iND80230 “Herzog”

Herzog EVKit starter guide

1/4/16 **Application Note**

1.0 TABLE OF CONTENTS

2.0 LIST OF FIGURES

3.0 HERZOG HARDWARE INTRODUCTION

3.1 HERZOG IC

Herzog integrates an ARM Cortex-M0 low cost 32-bit microcontroller containing 160kB of flash program memory and 8kB of SRAM together with a complete ISM band transceiver. It implements several general-purpose peripherals. Its main features are:

CPU Architecture:

- ARM Cortex-M0 processor running at 10MHz (Internal RC), 32.768kHz RTC, or 10kHz (Internal auxiliary RC), 30MHz Crystal Oscillator (Required for RF)
- System Tick Timer (SysTick – 24 bits, interruptible)
- Serial Wire Debugger
- Built-in Nested Vectored Interrupt Controller (NVIC)
- Programmable Watch-Dog Timer

Memory:

- 160kByte of Flash Program Memory
- 8kByte of SRAM

Peripherals:

- PIR Sensor Interface
- 300 MHz to 450 MHz ISM ASK transceiver
- 19 General purpose I/O ports
- 2 LED Drivers
- An ADC (8-10 bit including a windowing function), with a total of 19 input channels, with selectable input references and input gain block
- Low Battery Detection
- Wake-up Timer
- 30MHz oscillator; 32.786kHz RTC; 10kHz RC oscillator; 10MHz, 1% RC oscillator
- UART Interface
- SPI Interface

Package:

- 5mmx5mm 32 pin QFN package

Application:

Herzog provides abundant GPIO pins, the necessary hardware protocol interfaces, and enough memory for small applications. Herzog is designed to be an attractive solution for designers who are looking for an ISM-band radio transceiver together with a basic MCU with 32-bit performance, which nonetheless has comparable pricing to 8-bit MCUs.

Figure 1 shows the device block diagram. For more information, please check the Herzog Datasheet [1]

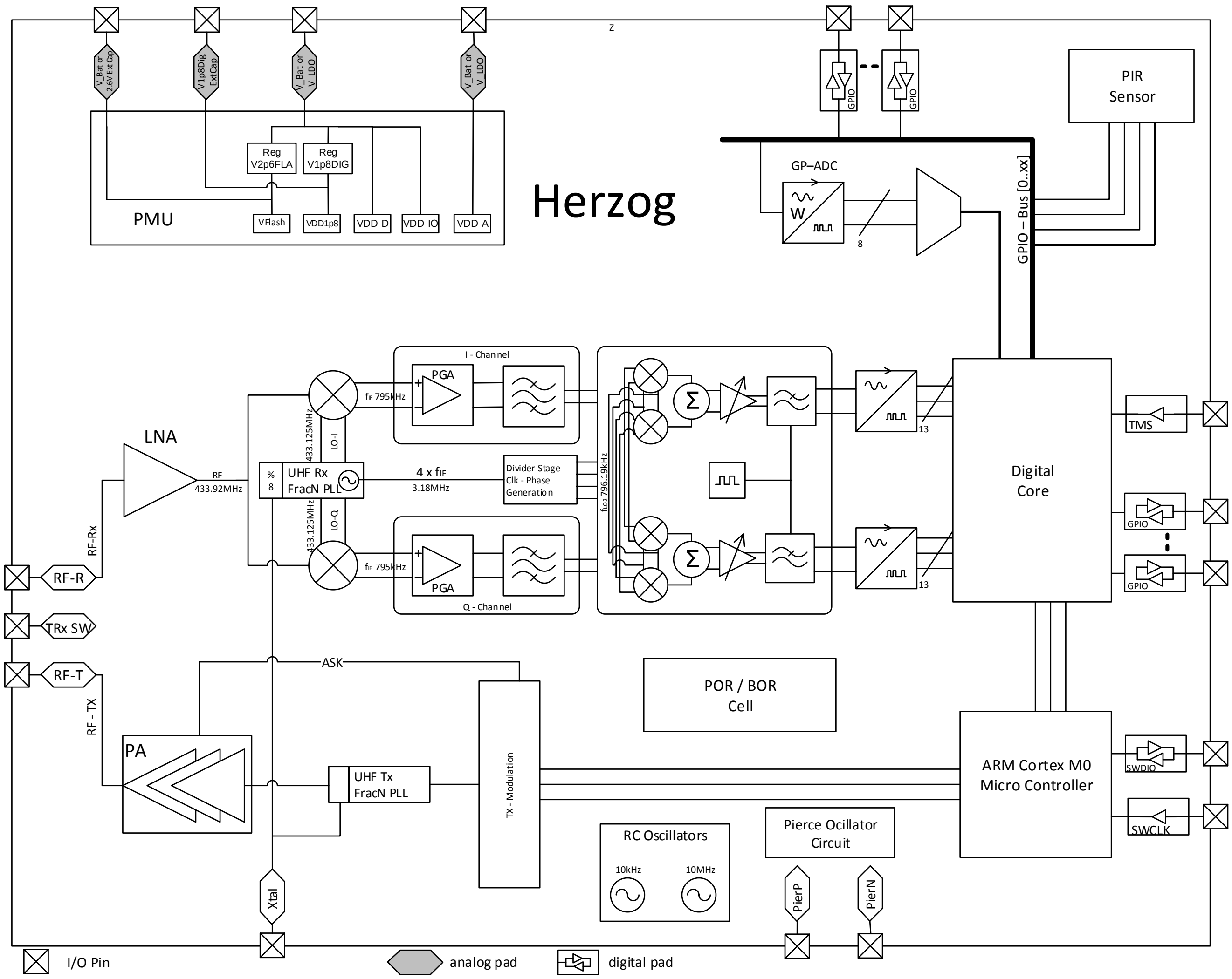


Figure 1 Herzog block diagram

3.2 HERZOG EVKIT

Herzog can be evaluated using an EVKit. This is a fast and easy way to program Herzog by connecting a board with USB micro B cable to the USB port of your laptop and launching IAR for project development.

The Herzog EVKit also include sensors and related programming so it can be used as a starting point for the user's own software development

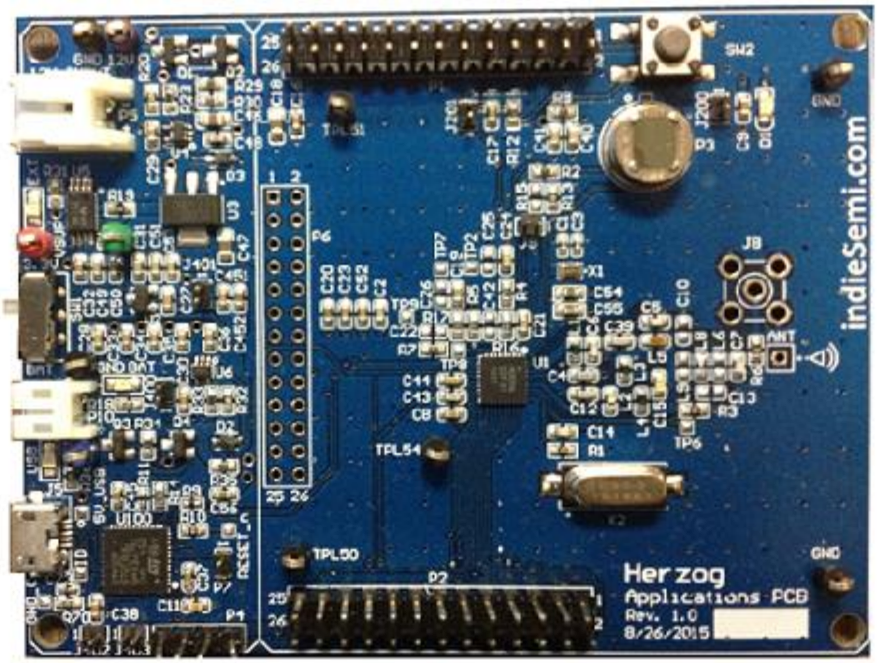


Figure 2 Herzog EVKit board

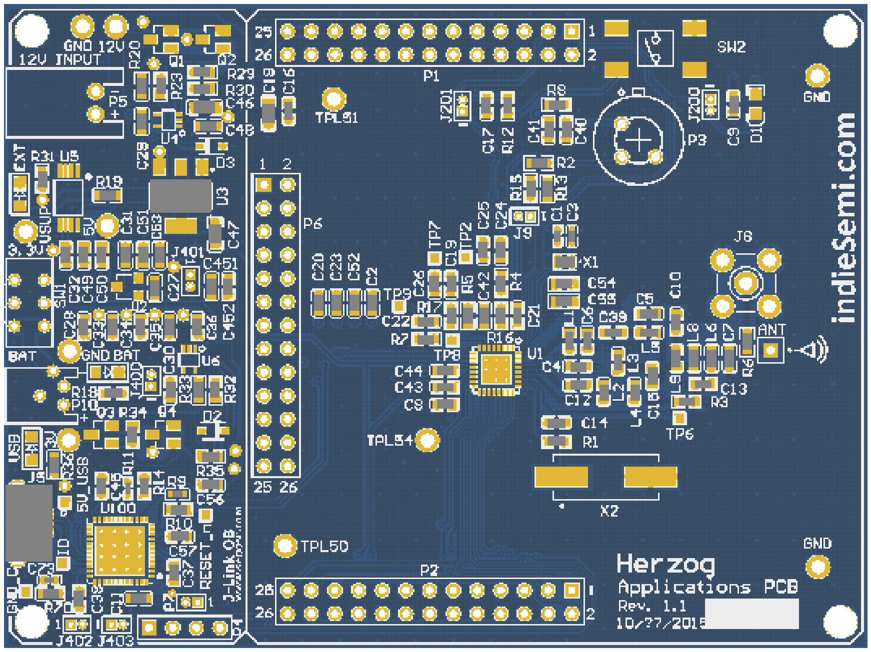


Figure 3 Herzog EVKit (layout)

Herzog EVKit Features

The Herzog EVKit has the following features:

- J-LINK-OB:

J-LINK-OB is the Segger low cost, space-saving solution as an onboard alternative to their standard J-LINK programmer/debugger. J-LINK-OB is designed with STM32F072 (Cortex-M). J-LINK-OB supports SWD (Serial Wire Debug) + the SWO interface. J-LINK-OB software is compatible with the regular J-LINK solutions, such as JLINK Commander, J-Flash, etc. J-LINK-OB can work with IAR Embedded Workbench, Keil MDK, and other popular IDEs.

J-LINK-OB also provides a Virtual COM UART (VCOM) port. This VCOM is connected to the Herzog micro to provide UART communication to the host PC through the USB port. Users can use terminal programs or a self-designed communication utility to interface with the Herzog EVKit. The PC driver for VCOM is readily available as plug-n-play driver. The VCOM uses no hardware handshaking (CTS and RTS are not implemented). Only RXD and TXD are needed for serial communication. Software handshaking is recommend for high speed communication.

J-LINK-OB USB can provide power for the entire PCB without any external supply. Refer to Power Management section.

- Herzog 32-bit Cortex M0

indie Semiconductor’s Herzog 32-bit ARM Cortex M0 (as described above). There is a 32.768kHz crystal on board to provide RTC function.

- Power Management

Herzog EVKit can be powered through J-LINK-OB USB. There is a 3.3V LDO regulator to power Herzog and other circuitry. The 3.3V LDO also powers the JLINK-OB STM32F072 circuit. The Herzog system can also be powered by an external 3.0V battery through a header (P10). A slider switch (SW1) is used to select between on-board 3.3V LDO and external battery. Jumpers (J400, J401) can be used to break the power supply path for the convenience of measuring system current consumption. Herzog EVKit also allows an external 12V input to power the system. The unit will be powered by either USB 5V or external 12V input (through 5V LDO) automatically. External 12V power and 3.0V battery power are reversible polarity protected. Furthermore, 3.0V battery power is also overvoltage protected (When input is >3.6V, the power is disconnected). For external 12V power, this feature is not fully implemented for the revision 1.0 PCB. Therefore, be careful not to exceed 13V on the 12V power input.

- PIR Sensor

Herzog EVKit features a PIR sensor (LHI968). LHI968 is specially designed for intrusion alarm applications. It includes a dual element pyroelectric ceramic with FET in source-follower connection and RF protection by a resistor and capacitor circuit. Refer to the LHI968 datasheet for details. Herzog EVKit uses integrated PIR sensor interfaces (two stage AC-coupled op-amp circuits) with external resistors and capacitors to optimize gain and bandwidth. LHI968 is biased with R2 (47Kohms). When R13 is depopulated, R15 populated and J9 is shorted, Herzog can optionally turn off LHI968’s bias to conserve battery power during sleep mode. This feature may not help designs where the application requires system wakeup through PIR. In that case, R2 can be tested with larger values to keep the LHI968 in good working condition but in the mean time reduce biasing current to conserve battery life.

- 433MHz Transceiver

This ASIC has an integrated super-heterodyne ISM band 300MHz to 450MHz ASK transceiver. The oscillator frequency for the RF is generated by fractional-N type frequency synthesizers from a 30MHz or 25MHz crystal oscillator

- User Switch

Herzog EVKit has a tactile switch that is connected to port D pin PD2. PD2 is capable of detecting pin state change to generate interrupts. This can be user programmed for his/her own application. Herzog has an internal pull-up resistor and therefore does not require an external resistor.

- LED indicators

Herzog EVKit has an on-board green color LED connected to Port D pin PD0, which has an integrated LED driver. This can be user programmed as required for the application. Another green LED (EXT) indicates the external 12V power supply is plugged in. Green LED (USB) indicates the USB power is connected. A Yellow LED (BAT) indicates the external 3.0V battery is connected.

- Extension Connectors

The Herzog EVKit has two extension headers that provide ground, supplies and all Herzog pins for users to connect to their system or to probe the Herzog pins.

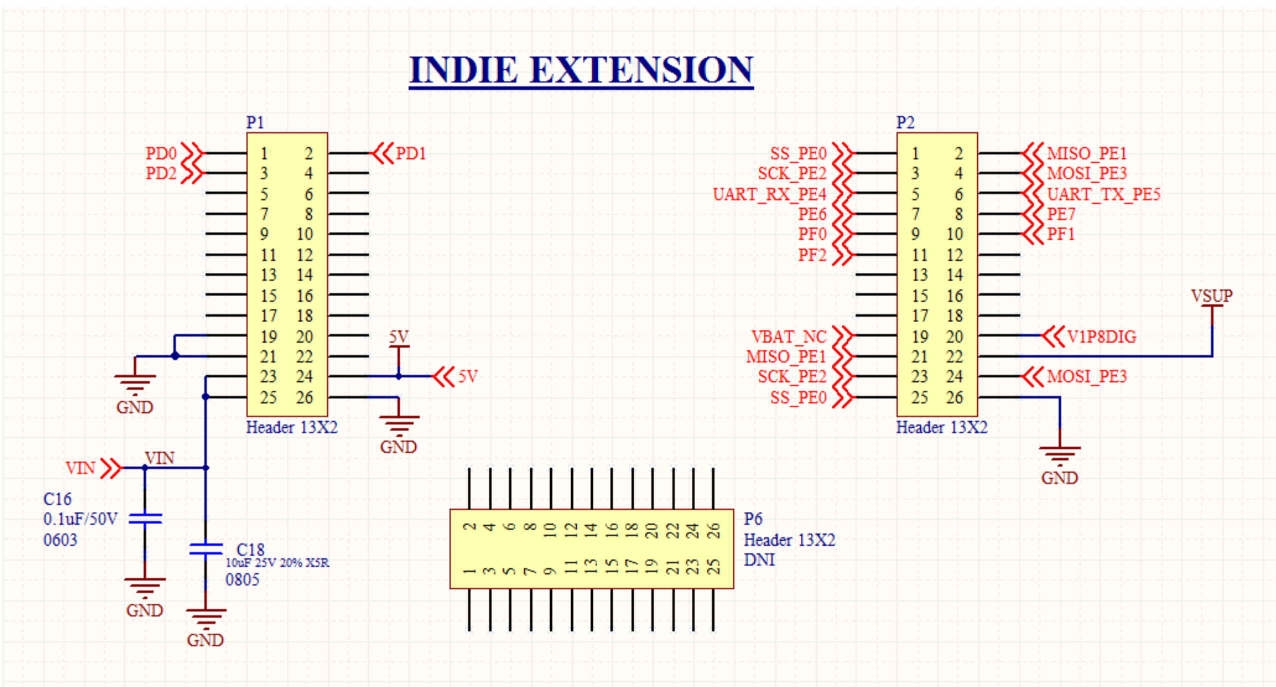


Figure 4 Herzog EVKit Extension Connectors mapping (P6 is reserved)

- Configuration Jumpers

Herzog EVKit has jumpers to configure the connection and setup. Please refer to 4.1.1 Jumpers and Switches.

