



## microYocto and the Internet of Tiny

Tom Zanussi, Intel

ELC • San Jose, CA • 30 Apr 2014

# Overview

- What is microYocto?
- Static Memory Footprints
- Reducing Dynamic Memory
- Quick (2-slide) Intro to Yocto
- Building/booting microYocto
- Future
- Questions



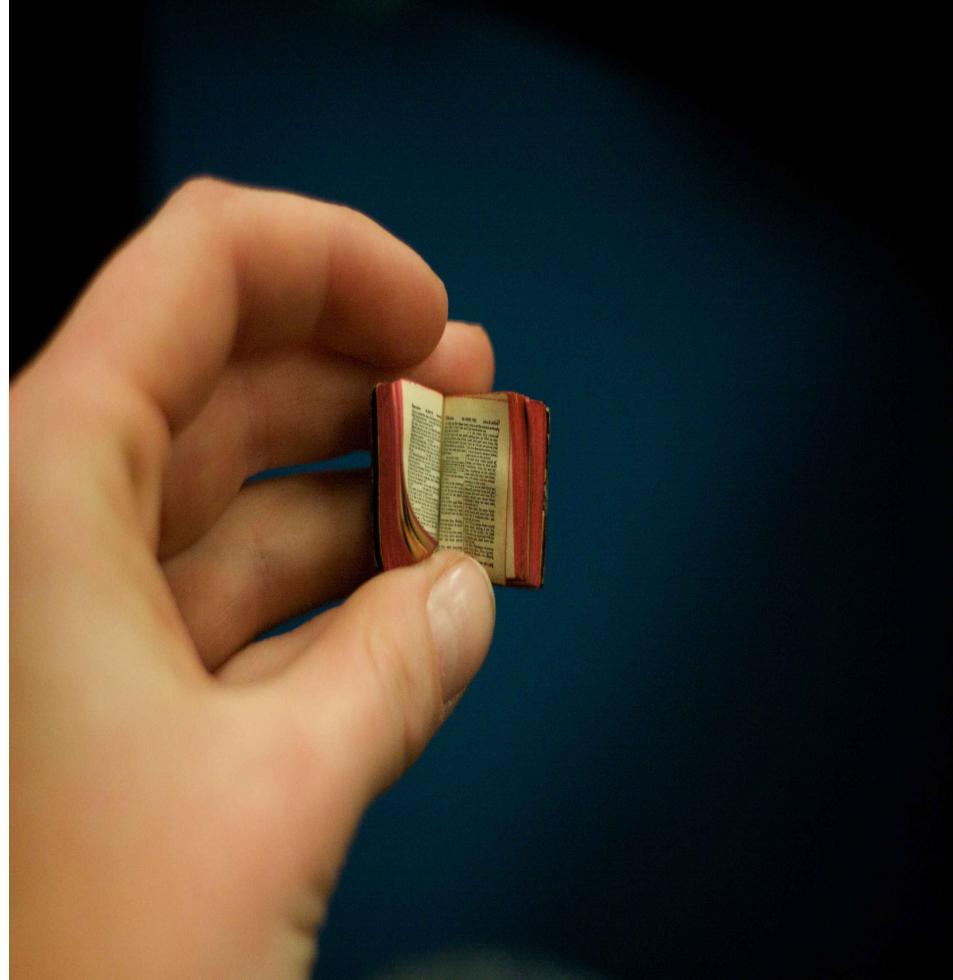
# What is microYocto?

- Tiny Yocto-based distro
  - Quark (currently Galileo)
  - 1.6 MB SRAM
  - 8 MB flash storage
- IOT == TCP/IP networking
- Single-purpose
  - IPMI/DCMI app
  - No Production Shell
  - Web server with CGI



# What to Make Tiny?

- **Storage**
  - < 8MB filesystem easy
- **User Space Memory Usage**
  - Remainder after kernel
  - Paging helps
- **Kernel Memory Usage**
  - Static - Always in RAM
  - Dynamic – As-needed



# Reducing Static Size

- Basically just remove code
- Disable CONFIG\_\*
  - Block layer (save 200k)
  - printk (save 120k)
- Create new CONFIG\_\*
  - PROC\_MIN (save 100k)
  - PERF (save 135k)



