

****SHANGHAI JIAO TONG UNIVERSITY

**数据库软件说明书**

|  |  |
| --- | --- |
| **项目名称：** | 类人机器人遥操作控制 |
|  |  |
| **学生姓名：** | 孙志超 |
|  |  |
| **指导教师：** | 贡亮 |
| **教学班级：** | 机电4班 |
| **学 院：** | 机械与动力工程学院 |
| **项目时间：** | 2021年 4月 10日 至 2021年 8月5 日 |

目 录

[1 软件安装说明 1](#_Toc25761)

[1.1 PyCharm 1](#_Toc9316)

[1.2 Mongodb 11](#_Toc3798)

[1.3 Robo 3T 12](#_Toc11587)

[2 软件源代码 15](#_Toc26597)

[2.1 数据写入 15](#_Toc10418)

[2.2 视频流数据写入 18](#_Toc15587)

[2.3 数据库构建 19](#_Toc17012)

[2.4 数据匹配搜索 21](#_Toc22914)

[3 代码说明 24](#_Toc4748)

[3.1 数据写入 24](#_Toc19583)

[3.2 视频流数据写入 26](#_Toc4188)

[3.3 数据库构建 28](#_Toc19669)

[3.4 数据匹配搜索 30](#_Toc10155)

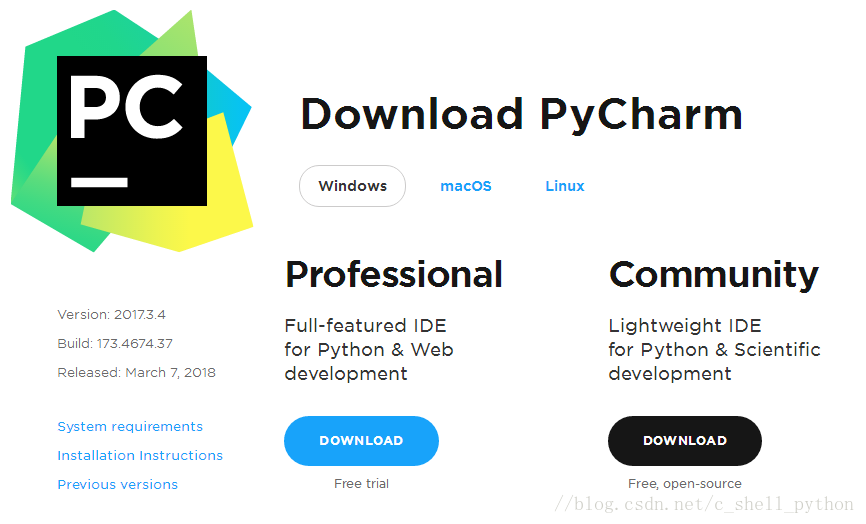
# 软件安装说明

## PyCharm

PyCharm 是一款功能强大的 Python 编辑器，具有跨平台性，鉴于目前最新版 PyCharm 使用教程较少，为了节约时间，来介绍一下 PyCharm 在 Windows下是如何安装的。

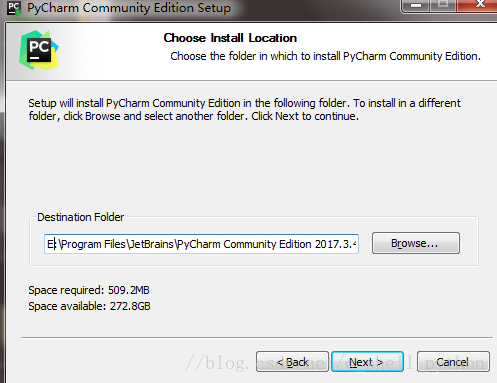
这是 PyCharm 的下载地址：[http://www.jetbrains.com/pycharm/download/#section=windows](http://www.jetbrains.com/pycharm/download/" \l "section=windows" \t "https://www.runoob.com/w3cnote/_blank)

进入该网站后，我们会看到如下界面:

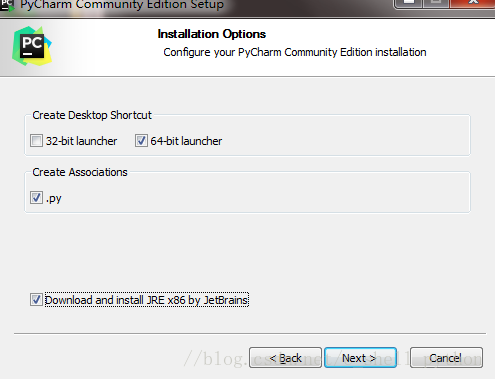


professional 表示专业版，community 是社区版，推荐安装社区版，因为是免费使用的。

1、当下载好以后，点击安装，记得修改安装路径，我这里放的是E盘，修改好以后，Next

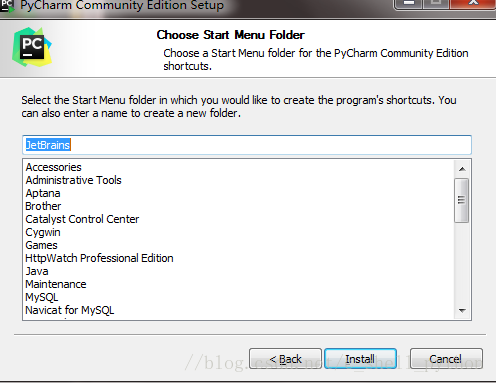


2、接下来是



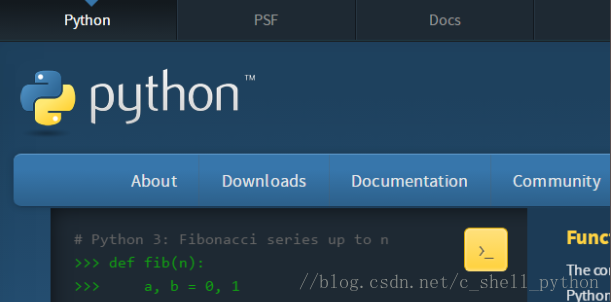
我们可以根据自己的电脑选择32位还是64位，目前应该基本都是64位系统吧

3、如下



点击Install,然后就是静静的等待安装了。如果我们之前没有下载有Python解释器的话，在等待安装的时间我们得去下载python解释器，不然pycharm只是一副没有灵魂的驱壳

