

вывать в аудиофайлы с расширением .mp3. Для этого удобно пользоваться ресурсом <<https://voxworker.com/ru>>. Автор использовал сказку Эрнеста Ильина о роботе-мальчике, разбив её на четыре фрагмента [6].

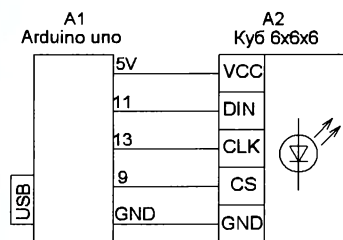


Рис. 6

Схема подключения сателлита—аниматроника показана на рис. 5, куба — на рис. 6. Подключения сохранены из предыдущих версий конструкций. Получив опыт в использовании Processing, можно самостоятельно создавать виртуальные версии собственных конструкций, дополняя их новыми функциональными возможностями, заменяя или синхронизируя работу с реальными.

#### ЛИТЕРАТУРА

1. Мамичев Д. Светодиодный куб 6x6x6 на Arduino. — Радио, 2018, № 1, с. 61—63.
2. Мамичев Д. Игрушка-аниматроник. — Радио, 2020, № 2, с. 60—62.

3. Arduino и Processing. Как управлять микроконтроллером по COM-порту. — URL: <https://habr.com/ru/post/450518/> (13.08.20).

4. Processing: Библиотеки/Sound/Класс SoundFile. — URL: [http://wikihandbk.com/wiki/Processing:Библиотеки/Sound/Класс\\_SoundFile](http://wikihandbk.com/wiki/Processing:Библиотеки/Sound/Класс_SoundFile) (13.08.20).

5. Processing: Библиотеки/Sound/Класс Amplitude. — URL: [http://wikihandbk.com/wiki/Processing:Библиотеки/Sound/Класс\\_Amplitude](http://wikihandbk.com/wiki/Processing:Библиотеки/Sound/Класс_Amplitude) (13.08.20).

6. Ernest Ilin. Сказка про Маленького Робота. — URL: <https://proza.ru/2002/09/06-128> (13.08.20).

От редакции. Скetch и файлы для проекта, а также видеозапись работы устройства находятся по адресу <http://ftp.radio.ru/pub/2021/02/virtual.zip> на нашем FTP-сервере.

## Ответы на викторину "Arduino: программная часть-2"

("Радио", 2021, № 1, с. 54, 55)

С. РЮМИК, г. Чернигов, Украина

**1. Ответ — 0.** Цифровые порты в Arduino двунаправленные. Следовательно, контакт A1:8, к которому подключается транзистор VT1 и реле K1, можно программно перевести из режима "Выход" в режим "Вход" с внутренним "pull-up" резистором. Этот резистор подключается одним выводом к цепи +5 В, другим — к выводу 8 Arduino. В результате образуется делитель напряжения, нижнее плечо которого составляют резистор R1 и открытый переход база—эмиттер транзистора VT1. Поскольку "pull-up" резистор высокоомный (20...50 кОм), напряжение на выводе 8 Arduino будет примерно на 0,1...0,2 В больше, чем напряжение на переходе транзистора VT1, т. е. 0,75...0,85 В, что воспринимается микроконтроллером как лог. 0.

**2. Ответ — 1.** Назначение скетча — циклическое включение и выключение реле K1 по следующему алгоритму: 2 с — включено, 8 с — пауза, 2 с — включено, 8 с — пауза и т. д. Длительность включённого состояния задаётся в строке 4 явным образом через количество миллисекунд, где число 2000 эквивалентно 2 с. Длительность паузы рассчитывается как разность между числами в строках 3 и 4, в итоге получается 10000 — 2000 = 8000 мс, или 8 с. Реле K1 включается

(выключается) напряжением лог. 1 (лог. 0) на выводе 8 Arduino. Смена логических уровней производится в строках 9, 10, где оператор **digitalWrite** имеет нестандартное заполнение. Внутри его скобок на месте, где указывается уровень, присутствует длинное логическое выражение **millis()%(allTime) < onTime**. Первые 2 с, до того как постоянно работающий счётчик миллисекунд **millis()** сравняется со значением, хранящимся в переменной **onTime**, логическое выражение истинно, оператор записи превращается в **digitalWrite(relayPin, 1)**, реле K1 включается. Как только в счётчике миллисекунд число 2000 будет превышено, логическое выражение станет ложным **digitalWrite(relayPin, 0)**, и реле K1 выключится.

