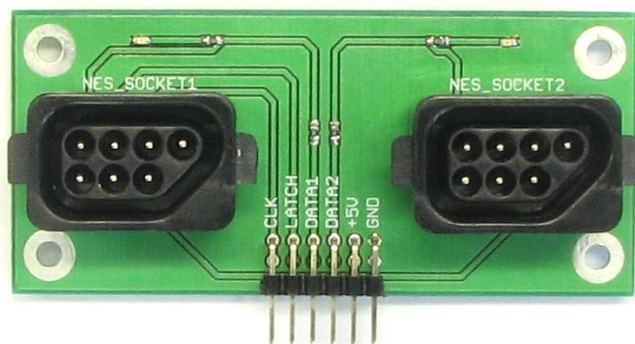


## NES Gamepad Controller Adapter (#32368)

The NES Gamepad Controller Adapter is a printed circuit board with two Nintendo® gamepad compatible sockets routed to a dual row of pins with breadboard spacing. This adapter makes it convenient to connect up to two gamepad controllers to a breadboard for gamepad-controlled projects.

NOTE: This adapter is designed for gamepads only. It does not have signal lines for Nintendo Zappers and Power Pads.



### Features

- Breadboard-friendly 0.1 inch pin spacing
- Works great with Nintendo gamepad style controllers (#32365)
- Plated holes for mounting in a project box
- Two adjacent SIP headers for stability
- Data connection indicator LEDs
- Compatible with BASIC Stamp®, SX, and Propeller™ microcontrollers
- Built-in series resistors protect 3.3 V Propeller I/O pin inputs

### Key Specifications

- Power requirements: +5 VDC, 25 mA
- Communication: TTL synchronous serial
- Dimensions: 3.1 x 1.6 x 0.91 in (79 x 41 x 23 mm)
- Operating temperature: 32 to 131 °F (0 to 55 °C)

### Application Ideas

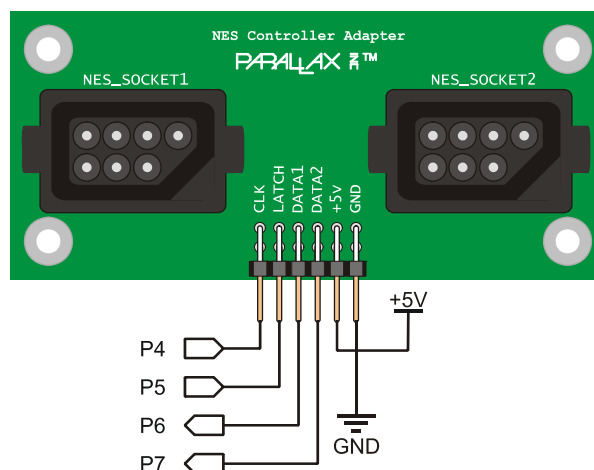
- Gamepad controlled robot
- Gamepad controlled mechanical arm
- Video game system on a breadboard

BASIC Stamp and Propeller are registered trademarks or trademarks of Parallax Inc. NES and Nintendo are registered trademarks or trademarks of Nintendo of America Inc. All trademarks herein are the property of their respective owners.

## Quick Start Circuit

Figure 1 shows a schematic that can be used with Nintendo gamepad controllers and the BASIC Stamp 2 and Propeller test programs featured in the Source Code section. P4 and P5 drive the clock and latch lines. P6 receives data from NES\_SOCKET1, and P7 receives data from NES\_SOCKET2.

