AVR и шины данных

Controller Area Network (CAN)

История о USB и с чем его «едят»

МпУА Колкер А.Б. Ф-т автоматики
НГТУ a.kolker@corp.nstu.ru

Распределенные системы

Современные технические системы строятся по «распределенной» схеме, где интеллектуальные датчики и исполнительные устройства объединяются в общую шину. Наиболее распространенные шины:

- RS485 (ModBus Profibus)
- I2C
- CAN
- Ethernet

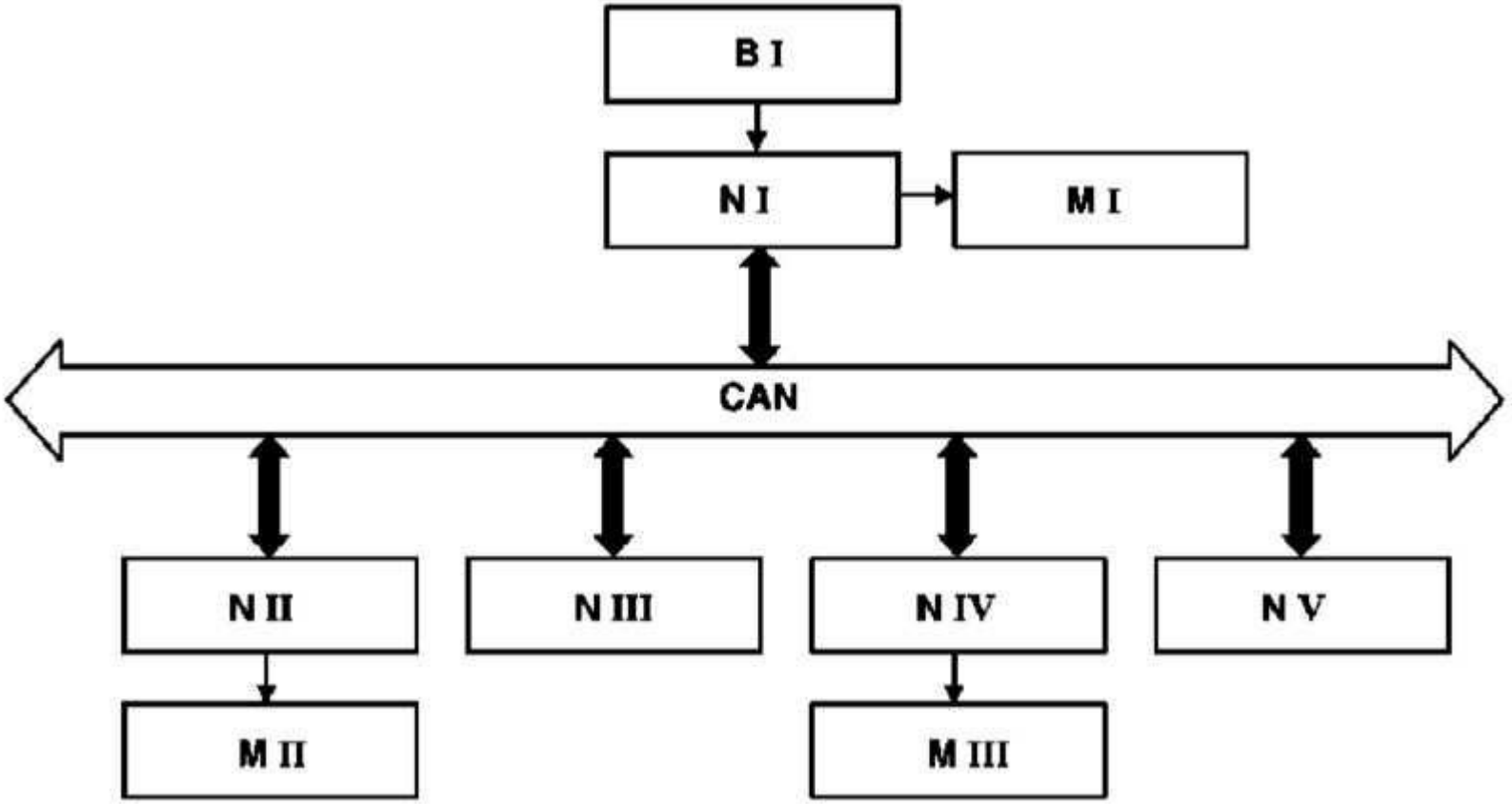
Controller Area Network

Во всех высокотехнологичных системах современного автомобиля применяется CAN-протокол для связи ЭБУ с дополнительными устройствам и контроллерами исполнительных механизмов и различных систем безопасности.

- Шина является полнодуплексной (или просто дуплексной), т.е. любое подключенное к ней устройство может одновременно принимать и передавать сообщения.

- Сигнал с чувствительного элемента соответствующего информационного (датчика) поступает в ближайший блок управления, который обрабатывает его и передает на шину обмена данными CAN.

- Любой блок управления, подключенный к шине данных CAN, может считывать этот сигнал, вычислять на его основе параметры управляющего воздействия и управлять исполнительным сервомеханизмом.



B — датчик

N — контроллер

M — исполнительные механизмы (сервоприводы)

Почему CAN?

Уменьшение количества кабелей. Провода от датчиков тянутся только к ближайшему блоку управления, который преобразует измеренные значения в пакет данных и передает его в шину CAN;

-
- Управлять исполнительным механизмом может любой блок управления, который по шине CAN получает соответствующий пакет данных, и на его основе рассчитывает значение управляющего воздействия на сервомеханизм;
- Улучшение электромагнитной совместимости;
- Уменьшение количества штекерных соединений и уменьшение количества контактных выводов на блоках управления;
- Снижение веса;
- Уменьшение количества датчиков, т.к. сигналы одного датчика (например, с датчика температуры охлаждающей жидкости) могут быть использованы различными системами;

