

Стартовый набор BBC micro:bit с комплектом датчиков

Проекты и эксперименты с micro:bit

Содержание

1 Для чего используется micro:bit?	4
2 Уроки	5
2.1 Подготовка к работе	5
2.2 Настройка среды разработки на ПК	5
2.3 Настройка среды на смартфоне	6
2.4 Основы ПО MakeCode	6
2.5 Плата расширения	7
2.6 Урок: Эффект биения сердца	8
2.7 Урок: Система «Кто быстрее» / Кнопки ответа	10
2.8 Урок: Текущий песок	11
2.9 Урок: Самодельный термометр	12
2.10 Урок: Компас / Стороны света	13
2.11 Урок: Работа со звуком (Воспроизведение музыки)	15
2.12 Урок: Электронная игральная кость (Датчик тряски)	16
2.13 Урок: День и Ночь (Датчик освещенности)	18
2.14 Урок: Эксперимент с двухцветным светодиодом	19
2.15 Урок: Эксперимент с RGB-светодиодом	22
2.16 Урок: Эксперимент с 7-цветным LED-светодиодом	25
2.17 Урок: Эксперимент с релейным модулем	28
2.18 Урок: Управление тремя LED-светодиодами	31
2.19 Урок: Эксперимент с лазерным модулем	34
2.20 Урок: Эксперимент с лёгкой (тактильной) кнопкой	36
2.21 Урок: Эксперимент с датчиком наклона	40
2.22 Урок: Эксперимент с герконовым датчиком	43
2.23 Урок: Эксперимент с активным зуммером	46
2.24 Урок: Эксперимент с пассивным зуммером	49
2.25 Урок: Эксперимент с U-образным фотоэлектрическим датчиком	51
2.26 Урок: Эксперимент с четырехразрядным цифровым индикатором (TM1637)	55
2.27 Урок: Эксперимент по обнаружению капель дождя	59
2.28 Урок: Эксперимент с джойстиком PS2	62
2.29 Урок: Эксперимент с модулем потенциометра	66
2.30 Урок: Эксперимент с датчиком Холла	68

2.31 Урок: Эксперимент с аналоговым датчиком температуры.....	71
2.32 Урок: Эксперимент с датчиком звука (Акустическое реле).....	72
2.33 Урок: Эксперимент с фоторезистором (Умный светильник).....	75
2.34 Урок: Эксперимент с датчиком пламени (Пожарная сигнализация)	77
2.35 Урок: Эксперимент с датчиком дыма (Детектор газа).....	80
2.36 Урок: Управление с помощью ИК-пульта.....	83
2.37 Урок: Сенсорный переключатель	86
2.38 Урок: Ультразвуковой дальномер (Измерение расстояния).....	88
2.39 Урок: Жидкокристаллический дисплей I2C 1602	91
2.40 Урок: Эксперимент с датчиком температуры DS18B20	93
2.41 Урок: Эксперимент с поворотным энкодером	94
2.42 Урок: Эксперимент по управлению сервоприводом	97
2.43 Урок: Эксперимент с датчиком влажности DHT11	98
2.44 Урок: ИК-сигнализация обнаружения препятствий.....	100
2.45 Урок: Эксперимент с OLED дисплеем.....	103
2.46 Урок: Измерение атмосферного давления (BMP280)	105
2.47 Урок: Эксперимент с датчиком освещенности BH1750.....	107
2.48 Урок: Модуль часов реального времени DS1302.....	111
2.49 Урок: Эксперимент с датчиком линии (препятствия)	114
2.50 Урок: Кнопочное управление светодиодом	116
2.51 Урок: Умный вентилятор (Эксперимент с тушением огня)	118
2.52 Урок: Датчик движения (PIR).....	121
2.53 Урок: Автоматическая система полива	124

1 Для чего используется micro:bit?

micro:bit может взаимодействовать с различными электронными компонентами через зажимы типа «крокодил». Он поддерживает чтение данных с датчиков, управление сервоприводами и светодиодными RGB-лентами, что позволяет легко использовать его в различных сценариях обучения программированию и разработки устройств.

Кроме того, micro:bit можно применять для создания электронных игр, интерактивных звуковых и световых проектов, управления роботами, проведения научных экспериментов, разработки носимой электроники и многое другое.

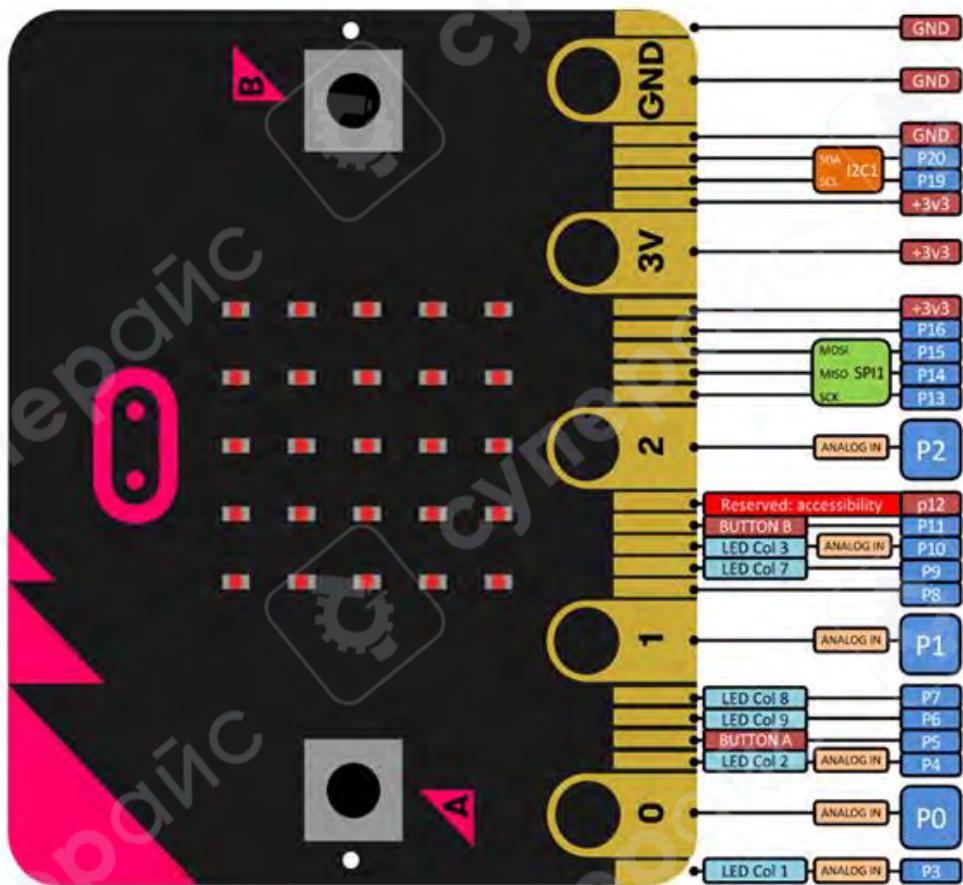


Схема распиновки платы

2 Уроки

2.1 Подготовка к работе

Цель: Подготовить оборудование и проверить комплектность.

1. **Проверка оборудования:**

- Возьмите плату BBC micro:bit.
- Найдите кабель Micro-USB.

Важно: Кабель должен поддерживать передачу данных, а не только зарядку. В комплекте идет правильный кабель.

- Подготовьте плату расширения (Shield) и датчики (пока не подключаем).

2. **Питание:**

- micro:bit можно питать через USB-кабель от компьютера.
- Для автономной работы используется батарейный отсек.

2.2 Настройка среды разработки на ПК

Цель: Научиться программировать плату через компьютер.

1. **Подключение:**

- Подключите micro:bit к компьютеру через USB-кабель.
- Компьютер определит плату как обычную флешку с названием MICROBIT.

2. **Среда программирования:**

ПРЕДПОЧИТЕЛЬНО:

- Вам не нужно ничего устанавливать. Откройте браузер (Chrome, Edge и т.д.).
 - Перейдите на сайт: makecode . microbit . org (**доступно на русском языке**)
 - Нажмите «New Project» (Новый проект) и дайте ему имя (например, "Test").
3. **Интерфейс:**
- Вы увидите рабочее поле. Слева — симулятор платы, посередине — блоки кода, справа — поле для сборки программы.

Если у вас нет доступа к интернету или вы предпочитаете установленные программы:

1. Откройте папку «2.makecode 离线版本» (MakeCode Offline).
2. Найдите установочный файл и запустите его.
3. Следуйте инструкциям установщика.
4. После установки на рабочем столе появится значок MakeCode. Запускайте его для программирования.

Если после подключения micro:bit компьютер не реагирует или в Диспетчере устройств появляется «Неизвестное устройство»:

1. Откройте папку «1.串口驱动程序» (Драйверы).
2. Запустите установщик.
3. Нажмите кнопку Install.
4. После завершения отключите и снова подключите плату.

Программирование на Python (для продвинутых)

Если вы хотите писать код на языке Python:

1. Откройте папку «3.python 开发环境».

2. Установите редактор Mu Editor.
3. В редакторе выберите режим BBC micro:bit.

2.3 Настройка среды на смартфоне

Цель: Программирование и прошивка через Bluetooth без проводов.

1. **Установка приложения:**

- Зайдите в App Store (iOS) или Google Play (Android).
- Найдите и скачайте официальное приложение «micro:bit».

Альтернативный способ (Только для Android, если нет интернета):

1. Скопируйте файл microbit11.apk из папки «第 3 讲 : microbit 在手机端搭建一个基本的开发平台» на ваш телефон.
2. Откройте его на телефоне через файловый менеджер.
3. Если система спросит разрешение на «Установку из неизвестных источников», разрешите её.
4. Дождитесь окончания установки.

2. **Сопряжение:**

- Включите Bluetooth на телефоне. Откройте приложение, нажмите «Connection» или «Pair micro:bit».
 - Переведите плату в режим сопряжения:
 1. Зажмите кнопки A и B одновременно.
 2. Не отпуская их, нажмите и отпустите кнопку RESET (сзади).
 3. Держите A и B, пока на экране не побежит надпись «PAIRING MODE» и не появится узор.
 - В приложении на телефоне появится запрос: «Введите узор, который видите на плате». Нажимайте на квадратики в приложении, чтобы скопировать рисунок с платы.

3. **Загрузка кода:** Теперь код можно писать на телефоне и отправлять на плату.

2.4 Основы ПО MakeCode

Цель: Понять, как создать и загрузить первую программу (<https://makecode.microbit.org/>).

1. **Основное блоки:**

- On Start (При начале): Код внутри сработает один раз при включении.
- Forever (Всегда/Постоянно): Код будет выполняться бесконечно по кругу.

2. **Пример "Сердце":**

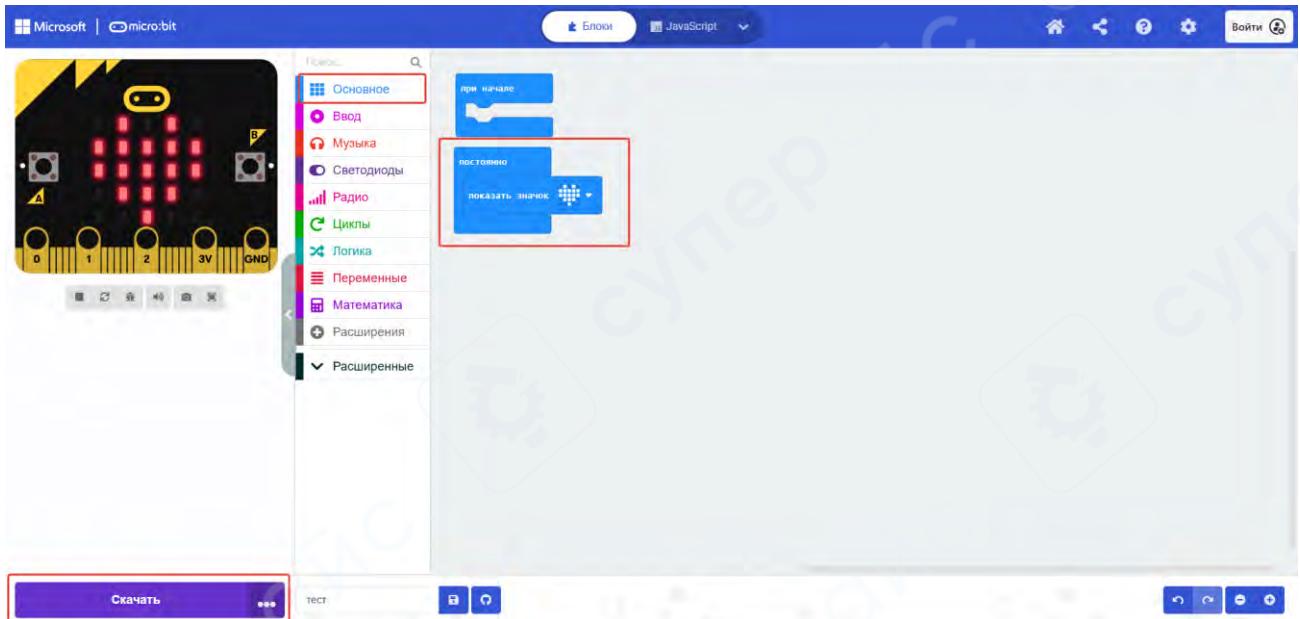
- Зайдите в меню Basic (Основное).
- Перетащите блок show icon (показать значок) внутрь блока Forever. Выберите иконку сердца.

- В левом углу экрана можно проверить работу кода, нажав Play.

3. **Загрузка (Прошивка):**

- Нажмите кнопку Download (Скачать) внизу экрана.
- Сохранится файл с расширением .hex.
- Перетащите этот файл мышкой на диск MICROBIT (как на обычную флешку).

- Желтый диод на плате замигает (идет запись). После окончания платы перезагрузится и покажет сердце.



2.5 Плата расширения

В этом уроке мы разберем, как соединять micro:bit с датчиками. На самой плате micro:bit контакты плоские («золотые зубчики»), и к ним трудно подключить много проводов. Для этого мы используется **плата расширения**.

1. Устройство платы

Эта плата выполняет роль «переходника». Она выводит каждый контакт micro:bit на удобные штырьки.

- Возьмите плату расширения и micro:bit.
- Вставьте micro:bit в черный слот (разъем) в верхней части платы расширения.
- Важно:** Убедитесь, что лицевая сторона micro:bit (с кнопками и светодиодами) направлена в ту же сторону, что и надписи на плате расширения. Плата должна войти плотно.

На плате расширения выведены контакты, подписанные маркировкой. Они разделены на группы:

- Контакты ввода-вывода (P0 – P20):**

Используются для подключения светодиодов, кнопок и простых датчиков.

P0, P1, P2: Основное пины (поддерживают аналоговый и цифровой сигнал). С ними работать проще всего.

P3 – P20: Дополнительные пины для управления (обратите внимание, часть пинов используется самим micro:bit для дисплея, поэтому при их использовании светодиодная матрица может отключаться).

- Пины питания (Power):**

3.3V: Выход напряжения 3.3 Вольта (стандартное напряжение для логики micro:bit).

5V: Выход напряжения 5 Вольт. Используется для более мощных датчиков или моторов.

GND: «Земля» (минус). Обязательно подключается к минусу любого датчика.

- Специальные интерфейсы (отдельная группа справа внизу):**

SDA / SCL: Интерфейс I2C. Используется для подключения дисплеев (OLED/LCD) и сложных цифровых датчиков (гироскопы, дальномеры).

RXD / TXD: Интерфейс UART (последовательный порт). Используется для обмена данными с другими устройствами (например, Bluetooth или Wi-Fi модули).

На левой стороне платы расширения есть разъем Micro-USB.

Он используется для подачи внешнего питания (5V), если вы подключаете много датчиков или сервомоторы, которым не хватает энергии от самого micro:bit.

При работе с простыми датчиками дополнительное питание в этот разъем можно не подключать, достаточно питания от micro:bit.



2.6 Урок: Эффект биения сердца

Цель: Создать анимацию пульсирующего сердца на светодиодном экране micro:bit, используя цикл и временные задержки.

Алгоритм работы:

Для создания эффекта «живого» сердца мы будем чередовать два изображения (большое сердце и маленькое сердце) с небольшой паузой между ними. Это создаст иллюзию сокращения мышцы.

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым. В данном уроке начальная настройка не требуется.

Блок «Постоянно» (Forever):

- Все действия будем помещать внутрь этого блока, чтобы анимация повторялась бесконечно.

Шаг 1: Показать большое сердце

- Зайдите в меню **Basic (Основное)**.
- Выберите блок **show icon** (показать значок) и перетащите его внутрь блока **Forever**.
- В выпадающем меню выберите иконку **Большое сердце** (заполненное).

Шаг 2: Пауза

- Из меню **Basic (Основное)** возьмите блок **pause (ms) (пауза)**.
- Поместите его под блоком с иконкой.
- Установите значение **200**. Это означает задержку в 200 миллисекунд (0.2 секунды).

Шаг 3: Показать маленькое сердце

- Снова добавьте блок **show icon (показать значок)** под паузой.
- В меню выберите иконку **Маленькое сердце** (точка в центре).

Шаг 4: Пауза

- Добавьте еще один блок **pause (ms) (пауза)** в самый низ.
- Установите значение **200**.
- **Важно:** Эта вторая пауза нужна, чтобы маленькое сердце оставалось на экране некоторое время перед тем, как цикл начнется заново и снова появится большое сердце.

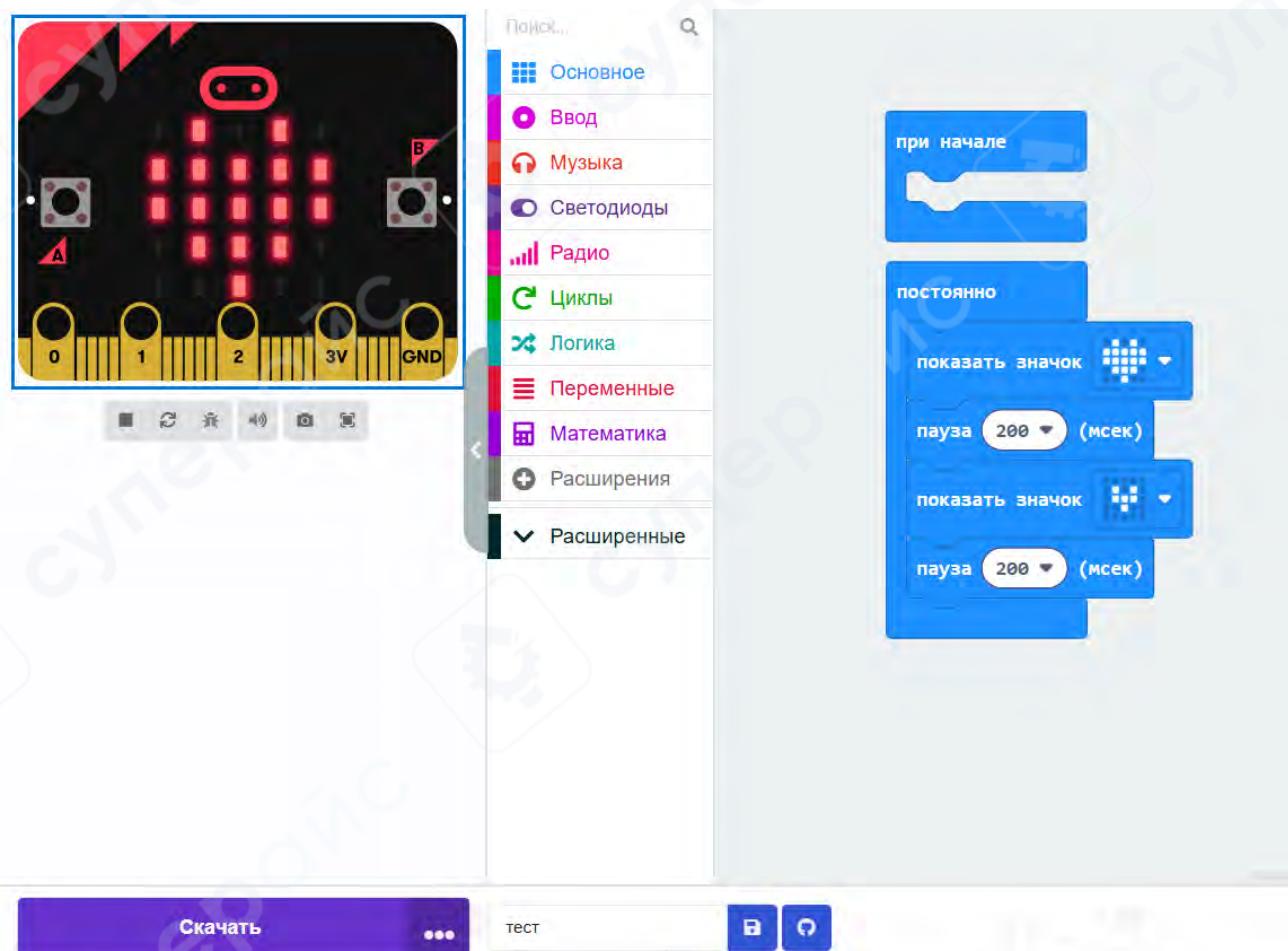
Результат:

После загрузки программы на экране micro:bit будет ритмично меняться изображение: Большое сердце -> Пауза -> Маленькое сердце -> Пауза.

Загрузка файла:

В папке с уроком вы можете найти готовый файл **心跳.hex**.

Вы можете скопировать этот файл напрямую на диск **MICROBIT**, чтобы сразу проверить результат, не набирая код вручную.



2.7 Урок: Система «Кто быстрее» / Кнопки ответа

Цель: Научиться работать с кнопками A и B, используя события ввода (Input Events). Создать простейшую систему индикации того, какая кнопка была нажата.

Принцип работы:

В отличие от предыдущих уроков, здесь блоки «При начале» и «Постоянно» остаются пустыми. Программа находится в режиме ожидания и срабатывает только тогда, когда происходит событие — физическое нажатие на кнопку.

Если нажата кнопка A (слева) -> Показываем стрелку влево.

Если нажата кнопка B (справа) -> Показываем стрелку вправо.

Программный код (сборка блоков):

Очистка рабочего поля:

Блоки **On Start (При начале)** и **Forever (Постоянно)** нам не понадобятся. Вы можете удалить их или просто оставить пустыми (как на скриншоте), они не повлияют на работу.

Шаг 1: Программируем кнопку A

- Зайдите в меню **Input (Ввод)** (фиолетовые блоки).
- Выберите блок **on button A pressed** (**когда кнопка A нажата**) и перетащите его на рабочее поле.
 - Зайдите в меню **Basic (Основное)**.
 - Выберите блок **show leds** (**показать светодиоды**) и поместите его внутрь фиолетового блока кнопки A.
 - Нарисуйте на матрице **стрелку влево** (закрасьте квадратики мышкой, как показано на левом скриншоте).

Шаг 2: Программируем кнопку B

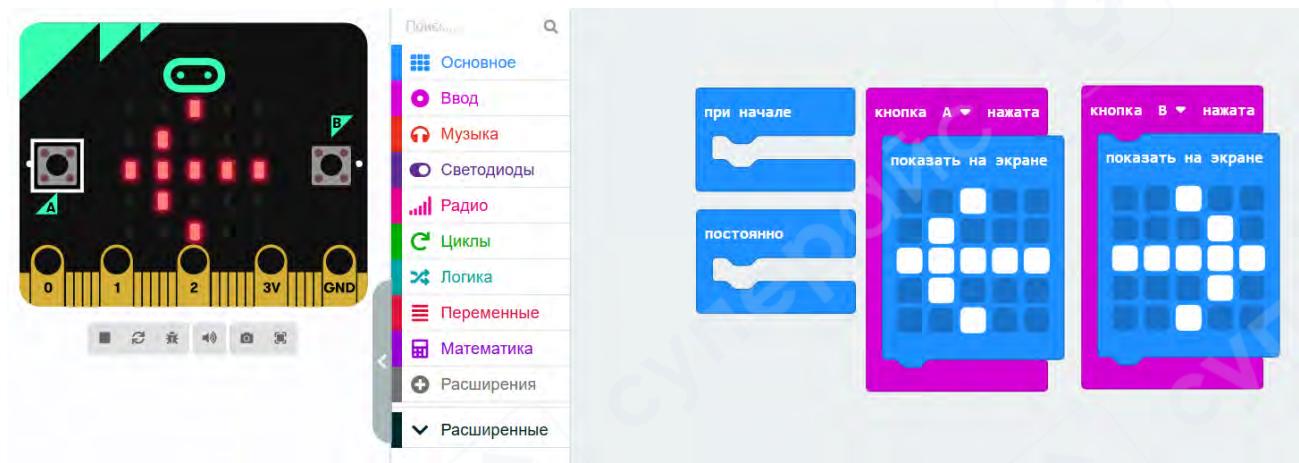
- Снова зайдите в меню **Input (Ввод)**.
- Возьмите еще один блок **on button A pressed**.
- В выпадающем списке в этом блоке измените **A** на **B**. Теперь блок называется **on button B pressed** (**когда кнопка B нажата**).
 - Зайдите в меню **Basic (Основное)**.
 - Выберите блок **show leds** (**показать светодиоды**) и поместите его внутрь блока кнопки B.
 - Нарисуйте на матрице **стрелку вправо** (как показано на правом скриншоте).

Результат:

После загрузки программы экран micro:bit будет выключен, пока вы ничего не трогаете.

- Нажмите кнопку A — загорится стрелка влево.
- Нажмите кнопку B — загорится стрелка вправо.

Это имитирует систему викторины: стрелка указывает на игрока, который нажал свою кнопку.



Нажата кнопка А

Загрузка файла:

В папке с уроком находится файл **抢答器.hex**.

Скопируйте его на диск **MICROBIT** для быстрой проверки работы устройства.

2.8 Урок: Текущий песок

Цель: Научиться использовать акселерометр для определения положения платы в пространстве (наклон, переворот, встряхивание) и управлять светодиодами в зависимости от гравитации.

Принцип работы:

Мы используем блоки событий «Ввод» (Input), которые срабатывают при изменении положения платы.

- Если наклонить плату влево — диоды зажигаются слева.
- Если вправо — справа.
- Если перевернуть — сверху или снизу.
- При встряхивании «песок» перемешивается.

Программный код (сборка блоков):

Все блоки берутся из меню **Input (Ввод)** (фиолетовые) и **Basic (Основное)** (синие). Порядок добавления блоков не важен.

Событие 1: Встряхивание (Перемешивание)

- Возьмите блок **on shake** (когда встряхивание / 当 振动).
- Внутрь поместите блок **show leds**. Выберите иконку «Шахматная доска» (или разбросанные точки), чтобы показать, что песок рассыпался по всему экрану.

Событие 2: Логотипом вверх (Песок внизу)

- Возьмите блок **on logo up** (когда логотипом вверх). Это стандартное вертикальное положение платы.

- Внутрь поместите **show leds**.
- Закрасьте **верхние 3 ряда** светодиодов.

Событие 3: Логотипом вниз (Песок вверху)

- Возьмите блок **on logo down** (когда логотипом вниз). Это положение «вверх ногами» (разъем USB смотрит вниз).

- Внутрь поместите **show leds**.
- Закрасьте **нижние 3 ряда** светодиодов.

Событие 4: Наклон влево (Песок слева)

- Возьмите блок **on tilt left** (когда наклон влево).
- Внутрь поместите **show leds**.
- Закрасьте **левые 3 столбца** светодиодов.

Событие 5: Наклон вправо (Песок справа)

- Возьмите блок **on tilt right** (когда наклон вправо).
- Внутрь поместите **show leds**.
- Закрасьте **правые 3 столбца** светодиодов.

Результат:

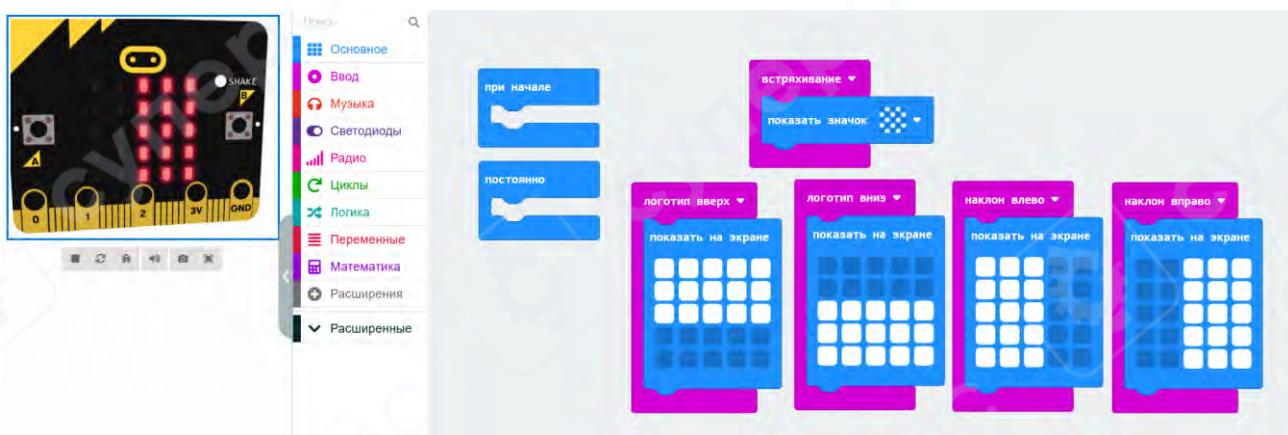
Загрузите код в micro:bit. Отключите кабель и подключите батарейный отсек (для свободы движений), либо просто аккуратно вращайте плату на кабеле.

- Поворачивайте плату в разные стороны: скопление светящихся точек будет «стекать» в самую нижнюю точку, как вода или песок.
- Резко встряхните плату: картинка сменится на узор, символизирующий хаос.

Файлы:

Файл урока называется **流动的沙子.hex**.

Скопируйте его на диск **MICROBIT** для проверки.



Наклон вправо

2.9 Урок: Самодельный термометр

Цель: Создать устройство, которое непрерывно измеряет и отображает текущую температуру окружающей среды на светодиодном дисплее.

Принцип работы:

Micro:bit не имеет отдельного датчика температуры воздуха. Он использует датчик, встроенный в главный процессор. Поскольку процессор почти не нагревается при простых задачах, его температура примерно равна температуре окружающей среды. Программа будет работать в бесконечном цикле, постоянно обновляя показания на экране.

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

- Мы используем этот блок, чтобы температура обновлялась в реальном времени, а не показывалась только один раз при включении.

Шаг 1: Вывод числа

- Зайдите в меню **Basic (Основное)**.
- Выберите блок **show number** (показать число).
- Перетащите его внутрь блока **Forever**.

Шаг 2: Получение данных о температуре

- Зайдите в меню **Input (Ввод)** (фиолетовые блоки).
- Найдите овальный блок **temperature (°C) (температура)**.
- Перетащите этот овальный блок **внутрь** блока **show number** (показать число), в то место, где обычно пишется цифра 0.

Результат:

После загрузки программы на светодиодной матрице начнут бежать цифры.

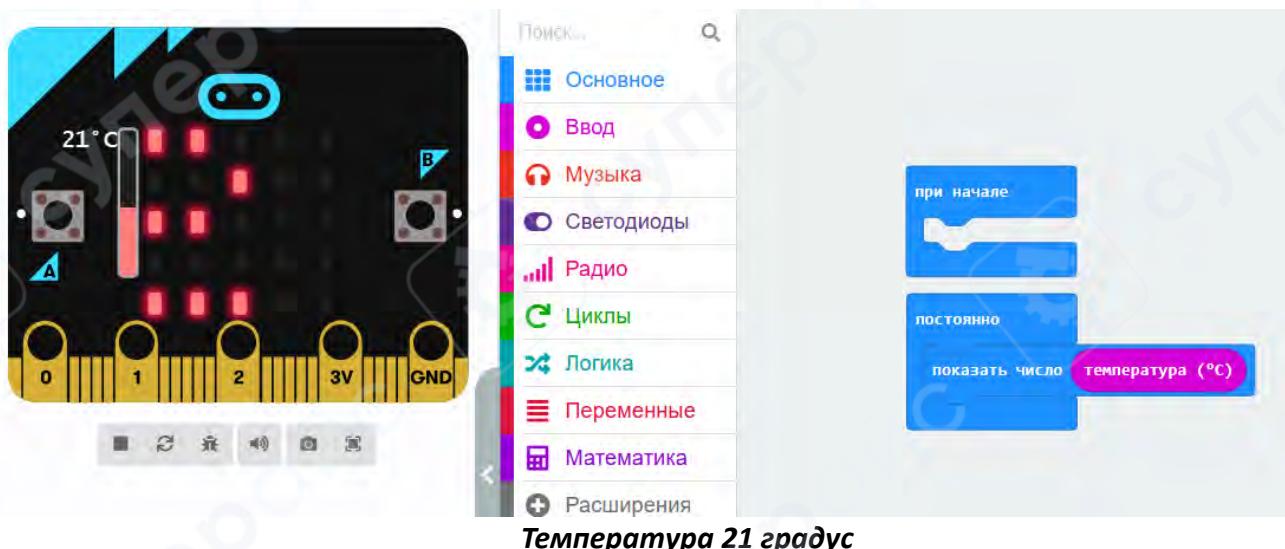
Например, если в комнате 25 градусов, по экрану проплынет число 25. Поскольку матрица маленькая (5x5), двузначные числа показываются в режиме бегущей строки.

Примечание: Если вы приложите палец к процессору (черный чип на задней стороне платы), показания начнут расти.

Файлы:

Файл урока: **自制温度计.hex**.

Скопируйте его на диск **MICROBIT** для проверки.



2.10 Урок: Компас / Стороны света

Цель: Создать цифровой компас, который определяет, в какую сторону света (Север, Юг, Запад, Восток) повернута плата, и выводит соответствующую букву на экран.

⚠ Важное примечание: Калибровка

При первом запуске программы с использованием компаса micro:bit попросит вас провести калибровку.

1. На экране появится надпись «**TILT TO FILL SCREEN**» (Наклоняй, чтобы заполнить экран).
 2. Начните вращать плату во всех направлениях.
 3. Вам нужно заполнить все светодиоды на экране, пока не появится смайлик.
- Только после этого компас начнет работать.

Логика работы:

Компас выдает значение от **0 до 360 градусов**: 0° — это Север (N) / 90° — это Восток (E) / 180° — это Юг (S) / 270° — это Запад (W).

Мы должны разбить этот круг на 4 сектора и для каждого сектора показать свою букву.

Программный код (сборка блоков):

1. Блок «Постоянно» (Forever):

Вся программа строится внутри него.

2. Логическая конструкция (Logic):

- Зайдите в меню **Logic** (Логика) (бирюзовые блоки).
- Возьмите блок **if true then... else** (если истина, то... иначе).
- Нажмите на значок (+) внизу этого блока 3 раза, чтобы добавить условия **else if** (иначе если), пока не получится конструкция из 4-х частей, как на скриншоте.

3. Сборка условий:

Нам нужно проверять диапазон градусов. Для этого используются блоки сравнения из меню **Logic**: «И» (and), «Больше или равно» (\geq), «Меньше» ($<$). Значение градусов берем в меню **Input**: блок **compass heading** ($^\circ$).

Сектор 1 (Север / N):

- Условие: Если направление ≥ 0 И направление < 90 .
- Действие: **Show string "N"** (Показать строку "N").

Сектор 2 (Запад / W):

- Условие: Если направление ≥ 90 И направление < 180 .
- Действие: **Show string "W"**.

Сектор 3 (Юг / S):

- Условие: Если направление ≥ 180 И направление < 270 .
- Действие: **Show string "S"**.

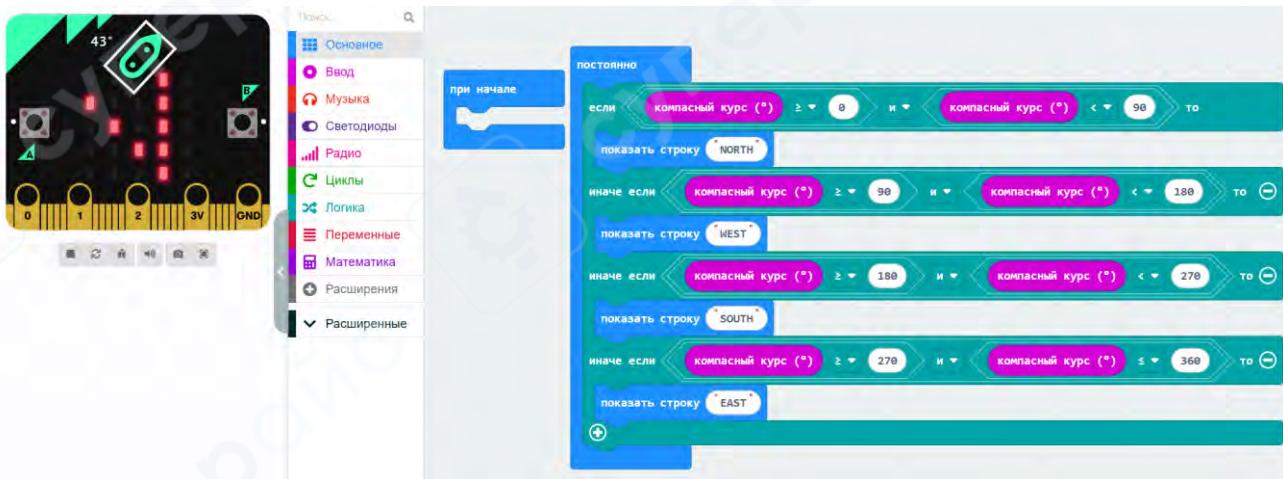
Сектор 4 (Восток / E):

- Условие: Если направление ≥ 270 И направление ≤ 360 .
- Действие: **Show string "E"**.

Файлы:

Файл урока: 东南西北.hex (East South West North).

Скопируйте его на диск **MICROBIT**. Не забудьте выполнить калибровку (вращение платы), если текст не появляется сразу.



Показывает север

2.11 Урок: Работа со звуком (Воспроизведение музыки)

Цель: Научиться подключать звуковой модуль (зуммер) и программировать micro:bit для воспроизведения мелодий.

1. Подготовка и подключение

Для воспроизведения звука нам понадобится **модуль пассивного зуммера (Buzzer)**.

- **Компоненты:** Плата micro:bit, плата расширения, модуль зуммера, соединительные провода.

• Схема подключения:

По умолчанию в среде MakeCode вывод звука настроен на пин P0.

1. Найдите на плате расширения разъем, соответствующий пину P0.
2. Подключите модуль зуммера проводами (соблюдая цветовую маркировку или обозначения):

S (Signal / Желтый) — к пину P0.

V (VCC / Красный) — к 3V (или VCC).

G (GND / Черный) — к GND (земля).

Примечание: Если у вас micro:bit версии V2 (с зубчатым логотипом спереди), у него есть встроенный динамик. Однако в обучающих наборах рекомендуется использовать внешний модуль для закрепления навыков схемотехники.

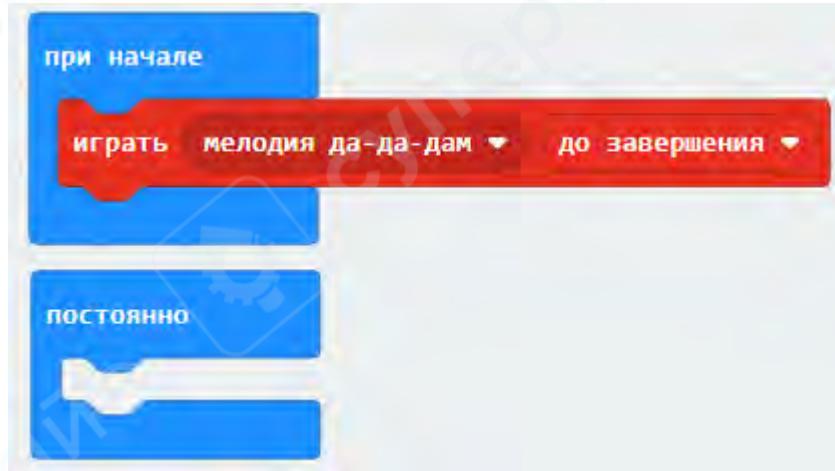
2. Программирование

Мы напишем программу, которая при запуске проигрывает мелодию.

1. Откройте **MakeCode**.
2. Откройте категорию **Music** (Музыка).
3. **Пример 1: Встроенная мелодия**
 - Найдите блок start melody ... repeating once (начать мелодию ... повторить один раз).
 - Перетащите его внутрь блока on start (при начале).
 - В выпадающем списке выберите мелодию.

Итоговый код (логика):

- **При включении:** Подается сигнал на пин P0, воспроизводящий заданную последовательность нот.



Файл урока: 课程 6-大家一起听音乐.hex.

Скопируйте его на диск **MICROBIT** для проверки

2.12 Урок: Электронная игральная кость (Датчик тряски)

Цель: Создать симулятор игральной кости (кубика), который показывает случайное число от 1 до 6 на экране при встряхивании платы. Изучить работу с переменными и логическими условиями.

Программирование

На скриншоте показана логика работы программы. Нам нужно создать переменную, которая будет хранить случайное число, и в зависимости от этого числа выводить картинку на экран.

Пошаговая инструкция по сборке кода (см. скриншот):

Событие «Тряска»:

В меню **Input** (Ввод) выберите блок **on shake** (при встряхивании). Все остальные блоки будем помещать внутрь него.

Создание переменной:

- Зайдите в меню **Variables** (Переменные), нажмите «**Make a Variable**» (создать переменную) и назовите **dice/кость**.

- Перетащите блок **set кость to ...** внутрь блока **on shake** (при встряхивании).

Генерация случайного числа:

- В меню **Math** (Математика) найдите блок **pick random 0 to 10**.
- Вставьте его в блок **set [кость] to**
- Измените диапазон: вместо 10 поставьте 5.

Пояснение: Мы выбираем числа от 0 до 5 (всего 6 вариантов), которые будут соответствовать граням кубика от 1 до 6.

Логика (Условные операторы):

- В меню **Logic** (Логика) возьмите блок **if <true> then ... else** (если истина, то... иначе).
 - Нам нужно расширить этот блок. Нажмайте значок (+) или шестеренку, чтобы добавить ветки **else if** (иначе если), пока у вас не получится цепочка для проверки всех чисел (как на скриншоте).

Настройка условий (Сравнение):

- В условие **if** вставьте блок сравнения из меню **Logic** ($0 = 0$).
- Замените первый ноль на вашу переменную **кость**.
- Заполните условия по порядку:

Если **кость = 0**

Иначе если **кость = 1**

Иначе если **кость = 2**

...и так далее до **5**.

Вывод изображения:

- Внутри каждого условия добавьте блок **show leds** (показать светодиоды) из меню **Basic**.
 - Нарисуйте точки игральной кости:
 - Для 0 -> нарисуйте **1 точку**.

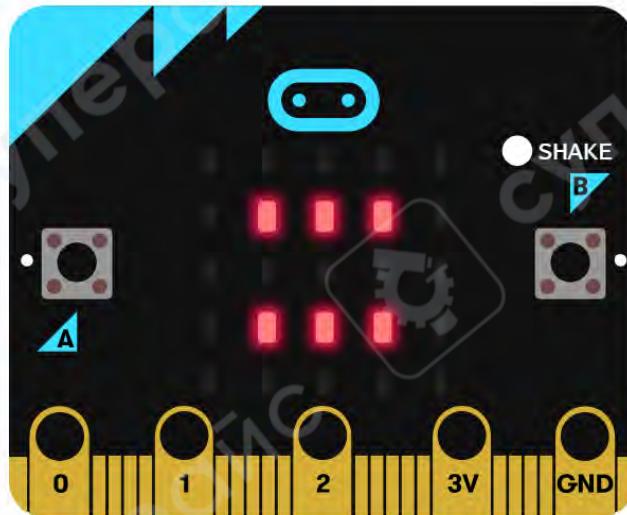
- Для 1 -> нарисуйте **2 точки**.
- Для 2 -> нарисуйте **3 точки**.
- Для 3 -> нарисуйте **4 точки**.
- Для 4 -> нарисуйте **5 точек**.
- Для 5 -> нарисуйте **6 точек**.

