

嵌入式系統 期末報告

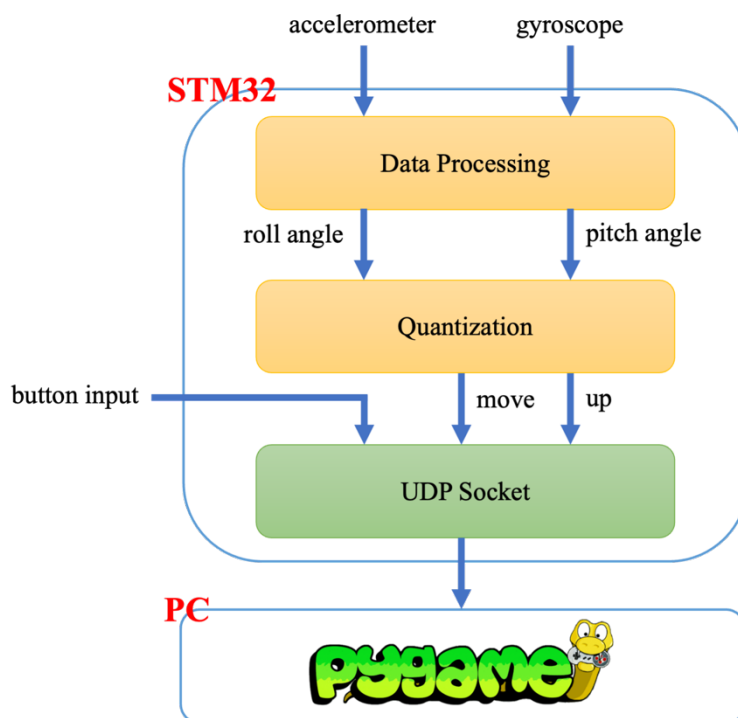
涂銘洋 毛楷維 古振宏

GitHub repo: <https://github.com/mingyang-tu/es-final>

動機

過去有許多組別嘗試用加速度計或陀螺儀進行手勢辨識，但往往因為方法單純，導致精確度不高。因此，我們希望能融合加速度計和陀螺儀的信號來實現更準確的手勢辨識，同時藉此熟悉上課所教的嵌入式系統開發技術。

架構圖



作法

1. Data processing

使用 complementary filter [1] 將加速度計和陀螺儀的資料轉換成 pitch angle（前後旋轉）及 roll angle（左右旋轉）。公式如下（以左右旋轉為例）

$$angle_{t+\Delta t} = 0.98 \times (angle_t + gyroData \times \Delta t) + 0.02 \times accData$$

$$gyroData = \frac{gyro_y}{1000}, \quad accData = \tan^{-1} \frac{acce_x}{acce_z}, \quad \Delta t = 0.025$$

註：*gyroData* 會扣除校正的角速度值，*accData* 也會扣除校正的角度。

校正：STM32 剛啟動時，會以 2ms 為間隔取樣 2000 個點，取其平均作為校正值。

2. Quantization

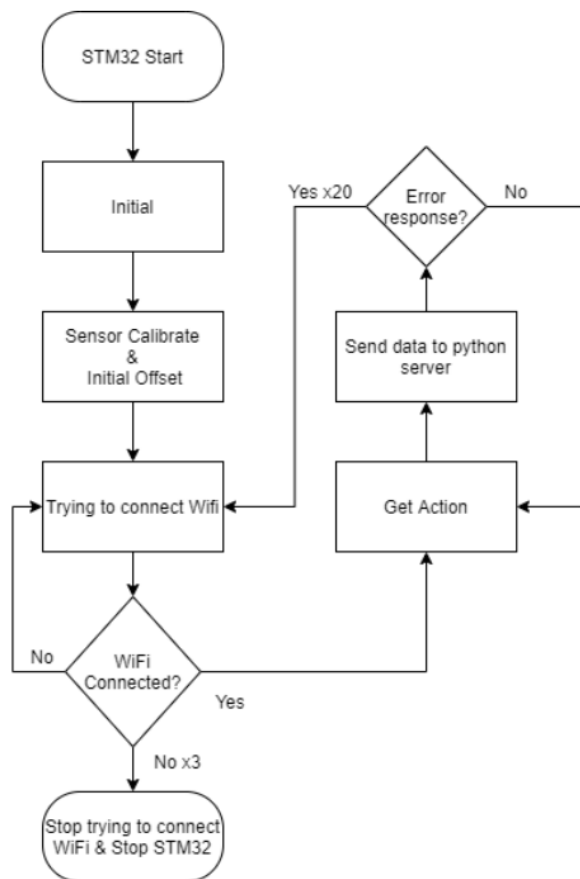
- Pitch angle (前後旋轉)：以 15° 為界，大於 15° 設為 1，小於則設為 0。
- Roll angle (左右旋轉)：以 4° 為間隔，把左右旋轉角度分成 21 個區間 ($-10 \sim +10$)。

