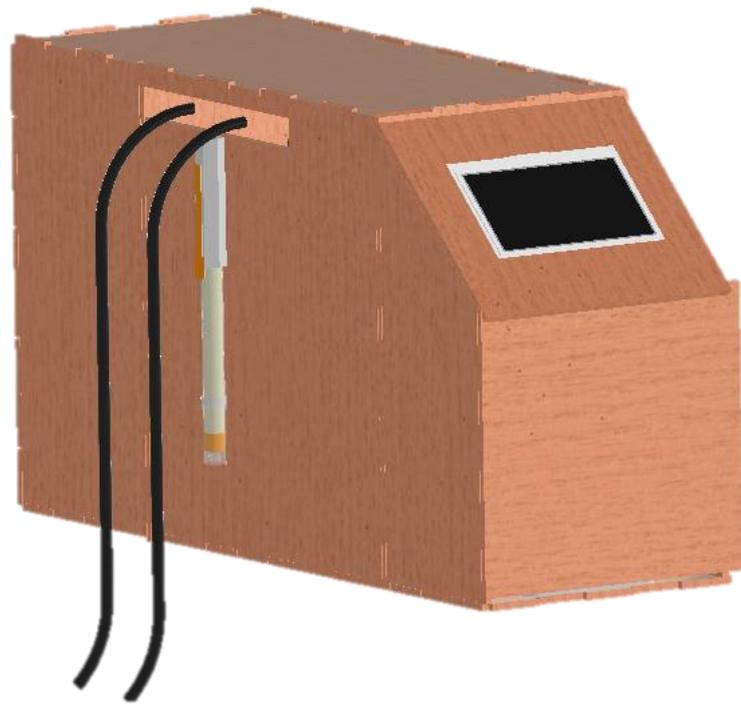


臺北市立大安高級工業職業學校專題製作競賽
「專題組」作品說明書



群別：電機與電子群

作品名稱：魚缸醫生

關鍵詞：HMI介面、自訂環境細部數值、隨時觀測

目錄

壹、摘要	1
貳、研究動機	1
參、主題與課程之相關性或教學單元之說明	1
一、硬體製作	1
二、程式撰寫	1
肆、研究方法	3
一、研究流程	3
(一)、研究步驟	3
(二)、操作步驟	4
二、使用材料及工具	6
(一)、零件介紹	6
(二)、機構原理	10
(三)、軟體介紹	10
伍、研究結果	12
一、硬體結構	12
(一)、餵食機	13
二、軟體通訊架構	12
(一)、餵食機	13
(二)、LINE Notify	13
(三)、Serial 串口通訊	13
三、成果展示	14
(一)、HMI 介面	14
(二)、LINE 介面	15
陸、討論	16
一、餵食機運轉不順	16
二、感測器互相影響	16
柒、結論	17
捌、參考資料及其他	18
一、網路資料	18

表目錄

表 1 時間分配表.....	3
表 2 ESP32 無線模組 規格.....	6
表 3 Mega2560 規格	7
表 4 溫度感測器 規格.....	7
表 5 E-201-C PH 感測器 規格.....	7
表 6 TDS 感測器 規格	8
表 7 沉水馬達 規格.....	8
表 8 步進馬達 規格.....	9
表 9 製冷晶片 規格.....	9
表 10 L298N 規格	10
表 11 電源供應器 RD-35A 規格.....	10
表 12 HMI 人機介面規格.....	10

圖目錄

圖 1	Arduino 程式介面	2
圖 2	HMI 編輯介面	2
圖 3	研究步驟.....	3
圖 4	溫度功能流程圖.....	4
圖 5	PH 控制流程圖	5
圖 6	TDS 水質軟硬度監測流程	5
圖 7	ESP32 無線模組.....	6
圖 8	Arduino Mega 2560 開發版.....	7
圖 9	溫度感測器.....	7
圖 10	E-201-C PH 感測器	7
圖 11	TDS 感測器	8
圖 12	船型開關.....	8
圖 13	沉水馬達.....	8
圖 14	步進馬達.....	9
圖 15	TEC1-12706 製冷機片	9
圖 16	水冷頭.....	9
圖 17	L298N	10
圖 18	RD-35A.....	10
圖 19	HMI 人機介面	10
圖 20	Autodesk Inventor Logo	11
圖 21	Autodesk Inventor 3D 圖繪製介面	11
圖 22	Arduino logo	11
圖 23	Arduino 程式撰寫	12
圖 24	3D 結構配置圖	12
圖 25	魚缸醫生實體外殼.....	13
圖 26	餵食機爆炸圖.....	13
圖 27	通訊關係圖.....	14
圖 28	HMI 主頁	14
圖 29	HMI 選擇魚種	15
圖 30	HMI 自訂數值	15
圖 31	HMI 設定完成	16
圖 32	LINE 介面.....	16
圖 33	電晶體控制感測器交替運作.....	17

【魚缸醫生】

壹、摘要

為了做到讓對養水族不同段位的使用者都能夠熟練操作我們的專題作品，我們利用 HMI 介面讓使用者透過觸控選定飼養物種或自訂環境細部數值，來針對個別飼養物種做照護或自行設定，並利用 PH 計、溫度感測棒和 TDS 水質軟硬度感測器探測出魚缸內環境後，Arduino mega 2560 運算是否注入藥劑調整水質或調整水溫，幫助使用者控制用藥和水溫，並可利用我們自製的餵食機定時投餵飼料，最後透過 Esp32 傳送魚缸內各項數值到使用者通訊軟體 Line 上以供其進行隨時觀測。

貳、研究動機

對於生活忙碌的現代人，對寵物提供多一分地照顧，如同多一分麻煩，特別是水族，除了照顧麻煩外，出狀況所需要調整的手段不但麻煩，徵兆又不明顯，需要進一步的觀察或不斷失敗的經驗累積才得以知曉情況，更不要說新手，在不了解的情況下，面對某些生命力差的物種，他所飼養水族可能要換好幾輪，所以我們決定以科學儀器做監測，利用程式運算測出的數值做出最準確的調整，就像家庭醫師對病患的診斷一樣，做精準的診斷、給最適當的治療，幫助養魚新手和無暇照顧寵物的人，最後我們決定研究一套完善的水族養殖自動化系統「魚缸醫生」。

參、主題與課程之相關性或教學單元之說明

一、硬體製作

高二電子學實習課中，我們學到利用 RDWorksV8 畫出我們需要在木板上切出的機構外殼形狀，再利用雷射切割機切割，和利用 Autodesk Inventor professional 2020 編輯我們自行研發的餵食機結構，並利用 3D 列印機列印。

二、程式撰寫

我們使用 Arduino IDE 為此次專題硬體撰寫控制程式，如圖 1，並使用 Mega2560、esp32 作為硬體平台。esp32 為此次專題的網路通訊器，將訊息傳自手機上。並且我們利用 USART HMI 來撰寫 HMI 面板的程式，如圖 2。

```

211 //making 10 sample and adding with 10 milli second delay
212 digitalWrite(LED_BUILTIN, HIGH);
213
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782 //
783 //
784 //
785 //
786 //
787 //
788 //
789 //
790 //
791 //
792 //
793 //
794 //
795 //
796 //
797 //
798 //
799 //
800 //
801 //
802 //
803 //
804 //
805 //
806 //
807 //
808 //
809 //
810 //
811 //
812 //
813 //
814 //
815 //
816 //
817 //
818 //
819 //
820 //
821 //
822 //
823 //
824 //
825 //
826 //
827 //
828 //
829 //
830 //
831 //
832 //
833 //
834 //
835 //
836 //
837 //
838 //
839 //
840 //
841 //
842 //
843 //
844 //
845 //
846 //
847 //
848 //
849 //
850 //
851 //
852 //
853 //
854 //
855 //
856 //
857 //
858 //
859 //
860 //
861 //
862 //
863 //
864 //
865 //
866 //
867 //
868 //
869 //
870 //
871 //
872 //
873 //
874 //
875 //
876 //
877 //
878 //
879 //
880 //
881 //
882 //
883 //
884 //
885 //
886 //
887 //
888 //
889 //
890 //
891 //
892 //
893 //
894 //
895 //
896 //
897 //
898 //
899 //
900 //
901 //
902 //
903 //
904 //
905 //
906 //
907 //
908 //
909 //
910 //
911 //
912 //
913 //
914 //
915 //
916 //
917 //
918 //
919 //
920 //
921 //
922 //
923 //
924 //
925 //
926 //
927 //
928 //
929 //
930 //
931 //
932 //
933 //
934 //
935 //
936 //
937 //
938 //
939 //
940 //
941 //
942 //
943 //
944 //
945 //
946 //
947 //
948 //
949 //
950 //
951 //
952 //
953 //
954 //
955 //
956 //
957 //
958 //
959 //
960 //
961 //
962 //
963 //
964 //
965 //
966 //
967 //
968 //
969 //
970 //
971 //
972 //
973 //
974 //
975 //
976 //
977 //
978 //
979 //
980 //
981 //
982 //
983 //
984 //
985 //
986 //
987 //
988 //
989 //
990 //
991 //
992 //
993 //
994 //
995 //
996 //
997 //
998 //
999 //
1000 //

```

圖 1 Arduino 程式介面



圖 2 HMI 編輯介面

肆、研究方法

一、研究流程

(一)、研究步驟

暑假一開始我們就開始訂題目，在七月終我們訂定題目方向為養魚方面的監控，原本打算做魚塭方面的監控與管理，一直到八月終暑假結束前夕我們發現我們的研究成果，套用到觀賞用魚缸的管理較為完善，所以最終題目訂為「魚缸醫生」。

表 1 時間分配表

	7月	8月	9月	10月	11月	12月	1月
1.資料蒐集							
2.制定題目							
3.元件採購							
4.餵食機製作							
5.程式撰寫							
6.通訊連接							
7.成品測試							
8.外殼製作							

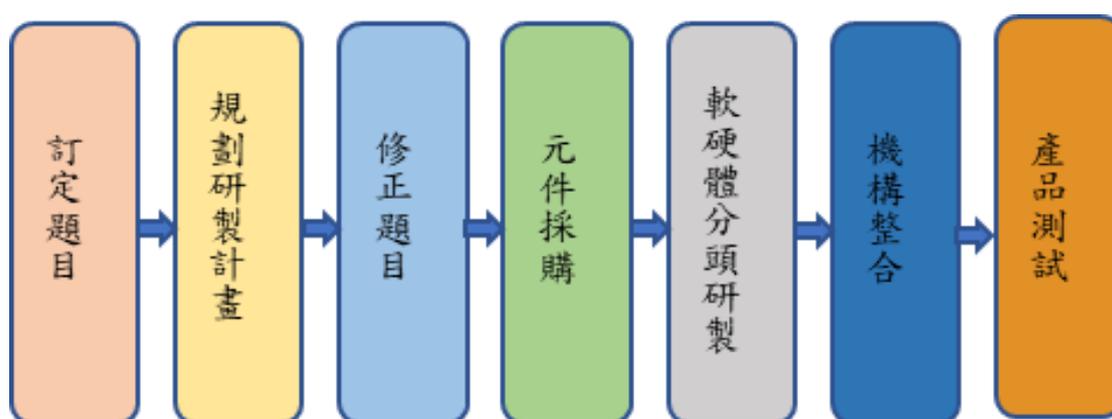


圖 3 研究步驟

(二)、操作步驟

為了清楚表達我們的動作流程，我們以文字解說輔以流程圖。

1. 溫控功能

(1) 溫控功能流程

我們對魚缸內水溫監控使用 Arduino P.H & Temp. sensor 的感溫棒，探測出水溫後回傳給與其連接的專用控制板，再傳輸到 Arduino Mega 2560 開發板進行判斷，如果溫度在設定值內，則不動作，如果需要調整，則驅動 L298N 繼電器控制魚缸內的抽水馬達，將水抽到散熱塔上的冷水頭內藉由下方的 TECC1-12706 製冷晶片的正反轉輸入，進行加熱或降溫，以達到控水溫的效果，並將檢測數值每五秒更新到 HMI 介面和使用者的通訊軟體 Line 上。如圖 4。

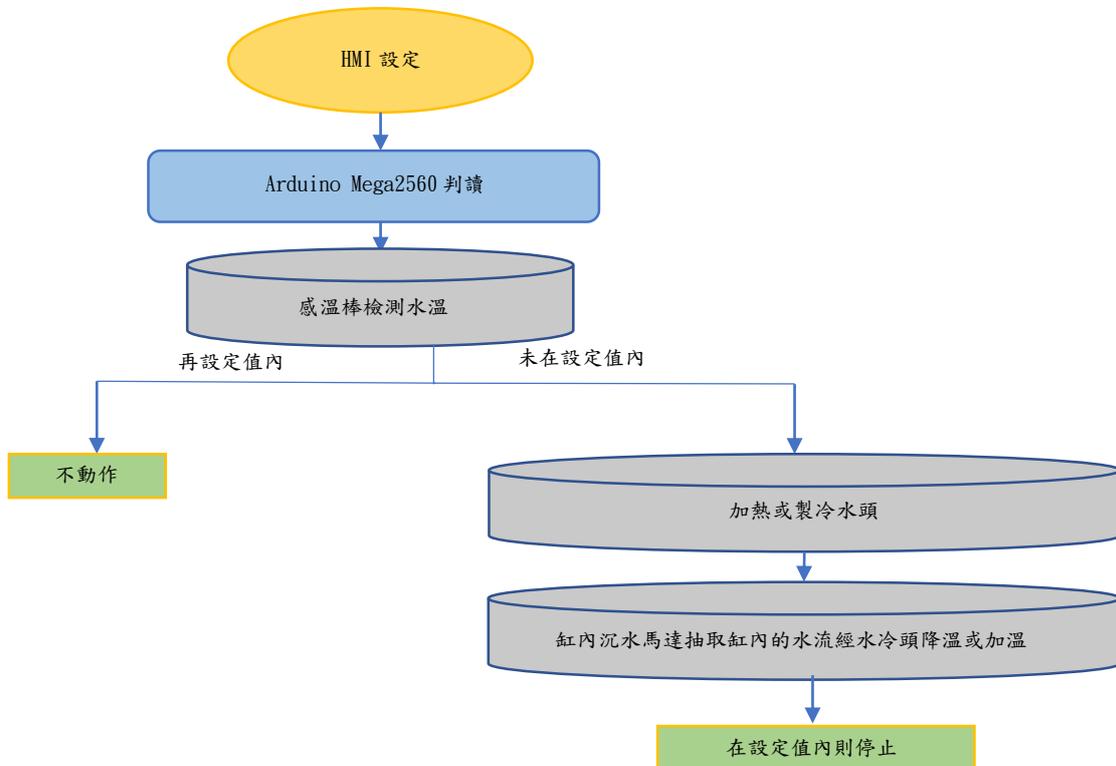


圖 4 溫度功能流程圖

(2) 水質酸鹼度控制

對於水質酸鹼度的調控，我們一樣利用 Arduino P.H & Temp sensor 的酸鹼度感測棒 E-201-C，探測出水質酸鹼度後，回傳到專用控制板判斷，在傳送到 Arduino Mega 2560 開發板進行判斷，如在設定範圍內則不動作，超出設定值，則驅動酸性藥水-亞馬遜黑水或鹼性藥水-蘆薈鹽飽和水溶液瓶內的抽水馬達，抽取藥水中和，並將檢測數值每五秒更新到 HMI 介面和使用者的通訊軟體 Line 上。如圖 5。

