**(這只是整理前面做過的實驗內容 不4考古QQ)**

嵌入式實驗-各實驗內容

Lab08 Serial RPC

4.2 Remote Procedure Call (RPC)

在電腦上下指令，透過serial以字串形式傳給mbed

Enter鍵:顯示可用物件

"/<object> RPC" 顯示可對物件用的method

"<Object name>/<Method name> <Arguments >"

例: /myled/write 0 結果:呼叫mbed的 myled.write(0);

Mbed處理字串指令:

用for迴圈 pc.getc()直到讀到’\r’ 每次讀完存進buf

RPC::call(buf, outbuf); buf->指令 outbuf->傳回去的字串

4.3 用python下指令取代在電腦上打指令做RPC

s.write(bytes("/myled/write 1\r", 'UTF-8')) 來下指令

4.4 自訂RPCfucntion

void LEDControl(Arguments \*in, Reply \*out);

RPCFunction rpcLED(&LEDControl, "LEDControl");

在main裡讀完buf後一樣RPC::call會進到RPCFunction 再進行指令的轉換

變成RPC有的指令後 存成buffer再RPC::call一次

這題一樣用pythom傳指令

4.5 python叫LCD顯示字串(display、locate)

4.6 python 下指令 取得mbed三軸加速計的address跟測值

\*在下指令的地方的method名稱就對應到main裡面的function名稱

Lab09 Wireless Communication - XBee

4.3 Xbee set up 用linux的screen執行AT指令

4.4 Xbee 互傳:master將訊息打在screen上 用pc.getc讀進來 xbee.printf傳給

slave slave用xbee.getc來接收

\*master跟slave的MY & DL addresses 互相相反(在main裡面設定)

4.5 由python 透過ttyUSB0 用s.write傳送訊息給master Xbee 再由與k66f連

接的slave xbee接收(xbee.getc) 又用pc.printf傳給電腦，python再用

line = s.read(5) 把訊息讀進來 最後print出來

\*一樣MY跟DL address設定要互相相反

4.6 用xbee傳送RPC指令

4.7 python作為廣播的host 輸入對應的字母來選擇要接收訊息的xbee xbee端

(接K66F)會判斷host送來的字母是不是選他

(receive\_data = xbee.getc();if (receive\_data == 'a')) 是的話就回傳訊息

python用s.read來接收並print

\*MY:隨意,DL:0x0,PAN ID: python及所有xbee(main)皆須相同(例:0x1234)

Lab10 WiFi and MQTT

4.4 ESP8266 set up

4.5 Connect ESP8266 to WiFi

4.6 set up MQTT

MQTT (Message Queue Telemetry Transport) :way for internet-connected

devices to send messages by publishing data and subscribe to topics.

\*在python跟main裡都要設定wifi的ip位址

sprintf(buff, "QoS0 Hello, Python! #%d", message\_num); 傳送訊息

python訂閱”Mbed” topic (main跟python都要設定)

# Loop forever, receiving messages

mqttc.loop\_forever()

Lab11 Servos, encoder and ping

4.3 設speed=0 校正continuous servo

4.4 test standard servo

4.5 encoder: count the number of times the light is detected through slits

每次encoder接收到光會傳訊號(DigitalIn)進來，用ticker格0.01秒後執

行函式把step加1

用last和value做flip-flop(只有從遮住到沒遮的edge會加)

4.6 ping送出5us的波 計算回音的時間以測量距離

\*只能測正前方且距離2公分以上的物體

4.7用encoder測輪子(馬達)在不同轉速下車子的實際移動速度(cm/s)

pc.printf 然後由python接收 做成關係圖

4.8用python算出想要的車子速度(cm/s)所需的speed值 再丟進main裡面使用

Lab12 BOE BOT Car

4.4 class BBCar 的功能:

void BBCar::stop()

servo0.set\_speed(0);servo1.set\_speed(0);servo0.set\_factor(1);servo1.set\_factor(1);

void BBCar::goStraight( int speed )

servo0.set\_speed(speed); servo1.set\_speed(-speed);

/\* factor: control the speed value with 0~1 control left/right turn with +/- \*/

void BBCar::turn( int speed, double factor )

servo0.set\_speed(speed); servo1.set\_speed(-speed);

if(factor>0){ servo0.set\_factor(factor); servo1.set\_factor(1); }

else if(factor<0){ servo0.set\_factor(1); servo1.set\_factor(-factor); }

使用:在main宣告Ticker servo\_ticker; PwmOut pin9(D9), pin8(D8);

BBCar car(pin8, pin9, servo\_ticker);

4.5 用encoder.get\_cm來控制車車走一段固定的距離

4.6 把用lab11建立的兩個馬達分別的pwm與速度table告訴car.setCalibTable做校正

4.7 在ping測到物體距離<10之前持續前進 然後car.stop

4.8 遙控車:

用rpc控制: #include "bbcar\_rpc.h"(12\_5) \*螢幕要用/dev/ttyACM0/(由pc傳送)

用xbee傳送跟接收(12\_6) \*螢幕要用/dev/ttyUSB0/ (由master xbee傳送)

用python判斷輸入的按鍵，s.write對應的指令(12\_8)

4.9 順/逆時針旋轉到指定的方位角(用三軸加速計判斷)

Lab13 Machine Learning on Microcontroller

5.3 用webcam拍照

5.4 照片轉成cifar10 dataset

5.5 machine training

5.6 mbed猜測圖片分類(category)