4、进入python官方网站：//www.python.org/



点击Downloads,进入选择下载界面

5、如下所示，选择我们需要的python版本号，点击Download



6、我选择的是python3.5.1，会看到如下界面



因为我们需要用到的是Windows下的解释器，所以在Operating System中可以选择对应的Windows版本，有64位和32位可以选择，我选择的是画红线的这个，executable表示可执行版，需要安装后使用，embeddable表示嵌入版，就是解压以后就可以使用的版本。

可执行版安装比较简单，一直默认就好了。embeddable需要注意，当我们解压IMG_263这个也是需要解压到同一路径的，这里面放着pip、setuptools等工具，如果不解压，我们将无法在pycharm中更新模块，比如需要用到pymysql，就无法下载。虽然也能用，但是就是"阉割版"的python解释器了。

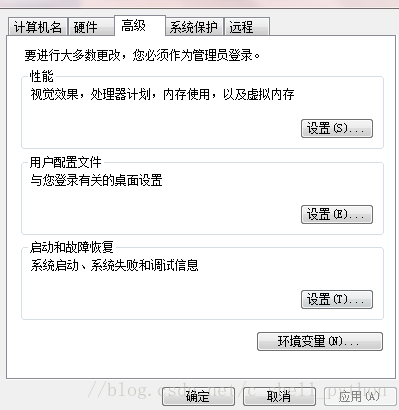
如果是embeddable版，记得把解释器所在的路径添加到环境变量里，不然pycharm无法自动获得解释器位置。

