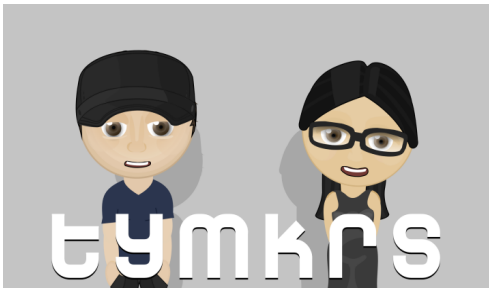
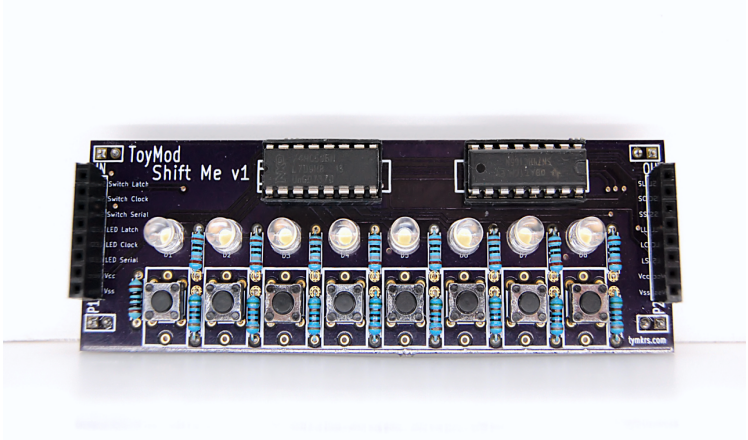


