

Рис. 3

жение, близкое к нулю. В результате на выходе ОУ DA1.4 импульсы исчезнут, и установится напряжение, близкое к напряжению питания, поэтому транзистор VT1 будет постоянно открыт, а светодиодная лента светить с максимальной яркостью. Когда на выходе ИК-датчика появится низкий уровень напряжения, процесс пойдёт в обратном порядке и светодиодная лента плавно погаснет.

Если применить выводные элементы, а микросхему — в корпусе DIP-14, их можно разместить на односторонней печатной плате из фольгированного стеклотекстолита, чертёж которой показан на рис. 2. Здесь можно применить резисторы МЛТ, С2-23, конденсаторы — К10-17 или импортные, клеммники — винтовые с шагом контактов 5 мм. Транзистор — MOSFET n-канальный мощный с напряжением открывания 4...5 В. Можно применить транзистор с неис-

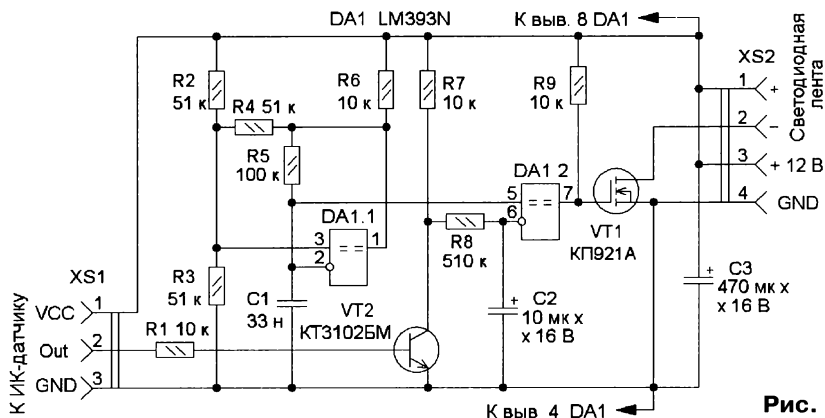


Рис. 4

правных материнских плат. Даже при коммутации тока 4...5 А теплоотвод для транзистора не требуется, так как транзистор управляется импульсами с достаточно высоким напряжением на затворе (около 12 В) и с крутыми фронтами.

Можно применить микросхему (LM324DR), диод (SM4001, LL4001), резисторы и конденсаторы для поверхностного монтажа, их можно разместить на печатной плате, чертёж которой показан на рис. 3. Обозначение "0" — это перемычка для поверхностного монтажа.

Устройство налаживания не требует. Номиналы элементов могут отличаться от указанных на схеме на  $\pm 50\%$ . При этом должны выполняться условия  $R1 = R2 + R3$  и  $R3 = (R1 + R2)/10$ , небольшое напряжение на этом резисторе, приложенное к выводу 5, устраняет влияние помех при низком уровне напряжения на выходе ИК-датчика.

Аналогичное устройство можно собрать на основе микросхемы сдвоенного компаратора LM393 (рис. 4). Работает оно аналогичным образом. На компараторе DA1.1 собран генератор пилообразного напряжения. Цепь R8C2 формирует плавно спадающее или нарастающее напряжение. Подбором элементов этой цепи можно установить

желаемое время включения/выключения светильника. Для этого устройства печатная плата не разрабатывалась, была использована макетная плата с частично навесным монтажом для проверки работы устройства.

При подборе блока питания надо измерить реальное потребление тока светодиодной лентой. Например, у приобретённой мной ленты с маркировкой 14,4 Вт/м метровый отрезок реально потреблял только 0,7 А вместо 1,2 А. Но блок питания желательно выбирать с небольшим запасом.

Несколько устройств, собранных по этим схемам, безотказно работают уже нескольких лет.

## ЛИТЕРАТУРА

1. Обзор инфракрасного датчика движения HC-SR501. — URL: <https://robotchip.ru/obzor-infrakrasnogo-datchika-dvizheniya-hc-sr501/> (04.06.21).
2. Как работает PIR-датчик HC-SR501, и его взаимодействие с Arduino. — URL: <https://radioprogram.ru/post/734> (04.06.21).

От редакции. Чертежи печатных плат находятся по адресу <http://ftp.radio.ru/pub/2021/08/lenta.zip> на нашем FTP-сервере.

## Ответы на викторину "Arduino: программная часть-5"

("Радио", 2021, № 7, с. 63, 64)

С. РЮМИК, г. Чернигов, Украина

**1** Ответ — 1. Плату Arduino UNO после покупки следует проверить на работоспособность от всех

источников питания: USB, +5 В, VIN. В последнем случае на сайте разработчика [1] указываются два интервала

входных напряжений: 6...20 В (limit) и 7...12 В (recommended).

