

API utility for

**GX1LCD/3.5-/S
786LCD/3.5-/S-/ST-/MG-/ETX-/5.25
886LCD/ATX-/ATXU-/ATXGE
886LCD-M/mITX-/Flex-/ATX
889LCD/ATX-/ATXU**

Supported by

Win2000, WinNT 4.0, WinXP, Win9x and Linux (Kernel 2.6, Kernel 2.4)

Introduction

This API utility is designed to enable users to access board features implemented on the GX1LCD, 786LCD, 886LCD, 886LCD-M and 889LCD families of boards in Windows environment.

Installation

The Linux API contains the following files

Device Driver	ltapi2.ko (2.6), ltapi2.o (2.4)
Programming library	ltapi2_lib.o
API function declaration	ltlcd.h

See: Install.txt for installation procedure.

The Windows API contains the following files

NT Device Driver	ltapi2.sys
Win9x Device Driver	ltapi2.vxd
Dynamic programming library	ltlcd.dll
API function declaration	ltlcd.h

Copy device driver ltlcd.sys to the C:\WinNT\System32 or where the System32 directory is located on NT platforms,

Copy device driver ltlcd.vxd to the C:\Windows\System or where the System directory is located on Win9x platforms,

WinNT/Win2K/WinXP Admin Rights.

Run registry file startdrv.reg (**Non Admin users Only**), this file contains driver-loading settings needed by the device manager. These keys must be added to the registry, if these keys are not added to the registry the device manager cannot start the driver during boot and the API interface cannot communicate with the device driver.

Note: If the User has Admin rights to the machine the API will try to load the device driver and the registry key settings are NOT needed.

All other files are platform independent and should be used within a programming project.

**API function descriptions:****DWORD OpenItLcd(VOID)**

This function opens the device driver Itlcd.sys for hardware communication and must be called in order to use any other functions within this API.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD CloseItLcd(VOID)

This function closes the device driver. After closing the driver no attempt to communicate with the driver will be accepted.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD ReadMonitor(HWMON *Mon)

This function takes a HWMON structure and fills the structure with valid data. For return structure see Itlcd.h for the individual data types.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetClrGPIO(BOOL SetClr, UCHAR GPIO)

This function set or clears a GPIO pin, located on the feature port. Make sure to set pin direction before calling this function.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD ReadGPIO(UCHAR GPIO)

This function reads a GPIO pin, located on the feature port. Make sure to set pin direction before calling this function.

Fail flag bit = 0x100.

Return The return value of this function is combined Data and a failure flag.

DWORD SetGPIDir(UCHAR GPIO)

This function set the direction of the GPIO pins, located on the feature port. Make sure to call this function before calling ReadGPIO or SetClrGPIO.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetFanSpeed(UCHAR Speed)

This function sets the fan speed in the interval between 0-127 where max. speed is 127. Any attempts to write values above 127 will be ignored.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD EnableWD(VOID)

This function enables the watchdog timer. The user must call SetWDTimer and SetWDTimerInterval before calling this function to prevent immediately reboot.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD DisableWD(VOID)

This function disables the watchdog timer. Any attempts to modify watchdog timers after calling this function will have no effect.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD GetMac(UCHAR nMac, PVOID Buffer)

This function receives the MAC address of a specified controller passed in nMac (1=1st, 2=2nd, 3=3th). The input buffer must be at least 6 bytes long.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetWDTimer(UCHAR Time)

This function sets the watchdog timer. An application must service this function and reload the timer to prevent reboot; the number of units is between 0-255.