Если поменять местами числа в строках 3 и 4, то счётчик миллисекунд никогда не сможет выйти за рамки 2000, следовательно, логическое выражение всегда будет истинным, и на выводе 8 Arduino установится уровень лог. 1. Реле K1 будет постоянно включено.

**3. Ответ — 0.** Назначение скетча — игра человека с Arduino в "Чёт-нечёт". Нажимая на клавиатуре компьютера клавиши 1 или 2, надо попы-

таться угадать, какое число "задумал" Arduino. Последний играет по-честному, генерируя случайные числа и не зная ходы человека. Теоретически на большой выборке поединков должна получиться боевая ничья.

Однако первая компиляция скетча приведёт к ошибке **expected initializer before 'Serial'** с подсветкой строки 3. Действительно, в ней пропущены фигурные скобки, должно быть **void setup() { Serial.begin(9600); }**. Но главный сюрприз ждёт дальше. Скетч, хоть и исправно компилируется, но при загрузке в Arduino "играет в поддавки". Если на клавиатуре компьютера постоянно вводить цифру 2, человеку выигрыш всегда обеспечен. Разгадка кроется в программном генераторе случайных чисел в строке 4. Правильная запись **a = random(2);**, поскольку функция **random()** генерирует целые числа от 0 до N-1, где N — число в скобках.

На этом ошибки, приводящие к неработоспособности программы, заканчиваются. Есть ещё одна неточность — нерациональное применение типа **long** в переменной **k** в строке 6. Здесь следовало бы указать тип **char**. Но к фатальным ошибкам в данном скетче это не относится (небрежность, недосмотр), хотя и приведёт к перерасходу памяти на 18 байт.

**4. Ответ — 0.** Назначение скетча — включение и выключение реле K1 с помощью шифра, состоящего из цифр, подаваемых с клавиатуры компьютера в среде Arduino IDE. Коды включения 123 и выключения 456 прописаны в строках 10, 11 скетча. Переменная **kod**, которая ими управляет, имеет размерность **unsigned int** (строка 9), следовательно, её значение всегда меньше, чем 65536. Получается, что шифр может состоять из четырёх любых цифр или из пяти



определённых цифр. Чтобы вводить шифр повышенной секретности из восьми цифр, следует задать для переменной **код** тип данных **unsigned long**.

**5. Ответ — 0.** При подаче между обкладками пьезоизлучателя BQ1 пульсирующего напряжения с частотой 1,5 кГц он будет издавать высокий звук, соответствующий ноте "соль" третьей октавы. В рассматриваемой схеме излучатель BQ1 и светодиод HL1 имеют общую точку соединения. Казалось бы, при мигании светодиода с частотой 0,5 Гц (два раза в секунду) звучание излучателя тоже должно быть прерывистым. Однако элемент BQ1 является униполярным. Он может подключаться вторым выводом как к цепи питания 5 В, так и к общему проводу GND. Мигание светодиода приводит лишь к периодической смене полярности пульсирующего напряжения, подаваемого на излучатель. Но результат будет одинаков — непрерывное звучание.

**6. Ответ — 1.** Назначение скетча — измерение температуры с помощью термистора RK1 и выдача результата на компьютер в градусах Цельсия. Температура рассчитывается по упрощённой формуле известного уравнения Стейнхарта — Харта [1]. В формуле фигурирует коэффициент В (константа **В**, **COEF** в строке 2), который указывается в справочных данных термистора RK1. Для разных типов термисторов он будет разным, причём с большим разбросом. К примеру, для термистора MMT-1 коэффициент температурной чувствительности В равен 2060...4300 К.