3.2.1 Herzog EVKit Block Diagram

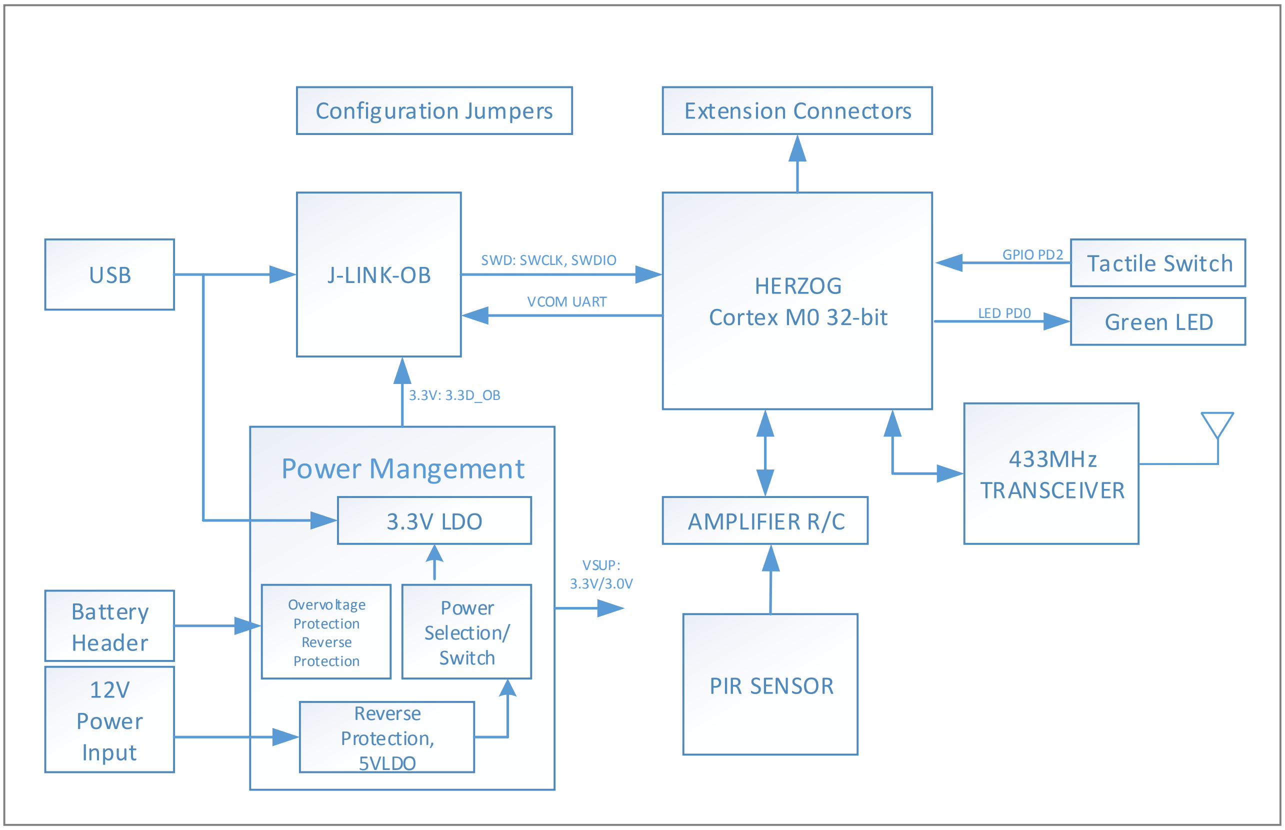
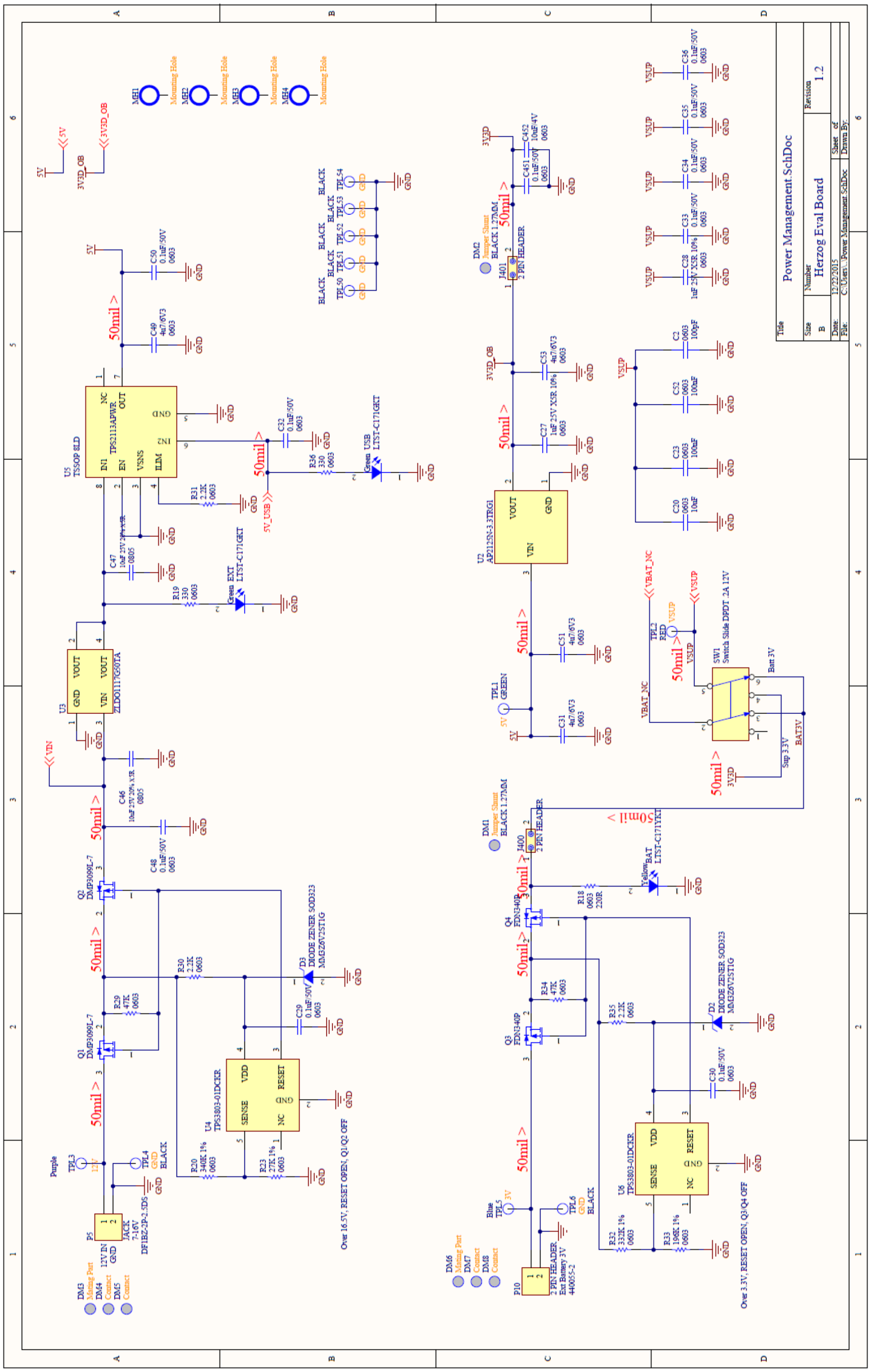
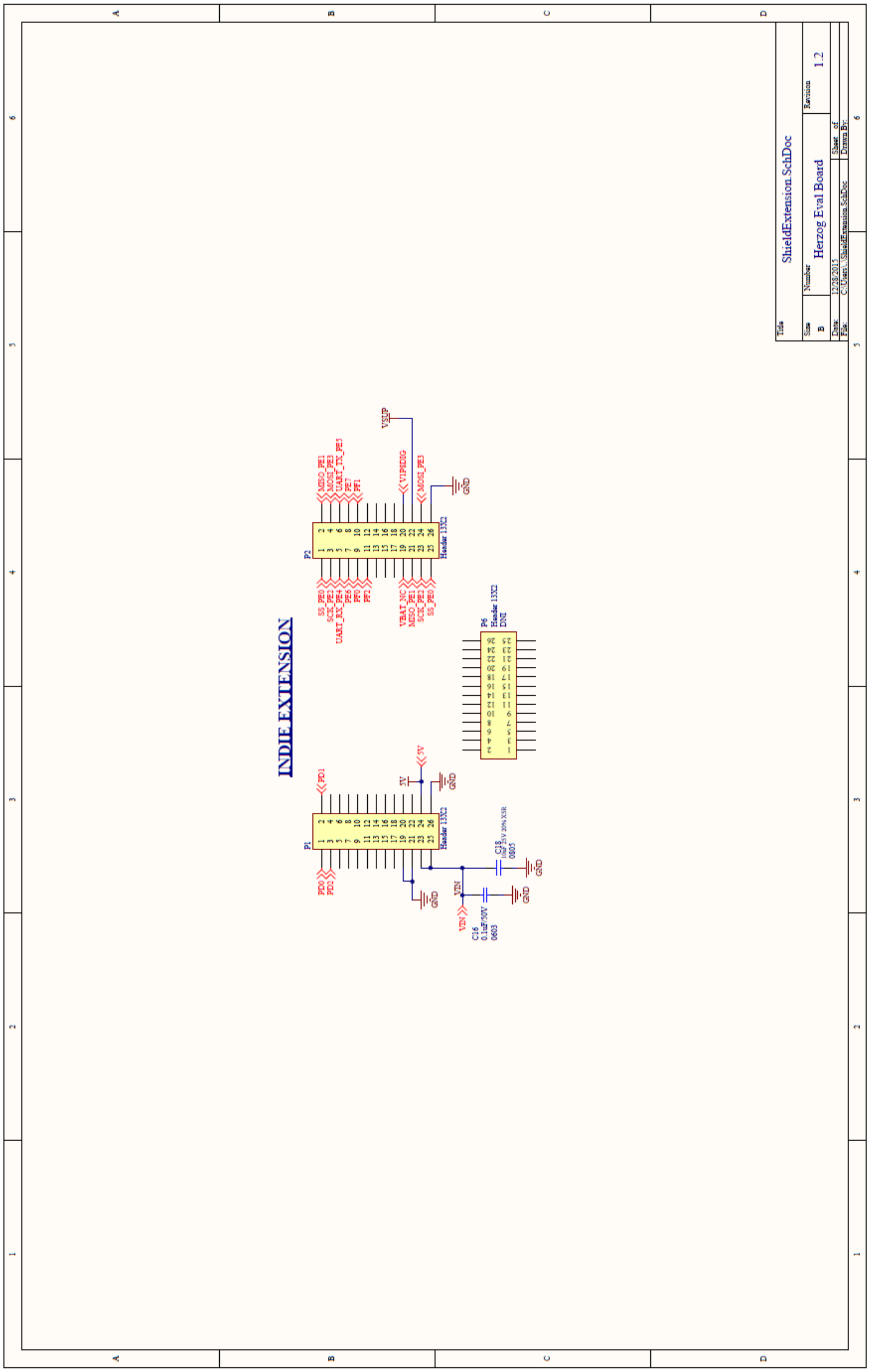
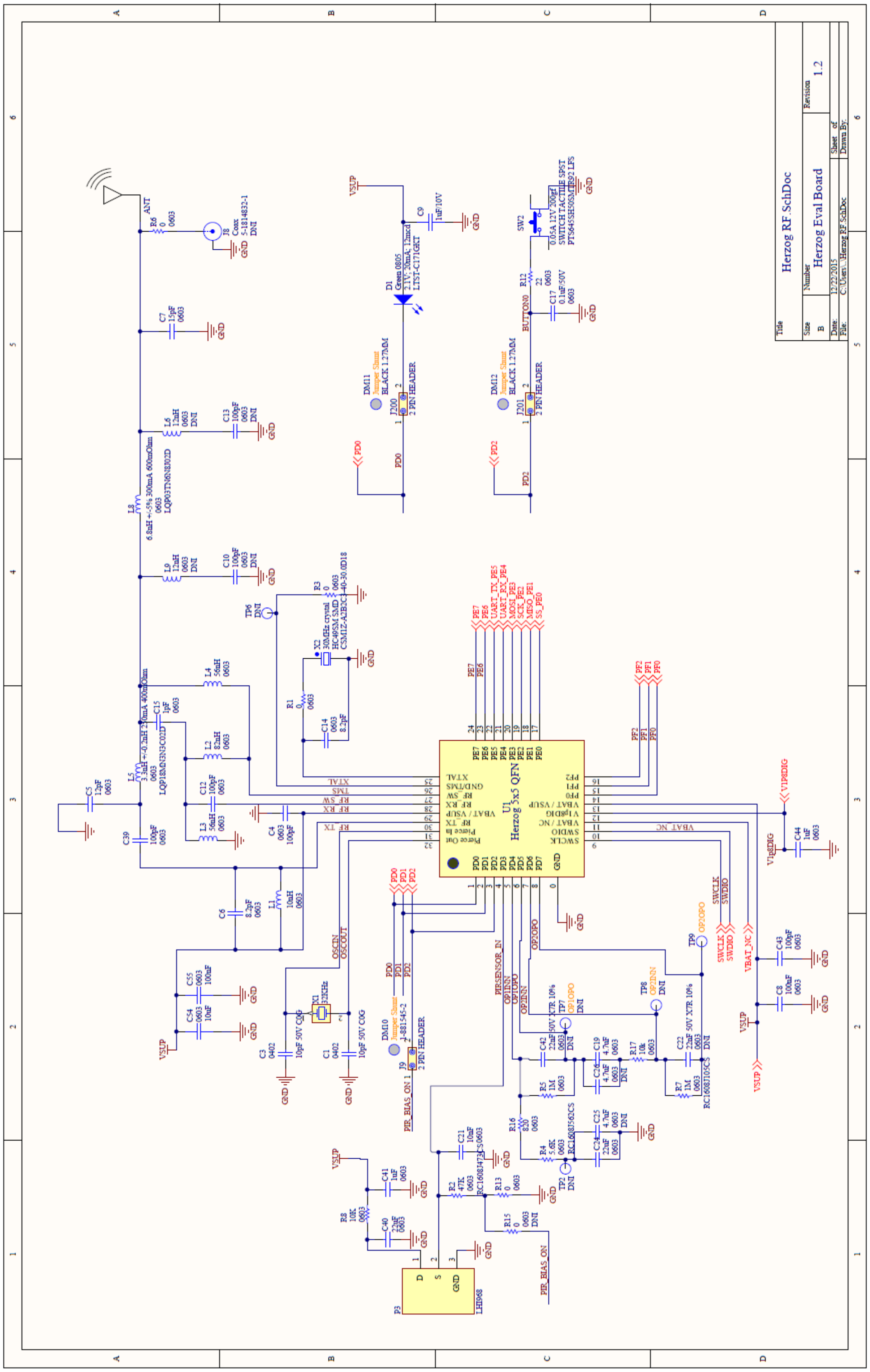