Note: 886LCD/ATX, 886LCD-M/mITX/Flex/ATX Interval Bit 0:5 is only supported

Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0h-3h will be ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and this allows timeouts ranging from 2.4 seconds to 38 seconds.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetWDTimerInterval(UCHAR VAL)

This function set the watchdog timer interval. The interval is multiplied with the WDTimer value and represents the time-out period. There are four selectable intervals listed in the Itlcd.h file. **4MS 1SEC 1MIN 1HOUR.**

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetCPUPhrottle(UCHAR DUTY)

This function set CPU throttle an application can call this function to slow down the CPU speed and save power. The selectable duty cycle intervals are listed in Itlcd.h.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetBLKControl(BOOL ON_OFF)

This function turns on/off the backlight on the LCD display.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetLCDVCCControl(BOOL ON_OFF)

This function turns on/off the LCDVVC on the LCD display.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetLCDVCC(BOOL VCC5_VCC3)

This function sets the LCDVVC voltage on the LCD display.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetSerialInterface(UCHAR INTERFACE)

This function selects the serial interface. The following interfaces are provided: RS232, RS422, and RS485. The values to be used are listed in Itlcd.h.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetSerialInterfaceDataControl(UCHAR PIN)

This function selects the hardware data flow control used on RS422 and RS485 interfaces. Selectable values are listed in Itlcd.h.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SetSerialInterfaceControl(UCHAR PIN)

This function selects the hardware control flow control used on RS422 and RS485 interfaces. Selectable values are listed in Itlcd.h.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD ReadBoardHeader(PVOID Buffer)

This function read the Inside Header Info from the Memory Area. The argument passed to the function must be a pointer to a structure of minimum 19 Bytes. The information returned include Board name, Version of Hardware and Software, and OUI network address. **(See Appendix A for structure info)**

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SelectFanTempTacChannel(UCHAR Channel)

This function selects the hardware Fan/Temp and Tachometer channel default is channel 0.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD SelectFanTempTacChannelType(UCHAR Type)

This function selects the hardware sensor type a channel should be selected before calling this function the following sensor types are Diode/Transistor or Resistor.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD StartThermalCruise (VOID)

This function enables Thermal Cruise Control the hardware monitor will automatically control the speed of CPU and System Fan. The target temperature of the CPU is 55 degree Celsius and the System is 40 degree Celsius.

Return If the function succeeds, the return value is nonzero Otherwise the value is zero.

DWORD GetIntruderStatus (PULONG Status)

This function returns the intruder status pin. The status result is returned in **Status**.

Status = 0x00000001 Intruder/Open case detected.

Status = 0x00000000 Intruder/Open case not detected.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD CtlrIntruderStatus (void)

This function clears the intruder status bit.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD GetIdentBlock(int DriveId, _IDENTBlock *IdentBlock)

This function fills the Identifier block structure of a physically drive see format Identifier structure in the itlcd.h file. DriveId is the Physical drive number 0 = Primary Master, 1 = Primary Slave, 2 = Secondary Master, 3 = Secondary Slave.

Note: This function is only supported in WinNt/Win2K/WinXp.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

DWORD GetIDCount(UCHAR DriveNo)

This function returns the number of S.M.A.R.T Attributes supported by the selected drive. This function should be used together with the GetHDDAttributes function.

DriveNo is the Physical drive number 0 = Primary Master, 1 = Primary Slave, 2 = Secondary Master, 3 = Secondary Slave.

Note: This function is only supported in WinNt/Win2K/WinXp.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

GetHDDAttributes(int DriveNo, _HDDDataBlock *pHDDDataBlock,DWORD dwIDCount)

This function fills the structure array of S.M.A.R.T Attributes passed to the function.

The dwIDConut is the number of structures passed to the function equals the number returned by the GetIDCount function. DriveNo is the Physical drive number 0 = Primary Master, 1 = Primary Slave, 2 = Secondary Master, 3 = Secondary Slave.

Note: This function is only supported in WinNt/Win2K/WinXp.

Return If the function succeeds, the return value is nonzero. Otherwise the value is zero.

Note: Some of the above listed functions are not supported for all types of boards (see next page). Unsupported functions return a value of zero.


