These are the course notes to accompany the UCSD Extension class: Embedded Controller Programming 1: *Assembly Language Programming*

**FIRST** send e-mail listing any e-mail addresses you would like to have class notices sent to:
- ecp1@hte.com

**NEXT** send an e-mail message to subscribe to the class discussion group by sending a message to: ucsdecp-request@luisa.hte.com with subject = subscribe

The class e-mail will consist of updates between meetings, Q&A, important notices, and interaction with the instructor and other students between classes.

Course web site: [http://www.hte.com/uconline/ecp](http://www.hte.com/uconline/ecp)


Ken Arnold  ecp1@hte.com
phone 858-679-1569 or 679-4821 (office)
phone 858-335-9361 (mobile)
fax 858-679-7569
HiTech
14260 Garden Road, Suite 2A
Poway, CA 92064-4972
Welcome to class!

- ECP 1 and 2 Course Overview
- Instructor & Student Introductions
- Processor Architecture
- Basic Instruction Set
- Introduction to the SDK
- Homework #1

Overview - ECP 1

- uC Architecture and Programming
- Assembly Language Programming
- 8051 Memory Model and Memory Usage
- 8051 Instruction Set
- 8051 Hardware Features
- Other Topics
Overview - ECP 2

- Focus on Applications
- Learning to use C for uC Programming
- Adapting C to the uC environment
- Focusing on Modular Programming
- Handling Basic Peripherals
  - Displays
  - Switches/Keypads
  - Motors/Controls
- Student Projects!

Administrative Stuff

- Fill Out Student Forms Please!
- Send e-mail to ecp1@hte.com
- Course Format, Policy
  - Lecture, Demo, Homework, Project
- Class Web Page:
  - http://www.hte.com/uconline/ecp
- Grading
  - 4 Homework Problems - Due Week after Assigned!
  - Programming Project, Comprehensive Final Exam
  - Be Here So We Can Start (and Finish) On Time!
Course Objectives

- Microcontroller vs. Desktop PC
- Familiarity with Hardware
- Familiarity with Tools
- Hands-on Exposure Required
- Low Level Programming, Interfacing
- Microcontroller Applications
- Polite, Invisible computing!

Course Format

- In-class:
  - Lecture and demonstrations
  - 3 hours * 6 meetings
  - Please Ask Questions!!

- Outside of Class:
  - Software Development Kit (SDK),
  - Development Setup:
    - SDK, Prototyping, and Test Equipment
Resources

- Software Tools
  - Assembler
- Hardware
  - SDK, Prototyping Board, Components
- Support Web Sites
  - http://www.hte.com/uconline
- General Information

The Toaster Fable

- The King Wants a New Toaster
  - The Pragmatic Solution
  - The Politically Correct Solution
  - The King Resolves the Conflict
- Moral:
  - CS emphasizes most general solution
  - EE emphasizes minimizing complexity
Instructor

- Father of 5, age 6 to 24
  - Preemptive Multi-tasking, Dynamic Priorities!
- Wireless Innovation
  - Product Development and Manufacturing
- UCSD Extended Studies
  - Embedded Certificate Programs

Student Introduction

- Your Name and Background

- What Do You Do?
  - (i.e. - EE at XYZ Corp., etc..)

- What Do You Want to Get out of This Class?
What Is a Microcontroller?

- What Are They?
- How Are They Used?
- Basic Features
- Sizes
- Families
- uC’s vs uP’s, DSP’s, PLD’s

µ Controller vs. µProcessor

- µC Chip Includes:
  - Central Processor
  - Program Memory
  - Data Memory
  - I/O
  - Highly Integrated
  - Low Cost
  - Specialized Architectures

- µP Chip Includes:
  - Central Processor
  - Separate Chips for:
    - Central Processor
    - Program Memory
    - Data Memory
    - I/O
  - Highest Performance
  - Highest Cost!
Von Neumann Architecture

- Single Memory for:
  - Programs
  - Data
- Familiar
- Most Flexible
- Used in PCs
- Speed Bottleneck:
  - Memory Interface

Harvard Architecture

- Separate Memory for:
  - Programs
  - Data
- Advantages:
  - Faster
  - Overlap Transfers
    - Instruction Fetch
    - Data Transfer
  - Can't execute Data!
Bus Oriented Microcomputer

Microcontroller Functions

Microprocessor Functions

CPU Memory I/O

Peripheral Devices

The Real World

One Chip Microcontrollers

Advantages:
- Fewer chips required
- Lower cost and smaller
- Lower power
- Fewer connections
- More user I/O pins
- Reliability is higher
- K.I.S.S.!

