#####MHF05 : 340] 6<3/
#MHF05 : 350] 6<3#



Yhuvlrq#413

h G h

G h

Hʊ Hʊ G

h G

h h G

h h

G

h

G

h G

h G

9 # #1111111111

4

414#

[redacted]

l

415#

5

514#

515#



516#

5164#

ч

I G G

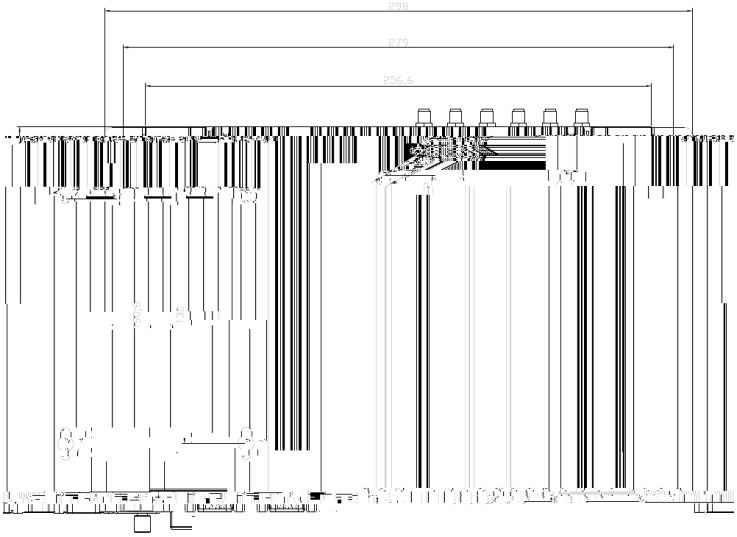
I G h G

I h G

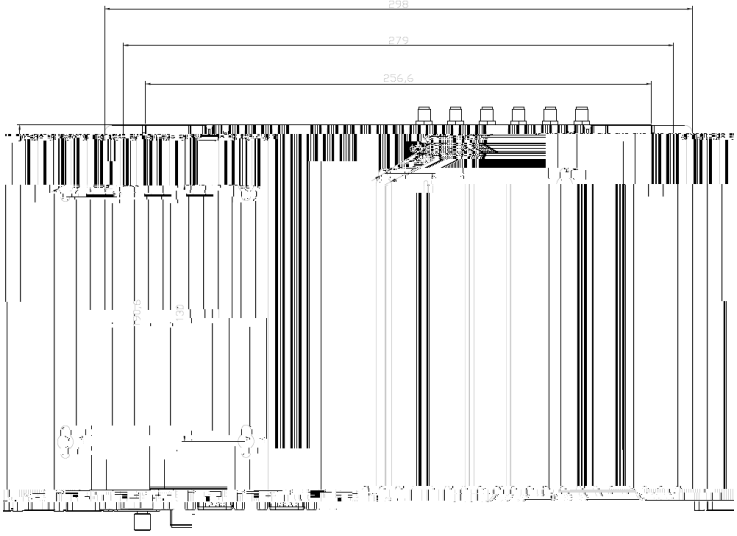
I G h

G

517#



616#



716#

DC IN	

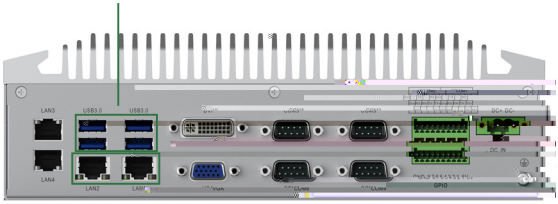
:

