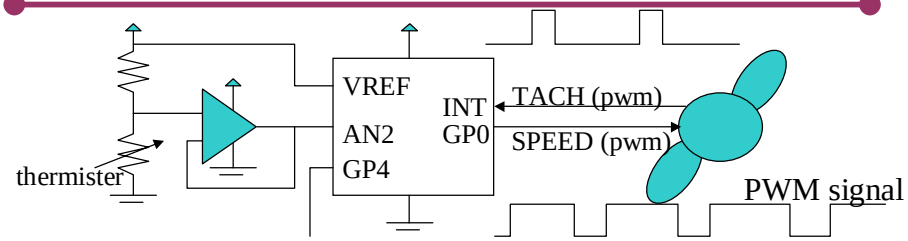
What is an Embedded System

- Its not a desktop system
  - Fixed or semi-fixed functionality (not user programmable)
  - Lacks some or all traditional human interfaces: screen, keyboard, pointing device, audio
  - May have stringent real-time requirements (Hard and Soft)
  - Usually has sensors and actuators for interface to physical world
- Figures of Merit
  - Reliability – it can never crash
  - Safety – Involves things that move and can harm/kill a person
  - Power Consumption – may run on limited power supply. Want slowest possible clock, least amount of memory. You will always be resource constrained!
  - Cost – Engineering Cost, Mfg Cost, Schedule tradeoffs
  - Product life cycle issues: maintainability, upgradeability, serviceability
  - Performance

Live without?

- Live Without
  - User Interface
  - Dynamic Linking and Loading
  - Virtual Memory, Protection Modes
  - Disk
  - Processes
- Instead we have
  - Real Time Kernel (very small OS)
  - Tasks (threads)
  - Task communication primitives
  - ADC
  - Timers
  - Event Capture
  - PWM

An Example: Temp Controller w/ 8-Pin PIC MCU



Task: Tachometer (external interrupt)
```
now = getTime();
period = then - now; //overflow?
then = now;
return;
```

Task: TempControl (periodic, soft constraint)
```
if  (Temp > setpoint) Thi++;
if  (Temp < setpoint) Thi--;
if  (period<min || period>max) GP4 = 1;
```

Task: FanPWM (periodic, hard constraint)
```
count++;
if (count == 0) GP0 = 1;
if (count > Thi) GP0 = 0;
return;
```

Task: Main
```
Thi = 0;
setup timer for 1ms interrupt;
setup timer for 100ms interrupt;
while (1) ;
```


Capacity

- Assume:
  - 4 MHz processor @ one instruction/cycle
  - Assume fan runs between 30Hz and 60Hz
  - Assume 256ms period on speed control PWM, with 1ms resolution.
- What percent of the the available cycles are used for the temperature controller?
  - [total instruction in one second] / (4m I/sec)
- How much RAM do you need?
- How much ROM?

Policies

- Design
  - Part of lecture will be devoted to design project design and lab ideas
  - Take turns writing up design meeting notes.
  - Build it in the lab
- Homeworks will be short but will **precede** lecture. Probably 2/week. Graded on an "effort" bases (1, 2, or 3 points). Must be turned in prior to start of class when due.
- Lecture
  - See Syllabus and Schedule. Generally coordinated with design problems
- Lab
  - Implementation of the design, as specified in class
  - Lab reports due prior to start of Wednesday lab section (2:30pm)
- Quizzes
  - Probably 2. Based on lecture and lab
- Final
- Reading and Source Material

Business Matters

- Sign-up for CSE466 mailing list (majordomo)
- I will generally distribute notes for lecture after using them
- Establish a rotation for design meeting write-up – alphabetical?
- Quizzes are open book
- Go to the 466/schedule link for links to lecture notes, labs, etc.
- If you have a home PC, get the tools!
- The Documents
  - Instruction set
  - Architecture overview (8051 Standards)
  - Hardware Description – Applies to all Atmel 8051 Variants (89C55 is like the 89C52 but less flash program memory)
  - 89C55 DataSheet, specifics for our part. Timer2, Electrical Specs, etc.
  - Online: Keil Getting Started Guide, C51 Manual, A51 Manual, BL51 Manual

Grading

- Lab reports: 10pts each. Demo required
- Homeworks: 3 points each (will be scaled by difficulty)
- Ratios
  - Lab: 25%
  - Homework 25%
  - Quizzes total 25%
  - Final 15%
  - Class Participation 10%

CSE466 Syllabus

- Introduction: What is an Embedded System.
  - Characteristics
  - Figures of Merit
  - An Example
  - Our Design Problem
- Embedded Hardware
  - Digital Circuits
  - Microcontrollers
  - Memory and I/O (RS-232, UART) (memory mapping)
  - Physical World Interfaces (sensors and actuators)
- The 8051 Architecture and the Keil Tools
- Development Environment
  - Assembly Language
  - Simulation
  - Debugging
  - C for Microcontrollers
  - Monitors, ICE, Single Stepping, Breakpoints, etc.
- Embedded Operating Systems (Real Time OS)
  - Scheduling
  - Synchronization and Communication
  - Response Time
  - Device Drivers
- Embedded Networking
- Safety, Ethics, and Societal Impact

Projects

- First Project – Streaming Music Synthesizer on an 8-bit micro-controller
- Second Project – Embedded Web Server on the Intrinsic Board

Lab1