



华南理工大学
South China University of Technology

《数字系统创意设计》

课程报告

题目：Startleggs ——下蛋华（滑）鸡（稽）

姓名、学号：林会东 201636665056

学院、班级：软件学院 2016 级 4 班

姓名、学号：连木明 201630664918

学院、班级：软件学院 2016 级 4 班

姓名、学号：李榷基 201630664833

学院、班级：软件学院 2016 级 4 班

姓名、学号：黄文禹 201636664523

学院、班级：软件学院 2016 级 4 班

华南理工大学
二〇一六年十月

目 录

1 系统总体介绍	1
2 技术方案	1
2.1 结构设计与实现	1
2.1.1 外型设计	1
2.1.2 布线设计	1
2.1.3 选择轮盘设计	2
2.1.4 颜色识别器设计	2
2.2 电路设计与实现	3
2.3 程序设计与实现	4
2.3.1 主程序设计	4
2.3.2 采用面向对象化的编程思想	4
2.3.3 让蜂鸣器奏出旋律	5
2.3.4 颜色识别器	5
2.3.5 代码版本管理	5
3 实验测试及结果	5
4 项目总结	6
4.1 作品总结	6
4.2 成员心得	6
5 附录	7
5.1 设备清单表	7
5.2 小组成员名单及分工	7
5.3 部分源代码	8
5.3.1 主程序	8
5.3.2 舵机角度控制	8
5.3.3 蜂鸣器频率控制	9
5.3.4 颜色识别器数据处理	10
5.4 附件说明	10
6 参考文献资料	11
7 特别鸣谢	11

1 系统总体介绍

本作品的外形是一只气宇轩昂的白羽公鸡，支持的动作有：

1. 唱校歌。
2. 识别喂食颜色。
3. 识别到红色，眼睛变红，长鸣叫并产红色的蛋。
4. 识别到蓝色，眼睛变蓝，长鸣叫并产蓝色的蛋。
5. 识别到绿色，摇头并短促鸣叫。

2 技术方案

2.1 结构设计与实现

2.1.1 外型设计

采用轻质的木棒做骨架，用玻璃胶进行粘接。头部与身体分离。外壳用白色泡沫纸覆盖。整个鸡身由两只脚支撑，我们采用了高分子材料塑造了两只仿真度较高的鸡脚，可沿木棒滑动或绕木棒旋转，使得平衡重心变得方便不少。

2.1.2 布线设计

主要部件——轮盘放在两脚之间的腹部。开发板及面包板置于胸腹部（如图*fig 1*）。这样一来重心前移不少，于是在尾部增加了几块配重以保持平衡。

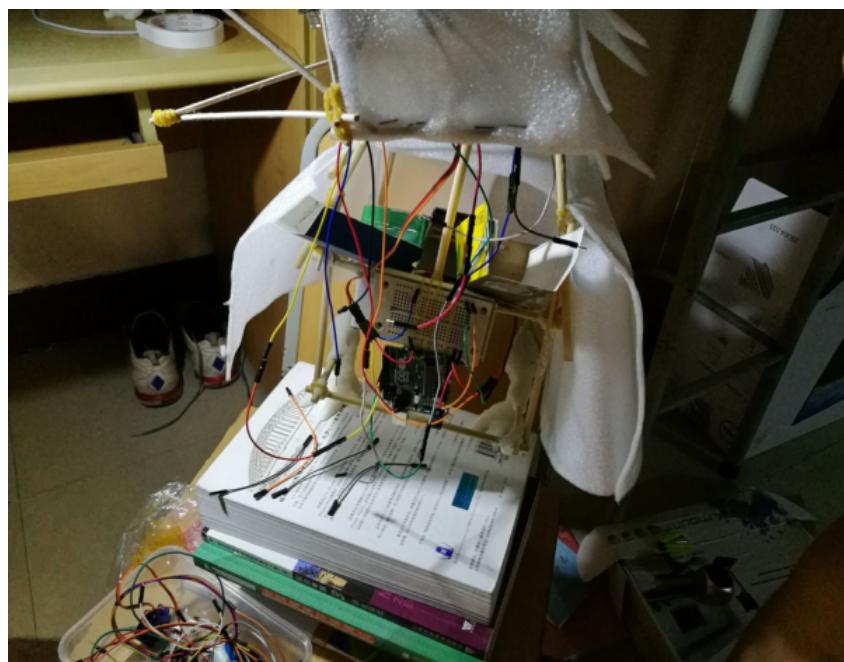


fig 1. 胸腹部的元件

2.1.3 选择轮盘设计

大体结构如 *fig 2* 所示，采用一个舵机控制选择槽（灰色）的转动。整个轮盘与水平面夹角约 45° 倾斜放置。在等候命令时，选择槽位于底盘（棕色）正中角度，两侧的挡板挡住了两个槽（红/蓝）中的乒乓球下落。当收到产红蛋的指令时，舵机逆时针旋转至左侧的红色槽，这时红色槽中的球落入选择槽，选择槽转动回正中位置，受重力作用从底板缺口处落下。当收到产蓝蛋指令时，进行相反方向转动。

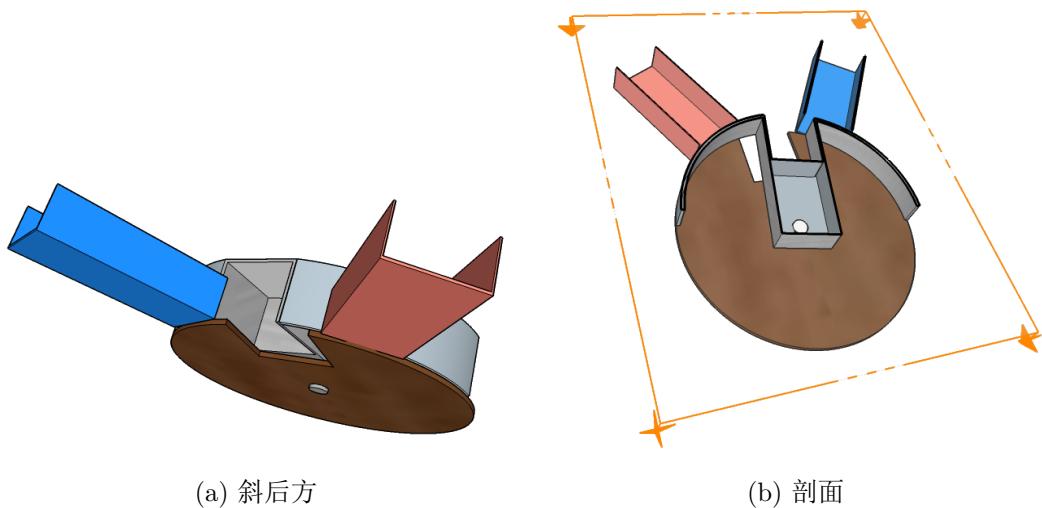


fig 2. 轮盘示意图

2.1.4 颜色识别器设计

为了获取所喂食物的颜色，我们把 TCS3200 由鼻腔位向下贴在上颚处（如图*fig 3*）。为了获得较大的色差对比，我们将下颚面补满白色，即方便了传感器的白平衡，又方便测量。

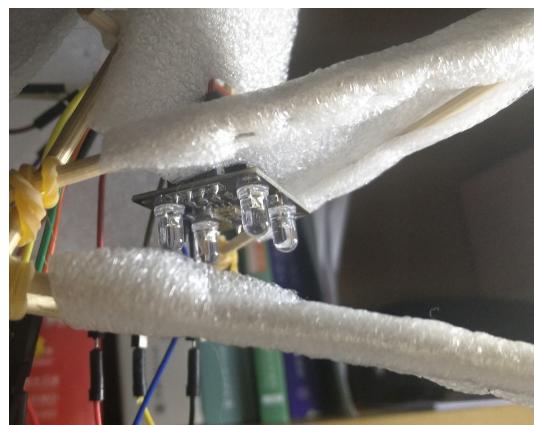


fig 3. TCS3200 的位置

2.2 电路设计与实现

所有元件通过面包板与 Uno 开发板连接，用充电宝即可提供电源。各元件针脚信息如 *tab 1* 所示。插线图如 *fig 4* 所示。

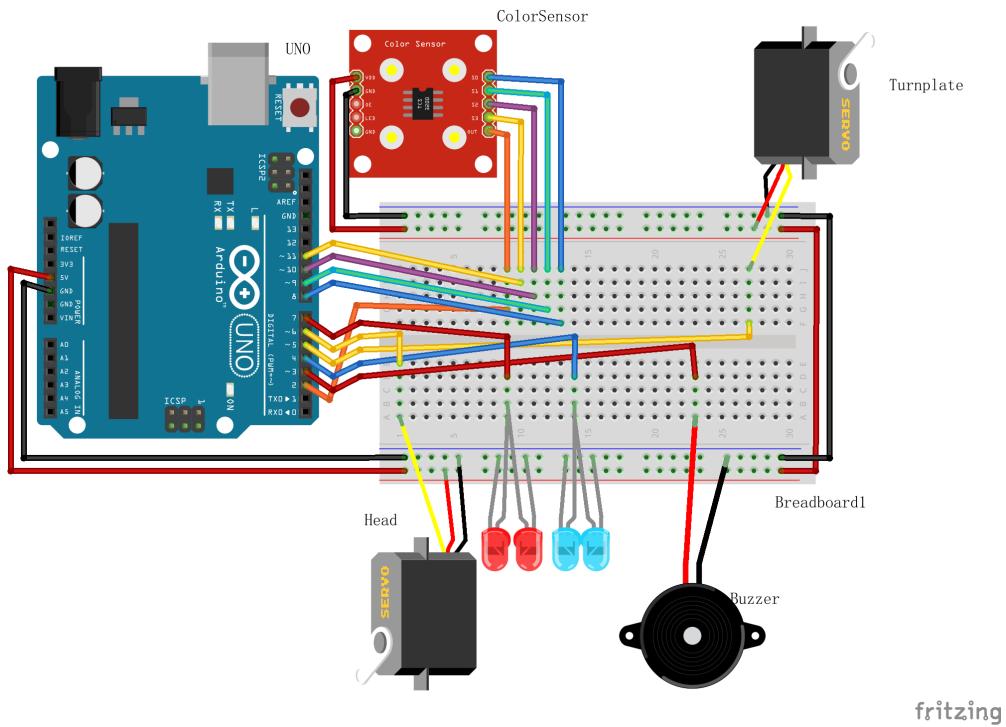


fig 4. 连线示意图

元件类型	元件名	元件引脚编号	开发板 pin 口编号
颜色识别器	ColorSensor	S0	D8
		S1	D9
		S2	D10
		S3	D11
		OUT	D2
		VDD	5V
		GND	GND
舵机	Head	SIG	D5
		VDD	5V
		GND	GND
舵机	Turnplate	SIG	D6
		VDD	5V
		GND	GND
蜂鸣器	Buzzer	SIG	D3
		GND	GND

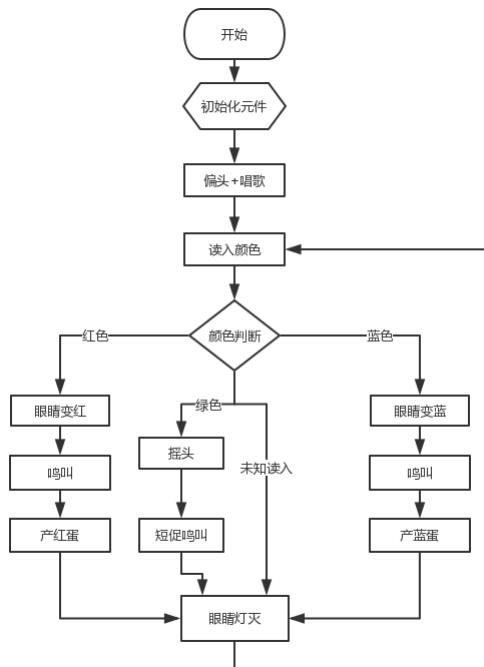
LED	RedEyes(L/R 并联)	+	D7
		-	GND
LED	BlueEyes(L/R 并联)	+	D4
		-	GND

tab 1. 针脚接线表

2.3 程序设计与实现

2.3.1 主程序设计

fig 5 表达了 Startleggs 的主流程。

*fig 5.* 主流程图

2.3.2 采用面向对象化的编程思想

为了让各元件分工明确、整个项目结构清晰、易于调试及维护，我们摒弃了 arduino 传统的面向过程化编程，而采用面向对象化。将各元件类型封装好，逻辑结构如 *tab 2* 所示。其中结构分层指该类在项目中的逻辑层级，第 I 层是最底层（主要是 pin 口操作），第 II 层是电子元件层（各元件的基础操作），第 III 层是动作层（数据综合分析并作出反应）。

虽然前期底层代码的编写耗费了比较多的时间，但是为工程中后期的维护及调试节约了大量的时间，提高了效率。

元件/实体名称	类名	结构分层
pin 口	Pin	I
蜂鸣器	Buzzer	II
颜色识别器	ColorSensor	II
LED 灯	LED	II
串口通信	Serial	II
舵机	SteeringGear	II
动作	Startleggs	III

tab 2. 类的逻辑结构

2.3.3 让蜂鸣器奏出旋律

为了让 Startleggs 能够发出有规律的、不同频率的声音，尤其是为了能够演奏校歌，我们需要得到关于音调的频率函数 $f(t)$ 。再通过 Arduino.h 中的 tone 函数发出特定频率的声音。

在 GitHub 上查询到了音调与脉冲频率的对应表，写入 pitches.h（详见附件）。

为了让两个音符之间有间隔区分，我们尝试了多种间隔时间，最后发现以 1.2 作为系数比较理想（详见代码）。

2.3.4 颜色识别器

TCS3200 的 OUT 口返回的是光线通过红、绿、蓝三种滤波器后的输出脉冲（通过修改工作滤波器分别返回），我们想要的值应该是读取到颜色的 RGB 值。所以需要用这三个脉冲参数推算出 RGB。

最终我们采用的方法是：依次选通三颜色的滤波器，利用 pulseIn 读入脉冲，再通过 map 函数与标准色脉冲数构造出一个近似 RGB 的值。这种方法误差虽然大，但是足以达到目标效果——区分红绿蓝。

2.3.5 代码版本管理

为了方便小组成员使用、更新代码，我们将项目发布到了 GitHub 上进行管理。

git: <https://github.com/OIdiotLin/arduino-startleggs>

3 实验测试及结果

系统共经历了 21 次调试，34 处代码修改。最终整体效果良好，各流程运行正常，符合初步设想。

4 项目总结

4.1 作品总结

通过多次的设计、编码、调试、测试、修改，Startleggs 如期完成。成员间协调顺利，工作分配恰当，大大加速了此系统的构建进程。

在最终展示环节表现较为突出，尤其能奏唱华工之歌，更是夺人眼目。Startleggs 的颜色识别技术和蜂鸣器的巧用，达到了本课程“创意设计”的目的，是一项较为满意的作品。希望今后能对其进行更好的维护。

4.2 成员心得

黄文禹：在这次 Startleggs 的制作中，我在实践中学会了 arduino 开发板的开发制作流程，同时在最后的视频制作过程我也更加熟练地掌握了 AE，PR 等软件的应用。同时我也再一次地感受到了团队的力量。在合理的分工合作下，我们顺利地完成了 Startleggs。每个人都按照自己的分工积极参与工程制作，虽然在调试的过程中也出现过许多小问题，但我们四人齐心协力，共同完成了这只 Startleggs。

林会东：第一次将所学知识融入了真实的项目，架构能力 ++，代码能力 ++，debug 能力 ++，积累了小型项目的团队开发经验，对单片机有了初步的了解。同时也加强了动手能力的培养。这门课程之于我的学习生涯是一片钥匙，打开了进入不一样的世界的大门。

李榷基：经过了为期一个多月的课程，我们从零开始，一点一点地向前迈进，终于完成了我们的作品。在这个过程中，我学到了很多接触比较少的知识，例如 arduino 的各个部件的实现等等。同时，我还从作品的装置和外型的设计过程中充分发挥自己的想象力和判断力，从而保证装置的可行性以及装置和外型之间的协调性。最后，我体会到在一个团队中每个人必须分工合作互相协调。

连木明：在这个数字系统创意设计课程中，虽然碰到很多困难，但却学到很多很有价值的东西：忙并快乐着。在参加课程过程中，ardunio 板的学习和实践让我们很忙，我有很多不懂，所以我查阅相关资料，请教同学老师，这个过程很忙，但是我学到很多知识，接触很多新奇的物品，这让感到很快乐；累并感动着。在具体的作品制作过程中，我负责的是立体结构设计，在这过程中，有时需要到实验室拿物品，有时碰到很难解决的立体结构问题，有时还会熬夜制作，但是我们四人互帮互助，在交流中解决问题，思想碰撞的过程想出很多很好的解决方案，我们团队共进退，同努力，虽然很累，但我感到感动。在这整个课程中，我学到 ardunio 的基础知识，学会了写程序实现小创意，学会了基础的立体结构设计，我觉得这个课程对我的以后的学习具有很重要的影响。

5 附录

5.1 设备清单表

tab 3 包含了本项目所用的所有主要材料。

名称	数量	用途
Arduino Uno	1	控制各元件按程序正常工作
颜色识别器 TCS3200	1	识别物体颜色
舵机	2	调整头部朝向、转动轮盘
蜂鸣器	1	发出声音
LED 灯	4	模拟眼睛
高分子材料	若干	制作鸡脚、固定元件
导线	若干	连接电路
面包板	1	连接电路
细木棒	若干	制作骨架

tab 3. 设备清单

5.2 小组成员名单及分工

姓名	工作内容
林会东 	程序设计与实现、电路设计、报告文档编辑
连木明 	结构设计与实现、组织统筹工作、外型设计与实现

李榷基	外型设计与实现、电路实现
黄文禹	电路设计与实现、摄像摄影、后期制作

tab 4. 成员分工

5.3 部分源代码

5.3.1 主程序

```
1 void setup() {
2     ancientGiantCock.init();
3     delay(2000);
4 }
5 void loop() {
6     ancientGiantCock.detectColor();
7     ancientGiantCock.changeEye();
8     if (ancientGiantCock.getCurrentColor() == GREEN) {
9         ancientGiantCock.shakeHead();
10    }
11    ancientGiantCock.layEgg();
12 }
```

5.3.2 舵机角度控制

```
1 /* These codes below are in SteeringGear.cpp */
2 void SteeringGear::servoPulse(int angle) {
3     // turn the angle to a pulse width
4     int pulseWidth = (angle * 11) + 500;
5     this->pin.setDigital(HIGH);
```

```

6   delayMicroseconds(pulseWidth);
7   this->pin.setDigital(LOW);
8   delay(20 - pulseWidth / 1000);
9 }
10 void SteeringGear::setAngle(int angle) {
11   this->servoPulse(angle);
12   delay(15);
13 }
14 void SteeringGear::sweep(int startAngle, int endAngle) {
15   if (startAngle < endAngle)
16     for (int angle = startAngle; angle <= endAngle; angle++)
17       this->setAngle(angle);
18   if (startAngle > endAngle)
19     for (int angle = startAngle; angle >= endAngle; angle--)
20       this->setAngle(angle);
21 }
```

5.3.3 蜂鸣器频率控制

```

1 /* These codes below are in Buzzer.cpp */
2 void Buzzer::beep(const int note, const double
3   noteDuration) {
4   // e.g. quarter note=1000/4, eighth note=1000/8, etc.
5   int realDuration = WHOLE_NOTE_DURATION / noteDuration;
6   tone(this->pin.getId(), note, realDuration);
7   // to distinguish the notes, set a minimum time
8   // between them.
9   delay(realDuration*INTERVAL_BETWEEN_NOTES);
10  noTone(this->pin.getId()); // stop the tone playing
11 }
12 /* int melody[] stores the frequency of the notes */
13 void Buzzer::playMelody(const int * melody, const double
14   *duration, const int size) {
15   for (int i = 0; i < size; i++) {
16     beep(melody[i], duration[i]);
17   }
18 }
```

5.3.4 颜色识别器数据处理

```

1  RGB ColorSensor::readRGB() {
2      RGB color;
3      shiftMode(1); // Setting red filtered
4      // Remapping the frequency to the RGB Model of 0 to 255
5      color.R = map(pulseIn(OUT.getId(), LOW), R.lower, R.
6          upper, RGB_MAX, 0);
7      delay(100);
8      shiftMode(2); // setting green
9      color.G = map(pulseIn(OUT.getId(), LOW), G.lower, G.
10         upper, RGB_MAX, 0);
11     delay(100);
12     shiftMode(3); // setting blue
13     color.B = map(pulseIn(OUT.getId(), LOW), B.lower, B.
14         upper, RGB_MAX, 0);
15     delay(100);
16     return color;
17 }
```

5.4 附件说明

/09STARTLEGGSS.zip

- └── arduino-startleggs.ino // 主程序
- └── Startleggs.h // 动作
- └── Startleggs.cpp
- └── Units // 元件及模块类
 - └── Buzzer.h // 蜂鸣器
 - └── Buzzer.cpp
 - └── ColorSensor.h // 颜色识别器
 - └── ColorSensor.cpp
 - └── LED.h // LED 灯
 - └── LED.cpp
 - └── Pin.h // pin 口
 - └── Pin.cpp
 - └── pitches.h // 音调及音值宏定义
 - └── Serial.h // 串口通信（调试用）
 - └── Serial.cpp
 - └── SteeringGear.h // 舵机
 - └── SteeringGear.cpp
- └── Report
 - └── report.tex // 报告 L^AT_EX
 - └── report.pdf // 报告文档
 - └── 09STARTLEGGSS.mp4 // 展示视频

fig 6. 目录树

6 参考文献资料

- Dejan Nedelkovski. *Arduino Color Sensing Tutorial – TCS230/TCS3200 Color Sensor*[CP/OL]. May 20, 2016.
<http://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/>
- Dan Power. *Arduino-Beep/pitches*[CP/OL]. Jul 4, 2014.
<https://github.com/dan-power/Arduino-Beep>
- 华南理工大学. 《华工之歌》简谱 [EB/OL]. 2008.
<http://www.scut.edu.cn/webpage/logo.htm>

7 特别鸣谢

- 感谢华南理工大学计算机科学与工程学院提供的实验室及设备支持。
- 感谢朱金辉教授的授课及指点。
- 感谢李扬助教的热心帮助。
- 感谢对我们的工作报以关注的每一位同学。