8

814#MHF05:340]6<3



81414#MHF05:340]6<3 #LR#

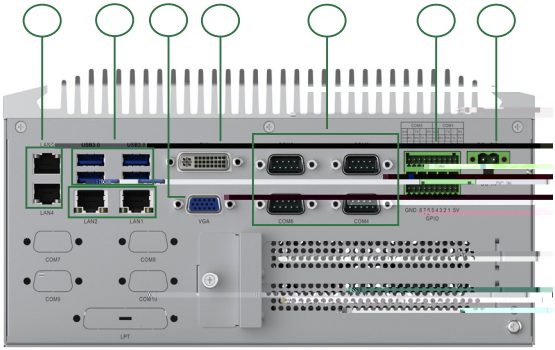


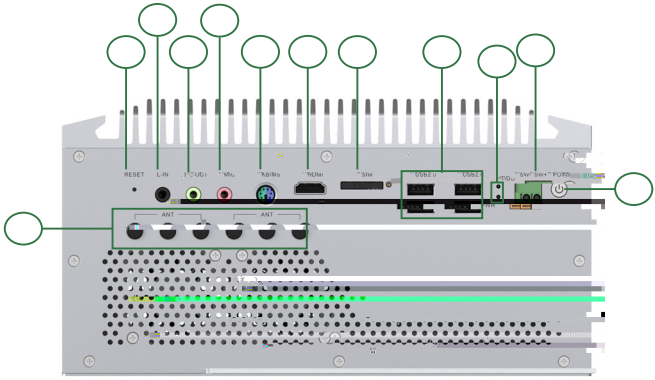
815#MHF05 : 350] 6<3





81514MHF05 : 350] 6<3 #LR#





1. LAN: RJ45
2. US . H RS . H p

81<#

81<14#



81<15#

MEDW#

MPH#

81<18#OSW4#

1	STB	2	AFD
3	PD0	4	ERR
5	PD1	6	INIT
7	PD2	8	SLIN
9	PD3	10	GND
11	PD4	12	GND
13	PD5	14	GND
15	PD6	16	GND
17	PD7	18	GND
19	ACK	20	GND
21	BUSY	22	GND
23	PE	24	GND
25	SLCT	26	NC

81<19#WVSN4#

1	SPK_R-
2	SPK_R+
3	SPK_L-
4	SPK_L+

ч 3W

81<1:#VPE#

1	3.3V
2	DATA
3	CLK
4	GND

81<1;#LQYW4#

1	12V
2	EN
3	PWM
4	GND

81<1<#0YGV4# 0YGV#

1	VCC	2	VCC
3	VCC	4	GND
5	GND	6	GND
7	CHA_D0-	8	CHA_D0+
9	CHA_D1-	10	CHA_D1+
11	CHA_D2-	12	CHA_D2+
13	GND	14	GND
15	CHA_CLK-	16	CHA_CLK+
17	CHA_D3-	18	CHA_D3+
19	CHB_D0-	20	CHB_D0+
21	CHB_D1-	22	CHB_D1+
23	CHB_D2-	24	CHB_D2+
25	GND	26	GND
27	CHB_CLK-	28	CHB_CLK+
29	CHB_D3-	30	CHB_D3+

