

# 中華民國第 60 屆中小學科學展覽會

## 作品說明書

---

高級中等學校組 工程學(一)科

團隊合作獎

052306

植基於二維陣列演算之魔術方塊互動學習系統

學校名稱：國立臺南高級工業職業學校

|                                 |                             |
|---------------------------------|-----------------------------|
| 作者：<br><br>職三 吳士彬<br><br>職三 侯冠辰 | 指導老師：<br><br>王怡婷<br><br>林國豪 |
|---------------------------------|-----------------------------|

關鍵詞：二維陣列演算、互動學習、步進馬達控制

## 摘要:

魔術方塊是許多世代於學生時期皆會碰觸到的機械類益智玩具，藉由三維旋轉，改變其排列組合，然許多初學者遇到的最大瓶頸是無法復原，花費金錢與時間卻無法還原，致使解決問題的興致大幅降低。本研究，旨趣於魔術方塊的互動學習，當吾人解題失敗或遇到窒礙，就可利用人機介面輸入目前的狀態，透過二為陣列演算，使系統進行還原分解動作，而玩家藉由觀察學習，除培養觀察力和專注力，進而啟發思考解決問題的能力，來著手自行解題，使魔術方塊的學習困難度降低，維持學習動機。

## 壹、研究動機

魔術方塊於校園風靡，是許多學生下課時間的益智玩具，本組研究成員也因深感興趣，而產生研究動機，希望能建構一個系統，可使用手機作為人機介面，輸入魔術方塊當前狀態，透過演算法解題，進而控制馬達來解析。起初我們參考了網路上已有的 6 軸步進馬達裝置，乙這樣的形式為原型(如圖 1、圖 2)來發展，希望降低運算與控制單元成本，在機構上使用 LEGO 積木為主體，並縮減為兩顆步進馬達配合二維控制。LEGO 是我從小到大都喜愛的一個積木玩具，它最大的優點是能自己自由拼裝出自己想要的架構來，又因在學校一、二年級時老師教我們用 Arduino 做出一些基本電路控制，且能和 APP Inventor2 做藍芽的傳輸，所以本研究最後決定利用 Arduino 搭配 LEGO 來完成這次的研究主題。

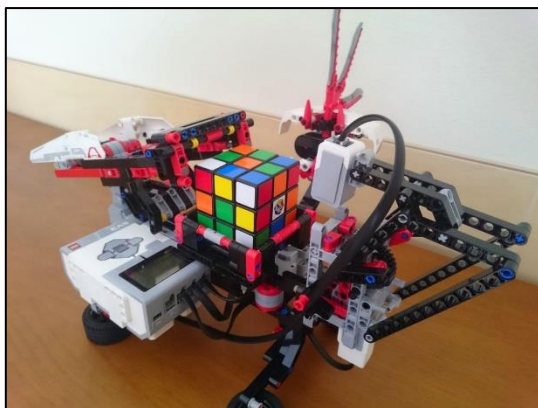


圖 1 LEGO EV3 魔術方塊還原機



圖 2 Arduino 魔術方塊還原

## 貳、研究目的

### 一、了解並設計機器架構

使用 LEGO 積木，設計魔術方塊互動學習系統爪具的機構，其中包含固定、旋轉、翻轉魔術方塊的機構設計。

### 二、了解步進馬達的使用及程式

使用步進馬達帶動魔術方塊互動學習系統機構，運用爪具機構連動，來帶動魔術方塊旋轉和翻轉，需要考慮到步進角的精準度，才能順利使魔術方塊旋轉到定位。

### 三、探討演算法運作方式，導入並結合機構

魔術方塊互動學習系統採用二為陣列的演算方法，透過程式實現於 Arduino 上，控制步進馬達與 LEGO 魔術方塊爪具機構，使魔術方塊得以達到還原的效果。

## 參、研究設備及器材

### 一、Arduino(UNO)簡介

Arduino 是一個開放源程式碼的硬體專案平台,該平台包括一塊具備簡單 I/O 功能的電路板以及一套程式開發環境軟體。它使用了 Atmel AVR 單晶片，建構於簡易輸出/輸入 (simple I/O) 介面板，並且具有使用類似 Java、C 語言的 Processing/Wiring 開發環境。因此剛開始我們使用 Arduinio UNO 板(如圖 3)來設計電路,再將設計好的電路自行刻出,內部使用的單晶片為 ATMEGA328。

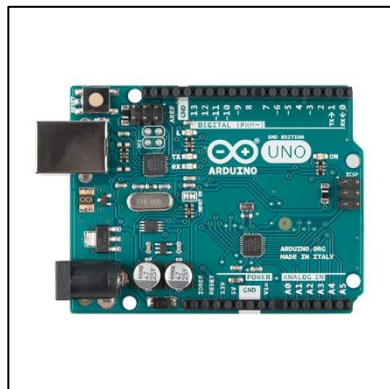


圖 3 Arduino Uno 板

Arduino 的特色:

(一)基於創用 CC 開放原始碼的電路圖設計。

(二)免費下載,也可依需求自己修改,但需遵照姓名標示。

(三)依據 Arduino 官方網站,取得硬體的設計檔,加以調整電路板及元件,以符合自己實際設計的需求。

(四)可簡單地與感測器,各式各樣的電子元件連接,如紅外線、超音波、熱敏電阻、光敏電阻、伺服馬達...等。

(五)支援多樣的互動程式,如 Adobe Flash, Max/MSP, VVVV, PureData, C, Processing...等。

(六)USB 介面,不需外接電源。另外有提供直流電源輸入。

## 二、APP Inventor

APP Inventor(如圖 4)是 Google 所提供的 Android 開發環境,目前版本為第二版(Beta),只需要有 Google 帳戶皆可免費使用。它不必學習困難的 Java 語言,而是利用類似圖形的方式做堆疊,由於他的簡單操作,並且還能支援控制 LEGO Nxt 及 EV3,所以對一些初學者或是想要利用 APP 控制 LEGO 機器人都是一個很好的選擇。

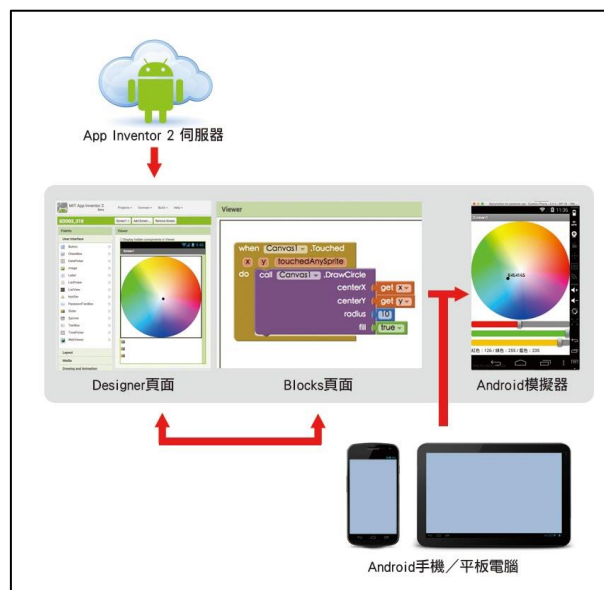




圖 4 App Inventor

### 三、藍芽

藍芽是一種無線通訊技術標準,用來讓固定與行動裝置,在短距離間進行交換資料,以形成個人區域網路。其使用短波特高頻無線電波,經由 2.4 至 2.485 GHz 的 ISM 頻段來進行通訊。它最初是希望建立一個 RS- 232 資料線的無線通訊替代版本。它能夠連結多個裝置,克服同步的問題。藍牙技術分為基礎率/增強資料和低耗能兩種技術類型。其中 BR/EDR 型是以對等網路拓撲結構建立一對一裝置通訊;LE 型則使用對等、廣播和網格等多種網路拓撲結構。藍芽有一個缺點是在 2.4GHz 的電波干擾下會產生傳送資料異常的問題,特別和無線區域路間的互相干擾問題。所以有干擾發生時,就以重新傳送封包的方法來解決干擾。

| 藍牙與Wi-Fi Direct技術比較 |   |   |
|---------------------|---|---|
| 特性                  |  Bluetooth® |  Wi-Fi DIRECT CERTIFIED® |
| 距離                  | 10~20米  | 100米以上  |
| 頻寬                  | 3Mbit/s(BT3.0+HS雖然利用WLAN傳輸較大檔案,但因只支援802.11g,目前只能達到54Mbit/s。)                                  | 300Mbit/s   |
| 耗電量                 | 較小  | 較大  |
| 配對方式                | PIN碼或配對鈕  | WPS、PIN碼或配對鈕  |
| 搜尋服務                | SDP   | Bonjour、UPnP  |
| 應用程式                | Bluetooth Profile如FTP、A2DP、HFP等   | TCP/IP網路應用程式如FTP、HTTP、DLNA等   |

資料來源：海華科技

圖 5 藍芽及 wifi 的比較圖

#### 四、L298N

L298N 直流馬達控制模組(如圖 6)所示，它可以驅動兩個直流馬達或是一個步進馬達，利用 Arduino 或是其它微控制器輸入控制訊號，就可以簡單控制直流馬達正反轉。馬達控制器上，附有一個 5V 穩壓器，使用者可以選擇是否要打開 5V 輸出介面，給 Arduino 或其它微控制器。

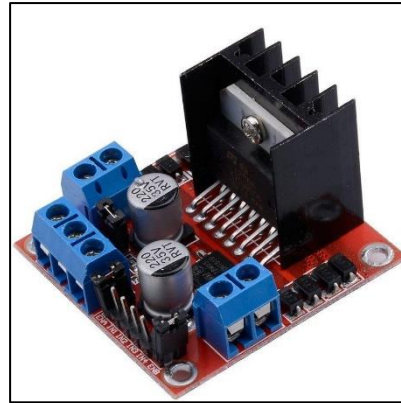


圖 6 L298N 直流馬達控制器

表 1 L298N 端子台

| NO | 名稱  | 說明              |
|----|-----|-----------------|
| 1  | A+  | Motor A+        |
| 2  | A-  | Motor A-        |
| 3  | VMS | 供電<br>(+5V~35V) |
| 4  | GND | Ground          |
| 5  | 5V  | 5V output       |
| 6  | B+  | Motor B+        |

表 2 L298N 排針

| NO | 名稱  | 說明              |
|----|-----|-----------------|
| 1  | ENA | Enable Bridge A |
| 2  | IN1 | Input 1         |
| 3  | IN2 | Input 2         |
| 4  | IN3 | Input 3         |
| 5  | IN4 | Input 4         |
| 6  | ENB | Enable Bridge A |

## 五、使用器材

表 3 材料表

| 材料名稱    | 規格      | 數量 | 功能          |
|---------|---------|----|-------------|
| Arduino | Uno     | 1  | 魔術方塊演算、控制步進 |
| 藍芽模組    | HC-05   | 1  | 數值傳送        |
| 步進馬達    | 42 型    | 2  | 轉動機台        |
| LEGO    |         |    | 主體          |
| 馬達控制器   | L298N   | 1  | 放大電流        |
| 手機      | Android | 1  | 輸入顏色        |

表 4 使用軟體

| 軟體名稱         | 說明              |
|--------------|-----------------|
| Arduino      | 撰寫馬達控制、演算法、藍芽接收 |
| APP Inventor | 撰寫手機 APP        |

## 肆、研究方法及過程

### 一、系統架構與設計

再與老師溝通討論後，本研究依據每位同學所擅長的領域進行工作的分配，也會與同學們互相討論，提出覺得可行的方法，例如馬達要採用 LEGO 原廠的伺服馬達或步進馬達、APP 的人機介面設計及演算法我們要採取哪一種方式做判斷，研究流程如圖 7 所示。