Disadvantages:
- Reduced flexibility
- Expansion is limited
- Limited performance
- Limited I/O
- Design compromised to fit everything on one chip
In order to get into details, we must look into a specific processor architecture. What criteria were used to decide which architecture to use for this course?

- A real device, in common use
- Availability from multiple manufacturers
- Free and very low cost development tools available for student use
The 8051 Microcontroller

- The most widely used microcontroller
- Multiple sources, hundreds of variants
- Free software development tools
  - Assembler
  - Simulator
  - C Compilers
- Low cost hardware components and tools

The 8051 Family

- Originally Designed by Intel
- Introduced in 1980
- PCs Shipped in the Millions per Year
- Billions of 8051s shipped in one year
  - 1B Sold just by one of the many manufacturers
- New Variants Come out All the Time
- Other Microcontrollers Ship in Billions/Year
Partial List of 8051 Vendors

- Intel
  - The Original 8051
  - The 80251 Family
- Philips
  - 8051 Family Variants
  - The 8051XA Family
- Atmel
  - 20 Pin 89Cxx51 Family
  - 8051 Family Variants
- Dallas Semiconductor
  - High Speed Versions
  - Non Volatile SRAM
- Analog Devices, Cygnal
- Temic, ISSI, Matra, OKI, Siemens, SMC, SSI - and Many Others Too !!!
- IP Cores for ASICs
  - Synopsis
  - Mentor

Embedded Memory

- Semiconductor Storage
- Implications of Storage Technology
  - Matching Technology to Application
  - Read-Write, Read-Only, Read-Mostly
- Non-ideal Memory Characteristics
  - Asymmetrical Read, Write
**Memory Volatility**

- **Volatile:**
  - Loses contents when power is removed
  - Used for temporary storage of changing values:
    - Variables
    - Stacks

- **Non-Volatile:**
  - Retains contents after power loss
  - Used for permanent storage of:
    - Programs
    - Constants
    - Look-up Tables

---

**8051 Memory Architecture**

- Separate Memory Address Spaces for:
  - Programs - Non-volatile
    - Internal ROM
    - External EPROM
  - Data - Volatile
    - External SRAM
    - Internal RAM
      - General Purpose Registers
      - Bit Addressable Registers
      - Special Function Registers
8051 Instruction Set

- Instructions:
  - Data Transfer
  - Arithmetic
  - Logical
  - Control
- Address Modes
  - Immediate
  - Direct
  - Indirect
- Examples:
  - MOV A, 90h
  - ADD A, #30h
  - ANL A, #0FEh
  - CALL subroutine
- Examples:
  - MOV A, #30h
  - MOV A, 30h
  - MOV A, @R0

8xC52 Program Memory

- On-chip Code Memory
  - Non-volatile
  - Different types:
    - 80C52 = Mask ROM
    - 87C52 = EPROM
    - 89C52 = Flash EPROM
- External Code Memory
  - Design Dependent
  - SDK has EPROM & SRAM

Reset vector: 0000
**Program Memory Usage**

- Processor execution
  - Begins at location 0000h
    - (The “Reset Vector”)
  - Continues with next instruction
- 8x52 has 8K bytes of Internal Code Memory on-chip
- 8x32 has NO code space on chip
- When Processor fetches external instructions, /PSEN pulses low

**SDK Program Memory Map**

- ROM: 0000-3FFFFh
  - Monitor Program
- RAM: 4000-
  - User Program/XData
    - 4000-BDFFh
    - User Programs and Data
  - Monitor Data
    - Reserved: BE00-BFFFFh
    - DO NOT Modify!
    - Temporary Storage
**Data Memory**

