Embedded Systems – ECE:3360

Lecture 1: Introduction

Reinhard R. Beichel
Office: 3312 SC
Tel: 335-4597
E-mail: reinhard-beichel@uiowa.edu

Motivation

- **Key questions:**
  - **What exactly is an embedded system (ES)?**
  - **Why should I be able to design embedded systems?**

What is a Computer?

- Most of us think of "desktop" computers
  - PC's, Laptops
  - Servers, Mainframes
- But, there is another kind of computing system that is far more common: **Embedded Systems**

Embedded Systems

- Hard to define *Embedded Computing System* precisely
- Computing systems embedded within electronic devices
- Nearly any computing system other than a desktop computer, mainframe, server ("traditional computers")
- Billions of units produced yearly, versus millions of desktop units
- Perhaps 50 per household and per automobile

Computers are in here...

and here…

and here…

and many other places…

A "Short List" of Embedded Systems

Anti-lock brakes	Modems
Auto-focus cameras	MPEG decoders
Automatic teller machines	Network cards
Automatic toll systems	Network switches/routers
Automatic transmission	On-board navigation
Avionic systems	Pagers
Battery chargers	Photocopiers
Camcorders	Point-of-sale systems
Cell phones	Portable video games
Cell-phone base stations	Printers
Cordless phones	Satellite phones
Cruise control	Scanners
Curbside check-in systems	Smart ovens/dishwashers
Digital cameras	Speech recognizers
Disk drives	Stereo systems
Electronic card readers	Teleconferencing systems
Electronic instruments	Televisions
Electronic toys/games	Temperature controllers
Factory control	Theft tracking systems
Fax machines	TV set-top boxes
Fingerprint identifiers	VCR's, DVD players
Home security systems	Video game consoles
Life-support systems	Video phones
Medical testing systems	Washers and dryers

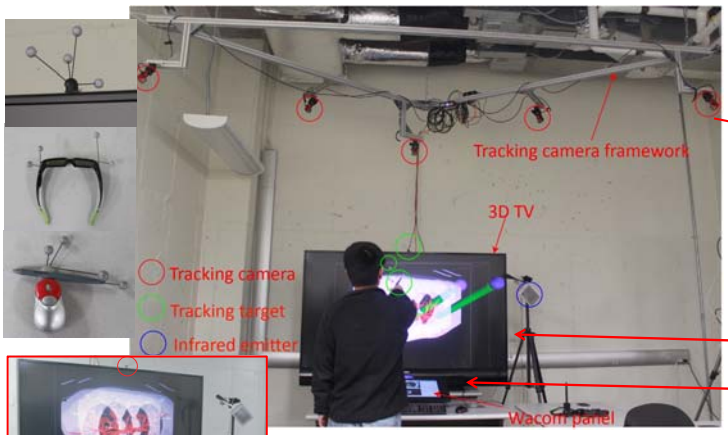
Today, almost all nontrivial electronic systems have at least one embedded processor

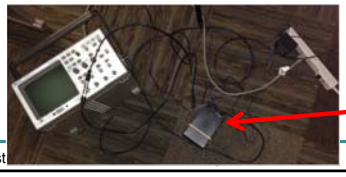
Embedded Systems - Some Examples

- **Oscilloscope watch:**
- http://www.youtube.com/watch?v=wBMdXtvPbZ4#t=53

- **HF SDR:**
- http://www.youtube.com/watch?v=xfro94lZw-U

An ES in the Virtual Reality System in my Lab





Tracking server

TCP/IP

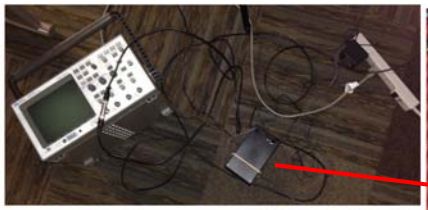
3D

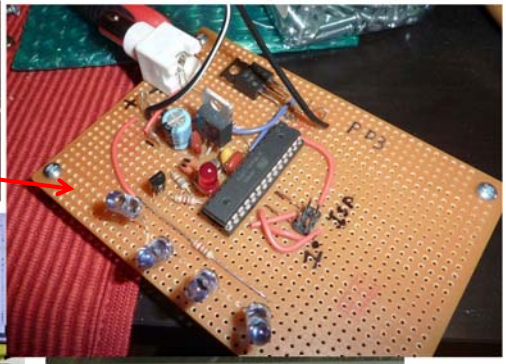
2D

Rendering server

This 'black box' makes the VR system work!

