

1 引言

本应用笔记描述了如何实现高清晰度多媒体接口(HDMI)-消费者电子控制(CEC)功能作为电视机或投影仪的控制接口，实现了基于 LPC5500 系列的 HDMI-CEC 协议，该实现使用了 GPIO 和 SCT 模块。

CEC 是一种单总线协议，允许 AV 产品在一个系统之间发现设备和通信。CEC 使各个通过 HDMI 连接的设备互操作成为可能，从而减少系统基本操作所需的 IR 遥控器和按键的数量。

HDMI-CEC 协议开销非常少，一般的低成本 MCU 就可以实现。

一般来说，CEC 总线允许系统中的所有设备发现并相互通信。此外，它还支持 HDMI 接口系统中的全局简化（单个远程）系统控制。

更多详情，请参见 [HDMI](#)。

本应用说明不包括 HDMI-CEC 协议的高级描述，但侧重于 MCU 上的物理级实现。

目录

1	引言.....	1
1.1	术语.....	1
2	HDMI-CEC.....	2
2.1	概述.....	2
2.2	物理层协议.....	2
2.3	块级协议.....	4
2.4	帧级协议.....	5
2.5	设备连接和寻址.....	5
2.6	CEC 消息描述.....	6
3	协议实现.....	6
3.1	概述.....	7
3.2	硬件连接.....	7
3.3	时序器捕获功能.....	8
4	示例和测试.....	10
5	结论.....	11
6	修订记录.....	12

1.1 术语

表 1. 术语

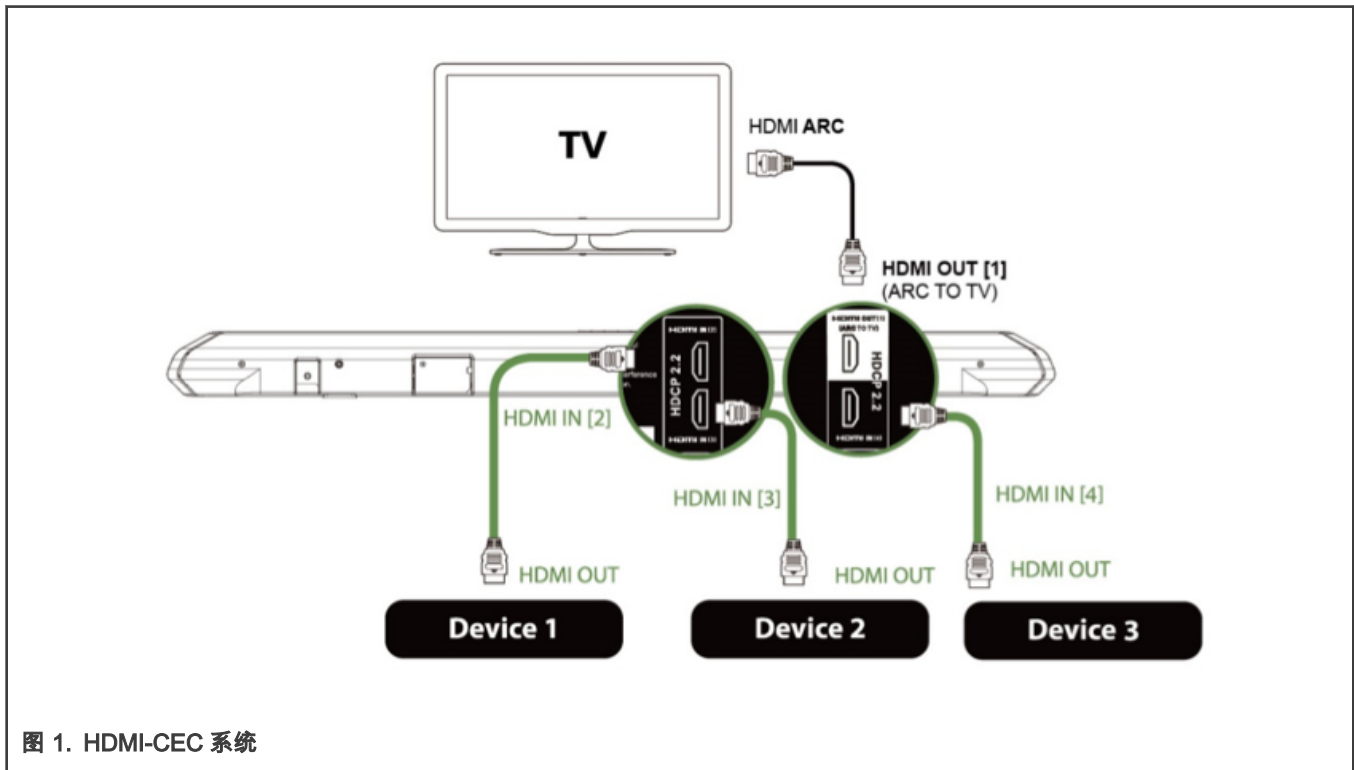
术语	说明
HDMI	高解析度多媒体接口
(HDMI) Source	具有 HDMI 输出能力的设备
(HDMI) Sink	具有 HDMI 输入能力的设备
Initiator 发起者（主机）	准备或者正在发送 CEC 数据的设备，一般会有一个从机响应
Follower 跟随者（从机）	收到 CEC 数据并响应的设备
CEC	消费电子控制
Broadcast Message 广播消息	发布到逻辑地址 15 的消息，所有设备都可以收到
TV	可以输入 HDMI 信号并显示图像影音信息的设备，一般来说，没有 HDMI 输出能力
Destination 目标设备	接收 CEC 消息的设备
Source Device 源设备	可以通过 HDMI 提供 AV 流的设备
Logical address 逻辑地址	分配给每个设备的唯一的标识地址



2 HDMI-CEC

2.1 概述

CEC 总线是一条单总线协议，通过标准 HDMI 电缆连接多达 10 个 AV 设备。该 CEC 协议包括物理地址（拓扑）发现、（基于设备的）逻辑寻址、仲裁、重传、广播和路由控制的自动机制。图 1 显示了 HDMI 系统的一个典型应用，其中所有 HDMI 设备都通过 HDMI 电缆串行连接在一起。HDMI-CEC 总线连接多达 10 个 HDMI AV 设备。



2.2 物理层协议

通讯总是在发起者和一个（或多个）接受者之间进行。initiator 和 follower 都可以写入一位。发起者写入一位提供数据，而跟随者发起位提供确认。物理层通信速率小于 500 位/秒。这些消息以一个长的开始位开始，然后紧接着是一些较短的数据位。当 CEC 总线空闲时，总线总是被拉到 VCC。

2.2.1 起始位

起始位脉冲格式如图 2 所示。它识别一个新的下降沿开始，一个低脉冲加上一个高脉冲被限制在一定的持续时间内。

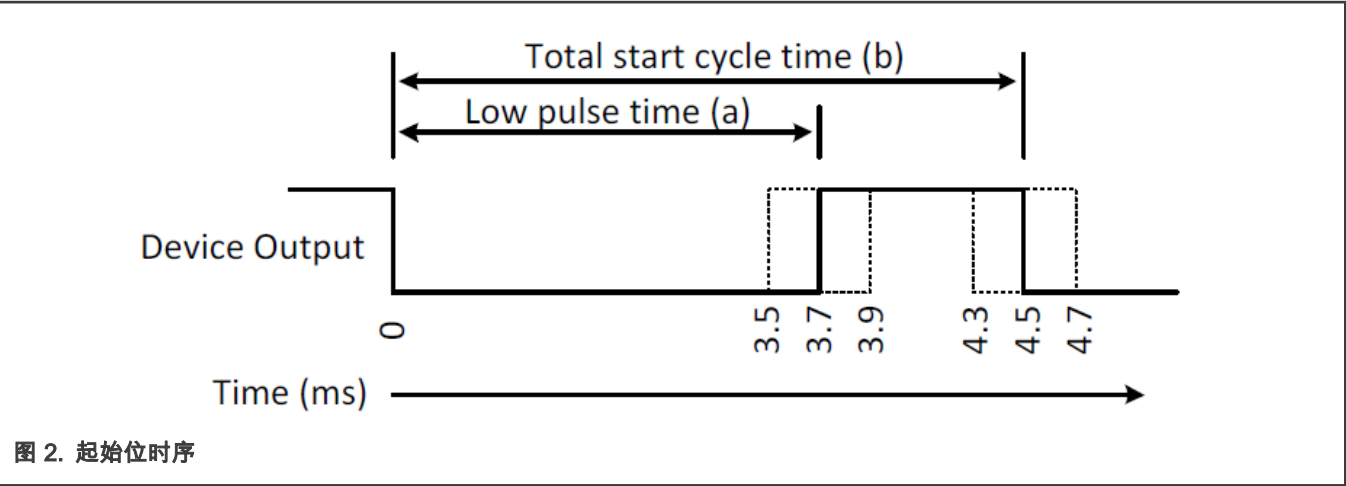


图 2. 起始位时序

2.2.2 数据位时序

数据位后面跟着开始位，每个数据位由低保持时间和高保持时间组成。数据位结束时的高到低转换是下一个数据位的开始，只有当有以下数据位时才会发生。在发送最终位后，CEC 线保持高，如 图 3 所示。

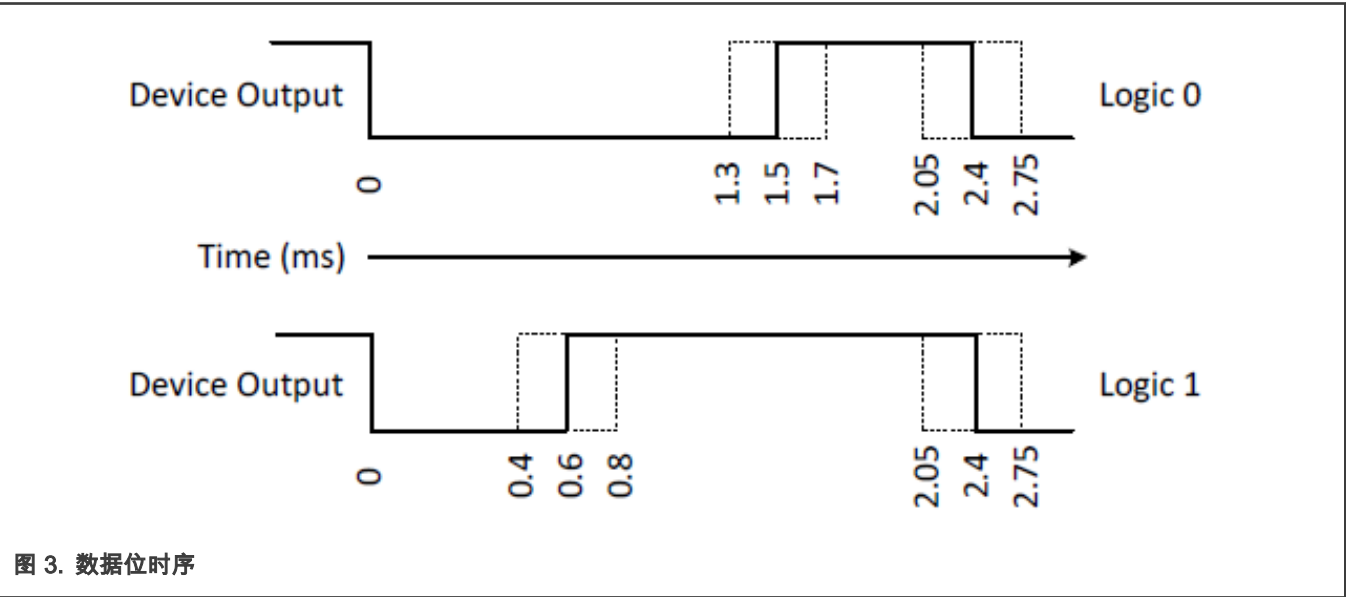
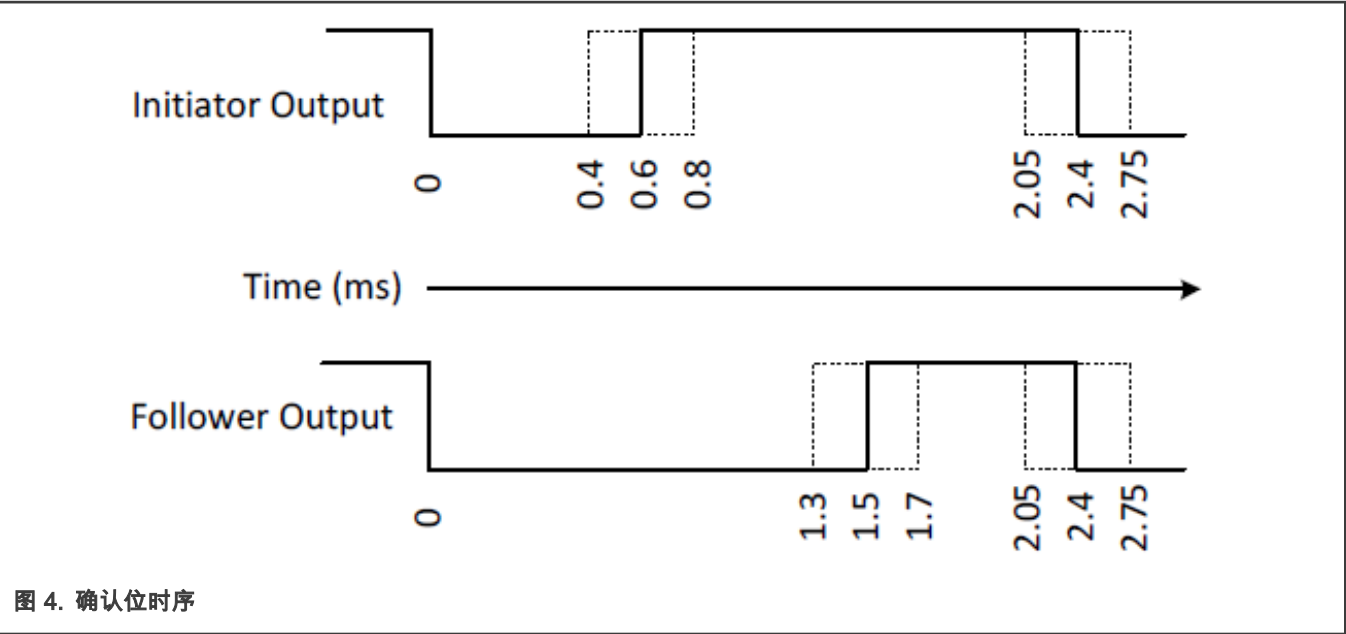


图 3. 数据位时序

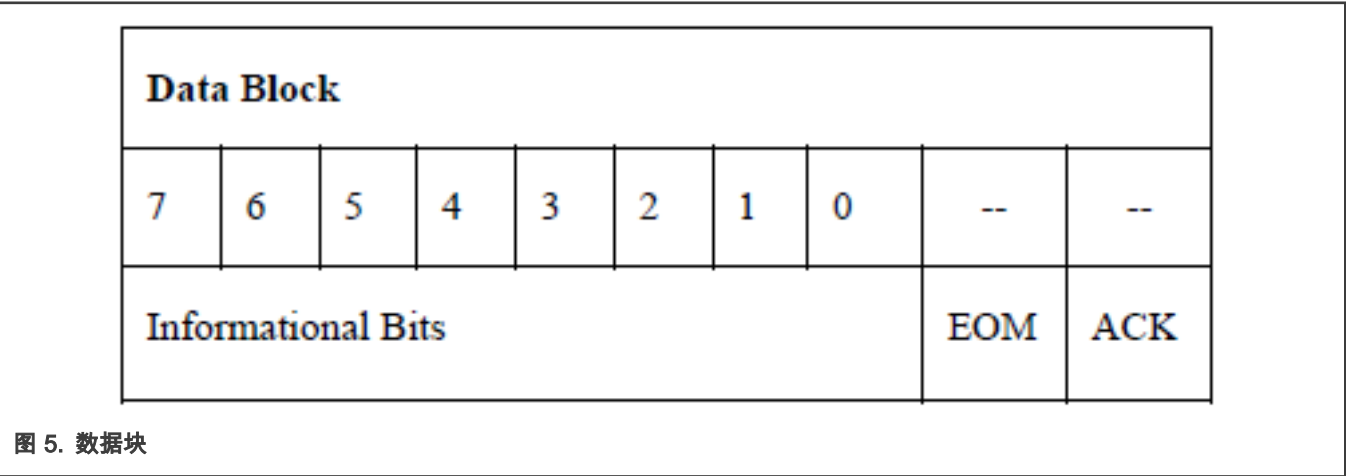
2.2.3 确认比特时间

确认位也为写入一位，数据位由发起者发送，跟随者可以将总线拉低到逻辑 0 以确认数据块。发起者输出逻辑 1，因此允许跟随者通过在安全安全采样时间内改变 CEC 状态，并将确认(ACK)位从逻辑 1 变为逻辑 0，如 图 4 所示。



2.3 块级协议

位分组为 10 位标头为一个数据块。头块和数据块都包括 8 位数据以及消息结束 (EOM) 和 ACK 位。EOM 位为消息中发出最终块的信号。一个'0'表示一个或多个块跟随，一个'1'表示消息已经完成。



- EOM :
 - 0 : 一个或多个数据块跟随。
 - 1 : 消息完成，不再提供数据。
- ACK
 - 0 : 接受者请求信息。
 - 1 : NACK。

头块与数据块相同。它后面紧跟着开始位，作为消息中的第一个“数据块”。头块的 8 个数据位指示发起者地址和目标地址，如图 6 所示。

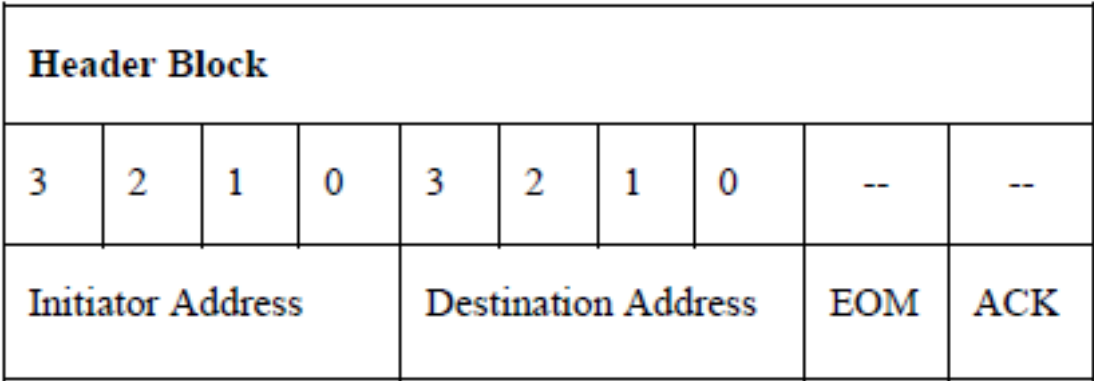


图 6. 头块

2.4 帧级协议

使用帧发送 HDMI-CEC 消息。每个 CEC 框架由一个开始位、一个头块和可选的数据块组成。一个 HDMI-CEC 帧的例子如 图 7 所示。

第一个数据块后面是头块，称为操作码，然后是零，一个或多个数据块，指示传输的实际数据。



图 7. HDMI-CEC 框架

2.5 设备连接和寻址

为了允许 CEC 寻址特定的物理设备和控制开关，所有设备都应具有物理地址。每当系统中添加新设备时，这个设备必须拥有唯一的设备地址。

物理地址发现过程只使用 DDC/EDID 机制，可以应用于所有 HDMI 集线器和设备。

将 CEC 线直接连接到网络上的所有节点。在发现自己的物理地址后，CEC 设备将它们的物理和逻辑地址发送给所有其他设备，因此，允许系统创建一个设备连接地图。

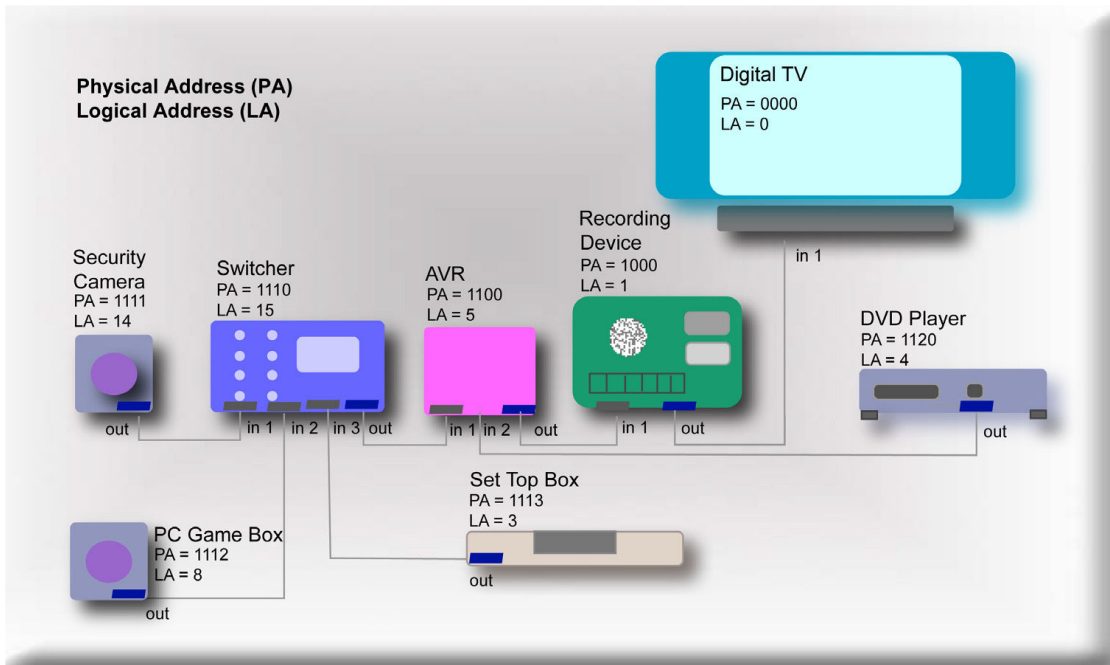


图 8. HDMI 系统中的物理地址

2.6 CEC 消息描述

表 2 为 CEC 消息中最常用的命令。有关更多操作码、参数和描述或其他消息，请参见 HDMI-CEC 规范。

表 2. CEC 消息

操作码	值	说明	参数	应答
<Polling Message>	—	任何设备，可用于设备发现	无	发送应答
<Give Physical Address>	0x83	发送给设备，设备返回物理地址	无	报告物理地址
<Report Physical Address>	0x84	发送给所有设备	物理地址和设备类型	—
<Active Source>	0x82	发送给一个新的源设备并且传输流数据	物理地址	当前在线的设备会采取相应的应用层操作
<Image View On>	0x04	从一个源设备发送给 TV 并进入运行模式	无	打开文本显示模式或者进入显示图像状态
<Give Device Power Status>	0x8F	用来获取当前功耗状态	无	报告功耗状态
<Report Power Status>	0x90	用于告知当前功耗状态	功耗状态	—

3 协议实现

3.1 概述

本节介绍如何使用 LPC5500 系列实现 CEC 总线协议的物理层。该 LPC5500 系列是 Arm® Cortex®基于 M33 的嵌入式应用微控制器。这些设备包括高达 320 KB 的片上 SRAM 和高达 640 KB 的片上闪存。它有 SCT 定时器，可用于捕获外部总线边缘变化。我们使用这个特性来捕获 CEC 总线中的上升和下降时间。

表 3. 单片机外设资源使用

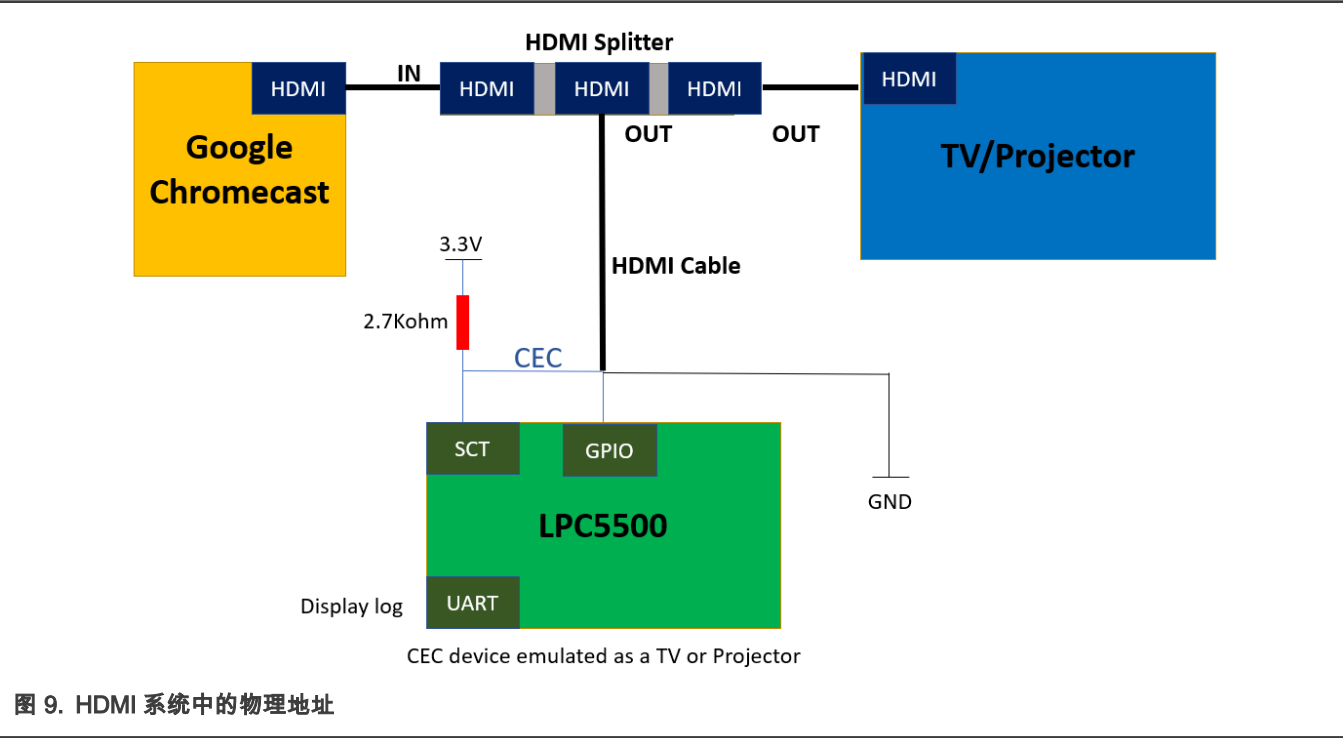
使用 IP	说明
SCT	捕捉边缘时间戳
GPIO	输出和输入 CEC 数据
UART	显示日志

3.2 硬件连接

为了 demoLPC5500 系列的 CEC 功能，使用 Chromecast 作为发起者，并使用 HDMI 分线盒来开发系统硬件。demo 的系统结构如 图 9 所示。

注意
单片机模拟电视的 CEC 控制器。

将 HDMI-CEC 引脚连接到 LPC5500 系列上的两个引脚上，一个用作 GPIO 输出，另一个用作 GPIO 输入和 SCT 输入。还启用了 SDK 中的默认调试控制台进行日志显示。



HDMI 连接器中显示 CEC 引脚和 GND 引脚定义，如 图 10 所示。

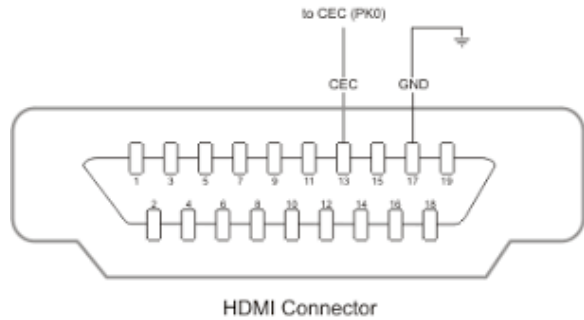


图 10. HDMI 连接器中的 CEC 引脚和 GND 引脚定义

3.3 时序器捕获功能

在 CEC 协议中实现的最重要的功能是输入捕获特性。此功能可通过 SCT 模块实现。

SCTimer/PWM (SCT) 是 NXP 半导体特有的外设。它可以像大多数传统时序器一样工作，但也增加了一台状态机，使其具有更高的可配置性和控制程度。这允许 SCT 被配置为多个 PWM、具有死时控制的 PWM 和具有复位能力的 PWM，以及许多其他与传统时序器不一样的配置。一旦 SCTimer/PWM 被配置，它就可以独立于微控制器核心自主运行，除非 SCTimer/PWM 中断到来，这需要 CPU 来服务中断。

本应用程序说明描述了如何将 SCT 配置为输入捕获的简单用法，它计算两个跳变电平边缘之间的时间。使用三个 SCT 事件寄存器，两个捕获寄存器，一个匹配寄存器，如 [表 4](#) 所示。

表 4. 使用的 SCT 资源

SCT 资源	说明
EVT0	配置为上升沿触发器
EVT1	配置为下降沿触发器
EVT2	配置为匹配触发器，用于超时控制
MAT0	用于超时计数
CAP0	使用 EVT0 捕捉上升沿时间
CAP1	与 EVT1 一起工作，捕捉下降边沿时间

在配置 SCT 资源寄存器之前，需要初始化 SCT、SCT 和 SET 时钟捕获和匹配寄存器。按照以下步骤：

1. SCT 输入时钟来自总线时钟，为 150 MHz。除以 150，所以 SCT 时钟为 150Mhz/150=1MHz。

```
uint32_t sctimerClock = CLOCK_GetFreq(kCLOCK_BusClk);

CLOCK_EnableClock(kCLOCK_Sct0);
CLOCK_EnableClock(kCLOCK_InputMux);

INPUTMUX->SCT0_INMUX[SCT_GPI0] = 7;

SCT0->CONFIG |= SCT_CONFIG_UNIFY_MASK;

/* config presclar
*/SCT0->CTRL &= ~SCT_CTRL_PRE_L_MASK;
SCT0->CTRL |= SCT_CTRL_PRE_L(149);
```


2. 配置 SCT 匹配和事件寄存器。在这个演示代码中，MAT0 用于超时时间设置。
3. 将匹配寄存器设置为 10000，等于 10000=100 ms。
4. 将 EVT0 设置为输入上升边缘触发器，EVT1 设置为输入下降边缘触发器。
5. 将 EVT2 设置为匹配触发器，当 SCT 时序器达到 MAT0 值时。触发 EVT2 并用作超时事件标志。

```
SCT0->MATCH[MAT_INDEX] = 10000;
SCT0->MATCHREL[MAT_INDEX] = 10000;

/* EVT0 config */
SCT0->EV[EVT0_INDEX].CTRL = SCT_EV_CTRL_MATCHSEL(MAT_INDEX) | SCT_EV_CTRL_COMBMODE(2) |
SCT_EV_CTRL_IOCOND(1) | SCT_EV_CTRL_IOSEL(SCT_GPIO); /* raising */
SCT0->EV[EVT0_INDEX].STATE = (1 << 0);

/* EVT1 config */
SCT0->EV[EVT1_INDEX].CTRL = SCT_EV_CTRL_MATCHSEL(MAT_INDEX) | SCT_EV_CTRL_COMBMODE(2) |
SCT_EV_CTRL_IOCOND(2) | SCT_EV_CTRL_IOSEL(SCT_GPIO); /* falling */
SCT0->EV[EVT1_INDEX].STATE = (1 << 0);

/* EVT2 */
SCT0->EV[EVT2_INDEX].CTRL = SCT_EV_CTRL_MATCHSEL(MAT_INDEX) | SCT_EV_CTRL_COMBMODE(1) |
SCT_EV_CTRL_IOCOND(2) | SCT_EV_CTRL_IOSEL(SCT_GPIO); /* ti */
SCT0->EV[EVT2_INDEX].STATE = (1 << 0);
```

6. 最后，执行以下杂项设置：

- a. 配置 CAP0 和 CAP1 作为输入捕获功能。
- b. 当 EVT0 和 EVT1 被触发时，重置 SCT 时序器。
- c. 启动 SCT 时序器。

注意

在此应用程序中，SCT 状态始终为 0；不使用 SCT 状态函数。

```
/* use as cap */
SCT0->REGMODE = (1<<CAP0_INDEX) | (1<<CAP1_INDEX);
SCT0->CAPCTRL[CAP0_INDEX] = (1<<EVT0_INDEX);
SCT0->CAPCTRL[CAP1_INDEX] = (1<<EVT1_INDEX);
/* Reset Counter L when Counter L event occurs */SCT0->LIMIT |=
SCT_LIMIT_LIMMSK_L((1<<EVT0_INDEX) | (1<<EVT1_INDEX));
/* Start the L counter */
SCT0->CTRL &= ~SCT_CTRL_HALT_L_MASK;
```

- d. 创建一个 wait_lv 函数，用于返回两个边更改之间的时间：

```
static uint32_t wait_lv(uint8_t lv)
{
    int cnt = 0;
    cnt = sct_wait_lv(lv) / 100;
    return cnt;
}
```

该 CEC 位实现 API 如下：

```
static uint8_t get_bit(uint8_t *val)
{
    uint32_t dt0, dt1;
    uint32_t ret = 0;
```

```

dt0 = wait_lv(0);
dt1 = wait_lv(1);
if(abs(dt0 - CEC_TIMING_BIT0_LOW) < CEC_TIMING_TOR && abs(dt1 - CEC_TIMING_BIT0_HIGH) <
CEC_TIMING_TOR)
{
    *val = 0;
    ret = 0;
}
else if(abs(dt0 - CEC_TIMING_BIT1_LOW) < CEC_TIMING_TOR && abs(dt1 - CEC_TIMING_BIT1_HIGH)
< CEC_TIMING_TOR)
{
    *val = 1;
    ret = 0;
}
else
{
    ret = 1;
}
return ret;
}
static void
    set_bit(uint8_t val)
{
if(val)
{
    pin_write(0);
    DelayUs(CEC_TIMING_BIT1_LOW*100);
    pin_write(1);
    DelayUs(CEC_TIMING_BIT1_HIGH*100);
}
else
{
    pin_write(0);
    DelayUs(CEC_TIMING_BIT0_LOW*100);
    pin_write(1);
    DelayUs(CEC_TIMING_BIT0_HIGH*100);
}
}
static uint32_t get_start_bit(void)
{
    uint32_t dt0, dt1;

    dt0 = wait_lv(0);
    dt0 = wait_lv(0);
    dt1 = wait_lv(1);
    if(abs(dt0 - CEC_TIMING_START_LOW) < CEC_TIMING_TOR && abs(dt1 - CEC_TIMING_START_HIGH)
< CEC_TIMING_TOR)
    {
        return 0;
    }
    return 1;
}

```

4 示例和测试

在这个演示中，当总线上没有其他 CEC 设备时，GoogleChromecast 充当发起者。Chromecast 通过发送头块多次轮询总线，如图 11 所示。

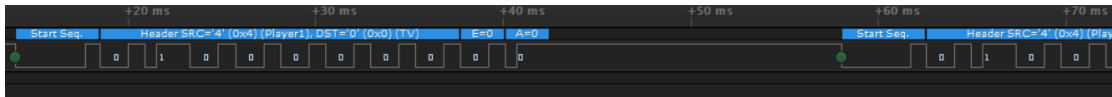


图 11. HDMI 系统中的物理地址

单片机的默认 CEC 地址是 0x00。当 MCU 连接到 CEC 总线时，响应目标地址 0x00。然后，Chromecast 检测到 ACK 位被拉下，并发送以下数据块，如 图 12 所示。



图 12. 发送：给设备电源状态(0x8F)帧

然后，单片机接收操作码，给出设备电源状态，并发送确认帧。这完成了一个完整的 CEC 总线转换，如 图 13 所示。单片机监控 CEC 总线活动，并在 UART 上显示日志。



图 13. 单片机响应：报告电源状态(0x90)帧

```
hdm cec test
CoreClock:150000000Hz
receive: 44
receive: 44
receive: 40
receive: 40 8F ← Chromecast send frame: GIVE_DEVICE_POWER_STATUS
GIVE_DEVICE_POWER_STATUS
transmit: 04 90 00 ← MCU responds frame: REPORT_POWER_STATUS
receive: 4F
receive: 40
receive: 40 8F
GIVE_DEVICE_POWER_STATUS
transmit: 04 90 00
receive: 4F
receive: 40
receive: 40 8F
GIVE_DEVICE_POWER_STATUS
transmit: 04 90 00
receive: 40 8F
GIVE_DEVICE_POWER_STATUS
transmit: 04 90 00
receive: 40 8F
GIVE_DEVICE_POWER_STATUS
```

图 14. 单片机 UART 日志

5 结论

本应用笔记描述了如何在 LC5500 系列上实现 HDMI-CEC 底层协议。它还提供了硬件和软件设置的示例代码和说明。介绍了 SCT 的基本用法，SCT 是 NXP 半导体独有的一种通用状态时序器。

6 修订记录

版本号	日期	说明
0	2020 年 2 月	初始版本
1	2020 年 10 月	LPC55S6x/LPC55S2x/LPC552x 替换成 LPC5500

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2020-2021.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2020 年 10 月

Document identifier: AN12732

arm