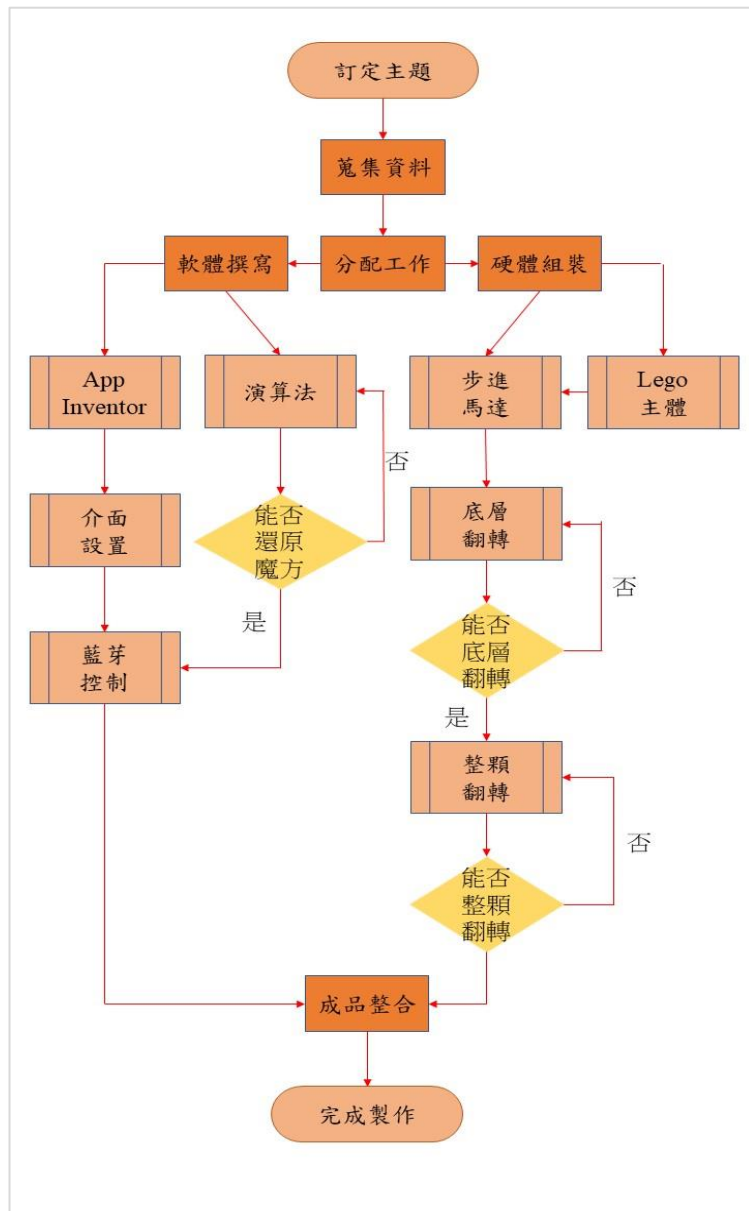


圖 7 系統架構概念



表 5 時間分配表

|          | 8 月 | 9 月 | 10 月 | 11 月 | 12 月 | 1 月 | 2 月 |
|----------|-----|-----|------|------|------|-----|-----|
| 1 收集資料   |     |     |      |      |      |     |     |
| 2 購買材料   |     |     |      |      |      |     |     |
| 3 馬達測試   |     |     |      |      |      |     |     |
| 4 程式設計   |     |     |      |      |      |     |     |
| 5 APP 測試 |     |     |      |      |      |     |     |
| 6 成品測試   |     |     |      |      |      |     |     |
| 7 製作書面   |     |     |      |      |      |     |     |

## 二、硬體架構.

魔術方塊的整面翻轉(如圖 8)，我們是利用包覆整個魔術方塊的方式，透過機械手臂逆時針轉，來使魔術方塊能夠藉著底座後方的桿子配合機子傾斜來順利完成魔術方塊的翻面。

魔術方塊的底層旋轉(如圖 9)，是讓機械手臂包覆著二、三層，只讓第一層能夠旋轉的方式來完成。

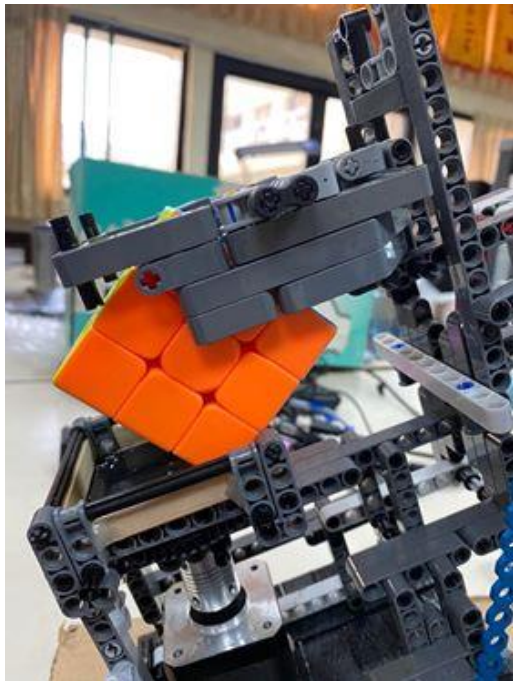


圖 8 魔術方塊翻面

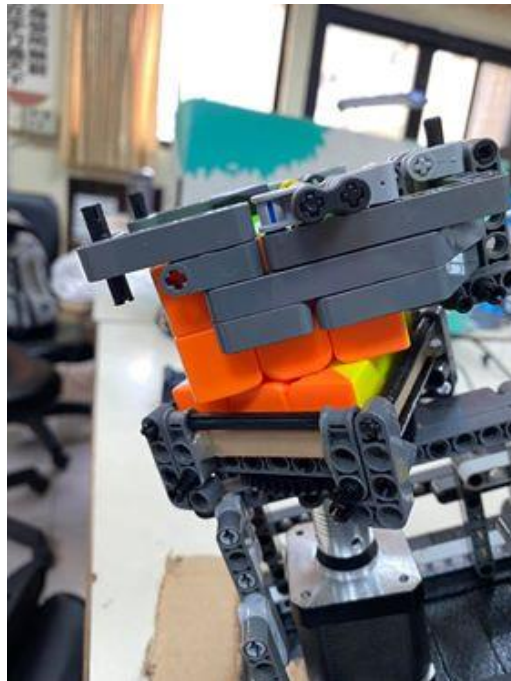


圖 9 魔術方塊旋轉底層

### 三、步進馬達控制程式

使用 L298N 馬達控制器來控制步進馬達，並根據上方表一、二的 L298 接腳表，來設計我們的佈線圖，由於 L298N 只能控制一個步進馬達，所以我們只好分成控制上方旋轉手臂，以及下方的底座。

我們用 `Stepper(int steps, pin1, pin2, pin3, pin4)`：建立一個步進馬達的物件，其中 `down` 是控制下方底座的步進馬達，而 `up` 是控制上方魔術方塊旋轉手臂的步進馬達。`step` 是指旋轉一圈所需的步數，假設使用馬達定義每步的角度，用 360 去除，就會得到步數，程式示意如圖 10。

```
sketch_dec27a$  
// Include the Arduino Stepper Library  
#include <Stepper.h>  
  
// Number of steps per output rotation  
const int stepsPerRevolution = 200;  
int i;  
// Create Instance of Stepper library  
Stepper down(stepsPerRevolution, 12, 11, 10, 9);  
Stepper up(stepsPerRevolution, 3, 4, 5, 6);  
  
void setup()  
{  
  // set the speed at 60 rpm:  
  down.setSpeed(60);  
  up.setSpeed(60);  
  // initialize the serial port:  
  Serial.begin(9600);  
}  
void loop()  
{  
  // step one revolution in one direction:  
  Serial.println("clockwise");  
  down.step(50);  
  delay(5000);  
}
```

圖 10 步進馬達控制示意圖

#### 四、人機介面—手機 APP 和藍芽傳輸

在人機介面上，本研究利用 APP INVENTOR 來撰寫我們手機 APP 的程式，其功能主要是讓使用者輸入魔術方塊個面的顏色，並且使用九宮格的方式讓使用者較好理解如何填入顏色，填完顏色後再根據 HC-05 藍芽控制器，發送相對應字母給 Arduino 做判斷，整體流程如圖 11。

