



# 2017 RoboCup 机器人世界杯中国赛

## 开发日志

队名：EMM-SOCCER

参赛单位名：澳門坊眾學校

队员：何建群 戴智龍 簡水源 林震

指导教师：韋建超

创建日期：

日志编号：01

活动日期：2016 年 6 月 25 日 - 2016 年 7 月 4 日

参与队员：何建群 簡水源

活动地点：學校

## 讨论过程

這次討論是我們在世界賽的時候發現有橘色球項目，為了應付這個項目，我們展開了一個討論。由於我們在 2016RCJ 世界賽已開展對 PIXY 的研究工作。而那時候我們的隊友還有黃懌靜師姐和郭濠強師兄，因此討論部分他們也參與其中。

日期：6 月 25 日

出席人員：郭濠強、黃懌靜、何建群、簡水源

內容：

黃：我昨天上官網發現，這次比賽新加了一個項目。

簡：甚麼？這都快比賽了，甚麼新項目啊？

黃：平常我們不是用 RCJ-05 調制球比賽的嘛，現在好像新增了一個橘色球，它不會發光。

何：不發光？那我們怎麼用復眼去檢測它

黃：我在規則上看到，好像使用攝像頭。

郭：我想他們的意思應該是利用攝像頭去追蹤球的顏色吧

黃：哦。我們要利用攝像頭去找橘色，從而判斷球在哪裏，對吧？

郭：我認為是這樣

何：可是，我們從來沒有用過攝像頭，該從哪裏入手啊？

簡：我認為我們應該先找攝像頭，然後再去想其他的吧。要是連最基本的攝像頭都沒有，還怎麼找球

黃：那我們該用甚麼攝像頭？

郭：我記得老師說過，舞蹈那邊曾用過攝像頭，當時還是讓我拿去給他們的呢

何：那我們去找他們問一下吧，看能不能借來用一用

郭：嗯

這次討論主要是說我們發現了橘色球這個新項目，準備用攝像頭去找球。

# 开发日志

## 开发及调试过程

2016/6/30 ~ 2016/7/4 為 RCJ 世界盃的舉辦日期。因為我們的隊伍在全國賽時拿下了冠軍，所以奪得了參加世界盃的資格。按照當時的比賽賽規，比賽用球仍是 RCJ-05 調制球，但另有一個橘色球作為附加的項目。不要求每隊參賽隊都參加這個附加賽，但若參加且勝出的隊伍，可獲加分。由於加分的緣故，我們都很重視這個項目。

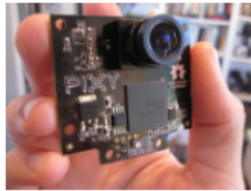
We are discussing general rules using the RCJ-05 electronic ball. Use of a passive orange ball will be tested and evaluated in games outside of the regular games. However, if teams wish to use vision-equipped robots during regular games, they should be allowed to do so.

A regular game could be played using the passive orange ball if both teams involved in the match are in agreement.

由於平常我們的機器人都是靠復眼去檢測 RCJ-05 調制球所發出的紅外光，從而判斷球的方向及距離。而新加的橘色球純粹是個普通的球，既不會發光，也不會發出甚麼信號讓我們接收，我們唯一可以做的，就是利用攝像頭，追蹤橘色的所在方位，從而判斷球的位置。



由於我們在出發去德國的前幾天才知道有這個附加項目，時間短促，而我們那時候對攝像方面也是一竅不通，情急之下我們跟老師說明了這個狀況，從老師口中得知機器人舞蹈小組那邊曾用過 Pixy(CMUcam5)攝像頭，於是便去找他們要來了 Pixy(CMUcam5)攝像頭。而我們也去了一些店鋪買來了橘色的玩具球充當着比賽球。

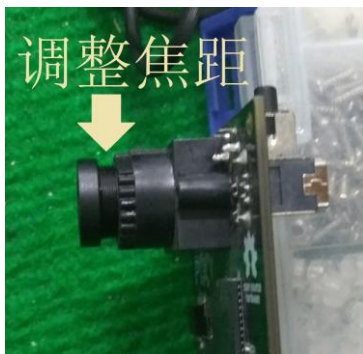


- Small, fast, easy-to-use, low-cost, readily-available vision system
- Learns to detect objects that you teach it
- Outputs what it detects 50 times per second
- Connects to Arduino with included cable. Also works with Raspberry Pi, BeagleBone and similar controllers
- All libraries for Arduino, Raspberry Pi, etc. are provided
- C/C++ and Python are supported
- Communicates via one of several interfaces: SPI, I2C, UART, USB or analog/digital output
- Configuration utility runs on Windows, MacOS and Linux
- All software/firmware is open-source GNU-licensed
- All hardware documentation including schematics, bill of materials, PCB layout, etc. are provided

有了攝像頭後，對它一無所知的我們當然去上網尋找一些有關 Pixy(CMUcam5)攝像頭的資料，並查看 Pixy(CMUcam5)攝像頭的使用方法。在網站上面有介紹到 PixyMon，PixyMon 是一個軟件，用來控制 Pixy(CMUcam5)攝像頭。除此之外，從網站上我們也學會了如何去調整 Pixy(CMUcam5)攝像頭的焦距，如何從電腦裏選取識別物件的顏色，以及如何修改 Pixy(CMUcam5)攝像頭對環境吸光量的大小等知識。

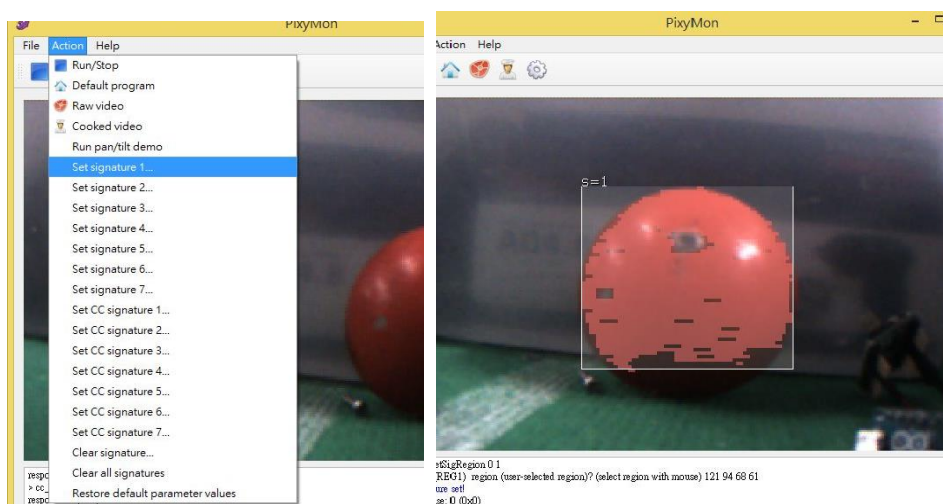
## ■ 調整 Pixy(CMUcam5)攝像頭的焦距

如下圖所示，Pixy(CMUcam5)的鏡頭是可以轉動的，用來對焦，就像普通相機的手動對焦功能一樣。



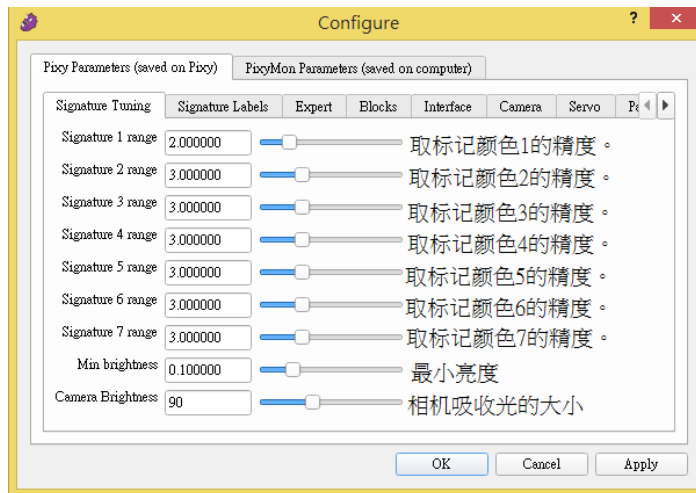
## ■ 從電腦裏選取識別物件的顏色

從電腦裏選取需要識別的顏色很簡單。按下 action 選項，然後便會看到 Set signature 1 到 Set signature 7，分別代表 7 個空間，可用來儲存七個顏色。選定其中一個後，再選取物件，便會記錄下物件的顏色。例如左圖，我按下 Set signature 1，然後選取橘色的部分，便會得到右圖的效果，記錄了橘色。仔細去看，右圖中間方框的左上角，有着 s=1 的字眼，意思就是方框裏的顏色為 Set signature 1 記錄的顏色。完成這幾個步驟，便能在電腦裏選取所需的顏色。



## ■ 修改 Pixy(CMUcam5)攝像頭對環境吸光量的大小

由於我們要準確地檢測球的顏色，因此 Pixy(CMUcam5)檢測顏色的精度必須要高，以免識別了其他不必要的顏色。比方說，我們要檢測橘色，若在精度不夠高的情況下，會連近似於橘色的紅色和黃色也一並檢測，這樣就很容易誤判了球的所在位置，因此我們需要在 PixyMon 裏修改 Pixy(CMUcam5)攝像頭對環境吸光量的大小。很簡單的，如下圖所示，打開 Pixy Parameters 界面，你會看到 Signature 1 range, Signature 2 range…… Signature 7 range，意思即標記顏色 1 到標記顏色 7 對環境的吸光量。選取數值越大，吸光量越大，精度相對會低；選取數值越小，吸光量越小，精度相對較高。

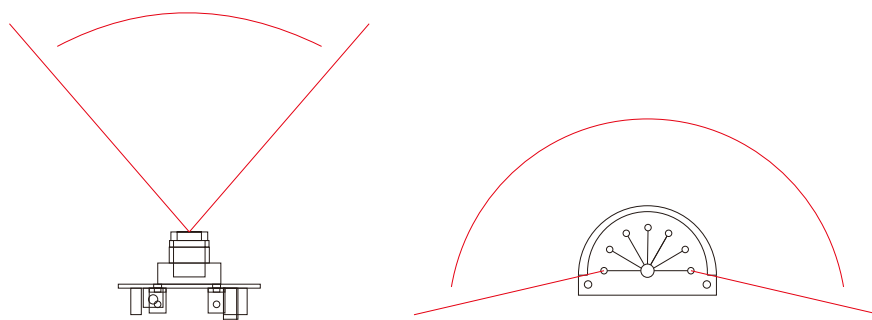


學會如何調整 Pixy(CMUcam5)攝像頭的焦距和如何準確捕捉我們想要的顏色這些基本知識後，要解決的問題就是如何把 Pixy(CMUcam5)攝像頭使用在機器人身上，如何讓機器人提取 Pixy(CMUcam5)攝像頭的數據。簡單來說，即要讓機器人控制器與 Pixy(CMUcam5)攝像頭進行通訊。但由於當時時間太短，我們還要準備世界賽的機器人以及面試，所以沒有足夠時間去進行深入研究，以致於我們最後用了一些不太正路的方法。我們知道當 Pixy(CMUcam5)攝像頭識別到球的時候，它前面的指示燈會亮，於是我們在指示燈上方加上一個灰度，用灰度來檢測 Pixy(CMUcam5)攝像頭的指示燈有沒有變亮，從而去知道 Pixy(CMUcam5)攝像頭有沒有識別到橘色球。由於我們用灰度來間接判斷有沒有識別到球，所以把灰度連接到機器人控制器上，就像平時一樣，這樣就能解決 Pixy(CMUcam5)攝像頭與機器人控制器通訊的問題。儘管這方法不太科學，但在當時也算是解決了燃眉之急。





硬件方面處理好後，我們把 Pixy(CMUcam5)攝像頭豎直裝在機器人的中板與底板之間，方便去檢測橘色球。裝好後，我們便着手程序的部分。因為我們加了灰度，所以程序也變得簡單起來。由於 Pixy(CMUcam5)攝像頭的檢測範圍有限，不像復眼一樣可以全方位去檢測，而我們現在只有一個攝像頭可用，所以會有很多盲點。我們的想法是，當機器人檢測不到球的時候，就回去場的中間，然後開始自轉，直到轉到球所在的方向，就直行走過去球那邊拿球。但我們認為只是衝過去這方案太過於單調並且沒有太大的攻擊性，所以我們萌生了一個想法，希望能夠利用盤球裝置去控住球，然後做一些動作，從而加強機器人的攻擊性。



從前我們用 RCJ-05 調制球的時候，由於球會發出紅外光，我們可在控球位上方裝一個火焰測量裝置，從而判斷球是否在控球位。但現在橘色球並不會發出紅外光，所以如何去判斷橘色球是否在控球位，是我們面臨的一個問題。再加上當球在機器人控球位的時候，球會遮擋住 Pixy(CMUcam5)攝像頭，這樣便會減弱光線，導致 Pixy(CMUcam5)攝像頭識別不到橘色球。因此，我們想出了一個方案。但在實踐的過程中，我們發現這個方案並不能解決問題，因此我們又想出了另一個方案。下面是我們想出的兩個方案，以及其好的地方、不好的地方：

- 方案一：

由於球在機器人控球位時會擋住 Pixy(CMUcam5)攝像頭，減弱光線令 Pixy(CMUcam5)攝像頭識別不到橘色球。因此我們想在 Pixy(CMUcam5)攝像頭附近加一些燈，從而解決光線不夠的問題。雖然這個方案能解決光線的問題，但並不能夠去判斷橘色球是否在控球位，所以我們放棄了這個方案。

- 方案二：

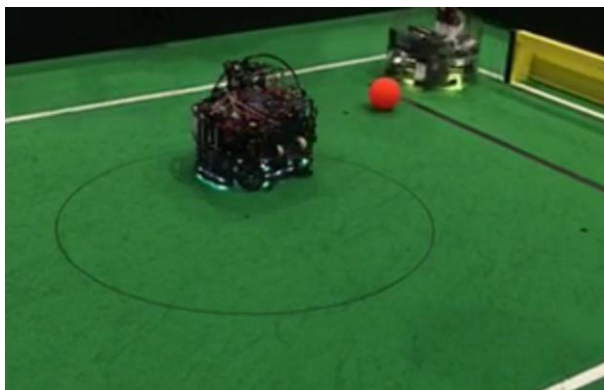
在回想方案一的時候，我們發現，我們的出發點是解決光線弱的問題，再去考慮如何讓機器人知道球是否在控球位。但正確的次序，應該是先解決了如何去判斷



球在控球位，再去想光線強弱的問題。因此，以判斷球在不在控球位為出發點，我們想出了第二個方案。我們在控球位附近多加了一個灰度，當橘色球接近控球位時，灰度的數值會發生變化，從而讓機器人知道球是否在控制位。解決了這個問題後，我們再去考慮光線的問題。我們發現，機器人已經能夠判斷球在不在控制位，因此光線不夠導致 Pixy(CMUcam5)攝像頭識別不到橘色球已經沒有關係。所以最後我們採用這個方案。

解決了如何判斷球在不在控球位後，便是盤球裝置的部分了。但我們想起，去年世界賽（中國）時，我們被裁判投訴盤球裝置速度過高，破壞球的表面，得了黃牌警告。這次我們亦害怕，若用盤球裝置控住橘色球，會破壞它的表面，導致歷史再現，再次獲黃牌警告。於是考慮到這個問題，再加上這只是一個加分賽，我們也沒有過多時間，最後我們便放棄盤球這個想法。用回最原始的，自轉找球，找到球後便直衝。

去到世界賽時，我們發現很多隊伍也和我們一樣，用 Pixy(CMUcam5)攝像頭來識別橘色球。而在加分賽期間，我們也向裁判證明了我們的機器人是有能力找到橘色球，對球有反應，並去追球。但由於我們對 Pixy(CMUcam5)攝像頭的使用也只是一知半解，且沒有作深入的研究，以致最後落敗。但與其他隊伍交流期間，我們發現他們的 Pixy(CMUcam5)攝像頭是利用 arduino 來直接與控制器進行通訊的，不像我們，利用灰度作為一個間接的溝通渠道。



# 开发日志

## 本次工作总结

- I. 這次對 PIXY 的研究，我們學會了調整 Pixy(CMUcam5)攝像頭的焦距，在 PixyMon 軟件裏選取識別物件的顏色，以及修改 Pixy(CMUcam5)攝像頭對環境吸光量的大小等知識。
- II. 在通訊方面，由於我們時間不足，而且當 Pixy(CMUcam5)攝像頭識別到球的時候，它前面的指示燈會亮，所以我們便直接在指示燈上方加一個灰度，用來檢測指示燈有沒有亮，從而知道有沒有檢測到球。
- III. 我們在控球位附近加了一個灰度，當橘色球靠近控球位，灰度的數值會隨之發生變化，我們便可知道球有沒有在控球位。
- IV. 策略：當機器人看到球，便往球的方向衝。當機器人看不到球，便自轉。由於顧慮盤球裝置會破壞球的表面，因此我們不使用盤球裝置。
- V. 在世界賽時，發現可用 Arduino 來作為 PIXY 與機器人控制器的溝通橋樑。
- VI. 可改善的地方：想出比自轉找球更好的方案，可考慮使用 Arduino 來溝通

日志编号：02

活动日期：2016 年 9 月 3 日 - 2016 年 9 月 18 日

参与队员：何建群 戴智龍 簡水源 林震    活动地点：學校

## 讨论过程

經過了一個暑假，我們又要開始工作，準備澳門區的選拔賽。為了分工，我們開了一個會。

日期：9 月 3 日

出席人員：何建群、戴智龍、簡水源、林震

內容：

何：我們今天開會的目的主要是讓大家從假期回歸到工作的狀態。暑假已經放完，我們要開始工作了。

林：一眨眼，郭濠強師兄和黃懌靜師姐都畢業了。現在重量組只剩下我們四個人了。

戴：對啊，真是光陰似箭，日月如梭啊！

何：現在四個人，那我們就兩兩分組？

簡：好呀！我想跟你（何建群）一組，我們比較有默契。

何：那麼巧，我也想和你一組耶！

林：既然如此，那我就和戴智龍一組吧。

何：好，那麼現在就確定了我和簡水源一組，林震跟戴智龍一組。沒有問題吧？

戴：沒有。

簡：OK

林：我也沒問題

何：那好，就這麼決定了。那麼，今年我們用甚麼策略呢？一攻一防還是雙攻？

簡：我個人認為繼續雙攻吧。去年我們也是雙攻的策略，整體效果也不錯，很有攻擊力。你覺得呢？

何：我認為雙攻也有雙攻的缺點。不過世上沒有事情是十全十美的，我們能做的就是盡量把不完美的地方都改掉。既然去年我們用的是雙攻的策略，而你也認為雙攻比較好，不如就一用到底，務求把雙攻的程序寫到完美吧！

簡：好

何：那你們呢（林震、戴智龍）？

戴：我今年第一年加入重量組，甚麼經驗都沒有。不過去年我在輕量組的時候用的是雙攻的策略，感覺很有力量。不過在重量組，只怕盤球的時候球甩了出來，導致烏龍球。

林：你說的有道理。不同的方案都有它的優缺點。

戴：我那麼多年都是做攻擊機，今年應該也不例外。就看你做防守機還是攻擊機了。

林：嗯…我去年做的是防守機，不過做的不好，有很多不足的地方。要不…我先試着做攻擊機，要真達不到預期的狀態，我再繼續研究我的防守機？

戴：也沒問題呀，可能試着試着會發現你比較適合做攻擊機呢

簡：如果這樣的話，林震你可要保留你現在的防守機，不要拆了啊。

林：嗯嗯。

何：好，那麼現在很清楚了，兩隊都採用雙攻的策略。

簡：四台攻擊機，想想也有點興奮

何：好了好了，先把機做好再說吧。

簡：既然決定了分組以及來年的方向，那今天的會不如就到這裏吧

戴：等等，我突然想起，我和林震還有另一個學術比賽，這段時間要去訓練，恐怕不能專注在機器人上了。

# 开发日志

林：對喔! 差點忘記了。我們應該要忙到十月中旬才有時間吧。

何：既然如此，你們忙你們的吧，剛好我也想繼續研究 pixy。

簡：好啊，我陪你一起。

林：那麼在這段時間裏，我們盡量抽出時間來做機器人，等到我那邊忙完了，就專注回機器人上面。

何：好，那我和簡水源就研究 pixy。那麼今天的會就到這裏吧，十月份的時候我們再開一個會，各自匯報一下進度。

戴：好的

簡：沒問題

林：no problem

這次會議，我們確定了分組、來年的策略以及分工：

- 何建群、簡水源一組，雙攻，研究 PIXY
- 戴智龍、林震一組，雙攻，先訓練另一個比賽，有時間就做機器人

下次開會時間：十月

內容：匯報進度

# 开发日志

## 开发及调试过程

### 何建群、簡水源同學

世界賽的時候，我們學會了如何調整 Pixy(CMUcam5)攝像頭的焦距、如何從電腦裏選取識別物件的顏色、如何修改 Pixy(CMUcam5)攝像頭對環境的吸光量等知識，利用灰度作為 Pixy(CMUcam5)攝像頭與機器人控制器的溝通渠道。當 Pixy(CMUcam5)攝像頭看不到球時機器人便自轉，直到轉到看到球了，就往球的方向直衝。從比賽中，我們發現別的隊伍是用 arduino 作為 Pixy(CMUcam5)攝像頭與機器人控制器的溝通橋樑，讓他們直接通訊。

所以現在有時間，我希望能繼續研究 Pixy(CMUcam5)攝像頭與機器人控制器的通訊問題。研究期間，我在網上發現有一款攝像頭，秉火 OV7725。秉火 OV7725 的價格是 78 人民幣，而對比起來，Pixy(CMUcam5)攝像頭要 485 元，多於 6 倍的差異。由於價格相差甚多，於是我上網尋找更多關於秉火 OV7725 的資料，看看秉火能否取代 Pixy(CMUcam5)。以下是秉火 OV7725 的一些資料：

- 處理器：NXP LPC4330，204 MHz，雙核
- 透鏡視野：75 度水平，47 度垂直
- 鏡頭類型：標準 M12（提供多種不同類型）
- 功耗：典型值為 140mA
- 電源輸入：USB 輸入（5V）或非穩壓輸入（6V 至 10V）
- 可用數據輸出：UART 串行，SPI，I2C，USB，數字，模擬
- 尺寸：2.1 X 2.0 X 1.4
- 重量：27 克

## 硬件参数

型号 OV7725 30W像素摄像头模块

传感器 OV7725、30W像素、彩色

电路设计 5V供电、板载12M有源晶振、384KB FIFO

接口 2\*10P DIP接口 或者 FPC长排线

镜头 3.6mm M12 镜头、光圈F2.0、视角95°、带红外滤光片

镜座 外框21\*21\*10mm，内框17.7\*17.7\*6mm

程序 提供STM32F103驱动程序、8位并行接口  
因为单片机速度不够快，必须通过FIFO缓存数据

数据格式 8bit、RGB565格式

实时显示 320\*240分辨率，RGB565格式，液晶实时显示帧率24帧，提供源码

**注意** OV7725本身是30万像素，即 640\*480的分辨率，但是STM32不够快，必须加一个FIFO来缓冲，这个时候就只能采集320\*240的分辨率



經過資料的對比，我們發現 Pixy(CMUcam5)比秉火 OV7725 更好，原因有以下幾點：

- 秉火 OV7725 只能使用它配制的控制器，不能與其他控制器通訊
- 秉火不知道有沒有 PC 用的軟件去控制，而 Pixy(CMUcam5)有 PixyMon
- 秉火 OV7725 不像 Pixy(CMUcam5)內部會做一些普通的處理
- 秉火 OV7725 的單片機運算速度很慢

綜合上述各點，秉火 OV7725 對我們來說很不方便，我和簡水源一致認為繼續使用 Pixy(CMUcam5)攝像頭。

放棄了秉火 OV7725 後，我們便繼續研究 Pixy(CMUcam5)與控制器的通訊問題。我們發現，Pixy(CMUcam5)可以兼容多種的數據輸出方式，例如 Arduino ICSP SPI、SPI with SS、I2C、analog/digital x、analog/digital y、LEGO I2C 等。這才知道，原來輸出物體在圖像上的橫坐標和縱坐標可以用模擬量作輸出，我頓時感嘆自己在世界賽的無知。

但它也衍生出一個問題：它只能輸出一個坐標值，意思就是要嘛只輸出橫坐標的值，要嘛只輸出縱坐標的值，魚與熊掌，不可兼得，因此我們決定用其他的輸出方式來提取數據。由於我們用的是中鳴的控制器，所以可直接否決了 LEGO I2C 的輸出方式，剩下的只有 Arduino ICSP SPI、SPI with SS 和 I2C 了。由於我們並不了解 SPI with SS 和 I2C 的輸出方式是如何在中鳴的控制器讀取數據，加上中鳴和 I2C 的協議是不同的，因此最後我選擇了 Arduino ICSP SPI。



# 开发日志

我在

[http://cmucam.org/projects/cmucam5/wiki/Hooking\\_up\\_Pixy\\_to\\_a\\_Microcontroller\\_\(like\\_an\\_Arduino\)](http://cmucam.org/projects/cmucam5/wiki/Hooking_up_Pixy_to_a_Microcontroller_(like_an_Arduino))

網站尋找了 Pixy(CMUcam5)在 Arduino 的使用方法以及開源例程，我們開始研究 Arduino 如何實現把 Pixy(CMUcam5)的數據反饋給中鳴控制器。我們找來的 Arduino 板是 Arduino Micro 和 Arduino Nano，因為它們的體積小且質量小。但在價格方面，Arduino Micro 要 195 人民幣，而 Arduino Nano 只要 16 人民幣。再加上我們用 Arduino 只是為了用作一個溝通的橋樑，不需要太高的規格，因此最後我們選擇用 Arduino Nano。

後來我們發現，原來舞蹈那邊購買了 Arduino Micro 板，於是我們便使用現成的。我們把 Pixy(CMUcam5)直接接在 Arduino Micro 板上，Arduino Micro 板上有個專門用來接 Pixy(CMUcam5)的接口。



把 Pixy(CMUcam5)接到 Arduino Micro 板上之後，我們研究了一段代碼，把 Pixy(CMUcam5)讀取到的數據在監視器上顯示出來。代碼部分主要是讀取標記顏色的  $x$ 、 $y$  值，以及讀取所識別到的色塊的長度和寬度。然後我們再新建一些變量去裝這些數值，以供我們使用。把我們定義的變量顯示在監視器上，就可以知道色塊的坐標以及大小了。

```
if (blocks)
{
    i++;
    // do this (print) every 50 frames because printing every
    // frame would bog down the Arduino
    if (i%50==0)
    {
        sprintf(buf, "Detected %d:\n", blocks);
        Serial.print(buf);
        for (j=0; j<blocks; j++)
        {
            //Debug message
            sprintf(buf, "  block %d: ", j);
            Serial.print(buf);
            pixy.blocks[j].print();
            //
            if (pixy.blocks[j].signature == 1) //orange or not
            {
                if (pixy.blocks[j].height > height) //the block is bigger than the record one
                {
                    x = pixy.blocks[j].x;
                    y = pixy.blocks[j].y;
                    height = pixy.blocks[j].height;
                    width = pixy.blocks[j].width;
                }
            }
            Serial.println((int)x);
            Serial.println((int)y);
            Serial.println((int)height);
            Serial.println((int)width);
        }
    }
}
```

知道色塊的坐標以及大小之後，就是如何把這些數據返回給機器人控制器的問題了。我們需要利用 I2C 作為中鳴控制器與 Arduino 的溝通橋樑，但我們在這方面懂的知識不是很多，所以我們去請教了已畢業的師兄，得知我們需要寫一段代碼，才能把數據反饋給機器人控制器。在師兄的指導之下，我和簡水源一同寫出了代碼。下面我將簡單介紹一下這段代碼。

## ■ 引用庫

在程序的一開始，我們要把幾個必要的庫引到程式裏面。分別有<Wire.h>、<math.h>、<SPI.h>和<Pixy.h>。

```
#include <Wire.h>
```

```
#include <math.h>
```

```
#include <SPI.h>
```

```
#include <Pixy.h>
```

- <Wire.h>是 Arduino 與 I2C 溝通的一個必要的庫。由於我們需要 I2C 作為 Arduino 與機器人控制器的一個溝通橋樑，所以必須有<Wire.h>這個庫。
- <math.h>是數學運算庫。由於我們有對 PIXY 的數據做了一些處理，需要數學運算，因此要有<math.h>庫。

- <SPI.h>庫和<Pixy.h>庫用來與 PIXY 溝通。由於需要拿取 PIXY 的數據，因此必須與 PIXY 進行溝通。

## ■ 定義變量

引入所需要的庫後，我們就定義我們所需要的變量。可以看到在這段程序的開頭部分，有幾個變量的類型是 unsigned long，這是因為我們需要計時，所以選用了容量大的變量類型。然後下面再定義其他所需的變量。

```
Pixy pixy;  
  
unsigned long starttime;  
  
unsigned long stoptime;  
  
unsigned long looptime;  
  
int detected;  
  
int s = 0, sr = 0;  
  
int jr = 0;  
  
int x, y, width, height;  
  
int mode;
```

## ■ 設置

定義變量後，我們便要設置主機和從機之間傳輸溝通時所要用到的函數。其中，主機指的是 RCU，從機指的是 Arduino。

Wire.onReceive(receiveEvent); // 從機接收主機發送過的數據，之後要執行對應的子程序，所需要調用的函數。

Wire.onRequest(requestEvent); // 主機請求從機發送數據，之後要執行對應的子程序，需要調用的函數。

```
void setup()
```

```
{  
    Wire.begin(1);           // join i2c bus with address #1  
    Wire.onReceive(receiveEvent); // register event  
    Wire.onRequest(requestEvent); // register event  
    pixy.init();  
}
```

## ■ 運算處理

之後，我們需要對一些數據（顏色、位置等）進行運算及處理，並判斷數據歸零的問題，以返回給機器人控制器。

```
void loop()                                if (s > sr)  
{                                           {  
    static int i = 0;                        jr = j;  
    int j;                                  sr = s;  
    uint16_t blocks;                        x = pixy.blocks[jr].x;  
    char buf[32];                           y = pixy.blocks[jr].y;  
    blocks = pixy.getBlocks();              width = pixy.blocks[jr].width; //1-320  
    if (blocks)                             height = pixy.blocks[jr].height; //1-200  
    {                                       x = x*5/8;  
        i++;                               width = width*5/8;  
        if (i % 1 == 0)                   }  
        {                                   }  
        for (j = 0; j < blocks; j++)      //sprintf(buf, " X= %d:Y=%d ",  
        {                                  pixy.blocks[jr].x , pixy.blocks[jr].y);  
            s = pixy.blocks[j].width *    //Serial.println(buf);  
            pixy.blocks[j].height;        starttime = millis();  
        }  
    }  
}
```

```
sr = 0;                                detected = 0;
jr = 0;                                }
}                                       else{
}                                       detected = 1;
stoptime = millis();                  }
looptime = stoptime - starttime;       Serial.println(x);
if(looptime>=25){                      }
```

## ■ 數據執行

這段代碼參考了 <https://www.arduino.cc/en/Tutorial/MasterWriter> 所寫的代碼來修改，內容為：每當從主機接收到的數據執行，來讀取接收到的字串（mode）。

```
// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void receiveEvent(int howMany) {
  while (1 < Wire.available()) { // loop through all but the last
    char c = Wire.read(); // receive byte as a character
    Serial.print(c);       // print the character
  }
  int x = Wire.read();     // receive byte as an integer
  Serial.println(x);       // print the integer
}
```

```
void receiveEvent(int howMany)
{
  while (Wire.available())
  {
    mode = Wire.read(); // receive byte as an integer
  }
}
```

## ■ 反饋數據

這段代碼則參考了 <https://www.arduino.cc/en/Tutorial/MasterReader>，當數據通過主設備 Arduino 請求執行，即反饋數據給機器人控制器。**Detected** 是用來判斷 PIXY 有沒有識別到球，再判斷 **mode** 的字串，返回相對應的數值。例如當 **mode = 1** 時，會把 **x** 的值返回給機器人控制器。

```
// function that executes whenever data is requested by master  
// this function is registered as an event, see setup()  
void requestEvent() {  
  Wire.write("hello "); // respond with message of 6 bytes  
  // as expected by master  
}
```

```
void requestEvent()  
{  
  if(detected==1){  
    switch (mode) {  
      case 1: Wire.write(x); break;  
      case 2: Wire.write(y); break;  
      case 3: Wire.write(width); break;  
      case 4: Wire.write(height); break;  
      default: Wire.write(0); break;  
    }  
  }  
  else{  
    Wire.write(0);  
  }  
}
```

# 开发日志

## 本次工作总结

- I. 這次工作，我們對比了秉火 OV7725 與 Pixy(CMUcam5)的一些基本資料，發現 Pixy(CMUcam5)的價格比秉火 OV7725 高出許多，但秉火 OV7725 只能夠使用它自身配制的控制器，內部不會做處理，且單片機運算速度很慢。鑒於各種缺點，我們決定繼續使用 Pixy(CMUcam5)。
- II. 選用 Arduino ICSP SPI 方式來輸出數據
- III. Arduino Micro 板和 Arduino Nano 板的體積小、質量小。但前者要 195 人民幣，後者只需 16 人民幣，因此決定使用 Arduino Nano。但由於舞蹈小組已有 Arduino Micro 板，所以最後用了 Arduino Micro 板。
- IV. 為知道 Pixy(CMUcam5)識別到的色塊的大小及位置，我們研究了一段代碼，去讀取標記顏色的 x、y 值，再把它們顯示在監視器上，從而把 Pixy(CMUcam5)讀取到的數據在監視器上顯示出來。
- V. 利用 I2C 作為中鳴控制器與 Arduino 的溝通橋樑。與師兄一起研究程序，把數據反饋給機器人控制器。



