

AVR® DA Training Manual

Low-Power Modes Using Curiosity Nano

Prerequisites

- Hardware Prerequisites
 - AVR128DA48 Curiosity Nano Board (DM164151) with a small hardware modification described in this document
 - Power Debugger (ATPOWERDEBUGGER)
 - Two Micro-USB cables: one for the Power Debugger and one for the Curiosity Nano Board
 - Two small wires with holes on both ends for pins
- Software Prerequisites
 - MPLAB[®] X 5.40
 - MCC plug-in for MPLAB X version 3.95
 - AVR library for MCC version 2.3.0
 - Atmel Studio newest version

Introduction

This document shows the differences in power consumption between a pure software approach and the use of CIPs for accomplishing different tasks.

This hands-on training covers the following topics:

- Assignment 1: Event System Compared to Interrupts
- Assignment 2: Software Accumulation Compared to Hardware Accumulation for Analog-to-Digital Converter (ADC)

Basic theory related to each assignment is introduced at the beginning of each assignment, the functionality is then verified through the Atmel Studio Data Visualizer.

Icon Key Identifiers

The following icons are used in this document to identify different assignment sections and to reduce complexity.

i	Info: Delivers contextual information about a specific topic.
	Tip: Highlights useful tips and techniques.
	To do: Highlights objectives to be completed.
	Result: Highlights the expected result of an assignment step.
À WARNING	Indicates important information.
	Execute: Highlights actions to be executed out of the target when necessary.

Table of Contents

Pre	requisi	tes	1
Intro	oductio	n	1
Icor	n Key lo	dentifiers	2
1.	Assigi	nment 1: Event System Compared to Interrupts	4
	1.1.	Hardware Description	4
	1.2.	Modifying the Curiosity Board to Obtain Accurate Power Measurements	4
	1.3.	Explaining the Modules Used	5
	1.4.	Creating the Software Implementation Program	5
	1.5.	Creating the Event System Implementation Project	13
	1.6.	Programming the Device and Comparing Efficiency	23
2.	Assigi	nment 2: Software Accumulation Compared to Hardware Accumulation for ADC	
	2.1.	Creating the Software Implementation Project	28
	2.2.	Creating the Hardware Implementation Project	34
	2.3.	Programming the Device and Comparing Power Efficiency	41
3.	3. Revision History		
The	The Microchip Website		
Pro	duct Cl	nange Notification Service	47
Cus	tomer	Support	47
Mic	Microchip Devices Code Protection Feature		
Leg	Legal Notice		
Trac	Trademarks		
Qua	Quality Management System		
Woi	ldwide	Sales and Service	

1. Assignment 1: Event System Compared to Interrupts

In this assignment, the power consumption of two different approaches will be compared. The first approach involves waking up the microcontroller from Sleep through a timer overflow interrupt that starts the ADC conversion to read the temperature sensor. When the conversion is done, the Result Ready Interrupt is triggered, and the result of the conversion is read.

In the second approach, the Event System (EVSYS) module will be used. This module allows the user to connect different peripherals through events. The previous approach will be modified such that the timer overflow will trigger an event that starts the ADC conversion that reads the temperature sensor. The end of the conversion will wake the microcontroller from Sleep to read the result.

1.1 Hardware Description

The Curiosity board is based on an AVR128DA48 microcontroller, a Nano embedded debugger, an LED, and a button. The Nano debugger allows the user to program the board and run debugging features without the need for external hardware. It also provides UART-USB bridge capabilities.

The Power Debugger is a programmer that also offers power outputs with configurable voltages and can be used to measure the current consumption of the devices connected to it.

1.2 Modifying the Curiosity Board to Obtain Accurate Power Measurements



To do: Modify the Curiosity Nano board by cutting a trace and soldering two pins.

If the power is observed form the micro-USB port, the Power debugger will consume some power and make the readings inaccurate.

Figure 1-1. POWER Trace Location



There is a small trace labeled POWER on the Curiosity board. By using a sharp tool, cut the trace until the two holes are not in contact any more. Check this with a voltmeter.

After the trace has been cut, solder two pins in the holes. To program the board, the two pins have to be connected with a wire from now on.

Assignment 1: Event System Compared to Int...

Figure 1-2. Modified Curiosity Board



1.3 Explaining the Modules Used

The modules that will be used in this assignment are the Real-Time Clock (RTC), the ADC, the Voltage Reference (VREF) and the Event System (EVSYS).

The RTC will work as a timer that counts to one second.

The ADC is a module that transforms an analog voltage into a digital 12-bit number. It can be made to start a conversion from the EVSYS. It also raises an event and an interrupt when it finishes a conversion. The ADC will be used to read the internal temperature sensor.

The Voltage Reference is a peripheral that provides stabilized voltages to other peripherals. It also enables the temperature sensor if it is set to provide 2.048V reference to the ADC.

The Event System is a routing network enabling inter-peripheral communication without involving the CPU. The communication is done by sending events from one peripherals to another triggering actions. Events are latency-free and never lost, enabling better real-time and reliable applications.

1.4 Creating the Software Implementation Program



To do: Configure the hardware modules that will be used in this project and add code to the project.

- 1. Create a new MPLAB X project for AVR128DA48.
 - 1.1. Open MPLAB X.
 - 1.2. Select $\underline{File} \rightarrow \underline{New \ Project}$ or the **New Project** button. Figure 1-3. New Project

File Edit View Navigate



1.3. Click Next (the Microchip Embedded Stand-alone Project is selected by default).

×

Assignment 1: Event System Compared to Int...

Creates a new standalone application project. It uses an IDE-generated makefile to build your

Next >

Cancel

Help

Finish

😰 New Project		
Steps	Choose Project	
1. Choose Project	Q Filter:	
2	Categories: Microchip Embedded Other Embedded Samples	Projects: Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project Import START MPLAB Project

Description:

project.

Figure 1-4. Project Type

1.4. In the Device field, search for AVR128DA48. In the Tool category, select the Curiosity Nano board if it is connected to the computer, otherwise select **None**. Click **Next**.

