

## Re: LCD\_Backlight

---

*Source:*

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2006-04/msg00694.html>

---

- *From:* Anurag <[Anurag@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Anurag@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 24 Apr 2006 04:06:01 -0700
- 

Hello voidcode please omit the last two lines and yes I doubt if it would make a difference if i use system activity event instead of useractivity in this case

"Anurag" wrote:

Hello voidcoder,

I am using the standard wince5.0 full power management. It is in the PM module (in Public root) it reads the name and timeout from the registry and is handling the same I am accessing the same events in my backlight driver. Also the document on "Activity timers" says that the PM module opens 3 events :  
PowerManager/ActivityTimer/SystemActivity (Auto reset event)  
PowerManager/SystemActivity\_Active, and  
PowerManager/SystemActivity\_Inactive (both manual reset event)  
The document also states  
Any number of threads may open handles to the manual-reset activity/inactivity events and wait on them to determine the status of the OS. For example, a backlight driver may use the pair of manual-reset events created for an activity timer to decide how to control backlight power and intensity.  
This is the reason why I am waiting on the two events. Now as per my requirement when i touch the screen or on some other activity if i want the back light to be ON I need to set this event ie tell the system that The system is active and keep the backlight on for another period of time. I am doing the same via SetEvent() which does not seem to set my activity event. I went through your code and am unable to understand which of this is a auto reset and which is a manual reset event. In manual reset i need to manually reset the event (which i am trying to do in my backlight driver). I fail to understand that if my thread is waiting for the active event and i set it through the touch why does'nt it reflect??

Now this is the reason why I have open handles to these events and am waiting on them

"voidcoder" wrote:

I do not understand why do you need TWO thread to wait on ativity timers. One is just enough, see my example. Also not clear why do you need to reset events in your touch driver? This is done automatically as part of windows input handling.

Btw why do you need SystemActivity? Isn't the !UserActivity! what are you looking for?

"Anurag" <Anurag@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message [news:49CEEE5B-3009-4372-9DC2-32CD4D485BDE@xxxxxxxxxxxxxxxxxxxx](mailto:news:49CEEE5B-3009-4372-9DC2-32CD4D485BDE@xxxxxxxxxxxxxxxxxxxx)

Hello voidcoder,

This are my registry settings:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power\ActivityTimers\Sy
"Timeout"=dword:A
```

In my backlight driver Init function I create two events with thier names

the return value is the handle to the existing events:

```
if((SysAct_Active = CreateEvent(NULL, FALSE, FALSE, _T("PowerManager/SystemActivity_Active")) != NULL)
```

```
{
if(ERROR_ALREADY_EXISTS == GetLastError())
NKDbgPrintfW(TEXT(" ERROR_ALREADY_EXISTS
1\r\n"));
else
NKDbgPrintfW(TEXT(" Event created 1 %d
\r\n"),SysAct_Active);
```

```
}
else
NKDbgPrintfW(TEXT("Event not created 1\r\n"));
```

```
if((SysAct_Inactive = CreateEvent(NULL, FALSE, FALSE, _T("PowerManager/SystemActivity_Inactive"))) != NULL)
```

```
{
if(ERROR_ALREADY_EXISTS == GetLastError())
NKDbgPrintfW(TEXT(" ERROR_ALREADY_EXISTS
2\r\n"));
else
NKDbgPrintfW(TEXT(" Event created 2
%d\r\n"),SysAct_Inactive);
```

Re: LCD\_Backlight

```
}  
else  
NKDbgPrintfW(TEXT("Event not created 2 %d\r\n"));  
  
// i create two threads  
hThread1 = CreateThread( NULL, 0, System_Active, 0, 0,  
NULL); // thread  
created as active  
hThread2 = CreateThread( NULL, 0, System_Inactive, 0,  
CREATE_SUSPENDED,  
NULL); // thread  
//created in suspended state  
NKDbgPrintfW(TEXT(" Both Threads created\r\n"));
```

the two threads are as follows:

```
static ULONG  
System_Active(  
PVOID Reserved  
)  
{  
  
WaitForSingleObject( SysAct_Active, INFINITE );  
NKDbgPrintfW(TEXT(" out of  
WaitForSingleObject:SystemActivity_Active\r\n"));  
Signal_BKL(1); //control back_light  
if(ResetEvent(SysAct_Active))  
NKDbgPrintfW(TEXT("ASA : SysAct_Active Event  
reset\r\n"));  
else  
NKDbgPrintfW(TEXT("ASA : SysAct_Active Event could  
not reset\r\n"));  
  
