C " " : 72"

" "

"

C "	"	"	"

"

" "

0'	"	"	"

"

" " "

" "

"

"

" "

"

"

3 " " C " '0000000000

9 " " 5" " 6 '0000000000

32 / " " 5" " 6 '0000000000

36 '0000000000

3" " " C " "

308" " " "

" "	"†" "†" "

4" '"

40 8" / "

-
-
-

40 8 8" / " "

"	" "	" " "	"

5" "

503" " "

504" " "

5 6 " "C "

		"	
		C "	
		C "	
		44 C "	
		C "	
		" "	

		"	FALSE CallTrustedFunction CallNonTrustedFunction
		" "	
		C "	
		"	

5 0 5 0 3 " " /C " 5 " " 6 "

		" "	
		"	
		C	
		2 5 "	
		" "	

"

5@5@4" " /C " 5" " 6 "

		"	
		"	
		"	

" "

" "

506" "C "

		" "	

5060 3" " /C "

		"	
		37 9"	

5060 4" " /C "

		"	
		37 9"	
		2"	

5 0 7 " " " C " "

		"	
		C "	

5 0 7 0 8 " " /C " "

" " " C " "

		" "	

"

" " " "

		" "	
		" "	
		" "	

5Œ8" C "C "

		2 5 "	

5Œ8Œ8" C " "

/C " "		"	"
		"	
		"	

5 9 " ' "

5 " " " " 5 " " 6 "

0 "

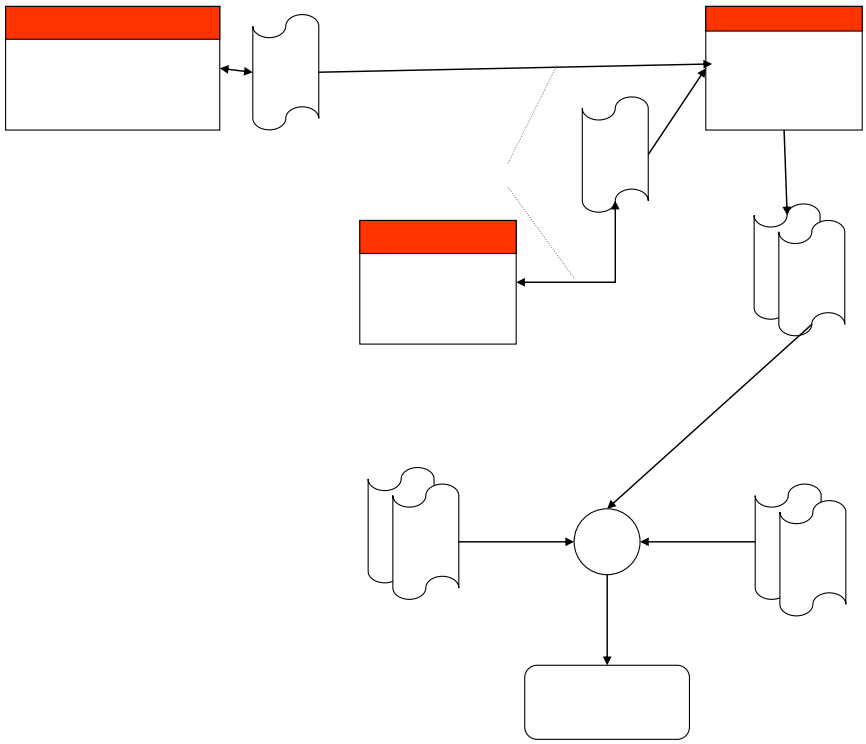
0 "

0 "

0 "

0 "

6" " "



6 0 8 " " "

-
-
-
-
-
-
-
-
-
-
-
-

-
-
-
-
-
-

-
-
-
-
-

OILFileNam
RTIDebugSupport
OILFileNam

7" " "

708" " "

704" " "

705" " "

706" " "