< Back

Figure 1-5. Device Selection

🔀 New Project		>
Steps	Select Device	
1. Choose Project 2. Select Device		
Select Header Select Plugin Board Select Compiler	Family:	All Families
6. Select Project Name and Folder	Device:	AVR 128DA48 V
	Tool:	AVR 128DA48 Curiosity Nano-SN: MCH 🗸 🔲 Show All
~		
		< Back Next > Finish Cancel Help

1.5. Select XC8 2.20 compiler and click Next.

Figure 1-6. Compiler Selection

🔀 New Project		×
 Steps Choose Project Select Device Select Header Select Plugin Board Select Compiler Select Project Name and Folder 	Select Compiler Compiler Toolchains 	
	< Back Next > Finish Cancel Hel	lp

1.6. Give a name to the project (and the location where to be saved) and click **Finish**. **Figure 1-7. Project Name**

🔀 New Project		×
Steps	Select Project Name	and Folder
Choose Project Select Device Select Header Select Plugin Board Select Compiler	Project Name: Project Location:	avr-da-cnano-low-power-lab-interrupt C: Wy stuff Projects Browse
6. Select Project Name and Folder	Project Folder:	stuff\Projects\avr-da-cnano-low-power-lab-interrupt.X
MPLAB X IDE	Overwrite existing Also delete source Set as main proje Use project locati Encoding: ISC Project name and for Try shortening the p	g project. es. ct on as the project folder D-8859-1 Vider path length are nearing the Windows limit. This may cause issues during build or p roject name or path.
		<back next=""> Finish Cancel Help</back>

- 2. Open MPLAB Code Configurator (MCC) and configure the peripherals.
 - 2.1. In System Module, choose the clock source as the internal oscillator with 24 MHz clock.



Info: Choose the Clock source as Internal Oscillator, the Oscillator Frequency Options as the 24 MHz system clock, and disable the Prescaler.

Figure 1-8. System Module Configuration

System Module				
🌐 Easy Setup 📃 Registers				
Clock Control				
Main Clock(Hz):	24000000			
Clock Source :	Internal Oscillator	•		
Internal Oscillator Frequency:	1-32MHz internal oscillator	•		
Oscillator Frequency Options:	24 MHz system clock	*		
PLL Enable:				
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000			
Prescaler Enable:				
Prescaler:	6Х	~		
Olock Out Enable:				
Watchdog Timer				
► Brown-out Detector				
► Voltage Level Monitor				

2.2. In the **Registers** tab of System Module, enable Sleep and set the mode to Standby.



Info: Go to the **Registers** tab in System Module and scroll down to SLPCTRL. Modify the CTRLA register to enable Sleep and select the mode.

Figure 1-9. Sleep Configuration

System Module	
Image: Swrst disabled	
▼ SLPCTRL	
✓ Register: SLPCTRLCTRLA 0x3 ✓ SEN enabled ✓ ✓ SMODE Standby Mode ✓	
✓ Register: SLPCTRLVREGCTRL 0x0 ✓ PMODE AUTO ▼	

2.3. Add the VREF module from Device resources and configure it to provide 2.048V reference to the ADC.



Info: The ADC voltage reference must be configured to the internal 2.048V reference. Do not enable the Force ADC voltage option.

Info: The VREF is used to enable the temperature sensor of the microcontroller.

Figure 1-10. VREF Configuration

VREF	
ல் Easy Setup 📄 Registers	
 Software Settings 	
API Prefix:	VREF_0
▼ Hardware Settings	
Enable Force ADC Voltage Reference:	
ADC Voltage Reference:	Internal 2.048V reference
Enable Force DAC/AC Voltage Reference:	
OAC/AC Voltage Reference:	Internal 1.024V reference
Enable Force AC Voltage Reference:	
AC Voltage Reference:	Internal 1.024V reference

2.4. Add the ADC module from device resources and configure it in 12-bit mode, right-adjusted results, no accumulation, one tick for sample length, and to Run-In Standby (RUNSTBY).



Info: Right-shifted results option must be disabled. The sample length and sample accumulation number are selected in the **Hardware Settings** tab. The 12-bit mode is set by default. The RUNSTBY bit can be changed from the **Registers** tab and it is found in the CTRLA register.

AVR[®] DA Training Manual

Assignment 1: Event System Compared to Int...

Figure 1-11. ADC Configuration

.DC0				
錄 Easy Setup 📃 Registers				
 Software Settings 				
API Prefix:	ADC0			
Result Selection :	12-bit mode			
Oifferential Mode Conversion :	disabled			
😮 Left Adjust Result :				
 Hardware Settings 				
enable ADC:	\checkmark			
③ Sampling Frequency(Hz):	272727 ≤ 8 57142 ≤ 923076			
ADC Clock(Hz):	12000000			
③ Sample Accumulation Number:	No accumulation			
Sample Length (# of ADC Clock) :	0 ≤ 1 ≤ 31			



DC0
Easy Setup Registers Register: COMMAND 0x0
⊗ SPCONV disabled ▼
STCONV disabled -
▼ Register: CTRLA Ox81
CONVMODE disabled •
ENABLE enabled
PREERUN disabled
LEFTADJ disabled
RESSEL 12-bit mode
RUNSTBY enabled
▼ Register: CTRLB Ox0
SAMPNUM No accumulation

2.5. Add the RTC module from device resources and configure it to use the internal 32.768 kHz oscillator, have a period of one second, a prescaling factor 1, and Run-In Standby (RUNSTBY) activated.



Info: The clock, prescaling factor, and period are set from the **Hardware Settings** tab, while the RUNSTBY bit is found in the register pane in the CTRLA register.

Figure 1-13. RTC Configuration

RTC		
🛱 Easy Setup 📃 Registers		
 Software Settings 	^	
API Prefix:	RTC	
 Hardware Settings 		
⑦ Enable RTC:		
RTC Clock(Hz):	32768	
	Internal 32.768 kHz oscillator	
External Clock(Hz):	1 ≤ 32000 ≤ 32000	
Prescaling Factor:	RTC Clock / 1	
Ompare:	1 s ≤ 0 s ≤ 2 s	
Actual Compare:	0 s	
Period:	1s ≤ 1s ≤ 2s	
Actual Period:	1 \$	
Periodic Interrupt Timer		
 Interrupt Settings 		
Compare Match Interrupt Enable:		
Overflow Interrupt Enable:		

Figure 1-14. RUNSTBY RTC Configuration

гс		
Easy Setup	legisters 0x0	
▼ Register: CNT	0×0	
	0x81	
CORREN	disabled 💌	
PRESCALER	RTC Clock / 1	
RTCEN	enabled 💌	
RUNSTDBY	enabled 💌	
	RL _{Ox0}	
DBGRUN	sabled 💌	

2.6. Enable the Global Interrupts and enable the Result Ready Interrupt for the ADC and the overflow interrupt for the RTC.



