

Prof Associate ,  
School of information engineering Jiangxi university of  
science and technology, China

EMAIL: [ajm@jxust.edu.cn](mailto:ajm@jxust.edu.cn)

## Instrumentation and Sensor Networks:

**ENGT5105**

**ARDUINO VS EMBEDDED SYSTEMS  
IN INSTRUMENTATION WORLD**

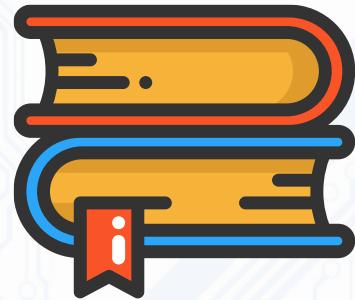
**Dr Ata Jahangir Moshayedi**

**Autumn \_2021**



江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



# Instrumentation and Sensor Networks:

## ENGT5105

### LECTURE 01: ARDUINO VS EMBEDDED SYSTEMS IN INSTRUMENTATION WORLD

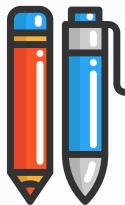


DE MONTFORT  
UNIVERSITY  
LEICESTER



江西理工大学 信息工程学院

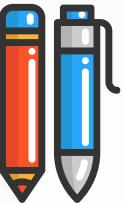
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



# Agenda

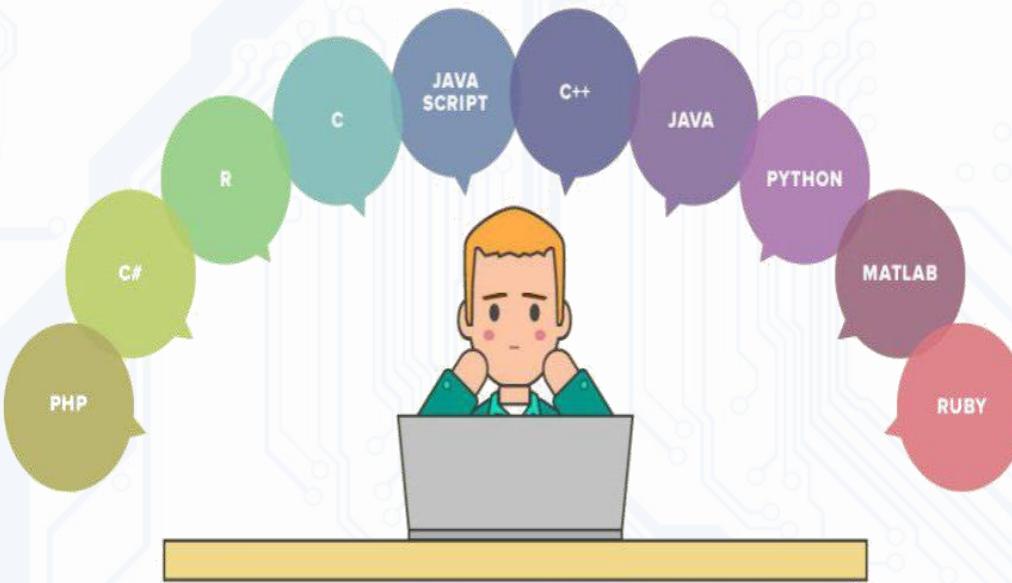
- A brief about embedded system their classification and applications
- What is Microcontroller??
- Introduction to Arduino and Arduino types
- Which Arduino do you need for your project?
- How to run your first code
- Introduction to tinker cad as Arduino simulator and some examples





# Nihau

- Let us have a brief view to our course
- Don't worry we will learn lots of thing this semester



你好

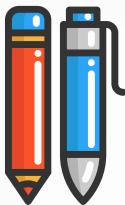
Nǐ hǎo



DE MONTFORT  
UNIVERSITY  
LEICESTER



江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



# Who am i?

## About me



Dr. Ata Jahangir Moshayedi

Researcher in the field of robotic and Automation  
**Ph.D. Electronic , in the Field of Mobile olfaction system**  
Pune University, India  
MSc. Instrumentation  
BE. Power electronic

### Prof Associate :

School of information engineering Jiangxi university of science and technology, China  
E-mail: ajm@jxust.edu.cn

**Dr Ata Jahangir Moshayedi**, Associate professor at Jiangxi University of Science and Technology, China, PhD. In Electronic Science from Savitribai Phule Pune University, India.

IEEE member, Instrument Society of India as a Life Member, Lifetime Member of Speed Society of India, member of the editorial team of various conference and journals like; International Journal of Robotics and Control, JSME, Bulletin of Electrical Engineering and Informatics, International Journal of Physics and Robotics Applied Electronics, etc., 80 papers published in national journals and conferences, 2 books published, Owns 1 patent, 5 copyright.

His research interest includes: Robotics and Automation/ Sensor modelling /Bio-inspired robot, Mobile Robot Olfaction/Plume Tracking, Embedded Systems / Machin vision-based Systems/Virtual reality, Machine vision/Artificial Intelligence

Dr Moshayedi, Presently working on his AGV (Automated Guided Vehicles) designed model and Food delivery service robot at Jiangxi University, china.

### 1. HONORS / AWARDS

- 2021: National Festival of Creativity, Innovation and Entrepreneurship ([stideas.ir](http://stideas.ir)) Selected as Ideas worthy of international acclaim
- 2019\_2020第11届中国大学生服务外包创新创业大赛,The 11th China College Students Service Outsourcing Innovation and Entrepreneurship, Competition, Rank: Second prize in central China area
- 2019\_2020年全国大学生“互联网+”创新大赛 联第七届“发现杯”全国大学生互联网软件设计大奖national University Students “Internet ” Innovation Competition and the 7th “Discovery Cup” National College Students Internet Software Design Competition in 2019, rank: Second prize in Central China.
- Best Invention for “Design of ergonomic mattress recognizer system ” in 7<sup>th</sup> festival of inventors imitator innovators Azad university,5-7 March,2019Najafabad branch, Iran
- Best robot Demo presentation in International Conference on Advanced Robotics and Intelligent Control (ICARIC 2018)Jishou, Hunan, China 11-14 Oct 2018
- 3<sup>rd</sup> rank IRANOPEN 2018; international ROBOTIC GAME UAV OUTDOOR, Tehran, Iran ,March 2018
- Best paper presentation(2 paper) ; The fifth international conference of electrical engineering and computer with emphasis of native knowledge, Alame tabatabae ,Feb 2018
- Selected Design ,Pipe inspection Robot, ICROM 2017,The 5th International Conference on Robotics and Mechatronics, Tehran, Iran November, 2017
- Distinguished member of Young Researchers and Elite Club ,2017,2018
- Distinguished member of Roshd center, Azad University Khomeinshahr, Iran 2017,2018
- First awarded paper in energy conference, Azad University khomeinshahr ,Iran 2015
- First awarded Design for design and development of Solar Refrigerator from Young Researchers and Elite Club,2016
- Founder of Kahroba Sante Espadan(KSE), elite company ,Iran 2015
- Distinguished student from represenstor of science and education, Iran
- During MSC course 2008, 2009
- Ph.D. Course 2010,2011,2013, 2014(silver grade)
- Avishkar 2012 \_OMC ( odor movie camera) selected for round III state level from Pune university ,India
- Ranked 5 th in MSC , Department of instrumentation science (among 47 participants) ,India
- Avishkar 2009,Design and development of fuzzy spin coater (team working) selected for round III state level from Pune university India
- Candida as the selected presentation for the interview of scientific prize of satish bihadte ,Pune university ,2009
- Distinguished Staff in T/L (transmission line) section in BARSAN company, Tehran, Iran,2006
- Candidate for the Khwarizmi Science Prize 2005 Selected research topic from Azad University for design and implementation of Labyrinth robot , Iran

# Who Am I?

**Dr. Ata Jahangir Moshayedi**

Ph.D. Electronic , in the Field of Mobile olfaction system Pune University, India

**Prof Associate :**

- S 203, School of information engineering Jiangxi university of science and technology, China

**EMAIL:**

**Academic:**

ajm@jxust.edu.cn

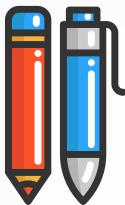
**Personal:**

drajm@yahoo.com



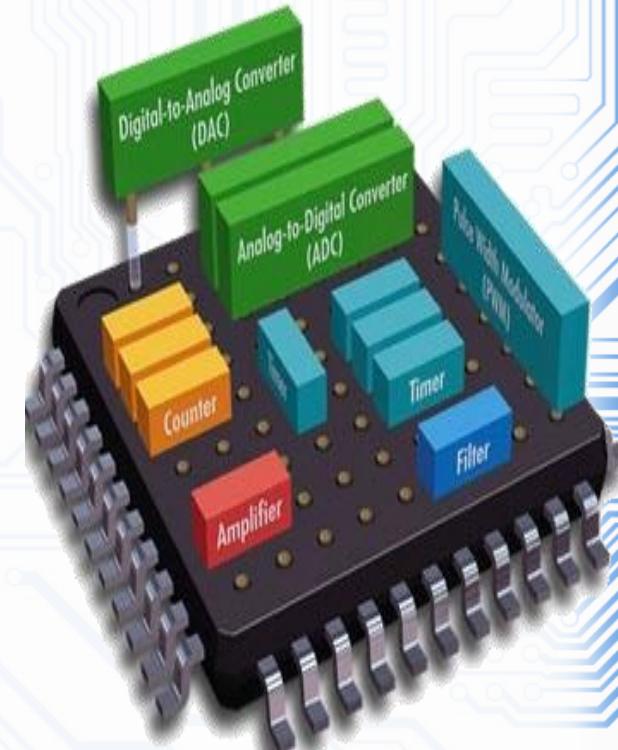
Researcher in the field of robotic and Automation  
PhD. Electronics since  
MSc. Instrumentation  
BE. Power electronic

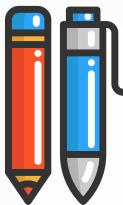
**Web page:** [www.ajmoshayedi.ir](http://www.ajmoshayedi.ir)



# A Brief About Embedded System

- An embedded system is an electronic system that has a software and is embedded in computer hardware.
- It is programmable or non-programmable depending on the application.
- An Embedded system is defined as a way of working, organizing, performing single or multiple tasks according to a set of rules.
- In an embedded system, all the units assemble and work together according to the program.



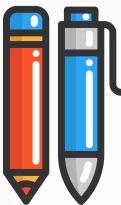


# A Brief About Embedded System

Examples of embedded systems include numerous products such as

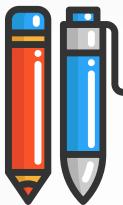
- Microwave ovens
- Washing machine
- Printers
- Automobiles
- Cameras, etc.

These systems use microprocessors, microcontrollers as well as processors like DSPs.