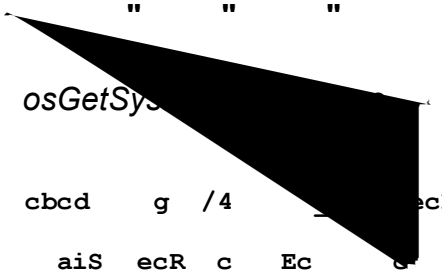
707" " " "

7070" " " "

osGetStackUsage

7070" " " "

osGetSy



cbcd g /4 _ ecR c9

_aiS _ecR c Ec G _aiS _ec& gb 9

7070" " " "

osGetISRStackUsage

cbcd g /4 _aiS _ecR c9

_aiS _ecR c Ec G P _aiS _ec&G PR c G G bc 9

8"

"

"

"

803"

- , C ac g Tca R_ c a V<
- , CGGLRTca R_ c a V<
- , c

8030"

8034"

8041" " " " "

8053" " "3"

80503" " " " "

For EIINT, you can use either of the following methods to declare a function as an interrupt routine:

- *Place #pragma ghs interrupt immediately before the function.*
- *Prepend the __interrupt keyword to the function definition.*

80504" " " "

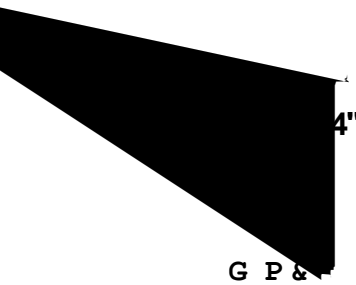
8 6 " " "4"

8 6 0 3" " "

3" " 5 "
EnableNesting

6 "
EnableNesting

8 6 0 4" " "



4" " "

G P & G P

-(G P d a g b (-

8 0 7" " "

" "

"

SECONDSPERTICK

not

" "

9" " " 5" " 6 "

903" " " "

" "	" " " " " " " "

9030" " "2"

9030" " " "

9@8@5" " " "

-
-

-
-
-

: " : 72" "

: 08" " " "

: 04" " " " "

: 05" "

" " "

03" C " "

0303" C "
3" " 5 " *DisableAllInterrupts*

6 " *DisableAllInterrupts*

0304" C "
3" " 5 " *EnableAllInterrupts*
DisableAllInterrupts
6 " *EnableAllInterrupts*
DisableAllInterrupts

0305" C "
3" " 5 " *SuspendAllInterrupts*

6 " *SuspendAllInterrupts*

0306" C "
3" " 5 " *ResumeAllInterrupts*
SuspendAllInterrupts
6 " *ResumeAllInterrupts*
SuspendAllInterrupts

0307" "
SuspendOSInterrupts

0308" "

ResumeOSInterrupts
SuspendOSInterrupts

" "

"

" "

0309" "

0310 " "

0310 " C D "

0312" "

0313" "

-
-
-
-
-
-

04" " "C "

04030B" " "

"
osuint8 osReadPeripheral8 (osuint16 area, osuint32 address) osuint16 osReadPeripheral16(osuint16 area, osuint32 address) osuint32 osReadPeripheral32(osuint16 area, osuint32 address)
"
area
address
" "
" "
" "   " "
" " "
"
" " "

04030401" " "

"
void osWritePeripheral8 (osuint16 area, osuint32 address, osuint8 value) void osWritePeripheral16(osuint16 area, osuint32 address, osuint16 value) void osWritePeripheral32(osuint16 area, osuint32 address, osuint32 value)
"

area	
address	
value	

" "
None

" "
" "   " "

" " "
"
" " "

04036" " "

"
void osModifyPeripheral8 (osuint16 area, osuint32 address, osuint8 clearmask, osuint8 setmask) void osModifyPeripheral16(osuint16 area, osuint32 address, osuint16 clearmask, osuint16 setmask) void osModifyPeripheral32(osuint16 area, osuint32 address, osuint32 clearmask, osuint32 setmask)

"	
area	
address	
clearmask	
setmask	

" "
None

" "
"   " " "

" " "
"
" " "

