

基于 Arduino 的硬币纸币双向兑换装置

(陆宽 3130100659 计科 1304 班 18868107501 1245554637@qq.com)

1. 概述：这是一个基于 Arduino 的硬币纸币兑换双向装置，当初做这个的灵感一方面

是在公交车上发现只有一张 5 元纸币而没有硬币无法坐车时涌现的，有了这样一个装置，如果能应用到各个公交车站上，就可以在人们因为只有纸币没有硬币无法坐公交车时提供便利。另一方面，我发现同学们经常钱包里有许多零钱，这不利于携带而且容易丢失硬币。所以基于人们在这两方面的需求。做了这样一个既能用硬币兑换纸币又能用纸币兑换硬币的装置。在暑假的立项报告中，我本来以为只能做出硬币兑换纸币的装置，最后经过实践，用木头，锯子，502 胶水和单片机等外围器件把一整套双向兑换的装置做了出来。它能够实现的功能如下：

1.1 硬币兑换纸币

支持的硬币为 1 元硬币和 5 角硬币，当投入硬币总金额为 5 的倍数时，兑换处相应张数的 5 元纸币

1.2 纸币兑换硬币

由于自制木质外壳大小的限制，只支持兑换 5 元纸币，塞入 5 元纸币时，兑换出 5 个一元硬币

1.3 实时信息显示

投入硬币或纸币时，投入的金额会在 LCD1602 显示屏上显示，当投入硬币不足 5 的倍数时，会提醒投入更多。兑换时会显示兑换中请等待

2. 装置组成：该装置一共分为六个模块：分别为硬币识别分类模块，硬币存储模块，硬币退出模块，纸币进入模块，纸币退出模块，LCD 显示模块。各个模块除了器件都是由我手工锯的木板构成的。

3. 器材介绍：该装置用到的器件如下

器件名称	器件数量	器件作用
Arduino Mega 2560 板	1 块	处理、控制所有器件
红外发射接收对	3 对	全部充当阻断传感器 一对用于识别硬币是否投入 一对用于识别纸币是否投入 一对用于计算硬币通过阻断传感器的时间
9g 舵机	6 个	四个用于硬币分类识别模块 一个用于硬币退出模块 一个用于纸币退出模块
步进电机	2 个	两个步进电机上下排列作为滚轮，卷入或者卷出纸币
LCD1602 显示屏	1 个	用于显示当前投入金额总数

		以及各种提示
电位器	1 个	用于调整 LCD 显示屏的显示亮度
10k 电阻	3 个	用于红外接收管
330Ω电阻	3 个	用于红外发射管，防止烧坏
12000mAh 移动电源	1 个	用于装置的供电，通过 USB 线，提供 5V 1A 的电流
5 号电池	2 节	为装置内红外发射管提供 3V 的电压
面包线	若干	作为电路连接线
面包板	一块	用于电路连接
5mm 桦木板	6 块	作为装置的外壳
3mm 桦木板	若干	作为装置内的各个支架及轨道或收集器
3*3mm 软木条	若干	延长舵机摆臂/轨道支架
1 元硬币直径 PVC 管	3 个	1 个用于存储 1 元硬币 2 个作为步进电机上的滚筒，用于卷入卷出硬币
5 角硬币直径 PVC 管	1 个	用于存储 5 角硬币

