

## ИК-барьер на микроконтроллере

**Инфракрасным барьером (далее просто ИК-барьером) называют устройство оптической охраны, способное формировать невидимую человеческому глазу оптическую преграду, пересечение которой приводит к срабатыванию сигнализации. Обычно данные устройства изготавливаются на цифровых микросхемах малой степени интеграции. Как результат — большие габариты устройств, сложные схемотехнические решения, десятки элементов, а отсюда и небольшая надежность.**

Еще одной особенностью устройств данного вида является то, что, как правило, передатчик и приемник изготавливаются в виде отдельных блоков с раздельным питанием от батарей. Но на практике сложно обеспечить своевременную замену элементов питания, а также синхронное включение обоих устройств без дополнительных связей. В описываемом устройстве питание обеих частей обеспечивается от одного источника питания, с которым они соединяются при помощи проводов, в результате обеспечивается одновременное включение устройства на нужный интервал времени. Также стоит сказать несколько слов об ИК-коде, который передается передатчиком и принимается приемником устройства. В простых устройствах, чаще всего, это импульс, следующий с определенным периодом. Ясно, что ни о какой защищенности такого кода (точнее импульса) не может быть и речи, так как данные устройства срабатывают даже от фоновых засветок.

Все вышеперечисленное побудило разработать устройство оптической охранной сигнализации, лишенное этих недостатков. Конечно, такого рода сигнализацию сложно представить себе без микроконтроллера. Применение микроконтроллера позволяет сильно упростить схему самого устройства, повысить его надежность, а также за счет применения более сложного ИК-кода повысить защиту системы.

Выбор был сделан в пользу микроконтроллера AT90S2313 фирмы Atmel. Почему был выбран именно этот микроконтроллер? Дело в том, что в состав этого микроконтроллера входит универсальный асинхронный приемо/передатчик (UART), при помощи которого можно легко организовать приемо/передатчик ИК-сигналов с минимальными затратами ресурсов процессора, которые могут пригодиться при построении на базе ИК-барьера полноценной системы сигнализации.

Принципиальная схема устройства приведена на рис. 1. К выводу 11 (PD6) подключен светодиод HL2, который сигнализирует о нормальной работе устройства. ИК-светодиод HL1, который, собственно, и является излучателем ИК-кода, через токограничительный резистор R1 подключен к выводу 8 (PD4) микроконтроллера. Переданный код принимается стандартным ИК-приемником DA1. Усиленный и инвертированный сигнал с выхода приемника подается на вывод 2 (PD0/TXD) микроконтроллера. Кроме этого, в состав устройства входит пьезоизлучатель BA1 и кнопка SB1 "Вкл.". Первый является звуковым сигнализатором, сигнал на который подается с выхода внутреннего ШИМ микроконтроллера (вывод 15). Кнопка "Вкл." подключена к выводу 6 (PD2/INT0) и используется для включения/выключения устройства. При выключении микроконтроллер переходит в режим Sleep. Энергопотребление процессора в этом режиме не превышает 10 мА.