MOYGV# +
0YGV2HGS#

1-2	3.3V
2-3	5V

LVDS

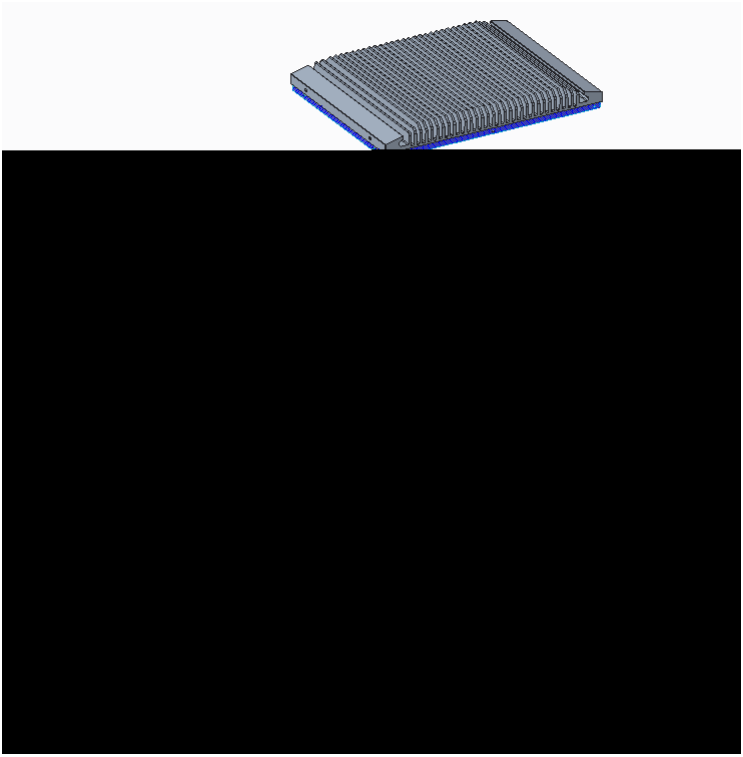
81<143#HGS# # HGS#

1	BKL_PWR	2	BKL_PWR
3	BKL_PWR	4	GND
5	GND	6	NC

EDP

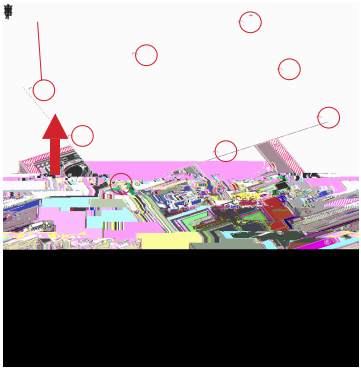
9

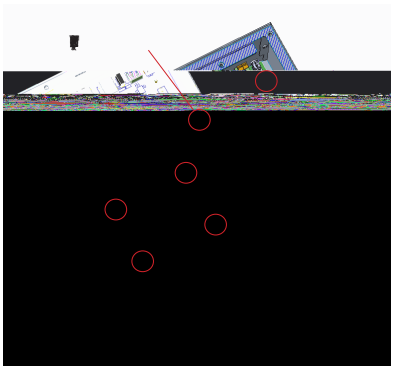
914#



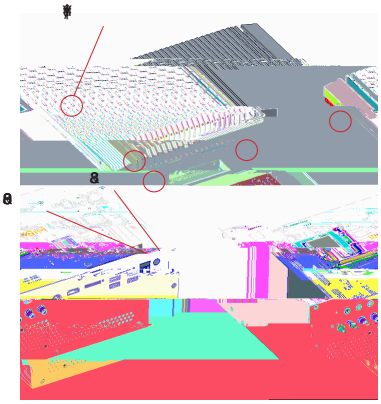
1		2		3	
4		5		6	
7		8		9	
10		11		12	
13					

915#FSX#





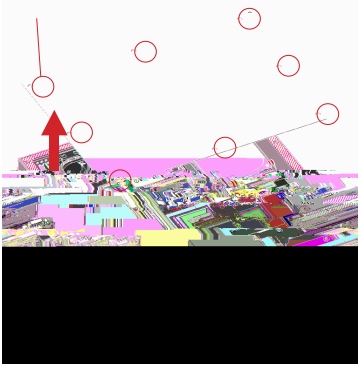


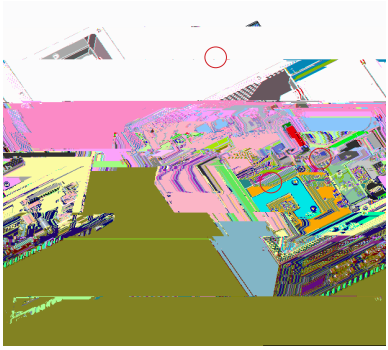


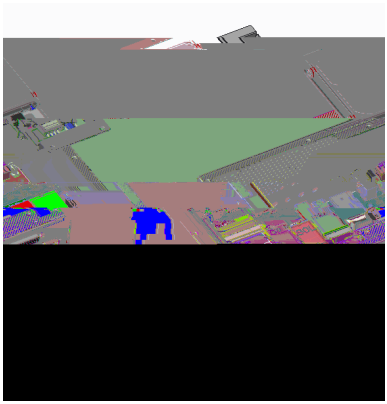
1 . 10 6#32 * 5 h Ꮒ
2 . 10 ⅢЗ Ꮒ
3 . 10 6#32 * 5 h Ꮒ
4 . h C PU h C PU h G

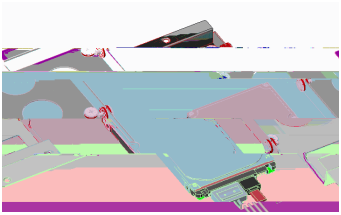
1 . C PU h C PU h C PU Ꮒ
2 . C PU C PU h G

