Extreme Business Cards

Technical Notes

The "Torch Card"

There is nothing particularly tricky about this one – a single cell, switch, LED and resistor. To work out the value of the resistor needed, you should really find out the voltage drop of the LED, subtract this from the voltage of the cell, and divide by the current required (let's say 15mA, or 0.015A). As an example, a typical LED has a voltage drop of 2V, so the resistor required is 2V/0.015A or 66 Ohms. Any standard value around this (68 or 72) would be fine. If you have an LED with a voltage of more than 3V, such as the 3.3V drop that is typical of most blue or white LEDs, then just leave out the resistor (short with a blob of solder or use a "0 Ohm" resistor), and hope for the best! Most high brightness white LEDs operate fine at 3V, even though you don't get maximum brightness from them – at least your keying torch will work for even longer!

The "Dialler Card"

If you know about electronics, you will find that I break all the rules with this one. The PIC is run at 6V, which is pushing it for the chip – although it is less than the absolute maximum ratings, it is outside the recommended operating conditions. In practice, however, this works fine, and most of the time the PIC is asleep anyway. The circuit works fine on 3V, but with the piezo I am using, the volume is not loud enough to reliably dial. I could have fixed this a number of ways, by using a step-up supply or an inductive circuit, but these are both more expensive in production and too thick for the design. I also have no decoupling capacitor, which for those in the know, is a big no-no as well. Fortunately, the circuit is small, and the impedance of the cell is low, and the current drawn by the piezo is almost zero, so again, in practice, there are negligible spikes on the supply, even when dialling. Theoretically, I should also have a pull-down resistor on the input line from the piezo as well, but again in practice this is not needed. The final rule that I break, is trying to generate DTMF tones with square waves – this shouldn't work at all because of harmonics, but in practice the waveform looks much more sinusoid when it enters the phone system because of the limited frequency response of both the piezo and microphone circuitry in the phone.

The circuit works as follows –

- The piezo is connected to two pins of the PIC – when dialling, one pin outputs one of the DTMF frequencies, one outputs the other, and simple maths can show that the overall differential signal applied to the piezo has the same frequency response as if you added the two frequencies rather than subtracted them.
- After dialling, the PIC goes to sleep until someone taps the card – the spike generated by the piezo then wakes up the PIC, and it dials again. The wakeup is designed to activate on a high logic level rather than a low, as the voltage differential needed on the pin for the transition is lower, making the card more sensitive. Technically, I should protect the PIC from overvoltage from the piezo as well, but the currents involved are miniscule, and again, in practice, it works fine. This is truly minimalistic design for the sake of a cheap product!

Design Improvements

All designs can be improved, and these ones are no different – as mentioned earlier, if producing the key ring torch professionally, I would reduce the PCB size even further, and put the contact details on the reverse. I would remove the cell holder and weld batteries directly – probably mounting them sunk into the PCB via a hole. I would possibly encapsulate the whole design as well, preferably in a see-through fashion, to prevent shorting on keys whilst in the pocket or handbag.

There are several changes I would make to the “dialler” as well – I would get the spacer cards combined into a single die-cut spacer (or maybe even laser cut!). I would probably also replace the front and back of the card with even thinner PVC. I am also working on changing the firmware so that the frequencies generated are in the form of pseudo sine waves, which would then be filtered via a single resistor and capacitor. The current design is operating right on the edge of the DTMF specifications (actually, it is out of spec most of the time), and the number gets recognised only about 50% of the time, even given some fine tuning of the piezo mounting. When I pipe a pseudo-sine wave to the piezo, I get near 100% recognition, so I know that it can work even more reliably. Interestingly, the frequency accuracy of the design is within DTMF limits, thanks to a calibration value pre-programmed into the RC oscillator of the PIC, but it is the harmonics and “twist” (relative amplitude of the two tones) that is out of spec. Stand by for version 2.00 of this instructable in the near future, for a commercial-ready version!