

Fork

Структура журнала



Содержание

1 Структуры и константы.....	3
1.1 mic_journal_t — структура журнала.....	3
1.2 mic_j_header_t — структура заголовка журнала.....	4
1.3 mic_j_record_t — структура записи.....	5
1.4 mic_j_datetime_t — структура даты и времени.....	6
1.5 mic_j_packed_datetime_t — структура даты и времени (упакованный тип).....	6
1.6 mic_j_var_desc_t — структура описания переменной.....	6
1.7 mic_j_var_t — структура переменной.....	6
1.8 mic_j_header_type_t — типы заголовка (uint8_t).....	7
1.9 mic_j_codepage_t — типы кодировки текста (uint8_t).....	7
1.10 mic_j_sr_modes_t — режимы сохранения и восстановления переменной (uint8_t).....	7
1.11 mic_j_var_ext_info_flags_t — типы переменной и дополнительные флаги (uint8_t).....	7
1.12 mic_j_record_save_mode_t — режимы записи (uint8_t).....	8
1.13 mic_j_config_mods_t — флаги изменения конфигурации (uint8_t).....	9
1.14 mic_j_record_ext_info_flags_t — флаги дополнительной информации о записи (uint8_t).....	9
2 Анализ журнала.....	10
2.1 Анализ заголовка журнала mic_j_header_t.....	10
2.2 Анализ записей mic_j_record_t.....	11

1 Структуры и константы

1.1 mic_journal_t – структура журнала

Наименование	Тип	Описание
<i>struct mic_journal_t {</i>		
header	mic_j_header_t	Заголовок журнала
<i>for (i = 0; i < header.record_count; i++) {</i>		header.record_count записей
record[i]	mic_j_record_t	Запись журнала
<i>}</i>		
<i>}</i>		

1.2 mic_j_header_t – структура заголовка журнала

Наименование	Tun	P	Описание (← P – размер в байтах)
<i>struct mic_j_header_t {</i>			
signature	uint16_t	2	Сигнатура заголовка (0x4D41)
signature_ext	uint16_t	2	Значение 0x4D41 + datetime_last_get_info.second
reserved	uint16_t	2	Зарезервировано (0x0000)
record_size	uint16_t	2	Размер записи журнала mic_j_record_t
record_count	uint16_t	2	Количество записей
header_size	uint16_t	2	Размер заголовка журнала mic_j_header_t (этой структуры); значение кратно 512
journal_size_in_sectors	uint16_t	2	Размер всего журнала mic_journal_t (заголовков + записи) в секторах по 512 байт
net_address	uint8_t	1	Сетевой номер контроллера (если < 0xFF) или признак расширенного заголовка (если == 0xFF)
<i>if (net_address < 0xff) {</i>			<i>Старый тип заголовка</i>
rs232_baudrate	uint32_t	4	Скорость обмена по RS-232 (бод)
rs485_baudrate	uint32_t	4	Скорость обмена по RS-485 (бод)
<i>} else {</i>			<i>Расширенный заголовок</i>
header_type	mic_j_header_type_t	1	Тип заголовка
codepage	mic_j_codepage_t	1	Кодировка текстовых данных (поля version , string[i] и var_desc[i].name)
ext_header[i]	uint8_t[6]	6	Зарезервировано (0x00)
<i>}</i>			
version	char[32]	32	Название прибора и версия ПО для контроллеров МИКОНТ-186 или идентификатор журнала для контроллеров SMART
datetime_first_power_on	mic_j_datetime_t	7	Дата и время первого включения контроллера
datetime_last_power_on	mic_j_datetime_t	7	Дата и время последнего включения контроллера
datetime_last_setup	mic_j_datetime_t	7	Дата и время последнего сохранения настроек
datetime_last_get_info	mic_j_datetime_t	7	Дата и время последнего запроса журнала
string_count	uint8_t	1	Количество строк, содержащих текстовую информацию о контроллере и его изготовителе
<i>for (i = 0; i < string_count; i++) {</i>			<i>string_count строк</i>
string[i]	char[19]	19	Строка, заканчивающаяся нулем (0x00)
<i>}</i>			
var_count_ext	uint8_t	1	Дополнительное количество переменных
var_count_base	uint8_t	1	Базовое количество переменных
<i>for (i = 0; i < var_count_base + var_count_ext; i++) {</i>			<i>var_count_base + var_count_ext описаний переменных</i>
var_desc[i]	mic_j_var_desc_t	8	Описание переменной i
<i>}</i>			
not_used	uint8_t[X]	X	X равен числу байтов для выравнивания размера заголовка журнала mic_j_header_t до ближайшего числа, кратного 512
<i>}</i>			

