

《图形化编程与开源硬件》
设计
(2020-2021-2 学期)



题 目： 基于 Arduino 的交互式门禁控制系统设计
姓 名： 常子儒
班 级： 计师本 1901
学 号： 201911990102
学 院： 信息科学与工程学院

一、摘要

实验所需元器件：

Arduino 主控板、面包板、双色 LED 灯、RGB LED 灯、开关按键模块、有源蜂鸣器、U 型光电门、光敏传感器、触摸开关传感器、直流电机、DC 变压器模块、OLED 显示屏、舵机及导线若干

二、实验目的

解决客人到来提示及招待不及时的问题，通过动力系统实现半自动化，为自己出行后返回或客人到来提供便利，达到全面细致的功能

三、作品功能

户外部分：U 型光电门、双色 LED 灯，光敏传感器、RGB LED 灯、OLED 显示屏

室内部分：触摸开关传感器、有源蜂鸣器、开关按键

动力部分：舵机、直流电机、DC 变压器模块

实现的原理和功能：

①U 型光电门可以检测到是否有人，当客人到来后，U 型光电门感应到有人在，触发蜂鸣器（蜂鸣器安放在房内），蜂鸣器发出声响提示户主门外有客人到来，②这时，如果户主没有应答，双色 LED 灯亮红灯，若户主听到声响，会选择接触摸开关传感器，按下触摸开关后，双色 LED 灯亮绿色且会触发舵机工作，舵机由原来 0 度转为 120 度，从而实现将外门打开。③此外，在无人时 OLED 显示屏会显示“你好”，当 U 型光电门感应到有人时会变成“请稍等”，④当白天因天气原因亮度太低或到了晚上时，在 U 型光电门感应到有人后，会激活光敏传感器工作，当亮度达到一定值后，光敏传感器会触发 RGB LED 灯，使其发光明亮。⑤在户主按下触摸开关传感器后会激活按键开关控制的直流电机，当触摸开关和按键开关同时按下，且触摸开关持续触发时，直流电机才会工作（直流电机转速共有三档，通过按键开关调节），此时若松开触摸开关并按下按键开关，则正处于工作状态的直流电机停止工作，且舵机和双色 LED 灯会恢复初始状态，⑥在 U 型光电门感应并且户主按下触摸开关后，OLED 显示“你好”->“请稍等”，当客人进入或选择离开后，U 型光电门不在感应，OLED 显示保持“请稍等”不变，当房内按键开关按下后，OLED 显示变回“你好”

四、实验代码

图形模块：



```
定义 General_Control
如果 读取数字引脚 7 那么执行
  设置数字引脚 8 输出为 低电平
  OLED 128*64 第 1 行显示 "请稍等"
否则
  设置数字引脚 8 输出为 高电平
  OLED 128*64 第 1 行显示 "你好"

如果 读取数字引脚 A0 那么执行
  设置数字引脚 12 输出为 低电平
  设置数字引脚 13 输出为 高电平
  设置 9 引脚伺服舵机为 120 度
否则
  设置数字引脚 12 输出为 高电平
  设置数字引脚 13 输出为 低电平
  设置 9 引脚伺服舵机为 0 度

如果 读取数字引脚 A0 那么执行
  Dynamic_System
否则
  设置 Speed1 的值为 0
  clockwise
```



```
定义 Dynamic_System
设置 buttonState 的值为 读取数字引脚 A3

如果 变量 buttonState = 1 那么执行
  如果 变量 buttonState = 1 那么执行
    等待直到 非 读取数字引脚 A3
    设置 stat 的值为 变量 stat + 1
    如果 变量 stat >= 3 那么执行
      设置 stat 的值为 0
  串口 字符串输出 变量 stat 换行

如果 变量 stat = 1 那么执行
  设置 Speed1 的值为 125
  clockwise
否则
  如果 变量 stat = 2 那么执行
    设置 Speed1 的值为 250
    clockwise
  否则
    设置 Speed1 的值为 0
    clockwise
```



