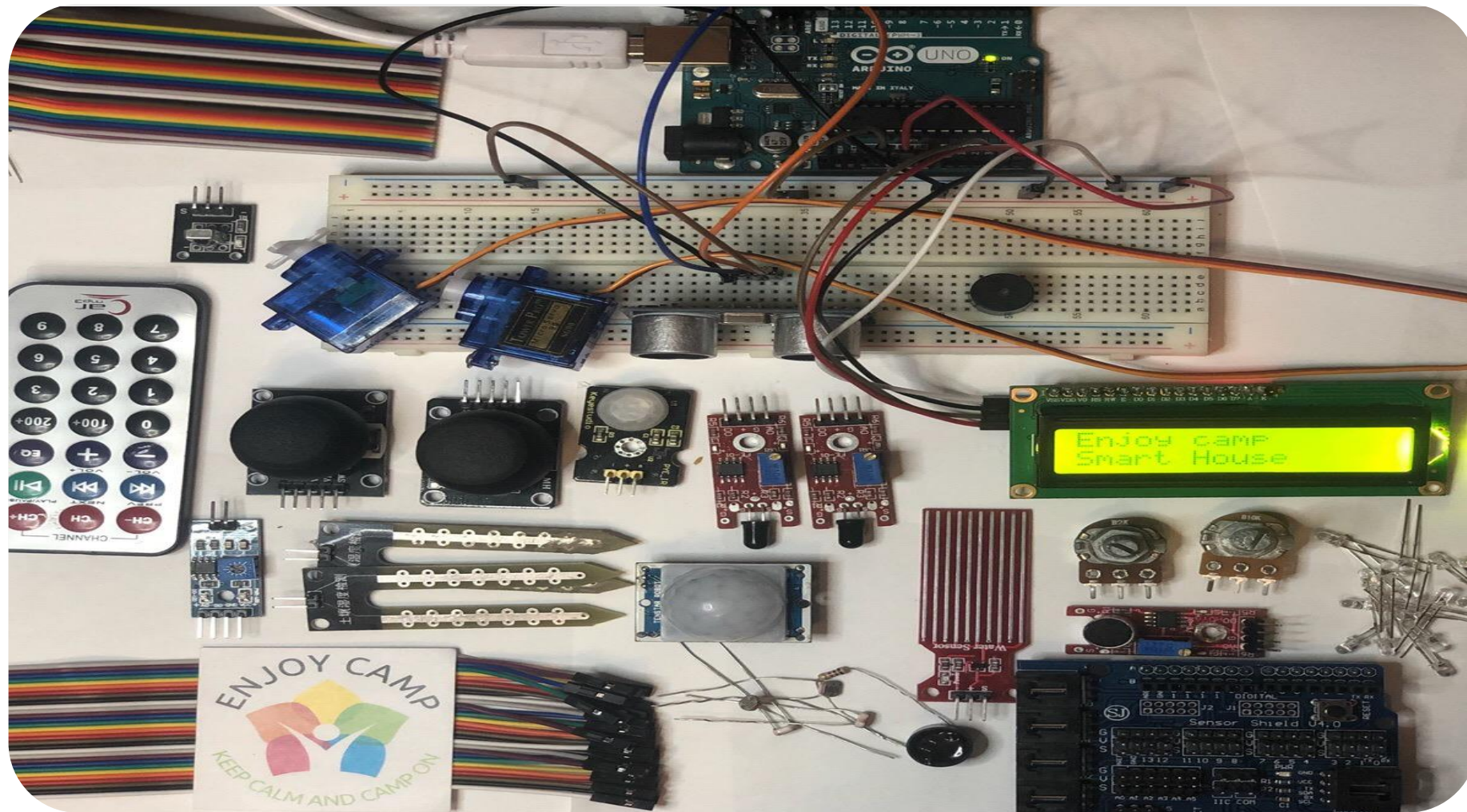




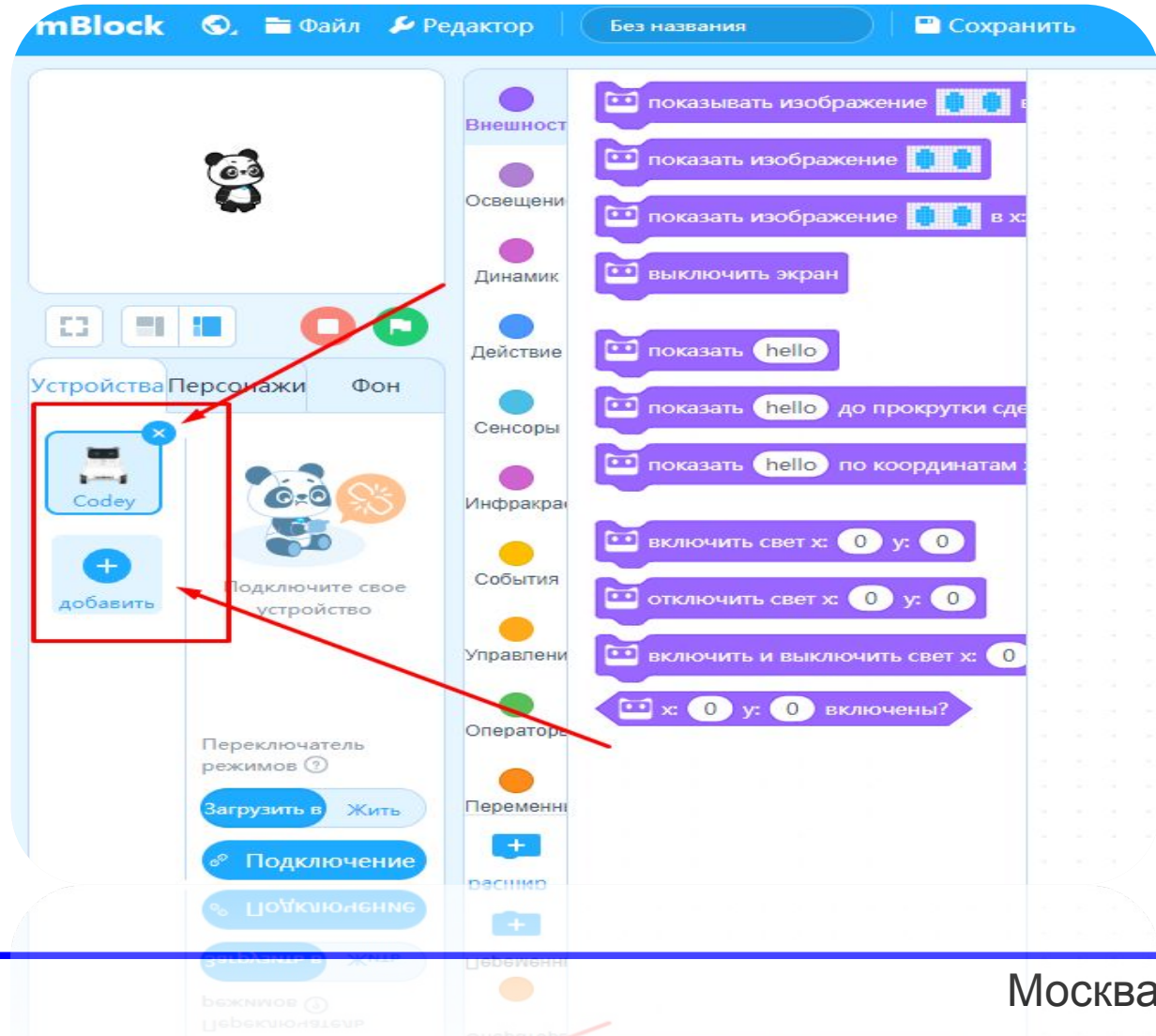
Датчики умного дома на базе Arduino Uno и mBlock

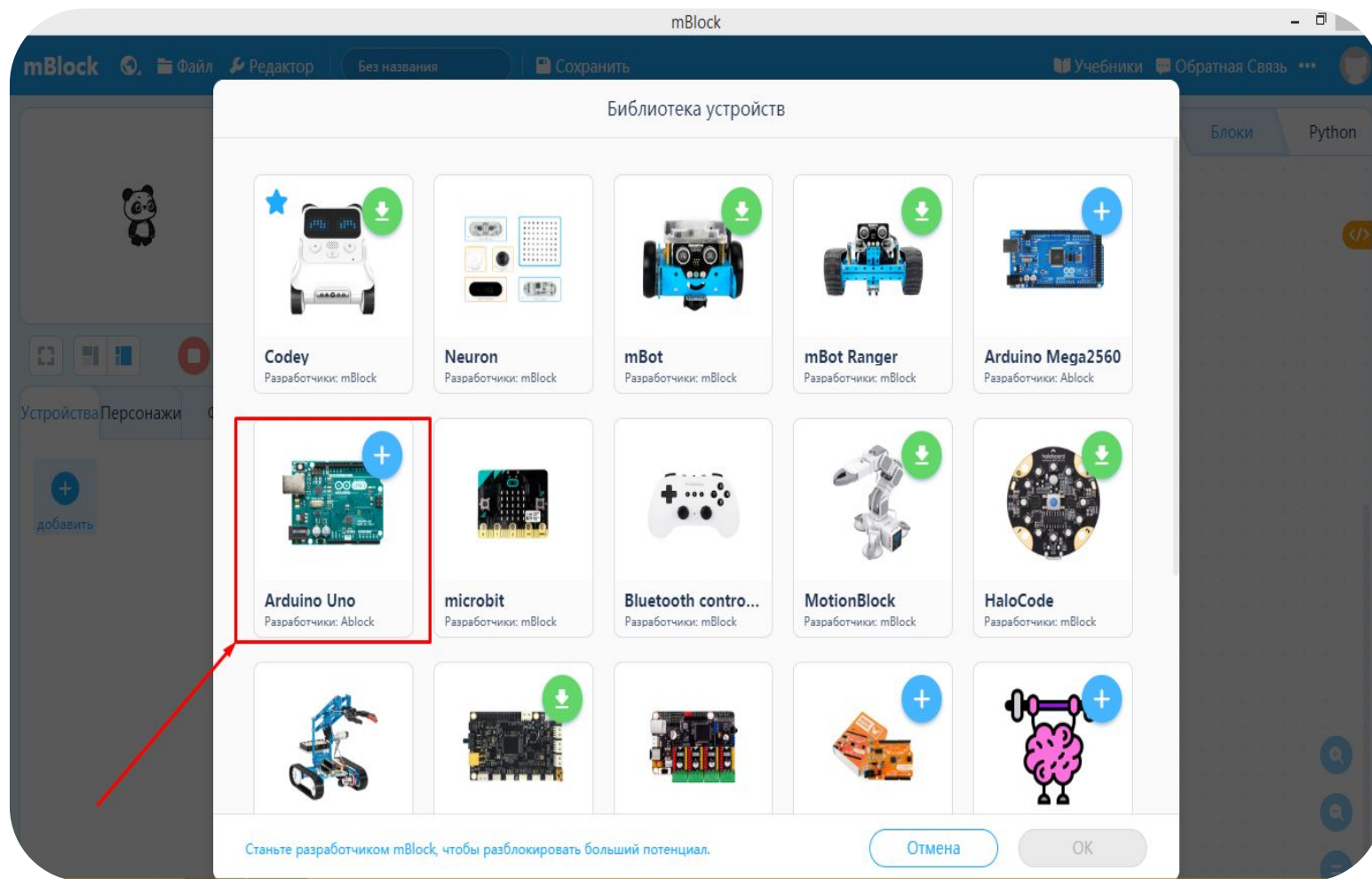




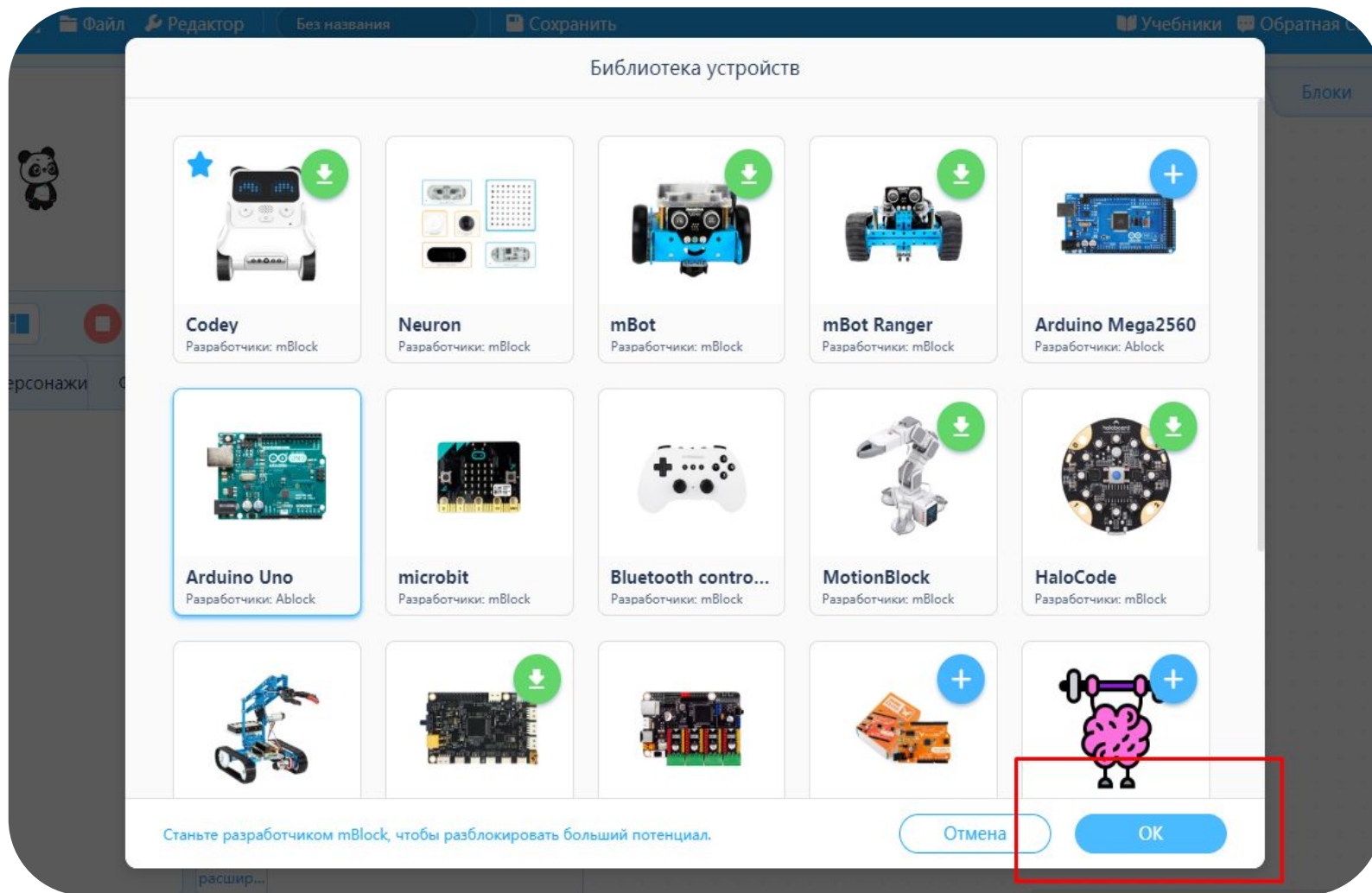
Выберем
вкладку
устройства,
удалив Codey

Подключим
Arduino к
компьютеру
кабелем
USB-B

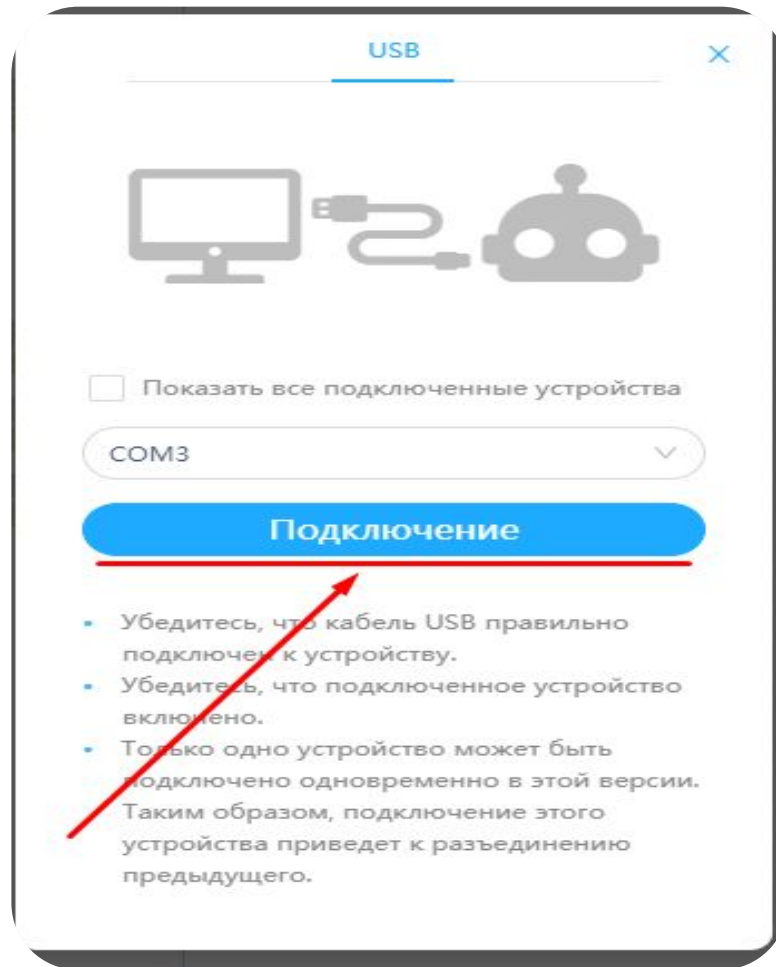
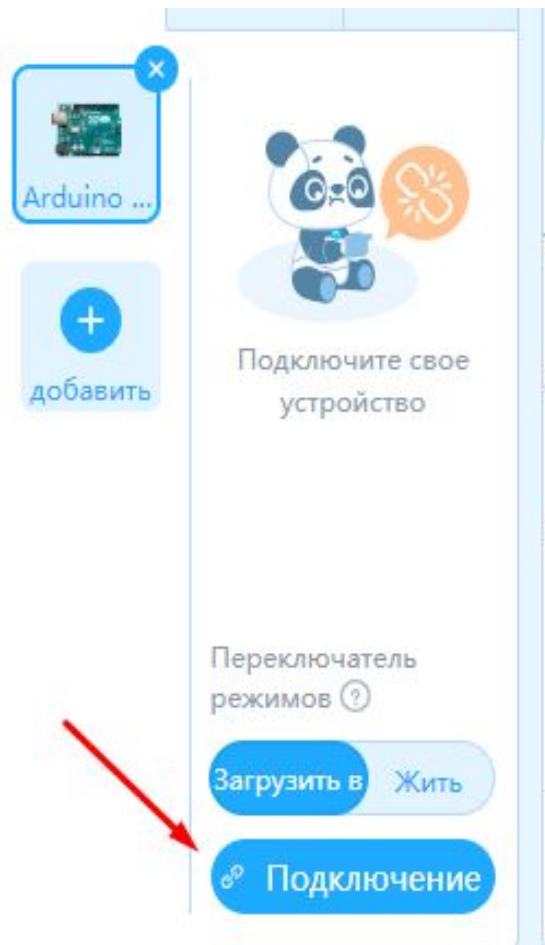




Добавим в
библиотеку
устройств
Arduino UNO



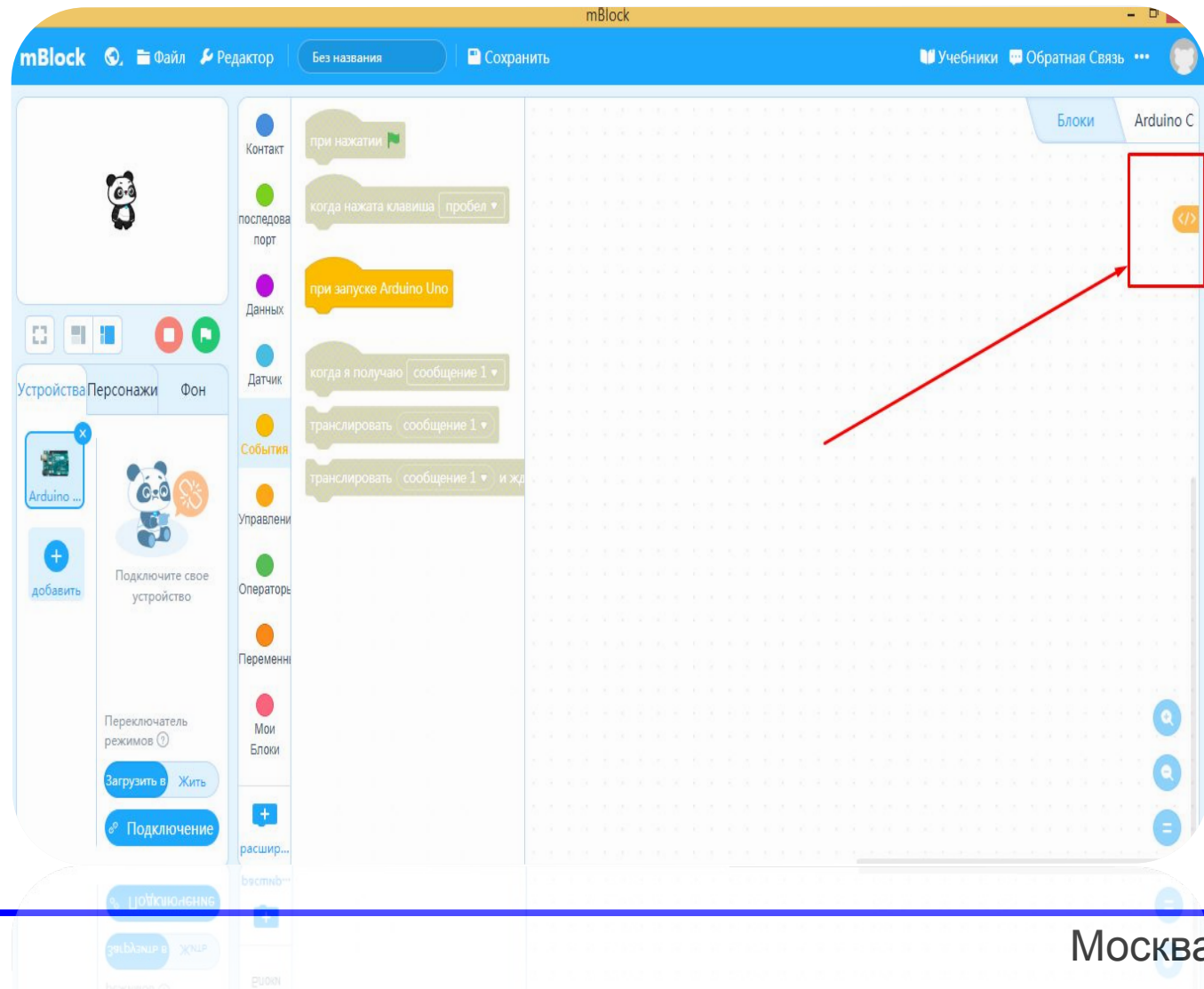
Подтвердим
нажав
клавишу ОК



Соединим по
последовательному
порту
микроконтроллер с
компьютером
Для каждой Arduino
COM может быть с
разной цифрой (не
пугайтесь если у вас
не COM3)



Отлично!
Теперь мы можем
использовать
команды,
функции, циклы,
библиотеки и
различные
методы,
позволяющие
программировать
Ардуино.





mBlock

Файл Редактор Без названия Сохранить

Устройства Персонажи Фон

Arduino ...

добавить

Подключите свое устройство

Переключатель режимов ?

Загрузить в Жить

Подключение

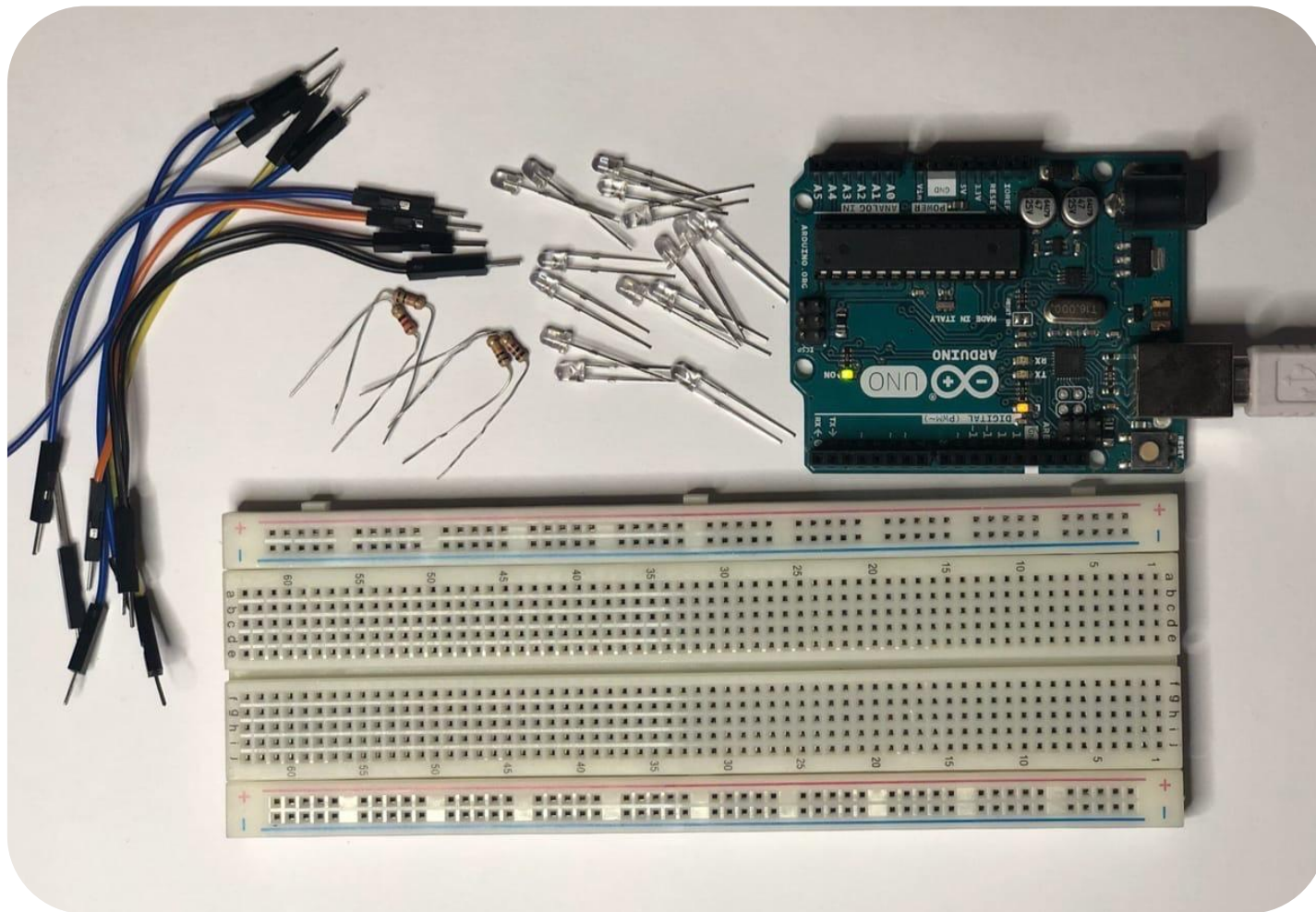
Скрыть окно подробного кода

Подробный код C++

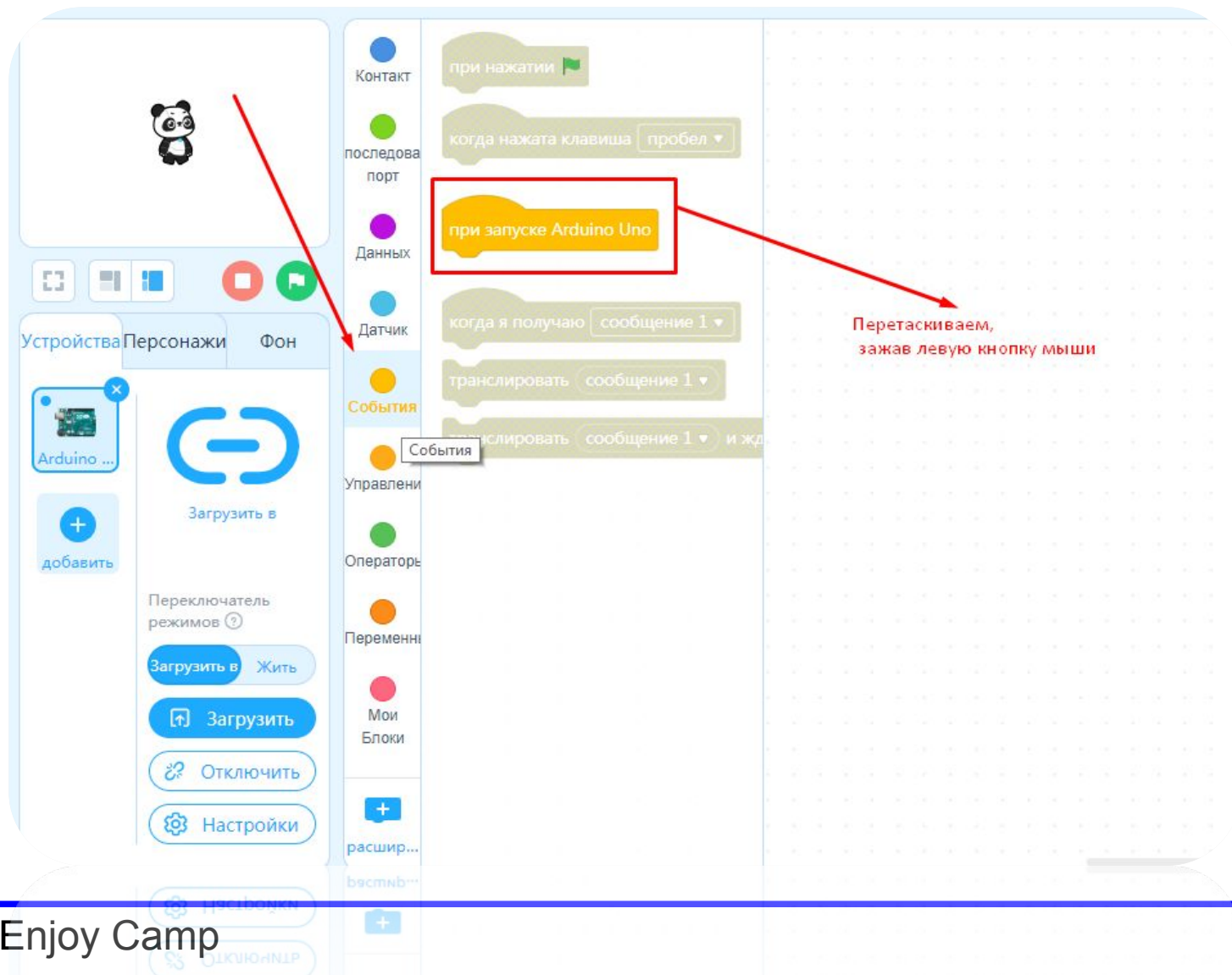
Arduino C

```
1 // generated by mBlock5 for <your product>
2 // codes make you happy
3
4 void _delay(float seconds) {
5     long endTime = millis() + seconds * 1000;
6     while(millis() < endTime) _loop();
7 }
8
9 void setup() {
10 }
11
12 void _loop() {
13 }
14
15 void loop() {
16     _loop();
17 }
```

Нажав на
символы `</>`
можно увидеть
подробный код
на языке C++

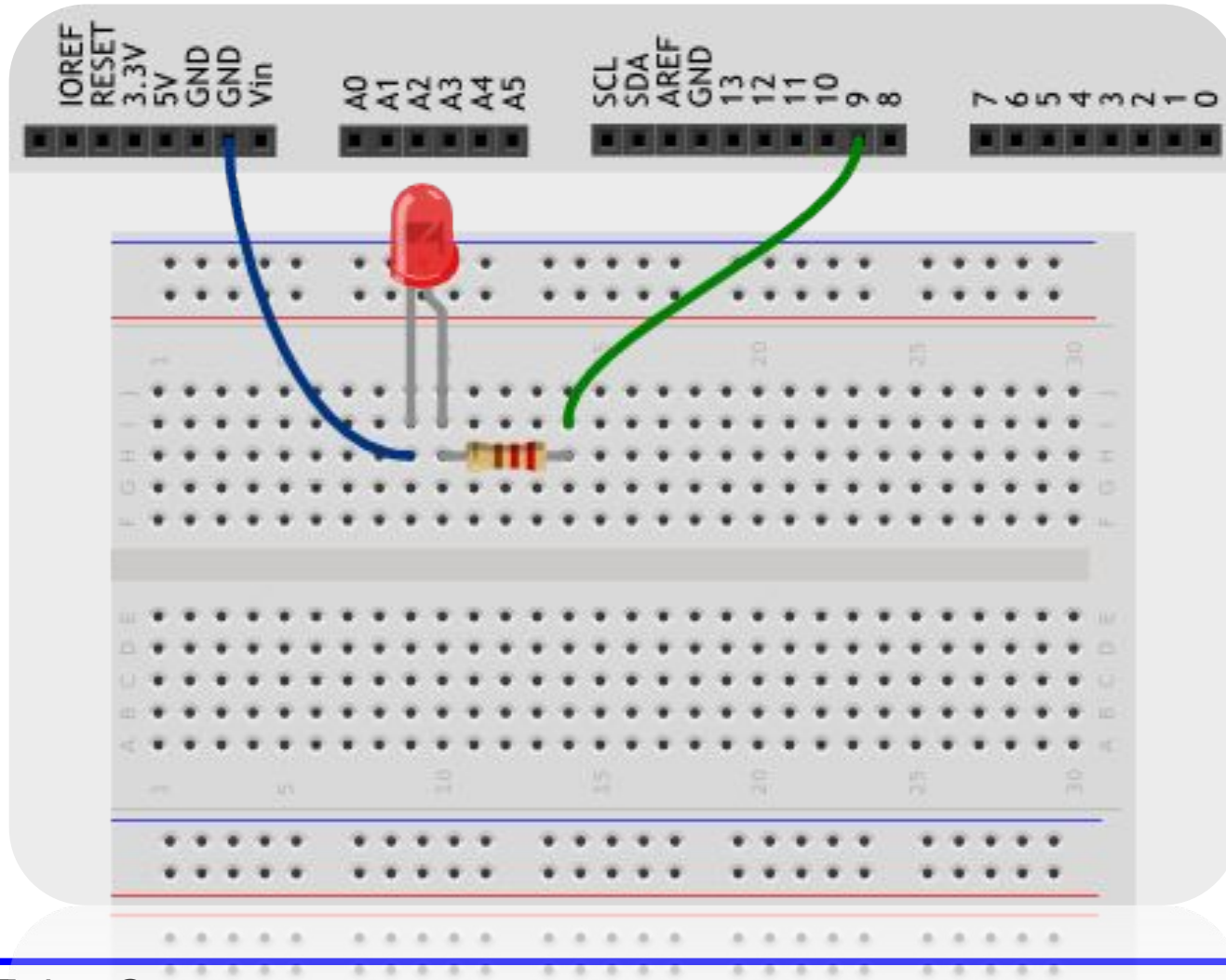


Для начала напишем программу, которая будет управлять светодиодами. Для этого нам понадобятся провода(папа-папа), светодиоды и резисторы и конечно, макетная плата, которую мы будем использовать для создания на ней наших схем



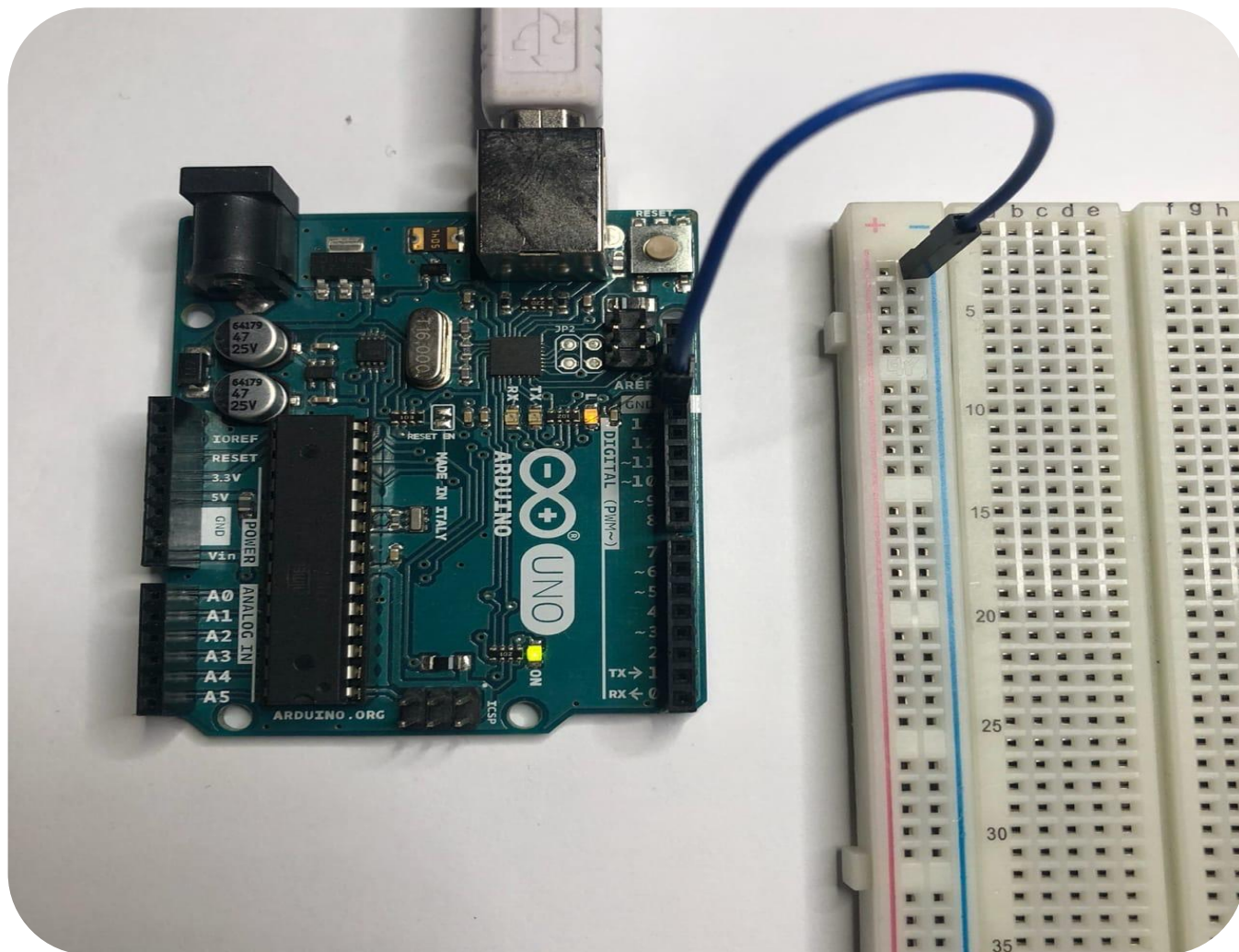
Перетаскиваем,
зажав левую кнопку мыши

В меню блоков
выберем «События»
и перетащим в
рабочее
пространство блок
«При запуске
Arduino Uno»
Этот блок будет
практически всегда
первым, нужен он
для запуска работы
программы

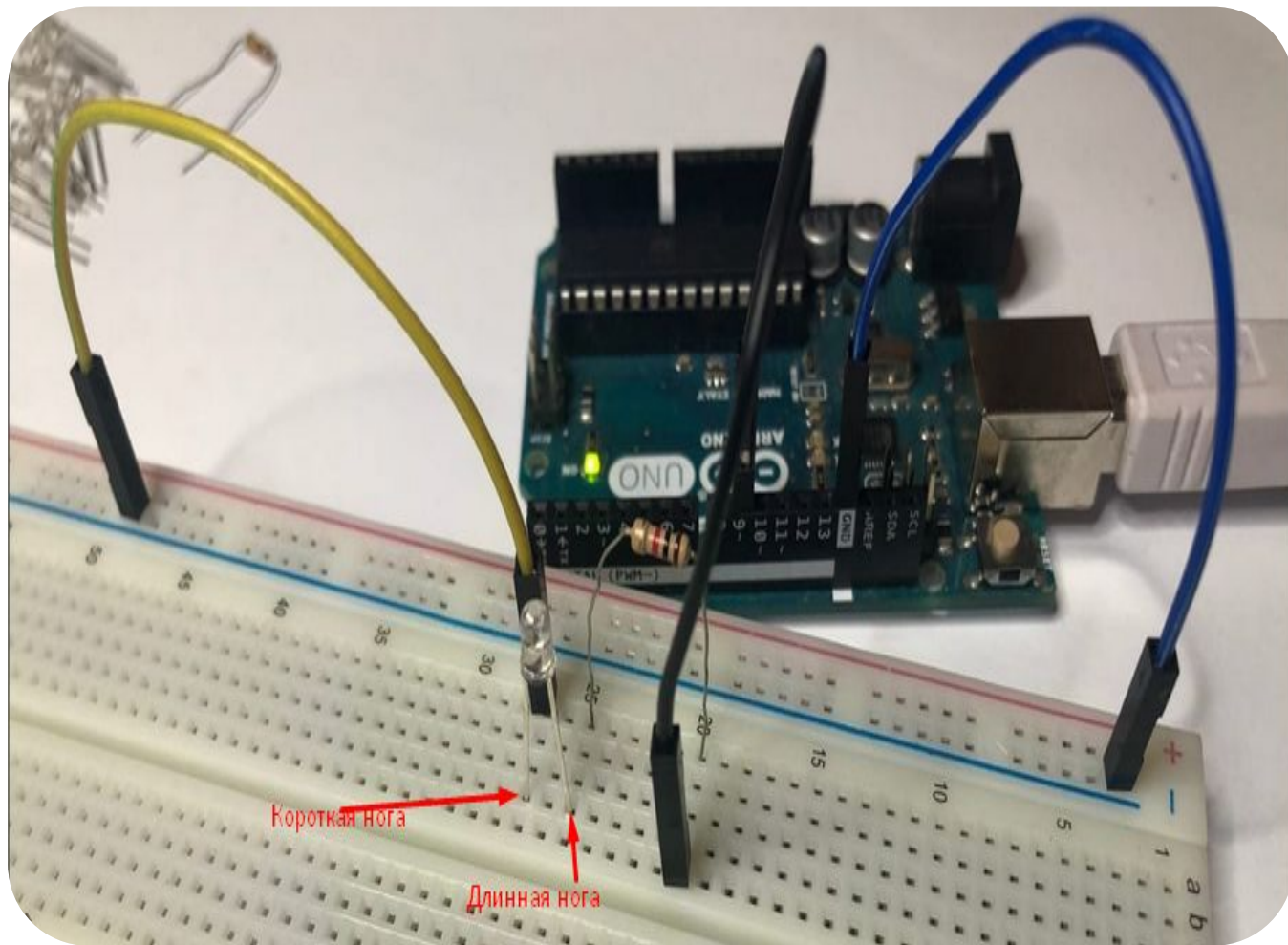


Соберем схему позволяющую использовать светодиоды. Для этого соединим одну ногу резистора с 9 Портом Arduino (PIN), а вторую с длинной ногой светодиода. Свободную ногу светодиода стоит соединить с «Землей» платы (GND)

Схему можно собрать в любом удобном для вас месте на макетке

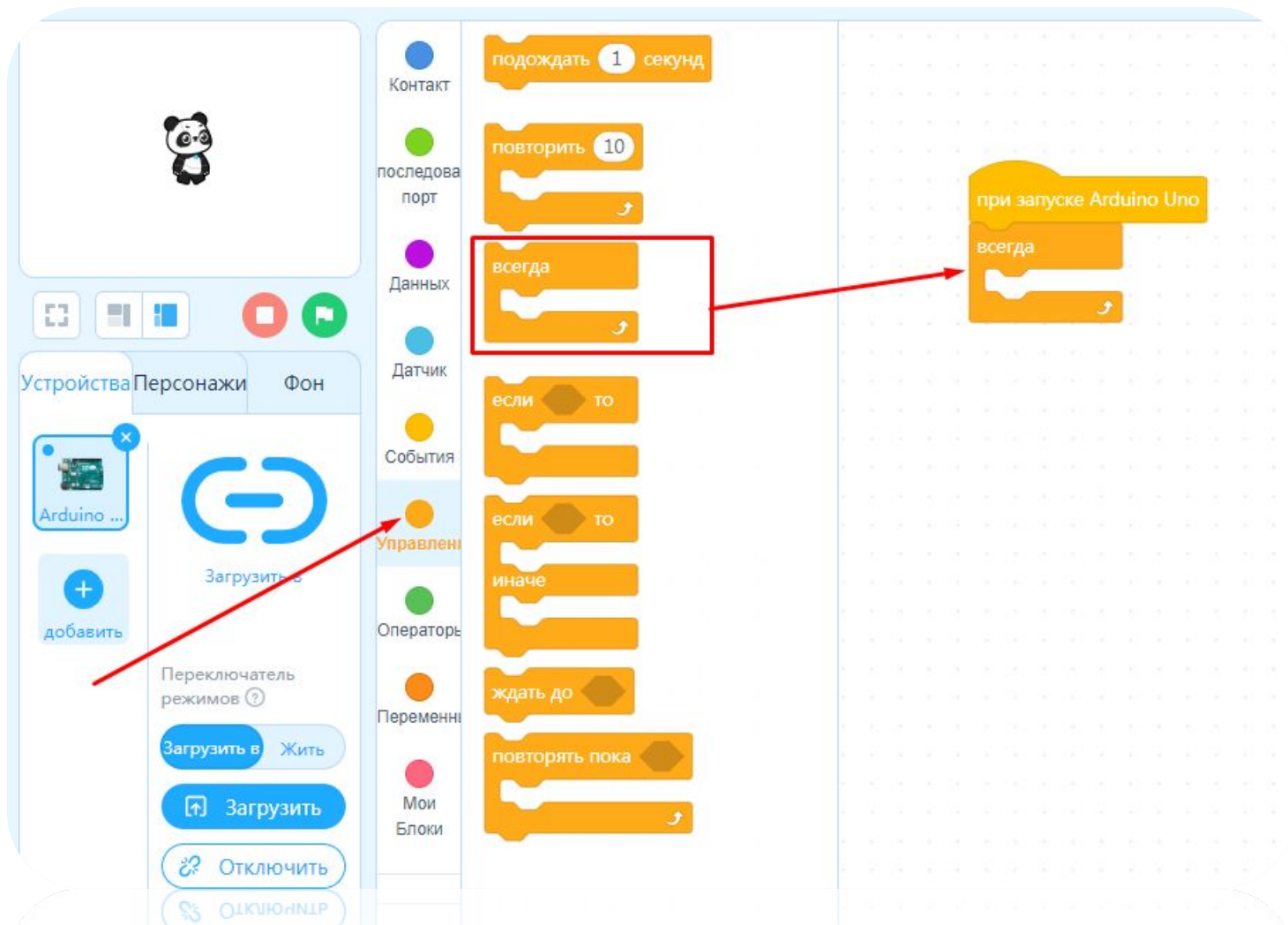


С порта GND выведем землю на длинную минусовую полосу макетной платы. Нужно это для того, что бы обеспечить землей все наши датчики. Их может быть много, а на плате всего 3 GND



Важно объединять
«ноги» радиоэлементов
в рядах с одинаковыми
номерами.