Загрузка и проверка результата

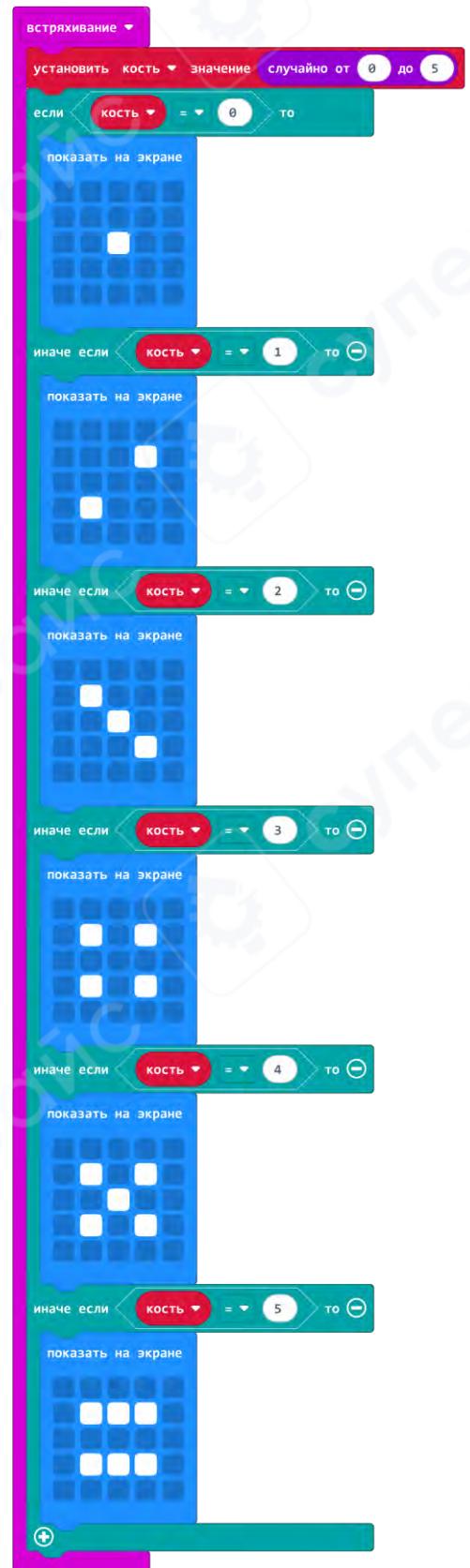
1. Скачайте .hex файл и запишите его на плату.
2. Отключите USB и подключите батарейки (чтобы плату было удобно трясти).
3. **Действие:** Резко встряхните плату micro:bit.
4. **Результат:** На экране должно появиться изображение точек (от 1 до 6). При каждом новом встряхивании число должно меняться случайным образом.

Файл урока: 摆骰子.hex.

Скопируйте его на диск MICROBIT для проверки



Выпала шестерка



2.13 Урок: День и Ночь (Датчик освещенности)

Цель: Создать умное устройство, которое автоматически определяет уровень освещенности и меняет картинку на экране: «Солнце» днем и «Луна» ночью.

Принцип работы:

У платы micro:bit нет отдельного видимого фоторезистора. Вместо этого в качестве датчика света используется сама **светодиодная матрица (экран)**.

Светодиоды могут не только излучать свет, но и улавливать его (работая в обратном режиме). Процессор измеряет уровень света в диапазоне от **0** (полная темнота) до **255** (очень ярко).

Программа будет проверять этот уровень: если он низкий — включать ночной режим, если высокий — дневной.

Программный код (сборка блоков):

Блок «При начале» (On Start):

Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

Все действия будут происходить здесь, чтобы проверка света шла непрерывно.

Шаг 1: Логическое условие

- Зайдите в меню **Logic** (Логика).
- Выберите блок **if <true> then ... else** (если <истина> то ... иначе).
- Перетащите его внутрь блока **Forever**.

Шаг 2: Сравнение значений

- Зайдите снова в **Logic** (Логика).
- Выберите блок сравнения **0 < 0** (или **0 = 0**).
- Вставьте его в поле **<true>** блока **if**.
- Поменяйте знак сравнения на **≤** (меньше или равно).
- В правое поле впишите число **20**. Это порог чувствительности (сумерки).

Шаг 3: Добавление датчика света

- Зайдите в меню **Input** (Ввод).
- Найдите овальный блок **light level** (уровень освещенности).
- Вставьте его в левое поле сравнения.
- Теперь условие звучит так: «Если уровень света ≤ 20 ...

Шаг 4: Действие «Ночь» (Если темно)

- В первую часть блока (где **then**) добавьте блок **show leds** (показать светодиоды) из меню **Basic** (Основное).

- Нарисуйте полумесяц (Луну), закрасив нужные квадратики (как на скриншоте).

Шаг 5: Действие «День» (Иначе)

- Во вторую часть блока (где **else / иначе**) добавьте блок **show icon** (показать значок).

- Выберите иконку **Солнца** (шахматная доска).

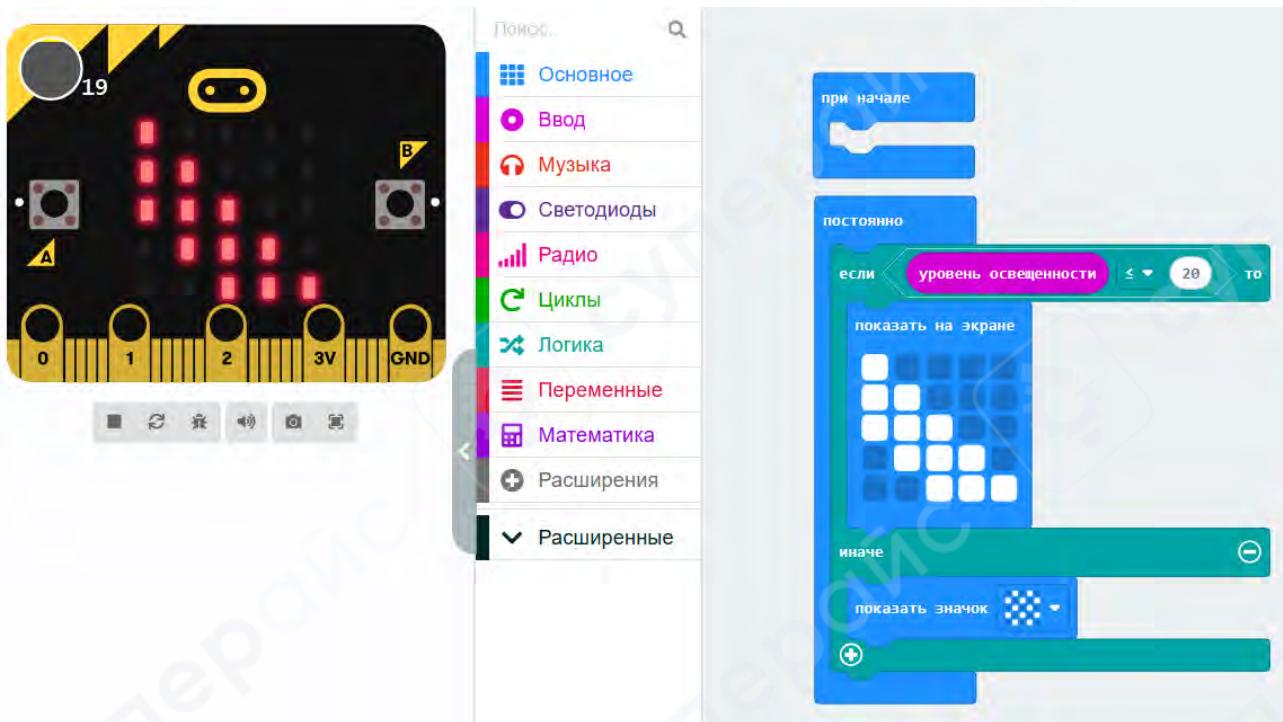
Результат:

1. Загрузите программу на плату.
2. В обычной комнате на экране должно гореть **Солнце**.
3. Закройте матрицу ладонью (создайте темноту). Изображение сменится на **Луну**.
4. Уберите руку — снова появится Солнце.

Файлы:

Файл урока: 白天黑夜.hex.

Скопируйте его на диск MICROBIT для проверки.



Матрица показывает ночь

2.14 Урок: Эксперимент с двухцветным светодиодом

Цель: Научиться управлять двухцветным (Red/Green) светодиодным модулем, используя цифровые и аналоговые сигналы для переключения цветов и их смешивания.

Принцип работы:

Модуль двухцветного светодиода содержит внутри корпуса два независимых светодиода (красный и зеленый) с общим катодом (общим контактом «минус»/GND).

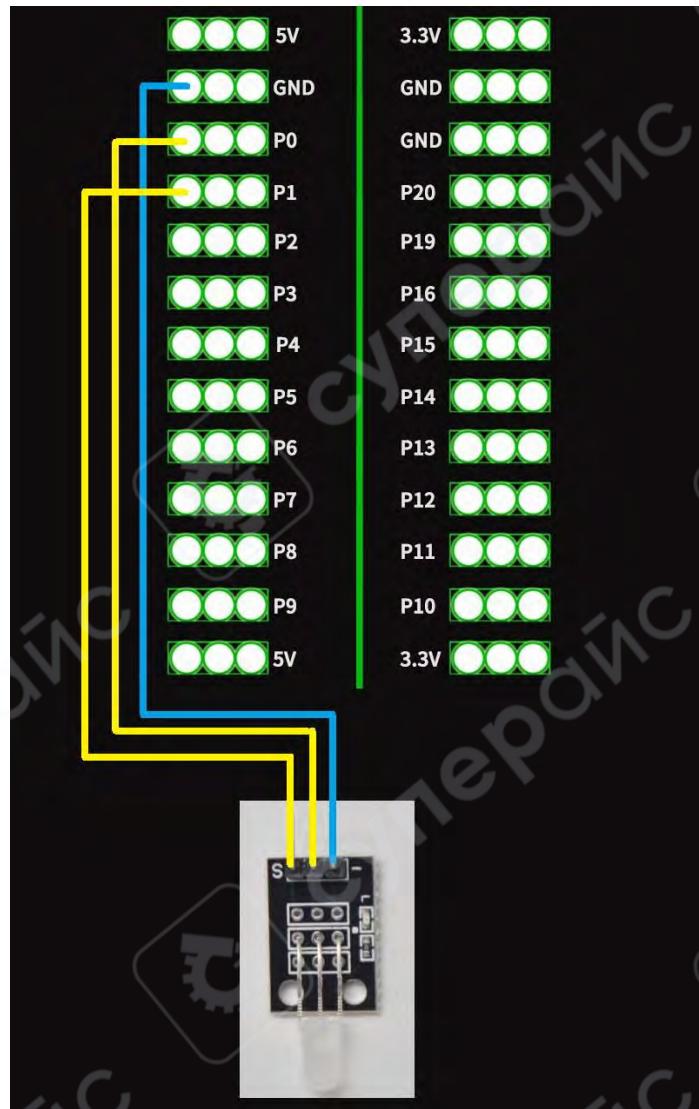
- Подавая напряжение на один контакт, мы включаем красный цвет.
- Подавая напряжение на второй — зеленый.
- Если подать напряжение на оба контакта одновременно, цвета смешиваются, и мы увидим желтый (или оранжевый) цвет.
- Используя **цифровой сигнал** (0 или 1), мы просто включаем/выключаем цвета.

Используя **аналоговый сигнал** (ШИМ/PWM от 0 до 1023), можно регулировать яркость каждого цвета отдельно.

Подключение оборудования:

Согласно схеме подключения, модуль соединяется с платой расширения следующим образом:

- GND (контакт «-» на модуле):** Подключите к контакту **GND** (синий провод).
- Средний контакт:** Подключите к порту **P0** (желтый провод).
- Крайний контакт «S»:** Подключите к порту **P1** (желтый провод).



Подключение светодиодного модуля (2 цвета)

Вариант 1: Цифровое управление (Переключение цветов)

В этом примере мы будем подавать сигналы высокого (1) и низкого (0) уровня, чтобы по очереди включать цвета.

Программный код (сборка блоков):

Блок «При начале» (On Start):

Оставьте пустым.

Блок «Постоянно» (Forever):

Программа выполняется циклически с паузами в 1 секунду.

1. Красный цвет:

Записать цифровой сигнал на **P0** значение **1**.

Записать цифровой сигнал на **P1** значение **0**.

Пауза 1000 мс.

2. Зеленый цвет:

Записать цифровой сигнал на **P0** значение **0**.

Записать цифровой сигнал на **P1** значение **1**.

Пауза 1000 мс.

3. Смешанный цвет (Желтый):

Записать цифровой сигнал на **P0** значение **1**.

Записать цифровой сигнал на **P1** значение **1**.

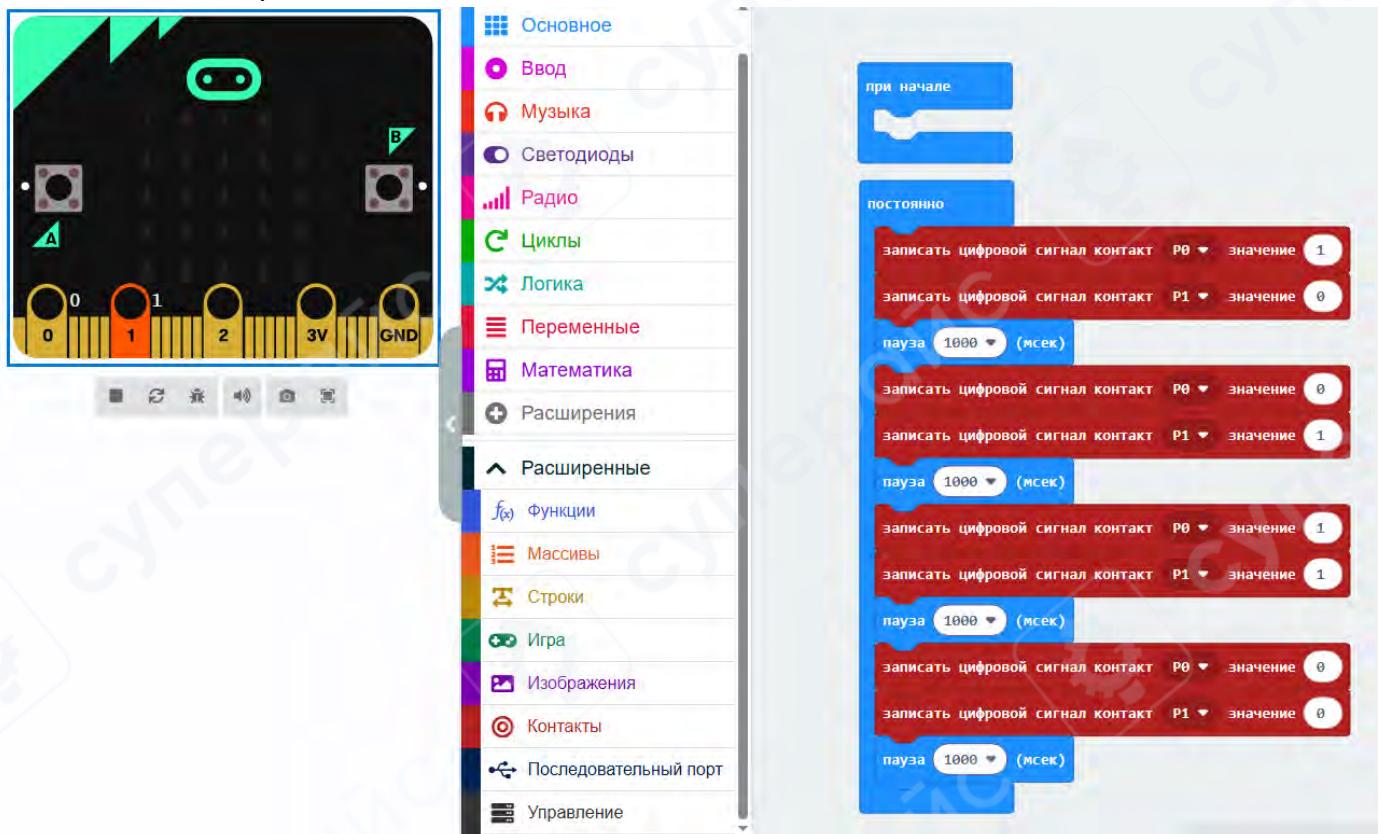
Пауза 1000 мс.

4. Выключение:

Записать цифровой сигнал на **P0** значение **0**.

Записать цифровой сигнал на **P1** значение **0**.

Пауза 1000 мс.



Вариант 2: Аналоговое управление (Регулировка яркости)

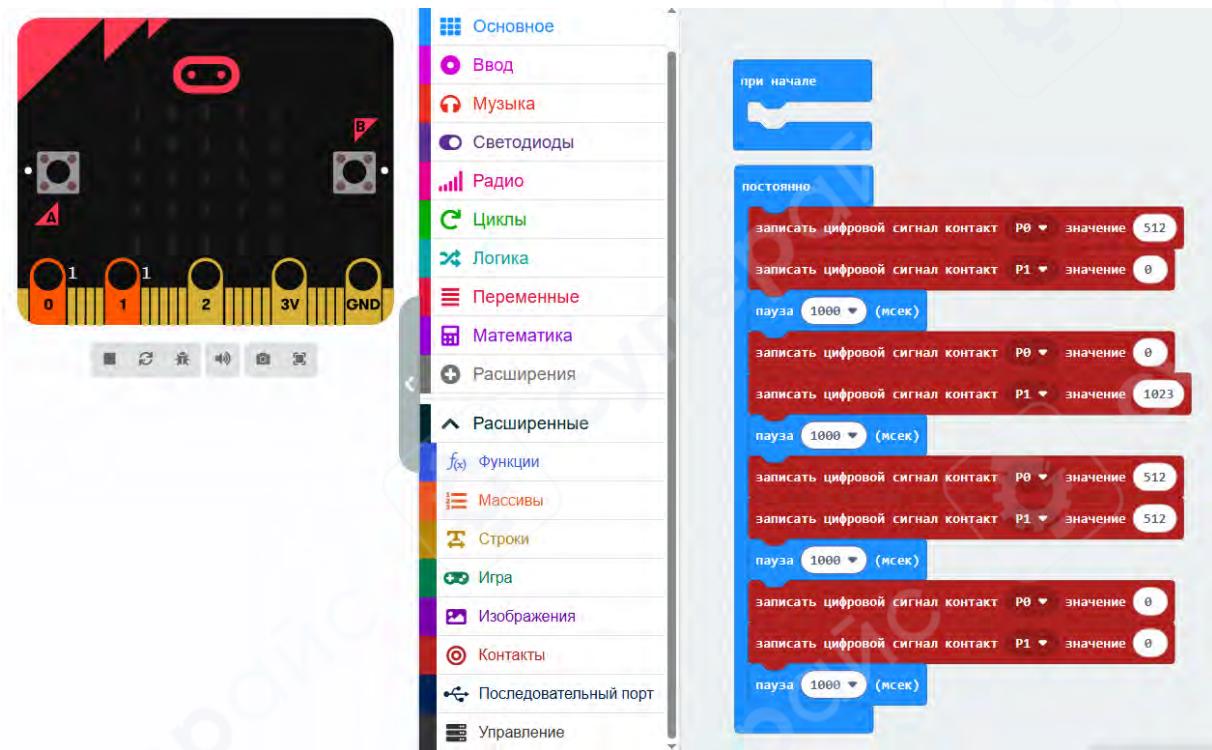
Во втором примере (второй скриншот кода) используются аналоговые сигналы.

Значение **1023** — это максимальная яркость, **512** — половина яркости, **0** — выключено.

Программный код (сборка блоков):

Блок «Постоянно» (Forever):

- P0 = 512, P1 = 0** (Один цвет на средней яркости).
- Пауза 1000 мс.
- P0 = 0, P1 = 1023** (Другой цвет на максимальной яркости).
- Пауза 1000 мс.
- P0 = 512, P1 = 512** (Оба цвета на средней яркости, смешивание).
- Пауза 1000 мс.
- P0 = 0, P1 = 0** (Все выключено).
- Пауза 1000 мс.



Результат:

После загрузки программы светодиод начнет менять цвета в заданной последовательности:

- Сначала загорится один цвет (например, красный).
- Затем второй цвет (например, зеленый).
- Затем загорятся оба, образуя желтый цвет.
- Светодиод погаснет на 1 секунду, и цикл повторится.

Файлы:

Файлы урока: 双色灯.hex. и 双色灯强度. hex

Скопируйте его на диск MICROBIT для проверки.

2.15 Урок: Эксперимент с RGB-светодиодом

Цель: Изучить принцип работы полноцветного RGB-светодиода и научиться смешивать три базовых цвета (Красный, Зеленый, Синий) для получения любых оттенков.

Принцип работы:

RGB-светодиод — это три маленьких светодиода (Red, Green, Blue) в одном корпусе.

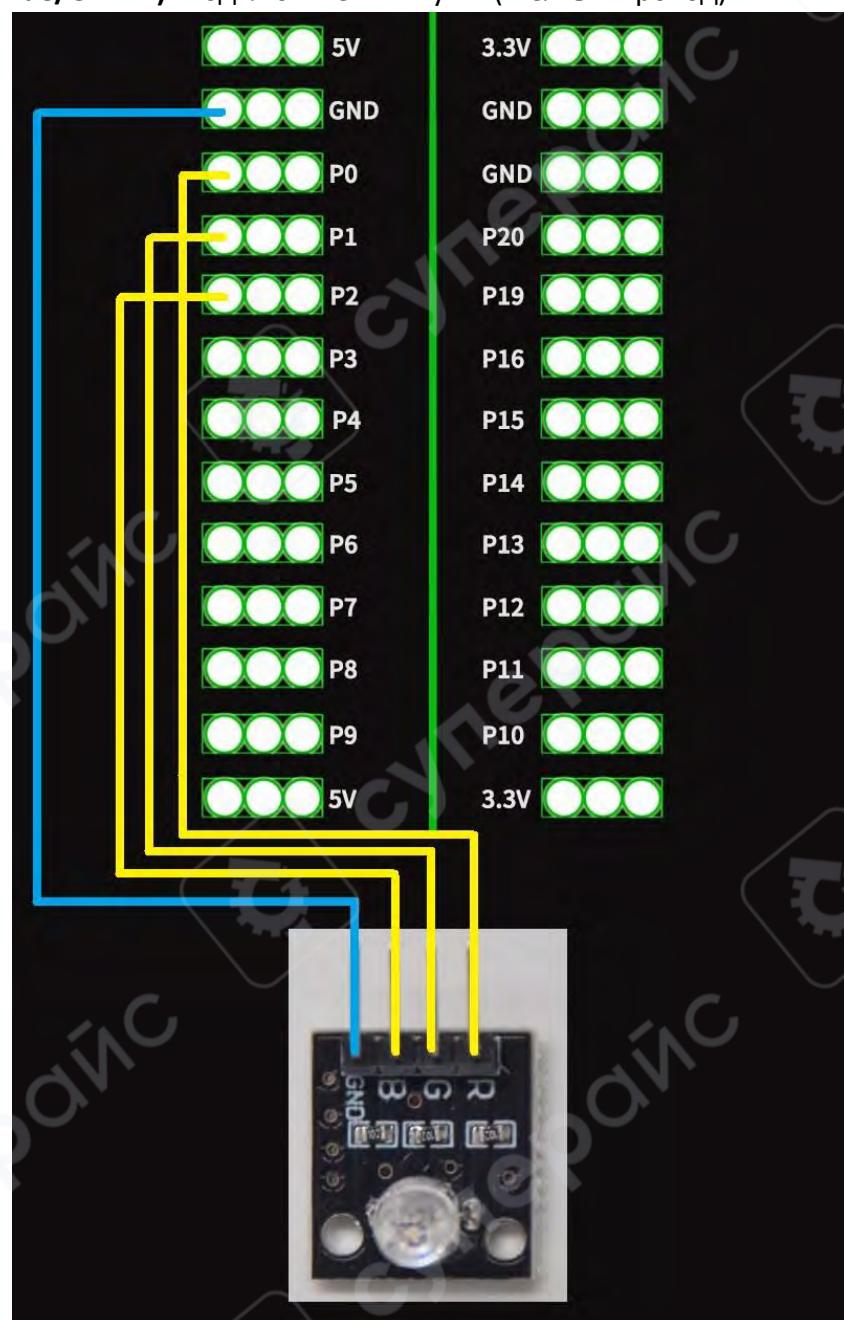
- Изменяя яркость каждого из них по отдельности, можно получить любой цвет спектра.
- В данном коде используется цветовая модель **0–255** (где 0 — выключено, 255 — максимальная яркость).
- Например: если включить Красный (255) и Зеленый (255) одновременно, глаз увидит **Желтый** цвет.

Подключение оборудования:

Внимание! Модуль имеет 4 контакта. Внимательно следите за маркировкой проводов на схеме.

- GND (или -):** Подключите к **GND** на плате (Синий провод).
- R (Red/Красный):** Подключите к пину **P0** (Желтый провод).

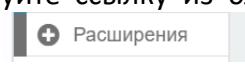
3. **G (Green/Зеленый):** Подключите к pinу **P1** (Желтый провод).
4. **B (Blue/Синий):** Подключите к pinу **P2** (Желтый провод).



Подключение RGB-светодиода

Программный код (сборка блоков):

Если вы создаете код с нуля и не можете найти этот блок, откройте файл урока *RGB.hex* — блоки появятся автоматически.

Из папки урока №15, скопируйте ссылку из блокнота «<https://github.com/zhuning239/cbit>». Найдите блок  Расширения, нажмите, в поле поиска введите скопированную ссылку. В этом уроке используется пользовательское расширение. Установите его.

Блок «Постоянно» (Forever):

1. Включаем Красный:

- Установить параметры: R=255, G=0, B=0.
- Пауза: 500 мс.

2. Включаем Зеленый:

- Установить параметры: R=0, G=255, B=0.
- Пауза: 500 мс.

3. Включаем Синий:

- Установить параметры: R=0, G=0, B=255.
- Пауза: 500 мс.

4. Смешивание: Желтый цвет:

- Включаем Красный и Зеленый вместе.
- Установить параметры: R=255, G=255, B=0.
- Пауза: 1000 мс.

5. Смешивание: Фиолетовый оттенок:

- Включаем Красный и немного Синего.
- Установить параметры: R=255, G=0, B=123.
- Пауза: 1000 мс.

6. Выключение:

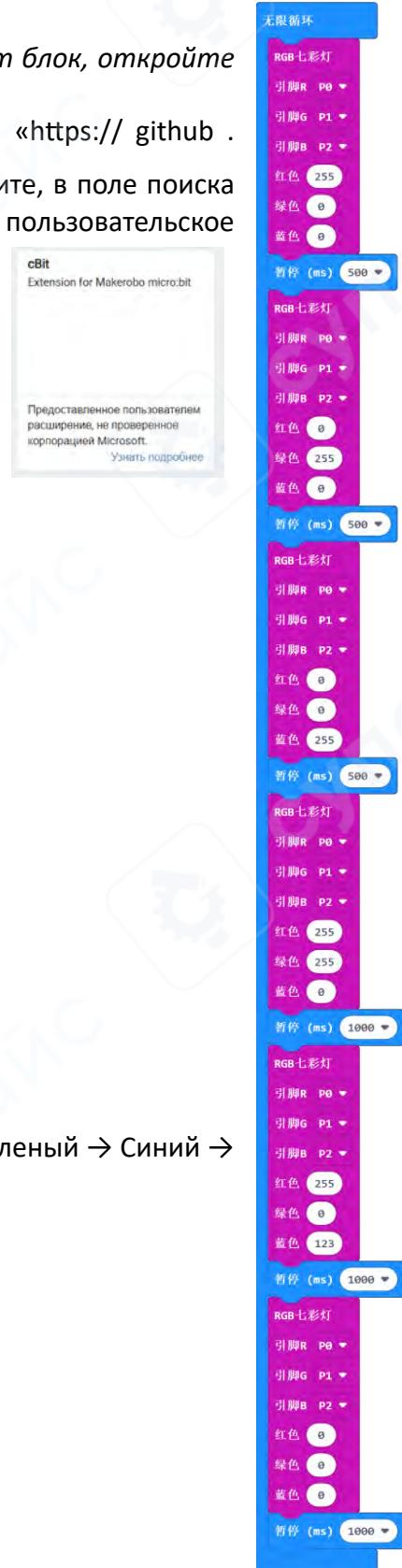
- Установить все цвета в 0.
- Пауза: 1000 мс.

Результат:

Светодиод будет циклически менять цвета: Красный → Зеленый → Синий → Желтый → Фиолетовый → (Темнота).

Файлы:

- Файл урока: *RGB.hex*.
- Скопируйте этот файл на диск **MICROBIT**.



2.16 Урок: Эксперимент с 7-цветным LED-светодиодом

Цель

Научиться управлять внешним светодиодом с помощью BBC micro:bit, используя цифровой вывод, и реализовать мигание светодиода с заданными временными интервалами.

Принцип работы

В эксперименте используется модуль 7-цветного светодиода (RGB LED), подключённый к плате расширения BBC micro:bit. Светодиод управляет цифровым сигналом, подаваемым на вывод P3 micro:bit.

Программа работает в бесконечном цикле:

- на вывод P3 подаётся логическая единица (1) — светодиод включается;
- выполняется задержка;
- на вывод P3 подаётся логический ноль (0) — светодиод выключается;
- выполняется повторная задержка.

Таким образом реализуется периодическое мигание светодиода.

Схема подключения

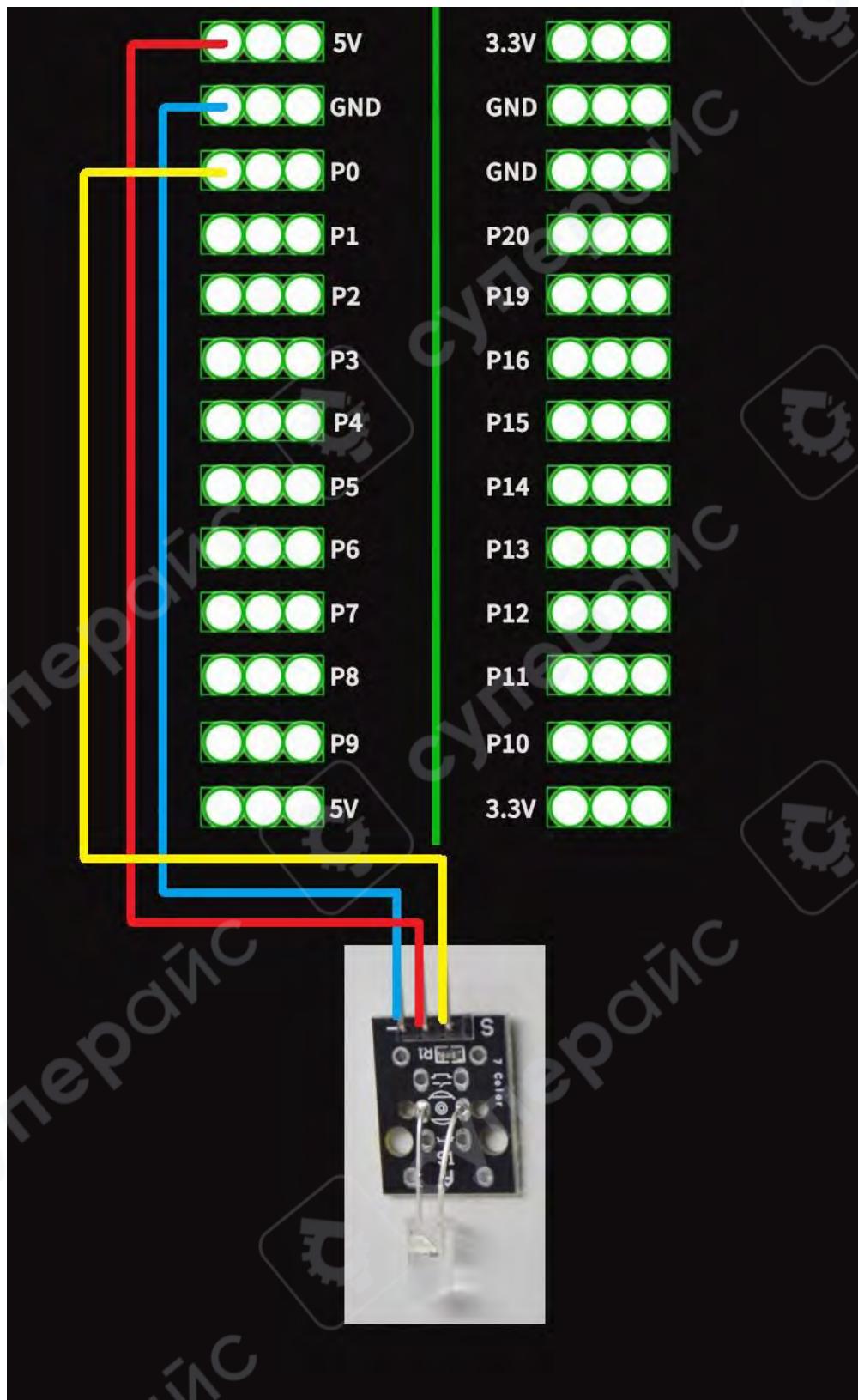
Используется плата расширения micro:bit GPIO.

Подключения модуля 7-цветного LED:

Контакт модуля Подключение

VCC	5V
GND	GND
S (Signal)	P0

⚠ Примечание: На модуле уже установлен токоограничивающий резистор (330 Ом), дополнительный резистор не требуется.



Подключение LED-светодиода

Программный код (сборка блоков)

Блок «При начале» (On Start)

- Установить состояние переменной/переключателя **LED Enabled** (включить экран) в значение **false** (ложь).

Назначение блока — инициализация состояния при запуске программы. В данном эксперименте дополнительная логика в блоке не используется.

Блок «Постоянно» (Forever)

Программа выполняется в бесконечном цикле.

Шаг 1: Включение светодиода

- Зайти в меню **Pins (Контакты)**.
- Выбрать блок **digital write pin P3 to 1** (Записать цифровой сигнал Р3 на 1).
- Установить значение **1** (логическая единица).

→ Светодиод включается.

Шаг 2: Пауза

- Зайти в меню **Basic (Основное)**.
- Добавить блок **pause (ms)**.
- Установить значение **2000 ms**.

→ Светодиод остаётся включённым в течение 2 секунд.

Шаг 3: Выключение светодиода

- Использовать блок **digital write pin P3 to 0** (Записать цифровой сигнал Р3 на 0).
- Установить значение **0** (логический ноль).

→ Светодиод выключается.

Шаг 4: Пауза

- Добавить блок **pause (ms)**.
- Установить значение **1000 ms**.

→ Светодиод остаётся выключенным в течение 1 секунды.

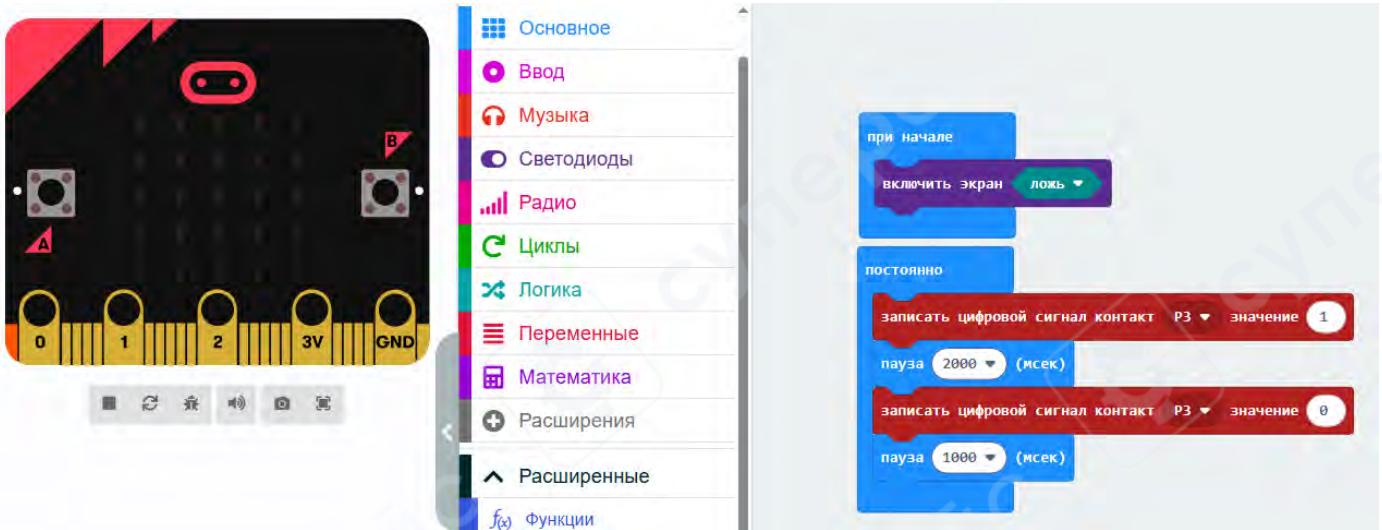
Результат

После загрузки программы в BBC micro:bit: 7-цветный светодиод загорается на 2 секунды; затем гаснет на 1 секунду; цикл повторяется непрерывно. Из-за особенностей модуля при подаче питания могут быть видны разные оттенки свечения.

Примечание

- Если изменить значения задержек, можно регулировать частоту мигания.
- При изменении пина в программе необходимо переподключить сигнальный провод.
 - Для сложных эффектов (переливы цветов) требуется использование аналогового управления (PWM) или нескольких выводов.

Файлы Файл урока: **7 色 LED 灯实验.hex**. Скопируйте файл на диск **MICROBIT** для загрузки и проверки работы программы.



2.17 Урок: Эксперимент с релейным модулем

Цель

Изучить принципы управления релейным модулем с помощью BBC micro:bit и научиться включать и выключать внешнюю нагрузку с использованием цифрового сигнала.

Принцип работы

Релейный модуль представляет собой электромеханическое устройство, позволяющее управлять цепями с более высоким напряжением или током с помощью низковольтного управляющего сигнала от micro:bit.

В данном эксперименте:

- вывод P0 micro:bit используется как управляющий сигнал (**SIG**);
- при подаче логической единицы (1) реле срабатывает (контакт замыкается);
- при подаче логического нуля (0) реле отключается (контакт размыкается).

Внутри модуля используется транзисторный ключ, токоограничивающие резисторы и защитный диод, предотвращающий повреждение схемы при отключении катушки реле.

Схема подключения

Используется релейный модуль с входами **SIG / VCC / GND**.

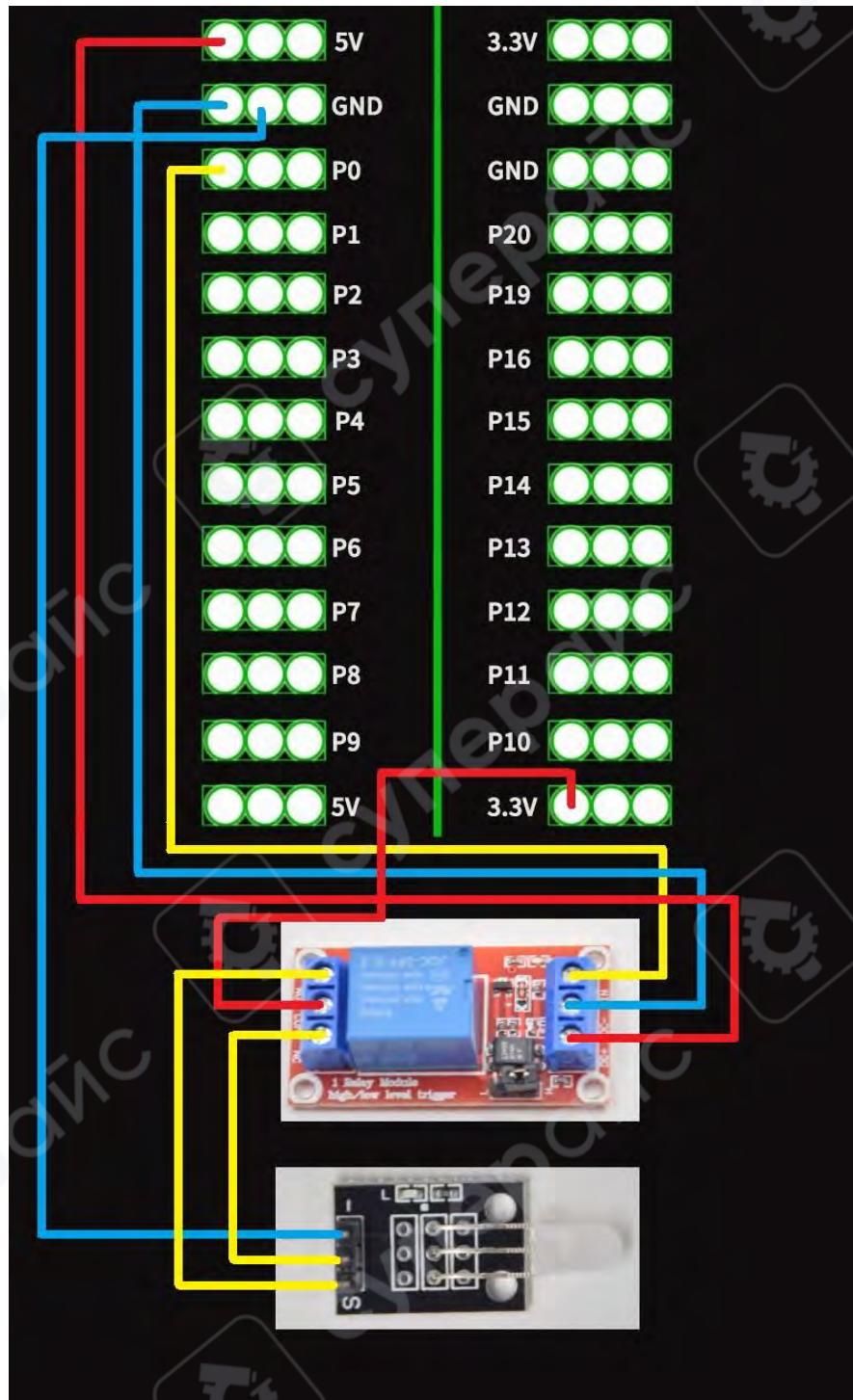
Подключение к плате расширения BBC micro:bit:

Контакт релейного модуля Подключение

SIG	P0
VCC	3.3V
GND	GND

⚠ Важно:

- Используется питание **3.3 В**, совместимое с уровнем логики micro:bit.
- Подключение внешней нагрузки (клеммы COM / NO / NC) в данном уроке демонстрационное и может отличаться в зависимости от задачи.



Подключение релейного модуля

Программный код (сборка блоков)

Блок «При начале» (On Start)

Блок не содержит команд. Используется стандартная точка инициализации программы.

Блок «Постоянно» (Forever)

Программа выполняется в бесконечном цикле.

Шаг 1: Включение реле

- Перейти в меню Pins (Выводы).

- Использовать блок **digital write pin P0 to 1** (Записать цифровой сигнал P0 на 1)
→ Реле срабатывает, управляющий контакт замыкается.

Шаг 2: Пауза

- Перейти в меню **Basic (Основное)**.
 - Добавить блок **pause (ms)**.
 - Установить значение **2000 ms**.
- Реле остаётся включённым в течение 2 секунд.

Шаг 3: Выключение реле

- Использовать блок **digital write pin P0 to 0** (Записать цифровой сигнал P0 на 0)
→ Реле отключается, контакт размыкается.

Шаг 4: Пауза

Добавить блок **pause (ms)**.

- Установить значение **2000 ms**.

→ Реле остаётся выключенным в течение 2 секунд.

Результат

После загрузки программы в BBC micro:bit: релейный модуль включается на 2 секунды;

- затем выключается на 2 секунды; цикл повторяется непрерывно.

При срабатывании реле может быть слышен характерный щелчок — это нормальная работа электромеханического реле.

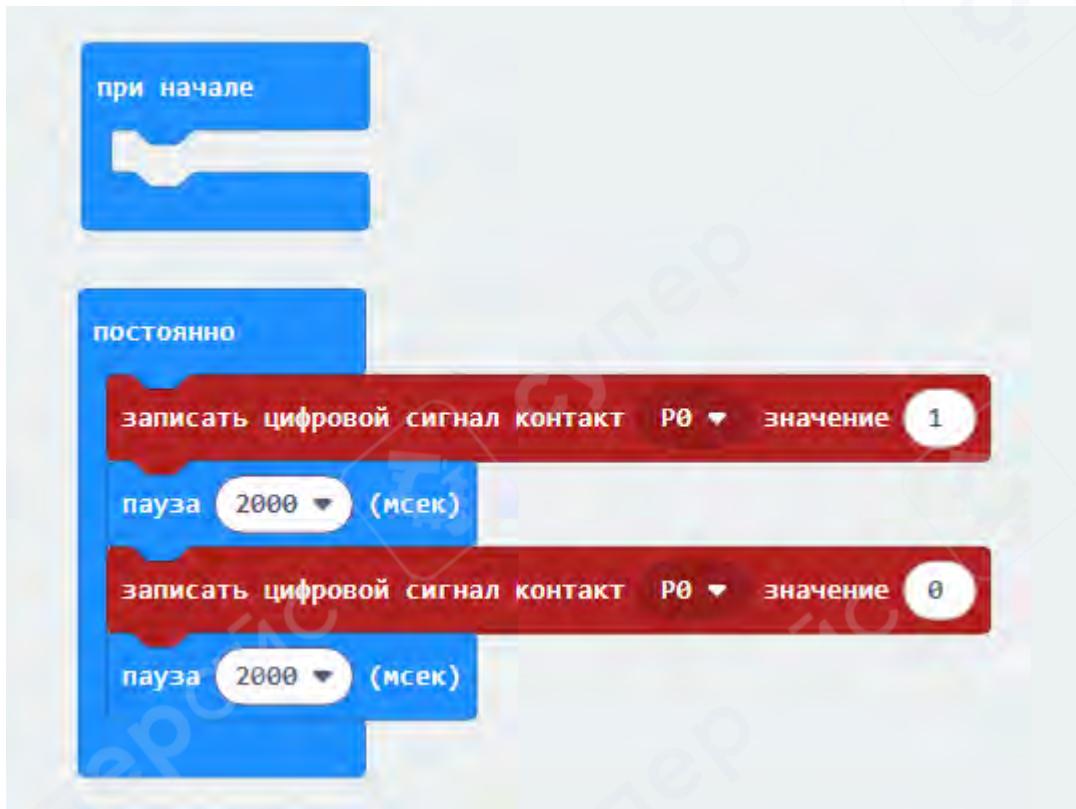
Примечания и меры безопасности

- Не подключайте к реле сетевое напряжение без соответствующей подготовки и изоляции.
- Перед изменением схемы отключайте питание.
- Если реле не срабатывает:
 - проверьте правильность подключения **SIG → P0**;
 - убедитесь, что используется цифровая запись, а не аналоговая;
 - проверьте подачу питания 3.3V и GND.

