11th Gen Intel® Core™ Processor Family for Internet of Things (IoT) Applications (Formerly Tiger Lake UP3)

HDMI-IN Enablement

Windows* Long-Term Servicing Channel (LTSC)

User Guide

April 2022

Document Number: 729027-1.0

Contents

Figures

Tables

Revision History

Date	Revision	Description
April 2022	1.0	Initial release.

§

1.0 Introduction

This document is a design guide to enable HDMI-IN function on 11th Gen Intel® Core™ Processor Family for Internet of Things (IoT) Applications (Formerly Tiger Lake UP3) platform with LONTIUM LT6911UXC chip for Windows* LTSC OS. Below features will be supported if the correct configurations (Hardware, BIOS, Lontium firmware, OS and driver) are meet.

- Video
  - Format

HDMI Signal (Input)	Video Format (MIPI Output)
RGB888	YUV422 8bits / YUY2 OR YUV420 Legacy 8bit / i420
YUV444 8bits	
YUV422 8bits	
YUV422 10bits	

  - Resolution and Refresh rate
    Up to 4k60 with support 720p, 1080p, 2K, 4K in both 30 and 60
- Audio

  PCM Audio 48KHz – 2 channels via i2s

The following section will introduce how to enable HDMI-IN function step by step.

1.1 Terminology

Table 1. Terminology

Term	Description
CSI	Camera Serial Interference
i2s	Inter-IC Sound, A data interface
I2C	Inter-Integrated Circuit Protocol
ISP	Image Signal Processor
LTSC	Long-Term Servicing Channel

1.2 Reference Documents

Log in to the Resource and Documentation Center (rdc.intel.com) to search for and download the document numbers listed in the following table. Contact your Intel field representative for access.

Note: Third-party links are provided as a reference only. Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Table 2. Reference Documents

Document	Document No./Location
Tiger Lake UP3 UP4 H35 UP3 Refresh Platform Design Guide	607872
11th Generation Intel® Core™ Processor External Design Specification (EDS), Volume 1 of 2	575683
Tiger Lake UP3/UP4/UP3-Refresh/ H35/H35-Refresh Processor External Design Specification Volume 2a of 2	575681
Intel® 500 Series Chipset Family On-Package Platform Controller Hub External Design Specification, Volume 1 of 2	576591
Intel® 500 Series Chipset Family On-Package Platform Controller Hub External Design Specification Volume 2 of 2	575857

§

2.0 HDMI-IN Function Enablement

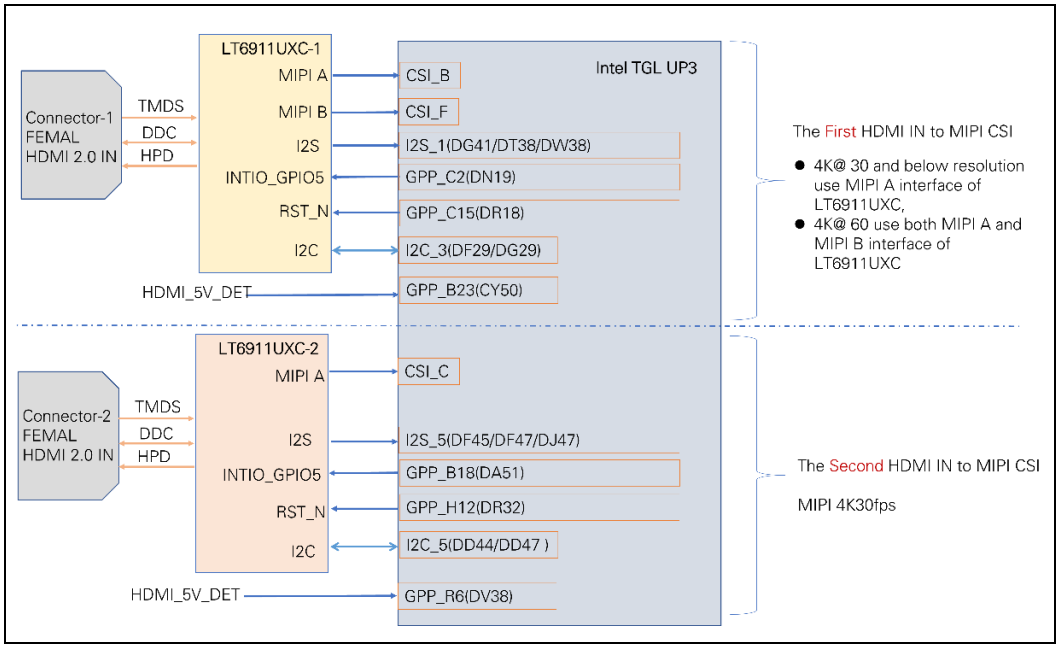
This chapter will introduce HDMI-IN function enablement details, including hardware design, BIOS, Lontium firmware and OS/driver installation.

2.1 Hardware Design

A brief hardware architecture diagram is provided with two HDMI-IN ports (HDMI-IN port1 supports 4K60 and HDMI-IN port2 supports 4K30), you can refer to the attached *TGL UP3_HDMI IN_TO_MIPI_CSI.pdf* for a detail schematic design of one HDMI-IN port.

Note: ONLY 11th Gen Intel® Core™ processors with IPU6 can support HDMI-IN functionality.

Figure 1. HDMI-IN Hardware Architecture Diagram



2.2 BIOS

2.2.1 BIOS Configurations

In this section we provide a reference BIOS configuration for HDMI-IN enablement on Tiger Lake-U RVP, there are three main configuration items, which reference to above hardware design architecture (two HDMI-IN ports) and the configuration is combined with MR4 driver (the BIOS configurations may vary for other driver versions):

- Configuring Serial IO
- MIPI Camera Configuration
- HD Audio

Power on the board and press F2/Del to boot to BIOS setup menu to config. If your BIOS does not have some following settings, contact IBV to update BIOS source code. Refer to the next chart on change summary to enable HDMI-IN function in BIOS source code.

2.2.1.1 Configuring Serial IO

Intel Advanced Menu → PCH-IO Configuration → Serial IO Configuration

Enable both I2C3 Controller and I2C5 Controller, I2C3 is for HDMI-IN port 1 and I2C5 is for HDMI-IN port 2.

Figure 2. Serial IO Configuration

2.2.1.2 MIPI Camera Configuration

Intel Advanced Menu → System Agent (SA) Configuration → MIPI Camera Configuration

Figure 3. MIPI Camera configuration

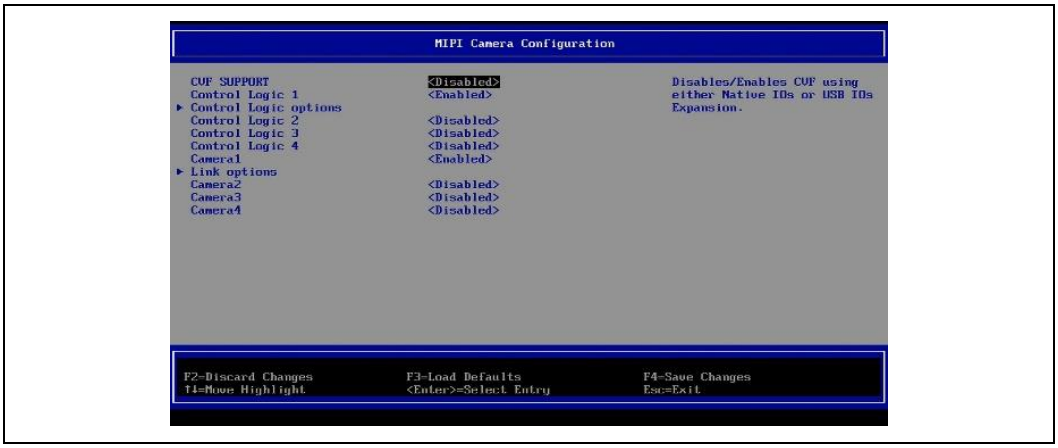
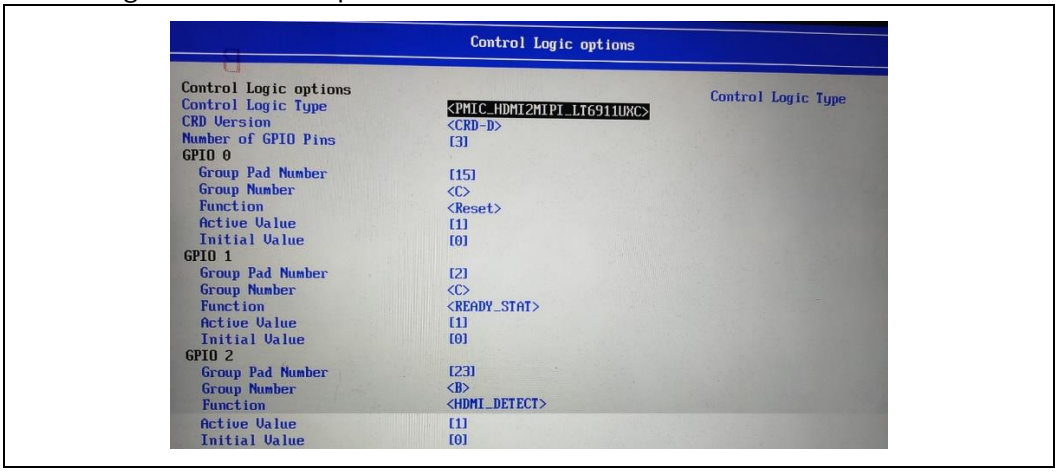


Figure 4. Control Logic options Configuration

Control logic 1 for HDMI-IN port1



Control logic 2 for HDMI-IN port2

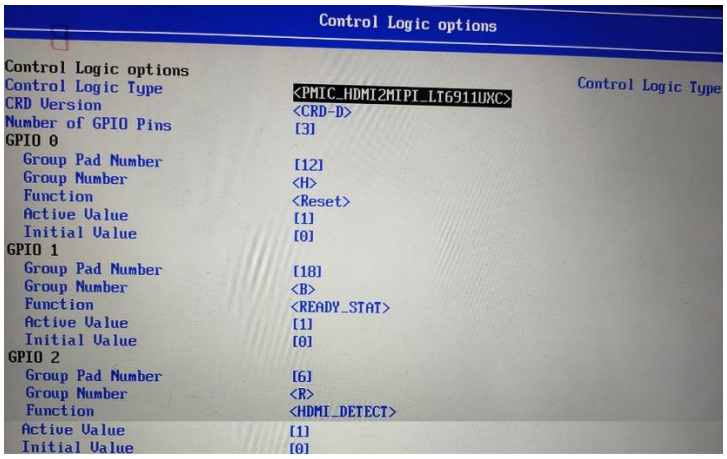
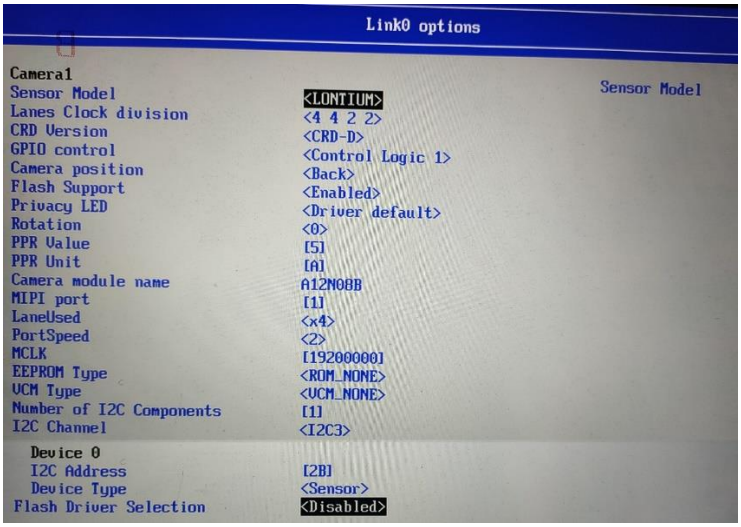
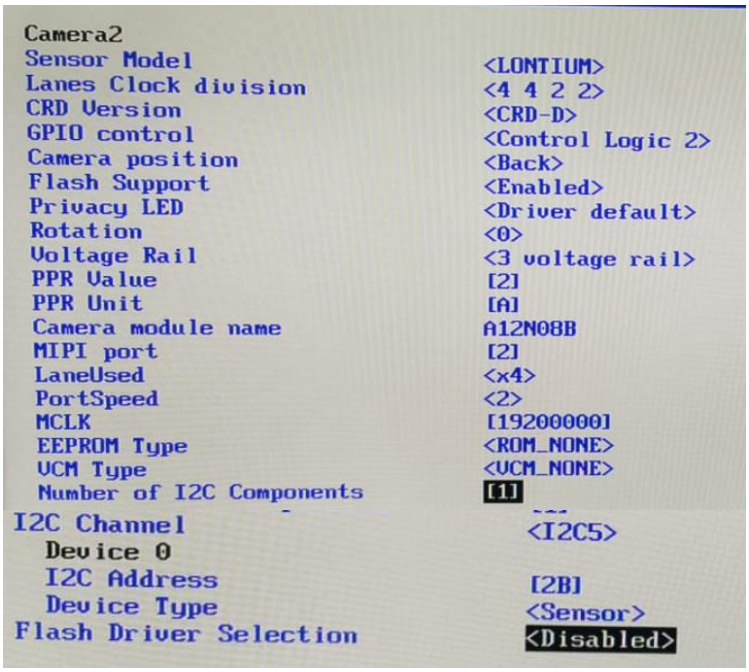


Figure 5. Link options Configuration

Link0 for HDMI-IN port1



Link1 for HDMI-IN port2



Note: For two MIPI ports use case (4K60), MIPI port used to configure the ID of left port, PPR value used to configure the ID of right port. For single MIPI port use case, the value of MIPI port and PPR value should be same.

2.2.1.3 HD Audio Configuration

Below settings enable HDMI-I2S only on audio side. Other settings are also required to enable other codec and audio endpoints.

HD Audio Configuration

HD Audio → Enabled

SSP #1 → select (for I2S1)

SSP #5 → select (for I2S5)

HD Audio DSP Features Configuration

i2s → Enabled

i2s Codec Select → LONTIUM

Figure 6. HD Audio Configuration

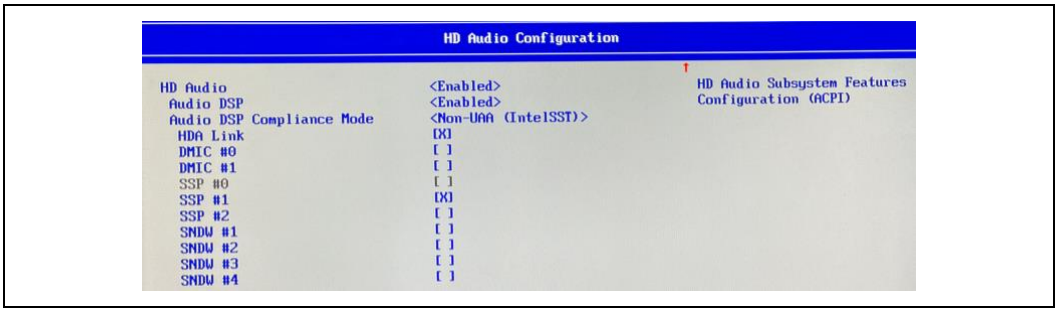
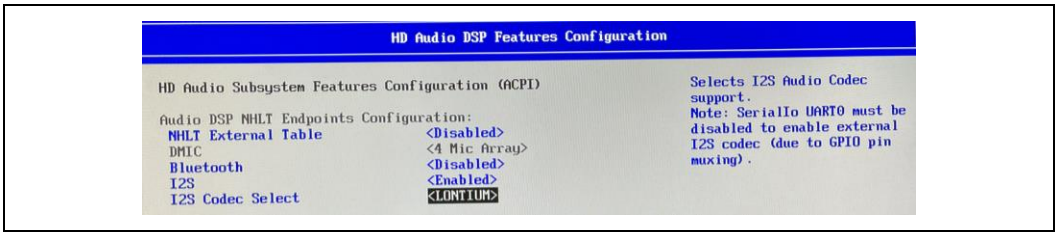


Figure 7. HD Audio DSP Features Configuration



2.2.2 BIOS Source Code Changes

Intel reference BIOS code with HDMI-IN support has been released to IBV, contact IBV to get the latest BIOS source code for implement. Here are the key changes in BIOS source code to support HDMI-IN function.

2.2.2.1 Control logic

BIOS configuration is added in on top of the existing MIPI control logic configuration to provide additional index for the OS to detect. Such as:

1. GPIO Pin Ready Status – 0x13
2. GPIO Pin HDMI Detect – 0x14
3. Control Logic Type LT6911UXC – 0x5
4. Control Logic CRD version LT6911UXC_V1 – 0x60

Existing MIPI GPIO configuration do not have input pin configuration. This change added in Input Pin configuration in functions “MipiCamGpioInit”. GPIO pins are set as below:

```
MipiCamConfigureGpio (
  IN GPIO_CONFIG *GpioConfig,
  IN UINT8 GpioFunctions,
  IN UINT8 GpioOutputState
  )
{
  ZeroMem(GpioConfig, sizeof(GPIO_CONFIG));

  if ( GpioFunctions == 0x13 ) { /* Ready Status 0x13 */
    DEBUG((DEBUG_INFO, "MIPI AOB Ready Status Input Pin\n"));
    GpioConfig->PadMode = GpioPadModeGpio;
    GpioConfig->HostSoftPadOwn = GpioHostOwnGpio;
    GpioConfig->Direction = GpioDirIn; /* Input Pin */
    GpioConfig->OutputState = GpioOutDefault;
    GpioConfig->InterruptConfig = GpioIntEdge | GpioIntApic;
    GpioConfig->PowerConfig = GpioPlatformReset;
    GpioConfig->ElectricalConfig = GpioTermDefault;
    GpioConfig->LockConfig = GpioPadUnlock;
  } else if ( GpioFunctions == 0x14) { /* / 0x14 */
    DEBUG((DEBUG_INFO, "MIPI AOB HDMI Detect Input Pin\n"));
    GpioConfig->PadMode = GpioPadModeGpio;
    GpioConfig->HostSoftPadOwn = GpioHostOwnGpio;
    GpioConfig->Direction = GpioDirIn; /* Input Pin */
    GpioConfig->OutputState = GpioOutDefault;
    GpioConfig->InterruptConfig = GpioIntBothEdge | GpioIntApic;
    GpioConfig->PowerConfig = GpioPlatformReset;
    GpioConfig->ElectricalConfig = GpioTermDefault;
    GpioConfig->LockConfig = GpioPadUnlock;
  } else {
```

ACPI devices LTH0,1,2,3,4,5 is added in “MipiCamSeasors.asl” to separate from the older MIPI ACPI devices. This device exposes the GPIO, Logic Type, and CRD version to the OS for the OS to use.

```
Device (LTH0) { // Control Logic  HDMI 0 is selected
  Name(_ADR, Zero)
  Name(_HID, "INT3472")
  Name(_CID, "INT3482")
  Name(_DDN, "PMIC-LTH0")
  Name(_UID, 0)

  Method(_CRS, 0x0, NotSerialized) {
    Store (ResourceTemplate() {}, Local0)
    // Create GPIO resource template using as parameters data provided in the BIOS setup
    If (LGreater (C0GP, 0)) {
      ADBG(Concatenate("Lontium HDMI 0 Input pin #1 is selected !!", ToHexString (C0F0)))
      Switch(C0F0) {
        case (0x13)  { Store (PINJ(C0P0, C0G0), Local1) } /* Ready Status 0x13 */
        case (0x14)  { Store (PINI(C0P0, C0G0), Local1) } /* HDMI detect 0x14 */
        Default      { Store (PINR(C0P0, C0G0), Local1) }
      }
      ConcatenateResTemplate(Local0, Local1, Local2)
      Store(Local2, Local0)
    }
    If (LGreater (C0GP, 1)) {
      ADBG(Concatenate("Lontium HDMI 0 Input pin #2 is selected !!", ToHexString (C0F1)))
      Switch(C0F1) {
        case (0x13)  { Store (PINJ(C0P1, C0G1), Local1) } /* Ready Status 0x13 */
        case (0x14)  { Store (PINI(C0P1, C0G1), Local1) } /* HDMI detect 0x14 */
        Default      { Store (PINR(C0P1, C0G1), Local1) }
      }
      ConcatenateResTemplate(Local0, Local1, Local2)
```

2.2.2.2 Camera link

4K60 camera is required to use 2 MIPI port to support for its high bandwidth, existing configuration is not able to support that, therefore it is required to update the interface. The BIOS used the Sensor Model “LONTIUM” to differentiate it from the legacy setting. PPR value “5” (MIPI port 5) is used as a workaround to indicate the second MIPI interface that associates with the MIPI port “1” configuration (this setting is included in the Initialize IPU PreMem Configuration).

```
if (SetupData.MipiCam_Link0 == 1) {
  UPDATE_POLICY (((FSPM_UPD *) FspmUpd)->FspmConfig.IpuLaneUsed[SetupData.MipiCam_Link0_DriverData_LinkUsed],\
                 IpuPreMemPolicy->LaneUsed[SetupData.MipiCam_Link0_DriverData_LinkUsed],  SetupData.MipiCam_Link0_DriverData_LaneUsed);
  UPDATE_POLICY (((FSPM_UPD *) FspmUpd)->FspmConfig.CsiSpeed[SetupData.MipiCam_Link0_DriverData_LinkUsed],\
                 IpuPreMemPolicy->CsiSpeed[SetupData.MipiCam_Link0_DriverData_LinkUsed],  SetupData.MipiCam_Link0_DriverData_CsiSpeed);

  if( SetupData.MipiCam_Link0_ == MIPICAM_LONTIUM) {
    if( SetupData.MipiCam_Link0_DriverData_PprValue != SetupData.MipiCam_Link0_DriverData_LinkUsed) {
      if( SetupData.MipiCam_Link0_DriverData_PprValue < MAX_CSI_PORT ) {
        UPDATE_POLICY (((FSPM_UPD *) FspmUpd)->FspmConfig.IpuLaneUsed[SetupData.MipiCam_Link0_DriverData_PprValue],\
                       IpuPreMemPolicy->LaneUsed[SetupData.MipiCam_Link0_DriverData_PprValue],  SetupData.MipiCam_Link0_DriverData_Lane
        UPDATE_POLICY (((FSPM_UPD *) FspmUpd)->FspmConfig.CsiSpeed[SetupData.MipiCam_Link0_DriverData_PprValue],\
                       IpuPreMemPolicy->CsiSpeed[SetupData.MipiCam_Link0_DriverData_PprValue],  SetupData.MipiCam_Link0_DriverData_CsiS
      } else {
        DEBUG ((DEBUG_ERROR, "LONTIUM 2MIPI port Invalid (PprValue %d) < %d\n", MIPICAM_LONTIUM, SetupData.MipiCam_Link0_DriverData_PprVal
        //ASSERT(FALSE);
```

2.2.2.3 HD Audio Configuration

New Audio codec devices type has added in for the user to select in RVP.

5. Lontium – 0x3

```
Method(_INI) {
  If (LEqual(I2SC, HDAC_I2S_ALC274)) {
    Store ("INT34C2", _HID)
    Store ("INT34C2", _CID)
    Store (0x1C, CADR)
  } ElseIf (LEqual(I2SC, HDAC_I2S_ALC1308)) {
    Store ("10EC1308", _HID)
    Store ("10EC1308", _CID)
    Store (0x0010, CADR)
  } ElseIf (LEqual(I2SC, HDAC_I2S_LONTIUM)) {
    Store ("INTC10B0", _HID)
    Store ("INTC10B0", _CID)
    Name (_STA, 0x0F)
  }
}
```

Blob file shown below are the binary setting required by the firmware to initialize the SSP port. You are only required to change if the configurations of channel, frequency, bits and related settings have changed.

```
// I2S/SSP Configuration BLOBs
// Audio Format and Configuration details
//
// Frequency: 48000kHz, PCM resolution: 24(32) bits
// TDM slots: 2
// Codec: Realtek Lontium, mode: slave
GLOBAL_REMOVE_IF_UNREFERENCED
CONST UINT32 I2sLontiumCapture2ch48kHz24bitFormatConfig[] =
{
  0x00000000, 0xFFFFFF10, 0xFFFFFFFF, 0xFFFFFFFF,
  0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF, 0xFFFFFFFF,
  0xFFFFFFFF, 0xC1D00077, 0xD3700000, 0x00000000,
  0x0A200200, 0x00000003, 0x00000003, 0x00004002,
  0x00000000, 0x07070F00, 0x00000000, 0x00000000,
  0x00000FFF, 0x00000000, 0x00000008, 0x00000001,
  0x00000001
};
```

Additional changes are the NHLT table, which is required by the Audio DSP to configure the Audio endpoint. Below table describes the endpoint to the OS. It consists of information such as VirtualBUSID, which indicate the SSPx is being used, and direction indicates in Capture device.

