

STP100/PP232-485F

The STP100 is the perfect choice for small to medium stepper motor applications where step precession to 32 bits is required. The STP100 is divided into four functional blocks (Embedded Microprocessor, Stepper Logic, Motor Driver Circuit, and Serial Line Drivers). The embedded microprocessor gives the STP100 the ability to keep track of an absolute motor position. Among other things, the position is a signed 32-bit number. Other features of the STP100 that are microprocessor controlled include:

Motor acceleration and deceleration

Human readable command set

Multiple board command processing

Cued commands

STP100 Board status queries:

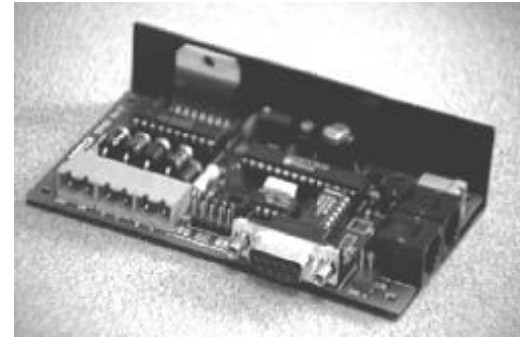
(Motor Position, Pin Conditions, etc.)

The stepping logic for the STP100 is supplied by a SGS Thompson Stepper chip with built in current limiting and step logic. The driver chip supplies up to two Amps per stepper motor winding. The STP100 has the ability to drive both RS-232A and RS-485 serial lines.

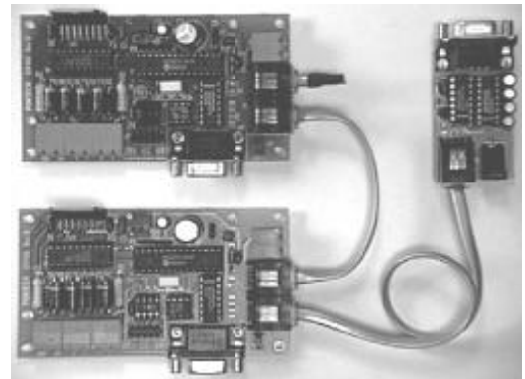
The STP100 accepts serial data from a host computer and outputs amplified control signals for a bipolar stepper motor. There are four general purpose I/O lines that can be used for detecting home positions or motor limits. In addition, two of the I/O lines can be configured as 8-bit Analog to digital converters.

With the addition of a PONTECH PP232-485F you can take advantage of the STP100 on board RS-485 multiple drop network connection for addressing up to 255 boards. Cued commands can then be used for controlling two or more axis systems.

Possible uses for the STP100 include automated manufacturing systems, x-y tables, robotics.



STP100



Two STP100's connected
to a PP232-485F

PONTECH™

© 1997-98 Pontech

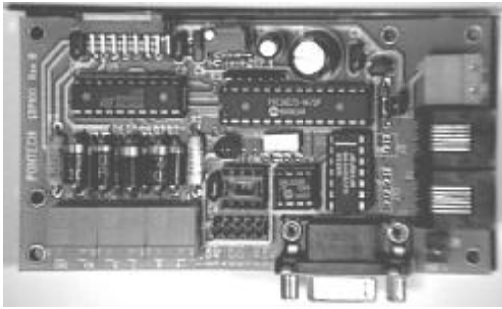
<http://www.pontech.com>

Phone: (714) 985-9286 - Fax: (714) 985-9288

2700 E. Imperial Hwy., Suite N - Brea, CA 92821 - USA

All trademarks and registered trademarks belong to their respective owners.

STP100



Sample interface program using QBASIC

```
OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
DO
PRINT: PRINT "***STP100 Controller**"
INPUT "Board ID Number :"; ID$
INPUT "Position of Stepper :"; Position$

PRINT #1,"BD";ID$
PRINT #1,"MI";Position$
LOOP
```

Example screen of above program when run.