Файлы

Файл урока: **继电器模块实验.hex**

Скопируйте файл на диск **MICROBIT** для загрузки и проверки работы программы.



2.18 Урок: Управление тремя LED-светодиодами

Цель

Научиться управлять несколькими внешними светодиодами с помощью BBC micro:bit, используя разные цифровые выводы, и реализовать их последовательное включение и выключение.

Принцип работы

В эксперименте используются три отдельных LED-модуля (красный, синий и зелёный), каждый из которых подключён к своему цифровому выводу micro:bit через плату расширения.

Программа выполняется в бесконечном цикле и последовательно:

1. включает первый светодиод;
2. выключает его;
3. включает второй светодиод;
4. выключает его;
5. включает третий светодиод;
6. выключает его.

Каждое состояние удерживается заданное время с помощью блока задержки.

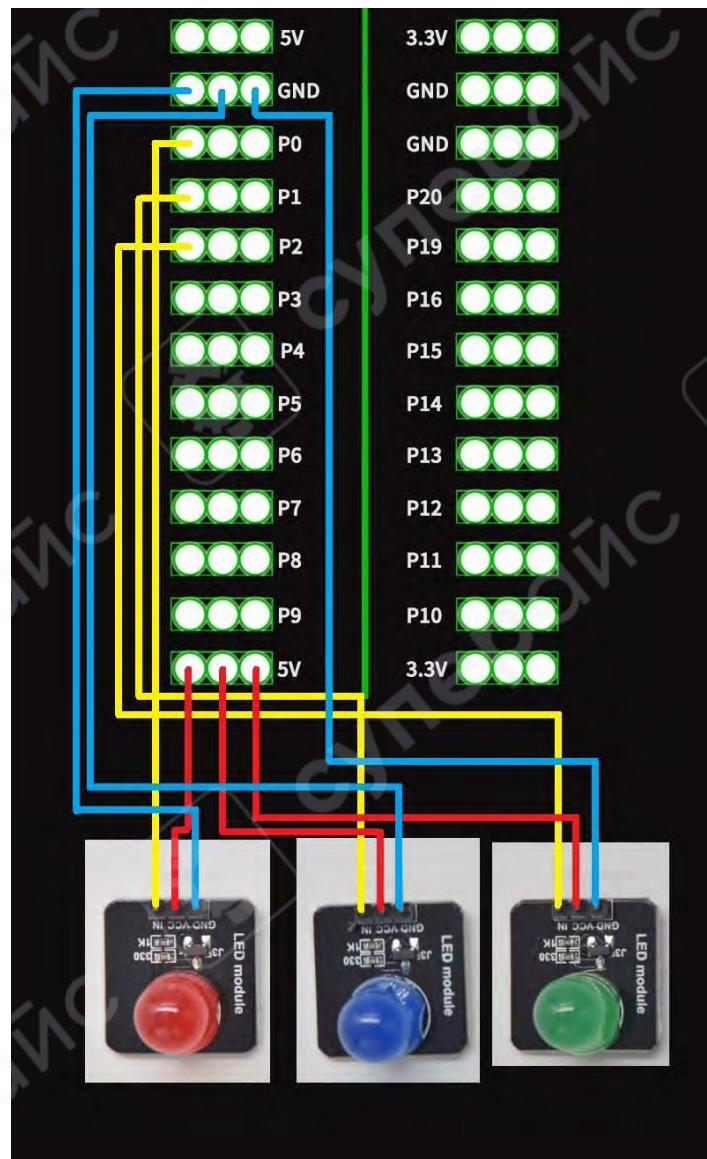
Схема подключения

Каждый LED-модуль имеет три контакта: **VCC**, **GND**, **SIG**. В модулях уже установлен токоограничивающий резистор, дополнительный резистор не требуется.

Подключение к плате расширения BBC micro:bit:

Светодиод	Контакт модуля	Подключение
	SIG	P0
LED 1 (красный)	VCC	5V
	GND	GND
	SIG	P1
LED 2 (синий)	VCC	5V
	GND	GND
	SIG	P2
LED 3 (зелёный)	VCC	5V
	GND	GND

⚠ Примечание: Все модули питаются от общей линии 5V и GND.



Подключение LED-светодиодов

Программный код (сборка блоков)

Блок «При начале» (On Start)

Блок не содержит команд. Используется стандартная инициализация программы.

Блок «Постоянно» (Forever)

Программа выполняется в бесконечном цикле.

Шаг 1: Включение первого светодиода (P0)

- Перейти в меню **Pins (Выходы)**.
- Использовать блок **digital write pin P0 to 1** (Записать цифровой сигнал P0 на 1).
- Добавить блок **pause (ms)** со значением **1000 ms**.
- Использовать блок **digital write pin P0 to 0**.

→ Первый светодиод загорается на 1 секунду и затем гаснет.

Шаг 2: Включение второго светодиода (P1)

- Использовать блок **digital write pin P1 to 1** (Записать цифровой сигнал P1 на 1).
- Добавить блок **pause (ms)** со значением **1000 ms**.
- Использовать блок **digital write pin P1 to 0** (Записать цифровой сигнал P1 на 0).

→ Второй светодиод загорается на 1 секунду и затем гаснет.

Шаг 3: Включение третьего светодиода (P2)

- Использовать блок **digital write pin P2 to 1** (Записать цифровой сигнал P2 на 1).
- Добавить блок **pause (ms)** со значением **1000 ms**.
- Использовать блок **digital write pin P2 to 0** (Записать цифровой сигнал P2 на 0).

→ Третий светодиод загорается на 1 секунду и затем гаснет.

Результат

После загрузки программы в BBC micro:bit:

- светодиоды загораются **по очереди**;
- каждый светодиод светится **1 секунду**;
- после выключения третьего светодиода цикл начинается заново.

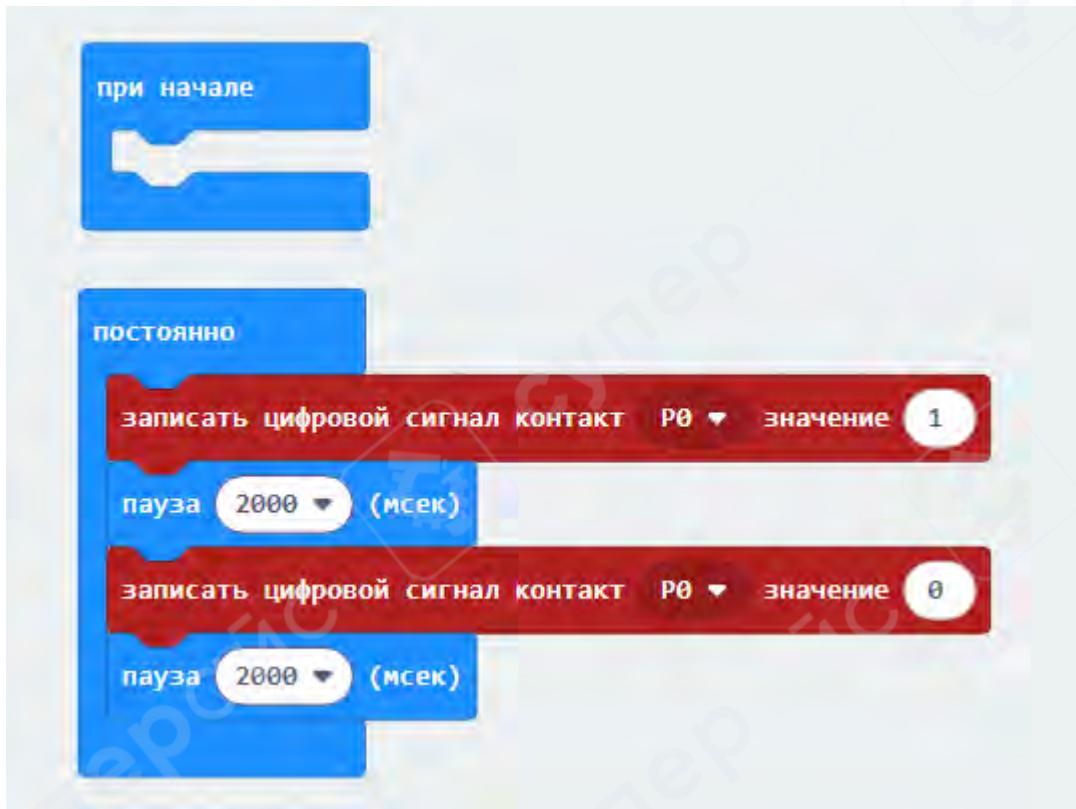
Примечания

- Порядок включения светодиодов можно изменить, переставив блоки в программе.
- Время свечения регулируется изменением значения в блоке **pause (ms)**.
- При изменении используемых выводов необходимо скорректировать и схему подключения.

Файлы

Файл урока: **三个 LED 灯控制.hex**

Скопируйте файл на диск **MICROBIT** для загрузки и проверки работы программы.



2.19 Урок: Эксперимент с лазерным модулем

Цель

Научиться управлять лазерным модулем с помощью BBC micro:bit, используя цифровой вывод, и реализовать периодическое включение и выключение лазера.

Принцип работы

Лазерный модуль представляет собой источник лазерного излучения, управляемый подачей питания и цифрового сигнала. В данном эксперименте управление осуществляется через цифровой вывод P0 micro:bit.

Программа работает в бесконечном цикле:

- на вывод P0 подаётся логическая единица (1) — лазер включается;
- выполняется пауза;
- на вывод P0 подаётся логический ноль (0) — лазер выключается;
- выполняется повторная пауза.

Таким образом реализуется мигание лазерного луча с фиксированным интервалом.

Схема подключения

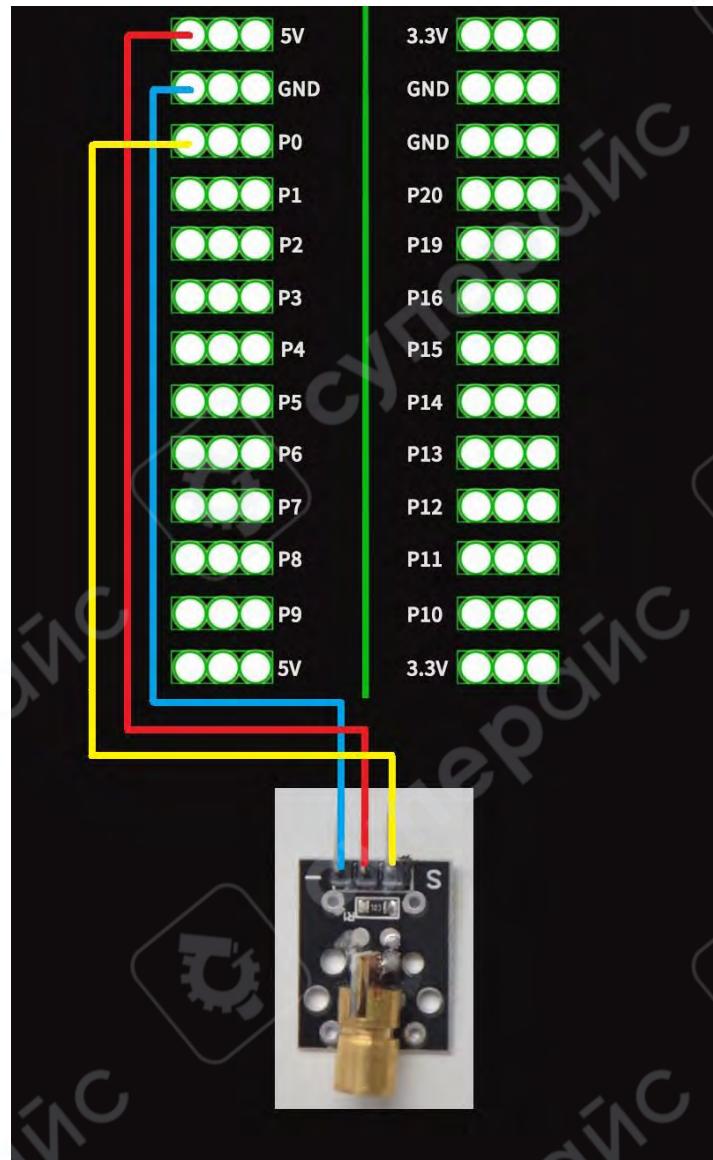
Лазерный модуль имеет три контакта: VCC, GND, SIG.

Подключение к плате расширения BBC micro:bit:

Контакт лазерного модуля Подключение

SIG	P0
VCC	5V
GND	GND

⚠ Примечание: В модуле установлен токоограничивающий резистор, поэтому дополнительный резистор не требуется.



Подключение лазерного модуля

Программный код (сборка блоков)

Блок «При начале» (On Start)

Блок не содержит команд. Используется стандартная инициализация программы.

Блок «Постоянно» (Forever)

Программа выполняется в бесконечном цикле.

Шаг 1: Включение лазера

- Перейти в меню **Pins (Выходы)**.
 - Использовать блок **digital write pin P0 to 1** (Записать цифровой сигнал P0 на 1)
- Лазерный модуль включается, появляется лазерный луч.

Шаг 2: Пауза

- Перейти в меню **Basic (Основное)**.
 - Добавить блок **pause (ms)**.
 - Установить значение **500 ms**.
- Лазер остаётся включённым в течение 0,5 секунды.

Шаг 3: Выключение лазера

- Использовать блок **digital write pin P0 to 0** (Записать цифровой сигнал P0 на 0)
→ Лазерный модуль выключается.

Шаг 4: Пауза

- Добавить блок **pause (ms)**.
- Установить значение **500 ms**.
→ Лазер остаётся выключенным в течение 0,5 секунды.

Результат

После загрузки программы в BBC micro:bit:

- лазерный луч включается на 0,5 секунды;
- затем выключается на 0,5 секунды;
- процесс повторяется непрерывно.

Примечания и меры безопасности

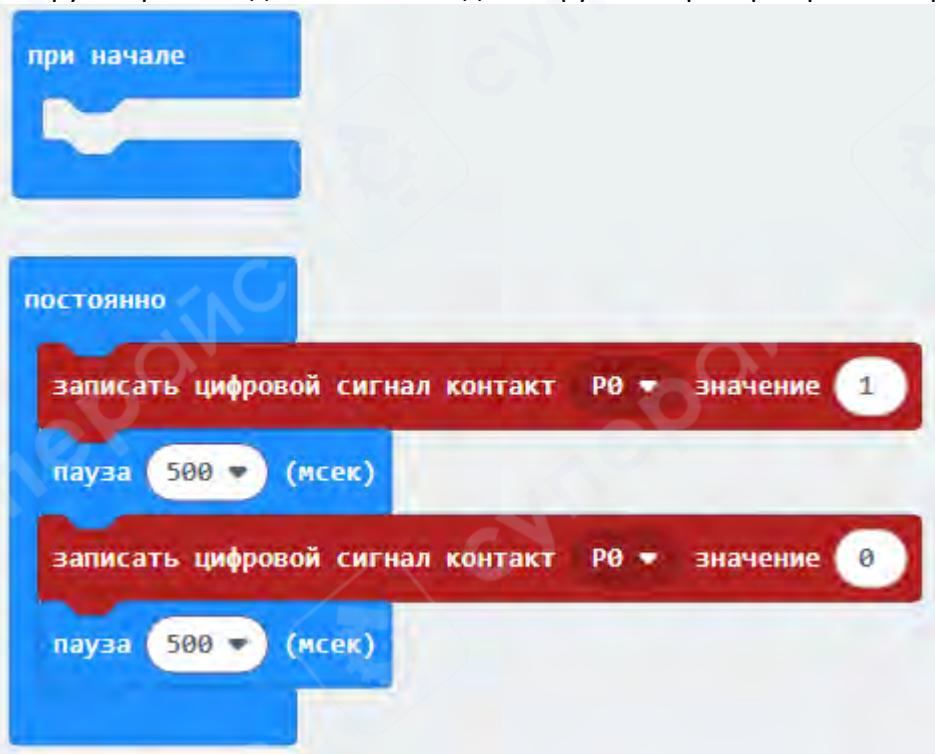
⚠ Внимание:

- Не направляйте лазерный луч в глаза людям или животным.
- Не используйте лазерный модуль вблизи отражающих поверхностей.
- При изменении пина управления необходимо изменить и схему подключения.

Файлы

Файл урока: **激光模块实验.hex**

Скопируйте файл на диск **MICROBIT** для загрузки и проверки работы программы.



2.20 Урок: Эксперимент с тактильной кнопкой

Цель

Научиться считывать состояние внешней кнопки с помощью BBC micro:bit и управлять светодиодами в зависимости от нажатия кнопки.

Принцип работы

В эксперименте используется модуль тактильной кнопки, подключённый к цифровому входу micro:bit. Кнопка формирует логический сигнал на выводе **P3**:

- при нажатии кнопки — сигнал «нажато»;
- при отпускании — сигнал «не нажато».

Программа в бесконечном цикле проверяет состояние кнопки:

- если кнопка нажата, включается светодиод на выводе **P0**, а светодиод на **P1** выключается;
- если кнопка не нажата, светодиод на **P0** выключается, а на **P1** включается.

Таким образом реализуется простая логика управления по условию.

Схема подключения

Используются:

- модуль кнопки;
- LED-модуль (2 цвета).

Подключение кнопочного модуля

Контакт модуля кнопки Подключение

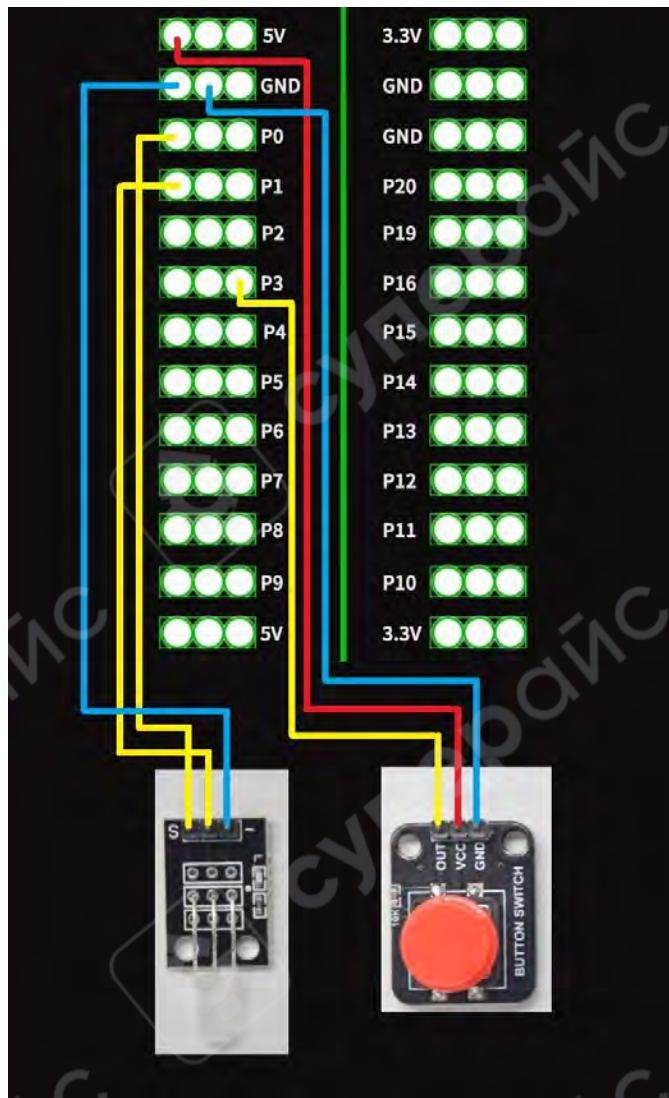
SIG	P3
VCC	5V
GND	GND

Подключение светодиодов

Устройство Контакт Подключение

	SIG	P0
LED	VCC	P1
	GND	GND

⚠ Примечание: Во всех используемых модулях уже присутствуют токоограничивающие резисторы, установка дополнительных резисторов не требуется.



Подключение тактильной кнопки и LED модуля

Программный код (сборка блоков)

Блок «При начале» (On Start)

Установить логическую переменную **LED Enabled** в значение false.

Блок используется для начальной инициализации состояния при запуске программы.

В этом уроке используется пользовательское расширение Cbit_输入类.

Блок «Постоянно» (Forever)

Программа выполняется в бесконечном цикле и постоянно опрашивает состояние кнопки.

Шаг 1: Проверка нажатия кнопки

- Перейти в меню **Logic** (Логика).
- Использовать блок **if / else** (если / иначе).

В условии выбрать: **button pin P3 returns pressed**

Шаг 2: Действия при нажатой кнопке (если)

- Использовать блок **digital write pin P0 to 1** (Записать цифровой сигнал P0 на 1).
- Использовать блок **digital write pin P1 to 0** (Записать цифровой сигнал P1 на 0).

→ Первый светодиод включается, второй выключается.

Шаг 3: Действия при ненажатой кнопке (иначе)

- Использовать блок **digital write pin P0 to 0** (Записать цифровой сигнал P0 на 0).
- Использовать блок **digital write pin P1 to 1** (Записать цифровой сигнал P1 на 0).

→ Первый светодиод выключается, второй включается.

Результат

После загрузки программы в BBC micro:bit:

- при нажатии кнопки:
 - светодиод на **P0** загорается;
 - светодиод на **P1** гаснет;
- при отпускании кнопки:
 - светодиод на **P0** гаснет;
 - светодиод на **P1** загорается.

Состояние обновляется в реальном времени.

Примечания

- Если светодиоды работают наоборот, проверьте:
 - правильность подключения выводов **P0, P1, P3**;
 - полярность подключения LED-модулей.
- Логику работы можно изменить, поменяв местами команды в ветках **если / иначе**.
- Для устранения дребезга контактов в сложных проектах может потребоваться программная задержка.

Файлы

Файл урока: **轻触按键实验.hex**

Скопируйте файл на диск **MICROBIT** для загрузки и проверки работы программы.



2.21 Урок: Эксперимент с датчиком наклона

Цель:

Создать устройство, которое определяет положение платы в пространстве (наклон) и сигнализирует об изменении состояния переключением цветов внешнего светодиодного модуля.

Принцип работы:

- Датчик наклона:** Внутри датчика (золотой цилиндр) находится металлический шарик. Когда датчик расположен вертикально, шарик замыкает контакты (цепь замкнута, сигнал 1). При наклоне шарик смещается, размыкая контакты (цепь разомкнута, сигнал 0).
- Двухцветный LED-модуль:** Модуль подключается к двум цифровым пинам. Подавая напряжение на один пин и отключая другой, мы меняем цвет свечения.
- Micro:bit:** Считывает цифровой сигнал с датчика наклона и управляет светодиодным модулем. **Важно:** Поскольку для считывания используется пин P4, необходимо отключить встроенный светодиодный экран micro:bit, так как он использует те же линии связи и может создавать помехи.

Компоненты:

- Плата Micro:bit.
- Плата расширения GPIO.
- Модуль датчика наклона (золотистый цилиндр).

4. Модуль двухцветного светодиода (3 контакта).
5. Соединительные провода.

Схема подключения:

- **Датчик наклона:**

Сигнальный вывод (S) → Пин P4

VCC (Питание) → 5V

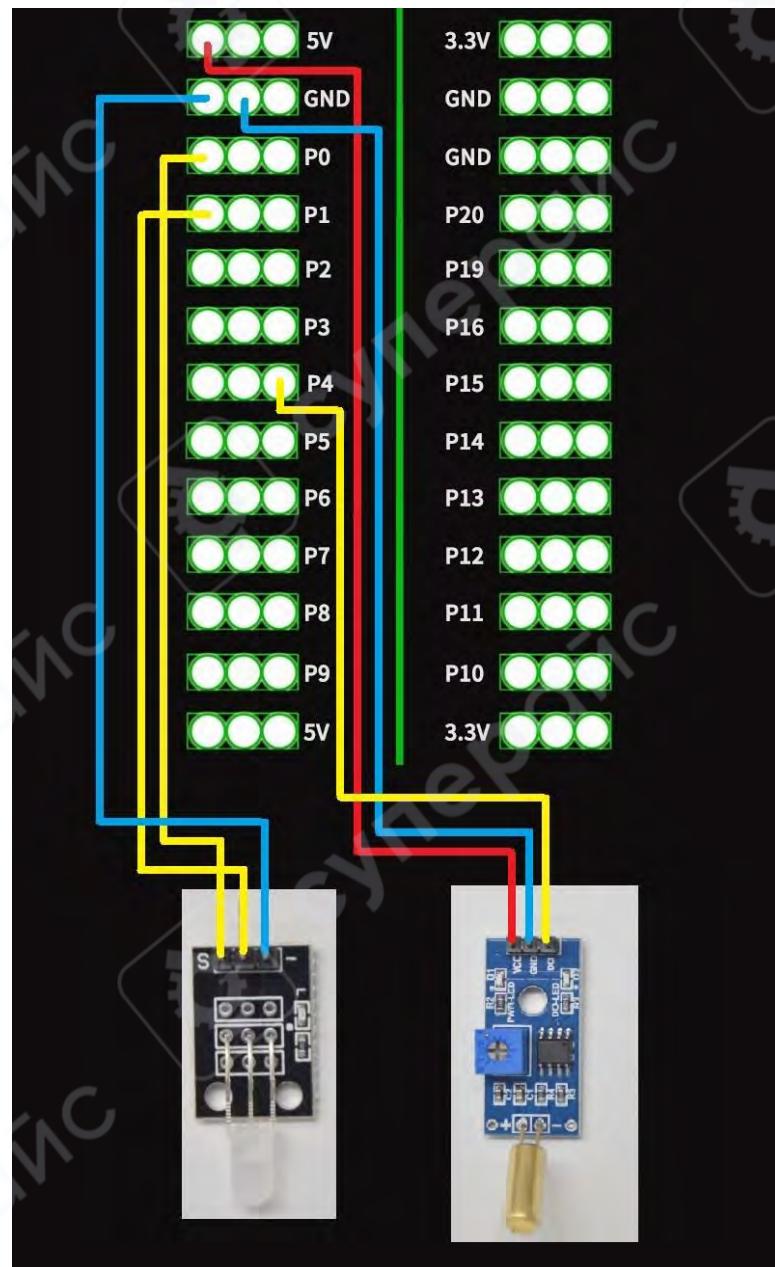
GND (Земля) → GND

- **LED-модуль:**

Сигнальный вывод 1 (L или S) → Пин P0

Сигнальный вывод 2 (R или -) → Пин P1

GND (Общий минус) → GND



Подключение датчика наклона и светодиода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Этот блок настраивает плату перед запуском основного цикла.
- Найдите категорию **LED** (Светодиоды).
- Выберите блок **led enable (включить экран)** и переключите его в состояние **false (ложь)**.

Пояснение: Это действие отключает экранную матрицу 5x5, освобождая pin P4 для работы с внешним датчиком.

Блок «Постоянно» (Forever):

- Внутри размещается логическая конструкция для непрерывной проверки датчика.

Шаг 1: Логическое условие

- Зайдите в меню **Logic (Логика)**.
- Выберите блок **if ... then ... else (если ... то ... иначе)**.
- В условие (шестиугольное поле после «если») вставьте блок сравнения **0 = 0**.

Шаг 2: Считывание датчика

- В левую часть сравнения вставьте блок из меню **Pins (Пины/Выводы)**: **digital read pin P4 (цифровое чтение пин P4)**.

- В правую часть впишите число **1**.
- *Итоговое условие: «Если цифровое чтение пин P4 равно 1».*

Шаг 3: Управление светодиодом (состояние 1)

- В раздел **then (то)** добавьте блоки управления пинами из меню **Pins**:
- **digital write pin P0 to 0 (цифровая запись пин P0 значение 0)**
- **digital write pin P1 to 1 (цифровая запись пин P1 значение 1)**
- *Это включит первый цвет светодиода.*

Шаг 4: Управление светодиодом (состояние 2)

- В раздел **else (иначе)** добавьте блоки:
- **digital write pin P0 to 1 (цифровая запись пин P0 значение 1)**
- **digital write pin P1 to 0 (цифровая запись пин P1 значение 0)**

Это переключит светодиод на второй цвет, когда датчик наклонен (сигнал 0).

Результат:

После загрузки программы:

1. Экран самого micro:bit погаснет (так как мы его отключили).
2. Внешний светодиодный модуль загорится одним цветом.
3. Если вы наклоните плату или потрясете датчик наклона (слышен стук шарика внутри), светодиод изменит цвет на другой.

Примечание:

Датчики наклона шарикового типа очень чувствительны к вибрации. При тряске цвета могут переключаться очень быстро (мерцать), так как шарик внутри хаотично размыкает и замыкает контакты. Это нормальное поведение для данного типа датчиков.

Файлы: Файл урока: 倾斜传感器.hex



2.22 Урок: Эксперимент с герконовым датчиком

Цель: Создать устройство, реагирующее на магнитное поле. При поднесении магнита к датчику цвет внешнего светодиода будет изменяться.

Принцип работы:

- **Геркон (Герметичный контакт):** Это датчик, состоящий из двух упругих ферромагнитных пластин, запаянных в стеклянную колбу. В обычном состоянии контакты разомкнуты. При воздействии внешнего магнитного поля (например, при поднесении магнита) контакты намагничиваются, притягиваются друг к другу и замыкают электрическую цепь.

- **Логика сигнала:**

Магнит рядом: Контакты замкнуты -> На выходе датчика низкий уровень напряжения (Логический 0).

Магнита нет: Контакты разомкнуты -> На выходе датчика высокий уровень напряжения (Логическая 1).

Micro:bit: Считывает состояние пина P6 и в зависимости от полученного сигнала (0 или 1) переключает полярность на светодиодном модуле, меняя его цвет.

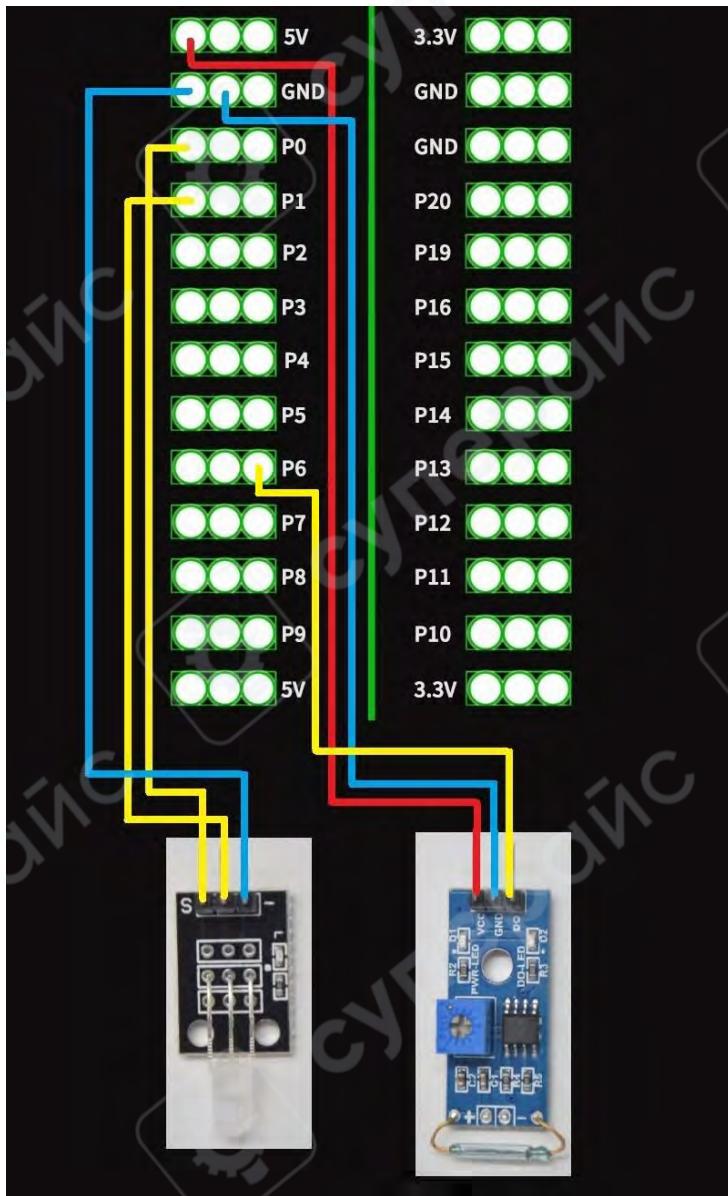
Компоненты:

1. Плата Micro:bit.
2. Плата расширения GPIO.
3. Модуль с герконом (Reed Switch Module) — синяя плата со стеклянной колбой.
4. Модуль двухцветного светодиода (LED Module) — черная плата со светодиодом.
5. Магнит (для активации датчика).
6. Соединительные провода.

Схема подключения:

Внимание! Строго соблюдайте подключение проводов согласно списку ниже. Ошибка в полярности может повредить датчики.

- **Модуль геркона (Reed Switch):**
 - Вывод D0 (или S) → Пин P6 (Желтый провод на схеме).
 - Вывод VCC (Питание) → 5V (Красный провод).
 - Вывод GND (Земля) → GND (Синий провод).
- **LED-модуль (Светодиод):**
 - Сигнальный вывод 1 (L или S) → Пин P0
 - Сигнальный вывод 2 (R или -) → Пин P1
 - GND (Общий минус) → GND



Подключение герконового датчика и светодиода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Для корректной работы внешних пинов рекомендуется отключить встроенный экран.
- Зайдите в категорию **LED** (Светодиоды).

- Выберите блок **led enable** (**включить светодиоды**) и установите значение **false** (**ложь**).

Блок «Постоянно» (Forever):

- Программа должна непрерывно проверять наличие магнита.

Шаг 1: Проверка условия (Магнит обнаружен)

- Возьмите блок **if ... then ... else** (**если ... то ... иначе**) из меню **Logic** (**Логика**).
- В условие вставьте сравнение: **digital read pin P6** (**цифровое чтение пин P6**) = **0**.

Пояснение: Почему 0? Потому что при замыкании геркона сигнал уходит в "землю" (GND). Значение 0 означает "Магнит есть".

Шаг 2: Действие при наличии магнита (To)

- Если магнит поднесен (условие истинно), включаем первый цвет:
- **digital write pin P0 to 1** (**цифровая запись пин P0 значение 1**)
- **digital write pin P1 to 0** (**цифровая запись пин P1 значение 0**)

Шаг 3: Действие при отсутствии магнита (Иначе)

- В раздел **else** (**иначе**) добавьте блоки для второго цвета:
- **digital write pin P0 to 0** (**цифровая запись пин P0 значение 0**)
- **digital write pin P1 to 1** (**цифровая запись пин P1 значение 1**)

Результат:

1. Загрузите код в micro:bit.
2. В состоянии покоя светодиод будет гореть одним цветом (например, красным).
3. Поднесите магнит к стеклянной колбе на модуле датчика.
4. Вы услышите тихий щелчок внутри колбы (замыкание контактов), и светодиод мгновенно сменит цвет (например, на зеленый).
5. Уберите магнит — цвет вернется к исходному.

Примечание:

Стеклянная колба геркона очень хрупкая. Будьте осторожны при монтаже, чтобы не раздавить её и не погнуть выводы у самого стекла.

Файлы:

Файл урока: Reed_Switch_Test.hex.



2.23 Урок: Эксперимент с активным зуммером

Цель:

Научиться управлять активным зуммером (пищалкой) и создать программу, которая заставляет его издавать прерывистый звуковой сигнал (ритмичный писк).

Принцип работы:

- Активный зуммер:** Внутри этого модуля уже встроен генератор звуковой частоты. Ему не нужно подавать сложный звуковой сигнал (волну). Достаточно просто подать питание (высокий логический уровень), и он начнет пищать.
- Управление:** Мы будем просто включать и выключать подачу напряжения на сигнальный pin с определенной задержкой.
- Сигнал 1 (High/Вкл)** — зуммер пищит.
- Сигнал 0 (Low/Выкл)** — зуммер молчит.

Компоненты:

- Плата Micro:bit.
- Плата расширения GPIO.
- Модуль активного зуммера (черная плата с круглым черным цилиндром, часто с наклейкой сверху).
- Соединительные провода.

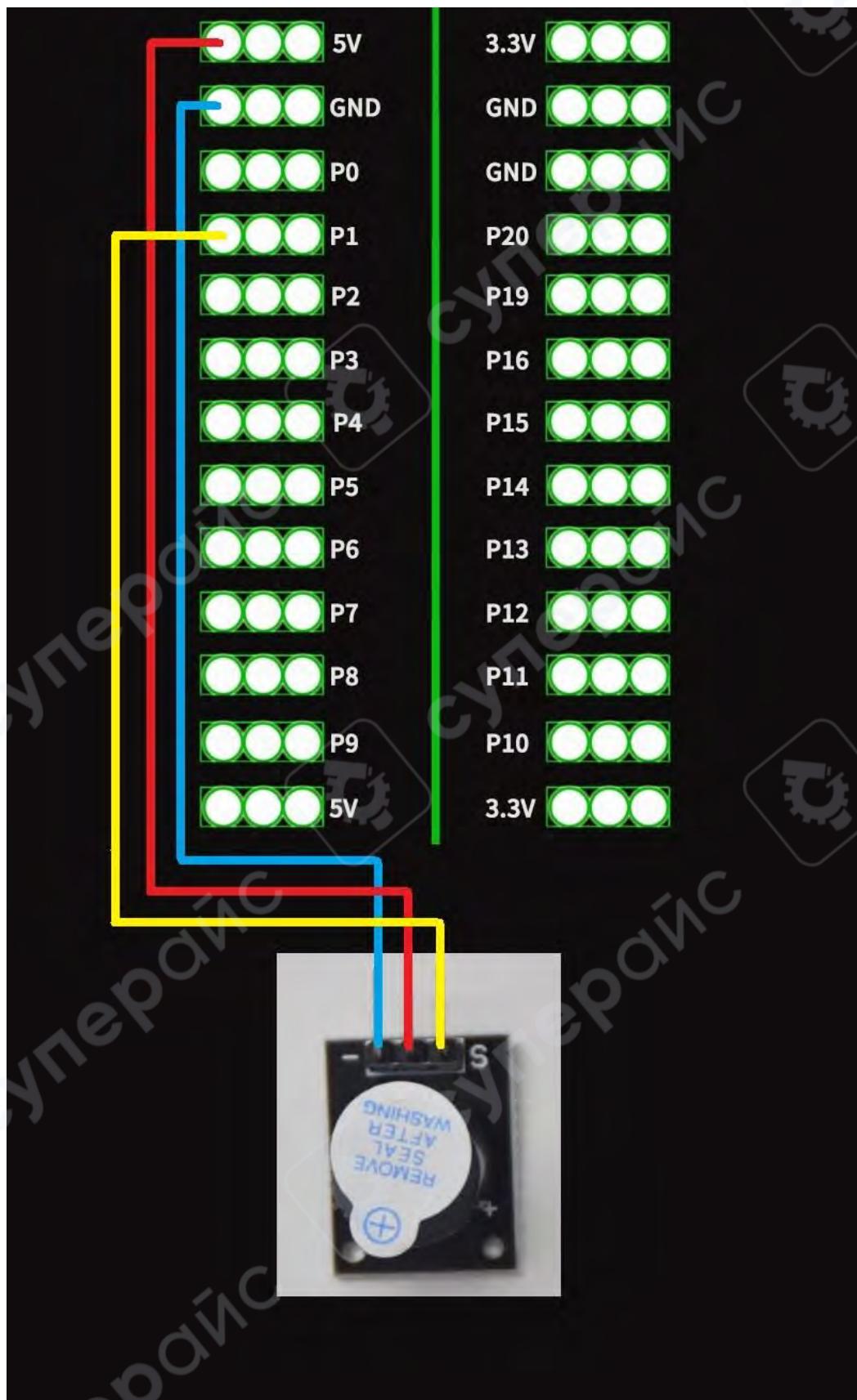
Схема подключения:

- Модуль зуммера:**

Вывод S (Сигнал) → Пин P1 (Желтый провод).

Вывод + (или средний контакт, VCC) → 5V (Красный провод).

Вывод - (GND) → GND (Синий провод).



Подключение активного зуммера

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

- В этом цикле мы создадим ритм.

Шаг 1: Включение звука

- Используйте блок управления активным зуммером (из специального расширения для вашего набора, коричневого цвета).

- Параметры блока: **Active Buzzer Pin** (Пин активного зуммера) выберите P1, значение **Ring** (Звенеть/Вкл).

Шаг 2: Пауза (длительность звука)

- Из меню **Basic (Основное)** добавьте блок **pause (ms)** (пауза мс).
- Установите значение **200**.

Шаг 3: Выключение звука

- Добавьте второй блок управления зуммером.
- Параметры: **Active Buzzer Pin** выберите P1, значение **No Ring** (Не звенеть/Выкл).

Шаг 4: Пауза (длительность тишины)

- Добавьте блок **pause (ms)** (пауза мс).
- Установите значение **200**.

Результат: После загрузки программы зуммер начнет издавать громкий прерывистый писк: «Бип-бип-бип» с интервалом в 200 миллисекунд.

Примечание:

1. **Наклейка:** На новых зуммерах часто наклеена белая защитная наклейка с надписью «REMOVE SEAL AFTER WASHING». **Обязательно удалите её**, чтобы звук был громким и чистым.

2. **Отличие от пассивного зуммера:** Активный зуммер может издавать звук только одной тональности (частоты). Вы не сможете сыграть на нем мелодию по нотам, только ритм. Для мелодий используется пассивный зуммер.

Файлы: Файл урока: 有源蜂鸣器.hex.



2.24 Урок: Эксперимент с пассивным зуммером

Цель:

Подключить внешний пассивный зуммер к плате расширения и запрограммировать micro:bit на воспроизведение мелодии «Happy Birthday».

Принцип работы:

В данном уроке используется **пассивный зуммер**. В отличие от активного зуммера, который издает звук при простой подаче питания, пассивному зуммеру для работы требуется внешний сигнал с разной частотой (PWM/ШИМ). Это позволяет ему воспроизводить мелодии и ноты, а не просто пищать.

Мы подключим его к контакту **P0**. Программа будет отключать встроенный динамик micro:bit (если используется версия V2), чтобы звук гарантированно шел на внешний модуль, и в цикле проигрывать заданную мелодию.

Схема подключения:

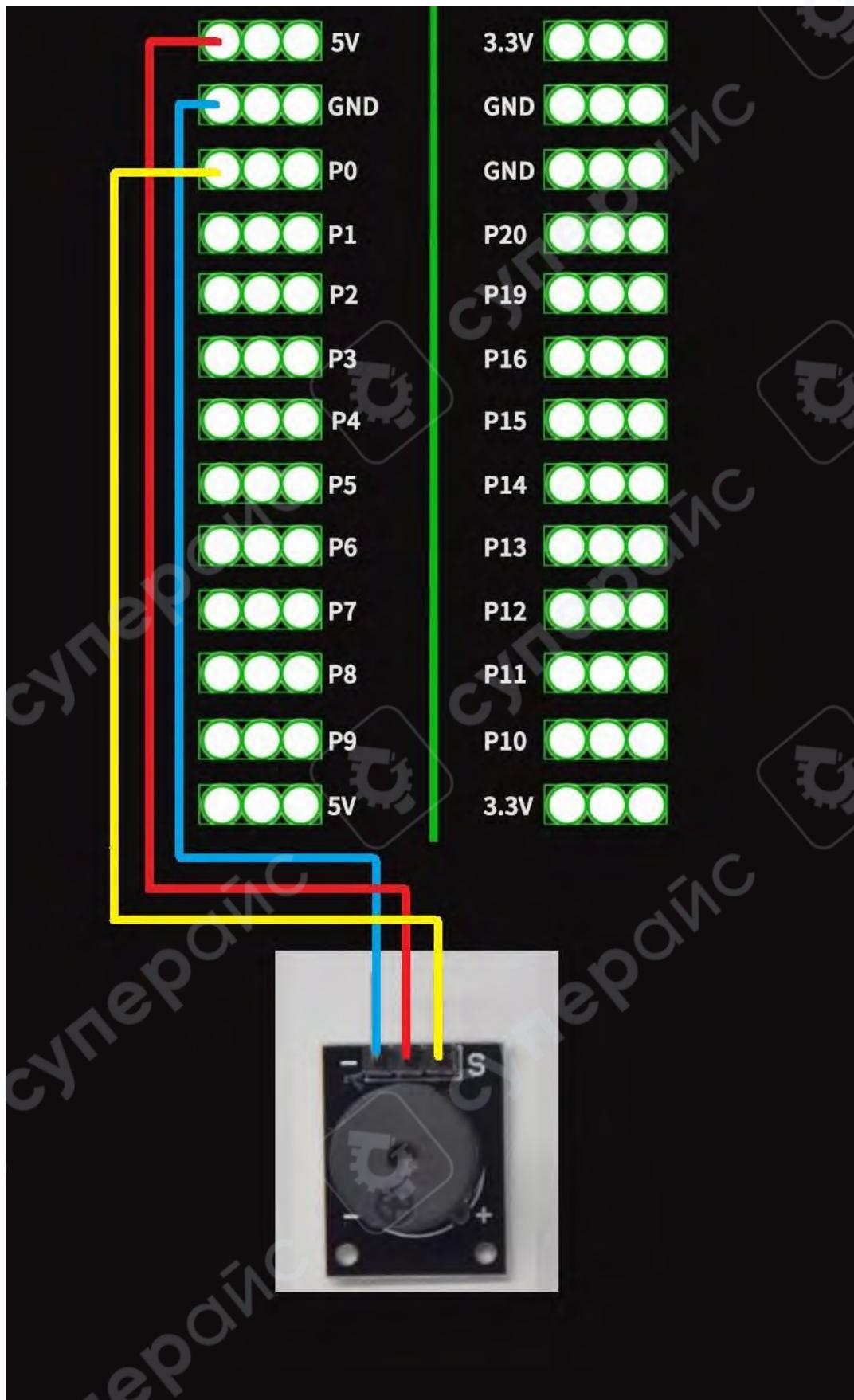
Для удобства используется плата расширения GPIO (которую мы рассматривали ранее). См. [приложенную схему](#).