# Net-diet and LTO Patches

- Andi Kleen's net-diet
  - Break up network stack
  - **CONFIG\_RTNETLINK**
  - **CONFIG\_FIB\_LIST**, etc.
- Link Time Optimization
  - Beyond compilation unit
  - Better inlining choices
- Total savings > 400k



# Reducing Static Size

- Some new 3.15 patches
  - **SYSFS\_SYSCALL** (1k)
  - **USELIB** (save 1k)
  - **BUG\_ON** fixes
- Upcoming? (Josh Triplett)
  - **X86\_IOPORT** (save 10k)
  - **CONFIG\_PTRACE**
  - **CONFIG\_SIGNALS**



# 'Internet of Things' Size

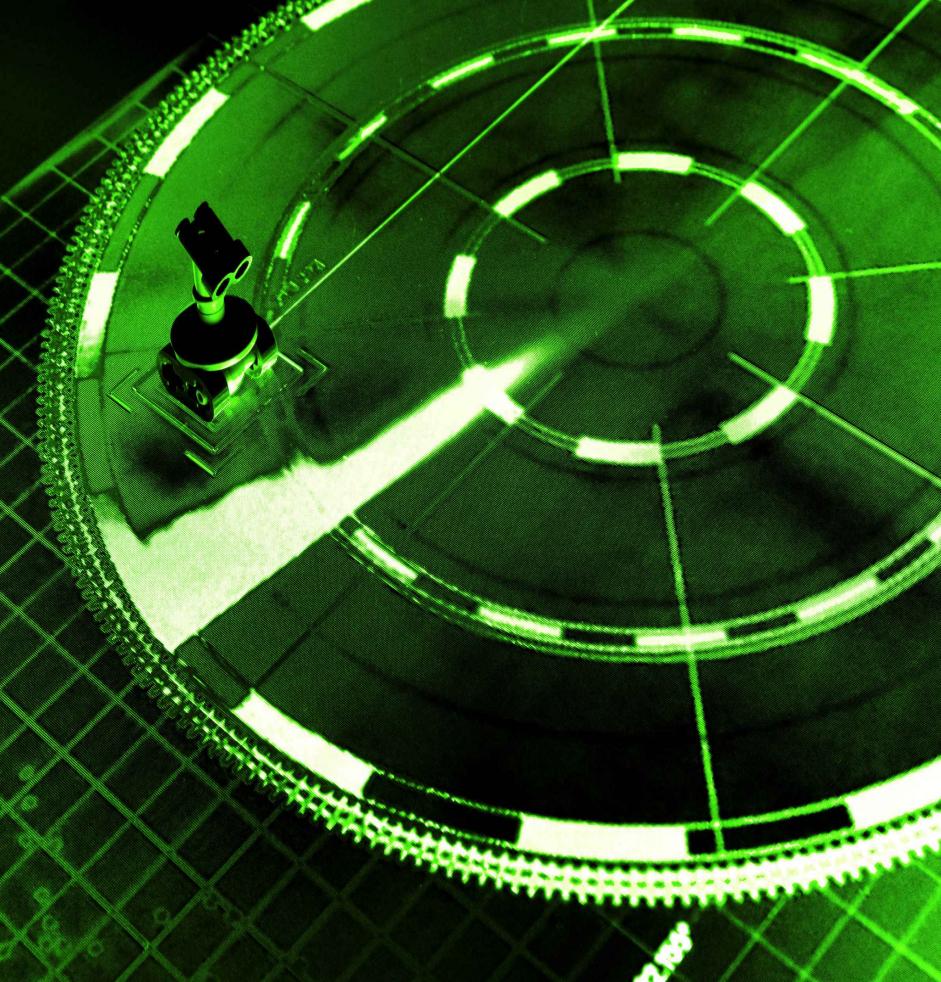
- Native networking stack
- Shell and utils (busybox)
- Webserver + CGI (nostromo)
- Running on Galileo Board
- Current size 766k TXT:

```
root@galileo:~# cat /proc/virt_kmem
virtual kernel memory layout:
  .init : 0xc10f0000 - 0xc1116000  ( 152 kB)
  .data : 0xc10bfb00 - 0xc10efc40  ( 192 kB)
  .text : 0xc1000000 - 0xc10bfb00  ( 766 kB)
```



