



**Программа для микроконтроллера MDR1219LGI
для реализации функции SPD (Serial Presence Detect)
для DDR5**

**Инструкция по обновлению ПО и другим
операциям**

Содержание

1	Введение	3
2	Подготовка в ОС Windows	4
2.1	Установка драйвера для программатора СН341.....	4
2.2	Установка Python	4
2.3	Установка и запуск скриптов	4
3	Подготовка в ОС Linux.....	7
3.1	Установка вспомогательного ПО.....	7
3.2	Установка и запуск скриптов	7
4	Работа с SPD	9
4.1	Чтение профиля SPD	9
4.2	Запись профиля SPD.....	9
4.3	Установка защиты от записи	10
4.4	Снятие защиты от записи.....	10
4.5	Проверка состояния защиты от записи	10
4.6	Чтение произвольного регистра MR.....	10
4.7	Запись произвольного регистра MR	10
4.8	Считывание показаний температурного датчика.....	10
4.9	Принудительная установка офлайн режима	11
4.10	Принудительный выход из офлайн режима.....	11
4.11	Проверка офлайн режима	11
4.12	Обновление ПО	11

1 Введение

Микросхемы MDR1219LGI поставляются с записанным программным обеспечением для функционирования микросхем как схемы последовательного детектирования (SPD – Serial Presence Detect).

Для обновления ПО SPD, загруженного в МК MDR1219LGI, а также других действий в ОС Windows потребуется:

- ПК с ОС Windows¹;
- программатор CH341 (с драйвером к нему);
- интерпретатор Python для Windows;
- Python-скрипты из архива «offline_tools», которые следует загрузить с сайта компании АО «ПКК Миландр».

Для тех же действий в ОС Linux понадобятся:

- ПК с ОС Linux²;
- программатор CH341;
- пакет pyusb для Python;
- Python-скрипты из архива «offline_tools», которые следует загрузить с сайта компании АО «ПКК Миландр».

¹ Проверка функционирования проводилась на Windows 11 Pro (64-bit)

² Проверка функционирования проводилась на Linux 6.8.0-48-generic, дистрибутив Ubuntu 24.04.1 LTS (64-bit)

2 Подготовка в ОС Windows

2.1 Установка драйвера для программатора CH341

Драйвер для программатора CH341 можно скачать на странице https://www.wch-ic.com/downloads/CH341PAR_ZIP.html.

После загрузки необходимо выполнить установить драйвер CH341PAR/SETUP.EXE

После установки в диспетчере устройств должно появиться устройство USB-EPP/I2C... CH341A:

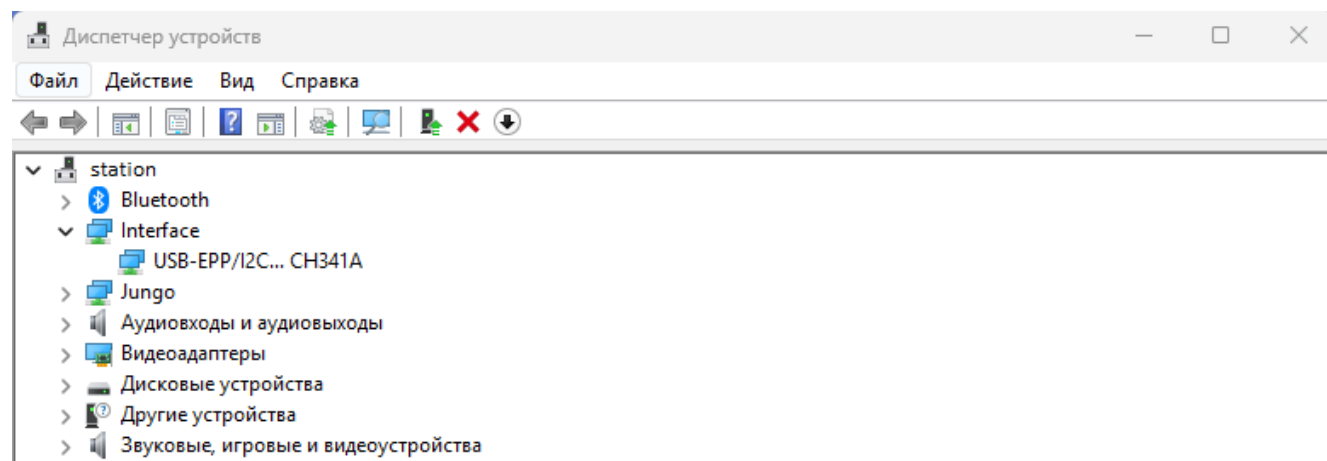


Рисунок 1 – Программатор CH341 в диспетчере устройств

2.2 Установка Python

Python можно скачать по ссылке:

<https://www.python.org/downloads/windows/>

Актуальной версией на момент написания настоящей инструкции является 3.13.14.