- **Модуль зуммера (Buzzer):**

Контакт **S** (Signal/Сигнал) — подключается к pinu **P0** (Желтый провод).

Контакт **Центральный/+** (VCC/Питание) — подключается к **5V** (Красный провод).

Контакт **-** (GND/Земля) — подключается к **GND** (Синий провод).



Подключение пассивного зуммера

Программный код (сборка блоков):

Блок «При начале» (On Start):

- В категории **Music (Музыка)** выберите блок **set built-in speaker [off]** (установить встроенный динамик [выкл]).

Зачем это нужно: Это отключает динамик на самой плате micro:bit (актуально для версии V2), чтобы убедиться, что мы слышим звук именно от нашего внешнего модуля.

Блок «Постоянно» (Forever):

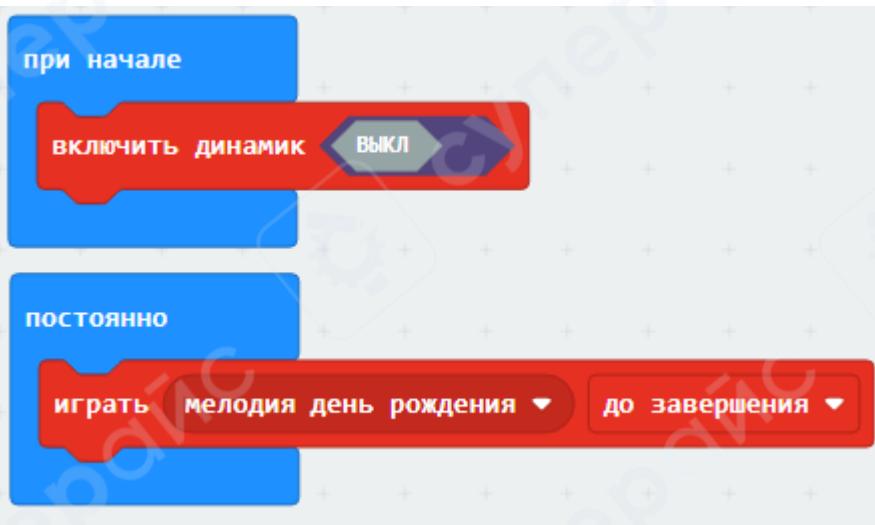
- В категории **Music (Музыка)** найдите блок **воспроизведения мелодии**.
- Выберите блок **play melody [Birthday] until done** (играть мелодию [День рождения] до завершения).
- В выпадающем списке выберите мелодию **Birthday** (День рождения).
- Режим воспроизведения: **until done** (до конца/до завершения), чтобы мелодия доиграла полностью перед повторным запуском цикла.

Результат:

После загрузки программы внешний зуммер начнет непрерывно проигрывать мелодию «Happy Birthday». Как только мелодия закончится, она начнется заново.

Файлы:

Файл урока: 无源蜂鸣器实验.hex . Скопируйте его на диск MICROBIT.



2.25 Урок: Эксперимент с U-образным фотоэлектрическим датчиком

Цель:

Создать систему, которая включает внешний светодиод, когда какой-либо объект перекрывает прорезь в датчике.

Принцип работы:

В этом уроке используется **U-образный фотоэлектрический датчик** (также называемый щелевым датчиком или фотопрерывателем).

1. **Как работает датчик:** На одной стороне «подковы» находится инфракрасный излучатель, а на другой — приемник. Когда прорезь пуста, свет попадает на приемник. Когда мы помещаем в прорезь непрозрачный предмет (например, лист бумаги), луч прерывается. Датчик отправляет цифровой сигнал (0 или 1) на micro:bit в зависимости от того, перекрыт луч или нет.

2. Особенность пина P7: Мы подключаем датчик к контакту **P7**. У micro:bit этот контакт совмещен с управлением встроенной светодиодной матрицей. Чтобы датчик работал корректно и не возникало помех, **необходимо программно отключить встроенный экран micro:bit**.

Схема подключения:

Используется плата расширения GPIO.

1. Модуль светодиода (LED module) — слева на схеме:

Контакт **IN** (Signal/Вход) — к pinу **P0** (Желтый провод).

Контакт **VCC** (+) — к **5V** (Красный провод).

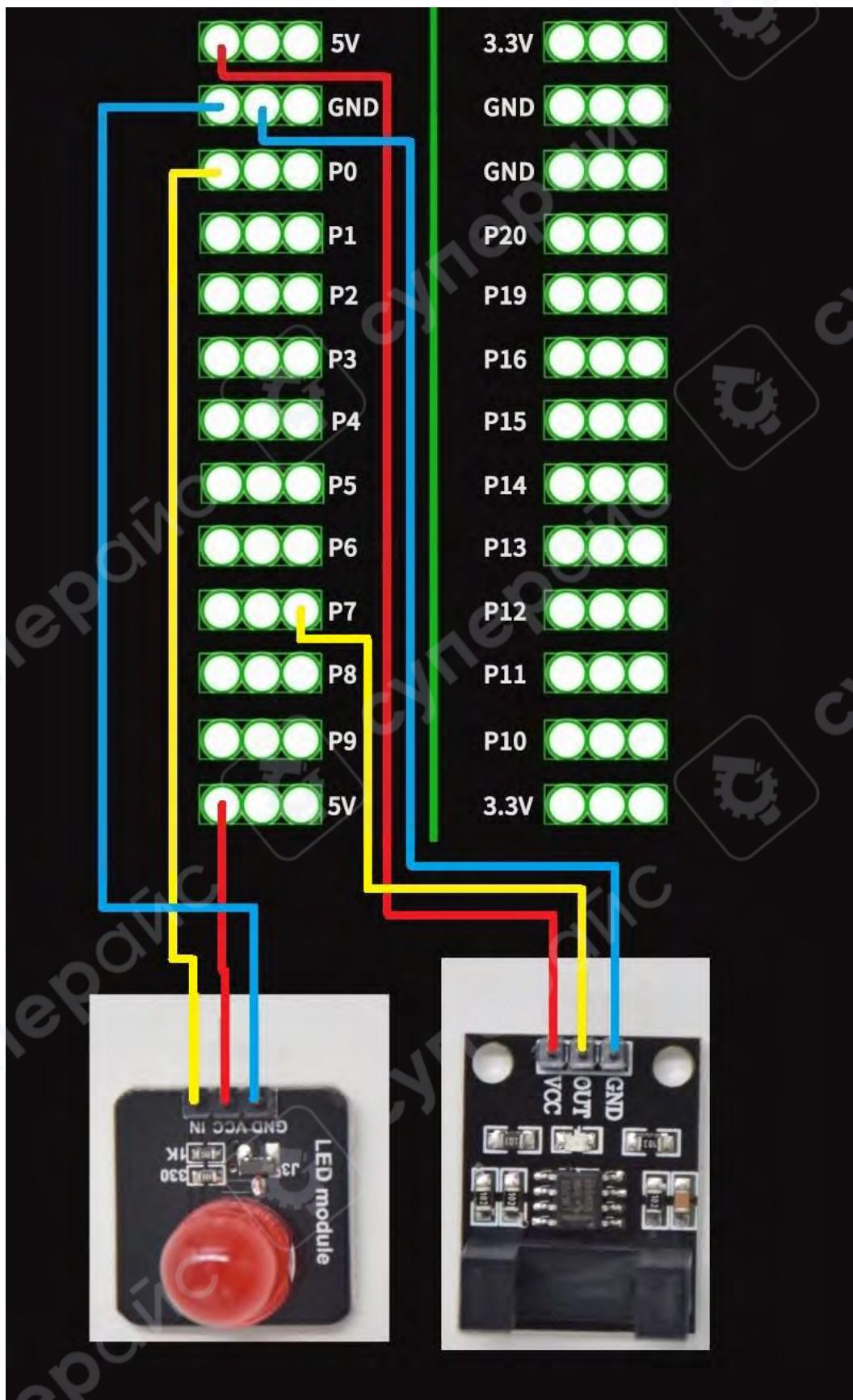
Контакт **GND** (-) — к **GND** (Синий провод).

2. U-образный датчик — справа на схеме:

Контакт **OUT** (Signal/Выход) — к pinу **P7** (Желтый провод).

Контакт **VCC** (+) — к **5V** (Красный провод).

Контакт **GND** (-) — к **GND** (Синий провод).



Подключение светодиода и фотоэлектрического датчика

Программный код (сборка блоков):

Блок «При начале» (On Start):

- В категории **LED (Светодиоды)** зайдите в раздел ...More (Дополнительно) и выберите блок led enable [false] (включить экран [ложь]).

Важно: Этот блок отключает встроенную матрицу 5x5 на лицевой стороне micro:bit. Это **обязательно**, так как мы используем пин P7 для датчика, а он электрически связан с дисплеем.

Блок «Постоянно» (Forever):

- Используем логический блок «Если... то... иначе» (If... then... else).

Условие (If):

- Блок сравнения =.

- Слева: digital read pin P7 (цифровое чтение пина P7) из категории **Pins (Пины)**.

- Справа: число 1.

Смысл: Проверяем, выдает ли датчик сигнал "1" (это происходит, когда луч перекрыт).

To (Then):

- digital write pin P0 to 1 (цифровая запись на пин P0 значения 1).

Действие: Включаем внешний светодиод.

Иначе (Else):

- digital write pin P0 to 0 (цифровая запись на пин P0 значения 0).

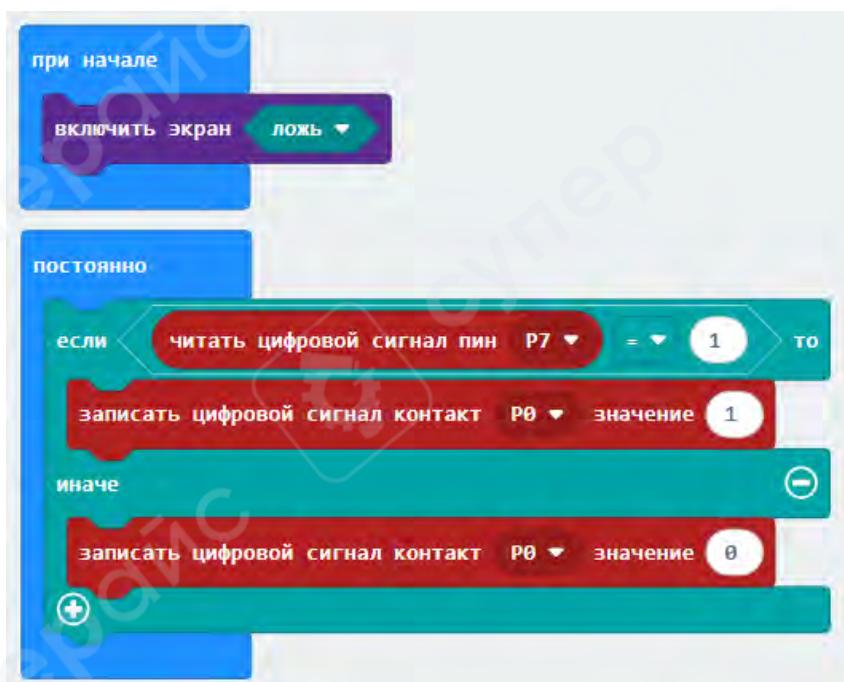
Действие: Выключаем внешний светодиод.

Результат:

После загрузки программы встроенный экран micro:bit погаснет.

1. Если прорезь датчика пуста — внешний красный светодиод **не горит**.
2. Возьмите плотный лист бумаги или пластиковую карту и перекройте прорезь датчика (вставьте внутрь «подковы»). Внешний светодиод **загорится**.
3. Уберите предмет — светодиод снова погаснет.

Файлы: Файл урока: U型传感器实验.hex. Скопируйте его на диск MICROBIT.



2.26 Урок: Эксперимент с четырехразрядным цифровым индикатором (TM1637)

Цель:

Научиться подключать и программировать внешний 4-разрядный семисегментный дисплей на базе драйвера TM1637 для отображения чисел.

Принцип работы:

В этом уроке используется модуль дисплея, состоящий из четырех «восьмерок» (семисегментных индикаторов).

Управление таким количеством светодиодов напрямую потребовало бы слишком много проводов (пинов). Поэтому на модуле установлен специальный драйвер — микросхема **TM1637**. Она позволяет управлять всем дисплеем всего по двум сигнальным проводам:

1. **CLK (Clock)** — тактовый сигнал (синхронизация).
2. **DIO (Data I/O)** — линия данных.

Схема подключения:

Используется плата расширения GPIO.

См. приложенную схему.

- **Модуль дисплея (4-Digit Display):**

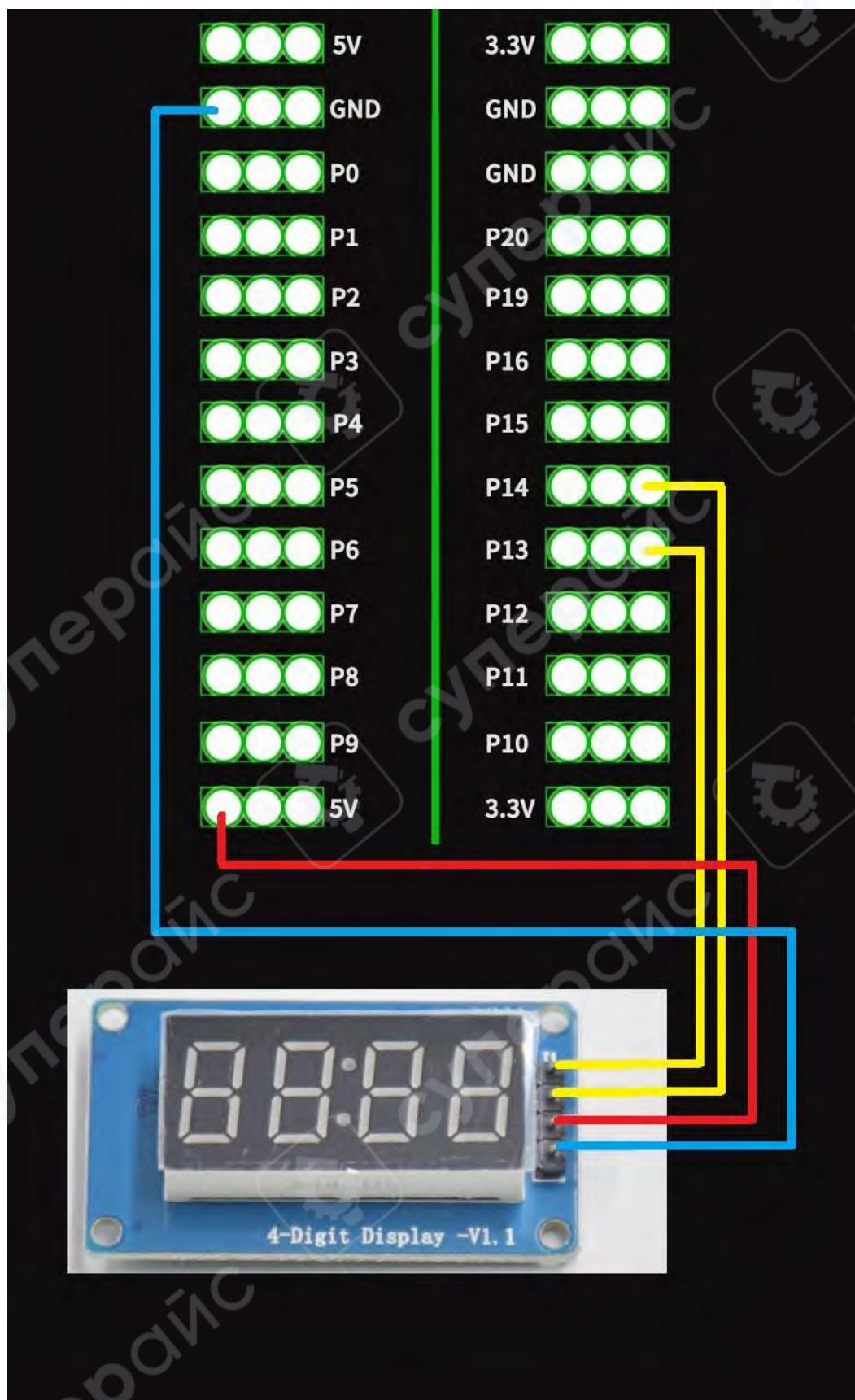
Контакт **GND** (Черный/Синий провод) — к **GND** на плате.

Контакт **VCC** (Красный провод) — к **5V** на плате.

Контакт **DIO** (Желтый провод) — к пину **P14**.

Контакт **CLK** (Желтый провод) — к пину **P13**.

Примечание: На модуле порядок пинов может отличаться, ориентируйтесь на подписи на самой плате дисплея.



Подключение четырехсегментного дисплея

Программный код (сборка блоков):

⚠ Предварительный шаг: Установка расширения

В редакторе MakeCode нажмите на меню **Extensions** (Расширения). В строке поиска введите «<https://github.com/zhuning239/TM1637>». Выберите найденный пакет. После этого в меню блоков появится новая категория.

8 TM1637

Блок «При начале» (On Start):

- Зайдите в категорию **TM1637**.
- Выберите блок инициализации: set TM1637 CLK [P13] DIO [P14] intensity [7] LED count [4].
 - **CLK:** Установите **P13**.
 - **DIO:** Установите **P14**.
 - **Intensity (Яркость):** Установите **7** (максимальная яркость).
 - **LED count (Количество знаков):** Установите **4**.

Блок «Постоянно» (Forever):

Мы будем поочередно выводить два числа: «2024» и «0823» с задержкой в 1 секунду.

1. Вывод первого числа (2024):

- Добавьте 4 блока TM1637 show digit [] at [].
 - show digit 2 at 0 (Показать цифру 2 на позиции 0 — самая левая).
 - show digit 0 at 1 (Показать цифру 0 на позиции 1).
 - show digit 2 at 2 (Показать цифру 2 на позиции 2).
 - show digit 4 at 3 (Показать цифру 4 на позиции 3 — самая правая).

2. Пауза:

- Добавьте блок pause (ms) 1000 (ждать 1 секунду).

3. Вывод второго числа (0823):

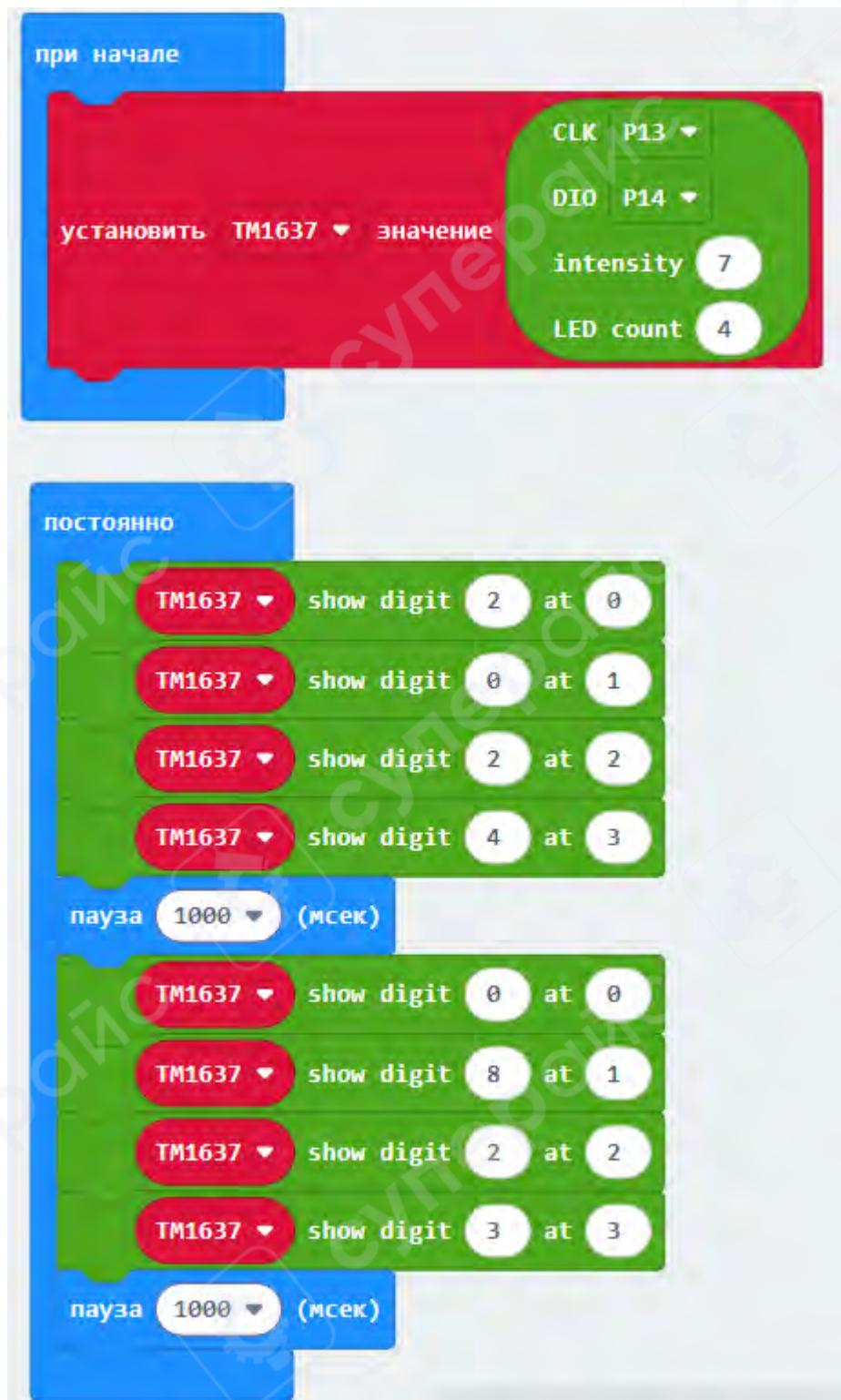
- Добавьте еще 4 блока TM1637 show digit [] at [].
 - show digit 0 at 0
 - show digit 8 at 1
 - show digit 2 at 2
 - show digit 3 at 3

4. Пауза:

- Добавьте блок pause (ms) 1000.

Файлы:

Файл урока: TM1637 四位数码管.hex. Скопируйте его на MICROBIT.



2.27 Урок: Эксперимент по обнаружению капель дождя

Цель: Создать устройство, которое определяет наличие осадков (дождя) и выводит информацию на экран micro:bit. Мы научимся работать как с аналоговым, так и с цифровым сигналом датчика.

Принцип работы:

Датчик дождя состоит из двух частей: сенсорной панели (дорожки на плате) и модуля управления (компаратора).

1. **Сенсорная панель:** Когда на дорожки попадает вода, сопротивление между ними уменьшается, так как вода проводит электричество.

2. **Модуль управления:** Преобразует это изменение сопротивления в электрический сигнал.

- **Аналоговый выход (AO):** Выдает точное значение влажности (чем больше воды, тем ниже сопротивление и меняется напряжение).

- **Цифровой выход (DO):** Выдает простой сигнал «Да/Нет» (1 или 0) в зависимости от порога чувствительности, который настраивается синим потенциометром на плате.

Необходимые компоненты:

- Плата BBC micro:bit.
- Плата расширения GPIO (T-Type).
- Модуль датчика дождя (сенсорная пластина + плата управления).
- Соединительные провода.

Схема подключения:

Обратите внимание на правильность подключения проводов к плате расширения согласно фото:

1. Соединение сенсорной пластины и платы управления:

Подключите сенсорную пластину к двум штырькам на одной стороне платы управления (полярность не важна).

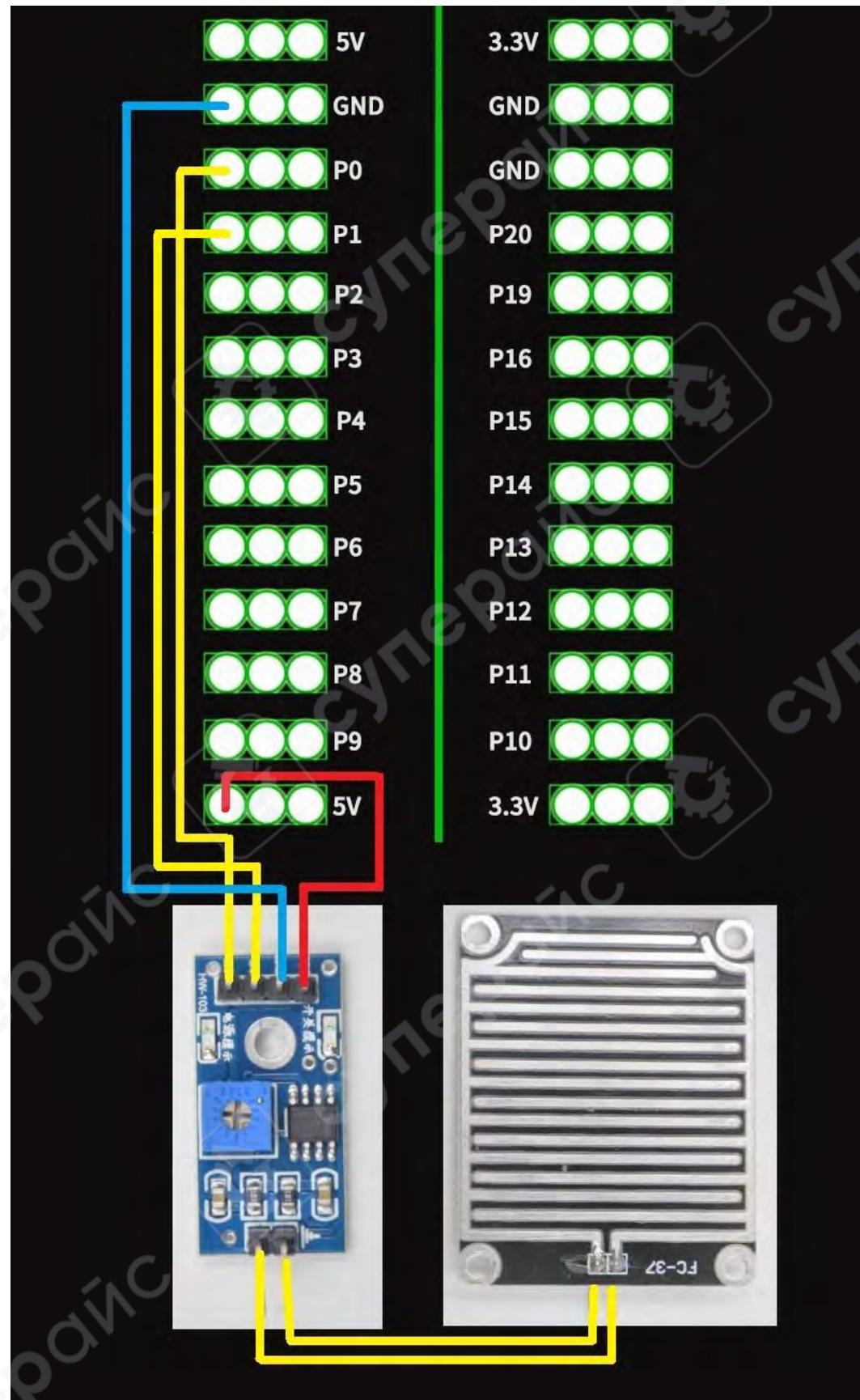
2. Подключение платы управления к GPIO (micro:bit):

VCC (Питание) → **5V** (Красный провод).

GND (Земля) → **GND** (Синий провод).

AO (Аналоговый выход) → **P0** (Желтый провод). Здесь мы считываем уровень влажности.

DO (Цифровой выход) → **P1** (Желтый провод). Здесь мы считываем статус "Есть дождь / Нет дождя".



Подключение датчика дождя

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

В этом цикле мы будем одновременно выводить числовое значение влажности и текстовое сообщение о статусе погоды.

Шаг 1: Вывод аналогового значения

- Зайдите в меню **Basic (Основное)** → выберите блок **show number (показать число)**.
- Зайдите в меню **Pins (Расширенные -> Пины/Контакты)**.
- Выберите блок **analog read pin P0** (аналоговое чтение пина P0).
- Вставьте его внутрь блока «показать число».

Теперь экран будет показывать число (от 0 до 1023), зависящее от количества воды на пластине.

Шаг 2: Логика определения дождя (Цифровой сигнал)

- Зайдите в меню **Logic (Логика)** → выберите блок **if ... then ... else (если ... то ... иначе)**.
 - В условие «если» вставьте блок сравнения (=).
 - В левую часть сравнения: Меню **Pins** → **digital read pin P1** (цифровое чтение пина P1).
 - В правую часть сравнения: Число 1.
- Логика:** Если на пине P1 сигнал равен 1 (Высокий уровень), значит, датчик сухой.
- **Раздел «To» (Then):**
 - Меню **Basic** → блок **show string (показать строку)**.
 - Введите текст: "Not Raining" (Не идет дождь).
 - **Раздел «Иначе» (Else):**
 - Меню **Basic** → блок **show string (показать строку)**.
 - Введите текст: "Raining" (Идет дождь).

Результат:

После загрузки программы:

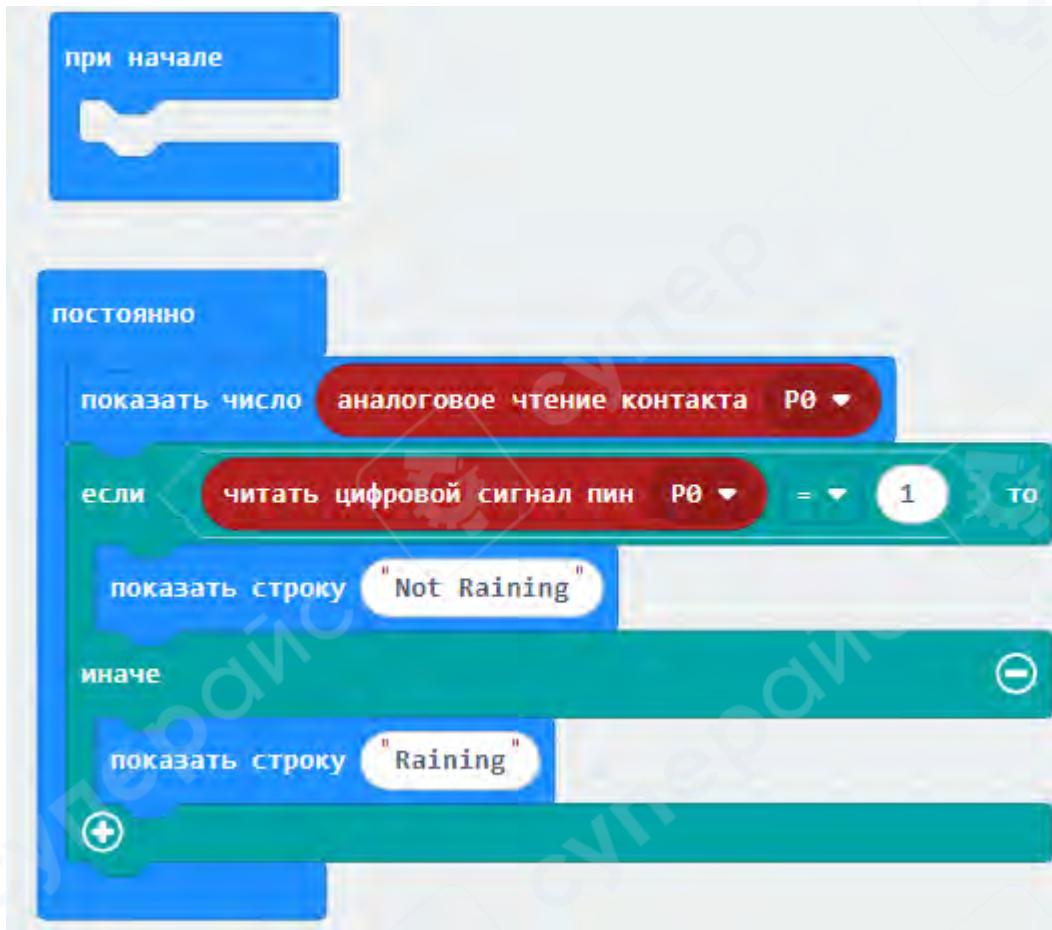
1. Сначала на экране пробежит число (значение с аналогового датчика P0).
2. Затем пробежит надпись:
 - Если пластина сухая: "**Not Raining**".
 - Если капнуть на пластину водой: "**Raining**".

Примечание (Настройка чувствительности):

На маленькой синей плате есть **потенциометр** (синий квадрат с крестиком). Если датчик срабатывает неправильно (например, показывает "Raining", когда сухо, или наоборот):

- Возьмите маленькую отвертку.
- Медленно крутите крестик, пока не загорится или не погаснет светодиодный индикатор на плате датчика, добиваясь нужной чувствительности.

Файлы: Файл урока: 雨滴检测实验.hex (пример названия).



2.28 Урок: Эксперимент с джойстиком PS2

Цель: Научиться управлять изображением на экране micro:bit с помощью аналогового джойстика. Мы запрограммируем отображение стрелок направления (вверх, вниз, влево, вправо) и реакции на нажатие кнопки джойстика.

Принцип работы: Джойстик PS2 состоит из двух потенциометров (переменных резисторов) и одной тактовой кнопки.

1. **Оси X и Y:** Работают как аналоговые датчики. Когда вы отклоняете ручку, меняется напряжение на выходе. Micro:bit считывает это как число от **0** до **1023**.

- В центре значение примерно **512**.
- Края диапазона — это **0** и **1023**.

2. **Кнопка (Z):** Работает как цифровой датчик. При нажатии замыкает цепь на землю (GND).

Необходимые компоненты:

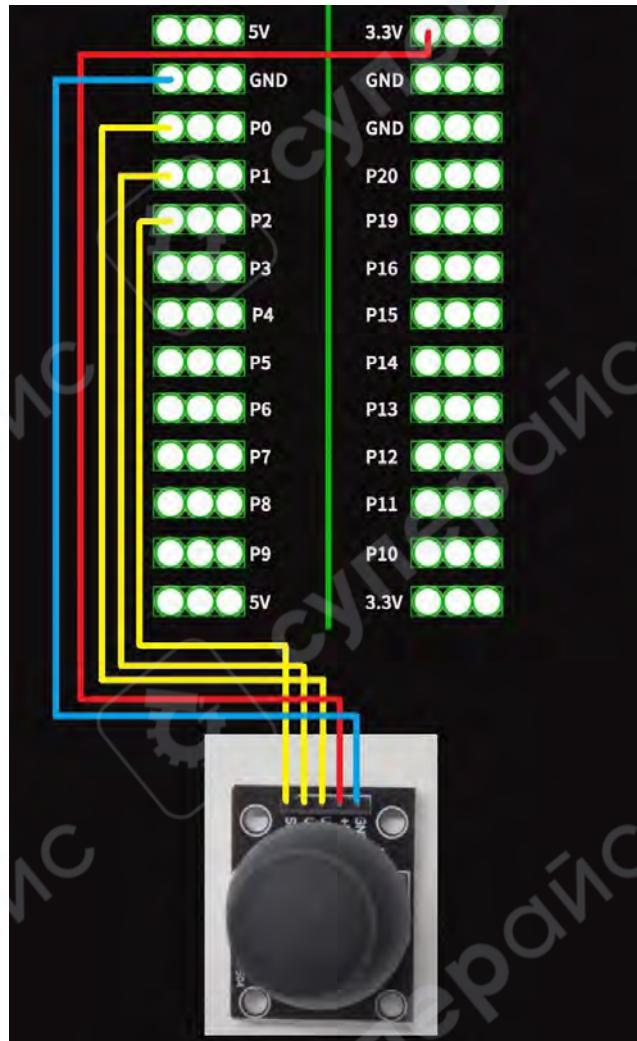
- Плата BBC micro:bit.
- Плата расширения GPIO (T-Type).
- Модуль джойстика (Joystick Module).
- Соединительные провода.

Схема подключения: Внимательно соедините провода согласно схеме на фото:

1. **GND** (Земля на джойстике) → **GND** (Синий провод на плате).

2. **+5V** (Питание на джойстике) → **3.3V** (Красный провод на плате). Примечание: Для micro:bit рекомендуется использовать 3.3V для корректного считывания аналогового сигнала.

3. **VRx** (Ось X) → **P0** (Желтый провод).
4. **VRy** (Ось Y) → **P1** (Желтый провод).
5. **SW** (Кнопка/Switch) → **P2** (Желтый провод).



Подключение джойстика

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Зайдите в меню Pins (Расширенные -> контакты).
- Выберите блок set pull pin P2 to UP (задать преобразования контакта P2 вверх).

Пояснение: Это обязательно для корректной работы кнопки джойстика, чтобы в ненажатом состоянии сигнал был устойчивым.

Блок «Постоянно» (Forever): Мы будем использовать конструкцию «Если — Иначе Если» для проверки всех направлений.

Логика условий:

1. **Движение ВЛЕВО (Ось X):**

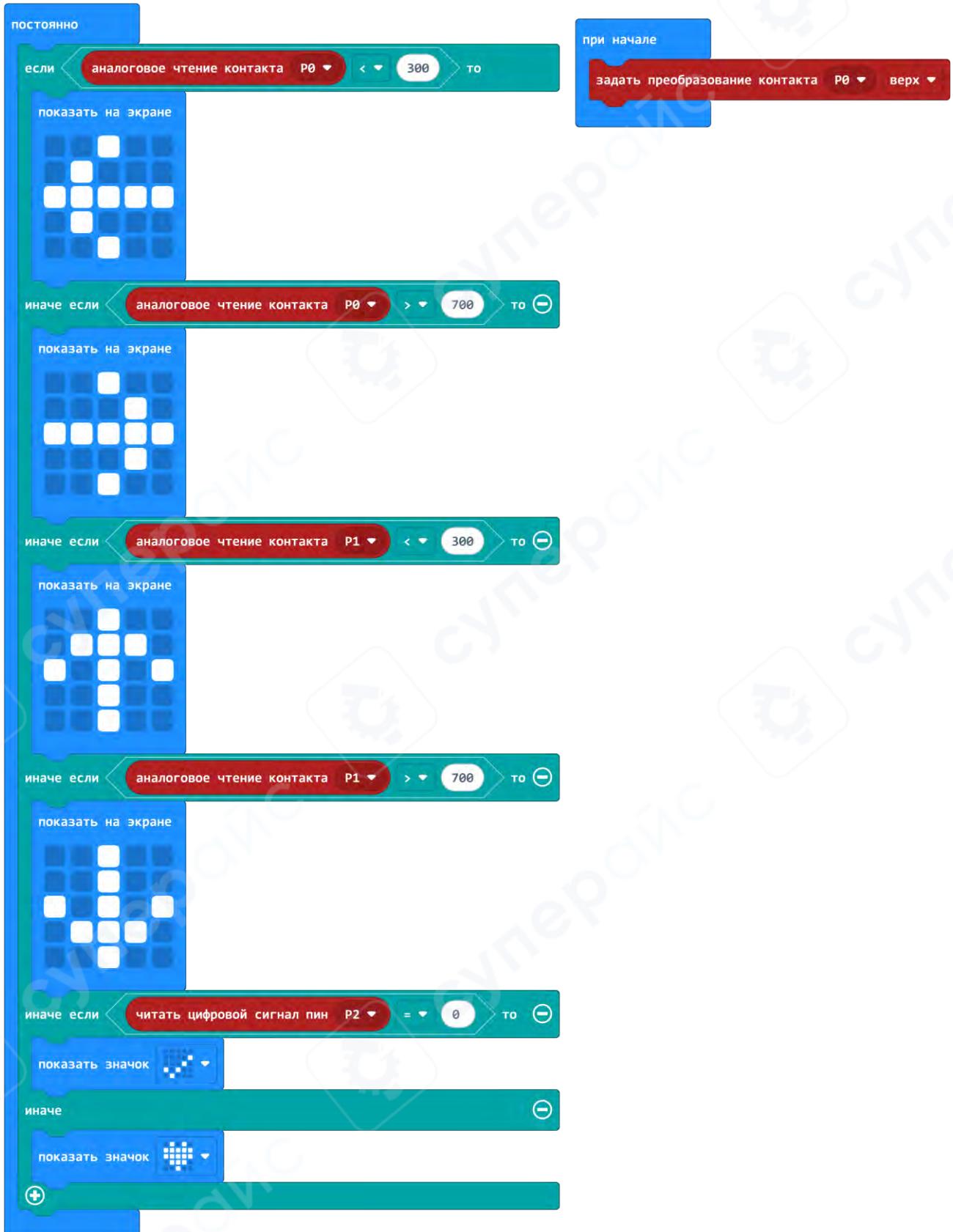
- Блок if (если): analog read pin P0 (аналоговое чтение пина P0) < (меньше) 300.

- То: Блок **show leds** (показать светодиоды) → Нарисуйте стрелку **Влево**.
- 2. **Движение ВПРАВО (Ось X):**
 - Нажмите «+» или значок настройки на блоке «если», чтобы добавить **else if** (иначе если).
 - Условие: **analog read pin P0** (аналоговое чтение пина P0) > (больше) 700.
 - То: Блок **show leds** → Нарисуйте стрелку **Вправо**.
- 3. **Движение ВВЕРХ (Ось Y):**
 - Добавьте **else if**.
 - Условие: **analog read pin P1** (аналоговое чтение пина P1) < (меньше) 300.
 - То: Блок **show leds** → Нарисуйте стрелку **Вверх**.
- 4. **Движение ВНИЗ (Ось Y):**
 - Добавьте **else if**.
 - Условие: **analog read pin P1** (аналоговое чтение пина P1) > (больше) 700.
 - То: Блок **show leds** → Нарисуйте стрелку **Вниз**.
- 5. **Нажатие КНОПКИ (Кнопка SW):**
 - Добавьте **else if**.
 - Условие: **digital read pin P2** (цифровое чтение пина P2) = (равно) 0.
 - То: Блок **show icon** (показать значок) → Выберите **Small Square** (Маленький квадрат) или любую другую иконку нажатия.
- 6. **Иначе (Else) — Джойстик в центре:**
 - В самом конце блока условий, если ни одно из условий выше не выполнено (ручка в центре, кнопка не нажата).
 - То: Блок **show icon** (показать значок) → Выберите **Heart** (Сердце). Это будет означать режим ожидания.

Результат: После загрузки программы:

- В спокойном состоянии на экране горит **Сердце**.
- Если потянуть джойстик в любую сторону — появится соответствующая **стрелка**.
- Если нажать на стик сверху (до щелчка) — появится **маленький квадрат**.

Файлы: Файл урока: PS2 遥感实验.hex



2.29 Урок: Эксперимент с модулем потенциометра

Цель: Создать устройство для плавной регулировки яркости светодиода с помощью поворотной ручки (потенциометра). Мы изучим принцип работы аналогового ввода и вывода (ШИМ).

Принцип работы:

1. **Потенциометр:** Это переменный резистор. Когда вы вращаете ручку, меняется напряжение на сигнальном выводе. Micro:bit считывает это напряжение через pin **P0** и преобразует его в числовое значение от **0** (минимум) до **1023** (максимум).

2. **Светодиод (LED):** Подключен к pinu **P1**. Чтобы менять яркость, мы используем команду «Аналоговая запись» (Analog Write). Чем больше число мы отправляем на светодиод, тем ярче он светит.

3. **Связь:** Программа просто берет число с потенциометра и передает его на светодиод.

Необходимые компоненты:

- Плата BBC micro:bit.
- Плата расширения GPIO (T-Type).
- Модуль потенциометра (Rotation Sensor).
- Модуль светодиода (одноцветный, например, белый или красный).
- Соединительные провода.

Схема подключения:

Следуйте цветам проводов на схеме для правильного соединения:

1. **Модуль потенциометра (Левый на фото):**

GND (Земля) → **GND** (Синий провод).

VCC (Питание) → **3.3V** (Красный провод).

