CEG453/653 Embedded Systems

- Computer : Processor, Memory, I/O, Bus
- Embedded Systems
  - See textbook p. 3 for a list of applications
  - Challenges: real-time execution, physical size, power consumption, multirate operation, cost, limited memory
- Technology
  - Microprocessor versus microcontroller; Flash, SRAM, DRAM; Sensors, Bluetooth, Ethernet, etc.
  - VLSI, SOC
  - FPGA, Hard/Soft Processor, SOPC
  - RTOS (Real-Time Operating System)
- What is this course about?

- EVB (Evaluation Board): Axiom CSM12C32 (www.axman.com)
- Chip: MC9S12C32 from Freescale (formerly, a Motorola division)
  - For chip documentation in PDF files, see C:\AxIDE3\CSM12C32\FREESCALE\9S12C32_ZIP
- 68HC12 versus 68HCS12

Lab 1 Preparation

- Address versus data
- Hexadecimal number system and Memory map
- Monitor, MON12, Monitor commands : MD, MM, LOAD, CALL
- C, Assembly, Machine Code
- ICC12 and as12
- Step by step instructions
- S record file, list file, map file

Sample C Programs

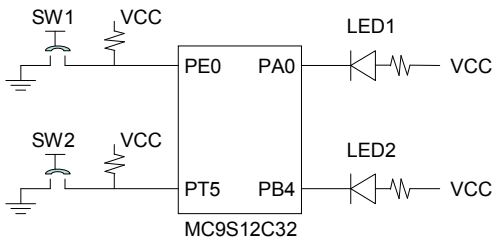
- C review (Chapter 3, sample_c.c)
  - main( ), function call
  - data types (char, unsigned char, int, short, long, float)
  - array and array initialization
  - decimal, hexadecimal constants
  - condition check (true, false, if, else)
  - for loop, while loop
  - putchar(), puts(), printf()

- tone.c : timer, output compare (p.179)
  - Flag: set by hardware, cleared by software
  - Why 4 KHz?
  - Why “Hello World” once every second?
- Various changes to tone.c
  - What if TC0 += 3000;
  - What if TCO += 1500;
  - What if TC0 += 1;
  - How to clear flags slowly (explicitly)?
  - How to produce a square wave of 25% duty cycle? (using O.C. instead of Pulse Width Modulation)

Simple Parallel Port Usage

- Let Y be a generic register which can be PORTA, PORTB, PORTE, PTAD, PTT, PTP, PTS, PTM, DDRA, DDRB, and so on.
- Contents of Y: $Y_7\ Y_6\ Y_5\ Y_4\ Y_3\ Y_2\ Y_1\ Y_0$
- Output (Use Y_1 as an example)
  - Set bits : Y |= 0x02;
  - Clear bits : Y &= ~0x02; or Y &= 0xFD;
  - Toggle bits : Y ^= 0x02;
- Input
  - Check if set : if(Y & 0x02)
  - Check if clear : if(!(Y & 0x02))
  - Wait until set : while(!(Y & 0x02))
  - Wait until clear: while(Y & 0x02)

A simple parallel port usage example: switch_LED.c



Measuring Pulse Width Without Using Input Capture

- Use while loops and read TCNT to get T1, T2
- If(T2 > T1) T = T2 – T1;
  else T = 0xffff – T1 + T2 + 1;
- Timer Overflow?