7、添加环境变量

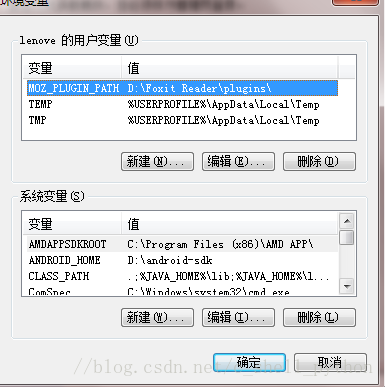
（1）右键我的电脑，点击属性，弹出如下界面



（2）点击高级系统设置，出现下图



（3）点击环境变量

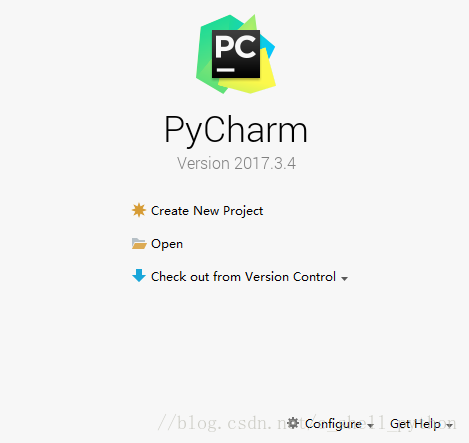


（4）找到系统变量里面的Path，编辑它，将python解释器所在路径粘贴到最后面，再加个分号。

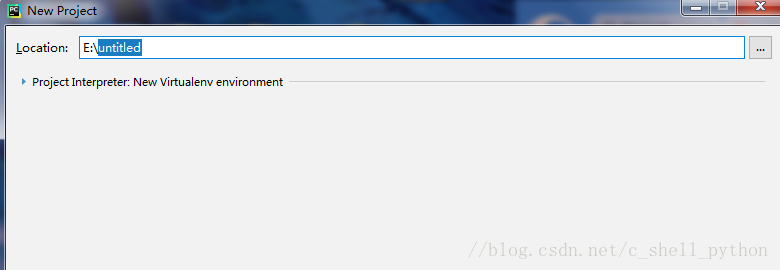


环境变量配置结束

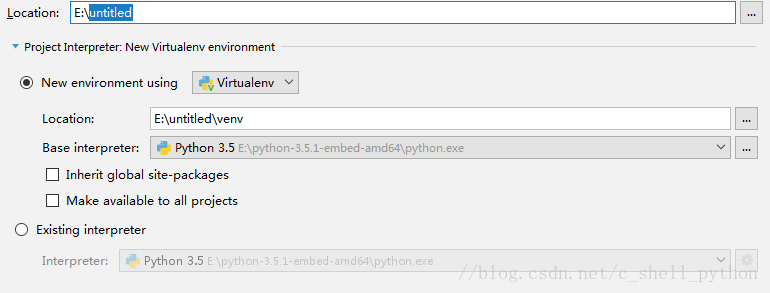
8、这时候Pycharm也装好了，我们进入该软件。



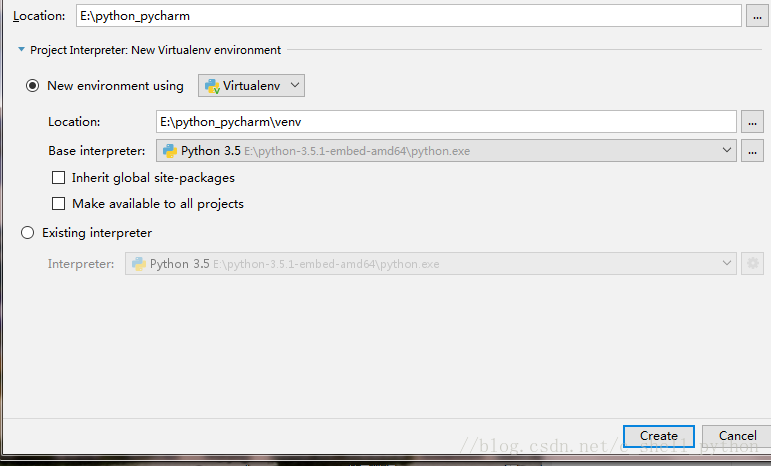
9、点击Create New Project，接下来是重点



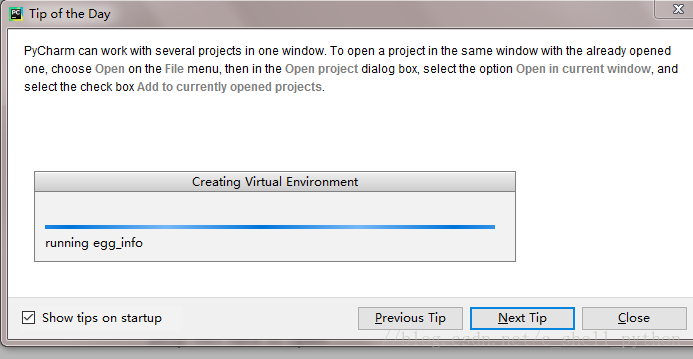
Location是我们存放工程的路径，点击IMG_270这个三角符号，可以看到pycharm已经自动获取了Python 3.5。



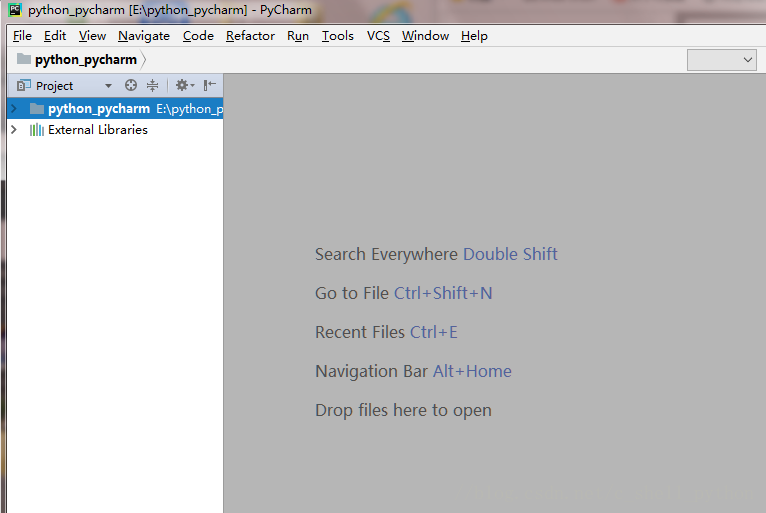
点击第一个IMG_272我们可以选择Location的路径，比如



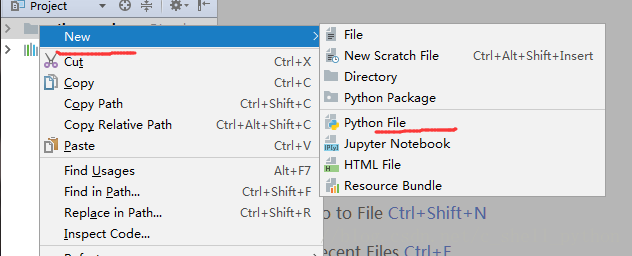
记住，我们选择的路径需要为空，不然无法创建，第二个Location不用动它，是自动默认的，其余不用点，然后点击Create。出现如下界面，这是Pycharm在配置环境，静静等待。最后点击close关掉提示就好了。



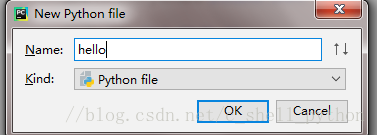
10、建立编译环境



右键IMG_276点击New,选择Python File



给file取个名字，点击OK

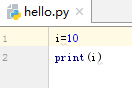


系统会默认生成hello.py

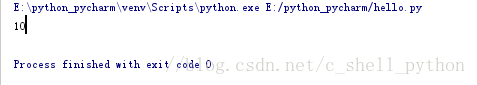


好了，至此，我们的初始工作基本完成。

11、我们来编译一下



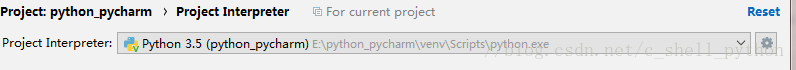
快捷键ctrl+shift+F10或者点击IMG_281绿色三角形,就会编译，编译结果如下