# 'Internet of Pings'

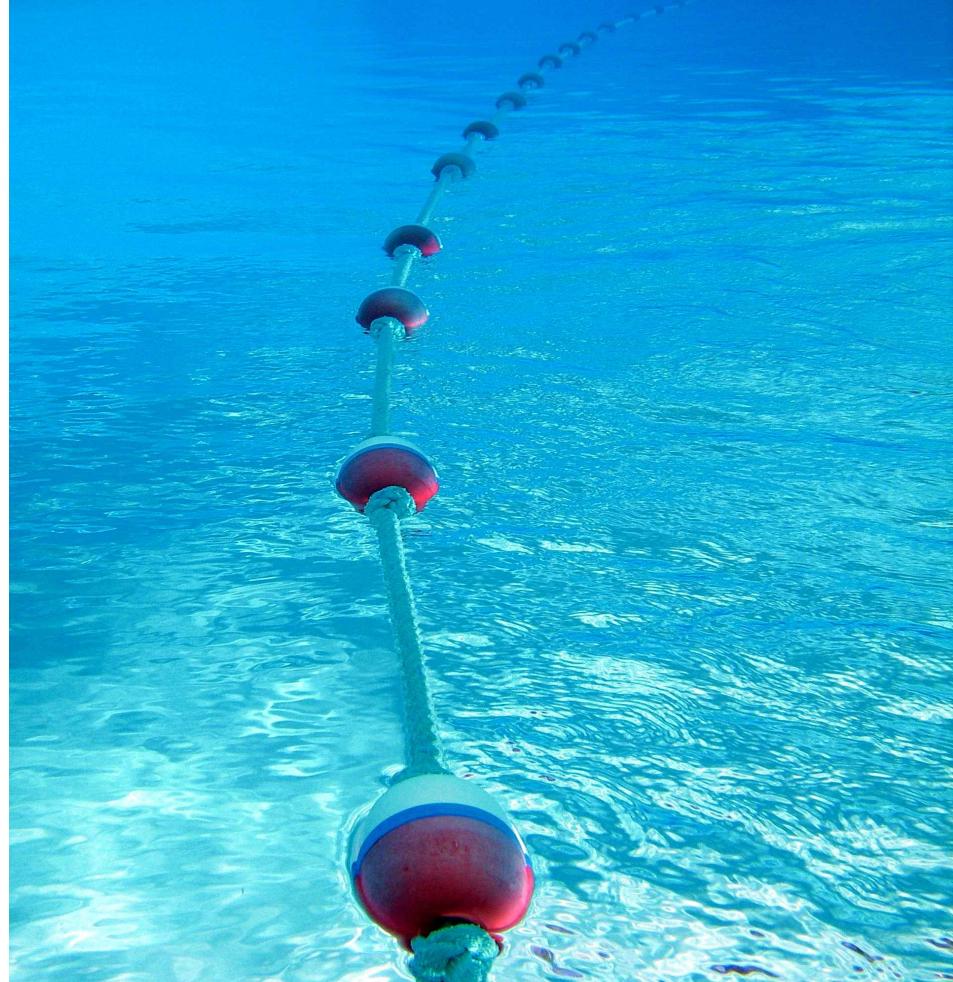
- Disable CONFIG\_INET (150k)
- Replace with userspace stack
- AF\_PACKET for ethernet\_if
- LWIP - two interfaces
  - High-level sockets
  - Low-level interface
- Hand-craft packets
  - IPMI: 6 simple UDP msgs



# 'Internet of Pings' Size

- Using Low-level API
  - TCP/UDP echo app
  - Non-trivial app changes
- Shell and utils (busybox)
- Running on Galileo Board
- Current size 620k TXT:

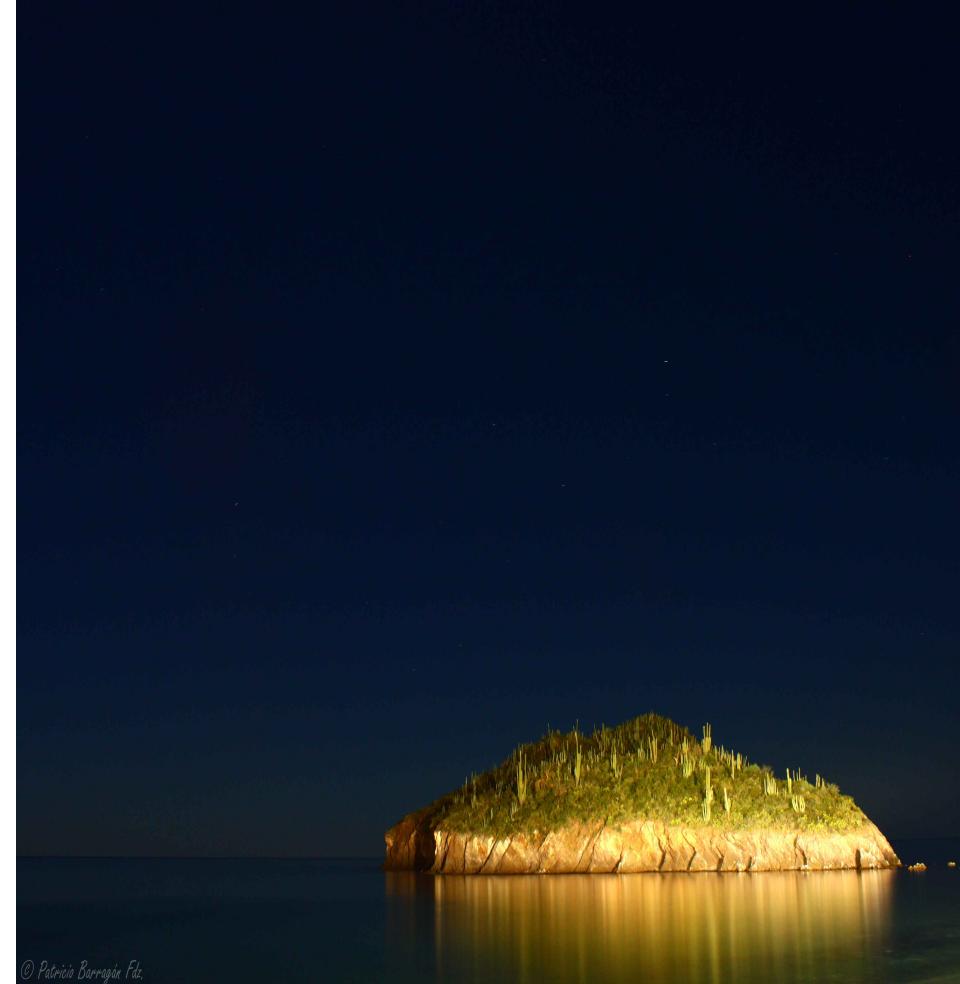
```
root@galileo:~# cat /proc/virt_kmem
virtual kernel memory layout:
  .init : 0xc10c9000 - 0xc10ef000  ( 152 kB)
  .data : 0xc109b130 - 0xc10c8c20  ( 182 kB)
  .text : 0xc1000000 - 0xc109b130  ( 620 kB)
```



# 'Just Things' Size

- No networking
  - Still useful (serial-only)
- Shell and utils (busybox)
- Running on Galileo Board
- Current size 535k TXT:

```
root@galileo:~# cat /proc/virt_kmem
virtual kernel memory layout:
  .init : 0xc10b0000 - 0xc10d4000  ( 144 kB)
  .data : 0xc1085ea0 - 0xc10afe60  ( 167 kB)
  .text : 0xc1000000 - 0xc1085ea0  ( 535 kB)
```