Info: The Global Interrupts are enabled from the Interrupt Manager, the ADC interrupt is enabled from the **Interrupt Settings** tab of the ADC module, and the RTC interrupt is enabled from the **Interrupt Settings** tab of the RTC module (seen in the pictures for the previous steps).

AVR[®] DA Training Manual

Assignment 1: Event System Compared to Int...

Figure 1-15.	Interrupt	Manager	Configuration
--------------	-----------	---------	---------------

Interrupt Manager		0
ல் Easy Setup 📃 Registers		
 Interrupt Setting 		
Global Interrupt Enable:	\checkmark	
 Interrupt Priority 		
Round-robin Scheduling Enable:		
Interrupt Level Priority:	0	
Interrupt Vector with High Priority:	0	

2.7. Press the **Generate** button.

Figure 1-16. Generate Button

Projects	Files	Services	Resource Management [MCC] ×	
Tree View	Flat View			
Project F	Resource	Generat	e Import Export	
Interru	upt Manaq	er		^
Pin Mo	odule			
Systen	n Module			
 Peripher 	als			
🛞 🔀 쎣 ADCO				
🛞 🔀	🖥 📸 RTC			
@ 🔀	🗲 VRE	F		Ų

- 3. Add code to the generated files.
 - 3.1. Add the following code to the main.c file.

<pre>#include <avr io.h=""> #include <avr sleep.h=""> #include "mcc_generated_files/mcc.h"</avr></avr></pre>
<pre>void TIMER_interrupt(void);</pre>
<pre>uint16_t result;</pre>
<pre>int main(void) {</pre>
<pre>SYSTEM_Initialize(); RTC_SetOVFIsrCallback(TIMER_interrupt);</pre>
<pre>while (1) { sleep_cpu(); result = ADC0_GetConversionResult(); }</pre>
<pre>void TIMER_interrupt()</pre>
ADC0_StartConversion(ADC_MUXPOS_TEMPSENSE_gc) }

i

Info: SYSTEM_Initialize() is defined in mcc.c, RTC_SetOVFIsrCallback() is defined in rtc.c and ADC0_StartConversion() and ADC0 GetConversionResult() are defined in adc.c.

- SYSTEM_Initialize() sets all the configuration registers for the CPU and peripherals. The function is generated by MCC.
- RTC_SetOVFIsrCallback() is a function that sets the callback (the function that
 will be called when the interrupt is triggered) to the TIMER_interrupt() functioned
 defined in the main.
- The TIMER_interrupt() function starts the conversion for the ADC.
- The ADC0_StartConversion() function starts the conversion on the temperature channel of the ADC.
- The ADC0 GetConversionResult() returns the last result of the conversion.
- 3.2. In the pin_manager.c file found in <u>Source Files → MCC Generated files → src</u> replace the code for the direction of the pins with:

```
PORTA.DIR = 0xFF;
PORTB.DIR = 0xFF;
PORTC.DIR = 0x3F;
PORTD.DIR = 0xFF;
PORTE.DIR = 0xFF;
PORTF.DIR = 0xFF;
```



Info: This is done to prevent floating pins that will disturb the power readings.

3.3. Press the **Clean and Build** button from the toolbar and verify that the program builds without errors. **Figure 1-17. Clean and Build**



1.5 Creating the Event System Implementation Project



To do: Configure the hardware modules that will be used in this project and add code to the project.

- 1. Create a new MPLAB X project for AVR128DA48.
 - 1.1. Open MPLAB X.
 - 1.2. Select $\underline{File} \rightarrow New Project$ or the **New Project** button.

AVR[®] DA Training Manual Assignment 1: Event System Compared to Int...

Figure 1-18. New Project



1.3. Click **Next** (the Microchip Embedded Stand-alone Project is selected by default). **Figure 1-19. Project Type**

🗴 New Project		
Steps	Choose Project	
1. Choose Project	Q Filter:	
	Categories: Microchip Embedded Other Embedded Samples	Projects: Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project Import START MPLAB Project Import Atmel Studio Project
	Description:	
	Creates a new standalone application project.	n project. It uses an IDE-generated makefile to build your
	< B	ark Next > Finish Cancel Heln

In the Device field search for AVR128DA48. In the Tool category, select the Curiosity Nano board if it is connected to the computer, otherwise select None. Click Next.

© 2020 Microchip Technology Inc.

AVR[®] DA Training Manual

Assignment 1: Event System Compared to Int...

Figure 1-20. Device Selection

🔀 New Project		×
Steps	Select Device	
Choose Project Select Device Select Header Select Plugin Board Select Compiler	Family:	All Families
 Select Compiler Select Project Name and Folder 	Device:	AVR 128DA48 V
rouci	Tool:	AVR128DA48 Curiosity Nano-SN: MCH 🗸 🔲 Show All
MPLAB X IDE		
		< Back Next > Finish Cancel Help

1.5. Select XC8 2.20 compiler and click Next. Figure 1-21. Compiler Selection

🔀 New Project	×
Steps 1. Choose Project 2. Select Device 3. Select Header 4. Select Header 5. Select Compiler 6. Select Project Name and Folder	Select Compiler Compiler Toolchains
	< Back Next > Finish Cancel Help

1.6. Give a name to the project (and the location where to save) and click **Finish**.

Figure 1-22. Project Name

😰 New Project			×
Steps	Select Project Name and	l Folder	
 Choose Project Select Device Select Header 	Project Name:	avr-da-cnano-low-power-lab-evsys	
 Select Plugin Board Select Compiler 	Project Location:	C:\My stuff\Projects Browse	
6. Select Project Name and Folder	Project Folder:	My stuff\Projects\avr-da-cnano-low-power-lab-evsys.X	
	Overwrite existing pro	oject.	
	Also delete sources.		
	🗸 Set as main project		
	Use project location a	as the project folder	
MPLAB X IDE	Encoding: ISO-88	59-1 🗸	
		< Back Next > Finish Cancel Hel	D

- 2. Open MPLAB Code Configurator (MCC) and configure the peripherals.
 - 2.1. In System Module, choose the clock source as the internal oscillator with 24 MHz clock.