An ES in the Virtual Reality System in my Lab





IoT – Internet of Things

An "equation" for the IoT:

Physical Object

+

μController, Sensor, and Actuators

Embedded + System

Internet

=

Internet of Things (IoT)

Example

IoT – Examples

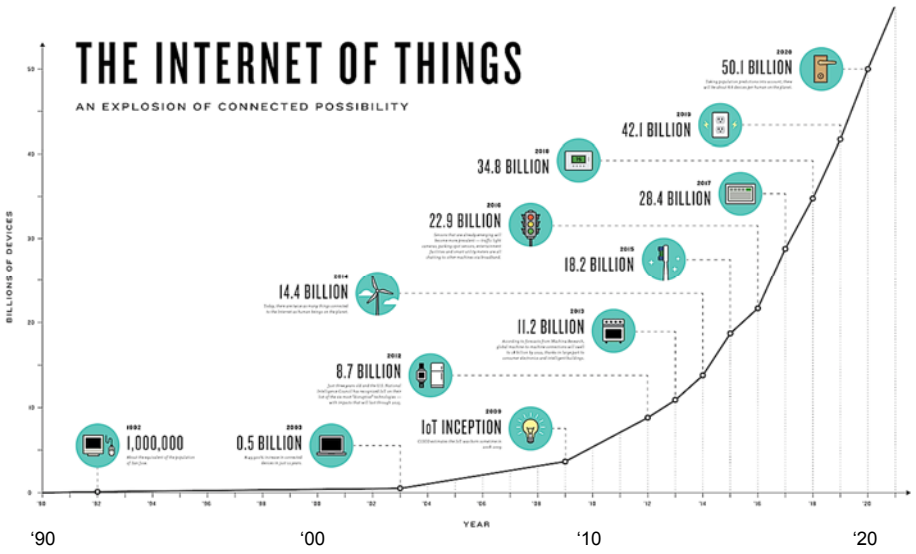
https://www.youtube.com/watch?v=7dcWd594lRw

https://www.youtube.com/watch?v=7TOsFqqJqj4

https://www.youtube.com/watch?v=ZPMf2HL938c

https://www.youtube.com/watch?v=L8TkhHgkBsg

IoT – # of "Connected" Devices



Embedded Systems vs. Desktop Computing

- **Most Embedded Systems are single-functioned**
  - Executes a single program, repeatedly
- **Generally, Embedded Systems are tightly-constrained**
  - Low cost, low power, small, fast, etc.
- **Most Embedded Systems are reactive and real-time**
  - Continually react to changes in the system's environment
  - Must compute results in "real-time" as opposed to "batch" or "off-line"

Embedded Design Challenge: Optimizing Design Metrics

- **Obvious design goal:**
  - Construct an implementation with desired functionality
- **Key design challenge:**
  - Simultaneously optimize numerous design metrics
- **Design metric**
  - A measurable feature of a system's implementation
  - Optimizing design metrics is a key challenge

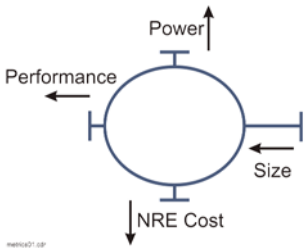
Embedded Systems Design Metrics

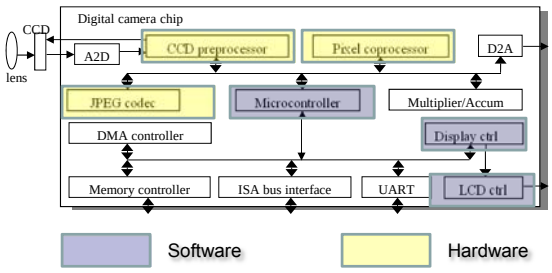
- **NRE cost (Non-Recurring Engineering cost)**
  - One-time monetary cost of designing the system
- **Unit cost**
  - Monetary cost of manufacturing each copy of the system, excluding NRE cost
- **Size**
  - Physical space required by the system
- **Performance**
  - Execution time or response time of the system
- **Memory**
  - Amount of memory required to hold the program and data
- **Power**
  - Amount of power consumed by the system

