

API:

```
void PrintDistance(double raw_distance)
```

用來協助印出距離

適用情況:

- (1) 作為超聲波與系統的檢測
- (2) 協助調參時判斷

已經寫在 `HAL_TIM_IC_CaptureCallback` 中，每一個 `htim→Instance` 下皆有一組已經註解起來的 `PrintDistance(distance)`;

將註解消除，經過 `build` → `debug as` → 按下上方的開始鍵

(確保 `serial port terminal` 有連接，每次重接上 `usb` 後皆要重新連線)

就可以在 `serial port terminal` 看到印出距離。

- 注意：一次測試只能印出一個方向的距離(一次只能反註解一個)
因為共用所有的數字皆共用一組 `char array` 和 `usart`，會互相干擾。

```
void DebugLED(int condition)
```

控制 LED 燈，`condition` 為 1 是亮，0 為暗，其他數字不會有反應。

適用情況:

- (1) 作為狀態標示，在試跑時可知道正在執行程式碼哪一段
- (2) 協助除錯

- 注意：記得要關燈 `DebugLED(0)`;

```
void MotorSet(int LeftThrust, int RightThrust)
```

簡化程式碼，在控制馬達時只需輸入左右馬力百分比。

兩個參數的輸入範圍皆為 `100~-100`，輸入範圍外的值默認馬力為 0

適用情況:

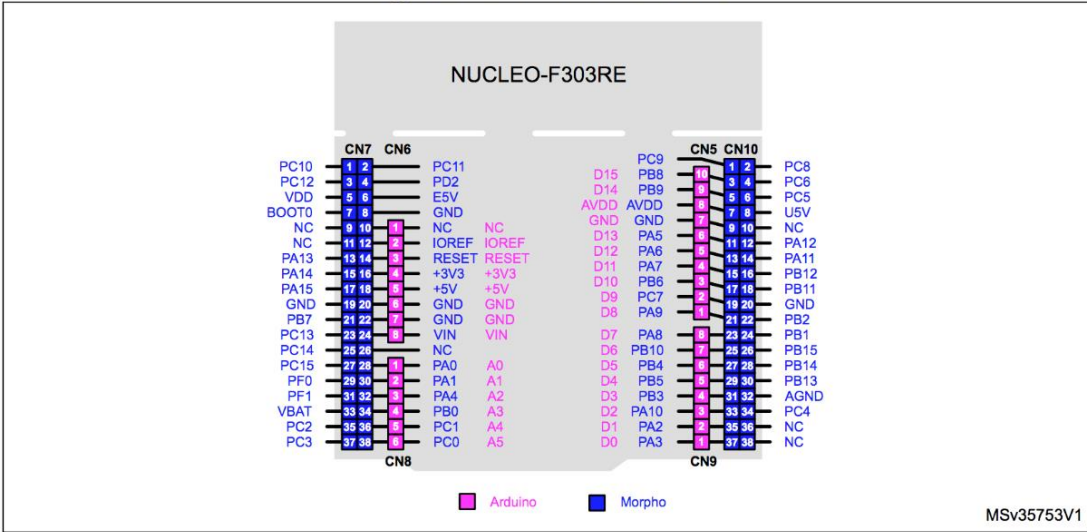
- (1) 馬達控制

- 範例:

```
MotorSet(100,100)    // 左右輪全速前進
MotorSet(0,0)        // 左右輪停止
MotorSet(-100,-100)  // 左右輪全速後退
MotorSet(-50,50)     // 左輪半速倒退，右輪半速前進(原地左轉)
```

腳位圖

Figure 16. NUCLEO-F303RE



	綁定 Timer	5V/Gnd	Trigger	Echo
前方超聲波	Timer 2	5V/GND 集中供電	PA1	PA0
左方超聲波	Timer 3			PA6
右方超聲波	Timer 15			PB14
左後超聲波	Timer 16			PA12
右後超聲波	Timer 4			PA11

硬體檢測：

上電前

- (一) 外部接線是否有脫落，對照上方腳位圖
驅動版是否接對(接在背面，四個 motor 接點朝下(stm32 LD1 為上))
- (二) 電池需 1.4V 以上，才能保證效能

上電後

- (三) 不定時檢查是否有發燙(電線、感測器)、燒焦味
- (四) 執行超聲波感測器檢測，依次註解消除 PrintDistance(distance)
是否走正確印出距離，每個超聲波感測器執行一次

編譯與燒錄

- (五) 在 USB 供電、電池供電切換前
下電 → E5V、U5V 切換 → 上電
- (六) 按下 debug as 後，是否進入 debug 頁面，LD1 是否有綠紅綠紅切換