Traditional Circuit Breaker

Limit to operation is high-current arcing when contacts are opened

http://electronics.howstuffworks.com/circuit-breaker2.htm
Just One Piece of the Power Grid

- The whole grid is moving toward Internet-based real time control
Where Are We Now?

- Where we’ve been:
  - Analog Output
  - Analog Input

- Where we’re going today:
  - Human I/O – LEDs and switches
  - Other typical embedded I/O

- Where we’re going next:
  - Gentle introduction to control
  - RTOS
  - Bluetooth & CAN
  - Booting & robust systems

Preview

- Switches
  - Debouncing
  - Switch Matrices

- LEDs
  - LED Matrices
  - Seven Segment Displays
  - LCD overview

- Linear electromagnetic devices
  - Relays
  - Solonoids

- Rotational Devices
  - Shaft encoders

- Other I/O-related concerns
  - Security
  - Human factors
Input Switches Revisited

◆ Pull-up input switch
  • Draws current from +5V when open

Figure 8.1
A simple switch interface.

◆ Pull-down input switch
  • More prone to noise fluctuations (less noise margin at 0V on most ICs)

Figure 8.2
Another simple switch interface.

CPU Module Switches Are Pull-Down/Active Low

◆ Horrible human interface marking – “OFF” side is really “ON”
  • Arrow points to OFF position in the DIP switches – but PCB says “ON”
  • May have to adjust your lab #9 code to deal with this
Switch Bounce

- Switches have mechanical contacts
  - They take time to move positions
  - They take time to mechanically stabilize when opening and closing
  - They create sparks, especially when opening at high current

Figure 8.4
Switch timing showing bounce on touch and release.

- Caution: 5 ms is representative for small switches, but not a universal number!

Hardware Switch Debounce

- Problem with bad circuit – causes arcing when closing the switch
  - Almost-infinite current when switch is first closed to discharge the capacitor
    - Switch bounces when being closed, so you get an arc from that high current

Figure 8.6
Switch bounce removed with a capacitor (this is a bad circuit).

Figure 8.11
A hardware interface that removes the bounce (good circuit compared to Figure 8.6).
Software Debouncing

- Saves external hardware components
  - Idea: wait until switch value is steady for 10 ms
  - BUT, counts on input hardware on IC to clean up indeterminate voltages
    - “Metastable” inputs – hang at middle of switching threshold for a long time
How Do You Read Multiple Switches?

- Simple way – one I/O pin per switch
  - This scales very poorly for large number of switches!

![Diagram of 68HC812A4 with switches PJ7 to PH0](image)

**Figure 8.24**
Hardware interface for the 6812.

Interfacing To Many Switches

- How can you reduce number of contacts for multiple switches?
  - For desktop keyboards – use an 8-bit microcontroller and serial interface
  - For other applications, need a clever switch arrangement

![Image of a keypad with keys](image)

**Figure 8.20**
Multiple keys are implemented by placing the switches in a matrix. (Notice there are fewer wires in the cable than there are keys.)
Switch Matrix

- Uses $2 \sqrt{N}$ pins for $N$ switches
  - Control input to switch as well as reading output
  - Activate only one Column ($P_{x0}$, $P_{x1}$, $P_{x2}$, $P_{x3}$) to high
    - Configure rest as ignored inputs or set to high impedance
  - Read Row values to find a depressed key ($P_{x4}$, $P_{x5}$, $P_{x6}$, $P_{x7}$)
  - What happens when two keys are pressed?
    - Short between two rows if they are in the same column, but that’s OK
    - Size resistors accordingly to ensure reasonable current with multiple closures

![Resistor Matrix Keypad Using Parallel Port Pins](image)

Figure 1. Resistor Matrix Keypad Using Parallel Port Pins

LEDs – Light Emitting Diodes

LEDs come in a wide variety of shapes, sizes, colors and configurations.

![LEDs](image)

Figure 8.28 (Valvano)
Driving LEDs

- The matrix trick works on driving LEDs too
  - Direct LEDs – one LED per output pin
  - Scanned LEDs:
    - Set one of Out4..Out7 high to activate a row of LEDs
    - Set one of Out0..Out3 high to activate a column of LEDs (output of inverter is low)
    - Only LED at intersection of selected row/column will see voltage difference

Multiplexed LEDs Scale Even Better

- To turn on one multiplexed LED:
  - Set Out4..Out7 to be the number of the row of the LED
    - For 4 output bits, there are 16 rows
  - Set one of Out0..Out3 to be the number of the column of the LED (output of inverter is low)
    - For 4 output bits, there are 16 columns
  - LED at specified row number/column turns on

- Practicalities:
  - One LED is always on
    - So perhaps 15x15 instead of 16x16
  - Need to turn each LED on long enough to be reasonable brightness
    - Perhaps make extra-bright
  - Diagram doesn’t show resistors to control LED current!
Seven-Segment Displays

Figure 8.35
Seven-segment common-anode LED interface.