Embedded Systems Design Metrics

- **Flexibility**
  - Ability to change functionality without incurring heavy NRE cost
- **Time-to-prototype**
  - Time needed to build a working version of the system
- **Time-to-market (TTM)**
  - Time required to develop a system to the point that it can be released and sold to customers
- **Maintainability**
  - Ability to modify the system after its initial release
- **Robustness**
  - System stability and reliability
- **Safety**
  - Assurance that the system will not expose people to dangers
- **…**

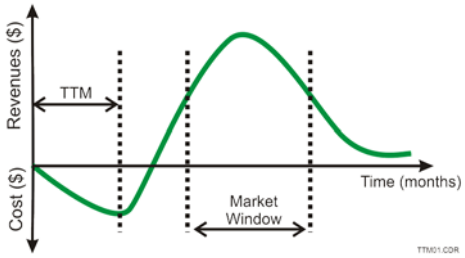
Design Metrics are Often at Odds





- **Expertise with both software and hardware is needed to optimize design metrics**
- **Not just a hardware or software expert, as is common**
- **A designer must be comfortable with various technologies in order to choose the best for a given application and constraints**

Time-to-Market (TTM): Critical Design Metric



- **Time required to develop a product to the point it can be sold to customers**
- **Market window**
  - **Period during which the product would have highest sales**
- **Average time-to-market constraint is about 8 months**
- **Delays can be costly**

NRE and Unit Cost Metrics

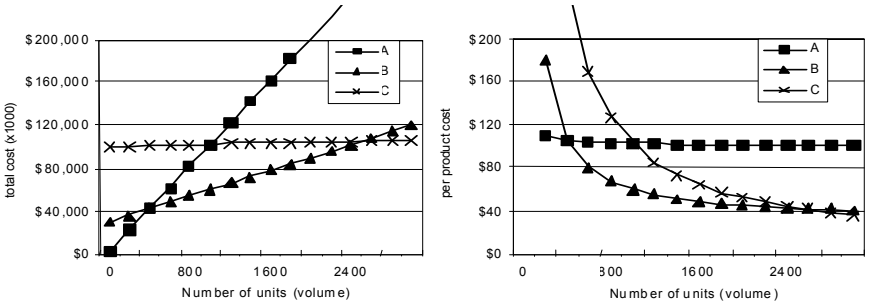
- **Costs:**
  - **Non-Recurring Engineering (NRE) cost: one-time monetary cost of designing the system**
  - **Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost**
  - ***total cost = NRE cost + Unit cost × # units***
  - ***per-product cost = total cost / # units***
  - ***= (NRE cost / # units) + unit cost***

Example

- NRE = $2000, unit = $100
- For 10 units
  - Total cost = $2,000 + 10×$100 = $3,000
  - Per-product cost = $2,000/10 + $100 = $300

Amortizing NRE cost over the units results in an additional $200 per unit

NRE and Unit Cost Metrics



- **Compare technologies by costs**
  - **Technology A: NRE=$2,000, unit=$100**
  - **Technology B: NRE=$30,000, unit=$30**
  - **Technology C: NRE=$100,000, unit=$2**
- **Must also consider Time-To-Market**

The Performance Design Metric

- **Widely-used performance metric, widely-abused**
  - **Clock frequency, instructions per second – not good measures**
  - **Digital camera example – a user cares about how fast it processes images, not clock speed or instructions per second**
- **Latency (response time)**
  - **Time between task start and end**
- **Throughput**
  - **Tasks per second, e.g. Camera A processes 4 images per second**
  - **Throughput may involve more than just latency due to concurrency, e.g., Camera B may process 8 images per second (by capturing a new image while previous image is being stored).**
- **Speedup of B over A = B's performance / A's performance**
  - **Throughput speedup = 8/4 = 2**