0 5" " 'C " " 5" " 6 "

```
!bcdg c   Pc_bGAP6&_bb     &( &&   g  6(  &_bb
!bcdg c   Pc_bGAP/4&_bb    &( &&   g  /4(  &_bb
```

" "	C " "
B/ -B/K	DDDC C .. + DDDC C 1C &CGA. CGA1/ DDDD .2. + DDDD /DC &CGA10 CGA033
C/ -C/K	DDDC C .. + DDDC C 1C &CGA. CGA1/ DDDD .2. + DDDD 1DC &CGA10 CGA3//
C/ +Daa0	DDDC C .. + DDDC C 1C &CGA. CGA1/ DDDD .2. + DDDD 1DC &CGA10 CGA3//
D/F	DDDC C .. + DDDC C 1C &CGA. CGA1/ DDDD .2. + DDDD 0 A &CGA10 CGA13.
D/	DDDD 7... + DDDD 7.1C &CGA.+ CGA1/ DDDD .2. + DDDD 010 &CGA10+ CGA06/
D/K	DDDC C .. + DDDC C 1C &CGA. CGA1/ DDDD .2. + DDDD 03A &CGA10 CGA075
N/K	DDDC C .. + DDDC C 1C &CGA. CGA1/ DDDD .2. + DDDD 0DC &CGA10 CGA161

05 08" " " " " "

:"

"
gb U g cGAP6& g 10 _bb * g 6 _ 9
"
_bb _
" "
gb
" "
(& g 6(_bb ; & g 6 _ 9

"

38"

"
gb U g cGAP/4& g 10 _bb * g /4 _ 9
"
_bb _
" "
gb
" "
(& g /4(_bb ; & g /4 _ 9

"

"

05 04" " " " " " "

"

"
gb c GAPK_ i & g 10 _bb 9
"
_bb
" "
gb
" "
(& g 6(_bb ; & g 6 . 6. 9

"

"
gb A c_ GAPK_ i & g 10 _bb 9
"
_bb
" "
gb
" "
(& g 6(_bb ; & g 6 . 5D9

GG" " " " " " " "

"

"

| "

05 6" ." " " " "D " " " " "

"

"
gb Ec GKI CG& g /4 g bc 9
"
g bc
" "
g 6
" "
c & g 6 &GKI CGKI g bc < 9

"

"
gb c GKI CG& g /4 g bc 9
"
g bc
" "
gb
" "
GKI CGKI g bc < ; /9

"

"
gb A c_ GKI CG& g /4 g bc 9
"
g bc
" "
gb
" "
GKI CGKI g bc < ; .9

05 7" " " " "

:"

"
gb U g cGKP6& g 10 _bb * g 6 _ 9
"
_bb _
" "
gb
" "
(& g 6(_bb ; & g 6 _ 9

38"

"
gb U g cGKP/4& g 10 _bb * g /4 _ 9
"
_bb _
" "
gb
" "
(& g /4(_bb ; & g /4 _ 9

54"

"
gb U g cGKP10& g 10 _bb * g 10 _ 9
"
_bb _
" "
gb
" "
(& g 10(_bb ; & g 10 _ 9

6" " "

6.8" "

6.4" "

6.5" "

6.6" "

6.7" "

6.8" C " 3" "
“osTimerInterrupt”

```
void PreAlarmHook(void)
{
   /* user specific code */
}
```

0 7" " " " " "

0 7 0 8" " C "

"
g 6 AfcaiKNS aac &a g 6(_bb
"
_bb
" "
g 6
" "
• • • • •
" " "
"

0704" " C "

"
_ R c Afcai bPcd c fKNS & gb
"
/ /"
" "
_ R c
" "
• • • •
" " "
"
▪ ▪

Œ" " " " " "

Œ Œ" " C "

"
_ R c Afcai bPcd c fRg c & gb
"
/ /"
" "
_ R c
" "
• • • •
" " "
"
▪ ▪

32" / " " 5" " 6 "

3203" "

3204" C "

