

Hex-Switch Decoder Uses Weighted-Capacitor Network to Reduce I/O Pin Count

By using this simple capacitor-network circuit and software, designs using the Arduino Uno need only two I/O pins to read a hex-switch decoder, leaving the remaining pins for other functions.

A pulse-width-modulated (PWM) signal and the RC-charging characteristics of a weighted-capacitor network can be used along with a CMOS Schmitt inverter to generate a pulse whose width is linearly proportional to the 16 combinations of a hex (thumbwheel) switch. The decoding is implemented using the Arduino Uno and requires only two pins (OCRA and INT0), leaving all of the other pins free for alternate functions.

The theory, which is the basis for this design idea, is the well-known capacitor-charging equation:

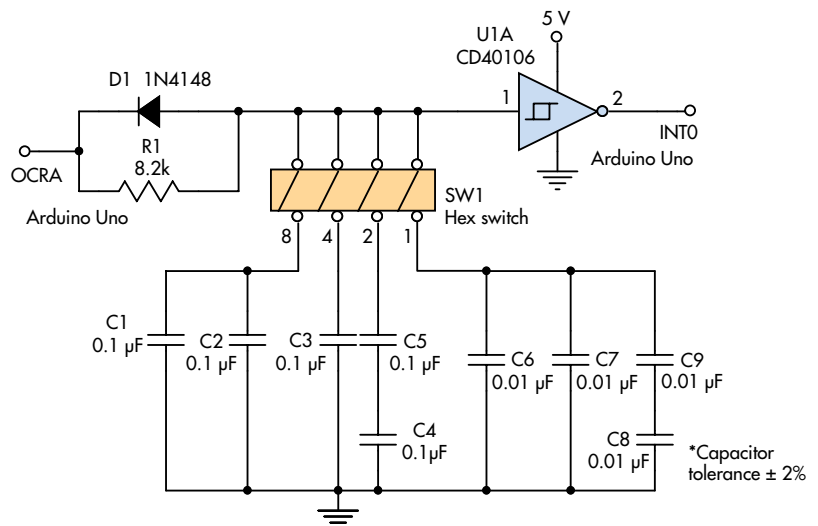
$$Vt = V(1 - e^{-t/RC})$$

Time T required to reach a voltage $k \times V$ is given by:

$$T = RC \ln[1/(1 - k)]$$

which shows a linear relationship with C for a fixed value of R .

PWM output OCRA of the Arduino Uno forms the input to the circuit (Fig. 1), with the output connected to the interrupt pin INT0. Capacitors in the ratio of 8:4:2:1 are selected by SW1, then charged through R1 during the high-state portion of the PWM signal and discharged through diode D1 on the



1. The functional block diagram consists of a hex switch connected to the weighted-capacitor network, charging resistor R1, discharge diode D1, and the Schmitt inverter.

low-state. At the start of the charge cycle, the output of the Schmitt inverter is high; it changes to low when the capacitor voltage reaches the upper threshold of the Schmitt inverter. This negative-going edge is used as an interrupt to the processor.

The Arduino Uno, operating with a 16-MHz clock, is programmed to use Timer0 to set up the PWM output OCRA at 244 Hz with the pulse-width set to 224/255 cycles. Figure 2 shows the waveforms when only switch #8 of the SW1 is ON. Parallel capacitors C1 and C2 are charged by the high-state

output V-high of OCRA, and the Schmitt inverter changes state at the input upper-threshold of approximately two-thirds the supply voltage. This negative transition uses INTO

to interrupt the processor, which then reads the current value of Timer0.

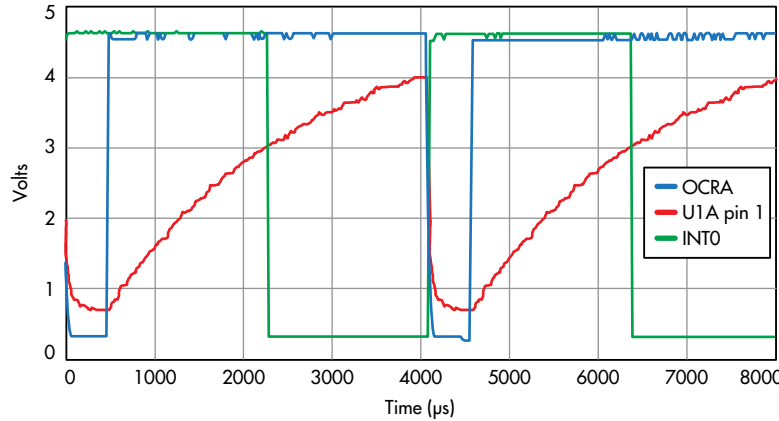
By approximating the value of V-high to be the same as the supply voltage allows pulse-width T to be derived using the formula:

$$T = RC \ln 3$$

which is independent of the supply voltage. With RC values of $R = 8.2 \text{ k}\Omega$ and $C = 0.2 \text{ }\mu\text{F}$, T is $1801 \text{ }\mu\text{s}$ of a PWM period of $4096 \text{ }\mu\text{s}$, giving a Timer0 count of 112. The experimental value obtained is 115.