ResumeThread(hThread2);  
SuspendThread(hThread1);  
return(TRUE);  
}
```

```
static ULONG  
System_Inactive(  
PVOID Reserved  
)  
{  
WaitForSingleObject( SysAct_Inactive, INFINITE );  
NKDbgPrintfW(TEXT(" out of  
WaitForSingleObject:SystemInactivity_Active  
\r\n"));  
Signal_BKL(0); //control back_light  
if(ResetEvent(SysAct_Inactive))
```

Re: LCD\_Backlight

```
NKDbgPrintfW(TEXT("ASA : SysAct_Inactive Event
reset\r\n"));
else
NKDbgPrintfW(TEXT("ASA : SysAct_Inactive Event could
not reset\r\n"));

ResumeThread(hThread1);
SuspendThread(hThread2);
return(TRUE);
}
```

In the touch driver MDD main.c I first open the handle to the event

```
if((SysAct_Reset =
OpenEvent(EVENT_ALL_ACCESS,FALSE,_T("PowerManager/SystemActivity_Active")))
NULL)
{
if(ERROR_ALREADY_EXISTS == GetLastError())
NKDbgPrintfW(TEXT(" ERROR_ALREADY_EXISTS
3\r\n"));
else
NKDbgPrintfW(TEXT("Reset Event created again %d
\r\n"),SysAct_Reset);
}
else
NKDbgPrintfW(TEXT("Event not created 1\r\n"));
```

then in the touch IST i try to set the event

```
if(SetEvent(SysAct_Reset))
NKDbgPrintfW(TEXT("ASA:Reset event set\r\n"));
else
NKDbgPrintfW(TEXT("ASA:Reset event NOT set\r\n"));
```

when i touch the screen it does not set the activity timer event and my backlight does not switch on. I would certainly appreciate your valuable help over this.

Thankyou

"voidcoder" wrote:

Ops, typo in my prev post:

Re: LCD\_Backlight

```
DWORD dwBacklightTimeout =  
ReadTimeoutFromRegistry(...);  
  
while (TRUE)  
{  
WaitForSingleObject(hInactivityTimer,  
INFINITE);  
  
if (WaitForSingleObject(hActivityTimer,  
dwBacklightTimeout) ==  
WAIT_TIMEOUT)  
{  
TurnBacklightOff();  
WaitForSingleObject(hActivityTimer,  
INFINITE);  
TurnBacklightOn();  
}  
}
```

"voidcoder" <voidcoder@xxxxxxxx>  
wrote in message  
[news:ewg0qm2ZGHA.1560@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:ewg0qm2ZGHA.1560@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

I'm not sure what are you  
doing there. Post you code.

Have not used activity  
timers this way, but it  
should  
be possible to use something  
like this:

```
DWORD  
dwBacklightTimeout =  
ReadTimeoutFromRegistry(...);  
  
while (TRUE)  
{  
WaitForSingleObject(hInactivityTimer,  
INFINITE);  
  
if  
(WaitForSingleObject(hActivityTimer,  
dwBacklightTimeout) ==  
WAIT_OBJECT_0)
```

Re: LCD\_Backlight

