Hardware-Software Codesign

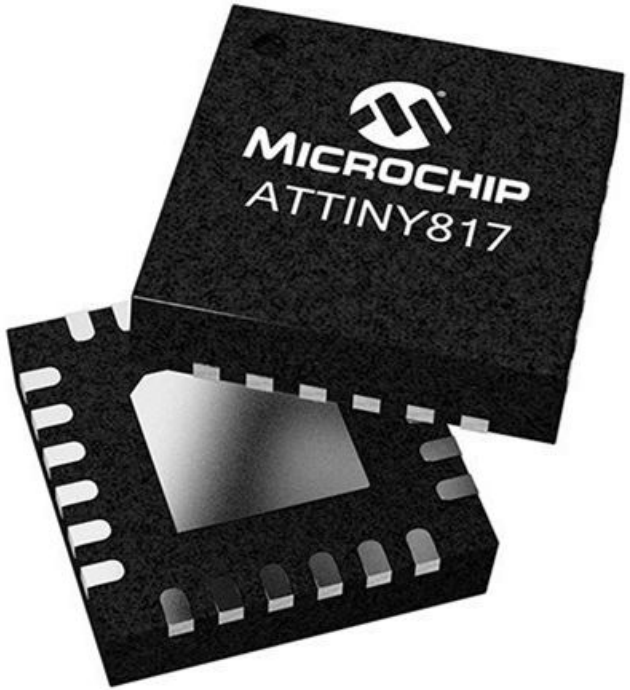
Nimal Skandhakumar

Faculty of Technology
University of Sri Jayewardenepura

2019

Microcontrollers

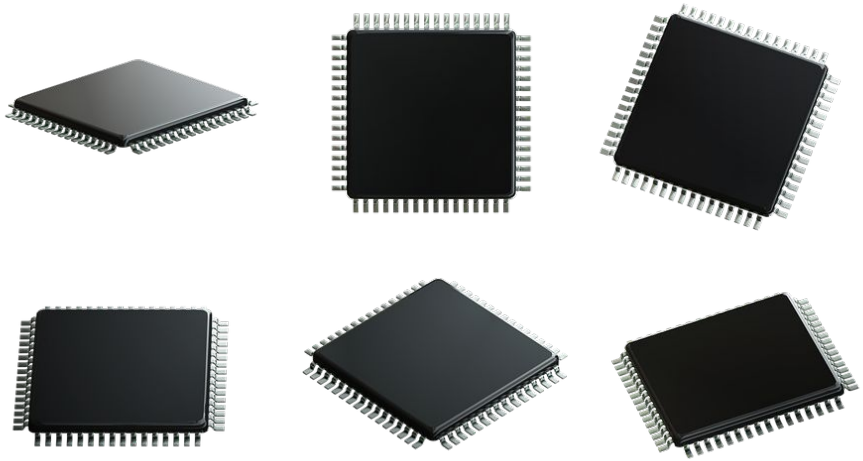
- A microcontroller is a compact integrated circuit
  - Designed to govern a specific operation in an embedded system
- A typical microcontroller includes:
  - Processor (8-bit, 16-bit, 32-bit)
  - Memory (limited)
  - Input/output (I/O) peripherals
- Generally no operating system
  - Programs written in high-level language and converted to machine code