# A Brief About Embedded System

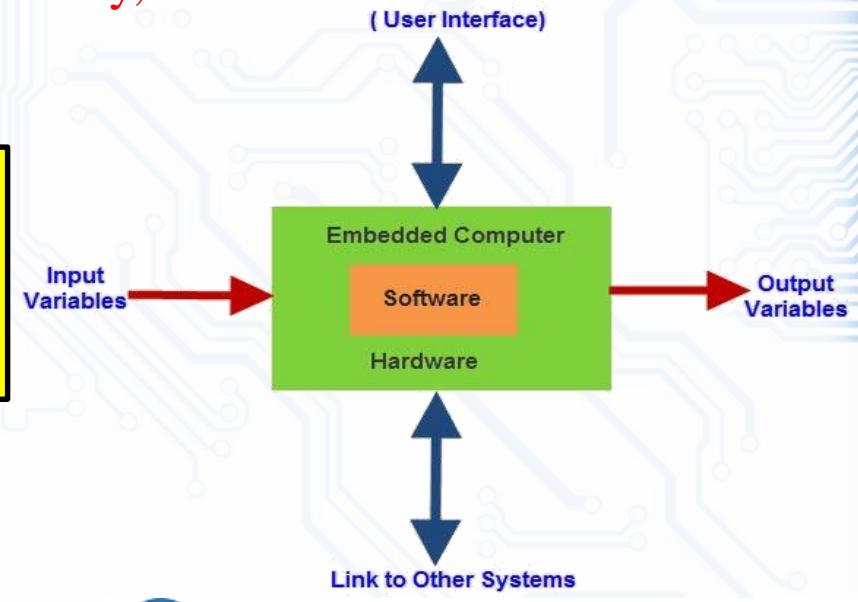
- The important characteristics of an embedded systems are:  
**speed, size, power, reliability, accuracy, adaptability.**
- Therefore, when the embedded system performs the operations at high speed, then it can be used for real -time applications.
- **The Size of the system and power consumption should be very low, then the system can be easily adaptable for different situations.**

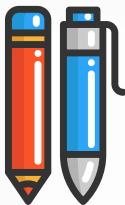


# What is an embedded system?

- An Embedded system is a combination of computer hardware and software. As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.
- The Embedded system hardware includes elements like user interface, Input/output interfaces, display and memory, etc.

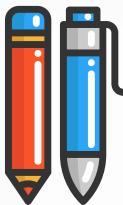
Generally, an embedded system comprises power supply, processor, memory, timers, serial communication ports and system application specific circuits.





# A Brief About Embedded System

- Embedded system software is written in a high-level language, and then compiled to achieve a specific function within a non-volatile memory in the hardware.
- Embedded system software is designed to keep in view of three limits.
- They are availability of system memory and processor speed. When the system runs endlessly, there is a need to limit the power dissipation for events like run, stop and wake up.

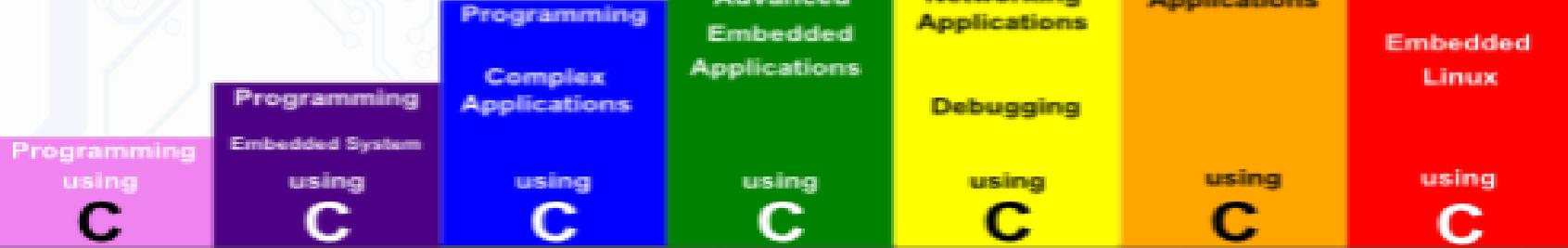


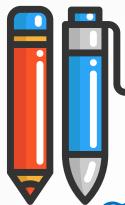
# A Brief About Embedded System

## Professional Embedded Engineer

### Seven Step

### Success Solution



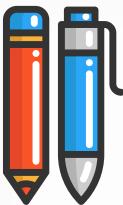


# Examples of embedded system

Some examples of embedded systems are below:-

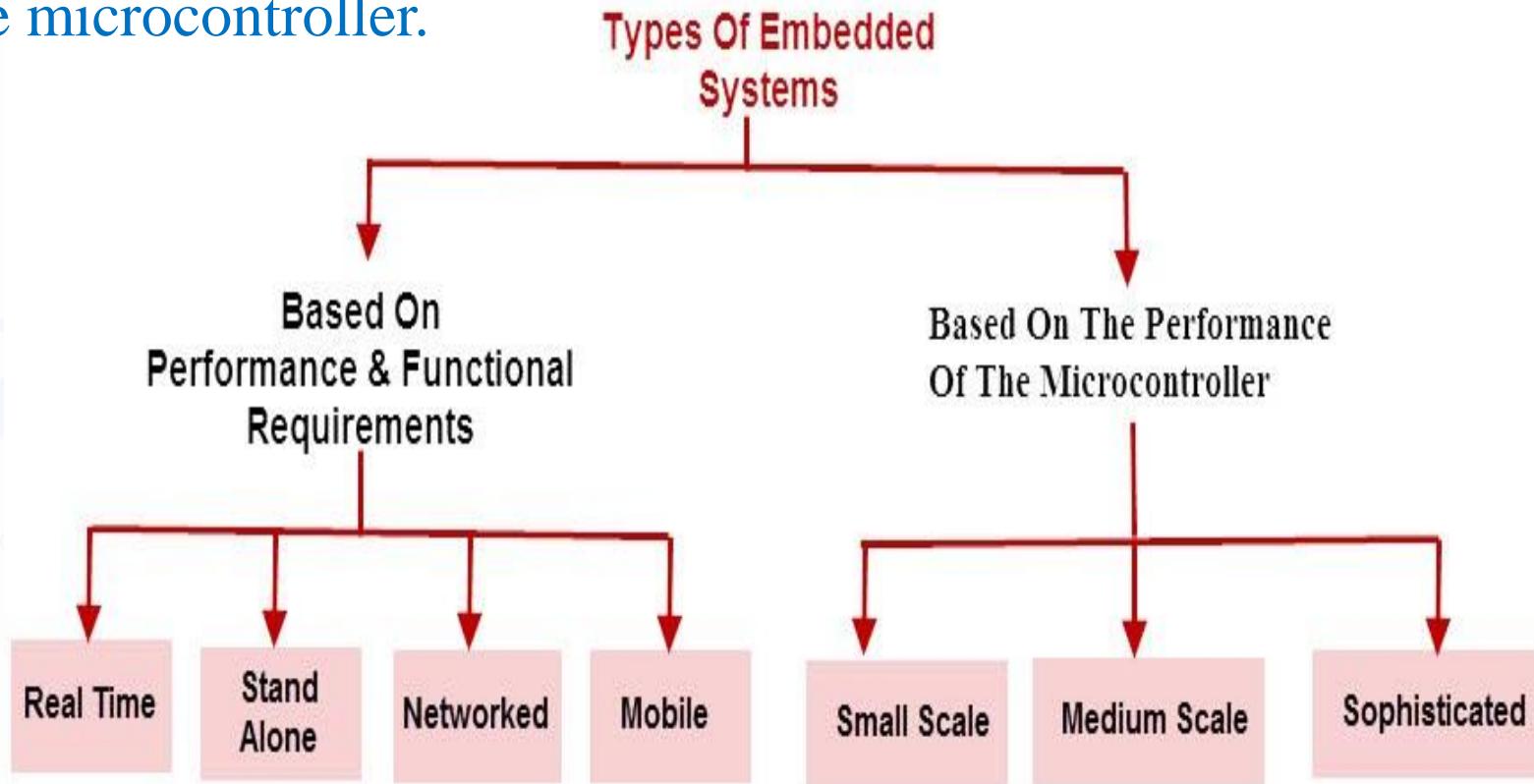


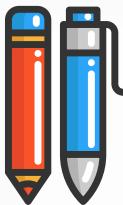
- ATM
- Digital Cameras
- Microwave ovens
- Factory controllers
- Washing machine
- Calculator
- TV remote
- Traffic lights
- Digital watches
- Mp3 player
- video games consoles
- Printers
- GPS receivers
- Dishwashers
- Thermostats
- Anti-lock banking system
- Medical imaging



# Types of Embedded Systems

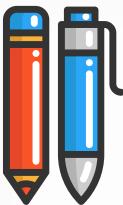
Embedded systems can be classified into different types based on performance, functional requirements and performance of the microcontroller.





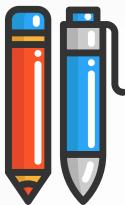
# Embedded systems are classified

- Embedded systems are classified into four categories based on their performance and functional requirements:
  - *Stand alone embedded systems*
  - *Real time embedded systems*
  - *Networked embedded systems*
  - *Mobile embedded systems*
- Embedded Systems are classified into three types based on the performance of the microcontroller such as
  - *Small scale embedded systems*
  - *Medium scale embedded systems*
  - *Sophisticated embedded systems*



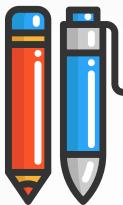
# Stand Alone Embedded Systems

- Stand alone embedded systems do not require a host system like a computer, it works by itself.
- It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices.
- *Examples for the stand alone embedded systems are:*
  - *mp3 players,*
  - *digital cameras*
  - *video game consoles*
  - *microwave ovens*
  - *temperature measurement systems.*



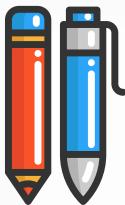
# Real Time Embedded Systems

- A real time embedded system is defined as, a system which gives a required o/p in a particular time.
- These types of embedded systems follow the time deadlines for completion of a task.
- Real time embedded systems are classified into two types such as soft and hard real time systems.



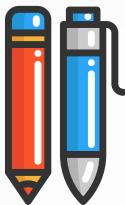
# Networked Embedded Systems

- These types of embedded systems are related to a network to access the resources. The connected network can be LAN, WAN or the internet.
- **The connection can be any wired or wireless. This type of embedded system is the fastest growing area in embedded system applications.**
- The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser.
- **Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP**



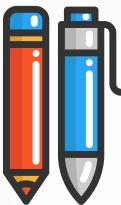
# Mobile Embedded Systems

- Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc.
- The basic limitation of these devices is the other resources and limitation of memory.



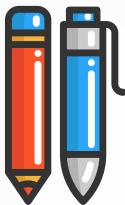
# Small Scale Embedded Systems

- These types of embedded systems are designed with a single 8 or 16-bit microcontroller, that may even be activated by a battery.
- For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).



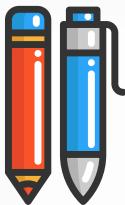
# Medium Scale Embedded Systems

- These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs.
- These types of embedded systems have both hardware and software complexities.
- For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.



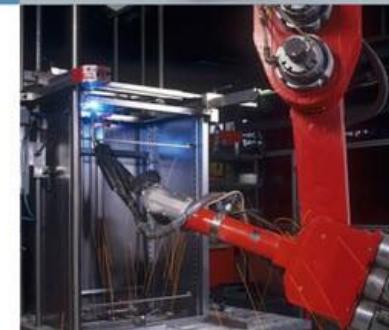
# Sophisticated Embedded Systems

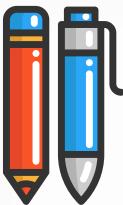
- These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors.
- They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.



# Applications of Embedded Systems:

**Embedded systems** are used in different applications like automobiles, telecommunications, smart cards, missiles, satellites, computer networking and digital consumer electronics.





