rev.1.0.8-20200921

GbtLinuxFunc API User Guide

Release Note

Release Date	Revision	Note
20200921	1.0.8	1. Add new HW support for GA-SBC5005 2. Add Ubuntu 5.4.0-47-generic Kernel support
20200703	1.0.7	3. Add new HW support for GA-SBC5005 4. Remove Ubuntu 16.04 Kernel support
20190827	1.0.6	1. Add new kernel support for a. Ubuntu 16.04.6-4.15.0-58-generic b. Ubuntu 18.04.5-5.0.0-25-generic 5. Add new HW support for GA-IMB4100TN
20190605	1.0.5	1. Add new kernel support for a. Ubuntu 16.04.6-4.15.0-51-generic b. Ubuntu 18.04.2-4.18.0-21-generic
20181025	1.0.4	1. Change Device Name as “GbtLinuxFunc”
20181024	1.0.3	1. Add new HW support for GA-IMBLAP3450
20180910	1.0.2	1. Add new HW support for GA-IMB310TN 2. Add new SW support for Watchdog Control API 3. Add new SW support for SpeakerBeep (internal speaker) Control API 4. Add new SW support for DebugPort write API
20180202	1.0.1	1. Redefine header26 pin number for GA-IMBLAP3350 and GA-N3160TN
20180201	1.0.0	1. First release 2. Add new HW support for c. GA-IMBLAP3350 d. GA-N3160TN e. GA-H110TN

		3. Add new SW support for GPIO Control API

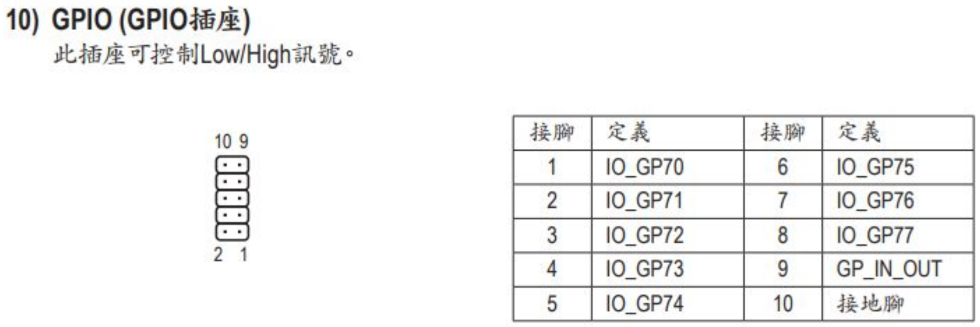
Supported Kernel Build Version:

5.4.0-47-generic
5.3.0-62-generic
5.0.0-25-generic
~~4.18.0-25-generic~~
4.18.0-21-generic (ubuntu-18.04.2)
4.18.0-20-generic
~~4.15.0-54-generic~~
~~4.15.0-51-generic~~
~~4.15.0-51-generic (ubuntu-16.04.6)~~
~~4.15.0-30-generic~~
~~4.13.0-26-generic (kingston request)~~
~~4.10.0-28-generic~~

Supported Hardware:

GA-SBC4200
GA-SBC5005SE
GA-IMB4100TN
GA-IMB310TN
GA-IMBLAP3450
GA-IMBLAP3350
GA-N3160TN
GA-H110TN

For GA-IMB310TN/GA-H110TN



10) GPIO (GPIO插座)

此插座可控制Low/High訊號。

接腳	定義	接腳	定義
1	IO_GP70	6	IO_GP75
2	IO_GP71	7	IO_GP76
3	IO_GP72	8	IO_GP77
4	IO_GP73	9	GP_IN_OUT
5	IO_GP74	10	接地腳

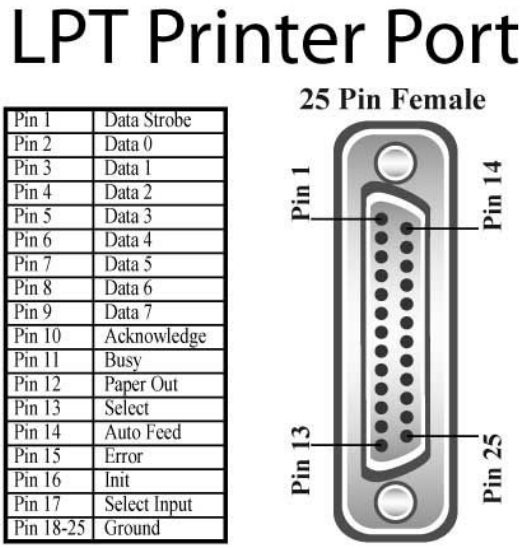
Note : Supported GPIO Pin : 1~8

註：若輸入無效的Pin Number(例如: 9,10),將回應Error Code.

