AND8264/D

Implement the Single Wire Protocol

Prepared by: Michael Bairanzade
ON Semiconductor

ON Semiconductor®

http://onsemi.com

APPLICATION NOTE

Abstract: the Single Wire concept has been developed to minimize the number of I/O, at MCU level, to control peripheral devices. Basic applications can be White LED driver, audio sound level dynamic adjust or extra functions, remotely controlled DC/DC converter, etc…

This Application Note depicts the Single Wire operation developed by ON Semiconductor, and provides an example (both hardware and software) to control the NCP5612 device.

BASIC CONCEPT

The remote control of a peripheral device can be achieved by either a parallel or a serial transfer of data between the MCU port and the final device. The parallel system is rarely used, except for the IEEE−488 / GPIB system, and most of the industrial applications use one of the existing protocol: SPI or the I2C, leaving aside the automotive dedicated LAN, CAN, etc., structures.

The SPI is very powerful with the capability to simultaneously Transmit and Receive data from the peripheral, but up to four I/O are necessary to handle the communication. However, the data can be transmitted at high speed, and 20 MHz or more operation frequency are achievable. Since the SPI is disconnected when /CS = High, it is easy to get a very low idle current, a key point for the portable applications.

The I2C needs two open drain lines only (one SCL clock and one SDA data), but it is speed limited and Read / Write of data shall be done on different cycles, thus reducing the data throughput. The bi−directional SDA line fulfill two functions:

- send the physical address of one peripheral among the bus
- send / receive the data from the controller to / from the peripheral.

The number of peripheral sharing a common bus depends upon the available addresses (one should avoid data collision) and the acceptable idle current due to the pull−up resistors. The other consideration with the I2C is that since it is a shared bus, there can be congestion or delays when multiple events need to be controlled on the bus.

In order to minimize the number of I/O requested, at MCU level, to achieve data exchange, a simple concept has been developed, the aim being to make the system flexible and easy to implement in existing software. Such a concept takes advantage of the relative low bandwidth required to set up a parameter in a multitask system: as an example, it is not necessary to run a high speed clock to dim the backlight of a cellular phone display!

The proposed S−WIRE uses a pulse count technique already existing in the data exchange systems. The protocol support broken transmission, assuming the hold time is shorter than the maximum 300 μs specified in the data sheet. To simplify the description, let us consider the Control Signal be named CNTL during the rest of this application note.

Based on the example provided in Figure 1, the CNTL pin supports two digital levels:

CNTL = Low → the system is shut−off and no further action takes place in the peripheral.

CNTL = High → the system is enable, the peripheral is active, according to the appropriate specifications.

Leaving aside the static logic levels (V_{IH} and V_{IL}), the S−WIRE counts the positive going pulses, the data sheet of the specific IC providing the truth table to control the implemented parameters.

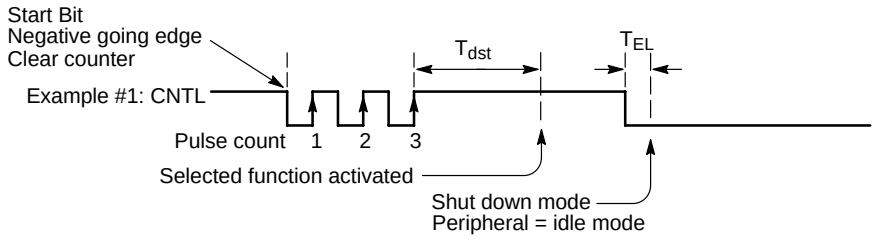

Figure 1. Basic SWIRE Timing

The T_{on} and T_{off} times are not critical and bounded solely by the need to identify the on going operation:

T_{off}: no maximum time limit, with a 200 ns minimum: the peripheral is in the idle mode as long as the signal is Low. All the internal logic circuits are cleared as soon as CNTL is detected Low. The chip is immune to any interrupt generated at system level since all the internal functions are de−activated.

T_{on}: we shall consider two situations:

1 – A transaction is on going → the MCU send data to the peripheral: in this case, the T_{on} shall be high for a minimum of 3 μs with a maximum period of 75 μs between two positive edges. That means, neglecting the T_{off}, the lowest clock is 1/75e−6 = 13 kHz, the maximum clock being 400 kHz.

