

HM2007

Создано системой Doxygen 1.5.6

Wed Sep 24 00:04:50 2008



# Оглавление

1	Титульная страница	1
2	Алфавитный указатель групп	3
2.1	Группы	3
3	Группы	5
3.1	НМ2007	5
3.1.1	Функции	7
3.1.1.1	hm_download	7
3.1.1.2	hm_init	7
3.1.1.3	hm_pattern_size	7
3.1.1.4	hm_read_byte	8
3.1.1.5	hm_ready	8
3.1.1.6	hm_recog	8
3.1.1.7	hm_reset	8
3.1.1.8	hm_result	8
3.1.1.9	hm_start_upload	9
3.1.1.10	hm_train	9
3.1.1.11	hm_wlen	9
3.1.1.12	hm_write_byte	10



# Глава 1

## Титульная страница

### Краткое описание микросхемы

НМ2007 - микросхема, предназначенная для записи и последующего распознавания речевых команд. Она требует минимум обвязки : 8 KiB ОЗУ, электретный микрофон, несколько резисторов и конденсаторов. Могут распознаваться или до 40 команд длиной до 0.9 с, или до 20 команд длиной до 1.9 с. Для связи с внешним контроллером используются 8 линий:

- 4-битная двунаправленная шина данных (K1-K4)
- 3 линии управления шиной данных (S1-S3)
- линия выбора длины команды (WLEN)

### Данные шаблонов

Шаблоны для распознавания, записанные с микрофона, сохраняются в ОЗУ. Размер шаблона не фиксирован и может составлять от 30 до 240 байт (определено экспериментальным путем). Даташит микросхемы не содержит информации о максимальном размере шаблона. Естественно, при выключении питания шаблоны теряются. Для долговременного хранения шаблонов следует переносить их в энергонезависимую память и восстанавливать при включении питания.



## Глава 2

# Алфавитный указатель групп

### 2.1 Группы

Полный список групп.

НМ2007 . . . . . 5





## Глава 3

# Группы

### 3.1 НМ2007

#### Функции

- void `hm_init` (void)  
Инициализация.
- void `hm_reset` (void)  
Очистка памяти.
- uint8\_t `hm_ready` (void)  
Проверка статуса микросхемы.
- void `hm_recog` (void)  
Переключение в режим распознавания.
- void `hm_train` (uint8\_t pattern)  
Переключение в режим обучения.
- uint16\_t `hm_result` (void)  
Результат распознавания.
- uint16\_t `hm_pattern_size` (uint8\_t pattern)  
Получить размер области памяти, необходимой для сохранения данных шаблона команды.
- void `hm_wlen` (uint8\_t len)  
Устанавливает длительность команд.
- void `hm_download` (uint16\_t size, uint8\_t \*buf)  
Считывает шаблон команды из памяти микросхемы и заносит его в буфер.
- uint8\_t `hm_read_byte` (void)  
Считывает очередной байт шаблона команды.

- void `hm_start_upload` (uint8\_t pattern, uint16\_t len)  
Иницирует процесс переноса шаблона в ОЗУ, подключенное к микросхеме.
- void `hm_write_byte` (uint8\_t byte)  
Записывает очередной байт шаблона в ОЗУ, подключенное к микросхеме.

## Конфигурация портов

Библиотека для работы с микросхемой распознавания речевых команд HM2007.

Подключение микросхемы.

