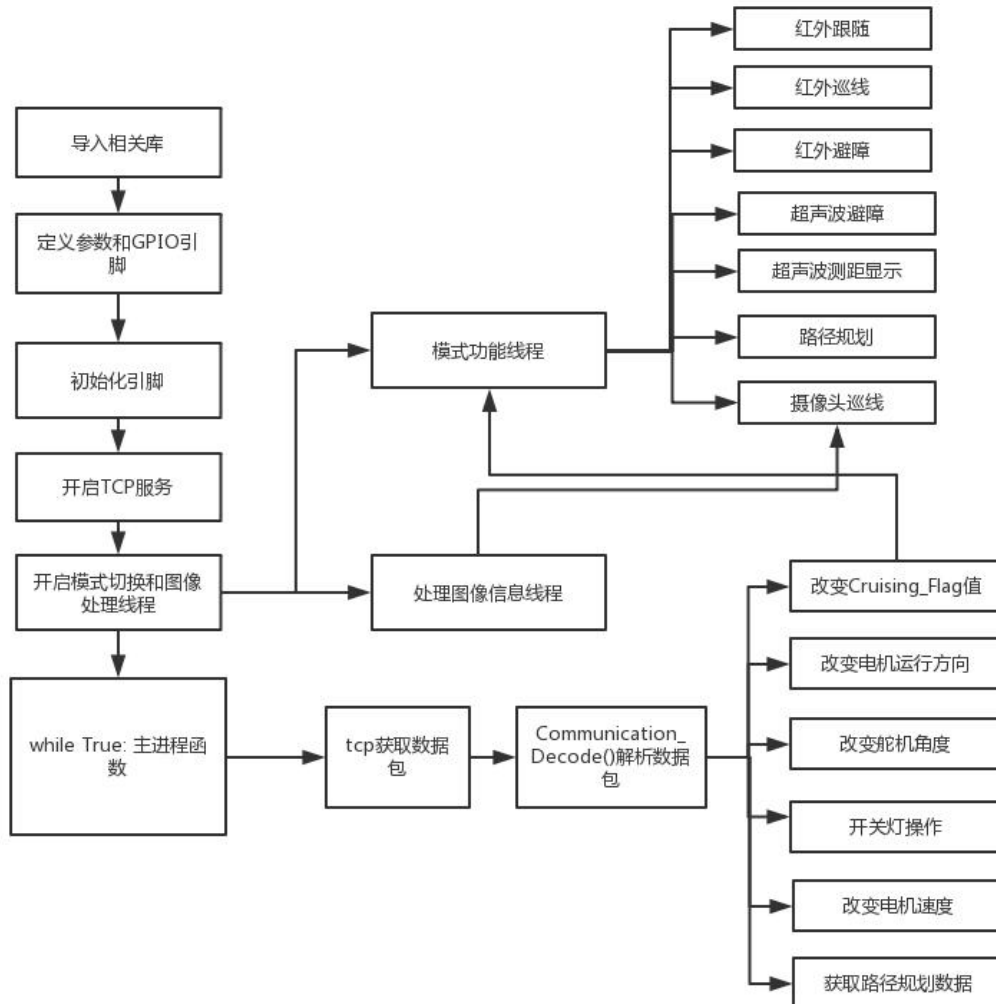


树莓派下位机源码解析

一. 树莓派下位机源码流程图



二. 树莓派所需要使用的库

```

import os    导入系统命令模块
    例: print 'hello '
from socket import * 导入 socket 模块
    例: tcpSerSock=socket(AF_INET,SOCK_STREAM)
from time import ctime 导入字符串日期模块
    例: print ctime()
        Thu Dec 15 15:00:40 2016
import binascii    导入二进制编码转换模块
    例: s = 'hello'
        b = b2a_hex(s)
        print b
    
```

68656c6c6f

```
import RPi.GPIO as GPIO  导入树莓派 python 的 GPIO 库
    例: GPIO.setup(LED0,GPIO.OUT,initial=GPIO.HIGH)
import time  导入时间库
    例: time.sleep(1)
import threading  导入线程模块
    例: threads = []
        t1 = threading.Thread(target=Cruising_Mod,args=(u'模式切换',))
        threads.append(t1)
from smbus import SMBus  导入舵机驱动模块
    例: XRservo.XiaoRGEEK_SetServo(0x01,90)
import cv2  导入 OpenCV-Python 库
    例: cap = cv2.VideoCapture(0)  使用 opencv 打开摄像头
import numpy as np  引入数组结构
from subprocess import call  导入子进程模块
    例: call("sh start_mjpg_streamer.sh &",shell=True)
```

注: python 引入模块时 import 与 from ... import 的区别

import datetime 是引入整个 datetime 包, 如果使用 datetime 包中的 datetime 类, 需要加上模块名的限定。

```
import datetime
print datetime.datetime.now()
```

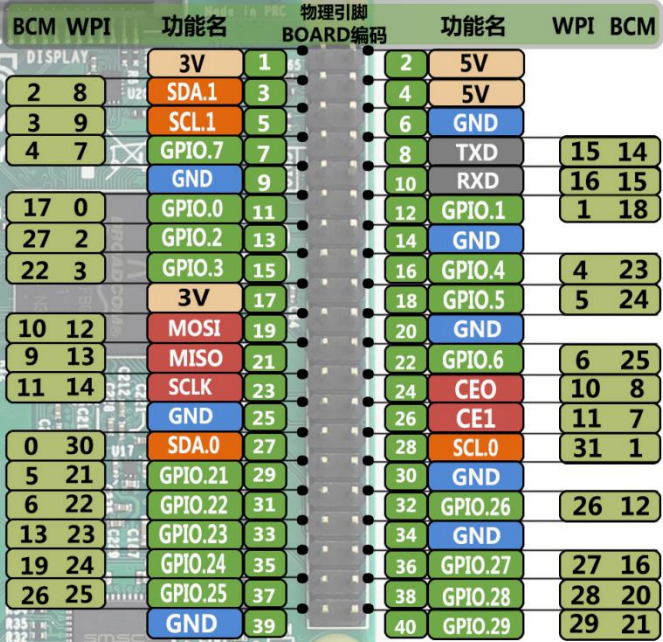
如果不加模块名限定会出现错误: TypeError: 'module' object is not callable \ AttributeError: 'module' object has no attribute 'now'

from datetime import datetime 是只引入 datetime 包里的 datetime 类, 在使用时无需添加模块名的限定。

```
from datetime import datetime
print datetime.now()
```

总结: Python 导入模块的方法有两种: import module 和 from module import, 区别是前者所有导入的东西使用时需加上模块名的限定, 而后者不需要。

三. 树莓派 python GPIO



也可以用 `putty` 登入树莓派系统后，输入 `gpio readall` 命令查看对应的 GPIO 对照表

```
pi@raspberrypi:~$ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      |      | 3.3v     |      |   | 1 | 2 |      | 5v       |      |      | |
| 2 | 8 | SDA.1    | ALT0 | 1 | 3 | 4 |      | 5v       |      |      |
| 3 | 9 | SCL.1    | ALT0 | 1 | 5 | 6 |      | 0v       |      |      |
| 4 | 7 | GPIO. 7  | IN   | 0 | 7 | 8 | 1 | ALT5 | TxD    | 15 | 14 |
|      |      | 0v       |      |   | 9 | 10 | 1 | ALT5 | RxD    | 16 | 15 |
| 17 | 0 | GPIO. 0  | OUT  | 0 | 11 | 12 | 1 | IN   | GPIO. 1 | 1  | 18 |
| 27 | 2 | GPIO. 2  | IN   | 1 | 13 | 14 |      | 0v       |      |      |
| 22 | 3 | GPIO. 3  | IN   | 1 | 15 | 16 | 1 | IN   | GPIO. 4 | 4  | 23 |
|      |      | 3.3v     |      |   | 17 | 18 | 1 | IN   | GPIO. 5 | 5  | 24 |
| 10 | 12 | MOSI     | OUT  | 1 | 19 | 20 |      | 0v       |      |      |
| 9  | 13 | MISO     | OUT  | 1 | 21 | 22 | 1 | OUT  | GPIO. 6 | 6  | 25 |
| 11 | 14 | SCLK     | ALT0 | 0 | 23 | 24 | 0 | OUT  | CE0     | 10 | 8  |
|      |      | 0v       |      |   | 25 | 26 | 1 | OUT  | CE1     | 11 | 7  |
| 0  | 30 | SDA.0    | IN   | 1 | 27 | 28 | 1 | IN   | SCL.0   | 31 | 1  |
| 5  | 21 | GPIO.21  | IN   | 1 | 29 | 30 |      | 0v       |      |      |
| 6  | 22 | GPIO.22  | IN   | 1 | 31 | 32 | 0 | IN   | GPIO.26 | 26 | 12 |
| 13 | 23 | GPIO.23  | OUT  | 1 | 33 | 34 |      | 0v       |      |      |
| 19 | 24 | GPIO.24  | OUT  | 0 | 35 | 36 | 0 | OUT  | GPIO.27 | 27 | 16 |
| 26 | 25 | GPIO.25  | OUT  | 0 | 37 | 38 | 1 | OUT  | GPIO.28 | 28 | 20 |
|      |      | 0v       |      |   | 39 | 40 | 0 | OUT  | GPIO.29 | 29 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
pi@raspberrypi:~$
```

引脚定义和相关参数定义

```
22  XRservo = SMBus(1)
23  print '...WIFIROBOTS START!!...'
24  global Path_Dect_px
25  Path_Dect_px = 320
26  global Path_Dect_on
27  Path_Dect_on = 0
28
29  #####
30  #####信号引脚定义#####
31  #####
32  GPIO.setmode(GPIO.BCM)
33
34  #####LED口定义#####
35  LED0 = 10
36  LED1 = 9
37  LED2 = 25
38
39  #####电机驱动接口定义#####
40  ENA = 13  #L298使能A
41  ENB = 20  #L298使能B
42  IN1 = 19  #电机接口1
43  IN2 = 16  #电机接口2
44  IN3 = 21  #电机接口3
45  IN4 = 26  #电机接口4
46
47  #####舵机接口定义#####
48
49  #####超声波接口定义#####
50  ECHO = 4  #超声波接收脚位
51  TRIG = 17 #超声波发射脚位
52
53  #####红外传感器接口定义#####
54  IR_R = 18  #小车右侧巡线红外
55  IR_L = 27  #小车左侧巡线红外
56  IR_M = 22  #小车中间避障红外
57  IRF_R = 23 #小车跟随右侧红外
58  IRF_L = 24 #小车跟随左侧红外
59  global Cruising_Flag
60  Cruising_Flag = 0  #当前循环模式
61  global Pre_Cruising_Flag
62  Pre_Cruising_Flag = 0  #预循环模式
63
64  global RevStatus
65  RevStatus = 0
66  global TurnAngle
67  TurnAngle=0;
68  global Golength
69  Golength=0
70  buffer = ['00','00','00','00','00','00']
71  global motor_flag
72  motor_flag=1
```


