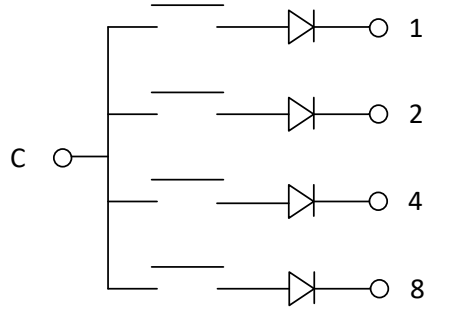
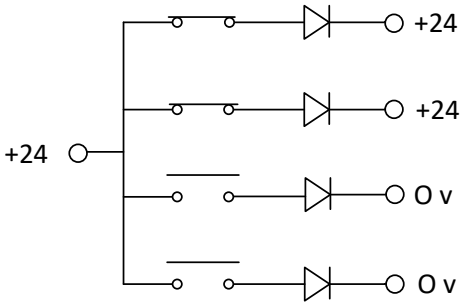
Chapter 9 Thumb Wheel Switch

Thumbwheels have been available for many years to enter numbers into computers. The ones below are capable of entering a single digit or more. They may be ganged into groups of 4 or 6 to gather a 4 or 6 digit number.



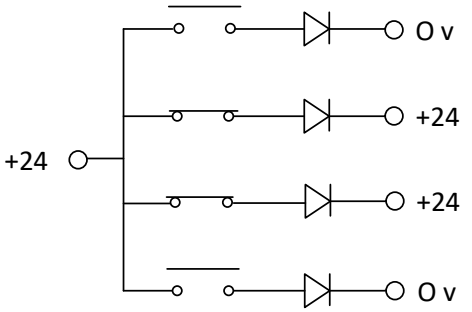
Display =
0000 = 0

The Thumb-Wheel Switch pictured at left shows a thumb-wheel switch as used in industry. The tws has been used in factories to enter numbers into computers and plc's. The switches show the number on the face and give the BCD bits shorted in the switch as the number is entered.



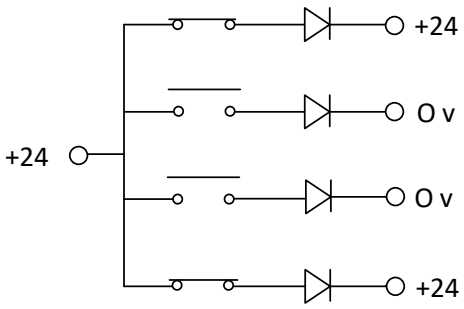
Display =
0011 = 3

For instance, if the number displayed is '3', the bits associated with the 1 bit and the 2 bit are shorted giving 24 V on these pins with 0 v on the 4 and 8 bit pins.



Display =
0110 = 6

For instance, if the number displayed is '6', the bits associated with the 2 bit and the 4 bit are shorted giving 24 V on these pins with 0 v on the 1 and 8 bit pins.

