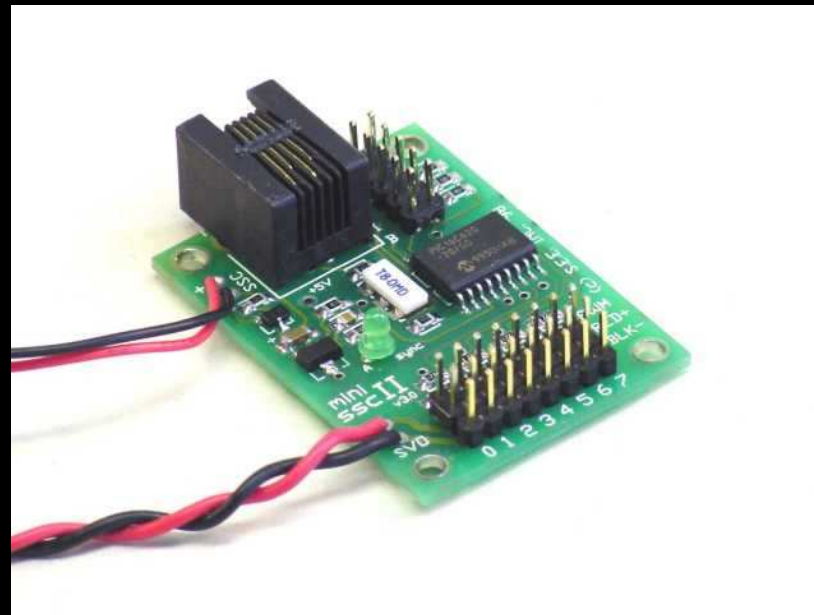
**Parallax Servo Controller (PSC)**



# Servo Motors (cont)

Dedicated Servo Controller

**Seetron MiniSSC-II**



# Servo Motors (cont)

## Dedicated Servo Controller

```
SYMBOL servo = 0
SYMBOL sync = 255
SYMBOL position = B2
```

```
start:
```

```
FOR position = 0 TO 254 STEP 1
  SEROUT 0, N2400, (sync,servo,position)
NEXT position
FOR position = 254 TO 0 STEP -1
  SEROUT 0, N2400, (sync,servo,position)
NEXT position
GOTO start
```

# Servo Motors (cont)

## Dedicated Servo Controller

```
SYMBOL servo = Seetron MiniSSC-II  
SYMBOL sync = 255  
SYMBOL position = B2
```

```
start:
```

```
FOR position = 0 TO 254 STEP 1  
    SEROUT 0, N2400, (sync,servo,position)  
NEXT position  
FOR position = 254 TO 0 STEP -1  
    SEROUT 0, N2400, (sync,servo,position)  
NEXT position  
GOTO start
```

# Servo Motors (cont)

## Dedicated Servo Controller

```
SYMBOL servo = Seetron MiniSSC-II  
SYMBOL sync = 255  
SYMBOL position = B2
```

```
start:  
FOR position = 0 TO 254 STEP 1  
  SEROUT 0, N2400, (sync,servo,position)  
NEXT position  
FOR position = 254 TO 0 STEP -1  
  SEROUT 0, N2400, (sync,servo,position)  
NEXT position  
GOTO start
```