日志编号：03

活动日期：2016 年 9 月 19 日 - 2016 年 9 月 30 日

参与队员：何建群 簡水源

活动地点：學校

## 讨论过程

由於我們認為以前只有一個 Pixy(CMUcam5)攝像頭的時候十分不方便，需要不停地自轉去找球，因此，我們希望能找出另一個更好的方案。

日期：9 月 19 日

出席人員：何建群、簡水源

內容：

何：我覺得世界賽的時候我們只裝一個 PIXY 真的很不方便，盲區又大，然後機器人只能一直自轉去找球。

簡：我也覺得是，好像傻子在轉圈哦。

何：對啊。而且自轉耗費時間，很多時候，對手的機器人都在我們自轉的期間拿找到了球，等到我們找到球的時候，別人已經快到龍門了。

簡：這的確是一個很大的問題，到底我們該用甚麼策略呢？

何：假如我們不自轉，pixy 可測範圍只有 75 度，要如何去找球？

簡：不自轉的話... 那我們在場上走？

何：甚麼意思？

簡：當我們看不到球的時候，就按照一個路線走，比方說，機器人順着場上的白線走，走的時候始終面向場內，那總會看得到球吧？

何：你說的沒錯，那不如我們試試吧。

簡：好。順着線走，應該比自轉要好。

何：欸，等等，順着線走的話... 能夠解決盲區大的問題，但.....好像一樣會花太多的時間。

簡：對哦，那還是不行啊。

何：到底該用甚麼策略呢？要是軟件不行，那硬件呢，有沒有方法？

簡：硬件... 硬件的話... 啊，一個 pixy 會有盲區，但如果我們用多幾個 pixy，不就能解決盲區的問題了。

何：對啊。之前我們時間太趕，只有一個。可是現在時間充裕，我們可以多買幾個。

簡：然後把它們圍在一起，就像復眼一樣，這樣就能做到全方位檢測了。

何：嗯。這樣一來，就不會再像個傻子一樣亂走亂轉了。

簡：那...我們要用幾個？

何：一個 pixy 可測 75 度，四個 300 度，我們至少也要用五個。

簡：把它們圍成一個正五邊形，那就可以了吧。

何：嗯，那我們趕緊去買吧。早買早試。

簡：好。

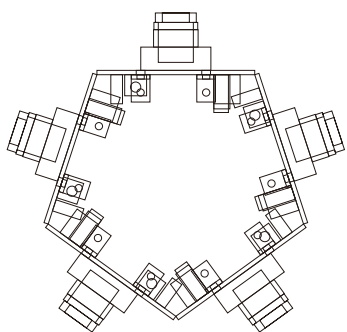
這次討論主要是找出有甚麼好的方案去找球，以取代之前的自轉找球。最後我們決定了使用 5 個 Pixy(CMUcam5)攝像頭來檢測球，把 5 個 Pixy(CMUcam5)攝像頭圍成正五邊形，以解決有盲區的問題。

## 开发及调试过程

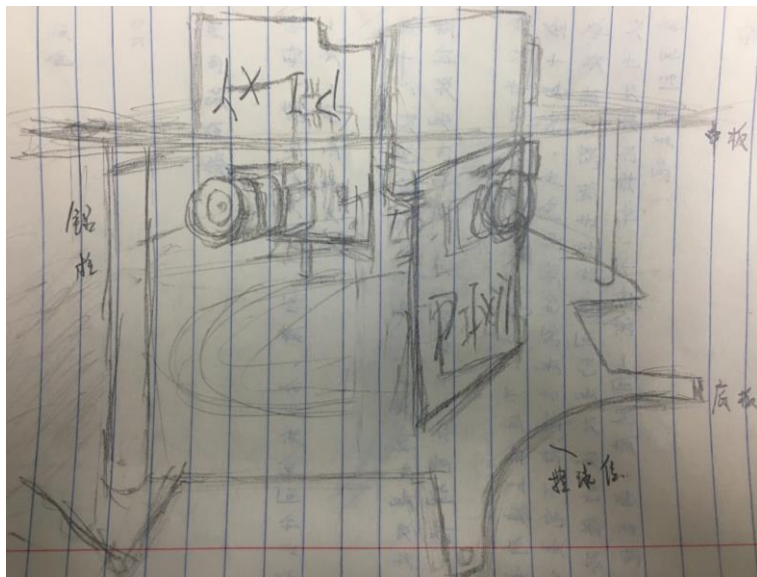
解決了 Pixy(CMUcam5)與中鳴控制器的溝通問題，成功把所識別到的色塊的  $x$ 、 $y$  值以及長度寬度返回給機器人控制器後，我們開始把 Pixy(CMUcam5)安裝到機器人上進行實踐。

Pixy 所能看到的範圍是 75 度，而我們平常用兩個復眼合起來有 360 度。因此，在我們六月份研究 pixy 的時候，用的策略是當機器人看不到球就自轉，轉到看到球後便往球的方向直走。我們認為，機器人看不到球的時候，一味自轉，有點愚蠢，且極有可能在自轉的過程中被別的機器人找到了球。所以，我們希望做出一部能夠檢測全方位的機器人。

我們想模仿復眼，把幾個 Pixy(CMUcam5)攝像頭合起來，以檢測各個方位。由於一個 Pixy(CMUcam5)攝像頭可測角度為 75 度，所以我們至少需要 5 個 Pixy(CMUcam5)攝像頭。把它們圍成一個正五邊形，便可以實現我們的想法了。



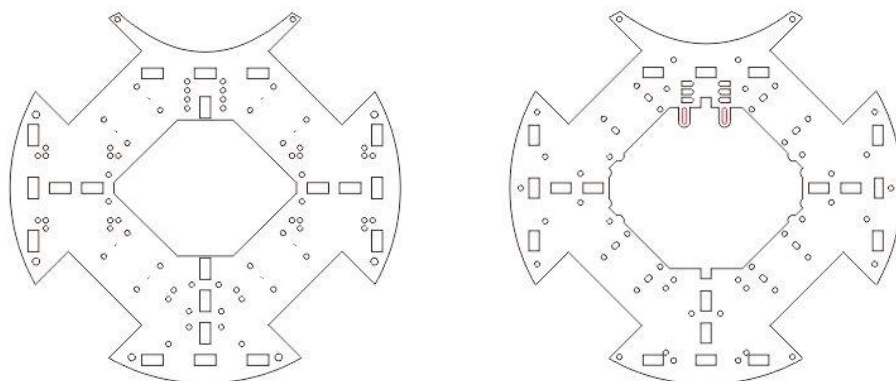
由於我們決定使用五個 Pixy(CMUcam5)攝像頭，若是像從前一樣，把它們放在底板上，那麼除了前面那個攝像頭外，其餘四個都會與馬達位置有衝突。但若我們把它們都放在中板上，那麼攝像頭就會離地面比較遠，過高以致看不到球。所以我們想把那四個攝像頭放在中板與底板之間，讓攝像頭穿過中板。在上面的四個攝像頭鏡頭向下，在底板上的攝像頭鏡頭向上，讓五個攝像頭鏡頭的高度一致。



想好方案後，我們使用 CorelDRAW 軟件畫圖，實現我們的想法。我們以舊的機器人的設計圖作為藍本，在上面修改和新增一些孔來安裝 Pixy(CMUcam5)攝像頭。

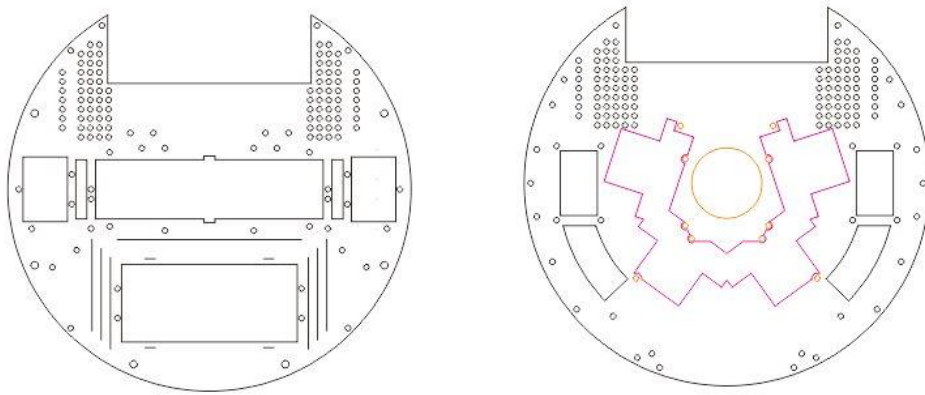
## ■ 底板

因為底板要安裝一個 Pixy(CMUcam5)攝像頭，所以下圖（右）中紅色部分為修改了的孔，用來安裝 Pixy(CMUcam5)攝像頭。



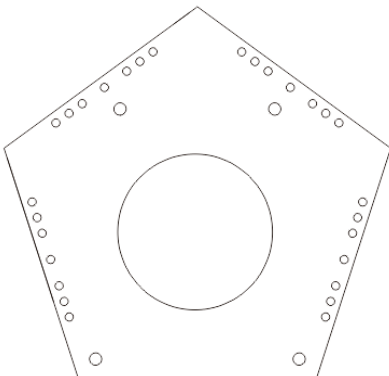
## ■ 中板

中板要裝四個攝像頭。下圖中紫紅色部分是用來安放四個攝像頭的。而中間橙黃色的圓圈是用來讓 Pixy(CMUcam5)攝像頭的線穿過中板，連接到 Arduino。

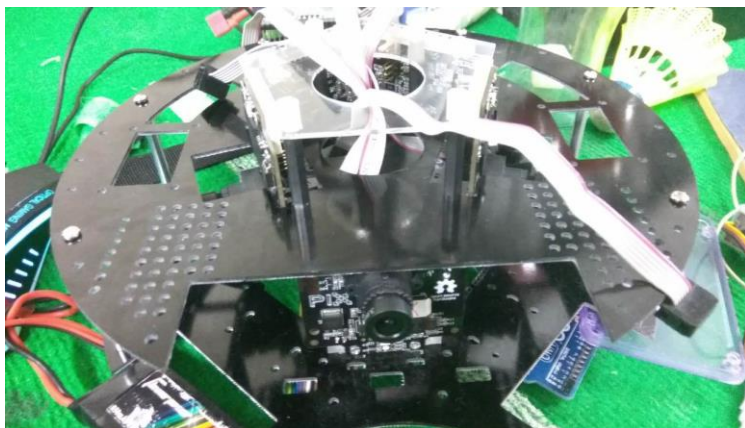


## ■ 輔助板

由於我們要讓四個攝像頭穿過中板，因此必須要有一個東西作為連接。所以我們設計了一塊板，用來連接固定四個攝像頭。



下面是機器人基本的樣子：



# 开发日志

## 本次工作总结

### I. 硬件上的轉變：

舊：機器人只有一個 Pixy(CMUcam5)攝像頭，因此會有很大的盲區。

新：機器人裝有五個 Pixy(CMUcam5)攝像頭，把它們圍成一個正五邊形，解決了盲區的問題

### II. 硬件擺放：

由於攝像頭（除了前面那個）的位置會與馬達重疊，因此不能放在底板上。所以我們就把四個攝像頭穿過中板，且鏡頭向下，與前面鏡頭向上的攝像頭保持一致的高度。

### III. 設計：

我們以舊的機器人的設計圖作為藍本，在上面修改和新增一些孔來安裝 Pixy(CMUcam5)攝像頭

### IV. 切板、組裝





日志编号：04

活动日期：2016 年 10 月 2 日 - 2016 年 10 月 13 日

参与队员：何建群 戴智龍 簡水源 林震    活动地点：學校

## 讨论过程

上次 9 月 3 日開會的時候說過，到十月份的時候需要開一個會，各自匯報一下進度。因此，我們在 10 月 2 日，放完國慶假，便展開了我們的會議。

日期：10 月 2 日

出席人員：何建群、戴智龍、簡水源、林震

內容：

何：好，現在已經十月份了，昨天也放完國慶假，按照上次會議所說，我們先各自匯報一下進度。

簡：我和何建群已經成功研究 Pixy 與 RCU 的通訊問題，能夠把橘色球的坐標以及大小反饋給 RCU。

何：是的，我們購買了 Arduino Micro 板來讓 PIXY 與 RCU 溝通，並寫出了代碼，把 PIXY 的數據返回給 RCU。

簡：解決了溝通問題，我們就着手機器的設計。由於一個 PIXY 可測的角度只有 75 度，為了檢測所有方位，我們把五個 PIXY 連成一個正五邊形，以做到檢測 360 度的效果，就像復眼一樣。

林：哇，這是個好方法。

戴：對啊，這樣一來，機器人就不用自轉找球了。

何：嗯，我們現在已經設計好圖，並且切好板，組裝好了。不過程序部分我們還沒開始寫。

簡：那你們呢？你們那邊忙完了嗎？

林：我們 13 號比賽，比完賽之後就可以專心做機器人了。

何：好，那你們自己多注意時間的安排。

戴：嗯。其實我們已經想好了，兩部機器的樣子、硬件的接口、擺位基本上是一樣的。

林：沒錯，為了方便，以及利於任務賽的合作題，我們希望兩台機器人是一樣的。

簡：這麼巧，我們也有這個想法耶。

何：這樣也不錯，節省了設計兩台機器人的時間。簡水源硬件方面好，設計也很有想法，所以我們的機器人主要由他來設計。你們呢？

戴：我對重量組一點經驗都沒有，加上林震畫設計圖經驗多，應該主要交給林震負責。林震，你可以吧？

林：嗯，沒問題。不過我認為我們要先找個時間討論一下，決定好機器人的結構、硬件擺放等問題。

何：這是一定要的，你們 13 號比賽，日子也快到了，你們先好好準備比賽吧。

簡：對呀，這幾天我可以和何建群繼續把 PIXY 的程序部分完成好，等到你們比完賽，我們就開始準備澳門賽，開始設計、組裝、寫程序。

戴：好的。

這次討論，確定了以下工作：

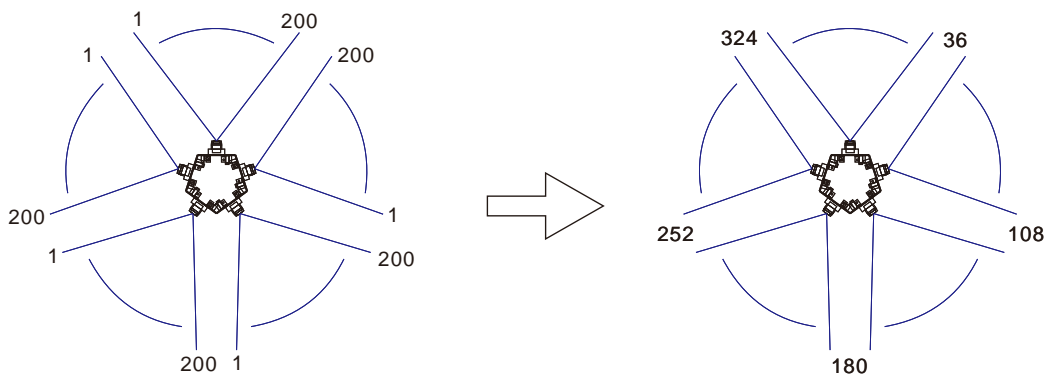
- 簡水源、何建群繼續研究 PIXY，寫程序
- 林震、戴智龍先比賽，13 號過後再回到機器人
- 13 號後各自討論機器人的設計，開始工作

## 开发及调试过程

何建群、簡水源同學

做好機器人後，我們便開始着手程序部分。上面曾說過，我們把 5 個 Pixy(CMUcam5) 攝像頭圍成一個正五邊形，以檢測全部方位。希望能夠把它當作 0 到 360 度，以方便我們知道球的方向。我們知道一個 Pixy(CMUcam5) 攝像頭的可測角度是 75 度，五個 Pixy(CMUcam5) 攝像頭就是 375 度。而機器人周圍，相當於一個圓，只有 360 度。因此，要實現 360 度，我們需要在程序上面作處理。

我們知道，PIXY 的可測角度是 75 度，而 PIXY 內部會自行處理數據，把橫坐標化為 1-320。經過 Arduino，我們便可得到球的坐標值。而由於我們希望方便自己，我們便在程序裏面做處理，把 1-320 變成 1-200。至於現在我們要把 5 個 PIXY 的橫坐標範圍合起來，並變成 360 度，因此我們需要把 360 度分成五份，每份 72 度。像平常一樣，以正北作為 0 度，正南作為 180 度，可得知每個攝像頭的分界值為：36、108、180、252、324。



在上面提到過，前面那個 Pixy(CMUcam5) 攝像頭鏡頭向上，而其餘四個攝像頭是倒着放的，鏡頭向下。因此在上圖中，其他四個攝像頭與前面那個攝像頭的範圍的最小最大值的位置是不一樣的。

那麼，我們如何處理數據以達至我們想要的效果呢？我們現在共知道兩個變量（PIXY 值與 360 度的效果值），並且它們之間是有關聯性的，360 度的值會隨着 PIXY 值的變化而變化。正在學函數的簡水源一想到有兩個變量，且它們之間有關聯，便想到用一次函數

去做處理。我們把 PIXY 的值設為未知數  $x$ ，360 度的值設為未知數  $y$ ，利用一次函數式  $y=kx+b$ ，聯立方程組便可解得  $k$  和  $b$  的值，從而解出把 PIXY 值變為 360 度的算式。

而由於每個攝像頭本來的範圍都是 1 到 200，變成 360 度後它們的範圍的數值必須不同，因此我們需要分開五種情況，對每個攝像頭做一次處理，五個攝像頭則共需做五次。

拿右邊的那個攝像頭來說，它本來的範圍是 200 到 1，而我們希望它變成 36 到 108。把 200 作為  $x_1$ ，1 作為  $x_2$ ，36 作為  $y_1$ ，108 作為  $y_2$ ，聯立方程組：

$$36 = k \cdot 200 + b \quad (1)$$

$$108 = k \cdot 1 + b \quad (2)$$

解得  $k = -0.36$ ， $b = 108$

所以對於右邊攝像頭的處理式為  $y = -0.36x + 108$

Handwritten derivation for the linear transformation:

$$\begin{aligned} 200 &\rightarrow 36 \\ 1 &\rightarrow 108 \\ \therefore \begin{cases} 36 = 200k + b \\ 108 = k + b \end{cases} & \text{把 } k = -0.36 \text{ 代入方程} \\ \text{兩式相減} & \text{求出 } b = 108 \\ -199k &= -72 \\ k &= -0.36 \\ \therefore \begin{cases} k = -0.36 \\ b = 108 \end{cases} & \\ \therefore y &= -0.36x + 108 \end{aligned}$$

按照同樣方法，可得出其餘三個攝像頭（不計前面那個）的處理式分別為：

$$y = -0.36x + 180$$

$$y = -0.36x + 252$$

$$y = -0.36x + 324$$

至於前面那個攝像頭，由於它的變化過程是  $324 \sim 0 \sim 36$ ，中間有個 0 作為分界線，因此在此處理前面攝像頭的數據時，我們需要分開兩種情況來寫。我們以它本來的分界值 100 作判斷，分別以  $x < 100$  和  $x \geq 100$  做處理，可得出下面兩條算式：

$$x < 100, y = 0.36x + 324$$

$$x > 100, y = 0.36x - 36$$

由於我們在程序裏面定義的變量的類型為 int，因此我們需要把 0.36 乘大 100 倍，也就是說，每條算式都乘大 100 倍。下面是最後所有的算式：

$$y = (36x + 32400) / 100$$

$$y = (36x - 3600) / 100$$

$$y = (-36x + 10800) / 100$$

$$y = (-36x + 18000) / 100$$

$$y = (-36x + 25200) / 100$$

$$y = (-36x + 32400) / 100$$

這樣一來，我們就成功把五個 PIXY 的橫坐標範圍變為 360 度了。

## 本次工作总结

- I. 由於我們把五個 Pixy(CMUcam5)攝像頭圍成一個正五邊形，而一個 Pixy(CMUcam5)攝像頭的可測角度是 75 度，合起來便有 375 度，超出 360 度。因此我們在程序上，把一個攝像頭的範圍變成 72 度。
- II. 利用一次函數把五個 Pixy(CMUcam5)攝像頭的範圍合起來，變成 360 度。

日志编号：05

活动日期：2016 年 10 月 14 日 - 2016 年 11 月 18 日

参与队员：何建群 戴智龍 簡水源 林震    活动地点：學校

## 硬件介紹

在介紹機器人設計之前，我們會先簡單介紹一下機器人所用到的硬件。

### ● 指南針：

我們所使用的指南針是中鳴的產品，其型號為 BE-2617。它有角度測量功能，也有歐拉角功能，另還有三軸加速度、三軸角速度等數據。不過我們平時不會有傾角問題，也沒有去研究，因此我們只使用了它角度測量的功能。

此外，它還有正北校正（將當前模塊的方向假定為正北方向，即 0 度）和硬鐵補償（消除當前環境裏面的大部分磁場干擾，提高其準確性）功能，這也是我們常常使用到的功能。

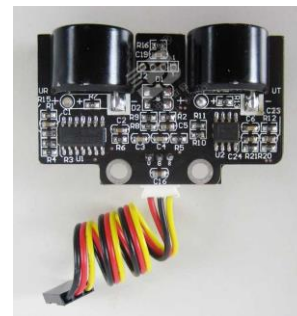
其長度為 33mm，寬度為 33mm，厚度為 8.3 ~ 10.4mm。淨重 5.8 克。



### ● 超聲：

我們所使用的超聲是中鳴的產品，其型號為 JMP-BE-6302。它的工作原理很簡單，發出超聲波，並計算出被物體反射的回波的時間差，從而計算出它和物體間的距離。

它是標準中鳴 RCU 通用三線接口，輸出量為模擬量。其測量範圍為 5 ~ 130 厘米。不必接觸物體便可測距，也不會發射紅外線影響光電類的傳感器。其狀態指示燈可以馬上反映測量





# 开发日志

結果，燈亮表示檢測到物體，燈滅表示沒有檢測到物體。

- 復眼：

我們所使用的超聲是中鳴的產品，其型號為 JMP-BE-1732。它由一個中央處理器和七個光敏二極管組成，能夠同時多方位測量紅外光的強弱。

它能夠讀取多個方向的紅外光強度，並且自動運算出最大值方向及其數值等等。我們平常就是利用這個功能去判斷球在哪個方向以及球的遠近。



- 控制器：

我們所使用的控制器是中鳴的產品，其型號為 X3-RCU。它是廣州中鳴數碼科技有限公司最新研發的機器人主控器，採用意法半導體公司新一代 M4 內核的 STM32F407 系列晶片設計而成，介面十分豐富，體積小巧、性能強勁且功耗極低。該控制器運行速度高達 168MHz，最大處理能力可達 210DMIPS，並且集成 DSP 和 FPU（浮點運算單元）



它支持圖形化編程和 C 語言編程，內置 USB2.0 接口，且控制器已內置模擬端口、指南針、復眼等實用性測試功能。

- 盤球裝置：

為了增加機器人的攻擊能力，提高其進球的能力，我們使用了盤球裝置，希望能控住球，做出一些攻擊性高且進球機率高的動作。

而我們使用的盤球裝置是我們自行設計的，其相



關結構我們會在下面（下一次日志）作說明。

- 馬達：

我們現在使用的馬達是 800 rad/min 的，電壓為 12V，額定功率為 15W。這款馬達是我們另行找人幫我們做的。我們把需求告訴他，然後他找了一些適合我們用的馬達給我們，根據我們想要的轉速，我們選擇其減速箱的齒輪比（14:1），他再幫我們做電機。

由於我們攻擊機追求力量與速度，因此我們曾在中鳴找來了 1700 轉的行星齒馬達。它能夠滿足我們對力量與速度的要求，是我們想要的東西。至於為甚麼現在不用了，是因為我們在測試的時候發現，這款電機很容易壞，並且從中鳴買來的這款馬達，各個馬達之間的速差很大，導致機器人行走路線不完美。再加上那個時候我們的機器人用不着如此高的速度，因此最後我們放棄使用中鳴的這款馬達。



- 全向輪：

從前我們使用的是中鳴的全向輪，因為在之前我們參加科協比賽時用的正是這款全向輪，並取得好的成效，機器人能夠全向運動，且動作順暢。但轉到 RCJ 的訓練時，我們發現這款輪子並不適合用於 RCJ，原因是這款全向輪容易「卡毛」，把場地（地毯）上的毛卡到每個小輪之間的間隙，導致機器人行走時動作不順暢。我們嘗試過拆開輪子，把毛清除掉，但拆掉輪子後，才發現這款輪子一旦拆了便裝不回去。於是我們只能不斷地更換輪子，每當輪子上的毛多時，就更換一個新的全向輪。

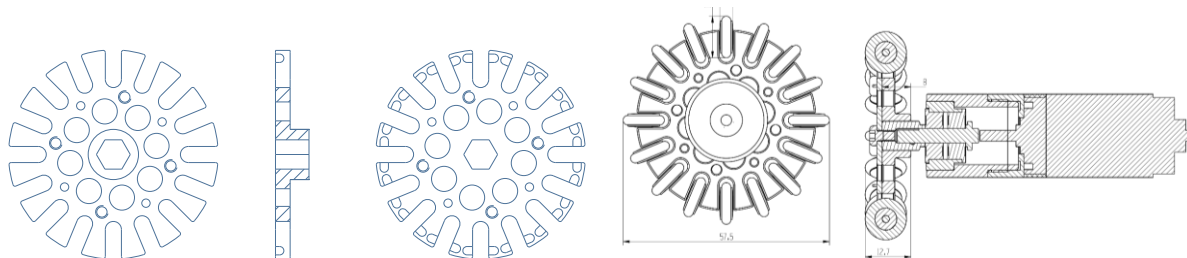


由於這樣過於麻煩，亦不太符合經濟效益，於是我們自己設計了一款全向輪，然後找人幫我們做出來。它依然會出現「卡毛」的情況，但它的好處是，它經拆卸後仍能重新組



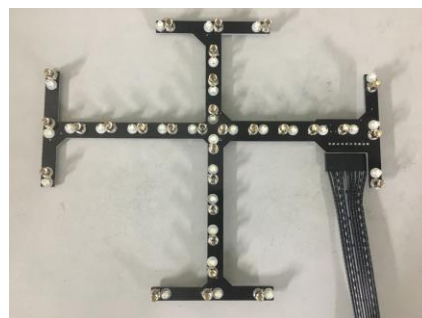
裝，繼續使用。這樣，我們便能清除掉輪子上的毛。

這款全向輪由 16 個小輪組成，直徑為 54mm，厚度為 5.5mm。我們能夠拆開輪子，把其小輪取出，更換其軟皮。這樣便不用購買新的全向輪，只需要耗費一點時間，把舊的軟皮取出，然後換上新的即可。



- 灰度板：

因為機器人一旦過了白線（出界）會被視為「損壞機器人」，必須離開場地一分鐘，這樣會大大減低我們整體的攻擊能力，連防守能力也會下降，因此「不出界」也是我們調試的一大重點。為此，我們放棄舊使用的中鳴灰度，而自行設計了一塊灰度板，至於其相關結構，我們會在下面（下一次日志）作說明



- 鋰電池：

我們主要使用兩種鋰電池，是從淘寶上購買的。

左邊那個我們慣稱「大電」，因為它比另一款電體積要大。它的電壓是 11.1V，容量 2600mAh，持續放電倍率 25C。

右邊那個我們慣稱「小電」，它的電壓是 7.4V，容量 850mAh，25C。



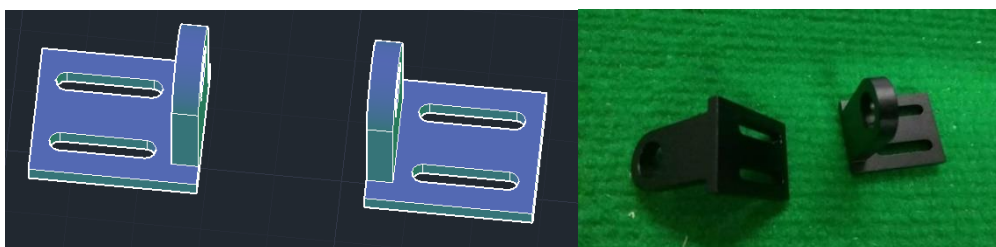
我們在上面（上一次日志）略略提到過盤球裝置以及灰度板，現在就詳細地介紹一下：

## 盤球裝置

我們所用的盤球裝置是我們自行設計的。它可分為四個部分：支架、主體、軸，與一些壓盤的部件。我們用 AutoCAD 去設計這一套盤球裝置，設計好後再去找商家做出來。

### ■ 支架

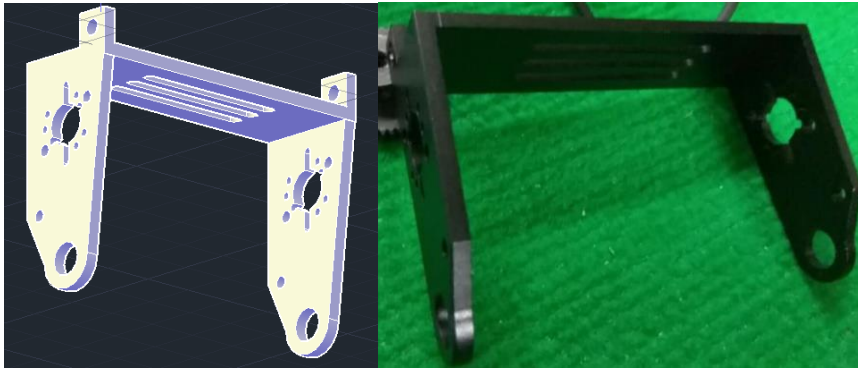
支架，顧名思義它是用來支撐整個盤球裝置，左右各有一個，且固定着盤球裝置的位置。在支架上面會有一個孔，用來安裝法蘭軸承來連接轉軸，有了軸承，盤球裝置就能夠上下擺動，在有球的時候，球會把盤球裝置頂上去，而沒有球的時候，盤球裝置就會向下擺，以增加碰到球的機率。



### ■ 主體

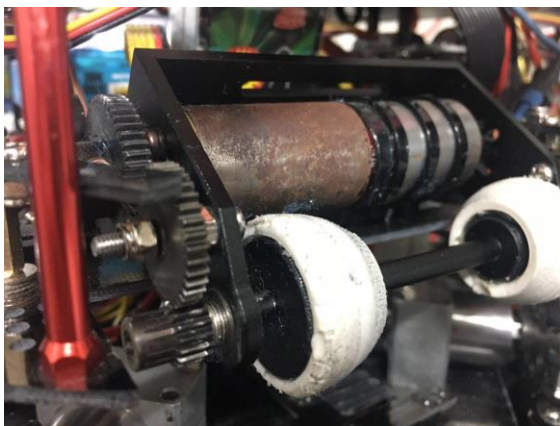
主體會安裝無刷電機、齒輪組和軸。我們用了 █ 的齒輪組來加速，分別有 █ 齒的大齒、█ 齒的小齒、以及一個中齒，中齒由於是中繼齒，因此不需去理會有多少齒。

不過，整個盤球裝置的大小會因為中齒的大小而有點變化，所以中齒的大小也不能完全不理會。那麼我們怎麼去決定中齒的大小呢？若是中齒太小的話，那麼大齒與小齒之間的距離就會小，導致盤球馬達與控球軸相撞。因此，我們先決定了盤球馬達與控球軸之間的距離，再去選定中齒的大小。



## ■ 軸

軸，連接着齒輪，上面套有軟皮。當盤球馬達轉動時，會帶動大齒轉動；大齒轉動時，通過中齒，小齒也會帶動軸轉動；軸轉動時，便會帶動輪子（軟皮）轉動，從而成功盤球。



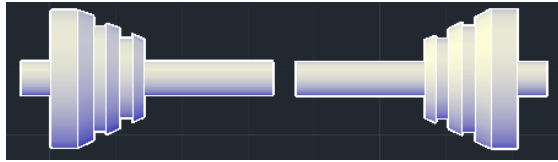
我們共設計過三代的軸，每一代都有它的優點，當然，也是因為它自身的缺點，我們才會一直改善，精益求精。

### ● 第一代：

第一代的控球軸是分開兩部分的，軸中空，中空的部分會有另一條一體化的軸作為連接，我們會用工業膠水（厭氧膠）把它黏到那條控球軸上，之後我們再塗熱熔膠到那兩個控球的位置，把軟皮固定在那上面，然後裝上小齒輪。其好處是當我們盤球裝置控到球時受到阻礙，就可以只弄鬆其中一邊的軸，而不會扭斷小齒那邊的一體化的軸，同時也容易組裝。而它的缺點是容易弄鬆那兩部分的軸，需要經常維

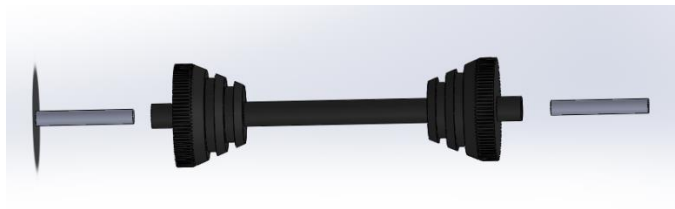


修。



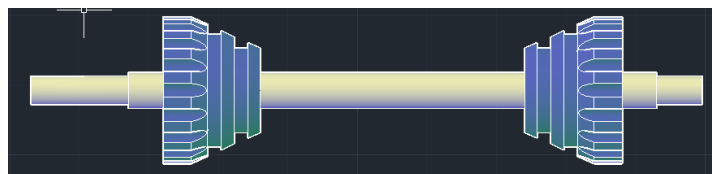
## ● 第二代：

第二代的控球軸變為一體化，然後兩邊分別用短軸承去連接，之後也跟第一代一樣，用熱熔膠上軟皮，然後裝小齒輪。第二代比第一代更加容易去組裝，且兩邊的短軸有更大的可調節性。其好處是假如盤球時再受到阻礙，我們不需再弄鬆其中一邊的控球軸，因為已經一體化了，比起之前更加穩定。而缺點則是當盤球受到阻礙時，會因為其扭力過大而弄鬆小齒輪那邊的短軸。



## ● 第三代：

第三代是我們目前的最後一代，控球軸的一體化不變，而旁邊的兩條短軸承則變回第一代的设计，成為了一條一體化的軸穿過控球軸。簡單地說，就是把分開兩部分的第一代變成一體化。接着便是一樣的步驟，上軟皮，裝齒輪。這個設計更為方便，更為容易組裝，由以前的「三套件」發展為「二套件」。其好處是當盤球受到阻礙，不需弄鬆控球軸，亦不怕扭力過大而弄鬆短軸，總的來說，第一代和第二代的顧慮都沒有了，比起以前，現在方便許多。但它同樣亦有缺點，雖然它改善了因扭力過大導致短軸鬆了的問題，但只是改善，減少其發生機率，並未能解決，且一但發生這個問題，整個控球軸便不能繼續使用。

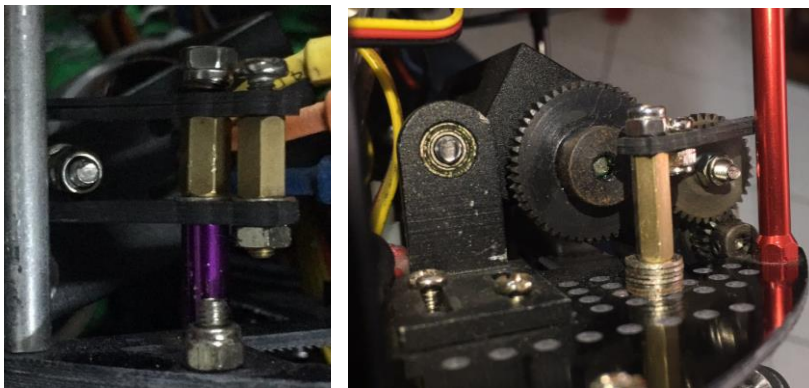


# 开发日志

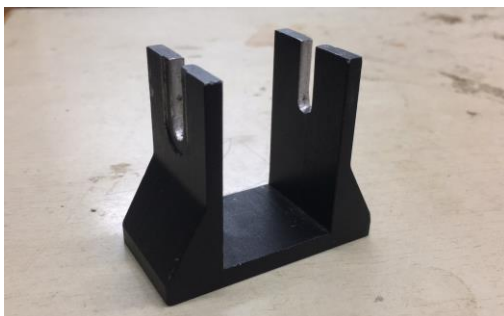
以上是我們三代的控球軸，每一代都有其優點與缺點，而我們現在使用的第三代亦有它的缺點，因此我們將會繼續改善，力求完美。

