ECE 271 – Microcomputer Architecture and Applications Lecture 27

Vince Weaver

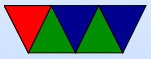
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

28 April 2022

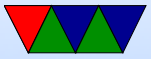
Announcements

- Don't forget course reviews
- Don't forget final: Thursday May 5th 8am Barrows 119
  Allowed one page (8.5"x11") of notes and calculator
- Try to get any outstanding labs checked off by Pascal this week, as Thursday is the last day of makeup labs



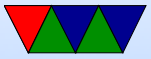
Question from Last Time: setup.s

- Even with an OS your main() call doesn't get called directly
- Usually some sort of startup code gets called first
- The binary image uploaded to the board, starts with the stack location, interrupt vector table, the reset vector, your code, and the data/bss
  (the linker puts these things in the right places based on a linker-script)
- At startup the stack pointer is loaded with the value

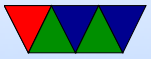


specified (usually top of RAM)

- Interrupt vectors are where they should be
- Reset vector is called, which has setup code
  - This copies writable global variables up into the RAM (where they can be modified)
  - This also clears out the BSS variables in RAM so they are zero (there's no guarantee RAM starts out as 0)
  - Any other init happens too (enable FPU for example)
  - Finally main() is called

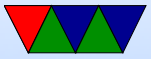


Other features of the STM32L476VG Discovery



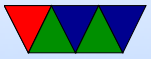
VREFBUF

- STM manual chapter 21
- Voltage Reference for ADC



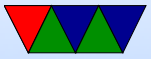
MCU Current ammeter

- Can be used to measure current
- 60na to 50mA
- External (with an ammeter)
- Jumper for internal measurement too
- How does that work?
  Looks like the MFX_V2 chip (multi-function expander chip), possibly via i2c 0x84
- How can you calculate power?
  Sense or shunt resistor, measure voltage drop, then

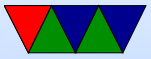


Ohm's Law. P=IV

- Why is this useful?

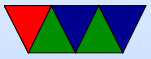


Hardware Accelerators



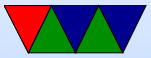
Firewall

- STM Chapter 4
- Can protect parts of FLASH or SRAM so they can't be modified
- Keep buggy code from touching things it shouldn't



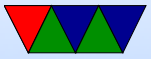
Hash Processor

- STM Chapter 29
- Can do crypto hashes, md5, sha-1, sha-256

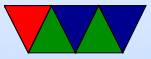


Device Electronic Signature

- STM Chapter 49
- Unique ID for board

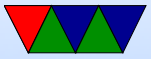


I/O Busses

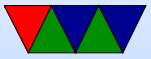


USB OTG FS

- STM Chapter 47
- USB on-the-go Full Speed
- USB-micro connector
- Allows devices to be either USB devices or hosts
- If a host, you can hook up devices to it like keyboards or USB keys
- If a device, you *act* like a device, so can be a keyboard, USB key, or anything else
- OTG can switch between, sort of like if you have a

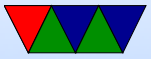


cellphone and when you hook to a computer want it to appear as a disk (to transfer photos) but if you hook up a keyboard want it to be like a computer



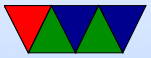
Single Wire Protocol – SWPMI

- STM chapter 44
- Single Wire Protocol
- *not* the same as 1-wire protocol
- Used for talking to SIM chips in phones? Also contactless interfaces?
- M to S by voltage modulation
- S to M by current modulation



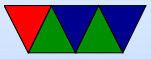
SDMMC (sd card)

- STM chapter 45
- 4-bit mode?