自动生成代码:

```

/*!
 * MindPlus
 * uno
 *
 */
#include <DFRobot_Servo.h>
#include <DFRobot_SSD1306_I2C.h>

// 动态变量
volatile float mind_n_stat,

```

```

mind_n_val, mind_n_Speed1,
mind_n_buttonState;

void DF_clockwise();
void DF_General_Control();
void DF_Dynamic_System();
// 静态常量
const char chCode chBuf[] = {
    {0x8bf7, 0xe8afb7, 0xc7EB, {0x00,
0x40, 0x40, 0x40, 0x27, 0xfc, 0x20, 0x40, 0x

```

```

03, 0xf8, 0x00, 0x40, 0xe7, 0xfe, 0x20, 0x00,
0x23, 0xf8, 0x22, 0x08, 0x23, 0xf8, 0x22, 0x
08, 0x2b, 0xf8, 0x32, 0x08, 0x22, 0x28, 0x02,
0x10}},
    {0x7a0d, 0xe7a88d, 0xC9D4, {0x08,
0x20, 0x1d, 0x24, 0xf0, 0xa4, 0x10, 0xa8, 0x
10, 0x20, 0xfd, 0xfc, 0x11, 0x04, 0x31, 0x04,
0x39, 0xfc, 0x55, 0x04, 0x55, 0x04, 0x91, 0x
fc, 0x11, 0x04, 0x11, 0x04, 0x11, 0x14, 0x11,
0x08}},
    {0x7b49, 0xe7ad89, 0xB5C8, {0x20,
0x40, 0x3f, 0x7e, 0x48, 0x90, 0x85, 0x08, 0x
01, 0x00, 0x3f, 0xf8, 0x01, 0x00, 0x01, 0x00,
0xff, 0xfe, 0x00, 0x00, 0x00, 0x20, 0x7f, 0x
fc, 0x08, 0x20, 0x04, 0x20, 0x04, 0xa0, 0x00,
0x40}},
    {0x4f60, 0xe4bda0, 0xC4E3, {0x08,
0x80, 0x08, 0x80, 0x08, 0x80, 0x11, 0xfe, 0x
11, 0x02, 0x32, 0x04, 0x34, 0x20, 0x50, 0x20,
0x91, 0x28, 0x11, 0x24, 0x12, 0x24, 0x12, 0x
22, 0x14, 0x22, 0x10, 0x20, 0x10, 0xa0, 0x10,
0x40}},
    {0x597d, 0xe5a5bd, 0xBAC3, {0x10,
0x00, 0x10, 0xfc, 0x10, 0x04, 0x10, 0x08, 0x
fc, 0x10, 0x24, 0x20, 0x24, 0x20, 0x25, 0xfe,
0x24, 0x20, 0x48, 0x20, 0x28, 0x20, 0x10, 0x
20, 0x28, 0x20, 0x44, 0x20, 0x84, 0xa0, 0x00,
0x40}}
};
// 创建对象
DFRobot_SSD1306_I2C oled12864;
Servo servo_9;

// 主程序开始
void setup() {
    oled12864.setChCode(chBuf);
    oled12864.begin(0x3c);
    servo_9.attach(9);
    Serial.begin(9600);
    mind_n_stat = 0;
    mind_n_val = 0;
}

void loop() {
    if (digitalRead(7)) {
        mind_n_val =

