

Рис. 4

тере. Игроки поочередно делают ходы, выставляя свою фишку (**рис. 5**) перед роботом, когда тот неподвижен и подсказывает глазами свои очередные действия. Робот делает движение и при удачном ходе сбивает её. Игроку набавляется число баллов на выпавшей грани (выпавшая грань — та, на которой лежит фишка). После хода он быстро убирает её с игрового поля и вслух произносит свой текущий счёт. Фишку ставит следующий игрок, и так по кругу. Если игрок своими действиями мешает сопернику делать свой ход, с него снимают текущий результат хода.

ЛИТЕРАТУРА

1. Мамичев Д. Тележка с дистанционным управлением на одном ведущем колесе. — Радио, 2019, № 7, с. 62—64.
2. Arduino библиотека Wire. — URL: <https://all-arduino.ru/biblioteki-arduino/arduino-biblioteka-wire/> (08.05.21).
3. OLED I²C дисплей 128x64 пикселя. — URL: <https://cxem.net/arduino/arduino227.php> (08.05.21).
4. SSD1306: контроллер/драйвер для дисплеев OLED/PLED. — URL: <http://microsin.net/adminstuff/hardware/ssd1-306-oled-controller.html> (8.05.21).
5. Визуализация для микроконтроллера.



Рис. 5

Часть 1. OLED дисплей 0.96" (128x64) на SSD1306. — URL: <https://datagor.ru/microcontrollers/microcontrollers-code-library/2912-vizualizaciya-mcu-part-1-oled-128x64-ssd1306.html> (08.05.21).

От редакции. На нашем FTP-сервере по адресу <http://ftp.radio.ru/pub/2022/02/glazki.zip> находятся материалы проекта, а также видеозаписи работы устройства.

Ответы на викторину "Arduino: программная часть-8"

("Радио", 2022, № 1, с. 63, 64)

С. РЮМИК, г. Чернигов, Украина

1 Ответ — 1. К плате Arduino подключается клавиатура 4x3. Клавиатура (нем. Tastatur) — это цифровое наборное поле, по-другому — кнопочный номеронабиратель. Выполняется он в виде сенсорных или механических кнопок, контакты которых "квадратно-гнездовым" способом электрически соединяют строки со столбцами (**рис. 1**).

Для опроса клавиатуры на её столбцы последовательно во времени подаются сигнал "бегущий ноль" с выходов D2—D4 Arduino. Это предполагает в каждый момент времени установку низкого уровня на одном и высокого уровня на других выходах. Высокий

уровень можно задать формированием лог. 1 или переводом линии порта в режим входа с внутренним "pull-up" резистором.

Если выставить на линии D2 лог. 0, а на линиях D3, D4 лог. 1, то при нажатии сразу на две кнопки вместе, например "1" и "3", линии D2 и D4 будут замыкаться накоротко, что недопустимо, ведь внутри клавиатуры ограничительных резисторов нет. Следовательно, порты Arduino, в целях безопасности, лучше настраивать как входы "pull-up".

2 Ответ — 0. Назначение скетча — опрос клавиатуры A1 и вывод на экран компьютера названия нажатой

кнопки. Для работы используется внешняя библиотека Keypad (строка 2), в которой задаются кнопочные массивы (строки 3—5), а также порты Arduino, подключённые к горизонталям (D5—D8, строка 6) и вертикалям (D2—D4, строка 7).

Информацию о нажатой кнопке выдаёт функция **getKey** в строке 12. Поскольку слово Key (англ. кнопка) указано в единственном числе, то в мониторе среды Arduino IDE будет печататься "5" или "7", в зависимости от того, какая из кнопок была нажата раньше по времени. Надпись "57" не может появиться в принципе, поскольку переменная **tast** в строке 12 рассчитана на приём только одного символа.

3 Ответ — 1. Назначение скетча — ожидание нажатия двух (любых) кнопок клавиатуры A1 в выбранной горизонтали, после чего на экран компьютера выводится слово ОК (строка 13). Нужная горизонталь задаётся в строках 6, 7, где выставляется лог. 0 на линии D6 Arduino. Согласно электрической схеме линия D6 соединяется с третьей сверху горизонталью (кнопки "7", "8", "9"), что и является ответом на вопрос.

