

ESLab Group2 Final Project Report

B07901022 吳沛林

B07901086 俞建璇

B07901087 蔡沛洹

Motivation

我們在看其他人做嵌入式系統實驗時做了不少遊戲，他們往往結合STM32的感測器功能，並且增加遊戲體驗，而且做得也十分有趣。在組內討論後，我們覺得既然要做，就做一個Universal Joy Stick來玩各式各樣的遊戲。如同市售遊戲中，有許多遊戲都是使用搖桿操作，我們打算直接用STM32的板子做為操作遊戲系統之媒介，結合我們自己想到的一些操作方式(平移加速、旋轉)等等，做出一個屬於我們自己的遊戲搖桿。

Methodology

Buttons

按鈕按下及放開時，OUT那個PIN會輸出不同的電壓，一個HIGH一個LOW，使用GPIO HAL functions讀取各按鈕的輸出值。儲存前一刻的輸出值與現在的輸出值做比較，就可以判斷click和release兩個event，並用event觸發電腦端的HID。

將各按鈕對應到電腦中遊戲介面上的各功能鈕，就可以達成在搖桿上按下該按鈕，等於用滑鼠點下介面中那個功能鈕的效果。如果該遊戲對於某些鍵短按和長按有不同效果的話，也可以從click和release兩個event的時間差判斷長或短。

Joystick

蘑菇頭由雙軸的可變電阻組成，當頭朝向不同方位時，會改變兩軸上的電阻值，進而改變輸出的電壓 Vx 和 Vy，用AnalogIn來讀取兩個電壓，就可以讓頭的方位在電腦上呈現。

另外，蘑菇頭也可以做為按鈕使用，它有一個腳位為sw，當按下蘑菇頭，sw將與GND相連，放開後則會成斷路，所以只要將sw連到STM32上，並將該腳位設為PullUp，那就可以達成按下時讀到0，放開後讀到1的效果。

蘑菇頭主要用來操控游標，由STM32接收到的Vx, Vy判斷頭的方位，並移動游標。sw則對應滑鼠的左鍵，待游標移至目標按鈕時，按下蘑菇頭，觸發sw就相當於用滑鼠點擊該鈕。

Sensor

在sensor這邊，我們觀察了STM32的加速度儀與陀螺儀的輸出之後，覺得最簡單的linear transformation就能解決資料處理的問題，而且其overhead也不會過大，所以就以這個為基底去做我們這部分的feature。在處理資料時，我們觀察到gyro本身其實是蠻準確的(我們組上的兩塊板子皆是如此)，所以我們在實作上有結合gyro的行為做為acceleration的輔助。gyro和acceleration的實作上，我們都有使用threshold的概念，一方面是因為其實STM32的有雜訊(訊號會有微幅震動)，而這並不是linear transformation能處理的問題，用threshold就可以初步濾掉。第二個點是其實用filter可能會造成操作上的延遲，所以最後就用最簡易的方式，再加上對於搖桿而言，絕對值本身並不是那麼重要，所以取其數值並且轉成我們自己discrete的強度我

們覺得很適合搖桿的作法。最後就是在acceleration中我們發現加速度值會有回彈現象，所以這部分我們是直接判斷他在位移之後，會讓他維持在相關狀態幾百ms，之後再恢復，就可以減輕其現象。

在搖桿製作的時候，我們在規劃搖桿時認為應該要把板子反著擺接在板子，再加上其實我們一般在玩遊戲的時候我們並不是固定角度遊玩，所以我們有把校正過程自動化，使用者就在開始遊玩前幾秒維持想要的角度，之後就可以正常遊玩。這部分因為想多採樣，所以有做multithread去多做幾次，(目前是acc/gyro各兩次)，最後就是試著結合EventQueue和thread讓update_sensor的功能可以獨立跑。但是其實我們主程式就表現不錯了，所以目前是沒有將其整合到我們的主程式上。不過未來如果真的有performance需求想要加入也是可以的。

HID

為了能夠支援盡量多的裝置、避免需要安裝額外的驅動程式、加快連線的速度，我們決定使用成熟且受到廣泛支援的HID(Human Interface Device)協定讓STM32板子與電腦進行溝通。

HID 協定適合實作人機互動介面例如滑鼠、鍵盤或我們這次所要做的遊戲搖桿。比較特別的是，我們這次主要目標是以滑鼠操作為主的小遊戲，所以我們雖然外觀是遊戲搖桿，但都在模擬滑鼠的信號與運作。