Figure 8. Blob Configuration for I2S1(HDMI-IN port1)

```
GLOBAL_REMOVE_IF_UNREFERENCED
ENDPOINT_DESCRIPTOR  HdaEndpointI2s1Capture = {
  0,                 // EndpointDescriptorLength
  HdaNhltLinkSsp,    // LinkType
  1,                 // InstanceId
  0x8086,           // HwVendorId
  0xae34,           // HwDeviceId
  1,                 // HwRevisionId
  1,                 // HwSubsystemId
  HdaNhltSspDeviceI2s,  // DeviceType
  1,                 // Direction
  1,                 // VirtualBusId
  { 0 },             // EndpointConfig
  { 0 },             // FormatsConfig
  { 0 }              // DevicesInformation
};
```

Figure 9. Blob Configuration for I2S5 (HDMI-IN port2)

```
GLOBAL_REMOVE_IF_UNREFERENCED
ENDPOINT_DESCRIPTOR  HdaEndpointI2s5Capture = {
  0,                 // EndpointDescriptorLength
  HdaNhltLinkSsp,    // LinkType
  1,                 // InstanceId
  0x8086,           // HwVendorId
  0xae34,           // HwDeviceId
  1,                 // HwRevisionId
  1,                 // HwSubsystemId
  HdaNhltSspDeviceI2s,  // DeviceType
  1,                 // Direction
  5,                 // VirtualBusId
  { 0 },             // EndpointConfig
  { 0 },             // FormatsConfig
  { 0 }              // DevicesInformation
};
```

2.3 Lontium Firmware

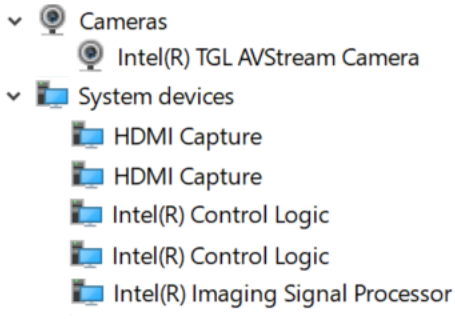
The Lontium firmware used on Intel RVP can be shared to customer for “Test purpose only” (contact your Intel representative to get the test firmware). Customer needs to get production firmware from Lontium directly.

2.4 OS and Driver Installation

MIPI Camera driver and i2s Audio driver are needed to enable HDMI-IN function. Those drivers are OS dependent. Install the correct OS version for HDMI-IN function verification (Follow the latest BKC to install drivers for all devices).

2.4.1 MIPI Camera Driver Installation

"iacamera64", "iaisp64", "iactrllogic64", "lt6911uxc" are needed to enable HDMI Capture Device (Serial IO driver should be installed first for control logic driver). After the installation, these items can be seen in Device Manager.

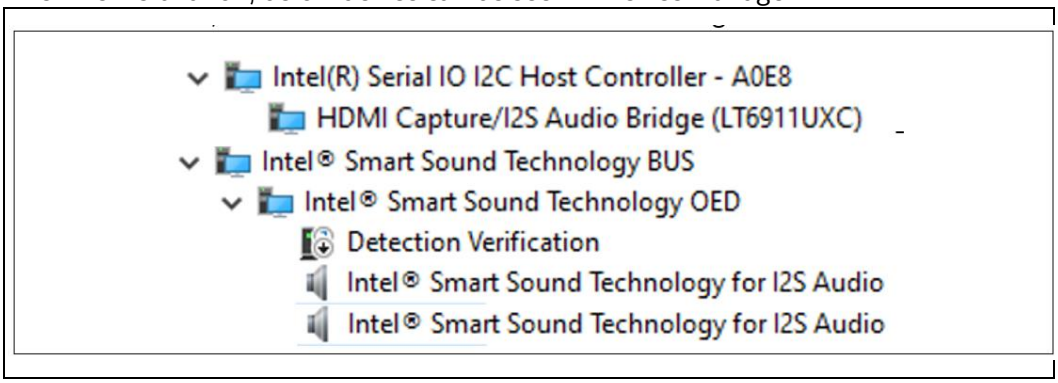


2.4.2 i2s Audio Driver Installation

Install SST driver first, then install the two Lontium i2s Audio driver:

- Install *LT6911Au.inf*
- Install *LT6911Au_CfgExt.inf*

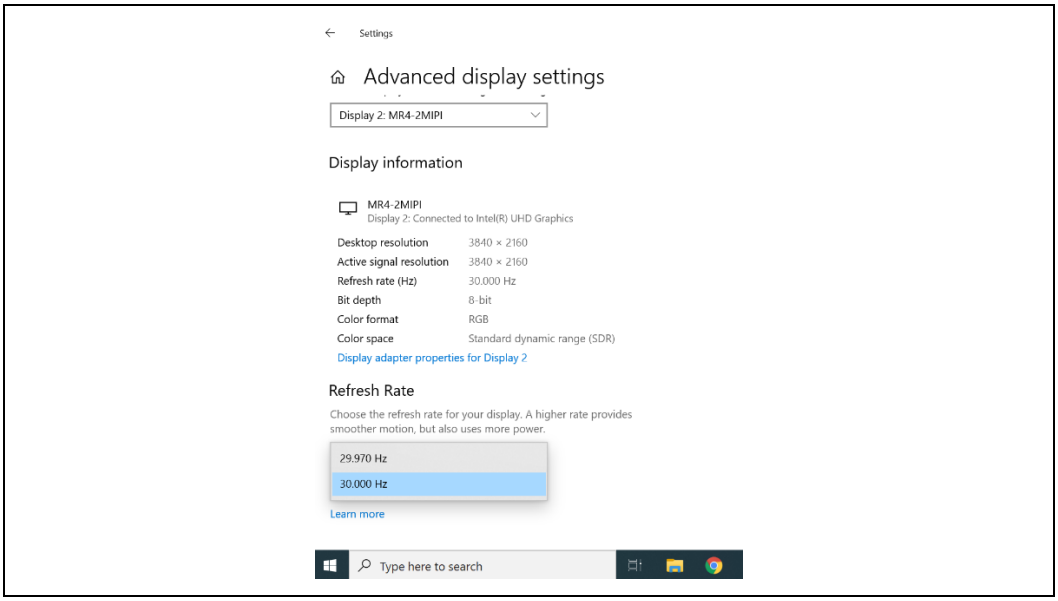
After the installation, below device can be seen in Device Manager.



2.5 HDMI-IN Video and Audio Verification

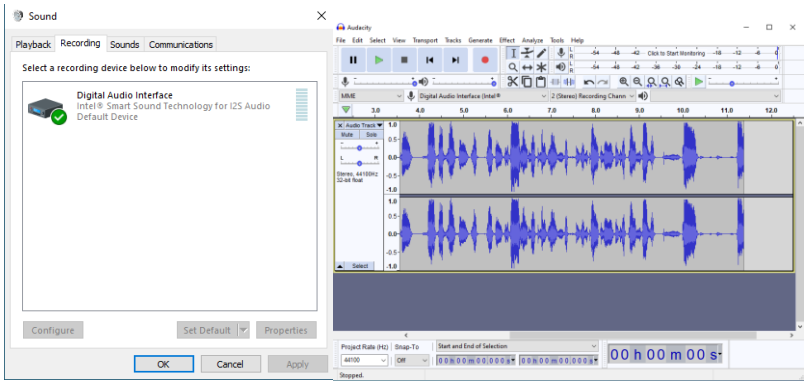
For HDMI-IN Video verification, you can run Windows Camera app for video capture and the HDMI-IN device will be detected by the source as a display. Make sure that the HDMI-IN source is compatible with the supported format.

Figure 10. HDMI-IN Video



To verify the HDMI-IN Audio, make sure that the HDMI-IN i2s Audio device is selected and using recording app to record the HDMI-IN audio for replay.

Figure 11. HDMI-IN i2s Audio



§