所有器件一共用去 6 个 (PWM 端口) +20 个(Digital 端口)+3 个 (Analog 端口) 共 29 个端口。

4. 各模块介绍：

4.1 硬币识别分类模块：

这是一个由 4 个舵机和 2 个红外接收对，以及木制硬币轨道构成的模块

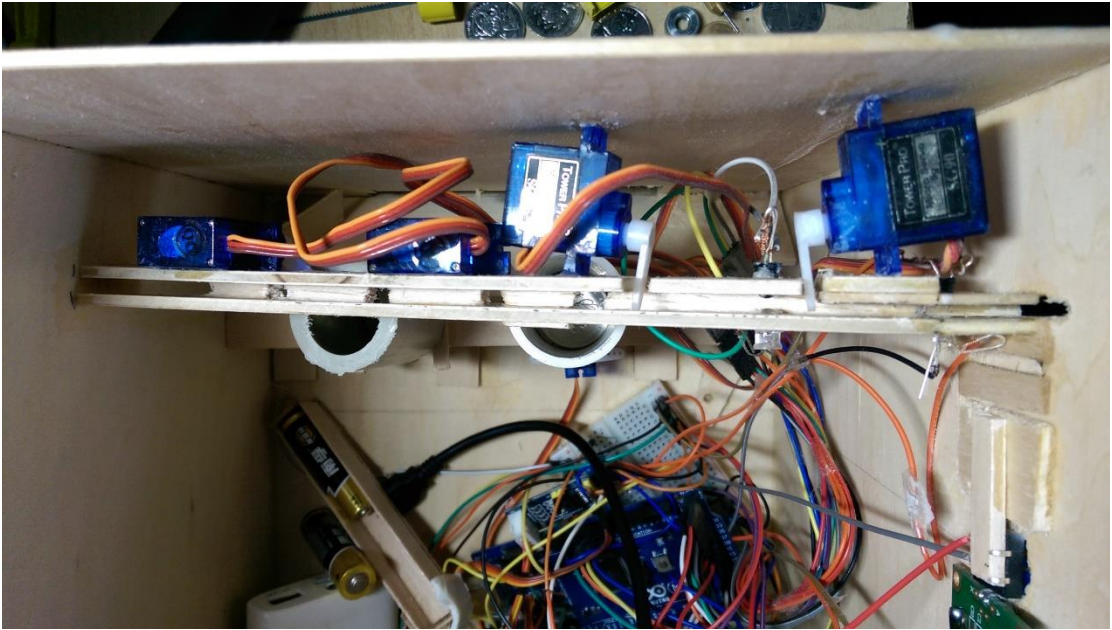


图 4.1 硬币分类识别模块

4.1.1 感应硬币投入：

首先，图中右边第一个顶端相对的红外接收对的作用是感应是否有硬币通过投入口投入，如果有硬币投入，红外接受对作为阻断传感器会识别到(未有东西阻断时 A1 端口读数为 10-20，当有硬币阻断时，在短时间内上升至 900 以上)，这个时候会打开第一个充当闸机的舵机，这个舵机的存在是为了保证每个硬币通过下一步的硬币大小识别轨道时的初速度相同，因为经过实验，没有这道闸机时识别会出现较大误差，在下一部分有详细叙述

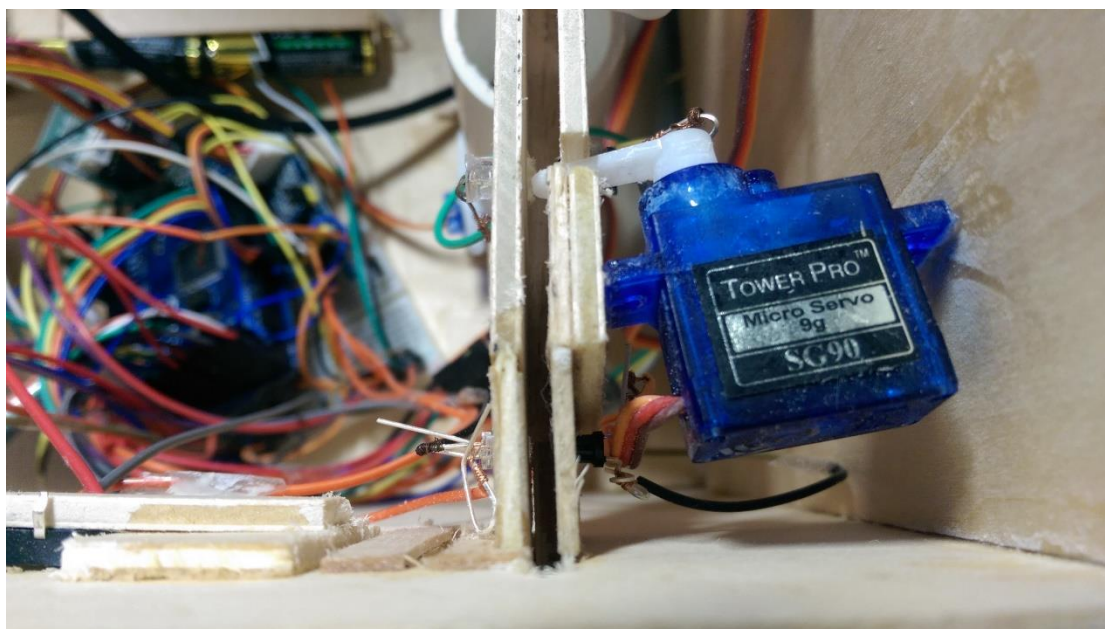


图 4.1.1 硬币投入感应

4.1.2 硬币大小识别

硬币的识别核心部件是一个红外接收对（如图 4.1.2），当硬币从上方滑下时会挡住从红外发射管发射出的红外线使红外接收管（接入端口为 A0）无法接收到，当硬币滑过这个接收对时，红外接收管又能接收到红外发射管发出的红外线，通过统计红外线被阻挡的时间，就可以计算出硬币的直径，因为 5 角和 1 元硬币直径不同，所以能够分别出 5 角，1 元，或者两种都不是的硬币比如 1 角，这个部分的控制变量是通过前文提到的第一个舵机充当的闸机实现的，硬币滑下的初速度相同，相同硬币遮挡红外线的时间才能相同。经过试验，1 元硬币的通过时间大约为 80ms 到 90ms 而 5 角硬币的通过时间为 60ms 到 75ms,如此一来便能识别出不同硬币。