LED Hex Digits

[Diagram of LED Hex Digits]

Valvano
Scanned Seven-Segment Display (3 digits)

- Turn on by: PBx high (one per segment) and PCx high (one per digit)
  - Resistor on PBx side so only one diode at a time goes through each resistor
  - PB side of diode goes low; PC side goes high, turning diode ON = light

\[
R = \frac{5V - V_D}{30mA} = \frac{5 - 0.7 - 1.1V}{30mA} = 40\Omega
\]

LCD – Liquid Crystal Display

- Low power, high contrast display technology
  - Pre-configured electrodes for display shapes
  - Display has two fixed polarization layers
  - Liquid crystals twist the polarization of light passing through
    - If crystals don’t twist, all light is blocked
      (horizontal | no twist | vertical ➔ no light)
    - If crystals twist, light gets through
      (horizontal | 90° twist | vertical ➔ all light)
  - Color & gray scale:
    - Gray scale: intermediate levels of twist by varying voltage
    - Color: use color filters on sub-pixels (3 colors per pixel)

- Interfacing methods vary
  - ASCII LCD has on-board microcontroller
    - Just send it bytes
  - Raw interface is usually done via scan lines
    - Similar to writing to DRAM
    - Uses a per-pixel capacitor to save row state
    - Must be refreshed periodically
  - Much lower power than LED
    - Passes through light source instead of emitting light

http://en.wikipedia.org/wiki/Liquid_crystal_display
A Word About Human Computer Interaction

- Not everyone is a 21 year old male engineering student
  - (Nor a female engineering student!)
  - So don’t design as if that is who your user population is
  - There are experts in the area of HCI – use them!

- Consider how well your system will work with these populations:
  - Non-English speakers
  - Left-handed (7-10% of population)
  - Color blind (esp. red/green – 7-10% of population)
  - Presbyopia (most people over 45 or so)
  - Polarized sunglasses (LCDs are also polarized; sometimes the wrong way)
  - Hearing impaired; wearing hearing protection
  - Gloves, coats, hats
  - Children (size, weight, child-proofing)
  - Arthritis (can’t manipulate small knobs, e.g., childproofing mechanisms)
  - Pets (e.g., cat on keyboard)

I have always wished that my computer would be as easy to use as my telephone.

My wish has come true.

I no longer know how to use my telephone.

– Bjarne Stroustrup, inventor of C++

http://www2.research.att.com/~bs/bs_faq.html#really-say-that
Example Usability Problem

- How do you flush this toilet without getting sprayed in the face?

Rotational Position Measurement

- Potentiometer
  - Variable resistance
  - Problem – requires A/D conversion
  - Problem – wears out
  - Problem – single turn – can’t put on a wheel
  - Problem – noise, calibration, …

- Alternate approach – how can we do this cheap & digital?
Optical Shaft Encoder

- **Idea** – use an optical pickup (e.g., photo transistor) looking at a disk on a shaft
  - Sense “white” or “black” as a “zero” or “one”
  - Can spin as many times as desired without mechanical limitation
  - No wear surfaces, no friction

- **Simplest encoder**
  - One digital “tick” per revolution
  - E.g., a piece of white tape on a black tire

- **Relative positional encoder**
  - Multiple “ticks” per revolution – like gear teeth
  - Need to keep track of how many ticks you’ve seen
  - Can put a really large number of ticks on a disk
    - Limit is tick size and diameter of shaft encoder disk

Absolute Position Shaft Encoder

- **Use multiple concentric shaft encoder values**
  - Any angle reads bars as a set of bits
  - Gives unique binary number for any rotational angle

- **Diagram is conceptual**
  - Real encoders don’t have the lines
  - Real encoders pattern on outer edge only
  - Resolution limited by inner-most band sized

- **Don’t use regular binary counting!**
  - Bit edges & read head won’t be perfectly aligned
  - Use gray code to eliminate glitches
    - Each adjacent # differs by only one bit
    - E.g.: 000, 001, 011, 010, 110, 111, 101, 100, 000

[Note: real disk doesn’t have the lines – just the large black bars]
Solonoids (Reminder Slide)

- **Used to generate a short-stroke linear motion**
  - Release driven by spring, gravity, or second solonoid on same armature

**Figure 8.64**
Mechanical drawing of a solenoid showing that the EM coil causes the armature to move.

Relays

- **Used when switching high currents**
  - Uses a solonoid + spring (or similar arrangement) to open and close a switch
  - Provides physical isolation when open (“air gap”)
  - Can be “open” or “closed”
  - For high current applications, need special care to deal with arcing
    - Software used to open switch at the zero crossing on AC circuits, reducing contactor wear

**Figure 8.65**
Picture of an EM, solid state, and Reed relay.
“Ice Cube” Control Relay (e.g., for Elevators)

- Metal “Common Contact” swings up and down with armature

http://www.wadeinstruments.com/relays/control_relay_tutorial.htm
Embedded Systems and the Internet

- General trend to add connectivity from embedded to external world
  - Add Internet connectivity to household appliances
  - Wireless vehicle-to-vehicle networks
  - Additional services for user in planes or cars
- Why? Enables some great features!