12、对了，因为我之前已经添加过了，所以可以直接编译，还有很重要的一步没说，不然pycharm无法找到解释器，将无法编译。

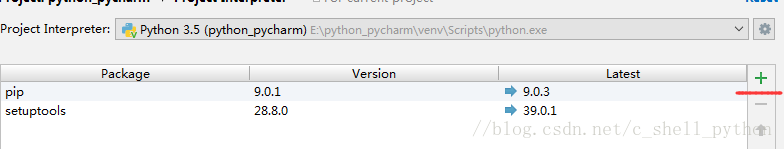
点击File,选择settings,点击IMG_283

添加解释器



最后点击Apply。等待系统配置。

如果我们需要添加新的模块，点击绿色+号



然后直接搜索pymysql



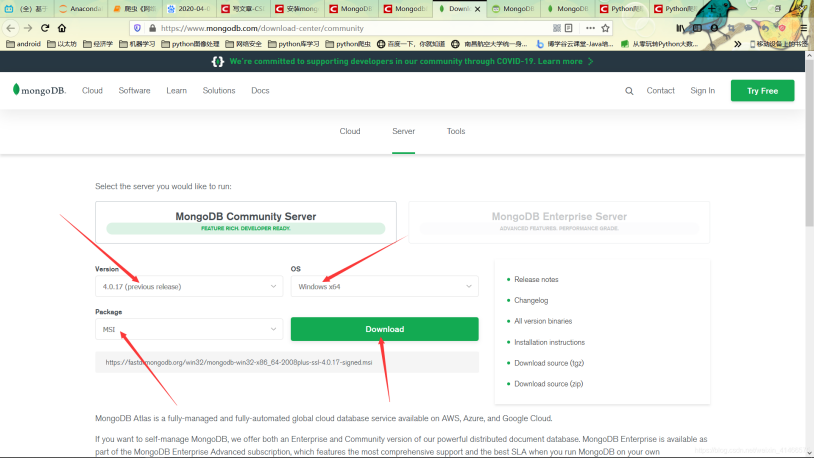
然后点安装

IMG_287

以上就是pycharm的安装过程以及初始化，还有Python解释器的安装配置。

## Mongodb

进入页面 https://www.mongodb.com/download-center/community ，选择如下：



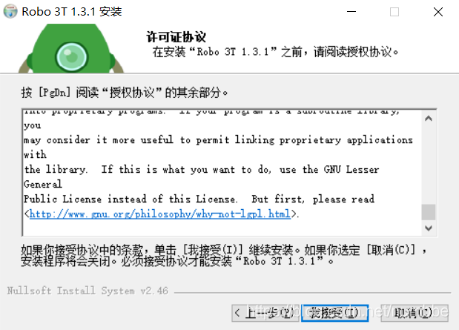
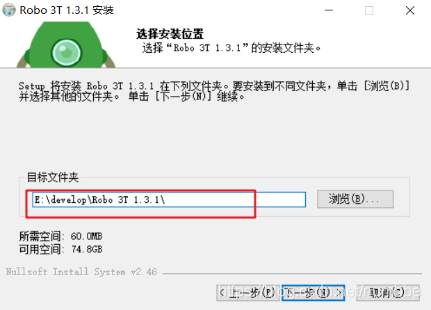
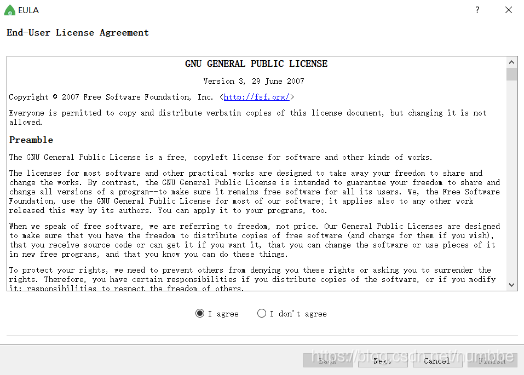
双击开始安装（以下几步注意，其他全部next）。选择你想安装到的位置。点击ok，然后点击next 点击之后 新版的Mongodb会主动在安装目录下创建log和data文件，这相比较早期版本有了改善 ，早期需要自己手动创建。但我们仍然需要在data下创建db文件夹。最后安装成功会弹出一个警告框。

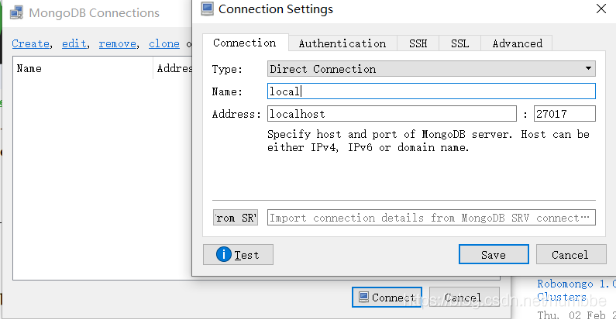
至此，MongoDB客户端已经安装完毕

## Robo 3T

<https://robomongo.org/>

双击文件。

robo3t-1.3.1-windows-x86\_64-7419c406.exe  
  




# 软件源代码

## 数据写入

import os  
import time  
import pymongo  
import numpy as np  
from numpy import \*  
import datetime  
  
  
  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["database"]  
#连接图片集合  
mycol1 = mydb1["ArmJointPosition"]  
  
b=np.loadtxt('F:/DBTEST/temp.txt',usecols=0,dtype=np.str)  
startTime = str(b[0])  
print(startTime)  
  
b=np.loadtxt('F:/DBTEST/temp.txt',skiprows=1,dtype=np.str)  
  
for hang in range(0,len(b)):  
 endTime = (datetime.datetime.strptime(startTime, "%Y%m%d\_%H%M%S\_%f") + datetime.timedelta(  
 seconds=1 / 60.000000)).strftime("%Y%m%d\_%H%M%S\_%f")  
 mydict = {"time": str(startTime), "手部X": str(b[hang,1]), "手部Y": str(b[hang,2]), "手部Z": str(b[hang,3]),"肘部X": str(b[hang,4]), "肘部Y": str(b[hang,5]), "肘部Z": str(b[hang,6])}  
 # 加入数据库  
 mycol1.insert\_one(mydict)  
 startTime = endTime  
  
  
  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["database"]  
#连接图片集合  
mycol1 = mydb1["ManipulatorMapping"]  
  
b=np.loadtxt('F:/DBTEST/temp.txt',usecols=0,dtype=np.str)  
startTime = str(b[0])  
print(startTime)  
  
b=np.loadtxt('F:/DBTEST/temp.txt',skiprows=1,dtype=np.str)  
  
for hang in range(0,len(b)):  
 endTime = (datetime.datetime.strptime(startTime, "%Y%m%d\_%H%M%S\_%f") + datetime.timedelta(  
 seconds=1 / 60.000000)).strftime("%Y%m%d\_%H%M%S\_%f")  
 mydict = {"time": str(startTime), "机械臂关节角1": str(b[hang,7]), "机械臂关节角2": str(b[hang,8]), "机械臂关节角3": str(b[hang,9]), "机械臂关节角4": str(b[hang,10]), "机械臂关节角5": str(b[hang,11]), "机械臂关节角6": str(b[hang,12])}  
 # 加入数据库  
 mycol1.insert\_one(mydict)  
 startTime = endTime  
  
  
  
  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["optimization"]  
#连接图片集合  
mycol1 = mydb1["CollisionData"]  
  
b=np.loadtxt('F:/DBTEST/temp.txt',usecols=0,dtype=np.str)  
startTime = str(b[0])  
print(startTime)  
  
b=np.loadtxt('F:/DBTEST/temp.txt',skiprows=1,dtype=np.str)  
  
for hang in range(0,len(b)):  
 endTime = (datetime.datetime.strptime(startTime, "%Y%m%d\_%H%M%S\_%f") + datetime.timedelta(  
 seconds=1 / 60.000000)).strftime("%Y%m%d\_%H%M%S\_%f")  
 mydict = {"time": str(startTime),"是否碰撞（-1：未准备，0：无碰撞，1：速度过快，2：危险，3：碰撞）": str(b[hang,13])}  
 # 加入数据库  
 mycol1.insert\_one(mydict)  
 startTime = endTime

## 视频流数据写入

import time  
from watchdog.observers import Observer  
from watchdog.events import PatternMatchingEventHandler  
import os  
import pymongo  
import re  
from pymongo import MongoClient  
  
  
#看门狗  
watch\_patterns = "\*.py;\*.txt" # 监控文件的模式  
ignore\_patterns = "" # 设置忽略的文件模式  
ignore\_directories = False # 是否忽略文件夹变化  
case\_sensitive = True # 是否对大小写敏感  
event\_handler = PatternMatchingEventHandler(watch\_patterns, ignore\_patterns, ignore\_directories, case\_sensitive)  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["visualization"]  
#连接图片集合  
mycol1 = mydb1["VideoStreaming"]  
ret = os.walk("F:/DBTEST/picture")#返回一个对象  
file\_size = 0 #初始化文件大小为零  
  
file\_name0="0"  
for base\_path,dir\_list,file\_list in ret:  
 for file\_name in file\_list:  
 file\_name0 =file\_name  
 print(file\_name0)  
 break  
file\_name0=file\_name0[0:15]  
print(file\_name0)  
  
ret = os.walk("F:\DBTEST\picture")#返回一个对象  
file\_size = 0 #初始化文件大小为零  
for base\_path,dir\_list,file\_list in ret:  
 for file\_name in file\_list:  
 file\_path = os.path.join(base\_path,file\_name)  
 size = os.path.getsize(file\_path)  
 tim = os.path.getctime(file\_path)  
 tim = time.ctime(tim)  
 type = os.path.splitext(file\_path)  
 #file\_size += size  
 #print(file\_path)  
 #print(file\_name)  
 a = str(file\_name)  
 a=a[0:15]  
 if a!=file\_name0:  
 print(a)  
 mydict = {"name": file\_name, "time": a, "value": file\_path}  
 mycol1.insert\_one(mydict)

## 数据库构建

import pymongo  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")   
#建立数据库   
mydb1 = myclient["database"]   
   
 #创建图片文件集合  
mycol1 = mydb1["ArmJointPosition"]  
mydict={"name":"ArmJointPosition0","time":0,"type":"0","value":"0"}  
mycol1.insert\_one(mydict)  
  
#创建音频文件集合  
mycol2 = mydb1["ManipulatorMapping"]  
mydict={"name":"ManipulatorMapping0","time":0,"value":"0"}  
mycol2.insert\_one(mydict)  
  
  
#创建算法库  
mydb2 = myclient["visualization"]  
  
#创建算法集合  
mycol6 = mydb2["VideoStreaming"]  
mydict={"name":"VideoStreaming0","time":0,"value":"0"}  
mycol6.insert\_one(mydict)  
  
  
#创建模型库  
mydb3 = myclient["optimization"]  
  
#创建模型集合  
mycol7 = mydb3["CollisionData"]  
mydict={"name":"CollisionData0","time":0,"value":"0"}  
mycol7.insert\_one(mydict)  
  
  
  
  
  
#创建算法库  
mydb4 = myclient["algorithm"]  
  
#创建算法集合  
mycol6 = mydb4["deeplearning"]  
mydict={"name":"deeplearning0","time":0,"value":"0"}  
mycol6.insert\_one(mydict)  
  
#创建模型库  
mydb5 = myclient["model"]  
  
#创建模型集合  
mycol7 = mydb5["transfer"]  
mydict={"name":"transfer0","time":0,"value":"0"}  
mycol7.insert\_one(mydict)  
  
#创建知识库  
mydb6 = myclient["knowledge"]  
  
#创建模型集合  
mycol8 = mydb6["target\_recongnition\_model"]  
mydict={"name":"model0","time":0,"value":"0"}  
mycol8.insert\_one(mydict)  
  
#创建知识图谱集合  
mycol9 = mydb6["knowledge\_graph"]  
mydict={"name":"graph0","time":0,"value":"0"}  
mycol9.insert\_one(mydict)

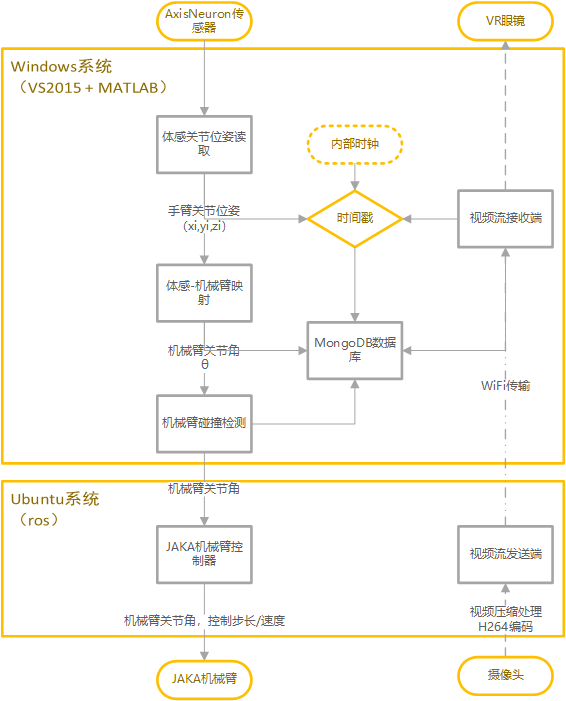
## 数据匹配搜索

import pymongo  
from pymongo import MongoClient  
import re  
  
  
# 创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
# 建立数据库  
  
  
path=0  
#keyW='2021\_07\_31\_04\_42\_05'  
keyW='20210804\_141445\_9'  
  
