

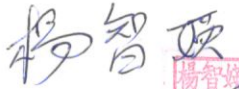
機電專題實作

CCU DOG 四足仿生機器狗外型及功能之設計以及實現

組員

408420001 王宇軒(貢獻度:40%)
408420013 莊哲維(貢獻度 10%)
408420097 邱宇志(貢獻度:50%)

指導教授

楊智嫻 
112.9.16.

專題製作背景

本次專題之目的為製作四足仿生機器狗(CCU DOG)

主要參考 Boston Dynamic 製作之機器狗，配合專題需求如下

- 1.起立: 單腳抬高高度 $>8\text{cm}$
- 2.蹲下: 蹲下高度變化 $>8\text{cm}$
- 3.行走: 直線行走 50cm
- 4.抬腳: 往前走一步的步驟
- 5.影像辨識數字: 利用影像辨識自動判讀數字

機器狗製作步驟與過程

1.設計概念與初步設想:

這次是要做四足機械狗，腳的部分要執行絕大多數的功能，所以我們主要先以腳的部位進行探討，以舵機為關節之發力點，並以壓克力板當肢幹，使用 L 型固定板使舵機與身體做完連接，對於 Arduino 施加特定程式指令，以達成專題所要求功能。

因此可分為以下兩部分進行:

(1.)硬體(外型設計與製作)

(2.)軟體(電路與程式設計)

2.外型設計:

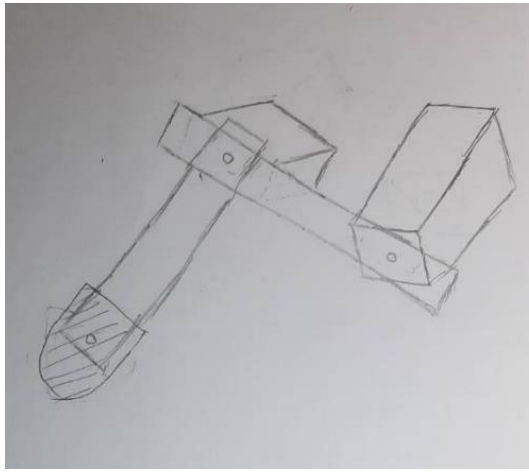
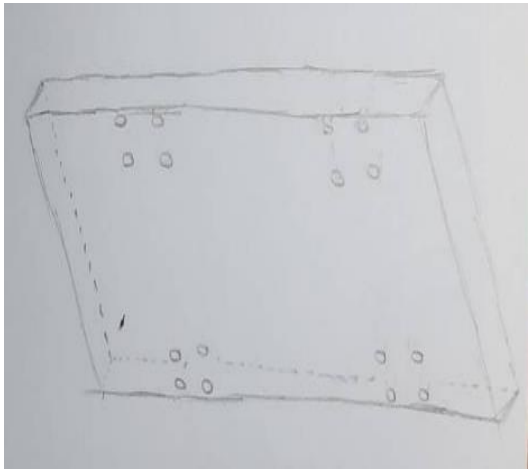
(1.)材料簡介:

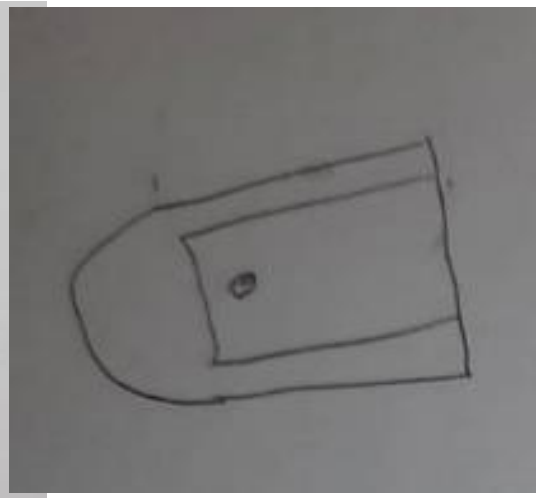
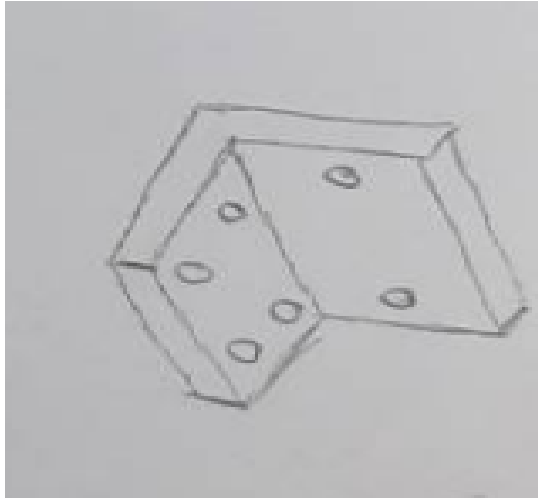
| | |
|------|------|
| 壓克力板 | 2 個 |
| 木板 | 0 個 |
| 舵機 | 8 顆 |
| 螺絲 | 48 顆 |

| | |
|----------------------|------|
| 螺帽 | 40 顆 |
| Arduino 板 | 1 個 |
| 麵包板 | 1 個 |
| 杜邦線(公、母) | N 條 |
| AI 視覺辨識模組(HuskyLens) | 1 組 |
| 七段顯示器 | 1 個 |

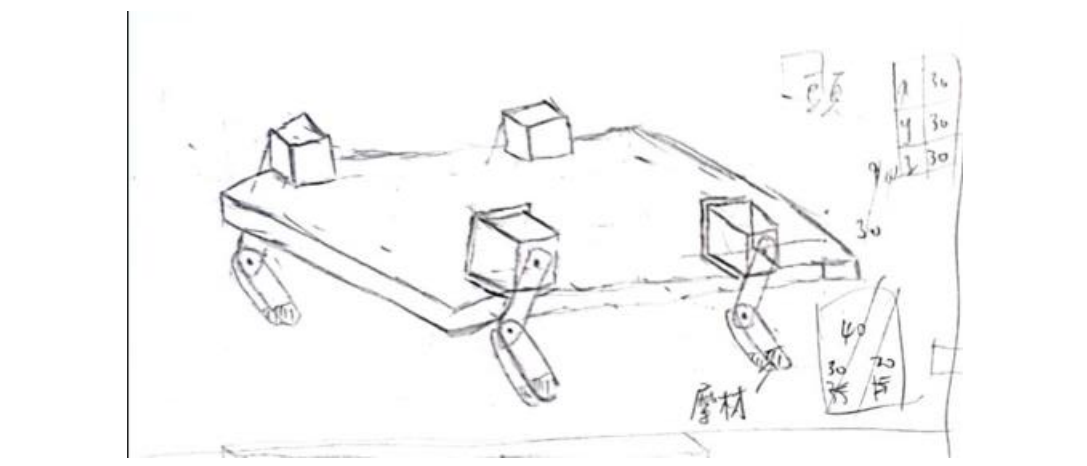
(2.)初始想法以及元件初稿(手繪):

