

團隊資訊

▼ 2021 MakeNTU

主題：會奔跑的智能鬧鐘

編號：Team 10

隊名：搞毛啊今年怎麼沒有吃到飽

組員：臺大電機大一 江秉城
臺大電機大一 巫竝儒
臺大電機大一 謝承修



會奔跑的智能鬧鐘

▼ 創作理念

如果今天可以不用出門上班，你會幾點起床？
不小心睡過頭後，你一定會感覺到很懊悔吧？

居家工作多了些自由，但少了必須回到工作崗位的壓力。對員工而言，非常容易...被人性所困(睏?!)因此我們希望開發出一套除了死人，誰都叫得醒的鬧鐘！

目標是讓使用者不會因為一時糊塗或反射性按掉鬧鐘，而錯過早晨的黃金工作時間。如此一來，不僅在起床的時候增添許多樂趣及工作產能，也能避免自己鑄下睡過頭的大錯後才來懊悔喔！

▼ 功能與特色

身為鬧鐘，就要有鬧鐘的基本功能：在設定的時間響起。

但不同於一般鬧鐘只會響，它還會開始跑！讓使用者無法隨便伸手就讓他安靜，繼續他的懶惰計畫～但如果你以為玩你跑我追就可以讓使用者清醒，那肯定是低估了人類的惰性。

有鑑於此，我們引入**STM32 X-CUBE-AI**套件以設計清醒測試。使用者抓到奔跑鬧鐘後，還必須在螢幕上寫特定字（預設"WAKEUP"）。如果寫得工整又正確，鬧鐘才會認為你真的醒了然後安靜。

接著鬧鐘會向你**播報天氣**，即使你今天不用出門上班～並且會顯示你下一個重要的行程，作為**備忘錄**的功能。

▼ 更新板demo影片

比賽當天的demo影片並不完整，因此我們重新附上影片。

1. <https://youtu.be/oLRXym1LWJc>

2. <https://youtu.be/qUD686MVLz8>

從第一個影片中可以看到，當時間來到九點的時候（起床時間可自行設置），會自動撥放音樂以及打開馬達（不放在地上跑是因為方便拍攝），提醒你該起床了。

接著，你必須在觸控面板上工整地寫下"**WAKEUP**"六個字(如果寫太醜的話會要求重寫)，寫完之後會自動關閉馬達與音樂。而此時，可以觀察到LCD螢幕交替顯示下個行程與當天天氣。其中，工作的備忘錄是提前寫好的，而天氣的播報則是由Python上網爬蟲後將資訊用藍芽的方式回傳到開發版上進而顯示出來的。

第二個影片是在DEMO車子行走的功能。影片中可以看到，當起床的時間一到，車子便會開始在地上「奔跑」。奔跑的目的是為了讓人要起身抓住它，所以車子會**不規則的橫衝直撞**。但同時，車子前方的超聲波模組可以幫忙探知前方是否有障礙物，避免車身受到撞擊。

▼ 使用技術

開發板：STM32F746G-DISCO、Arduino Nano、Arduino Uno

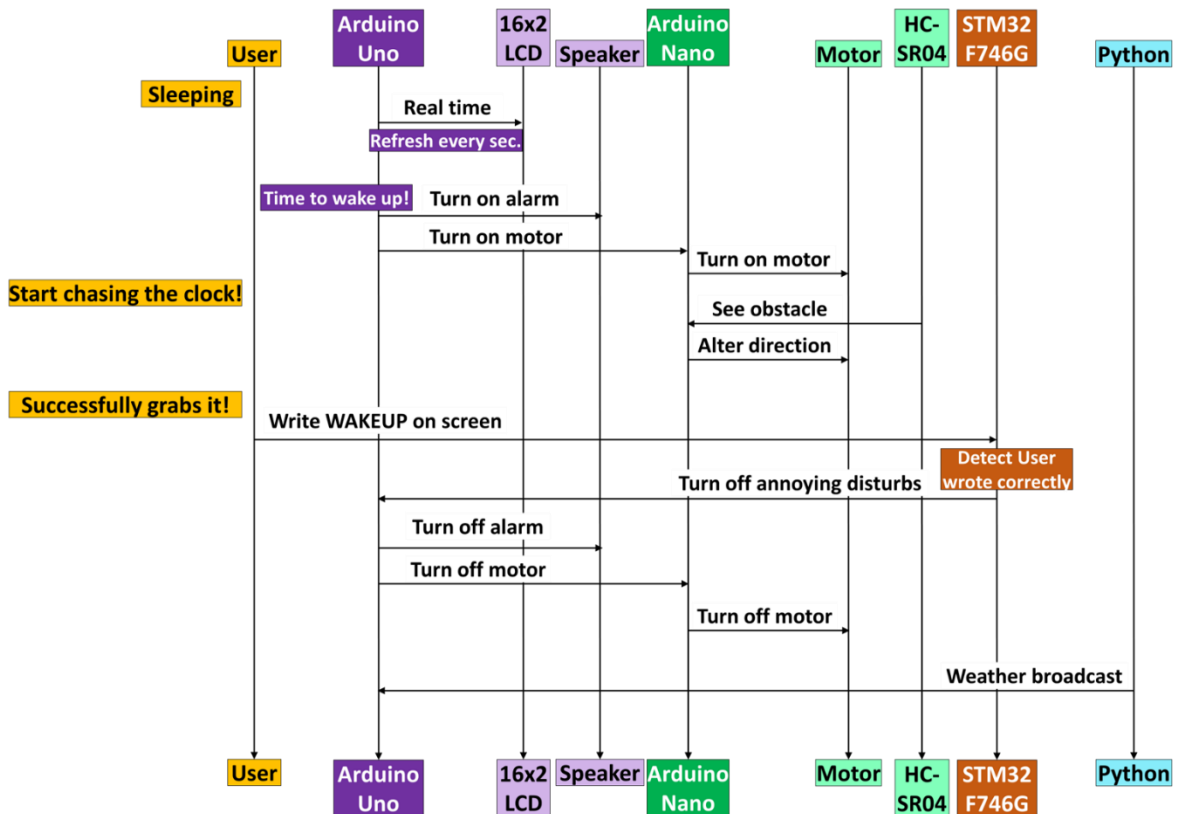
開發環境：STM32CubeIDE 1.2.0、Arduino IDE、Visual Studio Code

外接模組：HC-SR04 超音波測距模組、HC-05 藍芽模組、I2C 16x2 LCD 液晶螢幕、L298N 馬達控制模組、1W 8Ω揚聲器

車身設計：使用AutoCAD軟體進行繪圖，再利用雷射切割出車身。

▼ 系統方塊圖

STM32F746G : 觸控板寫字
 Arduino Uno : 掌控時間、接收天氣資訊
 Arduino Nano : 馬達控制與避障
 Python : 負責上網抓取天氣資訊並傳送

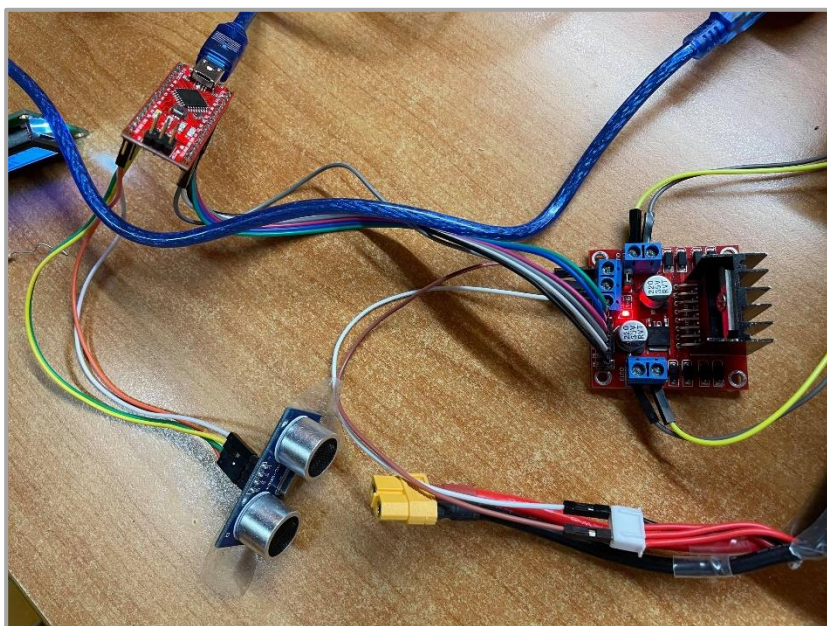


彼此之間的溝通（皆是透過Arduino Uno作為媒介）

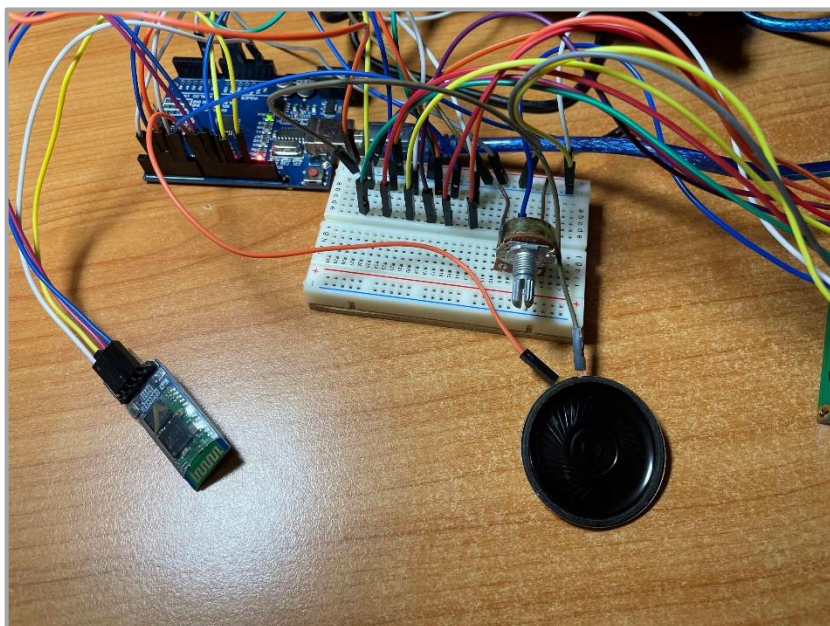
1. Arduino Uno的A0腳位連接STM32F746G的D10腳位
2. Arduino Uno的D6腳位連接Arduino Nano的A6腳位
3. Arduino Uno 上的藍芽連接終端（Python指令）

▼ 結構展示

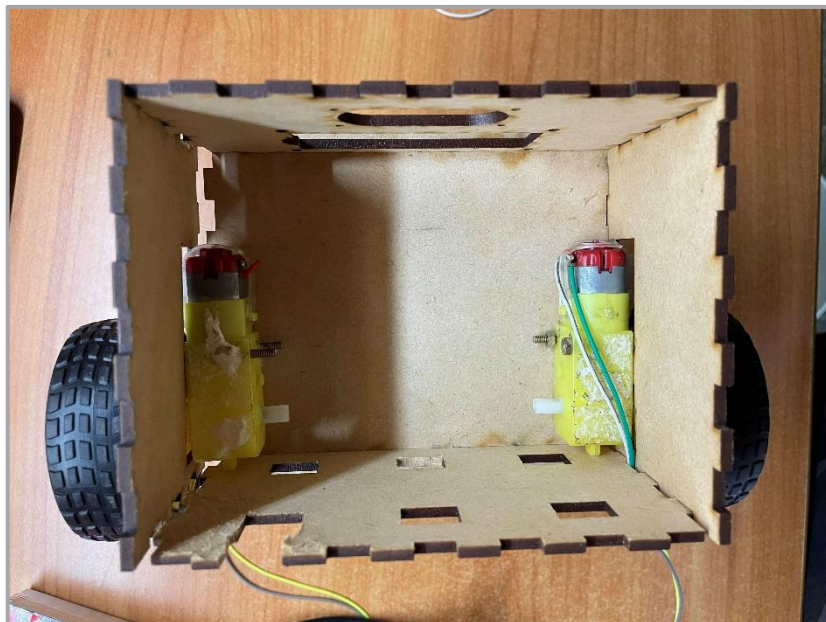
Arduino Nano 連接超音波測距模組與L298N馬達控制模組
(其中Arduino Nano 、 L298N需外接電源)



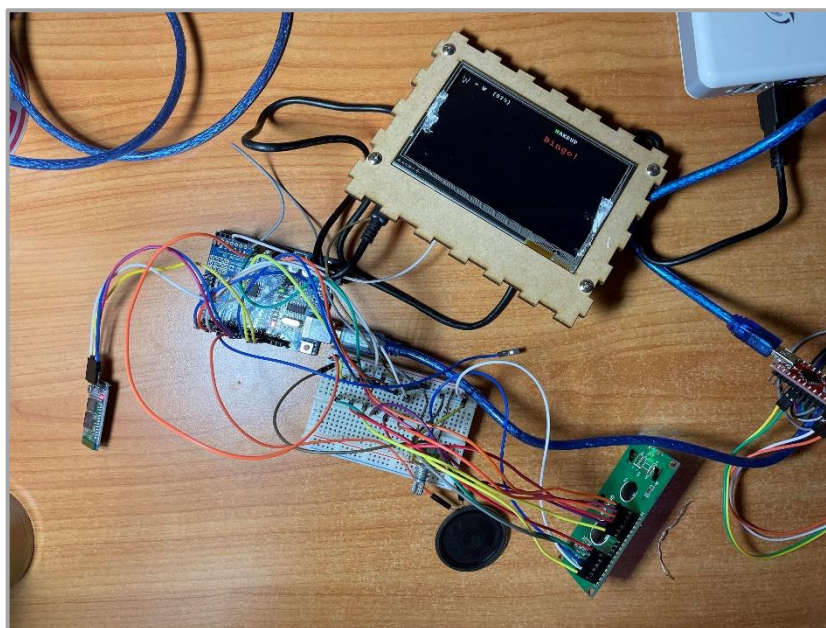
Arduino Uno 連接揚聲器、HC-05藍芽模組及I2C-LCD液晶螢幕
(其中Arduino Uno 需外接電源)