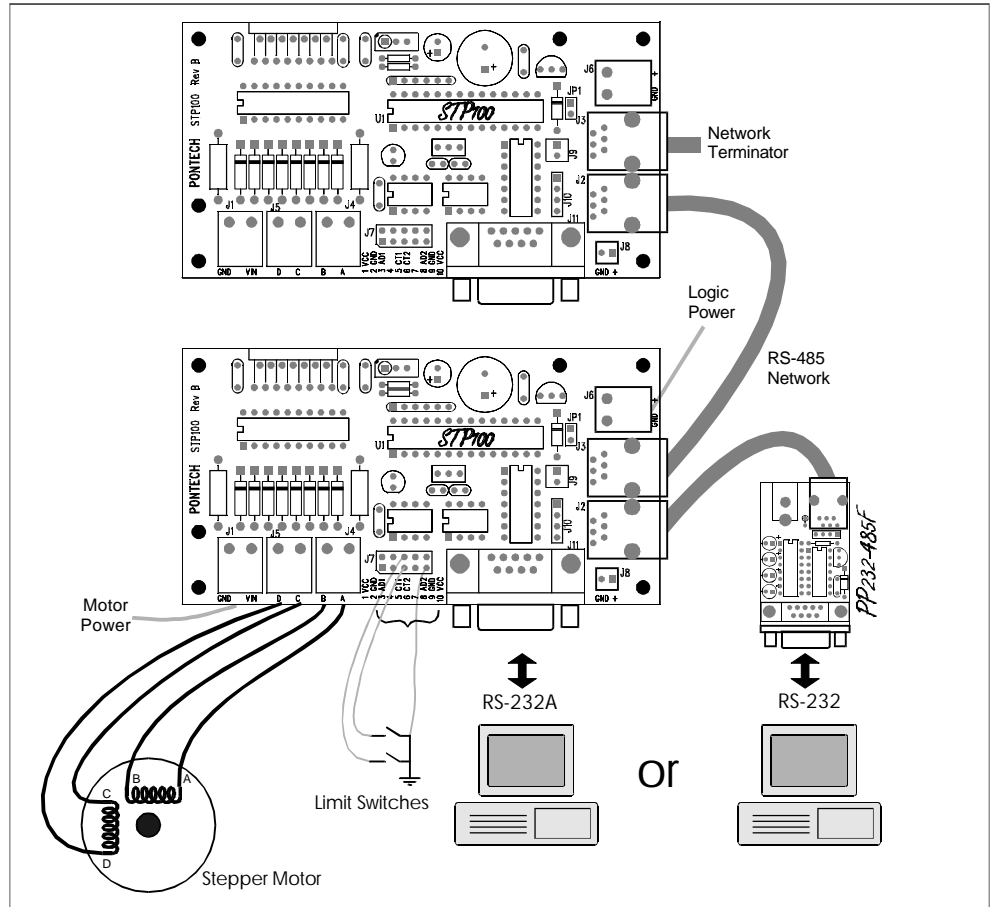
```
**STP100 Controller**      Enable Board with ID # = 1
Board ID Number : 1       move Stepper to
Position of Stepper : -200 position -200

**STP100 Controller**      Return all steppers to
Board ID Number : 0       home position
Position of Stepper : 0   if ID # = 0, all STP100
                          boards will be enable

**STP100 Controller**      Enable Board with ID # = 2
Board ID Number : 2       move Stepper to
Position of Stepper : 1000 position 1000
```

Partial Command Listing

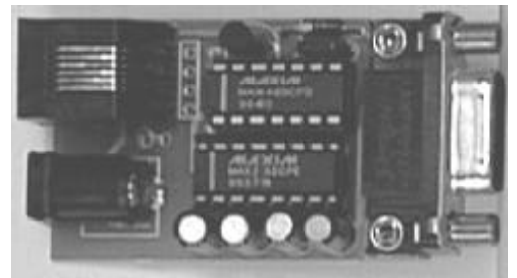
Command	Parameter (n)	Description
BDn	0 - 255	Board Select
Min	32 bit	Move
MCn	32 bit	Move Cued
IIn	32 bit	Increment
ICn	32 bit	Increment Cued
CU	None	Cue to Move
PCn	3,5,6,8	Pin Clear
PSn	3,5,6,8	Pin Set
RPn	3,5,6,8	Read Pin
ADn	1,2	Get A/D Value
WSS	None	Write System Settings EEPROM
RC	None	Read Current Pos.
RD	None	Read Dest. Pos.
RT	None	Read Delta Pos.
HMn	32 bit	Set bew home
HI	None	Halt Immediatley
H0	None	Halt w/ deceleration
H+	None	Move CW forever
H-	None	Move CCW forever
SP	None	Stepper Powered
SO	None	Stepper Off
SH	None	Step Half
SF	None	Step Full
SW	None	Step Wave
TCn	3,5,6,8	Test if pin Clear
TSn	3,5,6,8	Test if pin Set
SAn	1-255	Set Acceleration
SDn	16 bit	Set Step Delay
SMn	16 bit	Set Min Step Delay