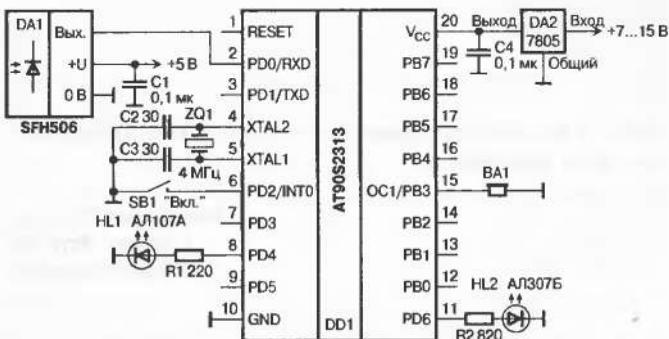


Рис. 1

Алгоритм работы устройства следующий. Сформированный байт код переписывается в регистр UDR передатчика UART. В это же время запускается подпрограмма, которая анализирует логическое состояние вывода TXD (выхода передатчика UART). При наличии низкого логического уровня запускается ШИМ (широко-импульсный модулятор), подключенный к выводу PD4 и построенный с использованием прерывания таймера/счетчика 0. Частота ШИМ равна частоте резонанса стандартного ИК-приемника кода RC-5 (36 кГц). При появлении на выводе TXD сигнала высокого уровня ШИМ выключается и на выводе PD4 устанавливается низкий логический уровень, при котором ИК-светодиод не излучает. Таким образом происходит модуляция каждого бита кода, а битов «старт» и «стоп», необходимых для правильной работы UART. Одновременно с передачей ИК-кода ведется также его прием, это возможно благодаря тому, что UART микроконтроллера AT90S2313 может работать в дуплексном режиме, то есть вести одновременно как передачу сигнала, так и его прием. Далее принятый байт сравнивается с переданным байтом. Если эти байты совпадают, то цикл передачи повторяется с той лишь разницей, что передаваемый байт изменяется. Менять код (байт) после каждого цикла передачи можно по любому математическому закону или вообще синтезировать случайным образом. В данном случае код изменяется простым суммированием единицы и предыдущего значения байта. Если же переданный и принятый байты не совпадают, а такое может случиться, если на оптической линии передачи сигнала появилась преграда, то включается сигнализация. Аналогичное включение сигнализации произойдет, если UART вообще не принял байт. Если по истечении времени работы сигнализатора (около 5 с) линия оптической связи будет восстановлена (исчезнет преграда), то система снова перейдет в режим охраны.

Программа устройства написана на языке программирования Ассемблер и приведена ниже.

```
.INCLUDE <2313def.inc>
=====
; Программа «ИК-барьера»
; Написана для AT90S2313.
=====
.org 000
    rjmp reset ; Вектор сброса
.org 001
    rjmp ext_int0 ; Вектор прерываний от вывода INT0
.org 006
    rjmp tim_ovf0 ; Вектор прерываний от таймера/счетчика 0.
=====
; Подпрограмма обработки прерываний от вывода INT0.
=====
ext_int0:
    ldi r31, 25
    rcall pause ; Вызываем подпрограмму задержки
    reti ; Выход из подпрограммы обработки прерываний
=====
; Подпрограмма обработки прерываний от таймера/счетчика 0.
=====
tim_ovf0:
    out tcnt0, r18 ; Загружаем таймер длительностью полупериода частоты модуляции
    inc r19 ; Увеличиваем значение регистра на единицу
    sbrc r19, 0 ; Проверяем на четность младший бит
```

```

r19    cbi portd,4 ;Устанавливаем на выводе PD4 логический
0      sbrs r19,0      ;Проверяем на четность младший бит
r19    sbi portd,4 ;Устанавливаем на выводе PD4 логический
1      reti          ;Выход из подпрограммы обработки прерывания
=====
; С этого места собственно и начинается основная программа.
=====
reset:
    ldi r16,0b01010010
    out ddrd,r16 ;Устанавливаем направление работы порта D
    ldi r16,0b01101100
    out portd,r16 ;Включаем подтягивающие резисторы
    ldi r16,0b00001000
    out ddrb,r16 ;Устанавливаем направление работы порта B
    ldi r16,0b11110111
    out portb,r16 ;Устанавливаем соответствующие уровни на выходах порта B
    ldi r16,$80
    out spl,r16 ;Инициализируем стек
    ldi r16,0b00011000
    out ucr,r16 :Разрешаем работу приемника и передатчика UART
    ldi r16,$ff
    out ubrr,r16 ;Задаем скорость работы UART
    ldi r16,$1
    out tccr0,r16 ;Синхронизируем работу таймера/счетчика 0 напрямую от тактового генератора
    ldi r16,$1
    out tccr1b,r16 :Синхронизируем работу таймера/счетчика 1
    ldi r16,$30
    out mcucr,r16 ;Разрешаем энергосберегающий режим работы микроконтроллера
    ldi r18,$cb ;Задаем полупериод частоты модуляции
    sei           ;Разрешаем прерывания
    clr r1       ;Очищаем регистр
start:
    sbis pind,2 ;Проверяем, не нажата ли кнопка
    rjmp sleep ;Если нажата, то переходим на метку sleep
    clr r2       ;Очищаем регистр
    inc r1       ;Увеличиваем значение регистра на единицу
    rcall uart   ;Вызываем подпрограмму модуляции сигнала
    sbis usr,rxc ;Проверяем, принял ли приемник UART переданный байт
    rjmp signal ;Если нет, то включаем сигнализацию
    sbic usr,fe   ;Проверяем, не было ли ошибки кадра
    rjmp signal ;Если была ошибка кадра, то включаем сигнализацию
    in r2,udr     ;Переписываем значение из приемника UART в регистр
    cpse r1,r2     ;Сравниваем переданный и принятый байт
    rjmp signal_1 ;Если не равно, то включаем сигнализацию
    ldi r31,1
    rcall pause
    rjmp start    ;Возвращаемся в начало цикла
=====
signal:  in r2,udr      ;Переписываем значение из приемника UART
signal_1: ldi r25,3      ;Задаем число сигналов сигнализации
signal_2: ldi r16,0b11000001
          out tccr1a,r16 ;Разрешаем работу ШИМ
=====
cbi portd,6 ;Гасим светодиод
ldi r16,170
out ocr1al,r16 ;Задаем частоту ШИМ
ldi r31,10      ;Задаем время задержки
rcall pause    ;Вызываем подпрограмму задержки
clr r16
out tccr1a,r16 ;Запрещаем работу ШИМ
sbi portd,6 ;Включаем светодиод
ldi r31,5       ;Задаем время задержки
rcall pause    ;Вызываем подпрограмму задержки
dec r25        ;Отнимаем единицу от значения регистра
breq start    ;Если нулевое значение, то идем на метку start
rjmp signal_2 ;Если нет, то повторяем цикл
sleep:
    cbi portd,6 ;Гасим светодиод
    ldi r31,25
    rcall pause ;Вызываем подпрограмму задержки
    ldi r16,$40
    out gimsk,r16 ;Разрешаем прерывания от вывода INTO
    sleep         ;Переходим в энергосберегающий режим
    sbi portd,6 ;Включаем светодиод
    ldi r16,$0
    out gimsk,r16 ;Запрещаем прерывания от вывода INTO
    rjmp start    ;Возвращаемся на start
=====
; Подпрограмма модуляции и передачи байта UART.
; =====
uart:
    out udr,r1      ;Загружаем байт в регистр передатчика UART
    up:
        out tcnt0,r18 ;Загружаем таймер длительностью полупериода частоты модуляции
        ldi r16,$2
        out timsk,r16 ;Разрешаем прерывания по переполнению таймера/счетчика 0
        up_1:
            sbis pind,1 ;Проверяем логический уровень на выводе PD1
            rjmp up_1
            ldi r16,$0
            out timsk,r16 ;Запрещаем прерывания по переполнению таймера/счетчика 0
            cbi portd,4 ;Устанавливаем логический 0 на выводе PD4
            down:
                sbis pind,1 ;Проверяем логический уровень на выводе PD1
                rjmp up
                sbic usr,txc ;Проверяем, не установлен ли флаг окончания передачи
                rjmp exit
                rjmp down
exit:
    sbi usr,txc      ;Сбрасываем флаг окончания передачи
    ret
; =====
; Подпрограмма задержки.
; =====
pause:
d_3: clr r30
d_2: clr r29
d_1: dec r29
    brne d_1
    dec r30
    brne d_2
    dec r31
    brne d_3
    ret
.EXIT