Info: Choose the Clock source as Internal Oscillator, the Oscillator Frequency Options as the 24 MHz system clock, and disable the Prescaler.

Figure 1-23. System Module Configuration

iystem Module				
🕃 Easy Setup 📃 Registers				
 Clock Control 				
Main Clock(Hz):	24000000			
Clock Source :	Internal Oscillator 💌			
Internal Oscillator Frequency:	1-32MHz internal oscillator 🔹			
Oscillator Frequency Options:	24 MHz system clock			
PLL Enable:				
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000			
Prescaler Enable:				
Prescaler:	6X			
Olock Out Enable:				
 Watchdog Timer 				
 Brown-out Detector 				
 Voltage Level Monitor 				

2.2. In the **Registers** tab of System Module, enable Sleep and set the mode to Standby.



Info: Go to the **Registers** tab in System Module and scroll down to SLPCTRL. Modify the CTRLA register to enable Sleep and select the mode.

Figure 1-24. Sleep Configuration

System Module	2
🛞 Easy Setup 📃 Registers	
SWRST disabled -	
▼ SLPCTRL	
✓ Register: SLPCTRLCTRLA 0x3	
▼ Register: SLPCTRL.VREGCTRL 0x0	
PMODE AUTO -	

2.3. Add the VREF module from Device resources and configure it to provide 2.048V reference to the ADC.



Info: The ADC voltage reference must be configured to the internal 2.048V reference. Do not enable the Force ADC voltage option.



Info: The VREF is used to enable the temperature sensor of the microcontroller.

Figure 1-25. VREF Configuration

VREF		?
같아 Easy Setup Easy Setup Registers		
 Software Settings 		
API Prefix:	VREF_0	
 Hardware Settings 		
Enable Force ADC Voltage Reference:		
ADC Voltage Reference:	Internal 2.048V reference 👻	
 Enable Force DAC/AC Voltage Reference: 		
OAC/AC Voltage Reference:	Internal 1.024V reference	
 Enable Force AC Voltage Reference: 		
AC Voltage Reference:	Internal 1.024V reference	

2.4. Add the ADC module from device resources and configure it in 12-bit mode, right-adjusted results, no accumulation, one tick for sample length and to Run-In Standby (RUNSTBY). Also enable the STARTEI bit so that a conversion starts when the event is received.



Info: The right-shifted results option must be disabled. The sample length and sample accumulation number are selected in the **Hardware Settings** tab. The 12-bit mode is set by default. The RUNSTBY bit can be changed from the **Registers** tab and is found in the CTRLA register. The STARTEI bit is found in the EVCTRL register in the **Registers** tab.

Figure 1-26. ADC Configuration

ADC0	[•
🛱 Easy Setup 📃 Registers		
Software Settings		<u> </u>
API Prefix:	ADC0	
Result Selection :	12-bit mode 🔹	
② Differential Mode Conversion :	disabled -	
② Left Adjust Result :		
 Hardware Settings 		
Parallel ADC:	\checkmark	
Sampling Frequency(Hz):	272727 ≤ 857142 ≤ 923076	
ADC Clock(Hz):	12000000	
Sample Accumulation Number:	No accumulation	
(2) Sample Length (# of ADC Clock) :	0 \$ 1 \$ 31	

Figure 1-27. RUNSTBY RTC Configuration

ADC0
Image: Setup Image: Registers ▼ Register: COMMAND 0x0 0x0
SPCONV disabled •
STCONV disabled -
▼ Register: CTRLA 0x81
CONVMODE disabled
In ENABLE enabled -
FREERUN disabled
RESSEL 12-bit mode
RUNSTBY enabled r
▼ Register: CTRLB 0x0
SAMPNUM No accumulation

AVR[®] DA Training Manual

Assignment 1: Event System Compared to Int...

Figure 1-28. Start Conversion on Event ADC	Configuration
--	---------------

ADCO
Easy Setup Registers
▼ Register: CTRLD 0x0
INITDLY Delay 0 CLK_ADC cycles
SAMPDLY 0x0
▼ Register: CTRLE 0x0
WINCM No Window Comparison 💌
▼ Register: DBGCTRL 0x0
DBGRUN disabled
▼ Register: EVCTRL 0x1
② STARTEI enabled ▼
▼ Register: INTCTRL 0x0
RESRDY disabled
WCMP disabled

2.5. Add the RTC module from device resources and configure it to use the internal 32.768 kHz oscillator, have a period of one second, prescaling factor 1 and Run-In Standby (RUNSTBY) activated.



Info: The clock, prescaling factor and period are set from the **Hardware Settings** tab, while the RUNSTBY bit is found in the register pane in the CTRLA register. Disable the Overflow Interrupt if it is enabled.

Figure 1-29. RTC Configuration

RTC	@
🛱 Easy Setup 📃 Registers	
 Software Settings 	^
API Prefix:	RTC
 Hardware Settings 	
② Enable RTC:	
RTC Clock(Hz):	32768
RTC Clock Source Selection:	Internal 32.768 kHz oscillator
External Clock(Hz):	1 ≤ 32000 ≤ 32000
Prescaling Factor:	RTC Clock / 1
Ompare:	1 s ≤ 0 s ≤ 2 s
Actual Compare:	0 s
Period:	1s ≤ 1s ≤ 2s
Actual Period:	1 s
Periodic Interrupt Timer	
 Interrupt Settings 	
Compare Match Interrupt Enable:	
Overflow Interrupt Enable:	

Assignment 1: Event System Compared to Int...

Figure 1-30. RUNSTBY RTC Configuration

RTC	
Easy Setup ■ Registers Register: CMP 0x0	^
▼ Register: CNT 0x0	
▼ Register: CTRLA _{0x81}	
CORREN disabled	
PRESCALER RTC Clock / 1	
RTCEN enabled	
RUNSTDBY enabled	
▼ Register: DBGCTRL 0x0	
DBGRUN disabled	

2.6. Add the EVSYS module. Configure channel 0 event generator to be RTC_OVF and the event that will be triggered ADC0START. Configure channel 1 event generator to be ADC0_RESRDY and the event that will be triggered EVSYSEVOUTC.