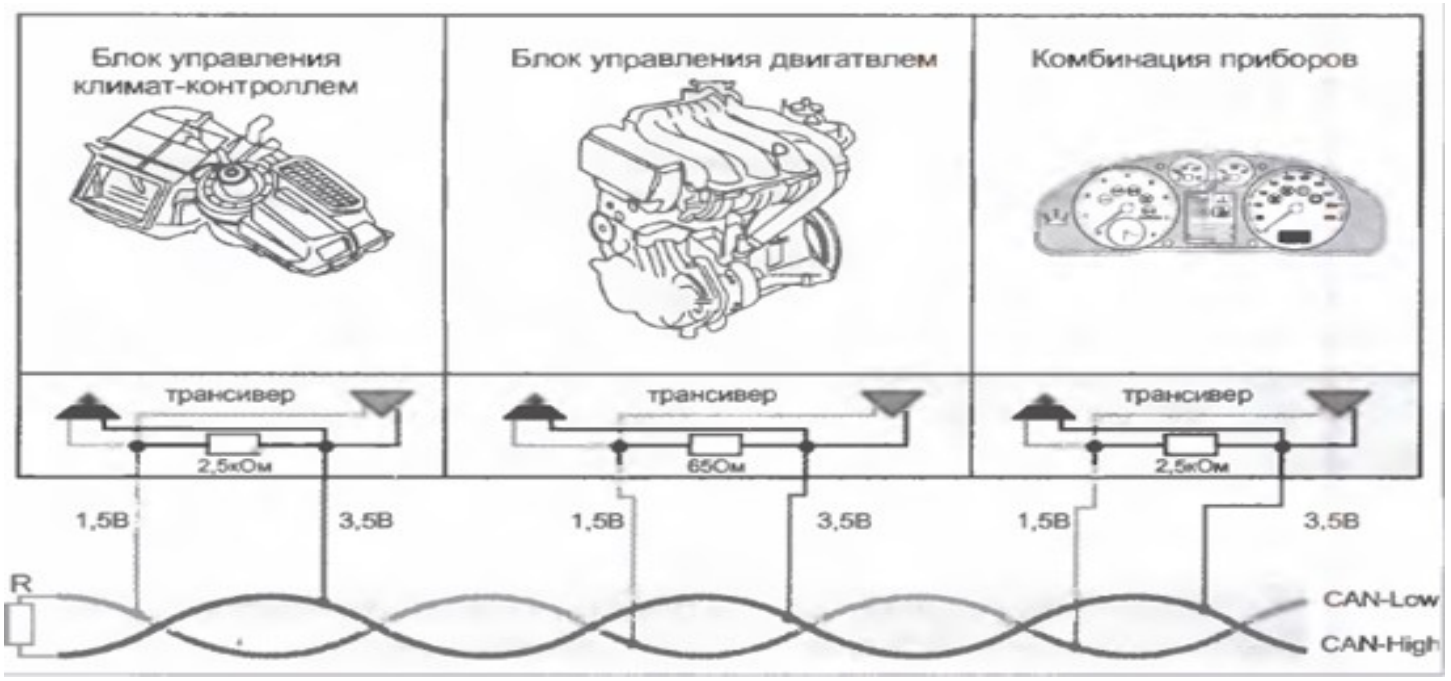
Почему CAN?

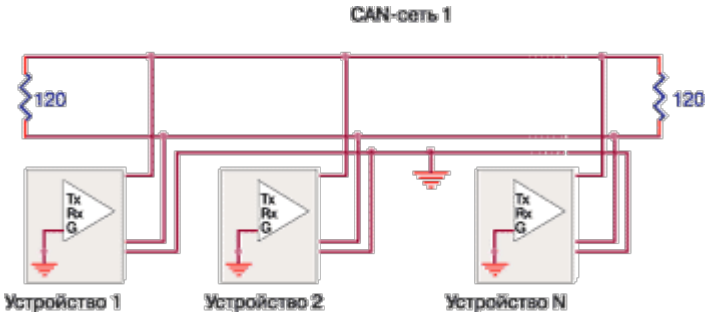
- Улучшение возможностей диагностирования,

  Пусть сигналы одного датчика (например, сигнал скорости) используются различными системами. Если сообщение о неисправности выдают все использующие данный сигнал системы, то неисправным является, как правило, датчик или блок управления, обрабатывающий его сигналы. Если же сообщение о неисправности поступает только от одной системы, хотя данный сигнал используется и другими системами, то причина неисправности, чаще всего, заключена в обрабатывающем блоке управления или сервомеханизме;
- Высокая скорость передачи данных – возможна до 1 Мбит/с при максимальной длине линии 40 м.
- Несколько сообщений могут поочередно передаваться по одной и той же линии.

Почему CAN?

В отличие от RS485, в CAN нет явного Ведущий-ведомый подчинения, и существует механизм арбитража.



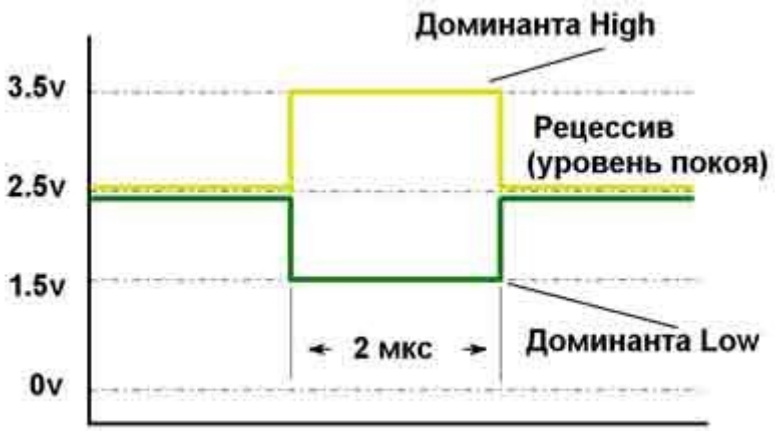


МпУА Колкер А.Б. Ф-т автоматики
НГТУ a.kolker@corp.nstu.ru

CAN уровни

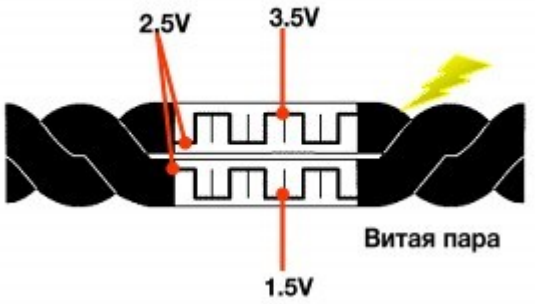
2+1 проводников:

- CAN-High
- CAN-Low
- Gnd



Две линии- инвертированы. Возможен аварийный режим однопроходной схемы.

