

Лабораторная работа №5

Универсальный асинхронный приемопередатчик UART

Теоретическая часть

UART (Universal Asynchronous Receiver/Transmitter) - универсальный асинхронный приемопередатчик, интерфейс для связи цифровых устройств, предназначенный для передачи данных в последовательной форме. Очень распространён и весьма востребован, имеет аппаратную реализацию во многих микроконтроллерах. Например, микроконтроллеры STM32 из семейства STM32F100xx, в зависимости от варианта исполнения, содержат 2 или 3 USART.

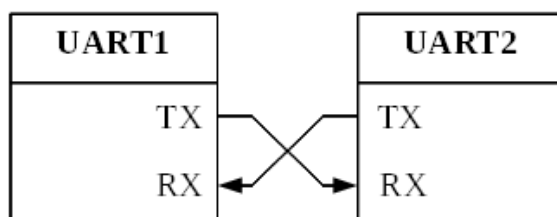
USART (Universal Synchronous-Asynchronous Receiver/Transmitter) - универсальный синхронно-асинхронный приемопередатчик - аналогичный UART интерфейс, но дополнительно к возможностям UART, поддерживает режим синхронной передачи данных - с использованием дополнительной линии тактового сигнала. Впрочем, синхронная передача используется гораздо реже асинхронной.

UART может использоваться как для взаимодействия компонентов внутри одного устройства, так и для подключения устройств между собой. Для внешних подключений сигналы с уровнями логики ТТЛ или КМОП подходят мало из-за низкой помехоустойчивости. Распространённым стандартом физического уровня для UART, который подходит для подключения внешних устройств является RS-232. Этому стандарту, в частности, соответствует последовательный порт (COM-порт) компьютера. Так что, микроконтроллер с помощью схемы преобразования уровней может обмениваться информацией с COM-портом компьютера.

Варианты подключения UART

В UART передача данных происходит в последовательной форме, т.е. по одному биту. Поэтому для передачи в одном направлении требуется один проводник; для полнодуплексной двунаправленной связи потребуются два проводника.

Выход обозначают TD или TX (transmitted data), вход - RD или RX (received data). Для подключения двух устройств выход одного подключают ко входу другого и вход первого - к выходу второго.



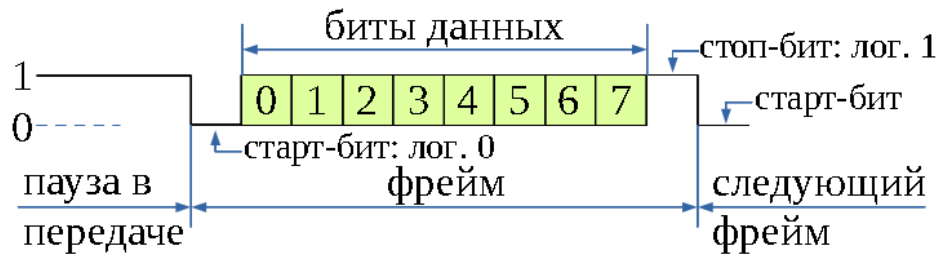
Как видим, объединять устройства с помощью UART очень просто. Для двунаправленного подключения требуются только три проводника (с учётом общего провода), а для однонаправленного или двунаправленного полудуплексного - всего два.

Формат передачи данных UART

В отсутствие передачи на выходе UART присутствует уровень лог. 1.

Данные передаются в виде посылок (фреймов), каждая из которых состоит из стартового бита, битов данных и одного или нескольких стоп-битов. Длительность всех битов одинакова, связана со скоростью передачи соотношением $T=1/S$. Существует ряд стандартных скоростей передачи: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 бод. Если внутри одного устройства связь можно осуществлять на произвольной скорости, то для связи с внешними устройствами следует

придерживаться стандартных величин.

