

# **ECE 471 – Embedded Systems**

## **Lecture 32**

Vince Weaver

`https://web.eece.maine.edu/~vweaver`

`vincent.weaver@maine.edu`

2 December 2024

# Announcements

- Don't forget course evals
- I have HW#6 and HW#7 graded, will send out
- Remember to return parts
- Will have midterm graded Wednesday
- Will discuss final Wednesday
- Will give sample project presentation Wednesday
- Will send out preliminary project schedule
- Don't forget HW#10 (Power)  
Due Friday. Might not have time to go over in class but



will post solutions



## Aside on HW#7

- For channel 0/1 want to write 0x80 or 0x90
- In binary this would be 1000.0000 1001.0000
- How do you write that in C? 0b10000000  
This was non-standard for a long time
- What if you try to just write out[1]=10000000
- It actually will work? 10,000,000 is 0x989680 and truncates when written to byte, so 0x80



# Project Update

- Will Send out tentative project schedule. Note can change date within reason. Also the order by day is arbitrary, usually I ask for volunteers
- Aim for about 7 minutes (5 presentation+2 demo/questions )  
might have to make this less on Friday
- Will give example presentation Wednesday

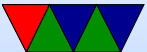


# Ethics in Software Engineering

- How do you define it?
- There's an IEEE document: <https://www.computer.org/education/code-of-ethics>
- It's long, a few bullet points
  - Public interest (be ethical to public)
  - Client+Employer (do best for company, but not if hurts public)
  - Product (make best product you can)
  - Judgement+Integrity



- Management (how you develop your software)
- Profession
- Colleagues
- Self



# Ethics in Software Engineering

- We talked about accidental bugs in software
- What about intentionally bad or misleading code?
- What if company wants you to code Dark Patterns?
- Are ethics involved when programming?





# Dark Patterns in Interfaces

- Making it easy to accidentally do bad/expensive things
- Making it hard to cancel or close windows
- Make interfaces confusing to trick people
- Have you ever tried to install Windows w/o a Microsoft account?



# Privacy / Tracking

- Privacy? Data Logging? Tracking?
- Tracking: should smart TV's snoop on what's on the screen and report back to advertisers?
- Accessibility / is your project usable by someone with a disability?
- Many computer companies make most of their money by tracking and logging user activity and selling it
- Is this wrong?



# Citing Sources

- Properly citing sources/giving credit
- Properly licensing code
- Do AI companies do this properly?



# Ethics Examples

- Unintentional security leaks: fitness trackers giving away military locations
- Thermostats: forget to change password if move or divorce, others now control your heating
- Amazon/Google devices, always listening in your house
- Get your youtube account banned, locked out of your google home, can't even contact a human to protest
- Web-cameras everywhere
- AI tracking all license plates



- Printers that won't let you use replacement cartridges
- Electric Toothbrushes that loudly beep if they think your brush head has expired
- Creating booby-trapped embedded systems (Lebanon pager incident)
- AI telling militaries which targets to blow up



# Can you control who uses your software?

- What if you have open-source code but someone uses it for evil purposes?
- What if you are a contractor and someone hires you to write evil code?



# Reporting Ethics Violations

- Who can you report them to?
- What about the IEEE/ACM who claim they have ethics hotlines?
- They are manned by lawyers and possibly won't do anything if they think they might get sued
- See the Huixiang Chen / ISCA incident



# Example: FTDI USB/Serial Bricking Incident

- Is cloning / counterfeiting popular chips wrong?
- What if you steal their USB-ID/ Trademarks?
- What if they fight back by having their driver disable counterfeit devices
- How do you feel if you didn't realize you had a counterfeit chip and suddenly your important embedded device stops working
- Who is to blame? Who are you likely to blame?





# Other I/O You'll find on Embedded Boards



# SD/MMC

- MultiMediaCard (MMC) 1997
- Secure Digital (SD) is an extension (1999)
- SDSC (standard capacity), SDHC (high capacity), SDXC (extended capacity), SDIO (I/O)
- Standard/Mini/Micro sizes
- SDHC up to 32GB, SDXC up to 2TB
- Support different amounts of sustained I/O. Class rating 2, 4, 6, 10 (MB/s)
- Patents. Need license for making.



# SD/MMC Hardware Interface

- 9 pins (8 pins on micro)
- Starts in 3.3V, can switch to 1.8V
- Write protect notch. Ignored on pi?



# SD/MMC Software Interface

- SPI bus mode
- One bit mode – separate command and data channels
- Four-bit mode
- Initially communicate over 1-bit interface to report sizes, config, etc.
- DRM built in, on some boards up to 10% of space to handle digital rights



# SDIO

- SDIO – can have I/O like GPS, wireless, camera
- Can actually fit full Linux ARM server on a wireless SDIO card



# eMMC

- eMMC = like SD card, but soldered onto board



# More Embedded Board Busses/Interfaces



# Camera Port

- The SoC has dedicated hardware for driving cameras
- CSI port (Camera Serial Interface) plus i2c bus to command it.
- Can read data in parallel, directly, without needing USB overhead.
- These chips often used in cell-phones, so makes sense to have support for camera-phone without extra chip being needed.
- Might need to use special tool to get still images (mmal





interface), until recently not using more common video-4-linux API



# Camera Models

- 2013 – 5 Megapixel
- 2016 – 8 Megapixels (Model2)
- 2023 – 12 Megapixels (Model3) (wide/narrow versions)
- NoIR – infrared filter removed so can take IR images
- GS – Global (vs rolling) shutter version
- 2024 – Pi AI Camera



# Touchscreen Display Port

- DSI
- Touchscreen display that can make a Pi look sort of like a smartphone
- GPU can output to it directly



# UART – serial port

- Note: Asynchronous, no clock (unlike USART)  
how do both sides agree on speed?
- Often useful on embedded boards and old systems, might be only way to reliably connect
- RS-232, originally for teletypes
- 3-15V high, -3 to -15V low
- start/stop bits, parity, bit-size
- Hardware vs Software flow control
- Speeds 300bps - 115000bps and beyond



- 50feet (15m) w/o special cables
- 3-pin version (transmit, receive ground). Also 5-pin HW flow control (CTS/RTS). Can have 2-pin version if only want to transmit
- These days often hook up USB connector
- What does 9600N81 mean?



# Pi Serial Ports

- Pi originally had two serial ports: good one and lousy one  
They switched them up with Pi3 (due to bluetooth)
- Pi4 adds a bunch more of them
- Pi does TTL (5v/0) not RS232
- Does support HW flow control, but need to activate those pins custom, is a bit complicated
- Use TTL to USB serial converter usually.



# Pi SMI

- <https://iosoft.blog/2020/07/16/raspberry-pi-smi>
- Secondary Memory Interface
- Available on Pis
- Allows creating wide parallel bus out of GPIOs
- Not well documented