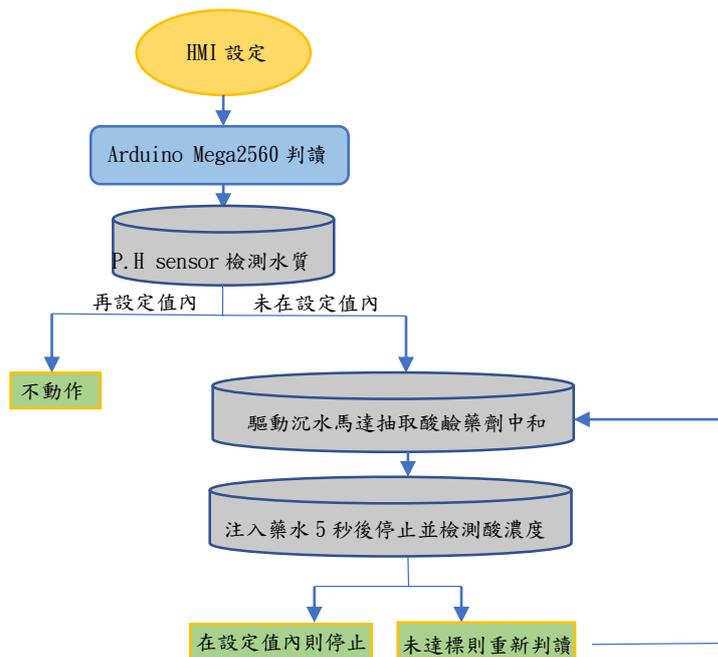


圖 5 P.H 控制流程圖

(3) 定時餵食功能

定時餵食功能是利用我們自製的餵食機，每天兩次定時投餵設定單位量的飼料

(4) TDS 水質軟硬度監測

水質軟硬度的監測可以理解為魚缸內的水是否髒到要替換的指標，當數值到 300~400ppm 時水就需要使用者進行換水的動作，TDS 的檢測是由 Arduino TDS sensor 的感測棒進行檢測後，將數值回傳給 Arduino TDS sensor 的控制板，再傳輸到 Arduino Mega 2560 開發板判斷，並將數值輸出到 HMI 界面和透過 ESP32，將數值傳輸到使用者的通訊軟體 Line 上，如圖 6。

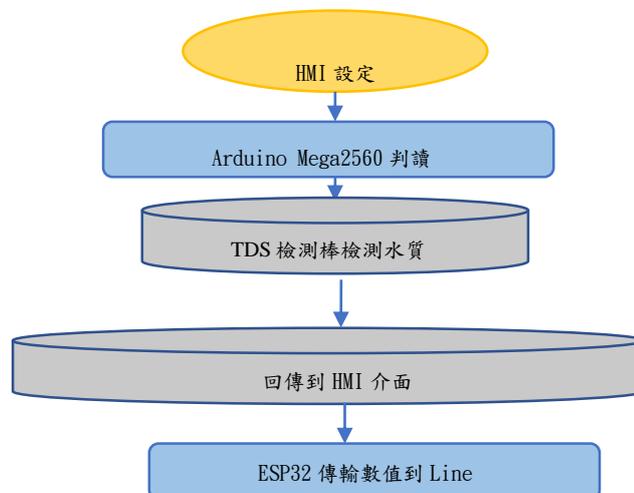


圖 6 TDS 水質軟硬度監測流程

(5)魚種設定和自訂設定值

透過點選我們裝設的 HMI 介面，選擇我們已輸入的魚種來設定環境各項指標，或者自行調控各項環境指標，來驅動魚缸醫生對缸內環境進行控制。

(6)遠端回饋數值給使用者

我們利用 ESP32 傳輸魚缸內環境如: P.H 值、溫度、水質軟硬度等數值到使用者的通訊軟體 Line 上，讓使用者能時時觀測到魚缸內是否有任何異常狀。

二、使用材料及工具

(一)、零件介紹

1. ESP32 無線模組

ESP32 無線模組是一款雙核心，並結合 WiFi 和藍牙功能的 32 位元微控制器，採用 Tensilica Xtensa LX6 微處理器，內建天線開關，RF 變換器，功率放大器，低雜訊接收放大器，濾波器和電源管理模組，如圖 7。

表 2 ESP32 無線模組 規格

廠牌	安信可
型號	NodeMCU-32S
腳位數	38 腳
核心處理器	Tensilica Xtensa LX6
核心	雙核 160/240 MHz
資料寬度	32 位元



圖 7 ESP32 無線模組

2. Mega2560

我們利用 Mega2560 進行程式的編寫和儲存，是目前 Arduino 程式編譯中最泛用的一顆單晶片，程式撰寫容易入門，且擁有大量網路資源及模組，對初學者來說，是一片非常友善的開發板。Mega2560 之規格與外型如表 3，圖 8：

表 3 Mega2560 規格

核心處理器	Mega2560
I/O 數	54
程式記憶體大小	256KB
程式記憶體類型	閃存
電壓 - 電源	DC 7V ~ 12V
時脈速度	16 Mhz



圖 8 Arduino Mega 2560 開發版

3. 溫度感測器

我們使用不銹鋼防水探頭型式的 DS18B20 溫度感測器，用來測量魚缸的水溫，而且 Arduino 已有內建 DS18B20 的函式庫可應用，而且溫度量測的範圍廣，準確度也高。其規格及外型如表 4 及圖 9：

表 4 溫度感測器 規格

溫度感測器	DS18B20
量測範圍	-55°C~125°C
準確度	±0.5°C
探頭材質	不銹鋼



圖 9 溫度感測器

4. E-201-C PH 感測器

我們利用 E-201-PH 感測器來感測魚缸中的 PH 值，適用於水族館，魚缸，實驗室等，其規格及外型如表 5 及圖 10：

表 5 E-201-C PH 感測器 規格

產品尺寸	r 6*120mm
測量範圍	0 - 14PH
元件功率	≤0.5W
工作溫度	-10~50°C
工作電壓	5±0.2V (AC·DC)
工作電流	5-10mA



圖 10 E-201-C PH 感測器

5. TDS 感測器

我們利用 TDS 感測器來感測魚缸中的軟硬度，該產品專為 arduino 設計，隨插即用，使用簡單方便。3.3~5.5V 的寬電壓供電，0~2.3V 的模擬訊號輸出，使得這款產品相容 5V、3.3V 控制系統，能非常方便的接到現成的控制系統中使用，其規格及外型如表 6 及圖 11：

