Radiation Hardened On-Board Computer – User's Manual

RH-OBC-1

Contents

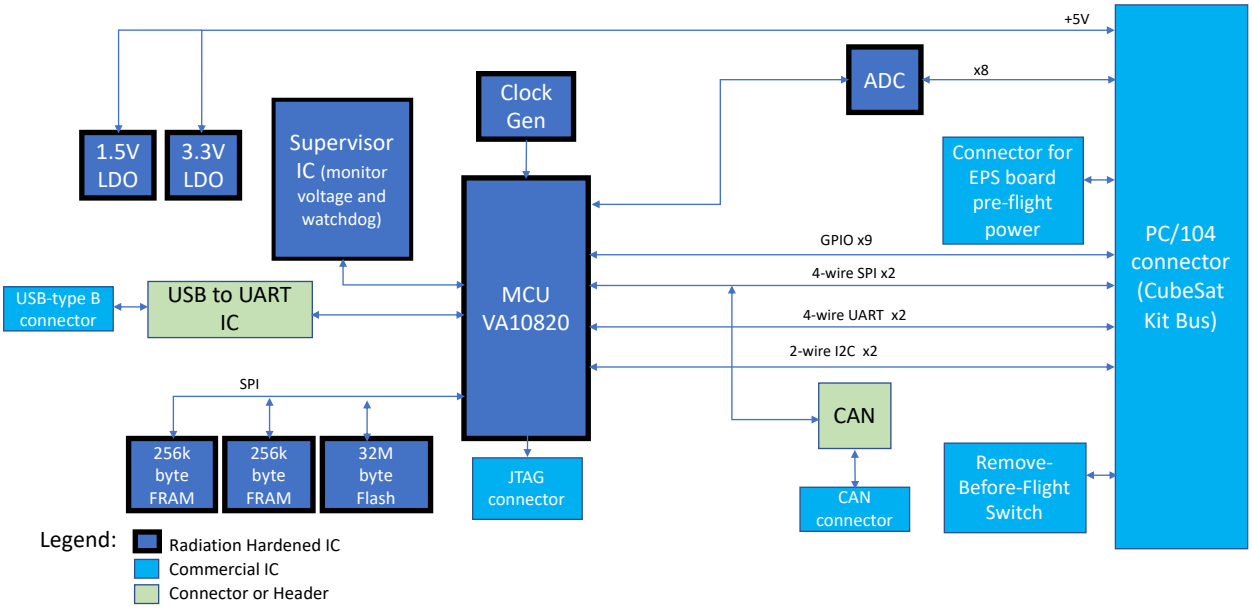
1. Introduction

The RH-OBC-1 is pin compatible with the Pumpkin Inc. CubeSat bus standard and provides a radiation hardened single board computer with a VA10820 MCU as its processing unit. The VA10820 is an ARM® Cortex®-M0 based MCU with 128k bytes of on-chip SRAM program space with a wide variety of serial peripheral modules and twenty-four 32-bit timers.

A block diagram of the kit and board is shown in Figure 1. The intent of the board is twofold: 1) provide a proven reference design for a radiation-hardened CubeSat control module and 2) provide a form-factor correct module so customers can deploy the board in a CubeSat assembly. Added benefits include an open source firmware development platform for the MCU.

A board support package (BSP) accompanies the board. The BSP and schematics can be obtained from voragotech.com. See the software requirements section of this document for details.

Figure 1 - Block diagram of RH-OBC-1



1.1. Key Features of RH-OBC-1

a. MCU direct connections to CubeSat Kit bus
   i. Two 4-wire SPI

   ii. Two I2C
   iii. Two 4-wire UART with RTS/CTS
   iv. Nine dedicated GPIO (Unused SPI and UART signals can be configured as GPIO)
b. 14-bit ADC with 11 channels (RHD5950)
   i. 8 channels tied the CubeSat Kit bus
   ii. 3 channels monitoring on-board voltages
c. 50 MHz clock source for MCU
d. 2 x 256k byte SPI FRAM for data storage
   i. One CYRS15B102 to boot from and one FM25V20A for data storage
e. 1 x 256M bit (32 Mbyte) SPI flash (CYRS16B256) for data storage
f. CAN – (HI-3110) (CAN controller + PHY with SPI interface)
   i. Separate connector for CAN interface (CAN_H, CAN_L, GND)
g. MCU power supplies from linear regulators:
   i. 3.3 V (200 mA max)
   ii. 1.5 V (100 mA max)
h. Independent watchdog chip. (ISL706)
i. 10-pin 0.05" center standard JTAG connector
j. PC/104™ connector adhering to CubeSat Kit bus pin assignments
k. USB to UART interface (FT232RL)
   i. Type B USB connector

1.2. Purpose of Document

This document is intended to provide instructions on how to use all features of the RH-OBC-1 board and provide a working platform for firmware development with the VA10800 from VORAGO. Appendix A has instructions on how to download the Keil IDE, open a project, download code and program firmware onto the board.

Separate documents are available for to help with other aspects of the design:

- Firmware User Manual (RH-OBC-1_Firmware_UM.pdf)
- Schematics and layout drawing (RH-OBC-1_revb_schematic.pdf)

1.3. Overview of firmware and BSP

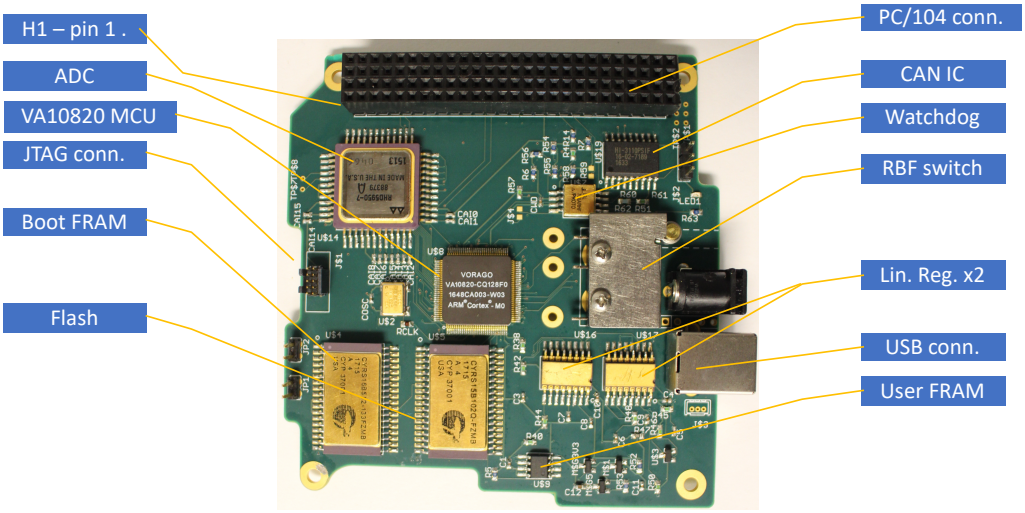
The board will come preprogrammed with demonstration firmware that conducts specific actions such as toggling a port pin or performing a data acquisition experiment when commanded to do so by the PC.

Source code for the MCU firmware is provided at voragotech.com, "RH-OBC-1_BSP.7z". A separate version of the firmware comes with support for FreeRTOS. "RH-OBC-1_frtos_BSP.7z"

1.4. RH-OBC-1 board component placement diagram

Figure 2 shows the board with various components labeled. The board schematic and layout are part of the download package and reference designators can be used to identify other components.

Figure 2 – Diagram of RH-OBC-1reference design with functional blocks identified



1.5. Materials List of Kit

- RH-OBC-1 board – Programmed with demonstration firmware
- Quick Start Guide
- USB cable (Type A male to Type B male)
- Important Notice sheet & xtra components: Schottky diodes, 0-ohm resistors, 5-pin header.
- USB Flash stick with documentation and firmware source code

1.6. Documentation accompanying kit (All included in RH-OBC-1_docs.7z)

- RH-OBC-1 Schematic
- RH-OBC-1 User Manual
- RH-OBC-1 Bill of Materials
- RH-OBC-1 Firmware User Manual

1.7. Support

For support, please contact VORAGO at the below email address. Reference RH-OBC-1in the email title.

Email: info@voragotech.com

2. Connector pin assignment tables

To assist with quickly finding which pins are tied to the various connectors on the board, the following set of tables are provided. Refer to the schematic (included in "RH-OBC-_docs.7z" file) for IC connection details.

2.1 PC/104 connector

The PC/104 connector on the board is made of headers H1 and H2. Both of these are 2 x 26 headers as shown in Figure 3. Tables 1 and 2 show all pin definitions.

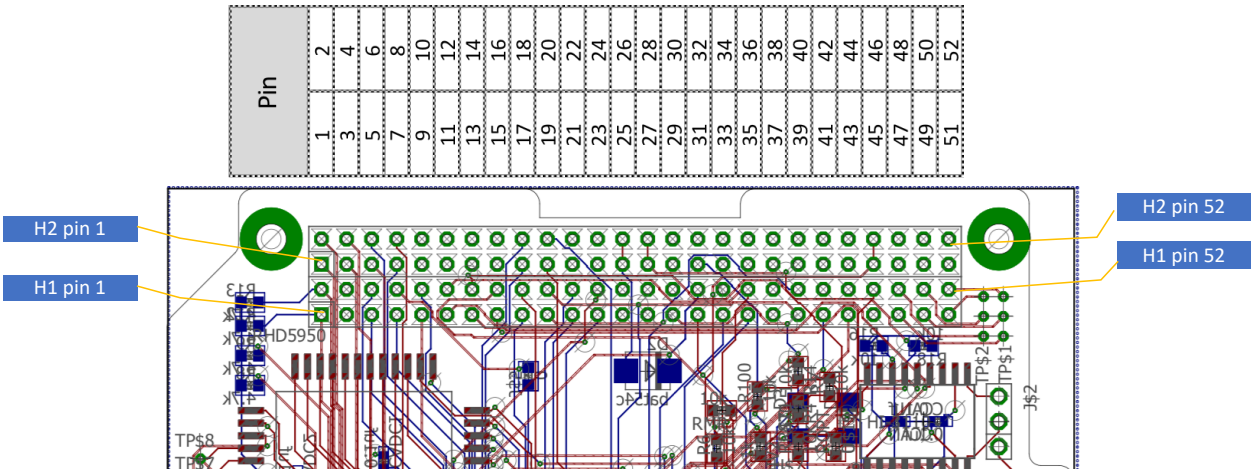


Figure 3 - Diagram showing PC/104 connector orientation

Table 1 – RH-OBC-1 H1 connector pin assignments

H1 Half of PC/104 connector (2x26)							
RH-OBC-1 function	VA10820 Port pin	CubeSat name	Pin	Pin	CubeSat name	VA10820 Port pin	RH-OBC-1 function
I2C1_SDA	SDA	IO.23	1	2	IO.22	SCL	I2C1_SCL
NC	--	IO.21	3	4	IO.20	--	NC
GPIO	PORTA[17]	IO.19	5	6	IO.18	PORTA[16]	GPIO
GPIO / TIM	PORTA[14]	IO.17	7	8	IO.16	PORTA[21]**	GPIO
UART1_CTSn	PORTB[16]	IO.15	9	10	IO.14	PORTB[17]	UART1_RTSn
UART0_CTSn	PORTB[20]*	IO.13	11	12	IO.12	PORTB[21]*	UART0_RTSn
SPI0_SCK	PORTA[31]	IO.11	13	14	IO.10	PORTA[29]	SPI0_MISO
SPI0_MOSI	PORTA[30]	IO.9	15	16	IO.8	PORTA[27]***	SPI0_SS0
UART1_RX	PORTB[18]	IO.7	17	18	IO.6	PORTB[19]	UART1_TX
UART0_RX	PORTB[22]*	IO.5	19	20	IO.4	PORTB[23]*	UART0_TX
SPI1_SCK	PORTA[20]	IO.3	21	22	IO.2	PORTA[18]	SPI1_MISO
SPI1_MOSI	PORTA[19]	IO.1	23	24	IO.0	PORTA[15]	SPI1_SS0
NC		-FAULT	25	26	VREF0	VREF+	NC
NC		SENSE	27	28	VREF1	NC	NC
EXTRESET of MCU and supervisor	EXTRESETn	-RESET	29	30	VREF2	VREF-	NC
Gate of NMOS to turn off regulators (ONn)		OFF_VCC	31	32	+5V_USB		Tied to 5V rail
GPIO / TIM	PORTB[3]	PWR_MHX	33	34	-RST_MHX		NC
UART0_CTS	PORTB[20]*	-CTS_MHX	35	36	-RTS_MHX	PORTB[21]*	UART0_RTS
NC		-DSR_MHX	37	38	-DTR_MHX	NC	NC
UART0_TX	PORTB[23]*	TXD_MHX	39	40	RXD_MHX	PORTB[22]*	UART0_RX
I2C0_SDA	I2C0_SDA	SDA_SYS	41	42	VBACKUP	NC	NC
I2C0_SCL	I2C0_SCL	SCL_SYS	43	44	RSVD0	NC	NC
NC	NC	RSVD1	45	46	PPS	PORTA[21]**	NC
Test Point	NC	USER0	47	48	USER1	NC	Test Point
Test Point	NC	USER2	49	50	USER3	NC	Test Point
Test Point	NC	USER4	51	52	USER5	NC	Test Point

Notes:

* Signal available on two different locations of PC/104

** Jumpers (J4 & J5) determine which pin is connected to MCU. PPS has a 50-ohm resistor (R62) to VSS. When board is shipped from VORAGO, both J4 and J5 are not populated.

*** IO.8 (PORTA[27]) has a 10k ohm pull-up to 3.3V.

Table 2 - RH-OBC-1 H2 connector pin assignments

H2 Half of PC/104 connector (2x26)							
RH-OBC-1 function	ADC or VA10820 Port pin	CubeSat name	Pin	Pin	CubeSat name	ADC or VA10820 Port pin	RH-OBC-1 function
ADC_CH7	ADC - AN7	IO.47	1	2	IO.46	ADC – AN6	ADC_CH6
ADC_CH5	ADC - AN5	IO.45	3	4	IO.44	ADC - AN4	ADC_CH4
ADC_CH3	ADC - AN3	IO.43	5	6	IO.42	ADC - AN2	ADC_CH2
ADC_CH1	ADC - AN1	IO.41	7	8	IO.40	ADC - AN0	ADC_CH0
NC	AN15	IO.39	9	10	IO.38	AN14	NC
NC	AN13	IO.37	11	12	IO.36	AN12	NC
NC	AN11	IO.35	13	14	IO.34	AN10	NC
NC	AN9	IO.33	15	16	IO.32	AN8	NC
GPIO/INT/TIM	PORTB[15]	IO.31	17	18	IO.30	PORTB[14]	GPIO/INT/TIM
GPIO/INT/TIM	PORTA[26]	IO.29	19	20	IO.28	PORTA[25]	GPIO/INT/TIM
NC	--	IO.27	21	22	IO.26	--	NC
NC	--	IO.25	23	24	IO.24	--	NC
Connected to +5V rail through a diode		+5V SYS	25	26	+5V SYS		Connected to +5V rail through a diode
3.3 V output of Reg		VCC_SYS	27	28	VCC_SYS		3.3 V output of Reg
VSS	Dig GND	DGND	29	30	DGND		VSS
VSS	Analog GND	AGND	31	32	DGND		VSS
NC	--	S0	33	34	S0	--	NC
NC	--	S1	35	36	S1	--	NC
NC	--	S2	37	38	S2	--	NC
NC	--	S3	39	40	S3	--	NC
NC	--	S4	41	42	S4	--	NC
NC	--	S5	43	44	S5	--	NC
ADC Input on Ch8. 22k/1k resistor divider	--	VBATT	45	46	VBATT		ADC Input on Ch8. 22k/1k resistor divider
NC	--	USER6	47	48	USER7	--	NC
NC	--	USER8	49	50	USER9	--	NC
NC	--	USER10	51	52	USER11	--	NC

2.2 Auxiliary supply for CubeSat System prior to flight

Prior to the satellite being launched, it is convenient to have an external power supply connected to the unit. This can be used to charge batteries or to power the stack.

Note: The electronics on the RH-OBC-1 are not powered directly from this connector. This connector is intended to power the entire board stack and tie into the power supply board via a separate harness.

Table 3 - CSKB-PWR connector pin assignments

Pin #	Connection
1	VSS
2	Pin 1 of barrel connector
key	
3	Pin 1 of barrel connector
4	VSS & Pin 2 of barrel connector

2.3 CAN connector

The HI-3110 physical interface pins are available on connector J$2.

Table 4 - CAN connector - J$2 pin assignments

Pin #	Connection
1	CAN_H
2	CAN_L
3	GND / VSS

3. Hardware

The RH-OBC-1 is intended to operate as the bottom board in a stack of boards connected through the PC/104 connector. Some of the hardware such as the barrel jack and the remove before flight switch are intended for system level duties and are not directly controlling power to the RH-OBC-1 electronics. In most CubeSat stacks there is a specialized power supply board with inverters for solar power supplies and batteries that will supply +5V to the board.

Before powering up the stack of boards, it is strongly advised to check all pins for potential conflicts with port pins and power supply levels.

This section contains information to supplement the schematic. It also contains information to help with the initial power-up of the board.

3.1 Power supply

The +5V supply required to run the board is provided through the PC/104 connector. (H2- pins 25 & 26 for +5V, and H2- pins 29,20 & 32 for GND)

The board is shipped configured such that: a) the USB does not power the board and b) the barrel connector does not power the board. It is possible to add a jumper wire to the board to make these connection which may simplify the bench setup during code development.

3.2 3.3V Supply

The regulator on the RH-OBC-1 is capable of supplying more current than the board requires. It is possible to provide other boards in the stack with 3.3V through .

3.3 Remove before flight switch

This switch does not directly control power on the OBC. It is intended to be used to remove power from the complete CubeSat stack. This component is not populated or supplied with the kit.

3.4 USB interface

The USB interface is intended to help with board bring-up and system integration testing. It is not based on radiation hardened silicon. One of the MCU's UART channels is routed to the FT232RL through 750-ohm resistors. If the FT232RL is not powered up, the resistors will prevent grounded Rx or Tx pins from corrupting the MCU. Likewise, if the FT232RL is powered but the MCU is not, the resistors will prevent the Rx & Tx lines from powering the MCU through port pins.

3.5 CAN interface

The HI-3110 from Holt is used for the CAN interface IC. This device is not specified to be radiation hardened. In order to correct any potential latch-up situations, the CAN IC has as switched power supply, +5VG. See page 1 of the schematic for the "PWR_PERIPH_EN_N" signal feeding transistors. PORTB[13] controls this supply. If

ever the CAN IC is not responding properly, it is advised to cycle power to the HI-3110.