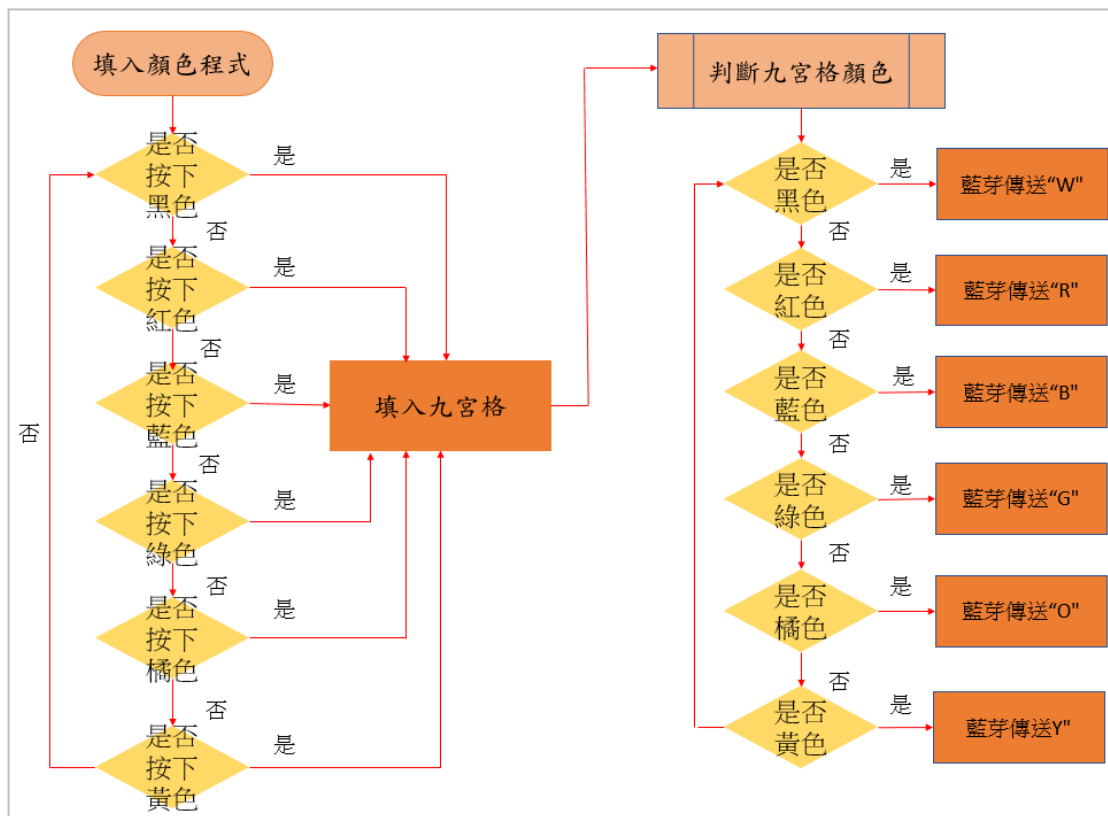


圖 11 人機介面-APP 流程圖

(一)下圖 12 為讓使用者選擇藍芽傳輸位置。

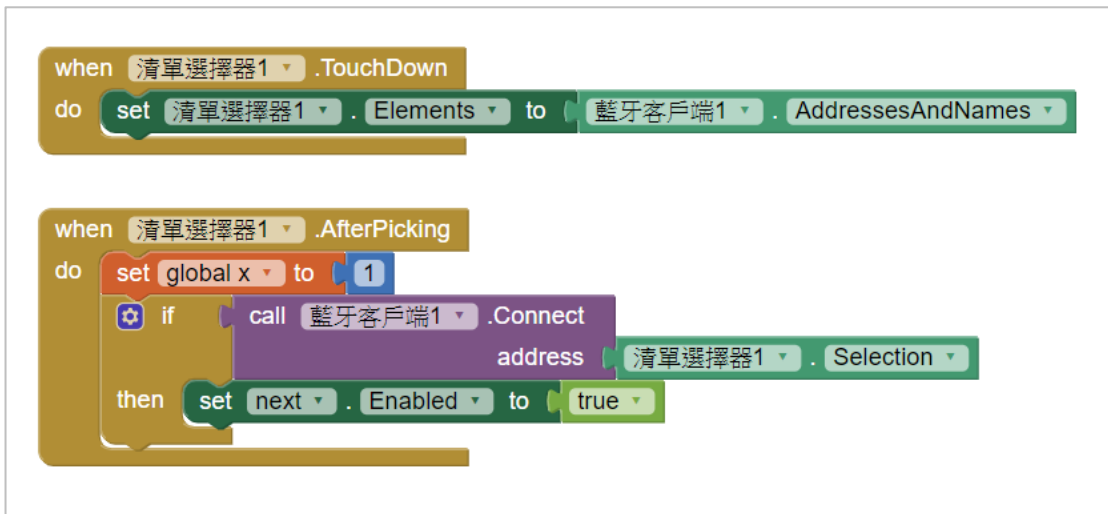


圖 12 選擇藍芽傳輸位置

(二)下圖 13 為，有先設置藍芽，點擊色塊能做填入九宮格的動作；如未連接藍芽則會顯示“未連接藍芽”。

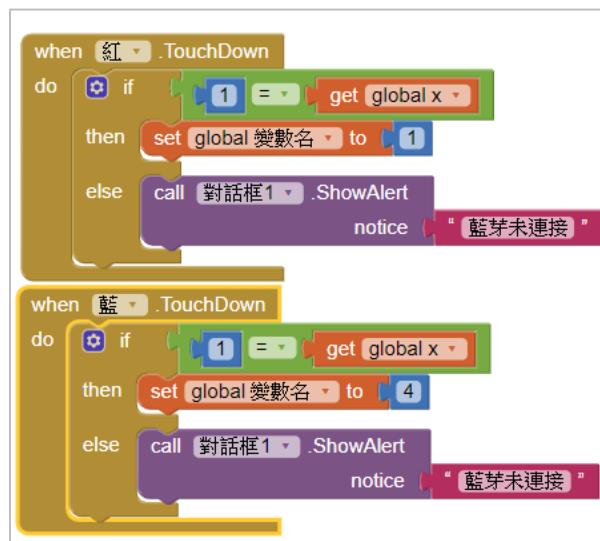


圖 13 色塊選擇按鈕

(三)下圖 14 為判斷各九宮格按下的顏色，並更改按鈕顏色和顯示。



圖 14 顯示九宮格顏色

(四)下圖 15 為判斷九宮格顏色，並用藍芽傳送數值。

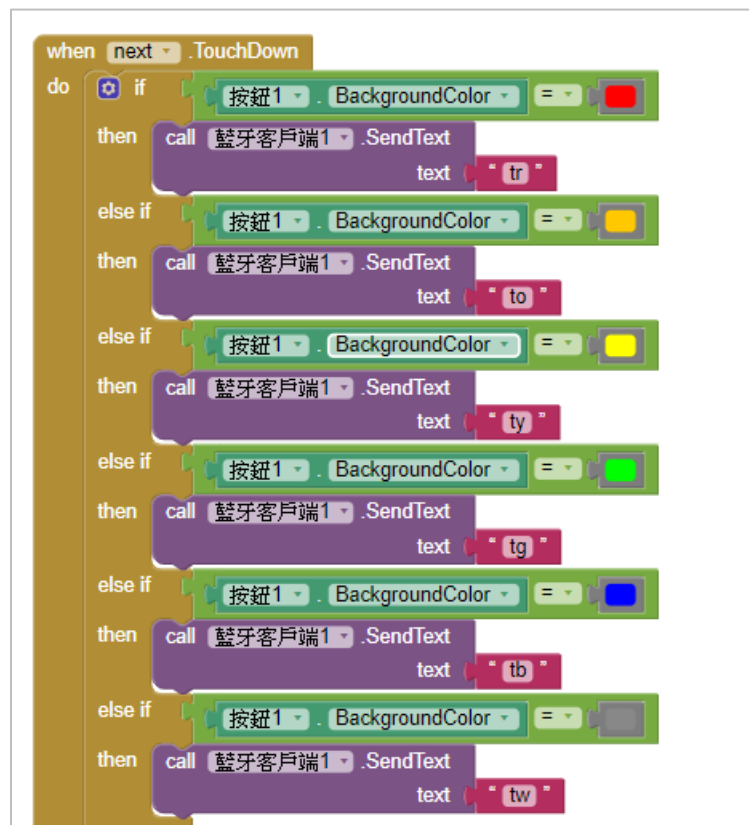


圖 15 判斷顏色並傳送

(五)下圖 16 為判斷是否六面選擇完成，跳到另一畫面做等待。

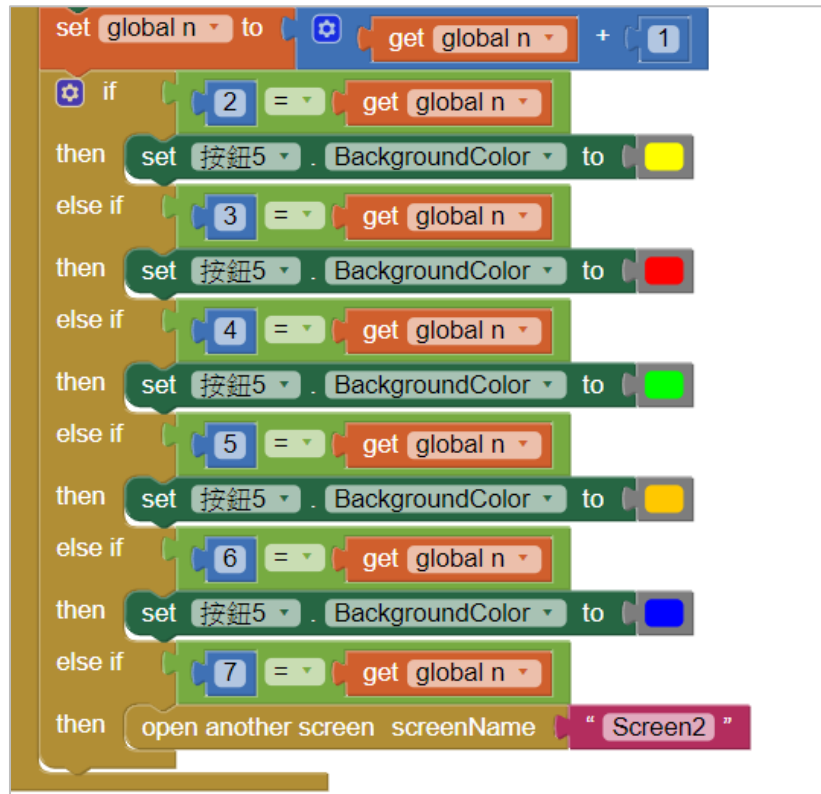


圖 16 判斷是否填完顏色

(六)藍芽:將從藍芽傳送的數值接收後做顯示(如圖 17)。

```
if (BT.available()) {  
  val= BT.read();  
  if(x<61){  
    color[x]=val;  
    Serial.print(val);  
    x++;  
  }  
  if(x==60){y=1;}  
}
```

圖 17 APP 藍芽接收程式示意

## 五、二維陣列演算法

本研究將二維陣列演算法透過程式，燒錄於 **Arduino** 上，乙利硬體進行運算與判斷，關於二維陣列的演算方式，行的部分是用以判斷面的九宮格(如圖 18)，順序是從做到右且上到下的方式；而列是判斷在哪一面，0 代表白色面、1 代表黃色面、2 代表紅色面、3 代表綠色面、4 代表橘色面、5 代表藍色面(如圖 19)。為了配合 **Arduino UNO** 板有限的記憶體容量，所以本研究將演算法精簡化，並維持一步一步的解析功能，以利玩家學習。

