Sheaff

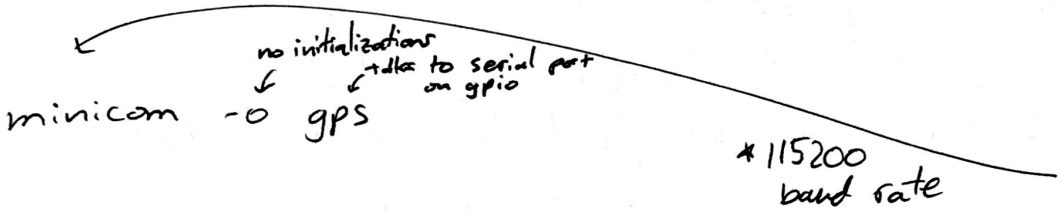
Nicholas LaJoie
ECE 331
3/21/17

* HW #5 has a spreadsheet rubric
* Cloning the kmorse repo...
  - clone, compile, load, verify device tree stuff is there

Flash Firmware:
- Patched AVRdude
- Firmware posted

* Makefile for kernmod...

Project:
- Use Linux dictionary (word-by-word)
- Debugging
  - Disable BT on Pi 3
  - boot/config.txt



* put spaces between letters for debugging

* How to set up minicom?

* Remove checksum/preamble - capture both - do diff to verify

---

Kernel Module:
- stat header (permissions?)
- device tree: (pretty new - still in dev) → file: .dts
  - probs hardware, makes text hash, finds drivers that match
- For RPi - at boot, i2c ~~bus~~ bus tells about hardware there
  - Graphics processor boots, tells ARM to load kernel
- Expansion board emulates "EEPROM"
  - Pin 4, 17, 22, 23 ← kernel driver will take ownership of these
- EEPROM settings file
  - device tree compiler combines .dts & EEPROM settings into bin. file, perl converts to C, dumps it onto board

[* Concurrency, locking]

* "Probe & Remove" "brcm", "bcm2835-morse"
name of board

[* "of" - open firmware (device tree is part of this)]

* null terminated array (sentinal) of matches) - morse_of_match[]

* MODULE_DEVICE_TABLE(of, morse_of_match)

* .probe = ... ← allows you to init members in order you want (gcc extension)

member (ptr. to func.)

[* initialization of a structure]

* Look @ the source of other drivers! (in Linux src)

[* MAKE SURE BOARD IS PLUGGED IN WHEN YOU LOAD
* Don't have programmer plugged in when you reboot!]

* /sys/firmware/devicetree/base/hat ← info about board
/soc/morse ← info on pin

* In kernel:

```
ret = -ENODEV; //Based on ERRNO
goto fail; //gonna need (always negative)
```

to use this — need to free memory, too many errors to possibly take place

(don't get kernel memory leaks!)

[* Check return values!]

* Kernel fault = back trace (need to reboot)

---

* Know: allocating memory in kernel (kmalloc) [* Use memset]

-> (size, type of memory) => [GFP_ATOMIC (ch. 7 of dev. driver book)
↑ get free pages
outside of process context, never sleeps
=> memory will stay on processor
alloc

* Addresses:
· Physical, virtual, kernel

[* Organize data with structures]