The Toymakers @ tymkrs.com
Questions? Please contact us:
feedback@tymkrs.com

DATASHEET



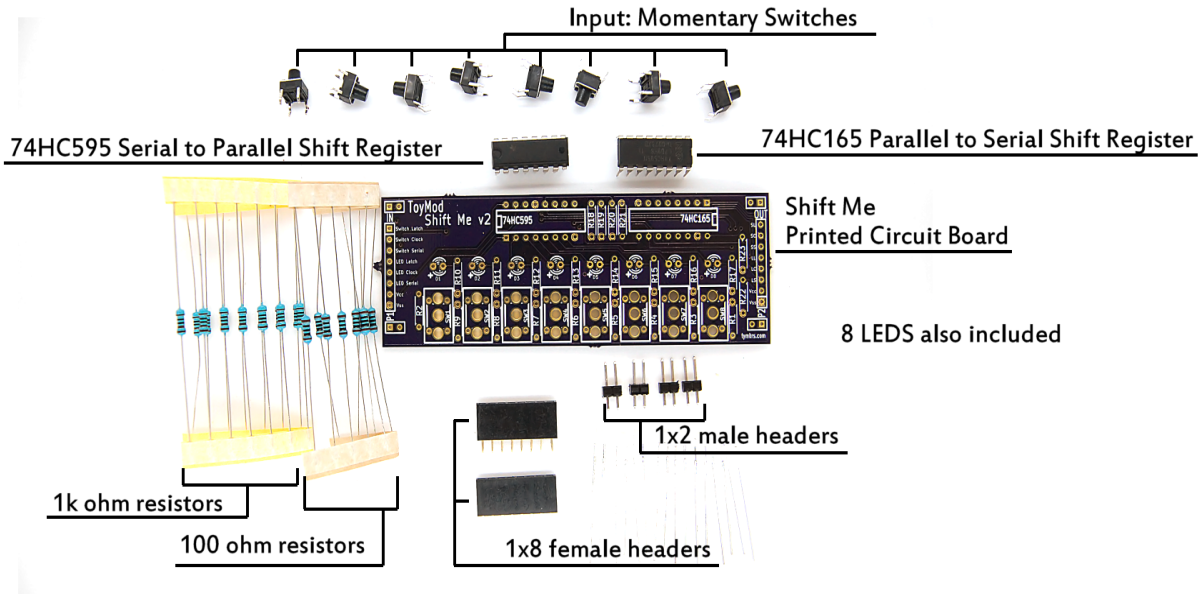
Shift Me

Shift Register based Input and Output Kit: Momentary/Toggle Switch Versions

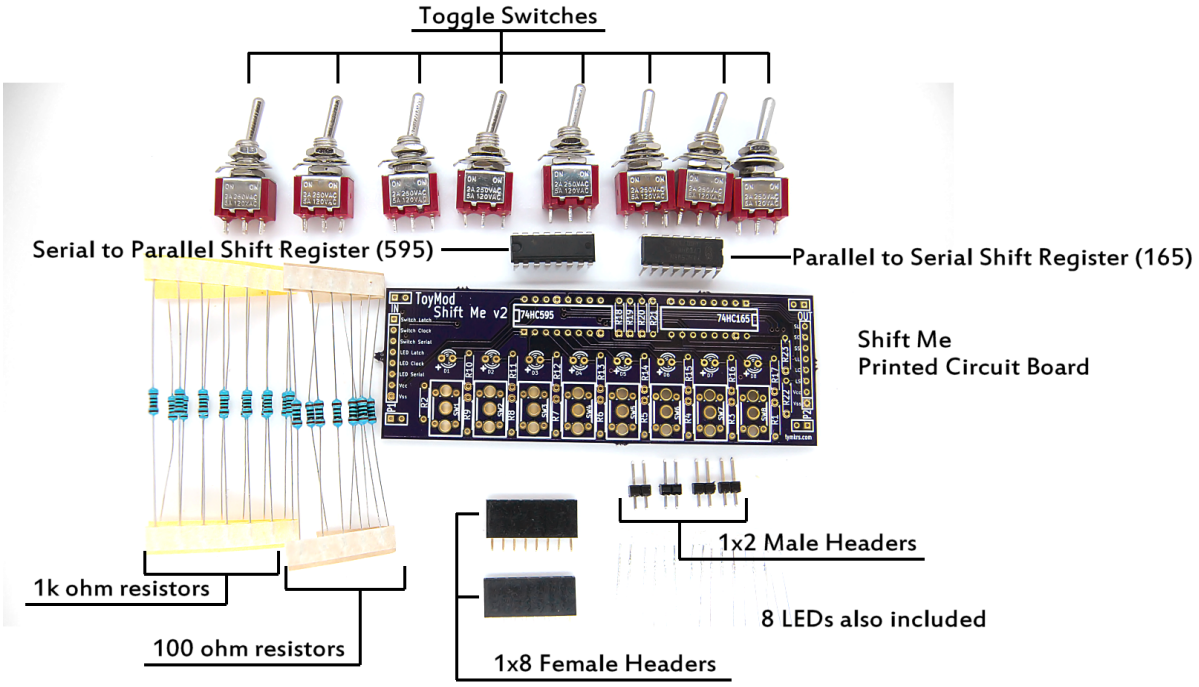
The Shift Me kit is a shift register based input and output kit controlled by only 3 wires!

- Kit Type: Through-hole soldering
- Assembly instructions: In datasheet
- Function: Input and Output Module
- This module requires only 5 wires from your microcontroller of choice to input serial data and visualize serial data output as well!

KIT CONTENTS



Toggle switches are also an option instead of Momentary Switches



Contents of the Shift Me Kit:

- Shift Me printed circuit board (101.40 x 35.38 x 1.60mm)
- 2 1x8 Female Headers
- 4 1x2 Male Headers
- Electrical Components

Electrical Components:

Reference	Quantity	Type	Value
R2, R10-R17	9	Resistor, 1/4W	100 ohm
R1, R3-R9, R18-R23	16	Resistor, 1/4W	1k ohm
SW1-SW8	8	Switch (SPST/SPDT)	Toggle/Momentary
74HC595	1	Shift Register	74HC595
74HC165	1	Shift Register	74HC165
D1 - D8	8	LED	Bright white

Electrical Characteristics:

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	ºC
Output Current (Qn)	+/- 35 per pin	mA
Supply Current	70	mA

Note: Absolute maximum ratings are stress ratings only and functional device operation is not implied. The device could be damaged beyond Absolute maximum ratings.