2 – the T_{on} must be continuously high for 200 μs to transfer the data word (number of counted pulses) into the final stage(s) of the peripheral. Moreover, the signal must be High to maintain the LED active: this is similar to the ENABLE signal used with simple control IC.

The key point is the capability for the chip to sustain interrupts while a programming sequence is on going. As a matter of fact, since no data transfer will occur until the 300 μs delay is completed, the MCU can serve any IRQ during this amount of time, then resume to the SWIRE protocol as depicted in Figure 2.

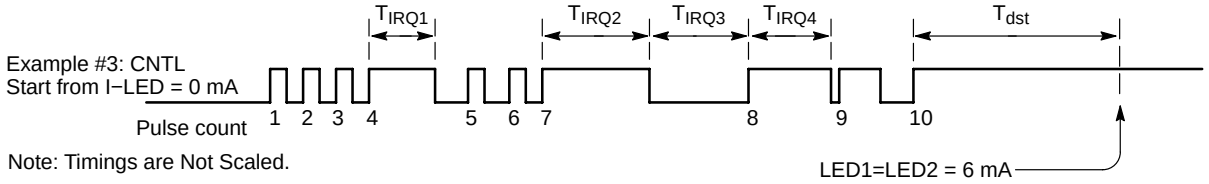

Figure 2. Typical IRQ Operation

Of course, when applicable, masking the interrupt during the total transmission time is a better solution since it avoid blanking the display for a relative long period. Assuming a typical 5 μs per pulse to drive the NCP5612, the maximum I−LED can be setup in 100 μs, a period acceptable for the IRQ masking.

The concept gives a possible infinite number of pulses, but physical limits bound the applications to realistic value: a maximum of 50 pulses looks more appropriate. On the other hand, it is possible to sub−encode the pulse count to access different registers into the same piece of silicon: such a function is embedded into the NCP5612 and will be depicted in the next paragraphs.

SWIRE APPLICATION

Let us consider the NCP5612 to illustrate the behavior of the SWIRE in real application. Based on the data sheet of this product, two internal registers shall be independently programmed:

1. – the back light control, with two LED activated, supporting a large dimming span with a quasi logarithmic slope
2. – the ICON control, with one LED activated only, handling four current steps

The pulse count has been split into two sub carriers:

A. the first four pulses address the ICON register only

B. following the fourth pulse, the next sixteen pulses address the Back Light register only.

The Back Light is de−activated when the ICON mode is selected and the ICON is de−activated when the Back Light is operating. The theoretical timings given Figure 3 illustrate the NCP5612 behavior: the third pulse is transferred to the ICON register when the T_{on} timing is completed, the chip returns to the idle mode when CNTL = Low, the two LED being switched Off.