https://www.youtube.com/watch?v=jKT4H0bstH8

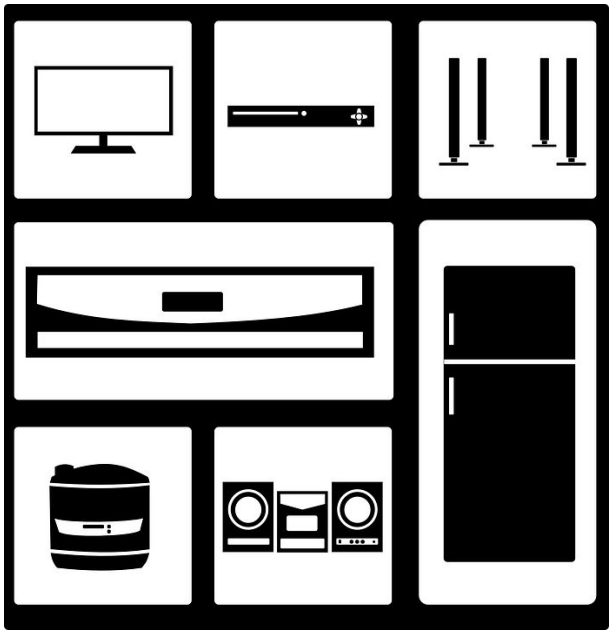
Microcontrollers

- A microcontroller is essentially a microprocessor with on-chip memories and I/O devices
- Designed for specific functions
- All in one solution
- Reduction in chip count
- Reduced cost, power, physical size, etc.



Applications of Microcontrollers

- Day to Day Life Devices:
  - Household electronics like mobile phones, camera and washing machine
  - Light sensing & control
  - Temperature sensing & control
  - Fire detection & safety devices
  - Automobiles

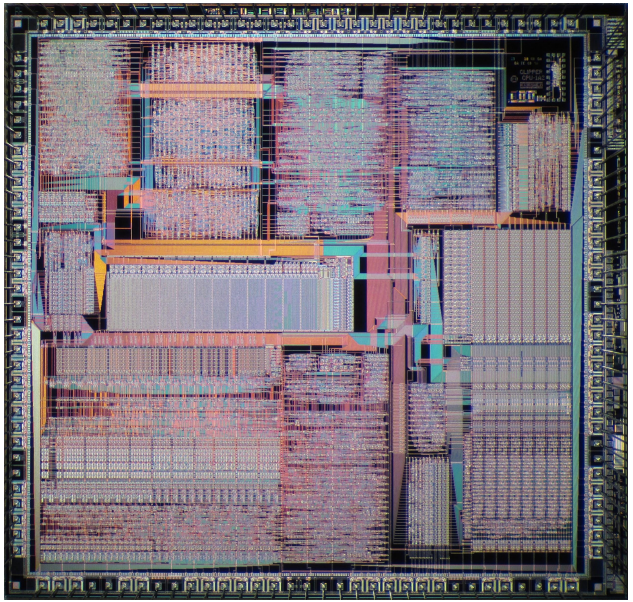


Applications of Microcontrollers

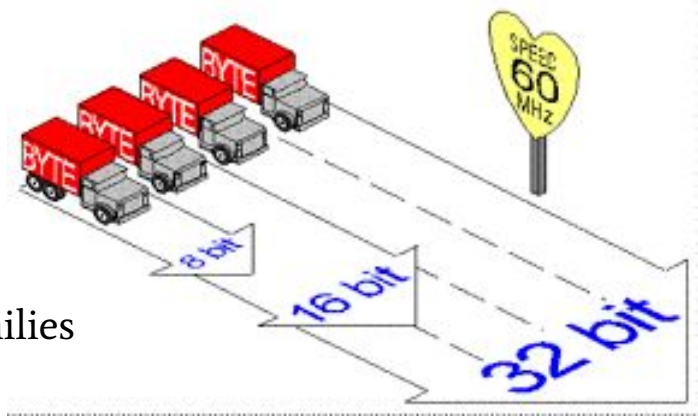
- Industrial Control Devices:
  - Industrial instrumentation devices
  - Process control devices
- Metering & Measurement Devices:
  - Volt Meter
  - Measuring revolving objects
  - Current meter
  - Hand-held metering systems

Why Study Microcontrollers

- Build useful applications
- Practice programming and debugging skills
- Understand the inside of computer
- It paves the way to learning embedded systems, computer design, operating systems, compilers, security and other topics.