## ■ 壓盤的部件

由於我們的盤球裝置的支架那裡連有一個軸承，令盤球裝置能夠上下活動，因此當機器人盤到球的時候，馬達帶動控球軸轉動，而球也會跟着一起轉動，就會導致整個盤球裝置不停的上下抖動。而當盤球裝置向上移動到一個距離的時候，球就會無法與控球軸接觸，然後就盤不到球了，所以我們必須限定盤球裝置的抖動範圍。我們用了一些銅柱、板之類的部件，去限制它的抖動範圍。由於當盤球裝置向上移動時這些部件會頂着它，壓住它，因此我們叫它壓盤的部件。另外，由於每台機器人的盤球裝置的活動範圍都可能會不同，因此壓盤的部件並不是一體化的。



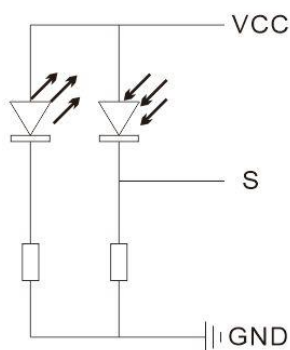
盤球裝置的四個部分，支架、主體、軸，與一些壓盤的部件我們已經介紹完了，另外還有一個輔助的部件可講。在我們利用熱熔膠上軟皮的時候，會經常出現熱熔膠分佈不均、凹凸不平 and 位置有偏差的問題，導致盤球裝置不穩定。因此我們設計一個架，去固定好控球軸，再用蠟燭去燒熱，把多餘的熱熔膠擠去，從而令熱熔膠平均分佈。



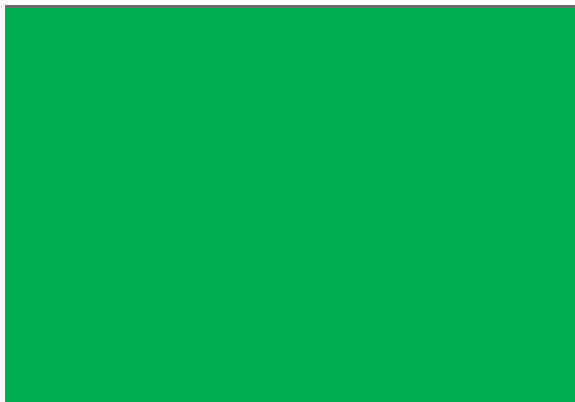
## 灰度板

由於機器人不出界很重要，為了避免機器人出界，我們希望能多加一些灰度，減少機器人檢測不到白線的機率。我們按照灰度的原理，當光敏電阻受到強光照時，它的阻值會有變化，然後測量它的電信號，就可得知我們想要的數據。我們利用了這一原理，設計出我們的灰度板，把多個「灰度」並聯在一起。我們用 Altium Designer 軟件來畫圖，畫完之後拿給淘寶的一個商家幫我們製造出來。

其中，我們把光敏電阻轉為用光敏二極管，因為光敏二極管的電信號比光敏電阻的電信號更穩定，而我們也簡單地畫了一個灰度的電路圖，如下：



我們根據這個電路圖去設計我們的灰度板，採用下拉電阻來測量定值電阻的電信號，當光敏二極管受到強光照時，光敏二極管的阻值會減少，定值電阻的電信號會增大，這樣便能實現：檢測到白界時 RCU 所反饋的信號增大。下面是我們一起設計的灰度板：





灰度板上有 28 盞發光二極管，28 盞光敏二極管，8 粒 1000 歐的電阻，28 粒 300 歐的電阻。我們在每盞發光二極管串聯一個 300 歐的電阻，以減少發光二極管的發熱量，而之後它們都是並聯起來的，再分組讀信號。

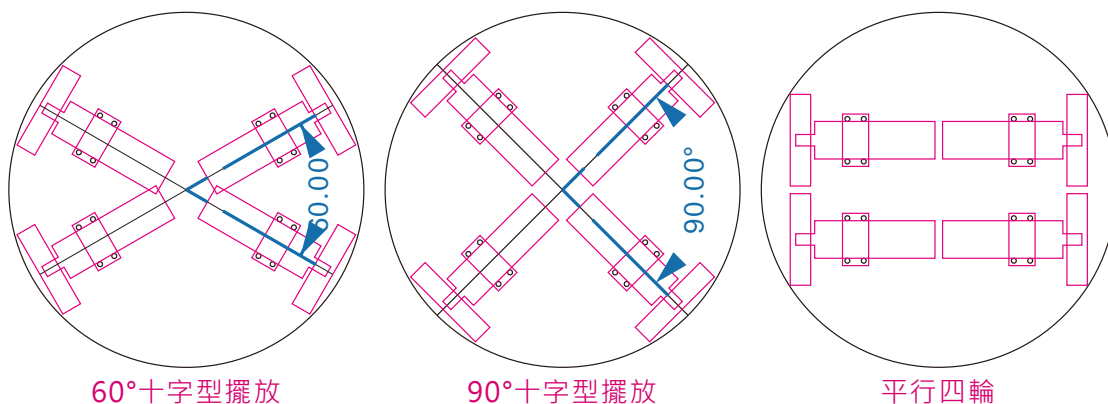
設計灰度板時要注意接線的粗細，之前有次設計我們沒有理會到接線的粗細，全都是細線，導致最後反饋的信號很弱。

而我們灰度板的外形是按照機器人去設計的，中間一個十字，然後十字的每一端都還有一排灰度，大概剛好在機器人底板的內側。

就是這樣，我們有了我們自己製造的灰度板。有一點要說明的是，上面所介紹到的只有 28 盞發光二極管的灰度板是用於雙盤機器人，而單盤機器人用的灰度板則配有 30 盞發光二極管，30 盞光敏二極管以及 30 粒 300 歐的電阻。

以上就是我們的盤球裝置和灰度板的設計。下面我們將說明一下我們的馬達擺放問題。

我們一直都將四個馬達擺成十字型，馬達與馬達之間的夾角是  $90^\circ$ ，這是為甚麼呢？我們在搜救項目中也看到很多隊伍的機器人的馬達是平行擺放的，這又是為甚麼呢？為此，我們特地對下面三種不同的馬達擺法作了研究，分別是  $60^\circ$  十字型擺放（前後馬達之間的夾角）、 $90^\circ$  十字型擺放和平行四輪擺放，看看它們之間的區別，有甚麼好處與壞處，驗證一下  $90^\circ$  十字型擺放是否我們的最佳選擇。



憑我們的認知與知識，我們認為：

➤  $60^\circ$  十字型擺放：能夠做全向運動，左右運動時速度小，前後運動時速度相對大；

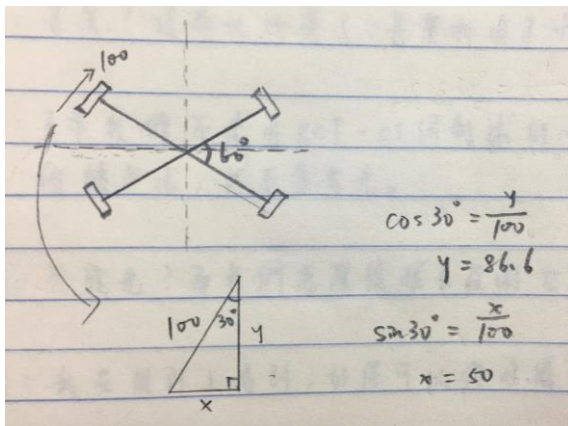
➤ 90°十字型擺放：能夠做全向運動，左右運動與前後運動的速度一樣；

➤ 平行四輪：前後運動速度大，但難以做全向運動。

我們的機器人要追球，其路線必然是弧線最好，因此機器人需要做全向運動。而平行四輪難以做全向運動，所以我們果斷放棄了平行四輪的擺法。至於剩下的 60°十字型擺放和 90°十字型擺放，從上面的結論來看，我們認為 60°十字型擺放的左右方向速度與前後方向速度不一，不太好。當然，我們不能紙上談兵，由於 60°十字型擺放和 90°十字型擺放的分別是其左右方向速度與前後方向速度的不同，因此我們分析對比了 60°十字型擺放和 90°十字型擺放兩種擺法的左右方向速度及前後方向速度（以左上的馬達作為研究對象），得出如下數據：

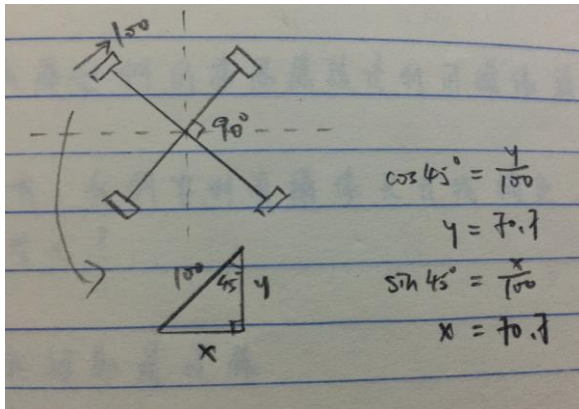
## ● 60°十字型擺放

當馬達最大速度 100 速時，利用三角函數，計算出其左右方向最大速度為 50 速，前後方向最大速度為 86.6 速。



## ● 90°十字型擺放

當馬達最大速度 100 速時，利用三角函數，計算出其左右方向最大速度為 70.7 速，前後方向最大速度也是 70.7 速。



通過上述兩組數據的對比，我們可以知道，60°十字型擺放的前後方向速度（86.6）比90°十字型擺放的前後方向速度（70.7）要大。但60°十字型擺放的左右方向速度（50）比90°十字型擺放的左右方向速度（70.7）要小。這也證實了我們一開始的想法是對的。其實簡單來說，我們的總結是，60°十字型擺放的話，前後運動速度與左右運動速度不一。相反，90°十字型擺放各個運動方向的速度都很平均。因此我們最後還是繼續選用90°十字型來擺放馬達。

## 讨论过程

由於上次開會所說的日子，13 號過了，因此我們兩小隊都各自開了一個會，討論自己隊的機器人。由於我們的會是分開進行的，因此我們將會分開寫討論過程。

### 一. 戴智龍、林震

日期：10 月 14 日

出席人員：戴智龍、林震

內容：

戴：終於完了比賽了，好累哦。

林：對啊，終於比完了。我們也是時候開始準備澳門賽的機器人了。

戴：好。按照我們之前所想，兩部機是一樣的結構。那…我們應該跟從前一樣，分為四層結構吧？

林：嗯，我認為可以繼續沿用底板、中板、上板和頂板四層的結構，而且也保持圓形的形狀，以用盡最大的空間。

戴：嗯，板的直徑用 219mm？

林：嗯…我個人就認為這樣不那麼保險，我一直都是用 217mm 的。

戴：那也沒問題，就 217mm 吧。不過頂板呢？頂板一直都是只裝指南針，頂版你也希望是圓形嗎？

林：其實…我希望能把機器人做的帥一點，頂板的話，我個人比較喜歡用一些特別的圖案。

戴：好啊，我記得你去年的機器人用的是 superman 的 logo 吧？

林：嗯，看，我想要的就是這種效果。能讓機器人有特色，也能讓別人記住這部機器人是屬於我的。

戴：這樣的感覺好好，會很有成就感吧？

林：嗯嗯，如果能獲獎的話，那就更好了。所以，我們一定要加倍努力

戴：好。那我們別扯太遠了，回到我們的討論吧。

林：去年我用的是 superman 的 logo，而蝙蝠俠、美國隊長、鋼鐵俠之類的圖案也都有人用過了。你有沒有甚麼想法啊？

戴：嗯…logo 的話…啊，飢餓遊戲怎麼樣？我覺得還挺帥的。

林：額…飢餓遊戲 logo 是甚麼樣的，我沒看過耶。

戴：沒關係沒關係，我現在找給你看。

（戴智龍在手機找了飢餓遊戲的 logo）

林：感覺不錯耶，那我們就用這個吧。

戴：好。那上中底板呢？應該沒有特定的圖案了吧

林：嗯，沒有。上中底板我一般都比較隨意，就先把硬件的位置放好，然後再畫上面那些孔啊之類的

戴：嗯，不過我有個意見。

林：甚麼？

戴：我看你們的大電都是橫在放在中板下面，每次拆卸我都覺得有點麻煩哦。為甚麼不豎着放呢？

林：豎着放…因為大電有點長，豎着放怕會影響到指南針。

戴：我們去測測看吧？

林：好。

（戴智龍和林震找來了林震的舊機，試着把大電豎直放在機器人的旁邊測量）

林：你這高度要在馬達上面才行，不然不準確。

戴：好。你看，這不是還留有一點位置嘛？

林：也是。按照現在的高度，豎直擺放大電也是沒有問題的。再加上我的舊機還沒有 220mm 那麼高，應該不會影響到指南針。

戴：那我們就豎着擺放咯？

林：嗯，這樣也方便我們換電。

戴：好，那設計的時候就注意每塊板都要留點位置放大電穿過。

林：嗯。那小電呢？小電我以前都放在控制器下方的，你有甚麼意見嗎？

戴：小電的話…我倒覺得沒甚麼問題。雖然我去年在輕量組的時候，是用紮帶固定電池的。不過紮帶更換有點麻煩，所以還是不用紮帶了吧。

林：我也覺得，換紮帶有點耗費時間。而且現在我們在重量組，也不太需要顧慮重量的問題。

戴：嗯，那小電就放在控制器下方吧。啊，對了，我們重量組，是不是意味着會用盤球裝置控球？

林：嗯，我有這個打算。攻擊機的話，還是有盤球裝置會吃香點。

戴：好啊好啊，那我們用一個盤球裝置還是兩個？

林：我認為嘛，我們兩個都沒有做盤球裝置的經驗，也沒有寫過控球的程序，要不就先用着一個盤球裝置吧。

戴：也是，用兩個的話，萬一我們應付不了就不好了。

林：嗯，我們這次先用着單盤，要是寫得不錯，純熟了，再加一個盤球裝置也不晚啊。反正我們都是力求完美的。

戴：嗯，那我們就做單盤機器人。盤球裝置應該是放在前面吧，還是會放後面？

林：嗯…我比較傾向於前盤，後盤的話我怕會烏龍球。

戴：你說的對，現在我們都沒有經驗，還是慢慢的來比較好。

林：嗯，那我們要用甚麼策略呀？

戴：我倒是覺得這個不急，我們可以等到做好機器人再去想，畢竟我們沒有做盤的經驗。

林：好，先抓緊時間設計。那…基本上的結構以及一些擺放問題都決定了，我們今天大概討論到這裏？

戴：嗯，那我們去工作吧。

## 二. 何建群、簡水源

日期：10 月 14 日

出席人員：何建群、簡水源

內容：

何：上次開學的時候我們決定了繼續使用雙攻策略，並且兩台機器人是一樣的，那你希望我們的機器人是甚麼樣子呢？

簡：不如…以上一年的設計為藍本？我們再看看有甚麼地方需要改善。

何：嗯…我認為我們主要有幾個地方要改善，第一是活頁，第二是馬達，第三是灰度板。

簡：活頁…你記得嗎，去年全國賽的時候，我們要重新組裝機器人，我記得那時候，裝活頁有點麻煩，有很多螺絲需要裝，耗費了一定的時間。而且當初之所以用活頁是為了方便我們拆盤，但現在我們盤球裝置壞的機率少了，所以使用活頁的頻率也就少了。

何：對哦，既然如此，我們今年就不用活頁這個設計吧。

簡：嗯。那至於馬達，有甚麼問題呢？

何：我們每次組裝的時候，馬達都要花一定的時間。因為我們固定馬達的東西是一個「套」，每次裝馬達的時候都要把全部的螺絲鬆了，才可以把馬達拆出來。

簡：的確。那…要不然我們想想別的方法來固定馬達吧。

何：有甚麼方法呢？整個拆下來會很麻煩，能不能把它分開來呢？

簡：啊，我想到了！我們可以用膠粒把板連接起來弄一個直角形啊！

何：等等…你…你在說甚麼？

簡：哎呀，現在甚麼都沒有很難說，等我設計好做出來給你看吧。

何：好。那灰度板呢？我發現，我們的灰度板很容易壞，特別是世界賽的時候。

簡：對對對，那時候灰度板燒了，我們把底板拆了，拿了灰度板出來並修好後，過一會又壞了。經過幾次的拆裝，才把灰度板弄好。每次都需要把底板拆出來，好麻煩哦。

何：我想說的就是這個問題。你不是懂得灰度板的設計嗎？你有沒有甚麼好辦法啊？

簡：這個得再研究研究。

何：啊，對了，還有火焰，平常試機的時候，因為我們機器人的板是碳纖做的，會導電，令火焰檢測短路。

簡：嗯……

何：啊，我們可以考慮用新的中鳴火焰傳感器，它的板上面新加了一個孔，可以讓我們固定。

簡：好。對了，這次設計新機，我希望有新的把手設計。

何：那你有想法了嗎？

簡：暫時還沒有。

何：那先把其餘的都設計好吧。

簡：嗯。

這兩次的討論，主要確定了我們兩隊機器人的大致結構以及一些要注意、要改善的地方：



➤ 戴智龍、林震：

沿用底板、中板、上板和頂板四層結構

底板、中板、上板保持圓形形狀，直徑為 217mm

頂板為飢餓遊戲的 logo

大電豎着穿過每塊板擺放，設計時注意每塊板都要留有位置讓大電穿過

小電放在機器人控制器下方

機器人為單盤機器人，盤球裝置在機器人的前方

➤ 何建群、簡水源：

以上一年的設計為藍本

不使用活頁

用別的方法固定馬達

改良灰度板

使用新版的火焰傳感器

設計新的把手

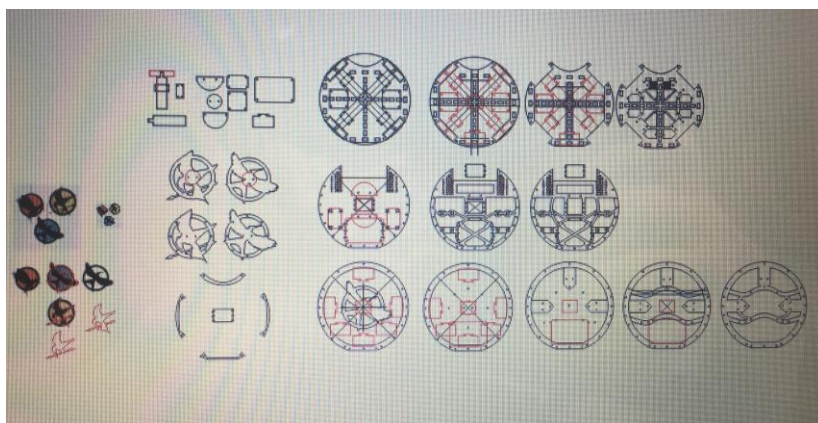
## 开发及调试过程

### 戴智龍、林震同學

在 10 月份我們小組討論的時候說過，我（林震）主要負責設計的部分。在與戴智龍商討機器人結構過後，我便開始設計，而戴智龍會從旁提意見，包括美觀，可行性等等的問題。

### 設計

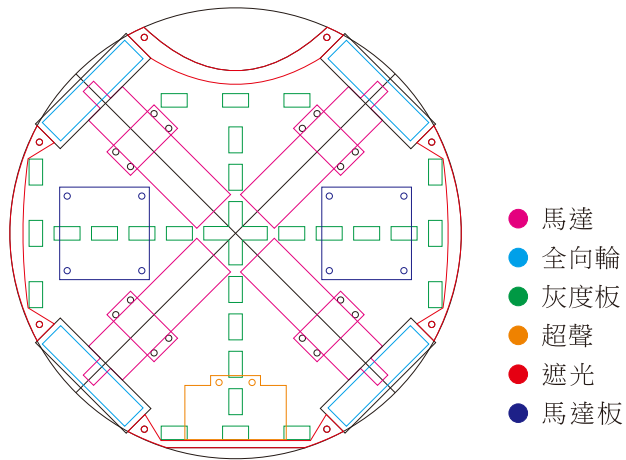
我們的機器人大致可分為四層板：底板、中板、上板、頂板。由於剛剛比完賽，才工作沒幾天，所以我們都比較鬆懈，沒甚麼緊張感，做事的進度也較為慢一點，本來可能只需要一個星期就能設計好的圖，我們花了將近兩個星期。由於時間比較充足，所以我們在畫圖的時候也比較仔細，每塊板都是一步一步循序漸進的來。



我們使用 CorelDRAW 軟件來畫圖。以下我將詳細地介紹一下我們每塊板的設計。

#### ■ 底板

為了讓機器人碰到球的機會大一點，我們希望機器人的佔地面積盡可能的大。由於大會賽規規定機器人的直徑限制為 220mm，為了避免超出限制，以及利用最大的空間，我們把底板設計為直徑 217mm 的圓形。在硬件方面，為了降低機器人的重心以便行走時更加順暢，我們決定在底板擺放較多的零件，有四個馬達、四個全向輪、一塊灰度板、一個超聲、若干遮光和兩塊馬達板。



- 馬達：

為了讓機器人向各個方向運動時的速度較為平均，我們選擇了 90 度的十字擺法。

- 全向輪：

很簡單的，全向輪跟馬達一樣，呈十字形擺放，裝在馬達上面。

- 灰度板：

由於我們設計灰度板時，是按照機器人的形狀設計的，所以直接把灰度板擺放在底板的中間即可。

- 超聲：

我們的機器人共裝有四個超聲，前後左右各一個。而由於中板和上板沒有足夠位置擺放後超聲（在介紹上板時會作說明），所以我們把它放在底板上。

- 遮光：

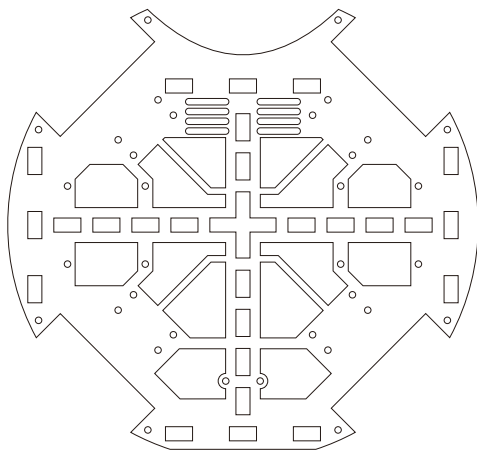
遮光是我們另行設計的板。由於我們灰度板的設計，每個光敏二極管都是完全暴露（如下圖所示）的，這樣的話，它就有可能受到外界的因素例如比賽用球所發出的紅外光的影響，所以我們便設計了一些板放在灰度板（光敏二極管）的周圍，以減少外界對灰度板（光敏二極管）的影響。



## ● 馬達板：

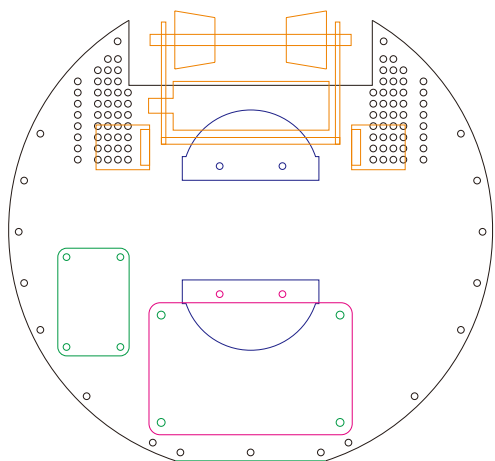
由於我們用的是二路馬達板，所以需要兩個馬達板來驅動四個馬達。而為了讓重心分佈平均，我們把它們對稱擺放在底板的左右兩邊。

以下是底板的設計圖：



## ■ 中板

為了用盡空間，中板和底板一樣，直徑為 217mm。在硬件方面，中板擺放了兩個復眼、一個開關、一個控制器和一個盤球裝置。



- 復眼
- 開關
- 控制器
- 盤球裝置

## ● 復眼：

我們使用了兩個復眼。由於每個復眼都是半圓形的，我們把它們合起來變成圓形狀擺放，這樣就便於機器人檢測各個方位的紅外線光值，以確定紅外球在哪個方向。

但由於我們決定把供馬達使用的鋰電池豎直地擺放在機器人中間，所以兩個復眼之間會留有一段距離。

## ● 開關：

我們把開關放在板的邊邊位置，以便於我們使用。我們使用的開關上有兩個按鈕，分別用來控制馬達和盤球裝置。有了開關去控制馬達，便於我們在運程序時讓機器人停下，去檢查機器人的某些數據。比方說，我想知道分區的程序有沒有錯，只需顯示區域的變量，在運程序後，關掉馬達的開關，便可自由移動機器人，檢查機器人所分的區是否跟想像的一致。

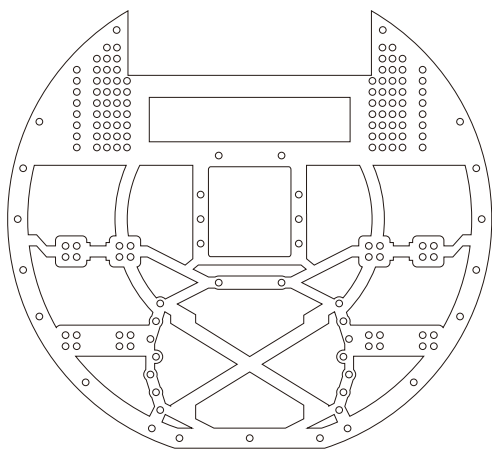
## ● 控制器：

我們把控制器放在中板的中後位置。之所以放在中板，是因為我們認為中板的高度便於我們使用，不會太高，也不怕低。而放在中後方的位置，顯而易見，也是為了方便我們使用，且能把重心集中在中間。

## ● 盤球裝置：

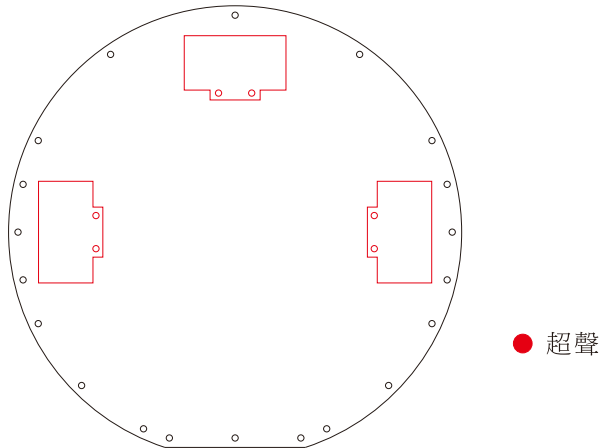
顧名思義，盤球裝置就是用來盤（控）住球的裝置。而我們的想法是單盤機器人，且盤球裝置在機器人的前方位置，機器人盤到球後便做出動作（後面程序部分會詳細說明）。

以下是中板的設計圖：



## ■ 上板

同樣地，為了盡可能的利用空間，上板的直徑也是 217mm。在硬件方面，為了降低重心，幾乎全部硬件都放在了底板以及中板，上板只擺放了三個超聲。

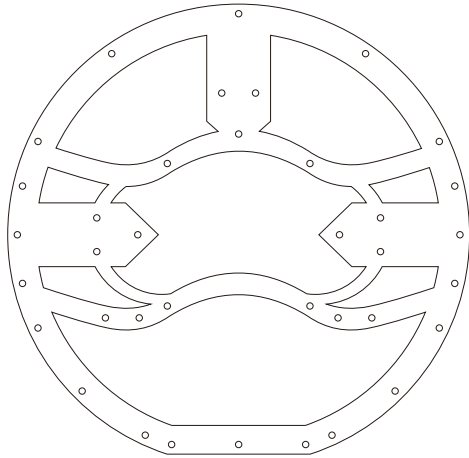


## ● 超聲：

在上面介紹底板時有提及到，我們的機器人共裝有四個超聲。但由於中板和上板之間的後方位置裝有控制器，如下圖所示，以致沒有足夠位置擺放後超聲，所以把它放在底板上。而前、左、右三個超聲，則分別擺放在上板的前、左、右三個位置。



以下是上板的設計圖：



其中，可以看到，在板的後方，留有一大塊空位，這是為了方便我們清楚地看控制器的顯示屏。而中間的那塊空位，很明顯的，是因為要放大電的緣故。

## ■ 頂板

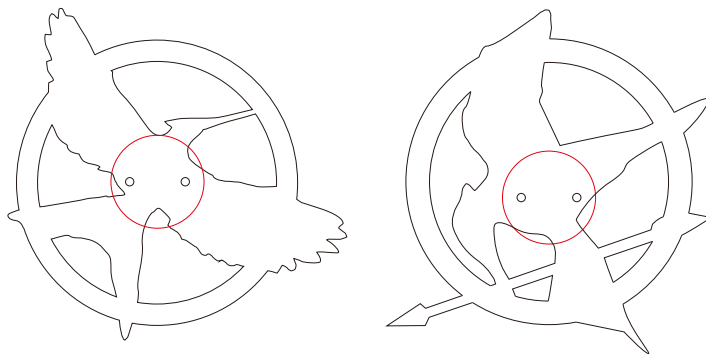
頂板只用於擺放指南針，所以它的面積對比起上中底板要小得多。我們決定把機器人弄得特別一點，可以讓別人一看就認得這台機器人是屬於我們的。於是我們參照美國電影「飢餓遊戲」的 logo，設計出下面兩塊板。設計兩種圖案一來是為了可以一次滿足兩個願望，二來可以用於區分這兩台「學生機器人」。



### ● 指南針：

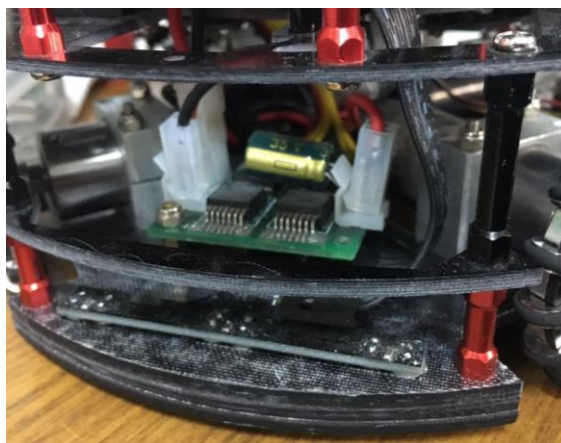
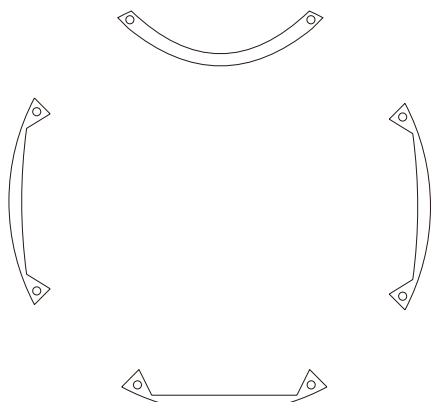
之所以把指南針放在機器人最高的位置---頂板，是為了盡可能地減低其他機器人對我方機器人所帶來的磁場影響。

以下是頂板的設計圖：



## ■ 其他

為了避免與其他機器人激烈碰撞時對自身產生了不必要的破壞（例如自身的零件、或是連接柱等），我們加了一些部件來保護機器人（由於是用來防止碰撞時產生的影響，下稱“防撞”）。由於它跟遮光一樣，是擺放在機器人周邊位置，所以它的設計也跟遮光一樣。



## 組裝盤球裝置

另外，由於我們都沒有組裝盤球裝置的經驗，因此在我畫設計圖的同時，戴智龍去向簡水源學習了如何組裝盤球裝置。下面是戴智龍同學的學習過程：

因為今年是我第一年作為重量組的一員去參加比賽，所以對於我來說，重量組的事物都很新鮮，尤其是盤球裝置，盤球裝置控到球後，機器人再做出相應的動作，一整個很帥



的感覺。但我對盤球裝置可以說是一竅不通的，因此在我的隊友林震設計機器人時，我找了我們隊當中經驗最豐富、硬件最好的簡水源同學教我。

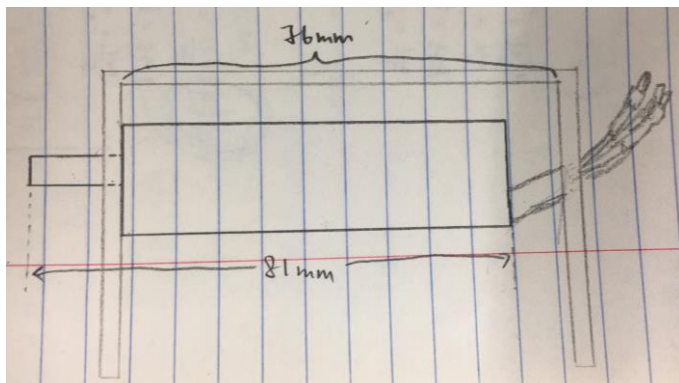
簡水源說，我們要把盤球裝置做好，才把它裝到機器人上面。他還教我組裝盤球裝置也不外乎是那幾個步驟，把步驟記牢了，按照步驟去組裝，就可以組裝出盤球裝置了。那個步驟是：包硅鋼 → 塞馬達 → 鎖大齒 → 上中齒 → 定軟皮 → 裝轉軸。

## ● 包硅鋼

我們不能把盤球馬達直接裝到主體部分上，因為這樣會影響到指南針。因此我們要先在盤球馬達的表面包一層硅鋼片，以減低盤球馬達對指南針的影響。

## ● 塞馬達

之後，我們會從盤球裝置的主體部分入手。由於主體的內側長為 76mm，盤球馬達（包括其轉動軸）長為 81mm，盤球馬達比主體的內側部分要長，而我們要使馬達的轉動軸凸出主體以外，必須用上蠻力，把馬達塞進去。同時，我們也要把馬達的那三條信號線穿出來。之後，我們在馬達轉動軸的那一邊，上了三顆小螺絲去固定馬達。



## ● 鎖大齒

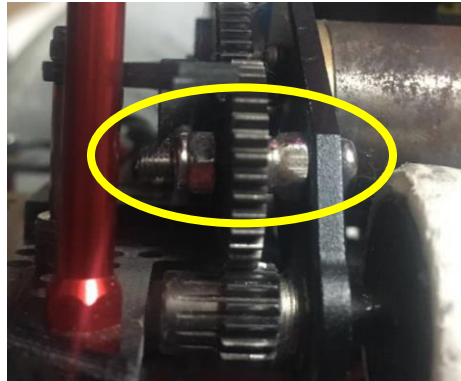
塞完馬達後，我們需要裝一些齒輪上去。因為我們盤球裝置的基本原理是，盤球馬達轉動，通過齒輪，帶動控球軸轉動，從而盤球。而在上面介紹盤球裝置的時候也提到過，我們用了三個齒輪。其中，    齒的大齒是連接在盤球馬達的轉動軸上的。因此，塞完馬達



之後，下一步就是鎖大齒，我們先在大齒上面塗上厭氧膠，然後穿過馬達的轉動軸，把它固定在馬達上面。之後再把一個小螺絲裝在齒輪上，固定好齒輪與馬達。

## ● 上中齒

鎖好大齒後，之後就到中齒了。由於中齒要銜接大齒，因此中齒到主體的距離必須跟大齒到主體的距離一樣。我們量度後，測得大齒到主體的距離大概是 3.5 毫米，於是我們用了 4 毫米的防滑螺母去隔開中齒與主體。之後再在中齒的另一端上一個螺母去固定中齒，不過這次不需要把中齒鎖緊，因為一旦鎖緊了，中齒就轉動不了。