3.6 ADC

An RHD5950 ADC from Aeroflex is employed. It uses the +5V supply rail as the positive reference. The board's +5V supply is derived externally and may not be precisely 5.0V. Two regulators on the board may be used as references to obtain absolute voltage results. The following equations provide the calculated voltage levels.

Vout(3.3V) = 0.6V * (1.5+6.8) / 1.5 = 3.32 V

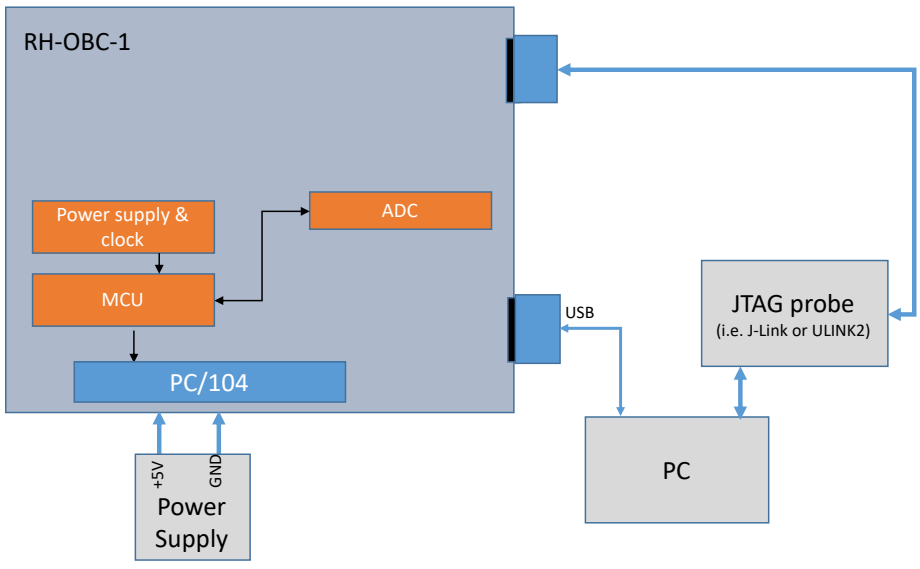
Vout(1.5V) = 0.6V * (2.4 + 3.6) / 2.4 = 1.50 V

Software drivers for making conversions is part of the BSP that can be downloaded.

3.7 Powering up the board

Prior to developing code for the OBC, it is recommended that the communication channels be checked using the minimum connection setup as shown in **Figure 4 - Minimum connections to test hardware**.

Figure 4 - Minimum connections to test hardware



Follow these steps to ensure all hardware connections have been made properly.

1) Connect the USB cable between PC and the RH-OBC-1diagnostic interface board. The PC should enumerate with the FTDI chip. If it does not, please go to the FTDI website (http://www.ftdichip.com/FTDrivers.htm) and download the Windows drivers for FT232R.
2) Start a terminal window (PuTTY is available from http://www.putty.org or Tera Term is available from http://download.cnet.com/Tera-Term/3000-2094_4-75766675.html), select baud rate = 115.2k. Choose the Communication Port. (To determine the COM port, start the Windows Device Manager and select "Ports (COM & LPT)". If the USB interface has enumerated, it will show up as a serial port.)
3) Enter "$version" followed by the return key on the terminal window. If the MCU is powered, it will respond with firmware and MCU identification information. If no response is made when key strokes are entered, please double check the hardware connections, terminal settings and the COM port number.

4. Command line control of the evaluation board.

By using "PuTTY" or "Tera Term", terminal window interface applications, it is possible enter commands and monitor the ADC output in a log file or terminal window.

Table 5 contains all the supported commands that the firmware will recognize.

Command	Arguments	Comments
$time		Reports the board time in seconds or reads time if no arguments are given. MCU software stores this as an unsigned 32-bit integer. Response: '$TIME: 978 sec' Example: "$time"
$framusr_test $framboot_test	None	Conducts a memory test on one of the FRAM devices. Only a small upper portion (16 bytes) of memory in each device is used. User FRAM starts at 0x00. Boot FRAM starts at 0x20000 Response: "$fram_test in progress FM25V20A write data to 0x10000: 012345…" FM25V20A read data from 0x10000: 012345…" or "$framboot_test in progress CYP15B102 write data to 0x10000: 012345…" CYP15B102 read data from 0x10000: 012345…" Example: "$framboot_test" or "$framusr_test"
$flash_read	<address>	Conducts a 32-byte memory read of the FLASH device starting at the address provided in the command. The board has a preprogrammed flash at location decimal 33550336. The rest of the memory will be erased and will read as 0xff. Example: **$flash_read 33550336 32** **CYRS16B256 read data from 0x1fff000:** **52482d4f42432d312068617320612043595253313642323536205350492046 6c**
$flash_write	<address>	Caution: The Flash has a program/erase endurance specification of 1k. Using this command will perform a program cycle and reduce the available program cycles. Use prudence when executing this command! Conducts a flash write operation starting at address provided and ending when # of bytes count has expired. Value programmed will be sequential starting at 0 and rolling over at 255 of 0xFF. Example: **$flash_write 4096 8** **Response:** **"CYRS16B256 write data to 0x1000: 0001020304050607fffff"**
$flash_erase	<address>	Caution: The Flash has a program/erase endurance specification of 1k. Using this command will perform a program cycle and reduce the available program cycles. Use prudence when executing this command! Conducts a flash sector erase operation. The provided address will dictate the 4kbyte sector. For example, any address between 4096 & 8191 will erase the 4k Example: **$flash_erase 4096** **Response:** **"CYRS16B256 erase entire sector around 0x1000"**
$adc	<ADC_ch><#ADC_samples>	Performs multiple reads of a single ADC channel and reports the hexadecimal result. Channel: 16 channels supported (valid entry is 0 to 15). Number of samples: 1 to 255 Response: '$OK: ADC conversion start' followed by data. '$ERROR channel > 15" if channel too large. Example: **"$adc 3 64".** Converts channel 3, 64 times. Data format will be 5 ASCII characters separated by commas. Average will be shown on last line.

$adc_mv	<ADC_ch><#ADC_samples>	Performs multiple reads of a single ADC channel and reports a value in mV. This assumes the input voltage is 5.00V. Channel: 16 channels supported (valid entry is 0 to 15). Number of samples: 1 to 255 Response: '$OK: ADC conversion start' followed by data. '$ERROR' if: 1) invalid channel or 2) invalid # of samples. Example: "$adc 3 64". Converts channel 3, 64 times. Data format will be 5 ASCII characters separated by commas. Average will be shown on last line.
$i2c_test	None	I2c loopback test: MCU will set I2CA to master mode and I2CB to slave mode. Data will be transferred, and the pass / fail result reported. Jumper wires must be placed on the PC/104 connector for this to pass: SDA: H1-41 to H1-1, SCL: H1-43 to H1-2. Response: If passing " I2C test REQUIRES I2CA and I2CB be externally connected together I2CB sladdr 0x18, datalen=7 Cmd: SUP:ND I2CA returned success Expected Data:'12,34,5' Actual Data:12,34,5 " If failing: " $i2c_test I2C test REQUIRES I2CA and I2CB be externally connected together I2CB sladdr 0x18, datalen=7 Cmd: SUP:ND I2CA Error: SUP_I2C_ERR=3 "
$reboot	--	Warm boot of CPU. Causes software to execute a system reset. Response: Startup banner, time will show as 0. – – RH-OBC-1 Out-of-Box Firmware – Firmware Version = 2018_06_25_v1 – CPUID = 0x851c34df – – $MSG: Reset source = SYSRSTREQ –1.5V rail: 1489 mV – 3.3V rail: 3350 mV – TIME: 0 sec – FLASH read ASCII from 0x1fff000: "RH-OBC-1 has a CYRS16B256 SPI Flash. This text was read from Flash memory starting at 0x1FFF000."
$version	–	Prints startup banner containing board information with unique CPU ID, Firmware version, source of last reset , 3.3V and 1.5V supply levels. Response: Startup banner. Time is maintained. – – RH-OBC-1 Out-of-Box Firmware – Firmware Version = 2018_06_25_v1 – CPUID = 0x851c34df – – $MSG: Reset source = SYSRSTREQ –1.5V rail: 1489 mV – 3.3V rail: 3350 mV – FLASH read ASCII from 0x1fff000: "RH-OBC-1 has a CYRS16B256 SPI Flash. This text was read from Flash memory starting at 0x1FFF000."

$can_test	None	Self-test of CAN IC. Puts the device in loopback and transmits several frames. No external hardware is required. Response: "HI3110 SPI HI3110_READ_INTE reg returned=0x40, exp=0x40 $MSG:HI3110 CNTRL0 reg returned=0x20, exp=0x20" if test passes. "CAN IC error" if test fails.
$port_read		PortA and PortB are read and a 32-bit hex value displayed. Response: "PortA = 0xabcd1234, PortB = 0xDC3311"
$set_porta	<pin #> <0 or 1>	Sets a single port A pin to second parameter. Only pins on the CubeSatBus PC/104 header are permitted. PORTA[14,15,18-21], PORTA[25-27]. Other pins in this range are set as inputs. Example: $set_porta 27 0. Response: "PORTA 27 set to 0" if port number is valid, "ERR:valid mask PORTB=0x00ffc080" is reported for invalid port pin number.
$tog_porta	<pin #>	Toggles a single port A pin until next "$" is received. Pin will be left in a low state as an output when the next $ is received.. Only pins on the CubeSatBus PC/104 header are permitted. PORTA[14,15,18-21], PORTA[25-27]. Example: $tog_porta 15. Response: "Toggling PORTA[15] until next '$'. "ERR:valid mask PORTA=0x0e3cc000" is reported for invalid port pin.
$set_portb	<pin #> <0 or 1>	Sets a single port B pin to second parameter. Only pins on the CubeSatBus PC/104 header are permitted. PORTB[14-23] Example: $set_portb 14 0. Response: "PORTB14 set to 0". "Error:" is reported for invalid port pin.
$tog_portb	<pin #>	Toggles a single port B pin until next "$" is received. Pin will be left in a low state as an output. Only pins on the CubeSatBus PC104 header are permitted. PORTB[14-23] Example: $tog_portb 14 Response: "Toggling PORTB[14] until next '$'. "ERR:valid mask PORTB=0x0e3cc000" is reported for invalid port pin.

