

行政院國家科學委員會補助
大專學生參與專題研究計畫研究成果報告

* 計畫 *
* : 立體影像物體追蹤定位系統之軟體設計 *
* 名稱 *

執行計畫學生：張文俊

學生計畫編號：NSC 97 - 2815 - C - 150 - 003 - E

研究期間： 年7月1日至 年2月底止，計8個月

指導教授：季永炤

執行單位：國立虎尾科技大學自動化工程系

中華民國

98年

3月

23日

摘要

本研究計劃使用兩台攝像機(CMUcam)所組成雙視覺系統裝置於機器上，雙視覺系統對欲搜尋物體做影像記憶與影像處理，找出物體質心，用伺服馬達調整攝像機(CMUcam)角度，使物體質心位於圖像中心，當物體移動時，立即調整伺服馬達角度讓攝像機(CMUcam)能立即追上物體之移動，當搜尋物體質心位於攝像機中間時，利用三角測距法計算物體與攝像機之間的距離。

目錄	
導論	I
目錄	II
圖目錄	III
表目錄	IV
第一章 序論	1
1-1 研究動機	1
1-2 研究方法	1
1-2 研究大綱	1
第二章 硬體架構	2
2-1 solid work 打樣	2
2-2 CMUcam	3
2-3 BS2	4
2-4 伺服馬達	5
2-5 完成品	5
第三章 系統架構	7
3-1 最佳相片	7
3-2 最大搜尋範圍	8
3-3 計算公式	9
3-4 系統流程	11
第四章 實驗紀錄	12
第五章 結論	15
參考文獻	16
附錄(1) BS2 程式	17

圖目錄

圖 1-1 系統架構圖	1
圖(2-1)solid work 圖	3
圖(2-2)solid work 圖	3
圖(2-3)CMUcam 圖	4
圖(2-4)CMUcam 流程圖	4
圖(2-5)BS2	5
圖(2-6)伺服馬達	6
圖(2-7)完成品 1	6
圖(2-8)完成品 2	7
圖(2-9)完成品 3	7
圖(3-1)複雜背景圖	8
圖(3-2)單純背景圖	8
圖(3-3)RGB 平均值與誤差值	8
圖(3-4)最大搜尋範圍圖	9
圖(3-5)三角測距(1)	10
圖(3-6)三角測距(2)	11
圖(3-7)三角測距(3)	11
圖(3-8)系統流程圖	12
圖(4-1)實驗紀錄 1	13
圖(4-2)實驗紀錄 2	13
圖(4-3)實驗紀錄 3	14
圖(4-4)實驗紀錄 4	14
圖(4-5)實驗紀錄 5	15

表目錄

表(1)BS2 規格	5
表(2)實驗數據 1.....	13
表(3)實驗數據 2.....	13
表(4)實驗數據 3.....	14
表(5)實驗數據 4.....	14
表(6)實驗數據 5.....	15

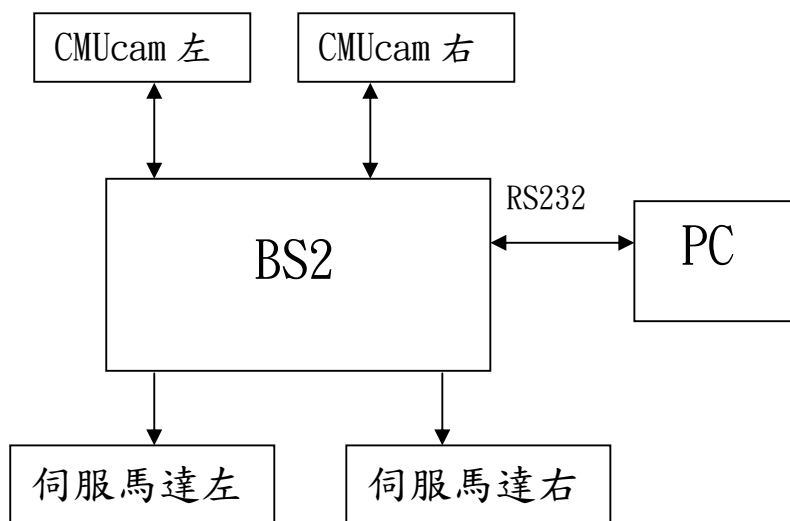
第一章 緒論

1-1 研究動機

目前大多數的小型機器人都以單攝相機取像做影像處理[1][2]，這樣的小型機器人只能判斷方向，而且需要大量的硬體去做影像處理，對於小型機器人來說是一種龐大的負擔，而 CMUcam[3]來說，他只搜尋物體質心，可以減少影像對硬體式的需求，而且雙視覺系統[4][5]可藉由伺服馬達之脈波計算物體與 CMUcam 的距離。

1-2 研究方法

首先建立微控制器(BS2)[6]與 CMUcam 之間的串列通訊，借用 CMUcam 回傳之質心座標來判斷物體位於何處，調整伺服馬達角度讓物體質心位於圖像中心，當物體質心於中間時，將伺服馬達脈波轉換成角度，將角度代入公式計算獲得物體與 CMUcam 之間距離[7]。



圖(1-1)系統架構圖

1-3 研究大綱

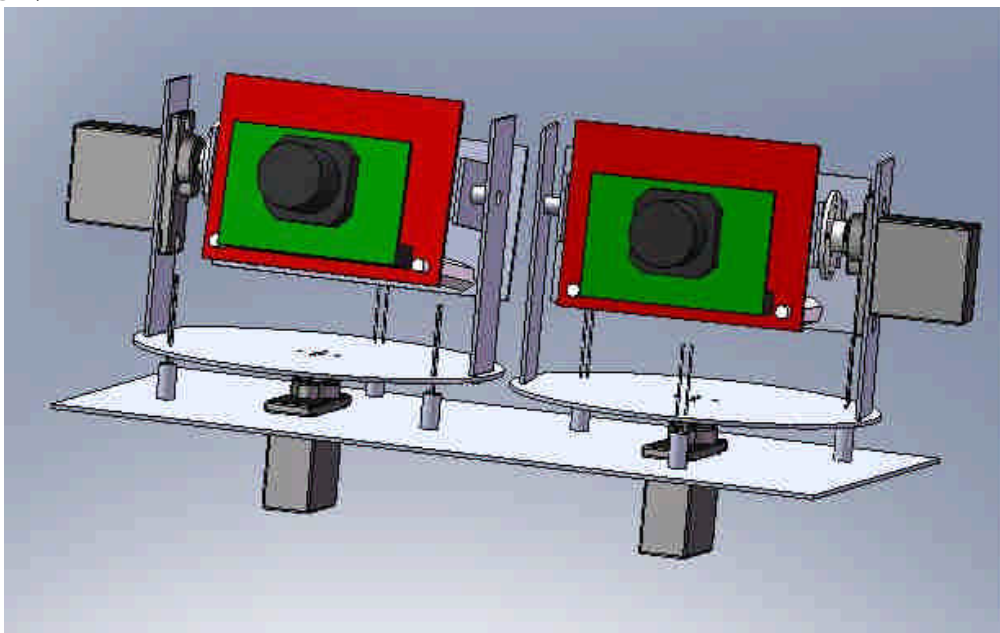
本計劃乃製作一雙視覺追蹤系統，此篇報告分為四個章節，第一章緒論，針對本計劃做出簡介，說明研究動機、目的及方法；第二章硬體架構，是關於本計劃所製作硬體，從 solid work 打樣到實體以及 CMUcam、BS2 和伺服馬達簡介；第三章系統架構，關於本計畫流程設計與運用公式；第四章為實驗數據，比較紀錄值與實際時；第五章為結論，為討論本計劃之結論與檢討。