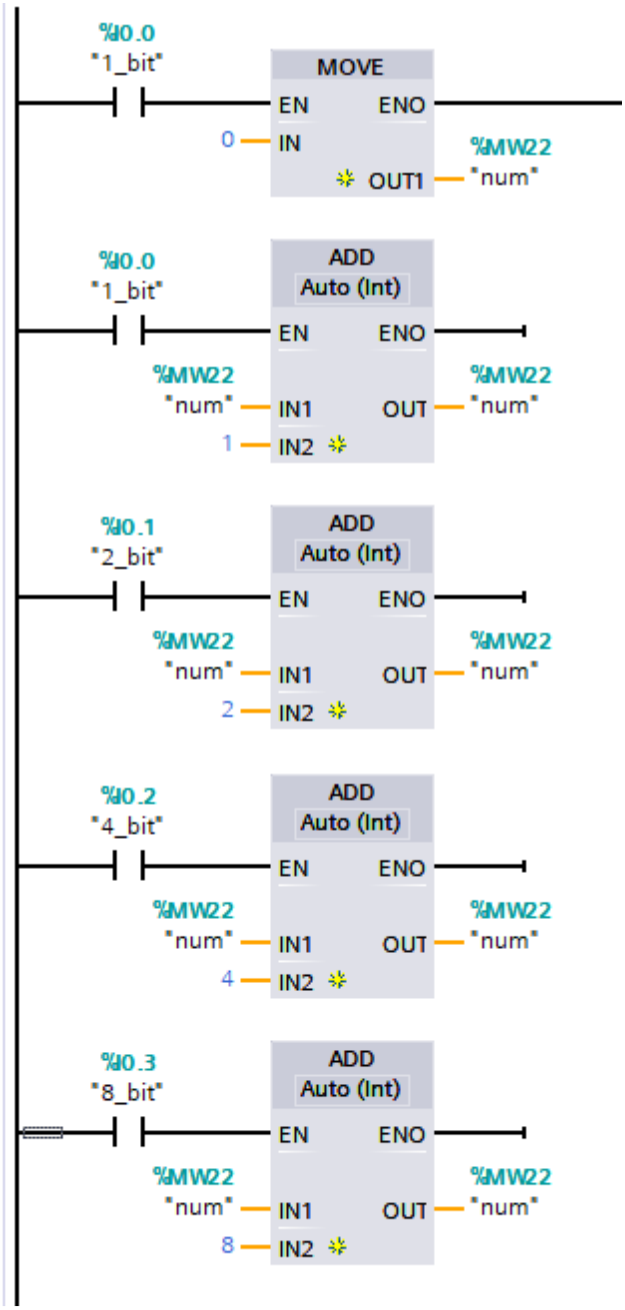


Display =
1001 = 9

Likewise, if the number displayed is '9', the bits associated with the 1 bit and the 8 bit are shorted giving 24 V on these pins with 0 v on the 2 and 4 bit pins.

Numbers input in this way are BCD or Binary Coded Decimal as opposed to integer. PLCs in early models had instructions for conversion a BCD number to an integer number. There is not as much need for this as Thumbwheel Switches (TWSs) have for the most part gone away with the development of HMI (Human Machine Interface) equipment.

For the BCD (Binary Coded Decimal) number above, if the four inputs were connected to the Siemens processor's inputs I0.0, I0.1, I0.2 and I0.3 and the word representing the tws number was stored in the variable 'num', the following program would be generated to move this single digit into the integer number 'num'. This program can be used to move a single BCD digit into an integer variable 'num'.

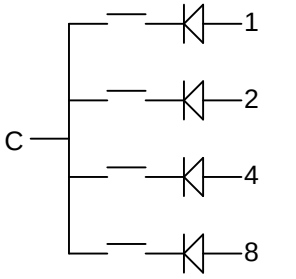


This lab requires the student to use the thumbwheel switches to enter a number from 0 to 9999 into an internal memory location (integer). The number is the one represented on the TWSs and can be changed. The internal memory location should be updated at least each half second.

With the tws above, we notice the diodes pointing to the right. This is the direction of positive current flow. We notice in the figures below the tws with diodes shown in the reverse direction. This is due to the idea that tws can be constructed with the diodes pointing either way. If pointing to the right, the wiring is for true high logic. If to the left, the wiring is for true low logic. However, one can still read the tws in the true low switch using true high logic if the bits are scanned one bit at a time through each tws. This is what is required in this lab assignment. It is a more difficult task and requires a timing circuit but still is possible. The diodes were originally put in so that more than one group of 4 or 6 digit tws could be read by a single digital input card. With this lab, the diodes were backwards for true high logic. The timing program still allows a programmer to read the number from the tws.

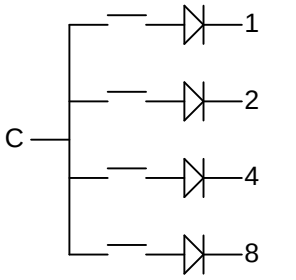
Allow only 4 inputs and 4 outputs to the PLC from the TWS's.