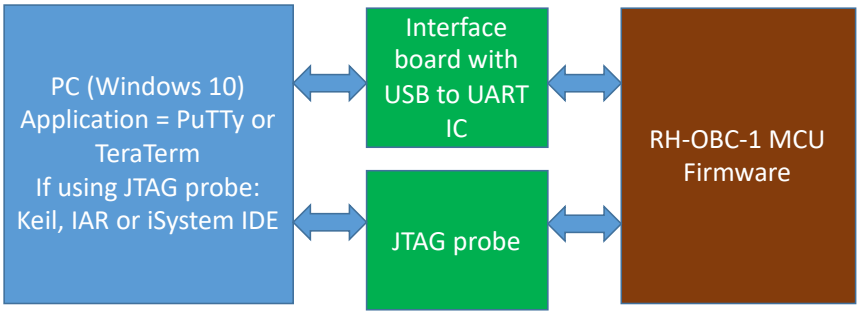
Table 5 - RH-OBC-1Firmware Commands

5. MCU Firmware Development Software Setup

The board comes preprogrammed with specific firmware to interface to the PC via a terminate window. It is possible to modify the existing code or to start from scratch.

An Integrated Development Environment (IDE) {Either Keil, IAR or iSystem} will need to be used to produce embedded code running on the VA10820 processor. The IDE allows code to be developed, compiled, debugged and programmed to an SPI Flash (BootROM) on the board. The IDE will use a JTAG probe connected to the JTAG interface header, J$1. See **Figure 5**.

Figure 5 - Software interaction for RH-OBC-1



Appendix A has:

- detailed instructions on how to download and install the tools,
- how to install programming algorithm specific to the RH-OBC-1

The software project supplied with the RH-OBC-1 board is based off the REB1 evaluation board code. All the examples and instructions in the REB1 manual will be applicable to the RH-OBC-1 project.

6. Frequently asked questions

Q1 – Can I use the USB +5V supply to power the board?

A1 – As shipped from the VORAGO, the USB +5V is not tied to the +5V rail of the board. It is possible to solder a wire from pin 20 of U$6 (FT232RL) to anode of D2. Be cognizant that the USB +5V may not be well regulated and ADC conversion accuracy will suffer.

Q2 – With no precision analog reference on the board, how can absolute value be derived?

A2 – The 3.3V regulator output and the 1.5V regulator output are available as ADC inputs. These can be used as reference voltages.

Q3 – Can several of the RH-OBC-1 boards be used in the same stack?

A3 – Yes, but the PC/104 connector would need to be removed and replaced with the "stackthrough" version. (Samtec - https://www.samtec.com/products/esq)

7. Reference documentation

Document – Description	URL
VORAGO VA10820 MCU datasheet and Programmer Guide:	http://www.voragotech.com/products/VA10800
VA108xx Evaluation board BSP (Includes User Manual)	http://www.voragotech.com/products/reb1
VORAGO application notes: (More examples of timer and SPI)	http://www.voragotech.com/resources
Cypress FRAM and Flash datasheet	Contact email: helmut.puchner@cypress.com
Intersil regulator datasheet	https://www.intersil.com/en/products/space-and-harsh-environment/rad-hard-power/rh-linear-regulation/ISL75052SEH.html
Intersil watchdog IC datasheet	https://www.intersil.com/en/products/space-and-harsh-environment/rad-hard-power/rh-supervisory/ISL706CRH.html
Holt CAN IC datasheet	http://www.holtic.com/products/3012-hi-3110-hi-3111-hi-3112-hi-3113.aspx
Aeroflex RHD5959 ADC datasheet	https://ams.aeroflex.com/pagesproduct/datasheets/rhd/rhd5950.pdf

Appendix A

RH-OBC-1 KEIL MDK INSTRUCTIONS

APPENDIX A Contents

A. Introduction

10.1 Purpose of Appendix Document

This document is intended to provide instructions on how to open the firmware project for the RH-OBC-1in the evaluation version of the Keil MDK. Details will be provided for programming the board.

10.2 Overview of Hardware and Software components

The RH-OBC-1kit includes a JTAG connector for interfacing with commercially available JTAG debug probes such as Segger J-Link and Keil ULINK. One of these JTAG debug adapters will need to be purchased separately to download new firmware to the board.

The RH-OBC-1board must be powered separately from the JTAG debug adapter. Power can be applied to head H2. H2: pin 25 => +5V, H2: pin 29 => GND. The supply should be well regulated and capable of supplying 0.5A.

Firmware for the board is provided on the VORAGO website. https://www.voragotech.com/products/rh-obc-1—-radiation-hardened-obc-c-dh-board-and-reference-design-kit. There is a complete Keil MDK project in the MCU folder with a compiled binary that can be directly downloaded to the MCU.

A specialized programming file for the RH-OBC-1 board is also included with the download. It has support for updating the watchdog during the programming process. This file contains MCU specific information that allows the Keil IDE to uniquely service the MCU with a known memory map, register definitions and programming algorithms.

10.3 Support for MCU firmware

Email: info@voragotech.com Reference RH-OBC-1in the email title.

B. Software Setup

A cross development platform describes a PC based IDE (integrated development environment) that generates code for an MCU residing on a separate board. The IDE used on the PC is the Keil MDK. A free evaluation version can be used for the reference firmware which is under 32 Kbytes. The IDE allows code to be developed, compiled, debugged and programmed to an SPI Flash device on the board.

11.1 IDE and debug probe download and install

The following downloads are required to allow full functionality of the RH-OBC-1kit. These tools are evaluation versions and are free of charge. Please download the Keil MDK as outlined in the below section. If the ULINK debug adapter is used, no unique downloads are required.

If the Segger J-Link probe is used, please download the Segger package as outlined in the below section.

11.2 Keil μVision Integrated Development Environment

Keil offers an evaluation version (MDK-Lite) of the IDE for free that will support compiled code of 32 Kbytes and below. http://www.keil.com/arm/mdk.asp (approximately size = 400 Mbytes). Please download and install the MDK-Lite program (lower left portion of web-page).

MDK Microcontroller Development Kit

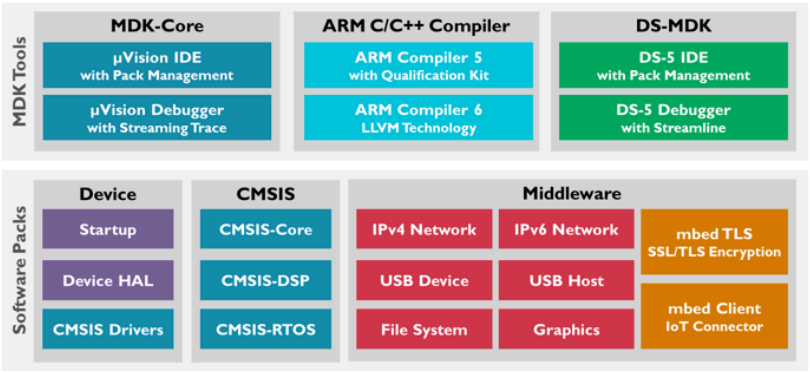
Keil® MDK is the most comprehensive software development solution for Arm®-based microcontrollers and includes all components that you need to create, build, and debug embedded applications.

Update: MDK v5.25 pre-release available

Quick Links
- Getting Started
- Online Manuals
- Middleware
- Compare MDK Editions
- Arm Tools Overview

Software Packs
- Device List
- Evaluation Boards
- MDK v4 Legacy Support
- Third-Party Software Packs
- Public Software Packs

Product Components



MDK-Core is based on µVision (Windows only) with leading support for Cortex-M devices including the new Armv8-M architecture.

DS-MDK contains the Eclipse-based (Windows and Linux) DS-5 IDE/Debugger and supports 32-bit Arm Cortex-A processors or heterogeneous systems (32-bit Arm Cortex-A and Arm Cortex-M).

MDK includes two **Arm C/C++ Compilers** with assembler, linker, and highly optimize run-time libraries that are tailored for optimum code size and performance.

Software Packs may be added any time to MDK-Core or DS-MDK making new device support and middleware updates independent from the toolchain. They contain device support, CMSIS libraries, middleware, board support, code templates, and example projects.