EVSYS							
😳 Easy Setup 📃 Registers							
 Software Settings 							
API Prefix:	EVSYS						
Event System Settings							
Event Generator	Channels						
		ADC0START	CCLLUT0A	CCLLUT0B	CCLLUT1A	CCLLUT1B	CCLLUT2A
RTC_OVF -	CHANNEL0	\checkmark					
ADC0_RESRDY -	CHANNEL1						
OFF -	CHANNEL2						
OFF -	CHANNEL3						
OFF -	CHANNEL4						
OFF •	CHANNEL5						
OFF -	CHANNEL6						
OFF -	CHANNEL7						
OFF -	CHANNEL8						
OFF -	CHANNEL9						
< (•	•					>

Figure 1-31. EVSYS Configuration

Figure 1-32. EVSYS Configuration

EVSYS								?
🍪 Easy Setup 目	Registers							
 Software Sett 	ings							
API Prefix:		EVSYS						
 Event System 	n Settings							
CCLLUT5A	CCLLUT5B	EVSYSEVOUTA	EVSYSEVOUTB	EVSYSEVOUTC	EVSYSEVOUTD	EVSYSEVOUTE	EVSYSEVOUTF	EVSYSEVOU
				\checkmark				

2.7. In Pin Manager: Grid View select **PC2** as EVOUTC for the EVSYS. **Figure 1-33. Pin Manager: Grid View**

Variables	Output		Noti	ifications [MC	c]	P	in M	anaç	er: 0	irid \	/iew	×																																		
Package:	QFN48	*		Pin No:	44	45	46	47	48	1	2	3	4	5	(7	8	9	1	0 11	1	2 13	3 1	6 1	17	18	19	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36	37	38	39	40
						_	-	Por	t A 🖲	,					Po	rt B	•					Po	rt C	•	-					i	Port	D 🔻		_	_		Por	t E 🔻			-	P	ort F	•	-	-
Module	Func	tion		Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	0	1	1	2 3	1	4 :	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6
ADC0	AINx		i	input																								î.	ĵu	ĵu	în I	ĥ	î.	ì	ĵ,	ĵ,	ĵ,	Ъ	Ъ	Ъ	ĵ,	ĵ,	ĵ,	î.	î.	
	EVOUTA	ι	0	output			ì					î.																																		
	EVOUTE		(output											1																															
E10010 -	EVOUTO	:	(output																	É	1				1	þ																			
EVSYS V	EVOUTE)	0	output																										î a					î.											
	EVOUTE		0	output																																		î.								
	EVOUTE		(output																																						î.				
	GPIO		i	input	ì	ì	ì	ì	j.	ì	j.	î.	j,	i î	ı î	î	i îs	ı î	ı î	ĥ	1	1	1	1	b '	b 1	b	î.	îa I	îa I	î.	ĥ	ì	ì	ì	ì	ì	î.	î.	î.	ĵ,	î.	î.	ì	ìa	î.
Pin Module V	GPIO			output	ì	î.	î.	î.	î.	î.	î.	î.	î.	î	î	î	n n	n n	ĥ	ĥ	n n	ı îs	ı î	1	6	in 1	è	în I	în I	în I	în I	î	ì	î.	î.	î.	ì	î.	î.	î.	î.	î.	în,	î	în.	î.
RSTCTRL	RESET		i	input																																										î.

2.8. In the Pin Module, enable the interrupt-on-change of the PC2 pin to sense both edges.

Figure 1-34. Pin Module Configuration

Pin Module								•	
승규 Easy Setup E Selected Package : C	Registers PN48								
Pin Name 🔺	Module	Function	Custom Name	OUTPUT	START HIGH	INVEN	PULLUPEN	ISC	
PC2	EVSYS	EVOUTC		\checkmark				Sense Both Edges	-

2.9. Enable Global Interrupts from Interrupt Manager.

AVR[®] DA Training Manual

Assignment 1: Event System Compared to Int...

Figure 1-35. Interrupt Manager Configuratio	Figure 1-35.	Interrupt	Manager	Configuration
---	--------------	-----------	---------	---------------

Interrupt Manager		0
😳 Easy Setup 📃 Registers		D
 Interrupt Setting 		
Global Interrupt Enable:	\checkmark	
 Interrupt Priority 		
Round-robin Scheduling Enable:		
Interrupt Level Priority:	0	
Interrupt Vector with High Priority:	0	

2.10. Press Generate.

Figure 1-36. Generate Button

Projects	Files	Services	Resource Management [MCC] ×	
Tree View	Flat View]		
Project I	Resource	es Generat	e Import Export	
▼ System				Â
Interro	upt Manag	er		
Pin M	odule			
Syster	n Module			
 Peripher 	als			
🛞 🔀	🖌 🕀 🕈	0		U
🛞 🔀	🛛 🛃 EVS	YS		
<u> </u>	2 🚔 ртс			~

- 3. Add code to the project.
 - 3.1. Add code to the main.c file:

<pre>#include "mcc_generated_files/mcc.h" #include <avr sleep.h=""></avr></pre>
<pre>void GPI0_Interrupt(void);</pre>
<pre>uint16_t volatile result;</pre>
<pre>int main(void) {</pre>
<pre>SYSTEM_Initialize(); ADC0.MUXPOS = ADC_MUXPOS_TEMPSENSE_gc; PORTC_PC2_SetInterruptHandler(GPI0_Interrupt);</pre>
while (1) {
<pre>sleep_cpu();</pre>
}
<pre>void GPI0_Interrupt(void) {</pre>
<pre>result = ADC0_GetConversionResult(); }</pre>

i

Info: SYSTEM_Initialize() is defined in mcc.c, ADC0_GetConverisonResult()
is defined in adc.c, PORTC_PC2_SetInterruptHandler() is defined in
pin manager.c.

- SYSTEM_Initialize() sets all the configuration registers for the CPU and peripherals. The function is generated by MCC.
- The register ADC0.MUXPOS is set in main because the Event System starts the conversion immediately and there is no way to give a channel at that time.
- PORTC_PC2_SetInterruptHandler(GPI0_Interrupt) sets the function that will be called when the interrupt-on-change on pin PC2 is triggered.
- The ADC0_GetConversionResult() returns the last result of the conversion.
- 3.2. In the pin_manager.c file found in <u>Source Files \rightarrow MCC Generated files \rightarrow src replace the code for the direction of the pins with:</u>

```
PORTA.DIR = 0xFF;
PORTB.DIR = 0xFF;
PORTC.DIR = 0x3F;
PORTD.DIR = 0xFF;
PORTE.DIR = 0xFF;
PORTF.DIR = 0xFF;
```



Info: This is done to prevent floating pins that will disturb the power readings.

3.3. Press the **Clean and Build** button from the toolbar and verify that the program builds without errors. **Figure 1-37. Clean and Build**



1.6 Programming the Device and Comparing Efficiency



To do: Load the project onto the microcontroller and get power readings.

- 1. Get power consumption readings from the software implementation program.
 - 1.1. Connect the Curiosity Nano board to the computer and the two soldered pins together.
 - 1.2. With the software implementation program set as the main program, press the **Make and Program Device** button from the toolbar. A pop-up might appear that asks to select the tool that will be used as a programmer. Select the AVR128DA48 Curiosity Nano board.

Figure 1-38. Make and Program the Device



1.3. Connect the Power Debugger to the computer. Connect the first pin (the one closest to the Nano embedded debugger) to the inward arrow of the A channel ammeter and the second pin (the one closer to the microcontroller) to the outward pointing arrow of the A channel ammeter.



Figure 1-39. Hardware Connections

- 1.4. In Atmel Studio, go to <u>Tools \rightarrow Data Visualizer</u>.
- 1.5. Two boxes will appear. Check the one called **DGI (Data Gateway Interface) Control Panel**. If this does not show up, the Configuration menu on the left side can be accessed, and in the <u>Modules \rightarrow </u> <u>External Connection</u> section, the Data Gateway Interface can be found.

Figure 1-40. Data Gateway Interface

Data	Visualizer 🕫 🗙	
Configuration	Configuration 2 Modules 2 Jals Gateway Interface (DGI) Sanal York > Valuation Visualization > Unitines > Protocols	DCI Control Fand Control Fand Control Fan
		Small Port Control Panel • *

1.6. Press **Connect** and a box labeled 'power' will appear. Select it and then press the wheel to open the menu. Disable the B channel.

 Δ warning In case the Power field does not appear, disconnect the board and connect it again.

Figure 1-41. Connect DGI

DGi Control Panel	^ ×
Power Debugger Data Gateway	Connect
7//4///08	Start
Autoletet routook	🛓 🗹 Reset MCU
Interfaces	

Assignment 1: Event System Compared to Int...

Figure 1-42. Power Settings

DGI Control Panel		* x
Power Debugger Data Gateway	Power Configuration ×	Disconnect
	Enable B Channel	Start
	Trigger calibration	ADP Logging Autodetect protocols V Reset MCU
Interfaces:	Enable Range Source	
□ SPI 🏠 □ USART 🏠 □ TWI 🏠 □ GPIO 🏠 🗹 Power 🛱 □ Code P	Profiling 🙀 Lock ChA to High Range 🗌	
Code Loca	ation 💕 Enable Voltage Output	
1 🌒 🖉 🖉 A Voltage 🌒	Voltage Output	
2 🌑 B Current 🌑	Averaging No *	
3 🌑 😐 🕑 B Voltage 🌑	OK Cancel	

1.7.

When everything is correctly configured, press **Start** and a new box that displays the power consumption in real time will appear.



Tip: Open the control panel on the left side of the box to adjust the display. Unchecking Show Zero will provide a clearer picture.

Figure 1-43. Start Monitoring



- 1.8. After the reading has stabilized, press **Stop** and then disconnect. The box with the power measurement will remain open.
- 2. Get power consumption readings from the Event System implementation program.
 - 2.1. Connect the two soldered pins on the Curiosity board together.
 - 2.2. With the Event System implementation project set as main program, press the **Make and Program Device** button on the toolbar in MPLAB X.

Figure 1-44. Make and Program the Device



- 2.3. Connect the Curiosity Nano Board to the Power Debugger. See Step 1.3 in this chapter for details and pictures.
- 2.4. Go back to the Data Visualizer in Atmel Studio and in the DGI panel press **Connect**.

Figure 1-45. Data Gateway Interface

(Gl Control Panel	^ X	l
Power Debugger Data Gateway	Connect	
24697699	Start	
ADPLogging Autodetect protocols	Reset MCU	
ntefaces		

2.5. Check the power box and edit the settings to disable the B channel.

Assignment 1: Event System Compared to Int...

Figure 1-46. Power Settings

DGI Control Panel							* ×
Power Debugger Data Gateway				8 Power Configuration	n	×	Disconnect
15020000566				Enable B Channel			Start
				Trigger calibration			ADP Logging Autodetect protocols V Reset MCU
Interfaces:				Enable Range Source			
🗆 SPI 🛱 🗌 USART 🛱 🗆 TWI 🙀 🗆	grio 🛱	Power 🛱	🗌 Code Profiling 🛱	Lock ChA to High Range			
	o 🧼 💿	A Current	Code Location	Enable Voltage Output			
1	1 🕬 💿	A Voltage 💕		Voltage Output	1600	mV	
2	2 🕬 🕚	B Current		Averaging	No	×	
3	s 🌑 💿	B Voltage 🌑		OK	Cancel		

2.6. Press **Start** and a new box that displays the real-time power consumption will appear. Disable Show zero for a better picture.



Figure 1-47. Start Monitoring

- 2.7. When the reading has stabilized, press **Stop** to have a fixed result.
- 3. Compare the two measurements.
 - 3.1. Both power consumption graphs are open on the same screen. They can now be analyzed to observe the way the programs behaved.

Figure 1-48. Software Implementation Compared to EVSYS

Ch A Current	1200μΑ - 1050μΑ - 900μΑ - 750μΑ - 600μΑ - 450μΑ - 300μΑ - 150μΑ - 0μΑ -																		1 1 1 1 1			 Control Panel
(* *	Ch A	rage Instant	while pressing and h	holding the left :	hift key to zoom	in the time axis	57213	5121.3	5726.0	2120.3	5729.0	5729.3	3730.0	3730.3	3/31.0	3731.3	5752.0	5732.3	3733.	573.3	57544	
Pow	ver Analysis																					^ X
Ch A Current	400μΑ - 340μΑ - 280μΑ - 220μΑ - 160μΑ - 100μΑ - 40μΑ - -20μΑ - -140μΑ - -200μΑ -	3007.5 39	08.0 3908	.5 390	1 1 1	9.5 3910	0.0 391	0.5 391	1.0 391	1.5 3912	.0 3912	2.5 3913	3.0 3913	1 1 1	.0 391/	1 1 1 1	5.0 39	15.5 3	916.0	1916.5 39	17.0	 Control Panel
ľ	Ch A	rage Instant	13.67]			5.0 - 591	0.5 591		1.5 5912	.0 3912							10.0 - 5	510.0	59		



Info: The top part displays the power consumption of the Event System implementation, while the bottom part displays the power consumption of the Interrupts implementation. As can be seen from the picture, the difference is fairly small of about 1.5 uA. This is due to the long amount of time the microcontroller spends in Sleep.