# Embedded Systems

- **Embedded Systems in Automobiles and in telecommunications**
  - Motor and cruise control system
  - Body or Engine safety
  - Entertainment and multimedia in car
  - E-Com and Mobile access
  - Robotics in assembly line
  - Wireless communication
  - Mobile computing and networking

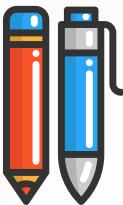
- **Embedded Systems in Smart Cards, Missiles and Satellites**
  - Security systems
  - Telephone and banking
  - Defence and aerospace
  - Communication



**DE MONTFORT  
UNIVERSITY**  
LEICESTER

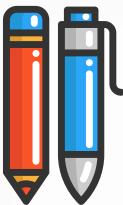


**江西理工大学 信息工程学院**  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



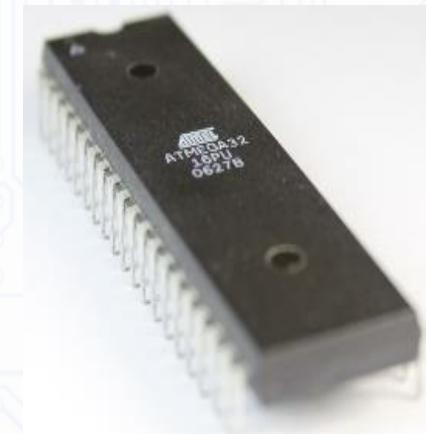
# Embedded Systems

- **Embedded Systems in Peripherals & Computer Networking**
  - Displays and Monitors
  - Networking Systems
  - Image Processing
  - Network cards and printers
- **Embedded Systems in Consumer Electronics**
  - Digital Cameras
  - Set top Boxes
  - High Definition TVs
  - DVDs

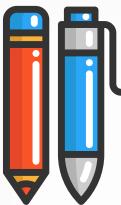


# What is Microcontroller??

- Microcontrollers are small computers integrated into a single chip
- They contain
  1. Processing core
  2. Flash Memory for program
  3. I/O peripherals
  4. RAM
  5. 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.



# Why embedded systems? Why Arduino?

- Today embedded systems are replacing various systems that used to be designed with a set of complex electronic circuits.
- Usually the heart of the embedded system is a microcontroller. One example of a microcontroller is Arduino.
- Arduino is an open source based prototyping platform used to sense and control physical devices.

## Development of an Arduino-Based Embedded System

27 Pages • Posted: 19 Jul 2019

Amrita Prakash

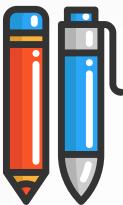
Patna Women's College

Ajit Singh

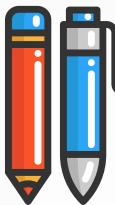
INSTITUT de DIPLOMATIE PUBLIQUE, United Kingdom

Date Written: July 15, 2019





- Have you ever wondered why Arduino is so popular? Let's have a closer look together.
- Arduino has been around for more than *10 years* now, and since then it has changed the world of *Rapid Prototyping* and *Embedded Studies*. With over *54 Million* Downloads of the *Arduino IDE*, it is no surprise that the Arduino Microcontrollers are here to stay.
- Back in the days, I used to work as a prototyping Engineer to develop US patents. My job was to prove the concepts and evaluate the feasibility of the patent. And believe it or not, most of our electronics builds included an Arduino. But why?
- Here are some reasons why it is so popular.
- Arduino has a very small learning curve.
- Its software is easy to use.
- Its Open-Source
- There are tons of online tutorials
- Documentation is solid
- Very Cheap



# Arduino: WHAT IT IS EXACTLY

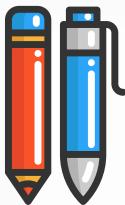
- The best way to explain what an Arduino is will be to start with what you can use it for.
- Put simply, an Arduino is a tool for controlling electronics. Think about a pencil. A pencil is a tool to help you write stuff.
- You need to write something down so you could grab a pencil. Same idea with Arduino. But Arduino is a tool for controlling electronics.



DE MONTFORT  
UNIVERSITY  
LEICESTER

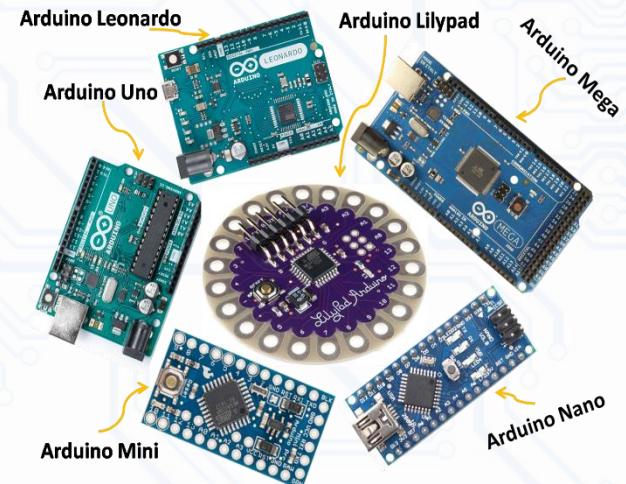


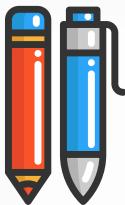
江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



# Why Arduino?

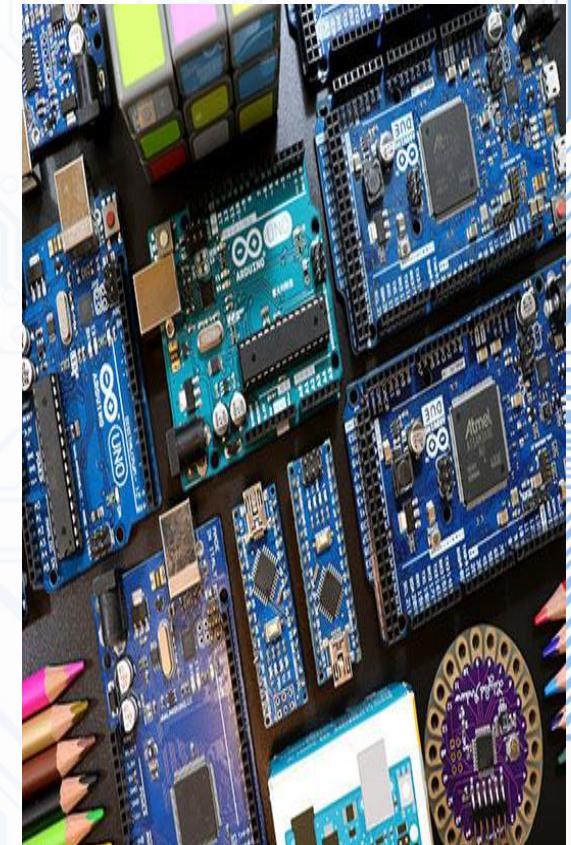
- Arduino is a great tool for developing **interactive objects**, taking inputs from a variety of switches or sensors and controlling a variety of lights, motors and other outputs. Arduino projects can be stand-alone or they can be connected to a computer using USB.
- The Arduino board is a microcontroller board, which is a small circuit (the board) that contains a whole computer on a small chip (the microcontroller).
- There are different versions of the Arduino board: they are different in components, aim and size, etc.
- Some examples of Arduino boards are: **Arduino Diecimila**, **Arduino Duemilanove**, **Freeduino**, **Arduino NG** and lot more.
- Arduino schematics are distribute using an open license so anyone is free to build his own Arduino compatible board.

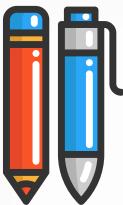




# Why Arduino?

- It is an **open-source project**, software/hardware is extremely **accessible** and very flexible to be customized and extended
- It is **flexible**, offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs
- It is **easy to use**, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers
- It is **inexpensive**, around 30 euro per board and comes with free authoring software
- Arduino is backed up by a growing **online community**, lots of source code is already available and we can share and post our examples for others to use, too!

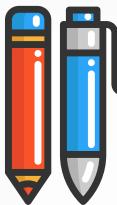




# The Importance of Arduino

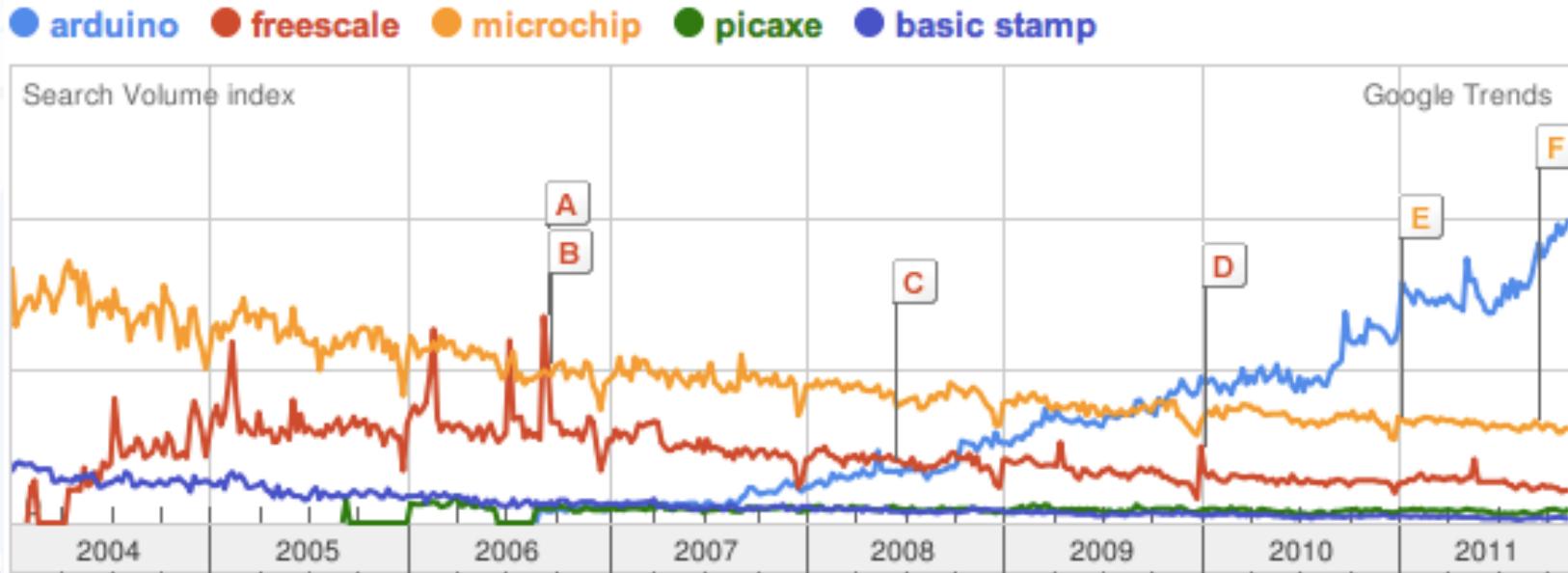
- While some of these Arduino projects may seem frivolous, the technology taps into several trends that will make it a potentially important force in the industry.
- The Internet of Things (IoT) is a popular phrase used in the tech community to describe everyday items that are connected to the internet and able to share information.
- Smart energy meters are an often-used example, which could regulate appliance usage to save money on energy.

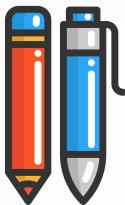




# Statistics

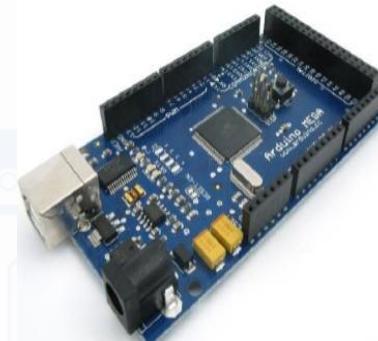
- By the Numbers
  - Year    Units sold
  - 2005    200
  - 2006    10,000
  - 2010    120,000
  - 2011    300,000(As of May 2011)
- Google trends data comparing Arduino with its biggest competitors



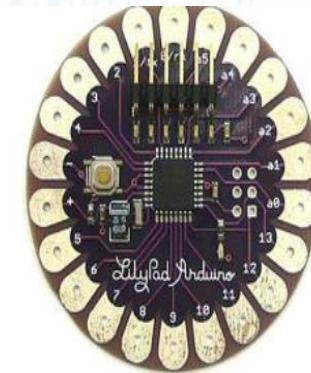


# Different flavours of Arduino !!!

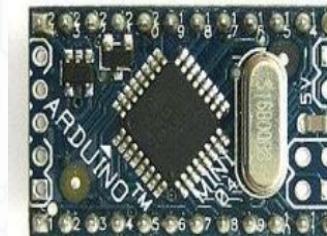
There are many versions of Arduino board. versions differ by size,microcontroller,etc



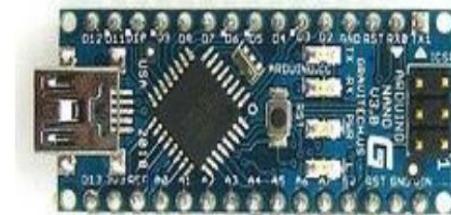
MEGA



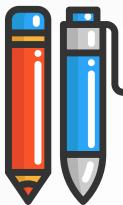
LILYPAD



MINI

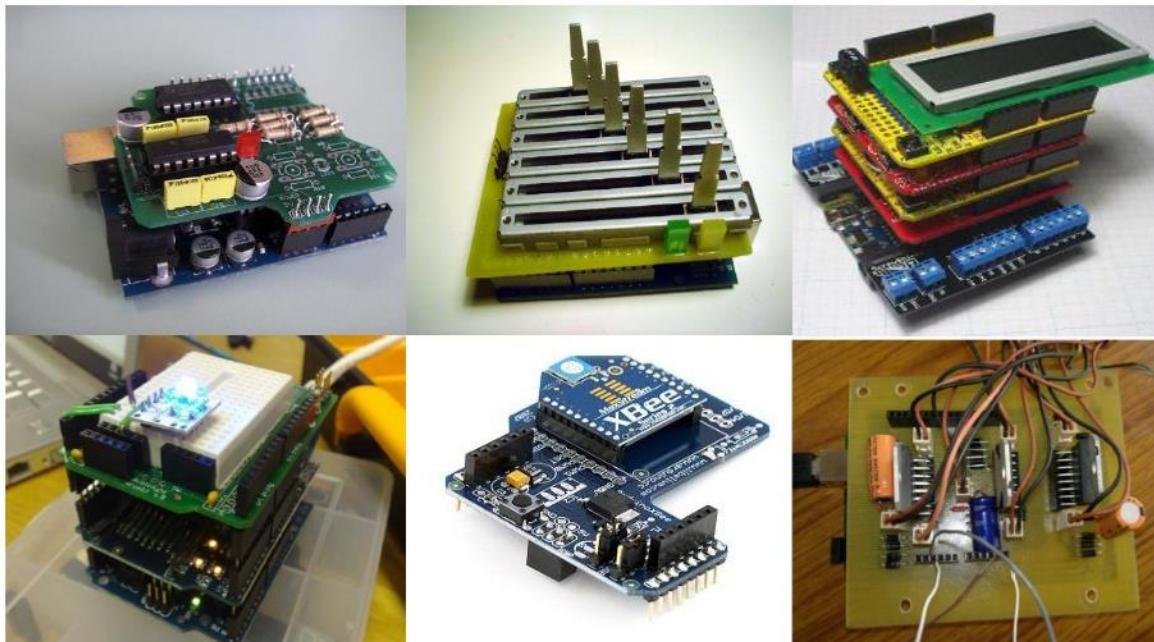


NANO 43mm x 18mm



# Arduino shields

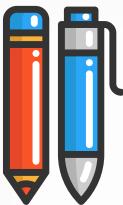
- Printed circuit boards that sit atop an arduino
- Plug into the normally supplied pin-headers of arduino.
- These are expansions to the base Arduino.
- For example: Arduino Ethernet shield, Xbee Shield, Touch Shield etc



DE MONTFORT  
UNIVERSITY  
LEICESTER

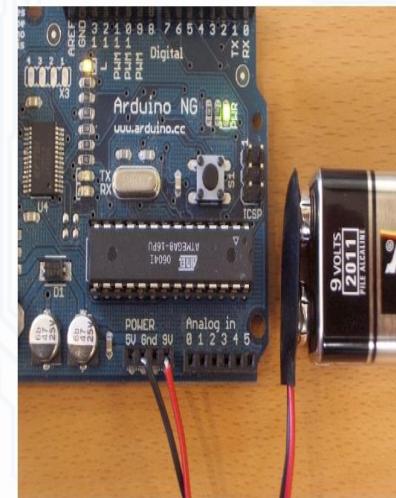
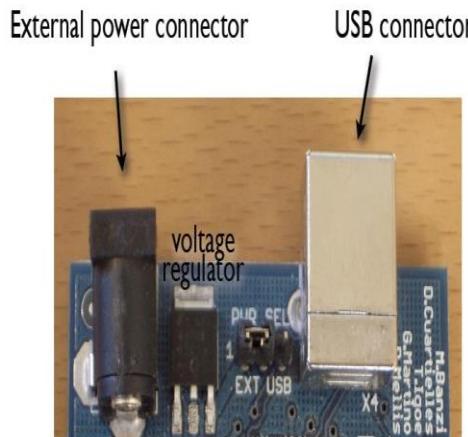


江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



# External power

- Should be between 9V and 12V DC.
- Must be rated for a minimum of 250mA current output.
- Must have a 2.1mm power plug on the Arduino end.
- The plug must be “centre positive”, that is, the middle pin of the plug has to be the + connection



Be careful about polarity! And shorts!



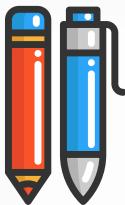
also solves polarity concerns



DE MONTFORT  
UNIVERSITY  
LEICESTER

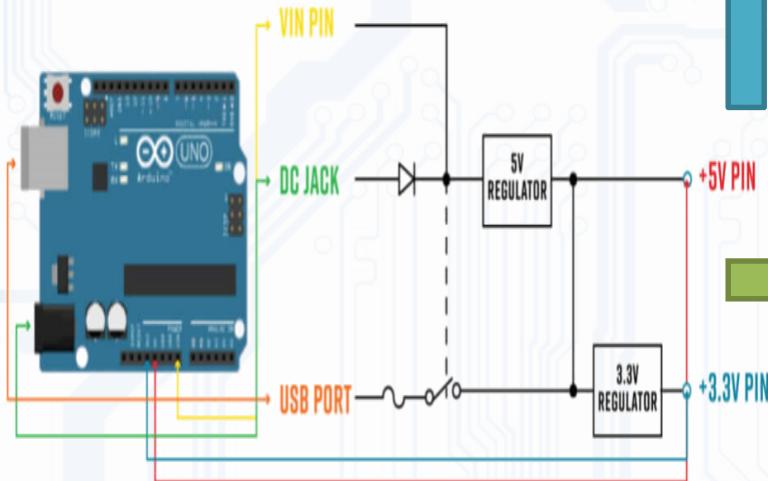


江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

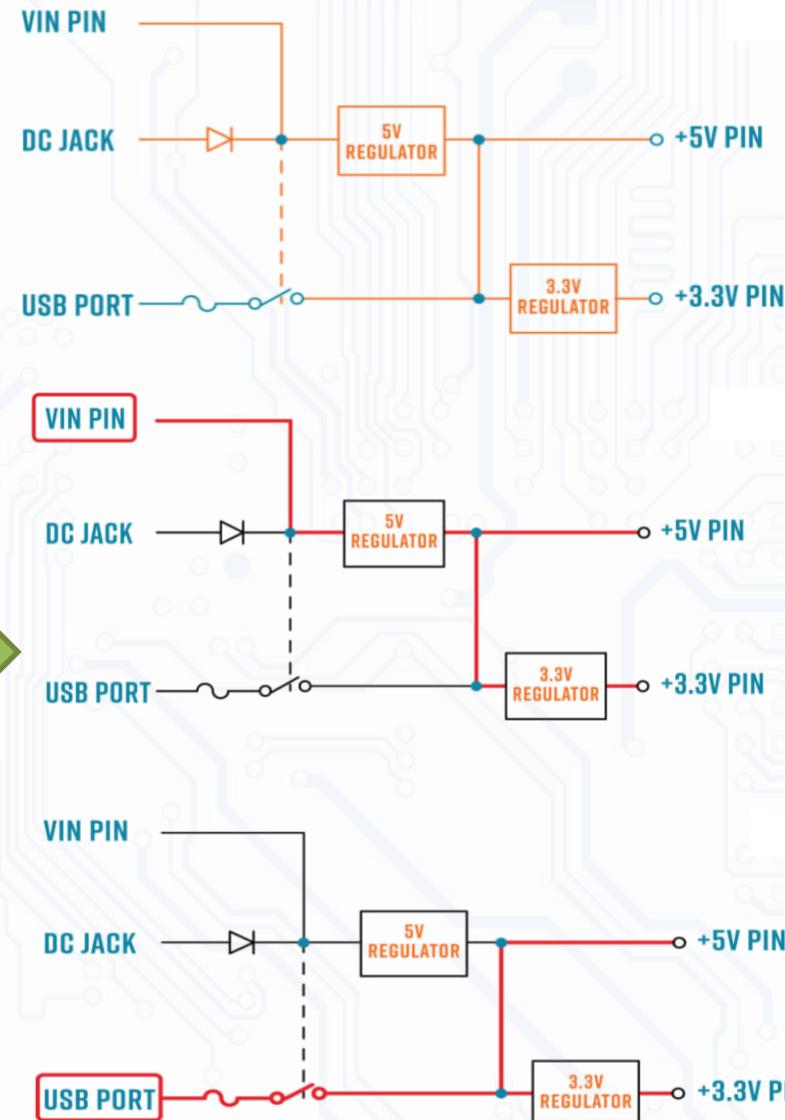


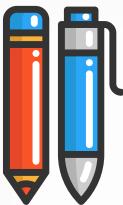
# External power

- As far as current goes USB can provide is 500mA for USB 1 and 2, and 900mA for USB3.
- USB1.x and USB2 provide 5V +/-5% with 500 mA maximum current
- USB3 provides 5V +/-5% with 900 mA maximum current



| Minimum Input voltage | Maximum Input voltage | Maximum Output current |       |
|-----------------------|-----------------------|------------------------|-------|
| +5V regulator         | 6.2V                  | 20V                    | 1A    |
| +3.3V regulator       | 3.58V                 | 16V                    | 150mA |

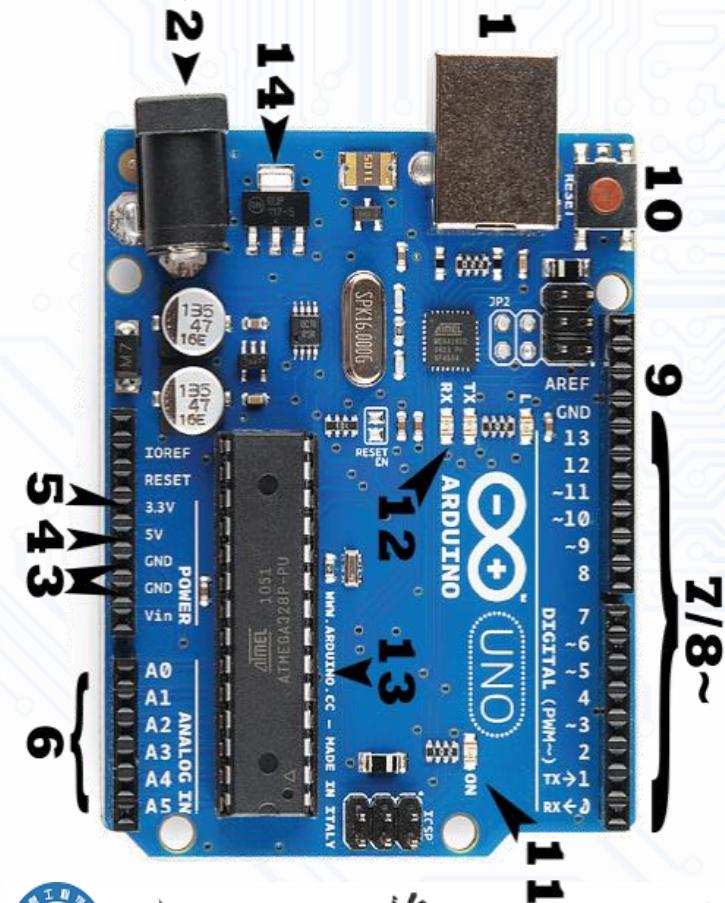


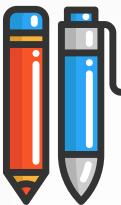


# Arduino from near look ?

- (1),(2) Power (USB / Barrel Jack)
- Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)
  - (3) GND
  - (4) 5V & (5) 3.3V
  - (6 )Analog
  - (7 )Digital
  - (8) PWM : (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO).
- (9) AREF :
- (10)Reset Button
- (11)Power LED Indicator
- (12)TX RX LEDs
- (13)Main IC
- (14)Voltage Regulator

Most of this part are common in Arduino boards





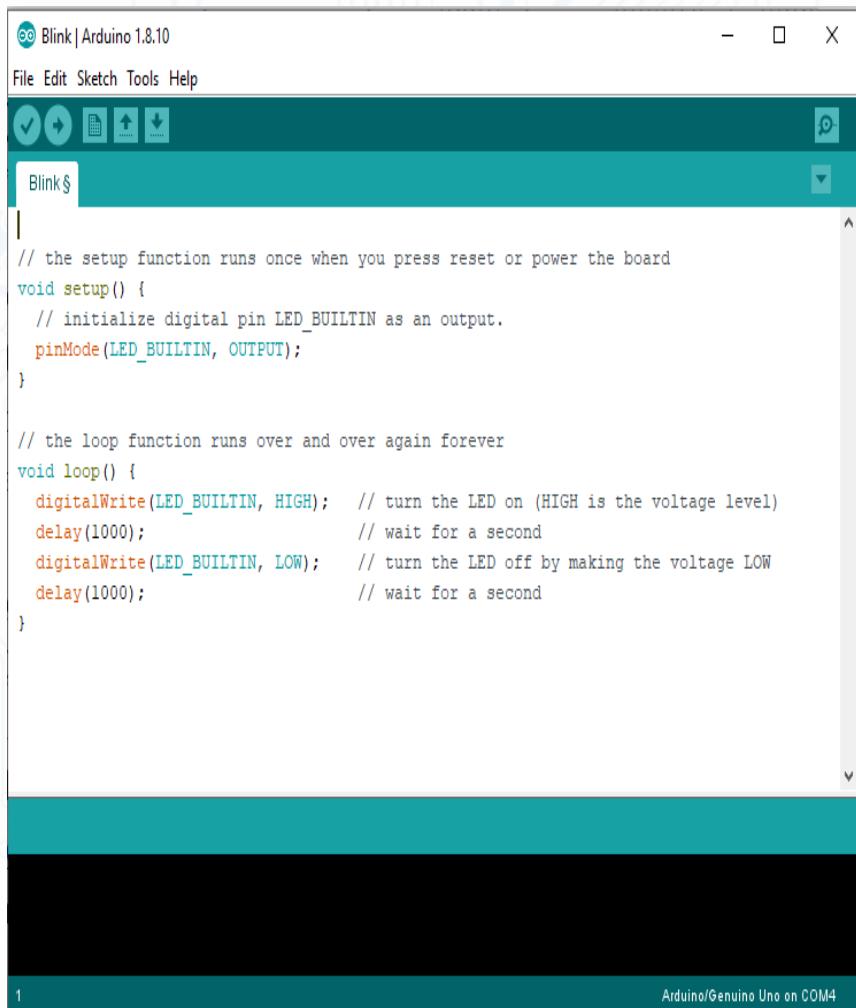
# ARDUINO: Software

Arduino boards can be controlled  
Using an implementation of Wiring,  
Which is a version of Processing  
developed specifically for electronic  
I/O.

Arduino looks like Processing, but  
is actually built in C, so there are a  
few differences to look out for.

Arduino.IDE can be downloaded from  
<http://www.arduino.cc>

Currently used version is 004.

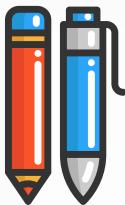


```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

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

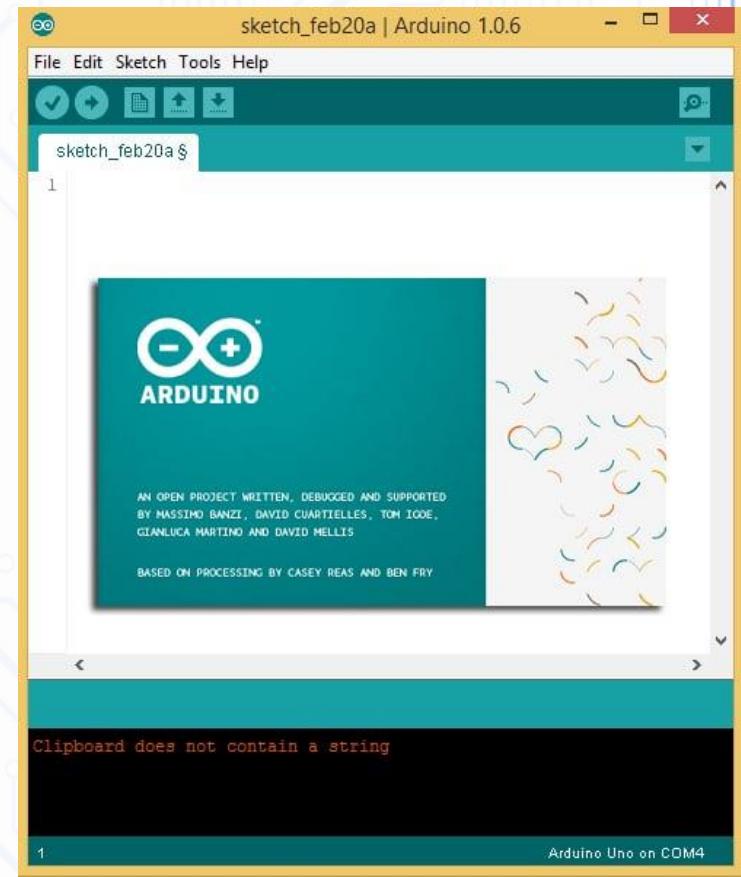
1

Arduino/Genuino Uno on COM4

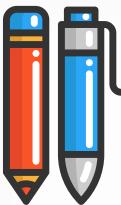


# The Arduino IDE

- The Arduino is programmed in C language.
- The language is very simple and provides many abstraction for simplicity of reading and writing powerfull applications.
- It provides a serial monitor to see the serial data from the USB virtual COM port.
- Allows one click compiling, verification and burning of code onto the Arduino.



Visit [playground.arduino.cc/Main/ DevelopmentTools](http://playground.arduino.cc/Main/DevelopmentTools)  
for alternatives to the base arduino IDE



# Review on IDE Important part

On the Arduino IDE, you will find 6 unique buttons. These are described below.

The screenshot shows the Arduino IDE interface with the following details:

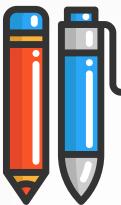
- Title Bar:** Shows "Blink | Arduino 1.8.10".
- File Menu:** Contains "File", "Edit", "Sketch", "Tools", and "Help".
- Toolbar:** Features icons for Verify (checkmark), Upload (blue arrow), New (document), Open (up arrow), Save (down arrow), and Serial Monitor (circular icon).
- Sketch Editor:** Displays the "Blink" sketch code:

```
int LED=9;
int dtime=1000;
void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(dtime);
  digitalWrite(LED, LOW);
  delay(dtime);
}
```
- Status Bar:** Shows "Done Saving." and a message about program storage usage: "Sketch uses 946 bytes (3%) of program storage space. Maximum is 32256 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes."
- Bottom Bar:** Shows "4" and "Arduino/Genuino Uno".

|  |  |
|--|--|
|  | This is used to verify (Arduino refers to compiling as verifying) the sketch.                        |
|  | This is used to upload (Arduino refers to burning a program as uploading) the sketch onto the board. |
|  | This is used to create a new sketch.   |
|  | This is used to open an existing sketch or built-in example.   |
|  | This is used to save the current sketch.   |
|  | This is used to open the serial monitor that comes with the Arduino IDE.                             |

**Note :** You can use any other serial terminal program if you want instead of the Arduino serial monitor. The serial monitor that comes with Arduino IDE can open only one serial port for communication.



# Review on IDE Important part

- Now that we have selected the appropriate board, communication port, and the programmer, we need to upload the Blink sketch on to the board.
- We can either verify the sketch before the upload process or skip the verification process.
- The IDE does the part of verifying the sketch before uploading as a part of the upload process.
- Since this is a built-in example sketch, we will not get any warnings or errors upon verifying. If there are any warnings or errors, they are displayed by the IDE in the black coloured area below the area for writing code. This area is shown in the image in the next point, highlighted in red.
- Now upload the sketch onto the board. When the upload process is done, you will get a done uploading message. You should be able to see the LED on your Arduino board blinking at 1 second intervals.

The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. The code is as follows:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

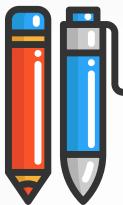
Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino m the Technical Specs of your board at https://www.arduino.cc/en/Main/Products

This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald

modified 2 Sep 2016
by Arturo Guadalupi
```

The status bar at the bottom right indicates "Arduino/Genuino Uno on COM9".

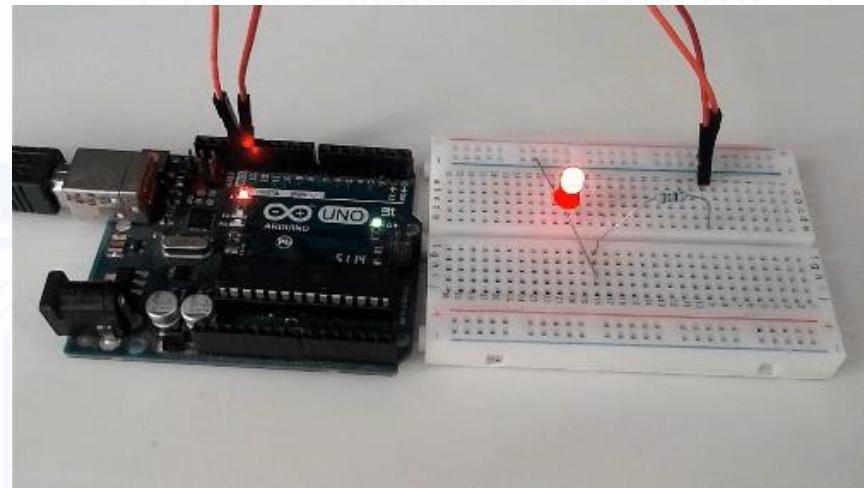


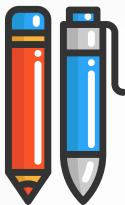
# Review on IDE Important part

```
/*  
 *  
 * Blink  
 *  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
  
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO  
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to  
 * the correct LED pin independent of which board is used.  
 * If you want to know what pin the on-board LED is connected to on your Arduino m  
 * the Technical Specs of your board at https://www.arduino.cc/en/Main/Products  
  
 * This example code is in the public domain.  
  
 * modified 8 May 2014  
 * by Scott Fitzgerald  
  
 * modified 2 Sep 2016  
 * by Arturo Guadalupi  
  
 */  
  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  // turn the LED on (HIGH is 1).  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on  
  delay(1000); // wait for a second  
  
  // turn the LED off (LOW is 0).  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off  
  delay(1000); // wait for a second  
}
```

Done uploading.  
avrduude done. Thank you.

Arduino/Genuine Uno on COM9





# Arduino Programming language v/s Processing

Arduino has two reserved functions:

`void setup()`

`void loop()`

There is no pop-up display window,hence `void draw()` is not special.

`Loop()` can be considered to do the same thing as `draw()` for the arduino.

There are three types of variable in Arduino:

`char`

`int`

`long`

Arduino has a few reserved constants,which donot need to be defined:

`HIGH//5 volts`

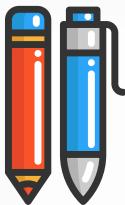
`LOW//0 volts`

`INPUT//pin is input`

`OUTPUT//pin is output`

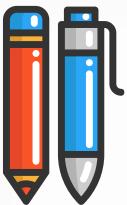
Conditional statements are the same as in Processing.

Functions can be defined the same as in Processing



# Arduino C Specific Functions

- **pinMode(pin, mode)**
  - Designates the specified pin for input or output
- **digitalWrite(pin, value)**
  - Sends a voltage level to the designated pin
- **digitalRead(pin)**
  - Reads the current voltage level from the designated pin
- **analog versions of above**  
**analogRead's range is 0 to 1023**
- **serial commands**  
**print, println, write**



# Arduino Programming language v/s Processing

## Arrays

### Arduino

```
int bar[8];  
bar[0] = 1;  
  
int foo[] = { 0, 1, 2 };
```

### Processing

```
int[] bar = new int[8];  
bar[0] = 1;  
  
int[] foo = { 0, 1, 2 };  
or  
int[] foo = { 0, 1, 2 };
```

## Loops

### Arduino

```
int i;  
for (i = 0; i < 5; i++) { ... }
```

### Processing

```
for (int i = 0; i < 5; i++) { ... }
```

## Printing

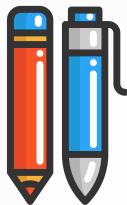
### Arduino

```
Serial.println("hello world");  
  
int i = 5;  
Serial.println(i);  
  
int i = 5;  
Serial.print("i = ");  
Serial.print(i);  
Serial.println();
```

### Processing

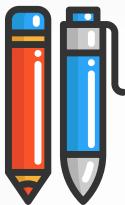
```
println("hello world");  
  
int i = 5;  
println(i);  
  
int i = 5;  
println("i = " + i);
```





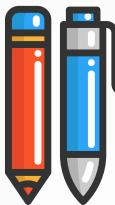
# Steps in Arduino programming

- Open the IDE
- Write code and logic
- Click the verify/compile button to check your program for errors
- Attach the arduino via USB to the PC
- Install drivers if first time
- Setup serial port being used.
- Setup board which we need to program.
- Click upload code to send code to arduino.



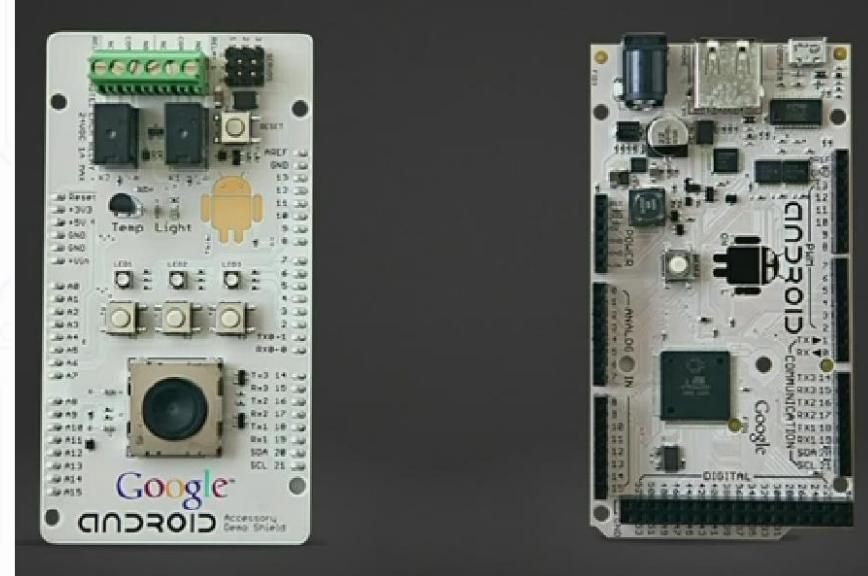
# Projects, just to name a few!!

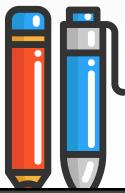
- Interactive real-time auditory feedback system
- GPS receiver module
- Ultrasonic sensor
- Infrared detectors
- SONARS
- Sensor tube for heart monitor
- Pulse rate monitor
- Seven segment LED dice
- Simple room temperature readout
- Lie detector
- Lilypad binary clock



# ANROID+ARDUINO=INFINITY EXISTS!!

- In google annual conference “GOOGLE IO 2011”, android executive announced its most ambitious venture “ANDROID OPEN ACCESSORY”
- They want to bring ANROID out of mobile phone/devices to real world.
- ANDROID sees the true potential of ARDUINO!!!
- They are going to use arduino to interface with the real world objects. Along with other sensors companies like LEADING science.
- They call it ADK-ANDROID DEVELOPMENT KIT





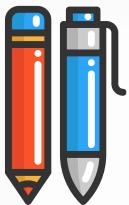
# ANDROID shows its creativity!!



Recent big project being tried  
with ARDUINO is:

- To create a Computer made out of ARDUINO
- Calling it “ARDUINO COM”



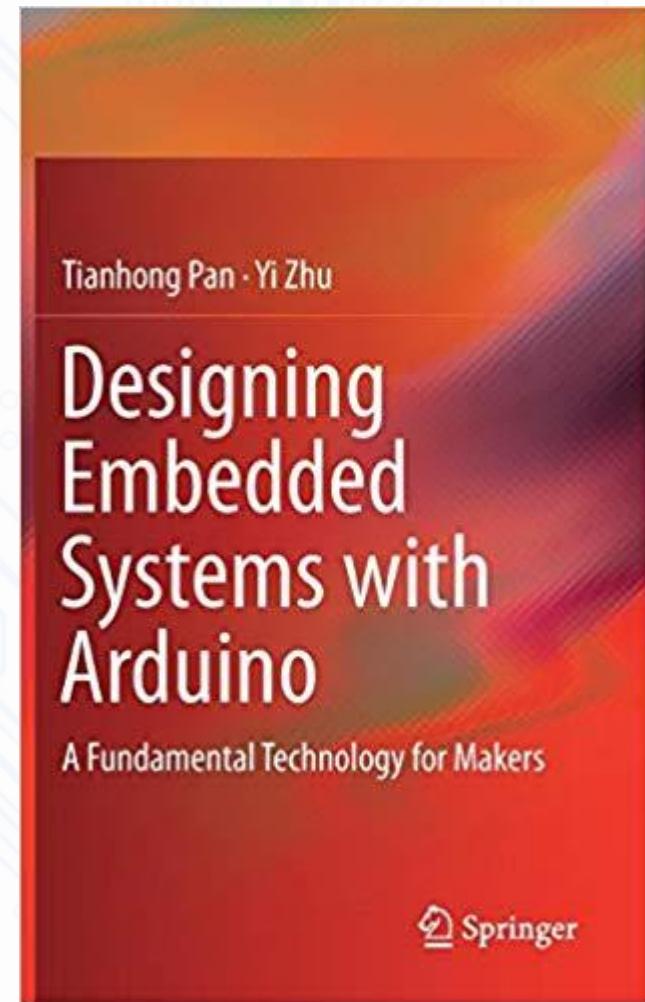


# Reference book

Designing Embedded Systems with Arduino:  
A Fundamental Technology for Makers

1st ed. 2018 Edition

by **Tianhong Pan , Yi Zhu**



# 嵌入式系统

## EMBEDDED SYSTEMS

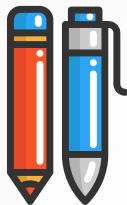


Practical Section:

Programming the Arduino

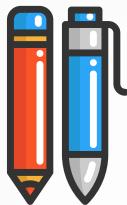
First program: LED Blink

How To Download The IDE  
IDE Installation And Part

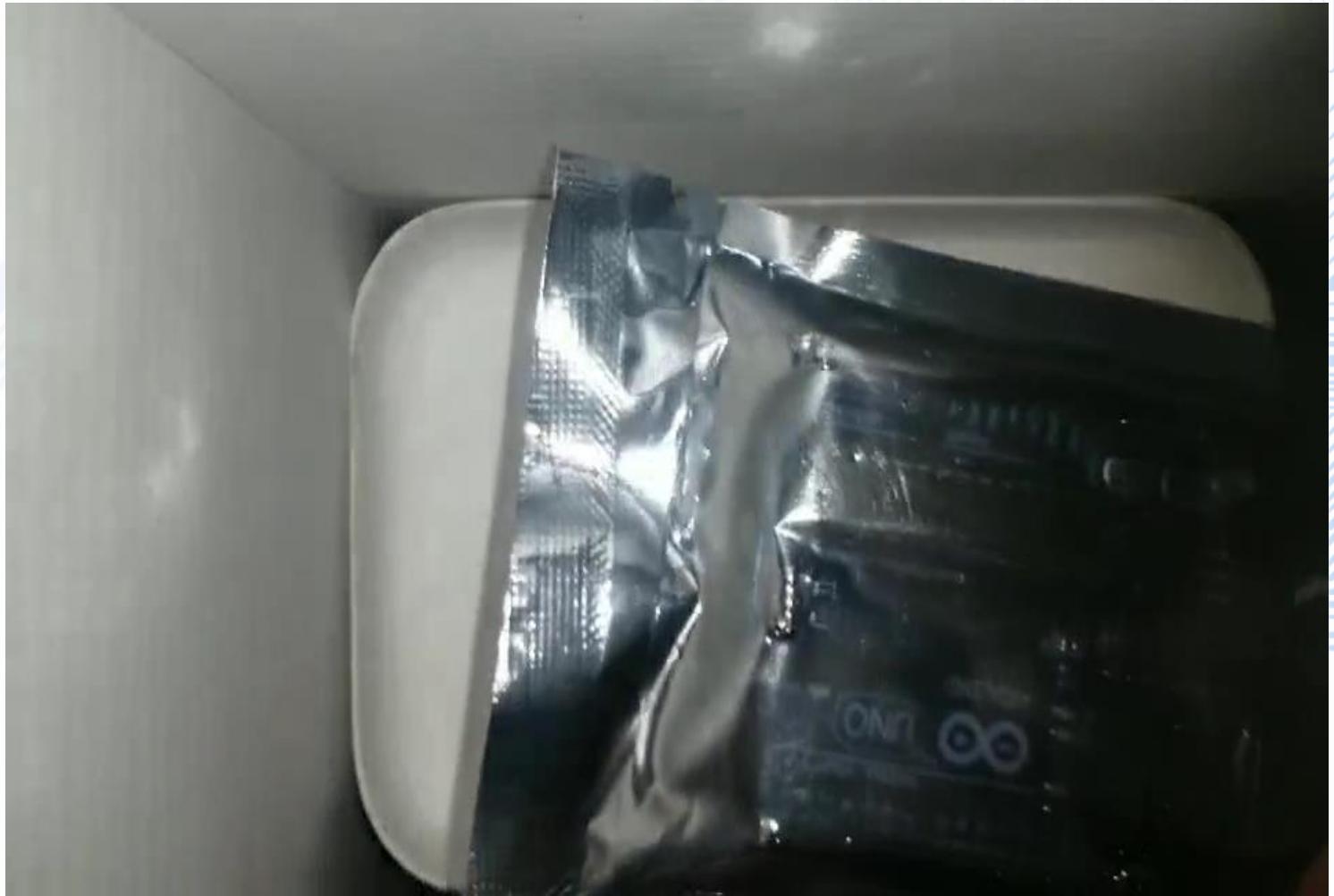


# How To Run Your First Code





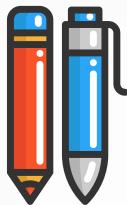
# How To Run Your First Code



DE MONTFORT  
UNIVERSITY  
LEICESTER



江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



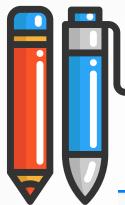
# How To Run Your First Code



DE MONTFORT  
UNIVERSITY  
LEICESTER



江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



# How To Download The IDE

Google - X

https://www.google.com 90%

Getting Started PID Persian Music | Iranian... New Tab Basic functions related...

Gmail Images Sign in

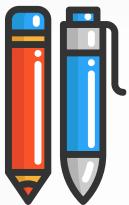
Google Search I'm Feeling Lucky

To all packaging, shipping, and delivery workers, thank you

Google offered in: Indonesia Basa Bali

Indonesia

Advertising Business About How Search works Privacy Terms Settings

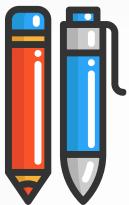


# How To Download The IDE

The screenshot shows a Firefox browser window with the URL <https://www.arduino.cc/> in the address bar. The page content is a list of links:

- Arduino - Software — [arduino.cc/en/Main/Software](https://arduino.cc/en/Main/Software)
- Arduino - Donate — [arduino.cc/en/Main/Donate](https://arduino.cc/en/Main/Donate)
- Arduino Reference — [arduino.cc/reference/en/language/functions/analog-io/analogread/](https://arduino.cc/reference/en/language/functions/analog-io/analogread/)
- www.google.com — [google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=15&cad=rja&uact=8&ved=2ahUKEwj0nbGZxeXoAihCQgAECBwE&usg=AOvVaw0DfJLcOOGzWzHmIjyPQ](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=15&cad=rja&uact=8&ved=2ahUKEwj0nbGZxeXoAihCQgAECBwE&usg=AOvVaw0DfJLcOOGzWzHmIjyPQ)
- Arduino Reference — [arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/](https://arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/)
- www.google.com — [google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwj06duInOPoAhCQgAECBwE&usg=AOvVaw0DfJLcOOGzWzHmIjyPQ](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwj06duInOPoAhCQgAECBwE&usg=AOvVaw0DfJLcOOGzWzHmIjyPQ)
- Arduino Reference — [arduino.cc/reference/en/language/functions/communication/serial/](https://arduino.cc/reference/en/language/functions/communication/serial/)
- www.arduino.cc — [arduino.cc/download\\_handler.php?f=/arduino-1.8.12-windows.zip](https://www.arduino.cc/download_handler.php?f=/arduino-1.8.12-windows.zip)
- Arduino Reference — [arduino.cc/reference/en/language/functions/digital-io/digitalread/](https://arduino.cc/reference/en/language/functions/digital-io/digitalread/)





# IDE Installation And Part

Screenshot of a Windows File Explorer window showing the contents of a folder named "need lib". The folder contains several files and subfolders related to Arduino software installation.

The file list is as follows:

| Name                            | Date modified    | Type               | Size       |
|---------------------------------|------------------|--------------------|------------|
| arduino-1.8.12                  | 19/03/2020 12:50 | File folder        |            |
| arduino-1.8.12-windows          | 15/04/2020 23:29 | WinRAR ZIP archive | 197,325 KB |
| arduino-1.8.12-windows.zip.part | 15/04/2020 20:31 | PART File          | 16,448 KB  |
| SevSeg                          | 11/04/2020 23:54 | WinRAR ZIP archive | 18 KB      |
| TM1637-master                   | 12/04/2020 01:50 | WinRAR ZIP archive | 1,395 KB   |

The left sidebar shows the navigation path: Computer > New Volume (F:) > 00000JIANXI > 000.book > 000.book with ebi > chalesh > need lib. The "Local Disk (C:)" section is expanded, showing "New Volume (F:)" and "New Volume (G:)".

At the bottom, the taskbar shows the following icons: Start button, Search, Task View, File Explorer, File Explorer (highlighted), Microsoft Edge, Microsoft Word, Microsoft Powerpoint, Microsoft Excel, File Explorer, Mozilla Firefox, and a camera icon.



DE MONTFORT  
UNIVERSITY  
LEICESTER



江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

13:03  
18/04/2020



## Practical Section:

---

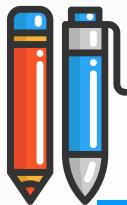
# RUN YOUR FIRST PROGRAM LED BLINK



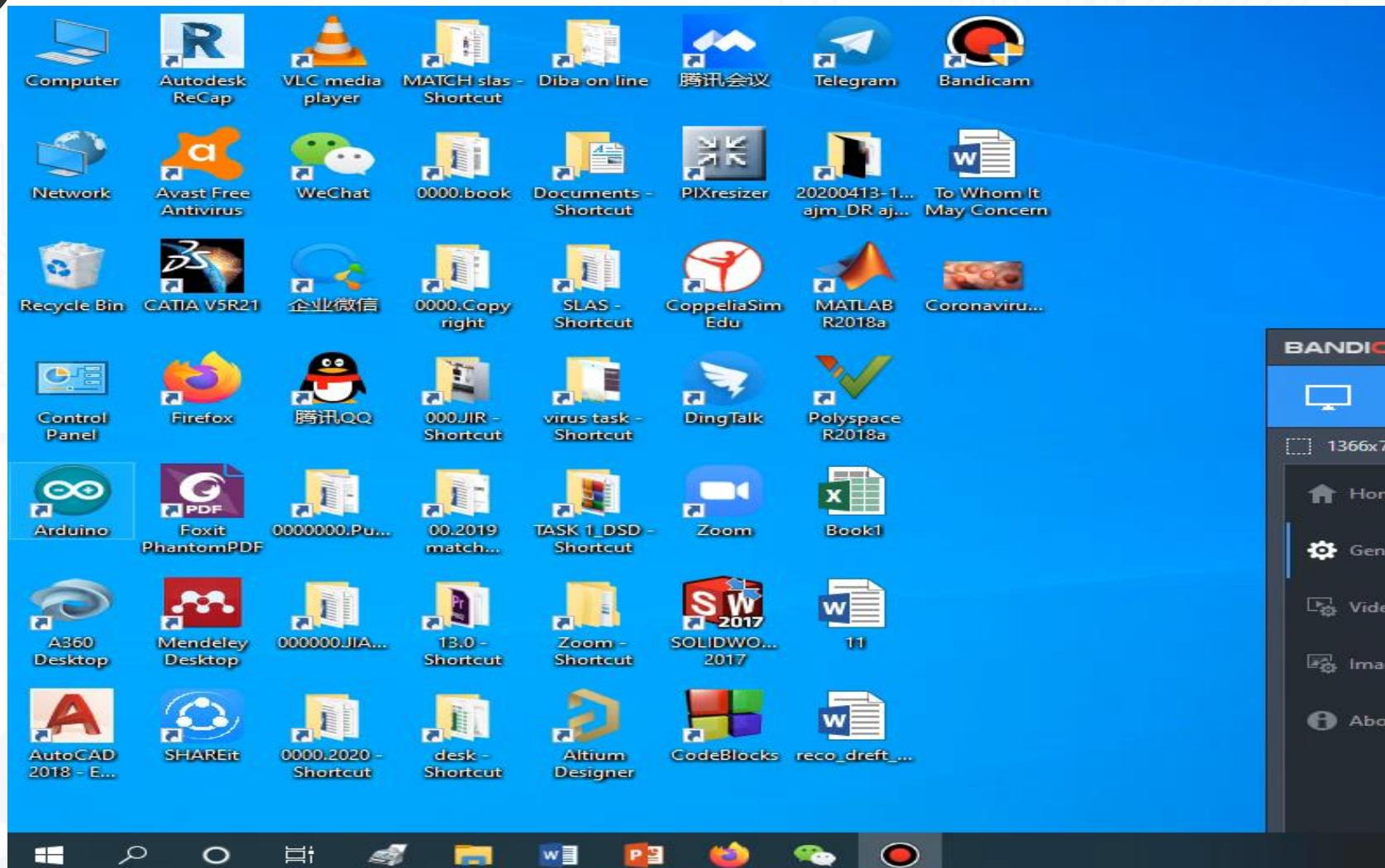
DE MONTFORT  
UNIVERSITY  
LEICESTER

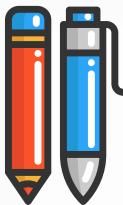


江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

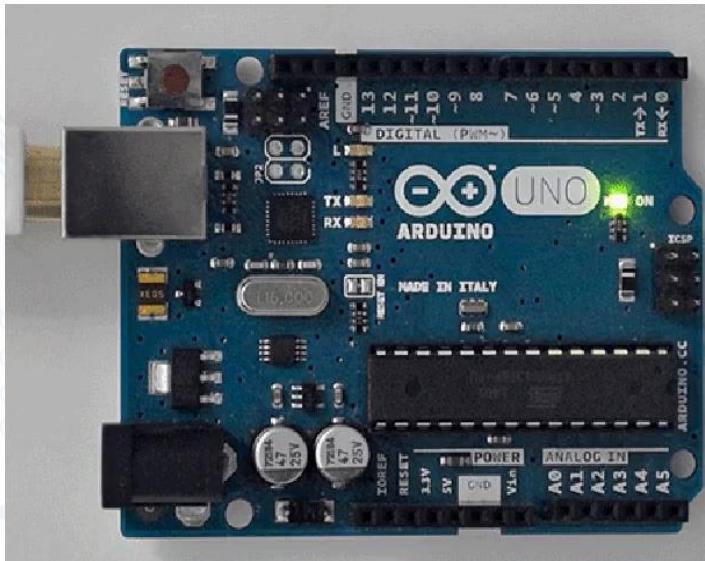


# How To Run Your First Code





# How To Run Your First Code

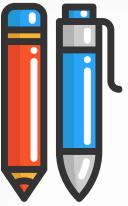


The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_apr18a | Arduino 1.8.9". The code editor contains the following sketch:

```
void setup() {
  // put your setup code here, to run once:
}

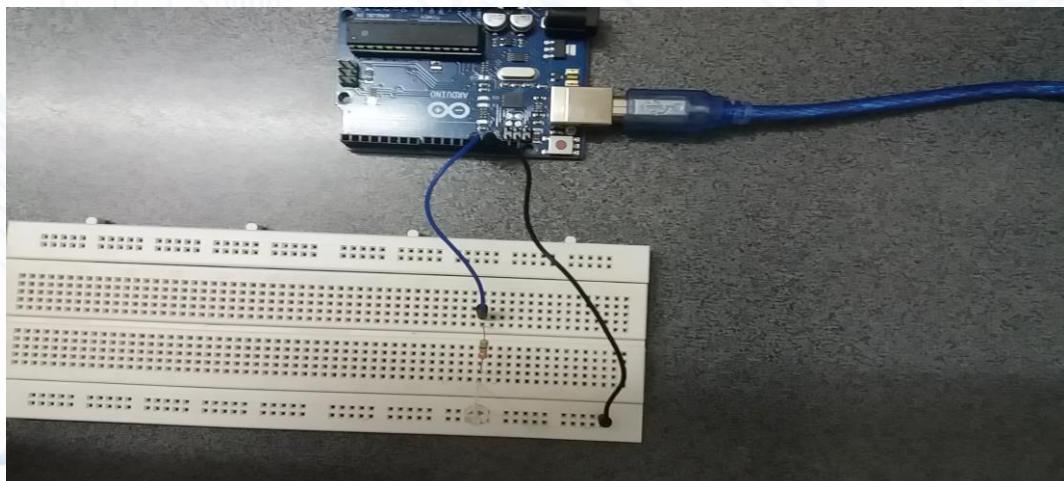
void loop() {
  // put your main code here, to run repeatedly:
}
```

The status bar at the bottom right indicates "Arduino/Genuino Uno on COM6".



# LED Blink

```
// Blink  
// the setup function runs once when you press reset or power the board  
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    // turn the LED on (HIGH is the voltage level)  
    delay(1000);  
    // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);  
    // turn the LED off by making the voltage LOW  
    delay(1000);  
}
```



# Arduino Programming Cheat Sheet

Primary source: Arduino Language Reference  
<http://arduino.cc/en/Reference/>

## Structure & Flow

### Basic Program Structure

```
void setup() {
  // runs once when sketch starts
}

void loop() {
  // runs repeatedly
}
```

**Control Structures**

```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
do { ... } while (x < 5);
for (int i = 0; i < 10; i++) { ... }
break; // exit a loop immediately
continue; // go to next iteration
switch (myVar) {
  case 1:
    ...
    break;
  case 2:
    ...
    break;
  default:
    ...
}
return x; // just return; for voids
```

## Variables, Arrays, and Data

### Data types

```
void
boolean (0, 1, true, false)
char (e.g. 'a' -128 to 127)
int (-32768 to 32767)
long (-2147483648 to 2147483647)
unsigned char (0 to 255)
byte (0 to 255)
unsigned int (0 to 65535)
word (0 to 65535)
unsigned long (0 to 4294967295)
float (-3.4028e+38 to 3.4028e+38)
double (currently same as float)
```

### Qualifiers

```
static (persists between calls)
volatile (in RAM (nice for ISR))
const (make read only)
PROGMEM (in flash)
```

### Arrays

```
int myInts[6]; // array of 6 ints
int myPins[]={2, 4, 8, 3, 6};
int mySensVals[6]={2, 4, -8, 3, 2};
myInts[0]=42; // assigning first
// index of myints
myInts[6]=12; // ERROR! Indexes
// are 0 though 5
char S1[8] =
{'A','r','d','u','i','n','o','\0'};
// unterminated string; may crash
char S2[8] =
{'A','r','d','u','i','n','o','\0'};
// includes \0 null termination
char S3[]="Arduino";
char S4[8]="Arduino";
```

## Operators

### General Operators

```
= (assignment operator)
+ (add) - (subtract)
* (multiply) / (divide)
% (modulo)
== (equal to) != (not equal to)
< (less than) > (greater than)
<= (less than or equal to)
>= (greater than or equal to)
&& (and) || (or) ! (not)
```

### Compound Operators

```
++ (increment)
-- (decrement)
+= (compound addition)
-= (compound subtraction)
*= (compound multiplication)
/= (compound division)
&= (compound bitwise and)
|= (compound bitwise or)
```

### Bitwise Operators

```
& (bitwise and) | (bitwise or)
^ (bitwise xor) ~ (bitwise not)
<< (shift left) >> (shift right)
```

## Variables, Arrays, and Data

### Constants

```
HIGH | LOW
INPUT | OUTPUT
true | false
143 (Decimal)
0173 (Octal - base 8)
0b11011111 (Binary)
0x7B (Hexadecimal - base 16)
7U (force unsigned)
10L (force long)
15UL (force long unsigned)
10.0 (force Floating point)
2.4e5 (2.4*105 = 240000)
```

### Pointer Access

```
& (reference: get a pointer)
* (dereference: follow a pointer)
```

### Strings

```
char S1[8] =
{'A','r','d','u','i','n','o','\0'};
// unterminated string; may crash
char S2[8] =
{'A','r','d','u','i','n','o','\0'};
// includes \0 null termination
char S3[]="Arduino";
char S4[8]="Arduino";
```

## Built-in Functions

### Pin Input/Output

```
Digital I/O (pins: 0-13 A0-A5)
pinMode(pin, [INPUT, OUTPUT])
int digitalRead(pin)
digitalWrite(pin, value)
  // Write HIGH to an input to
  // enable pull-up resistors
Analog In (pins: 0-5)
int analogRead(pin)
analogReference(
  [DEFAULT, INTERNAL, EXTERNAL])
PWM Out (pins: 3 5 6 9 10 11)
analogWrite(pin, value)
```

### Advanced I/O

```
tone(pin, freqhz)
tone(pin, freqhz, duration_ms)
noTone(pin)
shiftOut(dataPin, clockPin,
  [MSBFIRST, LSBFIRST], value)
unsigned long pulseIn(pin,
  [HIGH,LOW])
```

### Time

```
unsigned long millis()
  // overflows at 50 days
unsigned long micros()
  // overflows at 70 minutes
delay(ms)
delayMicroseconds(usec)
```

### Math

```
min(x, y) max(x, y) abs(x)
sin(rad) cos(rad) tan(rad)
sqrt(x) pow(base, exponent)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)
```

### Random Numbers

```
randomSeed(seed) // long or int
long random(max)
long random(min, max)
```

### Bits and Bytes

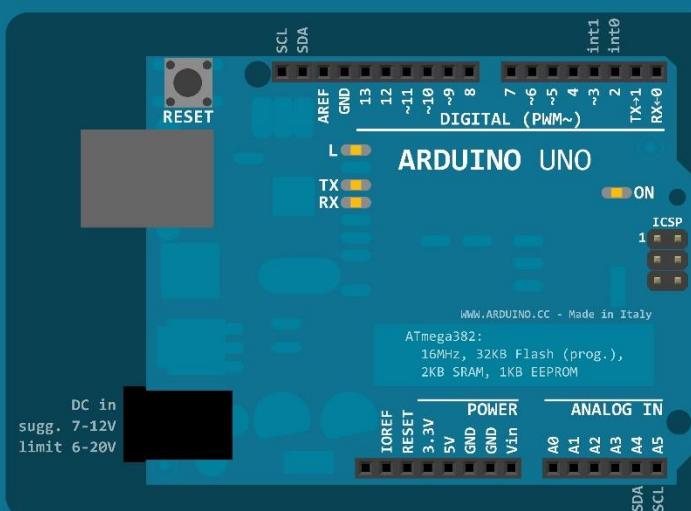
```
lowByte(x) highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn) // bitn: 0=LSB 7=MSB
```

### Type Conversions

```
char() byte()
int() word()
long() float()
```

### External Interrupts

```
attachInterrupt(interrupt, func,
  [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupt)
interrupts()
noInterrupts()
```



## Libraries

```
Serial (communicate with PC or via RX/TX)
begin(long Speed) // up to 115200
end()
int available() // #bytes available
byte read() // -1 if none available
byte peek()
flush()
print(myData)
println(myData)
write(myBytes)
SerialEvent() // called if data ready
```

```
SoftwareSerial (serial comm. on any pins)
(#include <SoftwareSerial.h>)
SoftwareSerial(rxPin, txPin)
begin(long Speed) // up to 115200
listen() // Only 1 can listen
isListening() // at a time.
read(), peek(), print(), println(), write
// all like in Serial library
```

```
EEPROM (#include <EEPROM.h>)
byte read(intAddr)
write(intAddr, myByte)
```

```
Servo (#include <Servo.h>)
attach(pin, [min_us, max_us])
write(angle) // 0 to 180
writeMicroseconds(us)
// 1000-2000; 1500 is midpoint
int read() // 0 to 180
bool attached()
detach()
```

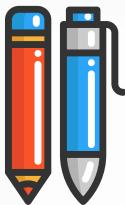
```
Wire (#include <Wire.h>)
begin() // join a master
begin(addr) // join slave @ addr
requestFrom(address, count)
beginTransmission(addr) // Step 1
send(myByte) // Step 2
send(char * myString)
send(byte * data, size)
endTransmission() // Step 3
int available() // #bytes available
byte receive() // get next byte
onReceive(handler)
onRequest(handler)
```



by Mark Liffiton

Adapted from:

- Original by Gavin Smith
- SVG version by Frederic Dufour
- Arduino board drawing original by Fritzing.org



# Student Task\_1:

## 解决mooc分享的问题

Repeat all the example in this PPT with your personal photo

And list the function which use with description

Please send your task based on template



## Send for Next lecture

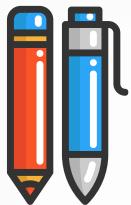


DE MONTFORT  
UNIVERSITY  
LEICESTER

## 发送下一个讲座



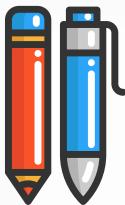
江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



DE MONTFORT  
UNIVERSITY  
LEICESTER



江西理工大学 信息工程学院  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



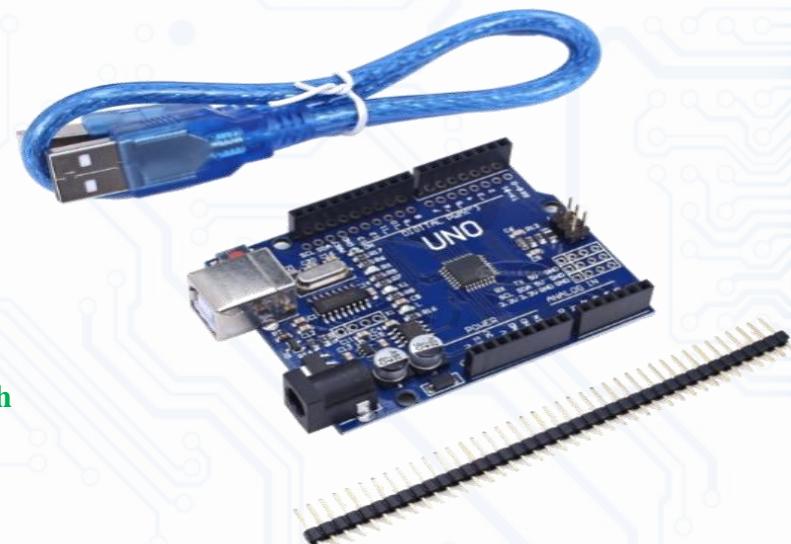
# MY SPECIAL SUGGESTION

- Please try to buy a Arduino board and start to work with this board even may be you may have embedded course but you can learn so many thing with this board.

**Arduino UNO R3 development board  
ATmega328P microcontroller improved version  
development learning control board**

[https://item.taobao.com/item.htm?spm=a230r.1.14.55.461ca7392h\\_KNER&id=533871686531&ns=1&abbucket=20#detail](https://item.taobao.com/item.htm?spm=a230r.1.14.55.461ca7392h_KNER&id=533871686531&ns=1&abbucket=20#detail)

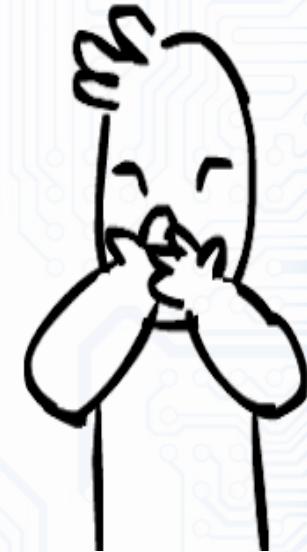
¥ 22.90



*Thank You*  
FOR LISTENING

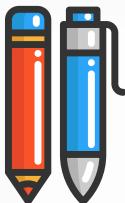


江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



Dr. Ata Jahangir Moshayedi

- School of information engineering Jiangxi university of science and technology, China
- E-mail: [ajm@jxust.edu.cn](mailto:ajm@jxust.edu.cn)



**"BE HUMBLE. BE HUNGRY.  
AND ALWAYS BE THE  
HARDEST WORKER  
IN THE ROOM."**



**DE MONTFORT  
UNIVERSITY  
LEICESTER**



**江西理工大学 信息工程学院**  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING