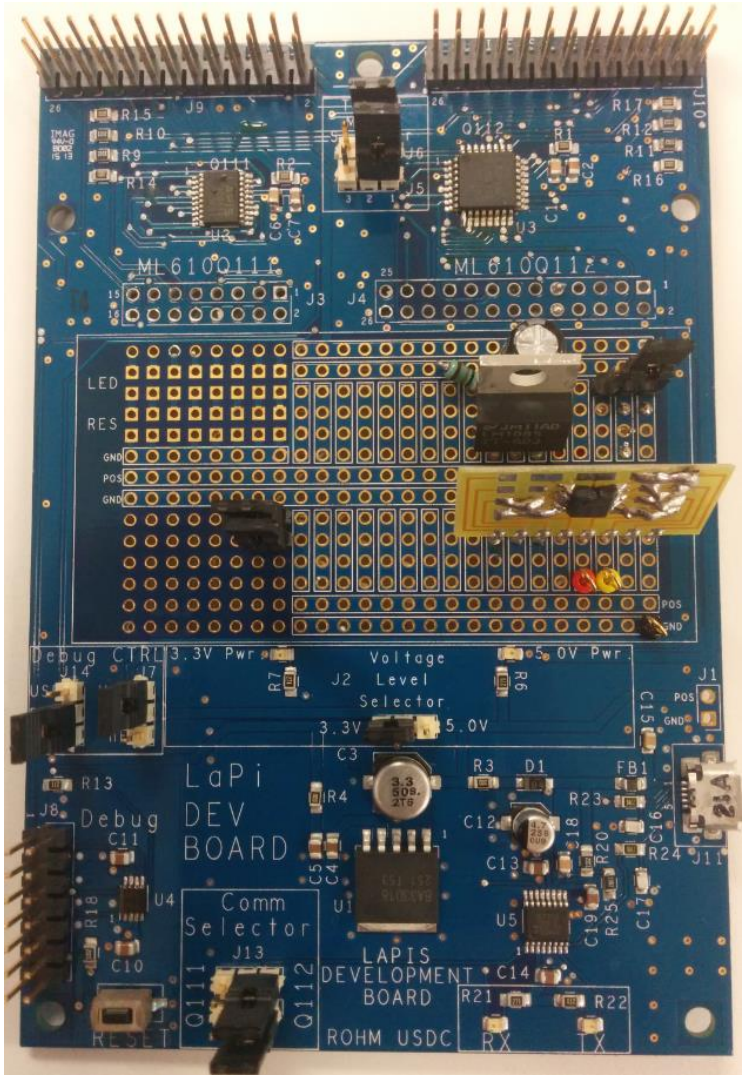
BD99935 I2C Emulator Board Manual

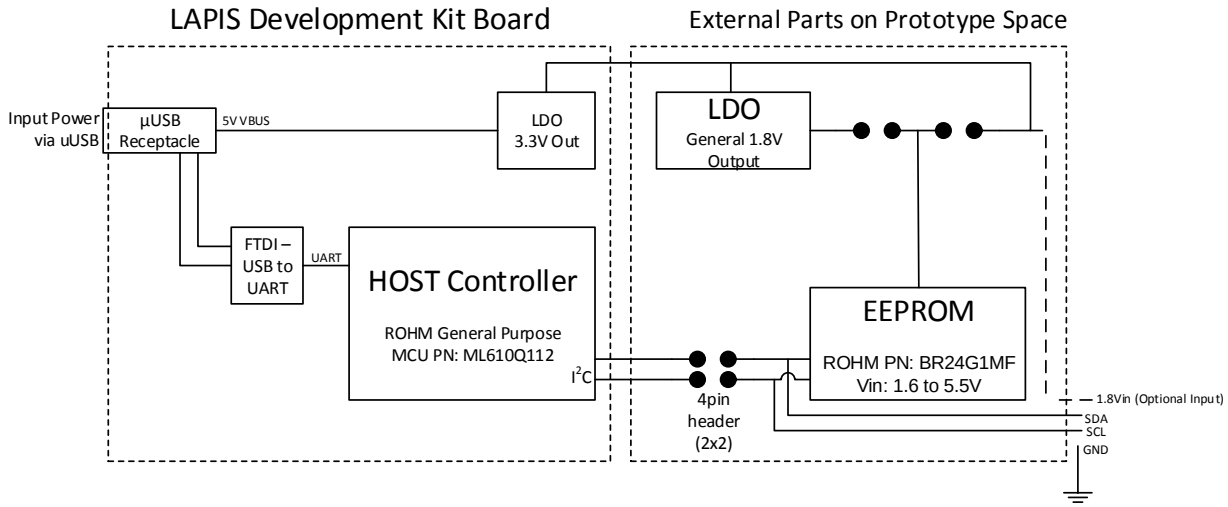


Above: Lapis Development Kit with added EEPROM and LDO for supporting BD99935 pre-testing

U.S. Design Center

Table of Contents

High Level Block Diagram



- The High level block diagram for this application board can be seen above. The left side of this picture shows the blocks used on the existing Lapis Development Kit. The right side of this picture shows the block used on the prototype space of the Lapis Development Kit.

Quick Start Guide

1. The jumpers should be set when we send the board, but jumpers should match the board as per the definitions in the "Hardware Board Explanations" section
2. Power the board using the a uUSB cable to Connector J11
3. Upon power-on the EEPROM will be initialized with the appropriate register map
    - Note: the RESET button can be used to re-initialize the EEPROM register map
4. When ready to connect to the different host, please disconnect the I2C pin jumpers to the Host MCU to ensure the I2C data bus is free.
5. Also, depending on the voltage level, adjust the jumper above the LDO
    - LEFT most Pins = 1.8V
    - RIGHT most Pins = 3.3V
6. Then connect the pins to the board
    - Orange = SDA
    - Yellow = SCL
    - Black = GND

EEPROM Device Address (7bit) = 0x50

IMPORTANT NOTE: This EEPROM requires a TWO BYTE register addressing scheme.

