

# Hardware-Software Codesign

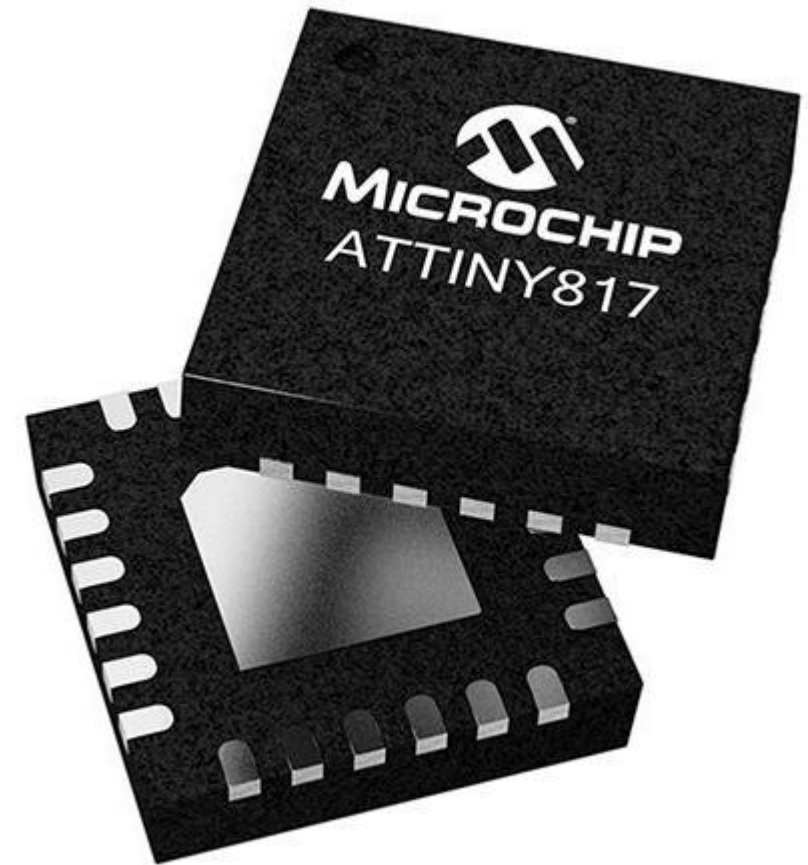
ICT 41205 Digital Control Systems

Nimal Skandhakumar

Faculty of Technology  
University of Sri Jayewardenepura

# 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>



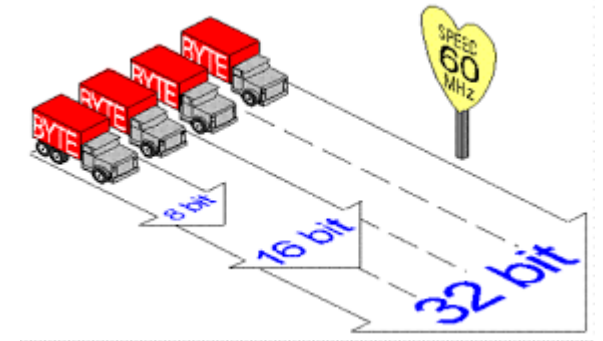
# Applications of Microcontrollers

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



# 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

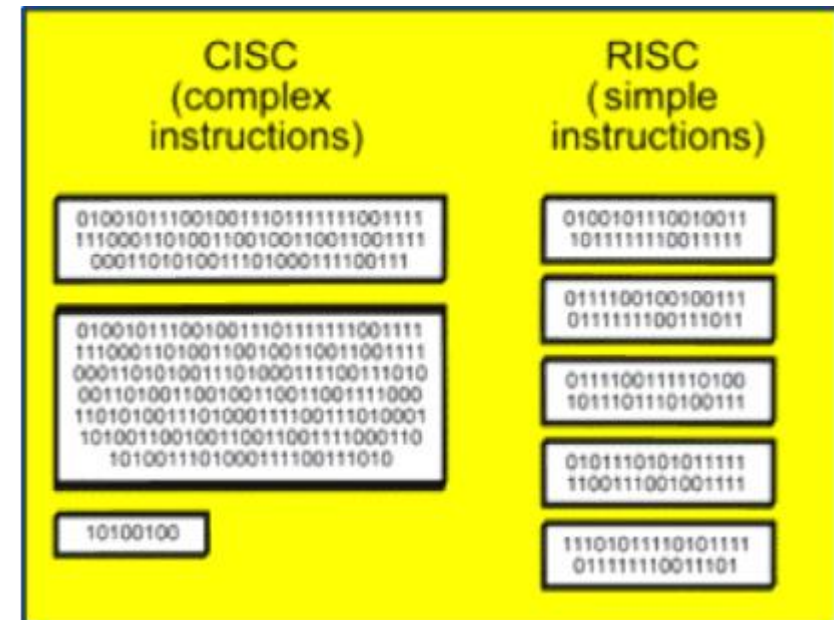
- Classification According to Memory Devices
  - Embedded memory microcontroller
    - Microcontroller unit has all the functional blocks such as program & data memory, I/O ports, serial communication, counters and timers and interrupts on the chip is an embedded microcontroller.
  - External Memory 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>

