

AN13755

简便易行地将RT600/LPC55S30用作I3C主设备

第0版 — 2023年1月17日

应用笔记

文档信息

信息	内容
关键词	I3C、RT600、RT685、LPC553x、LPC55S3x
摘要	本应用笔记介绍了如何使用恩智浦RT600/LPC55S30作为主设备来搭建一个简单的I3C网络。



1 介绍

本应用笔记介绍了如何使用i.MX RT685/LPC55S30微控制器板、配合不同供应商的I3C从设备以及MCUXpresso软件开发套件（SDK）搭建一个单主设备的I3C小型网络。

i.MX RT600是一个面向嵌入式应用的双核微控制器系列，其特色是Arm® Cortex®-M33 CPU与Cadence Xtensa HiFi4高级音频数字信号处理器CPU的组合。此Cortex-M33包含两个硬件协处理器，可以为一系列的复杂算法提供增强的性能。该系列提供了一套丰富的外设和极低的功耗。LPC55S3x/LPC553x是一款面向嵌入式应用的基于Arm Cortex-M33的微控制器。

I3C是一种串行通信接口，既具有I²C的简单、少引脚、易于设计和多点（相对于点对点）设计等优势，同时还引入了SPI的更高的数据速率、简化的管脚和更低的功耗。此外，I3C能够在给定频率下实现更高的吞吐量，提供带内中断（从从设备到主设备）、动态寻址、高级电源管理和热插拔设计等。

本应用笔记将完整地施行在一个带有传统I²C设备的网络中为I3C从设备分配动态地址的过程，其间会为I3C数据传输设置一个高的速率。

2 推荐读物

此演示软件提供了搭建一个简单的I3C网络所需的构建模块。为了让演示在不同的从设备参数（例如从设备的默认I2C地址）的组合下运行，需要更新演示代码。除了获取芯片数据手册外，用户还应熟悉以下文档，它们可从MIPI联盟网站（<http://www.mipi.org/>）下载：

- MIPI I3C® v1.0和I3C BasicSM v1.0的常见问题解答（FAQ）
- I3C BasicSM改进型内部集成电路规范1.0版（2018年7月19日）

3 i.MX RT685/LPC55S36评估板的I3C总线

RT685/LPC55S36微控制器自带一个I3C外设的实体（I3C0）。其3个引脚I3C0_SCL、I3C0_SDA和I3C0_PUR可分别在PIO2_29/30/31（MIMXRT685（音频）EVK板的J18接头）和P0_9/24/28（LPC55S36 EVK板J9接头的脚20、18和16）处进行选择。

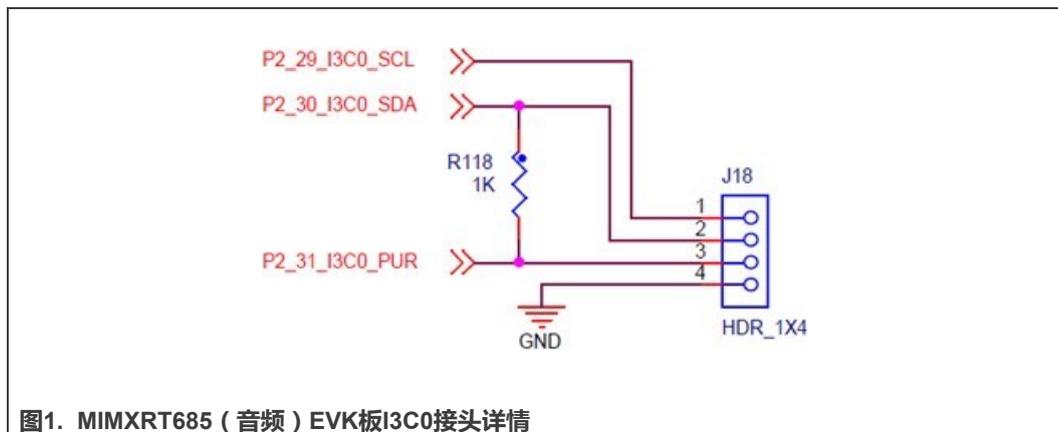


图1. MIMXRT685 (音频) EVK板I3C0接头详情

对于MIMXRT685 EVK板，可通过这些I3C引脚访问板上的传统I2C设备所使用的总线：

- WM8904 (U8, 面向便携式音频应用的超低功耗编解码器)，I2C地址0x1A
- TFA9894 (U12/17, 高效D类音频放大器)，I2C地址0x0E、0x34、0x35

I3C0_PURI3C主设备的上拉电阻控制脚，由RT600的I3C模块自动驱动。I3C规范提供了有关PUR功能及其作用的更多详细信息。

RT685音频EVK板和LPC55S36 EVK板的微控制器直连到板上的I3C从设备，没有任何传统I2C设备。

4 演示的软硬件要求

该演示使用RT685 EVK板、RT685 音频 EVK板和LPC55S36 EVK板进行开发和测试，为每个评估板都提供了一个专用的工程。虽然此代码的主要目标是展示如何使用上述评估板中的某一个搭建和运行一个简单的单主设备I3C网络，但它也可以用于其他任意RT600和LPC55S36的板子。

4.1 演示的软件要求

SDK的I3C驱动程序2.5.2版是此演示代码的基础（如需了解有关驱动程序版本的详细信息，请参见fsl_i3c.h）。该应用笔记的MCUX工程包含此驱动程序，只需重新编译即可。如果在工程中发现较早的I3C驱动程序版本，编译器会报错。

建立对I3C的支持所需的SDK文件：

工程的“drivers”文件夹中的文件：

- fsl_i3c.h, fsl_i3c.c

工程的“\component\i3c_bus”文件夹中的文件：

- fsl_component_i3c.h, fsl_component_i3c.c
- fsl_component_i3c_adapter.h, fsl_component_i3c_adapter.c

由于本应用笔记的功能性要求，需要对原始的SDK I3C驱动程序2.5.2版进行少量修改。

fsl_component_i3c.c中的例程如下所示：

- static uint8_t I3C_BusGetAddrSlotStatus(i3c_bus_t *bus, uint8_t checkAddr)
- static status_t I3C_CheckBusMasterOps(i3c_device_hw_ops_t *ops)
- static status_t I3C_BusMasterAssignDevDynamicAddr(i3c_device_t *masterDev)

它们不再是“静态”的，从而可以从代码中的任意位置访问。

例程如下：

- status_t I3C_BusMasterGetDeviceInfo(i3c_device_t *masterDev, uint8_t slaveAddr, i3c_device_information_t *devInfo)

目前已注释掉了与CCC GETMRL/GETMWL/GETDR相关的代码。

在fsl_component_i3c_adapter.c中：

- 静态例程为status_t I3C_MasterAdapterDoI2CTransfer(i2c_device_t *device, i3c_bus_transfer_t *xfer)

更新了传输描述符，以处理重复启动的情况。

demo_setup.h、hw_selection.c和main.c源文件既可用于RT600，也可用于LPC55S36 MCU。用户必须在demo_setup.h中选择正确的芯片才能编译和运行此演示：

```
#define DEMO_DEVICE DEMO_DEVICE_RT600
```

或

```
#define DEMO_DEVICE DEMO_DEVICE_LPC55S3X
```

根据所选的芯片，内嵌的 #if/#elif/#endif 机制会调整工程其余部分的代码。

如需详细了解如何在一个类似于“hello world”的简单工程中添加必要的I3C相关的文件，请参见[第18章](#)。

4.2 演示的硬件要求

除了[图2](#)所示的电路板之外，还使用了Saleae逻辑分析仪来帮助监测和分析I3C总线的活动。

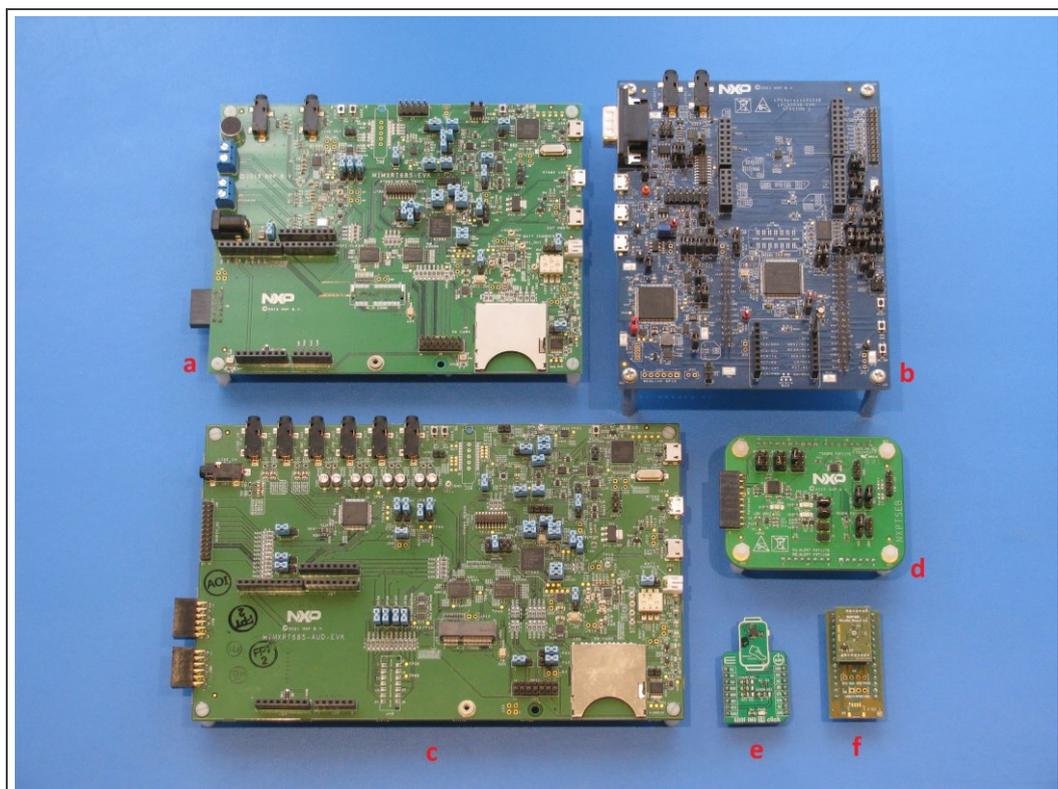


图2. 此应用笔记中使用的电路板

注意：图中恩智浦电路板A至D分别是：RT685 EVK板、LPC55S36 EVK板、RT685音频EVK板、配备P3T1085/1755温度传感器的NXPTSEB板；E是MikroElektronika的TDK/InvenSense ICM42688，F是Bosch Sensortec的BMP580气压传感器转接板（shuttle board）

注意：P3T1085和P3T1755器件最初推向市场时标称为P3T1108和P3T1175，但它们的名称后来被更改。

4.2.1 EVK的板载ICM-42688

RT685音频EVK板和LPC55S36 EVK板都自带了一个TDK/InvenSense的板载ICM-42688 MEMS器件（3轴陀螺仪和3轴加速度传感器）。这个有I3C功能的器件已连接到恩智浦MCU，有I2C地址为0x69的硬件连接，可以直接使用。

一些应用笔记的演示通常将板载ICM-42688配置为一个“轻敲”检测器（“tap”detector）。如需了解详细信息，请参见InvenSense AN-000173《ICM-426xx APEX运动功能：说明和用法》文档。

要触发这个轻敲检测器，请轻轻敲击EVK板的某个空白区域。

4.2.2 带P3T1085和P3T1755温度传感器的恩智浦NXPTSEB板

当连接配备了具有I3C功能的P3T1085和P3T1755温度传感器的外部NXPTSEB板时，请确保插入了以下跳线帽：

- 跳线帽J10[3:5]——将地址线A0设置为0
- 跳线帽J11[3:5]——将地址线A1设置为0
- 跳线帽J12[1:3]——将地址线A2设置为1

此操作将P3T1085和P3T1755的I2C地址分别配置为0x48和0x4C。

插入跳线帽JP2[1:2]和JP3[1:2]，将I3C的SDA/SCL信号从J13接头连接到温度传感器。将NXPTSEB板的J13连接器的引脚1和引脚2（I3C的SCL和SDA信号线）连接到RT600（音频）EVK板的J18连接器的引脚1和引脚2，或LPC55S36 EVK板的J9连接器的引脚20和引脚18。

要为NXPTSEB板上电，请将其J4.4和J4.7引脚连接到RT600（音频）EVK板的J29.4/7脚或LPC55S36 EVK板的J102.8/14脚（3.3V引脚和GND引脚）。使用NXPTSEB的跳线J1来选择P3T1085和P3T1755的VDD电源：插入跳线帽J1[1:2]选择1.8V，插入J1[2:3]选择3.3V。此“3.3V”是指外部电源电压，无论其电平是多少。

RT600（音频）EVK板的I3C总线仅在1.8V电压下工作。LPC55S36 EVK板的I3C总线电压由跳线JP1选择（1.9V或3.3V）。如果同时使用这两块电路板，则需要确保NXPTSEB的J1的选择与LPC55S36 EVK的JP1的选择相匹配。

当NXPTSEB是装置的一部分时，P3T1085/P3T1755被分别配置为每4秒/220毫秒读取一次温度。仅当来自同一温度传感器的读数与前一读数不同时，才会将其发送到控制台。这样，发送到控制台的数据量会显著减少，只向用户提供相关的信息。

注意：*P3T1085和P3T1755最初推向市场时的名称为P3T1108和P3T1175，但它们的名称后来被更改了。*

4.2.3 MIKROE-4237—外部ICM-42688

MikroElektronika的6DOF IMU板使用户能够在此装置中使用外部的ICM-42688。当与RT685音频EVK板或LPC55S36 EVK板一起使用时，该电路板的SPI_MISO必须连接到GND，以将ICM-42688的I2C地址设置为0x68。

将AP_CS连接到VDD，选择要使用的I3C接口；将电路板的SPI_SCK和MOSI脚连接到I3C的总线信号SCL/SDA。

如果使用RT685 EVK板（或其他未配备I2C地址为0x69的设备的电路板），MIKROE-4237可以使SPI_MISO=VDD（将其I2C地址设置为0x69），这样就可以连接到I3C总线了。

如果NXPTSEB和MIKROE-4237电路板是同一装置的一部分，则将MIKROE-4237的3.3 V脚和GND脚连接到NXPTSEB的J10.1和J15.1脚（VDD和GND脚）。这样，NXPTSEB的跳线J1选择的VDD电平（1.8 V/3.3 V）将自动施加到MIKROE-4237电路板上。

MIKROE-4237电路板可将其3.3 V和GND脚直接连接到RT600（音频）EVK板的J29.4/7脚或LPC55S36 EVK板的J7.7/8脚，以此来供电。

一些应用笔记的演示通常将外部ICM-42688配置为一个“倾斜”检测器。如需了解详细信息，请参见InvenSense AN-000173《ICM-426xx APEX运动功能：说明和用法》。

要触发此倾斜检测器，请轻轻将传感器电路板沿任一水平轴倾斜至少35度。请注意，当将电路板放回到其初始位置时，传感器会检测到又一个倾斜事件。有时，即使传感器没有倾斜，也可能在初始化后报告一个虚假倾斜事件。

4.2.4 带适配器的BMP580转接板3.0

这是Bosch Sensortec的一个用作气压传感器的评估板。在本应用笔记中，介绍了如何使用演示选项“总线速度”和“从设备轮询”从添加到网络的I3C设备收集基本参数（PID、BCR和DCR）。

5 构建演示

在构建这些应用笔记的工程时，必须使用MCUXpresso IDE v11.5.0 [Build 7332]或更高的版本。

本应用笔记准备了3个演示工程，一个用于RT685 EVK板，一个用于RT685音频EVK板，还有一个用于LPC55S36 EVK板。每个工程都是从SDK的“hello world”工程开始开发的，这些工程已随开发板的SDK一起提供。一旦编译完成后，代码就会加载到闪存中运行。

所有这3个演示都使用[第4.1节](#)中介绍的同一组文件。用户在demo_setup.h中设置作为I3C主设备的微控制器，其余的代码就会相应地被包含/排除。

在编译时，要验证SDK的I3C驱动程序的版本为2.5.2或更新的版本。如前所述，需要对原始的SDK的I3C驱动程序源代码进行少量修改，才能提供本演示所需的全部功能。如果终端用户尝试用一组更新的文件替换原始工程的驱动程序和/或组件文件，则应由用户来保证所需功能的存在。

这些RT600和LPC55S36的演示的主时钟分别设置为500 MHz和150 MHz。主时钟可通过MCUXpresso时钟工具轻松控制。所有3个演示的I3C的功能时钟都设置为25 MHz（对于LPC55S36是必须的，而RT600可以高达100 MHz）。由于I3C的功能时钟来自主时钟，因此主时钟速率必须是所选25 MHz的整数倍。如果选择了一个不支持的主时钟频率，演示会报错并停止。

由于I3C0_SCL和SW_DIO功能共用同一个LPC55S36的引脚，因此不能从调试器运行LPC55S36的I3C演示。

要运行LPC55S36 EVK板的I3C演示，用户必须首先在断开跳线JP27的情况下为LPC55S36 EVK板上电，然后将代码加载到闪存中，再给评估板断电。在JP27插入跳线帽的情况下给评估板上电，闪存中存储的代码即开始运行。

断开的JP27（默认设置）使MCU-Link能够驱动MCU/外部目标MCU的SW接口。JP27插入跳线帽后，MCU-Link不会驱动MCU的SW，而I3C的SCL功能就可以使用了。JP27位于评估板的左下方，靠近调试芯片（U16）。

一旦代码加载到闪存中，这些I3C的RT600演示就可以从调试器运行了。

所有演示都部署了一个UART控制台（115200，8N1，每行80个字符），用来与用户进行通讯。

6 运行演示

如果在启动演示时终端上没有输出，请按几次“ESC”键，让演示刷新输出：



图3. RT600/LPC55S36板的欢迎消息

此处必须选择运行演示的电路板。基于demo_setup.h中指定的微控制器，提供了相应的EVK板（RT685音频EVK板或LPC55S36 EVK板）以及“其他RT600/LPC55S3X”板（第二个选项可用于任意的RT600/LC55S3X电路板）。

如果选择了RT685音频EVK板或LPC55S36 EVK板，该演示需要一个I2C地址为0x69的板载I3C从设备（即InvenSense的ICM-42688）。

如果选择了LPC55S36 EVK板，则会弹出一条警告消息，提醒用户插入跳线帽JP27：

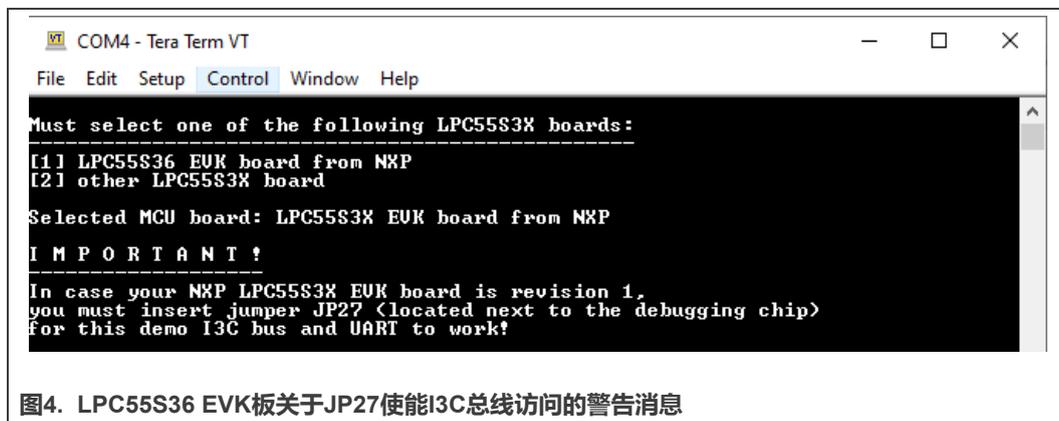


图4. LPC55S36 EVK板关于JP27使能I3C总线访问的警告消息

该演示需要获取NXPTSEB板（带有P3T1085和P3T1755温度传感器）和/或外部ICM-42688板是否存在的信息。

如果使用一个带有板载ICM-42688的EVK板，并且该装置中还包含一个外部的ICM-42688，则用户会收到一条警告消息，要求将外部ICM-42688的I2C地址更改为0x68：



图5. 外部ICM-42688的I2C地址警告消息

接下来，用户设置I3C的MCONFIG寄存器的偏移参数（如需了解详细信息，请参见微控制器手册）。默认值0适用于大多数硬件设置。

最后，用户从4个场景选择一个来运行演示：

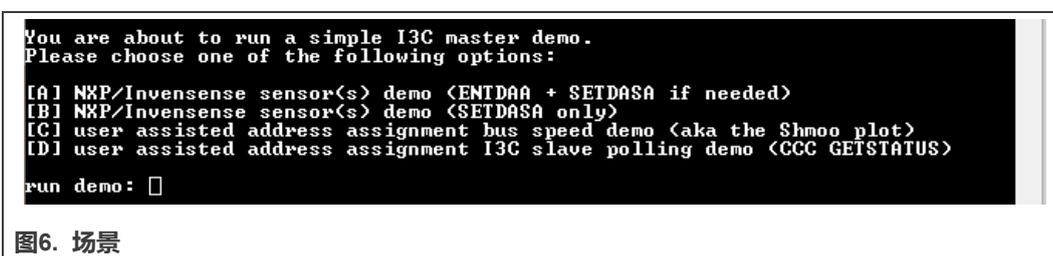


图6. 场景

7 动态地址分配（DAA）——此演示如何使用ENTDAA和SETDASA命令

I3C主设备使用其从设备的动态地址与从设备通信。上电后，有I3C功能的从设备通常作为I2C设备运行，对地址进行响应，而这些地址可以通过将一个封装引脚拉高/拉低来进行部分设置（在封装上，I2C设备通常有一个引脚用于控制I2C地址的最低有效位）。I3C规范在介绍I3C主设备用于访问总线上的传统I2C设备的地址时，使用了“静态地址”这一术语。

只有在I3C主设备为从设备分配了一个动态地址后，这个具有I3C功能的从设备才能得到I3C接口的改进所带来的好处，例如提高数据速率、降低功耗和带内中断等等。

I3C主设备在初始化完成后，必须先给具有I3C功能的设备分配动态地址，它们才能使用。这个过程称为动态地址分配（DAA），可以通过多种不同的方式执行。I3C规范详细介绍了与DAA相关的命令（如RSTDAA、ENTDAA、SETDASA、SETAASA和SETNEWDA）。在本应用笔记中，只介绍了ENTDAA和SETDASA。

恩智浦/Invensense传感器演示（主菜单的选项A和B）背后的目的是展示如何实现一个典型的I3C用例，即一个I3C主设备分配动态地址，初始化从设备并无限期地轮询它们。如需了解有关I3C的IBI驱动的应用，请参见《使用LPC553x/LPC55S3x构建I3C传感器网络》（文档[AN13577](#)）。

主菜单选项A和B之间的区别在于I3C的CCC命令，此命令用于I3C总线主设备为从设备分配动态地址。本演示中选择了一种从设备的组合，目的是展示为从设备分配动态地址相关的典型真实场景。

I3C主设备在建立网络时必须完成的第一项任务是为网络中的从设备分配动态地址。为此，主设备必须能够唯一地识别当前的每一个从设备。

I3C主设备在分配动态地址时使用的“go-to”命令是I3C CCC ENTDA A（输入动态地址分配）命令。I3C主设备使用此命令进入动态地址分配（DAA）过程，此时从设备用其8字节的“指纹”（由一个48位的唯一ID（临时ID，PID）、一个BCR（总线特征寄存器）和一个DCR（设备特征寄存器）字节组成）响应。在DAA过程中，从设备的SDA脚处于开漏模式，使得先输出0的从设备优先于同一时刻输出1的从设备。当主设备接收到来自某个从设备的唯一的64位序列时，主设备会为它分配一个动态地址。被分配了动态地址的从设备会忽略DAA过程的其余部分，从而允许主设备识别网络中其余的从设备。

I3C从设备的临时ID、BCR和DCR的值均可在该设备的数据手册中找到。

下面列出了RT600/LPC55S36 EVK的I3C板载ICM-42688、外部ICM-42688（由MikroElektronika模块提供）和恩智浦NXPTSEB板的I3C温度传感器P3T1085/1755的临时ID、BCR和DCR。各个从设备的I2C地址的设置如[第4.2节](#)所述。

表1. 本应用笔记中使用的I3C从设备

#	设备	硬件设置 静态/I2C 地址	48位临时ID+8 位BCR+8位 DCR读数	MIPI制造商 ID [47:33]	位 [32]	供应商零部件 ID [31:16]	实际 ID [15:12]	版本[11:0]	BCR [7:0]	DCR [7:0]
1	NXPTSEB P3T1085	0x48	0236_1529 0090_0363	000_0001 0001_1011 0x11B	0	0001_0101 0010_1001 0x1529	0000 0x0	0000_1001_0000 0x090	0x03	0x63
2	NXPTSEB P3T1755	0x4C	0236_152A 0098_0363	000_0001 0001_1011 0x11B	0	0001_0101 0010_1010 0x152A	0000 0x0	0000_1 001 _1000 0x098	0x03	0x63
3	外部 ICM-42688	0x68	046A_0000 0000_27A0	000_0010 0011_0101 0x235	0	0000_0000 0000_0000 0x0000	0000 0x0	0000_0000_0000 0x0000	0x27	0xA0
4	EVK板载 ICM-42688	0x69	046A_0000 0000_27A0	000_0010 0011_0101 0x235	0	0000_0000 0000_0000 0x0000	0000 0x0	0000_0000_0000 0x0000	0x27	0xA0

1/2号设备的“版本”列条目中有几个位是以粗体显示的。这些位是由设备的A2/I/O物理引脚控制的。1/2号设备将其以硬件选择的I2C地址传递到I3C的48位临时ID。

3/4号设备即两个ICM-42688单元，报告了相同的临时ID/BCR/DCR组合，与硬件选择的I2C地址无关。

CCC ENTDA A命令会触发DAA过程的运行。该过程取决于从设备的唯一临时ID/BCR/DMA的内容，以便主设备可以识别网络中的每一个I3C从设备，并为它们分配I3C动态地址。

为了使DAA过程成功完成处理[表1](#)中的4个I3C从设备的任务，应该有4个不同的临时ID/BCR/DCR的数据块。1/2号设备本身具有两个不同的64位数据块，但3号和4号设备会以相同的数据块来响应，即使它们具有不同的I2C地址（并且能够在同一I2C环境中正常运行）。

这个问题可以使用CCC SETDASA（从静态地址设置动态地址）命令解决。I3C主设备可以使用该命令通过从设备的I2C地址直接与相关的从设备进行通信，并为其提供一个I3C动态地址。通过SETDASA命令接收动态地址的从设备则会忽略之后可能出现的ENTDAA命令。

对于上述的小型四从设备网络，I3C主设备将在运行ENTDAA命令之前，使用SETDASA命令与至少一个呈现的ICM-42688（I2C地址为0x68和0x69的从设备）进行通信。在运行ENTDAA命令之前，通过将I3C动态地址分配给其中一个ICM-42688设备，其余的从设备（即将获得分配到的动态地址的设备）将组成一组具有独特指纹的设备——这样ENTDAA就可以处理了。

在这个AN演示中，ENTDAA分配的动态地址从0x09开始，而SETDASA分配的动态地址从0x40开始——用户可以很容易地在代码中更改这些内容（如有必要）。

以下的控制台截图来自以ENTDAA和SETDASA的命令组合，运行恩智浦/InvenSense传感器演示的RT685音频EVK板 + NXPTSEB板 + 外部ICM-42688板的装置：



该I3C网络概览的位置0始终显示I3C主设备的参数（动态地址0x08，静态/I2C地址0x1A，供应商ID 0x0345——为本演示随机选择，型号0x0，BCR/DCR为0x60/0x00）。I3C主设备参数总是由SDK的I3C驱动程序设置。对于列到位置1的从设备，其参数来自从设备本身，并通过I3C CCC的命令之一获取。静态地址00表示在DAA过程运行时，没有关于从设备I2C地址的信息。

图7所示的装置使用两个ICM-42688设备，其中一个设备必须通过SETDASA命令获得分配到的动态地址。特别是，RT685音频EVK的板载ICM-42688（I2C地址0x69）是使用SETDASA命令寻址的，并获得分配到的动态地址0x40。该ICM-42688设备的数据显示在位置1中。

位置2、3和4表示恩智浦温度传感器P3T1085和P3T1755以及那个外部的ICM-42688传感器。这些设备使用ENTDAA命令进行处理，并获取动态地址0x09、0x0A和0x0B。它们的供应商ID + 型号 + BCR + DCR的内容很好展示了在主设备启动的DAA过程中，所采用的以0在前的从设备获胜的优先级方案。

当SETDASA是唯一使用的命令时，通过运行主菜单的第二个选项，相同的I3C网络的概览将如下所示（为I3C主设备保留位置0）：

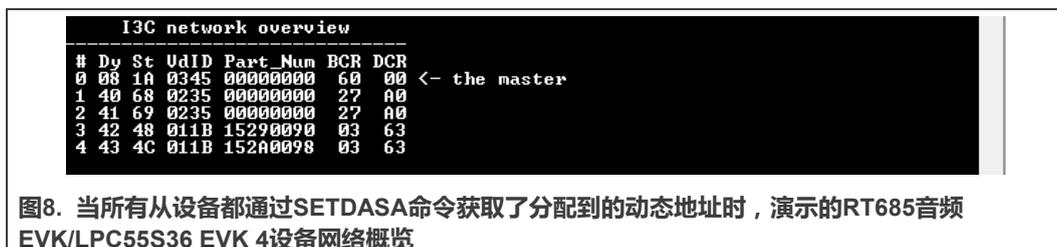


图8. 当所有从设备都通过SETDASA命令获取了分配到的动态地址时，演示的RT685音频 EVK/LPC55S36 EVK 4设备网络概览

主设备使用SETDASA命令给所有4个从设备分配动态地址。这要求每个从设备都有一个唯一的I2C地址，这样主设备就可以使用该地址与之通信。动态地址0x40到0x43分别分配给板外ICM-42688、板载ICM-42688、P3T1085和P3T1755。主设备通过发送专用的GETPID、GETBCR和GETDCR CCC命令来收集从设备的供应商ID、型号、BCR和DCR信息，随之发送从设备的动态地址。

8 演示详情

主菜单为用户提供了四种场景来选择，其中两个场景说明了为从设备分配动态地址所需的基本步骤，另外两个场景可以作为一组简单的工具来评估一个新的从设备是否适合当前的I3C装置。

两个全局变量控制演示的运行方式，分别是demo_hw_info和ch_demo_mode，均在main.c中定义。

有关运行演示的微控制器板（RT600_AUDIO_EVK_BOARD、RT600_GENERIC_BOARD、LPC55S3X_EVK_BOARD或LPC55S3X_GENERIC_BOARD）的信息和所用的传感器（NXPTSEB_BOARD、ONBOARD_ICM42688和EXTERNAL_ICM42688）的信息存储在demo_hw_info变量中。

用户对运行哪个场景的选择存储在ch_demo_mode中（共4个值，分别定义为ch_demo_main、ch_demo.main_setdasa_only、ch_demo_speed和ch_demo_poll_I3C_status）。

本应用笔记代码使用以下指针来简化对基于ICM-42688的敲击/倾斜检测器以及恩智浦温度传感器P3T1085和P3T1755的访问：node_icm426xx_tap_detector、node_icm426xx_tilt_detector、node_P3T1085和node_P3T1755。这些指针与i3c_device_t结构体有关。如需了解有关如何使用这些指针的详细信息，请参见main.c，如需了解该结构体的详细信息，请参见fsl_component_i3c.h。

主菜单场景使用这两个全局变量来选择要运行的代码部分，以便执行给定的任务。终端用户可以分析演示代码，并使用变量demo_hw_info和ch_demo_mode来识别该AN代码的各个部分，以便在客户定制的I3C应用中重复使用。

8.1 主菜单选项A和B：恩智浦Invensense传感器演示（ENTDAA + SETDASA，如需要）和（仅SETDASA）——实施细节

主菜单选项A和B要求连接的I3C从设备的数量和种类不要超出表1列出的范围。如果主设备没有检测到I3C从设备，演示则会停止。

[表1](#)中的设备可以少至只有一个，也可以所有四个设备全部连接。

虽然菜单选项A和B都依赖于SDK的I3C驱动程序，但它们适用于两个略有不同的I3C应用领域。这两个选项都是从创建一条I3C总线开始的，总线上有一个主设备——RT600或LPC55S36。这个主设备将成为SDK的I3C驱动程序要处理的I3C设备列表中的第一个条目。

选项A使用SETDASA（如需要）和必需的ENTDAA命令收集的设备参数扩展此列表——这些参数包括临时ID、BCR和DCR。在创建设备列表后，准备一组专用指针用于访问倾斜和敲击检测器（ICM-42688设备）以及恩智浦温度传感器。相较于一直在I3C总线实体的列表中上下跑动，这些指针可以使得对相关的设备的访问更容易。

运行菜单选项A时创建的设备列表有一个独特之处，那就是与ICM-42688设备匹配的第一个条目（通过读取候选设备的“WHO AM I”寄存器进行验证）会被标记为敲击检测器。如果发现了另一个ICM-42688，它将作为倾斜检测器。

恩智浦温度传感器P3T1085和P3T1755是作为单独的实体出现的，因此不会有操作方面的歧义。

如果运行ENTDAA命令时检测到了在[表1](#)中没有列出的设备，将在网络概览（本演示运行的一部分）中列出，但之后将不会以任何方式访问它们。

主菜单选项A的流程可以被视为一个关于如何处理一个潜在棘手情况的教科书示例（当有两个设备具有相同的临时ID/BCR/DCR组合并需要同时使用SETDASA和ENTDAA命令时）。一旦创建了总线上设备的列表，该流程将展示如何识别相关的设备（应用程序可以初始化的设备，即敲击/倾斜检测器，或设置温度传感器的采样率），以及如何通过使用专用的指针来简化后续对这些设备的访问。

菜单选项B更专注于一个预定义的传感器组，根本不关心是否存在任何其他设备。选项B流程不会扩展现有的I3C设备列表——其中只有一个条目用于描述主节点。该流程根据SETDASA命令的结果生成指向敲击/倾斜检测器和温度传感器的专用指针。这些SETDASA命令由用户提供的设置说明来驱动。

示例：如果使用RT685音频EVK板，并且该评估板连接了NXPTSEB电路板，但用户对菜单中是否存在NXPTSEB的问题回答“否”，则主菜单选项B（仅使用SETDASA命令的选项）将只在I2C地址0x69处查找板载的ICM-42688，而且会找到。不会尝试与任何其他从设备进行通信，因为没有（来自用户的）它们存在的指示。SETDASA命令的操作基于存在一个具有特定I2C地址的设备的假设。它并不是为了探索主设备周围的所有I2C地址而设计的（如同使用ENTDAA命令的情况）。

这就是为什么对于从定义一个网络到更倾向于主设备与从设备通讯的小转变，流程B更适用于数据流型的I3C应用。检测到的设备被映射到专用指针上，以便主设备可以立即访问它们。在大多数情况下，终端用户知道装置中存在哪些I3C从设备，能够完全控制它们的I2C地址，并且用户可以很容易准备主设备发送的SETDASA命令来构建网络。

两个菜单选项（A和B）都进入一个while(1)无限循环。在此循环中，I3C主设备使用一个从设备专用的指针对其进行轮询。对于敲击/倾斜检测器，读取状态寄存器。对于温度传感器，读取温度寄存器。如果来自同一传感器的两个连续读数有变化，则向控制台生成一份报告。

8.2 主菜单选项C和D：用户辅助地址分配总线速度演示（又名Shmoo图）和I3C从设备轮询演示（CCC GETSTATUS）——实施细节

这两个选项用作小工具，对任意I3C总线进行基本特征分析，包括可以实现的速度和在总线上检测到的设备。

这些选项探测I3C总线，寻找具有I3C功能的从设备。在运行一个基于DAA的通过程序之前，用户可以通过SETDASA命令指定要处理的I2C地址（总是发出CCC ENTDA命令）。在之前的演示中所做的电路板和外部模块的选择不会影响菜单选项C和D。如果DAA过程有遇到问题的可能（两个具有相同PID的从设备在线可能导致ENTDA失败），则用户必须使用其I2C地址和SETDASA机制对其中的一个从设备进行寻址。

总线速度演示为终端用户提供了一个工具，可以采用该演示支持的两种I3C时钟模式系列（对称和非对称SCL时钟）来获取该I3C装置的信息。有关I3C的SCL时钟的更多详细信息，请参见I3C规范（“总线配置”、“总线通信”和“电气规范”部分）。

对称SCL时钟可用于一个仅存在I3C从设备的I3C总线（参见I3C规范中的“纯总线”）。具有40 ns高电平状态的非对称SCL时钟可用于任意I3C总线，包括I2C和I3C从设备混合在一起的总线（参见I3C规范中的“混合快速总线”和“混合慢速/有限总线”）。I3C SCL的40 ns高电平状态会被I2C设备的50 ns干扰滤波器（glitch filter）滤掉，所以传统I2C设备可以忽略总线上的所有I3C数据流。

总线速度演示会改变对称SCL时钟模式的低电平和高电平状态，都从640 ns降低到40 ns（相当于将I3C的SCL速率从781 kHz提升到12.5 MHz）。运用非对称的I3C SCL时钟模式会将时钟低电平状态从640 ns变为40 ns，同时始终保持40 ns的高电平状态（有效运行频率从1.47 MHz至12.5 MHz）。

对称和非对称测试模式的SCL时钟低/高电平状态迭代步长均为40 ns。

对于给定的SCL参数集，总线速度演示会输出时钟配置的详细信息，I3C主设备会向每一个在DAA过程中识别的I3C从设备发送CCC GETPID命令。如果所有返回的PID都是正确的，则更新SCL参数，并进行下一次运行，直至所有的16个可用的配置测试完成。

仅使用板载的ICM-42688在RT685音频EVK板或LPC55S36 EVK板上运行总线速度演示几乎总是能成功达到12.5 MHz的SCL速率。在一个使用外部电路板（NXPNTSB传感器电路板和外部ICM-42688单元均存在）的装置中，I3C总线至少延长了30 cm至50 cm（从而增加了不想要的电容），则最大SCL速率下降。

如果速度演示没有报告成功运行（在测试完所有16个SCL速率配置后打印“test PASSED”（“测试通过”）），则发送到控制台的最后一组SCL参数为失败的参数。一个经过测试的装置的I3C应用可以成功地使用所报告的任意SCL配置，但不能使用最后一个。随着SCL速率的升高，此测试可能会以多种不同的方式失败。有时，驱动程序会检测到某个问题，随后出现一个“test FAILED!”（“测试失败！”）消息，但这也有可能是因为I3C主设备与从设备通讯时出现问题而导致测试“冻结”。

在主菜单选项D（I3C轮询演示）中，在成功完成DAA过程后，将部署用户选择的SCL配置，并开始对已识别的I3C从设备进行轮询。进入一个无限循环，CCC GETSTATUS命令被一次又一次地发送到每个从设备。GETSTATUS命令让主设备能够获得来自于从设备的少量I3C相关的信息，而无需实际配置从设备来执行应用中的任何功能（惯性模块、温度传感器等）。每完成100个读取周期后，就会显示一个总循环计数。

此I3C轮询演示是一个适用于任意装置的小工具，可以验证与其连接的I3C从设备是否能够被检测到，以及它们的动态地址是否被分配。它可以帮助验证I3C总线在SCL时钟配置了一个典型非对称时钟的情况下是否能够正常运行，此时使用CCC GETSTATUS命令运行无限循环来从设备轮询，根本无需配置从设备。

9 如何配置RT600/LPC55S3x I3C管脚

配置RT600/LPC55S3x的I3C管脚时有一些规则，以便基于这些设备的I3C主设备能够充分利用I3C总线。

不要使用内置上拉电阻。这只会给引脚增加负担，而且一致性很差。

驱动强度和斜率控制（I/O引脚配置参数FULLDRIVE和SLEWRATE）的使用应考虑到走线长度。在一个装置中如果电路板的电容较小，应该能够使用较低的驱动强度（FULLDRIVE = Normal）和无斜率（SLEWRATE = Standard mode）。如果走线或板外引线较长且电容较大（例如本演示的情况），请尝试使用一个更高的驱动（FULLDRIVE = Full），但要使用斜率控制以避免振铃（SLEWRATE = Slow mode）。这是根据硬件环境来选择的。

弱无源器件（如100 kΩ至200 kΩ电阻）可用于SDA和SCL的走线。对于初学者，可以参考RT685音频板和LPC55S36 EVK板的I3C总线实现细节。

PUR脚之所以存在，是因为RT600/LPC55S3x的管脚不能支持既有非常弱的驱动，又有高强度（1 kΩ）的上拉，这些管脚不应该上拉（I3C0_PUR脚P2_31的IOCON参数PUPDNA = 0，对应于上拉/下拉被禁用），也可以没有斜率控制。它在高阻和输出之间切换，将1-kΩ的负载施加到SDA上。

main.c中的configure_i3c_pins()例程配置了RT600/LPC55S3x的I3C引脚。

10 设置I3C波特率

I3C的SCL时钟来自I3C模块的功能时钟，本应用笔记将I3C的功能时钟设置为25 MHz。RT600的I3C功能时钟的最大速率为100 MHz，而LPC55S36的同一参数为25 MHz。为了方便起见，在本演示中使用25 MHz的I3C功能时钟，不管所选的I3C主设备是什么。

每次演示运行都会报告主时钟速率、CPU时钟速率和初始I3C时钟配置的详细信息。图9所示为无论演示如何运行，都应在控制台输出中看到的内容。由于I3C的时钟配置是在应用运行期间更新的，因此会报告最新的一组参数。

```

hello LPC55S36 world!
Main clock [Hz]: 150000000
CPU clock [Hz]: 150000000

Initial baudrate setting:
=====
I3C clk [Hz]: 25000000 <40.0 ns>
PP rate [Hz]: 833333 <H/L ratio: 15/15>
OD rate [Hz]: 416666 <H/L ratio: 15/45> ODHPP = 1
I2C OD [Hz]: 138888

hello RT600 world!
Main clock [Hz]: 500000000
CPU clock [Hz]: 250000000

Initial baudrate setting:
=====
I3C clk [Hz]: 25000000 <40.0 ns>
PP rate [Hz]: 833333 <H/L ratio: 15/15>
OD rate [Hz]: 416666 <H/L ratio: 15/45> ODHPP = 1
I2C OD [Hz]: 138888