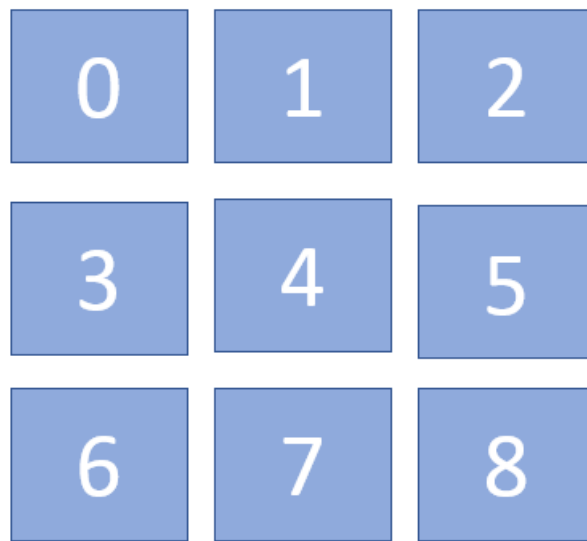


圖 18 魔術方塊九宮格對應數字

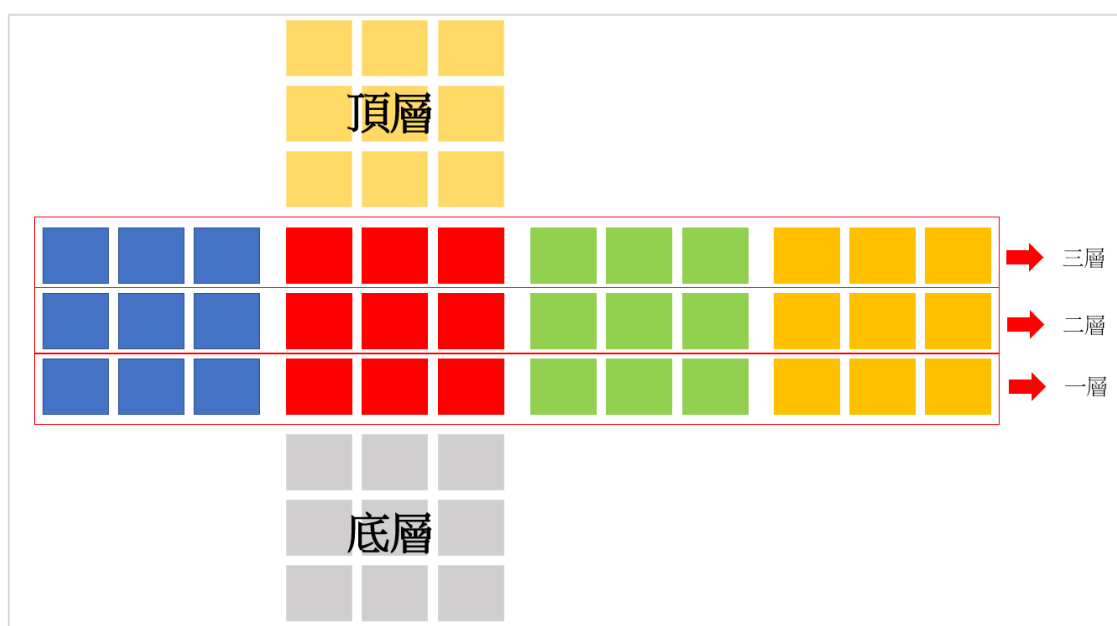


圖 19 魔術方塊分層名稱

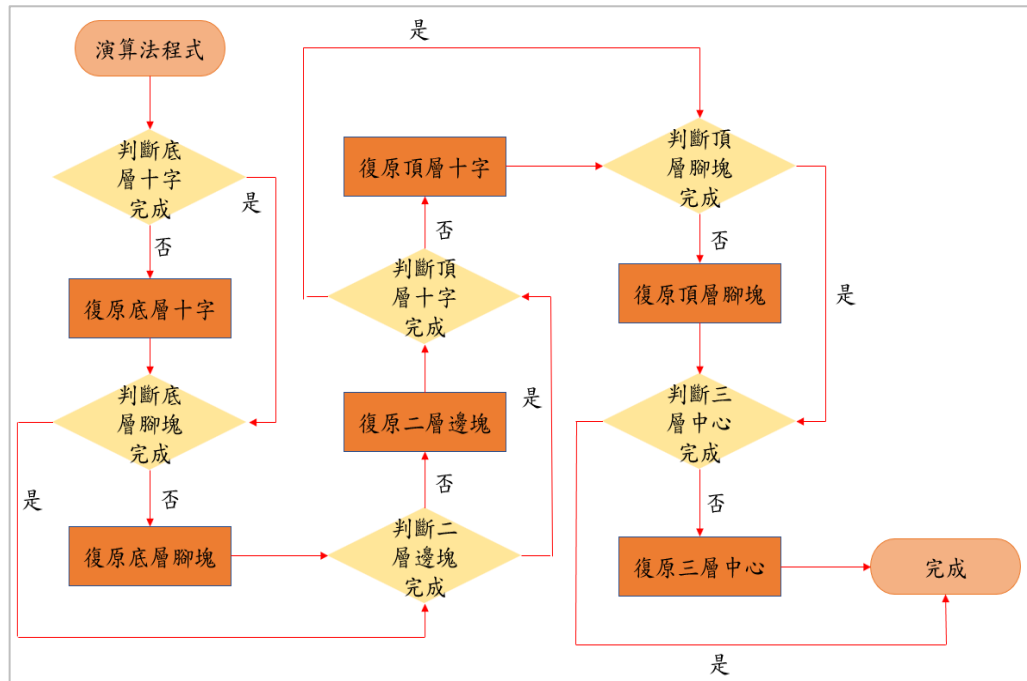


圖 20 演算法流程圖

(一)將接收到的數值做轉換並且填入到 cube 二維陣列裡(如圖 20、圖 21)。

|   |   |
|---|---|
| <pre> if (y) {     Serial.println(" ");     for (x=0; x&lt;61; x++)     {         if (color[x]=='w') {             checkcolor[y]=0;             y++;         } else if (color[x]=='y') {             checkcolor[y]=1;             y++;         } else if (color[x]=='r') {             checkcolor[y]=2;             y++;         } else if (color[x]=='g') {             checkcolor[y]=3;             y++;         } else if (color[x]=='o') {             checkcolor[y]=4;             y++;         } else if (color[x]=='b') {             checkcolor[y]=5;             y++;         }     } } </pre> | <pre> for (x=0; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } for (x=1; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } for (x=2; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } for (x=3; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } for (x=4; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } for (x=5; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } for (x=6; x&lt;54; x=x+9) {     changea[change]=cubein[x];     change++; } </pre> |
|---|---|



圖 20 判斷數值後用陣列存取

圖 21 將陣列數值傳給 cube 二維陣列

(二)判斷底層十字，第一層中心塊是否有顏色對應。

```
void solveCross() {
    int checkCross;
    if (cubies[1][0]==0 && cubies[3][0]==0 && cubies[5][0]==0 &&
cubies[7][0]==0 && cubies[7][2] == 2 && cubies[7][3] == 3 &&
cubies[7][4] == 4 && cubies[7][5] == 5) {
        checkCross=0;
        Serial.println(" ");
        Serial.println("Crossdone");
        checkBottom();
    }else{
        checkCross=1;}
    if(checkCross){
        bottomsolveEdge();
    }
    Serial.println(" ");}
```

(三)判斷含有白色邊塊的位置，並且依據位置做相對應的公式。

```
void bottomsolveEdge() {
    //Check edges on all sides
    for (int side=0; side<6; ++side) {
        for (int edge = 1; edge<9; edge+=2) {
            if (cubies[edge][side] == 0) {
                //Solve edge
                switch (side) {
```

(四)判斷底層是否為全部白色，並且第一層要全部顏色必須和中心塊顏色對應。

```
void checkBottom() {  
    int checkBottomLayer;  
  
    if (cubies[0][0] && cubies[2][0] && cubies[6][0] && cubies[8][0]  
        && cubies[6][2] && cubies[8][2] && cubies[6][3] && cubies[8][3]  
        && cubies[6][4] && cubies[8][4] && cubies[6][5] && cubies[8][5]) {  
        checkBottomLayer=0;  
  
        Serial.println("Bottomdone");  
  
        Serial.print("");  
  
        solveconers();}  
  
    else if (cubies[0][0]==0 && cubies[2][0]==0 && cubies[6][0]==0 &&  
cubies[8][0]==0) {  
        checkBottomLayer=0;  
  
        Serial.println("Bottomdone");  
  
        Serial.print("");  
  
        solveconers();}else {  
        checkBottomLayer=1;}  
}
```

(五)假如底層沒有全白，判斷含有白色的腳塊，並且依據位置移動到第三層。

```
if(checkBottomLayer) {  
    if (cubies[0][0] == 0 || cubies[6][2] == 0 || cubies[8][5] == 0) {  
        while (cubies[6][1] == 0 || cubies[0][2] == 0 || cubies[2][5] == 0) {  
            Serial.print(F("U,"));  
        }  
        Serial.print(F("FUF,"));  
    }  
  
    if (cubies[2][0] == 0 || cubies[8][2] == 0 || cubies[6][3] == 0) {
```

```
while (cubies[8][1] == 0 || cubies[2][2] == 0 || cubies[0][3] == 0 ) {  
    Serial.print(F("U,"));  
    Up(1);  
}  
Serial.print(F("RUR,"));  
}  
if (cubies[6][0] == 0 || cubies[8][4] == 0 || cubies[6][5] == 0) {  
    while (cubies[0][1] == 0 || cubies[2][4] == 0 || cubies[0][5] == 0 ) {  
        Serial.print(F("U,"));  
    }  
    Serial.print(F("LUL,"));  
}  
if (cubies[8][0] == 0 || cubies[8][3] == 0 || cubies[6][4] == 0) {  
    while (cubies[2][1] == 0 || cubies[2][3] == 0 || cubies[0][4] == 0 ) {  
        Serial.print(F("U,"));  
    }  
    Serial.print(F("BUB,"));  
}  
}  
}
```

(六)判斷在第三層含有白色腳塊的位置，依據位置判斷使用公式。

```
void solveconers(){
    int checkCorners;

    if (cubies[0][0]==0 && cubies[2][0]==0 && cubies[6][0]==0 &&
cubies[8][0]==0) {
        checkCorners=0;

        Serial.println(" ");

        Serial.println("conersdone");

        solveEdges();

    }

    else {

        checkCorners=1;}
}
```

(七)判斷二層是否與中心塊顏色對應，如果不是做判斷與解決。

```
void solveEdges() {
    int checkEdges;

    if (cubies[3][2] != 2 || cubies[5][2] != 2)
        checkEdges=0;

    else if (cubies[3][3] != 3 || cubies[5][3] != 3)
        checkEdges=0;

    else if (cubies[3][4] != 4 || cubies[5][4] != 4)
        checkEdges=0;

    else if (cubies[3][5] != 5 || cubies[5][5] != 5)
        checkEdges=0;

    else
        {checkEdges=1;

        Serial.println(" ");
}
```

```

    Serial.println("Edgesdone");

    solveOLLcross();}
while(checkEdges==0) {

    solveEdge();

    if (cubies[3][2] != 2 || cubies[5][2] != 2)

        checkEdges=0;

    else if (cubies[3][3] != 3 || cubies[5][3] != 3)

        checkEdges=0;

    else if (cubies[3][4] != 4 || cubies[5][4] != 4)

        checkEdges=0;

    else if (cubies[3][5] != 5 || cubies[5][5] != 5)

        checkEdges=0;

    else

        {checkEdges=1;

        solveEdges();}}

```

(八)判斷頂層十字並做復原。

```

void solveOLLcross() {

    int checkEdges;

    if(cubies[1][1] != 1 || cubies[3][1] != 1 || cubies[5][1] != 1 || cubies[7][1] != 1) {

        checkEdges=0;

        }else{

        Serial.println(" ");

        Serial.println("OLLcrossdone");

        checkEdges=1;

        solveOLL();}

```

```
if(checkEdges==0) {
    int state = findEdges();
    while (state == 0) {
        Serial.print (F("U"));
        Up(1);
        state=findEdges();
    }
    solveCase(state);
}
}
```

(九)判斷頂層腳塊並復原。

```
void PLLCorners() {
    int checkCorners;
    adjustUFace();
    if (cubies[0][2] == 2 && cubies[2][2] == 2 && cubies[0][3] == 3 && cubies[2][3] == 3 &&
cubies[0][4] == 4 && cubies[2][4] == 4 && cubies[0][5] == 5 && cubies[2][5] == 5) {
        checkCorners=1;
        Serial.println(" ");
        Serial.println("PLLCornersdone");
        PLLEdges();
    }
    else{
        checkCorners=0;}
    while (checkCorners==0) {
        solvePLLCorners();
        adjustUFace();
    }
}
```

```

if (cubies[0][2] == 2 && cubies[2][2] == 2 && cubies[0][3] == 3 && cubies[2][3] == 3
&& cubies[0][4] == 4 && cubies[2][4] == 4 && cubies[0][5] == 5 && cubies[2][5] == 5) {
    checkCorners=1;
    PLLCorners();}}

```

(十)判斷第三層中心塊並復原。

```

void PLLEdges() {
    int checkEdges;
    adjustUFace();
    if (cubies[1][2] == 2 && cubies[1][3] == 3 && cubies[1][4] == 4 &&
cubies[1][5] == 5) {
        checkEdges=1;
        Serial.println(" ");
        Serial.println("done!!");
        solve=1;}
    else { checkEdges=0; }
    while (checkEdges==0) {
        solvePLLEdges();
        if (cubies[1][2] == 2 && cubies[1][3] == 3 && cubies[1][4] == 4 &&
cubies[1][5] == 5) {
            checkEdges=1;
            PLLEdges();
        }
    }
}

```

## 伍、研究結果

### 一、電路佈局圖

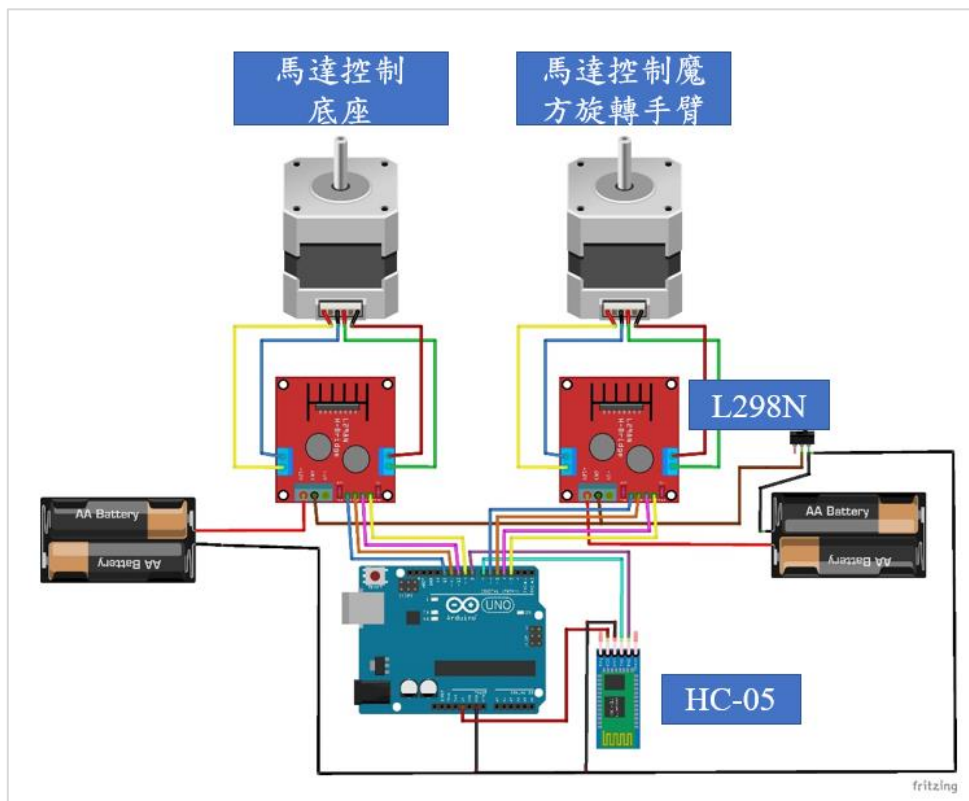


圖 22 電路佈局與設計

### 二、人機介面

如圖 23、圖 24 為我們利用 APP Inventor 來設計人機介面，圖 23 最上方為藍芽選擇清單，中間為顏色色塊的選擇，最下面為顏色填入的九宮格；圖 24 為輸入顏色完後的等待畫面。