HID最先在USB介面上被提出。因為HID非常實用以及開發跟相容性的考量，所以在藍芽發展時，藍芽上的HID協定幾乎沿用原本USB的HID規範。在USB上的HID裝置(Device)透過Report方式與主機(Host)溝通，而在BLE上的HID裝置(Peripheral)則利用GATT來與主機(Central)傳輸Report。

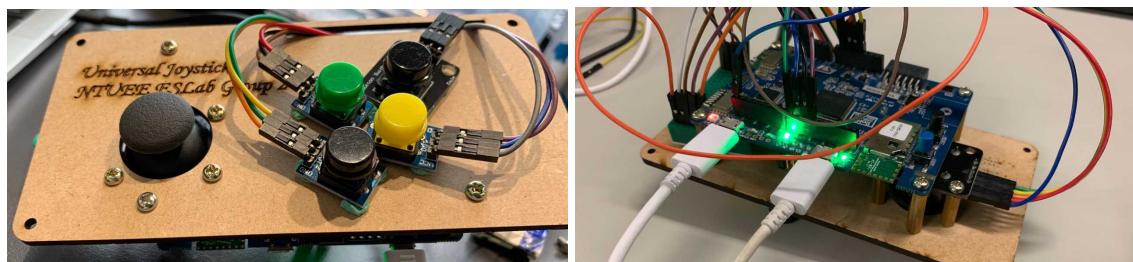
Main Program

主要就是將STM32板收到的buttons、joystick和sensor信號與HID結合，模擬滑鼠的運作。此外，在joystick和buttons讀取上是直接用busy waiting的方式，最大的降低延遲性。在程式中我們也有參考蘑菇頭Analog訊號，將相關操作變為較為類比的方式，也就是隨著Vx、Vy與 $\frac{1}{2}$ VCC之間差距的大小調整游標移動的速率。

Assembly of the Device

使用密迪板作為搖桿的底板，將蘑菇頭和按鈕等使用者介面固定在板子正面；板子背面則放STM32開發板及麵包板。

用AutoCAD設計底板形狀及其上面的開孔，讓STM32和蘑菇頭可以用螺絲鎖在上面，至於麵包板及按鈕，因為沒有鎖孔，所以用黏土固定在板子上。



Experiment and Conclusion

Buttons and Joystick

按鈕及蘑菇頭的測試很單純，就是測量電壓是否輸出正常。將VCC和GND分別接到STM32的5V和GND後，將OUT接到三用電表測量電壓。

發現大顆的黑按鈕沒有按下去時輸出5V，按下去後輸出0V，另外3個按鈕則相反。延遲很小，大部分情況，在按下去或放開之後1秒內就可以到達目的電壓，中間有時會短暫出現介於2~2.5V之間的中繼電壓，不過非常短暫，並不影響，而且2V在digital讀出來也是1，所以digital 0 -> 1，也就是大黑鈕的release和其他鈕的click幾乎完全沒延遲；而digital 1 -> 0，也就是大黑鈕的click和其他鈕的release會稍有一點點延遲。

蘑菇頭的部分，當頭在正中央時，Vx, Vy都是½ VCC。頭在X, Y軸上往不同方向傾斜時，會分別造成Vx, Vy的改變，往一個方向是增大，最大增到VCC；往另一方向則減小，最小減至GND，幾乎沒有延遲。由於AnalogIn讀到的電壓值是該電壓與參考值的比值，而參考值預設是3.3V，再加上麵包板孔位數量限制，所以為了方便，就把蘑菇頭的VCC改接STM32的3V3，如此只要將AnalogIn讀值與0.5的比大小就可以判斷頭方位。

Sensor

sensor部分的測試基本上都是用肉眼去看數值，這邊補充上述methodology沒講述的觀察：

1. 當板子穩定放置時，板子的acc, gyro其實不太會變(就是小雜訊)，不同板子絕對數值會略有差異，但是移動的相對數值其實差別不大 → 校正時去調整original point
2. acceleration在移動前其實有時候也會有一瞬間向相反方向移動，但是這個情形在加入threshold之後減低不少，所以並沒有特別去處理，否則應該就還是要開一個window size觀察其那一小段時間的表現，再去處理會比較準。

