Intro to Cortex M0 and LPCxpresso 1114
Minute Quiz

- Just kidding, but...

- What are the steps to go from source code to an executable?
  - Compile
  - Link
What makes "gcc" different from "arm-none-eabi-gcc"?

What does the "-mach" flag do to gcc?

All ARM assembly is the same, right?

- So why can't I run code for the SmartFusion on my ARM-based Android phone?
What is linking?

We added this to every SoftConsole, what is it?

- `-T ../CMSIS/startup_gcc/debug-in-actel-smartfusion-esram.ld`
Linking

- What is linking?
- We added this to every SoftConsole, what is it?
  - `-T ../CMSIS/startup_gcc/debug-in-actel-smartfusion-esram.ld`
Static vs Dynamic Linking

- What does "#include <stdlib.h>" do?
  - Compiler?
  - Linker?
- Where is stdlib.o?
  - On your laptop
    - /usr/lib/libc.so
  - On your SmartFusion?
Vocabulary

- Cross-compiler
  - arm-none-eabi-gcc
  - arm-none-eabi-clang

- Toolchain
  - compiler + linker + supporting scripts / configuration
Background

- ARM
  - Cortex M0
    - LPCxpresso 114
      - Peripherals
      - LPCxpresso (development suite)
      - CooCox (OS)
      - TinyOS
32-bit RISC ISA

- With 16-bit subset: Thumb

Simple, efficient cores

Dominant in mobile / embedded space

Trivia (ARM vs. ARM Holdings)

- The acronym ARM originally stood for Acorn RISC Machine. The company name ARM stands for Advanced RISC Machines. This name was changed, around the time of the IPO, to "ARM Holdings", since it was felt the term RISC, which indicates a type of CPU design, being phonetically identical to "risk," would deter people unfamiliar with computers. [wikipedia]
ARM: Cortex Family

- Cortex == ARMv7
- 3 "Families”
  - Cortex-A: Applications
    - Smartphones, etc
  - Cortex-R: Real-Time
  - Cortex-M: Microcontrollers
ARM: Cortex M Family

- Cortex-M4
  - ARMv7-ME
    - Thumb, Thumb2, FPU. Hardware MAC, SIMD, and divide
- Cortex-M3
  - ARMv7-M
    - Thumb2, hardware divide
- Cortex-M0
  - ARMv6-M
    - Thumb2 subset (16-bit Thumb instructions & BL, MRS, MSR, ISB, DSB, and DMB)
Cortex M0

- Simplest, smallest "current generation" ARM
- 85 µW/MHz (0.085 milliWatt)
- 12K gates
- Only 56 instructions
  - Subset of M3/M4; Thumb and some Thumb2
- 3-stage pipeline
- Interrupts: NMI + 1-32 physical interrupts
  - 16 cycle latency
- Complex Hardware Ops
  - Single-Cycle 32x32 multiply
LPCxpresso 1114 Specs

- Cortex-M0 @ 50 Mhz (max)
- 32kB Flash
- 8kB RAM
- 12.000 MHz clock crystal
- Timers:
  - 4 capture inputs, 13 match outputs
  - Two 32-bit counter/timers
  - Two 16-bit counter/timers
  - One Programmable Watchdog timer
LPCxpresso 1114 Specs

- Clocks:
  - 12 MHz RC oscillator, 1% accuracy
  - Crystal operator, ranged 1-25MHz
    - PLL allows CPU frequency up to 50MHz
  - Watchdog oscillator, ranged 7.8KHz-1.8MHz
  - Clock output with divider can source any clock
LPCxpresso 1114 Specs

- 42 GPIOs with configurable pull-up/down resistors
- Any GPIO usable as edge/level triggered interrupt
- High-current output driver (20 mA) on one pin.
- High-current sink drivers (20 mA) on two I2C-bus pins in Fast-mode Plus.
- Four general purpose counter/timers with a total of four capture inputs and 13 match outputs.
LPCxpresso 1114 Specs

- **Analog:**
  - 10-bit ADC
    - single 10-bit successive approximation ADC with eight channels
    - Measurement range 0 V to VDD.
    - 10-bit conversion time ≥ 2.44 μs.

- **Serial:**
  - UART
  - 1 SPI with SSP support
  - I2C with FastMode (up to 1 Mbit/s)
LPCxpresso 1114

Diagram showing the connection between LPC-LINK and LPC1114 Target, with various power and interface connections. The diagram indicates USB powering, LPC3154 internal powering, JTAG interface, and LED indicators on GPIO2 and PIO0_7. Further information is provided on schematic pages 2-3, 4, and 5.
Let there be Light!

- (emitted in a blinking pattern from diodes)
- Blinky
  - Set a 10ms timer
    ```c
    void TIMER32_0_IRQHandler(void)
    {
      LPC_TMR32B0->IR = 1; /* clear interrupt flag */
      timer32_0_counter++;
      return;
    }
    ```
  And loop forever:
  ```c
  while (1) {
    /* Each time we wake up... */
    /* Check TimeTick to see whether to set or clear the LED I/O pin */
    if ( (timer32_0_counter%LED_TOGGLE_TICKS) < (LED_TOGGLE_TICKS/2) )
    {
      GPIOSetValue( LED_PORT, LED_BIT, LED_OFF );
    } else {
      GPIOSetValue( LED_PORT, LED_BIT, LED_ON );
    }
    /* Go to sleep to save power between timer interrupts */
    __WFI();
  }
  ```
OS'es for Cortex M0

- Linux Kernel?
- make allnoconfig
  - With some editing to target the closest NXP board
  - And LZMA (best, slowest) compression...
- ls -lh
  - 943K Image
  - 343K zImage
- 32k of flash...
TinyOS

- Event-driven, non-preemptable
  - Except for the thread library
- More extensive networking stack
- Doesn't support M0 'out of the box'
  - "There is work underway to support the Cortex M3"
CooCox CoOS

- "CooCox CoOS is an embedded real-time multi-task OS specially for ARM Cortex M series."
  - Scalable, minimum system kernel is only 974Bytes
  - Supports preemptive priority and round-robin
  - Interrupt latency is 0
  - Stack overflow detection option
  - Semaphore, Mutex, Flag, Mailbox and Queue for communication & synchronisation

http://www.coocox.org/CoOS.htm
Programming the 1114 (on Linux)

- LPCxpresso tools installed and working
  - Build, run, debug, etc
- CoIDE and tools installed and mostly working
  - Everything except CoFlash works
  - But can flash from command line with LPC tools...
    - `crt_emu_lpc11_13_nxp -g -mi -2 -pLPC1114/301 -wire=winUSB -flash-load-exec=blinky.axf`

- Demo IDEs
Questions?