## ● 定軟皮

下一步呢，我們要把軟皮弄到控球軸上面去。我們先把兩個軟皮穿到控球軸，之後在擺放軟皮的地方塗上熱熔膠，再把軟皮套到那上面。

不過有可能會出現熱熔膠不均勻的狀況，而導致軟皮不完全緊貼着軸，因此我們要作出調整。我們用電烙鐵或者蠟燭去加熱控球軸，令熱熔膠熔化，在熔化的同時，我們可以逼出多餘的熱熔膠，這樣，就能讓軟皮緊貼控球軸了。

## ● 裝轉軸

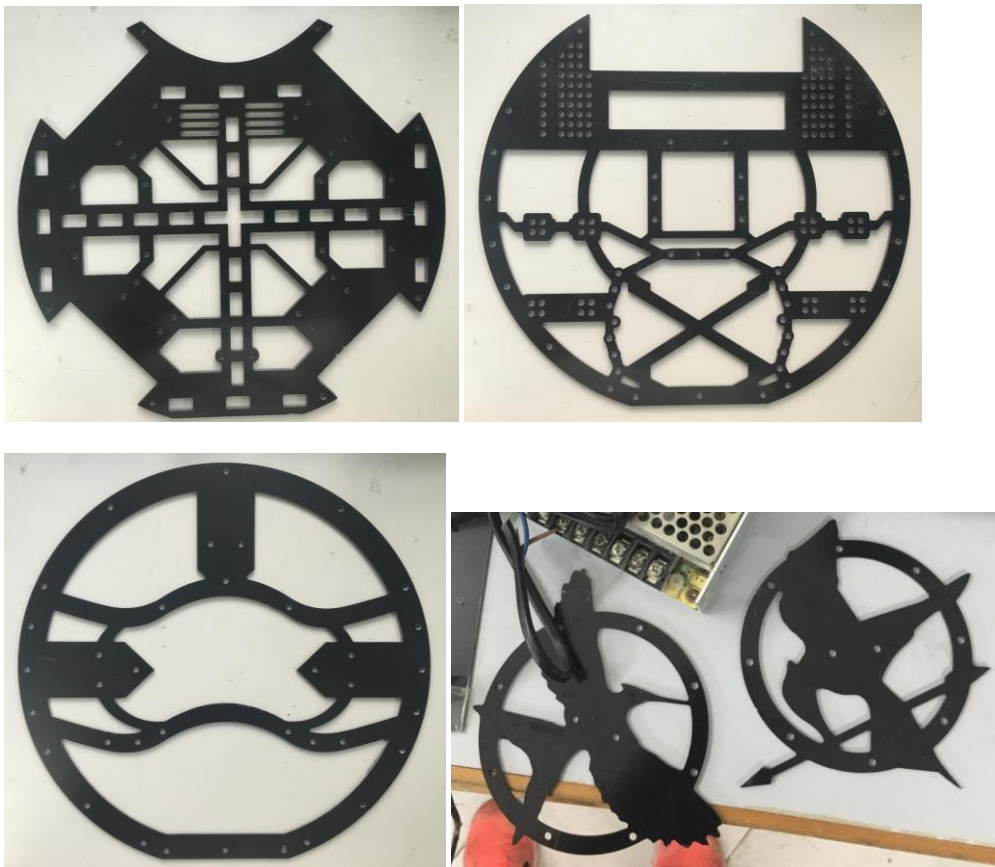
控球軸也搞定好之後，就要把它裝進主體部分。我們把控球軸塞進主體部分以後，在控球軸一端裝上小齒，然後在小齒上面裝上基米，固定小齒輪，控球軸另一端則用螺絲螺母固定好。

就是這樣，我完成了盤球裝置的組裝。之後就等林震設計好機器人、切好板，就把盤球裝置裝上去。

## 切板

設計好機器人後，我們使用 CNC 切割機把頂、上、中、底板切割出來。每塊板的材料都是環氧樹脂。由於底板被擺放的硬件最多且最重，所以底板最厚，厚度為 3mm。中板雖沒有擺放四個馬達如此重的硬件，但由於盤球裝置也不輕，所以它的厚度為 2mm。上板和頂板的硬件很少，因為選用了厚度為 1.5mm 的板。而遮光由於要用來包圍灰度板上的光敏二極管，所需的高度要 6mm，而我們最厚的板是 3mm，所以我們選擇了 3mm 厚的板，兩塊板合起來便有 6mm。

由於我們平常要上學，再加上兩台機器人的頂、上、中、底板和遮光板加起來總共要切 9 塊板，平均每塊板要切 35 到 55 分鐘，這樣算起來，要切好兩台機器人的板，總共要花大約七小時的時間，所以我們花了 3 到 4 天的時間來切板。下面是我們切割好的成品：



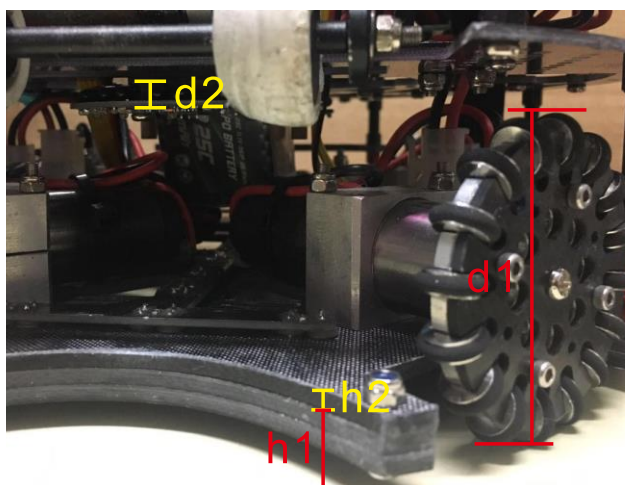
## 組裝

切好板後，便是組裝機器人。因為戴智龍同學去年是輕量組，所以他需要再焊接一塊新的灰度板。我們在上面提到過每個 LED 都設有一個電阻，每條杜邦線也設有一個電阻，而每個電阻的體積都很小，所以要焊接 40 個電阻，再加上 30 個 LED 和 30 個光敏二極管，要花一定的時間。而我（林震同學）因為還留有去年重量組的機器人，所以便直接把舊機上的灰度板拿來用，以節省再焊接一塊灰度板的時間。

然而組裝機器人並不是隨便拿些螺絲螺母連接好各個硬件，任意用幾條柱子把每塊板連接起來那麼簡單。板與板之間的距離，每個硬件的高度，還有一些在組裝時要注意的細節等方面，都是需要經過仔細的考量的。

### ■ 板與板之間的距離

在板與板之間的距離當中，中板與底板之間的距離尤其重要。它必須大於一個輪子的直徑減去底板的厚度以及底板到地面的距離，再加上一個復眼的高度。換成我們平常看的算式：輪子直徑 - 底板厚度 - 底板到地面的距離 + 復眼高度。設底板到地面的距離為  $h_1$ ，底板厚度為  $h_2$ ，輪子直徑為  $d_1$ ，復眼高度為  $d_2$ ，即  $d_1 - h_2 - h_1 + d_2$ 。圖文結合，可以得出：若中板與底板之間的距離不符合上述所說的條件，輪子便會遮擋到復眼。換句話說，復眼就會多了 4 個盲點（因為有 4 個輪子）。無論是對於攻擊機或防守機來說，都需要看到球之後才會有下一步的攻擊/防守動作。所以，在組裝的時候，必須注意中板與底板之間的距離。



下面是我們機器人板與板之間的距離：

|       | 距離(mm) |
|-------|--------|
| 底板到地面 | 11mm   |
| 中板到底板 | 51mm   |
| 上板到中板 | 55mm   |
| 頂板到上板 | 62mm   |

## ■ 硬件的高度

需要特別注意的硬件是超聲和指南針。在安裝超聲的時候要注意它們的高度，必須低於牆壁的高度。而指南針則是放得越高越好。一來減低別的機器人所帶來的影響，其次是自身的影響。由於大電豎直擺放在機器人中間，比平常橫着放的時候要高，所以指南針必須遠離大電，以防被影響。



## ■ 固定電池

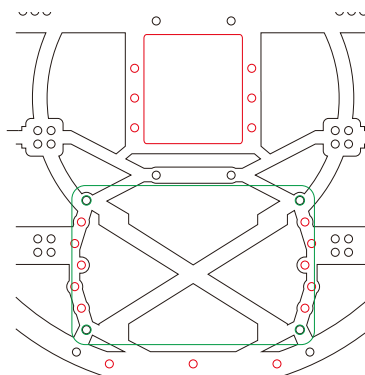
我們在討論機器人結構的時候有提過，我們把小電放在控制器下面，把大電豎直地放在機器人中間。由於在比賽的過程中，機器人經常會與別的機器人發生碰撞，因此無論是



大電或小電都會不停的搖晃。在搖晃的過程中，鋰電池有機會晃到外面（超出板的範圍），從而觸犯了規定（機器人的最大直徑為 220mm）。因此我們需要把鋰電池固定好，以免犯規。

戴智龍同學去年在輕量組的時候，是用尼龍紮帶把鋰電池固定好的。不過這種方法有一個缺點就是會有一點麻煩，因為需要在每次換電的時候更換紮帶。在比賽前更換還好，因為時間比較充裕。若是在中場的時候，特別是我們需要調試機器人的時候換電，可能時間就較為緊一點。所以這次我們不打算採用這種方法。再加上今年我們是重量組，不需要因為擔憂重量的問題而選用質量輕的紮帶。

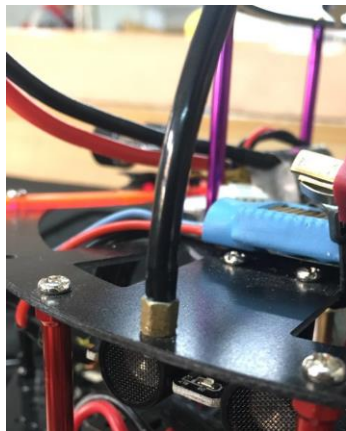
而今年我們所採用的方法，是用一些小零件來阻擋鋰電池搖晃。我們把小電放在控制器下面，然後在小電位置附近加上一些柱子、螺母、紮帶等小零件來限制小電的活動範圍，從而避免小電嚴重搖晃的情況。而對於豎直擺放的大電，我們在設計時，已限制了大電的可活動範圍。



通過先天的設計以及後天的加工，我們順利解決了這個問題。

## ■ 把手

在賽規當中，有一條規則：所有的機器人必須有一個牢固的把手，以便握持和移動。把手的尺寸可以超過機器人 22 厘米的高度限制（摘錄自 2017RoboCup 機器人世界杯中國賽比賽規則---RCJ 机器人足球賽項目 5.6 條）。有見及此，我們在機器人上板加了兩個銅柱，並在銅柱表面塗上厭氧膠，用來固定軟膠管。軟膠管就是機器人的把手。



## ■ 盤球裝置

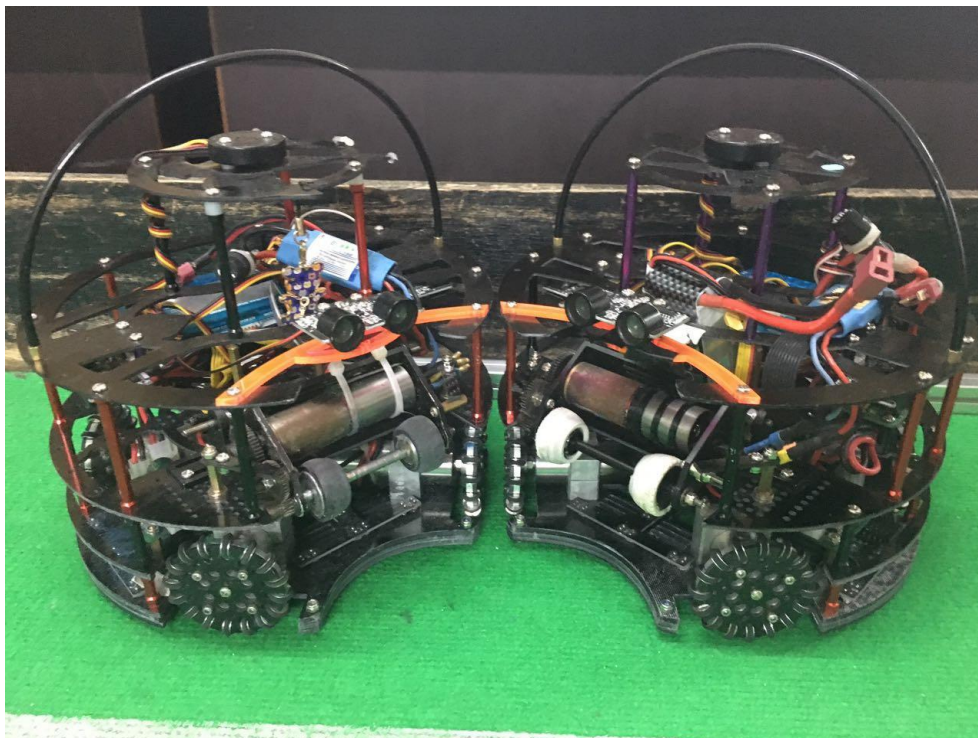
在之前，戴智龍已組裝好兩個盤球裝置，現在切好板，我們便把它們裝到機器人上面。首先，我們把支架連接到盤球裝置的兩端，然後決定好支架在中板的位置，即盤球裝置的位置。要注意的是，盤球裝置不能放太出，否則會超出 220 毫米的規定。之後，盤球裝置要保持在中間，不能偏左或偏右。按照上面的基本原則，決定好盤球裝置的位置後，就上螺絲，固定好支架，從而把盤球裝置固定在中板上。

之後呢，我們要用一些部件去壓盤。在介紹盤球裝置時有提過，盤球裝置可以上下活動，因此我們需要加一些部件去壓一壓盤球裝置，以防它抖動得太厲害，或者抬得太上導致接觸不了球。

我們用了一些金屬墊片、銅柱、螺絲、螺母和板去組成一個壓盤的裝置。



以下是我們兩台機器人的完成品：



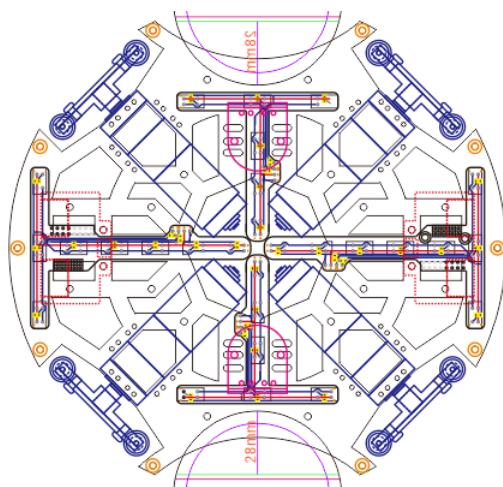


## 何建群、簡水源同學

在之前的討論裏，決定了由我（簡水源）主要負責設計的部分。在上次與何建群討論的時候，也決定以上年的設計為藍本，修改一些不好的地方，並加入一些新的設計。還有一點就是我們的設計是以低重心為目標的，

### ■ 底板

剛才說過，機器人的設計以低重心為目標，因此機器人的底板和中板放置了較多的零件，分別有四個馬達、四個全向輪、四塊灰度板（新版）、兩個小部件和防撞。



#### ● 馬達、全向輪：

馬達與全向輪位於上圖的藍色部分。為了行走時速度平均，馬達與全向輪呈十字形擺放。

#### ● 灰度板（新版）：

灰度板位於上圖的黑色（粗邊）部分。我們用的是新設計的灰度板。雖然是新版本，但它仍然是按照機器人的形狀結構而設計的，在外形上，簡單粗略地形容，就是把舊版分成前後左右四部分。所以也跟從前一樣，把灰度板整體放在板的中間位置，即前面的灰度板就放在板的前面，後面部分的灰度板就放在板的後面，如此類推。

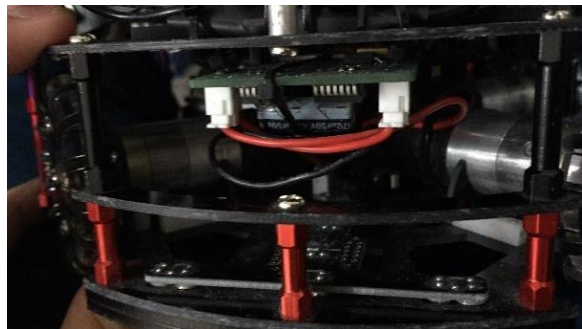
#### ● 小部件：

兩個小部件分別位於上圖的紫色部分。由於有一條賽規規定機器人控球區的最大深

度為 3cm，所以我們加了兩個小部件用來防止球進入到機器人裏面，超出限制。而選用了兩個的原因，顯而易見，是因為我們的設計是雙盤機器人。

- 防撞：

為了防止與別的機器人碰撞時，機器人內部的零件受損，我們設計了防撞板。在上圖可看到一些橙色的孔，是用來連接防撞與底板。

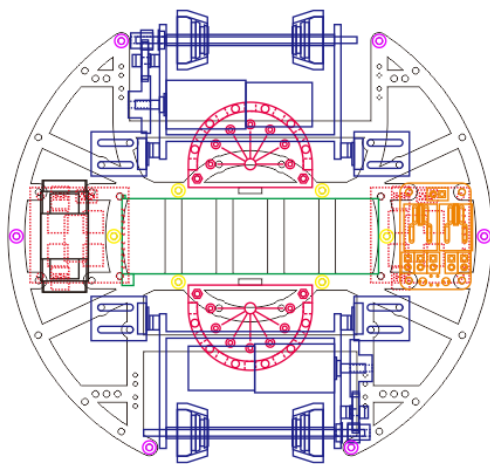


- 其他：

在底板設計圖的左右邊，有兩個紅色的虛線，預留超聲的位置。為了方便在全國賽時，任務賽可能需要不同位置的超聲，特意預留了位置。

## ■ 中板

由於低重心的目標，中板也放置了許多零件，分別有兩個盤球裝置、兩個復眼、兩塊馬達板、兩個電調、一個鋰電池和一個電源開關。



- 盤球裝置：

由於我們做的是雙盤機器人，前後各一個盤，因此位於上圖的藍色部分。

- 復眼：

復眼位於上圖的紅色部分。我們把兩個復眼擺放成圓形。但因為中間要放大電，因為兩個復眼間隔了一段距離。

- 馬達板：

我們使用的是二路馬達板，所以需要兩個。馬達板位於上圖的紅色虛線位置，左右對稱擺放。

- 電調：

電調位於上圖的黑色（粗邊）部分。由於是雙盤機器人，所以我們使用了兩個電調來控制電機。為節省空間且方便於我們操作，我們用紮帶把它們捆在一起。

- 鋰電池：

鋰電池擺放於上圖的綠色部分。在上面討論部分提及過，我們整部機只用一個 11.1V 的鋰電池供電。鋰電池擺放在中板中間位置，一來方便，二來亦能集中重心。而周圍有不少黃色的孔，用來安裝鋁柱以解決機器人行走時電池不停搖晃的問題。

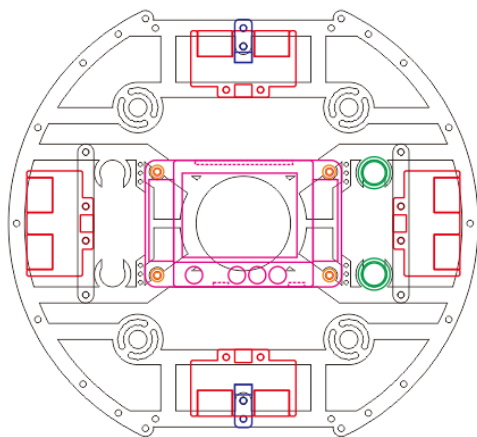
- 電源開關：

電源開關擺放在上圖的橙色部分。我們用它來控制馬達、盤球裝置以及控制器。

從上圖可看到，在中板上有很多的空隙，是為了能讓底板硬體的端子線能容易地穿過中板，連接到機器人的控制器。而同時也能減輕機器人的重量。

## ■ 上板

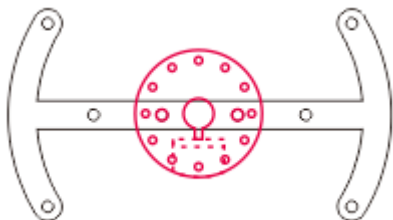
中板和底板都擺放了較多的零件，以致於上板只放置了四個超聲、兩個保險絲座、一個控制器和兩個火焰測量。



- 超聲：  
超聲擺放在上圖的紅色位置。
- 保險絲座：  
保險絲座位於上圖的綠色位置。用以防止電調因撞擊致使電流過大令其損壞。
- 控制器：  
控制器位於上圖的粉紫色位置。放在上板極為方便，因沒有過多的東西防礙我們操作。
- 火焰測量：  
火焰測量擺放在上圖的藍色位置。用以判斷盤球裝置是否盤到球。

## ■ 頂板

由於很多硬件都集中在機器人底部，所以頂板只擺放了指南針。頂板安裝在控制器的正上方，這樣，指南針就處於機器人頂端的中心位置。



## ● 指南針：

由於指南針很容易受到機器人的馬達的磁場干擾，我們希望把指南針放在機器人的最高處。因此，指南針安裝在機器人頂部。換句話說，頂板其實是為了安裝指南針才存在的。

以上就是我機器人的設計，與上一年對比，我們主要修改了：

### ➤ 更換了火焰測量模塊

原因：因為在以前的比賽中，我的機器人是用碳纖維板所制，而碳纖維會導電，加上舊的火焰測量的電子部分是外露的，沒有保護，所以在碰撞時火焰有機會與碳纖維板接觸而導致短路。由於新的火焰測量模塊有一個可以讓我們固定的孔，固定了火焰測量模塊就不用怕與碳纖維接觸而導致短路了。



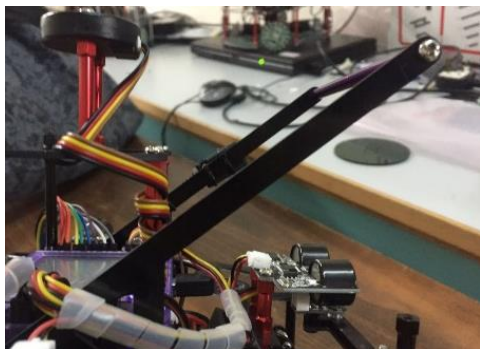
（新的火焰測量模塊）



（舊的火焰測量模塊）

### ➤ 更換了把手

新的把手是用環氧樹脂做的，而舊的把手是用 PU 空壓軟管所制。由於新把手是用環氧樹脂做的，因此它的硬度相對於舊把手來說會比較好。加上新的把手我們都是用螺絲去固定，而舊把手，只用膠水來固定，比起來容易脫落。上述兩點，都證明了這次更新是有用的。



（新的把手）

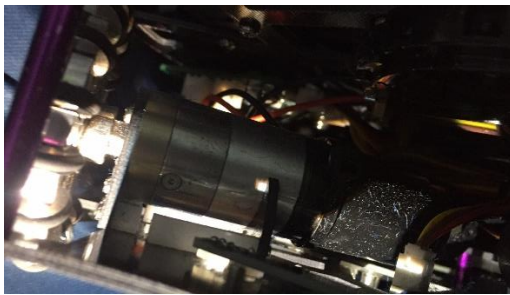


（舊的把手）

## ➤ 更換了馬達座

舊的馬達座是一個「套」，固定馬達時要把位置調好，如果不小心動了一動就會造成馬達位置的不同，有機會影響機器人行走時的合速度。因此我們想出了新的馬達座，它是一個架，分成兩個部分，前部分用來固定，後部分用來防止馬達往下壓。

相比起來，新的馬達座比舊的馬達座要方便許多。舊的馬達座是一個套，拆裝時要把整個馬達座拆下來。但新的馬達座只需要把前部分拆出來就可以進行拆裝了，大大加快了我們維修機器人的速度。



（新的馬達座）



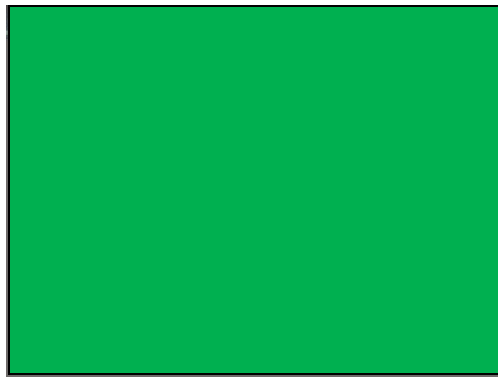
（舊的馬達座）

## ➤ 更換了灰度板

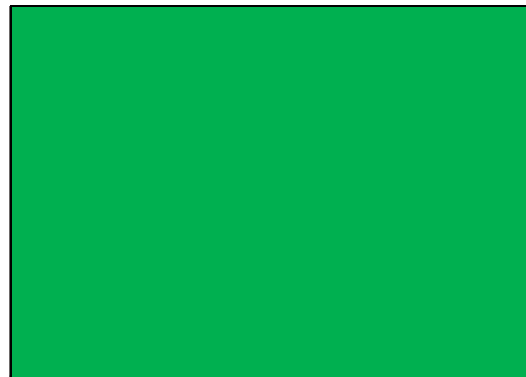
舊的灰度板是一體化的，一旦有一個光敏電阻壞了，就會很麻煩，需要把整個灰度板拆出來，換上新的光敏電阻，再把灰度板裝回機器人去。由於這樣太過麻煩，因此我設計了新的灰度板。新的灰度板有三部分，見下圖，其中較長的那塊板是放在左右兩邊用的，而較短的那塊板是放在前後兩邊用的，至於中間那塊小小的板，用以把信號線集在一起，然後直接把線接到機器人控制器上，這樣，就不怕太多線太亂的問題了。

新灰度板的最大好處，莫過於它容易拆卸及組裝，就算比賽時壞掉了，也能迅速換一塊新的上去。





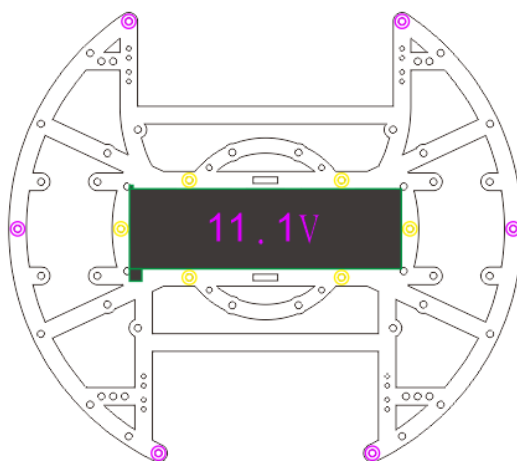
（新的灰度板）



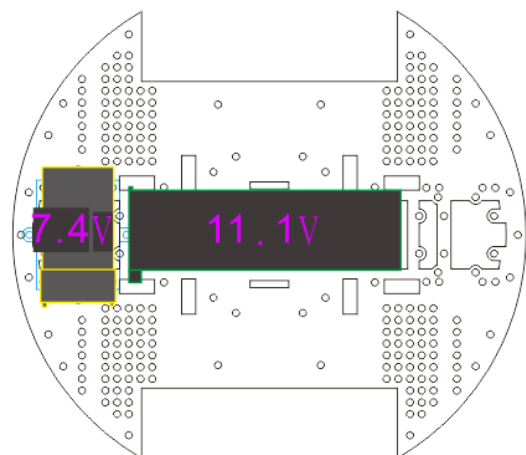
（舊的灰度板）

## ➤ 更換了鋰電池數量

以前我們都是用兩顆鋰電池，一顆是 7.4V，一顆是 11.1V。這次更新，我們只用一個 11.1V 的鋰電池去供應電量。只使用一個鋰電池的好處是方便更換，以前每次中場休息時都要換兩顆電，現在只需要換一顆。而且以前我們的大電擺放在機器人中間，小電則放一旁，現在沒有了小電，能集中機器人的重心，保持在中間。同時，只使用一顆電池也能減輕機器人的重量，以及增加我們可使用的空間。



（新的用電量）

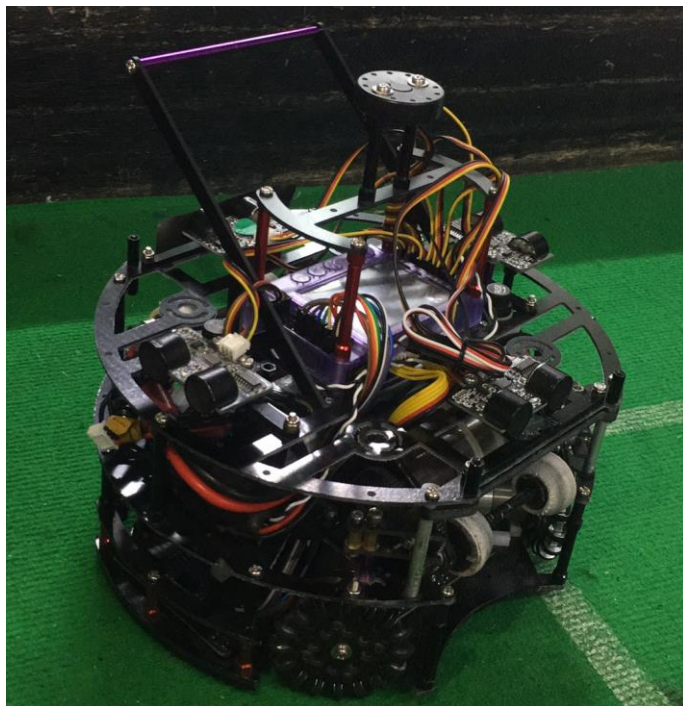


（舊的用電量）

設計好後，我們選用了環氧樹脂板作為我們機器人的板，我們的底板為 2mm，中板為 1.5mm，上板為 1.5mm，頂板為 1.5mm。

# 开发日志

以下是我們機器人的成品：





## 本次工作总结

- I. 對三種不同的馬達擺法（ $60^\circ$ 十字型擺放、 $90^\circ$ 十字型擺放、平行四輪擺放）作了研究
- II. 設計（其中，除了基本的設計外，簡水源同學設計出一塊新的灰度板）
- III. 切板
- IV. 組裝（其中，包含戴智龍和林震學習組裝盤球裝置的過程）

日志编号：06

活动日期：2016 年 11 月 7 日 - 2016 年 11 月 17 日

参与队员：戴智龍 林震

活动地点：學校

仔細看上一次的日志的話，你會發現，戴智龍與林震的機器人成品，其機器人的上板前方都多了一塊橙色的亞克力膠板。這是為甚麼呢？除了這個亞克力膠板，我們還有一個修改的地方，我們都將在這一次日志作說明。



## 讨论过程

這次討論，主要是我們發現了機器人存在的一些問題，然後尋找解決的辦法。

日期：11 月 7 日

出席人員：戴智龍、林震

內容：

戴：林震，我發現，我們機器人的控球深度超過了 3cm。

（林震拿了測控球深度的工具，驗證機器人的控球深度是否超過規定）

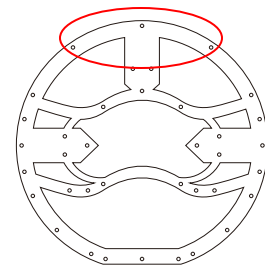
林：對哦，真的超過了。那我們在底板那裏加個東西，不讓球進去吧。

戴：嗯。欸，還有還有，你看看我們的上板。

林：咦？對哦，我們的上板也違規了。

戴：那我們把它改一改吧。

林：現在改設計的話沒問題，不過再切一塊板感覺就有點浪



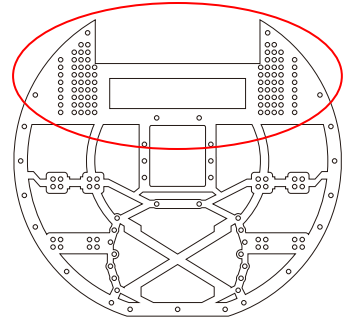
費了。

戴：我們可以先用電磨把違規的部分都磨掉，之後等我們的機器人基本程序都完成了，再切板也不遲啊。

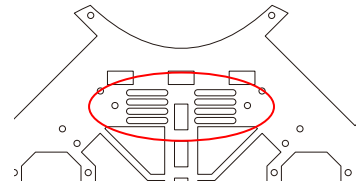
林：嗯，那就這樣吧。不過，難道我們每次都要等到切了板後，才能知道機器人的控球深度有沒有超過規定嗎？

戴：應該是吧，你看我們中板的設計，上面都有很多孔，不是為了讓我們調整盤球裝置的位置嗎？

林：嗯，上面的孔，我去年問郭濠強師兄，他說這些孔都是隨意畫的。因為你不知道盤球裝置的支架以及壓盤的部件大概會在哪個位置，再加上不同的盤球裝置，它們出來的位置也可能會不一樣，所以那些孔的位置都是隨意放的。



戴：哦~ 所以你一開始在底板畫了一些孔，就是用來加東西，頂住球，以防止控球深度超過 3cm？



林：嗯。哈哈，被你發現了。

戴：那…真的沒有方法了嗎？中板加那麼多孔的原因是因為有太多的不確定因素，那如果我們把那麼因素都確定了，是不是就可以了？

林：嗯…這個可以試試耶！

戴：你想到方法了？

林：方法倒是想到，不過不知道可不可行。

戴：沒關係，先試試吧。

林：嗯，那…我研究這個問題，你就把那些控球深度的問題解決了吧。

戴：好。

## 开发及调试过程

### 戴智龍

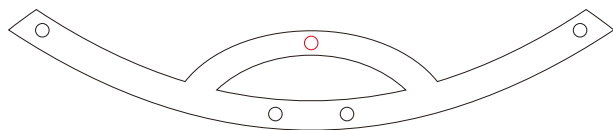
由於機器人的控球深度超過了 3cm，違犯了規定，於是我們想在底板上面加一些東西，以頂住球，限制控球的深度。至於要加甚麼東西呢？本來我的想法是，切一塊板出來，然後豎直放在底板上面。不過後來我在電腦旁邊看到了一個馬達夾，剛好它成直角形，而上面的孔也符合我們在底板的設計。於是為了節省畫圖、切板的時間，也為了節省板，我就直接拿了它來用，把它裝在底板上面。



至於上板，我按照底板控球位的弧度，在上板畫了個一樣的弧線，然後按照弧線，用電磨磨掉了上板的前部，這樣，機器人的控球深度就不會超過 3cm 了。



不過，等我再仔細地看了一看，我發現，原來我們沒有設計安裝火焰的孔。有了火焰，能夠判斷盤球裝置是否控到球，因此，我認為，有必要加上一個孔來裝火焰。之後，為了節省時間，我用了亞克力膠板作為材料，用激光切割機把板切出來。下圖中的紅色的孔，就是新加上去，用來裝火焰的孔。而這塊板也能起到一個連接的作用，因為在上次用電磨把上板前部磨去後，前部分就一直是分開來的。