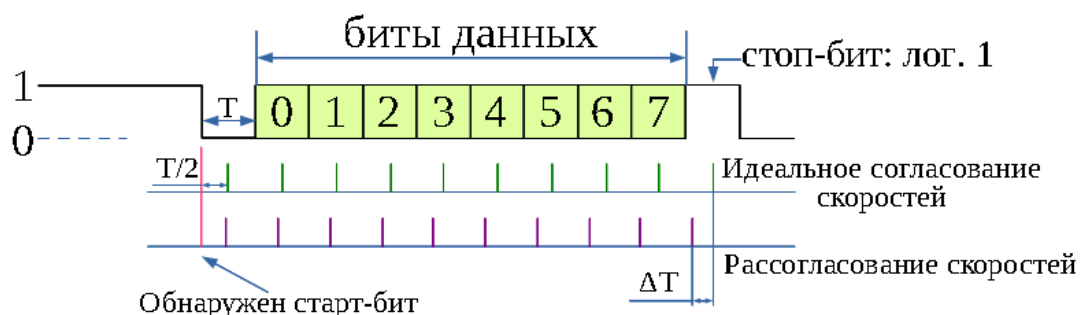


Посылка начинается со стартового бита, он всегда имеет значение лог. 0. После стартового бита передаются биты данных. Количество битов данных может составлять 5-9 в зависимости от настроек UART. Обычно передаётся 8 бит данных или 9 бит (8 бит собственно данных и один бит чётности). Завершается посылка стоп-битами, их значение - всегда лог. 1, количество обычно составляет 1, 1.5 или 2. Под количеством стоп-битов понимается длительность соответствующего им единичного импульса по отношению к длительности битов данных и старт-бита. Этим объясняется возможность выражать количество битов дробным числом. Сразу же после стоп-битов может начинаться передача следующей посылки или может быть пауза произвольной длительности, во время которой на выходе также формируется уровень лог. 1.

Так как во время передачи стоп-бита и пока линия свободна, на выходе присутствует единичное значение, а старт-бит имеет значение лог. 0, старт-бит позволяет выявить момент начала передачи данных, разделить две последовательные посылки и осуществить синхронизацию передатчика и приёмника.

Если передатчик и приёмник работают на одной скорости, настроены на работу с одинаковым количеством битов данных, стоп битов, одинаково сконфигурированы в отношении бита чётности, то для обмена данными не требуется передавать отдельно тактовый сигнал - он может быть восстановлен приёмником самостоятельно.

Обнаружив начало старт-бита, приёмник ждёт в течение половины длительности передачи бита, после чего начинает считывать сигнал на входе с частотой, равной скорости передачи данных. В идеальном случае момент каждого считывания приходится на середину принимаемого бита. В реальности генераторы тактовых импульсов передатчика и приёмника имеют рассогласование по частоте, в результате каждое новое считывание всё больше смещается относительно середины очередного бита. Важно, чтобы за время передачи одной посылки, смещение не превысило половины длительности бита, а с учётом переходных процессов - смещению лучше не превышать четвертой части длительности бита. Иначе вместо считывания бита произойдёт считывание соседнего бита (или считывание линии во время переходного процесса) и посылка будет принята неверно.



USART в STM32 (STM32F100xx)

USART в микроконтроллерах STM32 предоставляет гибкие средства для полнодуплексного обмена данными с внешними устройствами в последовательном формате с возможностью поддержки сигналов CTS/RTS; поддерживает полудуплексный обмен по однопроводной линии; может работать в широком диапазоне скоростей передачи. В мультибуферном режиме DMA достигается высокая скорость передачи данных, максимальное значение составляет 3 Мбит/с. Также поддерживается однонаправленная передача в

синхронном режиме; мультипроцессорная связь; LIN (local interconnection network) - сеть для локальной связи; smartcard протокол; инфракрасный протокол в соответствии со спецификацией IrDA (infrared data association) SIR ENDEC.