第二章 硬體架構

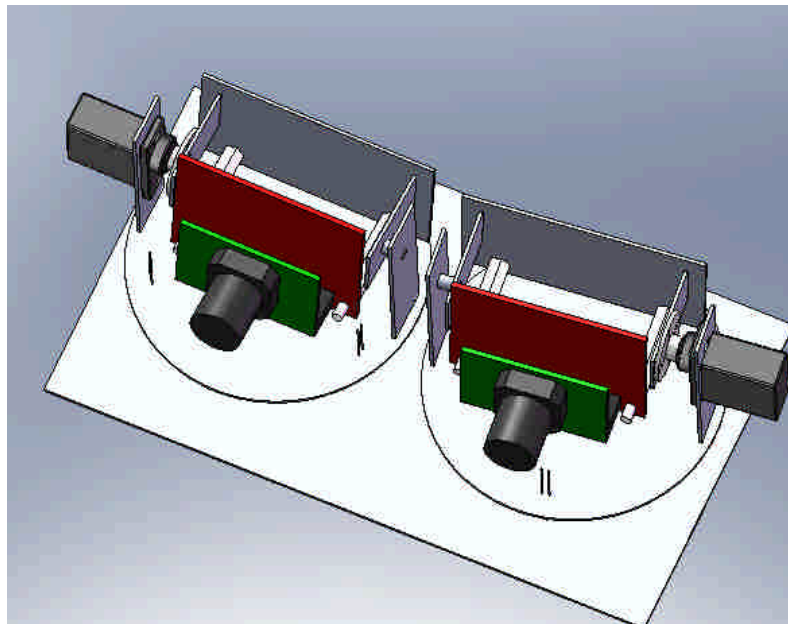
本章節主要說明本計劃所需之硬體，包括 solid work 電腦打樣、BS2、伺服馬達及最後實體。

2-1 solid work 打樣

因應本計劃所需，利用軟體 solid work 進行零件與結構設計，已符合本計劃所需之動態特性。



圖(2-1)solid work 圖



圖(2-2)solid work 圖

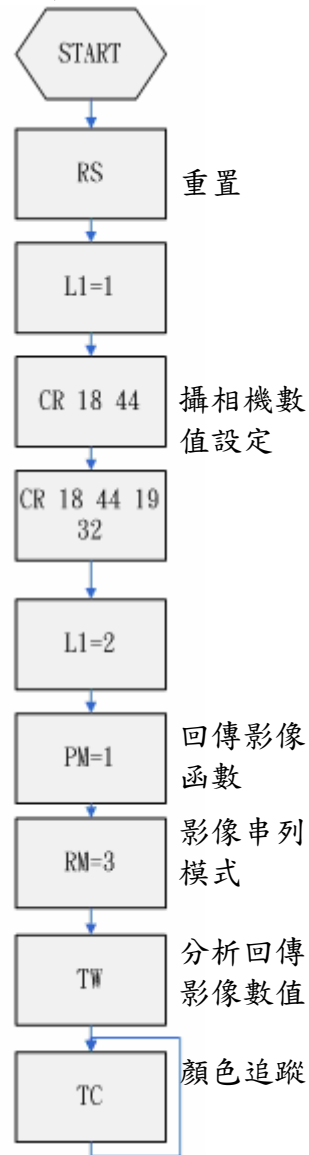
2-2 CMUcam

依本計劃所需，使用 CMUcam 攝相機，CMUcam 攝相機是由 CMU(卡內基大學)所設計，其基本規格如下：



圖(2-3)CMUcam 圖

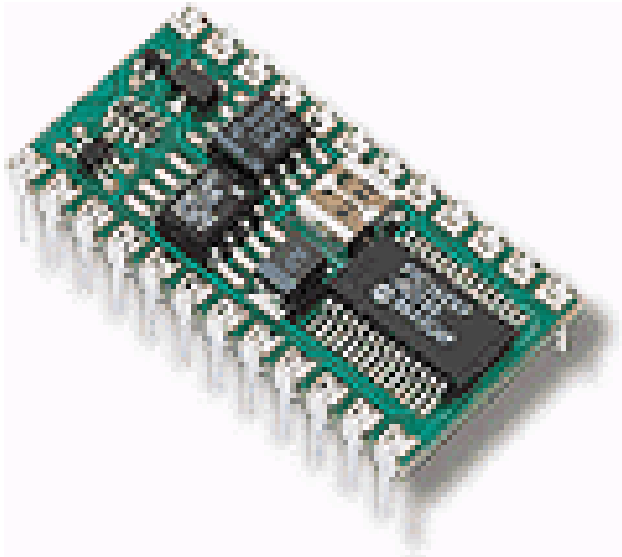
電壓:DC 6~9v
RS-232 串列通訊
鮑率:115200/38400/19200/9600
解析度:80*143



圖(2-4)CMUcam 流程圖

2-3 BS2

依本計劃所需，使用微控器為 BS2，是由智凌科技有限公司所設計生產，其中，主控制晶片是由 microchip 所生產的 PIC16F57 晶片，使用語言為 BASIC 語言，其基本規格所下所介紹：



圖(2-5)BS2

晶片	Microchip PIC 16C57
輸入 / 輸出數目	16 + 2 專屬串 聯 I/O
尺寸	30mm(L) x 16mm(W) x 9mm(D)
程式執行速度	4,000 指令 / 秒
處理器速度	20 MHz
EEPROM	2K 位元
Scratch Pad RAM	不適用
電流 @5v	8mA running, 100uA in sleep modes
個人電腦輸出 介面	串聯埠
每輸入/輸出所 需電流 (Source/Sink)	20 mA / 25 mA
PBASIC 指令	36
插座	24-pin DIP module
編輯程式	STAMP2. EXE