Конечно, после сборки
схемы светодиод не
будет светиться без
специальной программы,
которую мы сначала
напишем, а потом
загрузим в Arduino

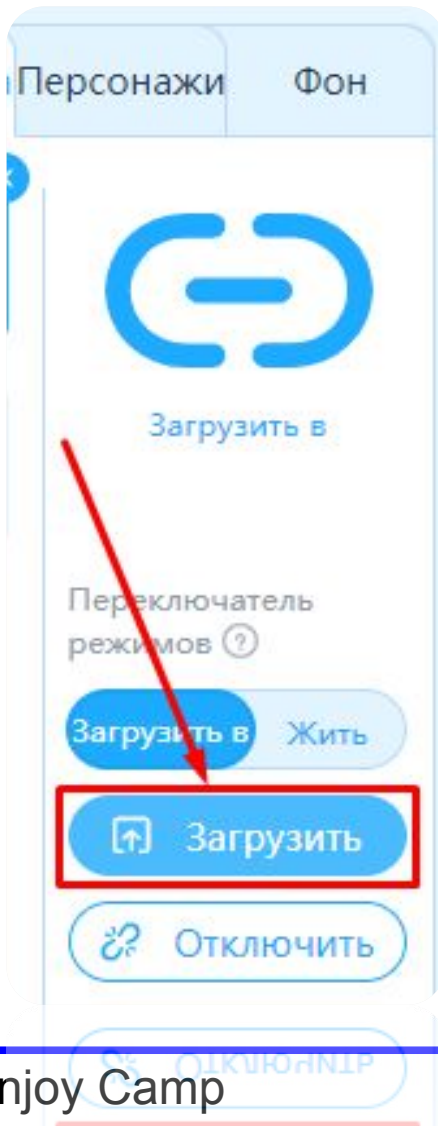


Прицепляем блок
«Всегда» из
вкладки блоков
Управление к
блоку который мы
разместили на
рабочей плоскости
кода ранее



Объединим блоки
«Всегда» и
«установить выход
цифрового порта...
как...»

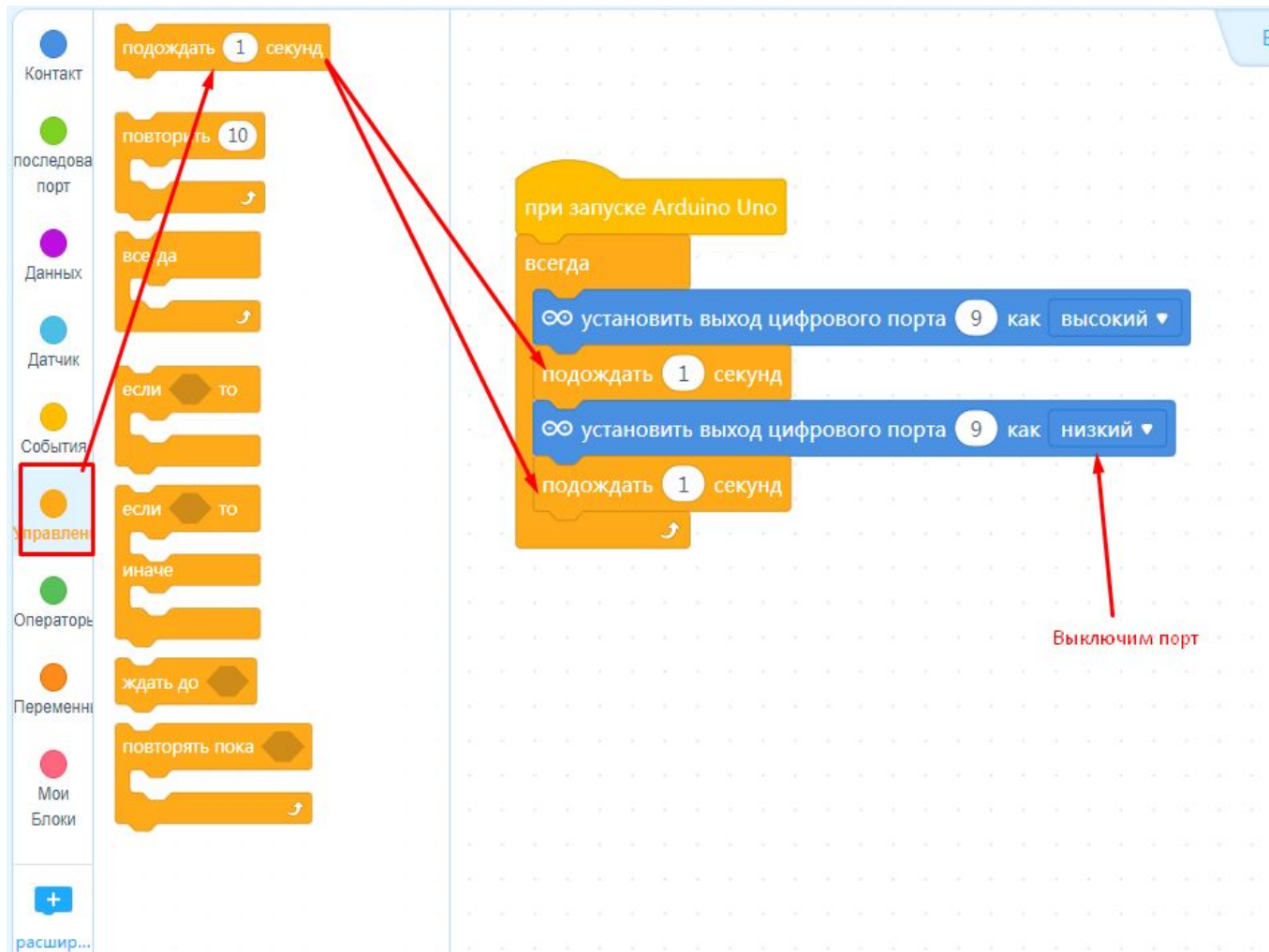
Т.к. наш светодиод
подключен к 9
цифровому порту
Arduino оставим цифру
9. Что бы светодиод
загорелся нужно
выбрать «Высокий», а
что бы погас –
«Низкий»



Для загрузки нашей волшебной программы
по зажиганию светодиода нажмем
«Загрузить»

Появится шкала, отвечающая за прогресс
загрузки, и по ее окончании – магия
программирования сработает и наш
светодиод засветится!

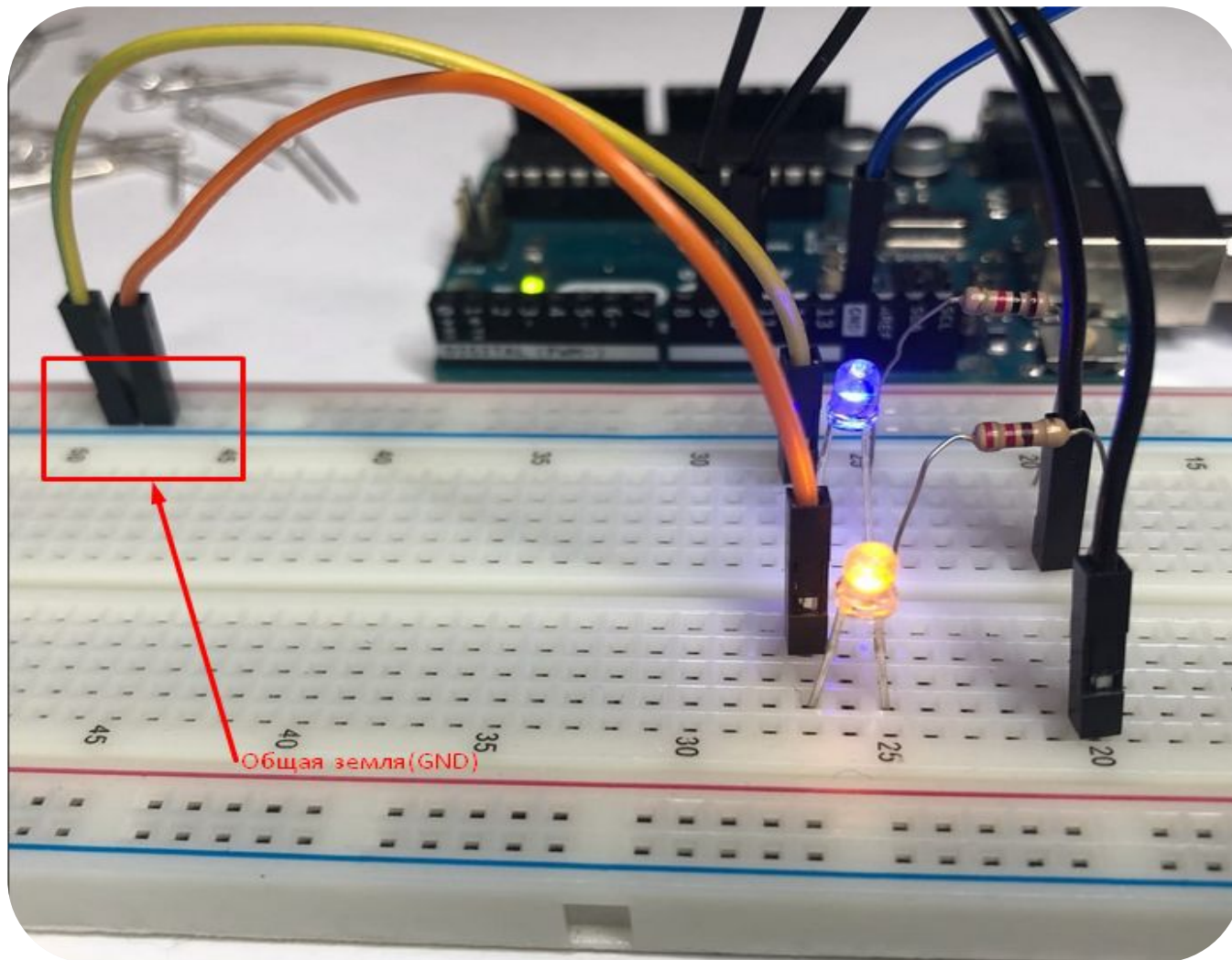
Поздравляю! Мы с Вами сделали
маленький, но очень серьезный шаг в
программировании Arduino датчиков умного
дома mBlock



Следующим шагом,
будет заставить наш
светодиод мигать. Это
мигание в электронике
называется
«Маячком»

Добавив несколько
блоков кода, добьемся
мигания светодиода.

Кстати, меняя
цифровку 1 в строчках
подождать можно
изменить частоту
мигания!



Усложним задачу, собрав абсолютно такую же схему, но теперь подключим к 10 порту А второй светодиод.

Конечно порт может быть любой от 1 до 13, но в начале Вашего пути программиста рекомендуется использовать 10 порт



при запуске Arduino Uno

всегда

установить выход цифрового порта 9 как высокий ▼

установить выход цифрового порта 10 как высокий ▼

подождать 1 секунд

установить выход цифрового порта 9 как низкий ▼

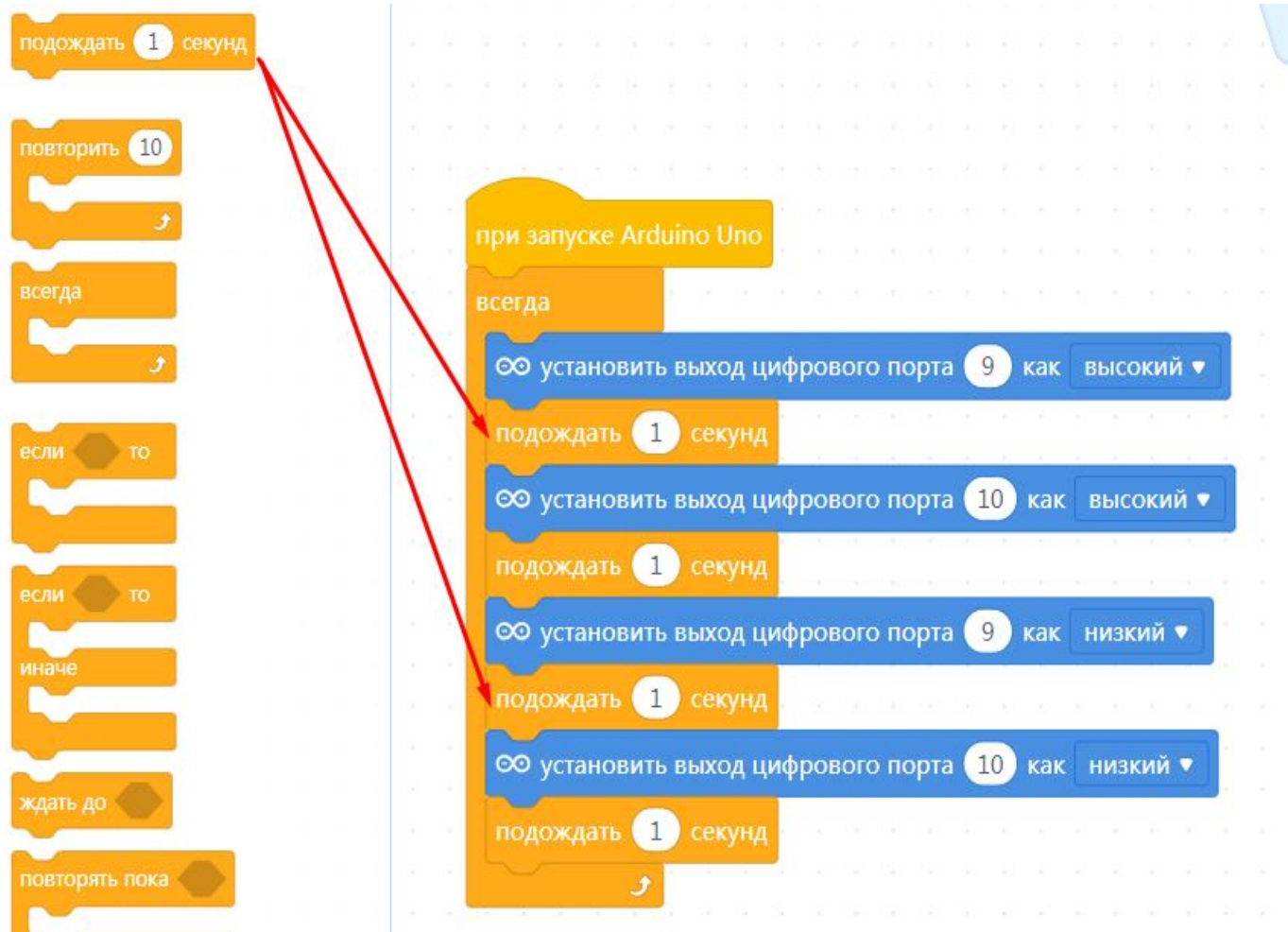
установить выход цифрового порта 10 как низкий ▼

подождать 1 секунд

Также добавим
необходимые блоки для
нашего второго светодиода,
вы уже их знаете

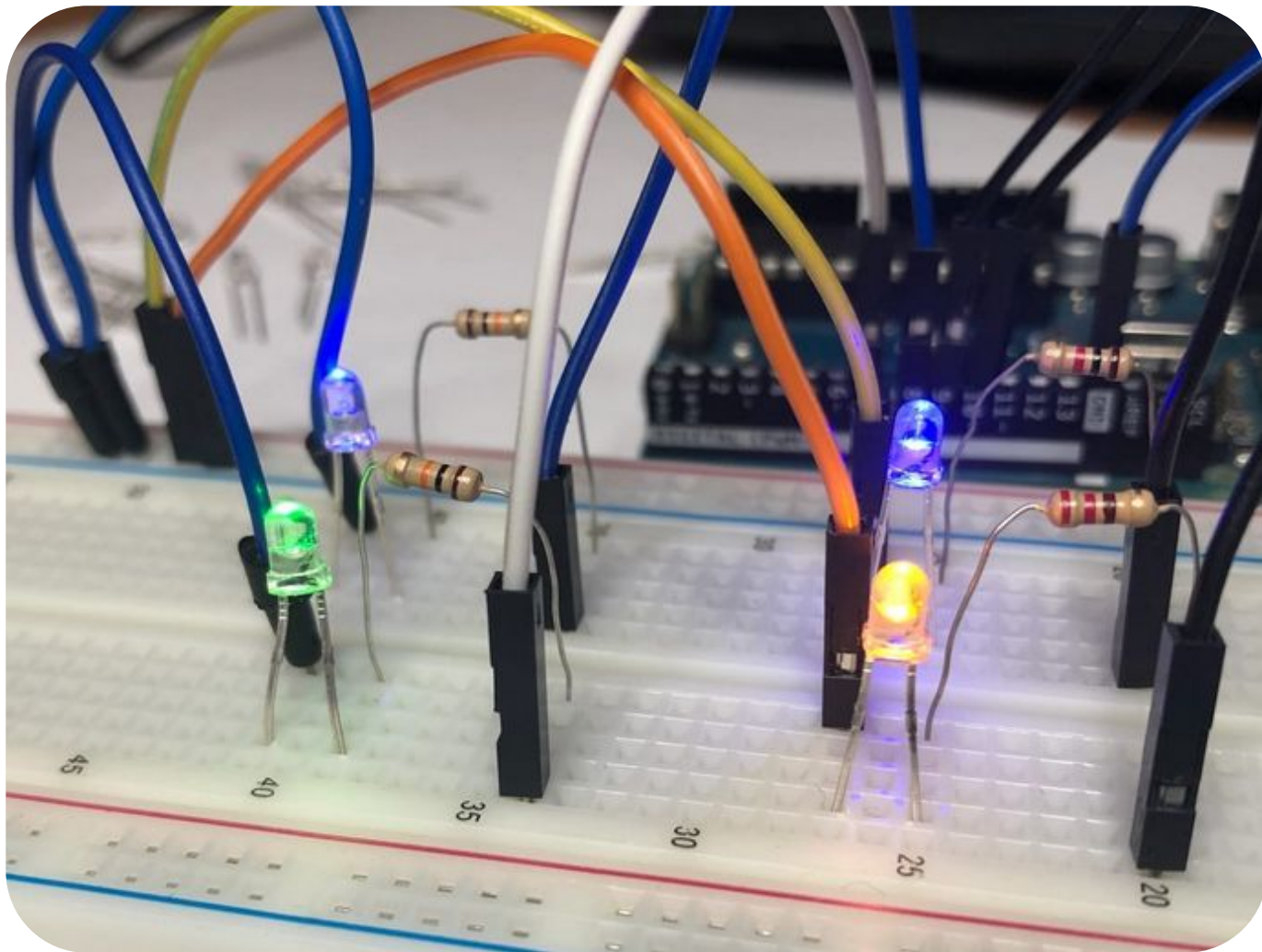
Когда светодиоды замигали,
поменяйте цифры «1» в
блоке «Подождать» сначала
на 0.25, а потом на 5

Что изменилось???



Синхронное мигание светодиодов довольно интересная затея, но добавив еще два блока «Подождать» в нужные места, можно заставить светодиоды светиться поочерёдно.

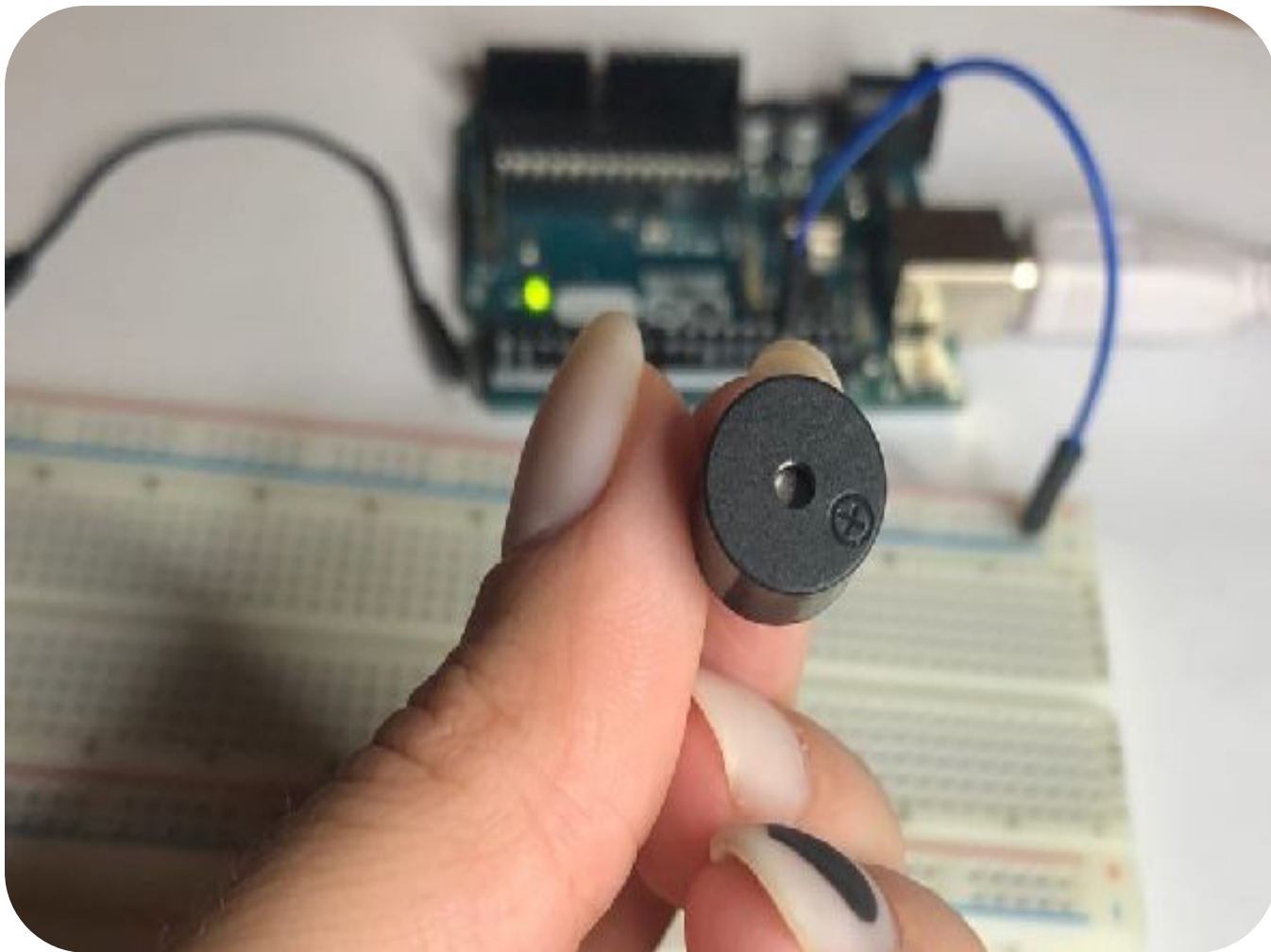
Попробуйте!!!



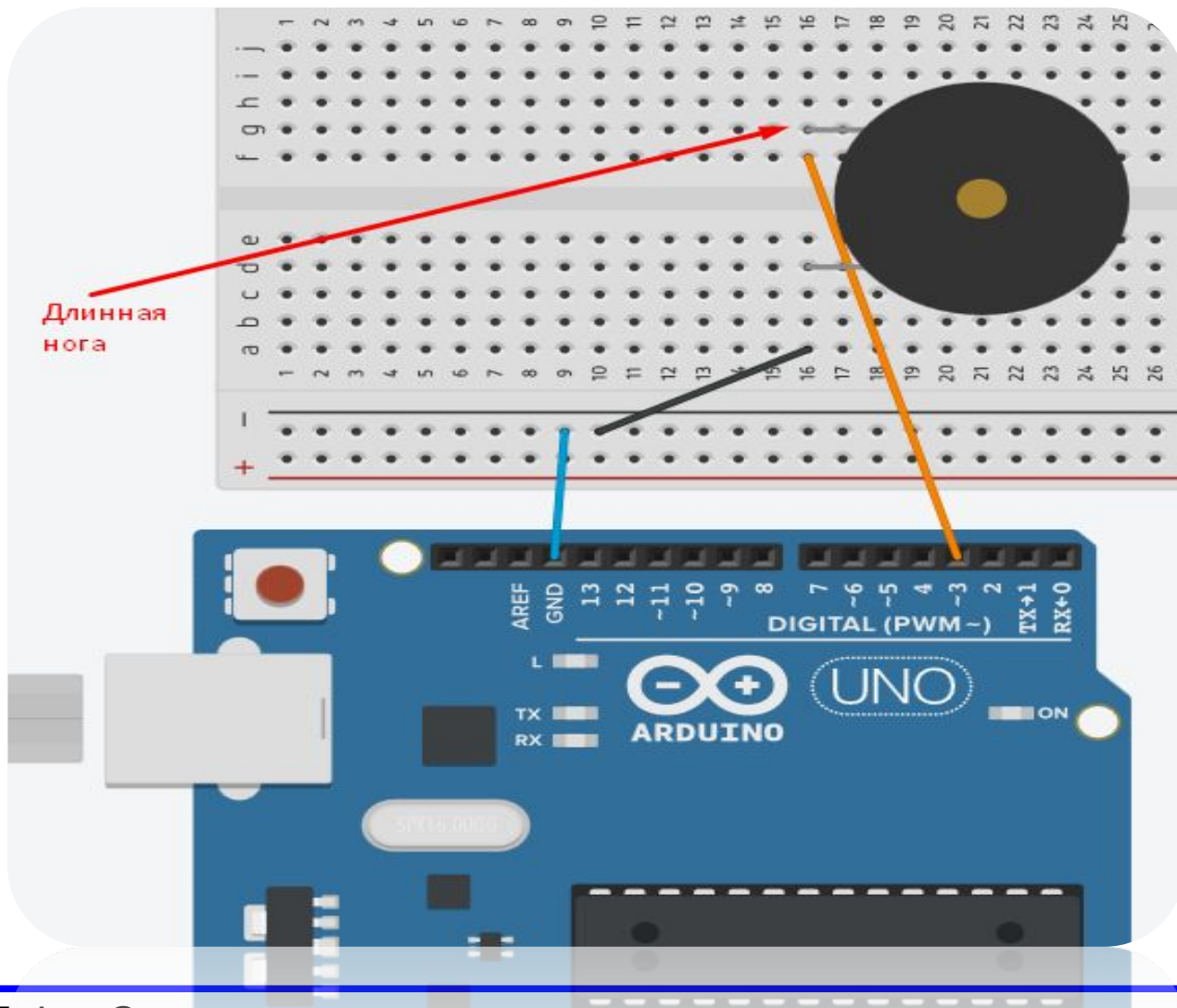
Поздравляю! Вы практически спец по светодиодам. Специальное задание – подключить светодиоды к портам 7,8,9,10 и настроить их мигание как Вам захочется! После этого можно переходить к подключению нового интересного датчика умного дома – «МЕРЗКОЙ ПИЩАЛКИ»



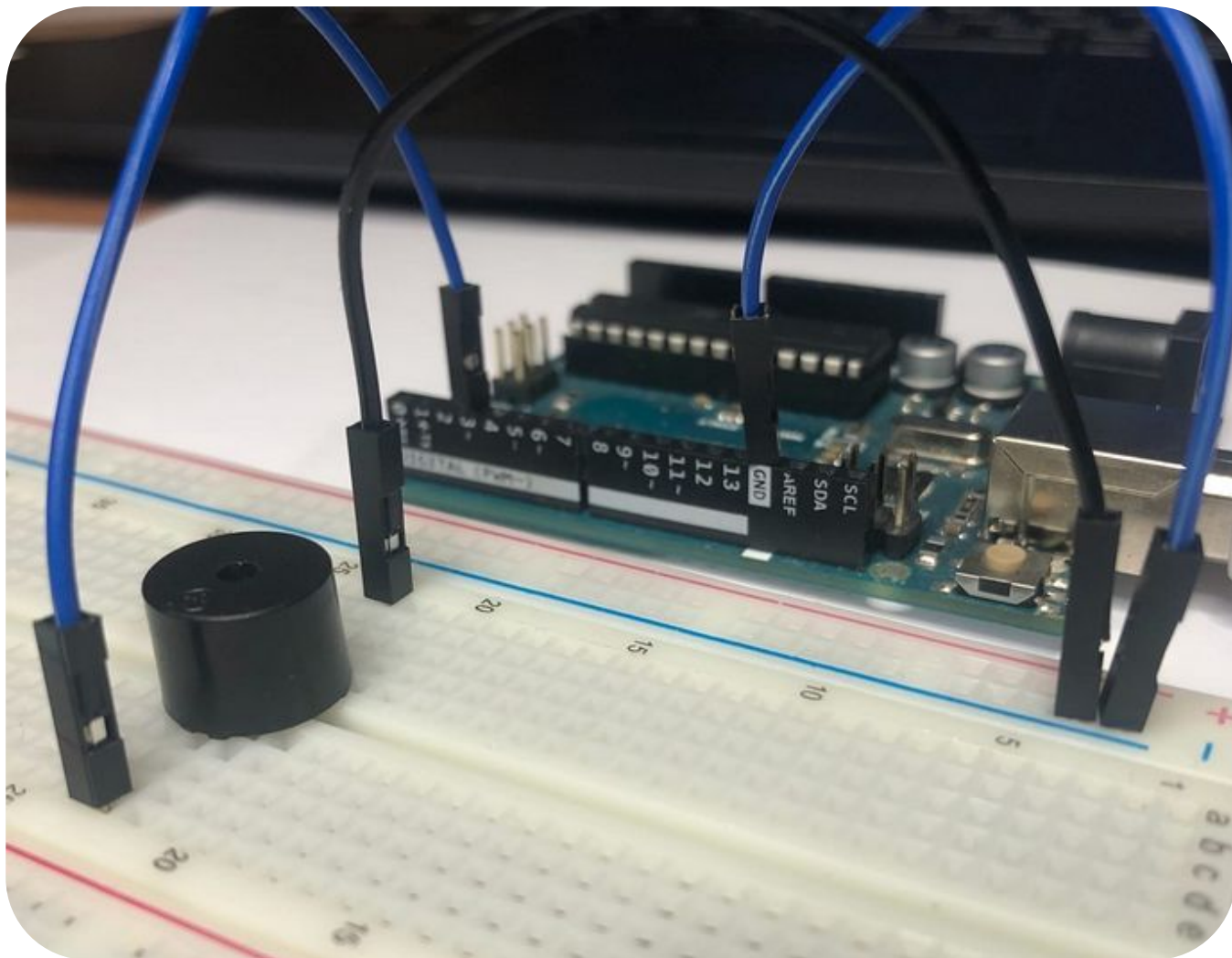
Пьезо-пищалка — звуковая сигнализация



Конечно звуковые
сигналы – очень
важная часть
умного дома.
Что бы выполнить
подключение
пищалки
понадобится
макетная плата и
всего пара
проводов



Следует
поставить
пищалку по
центру макетной
платы, подключив
длинную ногу в
цифровому
выходу **pin 3**, а
свободную ногу
просто к земле
GND



После сборки схемы
попробуем
заставить пищалку
издать один
единственный звук

Конечно, существует
огромное
количество песен и
мелодий, которые
наша «Мерзкая
пищалочка» может
сыграть



при запуске Arduino Uno

всегда

∞	порт	3	проигрывает ноту	C2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	D2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	E2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	F2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	G2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	A2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	B2 ▼	0.25	долей
∞	порт	3	проигрывает ноту	C3 ▼	0.25	долей

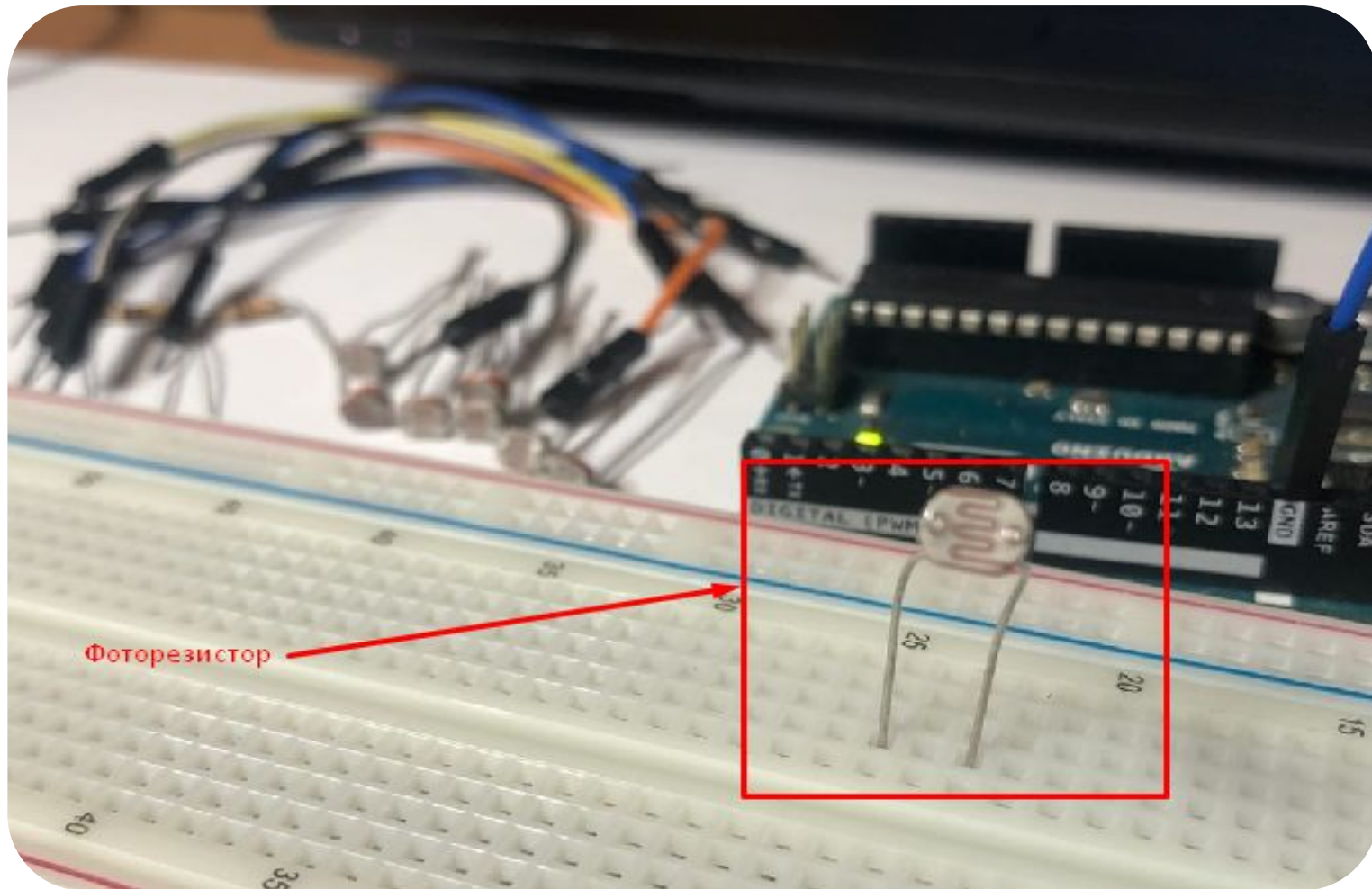
Теперь проиграем полную октаву

Подключение пищалки дело простое, поэтому попробуйте сделать свою мелодию!