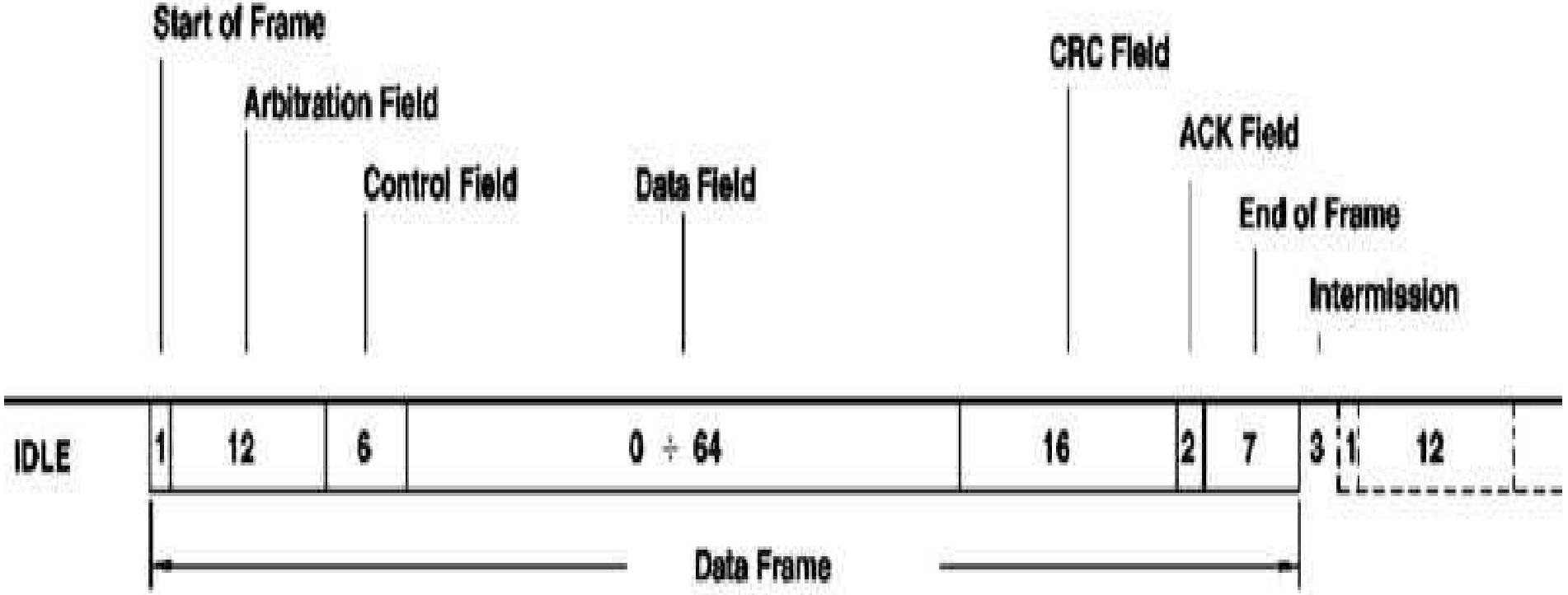
- Скорость до 1 Мбит до 40м.
- Длина сообщения 8 байт



CAN формат пакета

- **Data Frame** (кадр сообщения) для передачи сообщений по шине данных CAN (например: температура охлаждающей жидкости);
- **Remote Frame** (кадр запроса) для запроса сообщений по шине данных CAN от другого блока управления;
- **Error Frame** (кадр ошибки), все подключенные блоки управления уведомляются о том, что возникла ошибка и последнее сообщение по шине данных CAN является недействительным.

CAN



МпУА Колкер А.Б. Ф-т автоматики
НГТУ a.kolker@corp.nstu.ru

Назначение полей

Start of Frame (стартовый бит): Маркирует начало сообщения и синхронизирует все модули;

- **Arbitration Field** (идентификатор и запрос): Это поле состоит из идентификатора (адреса) в 11 бит и 1 контрольного бита (Remote Transmission Request-Bit). Этот контрольный бит маркирует кадр как Data Frame (кадр данных) или как Remote Frame (кадр удаленного запроса) без байтов данных;

- **Control Field** (управляющие биты): Поле управления (6 бит) содержит IDE-бит (Identifier Extension Bit) для распознавания стандартного и расширенного формата, резервный бит для последующих расширений и – в последних 4 битах – оличество байтов данных, заложенных в Data Field (поле данных);

- **Data Field** (данные): Поле данных может содержать от 0 до 8 байт данных; сообщение по шине данных CAN длиной 0 байт используется для синхронизации распределенных процессов;

CAN

- **CRC Field** (контрольное поле): Поле CRC (Cyclic-Redundancy-Check Field) содержит 16 бит и служит для контрольного распознавания ошибок при передаче;

- **ACK Field** (подтверждение приема): Поле ACK (Acknowledgement Field) содержит сигнал подтверждения приема всех блоков-приемников, получивших сообщение по шине CAN без ошибок;

- **End of Frame** (конец кадра): Маркирует конец пакета данных;

  ----------> далее состояние покоя

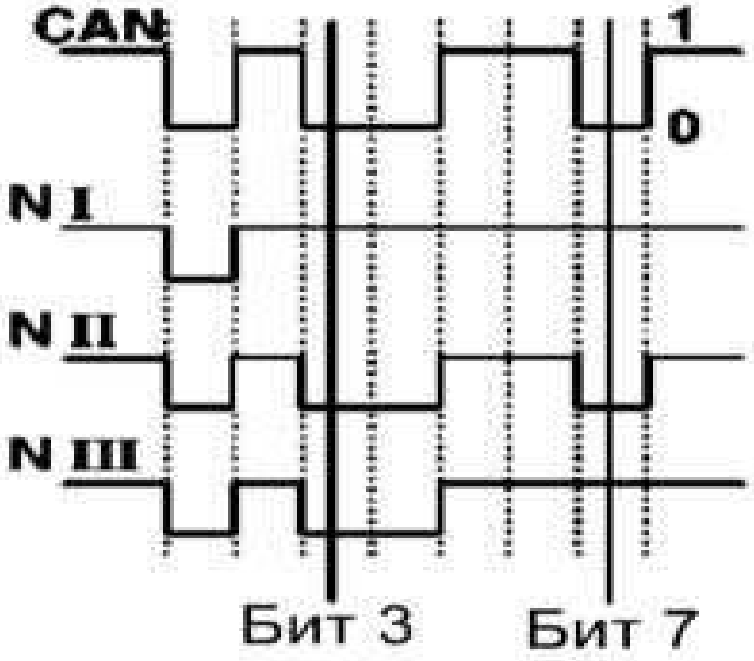
- **Intermission** (интервал): Интервал между двумя пакетами данных. Интервал должен составлять не менее 3 битов. После этого любой блок управления может передавать следующий пакет данных;

- **IDLE** (режим покоя): Если ни один блок управления не передает сообщений, то шина CAN остается в режиме покоя до передачи следующего пакета данных.

Приоритеты

- Передача данных в режиме реального времени требует механизма неотложного предоставления среды к общей шине для устройств.
- Приоритет, с которым сообщение передается по шине CAN, определяется идентификатором (адресом) соответствующего сообщения.
- Идентификатор, соответствующий меньшему двоичному числу, имеет более высокий приоритет, и наоборот.
  - НО арбитраж возможен только на этапе начала передачи, прервать передачу кадра уже нельзя.