Embedded Processor Technologies

- General-Purpose
- Single-Purpose
- Application-Tailored

General-Purpose Processor

- Programmable device used in a variety of applications
  - Also known as "microprocessor"
- Features
  - Program memory
  - General data path with large register file and general ALU
- User benefits
  - Low time-to-market and NRE costs
  - High flexibility
- Disadvantages
  - Higher unit cost
- E.g., Intel Pentium, i5, i7, etc. … well-known, but there are hundreds of others

Single-Purpose Processor

- Digital circuit designed to execute exactly one program
  - a.k.a. coprocessor, accelerator or peripheral
- Features
  - Contains only components needed to execute a single program
  - No program memory
- Benefits
  - Fast
  - Low power
  - Small size
- Disadvantage
  - High NRE cost

Application-Tailored Processors

- Programmable processor optimized for a particular class of applications having common characteristics
  - Compromise between general-purpose and single-purpose processors
- Features
  - Program memory
  - Optimized data path
  - Special functional units
- Benefits
  - Some flexibility, good performance, size and power

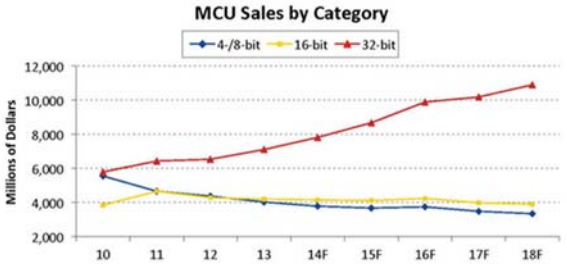
Key Player in Embedded Design: Microcontroller

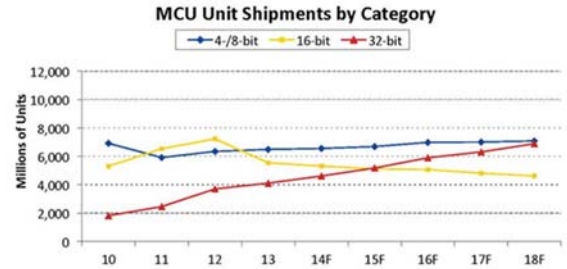
- Compromise between general-purpose and application-tailored processor
- Simple processor architecture
  - Reduced instruction set and functionality (RISC)
  - Small data path (often only 4 or 8 bits vs. 32 or 64 bits for typical general purpose processor)
- On-board memory (volatile and non-volatile)
- Multiple on-chip devices to support embedded applications
  - Timers
  - Digital and I/O serial I/O
  - Support for various interfacing protocols—e.g. I^2C
- Available in many different configurations, performance levels, etc.

The Advantages of Microcontrollers

- Low cost due to high volume production
- Low "chip count" due to integrated on-board features
- Good development tools and environments
- Extensive product families allow tailoring of processor to system design metrics
- Short product design cycles (compared to custom hardware design).
- Compatible with hardware/software co-design
  - Many microcontrollers are available as "VHDL Cores" for integration into a custom VLSI chip