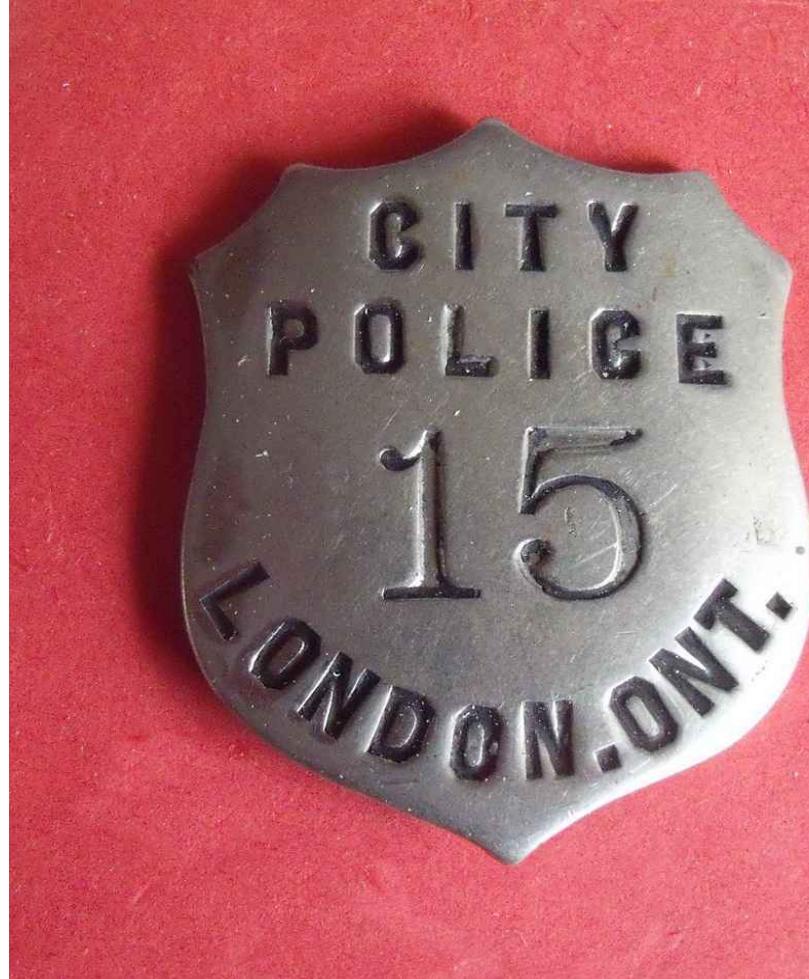
© Patricia Barragán Fdz.

# sendfile? We Don't Need No Stinkin' sendfile

- trace syscalls used

```
pid:nhttpd[590],    id:sys_read      vals: count:35
pid:nhttpd[590],    id:sys_close     vals: count:1040
pid:catafile[591],  id:sys_exit      vals: count:1
pid:nhttpd[591],    id:sys_execve    vals: count:1
pid:catafile[591],  id:sys_readlink  vals: count:1
pid:catafile[591],  id:sys_munmap    vals: count:1
pid:catafile[591],  id:sys_stat64   vals: count:1
```

- No `sys_sendfile`
  - So remove it
- Add `CONFIG_SPLICE`



# Dynamic Memory

- Just as important as static
  - Slab
  - Caches
  - Per-process
- Tools
  - `/proc/meminfo`
  - `/proc/slabinfo`
  - Various tracing tools



# Better Tools

- **Slabinfo** is great, but...
  - No drill-down
- Tracing tools great, but...
  - Don't work early
- **microYocto hash triggers**
  - Key, val any event field
  - 'bucketize' call chains
- e.g. all callers of kmalloc



# Hash Trigger Example – Trace event format

```
root@galileo:/sys/kernel/debug/tracing/events/kmem/kmalloc# cat format
name: kmalloc
ID: 378
format:
    field:unsigned short common_type;          offset:0;      size:2; signed:0;
    field:unsigned char common_flags;          offset:2;      size:1; signed:0;
    field:unsigned char common_preempt_count;  offset:3;      size:1; signed:0;
    field:int common_pid;        offset:4;      size:4; signed:1;

    field:unsigned long call_site;  offset:8;      size:4; signed:0;
    field:const void * ptr;        offset:12;     size:4; signed:0;
    field:size_t bytes_req;      offset:16;     size:4; signed:0;
    field:size_t bytes_alloc;    offset:20;     size:4; signed:0;
    field:gfp_t gfp_flags;       offset:24;     size:4; signed:0;
```