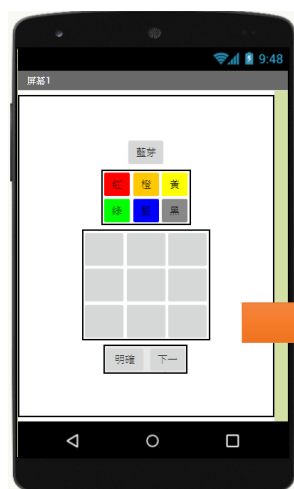


圖 23 手機 App 介面-1



圖 24 手機 App 介面-2



### 三、成品展示

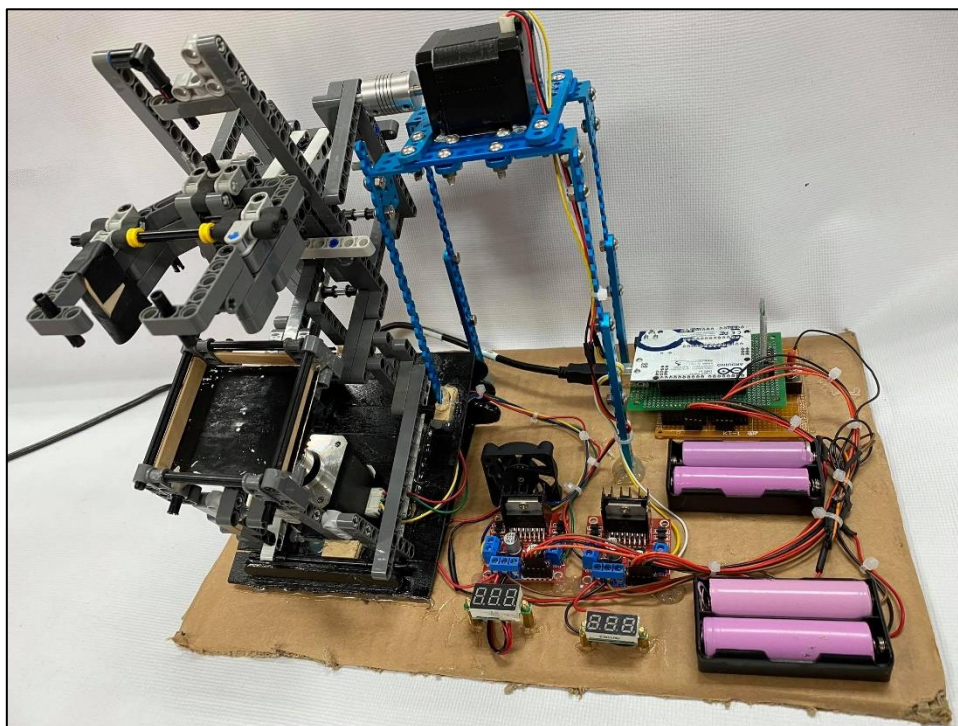


圖 25 實體成品

## 陸、討論

### 一、人機介面的輸入方式

本研究原先使用清單選擇的方式，讓使用者自行輸入色塊的動作，但因在測試的時候，每次都要先點開清單才能作選擇覺得很麻煩，為了直覺化設計，所以改成讓使用者點選色塊作填入，但一開始每填一個值，都要重選顏色，無法達到一種顏色馬達填兩三格，為了解決這個問題，本研究藉由設定一個全域變數，讓它能先儲存輸入值，這麼一來也就解決了每次都要先按色塊的問題。

### 二、藍芽傳輸裝置 HC-05 的值無法顯示

剛開始使用 HC-05 時，我們只要輸入顏色值，但輸出一直無法正常的顯示出來，之後在找查 **datasheet** 發現原來每一顆藍芽傳輸裝置的鮑率都不相同，而且鮑率會影響輸出的顯示，之後我們將鮑率改成相同但還是無法顯示，最後才發現原來是步進馬達與 HC-05 使用到相同的腳位了，修改 **pin** 腳後就能正常顯示。

### 三、二維陣列演算法撰寫

本研究原先擬將 C 語言程式寫好後，再用 **library** 的方式 **include** 到 **Arduino**，剛開始將撰寫好的 C 語言程式進行電腦單機測試，可以正常運作，但當我們要燒錄進 **Arduino** 時發現，有許多的函數庫無法讀取，像是 **string.h**、**iostream** 等等的，所以最後我們把演算法程式放入在 **Arduino** 裡作執行，但因 **Arduino Uno** 容量有限所以我們想辦法精簡演算法，並使之一步一步慢慢對使用者進行教學解析。

### 四、步進馬達無法正常運作

一開始使用 **L298N**，發送信號時，步進馬達無法正常運作，經過查詢原廠資料手冊後，發現需先將步進馬達和 **Arduino** 接地端相互連接(共地)，配線修改完成後，功能與動作正常了。

此外，初初開始，本研究使用 **Arduino** 控制步進馬達的函式庫時，測試發現它出現非預期的動作錯誤，經與老師詢問後，才知道原來步進馬達的腳位設定是是利用電力產生磁力的方式，假設接腳順序和要轉動方向街角順序不同時，導致磁力的順序也會跟著改變，而無法正常的工作，改善這個問題，動作就會正常。

## 柒、結論

本研究中，我們希望玩家透過系統，有互動的學習，而製作炫技的解題機器人，所以採用人們普遍有的智慧型手機作為人機介面，玩家可自行輸入，並填入畫面中魔術方塊相對應的顏色，目的是為了能讓玩家與系統進行溝通，產生想知道答案的興趣連結，進而增加觀察力以及專注力，激發他們動手嘗試的念頭。透過系統還原的步驟，一步一步慢慢摸索學習，訓練邏輯思考能力。此外，藉由本研究，將電機電子群三年內所學習的課程，像是程式設計、Arduino、單晶片實習、基礎電子實習等等，與研究做結合。透過研究，讓我們發揮自己的創造力，就像老師平常給我們練習的題目一樣，告訴我們，要按部就班地一一去測試完成，過程中遇到困難不是害怕逃避，而是要想辦法找出解決的方法，這是我們做研究時一直堅持的信念。

在這次的研究我們嘗試了許多的方法來完成，但有許多都因技術或經費限制問題，導致只能尋找其他的方法來完成，但也感謝這樣的限制，不僅拿過去時實習課常用的 Arduino 來當主要運算與控制器，甚至還去尋找步進馬達、藍芽、APP 的使用方法，途中我們也遇到了許多的困難，找查了各式各樣的資料及 datasheet，來解決所出現的問題，花了不少心思來處理錯誤。即使彼此在製作過程中而有不同的想法，但我們會先聽取雙方的意見作整合，並且一起討論分析來決定要採用哪種方式，雖然一路上跌跌撞撞，但我們都很享受這段學習的過程，也因為有這些失敗的經驗，從錯誤中成長，才能順利完成我們的研究。

## 捌、參考資料及其他

<1>陳會安(2018)•App Inventor 2 程式設計與應用•台北:全華圖書

<2>趙英傑(2016)•超圖解 Arduino 互動設計入門(第3版)•台北:旗標

<3>許溢适、陳坤正(2005)•步進馬達使用法•台北:全華圖書

<4>Arduino [https://en.wikipedia.org/wiki/Arduino\\_Uno](https://en.wikipedia.org/wiki/Arduino_Uno)

<5>APP inventor 圖

擷取自:[https://en.wikipedia.org/wiki/App\\_Inventor\\_for\\_Android](https://en.wikipedia.org/wiki/App_Inventor_for_Android)

<6>lego 圖 擷取自:<https://en.wikipedia.org/wiki/Lego>

<7>藍芽圖擷取自:<https://en.wikipedia.org/wiki/Bluetooth>

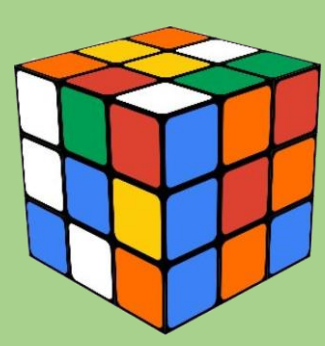
<8>步進馬達 [https://en.wikipedia.org/wiki/Stepper\\_motor](https://en.wikipedia.org/wiki/Stepper_motor)

<9>步進馬達圖 擷取自: <https://www.dimamotor.com.tw/product-detail-197560.html>

<10>Arduino 圖 擷取自: <https://www.taiwansensor.com.tw/product/arduino-uno-r3-%E7%BE%A9%E5%A4%A7%E5%88%A9%E5%8E%9F%E8%A3%9D-arduino-uno-rev33-%E9%96%8B%E7%99%BC%E6%9D%BF-made-in-italy/>

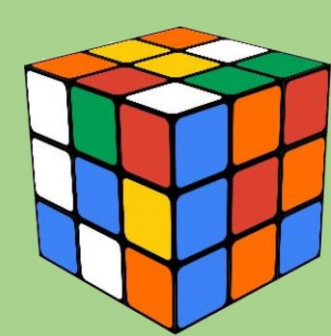
## 【評語】 052306

1. 本研究於魔術方塊的互動學習，可利用人機介面輸入目前的狀態，透過二維陣列演算，使系統進行還原分解動作，而玩家藉由觀察學習，除培養觀察力和專注力，進而啟發思考解決問題的能力。值得鼓勵！
2. 關於學習效益的提升，應建立評估基準與方法。
3. 坊間已有以各種軟硬體建構之魔術方塊自動解法機，關於本計畫的創新性和特殊性應進一步說明。
4. 應加強說明機構與翻轉機制的創意。
5. 以與使用者互動為訴求，可思考手動輸入初始值是否需要改變？例如：待解之魔術方塊現況建議導入視覺辨識，以拍攝六張照片方式自動帶入系統，省去人工輸入時間。
6. 應思考互動的部分應用場景為何？目前規劃似乎放入機器上就自動還原。
7. 應思考是否可以機器出題？



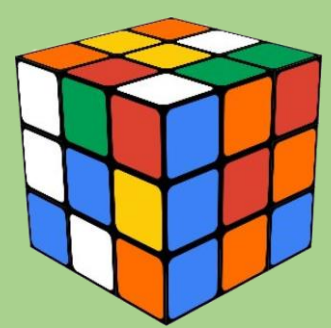
# 摘要

本研究旨趣於魔術方塊的互動學習，當吾人解題失敗或遇到窒礙，就可利用人機介面輸入目前的狀態，透過二為陣列演算，使系統進行還原分解動作，而玩家藉由觀察學習，除培養觀察力和專注力，進而啟發思考解決問題的能力，來著手自行解題，使魔術方塊的學習困難度降低，維持學習動機。

