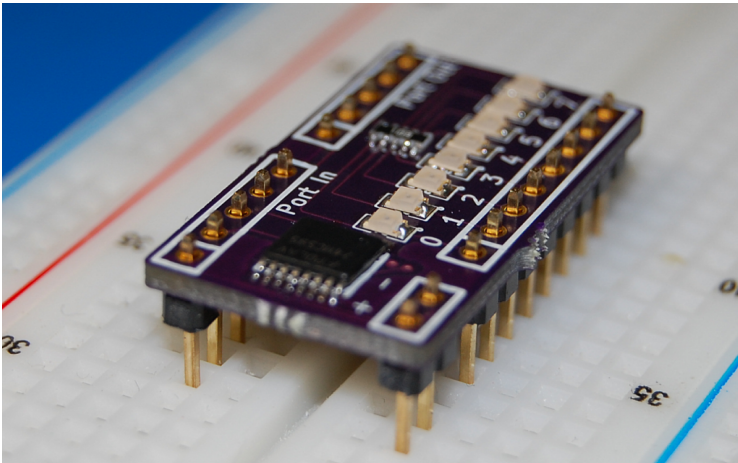


The Toymakers @ tymkrs.com
Questions? Please contact us:
feedback@tymkrs.com

DATASHEET

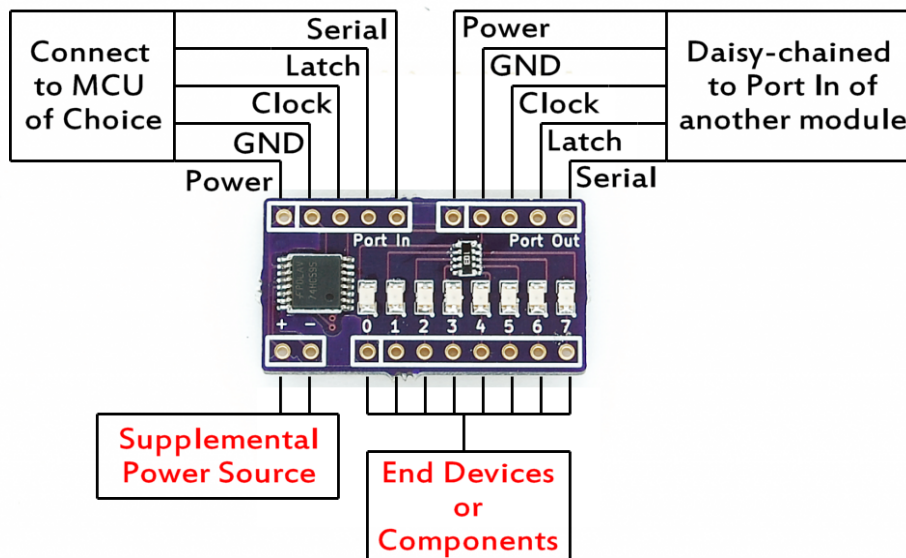


TTL8 Module

The TTL8 kit allows you to control 8 end devices/components via serial-to-parallel shift register.

- Kit Type: SMT kit with minimal throughhole soldering
- Function: Drives 8 end devices via use of serial-to-parallel shift register
- Uses 3 pins, Vcc, and GND on the MCU to send serial data to the shift register
- The module is daisy-chainable and extra power rails are available to supplement extra power needs.

KIT CONTENTS



Contents of the TTL8 Module Kit:

- TTL printed circuit board (16.61 x 29.31 x 1.60mm) – reflowed for you already
- 2 – 1x5 male headers
- 1 – 1x8 male header
- 1 – 1x2 male header

Electrical Components:

Reference	Quantity	Type	Value
D1 – D8 (optional)	8	LED, 0805, Vf = 1.8V, 20mA	Hyper red LED
-- (optional)	1	Resistor Array, 1206	240 ohm
--	1	Shift Register, 16-TSSOP	74HC595

75HC595 Shift Register Maximal Operating Conditions

Datasheet: http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf

Parameter	Maximal Ratings	Unit
Supply Voltage	-0.5 – +7.0	V
Operating Temperature	-40 to +125	°C
Output Current (Qn)	+/- 35 per pin	mA
Supply Current	70	mA

Recommended Operating Conditions

Parameter	Ratings	Unit
Supply Voltage	3.3 – 5.0	V
Ambient Temperature	25	°C

Tools and material required for assembly (not included with the kit):