- выводы S1-S3 подключены к выводам одного и того же порта, к последовательным выводам. Например, если S1 подключен к PD3, то S2 должен быть подключен к PD4, а S3 - к PD5
- выводы K1-K4 подключены к выводам одного и того же порта, к последовательным выводам. Например, если K1 подключен к PB2, то K2 должен быть подключен к PB3, K3 - к PB4, а K4 - к PB5.
- вывод WLEN подключен к любому выводу любого порта.
- выводы S1-S3 подтянуты к земле резисторами 5 .. 10 КОм
- `#define HM_S_PORT PORTB`  
Регистр вывода порта, к которому подключены линии S1-S3.
- `#define HM_S_DDR DDRB`  
Регистр направления для порта с линиями S1-S3.
- `#define HM_S_BIT 0`  
Номер бита, к которому подключена линия S1.
- `#define HM_K_PORT PORTC`  
Регистр вывода порта, к которому подключены линии K1-K4.
- `#define HM_K_DDR DDRC`  
Регистр направления порта с линиями K1-K4.
- `#define HM_K_PIN PINC`  
Регистр ввода порта, к которому подключены линии K1-K4.
- `#define HM_K_BIT 4`  
Номер бита, к которому подключена линия K1.
- `#define HM_WL_PORT PORTC`

Регистр вывода порта, к которому подключена линия WLEN.

- `#define HM_WL_DDR DDRC`

Регистр направления для порта с линией WLEN.

- `#define HM_WL_BIT 3`

Номер бита, к которому подключена линия WLEN.

### 3.1.1 Функции

#### 3.1.1.1 void hm\_download (uint16\_t size, uint8\_t \* buf)

Считывает шаблон команды из памяти микросхемы и заносит его в буфер.

Эта функция должна вызываться сразу после `hm_pattern_size()`. Разумеется, данные, помещенные в буфер, а также длину области памяти следует затем перенести в энергонезависимую память (флэш-память контроллера, внешнее ПЗУ, карта памяти и т.п.). Для восстановления шаблона при очередном включении питания используется `hm_start_upload()` и `hm_write_byte()`.

Аргументы:

size Размер области памяти, возвращенный функцией `hm_pattern_size()`.

buf Указатель на буфер. Буфер должен вмещать минимум size байт.

#### 3.1.1.2 void hm\_init (void)

Инициализация.

Инициализирует используемые порты микроконтроллера для обмена с HM2007. Эта функция должна быть вызвана в самом начале работы с микросхемой. Устанавливает длительность команд 0.9 сек.

См. также:

`hm_wlen()`

#### 3.1.1.3 uint16\_t hm\_pattern\_size (uint8\_t pattern)

Получить размер области памяти, необходимой для сохранения данных шаблона команды.

Данные в ОЗУ при выключении питания теряются. Если необходимо сохранять данные между включениями, их следует сохранять в энергонезависимой памяти. `hm_pattern_size()` возвращает размер области памяти, которая необходима для сохранения шаблона команды. Сразу после вызова этой функции необходимо вызвать или `hm_download()`, передав ей указатель на область памяти достаточного размера, или `hm_read_byte()`.

Аргументы:

pattern Номер команды. От 1 до 20 или 40, в зависимости от режима работы.

Возвращает:

Размер области памяти. Всегда кратен 4

См. также:

[hm\\_wlen\(\)](#)

Предупреждения:

Если передать номер команды, которая не была записана, или недопустимый номер, микросхема "зависает".

#### 3.1.1.4 uint8\_t hm\_read\_byte (void)

Считывает очередной байт шаблона команды.

После вызова [hm\\_pattern\\_size\(\)](#), эта функция должна быть вызвана столько раз, какое значение вернула [hm\\_pattern\\_size\(\)](#).

Возвращает:

Очередной байт шаблона.

#### 3.1.1.5 uint8\_t hm\_ready (void)

Проверка статуса микросхемы.

Возвращает:

1 если микросхема готова к приему очередной команды, иначе 0

#### 3.1.1.6 void hm\_recog (void)

Переключение в режим распознавания.

После вызова этой функции, следует проверять значение, возвращаемое функцией [hm\\_ready\(\)](#). Другие функции вызывать нельзя до тех пор, пока [hm\\_ready\(\)](#) не вернет 1. Результат распознавания затем может быть получен с помощью [hm\\_result\(\)](#).

#### 3.1.1.7 void hm\_reset (void)

Очистка памяти.

Все сохраненные команды стираются.

#### 3.1.1.8 uint16\_t hm\_result (void)

Результат распознавания.

Возвращает номер команды распознанной функцией [hm\\_recog\(\)](#) и вероятность распознавания.

Возвращает:

Целое без знака.

Младшие 8 бит содержат номер распознанной команды.

Старшие 8 бит содержат вероятность распознавания от 0 до 255. Если вероятность равна 255, значит команда не была распознана. 0 - 100% совпадение. Практически возвращаются значения от примерно 32 до 255.

См. также:

[hm\\_recog\(\)](#)

#### 3.1.1.9 void hm\_start\_upload (uint8\_t pattern, uint16\_t len)

Иницирует процесс переноса шаблона в ОЗУ, подключенное к микросхеме.

Эта функция используется для восстановления данных, ранее полученных с использованием функций [hm\\_pattern\\_size\(\)](#), [hm\\_download\(\)](#) и/или [hm\\_read\\_byte\(\)](#). После вызова этой функции следует len раз вызвать [hm\\_write\\_byte\(\)](#).

Аргументы:

pattern Номер команды

len Количество байт данных для записи. Соответствует значению, возвращенному функцией [hm\\_pattern\\_size\(\)](#)

#### 3.1.1.10 void hm\_train (uint8\_t pattern)

Переключение в режим обучения.

После вызова этой функции, следует проверять значение, возвращаемое функцией [hm\\_ready\(\)](#). Другие функции вызывать нельзя до тех пор, пока [hm\\_ready\(\)](#) не вернет 1.

Аргументы:

pattern Номер команды которую следует записать. От 1 до 20 или 40, в зависимости от режима работы.

См. также:

[hm\\_wlen\(\)](#)

Предупреждения:

При записи команды следует быть осторожным, чтобы не превысить максимальную длину команды (0.9 или 1.9 секунд, в зависимости от режима работы), иначе возможно "зависание" микросхемы.

#### 3.1.1.11 void hm\_wlen (uint8\_t len)

Устанавливает длительность команд.

Вызов этой функции влияет на допустимый диапазон аргументов функций [hm\\_pattern\\_size\(\)](#) и [hm\\_train\(\)](#).

Аргументы:

len 0 - 0.9 сек ; 1 - 1.9 сек.

Предупреждения:

После вызова этой функции, следует очистить память вызовом [hm\\_reset\(\)](#).

#### 3.1.1.12 void hm\_write\_byte (uint8\_t byte)

Записывает очередной байт шаблона в ОЗУ, подключенное к микросхеме.

Эта функция используется для восстановления данных, ранее полученных с использованием функций [hm\\_pattern\\_size\(\)](#), [hm\\_download\(\)](#) и/или [hm\\_read\\_byte\(\)](#). Должна вызываться сразу после [hm\\_start\\_upload\(\)](#).

Аргументы:

byte Байт для записи

# Предметный указатель

HM2007, [5](#)  
    hm\_download, [7](#)  
    hm\_init, [7](#)  
    hm\_pattern\_size, [7](#)  
    hm\_read\_byte, [8](#)  
    hm\_ready, [8](#)  
    hm\_recog, [8](#)  
    hm\_reset, [8](#)  
    hm\_result, [8](#)  
    hm\_start\_upload, [9](#)  
    hm\_train, [9](#)  
    hm\_wlen, [9](#)  
    hm\_write\_byte, [10](#)  
hm\_download  
    HM2007, [7](#)  
hm\_init  
    HM2007, [7](#)  
hm\_pattern\_size  
    HM2007, [7](#)  
hm\_read\_byte  
    HM2007, [8](#)  
hm\_ready  
    HM2007, [8](#)  
hm\_recog  
    HM2007, [8](#)  
hm\_reset  
    HM2007, [8](#)  
hm\_result  
    HM2007, [8](#)  
hm\_start\_upload  
    HM2007, [9](#)  
hm\_train  
    HM2007, [9](#)  
hm\_wlen  
    HM2007, [9](#)  
hm\_write\_byte  
    HM2007, [10](#)