初始化引脚

```

81 *****
82 *****管脚类型设置及初始化*****
83 *****
84 GPIO.setwarnings(False)
85
86 *****led初始化为000*****
87 GPIO.setup(LED0,GPIO.OUT,initial=GPIO.HIGH)
88 GPIO.setup(LED1,GPIO.OUT,initial=GPIO.HIGH)
89 GPIO.setup(LED2,GPIO.OUT,initial=GPIO.HIGH)
90
91 *****电机初始化为LOW*****
92 GPIO.setup(ENA,GPIO.OUT,initial=GPIO.LOW)
93 ENA_pwm=GPIO.PWM(ENA,1000)
94 ENA_pwm.start(0)
95 ENA_pwm.ChangeDutyCycle(100)
96 GPIO.setup(IN1,GPIO.OUT,initial=GPIO.LOW)
97 GPIO.setup(IN2,GPIO.OUT,initial=GPIO.LOW)
98 GPIO.setup(ENB,GPIO.OUT,initial=GPIO.LOW)
99 ENB_pwm=GPIO.PWM(ENB,1000)
100 ENB_pwm.start(0)
101 ENB_pwm.ChangeDutyCycle(100)
102 GPIO.setup(IN3,GPIO.OUT,initial=GPIO.LOW)
103 GPIO.setup(IN4,GPIO.OUT,initial=GPIO.LOW)
104
105
106
107 *****红外初始化为输入，并内部拉高*****
108 GPIO.setup(IR_R,GPIO.IN,pull_up_down=GPIO.PUD_UP)
109 GPIO.setup(IR_L,GPIO.IN,pull_up_down=GPIO.PUD_UP)
110 GPIO.setup(IR_M,GPIO.IN,pull_up_down=GPIO.PUD_UP)
111 GPIO.setup(IRF_R,GPIO.IN,pull_up_down=GPIO.PUD_UP)
112 GPIO.setup(IRF_L,GPIO.IN,pull_up_down=GPIO.PUD_UP)
113
114
115
116 *****超声波模块管脚类型设置*****
117 GPIO.setup(TRIG,GPIO.OUT,initial=GPIO.LOW)#超声波模块发射端管脚设置trig
118 GPIO.setup(ECHO,GPIO.IN,pull_up_down=GPIO.PUD_UP)#超声波模块接收端管脚设置echo
119

```

四. 树莓派运行函数

Open_Light() 开灯

Close_Light() 关灯

init_light() 流水灯

Motor_Forward() 电机前进

Motor_Backward() 电机后退

Motor_TurnLeft() 电机左转

Motor_TurnRight() 电机右转

Motor_Stop() 电机停止

forward() 校准后的前进方向

back() 校准后的后退方向

left() 校准后的左转方向

right() 校准后的右转方向

ENA_Speed(EA_num) ENA 速度控制

ENB_Speed(EB_num) ENB 速度控制

Angle_cal(angle_from_protocol) 将字符型角度转换成 int 型角度并返回

SetServoAngle(ServoNum,angle_from_protocol) 设置舵机角度，例：SetServoAngle(1,'90')

设置舵机 1 号的角度为 90 度

Avoiding() 红外避障函数

TrackLine() 巡线函数

Follow() 红外跟随函数

Get_Distance() 超声波测距函数

Avoid_wave() 超声波避障函数
Route() 路径规划函数
Send_Distance() 向上位机发送超声波数据
Cruising_Mod(func) 模式切换函数
Path_Dect() 摄像头巡线函数
Path_Dect_img_processing() 摄像头获取地图信息
Communication_Decode() 数据包解码

while True: 主进程函数

```
while True:
    print 'waitting for connection...'
    tcpCliSock,addr=tcpSerSock.accept()
    print '...connected from:',addr
    while True:
        try:
            data=tcpCliSock.recv(BUFSIZ)
            data=binascii.b2a_hex(data)
        except:
            print "Error receiving:"
            break

        if not data:
            break
        if rec_flag==0:
            if data=='ff':
                buffer[:]=[]
                rec_flag=1
                i=0
            else:
                if data=='ff':
                    rec_flag=0
                    if i==3:
                        print 'Got data',str(buffer)[1:len(str(buffer))-1],"\r"
                        Communication_Decode();
                        i=0
                    else:
                        buffer.append(data)
                        i+=1
                #print(binascii.b2a_hex(data))
    tcpCliSock.close()
    Motor_Stop()
    tcpSerSock.close()
```



官 网: www.xiao-r.com

论 坛: www.wifi-robots.com

官方商城: wifi-robots.taobao.com

微信公众号:

