

Обучающий набор

МЕТРО. Базовый

Вместо предисловия

Дорогие друзья!

Вы приобрели удивительный набор, с помощью которого в игровой форме познакомитесь с программированием различных модулей и научитесь создавать удивительные проекты.



В данном наборе Вы будете работать с модулями линейки «Metro», которые были специально разработаны нашей компанией и являются одними из наиболее дружелюбных и понятных при знакомстве новичков с платформой Arduino.

В комплекте с модулями идёт данная обучающая методичка, изучая и выполняя задания которой, Вы будете постоянно совершенствовать свои навыки программирования.

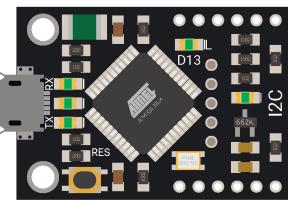


Для того, чтобы Ваше обучение было интересным и разнообразным, мы разделили методичку на 2 части: **в первой** мы познакомим Вас с уже готовыми вариантами использования набора, а **во второй** Вы научитесь работать с модулями в специальной программе, где создаются сценарии их работы.

Добро пожаловать в мир радиоэлектроники!

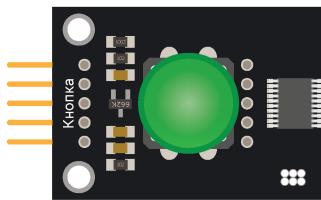
Первое знакомство

После того, как Вы откроете набор, перед Вами окажется комплект из следующих элементов:



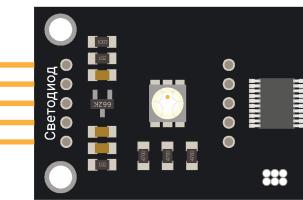
Управляющая плата

Используется для управления всеми модулями в цепи



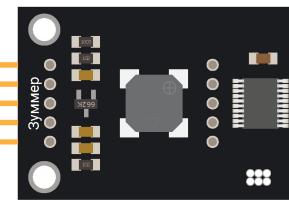
Кнопка

Используется для подачи сигналов, ввода данных, управления устройствами



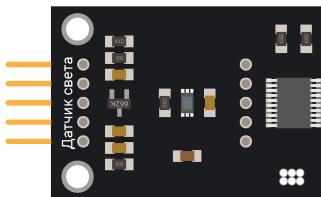
Светодиод

Используется для подачи световых и цветовых сигналов, индикации состояний устройств



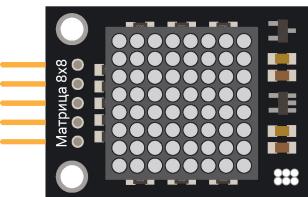
Зуммер

Используется для подачи звуковых сигналов, воспроизведения мелодий



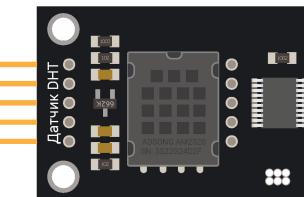
Датчик освещённости

Используется для определения освещённости (в люксах)



Светодиодная Матрица 8x8

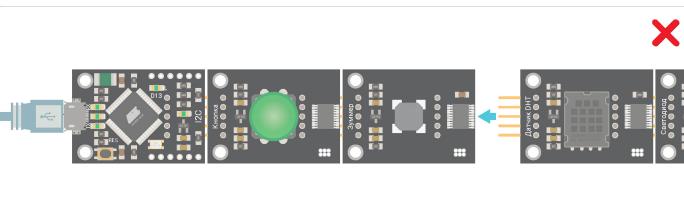
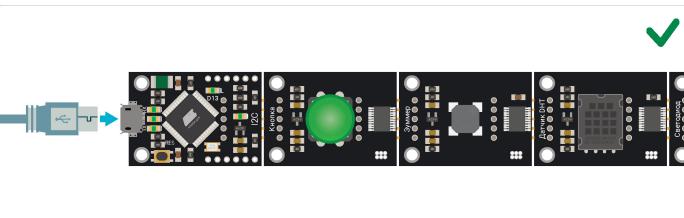
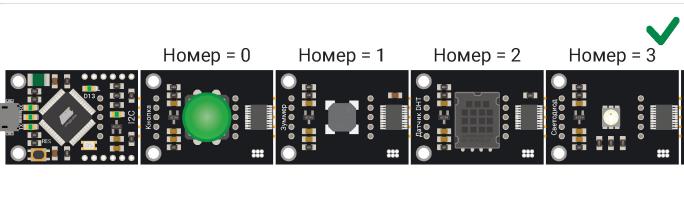
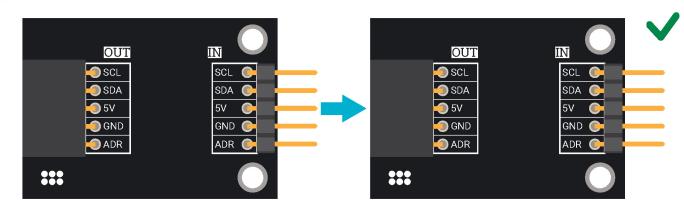
Используется для вывода изображений, символов и бегущей строки



Датчик температуры и влажности

Используется для определения температуры окружающей среды и относительной влажности воздуха

Работа с модулями



Давайте разберёмся, как работать с модулями:

Модули подключаются последовательно.

Для подключения модулей друг к другу достаточно соединить ножки следующего модуля с разъёмом на плате предыдущего модуля и вставить один в другой до упора.

Модули нумеруются от 0 до N.

После того, как модули подключены к управляющей плате, их нумерация в цепи начинается с 0 (номер ближайшего к управляющей плате модуля).

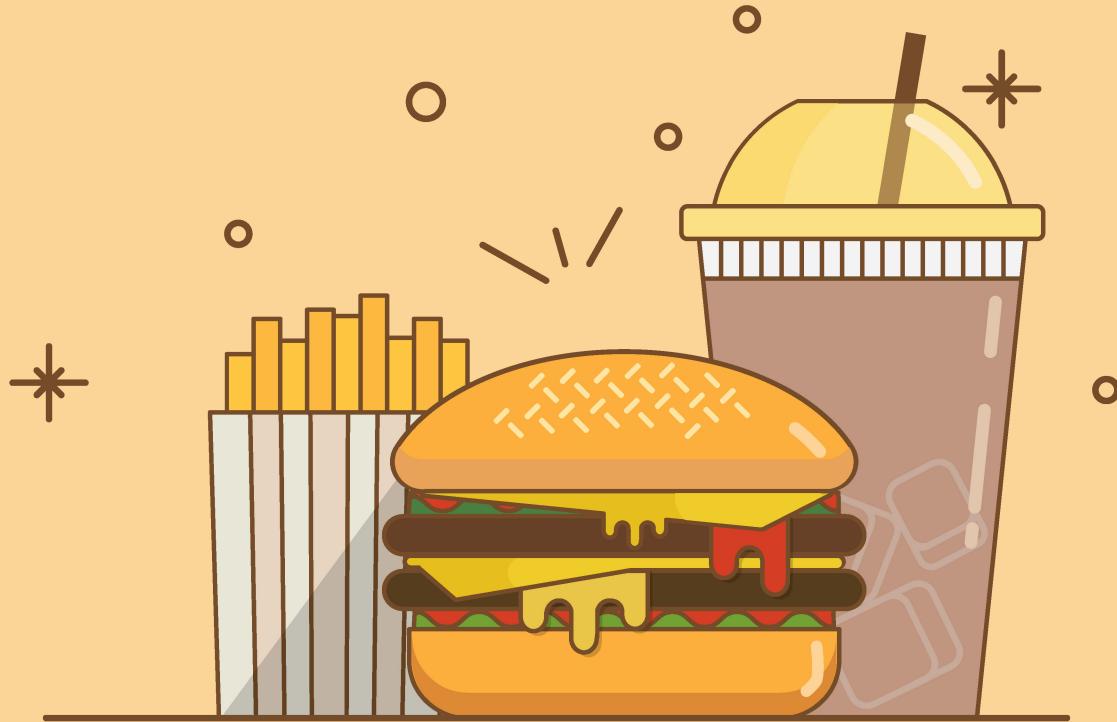
После того, как все необходимые модули соединены, подключите кабель USB.

Для подачи питания достаточно подключить к управляющей плате кабель USB от источника питания. Если Вы подаёте питание с ПК, то при первом запуске Вас попросят установить драйвера. О том, как это сделать, будет рассказано в одной из следующих глав, а до той поры просто игнорируйте это сообщение.

Не подключайте новые модули к цепи при наличии в ней питания!

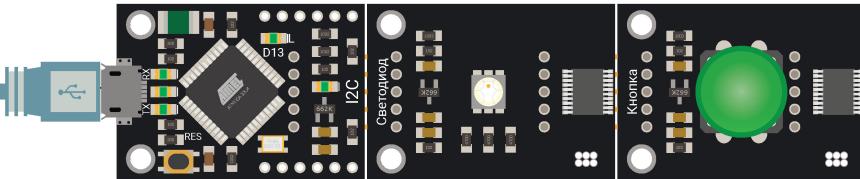
Быстрые схемы

данний блок включает проекты, для создания которых нет необходимости программировать, так как код уже находится в плате управления



Лампа

Схема сборки:



Управляющая плата

Светодиод

Кнопка

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате.

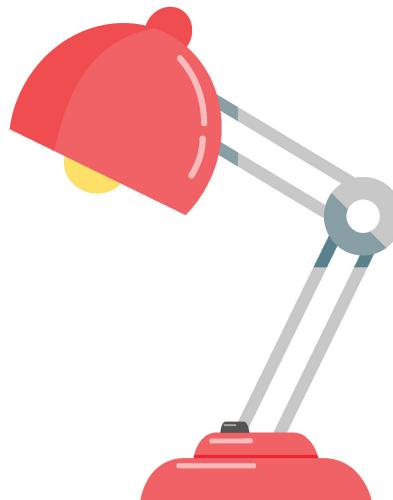
Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

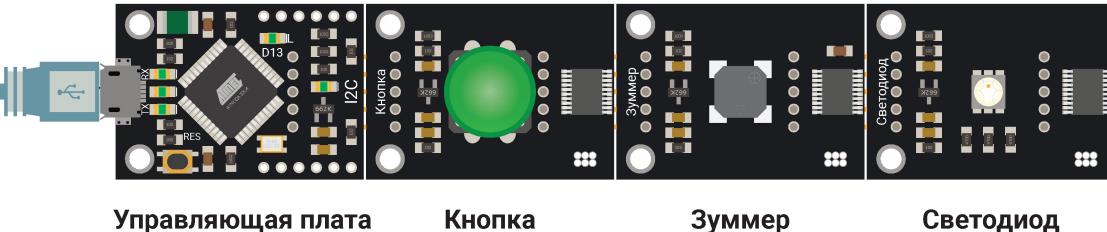
После того, как Вы соберёте данную цепь, нажмите на кнопку, и светодиод будет включен. Повторное нажатие на кнопку выключит его.

Данный принцип применяется во многих электроприборах, одним из которых является настольная лампа.



Светофор

Схема сборки:



Управляющая плата

Кнопка

Зуммер

Светодиод

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

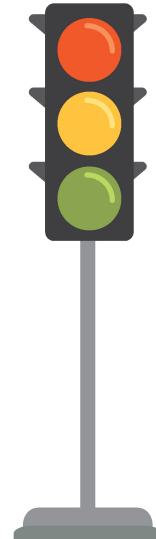
Подключите кабель USB к управляющей плате.

Важно!

Не соединяйте модули при подключенном кабеле USB!

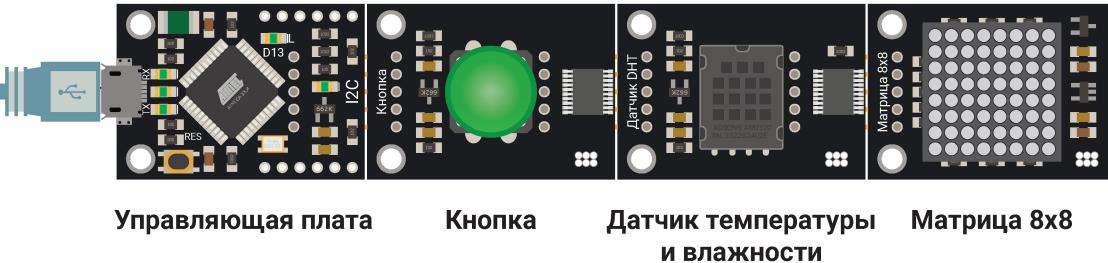
Описание:

После сборки цепи можно переходить к управлению. Нажатие кнопки сменит цвет светодиода на зелёный, и включится зуммер. Подобный алгоритм работы заложен в светофоре (как и в работе настоящего светофора, при включении зелёного цвета кнопка на нажатие не реагирует).



Метеостанция

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате.

Важно!

Не соединяйте модули при подключенном кабеле USB!

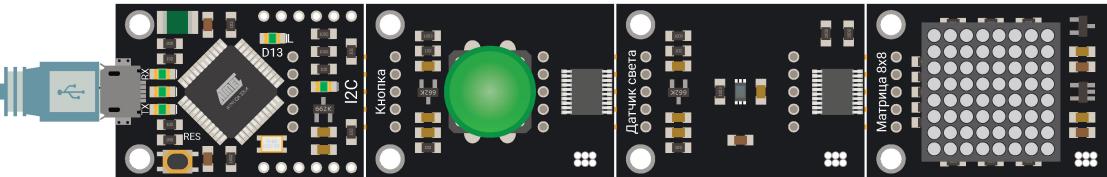
Описание:

После сборки цепи можно строить свою метеостанцию! Датчик температуры и влажности определит все необходимые показатели и выведет их значения на светодиодную матрицу. Нажатие на кнопку позволит Вам переключаться между показаниями.



Люксметр

Схема сборки:



Управляющая плата

Кнопка

Датчик освещённости

Матрица 8x8

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате.

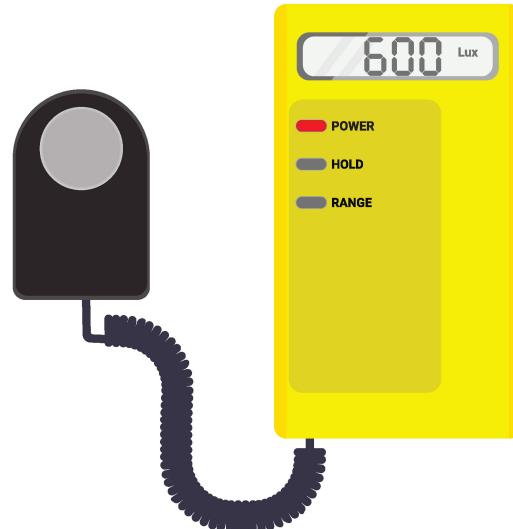
Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

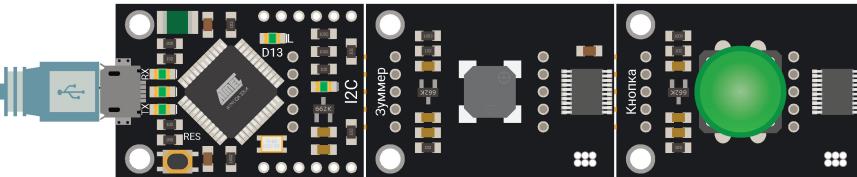
Собрав цепь, у Вас в руках окажется люксметр – прибор, который измеряет освещённость, нажатие на кнопку которого выведет её значение на дисплей.

Благодаря этому прибору Вы сможете определить яркость лампочек в доме и сравнить их с яркостью солнечного света.



Музыкальный автомат

Схема сборки:



Управляющая плата

Зуммер

Кнопка

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

После того, как Вы соберёте данную цепь, на Вашем столе будет далёкий предок нынешних плееров — музыкальный автомат.

Нажимая на кнопку, Вы сможете «пролистать» все записанные в память зуммера мелодии.

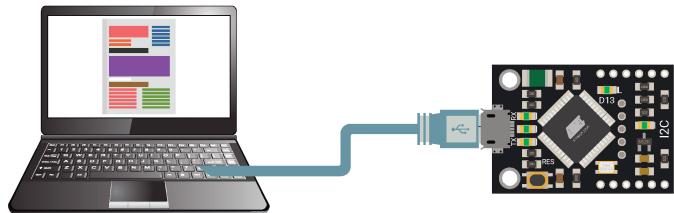


Через тернии к Metro

данний блок включает в себя вводные инструкции и 3 части заданий по программированию, каждая из которых имеет собственный уровень сложности



Инструкция по подключению



Прежде, чем приступить к работе с модулями, Вам необходимо выполнить несколько простых шагов:

Шаг 1

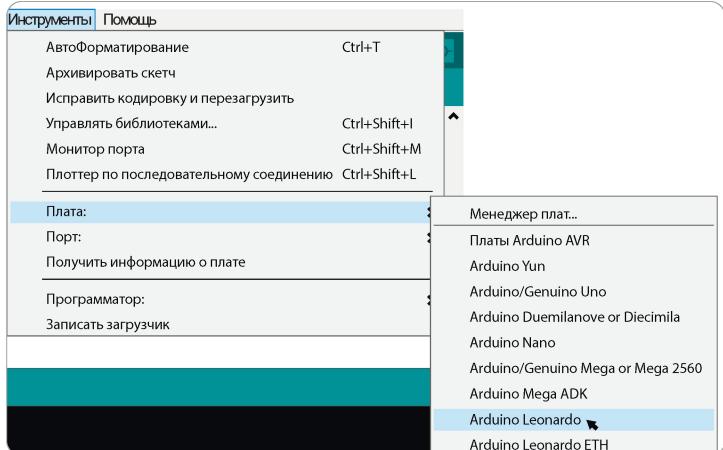
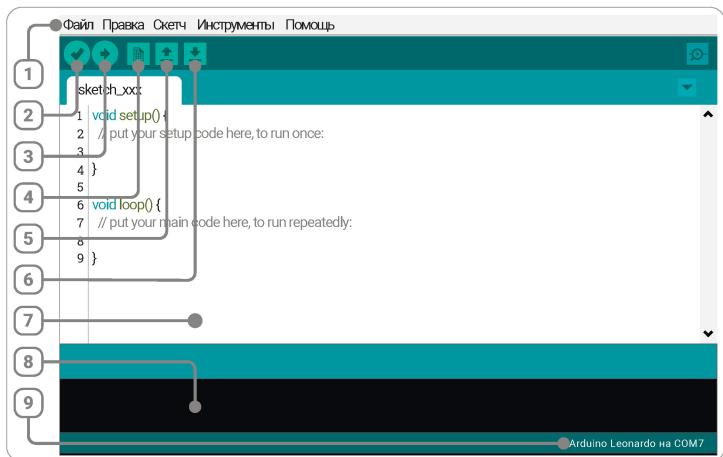
Скачайте программу Arduino IDE с официального сайта: <https://www.arduino.cc/en/Main/Software>

Шаг 2

Установите программу согласно инструкции на нашей странице Wiki: <http://nabor.iarduino.ru> (Данная страница содержит все указанные в методичке скетчи и библиотеки, а так же всю необходимую информацию для работы с данным набором).

Шаг 3

Используя кабель USB, подключите управляющую плату к ПК. При первом подключении на мониторе в области уведомлений, Вы увидите оповещение о подключении нового устройства. Все необходимые драйвера уже установлены вместе с Arduino IDE.



Вы запустили программу, и теперь на экране появилось следующее окно*:

- 1 Пункты меню** – доступ ко всем функциям Arduino IDE.
- 2 Проверить скетч** – проверка кода на ошибки.
- 3 Загрузить скетч в Arduino** – компиляция кода и его загрузка в плату.
- 4 Создать новый скетч** – новое окно с чистой рабочей областью.
- 5 Открыть скетч** – открытие скетча из папки со скетчами.
- 6 Сохранить скетч** – сохранение скетча в указанную папку.
- 7 Рабочая область** – текстовый редактор для написания кода программы (скетча).
- 8 Область уведомлений** – область, информирующая пользователя обо всех событиях при записи или проверке кода.
- 9 Название платы и Номер порта** – модель текущей платы и СОМ-порт, к которому она подключена.

До того, как Вы начнёте писать свой первый скетч, требуется настроить программу для работы с платой.

Первым делом укажите модель управляющей платы:

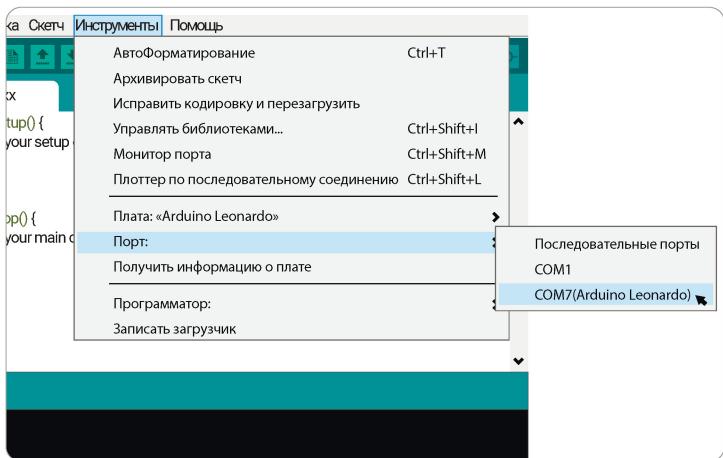
Шаг 4

Войдите в меню «Инструменты» и наведите курсор на строку «Плата».

Шаг 5

При наведении курсора справа отобразится список поддерживаемых плат. Наведите курсор на строку с надписью «Arduino Leonardo» и нажмите левую клавишу мыши.

*В разных версиях программы интерфейс может отличаться. В наборе описывается Arduino IDE версии 1.8.7



Файл Правка Скетч Инструменты Помощь

sketch_xxx

```
1 void setup() {
2 // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8 }
9 }
```

Теперь укажите порт, через который компьютер будет общаться с платой:

Шаг 6

Войдите в меню «Инструменты» и наведите курсор на строку «Порт».

Шаг 7

При наведении курсора на строку справа отобразится список доступных портов. Выберите тот, что появится после подключения платы к ПК, и рядом с которым указан текст **Arduino Leonardo**, наведите на него курсор и нажмите левую клавишу мыши.

Прежде, чем Вы перейдёте к программированию, обратите внимание, что в каждый новый скетч автоматически добавляются 2 блока:

void setup ()

В блоке **void setup()** указываются команды, которые выполняются **только один раз** в момент подачи питания или сразу после перезагрузки управляющей платы.

void loop ()

Команды, которые будут **бесконечно** выполняться, пока на плате есть питание, хранятся в блоке **void loop()**. Микроконтроллер будет выполнять код, дойдёт до последней строки и снова начнёт с первой. («loop» с англ.яз – петля, цикл)

Поздравляем, пришло время написать Ваш первый скетч!

