



Copyright © 1999 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited



- Instructor:
 - Pascal Stang, Teaching Fellow, EE
- Guest Lecturer:
 - John Gill, Associate Professor ISL
- TA:
 - David Black-Schaffer, Graduate Student, EE



- Administrative Trivia
- What is an embedded system?
- Designing embedded systems
- Introduction to the AVR
- Lab #0: Start Thinking About Your Project
- Lab #1: Blinking Lights on the STK-500



- Maximum Class Capacity
- Course Information Sheet
 - Contact Info
 - Lab
 - Lectures
 - Grading
 - Late Policy
 - Online resources



- got a windows PC? (or linux?)
- programmed in assembly and C?
- ever used a logic analyzer?
- ever worked with microcontrollers before?
- ever soldered before?
- ever built anything for fun?



- What makes a microcontroller:
 - Self Contained
 - CPU
 - Memory
 - I/O
 - Application or Task Specific
 - Not a general-purpose computer
 - Appropriately scaled for the job



- Embedded PCs?
- “Soft” Processors on PLDs?
- Systems On A Chip?



- Microcontrollers
 - Don't have keyboard and monitor jacks
 - Must use ports to perform I/O
 - Inputs – to sense things
 - Outputs – to control things
- Related Component Topics
 - Cool Parts
 - Common Interfaces
 - Part Packages



- Labs
 - Lab 0 – Think about your project
 - Lab 1 – Blinking Lights (pushbuttons and LEDs)
 - Lab 2 – Ascii-to-Morse Converter
 - Lab 3 – LCD Clock
 - Lab 4 – “Video Paint”
 - New ideas welcome...
- Presentation
- Final Project
 - Hardware
 - Report
 - Presentation





Copyright © 2000 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited



- AVR Studio Assembler Example
- Assembler Directives
- AVR Instruction Set
- More About The AVR



- What does it generate?
 - .obj
 - .hex
- How about blink.asm?
 - Set up a project
 - Run in simulation
 - Look at generated files...



- .device
- .include
- .org
- .def
- .equ
- .db



- What were they trying to do?
- How did they implement it?
- What are the “useful” instructions?



- What are the features of RISC?
 - 1 instruction per clock cycle (pipelined)
 - Lots of registers: 32 GP registers
 - Register-to-register operation
- Variations in the parts:
 - TINY to MEGA
 - ATtiny10
 - Processor has only 8 pins – what good is it?
 - ATmega128
 - Processor has 64 pins – what do I need them all for?



- Virtually all new part datasheets are available online.
- Paper databooks are static.
- Online errata can save you from headaches.



- What do you want to make?
 - Cool Toy
 - Communication Widget
 - Specialty Control System
 - Pointless Active Desk-Art
 - A “Killer” Device
- Details about the basic project requirements will be posted on the web page



- Blinking Lights.
 - Make sure you can make AVR Studio work.
 - Figure out some variations on the demonstration program.
 - Dealing with Button Bounce
 - Get comfortable with the AVR Instruction Set.