# Hash Trigger Example – Early Call Chains to kmalloc

```
# echo 'hash:stacktrace:bytes_req,bytes_alloc' > /sys/kernel/debug/tracing/events/kmem/kmalloc/trigger  
# cat /sys/kernel/debug/tracing/events/kmem/kmalloc/trigger
```

```
key: stacktrace:  
    __kmalloc+0xb6/0x1a0  
    __proc_create+0x67/0xb0  
    proc_mkdir_data+0x32/0x70  
    proc_mkdir+0x19/0x20  
    proc_tty_init+0x22/0x79  
    proc_root_init+0x5a/0x6d  
    start_kernel+0x2bb/0x2d0  
    i386_start_kernel+0x12e/0x131  
    vals: count:1 bytes_req:82,  bytes_alloc:96
```

```
key: stacktrace:  
    kmem_cache_alloc_trace+0xa1/0x170  
    do_execve_common+0x7f/0x5b0  
    do_execve+0xd/0x10  
    __call_usermodehelper+0x96/0xc0  
    call_helper+0x19/0x20  
    ret_from_kernel_thread+0x1b/0x30  
    vals: count:414 bytes_req:89424,  bytes_alloc:105984
```

Totals:

```
Hits: 11502  
Entries: 2550  
Dropped: 0
```

# Reducing Dynamic Usage

- Basically see what adds slab
- Callchains point to code
  - Reduce callchains
- e.g. CONFIG\_PROC\_MIN
  - 2525->2402 callchains
  - 100k savings
- More analysis work needed



# Interesting kmalloc Data

- Boot assumed not important
- Boot thru `start_kernel()`
  - 2550 callchains
  - 11500 kmallocs
- Kernel compile, mail, web
  - 538 callchains
  - 30 million kmallocs
- Important in callchain terms



## Better Tools (cont'd)

- Normal hash triggers fine
- Early hash triggers a hack
  - TODO early ftrace buffer
  - So can't use generic
  - Need glue code for each
  - Only set one at a time
- Need to set normal hash trigger to print early events



# Early Hash Trigger Example – Real-world Setup and Use

```
title image-micro-test kmalloc-bytes-initonly boot-hashttrigger
root (hd0,0)
kernel /bzImage-test root=/dev/ram0 console=ttyS1,115200n8
trace_event_hashtiggers=kmem:kmalloc:hash:stacktrace:bytes_req.override,bytes_alloc.override:initonly

# echo 'hash:stacktrace:bytes_req,bytes_alloc' > /sys/kernel/debug/tracing/events/kmem/kmalloc/trigger
# cat /sys/kernel/debug/tracing/events/kmem/kmalloc/trigger

hash:unlimited
Early kmem:kmalloc events:
key: stacktrace:
...
    vals: count:337 bytes_req:9007,  bytes_alloc:12592
Totals:
  Hits: 11678
  Entries: 2561
  Dropped: 0

key: stacktrace:
...
    vals: count:20 bytes_alloc:327680
Totals:
  Hits: 77
  Entries: 15
  Dropped: 0
```

# Yocto in a Nutshell

- **Builds custom systems**
  - Kernel + filesystem
- **'recipes':**
  - What and how to build
- **Output:**
  - Packages (rpm, ipk, etc.)
  - Image a set of packages
- **'captures' the code and config**



# Yocto Kernel in a Nutshell

- An upstream kernel plus:
  - 'patches' (git commits)
    - 'machine' branch
  - config fragments
    - 'meta' branch
- Recipes toggle fragments
- Machine + meta + recipes =>
  - /linux + .config



# Example linux-yocto-micro-3.14 config Fragment

- **cfg/proc-min-enable.cfg in the 'meta' branch:**

```
# CONFIG_PROC_FS is not set
CONFIG_PROC_FS_MIN=y
```