```

```

analogRead(A1);
        Serial.print("A1:");

        Serial.println(mind_n_val);
        if ((mind_n_val>=700)) {
            digitalWrite(4, HIGH);
            analogWrite(3, 255);
            analogWrite(10, 255);
        }
        else {
            digitalWrite(4, LOW);
            analogWrite(3, 0);
            analogWrite(10, 0);
        }
    }
    else {
        mind_n_val =
analogRead(A1);
        Serial.print("A1:");

        Serial.println(mind_n_val);
        if ((mind_n_val>=700)) {
            digitalWrite(4, LOW);
            analogWrite(3, 0);
            analogWrite(10, 0);
        }
        else {
            digitalWrite(4, LOW);
            analogWrite(3, 0);
            analogWrite(10, 0);
        }
    }
    DF_General_Control();
}

// 自定义函数
void DF_clockwise() {
    analogWrite(5, 0);
    analogWrite(6, mind_n_Speed1);
}

void DF_General_Control() {
    if (digitalRead(7)) {
        digitalWrite(8, LOW);

        oled12864.setCursorLine(1);
        oled12864.println("请稍
等");

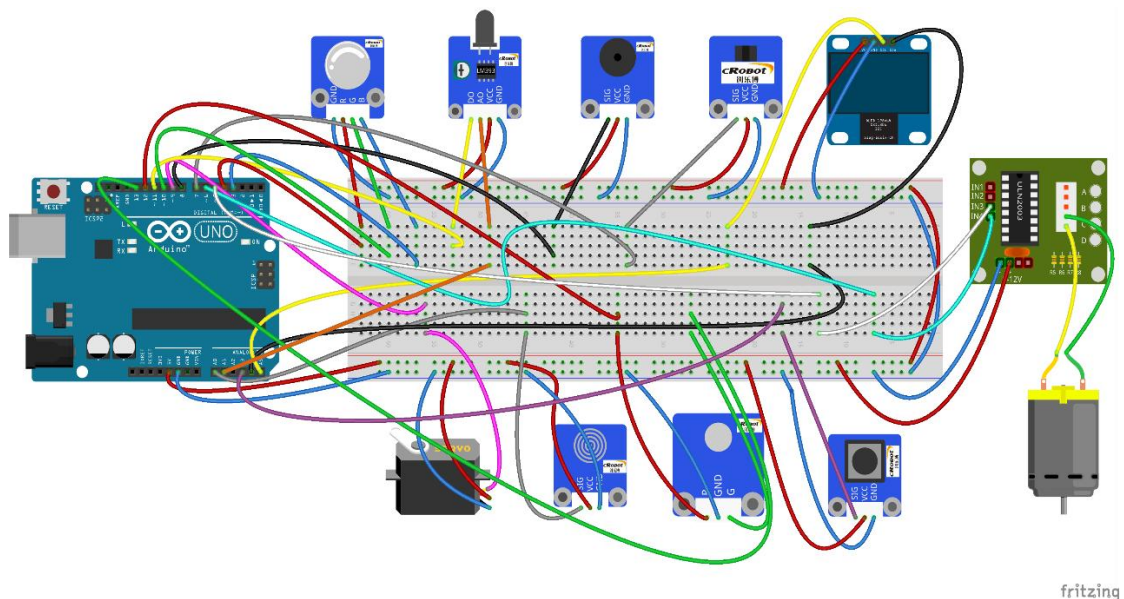
```

```

    }
else {
    digitalWrite(8, HIGH);
oled12864.setCursorLine(1);
    oled12864.println("你好");
}
if (digitalRead(A0)) {
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    servo_9.angle(abs(120));
}
else {
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
    servo_9.angle(abs(0));
}
if (digitalRead(A0)) {
    DF_Dynamic_System();
}
else {
    mind_n_Speed1 = 0;
    DF_clockwise();
}
}
void DF_Dynamic_System() {
    mind_n_buttonState =
digitalRead(A3);
    if ((mind_n_buttonState==1)) {
        if
        ((mind_n_buttonState==1)) {
            while
            ((!digitalRead(A3))) {
                mind_n_stat =
                (mind_n_stat + 1);
                if ((mind_n_stat>=3))
                {
                    mind_n_stat = 0;
                }
            }
            Serial.println(mind_n_stat);
            if ((mind_n_stat==1)) {
                mind_n_Speed1 = 125;
                DF_clockwise();
            }
            else {
                if ((mind_n_stat==2)) {
                    mind_n_Speed1 = 250;
                    DF_clockwise();
                }
                else {
                    mind_n_Speed1 = 0;
                    DF_clockwise();
                }
            }
        }
    }
}
}