Microcontroller Classifications



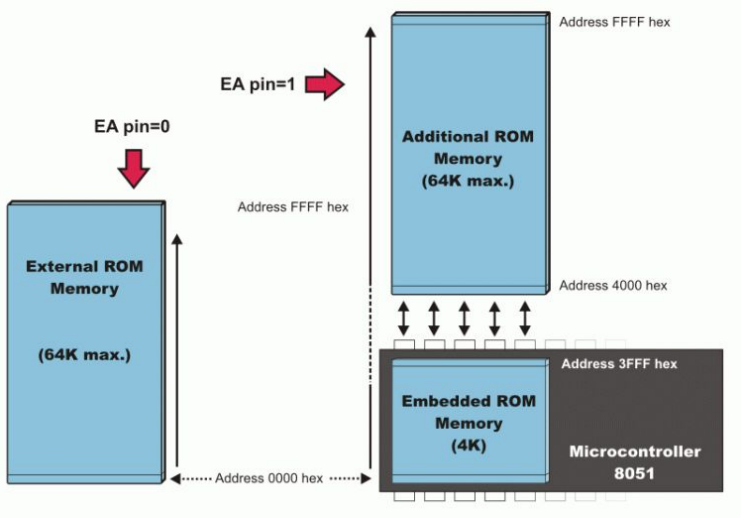
Classification According to Number of Bits:

- 8-bit microcontrollers
  - Intel 8031/8051, PIC1x and Motorola MC68HC11 families
- 16-bit microcontrollers
  - Performs greater precision and performance as compared to 8-bit
- 32-bit microcontrollers
  - Used in automatically controlled devices including implantable medical devices, engine control systems, office machines, appliances and other types of embedded systems

Microcontroller Classifications

Classification According to Memory Devices:

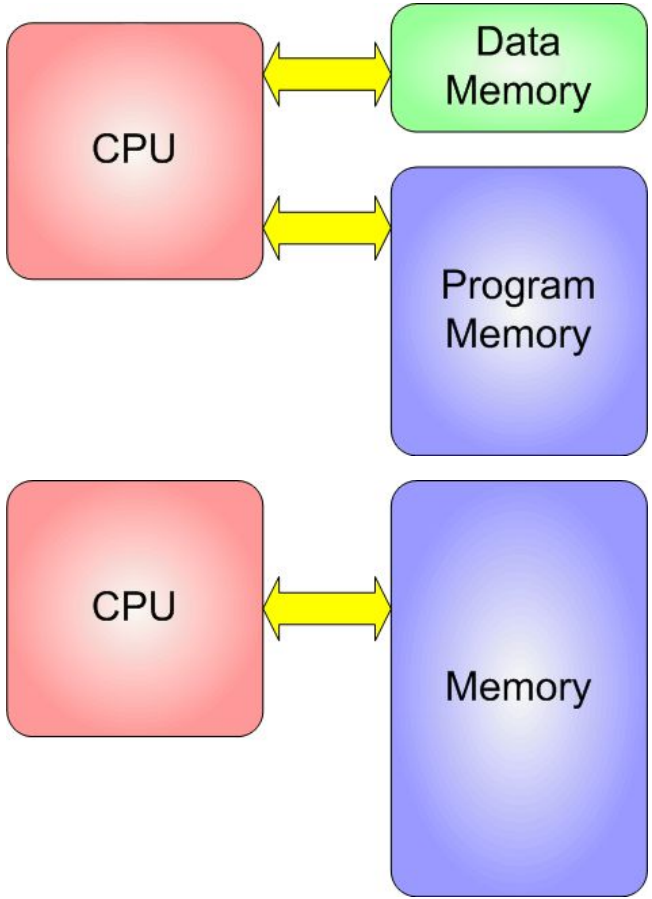
- Embedded memory microcontroller
  - Microcontroller unit has all the functional blocks such as program & data memory, I/O ports, serial communication, counters and timers and interrupts on the chip is an embedded microcontroller.
- External Memory Microcontroller
  - Microcontroller unit does not have all the functional blocks available on a chip.



Microcontroller Classifications

Classification According to Memory Architecture:

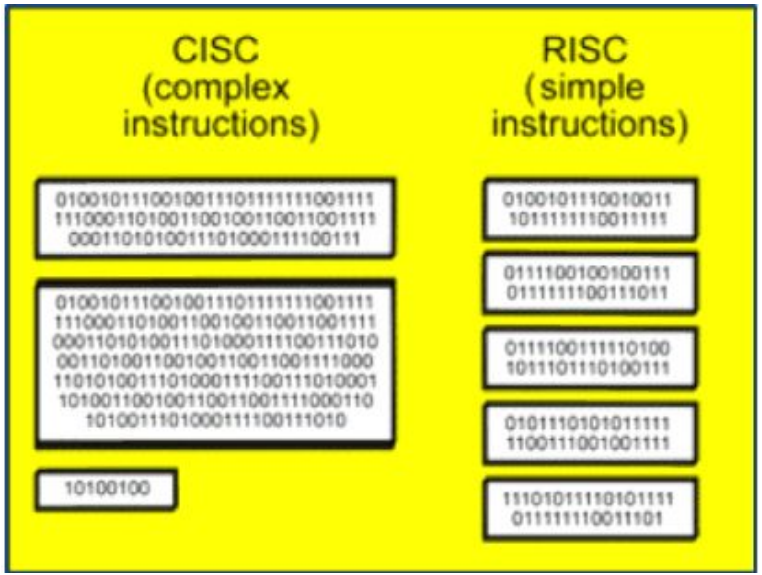
- Harvard Memory Architecture Microcontroller
  - Microcontroller unit has a dissimilar memory address space for the program and data memory.
- Princeton Memory Architecture Microcontroller
  - Microcontroller unit has a common memory address for the program memory and data memory.



Microcontroller Classifications

Classification According to Instruction Set:

- Complex Instruction Set Computer (CISC)
  - It allows the programmer to use one instruction in place of many simpler instructions.
- Reduced Instruction set Computer (RISC)
  - It allows each instruction to operate on any register or use any addressing mode and simultaneous access of program and data.



CISC (complex instruction set computer)

- Computers designed with a full set of computer instructions
  - intended to provide needed capabilities in the most efficient way
- Designed to make programming easier
  - either for assembly programmer or compiler programmer
- One instruction = multiple operations = multiple cycles
- Issues:
  - most programs use only small % of available instructions
  - instruction set & chip hardware become more complex with each generation

https://www.youtube.com/watch?v=BJpMmq9gQE8

RISC (reduced instruction set computer)

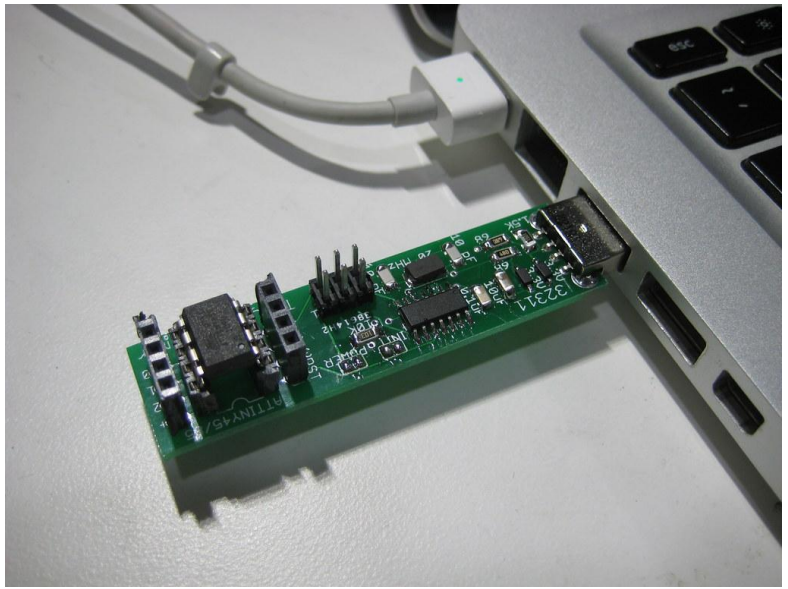
- Designed to perform a smaller number of types of computer instructions so that it can operate at a higher speed
- Majority of low end and mobile systems use RISC architecture
- ARM architecture dominates mobile/embedded systems markets

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

Programming microcontrollers

Typical steps to programming microcontrollers:

1. write program code on computer
2. compile the code with a compiler for the microcontroller
3. upload the compiled version of the program to the microcontroller



Assembly language

- Machine code is the lowest level of programming
  - too obscure and complex for software development
- Assembly language is low-level programming
  - designed for a specific family of processors
  - represents various instructions in symbolic code
  - more understandable form
  - converted by assembler to machine code