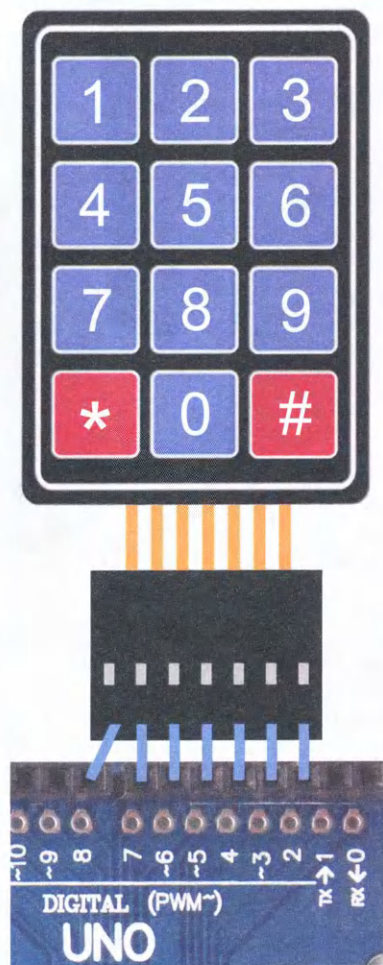


Рис. 1

Интересный нюанс. В скетче входы и выходы Arduino поменялись местами, т. е. выходные сигналы формируются на линиях D5—D8, а входные — на линиях D2—D4. Это допускается, поскольку клавиатура не имеет входов и выходов, внутри неё находятся лишь кнопки.

4. Ответ — 1. Операционный усилитель DA1 работает в режиме компаратора. Порог его срабатывания определяется напряжением на инвертирующем входе (вывод 2), которое, в свою очередь, зависит от скважности сигнала ШИМ на выходе D3 Arduino.

Конденсатор C1 служит для сглаживания пульсаций. Если подаются импульсы меандра частотой несколько килогерц, форма напряжения на конденсаторе C1, а значит, и на резисторе R2 будет близкой к треугольной.

Однако в вопросе викторины не указаны ни частота, ни скважность импульсов ШИМ. Следовательно, форма сигнала на резисторе R2 может быть как треугольной, так и пилообразной

или близкой к прямоугольной, например, при низкой частоте ШИМ.

5. Ответ — 1. Назначение скетча — вывод на экран компьютера длительности сигнала высокого уровня, приходящего на вход D2 Arduino. Порог срабатывания компаратора DA1 устанавливается в строке 9. Для его формирования применяется метод ШИМ с фильтрацией нижних частот конденсатором C1.

Частота сигнала ШИМ задаётся в строках 7, 8 непосредственно через регистры таймеров микроконтроллера [1]. Частота выбрана высокой — 62,5 кГц, чтобы уменьшить пульсации напряжения на конденсаторе C1 до 10 мВ. Если операторы в строках 7, 8 закомментировать, то частота сигнала ШИМ вернётся по умолчанию к 488 Гц, при этом пульсации на конденсаторе C1 возрастут по амплитуде до 2,3 В. Для точного измерения длительностей это не допустимо.

6. Ответ — 1. Назначение скетча — измерение длительностей положительной и отрицательной полуволн сигнала $U_{\text{вх}}$ и занесение данных в массив `tab[]`. Окончание прохода очередного цикла индицируется внутренним светодиодом Arduino.

Время измеряется в микросекундах функцией `pulseIn`. В ней запускается счётчик, который начинает работать в момент перехода сигнала из лог. 0 в лог. 1 для **HIGH** или из лог. 1 в лог. 0 для **LOW**. Заканчивается счёт в момент перехода лог. 1 в лог. 0 для **HIGH** или лог. 0 в лог. 1 для **LOW**.

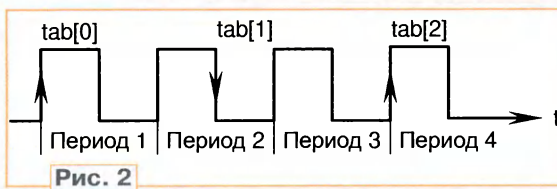


Рис. 2

На рис. 2 показана диаграмма заполнения первых ячеек массива `tab[]`. Как видно, измерить длительности лог. 1 и лог. 0 в одном и том же периоде не получится, поскольку функции `pulseIn(pin, HIGH)` и `pulseIn(pin, LOW)` следуют одна за другой в строках 10, 11. Более того, из измерений выпадает каждый третий период, поэтому заполнение массива будет происходить медленнее, чем может показаться на первый взгляд.