916#

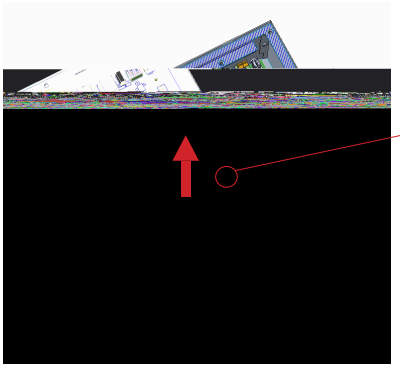








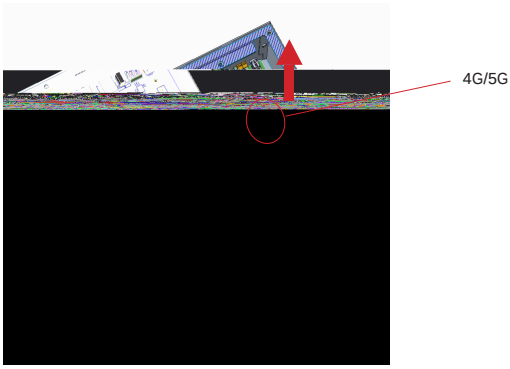
919#P15#5575255;3#



ч M.2 1 M2 h M.2 2242/2280 SATA SSD G

ч1. h 30° h h Ц
2. M.2 2242 h M.2 2280 M.2 2242

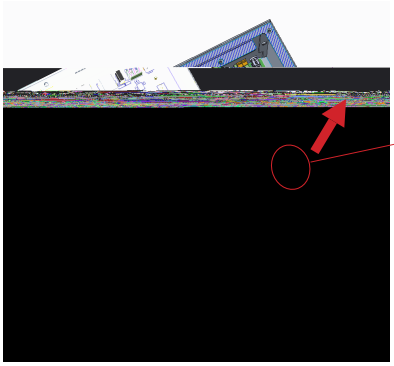
91:#7J28J#



ч 4G/5G G

ч 4G/5G h 30° G

91;#ZLIL#



ч WIFI g

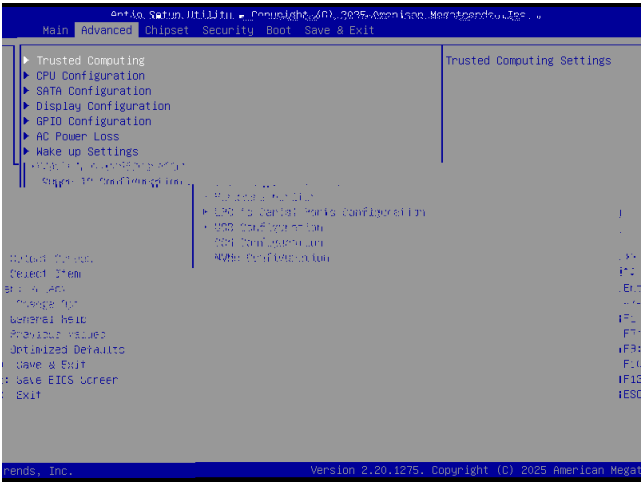
ч WIFI h 30° g

: #ELRV#

:14#ELRV#

:15#ELRV#

:18#Dgydqfhg



Trusted Computing		
CPU Conžguration	CPU	
SATA Conžguration	SATA	
Display Conžguration		
GPIO Conžguration	GPIO	
AC Power Loss		
Wake up Settings		
Watch Dog Conžguration		
Super IO Conžguration	IO	
Hardware Monitor		/
LPC to Serial Ports Conžguration		
USB Conžguration	USB	
CSM Conžguration	CSM	
NVMe Conžguration	NVMe	/

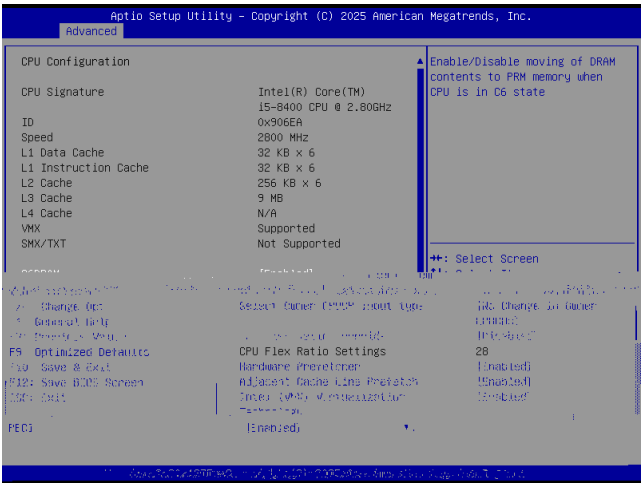
y

Vhfxulw|#Ghylfh#Vxssruw

VKD04#SFU#Edqn###

####

y



F9GUDP

CPU C6 DRAM PRM G ↳ Enabled,

Vriwzduh#Jxdug#H{whqvlrqv+VJ[,####

SGX G ↳ Software Controlled,SGX Enable.

Vhohfw#Rzqhu#HSRFK#lqsxw#w|sh####

TPM Owner EPOCHŸ ÿ G ↳ No Change in Owner EPOCHs.