初期先以手繪方式是想外型與功能如何達成

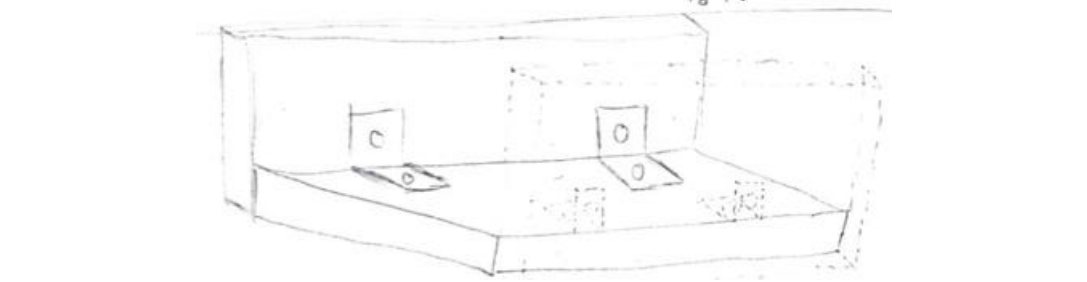
| 腿部 | 身體 |
|---|--|
|  |  |
| L 型固定板(身體腿腳連接) | 腳部摩擦材料 |



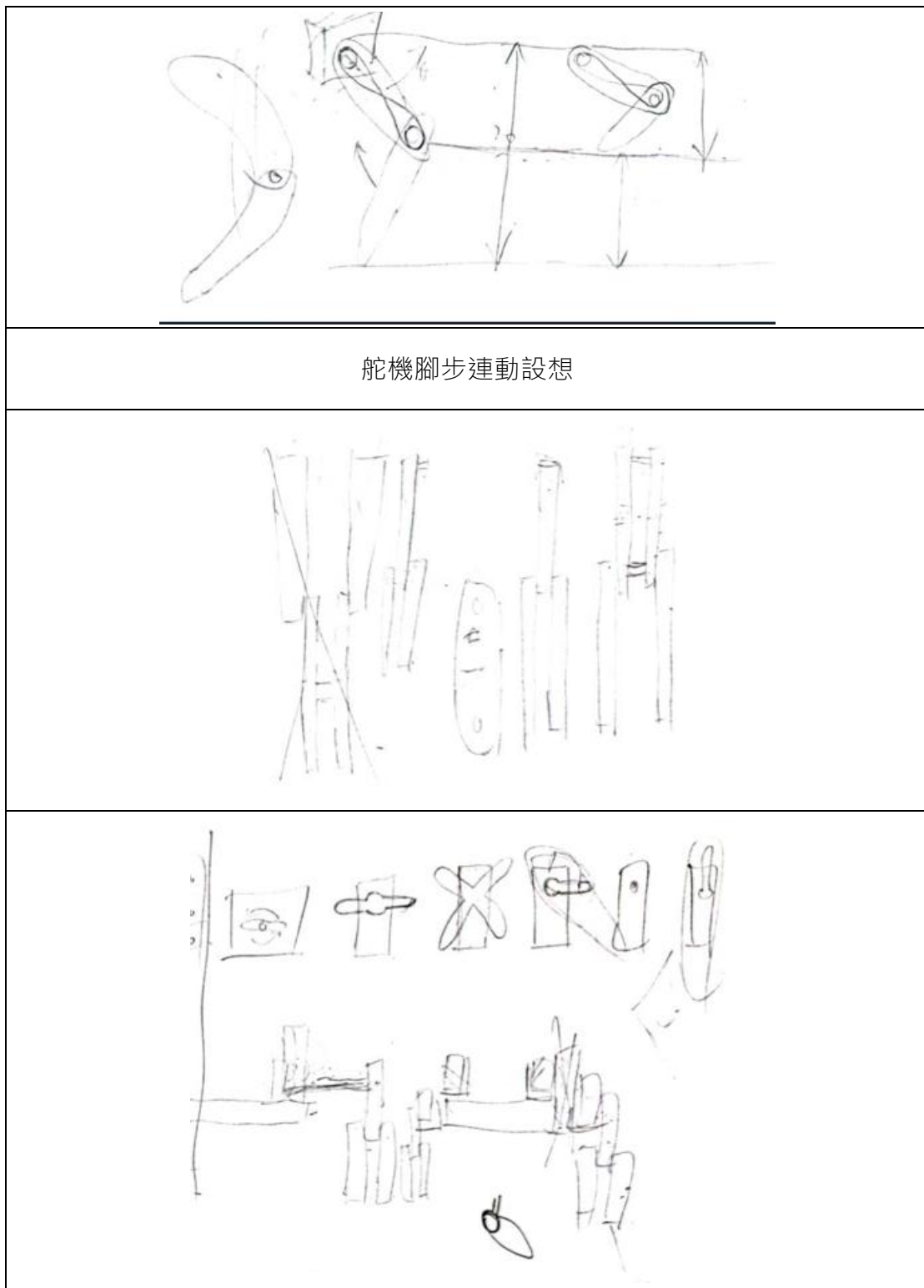