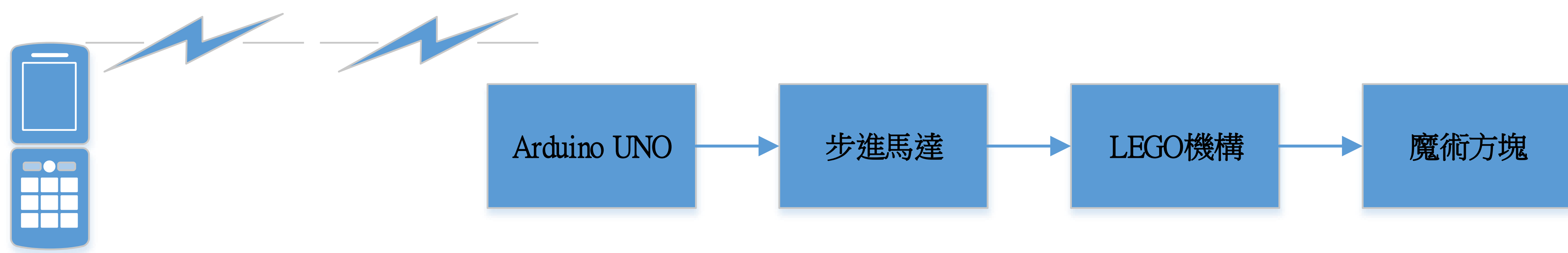


# 研究動機

魔術方塊於校園風靡，是許多學生下課時間的益智玩具，本組研究成員也因深感興趣，而產生研究動機，希望能建構一個系統，可使用手機作為人機介面，輸入魔術方塊當前狀態，透過演算法解題，進而控制馬達來解析。起初我們參考了網路上已有的6軸步進馬達裝置，以這樣的形式為原型來發展，希望降低運算與控制單元成本，在機構上使用LEGO積木為主體，並縮減為兩顆步進馬達配合二維控制。

