План-коспект занятия по робототехнике.

Разработал педагог дополнительного образования

Леоненко Иван Юрьевич

МКОУ СОШ №9

Тема: “Поющий робот”.

Цель занятия – научить Arduino воспроизводить мелодию при помощи пьезоизлучателя.

Рефлексия. Где Вам могут пригодиться эти знания, ребята?

Повторение изученного: давайте вспомним, ребята, что мы с Вами **знаем о роботах и Arduino,ответив на вопросы теста .**

Тест.

Вопрос 1.

Вопрос 2.

В чем необходимо обязательно убедиться перед загрузкой программы в контроллер Arduino? *

- Выбран тип платы
- В коде созданы макроопределения
- Плата физически подключена к компьютеру
- Выбран порт, к которому подключена плата

Вопрос 3.

Для назначения режима работы пинов Arduino используется: *

- директива #define
- функция pinMode()
- функция digitalWrite()
- функция digitalRead()

Вопрос 4.

Процедура void setup() выполняется *

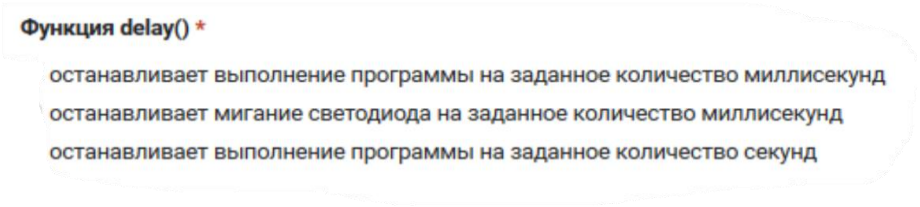
- только один раз
- один раз при включении платы Arduino
- все время, пока включена плата Arduino

Вопрос 5.

Как работает «=»? *

- Это оператор сравнения
- Это оператор присваивания, он помещает значение, расположенное справа от него, в переменную, стоящую слева
- Это оператор присваивания, он делает оба операнда равными большему из них

Вопрос 6.



Новый материал.

Теперь, чтобы продолжить наш урок, давайте познакомимся с новым способом ввода / вывода сигнала: void tone(pin, frequency, duration) – функция может принимать 2 (pin, frequency) или 3 (pin, frequency, duration) параметра. Это значит, что мы можем передать этой функции номер ножки, частоту колебаний в Герцах и длительность сигнала в миллисекундах. Если задать только pin и частоту, то сигнал будет воспроизводится до тех пор, пока мы не вызовем функцию noTone(pin).

Пробуем реализовать цель урока. Что нам потребуется ?

1. Контроллер.
2. Плата расширения контроллера .
3. Модуль звука.
4. Кабель 3pin.

Также, необходимо определиться со следующими основными значениями:

1. На какой pin будет подаваться напряжение.
2. Количество байт (под байтами мы будем понимать ноты).
3. Определиться с частотой байта и длительностью частот.

Ход занятия.

С чего начать ,чтобы записать мелодию?

Сначала пишем квалификатор const int, чтобы из наших значений мы могли сделать константу целого типа. Далее необходимо указать, куда будем подавать напряжение. Пусть это будет pin9. Таким образом, получаем первую строчку кода:

```
const int p=9;
```

Далее вводим параметр const byte (читай как байт), чтобы функция не могла изменить байт, на который она будет указывать, в нашем случае это будут ноты. Таким образом, получаем:

```
const byte COUNT_NOTES= количеству нот;
```

(сколько их будет, мы их укажем после того, как напишем частоты).

Далее указываем частоту и значение байта следующей строчкой кода – int frequencesCOUNT_NOTES] = {

Теперь мы можем записать какие угодно частоты, но я предлагаю нам записать следующие:

```
 260, 349, 349, 349, 349, 329, 392,
 260, 392, 392, 392, 392, 329, 415,415,
 261,349,349,349,349, 329, 392,
 261, 261, 293, 329, 349,
 523,523,523,466,587,
466,466,466,466,466,440,523,
 440,440,440,440,440,392,466,
 261,261,293,329,349

 };
```

Далее необходимо задать длительность частот для каждого значения частот с помощью

```
int durations[COUNT_NOTES]= {
```

Предлагаю снаписать следующие значения:

175, 175, 175, 175, 175, 175, 350,

175,175, 175, 175, 175, 175, 175, 175,

175,175, 175, 175, 175, 175,350,

175, 175,175, 175,700,

350,350,175,175,350,

175,175,175,175,175,175,350,

175,175,175,175,175,175,350,

175,175,175,175,700};

Теперь запишем подпрограмму , которая в нашем случае не возвращает значения, так как мы их и не зададим

void setup() {.

Запишем функцию, которая будет возвращать наши значения

pinMode(p, OUTPUT);

Теперь снова запишем подпрограмму, которая не возвращает значения.

}void loop() {

Теперь, ребята, я предлагаю сделать наш цикл не прекращающимся, пока один из его элементов не станет нулем. Т.е. не прекращающимся от нуля до количества нот.

Пишем строчку int i=0;

Конкретизируем и пишем

for (i = 0; i< COUNT_NOTES; i++){

Давайте в целях эксперимента сделаем повторение для частот и их длительности в нашем цикле.

tone(p, frequences[i], durations[i]*2);

delay(durations[i]*2);

Чтобы нам завершить воспроизведение мелодии в цикле просто пишем noTone и указываем пин (p);

```
 noTone(p);
   }
}
```