```

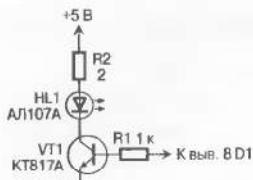


Рис. 2

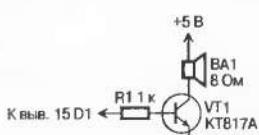


Рис. 3

Устройство размещено на плате размерами 50×50 мм. ИК-светодиод и ИК-приемник размещают в удобных местах, а к плате подключают при помощи соединительных кабелей (длиной до 2 м).

При помощи устройства, схема которого представлена на рис. 1, можно организовать оптическую линию охраны длиной до 2 м. Если требуется покрыть большее расстояние, можно воспользоваться доработкой, представленной на рис. 2. Еще одним вариантом увеличения расстояния оптической линии связи является использование в качестве излучателя ИК-лазера. В самом простом исполнении это может быть обычная лазерная указка. Как правило, данные указки уже имеют в своем составе гасящий резистор (около 200 Ом), поэтому ее подключение сводится к подсоединению выводов питания указки к выводам микроконтроллера. В качестве звукового сигнализатора для увеличения громкости можно использовать обычную динамическую головку (рис. 3).

Хочется заметить, что устройство получилось очень надежным. За полгода работы не наблюдалось каких-либо существенных сбоев. Также, благодаря наличию сложного кода устройству не страшны ни ИК-помехи, ни различные засветки. Как уже выше упоминалось, на базе устройства очень легко мож-

но построить полноценную систему сигнализации. Этому способствует наличие большого числа свободных выводов микроконтроллера, к которым можно подключить необходимые датчики (например, герконы, контактные датчики и др.).

В заключение нужно отметить, что освоение этого метода передачи данных UART с помощью ИК-излучения открывает большие перспективы. В первую очередь, это обмен информацией между микроконтроллерными устройствами или микроконтроллером и персональным компьютером в тех случаях, когда обычное соединение с помощью проводов невозможно. Вторая область видимого применения данного метода — это организация ИК-управления различными устройствами. Прием ИК-кода UART практически не отнимает полезных ресурсов контроллера, в то время как для приема других стандартов необходимо писать сложные декодеры.