[FEATURES]

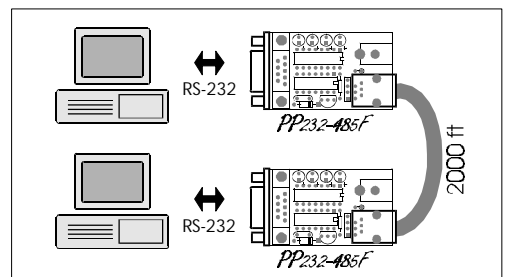
- * Controls one bipolar stepper motor with 32 bit absolute position
- * Interface to PC through RS232 or RS-485 (With optional board) Serial port (2400 to 19200 baud).
- * User definable board ID number allowing multiple board to share same serial line (RS-485 only). Addressable up to 255 boards
- * 2 Ch general purpose, 8 bit A/D input port for reading 0 - 5 Volts.
- * Drive up to 2 Amps per phase 5-46 VDC bipolar motor
- * Chopping current limiting & Over temperature protection
- * acceleration, deceleration, speed, direction, position
- * 4 bit digital input for home limits
- * Full, half or wave step modes
- * Size (4.3 x 2.4 x 1.1 in.)
- * Simple ASCII string commands with sample interface software

PP232-485F



[FEATURES]

- * Converts RS-232 level signals to RS-485 level signals
- * Use two to extend RS-232 cable runs to 2000 ft.
- * Operates on 7-15VDC

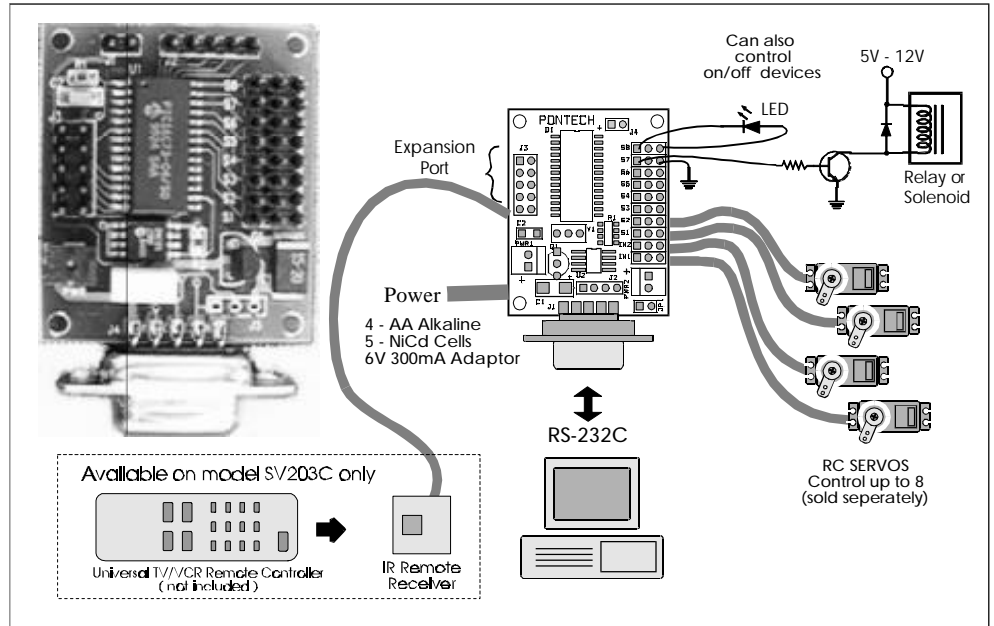


SV203

The SV203 is a Microchip PIC16C73 microcontroller based servo motor controller board. It accepts RS232 serial data signals from a host computer and outputs PWM (pulse width modulated) signals to control up to eight RC servo motors (servos used in radio-controlled model airplanes, cars, etc.). Unused servo pins can be reconfigured for digital output to drive on/off devices.

A 5 channel, 8 bit A/D input is available to read analog voltages between 0 to 5 Volts. Devices such as an analog joystick or potentiometers can be connected to this port and their position can be read by the PC and sent back to the board to control the servo position.