初始身體



初始構造



腳部初始設想

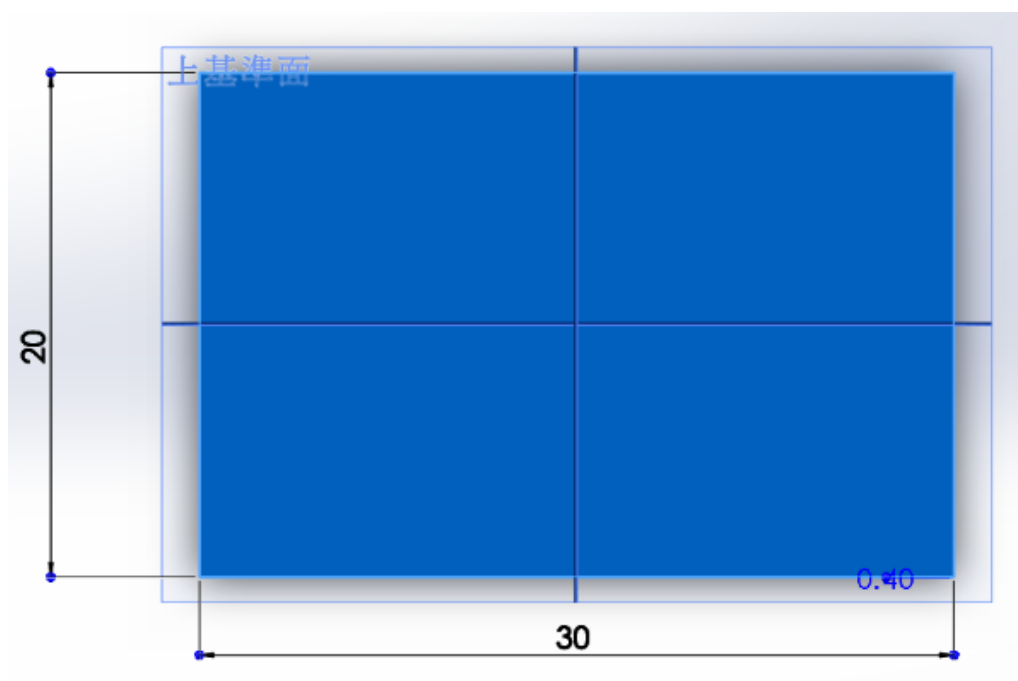


舵機腳步連動設想

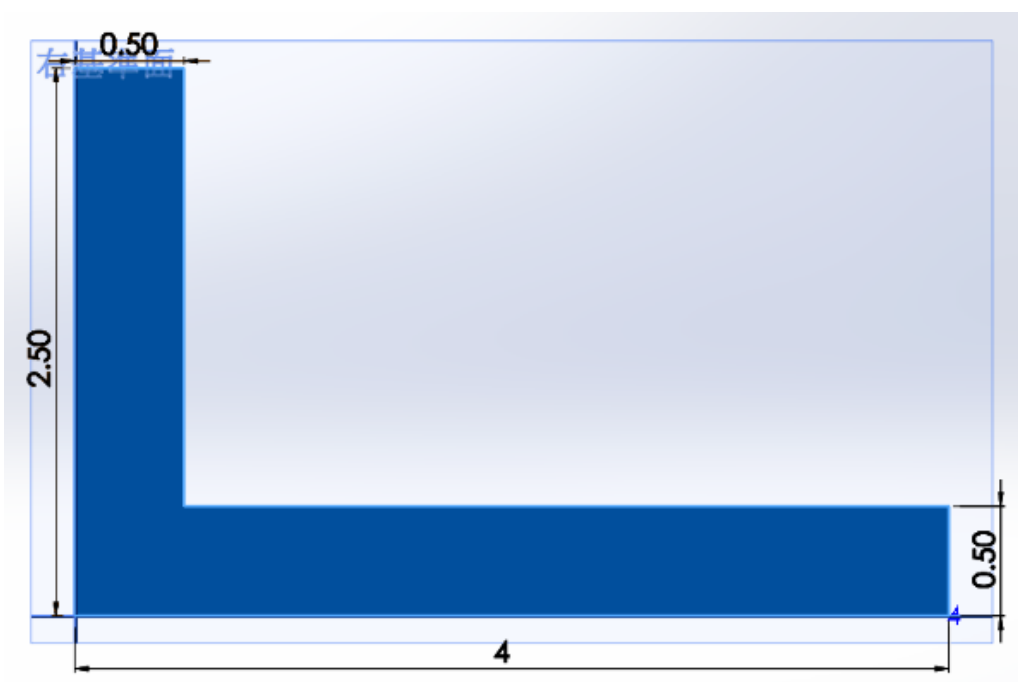
(3.)各部件起始細部介紹(單位:mm):