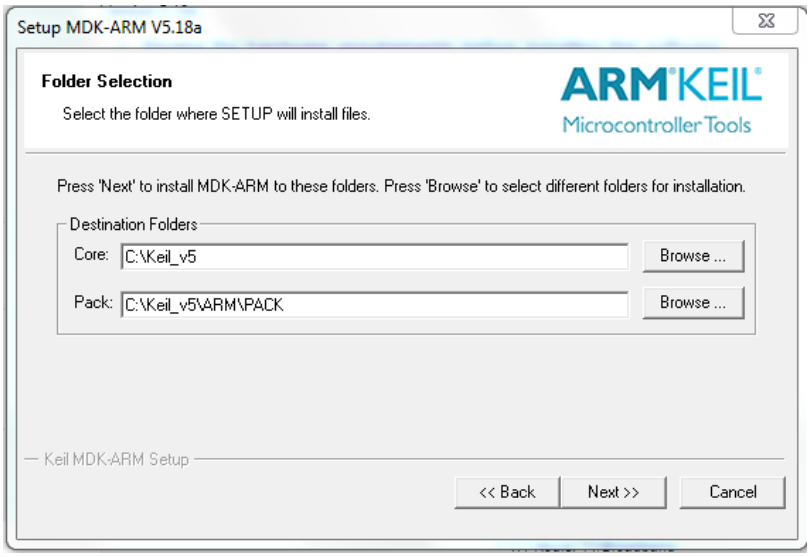
The IPv4/IPv6 networking communication stack is extended with Arm mbed™ software components to enable Internet of Things (IoT) applications.

MDK Editions

MDK is available in various editions. **Compare Editions >**

MDK-Lite	MDK-Essential	MDK-Plus	MDK-Professional
Product evaluation, small projects, and education. Code size restricted to 32	For Arm Cortex-M based microcontroller projects.	For Cortex-M, ARM7, ARM9. Includes middleware	For Cortex-M, Cortex-A, ARM7, ARM9. Includes middleware (IPv4/IPv6

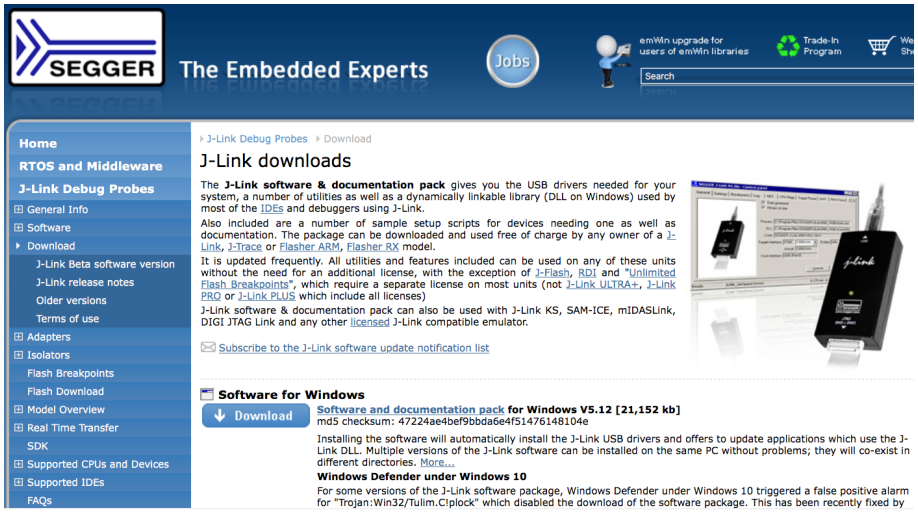
Please use the default directories for the “core” and “pack” when prompted for information during the install.



Before the MDK install is complete, the pack installer dialogue will open. Let it run to completion and then close it. The va108xx.pack will be loaded later when we open the project.

11.3 Segger J-Link

Please visit https://www.segger.com/jlink-software.html. Select Software and documentation pack for Windows V6.10n [23,162 KB]. If a later version is available, please use it.



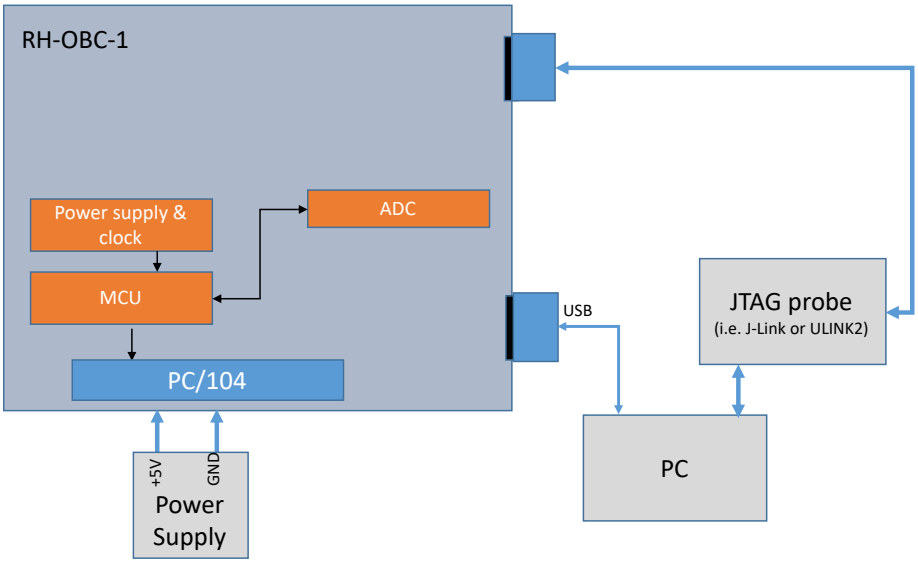
Install the J-Link software using the default selection for each prompt.

C. Hardware connections

Please refer to the below diagram, “. Before trying to communicate with the board, make sure that +5V is applied. The current draw on the +5V supply should be around 100mA. As more GPIO are utilized, the current draw will increase accordingly.

Insert the JTAG probe cable in the RH-OBC-1board. At this time, you should be ready for communicating with the MCU. The USB connection is optional, but many people find it convenient to have a terminal window to view messages as code is being executed.

If the JTAG probe only has a 0.1” center 2x5 connector, an adapter will be required for the 0.05” center 2x5 header on the board. Olimex makes a low cost adapter capable (ARM-JTAG-20-10) that can be obtained by popular distributors.

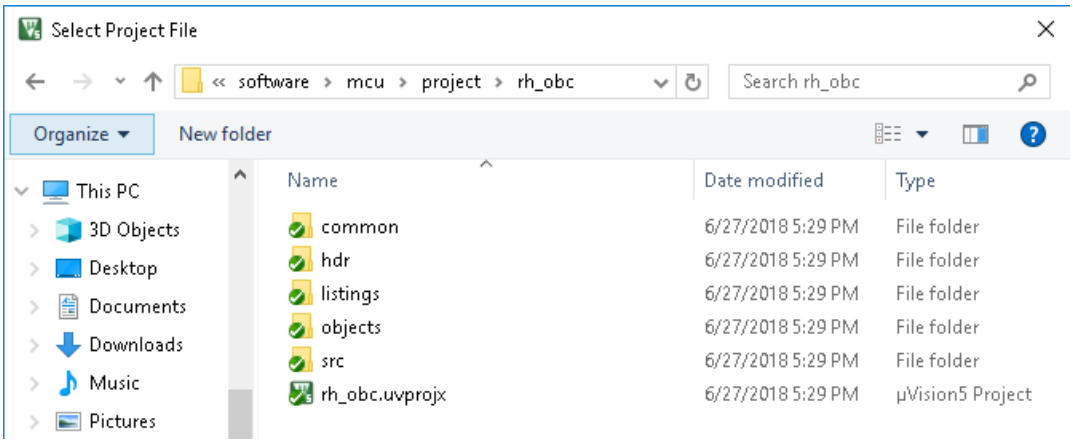


D. Starting the IDE and building a program

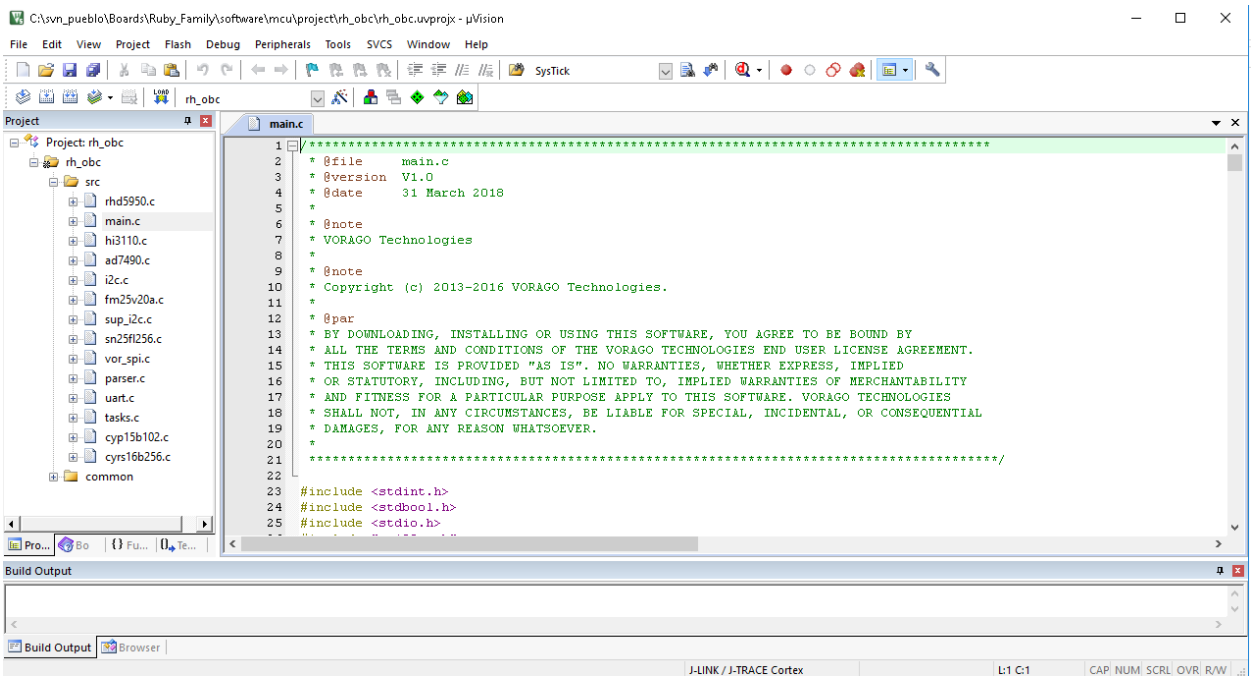
The following section provides step by step instructions for starting the IDE, downloading code and running a program on the VA10820.

13.1 Opening a project

The downloaded project has a file structure as outlined here.



Double click on "rh_obc.uvprojx" and the Keil IDE should open with this project loaded. The screen should look similar to the figure below. You may need to double click on "main.c" in the left-hand file explorer window for it to be displayed in the main editing window.



It is possible that you may get an error message referencing an invalid device being selected. This can be expected if the device pack file for the VA10820 was not installed yet. Continue to the next section to install the pack file.

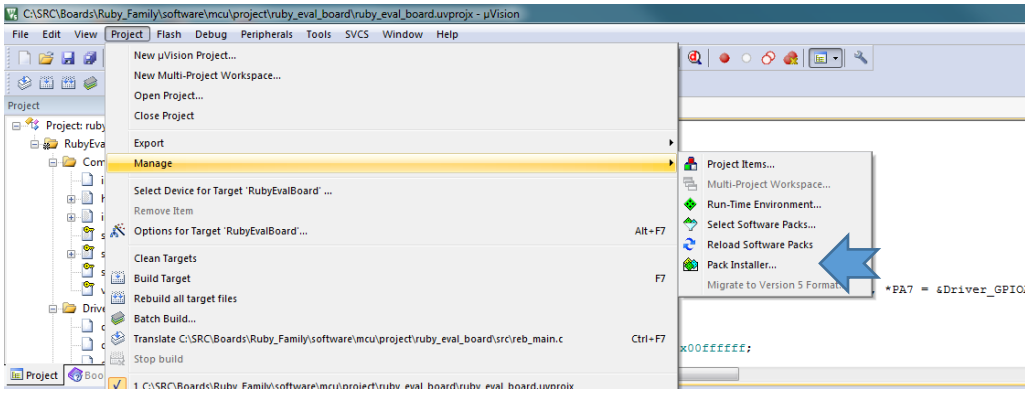
13.2 Installing a Pack File

As part of the multi-level CMSIS standard (http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php.), ARM has created an efficient way to pull in all the necessary information for an IDE to work with an MCU. The "pack" file has:

NVM programming code, header file information, documentation for the MCU, and an SVD file (System View Description).

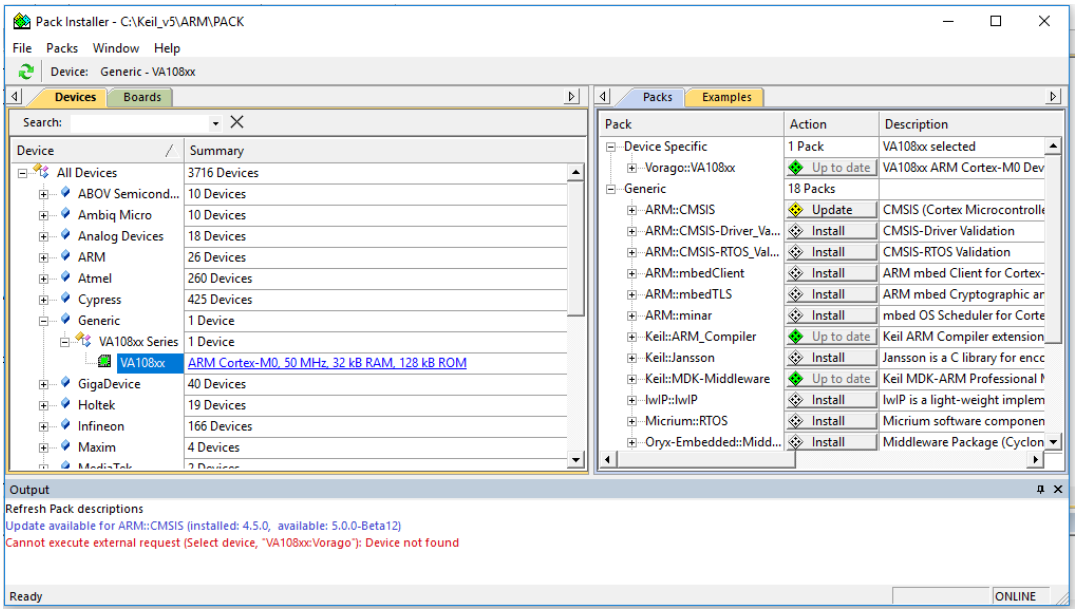
To import the pack file, follow these steps:

- From the Project pull-down menu, select “manage” and then “pack installer”. This will open another window.



- From the pack installer window, use the “file” pull-down menu and select “import”. Navigate to the “Vorago_VA108xx1.4.0.pack” file which was part of the BSP in “mcu_pack_file” folder.

-



At this point, the VA108x0 information will be installed in the Keil file structure and will be available for use. It includes:

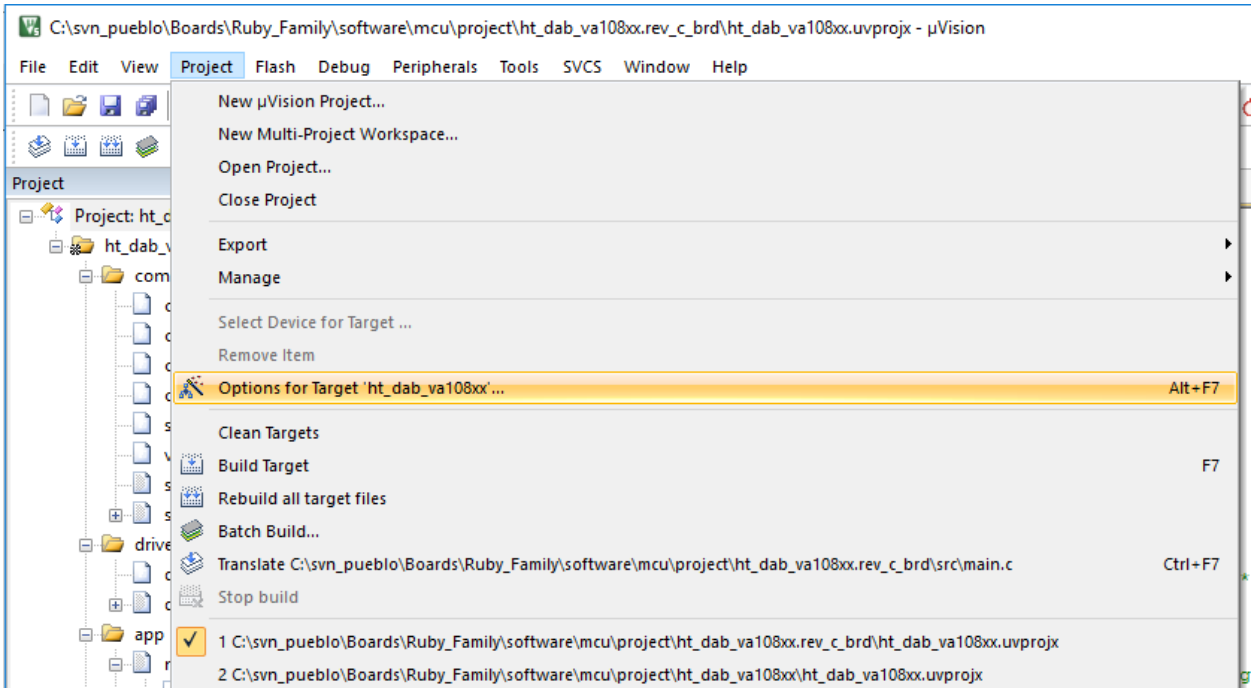
- Datasheet and programmers guide
- SVD file (used by debugger to show register and bit names)

- Va108xx.h file which has all the register and bit definitions
- Programming files for the SPI EE on the REB1 board and other SPI non-volatile memories.

13.3 Configuring options

The Keil IDE is a very powerful and flexible tool which requires several options to be assigned before it is operational with a specific MCU and JTAG debugger probe. The "rh-obc.uvprojx" file will preload the options. However, it is always a good idea to check the options. The following sections provide the minimum options for the VORAGO software development kit to function.

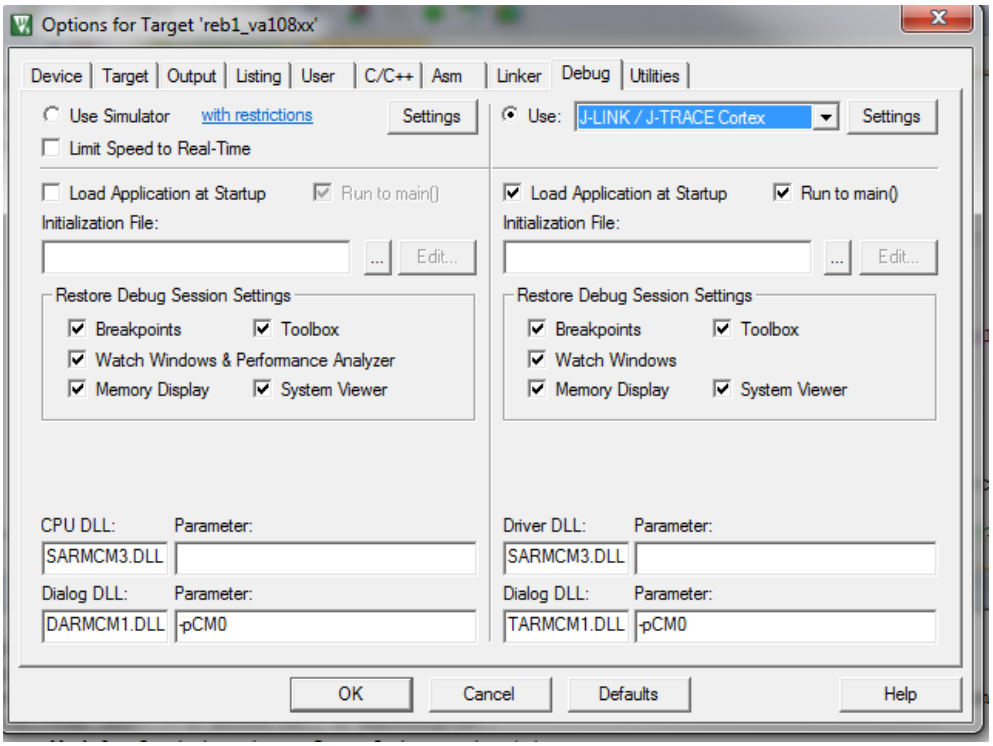
All the options can be accessed under the **Project** pull-down menu similar to the one shown here. An alternative way to access the options is to right-click on top level folder of the project and scroll to the "options" line.



An options window with 10 pull-down menus will open.

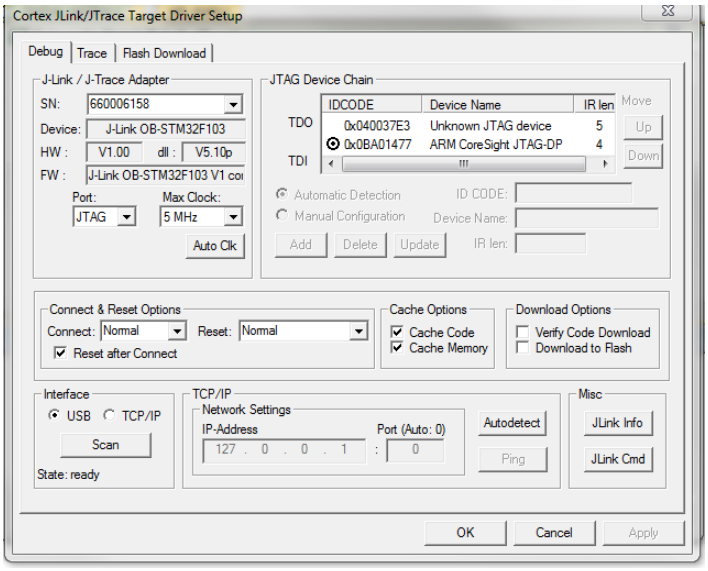
13.4 Debug options

The Keil IDE supports many forms of JTAG interfaces. The default debug connection is the ULINK2 from Keil. The IDE expects the user to specify which debug adapter is used. See below for screen captures on how to do this. First use the pull-down menu of the **Debug** options window to select "J-LINK/J-TRACE Cortex" if using a J-LINK or the appropriate one for the probe you are using.



Second, set the probe to use the JTAG connection. Click the right-most **Settings** button and select "JTAG" in the Port Window. The "Max Clock" selection can be anything up to 5 MHz. The board should be connected to a powered JTAG probe prior to setting up the Debugger. This will allow the tool to identify the J-LINK (with unique serial number) and the IDCODEs of the MCU's TAP controllers. If the connection attempt is successful, a screen similar to the below should be seen.

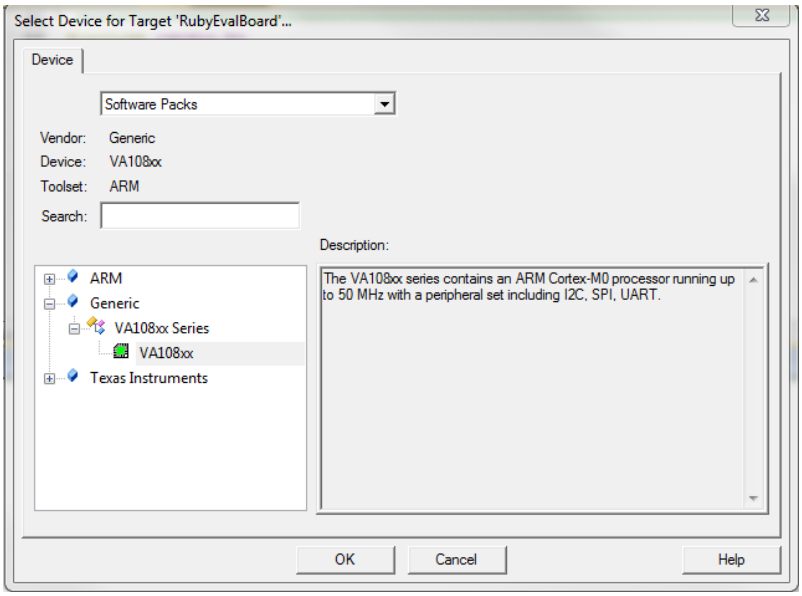
Note: An error message may be displayed stating the J-Link does not recognize the MCU. If that occurs, hit "ok" and select the generic Cortex M0 in the subsequent dialogue box. This error message should only occur on the first attachment to the board with the MDK.



Note: Two IDCODEs should appear in the "JTAG Device Chain" window. One is the standard ARM CoreSight JTAG-DP, the other is the test chain for the device. If the IDCODE field is blank, something is wrong with the connections. See section 3 for connection diagram.

13.5 Select device

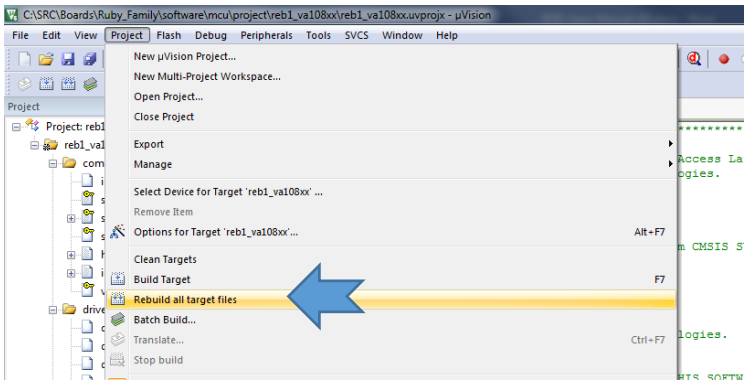
If the Vorago.VA108xx.vers.pack file has been imported, the VA108x0 will be shown under the generic group. The memory map for the VA10800 is a subset of the VA10820 and only one device is shown, VA108xx.



13.6 Project Build

The Keil tool has several ways to access many functions. There is always a pull-down menu available, but many functions have hot keys or icons that can be clicked on. To compile and link the entire project, the "Rebuild all target files" function must be called. The **Build** button translates modified or new source files and generates the executable file. The **Rebuild** command translates all source files regardless of modifications. Options for doing this include:

- Pull down menu

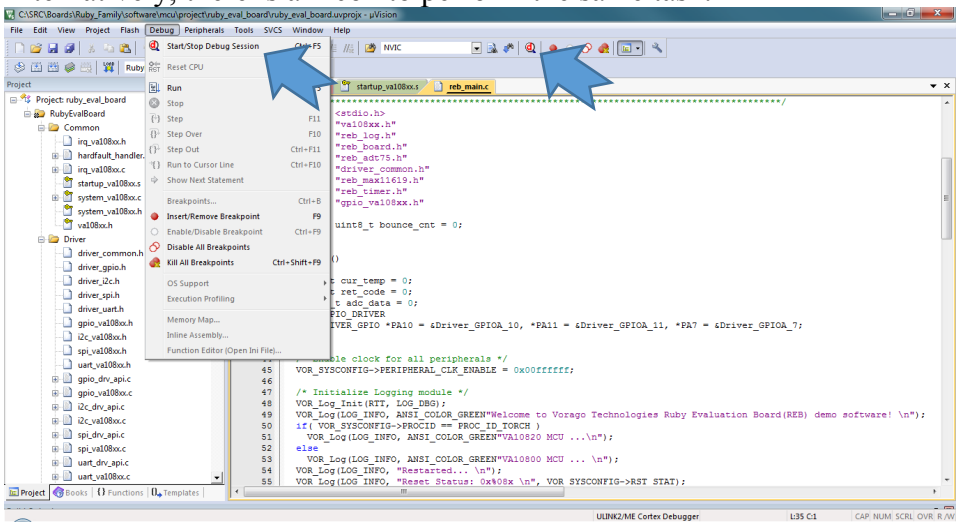


- Icon

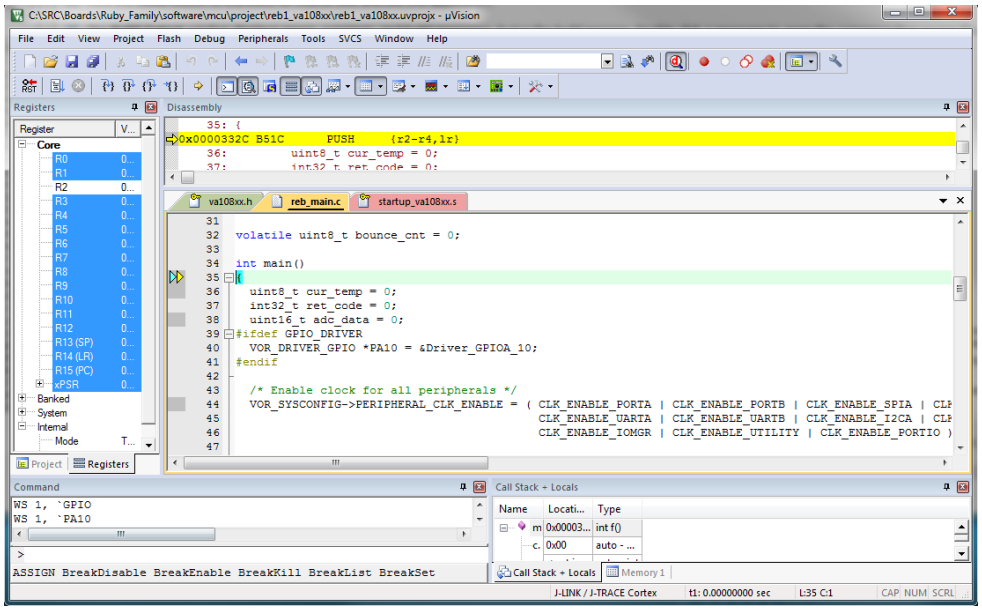
Note: The "F7" is a hot key for the **Build** Button**.**

13.7 Download and debug

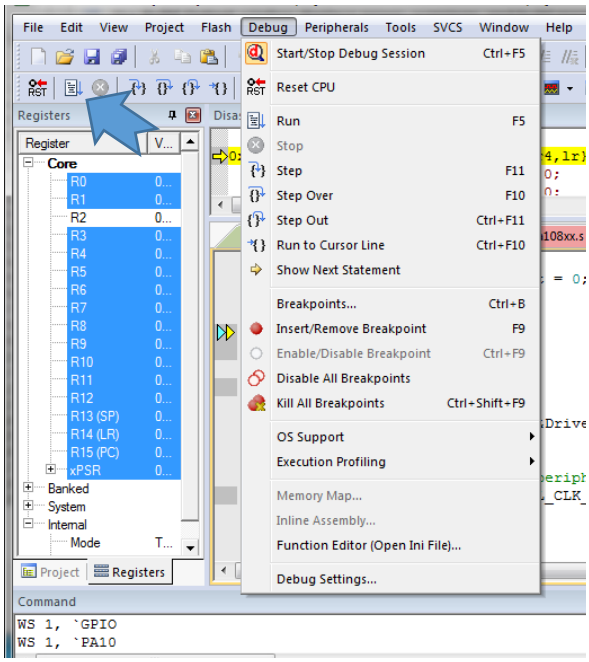
To enter a debug session, use the **Debug** pull-down menu and choose "Start/Stop Debug Session". Alternatively, there is an icon to perform the same task.



Once the IDE has entered debug mode, the screen appearance will resemble the following image.



Many options are available under the debug menu as shown here. Most of these functions have buttons on the menu bar for quick point and click access. If the program has been loaded, the **Run** button can be pressed to begin code execution.



13.8 Programming procedure (Keil Specific)

The VA108xx MCUs rely on an external SPI based memory device to boot from. The 128 Kbyte memory is transferred to the MCU's program RAM automatically during the MCU boot process by a hardware bootloader. The location of the code in the SPI memory device is the

same as the location inside the MCU. For instance, the RESET vector information is located at address 0x0000 in both devices.

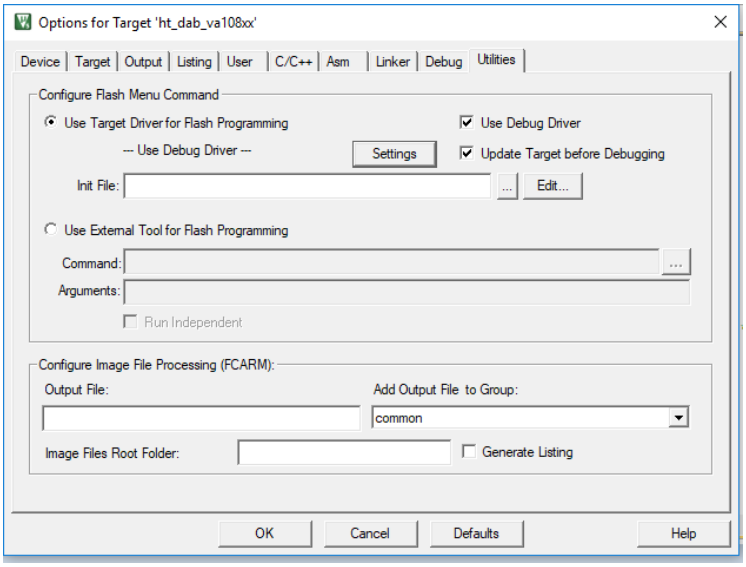
The Keil IDE provides options for programming either the flash embedded on MCUs or connected memories such as a SPI Flash device via the MCU's JTAG port. VORAGO has provided a specific programming algorithm for the RH-OBC-1 board. ("VA10820_RH_OBC.FLM"). This file will need to be manually placed in the Keil directory with other programming files: C:/Keil_V5/ARM/Flash.

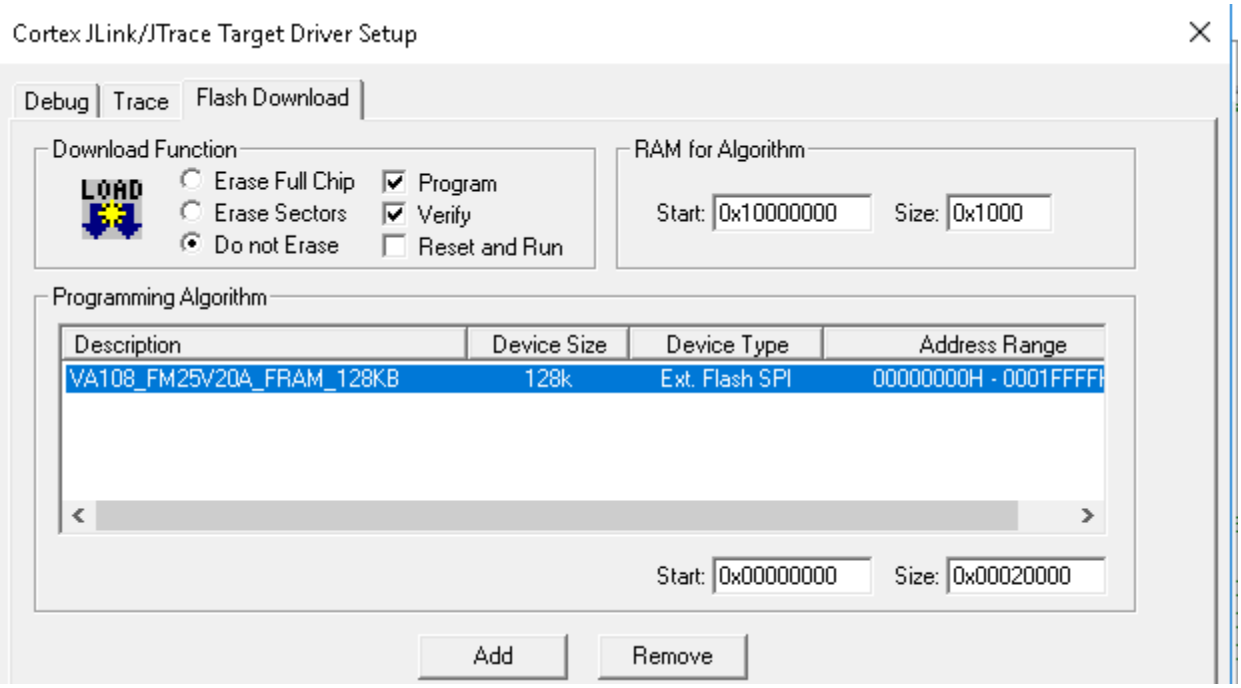
If the watchdog on the board has been disabled, the standard "VA108XX_FM25V20A_FRAM_128KB" algorithm can be used.

Programming the full 128 Kbytes of FRAM takes only a few seconds. Cypress FRAM, TT Semi Flash and Everspin MRAM is also available.

13.8.1 Utility Options

Before programming the FRAM that boots the VA10820, the Utilities options menu must be set up. First click on the **Use Target Driver for Flash Programming** button. Then click on the **Settings** button. Under the programming Algorithm section, click on the **Add** button, then select the "VA108_RH_OBC" or "VA108XX_FM25V20A_FRAM_128KB "in the list. Click "Ok" and "Ok" to exit the options menu.





Note: When not programming the RAM and just downloading to RAM, remove the VA108_xxx file from the programming algorithm list and click the **Use external flash programming** radial button. Failure to do both steps will result in unpredictable debugger operation.

E. Other resources for VA108x0 code

Vorago application notes: http://www.voragotech.com/resources

F. Revision history

1.0 June 2018

[end of file]