Many/Most Embedded Applications Require Modest Computational Power



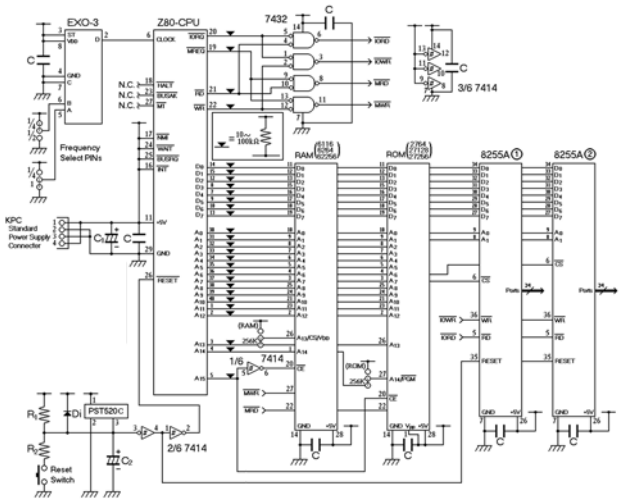


Source: IC Insights' *2014 McClean Report*

… Some History

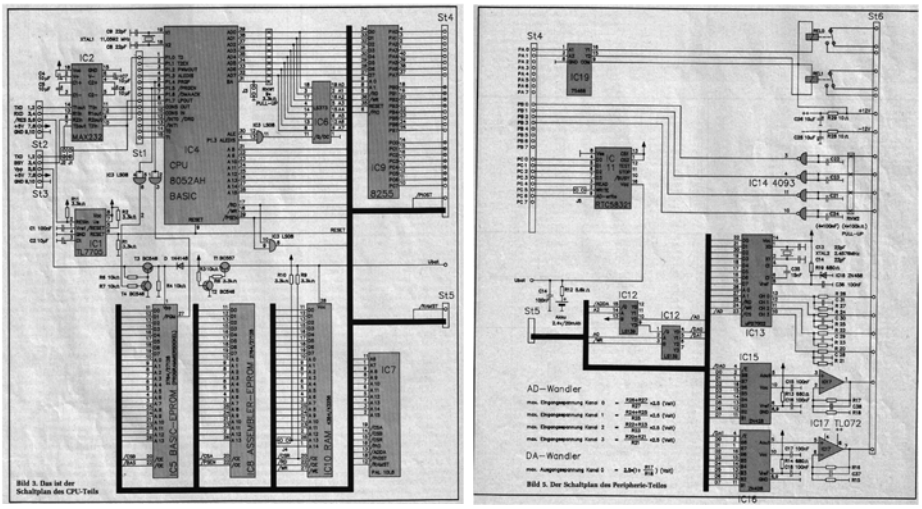
PDF

Schematic of a Z80-based Board for Building an ES



68000-based Board for Building an ES

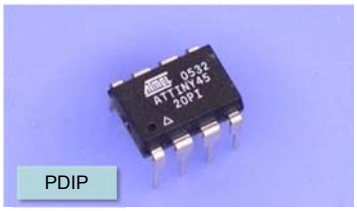
8052-based ES Board (1986)

