Оглавление

Задача

Списки дел и ежедневники – отличные помощники современного делового человека. Но вместе с тем, очевидно, что они даже близко не заменяют природную способность к запоминанию, ведь хорошая память важна сама по себе, поскольку является залогом успешной деятельности любого рода. Тренируя её, вы приучаете свой мозг к нагрузкам, и начинаете меньше забывать. И в этом помогают различные игры, которые призваны разбавить идентичные с точки зрения памяти задачи наших трудовых будней. Развлекательность и состязательность в данном отношении так же актуальна, как и полезность, ведь повышают интерес и устраняют усталость и перенапряжение.

Задача данного проекта создать игру «4 цвета», целью которой является развитие памяти игрока. Это достигается путём запоминания последовательно включающихся светодиодов.

Оборудование

- Arduino Leonardo – 1 шт.
- Кабель microUSB - USB – 1шт.
- Макетная плата (Breadboard) на 830 точек – 1 шт.
- Провод «папа-папа» - 28 шт.
- Зуммер (пассивный) – 1 шт.
- Светодиоды – 5 шт.
- Кнопка тактовая с колпачком – 4 шт.
- Резисторы 200 Ом – 5 шт.
- Резисторы 1кОм – 4 шт.

Схема подключения

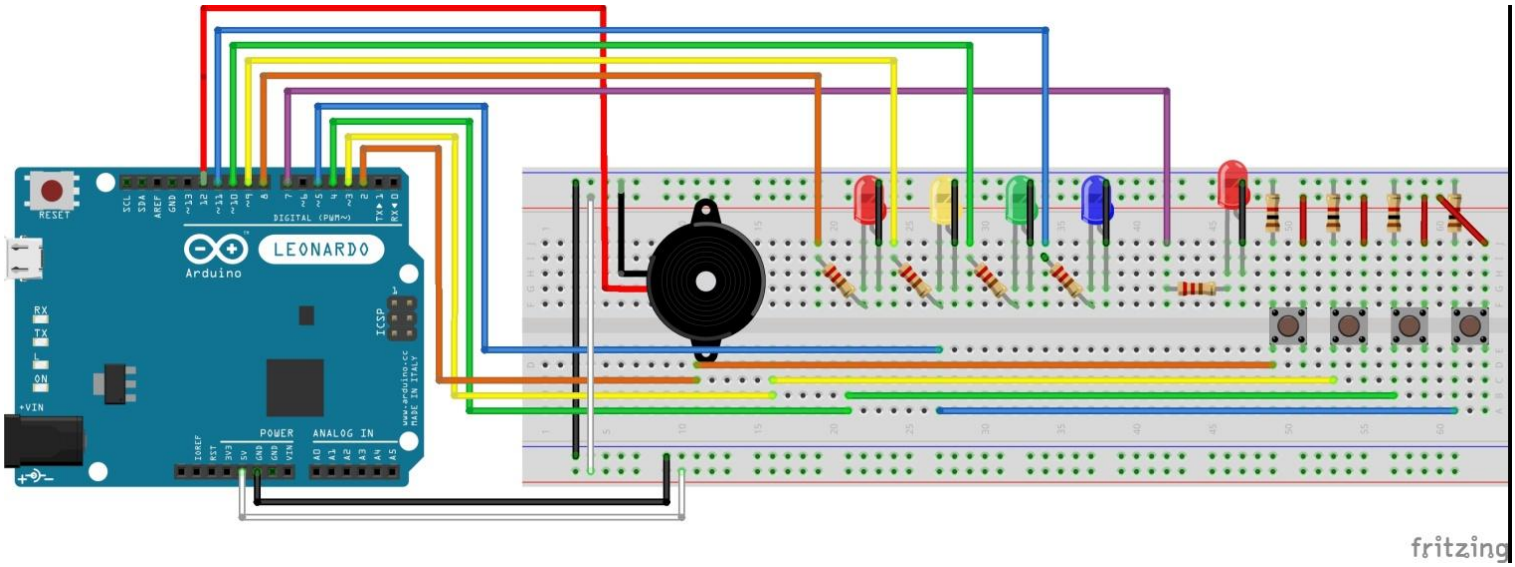

Рис.1 Схема подключения платы Arduino Leonardo

Схема подключения выполнена в программе Fritzing, предназначенной для облегчения процесса прототипирования.

Подключаем выход питания (5V) и землю (Gnd), белым и черным проводом соответственно к макетной плате.

Со 2-5 (включительно) цифровой пин подключаем каждую кнопку, тем самым на данные пины приходит сигнал о состоянии кнопки (нажата/ не нажата), который при необходимости можно считать. На выходе к каждой кнопке ставим подтягивающий резистор 1кОм, для определения точного сигнала на выходе (HIGH или LOW), т.к если вход оставить неподключенным, то на входе будет считываться HIGH или LOW случайным образом. Именно поэтому мы используем подтягивающий резистор, чтобы задать определенное значение при не нажатой кнопке.

