|  |  |
| --- | --- |
| **Student**: | Michael Escue |
| **Assignment**: | Assignment #3 |
| **Class**: | ECE 373 |
| **Instructors**: | Peter (PJ) Waskiewicz,  Charles Stoll |
| **Term**: | Spring 2019 |

**Target**: Advantech AIMB-212

**Question 1:**

1. *What is the audio device?*

AC-97 audio interface.

1. *What device is the GPIO connected to?*

The 8-bit GPIO is connected to a Super IO Winbond W83627DHG through a bi-directional connection.

1. *How many network (LAN) devices are on the motherboard and what are they?*

The board has two Gigabit Ethernet LANs, LAN 1 (Intel 82567V) and LAN 2 (Intel 82583V).

1. *How many total serial ports does the box support, inside and out?*

8 serial ports. Serial ATA 0, Serial ATA 1, and Serial Ports COM1 – COM6. COM2 is RS-232/422/485 and the other 5 COM ports are RS-232 only.

**Question 2:**

* *Find the datasheet for the LAN device connected through PCIe (not through the integrated chip’s LCI). Read the Pin interface chapter on how to physically connect to the chip. Find the memory mapped register and bit patterns to perform “things”.*

The device is a Gigabyte LAN2: Intel 82583V

1. *What pins control the LEDs?*

LED0 Lead # 31 Output

LED1 Lead # 30 Output

LED2 Lead # 33 Output

1. *What address offset is the Device Control Register?*

**CTRL** Register offset: 0x00000 / 0x00004

1. What bit in the Device Control Register will force a reset of the network chip?

Bit 26 when set initiates a device reset.

**Question 3:**

1. *What register (name and address) controls the LEDs?*

The configuration for LED outputs is specified via the **LEDCTL** register.

**LEDCTL** Register offset: 0x00E00

**Setting Hardware defaults:**

Non-Volatile Memory (NVM) Map Address Range: 0x00-0x3F

(NVM) LEDCTL1 Config Defaults/PHY config 0x1C

(NVM) LED0, LED2 Config Defaults: 0x1F

1. *What bit pattern should you use to turn off LED1?*

R-M(OR)-W:

0000\_0000\_0000\_0000\_0000\_**1111**\_0000\_0000

1. *What bit pattern should you use to make LED2 blink?*

R-M(OR)-W:

0000\_0000\_**1**000\_0000\_**111**0\_0000\_0000\_0000

**Question 4:**

1. *What company makes the Super I/O chip in this box?*

NUVOTON

1. *Can you find the datasheet on the web?*

YES: <http://www.nuvoton.com/resource-files/DA00-WW83627DHG-P.pdf>

1. *Where/how can you find this chip’s datasheet if you aren’t able to find it with a standard web search?*

From the vendor themselves.

**Part 2:**

See source code.

**Question 5:**

Confirmed. Char device opens, reads, and writes.

**Question 6:**

***lspci -n -v output:***

00:03.0 0200: 8086:100e (rev 02)

Subsystem: 8086:001e

Flags: bus master, 66MHz, medium devsel, latency 64, IRQ 19

Memory at f8200000 (32-bit, non-prefetchable) [size=128K]

I/O ports at d020 [size=8]

Capabilities: [dc] Power Management version 2

Capabilities: [e4] PCI-X non-bridge device

Kernel driver in use: **e1000**

Kernel modules: e1000

***lspci -s 00:03.0 -vv output:***

00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)

**Subsystem**: Intel Corporation PRO/1000 MT Desktop Adapter

**Control**: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-

**Status**: Cap+ 66MHz+ UDF- FastB2B- ParErr- DEVSEL=medium >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

**Latency**: 64 (63750ns min)

**Interrupt**: pin A routed to IRQ 19

**Region 0**: Memory at f8200000 (32-bit, non-prefetchable) [size=128K]

**Region** **2**: I/O ports at d020 [size=8]

**Capabilities**: [dc] Power Management version 2

**Flags**: PMEClk- DSI+ D1- D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot-,D3cold-)

**Status**: D0 NoSoftRst- PME-Enable- DSel=0 DScale=0 PME-

**Capabilities**: [e4] PCI-X non-bridge device

**Command**: DPERE- ERO+ RBC=512 OST=1

**Status**: Dev=ff:1f.0 64bit- 133MHz- SCD- USC- DC=simple DMMRBC=2048 DMOST=1 DMCRS=8 RSCEM- 266MHz- 533MHz-

**Kernel** **driver in use**: e1000

**Kernel modules**: e1000

1. Attach the device
   1. Implement .probe()
   2. Map the BAR
   3. Store physical address of the BAR
2. Clean up
   1. Implement .remove() routine
3. Hookup PCI driver into init\_module() routine
4. Hookup unregister in your exit\_moduel() routine
5. Unbind the e1000e driver prior to dmesg.