Tools for microcontroller development

https://mchck.org

Simon Schubert, Sternlabs
<simon@sternlabs.org>

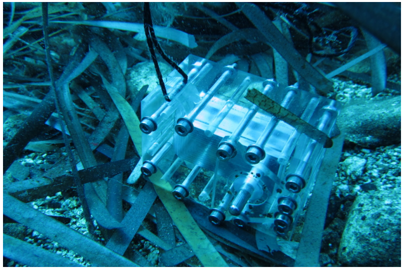
We need better tools

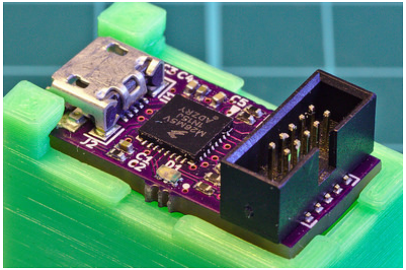
- Embedded has traditionally been a copy+paste fest
- Copy+Paste prevents downstreaming of fixes
- Where is the bug: in the copied code, in my code, in the interface?

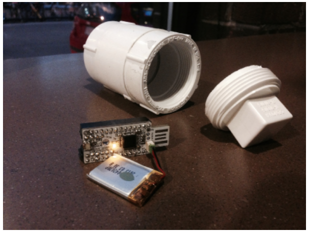
Experiment: The MC HCK

How much better can we do, if we focus on tools, rather than projects?

Hey, this is even being used!







Software: A Complete Stack

- Own Library OS
- SWD Programmer & Debugger
- Supporting Toolchain

The MC HCK philosophy

Do things properly, even if it means doing them yourself.

Library OS (1)

- Embedded (vendor) code is terrible
- Typically exposes interface of hardware to software

MC HCK library

- Develop library from scratch in proper OS style
- Implement a task-centric API
- Completely non-blocking; callback based

Library OS (2)

blink.c

```c
#include <mchck.h>

static struct timeout_ctx t;

static void
blink(void *data)
{
        onboard_led(ONBOARD_LED_TOGGLE);
        timeout_add(&t, 500, blink, NULL);
}

int
main(void)
{
        timeout_init();
        /* blink will also setup a timer to itself */
        blink(NULL);
        sys_yield_for_frogs();
}
```

Library OS (3)

lib/mchck/adc.c

```c
void
adc_init(void)
{
        /**
         * Enable bandgap buffer.  We need this later to calibrate our
         * reference scale.  However, we start it now, so that it will
         * have time to stabilize. */
        bf_set_reg(PMC_REGSC, PMC_REGSC_BGBE, 1);

        /* enable clock */
        bf_set_reg(SIM_SCGC6, SIM_SCGC6_ADC0, 1);

        /* enable interrupt handler */
        int_enable(IRQ_ADC0);

        /* setup ADC calibration */
        adc_sample_prepare(ADC_MODE_SAMPLE_LONG | ADC_AVG_32);
        adc_ctx.stat_a.cb = adc_calibrate_cb;
        adc_ctx.stat_a.active = 1;

```

SWD Programmer & Debugger

- Poor SWD support in Free Software
- OpenOCD is spaghetti and JTAG centric
- Commercial solutions are expensive *and* unreliable

SWD Programmer & Debugger

- Flash programmer & GDB stub
- Easy to adapt for new targets or programmer hardware (≈100 LOC)

Simple Makefiles

- Embedded build systems: usually old, shitty Makefiles recycled over and over

BSD-style Makefiles

- Semantic declaration
- Complexity hidden centrally

blink/Makefile

```
PROG=	blink

include ../../build/mchck.mk
```

Link & Compile only required sources

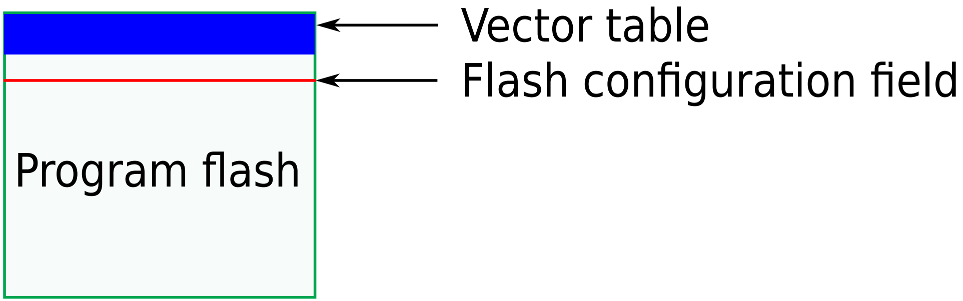
- Common approach: Fuzz your Makefile until you figured out all required sources.
- Typically systems link with `-ffunction-sections -fdata-sections`
- Cannot rely on linker pruning alone: weak symbols get linked.

Linkdep

- Compile source once
- Observe which symbols are provided and required
- Make only links required objects

Knapsack the Flash

- Kinetis has flash protection bits in flash at offset 0x400.



- The linker cannot just fill sections into a gap.

Knapsack

- A linker wrapper to fill the gap efficiently.

USB Descriptor Generator

- USB descriptors are tedious and difficult to keep coherent.

Descriptor Generator

- A small DSL for USB descriptors

usb-serial-loopback.desc

```
device(:cdc_device) {
  idVendor 0x2323
  idProduct 3
  iManufacturer "mchck.org"
  iProduct "MC HCK serial test"

  config {
    initfun :init_vcdc

    cdc {
    }
  }
```

Virtual USB to debug class drivers

- USB has timing constraints: difficult to debug
- Better run USB class drivers unmodified on host

VUSB

- Uses Linux USBIP module to connect to host USB subsystem
- Emulates USB host adapter + USB SIE
- currently broken :/