75HC165 Shift Register Maximal Operating Conditions
Datasheet: http://www.nxp.com/documents/data_sheet/74HC_HCT165.pdf

Parameter	Maximal Ratings	Unit
Supply Voltage	-0.5 – +7.0	V
Operating Temperature	-40 to +125	ºC
Output Current (Qn)	+/- 25	mA
Supply Current	50	mA

Note: Absolute maximum ratings are stress ratings only and functional device operation is not implied. The device could be damaged beyond Absolute maximum ratings.

Recommended Operating Conditions
Datasheet: http://www.nxp.com/documents/data_sheet/74HC_HCT165.pdf

Parameter	Maximal Ratings	Unit
Supply Voltage	2.0 – 5.0	V

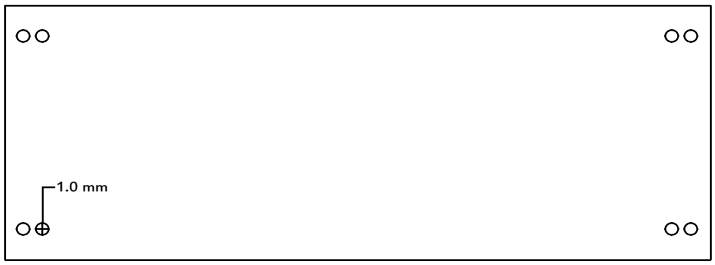
Tools and material required for assembly (not included with the kit):

- Soldering iron
- Solder
- Wire clippers

User provided items required for function:

- Microcontroller sending serial data to the shift register to control information displayed by the LEDs, or to deal with the input from the switches.

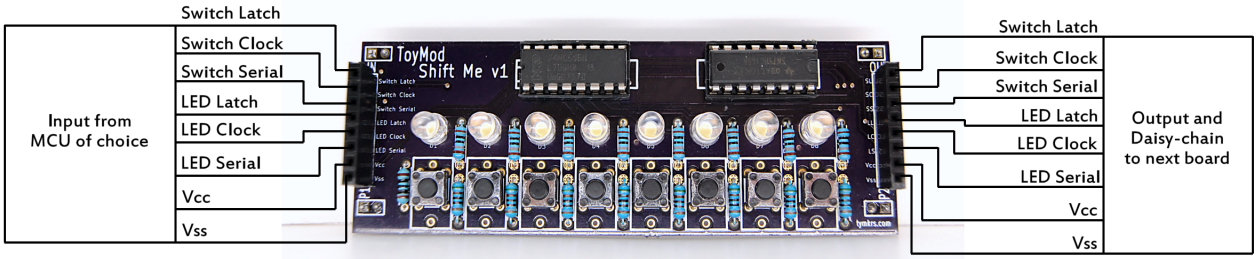
Mounting Holes:



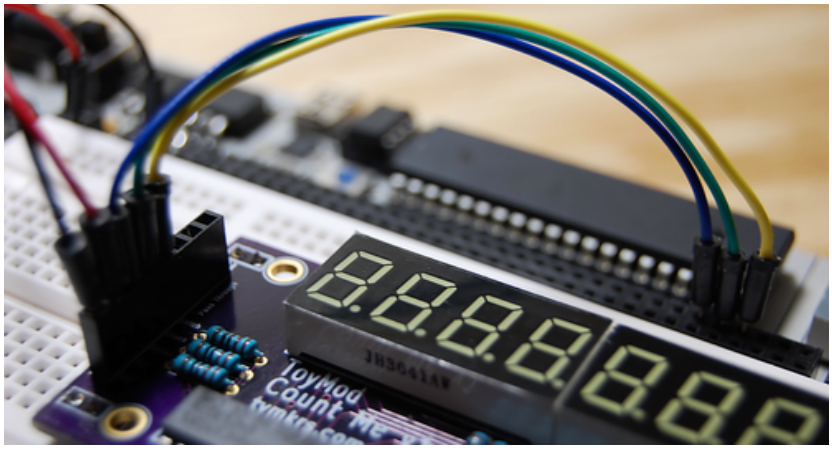
Additional physical/electrical specifications:

- Printed Circuit Board size: 3.99 x 1.39 x 0.063" (101.40 x 35.38 x 1.60mm)
- PCB thickness: 0.063" (1.60mm), not including any components
- PCB thickness: 0.472" (12mm), max height with all components
- Mounting holes: 1X2 header holes in all four corners are available and are not electrically connected to the circuit.

Board Connections:



Additional Photo:



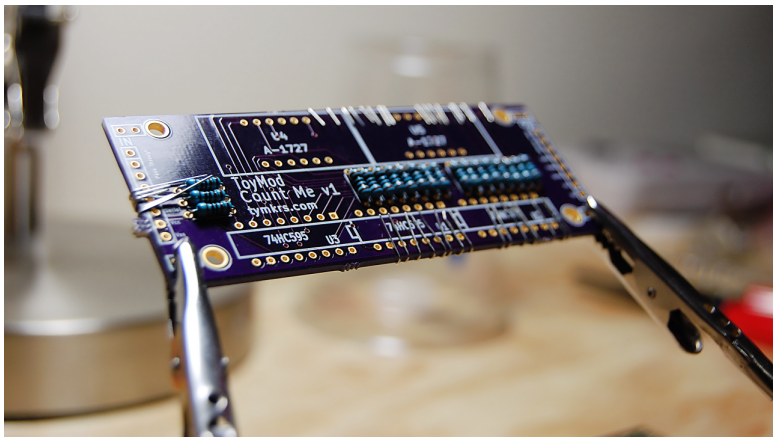
PCB on Professional Propeller Development Board

Assembly Instructions

Build Notes:

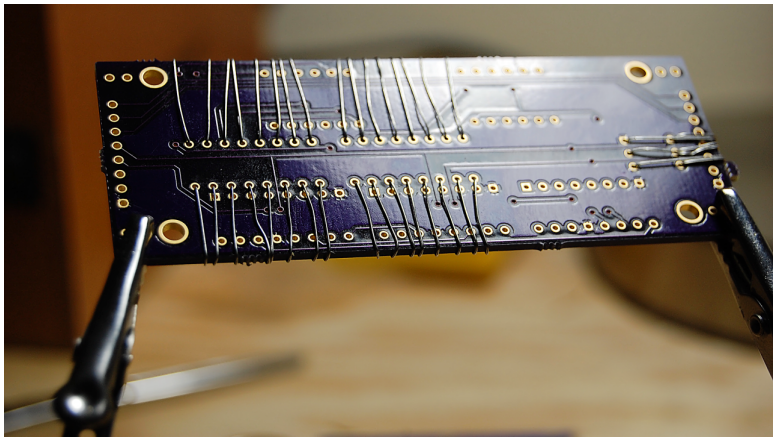
Note, the following instructions can be done in pretty much any order. I personally place all of the components on before soldering, but you are welcome to put in a component, solder it, then repeat with the rest of the components.

Step 1: Put in the components!