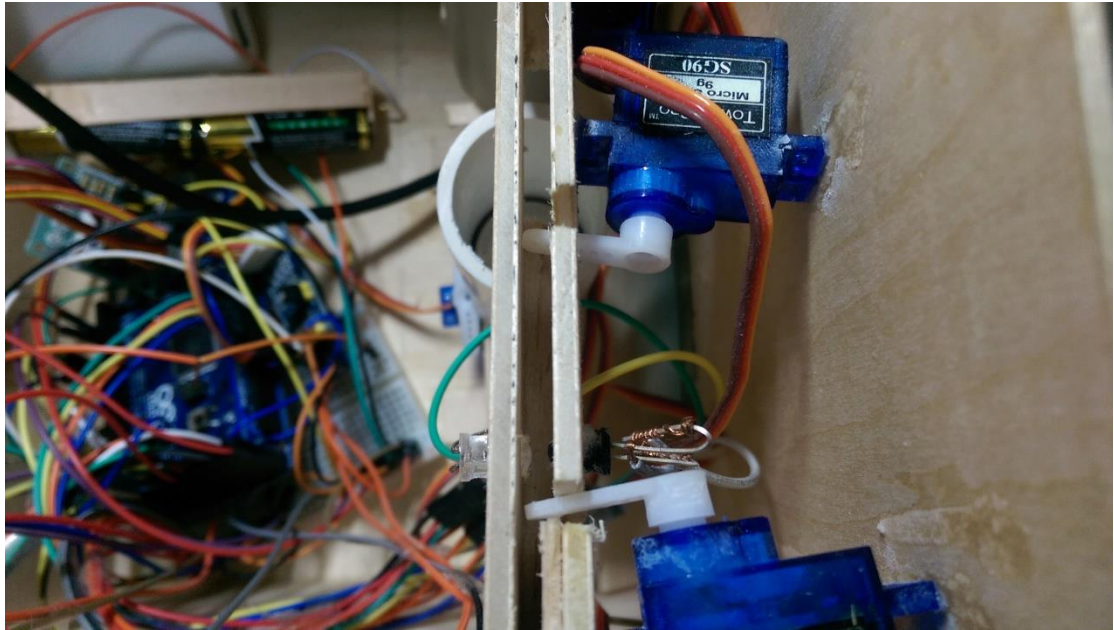


图 4.1.2 硬币大小识别

4.1.3 硬币分类

当硬币通过上述两个装置后将为被分类，这个是通过如图 3 个舵机实现的，第一个舵机让硬币在经过识别后停止，等待后两个舵机完成动作后再打开，这一步是因为两个存储桶上方的舵机完成旋转需要时间。当识别为 1 元硬币时，第一个舵机就会旋转 60 度，从而硬币轨道的一部分就会被移开，从而形成一道缺口，硬币滚落时就会掉入 1 元硬币的收集桶，一个一个的叠在桶中，同样道理当识别为 5 角纸币时，第二个舵机就会旋转一定角度同时第一个舵机还在初始位置，因此 5 角硬币就会掉入 5 角硬币收集桶。如果硬币的通过时间不符合 1 元或者 5 角的通过时间，那么两个分类舵机都不会转动，硬币会一直滚到末端掉落，并滑过斜坡从装置右壁的退出口中退出。

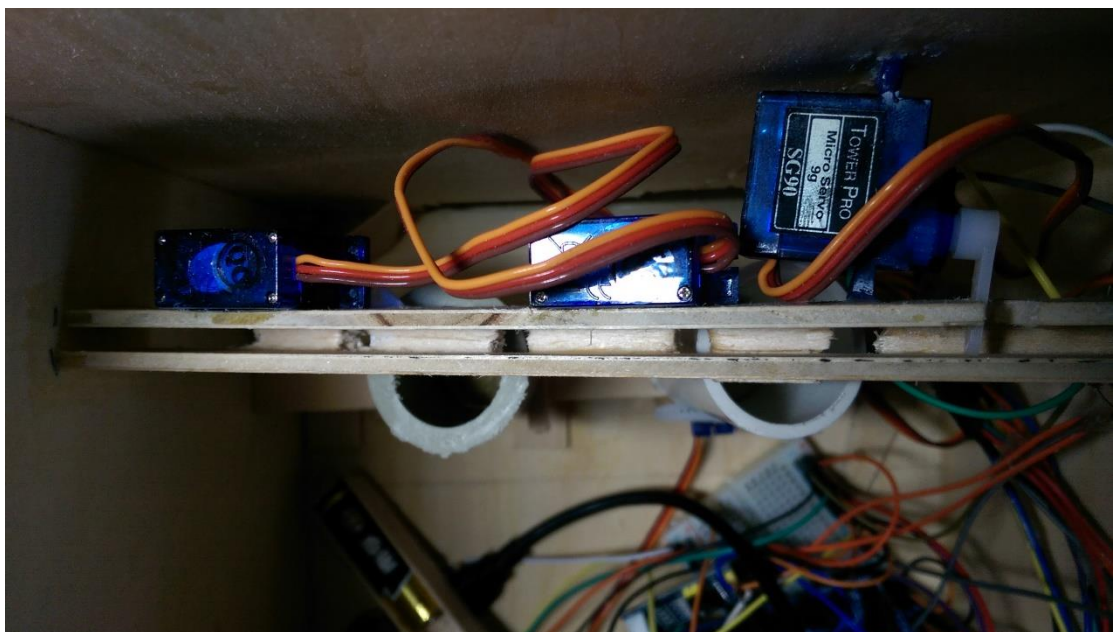


图 4.1.3 硬币分类

4.2 硬币存储模块

这个模块由 2 个分别与 5 角，1 元直径相同的 PVC 管构成，这样设计的原因是想当硬币掉落时，硬币能够整齐的一个接一个得堆叠在下面硬币的上面，这为接下来要叙述的硬币退出模块提供了便利，经过实验，白色的 1 元硬币存储桶能存储 30 个一元硬币，而灰色的 2 角硬币存储桶能够存储 30 个 5 角硬币，整个装置能存储总共金额为 45 元的硬币

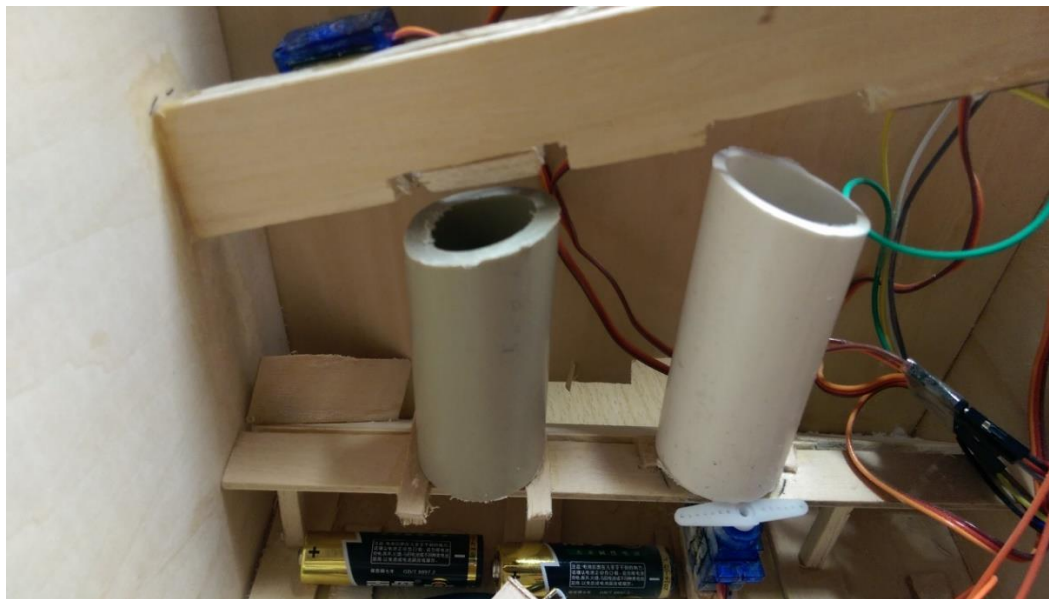


图 4.2 硬币存储模块

4.3 硬币退出模块

如图 4.3，在 1 元硬币存储装置的低端和支撑木板之间有一道与硬币厚度一样的缝隙，每次最低端硬币的两侧会暴露在外面。硬币退出装置通过一个舵机，当要退出硬币时，舵机会转动一定角度将最低端的硬币顶出硬币存储装置，这样就可以实现硬币的退出。当投入 5 元纸币时，舵机就会摆动 5 次，从而退出 5 个硬币。

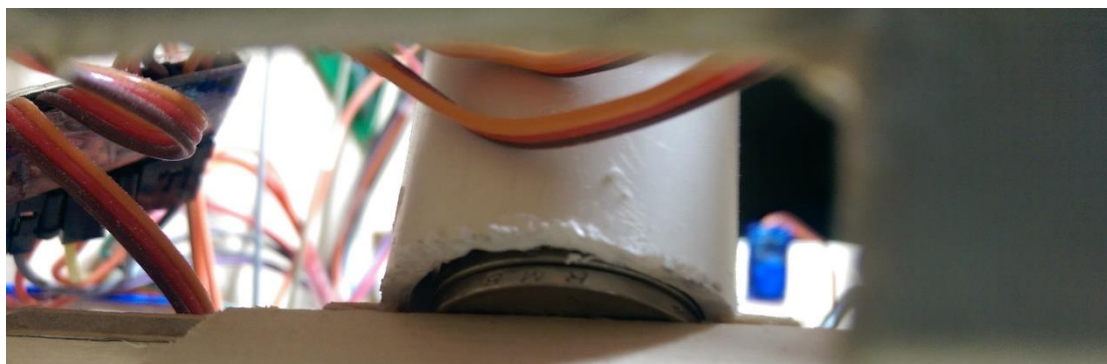


图 4.3 硬币退出模块



图 4.3 硬币退出模块

4.4 纸币进入模块

这个模块由 2 个步进电机和其连接的 PVC 管，一个红外接受对构成

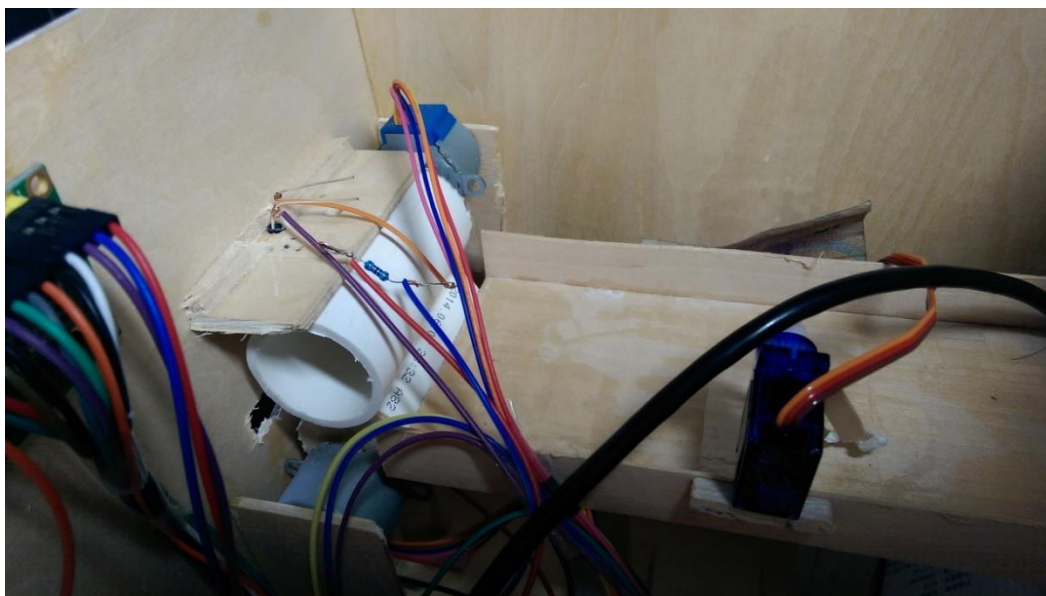


图 4.4 纸币进入模块

4.4.1 识别纸币投入

在纸币的投入口有一个充当阻断传感器的红外接受对（接入端口为 A2），当有纸币投入时红外线会被阻断（由于纸币能够透光，通过实验，被纸币阻断时 A2 端口接收值为 900-1000）所以，这个时候上下排列的两个步进电机就会带动 PVC 管作方向相反的旋转，在两个 PVC 管的缝隙中，纸币就会受到滚动 PVC 管的摩擦力被卷进装置



图 4.4.1 识别纸币投入

4.4.2 纸币存储

被卷入的纸币存储在一个用木头制作的小盒内，一张叠着一张排列，这个小盒是有几根软木棒支撑起来的，外形像一座桥梁。经过测试纸币存储装置能够存储 10 张以上的 5 元纸币。由于装置大小限制，只支持 5 元纸币的兑换和退出



图 4.4.2 纸币存储

4.5 纸币退出模块

在纸币的存储架上有一个舵机，舵机臂的末端有一小块泡沫胶。当硬币投入完毕按下按钮即开始兑换纸币。这时候舵机就会向纸币退出方向旋转一定角度(由 120 度转到 180 度)，等待舵机动作完成后，末端的泡沫胶将纸币向前挪动一段距离使其恰好能被 PVC 管组成的滚轮卷出，这时候步进电机反向转动将纸币通过纸币退出口退出装置。

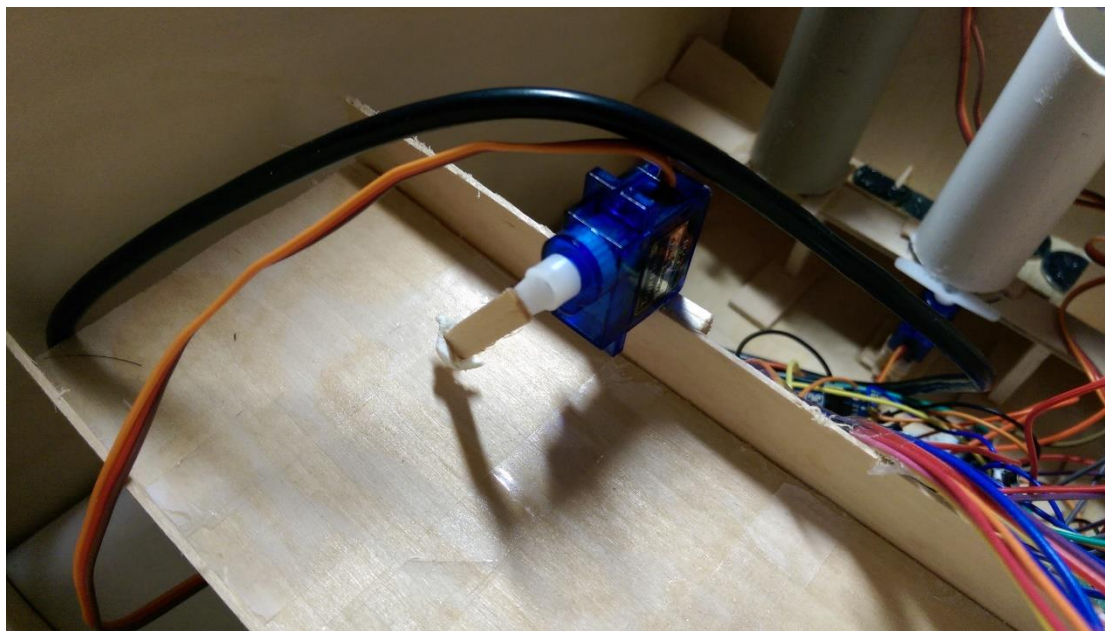


图 4.5 纸币退出模块

4.6 显示模块

4.6.1 投入金额指示

在默认状态下，显示屏上显示的内容为 “Insert coins:(当前投入的总金额)”



图 4.6.1 投入金额指示

4.6.2 金额不足指示

由于硬币兑换纸币必须是 5 元纸币的整数倍数，所以当硬币投入金额不为 5 的整数倍时，显示屏会提示 “Please Insert more coins” 请投入更多的硬币



图 4.6.2 金额不足指示

4.6.3 兑换中指示

当投入硬币或纸币金额符合要求时，按下蓝色按钮，显示屏显示 “Exchanging Please wait” 正在兑换，请等待的字样



图 4.6.3 兑换中指示

5. 展望

由于暑假时间的仓促，一些更加方便的功能并没有实现，在今后的时间我希望能够继续完善我的硬币纸币双向兑换装置。首先，硬币通过红外接受对识别大小，这并不具有真假硬币的识别功能，如果有一枚大小和 1 元硬币和 5 角硬币一样的假币那么系统一样会认作 1 元或 5 角硬币，我希望能够把硬币真假识别的模块做出来。另一方面，纸币也存在相同的问题，仅仅实现了纸币的吞吐没有真假的识别，而且由于木质外壳的大小限制，并没有 10 元纸币的输入，我会做一个更加大的外壳实现 10 元纸币的吞吐。最后，是硬币的退出模块，用舵机臂有些时候并不能将硬币完全顶出来，需要用手碰一碰才会掉落，这是因为硬币叠得很多的时候低端硬币会受到很大的摩擦力所以不能顺利掉落，我还在思考如何解决这个的方案。

我还想说下这个装置可能的应用，在一开始的介绍中我就指出了这个装置是为了满足人们日常生活中的两个金钱兑换的需求而设计的，所以我希望制作这个装置的理念和出发点能够被重视并成为一种便民的公共设施，如果成本能够压低那么政府在各个公交站上

可以安装并使用。另一方面，在每个寝室的楼下，同学们回来时往往口袋中有很多硬币（购物或者洗衣服的零钱），零碎又不方便存储，如果有了这样一个装置，那么就可以把这些零钱兑换成整钱，再把这些零钱换给便利店或者洗衣房。在我完善这个装置后我会把装置放在碧峰 1 舍的楼下试运行。

6. 源代码解释：

整个装置工作的顺序请参见下方源码旁的注释

```
#include <Servo.h>          //include 舵机库
#define RS 22
#define RW 23              //LED 屏引脚命名
#define E 24
#define M 34               //步进电机 1 引脚命名
#define N 35
#define O 36
#define P 37
#define M1 38              //步进电机 2 引脚命名
#define N1 39
#define O1 40
#define P1 41
int pinD[8];              //LED8 个数据引脚命名
Servo myservo;            //6 个舵机命名 myservo 为硬币初速度控制闸门舵机
Servo yourservo;          //yourservo 为等待分类舵机完成的闸门舵机
Servo hisservo;           //hisservo 为 1 元硬币存储桶控制掉落的舵机
Servo herservo;           //herservo 为 5 角硬币存储桶控制掉落的舵机
Servo itsservo;           //itsservo 为硬币退出模块的舵机
Servo himservo;           //himservo 为纸币退出模块的舵机
void setup() {
    myservo.attach(2);     //为舵机分配控制端口
    yourservo.attach(3);
    hisservo.attach(4);
    herservo.attach(5);
    itsservo.attach(6);
    himservo.attach(7);
    myservo.write(120);    //设置各个舵机初始位置
    yourservo.write(120);
    hisservo.write(90);
    herservo.write(90);
    itsservo.write(180);
    himservo.write(120);
    Serial.begin(9600);    //通信串口设置
    pinMode(33,INPUT_PULLUP); //设置按钮信号输入口
    int i;
    //设置 22-32 34-41 为输出端口
    for(i=22;i<=32;i++)
```

```

    pinMode(i,OUTPUT);
    for(i=34;i<=41;i++)
        pinMode(i,OUTPUT);
    for(i=0;i<8;i++)          //将数组 pin[]赋值为输出的端口，用于 LED 数据传送
        pinD[i]=25+i;
}
double sum;                  //定义全局变量 sum 因为有 5 角，用.5 表示，数据类型为
//double
void loop() {
    long long t;              //定义变量 t，用于存储时间
    LCD_Setup();              //自定义函数 用于初始化 LCD 屏幕
    display_Insert();          //自定义函数 使 LCD 显示屏显示 “Insert coins”
    if(analogRead(A2)>=900&&analogRead(A2)<=1000) //当纸币进入模块的红外接受
    对被阻断
    {
        //即 A2 端口读到红外接收管的数值在
        //900 到 1000 之间时（当纸币遮挡红外线时仍有红外线会被接收到，数值经过试验为 900-
        //1000，说明有纸币进入，跳转至纸币进入模块
        Note_insert();         //自定义函数，纸币进入模块
    }
    if(!digitalRead(33))       //在 loop 函数循环过程中，若检测到接
    入 33 号端口的按钮（拉高）被按下，则进入硬币投入金额的判断
    {
        if((int)sum%5==0&&sum!=0&&(int)(sum-0.5)%5!=0) //若投入硬币金额是 5 的整
        倍数且不为 0 或者不为 X.5
        {
            Note_Exchange();    //进入纸币兑换模块
        }
        else
        {
            display_Inadequate(); //否则，显示 “Please insert more coins”
            delay(1000);          //显示 1 秒后回到 loop 函数
        }
    }
    display_Figure();           //显示当前的硬币金额总数
    if(analogRead(A1)>200)       //如果硬币轨道最前端红外被阻断，说明有硬币
    进入
    {
        delay(400);              //延迟 100 毫秒，待硬币静止后
        myservo.write(60);        //打开闸口
        delay(10);               //延迟，等待硬币通过硬币大小识别与前一个闸口间
        的轨道
    }
    if(analogRead(A0)>=600)      //如果硬币大小识别的红外接受对检测到被阻断
    {

```

```

t=millis();                //将此时的时间赋给 t;
while(1)                   //进入死循环
{
    if(analogRead(A0)<=600) //直到重新接收到红外信号，即硬币通过后
    {
        t=millis()-t;    //计算被阻断的时间，即硬币通过的时间

        Serial.println((int)t);
        myservo.write(120); //关闭上一个闸口，防止再有硬币进入
        if(t<90&&t>=80) //实验得出的 1 元硬币通过时间的范围
        {
            hisservo.write(30); //1 元硬币存储桶上方闸口打开
            herservo.write(90); //5 角硬币存储桶上方闸口关闭
            yourservo.write(60); //闸门打开，硬币滑入分类轨道
            sum+=1;           //金币总额加 1
        }
        else if(t>=60&&t<=75) //实验得出的 5 角硬币通过时间的范围
        {
            hisservo.write(90); //1 元硬币存储桶上方闸口打开
            herservo.write(30); //5 角硬币存储桶上方闸口关闭
            yourservo.write(60); //闸门打开，硬币滑入分类轨道
            sum+=0.5;           //金币总额加 0.5
        }
        else                    //若都不符合上述要求
        {
            hisservo.write(90); //1 元、5 角存储桶闸口均关闭
            herservo.write(90);
            yourservo.write(60); //闸门打开，硬币通过无法识别硬币
轨道退出装置
        }
        break; //退出循环
    }
}

    delay(500); //延迟 500 毫秒后
    yourservo.write(120); //yourservo 复位
}

}

void LCD_Setup()           //自定义函数初始化 LCD 显示屏的函数
{
    sendByte(0x38,0);
    delay(5);
}

```

```
    sendByte(0x38,0);
    delay(5);
    sendByte(0x38,0);
    delay(5);
    sendByte(0x38,0);
    delay(1);
    sendByte(0x08,0);
    delay(1);
    sendByte(0x01,0);
    delay(1);
    sendByte(0x06,0);
    delay(1);
    sendByte(0x0C,0);
    delay(1);
}
```

void sendByte(byte x,int Datain) //向 LCD 显示屏发送数据或指令的函数

```
{
    int i;
    byte mask=0x01;
    if(Datain)
        digitalWrite(RS, HIGH);
    else
        digitalWrite(RS, LOW);
    for(i=0;i<8;i++)
    {
        if(mask&x)
            digitalWrite(pinD[i],HIGH);
        else
            digitalWrite(pinD[i],LOW);
        mask<<=1;
    }
    digitalWrite(E, HIGH);
    delayMicroseconds(1);
    digitalWrite(E, LOW);
    delayMicroseconds(1);
}
```