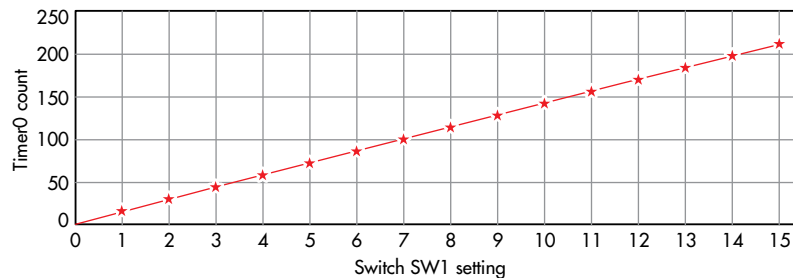
The *table* shows the Timer0 count captured by the interrupts for the 16-combinations of SW1. The count values at different supply voltages are also shown.

As shown in *Figure 3*, the count values captured at the different interrupts

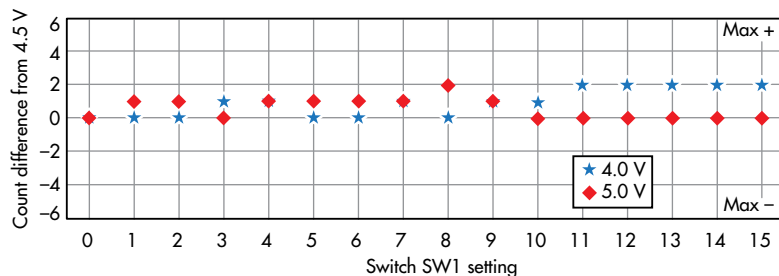


2. The Arduino PWM output OCRA charges the weighted-capacitor network, generating INTO at the Schmitt trigger's upper threshold of approximately 3 V.

EXPERIMENTAL VALUES OF TIMER0 COUNT WITH SW1 SETTINGS																
Sw1 Setting	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
VCC 4.5V	0	15	31	45	60	73	88	102	115	129	143	157	171	184	199	212
VCC 4.0V	0	15	31	44	59	73	88	101	115	128	142	155	169	182	197	210
VCC 5.0V	0	14	30	44	58	72	87	100	113	127	142	155	169	182	197	210



3. The count values seen at the different interrupts which match the sixteen combinations of SW1 show a good linearity of the result.



4. The plot of the count difference versus supply voltage demonstrates that changes for the 16 combinations of SW1 in the Timer0 count, as seen by the interrupts, are within acceptable limits.

corresponding to the 16 combinations of SW1 reveal a fairly linear characteristic, with a count difference of about 14 between settings. A software routine that permits a maximum-tolerable difference of ± 6 with respect to nominal

mid-values is used to decode the switch positions.

Figure 4 demonstrates that as both V-high of OCRA and the upper threshold of the Schmitt-inverter input vary proportionately with supply voltage, the change in the Timer0 count captured by the interrupts for the 16 combinations of SW1 are within acceptable limits. This is shown with respect to the values at 4.5 V, with the circuit operation at 4.0 V and 5.0 V.

Figure 5 shows the C code block used to decode the switch setting. A variable “duty” corresponding to the Timer0 count is passed to the function “get-switch,” which then returns the switch values based on comparing the count with the integer mid-values of the nominal count.

Using a common-output OCRA, this method could be expanded to use additional hex switches and the other

```

get_switch(uint8_t duty){
    if (duty<=7){ switch_value=0;}
    else{ if (duty> 7 && duty <=22 ){switch_value=1;}
        elseif (duty> 22 && duty <=37 ){switch_value=2;}
        elseif (duty>37 && duty <=51){switch_value=3;}
        elseif (duty> 51 && duty <=66){switch_value=4;}
        elseif (duty> 66 && duty <=80){switch_value=5;}
        elseif (duty> 80 && duty <=95){switch_value=6;}
    elseif (duty> 95 && duty <=108){switch_value=7;}
    elseif (duty> 108 && duty <=122){switch_value=8;}
    elseif (duty> 122&& duty <=136){switch_value=9;}
    elseif (duty> 136 && duty <=150){switch_value=10;}
    elseif (duty> 150 && duty <=164){switch_value=11;}
    elseif (duty> 164 && duty <=177){switch_value=12;}
    elseif (duty> 177 && duty <=191){switch_value=13;}
    elseif (duty> 191 && duty <=205){switch_value=14;}
    elseif (duty> 205){switch_value=15;}
    }
}
return (switch_value);
}

```

5. This block of C code decodes the Timer0 count and determines the switch value based on that count.

interrupts available on the Arduino Uno. The advantage of this approach is that it frees up analog-to-digital converters, Rx/Tx, and comparators, and can be used on minimum pin-count microcontrollers.

Ajoy Raman (www.ajoyraman.in) is retired after 37 years of R&D experience with the aeronautical development establishment in Bangalore, India. He has BTech and MTech degrees in electrical engineering from IIT Madras, India, and now pursues his love of electronics as a full-time hobby, with an emphasis on projects and low-cost teaching aids for students. He has received the Siemens prize for best academic record (M.Tech IIT, Madras), the Dr. V.M. Ghatge Award from the Aeronautical Society of India, the DRDO Scientist of the Year Award, and the IETE-CDIL Award for Industry. He can be reached at ajoyraman@gmail.com.

