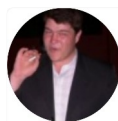


Сообщество Mitsubishi Lancer X

Как я с CAN шиной боролся. (Arduino nano + mcp2515 + lancer10)

2 мая 2024



danlosh

Был 20 часов назад

Я езжу на Mitsubishi Lancer 2.0 cvt 4B11
Москва

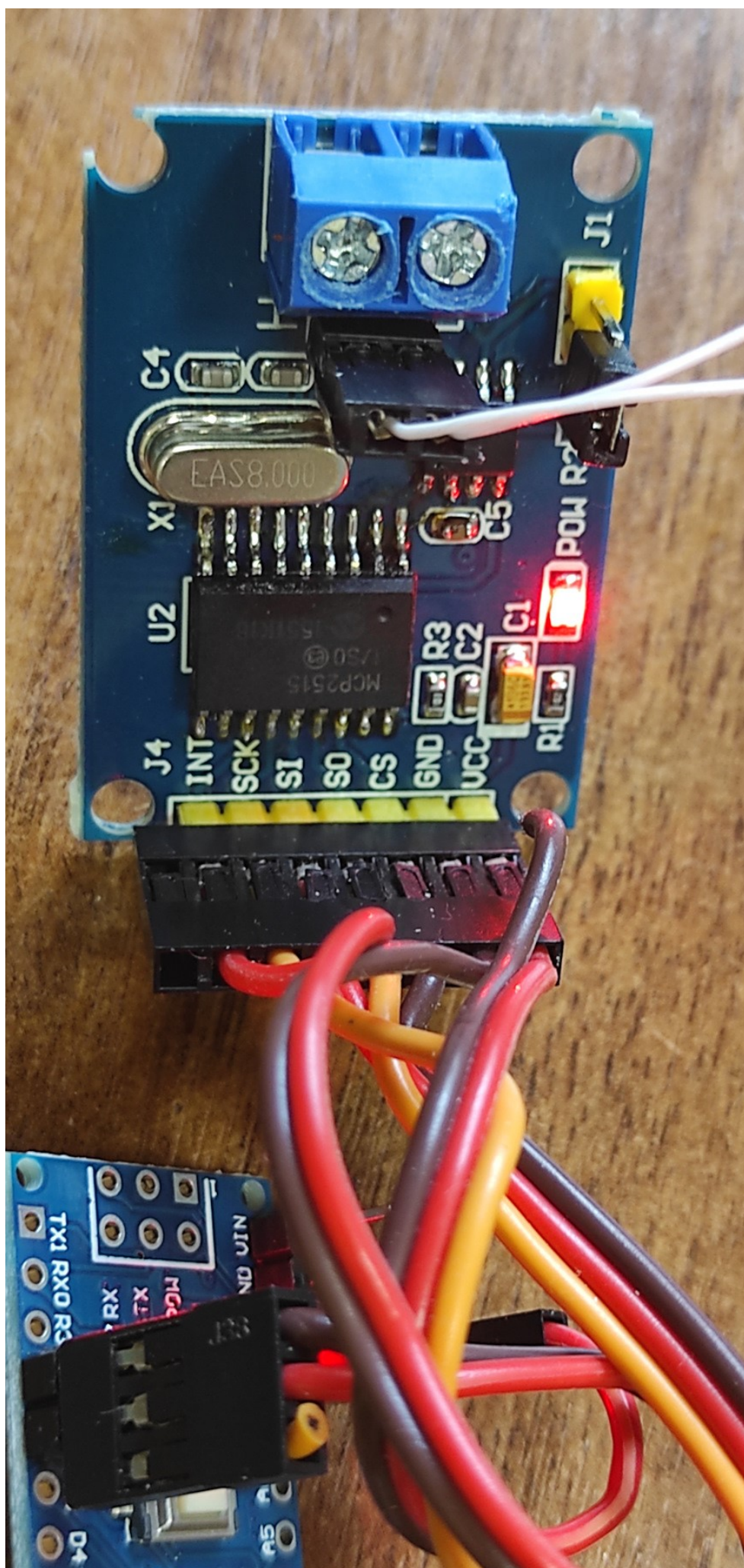
Подписаться

Сообщение

Решил залезть в кан шину автомобиля, купил самые дешёвые модули на али (300-400 рублей суммарно) и решил воспользоваться простой как 5 копеек инструкцией по созданию канхакера, но не тут то было, неделю мозг кипел, а заветные пакетики в программе не бегали.