Моргни, если всё верно

Вашей первой программой будет тестовый **скетч №1**, после загрузки которого должен начать мигать светодиод, расположенный на управляющей плате. Данным скетчем Вы проверите, что все шаги были успешно выполнены, и Arduino IDE настроена верно.

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(LED_BUILTIN, HIGH);  
7   delay(1000);  
8   digitalWrite(LED_BUILTIN, LOW);  
9   delay(1000);  
10 }
```

Скетч 1



При вводе кода соблюдайте **регистр** (ПРОПИСНЫЕ и строчные символы воспринимаются программой по-разному!). Каждый новый скетч необходимо будет **Сохранить** перед тем, как **Загрузить** его в плату.

скетч 1

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(LED_BUILTIN, HIGH);  
7   delay(1000);  
8   digitalWrite(LED_BUILTIN, LOW);  
9   delay(1000);  
10 }
```



Загрузка завершена.

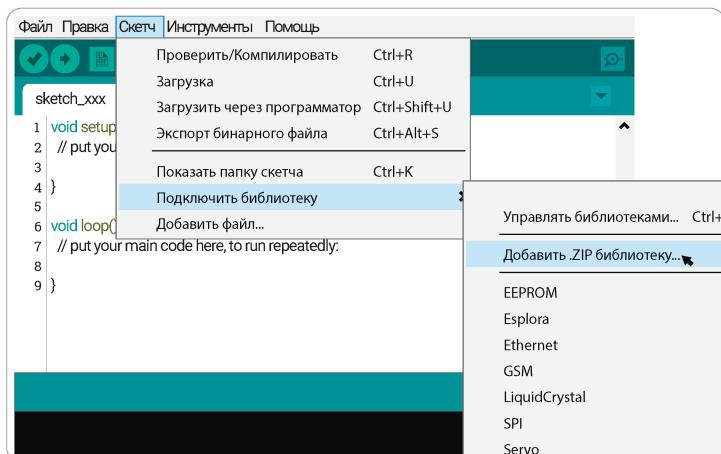
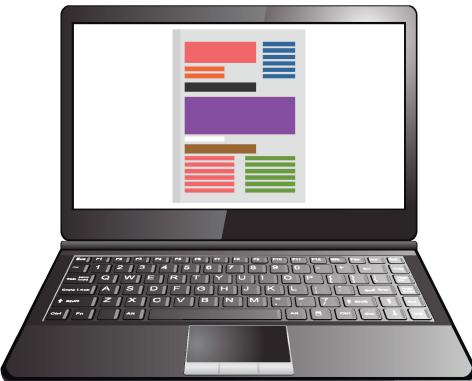
avrdude done. Thank you.

Если после нажатия кнопки «Загрузить» Вы видите надпись «**Загрузка завершена**», это значит, что все предварительные шаги были выполнены верно, и скетч был загружен в управляющую плату.

Если же после нажатия кнопки «Загрузить» Вы наблюдаете любую другую надпись, а цвет фона изменился на красный, значит, настройки платы и порта были выполнены неверно или скетч написан с ошибкой. Вернитесь к главе «**Инструкция по подключению**» на стр.12 и выполните все шаги ещё раз.

После того, как Вы загрузите скетч и убедитесь, что управляющая плата работает правильно, можно смело переходить к следующей главе, в которой Вы узнаете, как сделать работу с модулями простой и понятной.

Установка библиотеки



Чтобы Arduino IDE смогла работать с модулями Metro, необходимо установить библиотеку «**iarduino_Metro**».

Библиотека – это некоторая часть кода, подключаемая из внешнего файла и облегчающая работу с модулями или датчиками. Имеет расширение «**имя_файла.h**».

Среда программирования может существенно расширять свои возможности за счет использования библиотек. Arduino IDE имеет ряд встроенных библиотек, но Вы также можете скачивать или создавать свои собственные.

Шаг 1

Скачайте библиотеку «**iarduino_Metro**» с нашей страницы Wiki: <http://nabor.iarduino.ru>

Шаг 2

Запомните путь к файлу или скопируйте его на Рабочий стол вашего компьютера.

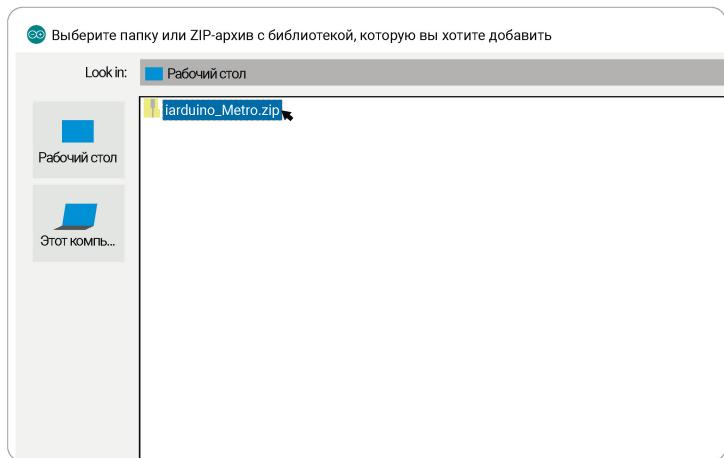
После того, как Вы скачали библиотеку, её необходимо установить:

Шаг 3

Войдите в меню «**Скетч**» и наведите курсор на строку «**Подключить библиотеку**».

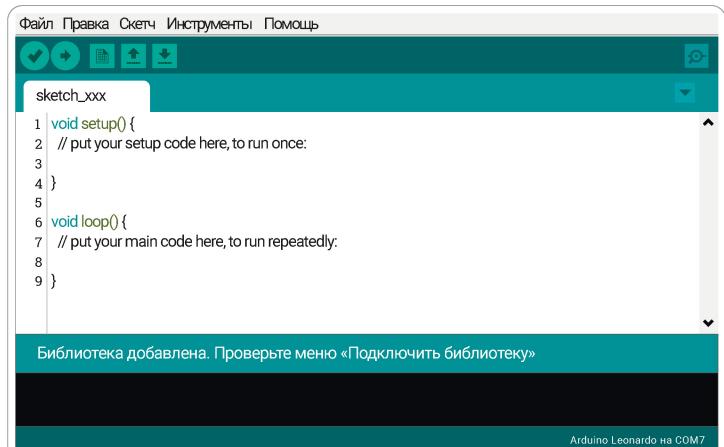
Шаг 4

При наведении курсора на строку, справа отобразится список доступных библиотек. Для того, чтобы добавить новую, наведите курсор на строку с надписью «**Добавить .ZIP библиотеку**» и нажмите левую клавишу мыши.



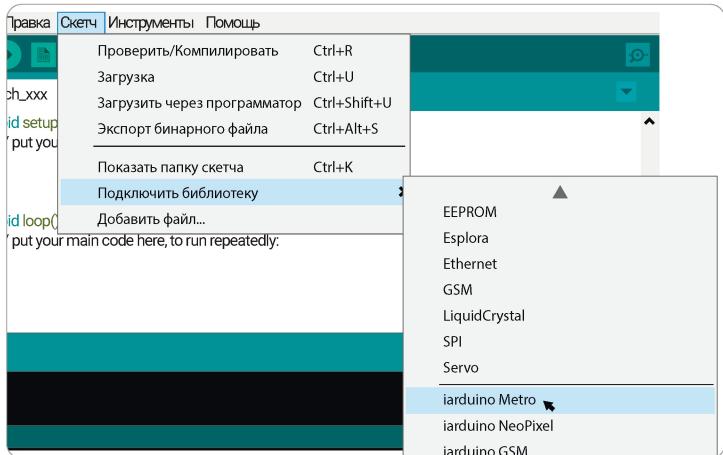
Шаг 5

Укажите путь к **ранее скачанному архиву** с библиотекой и дважды нажмите левую клавишу мыши, наведя курсор на файл.



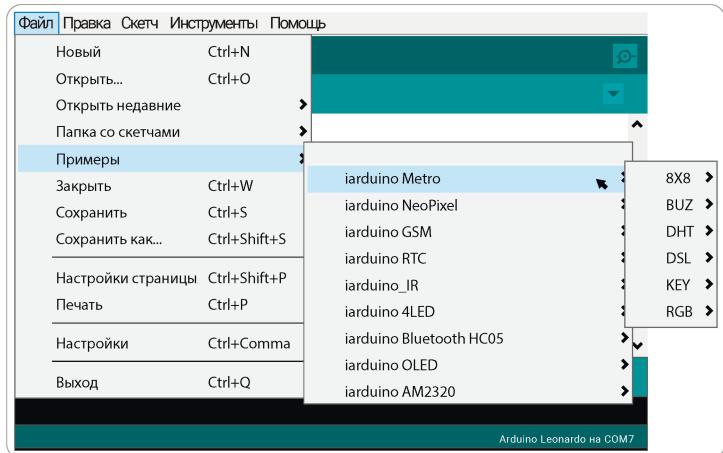
Шаг 6

После того, как архив будет загружен, в области уведомлений Вы увидите сообщение, что библиотека добавлена (Библиотека добавляется только один раз, а дальше Arduino IDE запомнит её, и Вам надо будет только **подключать** её. О том, как это сделать, Вы узнаете в ближайшем скетче!).



Шаг 7

Установленная библиотека находится в меню «Скетч», строка «Подключить библиотеку». Нажмите клавишу мыши, наведя курсор на строку с именем нужной библиотеки, и она будет добавлена в скетч.



Шаг 8

Помимо этого, в меню «Файл», строка «Примеры», можно увидеть примеры работ с библиотеками, уже установленными в Arduino IDE.

Вот, собственно, и всё, что Вам необходимо было сделать, чтобы начать создавать свои собственные проекты!

Теперь быстрее переходите к следующей странице, ведь там Вас ждёт первый блок заданий!

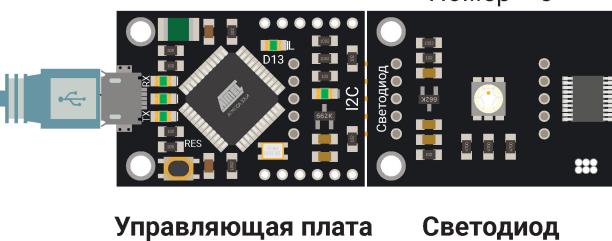
Стажёр

данний блок познакомит Вас с основными возможностями модулей, используя которые Вы создадите свои первые самостоятельные проекты



Автоматическое освещение

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №2**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном проекте Вам предстоит научиться управлять работой светодиода.

После загрузки скетча светодиод будет зажигаться и гаснуть через определённые промежутки времени.



Дополнительное задание:

В данном блоке будут содержаться вопросы и задания, которые будут способствовать как закреплению полученных знаний, так и расширению уже имеющихся.

Советуем переходить к данной части только после ознакомления со скетчем из урока и комментариями к нему.

Задание 1

Что необходимо изменить, чтобы светодиод горел дольше? Меньше? Почему?

Задание 2

Что будет, если в функции **Metro[0].on()** указать другую цифру в индексе?

```

1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup() {
5   iarduino_Metro_Start();
6 }
7
8 void loop() {
9   Metro[0].on(); delay(3000);
10  Metro[0].off(); delay(3000);
11 }

```

скетч 2

```

1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup() {
5   iarduino_Metro_Start();
6 }
7
8 void loop() {
9   Metro[0].on(); delay(3000);
10  Metro[0].off(); delay(3000);
11 }

```

Загрузка завершена.

avrdu done. Thank you.