- **The code in the 'machine' branch (standard/micro/galileo):**

```
commit 68379432afcfa82ac695d9f02892fcf48ade5ae8
Author: Tom Zanussi <tom.zanussi@linux.intel.com>
Date:   Tue Apr  8 16:23:28 2014 -0500

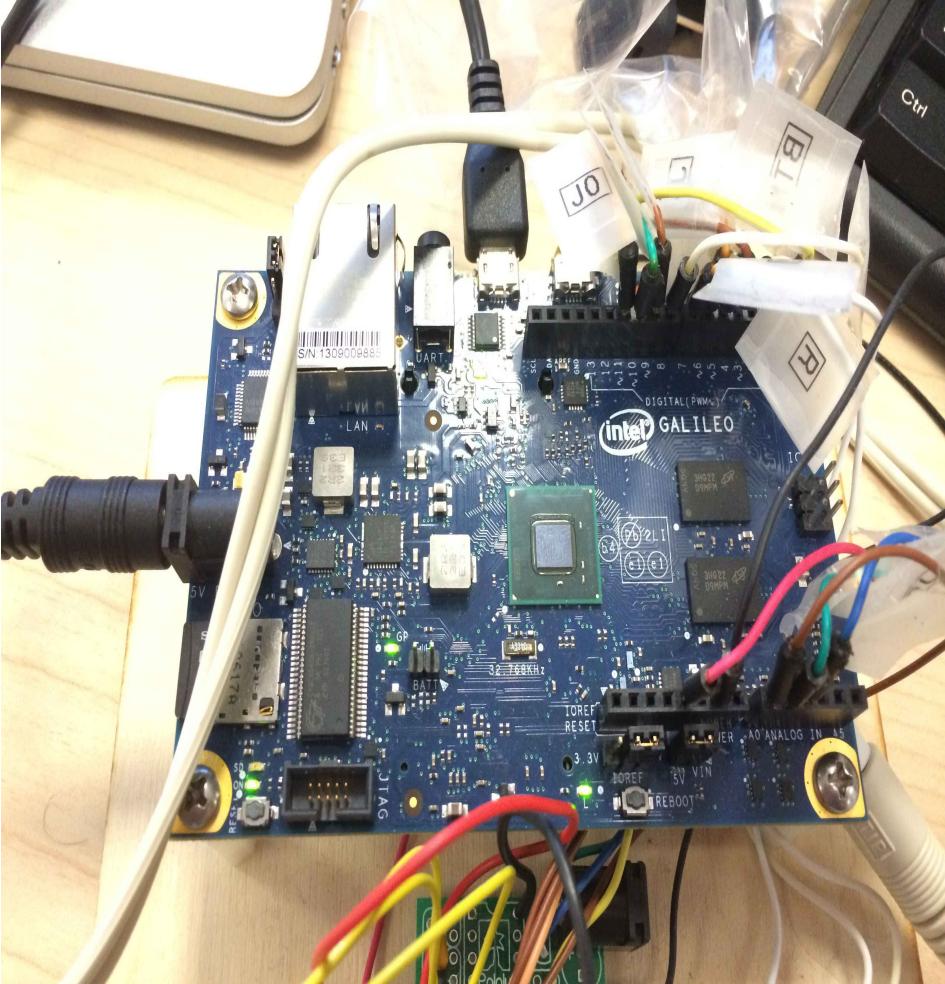
    proc: Add CONFIG_PROC_FS_MIN code
```

- **Toggle in the kernel recipe (linux-yocto-micro\_3.14.bbappend):**

```
KERNEL_FEATURES_append_galileo += "cfg/proc-min-enable"
```