```
{  
TurnBacklightOff();  
WaitForSingleObject(hActivityTimer,  
INFINITE);  
TurnBacklightOn();  
}  
}
```

"Anurag"

<Anurag@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote in message

[news:4810CF02-46E0-435B-9E63-BC98D849FEF9@xxxxxxxxxxxxxxxxxxxxx](mailto:news:4810CF02-46E0-435B-9E63-BC98D849FEF9@xxxxxxxxxxxxxxxxxxxxx)

Hello  
Voidcoder,  
Thankyou  
for reply  
!!!!  
Actually I  
am able to  
access the  
events in  
my  
backlight  
driver ie it  
does set  
the activity  
event("PowerManager/SystemActivity\_Active")  
during boot  
then  
after 10  
sec(set in  
registry) it  
sets the  
inactivity  
event  
("PowerManager/SystemActivity\_Inactive")  
and I am  
able to  
acknowledge  
it in  
my  
backlight  
driver. I am  
trying to set  
the activity  
event in my  
touch driver

Re: LCD\_Backlight

so that  
every time i  
touch the  
screen the  
activity  
event is set  
and the the  
activity  
timer count  
initializes.  
Now this  
does not  
happen!! I  
am setting  
the  
event as  
follows:  
if(SetEvent(SysAct\_Reset))  
NKDbgPrintfW(TEXT("ASA:Reset  
event  
set\r\n"));

Note:  
SysAct\_Reset  
is the  
handle to  
"PowerManager/ActivityTimer/SystemActivity"  
event

what  
happens is  
that once  
the  
inactivity  
event is set  
the light  
goes off and  
does not  
resume  
back. I am  
resetting the  
manual  
reset  
inactivity  
event  
("PowerManager/SystemActivity\_Inactive")  
by reset  
event.  
What could  
be the issue  
??? In

Re: LCD\_Backlight

anticipation  
of your  
reply  
ASAP.

"voidcoder"  
wrote:

Actually  
activity  
timers  
are  
named  
as  
follow:

PowerManager/ActivityTimer/\$Activity\$  
PowerManager/\$Activity\$\_Active  
PowerManager/\$Activity\$\_Inactive

Eg.

"PowerManager/ActivityTimer/SystemActivity"  
"PowerManager/SystemActivity\_Active"  
"PowerManager/SystemActivity\_Inactive"

Use  
OpenEvent()  
to  
access  
events.

"Anurag"  
<Anurag@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote  
in

message

[news:A2F1D358-DE0B-4FC6-B084-D283DAADF6F2@x](mailto:news:A2F1D358-DE0B-4FC6-B084-D283DAADF6F2@x)

Hello  
Voidcoder,  
Thankyou  
for  
your  
reply  
. .  
certainly  
setting  
the

Re: LCD\_Backlight

power  
states  
would  
be  
much  
better  
however  
for  
the  
purpose  
of  
curiosity

I  
tried  
adding  
that  
registry  
entry  
[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentCon  
"Timeout"=dword:A

and  
then  
created  
the  
events  
with  
the  
same  
names  
i.e,  
SystemActivity\_Active  
&  
SystemActivity\_Inactive  
in  
my  
driver  
however  
these  
events  
get  
created.  
Now  
according  
to  
WinCE  
documentation  
(wince  
help  
topic  
Activitytimers)

Re: LCD\_Backlight

Upon  
initialization,  
the  
Power  
Manager  
reads  
the  
registry  
to  
acquire  
a  
list  
of  
activity  
timer  
names  
and  
creates  
the  
following  
events:

A  
timer  
reset  
event.  
An  
active  
status  
manual-reset  
event.  
A  
manual-reset  
event.

hence  
when  
i  
create  
the  
events  
with  
their  
names  
Createevent()  
should  
return  
ERROR\_ALREADY\_EXISTS  
however  
this  
doesn't  
happen

Re: LCD\_Backlight

the  
events  
get  
created.  
why  
is  
it  
so????  
What  
needs  
to  
be  
done  
in  
this  
case??

"Voidcoder"  
wrote:

I  
would  
advise  
to  
not  
touch  
the  
system  
activity  
timer  
directly.  
Better  
build  
a  
simple  
driver  
and  
handle  
the  
power  
IOCTLs.  
You  
will  
need  
to  
handle  
IOCTL\_POWER\_SET  
ioctl  
this  
way:

Re: LCD\_Backlight

IOCTL\_POWER\_SET

-

D0

->

turn

backlight

on

IOCTL\_POWER\_SET

-

D1

->

turn

backlight

off

"Anurag"

<Anurag@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote

in

message

<news:ADEEAC32-2DCE-4ABF-B45C-E8>

Hello  
everybody,

I

am

trying

to

develop

a

screen

saver

like

facility

for

my

WINCE

Image

i.e,

after

a

predifined

time

of

inactivity

the

LCD

backlight

should

Re: LCD\_Backlight

Re: LCD\_Backlight

be  
OFF  
and  
on  
some  
activity  
like  
a  
touch  
on  
the  
touchscreen  
the  
backlight  
should  
get  
ON  
I  
referred  
the  
wince  
documentation  
"activity  
timer  
"and  
find  
that  
activity  
timers  
can  
be  
used  
i.e.,  
set  
an  
activity  
timer  
[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Power\PowerSettings\{8c5e7c9e-6c3a-412e-b479-d048c4322e13}\1c5c2c31-40a5-4a33-b9cc-446f705984a3]\Attributes\Attributes  
"Timeout"=dword:A  
"WakeSources"=multi\_sz:"0x20"

and  
the  
power  
manager  
handles  
the  
events..  
Also  
the

Re: LCD\_Backlight

event  
can  
be