Уже решил для себя, что вариантов то немного, то ли лыжи не едут, то ли я не совсем умный :-)

С огромным трудом удалось понять чего не хватает в нашей магической связке из ардуино модулей и нашего автомобиля лансерХ.





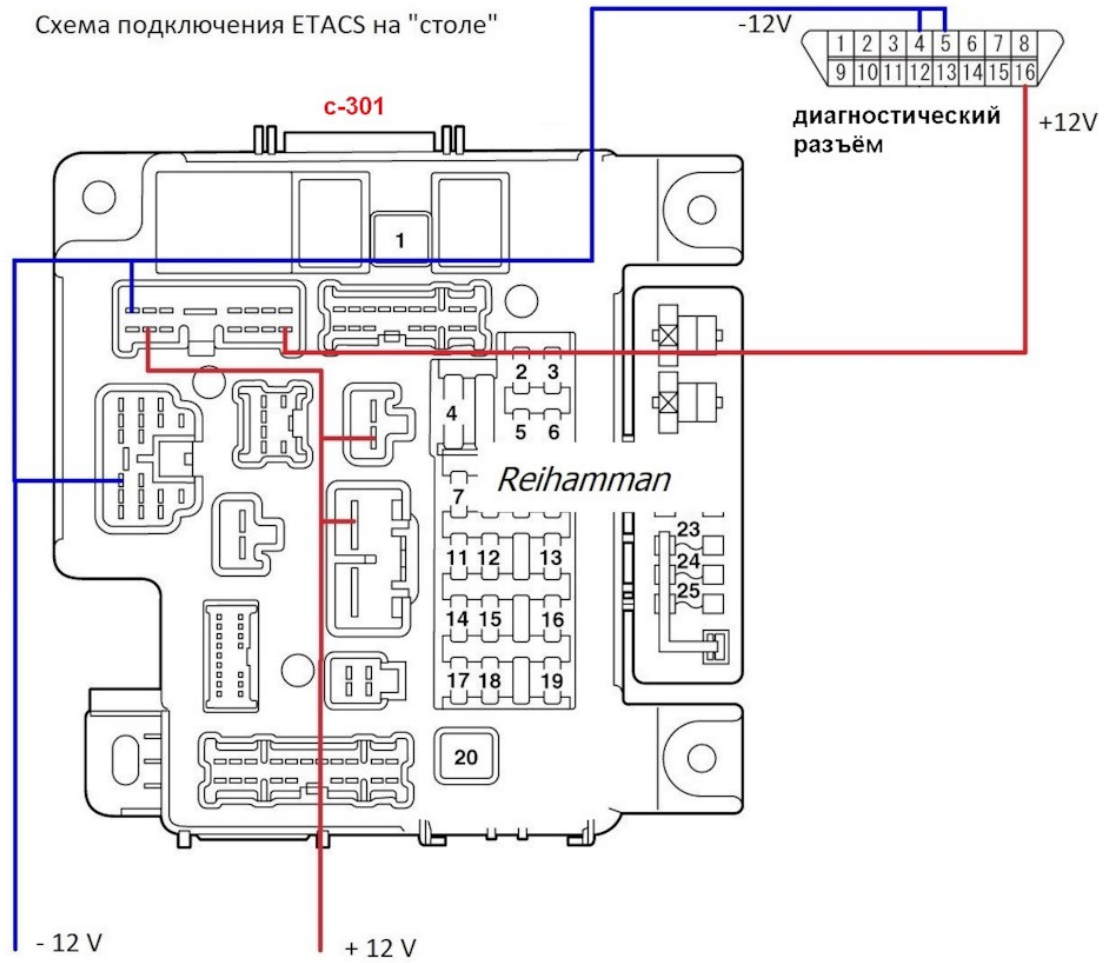
Изначально бегал дня три в машину и пытался подключаться к приборке и к диагностическому разъёму, результат — нулевой!