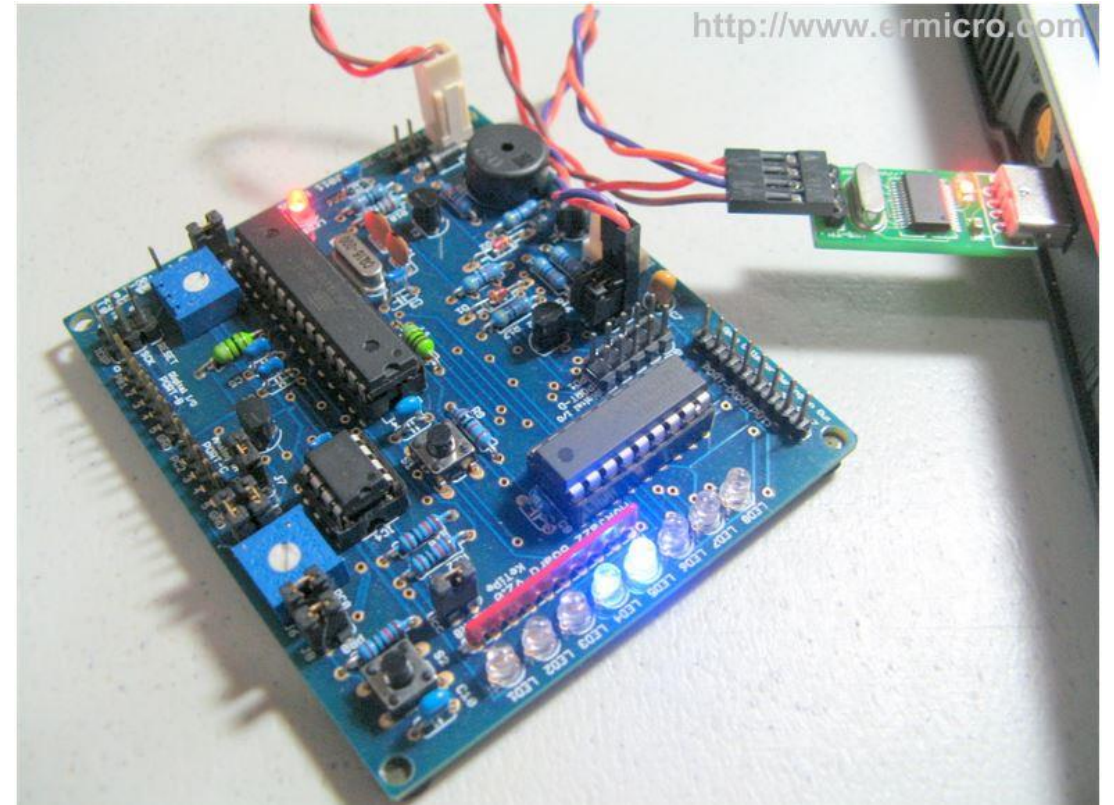


# RISC (reduced instruction set computer)

- 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



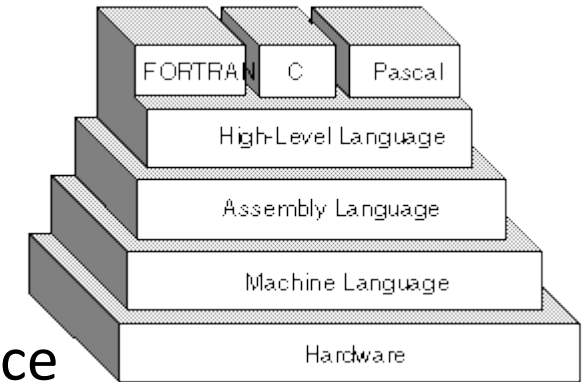
# Assembly language

- 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>

```
01 DATA SEGMENT
02     MESSAGE DB "HELLO WORLD!?!$"
03 ENDS
04
05 CODE SEGMENT
06     ASSUME DS:DATA CS:CODE
07 START:
08     MOV AX, DATA
09     MOV DS, AX
10     LEA DX, MESSAGE
11     MOV AH, 9
12     INT 21H
13     MOV AH, 4CH
