

Micrium, Inc.

Словарь акронимов, сокращений и мнемоник

Application Note AN-2001

Jean J. Labrosse
Jean.Labrosse@Micrium.com
www.Micrium.com

Acronyms, Abbreviations and Mnemonics Dictionary Application Note AN-2001

Перевод: Андрей Шлеенков
<http://andromega.narod.ru>
<mailto:andromega@narod.ru>

Содержание

1.00 Введение	3
2.00 Создание идентификаторов	4
3.00 Словарь	5
Ссылки	12
Контакты	12

1.00 Введение

При создании имен переменных и функций (идентификаторов) часто практикуется использование акронимов (например, OS, ISR, TCB и т.п.), сокращений (`buf`, `doc` и т.п.) и мнемоник (`clr`, `cmp`, и т.п.). Использование акронимов, сокращений и мнемоник (далее аббревиатур) позволяет идентификатору быть описательным, требуя меньшее количество символов. К сожалению, если аббревиатуры используются непоследовательно, они могут только увеличивать беспорядок. Чтобы гарантировать непротиворечивость, мы создали список аббревиатур, который используем во всех наших проектах. Те же самые аббревиатуры используются везде, как только были определены. Я называю этот список *Словарь акронимов, сокращений и мнемоник*. Список для Micrium показан в таблице раздела "3.00 Словарь". Как только нам понадобится большее количество аббревиатур, мы пересмотрим этот документ.

Могут быть случаи, когда в ваших продуктах или проектах наш список не будет иметь смысла. Например, если вы являетесь фирмой, разрабатывающей проекты для различных, полностью не связанных между собой клиентов и продуктов, то для каждого проекта было бы более подходящим иметь различные списки – словарь для сельского хозяйства не может быть таким же, как словарь для оборонной промышленности. Мы используем следующее правило – если некоторые продукты подобны, они используют тот же самый словарь.

Общий словарь для рабочей группы проекта также увеличивает ее производительность. Важно, чтобы однозначность поддерживалась во всем проекте независимо от индивидуальности программистов. Если принято, что `buf` означает *buffer*, это должно использоваться всеми членами проекта, вместо того чтобы использовать индивидуальные идентификаторы – один использовал бы `buffer`, а другой использовал бы `bfr`. Следуя этой концепции, вы должны всегда использовать `buf`, даже если ваш идентификатор может вместить полное имя. Придерживайтесь `buf`, даже если вы можете полностью записать слово `buffer`.

2.00 Создание идентификаторов

Определения констант, макросов, переменных и имен функций (то есть идентификаторов) должны использовать имя файла в качестве префикса. Этот префикс облегчает поиск идентификатора в среде большого проекта, а также помогает узнать, где эти идентификаторы объявлены и кому 'принадлежат'. Например, функции в файле с именем KBD.C и функции в файле с именем VIDEO.C могут быть объявлены следующим образом:

KBD.C

```
KbdGetChar()  
KbdGetLine()  
KbdGetFnctKey()
```

VIDEO.C

```
VideoGetAttrib()  
VideoPutChar()  
VideoPutStr()  
VideoSetAttrib()
```

Нет необходимости использовать полное имя файла или модуля как префикс. Например, файл по имени KEYBOARD.C мог бы иметь функции, начинающиеся с Kbd вместо Keyboard.¹ Также для отделения слов в идентификаторе предпочтительней использовать символы верхнего регистра, чем знаки подчеркивания. Подчеркивания не добавляют какой-либо смысл именам, но отнимают место для символов.

Насколько возможно, вместе с аббревиатурами используйте формат 'module-object-operation'. При создании идентификаторов сначала определите имя модуля (или подсистемы), продолжите именем объекта и затем именем операции как показано ниже.

```
OSSemPost()  
OSSemPend()
```

Здесь: имя модуля – OS (Operating System), объект – Sem (Semaphore) а операция, которая может выполняться над объектом – Post или Pend.