Our TWS's:



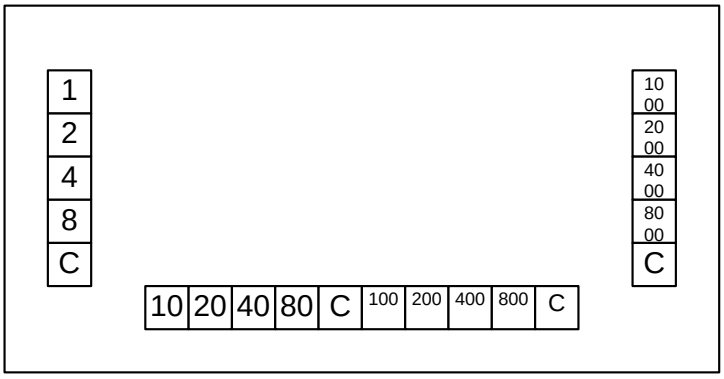
TWS with diodes installed

To work correctly, the diodes should be reversed as follows: (do not do this, however)



TWS with diodes reversed

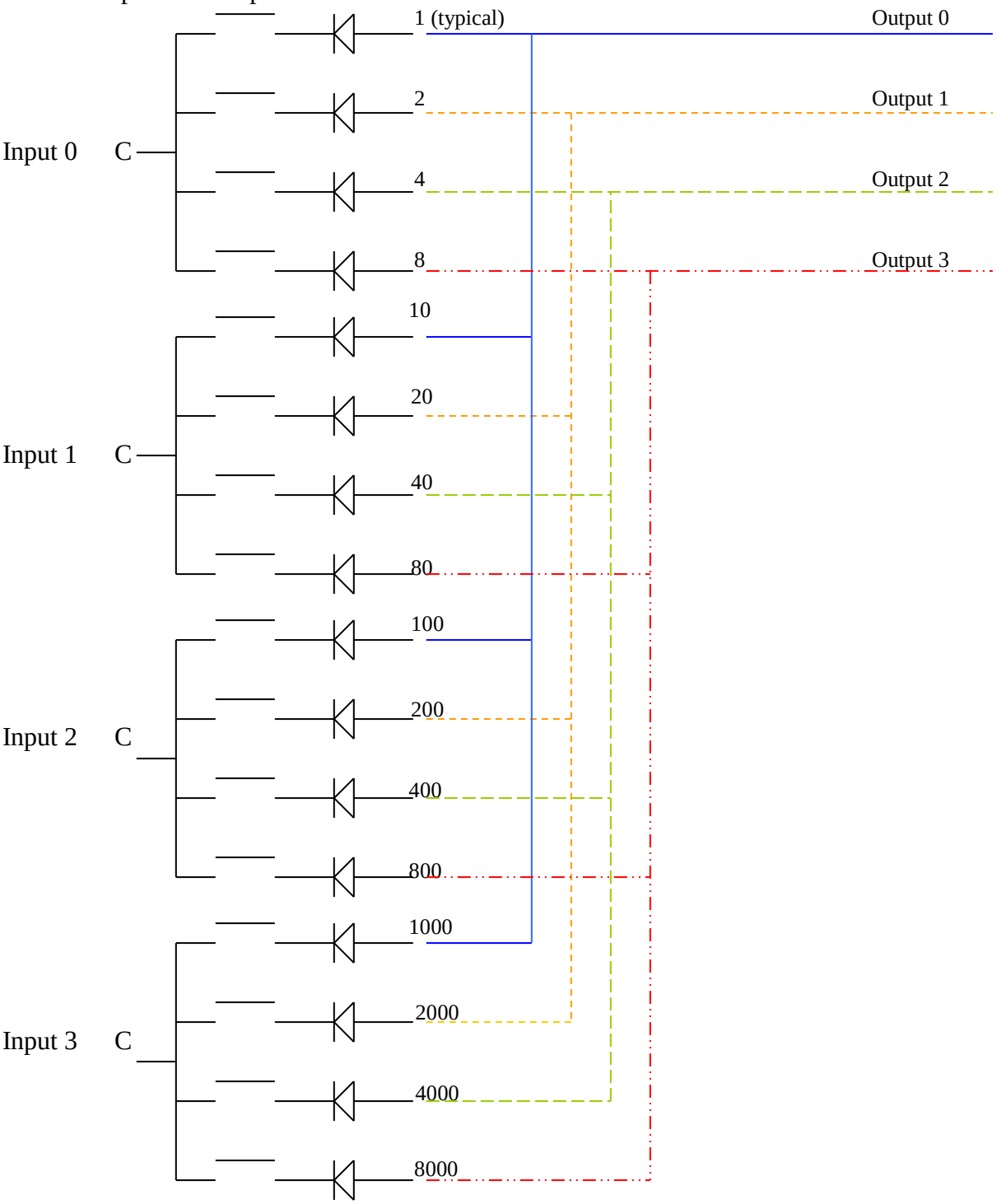
Wiring on the back of the TWS's is as follows:



Wiring Diagram Layout of Thumb Wheel Switches

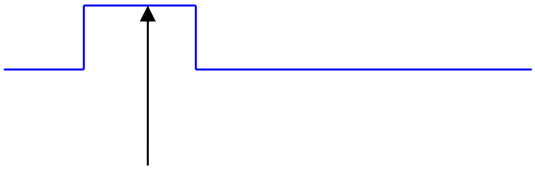
Note: Lab requires that code is executed too fast to be viewed on the monitor. Timing diagrams will have to be drawn to successfully complete this lab.

To complete the lab, set up a timing diagram similar to the traffic intersection but with very short time durations. Hint: When reading from outputs through the tws, tie the 1, 10, 100, and 1000 terminals together and read the result at an input. Four inputs are received, one from each common (C).
Wire the inputs and outputs as follows:



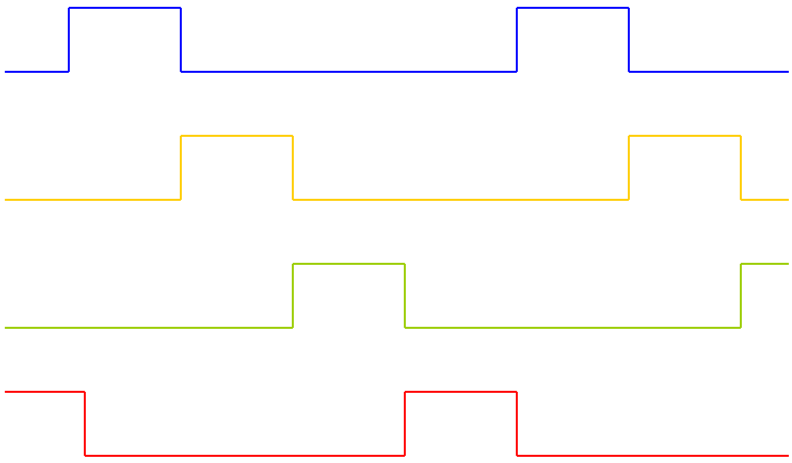
Note that care must be taken to not read inputs 1-4 immediately since the filter time for reading an input is not as quick as the time of a scan. A time delay must be introduced to read the input while the output is on.

Output 0



Read inputs 1-4 sometime after the filter time that is set for the processor. After Output 0 turns on and the inputs are read, output 0 can be turned off and the next input read. It is a good idea to read the inputs using a one-shot.