The SV203 processes commands sent by a host computer connected to the serial port. The commands are ASCII character strings that select the board, tell which servo to control, and the position of the servo.



SV203B

The SV203B has all the features of the SV203 plus the added feature of being able to run BASIC programs. It can operate on its own without being connected to a computer once the sequence of movements are programmed into the board. Its onboard 8K EEPROM can store special move commands such as slow moves, or different sequences of movements that can be programmed and played back later.

A special structured BASIC compiler and downloader are provided to make programming the board as easy as possible. BASIC commands and features such as For/Do loops, IF/Then/Else, Goto, Delays, Variables and Subroutines allow for very powerful sequences.

SV203C

The SV203C has one other feature added to model SV203B. It comes with an IR sensor that plugs into one of the I/O pins. The sensor accepts signals generated by a TV or VCR remote controller. The board will work with any universal IR remote controller, or any Sony brand TV remote controller.

The board can be programmed to perform different subroutines when a button is pressed on the remote controller.

The signals sent out by the IR remote can also be converted by the SV203C board and sent out via the RS232 port. This makes a simple way of interfacing an IR remote to a PC.

[FEATURES]

- * Controls 1 to 8 servos per board
8 bit resolution,
value from 1 to 255,
Under one degree precision
- * Servo port can be reconfigured for digital output to drive on/off devices. Source/Sink 25mA per pin
- * Interface to PC through RS232 Serial port (2400 to 19200 baud).
- * User definable board ID number (allowing multiple board to share same serial line).
- * 5 Ch, 8 bit A/D input port for reading 0 - 5 Volts. (Control and save servo positions via Joystick/Pot)
- * An SPI port for shifting in/out serial data
- * 8k EEPROM for storing stand alone routines or special movements. (version B & C)
- * IR-Receiver feature (version C)
- * Dimensions: 1.4 in X 1.7 in
- * Servo Connectors: 3 pin sip. Futaba J-type connectors.
- * Power supply: 4.8V to 6.0V

Sample interface program using QBASIC

```
OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1
DO
PRINT:PRINT "**SV203 Controller**"
INPUT "Board ID Number :"; ID$
INPUT "Servo # to control :"; Servo$
INPUT "Position of Servo :"; Position$

PRINT #1,"BD";ID$;"SV";Servo$;"M";Position$
LOOP
```

Example screen of above program when run.

```
**SV203 Controller**      Enable Board with ID # = 1
Board ID Number : 1      move Servo #2 to
Servo # to control : 2   position 200
Position of Servo : 200

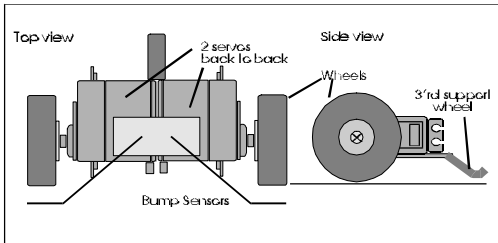
**SV203 Controller**      Turn Servo #3 off
Board ID Number : 0      if ID # = 0, board will
Servo # to control : 3   be enable regardless
Position of Servo : 0    of Board ID number

**SV203 Controller**      Enable Board with ID # = 2
Board ID Number : 2      move Servo #1 to
Servo # to control : 1   position 128
Position of Servo : 128
```

Partial Command Listing

Command	Parameter (n)	Description
BDn	0 - 255	Board Select
SVn	1 - 8	Servo Select
Mn	0 - 255	Move to Position
In	-128 - 127	Relative Move
ADn	1 - 5	Get A/D
SO n	0 - 155	Shift Out
SI	None	Shift In

Mini Robot



These application notes demonstrate some of the capabilities of the SV203B and SV203C.

The simple two wheel robot above was constructed by taping two servos to the back and attaching hobby airplane wheels to the servo shaft. The servos have been modified by removing the internal stop tab and potentiometer. The potentiometer is replaced with a fixed voltage divider resistor or a trim pot. This allows the servo to be used as a continuously spinning motor. The speed and direction, instead of position, can then be controlled with the SV203B or SV203C. The two bump sensors are just switch contacts connected to two of the A/D ports.

The code below can be compiled and downloaded to the SV203B/C and run autonomously.

