应用指南AN-102
InnoSwitch3-Pro/InnoSwitch4-Pro产品系列

Arduino代码库概览与指南

简介

本文档将介绍InnoSwitch3-Pro和InnoSwitch4-Pro产品系列Arduino库的应用。该代码被设计为可与不同的微控制器平台高度兼容。由于使用兼容Arduino的C++语言编写而成，可使用户更容易理解并根据自己的需要修改代码。通过本指南，用户可以充分了解如何实现器件与Arduino等简单的微控制器的协同工作。

InnoSwitch3-Pro

InnoSwitch3-Pro器件适合于输出电压及电流需要精细（10mV、50mA）调整的AC/DC电源应用。典型的实现方案包括一个系统微处理器或专用微处理器，其I²C端口可用于配置、控制和监测电源子系统的运行情况。uVCC引脚在独立方案（如USB PD适配器或充电器）中为微处理器提供偏置供电。

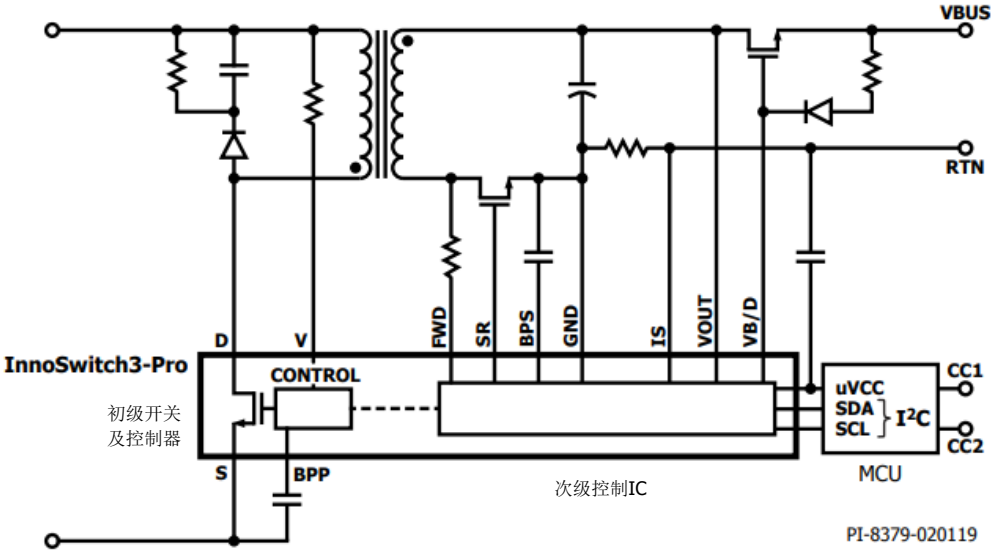


图1. InnoSwitch3-Pro的电路原理图

InnoSwitch4-Pro

与InnoSwitch3-Pro相比，InnoSwitch4-Pro上的命令和遥测寄存器有所更新。这些特性增加了灵活性并改善了故障响应。

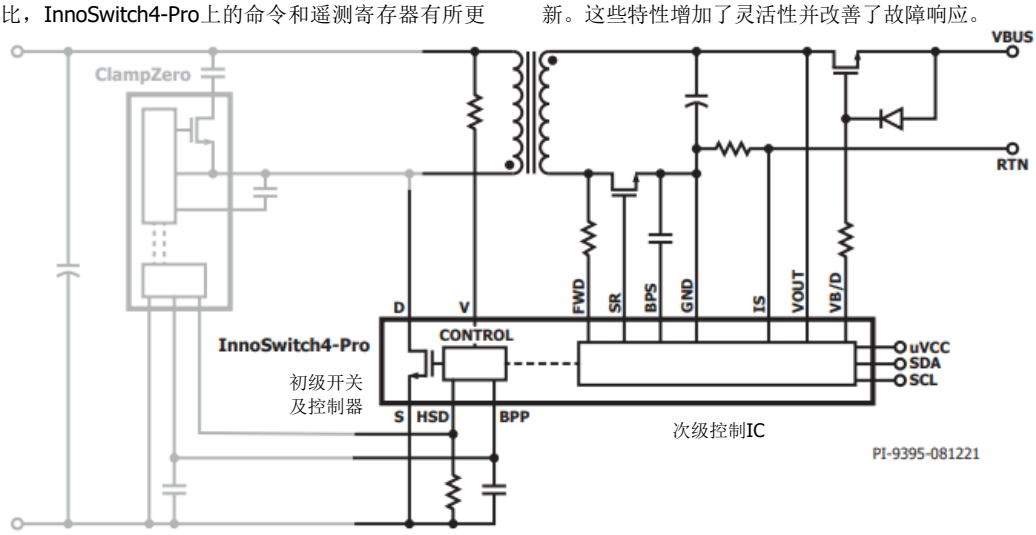


图2. InnoSwitch4-Pro的电路原理图

系统要求

硬件

InnoSwitch3-Pro和InnoSwitch4-Pro可通过板载微控制器进行控制，也可通过接口头由外部I²C主器件进行控制。本应用演示未使用板载微控制器，而是使用Arduino Uno作为I²C主器件，InnoSwitch3-Pro/InnoSwitch4-Pro作为从器件。本次演示将使用InnoSwitch3-Pro的参考设计RDK-641和InnoSwitch4-Pro的参考设计RDR-961进行。

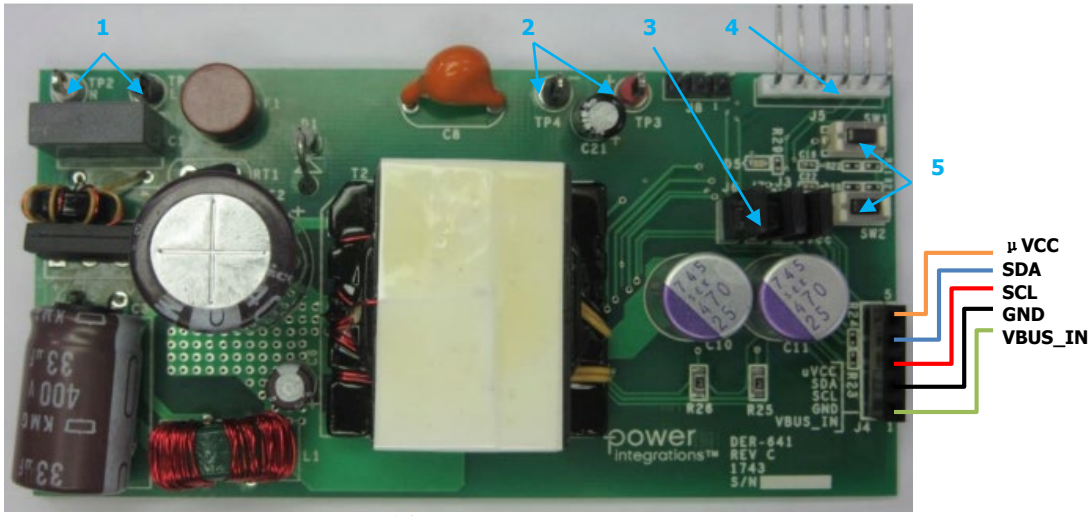


图3. RDK-641

编号	描述	标号
1	交流输入端子	TP1、TP2
2	直流输出端子	TP3、TP4
3	uVcc和I²C绝缘跳线	J3、J6、J7
4	PIC烧录头	J5
5	按钮	SW1、SW2

表1. RDK-641元件说明

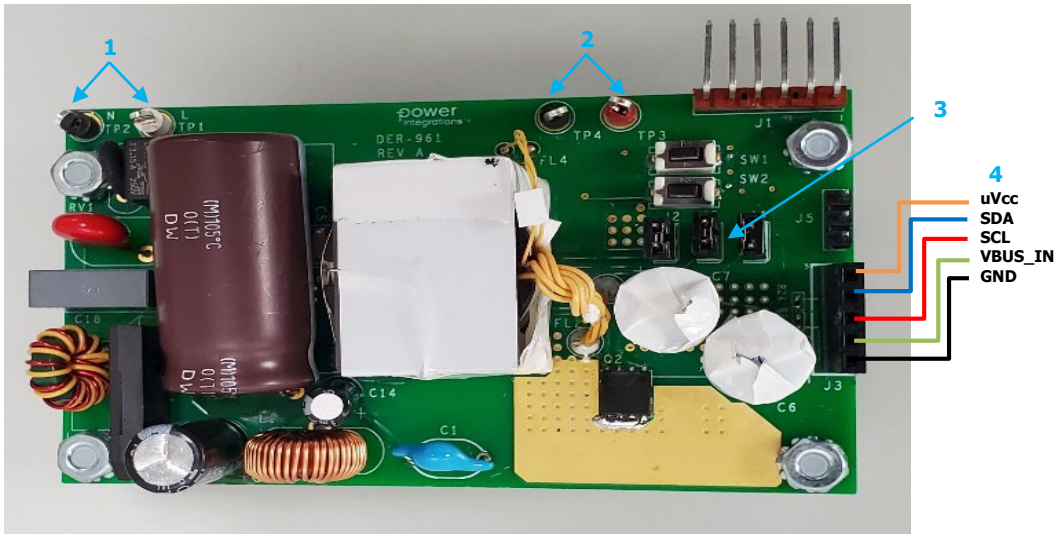


图4. DER-961

编号	描述	标号
1	交流输入端子	TP1、TP2
2	直流输出端子	TP3、TP4
3	uVcc和I²C跳线端子	J2、J4、J6
4	外部I²C接口接头	J3

表2. RDR-961元件说明

Arduino

本文档中使用的Arduino IDE版本为1.8.16，搭配Arduino Mega 2560微控制器板，因为其他某些库示例需要更多闪存。Arduino Uno板也可用于InnoSwitch3-Pro和InnoSwitch4-Pro库示例，这些示例不需要大量内存。

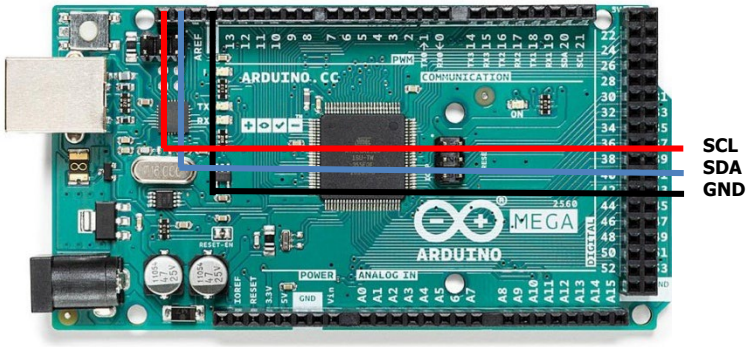


图5. Arduino Mega 2560

代码库

InnoSwitch3-Pro和InnoSwitch4-Pro Arduino代码库包含InnoSwitch3-Pro和InnoSwitch4-Pro的驱动程序和固件程序示例。这些示例的功能非常广泛，包括从简单的器件初始化到允许用户直接访问每个单独的命令和遥测寄存器。可以从以下链接下载InnoSwitch3-Pro和InnoSwitch4-Pro Arduino库。

适用于Arduino的InnoSwitch3-Pro和InnoSwitch4-Pro产品系列代码库和API | Power Integrations, Inc.
https://www.power.com/design-support/downloads/innoswitch3-pro-code-library-and-api-arduino

库的安装

前往Arduino IDE并转到*Sketch（固件程序）菜单> Include Library（包含库）> Add .ZIP Library（添加.ZIP库）*。选择并打开Arduino .zip库。用户可以在*Sketch（固件程序）菜单> Include Library（包含库）*菜单中检查是否已安装库。此外，还可以在下面的文件路径中看到该库。

- *C:|Users|username|Documents|Arduino|libraries|*

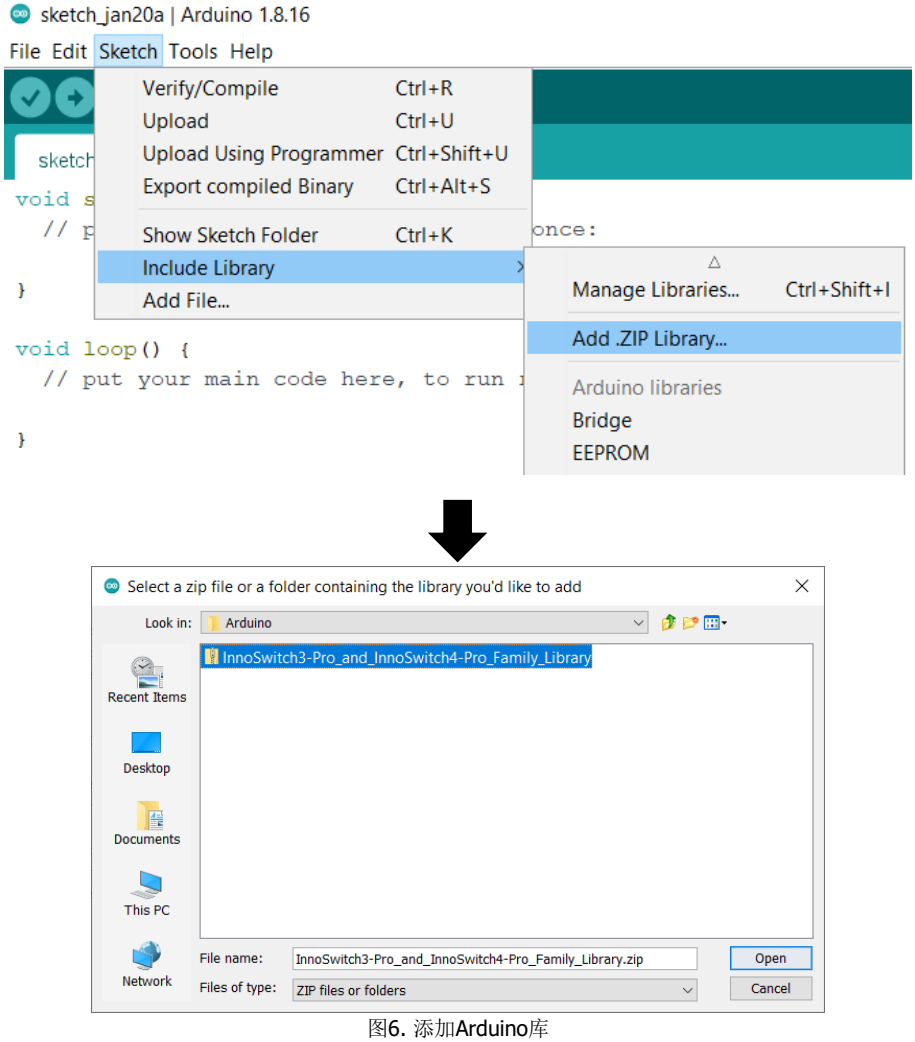


图6. 添加Arduino库

删除其他InnoSwitch3-Pro/InnoSwitch4-Pro库

需要删除旧版本的InnoSwitch3-Pro/InnoSwitch4-Pro库，因为它可能会导致Arduino IDE使用新库编译错误的源文件。使用旧的源文件可能会导致不同的问题，因此最好只使用一个版本的Arduino库来避免这种可能性。要删除其他Arduino库，请转到*File（文件）> Preferences（首选项）*中的文件路径，然后删除文件夹。在下面的示例中，文件路径为：*C:\Users\users\Documents\Arduino*。转到库并删除可能名为“InnoSwitch3-Pro_Library”的旧库文件夹。这将避免编译时出现冲突的标头文件。

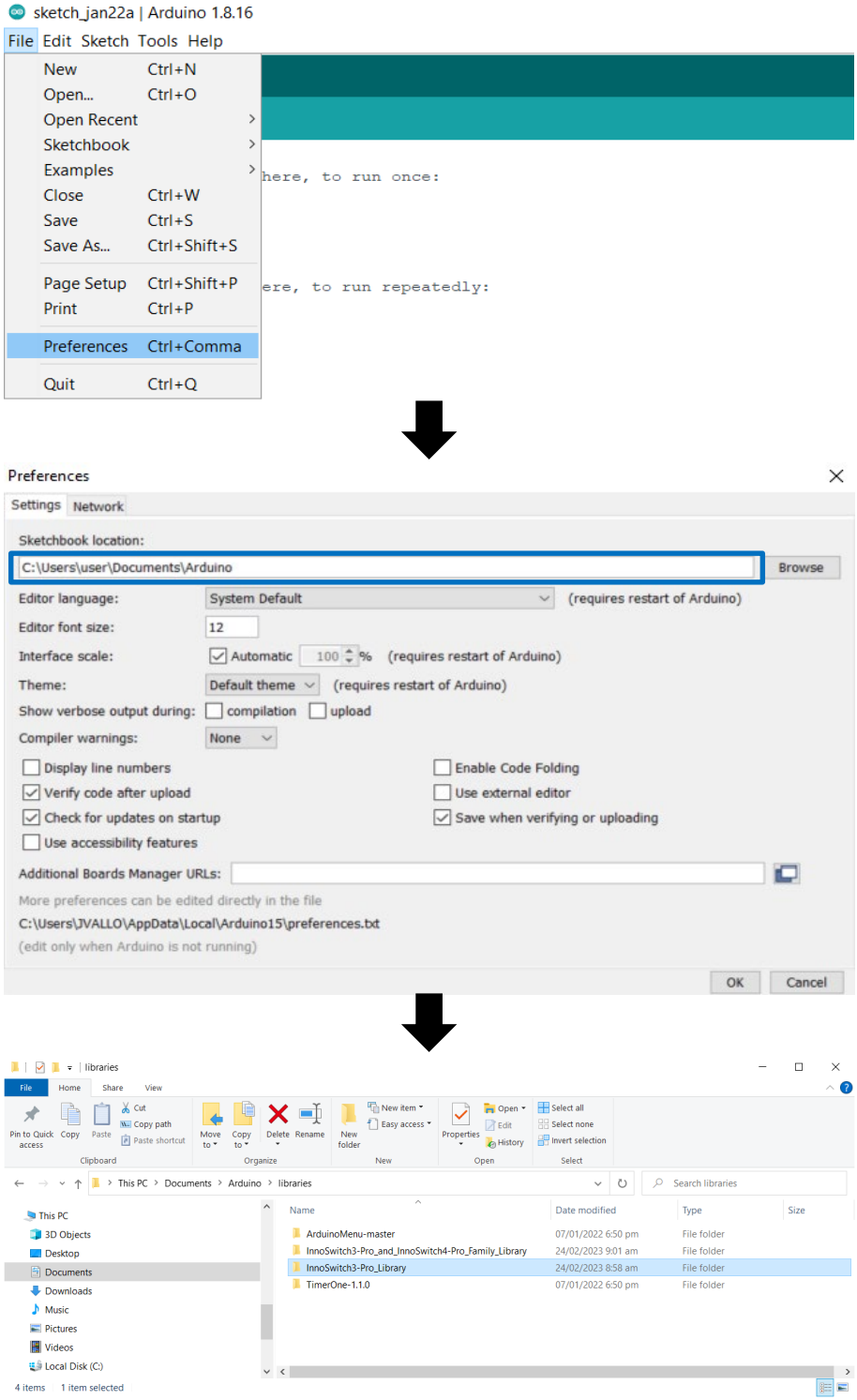
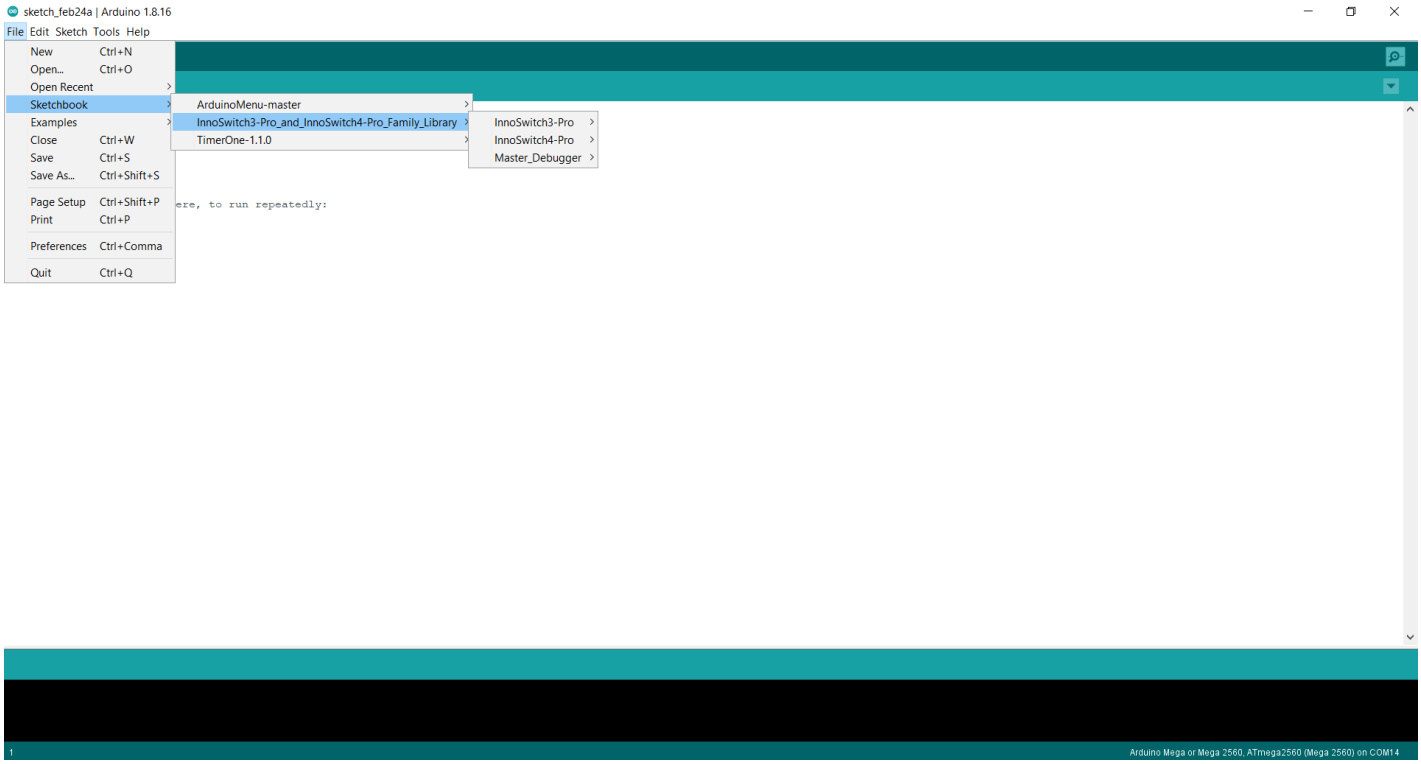


图7. 删除Arduino库

示例

安装该库后，用户可以使用*File（文件）> Sketchbook（固件程序集）> InnoSwitch3-Pro and InnoSwitch4-Pro Library（InnoSwitch3-Pro 和 InnoSwitch4-Pro库）*中的示例。这些固件程序提供了基本代码，可以作为框架，指导完成控制InnoSwitch3-Pro和InnoSwitch4-Pro器件的更复杂的实现。请注意，InnoSwitch3-Pro固件程序不适用于InnoSwitch4-Pro器件，反之亦然。



InnoSwitch3-Pro固件程序

- Inno3Pro_APDOs.ino
- Inno3Pro_Basic.ino
- Inno3Pro_Basic_Volts_Amps_OV_UV.ino
- Inno3Pro_PD_Hard_Reset.ino
- Inno3Pro_PDOs.ino
- Inno3Pro_Plotter.ino
- Inno3Pro_Ramp.ino
- Inno3Pro_Random_Volt_Time.ino
- Inno3Pro_Serial.ino
- Inno3Pro_SineWave.ino

InnoSwitch4-Pro固件程序

- Inno4Pro_APDOs.ino
- Inno4Pro_Basic.ino
- Inno4Pro_Basic_Volts_Amps_OV_UV.ino
- Inno4Pro_PD_Hard_Reset.ino
- Inno4Pro_PDOs.ino
- Inno4Pro_Plotter.ino
- Inno4Pro_Ramp.ino
- Inno4Pro_Random_Volt_Time.ino
- Inno4Pro_Serial.ino
- Inno4Pro_SineWave.ino

InnoSwitch3-Pro示例

示例1 - Inno3Pro_Basic.ino

Inno3Pro_Basic.ino固件程序实现了运行InnoSwich3-Pro IC的基本命令。使用此代码时，适配器将输出5V 3.1A。

此固件程序中有五个命令需要注意：

- Inno3Pro_Initialization(); - 用于初始化InnoSwitch3-Pro器件的函数
- Inno3Pro_Write_VI(); - 用于设置输出电压和电流的函数
- Inno3Pro_Write_Volt_Peak(); - 设置拐点电压(V_{KP})
- Inno3Pro_Vbus_Switch_Control(); - 控制母线开关

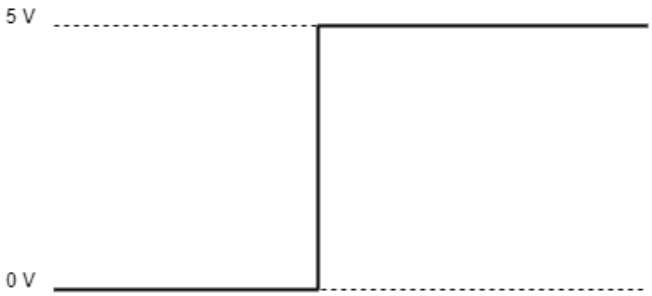


图8. Inno3Pro_Basic.ino输出电压波形

示例2 - Inno3Pro_PDOs.ino

此固件程序在多个输出电压之间循环。这是在模拟60W适配器的USBPD标准PDO。

使用的主要函数：

- Inno3Pro_Initialization(); - 初始化InnoSwitch3-Pro
- Inno3Pro_Write_Cable_Drop_Comp(); - 设置输出线压降补偿（以mV为单位）
- Inno3Pro_Write_Volt_Peak() - 设置拐点电压(V_{KP})
- Inno3Pro_Vbus_Switch_Control() – 控制母线开关
- clock_HasTimeElapsedMs(); - 当经过一定时间（以毫秒为单位）时返回1。
- Clock_GetTimeStampMs(); - 返回当前时间（以毫秒为单位）
- Inno4Pro_PD_Write_VI(); - 设置输出电压和电流

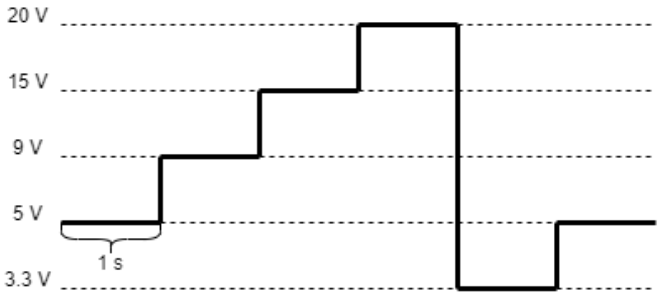


图9. Inno3Pro_PDOs.ino输出电压波形

```
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <InnoProBase.h>
#include "Inno3Pro.h"
#include "Inno3ProConfig.h"

//Step  2  :  Create the class instance
Inno3Pro_Application Inno3ProApp;

//Step 3 :  Write Initial Commands to Inno Pro
void setup()
{
   Inno3ProApp.Inno3Pro_Initialization();
}

//Step 4 :  Call the Functions on the Main Loop
void loop()
{
            //Control Functions Set-Up
            // 5V  Voltage SetPoint
            // 3.1A Constant Current
            Inno3ProApp.Inno3Pro_PD_Write_VI(5,3.1);

            // 300mV    , Cable Drop Compensation
            Inno3ProApp.Inno3Pro_Write_Cable_Drop_Comp(300);

            // 7V       , Constant Output Power Knee Voltage
            Inno3ProApp.Inno3Pro_Write_Volt_Peak(7);

            // ON       , Vbus Enable
            Inno3ProApp.Inno3Pro_Vbus_Switch_Control(1);
}
```

图10. Inno3Pro_Basic.ino代码

```
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <InnoProBase.h>
#include "Inno3Pro.h"
#include "Inno3ProConfig.h"

//Step  2  :  Create the class instance
InnoProBase_Rtc Inno3ProClk;
Inno3Pro_Application Inno3ProApp;

//Step 3 :  Write Initial Commands to Inno Pro
void setup()
{
            Inno3ProApp.Inno3Pro_Initialization();
            Inno3ProApp.Inno3Pro_Write_Cable_Drop_Comp(300);     // CDC  = 300mV
            Inno3ProApp.Inno3Pro_Write_Volt_Peak(24);            // VKP  = 24V
            Inno3ProApp.Inno3Pro_Vbus_Switch_Control(1);         // VBEN = ON
}

//Step 4 :  Call the Functions on the Main Loop
void loop()
{

   // Main Loop Variables
   static uint16_t u16_Main_State = 0;                //Initialize Main State
   static uint16_t u16_Request_Timer = 0;               //Initialize Request Timer

   //Timer Routine For Automatic Activation of  Requests
   if(Inno3ProClk.clock_HasTimeElapsedMs(u16_Request_Timer,1000))    //Delay Time
   {

     u16_Main_State++;                        //Change State
     u16_Request_Timer = Inno3ProClk.clock_GetTimeStampMs();     //Reset Timer
   }

   // Main Loop States
   switch(u16_Main_State)
   {
         case 0:
         u16_Main_State = 1;
         break;

     case 1:   //Activate 5V Configuration
          // CV = 5V and CC = 3.1A
         Inno3ProApp.Inno3Pro_PD_Write_VI(5,3.1);
         break;

     case 2:     //Activate 9V Configuration
         Inno3ProApp.Inno3Pro_PD_Write_VI(9,3.1);
         break;

     case 3:   //Activate 15V Configuration
         Inno3ProApp.Inno3Pro_PD_Write_VI(15,3.1);
         break;

     case 4:     //Activate 20V Configuration
         Inno3ProApp.Inno3Pro_PD_Write_VI(20,3.1);
         break;

     case 5:   //Activate 3.3V Configuration
         Inno3ProApp.Inno3Pro_PD_Write_VI(3.3,3.1);
         break;

     default:
         u16_Main_State = 1;
         break;
   }

 }
```

图11. Inno3-Pro_PDOs.ino代码

InnoSwitch4-Pro示例

示例1 – Inno4Pro_Basic.ino

Inno4Pro_Basic.ino固件程序实现了运行InnoSwich4-Pro IC的基本命令。使用此代码时，适配器将输出5V 3.1A。

此固件程序中有五个命令需要注意：

- Inno4Pro_Initialization(); - 用于初始化InnoSwitch4-Pro器件的函数
- Inno4Pro_Write_VI(); - 用于设置输出电压和电流的函数
- Inno4Pro_Write_Volt_Peak(); - 设置拐点电压(V_{KP})
- Inno4Pro_Vbus_Switch_Control(); - 控制母线开关

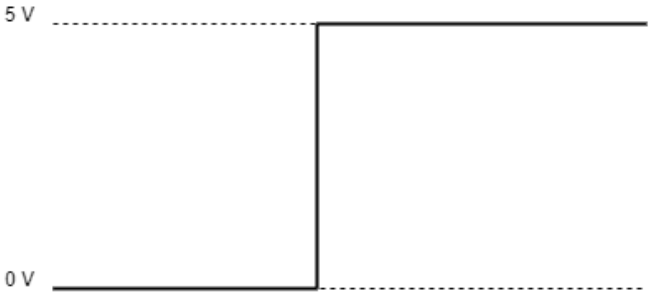


图12. Inno4Pro_Basic.ino输出电压波形

示例2 – Inno4Pro_PDOs.ino

此固件程序在多个输出电压之间循环。这是在模拟60W适配器的USBPD标准PDO。

使用的主要函数：

- Inno4Pro_Initialization(); - 初始化InnoSwitch4-Pro
- Inno4Pro_Write_Cable_Drop_Comp(); - 设置输出线压降补偿（以mV为单位）
- Inno4Pro_Write_Volt_Peak() - 设置拐点电压(V_{KP})
- Inno4Pro_Vbus_Switch_Control() – 控制母线开关
- clock_HasTimeElapsedMs(); - 当经过一定时间（以毫秒为单位）时返回1。
- Clock_GetTimeStampMs(); - 返回当前时间（以毫秒为单位）
- Inno4Pro_PD_Write_VI(); - 设置输出电压和电流

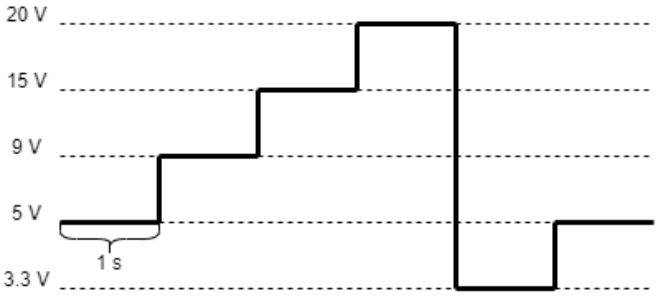


图13. Inno4Pro_PDOs.ino输出电压波形

```
//Step  1  :  Add the Header Files
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <InnoProBase.h>
#include "Inno4Pro.h"
#include "Inno4ProConfig.h"

//Step  2  :  Create the class instance
Inno4Pro_Application Inno4ProApp;

//Step 3 :  Write Initial Commands to InnoPro
void setup()
{
            Inno4ProApp.Inno4Pro_Initialization();
}

//Step 4 :  Call the Functions on the Main Loop
void loop()
{
    //Control Functions Set-Up
    // 5V  Voltage SetPoint
    //3.1A Constant Current
    Inno4ProApp.Inno4Pro_PD_Write_VI(5,3.1);

    // 300mV    , Cable Drop Compensation
    Inno4ProApp.Inno4Pro_Write_Cable_Drop_Comp(300);

    // 24V      , Constant Output Power Knee Voltage
    Inno4ProApp.Inno4Pro_Write_Volt_Peak(24);

    // ON       , Vbus Enable
    Inno4ProApp.Inno4Pro_Vbus_Switch_Control(1);
}
```

图14. 示例1 - Inno4Pro_Basic.ino代码

```
//Step  1  :  Add the Header Files
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <InnoProBase.h>
#include "Inno4Pro.h"
#include "Inno4ProConfig.h"

//Step  2  :  Create the class instance
InnoProBase_Rtc Inno4ProClk;
Inno4Pro_Application Inno4ProApp;

//Step 3 :  Write Initial Commands to InnoPro
void setup()
{
        Inno4ProApp.Inno4Pro_Initialization();
        Inno4ProApp.Inno4Pro_Write_Cable_Drop_Comp(300);
        Inno4ProApp.Inno4Pro_Write_Volt_Peak(24);
        Inno4ProApp.Inno4Pro_Vbus_Switch_Control(1);
}

//Step 4 :  Call the Functions on the Main Loop
void loop()
{

   // Main Loop Variables
   static uint16_t u16_Main_State = 0;          //Initialize Main State
   static uint16_t u16_Request_Timer = 0; //Initialize Request Timer

   //Timer Routine For Automatic Activation of Requests
   if(Inno4ProClk.clock_HasTimeElapsedMs(u16_Request_Timer,1000))
   {
     u16_Main_State++;                  //Change State

     u16_Request_Timer = Inno4ProClk.clock_GetTimeStampMs ();
   }

   // Main Loop States
   switch(u16_Main_State)
   {

     case 0:
          u16_Main_State = 1;
          break;

     case 1:   //Activate 5V Configuration
          Inno4ProApp.Inno4Pro_PD_Write_VI(5,3.1);
          break;

     case 2:    //Activate 9V Configuration
          Inno4ProApp.Inno4Pro_PD_Write_VI(9,3.1);
          break;

     case 3:   //Activate 15V Configuration
          Inno4ProApp.Inno4Pro_PD_Write_VI(15,3.1);
          break;

     case 4:    //Activate 20V Configuration
          Inno4ProApp.Inno4Pro_PD_Write_VI(20,3.1);
          break;

     case 5:   //Activate 3.3V Configuration
          Inno4ProApp.Inno4Pro_PD_Write_VI(3.3,3.1);
          break;

     default:
          u16_Main_State = 1;
          break;
   }

}
```

图15. 示例2 - Inno4Pro_PDOs.ino代码

源文件

Name	Date modified	Type	Size
Drv_i2c	16/12/2021 2:34 pm	CPP File	4 KB
Drv_i2c	16/12/2021 2:34 pm	H File	4 KB
Drv_Rtc	16/12/2021 2:34 pm	CPP File	3 KB
Drv_Rtc	16/12/2021 2:34 pm	H File	4 KB
Inno3Pro	22/12/2021 4:11 pm	CPP File	45 KB
Inno3Pro	22/12/2021 4:11 pm	H File	67 KB
Inno3ProConfig	23/12/2021 9:19 am	H File	26 KB
Inno4Pro	06/01/2022 11:09 am	CPP File	50 KB
Inno4Pro	22/12/2021 4:11 pm	H File	69 KB
Inno4ProConfig	23/12/2021 9:19 am	H File	28 KB
InnoProBase	16/12/2021 2:34 pm	CPP File	10 KB
InnoProBase	16/12/2021 2:34 pm	H File	14 KB
LcdKeypad	16/12/2021 2:34 pm	CPP File	4 KB
LcdKeypad	16/12/2021 2:34 pm	H File	3 KB

API – 处理命令序列、时序、寄存器设置、阈值计算、奇偶校验实现、遥测等。

InnoSwitch3-Pro和InnoSwitch4-Pro使用的代码核心

- *InnoProBase.h*
- *InnoProBase.cpp*

仅限于InnoSwitch3-Pro的代码核心

- *Inno3Pro.h*
- *Inno3Pro.cpp*

仅限于InnoSwitch4-Pro的代码核心

- Inno4Pro.h
- Inno4Pro.cpp

用于控制LCD Keypad Arduino Shield的代码核心

- *LcdKeypad.h*
- *LcdKeypad.cpp*

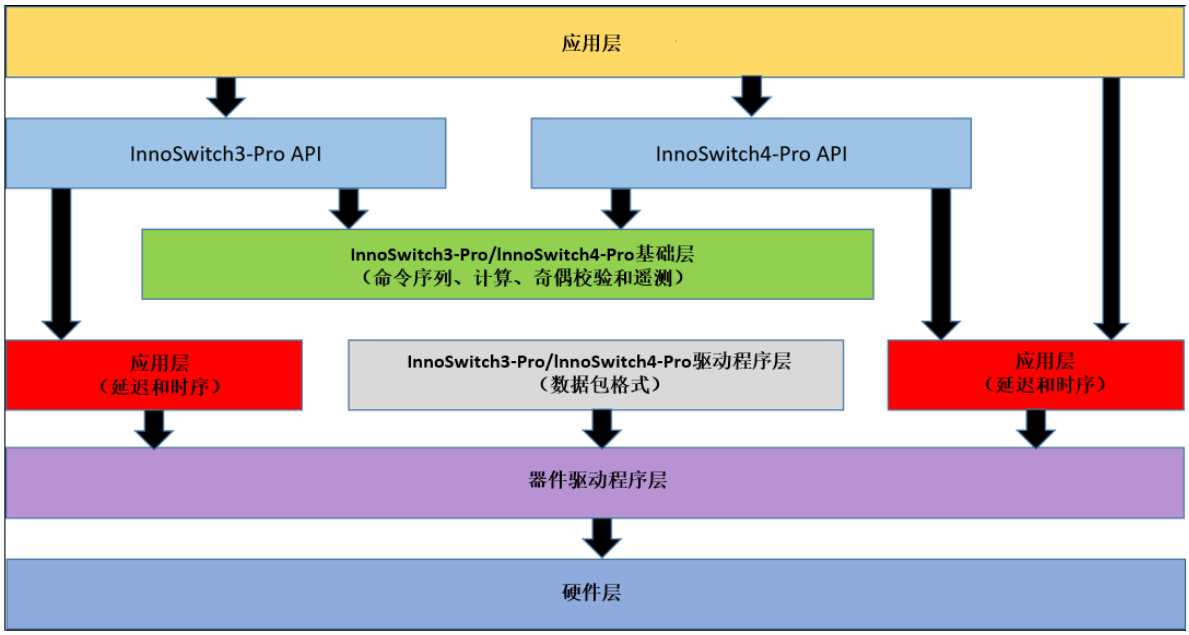
InnoSwitch驱动程序 – 根据InnoSwitch3-Pro/InnoSwitch4-Pro数据手册管理I²C数据包格式，用于写入和读取事务。Arduino Wire库用作较低级别的库。

- *Drv_I2C.h*
- *Drv_I2C.cpp*

时钟驱动程序 – 用于生成延迟和时序的模块

- *Drv_Rtc.h*
- *Drv_Rtc.cpp*

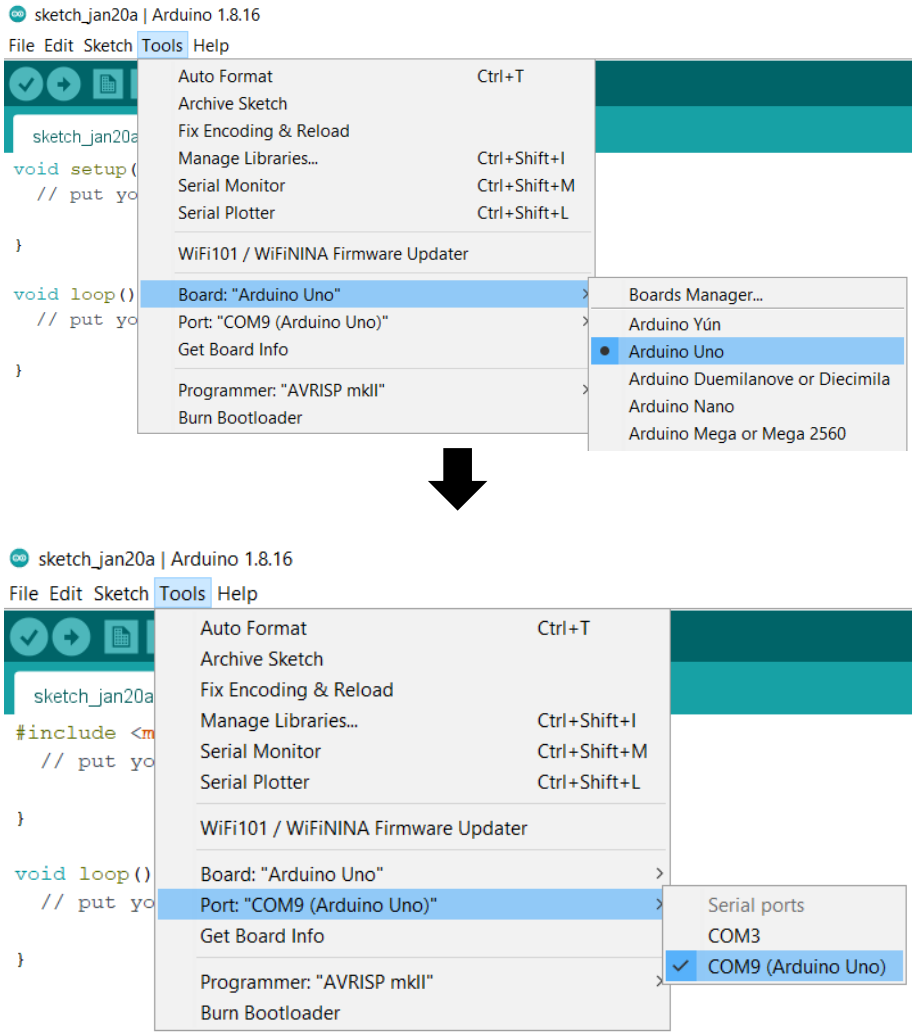
下图显示了各层之间的相互作用。应用层由实现InnoSwich3-Pro/InnoSwitch4-Pro和时钟驱动程序功能的InnoSwitch3-Pro/InnoSwitch4-Pro Arduino固件程序组成。



编译项目

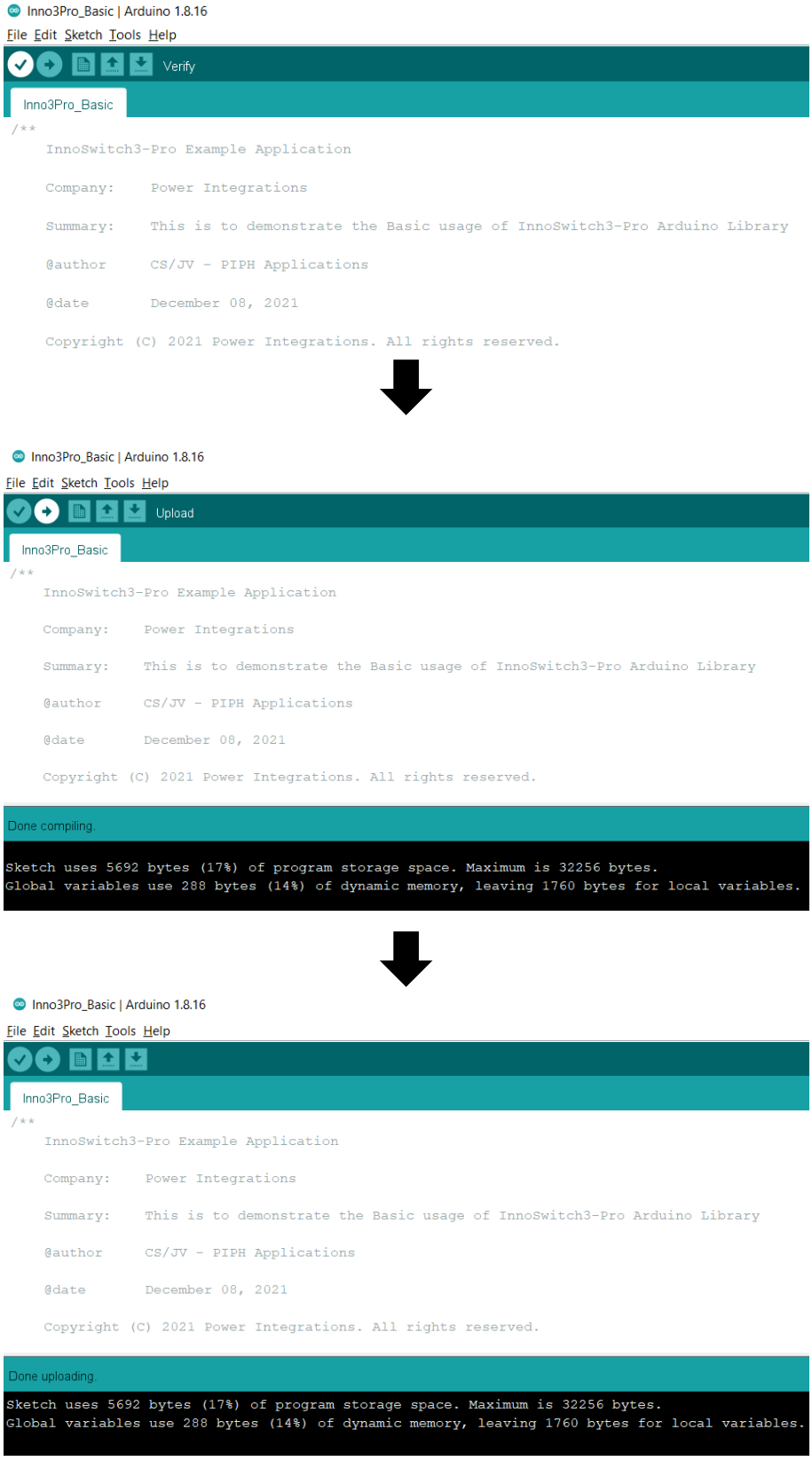
选择板

1. 转到*Tools（工具）菜单 > Board:（板：）*，选择所使用的Arduino器件。
2. 还可从*Tools（工具）菜单 > Port（端口）*中选择活动COM端口号，该端口决定Arduino板连接的USB端口。



上传固件程序

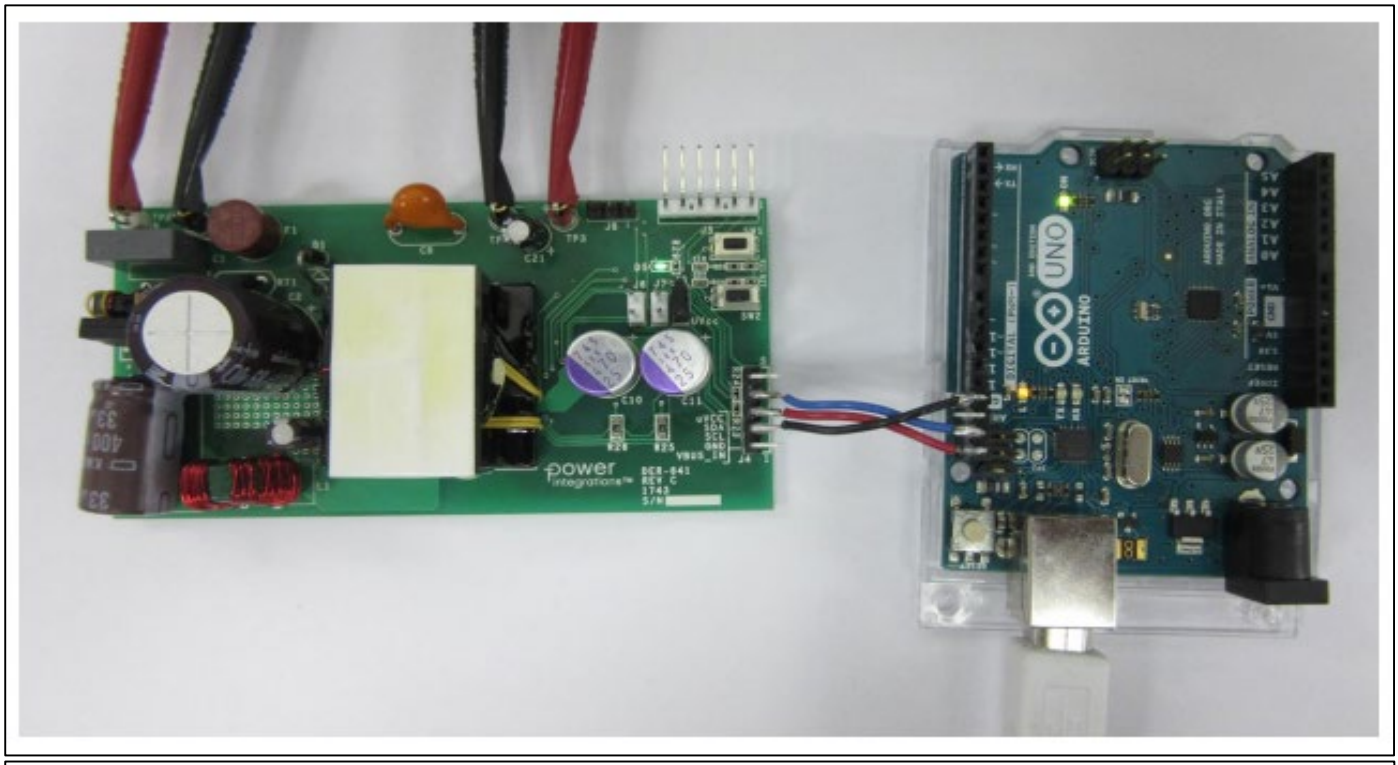
1. 从*File（文件）菜单 > Examples（示例）> InnoSwitch3-Pro and InnoSwitch4-Pro Library（InnoSwitch3-Pro和 InnoSwitch4-Pro库）*中选择要使用的固件程序。单击IDE左上角的“验证”图标。
2. 屏幕底部将显示一个指示器，显示“Done compiling”（已完成编译）以及固件程序使用的内存量。
3. 点击“验证”按钮旁边的箭头图标，将编译好的固件程序上传到Arduino板上。完成后，调试日志顶部的指示器将显示“Done Uploading”（上传完成）。

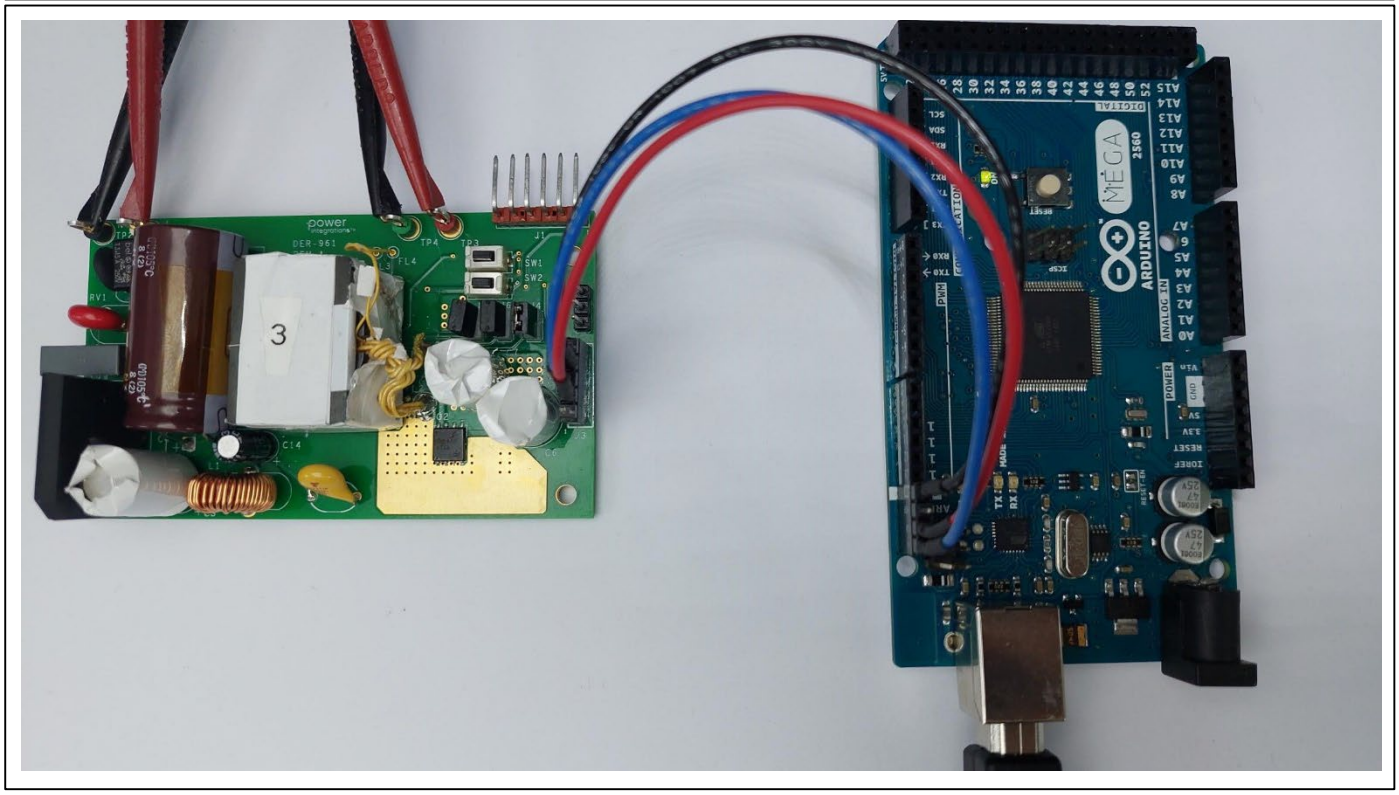


硬件设置

本节演示如何使用Arduino库控制InnoSwitch3-Pro器件。本例中使用的固件程序是Inno3_Basic.ino，它将InnoSwitch3-Pro初始化为输出5V和3.1A。移除RDK-641板上的跳线J6和J7。如下图所示，将Arduino板的I²C线路连接到RDK-641上。RDK-641上电后，固件程序就会上传到Arduino板。RDK-641的输出电压应为5V，电流限制为3.1A。

InnoSwitch4-Pro 设置使用 RDR-961 板而不是 RDK-641 。Inno4Pro_PDOs.ino固件程序使用多个Inno4Pro_PD_Write_VI()命令每秒更改一次输出电压，修改顺序为5V> 9V> 15V> 20V> 5V。

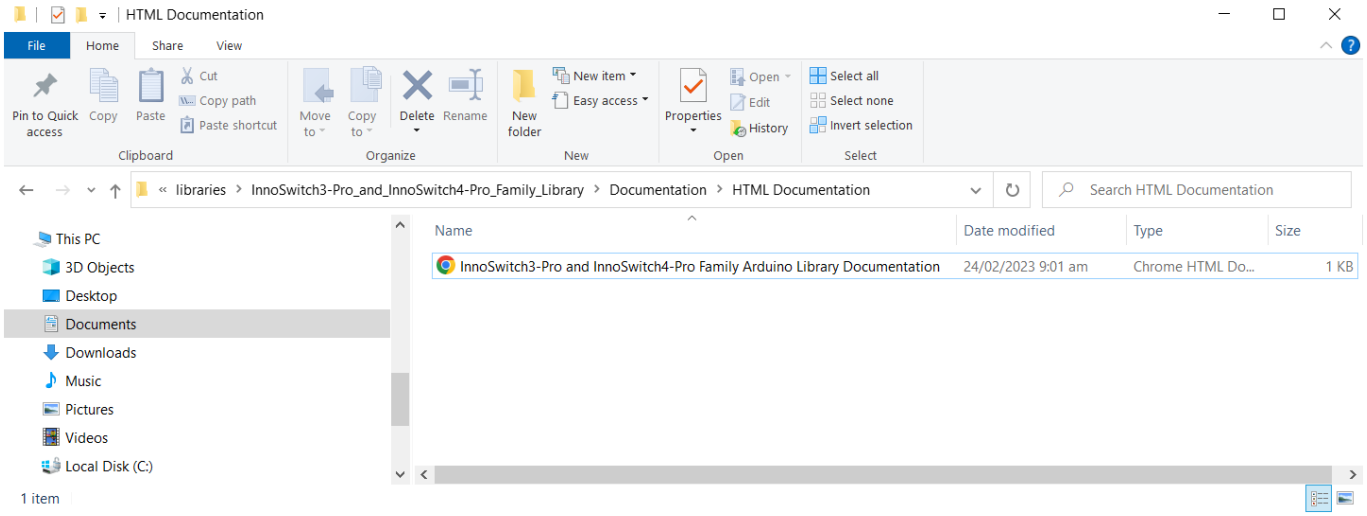




Doxygen文档

Documentation（文档）文件夹中有已编译的HTML (.chm)和HTML (.html)文件。这些文件包含InnoSwitch3-Pro和InnoSwitch4-Pro Arduino库的文档。其

中包含有关如何在API和核心驱动程序中使用每个函数的简要说明。文档中的示例简要介绍了代码的工作原理，以及如何使用每个固件程序。

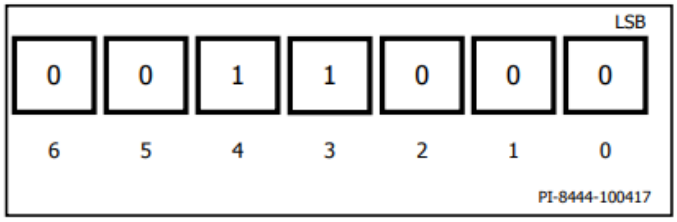


附录

寄存器定义

I²C从控地址

InnoSwitch3-Pro和InnoSwitch4-Pro的7位从控地址为0x18 (7’b001 1000)。

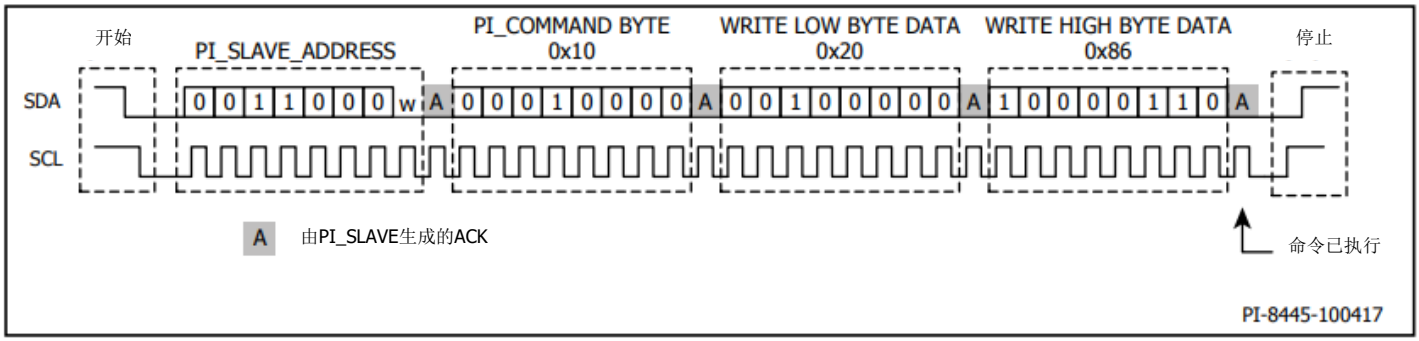


I²C协议格式

3字节写命令：

[PI_SLAVE_ADDRESS][W][A][PI_COMMAND][A][Byte][A] 或

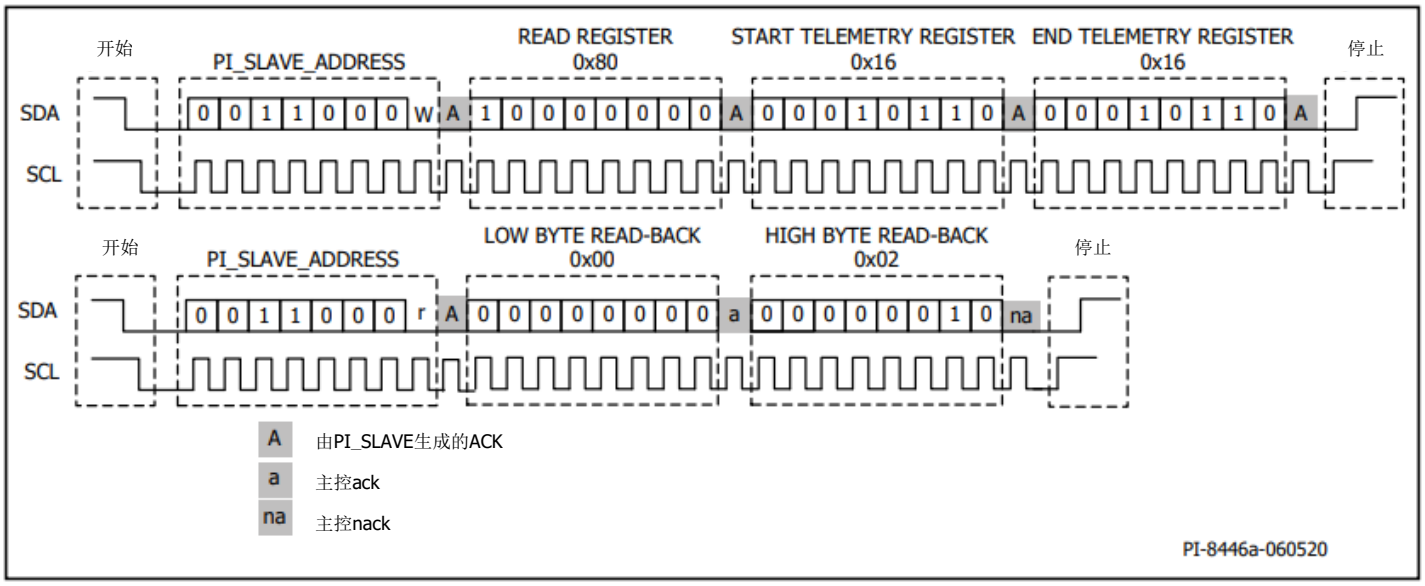
[PI_SLAVE_ADDRESS][W][A][PI_COMMAND][A][Low Byte][A][High Byte][A]



2字节读命令：

[PI_SLAVE_ADDRESS][W][A][PI_COMMAND][A][START_TELEMETRY_REGISTER_ADDRESS][A][END_TELEMETRY_REGISTER_ADDRESS][A]

[PI_SLAVE_ADDRESS][r][A][PI Slave responds Low Byte][a][PI Slave responds High Byte][na]



修订版本	注释	日期
A	初始版本。	01/20/23

有关最新产品信息，请访问：www.power.com

Power Integrations全球销售支持网络

全球总部
5245 Hellyer Avenue
San Jose, CA 95138, USA
Main: +1-408-414-9200
Customer Service:
Worldwide: +1-65-635-64480
Americas: +1-408-414-9621
e-mail: usasales@power.com

中国（上海）
徐汇区漕溪北路88号圣爱广场
1601-1603室
上海|中国，200030
电话：+86-21-6354-6323
电子邮箱：chinasales@power.com

中国（深圳）
南山区科技南八路二号豪威科技大厦17层
深圳|中国，518057
电话：+86-755-8672-8689
电子邮箱：chinasales@power.com

德国（AC-DC/LED业务销售）
Einsteinring 24
85609 Dornach/Aschheim
Germany
Tel: +49-89-5527-39100
e-mail: eurosales@power.com

德国（门极驱动器销售）
HellwegForum 1
59469 Ense
Germany
Tel: +49-2938-64-39990
e-mail: igbt-driver.sales@power.com

印度
#1, 14th Main Road
Vasanthanagar
Bangalore-560052 India
Phone: +91-80-4113-8020
e-mail: indiasales@power.com

意大利
Via Milanese 20, 3rd. Fl.
20099 Sesto San Giovanni (MI) Italy
Phone: +39-024-550-8701
e-mail: eurosales@power.com

日本
Yusen Shin-Yokohama 1-chome
Bldg. 1-7-9, Shin-Yokohama,
Kohoku-ku Yokohama-shi,
Kanagawa 222-0033 Japan
Phone: +81-45-471-1021
e-mail: japansales@power.com

韩国
RM 602, 6FL
Korea City Air Terminal B/D, 159-6 Samsung-Dong, Kangnam-Gu,
Seoul, 135-728, Korea
Phone: +82-2-2016-6610
e-mail: koreasales@power.com

新加坡
51 Newton Road
#19-01/05 Goldhill Plaza
Singapore, 308900
Phone: +65-6358-2160
e-mail: singaporesales@power.com

台湾地区
5F, No. 318, Nei Hu Rd., Sec. 1
Nei Hu Dist.
Taipei 11493, Taiwan R.O.C.
Phone: +886-2-2659-4570
e-mail: taiwansales@power.com

英国
Building 5, Suite 21
The Westbrook
Centre Milton Road
Cambridge
CB4 1YG
Phone: +44 (0) 7823-557484
e-mail: eurosales@power.com