Microcontroller Basics

Gabe Cohn

CSE 599U – February 8, 2010
Outline

• Overview of Embedded Systems
• What is a Microcontroller?
• Microcontroller Features
• Common Microcontrollers
• Choosing a Microcontroller
• Development Kits
Overview of Embedded Systems

- Minimal computation, simple software (no/simple OS)
- Low power (typically battery powered)
- Event Driven Design
- Mostly IO (Inputs and Outputs)
  - Sensors, Switches, Keypad
  - Displays, LEDs
  - Actuators, Servos
  - Data communication (wired or wireless)
- Data Conversion
  - Analog-to-Digital, Digital-to-Analog
Generic Embedded Systems

**INPUTS**
- Switches
- Keypad
- Sensors
- RF Rx
- Digital Data USB Serial Parallel

**CPU**
- Clock Logic
- Program Memory
- Timers

**OUTPUTS**
- LEDs Display Actuator Servo
- RF Tx
- Digital Data USB Serial Parallel

**INTEGRATION**
- ADC
- DAC
- Buffers
- Reset Logic
- Data Memory
- Peripherals
Example: Using Discrete Components
What is a Microcontroller?
Example: Using A Microcontroller
Microcontroller Features

• CPU
• Program and Data Memory (ROM and RAM)
• Reset and Oscillator Circuitry
• Timers
• Data Converters (ADC, DAC)
• Buffered GPIOs
• Simple Peripheral Interface
• Reduced System Size, Complexity and Cost
CPU

• RISC
• Harvard Architecture (separate program and data memory)
• Pipelined Load-Store Architecture
• Lower clock speeds (8-32 MHz)
• Optimized for low-level compilers like C
• Typically no OS is used (sometimes RTOS)
Memory

• On-chip RAM and ROM
• No external access to address and data buses
• Need a “programmer” to program the code into the ROM (typically Flash these days)
• Size range 10s of bytes to 100s of KB
  – main difference in price between similar products
Reset and Oscillator Circuitry

• Reset
  – Internal or External
  – Watchdog Timer (WDT)
  – Brownout Reset (BOR)

• Oscillator
  – Several sources to choose from
  – Internal or External
  – PLL and clock frequency adjustment
Timers

• Typically several different timers
  – Real-Time Clock (RTC)
  – Watchdog Timer (WDT)
  – PWM output common

• Several clocking options

• Event based notification (Interrupts)
  – Allows CPU to focus on foreground tasks
Data Converters

• ADC (very common)
  – Several channels
  – Multiple clocking options
  – Several different types
  – Comparators and other analog circuitry
  – Important for ratiometric sensors

• DAC (less common)
  – Several different types
GPIOs

• Many General Purpose Analog/Digital IOs
  – Buffered to drive typical embedded loads
  – Multiplexed for several functions
  – Switchable internal pull-up, latch, etc...
  – Edge detection
  – Schmitt trigger inputs on some
  – main difference in price between similar products
Simple Peripheral Interface

- Serial (Sync/Async, SPI, I²C)
- CAN bus (automotive)
- LED and LCD controllers
- Ethernet, USB, and Video controllers
- DMA, DRAM, SDRAM controllers
- Host Processor Interface, External Memory Bus
## Common Microcontrollers

<table>
<thead>
<tr>
<th>Family</th>
<th>Manufacturer</th>
<th>Word Size*</th>
<th>Common Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM</td>
<td>Various</td>
<td>32-bit</td>
<td>Consumer Electronics</td>
</tr>
<tr>
<td>AVR</td>
<td>Atmel</td>
<td>8-bit</td>
<td></td>
</tr>
<tr>
<td>PIC</td>
<td>Microchip</td>
<td>8-bit</td>
<td>Hobbyist</td>
</tr>
<tr>
<td>MSP430</td>
<td>TI</td>
<td>16-bit</td>
<td>Low Power</td>
</tr>
<tr>
<td>6805</td>
<td>Motorola/Freescale</td>
<td>8-bit</td>
<td>Automotive</td>
</tr>
</tbody>
</table>

* Many of these µCs now come in wider bus architectures as well
Choosing a Microcontroller

- All very similar, so stick with a family you know
- Required features
- Required number of GPIOs
- Memory requirements
- Availability of programmer (USB?)
- Availability of a good C compiler
- Packaging
Development Kits

- Fast and Easy!
- Everything you need is included
  - uC, power supply, USB connection, simple IO
- Low level code is written for you!
  - port_write(addr, value)
- Example code and projects
- Often large online forums for support

Example: Arduino (AVR based kit)