Арбитраж-пример



Доминантный бит — 0
Рецессивный бит — 1

Доминантный бит «перезаписывает» все рецессивные

- Первый блок управления (N I) утрачивает арбитраж с 3-го бита.
- Третий блок управления (N III) утрачивает арбитраж с 7-го бита.
- Второй блок управления (N II) сохраняет право доступа к шине данных CAN и может передавать свое сообщение

Правила повторной передачи

- Если передаваемый первым блоком-передатчиком рецессивный бит перезаписывается доминантным битом другого блока-передатчика, то первый блок-передатчик теряет свое право передачи (арбитраж) и становится блоком-приемником (обратитесь к иллюстрации выше).

- Другие блоки управления попытаются передать свои сообщения по шине данных CAN только после того, как она снова освободится. При этом право передачи опять будет предоставляться в соответствии с приоритетностью сообщения по шине данных CAN.

Контроль коллизий

- Пакеты данных могут передаваться только в том случае, если шина обмена CAN свободна (т.е., если после последнего пакета данных последовал интервал в 3 бита, и никакой из блоков управления не начинает передавать сообщение). При этом логический уровень шины данных должен быть рецессивным (логическая «1»).
- Если несколько блоков управления одновременно начинают передавать сообщения, то вступает в силу принцип приоритетности, согласно которому сообщение, обладающее наивысшим приоритетом, будет передаваться первым без потери времени или битов (арбитраж запросов доступа к общей шине данных).
- Каждый блок управления, утрачивающий право арбитража, автоматически переключается на прием и повторяет попытку отправить свое сообщение, как только шина данных вновь освободится.

Контроль ошибок (кто-то сказал невпопад...)

- Механизмы на уровне Data Frame

- **Cyclic-Redundancy-Check**

  Контроль четности там, где это возможно

- **Frame Check**

  Механизм проверяет структуру передаваемого блока (кадра), то есть перепроверяются битовые поля с заданным фиксированным форматом и длина кадра.

Механизмы на уровне битов

- **Говорю и слушаю:** каждый модуль при передаче сообщения отслеживает логический уровень шины данных CAN и определяет при этом различия между переданным и принятым битом. Благодаря этому обеспечивается надежное распознавание глобальных и возникающих в блоке-передатчике локальных ошибок по битам.

Bit Stuffing

- В каждом кадре данных между полем «Start of Frame» и концом поля «CRC Field» должно быть не более 5 следующих друг за другом битов с одинаковой полярностью.

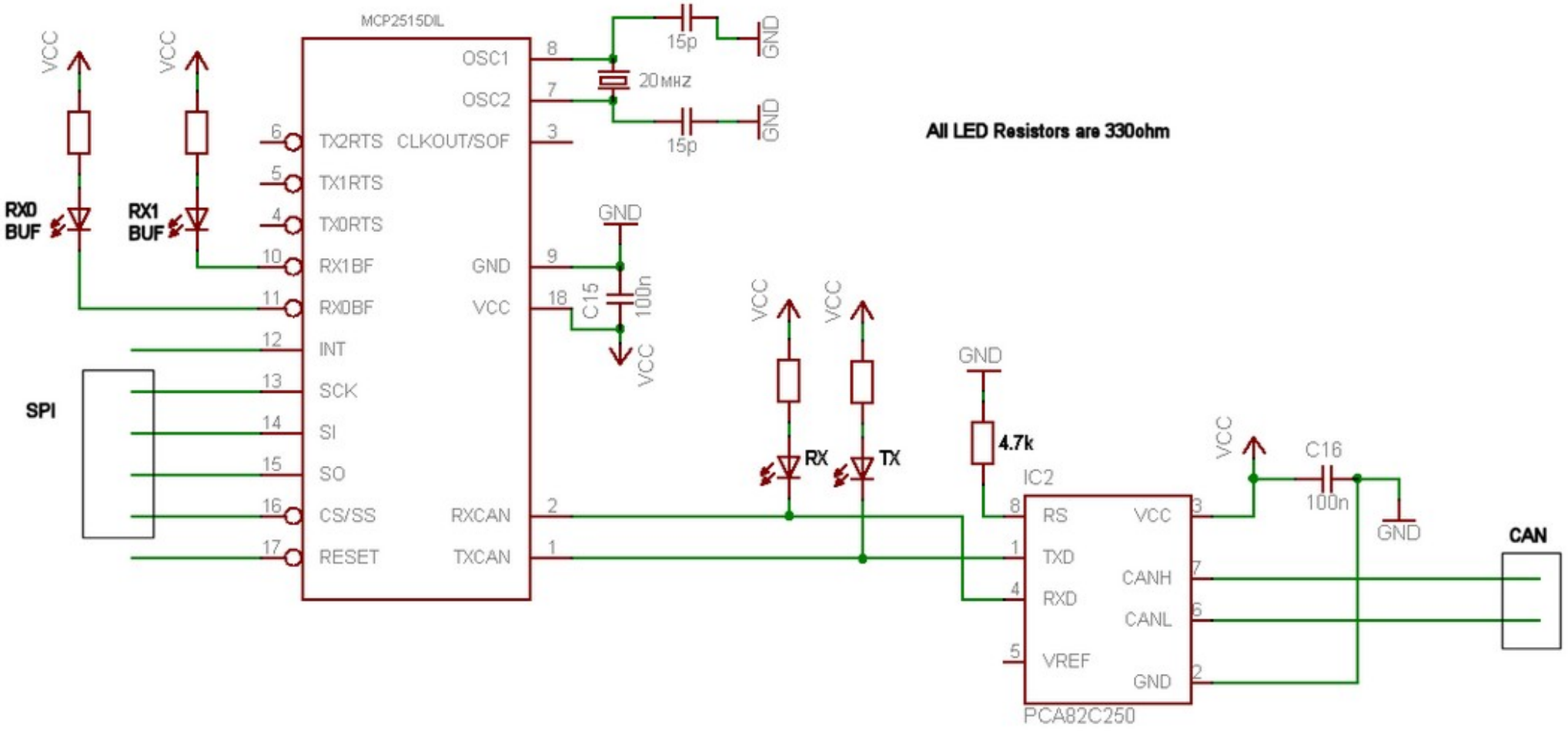
- После каждой последовательности из 5 одинаковых битов блок-передатчик добавляет в поток битов один бит с противоположной полярностью.

- Блоки-приемники удаляют эти биты после приема сообщения по шине данных CAN.