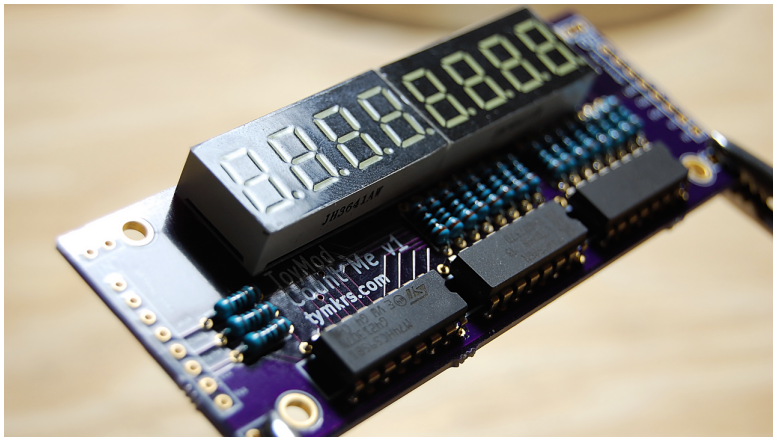
R1 - R3: 1k ohm Resistor

These 3 resistors go into R1 - R3 slots – polarity does not matter. (Brown-Black-Black-Brown-Brown)



R4 - R19: 100 ohm capacitor

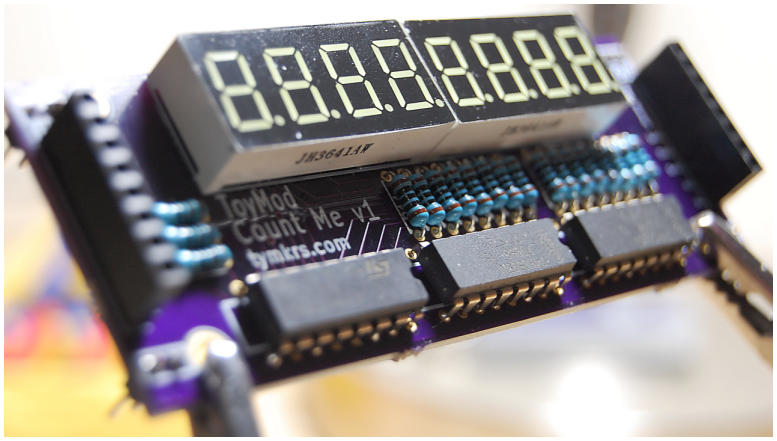
These resistors go into R4 – R19 slots – polarity does not matter (Brown – Black – Black – Black - Brown)



U1 – U3: 74HC595

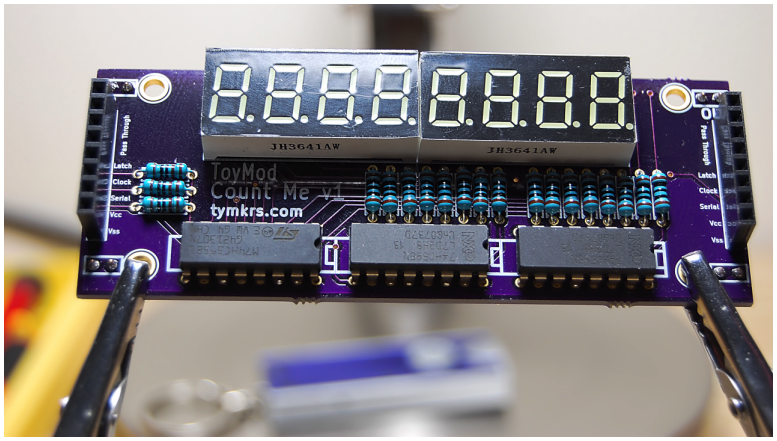
Be sure to line the divot in the chip to the graphic on the board!

DIP sockets can be soldered into the PCB so that the shift registers can be changed if necessary.