Перевод названия "limit" означает предельные, а "recommended" — рекомендуемые параметры. Физическим ограничением служит линейный стабилизатор напряжения, который находится внутри Arduino и понижает входное напряжение VIN до 5 В. Чем больше падение напряжения на стабилизаторе, тем выше потери мощности и выше шанс теплового перегрева его корпуса, особенно при большой токовой нагрузке по линиям портов. Чем меньше падение напряжения, тем ниже стабильность питания и хуже подавление помех при импульсных нагрузках.



**2. Ответ — 1.** Назначение скетча — тест производительности Arduino по методике [2]. Вначале фиксируется время старта в строке 5. Далее в течение нескольких десятков секунд выполняются более 100 млн арифметических и логических операций в строках 6—8. По окончании фиксируется время финиша в строке 9.

На экран компьютера в строке 11 выводится число миллисекунд, прошедших от старта до финиша. В реальном тесте было получено 31435 мс, но это число может отличаться для разных моделей Arduino. Чем время меньше, тем быстрее Arduino выполняет целочисленные расчёты.

"Лишний" оператор в строке 10 — это вынужденная мера для защиты от сверхбыстрого компилятора, который предполагает, что выполнение математических действий в строках 6—8 лишено здравого смысла. Печать содержимого переменной *j* в строке 10 должна "убедить" компилятор в обратном. Если строку 10 закомментировать, то строки 6—8 будут проигнорированы компилятором как бесполезные, и тест закончится, даже не начавшись, с нулевым числом миллисекунд.

**3. Ответ — 1.** Назначение скетча — проверка точности математических операций для чисел с плавающей точкой [3]. Известно, что микроконтроллеры семейства AVR, составляющие основу плат Arduino, на аппаратном уровне не поддерживают арифметические операции с дробными числами формата **float**. Такие вычисления реализуются программно. Как следствие, работает это не так быстро, как хотелось бы, появляются неточности округления в расчётах, снижается объём доступной памяти программ.

```
***Arduino float***
a = 0.20000004
b = 0.20000000
(a == b)? No
```

Рис. 1

В рассматриваемом примере в строке 5 после операции вычитания в переменной *a*, по идее, должно оставаться число  $1,2 - 1,0 = 0,2$ . В строке 6 переменной *b* для образца также присваивается число 0,2. Но вердикт в строке 10 однозначный — "No", т. е. числа в переменных *a* и *b* не равны друг другу. Как такое может быть?

Разгадка кроется в точности индикации чисел на экране компьютера. Если в строках 7 и 8 изменить опера-

торы печати **Serial.println(a,8)**, **Serial.println(b,8)**, то при восьми знаках после запятой будет видно, что переменная *a* на самом деле равна 0,20000004 (рис. 1). Это результат округления дробных чисел. Данный нюанс не зависит от страны изготовления платы Arduino. К нему надо приспособиться, а в своих программах не применять бездумно числа в формате **float**.

**4. Ответ — 0.** Внешняя кнопка сброса SB1 на электрической схеме подключается параллельно внутренней кнопке сброса SB2 и параллельно входу сброса микроконтроллера ATmega328 в Arduino. Идея в том, чтобы протестировать целостность связи между выводом RESET микроконтроллера и одноимённым гнездом контактной колодки Arduino. То есть, чтобы исключить ситуацию, когда внутренней кнопкой SB2 программа сбрасывается в начало, а внешней кнопкой SB1 — нет.

**5. Ответ — 1.** Назначение скетча — организовать мигание светодиода HL1 с периодом 2 с и одновременно вести подсчёт смены состояний в переменной **counter**. Задача эксперимента — в произвольный момент времени нажать на кнопку сброса SB1 или SB2 и наблюдать за показаниями счётчика в мониторе порта Arduino IDE. В момент нажатия на кнопку сброса счётчик должен остановиться, а в момент отпущения — возобновить счёт заново с нулевого состояния. Это хорошо согласуется с физикой процессов, когда во время подачи низкого уровня на вход RESET микроконтроллера ATmega328 последний находится в спящем состоянии и "просыпается" только после перехода низкого уровня в высокий.

**6. Ответ — 1.** Назначение скетча — определить время реакции человека на нажатие на кнопку сброса Arduino. Для этого применяется следующая процедура. Сразу после старта программы на мониторе компьютера появляется сообщение о том, что надо подождать 2—4 с (строка 8). Далее в компьютер по последовательному интерфейсу передаётся состояние инкрементного счётчика counter (строка 12). Как только будет нажата кнопка сброса SB1 или SB2, показания счётчика останавливаются. Чем меньше число в счётчике, тем быстрее реакция человека.