Interrupt Programming

- Interrupt versus polling
- Interrupt programming
  - Write an ISR (interrupt service routine)
  - Register the ISR (p. 153)
  - Enable ISR (locally and globally)
- tone_interrupt.c example
- On EVB, pressing the reset button clears the MON12 (user) interrupt vector table contents

ADC

- ADC basics, a 2-bit ADC example
  - Analog = V_L + Digital * (V_H-V_L)/(2^N-1)
- ADC Internal: Successive Approximation
- Signal Conditioning Circuits (based on OP Amp)
- 6812 ADC Programming
  - adc_scan.c (see p. 277 for registers)

ADC Timing (ATDCTL4)

- ATD Clock = Bus Clock / [2(PRS+1)]
  - Bus Clock : 24 MHz on EVB
  - PRS : bits 4 to 0 of ATDCTL4
- ATD Conversion Time per Sample =
  [2 + 2(SMP+1) + B] ATD clock cycles
  - SMP : bits 6 and 5 of ATDCTL4
  - B: 8 or 10 (for 8-bit and 10-bit ADC, respectively)

More Timer Functions

- Timer Overflow : TSCR2 bit 7 (TOI), TFLAG2 bit 7(TOF)
- Input Capture (p. 176) and Examples
  - Measure the period of a square wave
  - Very slow square wave => Timer overflow interrupt
- Real-Time Interrupt (Sec. 4.15):
  CRGINT bit 7 (RTIE), CRGFLG bit 7 (RTIF)
  m = RTICTL bits 6-4; n = RTICTL bits 3-0
  OSCCLK (16MHz on EVB)/[(m+1)$2^{(n+9)}$]
- Pulse Accumulator: a 16-bit counter
- Pulse Width Modulation (PWM) Function

- Parallel Ports (Chap. 5), I/O Synchronization and Handshaking
- Serial Interface : SCI (Sec. 4.18) vs. SPI (Sec. 4.19)
  - RS-232, Start-bit, Stop-bit, Parity
- SCI Registers
  - SCIBDH, SCIBDL
  - SCICR1, SCICR2
  - SCISR1, SCISR2
  - SCIDRH, SCIDRL

- Hardware Design Issues
  - Memory : Flash, SRAM, DRAM, Memory configuration
  - CPU Memory/IO Interface (Address Decoding)
  - Noise Consideration (Sec. 6.3)
  - Power Management (Sec. 6.6)
- Embedded System Examples (Chap. 7)
  - Wall-Following Mobile Robot System
  - Motor Speed Control with Optical Tachometer
  - Flying Robot

Chap. 8 RTOS

- An OS that handles multiple tasks in a timely manner
- RT Systems (p. 513): when missing deadlines
  - Hard (leads to a system failure)
  - Firm (a low occurrence can be tolerated)
  - Soft (leads only to performance degradation)
- RTOS
  - Hard RTOS: guarantee to meet deadlines
  - Soft RTOS: meet deadlines a percentage (say, 90%) of the time

RTOS Basics

- Kernel: task scheduling, dispatching, and inter-task communication
- Task States (p. 544 and p. 562): Dormant, Ready, Active, Waiting (for some time), Suspended (for resources), Rescheduled (until reschedule interval is expired) → multiple linked lists
- Task Control Block (p. 548)
- Task Partitioning (p.550): insert break points

Multitasking without a Commercial RTOS

- Round-Robin: similar to polled loop; tasks are allowed to run to completion; may use time-slicing (Special case: Polled Loop, poll inputs one by one)
- Round-Robin with interrupts (Hybrid): interrupts (foreground) for time-sensitive tasks, round-robin for mundane tasks (background)
- Interrupt-Driven: all tasks are inside ISRs

Types of (Commercial) RTOS

- Cooperative Multitasking
  - Priority based
  - Tasks voluntarily relinquish control back to the OS.
  - A low-priority task may never get processor time.
  - Eg. Salvo
- Preemptive Multitasking
  - Priority based
  - A lower priority task is preempted by a higher priority one.
  - Eg. VxWorks

RTOS Issues

- Concurrency: prevent simultaneous access to critical resources by (1) disabling interrupts, (2) employing semaphores or locks
- Reentrancy: create a reentrant function by (1) disabling interrupts, (2) using local variables and CPU registers (instead of using global variables)
- Inter-task Communication: mailbox (a shared memory location, with its key provided to one task at a time), message passing
- Fail-Safe Operation: in the event of system failure, enter a safe condition (e.g, red lights for a failed traffic light control system)