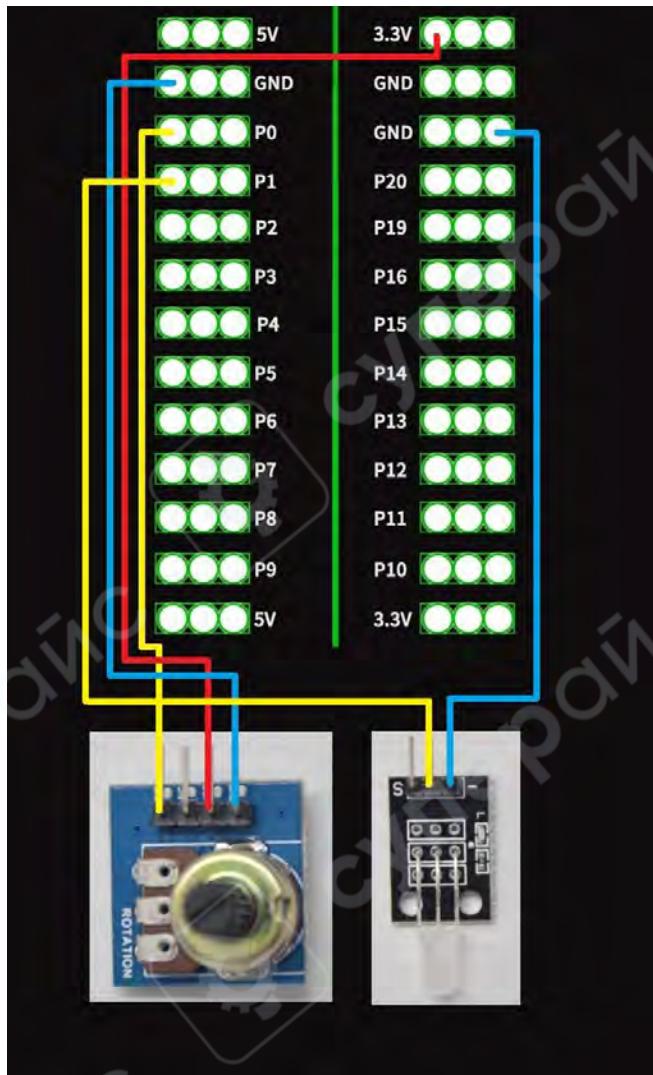
SIG (Сигнал) → **P0** (Желтый провод). Вход данных.

2. **Модуль светодиода (Правый на фото):**

- / **G** (Земля) → **GND** (Синий провод, подключен к соседнему pinu GND).

S (Сигнал) → **P1** (Желтый провод). Выход управления яркостью.

Примечание: Центральный контакт на светодиодном модуле в данной схеме не используется).



Подключение модуля потенциометра и светодиода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

В этом цикле мы непрерывно считываем положение ручки и обновляем яркость лампочки.

- Зайдите в меню Pins (Расширенные -> Пины).
- Выберите красный блок analog write pin P0 to 1023 (аналоговая запись на пин P0 значение 1023).

• **Измените параметры блока:**

1. В выпадающем списке пина выберите P1 (так как светодиод подключен к P1).
 2. В поле значения (вместо 1023) нужно вставить блок чтения.
- Снова зайдите в меню Pins.
 - Возьмите овальный блок analog read pin P0 (аналоговое чтение пина P0).
 - Вставьте этот овальный блок внутрь красного блока, в поле значения.

Итоговая конструкция должна читаться так:

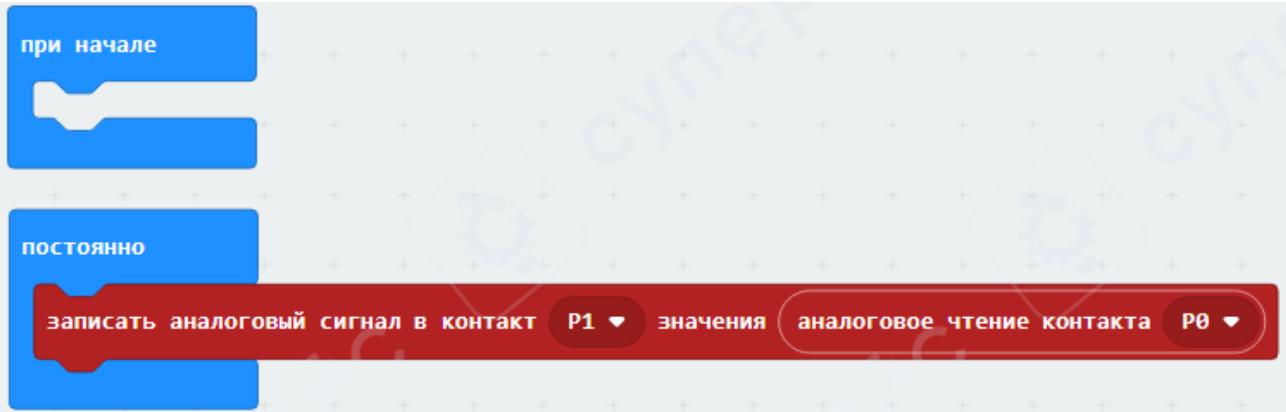
аналоговая запись на пин [P1] значение [аналоговое чтение пина P0]

Результат:

После загрузки программы:

- Попробуйте медленно вращать ручку потенциометра.
 - Светодиод будет плавно менять свою яркость от полного выключения до максимального свечения, синхронно с вращением ручки.

Файлы: Файл урока: 电位器实验.hex.



2.30 Урок: Эксперимент с датчиком Холла

Цель:

Создать устройство, реагирующее на магнитное поле. При приближении магнита к датчику Холла цвет светодиодного индикатора должен изменяться.

Принцип работы: В этом уроке используется датчик Холла — устройство, которое определяет наличие магнитного поля.

- Датчик подключен к цифровому pinу **P2**.
 - Когда магнита рядом нет, датчик выдает сигнал высокого уровня (1).
 - Когда магнит подносят близко, датчик срабатывает и выдает сигнал низкого уровня (0).
 - Программа считывает состояние датчика и переключает питание на двухцветном светодиодном модуле (подключенном к portам **P0** и **P1**), изменяя его цвет.

Подключение оборудования

Используйте плату расширения GPIO для соединения модулей. Внимательно следите за цветами проводов и маркировкой.

1. Модуль датчика Холла (красная плата):

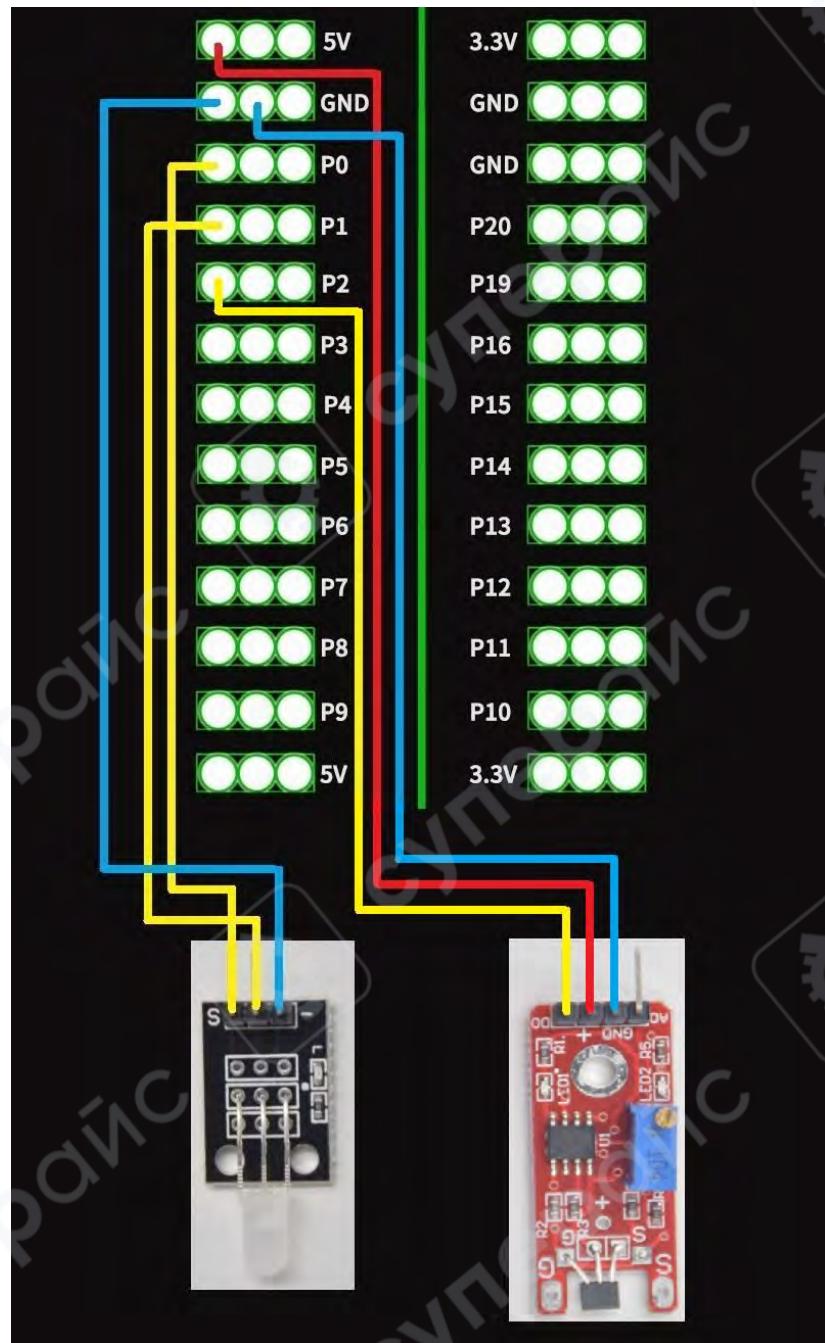
- S (Signal)** — подключить к **P2** (желтый провод).
+ (VCC) — подключить к **5V** (красный провод).
- (GND) — подключить к **GND** (черный провод).

2. Модуль двухцветного светодиода (черная плата):

- **(GND)** — подключить к **GND** (синий провод).

Крайний контакт (S) — подключить к Р0 (желтый провод).

Средний контакт – подключить к Р1 (желтый провод).



Подключение датчика Холла и светодиода

Программный код (сборка блоков)

Код работает в бесконечном цикле, постоянно проверяя состояние датчика.

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

Внутри этого блока мы размещаем логическое условие.

Шаг 1: Настройка условия

1. Зайдите в меню **Logic (Логика)**.
2. Выберите блок **if ... then ... else** (если ... то ... иначе).
3. Перетащите его внутрь блока **Forever**.

Шаг 2: Чтение данных с датчика

1. Нам нужно условие сравнения. Из меню **Logic** возьмите блок сравнения **0 = 0**.
2. Зайдите в меню **Pins** (**Расширенные -> Контакты**) (красные блоки).
3. Выберите блок **digital read pin P0** (цифровое чтение контакта P0). Измените **P0** на

P2.

4. Вставьте этот блок в левую часть сравнения.
5. В правой части сравнения напишите цифру **1**.

Шаг 3: Управление светодиодом (Раздел "То") Если магнита нет (условие истинно):

1. Из меню **Pins** возьмите блок **digital write pin P0 to 0** (цифровая запись на пин P0 значения 0).

2. Установите: Пин **P0** значение **0**.

3. Добавьте еще один такой же блок под ним: Пин **P1** значение **1**.

- **Действие:** Включается один цвет светодиода (например, зеленый), второй выключен.

Шаг 4: Управление светодиодом (Раздел "Иначе") Если магнит обнаружен (условие ложно):

1. В секцию **Else (Иначе)** добавьте блоки управления пинами:

2. Пин **P0** значение **1**.

3. Пин **P1** значение **0**.

Действие: Цвета меняются местами (например, загорается красный).

Результат:

1. Загрузите программу (файл .hex) на плату micro:bit.
2. В обычном состоянии (без магнита) светодиодный модуль будет гореть одним цветом.
3. Поднесите магнит к черному элементу на датчике Холла.
4. Светодиод мгновенно изменит цвет на другой.
5. Уберите магнит — цвет вернется к исходному.

Файлы: Файл урока: **霍尔传感器实验.hex**



2.31 Урок: Эксперимент с аналоговым датчиком температуры

Цель: Научиться считывать данные с внешнего аналогового датчика температуры (термистора) и выводить полученные значения на дисплей micro:bit.

Принцип работы: В отличие от встроенного в процессор датчика, этот модуль использует **термистор NTC**. Сопротивление термистора меняется в зависимости от температуры окружающей среды.

- Датчик подключен к аналоговому pinу P0.
- Micro:bit измеряет напряжение на этом пине (анalogовый сигнал).
- Программа преобразует полученный аналоговый сигнал в градусы Цельсия, используя математическую формулу (коэффициент B), и выводит число на экран.

Подключение оборудования

Используйте плату расширения GPIO. В данном эксперименте используется аналоговый выход датчика (AO), цифровой выход (DO) не задействован.

Модуль датчика температуры (синяя плата):

- AO (Analog Output) — подключить к P0 (желтый провод).

Примечание: На фото датчика это крайний левый штырек, если смотреть на подписи.

- GND (Ground) — подключить к GND (синий провод).

- VCC (Power) — подключить к 3.3V (красный провод).

Важно: На схеме подключения красный провод идет к правой колодке питания, подписанной как 3.3V.

Программный код (сборка блоков)

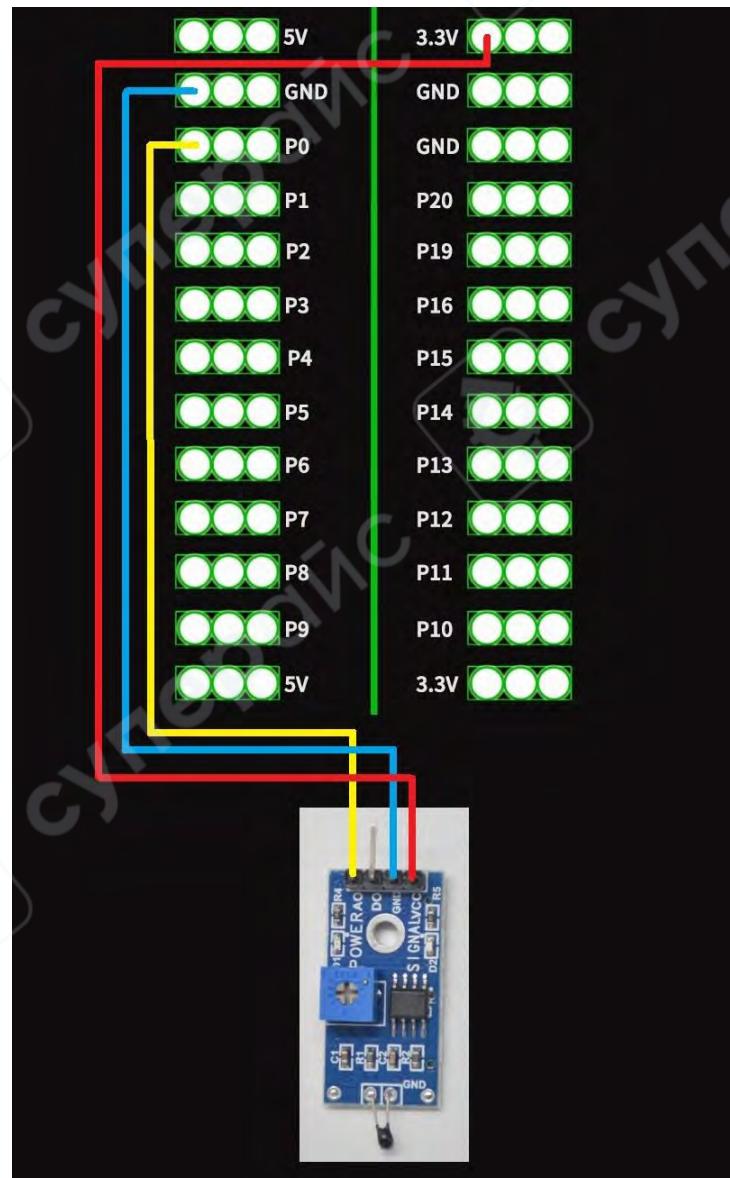
Для работы программы используются блоки из категории расширений (зеленого цвета), предназначенные для автоматического пересчета сигнала в температуру <https://github.com/zhuning239/NTC>.



Блок «При начале» (On Start):

Здесь мы настраиваем параметры термистора.

1. Найдите блок настройки датчика.
2. Выберите блок Set temperature coefficient B (Установить температурный коэффициент B).



3. Установите значение **3380**.
 - Пояснение: 3380 — это бета-коэффициент (Beta Value) термистора, используемого в данном модуле. Он нужен для точного расчета температуры.

Блок «Постоянно» (Forever):

Здесь происходит чтение и вывод данных.

1. Зайдите в меню **Basic (Основное)**.
2. Выберите блок **show number** (показать число).
3. Теперь нам нужно вставить внутрь значение температуры.
 - Возьмите блок **Get temperature** (Получить температуру) из меню расширений.
 - Внутри этого блока обычно находится слот для указания пина.
4. Зайдите в меню **Pins (Пины)** (красные блоки).
5. Выберите блок **analog read pin P0** (аналоговое чтение пина P0).
6. Вставьте блок **analog read pin P0** внутрь блока **Get temperature**.
7. Вставьте всю конструкцию внутрь блока **show number**.

Итоговая логика:

Программа считывает аналоговый сигнал с P0 -> Преобразует его в температуру (используя коэффициент 3380) -> Показывает число на экране.

Файлы: Файл урока: NTC 热敏模块.hex



2.32 Урок: Эксперимент с датчиком звука (Акустическое реле)

Цель: Создать «умную лампу», которая включается при хлопке в ладоши или громком звуке и автоматически гаснет через 1 секунду.

- Принцип работы:** Система использует **датчик звука (микрофон)** с компаратором LM358.
- Датчик настроен на определенный порог громкости.
 - В тишине датчик выдает на pin **P3** сигнал 0 (низкий уровень).
 - Когда микрофон улавливает громкий звук, компаратор срабатывает и посылает на pin **P3** сигнал 1 (высокий уровень).
 - Micro:bit считывает этот сигнал, включает внешний светодиодный модуль (на pinе **P4**) на 1 секунду, а затем выключает его.

Подключение оборудования

Используйте плату расширения GPIO. Обратите внимание, что в этом уроке используются два внешних модуля.

1. Модуль датчика звука (красная плата):

S (Signal) — подключить к P3 (желтый провод).

+ (VCC) — подключить к 5V (красный провод).

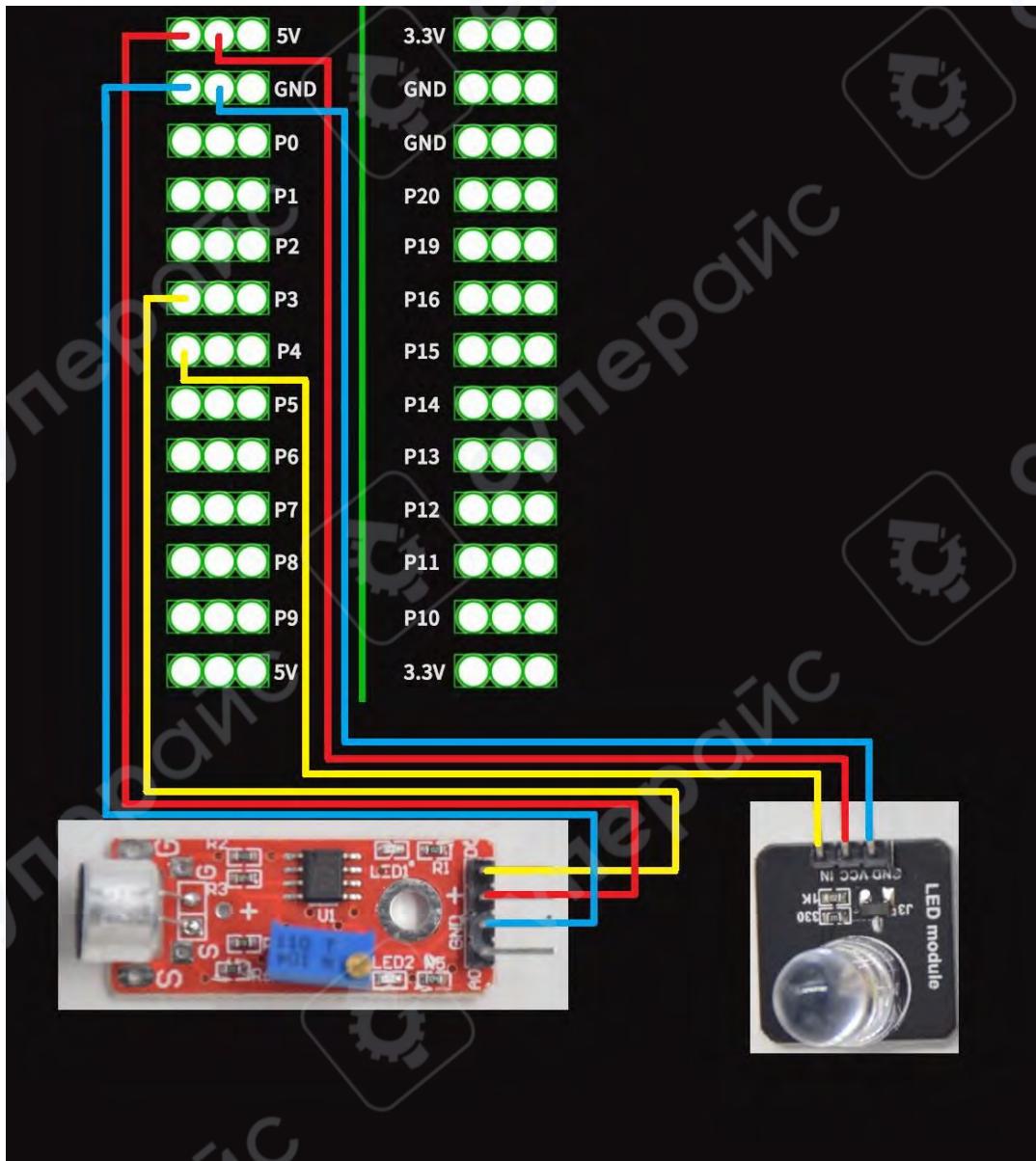
G (GND) — подключить к GND (синий провод).

2. Модуль светодиода (черная плата):

Контакт «VCC»: Подключен красным проводом к пину 5V на плате.

Контакт «GND»: Подключен синим проводом к пину GND на плате.

Контакт «IN» (Signal): Подключен желтым проводом к цифровому pinu P4 на плате.



Подключение акустического реле и светодиода

Программный код (сборка блоков)

Блок «При начале» (On Start): В этом уроке мы отключаем встроенный матричный дисплей micro:bit.

1. Зайдите в меню **LED (Светодиоды)**.
2. Выберите блок **led enable** (включить экран) и нажмите на переключатель, чтобы сменить его на **false** (ложь/выключить).

3. Вставьте его в блок «При начале».

Зачем это нужно? Это гарантирует, что работа встроенного экрана не будет создавать помех для чтения сигнала с датчиков на некоторых версиях плат.

Блок «Постоянно» (Forever): Реализуем логику: "Если слышу звук — включить свет, подождать, выключить".

1. **Условие:**

- Возьмите блок **if ... then ... else** (если ... то ... иначе) из меню **Logic**.
- Создайте условие сравнения: digital read pin P3 = 1. (Блок **digital read pin** находится в меню **Pins**, блок сравнения = в меню **Logic**).

Логика: Если на пине P3 появилась "единица" (сигнал звука).

2. **Действие (Если звук есть):**

- Вставьте блок **digital write pin P4 to 1** (записать 1 на пин P4). Это **включит** внешний светодиод.
- Добавьте блок **pause (ms) 1000** (пауза 1000 мс) из меню **Basic**. Светодиод будет гореть 1 секунду.

3. **Действие (Иначе / Тишина):**

- В секцию **Else** вставьте блок **digital write pin P4 to 0**. Это **выключит** внешний светодиод, когда звука нет.

Результат и Настройка:

1. Загрузите код в micro:bit.
2. Хлопните в ладоши рядом с микрофоном. Внешний светодиод должен загореться на 1 секунду и погаснуть.

Файлы: Файл урока: 声控灯.hex



2.33 Урок: Эксперимент с фоторезистором (Умный светильник)

Цель: Создать устройство, имитирующее работу автоматического уличного освещения или «умного ночника». Внешний светодиод должен автоматически включаться, когда датчик фиксирует падение уровня освещенности (наступление темноты), и выключаться, когда становится светло.

Принцип работы: В этом эксперименте используется **фоторезистор** — компонент, который меняет свое сопротивление в зависимости от количества падающего на него света.

1. Micro:bit считывает аналоговый сигнал с датчика (значение от 0 до 1023).
2. Если значение превышает заданный порог (в нашем случае 500), контроллер считает, что наступила темнота.
3. При наступлении темноты программа подает питание на внешний светодиодный модуль.

Подключение компонентов: Для сборки используйте плату расширения, провода-перемычки, модуль фоторезистора и модуль светодиода.

1. **Модуль фоторезистора (датчик света):**

S (Signal/Желтый провод) → Подключить к pinu P0.

V (VCC/Красный провод) → Подключить к 5V (или 3.3V).

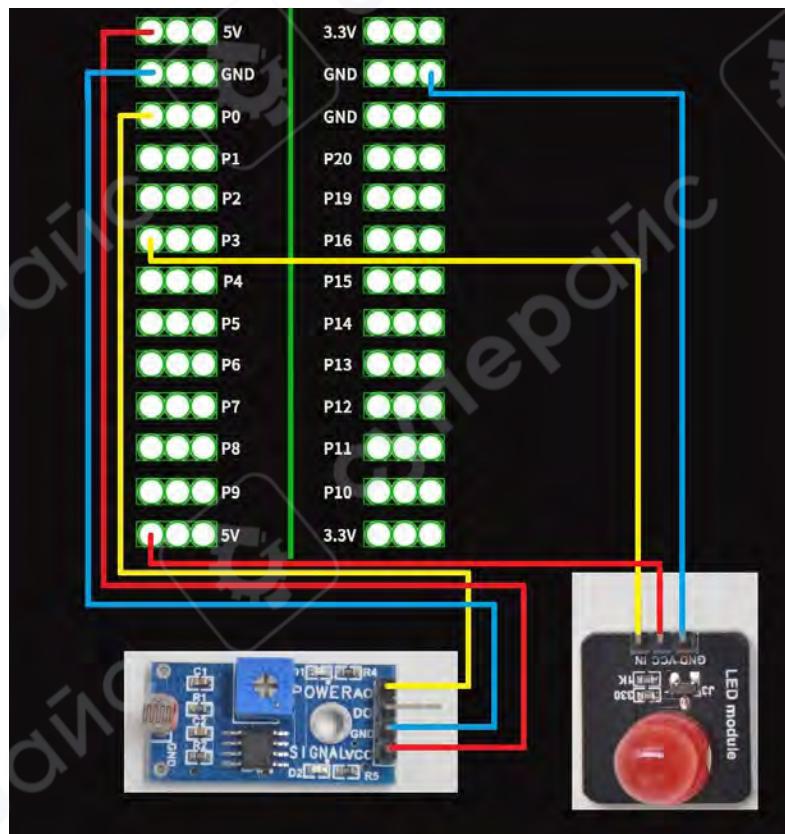
G (GND/Черный или синий провод) → Подключить к GND.

2. **Модуль светодиода (внешняя лампа):**

S (Signal/Желтый провод) → Подключить к pinu P3.

V (VCC/Красный провод) → Подключить к 5V (или 3.3V).

G (GND/Черный или синий провод) → Подключить к GND.



Подключение фоторезистора и светодиода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Для корректной работы внешних компонентов рекомендуется отключить встроенную матрицу micro:bit, чтобы освободить ресурсы пинов.
- Добавьте блок led enable false (включить экран: ложь) из раздела *Led* (Светодиоды) -> *More* (Дополнительно).

Блок «Постоянно» (Forever): Мы будем непрерывно проверять уровень освещенности. Используем логический блок условия.

Шаг 1: Настройка условия

- Зайдите в меню **Logic** (Логика).
- Выберите блок if... then... else (если... то... иначе) и поместите его внутрь блока «Постоянно».
- В условие (шестиугольное поле после «если») вставьте блок сравнения $0 < 0$ из раздела **Logic**.
- Замените знак $<$ на $>$ (больше).

Шаг 2: Чтение данных с датчика

- Зайдите в меню **Pins** (Контакты/Пины).
- Выберите блок analog read pin P0 (чтение аналогового сигнала P0).
- Вставьте этот блок в левую часть сравнения.
- В правой части сравнения напишите число **500**.

Пояснение: 500 — это условный порог срабатывания. Если значение с датчика больше 500, мы считаем, что темно.

Шаг 3: Действие при темноте (To / Then)

- Если условие истинно (темно), нужно включить свет.
- Зайдите в меню **Pins** (Контакты).
- Выберите блок digital write pin P3 to 1 (цифровая запись на контакт P3 значения 1).
- Поместите его в секцию «то» (под «если»). Значение **1** означает подачу напряжения (ВКЛ).

Шаг 4: Действие при свете (Иначе / Else)

- Если условие ложно (светло), нужно выключить свет.
- Скопируйте предыдущий блок записи на пин P3 и поместите его в секцию «иначе».
- Измените значение с **1** на **0**. Значение **0** означает отключение напряжения (ВЫКЛ).

Результат: После загрузки программы:

1. Если в комнате светло, внешний красный светодиод не горит.
2. Попробуйте закрыть датчик фотодиода ладонью или выключить свет в помещении.
3. Как только датчик окажется в тени, внешний светодиод загорится.
4. Уберите руку — светодиод погаснет.

Файлы: Файл урока: **光敏灯.hex**



2.34 Урок: Эксперимент с датчиком пламени (Пожарная сигнализация)

Цель: Создать систему безопасности, которая обнаруживает наличие огня. Если датчик фиксирует пламя, на дисплее micro:bit появляется предупреждение "Fire" (Огонь), в противном случае отображается статус "safe" (Безопасно).

Принцип работы: Датчик пламени реагирует на инфракрасное (ИК) излучение, которое испускает открытый огонь.

1. Датчик подключен к аналоговому порту micro:bit.
2. При отсутствии огня датчик выдает высокое значение напряжения.
3. При появлении пламени (ИК-излучения) сопротивление фототранзистора падает, и значение аналогового сигнала уменьшается (стремится к 0).
4. Программа сравнивает полученное значение с пороговым (200). Если значение **ниже** 200, срабатывает тревога.

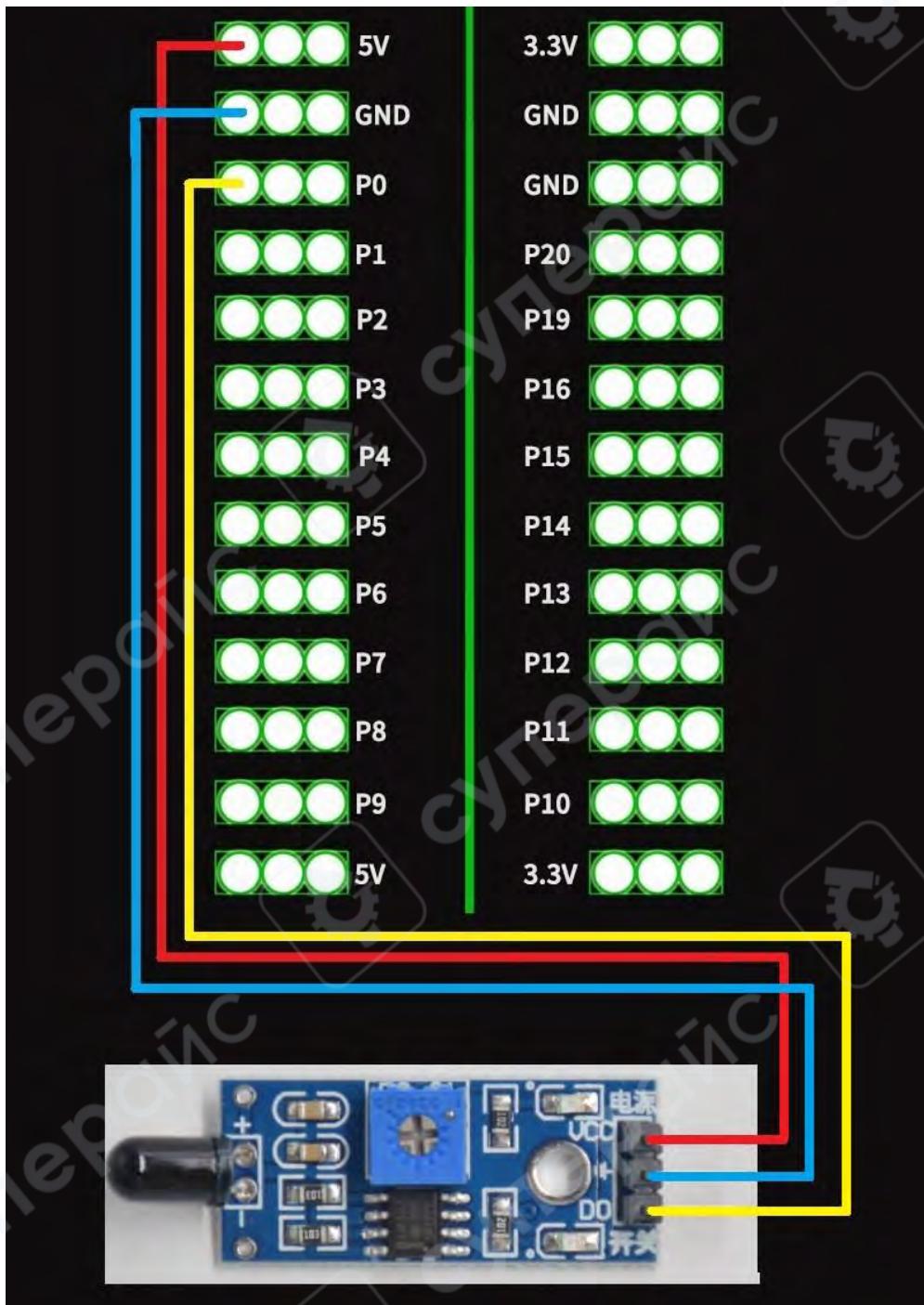
Подключение компонентов: Используйте 3-проводное соединение с платой расширения. Обратите внимание на маркировку контактов на модуле датчика.

- **Модуль датчика пламени:**

DO (Analog Output/Желтый провод) → Подключить к пину P0.

G (GND/Синий провод) → Подключить к GND.

+ (VCC/Красный провод) → Подключить к 5V.



Подключение датчика пламени

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever): Система должна работать непрерывно.

Шаг 1: Настройка логики

- Зайдите в меню **Logic** (Логика).
- Выберите блок **if... then... else** (если... то... иначе) и перетащите его внутрь цикла «Постоянно».

Шаг 2: Условие срабатывания

- Нам нужно проверить, упало ли значение датчика ниже порога.
- В меню **Logic** выберите блок сравнения $0 < 0$.
- В левую часть сравнения вставьте блок analog read pin P0 (аналоговое чтение пина P0) из меню **Pins**.
 - В правую часть впишите число **200**.
 - Логика: analog read pin P0 < 200.

Шаг 3: Действие при обнаружении огня (To / Then)

- Если сигнал меньше 200 (огонь обнаружен), нужно вывести предупреждение.
- Зайдите в меню **Basic** (Основное).
- Выберите блок show string (показать строку).
- Впишите текст: "Fire".

Шаг 4: Действие в безопасном режиме (Иначе / Else)

- Если сигнал больше или равен 200 (огня нет).
- Вставьте еще один блок show string (показать строку) в секцию «иначе».
- Впишите текст: "safe".

Результат:

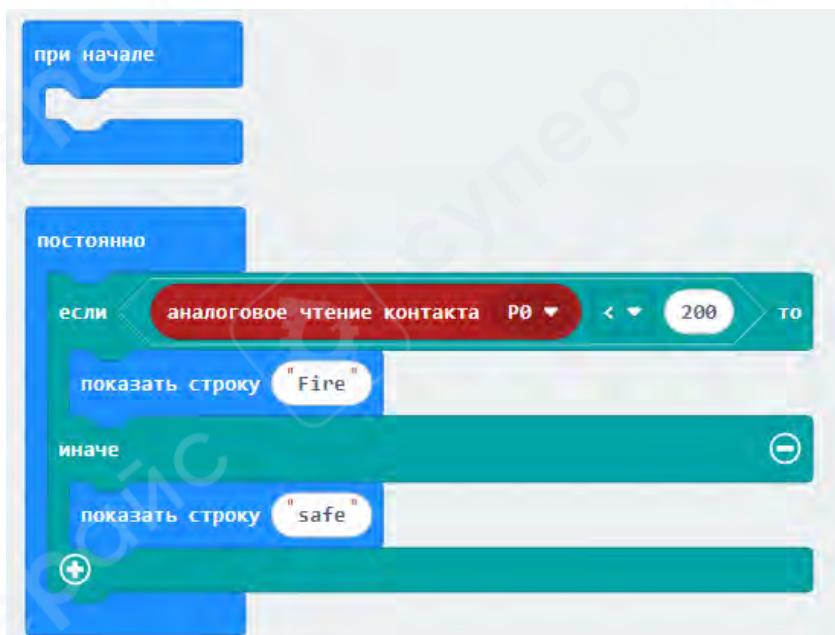
1. Загрузите код в micro:bit.
2. В нормальном состоянии на экране будет бегущая строка "safe".
3. Поднесите источник огня (зажигалку) к датчику.

⚠ ВАЖНО: Не подносите открытое пламя вплотную к черному диоду датчика, чтобы не расплавить его! Держите огонь на расстоянии нескольких сантиметров.

4. Как только датчик "увидит" ИК-излучение пламени, надпись на экране сменится на "Fire".

Примечание: Датчики пламени также реагируют на яркий солнечный свет или мощные лампы накаливания, так как в их спектре тоже есть инфракрасное излучение. Это нормальное поведение физического компонента.

Файлы: Файл урока: 火焰传感器.hex.



2.35 Урок: Эксперимент с датчиком дыма (Детектор газа)

Цель: Создать систему пожарной безопасности, которая отслеживает концентрацию газа или дыма в воздухе. Устройство должно выводить текущий уровень загазованности на экран и включать звуковую сирену (зуммер), если сработал пороговый датчик.

Принцип работы: Используется датчик газа (серии MQ, обычно MQ-2). Он имеет два выхода:

1. **Аналоговый (AO):** Выдает точное значение концентрации газа. Мы выводим его на экран для мониторинга.

2. **Цифровой (DO):** Срабатывает (меняет сигнал с 1 на 0), когда концентрация превышает предел, установленный потенциометром (синий регулятор на датчике). Мы используем этот сигнал для мгновенного включения сирены.

Подключение компонентов: Для этого урока требуется подключить два модуля: датчик газа и зуммер.

1. **Датчик дыма/газа (MQ-2):**

AO (Analog Output/Желтый провод) → Подключить к pinу P0.

DO (Digital Output/Желтый провод) → Подключить к pinу P1.

VCC (Красный провод) → Подключить к 5V (датчику дыма нужно хорошее питание для нагрева).

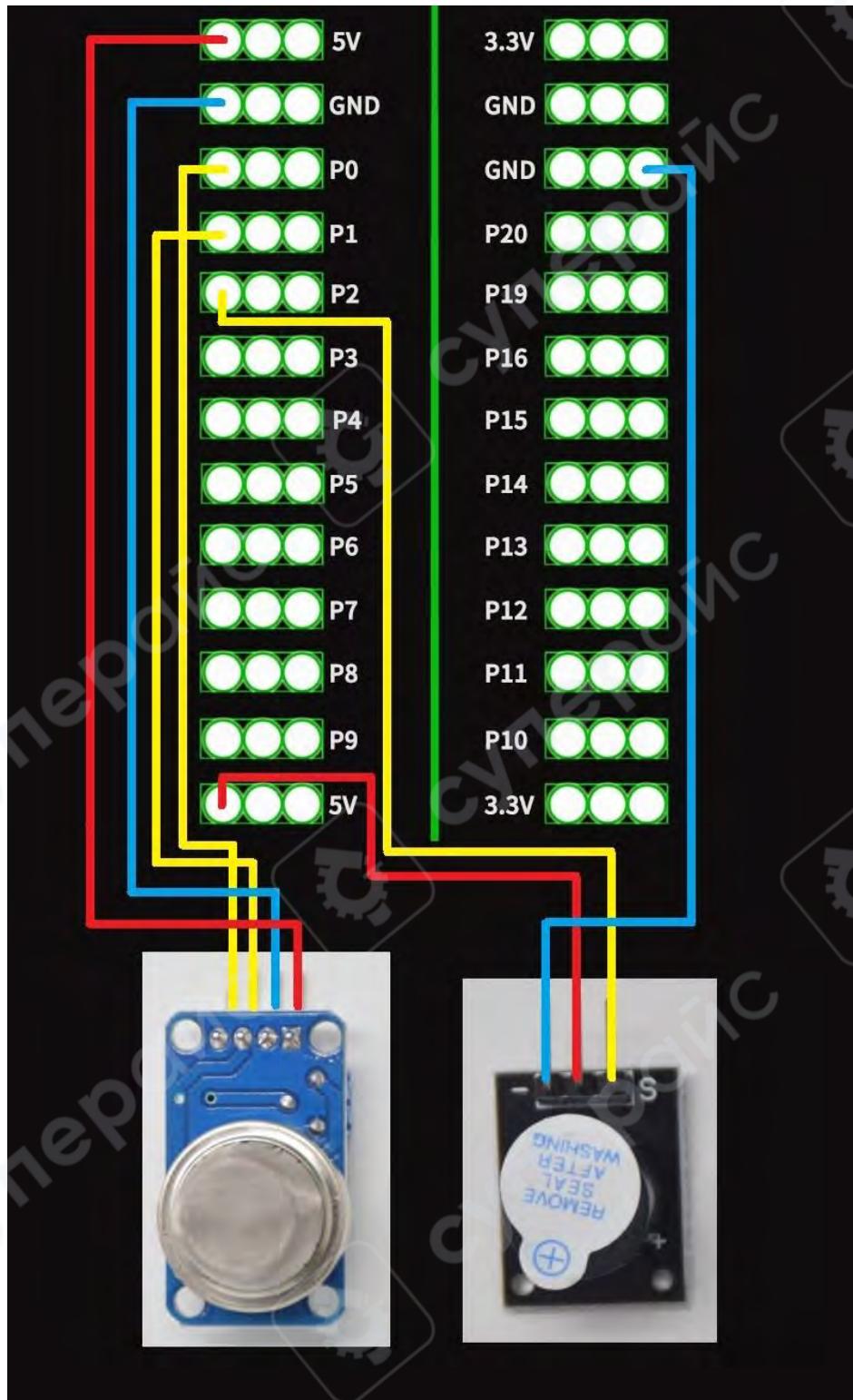
GND (Синий провод) → Подключить к GND.

2. **Модуль зуммера (Buzzer):**

S (Signal/Желтый провод) → Подключить к pinу P2.

VCC (Красный провод) → Подключить к 5V (или 3.3V).

GND (Синий провод) → Подключить к GND.



Подключение датчика дыма и зуммера

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте пустым.

Блок «Постоянно» (Forever):

Шаг 1: Мониторинг значений

- Зайдите в меню **Basic** (Основное).
- Выберите блок show number (показать число).
- Внутрь него поместите блок analog read pin P0 (аналоговое чтение пина P0) из меню **Pins**.

- Это позволит видеть на экране реальные цифры уровня газа.

Шаг 2: Настройка условия тревоги

- Используем цифровой сигнал для точного срабатывания.
- Из меню **Logic** возьмите блок if... then... else.
- Условие: digital read pin P1 = 0.
- Возьмите блок сравнения =, слева поставьте digital read pin P1 (цифровое чтение P1), справа напишите **0**.

Важно: На данном модуле логический «0» означает, что газ обнаружен (срабатывание датчика).

Шаг 3: Включение сирены (To / Then)

- Если газ обнаружен (P1 = 0):
- Из меню **Pins** выберите digital write pin P2 to 1.
- Это подаст напряжение на зуммер.

Шаг 4: Выключение сирены (Иначе / Else)

- Если газа нет (P1 = 1):
- Из меню **Pins** выберите digital write pin P2 to 0.
- Зуммер молчит.

Результат:

1. После включения дайте датчику прогреться около минуты (он может стать теплым — это нормально).
2. На экране бегут цифры (уровень чистоты воздуха).
3. Если подать газ (например, от зажигалки без огня) к сетке датчика:
 - Цифры на экране резко возрастут.
 - Как только сработает внутренний триггер датчика, раздастся писк зуммера.

⚠ Настройка чувствительности: На синей коробочке датчика дыма есть маленький винт (потенциометр).

- Если зуммер пищит постоянно — немного поверните винт по часовой стрелке (или против), пока писк не прекратится.
- Если зуммер не реагирует на газ — крутите винт в обратную сторону, пока не настроите чувствительность.

Файлы: Файл урока: **烟雾报警器实验.hex**.



2.36 Урок: Управление с помощью ИК-пульта

Цель: Создать систему дистанционного управления, позволяющую включать и выключать отдельные цвета на RGB-светодиодном модуле с помощью кнопок инфракрасного (ИК) пульта.

Принцип работы: В этом проекте используется **ИК-приемник**, который улавливает сигналы от пульта дистанционного управления. Каждый раз, когда вы нажимаете кнопку на пульте, он отправляет уникальный цифровой код. Micro:bit через приемник считывает этот код и выполняет соответствующую команду. Мы запрограммируем контроллер так, чтобы определенные кнопки пульта подавали напряжение (сигнал 1) или отключали его (сигнал 0) на пинах P0, P1 и P2, к которым подключен RGB-светодиод.