Вспомнил, что у меня на запчасти валяется раскуроченный блок etax от лансера, половины внутренностей уже нет, но основная часть с интерфейсами сап была на месте, уж лучше на столе экспериментировать, чем бегать до автомобиля.

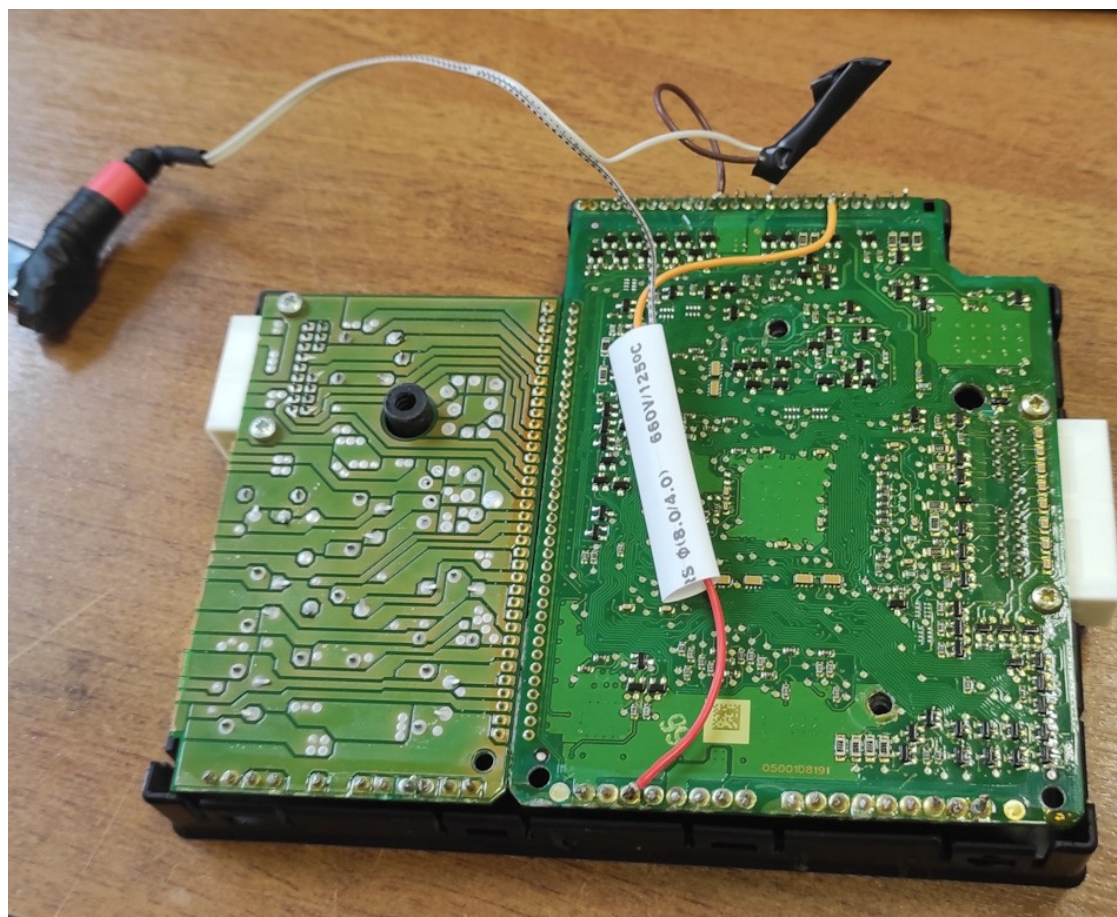
Для питания 12 вольт используем обычный бытовой блок на 12 вольт у меня 1,5 ампера, хватает.