FSX#Ioh{#Udwlr#Ryhuulgh

CPU G ↳ Disabled,Enabled.

Kdugzduh#Suhihwfkhu

G ↳ Disabled,Enabled.

Dgmdfhqw#Fdfkh#Olqh#Suhihwfk

G ↳ Disabled,Enabled.

Lqwho+YP[,#Yluxdol}dwlrq#

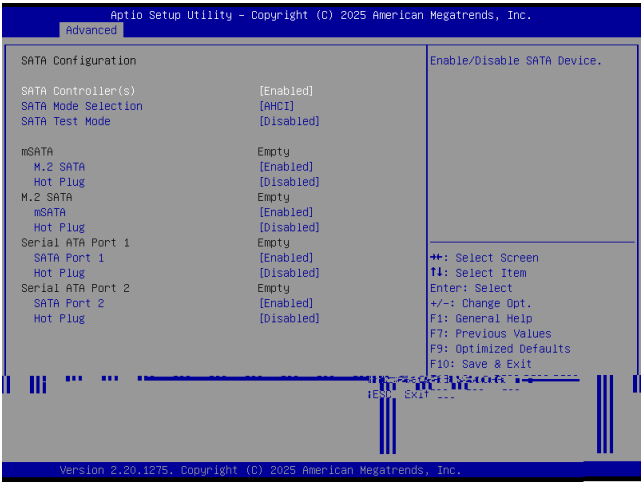
IntelŸVMXÿ G ↳ Disable,Enable.

SHFL

PCIE G ↳ Disabled,Enabled.

#

y



VDWD#Frqwurohu+v,

SATA G ↳Dis a bled,En a bled .

VDWD#Prgh#Vhohfwlrq

SATA G ↳AHC I,R a id .

VDWD#Whvw#Prgh

SATA G ↳Dis a bled,En a bled .

#P15#VDWD#

M . 2 SATA G ↳Dis a bled,En a bled .

Krw#Soxj

G ↳Dis a bled,En a bled .

#pVDWD

mSATA G ↳Dis a bled,En a bled .

VDWD#Sruw425

SATA G ↳Dis a bled,En a bled .

y

Zdwfk#Grj#Frqwur###

y Hardware Monitor <Enter> G

#Vpduw#Idq#Ixqfwlrq

G

y

Ohjdf|#XVE#Vxssruw

USB G ч l <EnRs ҷ Ȯ ӯ ᵠ ᵐ א Ř

XVE#wudqvihu#wlph0rxw

USB G ↳1sec,5sec,10sec,20sec .

Ghylfh#uhvhw#wlph0rxw

G ↳10sec,20sec,30sec,40sec .

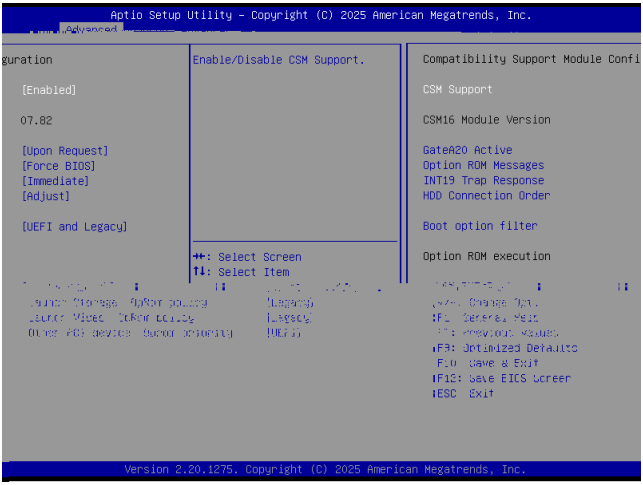
Ghylfh#srzhu0xs#ghod|

G ↳Auto,Manual .

DGDWD#XVE#lodvk#Gulyh#4133

G ↳Auto .

3 [redacted]



[redacted]FVP#Vxssruw

CSM G ↳Disable,Enable .

[redacted]JdwhD53#Dwlyh

20 G ↳Upon Request,Always .

[redacted]Rswlrq#URP#Phvvdjhv

ROM G ↳Force BIOS,Keep Current .

[redacted]LQW4<#Wuds#Uhvsrqvh

INT19 G ↳Immediate,Postpond .

[redacted]KGG#Frqqhfwlrq#Rughu

G ↳Adjust .