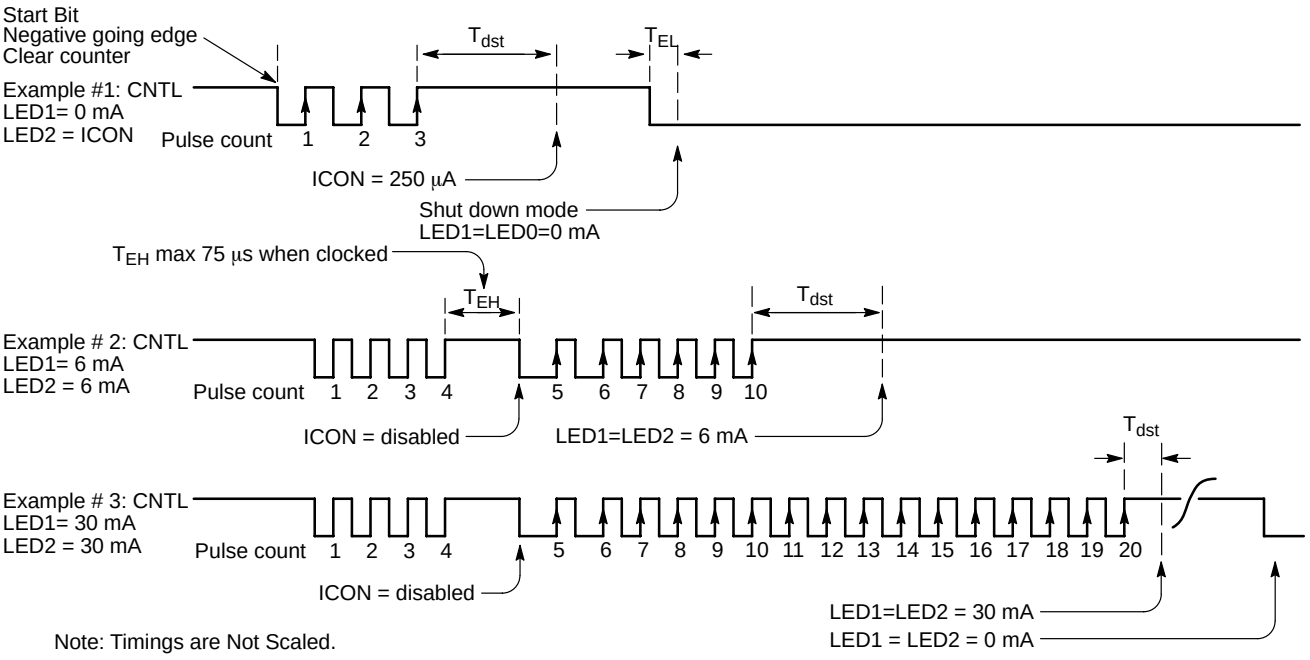

Figure 3. Typical NCP5612 SWIRE Operation

The rise and fall time – tr & tf − of the CNTL signal do not need accurate timing, assuming they are neither faster than 5 ns, or slower than 10 μs. As a matter of fact, very slow slopes are prone to induced noise which might result in unexpected behavior of the NCP5612 might yield non predictable lighting. Similarly, very fast edge rates may interact with the integrated ESD structures, and generate uncontrolled operations.

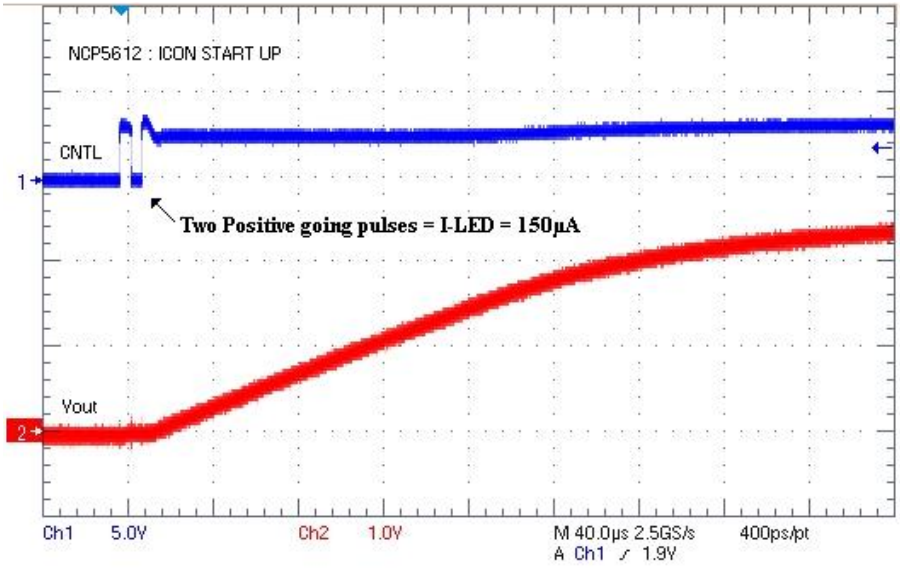

Figure 4. NCP5612 Control: ICON Active

The waveforms, given Figure 4, captured with the NCP5612 demo board, illustrate a situation to activate the ICON after the second pulse: the output voltage rises up to the Vf of the LED (typically 2.6 V @ If = 150 μA), the LED current being activated when the delay following the last positive going pulse is completed.

