

---

<b>Title</b>	<b><i>Getting Started with InnoSwitch™ 3-Pro Code Library using Arduino</i></b>
<b>Author</b>	Applications Engineering Department
<b>Document Number</b>	Application Note 77 (AN-77)
<b>Date</b>	September 6, 2018
<b>Revision</b>	1.0

## **Summary**

InnoSwitch3-Pro is a digitally controllable CV/CC QR Flyback Switcher IC with integrated High Voltage MOSFET, Synchronous Rectification and FluxLink Feedback.

RDK-641 is a reference design board rated for 40W output power and is programmable from 3V to 20V output voltage. This reference design features an on board PIC16F18325 microcontroller and uses the InnoSwitch3-Pro integrated power supply IC.

This application note describes use of Arduino code libraries provided by Power Integrations to develop control logic and firmware for customizing RDK-641.

Information presented in this application note was used to develop firmware for Arduino UNO.

## **PATENT INFORMATION**

The products and applications illustrated herein (including transformer construction and circuits external to the products) may be covered by one or more U.S. and foreign patents, or potentially by pending U.S. and foreign patent applications assigned to Power Integrations. A complete list of Power Integrations' patents may be found at [www.powerint.com](http://www.powerint.com). Power Integrations grants its customers a license under certain patent rights as set forth at <https://www.power.com/company/intellectual-property-licensing/>.

## Table of Contents

1	Introduction .....	3
2	System Requirements .....	4
3	Hardware Overview.....	5
3.1	Headers and Jumpers Settings.....	5
4	InnoSwitch3-Pro Arduino Code Library .....	7
4.1	Library Installation.....	7
4.2	Library Installation Complete.....	8
4.3	Library Examples .....	9
5	Folder Contents.....	11
5.1	File Description .....	12
6	Application Example.....	14
6.1	Step-By-Step Procedure .....	14
6.1.1	Header Files Inclusion .....	14
6.1.2	Class Instance Creation.....	14
6.1.3	InnoSwitch3-Pro Initialization.....	14
6.1.4	Basic Control Functions .....	15
6.1.5	Basic Code Examples.....	16
7	Building the Project .....	18
7.1	Arduino board selection .....	18
7.2	Select the Active Com Port .....	18
7.3	Verify / Compile.....	19
7.4	Upload.....	19
8	Demonstration of Operation .....	20
8.1	Running the Program.....	20
8.2	Constant voltage operation .....	21
8.3	Constant current operation.....	21
9	Doxygen Documentation .....	22
9.1	Opening html file .....	22
9.2	Viewing the API Functions .....	23
9.3	Functions summary.....	24
9.4	Functions definition .....	25
9.5	Examples .....	26
10	Revision History.....	27



# 1 Introduction

This application note describes the structure and the application interface of the InnoSwitch3-Pro Arduino Code Library as well as using it on a demo application. The code was designed to be highly portable to other microcontroller platforms, and was written in C++ language to be compatible with the Arduino library standards. This demo application runs on Reference Design RDK-641 (Figure 1) and Arduino UNO (Figure 3).

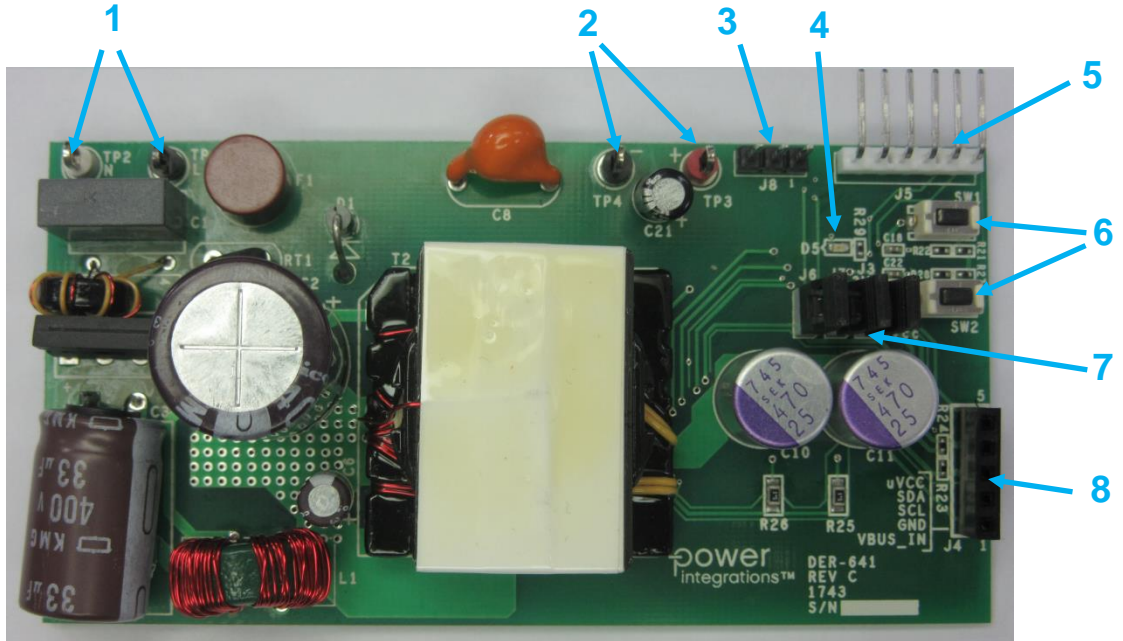


Figure 1 – RDK-641 Board Top

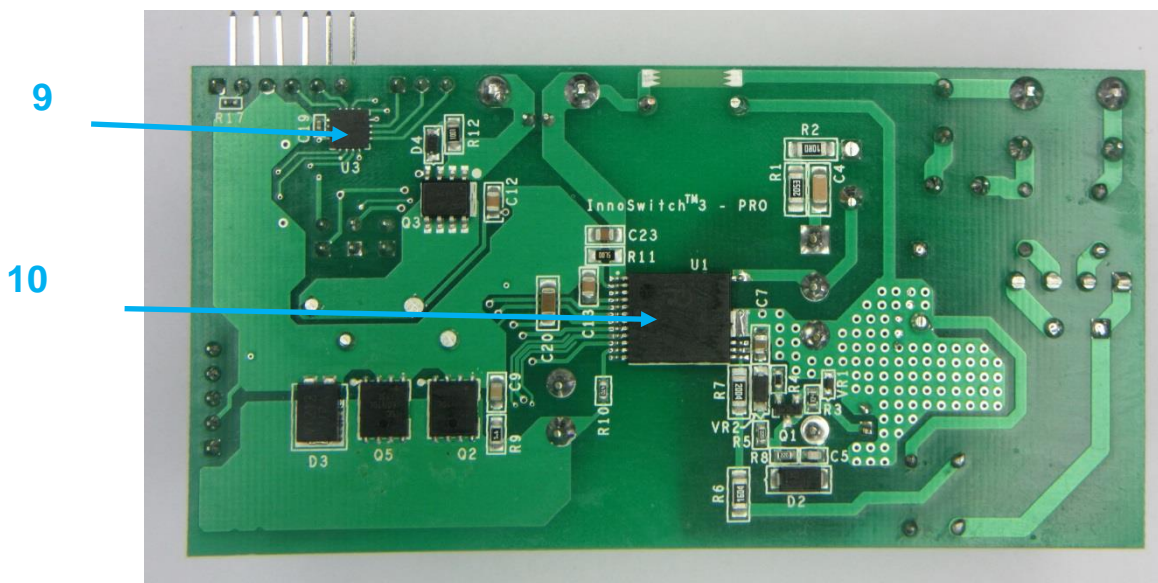
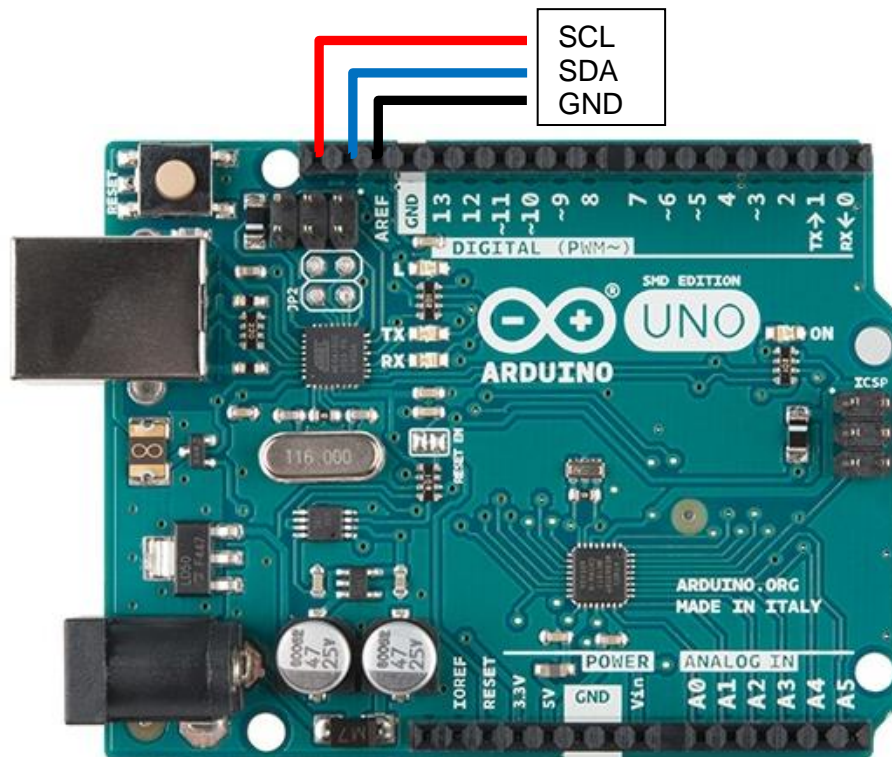


Figure 2 – RDK-641 Board Bottom

The RDK-641 board's key features are indicated on the table below

Number	Description	Label
1	AC Input Terminals	TP1, TP2
2	DC Output Terminals	TP3, TP4
3	MCU GPIO Headers	J8
4	Green LED Indicator	D5
5	Pickit3 Programming Header	J5
6	Push Buttons	SW1, SW2
7	uVCCand I <sup>2</sup> C Isolation Jumpers	J3, J6, J7
8	External Interface Header	J4
9	PIC16F18325 microcontroller	U3
10	InnoSwitch3-Pro IC	U1



**Figure 3 – Arduino Uno Rev 3**

## 2 System Requirements

The following are required to run the InnoSwitch3-Pro Arduino demo application

- Arduino Software version 1.8.2 or later
- Arduino UNO Rev3 SMD
- RDK – 641 Board rev C
- InnoSwitch3-Pro Arduino Library version 1.0.0

### 3 Hardware Overview

The Reference Design (RDK-641) hardware consists of an 8-bit Microchip microcontroller (PIC16F18325), interface headers and the user interface elements: two push buttons and a green LED.

The InnoSwitch3-Pro can be controlled using its on board microcontroller or by an external I<sup>2</sup>C Master through the interface header.

This Demo Application does not use the on board microcontroller but an Arduino Uno as an I<sup>2</sup>C Master and InnoSwitch3-Pro as slave device.

SDA and SCL lines pull-up resistors R24 and R23 respectively are available on the board. The output of the InnoSwitch3-Pro provides 3.6V pull up voltage from its  $\mu$ VCC output pin.

To further ease in development, the following documents are available and recommended as supplemental reference resources I<sup>2</sup>C

- [RDR-641 - 40 W Variable Output \(3 V to 8 V, 5 A; 8 V – 20 V Constant Power\) Supply Using InnoSwitch3-Pro and Microchip's PIC16F18325 Microcontroller](#)
- [AN-74 InnoSwitch3-Pro Programming Manual](#)

#### 3.1 Headers and Jumpers Settings

The table provides the description for each jumper available on the board.

Jumper	Description	Settings
J3	$\mu$ VCC and MCU Supply Jumper	If connected the $\mu$ VCC output pin of the InnoSwitch3-Pro will provide power to the on board microcontroller and provide pull up voltage to the I <sup>2</sup> C lines
J6 , J7	I <sup>2</sup> C Lines Isolation Jumper	The user can select whether or not the SDA and SCL lines from the MCU will be connected to the InnoSwitch3-Pro

The following headers are also available on the board.

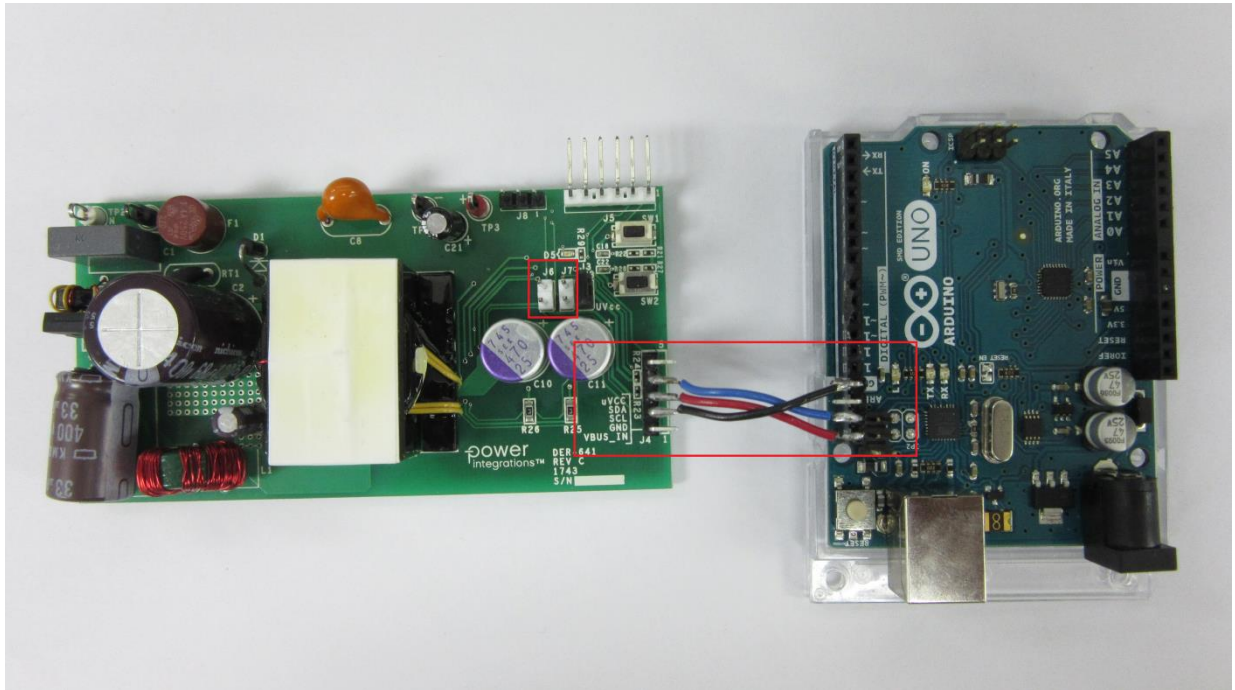
Header	Description	Settings
J4	InnoSwitch3-Pro I <sup>2</sup> C lines Header	When J6 and J7 are removed, an external I <sup>2</sup> C Master can be connected through these header
J5	PICKit3 Programming Header	For MCU Firmware Update using PICKit3 In-Circuit Debugger/Programmer
J8	MCU GPIO Header	This can be used as Debug Pins



By configuring the I<sup>2</sup>C lines isolation jumpers, the RDK-641 board can be controlled using Arduino UNO.

### Connection details:

- ▶ Remove Jumpers J6 and J7 and retain Jumper J3
- ▶ Connect J4 to Arduino Uno I<sup>2</sup>C lines (SDA, SCL and GND).
- ▶ Make sure to check the I<sup>2</sup>C labels of J4 and Arduino UNO board



- ▶ Wires on the Image above
  - Blue – SDA
  - Red – SCL
  - Black – GND

## 4 InnoSwitch3-Pro Arduino Code Library

To simplify the technicalities on controlling the InnoSwitch3-Pro, a simple code library is provided as a reference.

The library contains all the registers needed for controlling the device. These registers are organized as Command Registers and Telemetry registers. Command registers are sent to the device for performance control and Telemetry Registers are for reading back values.

Computation Macros are presented to aid in set point calculations. Register default values are also defined to simplify writing to the required registers at device initialization.

The InnoSwitch3-Pro Arduino code library is available from the Power integrations website.

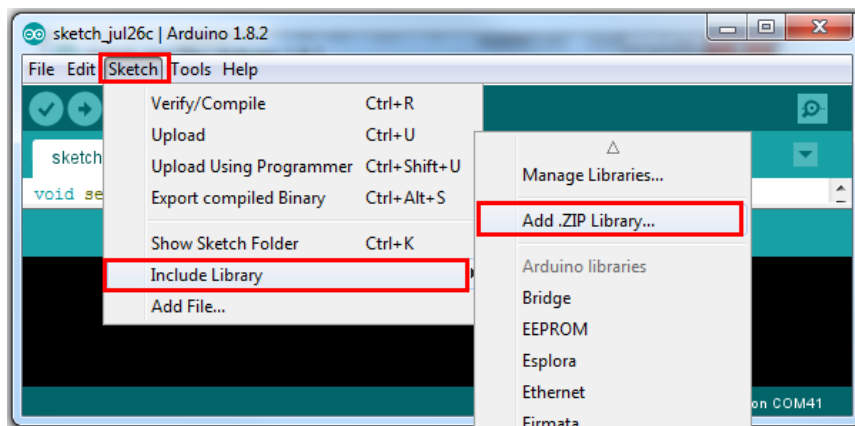
<https://ac-dc.power.com/design-support/articles/innoswitch3-pro-code-library-api-arduino/>

### 4.1 Library Installation

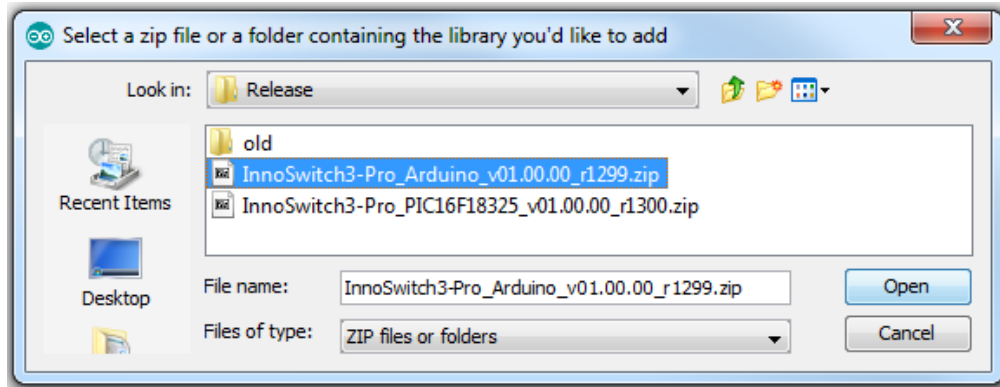
Full installation guide can be found on the link below

<https://www.arduino.cc/en/Guide/Libraries>

In the Arduino IDE, navigate to *Sketch > Include Library > Add .ZIP Library*. At the top of the drop down list, select the option to "Add .ZIP Library".

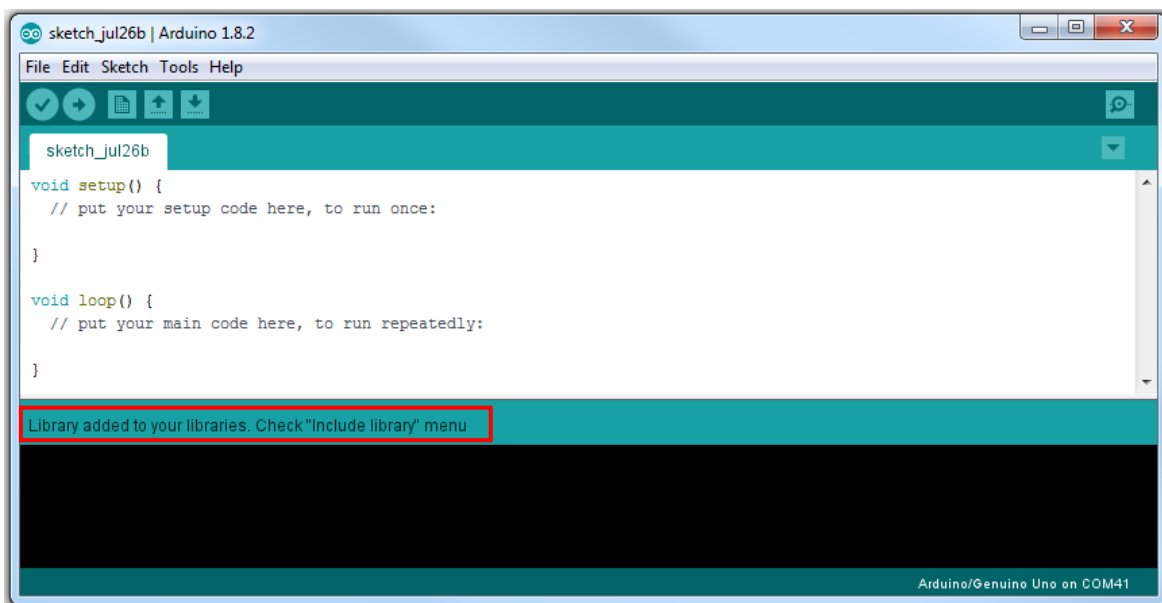


Navigate to the .zip file's location and open it.

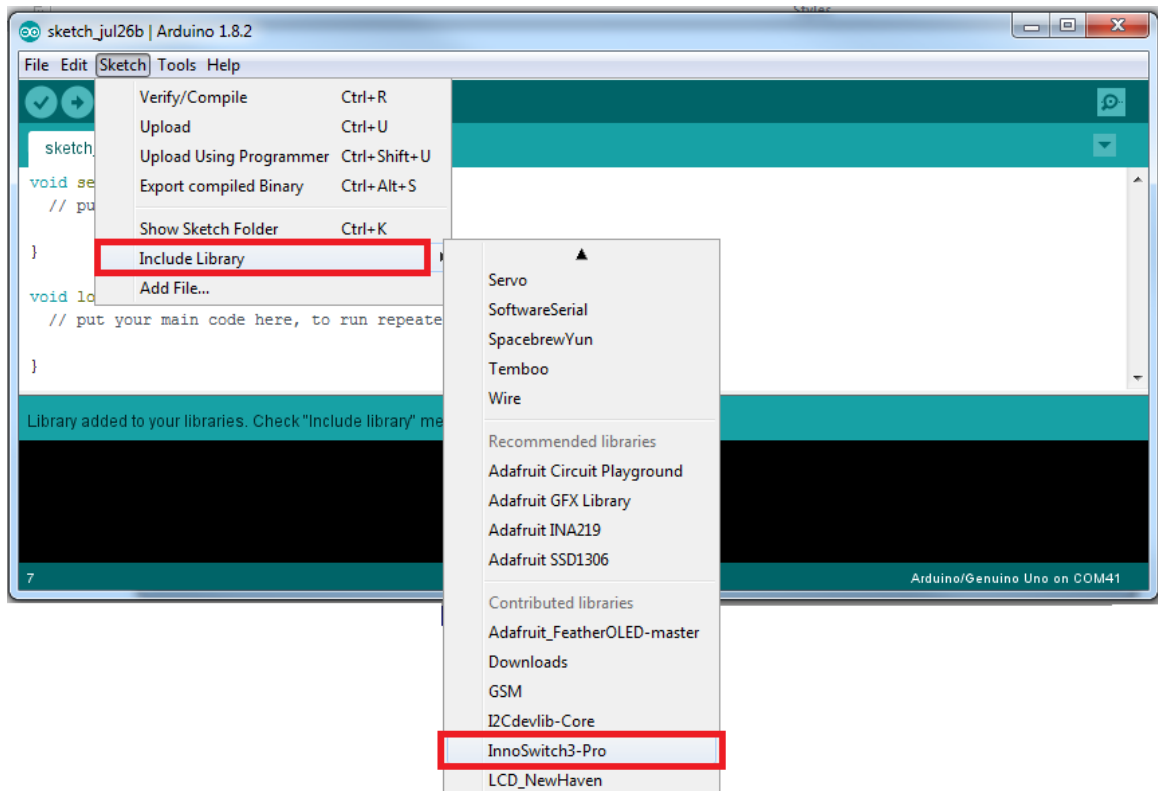


## 4.2 Library Installation Complete

Images below shows the InnoSwitch3-Pro library was added to the Arduino Library





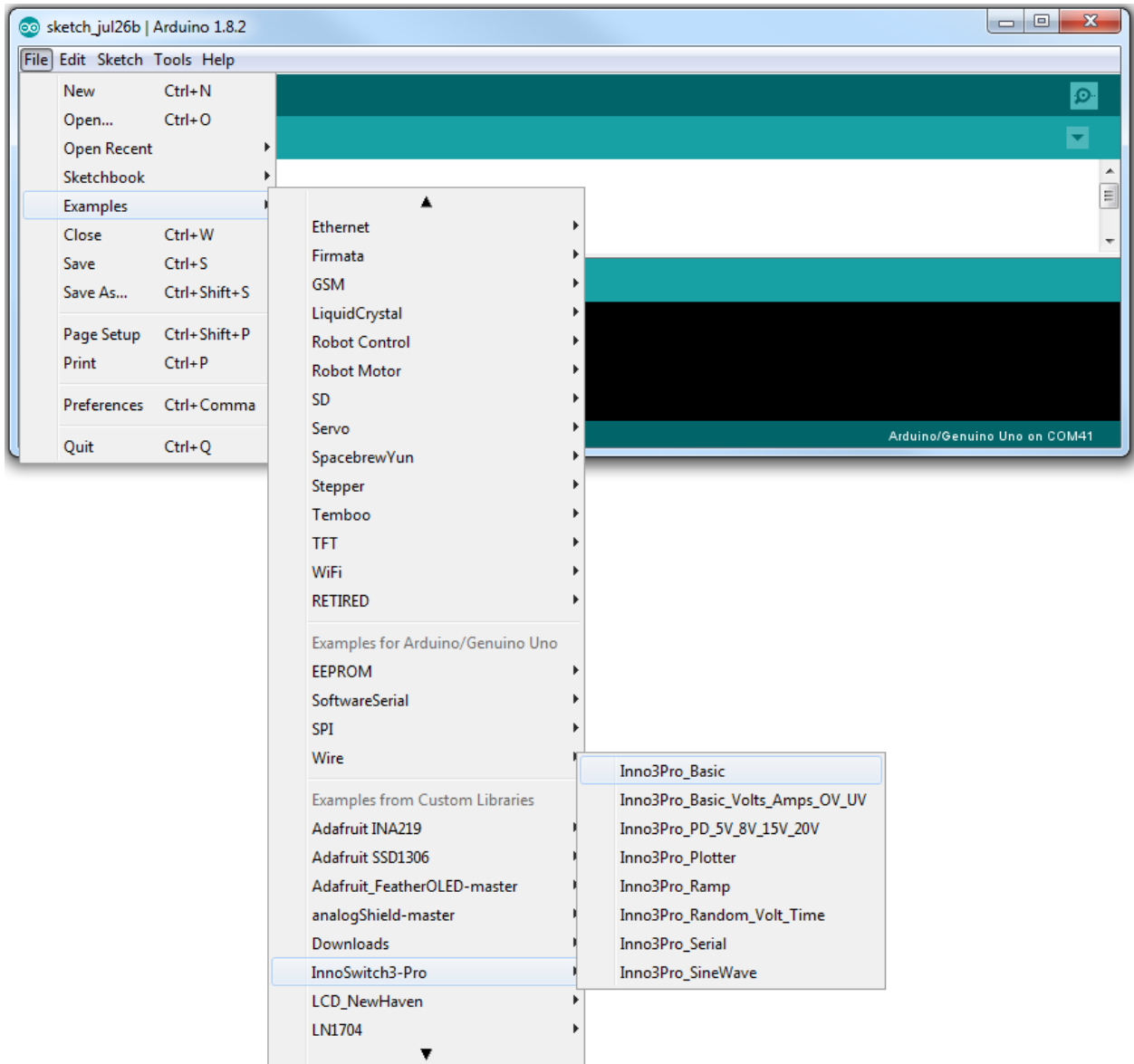


For windows users, the library can be viewed on this directory:

- ▶ C:\Users\username\Documents\Arduino\libraries\

### 4.3 Library Examples

The Library will be available to use in sketches, *File > Examples*



## 5 Folder Contents

The InnoSwitch3-Pro Arduino Code library consists of various folders and files as shown below

Documents library	
InnoSwitch3-Pro_Arduino_v01.00.00_r1299	
Name	Type
Documentation	File folder
examples	File folder
Config.h	H File
Drv_i2c.cpp	C++ Source File
Drv_i2c.h	H File
Drv_Rtc.cpp	C++ Source File
Drv_Rtc.h	H File
Inno3Pro.cpp	C++ Source File
Inno3Pro.h	H File
keywords.txt	Text Document
library.properties	PROPERTIES File
Main.dox	DOX File
README.md	MD File

Folder and files summary:

- ▶ **Examples** - Contains the InnoSwitch3-Pro Main Application Examples (\*.ino)
- ▶ **Documentation** - Contains the Doxygen HTML Documentation
- ▶ **\*.cpp and .h** - Library source and header files
- ▶ **Library.properties** - Arduino Library files format
- ▶ **Keywords.txt** - List of keywords for the library, provided syntax coloring
- ▶ **Main.dox** - Doxygen file
- ▶ **README.md** - Compatibility list

## 5.1 **File Description**

The Arduino Code library is layered and modular, implemented in, '[Clock Driver](#)', '[InnoSwitch3-Pro Driver](#)' and '[InnoSwitch3-Pro API](#)'. The Library architecture block diagram is schematically presented in Figure3

Below is a brief description of each layer:

### ■ **InnoSwitch3-Pro API**

- ▶ Simple Control Interface to control InnoSwitch3-Pro. This handles Command Sequences and Timings, Register Settings, Threshold Calculations, Parity Implementation, Telemetry

#### Related Files:

Inno3Pro.h - Contains the core of the library  
Inno3Pro.cpp

Config.h - Contains the Configuration Parameters of the Library

### ■ **InnoSwitch3-Pro Driver**

- ▶ Manages the I<sup>2</sup>C Packet format based on the InnoSwitch3-Pro Datasheet for Write and Read Transactions
- ▶ Built using Arduino Wire Library

#### Related Files:

Drv\_I2C.cpp  
Drv\_I2C.h

### ■ **Clock Driver**

- ▶ Module used for generating delays and timings involved for InnoSwitch3-Pro Control.
- ▶ Built using Arduino 'millis()' and 'micros()' functions

#### Related Files:

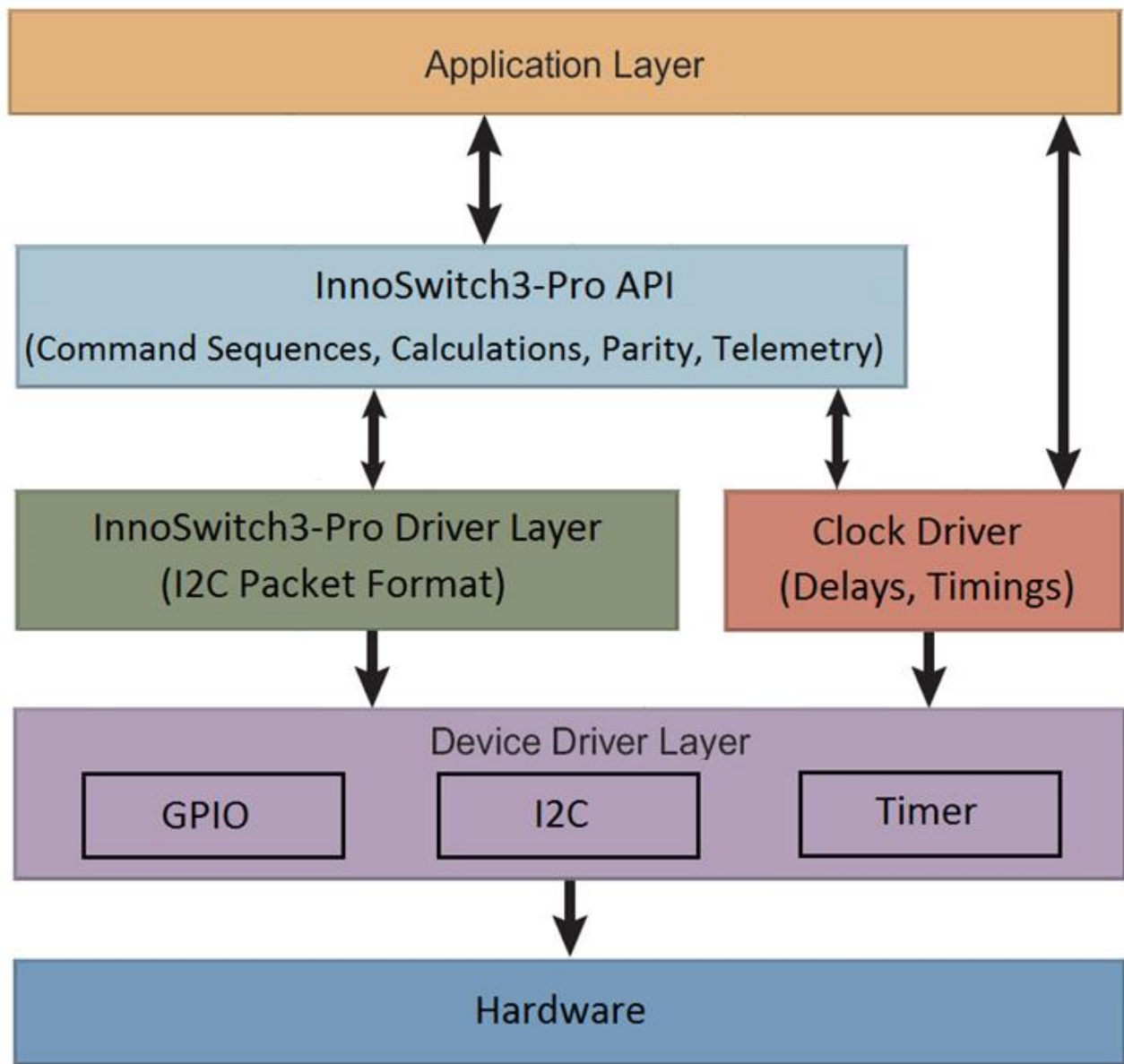
Drv\_Rtc.cpp  
Drv\_Rtc.h

### ■ **Application**

- ▶ Application Layer

#### Related Files:

Inno3Pro\_Basic.ino  
Inno3Pro\_Basic\_Volts\_Amps\_OV\_UV.ino



**Figure 3 – Firmware Architecture**

## 6 Application Example

This section describes the step-by-step procedures for setting up the Arduino sketch for InnoSwitch3-Pro.

### 6.1 *Step-By-Step Procedure*

#### 6.1.1 Header Files Inclusion

The library header files contain all of the function declarations and macro definitions. This must be included in the main page as shown.

```
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <Inno3Pro.h>
#include <Config.h>
```

#### 6.1.2 Class Instance Creation

Construct a Class instance to call the functions inside **Inno3Pro\_Application**. Constructing a Class instance of **Inno3Pro\_Rtc** is Optional.

```
Inno3Pro_Application    Inno3ProApp;
Inno3Pro_Rtc            Inno3ProClk;
```

#### 6.1.3 InnoSwitch3-Pro Initialization

Before continuous execution of the main code, the status of System Ready Signal is monitored to ensure the InnoSwitch3-Pro is ready to receive I<sup>2</sup>C commands. Afterwards initialization commands are sent to the device to configure the default settings. This initialization routine disables the watchdog timer and Fast VI Limit. UVL timer is also initialized to 64ms.

The 400 kHz clock frequency for I<sup>2</sup>C communication is set-up on initialization.

```
void setup()
{
    Inno3ProApp.Inno3Pro_Initialization();
}
```



#### 6.1.4 Basic Control Functions

##### **Updates the Output Voltage and Constant Current Setting**

- Follows a certain sequence of I<sup>2</sup>C commands in order to avoid inadvertent triggering of UV or OV faults
- Controls the VOUT pin strong bleeder when Decreasing the voltage from High to Low Setting
- Automatically updates the Over Voltage (OVA) and Under Voltage (UVA) settings
  - OVA is 124% of CV Setpoint
  - UVA is Fixed to 3V Setting

```
Inno3Pro.Inno3Pro_Write_VI( Volts, Amps )
```

##### **Updates the Output Voltage without Bleeder Control**

```
Inno3Pro.Inno3Pro_Write_Volts( Volts )
```

##### **Sets the Constant Current Setting**

```
Inno3Pro.Inno3Pro_Write_Amps( Amps )
```

##### **Sets the Over Voltage Setting**

```
Inno3Pro.Inno3Pro_Write_Over_Volts( Value )
```

##### **Sets the Under Voltage Setting**

```
Inno3Pro.Inno3Pro_Write_Under_Volts( Value )
```

##### **Sets the Cable Drop Compensation Value**

```
Inno3Pro.Inno3Pro_Write_Cable_Drop_Comp( Value )
```

##### **Sets the Constant Output Power Threshold**

```
Inno3Pro.Inno3Pro_Write_Volt_Peak( Value )
```

##### **Used for Turning On or Off the Bus Voltage Switch**

```
Inno3Pro.Inno3Pro_Vbus_Switch_Control( Value )
```

##### **Used for Turning On or Off the VOUT pin strong bleeder**

- The BLEEDER must not be enabled for extended period of time to prevent excessive power dissipation in the controller

```
Inno3Pro.Inno3Pro_Bleeder_Enable ( Value )
```

## 6.1.5 Basic Code Examples

### 6.1.5.1 Example 1 - Inno3Pro\_Basic.ino

#### **Demonstrates the basic usage of InnoSwitch3-Pro Arduino Library.**

- Initial commands are sent using the InnoSwitch3-Pro Initialization Routine.
- The Main Routine using write VI sets the output voltage to 5V and constant current current to 5.1A.
- Cable Drop Compensation is programmed to 300mV.
- Constant power is knee voltage is set to 7V and then Vbus Switch is turned ON

This code example is presented on "examples\Inno3Pro\_Basic\Inno3Pro\_Basic.c"  
Copy and paste these contents to your Arduino sketch.

```
//Step 1 : Add the Header Files
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <Inno3Pro.h>
#include <Config.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, 5.1A , Voltage and Constant Current
    Inno3ProApp.Inno3Pro_Write_VI(5 , 5.1);

    // 300mV , Cable Drop Compesation
    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);
}
```

## 6.1.5.2 Example 2 - Inno3Pro\_Basic\_Volts\_Amps\_OV\_UV.ino

**Demonstrates the basic usage of InnoSwitch3-Pro Arduino Library.**

- Initial commands are sent using the InnoSwitch3-Pro Initialization Routine.
- Output Over voltage is set to 6.2V and Output Under voltage is programmed to 3.6V
- The Main Routine sets the output voltage to 5V and constant current current to 5.1A.
- Cable Drop Compensation is programmed to 300mV.
- Constant power is knee voltage is set to 7V and then Vbus Switch is turned ON

This code example is presented on "examples\Inno3Pro\_Basic\_Volts\_Amps\_OV\_UV\Inno3Pro\_Basic\_Volts\_Amps\_OV\_UV.ino"

Copy and paste these contents to your Arduino sketch.

```
//Step 1 : Add the Header Files
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <Inno3Pro.h>
#include <Config.h>

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

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

  //Set Over Voltage Protection
  Inno3ProApp.Inno3Pro_Write_Over_Volts(6.2);

  //Set Under Voltage Protection
  Inno3ProApp.Inno3Pro_Write_Under_Volts(3.6);
}

//Step 4 : Call the Functions on the Main Loop
void loop()
{
  //Control Functions Set-Up

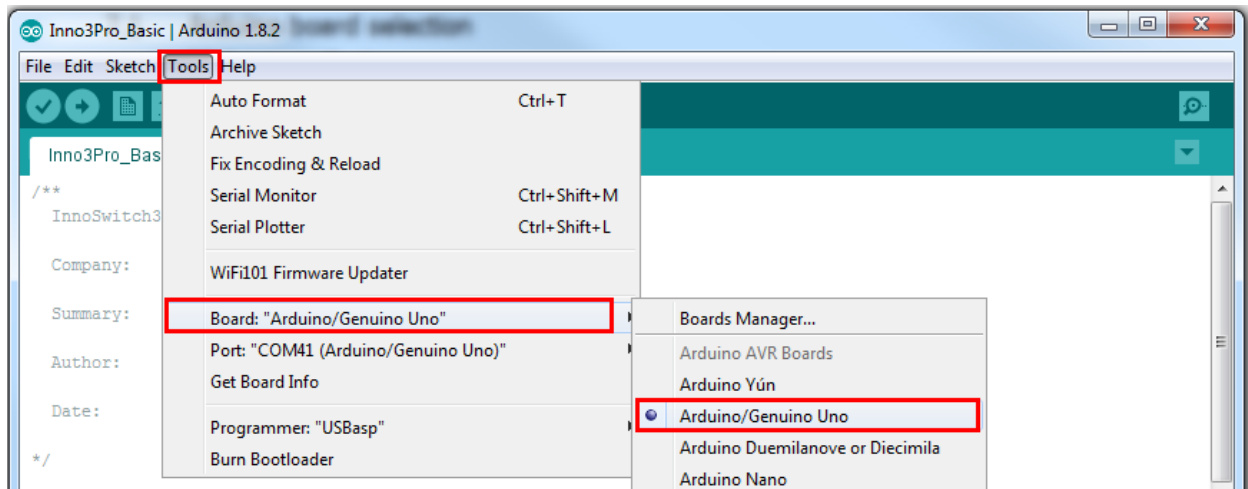
  // Main Loop Variable Initialization
  float fVolts = 5; //Initialize Voltage at 5V
  float fAmps = 5.1; //Initialize Constant Current at 5.1A
  float fCableDropComp = 300; //Initialize Cable Drop Compensation to 300mv
  float fVoltPeak = 7; //Initialize Knee Voltage at 7V
  float fVbusEn = 1; //Initialize Vbus Enable to at ON

  // Library Call in the Mainloop
  Inno3ProApp.Inno3Pro_Write_Amps(fAmps); //Set Constant Current
  Inno3ProApp.Inno3Pro_Write_Volts(fVolts); //Set Voltage
  Inno3ProApp.Inno3Pro_Write_Cable_Drop_Comp(fCableDropComp); //Set Cable Drop Compensation
  Inno3ProApp.Inno3Pro_Write_Volt_Peak(fVoltPeak); //Set Constant Output Power Knee Voltage
  Inno3ProApp.Inno3Pro_Vbus_Switch_Control(fVbusEn); //Set Vbus Enable
}
```

## 7 Building the Project

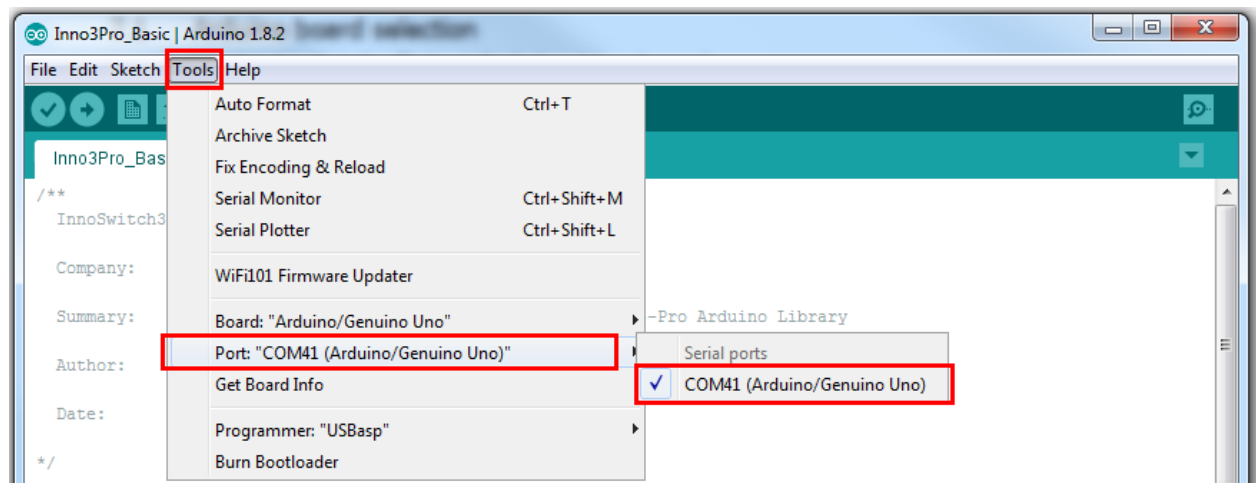
### 7.1 Arduino board selection

- ▶ Under tools menu, Select Arduino UNO board
- ▶ Make sure your Arduino Uno is already connected to your computer through the usb port



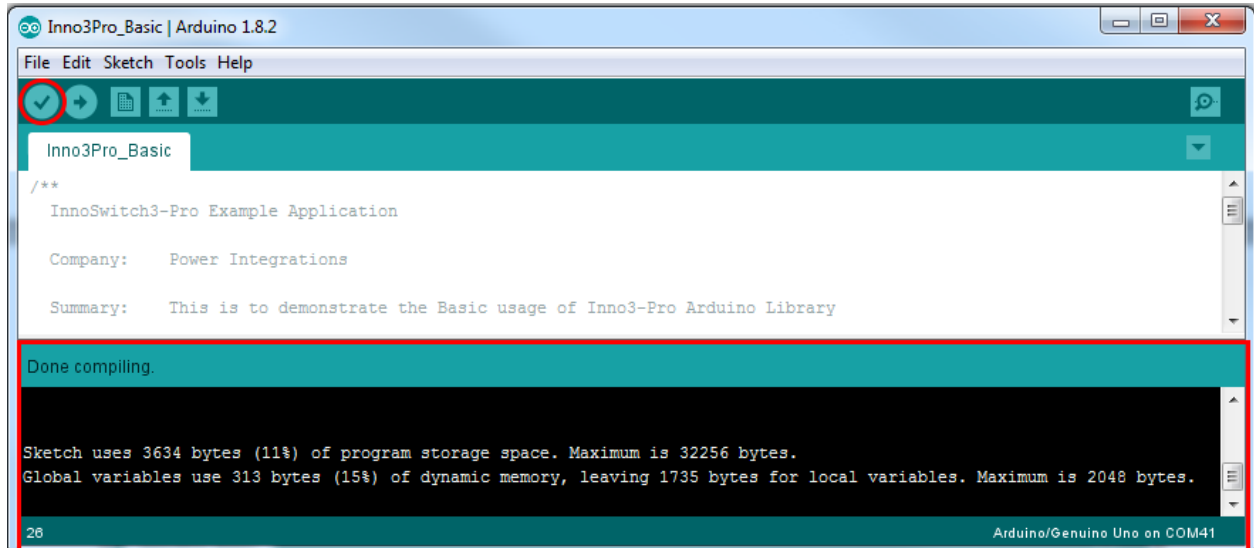
### 7.2 Select the Active Com Port

- ▶ Under tools menu, Select the correct port
- ▶ For Arduino UNO, the name will appear next to the serial port



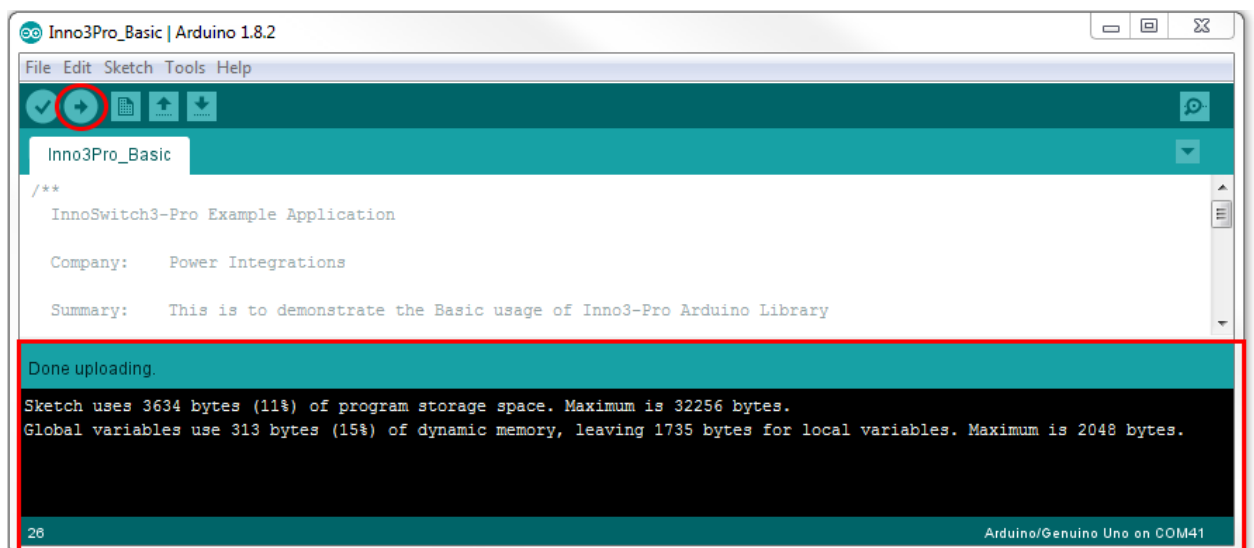
### 7.3 Verify / Compile

- ▶ Click the check icon to Verify
- ▶ After few seconds , “Done Compiling” should show up on the Notification Area  
This means the sketch is ready for uploading to the Arduino board



### 7.4 Upload

- ▶ Click the Arrow icon to Upload
- ▶ After few seconds , “Done Uploading” should show up on the Notification Area  
This means the upload was successful

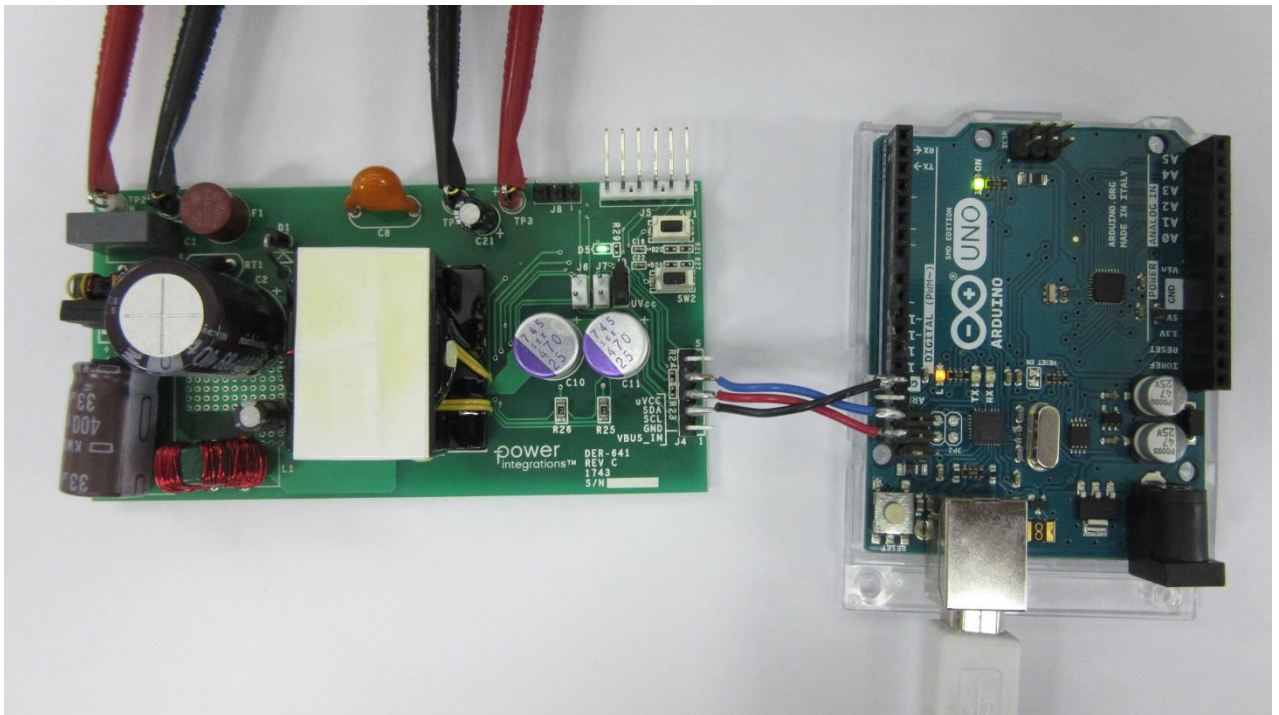


## 8 Demonstration of Operation

### 8.1 Running the Program

This section demonstrates code Example 1 or Example 2 in action. Upon power up, the Arduino program is uploaded to InnoSwitch3-Pro.

100VAC was applied to the AC input terminals and output terminals were connected to a Chroma DC electronic load. Arduino Uno board is connected to a USB port.





## 8.2 Constant voltage operation

- ▶ Image below shows the operation of RDK-641 at constant voltage of 5V and Full load of 5.1A



## 8.3 Constant current operation

- ▶ Image below shows the operation of RDK-641 at constant current mode

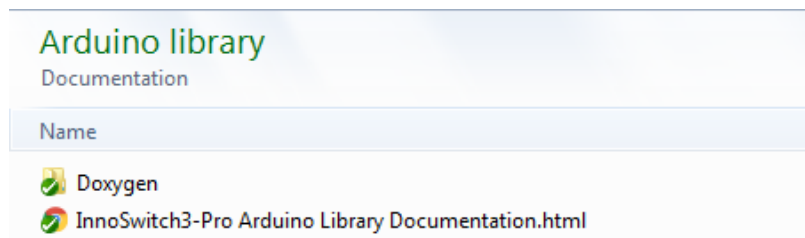


## 9 Doxygen Documentation

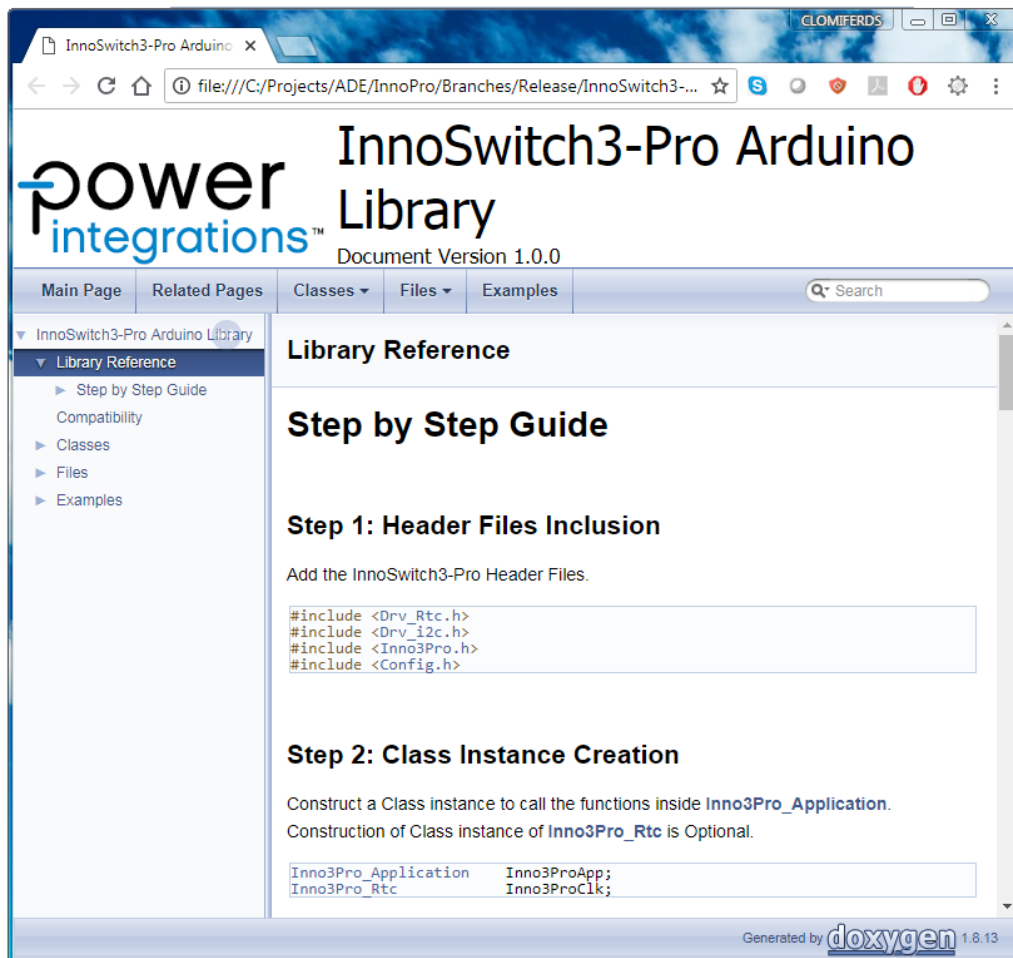
This document describes all of the data structures and functions that are part of the library. Many of these functions are intended to be used internally by the stack layers. The main public interfaces that are expected to be used by user code are summarized in the Doxygen Documentation Folder.

### 9.1 Opening html file

- ▶ Search the InnoSwitch3-Pro code library folder and Browse to documentation folder ,then Open the \*.html File



- ▶ Image below shows the doxygen html file:



## 9.2 Viewing the API Functions

- ▶ Under Inno3\_Application Class List,
- ▶ Select and Open Function Summary and Description:



The screenshot displays the InnoSwitch3-Pro Arduino Library documentation page. The page title is "InnoSwitch3-Pro Arduino Library" with "Document Version 1.0.0". The left sidebar shows the "Classes" section with "Inno3Pro\_Application" selected. The main content area is titled "API Write Functions" and lists several functions. The function "void Inno3Pro\_Bleeder\_Enable (bool bEnable)" is highlighted with a red box. The function description is "Handles Bleeder Setting. More...".

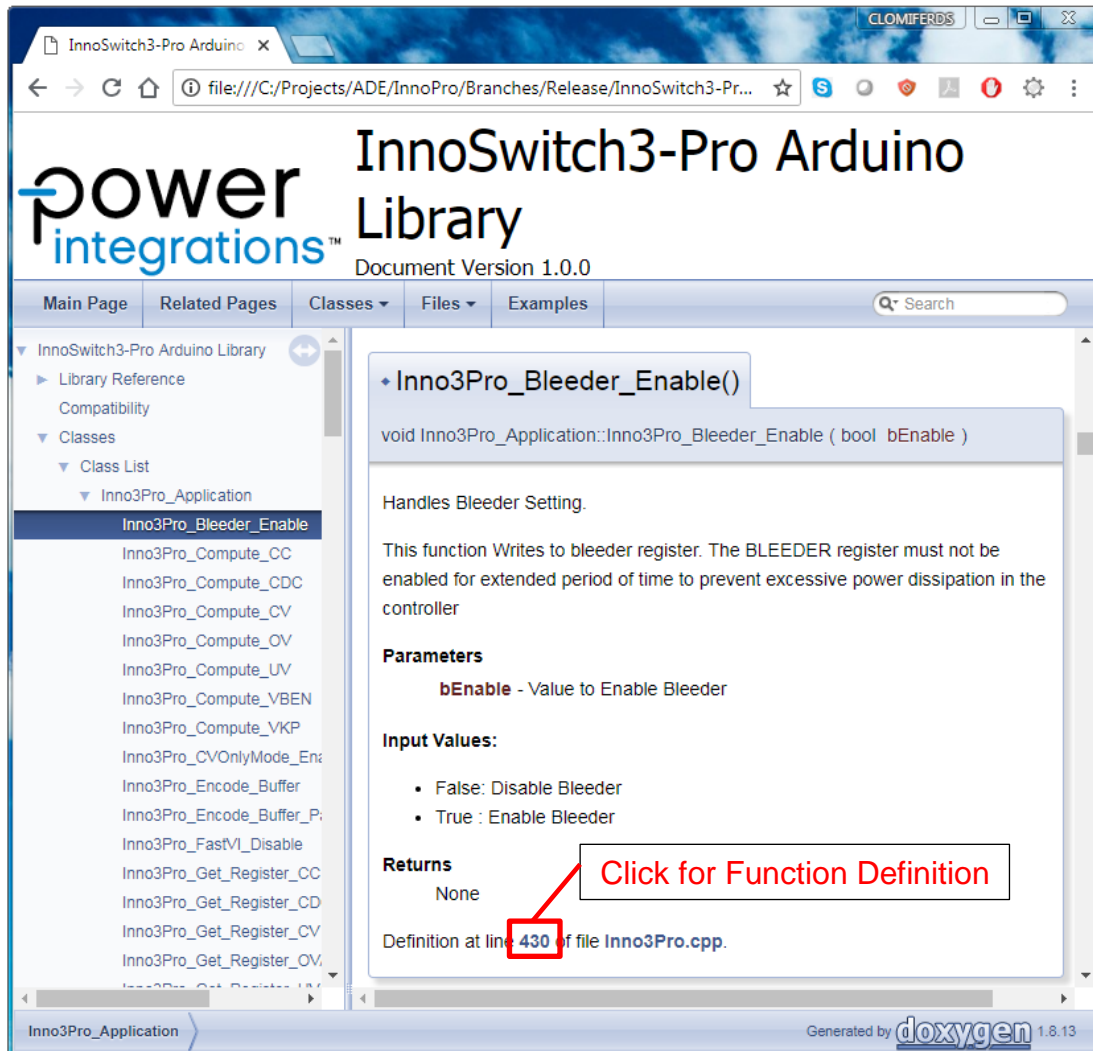
**API Write Functions**  
Application Programming Interface to control InnoSwitch3-Pro

- void **Inno3Pro\_Initialization** (void)  
Handles all Common I2C Configurations to be written to InnoSwitch3-Pro as initialization. More...
- void **Inno3Pro\_Vbus\_Switch\_Control** (bool bEnableVben)  
Vbus Switch Control (VBEN Control) More...
- void **Inno3Pro\_Bleeder\_Enable** (bool bEnable)  
Handles Bleeder Setting. More...
- void **Inno3Pro\_Load\_Discharge** (bool bEnable)  
Activates Vbus Load Discharge. More...
- void **Inno3Pro\_TurnOff\_PSU** (bool bEnable)  
Turns off the power supply. More...
- void **Inno3Pro\_FastVI\_Disable** (bool bDisable)  
CV and CC Commands speed limit. More...
- void **Inno3Pro\_CVOnlyMode\_Enable** (bool bEnable)  
Constant voltage only mode. More...

Inno3Pro\_Application | Generated by **doxygen** 1.8.13

### 9.3 Functions summary

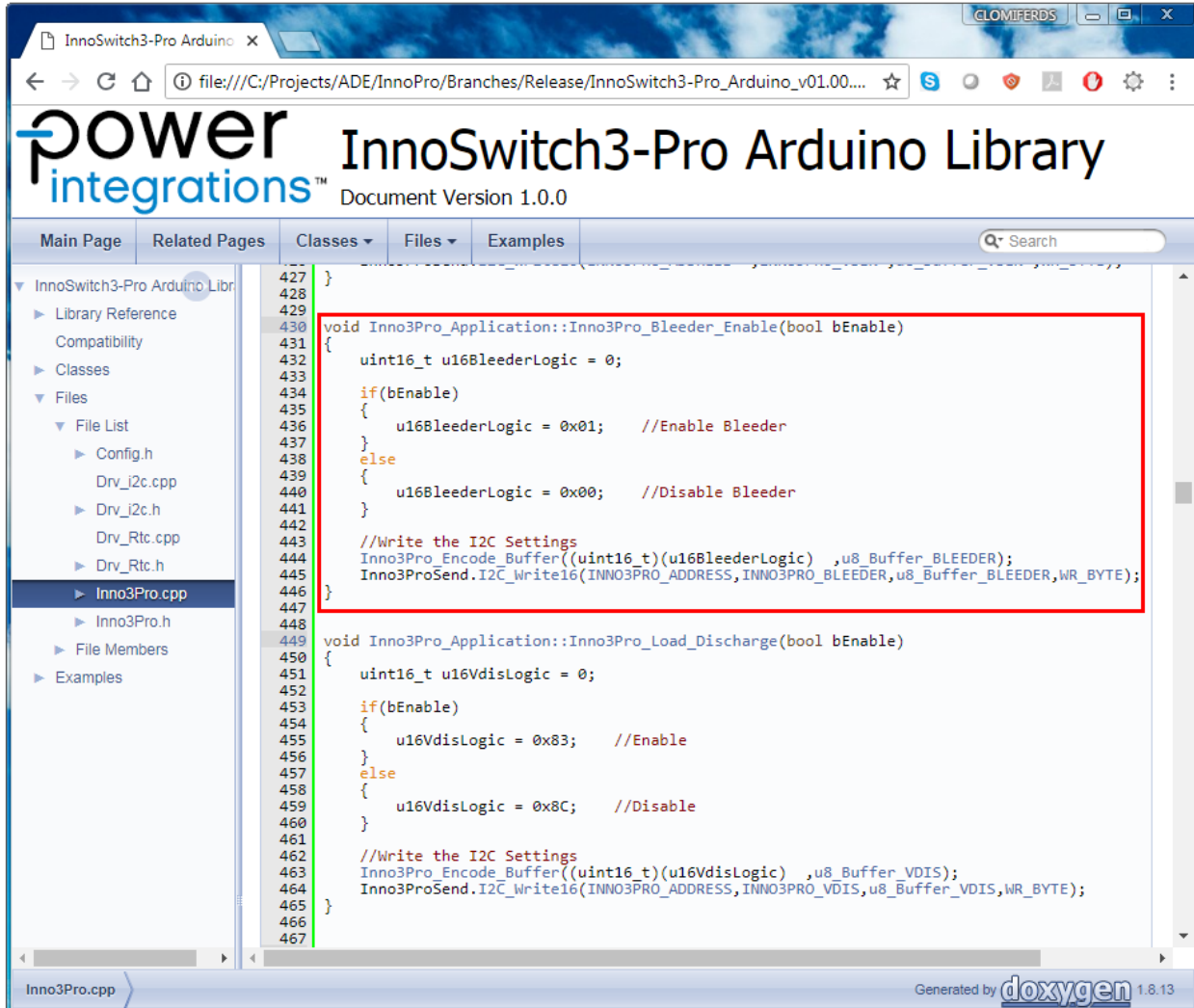
This section provides details and summary of how the function works.



The screenshot shows a web browser displaying the InnoSwitch3-Pro Arduino Library documentation. The page title is "InnoSwitch3-Pro Arduino Library" with a sub-header "Document Version 1.0.0". The navigation menu includes "Main Page", "Related Pages", "Classes", "Files", and "Examples". The left sidebar shows a tree view of the library structure, with "Inno3Pro\_Bleeder\_Enable" selected under the "Inno3Pro\_Application" class. The main content area displays the function signature: `void Inno3Pro_Application::Inno3Pro_Bleeder_Enable ( bool bEnable )`. Below the signature, it states "Handles Bleeder Setting." and provides a description: "This function Writes to bleeder register. The BLEEDER register must not be enabled for extended period of time to prevent excessive power dissipation in the controller". The "Parameters" section lists `bEnable` as the value to enable the bleeder. The "Input Values" section lists `False` as "Disable Bleeder" and `True` as "Enable Bleeder". The "Returns" section lists "None". A red box highlights the number "430" in the text "Definition at line 430 of file Inno3Pro.cpp.", with a red arrow pointing to a callout box that says "Click for Function Definition". The footer of the page indicates it was generated by doxygen 1.8.13.

## 9.4 Functions definition

This section provides the actual body and implementation of the function



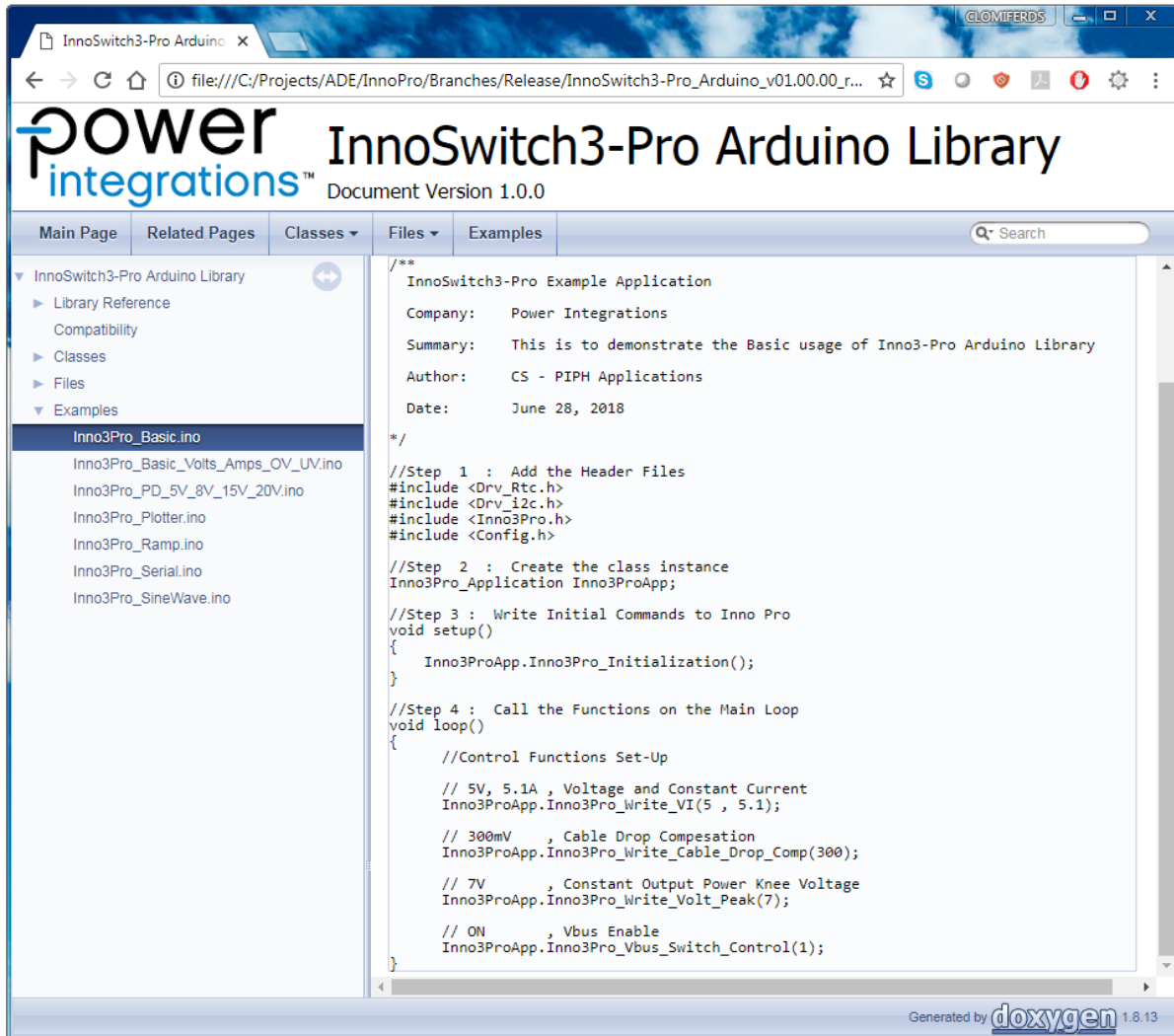
The screenshot shows a web browser displaying the InnoSwitch3-Pro Arduino Library documentation. The page title is "InnoSwitch3-Pro Arduino Library" and the document version is 1.0.0. The left sidebar shows a navigation menu with "Inno3Pro.cpp" selected. The main content area displays the C++ code for the `Inno3Pro_Application::Inno3Pro_Bleeder_Enable` function, which is highlighted with a red box. The code is as follows:

```
427 }
428
429
430 void Inno3Pro_Application::Inno3Pro_Bleeder_Enable(bool bEnable)
431 {
432     uint16_t u16BleederLogic = 0;
433
434     if(bEnable)
435     {
436         u16BleederLogic = 0x01; //Enable Bleeder
437     }
438     else
439     {
440         u16BleederLogic = 0x00; //Disable Bleeder
441     }
442
443     //Write the I2C Settings
444     Inno3Pro_Encode_Buffer((uint16_t)(u16BleederLogic) ,u8_Buffer_BLEEDER);
445     Inno3ProSend.I2C_Write16(INNO3PRO_ADDRESS,INNO3PRO_BLEEDER,u8_Buffer_BLEEDER,WR_BYTE);
446 }
447
448
449 void Inno3Pro_Application::Inno3Pro_Load_Discharge(bool bEnable)
450 {
451     uint16_t u16VdisLogic = 0;
452
453     if(bEnable)
454     {
455         u16VdisLogic = 0x83; //Enable
456     }
457     else
458     {
459         u16VdisLogic = 0x8C; //Disable
460     }
461
462     //Write the I2C Settings
463     Inno3Pro_Encode_Buffer((uint16_t)(u16VdisLogic) ,u8_Buffer_VDIS);
464     Inno3ProSend.I2C_Write16(INNO3PRO_ADDRESS,INNO3PRO_VDIS,u8_Buffer_VDIS,WR_BYTE);
465 }
466
467
```

The code is generated by doxygen 1.8.13.

## 9.5 Examples

This section provides different examples that showcase the use of the library functions



The screenshot displays the InnoSwitch3-Pro Arduino Library documentation website. The page title is "InnoSwitch3-Pro Arduino Library" with a sub-header "Document Version 1.0.0". The navigation menu includes "Main Page", "Related Pages", "Classes", "Files", and "Examples". The "Examples" section is selected, and the "Inno3Pro\_Basic.ino" file is highlighted in the left sidebar. The main content area shows the code for the "InnoSwitch3-Pro Example Application".

```
/**
 * InnoSwitch3-Pro Example Application
 * Company: Power Integrations
 * Summary: This is to demonstrate the Basic usage of Inno3-Pro Arduino Library
 * Author: CS - PIPH Applications
 * Date: June 28, 2018
 */

//Step 1 : Add the Header Files
#include <Drv_Rtc.h>
#include <Drv_i2c.h>
#include <Inno3Pro.h>
#include <Config.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, 5.1A , Voltage and Constant Current
    Inno3ProApp.Inno3Pro_Write_VI(5 , 5.1);

    // 300mV , Cable Drop Compesation
    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);
}
```

Generated by [doxygen](#) 1.8.13



## 10 Revision History

Date	Author	Revision	Description & changes	Reviewed
06-Sep-18	CS	1.0	Initial Release	Apps and Mktg



**For the latest updates, visit our website: [www.power.com](http://www.power.com)**

Reference Designs are technical proposals concerning how to use Power Integrations' gate drivers in particular applications and/or with certain power modules. These proposals are "as is" and are not subject to any qualification process. The suitability, implementation and qualification are the sole responsibility of the end user. The statements, technical information and recommendations contained herein are believed to be accurate as of the date hereof. All parameters, numbers, values and other technical data included in the technical information were calculated and determined to our best knowledge in accordance with the relevant technical norms (if any). They may be based on assumptions or operational conditions that do not necessarily apply in general. We exclude any representation or warranty, express or implied, in relation to the accuracy or completeness of the statements, technical information and recommendations contained herein. No responsibility is accepted for the accuracy or sufficiency of any of the statements, technical information, recommendations or opinions communicated and any liability for any direct, indirect or consequential loss or damage suffered by any person arising therefrom is expressly disclaimed.

Power Integrations reserves the right to make changes to its products at any time to improve reliability or manufacturability. Power Integrations does not assume any liability arising from the use of any device or circuit described herein. POWER INTEGRATIONS MAKES NO WARRANTY HEREIN AND SPECIFICALLY DISCLAIMS ALL WARRANTIES INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

**Patent Information**

The products and applications illustrated herein (including transformer construction and circuits' external to the products) may be covered by one or more U.S. and foreign patents, or potentially by pending U.S. and foreign patent applications assigned to Power Integrations. A complete list of Power Integrations' patents may be found at [www.power.com](http://www.power.com). Power Integrations grants its customers a license under certain patent rights as set forth at <http://www.power.com/ip.htm>.

The PI Logo, TOPSwitch, TinySwitch, LinkSwitch, LYTSwitch, InnoSwitch, DPA-Switch, PeakSwitch, CAPZero, SENZero, LinkZero, HiperPFS, HiperTFS, HiperLCS, Qspeed, EcoSmart, Clamless, E-Shield, Filterfuse, FluxLink, StadFET, PI Expert and PI FACTS are trademarks of Power Integrations, Inc. Other trademarks are property of their respective companies. ©Copyright 2015 Power Integrations, Inc.

**Power Integrations Worldwide Sales Support Locations****WORLD HEADQUARTERS**

5245 Hellyer Avenue  
San Jose, CA 95138, USA.  
Main: +1-408-414-9200  
Customer Service:  
Phone: +1-408-414-9665  
Fax: +1-408-414-9765  
e-mail: [usasales@power.com](mailto:usasales@power.com)

**GERMANY (IGBT Driver Sales)**

HellwegForum 1  
59469 Ense, Germany  
Tel: +49-2938-64-39990  
Email: [igbt-driver.sales@power.com](mailto:igbt-driver.sales@power.com)

**KOREA**

RM 602, 6FL  
Korea City Air Terminal B/D,  
159-6  
Samsung-Dong, Kangnam-Gu,  
Seoul, 135-728 Korea  
Phone: +82-2-2016-6610  
Fax: +82-2-2016-6630  
e-mail: [koreasales@power.com](mailto:koreasales@power.com)

**CHINA (SHANGHAI)**

Rm 2410, Charity Plaza, No. 88,  
North Caoxi Road,  
Shanghai, PRC 200030  
Phone: +86-21-6354-6323  
Fax: +86-21-6354-6325  
e-mail: [chinasales@power.com](mailto:chinasales@power.com)

**INDIA**

#1, 14<sup>th</sup> Main Road  
Vasanthanagar  
Bangalore-560052  
India  
Phone: +91-80-4113-8020  
Fax: +91-80-4113-8023  
e-mail: [indiasales@power.com](mailto:indiasales@power.com)

**SINGAPORE**

51 Newton Road,  
#19-01/05 Goldhill Plaza  
Singapore, 308900  
Phone: +65-6358-2160  
Fax: +65-6358-2015  
e-mail:  
[singaporesales@power.com](mailto:singaporesales@power.com)

**CHINA (SHENZHEN)**

17/F, Hivac Building, No. 2, Keji Nan  
8th Road, Nanshan District,  
Shenzhen, China, 518057  
Phone: +86-755-8672-8689  
Fax: +86-755-8672-8690  
e-mail: [chinasales@power.com](mailto:chinasales@power.com)

**ITALY**

Via Milanese 20, 3<sup>rd</sup> Fl.  
20099 Sesto San Giovanni (MI) Italy  
Phone: +39-024-550-8701  
Fax: +39-028-928-6009  
e-mail: [eurosales@power.com](mailto:eurosales@power.com)

**TAIWAN**

5F, No. 318, Nei Hu Rd.,  
Sec. 1  
Nei Hu District  
Taipei 11493, Taiwan R.O.C.  
Phone: +886-2-2659-4570  
Fax: +886-2-2659-4550  
e-mail: [taiwansales@power.com](mailto:taiwansales@power.com)

**GERMANY (AC-DC/LED Sales)**

Lindwurmstrasse 114  
80337, Munich  
Germany  
Phone: +49-895-527-39110  
Fax: +49-895-527-39200  
e-mail: [eurosales@power.com](mailto:eurosales@power.com)

**JAPAN**

Kosei Dai-3 Building  
2-12-11, Shin-Yokohama,  
Kohoku-ku, Yokohama-shi,  
Kanagawa 222-0033  
Japan  
Phone: +81-45-471-1021  
Fax: +81-45-471-3717  
e-mail:  
[japansales@power.com](mailto:japansales@power.com)

**UK**

Cambridge Semiconductor,  
a Power Integrations company  
Westbrook Centre, Block 5,  
2nd Floor  
Milton Road  
Cambridge CB4 1YG  
Phone: +44 (0) 1223-446483  
e-mail: [eurosales@power.com](mailto:eurosales@power.com)