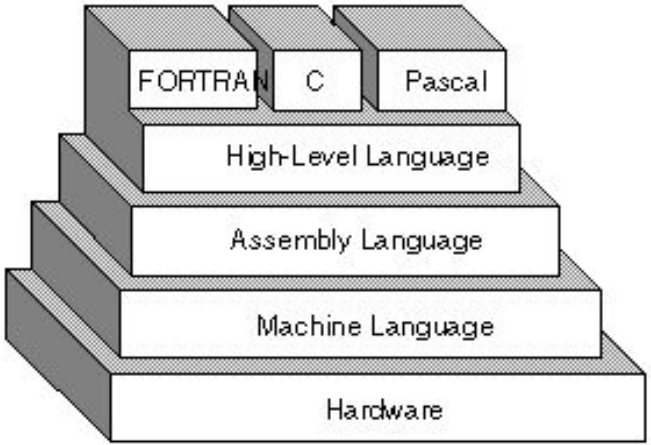
https://www.youtube.com/watch?v=wA2oMRmbrfo

```
MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE    2


C000                    ORG    ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START    LDS    #STACK

                        ************************************
                        * FUNCTION: INITA - Initialize ACIA
                        * INPUT: none
                        * OUTPUT: none
                        * CALLS: none
                        * DESTROYS: acc A

0013                    RESETA  EQU    %00010011
0011                    CTLREG  EQU    %00010001

C003 86 13     INITA    LDA A  #RESETA   RESET ACIA
C005 B7 80 04           STA A  ACIA
C008 86 11              LDA A  #CTLREG   SET 8 BITS AND 2 STOP
C00A B7 80 04           STA A  ACIA

C00D 7E C0 F1           JMP    SIGNON    GO TO START OF MONITOR

                        ************************************
                        * FUNCTION: INCH - Input character
                        * INPUT: none
                        * OUTPUT: char in acc A
                        * DESTROYS: acc A
                        * CALLS: none
                        * DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04  INCH     LDA A  ACIA      GET STATUS
C013 47                 ASR A            SHIFT RDRF FLAG INTO CARRY
C014 24 FA              BCC    INCH      RECIEVE NOT READY
C016 B6 80 05           LDA A  ACIA+1    GET CHAR
C019 84 7F              AND A  #$7F      MASK PARITY
C01B 7E C0 79           JMP    OUTCH     ECHO & RTS

                        ************************************
                        * FUNCTION: INHEX - INPUT HEX DIGIT
                        * INPUT: none
                        * OUTPUT: Digit in acc A
                        * CALLS: INCH
                        * DESTROYS: acc A
                        * Returns to monitor if not HEX input

C01E 8D F0     INHEX    BSR    INCH      GET A CHAR
C020 81 30              CMP A  #'0       ZERO
C022 2B 11              BMI    HEXERR    NOT HEX
C024 81 39              CMP A  #'9       NINE
C026 2F 0A              BLE    HEXRTS    GOOD HEX
C028 81 41              CMP A  #'A
C02A 2B 09              BMI    HEXERR    NOT HEX
C02C 81 46              CMP A  #'F
C02E 2E 05              BGT    HEXERR
C030 80 07              SUB A  #7        FIX A-F
C032 84 0F     HEXRTS   AND A  #$0F      CONVERT ASCII TO DIGIT
C034 39                 RTS

C035 7E C0 AF  HEXERR   JMP    CTRL      RETURN TO CONTROL LOOP
```

High-level language (HLL)

- High-level language is closer to “human readable”
  - independent of a particular type of computer
  - easier to program for different target systems
  - easier to read, write, and maintain
  - compiler or interpreter converts it for use on a specific device