[redacted]Errw#rswlrq# owhu

G ↳UEFI and Legacy,Legacy only,UEFI only .

[redacted]ODQ#S[H#Errw

G ↳Do not launch .

[redacted]Odxqfk#Vwrudjh#RsUrp#srolf|

Option Rom G ↳Do not launch,UEFI,Legacy .

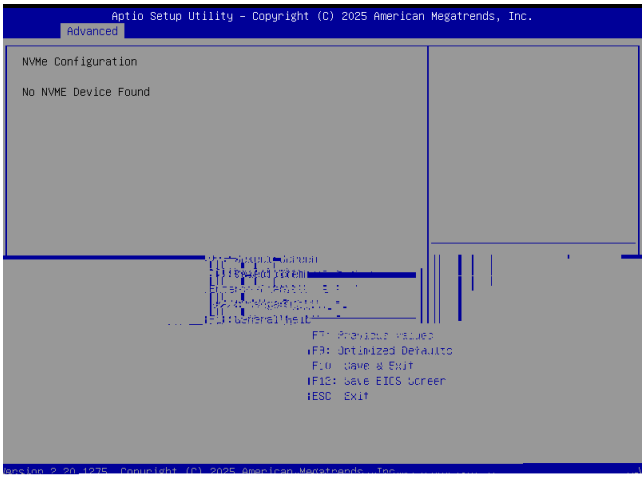
0dxqfk#Ylghr#RsUrp#srlf|

Option Rom g ↳Do not launch,UEFI,Legacy.

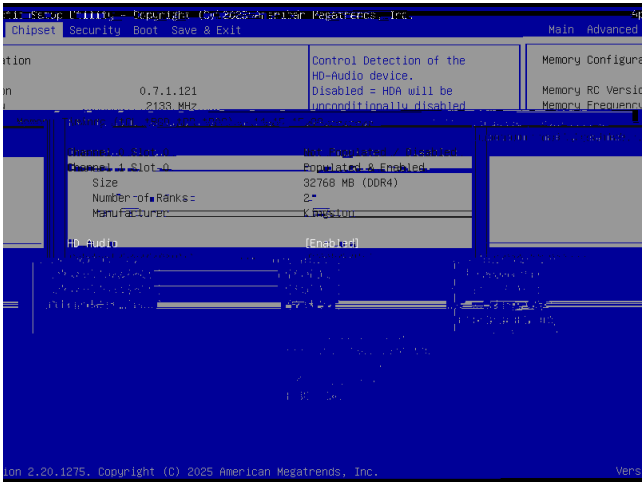
Rwkhu#SFL#ghylfh#Rsurp#sulruw|#

PCI Oprom g ↳Do not launch,UEFI,Legacy.

NVMe Conžguration <Enter> g



:19#Fklsvhw



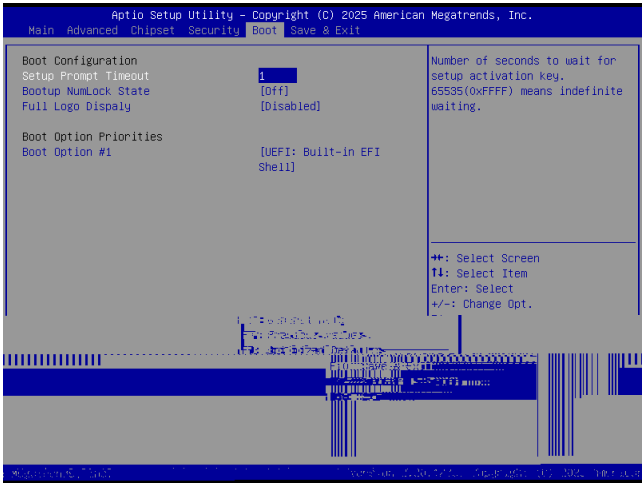
KG#Dxglr##

g ↳Disabled,Enabled.

Rqerdug#0dq4252627#

g ↳Disabled,Enabled.

:1;#Errw



Vhwxs#Surpsw#Wlphrxw

G

Errw#Qxp0rfn#Vwdwh

NumLock G ↯On,Oð .

Ixoo#Orjr#Glvsdo|

G ↯Disable,Enable .