14     INT 21H
15 ENDS
16 END START
17
```

# 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 its 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

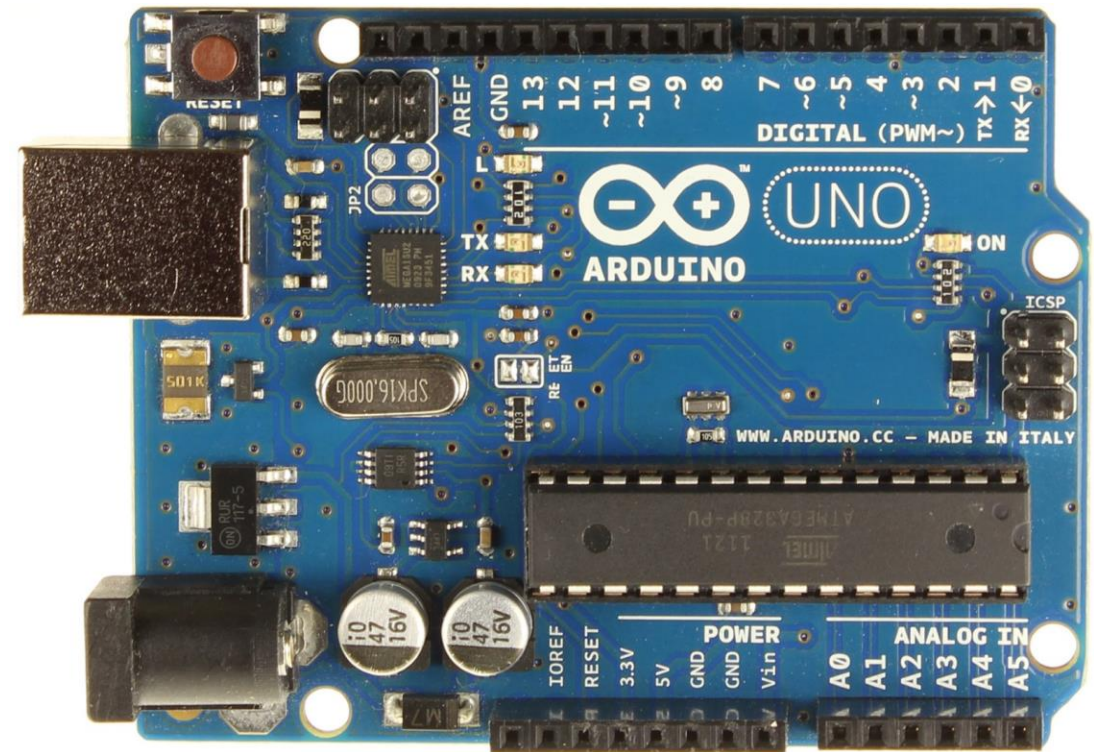
- Features of ATmega328:
  - 28-pin AVR microcontroller
  - Flash program memory of 32kbytes
  - EEPROM data memory of 1kbytes
  - SRAM data memory of 2kbytes
  - I/O pins are 23
  - Two 8-bit timers
  - A/D converter
  - Six channel PWM
  - In built USART
  - External Oscillator: up to 20MHz