還有一些沒有進搖桿，但是有稍微試過的feature：

1. 一開始我們有想過特殊動作，例如走路等等，雖然有做出雛形(就是可能偵測搖桿左右規律搖動一段時間)，然而因為我們的遊戲沒有用到，所以之後就沒有繼續做到主程式中。
2. magneto sensor方面原本有想過用地磁來做出搖桿的指向定位，但是可能是實驗室電子儀器較多，做出來發現其實數值相當的不規律，再加上我們想到的遊戲也沒有用到搖桿絕對指向的功能，因此停在實驗階段。

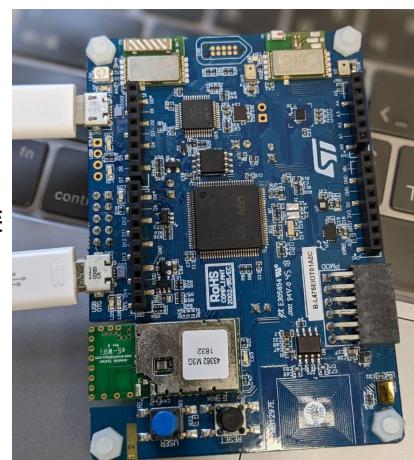
另外是在遊戲時的極端角度，我們自己遊玩的經驗是即使把搖桿偏到80幾度也不太會影響操作，雖90度可能會有極端數值，但正常操作應該都是可接受的(如果想要處理可以也是開一個window size去觀察數值有沒有持續性極大的情況，再做額外處理)

最後還有之前其實有在想操作上應該是要用相對的還是絕對的，我們是不是也可以用積分等等，但做過實驗之後發現可能數值上還是會有誤差(例如說搖桿左轉右轉後不會回到原點)，如果直接使用絕對數值誤差會漸漸累積，所以之後也就沒有做這方面。

HID

USB

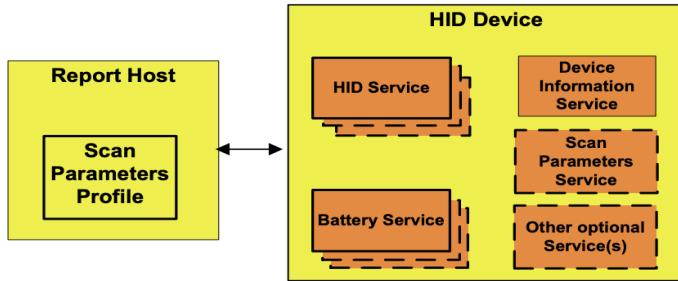
在mbed OS上開發USB的HID裝置相當方便簡易。只需要安裝好External USB(例如VDD、D+、D-、GND四個訊號接入USB頭，或是板子上現成的USB OTG接口)，並引入mbed OS內建



之USBHID 相關標頭檔(例如USBMouse.h)後，即可建立HID物件並呼叫他的物件函數來操作這個HID裝置。

BLE

在mbed OS上開發BLE的HID裝置較為麻煩。根據藍芽SIG的規範，BLE HID裝置須符合以下的架構：至少一個HID Service、至少一個Battery Service、剛好一個Device Information Service(且需有PnPID Characteristic)。在mbed OS中僅有提供Battery Service，故另兩個Service 需另外實作。



Future Work

我們在作品中是使用公用的protocol，所以在延伸的話其實手機電腦應該都可以使用，鍵盤如果將其登入成BLE service，那我們的搖桿應該也可以玩電腦的射擊遊戲等等，惟時間因素還有硬體限制(板子的BLE似乎故障)，造成implementation實務上有不少困難，我們也有把我們實作的prototype(HIDComposite.cpp)附在github中，期待之後有人能夠將功能做得更加完整。

Reference

GPIO HAL Functions:

https://os.mbed.com/docs/mbed-os/v6.15/mbed-os-api-doxy/group_hal_gpio.html?fbclid=IwAR3NGnyDXkkEJC6tS2GAN4uVe0X4yf1VDXjvP3cXPF4dC9s40kr6f7kIDJs

Mbed AnalogIn: <https://os.mbed.com/docs/mbed-os/v6.15/apis/i-o-apis.html>

USB HID: <https://os.mbed.com/docs/mbed-os/v6.15/apis/usbmouse.html>

BLE HID: <https://github.com/csash7/mbed-BLE-Mouse>