Errw##Rswlrq#&4

1 G ↯UEFI:Built-IN EFI Shell .

; #JSLR

;14#

JEC - 2701V4 . 0 GPIO G ч
GPIO Intel PCH h GPIO G
GPIO G

;1414# #ELRV#

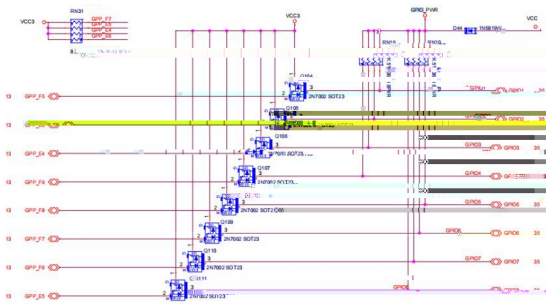
```
/**
Reads a 32-bit MMIO register.
Reads the 32-bit MMIO register specižed by Address. The 32-bit re
value is
returned. This function must guarantee that all MMIO read and writ
operations are serialized.
If Address is not aligned on a 32-bit boundary, then ASSERT().
If 32-bit MMIO register operations are not supported, then ASSERT()
@param Address The MMIO register to read.
@return The value read.
**/
UINT32
EFIAPI
MmioRead32 (
IN UINTN Address
)
{
//
// Make sure Address is aligned on a 32-bit boundary.
//
ASSERT ((Address & 3) == 0);
return (UINT32)MmioReadWorker (Address, EžCpuIoWidthUint32);
}
/**
Writes a 32-bit MMIO register.
specižed
by Value and returns Value. This function must guarantee that all
read
and write operations are serialized.
If Address is not aligned on a 32-bit boundary, then ASSERT().
If 32-bit MMIO register operations are not supported, then ASSERT()
@param Address The MMIO register to write.
@param Value The value to write to the MMIO register.
**/
```

```
   UINT32
EFIAPI
MmioWrite32 (
IN UINTN Address,
IN UINT32 Value
)
{
//
// Make sure Address is aligned on a 32-bit boundary.
//
ASSERT ((Address & 3) == 0);
return (UINT32)MmioWriteWorker (Address, EfiCpuIoWidthUint32, Value);
}
/**
Reads memory-mapped registers in the EFI system memory space.
Reads the MMIO registers specified by Address with registers width
specified by Width.
The read value is returned. If such operations are not supported,
ASSERT().
This function must guarantee that all MMIO read and write operation
are serialized.
@param Address The MMIO register to read.
The caller is responsible for aligning the Address
if required.
@param Width The width of the I/O operation.
@return Data read from registers in the EFI system memory space.
  **/
UINT64
EFIAPI
MmioReadWorker (
IN UINTN Address,
IN EFI_CPU_IO_PROTOCOL_WIDTH Width
)
{
EFI_STATUS Status;
UINT64 Data;
Status = mCpuIo->Mem.Read (mCpuIo, Width, Address, 1, &Data);
ASSERT_EFI_ERROR (Status);
return Data;
}
```

```
#/**
Writes memory-mapped registers in the EFI system memory space.
Writes the MMIO registers specižed by Address with registers width a
value specižed by Width
and Data respectively. Data is returned. If such operations are
supported, then ASSERT().
This function must guarantee that all MMIO read and write operation
are serialized.
@param Address The MMIO register to read.
The caller is responsible for aligning the Address
if required.
@param Width The width of the I/O operation.
@param Data The value to write to the I/O port.
@return Data read from registers in the EFI system memory space.
**/
UINT64
EFIAPI
MmioWriteWorker (
IN UINTN Address,
IN EFI_CPU_IO_PROTOCOL_WIDTH Width,
IN UINT64 Data
)
{
EFI_STATUS Status;
Status = mCpuIo->Mem.Write (mCpuIo, Width, Address, 1, &Data);
ASSERT_EFI_ERROR (Status);
return Data;
}
```

;1415#JSLR

#+4,#;# #JSLR#

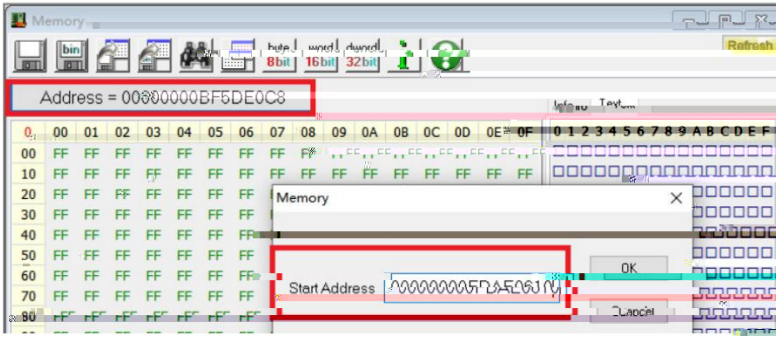


GPIO1 ч 0xFD6B0A20
GPIO2 ч 0xFD6B0960
GPIO3 ч 0xFD6B0940
GPIO4 ч 0xFD6B0A60
GPIO5 ч 0xFD6B0A50
GPIO6 ч 0xFD6B0A40
GPIO7 ч 0xFD6B0A30
GPIO8 ч 0xFD6B0950