表(1)BS2 規格

2-4 伺服馬達

依本計劃所需，使用四顆伺服馬達[8]，伺服馬達是由 公司生產，其基本規格如下所示：

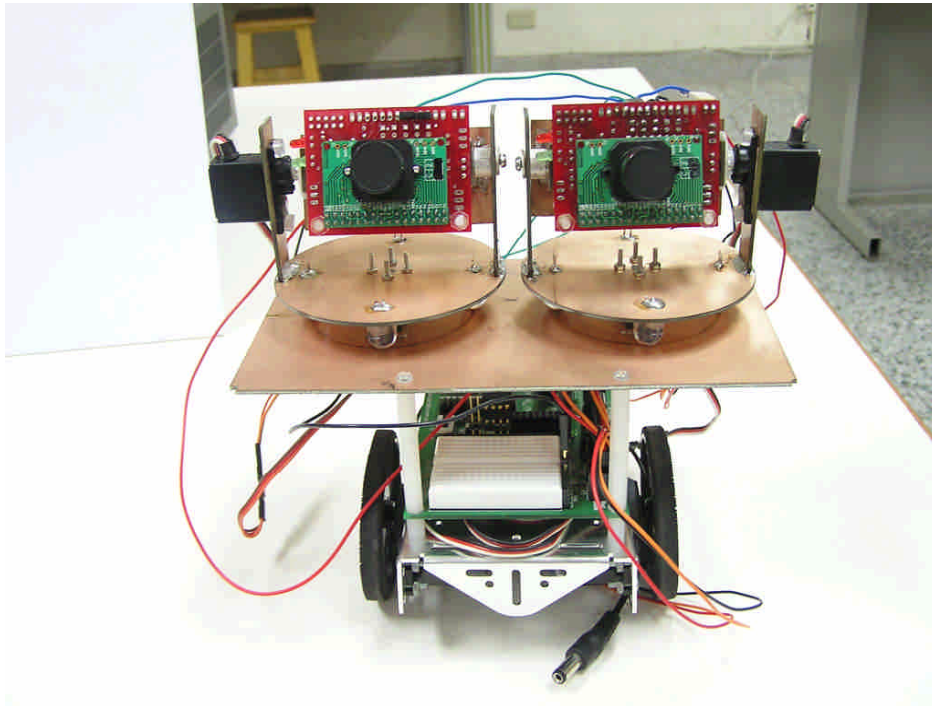


電壓:6V
扭力:0.84kg-cm
轉速:0.06

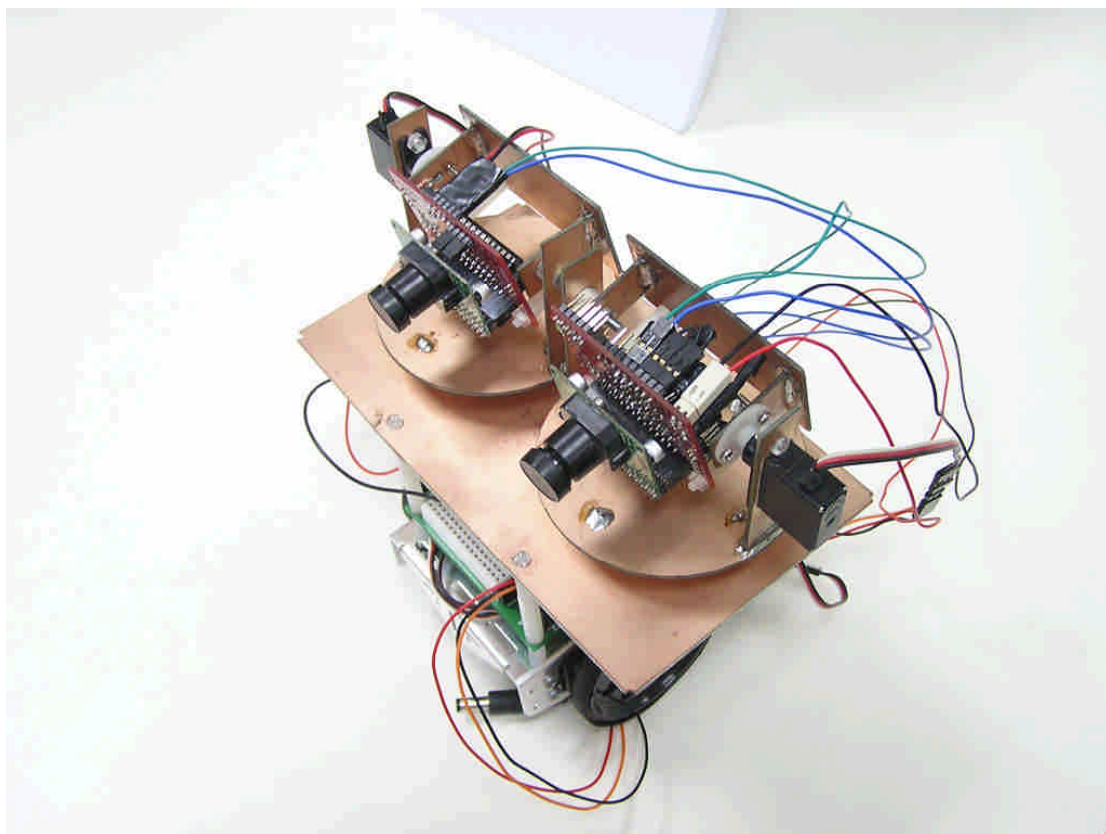
圖(2-6)伺服馬達

2-5 完成品

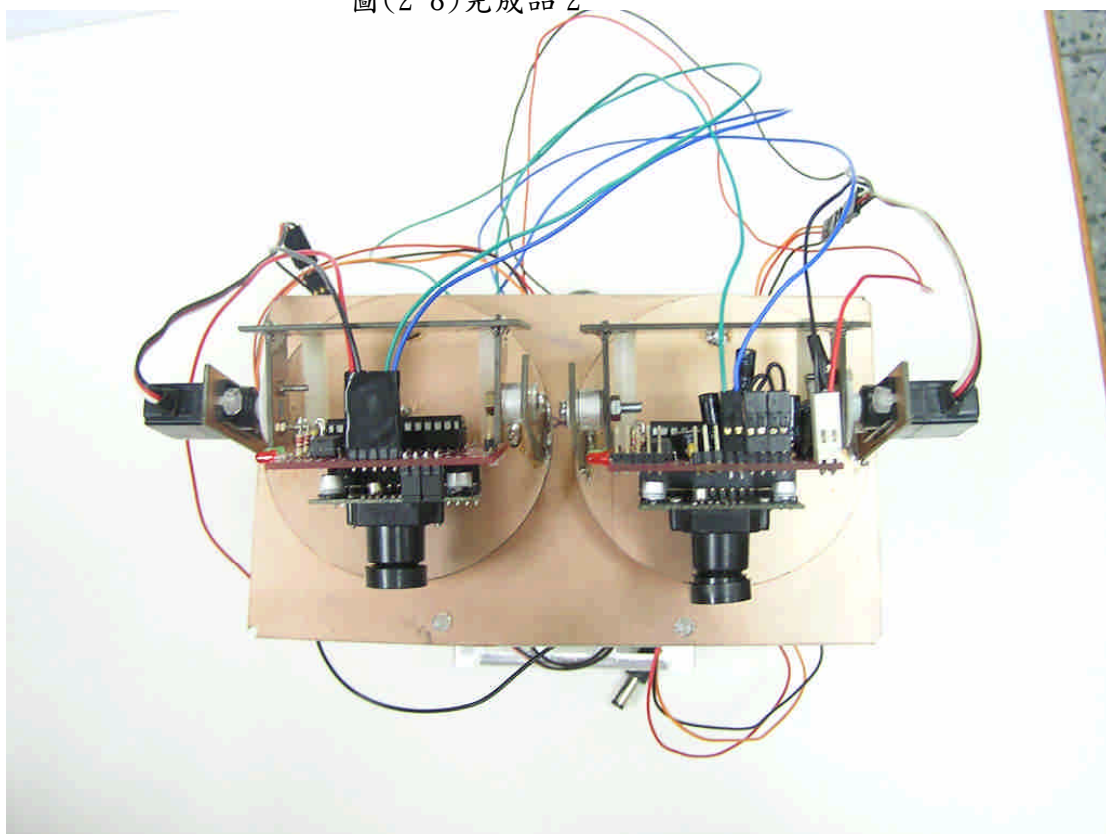
依 solid work 所打樣圖，利用 與雕刻機進行零件製作，將零件組裝後，接合處使用焊錫進行補強，其完成圖如下所示：



圖(2-7)完成品 1



圖(2-8)完成品 2



圖(2-9)完成品 3

第三章 系統架構

本章主要說明 CMUcam 攝相機如何獲得最佳相片、最大搜尋範圍、公式運用與程式流程。

3-1 最佳相片