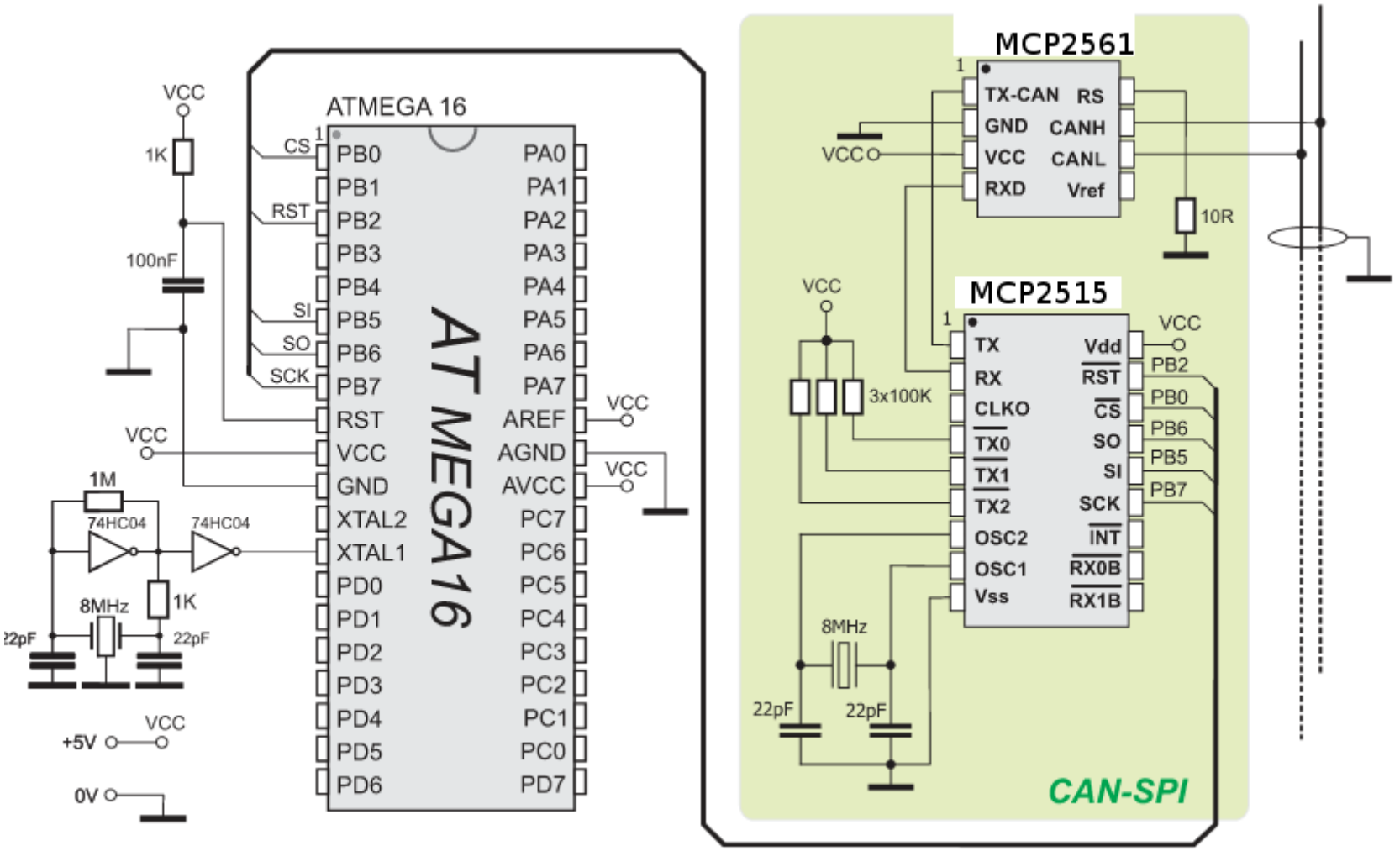
Резюме

- CAN шина требует хоть и несложных, но жестко оговоренных алгоритмических и аппаратных решений.
- Подключение микроконтроллера к CAN целесообразно выполнять при помощи специализированных CAN драйверов, задействуя со стороны МК — SPI (UART,i2c)

CAN<->AVR


MCP2515DIL
OSC1
OSC2
TX2RTS
CLKOUT/SOF
TX1RTS
TX0RTS
RX1BF
GND
RX0BF
VCC
INT
SCK
SI
SO
CS/SS
RXCAN
RESET
TXCAN
15p
20MHZ
15p
GND
GND
C15
100n
VCC
VCC
VCC
RX0 BUF
RX1 BUF
All LED Resistors are 330ohm
SPI
RX
TX
GND
4.7k
IC2
RS
TXD
RXD
VREF
VCC
CANH
CANL
GND
PCA82C250
VCC
C16
100n
GND
CAN

CAN<->AVR

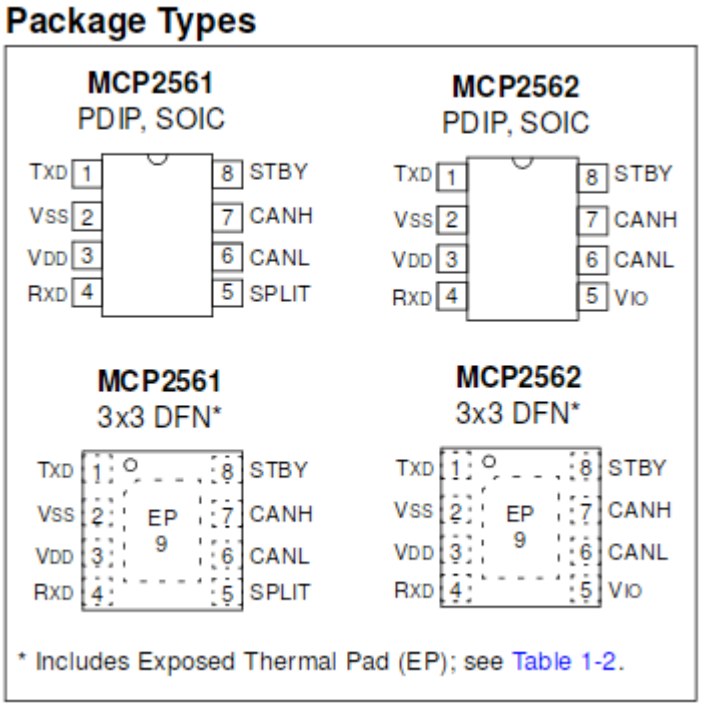


МпУА Колкер А.Б. Ф-т автоматики
НГТУ a.kolker@corp.nstu.ru

MCP2561/2

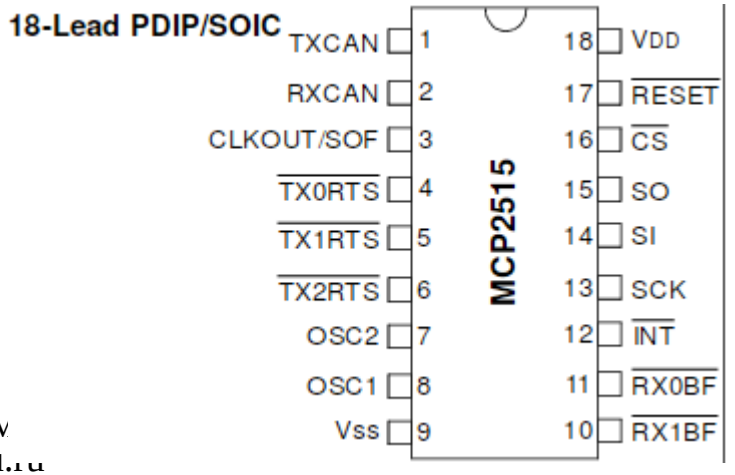
- Трансивер CAN шины, (2 версия). Применяется совместно с контроллером шины CAN. До 1 Мбит.

  http://ww1.microchip.com/downloads/en/DeviceDoc/20005167C.pdf



MCP2515 Stand-Alone CAN Controller with SPI Interface

- CAN2.0B до 1 Мбит/с
- Модуль поддерживает передачу и получение сообщений по шине CAN, получая управление по SPI.
- http://ww1.microchip.com/downloads/en/DeviceDoc/21801G.pdf
- PDIP,TSSOP,QFN



Алгоритим работы с MCP2515:

1. Инициализация, конфигурация
2. Запись данных на отправку в буфер TcnDF
3. Считывание данных из RxnDF

Библиотеки на сайте Микрочип,ATmel или альтернативные:

- http://www.atmel.com/tools/cansoftwarelibrary.aspx
- https://github.com/franksmicro/Arduino/tree/master/libraries/MCP2515

Режим работы MCP2515

Режим	Назначение
Configuration	Режим установки настроек.
Normal	Стандартный режим работы.
Sleep	Режим пониженного энергопотребления. SPI остается активным. Чип проснется от CAN если WAKIE/WAKIF interrupts are enabled
Listen	Чип в состоянии «слушаю»
Loopback	Чип закольцован «сам на себя»

Как фильтровать сообщения?

- MCP2515 разрешает посылать любые сообщения «наружу», но может ограничить их «внутрь»
- Для этого предназначены 6 настраиваемых фильтров с 2 масками

  подробнее можно почитать тут:

  http://modelrail.otenko.com/arduino/arduino-controller-area-network-can

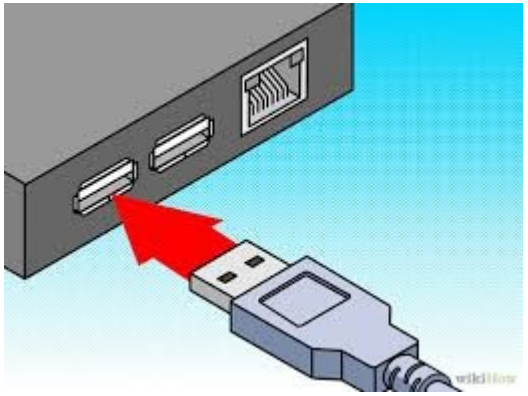
Многие контроллеры уже «в теме»

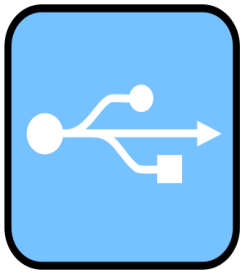
- AT90CAN32-Automotive : 8-bit Automotive AVR Microcontroller, 32KB ISP Flash, CAN Controller
- AT90CAN64-Automotive: 8-bit Automotive AVR Microcontroller, 64KB ISP Flash, CAN Controller
- AT90CAN128-Automotive: 8-bit AVR Microcontroller, 128KB ISP Flash, CAN Controller

Что дает Automotive? -40 +125 C

USB

Большинство современных компьютеров и планшетов благополучно избавилось от всех портов коммуникации, кроме USB и Ethernet





- USB «универсальная последовательная шина») — последовательный интерфейс для подключения периферийных устройств к вычислительной технике. Получил широчайшее распространение и фактически стал основным интерфейсом подключения периферии к цифровой технике.

- Интерфейс позволяет не только обмениваться данными, но и обеспечивать электропитание периферийного устройства. Сетевая архитектура позволяет подключать большое количество периферии даже к устройству с одним разъемом USB.

Как использовать USB в своих устройствах?

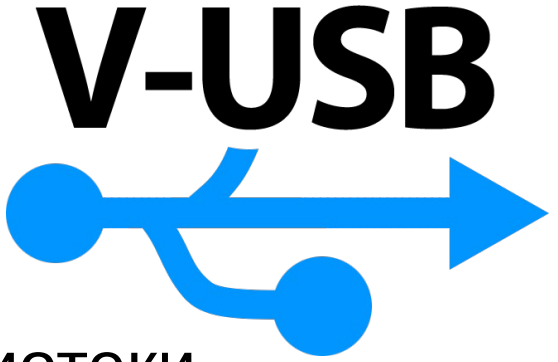
- Можно взять микроконтроллер, который имеет аппаратную поддержку USB интерфейса (например AT90USB*)
- Использовать универсальный конвертер (FTDI232 и др.) USB в «другой» интерфейс. В качестве «другого» может быть RS232, I2C
- Взять обычный микроконтроллер без аппаратной поддержки USB и программно эмулировать интерфейс USB.

Режимы USB 2.0

- Low Speed — 1.5 Mbit/s
- FullSpeed — 12Mbit/s
- HighSpeed — 480Mbit/s

- Программная эмуляция для микроконтроллера, в основном, возможна только USB1.1 или LS USB2.0