Основные возможности:

- Асинхронная полнодуплексная связь.
- Асинхронная однопроводная полудуплексная связь.
- Настраиваемый метод оверсэмплинга (супердискретизации) даёт возможность выбора между скоростью передачи и допустимым отклонением скорости.
- Передатчик и приёмник используют общую программируемую скорость передачи, которая может настраиваться в широких пределах; максимальное значение достигает 3 Мбит/с при 8-кратном оверсэмплинге.
- Программируемая длина слова (8 или 9 бит).
- Настраиваемое количество стоп-битов (1 или 2).
- В LIN режиме поддерживается отправка и обнаружение приёмником Break-посылки (генерируется 13-битная и детектируется 10/11 битная).
- Имеется выход тактового сигнала для синхронной передачи.
- IrDA SIR кодек для инфракрасной связи (поддерживается длительность бита 3/16 в нормальном режиме).
- Интерфейс Smartcard поддерживает асинхронный протокол смарт-карт как определено в стандарте ISO 7816-3; используется 0.5, 1.5 стоп-битов в операциях со смарт-картой.
- Конфигурируемая мультибуферная связь с использованием DMA (direct memory access).
- Флаги, устанавливаемые при обнаружении событий во время обмена данными (приёмный буфер заполнен; буфер для передачи пуст; передача завершена).
- Контроль чётности (можно настроить передатчик на формирование бита чётности и приёмник на контроль бита чётности).
- Четыре флага, устанавливаемые при обнаружении ошибок (ошибка переполнения; обнаружен шум в принимаемом сигнале; ошибка фрейма; ошибка чётности).
- 10 источников прерывания USART, связанных с флагами регистра состояния SR (изменение состояния CTS; обнаружение посылки LIN Break; регистр данных передатчика пуст; передача завершена; регистр данных приёмника заполнен; обнаружение события "линия свободна" (Idle line); ошибка переполнения; ошибка фрейма; обнаружение шума; ошибка чётности).
- Мультипроцессорная связь (переход в тихий режим, если не произошло сопоставление адреса).
- Пробуждение из тихого режима при обнаружении свободной линии (Idle line) или при обнаружении адресной метки.
- Два режима пробуждения приёмника - по адресному биту (9-й, старший бит) или при обнаружении, что линия свободна.

Как и с другой периферией, Mbed предоставляет удобный и простой в использовании интерфейс для работы с USART.

USART в mbed

Для работы с USART в mbed используется класс Serial. Пример программы с использованием этого класса приведен ниже:

```
1 #include "mbed.h"
2
3 Serial pc(SERIAL_TX, SERIAL_RX);
4
5 DigitalOut myled(LED1);
6
7 int main()
8 {
9     while(1) {
10         wait(1);
11         pc.printf("Hello world!\n", i++);
12         myled = !myled;
13     }
14 }
15
```

Эта программа инициализирует UART на пинах SERIAL_TX, SERIAL_RX и в бесконечном цикле

отправляет по ним сообщение «Hello world!».

На отладочных платах, используемых в лабораторной работе, эти пины подключены к так называемому виртуальному COM-порту. Он создается сразу же при подключении платы, и для операционной системы Windows виден как настоящий COM-порт. Это позволяет обмениваться данными с платой используя стандартные средства, такие как HyperTerminal, SerialLogger и другие, а так же без особого труда написать свою программу, которая сможет получать данные с платы и управлять подключенными к ней устройствами.

Методы и свойства класса Serial

Конструкторы:

Serial (PinName tx, PinName rx, const char *name=NULL, int baud=9600)

Создает объект последовательного порта на указанных пинах, со значениями скорости по умолчанию.

Serial (PinName tx, PinName rx, int baud)

Создает объект последовательного порта на указанных пинах и заданной скорости.

Методы:

void baud (int baudrate)

Устанавливает скорость передачи данных.

void format (int bits=8, Parity parity=SerialBase::None, int stop_bits=1)

Устанавливает формат передачи данных

int readable ()

Позволяет определить, есть ли данные, которые можно вычитать из порта

int writeable ()

Позволяет определить можно ли записать данные в порт

void attach (Callback< void()> func, IrqType type=RxIrq)

Подключает функцию-обработчик, которая будет вызываться каждый раз, когда приходят данные.

void gets(uint8_t * buffer, int count)

Вычитывает count байт из порта

void puts(uint8_t * buffer, int count)

Записывает count байт в порт

uint8_t getc()