С 7-11 (включительно) цифровой пин соединяем соответственно с каждым из разноцветных светодиодов, и при необходимости подаём на любой из светодиодов, после чего светодиод загорается. На входе перед каждым светодиодом стоит резистор на 220Ом, это обеспечивается для того, чтобы светодиод не перегорел.

12 цифровой пин соединяем с зуммером (пищалкой), и при необходимости подаём питание на зуммер, после чего происходит звуковой сигнал.

Каждый выход соединяем с землёй (Gnd).

Код программы

Программа выполнена в интегрированной среде разработки Arduino (Arduino IDE)

```
//buttons
const int buttonRed = 2;
const int buttonYellow = 3;
const int buttonGreen = 4;
const int buttonBlue = 5;

//leds
const int ledRed =  8;
const int ledYellow = 9;
const int ledGreen = 10;
const int ledBlue = 11;
const int ledAlarm = 7;

//buzzer
const byte buzz = 12;

// variables will change:
int buttonRedState = 0;
int buttonYellowState = 0;
int buttonGreenState = 0;
int buttonBlueState = 0;
boolean waitingFlag;

// variables for game
int *gameArray;
int pressedButton;
int currentPressButton;
int currentLed;

void setup() {

  // generate randomSeed for random values for LEDs
  randomSeed(analogRead(A0));

  // initialize the LED pin as an output:
  pinMode(ledRed, OUTPUT);
  pinMode(ledYellow, OUTPUT);
  pinMode(ledGreen,OUTPUT);
  pinMode(ledBlue, OUTPUT);
  pinMode(ledAlarm,OUTPUT);
```

```
  // initialize buttons
  pinMode(buttonRed,INPUT);
  pinMode(buttonYellow,INPUT);
  pinMode(buttonGreen,INPUT);
  pinMode(buttonBlue,INPUT);

 // initialize buzzer
  pinMode(buzz,OUTPUT);
 }

void waitingOne (){
  digitalWrite(ledRed,HIGH);
  delay(500);
  digitalWrite(ledYellow,HIGH);
  delay(500);
  digitalWrite(ledGreen,HIGH);
  delay(500);
  digitalWrite(ledBlue,HIGH);
  delay(500);
}
void waitingTwo(){
  digitalWrite(ledRed,LOW);
  delay(500);
  digitalWrite(ledYellow,LOW);
  delay(500);
  digitalWrite(ledGreen,LOW);
  delay(500);
  digitalWrite(ledBlue,LOW);
  delay(500);
}


boolean checkWaiting(){
      if (buttonRedState == LOW && buttonYellowState == LOW &&
buttonGreenState == LOW && buttonBlueState == LOW){
      return true;
      }
      else {
        return false;
           }
}
```

```
void blackout(){
   digitalWrite(ledRed,LOW);
   digitalWrite(ledYellow,LOW);
   digitalWrite(ledGreen,LOW);
   digitalWrite(ledBlue,LOW);
}

void soundStart(){
  tone(buzz,2000);
  delay(500);
  noTone(buzz);
  }

void soundRed(){
  tone(buzz,200);
  delay (500);
  noTone(buzz);
}

void soundYellow(){
  tone(buzz,300);
  delay (500);
  noTone(buzz);
}

void soundGreen(){
  tone(buzz,400);
  delay(500);
  noTone(buzz);
}

void soundBlue(){
  tone(buzz,500);
  delay(500);
  noTone(buzz);
}

void soundLose(){
  int i;
  for (i=0;i<4;i++){
  tone (buzz, 5000);
  digitalWrite(ledAlarm,HIGH);
  delay (500);
  digitalWrite(ledAlarm,LOW);
```

```
  noTone(buzz);
  delay(500);
  }
}


void Alarm(){
  digitalWrite(ledAlarm,HIGH);
  checkButtonState();
  delay(300);
  pressedButton=waitingPressButton();
  digitalWrite(ledAlarm,LOW);
}


void(*resetFunc)(void) = 0;

void checkButtonState(){
buttonRedState = digitalRead(buttonRed);
buttonYellowState = digitalRead(buttonYellow);
buttonGreenState = digitalRead(buttonGreen);
buttonBlueState = digitalRead(buttonBlue);
}


void loop() {
checkButtonState();
waitingFlag = checkWaiting();
if (waitingFlag == true){
waitingOne();
}
else startGame();

checkButtonState();
waitingFlag = checkWaiting();
if (waitingFlag == true){
waitingTwo();
}
else  startGame();

}

void startGame(){
    int i;
```

```
    for (i=0;i<3;i++){
    digitalWrite(ledRed,HIGH);
    digitalWrite(ledYellow,HIGH);
    digitalWrite(ledGreen,HIGH);
    digitalWrite(ledBlue,HIGH);
    soundStart();

    digitalWrite(ledRed,LOW);
    digitalWrite(ledYellow,LOW);
    digitalWrite(ledGreen,LOW);
    digitalWrite(ledBlue,LOW);
    noTone(buzz);
    delay(500);
    }
  mainGame();
}

void mainGame(){
int i,j,k,n;
int stepCount=0;

for (i=0;i<stepCount+1;i++){
 blackout();
 delay(200);

gameArray=(int*)realloc(gameArray,(stepCount+1)*sizeof(gameArray
));
 stepCount++;

 currentLed = random(8,12);
 gameArray[i] = currentLed;

 for (j=0;j<stepCount;j++){
 digitalWrite(gameArray[j],HIGH);
  switch (gameArray[j]){
  case 8:
  soundRed();
  break;
  case 9:
  soundYellow();
  break;
  case 10:
  soundGreen();
  break;
```

```
  case 11:
  soundBlue();
  break;
  }
  delay(500);
  digitalWrite(gameArray[j],LOW);
 }
 blackout();
 n=0;
 for (k=0;k<stepCount;k++){
 delay(1000);

 Alarm();
 if (pressedButton == gameArray[k])
 {
  n++;
 }
 else break;
}

 if (n == stepCount)
 {
  continue;
  }
  else{
lose(stepCount);
  }
 break;
}
resetFunc();
}

int waitingPressButton(){
  if (buttonRedState == LOW && buttonYellowState == LOW &&
buttonGreenState == LOW && buttonBlueState == LOW){
    digitalWrite(ledRed,LOW);
    digitalWrite(ledYellow,LOW);
    digitalWrite(ledGreen,LOW);
    digitalWrite(ledBlue,LOW);
  }
  checkButtonState();
  currentPressButton = pressingButtonFunction();
  return currentPressButton;
  }
```

```
 int  pressingButtonFunction(){
   checkButtonState();
   if (buttonRedState == HIGH)
  { digitalWrite(ledRed,HIGH);
    soundRed();
    digitalWrite(ledRed,LOW);
    return 8;
   }
  if (buttonYellowState == HIGH)
  { digitalWrite(ledYellow,HIGH);
    soundYellow();
    digitalWrite(ledYellow,LOW);
    return 9;
  }
  if (buttonGreenState == HIGH)
  { digitalWrite(ledGreen,HIGH);
    soundGreen();
    digitalWrite(ledGreen,LOW);
    return 10;
    }
  if (buttonBlueState == HIGH)
  { digitalWrite(ledBlue,HIGH);
    soundBlue();
    digitalWrite(ledBlue,LOW);
    return 11;}
}

void lose(int stepCount){
  int i;
  free(gameArray);
  for (i=0;i<stepCount;i++){
  gameArray[i]=0;
  }
  soundLose();
}
```