# Passive Infrared Motion Detector

**Circuit Specialists PIR**

**Part #: K76, \$11.95**



# Passive Infrared Motion Detector

## **Circuit Specialists PIR**

**Fully Assembled and Tested**

**Connections: Power, Ground and Output**

**Input Voltage: 5 – 10 VDC**

**Output Voltage: 0.5 sec Pulse**

**Range: ~10'**

# Digital Voice Recorder (cont)

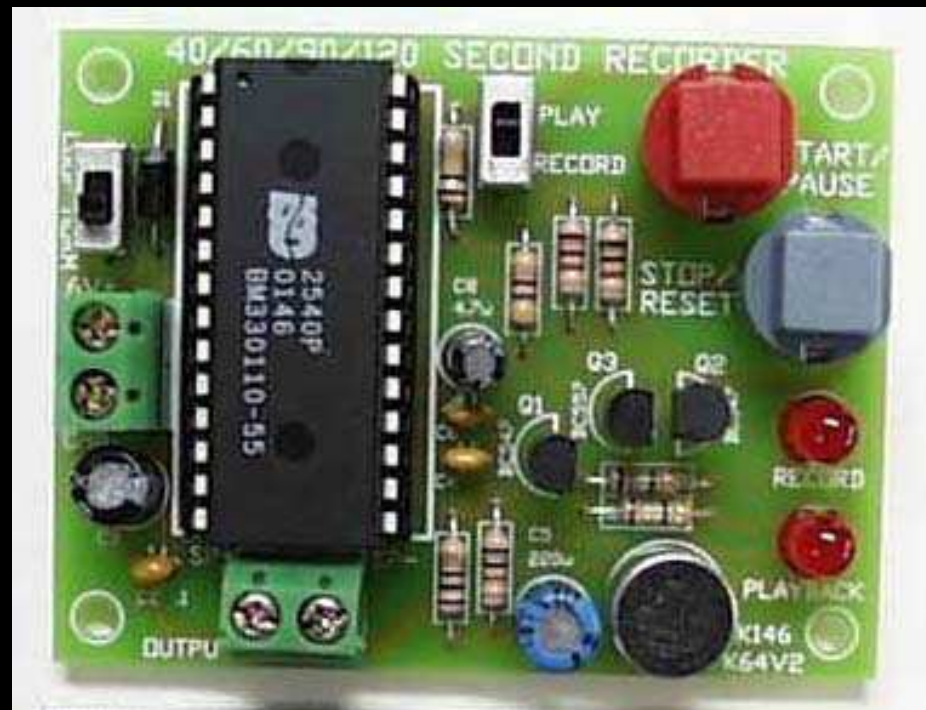
**Cowlacious ISD® Chipcorder® Player Board**  
**\$20.95 (Assembled unit with 60 Second ISD Chip)**