Вычитывает 1 байт

void putc(uint8_t byte)

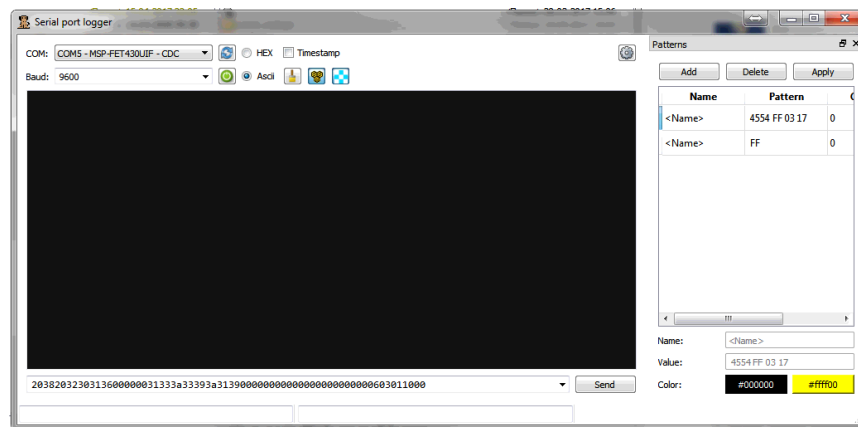
Записывает один байт

Следует отметить, что методы gets и getc – блокирующие. Это означает, что после их вызова программа остановится пока не получит указанное количество данных.

Подключение программы SerialLogger

SerialLogger – это программа предназначенная для подключения к устройствам через последовательный порт. Она позволяет анализировать и обрабатывать входящие данные, а так же передавать данные на устройство.

На рисунке ниже изображено основное окно программы:



Для того, чтобы подключиться устройству необходимо:

1. В поле COM выбрать номер порта соответствующего вашему устройству (как правило это тот, что не COM1)
2. В поле Baud указать желаемую скорость обмена
3. Выбрать режим обмена – бинарный HEX или текстовый Ascii
4. Нажать на зеленую кнопку «Открыть»

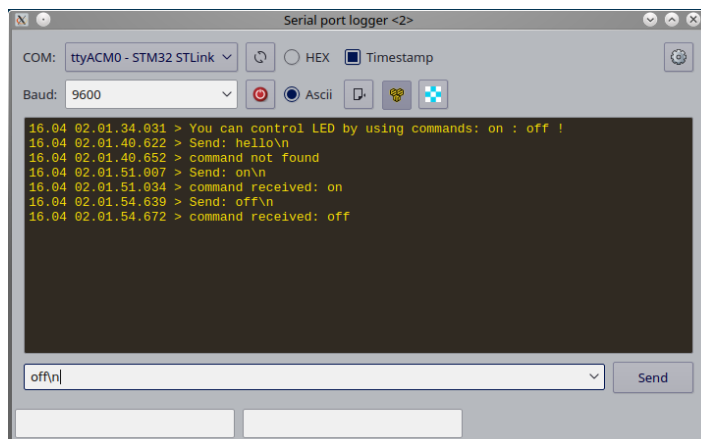
После подключения к устройству в окне программы будут отображаться все входящие данные. Для отправки данных на устройство нужно ввести их в нижнее поле ввода и нажать кнопку Send.

Пример

Напишем программу, которая будет работать следующим образом:

1. При старте отправляет сообщение о запуске
2. После этого переходит в ожидание команды
3. По команде «on» будет включать светодиод
4. По команде «off» будет выключать светодиод

