Introduction to Arduino

Prof. Dhaval Shah

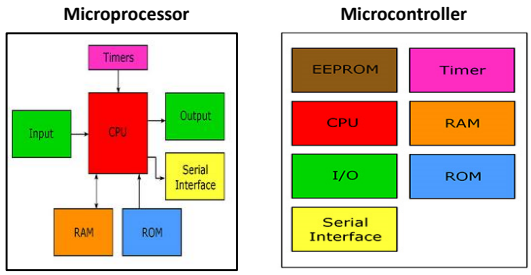
Outline

- What is Microcontroller
- Microprocessor vs Microcontroller
- Classification
- Arduino Boards
- Features of Arduino UNO
- ATmega328
- Arduino Programming

What is Microcontroller?

- Microcontrollers are small computers integrated into a single chip
- It contains :
  - Processing core
  - Flash Memory for program
  - I/O peripherals
  - RAM
  - Peripherals such as clocks, timers, PWM etc…
- Microprocessors are used for general purpose applications, while microcontrollers are self sufficient and are used for specific tasks.
- Microcontrollers are an example of embedded systems.

Microprocessor Vs Microcontroller



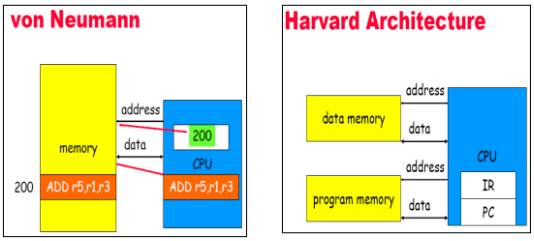
CPU and several supporting chips

CPU on single chip

Classification

- Based on Hardware
  - Von Neuman
  - Hardvard
- Based on Instruction Set Architecture
  - RISC (Reduced Instruction Set Computing)
  - CISC (Complex Instruction Set Computing)

Classification: Based on Hardware



Classification Based on ISA

Complex Instruction Set Computer (CISC)

- Requires multiple cycles for a execution.
- Different instructions of different length and format
- Limited general purpose register
- A large Number of instructions

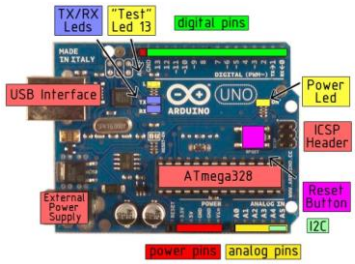
Reduced Instruction Set Computer (RISC)

- Instruction can be executed in a single cycle.
- Each instruction of fixed length and format
- Large general purpose register set
- Compaq instruction set

What is Arduino?

- The Arduino is a microcontroller development platform(not a microcontroller….) board with a USB plug.
- It is an open-source physical computing platform.
- It can be used to develop stand-alone interactive objects or can be connected to software on your computer.
- Easy-to-use hardware and software.
- It's intended for students, artists, designers, hobbyists and anyone who tinker with technology.
- It is programmed in Arduino Programming language(APL) similar to C/C++.

Arduino UNO



- 16 MHz with auto-reset,
- 6 Analog In,
- 14 Digital I/O
- 6 PWM

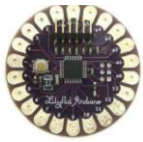
Courtesy: A000066-Arduino-datasheet-38879526.pdf (octopart.com)

Features of Arduino UNO

- An easy USB interface
- convenient power management and built-in voltage regulation
- An easy-to-find, and dirt cheap, microcontroller "brain"
  - timers, PWM pins, external and internal interrupts, and multiple sleep modes
- 16 MHz clock
- 32 KB of flash memory
- 13 digital pins and 6 analog pins
- An ICSP connector
- An on-board LED

Other Flavors

MEGA

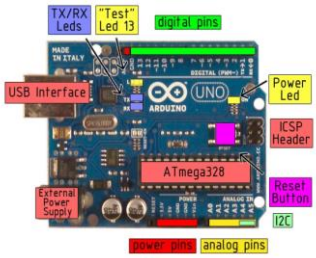


LILYPAD

MINI

NANO 43mm x 18mm

Arduino MEGA



- 16 MHz with auto-reset,
- 16 Analog In,
- 54 Digital I/O
- 6 PWM

Courtesy: A000066-Arduino-datasheet-38879526.pdf (octopart.com)