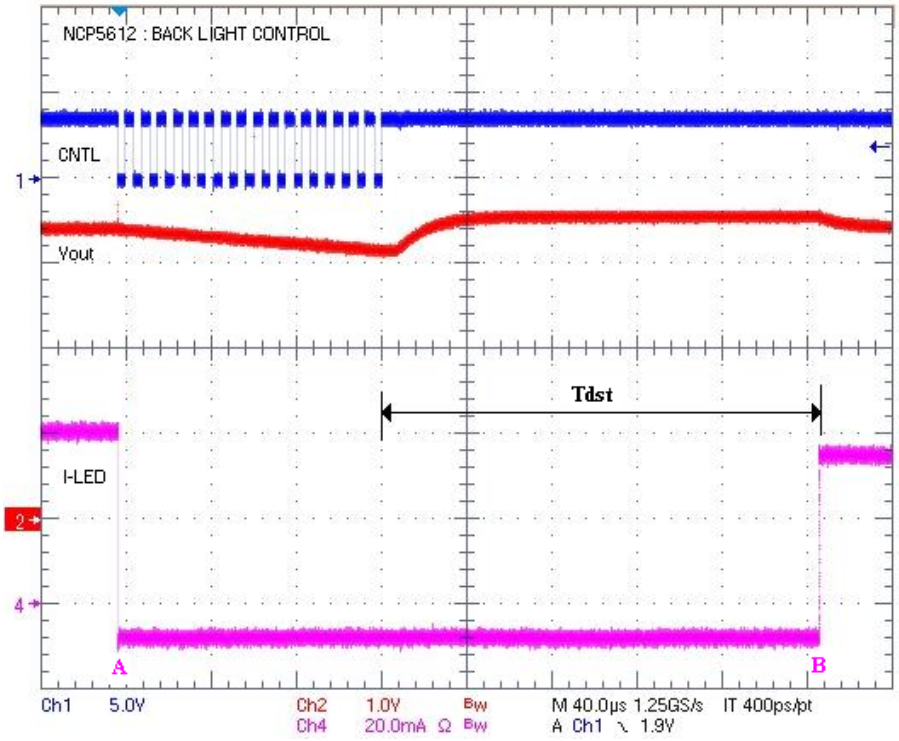

Figure 5. NCP5612 Control: Back Light Active

As depicted in the waveform given Figure 5, the I−LED is switched OFF as soon as the CNTL signal drops to zero (see point A) and jumps to the appropriate value when the T_{dst} delay (200 µs typical) is completed (see point B). Assuming the total time frame is below 1 ms, this transition is not visible since the human eye cannot detect such short pulses. However, if the CNTL signal is very slow, the display could be OFF during a long amount of time, which would be visible to the end user.

The V_{out} voltage drops to zero as the converter is switched OFF for every Low level at the CNTL pin, and returns to the normal operation level when CNTL = High.