以 CAD 軟體，solidworker 繪製 3D 圖

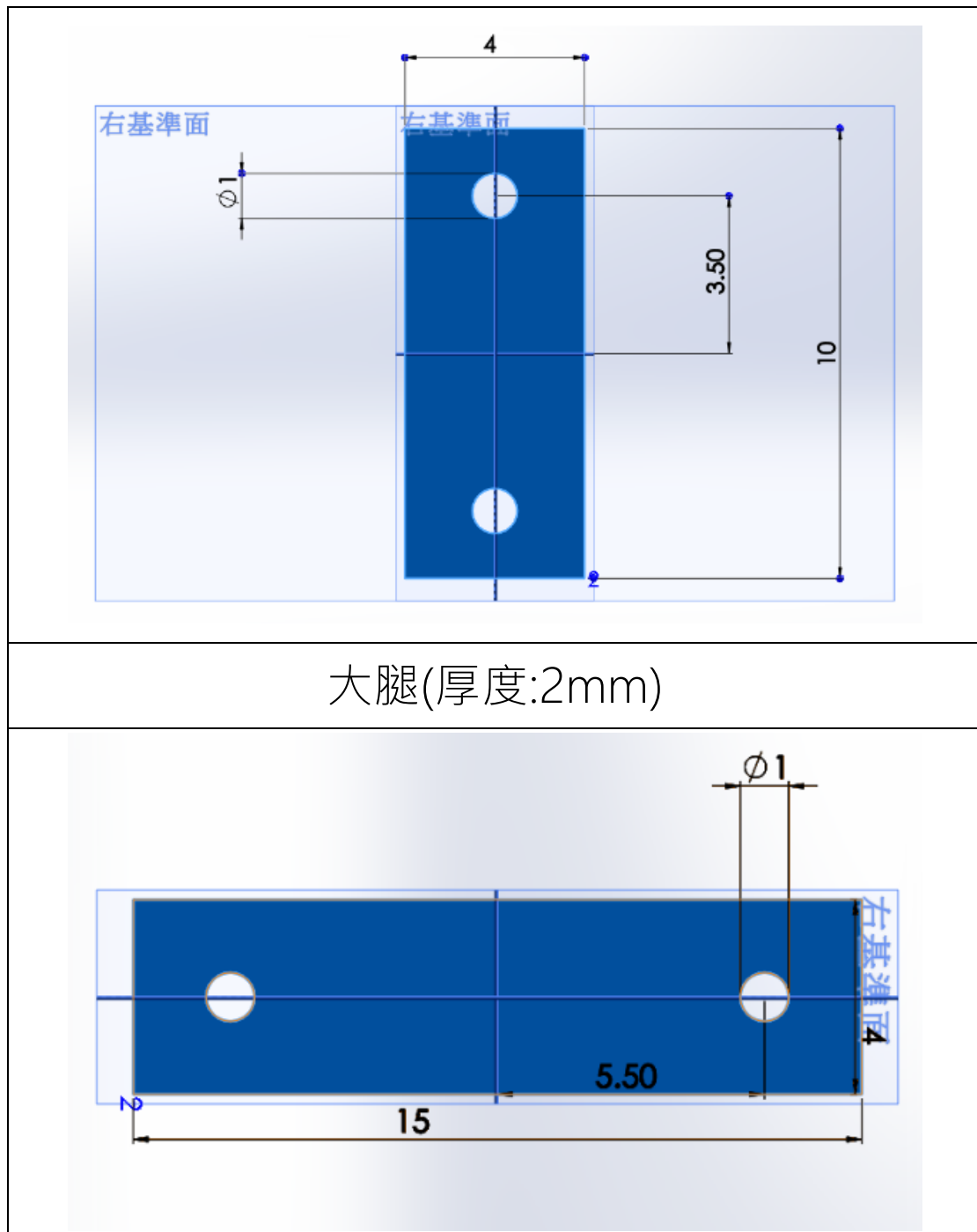
身體(厚度:0.4mm)



L 型固定器



小腿(厚度:0.5mm)



3.機器狗運動機構設計:

一開始我們想要大腿部分使用彈簧，膝蓋關節使用齒輪，想要的效果是讓下半部齒輪控制

大腿帶動小腿的旋轉，以及利用上半部身體的馬達或是舵機等動力來源拉伸彈簧，以結構控制帶動齒輪旋轉，藉以控制腿部運動。後來問過助教，了解了彈簧的使用方法以及在此方案中的不可行性

第二方案我們是想要改善彈簧-齒輪的結構，改用皮帶輪取代彈簧-齒輪的構造，但是在採購材料的時候發現經費超出範圍了。

所以目前的定案是想要使用八個舵機分別裝在四條腿的大腿-身體關節以及小腿-大腿關節以用來控制腿部運動。

4.機器狗部件補充及製作:

- (1.)除舵機以外之部件皆以 2mm 壓克力板雷射切割
- (2.)L 型固定器最終以舵機取代，當作關節使用
- (3.)身體使用一片壓克力板並在上面鑽孔以連接各部位，方便舵機帶動。

5. 電路設計

電路的設計主要可以分成電路接線與舵機控制。

(1.) 電路接線:

i. 舵機供電: 使用電池座連接麵包板，舵機的電線也連結到麵包板上。

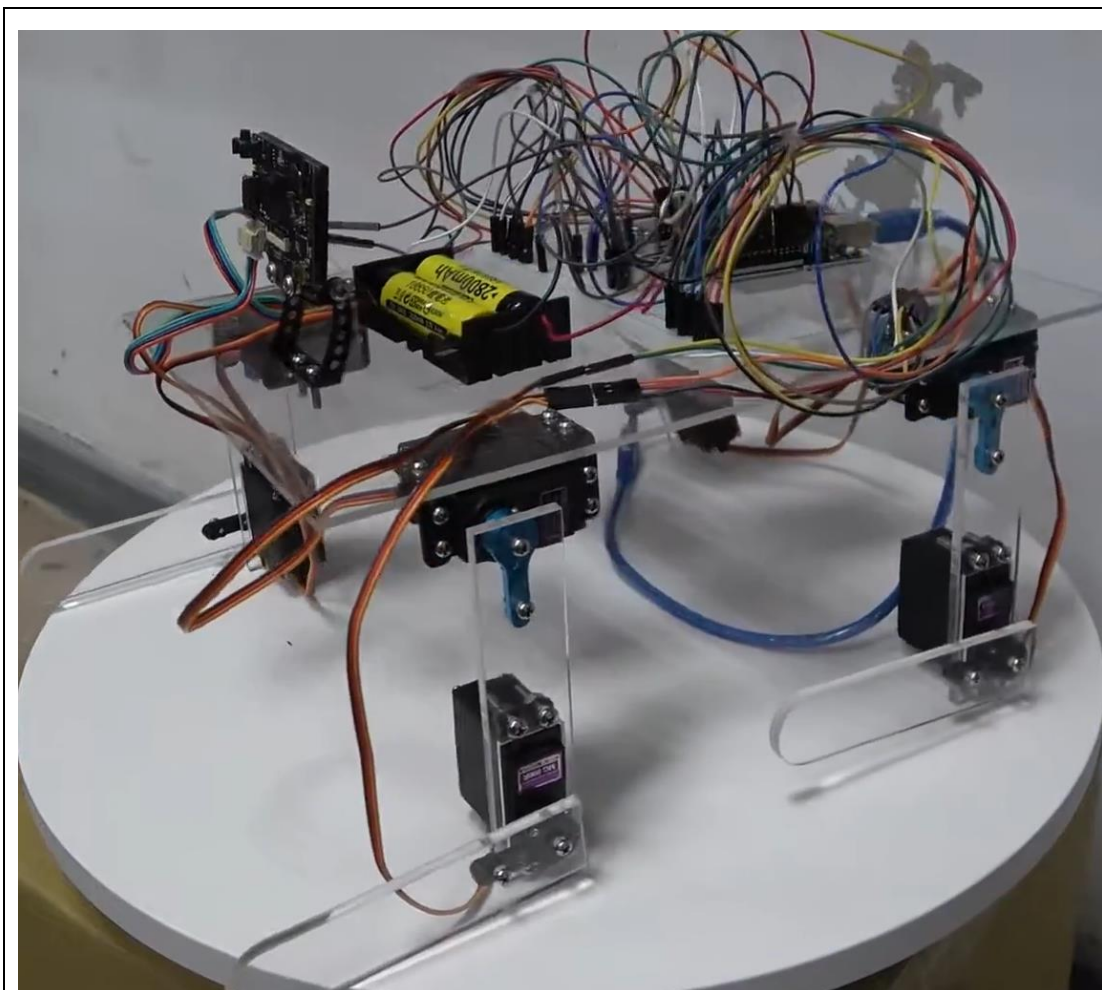
ii. Arduino 供電: 使用行動電源供電

一開始是用手持行動電源，但是這樣一來機器狗運作時要有人拿著行動電源。

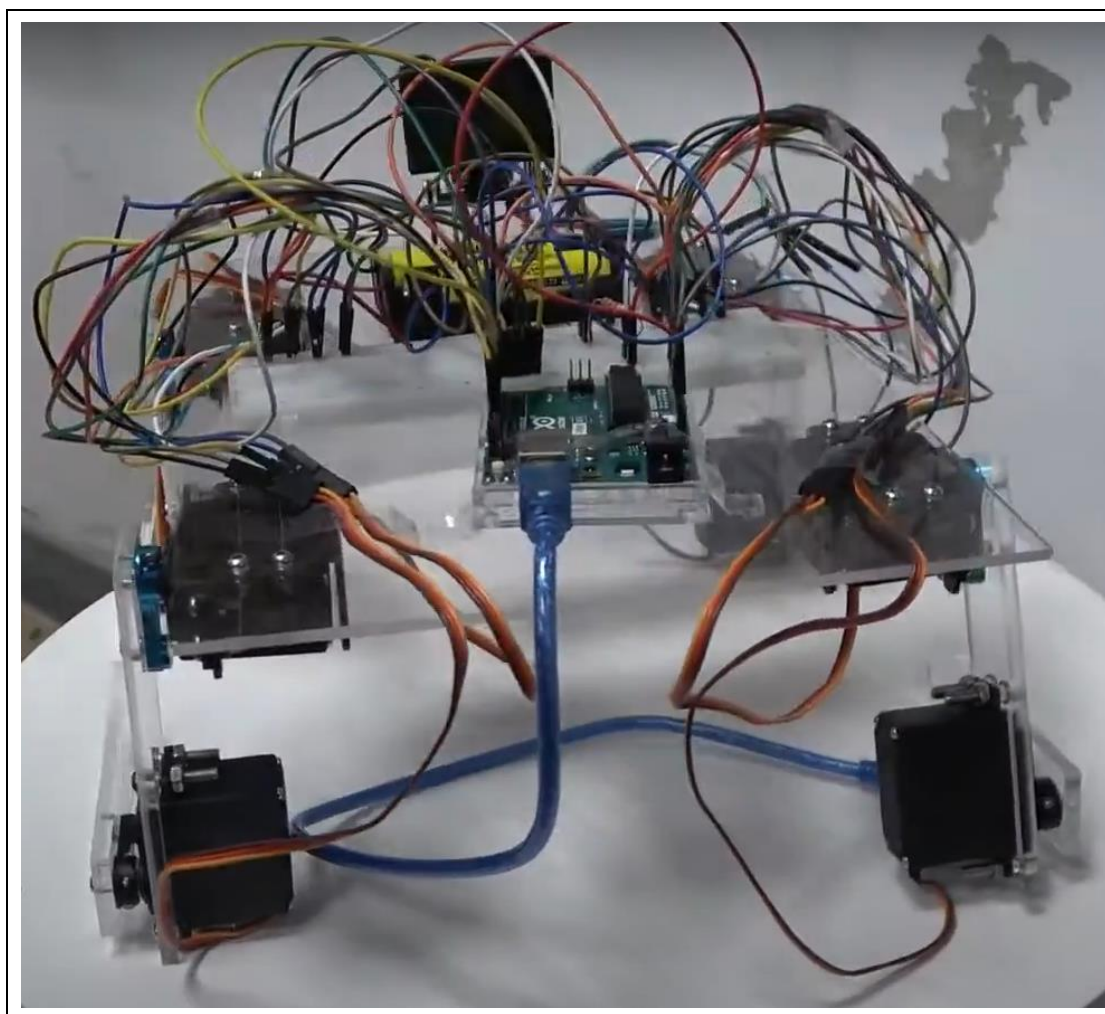
因為上述原因會導致機器狗不能離開人手自己運作，我們後來想出可以放置行動充電器於機器狗身上並將其固定，使其可以在行走時獨立運作，不須手持行動電源。

iii. Arduino 指令燒錄: 燒錄時移除行動電源使用筆電燒錄軟體程式碼。

| |
|-------|
| 機器狗左側 |
|-------|



機器狗後方



(2.)舵機控制:

Arduino 軟體程式控制，程式碼如附錄，不同功能使用之前要重新燒錄不同程式碼，我們主要分類為以下表格:

| | |
|-------|-------|
| 程式碼 1 | 起立&蹲下 |
| 程式碼 2 | 行走 |
| 程式碼 3 | 抬腳 |

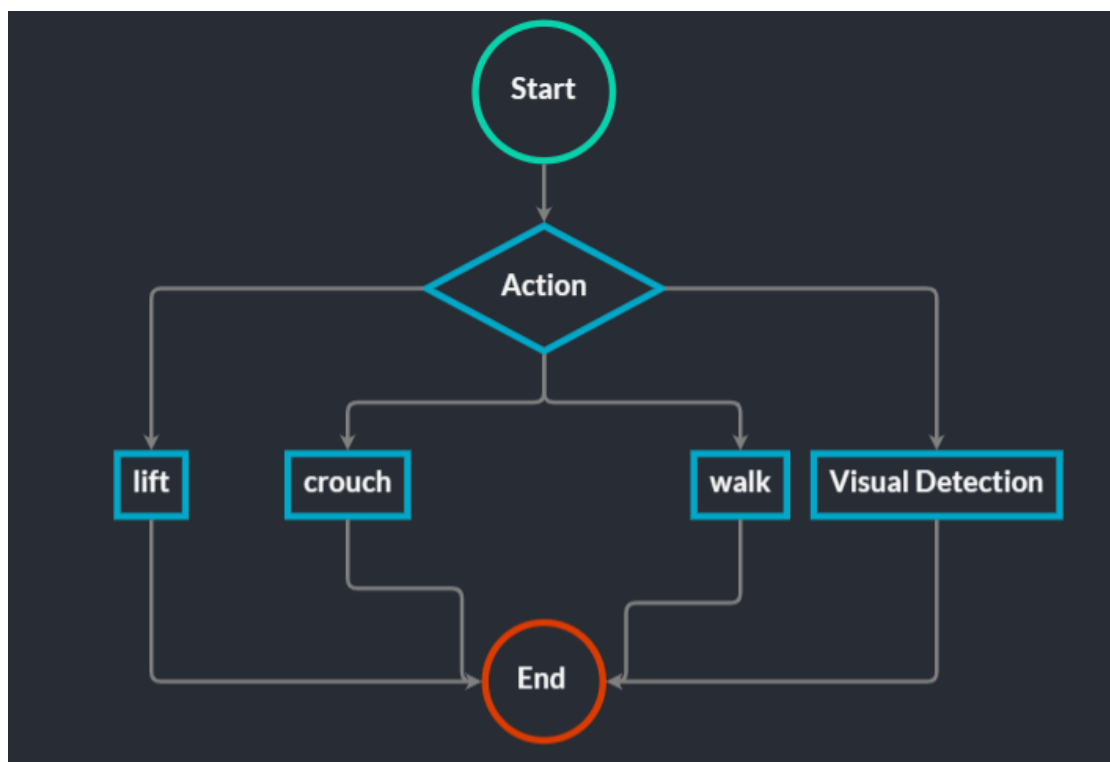
| | |
|-------|------|
| 程式碼 4 | 影像辨識 |
|-------|------|

(程式碼於附錄)

專題海報製作

使用網站 GoJS([Flowchart \(gojs.net\)](https://gojs.net))提供之 Flow chart 製作功能，該網站可使用手動方塊再進行編輯也可以使用程式碼，進行該 flow chart 製作時使用德是編輯程式碼的方法，因為相對位置會較為精準；雖然手動在操作上較為視覺化，但是會有不精準的位置誤差。

面板設計流程圖如下：



(流程圖程式碼於附錄，貼上至該網站即可重現)

海報最終成果:

國立中正大學機械工程學系
Department of Mechanical Engineering

機器人 CCUME Robot Competition

CAPSTONE COURSE 實作競賽

Legs(calf)

Component:
1.acrylic board
2.laser cutting

Design Concept:
Connect with thigh via steering gear, walk by friction with floor, controlling second angle



Legs(thighs)

Component:
1.acrylic board
2.laser cutting

Design Concept:
Connect body and thigh
Controlling first angle changing



Body

Component:
1.acrylic board
2.laser cutting

Design Concept:
connect thigh with steering gear



Joint

Component:
1.steering gear
2.screw-nut

Apply Concept:
Control the motion of legs




Flow chart



Electronic design



CCU Dog



主辦單位 國立中正大學機械工程學系

指導單位 智慧製造跨域整合人才培育計畫辦公室

協辦單位 國立中正大學前瞻製造系統頂尖研究中心

感謝單位 永進機械工業股份有限公司

發得科技工業股份有限公司

台灣三佳股份有限公司

新代科技股份有限公司

附錄(程式碼)

成品展示:

<https://youtu.be/roUsDg6dgGg?list=PLpoQKVpNUhOuH5BEZtyCszEVEbSvX2wP1>

軟體控制程式碼:

Github: [Hsuanyu-wang/CCU-DOG \(github.com\)](https://github.com/Hsuanyu-wang/CCU-DOG)

程式碼 1:蹲下&起立

```
#include <Servo.h>
```

```
Servo rfu; //宣告伺服馬達物件
```

```
Servo rfd;
```

```
Servo rbu;
```

```
Servo rbd;
```

```
Servo lfu;
```

```
Servo lfd;
```

```
Servo lbu;
```

```
Servo lbd;
```

```
void rfuw(int a){
```

```
    rfu.write(a+4);
```



```
}
```

```
void rfdw(int a){
```

```
    rfd.write(a-4);
```

```
}
```

```
void rbuw(int a){
```

```
    rbu.write(a-4);
```

```
}
```

```
void rbdw(int a){
```

```
    rbd.write(a+7);
```

```
}
```

```
void lfuw(int a){
```

```
    lfu.write(a+2);
```

```
}
```

```
void lfdw(int a){
```

```
    lfd.write(a-2);  
  
}  
  
void lbuw(int a){  
  
    lbu.write(a+8);  
  
}  
  
void lbdw(int a){  
  
    lbd.write(a-16);  
  
}  
  
void setup() {  
  
    // put your setup code here, to run once:  
  
    rfu.attach(A5);  
  
    rfd.attach(A4);  
  
    rbu.attach(A3);  
  
    rbd.attach(A2);  
  
    lfu.attach(10);  
  
    lfd.attach(11);  
  
    lbu.attach(12);  

```

```
lbd.attach(13);

rfuw(80);

rfdw(10);

rbuw(90);

rbdw(0);

lfuw(100);

lfdw(170);

lbuw(90);

lbdw(180);

delay(5000);

for (int i=1;i <= 20;i++){

    rfuw(80-4*i);

    rfdw(10+5.5*i);

    rbuw(90-4*i);

    rbdw(0+5.5*i);

    lfuw(100+4*i);

    lfdw(170-5.5*i);

    lbuw(90+4*i);

    lbdw(180-5.5*i);
```

```
    delay(1000);

}

for (int i=1;i <= 20;i++){

    rfuw(20+4*i);

    rfdw(110-5.5*i);

    rbuw(10+4*i);

    rbdw(100-5.5*i);

    lfuw(160-4*i);

    lfdw(70+5.5*i);

    lbuw(170-4*i);

    lbdw(80+5.5*i);

    delay(1000);

}

}

void loop() {

}

}
```