V-USB

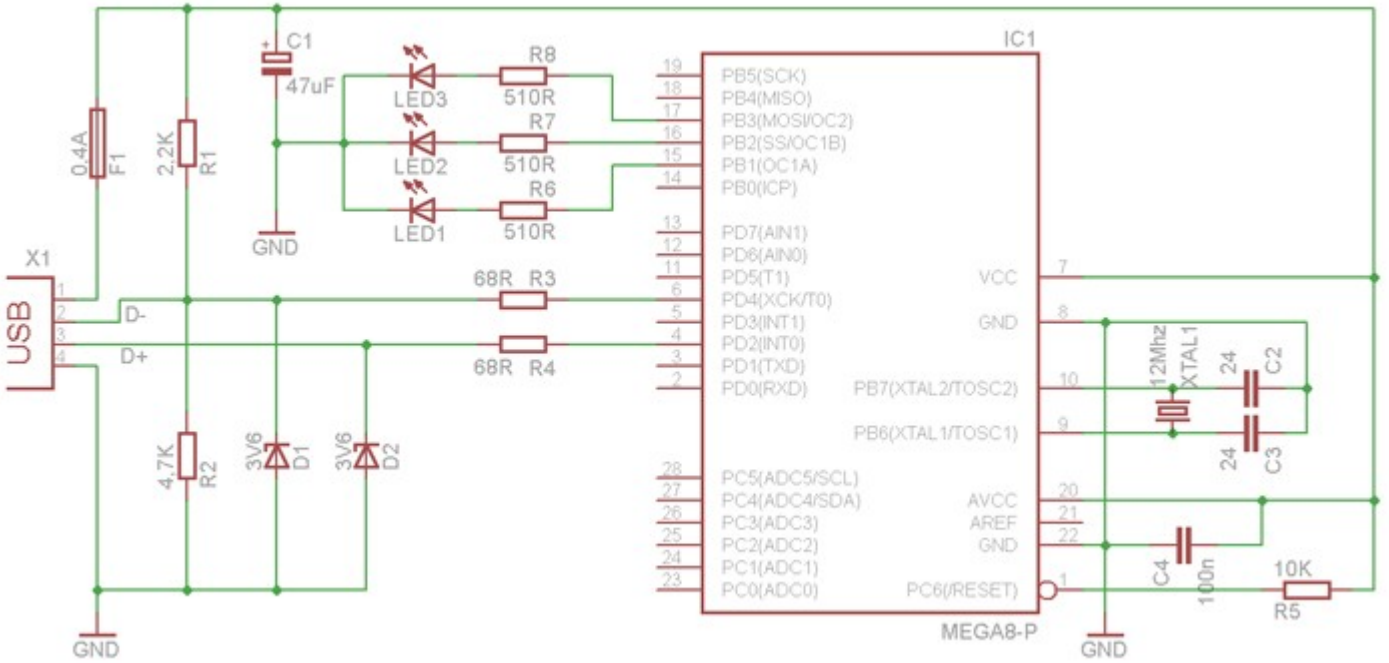


V-USB — название программной библиотеки, позволяющей получить поддержку протокола USB на микроконтроллерах AVR (семейств Classic, Tiny и Mega компании Atmel), которые не имеют аппаратной поддержки USB. Авторство библиотеки принадлежит компании Objective Development, которая распространяет и продвигает V-USB по лицензии GNU GPL и коммерческой лицензии библиотеки свободно доступны).

V-USB поддерживает только USB-HID девайсы, поддерживается только определенный ряд частот работы МК