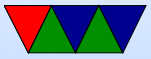
Infrared Interface (IRTIM)

- STM Chapter 35
- TIM16 and TIM17 joined up to generate proper pulses TIM17 high freq carrier, TIM16 modulation envelope
- To get signal use input capture
- PB9 can be configured to sink high current of LED

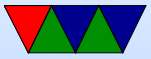


Serial Audio Interface (SAI)

- STM chapter 43
- Two of them
- Can be i2s, or AC'97, or other
- 8 to 32 bit
- FIFO
- 2-channel DMA
- Can handle SPDIF

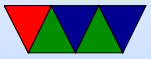


External Devices



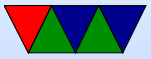
Audio DAC (note, not the internal DAC)

- CS43L22 (U13) – see notes on project
- Cirrus logic
- Connected via i2c, address 0x94
- Stereo output on audio jack
- Class D amplifier
- Beep generator



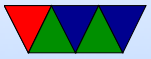
Chiptune Player (showed off a week ago)

- Want to play small tracked chiptune music
- pt3_lib decodes to series of AY-3-8910 register writes (this code based on some Turbo Pascal and z80 code with comments in Russian)
- Then we emulate an AY-3-8910 chip in software (existing code by Alexander Sashnov)
- We decode 0.5s or so of 44.1kHz 16-bit audio samples
- Placed in a buffer, which is DMAd to the SAI
- When half done, we call the decoder and generate

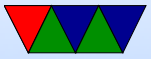


another 0.5s of audio

- Configuring the Hardware
  - Set to use 80MHz clock
  - Also configure joystick, LCD, and LEDs for debug and interface
  - Set up i2c to control DAC
  - Set up SAI to get data via DMA and output i2s compatible audio packets to the CS43L22 DAC
  - Set up the DAC with i2c to be unumuted, set the volume, frequency, sample size, etc
- The ay player uses some 64-bit division routines. We

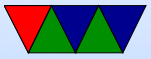


have to include some as our CPU doesn't support them natively.



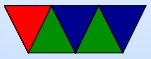
DFSDM – Digital Filter for Sigma Delta Modulator

- STM Chapter 24
- High-end DAC?
- Sort of reverse PWM? Takes analog signal and gives PWM like output



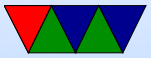
MEMS Microphone

- What is MEMS
- MP34DT01
- Hooked to DFDSM?
- L/R channel pin but seems not to be stereo, but in this case clock phase to have valid data



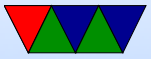
Quad-SPI Flash

- STM Chapter 17
- Regular SPI only transfers one bit at a time
- Quad-SPI can send 4-bits at a time
- Up to 40Mbit/so
- Can do things like erase/write flash



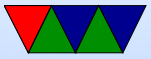
Some notes on FLASH storage

- Can you use it as general purpose RAM?
- Only has limited number of writes (10k?) before it wears out
- What happens when wear out? Has some extra blocks, wear-leveling can be done
- Possibly the FLASH has it's own micro-controller on board to handle the wear-leveling (remapping blocks)
- Can you sneak your own code onto this micro-controller?



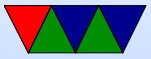
Microcontroller Clock Output (MCO)

- Can put the clock out on a pin



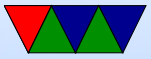
JTAG Debugging

- STM Chapter 48
- JTMS/JTCK/JTDI
- NJTRST



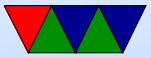
Independent Watchdog (IWDG) System Window Watchdog (WWDG)

- STM Chapters 36 + 37
- Counter counts down, if it hits 0 something was wrong and resets system
- Write special value 0x0000AAAA to re-up
- Dedicated low-speed clock in case something goes wrong with clock setup



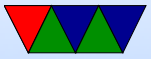
Real Time Clock

- STM Chapters 38
- Keeps time/date
- Can set alarm
- Can handle leap years
- daylight savings (not automatic, but quick mode to add/subtract 1 hour)



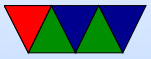
RTC Tamper Detection

- Can have switches (cover switch) and record if someone is tampering with system
- Also can zero out registers (secrets? Encryption keys?) if tampered



How Secure is hardware?

- All kinds of things can be done if you have physical access
- Decap chips
- Fuzzing/Glitching
- Hot-swapping RAM
- JTAG, probing



Looking at the Apple II