

Министерство образования, науки и молодежи Республики Крым
Малая академия наук «Искатель»
Международный фестиваль «Робофинист 2024»

Свободная творческая категория: средняя

РОБОТ, УПРАВЛЯЕМЫЙ ЖЕСТАМИ

Работу выполнил:

Сосновский Вячеслав Юрьевич

ученик 7 класса МБОУ «СОШ №26 им. М.Т.

Калашникова» г. Симферополя

Научный руководитель:

Курбет Елена Николаевна

Зав. отдел информационных технологий

«МАН «Искатель»

Цель работы - разработать энергоэффективную, компактную и не дорогую систему, позволяющую распознавать простые жесты человека и построить на ее базе робота для демонстрации работы системы.

Задачи работы

1. Подобрать комплектующие, осуществить сборку аппаратной части системы управления и робота;
2. Разработать в программе Компас 3D и распечатать на 3D принтере детали робота.
3. Разработать программное обеспечение для обработки данных от датчика жестов и выработки команд;
4. Выполнить настройку и тестирование робота, управляемого жестами.

Подробное описание робота.

Робот – исполнительное устройство с двумя или более программируемыми степенями подвижности, обладающее определенным уровнем автономности и способное перемещаться во внешней среде с целью выполнения поставленных задач.

Весной этого года я решил сделать робота, управляемого жестами. Он получился милый, но очень простой по функциональному назначению. Поэтому я решил сделать совершенно нового робота.

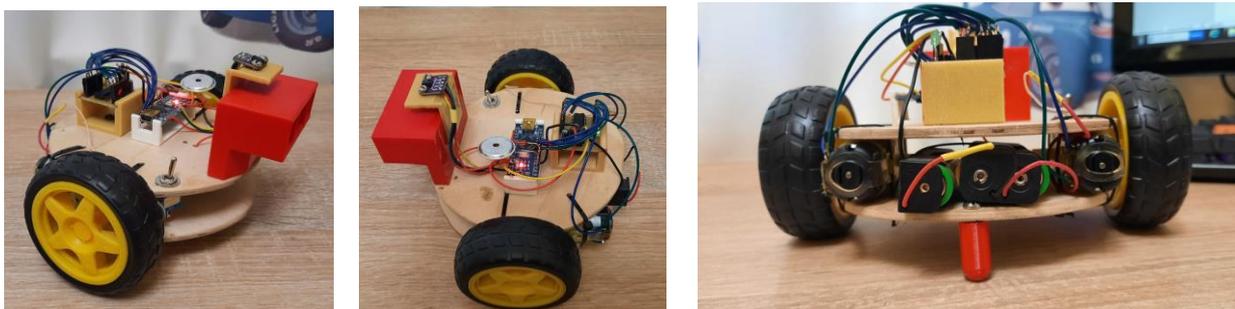


Рисунок 1. Робот, управляемый жестами 1.

Платформу (основу) для нового робота я вырезал из акрила. Сначала ножовкой сделал очертания шестиугольника. Затем закрепил с помощью болта и гайки на токарный станок и обточил деталь до формы круга.



Рисунок 2. Платформа (основа) робота.

В программе «Компас – 3D», я спроектировал и распечатал на 3D принтере основу для крепления колес и блока питания.

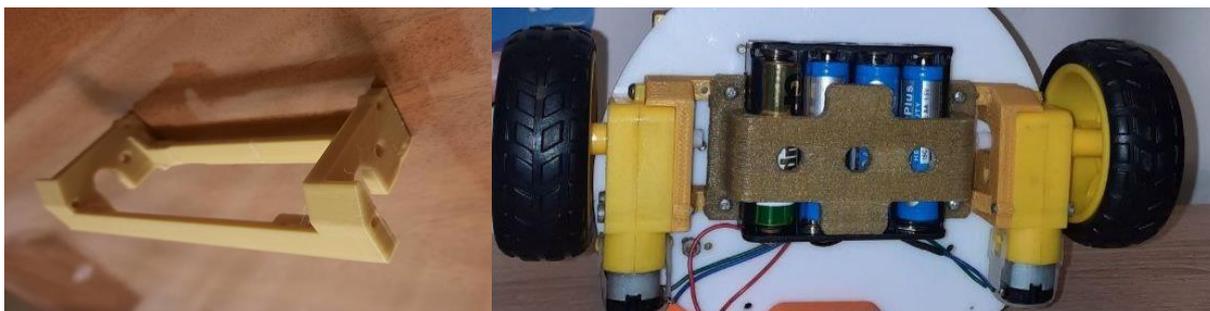


Рисунок 3. Основа для крепления колес и блока питания робота.

