Intro to Microcontrollers

Class 0: Overview and Setup

September 9, 2008

Outline

What is a Microcontroller?

How do I do That?

Outline

What is a Microcontroller?

How do I do That?

What is a micro?

It's a whole computer on a chip:

- Write programs in various languages (C, assembly, BASIC)
- CPU (1-20MHz)
- Dynamic memory (SRAM)
- Non-volatile memory (EEPROM)

But it's a *very* little computer:

- 8-bit words
- Not much memory (4kb program space, 512 bytes SRAM)
- No operating system
- Low-level input/output
- = halfway between a "component" and a "computer"

What can it do?

Damn-near anything!

- Super-fancy Blinkers
- Robots
- ROM readers
- Phone dialers
- Noisemakers
- GPS dataloggers
- What do you need to do?

Outline

What is a Microcontroller?

How do I do That?

Basic Functionality

What do they Actually Do?

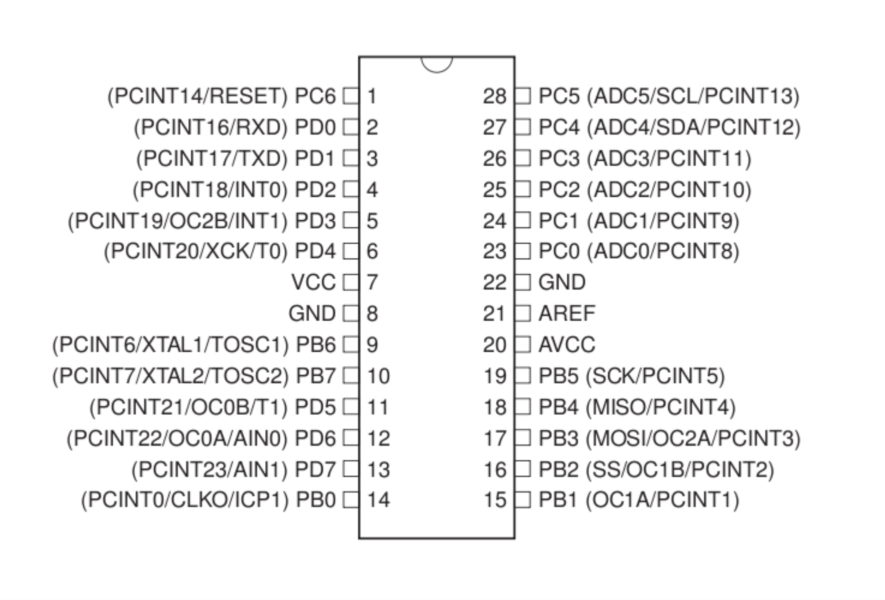
- Output: 5v or 0v for each pin.
  (Light up LEDs, flip switches, spin motors)
- Input: Digital (pushbuttons, threshold sensors) or Analog-to-Digital conversion (light levels, audio waveforms)
- Neither: ("Hi-Z") plays like it's disconnected from the circuit
- Pulse-width Modulation (PWM): Flip the digital output on and off quickly. Simple way of making an analog signal with a digital output pin

Other Stuff

Useful features

- Timers: They have (3) internal clocks,
  useful for both timing and scheduling events
- (Timers also make doing PWM and audio stuff easy)
- Serial I/O: built-in hardware-level routines for
  USART, SPI, I2C serial protocols
- Interrupts: Allow you to call a subroutine whenever a button is pushed or a certain timer event occurs (and more)

Pinouts



The Basic Workflow

What will we actually be doing?

- Write code in C (using whatever you want)
- Cross-compile for the chip → the machine-code version of your code
- Transfer the code to the chip:
  – Programmer to talk to the chip
  – Software to run the programmer
- Get feedback/debug until it works

The "Toolchain"

- Cross-compiler: GNU GCC and a bunch of help from avr-libc
- AVRdude: knows how to run a bunch of programmers
- Programmer (USBtiny)

So What are We Doing Today?

To Work, You!

- Install software (easy)
- Make programmers (harder and slightly involved)
- Test it out (flash in a program and blink lights!)
- ??
- Profit!

The End

◂ Outline