"
StatusType osCallNonTrustedFunction(NonTrustedFunctionIndexType FunctionIndex, NonTrustedFunctionParameterRefType FunctionParams);
"
" "
E_OK
E_OS_SERVICEID
" "
void NONTRUSTED_<name of the non-trusted function(NonTrustedFunctionIndexType, NonTrustedFunctionParameterRefType);
" " "
" " "
"

" "

"

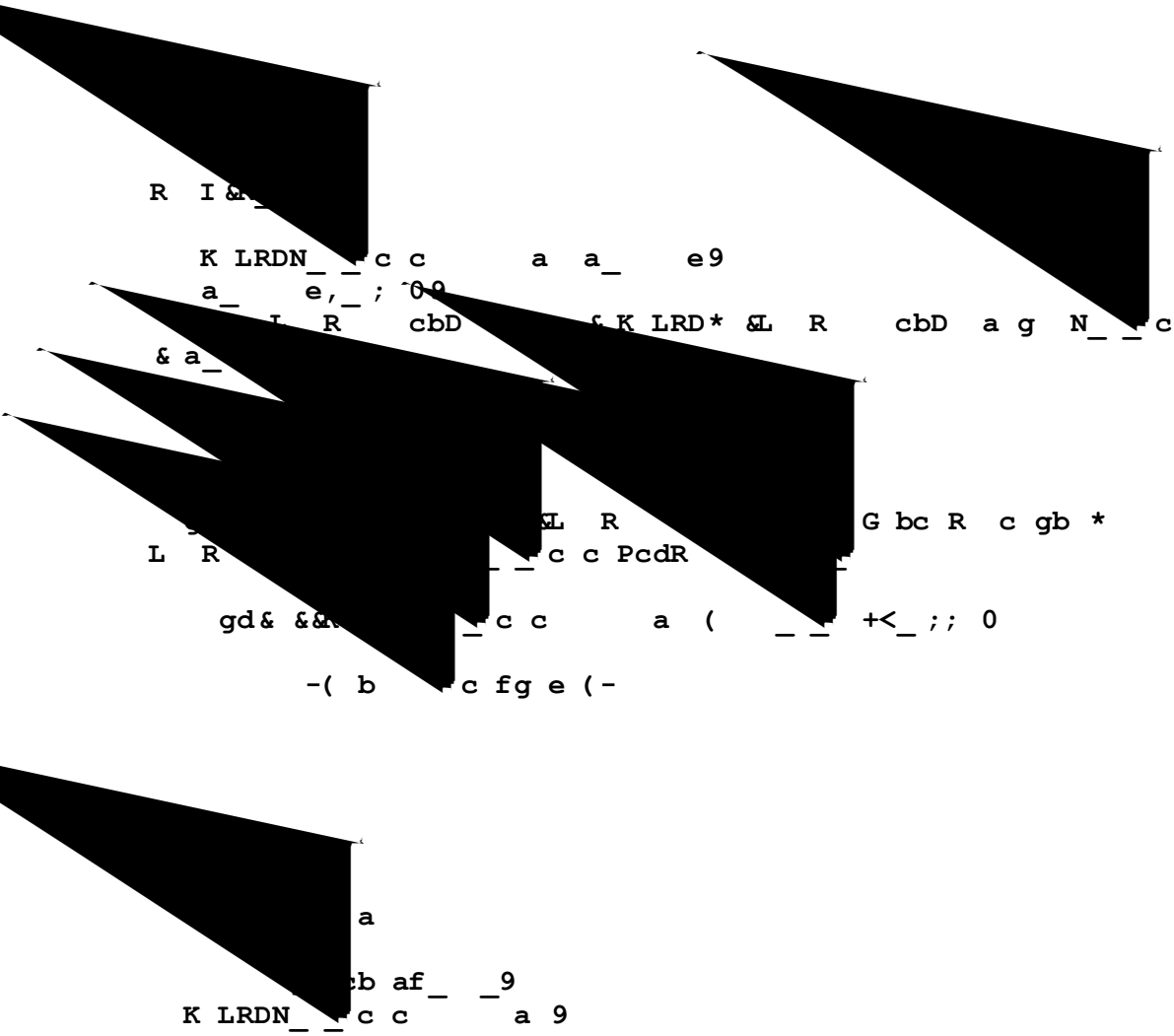
" "

3205" " "

□□

C " "	" "	C " "	" "

32 50 3" "



33" "

3303" "

330303" " "

GetCoreID().

