

Проект

Поющее растение

Методические рекомендации.

Тема «Операционный усилитель»

Цель: изучить процесс получение сигналов и увеличение их силы.

Задачи:

- изучить и закрепить на практике работу операционного усилителя LM358.
- получить и закрепить на практике знания, умения и навыки в области программирования и схемотехники.

Результат занятия: получение и усиление сигналов от растений.

Содержание

1. Начало урока.

(Учитель заранее раскладывает набор)

- Здравствуйте ребята. Мы продолжаем исследовать мир робототехники и автоматизированных устройств. Сегодня познакомимся с такой практикой в электронике, как усиление сигнала. Мы уже научились считывать сигналы (значения) с различных датчиков. Назовите их

Дети перечисляют датчики, с которыми уже работали (ультразвуковой датчик, датчик касания, датчик линии, датчик цвета).

- Отлично. Все эти датчики уже настроены, чтобы корректно получать и отправлять данные с окружающей среды. Но для некоторых исследований нет готовых датчиков или модулей. И поэтому их приходится собирать самим. Сегодня мы попробуем прочитать сигналы от растений.

Мы постараемся сегодня собрать такую установку и протестировать её.

Основная часть

- Сигналы от растений или электрические импульсы от нашего тела очень слабые. Поэтому их нужно усилить. Мы знаем, как усилить звук нашего голоса, например, кричать через рупор. А если хотим плавать в воде быстрее, то используем ласты.

- В электронике тоже есть свои инструменты усиления, такие как транзисторы и операционные усилители.

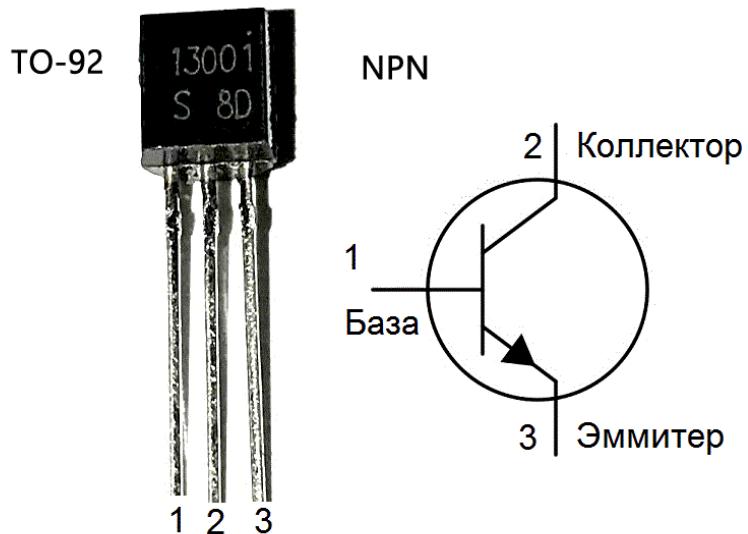


Рисунок 1 транзистор

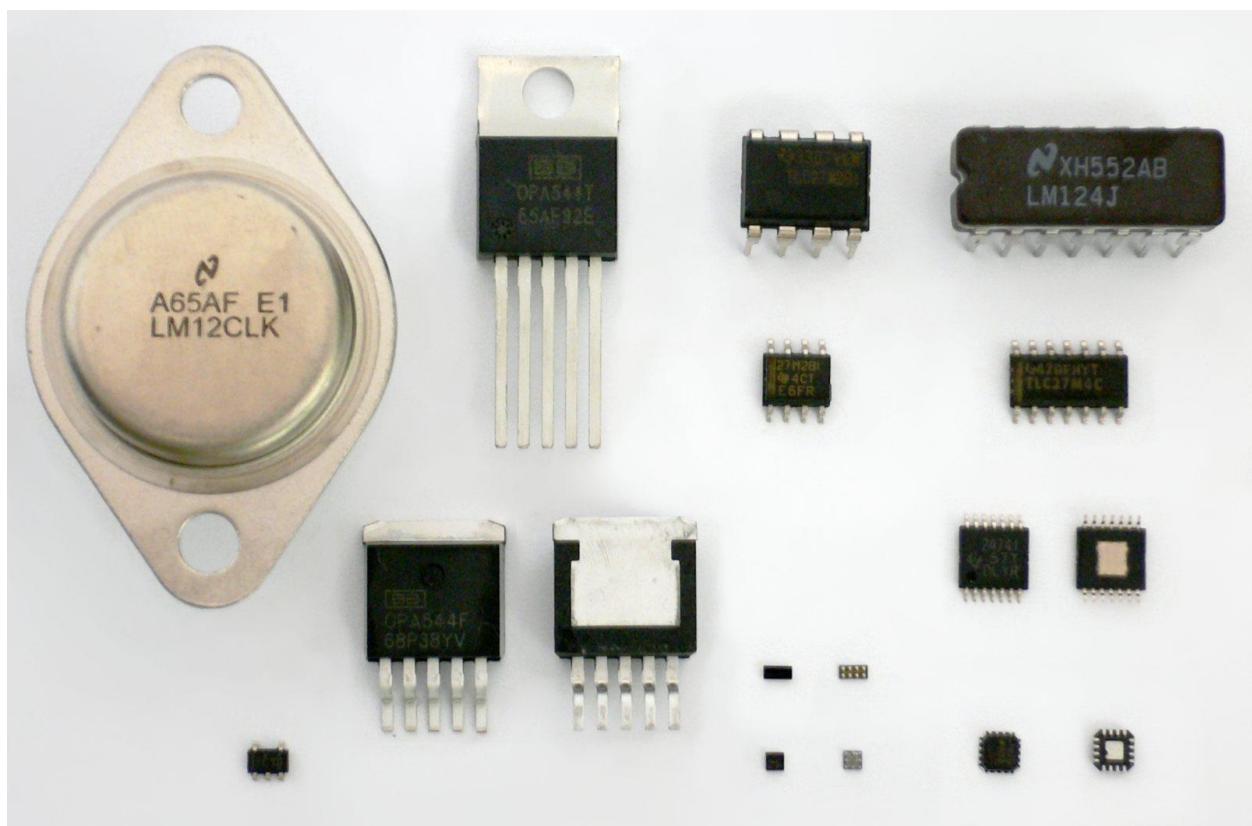


Рисунок 2 Операционные усилители

- Эти электронные компоненты усиливают ток, но коэффициент усиления у них разный. Этот коэффициент показывает во сколько ток на выходе усилен чем на входе в это электронное устройство.

- Сегодня мы будем использовать операционный усилитель LM358. Это небольшая микросхема. Она есть у каждого на столе и размещена на беспаечной плате.

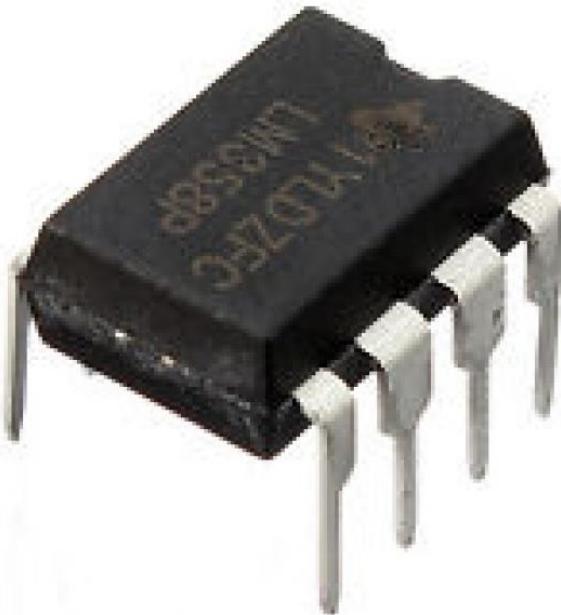


Рисунок 3 микросхема LM358

Теперь, нам нужно изучить контакты на ней. Контакты считаются слева вниз при положении микросхемы выемкой вверх

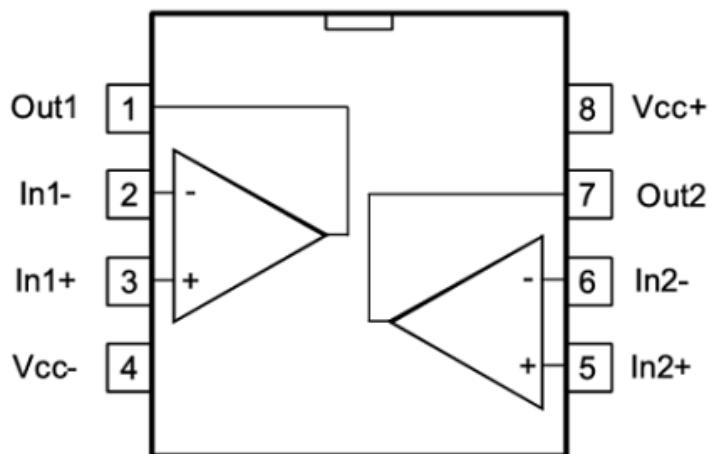


Рисунок 4 Внутренняя структура микросхемы

Щуп, которым будем прикладывать к растению, нужно будет подключить к контакту IN1, а IN2 подключим к GND (зелёному проводу) универсального шлейфа из набора Клик. Контакт VCC+ будет запитан красным проводом универсального шлейфа.

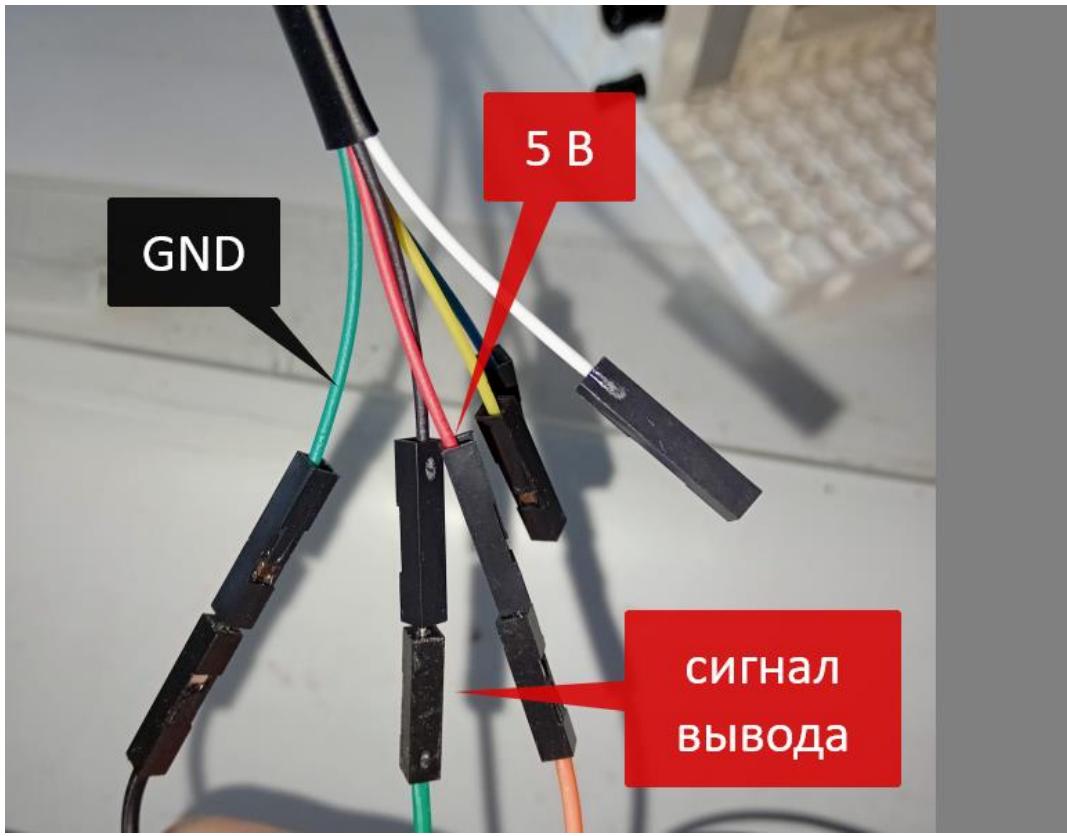


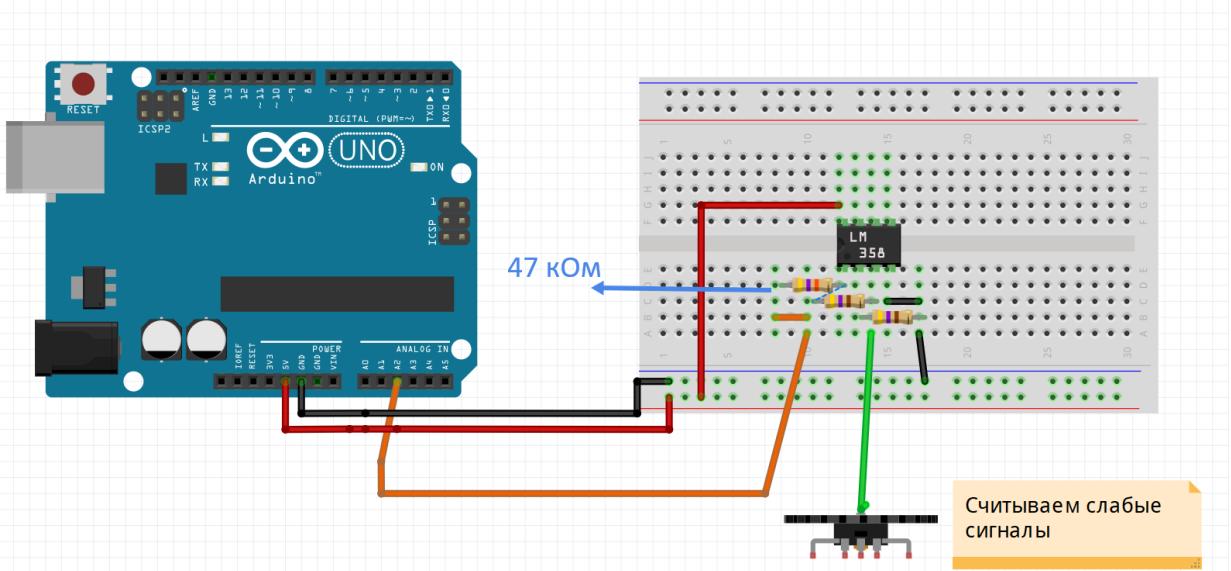
Рисунок 5 Подключение проводов к универсальному шлейфу.

Также нам понадобятся три резистора, из которых два имеют 470 Ом, а третий – 47 кОм.

Учитель заранее маркирует эти резисторы (можно дать задание определить сопротивление резисторов по меткам или с помощью мультиметра, в зависимости от возраста ученика).

Считывать выходной сигнал будем на 3 порт контроллера K1 Клика. К этому порту ведут пин A2 и A3.

Соберём схему с контроллером K1



fritzing

Рисунок 6 Схема усилителя сигнала

Дети собирают схему, а учитель проверяет сборку.

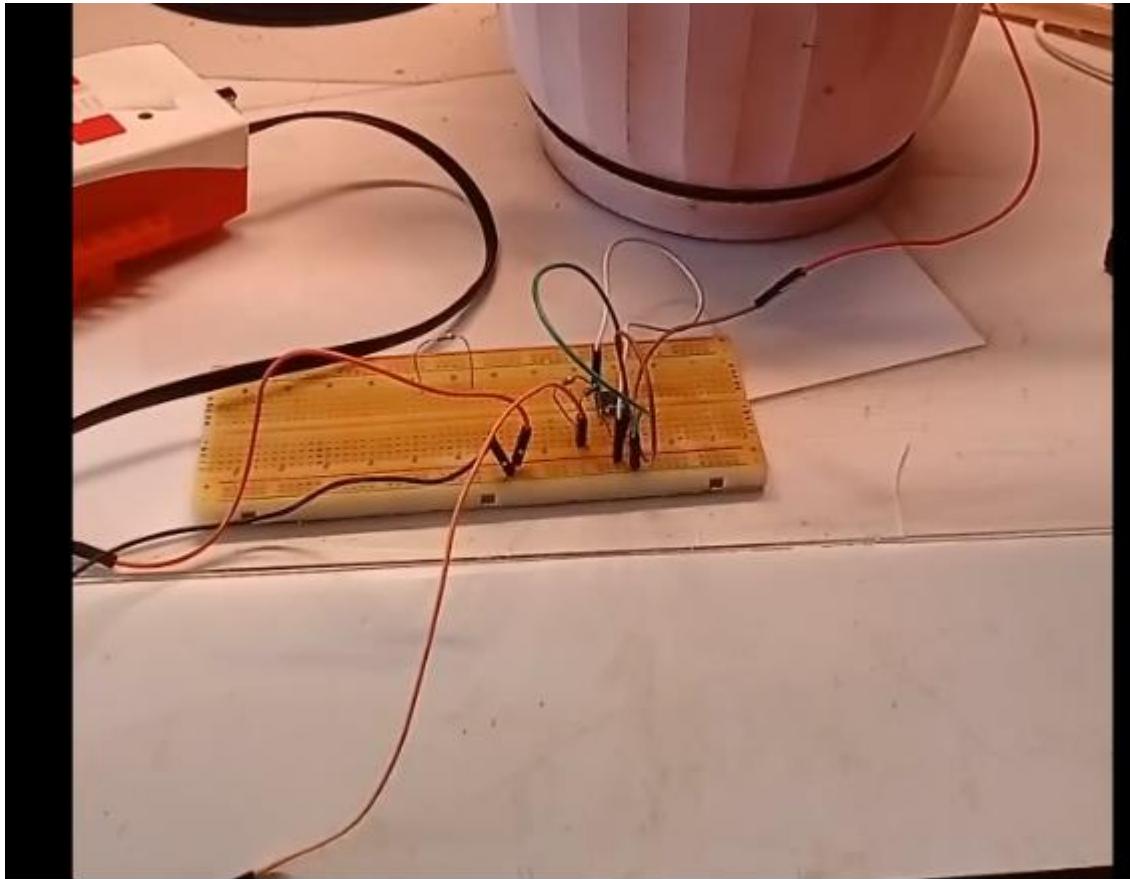


Рисунок 7 Пример сборки.

Теперь, необходимо написать программу для считывания данных и преобразовать их как звук с определённой частотой. Для этого будет считывать аналоговые данные с A2 и преобразовать звук с помощью функции tone. Значения выходящих данных могут быть в пределах 700 -1024. Это высокие частоты для зуммера. Поэтому поделим их на 5, тогда частота звука будет в пределах 50 -200 Гц.

Код программы.

```
int p = A0;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A2);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(100);           // delay in between reads for stability
  int sensorV = sensorValue/5;
```

```

    tone(p, sensorV);
}

}

```

При такой программе звук будет непрерывным и увеличиваться по частоте от прикосновения к растению.

Мы можем сделать так, чтобы схема издавала звук только при касании о растение. Для этого нужно считать первые 20 значений, которые будут калибровочными данными при которых звука не должно быть. Как только получаемые значения выходят за границы калибровки, то включаем звук.

Программа с калибровкой.

```

int p = A0;
int n = 0;
int m = 0;

int mm=0;
int nn=0;
// массив калибровочных данных
int datad[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int nomer = sizeof(datad) / sizeof(datad[0]); // Размер массива

// the setup routine runs once when you press reset:
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);

}

// the loop routine runs over and over again forever:
void loop() {
    // read the input on analog pin 0:
    int sensorValue = analogRead(A2);
    // print out the value you read:
    Serial.println(sensorValue);
    delay(100);           // delay in between reads for stability

    int sensorV = sensorValue/5;
    // запись в массив первые 10 данных
    if ((n >=10) &&(n <20))
    {
        datad[m]=sensorV;
        m+=1;
    }

    if (n<=20){
        n+=1;
    }

    if (n==20)
    {

        //Serial.println(" ");
        // Алгоритм пузырьковой сортировки
        for (int i = 0; i < nomer-1; i++) {
            for (int j = 0; j < nomer - i - 1; j++) {
                if (datad[j] > datad[j + 1]) {

```

```

        // Обмен элементов
        int temp = datad[j];
        datad[j] = datad[j + 1];
        datad[j + 1] = temp;
    }
}

// Вывод результата массива
//for (int i = 0; i < nomer; i++) {
//Serial.print(datad[i]);
//Serial.print(" ");

//}

// минимум и максимум массива
//Serial.println(" ");
mm =datad[0];
nn= datad[9];

//Serial.print("m= ");
//Serial.print(mm);
// Serial.print("n= ");
// Serial.print(nn);
}

// включаем звук при значениях, выходящих за границы массива

if ((sensorV<mm) || (sensorV>nn)){
tone(p, sensorV);
}
else
{
    tone(p, -1);
}
}

```

Загружаем программы и выводим данные на плоттер Arduino ide

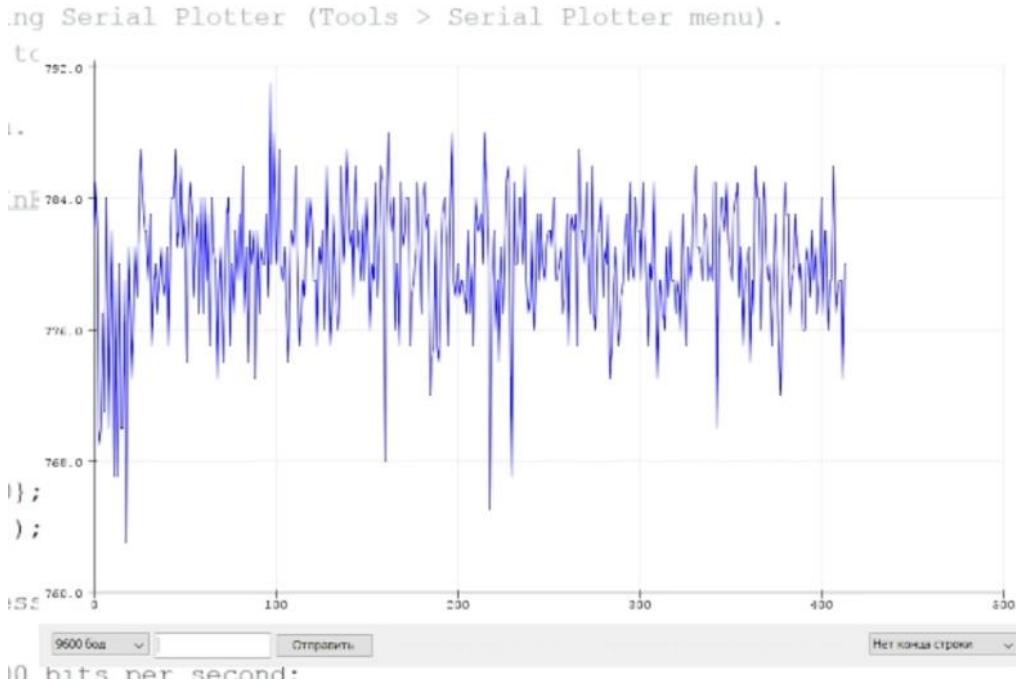


Рисунок 8 Вывод данных в плоттере arduino ide

Альтернативная сборка с CyberPi

Мы можем использовать контроллер cyberPi вместо контроллера K1



Рисунок 9 Подключаем провод с выводом данных к S1

Запитайте схему с контактов + и - с pocket shield cyberpi.

