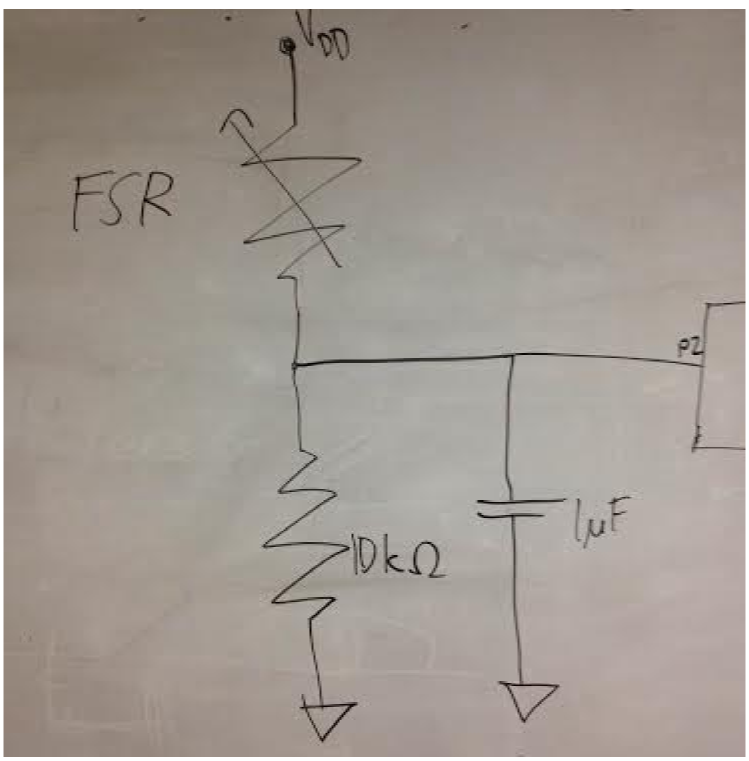
The KemereShoe
ELEC 327 Final Project
by Daniel Zdeblick and Vitor Andriotti

For our final project, we wanted to design something that would be both practical and entertaining, and so we created the KemereShoe, the perfect shoe for an incredible nighttime run unlike any other. This system attaches a Force Sensitive Resistor (FSR) to the sole of your shoe and an LED string to the side of your pant-leg and flashes the LEDs in time with your steps as you run. The KemereShoe can also tell you how long you've been running when you're done by plugging an LCD display in.

The FSR is simply a thin sheet of a conductive ink that becomes more conductive when compressed. We put this in a simple voltage divider (with a 10k resistor) with a capacitor to ground to reduce voltage spikes, generating a node voltage that can trigger interrupts on the MSP430 (see below).

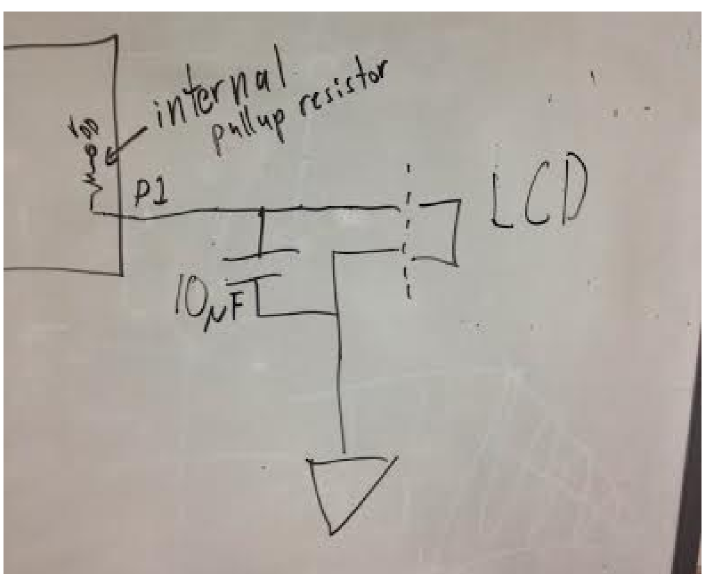


We designed our code so that the system waits in LPM3 for this interrupt, and then plans the next sequence of LED flashes in the running mode main loop before going back into LPM3. While the system is in LPM3, TA0 is configured to count seconds and measure the time between steps. In the running mode main loop, Timer A1 is configured to generate 58 interrupts over the time your last step took (an approximation of the step you are currently taking). Each interrupt adjusts and sends the SPI commands for the LEDs, changing which LEDs are on and what color they are. For aesthetic value, we chose to have two LED pulses that go in opposite directions up and down your leg synchronized and changing colors with the frequency of your steps. After each TA1 interrupt routine is done, the system enters LPM0 while it sends the SPI commands to the LEDs and then returns to LPM3 when it's done transmitting.