程式碼 2:行走

```
#include <Servo.h>
```

```
Servo rfu; //宣告伺服馬達物件
```

```
Servo rfd;
```

```
Servo rbu;
```

```
Servo rbd;
```

```
Servo lfu;
```

```
Servo lfd;
```

```
Servo lbu;
```

```
Servo lbd;
```

```
void rfuw(int a){
```

```
    rfu.write(a+4);
```

```
}
```

```
void rfdw(int a){
```

```
    rfd.write(a-4);
```

```
}
```

```
void rbuw(int a){
```

```
    rbu.write(a-4);  
}
```

```
void rbdw(int a){  
    rbd.write(a+7);  
}
```

```
void lfuw(int a){  
    lfu.write(a+2);  
}
```

```
void lfdw(int a){  
    lfd.write(a-2);  
}
```

```
void lbuw(int a){  
    lbu.write(a+8);  
}
```

```
void lbdw(int a){  
  
    lbd.write(a-20);  
  
}
```

```
void setup() {  
  
    rfu.attach(A5);  
  
    rfd.attach(A4);  
  
    rbu.attach(A3);  
  
    rbd.attach(A2);  
  
    lfu.attach(10);  
  
    lfd.attach(11);  
  
    lbu.attach(12);  
  
    lbd.attach(13);  
  
    rfuw(60);  
  
    rfdw(80);  
  
    rbuw(50);  
  
    rbdw(60);  
  
    lfuw(120);  
  
    lfdw(100);  
  
    lbuw(130);
```

```
    lbdw(120);

    delay(10000);

}

void loop()

{

    rfuw(60+5);

    rfdw(80+5);

    rbuw(50-5);

    rbdw(60-5);

    lfuw(120+7);

    lfdw(100+7);

    lbuw(130-5);

    lbdw(120-5);

    delay(200);

    rfuw(60+5);

    rfdw(80-5);

    rbuw(50-5);

    rbdw(60-5);

    lfuw(120+7);
```


lfdw(100+7);

lbuw(130-5);

lbdw(120+5);

delay(200);

rfuw(60);

rfdw(80-5);

rbuw(50);

rbdw(60);

lfuw(120);

lfdw(100);

lbuw(130);

lbdw(120+5);

delay(200);

rfuw(60);

rfdw(80);

rbuw(50);

rbdw(60);

lfuw(120);

lfdw(100);

lbuw(130);

lbdw(120);

delay(200);

rfuw(60-5);

rfdw(80-5);

rbuw(50+5);

rbdw(60+5);

lfuw(120-5);

lfdw(100-5);

lbuw(130+5);

lbdw(120+5);

delay(200);

rfuw(60-5);

rfdw(80-5);

rbuw(50+5);

rbdw(60-5);

lfuw(120-5);

lfdw(100+5);

lbuw(130+5);

lbdw(120+5);

delay(200);

rfuw(60);

rfdw(80);

rbuw(50);

rbdw(60-5);

lfuw(120);

lfdw(100+5);

lbuw(130);

lbdw(120);

delay(200);

rfuw(60);

rfdw(80);

rbuw(50);

rbdw(60);

lfuw(120);

lfdw(100);

lbuw(130);

lbdw(120);

```
    delay(1000);  
}
```

程式碼 3:抬腳

```
#include <Servo.h>  
  
Servo rfu; //宣告伺服馬達物件  
  
Servo rfd;  
  
Servo rbu;  
  
Servo rbd;  
  
Servo lfu;  
  
Servo lfd;  
  
Servo lbu;  
  
Servo lbd;  
  
  
void rfuw(int a){  
    rfu.write(a+4);  
}  
  
  
void rfdw(int a){  
    rfd.write(a-4);
```

```
}
```

```
void rbuw(int a){
```

```
    rbu.write(a-4);
```

```
}
```

```
void rbdw(int a){
```

```
    rbd.write(a+7);
```

```
}
```

```
void lfuw(int a){
```

```
    lfu.write(a+2);
```

```
}
```

```
void lfdw(int a){
```

```
    lfd.write(a-2);
```

```
}
```

```
void lbuw(int a){
```

```
    lbu.write(a+8);

}

void lbdw(int a){

    lbd.write(a-16);

}

void setup() {

    // put your setup code here, to run once:

    rfu.attach(A5);

    rfd.attach(A4);

    rbu.attach(A3);

    rbd.attach(A2);

    lfu.attach(10);

    lfd.attach(11);

    lbu.attach(12);

    lbd.attach(13);

    rfuw(60);

    rfdw(80);

    rbuw(60);
```

rbdw(50);

lfuw(120);

lfdw(100);

lbuw(120);

lbdw(130);

delay(10000);

rfuw(60);

rfdw(30);

lfuw(160);

lfdw(80);

lbuw(120);

lbdw(150);

delay(2000);

rfuw(60);

rfdw(30);

rbuw(10);

rbdw(30);

lfuw(160);

lfdw(80);

```
lbuw(120);  
  
lbdw(150);  
  
delay(8000);  
  
rfuw(60);  
  
rfdw(30);  
  
lfuw(160);  
  
lfdw(80);  
  
lbuw(120);  
  
lbdw(150);  
  
delay(2000);  
  
rfuw(60);  
  
rfdw(80);  
  
rbuw(60);  
  
rbdw(50);  
  
lfuw(120);  
  
lfdw(100);  
  
lbuw(120);  
  
lbdw(130);  
  
}
```



```
void loop() {
```

```
}
```

程式碼 4:影像辨識

```
#include "HUSKYLENS.h"
```

```
#include "SoftwareSerial.h"
```

```
#include <Servo.h>
```

```
Servo rfu; //宣告伺服馬達物件
```

```
Servo rfd;
```

```
Servo rbu;
```

```
Servo rbd;
```

```
Servo lfu;
```

```
Servo lfd;
```

```
Servo lbu;
```

```
Servo lbd;
```

```
SoftwareSerial mySerial(A0, A1);
```

```
HUSKYLENS huskylens;
```

```
byte seven_seg_digits[10][7] = {
```

```
{ 1,0,0,1,1,1,1 }, // = 1 // define the LED digit patterns, from 0 - 9
```