Если измеренная температура получается выше, чем показывает образцовый термометр, увеличивая константу **В**, **COEF** можно скомпенсировать погрешность. Для справки, градуировку в бытовых условиях обычно проводят в двух точках — в воде с кусочками льда 0 °С и в кипящей воде при 100 °С. В последнем случае не лишним будет учесть высоту местности над уровнем моря согласно таблице.

**7. Ответ — 1.** Назначение скетча — управление нагревательным элементом, регулирующим температуру в термостате. Поскольку реального нагревателя нет, его наличие имитирует светодиод HL1. Он светит, если температура выше нормы (нагреватель отключается), если погашен — температура ниже нормы (нагреватель включается). Границы изменения температуры в термостате задаются числами в относительных единицах, соответственно 510 и 490 в строках 9, 11. С ними сравниваются результаты измерения напряжения функцией **analogRead()**. Чем выше

температура термистора, тем меньше считываемое число, и наоборот.

Если считываемое число находится в промежутке между 490 и 510 (в вопросе это 500), в зависимости от направления процесса — остывание или нагрев, напряжение на выходе 8 Arduino может быть как лог. 0, так и лог. 1.

**8. Ответ — 1.** Назначение скетча — выполнить фильтрацию внешних помех при измерении температуры термистором RK1 через вход A0 Arduino. В скетче используется алгоритм "Бегущее среднее" (другое название — "Скользящее среднее" от англ. Running Average) [2]. Это простой и достаточно эффективный способ устранения наводок при замерах аналоговых напряжений, будь то термодатчика, фоторезистора или гигрометра. Настраивается фильтр подбором двух коэффициентов в строках 2 и 3. Первый из них **STEP** отвечает за число цифровых выборок в единицу времени. Частота следования выборок должна быть достаточно большой, чтобы захватить все основные гармоники в спектре сигнала согласно теореме Котельникова. Второй коэффициент **COEF** может изменяться от 0 до 1, при этом чем он меньше, тем лучше сглаживаются "иголки" в сигнале. В переводе на язык теории фильтров этот коэффициент определяет частоту среза цифрового ФНЧ.

**9. Ответ — 1.** Модуль датчика движения HC-SR501 не является новинкой для читателей журнала "Радио" [3]. Его невысокая стоимость и способность обнаруживать движущиеся тепловые объекты на рас-

стоянии до 3...7 м способствуют широкому применению в системах "умный дом", охранных сигнализациях, автоматизации быта. Напряжение питания на его выводе VCC рекомендуется 4,5...12 В. Однако выходной сигнал OUT имеет логические уровни 0 и 3,3 В, что наводит на мысль о возможности снижения напряжения питания до 3,3 В.

**10. Ответ — 1.** Назначение скетча — обнаружение движущихся объектов с выдачей в компьютер текстовой информации "Есть движение", "Нет движения". Как известно, на плате модуля HC-SR501 имеется перемычка, устанавливаемая в положение "Н" или "L". В зависимости от модификации модуля это может быть съёмная перемычка или паяное соединение между ламелями. Если перемычка находится в положении "L", то каждое движение объекта сопровождается формированием на выходе OUT отдельного импульса. Если перемычка — в положении "Н", то число движений объекта накапливается, уровень лог. 1 на выходе OUT держится до тех пор, пока объект не выйдет из зоны охраны или не замрёт на длительное время. Следовательно, скетч Arduino при положении перемычки "L" будет фиксировать большее количество движений, чем в положении перемычки "Н", но информативность для пользователя только ухудшится.

**11. Ответ — 0.** Назначение скетча — обнаружение движущихся объектов с включением внутреннего светодиода L на плате Arduino. Особенность — использование системы прерывания микроконтроллера, чтобы не пропустить начало импульса с выхода OUT модуля HC-SR501 и разгрузить процессорное ядро для выполнения других задач. Тип прерывания указывается при инициализации скетча в блоке **setup()**. Малоаметный ноль внутри скобок

Высота над уровнем моря, м	Температура кипения воды, °С
0	100
400	98,65
800	97,32
1200	96,02
1600	94,72

стоянии до 3...7 м способствуют широкому применению в системах "умный дом", охранных сигнализациях, автоматизации быта. Напряжение питания на его выводе VCC рекомендуется 4,5...12 В. Однако выходной сигнал OUT имеет логические уровни 0 и 3,3 В, что наводит на мысль о возможности снижения напряжения питания до 3,3 В.