表 6 TDS 感測器 規格

產品尺寸	42*32mm
測量範圍	0~1000ppm
工作電壓	DC 3.3~5.5V
工作電流	3~6mA



圖 11 TDS 感測器

6. 船形開關

船形開關裝設於機體後側，作為總電源，其外型如圖 12。



圖 12 船形開關

7. 沉水馬達

我們利用沉水馬達來讓讓水流動，其規格及外型如表 7 及圖 13：

表 7 沉水馬達 規格

產品尺寸	43mm X 2.3mm
水流量	1.2-1.6L/分鐘
重量	28 克
工作電壓	DC3-5V
工作電流	100-200mA



圖 13 抽水馬達

8. 步進馬達

我們利用步進馬達作為旋轉為時機的動力，其規格及外型如表 8 及圖 14：

表 8 步進馬達 規格

產品尺寸	28*20mm
步進角度	5.625/64
重量	44g
工作電壓	DC 5V



圖 14 步進馬達

9. 製冷晶片

我們利用製冷晶片來使水冷頭加溫及製冷，達到控制水溫的效果，其規格及外型如表 9 及圖 15：

表 9 製冷晶片 規格

產品尺寸	40*40*3.75mm
最大溫差	67°C 以上
工作電壓	DC12V
工作電流	5A



圖 15 TEC1-12706 製冷晶片

10. 水冷頭

我們利用水冷頭作為加熱及製冷的媒介，其外型如圖 16:



圖 16 水冷頭

11. L298N

我們利用 L298N 來控制製冷晶片，其規格及外型如表 10 及圖 17：

表 10 L298N 規格

產品尺寸	43*43*27mm
邏輯電壓	DC5V
邏輯電流	0mA-36mA
工作電壓	DC 5V-35V
最大電流	2A



圖 17 L298N

12. 電源供應器 RD-35A

我們利用 12V 的電源供應器提供給製冷晶片電源，5V 的提供給 Aduino mega 2560，其規格及外型如表 11 及圖 18：

表 11 RD-35A

產品尺寸	99(L)*82(W)*36(H)mm
輸入電壓	85~264VAC
輸出電壓	DC12V/5V
額定功率	32W



圖 18 RD-35A

13. HMI 人機介面

HMI 用於箱體上顯示目前魚缸的溫度、PH 值及 TDS 值，並可以調整溫度及 PH 值的範圍，其規格及外型如表 12 及圖 19：

表 12 HMI 人機介面

亮度	300nt
解析度	480*800
可是角度	135 度
對比度	1:500
響應速度	0.1s

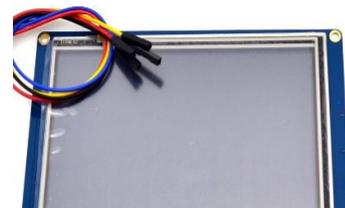


圖 19 HMI 人機介面

(二)、機構原理

利用與「Mega2560」偵測伺服器狀態，並控制步進馬達的旋轉，旋轉過程中飼料會從下方洞口掉出，上方洞口則會繼續補充飼料，達到餵食的效果。

(三)、軟體介紹

1. Autodesk Inventor

我們使用的 Autodesk Inventor，圖 20，是一款可以繪製 3D 立體圖、機構動作模擬、機械元件編輯和模型動畫編輯的一款軟體，圖 21。



圖 20 Autodesk Inventor logo

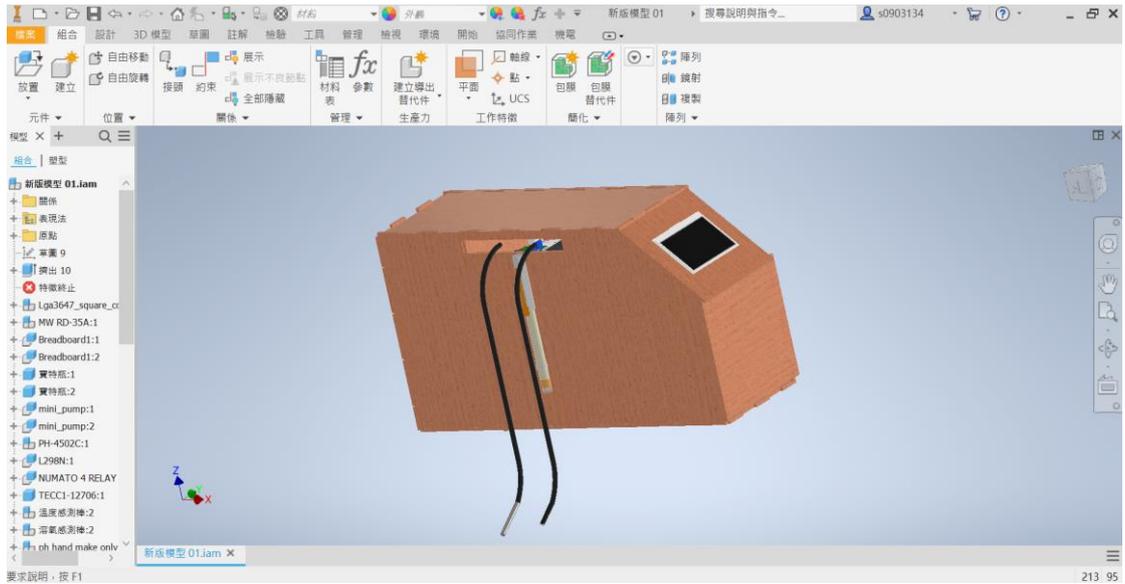


圖 21 Autodesk Inventor 3D 圖繪製介面

2. Arduino IDE

Arduino IDE 是一個開放原始碼的硬體程式語言編寫軟體，它兼具類似 java、C 等後端伺服器語言的開發環境，且擁有許多已模組化的套件與函式庫，提供初學者使用。由於它在控制單晶片的方便性，可以輕鬆連結硬體套件及通訊系統，所以我們選擇 Arduino IDE 作為硬體程式編寫的軟體，圖 22、圖 23。