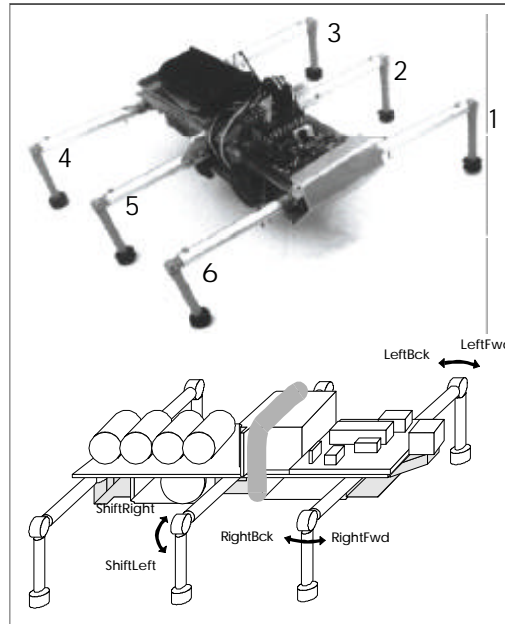
```
Dim BumpL as Bit 0 of RC      'Sensor 1
Dim BumpR as Bit 1 of RC    'Sensor 2
```

```
Const Stop = 0
Const Fwd1 = 10
Const Fwd2 = 245
Const Rev1 = 245
Const Rev2 = 10
```

```
Main:
Servo1 = Fwd1      'Forward
Servo2 = Fwd2
If BumpL = 1 then
  Call HitLeft
End if
if BumpR = 1 then
  Call HitRight
end if
goto Main
```

```
Sub HitLeft
  Servo1 = Rev1    'Reverse
  Servo2 = Rev2
  Delay 1000
  Servo1 = Fwd1   'Spin Right
  Delay 500
End Sub
```

```
Sub HitRight
  Servo1 = Rev1    'Reverse
  Servo2 = Rev2
  Delay 1000
  Servo1 = Fwd2   'Spin Left
  Delay 500
End Sub
```



Hexo Robot

The above six legged walking robot uses three servos. One of the servos is used to rock the body back and forth and, in turn, lifts up legs 1,3,5 or legs 2,4,6. The other two servos are used to swing legs 1,3 on the left side, or 4,6 on the right side, forward and backward.

By performing the sequences in the above-right figure, walking movements such as forward, reverse, spin left, and spin right can be accomplished.

The code in Listing 2 demonstrates the walking sequence and how the IR feature in the SV203C can be used.

```
Dim BumpL as Bit 0 of RC      'Sensor 1
Dim BumpR as Bit 1 of RC    'Sensor 2
Dim I as Byte                'Declare Variable
```

```
Const StepDelay = 300
Const LeftFwd = 70
Const LeftBck = 180
Const RightFwd = 180
Const RightBck = 70
Const ShiftLeft = 95
Const ShiftRight = 145
```

```
fIRenable = 1      'Enable IR receiver
```

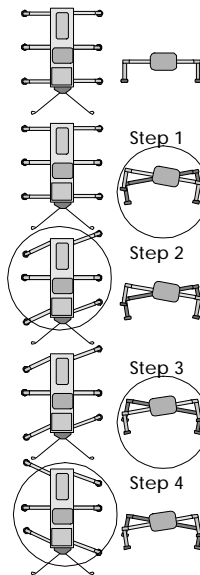
```
Main:
if fIRaval then    'IR received?
  If IRreg = "A" then Call WFWd 'Ch Up
  If IRreg = "B" then Call WRev 'Ch Dwn
  If IRreg = "C" then Call WLeft 'Vol Up
  If IRreg = "D" then Call WRight 'Vol Dwn
```

```
  If IRreg = "F" then Call Auto 'Power
```

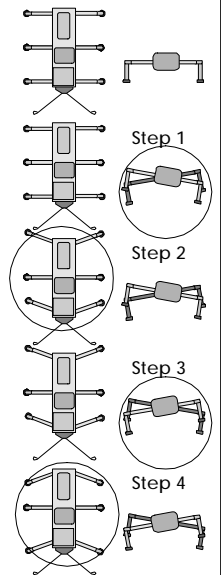
```
  fIRaval = 0     'Clear Flag
End if
Goto Main
```