$$q = \begin{cases} -10, & \theta \leq -40 \\ \text{sign}(\theta) \left\lfloor \frac{|\theta|}{4} \right\rfloor, & -40 < \theta < 40 \\ +10, & \theta \geq 40 \end{cases}$$

3. WIFI module

- 因應 doodle jump 此類在乎即時操作的跑酷遊戲，我們使用延遲較低的 UDP socket。
- 傳輸資料：以 json 格式傳送
 - move：介於 $-10 \sim +10$ ，代表左右移動的幅度。
 - up：0 或 1，在選單介面會將選項下移一格；在遊戲中會發射子彈。
 - enter：0 或 1，在選單介面會選擇當下選項；在遊戲中會暫停遊戲。
- 重連機制：如果 `sendto()` 回傳 -1 ，代表資料的傳送有發生錯誤，重複 20 次則嘗試重新連接。

▼ WIFI module 架構圖



4. Pygame

Two threads

- Main thread (T1)：遊戲渲染、碰撞檢測等。
- Another thread (T2)：接收資料並更新使用者輸入。

兩者的互動關係

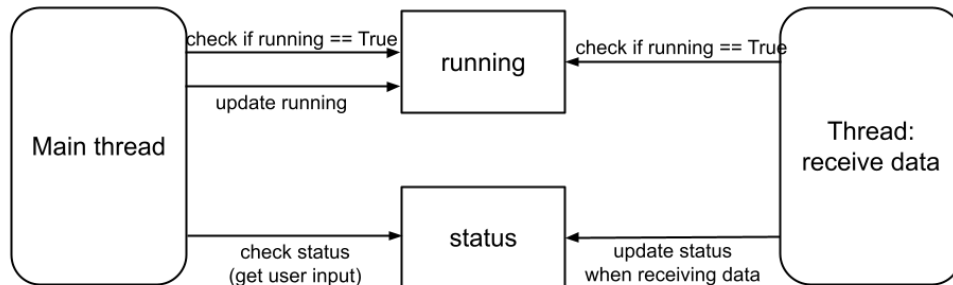
- 更新使用者輸入

T2 會在每個迴圈更新 `status` 的值，T1 則會在每個迴圈查看 `status` 的值並做出相對應的動作。
- 判斷是否連接

如果 T2 超過 1 秒沒接收到資料，會判斷沒有成功連接，將 `status["connected"]` 設為 `False`。此時 T1 會在畫面顯示紅色的 `disconnected` 字樣，若斷線時正在遊戲中，遊戲會自動暫停。反之，如果 T2 有接收到資料，會將 `status["connected"]` 設為 `True`。此時 T1 會顯示綠色的 `connected` 字樣。
- 使用者關閉遊戲

使用者可以從選項或角落的 X 關閉遊戲。此時 T1 會把 running 設為 False，導致 T2 的 while 迴圈終止並關閉 server，T1 則會在 T2 停止後才關閉程式。

▼ 兩個 thread 的互動關係圖



成果

1. 更加精準的手勢辨識及左右移動類比化。
2. 斷線後自動連接，並在前端做出相對應的提示。
3. 功能完整的 pygame 遊戲。

參考資料

- [1] Colton S, Mentor FRC. The balance filter. Presentation, Massachusetts Institute of Technology. 2007;
- [2] Adams VH. Complementary filters:
https://vanhunteradams.com/Pico/ReactionWheel/Complementary_Filters.html
- [3] NTUEE-ESLab/2021-pikachu_volleyball: https://github.com/NTUEE-ESLab/2021-pikachu_volleyball

Demo 影片

https://www.youtube.com/watch?v=z_7Y-ap9ra4