圖 22 Arduino logo

```

mega | Arduino 1.8.9
檔案 編輯 基礎 工具 說明
mega
delay(200);
digitalWrite(24, HIGH);
delay(400);
digitalWrite(24, LOW);
delay(1000);
digitalWrite(22, LOW);
update_data("get_data");
delay(5000);
update_data("get_data");
}
//指紋辨識
if (fp_status == 0) {
  if (getFingerprintIDez() == 1) {
    Serial.println("success");
    digitalWrite(26, LOW);
    digitalWrite(22, HIGH);
    delay(200);
    digitalWrite(24, HIGH);
    delay(400);
    digitalWrite(24, LOW);
    delay(1000);
    digitalWrite(22, LOW);
  }
}
//指紋註冊
if (fp_status == 1) {
  update_data("get_data");
  while (getFingerprintEnroll() != true);
  delay(15000);
}

```

圖 23 Arduino 程式撰寫

伍、研究結果

本專題結構可以分成硬體結構與軟體通訊架構，以下所示。

一、硬體結構

本次專題結構，如圖 24、25。

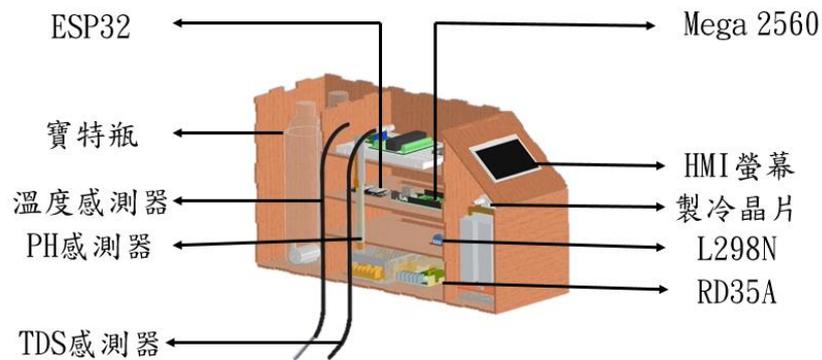


圖 24 Inventor 爆炸圖



圖 25 魚缸醫生實體外殼

(一)、餵食機

利用 3D 列印機製作多數機械零件，並用伺服馬達驅動餵食機中間的轉軸，以投餵設定格數的飼料如圖 26。

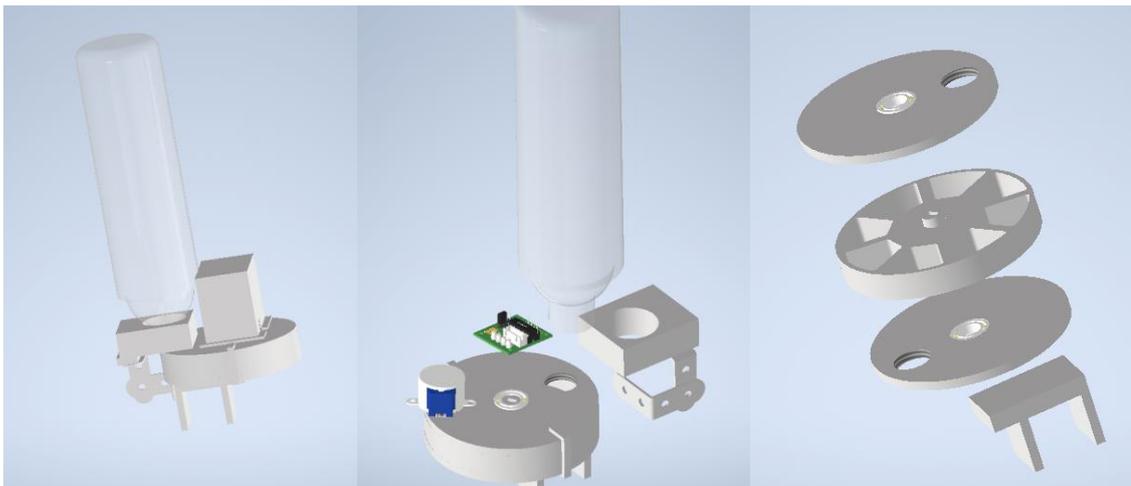


圖 26 餵食機爆炸圖

二、軟體通訊架構

魚缸醫生的功能就是能幫忙調整魚缸內的生活環境並且同時定時更新魚缸各項數值的情況，本專題是運用 LINE Notify 和串口通訊來傳遞指令與數值，通訊關係如圖 27。

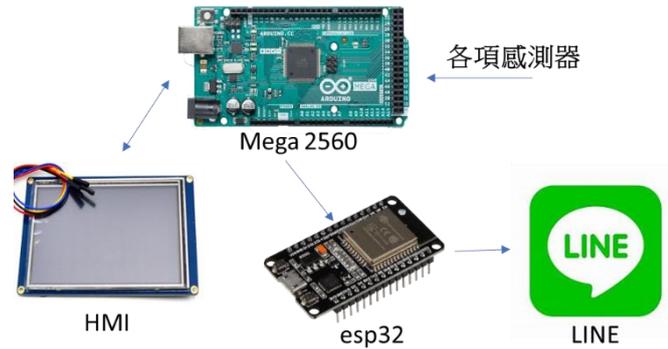


圖 27 通訊關係圖

(一)、LINE Notify

我們使用 LINE 的 LINE Notify 這項服務，當 esp32 接收到 Mega2560 的數值後，利用 WIFI 將訊息傳給手機的 LINE 上，即時更新魚缸的狀況。