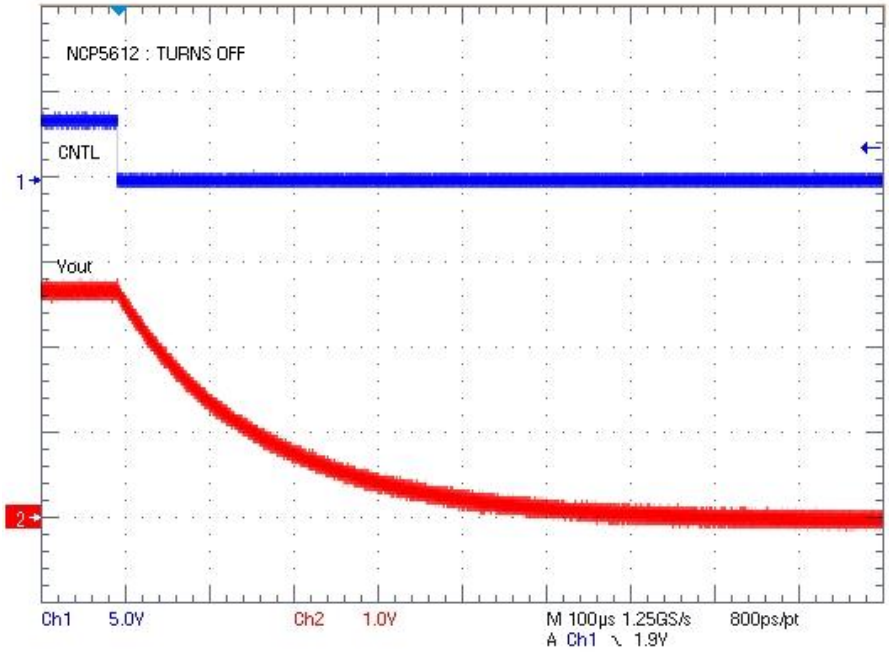

Figure 6. Typical Output Turn OFF Slope

The built−in active pull down resistor reduces the turn OFF slope to a typical 300 μs when C_{out} = 1 μF. On the other hand, all the internal structure associated to the V_{out} pin is disconnected from the load, except the ESD protection structure.

SOFTWARE

The S−WIRE is extremely simple to implement since only a small block of software code is required to implement the protocol. The example provided here below is based on an 8 bits MCU, using the Freescale assembler to generate the code. The routine shall be embedded into the main program with the appropriate labels and interrupts necessary to support the full system. Of course, using a high level tool (C or Pascal) is possible but not mandatory.

```
; MCU registers:
;                IX : not affected
;                IH : not affected
;                SP : not affected
;                ACCA : previous byte lost, hold $00 upon return
;
;Interrupts : not affected. However, no interrupt shall stop the sub routine for more than 300 μs : see NCP5612
;data sheet

gCNTL          rmb   1                  ;reserve one byte to store S−WIRE data
;
SendSwire:                              ;sub routine called by the main program
               lda   gCNTL              ;reload ACCA with S−WIRE number of pulses to send
               bne   setCNTLZ           ;force CNTL I/O to Low if ACCA = $00, exit sub routine
loopSW         bset  bCNTL,PTBD         ;I/O @ PORTB = H → send one positive going slope
               jsr   delay5             ;run a 5 μs delay to support low speed peripheral
               deca                     ;decrement the number of pulses
               cmpa  #$00               ;is it the last pulse?
               beq   exitSW             ;if yes, all pulses have been send, exit the sub routine
               bclr  bCNTL,PTBD         ;I/O @ PORTB = L → return the CNTL to Low
               jsr   delay5
               bra   loopSW             ;loop to send the next pulse
setCNTLZ       bclr  bCNTL,PTBD         ;I/O @ PORTB = L
exitSW         rts
;
delay 5:
               nop
               rts
;end of sub routines
```

With twelve instructions only, the routine occupies a very small area of memory and can be reduced to ten instructions if the peripheral supports the maximum 400 kHz clock frequency. In this case, the delay5 sub routine is no longer necessary.

The MC68HC9S08QG8 microcontroller, with built−in 20 MHz clock, is used to control the NCP5612 demo board. Associated with a dedicated Windows software, the demo board can be controlled by the standard Serial port of a PC machine.

The waveforms given Figure 5 illustrate the behavior of the software routine given here above: the 17 pulses have been send to the CNTL signal is 120 μs, followed by the 200 μs delay built in the NCP5612 chip prior to store the new current value into the LED driver.

With the addition of a small block of software in the micro−controller, the S−Wire protocol transforms a single line from the GPIO port of the micro−controller into a programmable interface to not only turn on and off a peripheral, but transfer data to change states or facility programming of different values, be they voltage or current, or time.

CONCLUSION

The Single Wire protocol, developed by ON Semiconductor, is capable to handle complex analog functions associated to the portable systems. The concept is robust, with the capability to support long interrupt during a data transfer, and support wide operating frequency. The software is simple and does not need special routines to generate the pulses specified by the SWIRE protocol. The built−in logic prevent the chip for wrong operation when out of range pulse count are send on the CNTL lie: in this case, the I−LED current is limited to the 30 mA define for each LED.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
P.O. Box 5163, Denver, Colorado 80217 USA
Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800−282−9855 Toll Free USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910
Japan Customer Focus Center
Phone: 81−3−5773−3850

ON Semiconductor Website: **www.onsemi.com**

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative