

2019/1/19

1071 單晶片系統實作

期末綜合實作

目錄

圖目錄.....	2
一、 題目:機械手臂.....	4
1. 說明: 利用六個伺服馬達製作六軸機械手臂，並實作其控制程式.....	4
2. 程式碼:.....	4
3. 靜態展示:.....	8
4. 動態展示:.....	11
Google Cloud.....	11
二、 學習心得.....	12

圖目錄

Figure 1: 機械手臂物件標頭檔 1.....	4
Figure 2: 機械手臂標頭檔 2.....	4
Figure 3: 機械手臂物件定義檔 1.....	5
Figure 4: 機械手臂定義檔 2.....	5
Figure 5: 機械手臂定義檔 3.....	6
Figure 6: 主程式 1.....	6
Figure 7: 主程式 2.....	7
Figure 8: 實作成品 1.....	8
Figure 9: 實作成品 2.....	9
Figure 10: 實作成品 3.....	9
Figure 11: 實作成品 4.....	10

一、 題目

題目：設計機械手臂

設計一個可以進行操作的機械手臂

作業報告內容：

- (1)封面
- (2)目錄(含報告目錄、圖目錄，要利用 Word 自動產生)
- (3)題目
- (4)程式碼(要有註解)
- (5)靜態展示(成果照片)
- (6)動態展示(拍成影片及配音說明)
- (7)學習心得

一、 題目：機械手臂

1. 說明：

利用六個伺服馬達製作六軸機械手臂，並實作其控制程式

2. 程式碼：

```
8  #ifndef ROBOTARM_H
9  #define ROBOTARM_H
10
11 #include <Arduino.h>
12 #include <Servo.h>
13 class RobotArm
14 {
15     public:
16         // Constructor
17         RobotArm(const int *pins, int axesm, const int *defaultDegrees);
18         //-----
19
20         // Deconstructor
21         ~RobotArm();
22         //-----
23
24         // Robot arm static control
25         void staticControl(const int *i_degrees, int axes);
26         //-----
27
28         // Robot arm dynamic test
29         void dynamicTest();
30         //-----
```

Figure 1: 機械手臂物件標頭檔 1

```
32     private:
33         // Functions:
34         //-----
35
36         // Data:
37
38         int axes = 6;
39         int pos = 0;
40         bool flag = true;
41
42         // create servo object to control a servo
43         Servo myservo1, myservo2, myservo3, myservo4, myservo5, myservo6;
44         const Servo m_servos[6] = {myservo1, myservo2, myservo3, myservo4, myservo5, myservo6};
45
46         int m_defaultDegree[6];
47         int bufferDegrees[6];
48         //-----
49     };
50 #endif // ROBOTARM_H
```

Figure 2: 機械手臂標頭檔 2

```

1  #include "RobotArm.h"
2
3  // Constructor
4  RobotArm::RobotArm(const int *pins, int axes, const int *defaultDegrees) : axes(axes)
5  {
6      Serial.begin(9600);
7      // m_servos = new Servo[this->axes];
8      // m_defaultDegree = new int[this->axes];
9
10     for (int i = 0; i < this->axes; ++i)
11     {
12         // Attaches the servo on pin to the servo objects
13         m_servos[i].attach(pins[i]);
14         // =====
15
16         // Setting default degrees
17         m_defaultDegree[i] = defaultDegrees[i];
18         // =====
19
20         // Normalize Servos
21         m_servos[i].write(m_defaultDegree[i]);
22         // =====
23     }
24 }
25 //-----

```

Figure 3: 機械手臂物件定義檔 1

```

27 // Deconstructor
28 RobotArm::~RobotArm()
29 {
30 }
31 //-----
32
33 // Robot arm static control
34 void RobotArm::staticControl(const int *i_degrees, int axes)
35 {
36     int i = 0;
37     for (i = 0; i < (this->axes - 1); ++i)
38     {
39         m_servos[i].write((i_degrees[i] <= 180 && i_degrees[i] >= 0) ? i_degrees[i] : m_defaultDegree[i]);
40     }
41
42     m_servos[i].write((i_degrees[i] <= 180 && i_degrees[i] >= 120) ? i_degrees[i] : m_defaultDegree[i]);
43     delay(15); // waits 15ms for the servo to reach the position
44 }
45 //-----

```

Figure 4: 機械手臂定義檔 2

```

47 // Robot arm dynamic control
48 void RobotArm::dynamicTest()
49 {
50     bufferDegrees[0] = pos;
51     staticControl(bufferDegrees, 6);
52     if(flag){
53         ++pos;
54     }else{
55         --pos;
56     }
57     delay(15);
58
59     if(pos == 180 || pos == 0){
60         flag = !flag;
61     }
62 }
63 //-----
64

```

Figure 5: 機械手臂定義檔 3

```

10 #include "RobotArm.h"
11
12 // Declare variables:
13
14 // Pins of servos:
15 int servos[6] = {2, 3, 4, 5, 6, 7};
16 // =====
17
18 // Pins of buttons:
19 int test_switch = A0; // pin of test button
20 // =====
21
22 const int defaultDegrees[6] = {90, 20, 20, 150, 90, 180}; // Robot arm default degrees
23 int bufferDegrees[6] = {90, 20, 20, 150, 90, 180}; // Robot arm degrees buffer
24 RobotArm *robotArm; // Robot arm object
25 //-----
26
27 // Forward declarations:
28 long mapping(int pin);
29 void controller();
30 void display_A();
31 //-----

```

Figure 6: 主程式 1

```

33 void setup()
34 {
35     robotArm = new RobotArm(servos, 6, defaultDegrees);
36     Serial.begin(9600); // for debug
37
38     // setting pins:
39     for (int i = 0; i < 6; ++i)
40     {
41         pinMode(servos[i], OUTPUT);
42     }
43     pinMode(test_switch, INPUT);
44     // =====
45 }
46
47 void loop()
48 {
49     if (analogRead(test_switch))
50     {
51         robotArm->staticControl(defaultDegrees, 6);
52         Serial.println("A");
53     }
54     else
55     {
56         robotArm->dynamicTest();
57         Serial.println("B");
58     }
59 }

```

Figure 7: 主程式 2

3. 靜態展示:



Figure 8:實作成品 1

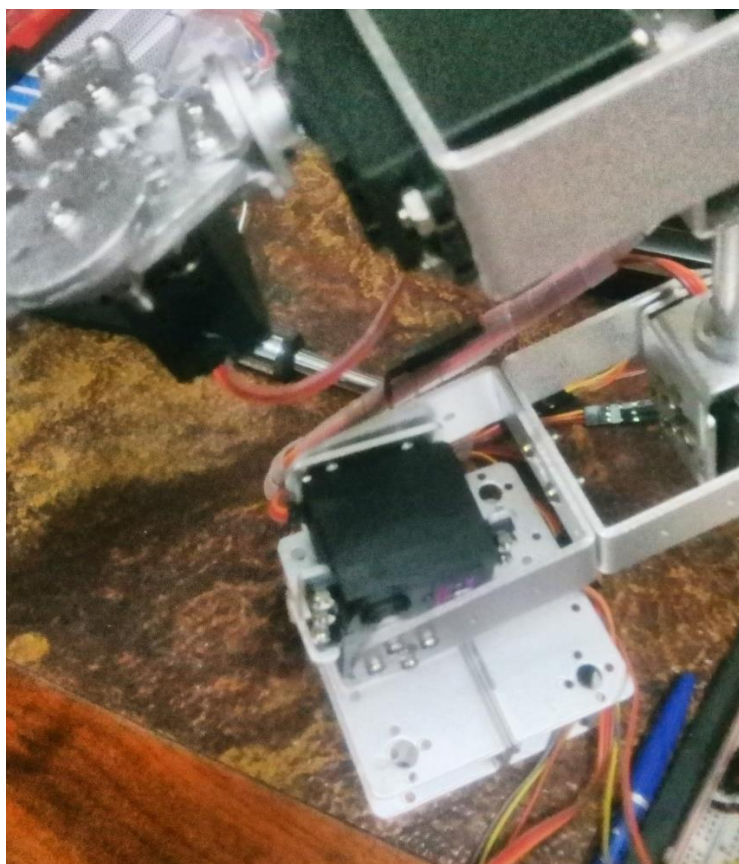


Figure 9: 實作成品 2

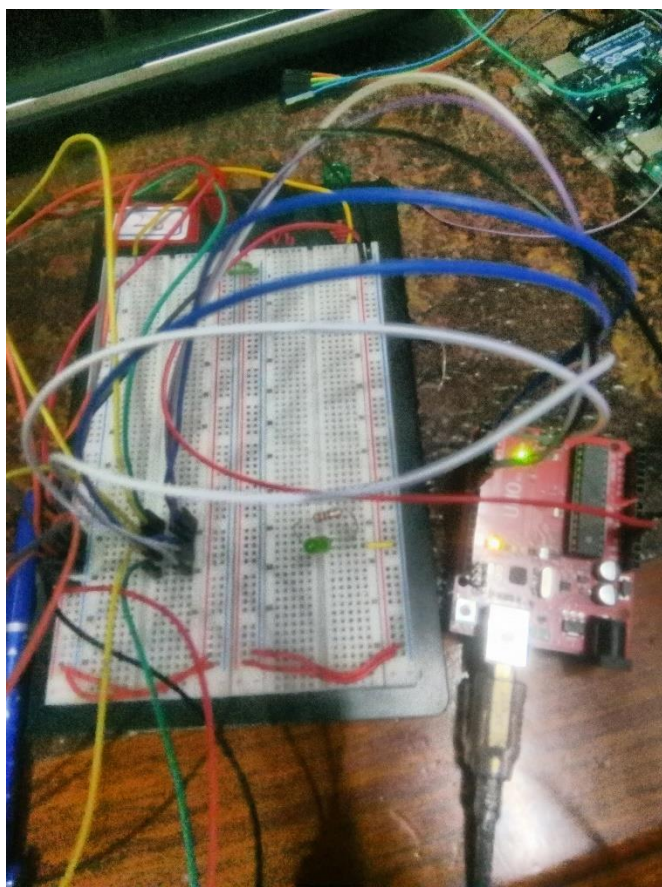


Figure 10: 實作成品 3



Figure 11: 實作成品 4

4. 動態展示：

[Google Cloud](#)

二、學習心得

這次作業原本是要用 wifi 來進行更多變化的操作，但因機械手臂的伺服馬達要求的電流較高且實作過程中電壓調整有誤造成了 ESP 部件損所以才臨時改成單一晶片操作的方式，且因為這件事讓我未來對電壓操作會更加小心。