Также было замечено, что ИК-приемо/передатчик, описываемый выше, хорошо работает на отражение. Используя его схему, можно построить устройство, аналогичное описанному в статье О. Николайчука «Интеллектуальный малогабаритный фотодатчик, работающий на отражение» («Схемотехника», 2001, № 6, с. 60—63) с меньшим числом деталей и сделать его более защищенным от фоновых засветок (так как передается не просто импульс, а сложный код). Более того, в результате экспериментов выяснилось, что чувствительность ИК-приемника сильно зависит от частоты модуляции ИК-сигнала, а это значит, что мы можем не только определить наличие объекта перед датчиком, но и с определенной точностью определить расстояние до этого объекта.

Марис Потапчук,  
mapic@online.com.ua

## Автоматический корректор часов

*В последнее время во многих регионах нашей страны происходит периодическое отключение электроэнергии, в результате чего выключаются все бытовые электроприборы, а электронные часы, не имеющие энергонезависимого режима работы, при этом сбрасываются. Сброс показаний часов приводит к неразберихе и утомляет регулярной переустановкой. Особые неудобства это доставляет приочных отключениях, поскольку во многих домах такие часы выполняют функции ночника. Для того, чтобы облегчить жизнь и избавиться от подобных неудобств, я предлагаю изготовить разработанный мною автоматический корректор часов.*

**К**орректор предназначен для отслеживания пропадания электроэнергии и восстановления показаний реального времени на любых электронных часах, имеющих органы ручной коррекции.

Данный корректор содержит в себе микроконтроллер с микросхемой энергонезависимых часов (CMOS-часы) и позволяет автоматически устанавливать истинное время на электронных часах, к которым он подключен, после появления напряжения в сети.

Устройство состоит из нескольких недорогих и легкодоступных элементов. Его структурная схема приведена на рис. 1. Данная схема не требует особых пояснений. Она показывает, из каких узлов состоит корректор и как он подключается к электронным часам.

Принципиальная схема устройства приведена на рис. 2. Из нее видно, каким образом соединяются все элементы и какие контакты корректора используются для подключения к часам.

Поскольку корректор не имеет собственного сетевого источника питания, а использует одно из питающих напряжений электронных часов от 5 до 40 В, в

нем установлен стабилизатор напряжения выполненный на микросхеме DA1. Конденсаторы СР1, С1—С3 обеспечивают дополнительную фильтрацию напряжения питания. Однокристальные часы корректора выполнены на микросхеме DD2. Кварцевый резонатор ZQ1 обеспечивает работу генератора часов на частоте 32768 Гц. Конденсаторы С6 и СТ1 предназначены для коррекции частоты генератора часов. Питание микросхемы происходит либо от стабилизатора напряжения DA1 через диод VD2, либо от элемента питания GB1 напряжением 3,6 В через диод VD3. Диоды обеспечивают защиту от разряда элемента питания через остальные элементы при отсутствии сетевого питания. Через диод VD4 и токоограничивающий резистор R2 производится подзарядка элемента питания GB1 до рабочего напряжения 3,6 В от стабилизатора DA1. Избыток зарядного напряжения гасится на диодах VD2 и VD4. Резисторы R3 и R4 подтягивают интерфейсные линии шины I<sup>2</sup>C к источнику питания. По этой последовательной двухпроводнойшине с помощью линии синхронизации SCL и линии данных SDA происходит запись и чтение времени микроконтроллером DD1 из однокристальных часов DD2. Микроконтроллер работает под управлением программы, зашиваемой в него внутрисхемно (без выпаивания) через разъем J1. Кнопка SB1 предназначена для установки времени в однокристальных часах DD2. При пропадании питания

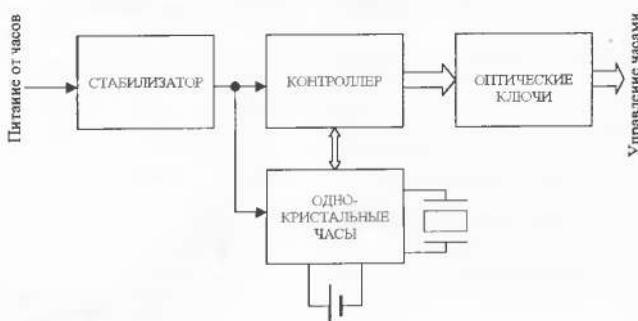


Рис. 1