Необходимое оборудование:

1. Плата BBC micro:bit.
2. Плата расширения GPIO.
3. Модуль RGB-светодиода.
4. Модуль ИК-приемника (IR Receiver).
5. ИК-пульт дистанционного управления.
6. Соединительные провода.

Схема подключения: Обязательно соблюдайте полярность и соответствие пинов.

1. ИК-приемник (IR Receiver):

GND / - (Минус) → GND (синий провод).

VCC / 5V (Плюс) → 5V (красный провод).

S (Сигнал) → P8 (желтый провод).

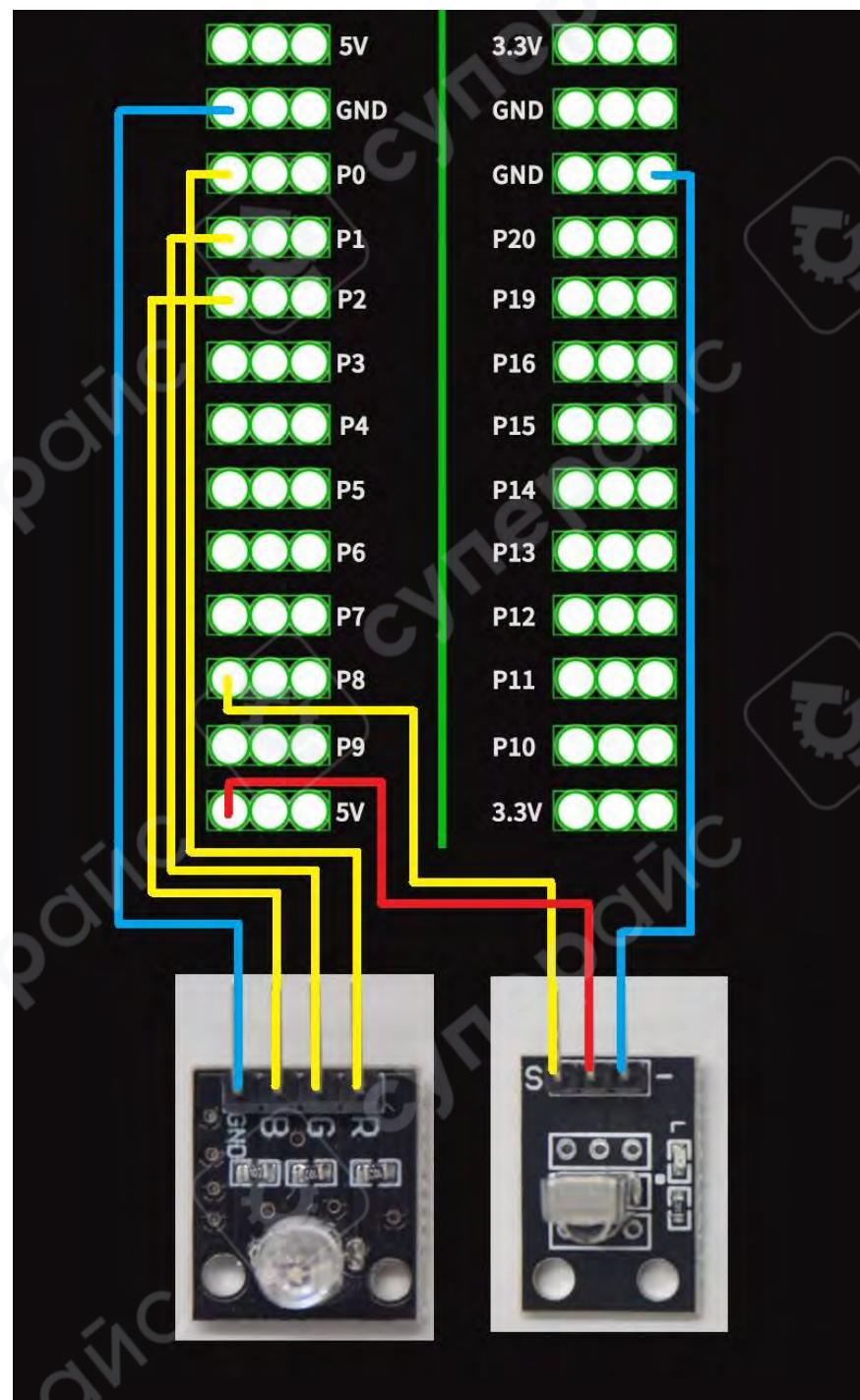
2. RGB-светодиод:

GND / - → GND (синий провод).

R (Red/Красный) → P0 (желтый провод).

G (Green/Зеленый) → P1 (желтый провод).

B (Blue/Синий) → P2 (желтый провод).



Подключение ИК приемника и RGB светодиода

Программный код (сборка блоков):

Примечание: Для работы с ИК-пультом необходимо добавить расширение «IR». Блоки

... IR Receiver

на скриншотах зеленого цвета.

Блок «При начале» (On Start): Инициализация приемника.

1. Выберите блок connect IR receiver at pin... (подключить ИК-приемник к пину...).
2. Установите пин: **P8**.
3. Выберите протокол декодирования: **Keyestudio** (или соответствующий вашему пульту).

Настройка управления (События кнопок): Нам нужно создать 6 пар событий: включение и выключение для каждого из трех каналов цвета.

Группа 1: Красный цвет (P0)

- **Включение:**
- Возьмите блок on IR button [] pressed (когда нажата ИК-кнопка).
- Выберите кнопку: **4**.
- Внутрь вставьте блок digital write pin P0 to 1 (цифровая запись п/п P0 значение 1).
- **Выключение:**
- Возьмите блок on IR button [] pressed.
- Выберите кнопку: **7**.
- Внутрь вставьте блок digital write pin P0 to 0 (цифровая запись п/п P0 значение 0).

Группа 2: Зеленый цвет (P1)

- **Включение:**
- Блок события on IR button pressed, кнопка: **5**.
- Действие: digital write pin P1 to 1.
- **Выключение:**
- Блок события on IR button pressed, кнопка: **8**.
- Действие: digital write pin P1 to 0.

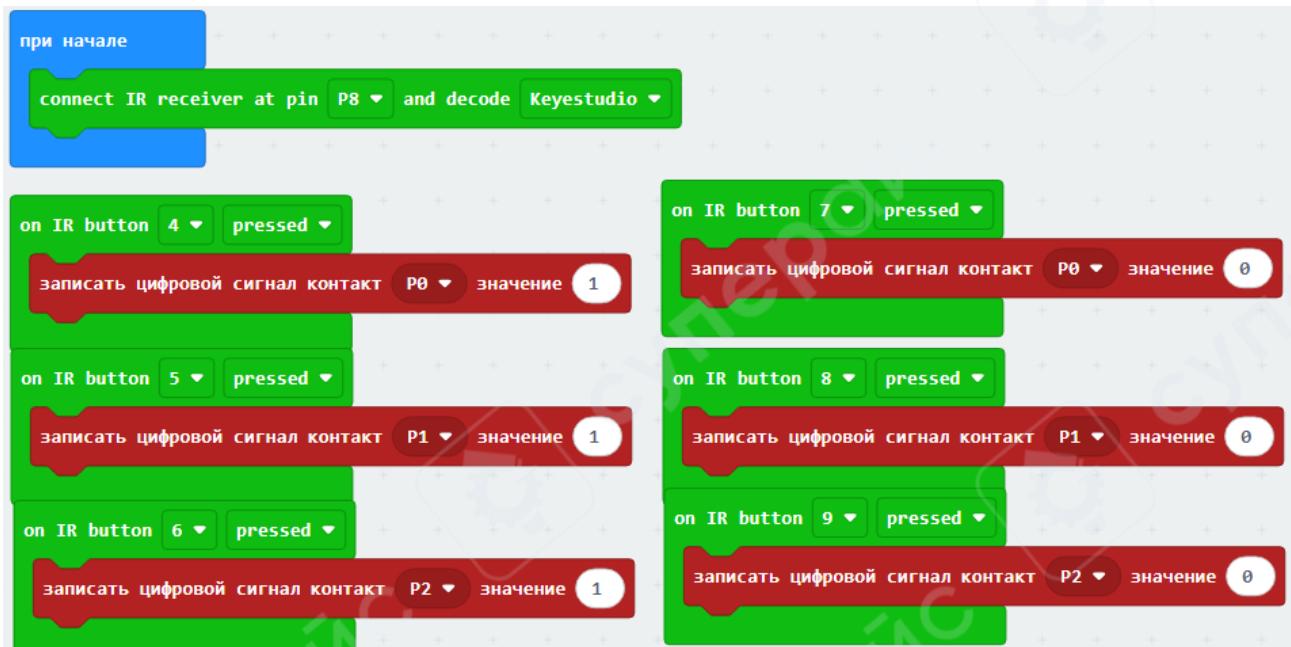
Группа 3: Синий цвет (P2)

- **Включение:**
- Блок события on IR button pressed, кнопка: **6**.
- Действие: digital write pin P2 to 1.
- **Выключение:**
- Блок события on IR button pressed, кнопка: **9**.
- Действие: digital write pin P2 to 0.

Результат: После загрузки программы направьте пульт на ИК-приемник.

- Нажатие кнопки **4** зажмет красный свет, кнопка **7** выключит его.
- Нажатие кнопки **5** зажмет зеленый свет, кнопка **8** выключит его.
- Нажатие кнопки **6** зажмет синий свет, кнопка **9** выключит его.
- Вы можете смешивать цвета, включая их одновременно (например, Красный + Синий = Фиолетовый).

Файлы: Файл урока: 红外遥控实验.hex



2.37 Урок: Сенсорный переключатель

Цель: Создать устройство, которое включает светодиод при касании пальцем сенсорной площадки и выключает его, когда вы убираете палец.

Принцип работы: В этом уроке используется **емкостный сенсорный датчик** (на базе чипа TTP223, см. схему). В отличие от обычной механической кнопки, здесь нет подвижных частей. Когда вы касаетесь круглой площадки, датчик реагирует на статическое электричество и емкость вашего тела. При касании он отправляет высокий электрический сигнал (1) на микроконтроллер. Программа считывает этот сигнал и включает светодиод.

Необходимое оборудование:

1. Плата BBC micro:bit.
2. Плата расширения GPIO.
3. Модуль сенсорного датчика (Touch Sensor, синий).
4. Модуль светодиода (LED Module, красный/черный).
5. Соединительные провода.

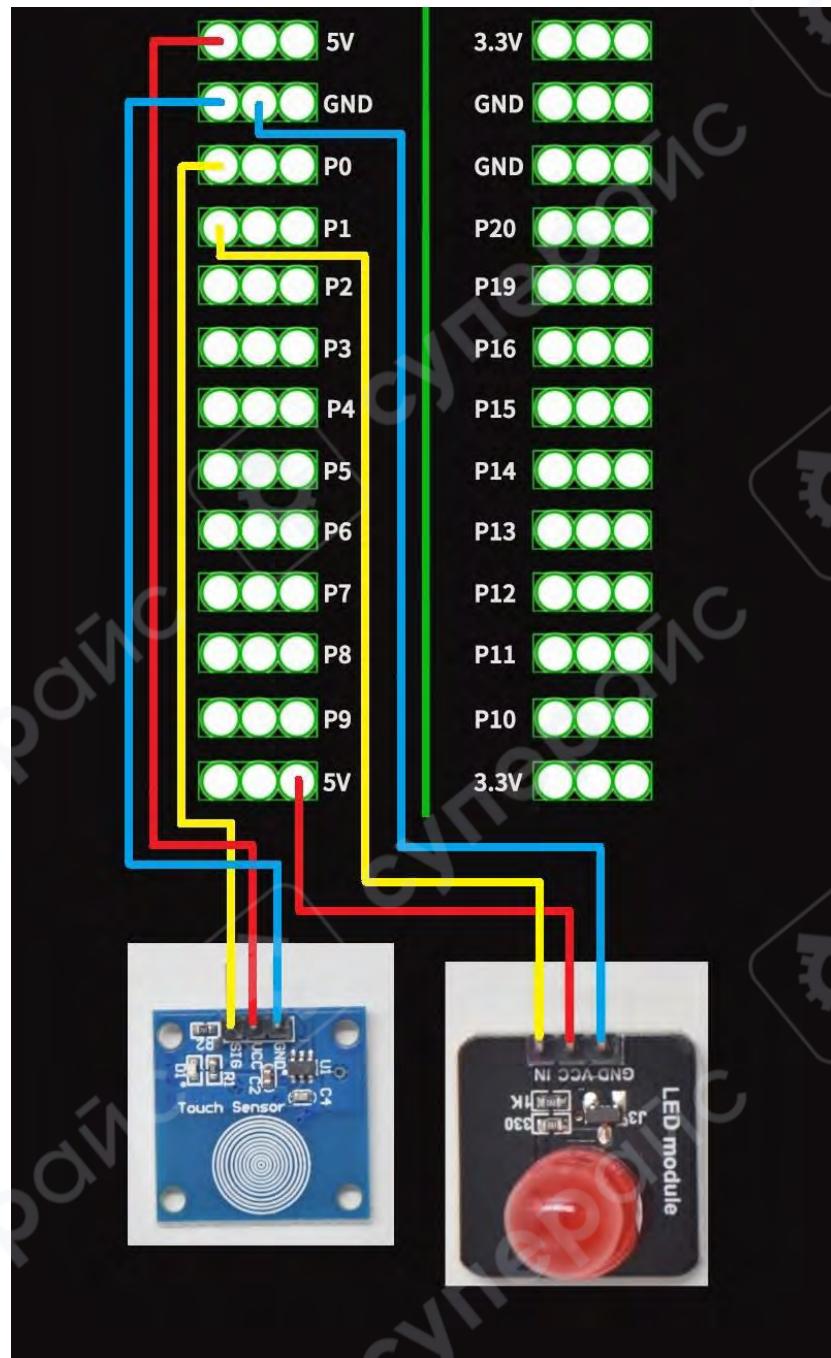
Схема подключения: Будьте внимательны с проводами питания (VCC и GND), чтобы не повредить датчики.

1. **Сенсорный датчик (Touch Sensor):**

GND (Минус) → GND (синий провод).
 VCC (Плюс) → 5V (красный провод).
 SIG (Сигнал) → P0 (желтый провод).

2. **Светодиодный модуль (LED):**

GND / IN- → GND (синий провод).
 VCC / IN+ → 5V (красный провод).
 IN (Сигнал) → P1 (желтый провод).



Подключение сенсорного датчика и светодиода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте пустым.

Блок «Постоянно» (Forever): Здесь мы используем условную конструкцию, чтобы проверять состояние датчика в реальном времени.

1. Зайдите в раздел **Logic** (Логика) и выберите блок **if ... then ... else** (если ... то ... иначе). Поместите его внутрь блока **Forever**.

2. **Условие (If):**

- В шестиугольное поле условия вставьте блок считывания сенсора.

- В расширении это блок: Touch sensor pin P0 return touched (Сенсор на пине P0 возвращает касание).

3. **Действие «To» (Then) — Включение:**

- Вставьте блок digital write pin P1 to 1 (записать в пин P1 значение 1). Это подаст напряжение на светодиод.

4. **Действие «Иначе» (Else) — Выключение:**

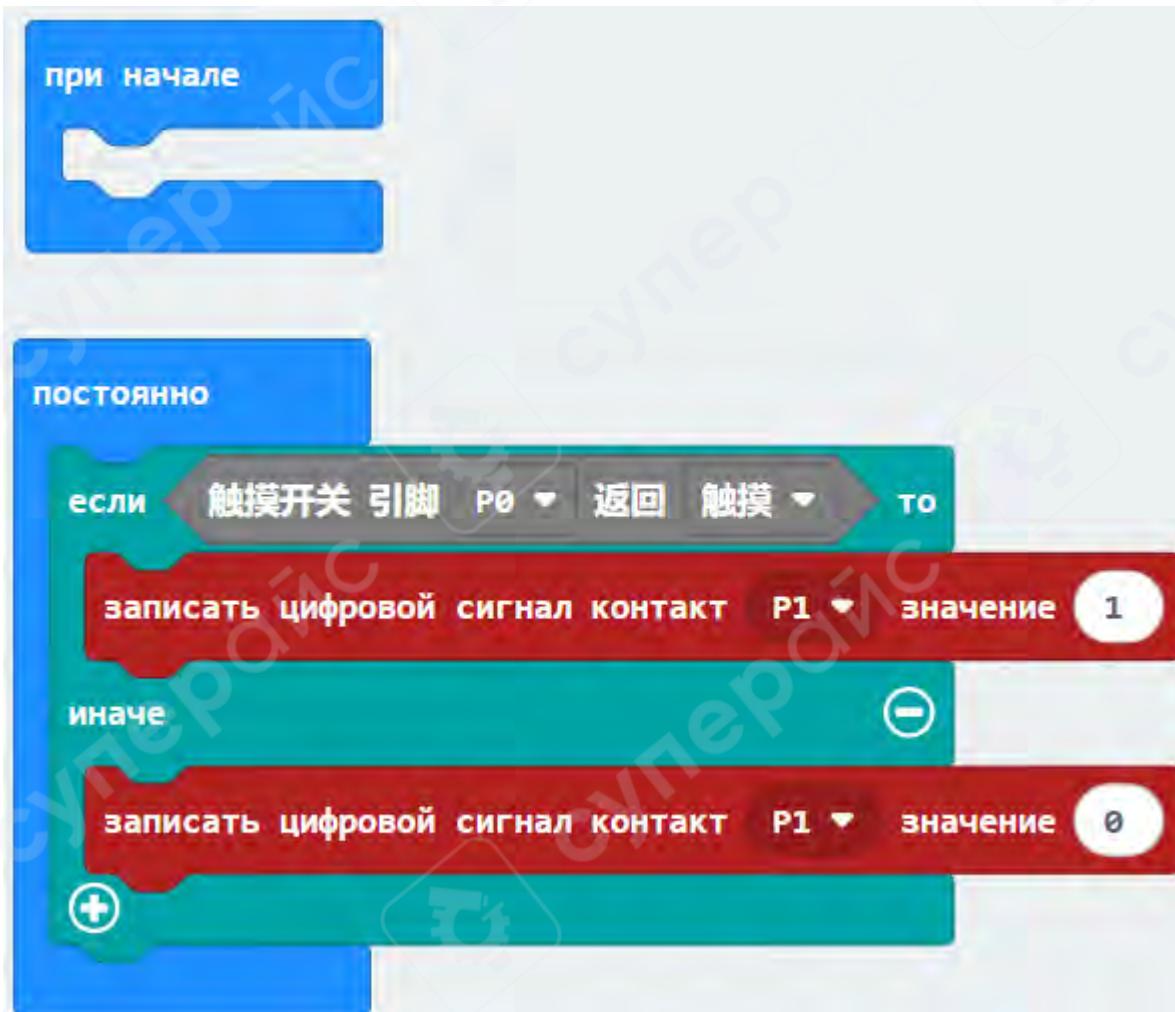
- Вставьте блок digital write pin P1 to 0 (записать в пин P1 значение 0). Это погасит светодиод, если касания нет.

Результат: После загрузки программы:

1. Коснитесь пальцем круглой зоны на синем сенсорном датчике — красный светодиод загорится.

2. Уберите палец — светодиод мгновенно погаснет.

Файлы: Файл урока: 触摸开关实验.hex



2.38 Урок: Ультразвуковой дальномер (Измерение расстояния)

Цель: Создать устройство, которое измеряет расстояние до ближайшего объекта в сантиметрах и отображает полученное значение на светодиодном дисплее micro:bit.

Принцип работы: Используется ультразвуковой датчик HC-SR04. Принцип его работы похож на эхолокацию летучих мышей:

- Датчик отправляет короткий ультразвуковой импульс через излучатель (контакт **Trig**).
- Звуковая волна отражается от препятствия и возвращается обратно.
- Приемник (контакт **Echo**) улавливает отраженный сигнал.
- Micro:bit замеряет время, прошедшее между отправкой и возвращением сигнала, и на основе скорости звука вычисляет точное расстояние.

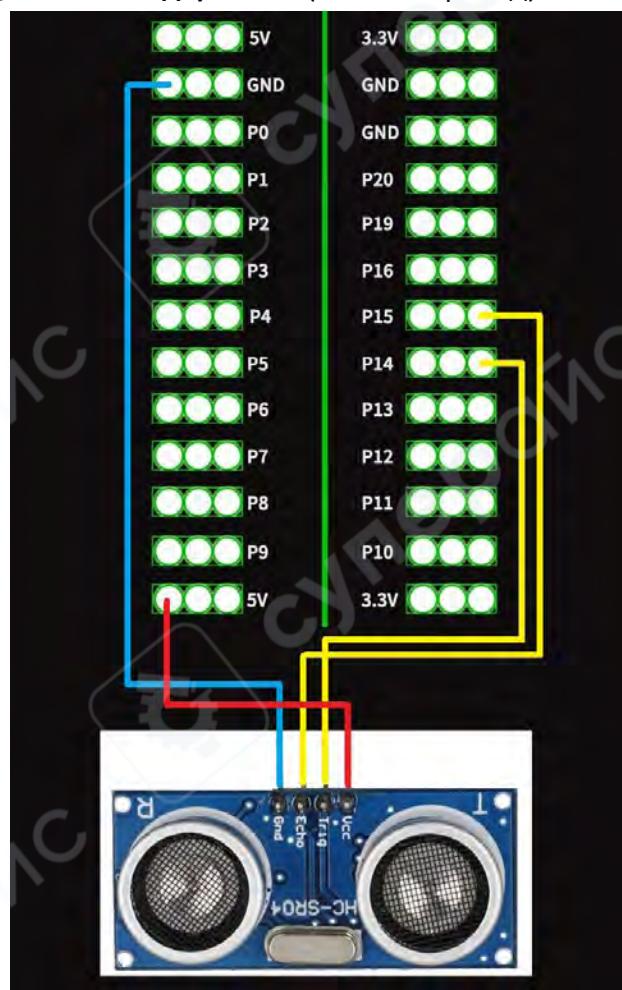
Необходимое оборудование:

- Плата BBC micro:bit.
- Плата расширения GPIO.
- Ультразвуковой датчик HC-SR04 (выглядит как два "глаза").
- Соединительные провода.

Схема подключения: Внимание: Датчик подключается к двум разным цифровым pinам (один для отправки, другой для приема).

1. Ультразвуковой датчик (HC-SR04):

- Vcc (Плюс) → 5V (красный провод). Важно подключить именно к 5V для стабильной работы.
- Gnd (Минус) → GND (синий провод).
- Trig (Триггер/Сигнал выхода) → P14 (желтый провод).
- Echo (Эхо/Сигнал входа) → P15 (желтый провод).



Подключение ультразвукового дальномера

Программный код (сборка блоков):

Примечание: Для работы с этим датчиком в редакторе MakeCode необходимо добавить расширение. Нажмите «Расширения» (Extensions) и введите в поиске «HC-SR04».

Выберите пакет sonar.



Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever): Мы будем непрерывно измерять расстояние и выводить его на экран.

1. Вывод на экран:

- Зайдите в меню **Basic** (Основное).
- Выберите блок **show number** (показать число).
- Перетащите его внутрь блока **Forever**.

2. Считывание данных (Sonar):

- Зайдите в меню **Sonar**
- Выберите блок **ping trig ... echo ... unit**
- Вставьте этот блок внутрь блока **show number** (вместо цифры 0).

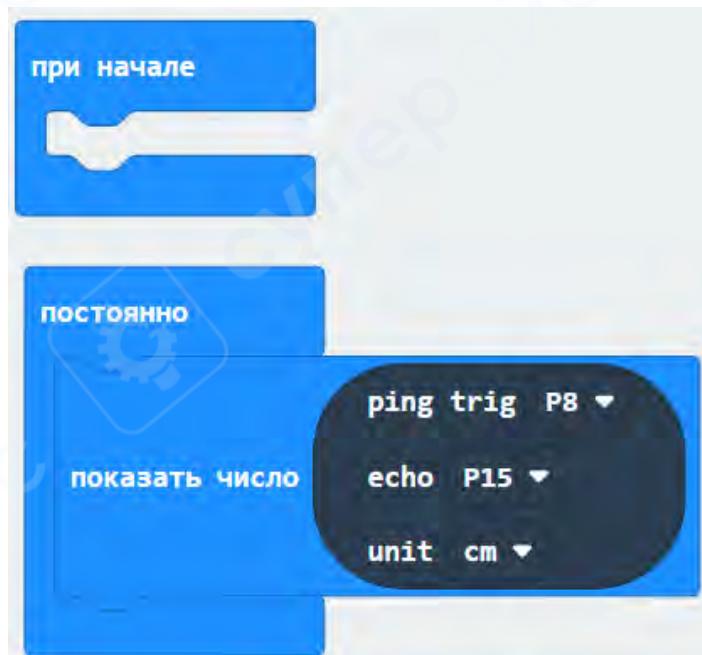
3. Настройка параметров:

- В выпадающем списке **trig** (триггер) выберите pin: **P14**.
- В выпадающем списке **echo** (эхо) выберите pin: **P15**.
- В поле **unit** (единицы измерения) выберите: **cm** (сантиметры).

Результат: После загрузки программы:

1. Направьте датчик на любой предмет (стену, руку, коробку).
2. На дисплее micro:bit будут бежать цифры, показывающие расстояние в сантиметрах.
3. Если приблизить предмет к датчику, число уменьшится. Если отдалить — увеличится.

Файлы: Файл урока: 超声波测距.hex.



2.39 Урок: Жидкокристаллический дисплей I2C 1602

Цель: Научиться подключать внешний текстовый экран к micro:bit, используя всего два сигнальных провода, и выводить на него текстовую информацию (буквы и цифры).

Принцип работы: Стандартная матрица micro:bit (5x5) не подходит для вывода длинных сообщений. Дисплей **LCD1602** может отображать **16 символов в 2 строки**. Для подключения используется протокол **I2C** (Inter-Integrated Circuit). Это позволяет управлять экраном, используя только два пина (SDA и SCL) на плате расширения, вместо того чтобы занимать 6-8 портов при прямом подключении.

- **SDA (Serial Data):** Линия передачи данных (Пин P20).
- **SCL (Serial Clock):** Линия тактирования (Пин P19).

Необходимое оборудование:

1. Плата BBC micro:bit.
2. Плата расширения GPIO.
3. Дисплейный модуль LCD1602 с I2C адаптером (черная плата сзади экрана).
4. Соединительные провода (4 шт).

Схема подключения: Внимание: Протокол I2C на micro:bit жестко привязан к пинам P19 и P20. Подключать к другим пинам нельзя.

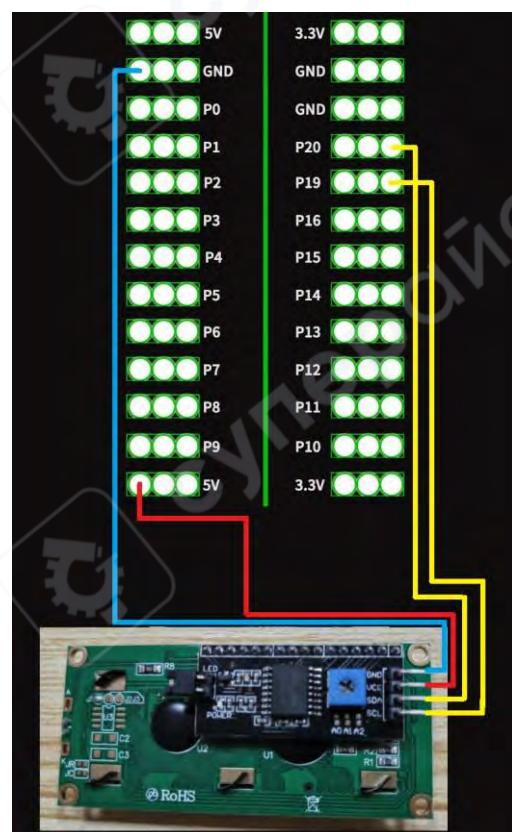
1. **LCD1602 (I2C адаптер):**

GND (Минус) → GND (синий провод).

VCC (Плюс) → 5V (красный провод). Обязательно 5V для четкости символов.

SDA (Данные) → P20 (желтый провод).

SCL (Часы) → P19 (желтый провод).



Подключение ЖК дисплея

Программный код (сборка блоков):

Примечание: Для работы требуется расширение. В меню «Расширения» (Extensions) введите «1602». Выберите пакет makerbit-lcd1602.



Блок «При начале» (On Start): Инициализация дисплея.

1. Выберите блок connect LCD at I2C address....

2. Установите адрес: **39**.

Важно: Адрес устройства зависит от чипа. Если 39 не сработает, попробуйте изменить на 63 (это десятичные значения для шестнадцатеричных адресов 0x27 и 0x3F). В данном уроке используется **39**.

Блок «Постоянно» (Forever): Вывод текста на экран.

1. **Строка 1:**

- Используйте блок LCD1602 show string ... at position
- Впишите текст: "microbit".
- Позиция (position): **1**. (Начало первой строки).
- Длина (length): **16**.

2. **Строка 2:**

- Используйте такой же блок.
- Впишите текст: например "Hello World").
- Позиция (position): **17**.

Пояснение: Поскольку в первой строке 16 символов, позиция 17 является началом второй строки.

Результат: После загрузки программы на синем экране появятся две строчки текста:

1. microbit
2. Hello World

Файлы: Файл урока: I2C-1602 显示频.hex



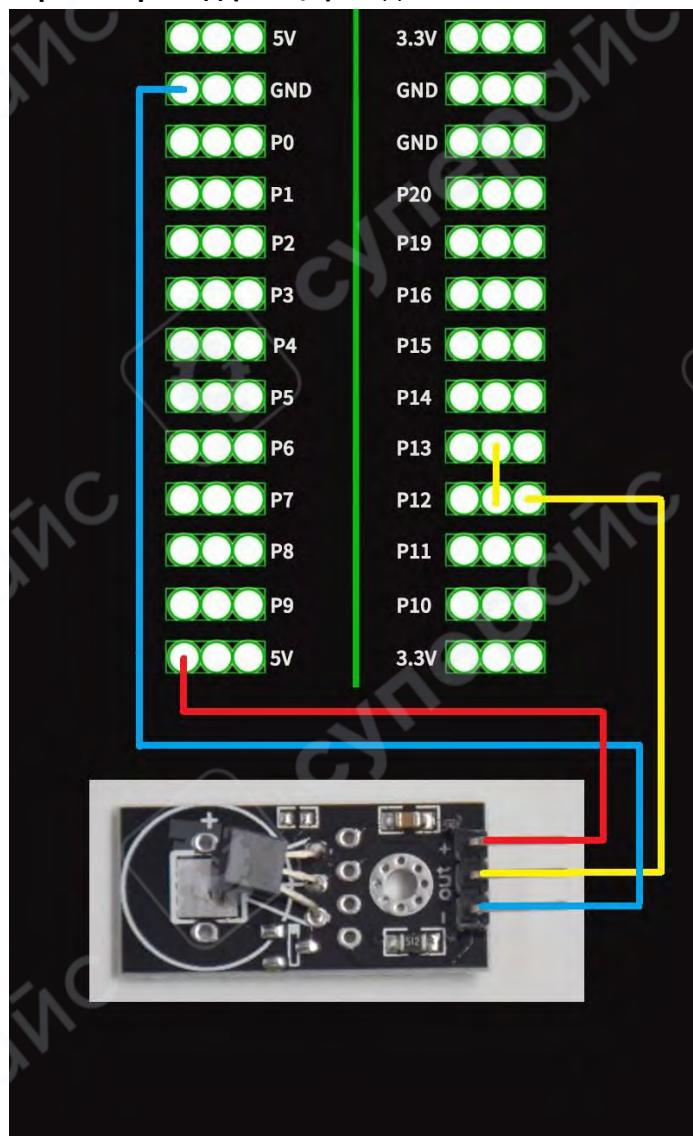
2.40 Урок: Эксперимент с датчиком температуры DS18B20

Цель: Создать устройство для измерения температуры с использованием выносного герметичного датчика DS18B20. Это позволяет измерять температуру жидкостей или почвы, что невозможно сделать встроенным датчиком micro:bit.

Принцип работы: Датчик DS18B20 — это цифровой термометр, который обменивается данными по протоколу 1-Wire. В отличие от аналоговых датчиков, он передает точное цифровое значение. В программе используется специальный блок для работы с этим датчиком («weather monitoring» или «Smart City»), который считывает данные. Полученное значение часто приходит умноженным на 100 (например, 2500 для 25.0°C), поэтому в коде применяется математическая операция деления.

Схема подключения: Для подключения используется плата расширения и провод-перемычки.

- **Желтый провод (Signal/Data):** подключается к порту P13.
- **Красный провод (VCC/+):** подключается к 5V.
- **Синий/Черный провод (GND/-):** подключается к GND.



Подключение датчика температуры

Программный код (сборка блоков):

Загрузка расширения: Для работы блоков требуется установить расширение. Найдите

DS18B20 и выберите библиотеку Weather-bit.



1. Блок «При начале» (On Start):

- Добавьте блок инициализации датчика: start weather monitoring (начать мониторинг погоды).

2. Блок «Постоянно» (Forever):

- Используйте блок show number (показать число).
- Внутрь поместите математический блок деления (/).

В левую часть деления поместите блок считывания сенсора: soil temperature(C) (температура почвы/датчика).

- В правую часть деления введите число 100.

Пояснение: Деление на 100 необходимо для перевода «сырых» данных в привычные градусы Цельсия.

Результат: После загрузки программы micro:bit будет непрерывно отображать температуру с внешнего щупа. Если опустить датчик в теплую воду или зажать в руке, цифры на дисплее будут увеличиваться.

Файлы: Файл урока: DS18B20 温度实验.hex.



2.41 Урок: Эксперимент с поворотным энкодером

Цель: Научиться использовать поворотный энкодер (ручку-крутилку) для изменения цифрового значения (переменной) от 0 до 100 и сброса значения нажатием на кнопку.

Принцип работы: Энкодер — это электромеханическое устройство, преобразующее угол поворота в цифровой сигнал. В отличие от потенциометра, он может вращаться бесконечно. Программа отслеживает направление вращения (влево или вправо) и соответственно уменьшает или увеличивает значение переменной. Также реализована защита от выхода за пределы диапазона (0–100).

Схема подключения: Согласно схеме на фото, энкодер подключается к плате расширения следующим образом:

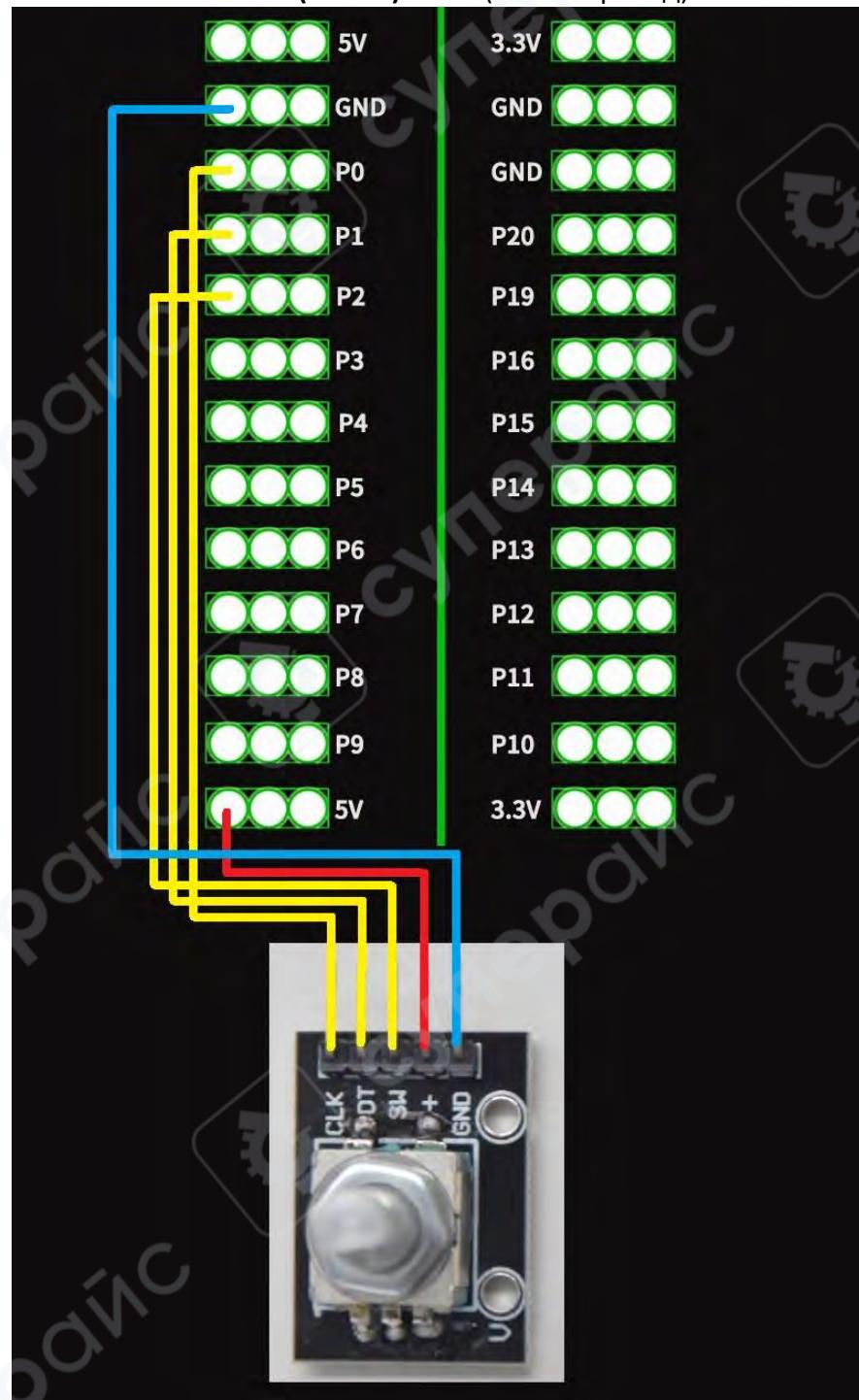
CLK (S1): порт P0 (Желтый провод).

DT (S2): порт P1 (Желтый провод).

SW (Кнопка): порт P2 (Желтый провод).

+ (Питание): 5V (Красный провод).

GND (Земля): GND (Синий провод).



Подключение энкодера

Программный код (сборка блоков):

Загрузка расширения: Для работы блоков требуется установить расширение. Найдите <https://github.com/zhuning239/ky04> и выберите библиотеку ky040.



1. Настройка (On Start):

- Создайте переменную num и установите ее значение в 100 (set num to 100).
- Инициализируйте энкодер специальным блоком: connect clk P0 dt P1 sw P2.

2. Вращение влево (on rotated Left):

- Добавьте условие «Если» (if): Если num ≤ 0.

То: Установить num в 0 (защита от отрицательных чисел).

Иначе (else): Изменить num на -1 (уменьшить значение).

3. Вращение вправо (on rotated Right):

- Добавьте условие «Если» (if): Если num > 100.

То: Установить num в 100 (защита от превышения максимума).

Иначе (else): Изменить num на +1 (увеличить значение).

4. Нажатие кнопки (on button pressed):

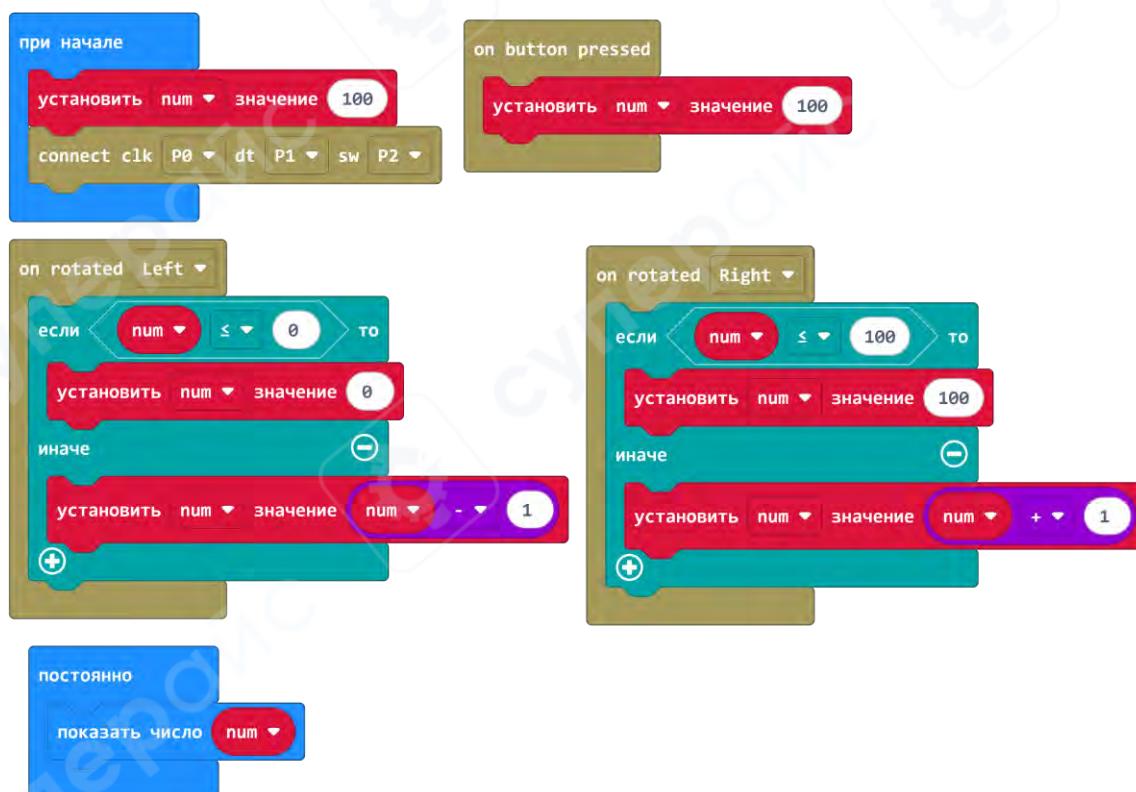
- При событии нажатия кнопки на энкодере: Установить num в 100 (сброс значения к начальному).

5. Отображение (Forever):

- В блоке «Постоянно» добавьте show number (показать число) и вставьте туда переменную num.

Результат: При вращении ручки энкодера вправо числа на экране micro:bit будут увеличиваться (до 100). При вращении влево — уменьшаться (до 0). При нажатии на ручку (как на кнопку) значение сбросится на 100.

Файлы: Файл урока microbit-旋转编码器传感器.hex и 编码器.hex



2.42 Урок: Эксперимент по управлению сервоприводом

Цель: Научиться управлять углом поворота сервомотора с помощью поворотной ручки (потенциометра).

Принцип работы: Сервопривод принимает команды угла поворота от 0 до 180 градусов. Потенциометр при вращении выдает аналоговый сигнал от 0 до 1023. Чтобы управлять мотором, нужно преобразовать (смаппить) значения с ручки (0–1023) в понятные для мотора углы (0–180).

Схема подключения:

- Потенциометр (Ручка):

Желтый провод (S) → Пин P0.

Красный провод (VCC) → Питание 3,3V.

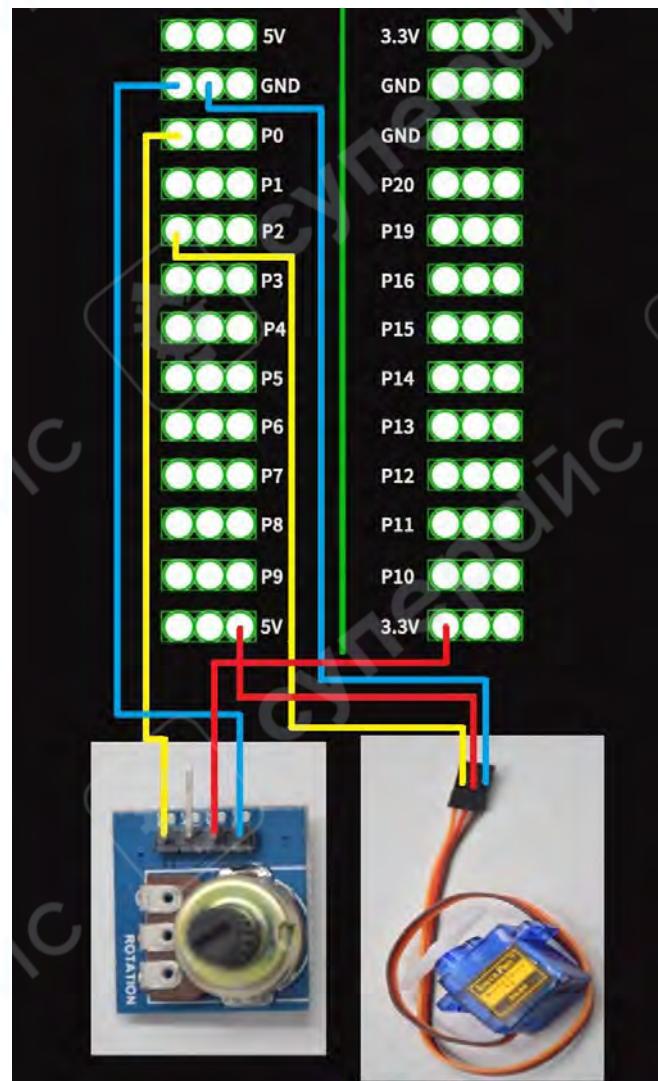
Черный провод (GND) → Земля GND.