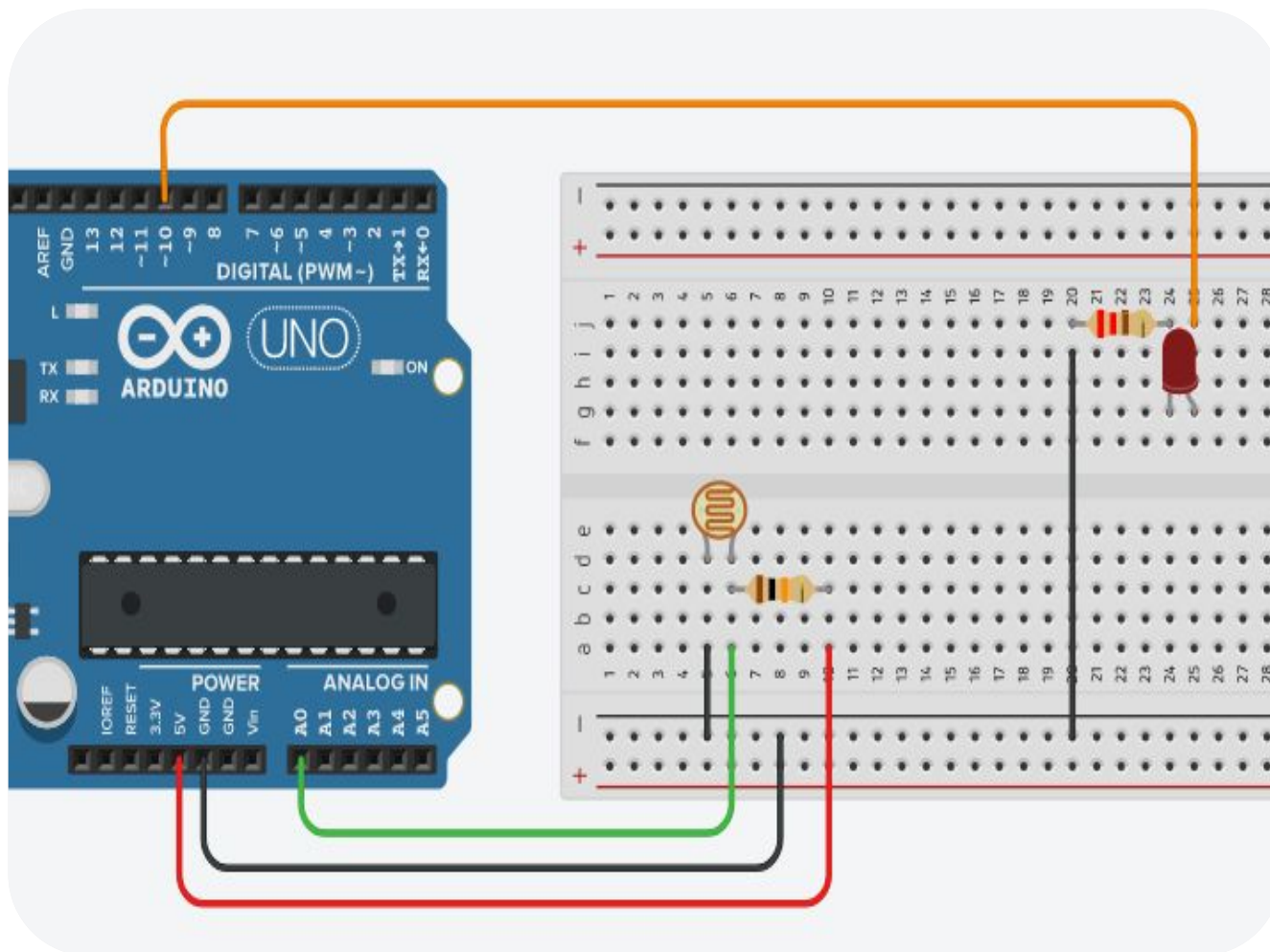
Далее, нам с Вами предстоит при помощи фоторезистора и пищалки сделать свой музыкальный инструмент, реагирующий на свет и тень - Терменвокс



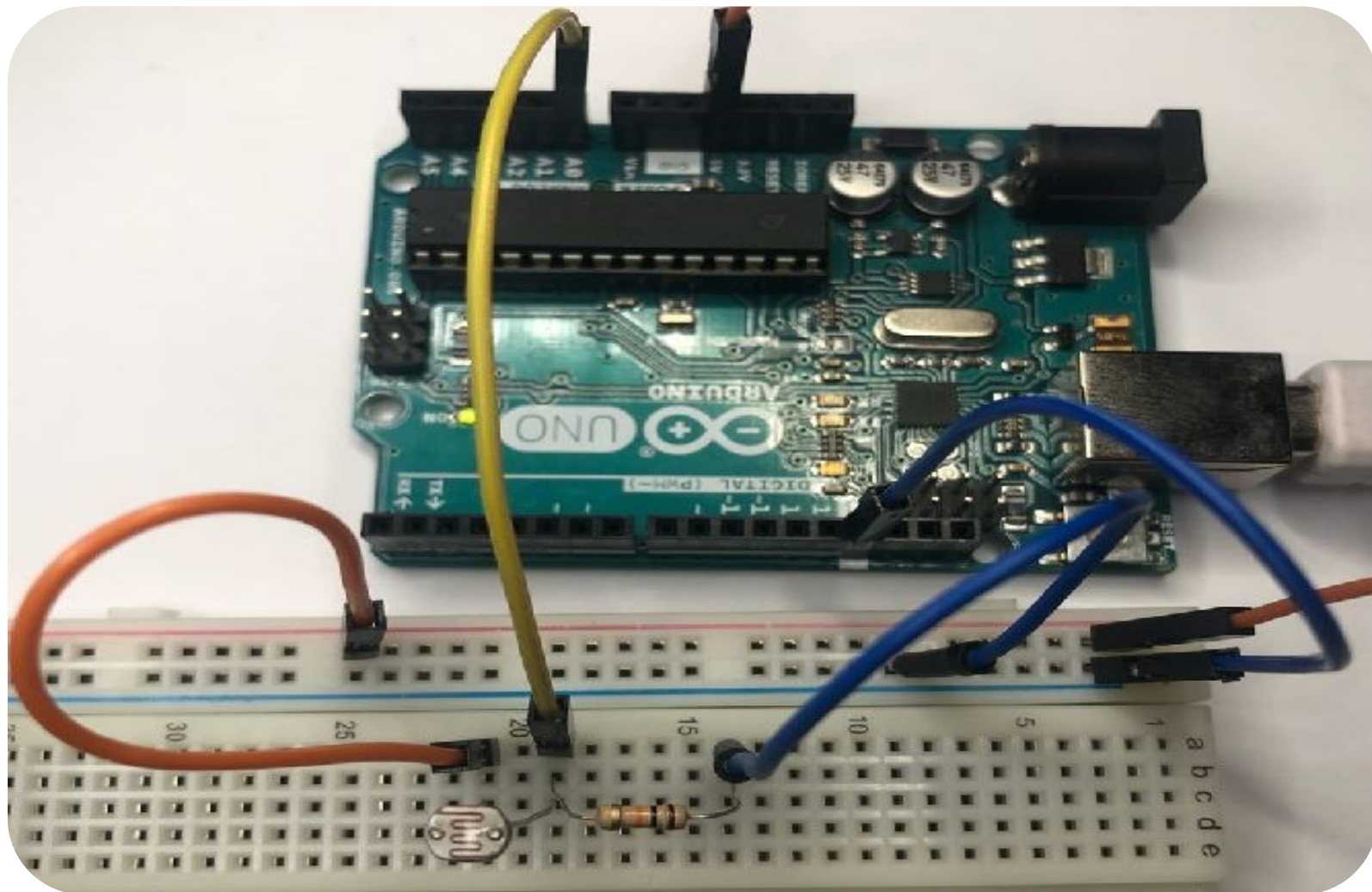
Фоторезистор — датчик света



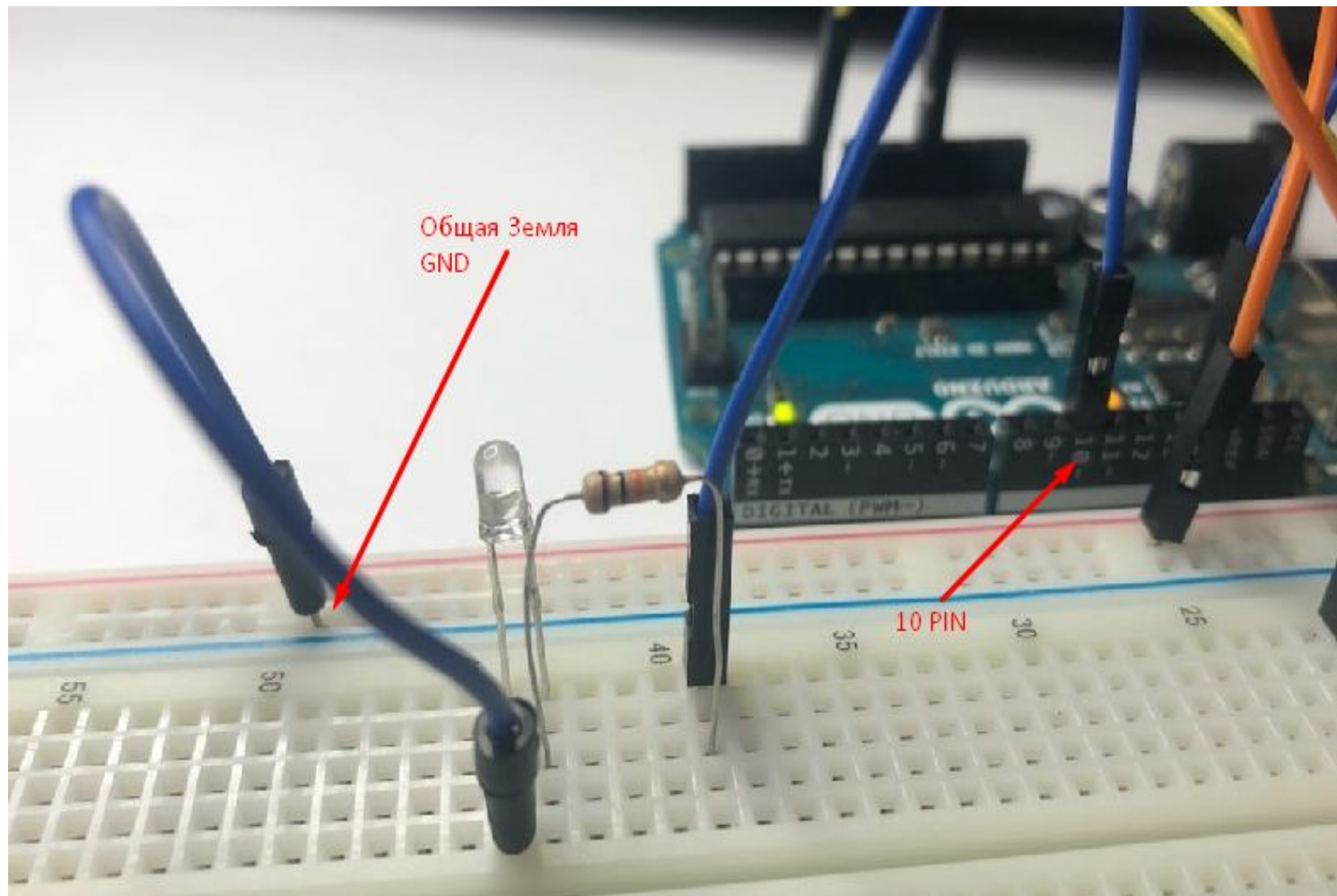
Фоторезистор
нужен для
считывания
уровня
количества
света, причем в
отличии от
светодиода ему
не важно где
правая нога, а
где левая нога



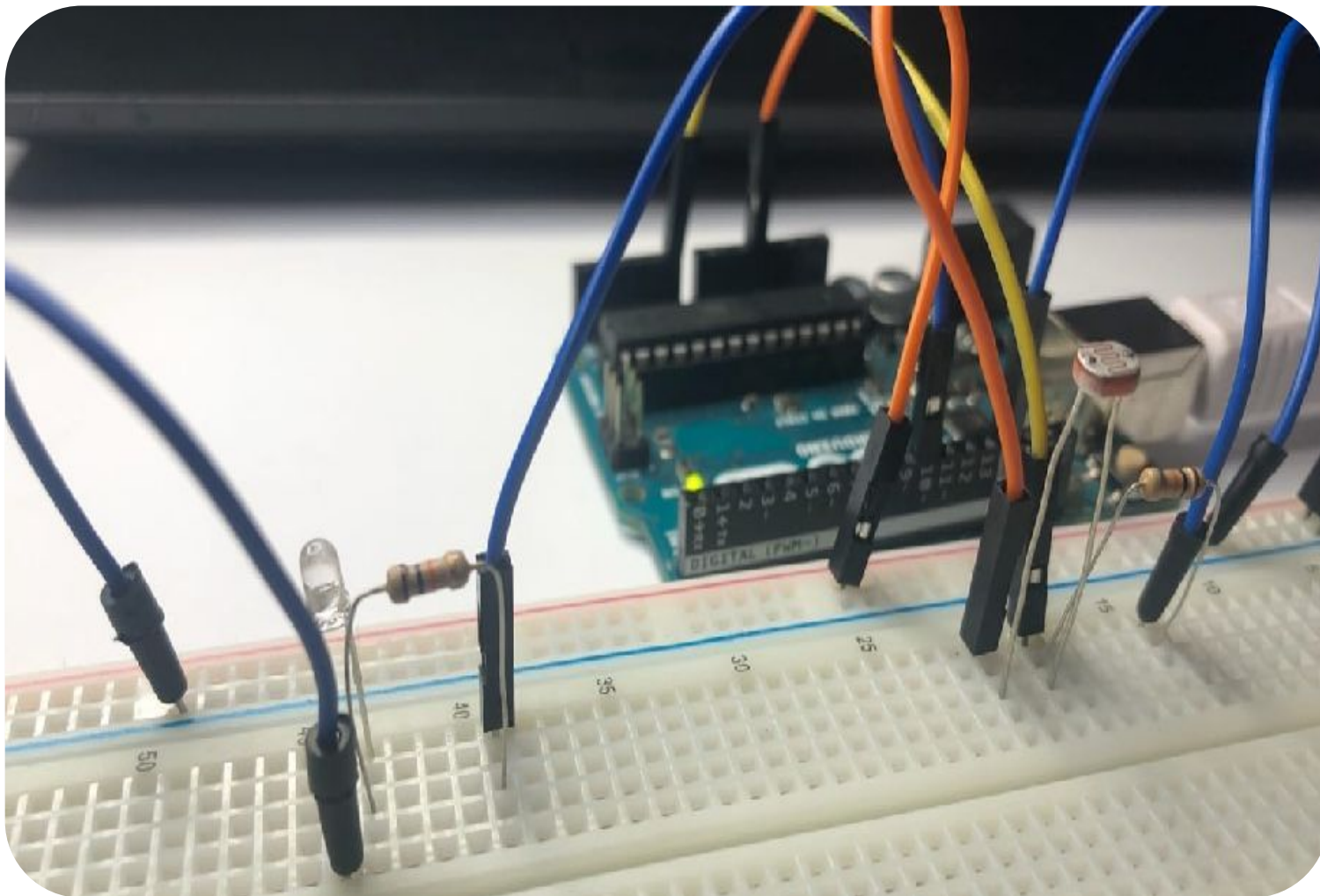
В этой схеме мы устанавливаем фоторезистор между питанием и аналоговым входом. Это нам нужно для того, чтобы при уменьшении освещенности мы получали меньшее напряжение на аналоговом входе.



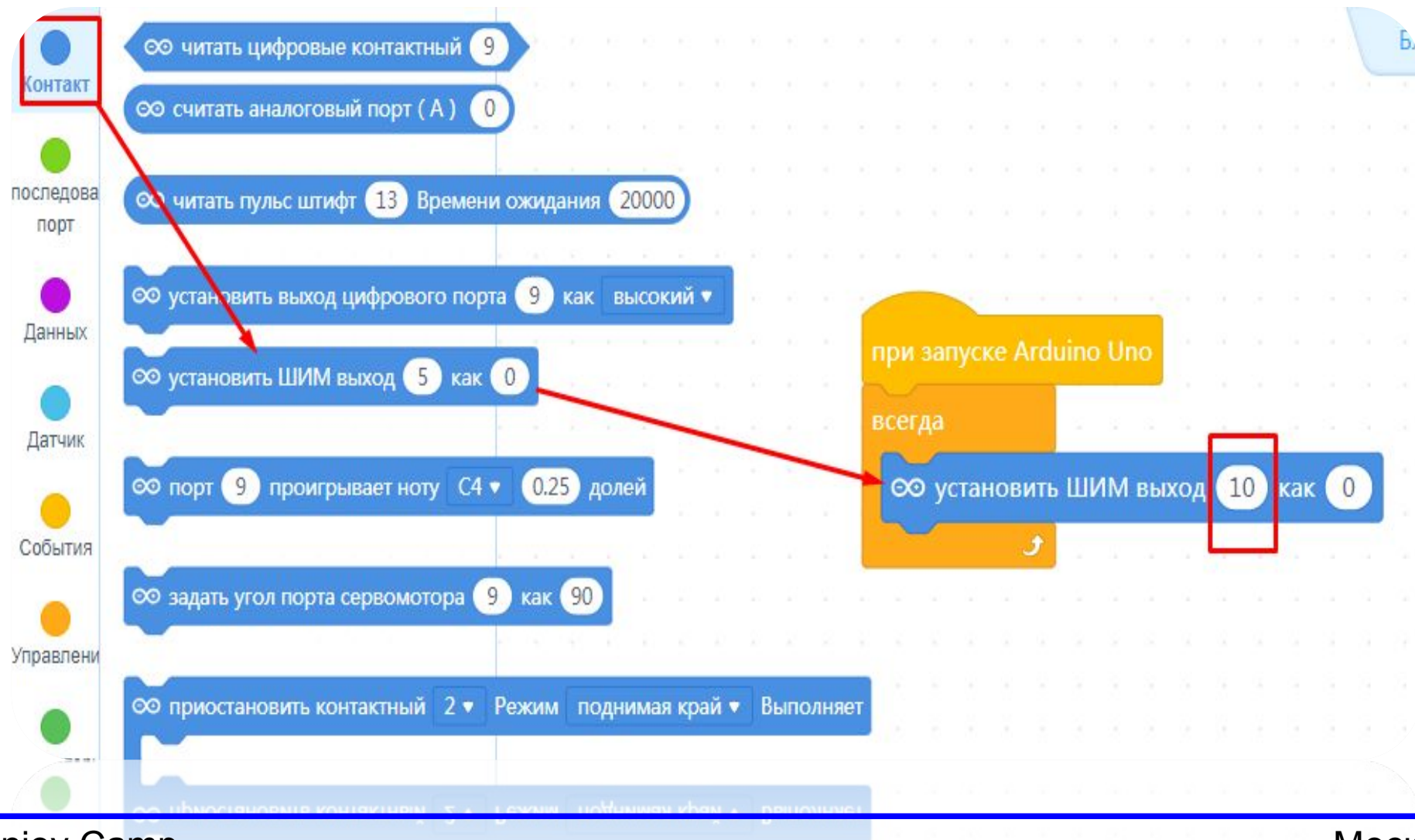
Вот что у вас
должно
получится, для
цепи с
фоторезистором



Теперь, по известной нам с Вами схеме, добавим светодиод, и подключим его к цифровому порту pin10



После полной сборки схемы, создадим программу, которая будет менять яркость светодиода в зависимости от того, как много или мало света получает фоторезистор





Контракт

последова
порт

Данных

Датчик

Карта 50 от 1 , 100 для 1 , 1000)

Зависимости 50 между 1 , 100)

123 преобразуется в целое число ▾

97 Преобразованный символ ASCII

а преобразованный номер ASCII

при запуске Arduino Uno

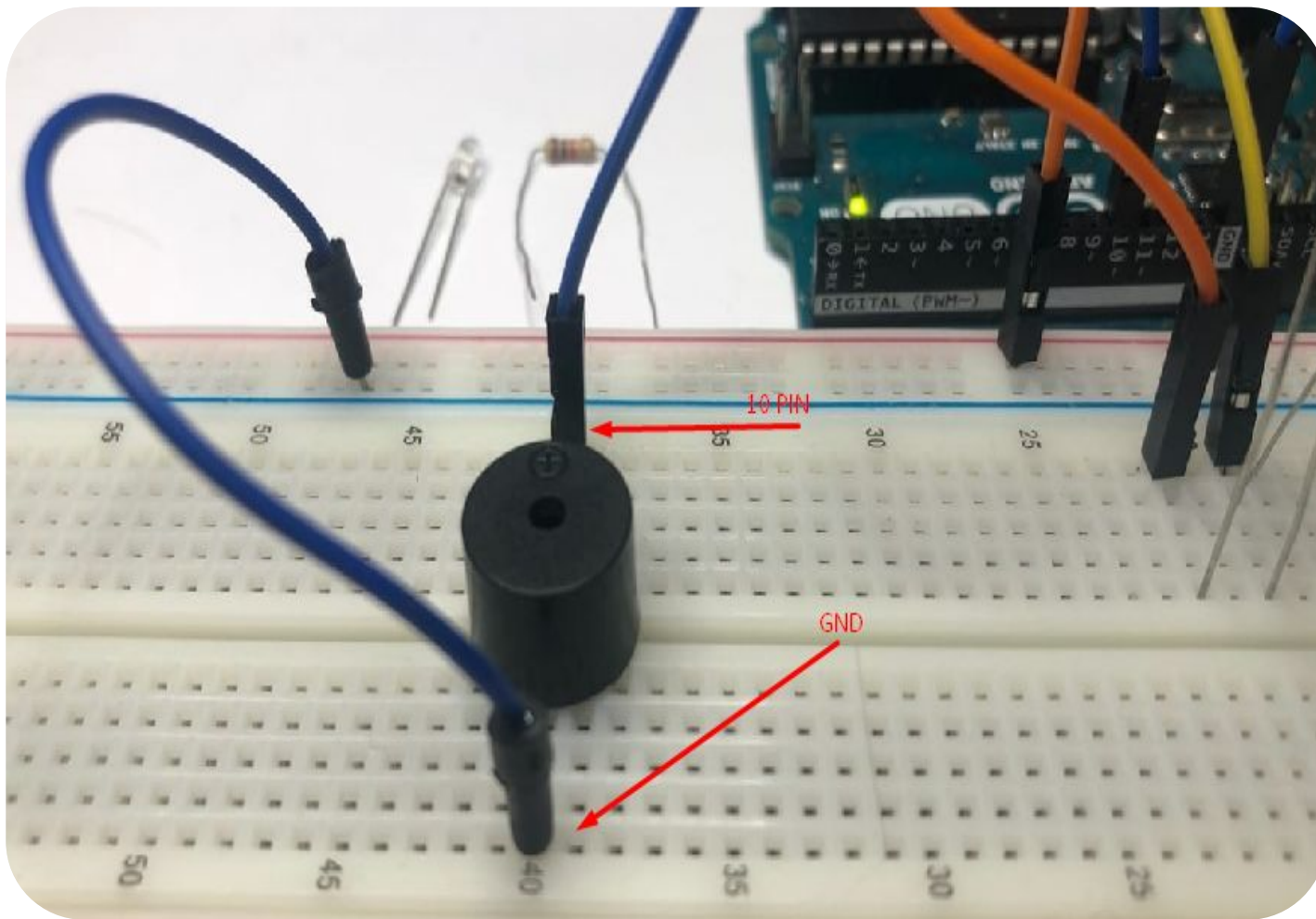
всегда

установить ШИМ выход 10 как Карта 50 от 0 , 1024 для 255 , 0)

Блок «Карта» (map) будет отбрасывать значения за пределами указанных диапазонов, а также масштабирует их по заданному правилу – переведет сигнал с фоторезистора в яркость светодиода



Теперь закройте фоторезистор тенью от руки или посветите фонариком на телефоне. Конечно, светодиод будет менять яркость



Пора узнать, почему
пищалку ранее мы
назвали «Мерзкой»

Для этого уберем
резистор и светодиод,
на место которых
поставим пищалку!
Терменвокс почти готов



при запуске Arduino Uno

всегда

установить ШИМ выход 10 как Карта считать аналоговый порт (A) 0 от 0 , 1024 для 2000 , 300)

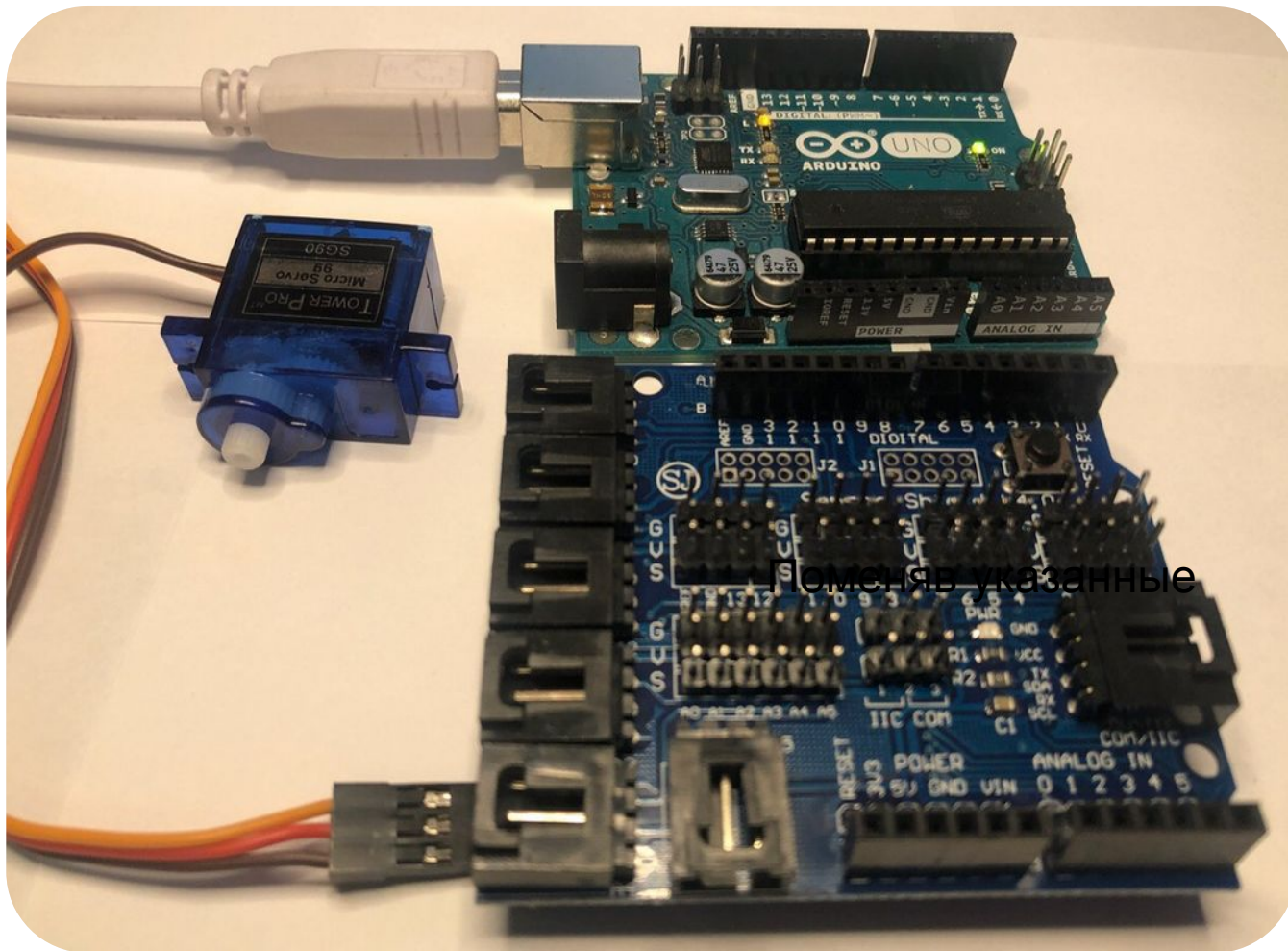
Поменяв указанные значения блока «Карта» можно изменять частоту звучания пищалки

Самые «мерзкие» ноты находятся в диапазоне от 50Гц до 20000Гц.

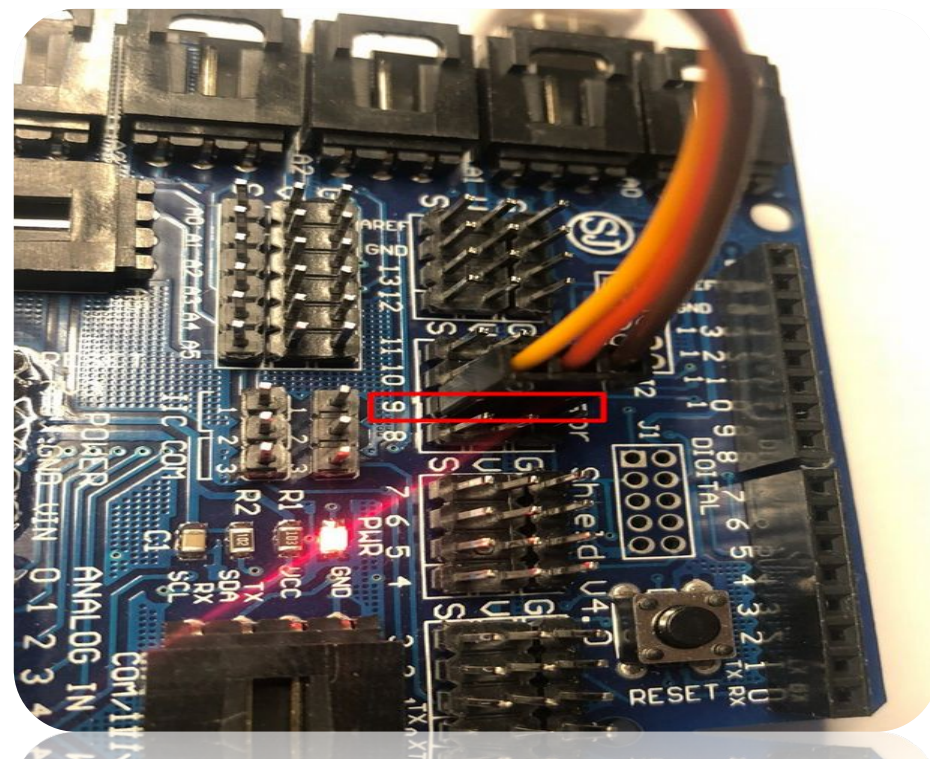
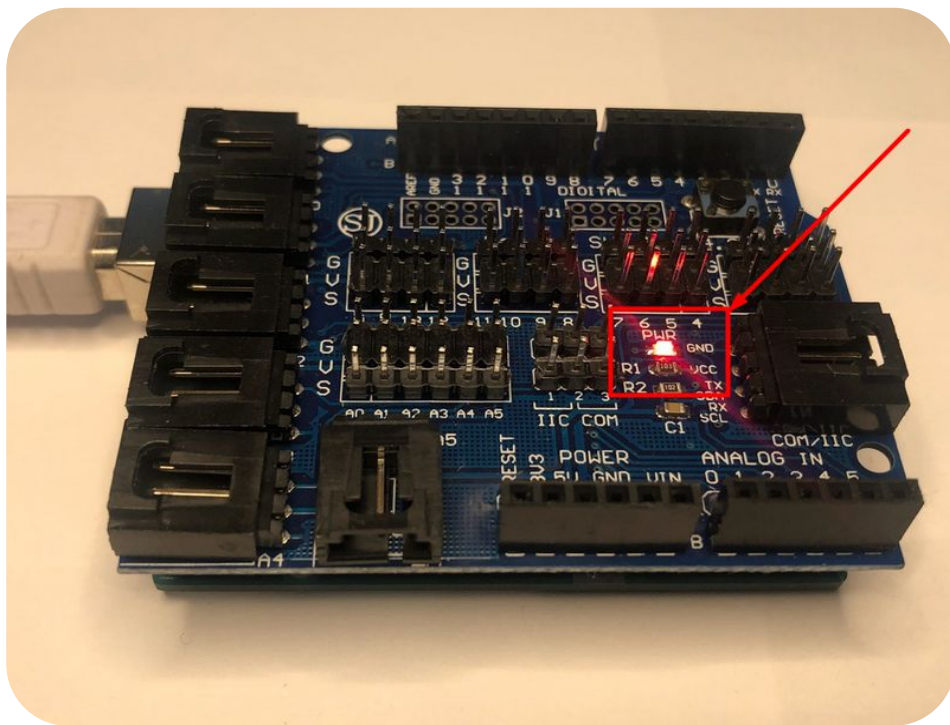
Преступник услышав такие волшебные звуки, никогда не решится залезть в наш будущий умный дом!!!



Sensor Shield и
сервопривод SG-90
—
Механика умного дома



Для простого подключения и управления сервоприводом мы с Вами будем использовать Motor Shield — так называемую плату расширения. Она позволяет подключать огромное количество периферии без использования лишних проводов и соединений



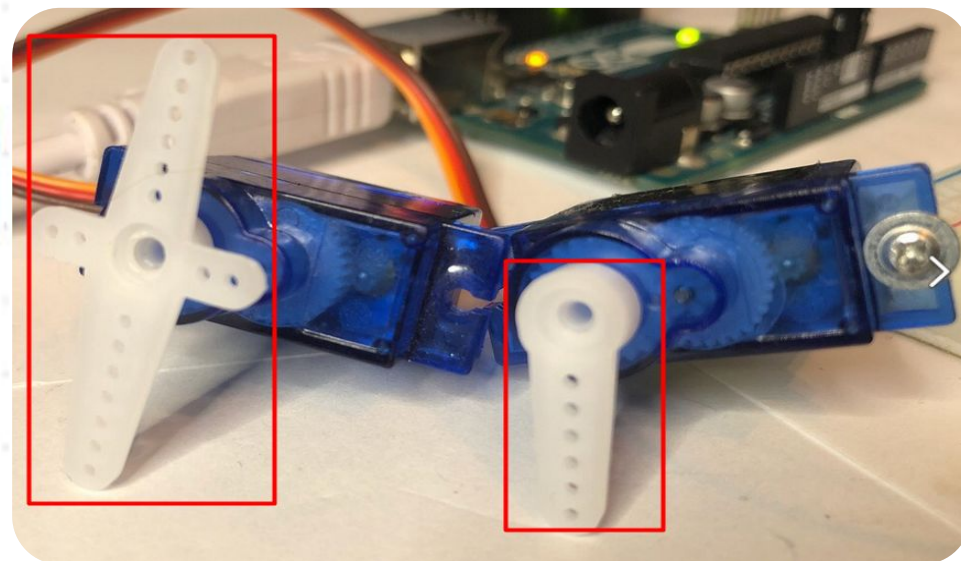
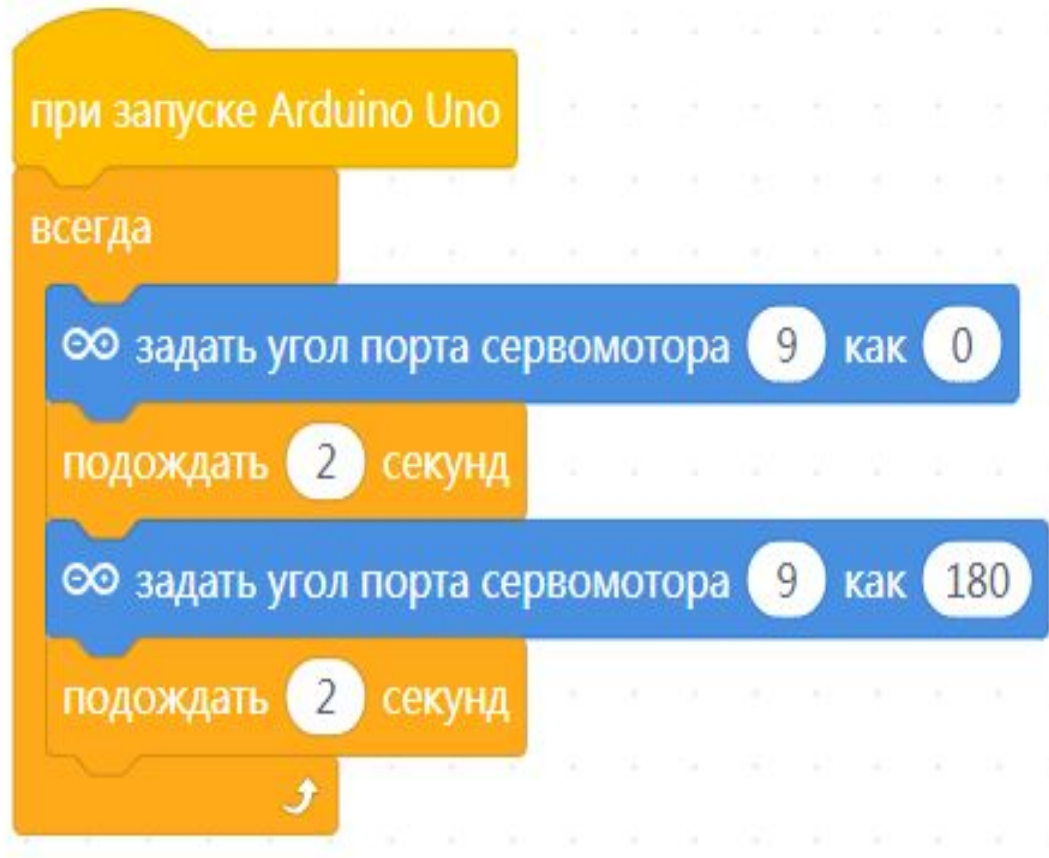
При правильном подключении платы красным светодиод на шилде должен загореться. Важно подключить шлейф сервопривода так, что бы желтый провод (сигнальный) был в ряду сигнальных пинов S



Перетянем блоки из модуля «Контакт» , предварительно установив управляющие блоки. Пин 9 будет опрвлять сервоприводом. Конечно, для большей наглядности не забудьте одеть валы-редукторы на сервопривод



Дополним нашу программу уже известными Вам блоками, и сервопривод придет в непрерывное движение

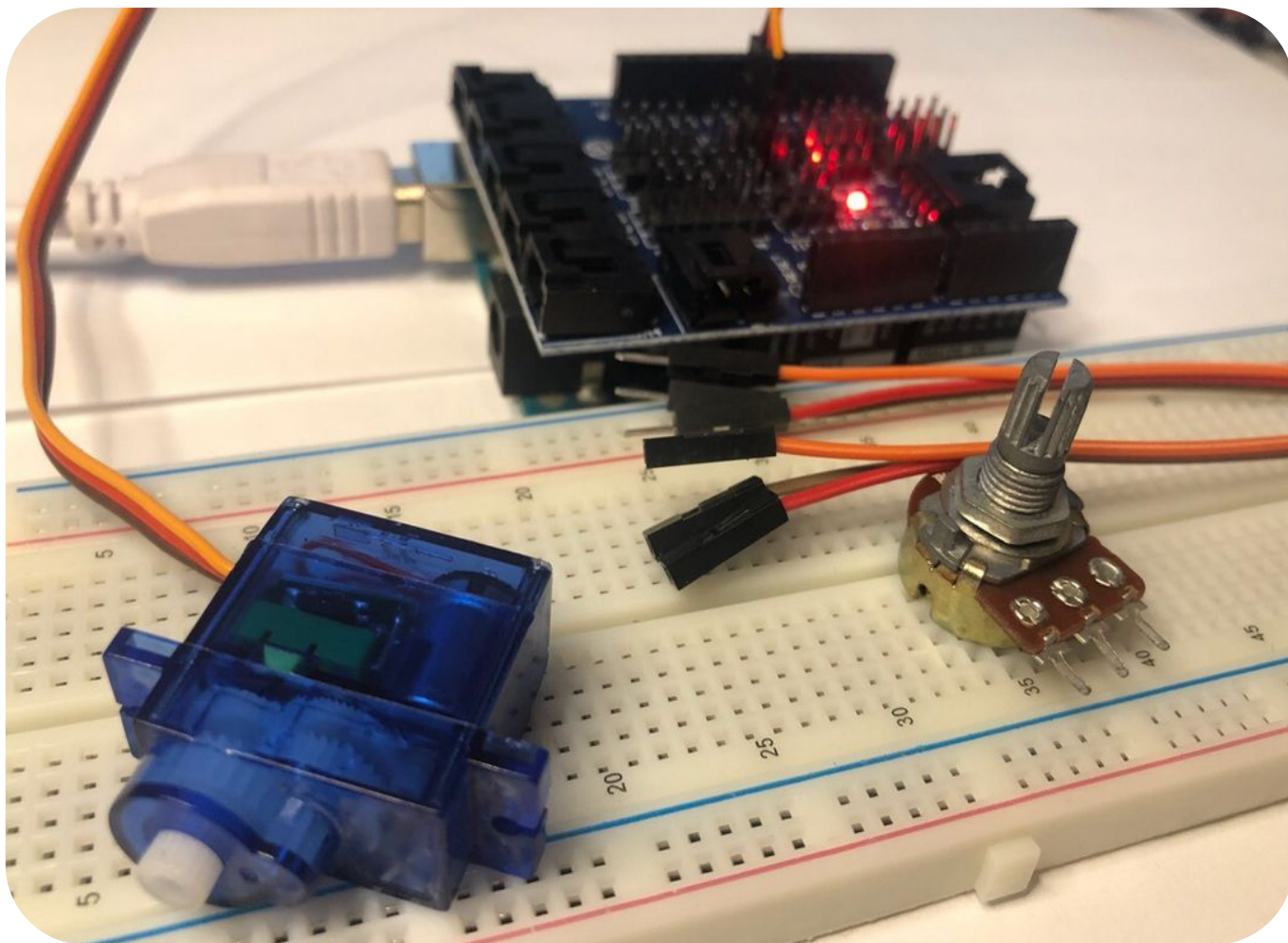




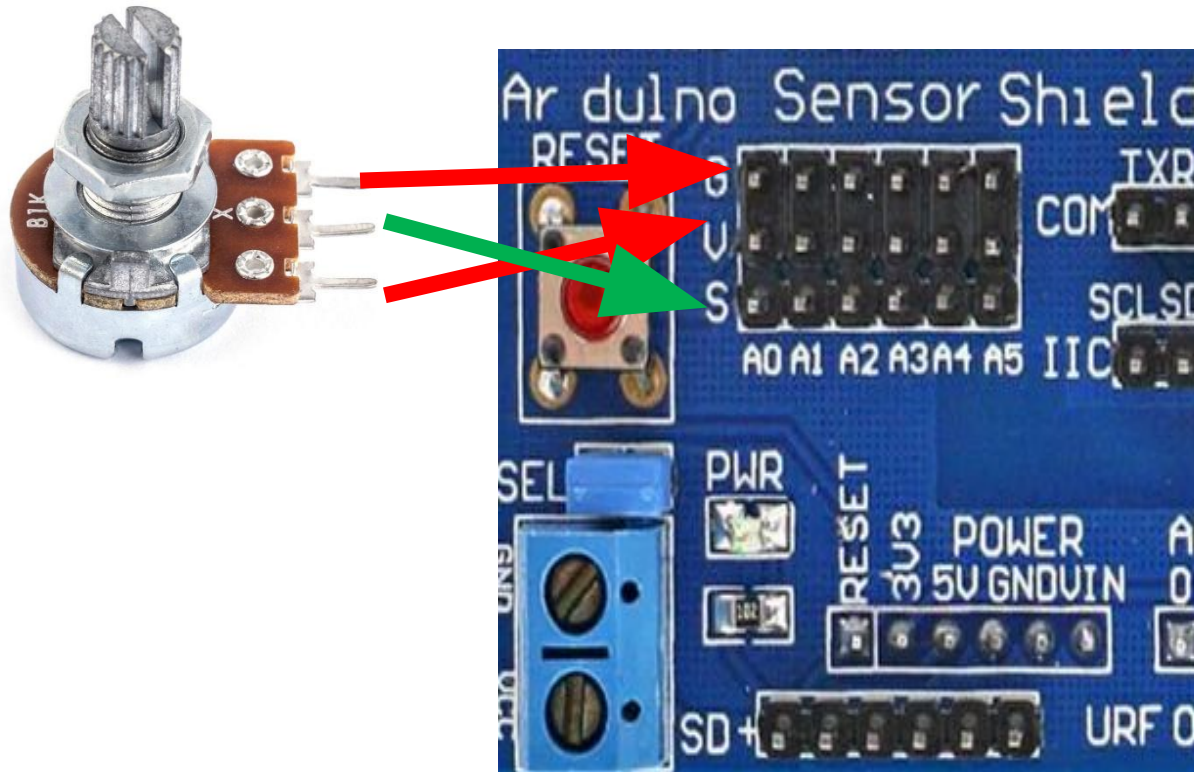
Sensor Shield и
сервопривод SG-90
и потенциометр