2. Assignment 2: Software Accumulation Compared to Hardware Accumulation for ADC

In this assignment, the power consumption of two different ADC accumulation techniques will be compared. Accumulating results in an ADC has the role of increasing the accuracy of the results and minimizing the effect of noise.

The first approach will trigger conversions again and again until 128 of these are completed. The results will be added to a variable that acts as an accumulator.

The second approach takes advantage of the built-in accumulator for the ADC. It will use the accumulated conversion mode of the ADC to run 128 conversions and add the results to the hardware accumulator.

2.1 Creating the Software Implementation Project



To do: Configure the hardware modules that will be used in this project and add code to the project.

- 1. Create a new MPLAB X project for AVR128DA48.
 - 1.1. Open MPLAB X.
 - 1.2. Select $\underline{File} \rightarrow \underline{New \ Project}$ or the **New Project** button. Figure 2-1. New Project

File Edit View Navigate



1.3. Click **Next** (Microchip Embedded Stand-alone Project is selected by default).

 \times

Assignment 2: Software Accumulation Compared ...

Creates a new standalone application project. It uses an IDE-generated makefile to build your

🔀 New Project		
Steps	Choose Project	
1. Choose Project	Q Filter:	
2	Categories:	Projects:
	Microchip Embedded	Standalone Project
	Other Embedded	Existing MPLAB IDE v8 Project
	i ⊡ i Samples	User Makefile Project
		Library Project
		Import START MPLAB Project
		Import Atmel Studio Project
	Description:	

Figure 2-2. Project Type

1.4. In the Device field search for: AVR128DA48. In the Tool category, select the Curiosity Nano board if it is connected to the computer, otherwise select None. Click Next.

project.

Figure 2-3. Device Selection

🔀 New Project		×
Steps	Select Device	
Choose Project Select Device Select Header	Family:	All Families
 Select Plugin Board Select Compiler Select Project Name and 	Device:	AVR 128DA48
Folder	Tool:	AVR 128DA48 Curiosity Nano-SN: MCH 🗸 🗌 Show All
		< Back Next > Finish Cancel Help

1.5. Select the **XC8 2.20 compiler** and click **Next**.

Figure 2-4. Compiler Selection

🔀 New Project		×
 Steps Choose Project Select Device Select Header Select Compiler Select Project Name and Folder 	Select Compiler Compiler Toolchains → XC8 → XC8 (v2.20) [C:\Program Files(\#icrochip\xc8\v2.20\bin] → XC8 (v2.10) [C:\Program Files (x86)\#icrochip\xc8\v2.05\bin] → XC8 (v2.05) [C:\Program Files (x86)\#icrochip\xc8\v2.05\bin] ⊕ -avrasm2 ⊕ -iAR for AVR ⊕ -pic-as	
	< Back Next > Finish Cancel Help	

1.6. Give a name to the project (and the location where to be saved) and click **Finish**. **Figure 2-5. Project Name**

🔀 New Proje	ct			Х
Steps		Select Project Name	and Folder	
 Choose P Select De Select He Select Plu 	roject vice ader igin Board	Project Name: Project Location:	avr-da-cnano-low-power-lab-interrupt C:Wy stuff\Projects Browse	
6. Select Co Folder	mplier roject Name and	Project Folder:	stuff\Projects\avr-da-cnano-low-power-lab-interrupt.X	
M		Overwrite existing Also delete source Set as main proje	g project. es. ct on as the project folder	
T	·/··	Encoding: ISC Project name and fo Try shortening the p	lder path length are nearing the Windows limit. This may cause issues during build roject name or path.	d or t
			< Back Next > Finish Cancel Help	

- 2. Open MPLAB Code Configurator (MCC) and configure the peripherals.
 - 2.1. Configure the System Module to run on the internal oscillator at 24 MHz.



Info: The clock source will be the internal oscillator and the frequency 24 MHz. The prescaler option must be disabled.

Figure 2-6. System Module Configuration

System Module		
🖏 Easy Setup 📃 Registers		
Clock Control		
Main Clock(Hz):	24000000	
Clock Source :	Internal Oscillator	
Internal Oscillator Frequency:	1-32MHz internal oscillator	
Oscillator Frequency Options:	24 MHz system clock	
PLL Enable:		
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000	
Prescaler Enable:		
Prescaler:	6X -	
Olock Out Enable:		
Watchdog Timer		
Brown-out Detector		
 Voltage Level Monitor 		

2.2. Add the VREF module from Device resources and configure it to provide 2.048V reference to the ADC.



Info: The ADC voltage reference must be configured to the internal 2.048V reference. Do not enable the Force ADC Voltage option.



Info: The VREF is used to enable the temperature sensor of the microcontroller.

Figure 2-7. VREF Configuration

VREF	
錄 Easy Setup 📃 Registers	
 Software Settings 	
API Prefix:	VREF_0
 Hardware Settings 	
Enable Force ADC Voltage Reference:	
ADC Voltage Reference:	Internal 2.048V reference
Enable Force DAC/AC Voltage Reference:	
DAC/AC Voltage Reference:	Internal 1.024V reference 🔹
Enable Force AC Voltage Reference:	
OR Voltage Reference:	Internal 1.024V reference 🔹

2.3. Add the ADC module from device resources and configure it in 12-bit mode, right-adjusted results, no accumulation and 31 ticks for sample length.



Info: Right-shifted results option must be disabled. The sample length is selected in the **Hardware Settings** tab. The 12-bit mode is set by default. The same is with the no accumulation selection.