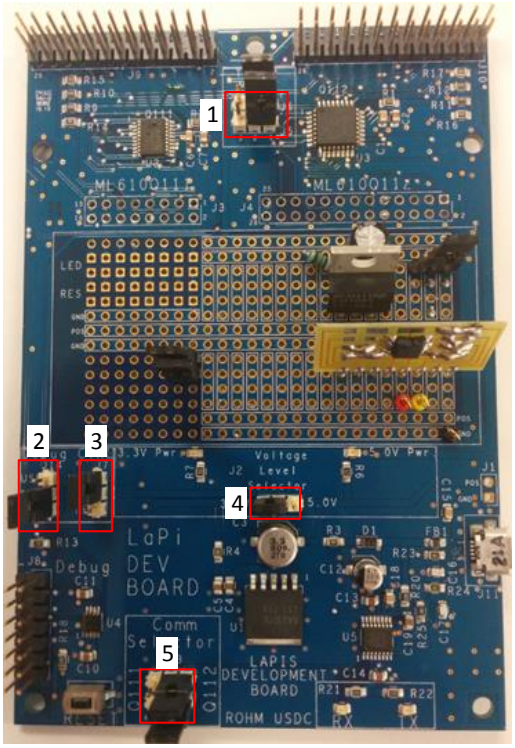
EEPROM Register Contents

Address	Reg Name	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Initial Value	
0x00	VENDORID	R	VENDORID[7:0]								0x1F	Defined
0x01	REVID	R	MAJREV[3:0]				MINREV[3:0]				0xA0	Defined
0x02	IRQLVL1	R/W	-	-	-	-	-	ADC	PWRSRC	TEMP	0x00	Level1 interrupt register
0x03	IRQLVL1M	R/W	-	-	-	-	-	ADCM	PWRSRCM	TEMPM	0x07	Level1 interrupt mask register
0x04	IRQTEMP	R/W	-	-	-	-	CRITTEMPOD	WARNTEMPOD	CRITTEMPEXT	WARNTEMPEXT	0x00	Temperature Second Level interrupt register
0x05	IRQPWRSRC	R/W	-	IRECTOCP	WC_READY	LMDONE	OVPWARNDET	OCVRECT	UVVRECT	OVVRECT	0x00	Power Source Second Level interrupt register
0x06	IRQADC	R/W	-	-	-	-	-	ADCMBICAL	ADCVRECTICAL	RND	0x00	ADC Second Level interrupt register
0x07	IRQTEMPM	R/W	-	-	-	-	CRITTEMPODM	WARNTEMPODM	CRITTEMPEXTM	WARNTEMPEXTM	0x0F	Temperature Second Level interrupt mask register
0x08	IRQPWRSRCM	R/W	-	IRECTOCPM	WC_READYM	LMDONEM	OVPWARNDETM	OCVRECTM	UVVRECTM	OVVRECTM	0x4E	Power Source Second Level interrupt mask register
0x09	IRQADCM	R/W	-	-	-	-	-	ADCMBICALM	ADCVRECTICALM	RNDM	0x07	ADC Second Level interrupt mask register
0x0A	VRCNT1	R/W	LDOEN	LDOPGEN	-	-	-	-	-	-	0xC0	Voltage Regulator Control Register1
0x0B	VRCNT2	R/W	SBEN	SBPGEN	-	-	-	-	WCR_EN	-	0xC2	Voltage Regulator Control Register2
0x0C	OVPWARN	R/W	OVP_SET[3:0]				-	OVPWARN[2:0]			0x00	OVP Warning Register
0x0F	LDMOD	R/W	-	-	-	-	-	-	WCOUTEN	LMEN	0x01	Load Modulation / Simple Buck Driver control register
0x30	ADCCNT1	R/W	-	MBICALOFFSETEN	MBICALEN	ADEN	ADSTRT	ADSLP[2:0]			0x18	ADC Control register1
0x31	ADCCNT2	R/W	-	VRECTBITSEL	CRITCNTEN	VRECTOFFSETEN	VRECTCALEN	ADCTHERM	-	RRDATARD	0x00	ADC Control register2
0x33	ADCADDR0	R/W	-	-	-	AVRGCH0	STOPCH0	ADSEL0[2:0]			0x00	ADC Address Register 0
0x34	ADCADDR1	R/W	-	-	-	AVRGCH1	STOPCH1	ADSEL1[2:0]			0x00	ADC Address Register 1
0x35	ADCADDR2	R/W	-	-	-	AVRGCH2	STOPCH2	ADSEL2[2:0]			0x00	ADC Address Register 2
0x36	ADCADDR3	R/W	-	-	-	AVRGCH3	STOPCH3	ADSEL3[2:0]			0x00	ADC Address Register 3
0x37	ADCADDR4	R/W	-	-	-	AVRGCH4	STOPCH4	ADSEL4[2:0]			0x00	ADC Address Register 4
0x38	ADCADDR5	R/W	-	-	-	AVRGCH5	STOPCH5	ADSEL5[2:0]			0x08	ADC Address Register 5