—

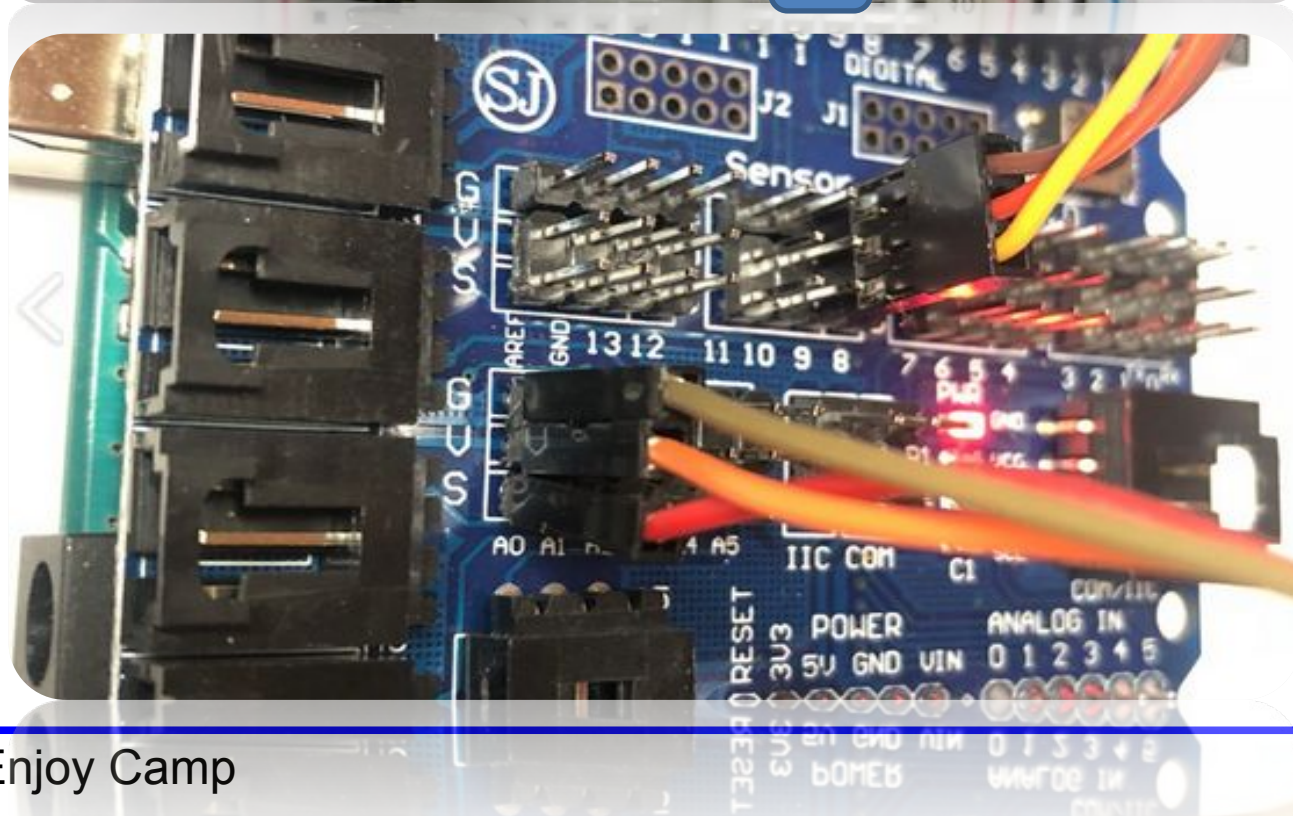
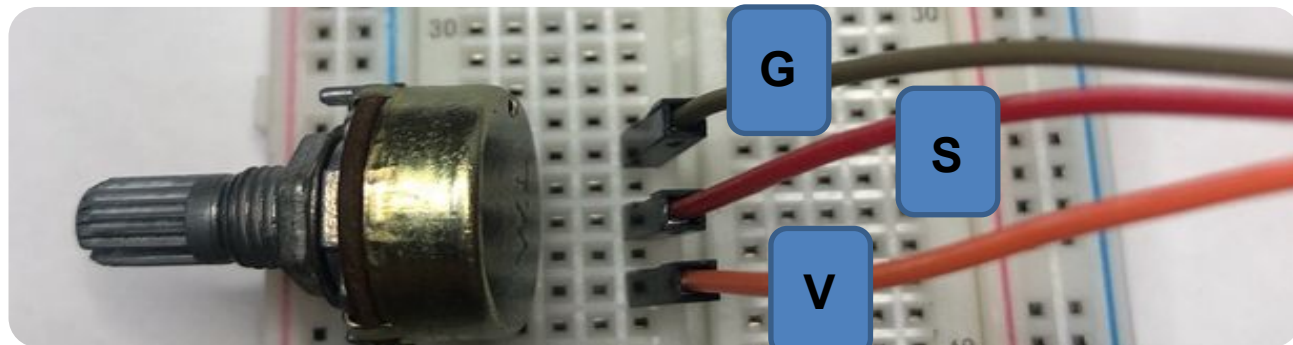
Управление механикой
умного дома



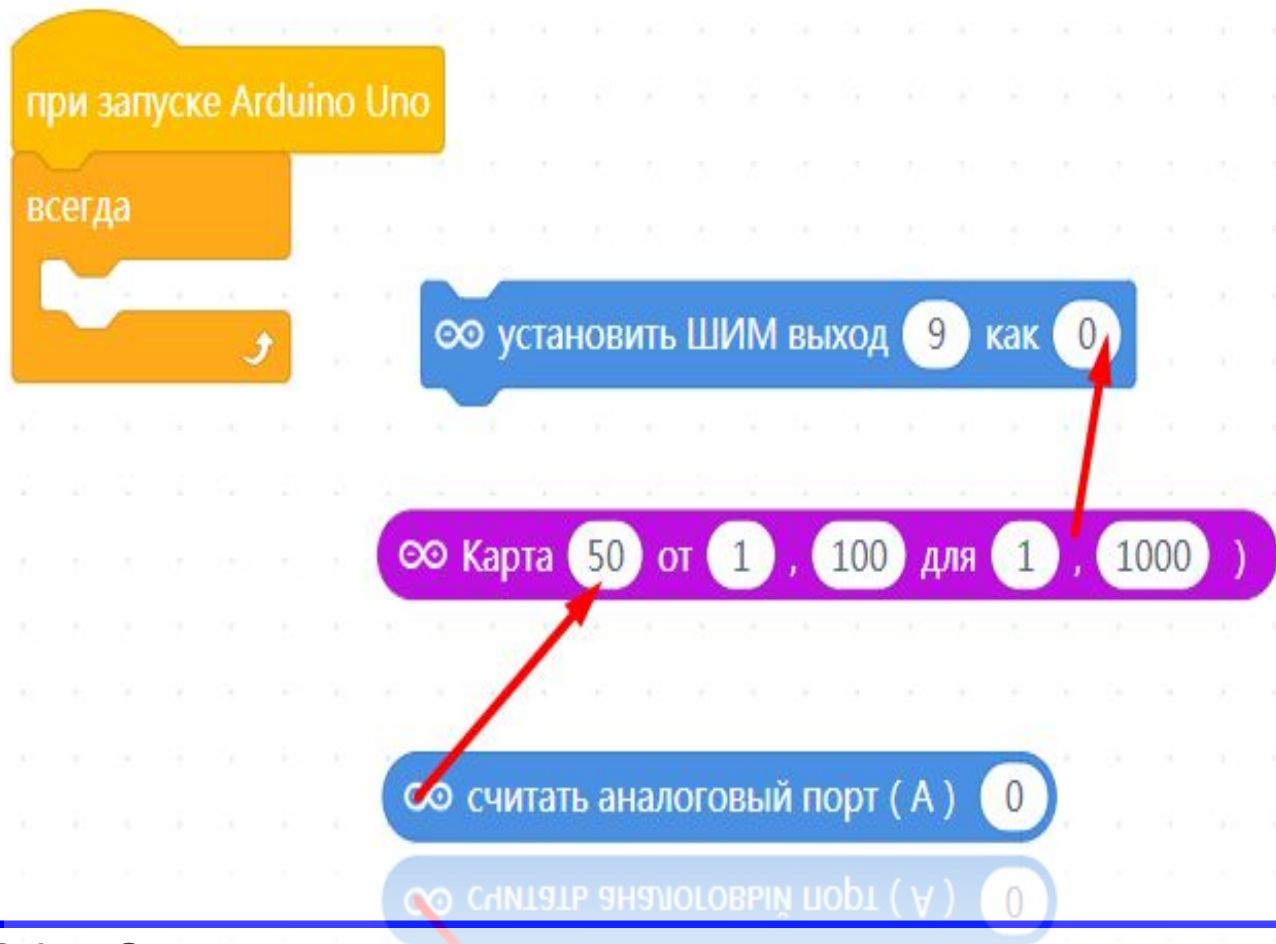
Потенциометр
позволит нам с Вами
управлять
сервоприводом
плавно, крутя ручку
потенциометра.
Потенциометр — тот
же самый резистор,
только мы можем
физически менять
его сопротивление



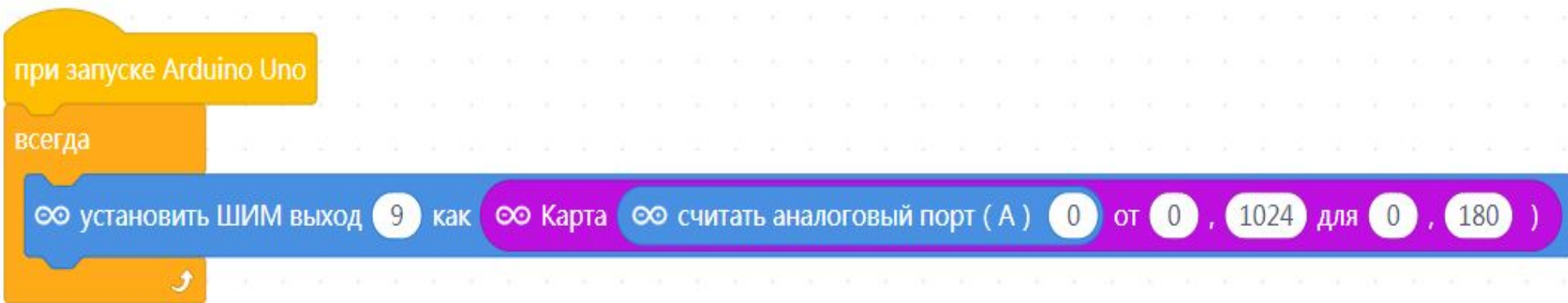
Центральный контакт (нога) отвечает за сигнал, который мы снимаем с потенциометра, поэтому очень важно подключить его к аналоговому пину A0, в ряд с сигнальными пинами S. Остальные две ноги подключаем к земле и питанию



Потенциометр
такого плана можно
воткнуть в макетку,
но также можно
подключить его
просто проводами к
Сенсор шилду и к
ногам
потенциометра



Воспользуемся блоком «Карта» который позволит переводить сигнал получаемый с потенциометра в сигнал, понятный для сервопривода — т. е. в диапазон градусов от 1 до 180



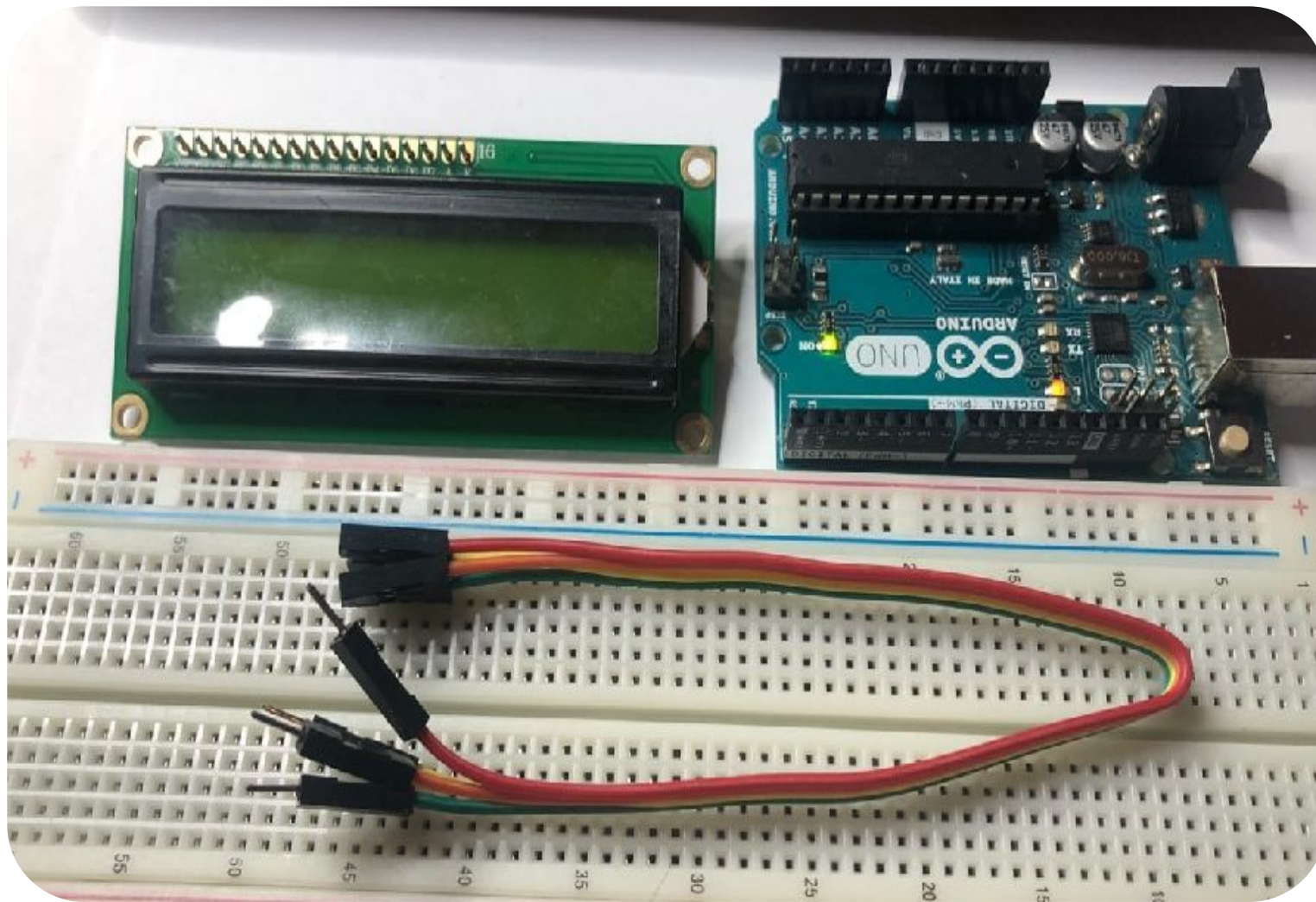
Теперь, поворачивая ручку потенциометра можно плавно регулировать поворот сервопривода. Проверьте правильность указанных значений в блоке «Карта»

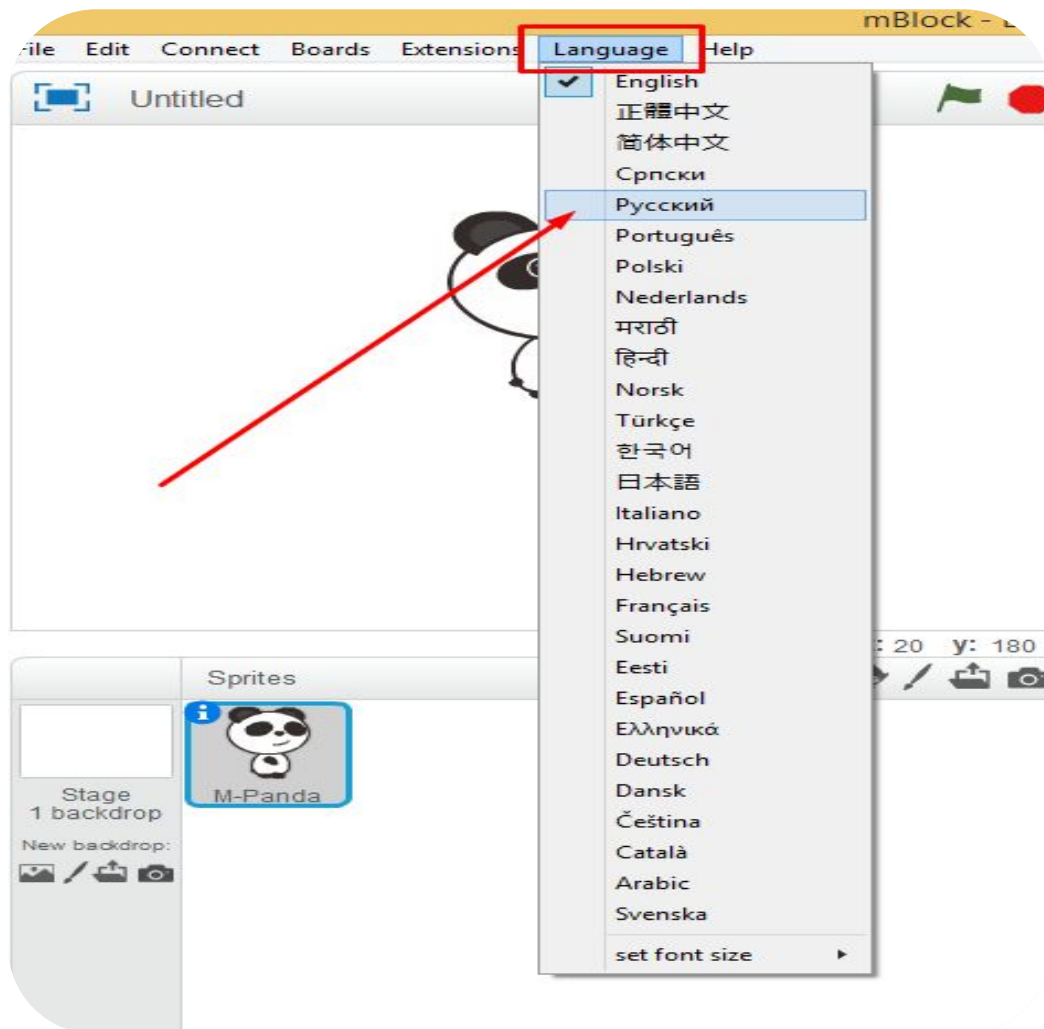


LSD дисплей

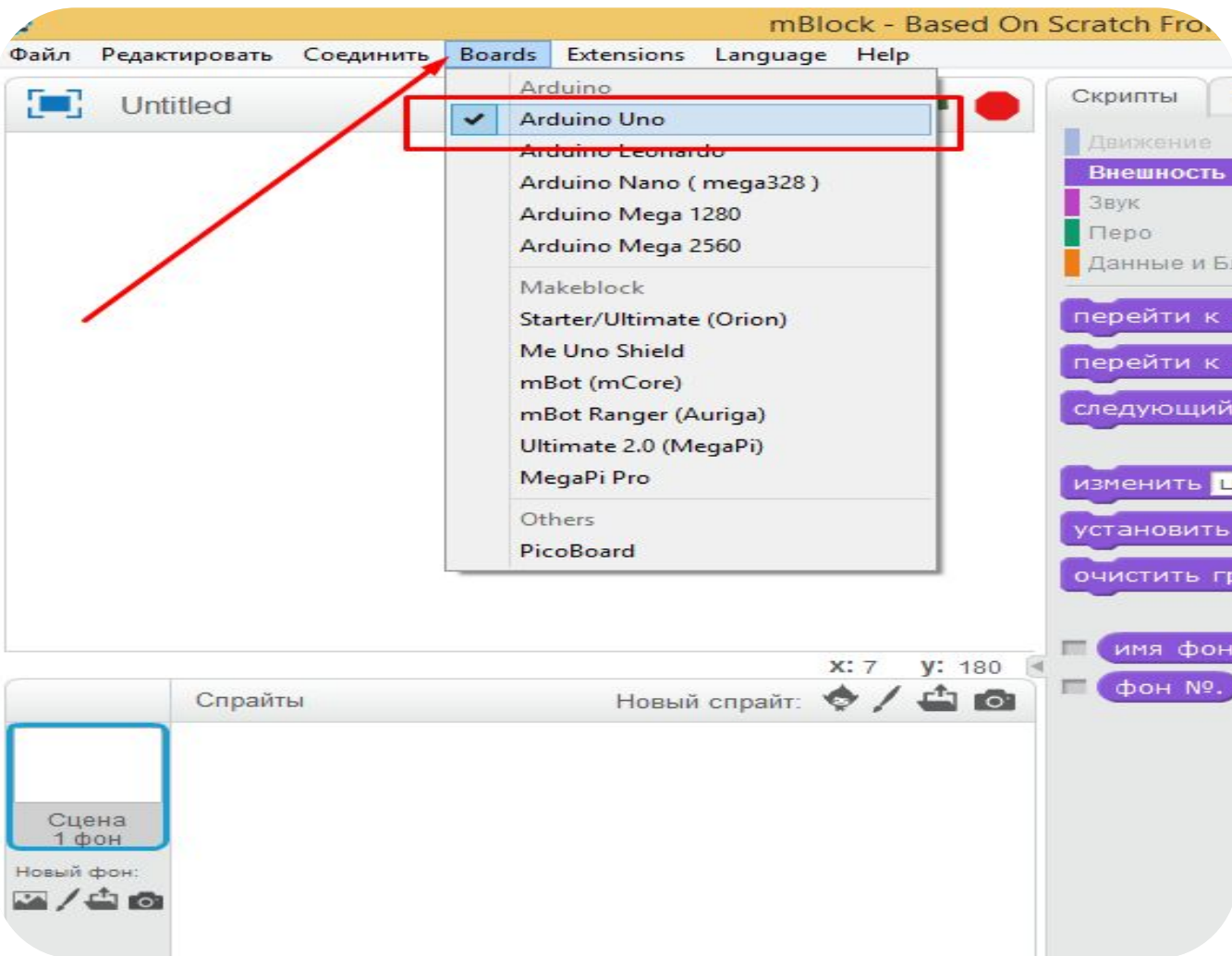


Пришло
время
подключить
один из самых
информативн
ых модулей —
LCD дисплей,
на который
мы с Вами
будем
выводить всю
информацию
с периферии
умного дома



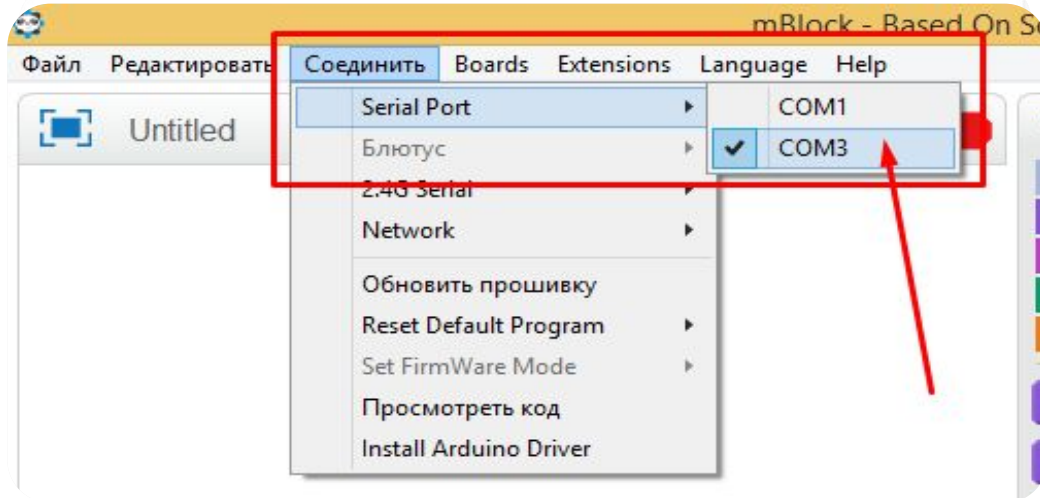
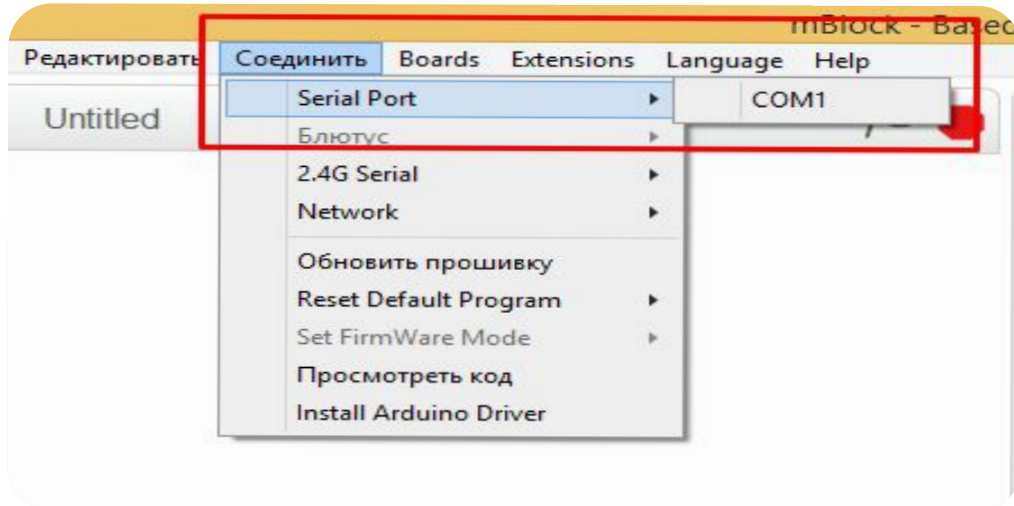


Версия mBlock 3.4
позволяет напрямую
работать с
библиотеками. Для
начала выберем
язык программы —
русский. Конечно,
программа не
полностью
адаптирована под
русскоязычных
пользователей, но
мы то с Вами знаем
Английский язык!

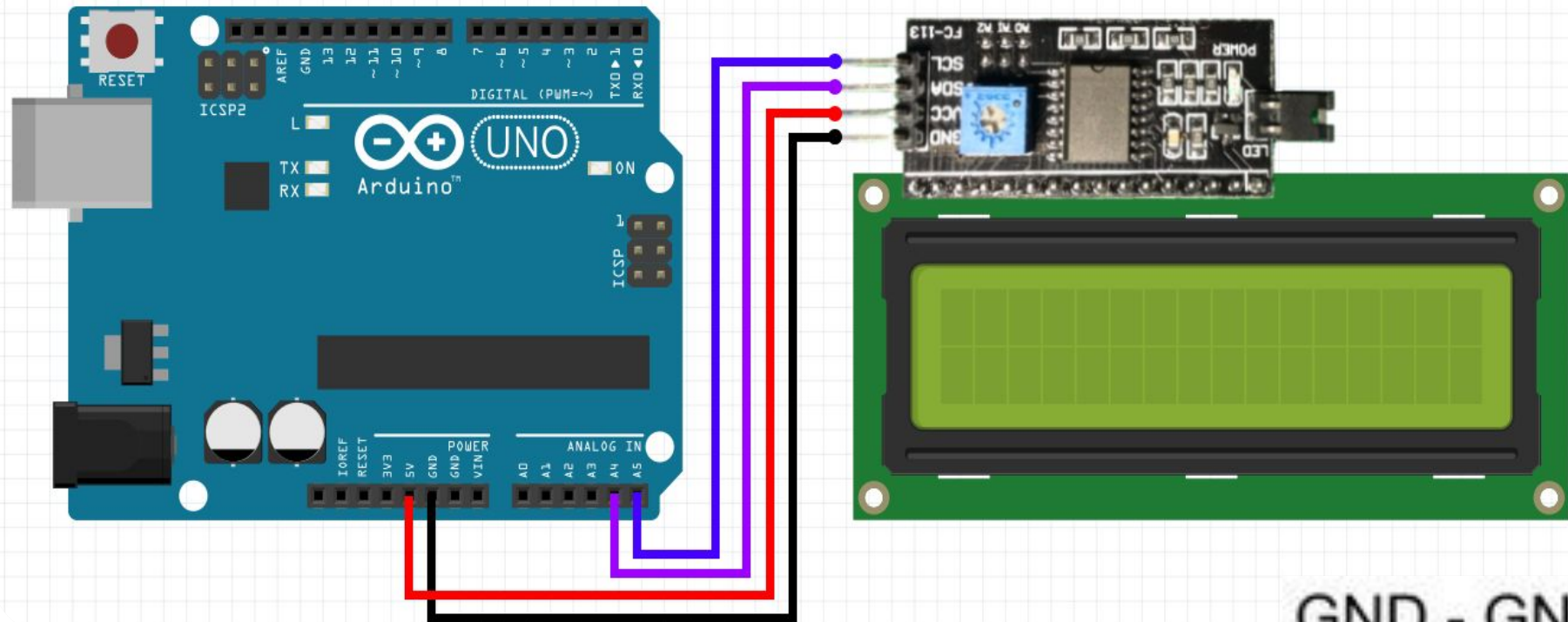


После смены языка
подключим во
вкладке «Платы»
Используемую нами
Arduino Uno.

Замечательно, что
изучаемая нами
версия mBlock
позволяет работать
практически со всеми
известными Arduino
платами

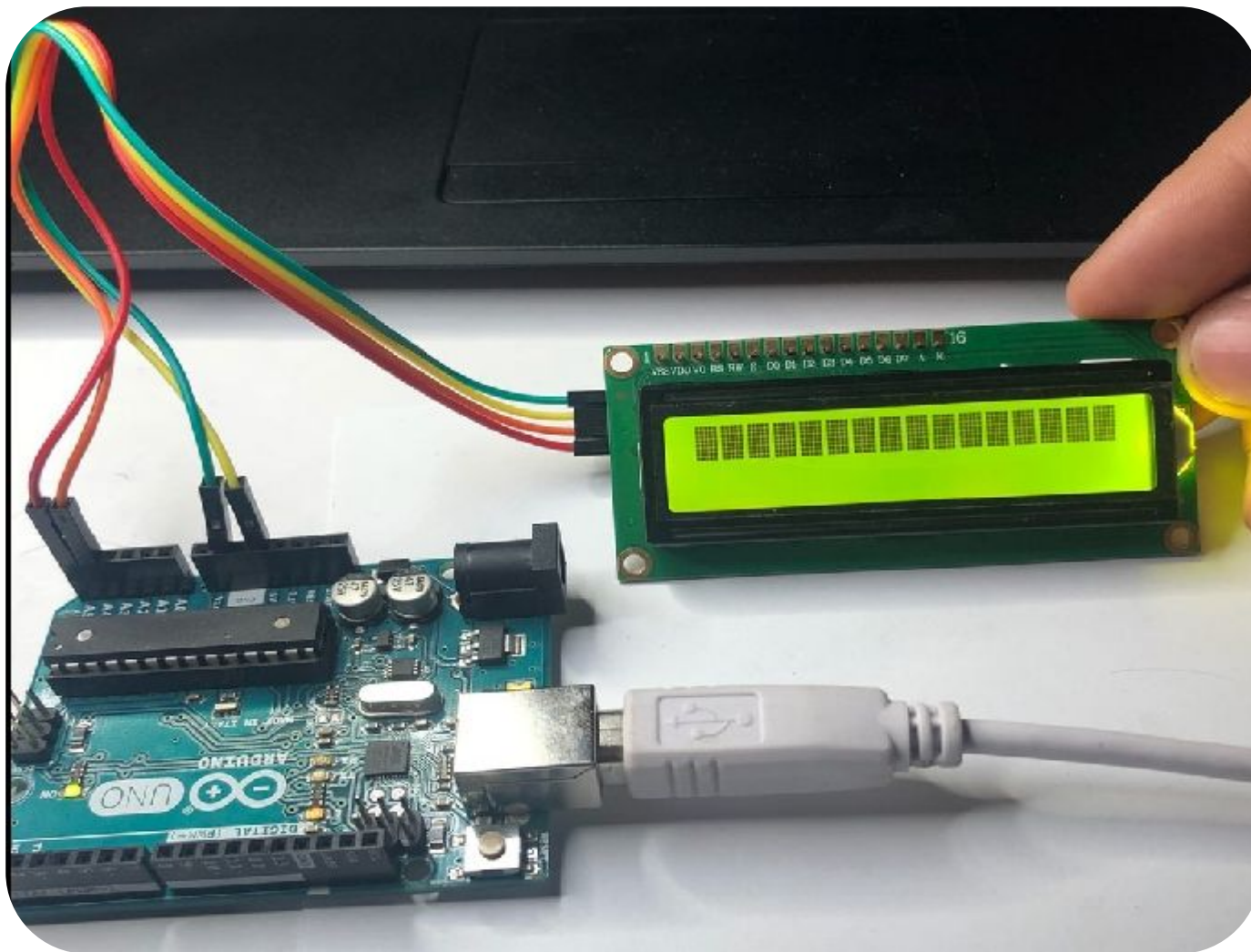


Сериал Порт позволяет компьютеру установить двух стороннюю адресную связь с компьютером. Выберем вкладку «Соединить» и выберем порт Arduino. После подключения кабелем нашего микроконтроллера сразу появится новый порт — он то как раз нас с Вами и интересует. Перед загрузкой программы всегда важно проверять подключена ли плата по ком порту!

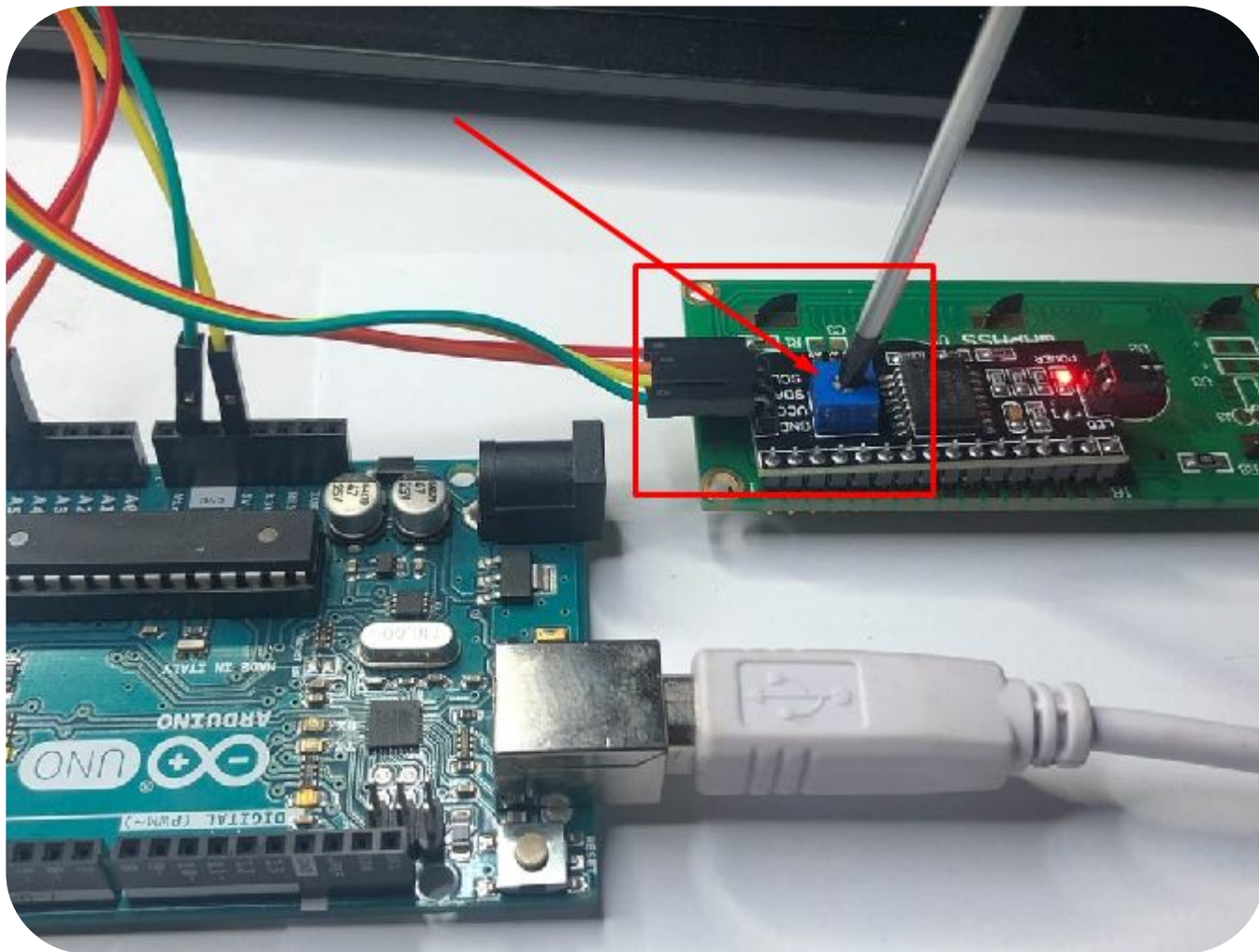


На этот раз, выполним подключение без макетной платы. Соблюдайте правильность подключения. В таблице можно видеть название портов — что к чему подключать.