```

五、实验接线图



六、实验结果

```
C:\Program Files (x86)\Mind\Arduino\hardware\tools\avr\bin\avr-g++ -c -g -Os -w -std-gnu++11
C:\Program Files (x86)\Mind\Arduino\hardware\tools\avr\bin\avr-gcc -w -Os -g -fno -fuse-li
"C:\Program Files (x86)\Mind\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -R .eeprom C:\
项目使用了11772字节，占用了(36%)程序存储空间，余留20484字节，最大为32256字节。
全局变量使用了689字节，(34%)的动态内存，余留1359字节局部变量，最大为2048字节。
C:\Program Files (x86)\Mind\Arduino\hardware\tools\avr\bin\avrdude -CC:\Program Files (x86)\Mind
avrdude: Version 6.3-20171130
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file is "C:\Program Files (x86)\Mind\Arduino\hardware\tools/a

Using Port : COM5
Using Programmer : arduino
Overriding Baud Rate : 115200
AVR Part : ATmega328P
Chip Erase delay : 9000 us
PAGEL : PD7
BS2 : PC2
RESET disposition : dedicated
RETRY pulse : SCK
serial program mode : yes
parallel program mode : yes
Timeout : 200
StabDelay : 100
CadenceDelay : 25
SyncLoops : 32
ByteDelay : 0
PollIndex : 3
PollValue : 0x53
Memory Detail
:

Block Poll
Memory Type Mode Delay Size Indx Paged Size #Pages MinM MaxM ReadBack
-----
eeprom 65 20 4 0 no 1024 4 0 3600 3600 0xff 0xff
flash 65 0 128 0 yes 32768 128 256 4500 4500 0xff 0xff
lfuse 0 0 0 0 no 1 0 0 4500 4500 0x00 0x00
hfuse 0 0 0 0 no 1 0 0 4500 4500 0x00 0x00
efuse 0 0 0 0 no 1 0 0 4500 4500 0x00 0x00
lock 0 0 0 0 no 1 0 0 4500 4500 0x00 0x00
calibration 0 0 0 0 no 1 0 0 0 0x00 0x00
signature 0 0 0 0 no 3 0 0 0 0x00 0x00

Programmer Type : Arduino
Description : Arduino
Hardware Version : 3
Firmware Version : 4.4
Vtarget : 0.3 V
Varef : 0.3 V
Oscillator : 28.800 khz
SCK period : 3.3 us

avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "C:\Users\ASUS\AppData\Local\DFScratch\build\dfrobot.ino.hex"
avrdude: writing flash (11772 bytes):

Writing | ##### | 100% 1.88s
avrdude: 11772 bytes of flash written
avrdude: verifying flash memory against C:\Users\ASUS\AppData\Local\DFScratch\build\dfrobot.ino.h
avrdude: load data flash data from input file C:\Users\ASUS\AppData\Local\DFScratch\build\dfrobot
avrdude: input file C:\Users\ASUS\AppData\Local\DFScratch\build\dfrobot.ino.hex contains 11772 by
avrdude: reading on-chip flash data:

Reading | ##### | 100% 1.51s
avrdude: verifying ...
avrdude: 11772 bytes of flash verified

avrdude done. Thank you.
```

七、实验总结与思考

实验也有些许不足或不合实际的问题，例如当U型光电门感应到人之后，房内蜂鸣器会发出声响，但当户主不想开门或房内没人时，这个功能或许有些多余，若是增加额外对蜂鸣器的控制开关，应该就可以解决这个问题；当房内的人按下触摸开关传感器后，原本工作的蜂鸣器不会因此停止工作；动力系统中的直流电机设计为传送物品，若外门到房内有点距离，可通过传送装置将行李或物品运到屋内或院子里，不过该功能所受限制过多。该系统既然是门禁系统，自然是一套完整的出入口管理系统，这样的话可以在入口（也就是外门）增加门禁卡识别以开放入口权限，此外应该还有些改进地方。

因为实验时间仓促，设计构思不够充分，设备实现功能受限，所以部分功能未实现。不过在这套系统上可以有更多的改进和优化。

八、实验演示图

