DOBOT

Dobot 2.0 Technical training tutorials

2019.06.20

Shenzhen Yuejiang Technology Co., Ltd

DOBOT

ONE、 Dobot2.0 Upgrade Highlights

1. Difference of Firmware
2. Update Points of Software

TWO、 How to Achieve the Secondary Development

1. Python Programming Basis
2. How does Python Control Dobot Actions? ----Dobot2.0 Communication Protocol Explanation
3. Secondary Development Tool ----Introduction of the Dobot API Development Library Usage
4. Practical Training ----Python Demo Development Examples Explain
5. Endless Possibilities----Introduction of Dobot 2.0 Related Scalable Applications

THREE 、 Overview of New Function： 3D Printing Basics

1. Forming Principle ： FDM， SLA， DLP
2. Squeezer Structure Understanding
3. FDM Common Software and Key Parameters Understanding
4. Printer Common Problems and Improvement Methods

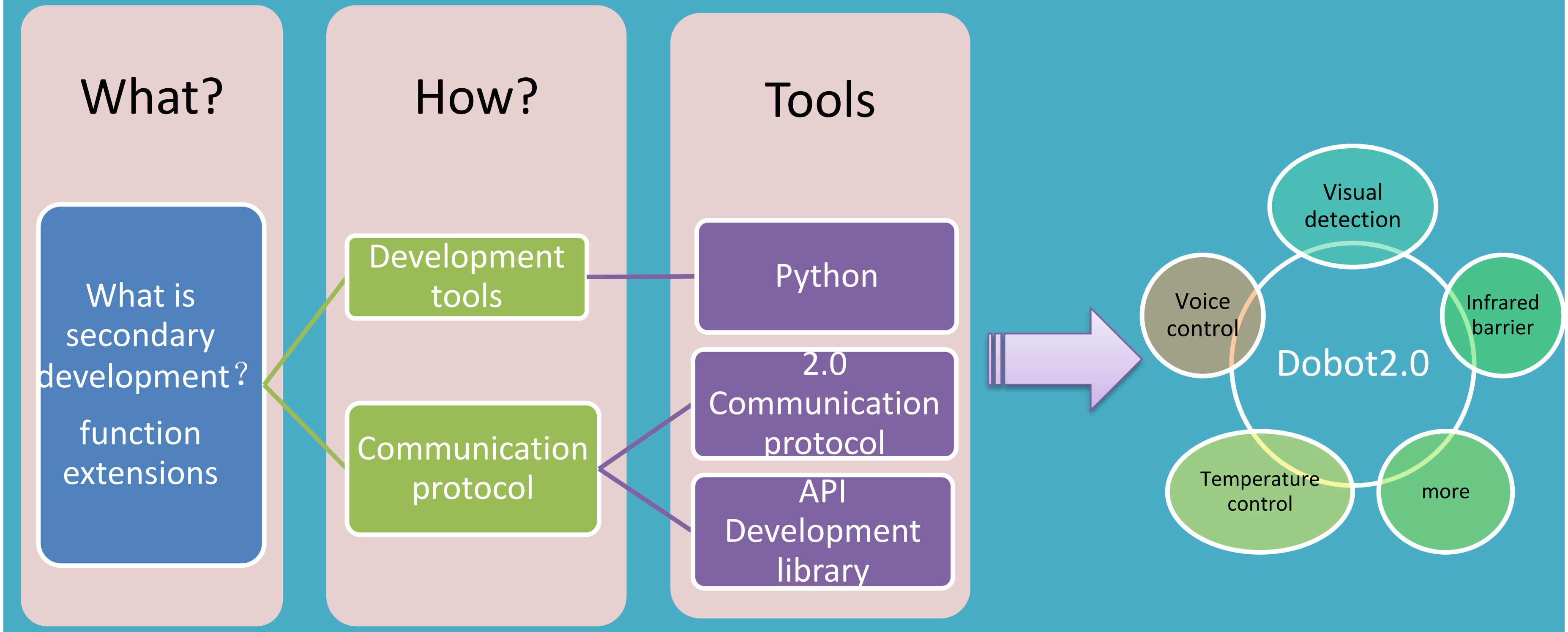
FOUR、 Industrial Robots

1. Industrial Robots Outline
2. Classification of Industrial Robots
3. Industrial Robots Technical Indicators

FIVE 、 Robot Vision

1. Robot Vision Outline
2. Image Capture
3. Image Recognize Principles

Two、 How to Achieve Secondary Development



What?

What is secondary development？

function extensions

How?

Development tools

Communication protocol

Tools

Python

2.0 Communication protocol

API Development library

Dobot2.0

Visual detection

Infrared barrier

more

Temperature control

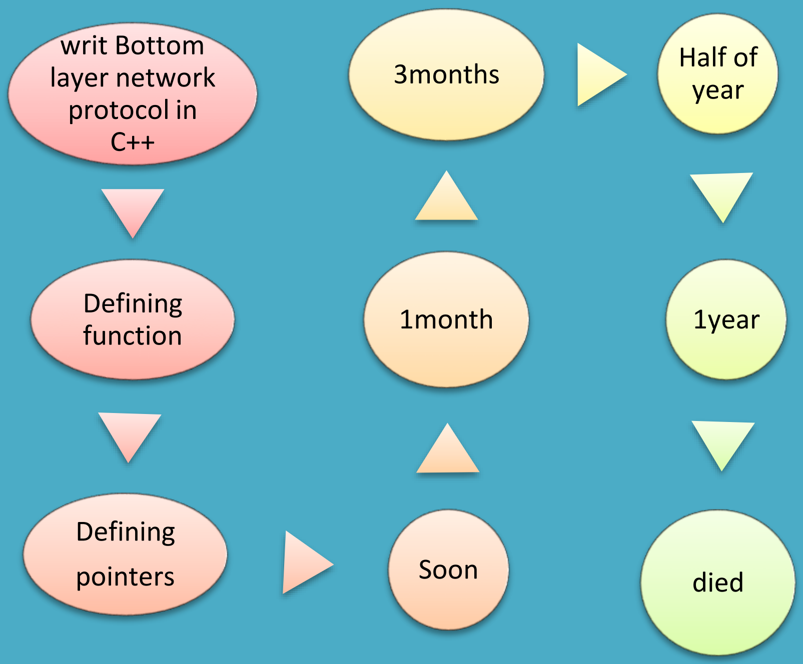
Voice control

Two、 How to Achieve Secondary Development

1. Python Programming Foundation

1.1 What is Python？

Python is an object-oriented, explanatory high-level language that provides us with a very well-developed code base, covering a wide range of networks, files, GUI, databases, text, and so on, which is graphically referred to as "batteries included“. Developed with Python, many features don't have to be written from zero ,you can use the ready-made one directly.



writ Bottom layer network protocol in C++

Defining function

Defining pointers

Soon

1month

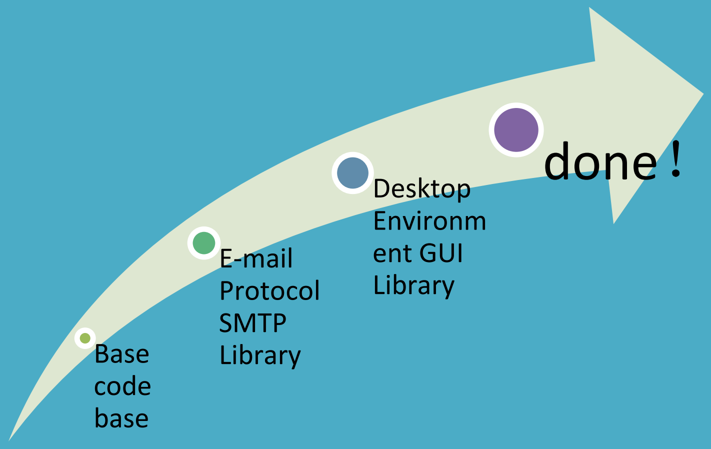
3months

Half of year

1year

died

Programmer A use other language to writ E-Mail client



Base code base

E-mail Protocol SMTP Library

Desktop Environment GUI Library

done！

Programmer B use Python to writ, done in one week！

Simple yet Powerful！

TWO、 How to Achieve Secondary Development

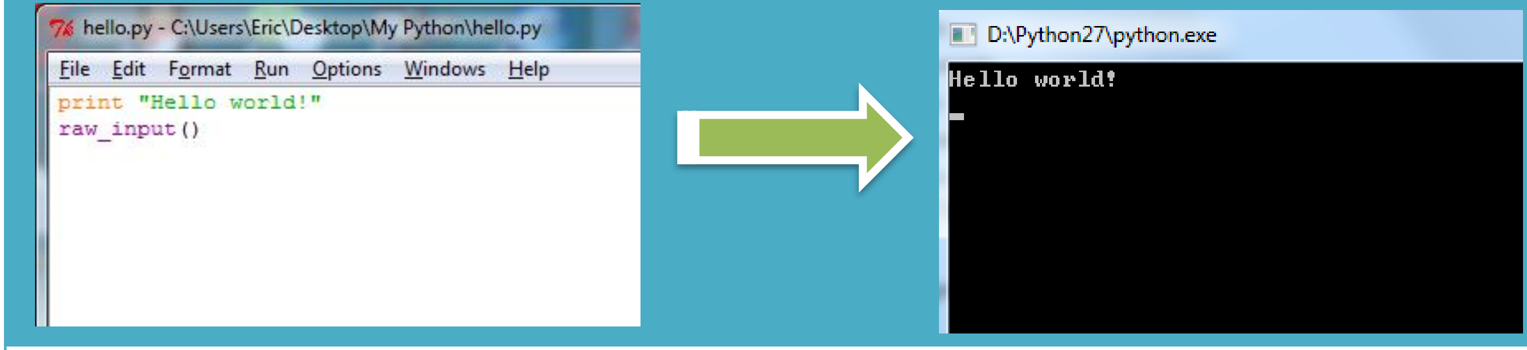
1.2 Python Programming Knowledge

The main content of computer language programming includes: number, word, loop, formula, variable

number	=>	1,2,3,4,5，-3
string	=>	apple，banana，pink
loop	=>	Loop is repeat to do a thing
function/method	=>	Z= x + y , terms is function or method
variable	=>	a, b， it is used while defining formulas

Quotations: here, basically programming has nothing to learn. The rest is to master the writing format of specific functions and loops of various programming languages and master the usage of functions that others have already written .You only need to combine them and use.

1.2.1 Python Trying： Hello world！


hello.py - C:\Users\Eric\Desktop\My Python\hello.py
File Edit Format Run Options Windows Help
D:\Python27\python.exe

```
print "Hello world!"
raw_input()
```

```
Hello world!
_
```

TWO、 How to Achieve Secondary Development

1.2.2 Define/Invoke Function

def a_b_pingfang(a, b):   # def is the abbreviation of define; a, b are parameters; Python uses indented spaces to represent paragraphs, colon (:) indicating the beginning of a paragraph.

   c = a**2 + 2*a*b + b**2   # In computer programming， * represent multiplication， / represent division， ** represent Multiplier，b**2 represent the second party of b

   return c   # c is variable， but in the function name ,it means parameter

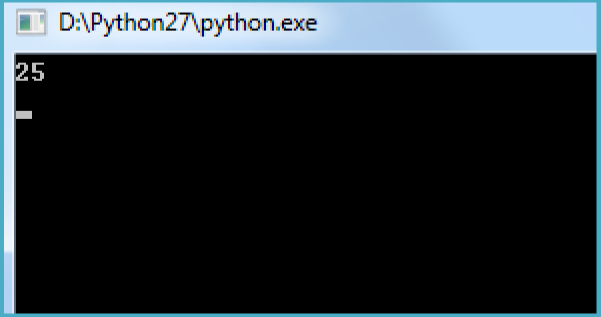
# return is to end the execution of function immediately， through passing the value behind return to the caller , it can pass back multiple values

Now we can invoke function already:

ping2=a_b_pingfang (2, 3)   #passing the value of return c to ping2
print (ping2)   #printing results

This is the whole program:

```
def a_b_pingfang(a, b):
   c = a**2 + 2*a*b + b**2
   return c
ping2 = a_b_pingfang(2, 3)
print(ping2)
```



```
25
_
```

Save to pingfang.py， run python pingfang.py ,it will output 25。

TWO 、 How to Achieve Secondary Development

1.2.3 Loop

How to make the program running repeatedly？Loop。We can use “while” or “for ”command to achieve cycle control.

```
a=0
while a<100:           # Other operators：  <, >, ==, <=, >=, != , and, or, not
    a = a + 1
```

When a < 100, increase a by 1, then repeat. When a is equal to 100, it is over.
We can also use break to interrupt the loop, and continue is return to the loop immediately to start the next loop, or use the if command to judge

```
a=0
while True:              # True  here means correct，  false  means faulty。So here is going to loop
    a=a+1
if a>=100:               # "If"  is judged whether "a“ is greater than or equal to 100, if it is, then execute the following paragraph of “if,” here we interrupted the loop

 break
```

The loop is that simple.

TWO、 How to Achieve Secondary Development

1.2.4 Python data concept： list， tuple， dict

Python data has its special data saving format：

list

list1=[b, c, 1, 3, 5, ‘I’, ‘LOVE‘, ‘YOU’]

- print (list1[0])   # print “b”
- for me in list1:   # visit all list1 content
      print (me)

list1[3] = “HATE” #The contents of the list can be changed
    print (list1[3])

tuple

tuple 2= (2, 3, “second”)

- tuple is list, but its content cannot be changed
- tuple2 = (2, 3, “hello")
  list2= list(tuple2)   # turn to [2, 3, “hello"]
  list2[0] = 0
  tuple2 = tuple(list2)   # turn back (0, 3, “hello")
- Strings are also a special tuple in python.

dict

dict3 = {“name”: “Dobot”, 20:178}

- Explanation of key or index is called "value" in behind. The index cannot be repeated and must be unique.
- The preceding word cannot be a list, because the contents of the list are mutable, and the variable things cannot be indexed.
- print(Dict3[name]) #print Dobot

Two、 How to Achieve Secondary Development

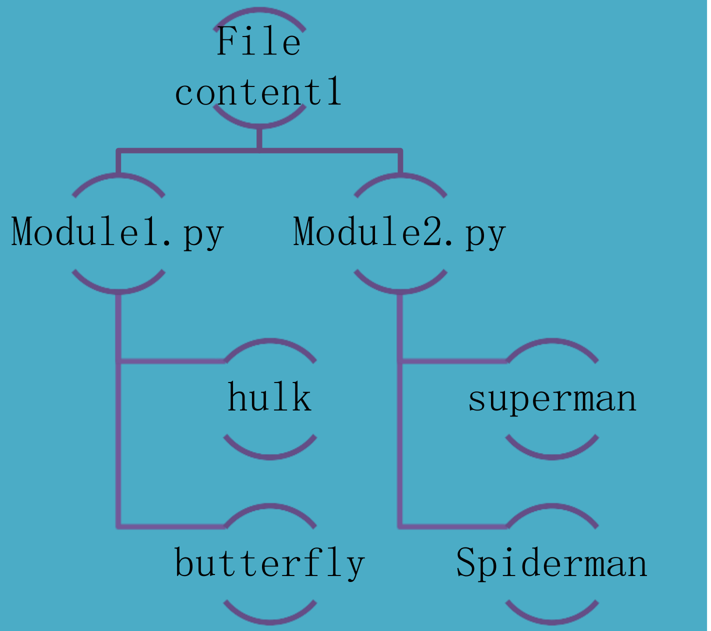
1.2.5 Module

- Python contains a lot of functions written by others, such as functions of mathematical calculations, functions of analyzing email files, functions of encryption, etc. These functions are sorted into things called "modules".
- A module is a python program file
- If the module needs to subdivide more layers, then each layer is a file directory.

To use the module, we need to use "import" command : import

```
import  time
now = time.clock()
print (now)
```

Here, we need to import time module first, and then use the function clock() in the time module to get the current time and print it out.



Two、 How to Achieve Secondary Development

1.2.6 Class

The modules provided by Python, all the functions are packaged in the class.

The idea of Python "object-oriented" is to package the same kind of functions into a thing called "class". This concrete class is called the "object".

```
class RenLei:
   def __init__(self, mingzi):
        self.Mingzi = mingzi
        def shui2(self):
        return self.Mingzi
```

In the class, the first parameter "self" the function is specified , representing the object itself

```
def __init__(wo3, mingzi
wo3.Mingzi = mingzi
```

- ➢ When calling a function in a class, you don't need to provide the self parameter, you just need to provide other parameters.

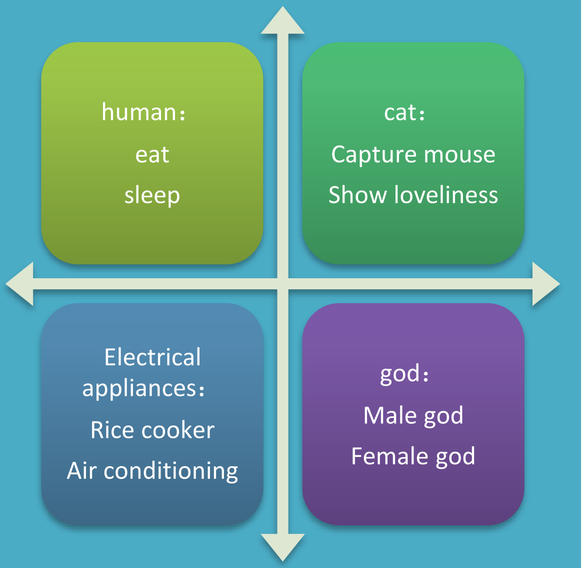
Concretized class， set up object:

```
xiaozhang1 = RenLei(“Eric")
Mingzi = xiaozhang1.shui2()   # invoke the function of class using the format
                                "object    name.function name()“
print Mingzi                   #print “Eric"
```

- ➢ There are two ways to invoke class

1） from cn import * import contents from cn module

2） import cn   tell python that we need to use the content of this module， increase cn at the beginning while using it to indicate that the content is come from cn module.



```
p = Person('marry',21)

p = cn.Person('marry',21)
```

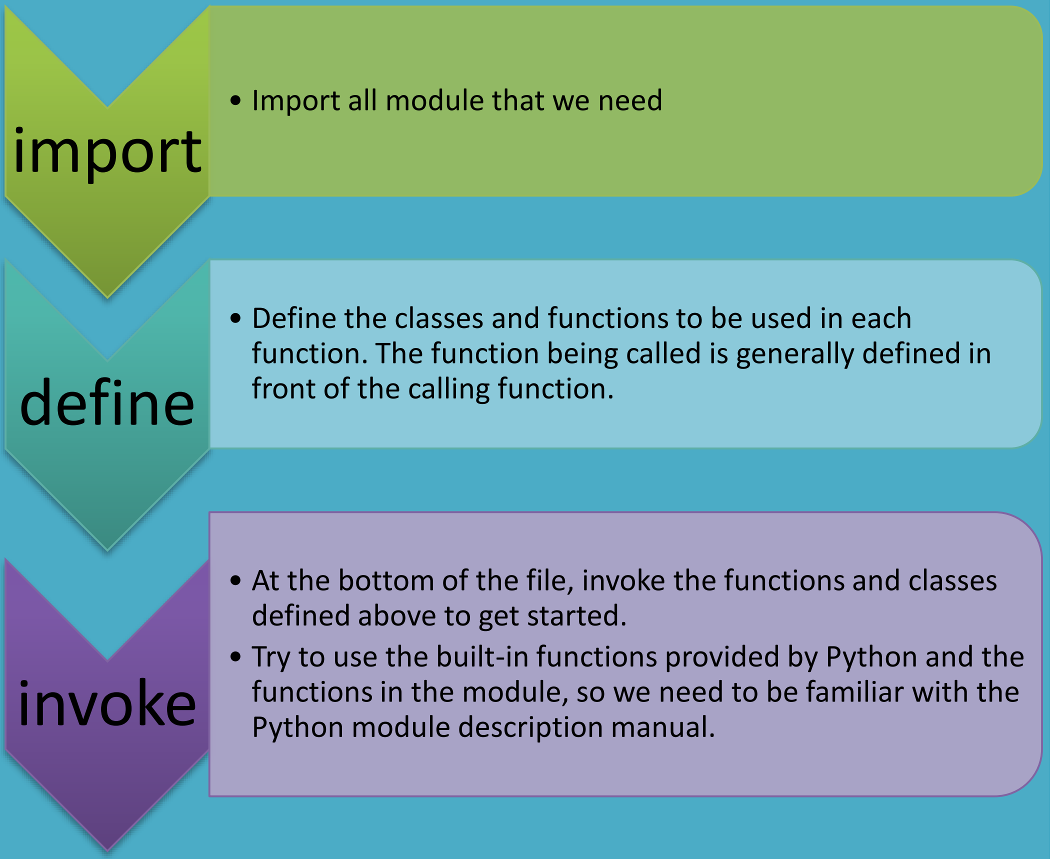
TWO、 How to Achieve Secondary Development

1.3 Python Programming Habit

- Python is run from up to down of the file.
- When we define functions and classes, they are just defined but not executed.
- If a command invoke the above function, execute function be saved is ok.
- The comment symbol in Python is "#". That is to say, the line after the # is an explanation

print (“ming”) # print (“too young too simple")

Only print “ming”,the last word behind “#” has been take as an explanation to be ignored



import

- Import all module that we need

define

- Define the classes and functions to be used in each function. The function being called is generally defined in front of the calling function.

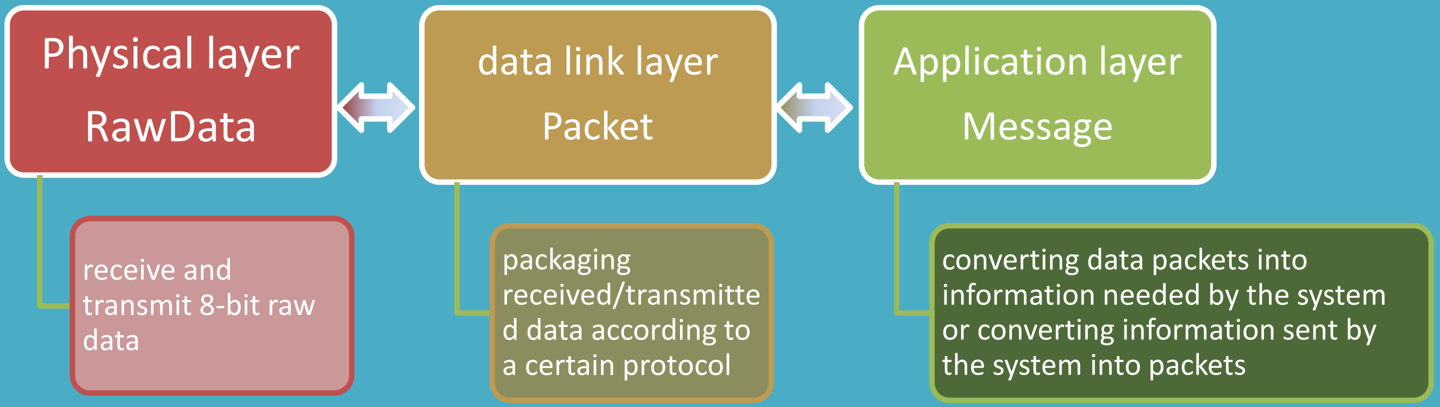
invoke

- At the bottom of the file, invoke the functions and classes defined above to get started.
- Try to use the built-in functions provided by Python and the functions in the module, so we need to be familiar with the Python module description manual.

TWO、 How to Achieve Secondary Development

2. How Python Controls Dobot Motion？

---- Dobot2.0 Communication Protocol Explanation



Physical layer

RawData

receive and transmit 8-bit raw data

data link layer

Packet

packaging received/transmitted data according to a certain protocol

Application layer

Message

converting data packets into information needed by the system or converting information sent by the system into packets

- Dobot V2.0/SCARA robotic arm can be controlled by PC/Android/iOS client, gestures, brain waves, etc.
- The completion of communication can be realized by hardware platforms such as RS232, RS485, CAN, Ethernet and other physical layers.

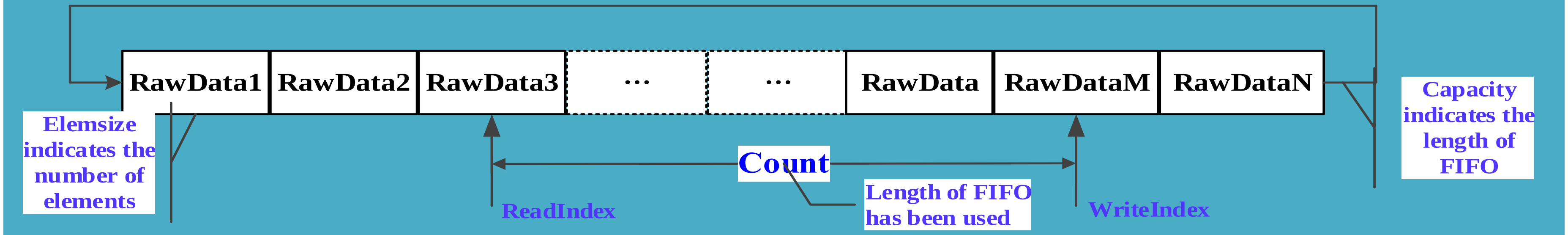
- Determine the start and end of data transmission, and verify the accuracy of the data.
- Communication protocols generally need to include header files, data payloads, load checksums, etc. to ensure accurate data transmission.

- The communication between the robot arm and the client is initiated by the client, and the robot arm does not actively upload data. The robot arm returns different parameters according to the command sent by the client.

TWO、 How to Achieve Secondary Development

2.1 Physical Layer Raw Data

Taking the RS232 serial port as an example, the MCU sends and receives data through the serial port interrupt. At the physical layer, a ring buffer is set to store the received data and the data to be sent, as shown in Figure 2.1.



RawData1 | RawData2 | RawData3 | … | … | RawData | RawDataM | RawDataN

Elemsize indicates the number of elements

Count

Length of FIFO has been used

ReadIndex

WriteIndex

Capacity indicates the length of FIFO

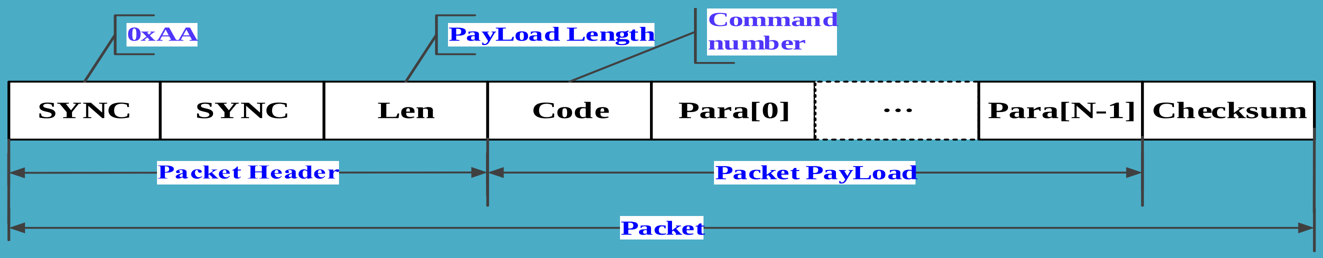
Figure 2.1 Application layer RawData ring buffer

In order to process the buffer of the data, the protocol provides operations such as buffer initialization, reading data, and writing data. In the serial port interrupt, after receiving the data, the queue is written to the corresponding WriteIndex++, Count++; when the serial port interrupt sends data, a data is taken from the buffer queueReadIndex++, Count--. Count indicates the current length that FIFO used. During using, it can judge whether the FIFO is empty, whether it is full, and calculate the free space of the FIFO.

TWO、 How to Achieve Secondary Development

2.2 Data Link Layer Packet

- When receiving data, the data link layer reads data from the RawData queue and performs verification, and sends the data packet containing the verification information to the application layer;
- When sending data, the data link layer parses the packet sent by the application layer and adds the check information to the RawData queue.
- The data processed by the data link layer is a packet containing the packet header file, the payload of the packet, and the payload checksum:



0xAA

PayLoad Length

Command number

SYNC	SYNC	Len	Code	Para[0]	…	Para[N-1]	Checksum

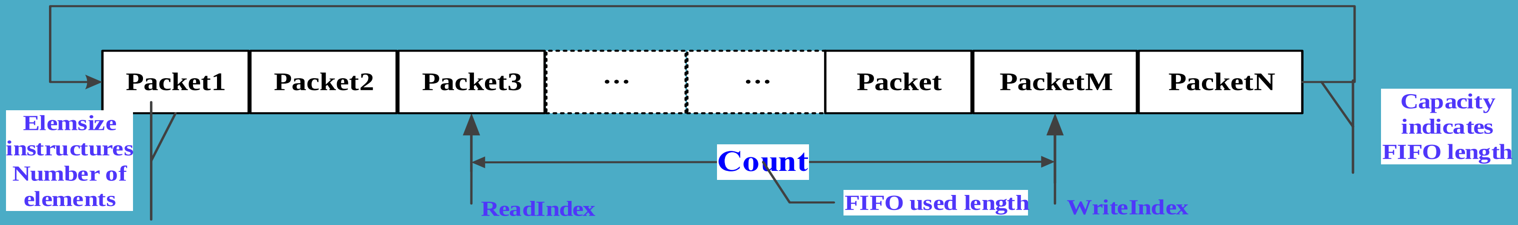
Packet Header

Packet PayLoad

Packet

- SYNC is used to verify the starting data and defines the maximum length of PayLoad. The PayLoad length cannot exceed SYNC. In this protocol, take MAX_PAYLOAD_SIZE =SYNC_BYTE – 1.
- The payload of the packet contains an ID and N data. In communication, data of variable length can be transmitted, but the maximum length should not exceed the limit of the PayLoad maximum.
- The checks sum of playload is the reverses the result of sum of the PayLoad section . After the receiver receives the data, it can perform verification by accumulating PayLoad and Checksum and determining whether the result is 0.

- In order to avoid data loss and data collision in communication, a ring buffer is set at the data link layer to store the received Packet and the Packet to be sent. Its buffer queue is as follows:



When the PacketReadProcess is executed, the data is read from the RawData FIFO, and the data is verified to conform to the data format of the Packet. If the read data is correct, the read data is combined into a Packet and stored in the Packet FIFO. When the PacketWriteProcess function is executed, a data is fetched from the Packet FIFO and written to the RawData FIFO in sequence, and the data check is performed by the receiver.

TWO、 How to Achieve Secondary Development

2.3 Application Layer Message

2.3.1 Application layer data format

The data at the application layer is the interface data of each terminal (MCU, upper computer, IOS, etc.), and its data format is shown in Figure 2.3. Message data can be converted from Packet data: Message.ID=Packet.Code; Message.Len=Packet.Len-1; The data at the application layer is eventually connected through the data link layer and the physical layer data . Message data is consisted of ID mission、Message data length and data.

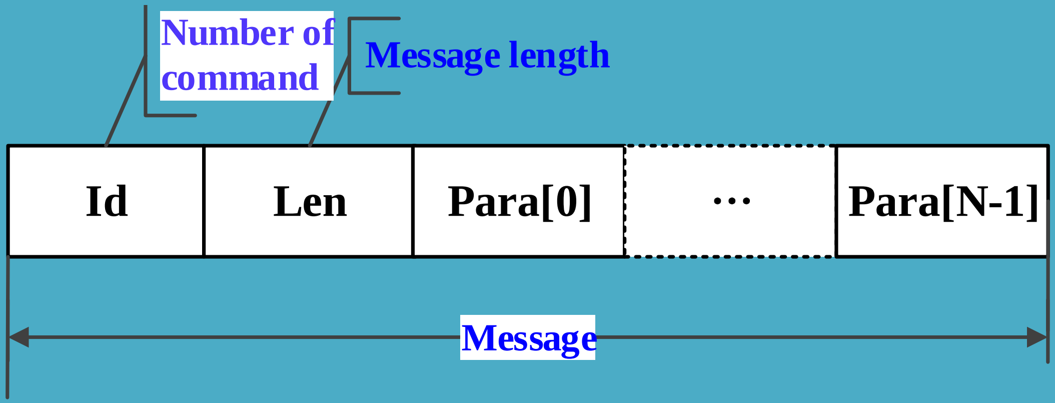
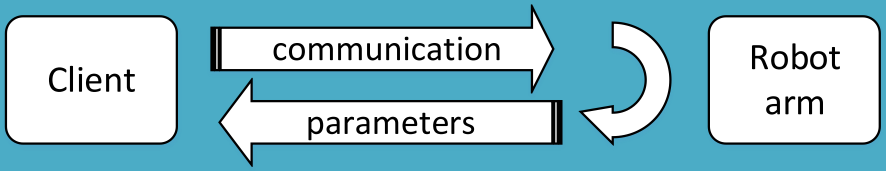


图2.3 Message data format

TWO、 How to Achieve Secondary Development

2.3.2 Application Layer Command



The application layer is divided into three types of functions: common function, teaching function, and reproduction function, as shown in the following table：

ID	function	describe	ID	function	describe
common function			teaching function		
0	SetEndType	Set the end type	9	SetJogJointParams	Set the JogJoint Parameters
1	GetEndType	Get the end type	10	GetJogJointParams	Get the JogJoint Parameters
2	GetPose	Get real-time pose	11	SetJogCoordianteParams	Set the JogCoordiante Parameters
3	ResetPose	Reset real-time pose	12	GetJogCoordinateParams	Get the JogCoordiante Parameters
4	GetPoseL	Get the real-time pose of the slide	13	SetJogCommonParams	Set the JogCommon Parameters
5	GetAlarmsState	Get Alarms State	14	GetJogCommonParams	Get the JogCommon Parameters
6	ClearAllAlarmsState	ClearAllAlarmsState	15	SetJogCmd	Set the Jog command
7	GetIODI	Read I/O imput level			
8	GetIODO	Read I/O output level			

Two、 How to Achieve Secondary Development

2.3.2 Command of Application Layer

ID	function	describe	ID	function	describe
Playback function			**CP function**		
16	SetPTPJointParams	Set the payback speed parameters	26	SetCPParams	Set the CP Parameters
17	GetPTPJointParams	Get the payback speed parameters	27	GetCPParams	Get the CP Parameters
18	SetPTPJumpParams	Set the payback JUMP parameters	28	GetCPCurrentLoopLine	Get the CP current Loop/Line
19	GetPTPJumpParams	Get the payback JUMP parameters	29	SetCPInstantCmd	Set the CP instant command
20	SetPTPCommonParams	Set the Playback Common Parameters	30	GetCPBufferSize	Get the CP Buffer Size
21	GetPTPCommonParams	Get the Playback Common Parameters	31	SetCPBufferCmd	Set the CP buffer commmand
22	SetPTPCoordinateParams	SetPTPCoordinateParameters	15	SetCPCmd	Set the CP function
23	GetPTPCoordinateParams	GetPTPCoordinateParameters	**Get user parameter function**		
			254	GetUserParams	Get the user parameter

TWO、 How to Achieve Secondary Development

2.3.3 Application Layer Command Example Explain
--Set and Get End Parameters (SetEndType & GetEndType)

Set the end parameters:

Function instruction	ID	Len	parameter1	parameter2	parameter3	parameter4
SetEndType	0		endType (uint8_t)	xBias (float)	yBias (float)	zBias (float)
		0X0D	1/2/3	-	-	-

Parameter 1 is endType, which is the type of end (0: custom; 1: sucker; 2: paw; 3: laser); parameter 2 is xBias; parameter 3 is yBias; parameter 4 is zBias.

(1) Set the end type (SetEndType), parameter 1 is the end type:

When the value of parameter 1 is 0, the end is a custom type. At this time, parameters 2 to 4 are the end parameters of X, Y, and Z, respectively.

When the value of parameter 1 is 1/2/3, the end is the preset end (suction cup, claw, laser), and no parameter needs to be sent at this time.

You need to add the length of the command when sending the command. The parameter 1 is that the end type is 1 byte in length, and the parameters 2, 3, and 4 are all 4 bytes, so the Len length is 13=0×0D.

Therefore, the entire instruction sent after processing through the data link layer is as follows:

ID	Len	Param[0]	Param[1]	Param[2]	Param[3]
0	0X0D	endType	xBias	yBias	zBias
		0XFF	0XFF00FF00	0XFF00FF00	0XFF00FF00

Two、 How to achieve secondary development

2.3.3 Application layer command example explanation -- Set and get the end parameters （SetEndType & GetEndType）

（2） Get the end type (GetEndType)，get the instruction only need to send the ID number without parameters，the robot will return the corresponding parameters. The instructions as follows:

ID	Len
1	0X00

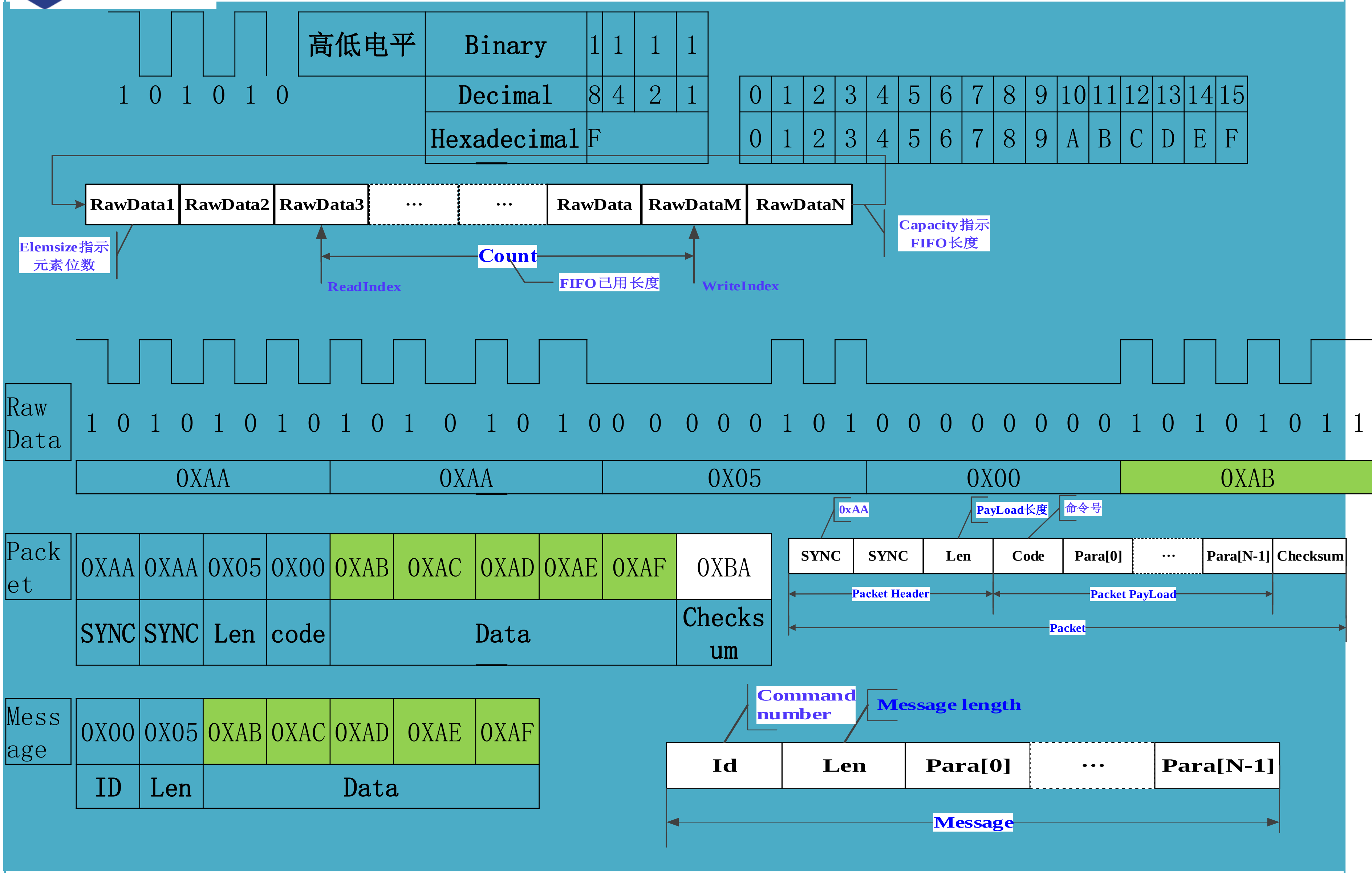
(3) When the above settings and the end parameter command are sent， the arm will return the parameters at the end.

a) The ID returned by the arm is determined according to the issued command: if the previous command is to set the end parameter command， it returns ID = 0； if the previous command is the end parameter command， it returns ID = 1.

b) When the end is a custom type， the arm returns the corresponding ID and end parameters.

c) When the end is the preset type， the arm will only return the corresponding ID. The specific return instructions are as follows:

Type	ID	Len	Param[0]	Param[1]	Param[2]	Param[3]
parameter	0/1		endType (uint8_t)	xBias (float)	yBias (float)	zBias (float)
user-defined		0X0D	0X00	0XFF00FF00	0XFF00FF00	0XFF00FF00
Perset		0X01	0XFF	---	---	---



1 0 1 0 1 0

高低电平	Binary	1	1	1	1
	Decimal	8	4	2	1
	Hexadecimal	F			

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

RawData1	RawData2	RawData3	…	…	RawData	RawDataM	RawDataN

Elemsize指示
元素位数

Count

ReadIndex

FIFO已用长度

WriteIndex

Capacity指示
FIFO长度

Raw Data 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1

0XAA	0XAA	0X05	0X00	0XAB

Packet	0XAA	0XAA	0X05	0X00	0XAB	0XAC	0XAD	0XAE	0XAF	0XBA
	SYNC	SYNC	Len	code	Data					Checksum

0xAA

PayLoad长度

命令号

SYNC	SYNC	Len	Code	Para[0]	…	Para[N-1]	Checksum

Packet Header

Packet PayLoad

Packet

Message	0X00	0X05	0XAB	0XAC	0XAD	0XAE	0XAF
	ID	Len	Data				

Command number

Message length

Id	Len	Para[0]	…	Para[N-1]

Message

TWO、 How to Achieve Secondary Development

3. Secondary Development Tool----Introduction of the Dobot API Development Library Usage

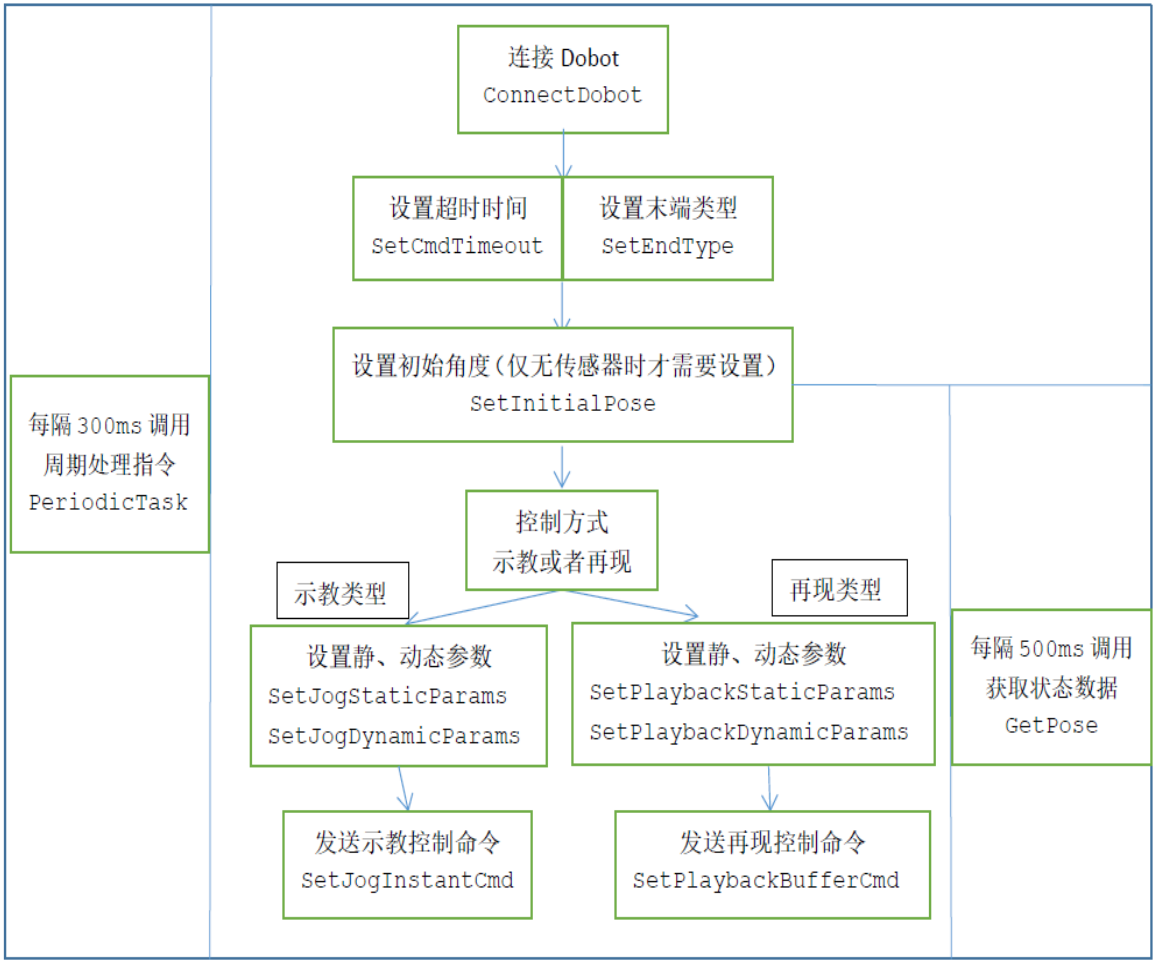
3.1 process of API development invoke

This document is intended for developing applications using the Dobot API in Windows and requires several files:

DobotDll.dll: Dobot API dynamic link library;

DobotDll.h: Dobot API interface declaration;

DobotType.h: all data structures, enumerations, etc. involved in the Dobot API



连接 Dobot
ConnectDobot

设置超时时间
SetCmdTimeout

设置末端类型
SetEndType

设置初始角度（仅无传感器时才需要设置）
SetInitialPose

每隔 300ms 调用
周期处理指令
PeriodicTask

控制方式
示教或者再现

示教类型

再现类型

设置静、动态参数
SetJogStaticParams
SetJogDynamicParams

设置静、动态参数
SetPlaybackStaticParams
SetPlaybackDynamicParams

每隔 500ms 调用
获取状态数据
GetPose

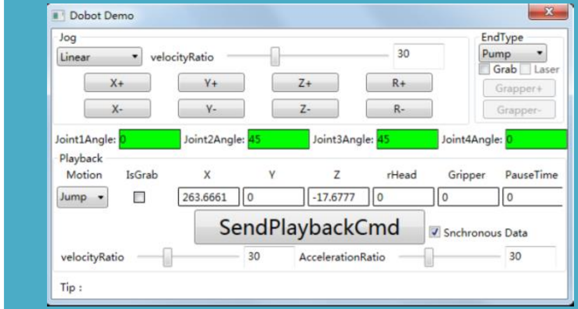
发送示教控制命令
SetJogInstantCmd

发送再现控制命令
SetPlaybackBufferCmd

TWO、 How to achieve secondary development

3.2 Demo Classification

WPF development example



Dobot Demo

Jog

Linear ▾ velocityRatio 30

EndType

Pump ▾

□ Grab □ Laser

Joint1Angle: 0 Joint2Angle: 45 Joint3Angle: 45 Joint4Angle: 0

Playback

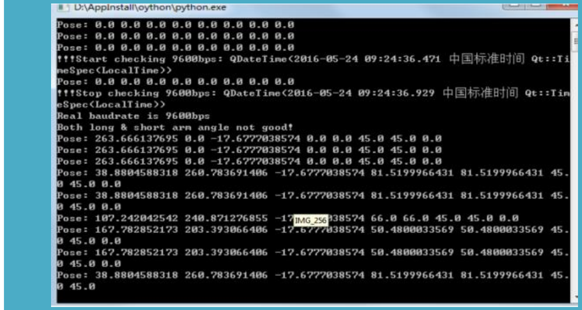
Motion	IsGrab	X	Y	Z	rHead	Gripper	PauseTime
Jump ▾	□	263.6661	0	-17.6777	0	0	0

☑ Snchronous Data

velocityRatio 30 AccelerationRatio 30

Tip :

Python development example

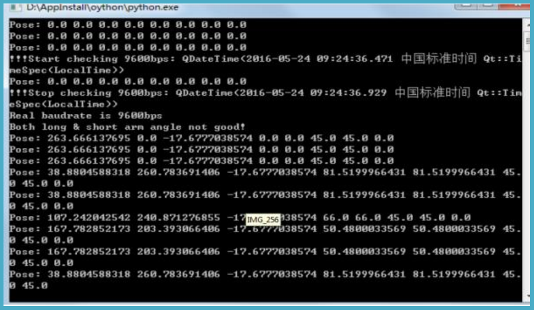


D:\AppInstall\oython\python.exe

```
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
!!!Start checking 9600bps: QDateTime(2016-05-24 09:24:36.471 中国标准时间 Qt::Ti
meSpec(LocalTime))
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
!!!Stop checking 9600bps: QDateTime(2016-05-24 09:24:36.929 中国标准时间 Qt::Tim
eSpec(LocalTime))
Real baudrate is 9600bps
Both long & short arm angle not good!
Pose: 263.666137695 0.0 -17.6777038574 0.0 0.0 45.0 45.0 0.0
Pose: 263.666137695 0.0 -17.6777038574 0.0 0.0 45.0 45.0 0.0
Pose: 263.666137695 0.0 -17.6777038574 0.0 0.0 45.0 45.0 0.0
Pose: 38.8804588318 260.783691406 -17.6777038574 81.5199966431 81.5199966431 45.
0 45.0 0.0
Pose: 38.8804588318 260.783691406 -17.6777038574 81.5199966431 81.5199966431 45.
0 45.0 0.0
Pose: 107.242042542 240.871276855 -17IMG_256038574 66.0 66.0 45.0 45.0 0.0
Pose: 167.782852173 203.393066406 -17.6777038574 50.4800033569 50.4800033569 45.
0 45.0 0.0
Pose: 167.782852173 203.393066406 -17.6777038574 50.4800033569 50.4800033569 45.
0 45.0 0.0
Pose: 38.8804588318 260.783691406 -17.6777038574 81.5199966431 81.5199966431 45.
0 45.0
```

Arduino development example

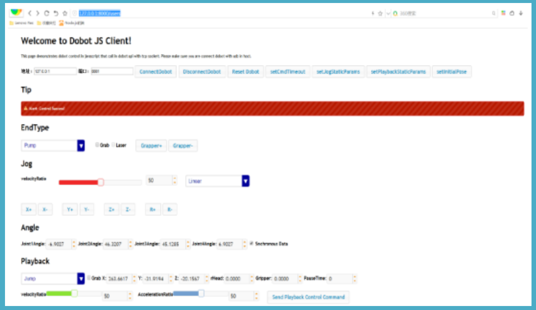
Java development example



D:\AppInstall\oython\python.exe

```
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
!!!Start checking 9600bps: QDateTime(2016-05-24 09:24:36.471 中国标准时间 Qt::Ti
meSpec(LocalTime))
Pose: 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
!!!Stop checking 9600bps: QDateTime(2016-05-24 09:24:36.929 中国标准时间 Qt::Tim
eSpec(LocalTime))
Real baudrate is 9600bps
Both long & short arm angle not good!
Pose: 263.666137695 0.0 -17.6777038574 0.0 0.0 45.0 45.0 0.0
Pose: 263.666137695 0.0 -17.6777038574 0.0 0.0 45.0 45.0 0.0
Pose: 263.666137695 0.0 -17.6777038574 0.0 0.0 45.0 45.0 0.0
Pose: 38.8804588318 260.783691406 -17.6777038574 81.5199966431 81.5199966431 45.
0 45.0 0.0
Pose: 38.8804588318 260.783691406 -17.6777038574 81.5199966431 81.5199966431 45.
0 45.0 0.0
Pose: 107.242042542 240.871276855 -17IMG_256038574 66.0 66.0 45.0 45.0 0.0
Pose: 167.782852173 203.393066406 -17.6777038574 50.4800033569 50.4800033569 45.
0 45.0 0.0
Pose: 167.782852173 203.393066406 -17.6777038574 50.4800033569 50.4800033569 45.
0 45.0 0.0
Pose: 38.8804588318 260.783691406 -17.6777038574 81.5199966431 81.5199966431 45.
0 45.0
```

JS development example



Welcome to Dobot JS Client!

This page demonstrates dobot control in javascript that call in dobot api with tcp socket. Please make sure you are connect dobot with usb in host.

地址: 127.0.0.1 端口: 8081

Tip

Alert: Control Succeed

EndType

Pump ▾ □ Grab □ Laser

Jog

velocityRatio 50 Linear ▾

Angle

Joint1Angle: -6.9027 Joint2Angle: 46.3207 Joint3Angle: 45.1285 Joint4Angle: 6.9027 ☑ Snchronous Data

Playback

Jump ▾ □ Grab X: 263.6617 Y: -31.9194 Z: -20.1567 rHead: 0.0000 Gripper: 0.0000 PauseTime: 0

velocityRatio 50 AccelerationRatio 50

TWO、 How to achieve secondary development

4. Practical Training ---- Explanation of Python Demo development example

```
import sys
from ctypes import *
import threading
import time

def PeriodicTask():
    dll.PeriodicTask()
    threading.Timer(0.1,
PeriodicTask).start()

def GetPoseTask():
    pose = Pose()
    dll.GetPose(byref(pose))
    print 'Pose:', pose.x, pose.y,
pose.z, pose.rHead,
pose.joint1Angle,
pose.joint2Angle,
pose.joint3Angle,
pose.joint4Angle
    threading.Timer(0.5,
GetPoseTask).start()
# For initial pose
class InitialPose(Structure):
    _fields_ = [("joint2Angle",
c_float), ("joint3Angle",
c_float)]
# For pose
class Pose(Structure):
    _fields_ = [
        ("x", c_float),
        ("y", c_float),
```

```
        ("z", c_float),
        ("rHead", c_float),
        ("joint1Angle", c_float),
        ("joint2Angle", c_float),
        ("joint3Angle", c_float),
        ("joint4Angle", c_float),
        ("isGrab", c_byte),
        ("gripper", c_float)
        ]
# For jog
class
JogStaticParams(Structure):
    _fields_ = [
        ("jointMaxVelocity",
c_float),
("jointMaxAcceleration",
c_float),
        ("servoMaxVelocity",
c_float),
("servoMaxAcceleration",
c_float),
        ("linearMaxVelocity",
c_float),
("linearMaxAcceleration",
c_float)
        ]
class
JogDynamicParams(Structure):
    _fields_ = [("velocityRatio",
c_float)]
class JogInstantCmd(Structure):
```

```
    _fields_ = [("isJoint", c_byte),
("cmd", c_int)]

sys.path.append(sys.path[0])
dll = cdll.LoadLibrary(sys.path[0]
+ '//DobotDll.dll');
PeriodicTask()
GetPoseTask()
errorString = [
    'Success',
    'Warning:Long arm angle not
good!',
    'Warning:Short arm angle
not good!',
    'Both long & short arm angle
not good!',
    'Error:Dobot not found!',
    "Error:COM port occupied!",
    "Error:No data uploaded!"
    ]
result = dll.ConnectDobot()
print errorString[result]

if (result < 4):
    # Set command timeout
    dll.SetCmdTimeout(3000)
    # Set initial pose
    initialPose = InitialPose()
    initialPose.joint2Angle = 45
    initialPose.joint3Angle = 45
    dll.SetInitialPose.argtypes =
```

```
[POINTER(InitialPose)]

dll.SetInitialPose(byref(initialPo
se))

    # Set static & dynamic param
    jogStaticParams =
JogStaticParams()
jogStaticParams.jointMaxVeloci
ty = 200;
jogStaticParams.jointMaxAccel
eration = 200;
jogStaticParams.servoMaxVelo
city = 200;
jogStaticParams.servoMaxAccel
eration = 200;
jogStaticParams.linearMaxVelo
city = 200;
jogStaticParams.linearMaxAcce
leration = 200;
    jogDynamicParams =
JogDynamicParams();
jogDynamicParams.velocityRati
o = 50;
dll.SetJogStaticParams(byref(jo
gStaticParams))
dll.SetJogDynamicParams(byref
(jogDynamicParams))
  dll.SetJogInstantCmd.argtypes
= [POINTER(JogInstantCmd)]
    test = JogInstantCmd()
```

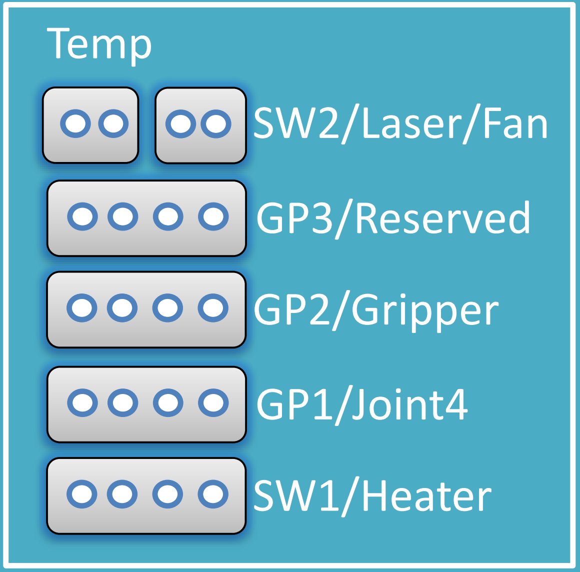
```
  test.isJoint = 1
  while 1:
    test.cmd = 1;

dll.SetJogInstantCmd(byref(test
))
    time.sleep(0.2)
    test.cmd = 0;

dll.SetJogInstantCmd(byref(test
))
    time.sleep(1)
    test.cmd = 2;
dll.SetJogInstantCmd(byref(test
))
    time.sleep(0.2)
    test.cmd = 0;
dll.SetJogInstantCmd(byref(test
))
    time.sleep(1)
dll.DisconnectDobot()
```

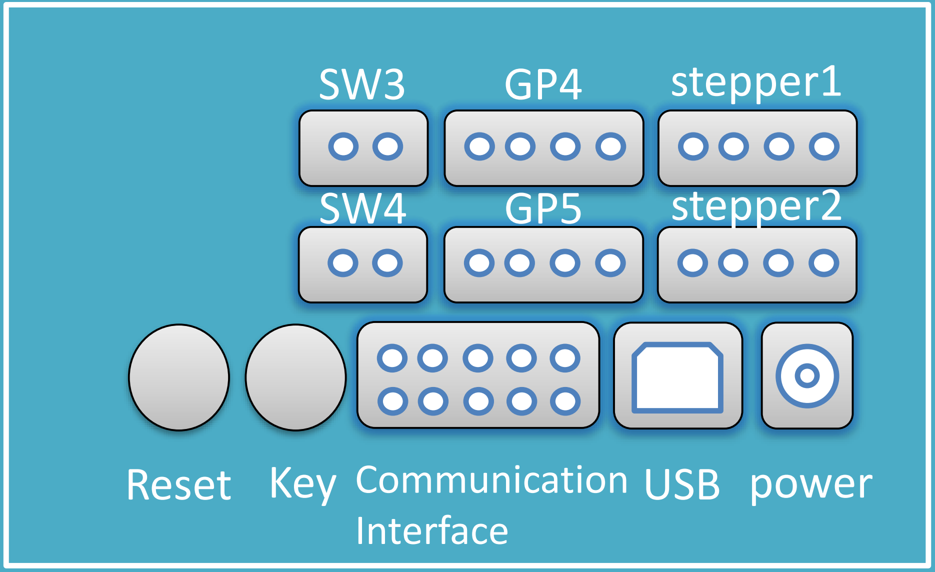
DOBOT

TWO、 How to achieve secondary development

5. Endless possibilities---- Introduction of Dobot2.0 related scalable applications

The reserved I/O of the robot arm is as follows：





- ➢ SW stands for a 12V power supply that can be cut off
- ➢ GP is a General port
- ➢ Communication interface

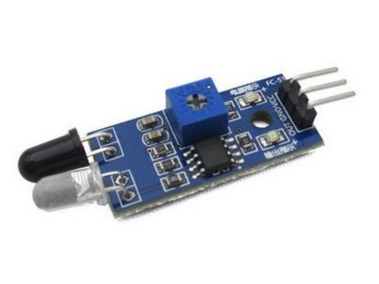
Communication interface

3.3V	RXD	RSTn	IO1	12V
GND	TXD	INTn	IO2	GND

General port

GND	3.3V	IO1/PWM	IO2/AD

TWO、 How to Achieve Secondary Development

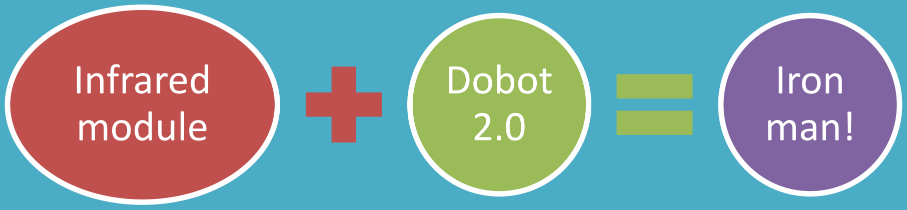


5. 1 Infrared Barrier Avoidance

What if Dobot hits things？

solutions： use one or more LM393 comparator r+ infrared emission / receiving tube module

Module pins：			
GND	VCC	OUT	
General port			
GND	3.3V	IO1/PWM	IO2/AD

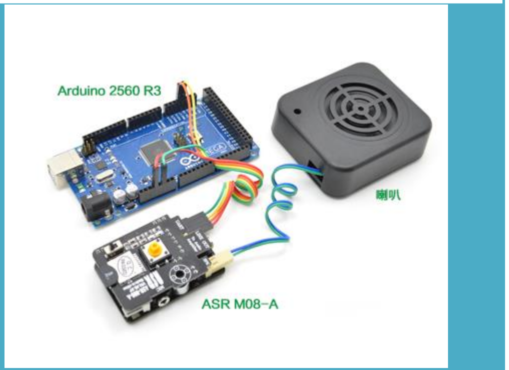


Infrared module + Dobot 2.0 = Iron man!

Introduction of module：

- When the module detected the obstacle signal in front, the green indicator on the board lights up, meanwhile, the OUT port will continuously output a low level signal.The detection distance of this module is 2~30cm and the detection angle is 35° . The detection distance can be adjust by potentiometer.
- The sensor actively detects infrared reflections, so the reflectivity and shape of the target are the key to detecting the distance. Among them, the black detection distance is the smallest, and the white color is the largest; the small-area object has a small distance and the large-area distance is large;The sensor module output port can be directly connected to the MCU IO port, and drive a 5V relay directly ;
- VCC input voltage range: 3.3V-5V

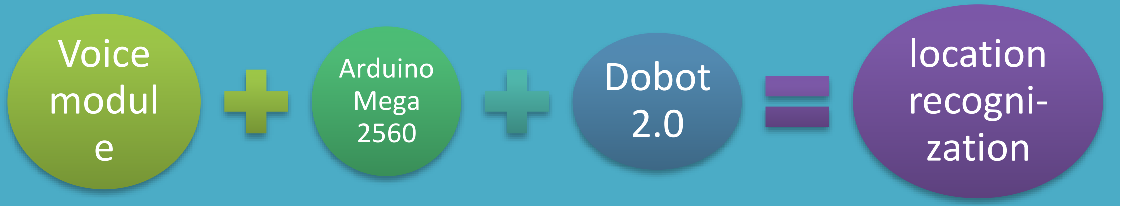
TWO、 How to achieve secondary development

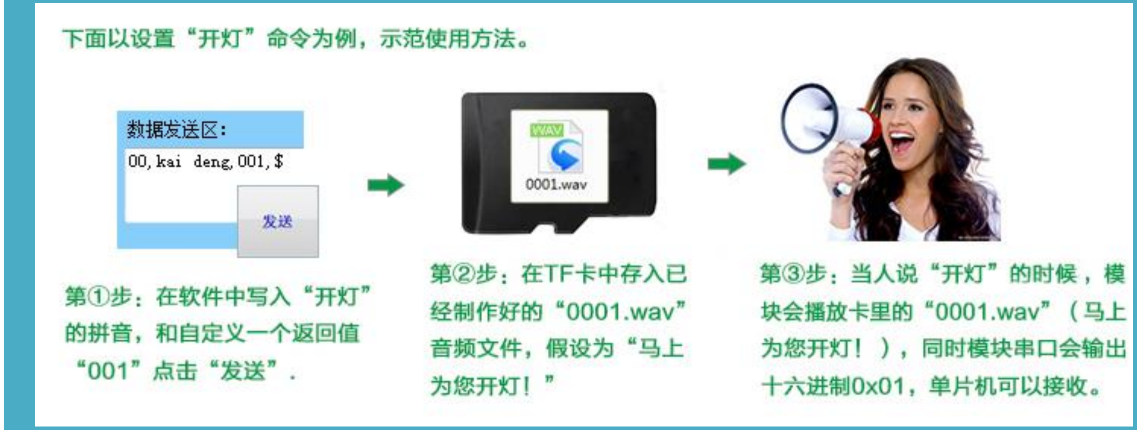


5. 2 voice recognition module

How to make Dobtot listen to you？
Increase a voice recognition module！

Voice module：			
GND	5V	TXD	RXD
Arduino Mega 2560			
GND	5V	TXD	RXD


Voice module + Arduino Mega 2560 + Dobot 2.0 = location recogni-zation


下面以设置“开灯”命令为例，示范使用方法。

数据发送区：
00, kai deng, 001, $

第①步：在软件中写入“开灯”的拼音，和自定义一个返回值“001”点击“发送”.

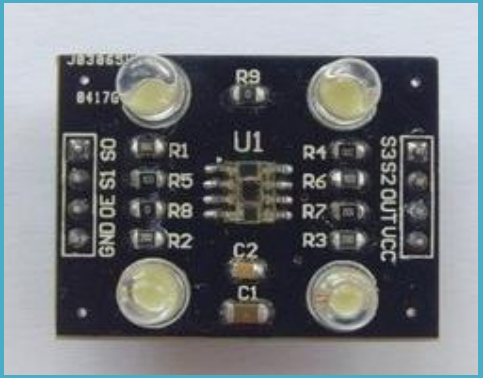
第②步：在TF卡中存入已经制作好的“0001.wav”音频文件，假设为“马上为您开灯！”

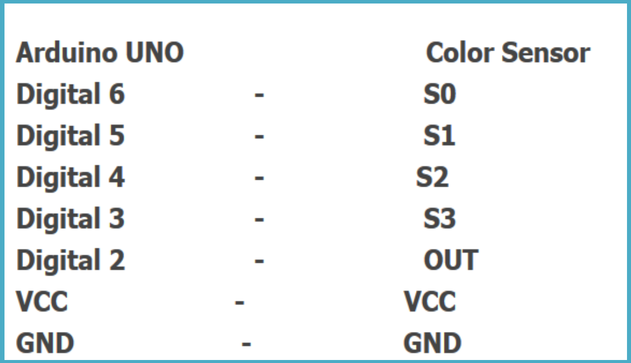
第③步：当人说“开灯”的时候，模块会播放卡里的“0001.wav”（马上为您开灯！），同时模块串口会输出十六进制0x01，单片机可以接收。

TWO、 How to achieve secondary development

5. 3 Vision Detection Module

CCD/TCS230 module
+
Arduino UNO
+
Dobot2.0
=
Vision





Arduino UNO		Color Sensor
Digital 6	-	S0
Digital 5	-	S1
Digital 4	-	S2
Digital 3	-	S3
Digital 2	-	OUT
VCC	-	VCC
GND	-	GND

The principle of TCS230 color recognition

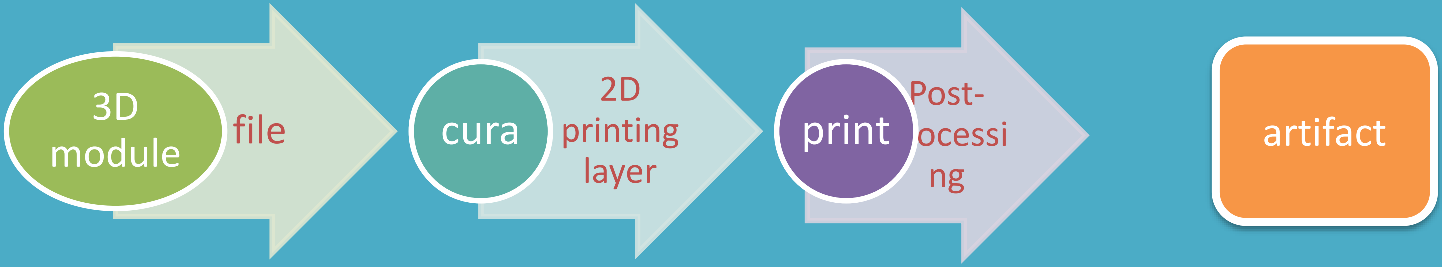
According to the above three primary color sensing principle, if the values of the three primary colors constituting the respective colors are known, the color of tested object can be known. For the TCS230, when a color filter is selected, it only allows a certain primary color to pass, the passage of other primary colors will be prevented.

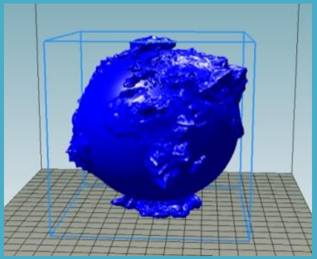
Example： When the red filter is selected, only red of incident light can pass through ， both blue and green light will be blocked, so that the intensity of the red light can be obtained. Similarly, by selecting other filters, blue and green light can be obtained. With these three values, the color of the light projected onto the TCS230 sensor can be analyzed.

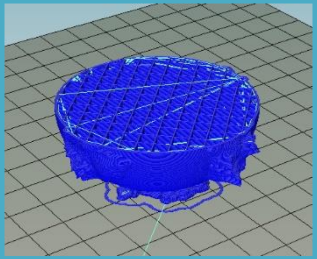
Three、 New Function Outline： 3D Printing Basics

1. Shaped Priciple： FDM、SLA/DLP、SLS

3D printing technology is the technology that quickly shaped. It is based on a digital model, using the bondable material such as powdered metal or plastic, and constructed object by layer-by-layer printing.







Three、 New Function Outline： 3D Printing Basics

3D Printing Main Molding Method

FDM melt layer forming technology

http://v.youku.com/v_show/id_XMTMxMjUyODAxMg==.html?from=s1.8-1-1.2

DLP digital light curing / SLA stereo curing

http://v.youku.com/v_show/id_XOTE0NzI2NjIw.html?from=s1.8-1-1.2#paction

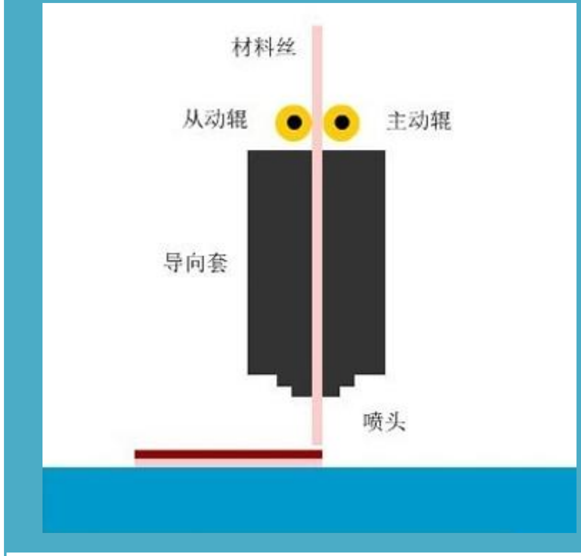
SLS selective laser sintering

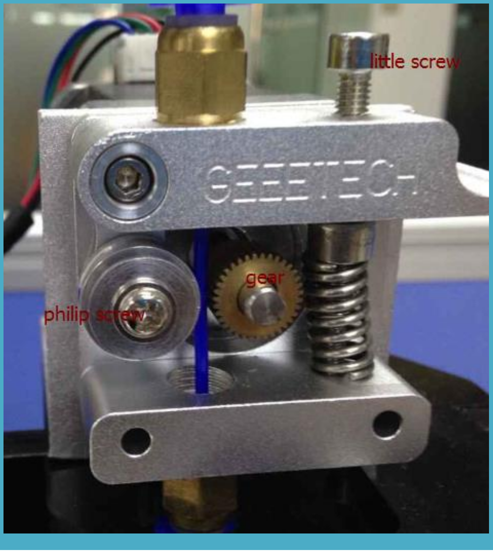
http://v.youku.com/v_show/id_XMTMxNDg5NTI5Mg==.html?from=s1.8-1-1.2

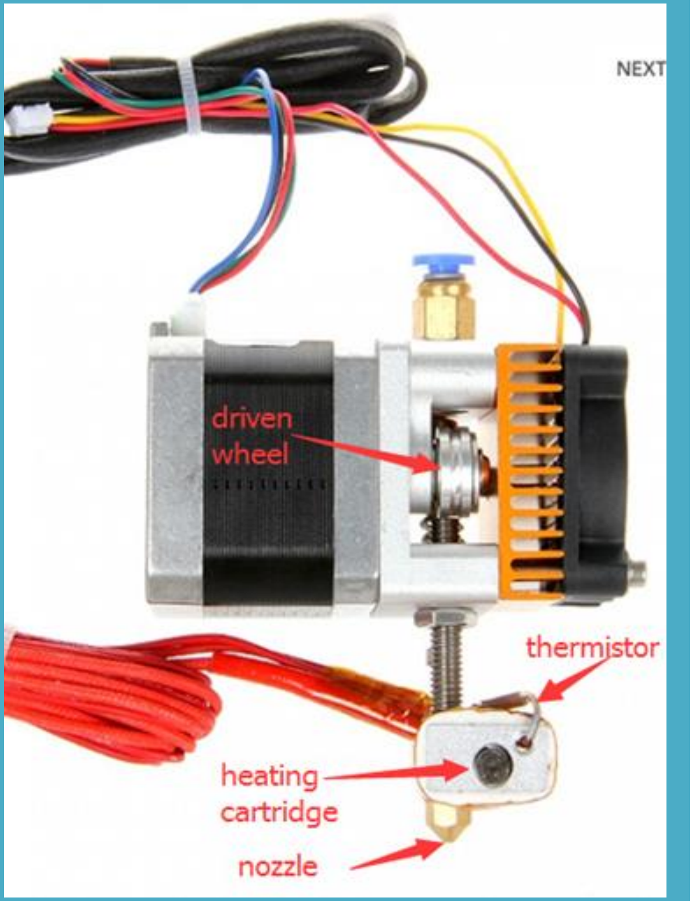
Three、 New Function Outline： 3D Printing Basics

2. FDM Printing Structure（MK8 extruder）

- Extruder
- Hotend
- Filament： ABS/PLA
- Heating cartridge
- Thermistor
- Driven wheel
- Nozzle： 0.2-0.5mm



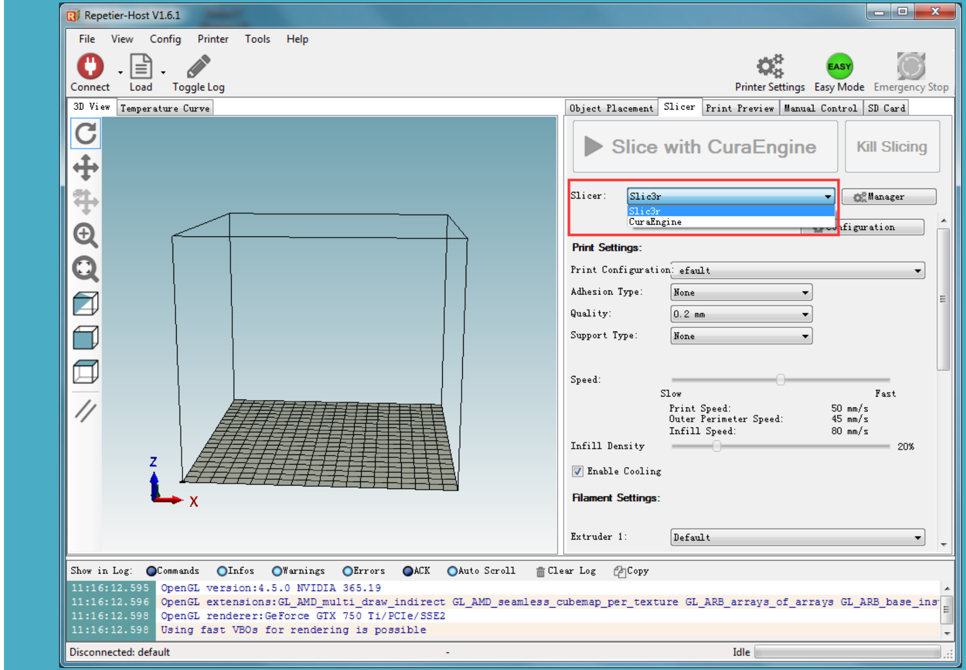




Three、 new function outline： 3D printing basics

3. 1 FDM Printer Software

The control software commonly used in printers is Repetier host, which is excellent and free . Its lice engine is Slic3r and cura, which can generate support, control slice height, print temperature, print speed and so on.



Slicer: Slic3r
- Slic3r
- CuraEngine

Print Settings:

Print Configuration: efault

Adhesion Type: None

Quality: 0.2 mm

Support Type: None

Speed:

```
Print Speed:              50 mm/s
Outer Perimeter Speed:    45 mm/s
Infill Speed:             80 mm/s
```

Infill Density 20%

☑ Enable Cooling

Filament Settings:

Extruder 1: Default

Show in Log: ● Commands ○ Infos ○ Warnings ○ Errors ● ACK ○ Auto Scroll Clear Log Copy

```
11:16:12.595 OpenGL version:4.5.0 NVIDIA 365.19
11:16:12.596 OpenGL extensions:GL_AMD_multi_draw_indirect GL_AMD_seamless_cubemap_per_texture GL_ARB_arrays_of_arrays GL_ARB_base_ins
11:16:12.598 OpenGL renderer:GeForce GTX 750 Ti/PCIe/SSE2
11:16:12.598 Using fast VBOs for rendering is possible
```

Disconnected: default - Idle

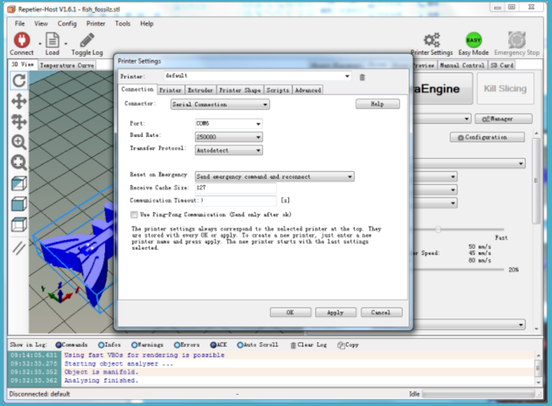
3. 2 key parameter：

- Print accuracy： layer height 0.1-0.3mm
- Print speed： 0-100mm/s
- Print temperature：
  PLA： 190-210 heat bed： 55-75
  ABS： 220-240 heat bed： 90-110
- Fill rate： 0-100%， usually set as 20%-30%

Three、 New Function Outline： 3D Printing Basics

3. 2 Repetier Host Software Using Method

- Connect and set printers parameter
- Imput stl file
- Select avaliable slice parameter
- Start printing



Printer Settings

Printer: default

Connection | Printer | Extruder | Printer Shape | Scripts | Advanced

Connector: Serial Connection Help

Port: COM6

Baud Rate: 250000

Transfer Protocol: Autodetect

Reset on Emergency: Send emergency command and reconnect

Receive Cache Size: 127

Communication Timeout:) [s]

☐ Use Ping-Pong Communication (Send only after ok)

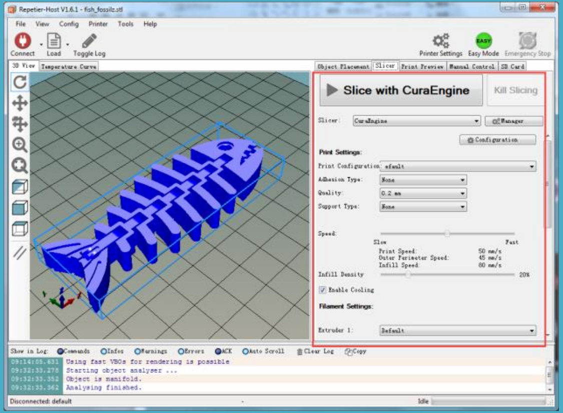
The printer settings always correspond to the selected printer at the top. They are stored with every OK or apply. To create a new printer, just enter a new printer name and press apply. The new printer starts with the last settings selected.

OK Apply Cancel

Show in Log: ● Commands ○ Infos ○ Warnings ○ Errors ● ACK ○ Auto Scroll Clear Log Copy

```
09:14:05.631  Using fast VBOs for rendering is possible
09:32:33.278  Starting object analyser ...
09:32:33.352  Object is manifold.
09:32:33.362  Analysing finished.
```

Disconnected: default - Idle



Slicer: CuraEngine

Print Settings:

Print Configuration: efault

Adhesion Type: None

Quality: 0.2 mm

Support Type: None

Speed: Slow … Fast

Print Speed: 50 mm/s

Outer Perimeter Speed: 45 mm/s

Infill Speed: 80 mm/s

Infill Density 20%

☑ Enable Cooling

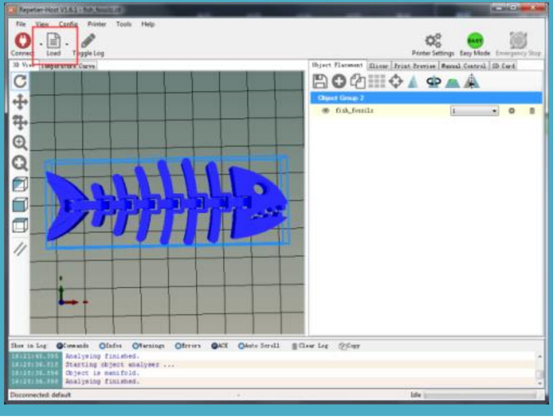
Filament Settings:

Extruder 1: Default

Show in Log: ● Commands ○ Infos ○ Warnings ○ Errors ● ACK ○ Auto Scroll Clear Log Copy

```
09:14:05.631  Using fast VBOs for rendering is possible
09:32:33.278  Starting object analyser ...
09:32:33.352  Object is manifold.
09:32:33.362  Analysing finished.
```

Disconnected: default - Idle



Show in Log: ● Commands ○ Infos ○ Warnings ○ Errors ● ACK ○ Auto Scroll Clear Log Copy

```
[illegible]  Analysing finished.
[illegible]  Starting object analyser ...
[illegible]  Object is manifold.
[illegible]  Analysing finished.
```

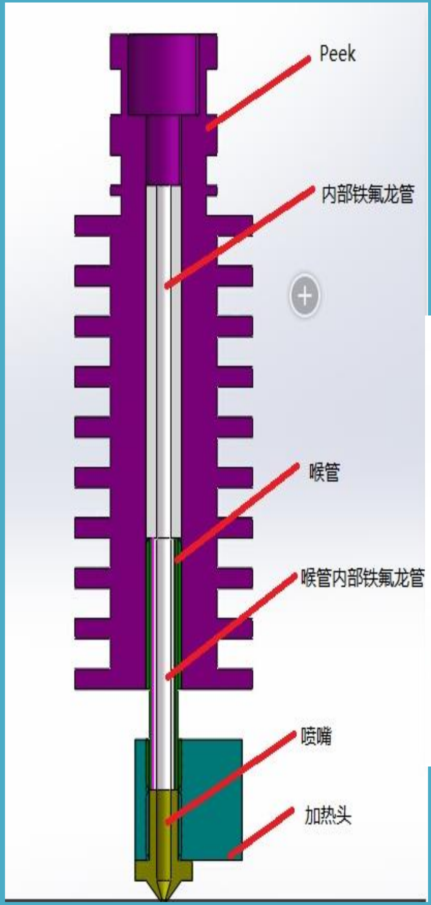
Disconnected: default - Idle

Three、 New Function Outline： 3D Printing Basics

4. Printer Common Problems and Improvement Methods

Printer Nozzle Blocked

- After the nozzle is blocked, the sound of "drips" will generally be emitted, and the stepper motor shaft will vibrate back and forth.
- The first layer is too close to the hot bed, the material problem or the carbonization of the filament in internal nozzle will cause blockage
- Manually adjust the nozzle temperature to 220 ° C. After the temperature is reached, manually squeeze the glue into the feed throat tube to make the nozzle work normally.
- Use a 0.3mm wire to clear the nozzle.

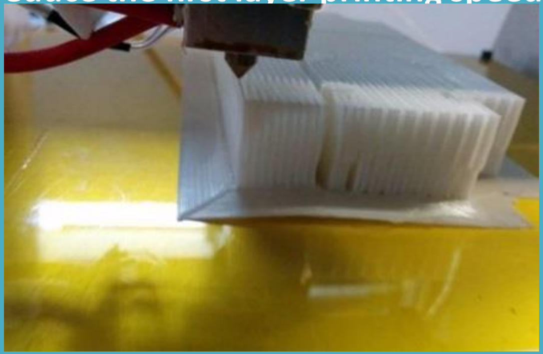


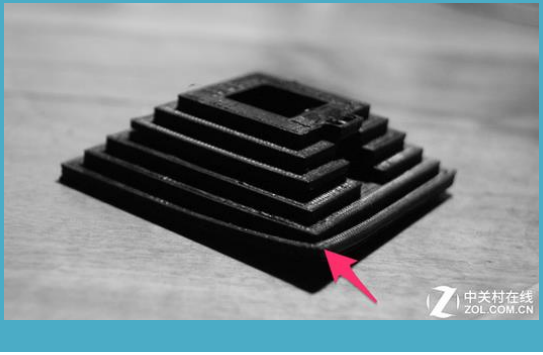
Three、 New Function Outline： 3D Printing Basics

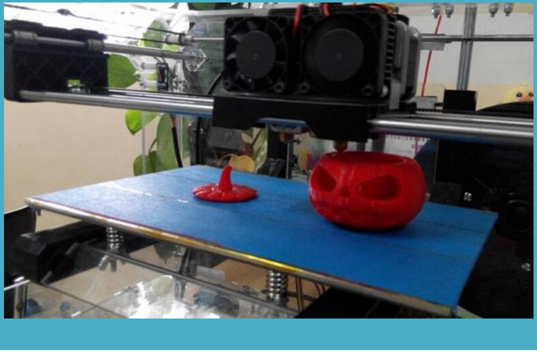
4. Printer Common Problems and Improvement Methods

Print object warp

The curling occurs when the first layer of plastic shrinks due to cooling and the edges of the mold are rolled up;
It is recommended to use a hot bed to keep the plastic at a temperature that will not cure;
Apply a thin layer of glue evenly on the print bed to increase adhesion;
At the same time reduce the first layer printing speed.







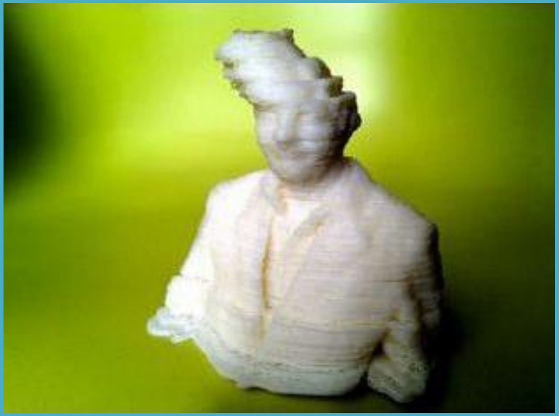
Three、 New Function Outline： 3D Printing Basics

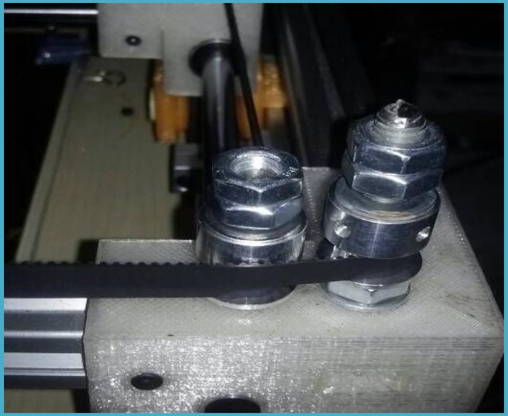
4. Printer Common Problems and Improvement Methods

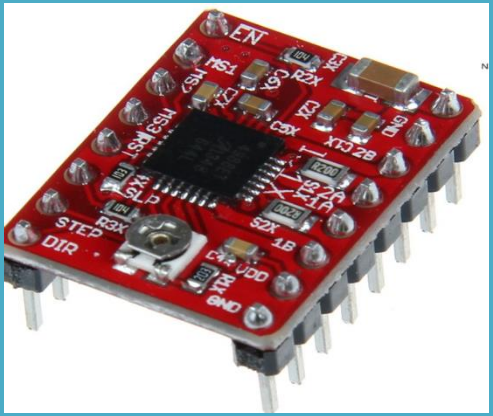
Layer of object printing mistakes

Generally, it is caused by lose step of the stepping motor that effected by resistance or other external force, or missing point caused by the loose belt of the X/Y.

- Need to check whether the bearings, belts, etc. are slippery or rusty ;
- Whether the current of motor driver is too small to result the torque insufficient .





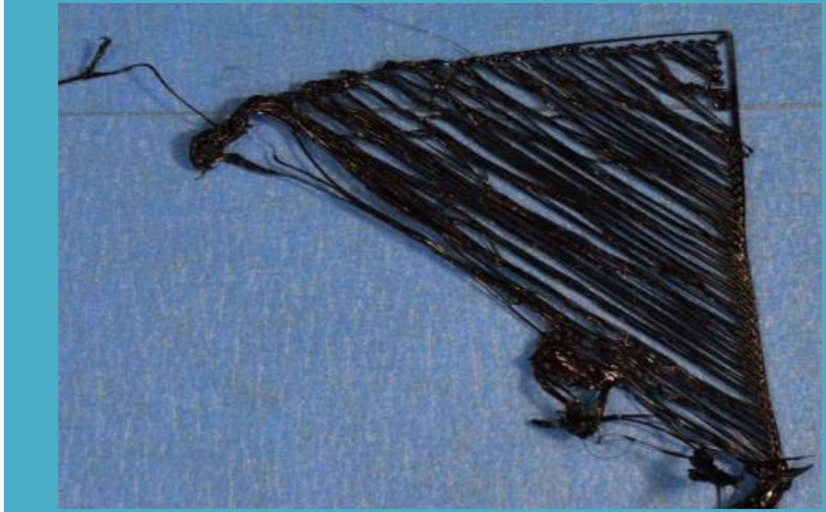


Three、 New Function Outline： 3D Printing Basics

4. Printer common problems and improvement methods

The first floor is not stuck on the print bed

- Make sure the printer bed is horizontal；
- Make sure the nozzle is at a proper distance from the print bed, too far to cause stickiness, too close to cause the nozzle to block；
- Try to reduce the speed of the first layer and use glue to improve adhesion.



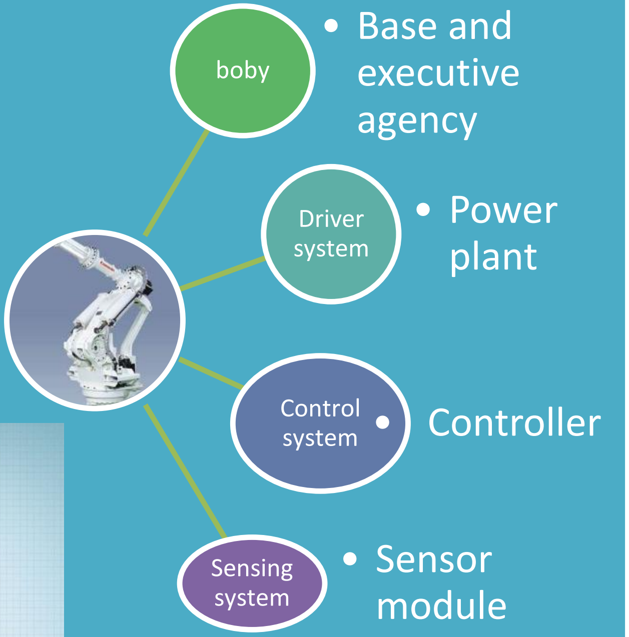
More detailed quality improvement methods please visit： https://www.simplify3d.com/support/print-quality-troubleshooting/#print-not-sticking-to-the-bed

Four 、 Industrial Robots

4.1 Industrial robots outline

Characteristics of industrial robot：

- ➢ programmable： “Flexible Startup” ,it can be reprogrammed as the work environment changes.
- ➢ Personification: “Biosensors” such as skin contact sensors, vision sensors, acoustic sensors, language functions, etc.
- ➢ Versatility： Different tasks can be done by replacing the hand end operator (hand grip, tool, etc.).



- Base and executive agency
- Power plant
- Controller
- Sensor module

Four 、 Industrial Robots

4.2 Industrial robots classification

Motion mode

- Cartesian coordinate
- Cylindrical coordinate
- Spherical coordinate
- Articular type

Function control

- Point type: point to point positioning
- Continuous track type: given track motion

Programming input

- Program iuput type
- Teaching input type

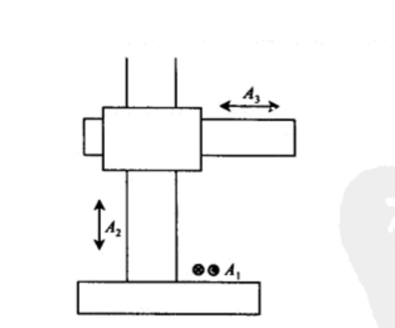


图 2-1 直角坐标型机器人

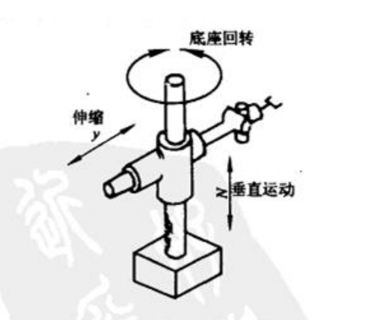


图 2-2 圆柱坐标型机器人

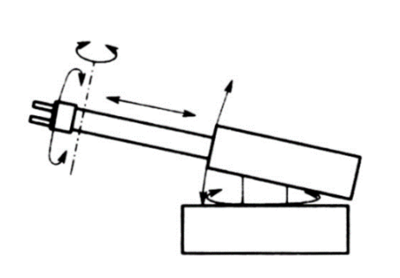


图 3 球坐标型机器人

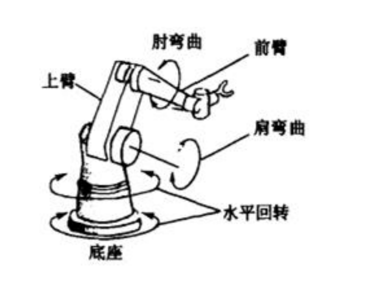


图 2-4 全关节型机器人

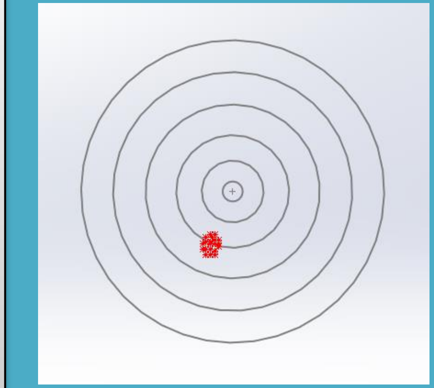
Four 、 Industrial Robots

4.3 Industrial robots technical indicators

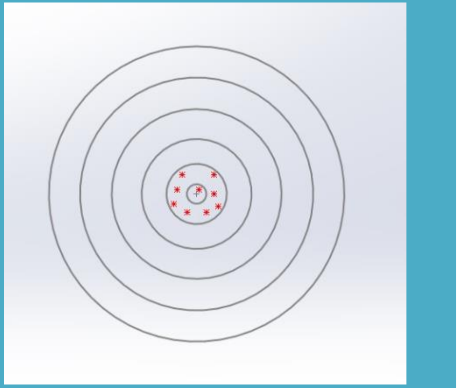
- Positioning Accuracy: Refers to the actual position reached when your CNC equipment is stopped and the position error you are asked to reach. For example, if you ask an axis to go 100 mm, the result is actually 100.01, and the extra 0.01 is the positioning accuracy.

- Repeated positioning accuracy: refers to the same position, the error caused by two positioning in the past. For example, if you ask an axis to go 100 mm, the first time he actually goes 100.01, repeating the same action, he walks 99.99, and the error 0.02 between them is the repeat positioning accuracy.

The repeat positioning accuracy mainly reflects the accuracy, and the positioning accuracy mainly reflects the accuracy, and the combination of the two is a comprehensive precision. For example, if the target is accurate, the positioning accuracy is good, but it is not necessarily on the 10 ring. If the positioning accuracy is good, it is close to the 10 ring. It may be that the 10 ring is distributed up and down.



Repeated positioning accuracy



Positioning Accuracy

Five、 Robot Vision

5.1 Robot Vison Outline

Machine vision, that is, using a machine instead of the human eye to make measurements and judgments.

Machine vision system refers to the image capture by machine vision products (ie, image capture device, divided into CMOS and CCD), and then the image is transmitted to the processing unit, through digital processing, according to pixel distribution and brightness, color, etc. Information to determine the size, shape, color, etc. Further, based on the result of the discrimination, the device operation at the site is controlled.

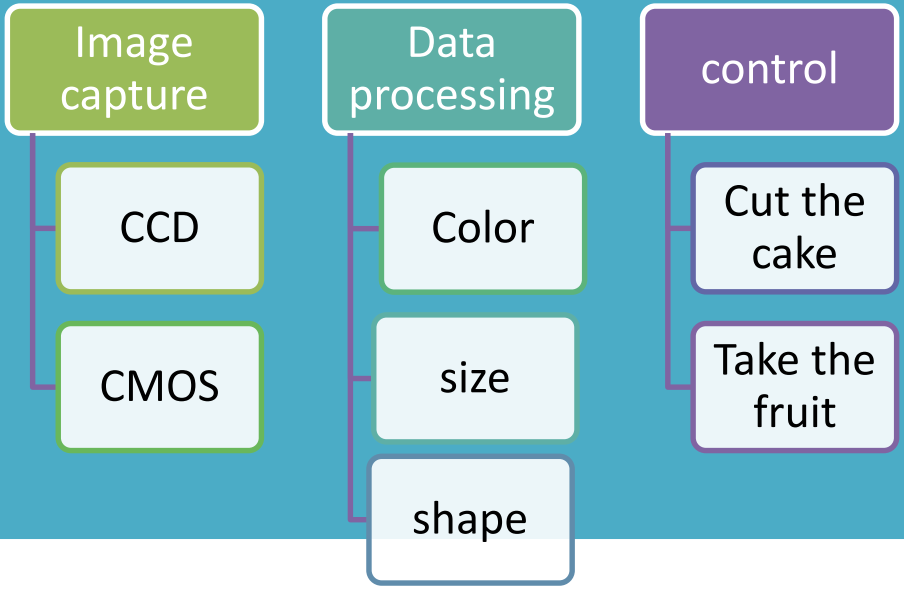


Image capture
CCD
CMOS
Data processing
Color
size
shape
control
Cut the cake
Take the fruit

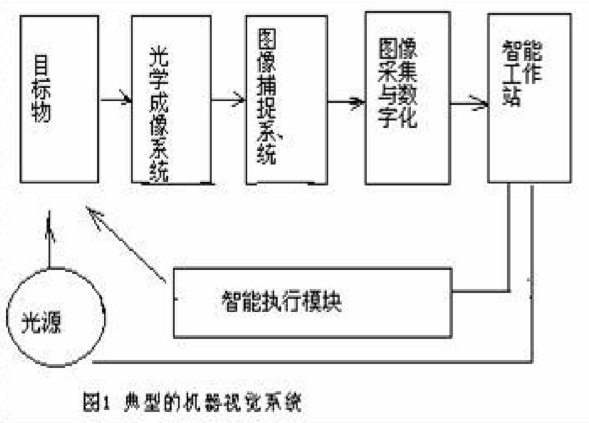
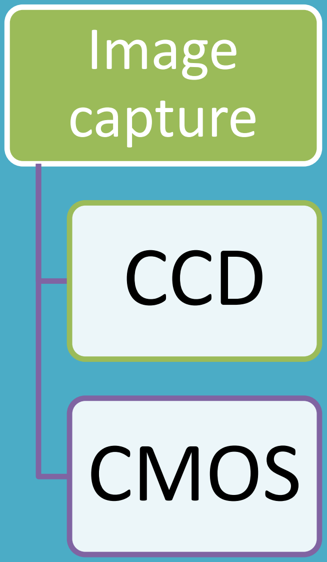
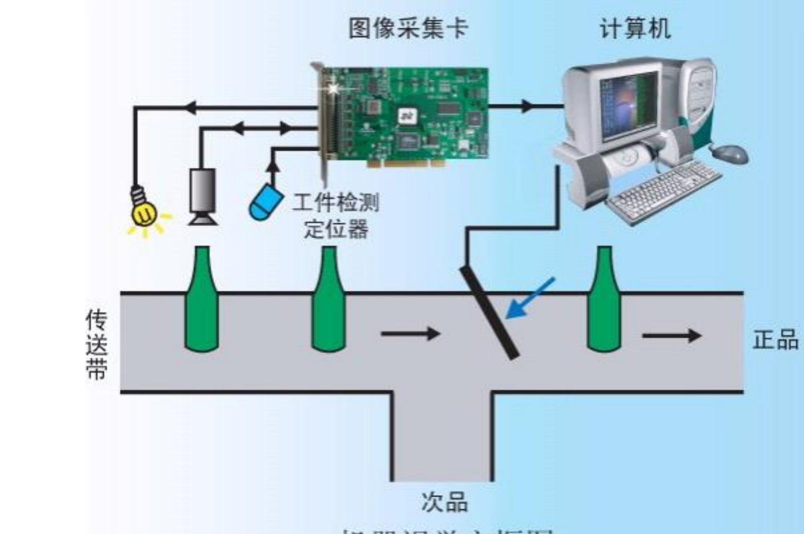


图1 典型的机器视觉系统

Five、 Robot Vision

5.2 Image Capture





机器视觉方框图

Five、 Robot Vision

5.3 Principle of image recognition

1. Template matching model----pattern recognition, It is a process of automatically recognizing and evaluating shapes, patterns, curves, numbers, character formats and graphics based on a large amount of information and data, based on expert experience and existing knowledge, using computer and mathematical reasoning methods.
State of learning：Select features for the sample and find the law of classification
State of achievement： according to classification rules to class and identify unknown sample

- The template matching model is simple and straightforward, and it is easy to get practical applications.
- However, this model emphasizes that the image must be fully compliant with the template in the brain to be recognized. At the same time, if each image identified has a corresponding template in the brain, it is also impossible.

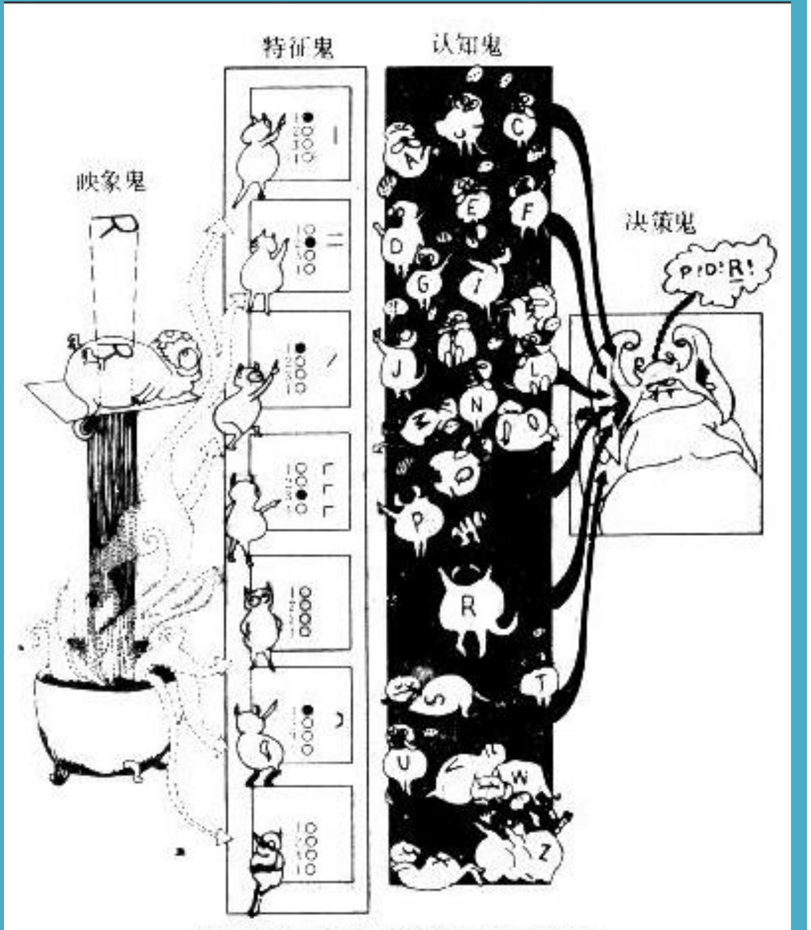
For example, if there is a letter A, if there is an A template in the brain, the size, orientation and shape of the letter A are exactly the same as the A template, and the letter A is recognized.

Five、 Robot Vision

5.3 Principle of image recognition

2. Prototype matching model ---- "Panmo" recognition model
Not storing a myriad of templates, but some "similarity" of the image
For example, when identifying the letter R:"Image ghost" encodes R first；

- “Character Ghost” reports a vertical line, two horizontal lines, one oblique line, three right angles and one discontinuous curve included in the image;
- "Cognitive ghosts" identify whether they are the letters they are responsible for based on the characteristics and their number. At this time, the D, P, and R ghosts will react, but the P ghost has only four characteristics to match it. The D ghost has only three features that match it. Only the R ghost has five features that match it, so the R ghost has the loudest shouting.
- The “decision ghost” makes it easy to make the decision to choose R.



"泛魔"识别模型图象识别示意图