之後還有一個問題就是，我們機器人的頂板雖然夠帥夠特別，卻太多尖銳的地方，我曾被它刮傷過很多次。因此，我在每個尖銳的地方都塗上一層熱溶膠。

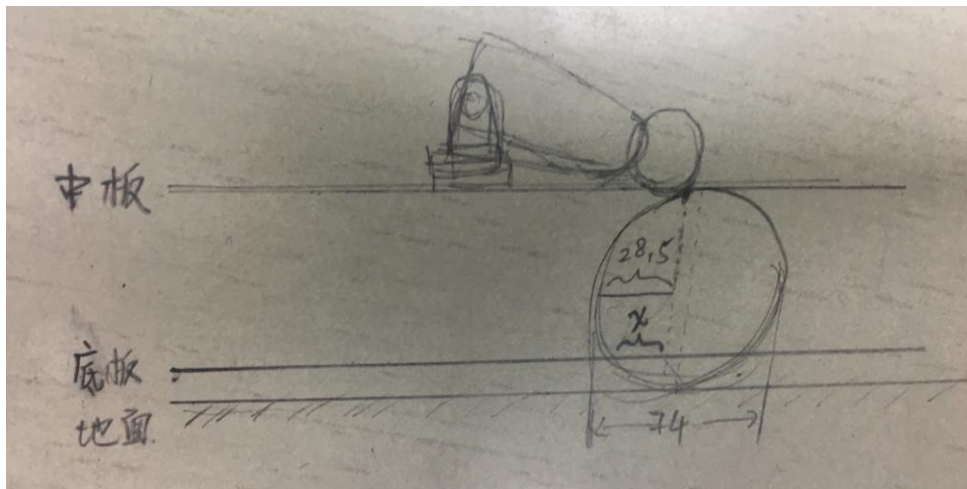


## 林震

現在我們都要等到整台機器人做出來後，才能知道機器人的控球深度有沒有超過規定。這樣太麻煩，且不科學，因此，我們希望研究出一個方法，去解決這個問題。

上面討論時也說過，如果把所有的不確定因素都確定了，也許就能利用設計去控制機器人的控球深度。那麼，我現在假設以後的機器人都是用一模一樣的盤球裝置，以我的機器人的盤球裝置作為研究對象，想辦法去研究一套方案出來。

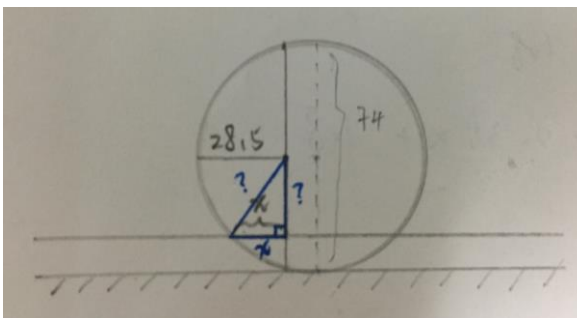
由於我希望能設計的時候，就定好機器人的控球深度，因此機器人的盤球裝置的大小、結構、位置等因素要保持不變，之後我再從底板下手，改變底板凹進去的距離，從而控制機器人的控球深度。我以我希望的控球深度 28.5mm，按照盤球裝置控到球時的樣子，畫出下面這個平面圖：



現在我所知道的量有：

- 球的直徑：74mm
- 控球深度：28.5mm
- 而我要求出的是：底板凹進去的距離  $x$

雖然我把平面圖畫出來了，但我想不到如何去算出底板凹進去的距離  $x$ 。一開始我打算用勾股定理，過控球深度 28.5 的端點作其垂直線，這樣，有了直角，以底板凹進去的距離  $x$  作為其中一條直角邊，畫出一個直角三角形（如右圖），我便可以用勾股定理去算了。但我找不出斜邊與另一條直角邊的長度，沒有足夠的條件，我算不出  $x$  的值。



於是隔天我就去請教了我的數學老師，和她討論這個問題。一開始，我們打算以相交弦定理去求出底板凹進去的距離  $x$ 。詳細做法如下：

$O$  為圓心， $AB$  為底板凹進去的距離  $x$ ， $C$  為  $AB$  延長線與圓的交點， $FG$  為控球深度。

過  $G$  點作  $FG$  的垂直線  $ED$

$$EG = GD \quad (\text{垂徑定理})$$

連接  $OD$ ，有  $Rt\triangle OGD$

$\therefore OD$  為圓的半徑  $= 37\text{mm}$

$$OG = FO - FG = 37 - 28.5 = 8.5$$

$$\therefore GD^2 = OD^2 - OG^2 = 37^2 - 8.5^2 \text{ (勾股定理)}$$

$GD = 36$

$$\therefore EG = GD = 36$$

$\therefore ED、AC$  為圓的兩條弦，且它們相交於  
 $B$  點

$$\therefore AB \cdot BC = DB \cdot BE$$

其中 AB 為底板凹進去的距離  $x$ ，BC、DB、BE 仍是未知數，因此我們亦計算不出 AB 的長度。於是，我們再想了另一個方法。這次我們不再用相交弦定理，而是利用勾股定理。詳細做法如下：

$O$  為圓心， $AB$  為控球深度， $D$  為球與地面的交點， $EF$  為底板凹進去的距離  $\alpha$ ， $KD$  為底板到地面的距離。

過  $A$  點作  $AB$  的垂直線  $CJ$

連結  $OC$ ，有  $Rt\triangle OAC$

$\therefore OC$  為圓的半徑  $= 37\text{mm}$

$$OA = OB - AB = 37 - 28.5 = 8.5 \text{ mm}$$

$$\therefore AC^2 = OC^2 - OA^2 = 37^2 - 8.5^2 \text{ (勾股定理)}$$

$$AC = 36 \text{ mm}$$



$$\because KD = 14\text{ mm} , OD (\text{半徑}) = 37\text{ mm} , AC = 36\text{ mm}$$

$$\therefore AF = 23\text{ mm}$$

過 E 點作 CJ 平行線 GE

過 G 點作 EF 平行線 GH

有矩形 GHFE，其中 OB 與 GE 交點為 I

連結 OE，有  $\text{Rt}\triangle OIE$

$$\because OE \text{ 為圓的半徑} = 37\text{ mm}$$

$$IE = AF = 23\text{ mm}$$

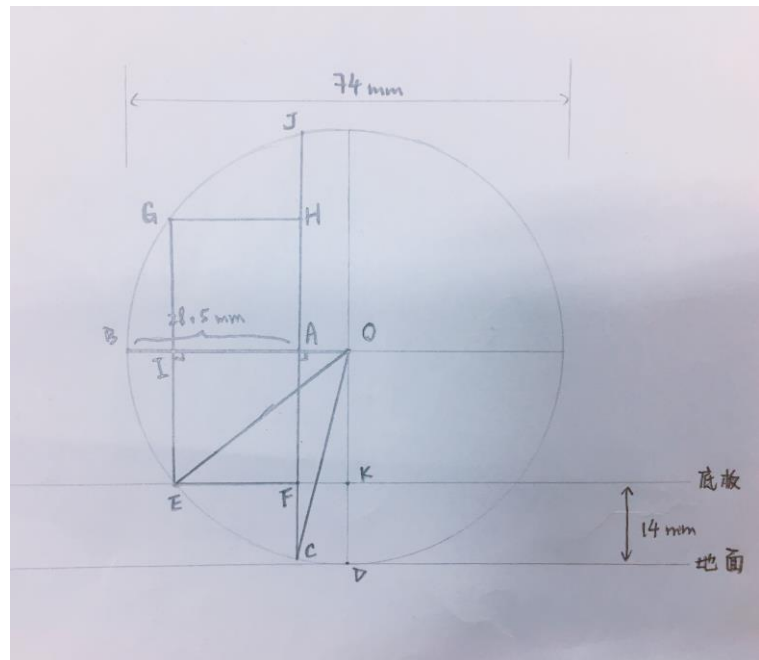
$$\therefore OI^2 = OE^2 - IE^2 = 37^2 - 23^2 \text{ (勾股定理)}$$

$$OI = 29\text{ mm}$$

$$\therefore EK = OI = 29\text{ mm}$$

$$\because KF = OA = 8.5\text{ mm}$$

$$\therefore EF = EK - KF = 29 - 8.5 = 20.5\text{ mm}$$

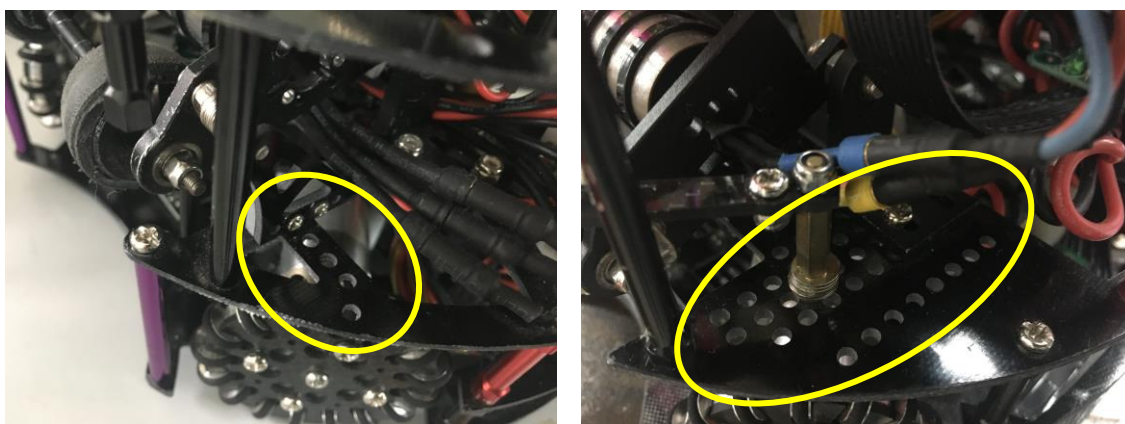


就是這樣，我們成功利用了勾股定理去求出底板凹進去的距離  $x$ 。在我總結並記錄我的計算過程時，我發現，原來還有一個更簡單、更快的方法。用上面的圖來說，由於我知道  $\text{Rt}\triangle OKE$  的斜邊 OE 是半徑 37mm，直角邊  $OK = OD (\text{半徑}) - KD = 23\text{ mm}$ ，然後用勾股定理去求出  $\text{Rt}\triangle OKE$  的另一條直條邊 EK，求得  $EK = 29\text{ mm}$ 。之後由於  $KF = OA = OB - AB = 37 - 28.5 = 8.5\text{ mm}$ ，所以  $EF$ （底板凹進去的距離  $x$ ）=  $EK - KF = 29 - 8.5 = 20.5\text{ mm}$ 。這個方法比上面那個要簡單許多，且用時很短，當我發現原來還可以這樣算的時候，我頓時感嘆自己的愚蠢。不過，這也證明了我算出來的答案是對的。

計算好之後，我向戴智龍分享了我的做法，分享的時候，剛好被簡水源聽到了，他說：「我佩服你的算數能力，不過，這樣感覺會沒了很多腦細胞。至少呢，我是想不到怎麼算的。」



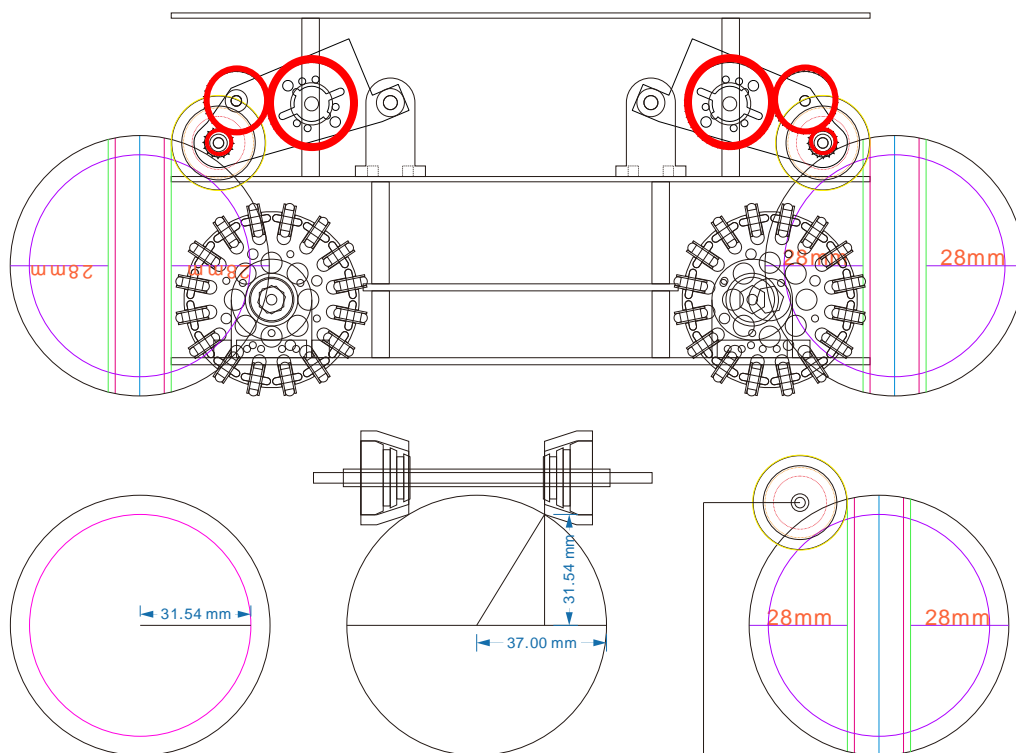
這時戴智龍看了看簡水源的機器人（下面左圖為簡水源隊的機器人，右圖為林震隊即我們隊的機器人），發現它的中板不像我們一樣，有着很多很多的孔，於是他便問簡水源：「簡水源，那你是怎麼畫的呀，我看你應該是在設計的時候就決定好盤球裝置的位置的吧？」



簡水源說：「哈哈，不錯嘛，開始觀察周邊事物了。的確，我跟你們的想法是一樣的，我在一開始就決定好機器人的控球深度以及盤球裝置的大小、位置、高低等問題，然後在設計的時候，按照我決定的那些東西，去畫設計圖。」

我好像想到些東西，卻又理不清是甚麼來的。在我左想右想東張西望之時，我看到了戴智龍的電腦裏正打開的 CorelDRAW 畫圖軟件，突然好像有靈感跑進我的腦袋裏面，我想我大概知道簡水源是用甚麼方法了。我問他：「我知道了我知道了，簡水源你硬件做的那麼好，畫設計圖也總是把東西畫得很仔細，連一孔一燈都不放過，你是不是利用畫圖，去算出機器人底板的凹入距離呀？」

簡水源說：「噢，這也被你發現了！好吧，我承認我是利用畫圖來做的，畫圖多方便呀，又不用計算，只要把所有的東西按照其大小畫上去，便可以輕易知道我們想要的數據了。」

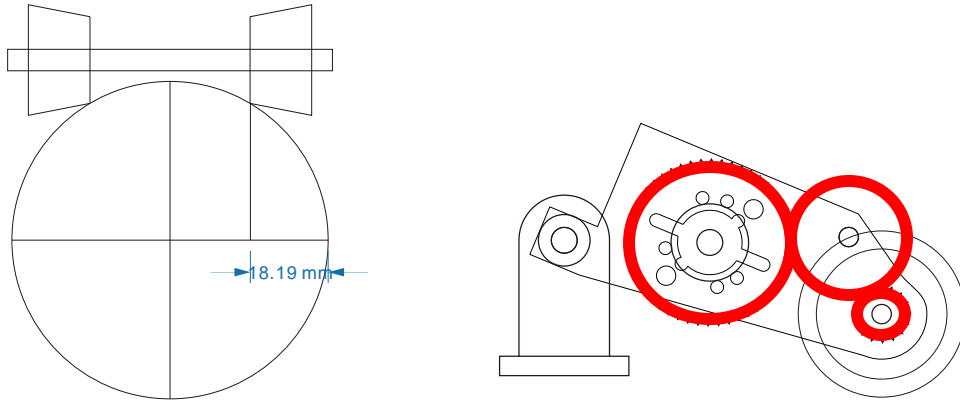


於是，我打算用畫圖的方法來做一次。在畫圖之前，我們得先理清盤球裝置、底板與球的關係。在以前機器人沒有裝盤球裝置的時候，照理說，機器人與球的接觸點應是中板與底板。不過如果中板與底板的最出點不一致且相差太遠，那麼機器人與球的接觸點可能就只有一個了。同樣道理，現在機器人裝了盤球裝置，所以機器人與球的接觸點應該是盤球裝置與底板。如果底板的凹位相對盤球裝置來說太出的話，就會頂住球，導致盤球裝置無法接觸球。而如果底板的凹位太入，那麼當盤球裝置控到球時，球基本上不會接觸到底板。

所以，我希望當盤球裝置碰到球時，球也能剛好碰到底板，與底板的凹位中點接觸。而按照我上面的計算方法，我求出來的凹入距離應是底板的最出點到其凹位中點之間的距離。因此，我必須讓底板的最出點與盤球裝置的控球點保持在同一線上（控球深度的垂直線），即它們要「一樣出」。而我看了看我的機器人，發現其底板的最出點與盤球裝置的控球點並不是一樣出。因此，這也是我們下一次設計要注意的地方。

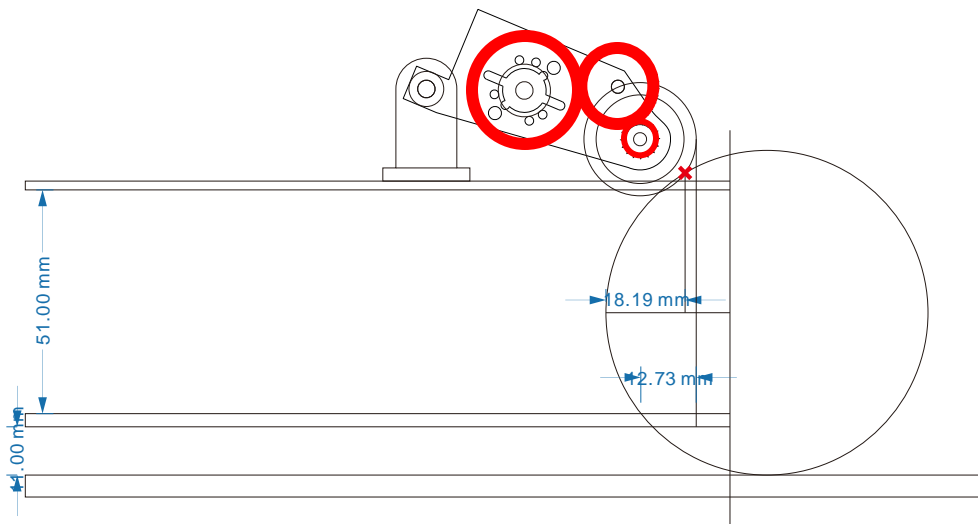
至於現在我要畫圖，我當然設定其底板的最出點與盤球裝置的控球點是一樣出的。由於需要用到盤球裝置的控球點，以及控球點在球上的位置，因此我先用游標卡尺量度盤球裝置控球軸、軟皮的大小。然後在 CorelDRAW 上粗略地畫出控球軸的正面圖。之後，我

再把球（74mm 的圓）畫出來，慢慢移到球與盤球裝置接觸的位置，這樣就找到了控球點的位置。

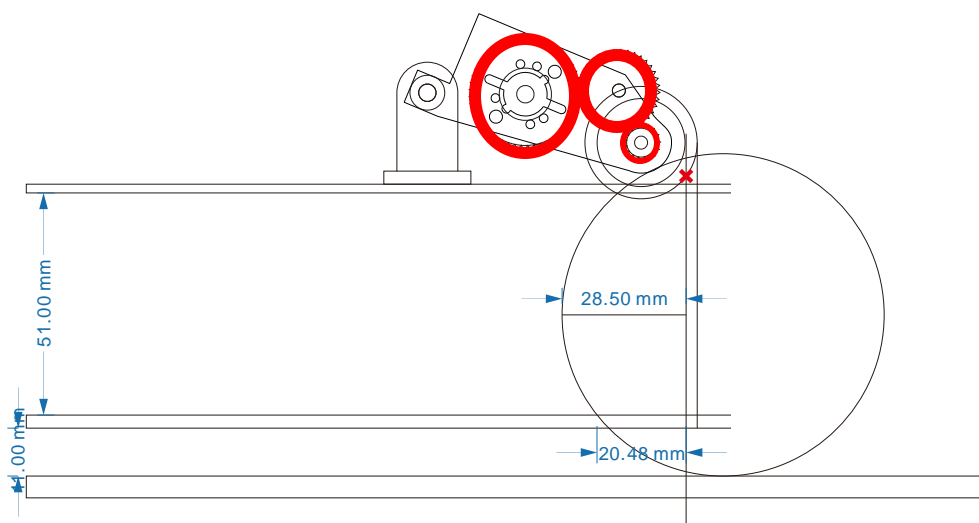


之後呢，由於我還要求出底板的凹入距離，於是我還是按照幾何數學一樣的方法，利用其側面圖去求出答案。其中，每塊板都可用矩形去代替，只要它們的厚度一樣，且板與板之間的距離一樣就行。而球的側面圖也很簡單，也是畫一個直徑為 74mm 的圓就可以了。不過盤球裝置的側面圖就有一點難畫，又有齒輪又有支架又有軟皮甚麼甚麼的，雖然齒輪呀那些東西對這次的研究沒有太大影響，不過我得知簡水源已經畫了，於是我便找他要來了盤球裝置的側面圖。

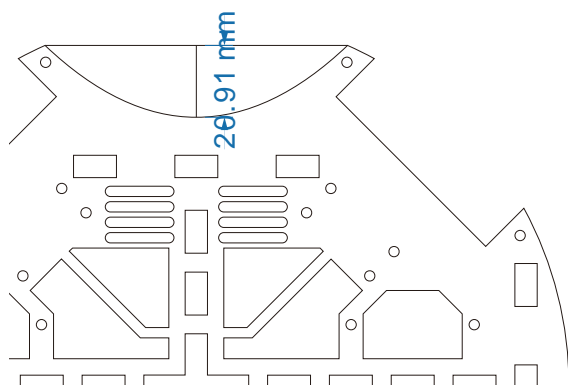
有了盤球裝置的側面圖，我便按照機器人的樣子畫出下面這個平面圖：



其中，由於在之前我已找出控球點，於是當正面圖轉成側面圖，其控球點也不過是左右調轉了而已。而上面中控球點標示了一個紅色的交叉，以記錄球與盤球裝置接觸的位置。仔細留意，你會發現，按照現在這個擺放，機器人實際上的控球深度只有 18.19mm。因此，交叉位置不變，把 18.5 的控球深度變成 28.5，之後把球向左平移，直到那條線與紅色交叉相交（如下圖所示）。這時出來的結構，機器人的控球深度便是 28.5mm 了。經過 CorelDRAW 軟件的量度工具，我們可知這時底板的凹入距離是 20.48mm，與我們算數出來的結果 20.5mm 相近，而它們之間的少少差異則是因為在上面算數的計算過程中，我把一些小數四捨五入了。



畫好後，我突然又想到了一個問題。在上上圖中，所顯示的應該是我現在的機器人。而它的控球深度是 18.19mm，底板的凹入距離是 12.73mm。這時，問題就來了。第一，一開始，我的機器人的控球深度是超過 30mm 的規定的，而現在畫出來的深度卻是 18.19mm。第二，我的底板的凹入距離實際上是 20.91mm 而並非 12.73mm。



想了想，我發現，在上面我把 18.19 的控球深度變成 28.5 時，我是直接把球向左平移，而盤球裝置是保持不動的。這樣的做法並不奏效，因為根本就不可能把球再往盤球裝置裏面推。因此，我們只能改變盤球裝置的位置，或者把控球軸升高（因為盤球裝置可上下擺動）。

到了這裏，由於我的程序還沒有寫過，加上我也沒急着要設計新機，因此我先把這個問題放下，等到程序寫好了再說。

## 本次工作总结

- I. 在底板加了一個東西來頂住球以避免犯規（機器人的控球深度不得超過 3cm）
- II. 用電磨磨掉上板前部以避免犯規（機器人的控球深度不得超過 3cm）
- III. 加了安裝火焰的孔，並設計一塊板來連接
- IV. 研究方案：如何在設計的時候就決定好機器人的控球深度。

不過還未研究成功，等到把程序寫好再繼續完善

日志编号：07

活动日期：2016 年 11 月 19 日 - 2017 年 1 月 15 日

参与队员：何建群 戴智龍 簡水源 林震    活动地点：學校

機器人的硬體弄好後，我們便開始着手程序的部分。由於程序需要不停地調試及完善，因此用時較久。再加上 12 月的 17 日到 21 日這段期間，我們參加了政府推出的一個計劃---「千人計劃」，地點在北京，其內容圍繞着科技。

因為程序部分我們都各寫各的，策略方面我們在設計的時候也討論過，因此這次寫程序的部分，沒有討論過程。

另外，我們也暫定了 1 月 17 日來一次組內賽，測試一下兩隊的成果。

## 开发及调试过程

我們的介紹順序是戴智龍 → 林震 → 何建群 → 簡水源，由於我們每個人的程序有些部分的思路都是大致相同的，因此越後介紹的部分，為免太過重複累贅，我們會只作簡略的說明或者不介紹。

### 戴智龍同學

其實我的比賽程序主要上是分開為兩條的，分別是主程序和另一條用來讀取灰度值的副程序。分開了兩條程序是有很大的好處的，分開了兩條的程序的話就可以減少了每條程序的內容，縮短了每一條程序的"長度"，這樣當我每次發現我的程序出現了問題的時候，我就可以根據出現的錯誤而決定去修改哪一條的程序，這樣我就能更快地找出問題的所在之處，不用浪費任何多餘的時間去漫無目的地尋找究竟是哪一個地方出現了問題，這樣就能大大提高我的工作效率，讓我更快地完成我的工作。

#### ■ 讀取灰度值

我的兩條程序，其中一條是用以讀取灰度值，整條的程序其實就是一直在讀取灰度值（雖然我是使用灰度板的，



但其實灰度板的本質也就是把八個灰度合而為一，因此我還是把它看作為八個單獨的灰度）。首先它就會把每次程序打開後第一次所讀取到的八個灰度的灰度值記錄下來，然後繼續讀取灰度值直至其中一個灰度讀取到的灰度值比上一次記錄到的更大，然後它就會把更大的那個數值記錄下來而去取代了原本的那個的灰度值，一直重覆。

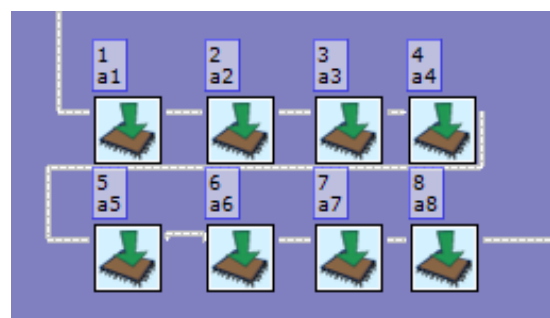
在一直讀取和記錄灰度值的同時，我也會一直來回拉動我的機器人來模擬機器人在比賽場地上移動的情況，因為同一個的比賽場地上我的機器人是有可能會讀取到不同的灰度值，就像是機器人在面向不同的方位時亦有可能會受到同一的外來的光源的不同方位的影響。但在我拉動機器人時要注意不要讓灰度有機會在白線上，因為這樣就會令到讀出來的數據不能利用（後面會說明原因）。我亦會一直拿着球在機器人的周圍移動，因為機器人在追球時可能會出現各種各樣的情況而導致它所讀取到的灰度值產生不同的變化。上面提到的兩種情況，可能會令到機器人所讀取到的灰度值變大，所以我們就要去記錄下這些情況，盡量減低機器人因為上面的情況而導致以為自己出界的機率。

在最後我還會把記錄到的外圍的那四個灰度所讀取到的灰度值，加上一個定值  $x$ ，把裏面的四個灰度所讀取到的灰度值，加上另一個定值  $y$ 。加上定值就是為了在出界與不出界的數值之間取一個中間值，這樣做也是為了進一步去避免機器人判斷錯誤。至於裏面和外圍的灰度加上不同的定值的原因，是因為它們所受到的外界影響有所不同，外圍的灰度受到球的紅外線以及外來光線的影響比裏面的灰度要大，所以外圍灰度加的定值要比裏面的要大。



其實我寫了這麼多的東西也只是為了降低機械人出界的可以性，因為只要不出界的話就算它進攻得不好但至少還能留下來防守，所以不出界對我來說就是王道，因此這條程序就是整個比賽程序裏的精髓。

因為這條程序只是比賽程序中的其中一條，因此我需要把在灰度程序所記錄到的數值傳到主程序中，這樣那些數據才能夠被主程序所使用。如果不能夠做到數據共享的話，就失去了把程序分開寫的意義，所以我把讀取到的灰度值儲存在 RCU 的數據庫中，

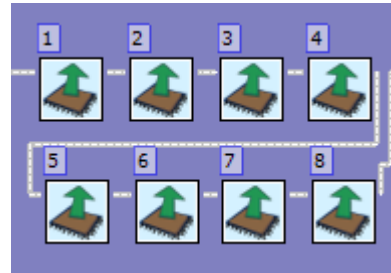




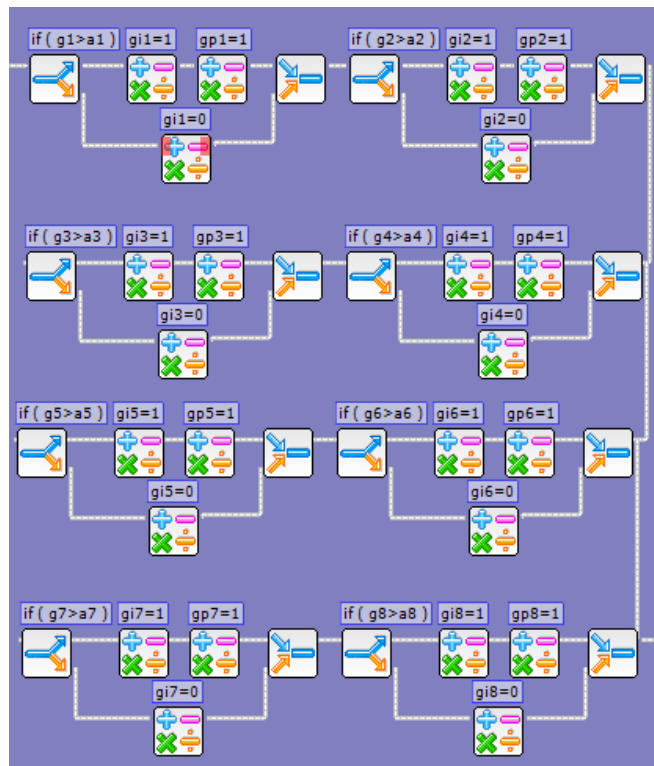
這樣只要在主程序中把灰度值從 RCU 的數據庫中取出來就可以在主程序中使用。

## ■ 判斷出界

介紹完了灰度的程序，就到了介紹主要程序的時候了。首先，整條程序的開頭就是要把之前在灰度程序中所讀取到並儲存到 RCU 的數據庫的灰度值提取出來，以供後面使用。



接下來，就是要判斷我的機器人有沒有出界的部分了。在程序的開頭所提取的灰度值就會派上用場了。首先機器人還是會一直去讀取灰度值，但與上一條程序不同的是，這次並不會記錄下任何的灰度值。而之前所記錄的灰度值就會作為一個界限值，當機器人的灰度讀取到的數值等於或大於該灰度的界限值的話，我就會把那個灰度當作是出了界的灰度（之前提到過在我讀取灰度值時絕不能把機器人拉到白線上就是這個原因，因為讀取到的灰度值是要作為機器人有沒有出界的界限，由於白色會反射所有的光，因此在白線上讀到的灰度值是最大的，順理成章的在白線讀取到的灰度值就成為了那個界限值，所以如果機器人進入了主程序然後又站在了白線上的話，它就會完全的無視了白線然後就直接出界）。然後我就會去用變量把各個灰度的情況記錄下來，每個灰度兩個，所以一共有十六個變量。其中，gi 和 gp 變量都是用來記錄灰度的狀態，只不過 gi 是瞬時變化，其狀態會一直更新，而 gp 則是要符合一定的條件才會更新其狀態。當灰度所讀到的灰度值大於或等於界限值時，瞬時變量（gi）和記錄變量（gp）都會等於 1，而當灰度值低於界限值時就有分別了，並不是瞬時變量（gi）和記錄變量（gp）都會一起變回 0，只有瞬時變量（gi）變回 0。



然後就會把八個的瞬時變量（gi）加起來變成瞬時變量總數（git），八個的記錄變量（gp）也加起來變成記錄變量總數（gpt），瞬時變量總數（git）的意義就是為了讓記錄變量（gp）可以有清零的機會，瞬時變量總數（git）為零時，（ga）就會開始就會開始疊加直到（ga）大於 10 或（git）不等於零時然後它就會變會回 0，這樣做的意義就用來防止的情況就是例如如果機械人的 3 號的灰度（最左邊的灰度）讀到白綫後機械人因某些情況而不能及時停下然後繼續向左走，

（git3）就會變回了 0 但其實機械人還是左邊出了界，它就不能回到白綫內，因此我就需要去幫我記下究竟那一個灰度才是真正出了界的灰度，而當機械人完全進入了白綫內後就會開始進行計時，過了一段的時間後它才會肯定自己已經完全進入了白綫內。

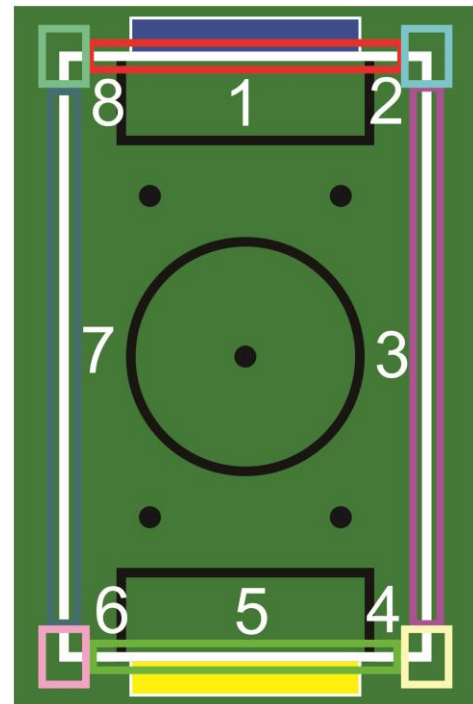
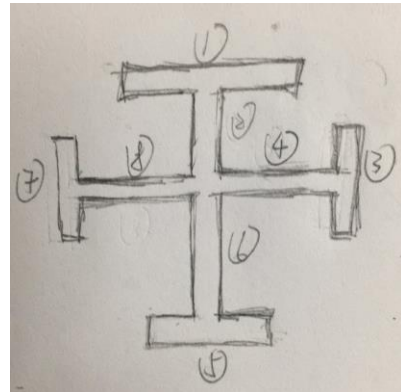
不過，單單知道自己出了界是不足夠的，最重要的是要知道自己在哪裏出界，這樣就算出了界還是可以回到白綫，所以下一步就是要去判斷機器人究竟是在哪裏出的界。機器人會先判斷瞬時變量總數（gpt）的大小，如果它的數值為 0 的話那就代表機器人所在位置的變量