https://www.youtube.com/watch?v=HtUQzhTt3gE



Types of Microcontrollers

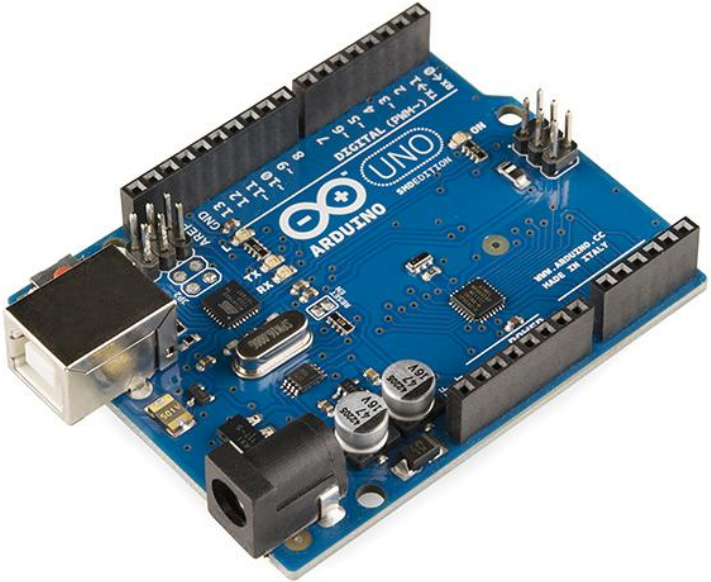
- 8051 Microcontroller
  - Developed by Intel in 1980 for use in embedded systems. It has CISC instruction architecture and Harvard memory architecture.
- PIC Microcontroller
  - PIC is a microcontroller, developed by General Instrument's Microelectronics. It has a RISC instruction architecture. Because of it's low cost and high availability, it's widely used globally.
- AVR Microcontroller
  - Developed by Alf-Egil Bogen and Vegard Wollan from Atmel Corporation. It has modified Harvard RISC architecture. The speed of AVR is high when compared to 8051 and PIC.

ATmega328 Microcontroller

- Features of ATmega328:
  - 28-pin AVR microcontroller
  - Flash program memory of 32kbytes
  - EEPROM data memory of 1kbytes
  - SRAM data memory of 2kbytes
  - I/O pins are 23
  - Two 8-bit timers
  - A/D converter
  - Six channel PWM
  - In built USART
  - External Oscillator: up to 20MHz

Arduino Platform

- Open Source electronic prototyping platform
- Based on flexible easy to use hardware and software
  - Arduino boards
  - Arduino programming language
  - Arduino Software (IDE)
- Inexpensive microcontroller platform for beginners



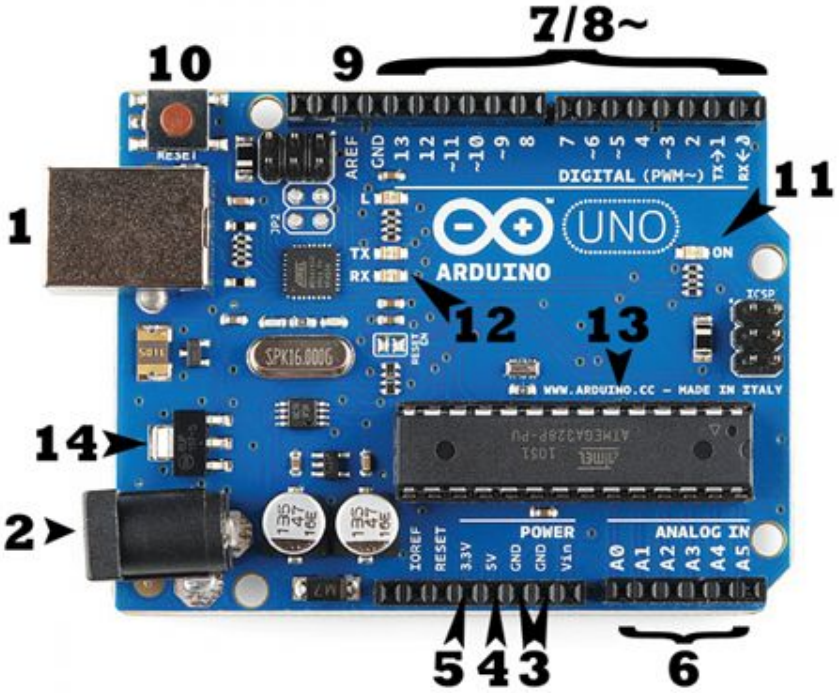
https://www.youtube.com/watch?v=CqrQmQqpHXc