Repeat the timing chart so that the thumb-wheel switches can be read in as quickly as possible. Use a timing diagram similar to the one used for the traffic intersection but with much quicker times as follows:



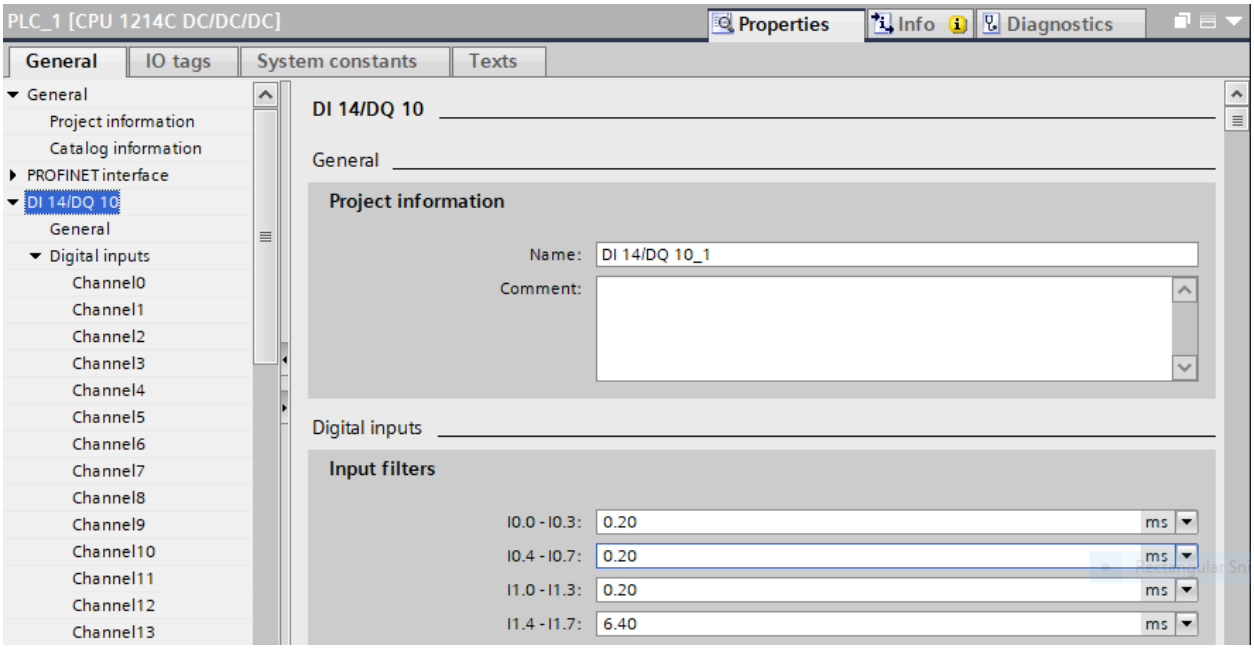
Timing Diagram of Repeating Outputs

Note: While these output waveforms move quickly, they are not one-shots and need to have time duration of 20-50 msec. Their duration must be long enough for the input signal to sufficiently settle and be longer than the filter time set for the input type.

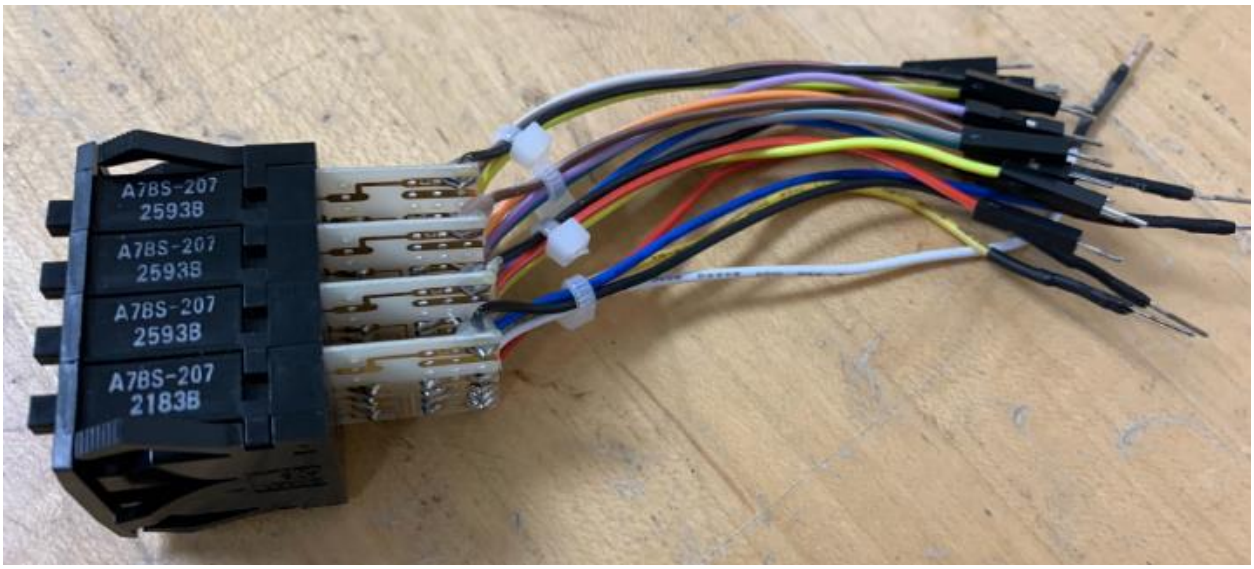
An outline for the program for reading the 4 digit thumbwheel switch:

1. Clear a holding location for the completed number
2. Turn on Output 0
3. Delay a short time period (20 or 30 ms)
4. Read the 4 inputs. If input 0 =1 then add 1 to the holding location
   If input 1 = 1 then add 10 to the holding location
   If input 2 = 1 then add 100 to the holding location
   If input 3 = 1 then add 1000 to the holding location
5. Delay slightly and turn off output 0
6. Turn on output 1
7. Delay a short time period (20 to 30 ms)
8. Read the 4 inputs. If input 0 = 1 then add 2 to the holding location
   If input 1 = 1 then add 20 to the holding location
   If input 2 = 1 then add 200 to the holding location
   If input 3 = 1 then add 2000 to the holding location
9. Delay slightly and turn off output 1
10. Repeat for output 2
11. Repeat for output 3
12. Move the contents of the holding location to a second location for display in N7.

How quickly can an input be read? There is an input filter on these inputs by group. Input filters are shown in the figure below. The following shows the input I0.0-I0.3 changed to 0.20 msec to see a change:



What these TWS's look like? The following figure shows a gang of 4 tws with diodes soldered in and jumpers provided to attach to a breadboard.

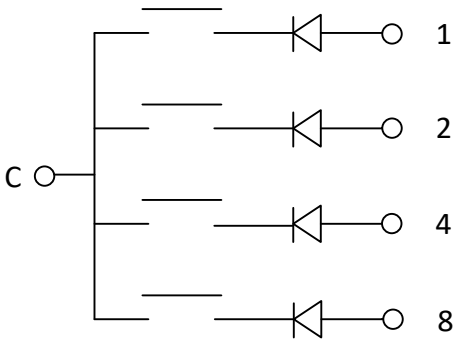


The Digi-Key Part Number for the above individual tws is given here. These connect together and are held in place per instructions provided. An end plate may also be ordered.

A7BS-207-1 Datasheet

Digi-Key Part Number	SW268-ND
Manufacturer	Omron Electronics Inc-EMC Div
Manufacturer Part Number	A7BS-207-1
Description	SWITCH THUMB BCD 0.1A 50V/28V
Manufacturer Standard Lead Time	8 Weeks
Detailed Description	Thumbwheel Switch BCD 0.1A @ 50VAC/28VDC Panel Mount, Snap-In

One can also build tws circuits by using a breadboard and jumpers in addition to diodes. LEDs may be used for the diodes if desired. The circuit below may be used. It does not have the ability to block number entry above 9 so the student take care to only allow jumpers that input numbers from 0 to 9 or allow the program to write full hexadecimal numbers to the integer variable.



You may build the thumbwheel switch manually on a breadboard with the circuit at left

Display of the number in the PLC may be either using the watch table of Siemens or the Data Table of Allen-Bradley. Be sure to set the input filters of the inputs to something less than the timing diagram rate for gathering the number information.

If a tws is not available and the circuit above is used, you may display a single digit or multiple digits. If multiple digits are displayed, then the number will range from 0 to 9999 (for 4 digits).