void display_Insert() //自定义函数 显示” Insert coins” 字样

```
{
    sendByte('I',1);
    sendByte('n',1);
    sendByte('s',1);
    sendByte('e',1);
}
```

```
    sendByte('r',1);
    sendByte('t',1);
    sendByte(' ',1);
    sendByte('c',1);
    sendByte('o',1);
    sendByte('i',1);
    sendByte('n',1);
    sendByte('s',1);
    sendByte(':',1);
}
```

```
void display_Exchange() //自定义函数 两行显示 “Exchanging please wait” 字样
{
    LCD_Setup();
    char *p="Exchanging           Please wait";
    while(*p!=0)
    {
        sendByte(*p,1);
        p++;
    }
}
```

```
void display_Inadequate()//自定义函数 两行显示 “Please insert more coins” 字样

{
    LCD_Setup();
    char *p="Please insert           more coins";
    while(*p!=0)
    {
        sendByte(*p,1);
        p++;
    }
}
```

```
void display_Figure() //自定义函数 显示当前投入金额
{
    if(sum<10)
    {
        sendByte((int)sum/1+'0',1);
        sendByte('.',1);
        sendByte(((int)(sum*10))%10+'0',1);
    }
    else if(sum>=10)
    {
```

```

        sendByte((int)sum/10+'0',1);
        sendByte((int)sum%10+'0',1);
        sendByte('.',1);
        sendByte(((int)(sum*10))%10+'0',1);
    }
}

void Note_Exchange() //函数 纸币兑换函数
{
    int i;
    long long t;    //定义 t 存储时间
    t=millis();      //t 赋值为当前时间
    display_Exchange(); //显示 “Exchanging please wait”
    for(i=1;i<=(sum/5);i++) //退出相应张数 5 元纸币
    {
        while(millis()-t<=15000) //运行时间为 15 秒钟，滚轮顺向旋转
        {
            Phase_A();
            Phase_A1();          //设置 A、A1 相位
            delay(3);

            Phase_B();           //设置 B、B1 相位
            Phase_B1();
            delay(3);

            Phase_C();           //设置 C、C1 相位
            Phase_C1();
            delay(3);

            Phase_D();           //设置 D、D1 相位
            Phase_D1();
            delay(3);
            if(millis()-t==500) //当运行 500 毫秒后
            {
                himservo.write(180); //让纸币退出模块舵机旋转一定角度将纸币塞入滚轮
                delay(1000);         //等待舵机完成动作
            }
        }
        himservo.write(120); //15 秒后，纸币退出，该舵机复位
    }
    sum=0;          //投入总金额清零
}

```

```
void Note_insert() //函数 纸币进入函数

{
    sum=sum+5;    //金额总数加 5
    display_Figure(); //显示当前金额总数
    while(1) //滚轮逆向旋转
    {
        if(!digitalRead(33)) //若按下兑换按钮，停止循环
            break;
        Phase_D();           //设置 D 相位
        Phase_D1();
        delay(3);

        Phase_C();           //设置 C 相位
        Phase_C1();
        delay(3);

        Phase_B();           //设置 B 相位
        Phase_B1();
        delay(3);

        Phase_A();
        Phase_A1();           //设置 A 相位
        delay(3);
    }
    Coin_output();           //函数 硬币退出模块
}

void Coin_output() //函数 硬币退出模块
{
    display_Exchange(); //显示 “Exchanging please wait”
    int i;
    for(i=1;i<=sum;i++) //推硬币舵机工作 5 次
    {
        itsservo.write(70); //舵机旋转推出硬币
        delay(1000);
        itsservo.write(180); //舵机归位，下一枚硬币滑落
        delay(2000);
    }
    sum=0; //金币总额清零
}

void Phase_A()
{
    digitalWrite(M,HIGH); //A1 引脚高电平
```

```
    digitalWrite(N,LOW);
    digitalWrite(O,LOW);
    digitalWrite(P,LOW);
}

void Phase_A1 ()
{
    digitalWrite(M1,HIGH);      //A1 引脚高电平
    digitalWrite(N1,LOW);
    digitalWrite(O1,LOW);
    digitalWrite(P1,LOW);
}

void Phase_B()
{
    digitalWrite(M,LOW);
    digitalWrite(N,HIGH);      //B1 引脚高电平
    digitalWrite(O,LOW);
    digitalWrite(P,LOW);
}

void Phase_B1 ()
{
    digitalWrite(M1,LOW);
    digitalWrite(N1,HIGH);      //B1 引脚高电平
    digitalWrite(O1,LOW);
    digitalWrite(P1,LOW);
}

void Phase_C()
{
    digitalWrite(M,LOW);
    digitalWrite(N,LOW);
    digitalWrite(O,HIGH);      //C1 引脚高电平
    digitalWrite(P,LOW);
}

void Phase_C1 ()
{
    digitalWrite(M1,LOW);
    digitalWrite(N1,LOW);
    digitalWrite(O1,HIGH);      //C1 引脚高电平
    digitalWrite(P1,LOW);
}
```

```
void Phase_D()
{
    digitalWrite(M,LOW);
    digitalWrite(N,LOW);
    digitalWrite(O,LOW);
    digitalWrite(P,HIGH);}    //D 引脚高电平
```

```
void Phase_D1()
{
    digitalWrite(M1,LOW);
    digitalWrite(N1,LOW);
    digitalWrite(O1,LOW);
    digitalWrite(P1,HIGH); }    //D1 引脚高电平
```