Address	Reg Name	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Initial Value	
0x40	ADCVRECTOFFSETL	R	ADCVRECTOFFSETL[7:0]								0x00	ADC Offset Data Register for VRECT Current Calibration(Low)
0x41	ADCVRECTOFFSETH	R	-	-	-	-	-	-	ADCVRECTOFFSETH[1:0]		0x00	ADC Offset Data Register for VRECT Current Calibration(High)
0x42	ADCMBIOFFSETL	R	ADCMBIOFFSETL[7:0]								0x00	ADC Offset Data Register for Main Buck Current Calibration(Low)
0x43	ADCMBIOFFSETH	R	-	-	-	-	-	-	ADCMBIOFFSETH[1:0]		0x00	ADC Offset Data Register for Main Buck Current Calibration(High)
0x44	ADCCRTTMPEXTL	R/W	ADCCRITTMPEXTL[7:0]								0x26	ADC cririca l external temp-sensor threshold level register(Low)
0x45	ADCCRTTMPEXTH	R/W	-	-	-	-	-	-	ADCCRITTMPEXTH[1:0]		0x00	ADC cririca l external temp-sensor threshold level register(High)
0x46	ADCWRNTMPEXTL	R/W	ADCWARNTMPEXTL[7:0]								0x51	ADC warning level external temp-sensor threshold level register(Low)
0x47	ADCWRNTMPEXTH	R/W	-	-	-	-	-	-	ADCWARNTMPEXTH[1:0]		0x00	ADC warning level external temp-sensor threshold level register(High)
0x48	ADCOCVRECTL	R/W	ADCOCVRECT[7:0]								0x28	ADC VRECT output load current threshold level register for MSB 8bit
0x49	IRECTOCP	R/W	IRECTOCPDIS	OCVRECTDIS	-	-	-	-	IRECTOCP[1:0]		0x00	VRECT input current over current protection level setting
0x4A	VRANGESET1	R	-	-	-	-	-	VRANGESET1[2:0]			0x--	ADC VRECT dynamic range setting Read Only. Read data depend on WC_READY
0x4B	VRANGESET2	R/W	WRITEEN	-	-	-	-	-	VRANGESET2[1:0]		0x00	ADC VRECT dynamic range setting
0x4C	LVLDETIREF	R/W	-	-	-	-	-	-	LVLDETIREF[1:0]		0x00	LVLDET block iref setting
0x50	ADCSNS0L	R	ADCSNS0L[7:0]								0x00	ADC Round Robin Data0(Low)
0x51	ADCSNS0H	R	ADCSNS0H[7:0]								0x00	ADC Round Robin Data0(High)
0x52	ADCSNS1L	R	ADCSNS1L[7:0]								0x00	ADC Round Robin Data1(Low)
0x53	ADCSNS1H	R	ADCSNS1H[7:0]								0x00	ADC Round Robin Data1(High)
0x54	ADCSNS2L	R	ADCSNS2L[7:0]								0x00	ADC Round Robin Data2(Low)
0x55	ADCSNS2H	R	ADCSNS2H[7:0]								0x00	ADC Round Robin Data2(High)
0x56	ADCSNS3L	R	ADCSNS3L[7:0]								0x00	ADC Round Robin Data3(Low)
0x57	ADCSNS3H	R	ADCSNS3H[7:0]								0x00	ADC Round Robin Data3(High)
0x58	ADCSNS4L	R	ADCSNS4L[7:0]								0x00	ADC Round Robin Data4(Low)
0x59	ADCSNS4H	R	ADCSNS4H[7:0]								0x00	ADC Round Robin Data4(High)
0x5A	ADCSNS5L	R	ADCSNS5L[7:0]								0x00	ADC Round Robin Data5(Low)
0x5B	ADCSNS5H	R	ADCSNS5H[7:0]								0x00	ADC Round Robin Data5(High)
0x60	UVLOVINDB	R/W	-	-	-	-	-	UVLOVINDB[2:0]			0x00	UVLO Debounce Time Select
0x61	SMBDBG	R/W	-	-	-	-	-	-	SMBCPCLKSEL[1:0]		0x00	Simple Buck Control Register
0x62	WCWAIT	R/W	-	-	-	-	-	-	WCWAITSEL[1:0]		0x00	After WC_READY Wait Time Control Register
0x70	SOFTRESET	R/W	SOFTRESET								0x00	Soft Reset Control Register