Текст программы приведен ниже. Алгоритм ее работы довольно простой. Мы символ за символом принимаем от пользователя текст команды, и каждый раз сравниваем введенный текст с искомым. Как только введенная команда совпадает с одной из поддерживаемых мы ее выполняем, и переходим в режим приема следующей команды. Сравниваются команды при помощи стандартной функции `strcmp`, описание которой можно посмотреть в интернете.



```

1 #include "mbed.h"
2 #include "string.h"
3
4 //Создаем объект Serial
5 Serial pc(SERIAL_TX, SERIAL_RX);
6
7 DigitalOut myled(LED1);
8
9 //Два массива, которые хранят поддерживаемые команды
10 const char on[] = "on";
11 const char off[] = "off";
12
13 int main()
14 {
15     //Приветственное сообщение
16     pc.printf("You can control LED by using commands: on : off !\n");
17
18     //Массив для хранения данных принятых от пользователя
19     char command[5]={0};
20     //счетчик байт принятых от пользователя
21     int i = 0;
22
23     while(1) {
24         //Вычитываем очередной символ из порта
25         char c = pc.getc();
26
27         //Если это символ переноса строки, значит команда принята полностью
28         if(c == '\n') {
29
30             //если длина команды 2 символа
31             //и эти два символа соответствуют команде on
32             if(i == 2 && strncmp(command, on, i) == 0) {
33                 //значит пришла команда включить светодиод
34                 pc.printf("command received: on\n");
35                 myled = 1;
36             }
37             //если длина команды 3 символа
38             //и эти три символа соответствуют команде off
39             else if(i == 3 && strncmp(command, off, i) == 0) {
40                 //значит это команда выключить светодиод
41                 pc.printf("command received: off\n");
42                 myled = 0;
43             }
44             else {
45                 //иначе - это неправильно введенная команда
46                 pc.printf("command not found\n");
47             }
48             //сбрасываем счетчик введенных символов, чтобы принять
49             //следующую команду
50             i = 0;
51         }
52         //если это любой другой символ
53         else {
54             //то сохраняем его в массиве
55             command[i] = c;
56             i++;
57             //если мы приняли больше символов чем можем сохранить
58             //то считаем что команда была неправильной
59             if(i > sizeof(command)){
60                 i = 0;
61             }
62         }
63     }
64 }

```

Задания

Вариант 1 и Вариант 2

Совместная работа двух бригад. Два устройства объединяются через последовательный порт на ножках Serial1_TX, Serial_RX. Так же необходимо объединить земли устройств. Скорость обмена 19200.

Устройство А управляет Устройством Б, отправляя определенные команды:

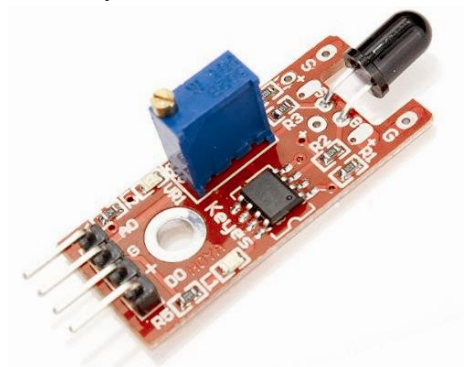
- Однократное нажатие на кнопку USER_BUTTON отправляет команду «переключить состояние светодиода»
- Двукратное нажатие на кнопку USER_BUTTON (как двойной клик мыши) отправляет команду «Запустить мигание светодиодом»

Если на Устройстве Б удерживается нажатой кнопка USER_BUTTON, то оно отказывается выполнять команду, если кнопка не нажата – то выполняет команду. После приема, Устройство Б отправляет статус выполнения команды. Устройство А приняв статус отображает его на своем светодиоде:

- Если команда выполнена, то Устройство А мигает светодиодом 1 раз
- Если команда не выполнена, то Устройство Б мигает светодиодом 3 раза

Вариант 3

Используется датчик пламени.



Модуль реагирует на открытое пламя. Воспринимающим элементом датчика служит фотодиод получающий инфракрасное излучение.

Может использоваться в газовых, соляровых, бензиновых горелках имеющих электроподжиг. В этом случае дуга создаваемая высоким напряжением поджигает топливо, а автоматика управляющая поджигом получает данные от датчика пламени определяет загорелось ли топливо и прекращает работу электроподжига. В дальнейшем происходит управление подачей топлива и контроль наличия пламени.

- A0 — аналоговый выход, выходное напряжение соответствует освещенности датчика ИК излучением.
- G — общий провод.
- + — питание.
- D0 — выход сигнала имеющего логические уровни.