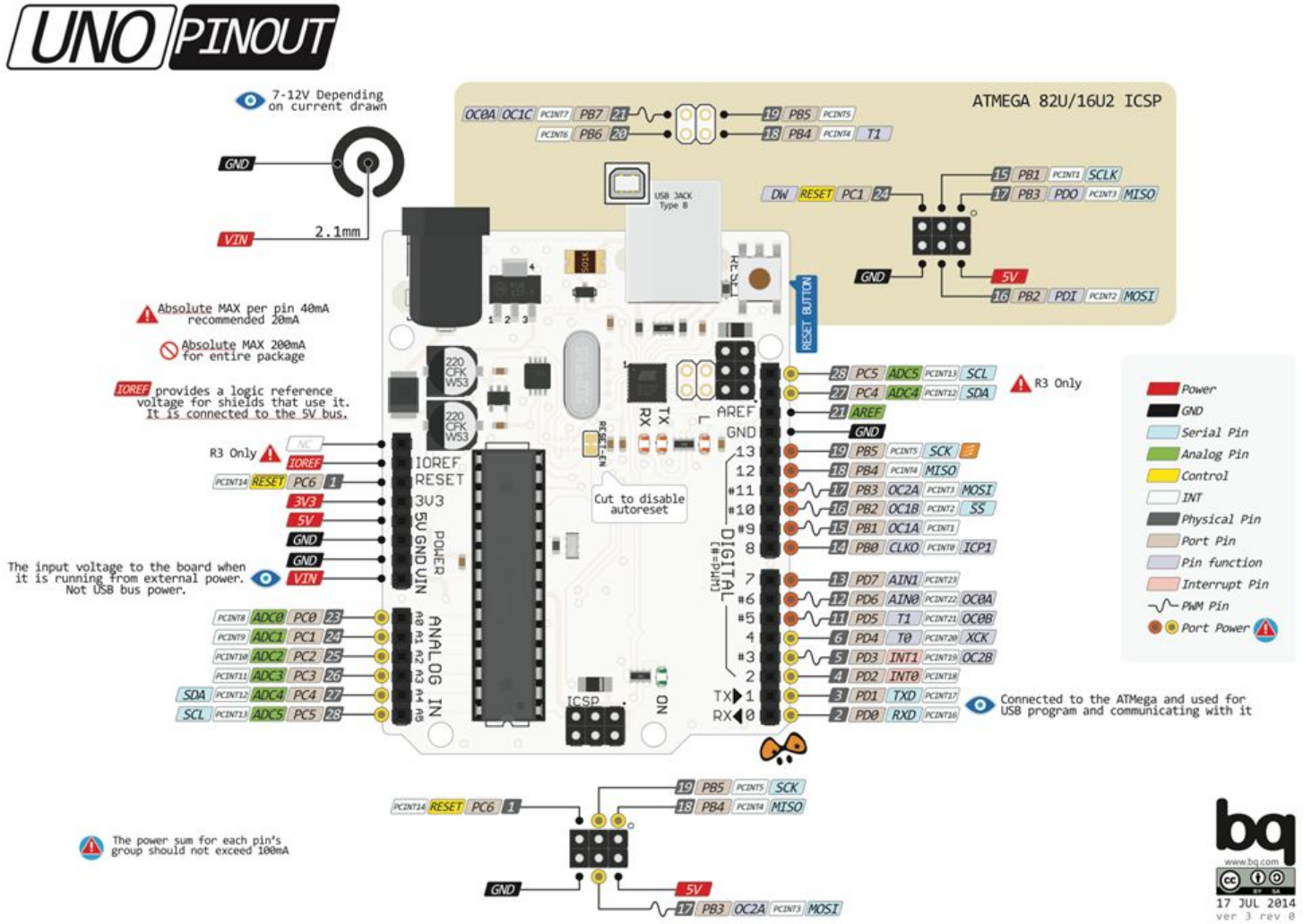
Arduino Boards

	Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
	Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
	Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
	Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
	Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
	Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
	Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1