Security Basics In One Informal Slide

- Properties you may care about:
  - Secrecy – nobody else can see your data
    - Huge in Internet systems; not necessarily important for embedded
    - Usually provided via encryption
  - Integrity – you are sure the data has not been altered
    - Usually the #1 concern for safety-critical systems
    - Best provided via digital signatures or secure hash functions
  - Authentication – only authorized sources can read/write/manipulate system
    - Important for all systems
  - Privacy – nobody can infer personal information about you
    - Not quite the same as secrecy – you might want to have privacy from trusted parties
  - Availability – system will operate when you need it to
    - “Denial of service attacks” are a financial problem for web sites, but potentially deadly for embedded systems
- Key insights for embedded systems
  - Encryption is often the wrong tool for integrity+authentication
  - It’s a really good idea to take a security course while you’re at CMU
Misconception #1: Security Through Obscurity

◆ It’s so complicated that we don’t need security!
  • It took a couple grad students a semester to reverse engineer the unpublished communications protocol for a wireless pacemaker
  • You can even access the debug mode
  • PS: the debug mode is a defibrillator.

◆ Reality: Reverse engineering is not as hard as you would imagine
  • It’s only a matter of time and money
  • Anti-tamper techniques are pretty tricky to get right

Misconception #2: We Can Trust the User

◆ Misconception 2.1: Assume user will maintain the system
  • This is semi-reasonable for PCs
  • But what about my mom’s sewing machine?
  • Is your mom a qualified sysadmin?
  • Will a bot-herder send spam from sewing machines?
    – (or a portable Windows-based oscilloscope?)

◆ Misconception 2.2: Assume outside attackers only, trust the user
  • In embedded systems, the user is often the most hostile attacker
    – Anyone jail-break their iPhone?
    – Did you secure it after the jail-break?
  • Even in critical systems – modifying car engine software
    – Re-tune engine for high performance/bad emissions
    – Over-ride max engine and vehicle speed
      » Put in place because OEM tires max out at 90 mph
**Misconception #3: Discipline Will Solve Security**

- Hackers can’t hurt your car if the infotainment system doesn’t talk to the braking system
  - Solution: don’t put a connection between the radio and the brakes…
    BUT – this is unrealistic; it will happen because customers demand it!

- **Product idea: Radio volume to achieve constant SNR**
  - Noise based on wheel speed, tire pressure, road surface
  - Which sensors have good information about this?
  - *The electronic stability control system!*
    - *Which is an evolution of anti-lock braking*
  - Reality: the connectivity will happen, denial is counter productive
  - Prototype vehicle of a Big-3 manufacturer suffered failure when the radio speaker caused an engine controller malfunction

---

**Misconception #4: Just Slap a Firewall in There**

- Obviously, we share some vulnerabilities that Internet and PCs have
  - Especially if we run standard Internet communication protocol stacks
  - And, we often have fewer run-time resources to fight off attacks, no sysadmin, etc.

- **Standard approach:**
  - Firewalls
  - Intrusion detection systems
  - Strong cryptographic mechanisms

- **Reality: Embedded network ≠ Internet**
  - Standard security solutions don’t necessarily fit

- **Quickly becomes a research area:**
  - What goes into an embedded gateway?
    - How do you transfer real time control data between CAN and Internet?
  - Strong cryptography?
    - How do you crowbar a 20 byte multicast authenticator into an 8 byte packet?
HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

A-330 Running Internet Explorer
Privacy

- **Embedded systems can collect all sorts of info**
  - Location, health, driving habits, activity, …
  - Consumption (food, electricity, gasoline, …)
  - Who should have access to that information? Who actually does have access?
  - Do you believe that anonymization really works? (mostly it doesn’t)

- **Privacy is really hard to get right**
  - Even if you have perfect crypto, there is more to it than that
  - People willingly give away info
  - Governments and companies actively try to get info without permission
  - Will privacy sell as a feature? How much would you pay?

- **Do we actually have privacy?**
  - “You have zero privacy anyway,” Sun Microsystems chief executive Scott McNealy famously said in 1999. “Get over it.”
  - December 2010, Google Chief Executive Eric Schmidt in a CNBC interview: “If you have something that you don’t want anyone to know, maybe you shouldn’t be doing it in the first place.”
  - 2014: NSA tracks who is on Tor; scans e-mail/chat/phone; etc.

Review

- **Switches**
  - Know how to do debouncing in software

- **Matrix interface**
  - Know how to read switches with a matrix
  - Know how to light LEDs with a matrix

- **Electromagnetic devices**
  - Know general principle of operation for a relay
  - Know general principle of operation for a solonoid
  - Know general principle of operation for a stepper motor

- **Rotational Devices**
  - Know how a shaft encoder works
  - Be able to recognize/create a gray code
  - Know how steps per revolution works on stepper motors

- **Security**
  - Key security properties
  - Four myths

- **Human Interaction Considerations**
  - Examples of things to consider