скетч 2

```

1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup() {
5   iarduino_Metro_Start();
6 }
7
8 void loop() {
9   Metro[0].on(); delay(3000);
10  Metro[0].off(); delay(3000);
11 }

```

iarduino_Metro.h: No such file or directory

exit status 1

Скетч 2

#include <> – команда, подключающая в скетч коды из различных файлов. В данном случае она подключает библиотеки. Обратите внимание, что в конце строки точка с запятой не ставится.

iarduino_Metro_Start() – функция обнаруживает все модули Metro и назначает им адреса.

Metro[индекс*].on(); – функция включает светодиод.

Metro[индекс*].off(); – функция выключает светодиод.

delay() – останавливает выполнение программы на заданное количество миллисекунд(в 1 секунде 1000 миллисекунд).

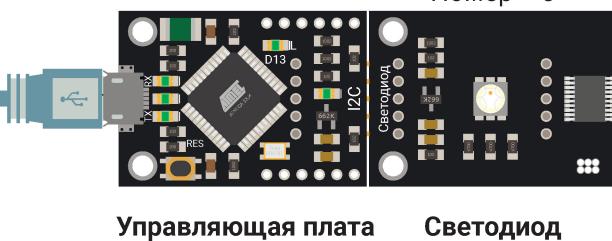
***индекс** – порядковый номер элемента. Обратите внимание, что нумерация элементов начинается с 0, при этом управляющая плата в нумерации не учитывается.

Если после нажатия кнопки «Загрузить» Вы видите надпись «Загрузка завершена», это значит, что все предварительные шаги были выполнены верно, и скетч был загружен в управляющую плату.

Если после нажатия кнопки «Проверить» или «Загрузить» Вы наблюдаете данную надпись, это значит, что библиотека не установлена или установлена неверно. Вернитесь к главе «Установка библиотеки» на стр.16 и выполните все шаги ещё раз.

Гирлянда

Схема сборки:



Управляющая плата

Номер = 0

Светодиод

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

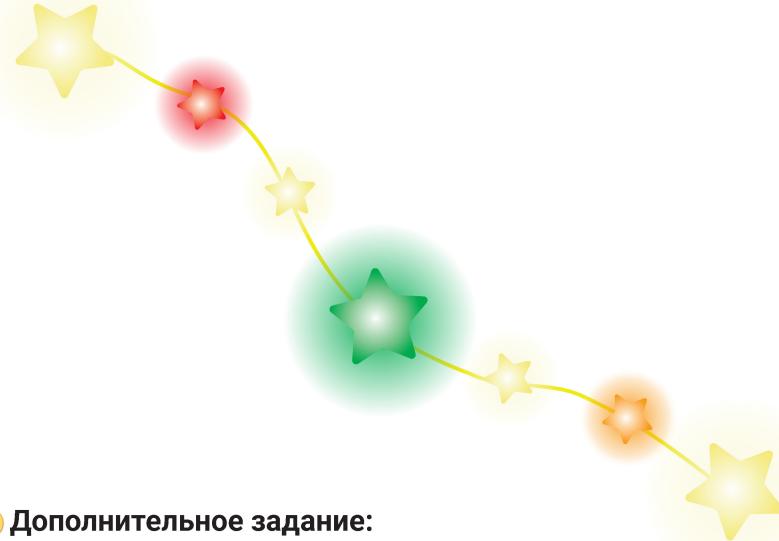
Подключите кабель USB к управляющей плате и загрузите **скетч №3**, а, разобравшись с алгоритмом его работы, переходите к загрузке скетча **№4**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

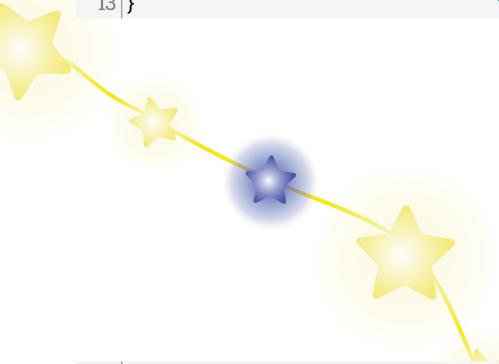
В данном проекте Вам предстоит научиться управлять цветом светодиода(**скетч 3**), и задавать режим его работы по времени(**скетч 4**).



Дополнительное задание:

Задание 1

Напишите скетч так, чтобы светодиод вывел все цвета радуги по порядку.



```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5     iarduino_Metro_Start();
6 }
7
8 void loop(){
9     Metro[0].on(0xffff00); delay(3000);
10    Metro[0].on(0x00ff00); delay(3000);
11    Metro[0].on(0x0000ff); delay(3000);
12    Metro[0].off();         delay(1000);
13 }
```

Скетч 3

Metro[0].on(ЦВЕТ*); – функция включает светодиод с заданным цветом.

Цвет задаётся кодом. Код можно условно разделить на 3 части по 2 символа: **0xRRGGBB**, где

2 символа RR – насыщенность красного цвета;

2 символа GG – насыщенность зелёного цвета;

2 символа BB – насыщенность синего цвета;

Для задания насыщенности необходимо в каждой части кода указать число от 0 до 255, но в шестнадцатеричной системе счисления (как переводить числа из одной системы в другую см. <http://nabor.iarduino.ru>).

Ниже представлены коды основных цветов:

0xff0000 – максимальная яркость красного цвета;

0x00ff00 – максимальная яркость зелёного цвета;

0x0000ff – максимальная яркость синего цвета;

0xff00ff – максимальная яркость фиолетового цвета;

0xffff00 – максимальная яркость жёлтого цвета;

0x00ffff – максимальная яркость голубого цвета;

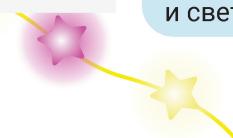
0xffffffff – максимальная яркость белого цвета;

*Полный список функций модулей, в том числе Светодиода, Вы всегда можете найти в нашей *Шпаргалке*.

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5     iarduino_Metro_Start();
6 }
7
8 void loop(){
9     Metro[0].on(0xffffffff, 2000); delay(3000);
10 }
```

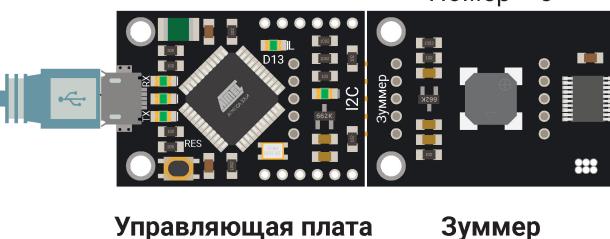
Скетч 4

Metro[0].on(ЦВЕТ, ВРЕМЯ); – функция включает светодиод с заданным цветом на указанное время в мс. В данном скетче сценарий таков: светодиод загорается на 2 секунды, а функция **delay** вместе со светодиодом выполняется 3 секунды. Светодиод через 2 секунды потухнет, функция задержки выполнится ещё 1 секунду, и светодиод снова загорится на 2 секунды.



Полицейская сирена

Схема сборки:



Управляющая плата

Номер = 0

Зуммер

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №5**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном проекте Вам предстоит научиться управлять работой зуммера, включая и выключая звук с определённой частотой и длительностью.

Дополнительное задание:

Задание 1

Измените звучание зумера так, чтобы звуки были более продолжительными; менее продолжительными;



Скетч 5

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5     iarduino_Metro_Start();
6 }
7
8 void loop(){
9     Metro[0].on(1000); delay(300);
10    Metro[0].on(4000); delay(300);
11 }
```

Metro[0].on(ЧАСТОТА, ДЛИТЕЛЬНОСТЬ); – функция включает звук указанной частоты и длительности.

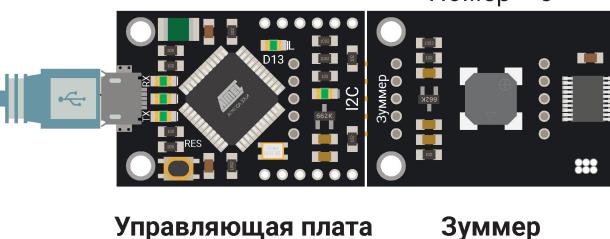
ЧАСТОТА - Целочисленное значение, определяющее высоту звука в Гц, от 10 до 10000. Если частоту не указать, то будет воспроизведен тональный сигнал с частотой 2,7 кГц;

ДЛИТЕЛЬНОСТЬ - Целочисленное значение, определяющее время воспроизведения тонального сигнала в мс от 0 до 25500. Если длительность не указана, то сигнал будет воспроизводиться непрерывно;



Музыкальный автомат

Схема сборки:



Управляющая плата

Номер = 0

Зуммер

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №6**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном проекте Вы познакомитесь со встроенными в зуммер мелодиями. Данный проект, но в более расширенной версии, встречался Вам в блоке, где код уже был загружен в управляющую плату.

26

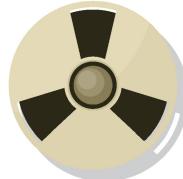
Дополнительное задание:

Задание 1

Определите, сколько мелодий записано в зуммер.

Задание 2

Напишите скетч так, чтобы зуммер воспроизвел несколько мелодий по порядку.



Скетч 6

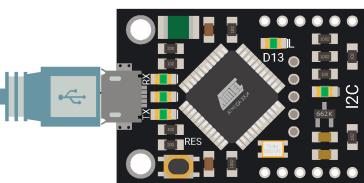
```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5     iarduino_Metro_Start();
6     Metro[0].on(MELODY(HOME));
7 }
8
9 void loop() {}
```

Metro[0].on(MELODY(HOME)) – воспроизведение мелодии, соответствующей указанному НОМЕРУ.



Бегущая строка

Схема сборки:



Управляющая плата Матрица 8x8

Номер = 0

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №7**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

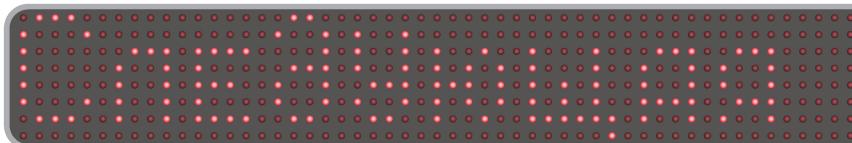
В данном проекте Вы познакомитесь со светодиодной матрицей 8x8.

После загрузки скетча на дисплей будет выведен текст, указанный в **двойных кавычках**.

Дополнительное задание:

Задание 1

Напишите скетч так, чтобы на дисплей выводилось несколько различных надписей.



```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5     iarduino_Metro_Start();
6     Metro[0].set("Hello world", 250, 1000);
7 }
8
9 void loop()
```

Скетч 7

Metro[0].set(«ТЕКСТ»,СКОРОСТЬ,ПАУЗА) – функция выводит текст в режиме бегущей строки.

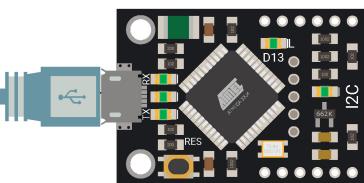
“ТЕКСТ” - содержимое бегущей строки, до 512 символов;

СКОРОСТЬ - скорость движения бегущей строки. Задаётся от 0 до 255. Если указан 0, то строка не движется.

ПАУЗА - время до повторного запуска бегущей строки. Указывается в миллисекундах от 0 до 25500. Если параметр отсутствует или равен 0, то строка будет выведена всего 1 раз и больше выводиться не будет.

Камера! Мотор! Снято!