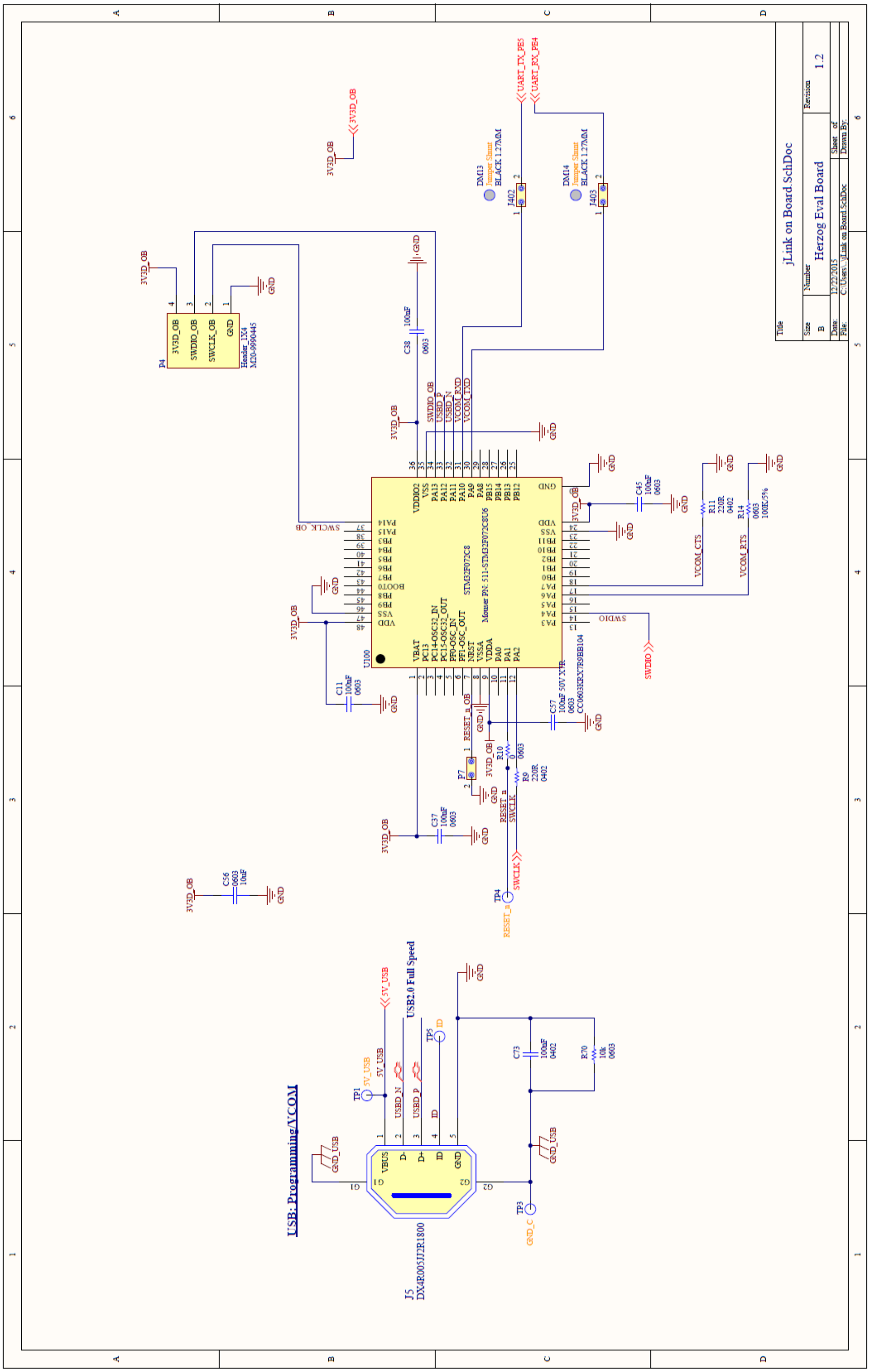
Figure 5 Herzog EVKit System Block Diagram

3.2.1 Herzog Schematics

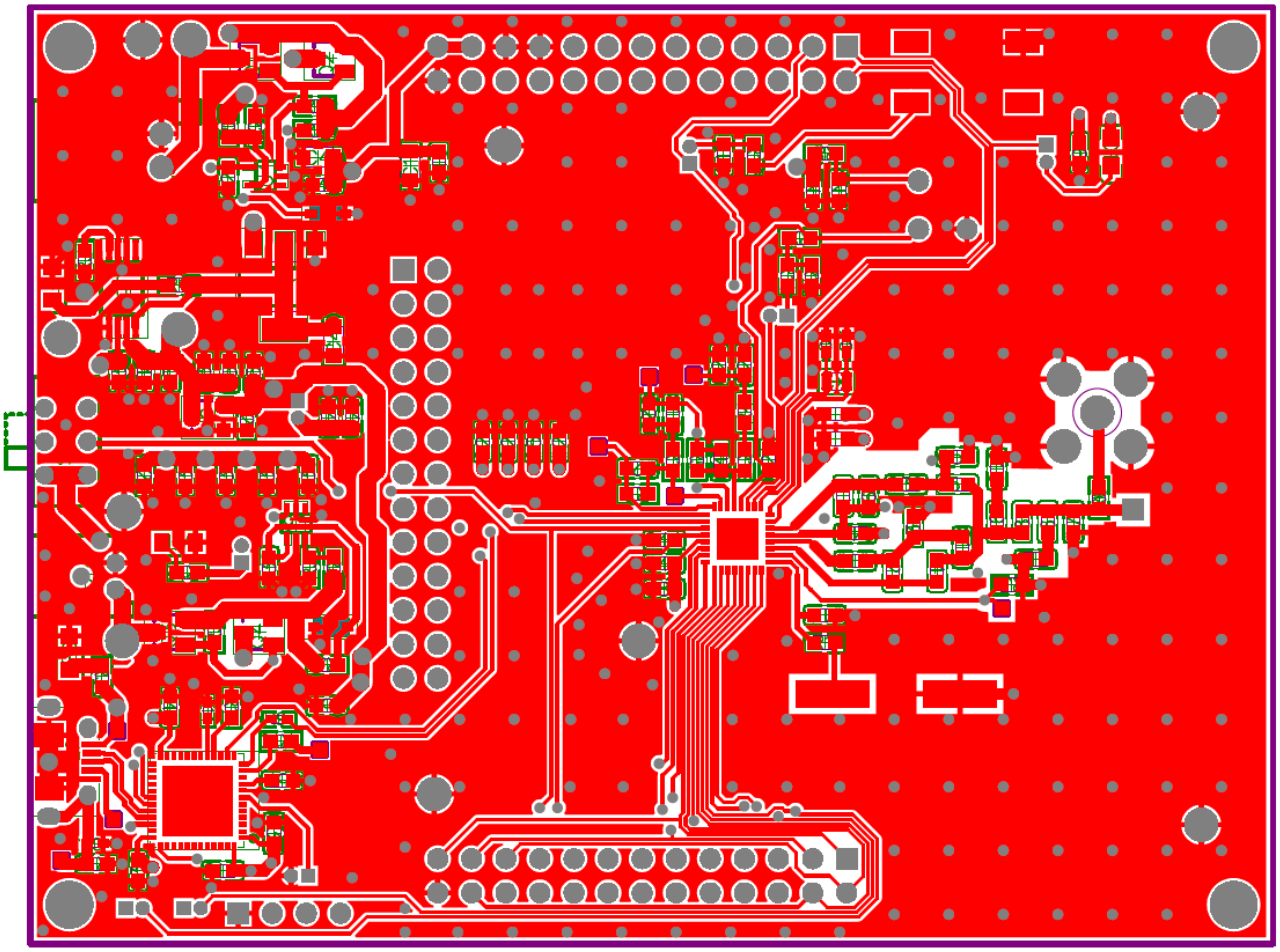

Power Management.SchDoc
Herzog Eval Board
Revision 1.2
Over 16.5V, RESET OPEN, Q1/Q2 OFF
Over 3.3V, RESET OPEN, Q3/Q4 OFF