  
  
if path==0:  
 mydb1 = myclient["database"]  
 #只有string才能查询，double不行  
 for u in mydb1.ArmJointPosition.find({'time': re.compile(keyW)}):  
 print(u)  
  
if path==1:  
 mydb1 = myclient["database"]  
 #只有string才能查询，double不行  
 for u in mydb1.ManipulatorMapping.find({'time': re.compile(keyW)}):  
 print(u)  
  
if path==2:  
 mydb1 = myclient["optimization"]  
 #只有string才能查询，double不行  
 for u in mydb1.CollisionData.find({'time': re.compile(keyW)}):  
 print(u)  
  
if path==3:  
 mydb1 = myclient["visualization"]  
 #只有string才能查询，double不行  
 for u in mydb1.VideoStreaming.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
  
  
  
if path==4:  
  
  
 mydb1 = myclient["database"]  
 # 只有string才能查询，double不行  
 for u in mydb1.ArmJointPosition.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
 mydb1 = myclient["database"]  
 # 只有string才能查询，double不行  
 for u in mydb1.ManipulatorMapping.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
 mydb1 = myclient["optimization"]  
 # 只有string才能查询，double不行  
 for u in mydb1.CollisionData.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
 mydb1 = myclient["visualization"]  
 #只有string才能查询，double不行  
 for u in mydb1.VideoStreaming.find({'time': re.compile(keyW)}):  
 print(u)

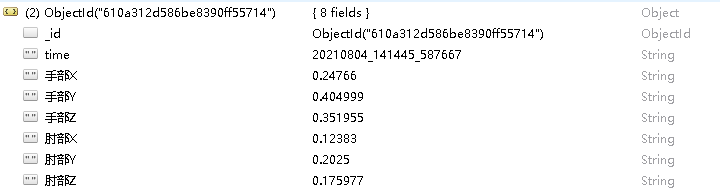
# 代码说明

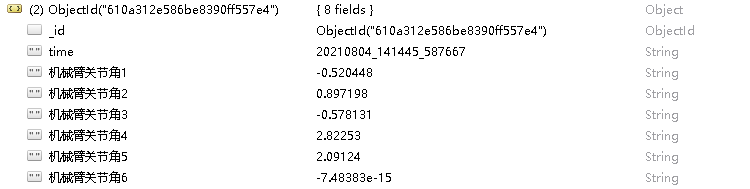
## 数据写入

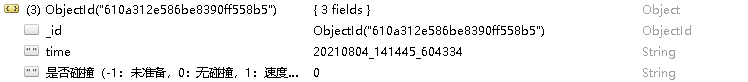
如下图所示，该部分代码主要涉及对手臂关节位姿，机械臂关节角以及碰撞信息数据的写入。



该部分存储的信息如下图所示，其中其中主要包括数据id，数据的时间戳，以及其他的相关信息。





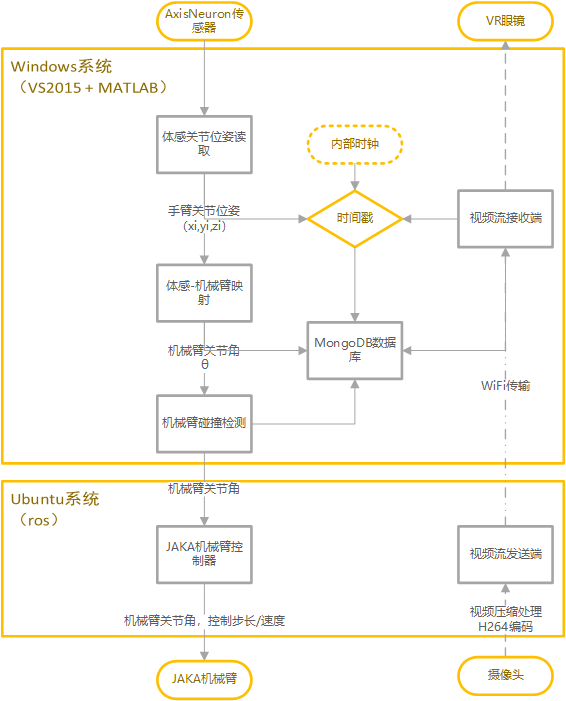


代码如下图所示。

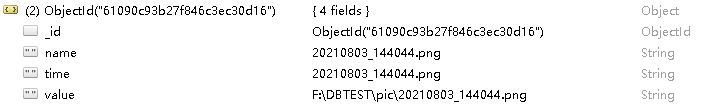
import os  
import time  
import pymongo  
import numpy as np  
from numpy import \*  
import datetime  
  
  
  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["database"]  
#连接图片集合  
mycol1 = mydb1["ArmJointPosition"]  
  
b=np.loadtxt('F:/DBTEST/temp.txt',usecols=0,dtype=np.str)  
startTime = str(b[0])  
print(startTime)  
  
b=np.loadtxt('F:/DBTEST/temp.txt',skiprows=1,dtype=np.str)  
  
for hang in range(0,len(b)):  
 endTime = (datetime.datetime.strptime(startTime, "%Y%m%d\_%H%M%S\_%f") + datetime.timedelta(  
 seconds=1 / 60.000000)).strftime("%Y%m%d\_%H%M%S\_%f")  
 mydict = {"time": str(startTime), "手部X": str(b[hang,1]), "手部Y": str(b[hang,2]), "手部Z": str(b[hang,3]),"肘部X": str(b[hang,4]), "肘部Y": str(b[hang,5]), "肘部Z": str(b[hang,6])}  
 # 加入数据库  
 mycol1.insert\_one(mydict)  
 startTime = endTime  
  