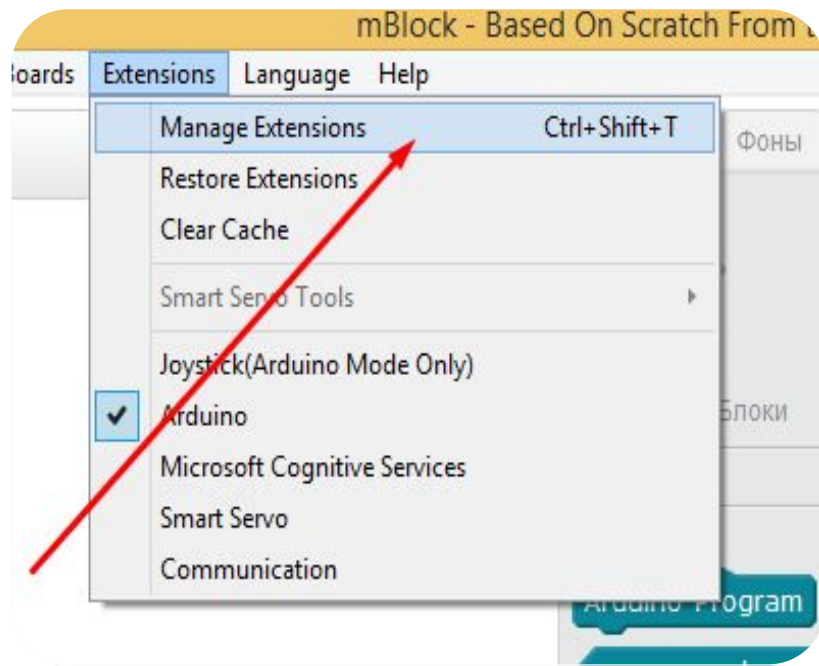
GND - GND
VCC - 5V
SDA - A4
SCL - A5



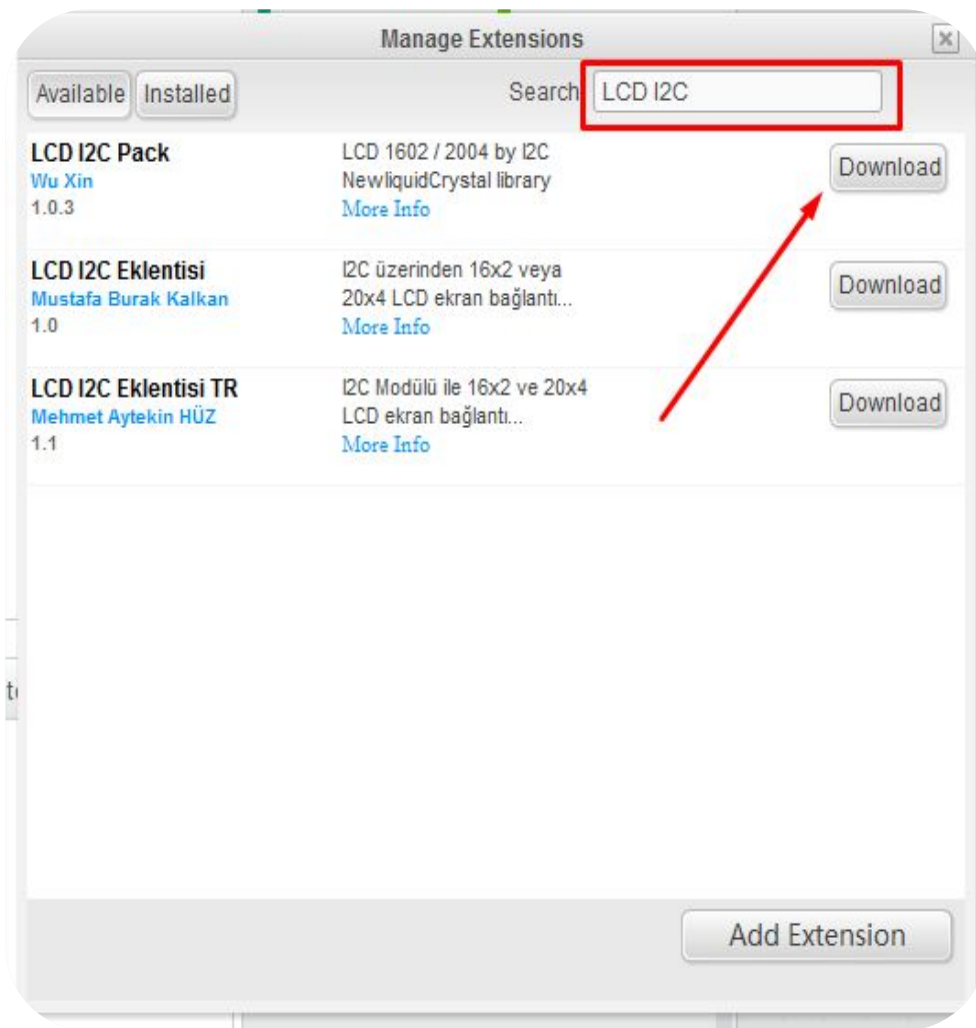
Как вы уже заметили, при правильном подключении, LCD дисплей сразу засветиться. На задней стороне дисплея нужно подкрутить потенциометр, пока на дисплее не появятся пиксельные «квадратики»



Следующим шагом
будет подключение
библиотеки для
работы с дисплеем
— это очень
облегчит его
программирование.
Рабочий код по
выводу слов сразу
сократится с 100500
строчек до 4.



Во вкладке расширение
выбираем — менеджер
библиотек и в поиске вводим
LCD I2C (I — большая буква i)
и нажимаем загрузить





Движение

Внешность

Звук

Перо

Данные и Блоки

События

Контроль

Сенсоры

Операторы

Робот

Arduino Program

читать цифровой pin 9

читать аналоговый pin (A) 0

read pulse pin 13 timeout 20000

установить цифровой pin 9 udgang как HIGH

установить PWM/ШИМ pin 5 udgang как 0

воспроизвести звук pin 9 на ноте C4 такт 1/2

установить серво pin 9 как 90 grader

serial write text hello

serial available bytes

serial read byte

read ultrasonic sensor trig pin 13 echo pin 12

таймер

Пора приступить к написанию серьезной, но очень простой модульной программы.

Перенесём блок Arduino Program — он отвечает за начало работы нашего кода в Arduino Uno



Данные и Блоки Робот

read pulse pin 13 timeout 20000

установить цифровой pin 9 udgang как HIGH

установить PWM/ШИМ pin 5 udgang как 0

воспроизвести звук pin 9 на ноте C4 такт 1/2

установить серво pin 9 как 90 град

serial write text hello

serial available bytes

serial read byte

read ultrasonic sensor trig pin 13 echo pin 12

таймер

перезапустить таймер

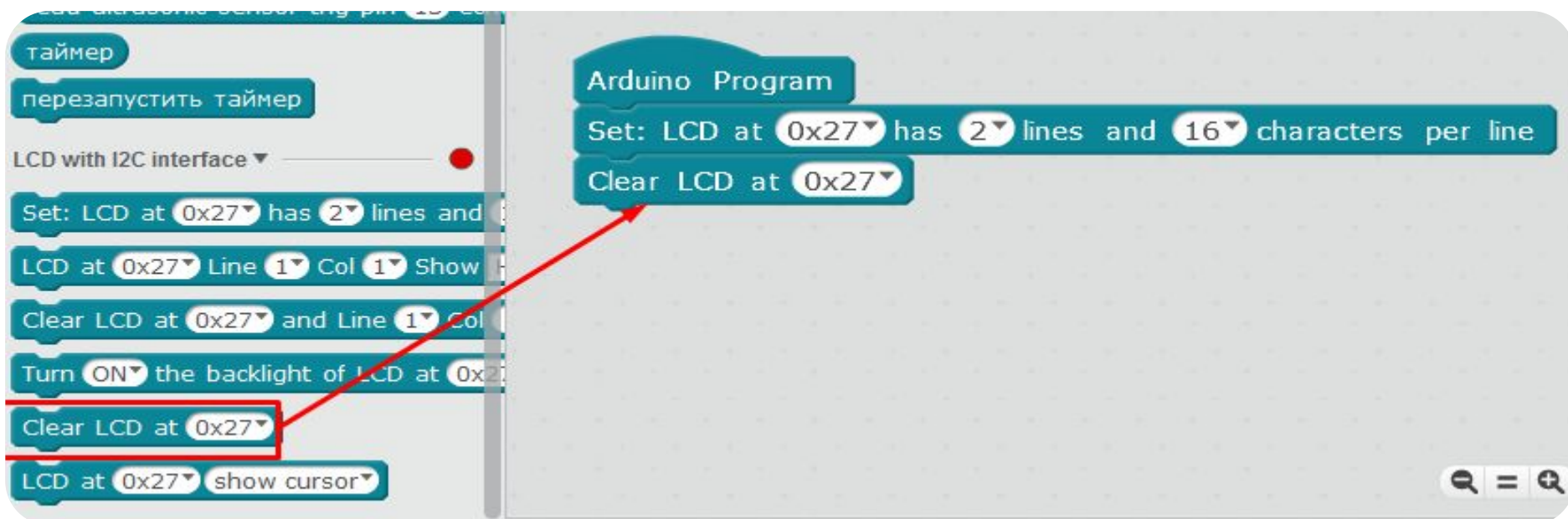
LCD with I2C interface

Set: LCD at 0x27 has 2 lines and 16 characters per line

Arduino Program

Set: LCD at 0x27 has 2 lines and 16 characters per line

Соединяем блок Arduino Program с блоком, из скачанной нами библиотеке. Этот блок отвечает за начало работы с LCD монитором — по стандарту там уже все настроено как нам с Вами надо (пока что, цифры менять не обязательно)



Конечно, что бы пропали «пиксельные квадратики» с нашего дисплейчика стоит его очистить. Для этого используем в нашем коде указанный блок. Вообще полезно им пользоваться после каждого вывода на LCD — это позволит избежать наслоений в тексте.



set pwm pin 5 output as 0

play tone pin 9 on note C4 beat Half

set servo pin 9 angle as 90

serial write text hello

serial available bytes

serial read byte

read ultrasonic sensor trig pin 13 echo pin 12

timer

reset timer

LCD I2C Pack

Set LCD at 0x27 has 2 lines and 16 characters per line

LCD at 0x27 Line 1 Col 1 Show Hello World!

Clear LCD at 0x27 and Line 1 Col 1 Show Hello World!

The backlight of LCD at 0x27 Turn ON

Clear LCD at 0x27

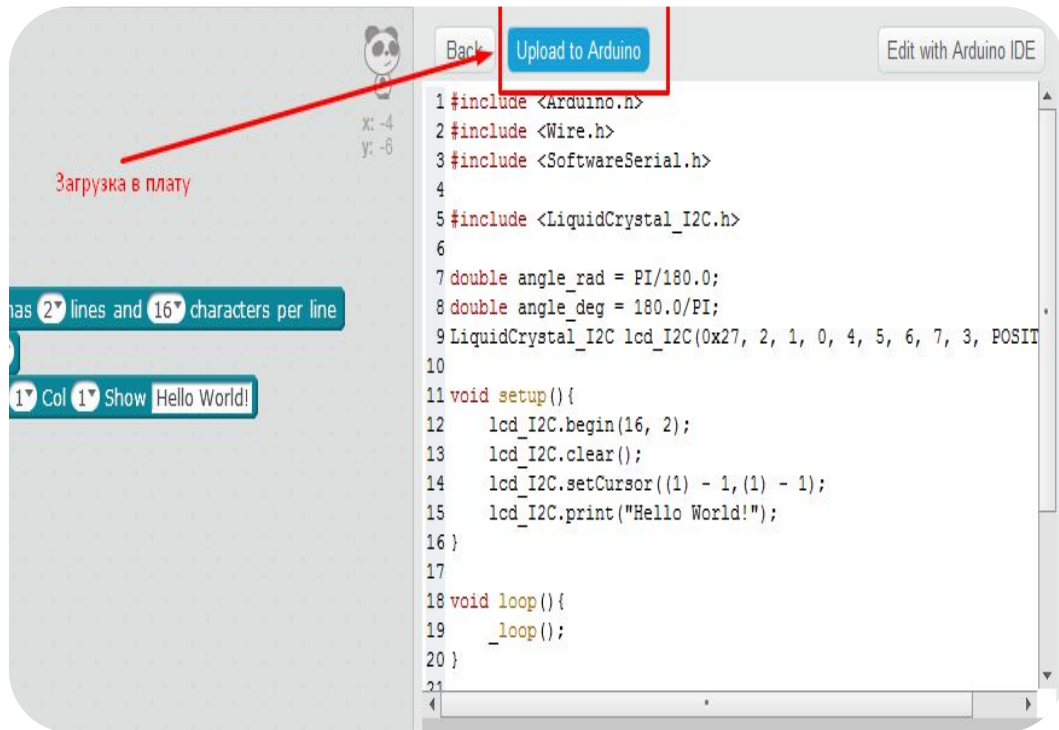
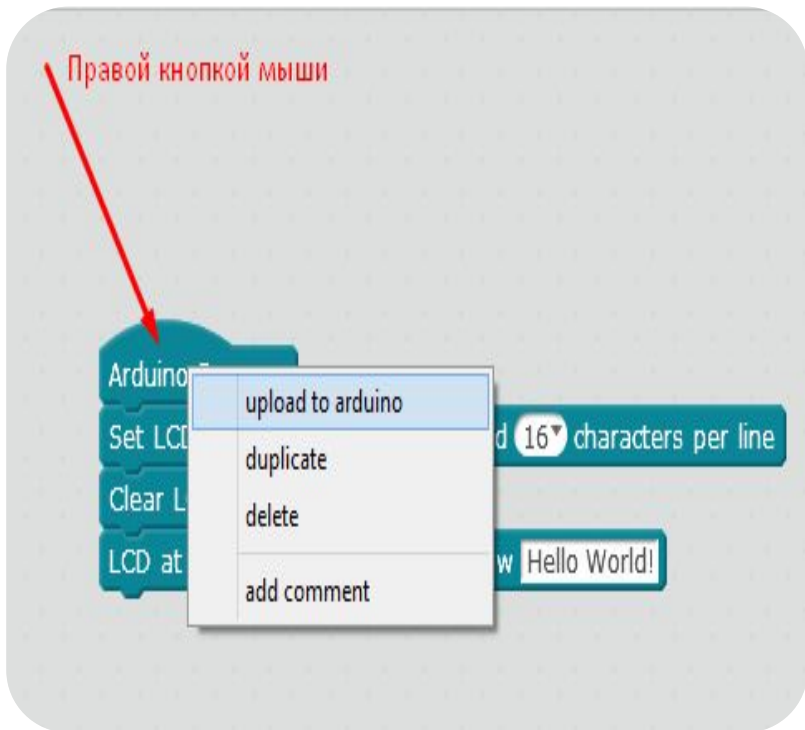
Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

Clear LCD at 0x27

LCD at 0x27 Line 1 Col 1 Show Hello World!

И самым
главным блоком
будет блок
отвечающий за
вывод текста
«Hello Word» в
первую строку,
начиная
выводить текст
с первого
символа нашего
дисплея



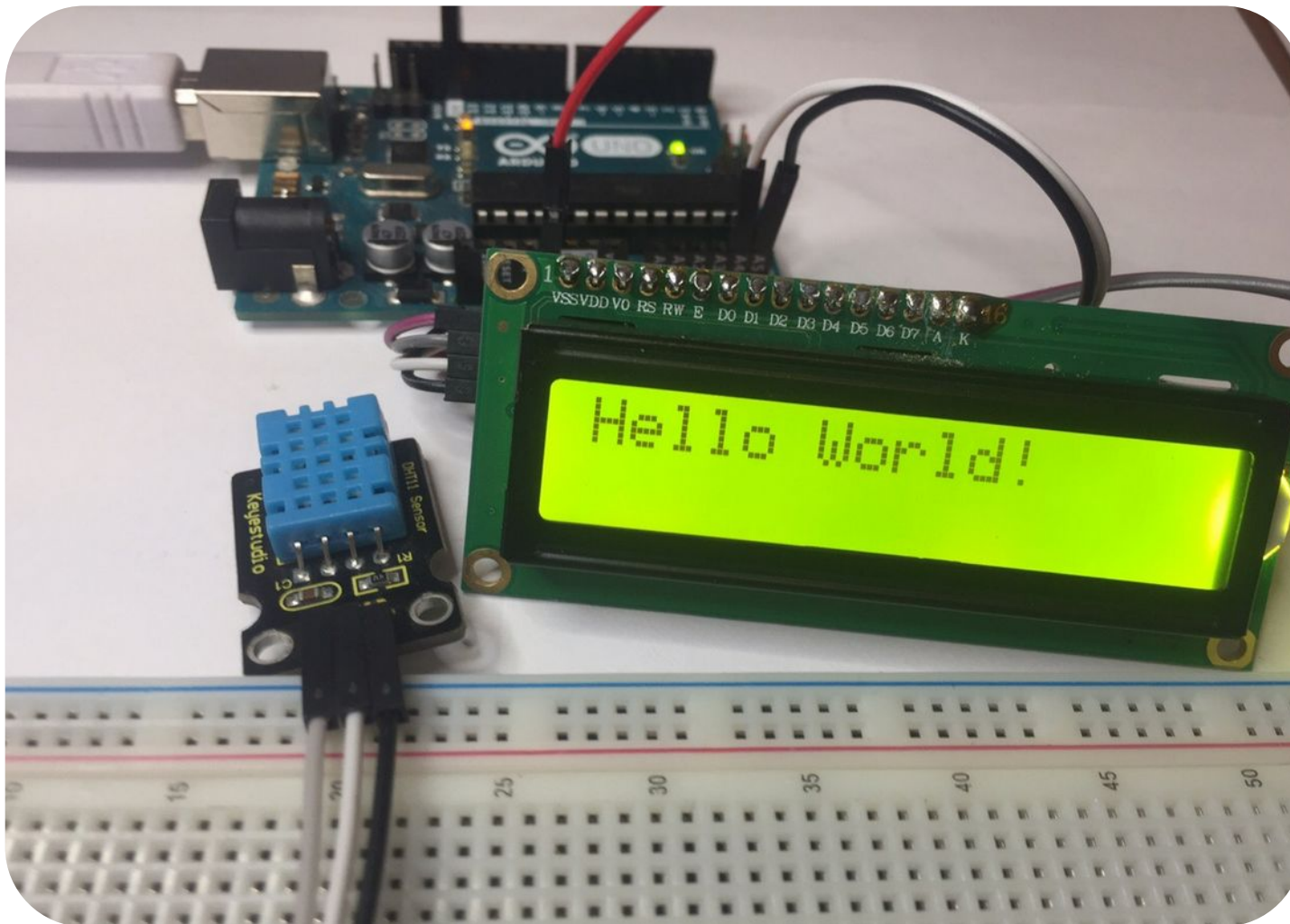
Для загрузки нажимаем ПКМ на блок Arduino Program. В открывшемся окне нажимаем «Upload to Arduino» тем самым, загружая созданную нами программу в Arduino



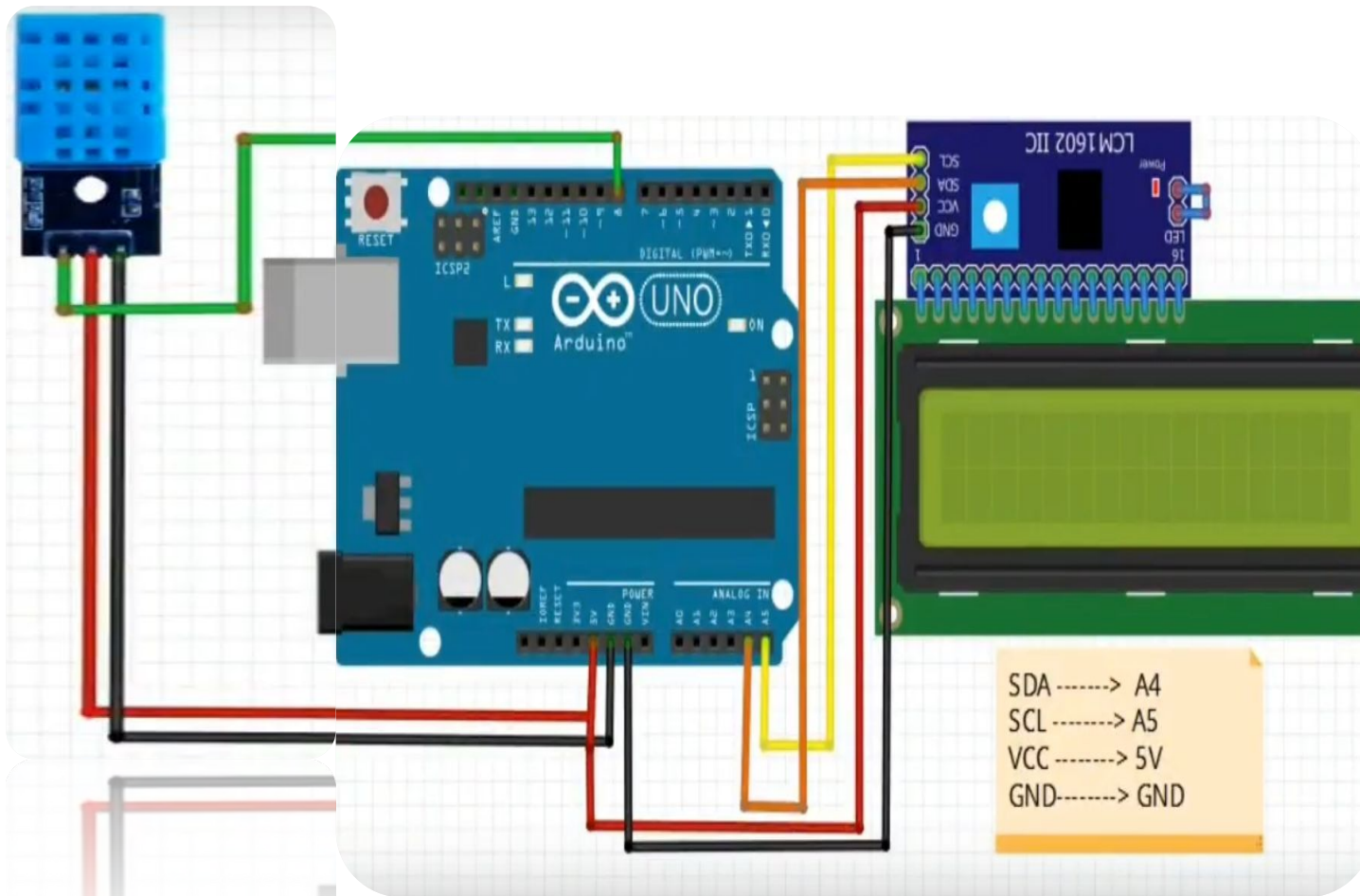
Если надпись видно плохо или тускло — отвёрткой подкрутим потенциометр до четкого появления символов. Попробуйте самостоятельно вывести слова на Ваше усмотрение (смайлики, символы) в первый и второй ряд.
Напоминаем — в ряду может поместится 16 символов



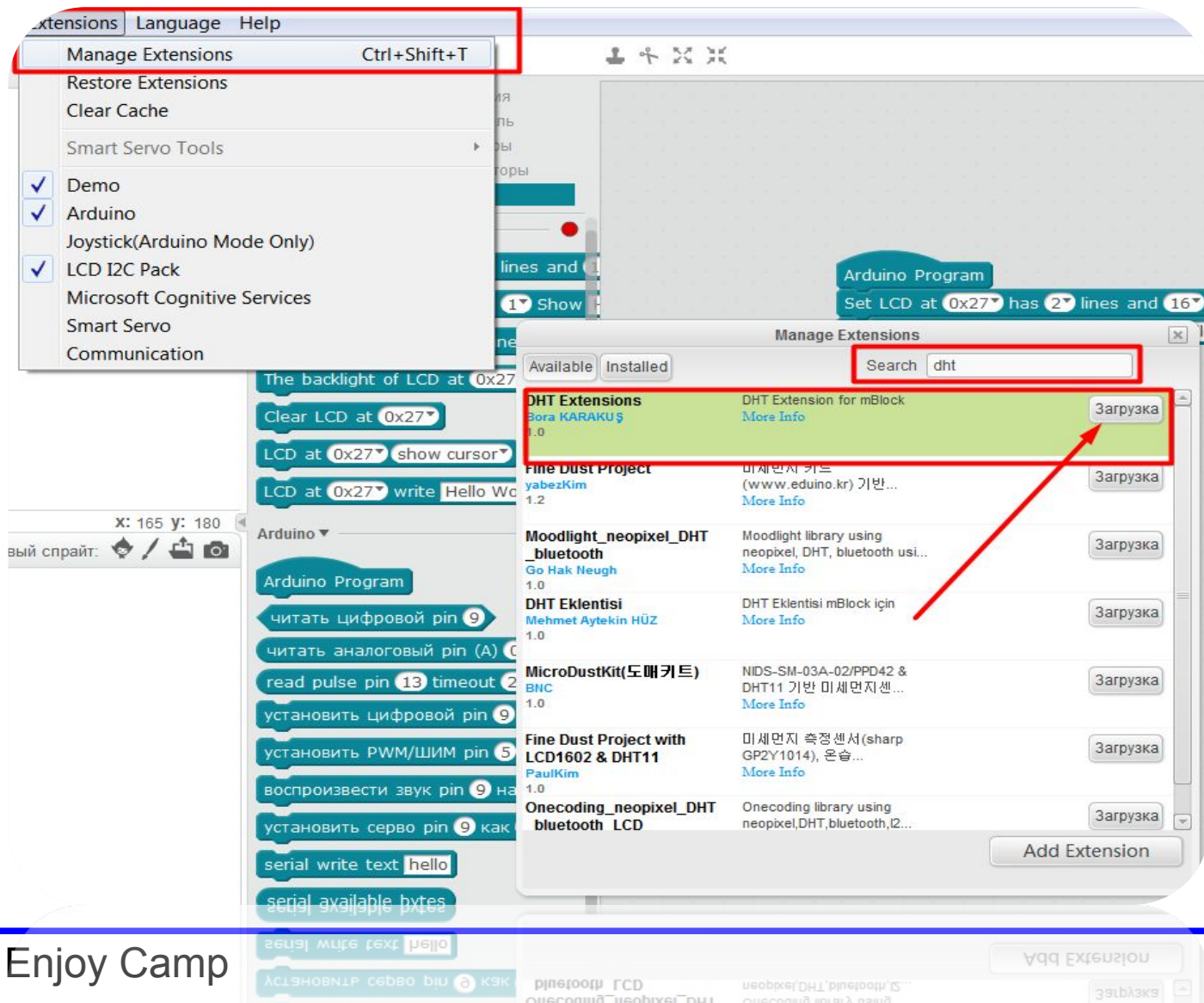
LSD дисплей
+
DHT11
—
Температурный модуль



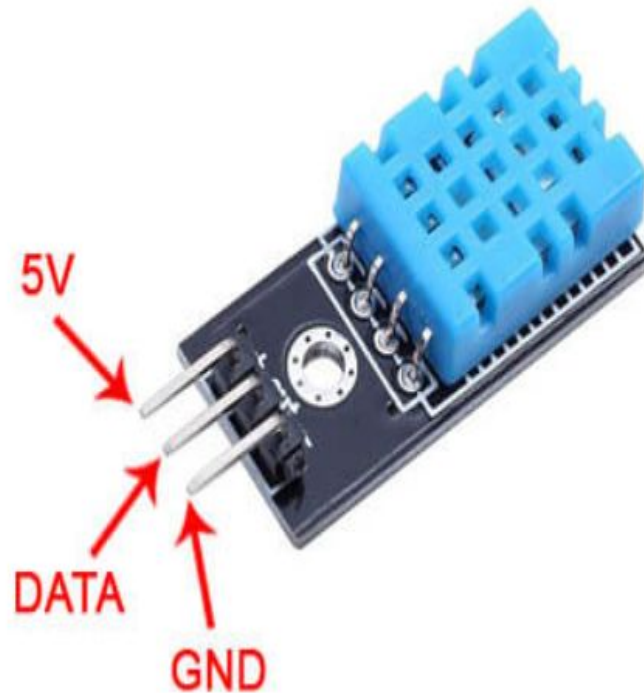
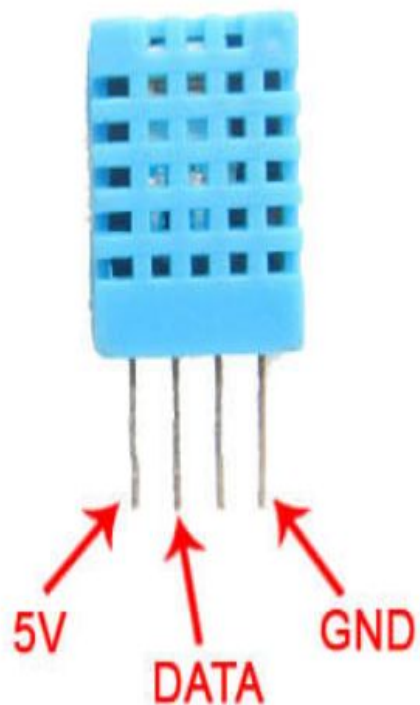
С дисплеем
разобрались, пора
выводить не него
реальные данные
влажности комнаты
и температуры
комнаты. В этом нам
поможет самый
часто используемый
в умных домах
датчик влажности и
температуры CH11



Выполним
подключение
датчика по
указанной схеме
— для удобства
подключения
рекомендуется
вывести землю
(GND) и питание
(5V) на макетную
млату.



Новый датчик —
новая библиотека!
Добавим
библиотеку для
работы с dht через
менеджер
библиотек. Вводить
в поиске следуем
маленькие буквы
dht



В зависимости от типа датчика DHT11 у него может быть разное количество ног. Подключение стоит выполнить в зависимости от типа датчика который у Вас на рабочем месте.

Как и в подключении дисплея, начальные блоки кода остаются прежними



LCD at 0x27 show cursor

LCD at 0x27 write Hello World!

DHT Extensions ▾

DHT No 1, Pin 2, Version DHT11

DHT No 1, Temperature

DHT No 1, Humidity

Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

DHT No 1, Pin 8, Version DHT11

PIN8

Из новой библиотеки dht добавим блок позволяющий очень просто подключить наш модуль к Arduino. Не забудьте поменять пин, к которому подключен датчик - по стандарту библиотеки это pin2, но в нашей схеме это pin8



The screenshot shows the Scratch IDE interface. On the left, the 'Control' menu is highlighted with a red rectangle. Below it, a 'wait 1 seconds' block is visible. In the center, a 'repeat 10 times' loop block is shown. On the right, an 'Arduino Program' block is displayed, containing the following code:

```
Arduino Program
Set LCD at 0x27 has 2 lines and 16 characters per line
DHT No 1, Pin 8, Version DHT11
всегда
```

A red arrow points from the 'всегда' (forever) block in the Scratch code to the 'всегда' block in the Arduino program.

Добавим блок «Всегда» из меню блоков «Контроль» - он отвечает за цикличную работы программного кода.



Перо
Данные и Блоки
Операторы
Робот

перезапустить таймер

LCD I2C Pack ▼

Set LCD at 0x27 has 2 lines and 16 characters per line

LCD at 0x27 Line 1 Col 1 Show

Clear LCD at 0x27 and Line 1 Col 1

The backlight of LCD at 0x27 Turn On

Clear LCD at 0x27

Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

DHT No 1, Pin 8, Version DHT11

всегда

LCD at 0x27 Line 1 Col 1 Show Hello World!

Далее, добавляем уже известный нам с Вами блок позволяющий выводить данные на LCD дисплей



The screenshot shows the Arduino IDE Scratch interface. On the left, a sidebar lists various categories: Движение, Внешность, Звук, Перо, Данные и Блоки, События, Контроль, Сенсоры, **Операторы**, and Робот. Under the 'Операторы' category, several blocks are visible, including a red-bordered box around the 'concatenate' block (labeled 'слить'). The main workspace shows an 'Arduino Program' script with the following blocks: 'Set LCD at 0x27 has 2 lines and 16 characters per line', 'DHT No 1, Pin 8, Version DHT11', a 'always' loop block, and an 'LCD at 0x27 Line 1 Col 1 Show' block. The 'Show' block contains the 'concatenate' block with 'hello' and 'world' as inputs. A red arrow points from the 'concatenate' block in the sidebar to the 'concatenate' block in the 'Show' block.

Оператор «Слить» позволит написать название строки значений, после которого уже будет идти непосредственно само значение



Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

DHT No 1, Pin 8, Version DHT11

всегда

LCD at 0x27 Line 1 Col 1 Show **слить Temp DHT No 1, Temperature**

DHT Extensions ▾

DHT No 1, Pin 2, Version DHT11

DHT No 1, Temperature

DHT No 1, Humidity

Назовем строку значений Temp, после которой следует расположить блок, выводящий значение температуры с датчика DHT11



Clear LCD at 0x27 and Line 1 Col 1

The backlight of LCD at 0x27 Turn On

Clear LCD at 0x27

LCD at 0x27 show cursor

LCD at 0x27 write Hello World!

DHT Extensions ▾

DHT No 1 , Pin 2 , Version DHT11

DHT No 1 , Temperature

DHT No 1 , Humidity

Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

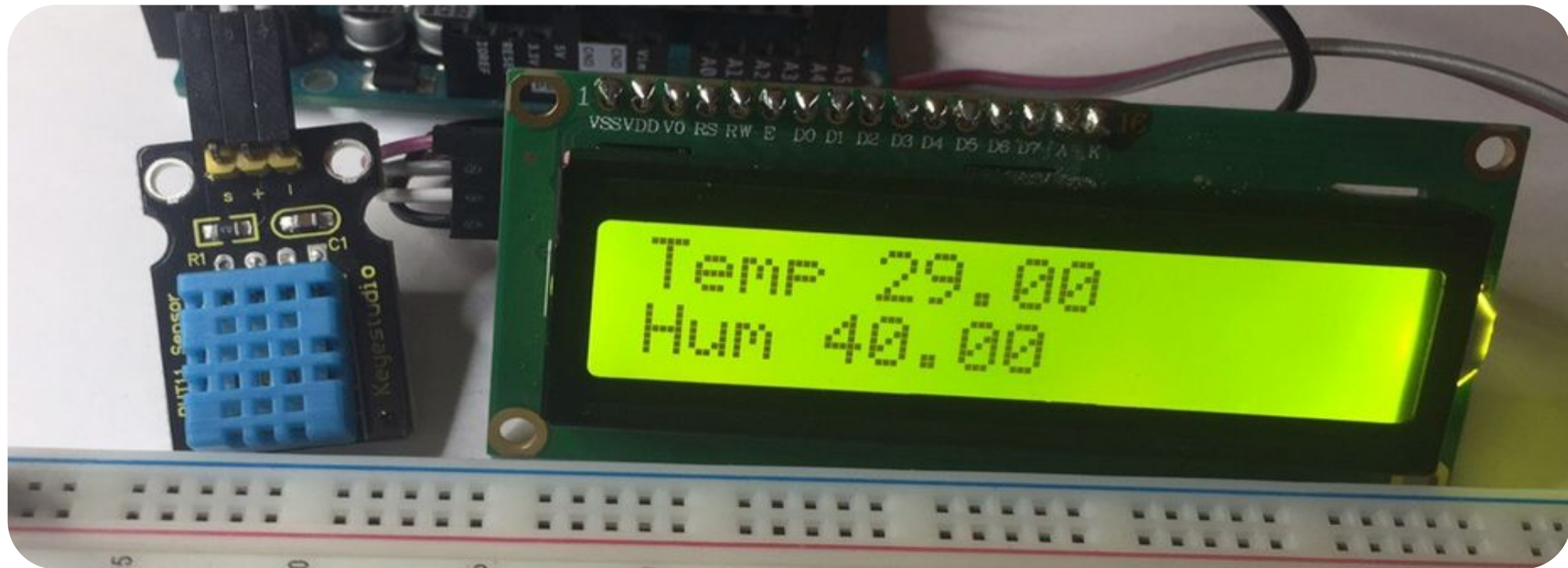
DHT No 1 , Pin 8 , Version DHT11

всегда

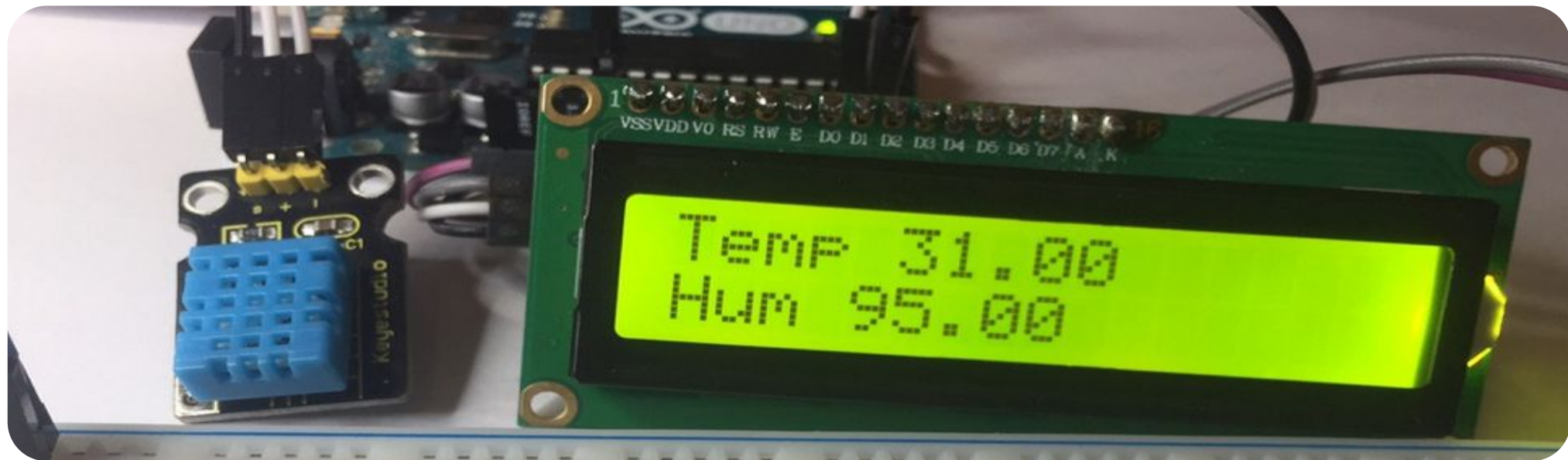
LCD at 0x27 Line 1 Col 1 Show слить Temp DHT No 1 , Temperature

LCD at 0x27 Line 2 Col 1 Show слить Hum DHT No 1 , Humidity

Сделаем все тоже самое, только для значения относительной влажности



Отлично! Теперь мы знаем температуру и влажность нашего умного дома — нормальный диапазон значений 20-30 для температуры, а для влажности 35-60



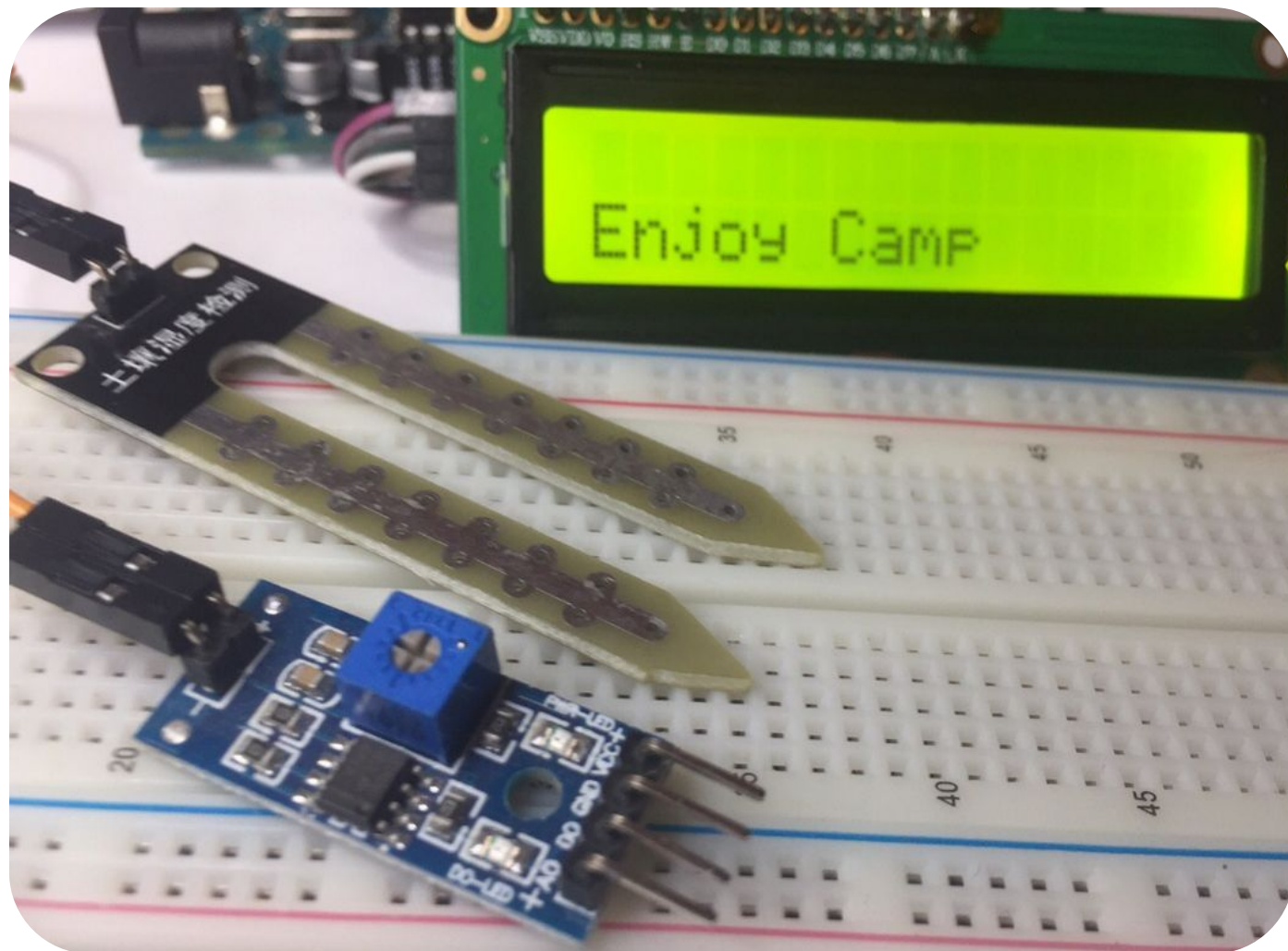
Пока преподаватели достают следующий модуль для изучения, попробуйте подуть на датчик, держа его на расстоянии 5 см от лица — так, можно изменить получаемые показания влажности и температуры



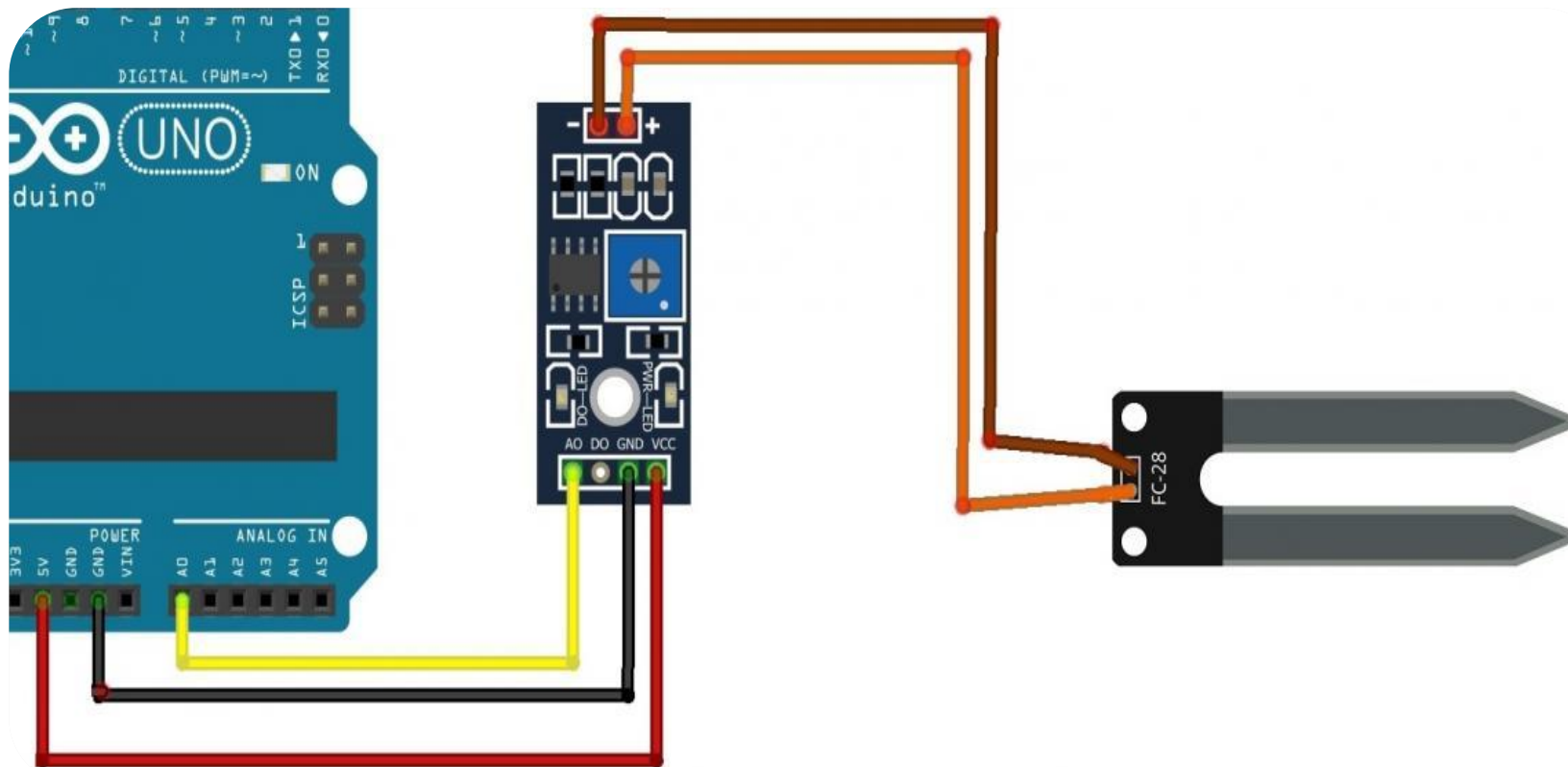
Датчик влажности почвы LM393

—

автополив растений
(гигрометр)



Конечно,
неотъемлемой
частью умного
дома является
автоматический
полив, контроль и
реализацию
которого позволяет
в полной мере
осуществить
цифровой датчик
влажности почвы
LM393



5V – VCC
pinA0 – A0
GND - GND

Датчик в подключении довольно прост – аналоговый порт подключается к аналоговому пину A0, два других к земле и питанию – важно не перепутать их



Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

DHT No 1, Pin 8, Version DHT11

всегда

LCD at 0x27 Line 1 Col 1 Show Hello World!