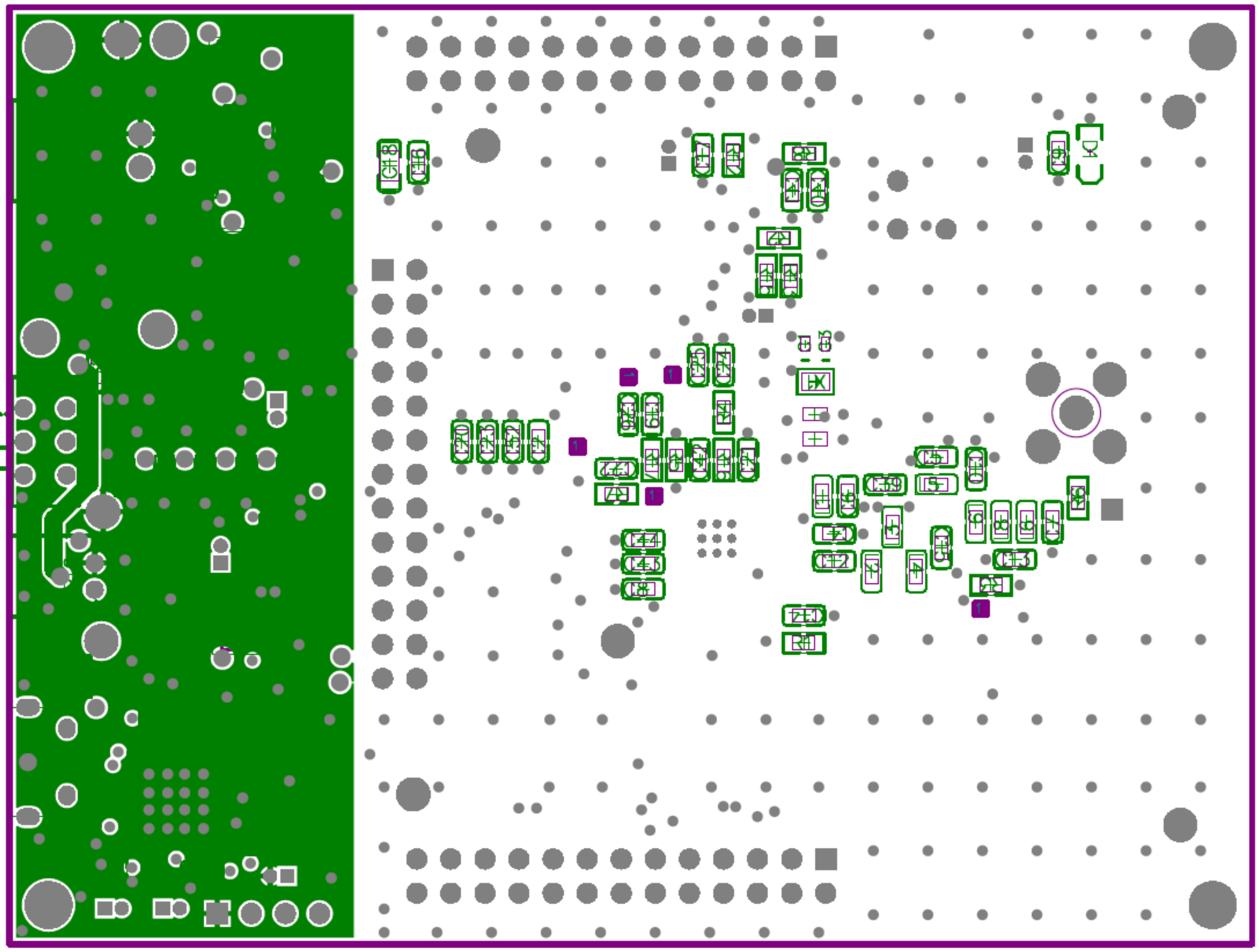

INDIE EXTENSION
ShieldExtension.SchDoc
Herzog Eval Board
1.2

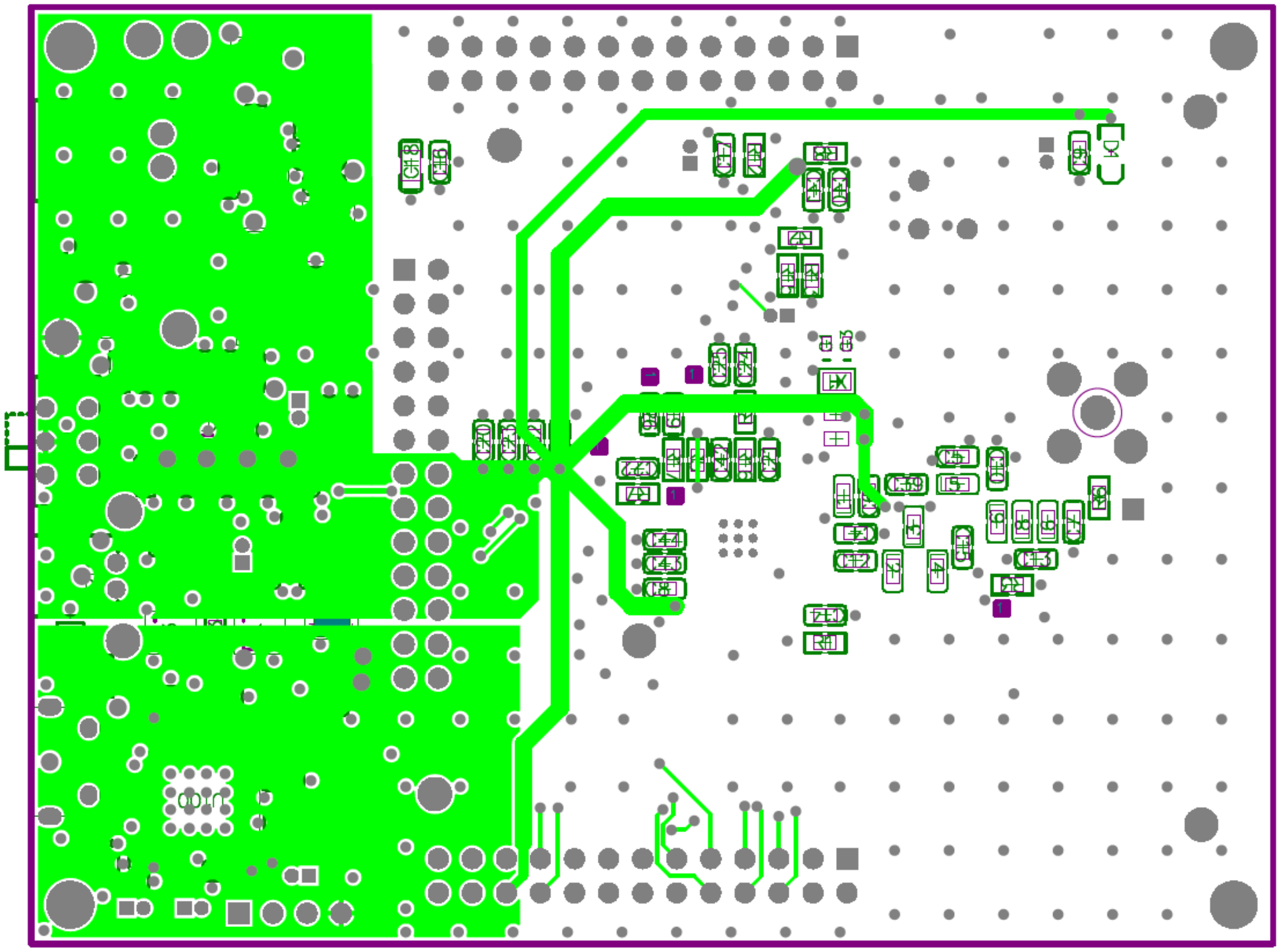

Herzog RF.SchDoc
Herzog Eval Board
Herzog 5x5 QFN

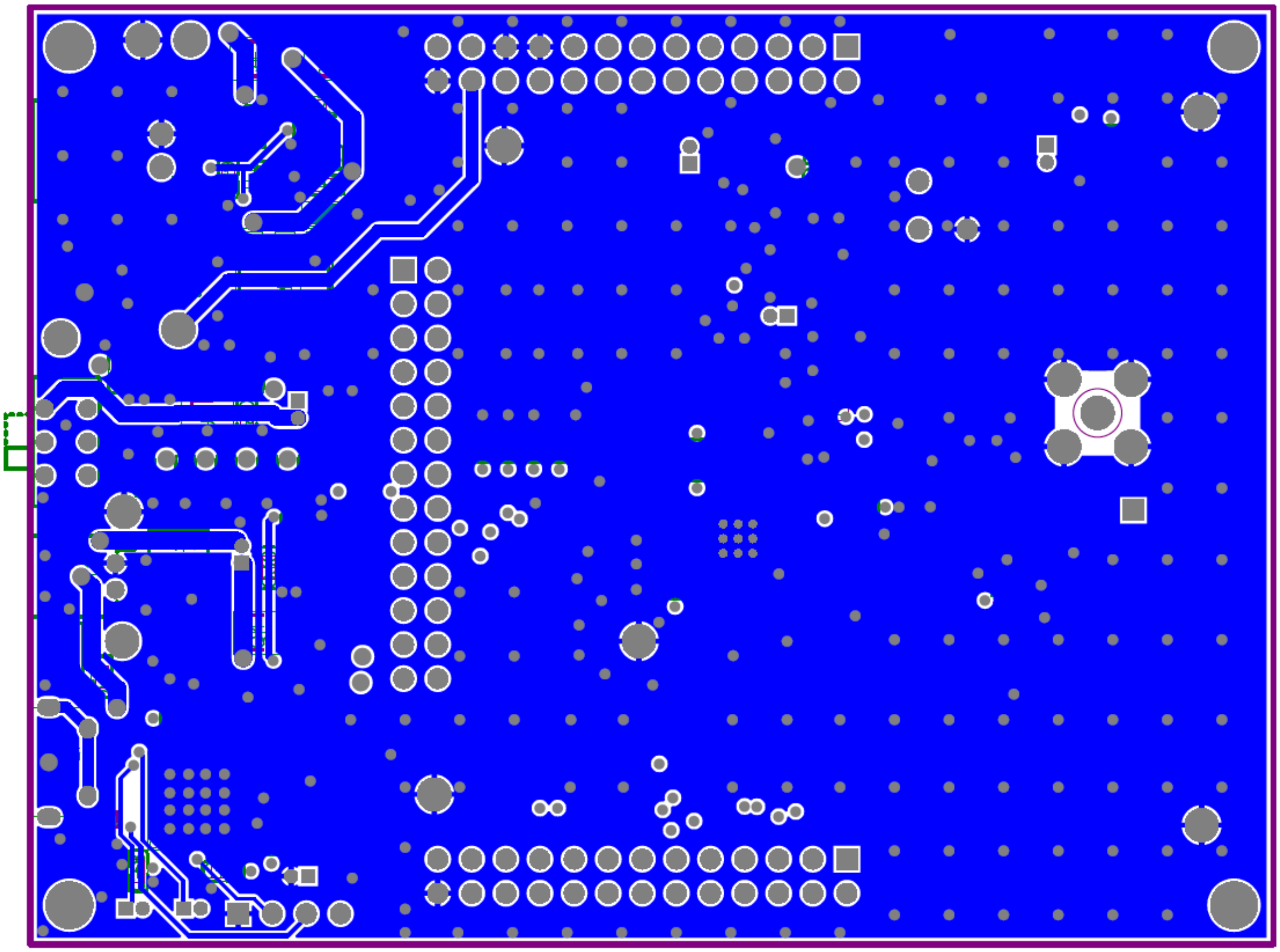

jLink on Board.SchDoc
Herzog Eval Board
1.2
USB: Programming/VCOM
USB2.0 Full Speed
STM32F072C8
J5
DX4R005JJ2R1800

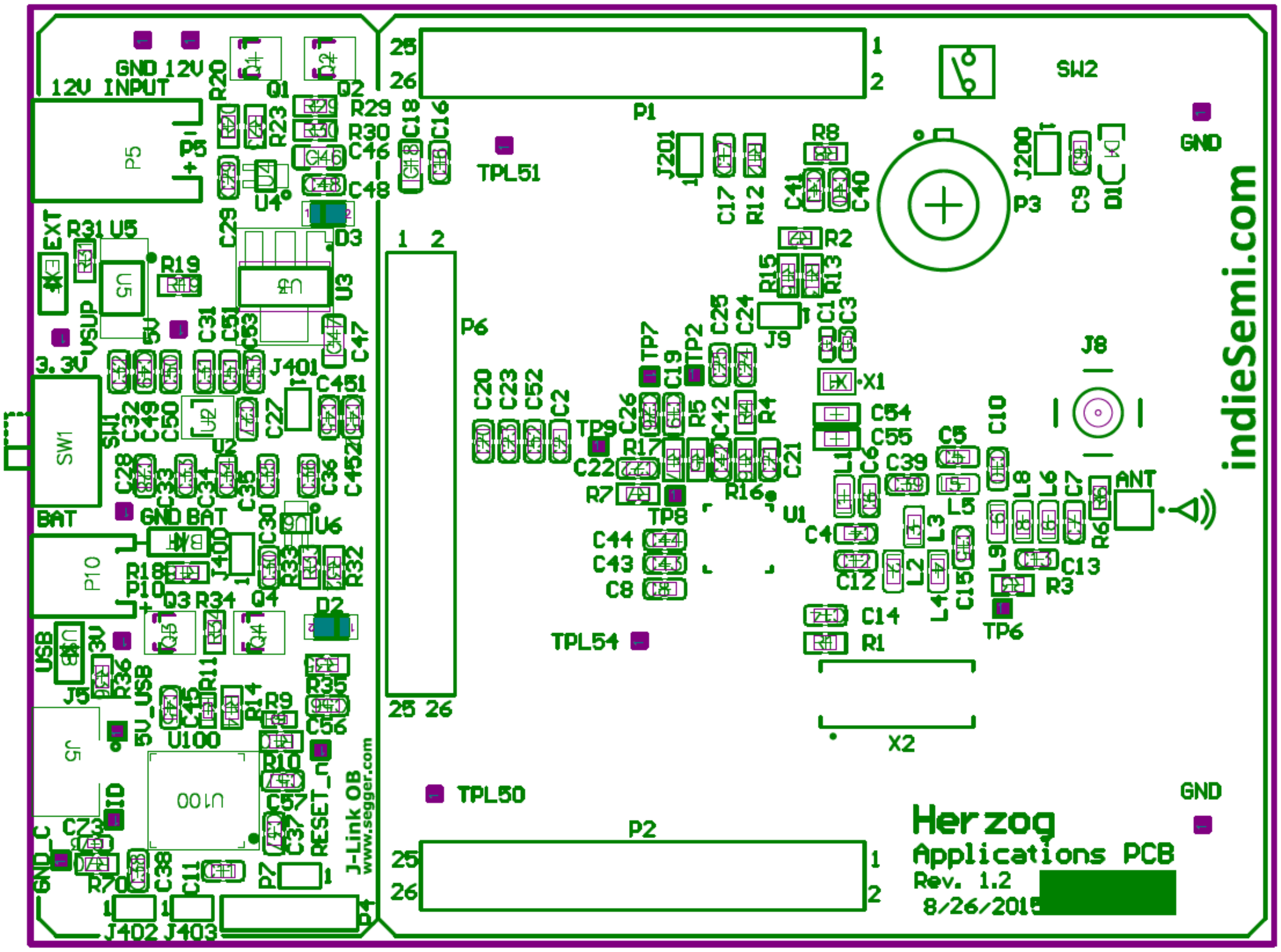
3.2.2 Herzog PCB Layout











3.2.3 EVKit BOM

Designator	Comment	Description	Manufacturer Part Number	Place	Quantity	Value
BAT	LED	LED YELLOW CLEAR THIN 0805 SMD	LTST-C171YKT		1	Yellow
C1, C3	0402	Capacitor	C1005C0G1H100D050BA		2	10pF 50V C0G
C2, C4, C10, C12, C13, C39, C43	0603	CAPACITOR	C1608C0G1H101J080AA	DNI	7	100pF
C5	0603	Capacitor	GQM1885C1H120GB01D		1	12pF
C6, C14	0603	Capacitor	GQM1885C1H8R2CB01D		2	8.2pF
C7	0603	Capacitor	GQM1885C1H150GB01D		1	15pF

Designator	Comment	Description	Manufacturer Part Number	Place	Quantity	Value
C8, C11, C16, C17, C23, C29, C30, C32, C33, C34, C35, C36, C37, C38, C45, C48, C50, C52, C55, C57, C451	0603	Capacitor	CC0603KRX7R9BB104		21	0.1uF/50V, 100nF, 100nF 50V X7R
C9	0603	Capacitor	C1608X6S1H105K080AC		1	1uF/10V
C15	0603	Capacitor	ML03511R0AAT2A		1	1pF
C18, C46, C47	0805	Capacitor	GRM21BR61E106MA73L		3	10uF 25V 20% X5R
C19, C25, C26, C31, C49, C51, C53	0603	Capacitor	GRM188R60J475ME19D	DNI	7	4.7uF, 4u7/6V3
C20, C54, C56, C452	0603	Capacitor	AMK107BJ106MA-T		4	10uF, 10uF/4V
C21	0603	Capacitor	GRM188R71H103KA01J		1	10nF
C22, C42	0603	Capacitor	C1608X7R1H223K080AA	DNI	2	22nF 50V X7R 10%
C24, C40	0603	Capacitor	CL10A226MQ8NRNE		2	22uF
C27, C28	0603	Capacitor	C1608X7R1A105K080AC		2	1uF 25V X5R 10%
C41, C44	0603	Capacitor	GRM188R61E105KA12D		2	1uF
C73	0402	Capacitor	LMK105BJ104KV-F		1	100nF
D1, EXT, USB	Green 0805, LTST-C171GKT	Everlight LED, LED GREEN CLEAR THIN 0805 SMD	LTST-C171GKT		3	2.1V; 20mA; 12mcd
D2, D3	DIODE ZENER SOD323		MM3Z6V2ST1G		2	MM3Z6V2ST1G
DM1, DM2, DM11, DM12, DM13, DM14	Jumper Shunt	Jumper Shunt	M50-2000005		6	BLACK 1.27MM
DM3	Mating Part	CONN PLUG HOUSING EP .156 2POS	DF1B-2S-2.5R		1	
DM4, DM5	Contact	Contact 18-22AWG	DF1B-2428SCF		2	
DM6	Mating Part	CONN RECEPT 2POS 24AWG MTA100	440129-2		1	
DM7, DM8	Contact	Contact 18-22AWG	1735801-1		2	
DM10	Jumper Shunt	Jumper Shunt	881545-2		1	1-881545-2
J5	DX4R205JJA	Molex_5000751517_CON_USB_5POS_VERTICAL	DX4R005JJ2R1800		1	
J8	Coax		5-1814832-1	DNI	1	
J9, J200, J201, J400, J401, J402, J403, P7	2 PIN HEADER	Header_1X9	GRPB021VWVN-RC		8	
L1	0603	Inductor	ELJ-RE10NJFA		1	10nH
L2	0603	Inductor	HK160882NJ-T		1	82nH
L3, L4	0603	Inductor	ELJ-RE56NJFA		2	56nH

Designator	Comment	Description	Manufacturer Part Number	Place	Quantity	Value
L5	0603	Inductor	LQP18MN3N3C02D		1	3.3nH +/-0.2nH 250mA 400mOhm
L6, L9	0603	Inductor	ELJ-RE12NGFA	DNI	2	12nH
L8	0603	Inductor	LQP03TN6N8J02D		1	6.8nH +/-5% 300mA 600mOhm
P1, P2, P6	Header 13X2	Header, 13-Pin, Dual row	PRPC013DAAN-RC	DNI	3	
P3	LHI968		LHI968		1	
P4	Header_1X4	Header_1X4	M20-9990445		1	
P5	JACK	CONN HEADER .156 VERT 2POS TIN	DF1BZ-2P-2.5DS		1	7-16V
P10	2 PIN HEADER	Header, 2-Pin	440055-2		1	Ext Battery 3V
Q1, Q2	MOSFET-P	P-Channel MOSFET	DMP3099L-7		2	
Q3, Q4	MOSFET-P	P-Channel MOSFET	FDN340P		2	
R1, R3, R6, R10, R13, R15	0603	Resistor	RC0603JR-070RL	DNI	6	0
R2, R29, R34	0603	Resistor	RC1608J473CS		3	47K
R4	0603	Resistor	RC1608J562CS		1	5.6K
R5, R7	0603	Resistor	RC1608J105CS		2	1M
R8, R17	0603	Resistor	RC1608J103CS		2	10K
R9, R11	0402	Resistor	ERJ-2GEJ221X		2	220R
R12	0603	Resistor	RC1608J220CS		1	22
R14	0603	Resistor	ERJ-3GEYJ104V		1	100K/5%
R16	0603	Resistor	RC1608J821CS		1	820
R18	0603	Resistor	ERJ-3GEYJ221V		1	220R
R19, R36	0603	Resistor	RMCF0603JT330R		2	330
R20	0603	Resistor	RC0603FR-07340KL		1	340K 1%
R23	0603	Resistor	ERJ-3EKF2702V		1	27K 1%
R30, R31, R35	0603	Resistor	RC1608J222CS		3	2.2K
R32	0603	Resistor	RC0603FR-07332KL		1	332K 1%
R33	0603	Resistor	ERJ-3EKF1963V		1	196K 1%
R70	0603	Resistor	ERJ-3GEYJ103V		1	10k
SW1	Switch Slide DPDT .2A 12V	Switch Slide 4PDT .2A 12V	MHS222		1	
SW2	SWITCH TACTILE SPST	Single-Pole, Single-Throw Switch	PTS645SH50SMTR92 LFS		1	0.05A 12V 200gf
TPL1	5V	TEST POINT PC COMPACT T/H GREEN	5121		1	GREEN
TPL2	VSUP	TEST POINT PC COMPACT .063"D RED	5005		1	RED
TPL3	12V	TEST POINT PC COMPACT .063"D BLK	5124		1	Purple

Designator	Comment	Description	Manufacturer Part Number	Place	Quantity	Value
TPL4, TPL6, TPL50, TPL51, TPL52, TPL53, TPL54	GND	TEST POINT PC COMPACT .063"D BLK	5006		7	BLACK
TPL5	3V	TEST POINT PC COMPACT .063"D BLK	5122		1	Blue
U1	Herzog 5x5 QFN	QFN 32 Pin 5X5	Herzog 5x5 QFN		1	
U2	AP2125N-3.3TRG1	LDO Regulator (3.3V, +/-2%, 300mA, CMOS)	AP2125N-3.3TRG1		1	
U3	ZLDO1117G50TA	SOT223	ZLDO1117G50TA		1	
U4, U6	IC	SOT23-5	TPS3803-01DCKR		2	TPS3803-01DCKR
U5	TSSOP 8LD	IC OR CTRLR SRC SELECT 8TSSOP	TPS2113APWR		1	TPS2113APWR
U100	STM32F072C8	UFQFPN48	STM32F072C8U6		1	
X1	32KHz	SMT 32kHz quartz crystal 20ppm 12.5pF 2.0 x1.2mm	ABS06-32.768kHz-T		1	
X2	HC49SM SMD	Crystal frequency reference	CSM1Z-A2B2C3-40-30.0D18		1	30MHz crystal

4.0 RUNNING DEMO

4.1 HERZOG EVKIT BOARD CONFIGURATION

4.1.1 Connectors, Jumpers and Switches

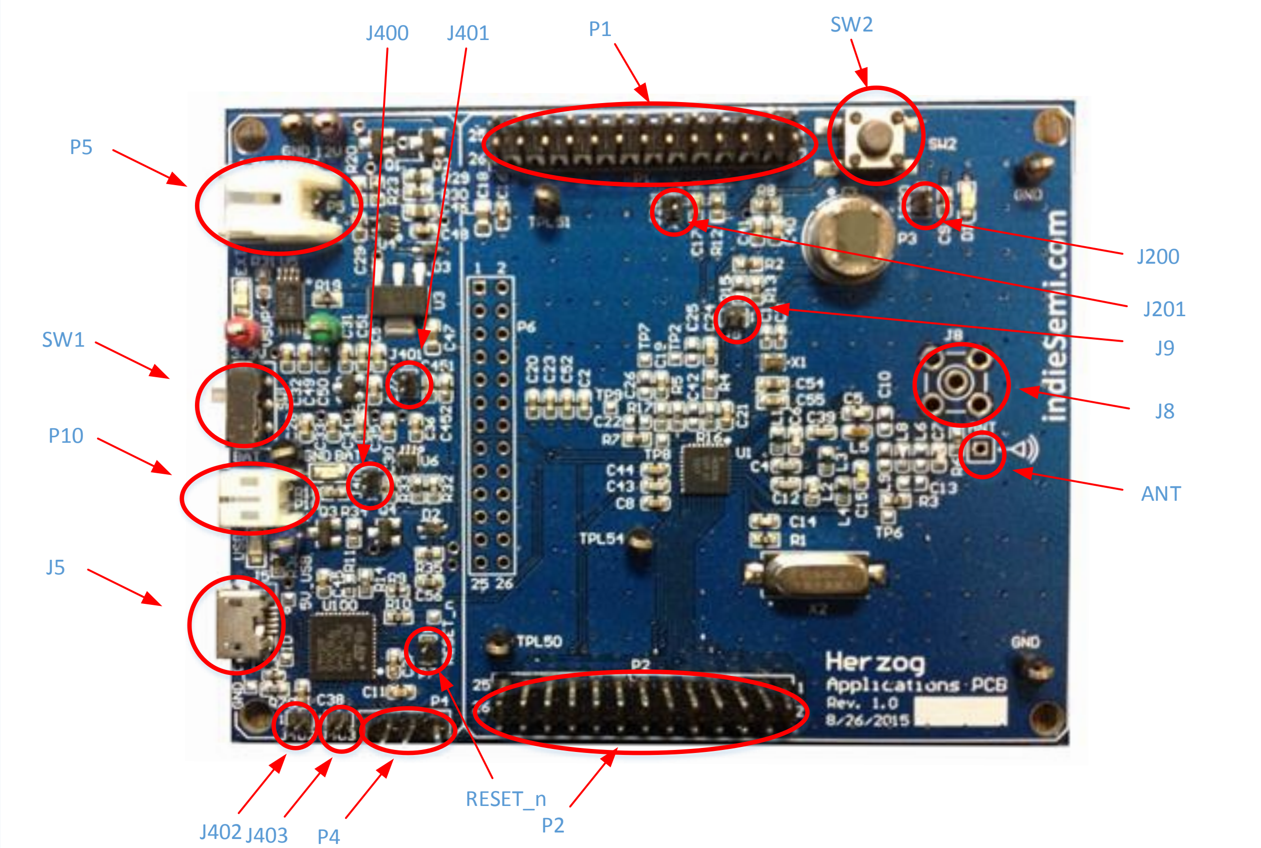


Figure 6 Herzog EVKit Connectors, Jumpers and Switches

Table 1: Connectors, Jumper and Switch Configurations		
Designator	**Description**	**Setting**
P5	12V Power Input. Reverse Protected. Overvoltage protection is not functioning with current revision.	
SW1	Herzog Power Source Selection	Upper Position (3.3V): on-board 3.3V LDO Lower Position (BAT): 3.0V battery from P10

P10	3V Battery Input. Reverse Battery Protected. Overvoltage shutoff at 3.6V.	
Designator	**Description**	**Setting**
J5	Micro B USB connector for JLINK-OB. Virtual COM (VCOM) and alternative power source to the entire unit.	
J402	VCOM RXD connection	ON: Enable VCOM RXD, Herzog TXD (PE5) is not available for application. OFF: Disable VCOM RXD, Herzog TXD (PE5) is available for application.
J403	VCOM TXD connection	ON: Enable VCOM TXD, Herzog RXD (PE4) is not available for application. OFF: Disable VCOM TXD, Herzog RXD (PE4) is available for application.
P4	JLINK-OB JLINK – Reserved for factory	
P7	JLINK-OB Reset – R Reserved for factory	
P2	Herzog Extension Connector. All pins are connected to Herzog pin directly. Please ensure proper protection (ESD, short, and Voltage levels) when used in applications.	
ANT	433 MHz Transceiver antenna. User can solder a short rigid wire to be the antenna. For best performance, matching network should be tuned. (L9, C10, L8, L6, C13, C7).	
J8	433 MHz Transceiver external antenna connection. For best performance, matching network should be tuned. (L9, C10, L8, L6, C13, C7). R6 should be populated.	
J9	PIR biasing controlled by firmware. If this feature is required, R15 should be populated and R13 should be de-populated.	ON: Biasing of PIR sensor P3 is controlled by PD2. When PD2 is logic high, PIR biasing will be turned off to reduce standby current. PIR is turned off. When PD2 is logic low, PIR biasing will be enabled (set by R2). PIR is ready for detection. OFF: Biasing of PIR sensor P3 is set by R2. (R13 must be populated).
J201	SW2 enable	ON: SW2 is connected to PD2 OFF: PD2 is available for application
J200	On board LED enable	ON: LED is connected to PD0 OFF: PD0 is available for application
SW2	Tactile Switch, connected to PD2 if J201 is on	
P1	Herzog Extenstion Connector. All pins are connected to Herzog pin directly. Please ensure proper protection (ESD, short, and Voltage levels) when used in applications.	
J401	3.3V current measurement	ON: Normal operation OFF: for attaching current meter to measure current consumption for Herzog when powered by on board 3.3V LDO

J400	3V Battery current measurement	ON: Normal operation OFF: for attaching current meter to measure current consumption for Herzog when powered by external battery

Demo Setting:

USB power, LED enabled, Tactile Switch Enabled, Virtual Com enabled

a. SW1: UP
b. J402: On
c. J403: On
d. J401: On
e. J400: On
f. J201: On
g. J200: On
h. J9: Off
i. ANT: Wire soldered

4.1.2 PC connection

Connect the Herzog EVKit to a PC with a USB Micro B cable.

4.2 HERZOG EVKIT PC COMMUNICATION

Herzog EVKit can be used with Windows. Terminal programs like HyperTerminal, TeraTerm, etc. can be used to communicate with the Herzog EVKit.

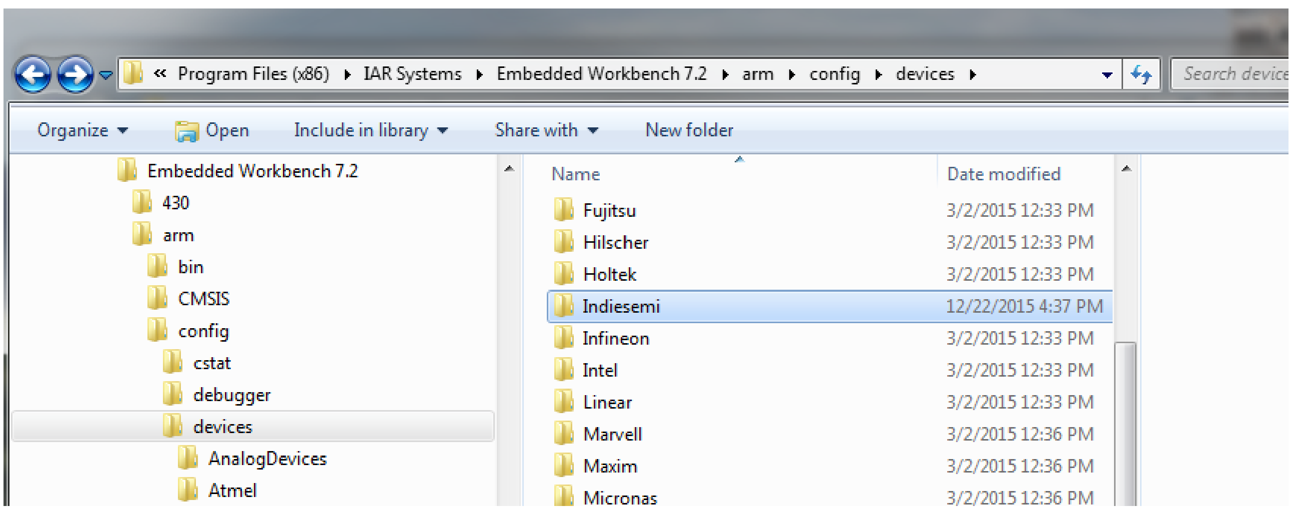
The setting is:

- Baud rate: 57600
- Data: 8 bit
- Parity: None
- Stop: 1 bit
- Flow control: None

5.0 LAUNCHING DEMO CODE (IAR): STARTING AND CONFIGURATION

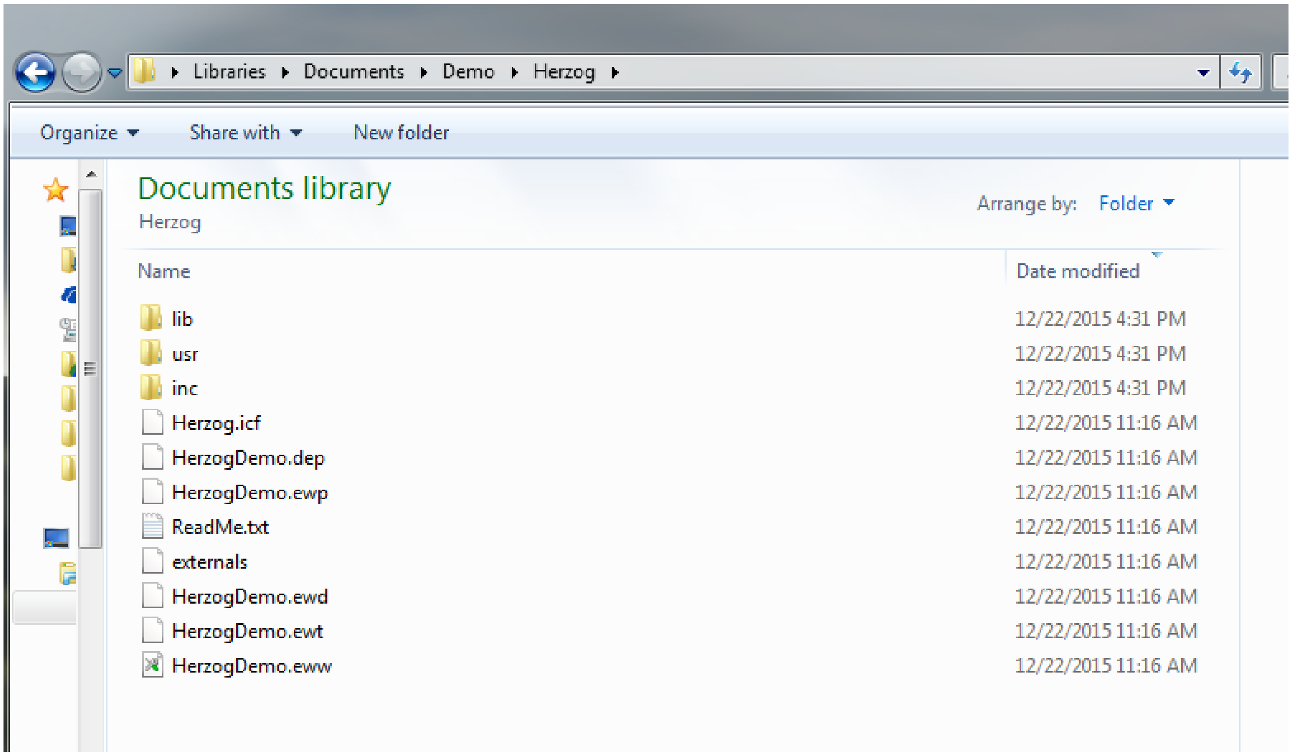
- The Herzog EVKit demo software works with IAR open (free) version. The free version allows the creation of downloadable code limited to 16kBytes. All demos can be individually selected and the compiled code fits within the 16kB limit. If more complex programs are compiled it may be necessary to license the full version of IAR.
- Before launching IAR, it is necessary to add the 2 Herzog config product files within the ARM config directory.

1) First, create a "/Indiesemi/" folder within the IAR ARM devices area:



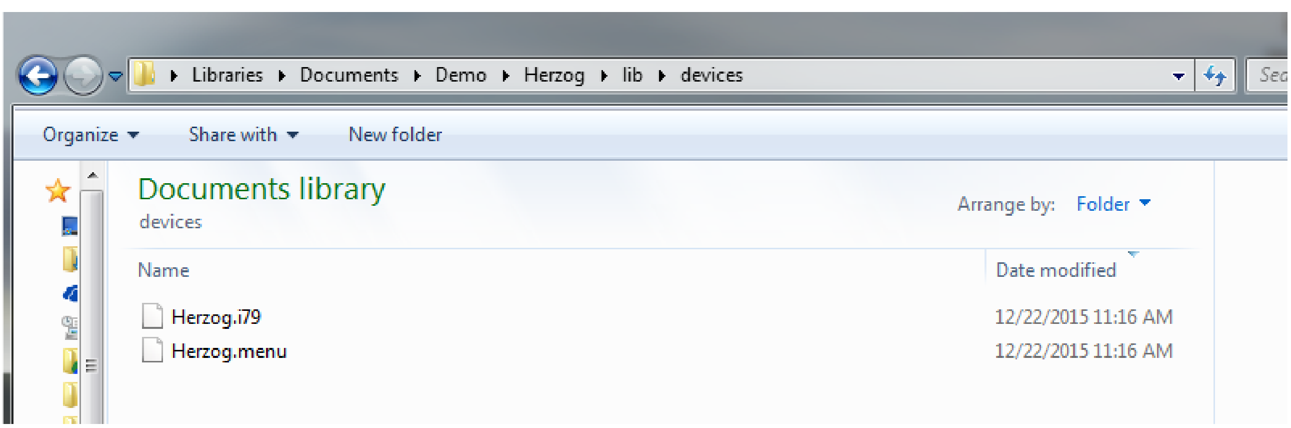
2) Copy the indie Herzog Software Development Kit into your working area:

Unzip the provided file - libdev.herzog.00.00.01.10606.zip (or similar / later revision) anywhere in your working environment.

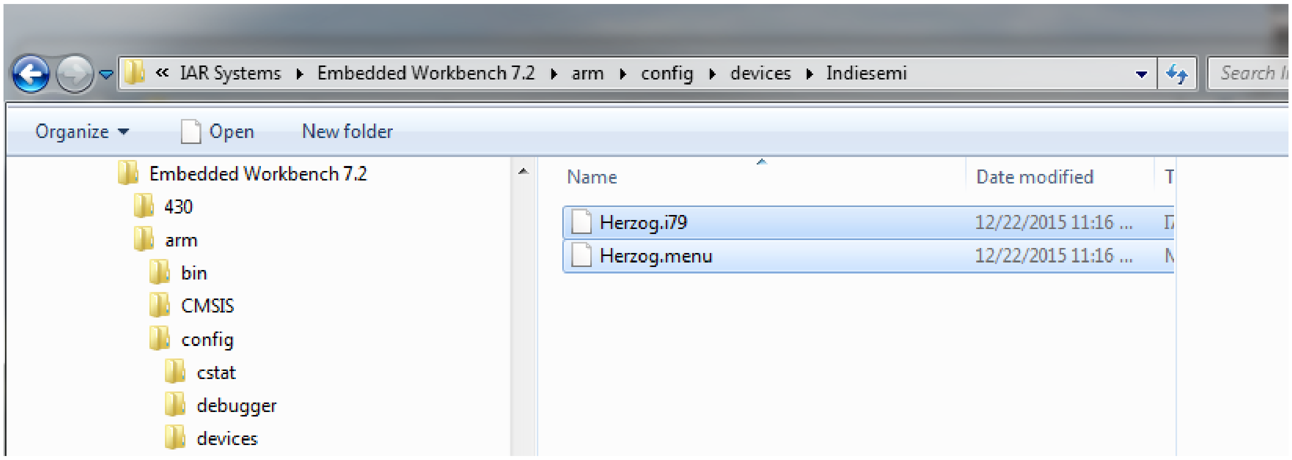


3) Locate the two Herzog config files in the indie Herzog subfolders:

The two files (Herzog.i79 and Herzog.menu) are provided within the Herzog software development kit. They are located under the following directory: \lib\devices\

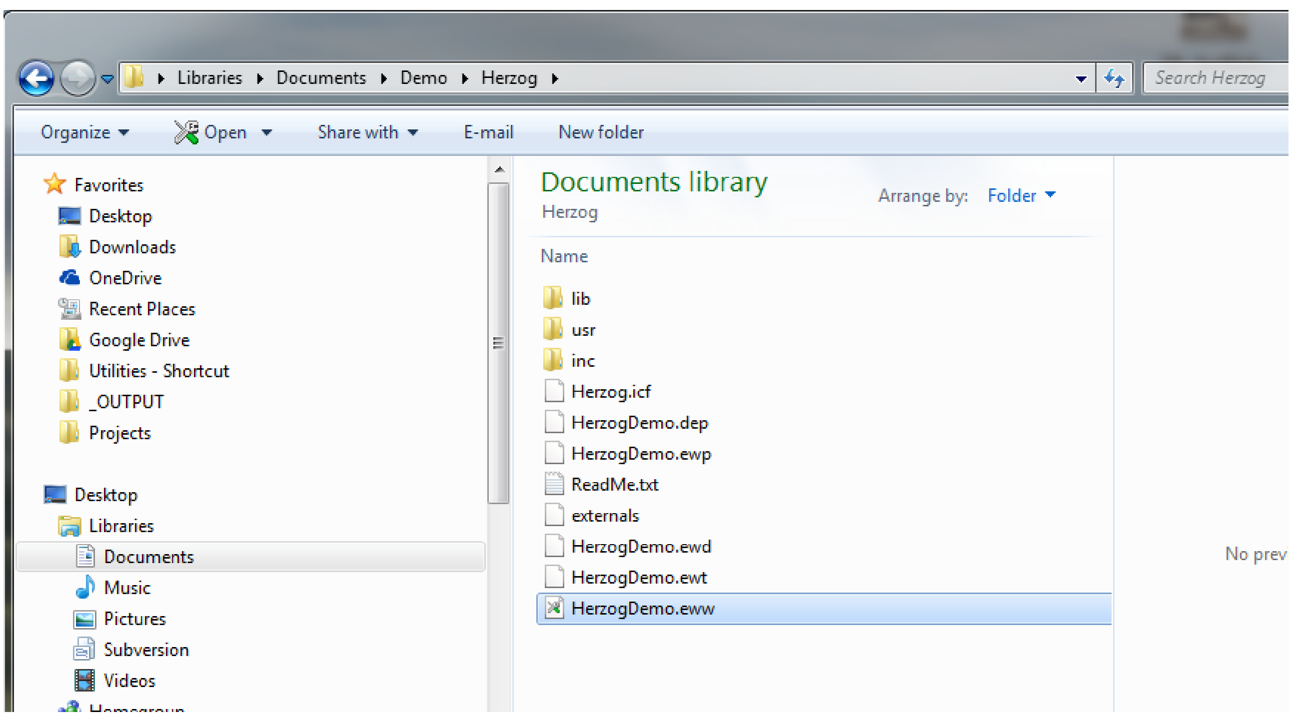


4) Copy these two files and paste them into the /indiesemi directory created in 1)

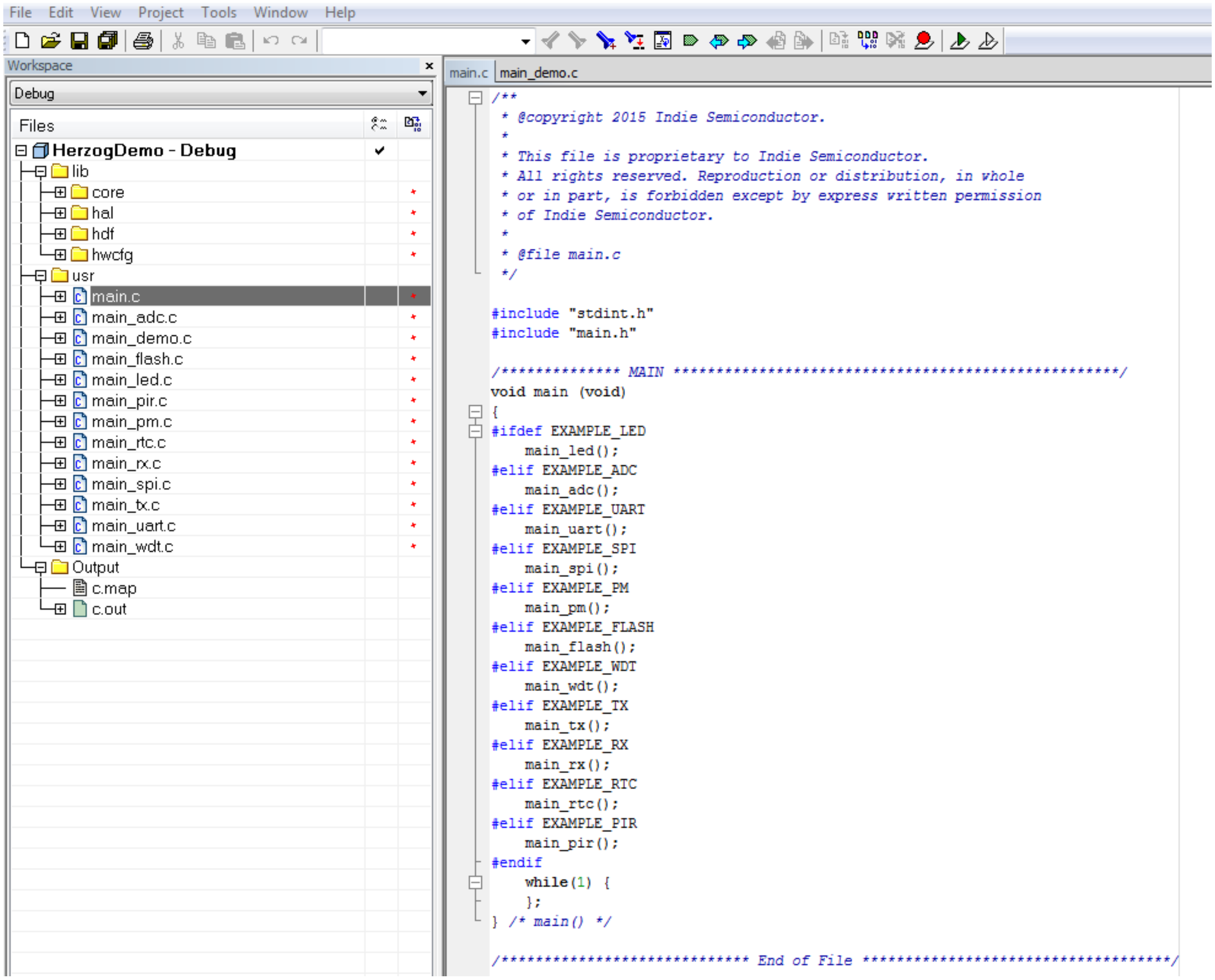


You are now ready to launch IAR and the indie Herzog demo code:

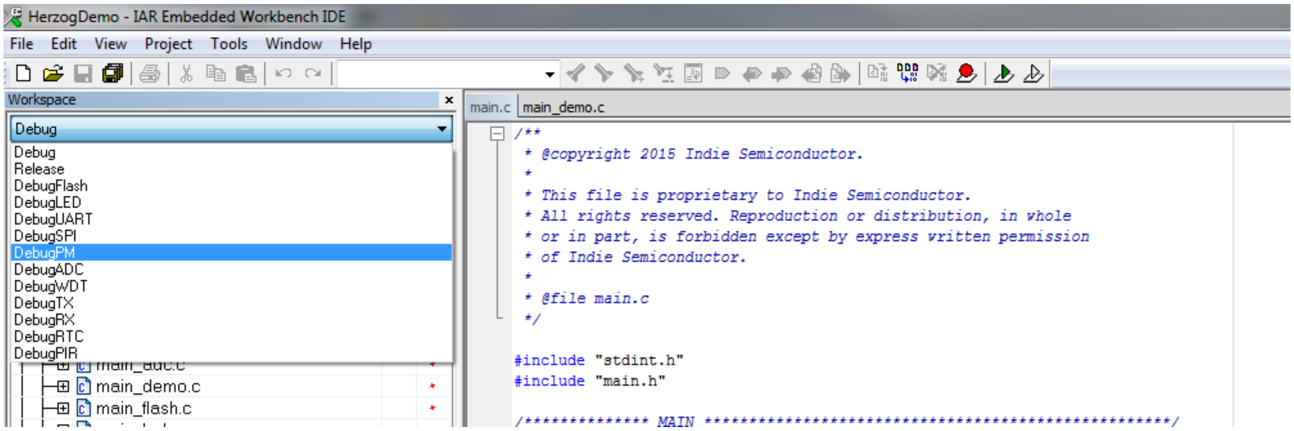
5) Go back to the indie Herzog SDK directory and launch HerzogDemo IAR project (IAR IDE Workspace)



The HerzogDemo project will open with all the options:



You can now select the type of demo you would like to upload into the Herzog EVKit:

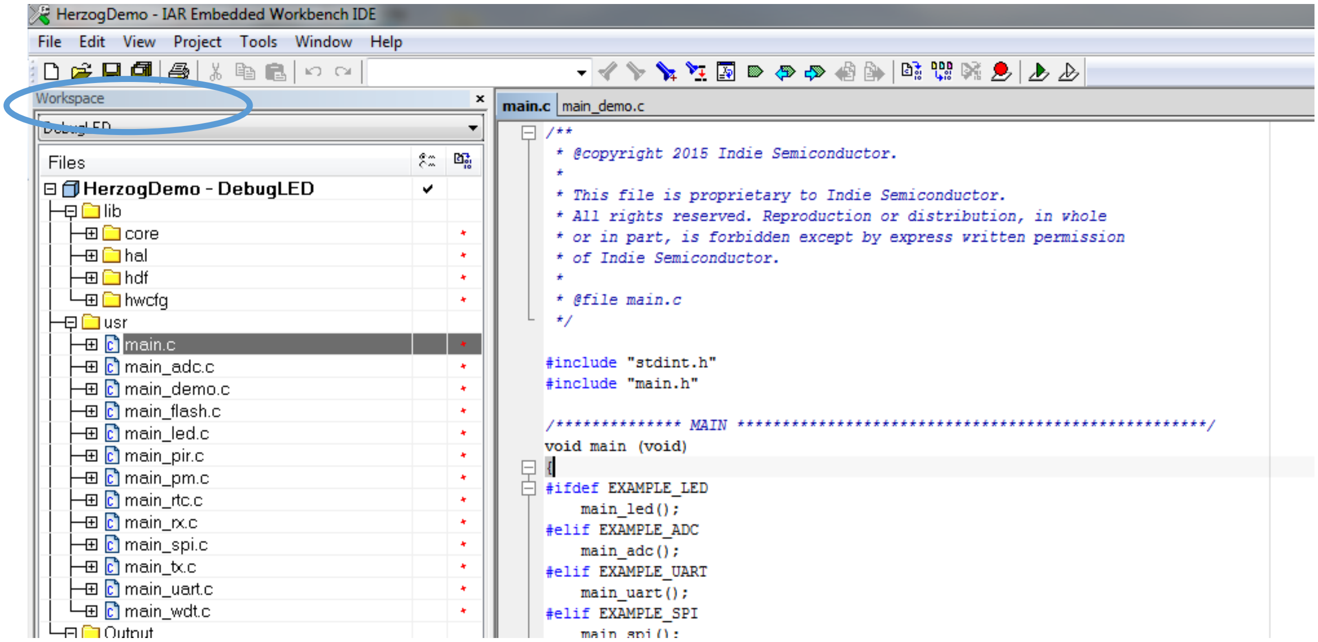


There are 13 options that can be run:

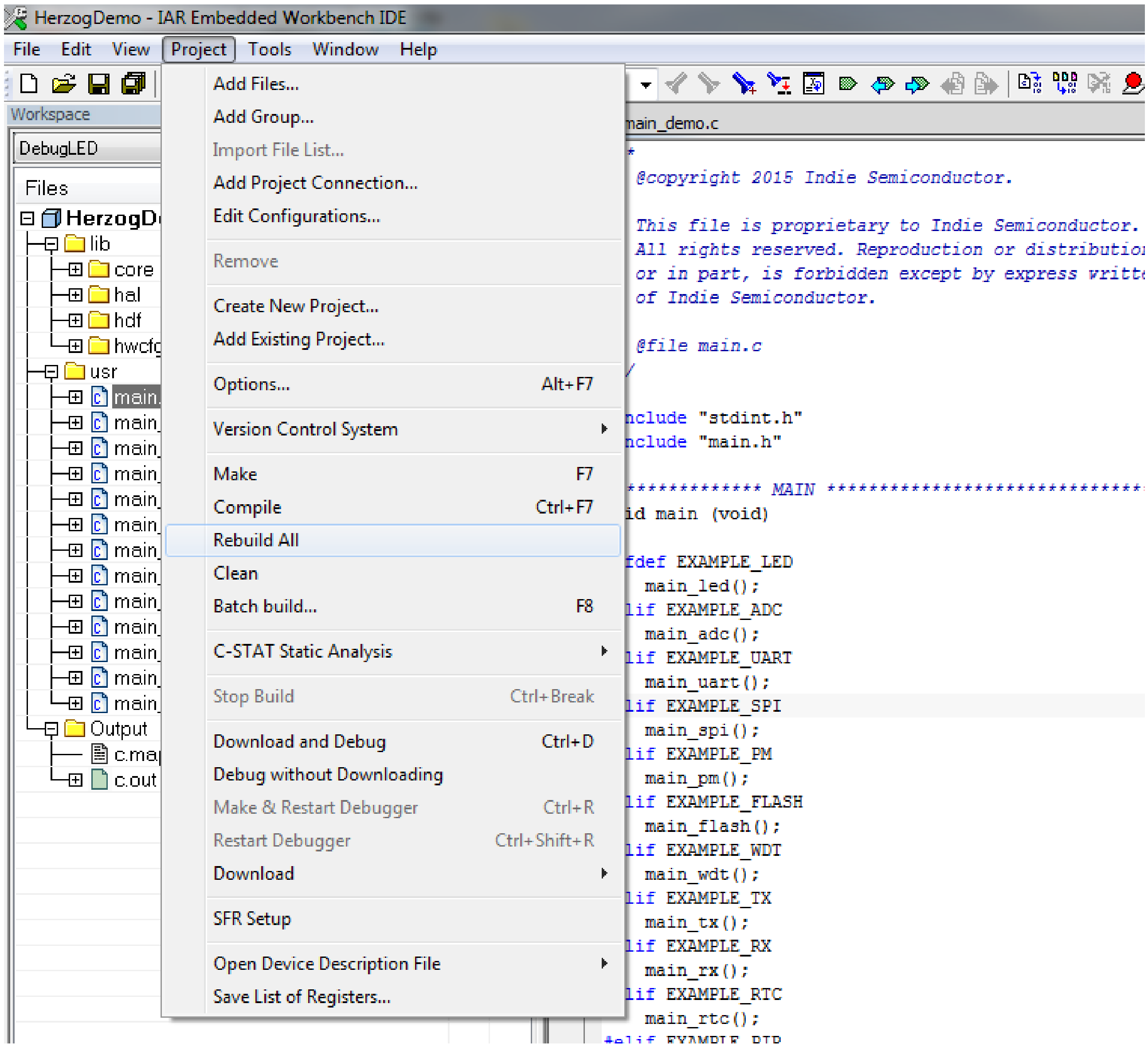
1. Debug: Allows users to write their own code in debug mode.
2. Release: Users write their own code in release mode
3. DebugFlash: Test Flash memory for user data storage.
4. DebugLED: Example code to work with Timer, GPIO interrupt and LED
5. DebugUART: Example code to use UART driver with Windows Communication Terminal
6. DebugSPI: Example code to use SPI driver with SPI/I2C tester
7. DebugPM: Test power management for sleep and wakeup.
8. DebugADC: Test ADC
9. DebugWDT: Test Watchdog
10. DebugTX: Test Radio transmit.
11. DebugRX: Test Radio receive.
12. DebugRTC: Test RTC clock.
13. DebugPIR: Test PIR intrusion detection

Select the demo example program you would like to run:

➔ Let's start with the DebugLED

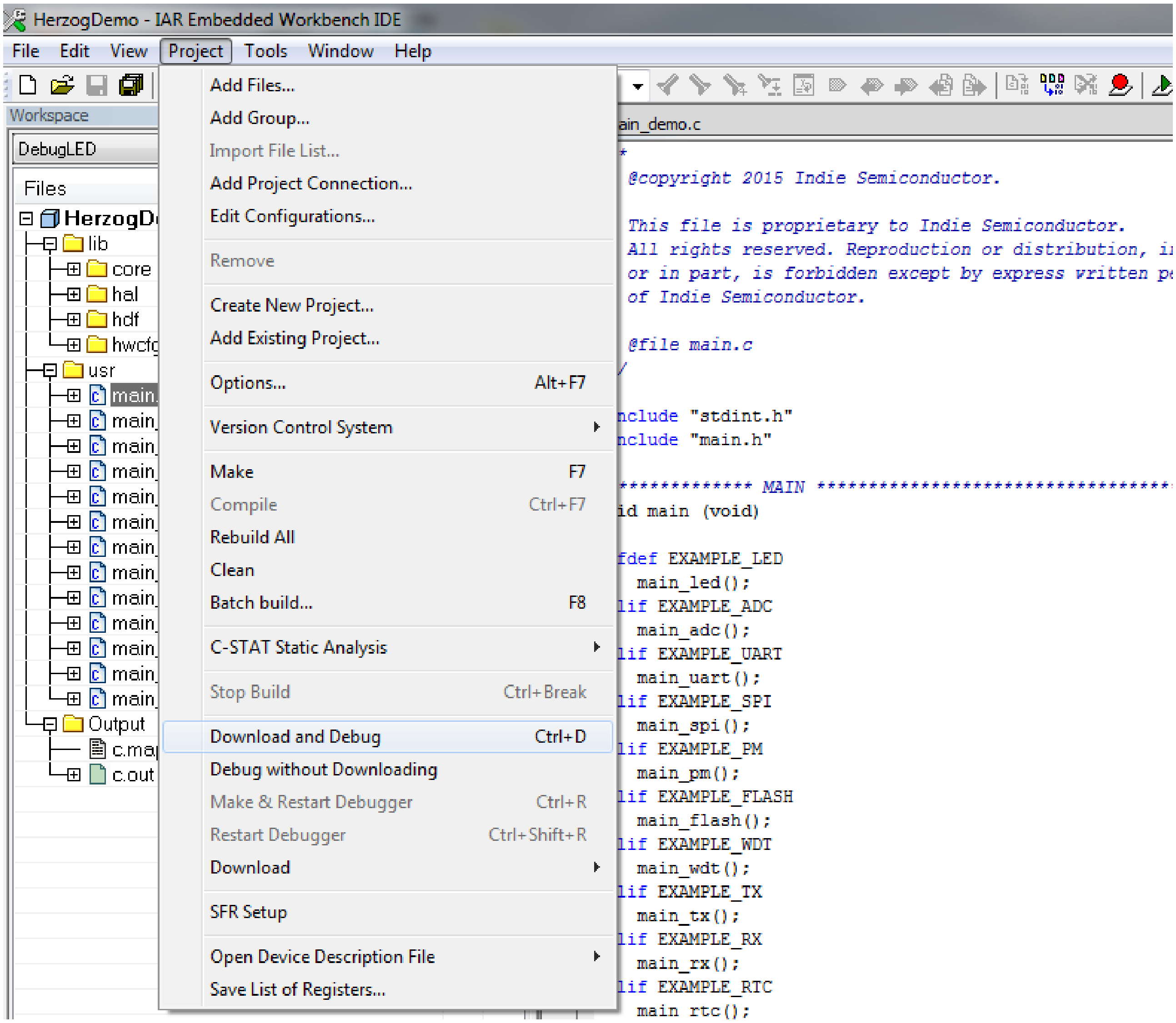


Now you can build the demo code:



There may be a list of Warnings and these can be ignored.

The code is now ready to be uploaded into the Herzog microcontroller:

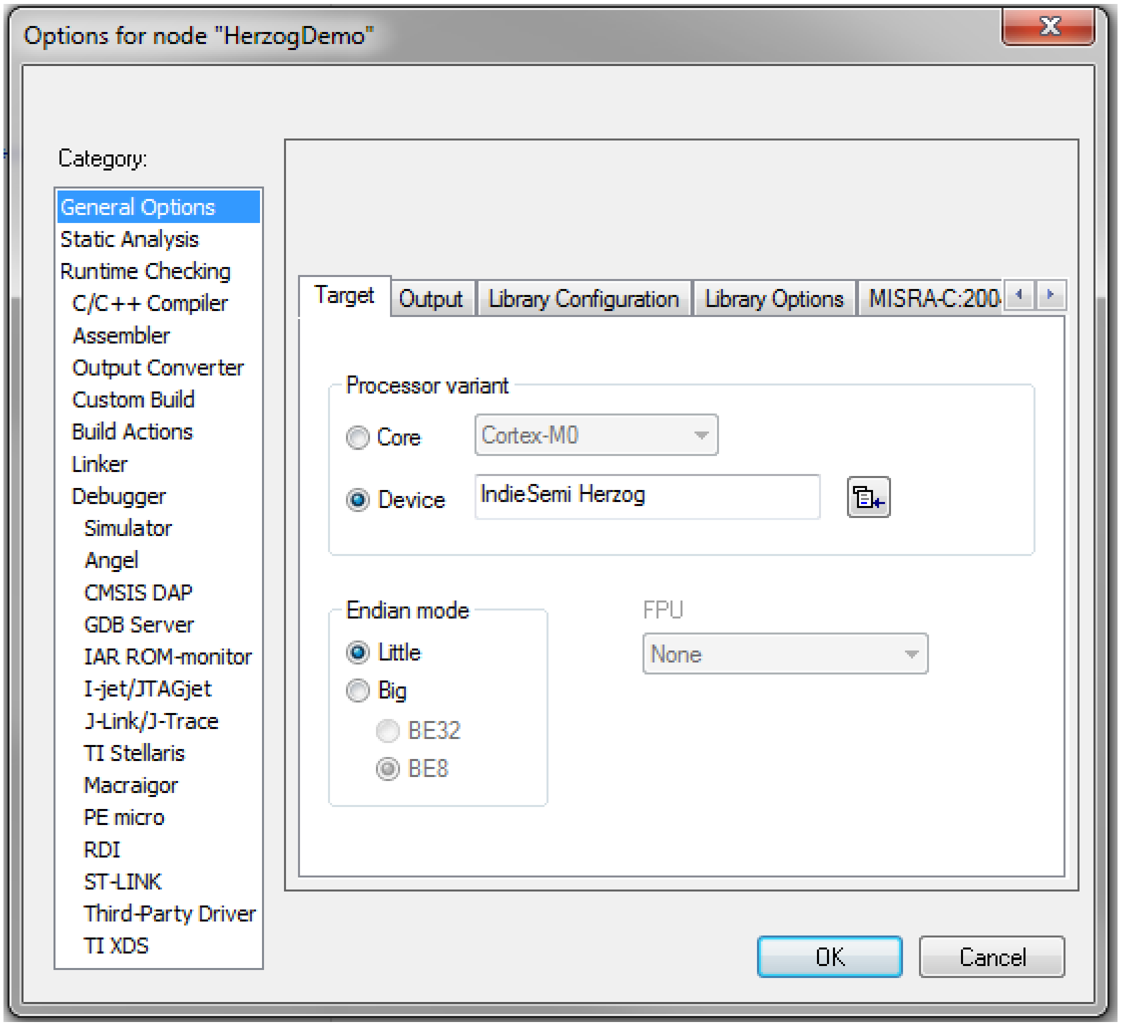


- Once the code has been uploaded into the MCU, you are now able to run it (DEBUG\GO)
- As an example, the DemoLED code demonstrates the setup of a GPIO controlling LED 2 on the Herzog EVKit. Once the button is pressed, the LED will turn on for 500ms.

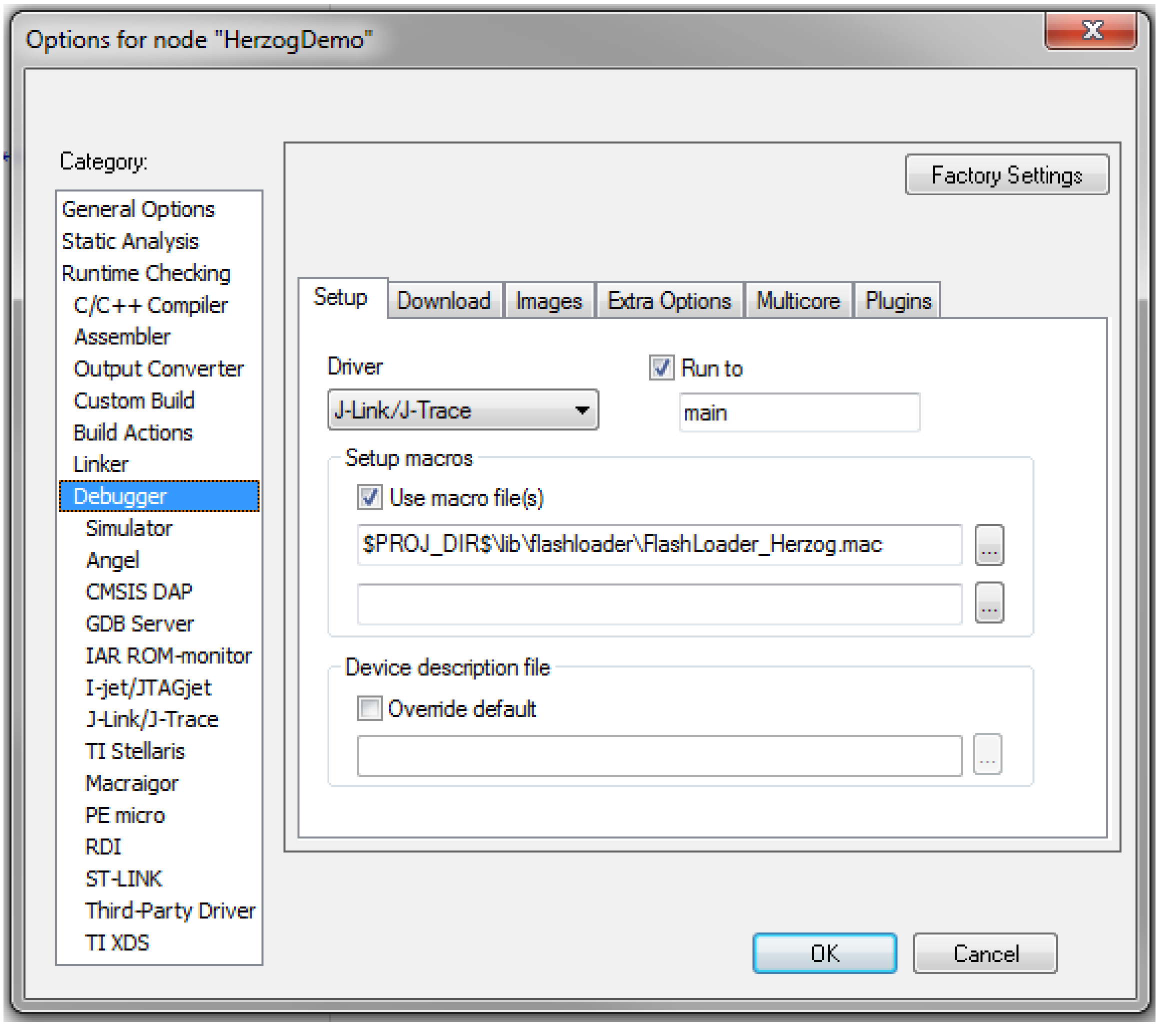
6.0 ANNEX: OVERVIEW OF IAR CONFIG

The IAR environment should be configured for Herzog demo and Flash programming:

➔ General Options within the IAR Project should indicate the indiesemi Herzog device



➔ The links to debugger and flash programmer are within the indie Herzog SDK:



You can now re-Flash the Herzog evaluation board with the code, and explore its structure.

The Herzog demo includes all the libraries needed to check all peripherals as well as all communications.

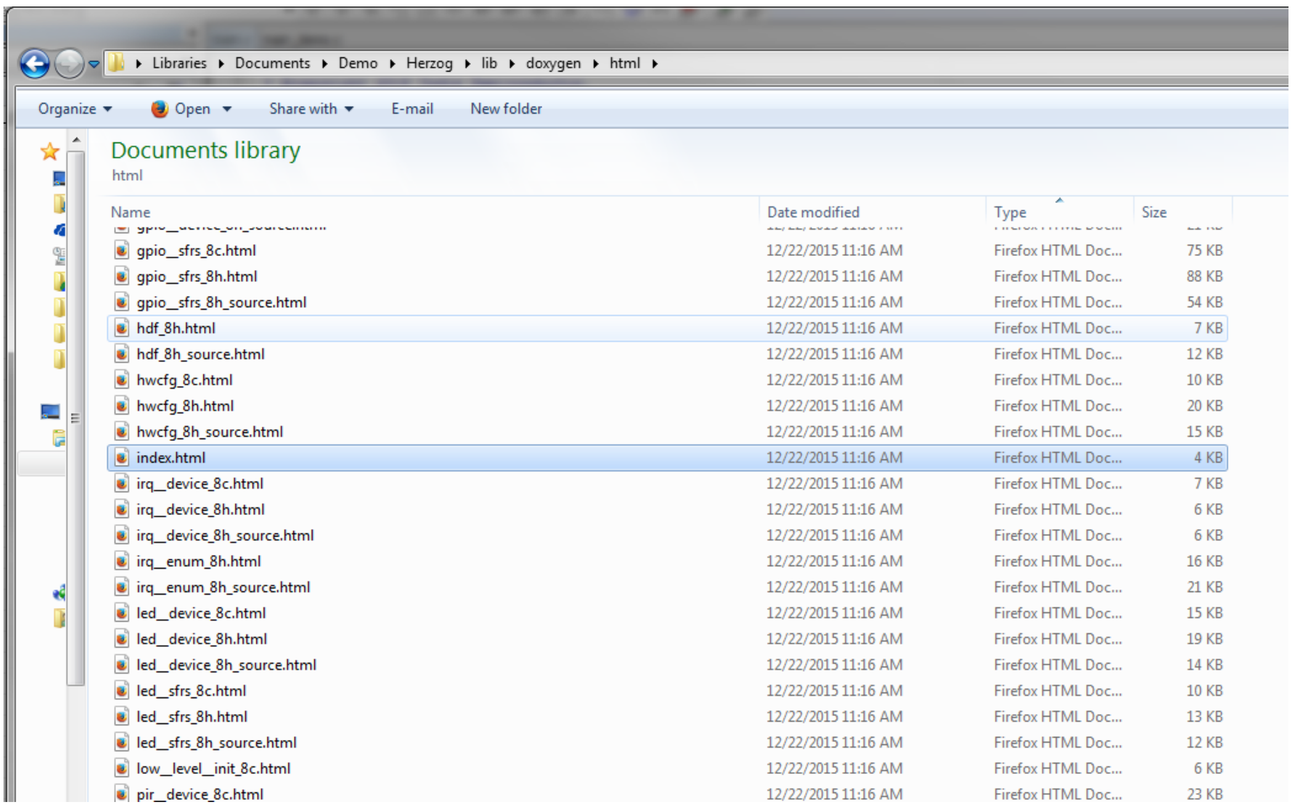
7.0 INDIE HERZOG LIBRARIES AND DOCUMENTATION

indie provides all the software to access and control the Herzog peripherals. The software architecture is built through multiple dedicated header files with specific functions (SPI, ADC, etc.). Every library has a specific set of functions to control the mentioned peripherals, configure the device, or run the demo.

There is detailed documentation included in the indie Herzog Software Development Kit, located in the

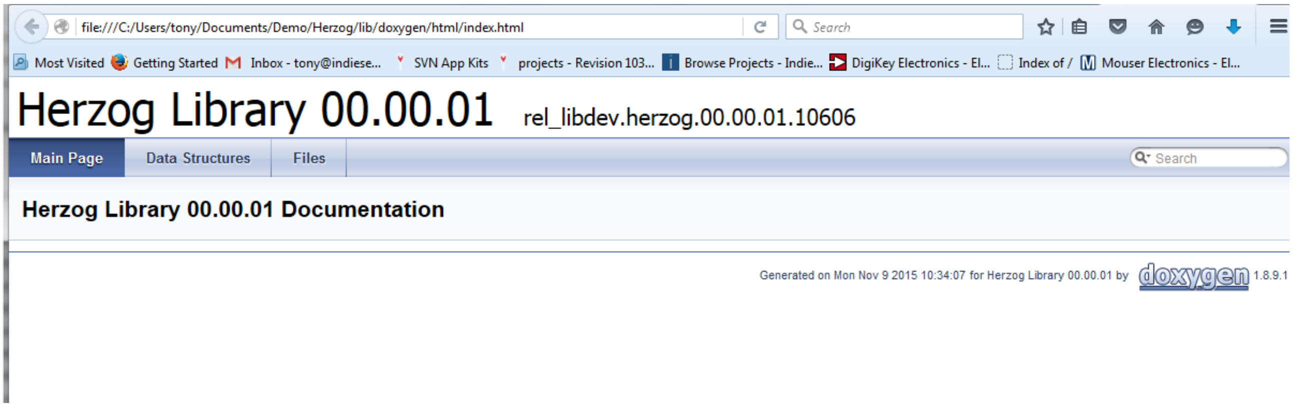
\lib\doxygen\html

To launch the documentation, double click on index.html as shown below:



Herzog library documentation is created using doxygen. Doxygen is a documentation generator, a tool for writing software reference documentation. The documentation is written within code, and is thus relatively easy to keep up to date. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code. Doxygen is free software, released under the terms of the GNU General Public License.

After clicking on index.html, the full documentation page opens:



You can navigate through the all library details and data structures being used as well as the manual for all the files and functions:

All the header files used in the SDK provided and the demo files are included in this library.

Herzog Library 00.00.01 rel_libdev.herzog.00.00.01.10606

Main Page | Data Structures | Files

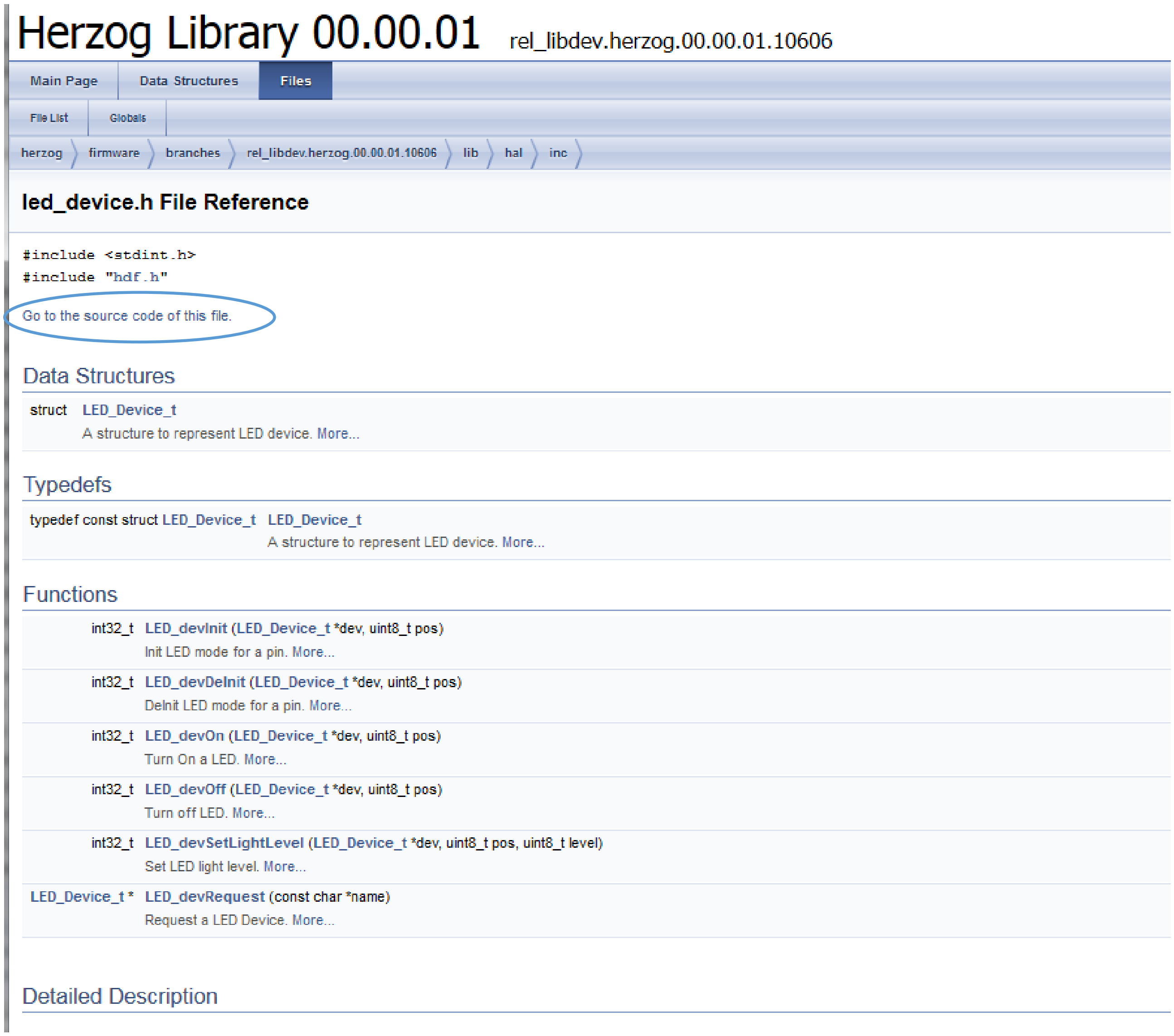
File List | Globals

File List

Here is a list of all files with brief descriptions:

- herzog
  - firmware
    - branches
      - rel_libdev.herzog.00.00.01.10606
        - lib
          - core
            - inc
              - clough_core.h
              - irq_enum.h
            - src
              - cstartup_M.c
              - low_level_init.c
          - hal
            - inc
              - adc_device.h
              - clock_device.h
              - errno.h
              - flash_device.h
              - gpio_device.h
              - irq_device.h
              - led_device.h
              - pir_device.h
              - pmu_device.h
              - rf_codec.h
              - rtc_device.h

Selecting one file will provide all details on functions included, as well as source code:



Herzog Library 00.00.01 rel_libdev.herzog.00.00.01.10606

Main Page | Data Structures | Files

File List | Globals

herzog › firmware › branches › rel_libdev.herzog.00.00.01.10606 › lib › hal › inc

led_device.h File Reference

```
#include <stdint.h>
#include "hdf.h"
```

Go to the source code of this file.

Data Structures

struct	LED_Device_t A structure to represent LED device. More...

Typedefs

typedef const struct LED_Device_t	LED_Device_t A structure to represent LED device. More...

Functions

int32_t	LED_devInit (LED_Device_t *dev, uint8_t pos) Init LED mode for a pin. More...
int32_t	LED_devDeInit (LED_Device_t *dev, uint8_t pos) DeInit LED mode for a pin. More...
int32_t	LED_devOn (LED_Device_t *dev, uint8_t pos) Turn On a LED. More...
int32_t	LED_devOff (LED_Device_t *dev, uint8_t pos) Turn off LED. More...
int32_t	LED_devSetLightLevel (LED_Device_t *dev, uint8_t pos, uint8_t level) Set LED light level. More...
LED_Device_t *	LED_devRequest (const char *name) Request a LED Device. More...

Detailed Description

Herzog Library 00.00.01 rel_libdev.herzog.00.00.01.10606

Main Page | Data Structures | Files

File List | Globals

herzog › firmware › branches › rel_libdev.herzog.00.00.01.10606 › lib › hal › inc

led_device.h

Go to the documentation of this file.

```
#ifndef __LED_DEVICE_H__
#define __LED_DEVICE_H__

#include <stdint.h>
#include "hdf.h"

typedef const struct LED_Device_t {
    void (*On)(uint8_t pos);
    void (*Off)(uint8_t pos);
    void (*SetLightLevel)(uint8_t pos, uint8_t level);
    const char *Name;
} LED_Device_t;

int32_t LED_devInit(LED_Device_t *dev, uint8_t pos);

int32_t LED_devDeInit(LED_Device_t *dev, uint8_t pos);

int32_t LED_devOn(LED_Device_t *dev, uint8_t pos);

int32_t LED_devOff(LED_Device_t *dev, uint8_t pos);

int32_t LED_devSetLightLevel(LED_Device_t *dev, uint8_t pos, uint8_t level);

LED_Device_t *LED_devRequest(const char *name);

#endif /* __LED_DEVICE_H__ */
```

Generated on Mon Nov 9 2015 10:3

8.0 REFERENCES

[1] iND80230 data sheet

9.0 REVISION HISTORY

Rev #	Date	Action	By
0.1	12/21/2015	Herzog EVKit starting guide initial draft	TL
0.2	12./23/2015	Minor corrections	PH

10.0 CONTACTS

United States
32 Journey
Aliso Viejo, California 92656, USA
Tel: +1 949-608-0854
sales@indiesemi.com

China
232 Room, Donghai Wanhao Plaza,
South Hi-tech 11th Road, Hi-tech Industry Park,
Nanshan District, Shenzhen, China.
Tel: +86 755-86116939

Scotland
5th Floor, The Auction House
63a George Street
Edinburgh EH2 2JG, Scotland
Tel: +44 131 718 6378

http://www.indiesemi.com/

Important Notice

indie semiconductor reserves the right to make changes, corrections, enhancements, modifications, and improvements to indie semiconductor products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on indie semiconductor products before placing orders. indie semiconductor products are sold pursuant to indie semiconductor's terms and conditions of sale in place at the time of order acknowledgement. Purchasers are solely responsible for the choice, selection, and use of indie semiconductor products and services described herein. indie semiconductor assumes no liability for the choice, selection, application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by indie semiconductor by this document.

The materials, products and information are provided "as is" without warranty of any kind, whether express, implied, statutory, or otherwise, including fitness for a particular purpose or use, merchantability, performance, quality or non-infringement of any intellectual property right. Indie semiconductor does not warrant the accuracy or completeness of the information, text, graphics or other items contained herein. indie semiconductor shall not be liable for any damages, including but not limited to any special, indirect, incidental, statutory, or consequential damages, including without limitation, lost of revenues or lost profits that may result from the use of the materials or information , whether or not the recipient of material has been advised of the possibility of such damage.

Unless expressly approved in writing by two authorized indie semiconductor representatives, indie semiconductor products are not designed, intended, warranted, or authorized for use as components in military, space, or aircraft, in systems intended to support or sustain life, or for any other application in which the failure or malfunction of the indie semiconductor product may result in personal injury, death, or severe property or environmental damage.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.