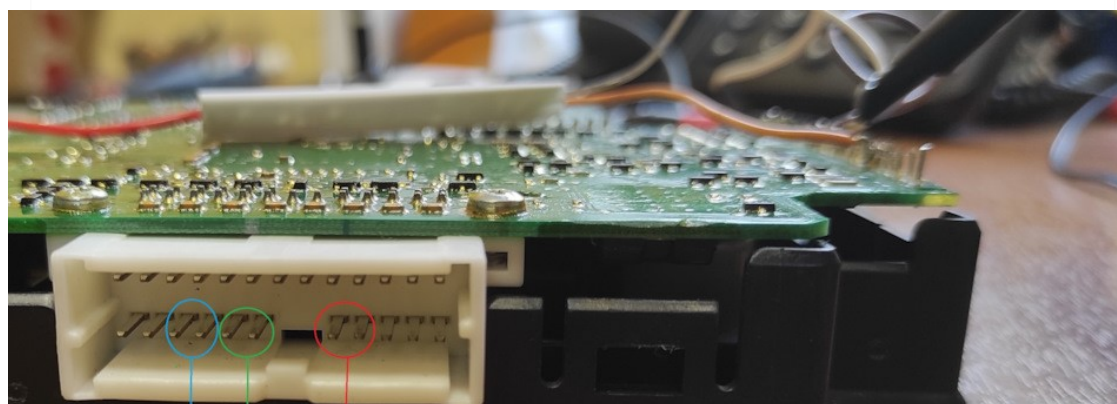
Схему подключения 12 вольт на столе для блока етакс нашёл в интернете быстро:



потом слегка мультиметром побегал по схеме и определил места, куда надо подать 12 вольт, чтобы заработало, у меня это выглядит вот так, плюсы на красный и оранжевый, минусы на белый и коричневый.



После этого гуглим распиновку разъёма c301, у меня он сверху, интересуют нас вот эти пары, это разные выводы на разные места автомобиля.



C-301

4-5 кан шина на диагностический разъём, на столе молчит как рыба об лёд.

6-7 canH-canL - шина 83,3кбит/с к приборке и к магнотолле

8-9 canL-canH - шина 500кбит/с в сторону ECU двигателя.

Шина кан 500кбит/с идёт из etacs в сторону еси двигателя, подключаться туда наверное потенциально проблематично, лучше уж подключиться к шине кан 83.3кбит/с вывод есть в разъёме магнитолы, и он же идёт на климат и на приборку, по сути это одни и те же провода с параллельным подключением в одно и то же место.

В процессе поиска решения наткнулся на разные "уже готовые библиотеки" но как правило ссылка на них уже просрочена и скачать не удавалось по прошествии времени, чтобы уж оставить решение "на века" расскажу как добиться работы модуля самостоятельно исправив пару мест в стандартных библиотеках.

Сперва удалось добиться работы шины 500кбит/с, затем пришлось ещё пару дней гуглить и наткнулся на правильную мысль!

На али продаётся в подавляющем большинстве только версия платы кан с кварцем 8 мгц, но он не поддерживает работу на частоте 83,3. На 500кбит/с можно не заморачиваться с правкой файлов библиотек, будет работать сразу, просто заливаем скетч, который в конце и всё сразу работает.

Для работы с шиной 83,3кбит/с надо сделать вот так!

На гитхабе берём стандартные библиотеки:

[канхакер](#)

[кан модуль mcp2515](#)

скачиваем их как "зип", добавляем их в ардуино ide

Идём в папку документы -> arduino -> libraries -> autowp-mcp2515 открываем на редактирование файл mcp2515.h

добавляем в этот файл строки:

```
#define MCP_8MHz_83k3BPS_CFG1 (0x01)
#define MCP_8MHz_83k3BPS_CFG2 (0xbe)
#define MCP_8MHz_83k3BPS_CFG3 (0x07)
```

у меня это вставлено в районе 28 строки, среди прочих таких же.

Это самая главная магия, в этих строках мы рассказываем кан-плате как именно надо работать на нужной нам частоте, с помощью трёх уникальных важных значений.

Сохраняем файл.

Теперь изменяем файл в той же папке mcp2515.cpp.

в районе строки 268 в switch (canSpeed), на 8мгц, добавляем ещё один case:

```
case (CAN_83K3BPS): // 83.333Kbps !
    cfg1 = MCP_8MHz_83k3BPS_CFG1;
    cfg2 = MCP_8MHz_83k3BPS_CFG2;
    cfg3 = MCP_8MHz_83k3BPS_CFG3;
    break;
```

Здесь мы рассказываем библиотеке, что если канхакер будет просить нужную нам частоту, то в этом месте функцией define будут подставлены наши уникальные важные значения.

Сохраняем, теперь идём в папку с libой канхакера, у меня это по пути документы -> arduino -> libraries -> arduino-canhacker-1.0.1

и открываем для изменения файл CanHacker.cpp

примерно в районе 164 строки находим switch(buffer[1])

и выбираем любой ненужный нам case, например у меня этот:

```
case '1':
    writeDebugStream(F("Set bitrate 20KBPS\n"));
    bitrate = CAN_20KBPS;
    break;
```

В этом кейсе заменяем строку

```
bitrate = CAN_20KBPS;
```

на строку:

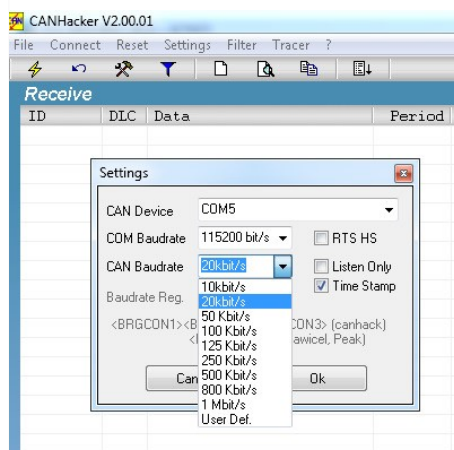
```
bitrate = CAN_83K3BPS;
```

получится общий вид этого кейса:

```
case '1':
    writeDebugStream(F("Set bitrate 20KBPS\n"));
    bitrate = CAN_83K3BPS;
    break;
```

И сохраняем файл.

Это изменение позволит нам в программе канхакер выбирать в списке строку с битрейтом 20кбит/с, а программа при этом будет настраиваться и подключаться к шине со скоростью 83,3кбит/с.



Стандартный скетч из примеров библиотеки канхакера usb_cdc не работает, вернее работает, но выводит одно первое сообщение и всё, на этом вся работа и заканчивается, остальные пакеты игнорируются, незнаю из-за чего так, но нашёл рабочий скетч в другом месте.

В нашу плату заливаем вот такой скетч, он с дебагом, но он работает, его даже не менял совсем.

Рабочий скетч для arduino nano v3:

```
#include <can.h>
#include <mcp2515.h>
#include <canhacker.h>
#include <canhackerlinereader.h>
#include <lib.h>
#include <spi.h>
#include <softwareserial.h>
const int SPI_CS_PIN = 10;
const int INT_PIN = 2;
const int SS_RX_PIN = 3;
const int SS_TX_PIN = 4;
CanHackerLineReader *lineReader = NULL;
CanHacker *canHacker = NULL;
SoftwareSerial softwareSerial(SS_RX_PIN, SS_TX_PIN);
void setup() {
  Serial.begin(115200);
  while (!Serial);
  SPI.begin();
  softwareSerial.begin(115200);
  Stream *interfaceStream = &Serial;
  Stream *debugStream = &softwareSerial;
  canHacker = new CanHacker(interfaceStream, debugStream, SPI_CS_PIN);
  canHacker->setClock(MCP_8MHZ); // For 8MHz crystal oscillator
  //canHacker->enableLoopback(); // uncomment this for loopback
  lineReader = new CanHackerLineReader(canHacker);
  pinMode(INT_PIN, INPUT);
}
void loop() {
  CanHacker::ERROR error;
  if (digitalRead(INT_PIN) == LOW) {
    error = canHacker->processInterrupt();
    handleError(error);
  }
  error = lineReader->process();
  handleError(error);
}
void handleError(const CanHacker::ERROR error) {
  switch (error) {
    case CanHacker::ERROR_OK:
    case CanHacker::ERROR_UNKNOWN_COMMAND:
    case CanHacker::ERROR_NOT_CONNECTED:
    case CanHacker::ERROR_MCP2515_ERRIF:
    case CanHacker::ERROR_INVALID_COMMAND:
      return;
    default:
      break;
  }
  softwareSerial.print("Failure (code ");
  softwareSerial.print((int)error);
  softwareSerial.println(")");
  digitalWrite(SPI_CS_PIN, HIGH);
  pinMode(LED_BUILTIN, OUTPUT);
  while (1) {
    int c = (int)error;
    for (int i=0; i<c; i++) {
      digitalWrite(LED_BUILTIN, HIGH);
      delay(500);
      digitalWrite(LED_BUILTIN, LOW);
      delay(500);
    }
    delay(2000);
  }
};
}
```

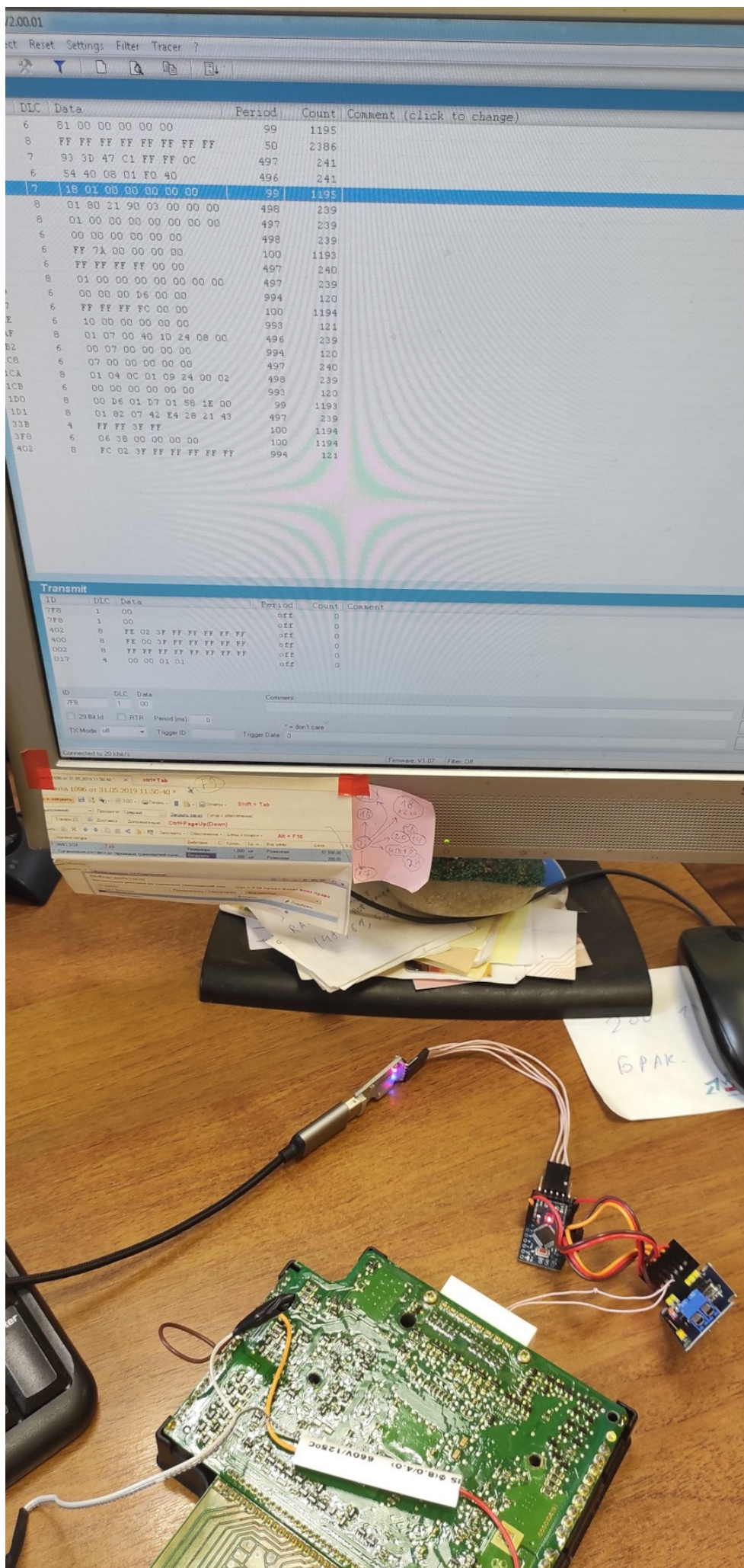
CANHacker V2.00.01

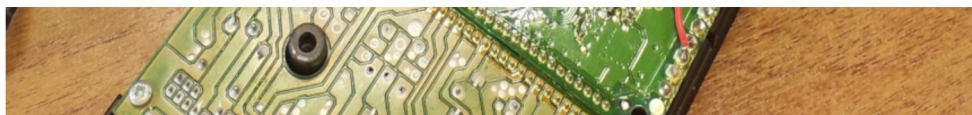
File
Disconnect
Reset
Settings
Filter
Tracer
?

✖
↶
↷
⏏
🔍
📄
📁
📡

Receive

Connect/Disconnect	Data	Period	Count	Comment (click to change)
000	6	81 00 00 00 00 00	101	609
002	8	FF FF FF FF FF FF FF FF	50	1217
003	7	93 3D 47 C1 FF FF 0C	499	122
004	6	54 40 08 01 F0 40	498	123
006	7	18 01 00 00 00 00 00	101	609
010	8	01 80 21 90 03 00 00 00	500	123
011	8	01 00 00 00 00 00 00 00	499	123
013	6	00 00 00 00 00 00 00	500	123
015	6	FF 7B 00 00 00 00 00	102	609
02C	6	FF FF FF FF 00 00	498	123
110	8	01 00 00 00 00 00 00 00	500	123
159	6	00 00 00 E3 00 00	998	61
1A7	6	FF FF FF FC 00 00	99	608
1AE	6	10 00 00 00 00 00 00	998	61
1AF	8	01 07 00 40 10 24 08 00	498	123
1B2	6	00 07 00 00 00 00 00	998	61
1C8	7	93 D4 7C 1F FF F0 C0	0	1
1C8	6	07 00 00 00 00 00 00	499	122
1CA	8	01 04 0C 01 09 24 00 02	500	123
1CB	6	00 00 00 00 00 00 00	999	60
1CB	0		0	1
1D0	8	00 8D 1E CE 01 CF 01 00	101	609
1D1	8	01 82 07 42 E4 28 21 43	499	123
33B	4	FF FF 3F FF	100	609
3F8	6	06 38 00 00 00 00 00	101	609
402	8	FC 02 3F FF FF FF FF FF	998	62
B60	0		0	1





Надеюсь когда-то, кому-то эта запись сэкономит неделю гугления :-)

Ну а я теперь пойду перенесу свои эксперименты со стола в машину...

upd: проверил в авто, на диагностическом разъёме молчит, на подключении к кан приборки работает! я аж кипятком писаюсь! даже штатный автозапуск работает!

arduino + mcp2515 + lancer10 = canhacker!



2 мая 2024 Пожаловаться

13

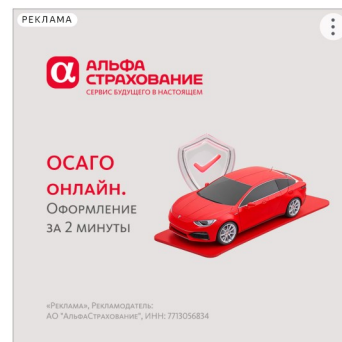
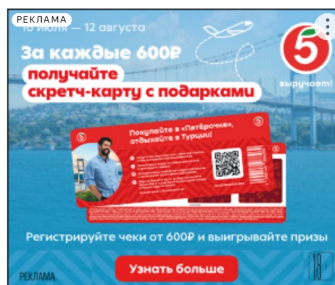
1

9

Поделиться:

Реклама

Разместить рекламу



Смотрите также

Установка климат-контроля
Mitsubishi Lancer

Вторая часть доработок автомобиля, установка ПТФ и многое другое...
Lada Гранта

Продаю свою машину
KIA Sorento

Chrysler, крепкое...

Chrysler 300M

Установка быстросъемных
креплений кнопка на передний
бампер

Mitsubishi Lancer

Накидка на панель приборов

Mitsubishi Lancer

Комментарии 1

Участвовать в обсуждениях могут только участники сообщества.



Djarull

Я езжу на Mitsubishi Lancer X

Не имею хоть сколь нибудь уверенных знаний по этой части, но общий уровень технического образования моего — подсказывает что статья более чем полезная.

Думаю стоит грамотные хеш-теги добавить в конце б/ж, для облегчения поиска другими людьми.

4 3 месяца