(二)、Serial 串口通訊

串口通訊是利用 RX、TX 在不同的電子設備傳遞資料，使 Mega2560、esp32 與 HMI 進行資料之間的交換。Mega2560 能接受 HMI 傳遞過來的指令，並且 Mega2560 可以將感測器讀取的數值傳向 HMI 跟 esp32，讓 HMI 能將數值顯示在螢幕上、esp32 能將魚缸的狀態傳給手機的 LINE。

三、成果展示

(一)、HMI 介面

HMI 分為三部分，主頁可更換模式，選擇魚種模式，可以選擇四種魚，設定數值皆是該魚種的最佳生存範圍，自訂數值模式，可以自行更改溫度及 PH 值得上限及下限，當調整完成後，點選確定及更新數值，上方的目標數值就會更新，在下方的三個數值就是魚缸目前的狀況。如圖 28、29、30、31。

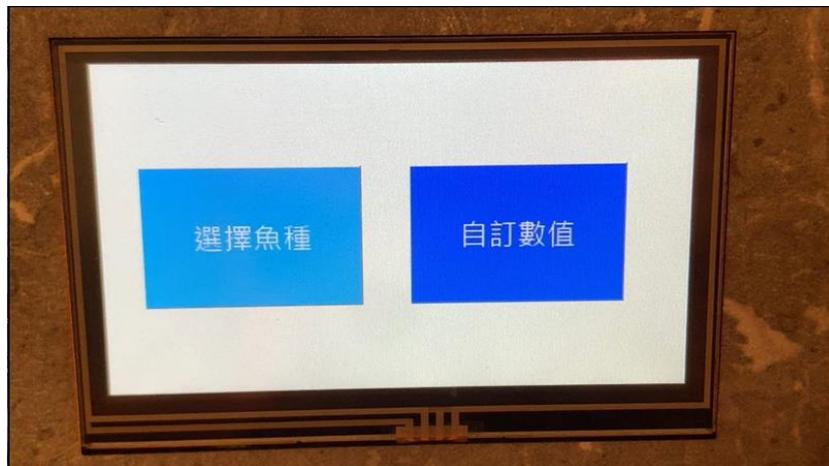


圖 28 HMI 主頁



圖 29 HMI 選擇魚種

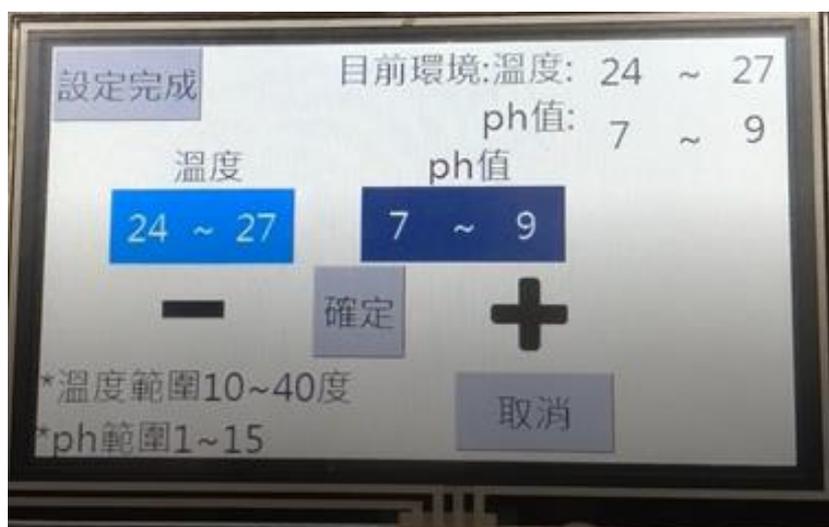


圖 30 HMI 自訂數值

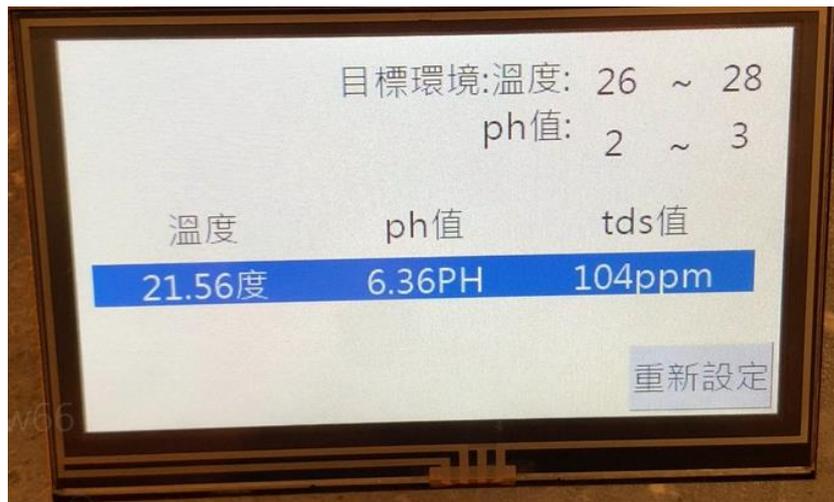


圖 31 HMI 設定完成

(二)、line 介面

機構會定時傳送魚缸水質的數值，如果魚缸中的水太髒，則會發送建議換水的訊息。如果機構在改變水質，則會發送數值調整中。如圖 32。



圖 32 LINE 介面

陸、討論

一、餵食機運轉不順

我們的餵食機設計是先透過自行發想，再參考市面上的機型進行改良所產出，但由於設計規畫做得不夠精細，許多問題都是在零件產出並裝設後才發現，最嚴重的問題是圓盤間的摩擦力過大和採用的步進馬達馬力不足，再加上飼料有時會被旋轉的圓盤縫隙絞碎，並卡住機構，目前想到最保險的方法是購買馬力更大的步進馬達。