слить Gigro world

читать аналоговый pin (A) 0

Воспользовавшись уже знакомыми нам блоками, построим алгоритм считывания значений с датчика и выведем их в первую строку нашего LCD дисплея. Строку вывода переменной назовем Gigro, в честь великолепного прибора - гигрометра



Arduino Program

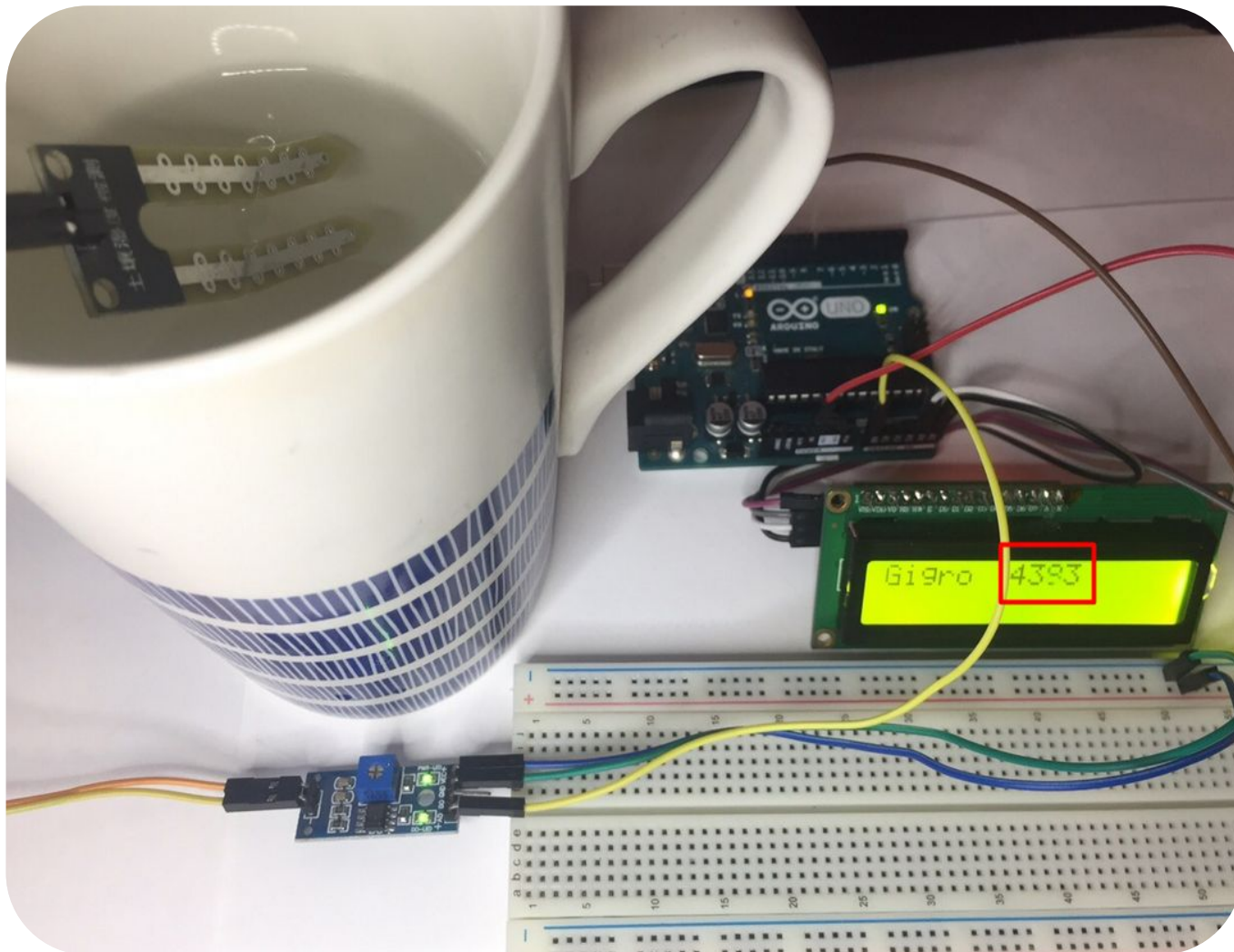
Set LCD at 0x27 has 2 lines and 16 characters per line

DHT No 1, Pin 8, Version DHT11

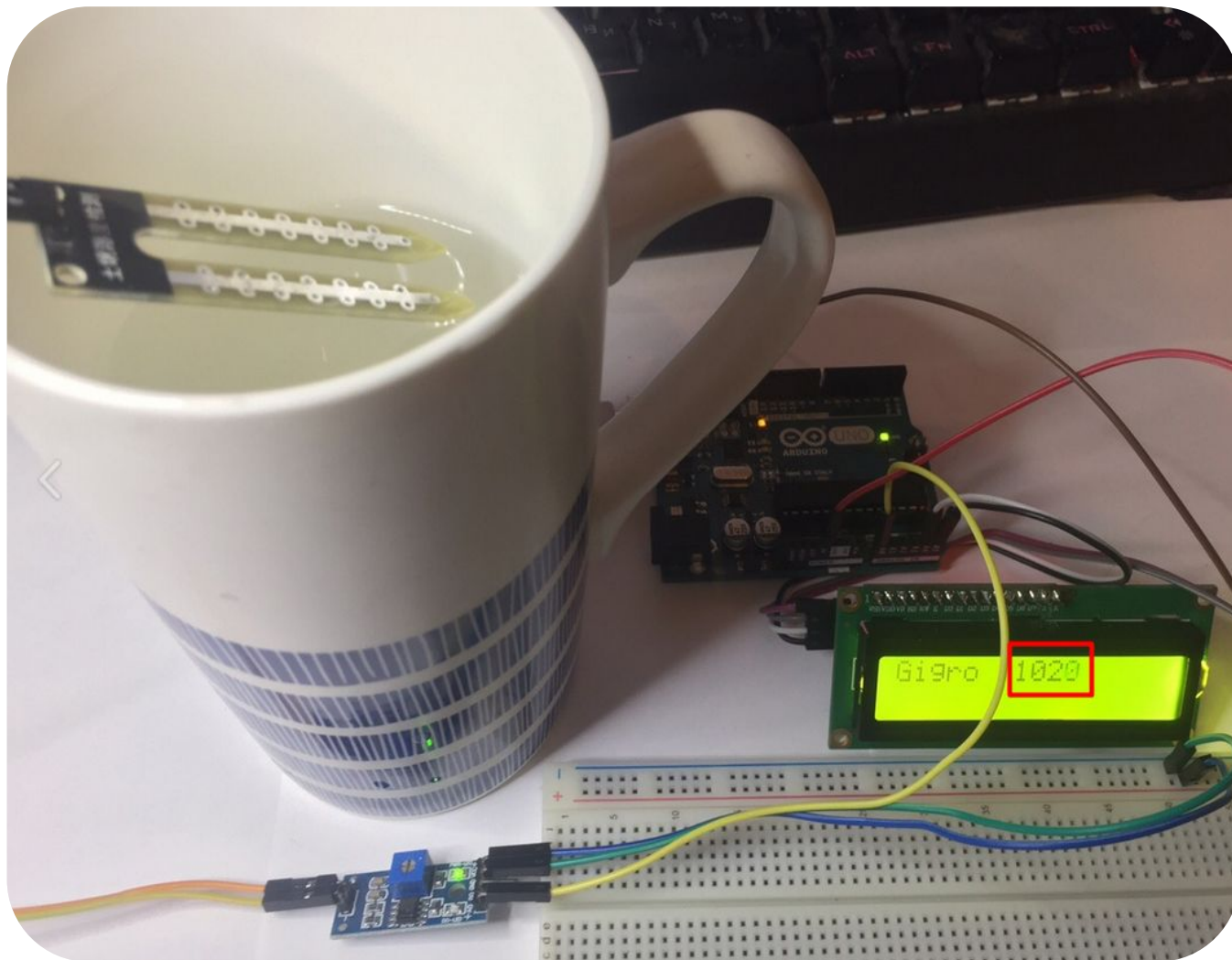
всегда

LCD at 0x27 Line 1 Col 1 Show слить Gigro читать аналоговый pin (A) 0

Отлично! Написанный код из блоков готов к загрузке в плату Arduino



Конечно что бы
проверить
работоспособност
ь датчика, следует
постепенно
погружать и
доставать его из
воды, при этому
будет меняться
выводимое
значение уровня
влажности,
детектируемое
датчиком

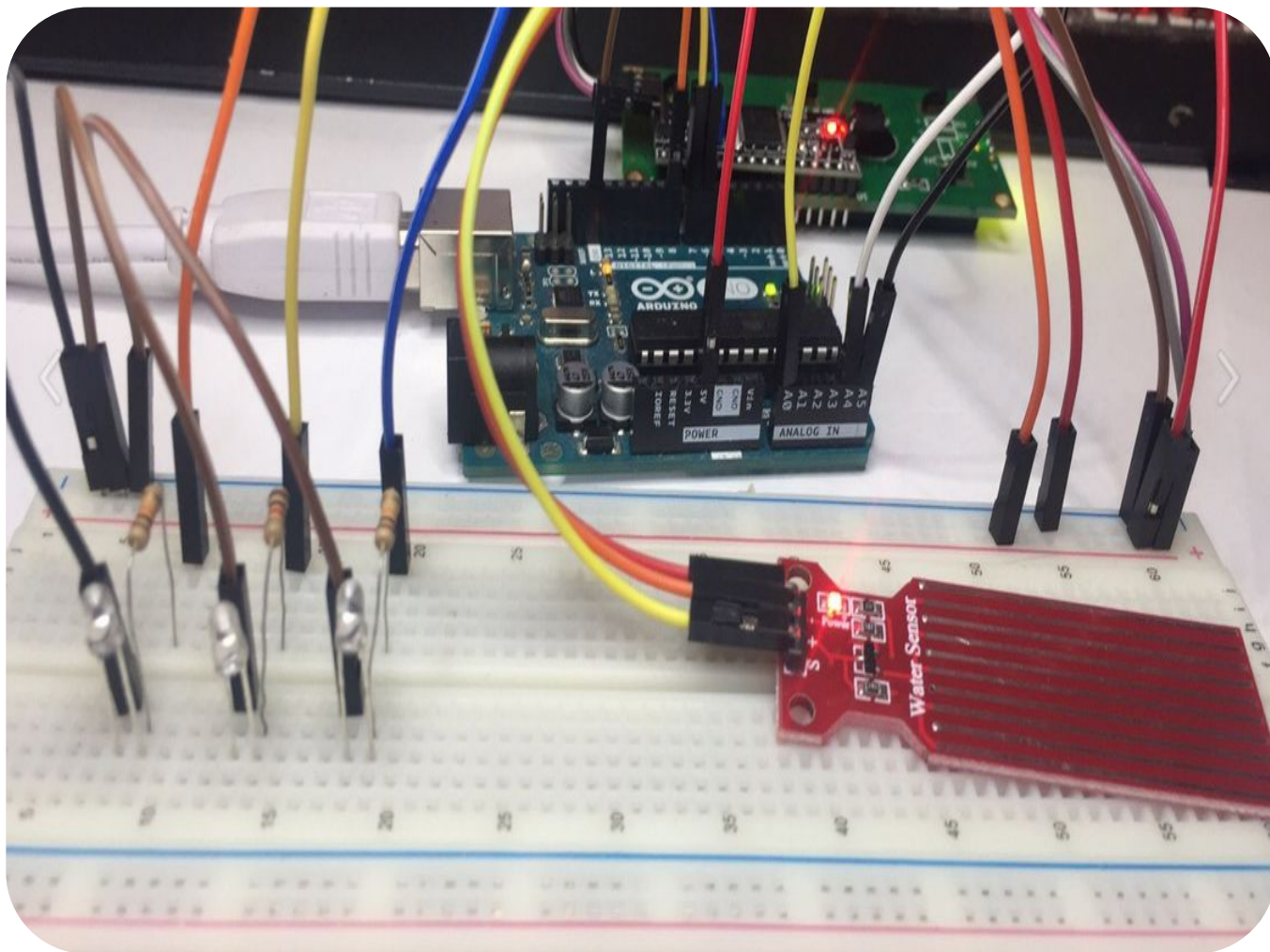


Чем меньше воды в почве
тем меньше уровень
сигнала и наоборот, чем
сильнее полит цветок тем
больше будет значение
влажности

Стоит попросить
преподавателя принести
Вам емкость с водой, для
полноценной реализации
проводимого
эксперимента

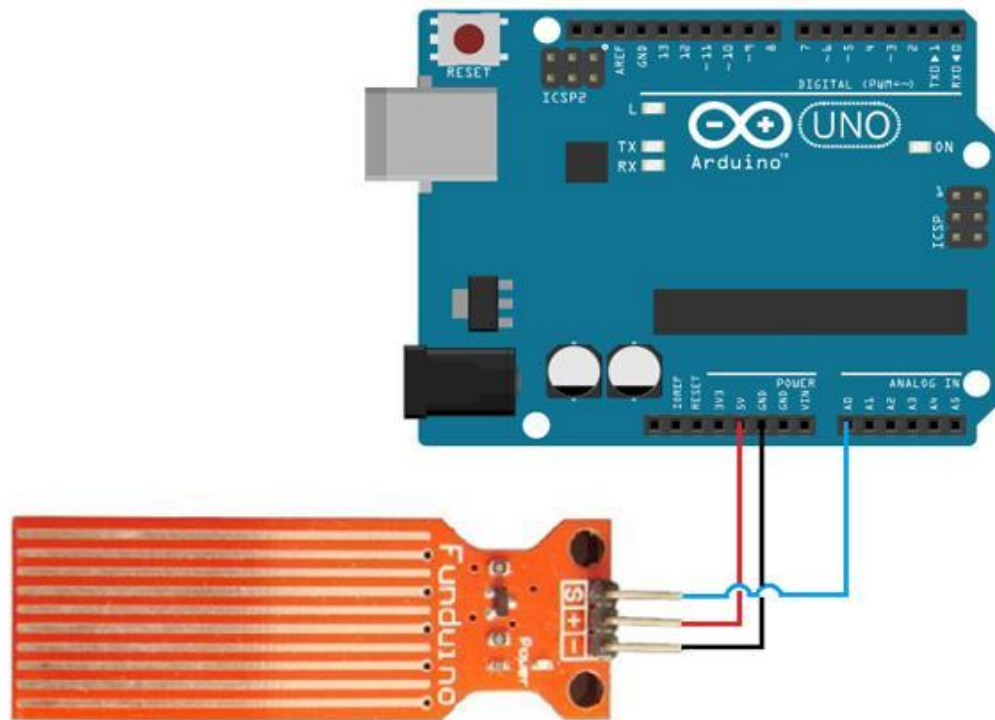


Water sensor — измеритель уровня воды в резервуаре



Что бы полить цветок или например помыться в душе, нужно брать откуда то воду, и не просто брать, а контролировать уровень воды в резервуаре, что бы полностью набрать его при необходимости.

В умных домах этом помогает измеритель уровня воды Water Sensor



LED1 – pin8

LED2 – pin7

LED3 – pin6

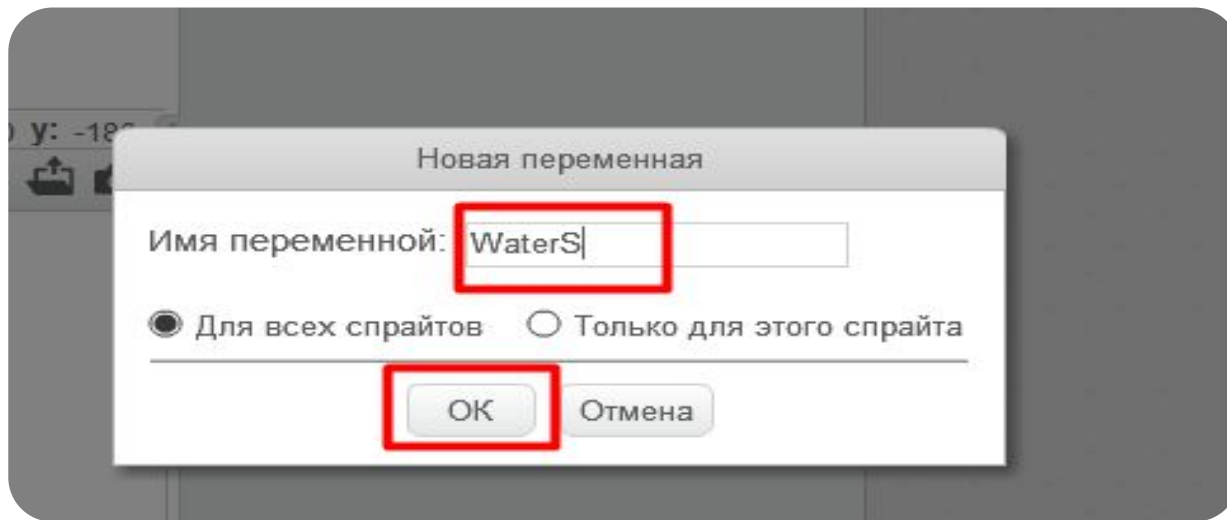
Контакты датчика,
считывая разницу
падения или повышения
сопротивления на них,
без труда
сигнализируют о том что
воды слишком много,
или она закончилась
совсем.

Индикацией уровня
будут служить ТРИ
светодиода – их следует
подключит по известной
Вам схеме



Скриншот интерфейса mBlock. В левой панели видны категории: Скрипты, Костюмы, Звуки. В правой панели — Перо, Операторы, Робот. В центре — панель «События» (События), где выделен блок «когда щелкнут по флажку». В правой панели — панель «Данные и Блоки», где выделен блок «Создать переменную».

Давайте попробуем уговорить прекрасную панду-помощника mBlock сказать нам уровень сигнала датчика. Для этого нужно будет использовать блок «Когда щелкнут по (Флажку)» вместо привычного нам «Arduino Program»



Мы с Вами дошли до создания нашей первой переменной в среде mBlock! Назовем ее WaterS, после поместим ее в блок цикла «Всегда» и изменим ее на уровень сигнала, считываемый Arduino с pinA0



Данные и Блоки Робот

Arduino ▾

Arduino Program

читать цифровой pin 9

читать аналоговый pin (A) 0

read pulse pin 13 timeout 20000

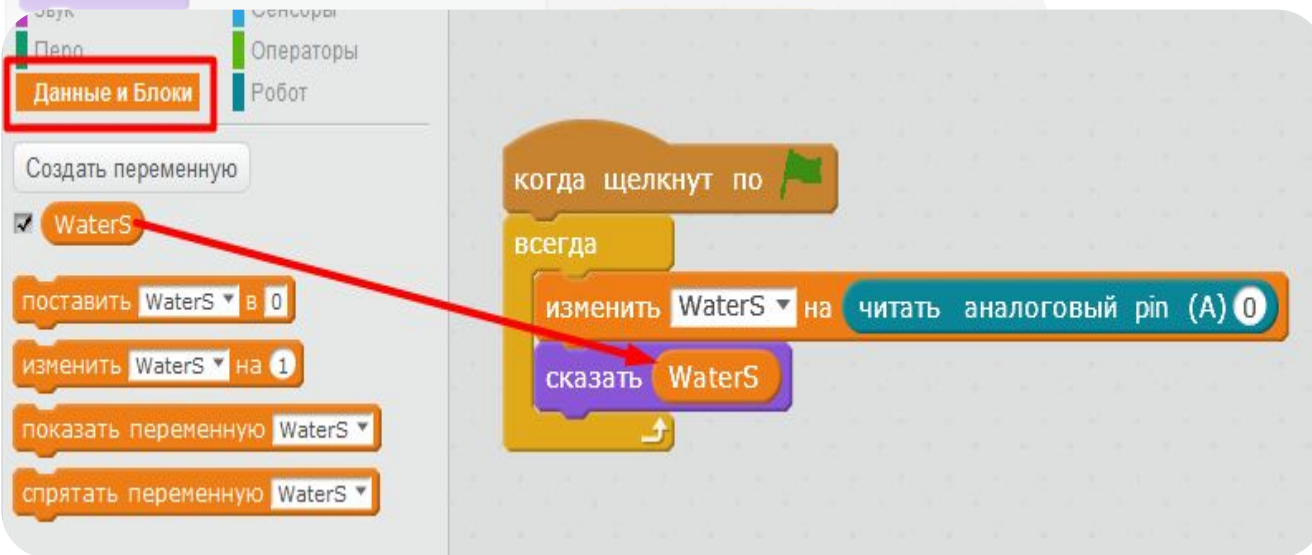
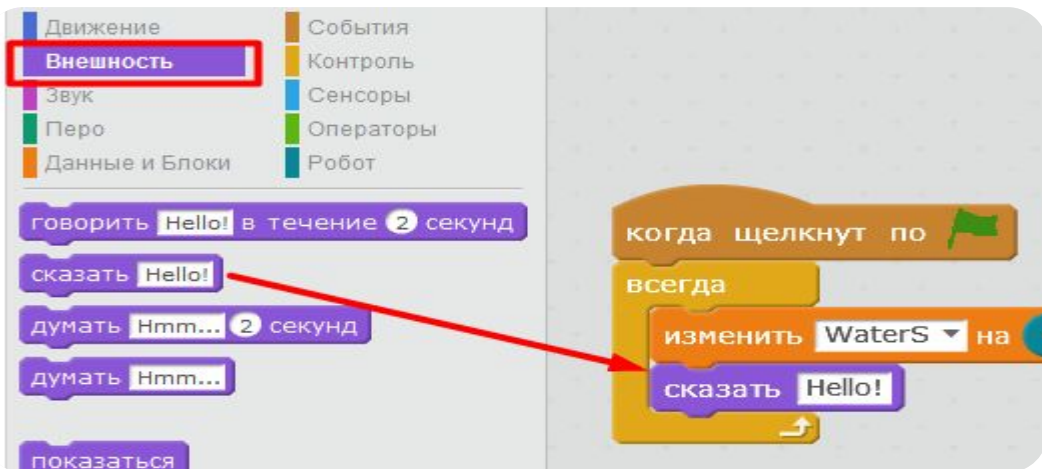
установить цифровой pin 9 udgang к

когда щелкнут по [флаг]

всегда

изменить WaterS на читать аналоговый pin (A) 0

Установим блок считывания сигнала в блок цикла «Всегда»



В меню блоков выберем «Внешность» - нас интересует блок «Сказать», только вместо Привет панда mBlock должна сообщить нам переменную созданную нами для хранения в ней значения с датчика уровня жидкости

Untitled

WaterS 70

70

Скрипты Костюмы Звуки

Движение Внешность Звук Перо Данные и Блоки

События Контроль Сенсоры Операторы Робот

Создать переменную

☒ WaterS

поставить WaterS в 0

изменить WaterS на 1

показать переменную WaterS

спрятать переменную WaterS

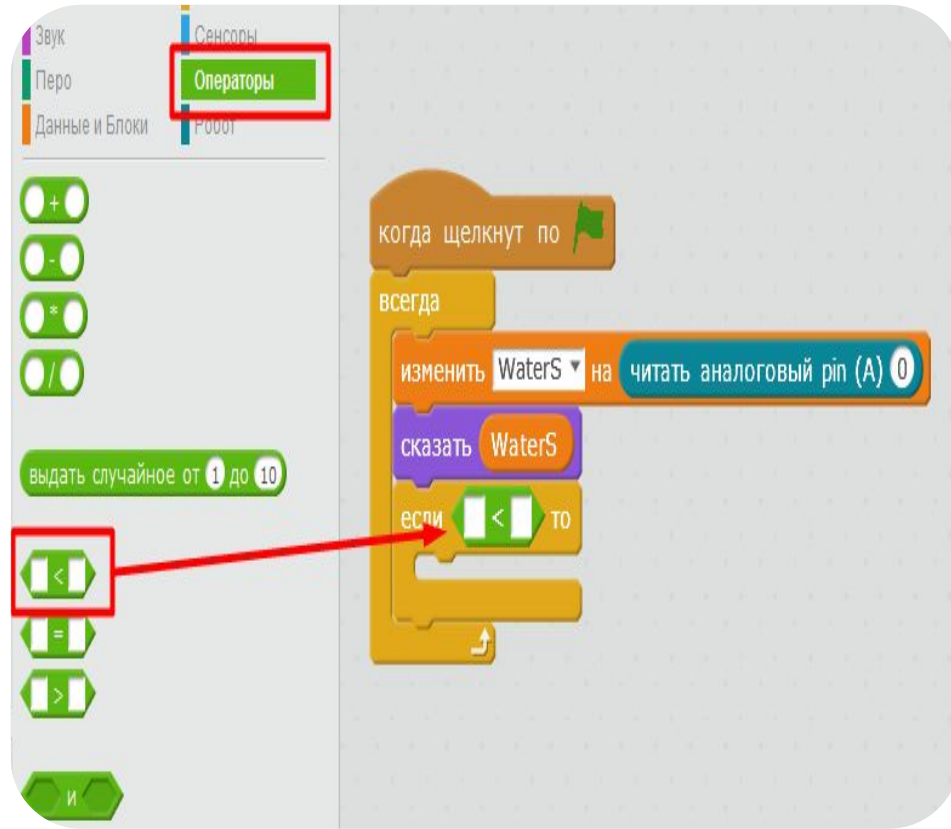
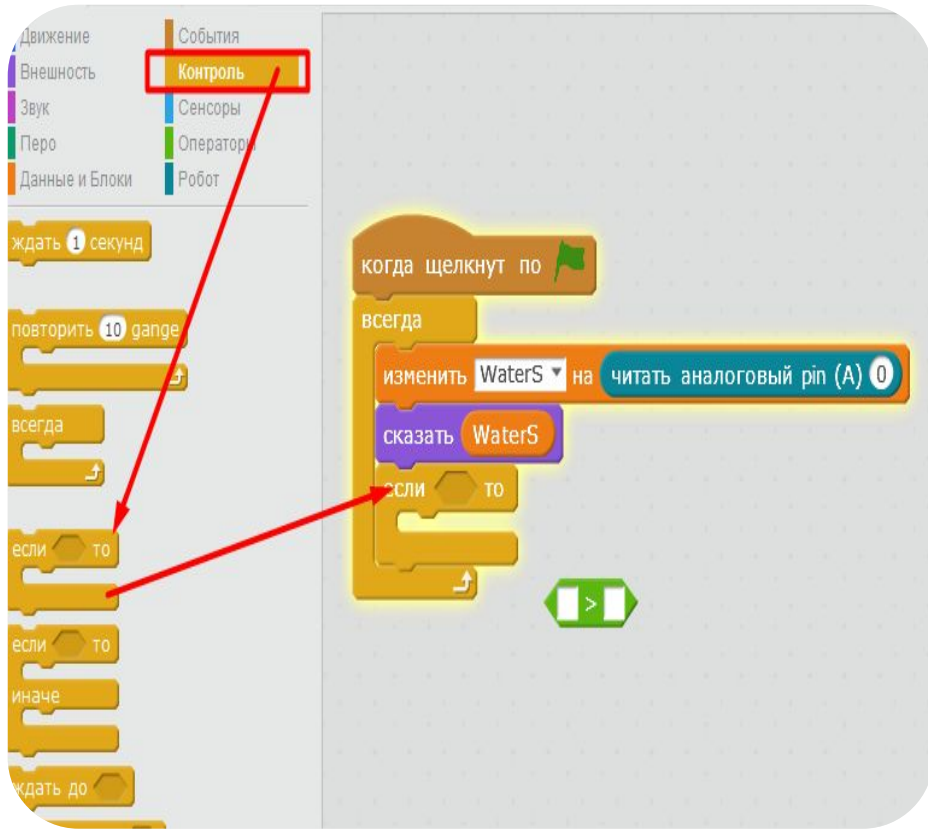
когда щелкнут по [флаг]

всегда

изменить WaterS на [читать аналоговый pin (A) 0]

сказать WaterS

Отлично! Теперь мы знаем стартовое значение, на которое мы будем опираться в создании нашей программы. Знать его нужно для того, что бы знать какой уровень сигнала соответствует логическому «0» датчика уровня воды



Добавляем условий и логических операторов (ну куда же без них)



Данные и Блоки Робот

LCD I2C Pack ▾

Arduino ▾

Arduino Program

читать цифровой pin 9

читать аналоговый pin (A) 0

read pulse pin 13 timeout 20000

установить цифровой pin 9 udgang как HIGH ▾

установить PWM/ШИМ pin 5 udgang как 0 ▾

воспроизвести звук pin 9 на ноте C4 ▾ такт 1/2 ▾

Arduino Program

всегда

поставить WaterS ▾ в читать аналоговый pin (A) 0

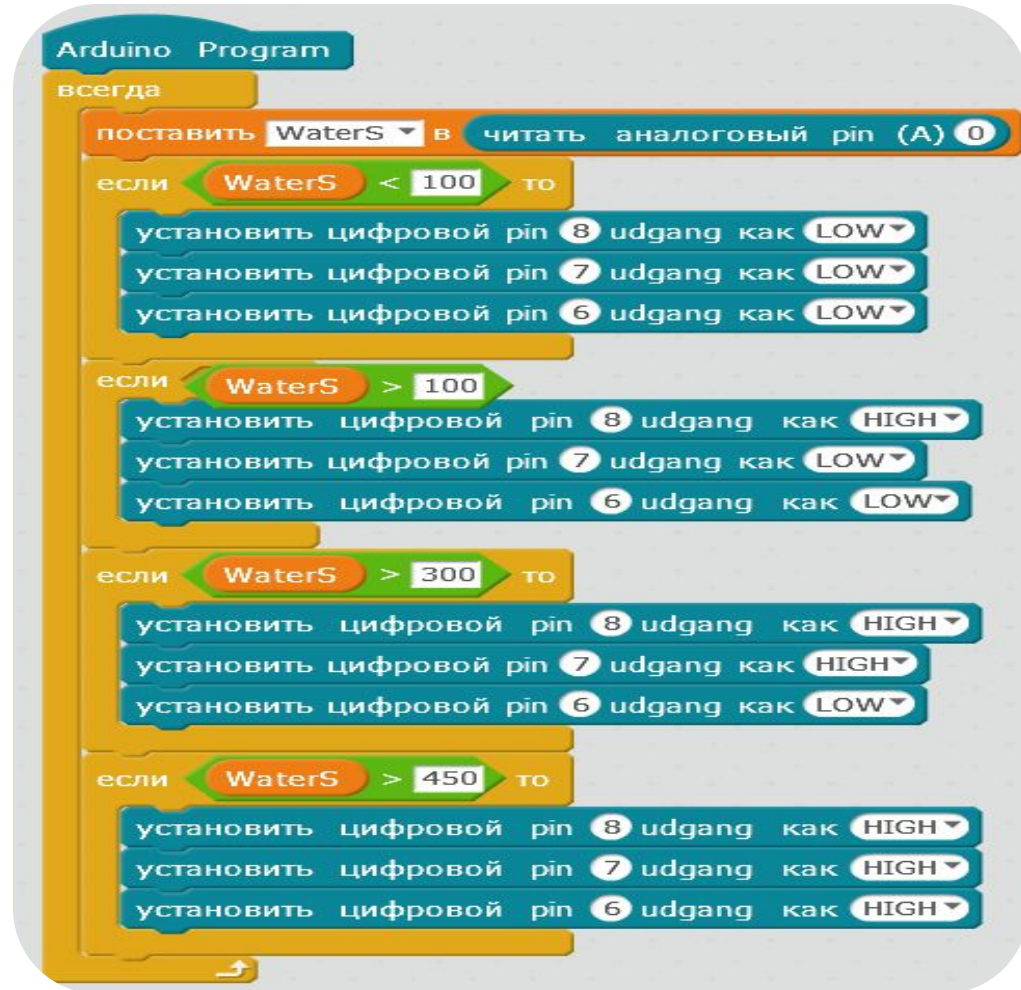
если WaterS < 100 то

установить цифровой pin 8 udgang как LOW ▾

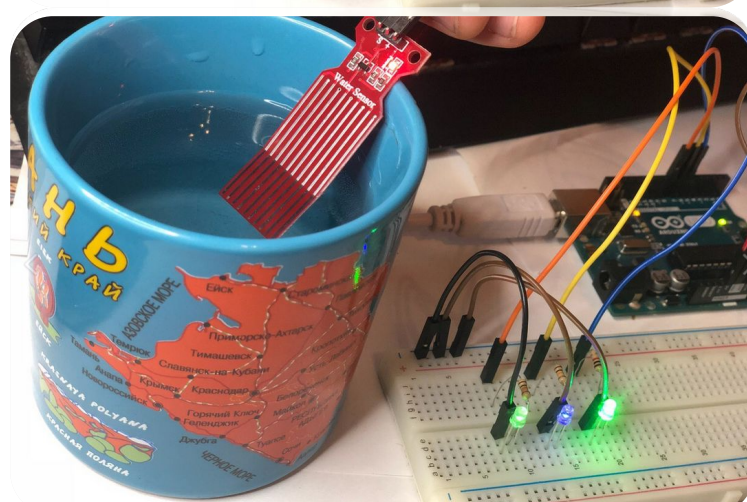
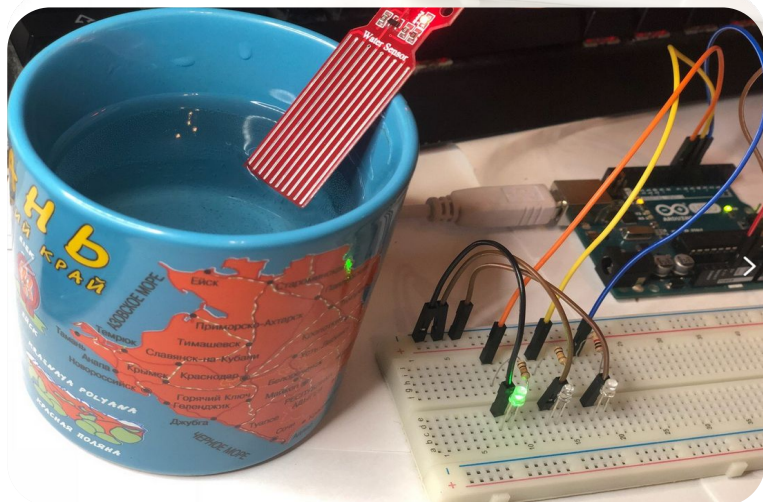
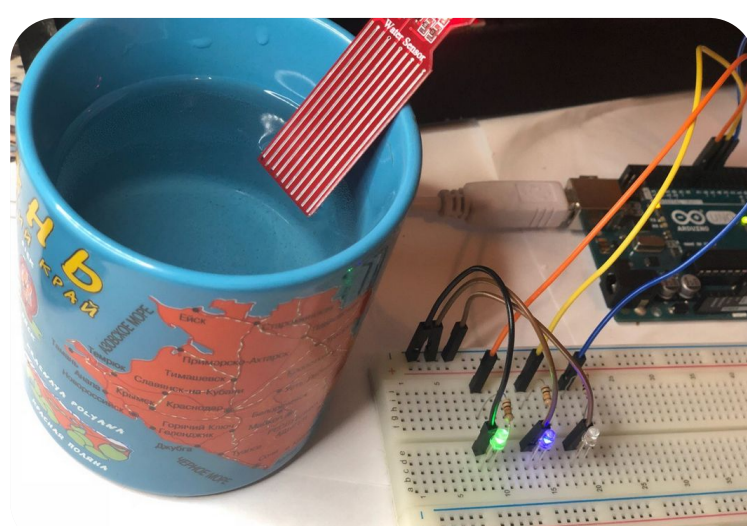
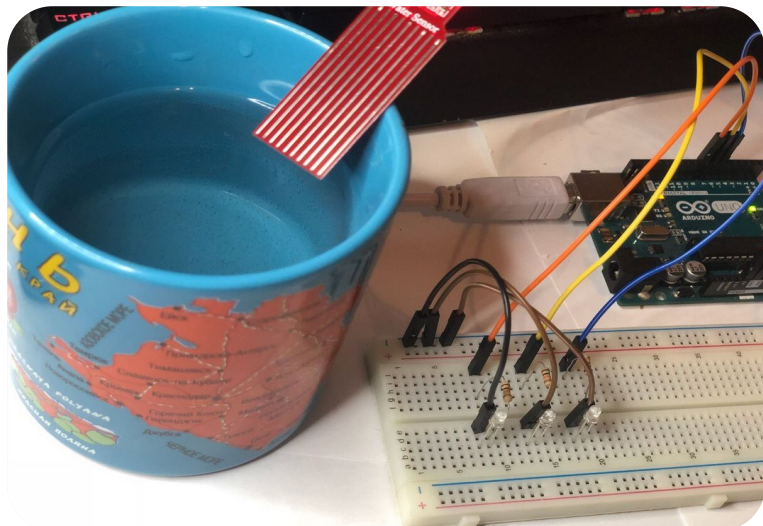
установить цифровой pin 7 udgang как LOW ▾

установить цифровой pin 6 udgang как LOW ▾

Настраиваем порты к которым подключены светодиоды в правильный режим работы: если сигнал меньше 100(стартовый порог который нам сказала панда) то светодиоды совсем не горят(LOW). Когда светодиоды не светятся это говорит нам о том, что жидкость в резервуаре закончилась



Ну тут придется немного похитрить. Добавляем пару аналогичных циклов для выполнения условий – если уровень сигнала больше 100 – горит светодиод на 8 порту, если больше 300 то светодиод загорается и на 7, ну а если уровень сигнала 450 то горят все три светодиода, что говорит о полном заполнении нашего резервуара с водой



Как всегда,
просим вожаков
помочь Вам
погрузить
датчики в
заранее
подготовленные
резервуары с
водой