Фотодиод соединен со входом компаратора выполненного на микросхеме LM393. С помощью подстроечного резистора выполняется настройка порога срабатывания компаратора. Так устанавливается чувствительность датчика огня. При обнаружении пламени яркостью выше установленной при настройке на выходе D0 будет высокий уровень напряжения. Если огня нет или его яркость мала, то на выходе D0 низкий уровень. На аналоговый выход поступает усиленный сигнал фотодиода.

Светодиод L1 показывает включение питания. L2 сообщает о срабатывании датчика и формировании на выходе D0 высокого уровня. Светодиод L2 облегчает настройку модуля KY-02.

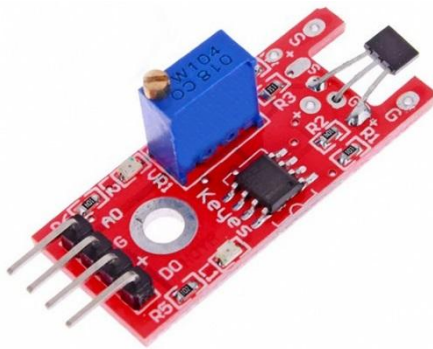
Для выполнения задания необходимо подключить устройство к ПК через виртуальный последовательный порт на скорости 14400. Нужно поддерживать команды «Поставить на охрану», «Снять с охраны». В состоянии «Охрана» прибор отправляет на ПК сообщение «Fire!» при обнаружении пожара. В состоянии «Снят с охраны» не отправляет ничего.

Текущее состояние отображается на светодиоде

- Снят с охраны — светодиод выключен
- Поставлен на охрану — светодиод включен
- Обнаружен огонь — светодиод мигает

Вариант 4.

Используется модуль на основе датчика Холла.



Модуль определяет присутствие поля постоянного магнита или магнитного поля катушки проволоки подключенной к постоянному току. Наиболее часто применяется для определения скорости вращения различных деталей механизмов.

- A0 Напряжение сигнала этого выхода соответствует напряженности магнитного поля.
- G Общий.
- + Питание положительный полюс.
- D0 Выход цифрового сигнала сообщающего о превышении установленного порога.

Воспринимающий элемент модуля – микросхема датчик Холла [SS49E](#). Она соединена со входом компаратора на микросхеме [LM393YD](#). С помощью подстроечного резистора выполняется установка порога срабатывания компаратора. При этом устанавливается чувствительность датчика магнитного поля. При воздействии поля напряженностью более чем установлена при настройке на выходе D0 меняется уровень напряжения. На аналоговый выход поступает усиленный сигнал воспринимающего элемента.

Светодиод L1 показывает включение питания. L2 светится постоянно и гаснет при срабатывании датчика на магнитное поле установленной напряженности. При настройке порога чувствительности помогает светодиод L2, можно обойтись без вольтметра для напряжения выхода.

Для выполнения задания необходимо подключить устройство к ПК через виртуальный последовательный порт. Нужно поддерживать команды «Поставить на охрану» и «Снять с охраны».

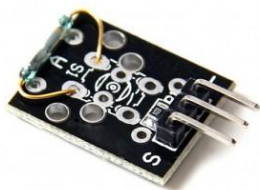
- Команда «Поставить на охрану» выполняется только если к датчику поднесен магнит.
- Если магнита нет, то в ответ на команду «Поставить на охрану» выдается сообщение «Ошибка взятия».
- Если во взятом состоянии магнит удаляется от устройства, на ПК передается сообщение «Тревога»

Текущее состояние отображается на светодиоде

- Снят с охраны – светодиод выключен
- Поставлен на охрану – светодиод включен
- Обнаружен взлом – светодиод мигает

Вариант 5.

Используется модуль на основе геркона.



Геркон – датчик магнитного поля. Внутри зеленой стеклянной колбы смонтированной на плате модуля КУ-021 находятся подвижные контакты из особого сплава. При воздействии магнитного поля контакты геркона замыкаются.