2.3 Установка и запуск скриптов

Для установки скриптов требуется загрузить архив «offline_tools» с сайта компании АО «ПМК Миландр» (<https://ic.milandr.ru/soft/>), после чего распаковать архив в удобное место, например, C:\Milandr\offline_tools.

Для запуска нужно запустить командную строку и перейти в каталог с распакованными скриптами, после чего выполнить команду:

```
python top.py
```

Если драйвер программатора CH341A установлен неправильно или программатор не подключен к USB, будет показано сообщение об ошибке:

```
PS C:\Milandr\offline_tools> python top.py
```

```
*****  
Start 2026-07-28 15:11:38
```

```
Error! USB CH341 Open Failed!
USB CH341Dev Index= 0 is not ready
Failed to issue I2C Set Speed Command
```

Если программатор подключен, драйвер установлен, но SPD не подключен по I2C (либо отсутствует его питание), также будет отображаться ошибка:

```
PS C:\Milandr\offline_tools> python top.py
```

```
*****
Start 2026-07-28 15:14:16
Speed set to 100
No SPD device found
```

Скрипт устроен таким образом, что при запуске сканирует устройства в диапазоне адресов SPD. При успешном запуске будет найдено одно устройство в диапазоне адресов 0x50 – 0x57 (в зависимости от сопротивления между выводом HSA до GND):

```
PS C:\Milandr\offline_tools> python top.py
```

```
*****
Start 2026-07-28 15:19:33
Speed set to 100
Found SPD devices:
    0x50
Using first device at 0x50 (HID = 0)
Device type:          SPD5 HUB Device w/ Temperature Sensor (MR0 =
0x51, MR1 = 0x18)
Device revision:0.0 (MR2 = 0x0)
Vendor ID:           0x4d (MR3), 0x49 (MR4)
                    Milandr SPD5 device 1.0.3
Capabilities:         HUB, No temp. sensor (MR5 = 0x1)
Offline mode:         YES (MR48 = 0x4)
Write protection:     YES (MR12 = 0xa0, MR13 = 0xad)
Responses from local bus:
0x0
0x8
0x10
0x18
0x20
0x28
0x30
0x38
0x40
0x48: <- device
0x50: <- device
```

0x58
0x60
0x68
0x70
0x78

Select task:

- 1 - read SPD NVMEM to file
- 2 - write SPD NVMEM from file
- 3 - set SPD NVMEM write protection
- 4 - reset SPD NVMEM write protection
- 5 - check SPD NVMEM write protection
- 6 - read MR
- 7 - write MR
- 8 - read temperature sensor
- ***** Milandr SPD5 only *****
- 9 - set offline mode
- 10 - reset offline mode
- 11 - check offline mode
- 12 - reset and clear NVMEM
- 13 - read SA pins captured result
- 14 - read temperature sensor (raw data)
- 15 - read unique_id
- 16 - update firmware
- 0 - exit

Скрипт будет работать с первым устройством из диапазона 0x50 – 0x57. Выбор нужной функции осуществляется через ввод номера на клавиатуре и подтверждение клавишей ВВОД.

Функции 12, 13, 14 и 15 отладочные, предназначены для внутреннего использования и в настоящем документе не описываются.

После выполнения выбранной функции скрипт независимо от результата (успешное выполнение или с ошибкой) завершает работу. Если нужно продолжить работу, следует снова запустить его командой `python top.py`.

Пункты 1 – 8, 11 относятся к стандартным возможностям SPD5. Остальные пункты – только к MLDR1219LGI. Расширение стандартного набора возможностей реализовано через зарезервированные регистры MR126 и MR127.

3 Подготовка в ОС Linux

3.1 Установка вспомогательного ПО

Для корректной работы скрипта прошивки необходимо установить дополнительные пакеты для Python. Предполагается что python3 и libusb уже установлены в системе.

Рекомендуется работать в виртуальном окружении Python чтобы избежать конфликтов с системными библиотеками. Виртуальное окружение создается командой:

```
$ python3 -m venv venv
```

После чего активируется командой:

```
$ source venv/bin/activate
```

Выход из виртуального окружения выполняется командой:

```
$ deactivate
```

Активировать созданное виртуальное окружение нужно однократно перед работой со скриптами. Дальнейшие команды выполняются внутри виртуального окружения.

Установка pyusb:

```
$ pip install pyusb
```

Альтернативный способ – скачать и установить *.whl файл, например, по ссылке <https://www.piwheels.org/project/pyusb/> (версия файла может отличаться):

```
$ pip install ./pyusb-1.3.1-py3-none-any.whl
```

Проверка что pyusb установлен:

```
$ pip show pyusb
```

или:

```
$ pip list
```

При успешной установке в выводе команды `pip list` будет присутствовать pyusb соответствующей версии:

```
Package Version
-----
pip       24.0
pyusb     1.3.1
```

3.2 Установка и запуск скриптов

Для установки скриптов требуется загрузить архив «offline_tools» с сайта компании АО «ПМК Миландр» (<https://ic.milandr.ru/soft/>), после чего распаковать архив в удобное место, например, ~/offline_tools и перейти в него.

Работа с СН341 в Linux требует прав администратора. Чтобы дать доступ к СН341 без sudo, нужно установить правило:

```
$ sudo cp 99-ch341.rules /etc/udev/rules.d/
```

```
$ sudo udevadm control --reload-rules
```

```
$ sudo udevadm trigger
```

Файл 99-ch341.rules находится внутри папки offline_tools. После этого можно запускать скрипт командой (внутри виртуального окружения):

```
$ python top.py
```

Если по каким-либо причинам установка прав нежелательна, можно запускать скрипт с sudo, непосредственно указывая интерпретатор (для этого необязательно находиться в виртуальном окружении Python):

```
$ sudo ./venv/bin/python top.py
```

Диагностические сообщения и общие сведения соответствуют описанию в разделе 2.3.

4 Работа с SPD

4.1 Чтение профиля SPD

Для считывания 1024 байт профиля SPD и сохранения его в файл нужно выбрать функцию с номером 1:

1 - read SPD NVMEM to file.

Содержимое памяти профиля будет считано и сохранено в подкаталоге dumps/ с именем файла вида dump_<TIMESTAMP>_<ADDRESS>_<CRC32>.bin. Файл сохраняется в бинарном виде.

4.2 Запись профиля SPD

Для записи 1024 байт профиля SPD из файла нужно выбрать функцию с номером 2:

2 - write SPD NVMEM from file

После этого будет предложено выбрать требуемый файл из подкаталога dumps/:

Testing SPD device 0x50

* Run test write_dump on 0x50

<module 'write_dump' from

'C:\\Milandr\\offline_tools\\write_dump.py'>

Found files in dumps\:

1: 1024-byte-file.bin

2: dump_ADATA_8GB_DDR5_4800.bin

3: dump_Apacer_8GB_DDR5_4800.bin

4: dump_ELPITECH_8GB_DDR5_4800_ET417R103S910003.bin

5: dump_ELPITECH_8GB_DDR5_4800_ET417R103S910004.bin

6: dump_KingFast_16GB_DDR5_5600_24YK121100003.bin

7: dump_KingFast_16GB_DDR5_5600_24YK121100004.bin

8: dump_Kingston_Fury_8GB_DDR5_4800.bin

9: dump_Kingston_Fury_8GB_DDR5_4800_num2.bin

10: dump_Patriot_8GB_DDR5_4800.bin

11: dump_Patriot_8GB_DDR5_4800_num2.bin

12: dump_TMI_16GB_DDR5_4800_0B7A392512000000008.bin

13: dump_TMI_16GB_DDR5_4800_0B7A39251200000000C.bin

14: dump_TMI_8GB_DDR5_4800_0B7A372511000000007.bin

15: dump_TMI_8GB_DDR5_4800_0B7A372511000000008.bin

16: dump_Wolveswalk_16GB_DDR5_5600.bin

Select file (1 - 16):

Выбор файла осуществляется так же, как и выбор нужной функции. После подтверждения происходит запись и верификация данных. Файл должен быть в бинарном виде. Процедура записи профиля может завершиться с ошибкой, если установлены какие-либо биты защиты. Если защита снята, записывать профиль можно и в онлайн- и в офлайн-режиме.

4.3 Установка защиты от записи

Для установки защиты от записи нужно выбрать пункт:

3 - set SPD NVMEM write protection

Скрипт устанавливает защиту от записи всех блоков. Установить защиту от записи можно в любом режиме, онлайн или офлайн.

4.4 Снятие защиты от записи

Для сброса защиты от записи нужно выбрать пункт:

4 - reset SPD NVMEM write protection

Скрипт снимает защиту от записи всех блоков. Снять защиту от записи можно только в офлайн-режиме.

4.5 Проверка состояния защиты от записи

Для проверки состояния защиты от записи нужно выбрать пункт:

5 - check SPD NVMEM write protection

Скрипт выдаст информацию об установленной или снятой защите от записи. Проверить состояние защиты от записи можно в любом режиме, онлайн или офлайн.

4.6 Чтение произвольного регистра MR

Для чтения регистра нужно выбрать пункт:

6 - read MR

Затем указать адрес регистра в десятичной или шестнадцатеричной системе счисления (например, для чтения регистра MR51 ввести число без префикса, например, 51 или с префиксом - 0x33).

4.7 Запись произвольного регистра MR

Для записи регистра нужно выбрать пункт:

7 - write MR

А затем указать адрес регистра и значение, записываемое в него. Адрес и значение указываются в десятичной или шестнадцатеричной системе счисления, как описано в п. 5.6.

4.8 Считывание показаний температурного датчика

Чтобы считать результат измерения температуры, нужно выбрать пункт:

8 - read temperature sensor

Температурный датчик реализован в ПО версии 1.0.5 и выше. Если используется более ранняя версия ПО или напряжение питания ниже 2.45 В, температурный датчик не обрабатывается.

4.9 Принудительная установка офлайн режима

В ПО SPD5 MLDR1219LGI есть возможность принудительно перевести SPD5 в офлайн-режим. Если при включении питания вывод HSA подключен к земле через резистор 10 кОм и более, активируется онлайн режим – штатная работа в составе материнской платы. Если вывод HSA подключен к земле напрямую, активируется офлайн-режим – в этом режиме в дополнение к обычным функциям разрешается снятие защиты памяти. Состояние вывода HSA проверяется только при включении питания, после того как был определен адрес SPD5 и режим работы, вывод HSA более не обрабатывается. Для принудительного перехода из онлайн-режима в офлайн-предусмотрена эта команда. При выполнении этой команды SPD5 не изменяет адрес, полученный в результате измерения внешнего сопротивления между HSA и землей.

9 - set offline mode

4.10 Принудительный выход из офлайн режима

Операция, обратная п. 5.9. Переводит SPD5 в онлайн-режим.

10 - reset offline mode

4.11 Проверка офлайн режима

Выводит текущее состояние SPD5: нормальный рабочий или офлайн-режим.

11 - check offline mode

4.12 Обновление ПО

Обновить программное обеспечение SPD5 можно с помощью пункта:

16 - update firmware

Аналогично выбору файла для записи профиля, скрипт предложит выбрать файл прошивки из подкаталога /firmware. Ниже показан лог успешного обновления ПО. Возможность обновления ПО не зависит от онлайн или офлайн режима.

```
Testing SPD device 0x50
* Run test update_fw on 0x50
<module 'update_fw' from 'C:\\Milandr\\offline_tools\\update_fw.py'>
Found files in firmware\:  
1: spd5-mlldr1206_8d7f129b_v1.0.5.bin  
2: spd5-mlldr1206_b0a5a0cf_v1.0.5.bin  
3: spd5-mlldr1206_c9dcab59_v1.0.5.bin  
4: spd5-mlldr1206_faf0aeb9_v1.0.5.bin  
5: spd5-mlldr1206_fd2e4874_v1.0.5.bin  
6: spd5_v1.0.3.bin  
7: spd5_v1.0.4.bin  
8: spd5_v1.0.5_rc1_7f665fee.bin  
9: spd5_v1.0.5_rc2_336e86f5.bin  
10: spd5_v1.0.5_rc3_442185b3.bin
```

```
Select file (1 - 10):
Update FW with file:
firmware\spd5_v1.0.5_rc3_442185b3.bin
Erasing buffer...
Written 4096 of 29308
Written 8192 of 29308
Written 12288 of 29308
Written 16384 of 29308
Written 20480 of 29308
Written 24576 of 29308
Written 28672 of 29308
Written 29308 of 29308
Wait for 5 seconds...
Wait for 4 seconds...
Wait for 3 seconds...
Wait for 2 seconds...
Wait for 1 seconds...
Result: Success (MR126 = 0x0)
* End test update_fw, result: 0 , 32.776905 s
```

=====

Summary:

```
    update_fw OK
Total 1 tests
Time 35.036101 s
```

[illegible]