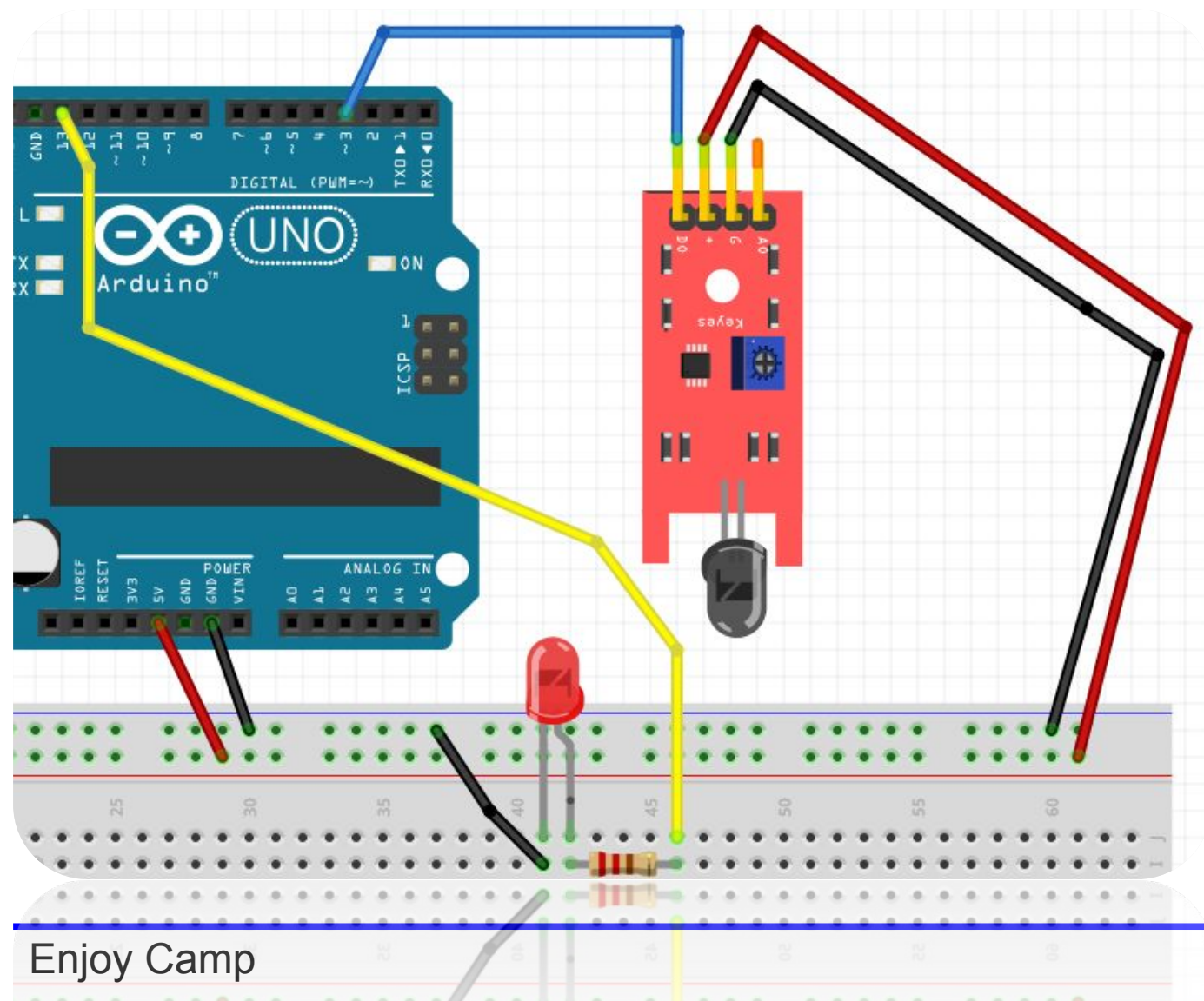
Fire sensor
(датчик пламени)

—

Пожарная безопасность
умного дома

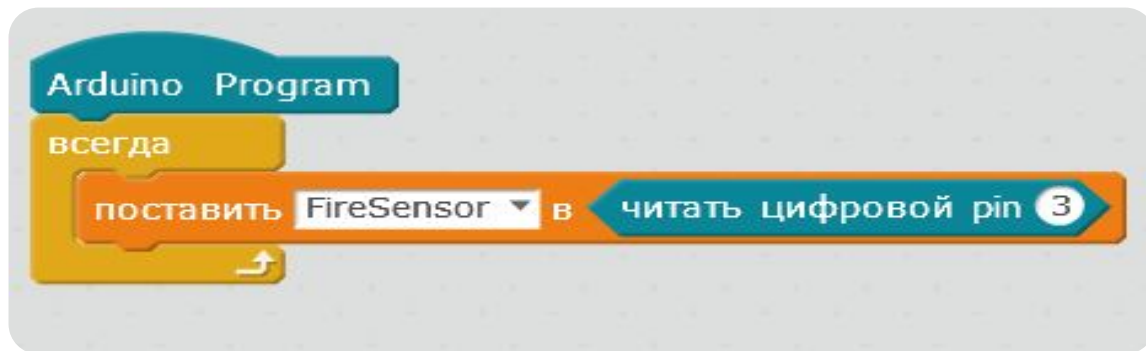
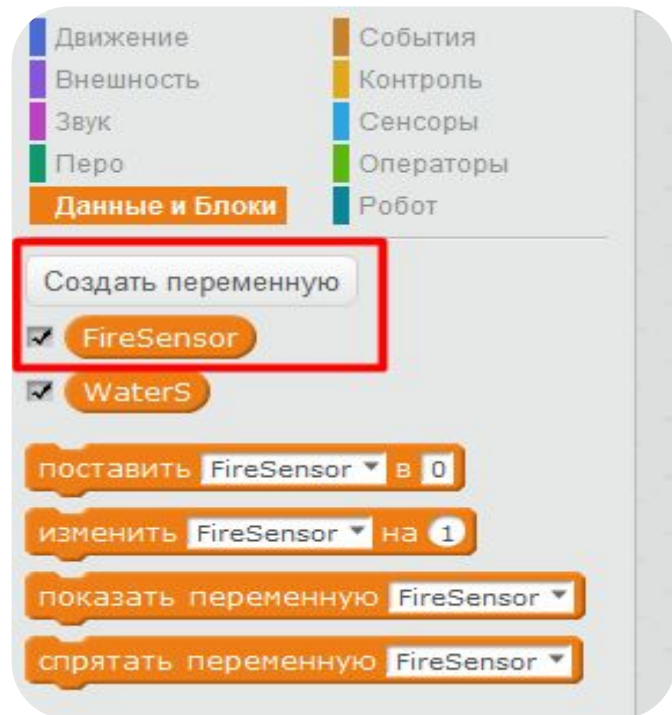


Пожарная
без опасть
умного дома
напрямую
связанна с
датчиками
огня



Подключаемый датчик
позволяет
детектировать очаги
открытого огня в
радиусе 3м

5V – VCC
Pin3 – D0
GND – GND
Pin13 – LED



Новый датчик – новая переменная FireSensor. Так как датчик подключен к pin3, то и считывать показания будем конечно с цифрового pin3



Arduino Program

всегда

поставить FireSensor в читать цифровой pin 3

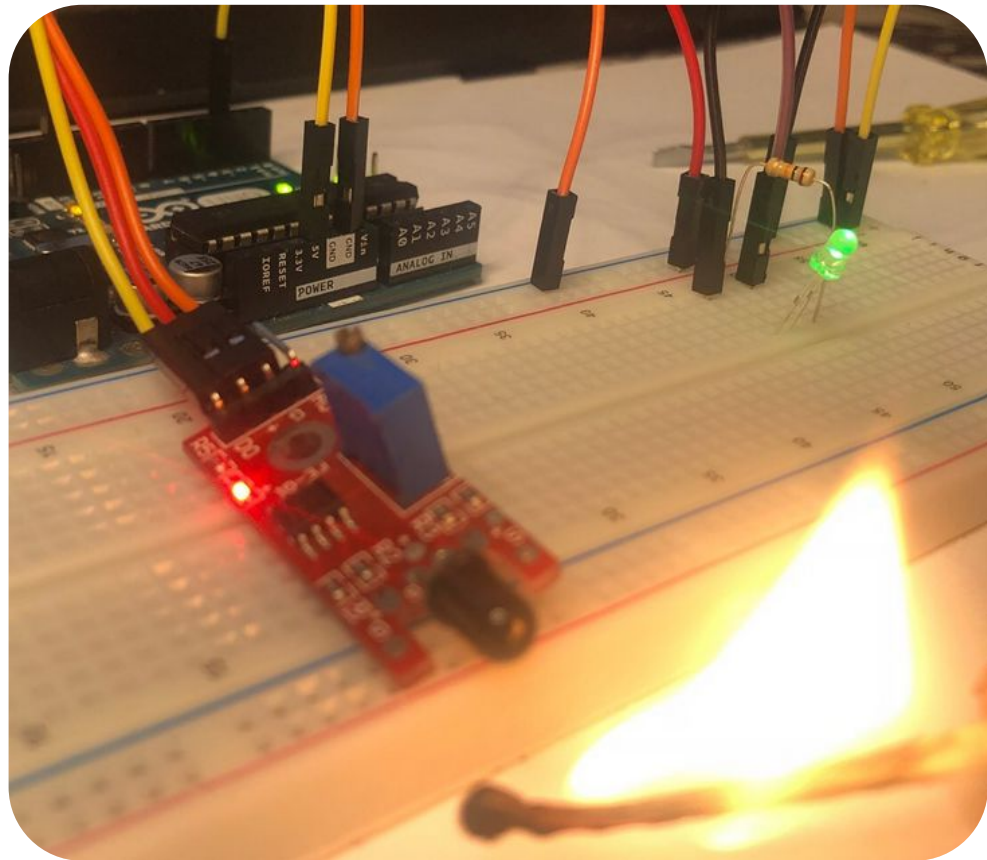
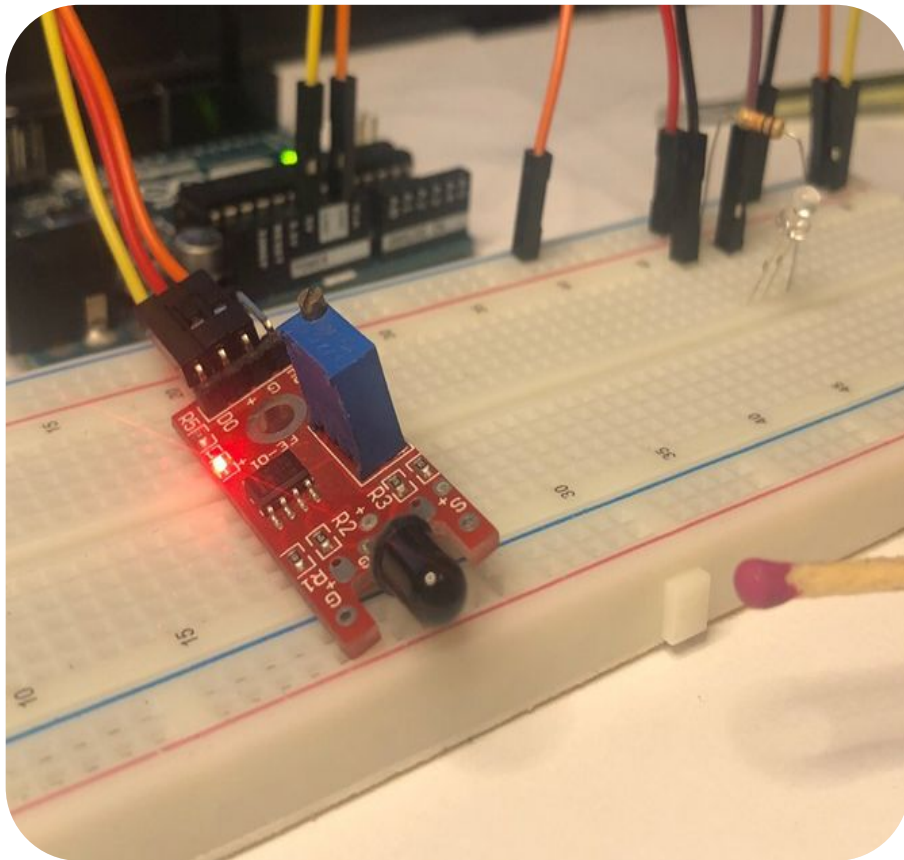
если FireSensor = 1 то

установить цифровой pin 13 udgang как HIGH

иначе

установить цифровой pin 13 udgang как LOW

Составим простой
блоковый алгоритм,
который будет
включать светодиод
на 13 порту, когда
датчик фиксирует
наличие теплового
излучения т.е.
открытого огня



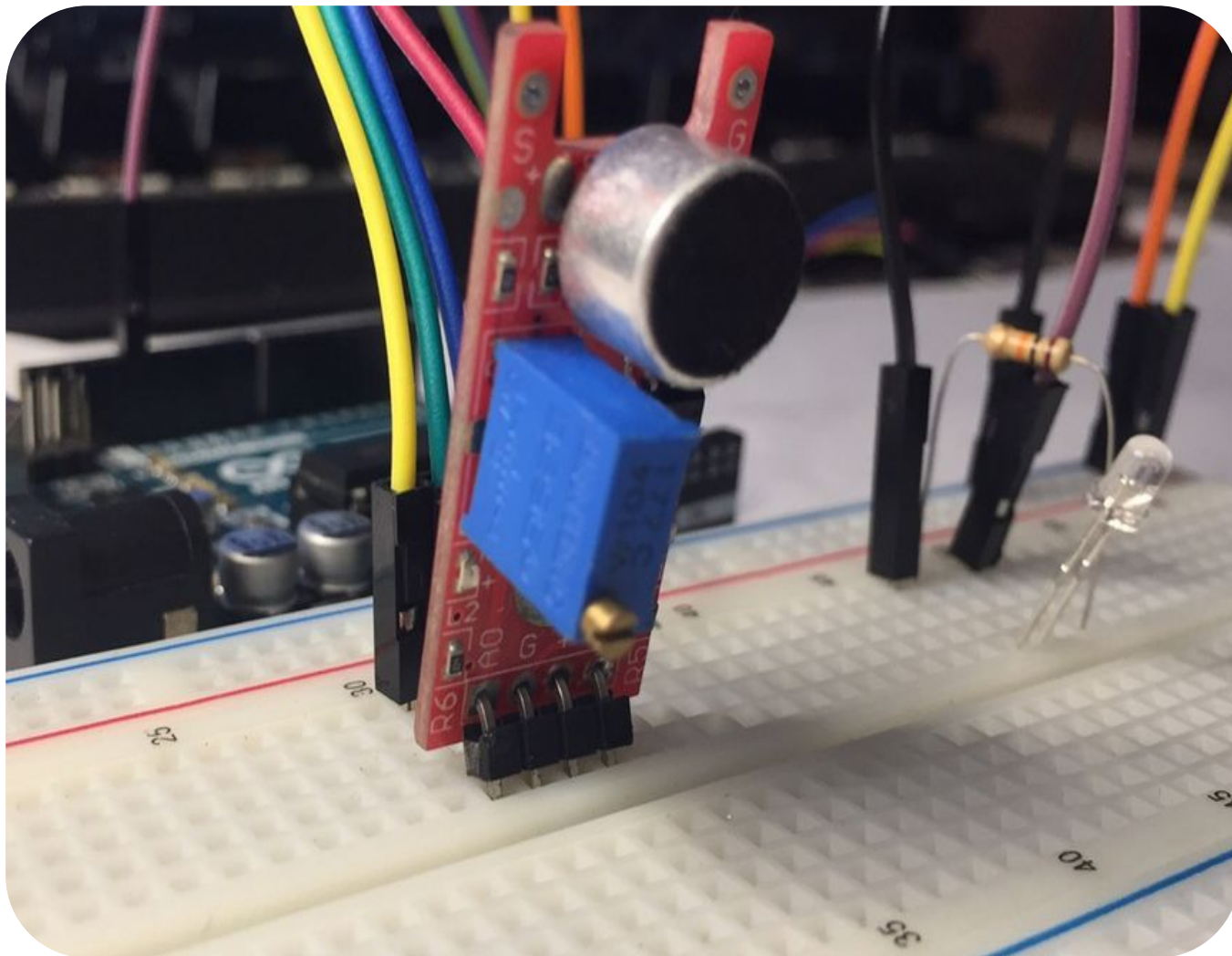
Детям спички не игрушка! Самим поджигать естественно ничего не стоит, просим преподавателей помочь Вам в этом нелегком деле.



КУ-037
(датчик звука)

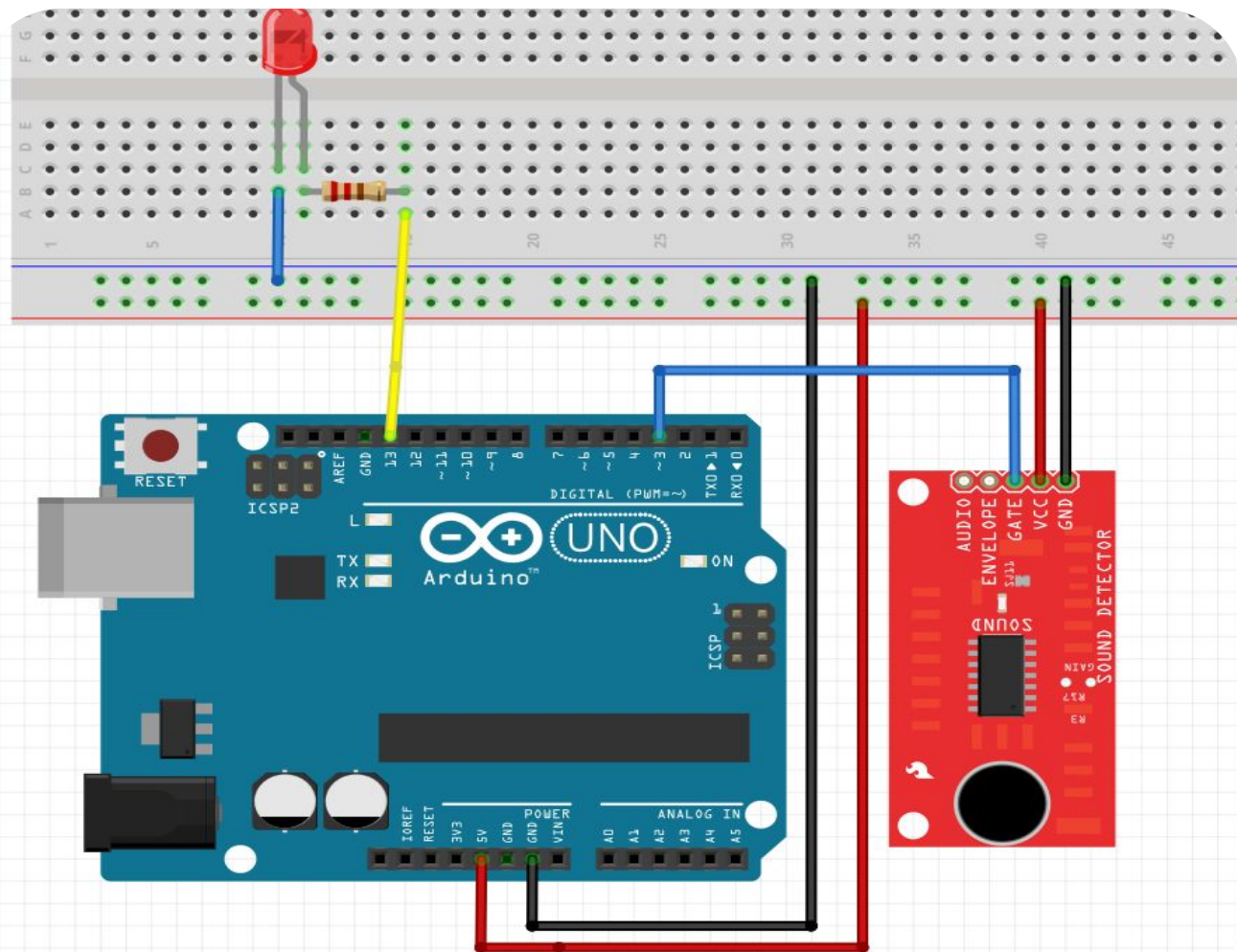
—

Выключение и включение
света по хлопку



Данный модуль может
быть настроен на прием
различной силы
звуковых частот –
например, можно
сделать выключение
света по хлопку.

Давайте попробуем!



При подключении A0 датчика нам не понадобится, а вот D0 стоит подключить к 3 порту Arduino

5V – VCC
Pin3 – D0
GND – GND
Pin13 – LED



Создать переменную

- ☒ Clap
- ☒ FireSensor
- ☒ WaterS

поставить Clap в 0

изменить Clap на 1

показать переменную Clap

спрятать переменную Clap

Arduino Program

всегда

поставить Clap в 0

Arduino Program

всегда

поставить Clap в читать цифровой pin 3

Новый модуль, новая переменная – создадим переменную Clap значение которой будем получать считывая цифровой порт pin3



Arduino Program

всегда

поставить Clap в читать цифровой pin 3

если Clap = 1 то

установить цифровой pin 13 udgang как HIGH

иначе

установить цифровой pin 13 udgang как LOW

ждать 2 секунд



Далее установим уже известную логику «Если – Иначе»

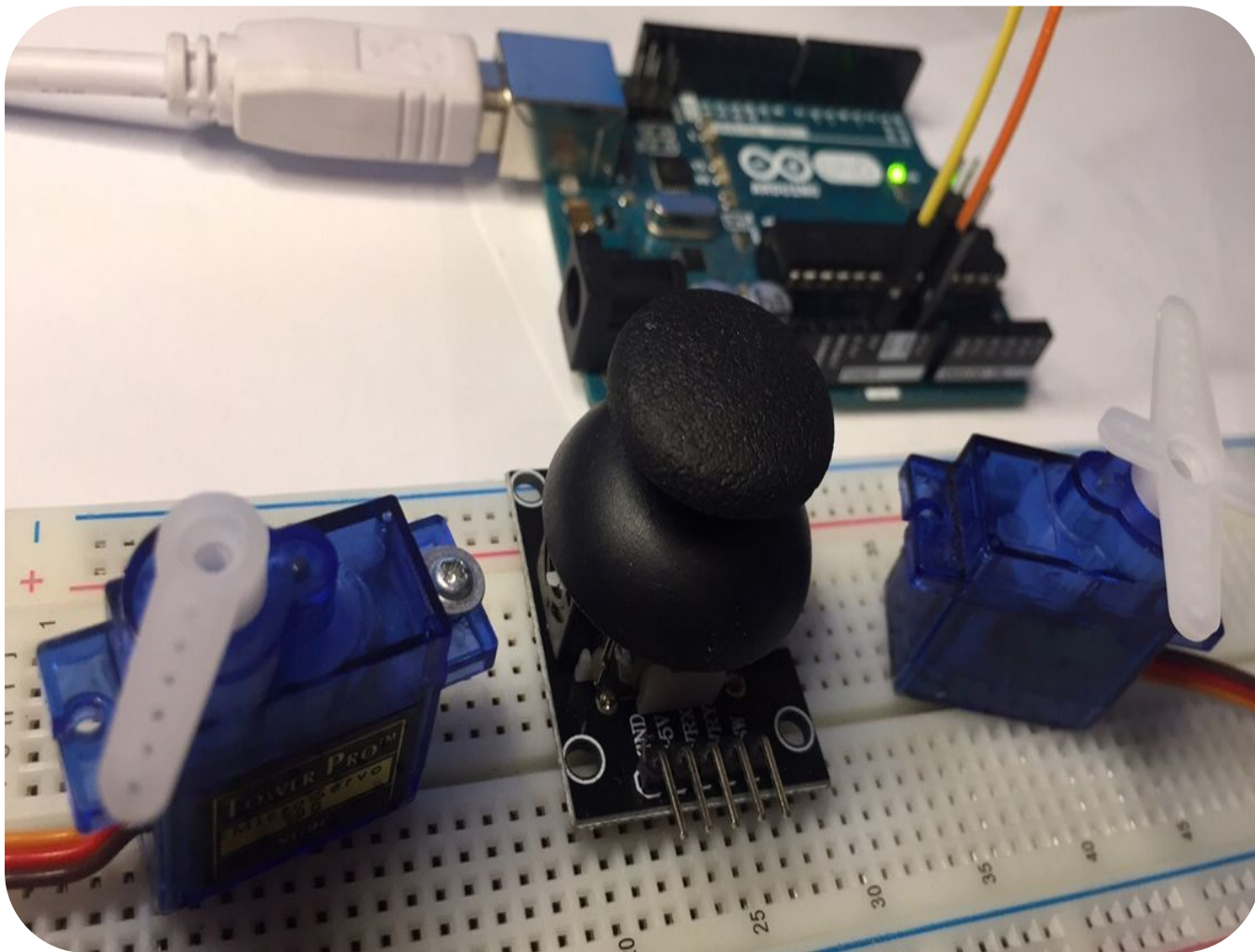
Уровень восприятия звуковых волн можно регулировать плавно крутя потенциометр на самом датчике – так можно добиться необходимой реакции на хлопок



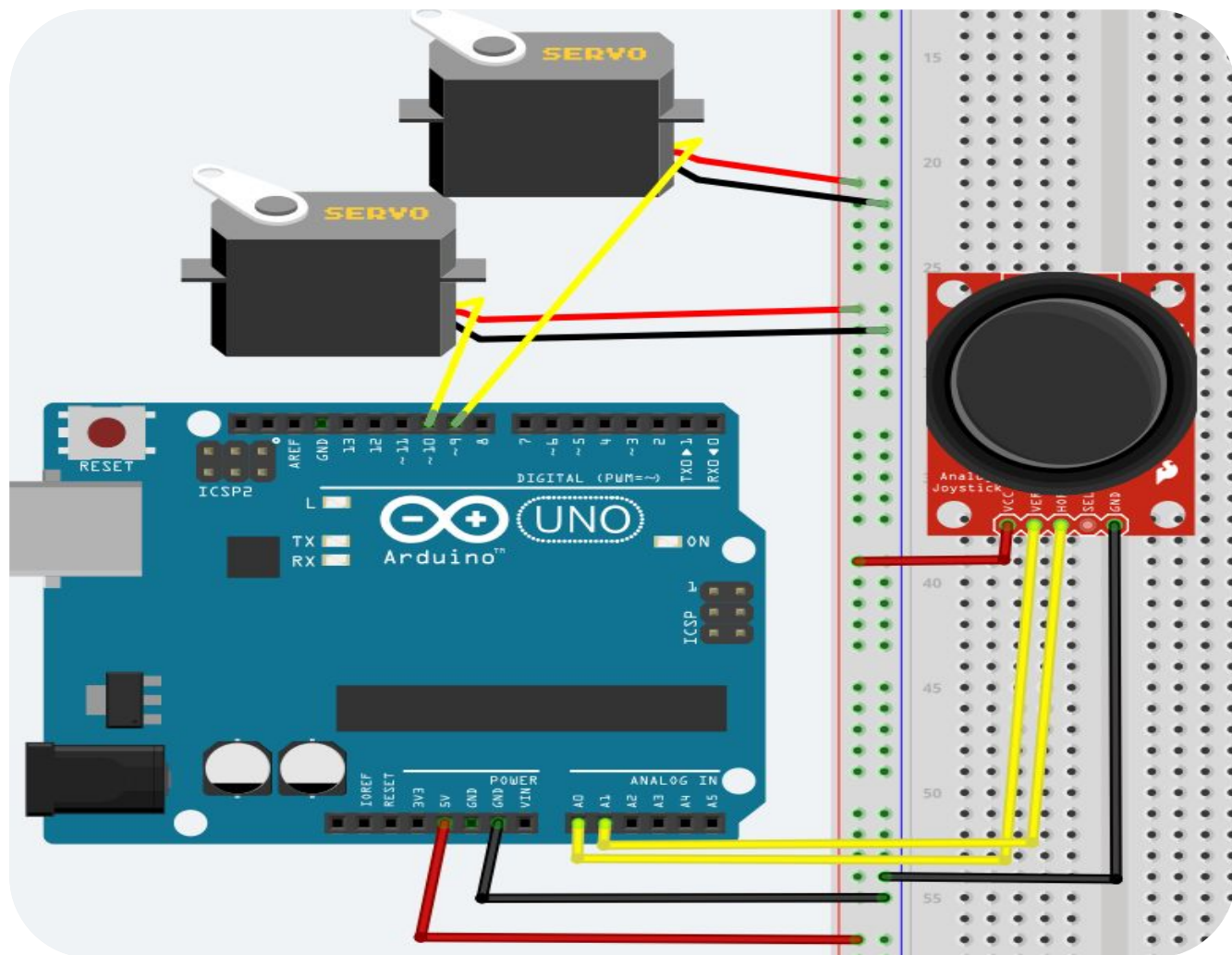
Джойстик

—

Управление периферией
умного дома
(2 Servo SG-90)

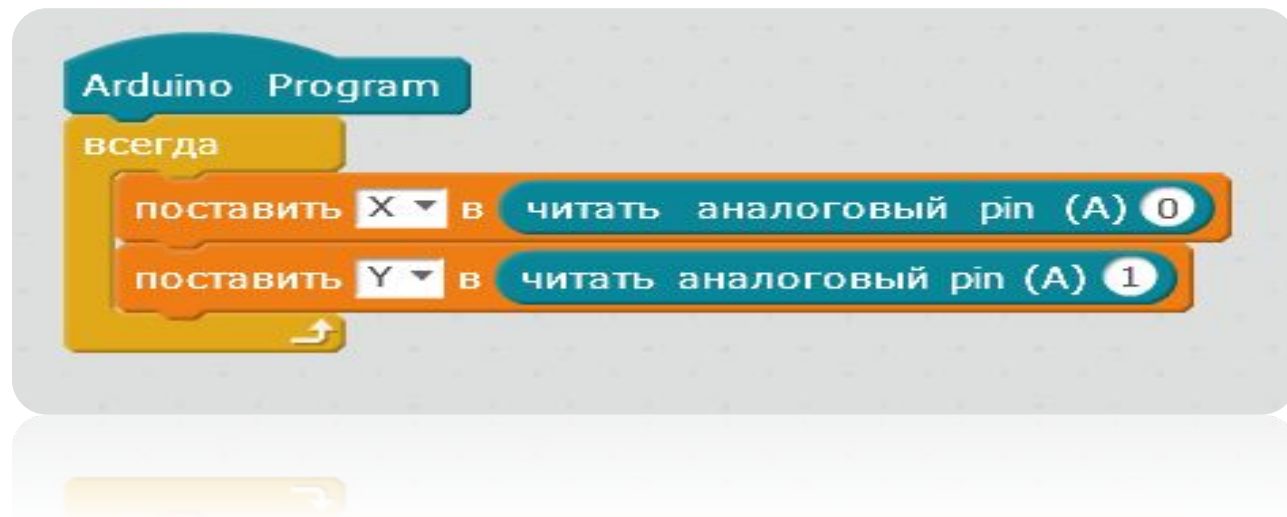
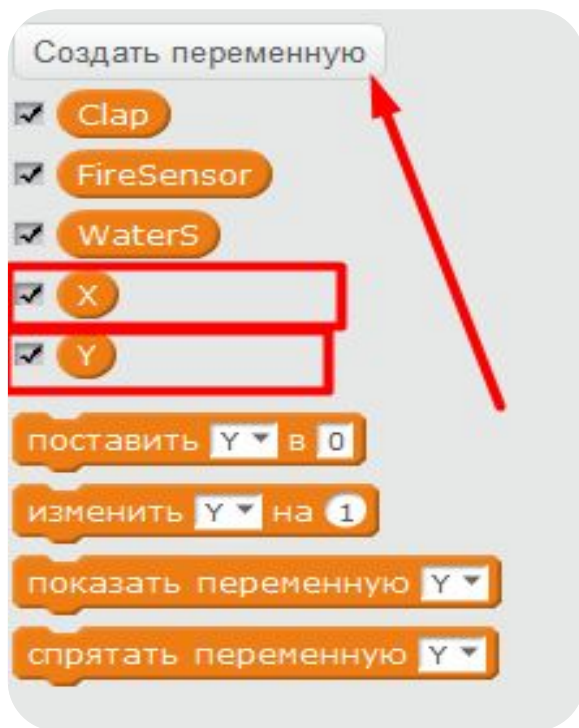


Конечно, джойстик можно использовать как облегченный инструмент управления. Например, можно переключать пункты меню, которые выводятся на LCD монитор или сделать с помощью них игру. Мы же с вами постараемся плавно управлять двумя сервоприводами



Выход джойстика
который мы не
задействуем отвечает за
нажатие на джойстик.

Servo1 – pin9
Servo2 – pin10
GND – GND
VCC – 5V
VRX – A1
VRY – A2



Создадим переменные отвечающие за направление движений джойстика по осям – X и Y. В этих переменных будут храниться значения уровней сигнала аналоговых входов подключенных к джойстику



Arduino Program

всегда

поставить X в читать аналоговый pin (A) 0 / 6

поставить Y в читать аналоговый pin (A) 1 / 6

Arduino Program

всегда

поставить X в читать аналоговый pin (A) 0 / 6

поставить Y в читать аналоговый pin (A) 1 / 6

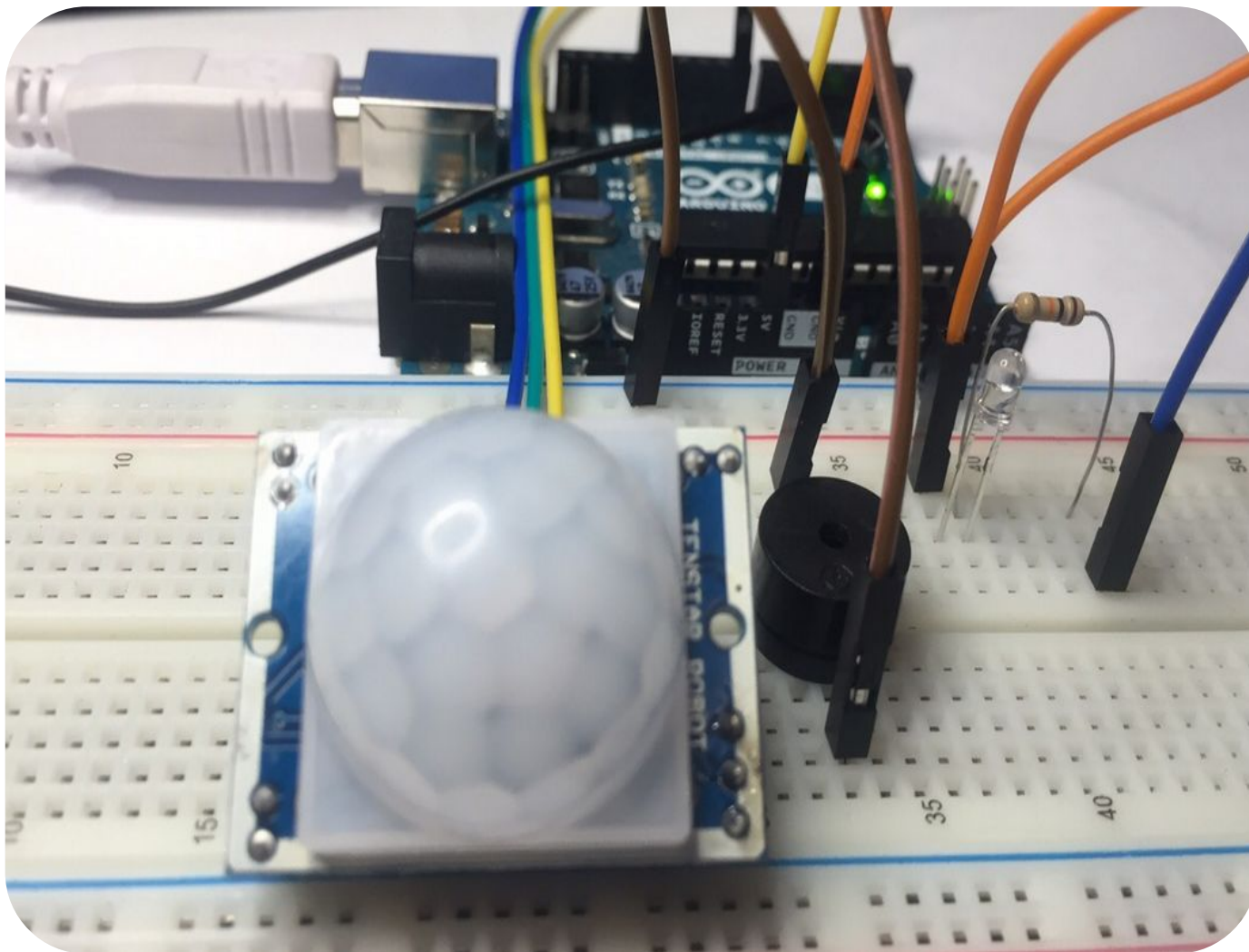
установить PWM/ШИМ pin 9 udgang как X

установить PWM/ШИМ pin 10 udgang как Y

Очень важно, разделить получаемый сигнал на 6 – нужно это для правильного восприятия сервоприводами сообщаемого угла поворота. Угол должен быть в диапазоне от 0 до 180 градусов, а с джойстиков мы получаем сигнал от 0 – 1024. Самостоятельное задание заключается в выведении переменных X и Y на LCD дисплей



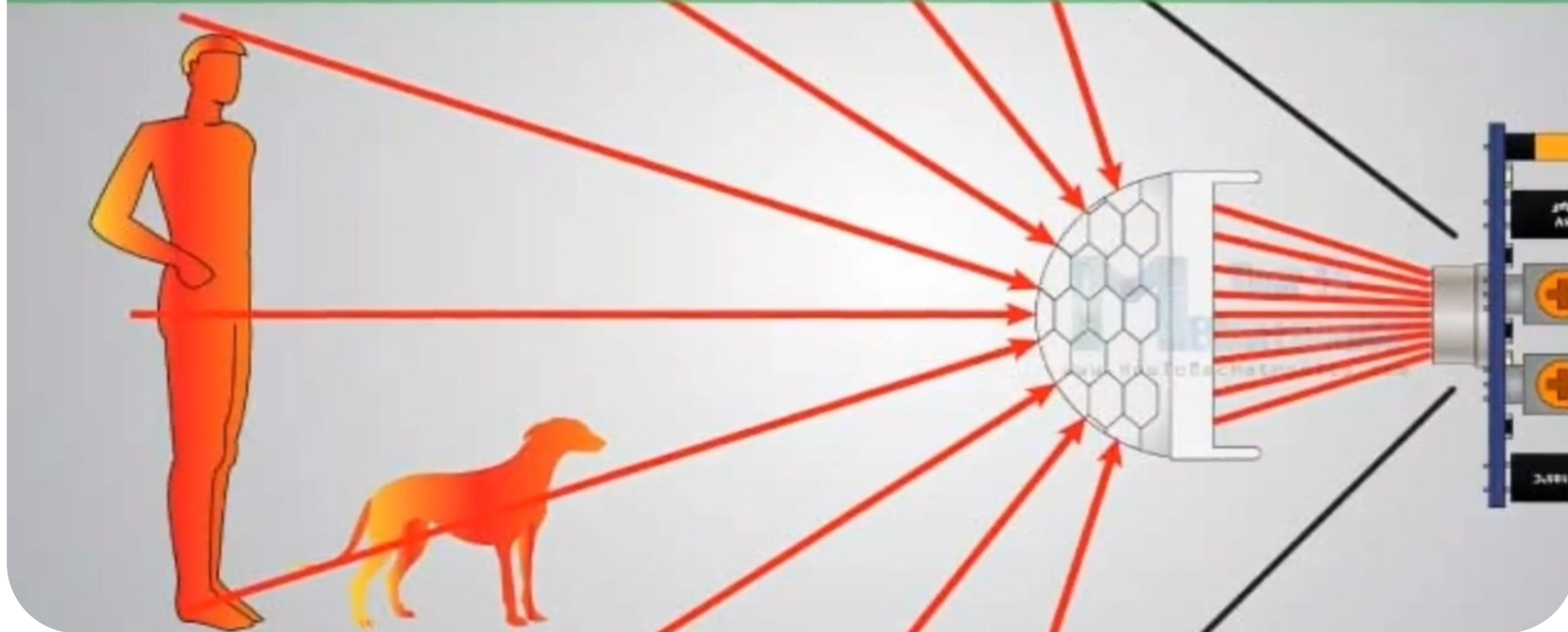
PIR
(датчик движения)
—
Умная сигнализация