Модуль позволяет закрепить датчик на неподвижном элементе конструкции. На подвижной детали устанавливают постоянный магнит. Для этого удобно использовать неодимовые магниты. При сближении геркона и магнита замыкаются контакты геркона. Модуль на основе геркона используется в случаях, когда определяется приближение магнита вплотную. Например, датчик плотного закрытия двери. Применяется в линейно перемещающихся деталях механизмов для определения перемещения в крайнее положение, например принтер, простой станок.

- - Общий.
- Средний контакт Питание положительный полюс.
- S Выход цифрового сигнала сообщающего о замыкании геркона

Для выполнения задания необходимо подключить устройство к ПК через виртуальный последовательный порт. Скорость обмена данными 19200. Нужно поддерживать команды «Поставить на охрану» и «Снять с охраны».

- Команда «Поставить на охрану» выполняется только если к датчику поднесен магнит.
- Если магнита нет, то в ответ на команду «Поставить на охрану» выдается сообщение «Ошибка взятия».
- Если во взятом состоянии магнит удаляется от устройства, на ПК передается сообщение «Тревога»

Текущее состояние отображается на светодиоде

- Снят с охраны – светодиод выключен
- Поставлен на охрану – светодиод включен
- Обнаружен взлом – светодиод мигает

Вариант 6

Используется датчик на основе микрофона



Датчик срабатывает при звуках громкостью выше установленного порога. Модуль датчика звука применяется в приборах содержащих аналоговую или цифровую электронику, в том числе и электронику на базе микроконтроллеров. Он может служить датчиком присутствия в охранных системах. Реагирует на шаги, щелчки замка и другие шумы. Также модуль КУ-037 применяется в автоматике управления освещением. Реагирует на звук шагов, шум двигателя автомобиля управляя при этом работой светильников.

Модуль КУ-037 содержит микрофон. Определение превышения порога громкости звука происходит с помощью микросхемы компаратора [LM393YD](#). Настройка порога срабатывания выполняется потенциометром находящимся на плате модуля. Если порог громкости звука превышен, то на выходе D0 появляется высокий уровень напряжения. Уровень напряжения на аналоговом выходе A0 соответствует уровню громкости окружающих шумов. Если применяется микроконтроллер, то возможна программная обработка колебаний уровня напряжения на выходе A0 и сравнение с имеющейся базой в памяти МК колебаний аналогового сигнала, что позволяет распознавать некоторые события происходящие рядом с датчиком.

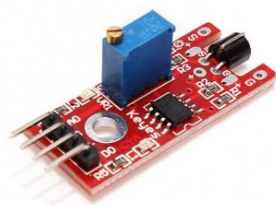
- A0 Напряжение соответствует уровню громкости окружающих шумов.
- G Общий провод.
- + Питание положительный полюс.
- D0 Выход логического сигнала сообщаящего о превышении установленного порога.

Для выполнения задания необходимо подключить устройство к ПК через виртуальный последовательный порт. Скорость обмена данными 9600.

Необходимо поддерживать команды «Разрешить управление хлопком», «Запретить управление хлопком». Если управление разрешено, то по одинарному хлопку необходимо выключить светодиод, а по двойному хлопку – включить светодиод. Каждое изменение состояния светодиода должно выводиться на ПК.

Вариант 7

Используется модуль на основе датчика касания



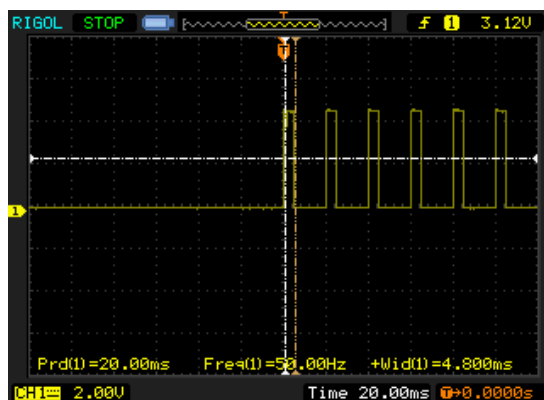
Устройство КУ-036 позволяет оснастить прибор находящийся в жилых или промышленных помещениях функцией сенсорной кнопки. В качестве сенсорной поверхности используется металлическая деталь напоминающая обычную кнопку, корпус электроприбора или другие металлические элементы конструкции.