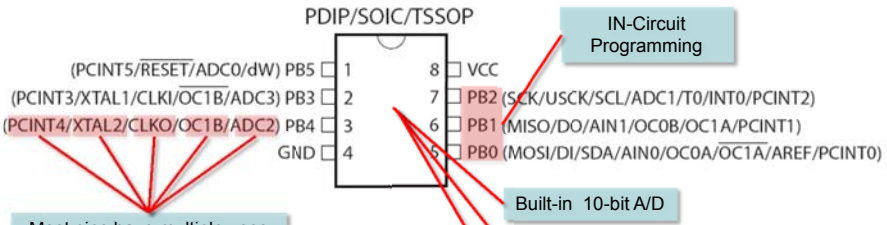


What we will be using in the lab …

A Low End Microcontroller: ATtiny25/45/85

Pinout for ATtiny25/45/85





IN-Circuit Programming

Most pins have multiple uses

Built-in 10-bit A/D

Internal Oscillator

Up to 20 MHz clock

Built-in Analog Comparator

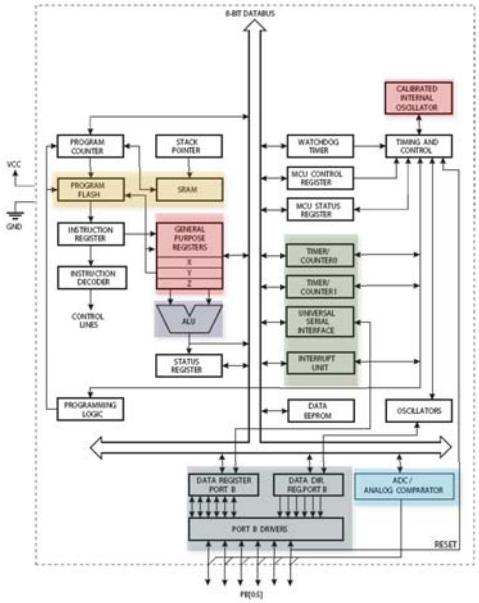
~$1 in Large quantities

I_{supply} at 1 MHz, 1.8V: 300 µA

V_{cc} for some variants as low as 1.8 V

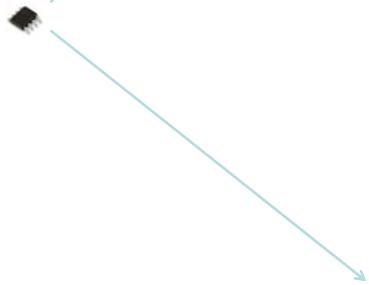
A Low End Microcontroller: ATtiny25/45/85

Block Diagram of ATtiny25/45/85



A Low End Microcontroller: ATtiny25/45/85

ATtiny25/45/85 Features



- Advanced RISC Architecture
  - 120 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
- Non-volatile Program and Data Memories
  - 2/4/8K Bytes of In-System Programmable Program Memory Flash
    - Endurance: 10,000 Write/Erase Cycles
  - 128/256/512 Bytes In-System Programmable EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 128/256/512 Bytes Internal SRAM
  - Programming Lock for Self-Programming Flash Program and EEPROM Data Security
- Peripheral Features
  - 8-bit Timer/Counter with Prescaler and Two PWM Channels
  - 8-bit High Speed Timer/Counter with Separate Prescaler
    - 2 High Frequency PWM Outputs with Separate Output Compare Registers
    - Programmable Dead Time Generator
  - USI – Universal Serial Interface with Start Condition Detector
  - 10-bit ADC
    - 4 Single Ended Channels
    - 2 Differential ADC Channel Pairs with Programmable Gain (1x, 20x)
    - Temperature Measurement
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - debugWIRE On-chip Debug System
  - In-System Programmable via SPI Port
  - External and Internal Interrupt Sources
  - Low Power Idle, ADC Noise Reduction, and Power-down Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal Calibrated Oscillator
- I/O and Packages
  - Six Programmable I/O Lines
  - 8-pin PDIP, 8-pin SOIC, 20-pad QFN/MLF, and 8-pin TSSOP (only ATtiny45/V)
- Operating Voltage
  - 1.8 - 5.5V for ATtiny25V/45V/85V
  - 2.7 - 5.5V for ATtiny25/45/85
- Speed Grade
  - ATtiny25V/45V/85V: 0 – 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
  - ATtiny25/45/85: 0 – 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Industrial Temperature Range
- Low Power Consumption
  - Active Mode:
    - 1 MHz, 1.8V: 300 µA
  - Power-down Mode:
    - 0.1 µA at 1.8V

A Low End Microcontroller: ATtiny25/45/85

ATtiny25/45/85 Features

- Advanced RISC Architecture
  - 120 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
- Non-volatile Program and Data Memories
  - 2/4/8K Bytes of In-System Programmable Program Memory Flash
    - Endurance: 10,000 Write/Erase Cycles
  - 128/256/512 Bytes In-System Programmable EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 128/256/512 Bytes Internal SRAM
  - Programming Lock for Self-Programming Flash Program and EEPROM Data Security

A Low End Microcontroller: ATtiny25/45/85

ATtiny25/45/85 Features

- Peripheral Features
  - 8-bit Timer/Counter with Prescaler and Two PWM Channels
  - 8-bit High Speed Timer/Counter with Separate Prescaler
    - 2 High Frequency PWM Outputs with Separate Output Compare Registers
    - Programmable Dead Time Generator
  - USI – Universal Serial Interface with Start Condition Detector
  - 10-bit ADC
    - 4 Single Ended Channels
    - 2 Differential ADC Channel Pairs with Programmable Gain (1x, 20x)
    - Temperature Measurement
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator

A Low End Microcontroller: ATtiny25/45/85

ATtiny25/45/85 Features

- Special Microcontroller Features
  - debugWIRE On-chip Debug System
  - In-System Programmable via SPI Port
  - External and Internal Interrupt Sources
  - Low Power Idle, ADC Noise Reduction, and Power-down Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal Calibrated Oscillator
- I/O and Packages
  - Six Programmable I/O Lines
  - 8-pin PDIP, 8-pin SOIC, 20-pad QFN/MLF, and 8-pin TSSOP (only ATtiny45/V)