- Soldering iron
- Solder

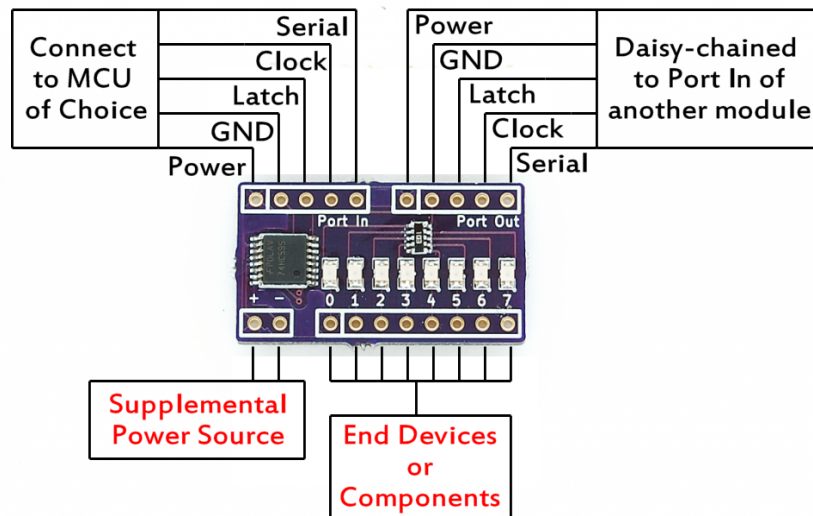
User provided items required for intended function:

- End device (such as LEDs, triggers, motors, etc)

Additional physical/electrical specifications:

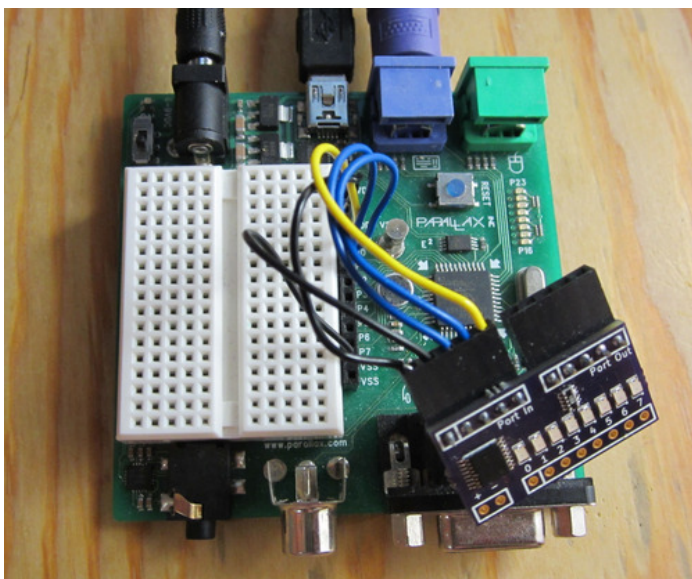
- Printed Circuit Board size: 0.65 x 1.15 x 0.063" (16.61 x 29.31 x 1.60mm)
- PCB thickness: 0.063" (1.60 mm), not including any components
- PCB thickness: 0.433" (11.0 mm), max height with headers.
- Headers are breadboard friendly.

Use Instructions



- **Method of use:** Control of the TTL8 by the microcontroller of choice requires at minimum Clock, Latch, Serial, and GND. Power and GND can come from the microcontroller or by way of the supplemental power header.
- **Requirements:** GND of the microcontroller should be connected to GND of the NPN8 module. The voltage between the microcontroller and the TTL8 must also be the same. I.e. If your microcontroller runs off of 3.3V, your board needs to be powered with 3.3V. This is due to the comparator in the shift register which determines what a 1 and what a 0 is.
- **Daisy-chaining:** Multiple TTL8s can be connected to each other. The Port Out of one TTL8 can be connected to the Port In of another TTL8. The microcontroller would still send the serial data to all chained TTL8s.
- For high current needs, please go to the NPN8.

Additional Picture:



TTL8 Module connected to the Propeller MCU

Example Code

File: TTL8_Demo_1.spin
Version: 1.0
Copyright (c) 2013 Tymkrs
See end of file for terms of use.
'This code shows the binary value of what you type out on a keyboard in LEDs
Author: Whisker

CON

```
_clkmode = xtall + pll16x
_xinfreq = 5_000_000
CLS = 16
CR = 13
Clock = 2 'Set the MCU pins that will control the NPN8 module
Latch = 1
Serial = 0
```

OBJ

```
keys: "Keyboard"
debug: "FullDuplexSerial"
```

PUB KeyboardDisplay | Index

```
keys.start(26,27)
'debug.start(31,30,0,57600)
dira[Latch] := 1 'set MCU pins as outputs
dira[Clock] := 1
dira[Serial] := 1

waitcnt(clkfreq + cnt)

'debug.str(string(CLS, "Type characters on", CR, "the PS/2 keyboard:", CR, CR))

repeat
  if keys.getKey 'This monitors the keyboard to see what key you have pressed
    dirb := keys.getKey
    debug.tx(keys.getKey)

  repeat Index from 0 to 7
    'Set the state of LED Serial Pin for this LED to the value stored in its slot of
the SwitchState array
    outa[Serial] := dirb[Index]
    'Pull the LED Clock Pin high then low to write this LED's state into the 595's
register
    outa[Clock] := 1
    outa[Clock] := 0
    'Pull the LED Latch Pin high then low to apply the contents of the 595's register
to the 595's output pins (LEDs)
    outa[Latch] := 1
    outa[Latch] := 0
```

TERMS OF USE: MIT License
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.