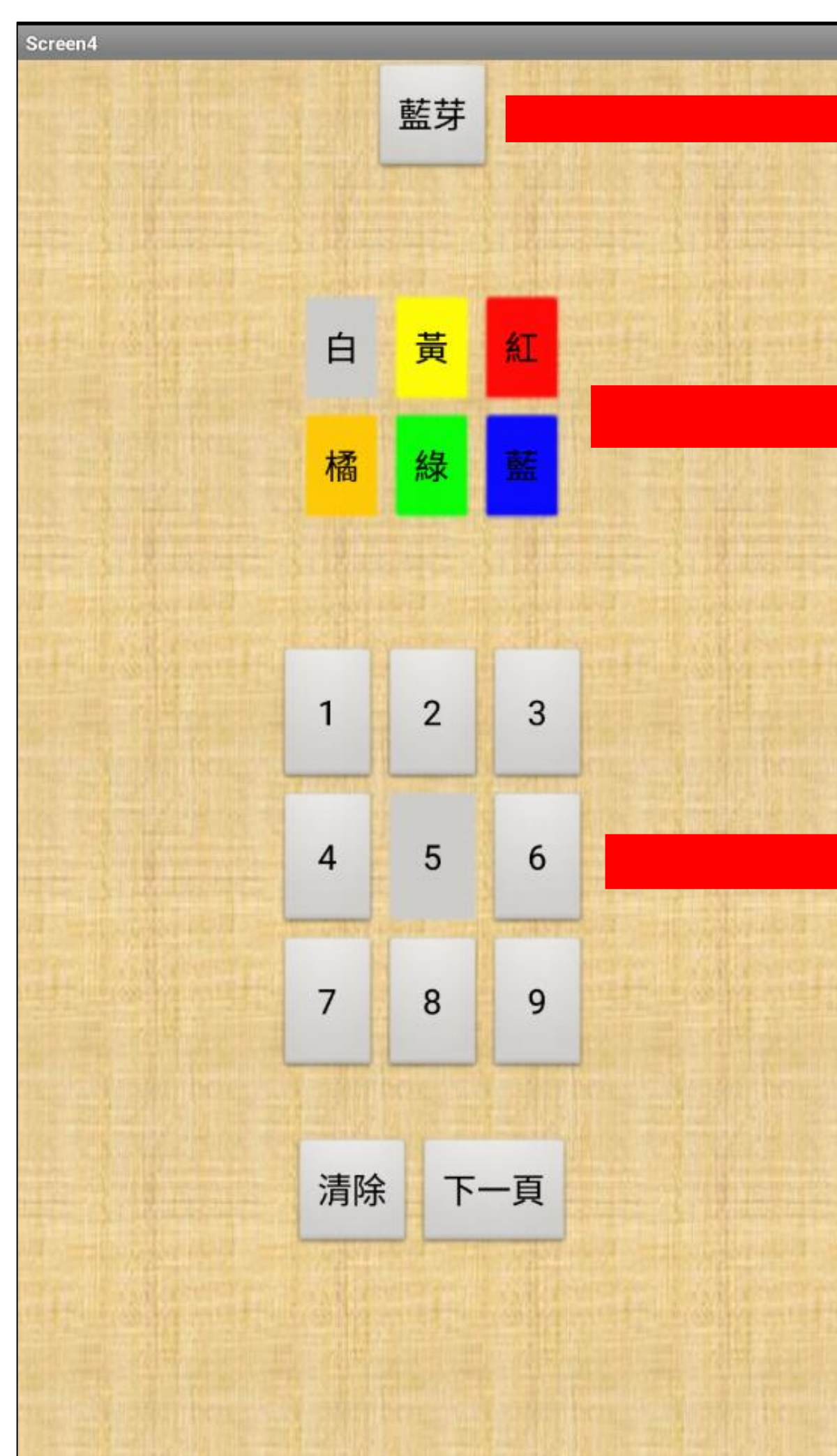


# 研究過程



▲系統架構圖

## 一、手機APP

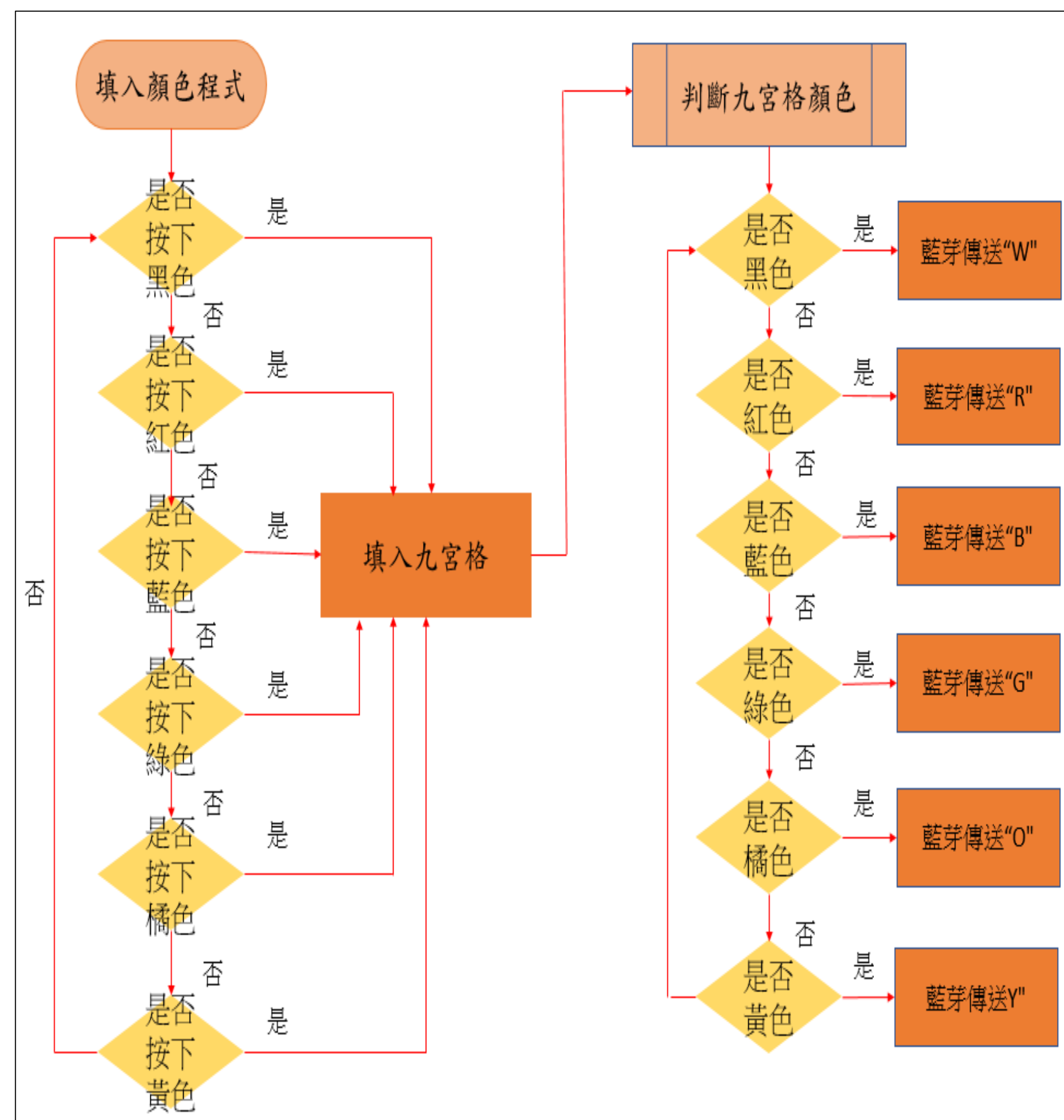


選擇HC-05藍芽位置

選擇要填入下方九宮格之顏色

第5格為白色，代表現在填入白色面的九宮格

▲人機介面說明



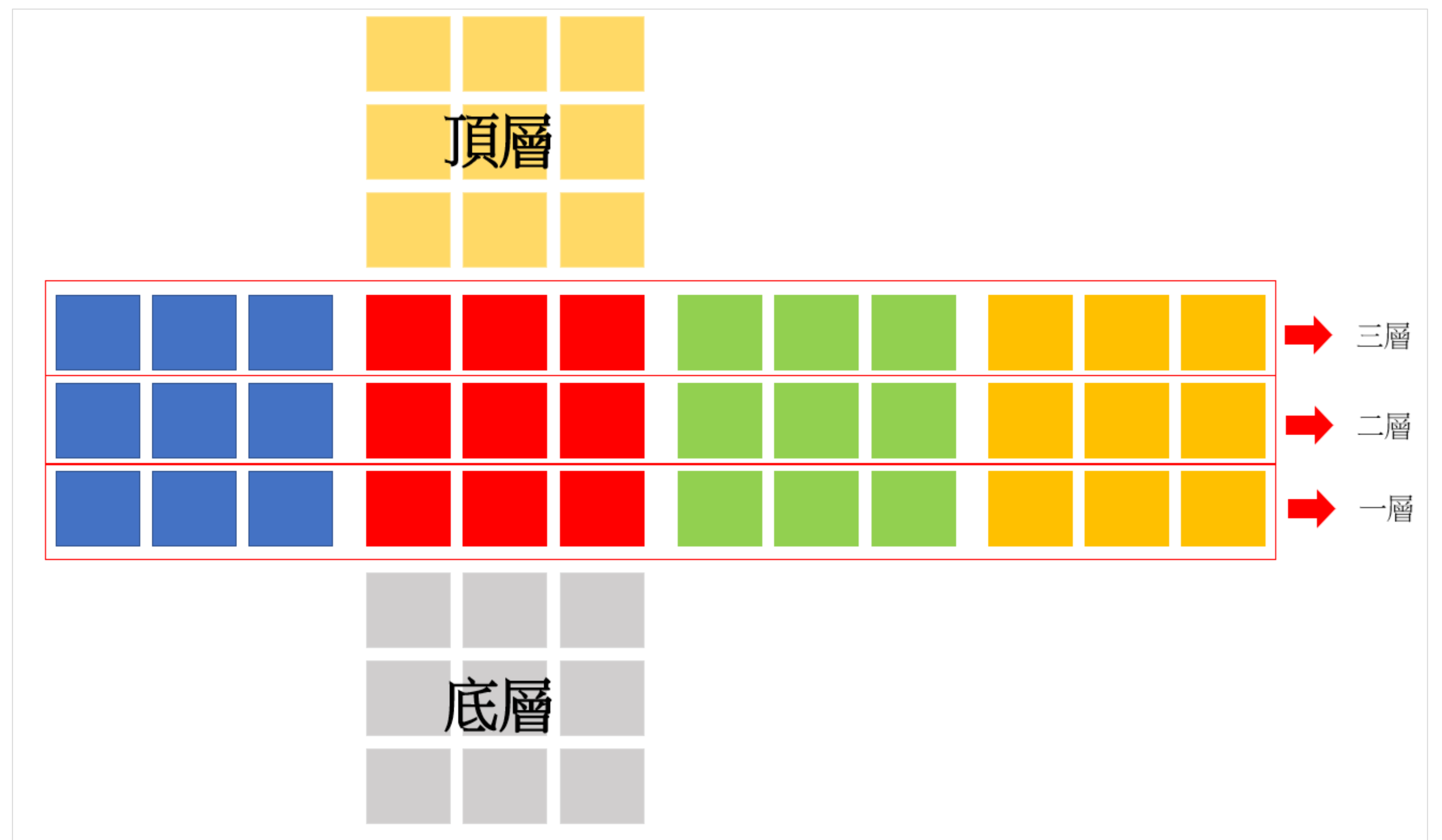
▲人機介面流程圖

本研究以App Inventor開發人機介面，玩家先點擊藍芽功能並選取HC-05藍芽通訊的位址後，依據已打亂魔術方塊的顏色，可在App裡空白格子中填入相對應的顏色，而clear鍵提供玩家重新填入之用。人機介面設計自行輸入相對應的顏色之目的，就是為了增加玩家對魔術方塊的趣味性與培養玩家空間概念的能力。

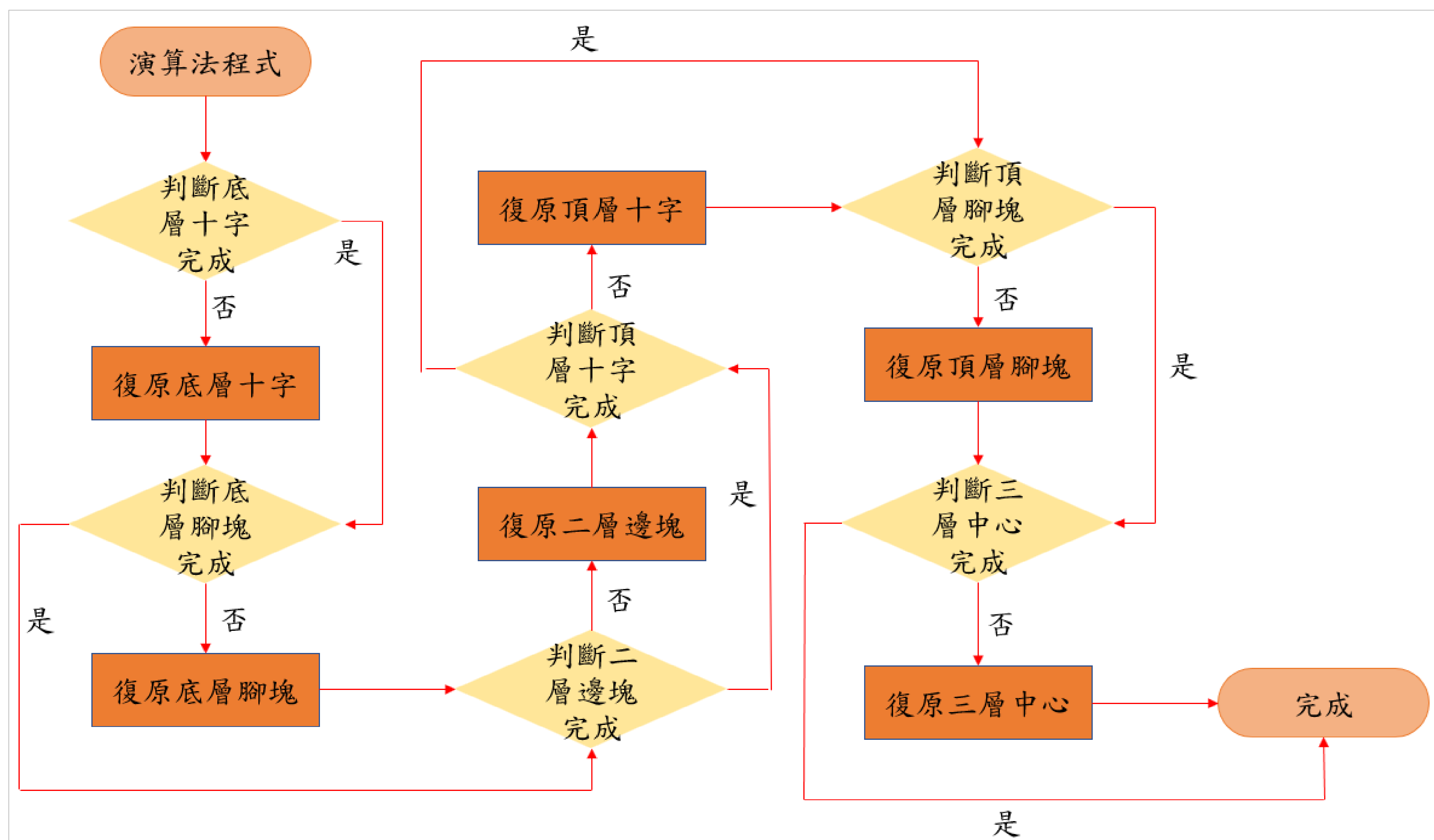
## 二、演算法

**CFOP**:是以四個步驟「Cross、F2L、OLL、PLL」原文的第一個字母合起來而成的。**Cross**是解決白色面的十字；**F2L**是第一層及第二層一次完成；**OLL**是頂層黃色面復原；則**PLL**是負責解決第三層邊塊部分。