U4 - U5: 7-segment displays

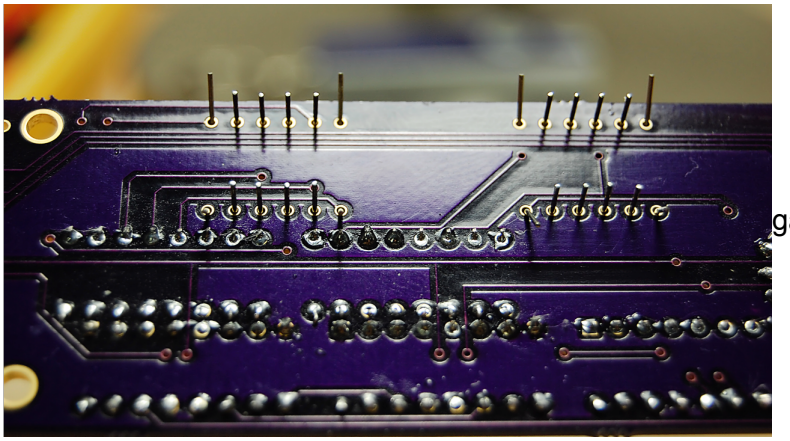
Make sure to solder these with the decimal points facing the bottom of the board.



1x8 Female Headers

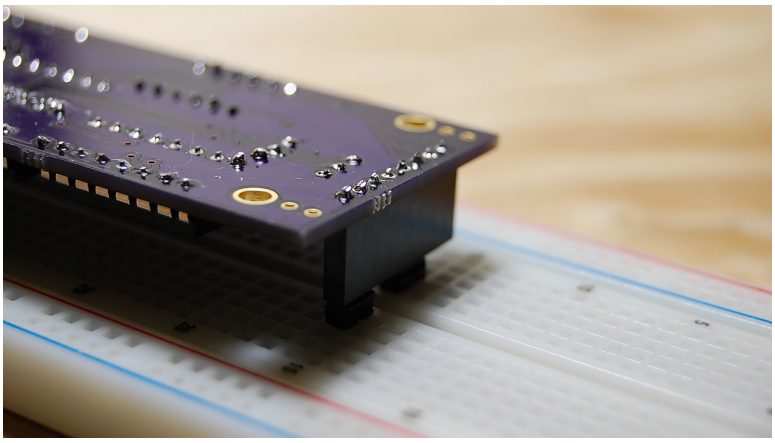
These are to be soldered on the ends! We suggest soldering them facing upwards.

Step 2: Solder the electrical components in!



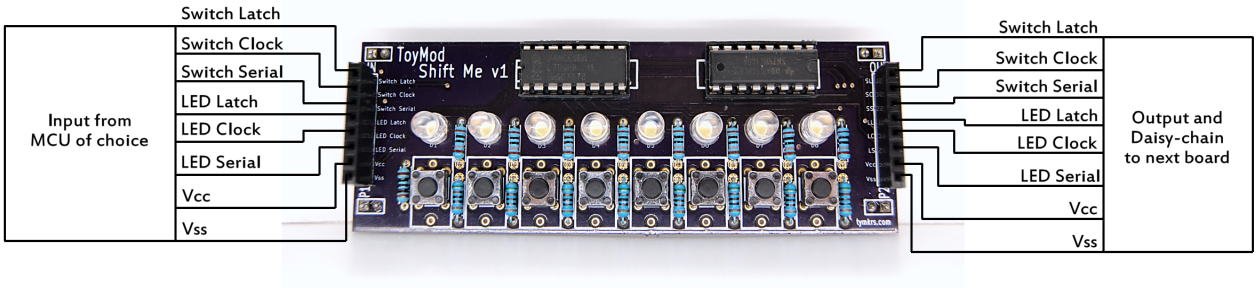
I use 60/40 0.38mm gauge solder for these pads. But also have 1.3mm gauge solder for the larger solder pads.

Step 3: Trim the extra leads off of the electrical components!



Once you're done soldering the components, it's a good idea to clean up all of the extra leads from the electrical components. We use a spare pair of nail clippers – it works quite easily!

Use Instructions



On the Input side:

- Switch Latch, Clock, and Serial are monitored by your microcontroller for inputs from the 8 different switches (momentary or toggle).
- LED Latch, Clock, and Serial are used by your microcontroller to send input for visualization to the LEDs.
- Remember to connect the GND on your microcontroller to the GND of the Shift Me

On the output side:

- This side is to daisy-chain the information sent by the microcontroller out to additional modules (such as the Shift Me or Count Me).