（line）就會等於 0，也就是代表機器人在白綫內；如果（gpt）等於 1 的話，就代表有一個灰度讀到了白綫，然後就會繼續作出判斷，當（gp1）等於 1 時就代表機器人在上方的紅色區域，因此（line）就會等於 1，當（gp3）等於 1 時就代表機器人在右方的紫色區域，因此（line）就會等於 3，當

（gp5）等於 1 時就代表機器人在下方的綠色區域，因此（line）就會等於 5，當（gp7）等於 1 時就代表機器人在左方的藍色區域，因此（line）就會等於 7；如果（gpt）等於 2 的話，就代表有兩個灰度讀到了白綫，然後亦會繼續作出判斷，當

（gp1）和（gp3）都等於 1 時就代表機器人在右上方的紅色區域，因此（line）就會等於 2，當

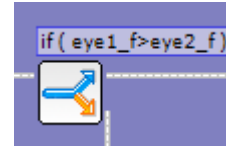
（gp1）和（gp3）都等於 1 時就代表機器人在右上方的淺藍色區域，因此（line）就會等於 4，當（gp3）和（gp5）都等於 1 時就代表機器人在右下方的米白色區域，因此（line）就會等於 6，當（gp5）和（gp7）都等於 1 時就代表機器人在左下方的粉色區域，當



(gp7) 和 (gp1) 都等於 1 時就代表機器人在左上方的淺綠色區域，因此 (line) 就會等於 8。這樣的話當機器人讀取到了白線後就可以知道自己大概的位置了。

## ■ 判斷球的位置

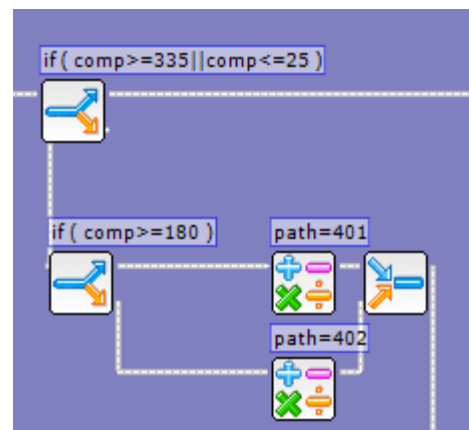
接下來就是要去搞清楚球的位置了，我判斷球的位置不再是只根據復眼是哪一隻眼看到後就確定球在哪個方位，而是會去將其細化到要知道球對於機器人的仔細角度。首先第一步就是要去確定球是在機器人的前面還是後面，因為我的機器人的復眼是前後擺放的。(eye1\_f) 和 (eye2\_f) 分別就是前後兩個復眼所讀取到的光值，而我就是通過比較這兩個變量的大小而去判斷球在機器人的前面還是後面。



在我清楚知道球在機器人的前方還是後方之後，我就需要去判斷是哪一隻眼看到球，但由於球通常都不會正對着復眼的光敏二極管，會向左右偏移，所以我需要知道球究竟是向哪一邊偏移了，通過比較每隻眼的光值，我們就能知道球偏去了哪一邊。所以，我們就利用了光值、方向等數據寫出了一條函數式，得到球對於機器人的仔細角度。在我利用這條函數式之後，得到了我想要的變量-----球對機器人的角度 (fc)，這樣就算是完成了判斷球的位置了。

## ■ 保持正方向

接下來的程序部份都是基於機器人要一直面向同一方向才可以正常的運作，而機器人需要一直面向的方向我們叫做「正方向」。因為這關係到角度，因此我們會使用和指南針有關的模塊，通過按一下指南針上的按鈕來確定 0 度的位置，確定了 0 度之後，以 0 度為中心向左右兩邊各取 25 度，所以機器人的正方向就是前面的 335~25 度。只有當機器人的指南針所讀取到的數據在這個範圍之內，它才會繼續運行接下來的程序，如果指南針的讀數不在正方向的範圍內的話，機器人就會作出調整。它會根據自己現在所面向的方向而去決定如何去修正，判斷現在面向的是 26~179 度（右邊）還是 180~334 度（左邊），知道自己所面向的方向之後，它就會通過自轉去調整自己所面向的方向，在 26~179 度的範圍內時機器人就會沿着逆時針方向旋轉（402），而 180~334 度的範圍內時，機器人就會沿着順時針方向旋轉（401）。變量 path 就是指機械人移動的方向，之後會作出更加詳細



的關於變量 path 的說明。

## ■ 分區

在判斷到指南針的讀數是在正方向的範圍內後就繼續運行接下來的程序部分。在之前我用了灰度去分了區（line），當機器人出界之後我就能知道機器人出界的位置。但當機器人回到白線內，我就不清楚它的位置了，這樣對於我來說是不足夠，我需要的是當機器人在白線內時的精準位置，因此我用機器人的超聲去確定它的位置。至於為什麼我不用灰度呢？這是因為當機器人在白線內時，由於場地是一片綠色，各個灰度在不同的位置所讀取到的灰度值相差不大，所以利用灰度去確定位置的話是十分困難的，這就是為什麼我會選擇超聲。

但在一場比賽中，超聲所讀取到的數值難免會被不同的因素所影響，例如是場上其他的機器人有可能會擋着我的超聲導致它不能正常讀數，又或者是會被其他機器人的超聲所發出的信號干擾導致其得到的數值不準確，因此在我利用超聲時我需要去知道究竟哪些超聲的數據是可用的，哪些超聲的數據是不可用的。為此，我在程序中作了一些處理。

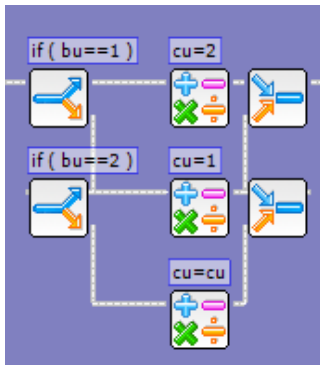
程序中就會經過一段的時間就會定期的讀取左右超聲的數值，有了記錄下的數據之後，就可以以超聲的讀數和記錄到數據來對超聲的狀態作出判斷，首先，如果左超聲的讀數和右超聲的讀數都沒有出現任何的異常，這就代表兩個超聲都處於正常運作的情況，因為當機械人在比賽場地中間的位置時而左右超聲都沒有被其他的東西所阻擋的話，而當超聲壞掉或有物體很接近超聲的時候，超聲的讀數就會等於 4095，所以當超聲的讀數不在上面的情況的範圍時就代表超聲都處於正常運作的情況，而它讀到了 4095 就代表出現了問題。而右超聲的判別方法也是這樣。



接下來就會根據機械人所處的不同的特殊的位置而去決定，第一步就是如果 [redacted]，（cu）就會等於 1（cu 就是去記錄哪一個超聲是正常的超聲的變量），然後再判定一下 [redacted]，如果條件成立的話（bu）等於 1，這個判斷的意思就是當如果機械人在場地的右邊的話，就會去使用右超聲，因為機械人已經位於場地的最右方，而機械人在場地邊界的原因只有追球或是在覆位的過程中，而當機械人在覆位時就會檢測不到球，別的機械人也檢測不到球因此它們也會

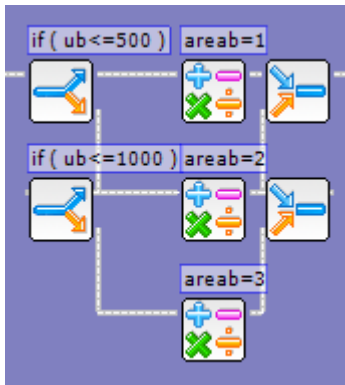
覆位，所以別的機械人並不會長期的阻擋着超聲，所以並這不會對超聲有太大的影響；而機械人是因為檢測到了球之後才會去追球，所以球也會在場地的右邊，所以別的機械人也會向球的方向移動而就可能就會一直擋着左超聲直至球離開了場地它們才會離開，它們追球至到離開的這段時間是十分長的，因而我會選擇去使用右超聲，其中也有一種的可能性就是別的機械人在我方機械人的右方出了界的話，但這個的情況對右超聲的影響也不太，因為如果別的機械人是真的出了界的話就會很快的被裁判拿出場地，如果它並不是出界而是被推出的話裁判也會把它推回白綫內，因此這並不會對右超聲造成太大的影響。而當機械人在場地的左邊時也是如此。

然後再根據（bu）的大小去決定（cu）的大小，也就是當左超聲不能使用時就用右超聲，右超聲不能使用時就用左超聲。

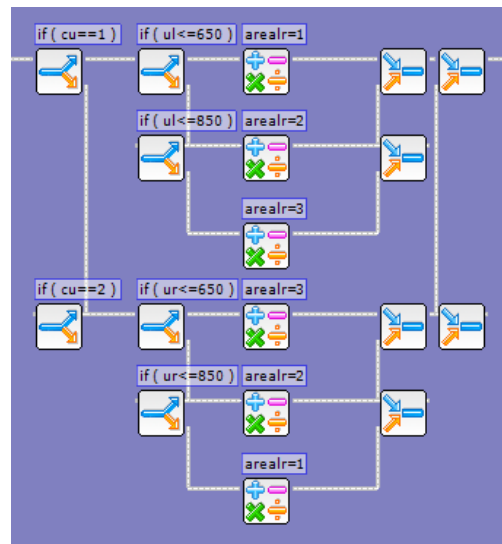


知道了超聲的狀態之後就可以正式的進入到去超聲判別位置的部份。首先就會先利用後超聲去把場地分為三份（因為我的機械人的覆位的位置比較後，因此後超聲並不會受到別的機械人影響，而前超聲就會一共受到三台的機械人影響，所以用後超聲去分區會比較好），然後就會根據後超聲的數值去把場地由後至前分為三份，當（ub）小於或等於 500 的話，（areab）就會等於 1；當（ub）小於或等於 1000 的話，（areab）就會等於 2；當（ub）大於 1000 的話，（areab）就會等於 3，所以（areab）將場地由後至前分為 1 號區、2 號區和 3 號區。

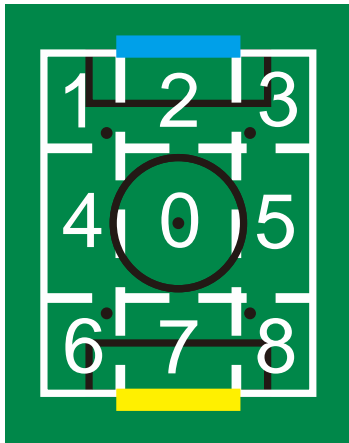




然後就到了使用左右超聲去分區，首先就是先根據可用的超聲也就是（cu）大小去決定去哪一個超聲分區，如果（cu）等於 1 就會用左超聲去分區而（cu）等於 2 就會用右超聲去分區，然後就會根據左超聲或右超聲的數值去把場地由左至右分為三份，使用左超聲的話當（ul）小於或等於 650 的話，（arealr）就會等於 1；當（ul）小於或等於 850 的話，（arealr）就會等於 2；當（ul）大於 850 的話，（arealr）就會等於 3。使用右超聲的話當（ur）小於或等於 650 的話，（arealr）就會等於 3；當（ur）小於或等於 850 的話，（arealr）就會等於 2；當（ur）大於 850 的話，（arealr）就會等於 1。所以（arealr）將場地由後至前分為 1 號區、2 號區和 3 號區。



由（areab）和（arealr）這兩個的分區交疊在一起而將場地分割為一個九宮格，而這裏又會有一個新的變量去表示這個新的分區，那就是（aream）。在我利用了超聲去分好了區之後就可以開始追球的部份。



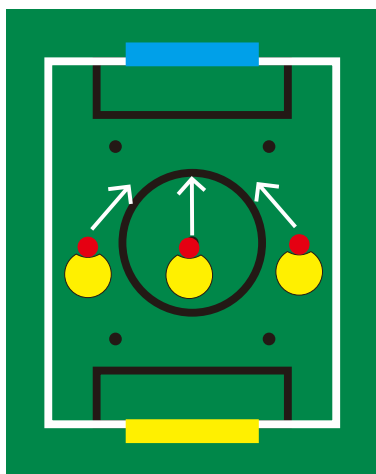
## ■ 復位

當不符合上述的條件之後就會進入到覆位的程序，覆位的動作是根據之前得到的（aream）而去決定的，比如當（aream）等於 1 時，機械人就會向 135 度方向移動，所（path）就會等於 135；當（aream）等於 3 時，機械人就會向 225 度方向移動，所以（path）就會等於 225；當（aream）等於 4 或 6 時，機械人就會向 90 度方向移動，所以（path）就會等於 90 等等。簡單來說，當機械人在場地的左右兩邊就會向中間移動，而當機械人在中間後就會再根據後超聲的數值去作出調整，如果（ub）小於或等於 400 時（400 這個數值就是經過我的測量得出的，而大概的位置就是在白綫內），機械人就會停下來，如果不是的話就會繼續的向後移動，原本我的程序應該是覆位到了場地中央就可以了但為了配合我的隊友作為主要的進攻機械人，所以我的機械人就會將覆位的位置調後，這樣就不會阻碍到隊友機械人的進攻也可以在一定程度上增加我的機械人的防守性。

## ■ 追球



上面都是當不符合可以追球的條件而做的覆位動作，而接下來就是符合追球的條件而採取的動作，這時就主要會用到（fc）即球相對於機械人的角度和（path）即機械人動作的角度，首先會是根據（fc）而去決定機械人的動作的角度，如果球在機械人的前面並且離機械人足夠近時，在根據機械人所在的位置再作出不同的動作，如機械人在場地的左邊就要向右上方推球，而如果它在場地中間的話，機械人就只需要向前方推球，而機械人在右邊的話就需要向左上方推球。



那如果機械人還不足夠接近球時，就要向着球所在的方向走以接近球；如果球在機械人的前面的兩側時，此時只需要向着球所在的方向走就可以令球進入到機械人的前面；而球在其他位置時，機械人只就沿球相對機械人的角度的切綫方向走。這樣主要的追球的程序就完成了，接下來就是防止在追球時會出界的措施。

## ■ 動作

接下來就是如何根據在不同的情況之下我們想機械人作出的動作實現的程序部份，原本只單單最直接用馬達就只能讓機械人做出八個方向、自轉和停下等簡單的動作，但向除了那八個主要方向之外的角度移動的動作就做不到了，這樣的話在機械人追球時也只能利用那八個動作去追球。

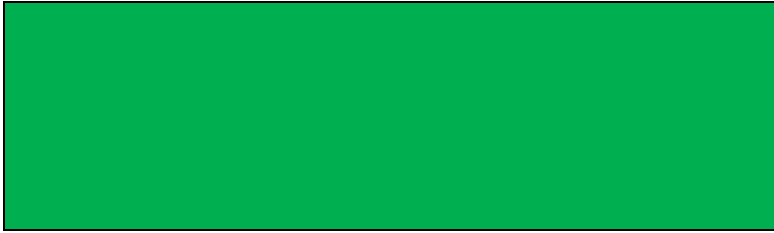
雖然這樣也能追球但追球的過程就會變得十分的不流暢，為了去細化追球的動作，我就要把那八個動作分得更細，分成 360 個動作，再配合（fc）就能做出更精準的追球的動作。

首先就以前後左右這四大方向去把這 360 個動作分為四份，每份就是 90 度，然後就先決定機械人在向這四個方向前進時的馬達的轉向。而在四大範圍內的動作的馬達的轉向都是和該範圍中間的四大方向的馬達轉向一樣。決定了馬達的轉向之後就要通過調整馬達的速度而去讓機械人作出除那四個方向以外的動作，四個動作的範圍都被四大動作再分開了兩部份，一共就是八份，然後就要知道機械人所要進行的動作是佔了以每一份的多少，再經過計算，就可以得到了另一個的速度，再經過改變馬達的速度就可以實現向不同方向的移動。



## ■ 邊行邊調

接下來就是會去再優化一下追球的程序，在機械人追球的過程有可能會因不同的因素而令到機械人不再對着正方向，而在追球的時候通過自轉來讓機械人重新面向正方向就會令到整個追球的過程變得斷斷續續，所以我就要令到機械人追球的過程中也能調整自己的角度，這樣又要需要去調整馬達的速度了，例如：當機械人向右邊偏了然後又要向右邊移動的話，四個馬達相比之下就會發現前面的馬達偏了向右，因此就會去把前面兩個馬達的速度調慢，這樣就可以讓後面的馬達趕上前面的馬達，這樣就能慢慢的調整到機械人面向的角度。所以就只需要依這個方法去把相對較快的馬達的速度調慢就可以了，所以將要變慢的馬達的速度乘以（odds）除以 10 就可以令到它的速度變慢了。



## ■ 盤球

因為我在組內賽之前才剛學會了使用盤球裝置沒多久，並且我在組內賽之前的那一段時間都忙於另外一個的比賽，因此我雖然完成了盤球裝置，但我還是沒有足夠的時間去完成相應的盤球的程序。因此我在組內賽時的程序是沒有盤球部分的程序。

至於我的隊友，他和我一樣都有別的事情要忙，而他為甚麼寫得完呢？我認為是他在重量組比我多一年的經驗。

## 林震同學

在機器人組裝好後，我便開始寫程序。由於在程序部分，我和戴智龍是不一樣的，因此我們將會分開來介紹我們的程序。

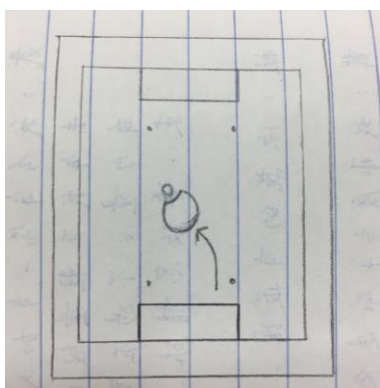
討論時決定，我和戴智龍做的都是單盤攻擊機。而由於我們沒有做盤的經驗，而且簡水源和何建群也正在忙他們的事情，因此我們決定先把基本的攻擊程序寫好，然後再做盤、寫控球動作的程序。

我寫了大概四年多的程序，經驗告訴我，編程要一步一步，循序漸進的來。若是你一開始就把大部分的程序都寫好，然後再去調試，那麼當你發現程序有問題，想要找出哪部分寫錯的時候，你會很茫然，不知道哪裏有錯，因為這個時候你的程序已經寫了很多，因此你會像在大海撈針一樣。相反的，如果你一步一步的寫，每當你寫完一部分，你就去調試，去檢查有沒有錯，那就不會出現剛才所說的問題了。

接下來，我會粗略介紹一下我的攻擊程序：

### ■ 鎖頭

首先，是鎖頭的部分，這部分的程序是機器人執行任何動作的前提。那麼，為甚麼鎖頭如此重要呢？鎖頭又是甚麼呢？鎖頭這個詞，聽上去或許會讓人很不解。其實鎖頭，意思就是鎖住機器人的偏角範圍。平常機器人在場上行走時，有可能會因為四個馬達間有速差而導致其路線（以直線為例）變彎，形成一條弧線（如下圖所示），這個時候，機器人的角度可能就已經偏了，跟原始的不同。也有可能在與其他機器人碰撞期間，導致機體自身偏移了。其會帶來很嚴重的影響。試想像，一個機械人的角度偏了，那麼本來該在正前方的控球位也會跟著偏了，這樣的話，別說進球，可能連追球都會做不好。因此，我們必須限制機器人的偏角範圍，不能太偏，否則會失去方向。



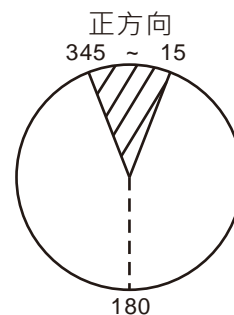
# 开发日志

那麼，該如何鎖住機器人的偏角範圍呢？首先，很簡單的，偏角跟角度有關，我們可根據指南針返回的數據去知道機器人現在所面向的角度，從而知道機器人有沒有偏了。但要注意的，機器人行走的時候，難免會有一兩度的偏角，再加上機器人在場上不同的地方會檢測到不同的角度，因此我們需要根據指南針數值的變化，設一個範圍，叫做機器人的「正方向」。我們在機器人控制器上查看指南針的數值變化，其中左圖為機器人在場地的左上角時指南針的數值，右圖為機器人在場地的左下角時指南針的數值。可以看到，其偏差不小，因此設正方向是有必要的。因為指南針數值在場內普遍上都在 345 到 15 之間，所以，我機器人的正方向範圍為 345 到 15。



如果機器人在正方向內，便可執行指定的動作如追球。若超出範圍，不在正方向內，便會先自轉，等到轉回正方向裏面再做本來的動作。簡單的說，機器人的角度在正方向範圍內，是它做任何動作的前提條件。

```
if ( compass > 345 || compass < 15 ) //指南針數值在正方向範圍內
{
    action; //指定動作，例如追球、復位
}
else
{
    rotate; //自轉
}
```



如此一來，機器人便能做到「鎖頭」了。但這個時候，其實這部分的程序還不夠完善。出來的效果是無論機器人有多偏，偏向左或右，都會向同一個方向自轉。比方說，假

設上面程序所指的自轉是順時針方向，當機器人偏向左，那麼它就會順時針自轉，轉回正方向。看似沒有問題，很完美，但想多一層，便會留意到當機器人偏向右邊，它還是順時針自轉的話，那麼便會耗費了很多時間。因此，我們會以 180 度作為分界，當機器人偏向左，順時針自轉。相反，當機器人偏向右，逆時針自轉。

```
if ( compass >345 || compass < 15 ) //指南針數值在正方向範圍內
```

```
{
```

```
    action; //指定動作，例如追球、復位
```

```
}
```

```
else
```

```
{
```

```
    if ( compass < 180 )
```

```
    {
```

```
        counterclockwise rotate; //逆時針自轉
```

```
    }
```

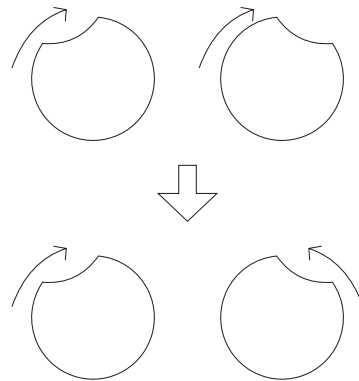
```
    else
```

```
    {
```

```
        clockwise rotate; //順時針自轉
```

```
    }
```

```
}
```



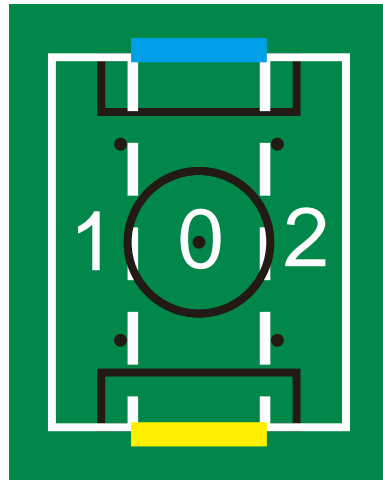
## ■ 分區

分區，很簡單、很容易就能理解，就是把場地分開成幾個區域的意思。分區能讓機器人知道自己在場上的哪個區域，以便於做出相應的動作，更甚是執行相應的策略。

我自己呢，就利用超聲，把場地分成左、中、右三個區域，大概以場地龍門的左右兩端作為分界線。

```
if ( ultrasound > 950 )
```

```
{  
    area = 2;  
}  
  
else  
{  
    if ( ultrasound > 650 )  
    {  
        area = 0;  
    }  
    else  
    {  
        area = 1;  
    }  
}
```



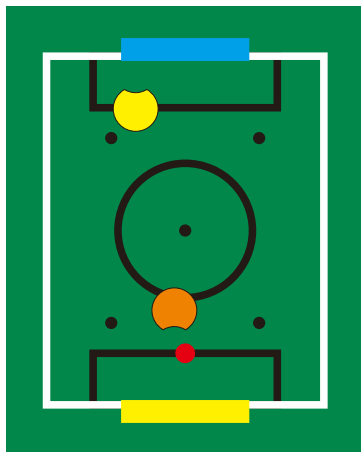
## ■ 復位

復位，即回復、回歸到原來的位置，而這個位置是我們在程序裏面一早就設定好的。在機器人看不到球的時候，它就會執行復位的程序。大家都知道，檢測到球是機器人追球的前提。那麼當機器人檢測不到球的時候，它自然就會停下來。而我們認為，在沒有球的時候，只是純粹的原地不動，這樣不好。原因主要有兩個：

- (1) 機器人有可能需要花較多的時間去找球。比方說，機器人有可能停在了場的左上角，而這時裁判如果把球放在左下角的點，那麼機器人就會耗費更多的時間去找球。
- (2) 機器人看不到的球的原因可能是因為其他的機器人遮擋住球，那麼如果機器人一直保持不動的話，只有當球靠近機器人，機器人才會繼續追球，展開攻勢。試想像，當機器人停在了對手的半場位置，而球在我方龍門的半場位置，如果對方機器人持續進攻，那麼球只會靠近我方的龍門，而停在對手半場位置的那台機器人就會一直檢測不到球。這樣導致的後果，不但我方攻勢大大減弱，就連防守也會



失去。

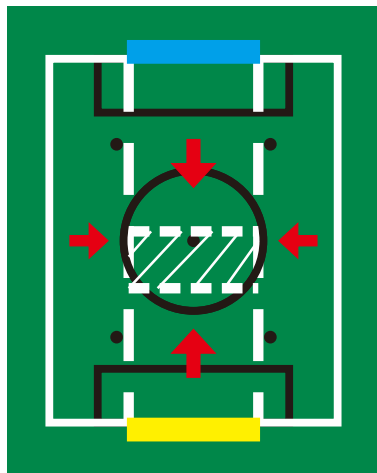


由於上面所說的原因，在機器人看不到球的時候，它不會原地停下，而是會回到我們指定的位置。而至於位置在哪裏，一般都是看個人喜好，而且會因應其戰略而有所不同。一般來說，防守機都會復在禁區裏面，而攻擊機為了更快地拿到球，一般都會復得比較前。至於我的機器人，是復在中間那個中立點後一點點的位置。這樣一來，之前所說的兩個問題，也會得到改善：

- (1) 機器人有可能需要花較多的時間去找球。對於這個問題，有了復位程序也並不能保證機器人會比別人快找到球，但最起碼，機器人回到中間的位置，無論裁判把球放在哪一個中立點，機器人找到球的時間都很平均，因此不會耗費多餘的時間。
- (2) 同樣道理，機器人回到中間的位置，到場上每個角的各個距離也都很平均，被別的機器人遮擋球後，再次看到球的機會就會相對平均。

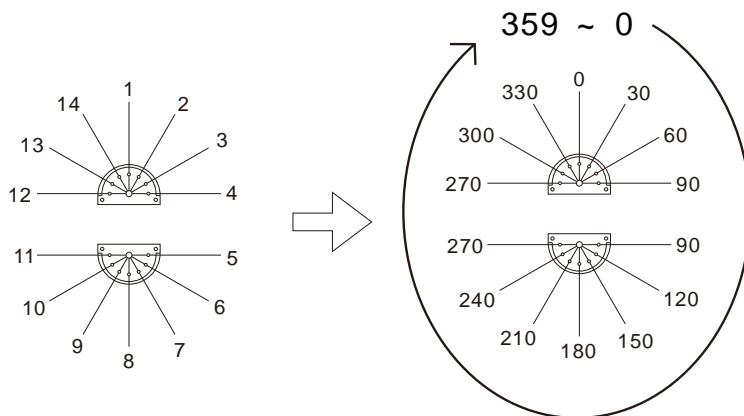
可以看到，復位程序雖不像追球那麼重要，但卻讓機器人在各個情況都能得到相對穩定的解決。

那麼，很簡單的，既然機器人要回到場內的中間位置，意味着當機器人在左邊，就往右走；當機器人在右邊，就往左走；在前面，向後走；在後面，則向前走。其中，我們利用分區去判斷機器人在左或右或中間；用前後超聲去判斷機器人在前或後或中間。



## ■ 復眼處理

我們共使用了兩隻復眼，一隻復眼有七個光敏二極管，兩隻則有十四個。也就是說，我們能往十四個方向去看球，去判斷球在十四個之中的哪一個方向。但我們認為，只有十四個方向還遠遠不夠，一個方向所涉獵的範圍太大，我們不能很精確的知道球的方向。因此，我們對復眼的數據作處理，讓本來的 14 個方向，變成我們熟悉的 360 度。



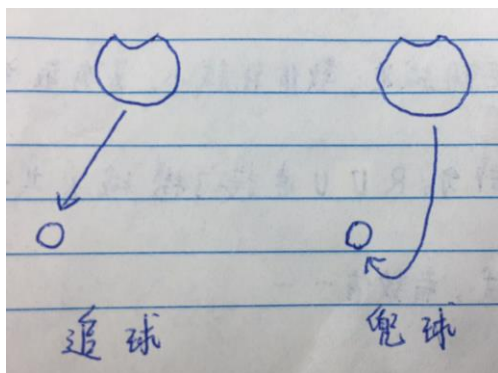
其基本原理是，先把 14 隻眼的 1、2、3、……、14 號眼變成 0、30、60、……、330，然後再根據球偏向哪隻眼，其光值又是多少，從而化出中間的數值。比方說，球現在在 0 到 30 度之間，於是我們就看 1 號眼和 2 號眼所讀取到的光值，哪一個光值大，球就偏向哪一隻眼。假設球偏向 1 號眼，那麼經過處理後，它的數值肯定在 0 到 15 之間，不會大於 15。至於怎麼細分其角度，也就是怎麼去知道球偏向 0 偏多少，是更接近 0，還是更接近 15，我們會對比 1、2 兩隻眼的光值的差值大小，若球接近 0，那麼其差值一定

會大，若球接近 15，其差值相對就會小。按照這個思路，對其數據做一點處理，便可把 14 個方向化成 360 度了。

## ■ 兜球

兜球，跟追球差不多，其差別在於追球是往球的方向追，兜球則是兜着球走。我們都知道，球進了龍門才有分，所以必須要對球有反應並做出相應的攻擊動作。就像平常人們踢足球一樣，人們都會往球的方向跑，比別人先拿到球，那麼你贏的機會又大了一半了。球在前面機器人向前衝，這個道理我們都懂，但當球不在前面，你只顧盲目地往球的方向走，只會把球撞走，還有可能導致烏龍球的發生。因此，當球不在前方，我們會往球的切線方向走，即兜着球走，一來避免把球撞走，二來更能防止烏龍球的發生。

以前我們會分開 14 種情況，根據球的方向，做出相應的動作。而現在我們對復眼數據作了處理，把 14 變成了 360。由於精準了許多，我們也相應地寫了處理式，去處理機器人的兜球動作。基本上是根據球的方向及遠近，決定機器人的動作。



## ■ 突變

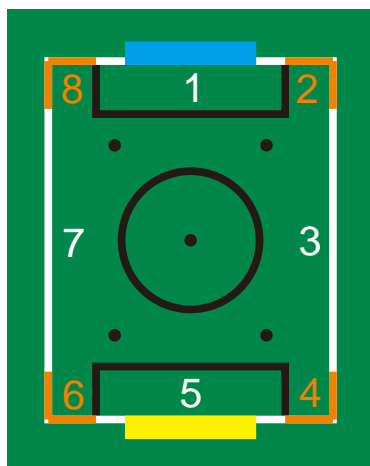
突變，是為了應對超聲發生突變而有的程序。在比賽期間，機器人的超聲很容易就會受到干擾，發生突變。就比如突然有機器人走到你的旁邊，或者機器人走到了龍門旁邊等等，超聲的數值都會突然發生了一定的變化。因此，為了防止超聲突變而導致分區有誤或其他部分受到影響，我們加了「突變」的部分。

這部分內容主要是判斷機器人的超聲有沒有突然發生了很大的變化。若有，則視為「突變超聲」，我們將會暫停取用那隻超聲的數值，而選用另一個沒有發生突變的超聲的數值。比方說，左超聲發生了突變，那麼我們將會停用左超聲，用右超聲來測距。這也是我們有前後左右四個超聲的原因。



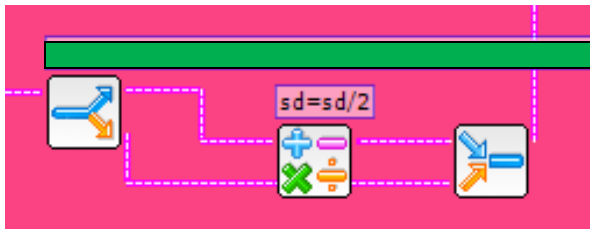
## ■ 分界

我們知道，若是機器人出了白線，會被視為「損壞機器人」，必須離開場地一分鐘。所以，我們必須做到不出界。為了不出界，碰到界時能即時返回場地，更好地防止出界，我們把白線分成八個區域，1 到 8，如下圖所示。當機器人踩到 1 號界，便會向後走，踩到 3 號界，便會向左走，如此類推，機器人碰到界後便懂得回去了。



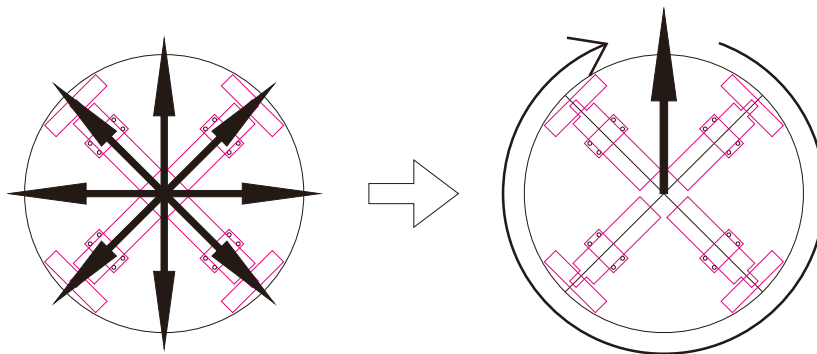
## ■ 減速

由於分了界，寫了復位，現在機器人一踩到白界，就會執行復位動作。但這個時候，會萌生一個問題：當球滾到界外，機器人跑去追球，踩了白線，執行復位。如若這時球還在慢慢地滾，或者還在五秒之內（五秒內裁判不會拿球），那麼機器人檢測到球，將會再次出去追球。由於機器人速度很快，五秒內它可能已經這樣來來回回了很多次。儘管有着復位程序，機器人一直這麼快速地圍繞白線往復運動，還是有着出界的風險。因此，我們加了「減速」的部分，當機器人踩到白界復位之後，再次往上次出界的位置走的時候，機器人便會減速。



## ■ 動作

為了讓機器人的兜球路線更加順暢、更加完美，我們希望能讓機器人走出更多的動作，向更多不同的方向走。從前我們一般只向東、南、西、北、東北、東南、西南、西北八個方向運動，為了實現我們的想法，我們研究了不同動作時馬達的轉向以及其速度變化，根據函數式對馬達速度作處理，把本來的八個動作變成 360 個動作。