Как известно, время простой реакции на световой раздражитель зависит от возраста человека, его физического и психологического состояния и обычно составляет 100...300 мс. В переводе на показания счётчика **counter** должно получиться примерно 30—90 единиц.

Время единичного приращения счётчика **counter** определяется скоростью выдачи информации в компьютер. Задаётся она в строке 7, где прописана символьная скорость 9600 бод. Это характеристика полной информационной ёмкости канала, включая служебные символы. По умолчанию оператор **Serial.begin(9600)** устанавливается режим передачи 8-N-1 с одним стартовым и одним стоповым битами. Итого на передачу одного байта полезной информации (8 бит) уходит 10 бит данных. Общее время передачи байта составляет  $10/9600[\text{бод}] = 0,00104$  с или 1,04 мс.

Оператор **Serial.println(counter++)** в строке 12 пересылается не одним, а несколькими байтами подряд, поскольку к числовой информации счётчика добавляются вспомогательные символы возврата каретки ( $\backslash r$ , код 13) и перевода строки ( $\backslash n$ , код 10). Итого передаются 3 байта для однозначных, 4 байта для двузначных, 5 байтов для трёхзначных чисел. Следовательно, на однократное выполнение команды в строке 12 потребуется время не менее 3 мс, что и является ответом на вопрос.

**7. Ответ — 0.** Вывод AREF на контактной колодке Arduino обычно оставляют свободным. Его используют, если требуется повысить точность измерений в канале АЦП. Для этого на вывод AREF подают внешнее напряжение, к примеру, от внутреннего стабилизатора 3,3 В Arduino. Во избежание повреждения микроконтроллера при ошибках в программе рекомендуется подключать вывод AREF через токоограничивающий резистор (в данном случае это R1).

Однако внутреннее сопротивление канала AREF (Rref) в режиме работы от внешнего ИОН не бесконечно. Если быть точным, то для микроконтроллера ATmega328, согласно справочным данным [4], разброс Rref составляет 25,6...38,4 кОм, типовое значение — 32 кОм.

В рассматриваемой схеме резистор R1 совместно с внутренним резистором Rref образуют делитель, напряжение в средней точке которого рассчитывается по формуле  $AREF[B] = 3,3[B] \cdot Rref[k\Omega] / (R1[k\Omega] + Rref[k\Omega]) = 3,3 \cdot 32 / (4,7 + 32) = 2,87$  В.

Реально измеренное напряжение может отличаться от расчётного, но на небольшую величину. Здесь следует понимать, что для получения напряжения на выводе AREF ровно 3,3 В необходимо закортить резистор R1. Правда, при этом надо быть абсолютно уверенным, что скетч написан без ошибок. Например, что в нём отсутствует оператор **analogReference**, пе-



реключающий в процессе работы режим AREF с внешнего ИОН на внутренний источник.

**8.** Ответ — 1. Назначение скетча — измерение аналогового напряжения на произвольно выбранном пине A0 через канал АЦП. В качестве ИОН используется стабильное напряжение, подаваемое через резистор R1 на вывод AREF.

В исходном состоянии при наличии резистора R1 показания переменной **sensor** будут колебаться от 0 до 1023 единиц из-за помех, которые наводятся на "висящий в воздухе" вывод A0. Реальные измерения показывают 400—700 единиц.

Если удалить резистор R1, то вывод AREF будет привязан к общему проводу через внутреннее сопротивление Rref 32 кОм. Следовательно, напряжение на выводе AREF будет близким к нулю, поэтому любая помеха на входе A0 будет восприниматься как максимально большое напряжение и оцифровываться с уровнем 1023 единицы.

**9.** Ответ — 1. Назначение скетча — измерить аналоговое напряжение на контакте A0 при двух напряжениях AREF: 5 В (напряжение питания Arduino), 2,8 В (от внешнего источника 3,3 В). Переключение режимов производится в строках 6 и 9. Далее делается пауза на 2 с (строки 7, 10) и производится замер напряжения через канал АЦП в строках 8 и 12. Разница между замерами в двух режимах в относительных единицах должна быть близкой к отношению напряжений на выводе AREF, а именно,  $5/2,8 = 1,78$ .

На практике отношение получается 1,05...1,1, т. е. показания не достоверные. Причина такого эффекта указана в справочных данных на микроконтроллер ATmega328 [4], где рекомендуется после смены ИОН проводить несколько "пустых" замеров в канале АЦП для стабилизации показаний. И действительно, если в строках 7 и 10 до и после оператора задержки **delay(2000)** вставить по одному оператору **analogRead(A0)**, результаты измерений приходят в норму. При этом для чистоты эксперимента на контакт A0 должно быть подано фиксированное напряжение — пусть это будет 2,5 В от делителя, состоящего из двух резисторов сопротивлением 10 кОм, подключённых к линии питания 5 В и общему проводу.