Расчёт. При входном сигнале с частотой 100 Гц на прохождение одного цикла оператора `for` в строке 9 требуются три периода по 10 мс, в сумме 30 мс. Для заполнения всего массива `tab[]` нужно 50 циклов, итого $30 \cdot 50 = 1500$ мс. Через это время светодиод Arduino изменит своё состояние на противоположное в строке 13. Период мигания светодиода в два

раза больше, т. е. 3000 мс, или 3 с.

7. Ответ — 0. На вход D3 Arduino поступает цифровой сигнал от логического элемента микросхемы DD1. Двухцветный светодиод HL1 индицирует его логическое состояние, для чего на выходе D9 Arduino выставляется лог. 0 при активации зелёного или лог. 1 при активации красного излучателей.

Для нормальной работы светодиода HL1 требуется знакопеременное напряжение, т. е. нужны два источника сигнала или два выхода. В рассматриваемой схеме один выход — это линия D9 Arduino, а другой — вывод 3 элемента 2И-НЕ микросхемы DD1. Это помогает сократить число требуемых выводов Arduino, иначе пришлось бы

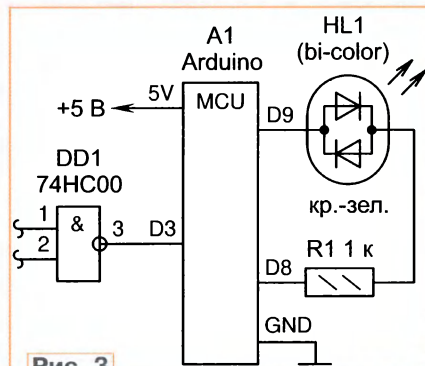


Рис. 3

воспользоваться схемой, как показано на рис. 3.

8. Ответ — 1. Назначение скетча — при низком уровне на входе D3 Arduino на светодиод HL1 будет подаваться уровень лог. 1 с выхода D9, и, наоборот, при этом светодиод будет красным или зелёным. При входном сигнале с формой меандра и частотой 100 Гц свечение будет близким к жёлтому. Это происходит вследствие инерционности зрительного аппарата человека. На аналогичном эффекте основан кинематограф, когда быстро чередующиеся картинки сливаются в одно плавное движение.

При входном сигнале частотой 1 Гц смешивания цветовых оттенков не получится, индикатор HL1 будет мигать хорошо различимыми цветами по 0,5 с каждого из них.

9. Ответ — 0. Назначение скетча — тестовое мигание светодиода HL1 перед началом тела основной программы `loop`. Так часто делают, когда хотят визуально убедиться, что скетч начал работу, а микроконтроллер и светодиод исправны. Дополнительно можно по цвету мигания светодиода определить, какой начальный уровень был установлен на выходе логического элемента микросхемы DD1.

Число миганий определяется циклом `for` в строке 6. При каждом его про-

ходе светодиод HL1 в течение 200 мс светит и на 200 мс гаснет. Однако при лог. 0 на входе D3 Arduino светодиод будет мигать красным цветом 10 раз, а при лог. 1 — зелёным цветом, но только 9 раз. Дело в том, что при последнем проходе цикла **for** на выходе D9 Arduino устанавливается лог. 0 (строка 9), из-за чего зелёный светодиод не может завершить последнее десятое мигание. Он так и останется во включённом состоянии при переходе на функцию **loop** в строке 13.

10 ● **Ответ — 1.** Имитатор пульта ДУ содержит кнопку SB1, при нажатии на которую излучающий ИК-диод VD1 по программе Arduino посылает к ИК-приёмнику телевизора пачки импульсов в выбранном стандарте.

Казалось бы, в данном устройстве реализуется принцип прямой, а не обратной связи. Однако надо смотреть шире, ведь в данном случае в систему управления добавляется... человек. Действительно, человек нажимает на кнопку SB1 и наблюдает, включился телевизор или нет. Если не включился, нажимает на кнопку повторно, поворачивает ИК-диод в нужном направлении, проверяет исправность телевизора и т. д. Человек выполняет эти действия до тех пор, пока не получит результат, тождественно совпадающий с образом, который был заранее сформирован в его сознании, — образом включённого телевизора.