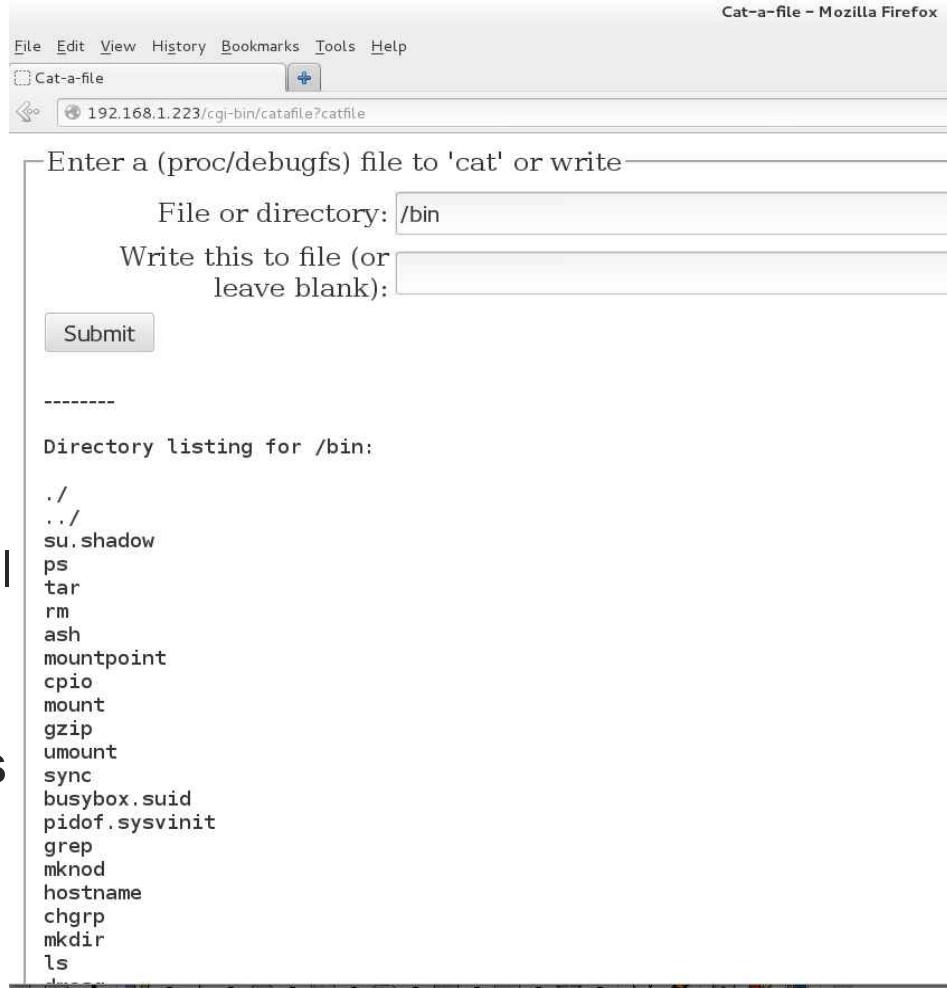
# Build and Run on Galileo

- See README in meta-galileo
- Run 'clone-galileo mygalileo'
  - Clones and checks out
- Run 'bitbake image-micro'
  - Builds the image
- Run 'wic create...'
  - Creates sdcard image
- 'dd' image and boot



# Log in and Try Web Server

- IP address is 192.168.1.223
- Ssh in or use serial
  - Login as root w/ empty pw
  - Busybox shell and utils
- 'cgi-bin/catafile' is a simple CGI
  - Written in C
  - Lets you 'cat' files, list dirs
  - Also write to files
  - So you don't need shell



# Future

- More cutting
- XIP kernel
- Easier-to-use kernel config
- Single-purpose images
  - 'single' image feature
    - Single app as init
- Trace-Guided Optimization
  - Remove all untraced



# Acknowledgements

- Special thanks to:
  - Andi Kleen
  - Alan Cox
  - Josh Triplett



# Questions?

- **Code:**

```
$ wget  
https://github.com/tzanussi/meta-  
galileo/raw/daisy/meta-galileo/README
```
- **Build problems? Please try  
'Yocto Quickstart' first:**  
[www.yoctoproject.org/documentation](http://www.yoctoproject.org/documentation)
- **Questions/comments:**  
[tom.zanussi@linux.intel.com](mailto:tom.zanussi@linux.intel.com)



# Image Attributions

"Tiny frog" by Jason Ahrns

"Tiny book" by Amelia Schmidt

"Static Electricity Ball at the St. Louis Magic House" by Adrienne May

"Anchor Men - Mauretania" by Tyne & Wear Archives & Museums

"Pure Network" by Michael Heiss

"Lone Island @ Night" by Patricio Barragán

"BADGE - Canada - ON - City of London Police shield (very old)"  
by Dave Conner

"This is the spinning machine that sets the fish" by ultraBobban

"Day 98 - Static" by Nicholas Wilson

"Dimensions" by Steve Jurvetson

"Queen Anne high school" by angus mcdiarmid

"We've picked something up on radar!" by Don

"watercolor'd laser etched yo-yo" by Doctor Popular

"Work Tools." by Matteo Paciotti

"Tomorrowland" by Rob Young

"dorothy and joan beach exercise" by Allison Marchant

"standing on the shoulders of giants" by Klaus M

"Question!" by Stefan Baudy

"Frozen pea in a shell" by Uwe Hermann

"Golden Groudnuts" by Thanashyam Raj

no title by Send me adrift.

"North Side Empty" by Phil Gibbs

# Backup