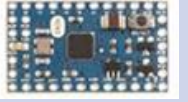



# Arduino Platform

- 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

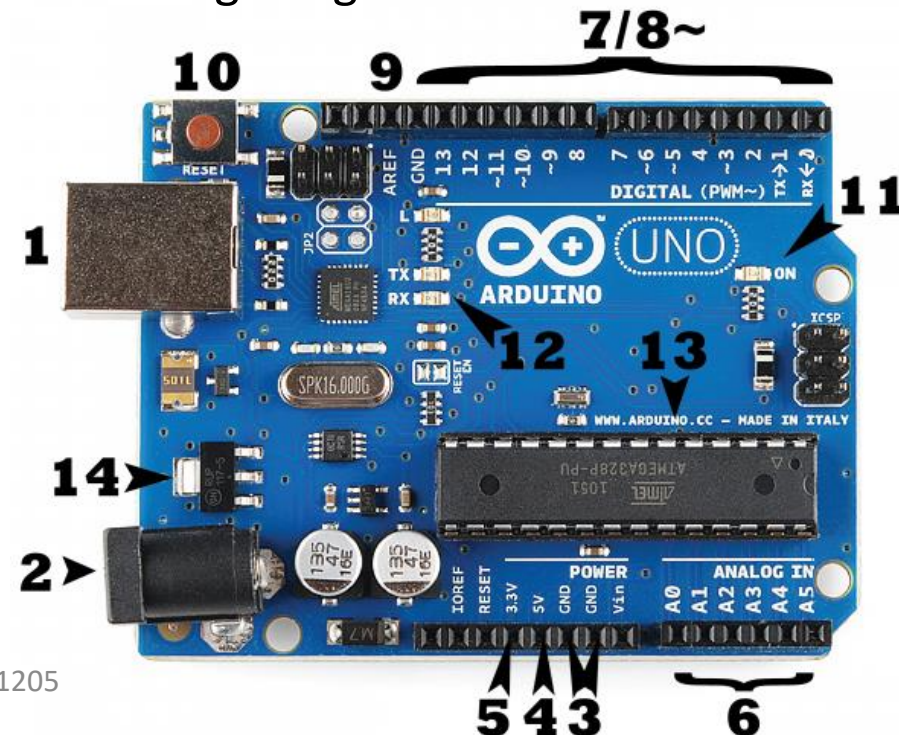
|   | Name                     | Processor               | Operating/Input Voltage | CPU Speed | Analog In/Out | Digital IO/PWM | EEPROM [kB] | SRAM [kB] | Flash [kB] | USB     | UART |
|---|--------------------------|-------------------------|-------------------------|-----------|---------------|----------------|-------------|-----------|------------|---------|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