# Digital Voice Recorder (cont)

**QualityKits 40 Second Message Recorder Kit**

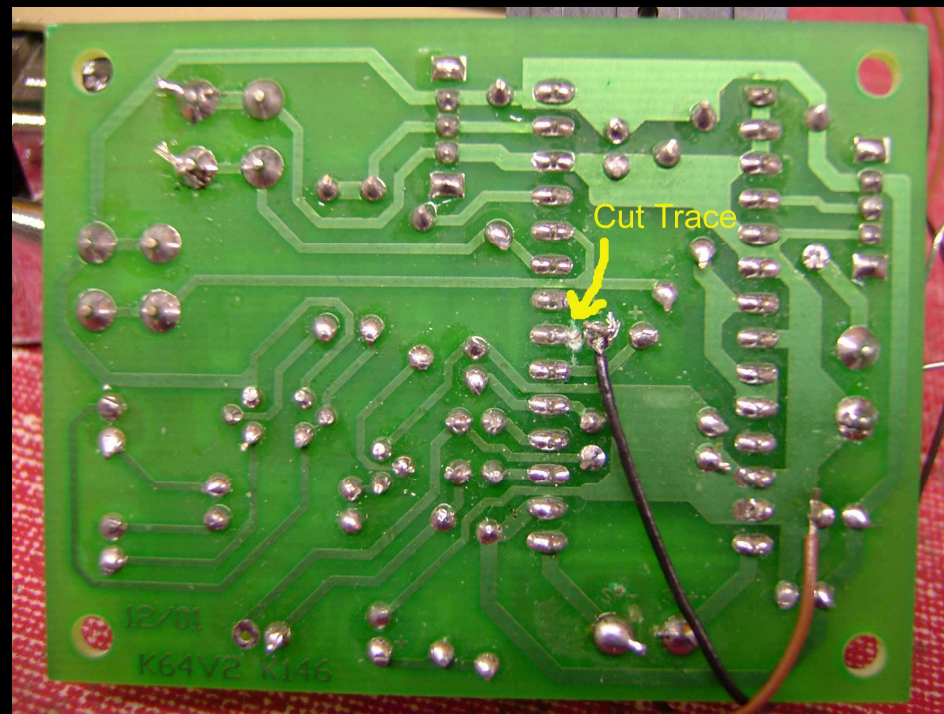
**QualityKits Part #: QK146, \$18.95**



# Digital Voice Recorder (cont)

## QualityKits 40 Second Message Recorder Kit

QualityKits Part #: QK146, \$18.95

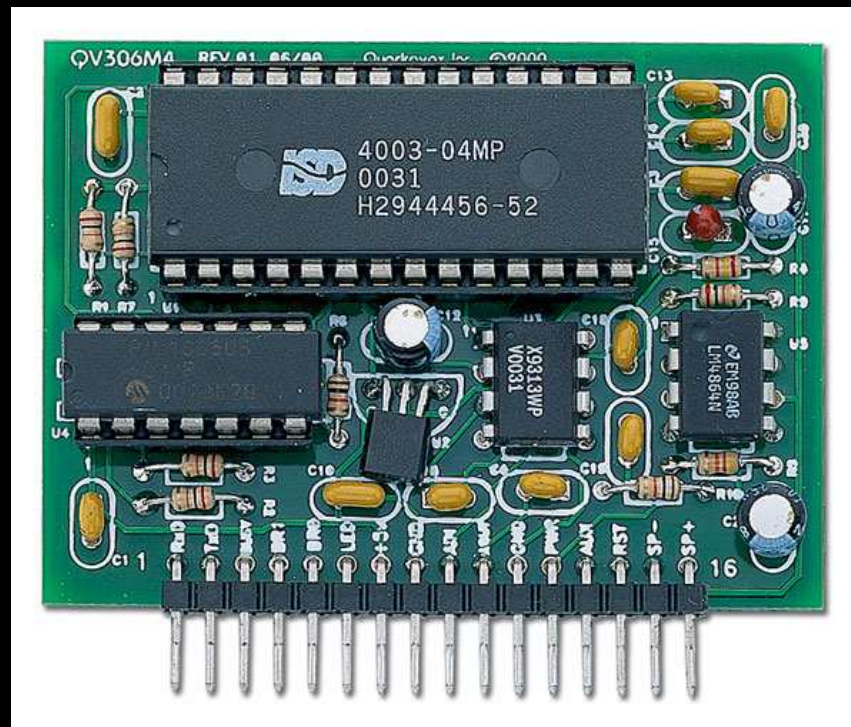




# Digital Voice Recorder (cont)

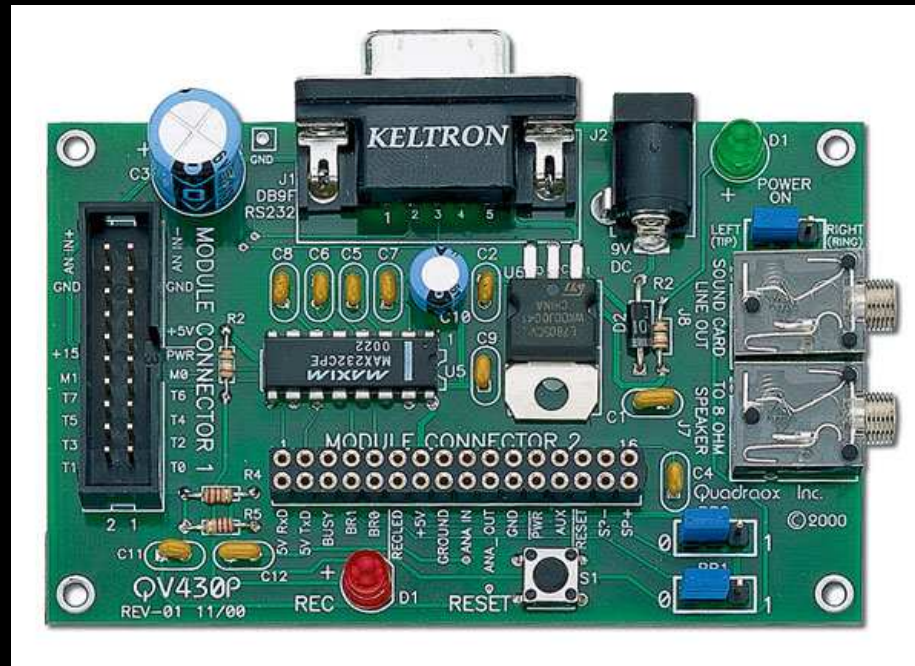
## Quadravox QV306M4P Playback Module

Parallax Part #: 27967, \$69.00



# Digital Voice Recorder (cont)

**Quadravox QV430P Sound Programmer**  
**Parallax Part #: 27968, \$79.00**



# Digital Voice Recorder

## Radio Shack 20-sec Recording Module

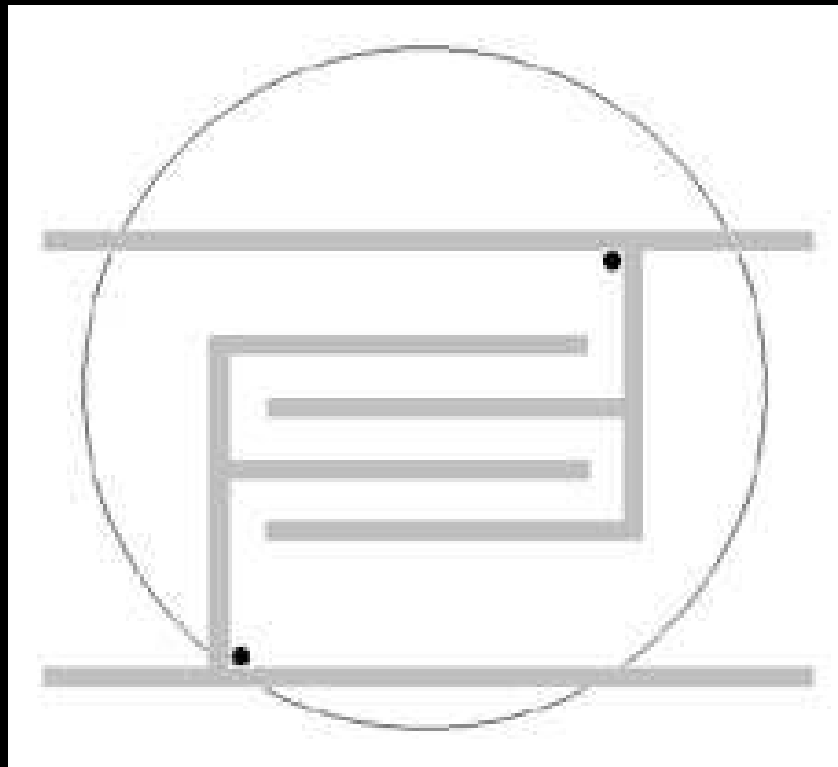
Radio Shack Part #: 276-1323, \$9.99





# Digital Voice Recorder (cont)

## Radio Shack 20-sec Recording Module



# Haunt Controllers (cont)

## Example: Single-Input, Multi-Output, Single-Step

```
SYMBOL cylinder = 0
OUTPUT cylinder
SYMBOL dvr = 1
OUTPUT dvr
SYMBOL pir = PIN7
INPUT pir

HIGH dvr

start:
IF pir = 0 THEN start
HIGH cylinder
LOW dvr
PAUSE 100
HIGH dvr
PAUSE 3000  ` Delay 3 sec
LOW cylinder
PAUSE 30000 ` Delay 30 sec
GOTO start
```

# Haunt Controllers (cont)

## Example: Single-Input, Multi-Output, Single-Step (cont)

```
SYMBOL cylinder = 0
OUTPUT cylinder
SYMBOL dvr = 1
OUTPUT dvr
SYMBOL pir = PIN7
INPUT pir

HIGH dvr

start:
IF pir = 0 THEN start
HIGH cylinder
LOW dvr
PAUSE 100
HIGH dvr
PAUSE 3000  ` Delay 3 sec
LOW cylinder
PAUSE 30000 ` Delay 30 sec
GOTO start
```

# Haunt Controllers (cont)

## Example: Single-Input, Multi-Output, Single-Step (cont)

```
SYMBOL cylinder = 0
OUTPUT cylinder
SYMBOL dvr = 1
OUTPUT dvr
SYMBOL pir = PIN7
INPUT pir

HIGH dvr

start:
IF pir = 0 THEN start
HIGH cylinder
LOW dvr
PAUSE 100
HIGH dvr
PAUSE 3000  \ Delay 3 sec
LOW cylinder
PAUSE 30000 \ Delay 30 sec
GOTO start
```

# Haunt Controllers (cont)

## Example: Single-Input, Multi-Output, Single-Step (cont)

```
SYMBOL cylinder = 0
OUTPUT cylinder
SYMBOL dvr = 1
OUTPUT dvr
SYMBOL pir = PIN7
INPUT pir

HIGH dvr

start:
IF pir = 0 THEN start
HIGH cylinder
LOW dvr
PAUSE 100
HIGH dvr
PAUSE 3000  \ Delay 3 sec
LOW cylinder
PAUSE 30000 \ Delay 30 sec
GOTO start
```

# Real-Time BS1 Code Generator

## WSHACS-BS1

WSHACS-BS1

Wait For:

Front-End Delay:  seconds

	Initial	Mute	State
Pin 0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Pin 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Pin 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Pin 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Pin 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Pin 5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
Pin 6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>