**F2L**:當中的F2L由於有較多的組合，因此提升了程式撰寫難度，因此我們換了一個方式，我們延續**Cross**，並且先把白色面的腳塊復原後，再復原第二層邊塊，雖然這兩個運算後結果一樣，但F2L會較快些。

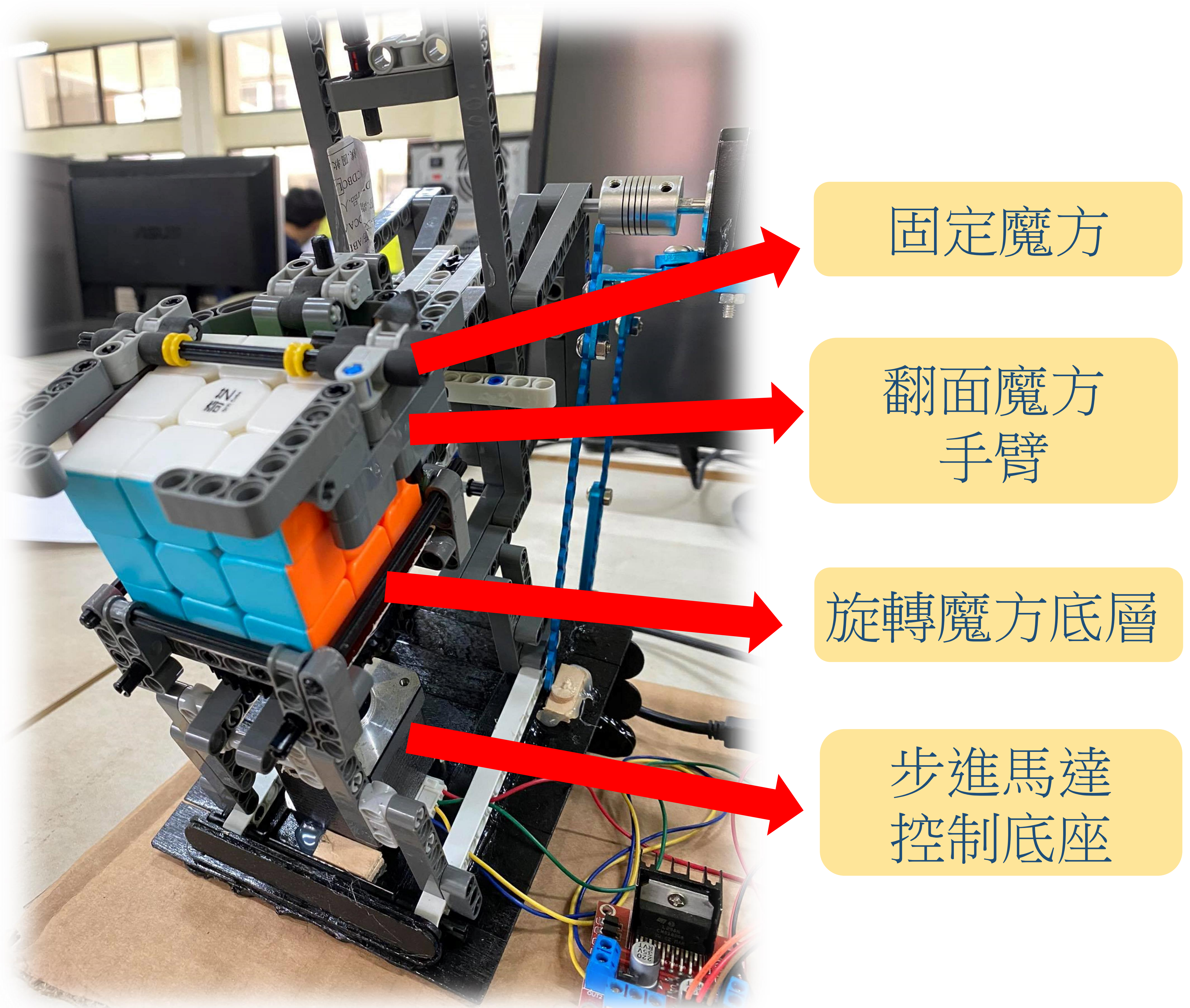


▲ 魔術方塊分層名稱

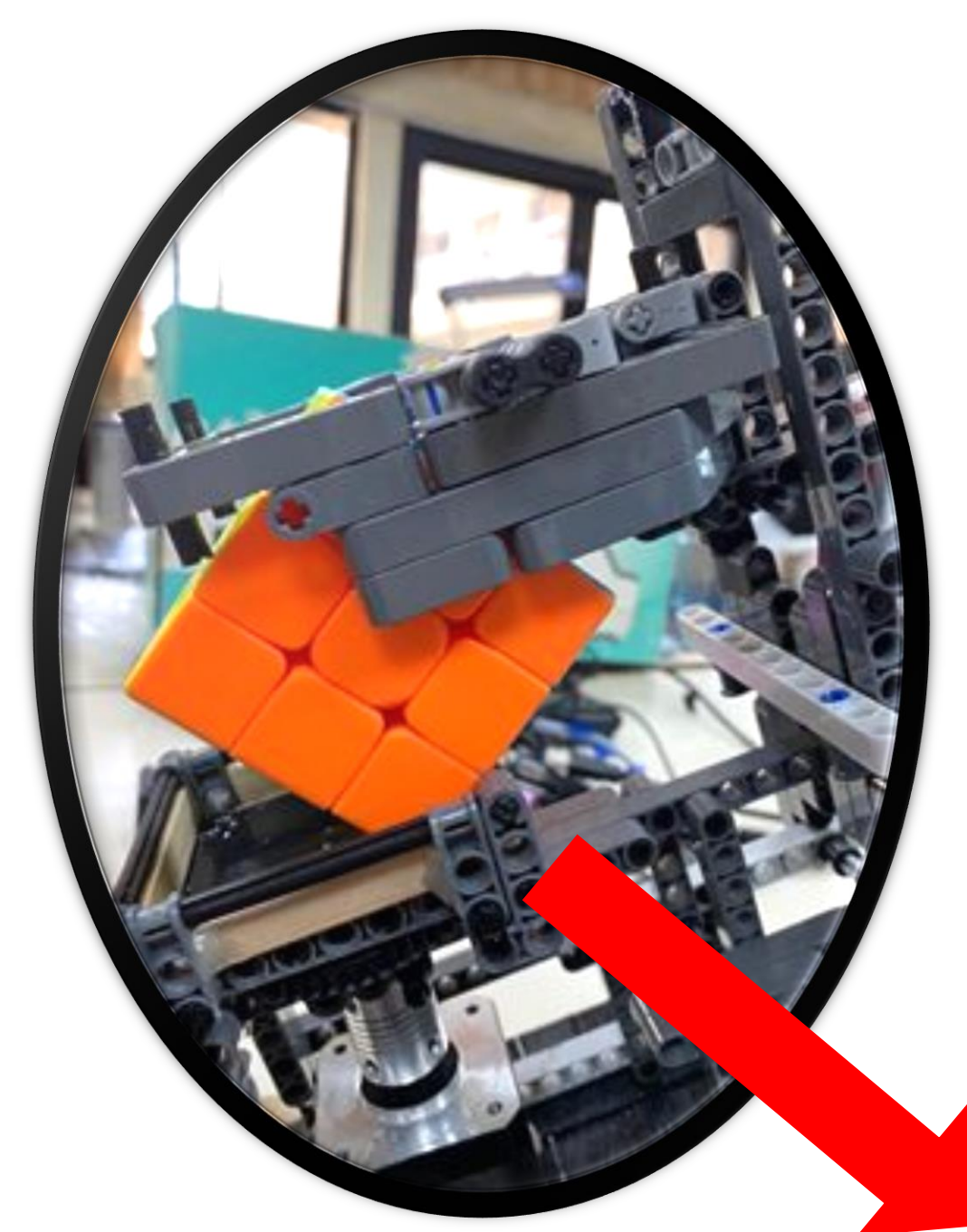


▲ 演算法流程圖

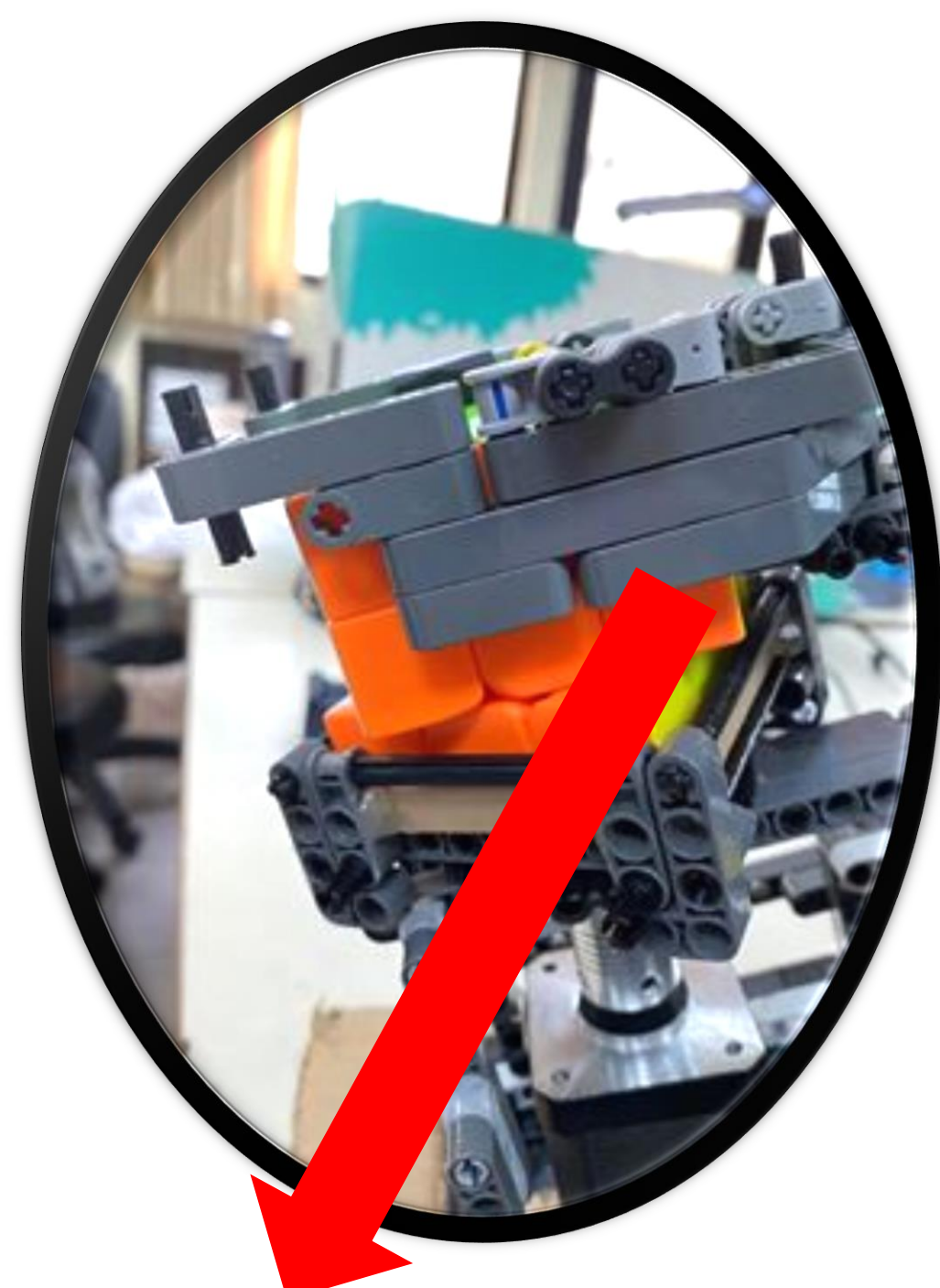
## 三、機械架構



當系統下達翻面指示時，機構執行命令，上方之翻轉步進馬達動作，帶動魔術方塊翻面手臂，藉著與底座後面的桿狀阻擋結構，做翻面的動作，並且因為我們的機構傾斜的設計，所以魔術方塊可依靠自身重力來完成翻面。

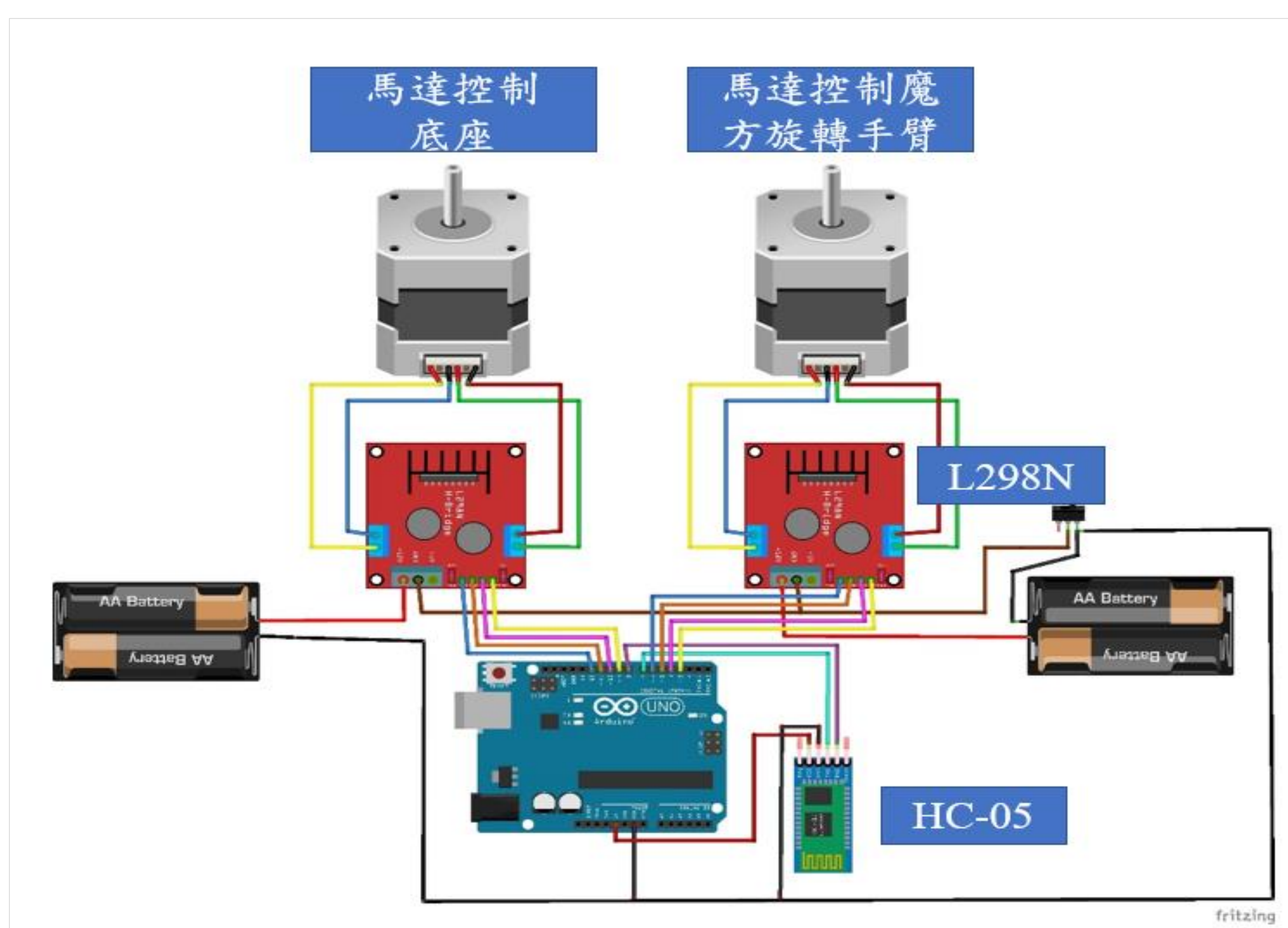


藉由此桿狀阻擋結構翻面

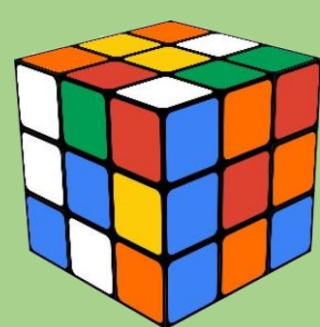


能夠遮蔽到二層

翻面時，翻面手臂可包覆到魔術方塊的第二層，讓底層做旋轉時，上面兩層會保持原本的圖形，不會因為旋轉底層而跑掉，藉此完成人們自行旋轉的操作。



▲控制電路示意圖



## 結論

本研究中，我們希望玩家透過系統，產生具有互動的學習，而非炫技的解題機器人，所以透過人們普遍有的智慧型手機，搭載自行開發的App作為人機介面，玩家可自行輸入、並填入畫面中魔術方塊相對應的顏色，目的是為了能讓玩家與系統進行溝通，產生想知道答案的興趣連結，進而增加觀察力以及專注力，激發他們動手嘗試的念頭。透過系統還原的步驟，一步一步慢慢摸索學習，訓練邏輯思考能力。此外，藉由本研究，將電機電子群三年內所學習的課程，像是程式設計、Arduino、單晶片實習、基礎電子實習等等，與研究做結合。透過研究，讓我們發揮自己的創造力，就像老師平常給我們練習的題目一樣，告訴我們，要按部就班地一一去測試完成，過程中遇到困難不是害怕逃避，而是要想辦法找出解決的方法，這是我們做研究時一直堅持的信念。