因本計劃中使用 CMUcam 攝相機，CMUcam 攝相機在追蹤物體前需記憶顏色，故在取像前，將背景簡單化，如單一顏色(白色為最佳)，下面將比較背景對追蹤物體之 RGB 影響。



圖(3-1)複雜背景圖



圖(3-2)單純背景圖

	Mean	Deviation
Red	109	37
Green	88	51
Blue	48	33
low	mean	high

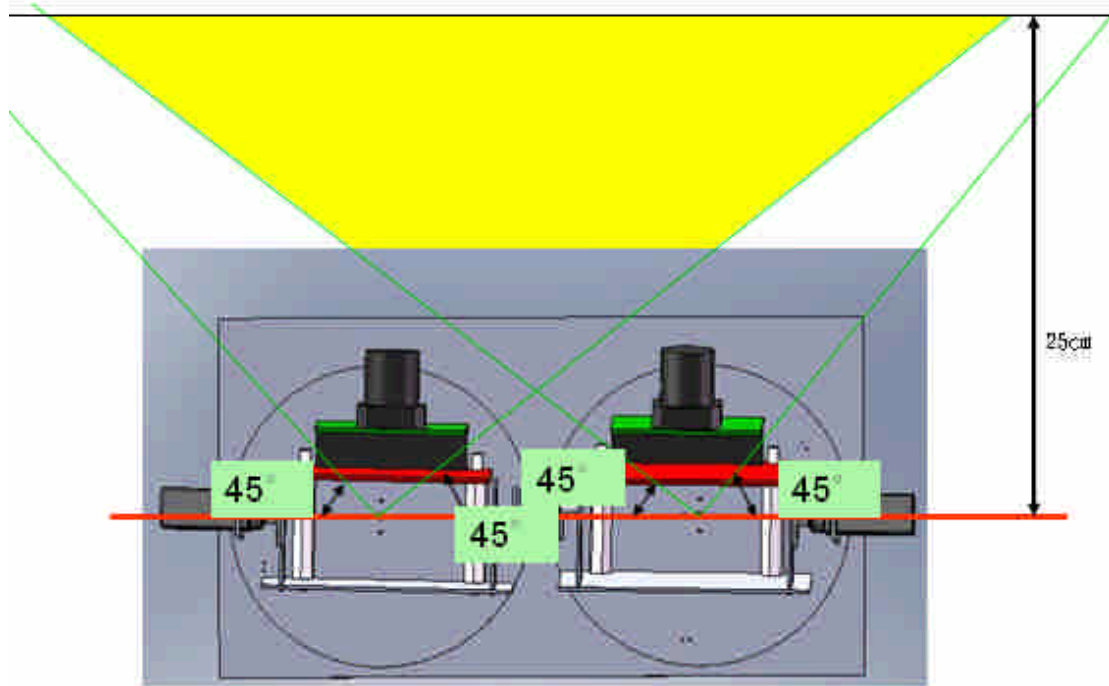
	Mean	Deviation
Red	98	18
Green	118	53
Blue	82	39
low	mean	high

圖(3-3)RGB 平均值與誤差值

由上圖()與圖()中，可以獲得背景對追蹤物體的 RGB 值之影響，在白色背景下，R 值的平均值較背景複雜來的低，固第一次取像做顏色記憶時，盡量將背景改為白色，這樣可以降低因背景所帶來的誤差值。

3-2 最大搜尋範圍

因本計劃中使用 CMUcam 攝相機與伺服馬達組合，擴大了 CMUcam 攝相機可搜尋之範圍，但還是無法搜尋到的地方，下圖將說明可搜尋之範圍。

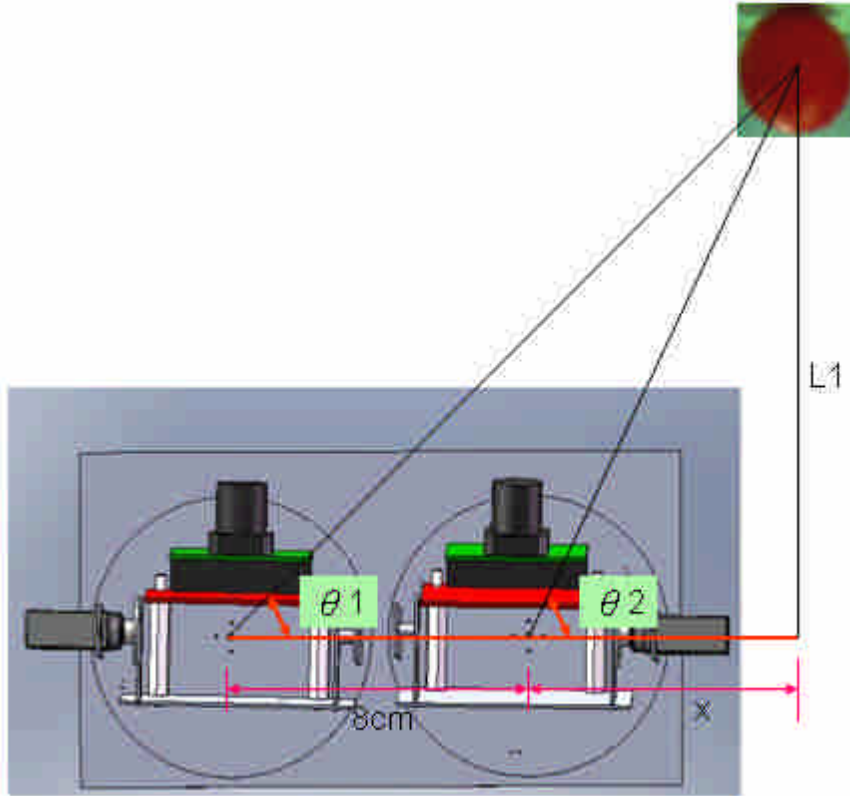


圖(3-4)最大搜尋範圍圖

伺服馬達工作範圍從 35° 到 135° 以及距離 25cm 處較佳，圖()黃色處為可搜尋之範圍，黃色以外的地方，則因為容易受到機器本身影響，誤差值較大，故不採用。

3-3 計算公式

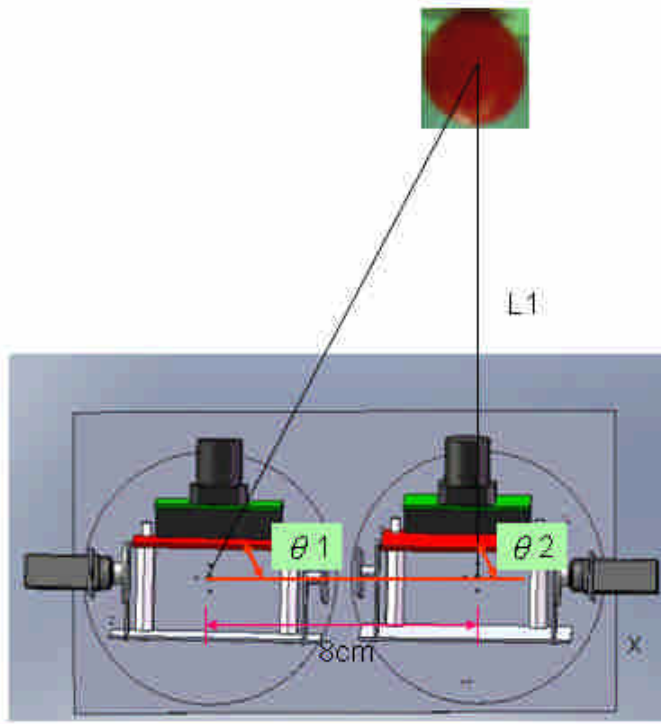
因本計劃中要計算距離，故將伺服馬達脈波數轉換成角度，利用三角測距法計算距離，但是當兩個角度其中一個為 90 度時，則需要特殊計算，計算公式如下所示：



圖(3-5)三角測距(1)

$$x = \frac{8 \tan \theta_1}{\tan \theta_2 - \tan \theta_1}$$

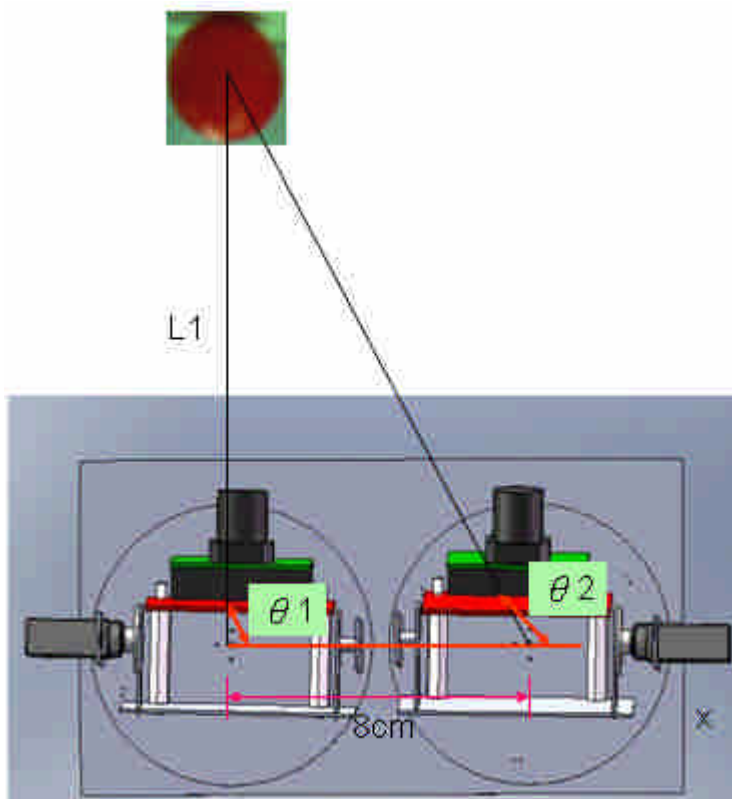
$$L1 = \frac{8 \tan \theta_1 \tan \theta_2}{\tan \theta_2 - \tan \theta_1}$$



圖(3-6)三角測距(2)

$$x = 0$$

$$L1 = 8 \tan \theta_1$$

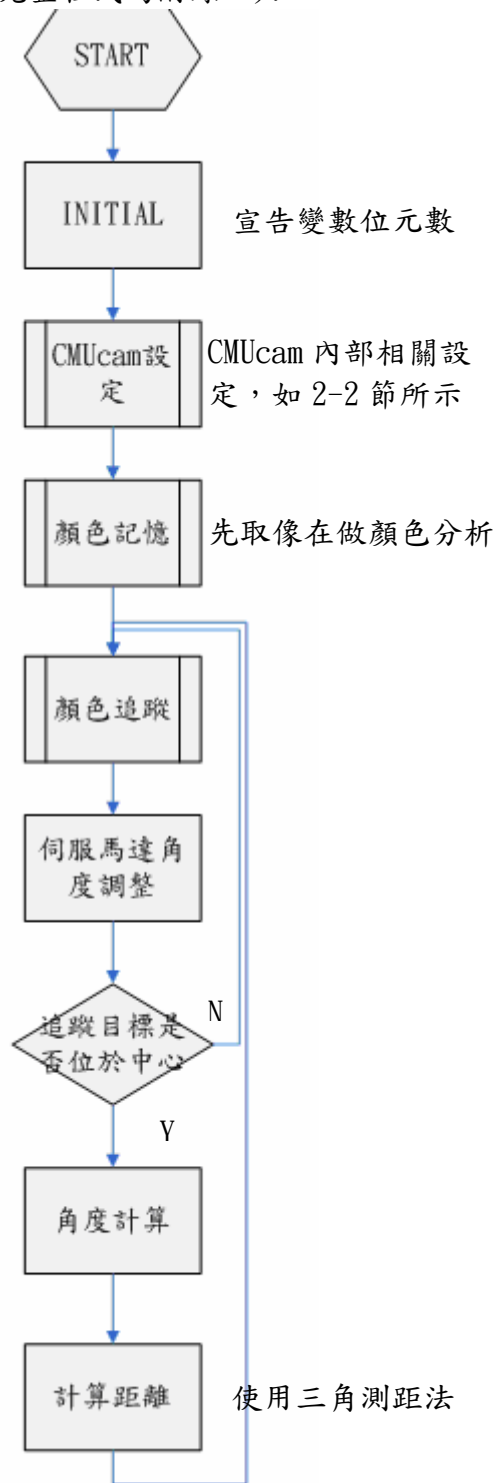


圖(3-7)三角測距(3)

$$L1 = 8 \tan \theta_2$$

3-4 系統流程

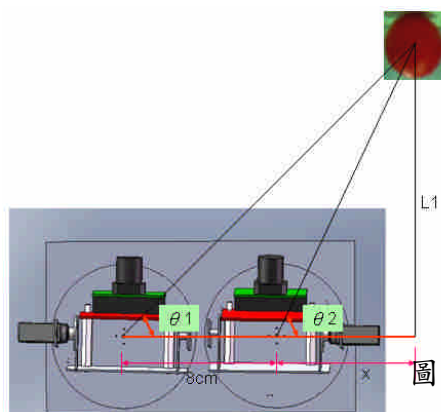
根據本計劃所需，微控制器控制 CMUcam 攝相機與伺服馬達，需要一套完整的程式，下面為一套完整的程式流程圖(完整程式為附錄一)：



圖(3-8)系統流程圖

第四章 實驗紀錄

(1)

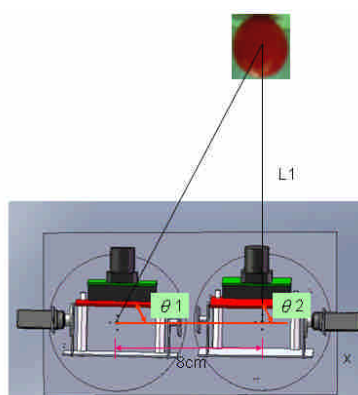


圖(4-1)實驗紀錄 1

實際距離 單位(cm)	計算距離(L1) 單位(cm)	實際距離 單位(cm)	計算距離(L1) 單位(cm)
10	11.5	23	22.1
13	13.9	25	26.8
15	15.5	28	29.5
18	18.1	30	28.7
20	19.6	35	38.5

表(2)實驗數據 1

(2)

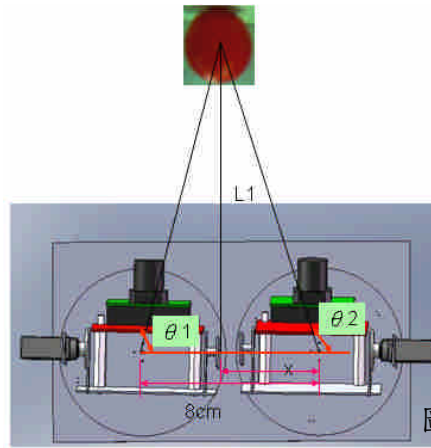


圖(4-2)實驗紀錄 2

實際距離 單位(cm)	計算距離(L1) 單位(cm)	實際距離 單位(cm)	計算距離(L1) 單位(cm)
10	116	23	23.5
13	14.5	25	24.8
15	16.5	28	30
18	18.6	30	33
20	19.8	35	37.5

表(3)實驗數據 2

(3)

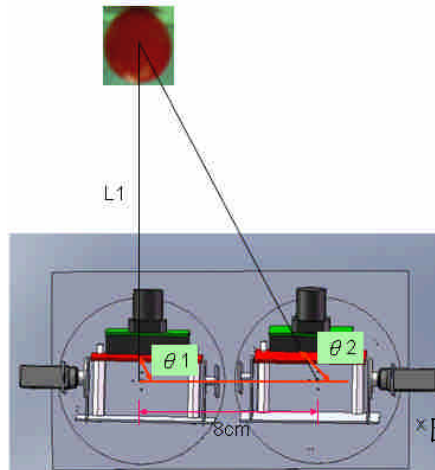


圖(4-3)實驗紀錄 3

實際距離 單位(cm)	計算距離(L1) 單位(cm)	實際距離 單位(cm)	計算距離(L1) 單位(cm)
10	12.7	23	23.1
13	13.5	25	25.5
15	15.2	28	29.3
18	17.8	30	31.7
20	19.8	35	39.5

表(4)實驗數據 3

(4)

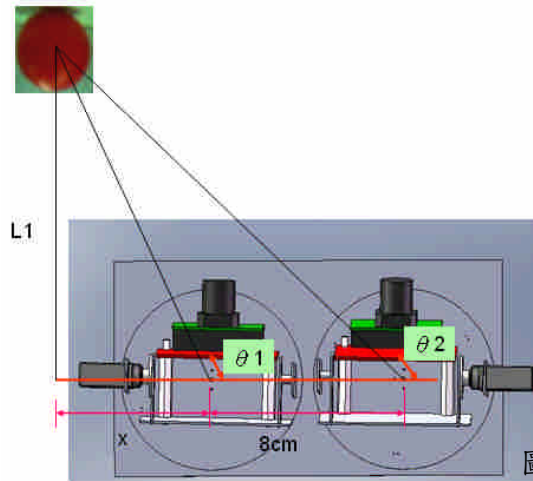


圖(4-4)實驗紀錄 4

實際距離 單位(cm)	計算距離(L1) 單位(cm)	實際距離 單位(cm)	計算距離(L1) 單位(cm)
10	11.5	23	23.6
13	14.4	25	25.9
15	15.8	28	29.5
18	18.3	30	31.5
20	20.5	35	38

表(5)實驗數據 4

(5)



圖(4-5)實驗紀錄 5

實際距離 單位(cm)	計算距離(L1) 單位(cm)	實際距離 單位(cm)	計算距離(L1) 單位(cm)
10	11	23	23.6
13	13.6	25	25.4
15	15.3	28	27.4
18	18.2	30	30.9
20	20	35	38.4

表(6)實驗數據 5

由上面表(2~6)所知，最佳計算距離在 15~25cm，平均誤差不超過 0.6cm，但是當有一角度為 90 度時，誤差值會上升到最大 1.5cm，這是因為在 90 度時，計算角度會只剩下一個，固誤差值會偏大；距離再 10~13cm 時，會因機械問題而產生較大的誤差值，至於 25cm 以上，則會因為物體所佔的像數過低，產生較大的誤差，而且，追蹤時所費的時間也較長。