Обратная связь будет не положительной, а отрицательной, поскольку человек тем или иным способом старается добиться включения телевизора, т. е. он стабилизирует ситуацию, используя в качестве исполнительного механизма свои рабочие органы [2].

11 ● **Ответ — 1.** Назначение скетча — формирование кодовой посылки в формате Sony SIRC-12 на

включение телевизора (строка 10) при нажатии на кнопку SB1 (строки 7, 8).

Важный момент. Импульсная последовательность посылается не один, а три раза подряд с паузами 100 мс (строки 9—12). Так заложено в протоколе управления микроконтроллера, находящегося внутри телевизора. Более того, паузы между посылками должны быть не слишком большими, иначе команда будет проигнорирована. Например, если установить в строке 11 паузу не 100 мс, а 1 с, то телевизор включаться не будет, чему есть подтверждение в Интернете [3].

12 ● **Ответ — 0.** Назначение скетча — тестовая проверка исправности излучающего ИК-диода

```
int main(void)
{
  init();
  initVariant();
  #if defined(USBCON)
    USBDevice.attach();
  #endif
  setup();
  for (;;) {
    loop();
    if (serialEventRun)
      serialEventRun();
  }
  return 0;
}
```

Рис. 4

VD1. Для этого в строках 8—13 в бесконечном цикле формируются посылки на 0,2 с включения и на 1 с выключения диода. Визуально диод светится не будет, но его излучение легко об-

наружить фотокамерой смартфона, которая, как правило, имеет достаточную чувствительность в ИК-диапазоне.

Особенностью скетча является отсутствие в нём функций **setup** и **loop**. Оказывается, они для Arduino не обязательны! Дело в том, что язык Wiring, на котором пишутся программы для Arduino, является надстройкой над платформой C/C++. Компиляция скетчей производится компиляторами C/C++, а не Wiring. Более того, если приглядеться к файлу **main.cpp**, который находится в папке **C:\Program Files\Arduino\hardware\arduino\avr\cores\arduino**, становится понятно, как и в какой последовательности на самом деле выполняются функции **setup** и **loop** (рис. 4).

Если в скетче Arduino обнаружится главная для языков C/C++ функция **main**, то управление передаётся на неё, а не на файл **main.cpp**. Как следствие, не будут выполняться ардуиновские инициализации и проверки. В этом случае программист берёт всю ответственность на себя, поэтому начинающим проводить такие эксперименты не рекомендуется. В чём выигрыш? В уменьшении длины скомпилированного кода программы и файла прошивки. Для сравнения, с функцией **main** — 186 байт, с функциями **setup** и **loop** — 484 байта.

ЛИТЕРАТУРА

1. Увеличение частоты ШИМ. — URL: <https://alexgyver.ru/lessons/pwm-overclock/> (10.11.21).
2. Отрицательная обратная связь как проявление отношения человека и предмета труда. — URL: <http://philosophystorm.org/diletant/2754> (10.11.21).
3. IR send problem #311. — URL: <https://github.com/Arduino-IRremote/Arduino-IRremote/issues/311> (10.11.21).

Источник питания с переключаемой полярностью

С. БИРЮКОВ, г. Москва

Для игрушечной железной дороги, изготовленной в ГДР более 50 лет назад, вместо тяжёлого блока питания с регулируемым трансформатором и селеновым выпрямителем было решено изготовить блок питания на современной элементной базе.

Для того чтобы поезд игрушечной железной дороги мог двигаться вперёд, и назад с регулируемой скоро-

стью, необходим источник питания постоянного тока с напряжением от 0 до 15 В и изменяемой полярностью.

Полярность должна переключаться автоматически при прохождении движком регулятора выходного напряжения среднего положения, при котором выходное напряжение равно нулю.

Схема разработанного источника приведена на рис. 1. Его основой служат, как и в зарядном устройстве, описанном в моей статье "Зарядное устройство ... из кубиков" ("Радио", 2021, № 12, с. 13, 14), преобразователь AC/DC сетевого напряжения в постоянное 24 В, 1,5 А (A1) и регулируемый понижающий преобразователь DC/DC на ток до 3 А (A2), такой же, как в упомянутой выше статье, приобретённые в интернет-магазине.

Преобразователь A2 выдаёт на своём выходе напряжение, которое обес-