For GA-IMBLAP3350-CF/GA-IMBLAP3450

For GA-N3160TN

For GA-IMB4100TN



LPT Printer Port

Pin 1	Data Strobe
Pin 2	Data 0
Pin 3	Data 1
Pin 4	Data 2
Pin 5	Data 3
Pin 6	Data 4
Pin 7	Data 5
Pin 8	Data 6
Pin 9	Data 7
Pin 10	Acknowledge
Pin 11	Busy
Pin 12	Paper Out
Pin 13	Select
Pin 14	Auto Feed
Pin 15	Error
Pin 16	Init
Pin 17	Select Input
Pin 18-25	Ground

Note : Supported GPIO Pin : 1~14 , 16~17

PIN	SIGNAL NAME	PIN	SIGNAL NAME
01	SIO_GP87	14	SIO_GP86
02	SIO_GP70	15	NA
03	SIO_GP71	16	SIO_GP85
04	SIO_GP72	17	SIO_GP84
05	SIO_GP73	18	GPIOPWR
06	SIO_GP74	19	GPIOPWR
07	SIO_GP75	20	GND
08	SIO_GP76	21	GND
09	SIO_GP77	22	GND
10	SIO_GP83	23	GND
11	SIO_GP82	24	GND
12	SIO_GP81	25	GND
13	SIO_GP80	26	NC

註1：若輸入無效的Pin Number(例如:0,15,18~26),將回應Error Code.
註2：掛上Driver啟動後,Driver會強制切到GPIO mode,不會理會硬體LPT_SEL jumper 設定

Folder

```
$ tree
.
├── Examples
│   ├── Makefile
│   ├── GbtLinuxFunc.h
│   ├── GbtLinuxFunc.c
│   ├── watchdogtst.c
│   ├── digitalIotst.c
│   ├── speakerbeeptst.c
│   └── debugporttst.c
└── Driver
    ├── GbtLinuxFuncDrv.5.4.0-47-generic.ko
    ├── GbtLinuxFuncDrv.4.18.0-25-generic.ko
    ├── install.sh
    └── uninstall.sh
```

---

install.sh

```
$ cd Driver
$ sudo modeporbe wmi

$ sudo insmod GbtLinuxFuncDrv.ko
$ sudo chmod 666 /dev/GbtLinuxFunc

$ modinfo ./GbtLinuxFuncDrv5.4.0-47-generic.ko
mark@mark-virtual-machine:~/GbtLinuxFunc/GbtLinuxFunc_release/Driver$ modinfo ./GbtLinuxFuncDrv.5.4.0-47-generic.ko
filename:       /home/mark/GbtLinuxFunc/GbtLinuxFunc_release/Driver/./GbtLinuxFuncDrv.5.4.0-47-generic.ko
version:        1.0.8
```

```
license:       GPL
description:    Gigabyte Embedded Board Linux Control Driver
author:        Mark Tsai<mark@gigabyte.com><marktsai0316@gmail.com>
alias:         wmi:DEADBEEF-2001-0000-00A0-C90629100000
alias:         wmi:ABBC0F6F-8EA1-1459-00A0-C90629100000
srcversion:     5A50DD8C6A7CBD078732E94
alias:         dmi*:rn*SBC5005*:
alias:         dmi*:rn*SBC4200*:
alias:         dmi*:rn*IMB4100TN*:
alias:         dmi*:rn*IMB310TN*:
alias:         dmi*:rn*IMBLAP3450*:
alias:         dmi*:rn*IMBLAP3350*:
alias:         dmi*:rn*N3160TN*:
alias:         dmi*:rn*H110TN*:
depends:        wmi
retpoline:     Y
name:          GbtLinuxFuncDrv
vermagic:       5.4.0-47-generic SMP mod_unload
```

```
$ lsmod | grep GbtLinuxFuncDrv
GbtLinuxFuncDrv          16384  0
```

```
$ dmesg | tail -n 4
[74901.330051] GbtLinuxFuncDrv: GbtLinuxFunc module initial.
[74901.330053] Identified model 'GA-IMB310TN' ID=0xA0
[74901.330058] Force switch SIO mode from LPT to GPIO
[74901.330625] I got: 131072 bytes of memory
```

註2 : 掛上Driver啟動後,Driver會強制切到GPIO mode,不會理會硬體LPT_SEL jumper 設定

```
$ cat /proc/devices | grep GbtLinuxFunc
245 GbtLinuxFunc
```

```
$ ls /dev/GbtLinuxFunc -la
crw------- 1 root root 245, 0  7月 21 13:00 /dev/GbtLinuxFunc
```

uninstall.sh

```
$ sudo rmmod GbtLinuxFuncDrv
```

```
[74908.137850] GbtLinuxFuncdrv: Goodbye, GIGABYTE!
```

```
sudo apt install make g++
```

DigitalIO API Library

see GbtLinuxFunc.h and digitalIo.c

```
//Defining Digital Pins modes: INPUT, INPUT_PULLUP, and OUTPUT

#define INPUT         0
#define INPUT_PULLUP  1
#define OUTPUT        2

//Defining Pin Levels: HIGH and LOW
#define HIGH          1
#define LOW           0

int DigitalIo_Init();
void DigitalIo_Uninit(int fd);
int DigitalIo_PinMode(int fd, int pin , int mode);
int DigitalIo_DigitalWrite(int fd, int pin,int value );
int DigitalIo_DigitalRead(int fd, int pin );
```

Example :

```
#include "GbtLinuxFunc.h"
int main(void)
{

        int fd=DigitalIo_Init();
        int ledPin = PIN1;

        DigitalIo_PinMode(fd, ledPin, OUTPUT);
        //sleep(1); // one second
        usleep(1000000); //one second
        DigitalIo_DigitalWrite(fd, ledPin, HIGH ); //sets the LED on
        usleep(1000000); //one second
        DigitalIo_DigitalWrite(fd, ledPin, LOW );  //sets the LED off
        usleep(1000000); //one second
        DigitalIo_DigitalWrite(fd, ledPin, HIGH ); //sets the LED on
        usleep(1000000); //one second
        DigitalIo_DigitalWrite(fd, ledPin, LOW );  //sets the LED off
        usleep(1000000); //one second

        DigitalIo_Uninit(fd);
        return 0;
}
```

Watchdog API Library

```
int WatchDog_Control(int fd, int interval); //0 : Disable  , 1~255 unit in second
int WatchDog_Status(int fd, unsigned short *pTimeoutValue, unsigned short *pWdtStatus);
int WatchDog_BeatBeep(int fd, int Enable);
```

Example :

```
int main(void)
{
        unsigned short TimeoutValue,WdtStatus;
        int fd=GbtLinuxFuncDrv_Init();
        //You can enable BeatBeep to hear system's heartbeat by internal speaker
        WatchDog_BeatBeep(fd, 1);  //Enable BeatBeep

        WatchDog_Status(fd,&TimeoutValue,&WdtStatus);
        printf("Current Timeout value is %d secs, WDT is %s\n",
                TimeoutValue ,(WdtStatus == 0) ? "Disabled" : "Enabled" );
        WatchDog_Control(fd, 15);  //set WDT Timeout Value as 15 secs
        WatchDog_Status(fd,&TimeoutValue,&WdtStatus);
        printf("Current Timeout value is %d secs, WDT is %s\n",
                TimeoutValue ,(WdtStatus == 0) ? "Disabled" : "Enabled" );

        usleep(1000000*5); //5 secs
        WatchDog_Control(fd, 20);  //set WDT Timeout Value as 20 secs
        WatchDog_Status(fd,&TimeoutValue,&WdtStatus);
        printf("Current Timeout value is %d secs, WDT is %s\n",
                TimeoutValue ,(WdtStatus == 0) ? "Disabled" : "Enabled" );

        WatchDog_Control(fd, 0);  // Disable function
        WatchDog_Status(fd,&TimeoutValue,&WdtStatus);
```

```
    printf("Current Timeout value is %d secs, WDT is %s\n",
        TimeoutValue ,(WdtStatus == 0) ? "Disabled" : "Enabled" );

    //If you want system to mute ,You can disable BeatBeep
    //WatchDog_BeatBeep(fd, 0);  //Disable BeatBeep

    GbtLinuxFuncDrv_Uninit(fd);
    return 0;
}
```

SpeakerBeep (Internal Speaker) API Library

```
int SpeakerBeep_Control(int fd, unsigned short note,unsigned short duration);
```

Example :

```
int main(void)
{
    int fd=GbtLinuxFuncDrv_Init();
    int i;
    for(i=1;i<=21;i++)
    {
        SpeakerBeep_Control(fd,i,500); //set duration == 500 milisecond
    }

    GbtLinuxFuncDrv_Uninit(fd);
    return 0;
}
```

Debugport API Library

```
int DebugPort_Write(int fd, unsigned char value );
```

Example :

```
int main(void)
{
    int fd=GbtLinuxFuncDrv_Init();
    DebugPort_Write(fd, 0xAA );
    usleep(1000000*1); //one second
```

```
        DebugPort_Write(fd, 0x55 );
        usleep(1000000*1); //one second
        DebugPort_Write(fd, 0xAA );
        usleep(1000000*1); //one second
        DebugPort_Write(fd, 0x55 );
        usleep(1000000*1); //one second
        GbtLinuxFuncDrv_Uninit(fd);
        return 0;
}
```

---

dmidecode

```
mark@test:~/GbtLinuxFunc/Driver$ sudo dmidecode -s baseboard-manufacturer
Gigabyte Technology Co., Ltd.
mark@test:~/GbtLinuxFunc/Driver$ sudo dmidecode -s baseboard-product-name
IMBLAP3350-CF
mark@test:~/GbtLinuxFunc/Driver$ sudo dmidecode -t 2
# dmidecode 3.0
Getting SMBIOS data from sysfs.
SMBIOS 3.0.0 present.

Handle 0x0002, DMI type 2, 15 bytes
Base Board Information
        Manufacturer: Gigabyte Technology Co., Ltd.
        Product Name: IMBLAP3350-CF
        Version: Default string
        Serial Number: Default string
        Asset Tag: Default string
        Features:
                Board is a hosting board
                Board is replaceable
        Location In Chassis: Default string
        Chassis Handle: 0x0003
        Type: Motherboard
        Contained Object Handles: 0
```