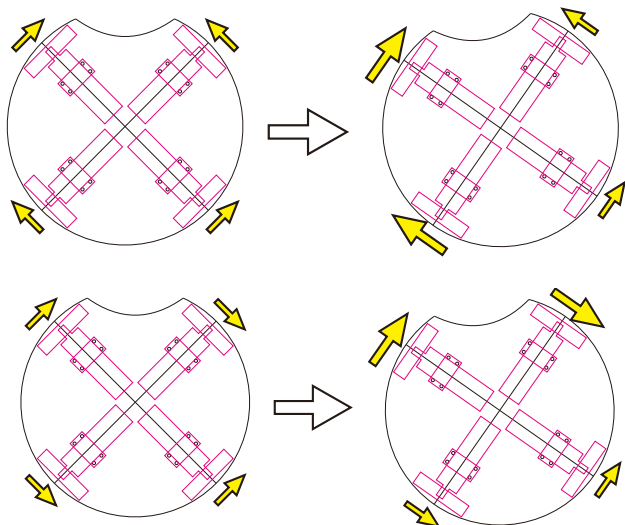


## ■ 邊行邊調

由於機器人行走時可能有速差，導致機體偏了，因此我們設定了一個正方向。而我正方向的範圍是小於等於 15 或大於等於 345，這範圍加起來有 30 度。我想要讓機器人的頭鎖得更小，所對方向更加的準確，但我不能夠把正方向的範圍縮小，如上面介紹過的，機器人的角度會不斷變化，因此如果我縮小了正方向的範圍，就會導致機器人不停地搖擺，其搖擺的實質是它一直在順時針/逆時針自轉，轉來轉去都轉不進正方向的範圍（範圍太小）裏面。

正方向範圍小機器人會不停自轉，以致它不斷搖擺，不能正常的做追球動作。因此，我從另一個角度去入手，想一個不與追球動作有衝突的方法。為了不衝撞追球動作，我在機器人追球時調整角度，機器人一邊行走一邊調角度。怎麼說呢，比如當機器人向前走時，它偏向了左邊，那麼，我就會加快左邊那兩個馬達的速度（如下圖），以調回去。再

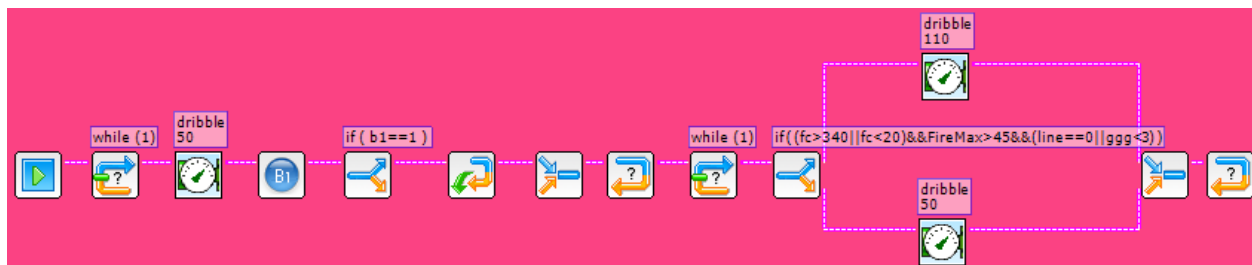
舉個例子，機器人向右走，偏向了左邊，那麼我就加快上面那兩個馬達的速度。總的來說，我利用速差去達成「邊行邊調」。



## ■ 控球動作

由於我是第一次使用盤球裝置，因此對於程序裏面如何去控制盤球裝置，我是一竅不通的。本來在學會組裝盤球裝置後，我便打算學習它的程序部分，但那時簡水源正忙着設計灰度板，而何建群也在研究 pixy，因此我先把這事擱置下來，打算等到把基本程序完成好了再去學。

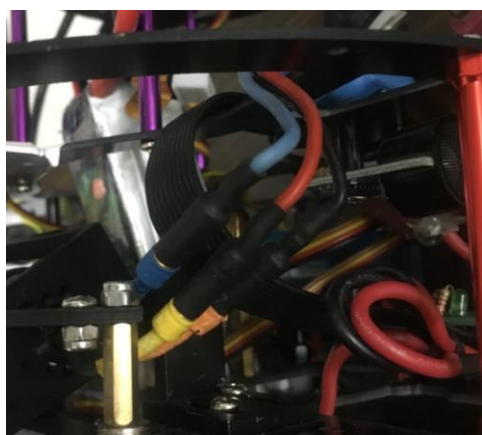
而現在，我把分區、追球、復位等等的程序都寫好了，因此我就去請教了簡水源。他當時就給了一條短短的程序讓我看：



上圖的伺服馬達模塊則代表我現在用的盤球馬達，而當中的 if 所寫的語句  $((fc > 340 || fc < 20) \&\& FireMax > 45 \&\& (line == 0 || ggg < 3))$  則是它開盤（啟動盤球馬達）的條件，因為我是前盤機器人，因此當球在機器人的前方且光值足夠大的時候，我便會開盤。你可以看到，當盤球馬達開着的時候，它的 angle 值為 110，而關着的時候則是 50。然後

呢，在程序一開頭，你需要有一個循環，而這個循環純粹是讓盤球馬達關着，直到你按下控制器的按鈕，才會跳出循環。這樣寫的意義是要給足夠的時間讓盤球馬達「熱身」。就是這樣，我學會了如何啟動盤球馬達去控球。

在這之後，簡水源又告訴我說，你現在這樣未必可行的。我感到懊惱，十分不解，簡水源就叫我去場上盤一下球試試。我聽了他的話，把機器人拿到場地上面，之後把球移近機器人以符合開盤的條件，之後，機器人也是成功的盤到球。我在想，到底簡水源指的是甚麼呢，難道是我軟皮弄不好？也不是，它還是牢固地在控球軸上面。到底是甚麼原因呢？我仔細觀察了盤球裝置一會兒，發現球原來是向外面旋轉的。於是我去問簡水源該如何解決這個問題，他說，盤球馬達連着電調的那三條線，只要把它們之間連接的順序換一下就好了。於是最後，我試出來的結果是：盤球馬達的藍色線連着電調的藍色線，盤球馬達的黃色線連着電調的紅色線，盤球馬達的橙色線連着電調的黑色線。

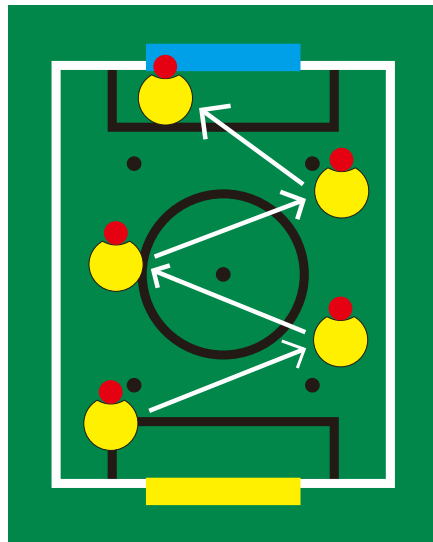


這樣，我就成功控球了。之後，就是機器人控到球以後所要做的動作。我有兩個想法，一是轉身甩球，二是斜行進攻。其中第一個方案，我的寫法是當火焰測到的值夠大（即機器人控到球），機器人就開始自轉，把球甩出去。但這樣寫，無論轉身的速度是快還是慢，我始終做不出理想的效果。再加上當時已經快到組內賽的時間，沒有足夠時間讓我去研究、調試，因此我放棄了這個方案，去嘗試第二個方案。

對於斜行進攻，我的想法是，當機器人控到球，就走「Z」字形路線，以讓別的機器人沒那麼反應。再加上斜行進球的話，對於那些以左右動作為主的防守機可能比較有優勢。雖然它的攻勢好像沒比轉身甩球要大，不過至少我們不用擔心「烏龍球」的問題，也不用花過多時間去測試甩球的弧線。



而我的寫法是，當機器人成功控球後，如若機器人在場地的左邊，就向右上方斜走，如若機器人在場地的右邊，則向左上方斜走。那麼如果機器人在場地中間呢，向前直衝？當然不是，向前直衝的話就不叫斜行進攻了，也做不出「Z」字形的效果。因此，當機器人在場地中間，我就會看機器人最近一次是在場地的左邊還是右邊，若左邊，則繼續向右上方走，若右邊，則繼續向左上方走。我的斜行進攻程序大概就是這部分，出來的效果好像也不差，最起碼整個「Z」字形的效果出來了，進球機率也有一半以上。

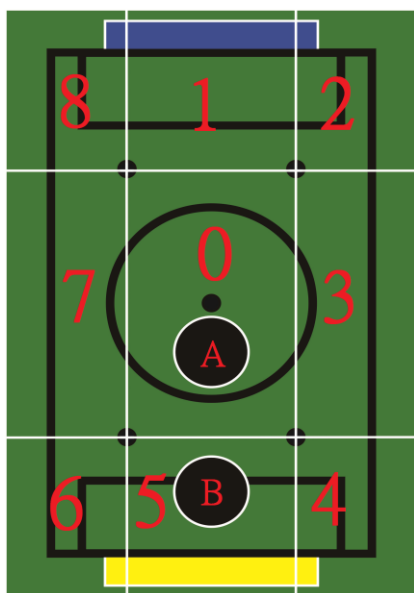


## 何建群同學

我的程式部分與上年差不多，因為我上一年也是進攻機器人，而我的程式與是在上一年的基礎上去改進的。而上一年我的打法是一攻一防，所以我的程式是有特別的進攻模式，例如我上一年的戰術是用球和一個連貫的弧型進攻動作，但這個是非常之複雜的程式，也要在不同的場地上進行修改。而上一年的全國賽程式是即場去寫的，所以因為時間的問題，我們今年的方案就改為雙攻，因為他們之間的程式大部分相同。

而另一部機器人只要略改些少便能以一條程式供兩部機器人去使用。而我的程式也跟我的隊友很像，因為是我們大家一起去取長補短，互相改進，例如我們的復眼算法是一樣的，出界判區，復位動作，開盤程式，動作算法，超聲突變算法等也是相似的。當然，也有一些部分是各不相同的，追球的算法，一些特別的動作也要改變，因為我們這次是雙攻的打法。

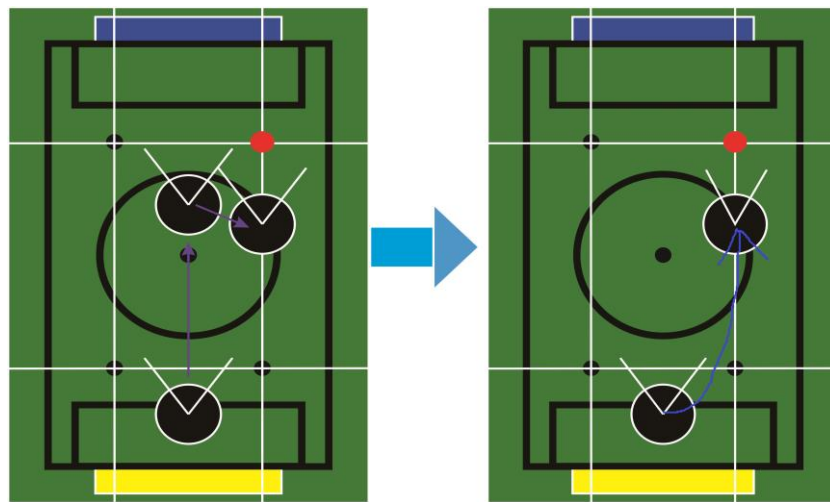
我們的打法首先要變的是一前一後的復位。我們的復位分區編號是一樣的，但復位的地方不一樣，因為我們雙攻的打法是一部是全進攻，即 A 機器人作為主要的攻擊機。而另一部機器人即 B 機是偏防守的機器人，而我這次寫的程式便是 B 機器人，即偏向防守的進攻機。我要改的第一個部分就是復位，因為我們不能兩部機器人都在 0 區上復位，會導致兩部機器人不停地碰撞。加上我是偏防守的，因此我的機器人復位要改成 5 區，其好處是不會與 A 機器人再一開始便一起搶球，而導致阻礙而錯失進攻機會。



而我認為 B 機器人復在禁區能有防守員優先，也能在後排阻擋球過來和能提早去做迎球的動作。而現在對防守機器人的達標要求很嚴格，因此雙攻的戰術就能消除這灰色地帶。

而第二樣要改的東西是兜球，我們的兜球是由一個算法去做的，簡單來說就是先把復眼變成 360 度，再根據球的遠近由直線變成切線去追球，之後我們會設定向前行走的角度範圍。但如果直接這樣使用的話，是會有缺陷的。所以我的兜球有加一個處理的程序，而這個程序以前的一位師兄教我的，這部分程序就是「縮頭」。

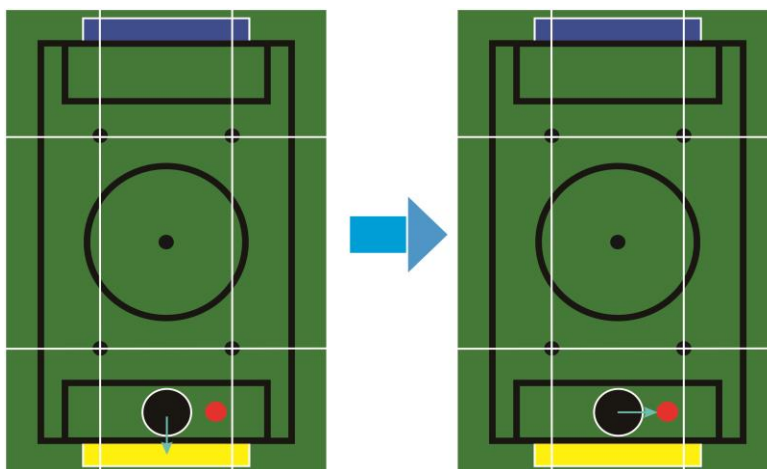
在未做處理（縮頭）之前，機器人一看到球，就會去兜球，然後就會因為慣性，導致兜過頭，做出多餘的動作，浪費了時間，也會給別人去拿球或擋球的時間，對我們是非常不利的。而做了縮頭的處理之後，當機器人愈來愈接近球時，它就會慢慢地把原本看到球向前走的角度縮少，然後它的兜球軌跡就會變成一個弧線，這樣就能比較準確的去追球，而程式裏面就是一條利用正比例函數的算法，即利用  $Y=KX+B$  去做的。



而第三樣要改的就是機器人在後場看到球的動作：因為我是要做一部有防守能力的攻擊機器人，因此我的兜球不能太過進取，如果我的機器人太過進取的話就會發生搶球或阻礙前面攻擊機去做一些進攻動作，所以我就會做一些有防守性的動作。而在前半場的球我就不會去追它，而把它留給前面的進攻機械人。

之後我也改了機器人在禁區內看到在左右兩邊的球的動作。未改之前，攻擊機器人看到在左右方的球，會向後走以兜球，但如果在禁區還是向後走的話，便會卡往龍門而清不

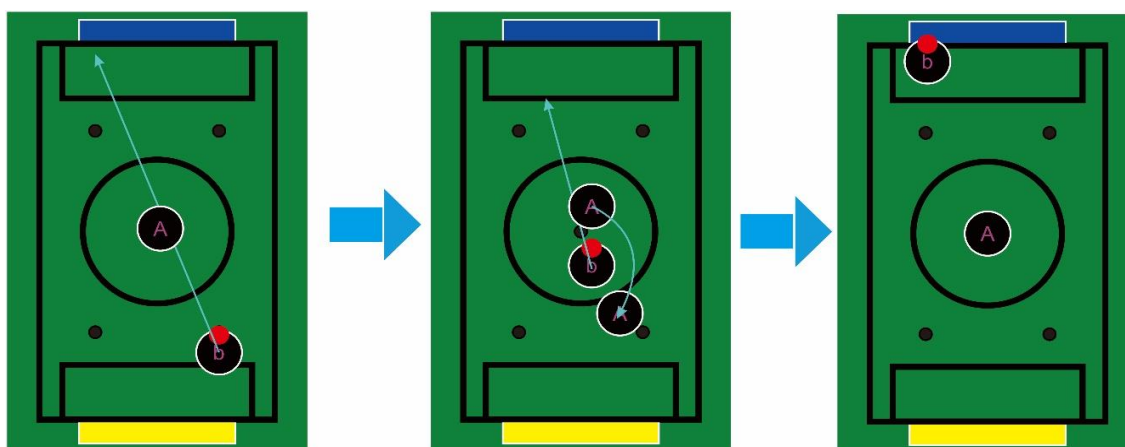
了球，而被人進球。因此我便改了後場不兜球，而是改成左右清球的形式，這樣便能在後場時有一定的防守性。



而之後我的機械人是雙盤的結構，所以當我的機器人拿到球後會做一些進攻動作。

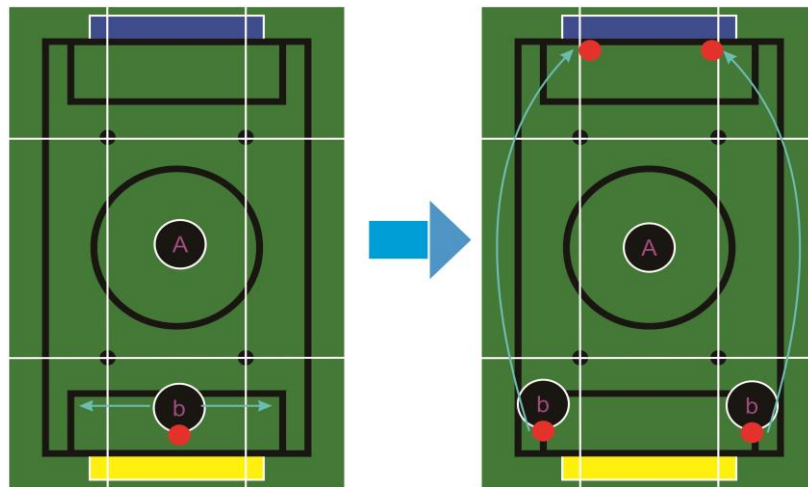
## ● 前盤

我前盤拿到球後會做斜行的動作。機器人在後場拿到球後，就會斜行，之後會發現前面有我方的 A 機器人。但其實 A 機器人是想要做一個兜球的動作，因此 A 機器人就會因兜球的動作而避開了正在斜行的 B 機器人，從而 B 機器人就會成功做到進攻的動作，而 A 機器人也會因為看不到球而進行復位回到中間。這就能做到一個攻防交替的動作，既不會阻礙另一部機器人的進攻動作，也能增加 B 機器人的進攻性。



## ● 後盤

而當後面的盤拿到球後就會到指定的區域甩球，把球傳去前場，假如運氣好的話就能進球，而這樣後面的盤球裝置便不會荒廢。至於我選擇這樣去甩球是因為這樣不會讓別人輕易地拿到球，因為界外一定不會有機器人的存在，還可以借助斜台去令球返回場內，這樣又增加了進攻性。

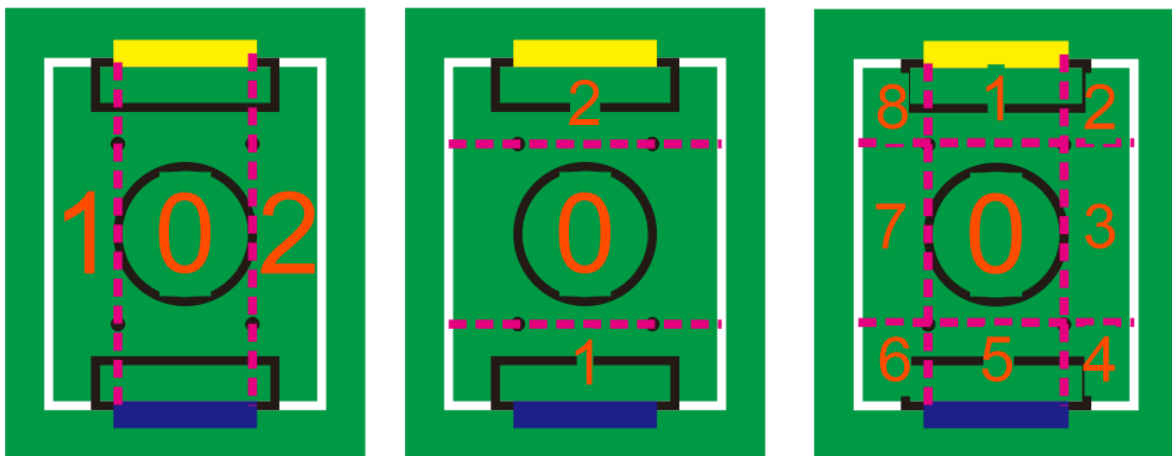


## 簡水源同學

我的程式是以上一年重量組的程式為藍本寫的。由於上年比賽時的超聲分區和進攻方法良好，所以我們繼續使用這些進攻方法和寫法。

### ■ 超聲分區

首先，我們利用左、右超聲將比賽的場地分成左為 1、中為 0、右為 2 的三個小區，然後再用後超聲將比賽場地分成前為 2、中為 0、後為 1 的三個小區，最後再利用這兩種分區，將比賽場地分為九個大區，區號如圖所示。



因為我們如果右邊出界而左邊有機器人，我們就不會用左邊的超聲，因為用左邊的超聲就會被左邊的機器人影響，另一方法又相同，所以我們這種復位方法的準確率很高，經過多次實驗後，機器人幾乎沒有出界。

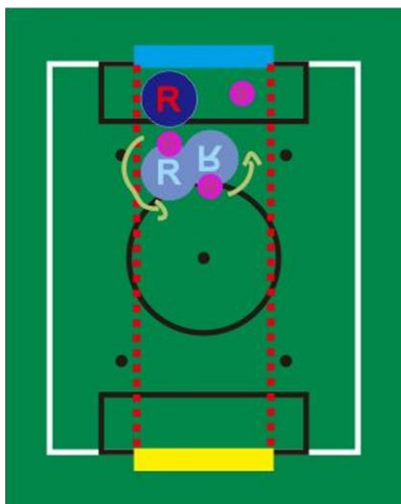
但是這一方法也有一種缺陷，就是如果超聲測距模組在比賽的過程中由於碰撞而發生故障，沒有讀數，那麼我們的分區就會錯誤，這一方法就不管用了。但是壞超聲的機會很少，所以我們用了這種寫法。

### ■ 前盤的進攻動作

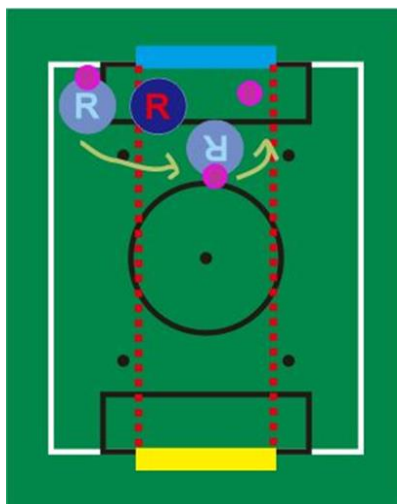
#### ● 舊前盤的進攻方式

舊前盤的進攻方式可分為兩種情況。一，當機器人在場地左右兩邊的區域裡盤到球時，先將球帶到底線，將對方機器人引誘到場地的左邊或右邊，然後一個大範圍地

轉身，繞過對方機器人，並在移動到另一邊的同時，把球甩出。二，當機器人在場地盤到球時，先向斜行靠近對方機器人，把對方的機器人引誘到偏左或偏右邊，然後一個迅捷地轉身，移動到另一邊的同時把球甩出。



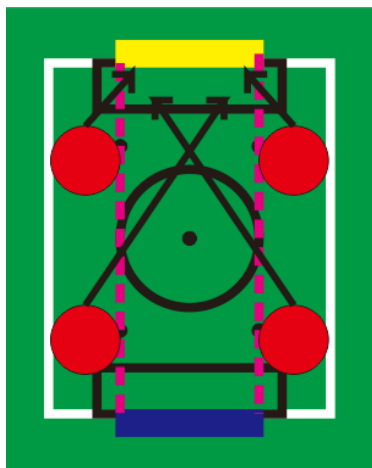
(圖 前盤的進攻方式一)



(圖 前盤的進攻方式二)

## ● 新前盤的進攻方式

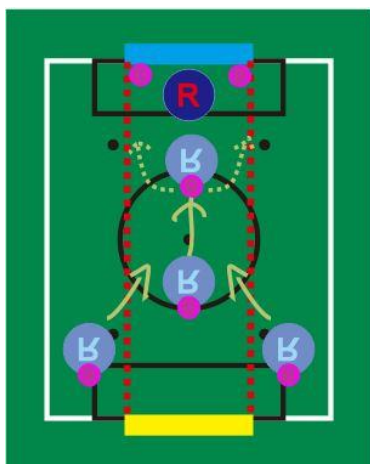
新前盤的進攻方式比舊前盤的進攻方式簡單得多，只是前盤盤到球後向左右兩邊的球門死角斜行進攻。這不像舊前盤的進攻方式很容易甩球出現烏龍球。我們還可以跟據自己的位置改變斜行的角度





## ■ 舊後盤的進攻動作

後盤的進攻方式很簡單，也很有效。但由於我們的策略是雙攻，所以我們用後盤有很大的風險。



## 本次工作总结

- I. 各自寫了自己的程序，有追球呀，復位呀，分區呀以及策略戰術等部分。盡量把程序寫到最好，增加比賽優勢。
- II. 暫定 1 月 17 日為組內賽

日志编号：08

活动日期：2017 年 1 月 17 日 - 2017 年 2 月 12 日

参与队员：何建群 戴智龍 簡水源 林震    活动地点：學校、北京

## 讨论过程

日期：1 月 17 日

出席人員：何建群、戴智龍、簡水源、林震

內容：

簡：終於比完了，沒想到最後居然能贏。

戴：唉，薑還是老的辣！

林：不過感覺我們兩隊都還有很大的改善空間。

何：你說的沒錯，這次小小的組內賽，足以看到我們很多的弊端。

林：額… 最多弊端的非我莫屬啦，我的機器人不停出界，留在場上的時間很少，失去了攻勢啊。

戴：對，你怎麼出那麼多界啊？

林：額，這是個好問題。我感覺應該不是程序的問題，再說了，我這塊灰度板從去年用到現在了，也該重新焊接一塊。

戴：嗯嗯。

何：那我們呢，你覺得我們有甚麼問題啊？

戴：嗯…可能就是有的時候追球不太準吧。

簡：這個我也看出來了。

林：不過…戴智龍你也是啊。你的球兜得也不準。我離了場之後，很多時候你都比他們慢拿到球。

戴：嘿嘿，這個我會再改進的。

簡：那你們有沒有覺得我們的攻勢太弱啊？

林：不弱，不過也不夠強。你們甩球之後成功進球的機率還得再提高。不過我的進球機率更低……

戴：不要緊不要緊，這次比賽讓我們知道了我們的不足之處啊，這樣我們就能越做越好了。

林：嗯，對了，你甚麼時候把控球的程序寫好啊？

戴：這個嘛，等我把追球部分改好就寫吧。

林：嗯。

簡：這樣討論一下也不錯耶，讓我們互相知道彼此的問題。

何：對啊，所以有時候，討論是必須的。

## 开发及调试过程

程序初步完成後，也到了我們組內賽的日子。最後何建群-簡水源隊獲勝。這次組內賽不那麼正式，也只是初步地測試一下我們兩隊的機器人。因此，賽後我們每個人都對這次比賽做了一個小總結，找出當中的問題並解決。

下面是我們每個人的賽後總結：

### 簡水源

我們在打完組內賽後，我就開始找我們的不足之處，首先，我們的機械人是新整的，有些地方可能有點問題。第二，我們的寫程式時間有點小，因為當我們整好機械人後，過一個星期就要進行組內賽，因此程式是用上年的程式改的，但很多數據還沒有改善，例如進攻動作，追球，超聲分區等等。導致我們在組內賽時分區不準和追球不準，使我們錯失很多進攻機會和失分。

所以我們會將把我們不足之處進行改善，設計出一部新的機械人，並改善我的程式。

### 何建群

在我們打完組內之後我發現我的程式有一些缺陷，而我的程式是上一年的程式，因為打組內賽的時間太過於倉拙了，而我把上一年的程式下載到我的新機械時，很多的數據都不適用於我的新機械中，而導致了有一些出界的情況頗嚴重和兜球會兜過龍，而我也開始進行必要的修改，如改了超聲的數據和一些兜球的算法，但在我改進我的程式時，我的隊友跟我說改機的問題，而我便停止去繼續修改我的程式，也開始和我的隊友一起改進我們的機械人，我也打算把程式放在整完新的機械人再把它改進和完善。

### 林震

這次組內賽，我們輸了。我認為，輸的原因可能有兩個。第一，我們沒有過寫盤球裝置的經驗，因此控球控得不太好，控到球以後做的動作也不好。有很多次，我的機器人正準備斜行進攻，但每次不是半路被對方機器人截停了，就是卡在龍門旁門，沒有一次是成功把球帶進龍門的。第二，由於我是直接拿去年的灰度板來用，所以效果不太好，五分鐘裏面我出了四五次界，大大損失了進攻機會。加上戴智龍不夠時間寫控球程序，因此，可以說我是我們隊的主力攻擊機。一場比賽中我離場那麼多次，輸掉比賽也不是沒有道理的。

另外，在比賽中，我的機器人也出現了一個問題，就是之前我加在底板用來防止犯規（控球深度）的那個馬達夾，從底板中脫落出來，其原因並不是螺絲螺母鬆了，而是固定馬達夾的那個地方斷開了。



因此，之後我打算把控球斜行的程序完善好，提高其進球的機率，並焊接一塊新的灰度板，也修改底板的設計。

## 戴智龍

我就是第一次以重量組的身份去參加我們的組內賽，這次的組內賽就讓我對重量組的比賽有了新的認識，我就發現到重量組的比賽不只只是追求速度，更多的是會追求程序上的穩定或還有變化，在重量組的比賽當中機械人追球之不單單只是把球直接推進對手的球門，而是可能會使用到盤球裝置去把球盤住然後把球甩去對手的球門或是繞開對手的機械人再進攻，有的隊伍甚至會再加上射球裝置，在盤着球的時候突然把球射過來，殺對手一個措手不及，這就我所說的變化；與此同時，隨着別人或自己的進攻方式變得更複雜，就更要講究自己程序的穩定性，自己穩定了才可以擋下對手變幻莫測的進攻，而如果是我要以不同的方式去進攻的話就更需要程序的穩定了，因為如果我連最基本的最主要的程序都寫不好的話那我又怎麼去讓我的程序變化呢？

在這次的組內賽的過程中，我對自己的機械人的表現都作出了一些總結，首先，我認為我的機械人的進攻能力有提升的空間，雖然我的機械人上是有盤球裝置的，但因為我還沒有完成任何關於盤球的程序，因此我就不使用盤球裝置，但和我的對手相比之下，不使用盤球裝置進攻能力真的差很多，所以我想在進攻方面的程序上還可以再加強，雖然我為了不阻礙隊友的進攻已經把覆位的位置調後了，但還是會阻礙到隊友，所以我還是要在這方面作出改善。但其實我的機械人的表現也不完全是差的，機械人的出界的次數比我預計的少，雖然還是會出界，所以我接下來也要去盡量去改善覆位的程序。

在組內賽過後，我們放了寒假。放完寒假的一兩天，我們隊的高三同學 何建群就前往了北京，去參加「RoboCup 青少年世界杯（RCJ 中國）優秀師生培訓」。下面將是何建群同學在北京的學習情況：

我在二月五日至二月九日這五天參與了在北京舉辦的 RoboCup 青少年世界杯（RCJ 中國）優秀師生培訓學習之旅，而我一開始是拒絕的，因為只有我自己去這個培訓之旅，感覺會很無聊。不過因為我是高三的學生，且已成功保送，老師不想影響他們的學業，因此就找了我去。最後，以大局為重，我必須出席這個培訓。

## ■ 二月五日

在我微信的朋友圈中發現了我之前在合肥世界賽所認識的一個深圳實驗中學的朋友也有參加這次培訓之旅，瞬間感覺好多了，至少還有認識的人。之後呢，我從澳門出發去北京，而這次的培訓地點是清華大學，清華大學可能是我這一生都不會踏足到的地方，因此有機會去清華大學真的令我感到些許興奮和期待。

而到了賓館之後，第一個悲訊就到我的耳邊來了，就是第一天的晚餐要自理，只有老師才有飯食，而我就要去賓館的飯堂吃飯。我一開始還以為國內的東西應該是比較便宜的，但是在我打開菜單的時候令我驚呆了，我看到有各式各樣的飯菜，且價格十分昂貴，一碟的飯餸最便宜也要三十元的人民幣以上。

在我吃完飯後我便馬上回房間去鑽研一下出發前老師給我的軟件，而在軟件裏面中也有一些注釋。

## ■ 二月六日

早上：

這一天我認為是比較沉悶的一天，尤其是早上。不過在早上演講的內容中，我比較有興趣的是 2016 比賽組織工作總結、2017 比賽組織工作規劃，這些內容是由李實博士來演講的。我在這裏真正的了解到甚麼才算合格的日志，也了解到一些 2017 年的改革和工作規劃。

之後韓老師也演講了機械人的一些歷史和設計時候要重視的一些東西，這一部分我也獲益良多，也令我更加明白設計前要仔細地考慮多方面因素。而我以前設計一部機器人是我的想像再加上畫圖軟件的輔助去畫的，並沒有深入去研究我的機器人的一些運動參數，例如功率或者一些機械人的加速度之類。而也沒有去測試機器人的結構，只是知道要



# 开发日志

盡量降低重心，和在比賽知道的一些位置不能這樣設計的經驗累積，並沒有一些實際測量和數據支持。



下午：

而下午也是我最有興趣的課之一，3D 列印與電腦建模。首先是 3D 列印，我對 3D 列印這一個名詞並不陌生，因為在我們學校也有 3D 的列印機，但是買了不夠一個月就壞了。而在 3D 列印的課上，我知道了 3D 列印在各行各業都有使用，但我想不到的是連小學也開始普及起來，而在臺上介紹的那個 3D 列印機也是非常的方便，而且它的功能強大，令我也想買一部那樣的 3D 列印機放在我們學校的機械人室。



# 开发日志

之後的計算機建模的課我也不知道自己在學甚麼和學了甚麼，我其實也非常專心去聽課，但無奈阻止不了周公來找我。

晚上：

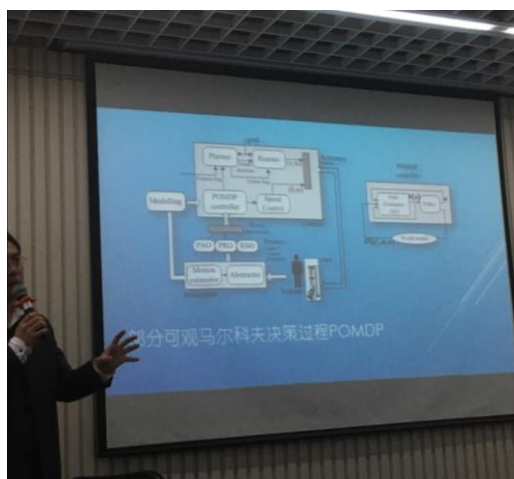
我在拿到零件之後便組裝到機器人上，也去測試了一下零件，我還發現那個控制器裏面有個小遊戲，通關條件感覺像是我們寫任務賽一樣，要繞過所有的障礙物才能通關。經過多次的闖關，以及對其操作系統的好奇，才發現原來這個遊戲是利用了陀螺儀去做的。