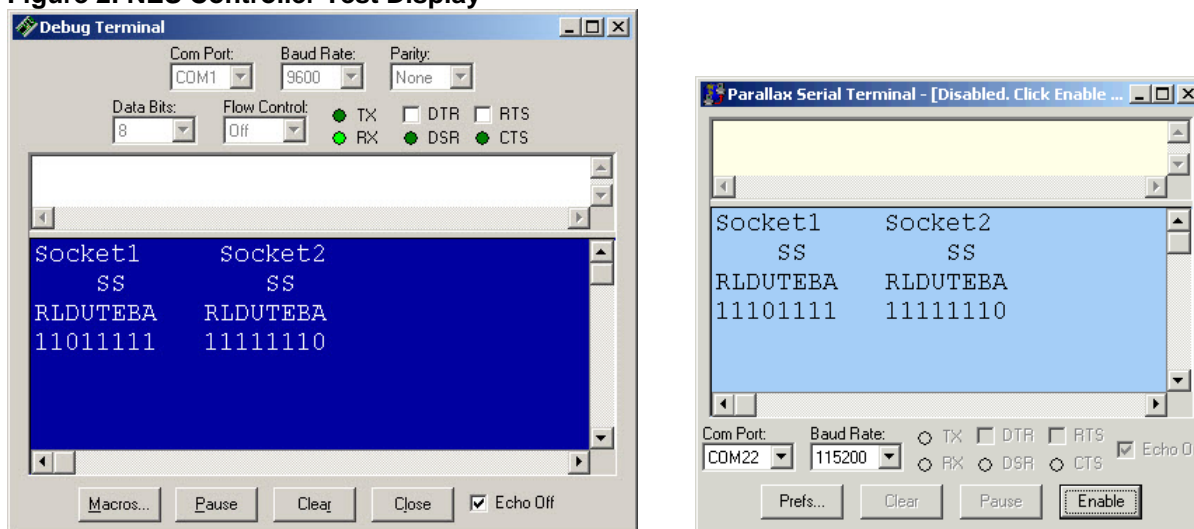
Figure 1: Quick Start Schematic



## Connecting and Testing

Figure 2 shows test results for the BASIC Stamp and Propeller microcontrollers while they execute the test programs featured in the Source Code section. Messages from the BASIC Stamp are displayed by the Debug Terminal (left), and messages from the Propeller chip are displayed by the Parallax Serial Terminal (right). The buttons from left to right are Right, Left, Down, Up, Start, Select, B, and A. When one of those buttons is pressed and held, the corresponding bit displays a 0; otherwise, it displays a 1. The AA button causes the A bit to cycle on/off while, and likewise with the BB button. The SLOW button toggles the ST bit. Press it once, and the ST bit cycles on/off; press it again, and it returns to 1.

Figure 2: NES Controller Test Display



- ✓ Build the circuit shown in Figure 1.
- ✓ Download the Source Code from the 32368 product page at [www.parallax.com](http://www.parallax.com).
- ✓ Load it into your microcontroller.
- ✓ Use the Debug Terminal (BASIC Stamp) or the Parallax Serial Terminal (Propeller) to display the status display from your microcontroller. See Figure 2.
- ✓ Try pressing the buttons on the controller and examine the way each bit responds.

## Resources and Downloads

The NES Gamepad Controller Adapter product page has this PDF, source code, and a link to Chapter 6 in the Hydra Game Development System manual for additional information about the Nintendo gamepad. Go to [www.parallax.com](http://www.parallax.com) and "search" for 32368.

## Device Information

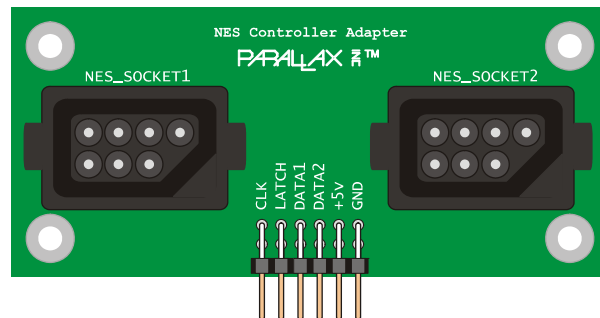
### Pin Definitions

Figure 3 shows the module pinout for the pin definitions listed in Table 1. Ratings are not listed because they will be specific to the make and manufacture of the gamepad that gets connected to the adapter.

**Table 1: Pin Definitions**

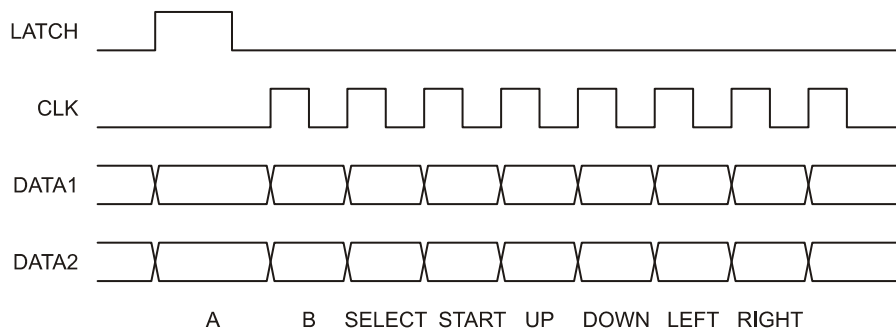
Name	Function
CLK	Clock input
LATCH	Latch input
DATA1	Data output for NES_SOCKET1
DATA2	Data output for NES_SOCKET2
+5V	Positive 5 V supply connection
GND	Ground 0 V supply connection