```

图9. RT685音频EVK板和LPC55S36 EVK板控制台主时钟、CPU和初始I3C时钟的报告

RT685的主时钟和CPU时钟分别为500 MHz/250 MHz，LPC55S36的主时钟和CPU时钟分别为150 MHz/150 MHz。无论哪种情况，I3C的功能时钟均为25 MHz。无论哪个设备充当I3C主设备，都具有相同的功能时钟，这有助于在整个演示过程中保持I3C的时钟设置。原本，I2C开漏/SCL速率为138.8 kHz，I3C开漏SCL速率为416.6 kHz，并且I3C推挽SCL速率是833.3 kHz。选择了相对较低的初始SCL时钟速率，从而使得作为I2C器件上电的具有I3C功能的设备可以在快速模式（SCL < 400 kHz）的边界内与I3C主设备通信，正如某些具有I3C功能的设备所要求的那样。

SDK的I3C驱动程序例程I3C_MasterSetBaudRate()根据用户的要求配置I2C和I3C的波特率。默认情况下，SDK驱动程序将I3C的推挽SCL速率配置为占空比50%的信号。由于默认设置为MCONFIG.ODHPP = 1，I3C开漏SCL的波形将具有与I3C推挽SCL时钟模式相同的高电平状态。然而，SDK驱动程序2.5.2版中的这个SDK例程不能将I3C推挽/开漏的高电平状态设置为50 ns以下，除非指定一个12.5 MHz的时钟频率（以满足“混合快速总线”和“混合慢速/有限总线”规范的要求）。在配置实际设置时请记住这一点。

通常在上电后，具有I3C功能的从设备会作为I2C设备运行，且能够达到I2C快速模式+（SCL时钟频率高达1 MHz）的数据速率。这就是本演示的初始I3C时钟速率不超过1 MHz的原因。I3C推挽/开漏和I2C开漏的波特率要求分别为800/400/400 kbit/s。实际波特率取决于在I3C_MasterSetBaudRate()中实现的算法。

I3C波特率可根据用户的请求在演示过程中进行更新。I3C波特率更新的每个实例都提供了大量注释，以便在实际应用中重复使用此类代码和/或根据需要进行修改。

下面是将I3C推挽波特率设置为6.25 Mbit/s (主菜单选项I)的情况：

```
// I3C SCL = 6.250 Mbit/s (mixed bus spec)
// 25 MHz/ 40 ns driven timing
// OD: LOW = 200 ns, ODHPP=1 => HIGH = 40 ns => freq(OD) = 1/
(200 ns + 40 ns) = 4.166 MHz
// PP: HIGH = 40 ns, LOW = 120 ns <=> freq(PP) = 1/(40 ns + 120
ns) = 6.250 MHz
// I2C OD: freq = 1/(1400 ns + 1400 ns) = 357 kHz
pnt_i3c_aux = ((i3c_master_adapter_resource_t
*)demo_i3cBus.currentMaster->devControlInfo->resource)->base;
demo_I3C_clkdiv(demo_main_clk_to_25MHz_div);
pnt_i3c_aux->MCONFIG =
(pnt_i3c_aux->MCONFIG & ~(I3C_MCONFIG_ODHPP_MASK |
I3C_MCONFIG_PPBAUD_MASK | I3C_MCONFIG_PPLOW_MASK |
I3C_MCONFIG_ODBAUD_MASK | I3C_MCONFIG_I2CBAUD_MASK)) |
I3C_MCONFIG_ODHPP(1) | // Open-Drain High = 1 use PPBAUD for
open-drain count for I3C messages
I3C_MCONFIG_PPBAUD(0) | // THPP = TFCLK*(0+1) = 40 ns * (0+1) =
40 ns
I3C_MCONFIG_PPLOW(2) | // TLPP = THPP + TFCLK*2 = 40 ns *
(0+1+2) = 120 ns
I3C_MCONFIG_ODBAUD(4) | // ODBAUD = THPP*(4+1) = 40 ns * (4+1)
= 200 ns
I3C_MCONFIG_I2CBAUD(12); // I2C_HIGH/LOW = ODBAUD*((12>>1)+1) =
200 * 7 = 1400 ns
```

以下的I3C波特率推挽/开漏时序的计算基于用户手册I3C的MCONFIG寄存器的描述。

由于I3C的功能时钟为25 MHz，因此时钟分辨率/步长为40 ns。

MCONFIG.PPBAUD = 0，因此I3C的推挽高电平状态为 $(0+1)*40\text{ ns} = 40\text{ ns}$ 。

MCONFIG.PPLOW = 2，因此I3C的推挽低电平状态是通过将该参数加到推挽高电平状态上计算得到的： $((0+1)+2)*40\text{ ns} = 120\text{ ns}$ 。

MCONFIG.ODHPP = 1使得I3C的开漏高电平状态与I3C推挽高电平状态相同，即40 ns (参见PPBAUD)。

MCONFIG.ODBAUD = 4产生 $(4+1)*40\text{ ns} = 200\text{ ns}$ 的开漏速率。

MCONFIG.I2CBAUD = 12；得到 $((12>>1)+1)*\text{速率}(\text{ODBAUD}) = 7*200\text{ ns} = 1400\text{ ns}$ 的I2C高/低电平状态。

I3C的开漏波特率为：

$\text{I3C_od_low} + \text{I3C_od_HIGH} = 200\text{ ns} + 40\text{ ns} = 240\text{ ns} \Leftrightarrow 4.166\text{ MHz}$

I3C的推挽波特率为：

$\text{I3C_pp_low} + \text{I3C_pp_high} = 120\text{ ns} + 40\text{ ns} = 160\text{ ns} \Leftrightarrow 6.25\text{ MHz}$

I2C的波特率为：

$2*\text{I2C_od_low_high} = 2*1400\text{ ns} = 2800\text{ ns} \Leftrightarrow 357\text{ kHz}$

此波特率配置可以保证I3C的SCL高电平状态保持在50 ns以下，使其适用于混合有I3C和传统I2C设备的I3C总线。同时，I2C的波特率保持在400 kHz以下，以便I3C主设备也可以访问传统的I2C设备 (如需要)。

11 使用CCC GETPID命令进行压力测试

I3C接口支持高达12.5 Mbit/s的数据速率。对于一个给定的装置，可以实现的数据速率取决于I3C总线的物理特性，而主菜单选项“总线速度演示”可用于评估硬件可以达到的最高数据速率。

总线速度演示使用CCC GETPID命令作为工具，以验证一个给定的I3C波特率和时钟配置至少可以支持少量数据的传输。当一个I3C从设备接收到CCC GETPID命令时，它会向主设备返回6个字节的内容。这些字节以瞬时突发的方式背靠背发送。发送CCC GETPID命令是在通用I3C环境中通过使用I3C命令制造尽可能大的压力的方法，这是因为所有从设备都必须响应此命令，不管它们在应用中提供什么功能。惯性传感器可以轻易地将数百字节的数据背靠背地发送回给主设备，但必须先对这些传感器进行配置，并且每个供应商的配置都不同。

默认情况下，当运行DAA过程时，主设备会在演示的早期收集从设备的PID（PID封装在从设备对ENTDAA命令的响应中。如果使用SETDASA命令，则会部署专用的CCC GETPID、GETBCR和GETDCR命令来收集此信息）。随后在演示中，主设备将后续的PID读数与DAA过程中设置的黄金模式进行比较。如果不匹配，则会向控制台发送一条错误消息，并停止演示。

12 为什么需要I3C_BusMasterCreate_basic()例程？

SDK提供了I3C_BusMasterCreate()例程。该例程通过广播CCC RSTDAA命令来初始化I3C主设备并重置所有从设备的动态地址。如有必要，使用SETDASA命令分配动态地址，通过下发ENTDAA命令运行DAA过程，并使用一个广播的CCC ENEC命令启用热插入、主设备身份和从设备中断请求。I3C_BusMasterCreate()例程与SDK的I3C提供的演示工程一起使用。

本应用笔记演示代码有一组需要一个不同例程的需求。

I3C_BusMasterCreate_basic()例程仅初始化主设备并更新相应地I3C设备列表。这是因为已经确定了与实现DAA过程相关的两种截然不同的场景。I3C_BusMasterCreate_basic()例程到此停止。

在第一个场景（“教科书式的DAA过程”）中，SETDASA和ENTDAA命令组合在一起。主演示选项A，“恩智浦/Invensense传感器演示（ENTDAA + SETDASA，如需要）”，表现了这种情况。SETDASA命令用于处理那些提供了与网络中其他设备相同的临时ID的从设备，但可以使用一个专用的静态I2C地址对它们进行寻址。发送SETDASA命令“清洁了空气”，让ENTDAA可以处理总线上其余的I3C从设备。此场景完全依赖于所提供的SDK I3C驱动程序和其中使用的软件技术（广泛使用列表和内存分配-malloc()）。

在主演示选项B所展示的第二个场景，“恩智浦/Invensense传感器演示（仅SETDASA）”中，只使用了SETDASA命令。当I3C主设备提前知道所有要连接的从设备时，这更像是一种“动手操作”的方法。I3C主设备根据从设备的静态I2C地址直接（逐个）向其发送SETDASA命令，并发布专用的动态地址。

这种方法使主设备更容易从完成DAA过程转到与从设备进行通讯。

这就是I3C_BusMasterCreate_basic()例程对I3C主设备进行预置的原因，以便它可以继续执行第一个场景（SETDASA + ENTDA A）或第二个场景（仅SETDASA）。当该例程完成时，通过广播CCC DISEC命令来禁用热插入、主设备身份和从设备中断请求。随后广播CCC RSTDA A命令，确保之前分配给从设备的任何动态地址都无效。随后是第一个或第二个DAA场景。

13 I2C和I3C传输是否可以在同一条总线上共存？

有两种场景，可以在预计用作I3C的总线上观察到I2C传输。

场景1：配置一个具有I3C功能的设备使其从I2C切换到I3C模式

I3C设备在上电之后作为I2C设备。通过将从设备封装上的专用引脚拉到预定义的电平，使其能够使用I3C接口。对于ICM-42688，通过将AP_CS引脚拉到高电平，该设备被配置为可使用I2C/I3C接口。另一方面，一些具有I3C功能的设备需要使用I2C写命令来写入某个特定的寄存器，从那时起就可以使用I3C接口了（以便从设备响应DAA过程）。

在运行此演示和前面列出的从设备时，不会遇到这种情况。但是，代码中有一个占位符，使得此应用笔记也可用于此类设置。只需找到调用add_I2C_traffic()例程的代码，就可以根据系统的需求对其进行修改。主菜单选项C和D（总线速度演示和I3C从设备轮询演示）已准备好处理此情况。

此AN中实现的附加I2C传输示例从ICM-42688的I2C地址0x69（可在RT685音频板和LPC55S36 EVK板上使用）和寄存器地址0x75（WHO_AM_I寄存器）处执行了一个I2C读取的操作。演示代码不会验证此寄存器读数是否为0x47（应该是）。

场景2：I3C主设备与共享同一总线的I2C和I3C设备进行通信（“混合快速总线”）

一旦具有I3C功能的设备获取了分配到的动态地址，它就会仅响应I3C传输。例如，如果RT685音频/LPC55S36 EVK的板载ICM-42688（硬件确定的I2C地址0x69）被分配了0x09的动态地址，则此时它将仅响应寻址到0x09的I3C传输，并完全忽略任何试图对I2C地址0x69进行的通讯。

基于RT600/LPC55S36的I3C主设备可以和那些与I3C设备共享总线的I2C设备通信，不会有问题。在本应用笔记中，可以在主菜单选项A和B（恩智浦/Invensense传感器演示）中生成所谓的“虚拟”I2C传输。DUMMY_I2C_TRAFFIC_ENABLE定义了对此类操作的控制。如果生成，则该I2C传输以I2C地址在0x51处的从设备为目标，尝试从其地址在0x75处的寄存器读取数据。由于不存在这样的I2C从设备，初始写入命令会收到NACKED，I2C传输会在此终止。

此处讨论的现有虚拟I2C传输可以很容易地修改来做其他事情。只需在main.c中搜索DUMMY_I2C_TRAFFIC_ENABLE即可查看更多详细信息。

14 RT600/LPC55S36 EVK板的测试结果

以下是全套的RT600/LPC55S36 EVK板的测试结果。所有可能的电路板/外部模块/主菜单场景都得到了运行，产生了一个参考框架，以便终端用户了解在运行此应用笔记演示工程时应该产生的结果。

14.1 RT685音频EVK板

RT685音频EVK板测试外部电路板的电源详情：

- 仅添加一个ICM-42688 (I2C地址0x68)：将该电路板的3V3和GND脚分别连接到RT685音频EVK板的J31.2和J31.12脚。
- 仅添加一个NXPTSEB：将NXPTSEB的J4.4/7脚 (+3V3/GND输入) 连接到RT685音频EVK板的J29.4和J29.7引脚 (真正的3.3V和GND；如果该特定GND脚不可用，则任何其他RT685音频EVK板的GND引脚都可用)。插入NXPTSEB的JP1[2:3]，以选择1.8 V工作模式。
- ICM-42688和NXPTSEB都添加：如上所述设置NXPTSEB电路板的电源。将ICM-42688的3V3和GND脚分别连接到NXPTSEB电路板的J10.1和J15.1脚。

表2. RT685音频EVK板演示的测试结果

RT685音频EVK板	无外部电路板	ICM-42688 (I2C地址0x68) ——在菜单中选择的外部ICM电路板	NXPTSEB I3C P3T1085 (0x48) 和P3T1755 (0x4C)	ICM-42688和NXPTSEB
恩智浦 /Invensense 演示 (ENTDAAs + SETDASA)	敲击OK	倾斜+敲击OK	敲击+温度OK	倾斜+敲击+温度OK
恩智浦 /Invensense演示 (仅SETDASA)	敲击OK	倾斜+敲击OK	敲击+温度OK	倾斜+敲击+温度OK
总线速度演示	1/2：无需输入任何地址 PASSED (通过)	1/2：必须至少输入地址68或69中的一个 PASSED (通过)	1/2：无需输入任何地址 FAILED (失败) @ 12.5MHz	1/2：必须输入地址69 FAILED (失败) @ 12.5MHz
I3C从设备轮询演示	无需输入任何地址OK	1/2：必须至少输入地址68或69中的一个 PASSED (通过)	无需输入任何地址OK	1/2：必须输入地址69；OK

14.2 RT685 EVK板

RT685 EVK板测试外部电路板电源的详细信息如下：

- 添加一个ICM-42688 (I2C地址0x68)：将该电路板的3V3和GND脚分别连接到RT685 EVK板的J31.2和J31.12脚。

- 仅添加一个NXPTSEB：将NXPTSEB的J4.4/7引（+3V3/GND输入）连接到RT685 EVK板的J29.4和J29.7脚（真正的3.3 V和GND；如果该特定GND脚不可用，则任何其他RT685 EVK的GND脚都可用）。插入NXPTSEB的JP1[2:3]，以选择1.8 V工作模式。
- ICM-42688和NXPTSEB都添加：如上所述设置NXPTSEB电路板的电源。将ICM-42688的3V3和GND脚分别连接到NXPTSEB电路板的J10.1和J15.1脚。