"		

3304" / " / "

osInitMultiCoreOS() osInitSlaveCore()

osInitMultiCoreOS()
osInitSlaveCore()
StartCore()
StartNonAutosarCore()
osInitMultiCoreOS() osInitSlaveCore()

" "

"
M RK. AKN
M RK. AKN

" "

33040 3" D " " " " "

```
  gb main()
{
   […]
     g af(GetCoreID())
   {
        a_ c OS_CORE_ID_0:
             […]
             osInitMultiCoreOS();
             StartCore(OS_CORE_ID_1);
             StartOS(OSDEFAULTAPPMODE);
               c_i;
        a_ c OS_CORE_ID_1:
             […]
             osInitSlaveCore();
             StartOS(OSDEFAULTAPPMODE);
               c_i;
        bcd_   :
             […]
               c_i;
   }
   […]
}
```

33040 4" " 3" " " "

```
  gb main()
{
   […]
     g af(GetCoreID())
   {
        a_ c OS_CORE_ID_0:
             […]
             osInitMultiCoreOS();
             StartNonAutosarCore(OS_CORE_ID_1);   /* may be called later */
             StartOS(OSDEFAULTAPPMODE);
               c_i;
        a_ c OS_CORE_ID_1:
             […]
             osInitSlaveCore();
             […]
               c_i;
        bcd_   :
             […]
               c_i;
   }
   […]
}
```

3304 5" " 4" " " "

```
  gb main()
{
   […]
     g af(GetCoreID())
   {
        a_ c OS_CORE_ID_0:
             […]
             osInitMultiCoreOS();
             StartCore(OS_CORE_ID_1);
             […]
               c_i;
        a_ c OS_CORE_ID_1:
             […]
             osInitSlaveCore();
             osInitMultiCoreOS();
             StartNonAutosarCore(OS_CORE_ID_0);    /* adjusts the state of PE1*/
             StartOS(OSDEFAULTAPPMODE);
               c_i;
        bcd_   :
             […]
               c_i;
   }
   […]
}
```

34" " " 6 "

3403" 'C "

: 72" " 6" "

TimingProtectionTimerClock

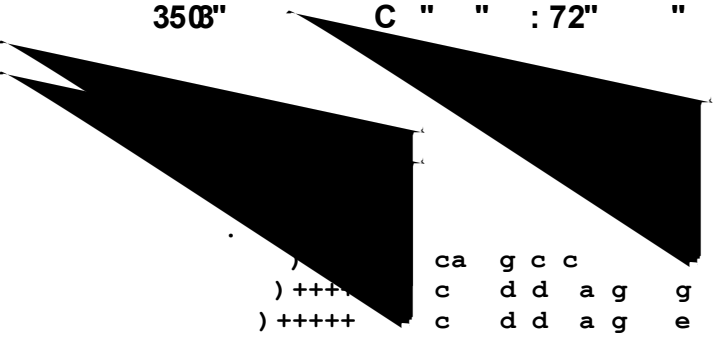
TimingProtectionTimerClock = 20000

3404" " " 6" "

-
-
-

35" " "

3503" C " " : 72" " "



ca g c c c
c d d a g g fc d a g e
c d d a g e

3503030" : 72" " " "

"	"	"	"

"	"	"	"

36" "

360B" " "

" "	"	3"	5"	6"
		"	"	"
			"	"
		"3"	"	"

[1] Only for multi core systems

360 4" " "

" "	"	3"	5"	6"
		"	"	"
		"	"	"
		"	"	"
		"	"	"

37" "

-
-
-
-
-
-