**Figure 3: Pinout**

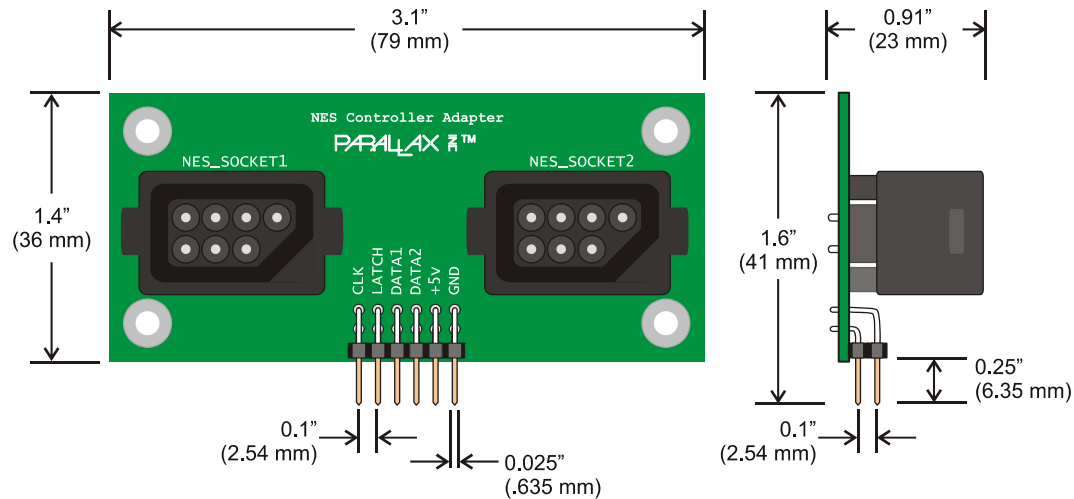


### Communication Protocol

A pulse to the LATCH pin causes the gamepad to latch the button values and transmit the first (A button) bit. The rising edges of the next seven clock pulses deliver the remaining bits (B through RIGHT buttons). Although both data lines can be shifted in simultaneously, the microcontroller does not necessarily have to record both data lines at once. For example, the BASIC Stamp 2 test code in the source code section checks each DATA line individually with separate PULSOUT and SHIFTIN commands. Note that since the first data value is available before the first clock pulse, the PBASIC SHIFTIN command uses the LSBPRE argument to specify that the data line transmits the latest value as a pre-clock pulse.



### Module Dimensions



## Source Code

### BASIC Stamp<sup>®</sup> 2 Program

```
' Test NES Controllers.bs2
' Display NES controller bit states, see Parallax NES Controller
' Adapter (Item# 32368) documentation at www.parallax.com.

' {$STAMP BS2}
' {$PBASIC 2.5}

value          VAR      Byte(2)
' Store NES data

PAUSE 1000
' Wait a second
DEBUG "Socket1      Socket2", CR,
      "      SS      SS", CR,
      "RLDUTEBA      RLDUTEBA", CR

LOW 5
' Initialize LATCH
LOW 4
' Initialize CLK

DO
' Main loop

  PULSOUT 5, 10
  SHIFTIN 6, 4, LSBPRE, [value(0) \8]
' Latch the data
' Shift in socket 1 data

  PULSOUT 5, 10
  SHIFTIN 7, 4, LSBPRE, [Value(1) \8]
' Latch the data
' Shift in socket 1 data

  DEBUG CRSRXY, 0, 3, BIN8 value(0),
          CRSRX, 11, BIN8 value(1)
' Display values

LOOP
' Repeat main loop
```

## Propeller Test Application

```

''File: Test Two NES with PST.spin
''Test two NES controllers and display with Parallax Serial Terminal.
''Requires the NES.spin from John Abshier's NES Controller object. It's available from the
''Parallax Object Exchange at obex.parallax.com.

```

CON	' Constant declaration block
<pre> _clkmode = xtall + pll16x _xinfreq = 5_000_000  DATA2      = 7 DATA1      = 6 LATCH      = 5 CLK        = 4 </pre>	<pre> ' 5 MHz crystal → 80 MHz system clock  ' I/O pin &amp; power connections '       P7 ← DATA2 '       P6 ← DATA1 '       P5 → CLK '       P4 → LATCH '       5 V → +5V '       (Vss) GND → GND </pre>
OBJ	' Object declaration block
<pre> pst      : "Parallax Serial Terminal" nes[2]   : "NES" </pre>	<pre> ' Serial Terminal display ' NES object </pre>
PUB Main   buttons[2], index, dataPin	' Main method
<pre> ' Initialize display pst.Start(115200) pst.Str(String("Socket1   Socket2", pst#NL)) pst.Str(String("   SS       SS", pst#NL)) pst.Str(String("RLDUTEBA   RLDUTEBA", pst#NL))  ' Initialize NES Objects repeat index from 0 to 1   dataPin := lookupz(index: DATA1, DATA2)   NES[index].Init(LATCH, datapin, CLK)  ' Main Loop repeat   pst.Home   repeat index from 0 to 1     buttons[index] := nes[index].buttons     pst.Position(11*index, 3)     pst.Bin(buttons[index], 8)     Pst.NewLine   waitcnt(clkfreq/10+cnt) </pre>	<pre> ' Start Parallax Serial Terminal ' Display headings  ' Loop through 2 NES objects ' Get data pin ' Get type of game pad  ' Main loop ' Cursor to top-left home position ' Loop through 2 NES objects ' Get button states from game pad  ' Display button states ' Carriage return ' 1/10 s delay </pre>

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Parallax:

32368