На рис. 1 показано внутреннее устройство модуля HC-SR501. Логика работы следующая: PIR-датчик B1 преобразует ИК-излучение от удалённого объекта в электрический сигнал, который обрабатывается в специали-

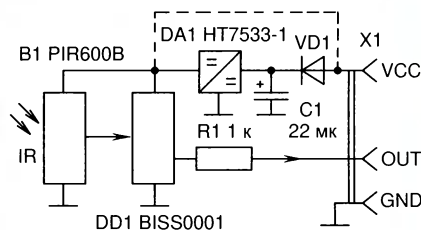


Рис. 1



функции `attachInterrupt()` в строке 6 указывает на тип прерывания INT0.

**12** ● Ответ — 0. Назначение скетча — подсчёт количества движений объекта (объектов) за одну минуту и вывод этой информации в компьютер для прорисовки на плоттере в среде Arduino IDE. Применяться может, например, для оценки трафика движения в людных местах, а в быту — как трекер фаз сна. В последнем случае учитывается, что на протяжении ночи происходит смена фаз так называемого быстрого и медленного сна. В зависимости от того, в какую из фаз человек проснулся, утро может быть как добрым, так и не очень ("встал не с той ноги"). Во время медленного сна человек, как правило, лежит неподвижно, сон крепкий, просыпаться тяжело, будить не рекомендуется. Во время быстрой фазы человек часто видит сны, может переворачиваться с боку на бок, но сон поверхностный и пробуждение будет лёгким.

Скетч Arduino подсчитывает число движений, которое сделал спящий человек в течение ночи. За это отвечает переменная `a` в строке 9. Переменная `count` отсчитывает промежутки времени, равные 60 с. Мерной линейкой служит число 1400000L в строке 12, подбирается экспериментально. При составлении своего скетча это число рекомендуется вынести в шапку программы отдельной константой, чтобы можно было оперативно задавать другие отрезки времени. Для примера на рис. 2 показан график сна с пятиминутными подсчётами числа движений.

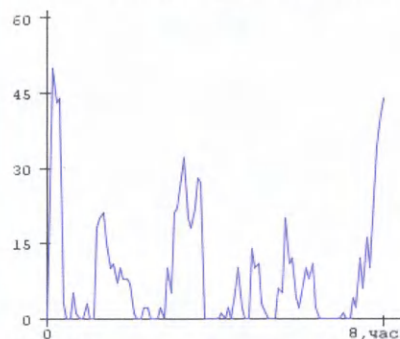


Рис. 2

Чем выше пик на графике, тем чаще человек ворочался, что указывает на быструю фазу сна, и наоборот.

**13** ● Ответ — 1. Принцип работы и параметры ультразвукового дальномера HC-SR04 подробно освещены в [4]. Точность измерения, в первую очередь, зависит от температуры окружающей среды, поскольку она напрямую влияет на скорость распространения звуковых волн с ростом примерно 0,6 м/с на градус. Об этом знают все. А вот про влияние атмосферного

давления и относительной влажности воздуха информации меньше.

Согласно справочным данным ГСССД МР 176-2010 ("Расчётное определение скорости звука во влажном воздухе..."), скорость звука повышается с увеличением относительной влажности воздуха. Разница при показаниях гигрометра 10 % и 90 % составляет около 1 м/с при комнатной температуре. Для сравнения, такой же результат обеспечивает изменение температуры окружающего воздуха всего лишь на 1,5 °C.

Атмосферное давление связывают со скоростью звука в барографических таблицах, но там свой вклад вносит спад температуры воздуха в зависимости от высоты над уровнем моря. Давление начинает заметно влиять на скорость звука при кратном (в 5...10 раз) его изменении. Этот фактор учитывают на производстве при испытаниях в барокамерах. В естественных погодных условиях влиянием атмосферного давления обычно пренебрегают.