Supported API's:

Functions	786LCD -/S -/ST -/3.5 -/MG	786LCD -/ETX	786LCD -/5.25	GX1LCD -/S -/3.5	886LCD -/ATX -/ATXU -/ATXGE	886LCD-M -/mITX -/Flex -/ATX	889LCD -/ATX -/ATXU
OpenItLcd	X	X	X	X	X	X	X
CloseItLcd	X	X	X	X	X	X	X
ReadMonitor	X	X	X	X	X	X	X
SetClrGPIO	X	X	X	X		X	
ReadGPIO	X	X	X	X		X	
SetGPiODir	X	X	X	X		X	
SetFanSpeed	X	X	X		X	X	X
EnableWD	X	X	X		X	X	X
DisableWD	X	X	X		X	X	X
SetWDTimer	X	X	X	X	X	X	X
SetWDTimerInterval	X	X	X				
SetCPUThrottle	X	X	X		X	X	X
SetBLKControl	X	X		X			
SetLCDVCCControl	X	X		X			
SetLCDVCC	X	X		X			
SetSerialInterface	X			X			
SetSerialInterfaceDataControl	X			X			
SetSerialInterfaceControl	X			X			
ReadBoardHeader	X	X		X		X	
SelectFanTempTacChannel	X	X	X		X	X	X
SelectFanTempTacChannelType	X	X	X		X	X	
GetIntruderStatus					X	X	X
ClrIntruderStatus					X	X	X
GetIdentBlock	X	X	X	X	X	X	X
GetIDCount	X	X	X	X	X	X	X
GetHDDAttributes	X	X	X	X	X	X	X
StartThermalCruise						X	
GetMac						X	



Appendix A: How to read the IIH (Integrated Info Header) from the BIOS of Kontron Technology SBC's, in order to get board identification.

The IIH is implemented on 886LCD-M family, the 786LCD family, the 686 LCD family from BIOS version 117, and on the 586/686 GXM from BIOS version 1.5. For 886LCD-M you can use DMI (Desktop Management Interface) or API function Readboardheader() .

Disclaimer: KONTRON Technology A/S reserves the right to make changes without notice.

The IIH block contains the information described in the table:

Field	Size	Contents	Offset
Magicscan	4 bytes	'\$IIH' (24h, 49h, 49h, 48h)	0-3
Infosize	1 byte	Amount of info in bytes, exclude header and this byte	4
Boardinfo	1 byte	0 = 486LCD 1 = 586LCD 2 = 686LCD/S 3 = 686LCD/MG (only distinguished if SCSI controller is present) 4 = 586GXM 5 = 686GXM 6 = GXMBASIC 10 = GX1LCD/S 11 = GX1LCD/S Plus 18 = GX1LCD/3"5 Standard 19 = GX1LCD/3"5 Plus 20 = 786LCD/S 21 = 786LCD/MG 24 = 786LCD/ST 28 = 786LCD/5.25 30 = ETXplus-786 40 = 886LCD-M/Flex 48 = 886LCD-M/mITX 50 = 886LCD-M/ATX	5
BIOSmjr	1 word	BIOS MAJORVERSION (in Hex value)	6-7
BIOSmnr	1 word	BIOS MINORVERSION (in Hex value)	8-9
PALid	1 byte	This byte reflect the ID from CPLD	10

In case of 886LCD-M:

Field	Size	Contents	Offset
S/N	4 bytes	S/N in BCD	11-14
P/N	4 bytes	P/N in BCD	15-18

In case of 786LCD family or GXMLCD family BIOS version 1.6 and above:

Field	Size	Contents	Offset
=OUI1	1 word	This word is the first word of the OUI address (00 E0)	11-12
=OUI2	1 word	This word is the second word of the OUI address (F4 xx)	13-14
=OUI3	1 word	This word is the third word of the OUI address (xx xx)	15-16

In case of 686LCD family bios version 119 and above:

Field	Size	Contents	Offset
OUI	6 bytes	OUI address (00 E0 F4 xx xx xx)	11-16