  
  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["database"]  
#连接图片集合  
mycol1 = mydb1["ManipulatorMapping"]  
  
b=np.loadtxt('F:/DBTEST/temp.txt',usecols=0,dtype=np.str)  
startTime = str(b[0])  
print(startTime)  
  
b=np.loadtxt('F:/DBTEST/temp.txt',skiprows=1,dtype=np.str)  
  
for hang in range(0,len(b)):  
 endTime = (datetime.datetime.strptime(startTime, "%Y%m%d\_%H%M%S\_%f") + datetime.timedelta(  
 seconds=1 / 60.000000)).strftime("%Y%m%d\_%H%M%S\_%f")  
 mydict = {"time": str(startTime), "机械臂关节角1": str(b[hang,7]), "机械臂关节角2": str(b[hang,8]), "机械臂关节角3": str(b[hang,9]), "机械臂关节角4": str(b[hang,10]), "机械臂关节角5": str(b[hang,11]), "机械臂关节角6": str(b[hang,12])}  
 # 加入数据库  
 mycol1.insert\_one(mydict)  
 startTime = endTime  
  
  
  
  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["optimization"]  
#连接图片集合  
mycol1 = mydb1["CollisionData"]  
  
b=np.loadtxt('F:/DBTEST/temp.txt',usecols=0,dtype=np.str)  
startTime = str(b[0])  
print(startTime)  
  
b=np.loadtxt('F:/DBTEST/temp.txt',skiprows=1,dtype=np.str)  
  
for hang in range(0,len(b)):  
 endTime = (datetime.datetime.strptime(startTime, "%Y%m%d\_%H%M%S\_%f") + datetime.timedelta(  
 seconds=1 / 60.000000)).strftime("%Y%m%d\_%H%M%S\_%f")  
 mydict = {"time": str(startTime),"是否碰撞（-1：未准备，0：无碰撞，1：速度过快，2：危险，3：碰撞）": str(b[hang,13])}  
 # 加入数据库  
 mycol1.insert\_one(mydict)  
 startTime = endTime

## 视频流数据写入

如下图所示，该部分代码主要涉及视频流数据的写入。



该部分存储的信息如下图所示。



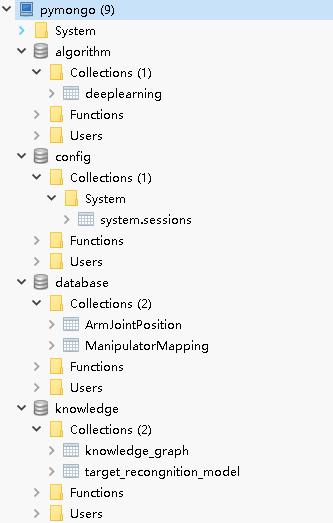
代码如下。

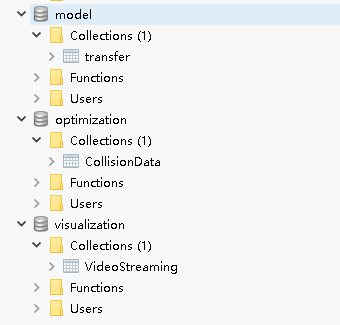
import time  
from watchdog.observers import Observer  
from watchdog.events import PatternMatchingEventHandler  
import os  
import pymongo  
import re  
from pymongo import MongoClient  
  
  
#看门狗  
watch\_patterns = "\*.py;\*.txt" # 监控文件的模式  
ignore\_patterns = "" # 设置忽略的文件模式  
ignore\_directories = False # 是否忽略文件夹变化  
case\_sensitive = True # 是否对大小写敏感  
event\_handler = PatternMatchingEventHandler(watch\_patterns, ignore\_patterns, ignore\_directories, case\_sensitive)  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
#连接数据库  
mydb1 = myclient["visualization"]  
#连接图片集合  
mycol1 = mydb1["VideoStreaming"]  
ret = os.walk("F:/DBTEST/picture")#返回一个对象  
file\_size = 0 #初始化文件大小为零  
  
file\_name0="0"  
for base\_path,dir\_list,file\_list in ret:  
 for file\_name in file\_list:  
 file\_name0 =file\_name  
 print(file\_name0)  
 break  
file\_name0=file\_name0[0:15]  
print(file\_name0)  
  
ret = os.walk("F:\DBTEST\picture")#返回一个对象  
file\_size = 0 #初始化文件大小为零  
for base\_path,dir\_list,file\_list in ret:  
 for file\_name in file\_list:  
 file\_path = os.path.join(base\_path,file\_name)  
 size = os.path.getsize(file\_path)  
 tim = os.path.getctime(file\_path)  
 tim = time.ctime(tim)  
 type = os.path.splitext(file\_path)  
 #file\_size += size  
 #print(file\_path)  
 #print(file\_name)  
 a = str(file\_name)  
 a=a[0:15]  
 if a!=file\_name0:  
 print(a)  
 mydict = {"name": file\_name, "time": a, "value": file\_path}  
 mycol1.insert\_one(mydict)