Сервопривод (Мотор):

Оранжевый/Желтый провод (S) → Пин P2.

Красный провод (+) → Питание 5V.

Коричневый провод (-) → Земля GND.



Подключение потенциометра и сервопривода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

- Этот блок будет непрерывно считывать положение ручки и обновлять положение мотора.

Шаг 1: Команда мотору

- Зайдите в меню Pins (Пины).
- Выберите красный блок servo write pin P0 to 180. • Перетащите его в блок Forever.
- Измените пин P0 на P2 (так как мотор подключен к P2).

Шаг 2: Преобразование значений (Mapping)

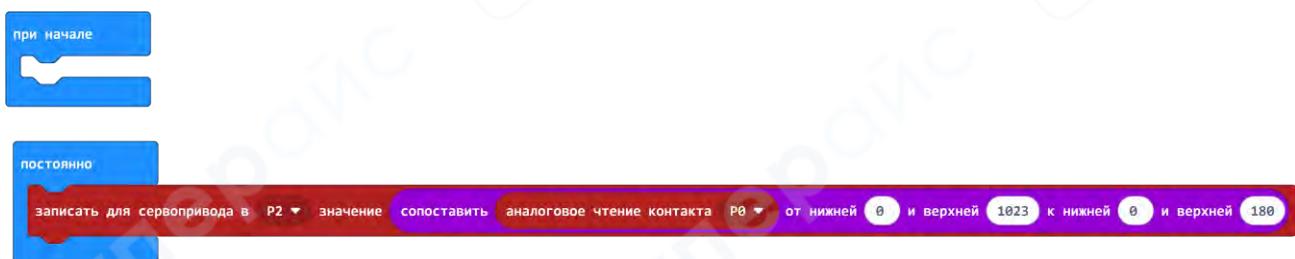
- Зайдите в меню Математика.
- Найдите фиолетовый блок map value... from low....
- Вставьте этот блок в поле, где написано число 180 (в блоке сервопривода).
- Настройте диапазоны в блоке map:
 - from low (от мин): 0
 - from high (от макс): 1023
 - to low (до мин): 0
 - to high (до макс): 180

Шаг 3: Считывание ручки

- Зайдите в меню Pins.
- Возьмите овальный блок analog read pin P0.
- Вставьте его в первое поле блока map (вместо value).

Результат: При вращении ручки потенциометра вал сервомотора будет синхронно поворачиваться на соответствующий угол.

Файлы: Файл урока: 单独舵机控制实验.hex и 舵机控制实验.hex



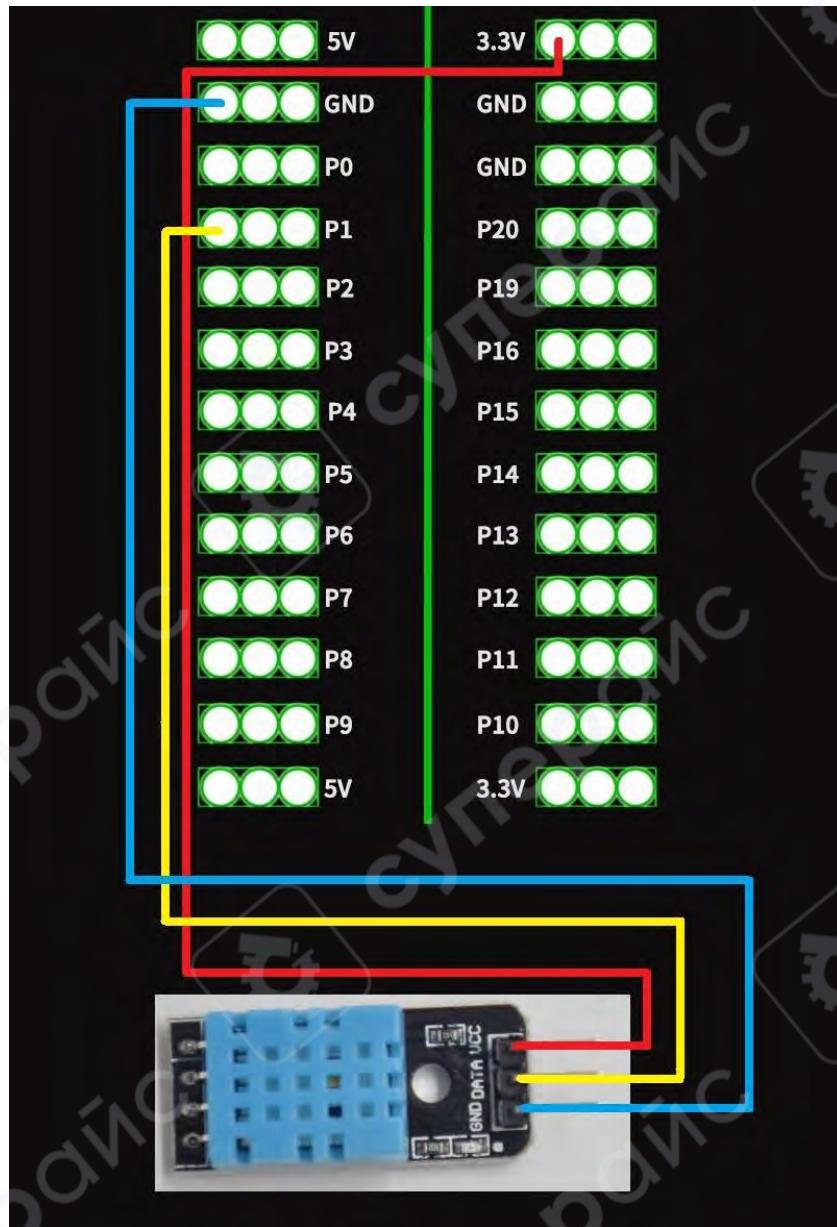
2.43 Урок: Эксперимент с датчиком влажности DHT11

Цель: Подключить внешний цифровой датчик DHT11 и вывести показания температуры и влажности на дисплей micro:bit.

Принцип работы: Датчик DHT11 — это комбинированный модуль, измеряющий и температуру, и влажность воздуха. В отличие от аналоговых датчиков, он передает данные в цифровом виде, поэтому для работы с ним требуется установка специального расширения (библиотеки) в программу.

Схема подключения: • Датчик DHT11 (синий):

- Желтый провод (DATA) → Пин P1 (желтый ряд на плате расширения).
- Красный провод (VCC) → Питание 3.3V (красный ряд).
- Черный провод (GND) → Земля GND (черный ряд).



Подключение датчика DHT11

Программный код (сборка блоков):

Подготовка (Установка расширения):

В меню блоков нажмите «Расширения». Введите в поиске https://github.com/alankrantas/pxt-DHT11_DHT22. Выберите пакет | DHT11/DHT22

Блок «При начале» (On Start):

- Зайдите в меню DHT11 (оранжевые блоки).
- Выберите блок инициализации (Query DHT11 / Pin pull up).
- Перетащите его внутрь блока On Start.
- Установите настройки: Data pin на P0, Pin pull up на true, Wait 2 sec на true.

Блок «Постоянно» (Forever):

- Мы используем этот блок для постоянного обновления данных на экране.

Шаг 1: Вывод числа

- Зайдите в меню Basic (Основное).
- Выберите блок show number (показать число).
- Перетащите его внутрь блока Forever.

Шаг 2: Получение данных

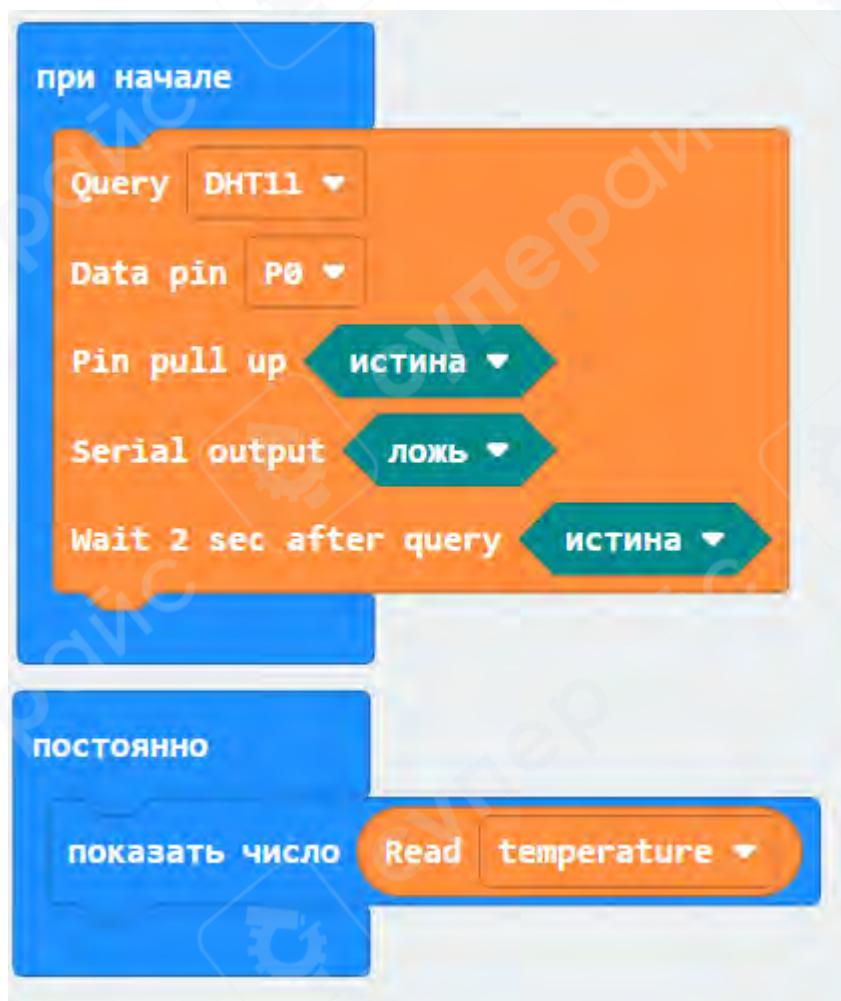
- Зайдите в меню DHT11.
- Найдите овальный блок Read temperature (считать температуру) или Read humidity (считать влажность).

• Перетащите этот овальный блок внутрь блока show number, вместо цифры 0.

Результат: После загрузки программы на матрице начнут бежать цифры, показывающие текущую температуру в комнате. Если подышать на датчик, показания изменятся.

Примечание: Обратите внимание, что на схеме (фото) желтый провод подключен к P1, но в программе выбран P0. Подключайте провод строго к **P0**, чтобы код сработал.

Файлы: Файл урока: DHT11t.hex



2.44 Урок: ИК-сигнализация обнаружения препятствий

Цель: Создать устройство безопасности, которое автоматически включает сигнальный светодиод, когда инфракрасный датчик обнаруживает препятствие.

Принцип работы:

Система использует инфракрасный датчик для «видения» объектов и внешний светодиодный модуль для сигнализации.

- **Датчик:** Подключен к контакту **P6**. Когда перед ним есть препятствие, он посыпает сигнал 0. Когда путь свободен — сигнал 1.
- **Светодиод:** Подключен к контакту **P7**. Программа подает на него питание (сигнал 1), когда датчик видит препятствие.

Важно: Для корректной работы внешних устройств мы отключаем встроенный экран micro:bit, чтобы освободить ресурсы питания и портов.

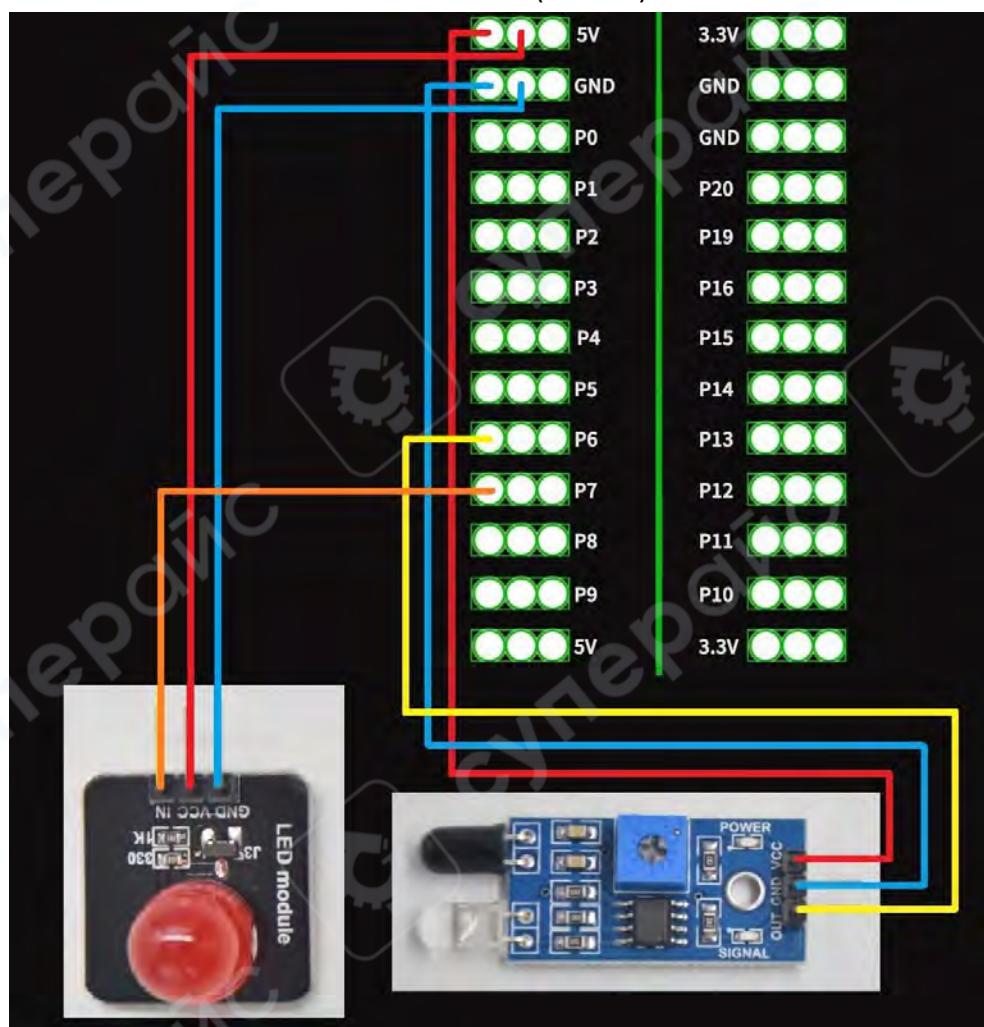
Схема подключения: Используйте плату расширения и провода.

- **ИК-датчик препятствий (IR Sensor):**

VCC (+) → **5V**
GND (-) → **GND**
OUT (Сигнал) → **P6**

- **Светодиодный модуль (LED):**

VCC (+) → **5V**
GND (-) → **GND**
IN (Сигнал) → **P7**



Подключение ИК датчика и светодиода

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Этот блок настраивает плату перед запуском основного цикла. Здесь мы отключаем встроенную светодиодную матрицу 5x5.

Блок «Постоянно» (Forever):

- В этом блоке происходит непрерывная проверка датчика. Micro:bit постоянно спрашивает: «Есть ли препятствие?» и мгновенно включает или выключает свет.

Шаг 1: Отключение матрицы

- Зайдите в меню **Led** (Светодиоды), раздел **More** (Дополнительно).
- Выберите блок **led enable** (светодиоды включить).
- Перетащите его в блок «При начале» и установите значение **false** (ложь).

Шаг 2: Логика проверки

- Зайдите в меню **Logic** (Логика).
- Выберите блок **if ... then ... else** (если ... то ... иначе).
- Перетащите его внутрь блока «Постоянно».

Шаг 3: Чтение датчика

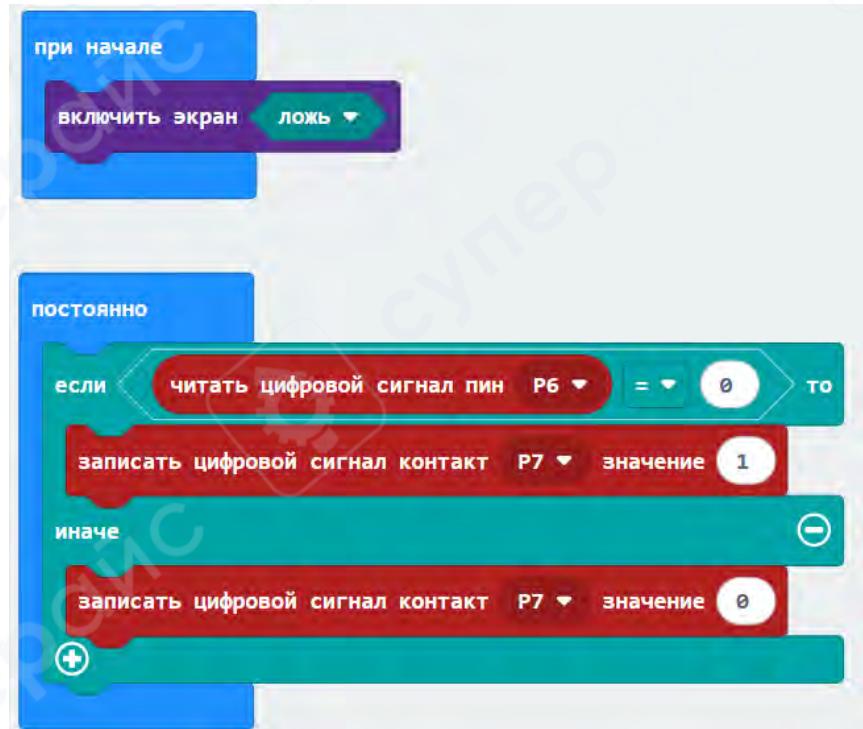
- В условие «если» (шестиугольное поле) вставьте блок сравнения (=).
- В левую часть сравнения поместите блок **digital read pin** (цифровое чтение контакта) из меню **Pins** (Контакты) и выберите **P6**.
- В правую часть сравнения впишите число **0**.

Шаг 4: Управление светом

- В раздел «то» (then) поместите блок **digital write pin** (цифровая запись контакта) из меню **Pins**. Выберите **P7** и значение **1**.
- В раздел «иначе» (else) поместите такой же блок для **P7**, но со значением **0**.

Результат: После загрузки программы, если поднести руку или предмет к инфракрасному датчику, загорится внешний красный светодиод. Если убрать препятствие, светодиод погаснет.

Файлы: Файл урока: **红外避障报警器 .hex**.



2.45 Урок: Эксперимент с OLED дисплеем

Цель: Научиться выводить текстовую информацию на внешний OLED-экран, используя протокол связи I2C.

Принцип работы: OLED-дисплей подключается к micro:bit по шине I2C. Это позволяет передавать сложные данные (текст, графику) всего по двум проводам. Для управления экраном используется специальная библиотека.

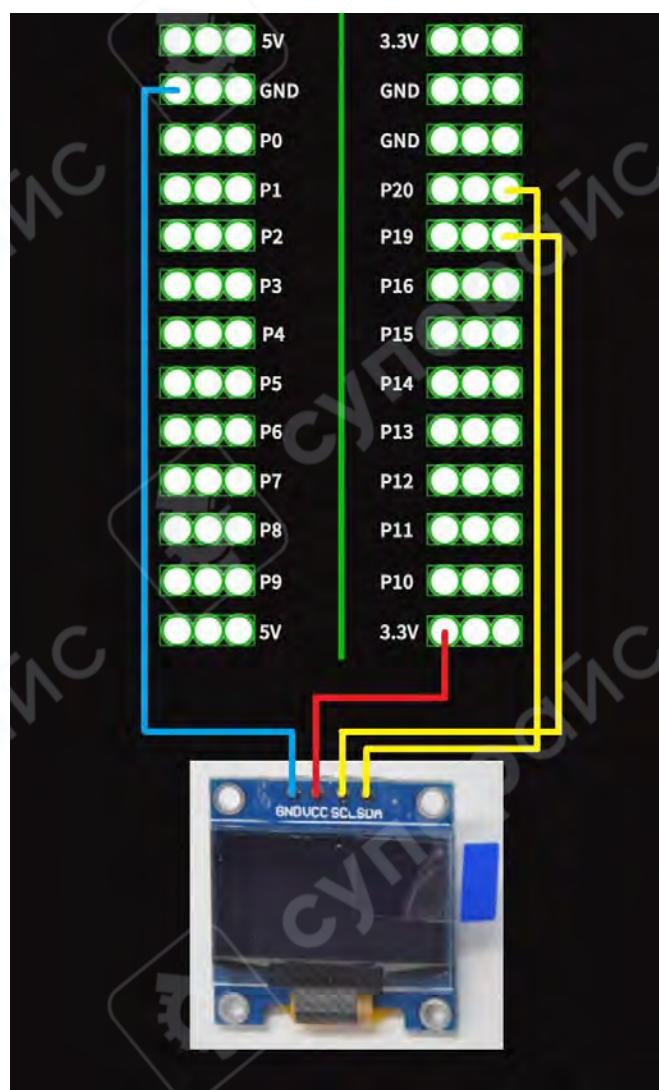
Схема подключения: Подключите дисплей к контактам I2C на плате расширения.

VCC → 3.3V (Внимание: не подключайте к 5V, это может повредить экран).

GND → GND

SCL (Линия тактирования) → P19

SDA (Линия данных) → P20



Подключение OLED дисплея

Программный код (сборка блоков):

Блок «При начале» (On Start):

- Инициализация дисплея. Экран запускается один раз при включении.

Блок «Постоянно» (Forever):

- Отображение информации на экране.

Шаг 1: Добавление библиотеки

- В редакторе нажмите «Расширения». Найдите **OLED (SSD1306)** и добавьте его.

 **OLED12864_I2C**

Шаг 2: Инициализация

- Зайдите в меню **OLED**.
- Выберите блок **init OLED with addr** (инициализировать OLED с адресом).
- Перетащите его в блок «При начале». Установите адрес **60**.

Шаг 3: Вывод первой строки

- Зайдите в меню **OLED**.
- Выберите блок **show string at** (показать строку в).
- Перетащите его в блок «Постоянно».
- Установите: x: **0**, y: **0**, текст: "Hello!", цвет: **1**.

Шаг 4: Вывод второй строки

- Добавьте еще один блок **show string at** ниже.
- Установите: x: **0**, y: **1**, текст: "world!", цвет: **1**.

Результат: После включения на OLED-дисплее отобразятся две строки текста: "Hello!" и "world!".

Файлы: Файл урока: OLED.hex.



2.46 Урок: Измерение атмосферного давления (BMP280)

Цель: Создать простую метеостанцию, которая считывает данные о температуре и атмосферном давлении с датчика BMP280 и отображает их на OLED-дисплее.

Принцип работы:

- **BMP280:** Это точный датчик, измеряющий температуру и атмосферное давление. Он передает данные в цифровом виде по шине I2C.
 - **Шина I2C:** Позволяет подключать несколько устройств (экран и датчик) к одним и тем же двум проводам (SCL и SDA). Процессор различает их по уникальным адресам (адрес экрана — 60, адрес датчика — 0x76).
 - **Программа:** Считывает «сырые» данные с датчика, округляет значение давления до целого числа и выводит всё на экран.

Схема подключения: Вам нужно подключить **оба** устройства (Датчик и Экран) к плате.

1. Датчик BMP280 (фиолетовая плата):

VCC → 3.3V (Не подключайте к 5V!)

GND → GND

SCL → P19

SDA → P20

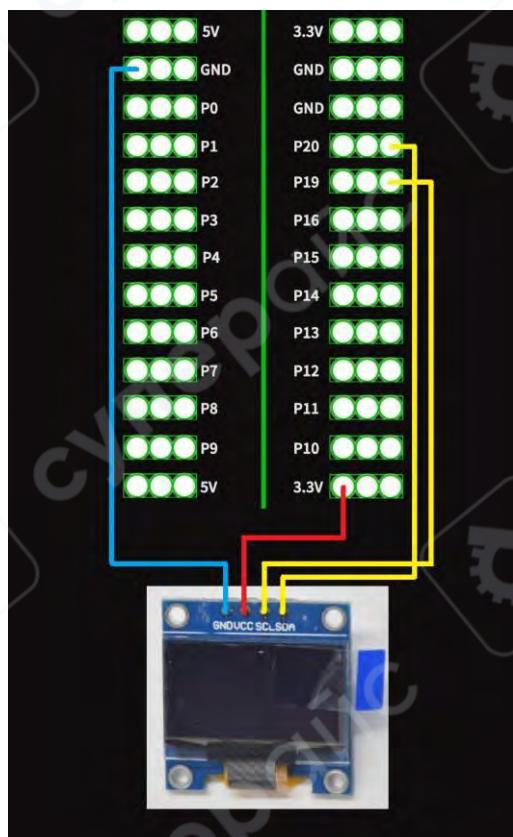
2. OLED дисплей:

VCC → 3.3V

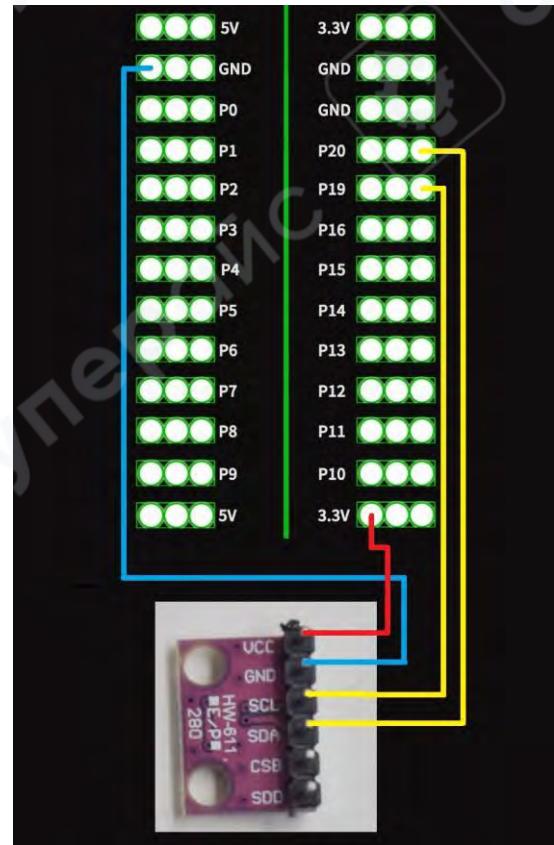
GND → GND

SCL → P19

SDA → P20



Подключение дисплея



Подключение BMP280

Программный код (сборка блоков):

Для работы потребуется установить расширение для BMP280. Нажмите **Extensions**. Введите в поиск <https://github.com/zhuning239/BMP280>. Выберите библиотеку. Появятся блоки голубого цвета.

Блок «При начале» (On Start):

- Здесь мы инициализируем (запускаем) оба устройства.

Блок «Постоянно» (Forever):

- Цикл опроса датчика и обновления экрана.

Шаг 1: Инициализация

- Из меню **OLED** вставьте блок **init OLED with addr 60**.
- Из меню **BMP280** (голубые блоки) вставьте блок **set address** (установить адрес).

Убедитесь, что выбрано значение **0x76** (стандартный адрес для модуля HW-611).

Шаг 2: Вывод температуры (Строка 0)

- Из меню **OLED** выберите **show a Number at** (показать число в).
- Координаты: x: **0**, y: **0**.
- В поле **number** вставьте блок **get temperature** (получить температуру) из меню **BMP280**.

- Цвет: **1**.

Шаг 3: Вывод давления (Строка 1)

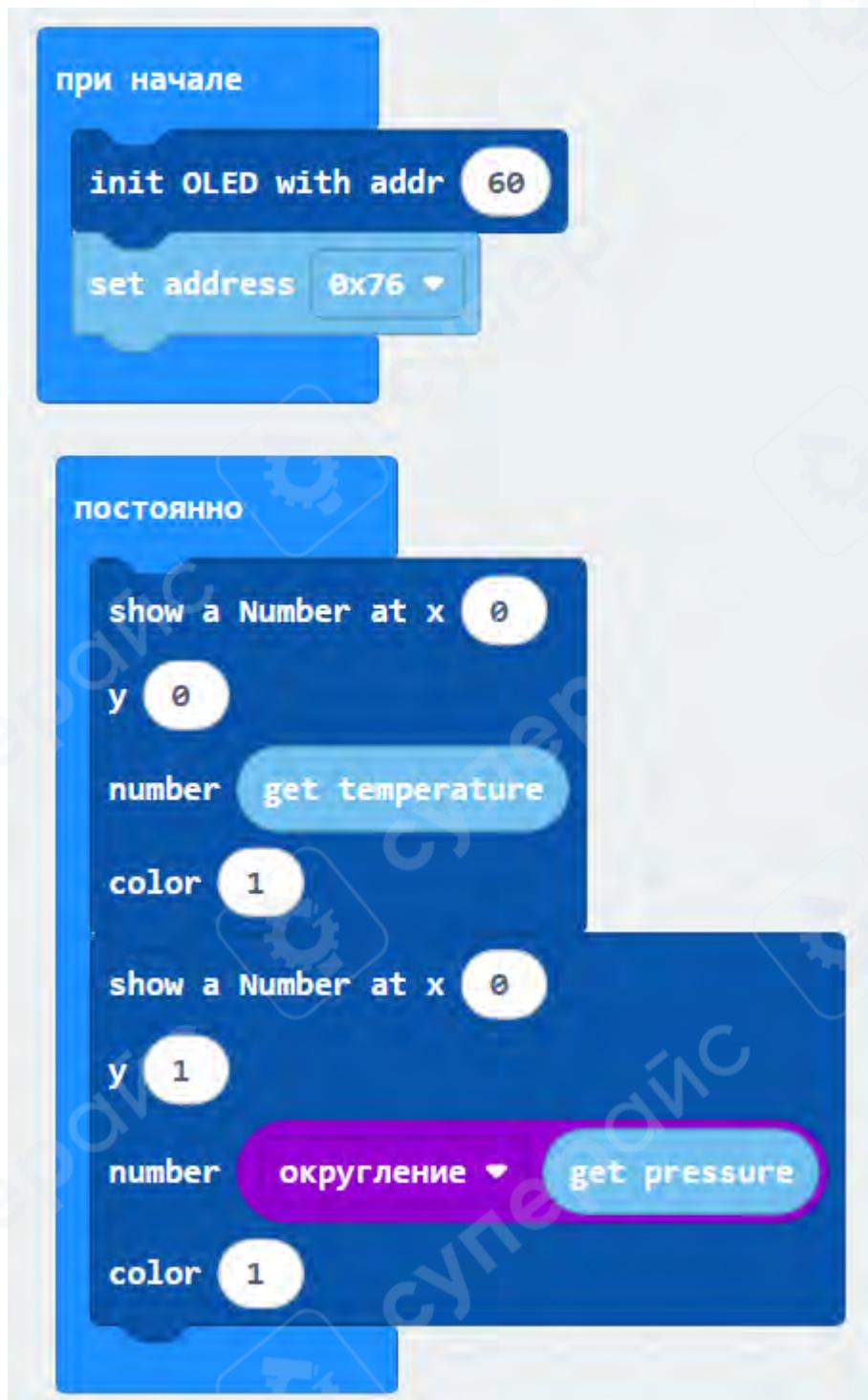
- Добавьте еще один блок **show a Number at**.
- Координаты: x: **0**, y: **1**.
- **Обработка числа:** Давление измеряется в Паскалях (Pa) и число может быть очень длинным. Чтобы оно красиво смотрелось на экране, мы его округлим.

- Зайдите в меню **Math** (Математика).
- Возьмите блок **round** (округлить).
- Вставьте его в поле **number** блока **OLED**.
- Внутрь блока **round** поместите блок **get pressure** (получить давление) из меню **BMP280**.

- Цвет: **1**.

Результат: На первой строке дисплея будет отображаться температура в градусах Цельсия. На второй строке — атмосферное давление в Паскалях (например, 101325).

Файлы: Файл урока: BMP280.hex.



2.47 Урок: Эксперимент с датчиком освещенности BH1750

Цель: Создать устройство, которое измеряет уровень освещенности в люксах (I_x), выводит значение на OLED-дисплей и автоматически включает светодиодную матрицу micro:bit, если становится слишком темно (режим «умный ночник»).

Принцип работы: Датчик BH1750 — это цифровой сенсор, измеряющий интенсивность света. Он общается с процессором по протоколу I2C. В этом уроке мы используем общую шину I2C для подключения и дисплея, и датчика к одним и тем же пинам (P19 и P20).

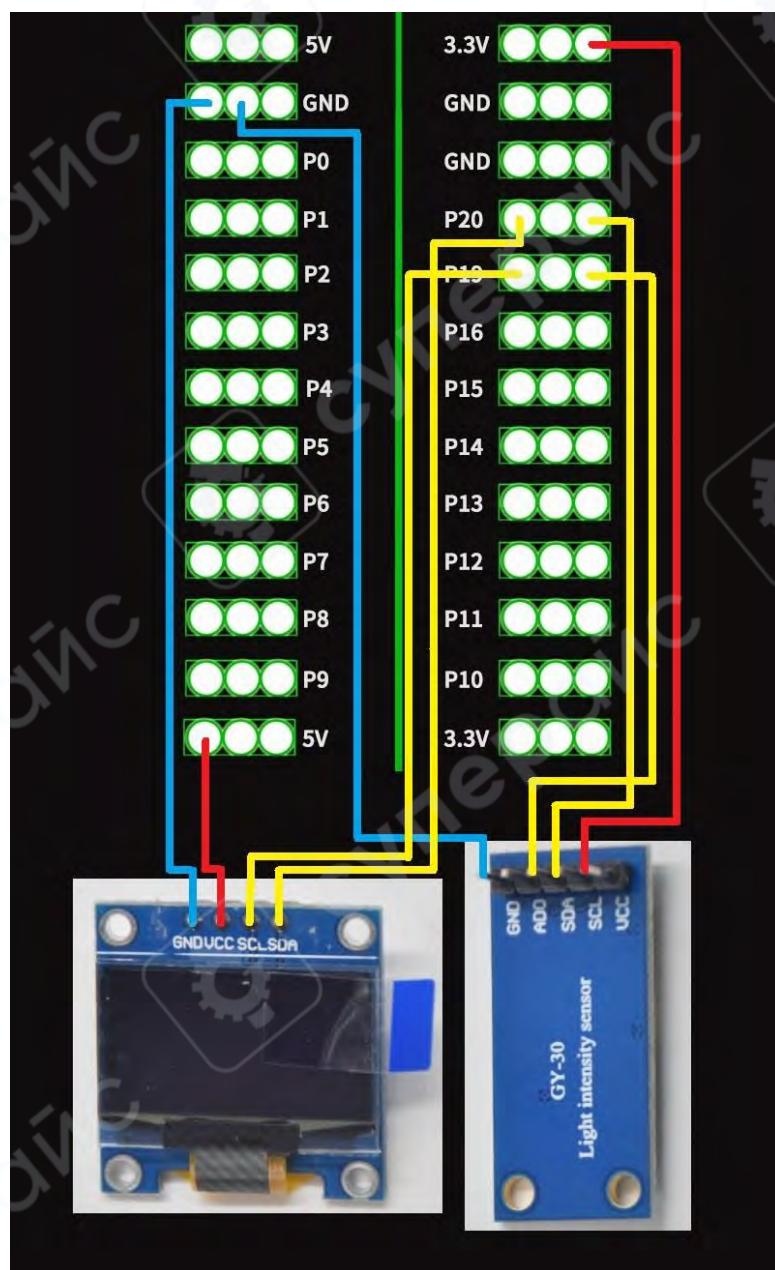
Схема подключения: Обратите внимание: И дисплей, и датчик подключаются к одним и тем же пинам (параллельно).

1. OLED дисплей:

VCC → 5V
GND → GND
SCL → P19
SDA → P20

2. Датчик BH1750:

VCC → 3,3V
GND → GND
SCL → P19
SDA → P20



Подключение датчика освещенности и дисплея

Программный код (сборка блоков):

Внимание: Для работы потребуется установить расширения) датчика BH1750

(<https://github.com/zhuning239/BH1750>).  BH1750

- **Блок «При начале» (On Start):**

- Инициализация датчика: Установить адрес (Set address) на **35**.

- Инициализация дисплея: init OLED with addr 60.

- **Блок «Постоянно» (Forever):**

1. **Вывод данных:**

- show a Number at x 0 y 0: Вставить блок считывания Light intensity (lx) с математическим округлением (если требуется).

- show string at x 5 y 0: Текст "lx" (единица измерения).

- Пауза: 500 мс.

- clear (очистка экрана OLED).

2. **Логика (If/Else):**

- **Если** Light Intensity (lx) < 50 (стало темно):

Show LED: нарисовать полностью закрашенный квадрат (свет включен).

- **Иначе (Else):**

Show LED: нарисовать пустой квадрат (свет выключен).

Результат: На OLED дисплее отображается текущая освещенность. Если закрыть датчик рукой (значение упадет ниже 50), на плате micro:bit загорятся светодиоды.

Файлы: Файл урока: BH1750 光照传感器.hex.



2.48 Урок: Модуль часов реального времени DS1302

Цель: Создать цифровые часы, которые показывают год, месяц, день, время и день недели на OLED-дисплее. Время будет сохраняться даже при отключении питания благодаря батарейке на модуле.

Принцип работы: Модуль DS1302 имеет собственную батарейку и кварцевый резонатор, что позволяет ему "тикать" автономно. Micro:bit считывает готовые данные о времени через цифровые пины.

Схема подключения:

1. OLED дисплей (I2C):

SCL → P19

SDA → P20

VCC → 5V / GND → GND

2. Модуль DS1302:

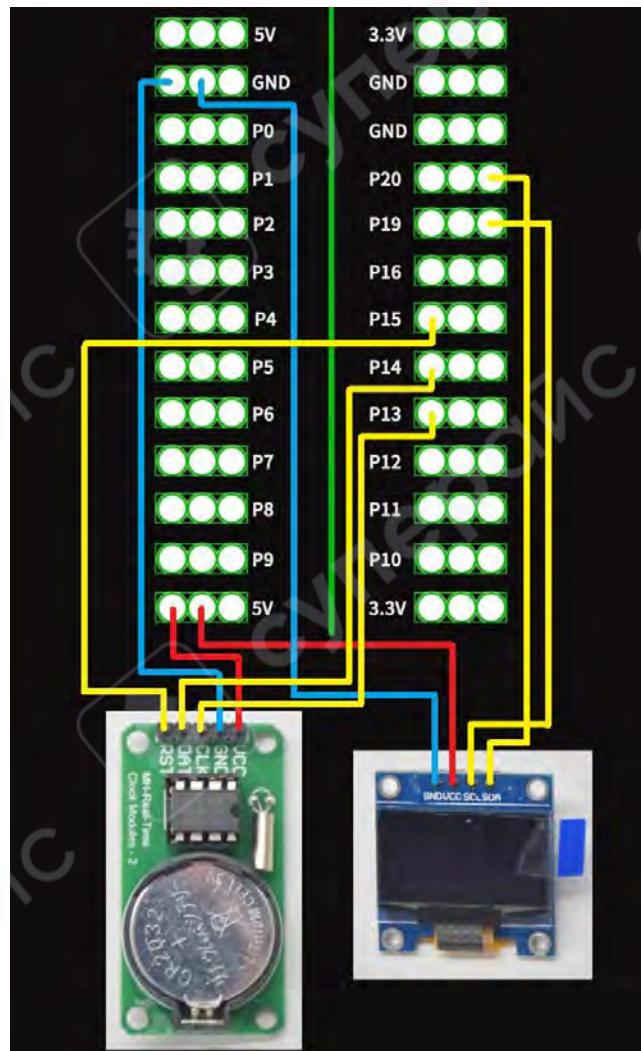
VCC → 5V

GND → GND

CLK (Clock) → P13

DAT (Data) → P14

RST (Reset/CS) → P15



Программный код (сборка блоков):

Требуется расширение для DS1302. Введите DS1302 в поиске, чтобы загрузить библиотеку часов DS1302

- **Блок «При начале» (On Start):**

- init OLED with addr 60.
- **Настройка модуля:** Блок Connect DS1302 (установить пины: CLK=P13, DIO=P14, CS=P15).
- **Установка времени:** Блок Set Time (Год, Месяц, День, День недели, Час, Минута, Секунда).

Примечание: Блок установки времени нужно использовать один раз для настройки, затем его можно убрать, иначе время будет сбрасываться при каждом включении.

- **Блок «Постоянно» (Forever):**

- Последовательный вывод значений на OLED (используйте координаты X/Y для расстановки):
 - Год: блок ds get Year
 - Месяц: блок ds get Month
 - День: блок ds get Day
 - Час: блок ds get Hour
 - Минута: блок ds get Minute
 - Секунда: блок ds get Second
 - День недели: блок ds get Weekday
- Pause (ms) 1000 (обновление раз в секунду).
- clear (очистка дисплея перед новым циклом).

Результат: На экране OLED бегут секунды, меняются минуты и часы в соответствии с установленным временем.

Файлы: Файл урока: 时钟显示频.hex.

```

при начале
    init OLED with addr 60
    установить ds ▾ значение CLK P13 ▾ DIO P14 ▾ CS P15 ▾
        ds ▾
        set Date and Time: Year 2025
        Month 5
        Day 26
        WeekDay 1
        Hour 23
        Minute 59
        Second 55
    постоянно
        show a Number at x 0
        y 0
        number ds ▾ get year
        color 1
        show a Number at x 5
        y 0
        number ds ▾ get month
        color 1
        show a Number at x 8
        y 0
        number ds ▾ get day
        color 1
        show a Number at x 0
        y 0
        number ds ▾ get hour
        color 1
        show a Number at x 5
        y 0
        number ds ▾ get minute
        color 1
        show a Number at x 8
        y 0
        number ds ▾ get second
        color 1
        show a Number at x 0
        y 0
        number ds ▾ get weekday
        color 1
        пауза 1000 ▾ (мсек)
        clear

```

2.49 Урок: Эксперимент с датчиком линии (препятствия)

Цель: Научиться считывать цифровой сигнал с инфракрасного (ИК) датчика и управлять внешним светодиодным модулем в зависимости от наличия препятствия или цвета поверхности.

Принцип работы: ИК-датчик излучает свет и ловит отражение. Если отражение есть (светлая поверхность или препятствие рядом) — выдается один сигнал. Если отражения нет (черная линия или пустота) — сигнал меняется. Мы считываем этот сигнал как 0 или 1.