Схема сборки:



Управляющая плата Матрица 8x8

Номер = 0

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №8**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

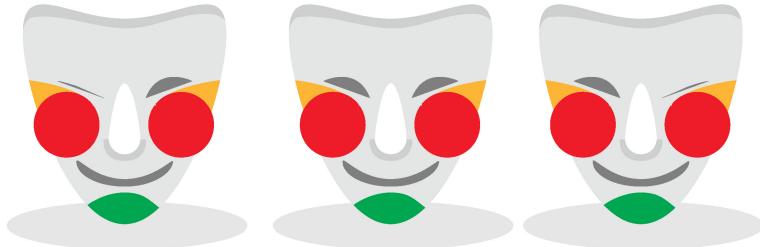
Описание:

В данном проекте Вы научитесь создавать на дисплее матрицы анимацию. Для этого Вам надо будет создать «кадры» и задать правильный порядок их отображения.

Дополнительное задание:

Задание 1

Выведите на дисплей своё изображение, создав анимацию.



Скетч 8

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 uint8_t image1[8] = {
4     0b00111100,
5     0b01000010,
6     0b101000101,
7     0b10000001,
8     0b101000101,
9     0b100111001,
10    0b01000010,
11    0b00111100 };
12
13 uint8_t image2[8] = {
14     0b00111100,
15     0b01000010,
16     0b100000101,
17     0b10000001,
18     0b101000101,
19     0b100111001,
20     0b01000010,
21     0b00111100 };
22
23 uint8_t image3[8] = {
24     0b00111100,
25     0b01000010,
26     0b10100001,
27     0b10000001,
28     0b101000101,
29     0b100111001,
30     0b01000010,
31     0b00111100 };
32
33 void setup(){
34     iarduino_Metro_Start();
35 }
36
37 void loop(){
38     Metro[0].on(image1); delay(1000);
39     Metro[0].on(image2); delay(1000);
40     Metro[0].on(image1); delay(1000);
41     Metro[0].on(image3); delay(1000);
42 }
```

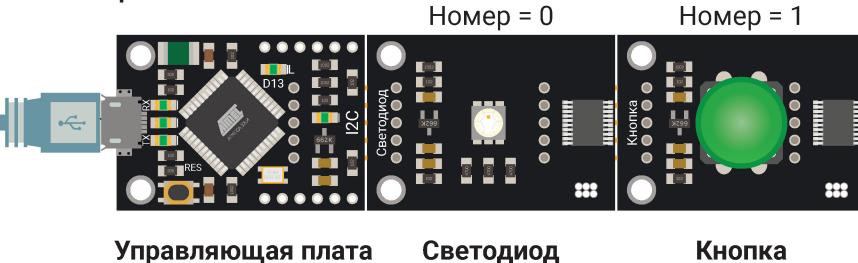
const uint8_t image1[8] – массив, содержащий 8 значений.

0b00111100 – значение соответствует строке матрицы, а 1 и 0 соответствуют светодиодам в этой строке. Таким образом, если указана 1, то в этой строке и в этом столбце светодиод будет гореть, а если указан 0, то светодиод гореть не будет.

Metro[0].on(image1); delay(1000); – Выводим изображение и выполняем задержку в 1 секунду. Благодаря выводу нескольких изображений друг за другом, возникает эффект анимации.

Лампа

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №9**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном уроке Вы разберёте принцип работы лампы, проект которой встречался Вам в блоке, где код уже был загружен в управляющую плату.

Дополнительное задание:

Задание 1

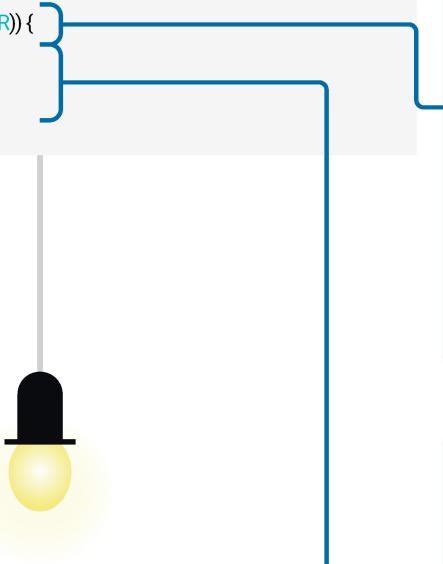
Модифицируйте лампу и сделайте так, чтобы она могла гореть разными цветами.



```

1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5     iarduino_Metro_Start();
6 }
7
8 void loop(){
9     if (Metro[1].read(KEY_TRIGGER)) {
10         Metro[0].on();
11     } else {
12         Metro[0].off();
13     }
14 }

```



Скетч 9

Metro[1].read(ПАРАМЕТР); – функция считывает состояние кнопки.

ПАРАМЕТРЫ (основные):

KEY_PUSHED – вернуть **true**, если было событие «**нажимается**» (при этом не важно как давно событие происходило), после чего функция будет возвращать **false**, пока опять не произойдёт событие «**нажимается**»;
KEY_RELEASED – вернуть **true**, если было событие «**отпускается**» (при этом не важно, как давно событие происходило), после чего функция будет возвращать **false**, пока опять не произойдёт событие «**отпускается**».

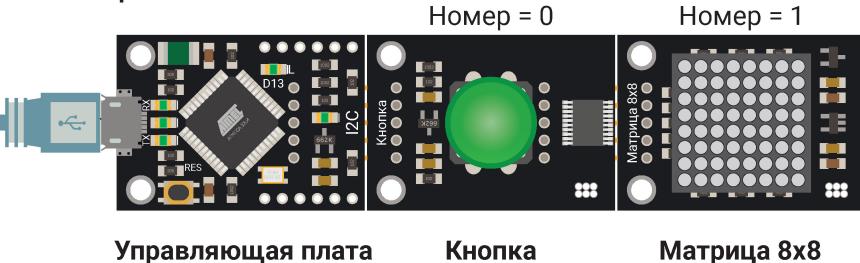
KEY_PRESSED – вернуть **true**, если кнопка нажата, иначе вернуть **false**. Данный параметр используется по умолчанию (если функция **read()** вызвана без параметра);
KEY_TRIGGER – вернуть либо **true**, либо **false**, меняя это значение с каждым новым нажатием кнопки (можно использовать для вкл/выкл устройств по нажатию);

if(Metro[1].read(KEY_TRIGGER)) – оператор **if** проверяет условие внутри скобок, и, если оно принимает значение «**ИСТИНА**» (**true**), то выполняется функция включения светоизода. Если же условие имеет значение «**ЛОЖЬ**» (**false**), тогда выполняется функция выключения светоизода.

В данном примере условие будет менять значение с «**ИСТИНЫ**» на «**ЛОЖЬ**» и обратно при каждом нажатии на кнопку.

Считаем ворон

Схема сборки:



Управляющая плата

Номер = 0

Кнопка

Номер = 1

Матрица 8x8

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №10**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

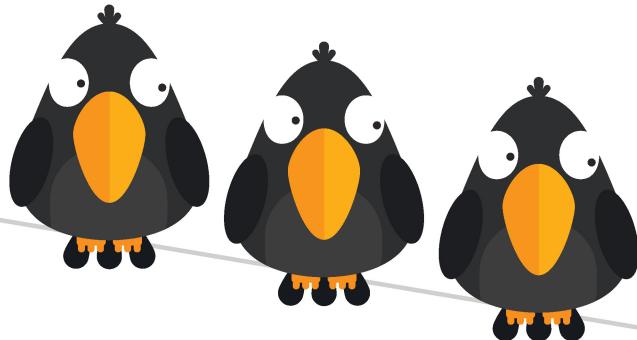
Описание:

В данном проекте Вы создадите удобный счётчик, с помощью которого можно считать всё что угодно, даже ворон!

Дополнительное задание:

Задание 1

Создайте скетч, в котором на дисплей будет выводиться счёт с спонснериами. Пример: «Количество ворон: 7»



Скетч 10

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 uint8_t i = 0;
4 String A;
5
6 void setup(){
7   iarduino_Metro_Start();
8   Metro[1].off();
9 }
10
11 void loop(){
12   if (Metro[0].read(KEY_PUSHED)){
13     i++;
14     A = i;
15     Metro[1].set(A, 245, 300);
16   }
17   if (i > 100){
18     i = 0;
19   }
20 }
```

`uint8_t i = 0;` – переменная, выполняющая функцию счётчика.

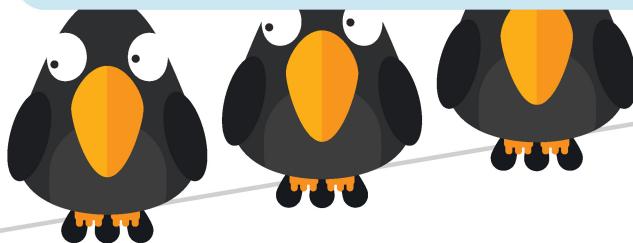
`String A;` – строка для хранения символов.

`if(Metro[0].read(KEY_PUSHED))` – оператор проверяет, нажимается кнопка или нет.

`i++;` – увеличивает значение переменной на 1.

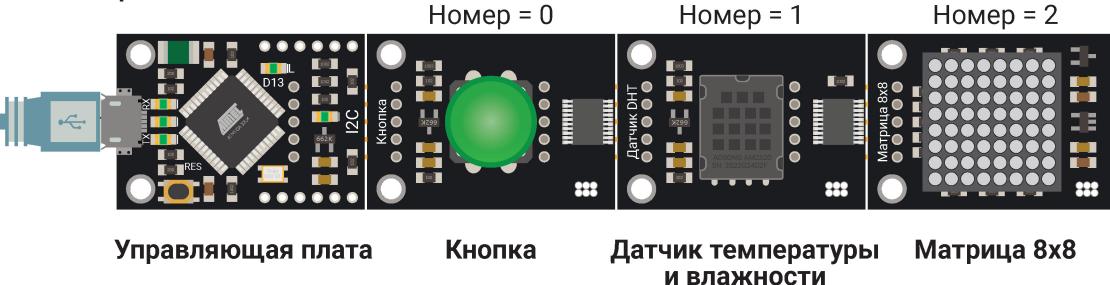
`A = i;` – сохраняет значение переменной «`i`» в строку «`A`», чтобы затем вывести эту строку на дисплей. Это сделано для того, чтобы числа больше 9 также могли отображаться на дисплее, но в режиме бегущей строки.

`Metro[1].set(A, СКОРОСТЬ, ПАУЗА);` – функция выводит значение строки `A` на экран матрицы с указанной **СКОРОСТЬЮ** (значение от 0 до 255) и **ПАУЗОЙ** между её повторным отображением на дисплее.



Метеостанция

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите [скетч №11](#).

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном проекте Вы познакомитесь с принципом работы табло метеостанции, проект которого встречался Вам в блоке, где код уже был загружен в управляющую плату.

36

Дополнительное задание:

Задание 1

Измените скетч таким образом, чтобы значения выводились по нажатию: первый раз выводилась температура, а второй раз - влажность.