Программа для CyberPi пишется в mBlock5.

Подключите расширение «**pocket shield**» и «**display**».

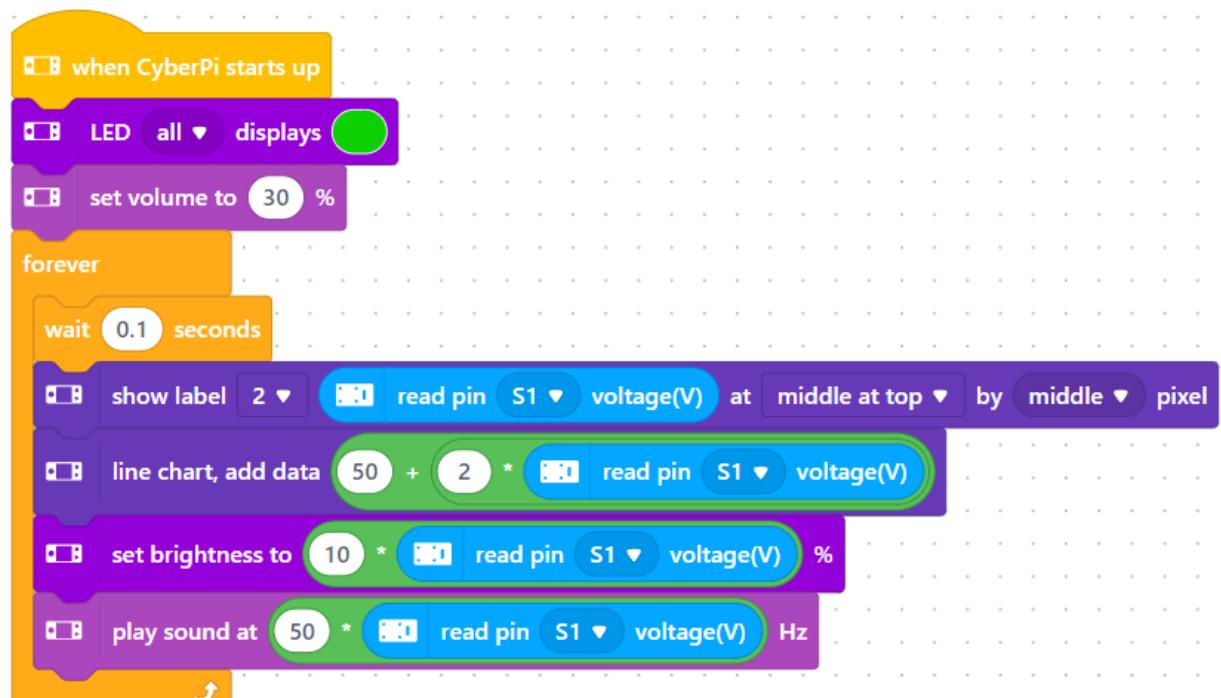


Рисунок 10 Первый вариант программы

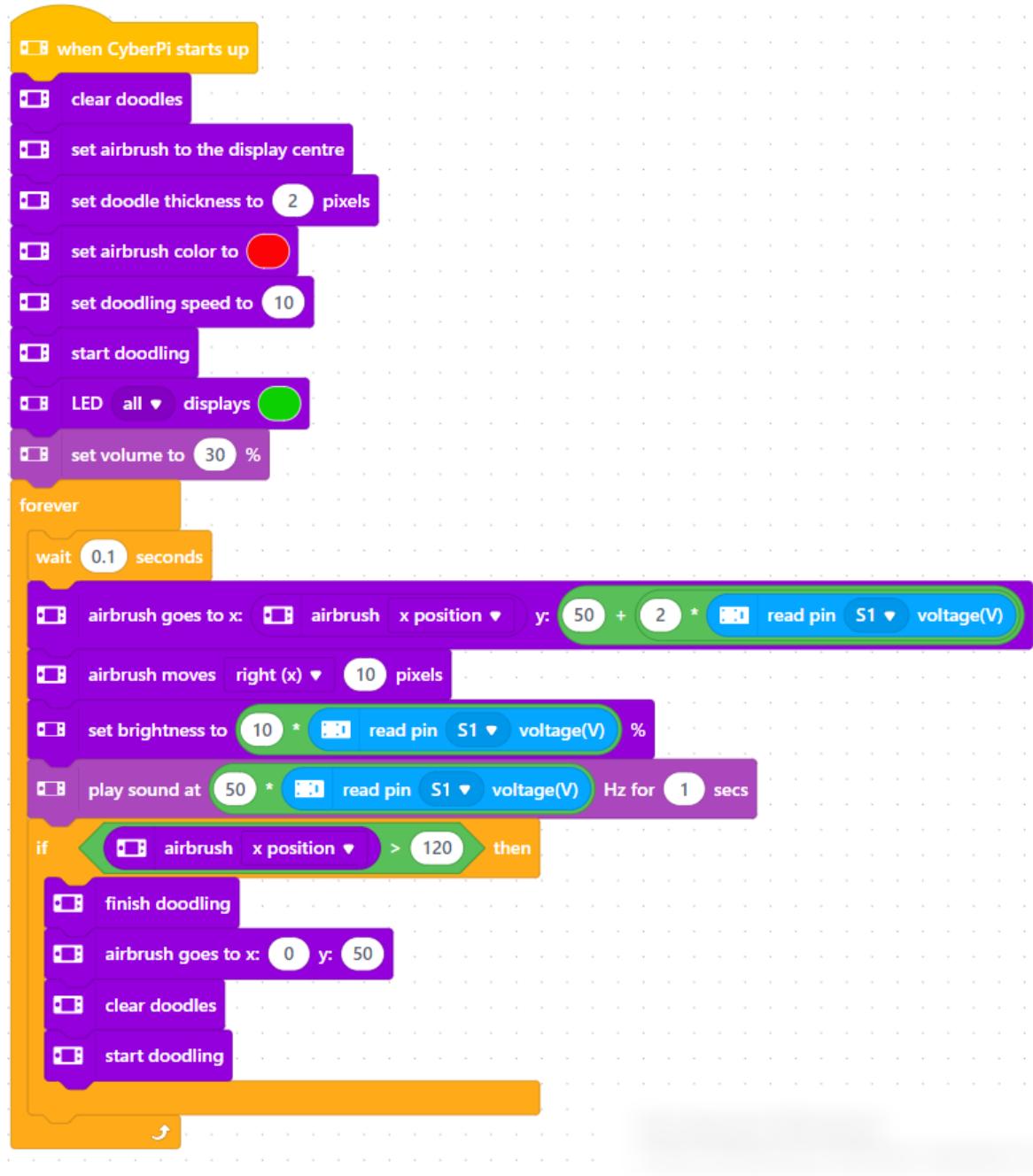


Рисунок 11 Второй вариант программы

Заключение

На основе сегодняшней работы:

- мы смогли собрать схему усилителя сигналов;
- написали программы для считывания данных и преобразования их в звук и свет.

ПРИЛОЖЕНИЕ

Разъясняющий ролик на rutube: <https://rutube.ru/video/c39066ae36bb48982cfb77f3512bded9/>

Ссылка на проект: <https://github.com/Antipat/Singing-plant>