Схема подключения:

1. ИК-датчик (датчик линии):

VCC → 5V

GND → GND

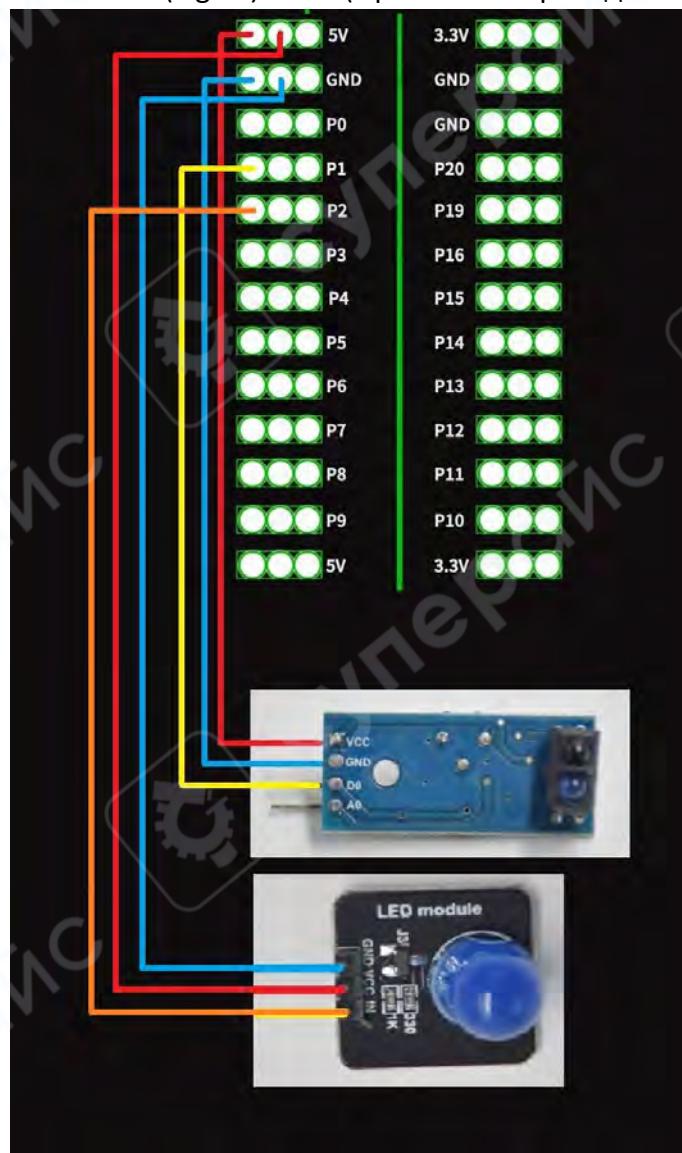
D0 (Digital Out) → P1 (Желтый провод на схеме)

2. Светодиодный модуль (LED):

VCC / IN+ → 5V (Красный)

GND / IN- → GND (Черный)

SIG (Signal) → P2 (Оранжевый провод на схеме)



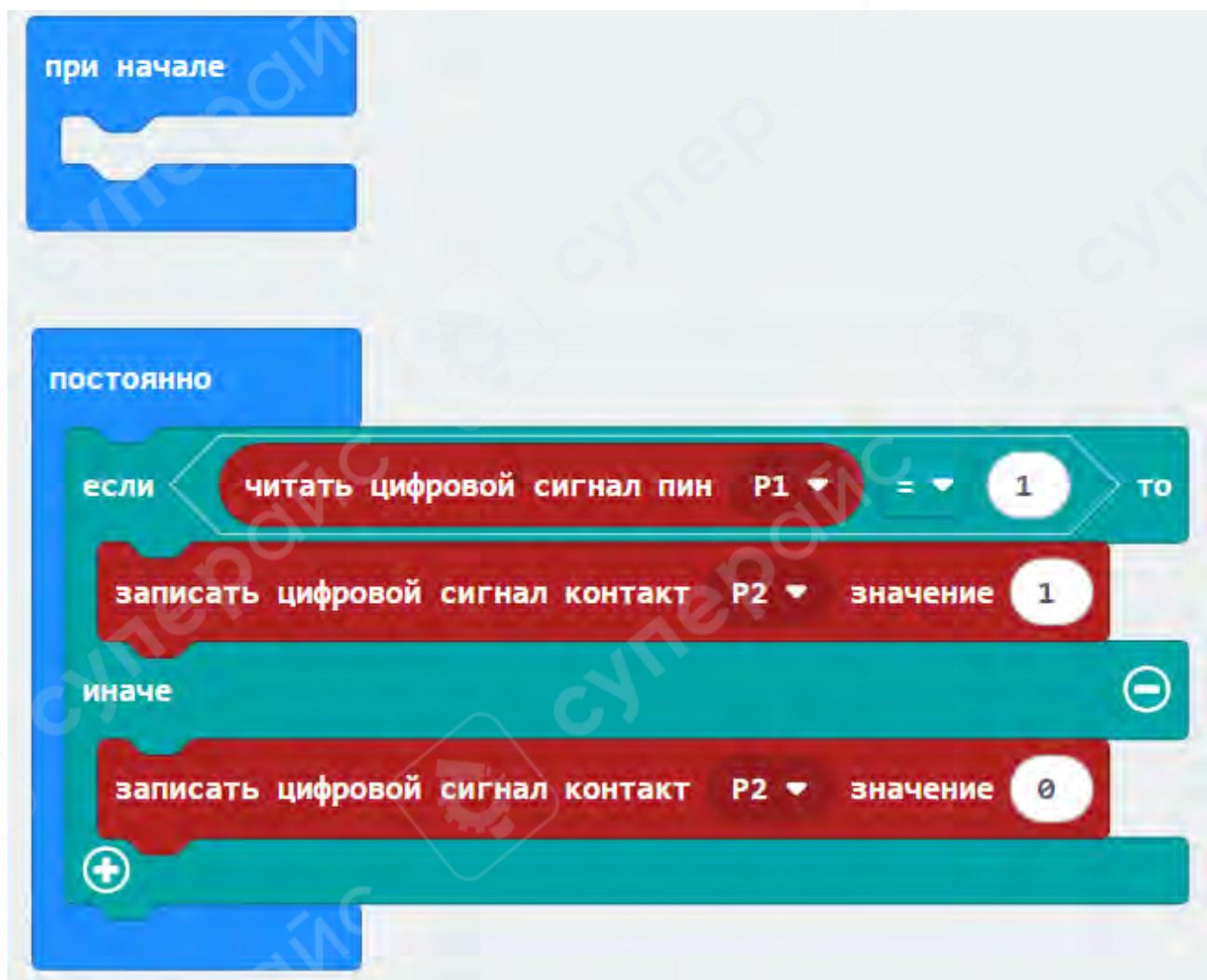
Программный код (сборка блоков):

- Блок «При начале» (On Start):
 - Пусто (или можно задать начальное состояние пинов).
- Блок «Постоянно» (Forever):
 - Логика (If / Else):
 - Условие: **Если** digital read pin P1 (цифровое чтение пина P1) = 1.
 - **То:** digital write pin P2 to 1 (включить внешний светодиод).
 - **Иначе:** digital write pin P2 to 0 (выключить внешний светодиод).

Примечание: В зависимости от типа датчика логика может быть инвертированной (1 — нет препятствия, 0 — есть препятствие). Если светодиод работает "наоборот", поменяйте 1 и 0 местами в блоках write pin.

Результат: Когда вы подносите руку к датчику (или помещаете его над светлой поверхностью), внешний светодиод загорается. Когда убираете руку (или датчик над черной линией) — светодиод гаснет.

Файлы: Файл урока: 循迹传感器.hex.



2.50 Урок: Кнопочное управление светодиодом

Цель: Создать устройство, в котором светодиод загорается при нажатии кнопки и гаснет, когда кнопку отпускают.

Принцип работы: Мы подключаем внешний модуль кнопки к pinу P0, а модуль светодиода — к pinу P1. Micro:bit будет считывать состояние кнопки. Когда кнопка нажата, она посылает цифровой сигнал «0». Программа проверяет это условие и подает напряжение на светодиод.

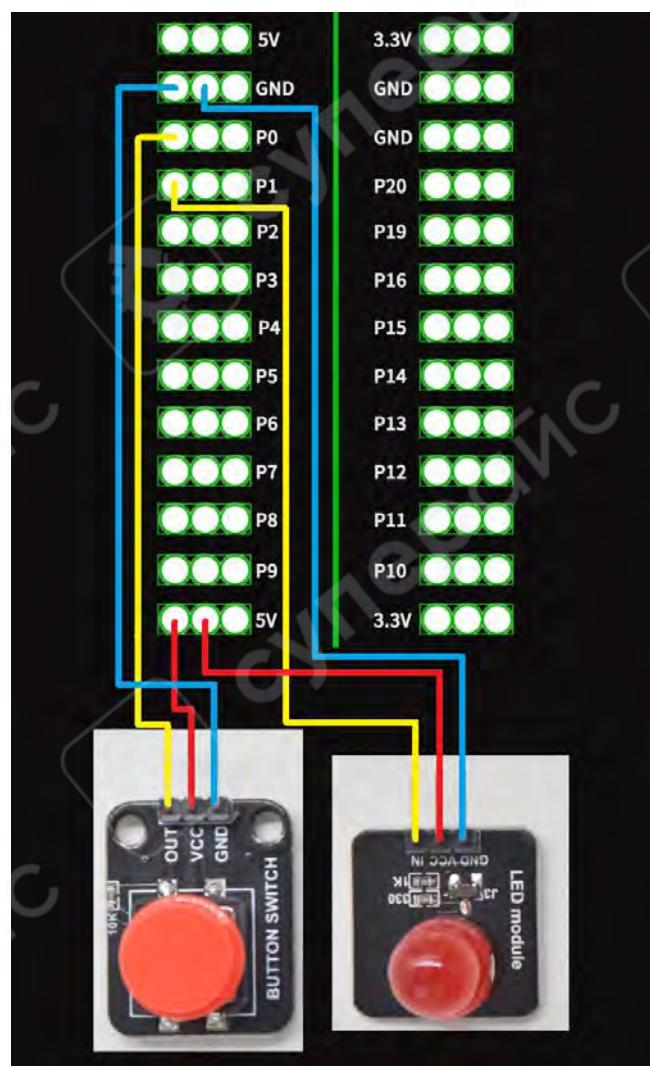
Подключение: Используйте провода (перемычки) для соединения модулей с платой расширения:

1. **Модуль кнопки (Button Switch):**

- Желтый провод (S) — к pinу P0.
- Красный провод (VCC) — к 5V.
- Черный провод (GND) — к GND.

2. **Модуль светодиода (LED Module):**

- Желтый провод (IN) — к pinу P1.
- Красный провод (VCC) — к 5V.
- Черный провод (GND) — к GND.



Программный код (сборка блоков):

Блок «При начале» (On Start): • Оставьте этот блок пустым.

Блок «Постоянно» (Forever): • Мы используем этот блок для непрерывной проверки состояния кнопки.

Шаг 1: Логическое условие

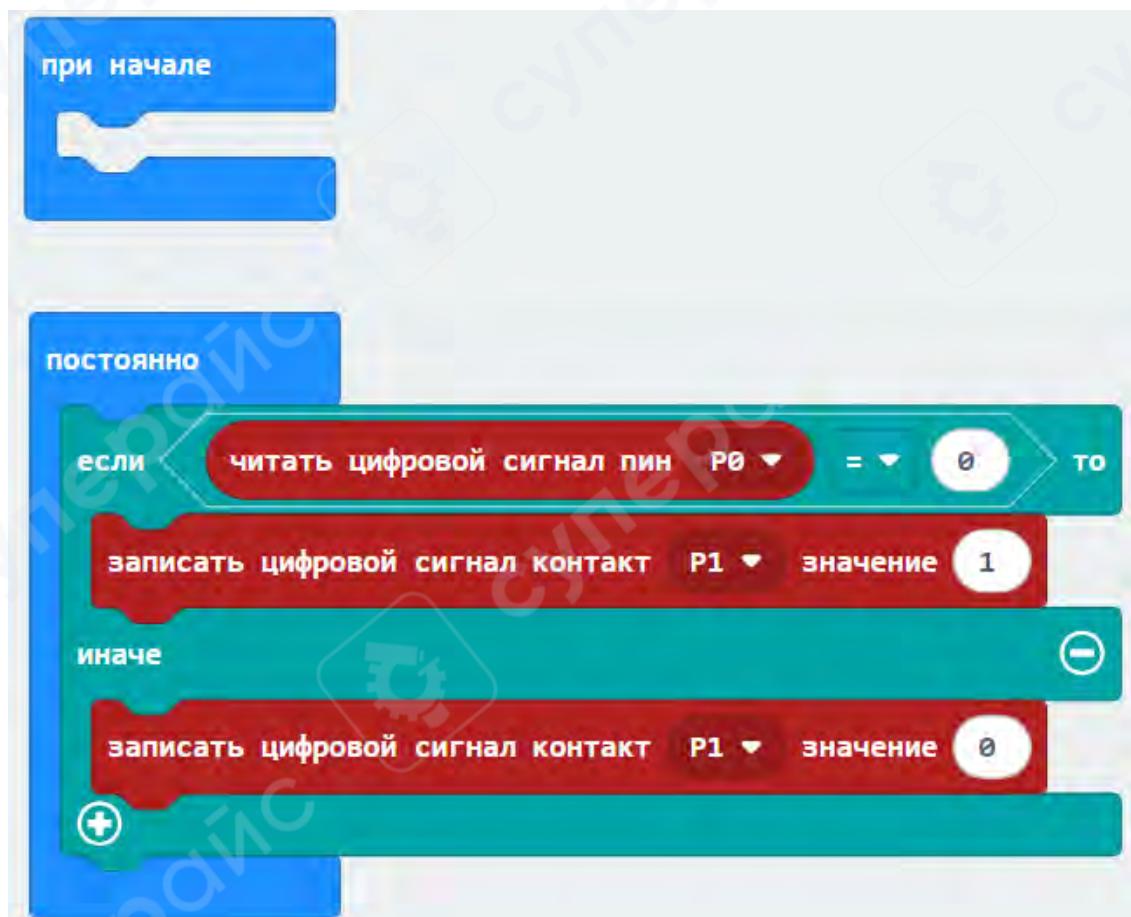
- Зайдите в меню **Logic (Логика)**.
- Выберите блок «Если... то... иначе» (If... then... else).
- В условие вставьте блок сравнения (=).
- Из меню **Pins (Пины)** выберите «цифровое считывание пин P0» (digital read pin P0) и вставьте в левую часть равенства.
- В правую часть напишите **0** (это означает нажатие).

Шаг 2: Управление светодиодом

- В секцию «То» (Then) поместите блок из меню **Pins**: «цифровая запись пин P1 значение 1» (digital write pin P1 to 1). Это включит свет.
- В секцию «Иначе» (Else) поместите блок «цифровая запись пин P1 значение 0» (digital write pin P1 to 0). Это выключит свет.

Результат: При нажатии на красную кнопку модуля загорается красный светодиод. При отпускании кнопки светодиод гаснет.

Файлы: Файл урока: **浇花代码.hex**.



2.51 Урок: Умный вентилятор (Эксперимент с тушением огня)

Цель: Создать автоматическую систему пожаротушения, которая включает вентилятор при обнаружении пламени.

Принцип работы: Используется датчик пламени (подключен к P0) и реле (подключено к P1), управляющее мотором вентилятора. Датчик пламени меняет свои показания в зависимости от интенсивности инфракрасного излучения. Когда датчик "видит" огонь, значение аналогового сигнала падает. Если значение опускается ниже 200, Micro:bit включает реле, которое запускает вентилятор.

Подключение:

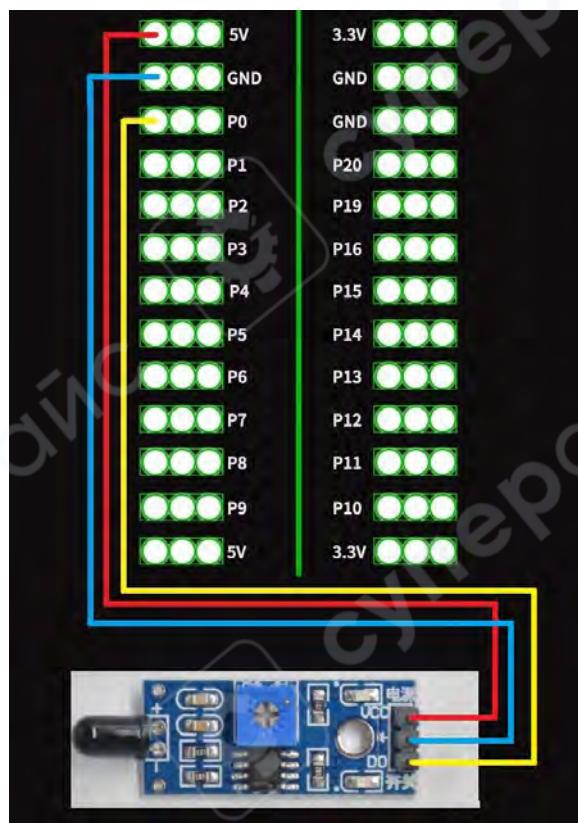
1. Логическая часть (к плате расширения):

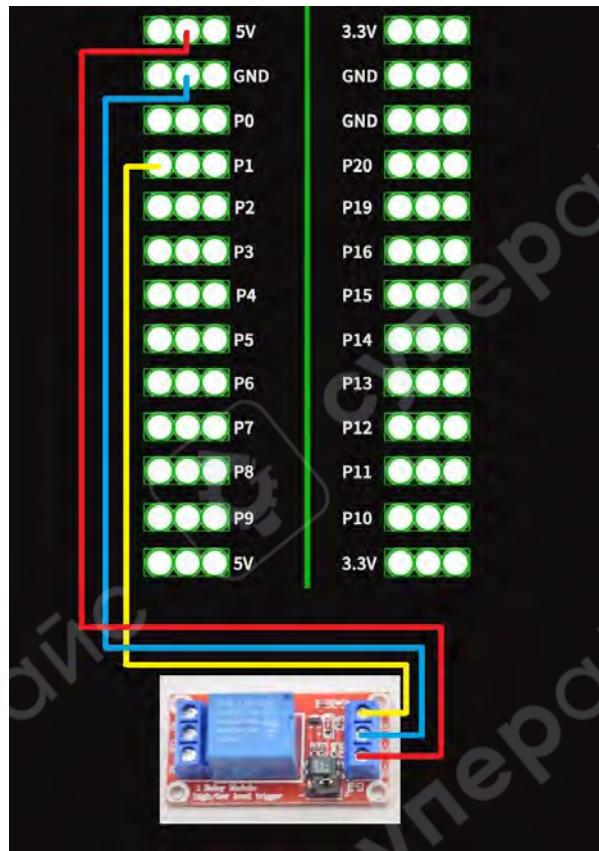
- Датчик пламени:

- Желтый провод (DO) — к pinu **P0**.
- Красный провод (VCC) — к **5V**.
- Черный провод (GND) — к **GND**.

- Модуль реле:

- Желтый провод (IN) — к pinu **P1**.
- Красный провод (VCC) — к **5V**.
- Черный провод (GND) — к **GND**.

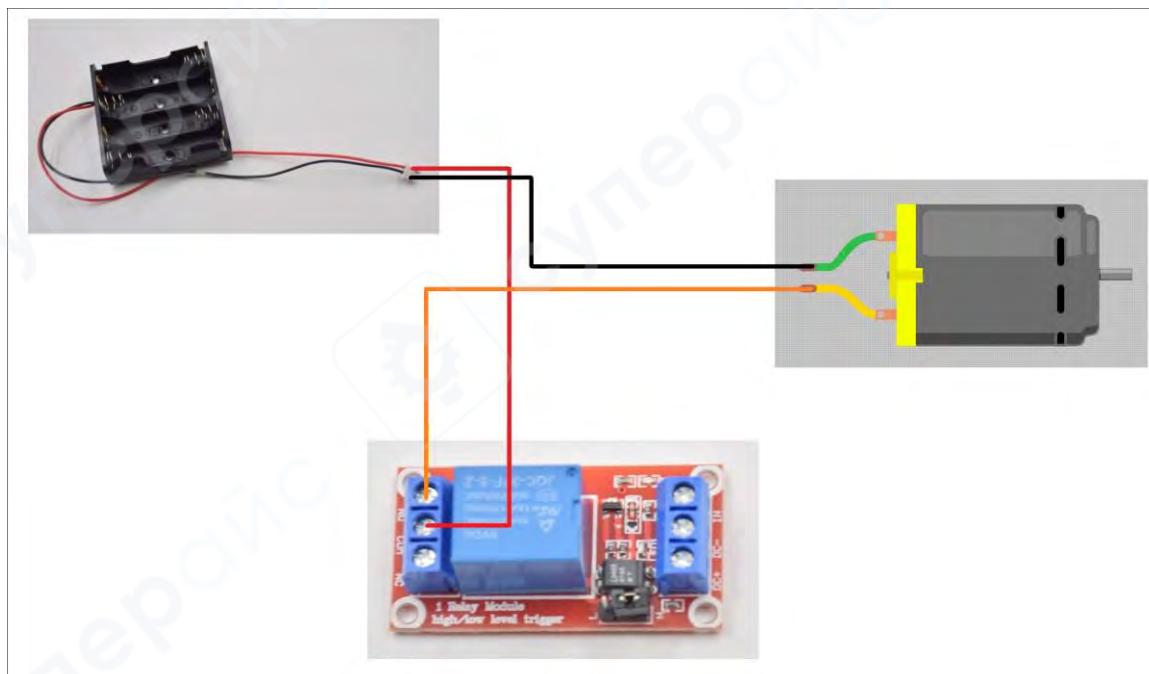




2. Силовая часть (Мотор и батарейный отсек):

Внимание: Мотор подключается не к плате, а через реле к батарейкам!

- Чёрный провод от батарейного отсека соедините с проводом мотора напрямую.
- Красный провод от батарейного отсека зажмите в **центральной клемме** реле (COM).
- Другой провод мотора зажмите в **верхней клемме** реле (NO - нормально разомкнутый).



Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

- Программа должна постоянно следить за показаниями датчика.

Шаг 1: Настройка датчика пламени

- Зайдите в меню **Logic (Логика)** и выберите блок «Если... то... иначе».

- Возьмите блок сравнения и измените знак на «<» (меньше).

• Из меню **Pins (Пины)** возьмите блок «аналоговое считывание пин P0» (analog read pin P0).

• Установите пороговое значение **200**. Логика: если показания < 200, значит, огонь близко.

Шаг 2: Запуск вентилятора

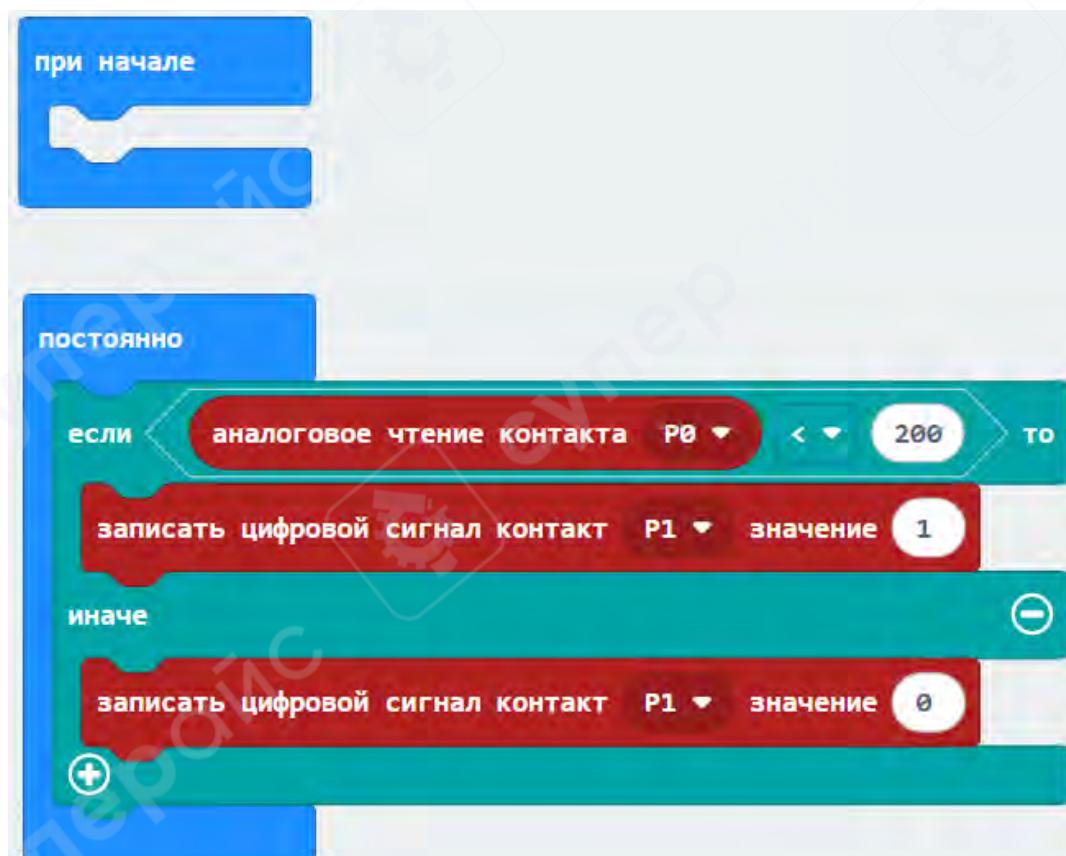
• В секцию «To» (Then) добавьте блок «цифровая запись пин P1 значение 1» (digital write pin P1 to 1). Это включит реле.

• В секцию «Иначе» (Else) добавьте блок «цифровая запись пин P1 значение 0» (digital write pin P1 to 0). Это выключит реле.

Результат: Если поднести зажженную зажигалку к датчику пламени, вентилятор автоматически включится. Когда вы уберете огонь, вентилятор остановится.

Примечание: Вентилятор питается от отдельного батарейного блока через реле, так как требует больше энергии, чем может дать micro:bit. Если датчик не реагирует, отрегулируйте чувствительность синим потенциометром на плате датчика.

Файлы: Файл урока: 灭火风扇.hex



2.52 Урок: Датчик движения (PIR)

Цель:

Создать охранную систему, которая реагирует на присутствие человека. При обнаружении движения на дисплее micro:bit должен загореться значок «Сердце», а в состоянии покоя — значок «Крестик» (или другой индикатор отсутствия движения).

Принцип работы:

В этом уроке используется датчик движения HC-SR501 (PIR-сенсор). Он реагирует на изменение инфракрасного излучения (тепла), исходящего от тела человека.

- Датчик имеет три контакта: Питание (VCC), Земля (GND) и Сигнал (OUT).
- Когда датчик замечает движение, он посылает на сигнальный пин высокий уровень напряжения (цифровая «1»).
- Когда движения нет, он посылает низкий уровень (цифровой «0»).

Программа будет непрерывно считывать состояние пина P0 и менять картинку на экране.

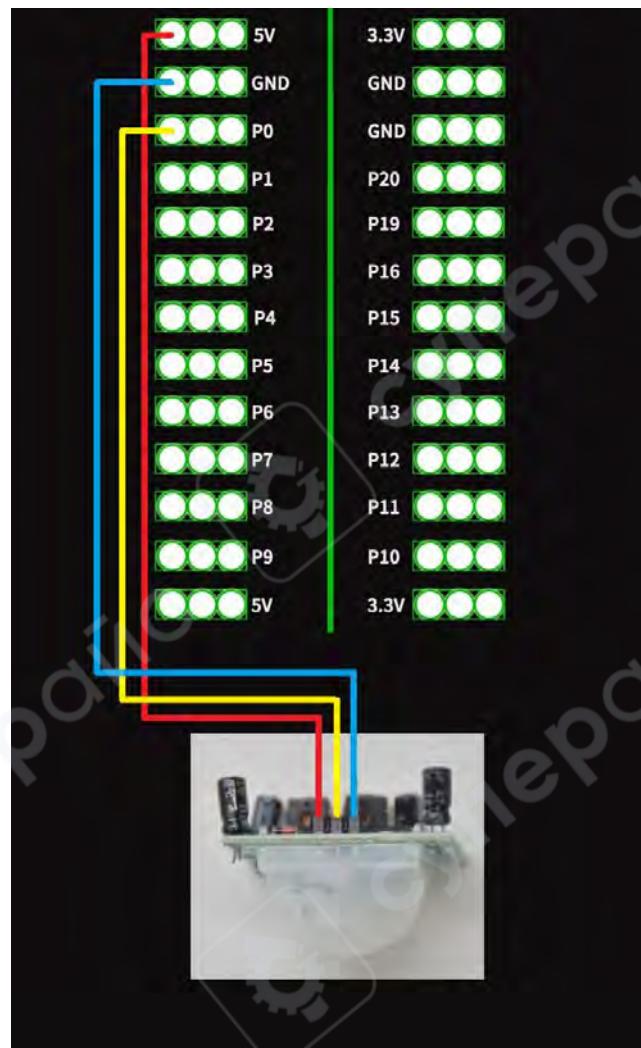
Схема подключения:

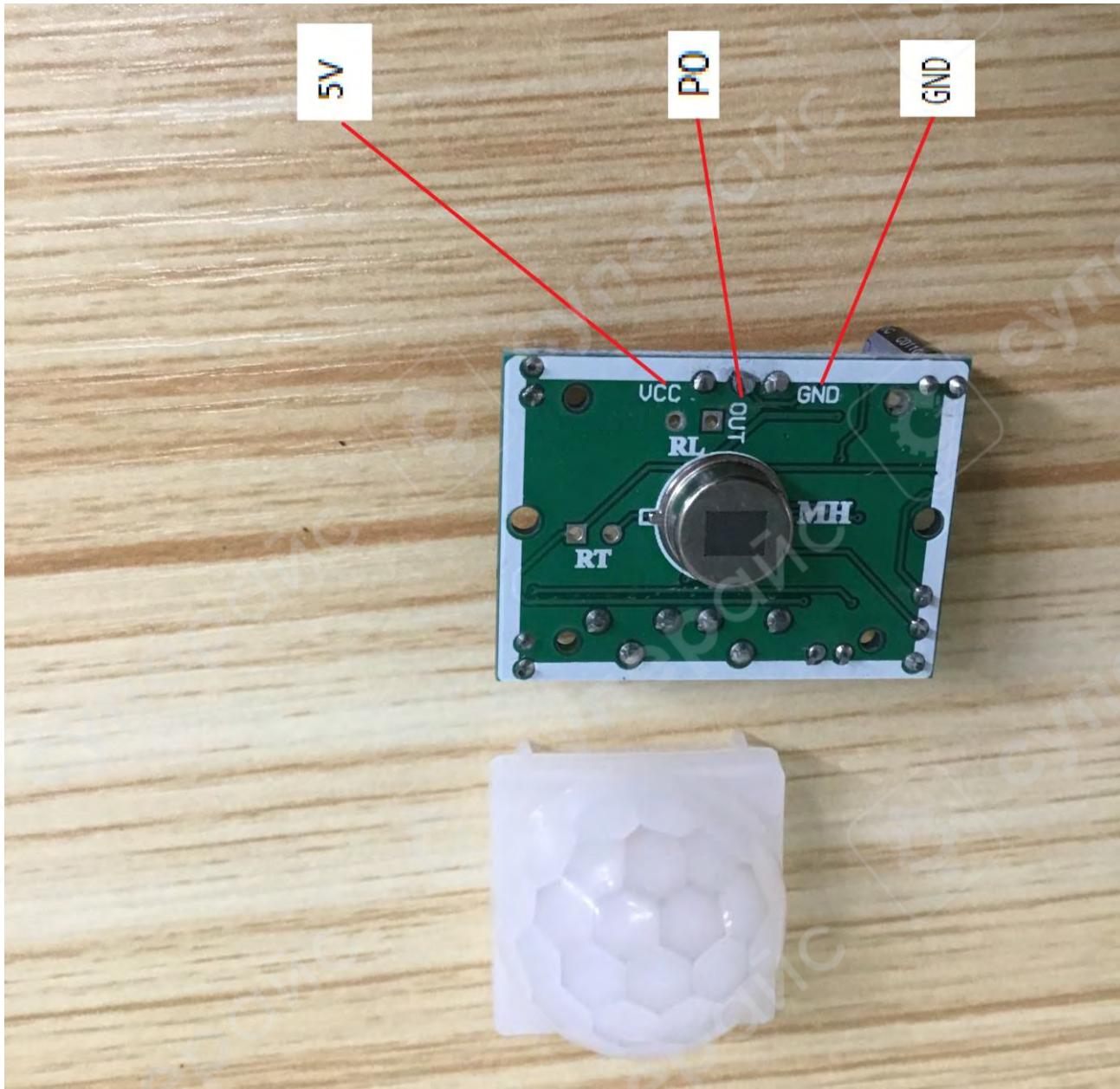
Для этого урока понадобится плата расширения GPIO и соединительные провода (типа «мама-мама»). Ориентируйтесь по надписям на датчике (снимите белую линзу, если надписи не видны, или посмотрите на обратную сторону платы датчика).

Таблица соединений:

Контакт датчика (PIR)	Провод на схеме	Контакт на плате расширения
VCC (или 5V)	Красный	5V (группа пинов питания)
GND	Синий	GND (Минус/Земля)
OUT (Центральный)	Желтый	P0 (Сигнальный пин)

Примечание: На датчике есть два оранжевых регулятора. Они отвечают за чувствительность и время задержки. Если датчик работает некорректно, попробуйте аккуратно подкрутить их отверткой.





Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте этот блок пустым.

Блок «Постоянно» (Forever):

- Мы используем этот блок для непрерывного мониторинга датчика.

Шаг 1: Логическое условие

- Зайдите в меню **Logic** (Логика).
- Выберите блок **if <true> then... else** (если <истина> то... иначе).
- Перетащите его внутрь блока **Forever**.

Шаг 2: Считывание сигнала

- Зайдите в меню **Logic** (Логика) и возьмите блок сравнения $0 = 0$. Вставьте его вместо `<true>`.

- Зайдите в меню Pins (Пины/Выводы).
- Выберите блок digital read pin P0 (цифровое чтение пина P0).
- Вставьте этот блок в левую часть равенства.
- В правой части равенства установите цифру 1.
- Конструкция: Если (digital read pin P0 = 1) то...

Шаг 3: Настройка индикации

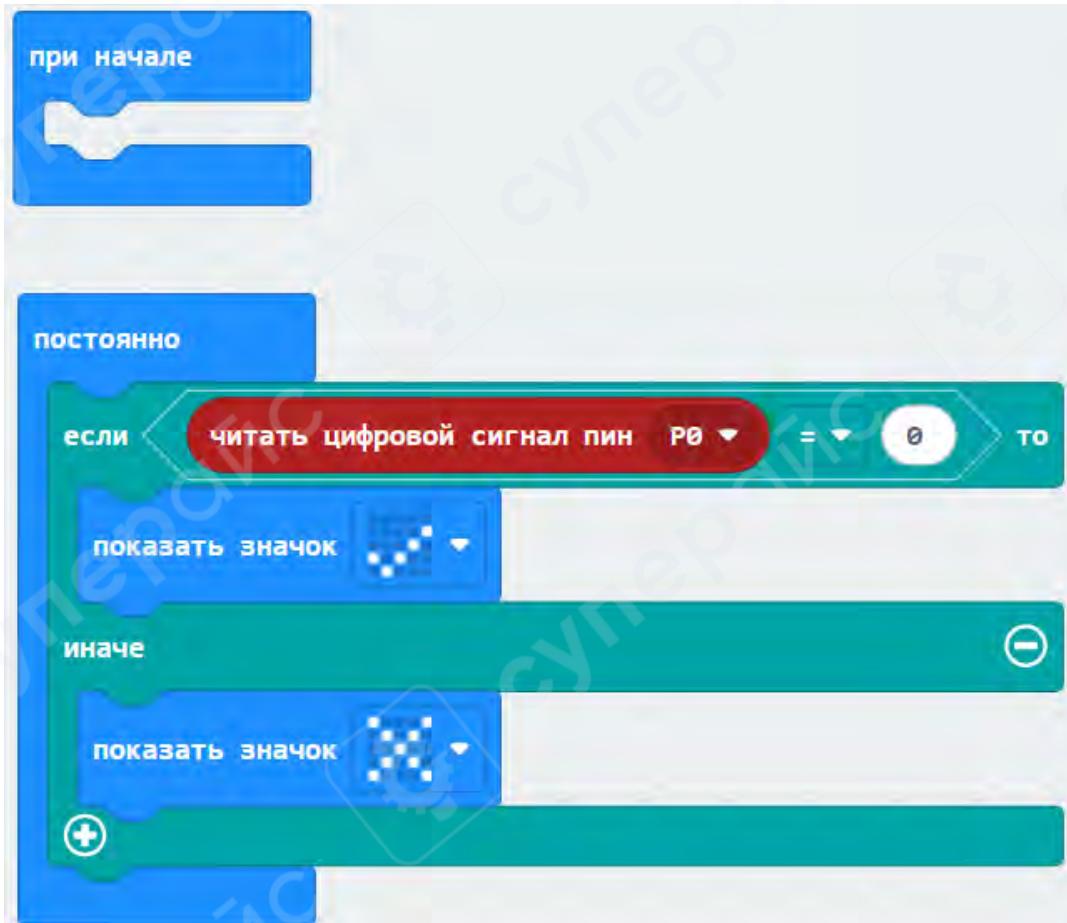
- В раздел «то» (then) (Движение есть):
- Зайдите в меню Basic (Основное).
- Выберите блок show icon (показать значок) -> Галочка.
- В раздел «иначе» (else) (Движения нет):
- Добавьте блок show icon (показать значок) -> Крестик (или маленький квадрат).

Результат:

После загрузки программы:

1. Если перед датчиком нет движения, на матрице micro:bit горит Крестик.
2. Стоит провести рукой перед белым куполом датчика, как изображение сменится на галочку.

Файлы: Файл урока: microbit-人体感应.hex.



2.53 Урок: Автоматическая система полива

Цель: Создать устройство, которое отслеживает влажность почвы и автоматически включает водяной насос, если земля становится слишком сухой.

Принцип работы: Система состоит из трех основных частей:

1. **Датчик влажности почвы:** Измеряет электропроводность земли. Влажная земля проводит ток хорошо, сухая — плохо. Датчик выдает как аналоговый сигнал (точное значение влажности), так и цифровой (сигнал «Сухо/Влажно»), настроенный с помощью порогового регулятора.

2. **Релейный модуль:** Работает как электронный выключатель. Micro:bit не может напрямую питать мощный мотор насоса, поэтому он посылает слабый сигнал на реле, которое замыкает цепь внешней батареи для насоса.

3. **Водяной насос:** Качает воду, когда реле замкнуто.

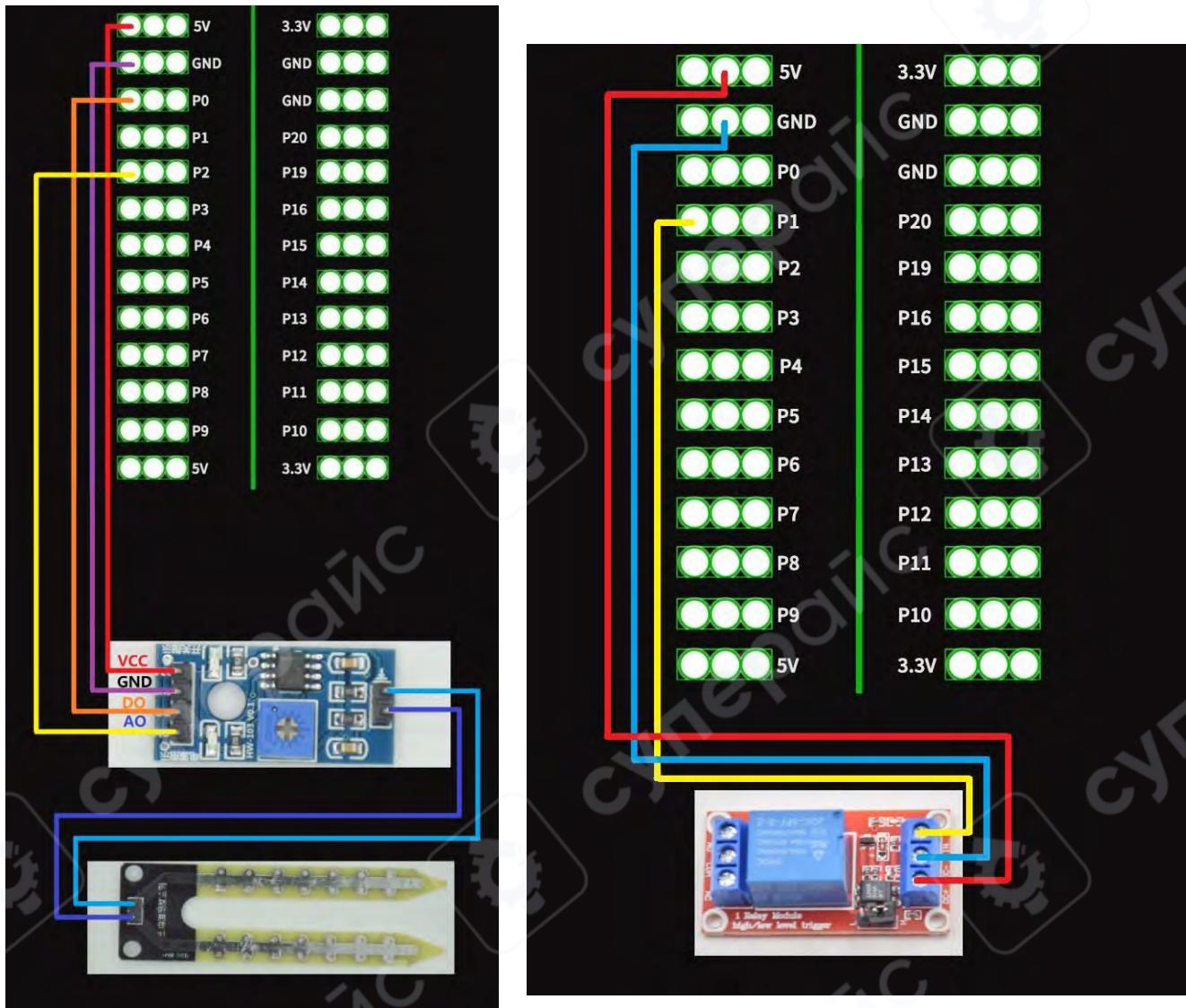
Схема подключения:

В этом проекте много соединений, поэтому будьте внимательны. Схема делится на две части: слаботочную (управление) и силовую (питание насоса).

Часть 1: Подключение датчиков к плате расширения micro:bit

Устройство	Контакт устройства	Цвет провода (на схеме)	Контакт на плате расширения
Датчик влажности (Плата управления)	VCC	Красный	5V
	GND	Фиолетовый	GND
	DO (Цифровой выход)	Оранжевый	P0
	AO (Аналоговый выход)	Желтый	P2
Релейный модуль (Сторона управления)	VCC (DC+)	Красный	5V
	GND (DC-)	Синий	GND
	IN (Сигнал)	Желтый	P1

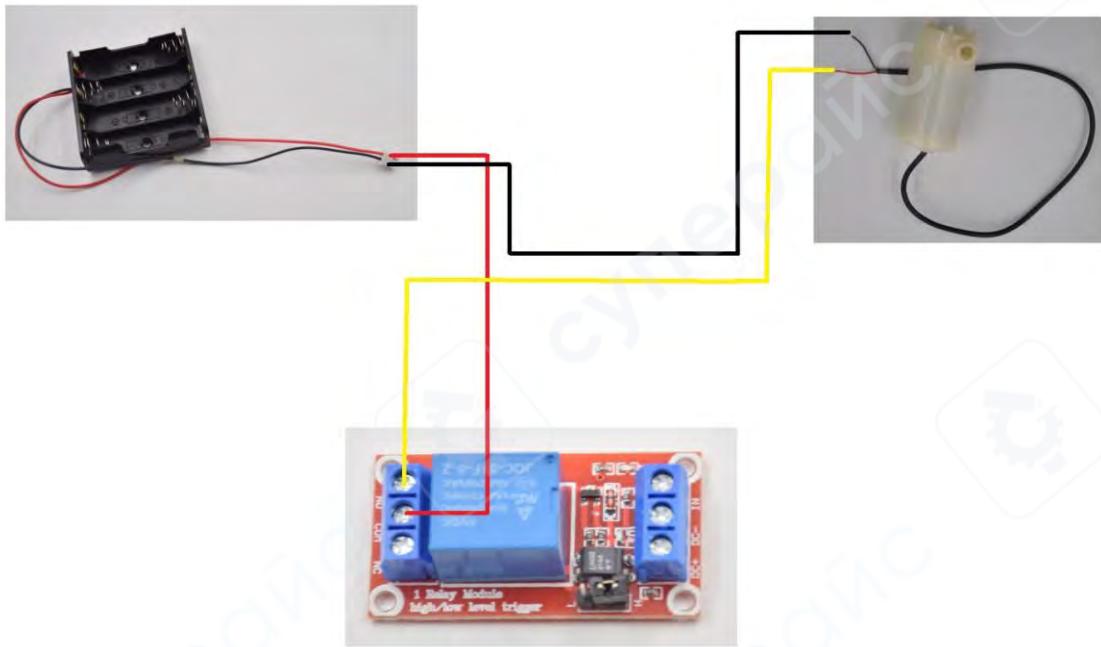
Примечание: Щупы датчика влажности (вилка) подключаются к плате датчика двумя любыми проводами (на схеме синий и фиолетовый).



Часть 2: Силовая цепь (Насос и Батарейный отсек)

Внимание: Эта цепь собирается на винтовых клеммах реле (синий кубик).

1. Красный провод от батарейного отсека (+) подключите к центральному контакту реле (**COM**).
2. Второй контакт реле (**NO** — нормально разомкнутый) соедините с **красным проводом** насоса.
3. Черный провод от батарейного отсека (-) соедините напрямую с **черным проводом** насоса (скрутите их или используйте соединитель).



Программный код (сборка блоков):

Блок «При начале» (On Start):

- Оставьте пустым.

Блок «Постоянно» (Forever):

- Этот цикл будет постоянно проверять состояние почвы.

Шаг 1: Вывод данных на экран

- Зайдите в меню **Basic** (Основное).
- Выберите блок **show number** (показать число).
- Зайдите в меню **Pins** (Пины).
- Выберите **analog read pin P2** (аналоговое чтение пина P2).
- Вставьте его внутрь блока **show number**.
- Теперь на экране будет отображаться уровень влажности (от 0 до 1023).

Шаг 2: Логика управления насосом

- Зайдите в меню **Logic** (Логика).
- Возьмите блок **if <true> then... else** (если <истина> то... иначе).
- Для условия <true> возьмите блок сравнения = из меню Логика.
- В левую часть сравнения вставьте **digital read pin P0** (цифровое чтение пина P0).
- В правую часть напишите **1**.
- Пояснение: Датчик настроен так, что выдает «1», когда почва сухая.

Шаг 3: Включение и выключение

- В раздел «**то**» (**then**) (Почва сухая):
- Зайдите в **Pins** (Пины).
- Выберите **digital write pin P1 to 1** (цифровая запись на пин P1 значение 1).
- Это включит реле и насос.

- В раздел «иначе» (else) (Почва влажная):
- Выберите **digital write pin P1 to 0** (цифровая запись на пин P1 значение 0).
- Это выключит насос.

Результат:

1. Опустите датчик в сухую почву (или просто держите на воздухе). На экране появятся числа, реле щелкнет, и насос начнет качать воду.

2. Опустите датчик в стакан с водой. Насос должен остановиться.

Важное примечание:

На синей плате датчика влажности есть **синий потенциометр** (под крестовую отвертку). Он регулирует чувствительность.

- Если насос работает постоянно — покрутите винт, пока не погаснет один из светодиодов на датчике.
- Если насос не включается даже в сухом состоянии — крутите винт в обратную сторону.

Файлы: Файл урока: **浇花代码.hex**