Принцип работы программы

Программа состоит из двух режимов. Первый режим – режим ожидания, а второй режим - режим игры.

Режим ожидания – происходит последовательное включение, а затем последовательное отключение четырёх разноцветных светодиодов слева направо. Продолжается до тех пор, пока игрок не зажмёт одну из тактовых кнопок.

Режим игры – этот режим начинается с последовательного мигания четырёх разноцветных светодиодов, и сопровождающим звуковым сигналом в течение трёх секунд. Далее начинается сама игра. Сначала на 1 секунду загорается один из четырёх разноцветных светодиодов, а затем гаснет. После этого загорается предупреждающий светодиод, который находится рядом с кнопками, сигнализирующий о том, что необходимо зажать соответствующую кнопку. В случае, когда игрок не успевает зажать соответствующую кнопку или зажимает неправильную кнопку, происходит включение и отключение предупреждающего светодиода в течение 4 секунд, которое сопровождается звуковым сигналом, свидетельствующим о том, что игрок проиграл, и игра начинается сначала. Если же игрок зажимает правильную кнопку, то игра продолжается. При следующем ходе количество последовательно включающихся светодиодов увеличивается на один, тем самым увеличивая количество необходимых вовремя зажатых кнопок, которые соответствуют ранее загоравшимся светодиодам.

Стоить заметить, что светодиоды загораются в случайном порядке.

Это реализовано с помощью функции генератора псевдослучайных чисел - randomSeed(analogRead(A0)), где в качестве входящего параметра используется чтение аналогового сигнала с порта A0 (входящий параметр задаёт начало выдачи псевдослучайных значений на последовательности).

Таким образом, заранее неизвестно, какой загорится следующий светодиод в режиме игры.

Список литературы

1. Справочник языка Arduino [электронный ресурс]

URL: https://www.arduino.cc/en/Reference/HomePage

2. Справочник языка Arduino на русском языке [электронный ресурс]

URL: http://arduino.ru/Referencee