Starting the ARM Cortex-M0: What’s Really Happening Here?

Thursday, September 5, 2013

ECEN-3000
MCU vs. PC Development

- **PC** (You develop on the same machine that is running your new code)
  - Environment: From text editor & gcc/g++ to full-blown IDE which **leverages a shared operating system**
  - Debugging: Console, breakpoints, memory scope, etc.

- **MCU** (Your code executes on a physically separate machine from your development machine)
  - Environment: Full-blown IDE (e.g. LPCXpresso) required, **MCU often has no operating system—it’s just your code!**
  - Debugging: Console, breakpoints, memory scope, **in addition, physical observation, LEDs, logic analyzer, oscilloscope, etc.**
  - Interface: SWD, JTAG, etc.
MCU Development Flow

Evaluate

Explore

Develop

Any Custom Board with a JTAG Connector
Architecture: Cortex-M0
JTAG: “Joint Test Action Group”

- **TDI** (Test Data In)
- **TDO** (Test Data Out)
- **TCK** (Test Clock)
- **TMS** (Test Mode Select)
- **TRST** (Test Reset) optional
C & Assembly Compilation Flow

- C files (.c)
- Linker script
- Memory layout
- Assembly files (.s)
- Object files (.o)
- Object files (.o)
- Linker (.ld)
- (linker)
- gcc (compile + link)
- Executable image file
- Binary program image (.bin)
- objcopy
- objdump
- Disassembled code (.txt)
Bootloader

- **Definition**: Fixed ROM, defines how your program is to be written to memory (+ some additional low-level housekeeping)
- **Adds Flexibility**: e.g. how to execute the code, i.e. at what mem address does application code live, programming from alternative sources, e.g. over Bluetooth
- **Optional**: the ARM core might instead execute your code from the start
Reset Handler

- What caused the reset? (e.g. power-on, reset pin, GPIO pin, battery brown-out, etc.)
  - Depending on the answer, the application may behave differently
C Startup Code

- **HW Initialization**: e.g. clock selection, PLL, etc.
- **Initialization of variables and memory addresses needed by the application**: CMSIS vars, stack-size/address, heap-size/address, Reset/HardFault/ISR Handlers
• main.c:
  – High-level Initialization: Peripherals (GPIO/serial comms/timers/etc.), NVIC reconfiguration
  – while(1) { <application operational code here!> }