車身是用雷射切後的木板拼裝而成（共六塊，含上蓋）



三塊開發版間的連接情形



成品圖！



▼ 未來展望

在這次賽期中，我們研究的套件比用上的多太多了，包括 TouchGFX、X_CUBE_AUDIO 等等。最主要的問題是我們太晚發現 **FreeRTOS** 這個專案流程整合套件，沒時間把他學會，使鬧鐘的各功能無法合併到一個開發板上。

為使鬧鐘仍能滿足我們的要求，目前的替代方案是使用多個板子進行各自的工作，而不同板子間的溝通由腳位的 I/O 達成。例如使用者在 STM32F746G-DISCO 的觸控板寫完字後，該板子會在設定好的腳位輸出高電位，當 Arduino Nano 讀到此訊號，就會讓馬達停止運作。

(下頁續)

但這樣的缺點就是高成本與增加硬體的複雜程度，由外接電池的數量就能看出。若我們學會FreeRTOS，我們能夠達成以下事項：

1. 行事曆功能

我們原本要利用TouchGFX製作像圓餅圖的時間表，隨著時間過去，會有更多面積被塗顏色，而每個顏色代表一個行程。如此一來能使在家工作者更有時間規劃能力，提升工作效率。

實作成果影片：https://youtu.be/ccbx_Q9iFEc

2. 音響USB隨插隨放

我們有利用X_CUBE_AUDIO實作出播放USB中mp3&wav檔的專案，若此功能也被放入鬧鐘，我們的鬧鐘還可以拿來當作播放自己最愛音樂的好音響！

實作成果影片（影片中micro-USB連接出的裝置便是一個隨身碟）：

https://youtu.be/_JEIA93tt-Y

3. 大幅縮減開發板跟電池數量

若能用FreeRTOS將各個專案結合，我們就能只用一塊STM32F746G達到以上事項。如此一來就不用外接那麼多電池給其他板子供電，鬧鐘所需要的體積也會變得更小、重量更輕，使鬧鐘變的小巧具有攜帶方便性。方便在後端設計電源插孔，將產品外觀包裝得更直覺、簡潔。

4. 使用者圖形化介面

若克服技術性問題，我們也可以開發手機的app。透過app的圖形化介面，使用者能更方便在睡前設定隔天的鬧鐘與行程。合併上述的行事曆功能，必然會增加智能鬧鐘的使用彈性與便利性。

5. 人聲播報天氣

若能夠克服藍芽傳訊的問題，傳送mp3的格式（兩邊都要能進行轉檔的動作，因為藍芽無法直接以藍芽格式傳遞訊息，目前有實作出STM32轉檔的功能，在X_CUBE_AUDIO中），我們便能實現以人聲播報天氣的想法，也可以讓使用者可以更方便的接收其他訊息（用python爬蟲即可）

實作成果影片：<https://youtu.be/z2YlpTxjsAM>

▼ 程式碼（皆已push到Github上）

本作品有實際用上的

1. Arduino Nano Code

https://github.com/MakeNTU/2021_team10_/blob/main/src/obstacle_avoidance.ino

2. Arduino Uno Code

https://github.com/MakeNTU/2021_team10_/blob/main/src/LCD_display_and_music.ino

3. STM32F746G-DISCO Code

https://github.com/MakeNTU/2021_team10_/blob/main/src/main%20-%20AI.c

4. Python Code here（顯示天氣至LCD面板）

[https://github.com/MakeNTU/2021_team10_/blob/main/src/weather_report%20\(version_1\).py](https://github.com/MakeNTU/2021_team10_/blob/main/src/weather_report%20(version_1).py)

5. Python Code here（讓speake以中文播報天氣）

[https://github.com/MakeNTU/2021_team10_/blob/main/src/weather_report\(version_2\).py](https://github.com/MakeNTU/2021_team10_/blob/main/src/weather_report(version_2).py)

本作品未來展望可能用得上的

1. STM X-CUBE AI Code

https://github.com/MakeNTU/2021_team10_/tree/main/src/STM_X-CUBE-AI

2. STM TouchGFX Code

https://github.com/MakeNTU/2021_team10_/tree/main/src/STM_TouchGFX

3. STM X-CUBE AUDIO Code

https://github.com/MakeNTU/2021_team10_/tree/main/src/STM_X-CUBE-AUDIO