- External SRAM
  - External chip
  - Typically SRAM
- Internal RAM
  - General Purpose Registers
  - Bit Addressable Registers
  - Special Function Registers

**Data Memory Addresses**

- External Data 0000-FFFFh
- Internal Data 00-FFh
  - General Purpose 00-7Fh
    - G.P. Register Banks
    - Bit Addressable
    - General Purpose “Scratch Pad”
  - Indirect Access 80-FFh
    - Use MOV @R0 or @R1
  - SFRs Direct 80-FFh
    - Overlapped addresses
    - Direct address MOV 00-FFh
    - Special Internal Registers
Internal Data Memory

- Registers R0..7
  - 8 registers per bank
  - 4 Banks available
- Bit Addressable
- General Purpose
- Special Function Registers (SFRs)
  - Accumulator, I/O
  - Timers, misc. registers

Bit Addressable Memory

- Internal Data Memory
  - Byte Addresses 20-2Fh
  - Bit Addresses 00-7Fh
- Allows individual bit operations:
  - MOV bits to/from Carry
  - SETB sets a bit to 1
  - CLR clears a bit to 0
  - JB conditional jump
8051 Instruction Set

- Instructions:
  - Data Transfer
  - Arithmetic
  - Logical
  - Control

- Address Modes
  - Immediate
  - Direct
  - Indirect

Examples:
- MOV A, 90h
- ADD A, #30h
- ANL A, #0FEh
- CALL subroutine

Examples:
- MOV A, #30h
- MOV A, 30h
- MOV A, @R0

Simple 8051 Block Diagram
Development Tools

Software
- Translators
  - Assemblers
  - Compilers
- Linkers
- Debug Monitor
  - SDK Monitor ROM
- Performance Analyzers
  - Find Execution Bottlenecks

Hardware
- In-Circuit Emulators
  - Substitutes for CPU chip
  - Allows seeing “inside” uC
- Logic Analyzer
  - View Timing and Bus Cycles
- Logic Probe
- Oscilloscope
- Lights and Bepers

Hardware Handling Issues

Precautions Regarding:
- Take Precautions against ESD
- Avoid Touching Contacts: Metal Oxidation
- Power = heat and smoke

But Don’t Be Afraid!!
- SDKs are easy to fix and connect to probes
- Sockets for all ICs, and are easily replaced
- Unlike surface mounted components!
Introduction to the SDK

- Software Development Kit Connections
  - Power, Serial Port, ICE Cable
  - SDK Users manual: sdk31man.pdf
  - SDK Schematic: sdk31sch.pdf
- PC Setup, Software Setup
- Common Problems

PC Software for SDK

- Hyperterminal, MTTY, or Procomm
  - Terminal Emulator to connect to SDK
  - Command line monitor ROM on SDK
- asm51 8051 Cross assembler translates
  - Input, 8051 source: *.asm
  - Output, Intel Hex object format: *.hex
  - and listing file: *.lst
SDK Introduction

- Demonstrate System Setup
- Introduce SDK Operation
- Code Development Cycle
- Download and Test

SDK - Top View
Demonstration of SDK

- Connecting the SDK
- Editing “Hello World” program
- Assembling program
- Downloading Hex file to SDK
- Running the modified SDK program

SDK Power Connection
Summary

- Introduction
- Microcontroller Architecture
- Memory
- Instruction Set Intro
- Development Tools
- SDK Intro

Homework Assignment

- Setup SDK and Software
- Modify “Hello World” Program
  - Change output string to Hello <your name>
  - Optional: Try other changes...
    - Increment Port 1 Outputs
    - Blink an LED!
    - Echo characters
    - Play!!!
## References

- SDK User's Manual
- Arnold, “Embedded Controller Hardware Design”
- Ayala, “The 8051 Microcontroller”
- Cook, "A First Course in Digital Electronics"
- Foster, “Real Time Programming”
- Horowitz & Hill, “The Art of Electronics”
- Wakerly, “Digital Design”
- Schultz, “C and the 8051” vol I and II