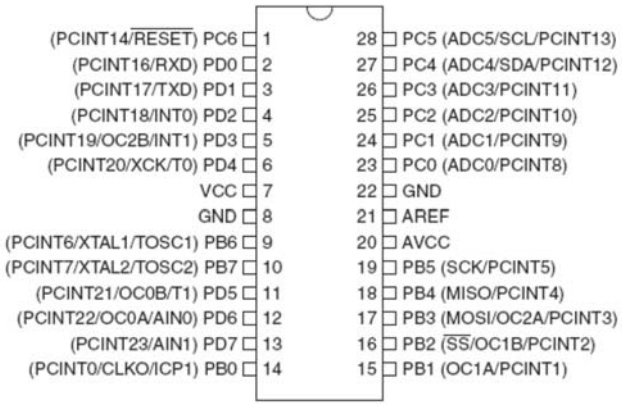
A Low End Microcontroller: ATtiny25/45/85

ATtiny25/45/85 Features

- Operating Voltage
  - 1.8 - 5.5V for ATtiny25V/45V/85V
  - 2.7 - 5.5V for ATtiny25/45/85
- Speed Grade
  - ATtiny25V/45V/85V: 0 – 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
  - ATtiny25/45/85: 0 – 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Industrial Temperature Range
- Low Power Consumption
  - Active Mode:
    - 1 MHz, 1.8V: 300 µA
  - Power-down Mode:
    - 0.1 µA at 1.8V

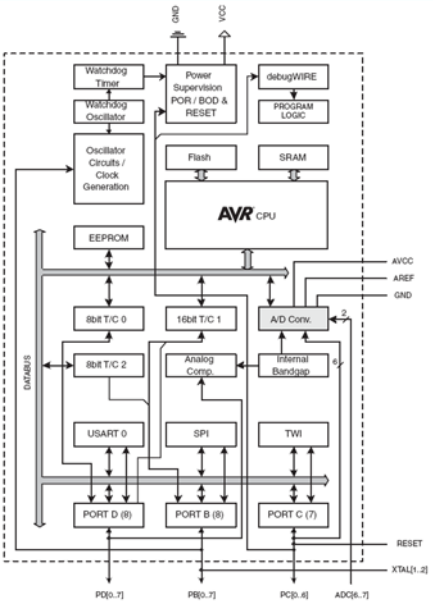
"Smaller" Microntroller: ATmega48P/88P/168P/328P

Pinout for ATmega48P/88P/168P/328P



"Smaller" Microntroller: ATmega48P/88P/168P/328P

Block Diagram for ATmega48P/88P/168P/328P



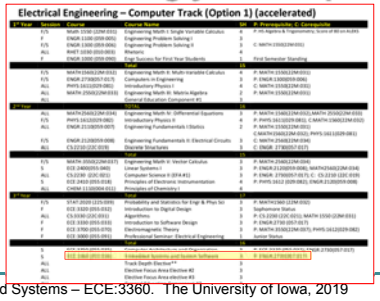
Some ATmega48P/88P/168P/328P Features

- **Clock rate up to 20 MHz**
- **Up to 20 MIPS throughput at 20 MHz**
- **4/8/16/32K Bytes of In-System Self-Programmable Flash Program Memory**
- **512/1K/1K/2K Bytes Internal SRAM**
- **256/512/512/1K Bytes EEPROM**
- **35 digital I/O**
- **4 timers**
- **2 PWM/Capture/Compare modules**
- **UART for serial I/O**
- **6-channel 10-bit ADC in PDIP Package**
- **Interrupt sources**
- **Cost: Approx $2.30 in quantity**

Organizational

Hard and "Soft" Pre/Co-requisites

- **Prerequisites:**
  - **Computers in Engineering (ENGR:2730)**
  - **Introduction to Digital Design (ECE:3320)**
  - **Engineering Fundamentals II: Electrical Circuits (ENGR:2120) (*)**
  - **Introduction to Software Design (ECE:3330)**
- **Corequisites:**
  - **Principles of Electronic Instrumentation (ECE:2410)**
- **→consider when selecting your lab partner!**