Arduino MEGA – Technical Specs

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Courtesy: A000066-Arduino-datasheet-38879526.pdf (octopart.com)

ATmega328

- High Performance, Low Power Atmel®AVR® 8-Bit Microcontroller
- **Advanced RISC Architecture**
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program

  Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- In-System Programming by On-chip Boot Program

Arduino Programming

- Arduino programming language is based on C/C++
- Simpler and easier to learn (Arduino programming is like building with LEGO blocks)
- Certain rules must be followed and different building blocks can be used to build bigger parts.
  - Every line must either end with a semicolon ';' unless it's a conditional, loop, or function
  - Comments start with a //
  - Comments are text that the program ignores

Arduino Programming

- Arduino programming language can be divided in three main parts:
  - functions
    - Digital I/O, Analog I/O, Interrupts, Communication, Time, etc..
  - Values
    - Constants, Variables, datatypes, conversions
  - Structures
    - Sktech (loop, setup), operators (arithmetic, comparison, boolean, bitwise, etc…), control structure (break, if, else, for, while, etc…..)

Constants and Variables

- Constants and variables hold data according to their datatype.
- Constants hold data that **will NOT** change while a program is running.
- Constants usually contain pin numbers or sensor threshold values.
- Variables contain data that **WILL change** while a program is running.
- Variables usually contain sensor values and other values that need to have mathematical operations done on them

Constants

- HIGH
  - a voltage greater than 3.0V is present at the pin (5V boards)
  - a voltage greater than 2.0V is present at the pin (3.3V boards)
- LOW
  - a voltage greater than 1.5 V is present at the pin (5V boards)
  - a voltage greater than 1.0V (Approx) is present at the pin (3.3V boards)
- true : It is often said to be defined as 1
- false: It is often said to be defined as 0
- Integer Constants:
- LED_BUILTIN : It is the number if pin to which the on board LED is connected. Most of the boards have this LED connected to digital pin 13

Constants

- Floating Point Constants

FLOATING-POINT CONSTANT	EVALUATES TO:	ALSO EVALUATES TO:
10.0	10	10
2.34E5	2.34 * 10^5	234000
67e-12	67.0 * 10^-12	0.000000000067

- Integer Constants

BASE	EXAMPLE	FORMATTER	COMMENT
10 (decimal)	123	none	
2 (binary)	0b1111011	leading "0b"	characters 0&1 valid
8 (octal)	0173	leading "0"	characters 0-7 valid
16 (hexadecimal)	0x7B	leading "0x"	characters 0-9, A-F, a-f valid

Data Types

Datatypes are the different kinds of data values that can be used, manipulated and stored using C++.

Datatype	What it stores (examples)	Default value	Notes
Boolean	A true value (1, TRUE, HIGH) or a false value (0, FALSE, LOW)	0, FALSE, LOW	-
Int	An integer number (-5, 15, 1047, etc.)	0	Can be positive or negative
double	A decimal number (-0.5, 123.77, etc.)	0	Can be positive or negative
Char	A single character ('c', 'A', '5', '?', etc.)	Indeterminate	Must be enclosed in single quotes
String	A sequence of characters ("Hello World!", "10", "157+5", etc.)	Empty ("")	Must be enclosed in double quotes

Operators

- The results of these operations are usually stored in a variable.

Operator	What it does
=	Assigns a value to a variable
+	Adds two or more values
-	Subtracts two or more values
*	Multiplies two or more values
/	Divides two or more values
++	Increment by 1
--	Decrement by 1
==	Checks if two value are equal
!=	Checks if two value are not equal
> or <	Greater than/ Less than comparison
<= or >=	Less than/greater than or equal to comparison
&& or ||	Boolean AND or Boolean OR Used to cascade multiple Boolean operations

Arduino IDE

- Get the latest version of Arduino IDE from https://www.arduino.cc/en/Main/Software

How to write a code in Arduino?


```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
```