**10.** Ответ — 0. Аналоговые и цифровые линии портов Arduino могут работать как на вход, так и на выход. Возникает идея подключить радиоэлементы к выбранному контакту разъёма Arduino так, чтобы без перепайки провести комплексную

проверку приёмного и передающего трактов. Разными будут лишь управляющие программы.

Один из возможных вариантов тестового узла содержит два резистора и один светодиод [5]. Для проверки выходов Arduino микроконтроллер должен генерировать импульсные сигналы. Они будут индифицироваться двухцветным светодиодом HL1 при установке движка переменного резистора R2 в крайние положения. Для проверки входов Arduino переменный резистор R2 превращается в датчик напряжения, а микроконтроллер его измеряет на избранном контакте. Резистор R1 ограничивает ток через светодиод HL1.

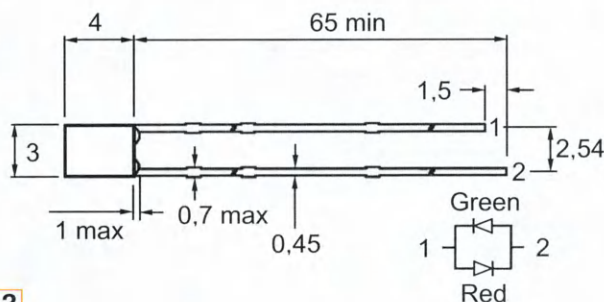


Рис. 2

Если заменить светодиод миллиамперметром, по входу он будет эквивалентен низкоомному резистору. А раз так, то сигналы от датчика напряжения на резисторе R2 будут беспрепятственно проходить к микроконтроллеру Arduino. Вывод — проверки возможны.

**11.** Ответ — 1. Назначение скетча — регулирование скважности импульсов, генерируемых на контакте D9 по командам от компьютера. В мониторе порта Arduino IDE с клавиатуры вводятся цифры 0—9, которые принимаются на противоположной стороне в виде чисел 48—57 в строках 8, 9 скетча. Переменная **cpu** устанавливает относительную скважность ШИМ 25—250 в строке 10. Соответственно, индикатор HL1 будет светиться с разной яркостью красным или зелёным цветом.

Можно ли определить, каким именно цветом будет светить светодиод HL1, если движок переменного резистора R2 устанавливается в верхнее по схеме положение? При существующей схеме этого сделать нельзя, ведь у светодиода HL1 не указан тип и не обозначена цоколёвка выводов. Например, если бы применялся светодиод L-91A6SURKCGKW (Kingbright), по его справочным данным легко идентифицировать выводы 1 и 2 (рис. 2), которые однозначно определяют, где красный, а где зелёный излучатель.

В рассматриваемом узле расположение излучателей в светодиоде HL1 не принципиально, ведь схема тестовая. Однако в других случаях следует обязательно указывать цифры 1 и 2 возле условного графического обозначения светодиода для исключения неоднозначности толкования.

**12.** Ответ — 1. Назначение скетча — опрос состояния цифрового входа D9 и вывод информации в виде цифр 0 или 1 в компьютер. Переключение уровней производится переменным резистором R2, который служит своеобразным двухпозиционным переключателем. В верхнем по схеме положении его

движка формируется лог. 1, в нижнем — лог. 0.

В режиме входа без внутреннего подтягивающего резистора сопротивление линии порта микроконтроллера относительно общего провода составляет мегаомы. Следовательно, ток, протекающий через светодиод HL1, будет недостаточным для его свечения. Однако тестирование входа D9 происходит нормально, и его можно наблюдать визуально в "Плоттере по последовательному соединению" в закладке "Инструменты" Arduino IDE.

#### ЛИТЕРАТУРА

1. Uno Rev3. — URL: <https://store.arduino.cc/arduino-uno-rev3> (10.05.21).
2. Мокн С. Программируем Arduino: профессиональная работа со скетчами. — СПб.: Питер, 2017.
3. Этюды для начинающих: Тип float. Как не утонуть? — URL: <http://arduino.ru/forum/programirovanie/etyudy-dlya-nachinayushchikh-tip-float-kak-ne-utonyut> (10.05.21).
4. ATmega328P. Datasheet. — URL: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) (10.05.21).
5. Рюмик С. М. 1000 и одна микроконтроллерная схема. Выпуск 3. — М.: ДМК-Пресс, 2016.