表3. RT685 EVK板演示的测试结果

RT685 EVK板/通用RT600电路板	ICM-42688 (I2C地址0x68) ——在菜单中选择的外部ICM电路板	NXPTSEB I3C P3T1085 (0x48) 和 P3T1755 (0x4C)	ICM-42688和NXPTSEB
恩智浦/Invensense演示 (ENTDA+SETDASA)	敲击OK	温度OK	敲击OK
恩智浦/Invensense演示 (仅SETDASA)	倾斜OK	温度OK	倾斜OK
总线速度演示	1/2：无需输入任何地址 FAILED (失败) @ 12.5 MHz	1/2：无需输入任何地址 FAILED (失败) @ 12.5 MHz	1/2：无需输入任何地址 FAILED (失败) @ 12.5 MHz
I3C从设备轮询演示	无需输入任何地址 OK	无需输入任何地址 OK	无需输入任何地址 OK

14.3 LPC55S36 EVK板 @ 1.8 (9) V

LPC55S36 EVK板的1.8 (9) V电源详情如下：

- LPC55S36 JP1[2:3]：插入跳线帽JP1[2,3]，以选择I3C的1.9 V工作模式。
- 仅添加一个ICM-42688 (I2C地址0x68)：将此电路板的3V3和GND脚连接到LPC55S36 EVK板的J7[7:8]脚。
- 仅添加一个NXPTSEB：将NXPTSEB的J4.4/6脚（+3V3和GND输入）连接到LPC55S36的JP1.1和JP3.3脚（真正的3.3 V和GND；如果该特定GND引脚不可用，则任何其他LPC55S36的GND脚都可用）。插入NXPTSEB的JP1[2:3]，以选择1.8 V工作模式。
- ICM-42688和NXPTSEB都添加：使用上述详细信息，因为这两块电路板可独立供电。

表4. LPC55S36 EVK板1.9 V演示的测试结果

LPC55S36 EVK板 I3C @ 1.9 V	无外部电路板	ICM-42688 (I2C地址0x68)	NXPTSEB P3 T1085/75 (I2C地址0x48/4C)	ICM-42688和NXPTSEB
恩智浦/Invensense演示 (ENTDA+SETDASA)	敲击OK	FAILED (失败) 通过了DAA过程， 但外部电路板 “WHO AM I” 读取失败	敲击 + 温度 OK	FAILED (失败) 通过了DAA过 程，但外部电 路板 “WHO AM I” 读取失败
恩智浦/Invensense演示 (仅SETDASA)	敲击OK	FAILED (失败) 外部电路板PID读 取失败	敲击 + 温度 OK	FAILED (失败) 外部ICM电路板 PID读取失败

表4. LPC55S36 EVK板1.9 V演示的测试结果 (续)

LPC55S36 EVK板 I3C @ 1.9V	无外部电路板	ICM-42688 (I2C地址0x68)	NXPTSEB P3 T1085/75 (I2C地址0x48/4C)	ICM-42688和NXPTSEB
总线速度演示	1/2 : 无需输入任何地址 FAILED (失败) @ 12.5 MHz	FAILED (失败) 如果输入为68, 则外部电路板PID读取失败; 如果输入为69, 则会通过DAA过程, 但测试的第一个时钟配置会失败	1/2 : 无需输入任何地址 FAILED (失败) @ 12.5 MHz	FAILED (失败) 如果输入为68, 则外部电路板PID读取失败; 如果输入为69, 则会通过DAA过程, 但测试的第一个时钟配置会失败
I3C从设备轮询演示	无需输入任何地址 OK	FAILED (失败) 外部电路板PID读取失败	无需输入任何地址 OK	FAILED (失败) 外部电路板PID读取失败

14.4 LPC55S36 EVK板 @ 3.3 V

LPC55S36 EVK板3.3 V电源的详细信息如下：

- LPC55S36 JP1[1:2]：插入跳线帽JP1[1,2]，以选择I3C的3.3 V工作模式。
- 仅添加一个ICM-42688 (I2C地址0x68)：将此电路板的3V3和GND脚连接到LPC55S36 EVK板的J7[7:8]脚。
- 仅添加一个NXPTSEB：将NXPTSEB的J4.4/6脚 (+3V3和GND输入) 连接到LPC55S36 EVK板的J7.7/8脚。插入NXPTSEB的JP1[1:2]，以选择3.3V工作模式。
- ICM-42688和NXPTSEB都添加：如上所述连接NXPTSEB。将ICM-42688电路板的3.3 V和GND脚连接到NXPTSEB的J10.1和J15.1脚。

表5. LPC55S36 EVK板3.3 V演示的测试结果

LPC55S36 EVK板 I3C @ 3.3V	无外部电路板	ICM-42688 (I2C地址0x68)	NXPTSEB P3 T1085/75 (I2C地址0x48/4C)	ICM-42688和NXPTSEB
恩智浦/Invensense演示 (ENTDA + SETDASA)	敲击OK	FAILED (失败) 会通过DAA过程, 但外部电路板“WHO AM I”读取会失败	敲击+温度OK	FAILED (失败) 会通过DAA过程, 但外部电路板“WHO AM I”读取会失败
恩智浦/Invensense演示 (仅SETDASA)	敲击OK	FAILED (失败) 外部电路板PID读取失败	敲击+温度OK	FAILED (失败) 外部ICM电路板PID读取失败
总线速度演示	1/2 : 无需输入任何地址 PASSED (通过)	FAILED (失败) 如果输入为68, 则外部电路板PID读取会失败; 如果输入为69, 则会通过DAA过程, 但测试的第一个时钟配置会失败	1/2 : 无需输入任何地址 FAILED (失败) @ 12.5MHz	FAILED (失败) 如果输入为68, 则外部电路板PID读取会失败; 如果输入为69, 则会通过DAA过程, 但测试的第一个时钟配置会失败
I3C从设备轮询演示	无需输入任何地址 OK	FAILED (失败) 外部电路板PID读取失败	无需输入任何地址 OK	FAILED (失败) 外部电路板PID读取失败

15 示例

以下示例展示的UART控制台输出和Saleae逻辑分析仪截图，可以运行各种菜单选项和硬件配置。

15.1 RT685音频EVK板运行主菜单选项A，“恩智浦演示（ENTDAA + SETDASA）”

该演示是只有一个从设备（即板载ICM42688，I2C地址0x69）可与I3C主设备（RT685）通讯时，一个典型的演示主菜单选项A的运行情况。

用户在指定环境（主板和外部模块）后，选择一个演示。在本示例中，选择的是选项A（“恩智浦/Invensense传感器演示（ENTDAA + SETDASA，如需要）”），并输入此演示所使用的I3C SCL速率。

当运行主菜单选项A时，只有当RT685音频/LPC55S36 EVK板（从而保证存在I2C地址为0x69的板载ICM-42688设备）已选择且连接了外部ICM42688电路板（预期I2C地址为0x68）时，才会发送I2C地址为0x69的SETDASA命令。由于ENTDAA命令无法处理两个ICM-42688设备，因此必须使用SETDASA命令处理其中的一个ICM-42688设备。当SETDASA命令完成后，演示代码会发布ENTDAA命令。

在ICM-42688设备数量不超过1的装置中运行主菜单选项A，则结果是仅执行ENTDAA命令，而根本不使用SETDASA命令。

在分配了动态地址后，演示会更新SCL的速率，并通过运行前述的GETPID速度测试来验证主从通讯仍然正常。如果没有检测到问题，则会进入while(1)无限循环，主设备轮询从设备的内部寄存器，查找是否有检测到的敲击。



图10. RT685音频EVK板，“恩智浦演示 (ENTDAA + SETDASA) ” 的截图

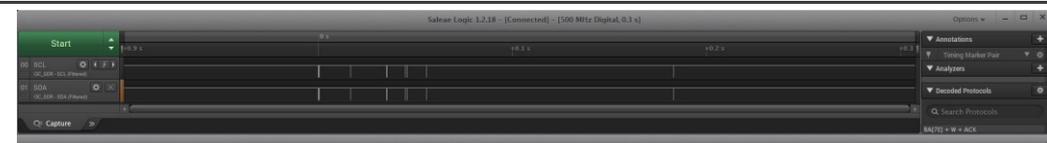


图11. RT685音频EVK板 (ENTDAA + SETDASA) 的测试概览

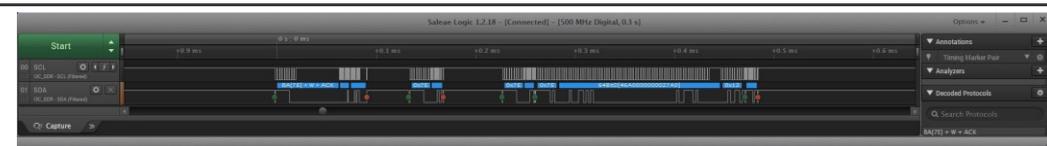


图12. RT685音频EVK板 (ENTDAA + SETDASA) 的DAA阶段

注意：在最开始，会发送一个DISEC + RSTDAA + ENTDAA的序列。

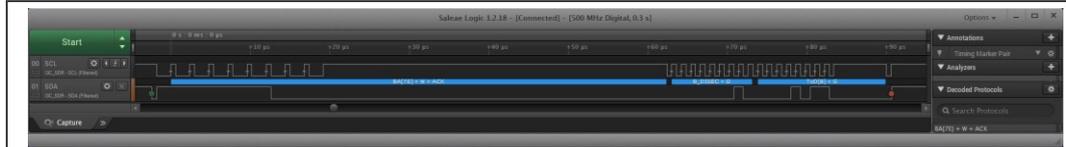


图13. RT685音频EVK板 (ENTDA A + SETDASA) 广播DISEC (热插入、主设备身份、从设备中断)

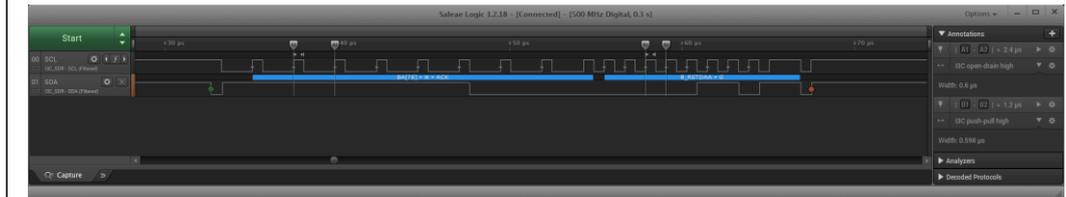


图14. RT685音频EVK板 (ENTDA A + SETDASA) 广播RSTDAA ; I3C的开漏和推挽高电平状态相等 (0.6微秒) ; 开漏和推挽SCL速率分别为417kHz和833kHz ; 详见A1/2和B1/2标记

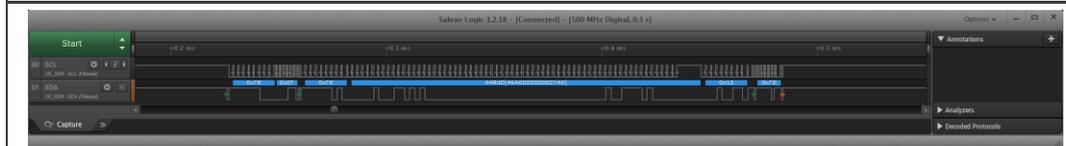


图15. RT685音频EVK板 (ENTDA A + SETDASA) 的ENTDA A概览



图16. RT685音频EVK板 (ENTDA A + SETDASA) 的ENTDA A——开始

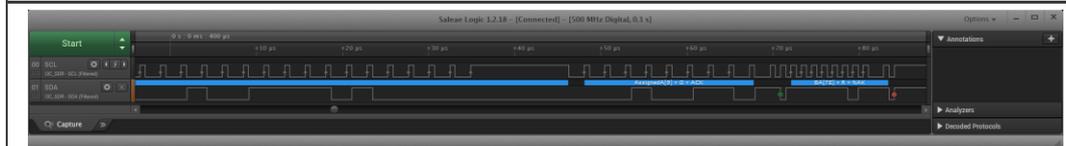


图17. RT685音频EVK板 (ENTDA A + SETDASA) 的ENTDA A——结束 ; 已分配动态地址 0x09 , 无其他可用设备 (0x7E NACKED)



图18. RT685音频EVK板 (ENTDA A + SETDASA) : 通过读取 “WHO_AM_I” 寄存器 (地址 0x75) 对板载ICM-42688 (动态地址0x09) 进行验证

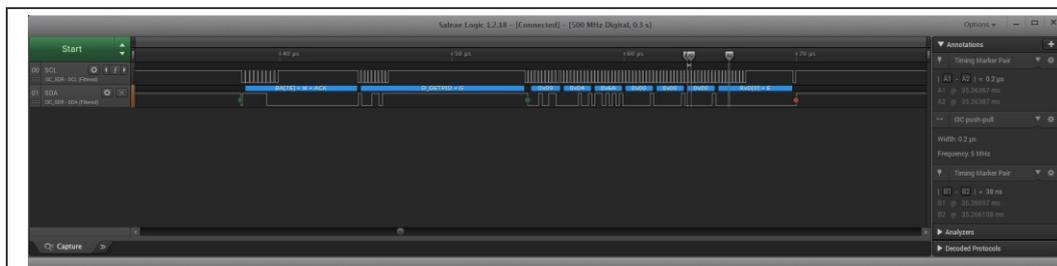


图19. RT685音频EVK板 (ENTDA + SETDASA) : ICM-42688 (动态地址0x09) 的PID测试 , 以I3C推挽SCL 5MHz的速率运行 (40ns高电平状态) ; 详见A1/2和B1/2标记

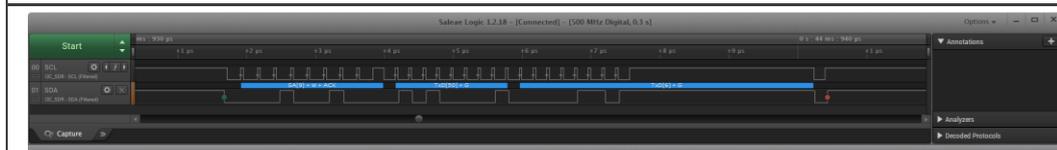


图20. RT685音频EVK板 (ENTDA + SETDASA) : 读取ICM-42688 (动态地址0x09) “敲击”检测器的初始化, 寄存器0x50写入了0x06

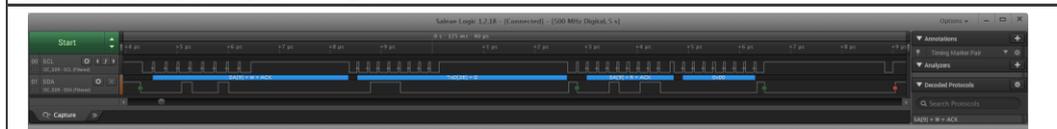


图21. RT685音频EVK板 (ENTDA + SETDASA) : 读取ICM-42688 (动态地址0x09) 的INT_STATUS3寄存器0x38 ; 0x00表示未检测到敲击

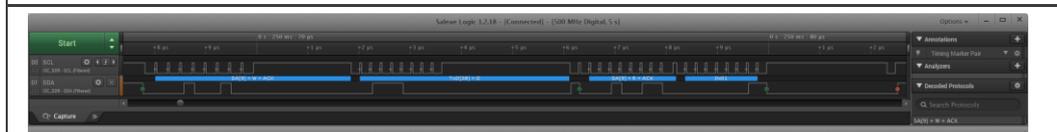


图22. RT685音频EVK板 (ENTDA + SETDASA) : 读取ICM-42688 (动态地址0x09) 的INT_STATUS3寄存器0x38 ; 0x01表示检测到敲击

15.2 RT685音频EVK板运行主菜单选项B , “恩智浦演示 (仅SETDASA) ”

这次使用SETDASA命令分配板载ICM-42688的动态地址 (0x40) (有关详细信息, 请参见截图“I3C网络概览”)。除了DAA过程外, 演示的其余部分都与主菜单选项A的事件顺序相匹配。



图23. RT685音频EVK板，“恩智浦演示（仅SETDASA）”的截图



图24. RT685音频EVK板（仅SETDASA），SETDASA用于将动态地址0x40分配给I2C地址为0x69的从设备

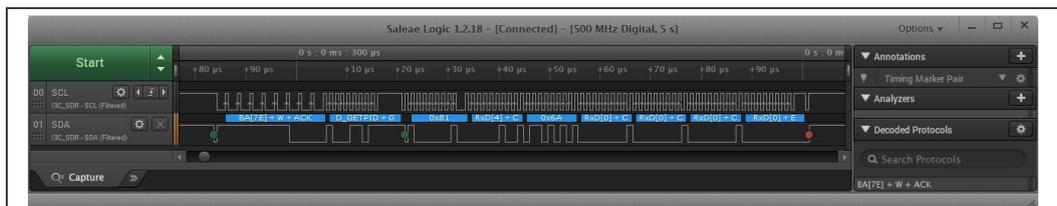


图25. RT685音频EVK板（仅SETDASA），GETPID命令发送到动态地址0x40；接收到PID：046A00000000

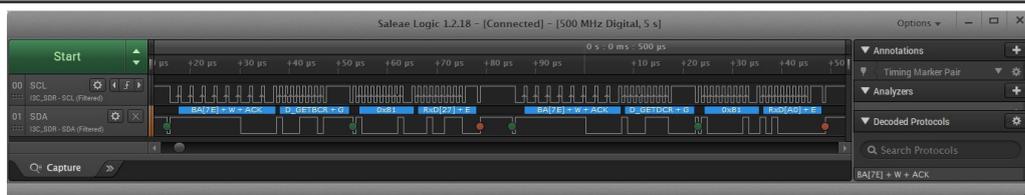


图26. RT685音频EVK板（仅SETDASA），GETBCR和GETDCR命令发送到动态地址0x40；接收到数据0x27和0xA0

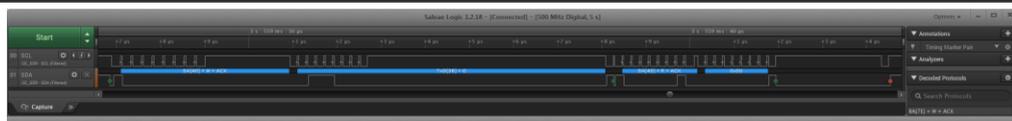


图27. RT685音频EVK板（ENTDAA + SETDASA），读取ICM-42688（动态地址0x40）的INT_STATUS3寄存器0x38；0x00表示未检测到敲击

15.3 RT685音频EVK板运行主菜单选项C，“用户辅助的地址分配总线速度演示（又名Shmoo图）”

在本示例中，对称和非对称时钟测试都会执行，并且两者都达到了12.5 Mbit/s的波特率。



图28. RT685音频EVK板总线速度演示，对称时钟测试示例，截图

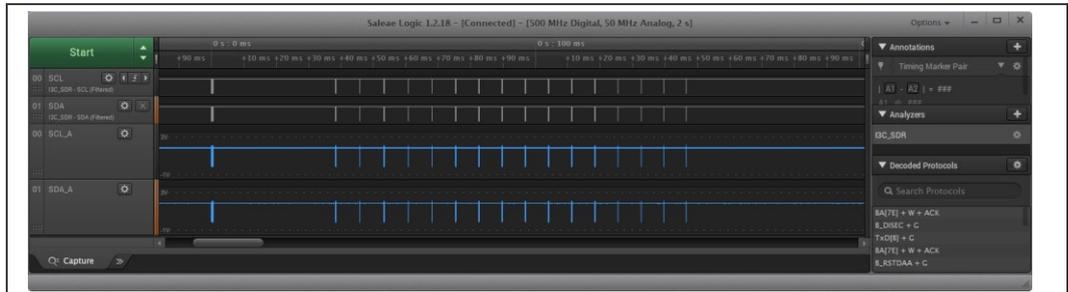


图29. RT685音频EVK板总线速度演示，对称时钟测试示例，初始化 + 16种SCL模式

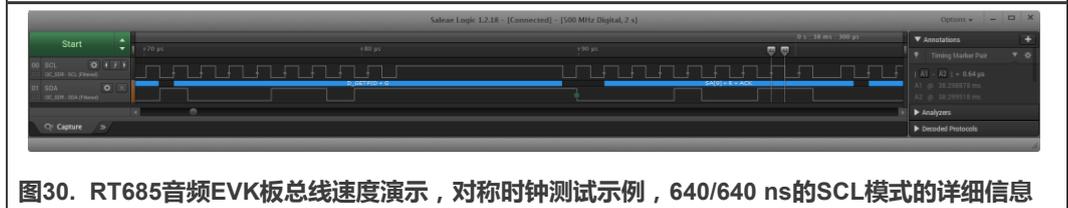


图30. RT685音频EVK板总线速度演示，对称时钟测试示例，640/640 ns的SCL模式的详细信息

注意：A1/2标记确认SCL高/低电平状态为640 ns。



图31. RT685音频EVK板总线速度演示，对称时钟测试示例，40/40 ns的SCL模式的详细信息



图32. RT685音频EVK板总线速度演示，非对称时钟测试示例，截图

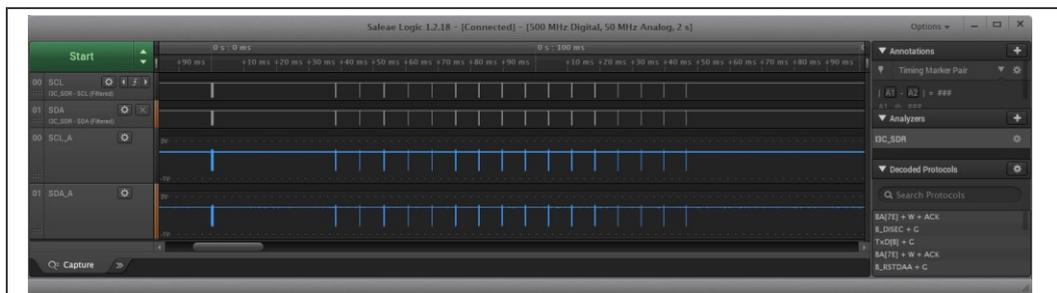


图33. RT685音频EVK板总线速度演示，非对称时钟测试示例，初始化+16种SCL模式

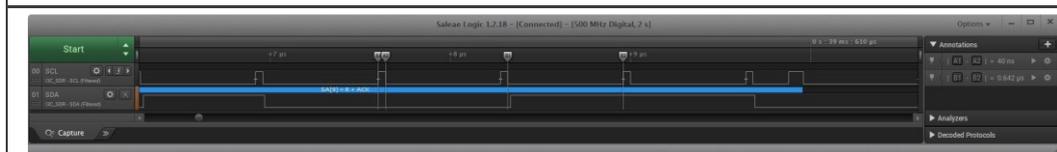


图34. RT685音频EVK板总线速度演示，非对称时钟测试示例，40/640 ns的高/低电平SCL模式的详细信息

注意：A1/2和B1/2标记确认SCL高/低电平状态为40/640 ns。

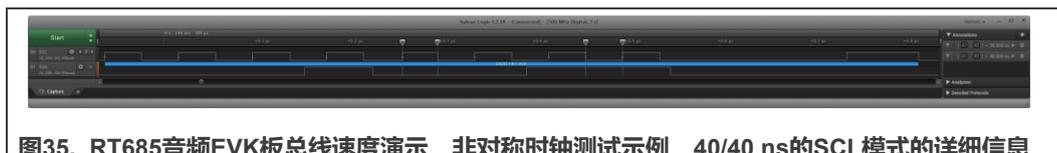


图35. RT685音频EVK板总线速度演示，非对称时钟测试示例，40/40 ns的SCL模式的详细信息

注意：A1/2和B1/2标记确认SCL高/低电平状态为40/40 ns。

15.4 RT685音频EVK板运行主菜单选项D，“用户辅助的地址分配I3C从设备轮询演示 (CCC GETSTATUS)”

在此装置中只有一个从设备，没有I2C地址的输入（因此不发送SETDASA命令），而ENTDAA命令用于识别从设备并为其分配动态地址0x09。在通过执行GETPID速度测试确认主设备和从设备能够以用户指定的SCL速率进行通讯后，代码使用GETSTATUS命令进入轮询从设备的无限循环。

由于从设备未初始化（作为倾斜/敲击检测器运行），因此GETSTATUS命令预计会读取全0。



图36. RT685音频EVK板从设备轮询演示，截图

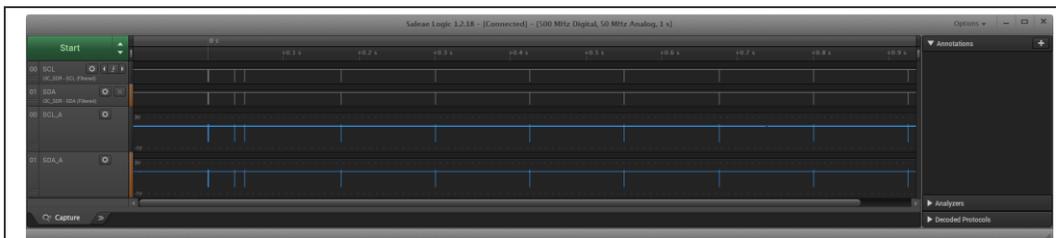


图37. RT685音频EVK板从设备轮询演示，概览（初始化 + 定期GETSTATUS读取），截图

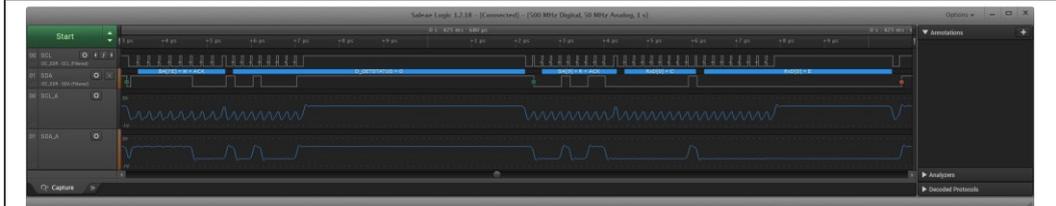


图38. RT685音频EVK板从设备轮询演示，GETSTATUS的详细信息

15.5 RT685音频EVK板运行主菜单选项A，“恩智浦演示（ENTDAA + SETDASA）”，其中主设备控制4个I3C从设备

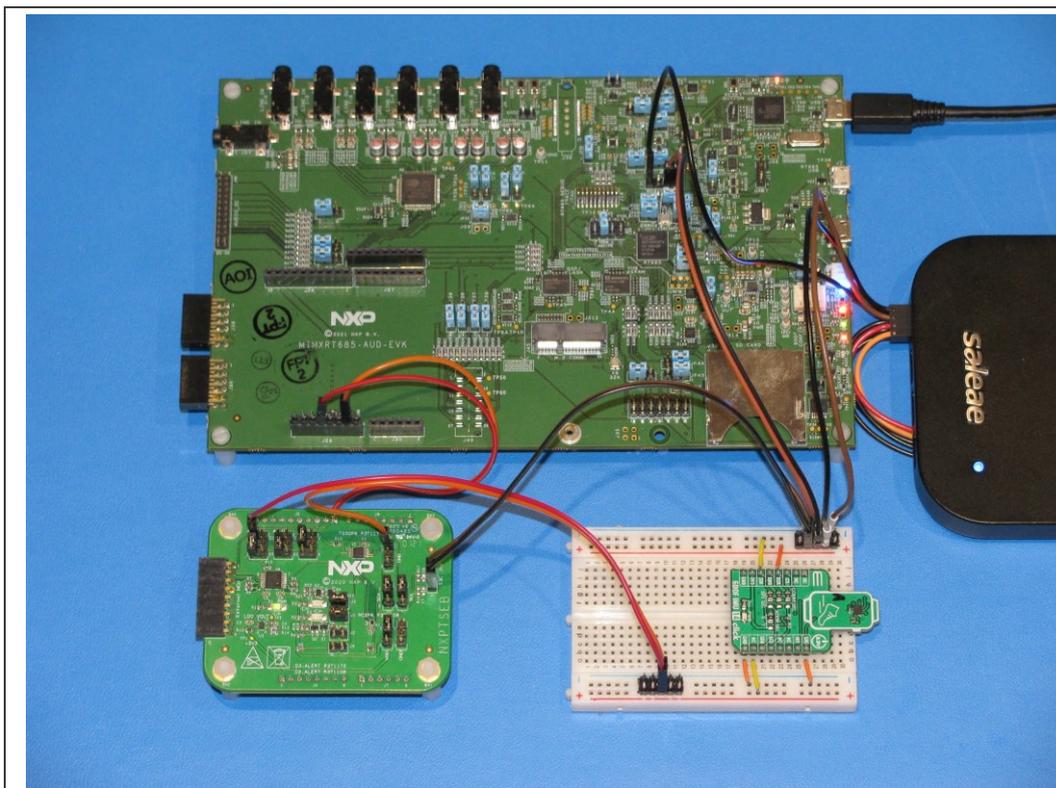


图39. RT685音频EVK板，配备2个外部模块/4个从设备网络

注意：NXPNTSEB P3T1085/P3T1755电路板（左下）通过RT685音频EVK板Arduino连接器J29的J4连接器供电（脚4和脚7，3.3V和GND）。NXPTSEB的JP1[2:3]将板载电源设置为1.8V。MikroElektronika ICM-42688电路板（右下）由NXPTSEB提供的1.8V电源供电（J10.1/J51.1-VDD/GND——连接到面包板底部的“+”和“-”电源轨线）。

RT685 音频EVK板的I3C总线引脚 (J18连接器, 脚1和脚2) 连接到面包板顶部的轨线, 就像是连接NXPTSEB、ICM-42688的I3C SCL和SDA线, Saleae探头0和1也是如此。ICM-42688板的顶部边缘脚3V3、CS和SDO使用MikroElektronika模块下的跳线连接到VDD和GND (按需)。

RT600主设备使用SETDASA命令为板载ICM-42688 (I2C地址0x69) 分配动态地址0x40, 并使用ENTDAA命令为其余3个从设备 (P3T1085/75和外部ICM-42688) 分配动态地址0x09/A/B。在各个从设备初始化之后, 通过运行PID速度测试来验证其波特率 (5 Mbit/s), 并进入一个无限轮询循环。

```

COM8 - Tera Term VT
File Edit Setup Control Window Help

Must select one of the following RT600 boards:
-----
[1] RT600 AUDIO EVK board from NXP
[2] other RT600 board

Selected MCU board: RT600 AUDIO EVK board from NXP
Do you have an NXP I3C NXPTSEB board (P3T1108 & P3T1175) attached (Y/N)? Yes
Do you have an external ICM-42688 board attached (Y/N)? Yes
I M P O R T A N T !
-----
The EVK on-board ICM-42688 I2C address is fixed to 0x69 in hardware
and can not be changed. Make sure the external ICM-42688 is wired so
that its I2C address is set to 0x68 for this demo to work!

Select I3C MCONFIG SKEW parameter from 0 to 7 (IP reset value is 0): 0

You are about to run a simple I3C master demo.
Please choose one of the following options:
[A] NXP/Invensense sensor(s) demo (ENTDAA + SETDASA if needed)
[B] NXP/Invensense sensor(s) demo (SETDASA only)
[C] user assisted address assignment bus speed demo (aka the Shmoo plot)
[D] user assisted address assignment I3C slave polling demo (CCC GETSTATUS)

run demo: A

Choose the I3C SCL rate to deploy:
[D] I3C SCL == 0.960 Mbit/s (symmetrical clk)
[E] I3C SCL == 1.000 Mbit/s (mixed bus spec)
[F] I3C SCL == 1.125 Mbit/s (mixed bus spec)
[G] I3C SCL == 1.166 Mbit/s (mixed bus spec)
[H] I3C SCL == 1.250 Mbit/s (mixed bus spec)
[I] I3C SCL == 1.250 Mbit/s (mixed bus spec)
[J] I3C SCL == 1.333 Mbit/s (mixed bus spec)
[K] I3C SCL == 12.500 Mbit/s (mixed bus spec)

I3C SCL rate: H
hello RT600 world!
Main clock [Hz]: 500000000
CPU clock [Hz]: 250000000

Initial baudrate setting:
-----
I3C clk [Hz]: 25000000 (40.0 ns)
PP rate [Hz]: 833333 (H/L ratio: 15/15)
OD rate [Hz]: 416666 (H/L ratio: 15/45) ODHPP = 1
I2C OD [Hz]: 138888

I3C network overview
-----
# Dv St UdID Part_Num BCR DCR
0 08 1A 0345 00000000 60 00 <- the master
1 40 69 0235 00000000 27 00
2 09 00 0110 15290090 03 63
3 0A 00 0110 152A0098 03 63
4 0B 00 0235 00000000 27 00

Updated baudrate setting:
-----
I3C clk [Hz]: 25000000 (40.0 ns)
PP rate [Hz]: 5000000 (H/L ratio: 1/4)
OD rate [Hz]: 4166666 (H/L ratio: 1/5) ODHPP = 1
I2C OD [Hz]: 357142

CCC GETPID based speed test passed.

Initializing P3T1108... Read/update P3T1108 registers Conf, Tlow, Thigh
22 10, b5 0, 7f f0
2 10, b5 0, 7f f0
...done

Initializing P3T1175... Read/update P3T1175 registers Conf, Tlow, Thigh
28 4b 0, 50 0
58 4b 0, 50 0
...done

Initializing ICM-426xx (dynamic address = 0xB) as a tilt detector... done!
Initializing ICM-426xx (dynamic address = 0x40) as a tap detector... done!

```

图40. 配备2个外部模块/4个从设备网络的RT685音频EVK板，初始化截图1/2

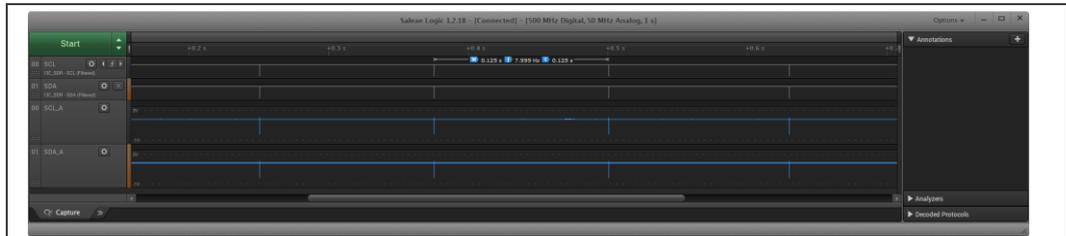


图43. 配备2个外部模块/4个从设备网络的RT685音频EVK板，定期轮询

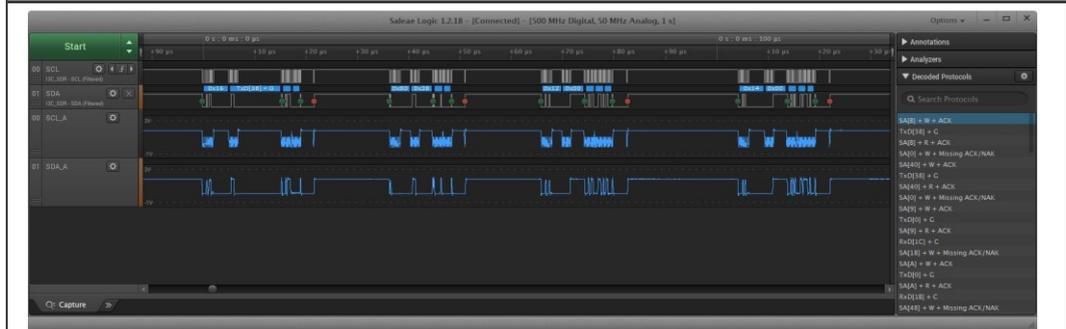


图44. 配备2个外部模块/4个从设备网络的RT685音频EVK板，组轮询详细信息

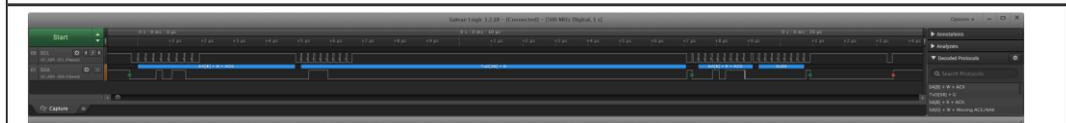


图45. 配备2个外部模块/4个从设备网络的RT685音频EVK板，轮询外部ICM-42688（动态地址0x0B），读取其寄存器0x38

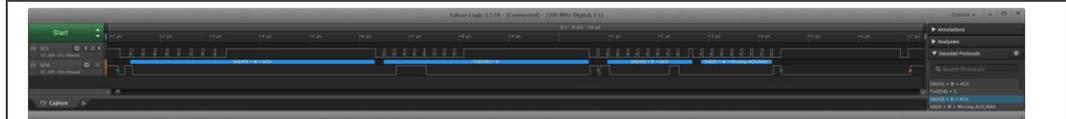


图46. 配备2个外部模块/4个从设备网络的RT685音频EVK板，轮询板载ICM-42688（动态地址0x40），读取其寄存器0x38

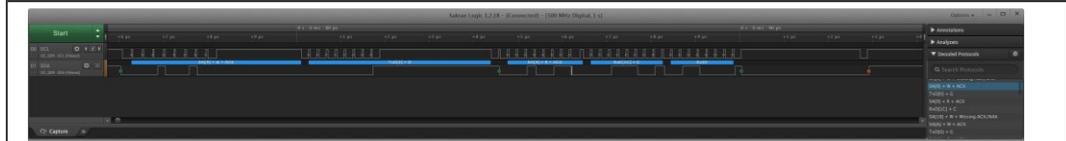


图47. 配备2个外部模块/4个从设备网络的RT685音频EVK板，轮询P3T1085温度传感器（动态地址0x09），从寄存器地址0x00开始读取两个字节

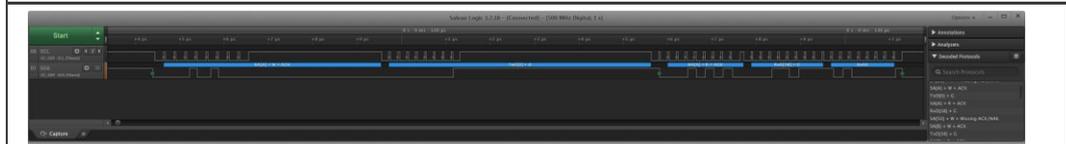


图48. 配备2个外部模块/4个从设备网络的RT685音频EVK板，轮询P3T1755温度传感器（动态地址0x0A），从寄存器地址0x00开始读取两个字节

15.6 LPC55S36 EVK板采用博世BMP580运行主菜单选项C和D，“总线速度/从设备轮询演示”

在本例中，Bosch Sensortec压力传感器BMP580作为一个I3C设备添加到装置中。在配置之前，请确保I3C主设备检测到BMP580的存在，并能够建立可靠的通讯。此时用户可以采用“总线速度”和“从设备轮询”演示菜单选项。

在运行总线速度演示（部署了非对称时钟模式）后，图49中所示的硬件设置可以使I3C主设备和从设备以低至 $40\text{ ns} + 80\text{ ns} = 120\text{ ns}$ 的时钟速率周期进行通讯。最后一次尝试的时钟配置（相当于 12.5 MHz ）失败了（高/低电平时钟状态分别为 40 ns ）。

在确定了最大SCL速率后，可以从菜单运行从设备轮询选项。不超过 8.333 MHz （ 120 ns ）的任意SCL频率都可以使用，选择了 5 MHz 。图50中的控制台截图所示为发现的设备（板载ICM-42688和BMP580），以及从设备发现过程中和发现之后，使用CCC GETSTATUS命令轮询ICM-42688和BMP580，LPC55S36 I3C主设备进入无限循环时所使用的时钟的详细信息。

在成功运行总线速度和从设备轮询演示后，系统集成人员就有足够的信息来准备目标应用的硬件和软件，从而可以将BMP580添加到其中来。下一步是配置BMP580设备。

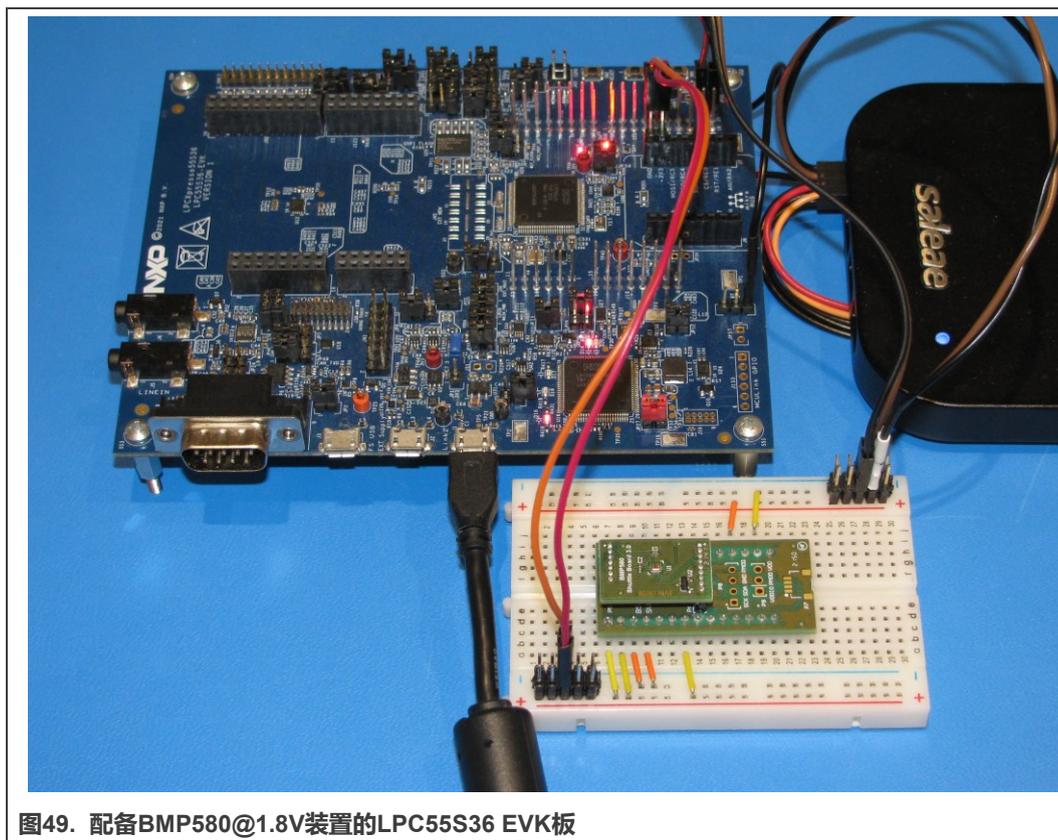


图49. 配备BMP580@1.8V装置的LPC55S36 EVK板

注意：插入LPC55S36的JP27跳线帽（见电路板右下角；红色跳线帽），以使微控制器能够在SWD断开的情况下运行I3C总线和UART控制台。通过插入JP1[2:3]（电缆插入之处的USB连接器附近的蓝色跳线帽），将电路板VCC_3V3_1V9的电源设置为 1.9 V 。

mikroBUS的J7.7接头标记为“3v3”——该脚连接到VCC_3V3_1V9电源域。在插入JP1[2:3]后，该脚可提供1.9 V电压。这就是为什么转接板（承载BMP580）的电源线（VDD/VDDIO和GND）分别连接到J7.7和J7.8的原因。

该面包板的底部电源轨为I3C传感器供电，而顶部电源轨则为I3C总线供电。

该面包板底部边缘的5对跳线根据需要连接博世BMP580转接板的VDD、VDDIO、GND、SDO（将I2C地址的LSB设置为0）和CS（设置为VDD以启用I3C）。此板顶部边缘的跳线将转接板的SCL和SDA信号连接到I3C总线。



图50. 配备BMP580@1.8V的LPC55S36 EVK板的总线速度演示，在12.5 MHz时失败

简便易行地将RT600/LPC55S30用作I3C主设备

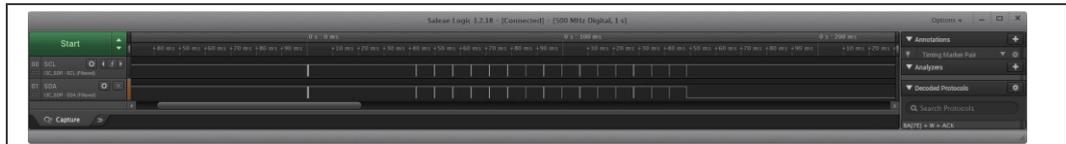


图51. 配备BMP580@1.8V的LPC55S36 EVK板的总线速度演示，概览



图52. 配备BMP580@1.8V的LPC55S36 EVK板的总线速度演示，网络初始化



图53. 配备BMP580@1.8V的LPC55S36板的总线速度演示，16个GEPTPID序列

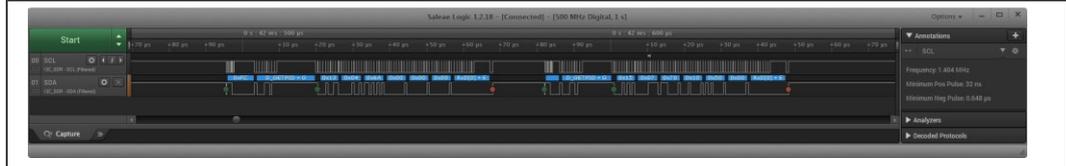


图54. 配备BMP580@1.8V的LPC55S36 EVK板的总线速度演示，第一个GEPTPID序列通过

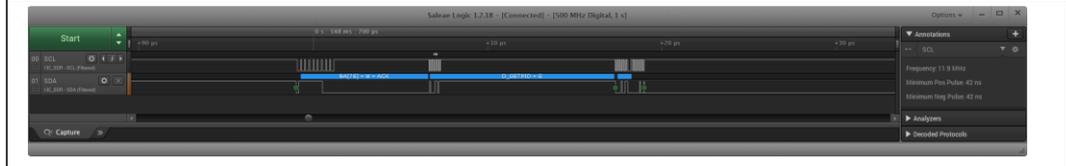


图55. 配备BMP580@1.8V的LPC55S36 EVK板的总线速度演示，第16个GEPTPID序列失败



图56. 配备BMP580@1.8V的LPC55S36 EVK板的从设备轮询演示，截图

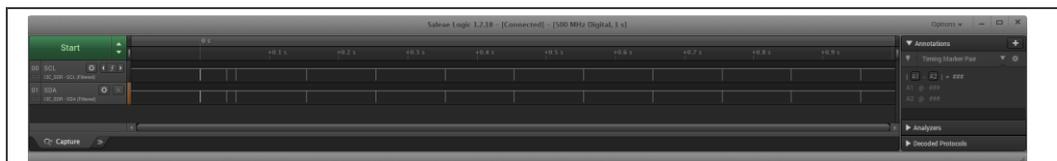


图57. 配备BMP580@1.8V的LPC55S36 EVK板的从设备轮询演示，概览

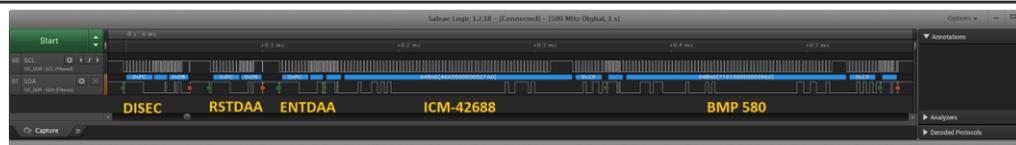


图58. 配备BMP580@1.8V的LPC55S36 EVK板的从设备轮询演示，网络初始化

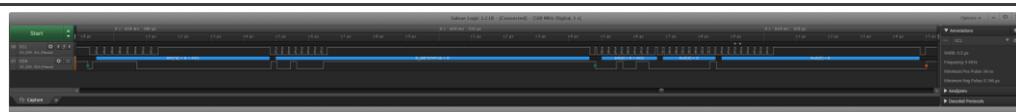


图59. 配备BMP580@1.8V的LPC55S36 EVK板的从设备轮询演示，ICM-42688 GETSTATUS

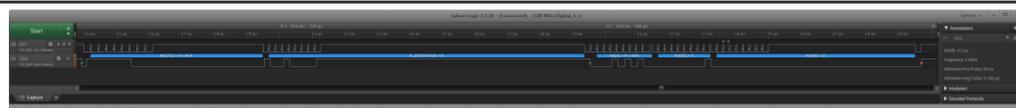


图60. 配备BMP580@1.8V的LPC55S36 EVK板的从设备轮询演示，BMP580 GETSTATUS

16 结论

I3C是构建在众所周知且广泛采用的I2C之上的一个强大接口。它增加了许多新功能，帮助用户设计比以往更先进的传感器应用。

要建立一个I3C网络，有两个同样重要的方面——硬件和软件组件。RT600和LPC55S36 MCU硬件集成了I3C的IP和恩智浦SDK的I3C驱动程序，为此类新应用的开发和运行提供了软件基础。

可以肯定地说，由于恩智浦I3C的IP硬件和MIPI规范二者的复杂程度，恩智浦SDK的I3C驱动程序将在可预见的未来不断演进。随着每个驱动程序新版本的发布，将支持更多的CCC命令，同时I3C功能的覆盖范围也将不断扩展。SDK的I3C驱动程序示例表明了对关键I3C功能的支持，而恩智浦的I3C应用笔记则包含了更多的详细信息。

在本应用笔记中，使用了3块电路板（RT685音频EVK板、RT685 EVK板和LPC55S36 EVK板）进行开发和测试。这3块电路板使用户能够通过一个板载接头访问I3C信号，探测SCL和SDA线路，并将外部I3C设备连接到总线。

RT685音频EVK板和LPC55S36 EVK板配有一个可用的板载I3C设备（ICM-42688）。这简化了用户进入I3C世界的第一步。本应用笔记首先设计了RT685 EVK板，并将有关I3C接口的经验教训纳入新的RT685音频EVK板和LPC55S36 EVK板的设计中。关键改进是在I3C的走线布局方面。

正如本应用笔记中提供的EVK板测试结果所示，RT685 EVK板在与某些I3C模块通讯时可能会遇到困难（例如，即使ICM-42688电路板设置了正确的I2C地址，也无法同时让两块MikroElektronika ICM-42688板与之协同工作）。LPC55S36 EVK板要求在未连接调试器的情况下运行I3C应用程序，这可能会影响应用程序的开发和测试。

事实证明，对于一个恩智浦I3C设备的新用户来说，RT685音频EVK板是一个最为多用途的环境。它为硬件和软件开发人员提供了一个易于搭建、测试和调试的生态系统。板载ICM-42688让用户在开箱后就能快速运行SDK的I3C驱动程序示例，标准的0.1英寸间距接头便于使用逻辑分析仪或类似的设备进行总线探测。

本应用笔记代码源文件可用于RT600系列产品和LPC55S36 MCU。这里描述的典型使用场景是使用标准构建模块构建的，例如设置I3C主设备、配置I3C引脚、部署特定的I3C比特率、执行动态地址分配、通用主从设备读/写传输等等。可以很容易地重复使用相同的代码块来创建一个新的I3C应用程序，用以处理一组不同的I3C从设备和任务。

17 修订历史

修订历史

版本号	日期	说明
第0版	2023年1月17日	初始版本

18 附录

18.1 将I3C驱动程序组件添加到“hello world”类型的工程中

使用此应用笔记的演示工程文件为您的工程提供I3C支持。

将演示工程“drivers”文件夹中的fsl_i3c.c/h文件复制到“helloworld”“drives”文件夹中。

在“helloworld”“component”文件夹下创建“i3c_bus”文件夹；将此应用笔记“i3c_bus”的fsl_component_i3c.c/h和fsl_component_i3c_adapter.c/h文件复制到“hello_world”“component/i3c_bus”文件夹中。

当您的工程在MCUX环境中处于活动状态时，转到界面顶部并打开：

Project→Properties→C/C++ Build→Settings

转到“Include paths”并点击绿色的“+”号。要输入“Directory:”行的内容，请选择“Workspace...”，然后打开工程文件结构，选中“component”下的“i3c_bus”并点击“OK”。之后，点击“Apply and Close”。

下次重建工程时，所有必需的I3C文件就都在其中了。

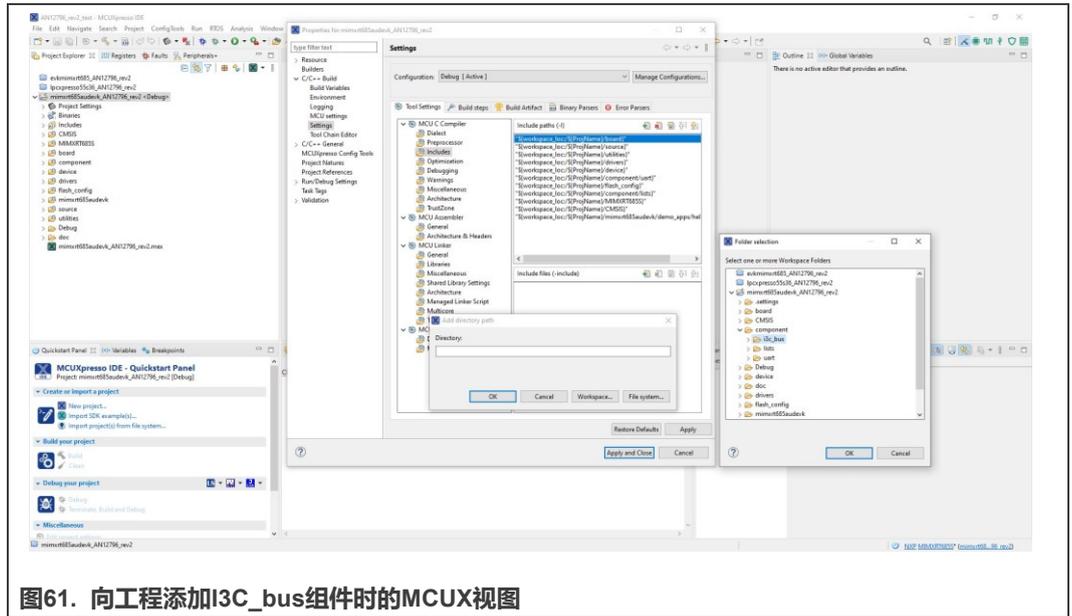


图61. 向工程添加I3C_bus组件时的MCUX视图

19 Legal information

19.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

19.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.nxp.com.cn/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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.

19.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

目录

1	介绍	2	15.4	RT685音频EVK板运行主菜单选项D, “用户辅助的地址分配I3C从设备轮询演示 (CCC GETSTATUS) ”	32
2	推荐读物	2	15.5	RT685音频EVK板运行主菜单选项A, “恩智浦演示 (ENTDA + SETDASA) ”, 其中主设备控制4个I3C从设备	34
3	i.MX RT685/LPC55S36评估板的I3C总线	2	15.6	LPC55S36 EVK板采用博世BMP580运行主菜单选项C和D, “总线速度/从设备轮询演示”	39
4	演示的软硬件要求	3	16	结论	44
4.1	演示的软件要求	3	17	修订历史	45
4.2	演示的硬件要求	4	18	附录	45
4.2.1	EVK的板载ICM-42688	5	18.1	将I3C驱动程序组件添加到 “hello world” 类型的工程中	45
4.2.2	带P3T1085和P3T1755温度传感器的恩智浦NXPTSEB板	5	19	法律声明	47
4.2.3	MIKROE-4237—外部ICM-42688	6			
4.2.4	带适配器的BMP580转接板3.0	7			
5	构建演示	7			
6	运行演示	8			
7	动态地址分配 (DAA) ——此演示如何使用ENTDAA和SETDASA命令	9			
8	演示详情	12			
8.1	主菜单选项A和B: 恩智浦Invensense传感器演示 (ENTDA + SETDASA, 如需要) 和 (仅SETDASA) ——实施细节	12			
8.2	主菜单选项C和D: 用户辅助地址分配总线速度演示 (又名Shmoo图) 和I3C从设备轮询演示 (CCC GETSTATUS) ——实施细节	14			
9	如何配置RT600/LPC55S3x I3C管脚	15			
10	设置I3C波特率	15			
11	使用CCC GETPID命令进行压力测试	18			
12	为什么需要I3C_BusMasterCreate_basic()例程 ?	18			
13	I2C和I3C传输是否可以在同一条总线上共存 ?	19			
14	RT600/LPC55S36 EVK板的测试结果	20			
14.1	RT685音频EVK板	20			
14.2	RT685 EVK板	20			
14.3	LPC55S36 EVK板 @ 1.8 (9) V	21			
14.4	LPC55S36 EVK板 @ 3.3 V	22			
15	示例	23			
15.1	RT685音频EVK板运行主菜单选项A, “恩智浦演示 (ENTDA + SETDASA) ”	23			
15.2	RT685音频EVK板运行主菜单选项B, “恩智浦演示 (仅SETDASA) ”	26			
15.3	RT685音频EVK板运行主菜单选项C, “用户辅助的地址分配总线速度演示 (又名Shmoo图) ” ..	28			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.