Arduino Uno – Board Components

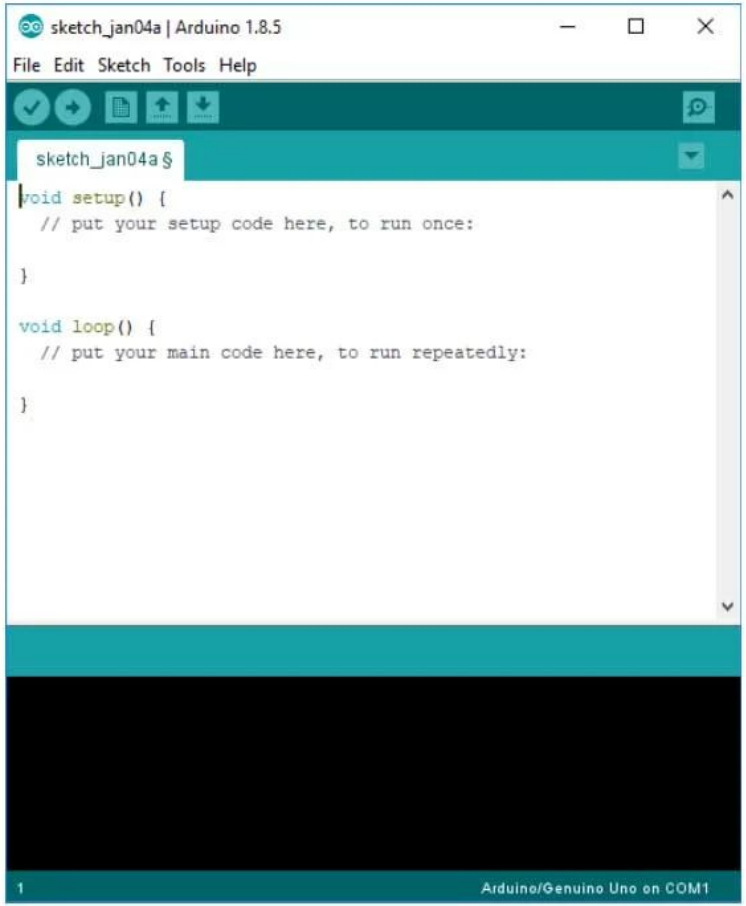
1. USB connection
2. Power supply barrel jack
3. GND (ground) pins
4. 5V pin supplies
5. 3.3V pin supplies
6. Analog In pins
7. Digital pins (in/out)
8. ~ Pulse-Width Modulation (PWM) pins
9. AREF Analog Reference
10. reset button
11. Power LED Indicator
12. TX (transmit) RX (receive) LEDs
13. Main Integrated Circuit
14. Voltage Regulator




UNO PINOUT
7-12V Depending on current drawn
GND
VIN
2.1mm
Absolute MAX per pin 40mA recommended 20mA
Absolute MAX 200mA for entire package
IOREF provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
R3 Only
The input voltage to the board when it is running from external power. Not USB bus power.
The power sum for each pin's group should not exceed 100mA
ATMEGA 82U/16U2 ICSP
USB JACK Type B
RESET BUTTON
Cut to disable autoreset
POWER
ANALOG IN
DIGITAL (PWM~)
ICSP
ON
TX
RX
Connected to the ATMega and used for USB program and communicating with it
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power
www.bq.com
17 JUL 2014
ver 3 rev 0

Arduino IDE

- Arduino sketches (synonym for program in Arduino language) are written in the Arduino Integrated Development Environment (IDE).
- The Arduino programming language is based on a very simple hardware programming language called processing.
- This is similar to the C language.
- After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution.



Learning Arduino

- Arduino Getting Started
  - https://www.arduino.cc/en/Guide/HomePage
- Arduino Tutorials
  - https://www.arduino.cc/en/Tutorial/HomePage
- Arduino Starter Kit - Video Tutorials
  - https://www.youtube.com/playlist?list=PLT6rF_I5kknPf2qlVFlvH47qHvqvzkknd
- Arduino in 15 minutes
  - https://www.youtube.com/watch?v=nL34zDTPkcs

Simulation Practicals

Why simulate?

- Testing before building
- Verifying before commiting
- Save time & money
- Essential tool in designing systems

Arduino in Proteus - https://www.instructables.com/id/How-to-Simulate-Arduino-in-Proteus/

Circuits on Tinkercad - https://www.tinkercad.com/circuits

Project Design...