Положительным качеством этой методики является то, что она позволяет вам группировать подобные элементы вместе. Конечно, может быть, что в начале к этому не очень легко привыкнуть, потому что более естественно думать в терминах Action-Object типа OSPostSem(). Мы предпочитаем метод Object-Action, потому что он группирует подобные объекты с их действиями. Например:

```
OSSemAccept()  
OSSemCreate()  
OSSemDel()  
OSSemPost()  
OSSemPend()  
OSSemQuery()
```

Недавно мы начали добавлять символ подчеркивания после имени модуля² для отделения модуля от сочетания Object-Action (См. документ AN-2000). Используя вышеупомянутые имена функций как примеры, теперь они могут выглядеть следующим образом:

```
OS_SemAccept()  
OS_SemCreate()  
OS_SemDel()  
OS_SemPost()  
OS_SemPend()  
OS_SemQuery()
```

¹ Примечание переводчика: При создании аббревиатур полезно помнить, что, например, правила русского языка не разрешают сокращение слова менее чем на 2 буквы.

² Примечание переводчика: Использование в имени функции префикса, полученного из имени модуля, а также знака подчеркивания может быть предпочтительным для глобально видимых функций проекта.

3.00 Словарь

Следующая таблица представляет текущий словарь аббревиатур Micrium. Список может быть изменен, но наиболее вероятно только добавление имен. Как только аббревиатуры определены, они не должны подвергаться изменениям.

Таблица: словарь аббревиатур

Термин	Аббревиатура
Absolute	Abs
Action	Act
Active	Active
Add	Add
Address	Addr
Address Resolution Protocol	ARP
Adjust	Adj
Alarm	Alm
Alarming	Alarming
Application	App
Analog Input	AI
Analog Output	AO
Analog to Digital Converter	ADC
Assign	Assign
Assignment	Assign
Attributes	Attrib
Automatic	Auto
Auxiliary	Aux
Average	Avg
Background	Bgnd
Bank	Bank
Binary	Bin
Bit	Bit
Buffer	Buf
Cache	Cache
Calculate	Calc
Calculation(s)	Calc
Calibration	Cal
Change	Chng
Channel	Ch
Character	Chr
Check	Chk
Clear	Clr
Clock	Clk
Column	Col
Command	Cmd
Communications	Comm
Compare	Cmp
Complement	Cpl

Термин	Аббревиатура
Computation	Comp
Compute	Comp
Condition	Cond
Configuration	Cfg
Configure	Cfg
Consecutive	Consec
Constant	Const
Control	Ctrl
Control Byte	CB
Control Word	CW
Convention	Conv
Conversion	Conv
Count(s)	Cnt(s)
Counter	Ctr
Create	Create
Creation	Create
Current	Cur
Date	Date
Day	Day
Debug	Dbg
Decrement	Dec
Default	Dflt
Defined	Def
Definition	Def
Delay	Dly
Delete	Del
Denominator	Denom
Depth	Depth
Destination	Dest
Detect	Detect
Development	Dev
Device	Dev
Digit	Dig
Digital	Dig
Digital to Analog Converter	DAC
Direction	Dir
Directory	Dir
Disable	Dis or DISABLE
Disabled	Dis or DISABLED
Discharge	Disc
Disk	Disk
Display	Disp
Divider	Div
Divisor	Div
Driver	Drv
Down	Down

Термин	Аббревиатура
Duration	Dur
Dynamic	Dyn
Edge	Edge
Efficiency	Eff
Elapsed	Elapsed
Enable	En or ENABLE
Enabled	En or ENABLED
Engineering Units	EU
Entry	Entry
Error	Err
Ethernet	Ether
Exception	Except
Extension	Ext
Extern	Ext
External	Ext
Family	Fam
Fault	Flt
Find	Find
Flag	Flag
Flow	Flow
Flush	Flush
Flywheel	Fly
Force	Force
Forced	Forced
Foreground	Fgnd
Fraction	Fract
Frame	Frame
Frequency	Freq
Function	Fnct
Ground	Gnd
Group	Grp
Hardware	Hard
Header	Hdr
Hour(s)	Hr
I.D. #	ID
Identifier	ID
Inactive	Inact
Increment	Inc
Index	Ix
Initialize	Init
Input	In
Input/Output	IO
Interface	IF
Internet Control Message Protocol	ICMP
Internet Protocol	IP
Interrupt	Int

Термин	Аббревиатура
Interrupt Service Routine	ISR
Jump	Jump
Key	Key
Keyboard	Kbd
Lamp	Lamp
Last	Last
Latch	Latch
Liquid Crystal Display	LCD
Light Emitting Diode	LED
Left	Left
Length	Len
Level	Level
Limit	Lim
Linear	Line
Linear	Lin
Link	Link
List	List
Local	Local
Location	Local
Manual	Man
Map	Map
Mask	Msk
Master	Master
Maximum	Max
Measurement	Meas
Memory	Mem
Message	Msg
Minimum	Min
Minute	Min
Mode	Mode
Module	Module
Modulo	Mod
Multiplier	Mult
Negative	Neg
Network	Net
Next	Next
Nominal	Nom
Number	Nbr
Numerator	Numer
Offset	Offset
Operating System	OS
Operation	Oper
Optimize	Opt
Optimizer	Opt
Option	Opt
Optional	Opt

Термин	Аббревиатура
Ordinal	Ord
Origin	Org
Output	Out
Over	Ovr
Overflow	Ovrf
Packet	Pkt
Parameter	Param
Password	Pswd
Pattern	Pattern
Point	Pt
Pointer	Ptr
Port (i.e. I/O)	Port
Positive	Pos
Point to Point Protocol	PPP
Power	Pwr
Previous	Prev
Priority	Prio
Procedure	Proc
Process	Process
Product	Prod
Protocol	Protocol
Protect	Prot
Public	Pub
Pulse Width	PW
Pulse Width Modulation	PWM
Qualified	Qual
Qualifier	Qual
Qualify	Qual
Quantity	Qty
Query	Query
Queue	Q
Quick	Quick
Range	Range
Reverse Address Resolution Protocol	RARP
Ratio	Ratio
Read	Rd
Ready	Rdy
Recall	Rcl
Receive	Rx
Receiver	Rx
Register	Reg
Relative	Rel
Request	Req
Reserved	Rsvd
Reset	Reset
Respond	Resp

Термин	Аббревиатура
Response	Resp
Restart	Restart
ReSynchronize	ReSync
Retard	Retard
Retries	Retries
Return	Rtn
Right	Right
Row	Row
Run	Run
Scale	Scale
Scaling	Scaling
Schedule	Sched
Scheduler	Sched
Screen	Scr
Second	Sec
Select	Sel
Semaphore	Sem
Send	Send
Sender	Sender
Sense	Sense
Sensitivity	Sens
Sensor	Sensor
Serial Line Interface Protocol	SLIP
Serial Number	SN
Service	Serv
Set	Set
Setpoint	Stp
Shutdown	Sd
Size	Size
Slave	Slave
Slope	Slope
Socket	Sock
Software	Soft
Source	Src
Specification	Spec
Speed	Spd
Stack	Stk
Standard	Std
Start	Start
State	State
State Machine	SM
Static	Static
Statistic(s)	Stat
Status	Status
Stop	Stop
Storage	Sto

Термин	Аббревиатура
Store	Sto
Symbol	Symbol
Synchronize	Sync
System	Sys
System Network Management Protocol	SNMP
Table	Tbl
Target	Target
Task	Task
Task Control Block	TCB
Test	Test
Text	Text
Time	Time
Time Base	TB
Timeout	Timeout or TO
Timer	Tmr
Time To Live	TTL
Timing	Timing
Transmit	Tx
Transmission Control Protocol	TCP
Underflow	Undf
User Datagram Protocol	UDP
Units	Units
Update	Update
Upload	Upload
User	User
Valid	Valid
Value(s)	Val
Variable(s)	Var
Vector	Vector
Verify	Verify
Version	Ver
Volume	Vol
Width	Width
Write	Wr

ССЫЛКИ

µC/OS-II, The Real-Time Kernel, 2nd Edition

Jean J. Labrosse
R&D Technical Books, 2002
ISBN 1-57820-103-9

Embedded Systems Building Blocks

Jean J. Labrosse
R&D Technical Books, 2000
ISBN 0-87930-604-1

КОНТАКТЫ

Micrium, Inc.
949 Crestview Circle
Weston, FL 33327
954-217-2036
954-217-2037 (FAX)
e-mail: Jean.Labrosse@Micrium.com
WEB: <http://www.Micrium.com>

R&D Books, Inc.
1601 W. 23rd St., Suite 200
Lawrence, KS 66046-9950
(785) 841-1631
(785) 841-2624 (FAX)
e-mail: rdorders@rdbooks.com
WEB: <http://www.rdbooks.com>