1.3 mic_j_record_t – структура записи

Наименование	Tun	P	Описание (← P – размер в байтах)
<i>struct mic_j_record_t {</i>			
<i>switch (ext_info_flags & MIC_J_REI_RECORD_TYPE_MASK) {</i>			
<i>case MIC_J_REI_RECORD_TYPE_OLD:</i>			<i>Старый тип записи</i>
datetime	mic_j_datetime_t	7	Дата и время сохранения записи
u	uint16_t	2	Напряжение питания сети (В) × 100
f	uint16_t	2	Частота сети (Гц) × 100
t	uint16_t	2	Температура внутри корпуса (°C) × 100
<i>break;</i>			
<i>case MIC_J_REI_RECORD_TYPE_EXT:</i>			<i>Расширенный тип записи</i>
datetime	mic_j_packed_datetime_t	6	Дата и время сохранения записи (упакованный тип)
config_mods	mic_j_config_mods_t	1	Комбинация флагов изменения конфигурации
u	uint16_t	2	Напряжение питания сети (В) × 100
f	uint16_t	2	Частота сети (Гц) × 100
t	uint16_t	2	Температура внутри корпуса (°C) × 100
<i>break;</i>			
<i>case MIC_J_REI_RECORD_TYPE_EXT_RECNO:</i>			<i>Расширенный тип записи с логическим номером</i>
datetime	mic_j_packed_datetime_t	6	Дата и время сохранения записи (упакованный тип)
config_mods	mic_j_config_mods_t	1	Комбинация флагов изменения конфигурации
u	uint16_t	2	Напряжение питания сети (В) × 100
record_no	uint32_t	4	Логический номер записи
<i>break;</i>			
<i>default:</i>			<i>Неизвестный тип заголовка записи</i>
unknown	uint8_t[13]	13	Запись считать недействительной
<i>}</i>			
save_mode	mic_j_record_save_mode_t	1	Режим записи
var_count	uint8_t	1	Количество переменных
ext_info_flags	mic_j_record_ext_info_flags_t	1	Комбинация флагов дополнительной информации о записи
<i>for (i = 0; i < var_count; i++) {</i>			
var[i]	mic_j_var_t	4	Значение переменной i
<i>}</i>			
not_used	uint8_t[X]	X	X равен числу байтов для выравнивания размера записи mic_j_record_t до значения record_size в заголовке журнала mic_j_header_t
<i>}</i>			

1.4 mic_j_datetime_t — структура даты и времени

Наименование	Tun	P	Описание (← P — размер в байтах)
<i>struct mic_j_datetime_t {</i>			
year	uint16_t	2	Год (1996—4000)
month	uint8_t	1	Месяц (1—12)
day	uint8_t	1	День (1—31)
hour	uint8_t	1	Часы (00—23)
minute	uint8_t	1	Минуты (00—59)
second	uint8_t	1	Секунды (00—59)
<i>}</i>			

1.5 mic_j_packed_datetime_t — структура даты и времени (упакованный тип)

Наименование	Tun : количество бит	P	Описание (← P — размер в байтах)
<i>struct mic_j_packed_datetime_t {</i>			
month	uint16_t : 4	2	Месяц (1—12)
year	uint16_t : 12		Год (1996—4000)
minute	uint16_t : 6	2	Минуты (00—59)
hour	uint16_t : 5		Часы (00—23)
day	uint16_t : 5		День (1—31)
msec	uint16_t : 10	2	Миллисекунды (0—999).
second	uint16_t : 6		Секунды (00—59)
<i>}</i>			

1.6 mic_j_var_desc_t — структура описания переменной

Наименование	Tun	P	Описание (← P — размер в байтах)
<i>struct mic_j_var_desc_t {</i>			
attr	uint8_t	1	Зарезервировано (значение неопределено)
name	char[5]	5	Название переменной, заканчивающееся значением 0x00
sr_modes	mic_j_sr_modes_t	1	Комбинация режимов сохранения и восстановления переменной
ext_info_flags	mic_j_var_ext_info_flags_t	1	Комбинация типа переменной и дополнительных флагов
<i>}</i>			

1.7 mic_j_var_t — структура переменной

Наименование	Tun	P	Описание (← P — размер в байтах)
<i>union mic_j_var_t {</i>			
f	float	4	Вещественное число одинарной точности
u	uint32_t		Беззнаковое целое 32-битное число
s	int32_t		Знаковое целое 32-битное число
<i>}</i>			

1.8 mic_j_header_type_t – типы заголовка (uint8_t)

Наименование	Описание	Значение
MIC_J_HEADER_TYPE_EXT	Признак расширенного заголовка	0x02

1.9 mic_j_codepage_t – типы кодировки текста (uint8_t)

Наименование	Описание	Значение
MIC_J_CODEPAGE_WINDOWS_1251	Кодировка Windows-1251	0x00
MIC_J_CODEPAGE_IBM_866	Кодировка IBM 866	0x01
MIC_J_CODEPAGE_UTF_8	Кодировка UTF-8	0x02
MIC_J_CODEPAGE_MAX	Максимальное значение для кодировки	0x02

1.10 mic_j_sr_modes_t – режимы сохранения и восстановления переменной (uint8_t)

Наименование	Описание	Значение
MIC_J_SR_MODE_N	Сохранение в нормальном режиме	0x01
MIC_J_SR_MODE_E	Сохранение при выключении питания и при стирании журнала	0x80
MIC_J_SR_MODE_ER	Сохранение при выключении питания с последующим восстановлением при включении	0xC0

1.11 mic_j_var_ext_info_flags_t – типы переменной и дополнительные флаги (uint8_t)

Наименование	Описание	Значение
MIC_J_VEI_VAR_TYPE_MASK	Маска типа переменной	0b1111 (0x0F)
MIC_J_VEI_VAR_TYPE_FLOAT	Вещественное число одинарной точности (32 бита)	0b0000 (0x00)
MIC_J_VEI_VAR_TYPE_UINT32	Беззнаковое целое 32-битное число	0b0001 (0x01)
MIC_J_VEI_VAR_TYPE_INT32	Знаковое целое 32-битное число	0b0011 (0x03)
MIC_J_VEI_VAR_TYPE_DOUBLE_LSB	Вещественное число двойной точности (младшие 32 бита)	0b0100 (0x04)
MIC_J_VEI_VAR_TYPE_DOUBLE_MSB	Вещественное число двойной точности (старшие 32 бита)	0b1100 (0x0C)
MIC_J_VEI_VAR_TYPE_INT64_LSB	Знаковое целое 64-битное число (младшие 32 бита)	0b0111 (0x07)
MIC_J_VEI_VAR_TYPE_INT64_MSB	Знаковое целое 64-битное число (старшие 32 бита)	0b1111 (0x0F)
MIC_J_VEI_VAR_TYPE_COMPOSITE_INT32	Целая часть составного 64-битного числа (знаковое целое 32-битное число)	0b1011 (0x0B)
MIC_J_VEI_VAR_TYPE_COMPOSITE_FLOAT	Дробная часть составного 64-битного числа (вещественное число одинарной точности, 32 бита)	0x1000 (0x08)
MIC_J_VEI_NETWORK_BATCH_READ	Признак группового чтения по связи (OPC)	0x20
MIC_J_VEI_NETWORK_WRITE	Признак разрешения модификации переменной по связи	0x80

1.12 mic_j_record_save_mode_t – режимы записи (uint8_t)

Наименование	Описание	Значение	Версия ОС				
			1	2	3	4	5
MIC_J_RSM_NORMAL	Сохранение в нормальном режиме (в записи находятся значения переменных с флагом MIC_J_SR_MODE_N)	0x00	+	+	+	+	+
MIC_J_RSM_SETUP_SUPERUSER	Вход в меню настроек в режиме изготовителя	0x01	+	+		+	+
MIC_J_RSM_OPERATOR	Вход в меню настроек в режиме оператора	0x02	+	+			
MIC_J_RSM_SETUP_ENGINEER	Вход в меню настроек в режиме инженера	0x03	+	+		+	+
MIC_J_RSM_MFR_AGENT	Вход в меню настроек в режиме представителя изготовителя	0x04	+	+			
MIC_J_RSM_POWER_ON	Включение питания без восстановления значений переменных	0x05	+	+		+	+
MIC_J_RSM_POWER_ON_RESTORE	Включение питания с восстановлением переменных из журнала (в записи нет переменных)	0x06	+	+		+	+
MIC_J_RSM_POWER_OFF	Выключение питания (в записи находятся значения переменных с флагом MIC_J_SR_MODE_E)	0x07	+	+	+	+	+
MIC_J_RSM_POWER_ON_RESTORE_1	Включение питания с восстановлением из оперативного (резервного) журнала (в записи нет переменных)	0x08	+	+			
MIC_J_RSM_REGISTER_EVENT	Регистрация события (в записи находятся значения переменных с флагом MIC_J_SR_MODE_E)	0x09					+
MIC_J_RSM_CONTROL	Режим «Контроль» (в записи находятся значения переменных с флагом MIC_J_SR_MODE_E)	0x09	+	+			
MIC_J_RSM_RTC_SET_AUTO	Автокоррекция часов реального времени в случае сбоя	0x0A	+	+			
MIC_J_RSM_RTC_SET_NETWORK	Корректировка часов реального времени по сети	0x0B	+	+	+	+	+
MIC_J_RSM_RTC_SET_MENU	Корректировка часов реального времени из меню	0x0C	+	+	+	+	+
MIC_J_RSM_RTU_SETUP	Настройка УСО	0x0D		+		+	+
MIC_J_RSM_RTU_NOT_FOUND	УСО не найдено	0x10		+		+	+
MIC_J_RSM_RTU_NOT_ALL	Нет по крайней мере одного УСО	0x11		+			
MIC_J_RSM_RTU_NOT_RESPONDING	УСО не отвечает	0x12		+			
MIC_J_RSM_RTU_NETWORK_ERROR	Ошибка связи с УСО (ошибка CRC и др.)	0x13		+			
MIC_J_RSM_RTU_INTR_ERROR	Ошибка связи с УСО (потеря прерывания передатчика)	0x14		+			
MIC_J_RSM_CALC_STOP	Останов расчёта	0x15				+	+
MIC_J_RSM_CALC_START	Запуск расчёта	0x16				+	+
MIC_J_RSM_JOURNAL_ERASE	Стирание журнала из меню (в записи находятся значения переменных с флагом MIC_J_SR_MODE_E)	0xFA		+	+	+	+
MIC_J_RSM_ERASURE_NETWORK	Стирание журнала по сети (в записи находятся значения переменных с флагом MIC_J_SR_MODE_E)	0xFB		+			

Обозначения версий ОС:

- 1 – БКТ.М (MICOS)
- 2 – МИКОНТ-186 (MICOS)
- 3 – SMART OS версии < 1.18.11
- 4 – SMART OS версии ≥ 1.18.11 и < 1.19.1
- 5 – SMART OS версии ≥ 1.19.1

1.13 mic_j_config_mods_t – флаги изменения конфигурации (uint8_t)

Наименование	Описание	Значение
MIC_J_CONFIG_MOD_COMMON	Общий признак изменения параметров	0x01
MIC_J_CONFIG_MOD_NETWORK	Изменение параметров по связи	0x02
MIC_J_CONFIG_MOD_DATETIME	Дата/время	0x04
MIC_J_CONFIG_MOD_SENSOR	Параметры связи «канал-датчик» (параметры каналов измерения)	0x08
MIC_J_CONFIG_MOD_TABLE	Таблицы описания объекта	0x10
MIC_J_CONFIG_MOD_CALC	Параметры расчета (изменение типа, значения переменной или константы, состав переменных для сохранения в журнале и т. д.)	0x20
MIC_J_CONFIG_MOD_ERASE	Предупреждение о необходимости стирания журнала	0x40
MIC_J_CONFIG_MOD_RESERVED	Зарезервировано	0x80

1.14 mic_j_record_ext_info_flags_t – флаги дополнительной информации о записи (uint8_t)

Наименование	Описание	Значение
MIC_J_REI_MFR_KEY_SET	Установлен ключ изготовителя	0x01
MIC_J_REI_RECORD_TYPE_MASK	Маска типа записи	0x60
MIC_J_REI_RECORD_TYPE_OLD	Старый тип записи	0x00
MIC_J_REI_RECORD_TYPE_EXT	Расширенный тип записи с упакованной датой/временем	0x20
MIC_J_REI_RECORD_TYPE_EXT_RECNO	Расширенный тип записи с логическим номером	0x40
MIC_J_REI_RECORD_TYPE_EXT_RSRVD	Зарезервировано	0x60
MIC_J_REI_BAD_RECORD	Признак неисправной (некорректной) записи	0x80

2 Анализ журнала

2.1 Анализ заголовка журнала `mic_j_header_t`

2.1.1 Проверка корректности

Условия некорректности проверяются последовательно. Если условие выполняется, то дальнейший анализ не производится, весь журнал считается некорректным.

Условие некорректности	Сообщение
<code>header_size < 512</code>	Размер заголовка < 512 байт
<code>(header_size % 512) != 0</code>	Размер заголовка журнала не кратен 512 байтам
<code>signature != 0x4D41</code>	Неверная сигнатура журнала
<code>(signature_ext - 0x4D41) != datetime_last_get_info.second</code>	Неверное количество секунд в сигнатуре
<code>record_size < 16</code>	Размер записи должен быть ≥ 16 байт
<code>(net_address == 0xFF) && (header_type != MIC_J_HEADER_TYPE_EXT)</code>	Неизвестный тип журнала <code>header_type</code>
Достигнут конец заголовка журнала прежде, чем прочитано <code>string_count</code> строк и <code>var_count_base + var_count_ext</code> описаний переменных	Неожиданный конец заголовка журнала

2.1.2 Предупреждения

Если выполняется условие предупреждения, то оно каким-либо образом регистрируется и производится соответствующее действие. Производится дальнейший анализ журнала.

Условие предупреждения	Сообщение	Действие
<code>(net_address == 0xFF) && (codepage > MIC_J_CODEPAGE_MAX)</code>	Неизвестная кодировка	Использовать кодировку Windows-1251
По крайней мере одно из значений <code>datetime_first_power_on</code> , <code>datetime_last_power_on</code> , <code>datetime_last_setup</code> , <code>datetime_last_get_info</code> содержит некорректную дату или время	Поле [имя поля] содержит некорректную дату.	Установить признак некорректного значения
<code>var_count_base == 255</code>	Количество переменных должно быть < 255	—
<code>var_count_ext == 255</code>	Количество дополнительных переменных должно быть < 255	—

2.1.3 Действия по умолчанию

Условие использования	Действие
<code>net_address < 0xFF</code>	Использовать кодировку OEM 866
<code>(record_count == 0) (record_count == 65535)</code>	Рассчитать <code>record_count = (journal_size_in_sectors * 512 - header_size) / record_size</code>

2.2 Анализ записей `mic_j_record_t`

2.2.1 Проверка корректности

Условия некорректности проверяются последовательно. Если условие выполняется, то дальнейший анализ записи не производится и устанавливается признак некорректной записи.

Условие некорректности	Сообщение
<code>ext_info_flags & MIC_J_REI_BAD_RECORD</code> (за исключением записей, полностью состоящих из значений 0xff)	Некорректная запись
Значение <code>datetime</code> содержит некорректную дату или время	Некорректная дата
<code>(ext_info_flags & MIC_J_REI_RECORD_TYPE_MASK) == MIC_J_REI_RECORD_TYPE_EXT_RSRVD</code>	Неизвестный тип записи
Неизвестное значение <code>save_mode</code>	Неизвестный режим записи
Неверное количество переменных (см. пункт 2.2.3)	Неверное количество переменных

2.2.2 Порядок записей

Если используется расширенный тип записи с логическим номером (`ext_info_flags & MIC_J_REI_RECORD_TYPE_MASK == MIC_J_REI_RECORD_TYPE_EXT_RECNO`), то порядок определяется сначала по полю `record_no`, а затем по полю `datetime`; иначе только по полю `datetime`.

2.2.3 Количество переменных

Если в поле `var_count` находится ненулевое значение, то, в зависимости от режима записи, оно должно быть меньше или равно количеству переменных в заголовке журнала, у которых установлен соответствующий флаг `mic_j_sr_mode_t`:

Режим записи <code>save_mode</code>	Количество переменных в заголовке журнала с выполненным условием
<code>MIC_J_RSM_NORMAL</code>	<code>header.var_desc[i].sr_modes & MIC_J_SR_MODE_N</code>
<code>MIC_J_RSM_POWER_OFF</code>	<code>header.var_desc[i].sr_modes & MIC_J_SR_MODE_E</code>
<code>MIC_J_RSM_REGISTER_EVENT</code>	<code>header.var_desc[i].sr_modes & MIC_J_SR_MODE_E</code>
<code>MIC_J_RSM_JOURNAL_ERASE</code>	<code>header.var_desc[i].sr_modes & MIC_J_SR_MODE_E</code>

Если режим записи отличается от вышеперечисленных или значение `var_count` больше указанного, то такую запись считать некорректной.

Порядок переменных `var[i]` в записи соответствует порядку переменных `var_desc[i]` в заголовке журнала.