Helpful Links

- http://tymkrs.tumblr.com/post/53522451066/tymkrs-shift-me-chainable-shift-register-header-kit
- **Assembly Video**: http://www.youtube.com/watch?v=6FibM0PAFvY

Example Code

```
{{
┌──────────────────────────────────────────┐
File: ShiftMe_BitBang_Demo.spin
Version: 1.0
Copyright (c) 2013 Tymkrs
See end of file for terms of use.

Author: Whisker
└──────────────────────────────────────────┘
}}

{
HISTORY:
  This object is made as an example for using the ShiftMe ToyMod kit from http://tymkrs.com/

USAGE:

  • Connect ShiftMe pins Switch Latch, Switch Clock, Switch Serial, LED Latch, LED Clock, and LED Serial to
    Propeller Pins Switch_Latch_Pin, Switch_Clock_Pin, Switch_Serial_Pin, LED_Latch_Pin, LED_Clock_Pin, and LED_Serial_Pin

  • Connect ShiftMe pins Vcc and Vss to +3.3v and GND

  • Use Parallax Serial Terminal to monitor the position of the eight switches

}
Con
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  'Which Pins is ShiftMe connected to?
  Switch_Latch_Pin  = 0
  Switch_Clock_Pin  = 1
  Switch_Serial_Pin = 2
  LED_Latch_Pin     = 3
  LED_Clock_Pin     = 4
  LED_Serial_Pin    = 5

OBJ
  pst : "Parallax Serial Terminal" 'As they change, positions of the eight input switches are printed to a Parallax Serial Terminal window on your PC

Var
  long  SwitchState[8] 'The positions of the eight switches are stored as 1s and 0s in this array

Pub Main | Index

  'Set these ShiftMe pins to outputs
  dira[Switch_Latch_Pin]  := 1
  dira[Switch_Clock_Pin]  := 1
  dira[LED_Latch_Pin]     := 1
  dira[LED_Clock_Pin]     := 1
  dira[LED_Serial_Pin]    := 1

  'Set these ShiftMe pins to inputs
  dira[Switch_Serial_Pin] := 0

  'Start Parallax Serial Terminal
  pst.Start(115_200)

  'Wait 1 second for user to enable Parallax Serial Terminal software
  waitcnt((clkfreq) + cnt)

  'Print greeting and initial values to serial terminal
  pst.str(String("Demonstation is running! Shift Me!"))
  pst.char(13)

  'Set the initial state of the Switch Latch Pin to High
```

```
  outa[Switch_Latch_Pin] := 1

  repeat
    'Pull the Switch Latch State low then high to store the positions of the 8 switches into 165's
register
    outa[Switch_Latch_Pin] := 0
    outa[Switch_Latch_Pin] := 1

    'Read the eight switch positions from the 165 shift register into the SwitchState array one at a
time
    repeat Index from 0 to 7
      SwitchState[Index] := ina[Switch_Serial_Pin]
      'Pull the Switch Clock Pin high then low to read the next register value onto the Switch
Serial Pin
      outa[Switch_Clock_Pin] := 1
      outa[Switch_Clock_Pin] := 0

    'Print the current states of the 8 switches to Parallax Serial Terminal running on the host PC
    repeat Index from 0 to 7
      pst.dec(SwitchState[Index])
      pst.char(32)
    pst.char(13)

    'Write the eight states stored in SwitchState out to the 595 shift register (LEDs)
    repeat Index from 7 to 0
      'Set the state of LED Serial Pin for this LED to the value stored in its slot of the
SwitchState array
      outa[LED_Serial_Pin] := SwitchState[Index]
      'Pull the LED Clock Pin high then low to write this LED's state into the 595's register
      outa[LED_Clock_Pin] := 1
      outa[LED_Clock_Pin] := 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[LED_Latch_Pin] := 1
    outa[LED_Latch_Pin] := 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.

}}