■ 二月七日

這一天我最期待的便是新的控制器的硬件部分和軟件的一些介紹和教學和一些其他的部件，因為我聽說可能在未來的幾年 demo 都要用這一套新型產品來和大家一起競爭，而這一塊經過我昨夜的研究好像還有很多新的東西。

早上：

早上的第一位講師講的東西讓我感到太無聊了，我很多東西都不懂，像是甚麼馬爾科夫決策過程，甚麼模糊數學啊等等的，我聽都沒聽過。尤其是模糊數學，我一開始還是挺感興趣的，因為我聽到是由一位伊朗的數學家去定義的，而他還解決了日本新幹線的控制問題，而令我最感興趣的是他在一個偏僻的國家出生，又能創造出一個斬新的理論，真的是很不簡單。而我也在網上查了一些關於模糊數學的相關知識，但我還是不理解，不過我相信我以後還是有機會去接觸它並且應用它。

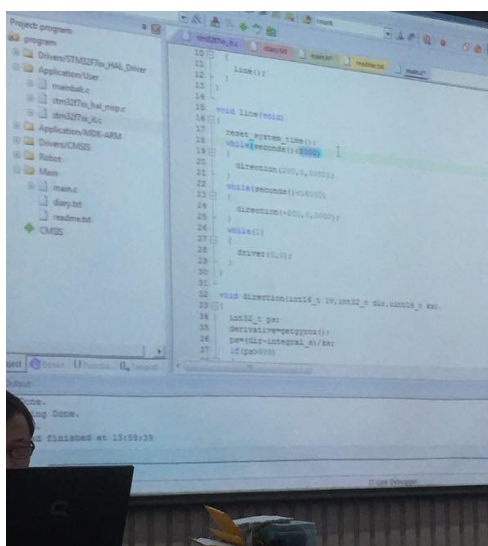


# 开发日志

在早上也學到了一些關於視像的東西，也看到了大學 robocup 的中型組是如何利用視像去找球並且能夠精準地去判別顏色。視像也是我們足球之後的改革，所以我很努力地去聽課，希望能學到一些視像方面的知識，不過，這些知識對我來說還是太難了。

下午：

終於進入了我所期待的課程，就是學習新的產品，而在新產品裏我覺得最厲害的便是陀螺儀，它能測量每一個方向的加速度，而黎老師也教了很多基本的程式和介紹了一些的硬件資訊的內容，最後也派了新的灰度，但我並不覺得有甚麼特別。



晚上：

而在晚上我便自己去研究一下他的軟件，那時我才發現了我的灰度板壞了，我連忙地去做一些急救，但補救不了。而我本來沒有預留位置來擺放灰度，因此我馬上拆了我的機器人，拿了一些膠紙去做一些暫時的固定，幸好我的底板有很多的空洞。

■ 2月8日

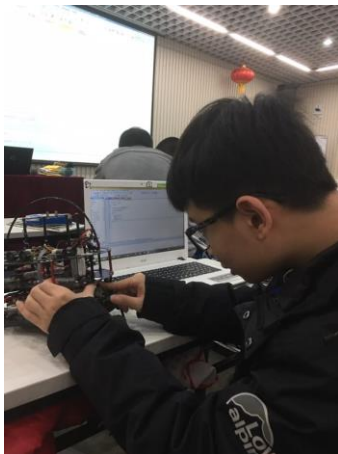
早上：

繼續教新的產品，而在早上我還見識到了陀螺儀的作用，而黎老師做了一些記錄機械人位移和再把它還原出來，這令我感到非常的厲害，還給我們看了一些用陀螺儀去做的動作，和他自己的閉環馬達。

# 开发日志

下午：

一場激烈的 demo 競賽，我用盡這幾天所學到的東西去完成它，但我並沒有全部完成。儘管只有我自己一個人，我也歇盡所能，希望能做到最好，因為我是一個怕輸的人，所以我想贏，以致我太努力去做，被偷拍也不知道。不過最後我也得到了一個好成績。



晚上：

外出食飯，導師請我與在北京大學就讀的師姐吃飯慶祝。

以上是我這幾天裏面的學習過程。下面我將說明一下我在 2 月 8 日所寫的任務題。

## ■ 第一題

題目：按不同的按鈕在主控器的 LED 燈上就顯示不同的顏色，如按下按鈕一就顯示紅色，按下按鈕二就顯示綠色，如此類推。

而我的寫法就是開三個新的變量，每個按鈕就放一個變量去讀它的返回數，然後再判斷按了哪個按鈕就顯示相對應的燈。

我一開始碰到的困難並不是沒有想法，而是不知道哪個代碼是控制主控器上的 LED 燈，所以我就去 readme.txt 裏面找，找到之後我也就順利地完成了第一條 demo。

```
int main()
{
    RobotInit();
```

```
LCD_Clear(BACK_COLOR);
LCD_POINT(BLACK);
while (1)
{
    a=GetButton1();
    b=GetButton2();
    c=GetButton3();
    if (a==1)
        Led_R(1);
    if (b==1)
        Led_G(1);
    if (c==1)
        Led_B(1);
    if (a==0&&b==0&&c==0)
    {
        Led_R(0);
        Led_G(0);
        Led_B(0);
    }
    LCD_XY(10,10);
    printf("a=%3d",a);
}
}
```

## ■ 第二題

題目：我們要按要求去設置一個密碼，就是要按順序按對了才會開始我們的機器人。而老師讓我設置的密碼是 213，就是先按二按鈕，再按一按鈕，然後再按三按鈕之後才能夠執行之後的程序。密碼對了之後就要讓機器人面對指定的龍門，之後再自轉兩圈半，然後再回到場的中心（不能使用超聲）。然後最後還有一個子任務是關於球的，但我忘了具體內容。

而一開始的密碼部分令我想了很久，我一開始並沒有特別的想法去寫。經過思考，我決定按順序去判斷第一個按鈕有沒有按對，而如果有按對的話就再去判斷第二個，如果沒有的話就讓它死在第一條的循環裏面，不讓它跳出去，而程序也不會繼續做下一部分的任務，而程序裏的寫法：

```
while (1)
{
    a=GetButton1();
    b=GetButton2();
    c=GetButton3();
    if (b==1&&d==0)
    {
        while(1)
        {
            d=1;
            b=GetButton2();
            if (b==0)
                break;
        }
    }
    if (a==1)
        d=2;
```

```
        if (c==1)
            d=3;
            if (d==1)
                break;
    }
    d=0;

    Led_B(1);
```

這是判斷第一個按鈕是否是按對的程式碼，也是一個一個去判，而 d 的作用是用來判斷有沒有去按過其它錯的按鈕和有沒有按過按鈕，而 d=0 的時候就是沒有按過任何按鈕，而如果按了其它按鈕的話我就會讓 d 這個變量賦不同的值，如按了一按鈕的話 d=2，而它就不會再理會之後是否按二按鈕，因為 b==1&&d==0，而一開始按了其它按鈕就不會進這個判斷。而我有加一段是用來防止長按按鈕，因為你按一下按鈕程序就會走很多次，而我有寫一段防止這個情況的，現在就變成按下去到放手才算完成一次的按鈕動作，而到最後，為了方便測試，我寫了如果按對按鈕的話，它就會亮燈，以代表它有跳出循環。之後另外兩個密碼判斷也是用這個方法。

之後轉到指定龍門：

這個簡單，也就是用指南指轉回必要的角度。

到自轉兩圈半：

我這個也是用指南針來寫的，而我還想到用一個更簡單的方法就是計時，來走動作。

之後就是回場地中心：

而我覺得這是整條 demo 裏最難的一個部分，因為不能夠使用超聲。我用的方法就是先到場上的一個角落，之後再用計時的方法來走回中心。我的想法就是定點到定點，這樣就能夠用死動作去走。而找角落的方法當然是用灰度來找，在一開始先找左右隨便一邊的白線，找到了之後再向前或後去測底白線或定白線，這樣就能定了一個角落。

而最後的球的子任務因為我沒時間寫所以我也沒有完成它。



## 本次工作总结

- I. 組內賽後做了總結，找出當中的問題並解決。
- II. 「RoboCup 青少年世界杯（RCJ 中國）優秀師生培訓」
  - 了解到合格日志的標準
  - 了解到一些 2017 年的改革和工作規劃
  - 明白設計前要仔細地考慮多方面因素
  - 知道了 3D 列印技術的普及化且其方便之處
  - 上了計算機建模的課
  - 發現利用陀螺儀來寫的遊戲
  - 看到了大學 robocup 的中型組是如何利用視像去找球並且能夠精准地去判別顏色
  - 研究其軟件，換灰度
  - 見識到了陀螺儀的作用，看到如何記錄機器人的位移及把它還原出來
  - 任務賽

日志编号：09

活动日期：2017 年 2 月 15 日 - 2017 年 3 月 7 日

参与队员：何建群 戴智龍 簡水源 林震    活动地点：學校

## 讨论过程

由於距離比賽只剩下兩個月不夠，而何建群已完成了培訓，從北京回來了。經過幾天的休息，我們展開了對中國賽的準備工作。

日期：2 月 15 日

出席人員：何建群、戴智龍、簡水源、林震

內容：

何：終於回來了，我們都多久沒見了。

簡：對啊，好久沒見了。

戴：你這次培訓學到了甚麼東西啦~~

何：學的也挺多的。對那塊控制板也上手了。

林：那就好。那你還會像往常一樣練習做任務賽嗎？

何：會啊，而且這次我打算用那塊新的控制板來寫。

林：咦，這樣不錯啊。我也有打算練習做任務題，不過不是用那塊板，因為我不會用。

何：嗯，對了，簡水源你也要練習一下。你的經驗多，基礎好。

簡：好吧。

戴：那…我呢？

何：剛好，我有工作可以交給你做。

戴：甚麼？

何：這次我去北京，聽了韓老師講的話，發現其實在設計時我們還有很多東西可以注意。而聽了這次的課，我也想測一下我機器人的加速度，你可以幫我這個忙吧？

戴：嗯，沒問題。

林：啊，對了，你要順便找一找一些硬件的相關理論嗎，要是筆試考到了而我們不會就不好了。

戴：這個也沒問題啊，包在我身上了。

這次討論，我們主要分配了對中國賽的準備工作。

何建群：用新控制板寫任務

戴智龍：測加速度、搜尋一些資料

簡水源：用舊控制器寫任務

林震：用舊控制器寫任務

## 开发及调试过程

### 戴智龍

我們想測試一下我們機器人的加速度，然後我們又想到我們學校有一個先進的物理實驗室，所以我就想利用這個物理實驗室來測一下機器人的加速度。我們會先用理想的條件來計算，然後再用我們實驗室的儀器來驗算。



我們先用理論來計算了我們的四輪機械人和平行四輪機械人的加速度會否因為馬達位置的的不同而會不一樣。我們要計算加速度的公式是  $a = \frac{F}{m}$ ，但我們只知道  $m=2.4\text{kg}$ ，我們還需要求到  $F$  的值才可以求到加速度  $a$ ，所以我們就要先求  $F$ 。

#### ■ 90 度夾角四輪

求  $F$  的公式為  $M=FL$ ，我們參考了馬達上印的數據已知  $M=3.3(\text{KG}*\text{CM})$ ， $L=0.11\text{m}$

$$M=3.3(\text{KG}*\text{CM})*9.8*0.01=0.323\text{Nm}$$

$$F=\frac{M}{L}$$

$$F=\frac{0.323}{0.11}$$

$$F=2.94\text{N}$$

再根據勾股定理我們求得  $F$  合為  $\cos 45 = \frac{F_{\text{合}}}{2.94}$

$$F_{\text{合}}=2.08\text{N}$$

方法錯了

四個輪子的合力是  $F_{\text{合}} = 2.08\text{N} \times 4 = 8.32\text{N}$

我們機械人的  $F_{\text{合}} = 8.32\text{N}$ ，質量  $m = 2.4\text{kg}$

$$a = \frac{F_{\text{合}}}{m}$$

$$a = \frac{8.32}{2.4}$$

$$a = 3.47\text{m/s}^2$$

所以四輪機械人的加速度為  $3.47\text{m/s}^2$

方法錯了

## ■ 平行四輪

求到  $F$  的公式為  $M = FL$ ，我們參考了馬達上印的數據已知  $M = 3.3(\text{KG} \cdot \text{CM})$ ， $L = 0.11\text{m}$

$$M = 3.3(\text{KG} \cdot \text{CM}) \times 9.8 \times 0.01 = 0.323\text{Nm}$$

$$F = \frac{M}{L}$$

$$F = \frac{0.323}{0.11}$$

$$F = 2.94\text{N}$$

由於是平行四輪，所以我們的馬達是直接向前的， $F_{\text{合}} = F$

平行四輪的合力是  $F_{\text{合}} = 2.94\text{N} \times 2 = 5.88\text{N}$

我們機械人的  $F_{\text{合}} = 5.88\text{N}$ ，質量  $m = 2.4\text{kg}$

$$a = \frac{F_{\text{合}}}{m}$$

$$a = \frac{5.88}{2.4}$$

$$a = 2.45\text{m/s}^2$$

所以平行四輪機械人的加速度為  $2.45\text{m/s}^2$

方法錯了

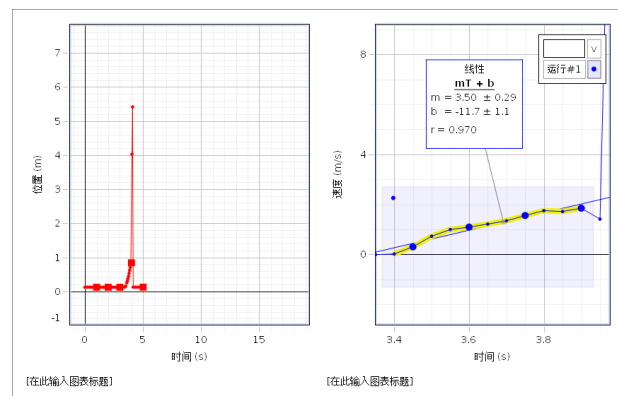
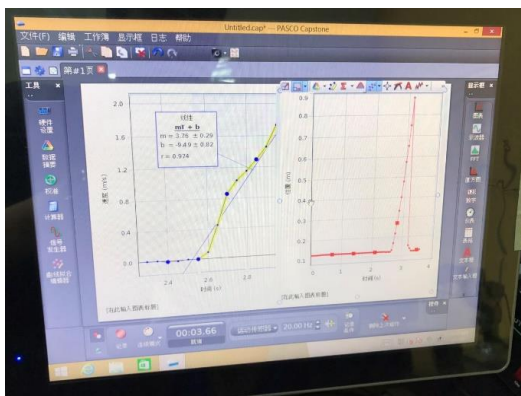
方法錯了

# 开发日志

我們用理論計算得出了四輪機械人的加速度為  $3.47\text{m/s}^2$ ，而平行四輪機械人的加速度為  $2.45\text{m/s}^2$ 。

現在我們就要借助我們實驗室的儀器來驗證我們的結果是否準確，我們是用了四輪機械人來做這次的實驗。實驗的結果顯示我們的結果是準確的，四輪機械人的  $a=3.47\text{m/s}^2$ ，所以這個擺法的優點是起步快，而缺點是最大的速度小。

平行四輪機械人的加速度為  $2.45\text{m/s}^2$ ，所以這個擺法的優點是機械人在後半段會超前，缺點是在起步會慢一些。所以我們最後還是選擇了速度比較平均和加速度較快的四輪擺法。

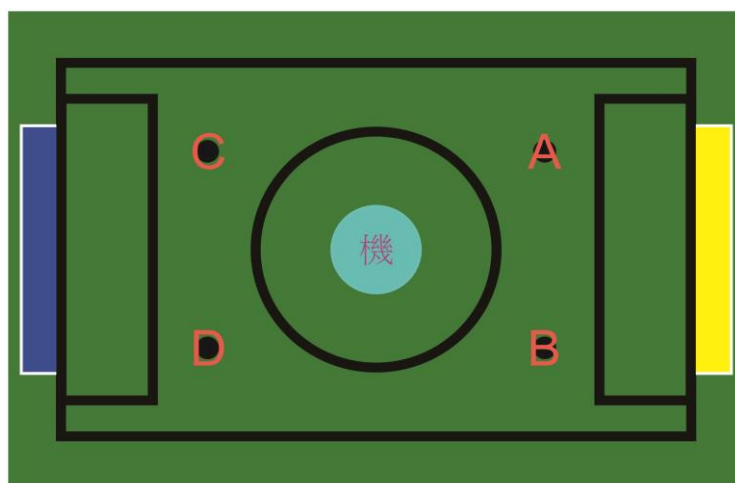


## 何建群

而我回到澳門後也有用這套產品安裝在機機人裏去做 demo，因為聽說會在今年全國賽的 demo 要用這套產品，而我為了熟悉這套東西我也用這套用來去做一些 demo，而我做了兩條的 demo。

### ■ 第一條

第一條的 demo 是關於球的，因為我在北京的學習之旅並沒有用過復眼去做一些的追求之類的動作，而我的隊友們就憑記憶推薦了一條以前的 demo，而他們是用文字去表達的，而我就畫了一幅圖去更清晰出表達題目的意思：



而文字表述就是會有兩個球，機械人放中間，而一球放 A 或 B 點，而另一球放在 C 或 D 點，由抽籤決定。而把沒放球的點都放上障礙物，而機械人必須把面向它的球先推進背向的龍門，而過程中不能觸碰另一球，直到機器人把首球推了進去，之後才可以把另一球推進另一龍門，最後機械人要回到中間。

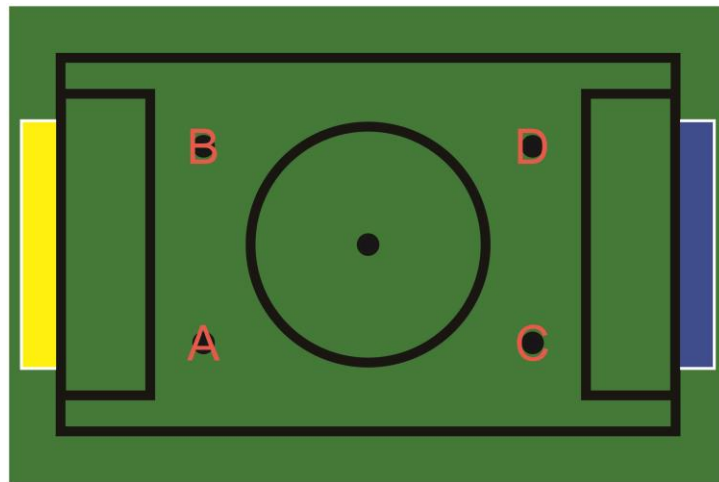
而我也認為這個題目能測試一下新復眼，我做這個題目的思路是：一開始先向前走直到禁區門口，而這樣就能去接近我第一個要去接觸的球，而之後我就用指南針去調整一下我的機械所要進攻的龍門，而之後我就寫了一條簡單的追求，而這個追求我有加一些特別的動作，例如我會用左右超聲去把球運去中場，而先要條件當然是球在我的機械人的控球位中，而當左超聲大於右超聲的話，我就會令機械人向左上斜走，而到中場後就不再用左右超聲調整，因為到前場之後會有障礙物去影響我的超聲，令我的機子向另一球的方向偏從而有機會撞到另一個球，而我會用前超聲去判斷我的球有沒運入龍門，而運完之後，我



會再用指南針去再調整去另一龍門的方向，而後再重覆一次剛才的兜球動作，而最後兩個球都進了球門之後，我會再用前後左右超聲去回到中間位置，而這個 demo 就成功去完成了，而我也認為這個復眼雖然有跳眼的最大通號，但不會太嚴重，而我的解決方法也是用物理方法去調整，而我也認為是個復眼整體上還是不錯的。

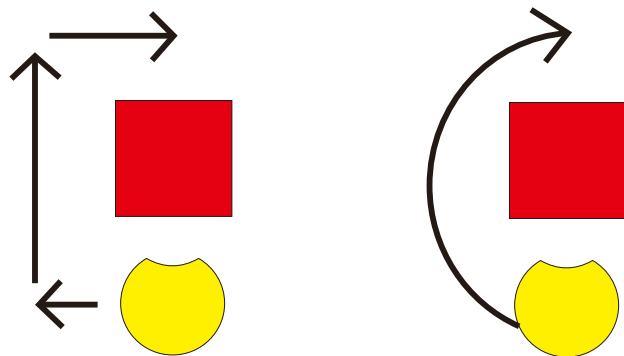
## ■ 第二條

而第二條 demo 是上一年憑記憶記下的 demo，而在上一年，因為這條 demo 有些難和時間不夠，所以我們沒有做：



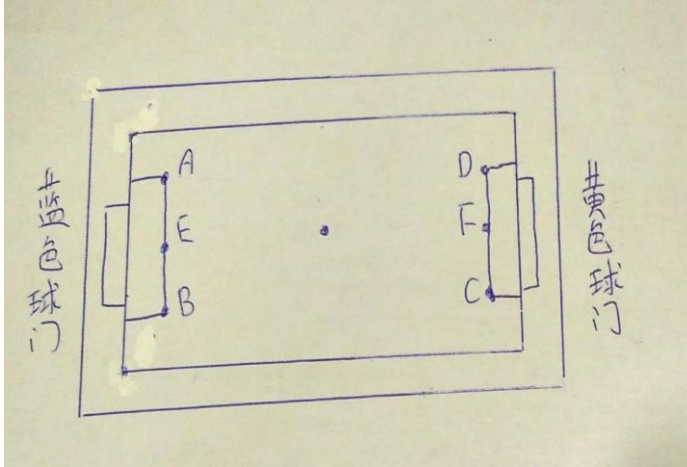
而題目就是 A、B、C、D 四個點，隨機放三個障礙物，而要按照順序出繞障礙物，而重點是碰到障礙物會被扣分，由藍色球門去發而最後要回到黃色球門。

雖然題目是挺簡潔，但是非常難的，而我的做法是：首先我會嘗試繞障礙物的速差，因為我選擇了用繞圓心的方法，而另一種方法是用走正方形的方法，但我認為走正方形的方法是很累贅的，因為這種寫法要分多段即向左一段，向前一段，向左又一段，但走圓就不同，因為它只是利用向左或向右的動作之後再因速度的不同，而它的軌跡便發生變化，即便成圓的軌跡，而這個速差也是非常之難找的，而我用了一段時間去試，而因為它的控制器並不能直接調節馬達速度，而中鳴的主控器就能做到，而這一點也是中鳴的主控器的一個優勢，而在找到速差之後，我就正



式去寫，而一開始我會令機械人調整角度，之後去黃色禁區然後再開始判 A 點有沒有障礙物之後判 B 點……，並沒有技巧，而我判障礙物，而我是利用前超聲有沒有變少去判點有沒有障礙物，而我認為超聲的擺放高很重要，因為障礙物是汽水瓶而汽水瓶上面是一個大面積之後中間是一個凹位，而當超聲測到凹位時，它返回的值回是 4095，即測量不了，所以我便要去調整，而我也用了一段時間去調整，而其他並沒有太難，只是煩，都是用超聲去每一個點去判而已。

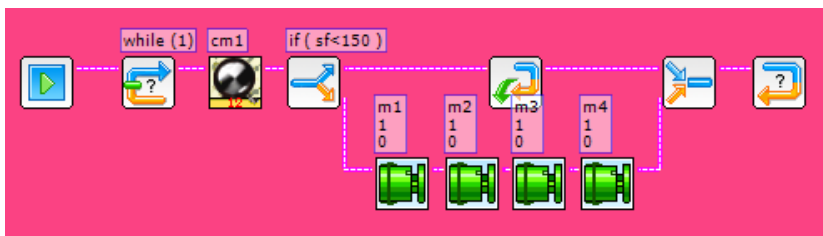
林震



任务要求：

- 前超声启动机器人。机器人放置于场中心点，机器人朝向随机；
- 场内两个球，一个发光，另一个不发光，A,E,B 抽籤放置一个发光球，D,F,C 抽籤放置一个不发光球；
- 场内也有四个障碍物，剩下的四个点就是放置障碍物的地方。
- 机器人要将发光球推入黄色球门之中，不发光球则推入蓝色球门中，机器人需要带球进入禁区内后才可以进行射门；机器人如果碰到球门和障碍物的话扣 2 分（重复触碰不重复扣分）若机器人出界或者射错球门的话则直接视为失败。任务必须在三分鐘内完成。

這題 DEMO 說難不難，但是也不簡單，只要想清楚方案就可以很順利的完成。我們首先寫一個迴圈，前超聲小於 150 就跳出這個迴圈，如果大於 150 就不斷迴圈這個迴圈。

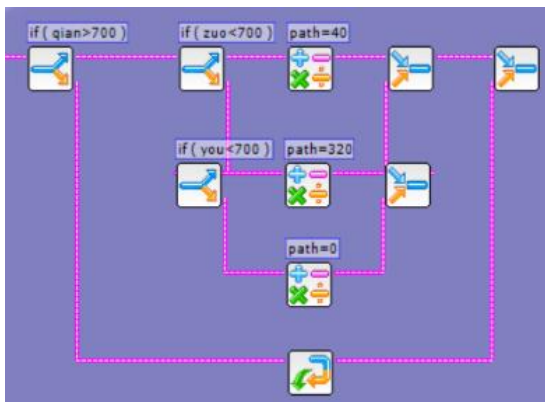


# 开发日志

接著我選擇先把發光的球推入黃色球門先，我們先用超聲寫一段程式讓機器人不碰到任何東西的情況下走到藍色龍門前且機器人持球位正對黃色龍門。

接下來我們要知道因為 2 升的可樂瓶的高是高過球的高，所以我們可以利用前超聲來判斷自己面前的究竟是障礙物還是球，但是我們要把前超聲裝在高於球的高的地方才行，這樣做避免讀到球，知道這些後我們就寫左超聲（當然也可以用右超聲）在讀到的數值是在 A 或 E 或 B 範圍的時候而且前超聲讀數還小於 500 的時候我們就可以知道這個點上面是球還是障礙物。

找到球在哪一個點後，就可以開始把球往前推了，因為出界直接視為失敗，所以這時候我們要限制球不會被機器人推到白色界外，怎麼寫呢？就是用左右超聲來判斷，如果左超聲小於 700 就向前斜右走，如果右超聲小於 700 就向前斜左走，如果前超聲小於 700 的時候那時候就是整個 DEMO 最難的地方，對面的 D,F,C 點上面是有障礙物和球的，我們怎麼樣避開球和障礙物並且在他們的中間穿過到達黃色龍門的禁區裡呢？這是最難寫的地方我們要用左超聲（當然也可以用右超聲）來進行判斷這一情況，我們先選擇一下想從哪兩個點中間經過，我選從 D,F 這兩個點中間穿過（當然從 F,C 中間穿過也行），當左超聲測到的數值等於 D,F 兩點的中間數值的時候就向前走；如果左超聲測到的數值不是等於 D,F 兩點的中間數值的時候，偏左了就向右走，偏右了就向左走。



然後如果前超聲小於 400 的時候，就不向前走先，先把自己的機器人調整到 D,F 點的中間後，就不再用左超聲來調整自己的位置，因為那時候機器人向前時左超聲會測到障礙物就會使左超聲的判斷出錯。經過障礙物後，就短暫加速一下，把球踢進龍門，然後超聲小於 120 就停下來。這樣子就搞定了一半的 demo 了。

接著機器人自轉把持球位對正藍色球門的那邊，然後就可以複製之前找發光球的程式，只要把裡面的一些動作改一下就搞定了。

在寫這條 demo 途中我們遇到了一些問題，就是指南針會偏！因為場地的磁場有點大使機器人從一個龍門到另外一個龍門的時候會越走越歪，機器人自身越來越歪的話會影響到超聲的讀數，到嚴重的時候有可能會變成 4095！這個情況很無語，磁場影響指南針的讀數是寫 DOME 時候最頭痛的問題，我先會在場的中心點先使指南針硬鐵補償一下，然後在場的中心點定指南針的正方向，這樣子會好一點。

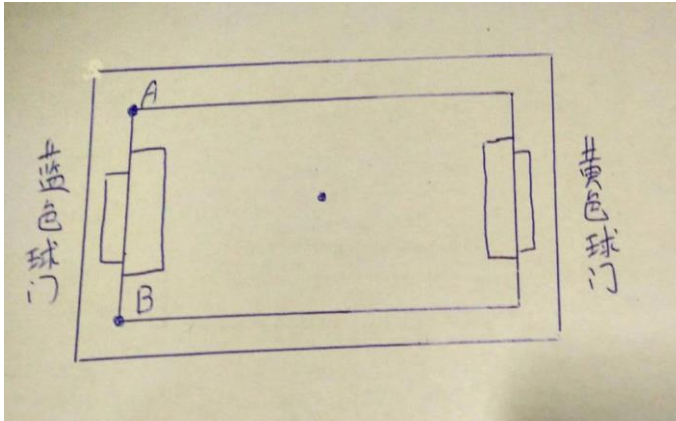
還有一個問題是控球的問題，會掉！！使用要開的很慢很慢，除此之外機器人如果有左右走的動作的話，那個左右走的速度要開的更加慢，因為左右走有可能將自己控的球給磕走這個也是一個很大的麻煩對於這條 demo。

還有最後一個問題就是機器人在龍門前時候左右超聲會讀到龍門，所以我會把左右超聲向前裝點（持球位方向），為什麼呢？因為機器人在禁區的時候正對著對面的龍門的時候才會用到左右超聲，所以那時候我們要儘量把左右超聲裝的遠離自己所在禁區的那個龍門。

我是用熱溶膠占時性的改動超聲等等硬體的位置，這樣做比較快捷方便，也挺實用的。

最後我有個小總結，就是如果改動硬體的位置能使 demo 更方便，準確的話，那麼我們改動硬體會好過改程式，比如自己的前超聲裝的高了，測不到龍門，我們就要把超聲裝低，這樣做會簡單過改程式。

## 簡水源



任務要求：

- 前超聲啟動機器人。機器人朝向隨機，放置于黃色龍門禁區內。
- A, B 點抽籤，抽到的那一個點放一個發光的紅外線球。
- 機器人需要將球帶入黃色龍門內，
- 機器人帶球進入禁區內後才可以進行射門
- 若機器人出界的話則直接視為失敗。
- 任務必須在三分鐘內完成。

因為這次我們用 c 語音來程式設計，機器人用的是 X2RCU，一開始我們要寫 `#include "HardwareInfo.c"` 和 `#include "JMLib.c"` 在程式的最前面，然後在 `int main(void){ }` 的 `{ }` 裡面要寫 `X2RCU_Init();` 在裡面的最前面（X 幾 RCU 根據自己使用的 rcu 來改數值）所以最後變成 `int main(void){ X2RCU_Init(); }` 這樣子。

定義好所有的埠和變數後，就寫一段前超聲啟動機器人的迴圈，如果前超聲的讀數小於 150 就跳出迴圈進行下一個迴圈，如果前超聲大於 100 的話就一直迴圈著這個迴圈，為什麼是前超聲讀數小於 100 那麼近呢？而不是小於 200，那你這時候要想想因為你的機器人是放在龍門前面的而且朝向是隨機的，如果你的前超聲朝向是向龍門的話，前超聲又小於 200 跳出迴圈的話，那麼你的機器人一放下去就會啟動了，就會違背題目，變成不是裁判用手遮啟動機器人了。

然後就開始寫 demo 的正題了，我們首先要寫一段馬達轉向的程式，到時候機器人要走什麼方向就會很方便，我寫了 6 個不同的動作。

```
81
82 if(path==9)
83 {e1=2;e2=2;e3=2;e4=2;}
84 else
85 {if(path==0)
86 {e1=0;e2=0;e3=2;e4=2;}
87 else
88 {
89 if(path==90)
90 {e1=2;e2=0;e3=0;e4=2;}
91 else
92 {
93 if(path==270)
94 {e1=0;e2=2;e3=2;e4=0;}
95 else
96 {
97 if(path==180)
98 {e1=2;e2=2;e3=0;e4=0;}
99 else
100 {e1=0;e2=0;e3=0;e4=0;}
101 }
102 }
103 }
104 }
105
106 SetMotor( _MOTOR_1_ ,e1 ,sd);
107 SetMotor( _MOTOR_2_ ,e2 ,sd);
108 SetMotor( _MOTOR_3_ ,e3 ,sd);
109 SetMotor( _MOTOR_4_ ,e4 ,sd);
```

(e1, e2, e3, e4 是代表馬達的轉向；sd 代表速度；path 代表機器人行走的方向)

這樣做後寫程式就會變的簡短很多，很方便，很實用，如果我機器人要停的話我就會寫 sd=0 就代表速度是 0 就不動了。

我們先要把機器人的正方向調好，正向著藍色龍門後就開始用複眼追球，當光值大於 130 時就停下來，判斷一下自己是在場地的左邊還是右邊，如果我們的機器人是在場地的左邊，接下來的追球向前動作裡面的速度就會變成左邊的兩個馬達速度是 25，然後右邊的兩個馬達速度是 10，這樣子我們追球向前時就會慢慢的拐彎，把球從那個角落扣出來，至於會不會出界和能不能順利的把球扣出來，那就要把握好向前時馬達的速差。

順利的把球扣出來之後機器人就要走回場的中間，避免機器人走出界外，要怎麼寫呢？就是寫如果左超聲讀數小於 700 或右超聲讀數小於 700 的話，那麼追球時如果是向前的話，那麼就把向前的動作它的速度改了，就是一組對角馬達的速度小一點，另外一組馬達的速度大一點，這樣做左就可以做到向前打斜走，就可以調整好機器人的位置。

最後如果前超聲的數值小於 220 的話向前的動作速度就要加快，把球踢進球門，如果前超聲小於 120 的話就停下來，就結束了這個 demo 了。

小總結：我寫的這個 demo 唯一一個比較難把握的地方就是向前動作的速差，但是在很多 demo 中利用速差來進行一些轉彎時是很方便的。



## 本次工作总结

I. 何建群：用新控制板寫任務

II. 戴智龍：測加速度、搜尋一些資料

四輪機械人的  $a=3.47\text{m/s}^2$ ，所以這個擺法的優點是起步快，而缺點是最大的速度小。

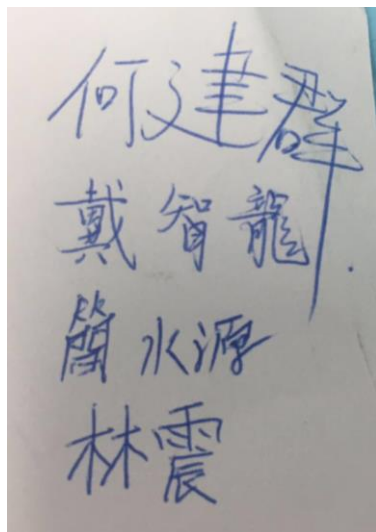
平行四輪機械人的加速度為  $2.45\text{m/s}^2$ ，所以這個擺法的優點是機械人在後半段會超前，缺點是在起步會慢一些。所以我們最後還是選擇了速度比較平均和加速度較快的四輪擺法。

III. 簡水源：用舊控制器寫任務（c 語言）

IV. 林震：用舊控制器寫任務（圖形化）

# 开发日志

所有队员签名：



日期：

2017/3/10