We needed to use LPM0 for SPI communication because the long string of LEDs requires a fast clock (we used the DCO at 16MHz) to function without errors. The long string also needs a better voltage source than the coin cell batteries, so we needed to purchase a rechargeable 3.7 V, 2500 mAh lithium ion battery that attaches externally to power our system. As recommended, we also sent the data signal to the LEDs through a 300 Ohm resistor to reduce voltage spikes. With all of these precautions taken, the string functions quite well.

If the last TA1 interrupt finishes, then the system halts the timer returns to LPM3, waiting for the next interrupt from the FSR (on P2). If this interrupt comes before the last TA1 interrupt, then the cycle is reset and starts anew without being allowed to finish. Provided that the runner does not drastically change their gait between steps (which is very difficult when running), this looks quite natural.

When the runner is done running, they need simply to plug in their LCD display, which they can keep in their pocket or at home. The LCD display uses parallel communication with three control signals (enable, register select, and read/write) and either 8 or 4 data buses. We did not need to read from the LCD and we only used 4 data buses, so we had plenty of unused (grounded) pins on our display. We manually shorted two of these pins together so that when the LCD is plugged in, it pulls a pin on our PCB low, generating a P1 interrupt (see below). We also used a 10uF capacitor connected to ground for debouncing purposes.



When this interrupt happens, it changes the mode of the entire system from running to reading. This disables TA1, resets the LED commands and re-sends them, and disables FSR interrupts. Because each LCD instruction requires a finite amount of time to process, we also needed to reconfigure TA0 for waiting in LPM0 while the instructions process. Finally, we set up the Watchdog Timer to wait a couple seconds in LPM3 after the P1 interrupt to give the user enough time to plug in the device securely and let it power on.

Once the Watchdog Timer has counted up to roughly two seconds, the program re-enters the main loop, but now in reading mode. This part of the loop initializes the LCD display, sends the number of steps and the duration of the run, and then resets these variables and reconfigures the P1 interrupt to occur when the LCD is removed. This interrupt will return the system to running mode, all ready for the next run!

As expected, we had a number of challenges with this project, but unexpectedly, the biggest problems were related to the hardware. This system is part of a shoe, and our prototype was just duct-taped on. Also, the FSR, battery, and LED strip all required jumper cable connections, which can easily become inconstant when the runner is running. Such loose connections cause unwanted interrupts, making our system nonfunctional. We fixed these problems with careful soldering on these connections, and more duct tape.

The bugs we encountered in the software were all relatively minor, and most had to do with logical errors in switching between running and reading mode. We also struggled with doing enough software debouncing for the FSR and LCD interrupts to account for a couple of bounces, although most of the bounces are removed by the capacitors to ground.

Our final system functions very well, catching almost exactly one step per actual step. We decided that what few misses remained after we had optimized our connections resulted from the variability of the FSR, sliding around in the shoe.

If we were redesigning the system for commercial production, we would include the shoe in the product, and insert the FSR and control mechanisms into the shoe itself instead of duct-taping them on. This would make the shoe look a lot more fashionable and improve the reliability of the FSR. We would also We would also make it possible to recharge the battery without disconnecting it from the shoe, allowing us to change all of the jumper connections to soldered ones. Pins or maybe velcro attached to the LED strip for affixation to the pant-leg would also be ideal (we just used duct tape for this).

The one major design change we would make to the system is replacing the LCD display with a reset button and one of two things (or maybe even both): either a smaller display that stays on the shoe and continuously displays your steps and runtime, or a bluetooth connection to a smartphone or smartwatch that sends the data when you’re done running. The latter especially seems like it would be a popular idea in our modern age of digital health.