第五章 結論

本計畫完成一具有追蹤動態物體能力之雙視覺系統，其中包括兩顆 CMUcam 攝像機、四顆伺服馬達控制並已以 BS2 用 BASIC 語言控制，利用 RS-232 將 BS2 計算結果回傳回電腦顯示。CMUcam 攝像機第一次取像時會做顏色記憶，可透過電腦顯示可以得知這次取像是否良好，接下來做顏色追蹤，可藉由 CMUcam 攝像機所回傳之圖像座標做伺服馬達角度調整，當物體位於圖像中心時，將伺服馬達脈波轉換成角度，利用這些角度做三角測距，計算出距離。由於雙視覺所需之硬體簡單，固適合應用在小型機器人身上，做動態物體追蹤。

參考文獻

- [1] 蔡哲民, 傅耀賢, 周家弘, 賴峻廷, ” 具標示與追蹤能力之植物影像教學平台”
- [2] 周銘章 “藍芽通訊足球機器員之製作”
- [3] Rowe, A. ; Rosenberg, C. ; Nourbakhsh, I. ” **A Second Generation Low Cost Embedded Color Vision System**” [Computer Vision and Pattern Recognition, 2005 IEEE Computer Society Conference on](#)
Volume 3, 20-26 June 2005 Page(s):136 - 136
- [4] 台灣科技大學電機系 郭重顯·長庚大學醫療機電工程研究所 黃方均, ” 影像伺服外科手術導航機器人開發” , 2007.11.06
- [5] CMUcam Vision Board
- [6] “機器人學與Boe-Bot 版本 2.2” 版權(c) 2007 Parallax Inc.
- [7] 季永炤, ” 多足式機器人概念式智慧行為控制系統之設計與測試分析”
2007/11/2

附錄(1)BS2 程式

```
' {$STAMP BS2}
' {$PBASIC 2.5}
' =====
' 宣告
' =====
A VAR Word      ' SERVO_RIGHT_DOWN
B VAR Word      ' SERVO_RIGHT_UP
C VAR Word      ' SERVO_LEFT_DOWN
D VAR Word      ' SERVO_LEFT_UP
CMU VAR Byte(10)
Q1 VAR Word
Q2 VAR Word
' COU VAR Byte  ' COUNT
X VAR Word
Y VAR Word
' =====
' TABLE
' =====
VOUT      DATA Word 256 , Word 265 ,Word 274 ,Word 284 ,Word 294
          DATA Word 305 , Word 316 ,Word 327 ,Word 339 ,Word 352
          DATA Word 365 , Word 379 ,Word 397 ,Word 409 ,Word 426
          DATA Word 443 , Word 461 ,Word 481 ,Word 502 ,Word 524
          DATA Word 548 , Word 574 ,Word 603 ,Word 633 ,Word 666
          DATA Word 703 , Word 743 ,Word 787 ,Word 837 ,Word 892
          DATA Word 955 , Word 1026 ,Word 1108 ,Word 1204 ,Word
1316
          DATA Word 1451 ,Word 1615 ,Word 1820 ,Word 2084 ,Word
2434
          DATA Word 2924 ,Word 3658 ,Word 4880 ,Word 7321 ,Word
14627
' =====
' INITIAL
' =====
A = 800
B = 850
C = 600
D = 350
FOR Q1 = 1 TO 20
PULSOUT 15, A
PAUSE 10
PULSOUT 14, B
PAUSE 10
PULSOUT 13, C
PAUSE 10
PULSOUT 12, D
```

```

    PAUSE 10
    NEXT
' =====
' CMUcam SET
' =====
' Pause 1 second for CMUcam1 startup
    PAUSE 1000
START:
' Send "reset" to sync CMUcam1 and Stamp
    SEROUT 7, 84, ["RS", CR]
    PAUSE 10
    SEROUT 6, 84, ["RS", CR]
    PAUSE 1000
' Green LED on
    SEROUT 7, 84, ["L1 1", CR]
    PAUSE 10
    SEROUT 6, 84, ["L1 1", CR]
    PAUSE 100
' Turn on auto adjust for 5 seconds
    SEROUT 7, 84, ["CR 18 44", CR]
    SEROUT 6, 84, ["CR 18 44", CR]
    PAUSE 100
' Pause 5 seconds for CMUcam1 to auto adjust to lighting conditions
    PAUSE 5000
' Turn off auto adjust
    SEROUT 7, 84, ["CR 18 44 19 32", CR]
    SEROUT 6, 84, ["CR 18 44 19 32", CR]
    PAUSE 100
' SET WINDOW 80*143
' Green LED auto mode
    SEROUT 7, 84, ["L1 2", CR]
    SEROUT 6, 84, ["L1 2", CR]
    PAUSE 100
' Give user time to place color target close in front of camera
    PAUSE 5000
' Send command - Set poll mode - only sends one return packet -
' of data after each command - reduces data flow
    SEROUT 7, 84, ["PM 1", CR]
    SEROUT 6, 84, ["PM 1", CR]
    PAUSE 100
' Send command - Set raw data mode - also suppress Ack:/Nak: to -
' further reduce serial data
    SEROUT 7, 84, ["RM 3", CR]
    SEROUT 6, 84, ["RM 3", CR]
    PAUSE 100
' Track Window command looks at the center of CMUcam1 image -
' grabs the color information and sends to the Track Color function
' Send command - Track window

```

```

SEROUT 7, 84, ["TW",CR]
' Display the S Statistics packet from TW command
SERIN 9, 84, 5000, START, [STR CMU\8]
' DEBUG "Red Mean ",DEC CMU(2) ,CR
' DEBUG "Green Mean ",DEC CMU(3) ,CR
' DEBUG "Blue Mean ",DEC CMU(4) ,CR
' DEBUG "Red Deviation ",DEC CMU(5) ,CR
' DEBUG "Green Deviation ",DEC CMU(6) ,CR
' DEBUG "Blue Deviation ",DEC CMU(7) ,CR
' DEBUG " ", CR
SEROUT 6, 84, ["TW",CR]
SERIN 8, 84, 5000, START, [STR CMU\8]
' DEBUG "Red Mean ",DEC CMU(2) ,CR
' DEBUG "Green Mean ",DEC CMU(3) ,CR
' DEBUG "Blue Mean ",DEC CMU(4) ,CR
' DEBUG "Red Deviation ",DEC CMU(5) ,CR
' DEBUG "Green Deviation ",DEC CMU(6) ,CR
' DEBUG "Blue Deviation ",DEC CMU(7) ,CR
' DEBUG " ", CR
',
=====
' TRACK COLOR AND SERVO 參數
',
=====
TC_R:
Y = 0
PAUSE 500
SEROUT 6, 84, ["TC 72 140 9 61 0 59" ,CR]
SERIN 8, 84, 5000, ERR, [STR CMU\10]
PAUSE 10
IF (CMU(2)) > = 65 THEN R_LEFT1
IF (CMU(2)) > = 50 THEN R_LEFT2
IF (CMU(2)) > = 43 THEN R_LEFT3
IF (CMU(2)) < = 37 THEN R_RIGHT1
IF (CMU(2)) < = 30 THEN R_RIGHT2
IF (CMU(2)) < = 15 THEN R_RIGHT3
Y = Y + 1
GOTO TC_R1
R_LEFT1:
A = A + 80
GOTO TC_R1
R_LEFT2:
A = A + 40
GOTO TC_R1
R_LEFT3:
A = A + 20
GOTO TC_R1
R_RIGHT1:
A = A - 20
GOTO TC_R1