Металлическая сенсорная поверхность обладает большим запасом прочности, что важно в аппаратуре с антивандальными требованиями. В отличие от механических кнопок ресурс сенсорных не ограничен. Модуль датчика касания также применяется в оборудовании используемом при опасных видах работ для снижения риска несчастного случая.

На торце платы модуля КУ-036, на компоненте расположен контакт соединяемый с сенсорной металлической поверхностью. Прикосновение к поверхности способствует приему электромагнитных волн частотой 50 Гц излучаемых электропроводкой помещения. Тело человека при этом служит антенной, а входная часть модуля радиоприемником. Сигнал частотой 50 Гц усиливается микросхемой LM393YD и подается на выходы устройства. Модуль датчика касания содержит переменный резистор, с помощью которого устанавливается чувствительность датчика КУ-036. На плате есть светодиод индикации включения питания и светодиод мигающий при касании.

- A0 – инвертированный сигнал 50 Гц.
- G – общий провод.
- + питание положительный полюс.
- D0 – сигнал с размахом логических уровней частотой 50 Гц.

На выходе D0 при касании формируется сигнал из импульсов следующих с частотой 50 Гц.



Для выполнения задания необходимо подключить устройство к ПК через виртуальный последовательный порт. Скорость обмена данными 9600.

На ПК необходимо передавать следующую информацию:

- Сообщение о поднесении пальца к датчику
- Сообщение об отведении пальца от датчика
- Сообщение о времени в течении которого палец касался датчика

Вариант 8

Используется модуль на основе датчика температуры.



Датчик применяется для контроля температуры воздуха в помещении. Используется в следующих случаях: регулятор температуры, автоматика систем отопления, автоматизация систем вентиляции. Определяет превышение температурой установленного порога и одновременно позволяет грубо оценивать величину температуры. Пороговое значение настраивается точно, но для точного измерения температуры или для сборки электронного термометра используют другой датчик [KY-001](#) с цифровым выходом. Благодаря выходу на котором формируется сигнал с уровнем логической единицы или нуля, что говорит о превышении или нет температурного порога, модуль датчика температуры удобно использовать в схемах на дискретных элементах без использования микроконтроллера. Например, выход модуля KY-028 соединяют с затвором полевого транзистора достаточной мощности для включения вентилятора работающего от 12 В. По такой схеме можно собрать автомат охлаждения электронного прибора.

Воспринимающий элемент датчика – терморезистор. Он соединен со входом микросхемы компаратора LM393YD. С помощью подстроечного резистора выполняется настройка порога срабатывания компаратора. Так устанавливается температурный порог. При превышении температурой установленного порога на выходе D0 будет высокий уровень напряжения. Если температура мала, то на выходе D0 низкий уровень.

Индикатор L1 сообщает о подаче питания. Светодиод L2 включается при превышении температурой окружающего воздуха установленного порога. Обращая внимание на L2 удобно проводить настройку модуля KY-028.

- A0 Напряжение сигнала соответствует температуре.
- G Общий провод
- + Питание
- D0 Выход логического сигнала сообщающего о превышении установленного порога.

Для выполнения задания необходимо подключить устройство к ПК через виртуальный последовательный порт. Скорость обмена данными 19200. Нужно поддерживать команды «Поставить на охрану» и «Снять с охраны».

- Команда «Поставить на охрану» выполняется только если не превышен порог температуры.
- Если порог превышен, то в ответ на команду «Поставить на охрану» выдается сообщение «Ошибка взятия».
- Если во взятом состоянии происходит превышение порога температуры, на ПК передается сообщение «Пожар»

Текущее состояние отображается на светодиоде

- Снят с охраны – светодиод выключен
- Поставлен на охрану – светодиод включен
- Обнаружен пожар – светодиод мигает