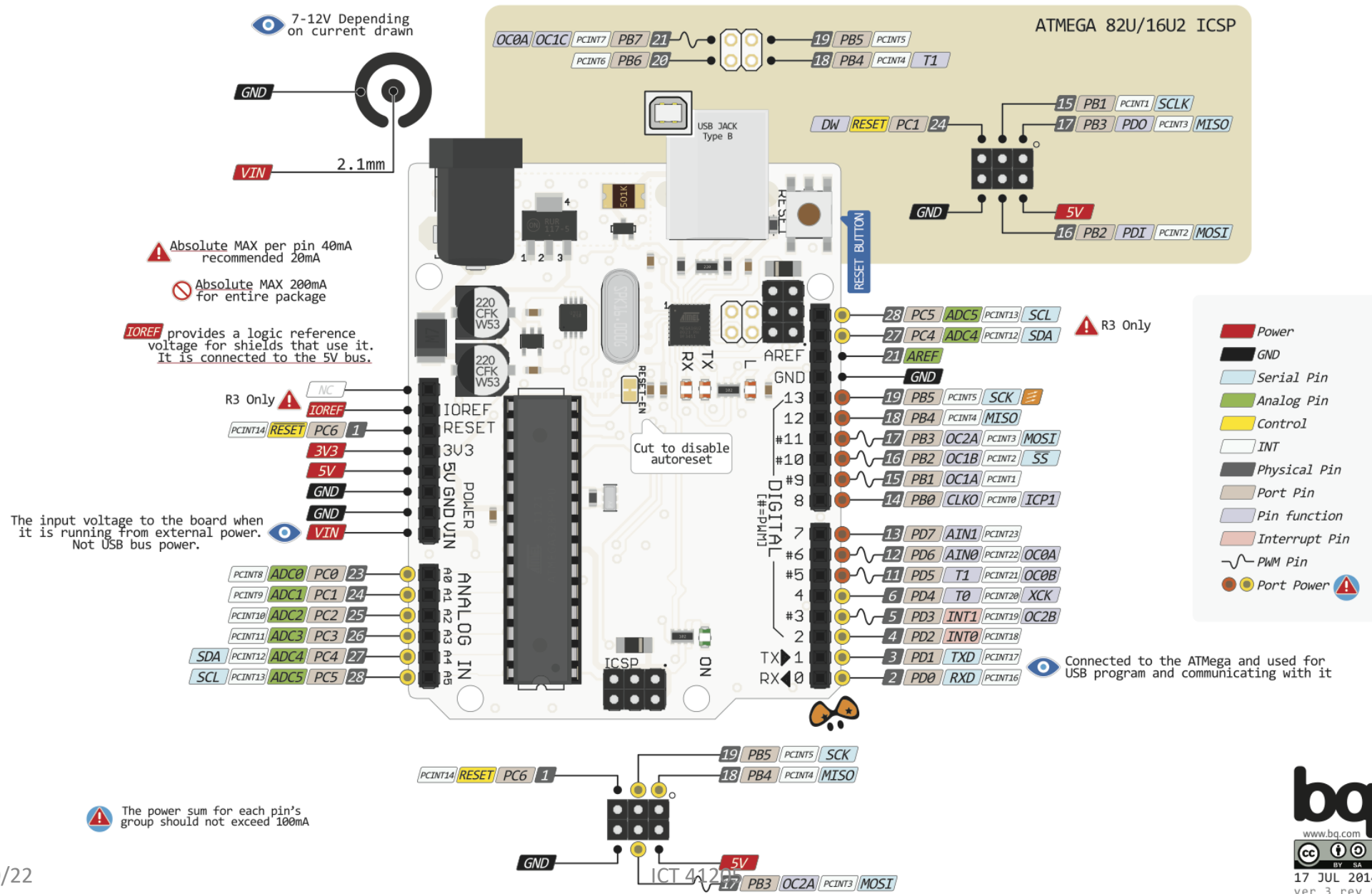
# Arduino Uno – Board Components

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

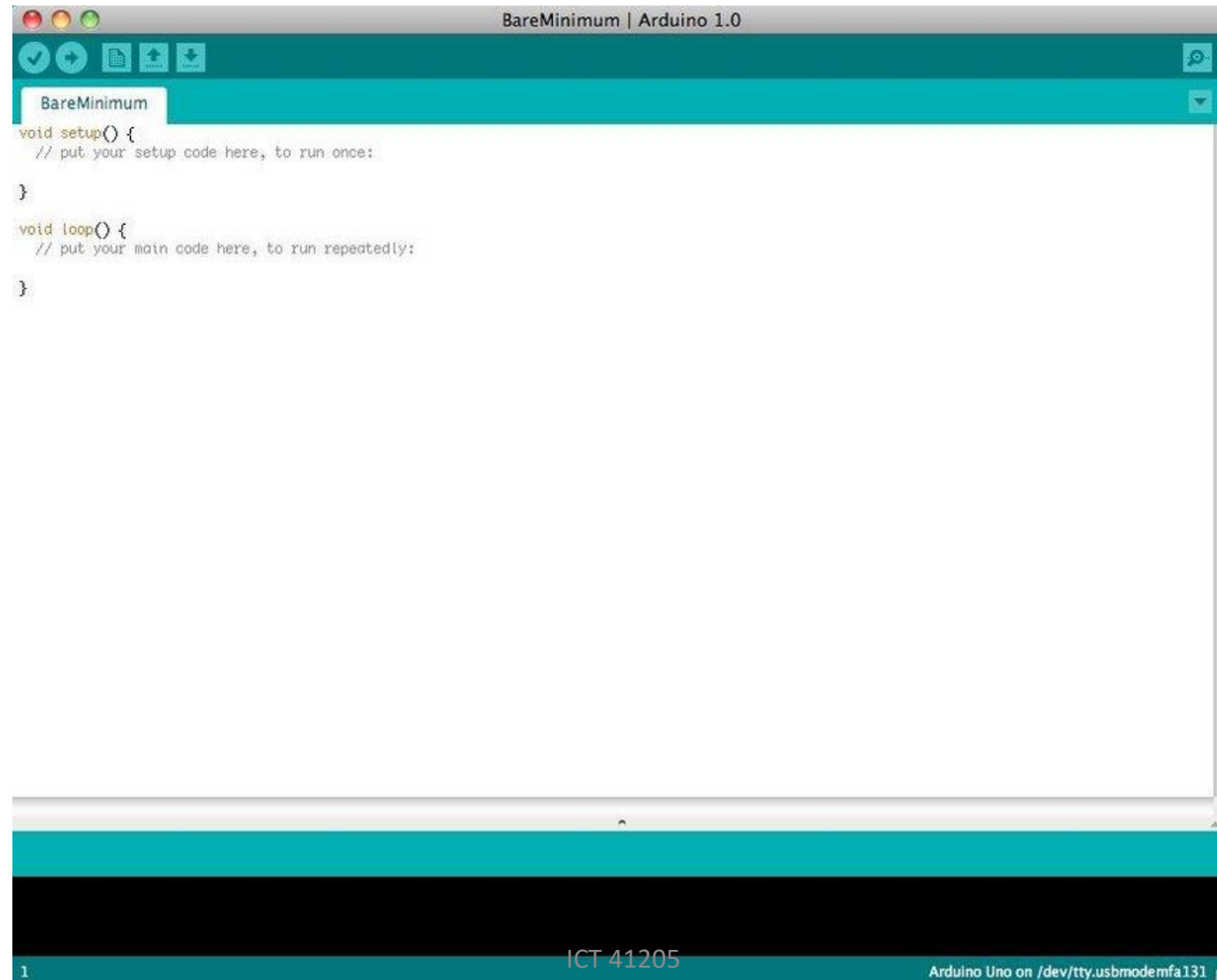
11. Power LED Indicator
12. TX (transmit) RX (receive) LEDs
13. Main Integrated Circuit
14. Voltage Regulator



**UNO** **PINOUT**



# Arduino IDE



# 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\\_I5kknPf2qIVFlvH47qHvgvzkknd](https://www.youtube.com/playlist?list=PLT6rF_I5kknPf2qIVFlvH47qHvgvzkknd)
- Arduino in 15 minutes
  - <https://www.youtube.com/watch?v=nL34zDTPkcs>