{ 0,0,1,0,0,1,0 }, // = 2 // 0 = LED on, 1 = LED off, in this order:

{ 0,0,0,0,1,1,0 }, // = 3 //Arduino pin: 2,3,4,5,6,7,8 (Mapping a,b,c,d,e,f,g of

Seven-Segment LED)

{ 1,0,0,1,1,0,0 }, // = 4

{ 0,1,0,0,1,0,0 }, // = 5

{ 0,1,0,0,0,0,0 }, // = 6

{ 0,0,0,1,1,1,1 }, // = 7

{ 0,0,0,0,0,0,0 }, // = 8

{ 0,0,0,1,1,0,0 }, // = 9

{ 0,0,0,0,0,0,1 } // = 0

};

void rfuw(int a){

 rfu.write(a+4);

}

void rfdw(int a){

 rfd.write(a-4);

}

```
void rbuw(int a){  
  
    rbu.write(a-4);  
  
}
```

```
void rbdw(int a){  
  
    rbd.write(a+7);  
  
}
```

```
void lfuw(int a){  
  
    lfu.write(a+2);  
  
}
```

```
void lfdw(int a){  
  
    lfd.write(a-2);  
  
}
```

```
void lbuw(int a){  
  
    lbu.write(a+8);  
  
}
```

```
void lbdw(int a){  
  
    lbd.write(a-16);  
  
}  
  
void setup() {  
  
    pinMode(2, OUTPUT);  
  
    pinMode(3, OUTPUT);  
  
    pinMode(4, OUTPUT);  
  
    pinMode(5, OUTPUT);  
  
    pinMode(6, OUTPUT);  
  
    pinMode(7, OUTPUT);  
  
    pinMode(8, OUTPUT);  
  
    rfuw(60);  
  
    rfdw(80);  
  
    rbuw(60);  
  
    rbdw(50);  
  
    lfuw(120);  
  
    lfdw(100);  
  
    lbuw(120);
```

```

lbdw(130);

Serial.begin(115200);

mySerial.begin(9600);

while (!huskylens.begin(mySerial))

{

    Serial.println(F("Begin failed!"));

    Serial.println(F("1.Please recheck the \"Protocol Type\" in
HUSKYLENS (General Settings>>Protocol Type>>Serial 9600)"));

    Serial.println(F("2.Please recheck the connection."));

    delay(100);

}

huskylens.clearCustomText();

}

void sevenSegWrite(byte digit) { // 在七段顯示器上顯示指定的一個數字

byte pin = 2;

for (byte seg = 0; seg < 7; ++seg) {

digitalWrite(pin, seven_seg_digits[digit][seg]);

++pin;

```

```
}
```

```
}
```

```
void loop()
```

```
{
```

```
    if (!huskylens.request()) Serial.println(F("Fail to request data from  
HUSKYLENS, recheck the connection!"));
```

```
    else if(!huskylens.isLearned()) Serial.println(F("Nothing learned, press  
learn button on HUSKYLENS to learn one!"));
```

```
    else if(!huskylens.available()) Serial.println(F("No block or arrow appears  
on the screen!"));
```

```
    else
```

```
    {
```

```
        //Serial.println(F("#####"));
```

```
        while (huskylens.available())
```

```
        {
```

```
            HUSKYLENSResult result = huskylens.read();
```

```

        sevenSegWrite(result.ID-1);

    }

}

}

```

流程圖程式碼:

```

{ "class": "GraphLinksModel",

  "linkFromPortIdProperty": "fromPort",

  "linkToPortIdProperty": "toPort",

  "nodeDataArray": [

    {"key":-1,"category":"Start","loc":"200 0","text":"Start"},

    {"key":-2,"category":"End","loc":"200 300","text":"End"},

    {"category":"Conditional","text":"Action","key":-3,"loc":"200 100"},

    {"text":"lift","key":-4,"loc":"0 200"},

    {"text":"crouch","key":-5,"loc":"100 200"},

    {"text":"walk","key":-6,"loc":"300 200"},

    {"text":"Visual Detection","key":-7,"loc":"400 200"}

  ],

  "linkDataArray": [

```

```

{"from":-1,"to":-
3,"fromPort":"B","toPort":"T","points":[200,36.75,200,46.75,200,51.5622756958
0078,200,51.56227569580078,200,56.37455139160156,200,66.3745513916015
6]],
{"from":-3,"to":-
6,"fromPort":"B","toPort":"T","visible":false,"points":[200,133.62544860839841,
200,143.62544860839841,200,157.9688621520996,300,157.9688621520996,3
00,172.3122756958008,300,182.3122756958008]],
{"from":-3,"to":-
7,"fromPort":"R","toPort":"T","visible":false,"points":[261.46607971191406,100,2
71.46607971191406,100,400,100,400,136.1561378479004,400,172.31227569580
08,400,182.3122756958008]],
{"from":-3,"to":-
4,"fromPort":"L","toPort":"T","visible":false,"points":[138.53392028808594,100,1
28.53392028808594,100,0,100,0,136.1561378479004,0,172.3122756958008,0,1
82.3122756958008]],
{"from":-3,"to":-
5,"fromPort":"B","toPort":"T","visible":false,"points":[200,133.62544860839841,
200,143.62544860839841,200,157.9688621520996,100,157.9688621520996,10

```



```

0,172.3122756958008,100,182.3122756958008]],

{"from":-4,"to":-

2,"fromPort":"B","toPort":"L","points":[0,217.68772430419924,0,227.68772430

419924,0,300,79.125,300,158.25,300,168.25,300]]},

{"from":-7,"to":-

2,"fromPort":"B","toPort":"R","points":[400,217.68772430419924,400,227.6877

2430419924,400,300,320.875,300,241.75,300,231.75,300]]},

{"from":-5,"to":-

2,"fromPort":"B","toPort":"L","points":[100,217.68772430419924,100,227.68772

430419924,100,252.66667938232422,158.25,252.66667938232422,158.25,30

0,168.25,300]]},

{"from":-6,"to":-

2,"fromPort":"B","toPort":"R","points":[300,217.68772430419924,300,227.6877

2430419924,300,260.0000228881836,241.75,260.0000228881836,241.75,300,

231.75,300]]

}}

```