- https://www.obdev.at/products/vusb/index.html

Пример-цепляем

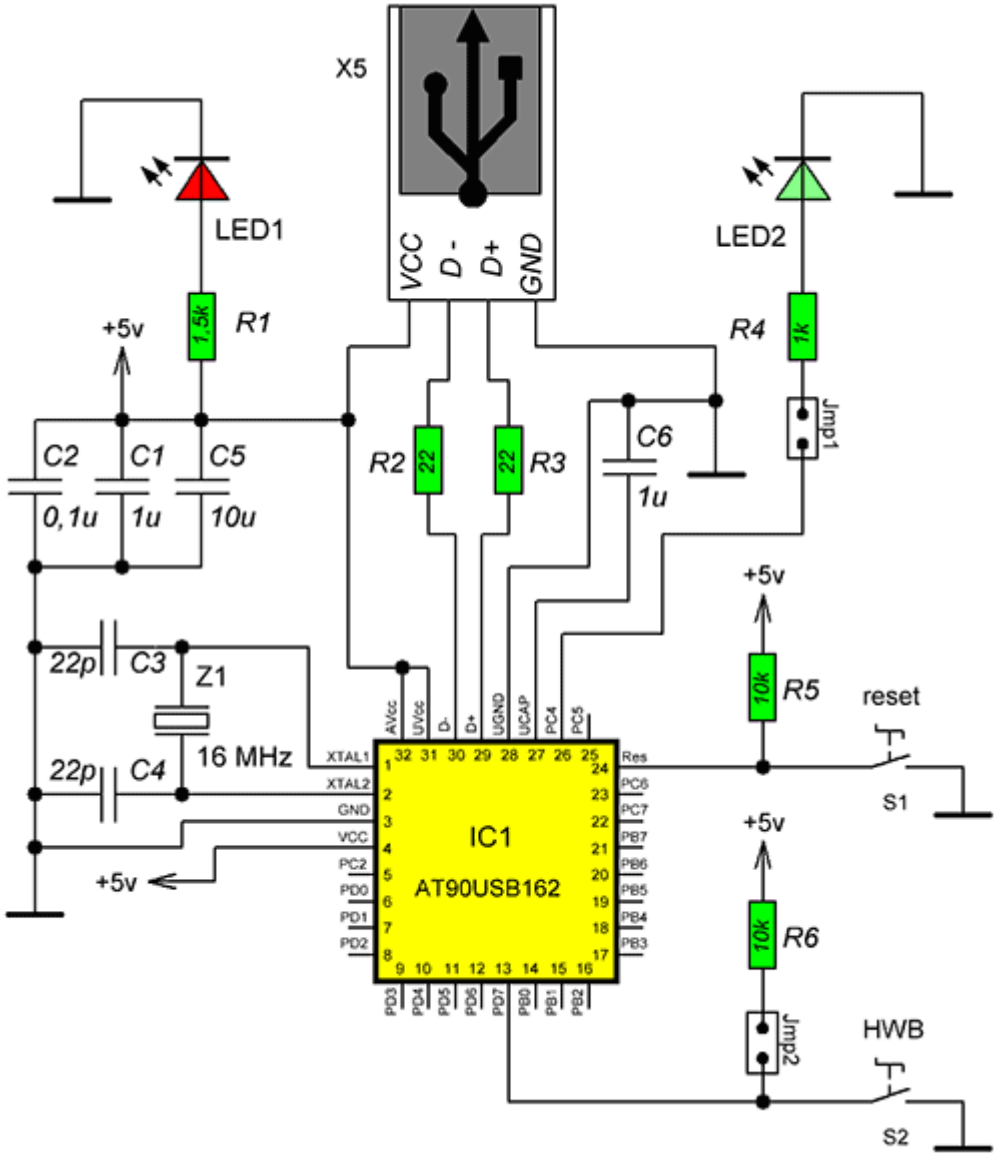


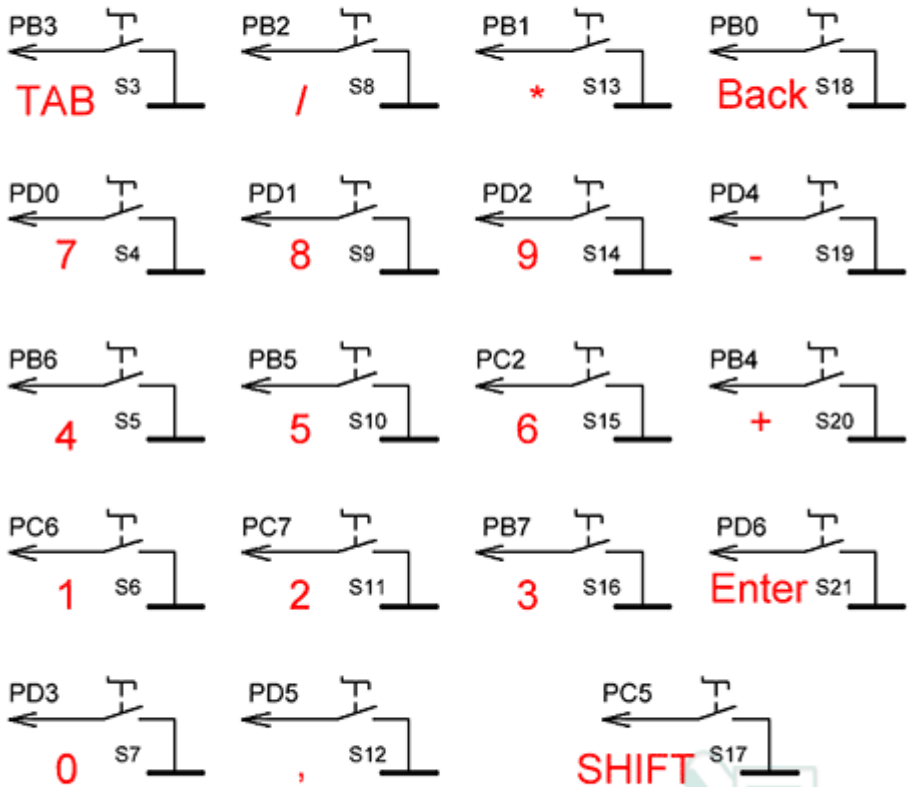
Ассемблерные части V-USB написаны с поддержкой только таких частот: 12 MHz, 12.8 MHz, 15 MHz, 16 MHz, 16.5 MHz, 18 MHz и 20 MHz

Почему HID?

- human interface device — драйвер этих устройств уже есть во всех уважающих себя ОС.
- VUSB предоставляет свободную, легализованную пару VID/PID
- Использование прочих кодов VID и PID требует покупки лицензии на код. Для того, чтобы получить сственный Vendor-ID нужно заплатить usb.org около 2000$

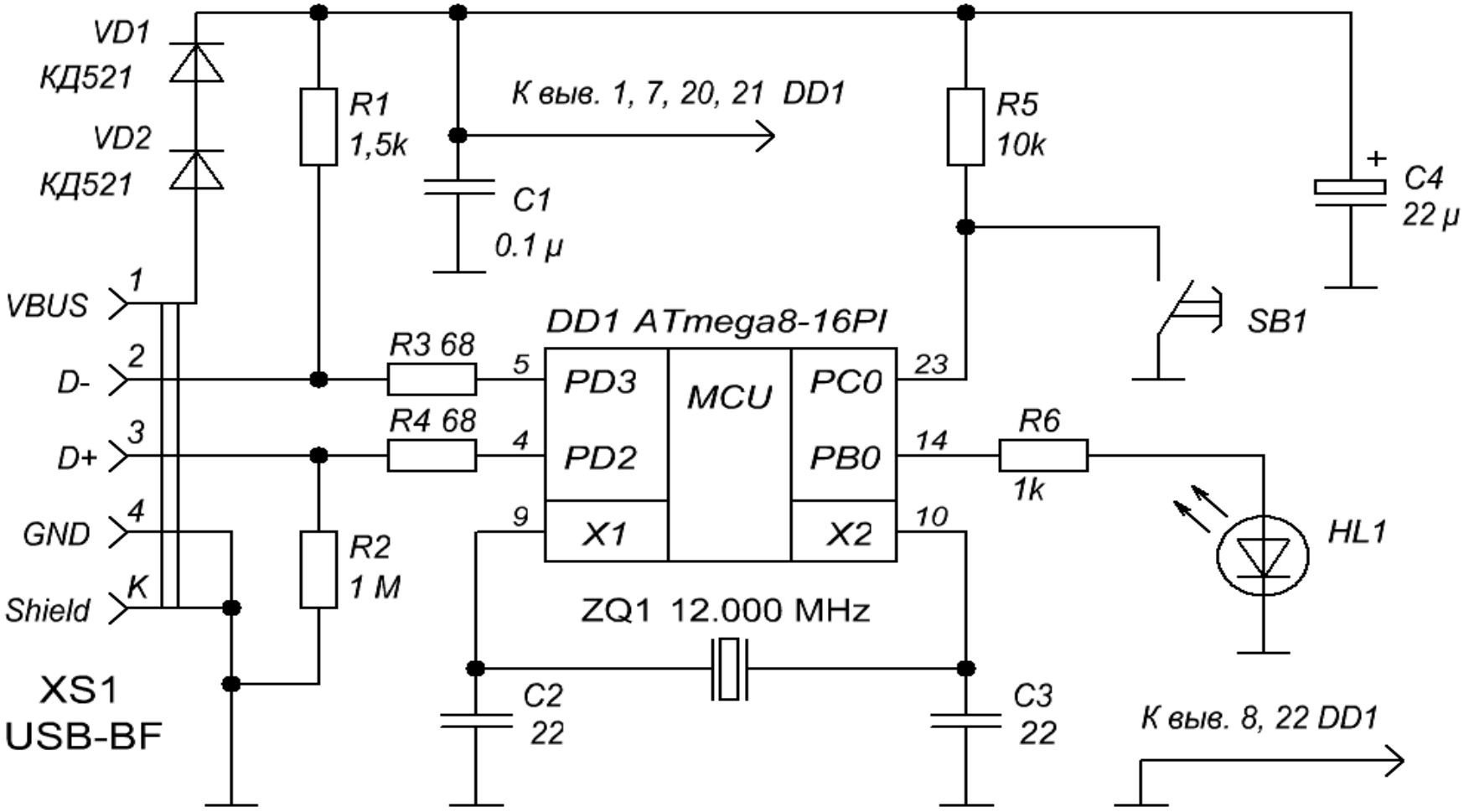
Контроллер с USB





Простое HID

- http://purebasic.mybb.ru/viewtopic.php?id=211



Литература

- http://download.mikroe.com/documents/articles/eng/featured-articles/can/avr/elektor-en-article-easyavr5a-can-spi-c.pdf
- www.usb.org/developers/docs — официальная документация по USB.
- www.beyondlogic.org/usbnutshell/usb1.htm — хороший обзор важных частей USB спецификации.
- www.lvr.com/usb.htm — много хороших ссылок связанных с USB
- https://www.obdev.at/products/vusb/hidkeys.html