Organizational

- **Class Website (be sure to check it often):**
- http://www.engineering.uiowa.edu/~rbeichel/lectures/es_s19/index.html
- **Syllabus: see class website**
- **Primary course materials:**
  - Lecture notes (on ICON), microcontroller data sheets, and handouts on the resources page
  - Lab instructions
  - The following books are optional:
    - Programming and Customizing the AVR Microcontroller, Dhananjay Gadre, McGraw Hill, 2000.
    - Embedded System Design– A Unified Hardware/Software Introducton, by Frank Vahid and Tony Givargis, John Wiley & Sons Inc., 2002
- **Goal:**
  - Ability to design embedded microcontroller systems

Organizational

- **There are 5 TAs and they will hold office hours in the lab.**
- **The labs are located in 2244 SC**
- **There are five to six labs with 2 – 3 weeks in duration**
  - Labs can have a pre-lab and/or post-lab component
  - Labs overlap
  - TAs must sign off on labs
- **There is also a larger project with in-class demonstration at the end of the semester**
- **Labs and project will count 55% towards the course grade**
- **Students work in teams of two (selection!)**
  - Let us know ASAP who your lab partner is → sign up sheet in ECE office; **deadline: Thursday (01/17)**
  - We can also assign you randomly a partner
- **Students commonly get into "trouble" because they underestimate the time it takes to complete lab**
  → murphy's law

Organizational

- **No scheduled lab times (except introduction)**
  - You may use the lab any time there is an open slot (24 x 7)
  - Access via UI ID card
  - TAs will hold office hours in lab
- **Pre-and post lab check off**
  - Students work in teams, and both must submit and be present for check off
  - Cannot change teams during labs
  - Bonus points for consistency (→ same partner for all labs and project) !
- **Late check off are penalized by 1% per hour**

Office Hours & Duties

- **TAs will hold office hours in lab. Their responsibilities are**
  - Help students with equipment.
  - Answer general, conceptual questions
  - Check off labs and assign lab grades
- **TAs will not/cannot**
  - Will not help debug code beyond simple troubleshooting
  - Cannot reset time stamps for penalties for late lab reports
- **My office hours and contact information are on the class website**

Exams & Grades

- No homework
- One midterm exam (before Spring Break)
- One final exam
- Grade Calculation
  - Midterm exam: 22.5%
  - Final exam: 22.5%
  - Labs: 35% Project: 20%
- Exam details
  - No redo exam
  - Open book, open notes

The First Lab

- Activities
  - Learn how to solder
  - Assemble development board
  - Load a supplied program on the µC
  - Make small changes to the program
  - Write and submit the first lab report
- Each team: purchase one Kit A from CoE Electronics Shop
- The Electronics Shop technicians and TAs will conduct introductory sessions on Thursday and Friday afternoons → sign up sheet
  - Soldering skills
  - Tips for board assembly and testing
- See class website/Labs/Lab1 for session times
- You should sign up for one of these sessions
- Each group will need to purchase an In-System Programmer (ISP) for the AVR microcontroller (+USB cable)
  → Electronics Shop

Lab Check Off

- **A) Check off before the day your work is due:**
  - **See a TA during regular office hours. You will receive 5% bonus points for your work.**
- **B) Check off at the day your work is due:**
  - **To check off, you must sign up for a time slot in advance. Time slots are limited to ~10 min. Consequently, you should come early and prepare for check off (e.g., print the check off sheet, setup of microcontroller board and/or oscilloscope, etc.) so that you are ready at the selected check off time. If you are not prepared and ready to demonstrate your work at the selected check off time, you will lose your time slot and the TA will check off another group, which will likely result in a late check off.**
- **C) Check off after due date:**
  - **See a TA during regular office hours. Late reports are penalized as outlined on the course website.**
- **Again, please note that both lab partners must be present at check off.**
- **TA office hours will likely depend on check off deadlines!**

http://www.engineering.uiowa.edu/~rbeichel/lectures/es_s19/index.html

… EOL