**14** ● Ответ — 0. Назначение скетча — измерение расстояния до препятствия модулем HC-SR04 и вывод информации в компьютер. Скетч составлен без применения библиотек функций с помощью прямого формирования зондирующего сигнала длительностью 10 мкс (строки 7—9) и приёма отражённого сигнала в строке 10.

В строке 11 скетча в формуле для расчёта дистанции принята скорость распространения звука 343 м/с (коэффициент 0,0343). Эта скорость, согласно справочных таблиц, подразумевает температуру окружающей среды +20 °C. При температуре +25 °C скорость составляет 346 м/с, следовательно, если имелась в виду эта температура, в формуле был бы коэффициент 0,0346.

Важно понимать, что формула расчёта расстояния, применённая в этом и многих других простейших скетчах в Интернете, не учитывает геометрию распространения излучаемой и отражённой волн. В исследованиях [5] показано, что при расстоянии 2 см до объекта погрешность составляет 25 %, а при измерении расстояния 400 см погрешность снижается до 0,0007 %.

**15** ● Ответ — 0. Назначение скетча — аналогично вопросу 14, но с применением библиотеки функций [6]. Инициализация библиотечных параметров производится в строке 3, где числа в скобках означают: 13 — номер вывода Arduino, к которому подключается сигнал `Trig` модуля HC-SR04; 12 — то же, но для сигнала `Echo`; 21 — температура окружающей среды в градусах Цельсия; 250 — максимально измеряемое расстояние в сантиметрах. Строка 11, упоминаемая в вопро-

се, содержит оператор `Serial.println()`, который посылает информацию о дистанции `d` в компьютер.

Если сравнивать функциональную роль операторов в строках 3 и 11, то видно, что точность измерения дистанции определяется температурой в строке 3, в отличие от строки 11, где число 1 указывает лишь на точность округления результата до десятых долей сантиметра.

**16** ● Ответ — 0. Назначение скетча — использование модуля HC-SR04 в качестве датчика присутствия активного типа. Логика включения и выключения светодиодов HL1, HL2, а также пьезоизлучателя с встроенным генератором HA1 точно такая же, как в конструкции [4]. Нижняя граница дистанции, начиная с которой происходит обнаружение объекта, задают в строке 10 числом 50 в сантиметрах. Верхнюю границу 300 см указывают при инициализации библиотеки в строке 3 и ещё раз повторяют в строке 10 в виде числа 300. Зачем? Дело в том, что применяемая в скетче библиотека "HCSR04" отсекает дистанцию 0...4 см, как дающую недостоверные показания, и вместо 0...4 выдаёт сверхбольшое число для облегчения идентификации события. Следовательно, если в строке 10 проверить только нижний предел 50 см (как звучит в вопросе), но не проверять верхний предел 300 см, то при малых дистанциях до объекта устройство перестанет его "видеть".

#### ЛИТЕРАТУРА

1. Термистор и Arduino. — URL: <http://arduino-diy.com/arduino-thermistor> (11.11.20).
2. Полезные алгоритмы для Arduino. Фильтры значений. Бегущее среднее (Running Average). — URL: <http://alexgyver.ru/arduino-algorithms/> (11.11.20).
3. Савченко А. Автомат управления лестничным освещением с датчиком движения. — Радио, 2017, № 1, с. 47—48.
4. Нечаев И. Ультразвуковой дальномер без микроконтроллера. — Радио, 2018, № 4, с. 59—62.
5. Жмудь В. и др. Ультразвуковой датчик измерения расстояния HC-SR04. — Автоматика и программная инженерия, 2017, № 4 (22), с. 18—25.
6. HCSR04. — URL: <https://github.com/enjoyneering/HCSR04> (11.11.20).

**От редакции.** Скетчи программ, описание дальномера HC-SR04 и справочные данные о скорости звука в зависимости от влажности и температуры воздуха находятся по адресу <http://ftp.radio.ru/pub/2021/02/arduino2.zip> на нашем FTP-сервере.