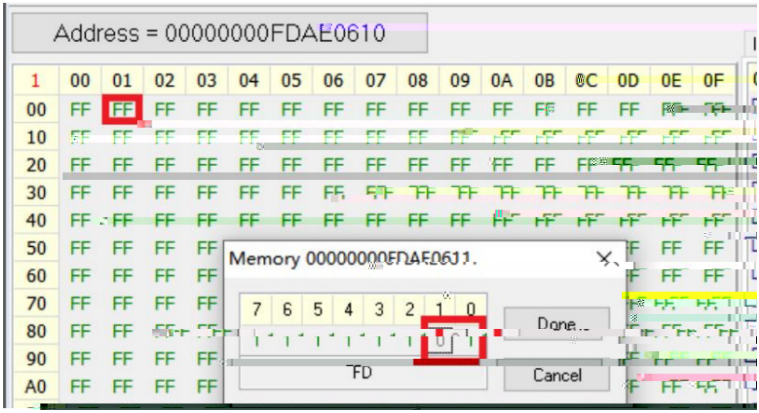
+5,# #JSLR4#

UINT32 Data32Ђ
JSLR#
22d1# #JSLR4#
Data32 = MmioRead32(0xFD6B0A20)[redacted]
MmioWrite32Ў0xFD[redacted]FFFFFDFF[redacted]

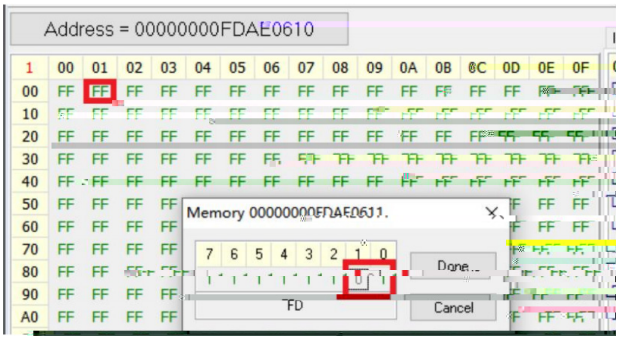
Ÿ2ÿ dAddressd h dStart Addressd GPIO
h dOKd h GPIO Ꮒ



Ÿ3ÿ GPIO Ꮒ GPIO Ꮒ
#JSLR# 00 h01 h h bit1
0hbit0 1 h dDoned GPIO g



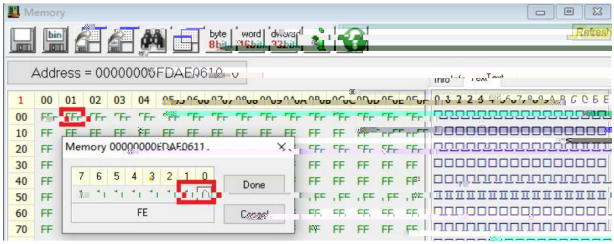
#JSLR# 00 h00 h bit1 h
0h GPIO Ꮒ 1h GPIO g



Ÿ 4 ÿ GPIO Hu g

¡ #JSLRzC - 6 / 5 G EÎ/ p¿ 00 h 01 h h bi t 1

1 h bi t 0 0 h d Done d GPIO g



#JSLR# 00 h 00 h bi t 0

0 h d Done d h GPIO H bi t 0 1 h d Done d h GI

g

43

h G
h h
h G

44

4414#

		/
		1Hυ 2Hυ 12
	d d	
		G h
		1Hυ 2Hυ 3Hυ h
		ÿ ÿ Hυ
		ÿ Winpe ÿ h h
IH0 h		PCI ISA Hυ Hυ h
	BIOS	h BIOS Setuph BIOS Setup
BIOS h		
Boot device not h found	BIOS h	BIOS d d (Boot) h
USB	USB 2.0 h USB 2.0	USB 2.0
	USB	1Hυ USB Legacy Support ÿ USBÿ 2Hυ h USB