Скетч 11

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 String A;
4
5 void setup() {
6   iarduino_Metro_Start();
7   Metro[2].off();
8 }
9
10 void loop() {
11   if (Metro[0].read(KEY_PUSHED)) {
12     A = Metro[1].read(DHT_TEMPERATURE);
13     A += "c ";
14     A += Metro[1].read(DHT_HUMIDITY);
15     A += "%";
16     Metro[2].set(A, 245);
17   }
18 }
```



Metro[1].read(ПАРАМЕТР); – функция, в зависимости от указанного **ПАРАМЕТРА**, возвращающая значение температуры или влажности.

ПАРАМЕТР:

DHT_TEMPERATURE – Вернуть текущую температуру в °C, от -40 до +80;

DHT_HUMIDITY – Вернуть относительную влажность воздуха в %, от 0 до 100;

DHT_CHANGED_TEM – Вернуть **true**, если температура изменилась, иначе вернуть **false**;

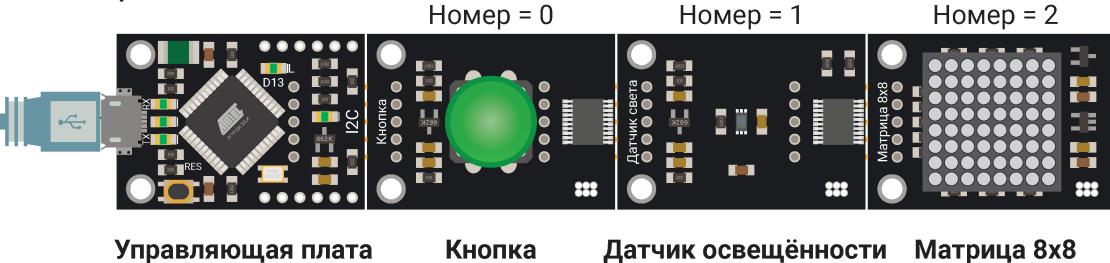
DHT_CHANGED_HUM – Вернуть **true**, если влажность изменилась, иначе вернуть **false**;

+= – оператор составного сложения, позволяющий к уже имеющемуся тексту добавить новый.



Люксметр

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №12**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

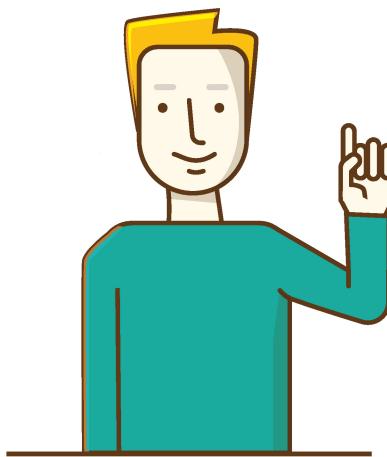
Описание:

Теперь Вы познакомитесь с принципом работы люксметра, который встречался Вам в блоке, где код уже был загружен в управляющую плату. Значение в **150-350** люкс является примерной нормой освещённости помещений в жилых домах.

Дополнительное задание:

Задание 1

Дополните скетч так, чтобы при слишком низком или слишком высоком значении освещённости на дисплей матрицы выводилось сообщение об этом.



Скетч 12

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 String A;
4
5 void setup() {
6   iarduino_Metro_Start();
7   Metro[2].off();
8 }
9
10 void loop() {
11   if (Metro[0].read(KEY_PUSHED)) {
12     A = Metro[1].read(DSL_LUX);
13     A += " lux";
14     Metro[2].set(A, 245);
15   }
16 }
```



Metro[1].read(ПАРАМЕТР); – функция, возвращающая значение освещенности в люксах (единицы измерения освещённости).

ПАРАМЕТР:

DSL_LUX - Вернуть текущую освещённость в лк, от 0 до 65535;

DSL_CHANGED - Вернуть **true**, если освещённость изменилась, иначе вернуть **false**;

Накаливания
40Вт



~5 люкс

Галогенная
28Вт



~10 люкс

Люминесцентная
15Вт



~12 люкс

Светодиодная
18Вт



~20 люкс

Диплом

об окончании учебного блока «Стажёр».

Диплом вручается



Дата

Подпись

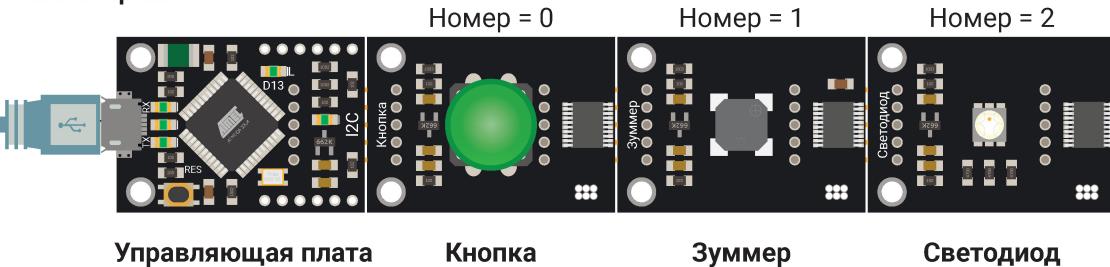
Дело мастера боится

в данном блоке Вы научитесь создавать сложные и интересные проекты, использовать которые Вы сможете и в повседневной жизни



Светофор

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №13**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

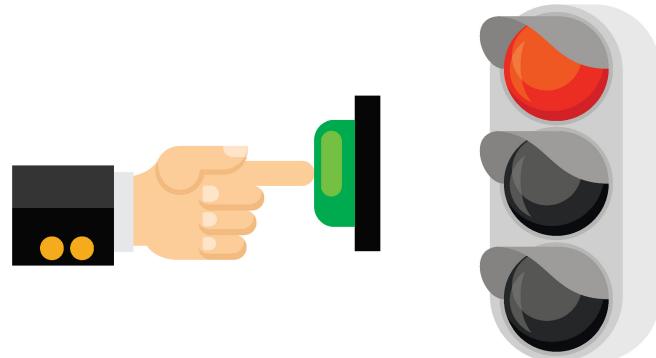
Описание:

В данном проекте Вы разберёте схему работы светофора, проект которого встречался Вам в блоке, где код уже был загружен в управляющую плату.

Дополнительное задание:

Задание 1

Измените режим светофора так, чтобы он работал без участия кнопки, а только по времени.



Скетч 13

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3
4 void setup(){
5   iarduino_Metro_Start();
6 }
7
8 void loop(){
9   Metro[2].on(0xffff0000);
10  if(Metro[0].read(KEY_PUSHED)){
11    Metro[2].off(); delay(500);
12    Metro[2].on(0xffff0000); delay(1000);
13    Metro[2].off(); delay(500);
14    Metro[2].on(0xffff0000); delay(1000);
15    Metro[2].off(); delay(500);
16    Metro[2].on(0xffff00); delay(2000);
17    Metro[2].off(); delay(500);
18    Metro[2].on(0x00ff00);
19    Metro[1].on(1000); delay(4000);
20    Metro[1].off();
21    Metro[2].off(); delay(500);
22    Metro[2].on(0x00ff00);
23    Metro[1].on(1000); delay(500);
24    Metro[1].off();
25    Metro[2].off(); delay(500);
26    Metro[2].on(0x00ff00);
27    Metro[1].on(1000); delay(500);
28    Metro[1].off();
29    Metro[2].off(); delay(500);
30    Metro[2].on(0x00ff00);
31    Metro[1].on(1000); delay(500);
32    Metro[1].off();
33    Metro[2].off(); delay(500);
34    Metro[2].on(0xffff00); delay(2000);
35    Metro[2].off(); delay(500);
36  }
37 }
```

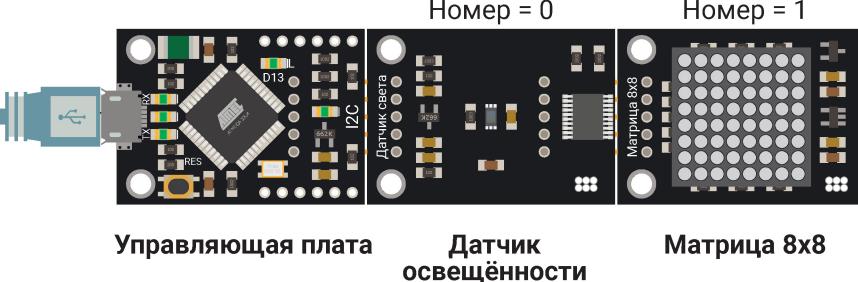
Если кнопка была нажата, то выполняется алгоритм, включающий и выключающий светодиод [2] и зуммер [1], имитируя тем самым работу светофора.

Так как в скетче используется функция **delay()**, то нажатие кнопки повторно в тот момент, когда светодиод меняет свой цвет, не приведёт ни к чему. Однако, само нажатие будет зафиксировано и после выполнения всех строчек скетча (с 11 по 35), цикл выполнится повторно столько раз, сколько раз была нажата кнопка.



Стой! Кто идёт?

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите [скетч №14](#).

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном проекте Вы симулируете работу системы контроля доступа по отпечатку пальца, когда, приложив палец к датчику, будет открываться шлагбаум.

Дополнительное задание:

Задание 1

Измените режим работы так, чтобы при приближении к датчику на дисплее возникла приветственная надпись, а после этого уже анимация открытия шлагбаума.



Скетч 14

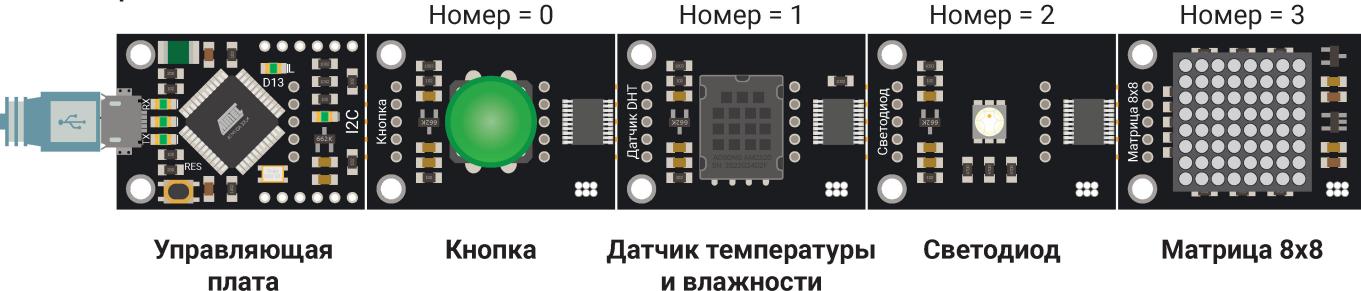
```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 uint8_t image1[8] = {
4     0b00000000,
5     0b00000000,
6     0b00000000,
7     0b00000000,
8     0b11111111,
9     0b11000000,
10    0b11000000,
11    0b11000000 };
12 uint8_t image2[8] = {
13     0b00000000,
14     0b00000000,
15     0b00000110,
16     0b00011000,
17     0b11100000,
18     0b11000000,
19     0b11000000,
20     0b11000000 };
21 uint8_t image3[8] = {
22     0b00000100,
23     0b00001000,
24     0b00010000,
25     0b00100000,
26     0b11000000,
27     0b11000000,
28     0b11000000,
29     0b11000000 };
30 void setup() {
31     iarduino_Metro_Start();
32     Metro[1].off();
33 }
34 void loop() {
35     if (Metro[0].read(DSL_LUX) <= 50) {
36         Metro[1].on(image2); delay(500);
37         Metro[1].on(image3); delay(3000);
38         Metro[1].on(image2); delay(500);
39     }
40     else {
41         Metro[1].on(image1);
42     }
}
```



Проверяем в цикле, не снизилась ли яркость ниже пограничного значения, и, если снизилась, то, значит, палец был приложен, и требуется «открыть шлагбаум». Если же яркость в норме, то «шлагбаум закрыт».

Климат-контроль

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №15**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

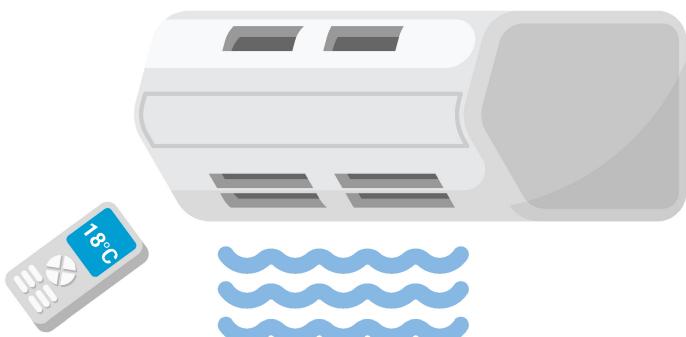
Пришло время освоить систему климат-контроля. Она позволяет задать температуру в диапазоне от 18°C до 32°C и отобразить цветом отношение заданной температуры к реальной (подробнее см. комментарии к скетчу).



Дополнительное задание:

Задание 1

Измените скетч так, чтобы на дисплей выводилось сообщение при уравновешивании температур.