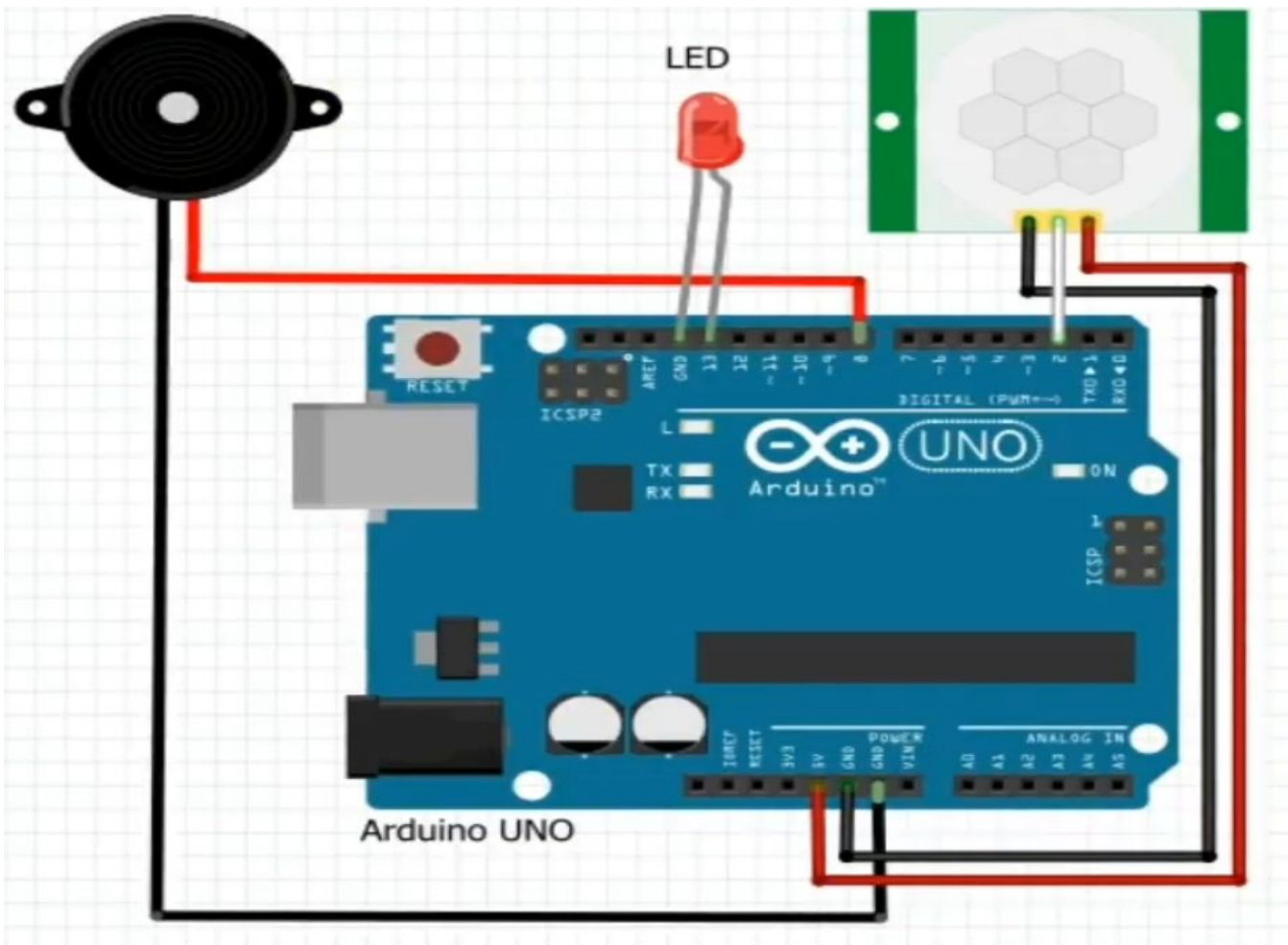


Основным датчиком умного дома, несомненно, является датчик IR(инфракрасный) сигнализации или просто – PIR сенсор. Он воспринимает тепло живых организмов в инфракрасном диапазоне, что позволяет использовать его в качестве детектора движения



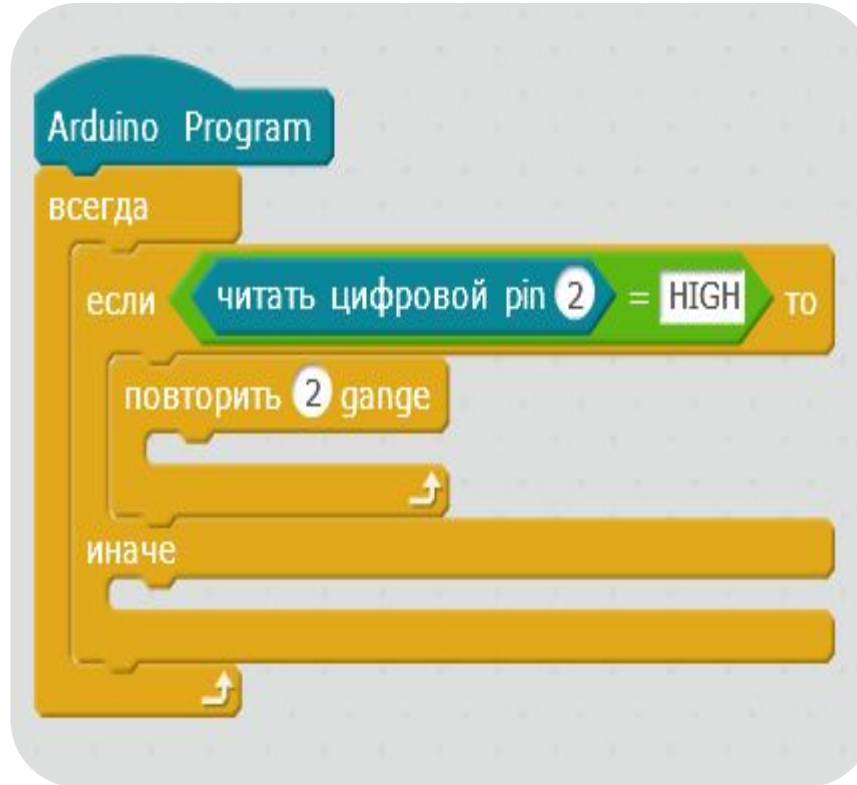
PIR Sensor





Индикацией будет
служить подключенный
светодиод к pin13 и
конечно «мерзкая
пищалка» для звуковых
сигналов сигнализации

PIR(out) – pin2
GND – GND
VCC – 5V
LED – 13
Buzzer – 8



Создадим логическое условие – если с датчика приходит высокий сигнал(HIGH), который будет возникать при любом движении вблизи датчика, то будет запускаяется цикл оповещения – мигать светодиод и пищалка будет издавать несколько коротких звуковых сигналов



Arduino Program

всегда

если **читать цифровой pin 2** = **HIGH** то

повторить **2** gange

иначе

установить цифровой pin **8** udgang как **LOW**

установить цифровой pin **13** udgang как **LOW**

Пищалка на pin8 и
светодиод на
pin13 будут
выключаться при
невыполнении
установленного
нами условия
работы цикла
«Если, Иначе»



Arduino Program

всегда

если **читать цифровой pin 2** = **HIGH** то

повторить 2 gange

установить цифровой pin 8 udgang как HIGH

установить цифровой pin 13 udgang как HIGH

ждать 0.3 секунд

установить цифровой pin 8 udgang как LOW

установить цифровой pin 13 udgang как LOW

ждать 0.3 секунд

иначе

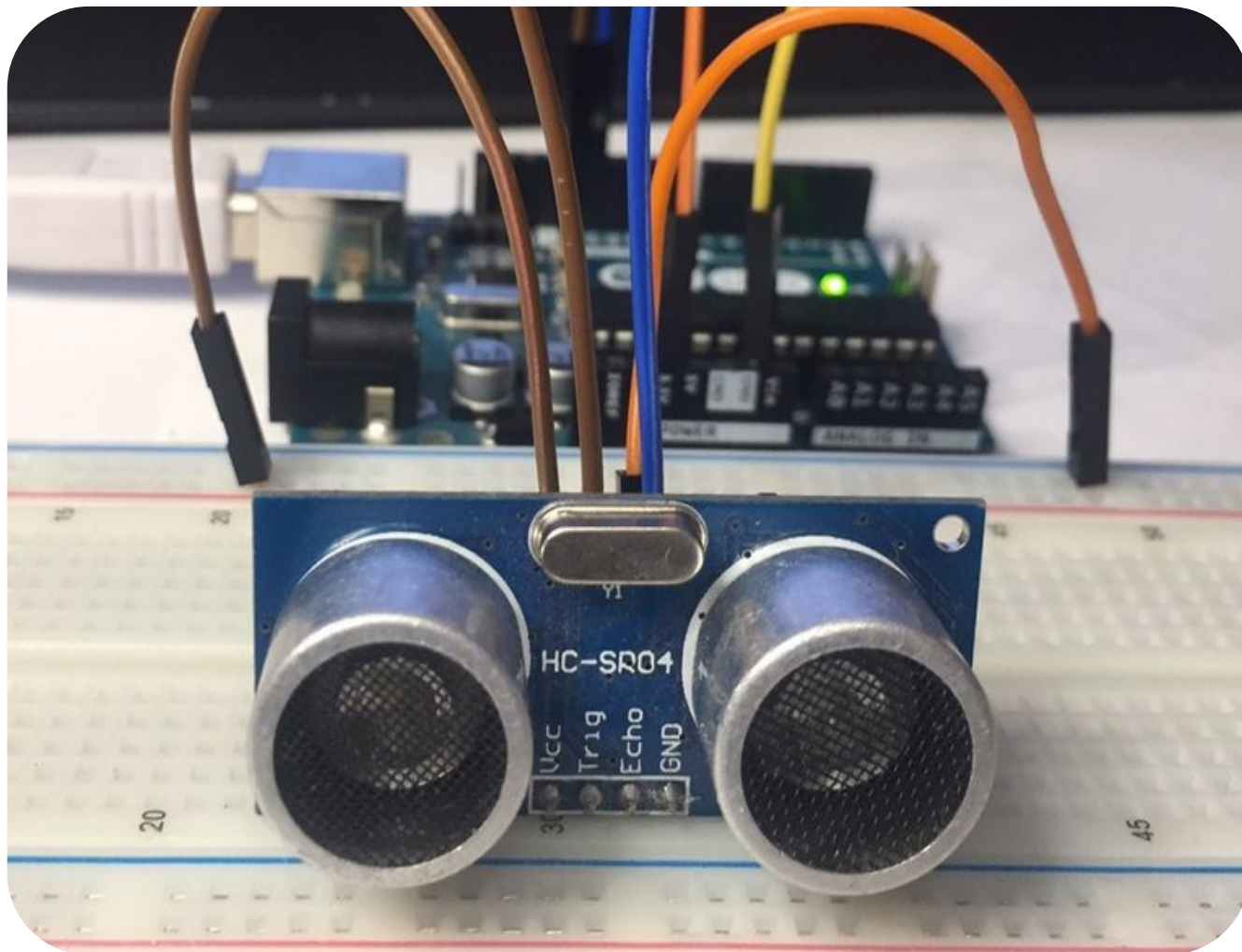
установить цифровой pin 8 udgang как LOW

установить цифровой pin 13 udgang как LOW

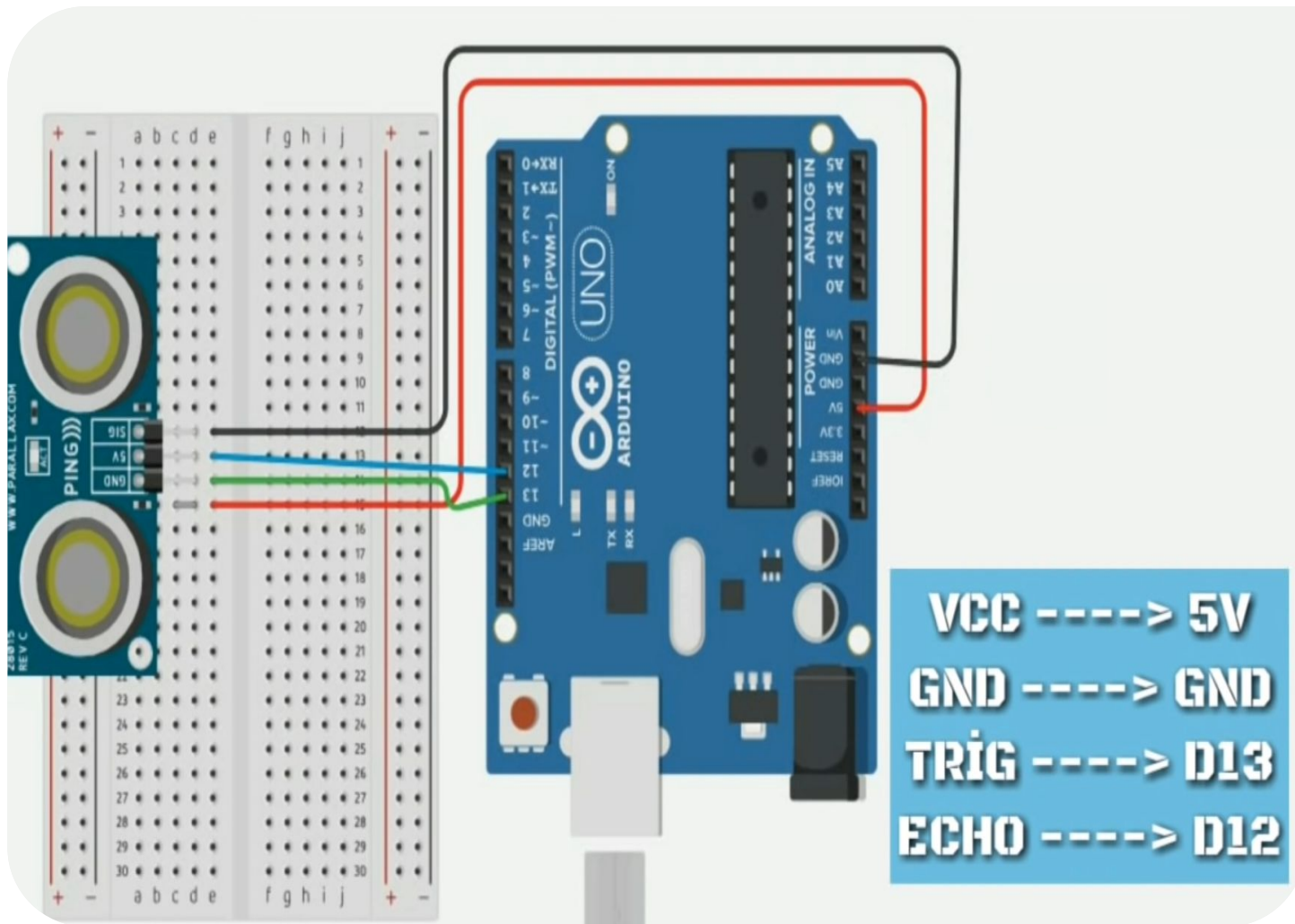
После прошивки Arduino финальной программой стоит постараться не двигаться в области восприятия PIR – сенсора. Требуется это для наблюдения нормальной работы сигнализации



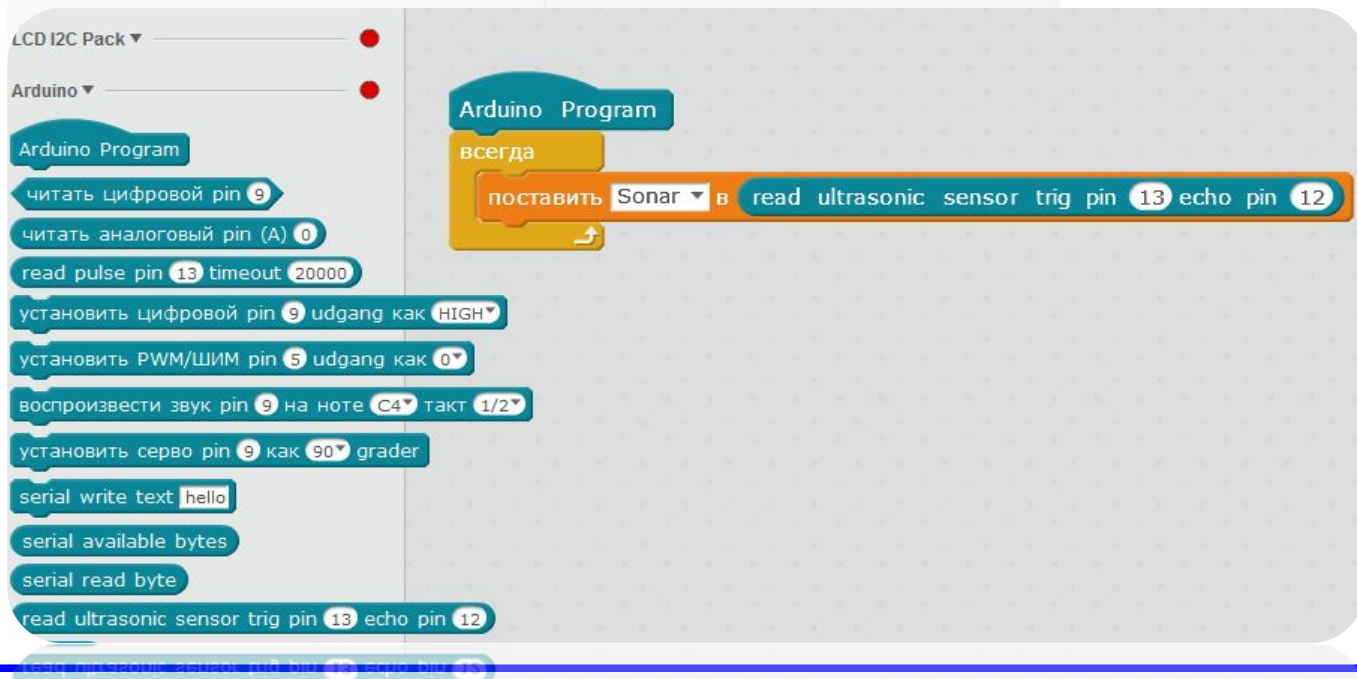
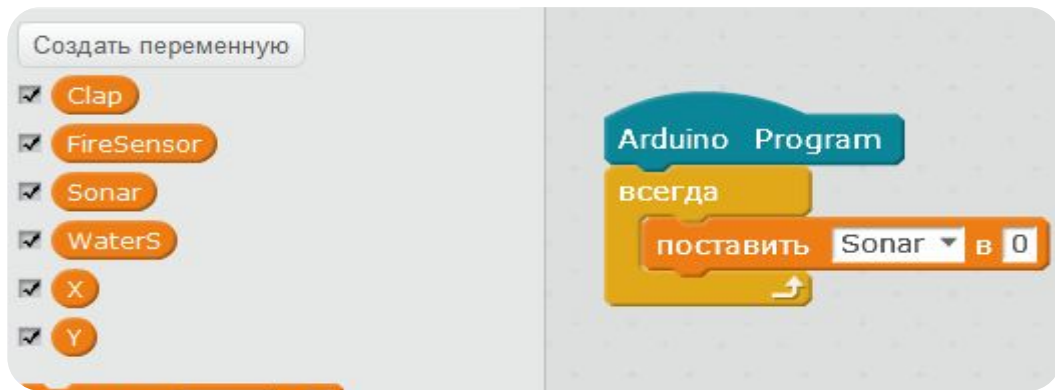
HC-SR04
(датчик расстояния)
—
«Радар» умного дома



Ультра звуковой дальномер является базой огромного количества проектов и технических решений. Например, установив его на сервопривод можно создать полноценный сонар. Датчик работает в диапазоне звуковых волн – испускает и регистрирует отраженную звуковую волну



Собирать схему
можно не только
на макетной
плате, но и
просто подключив
провода к
дальномеру.
Важно не
перепутать
выходы trig и echo



Создадим переменную сонар, в которую будем считывать разницу между отправляемым сигналом с pin13 и получаемым на pin12. Его то мы с Вами сначала попросим озвучить панду, а потом выведем значение уровня сигнала ультразвукового дальномера на LCD дисплей

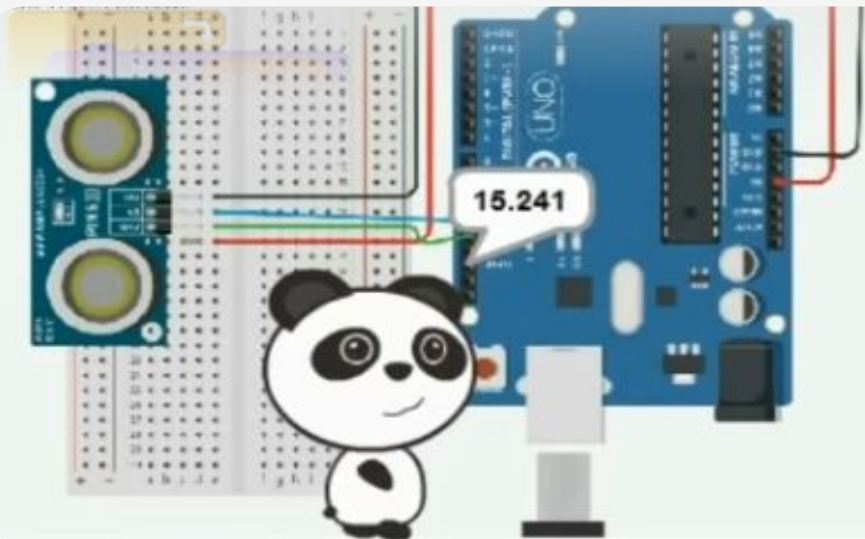


когда щелкнут по

всегда

поставить Sonar в read ultrasonic sensor trig pin 13 echo pin 12

сказать Sonar



Попросим панду mBlock говорить нам показания созданной переменной Sonar. После можно смело переходить к написанию алгоритма вывода значений на LSd дисплей



Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

Clear LCD at 0x27

всегда

поставить Sonar в read ultrasonic sensor trig pin 13 echo pin 12

LCD at 0x27 Line 1 Col 1 Show слить Sonar Sonar

ждать 0.3 секунд



ждать 0.3 секунд

Используем блоки
Из библиотеки для
работы с LCd
дисплеем. Значения
будем выводим в
первую строку после
слова Sonar



Arduino Program

Set LCD at 0x27 has 2 lines and 16 characters per line

Clear LCD at 0x27

всегда

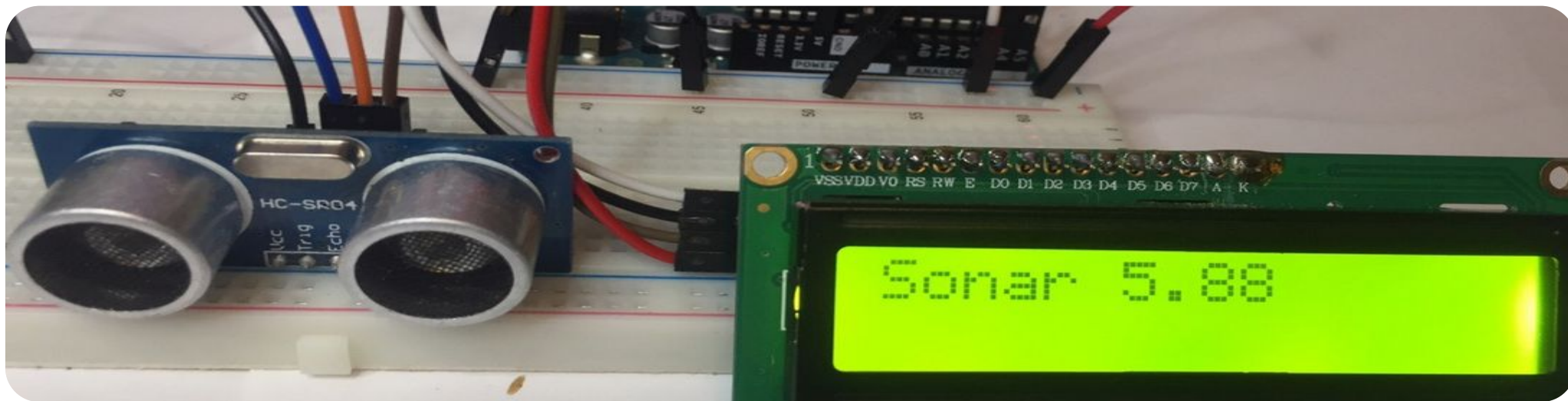
поставить Sonar в read ultrasonic sensor trig pin 13 echo pin 12

LCD at 0x27 Line 1 Col 1 Show слить Sonar Sonar

ждать 0.3 секунд

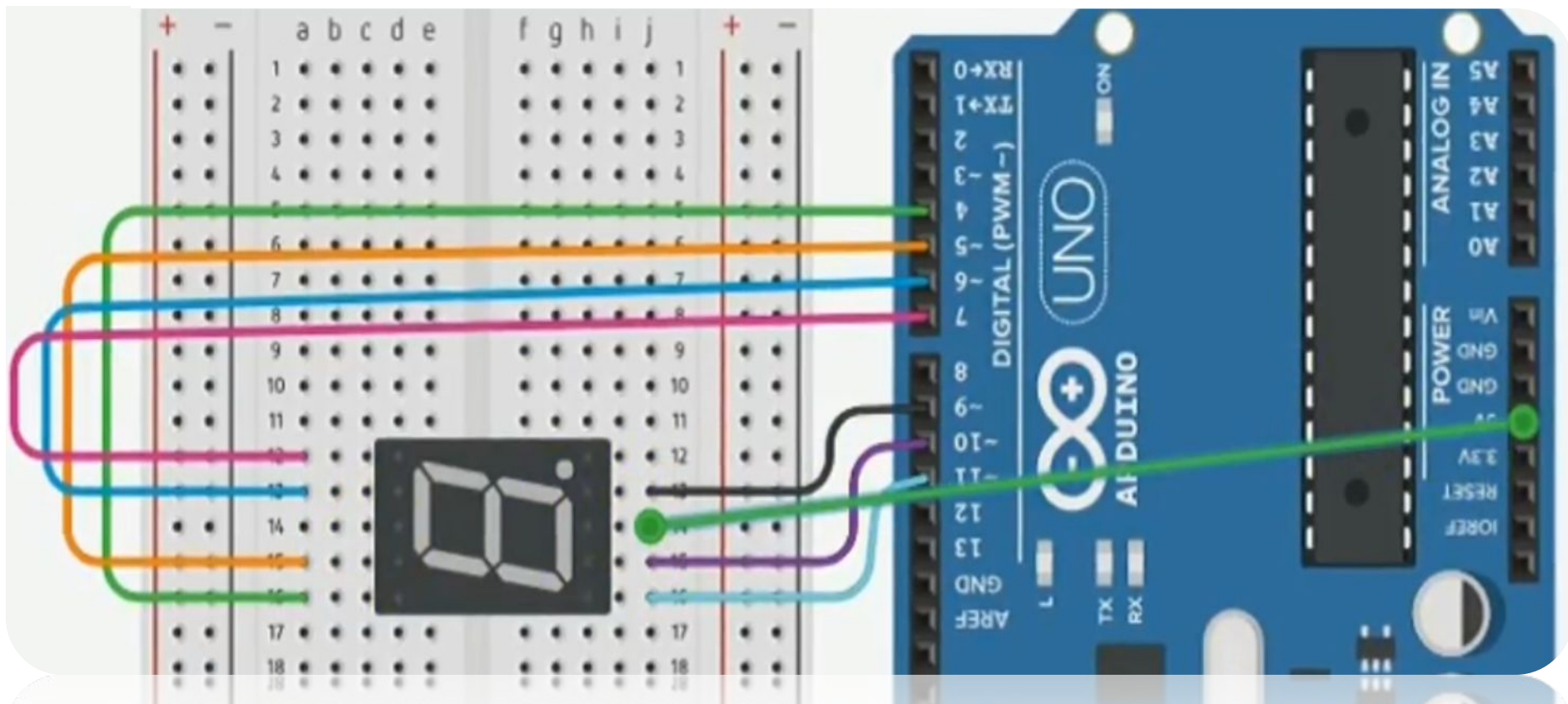


Теперь приближая или отдаляя объекты от HC-SR04 будем получать расстояние до них в CM

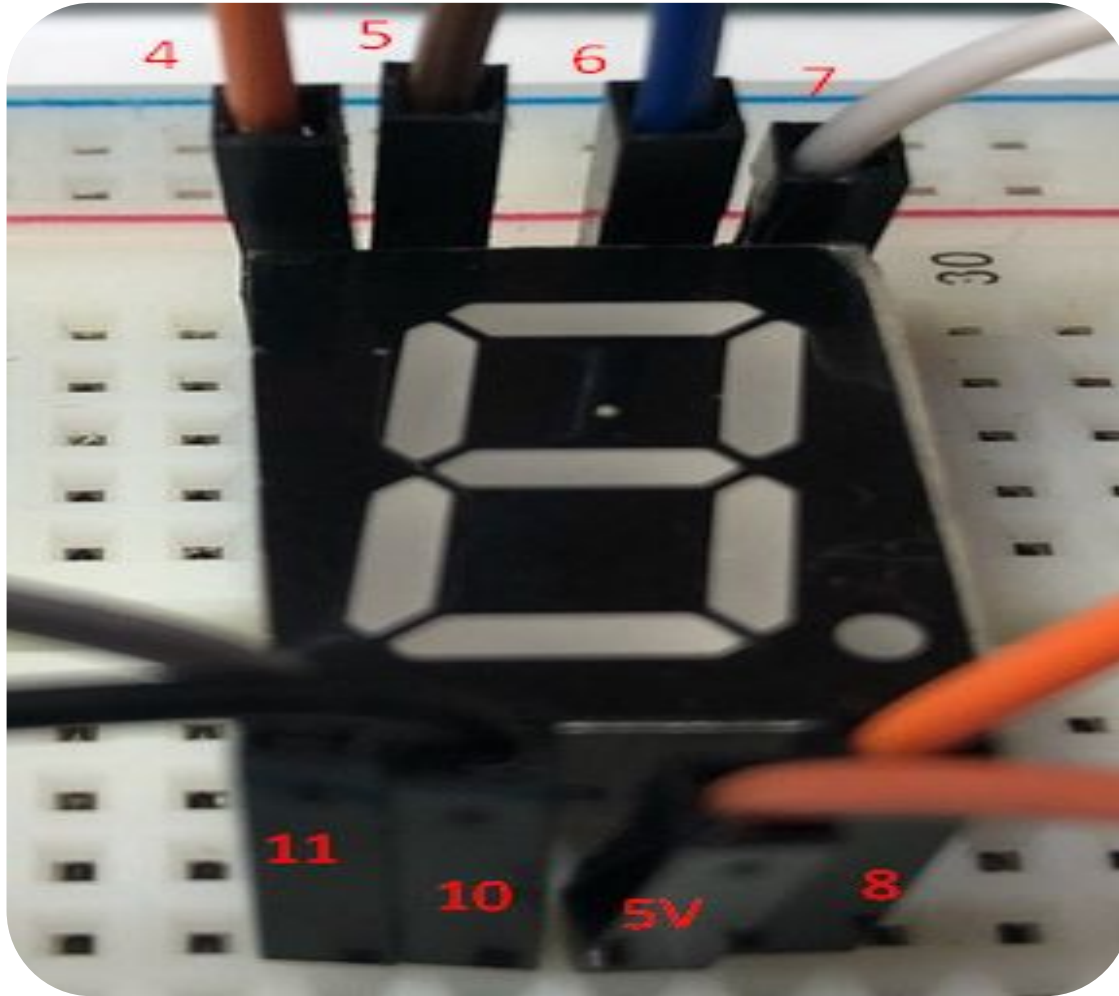




Семисегментный индикатор — циферблат



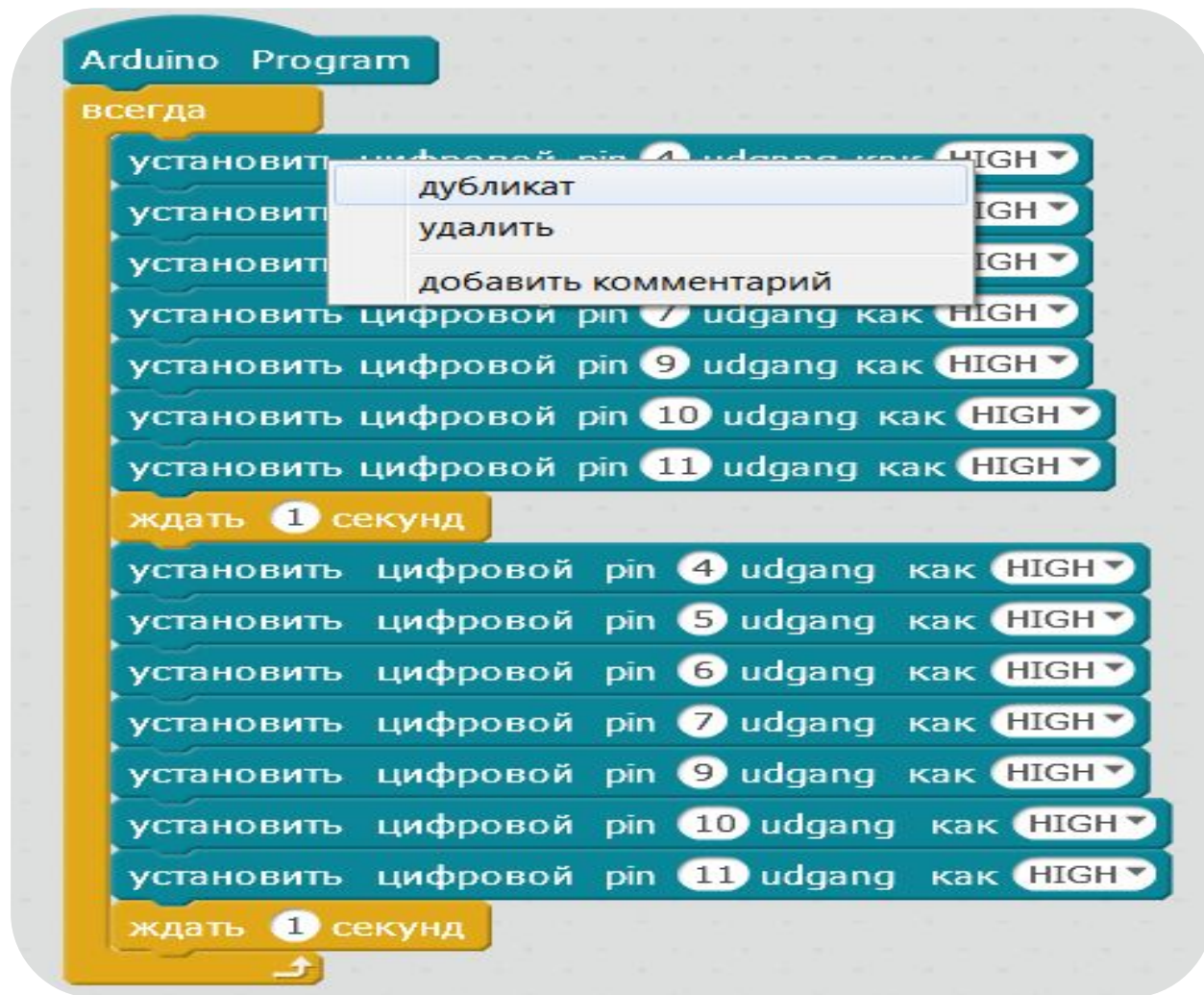
Последний незадействованный в низу пин отвечает за влечение точки – если возникнет необходимость ее подключить можно использовать pin12



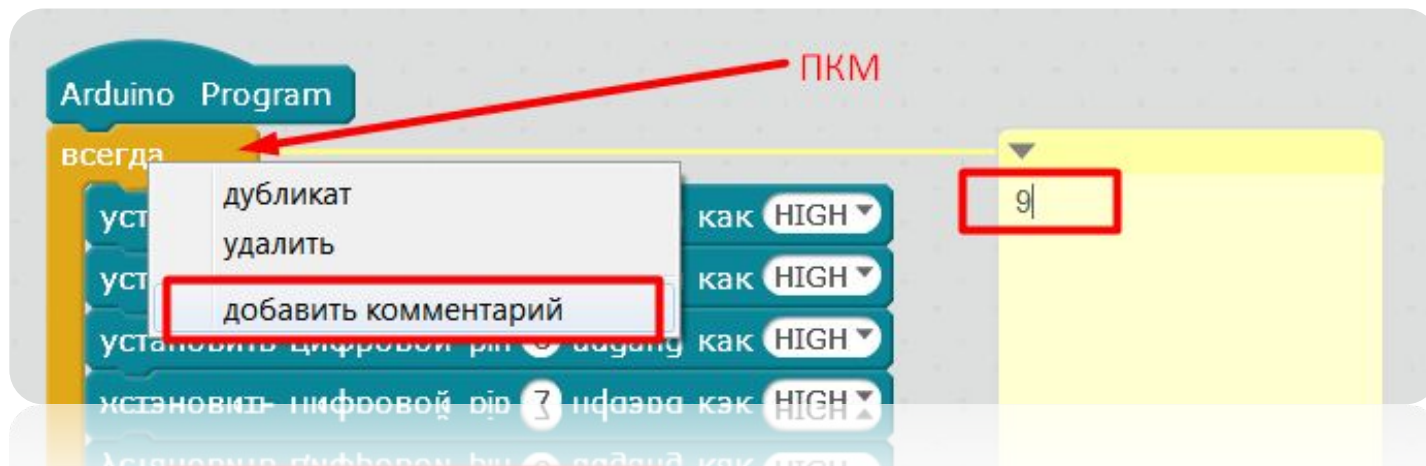
Пожалуй один из самых сложных в подключении и программировании датчик. Его применение очень широкое – от циферблата часов до все различной индикации данных. Мы с вами создадим программу считающую с 9 до 0 с интервалом в одну секунду



Выставляем высокий уровень сигнала на всех используемых семисегментным индикатором портах. После загрузки, это приведет к включению всех сегментов



Используем дублирование, что бы не перетаскивать много блоков. Нажав на блок с которого будем выполнять дублирование ПКМ и выбрав пункт подменю дубликат



Добавим комментарий, что бы понимать как выглядит какая цифра, да и вообще, ориентироваться в нашем с Вами блоковом коде



Arduino Program

всегда

► Цифра 9

установить цифровой pin 4 udgang как HIGH

установить цифровой pin 5 udgang как HIGH

установить цифровой pin 6 udgang как HIGH

установить цифровой pin 7 udgang как HIGH

установить цифровой pin 9 udgang как HIGH

установить цифровой pin 10 udgang как HIGH

установить цифровой pin 11 udgang как LOW

ждать 1 секунд

► Цифра 8

установить цифровой pin 4 udgang как HIGH

установить цифровой pin 5 udgang как HIGH

установить цифровой pin 6 udgang как HIGH

установить цифровой pin 7 udgang как HIGH

установить цифровой pin 9 udgang как HIGH

установить цифровой pin 10 udgang как HIGH

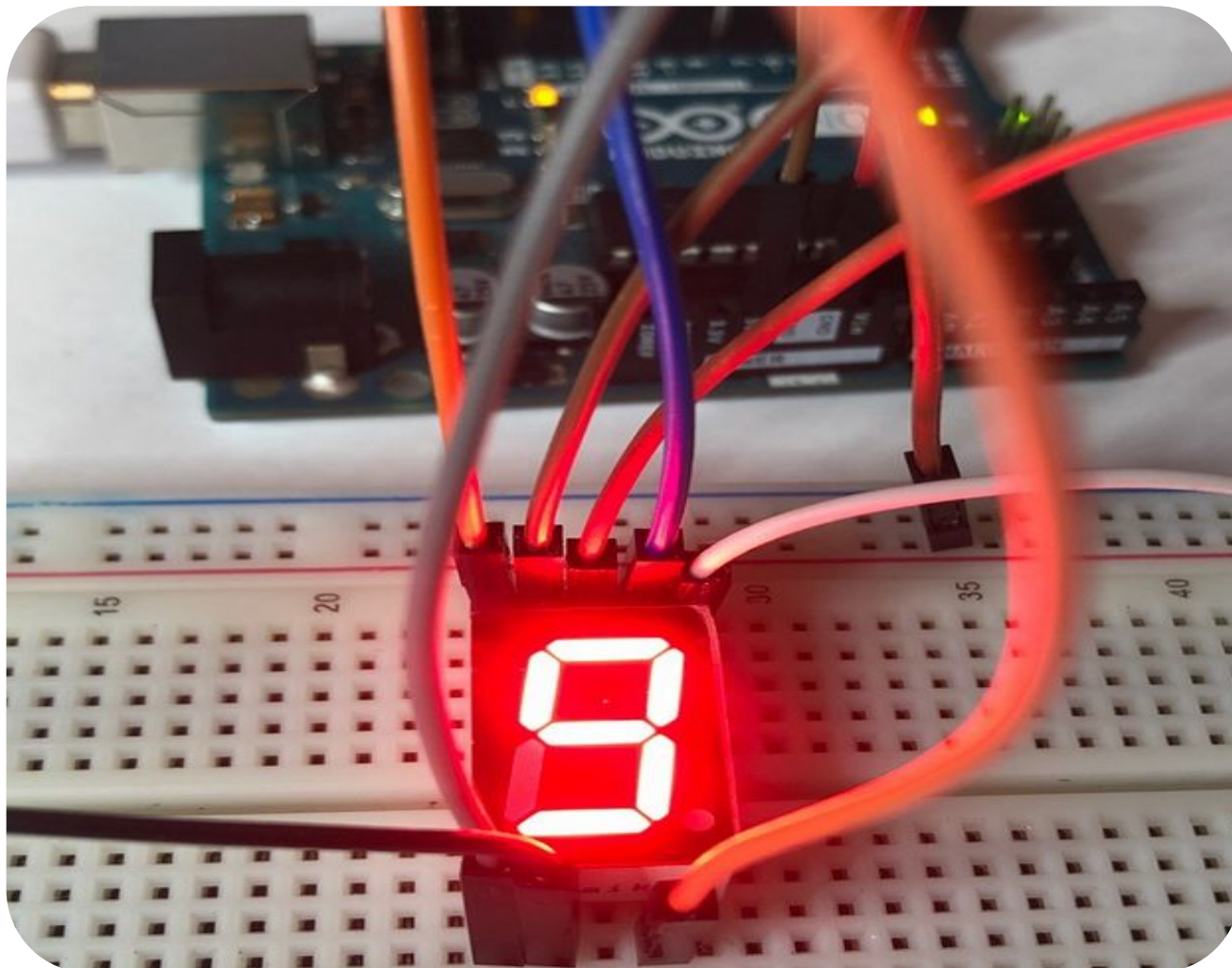
установить цифровой pin 11 udgang как HIGH

ждать 1 секунд



Вместе сделаем
Программу
позволяющую
выводить две самые
простые цифры – 9 и
8. Сделать это можно
отключая и включая
нужные сегменты
индикатора. Вывод
остальных семи цифр
и «0» рекомендуется
выполнить
самостоятельно





Следующий
слайд поможет
вам в создании
щадуманной
нами программы,
которая считает
с 9 до 0 с
интервалом в
одну секунду.

Успехов!