二、感測器互相影響

我們發現在 tds 感測器和 ph 感測器放在同一缸水的時候會相互影響，導致所顯示的數據異常。我們之後在一個叫 electronicclinic 的網站發現有人遇到了相同的問題並有相應的解決方式，他就利用兩顆電晶體，控制 tds 感測器的電源，並讓他交替運作，接線如圖 33，之後我們又對他的接線做了些調整，原本他只有讓 tds 感測器交替運作，後來我們將 ph 感測器也交替運轉，並讓兩感測器運作的時間錯開。經過調整之後我們的到了更穩定的數據，感測器也可以維持原本的運作。

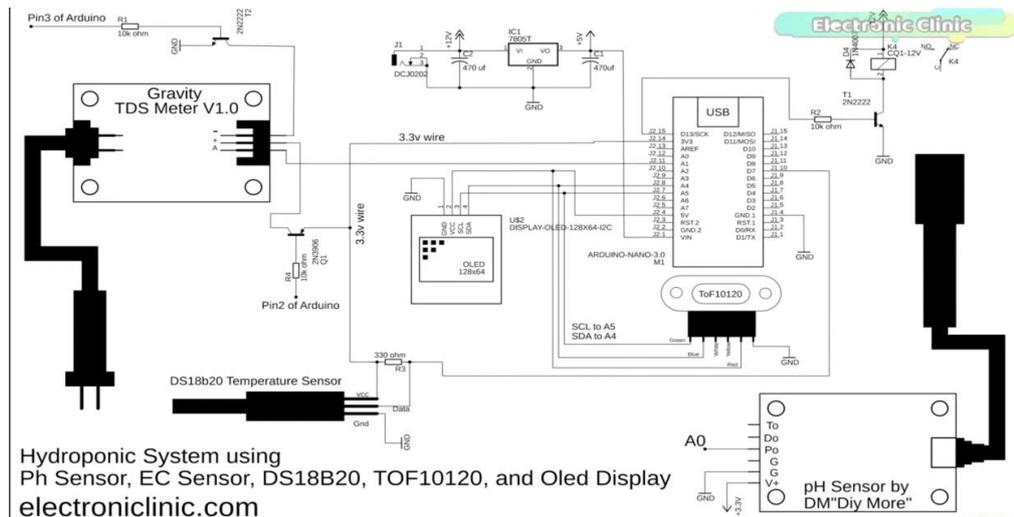


圖 33 電晶體控制感測器交替運作

柒、結論

在經過組員間的討論，我們將自製的冷水機放置在主要的機構內，這樣雖造成整體機構體積的增加，但就省去要另外設置冷水機的空間。調整酸鹼值的部分是採用兩個寶特瓶搭配 Arduino 沉水馬達，抽取調配好的酸水或是鹼水來和魚缸中的水進行中和。而溫度的部分則是透過冷水機上的制冷晶片進行加熱或制冷上方的水冷頭，再利用沉水馬達抽取魚缸中原本的水，讓水流過水冷機上方的水冷頭，達到控制溫度的效果。

為了使用者可以方便設定和監控溫度與酸鹼值，我們採用 HMI 人機界面，我們主要分成兩種功能，一個是選擇魚種，另一個是自訂數值，在選擇魚種的介面中你可以選擇你所養的魚種，機構就會讓魚缸的環境範圍保持在該魚種所適合的生存環境。在自訂數值的介面中，你可以自由設定 ph 值及溫度的上限和下限，去符合你理想的魚缸環境。

而我們遠端通知的部分是使用 Esp32 進行製作，我們透過 Arduino 的程式撰寫，去連接 Line 官方出的應用程式 Line notify，再將感測器所讀到的數據傳到手機中，這樣就算出門在外，也可以隨時得知魚缸內的情況。雖然我們專題的功能有許多，但還是有許多缺點需要改進，像是沉水馬達啟動後造成感測器的不穩定，和餵食機旋轉不順導致飼料沒有順利掉出等問題。

我們的專題歷時長達六個月，從一開始選定主題，到採購材料、設

計，最終完成作品，每一位組員竭盡全力向著同一個目標邁進，不管是負責程式撰寫，或是機構設計，或是上台報告，都能夠互相幫助，並適時地提供意見，甚至花費自己額外的時間對專題做出貢獻，就是這些一點一滴努力，積少成多，最後成為了我們的專題。雖然會有意見不合、遇到困難的時候，這時組內也會積極尋找解決的辦法，並請相關的專業人士協助，提供我們更多的想法與意見，來補足專題的不足之處。專題不只讓我們學習到許多新技能，也讓我們藉由實際的操作，去將平時所學應用在專題，其中也學習如何發現問題並解決問題，學習如何溝通、合作等，都是我們繼續進步的動力。

捌、參考資料及其他

一、網路資料

1. ESP32 連結 Line Notify。取自：
<https://ithelp.ithome.com.tw/articles/10271219>
2. 電晶體控制感測器交替運作。取自：
<https://www.electronicclinic.com/arduino-hydroponics-diy-hydroponics-system-using-ph-sensor-ec-sensor-hydroponic/>
3. AS608 Fingerprint Sensor Module With Arduino。取自：
<https://www.ncnynl.com/archives/201606/91.html>
4. USART HMI 智慧串列屏介紹。取自：
<https://www.itread01.com/content/1548685447.html>
5. HMI 程式書寫方法。取自：
<http://wiki.tjc1688.com/doku.php?id=6.%E6%8C%87%E4%BB%A4%E9%9B%86:2.%E4%B9%A6%E5%86%99%E8%AF%AD%E6%B3%95>