Далее, я собрал тележку с двумя моторами и колёсами. Но получившаяся тележка не держала равновесие и заваливалась назад. Для того, чтобы робот держал равновесие я установил на него еще одно колесо.

Подставку для колеса я спроектировал в программе «Компас – 3D» и распечатал на 3D принтере.

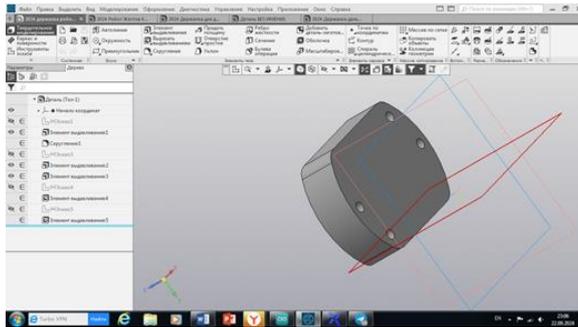


Рисунок 4. Подставка для колеса.

Так как мой робот будет питаться от четырех батареек, я спроектировал в программе «Компас – 3D» и распечатал на 3D принтере крепление для блока питания.

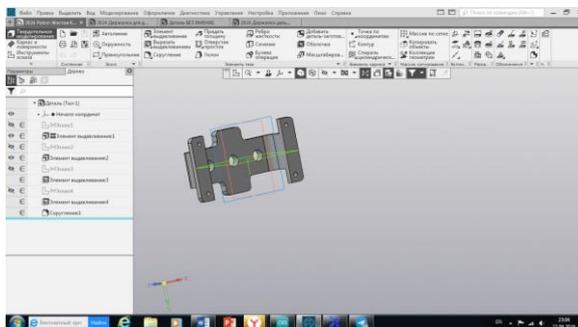


Рисунок 5. Крепление для блока питания.

Включать и выключать питание робота я решил тумблером. Для тумблера необходимо было сделать крепление.

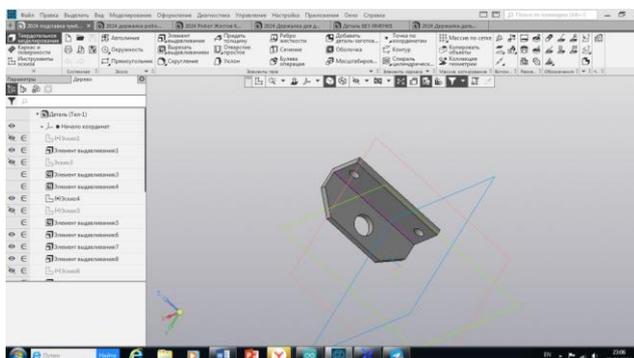


Рисунок 6. Крепление для тумблера.

Я болтами прикрепил все детали, припаял провода, смонтировал моторы и колеса, после чего вид у моего робота был такой.



Рисунок 7. Робот, вид снизу.

«Очки» у прежнего робота мне очень понравились, я решил сделать подобные и в новом роботе. Изначально я планировал установить в «очках» инфракрасный дальномер, но в связи с тем, что он работает в диапазоне от 20 см., его пришлось установить на другую подставку дальше, в «хвосте» робота. А «очки» используются для узнаваемости и как подставка для датчика распознавания жестов.

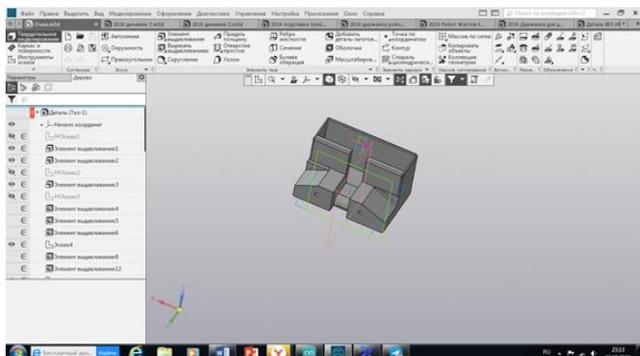


Рисунок 8. «Очки».

На очки я приклеил подставку для датчика распознавания жестов.

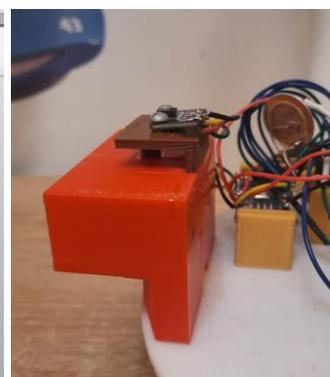
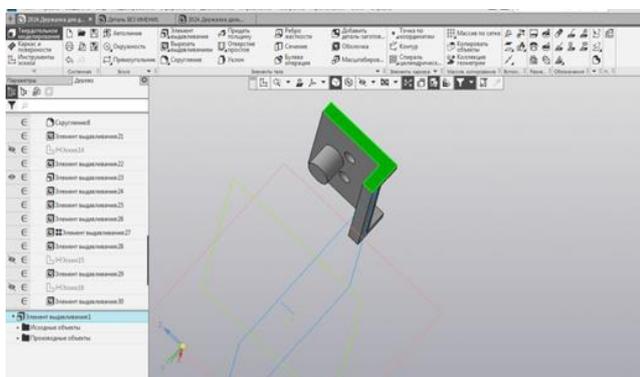


Рисунок 9. Подставка для датчика распознавания жестов.

В работе я использовал драйвер двигателя L293D и датчик распознавания жестов PAJ7620U2.



Рисунок 10. Датчик распознавания жестов

В корпус датчика встроены инфракрасный излучатель, инфракрасные сенсоры, микропроцессор и цифровой интерфейс. Движение руки определяется в диапазоне от 5 до 15 см от датчика, угол обзора 60°. Датчик настроен на распознавание 6 жестов (вперед, назад, вправо, влево, по часовой стрелке и против часовой стрелки). Датчик позволяет подключить его к «Arduino nano».

«Arduino Nano», это небольшая полнофункциональная отладочная плата, адаптированная для работы с макетными платами, построенная на базе контроллера ATmega328. Выводы для подключения «Arduino Nano» описаны на рисунке 11.

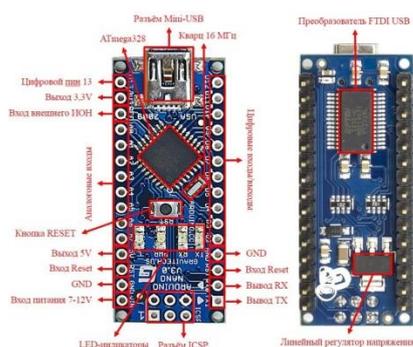


Рисунок 11. Плата «Arduino Nano».

Для вступительного слова перед демонстрацией работы робота и звукового сопровождения выполнения команд был использован аудио модуль (MP3-плеер) DFPlayer mini, карта памяти и динамик S1422 (с проводами 0.5 Вт 32 Ом).

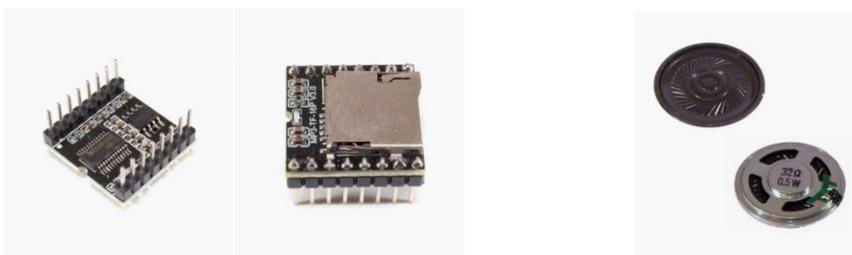


Рисунок 12. MP3-плеер и динамик.

Для крепления драйвера двигателя, «Arduino Nano» и MP3-плеера разработаны подставки.

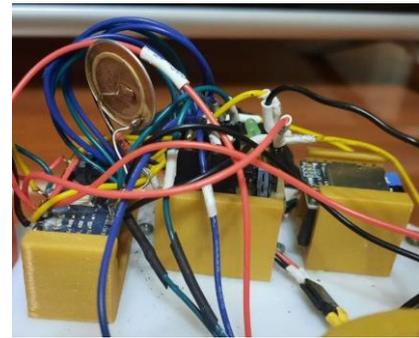
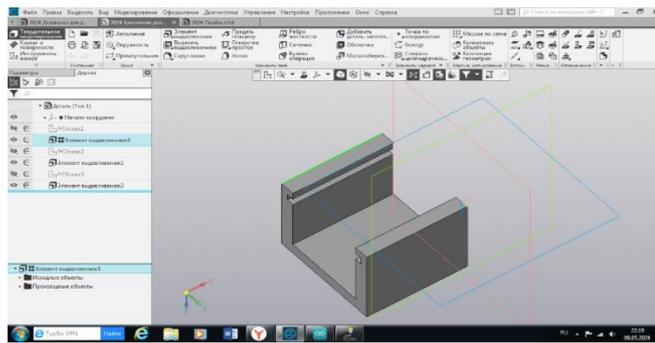


Рисунок 13. Подставки для драйвера двигателя, «Arduino Nano» и MP3-плеера.

Для улучшения качества звука была разработана колонка. Колонка прикреплена к платформе робота болтами.

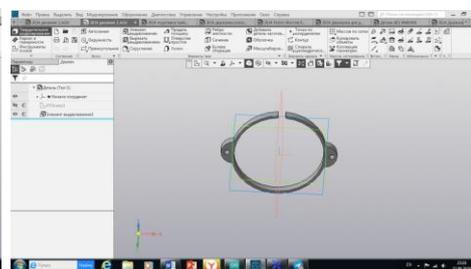
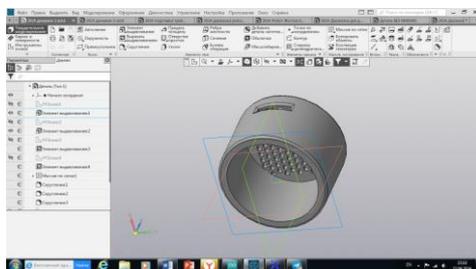


Рисунок 14. Колонка и кольцо - крепление для колонки.

Для измерения расстояния до препятствий использовался инфракрасный датномер Sharp 2Y0A02 F 74, предназначенный для измерения расстояния до препятствий от 20 до 150 см. (инфракрасный датномер подключается к любому аналоговому входу «Arduino Nano»).



Рисунок 15. Инфракрасный датчик расстояния Sharp 2Y0A02 F 74.

Для крепления инфракрасного датчика расстояния разработана подставка.

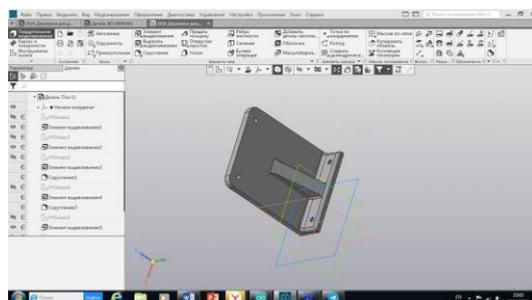


Рисунок 16. Подставка для инфракрасного датчика расстояния

Внешний вид робота

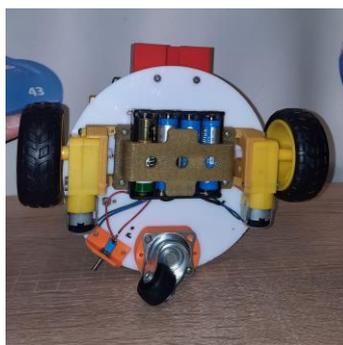
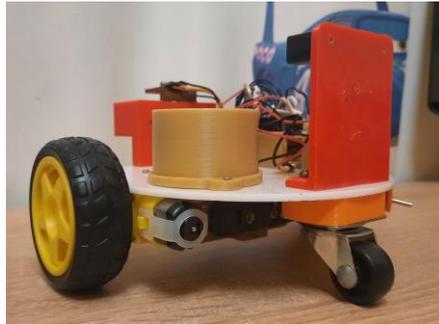
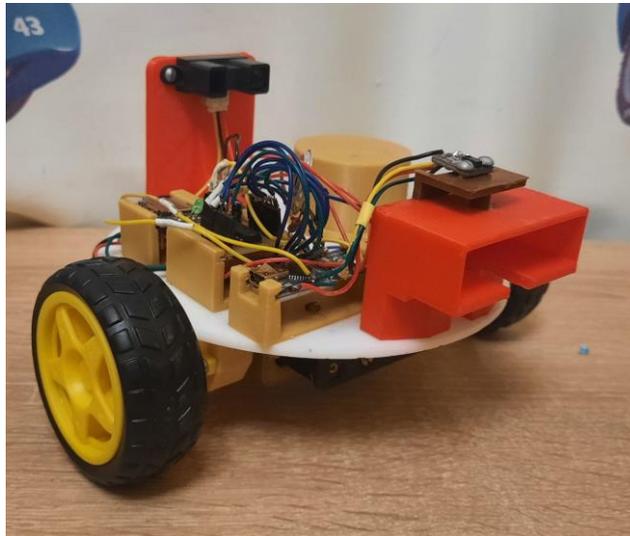


Рисунок 16. Внешний вид робота

Вывод

Я создал робота, он включает в себя механическую и электрическую часть, датчик распознавания жестов, плату «Arduino Nano», аудио модуль (MP3-плеер), карту памяти и динамик, инфракрасный дальномер Sharp, цифровой интерфейс и драйвер двигателя. Для работы робота я разработал и написал программу, которая принимает коды жестов и формирует команды выходов. Команды через драйвер двигателя управляют колесами робота. Также робот производит голосовое сопровождение выполнения команд и представление своей работы, измеряет расстояние до препятствий и определяет возможность движения робота вперед.

Уникальность робота в том, что я объединил в единую мобильную платформу компактного робота систему управления жестов и голосовую обратную связь. Данную систему можно применять на различных устройствах, так как далее все определяется написанной программой, вместо управления колесами может быть все что угодно, например включение/выключение света, открытие /закрытие штор, управление громкостью в устройстве и т.д. (для этого меняется получатель команд и производится небольшая корректировка программы).

Программный код

```
#include <DFRobot_PAJ7620U2.h> // Подключение библиотеки DFRobot_PAJ7620U2.h
#include <SoftwareSerial.h> // Подключаем библиотеку SoftwareSerial
#include <DFPlayerMini_Fast.h> // Подключаем библиотеку DFPlayerMini_Fast
SoftwareSerial mySerial(11, 12); // Указываем к каким портам подключен DFPlayer
DFPlayerMini_Fast myMP3;
DFRobot_PAJ7620U2 paj; // Создание экземпляра paj, класса DFRobot_PAJ7620U2
int donkey = 0;
int g1 = 1;
int g2 = 2; // Вызов переменной и присвоение ей значения 2
int range;
int j;
int arr[5];

void beep1(){ // Звуковой сигнал
  tone(3, 860, 1200);
}
void beep2(){
  tone(3, 790, 1200);
}
void beep3(){
  tone(3, 670, 1200);
}
void beep4(){
  tone(3, 580, 1200);
}
void forward(){ //Функция вперёд
  digitalWrite(7, HIGH); //На 7 пин подаётся 1
  digitalWrite(8, LOW); //На 8 пин подаётся 0
  digitalWrite(5, HIGH); //На 5 пин подаётся 1
  digitalWrite(9, HIGH); //На 9 пин подаётся 1
  digitalWrite(10, LOW); //На 10 пин подаётся 0
  digitalWrite(6, HIGH); //На 6 пин подаётся 1
  //myMP3.play(1);
  delay(390); //Задержка 2с.
  digitalWrite(5, LOW); //На 5 пин подаётся 0
  digitalWrite(6, LOW); //На 6 пин подаётся 0
}
void backward(){ //Функция назад
  digitalWrite(7, LOW); //На 7 пин подаётся 0
  digitalWrite(8, HIGH); //На 8 пин подаётся 1
  digitalWrite(5, HIGH); //На 5 пин подаётся 1
  digitalWrite(9, LOW); //На 9 пин подаётся 0
  digitalWrite(10, HIGH); //На 10 пин подаётся 1
  digitalWrite(6, HIGH); //На 6 пин подаётся 1
  //myMP3.play(2);
  delay(390); //Задержка 2с.
  digitalWrite(5, LOW); //На 5 пин подаётся 0
  digitalWrite(6, LOW); //На 6 пин подаётся 0
}
void left(){ // Функция влево
  digitalWrite(9, HIGH);
  digitalWrite(10, LOW);
  digitalWrite(6, HIGH);
  //myMP3.play(4);
  delay(650);
}
```

```

    digitalWrite(6, LOW);
}
void right(){ // Функция вправо
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(5, HIGH);
    //myMP3.play(3);
    delay(650);
    digitalWrite(5, LOW);
}
void clockwise(){ // Функция по часовой
    tone(3, 650, 1000);
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(9, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(6, HIGH);
    //myMP3.play(5);
    delay(1400);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}
void anticlockwise(){ // Функция против часовой
    tone(3, 750, 1000);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(5, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(6, HIGH);
    //myMP3.play(6);
    delay(1400);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}
void setup() // Процедура setup, выполняется однократно
{
    pinMode(9, OUTPUT); // Настройка вывода 9 как выход
    pinMode(10, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(5, OUTPUT);
    delay(5);
    mySerial.begin(9600); //установка скорости
    delay(100);
    myMP3.begin(mySerial); // инициализация
    delay(100);
    myMP3.volume(30); // Указываем громкость (0-30)
    delay(100);
    Serial.begin(115200); // Инициализация последовательного соединения, задание скорости 115200
    бит в сек
    delay(300);
    Serial.println("Gesture recognition system base on PAJ7620U2"); // Вывод сообщения о программе
    while(paj.begin() != 0){ // Если экземпляр paj не смог начать работу
        Serial.println("initial PAJ7620U2 failure! Please check if all the connections are fine, or if the wire
sequence is correct?"); // Вывод сообщения об ошибке
        delay(500);
    }
}

```

```

Serial.println("PAJ7620U2 init completed, start to test the gesture recognition function"); // Вывод
сообщения
myMP3.play(8);
*/
paj.setGestureHighRate(false); // paj.setGestureHighRate(true);
}
void loop() // Процедура цикла, вечное повторение
{
j++;
if(j > 4) j = 0; //если j больше 4, то j = 0
arr[j] = analogRead(A1);
range = (arr[0] + arr[1] + arr[2] + arr[3] + arr[4])/5; //range это средняя поступающая с порта A1
Serial.println(range);

DFRobot_PAJ7620U2::eGesture_t gesture = paj.getGesture(); // Считывание номера жеста
if(gesture != paj.eGestureNone) // Если ЖЕСТ не равен отсутствию жеста (т.е. успешно
считан) - открывается скобка,
{
}
String description = paj.gestureDescription(gesture); //Преобразовать номер жеста в описание строки
switch (gesture) { // Переключатель по значениям переменной gesture
case 1:
Serial.println("gesture number = 1");
Serial.println("gesture = backward"); // жест назад
//tone(3, 587, 1000);
myMP3.play(2);
backward();
break;
case 2:
Serial.println("gesture number = 2");
Serial.println("gesture = forward"); //жест в перёд
//range = analogRead(A1);
//Serial.print(range);
if ((range < 330) && (range >= 0)){
myMP3.play(1);
forward();
}
else {
myMP3.play(7);
}
//myMP3.play(1);
//forward();
break;
case 4:
Serial.println("gesture number = 4");
Serial.println("gesture = right"); //жест в право
//tone(3, 698, 1000);
myMP3.play(4);
right();
break;
case 8: // Если gesture равна 8
Serial.println("gesture number = 8");
Serial.println("gesture = left"); // жест влево
//one(3, 659, 1000); // Включение звука на выводе 3 частотой 659 (Нота Ми 2
октава) Гц на 1 сек
myMP3.play(3);
left();

break;
case 64:

```

```
Serial.println("gesture number = 64");  
Serial.println("gesture = clockwise");           //жест по часовой  
myMP3.play(5);  
clockwise();  
break;  
case 128:  
Serial.println("gesture number = 128");         //жест против часовой  
Serial.println("gesture = anticlockwise");  
myMP3.play(6);  
anticlockwise();  
break;  
}  
}
```