```

1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 String A;
4 String B;
5 uint8_t your_temp = 18;
6
7 void setup(){
8   iarduino_Metro_Start();
9   Metro[3].off();
10  Metro[2].off();
11  Metro[0].set(1000, 5000, 10000);
12 }
13 void loop(){
14   if (Metro[0].read(KEY_PRESSED) == 0) {
15     if (your_temp > Metro[1].read(DHT_TEMPERATURE)) {
16       Metro[2].on(0x0000ff);
17     }
18     else if (your_temp < Metro[1].read(DHT_TEMPERATURE)) {
19       Metro[2].on(0xffff00);
20     } else {
21       Metro[2].on(0x00ff00);
22     }
23     A = Metro[1].read(DHT_TEMPERATURE);
24     A += "C ";
25     A += Metro[1].read(DHT_HUMIDITY);
26     A += "%";
27     Metro[3].set(A, 250); delay(4500);
28   }
29   if (Metro[0].read(KEY_HOLD2)) {
30     while (Metro[0].read(KEY_PRESSED)) {
31       your_temp++;
32       if (your_temp > 32) {
33         your_temp = 18;
34       }
35       B = your_temp;
36       Metro[3].set(B, 250); delay(2000);
37     }
38   }
39 }

```

Скетч 15

Metro[0].set(ВРЕМЯ 1, ВРЕМЯ 2, ВРЕМЯ 3); – задаёт время удержания кнопки для срабатывания функции `read()` с параметрами **KEY_HOLD1**, **KEY_HOLD2**, **KEY_HOLD3**.

ВРЕМЯ 1 – целочисленное значение, определяющее время удержания кнопки, по истечении которого функция `read()`, вызванная с параметром **KEY_HOLD1**, станет возвращать `true`;

ВРЕМЯ 2 – то же, что предыдущее значение, но относящееся к параметру **KEY_HOLD2** функции `read()`;

ВРЕМЯ 3 – то же, что предыдущее значение, но относящееся к параметру **KEY_HOLD3** функции `read()`;

Условия работы системы. Условно делятся на 3 части:

- **Вывод температуры и влажности на дисплей;**
- **Переход в режим задания температуры при удержании кнопки больше заданного времени.** При отпускании кнопки происходит выход из режима;
- **Сравнение заданной и реальной температур и вывод результата сравнения через цвет светодиода:**

синий - заданная температура выше реальной – нагрейте датчик;

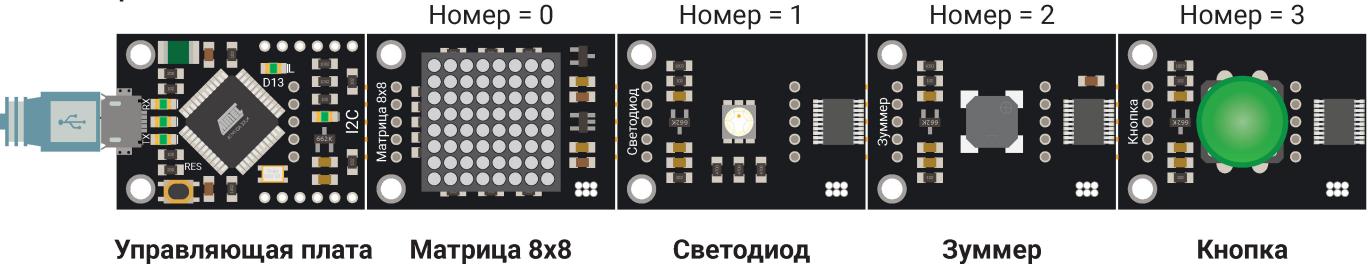
зелёный - температуры равны;

красный - заданная температура ниже реальной – охладите датчик;

Metro[3].set(A, 250); delay(4500); – время задержки при отображении текста составляет 4,5 секунды, следовательно, время удержания кнопки должно быть больше этого (**ВРЕМЯ 2 = 5000 мс**).

Игра «Запомни!»

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите [скетч №16](#).

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В данном уроке Вы создадите игру на внимательность! Необходимо запомнить, сколько раз светодиод моргнул тем или иным цветом, а потом дать правильный ответ на вопрос, нажав нужное количество раз на кнопку.

48

Дополнительное задание:

Задание 1

Усложните игру – увеличьте скорость, количество цветов и количество морганий!



Скетч 16

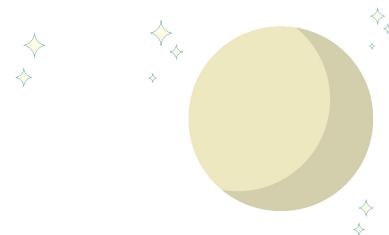
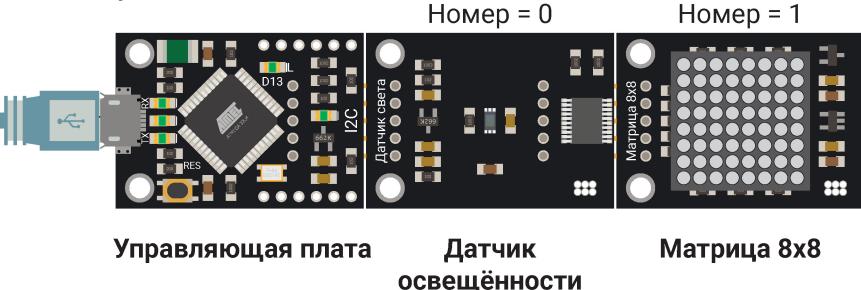
```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 uint32_t game_time = 0;
4 uint32_t guess_time = 10000;
5 uint32_t COLOR[] = {0xffff00, 0x00ff00, 0x0000ff};
6 uint8_t game_number = 10;
7 uint8_t your_number;
8 uint8_t right_answer;
9 uint8_t q_number, c_number;
10 uint8_t R_color, G_color, B_color;
11 uint8_t i;
12 bool flg_game;
13 String Question;
14 String Answer;
15 String Q_type[] = {"kpacHbLu", "3eLeHbLu", "cuHuu"};
16
17 void setup() {
18     iarduino_Metro_Start();
19     Metro[0].off(); Metro[1].off(); Metro[2].off();
20     randomSeed(analogRead(A0));
21     flg_game = true;
22 }
23 void loop() {
24     if (flg_game) {
25         flg_game = false;
26         i = 0; R_color = 0; G_color = 0; B_color = 0;
27         Metro[0].set("3anoMHu->", 250); delay(6000);
28         while (i < game_number) {
29             for (i; i < game_number; i++) {
30                 c_number = random(3);
31                 Metro[1].on(COLOR[c_number]); delay(700);
32                 Metro[1].off(); delay(700);
33                 if (c_number == 0) {R_color++;} else
34                 if (c_number == 1) {G_color++;} else
35                     {B_color++;}
36             }
37         }
38         q_number = random(3);
39         Question = "CkoLko pa3 ropeL " + Q_type[q_number] + '?';
40         Metro[0].set(Question, 250); delay(12000);
41         game_time = millis();
42     }
```

```
43     if (q_number == 0) {right_answer = R_color;} else
44     if (q_number == 1) {right_answer = G_color;} else
45         {right_answer = B_color;}
46     while ((game_time + guess_time) > millis()) {
47         if (Metro[3].read(KEY_PUSHED)) {your_number++;}
48         if (your_number > (game_number-1)) { your_number = 0; }
49         Answer = your_number;
50         Metro[0].set(Answer);
51     }
52     if ((game_time + 10000) < millis()) {
53         if (right_answer == your_number) {
54             Metro[2].on(MELODY(3)); delay(500);
55             Metro[2].off();
56             Metro[0].set("Victory!", 250, 300);
57             Metro[2].on(MELODY(8)); delay(27000);
58             your_number = 0; flg_game = true;
59         } else {
60             Metro[2].on(MELODY(1));
61             Metro[0].set("Defeat!", 250); delay(5000);
62             your_number = 0; flg_game = true;
63         }
64     }
65 }
```

random(min, max) – возвращает случайное число, где *min* – нижняя граница значений (включая *min*); *max* – верхняя граница значений (исключая *max*); **random(max)** идентична верхней функции, но *min* = 0. **randomSeed(ПАРАМЕТР)** – выбирает начальную позицию в последовательности чисел для функции **random()**. Функция выполняется один раз в блоке **void setup()**. **ПАРАМЕТР** – точка, с которой начинается генерация чисел. Чтобы получить последовательность различных случайных чисел, используйте данную функцию со случайным параметром, например с функцией **analogRead()**, считывающей значение с неподсоединеного вывода.

Альтернативная астрономия

Схема сборки:



Управляющая плата

Номер = 0

Номер = 1

Датчик
освещённости

Матрица 8x8

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите **скетч №17**.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

В этом уроке Вы создадите устройство, определяющее время суток по уровню освещённости.

В прошлых уроках изображения в скетче были записаны с помощью 0 и 1, а в этом уроке мы перевели их из **2-ичной системы счисления** в **10-ичную**.

Дополнительное задание:

Задание 1

Переведите числа из массивов с изображениями из 10-чной в 16-ричную систему счисления*.

*Для более подробного знакомства с системами счисления и способами перевода чисел из одной системы в другую, посетите страницу Wiki: <http://nabor.iarduino.ru>



Скетч 17