'----Continued on the right ---

Forward/Reverse



Spin Left/Right



```
Sub WFWd
  Servo3 = ShiftLeft 'Step1
  Delay StepDelay
  Servo1 = LeftFwd   'Step2
  Servo2 = RightBck
  Delay StepDelay
  Servo3 = ShiftRight 'Step3
  Delay StepDelay
  Servo1 = LeftBck   'Step4
  Servo2 = RightFwd
  Delay StepDelay
End Sub
```

```
Sub WBck
  Servo3 = ShiftLeft 'Step1
  Delay StepDelay
  Servo1 = LeftBck   'Step4
  Servo2 = RightFwd
  Delay StepDelay
  Servo3 = ShiftRight 'Step3
  Delay StepDelay
  Servo1 = LeftFwd   'Step2
  Servo2 = RightBck
  Delay StepDelay
End Sub
```

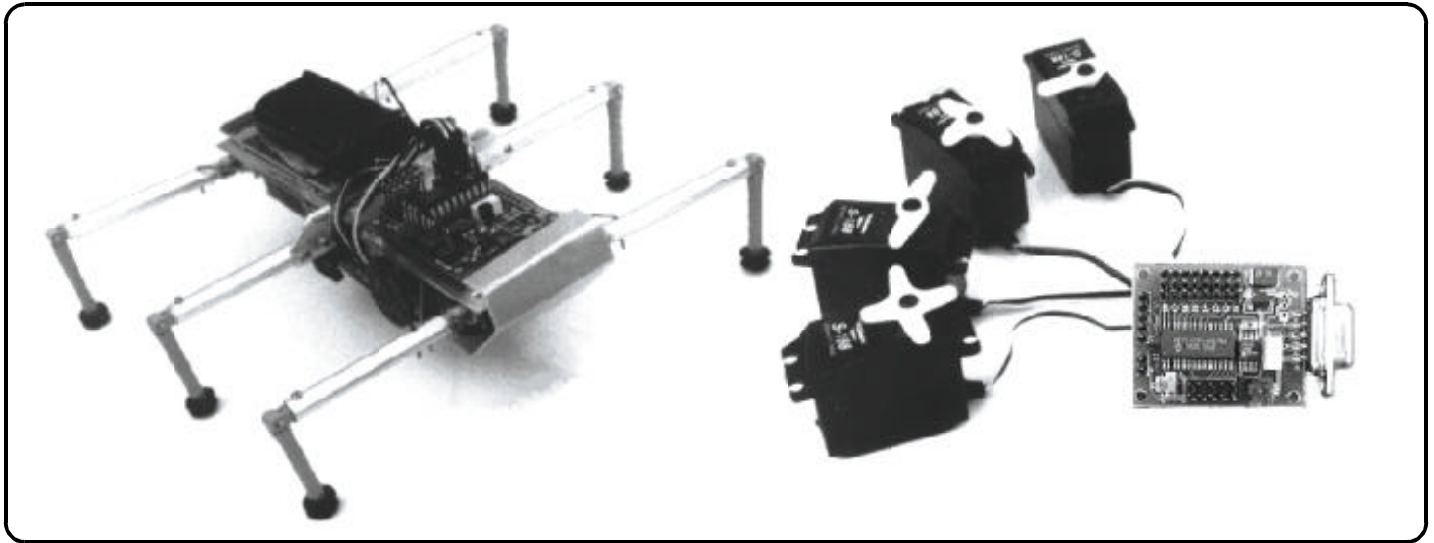
```
Sub WLeft
  Servo3 = ShiftLeft 'Step1
  Delay StepDelay
  Servo1 = LeftBck   'Step2
  Servo2 = RightBck
  Delay StepDelay
  Servo3 = ShiftRight 'Step3
  Delay StepDelay
  Servo1 = LeftFwd   'Step4
  Servo2 = RightFwd
  Delay StepDelay
End Sub
```

```
Sub WRight
  Servo3 = ShiftLeft 'Step1
  Delay StepDelay
  Servo1 = LeftFwd   'Step4
  Servo2 = RightFwd
  Delay StepDelay
  Servo3 = ShiftRight 'Step3
  Delay StepDelay
  Servo1 = LeftBck   'Step2
  Servo2 = RightBck
  Delay StepDelay
End Sub
```

```
Sub Auto
  ....
```

Listing 1.

Listing 2.



2700 E. Imperial Hwy., Suite N
Brea, CA 92821