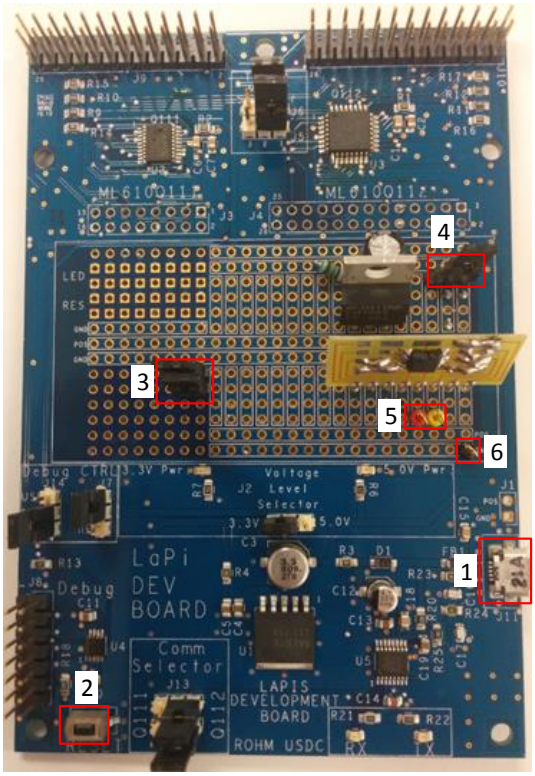
Hardware Board Explanation

Jumper Explanations



1. This is a selector Jumper for choosing which MCU to be programmed. For this application this is N/A since the user will not be re-programming the MCU.
    a. J5 = Jump Pins 1 and 2
    b. J6 = Jump Pins 1 and 2
2. This is a selector jumper to choosing how to connect the RESET pin. For this application, we want to use the RESET push button to reset the application if an EEPROM re-write is required.
    a. J14 = Jump USR to RST
3. This is a selector jumper for choosing how to source power to this board. For this application we want to source 5V power from the uUSB connector.
    a. J7 = Jump USR to PWR
4. This is a selector jumper for choosing the voltage level on the “POS” lines of the prototyping space and MCU source power. For this application can use either 3.3V or 5V (but tested under 3.3V condition)
    a. J2 = Jump 3.3V to middle pin
5. This is a selector jumper for choosing the output MCU for the UART lines for either the Q111 or Q112 MCU. This function is not used in this application
    a. Default: Connect Q112 side to middle pin for both jumpers

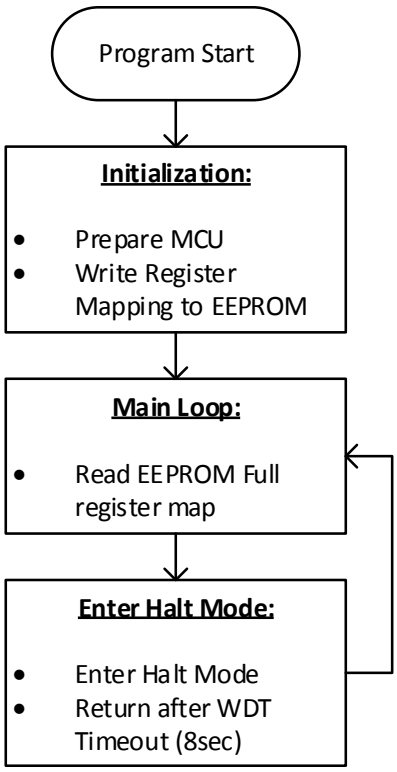
Primary Hardware Points



1. uUSB Receptacle
    a. This serves two functions
        i. Power the board using the USB 5V VBUS line
        ii. Sets up a UART debug COM port on the host PC
    b. For this application, only 5V Vbus is used
2. Reset Button for the LAPIS Q112 MCU
    a. For this application, this will serve to re-write the EEPROM to the original settings
3. I2C Connection Jumpers
    a. This connects the I2C bus lines to the onboard LAPIS Q112 MCU.
    b. When connecting to an external host, these should not be jumped
    c. When resetting the EEPROM memory map, this should be jumped
4. On-Board Power Selector
    a. This connector allows the user to select the voltage rail of the EEPROM.
        i. Jump Left = 1.8V
        ii. Jump Right = 3.3V
5. External Jumpers for I2C Connection
    a. Orange = SDA
    b. Yellow = SCL
6. External Jumper for GND

Firmware Flow Chart

High Level Flowchart of Operation

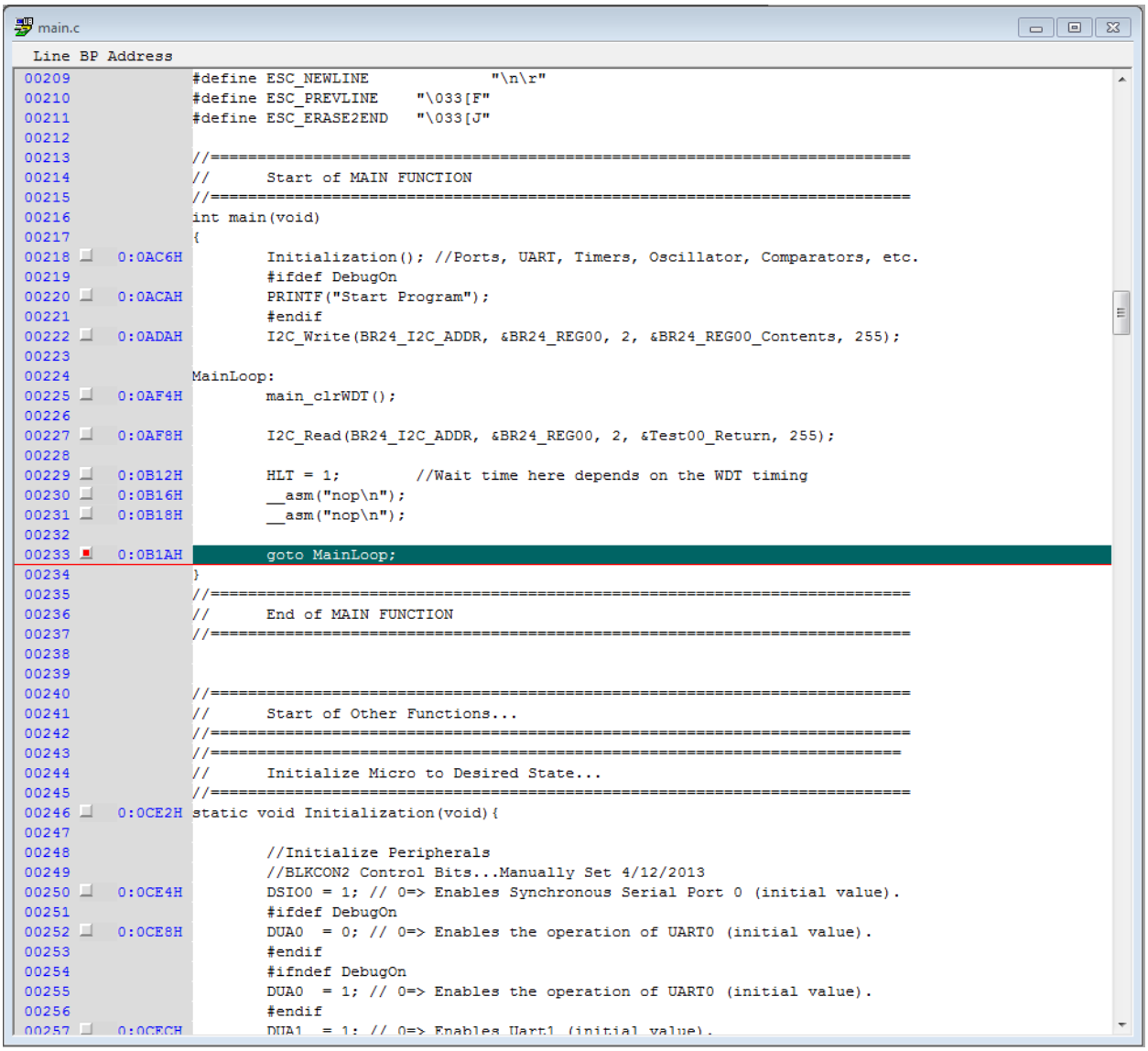


- The Purpose of the MCU on this board is to complete only 1 task: to write the default register mapping to the EEPROM device.
- In the main loop, the EEPROM is read to check the register contents, but ultimately is not helpful to the end user.

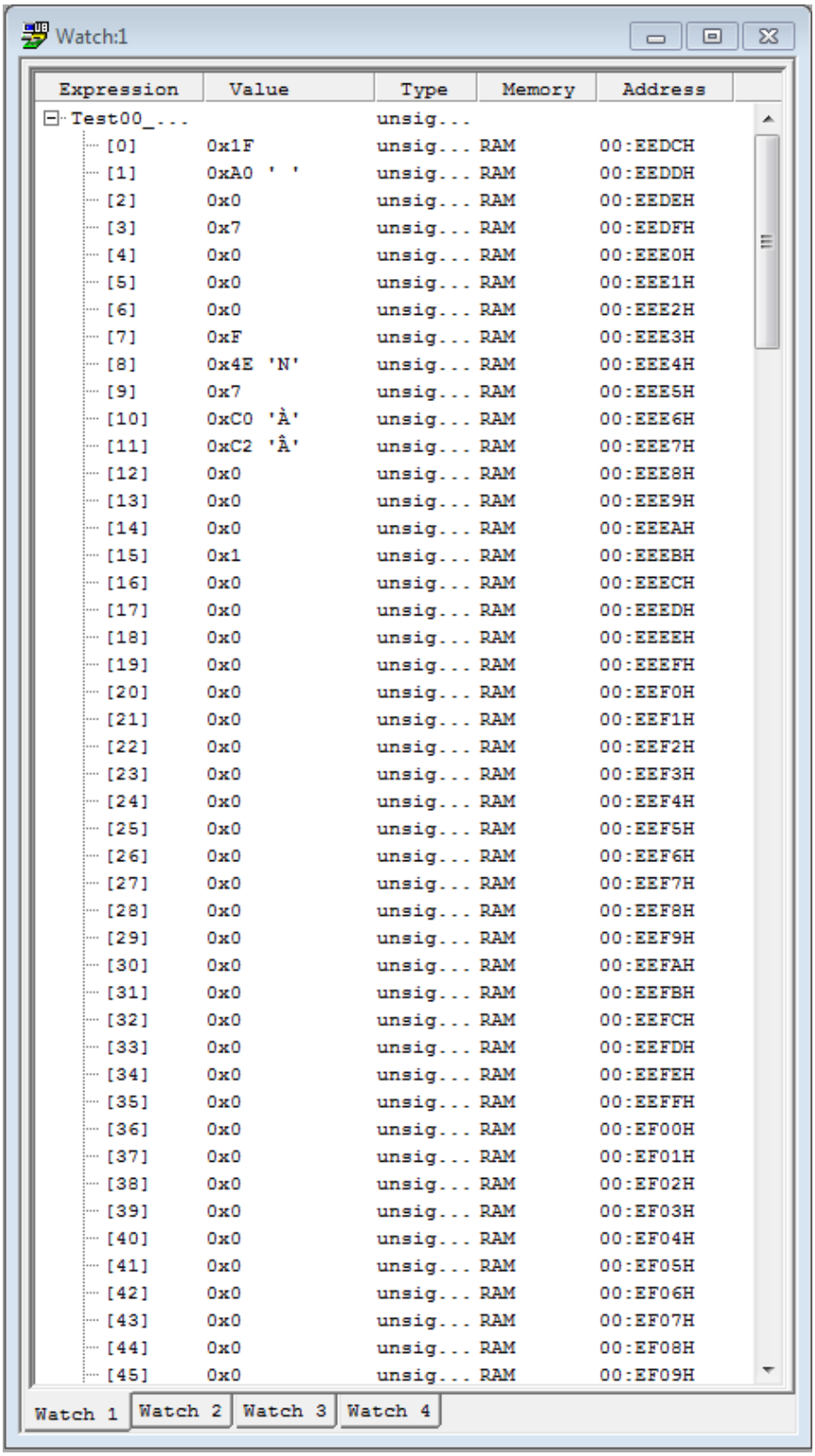
EEPROM Register Content Confirmation

In order to ensure the EEPROM is working as specified, we confirmed operation of this device using two methods.

Method 1: Checking the EEPROM using the on-board I2C Reads

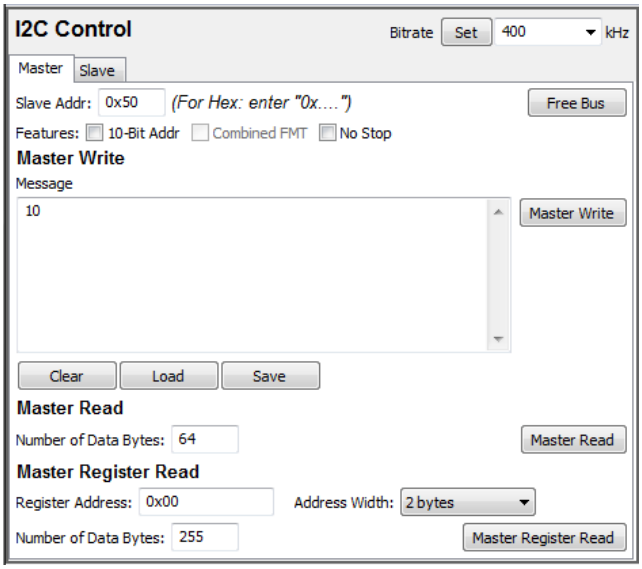


In the above picture, we have setup a breakpoint after the MCU goes into Halt Mode. Thus, at this point, the MCU has written and re-read the register contents of the EEPROM. The Contents are then placed in the character array “Test00_Return”. Upon checking this variable in the debugger, we can see that the register values were taken successfully


Watch:1
Expression
Value
Type
Memory
Address
Test00_...
unsig...
[0] 0x1F unsig... RAM 00:EEDCH
[1] 0xA0 ' ' unsig... RAM 00:EEDDH
[2] 0x0 unsig... RAM 00:EEDEH
[3] 0x7 unsig... RAM 00:EEDFH
[4] 0x0 unsig... RAM 00:EEE0H
[5] 0x0 unsig... RAM 00:EEE1H
[6] 0x0 unsig... RAM 00:EEE2H
[7] 0xF unsig... RAM 00:EEE3H
[8] 0x4E 'N' unsig... RAM 00:EEE4H
[9] 0x7 unsig... RAM 00:EEE5H
[10] 0xC0 'À' unsig... RAM 00:EEE6H
[11] 0xC2 'Â' unsig... RAM 00:EEE7H
[12] 0x0 unsig... RAM 00:EEE8H
[13] 0x0 unsig... RAM 00:EEE9H
[14] 0x0 unsig... RAM 00:EEEAH
[15] 0x1 unsig... RAM 00:EEEBH
[16] 0x0 unsig... RAM 00:EEECH
[17] 0x0 unsig... RAM 00:EEEDH
[18] 0x0 unsig... RAM 00:EEEEH
[19] 0x0 unsig... RAM 00:EEEFH
[20] 0x0 unsig... RAM 00:EEF0H
[21] 0x0 unsig... RAM 00:EEF1H
[22] 0x0 unsig... RAM 00:EEF2H
[23] 0x0 unsig... RAM 00:EEF3H
[24] 0x0 unsig... RAM 00:EEF4H
[25] 0x0 unsig... RAM 00:EEF5H
[26] 0x0 unsig... RAM 00:EEF6H
[27] 0x0 unsig... RAM 00:EEF7H
[28] 0x0 unsig... RAM 00:EEF8H
[29] 0x0 unsig... RAM 00:EEF9H
[30] 0x0 unsig... RAM 00:EEFAH
[31] 0x0 unsig... RAM 00:EEFBH
[32] 0x0 unsig... RAM 00:EEFCH
[33] 0x0 unsig... RAM 00:EEFDH
[34] 0x0 unsig... RAM 00:EEFEH
[35] 0x0 unsig... RAM 00:EEFFH
[36] 0x0 unsig... RAM 00:EF00H
[37] 0x0 unsig... RAM 00:EF01H
[38] 0x0 unsig... RAM 00:EF02H
[39] 0x0 unsig... RAM 00:EF03H
[40] 0x0 unsig... RAM 00:EF04H
[41] 0x0 unsig... RAM 00:EF05H
[42] 0x0 unsig... RAM 00:EF06H
[43] 0x0 unsig... RAM 00:EF07H
[44] 0x0 unsig... RAM 00:EF08H
[45] 0x0 unsig... RAM 00:EF09H
Watch 1
Watch 2
Watch 3
Watch 4

Method 2: Checking the EEPROM using an Aardvark

In the below picture, the Total Phase Aardvark I2C tool was used to confirm the operation of this device



After Performing the full register map read, we can see that the contents are loaded successfully.