```

1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 uint8_t sun_1[8] = {0, 84, 56, 124, 56, 84, 0, 0};
4 uint8_t sun_2[8] = {0, 0, 84, 56, 124, 56, 84, 0};
5 uint8_t sun_3[8] = {0, 0, 42, 28, 62, 28, 42, 0};
6 uint8_t sun_4[8] = {0, 42, 28, 62, 28, 42, 0, 0};
7 uint8_t cloud_1[8] = {0, 0, 28, 54, 99, 255, 0, 0};
8 uint8_t cloud_2[8] = {0, 0, 7, 13, 24, 191, 0, 0};
9 uint8_t cloud_3[8] = {0, 0, 1, 131, 198, 239, 0, 0};
10 uint8_t cloud_4[8] = {0, 0, 192, 96, 49, 251, 0, 0};
11 uint8_t cloud_5[8] = {0, 0, 112, 216, 140, 254, 0, 0};
12 uint8_t moon_1[8] = {0, 0, 0, 0, 0, 0, 0, 0};
13 uint8_t moon_2[8] = {0, 12, 6, 2, 6, 12, 0, 0};
14 uint8_t moon_3[8] = {0, 12, 14, 14, 14, 14, 12, 0};
15 uint8_t moon_4[8] = {0, 60, 62, 62, 62, 62, 60, 0};
16 uint8_t moon_5[8] = {0, 60, 126, 126, 126, 126, 60, 0};
17 uint8_t moon_6[8] = {0, 60, 124, 124, 124, 124, 60, 0};
18 uint8_t moon_7[8] = {0, 48, 112, 112, 112, 112, 48, 0};
19 uint8_t moon_8[8] = {0, 48, 96, 64, 64, 96, 48, 0};
20
21 void setup() {
22 iarduino_Metro_Start();
23 }

```

Перевод из 2-ичной в 10-ичную систему счисления.

- Напротив каждой из 8 цифр написать **2** в степени, равной порядковому номеру этой цифры(считая справа налево).
- Умножить **2** в степени на цифру напротив.
- Сложить все результаты умножения.

Пример: $0xb11001101_2$

$$\begin{array}{r}
 \times 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \\
 \hline
 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \\
 \end{array}$$

$$\begin{array}{r}
 2^7 + 2^6 + 0 + 0 + 2^3 + 2^2 + 0 + 2^0 = \\
 = 128 + 64 + 0 + 0 + 8 + 4 + 0 + 1 = \mathbf{205}
 \end{array}$$

Таким образом: $11001101_2 = 205_{10}$

```

24 void loop() {
25 if (Metro[0].read(DSL_LUX) >= 150) {
26 Metro[1].on(sun_1); delay(1000);
27 Metro[1].on(sun_2); delay(1000);
28 Metro[1].on(sun_3); delay(1000);
29 Metro[1].on(sun_4); delay(1000);
30 } else if (Metro[0].read(DSL_LUX) >= 75 && Metro[0].read(DSL_LUX) < 150) {
31 Metro[1].on(cloud_1); delay(1000);
32 Metro[1].on(cloud_2); delay(1000);
33 Metro[1].on(cloud_3); delay(1000);
34 Metro[1].on(cloud_4); delay(1000);
35 Metro[1].on(cloud_5); delay(1000);
36 } else {
37 Metro[1].on(moon_1); delay(1000);
38 Metro[1].on(moon_2); delay(1000);
39 Metro[1].on(moon_3); delay(1000);
40 Metro[1].on(moon_4); delay(1000);
41 Metro[1].on(moon_5); delay(1000);
42 Metro[1].on(moon_6); delay(1000);
43 Metro[1].on(moon_7); delay(1000);
44 Metro[1].on(moon_8); delay(1000);
45 }
46 }

```

Перевод из 10-ичной в 2-ичную систему счисления.

- Делим число на 2 до тех пор, пока остаток от деления не станет меньше 2.
- Записываем цифры остатков от деления в обратном порядке.

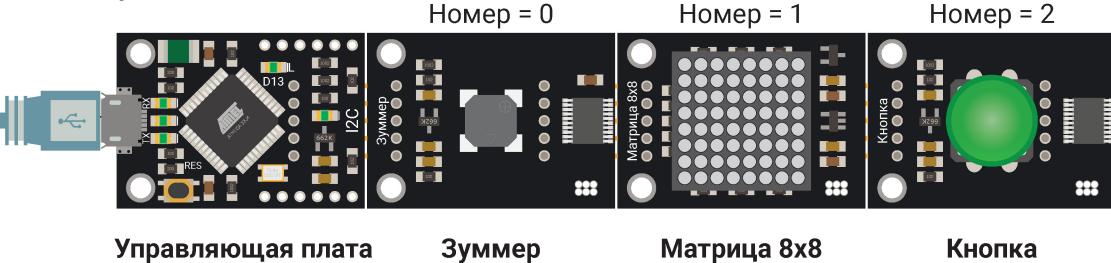
Пример: 25_{10}

$$\begin{array}{r}
 25 \Big| 2 \\
 24 \Big| 12 \quad 2 \\
 1 \quad \Big| 12 \quad 2 \\
 0 \quad \Big| 6 \quad 2 \\
 0 \quad \Big| 3 \quad 2 \\
 0 \quad \Big| 1 \quad 2 \\
 1
 \end{array}$$

Таким образом: $25_{10} = 11001_2$

Игра «Электронный тир»

Схема сборки:



Управляющая плата

Номер = 0

Зуммер

Номер = 1

Матрица 8x8

Номер = 2

Кнопка

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите [скетч №18](#).

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

Пришло время вновь создать игру! Цель её будет в том, чтобы нажать кнопку в то время, когда на экране появится символ «Х». Количество попыток ограничено, а частота смены символов постоянно растёт, так что будьте внимательны!

Дополнительное задание:

Задание 1

Усложните игру, добавив большее количество элементов, а также возможность задавать нужный символ кнопкой.



Скетч 18

```
1 #include <Wire.h>
2 #include <iarduino_Metro.h>
3 uint8_t    img_list [3][8] = { {0, 66, 36, 24, 24, 36, 66, 0},
4                                {0, 126, 66, 66, 66, 66, 126, 0},
5                                {0, 60, 66, 66, 66, 66, 60, 0} };
6 uint8_t    life      = 8;
7 uint16_t   game_level = 1500;
8 bool       start_game = true;
9 bool       victory   = false;
10 bool      game_over = false;
11 bool      life_numbers = true;
12 uint8_t   img_number;
13 uint8_t   answer;
14 String    print_life;
15
16 void setup() {
17     iarduino_Metro_Start();
18     randomSeed(analogRead(0));
19     Metro[0].off(); Metro[1].off();
20     Metro[1].set("honagu B X", 250); delay(5000);
21     Metro[1].off();
22 }
23 void loop() {
24     if (start_game) {
25         if (Metro[2].read(KEY_PUSHED) == 0) {
26             if (life_numbers) {
27                 life_numbers= false;
28                 print_life = "nonblTok: ";
29                 print_life += life;
30                 Metro[1].set(print_life, 251); delay(5000);
31                 Metro[1].off(); delay(500);
32             }
33             img_number = random(3);
34             Metro[1].on(img_list[img_number]); delay(game_level);
35             Metro[1].off();
36         }
37     } else {
38         answer = img_number;
39         if (answer == 0) {
40             Metro[0].on(MELODY(3)); delay(500);
41             if (game_level <= 150) {
```

```
42             start_game = false;
43             victory = true;
44         }
45         else {game_level -= 50;}
46     }
47     else {
48         Metro[0].on(MELODY(1)); delay(500);
49         if (life <= 1) {
50             start_game = false;
51             game_over = true;
52         }
53         else {life--;}
54         life_numbers= true;
55     }
56 }
57
58 if (victory) {
59     victory = false;
60     Metro[0].on(MELODY(random(4, 23)));
61     Metro[1].set("ho6eguL!!!", 250, 200); delay(15000);
62     game_level = 1500;
63     life = 8;
64     start_game = true;
65 }
66 if (game_over) {
67     game_over = false;
68     Metro[0].on(MELODY(12));
69     Metro[1].set("hpourpaL...", 250, 200); delay(10000);
70     game_level = 1500;
71     life = 8;
72     start_game = true;
73 }
74 }
```

Диплом

об окончании учебного блока «Дело мастера боится».

Диплом вручается



Дата

Подпись

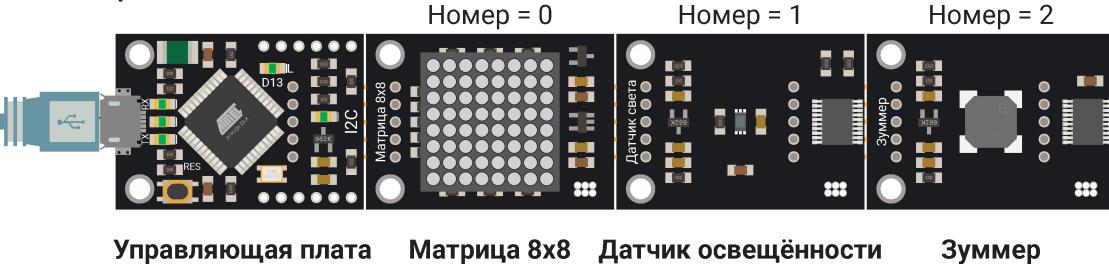
Светлая голова

в данном блоке Вам придётся, как и подобает истинному профессионалу, создать по нашим наброскам реальные проекты



Умные часы

Схема сборки:



Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите Ваш скетч.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

Предлагаем Вам самостоятельно написать скетч, который будет имитировать работу «умных» часов: в зависимости от уровня освещенности меняется яркость матрицы, отображающей время, а зуммер постоянно сигнализирует о прошествии 1 минуты.

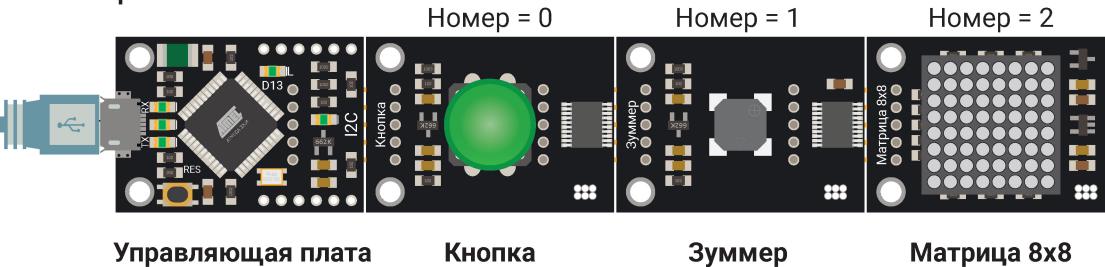
Дополнительное задание:

Задание 1

Добавьте в цепь кнопку, а в скетч - вывод показаний уровня яркости, температуры и влажности при нажатии кнопки.



Схема сборки:



Управляющая плата

Номер = 0

Кнопка

Номер = 1

Зуммер

Номер = 2

Матрица 8x8

Действия:

Шаг 1

Подключите указанные модули к управляющей плате, соблюдая последовательность.

Шаг 2

Подключите кабель USB к управляющей плате и загрузите Ваш скетч.

Важно!

Не соединяйте модули при подключенном кабеле USB!

Описание:

Предлагаем Вам самостоятельно создать «переводчик азбуки Морзе», который будет считать время нажатия на кнопку и присваивать ему параметр «точка» или «тире», а зуммер будет дублировать нажатие кнопки звуковым сигналом. Собрав верную комбинацию, на дисплее должна отобразиться цифра.

Дополнительное задание:

Таблица цифр из азбуки Морзе:

0	— — — —	— — — —	— — — —	— — — —	— — — —
1	• — — —	— • — —	— — • —	— — — •	— — — —
2	• • — —	• — • —	— — • •	— — — —	— — — —
3	• • • —	• • — •	— — • •	— — — —	— — — —
4	• • • •	• • — —	— — • •	— — — •	— — — —
5	• • • •	• • — —	— — • •	— — — •	• — — —
6	— — — —	• — — —	— — • •	— — — •	• • — —
7	— — — —	— — — —	• — — —	— — — •	• • — —
8	— — — —	— — — —	— — — —	• — — —	• • — —
9	— — — —	— — — —	— — — —	— — — —	• • • •

Диплом

об окончании учебного блока «Светлая голова».

Диплом вручается



Дата

Подпись

Вместо послесловия





МЫ В INSTAGRAM

@IARDUINO.RU | #IARDUINO



МЫ В ВКОНТАКТЕ

WWW.VK.COM/IARDUINO



МЫ В YOUTUBE

WWW.YOUTUBE.COM/IARDUINO