



---

**Программа для микроконтроллера MDR1215LGI  
для реализации функции SPD (Serial Presence Detect)**

**Инструкция по обновлению ПО и другим операциям**

## Содержание

1	Введение.....	3
2	Установка вспомогательного ПО .....	4
3	Установка драйвера для программатора СН341 .....	5
4	Обновление ПО SPD.....	6
4.1	Ввод пароля для доступа к обновлению .....	6
4.2	Генерация скрипта загрузки новой версии ПО .....	7
4.3	Выполнение скрипта загрузки новой версии ПО .....	8
4.4	Проверка статуса обновления и версии ПО .....	8
5	Работа с защитой от записи.....	9
5.1	Чтение статуса защиты от записи всех блоков .....	9
5.2	Установка защиты от записи блока n_block .....	9
5.3	Снятие защиты от записи со всех блоков .....	9

## **1 Введение**

Микросхемы MDR1215LGI поставляются с записанным программным обеспечением для функционирования микросхем как схемы последовательного детектирования (SPD – Serial Presence Detect).

Для обновления ПО SPD, загруженного в МК MDR1215LGI, с помощью программатора CH341 потребуется:

- ПК с ОС Linux<sup>1</sup>;
- программатор CH341 (с драйвером к нему);
- скрипты fwupdate.sh, pwd.sh и fwid.sh, которые следует загрузить с сайта компании АО «ПКК Миландр».

Для работы с защитой от записи дополнительно потребуются скрипты gpr.sh, swp.sh и cwp.sh.

---

<sup>1</sup> Проверка функционирования проводилась на Linux 6.8.0-48-generic, дистрибутив Ubuntu 24.04.1 LTS (64-bit)

## **2 Установка вспомогательного ПО**

Для корректной работы скрипта прошивки необходимо установить дополнительные пакеты. Их можно установить командой `sudo apt-get install`:

```
$ sudo apt-get install i2c-tools  
$ sudo apt-get install awk  
$ sudo apt-get install xxd  
$ sudo apt-get install libarchive-zip-perl  
$ sudo apt-get install git
```

### 3 Установка драйвера для программатора CH341

Драйвер для программатора CH341 можно скачать по ссылке:

<https://github.com/frank-zago/ch341-i2c-spi-gpio/tree/master>.

Скачивание драйвера ch341-i2c-spi-gpio и переход в директорию после его скачивания осуществляется командами:

```
$ git clone https://github.com/frank-zago/ch341-i2c-spi-gpio
```

```
$ cd ch341-i2c-spi-gpio
```

После загрузки необходимо выполнить сборку драйвера с помощью команды make:

```
$ make
```

После сборки драйвера требуется загрузить следующие модули для работы с i2c-устройствами:

```
$ sudo insmod ch341-core.ko
```

```
$ sudo insmod i2c-ch341.ko
```

После подключения программатора CH341 к ПК при выполнении команды i2cdetect должно отобразиться новое устройство i2c:

```
$ sudo i2cdetect -l
```

```
user24@pc-support:~/SPD/drivers/ch341-i2c-spi-gpio$ sudo i2cdetect -l
i2c-0  i2c          igb BB                I2C adapter
i2c-1  i2c          igb BB                I2C adapter
i2c-2  i2c          igb BB                I2C adapter
i2c-3  i2c          igb BB                I2C adapter
i2c-4  smbus        SMBus I801 adapter at f040  SMBus adapter
i2c-5  i2c          i915 gmbus dpc        I2C adapter
i2c-6  i2c          i915 gmbus dpb        I2C adapter
i2c-7  i2c          i915 gmbus dpd        I2C adapter
i2c-8  i2c          AUX D/DDI D/PHY D     I2C adapter
i2c-9  i2c          CH341 I2C USB bus 001 device_006  I2C adapter
```

Рисунок 1 – Список устройств по команде i2cdetect -l

*ВАЖНО! Если ранее были загружены другие модули для работы с CH341, то их необходимо удалить из ядра с помощью команды `rmmod`. Для вывода загруженных модулей ядра можно воспользоваться командой `lsmod`.*

## 4 Обновление ПО SPD

### 4.1 Ввод пароля для доступа к обновлению

ПО SPD поддерживает функцию обновления основной и загрузочной программы по шине I2C, адрес для обновления –  $0x18 + SA[2:0]$ , однако устройство FUA (Firmware Update Address) по умолчанию скрыто на шине I2C после подачи питания. Для его отображения необходимо предъявить пароль устройству SPD.

Процедура предъявления пароля заключается в подаче 8 команд чтения по заданному адресу по 1 байту без подачи других команд между ними.

Пароль представляет собой последовательность адресов, по которым необходимо выполнить чтение 1 байта.

Текущий пароль: 0xFB, 0x0D, 0xDE, 0x39, 0x1B, 0x64, 0x35, 0xC5.

Перед вводом пароля необходимо проверить, на какой шине определяется программатор CH341.

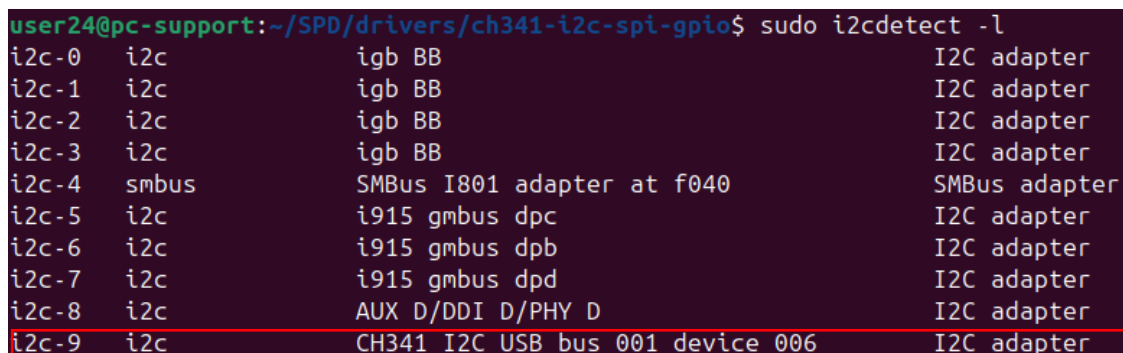
Определение шины I2C для программатора CH341 выполняется с помощью команды:

```
$ sudo i2cdetect -l
```

Результат выполнения команды `i2cdetect` представлен на рисунке 2.

Необходимо убедиться, что программатор есть в списке и определить номер его шины:

```
i2c-9  i2c      CH341 I2C USB bus 001 device 006      I2C adapter
```



```
user24@pc-support:~/SPD/drivers/ch341-i2c-spi-gpio$ sudo i2cdetect -l
i2c-0  i2c      igb BB                                I2C adapter
i2c-1  i2c      igb BB                                I2C adapter
i2c-2  i2c      igb BB                                I2C adapter
i2c-3  i2c      igb BB                                I2C adapter
i2c-4  smbus    SMBus I801 adapter at f040           SMBus adapter
i2c-5  i2c      i915 gmbus dpc                       I2C adapter
i2c-6  i2c      i915 gmbus dpb                       I2C adapter
i2c-7  i2c      i915 gmbus dpd                       I2C adapter
i2c-8  i2c      AUX D/DDI D/PHY D                   I2C adapter
i2c-9  i2c      CH341 I2C USB bus 001 device 006     I2C adapter
```

Рисунок 2 – Список устройств по команде `i2cdetect -l`

Ввод пароля реализован в скрипте `pwd.sh`. Скрипт вызывается с двумя аргументами: номер шины I2C (`n_bus`) и адрес SPD (`spd_addr = 0x50 + SA[2:0]`).

```
$ sudo bash pwd.sh n_bus spd_addr
```

```

user24@pc-support:~/SPD/scripts/etc$ sudo bash pwd.sh 9 0x50
0x00
0x03
0x00
0x00
0x11
0x00
0x00
0x00
0x00
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:
10: -- -- -- -- -- -- -- -- -- 18 -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: 30 31 -- -- 34 35 36 -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
    
```

Рисунок 3 – Результат работы скрипта ввода пароля pwd.sh

После ввода пароля устройство FUA отображается и виден его адрес (в данном примере – 18). Адрес может меняться в зависимости от состояния на выводах SA[2:0].

Далее можно приступить к обновлению ПО SPD.

## 4.2 Генерация скрипта загрузки новой версии ПО

Для этого необходимо воспользоваться скриптом fwupdate.sh, который на основе заданных параметров генерирует скрипт для загрузки новой версии ПО в МК MDR1215LGI.

Запуск скрипта fwupdate.sh с помощью интерпретатора bash осуществляется следующей командой:

```
$ sudo bash fwupdate.sh
```

После запуска скрипта в fwupdate.sh в интерактивном режиме необходимо:

- ввести имя bin-файла (при необходимости с относительным путем), содержащего новую версию ПО SPD (основная или загрузочная программа);
- ввести адрес обновления ПО, полученный после ввода пароля, без 0x, например, 18;
- ввести номер шины I2C для программатора CH341;
- ввести тип ПО SPD: основная программа – 1 или загрузочная программа – 0.

Пример работы скрипта представлен на Рисунок 4.

```

user24@pc-support:~/SPD/scripts/fwupdate$ sudo bash fwupdate.sh
Insert filename for load to MEM (default ):SPD.bin
Continue? (Y/N): y
WRITESCRIPT=SPD.bin.sh
Insert address to write:18
Insert bus address:9
Insert place to write 0 (boot) 1 (programm):1
    
```

Рисунок 4 – Результат работы скрипта fwupdate.sh

После ввода всех параметров в той же директории, откуда запускался fwupdate.sh, будет сгенерирован скрипт для загрузки нового ПО в МК MDR1215LGI с именем «имя\_новой\_прошивки.bin.sh».

*ВАЖНО! Перед очередным выполнением скрипта fwupdate.sh необходимо удалить предыдущий сгенерированный им скрипт записи «имя\_новой\_прошивки.bin.sh».*

### 4.3 Выполнение скрипта загрузки новой версии ПО

Запуск сгенерированного скрипта «имя\_новой\_прошивки.bin.sh» с помощью интерпретатора bash выполняется с помощью команды:

```
$ sudo bash имя_новой_прошивки.bin.sh
```

В процессе выполнения скрипт выведет в терминал сообщения «Start (начало загрузки ПО SPD)» и «Done» (конец загрузки ПО SPD). После выполнения скрипта SPD перезагрузится, отображение адреса adr\_upd, по которому можно узнать статус обновления и версию ПО, снова станет скрытым, поэтому необходимо еще раз **ввести пароль**, который вводился перед прошивкой. Если осуществлялось только обновление boot, то повторно вводить пароль не нужно.

```
user24@pc-support:~/SPD/scripts/fwupdate$ sudo bash SPD.bin.sh
Start
Done
```

Рисунок 5 – Отработка скрипта прошивки

### 4.4 Проверка статуса обновления и версии ПО

Статус обновления, версию ПО и дополнительную информацию о прошивке можно узнать с помощью чтения специальных адресов. Данная процедура реализована в скрипте fwid.sh:

```
$ sudo bash fwid.sh n_bus adr_upd
```

где n\_bus – номер шины I2C для программатора CH341, adr\_upd – адрес для обновления ПО в формате 0xhh.

Считанное значение статуса обновления ПО (Manufacturer ID): 0xaa – обновление прошло успешно, 0xee – ошибка записи.

Считанное значение версии ПО (Device/Revision): 0x22hh, где hh – идентификационный номер ПО.

Считанная дополнительная информация (Vendor Details) отображает специфику прошивки (например, THERMO supported).

```
user24@pc-support:~/SPD/scripts/etc$ ./fwid.sh 0 0x52
Manufacturer ID: 0xaa
Device/Revision: 0x2285
Vendor Details: 0x0001
- THERMO supported
```

Рисунок 6 – Статус обновления и версия ПО

## 5 Работа с защитой от записи

### 5.1 Чтение статуса защиты от записи всех блоков

Скрипт rps.sh предназначен для проверки статуса защиты от записи всех блоков памяти устройства.

Запуск скрипта:

```
$ sudo bash rps.sh n_bus
```

где n\_bus – номер шины I2C для программатора CH341.

После выполнения скрипт выведет статус защиты для каждого из четырех блоков памяти.

```
user24@pc-support:~/SPD/scripts/etc$ ./rps.sh 0
Block 0: Unprotected
Block 1: Unprotected
Block 2: Unprotected
Block 3: Unprotected
```

Рисунок 7 – Чтение статуса защиты

### 5.2 Установка защиты от записи блока n\_block

Скрипт swp.sh позволяет установить защиту от записи для выбранного блока памяти.

Запуск скрипта:

```
$ sudo bash swp.sh n_bus n_block
```

где n\_bus – номер шины I2C для программатора CH341, а n\_block – номер блока памяти (0-3).

После выполнения скрипт установит защиту от записи для указанных блоков памяти.

```
user24@pc-support:~/SPD/scripts/etc$ ./swp.sh 0 0
user24@pc-support:~/SPD/scripts/etc$ ./swp.sh 0 1
user24@pc-support:~/SPD/scripts/etc$ ./swp.sh 0 2
user24@pc-support:~/SPD/scripts/etc$ ./swp.sh 0 3
user24@pc-support:~/SPD/scripts/etc$ ./rps.sh 0
Block 0: Protected
Block 1: Protected
Block 2: Protected
Block 3: Protected
```

Рисунок 8 – Установка защиты от записи блоков 0-3

### 5.3 Снятие защиты от записи со всех блоков

Скрипт cwp.sh используется для снятия защиты от записи со всех блоков памяти устройства.

Запуск скрипта:

```
$ sudo bash cwp.sh n_bus
```

где n\_bus – номер шины I2C для программатора СН341.

После выполнения скрипт снимает защиту от записи со всех блоков памяти устройства.

```
user24@pc-support:~/SPD/scripts/etc$ ./cwp.sh 0
user24@pc-support:~/SPD/scripts/etc$ ./rps.sh 0
Block 0: Unprotected
Block 1: Unprotected
Block 2: Unprotected
Block 3: Unprotected
```

Рисунок 9 – Снятие защиты со всех блоков