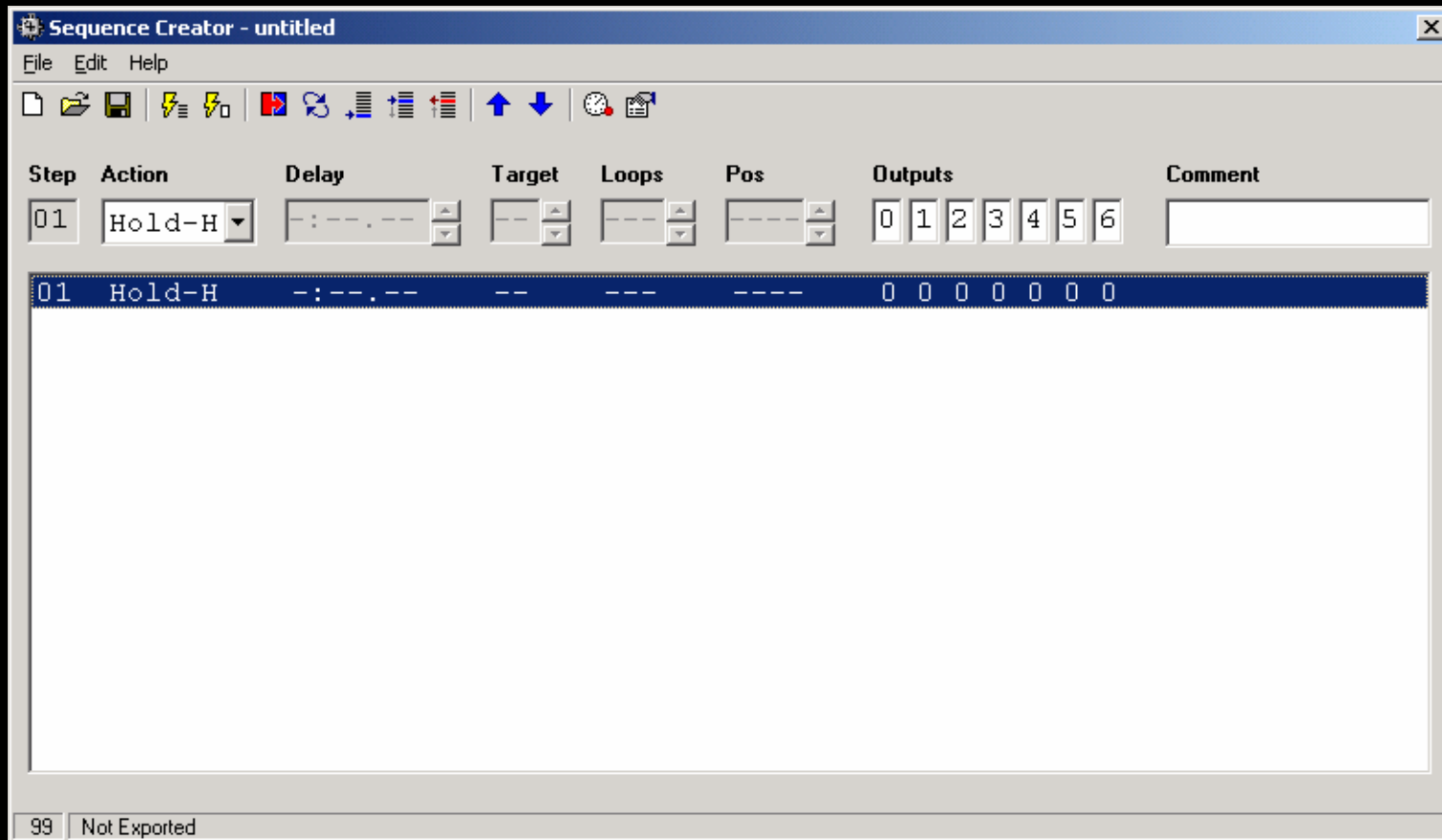
Elapsed Time:  seconds

Back-End Delay:  seconds

Output File:  (8 chars max)

# Real-Time B51 Code Generator (cont)

## Sequence Creator



For More Information



<http://www.methodzofmadness.com>



For More Information (cont)



<http://www.wickedstone.com>

# For More Information (cont)

## **Terror by Design**

<http://www.terrorbydesign.com>

## **Cowlacious Designs**

<http://www.cowlacious.com>

## **Circuit Specialists**

<http://www.web-tronics.com>

## **Radio Shack**

<http://www.radioshack.com>

# For More Information (cont)

## **Parallax**

<http://www.parallax.com>

## **Scott Edward's Electronics , Inc. (seetron)**

<http://www.seetron.com>

## **Chauvet Lighting**

<http://www.chauvetlighting.com>

## **High Energy Lighting and Sound**

<http://www.cheaplights.com>