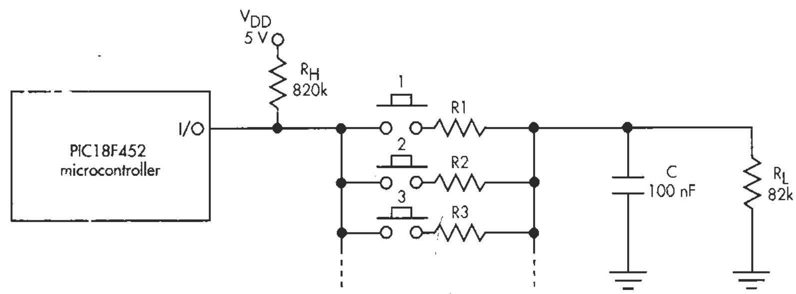
Power-Saving Keypad Controls Multiple Keys Through One MCU Pin

MEHMET EFE OZBEK | Atilim University, Incek, Golbasi, Ankara, Turkey
efe_ozbek@atilim.edu.tr

Traditionally, interfacing a microcontroller with an n-by-m keypad required n + m of the microcontroller's I/O pins for keypad scanning. Keypad designs that conserve microcontroller pins have been developed, but they require additional resources, such as external ICs or a built-in analog-to-digital converter (ADC). The design presented here uses only one I/O pin and requires only resistors and a capacitor as external components.



This keypad interface technique controls multiple keys using only one (bidirectional) pin on the microcontroller and a minimum of external components.

I/O is a bidirectional pin initially configured as an input *(see the figure)*. When no key is pressed, the capacitor is discharged and the pull-up resistor, R_H, keeps I/O High. The microcontroller is in sleep mode and will wake up only when a change in I/O's state generates an interrupt. When a key is pressed, I/O changes to Low, since the pull-down network is stronger than the pull-up. The microcontroller then wakes up to execute the following steps:

1. Wait for contact debouncing.
2. Change I/O to an output and set it High. The capacitor then starts charging to the High-state voltage. The charging time, T_i, is determined by the key pressed and its associated R (1, 2,...i).
3. Wait until T_1.
4. Make I/O an input. Charging of C pauses.
5. If I/O is High, key 1 was pressed. If I/O is Low, make I/O an output and set it High to continue charging.
6. Wait until T_2.
7. Make I/O an input. Charging of C pauses.
8. If I/O is High, key 2 was pressed. If I/O is Low, make I/O an output and set it High to continue charging.
9. Continue for T_3 through T_i.

Resistors should be chosen to make $T_1 < T_2 < T_3$...

Charging time can be determined as follows: When charging pauses, the voltage at I/O is *(Equation 1)* where V_C is the capacitor voltage *(Equation 2)*. Equation 3 solves the charging time by equating $V_{I/O}$ to the switching threshold voltage V_{TH}. Here, V_{TH} is the switching threshold voltage for I/O.

As R_i is increased, T_i initially increases. But then it reaches a maximum and starts to decrease. This imposes an upper limit on R_i and, hence, on the number of keys that can be connected to the circuit.

T_i may vary between $T_{i,min}$ and $T_{i,max}$ due to resistor tolerances and variations in V_{TH}. Therefore, the values should be chosen so that $T_{i,max} < T_{i+1,min}$. Assuming resistors with 5% tolerances and a maximum V_{TH} variation of 5%, a maximum of 15 keys can be connected to the circuit using the following R_i values (in kΩ): 0.01, 0.27, 0.62, 1.1, 1.8, 2.7, 3.9, 5.6, 8.2, 11, 15, 22, 30, 43, and 68. The number of keys can be increased if resistor tolerances are tighter.

This design saves power in three ways. First, energy of CV^2 is dissipated each time a capacitor is charged to V and discharged. In this design, charging stops as soon as I/O goes High and the capacitor is charged to about V_{TH} (less than 2 V), rather than V_{DD}. Second, the capacitor is charged (and discharged) only once for each key press. Finally, after determining which key was pressed, the microcontroller enters sleep mode and remains asleep until the key is released and the state of I/O changes back to High. So even when some of the keys are stuck or held down, power consumption is minimized.

$$V_{I/O} = (V_{DD} - V_C)\frac{R_T}{R_T + R_H} + V_C \qquad (1)$$

$$V_C = V_{DD}\frac{R_L}{R_L + R_i}\left(1 - \exp\left(\frac{-t}{C\frac{R_i R_L}{R_L + R_i}}\right)\right) \qquad (2)$$

$$T_i = V_{DD}\frac{R_L}{R_L + R_i} C \log_e\left(1 - \left(V_{TH} - V_{DD}\frac{R_i}{R_H + R_i}\right)\frac{R_H + R_i}{R_H + 2R_i}\frac{R_L + R_i}{V_{DD}R_L}\right) \qquad (3)$$

MEHMET EFE OZBEK *is an assistant professor in the Electrical and Electronics Engineering Department at Atilim University.*

ED ONLINE 16242