## 数据库构建

如下图所示，该部分代码主要涉及数据库架构的建立。

建立的数据库结构如下图所示。





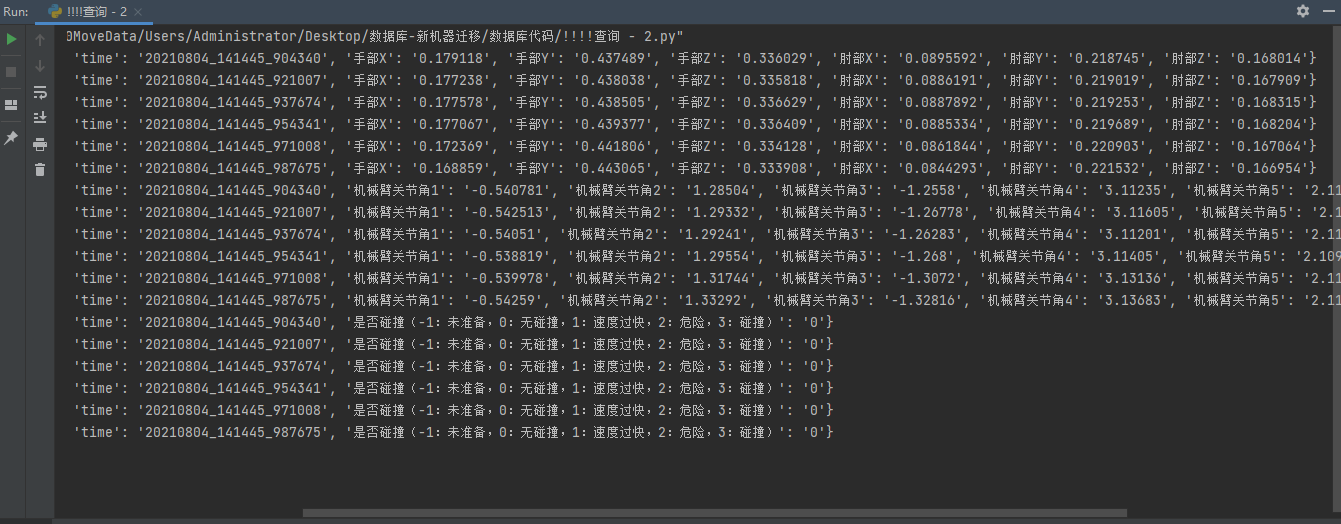
代码如下。

import pymongo  
  
#创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")   
#建立数据库   
mydb1 = myclient["database"]   
   
 #创建图片文件集合  
mycol1 = mydb1["ArmJointPosition"]  
mydict={"name":"ArmJointPosition0","time":0,"type":"0","value":"0"}  
mycol1.insert\_one(mydict)  
  
#创建音频文件集合  
mycol2 = mydb1["ManipulatorMapping"]  
mydict={"name":"ManipulatorMapping0","time":0,"value":"0"}  
mycol2.insert\_one(mydict)  
  
  
#创建算法库  
mydb2 = myclient["visualization"]  
  
#创建算法集合  
mycol6 = mydb2["VideoStreaming"]  
mydict={"name":"VideoStreaming0","time":0,"value":"0"}  
mycol6.insert\_one(mydict)  
  
  
#创建模型库  
mydb3 = myclient["optimization"]  
  
#创建模型集合  
mycol7 = mydb3["CollisionData"]  
mydict={"name":"CollisionData0","time":0,"value":"0"}  
mycol7.insert\_one(mydict)  
  
  
  
  
  
#创建算法库  
mydb4 = myclient["algorithm"]  
  
#创建算法集合  
mycol6 = mydb4["deeplearning"]  
mydict={"name":"deeplearning0","time":0,"value":"0"}  
mycol6.insert\_one(mydict)  
  
#创建模型库  
mydb5 = myclient["model"]  
  
#创建模型集合  
mycol7 = mydb5["transfer"]  
mydict={"name":"transfer0","time":0,"value":"0"}  
mycol7.insert\_one(mydict)  
  
#创建知识库  
mydb6 = myclient["knowledge"]  
  
#创建模型集合  
mycol8 = mydb6["target\_recongnition\_model"]  
mydict={"name":"model0","time":0,"value":"0"}  
mycol8.insert\_one(mydict)  
  
#创建知识图谱集合  
mycol9 = mydb6["knowledge\_graph"]  
mydict={"name":"graph0","time":0,"value":"0"}  
mycol9.insert\_one(mydict)

## 数据匹配搜索

如下图所示，该部分代码主要涉及数据匹配搜索。

数据匹配搜索的实例如下图所示。



代码如下：

import pymongo  
from pymongo import MongoClient  
import re  
  
  
# 创建连接MongoDB数据库的Client  
myclient = pymongo.MongoClient("mongodb://localhost:27017/")  
# 建立数据库  
  
  
path=0  
#keyW='2021\_07\_31\_04\_42\_05'  
keyW='20210804\_141445\_9'  
  
  
  
if path==0:  
 mydb1 = myclient["database"]  
 #只有string才能查询，double不行  
 for u in mydb1.ArmJointPosition.find({'time': re.compile(keyW)}):  
 print(u)  
  
if path==1:  
 mydb1 = myclient["database"]  
 #只有string才能查询，double不行  
 for u in mydb1.ManipulatorMapping.find({'time': re.compile(keyW)}):  
 print(u)  
  
if path==2:  
 mydb1 = myclient["optimization"]  
 #只有string才能查询，double不行  
 for u in mydb1.CollisionData.find({'time': re.compile(keyW)}):  
 print(u)  
  
if path==3:  
 mydb1 = myclient["visualization"]  
 #只有string才能查询，double不行  
 for u in mydb1.VideoStreaming.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
  
  
  
if path==4:  
  
  
 mydb1 = myclient["database"]  
 # 只有string才能查询，double不行  
 for u in mydb1.ArmJointPosition.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
 mydb1 = myclient["database"]  
 # 只有string才能查询，double不行  
 for u in mydb1.ManipulatorMapping.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
 mydb1 = myclient["optimization"]  
 # 只有string才能查询，double不行  
 for u in mydb1.CollisionData.find({'time': re.compile(keyW)}):  
 print(u)  
  
  
 mydb1 = myclient["visualization"]  
 #只有string才能查询，double不行  
 for u in mydb1.VideoStreaming.find({'time': re.compile(keyW)}):  
 print(u)