```



```

R_RIGHT2:
    A = A - 40
    GOTO TC_R1
R_RIGHT3:
    A = A - 80
TC_R1:
    DEBUG "CMU_R X ",DEC CMU(2) ,CR
    DEBUG "CMU_R Y ",DEC CMU(3) ,CR
    DEBUG "Confidence ",DEC CMU(9) ,CR
    DEBUG "SERVO_R " ,DEC A ,CR
    DEBUG " ", CR
TC_L:
    SEROUT 7, 84, ["TC 72 140 9 61 0 59" ,CR]
    SERIN 9, 84, 5000,ERR1, [STR CMU\10]
    PAUSE 10
    IF (CMU(9)) < 50 THEN L_RIGHT2
    IF (CMU(2)) > = 65 THEN L_LEFT1
    IF (CMU(2)) > = 50 THEN L_LEFT2
    IF (CMU(2)) > = 43 THEN L_LEFT3
    IF (CMU(2)) < = 36 THEN L_RIGHT1
    IF (CMU(2)) < = 30 THEN L_RIGHT2
    IF (CMU(2)) < = 15 THEN L_RIGHT3
    Y = Y + 1
    GOTO TC_L1
L_LEFT1:
    C = C + 80
    GOTO TC_L1
L_LEFT2:
    C = C + 40
    GOTO TC_L1
L_LEFT3:
    C = C + 20
    GOTO TC_L1
L_RIGHT1:
    C = C - 20
    GOTO TC_L1
L_RIGHT2:
    C = C - 40
    GOTO TC_L1
L_RIGHT3:
    C = C - 80
TC_L1:
    DEBUG "CMU_L X ",DEC CMU(2) ,CR
    DEBUG "CMU_L Y ",DEC CMU(3) ,CR
    DEBUG "Confidence ",DEC CMU(9) ,CR
    DEBUG "SERVO_L ",DEC C, CR
    DEBUG " ", CR
    GOSUB SERVO

```

```

    IF Y > 1 THEN MATH
    IF Y <= 1 THEN TC_R
ERR:
    DEBUG "CMU_R ERROR", CR
    GOTO TC_L
ERR1:
    DEBUG "CMU_L ERROR", CR
    GOTO TC_R
SERVO:
    FOR Q1 = 1 TO 20
    PULSOUT 15, A
    PAUSE 10
    PULSOUT 13, C
    PAUSE 10
    NEXT
    PAUSE 100
    RETURN
MATH:
    DEBUG "A = " , DEC A, CR
    IF A >= 550 AND A < 800 THEN Q1_45_90
    IF A >= 800 AND A < 1000 THEN Q1_90_135
    IF A < 550 THEN Q1_45
    IF A > 1000 THEN Q1_135
MATH1:
    DEBUG "C = " , DEC C , CR
    IF C >= 450 AND C < 700 THEN Q2_45_90
    IF C >= 700 AND C < 900 THEN Q2_90_135
    IF C < 450 THEN Q2_45
    IF C > 900 THEN Q2_135
Q1_45:
    ' DEBUG "SERVO_R < 45 度", CR
    GOTO TC_R
Q1_135:
    ' DEBUG "SERVO_R > 135 度", CR
    GOTO TC_R
Q1_45_90:
    Q1 = (A - 550)*45 /250 +45
    DEBUG "Q1 " , DEC Q1, CR
    IF Q1 = 90 THEN Q1_90
    Q1 = (Q1 - 45) *2
    ' FOR CMU(1) = 0 TO Q1 STEP 2
    READ ( VOUT + Q1) , Word X
    ' NEXT
    Q1 = X
    DEBUG "TANQ1= " , DEC Q1, CR
    GOTO MATH1
Q1_90_135:
    Q1 = (A - 800) *45 /200 +90

```

```

DEBUG "Q1 " , DEC Q1, CR
IF Q1 = 90 THEN Q1_90
Q1 = (45-Q1 +90 ) *2
' FOR CMU(1) = 45 TO Q1 STEP 2
READ (VOUT + Q1), Word X
' NEXT
Q1 = 65535 - X
CMU(1) = 1
DEBUG "TANQ1= " , DEC Q1, CR
GOTO MATH1
Q1_90:
Q1 = 0
GOTO MATH1
Q2_45:
' DEBUG "SERVO_L < 45 度", CR
GOTO TC_R
Q2_135:
' DEBUG "SERVO_L > 135 度", CR
GOTO TC_R
Q2_45_90:
Q2 = (C - 450) *45 /250 +45
DEBUG "Q2 " , DEC Q2, CR
IF Q2 = 90 THEN Q2_90
Q2 = (Q2 -45 ) *2
READ (VOUT + Q2) , Word X
' NEXT
Q2 = X
DEBUG "TANQ2= " , DEC Q2, CR
GOTO MATH2
Q2_90_135:
Q2 = (C -700) *45 /200 +90
DEBUG "Q2 " , DEC Q2, CR
IF Q2 = 90 THEN Q2_90
Q2 =( 45- Q2 +90 ) *2
' FOR CMU(2) = 45 TO Q2 STEP -1
READ (VOUT + Q2) , Word X
' NEXT
Q2 = 65535 - X
DEBUG "TANQ2= " , DEC Q2, CR
CMU(1) = CMU(1) + 1
GOTO MATH2
Q2_90:
Q2 = 1
GOTO MATH2
MATH2:
IF CMU(1) = 1 THEN MATH3
X = Q1 - Q2
X = 7 * Q1 /X

```

```
Y = 8 * Q2 / 256  
DEBUG "L= ", DEC Y, CR  
GOTO TC_R
```

MATH3:

```
X = Q1 + Q2  
X = 8 * Q1 /X  
Y = X * Q2 /256  
DEBUG "L= " , DEC Y , CR  
GOTO TC_R
```

END