Figure 2-8. ADC Configuration

ADC0		۲
🔅 Easy Setup 📄 Registers		
 Software Settings 		
API Prefix:	ADC0	
Result Selection :	12-bit mode 🔹	
Ø Differential Mode Conversion :	disabled 👻	
😮 Left Adjust Result :		
 Hardware Settings 		
	\checkmark	
(2) Sampling Frequency(Hz):	272727 ≤ 272727 ≤ 923076	
ADC Clock(Hz):	12000000	
② Sample Accumulation Number:	No accumulation	
Sample Length (# of ADC Clock) :	0 ≤ 31 ≤ 31	
Interrupt Settings		
Select Channels		
► Window Settings		

2.4. Press the **Generate** button.

AVR[®] DA Training Manual

Assignment 2: Software Accumulation Compared ...

Projects	Files	Services	Resource Management [MCC] ×	
Tree View	Flat View			
Project	Resource	Generat	e Import Export	
 System 				
Interr	upt Manag	er		
Pin M	lodule			
Syster	m Module			
▼ Periphe	rals			
🛞 👂	🛾 🕀 ADO	0		
(2)	🛛 🦟 VRE	F		

- 3. Add code to the project.
 - 3.1. Add the following code to the main.c file.



Info: SYSTEM_Initialize() is defined in mcc.c, ADC0_StartConversion(channel), ADC0_IsConversionDone(), and ADC0_GetConversionResult() are defined in adc0.c.

- SYSTEM_Initialize() is a function that configures the microcontroller and peripherals according to what was set in MCC. It is generated by MCC and needs to be called at the beginning of the program.
- ADC0_StartConversion(channel) starts an ADC conversion on the given channel
- ADC0_IsConversionDone() returns the status of the conversion: 0 for in progress and 1 for finished
- ADC0_GetConversionResult() returns the result of the last conversion
- The for loop repeats the steps for starting a conversion, waiting for it to end and adding the result to the accumulator 128 times

AVR[®] DA Training Manual Assignment 2: Software Accumulation Compared ...

3.2. In the pin_manager.c file found in <u>Source Files \rightarrow MCC Generated files \rightarrow src replace the code for the direction of the pins with:</u>

PORTA.DIR = 0xFF; PORTB.DIR = 0xFF; PORTC.DIR = 0x3F; PORTD.DIR = 0xFF; PORTE.DIR = 0xFF; PORTF.DIR = 0xFF;



Info: This is done to prevent floating pins that will disturb the power reading.

3.3. Press the **Clean and Build** button from the toolbar and verify that the program builds without errors. **Figure 2-10. Clean and Build**



2.2 Creating the Hardware Implementation Project



To do: Configure the hardware modules that will be used in this project and add code to the project.

- 1. Create a new MPLAB X project for AVR128DA48.
 - 1.1. Open MPLAB X.
 - 1.2. Select $\underline{File} \rightarrow \underline{New \ Project}$ or the **New Project** button. Figure 2-11. New Project

File Edit View Navigate



1.3. Click Next (Microchip Embedded Stand-alone Project is selected by default).

Assignment 2: Software Accumulation Compared ...

New Project	
teps	Choose Project
1. Choose Project 2	Categories: Projects: Image: Categories: Standalone Project: Image: C
	Description: Creates a new standalone application project. It uses an IDE-generated makefile to build your project.

----40

1.4. In the Device field, search for: AVR128DA48. In the Tool category, select the Curiosity Nano board if it is connected to the computer, otherwise select None. Click Next.

Figure	2-13	Device	Selection
iguic	Z -10.	DCVICC	0010011011

🔀 New Project		×
Steps	Select Device	
1. Choose Project 2. Select Device		
 Select Header Select Plugin Board 	Family:	All Families V
 Select Compiler Select Project Name and Folder 	Device:	AVR 128DA48 ~
	Tool:	AVR 128DA48 Curiosity Nano-SN: MCH 🗸 🗌 Show All
MPLAB X IDE		
		c Rade New Cancel Help

1.5. Select the XC8 2.20 compiler and click Next.

Figure 2-14. Compiler Selection

🔀 New Project		×
 Steps Choose Project Select Device Select Header Select Project Name and Folder 	Select Compiler Compiler Toolchains 	
	< Back Next > Finish Cancel Help	>

1.6. Give a name to the project (and the location where to be saved) and click **Finish**. **Figure 2-15. Project Name**

🔀 New Project		×			
Steps	Select Project Name	and Folder			
 Choose Project Select Device Select Header 	Project Name:	avr-da-cnano-low-power-lab-interrupt			
4. Select Plugin Board 5. Select Compiler 6. Select Project Name and Folder	Project Location: Project Folder:	C: Wy stuff Projects Browse :tuff Projects \avr-da-cnano-low-power-lab-interrupt.X			
	Overwrite existing	g project.			
	Also delete sourc	Also delete sources.			
	✓ Set as main project				
		on as the project folder			
MPLAB					
XIDE	_				
	Encoding: ISC	0-8859-1			
	Project name and for Try shortening the p	lder path length are nearing the Windows limit. This may cause issues during build or $\mathfrak g$ roject name or path.			
		<back next=""> Finish Cancel Help</back>			

- 2. Open MPLAB Code Configurator (MCC) and configure the peripherals.
 - 2.1. Configure the System Module to run on the internal oscillator at 24 MHz.



Info: The clock source will be the internal oscillator and the frequency 24 MHz. The prescaler option must be disabled.

Figure 2-16. System Module Configuration

System Module		?
😳 Easy Setup 📃 Registers		
Clock Control		
Main Clock(Hz):	24000000	
(?) Clock Source :	Internal Oscillator	
Internal Oscillator Frequency:	1-32MHz internal oscillator	
Oscillator Frequency Options:	24 MHz system clock	
PLL Enable:		
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000	
Prescaler Enable:		
Prescaler:	6X 👻	
Olock Out Enable:		
 Watchdog Timer 		
Brown-out Detector		
 Voltage Level Monitor 		

2.2. Configure the Sleep options so that Sleep is enabled, and the mode is Standby.



Info: Go to the **Registers** page of the System Module and scroll down until reaching SLPCTRL. In the SLPCTRL.CTRLA register, modify the options to be enabled and Standby mode.

Figure 2-17. Sleep Configuration

System Module	2
Image: Second	
▼ SLPCTRL	
▼ Register: SLPCTRLVREGCTRL _{0x0}	
PMODE AUTO -	

2.3. Add the VREF module from Device resources and configure it to provide 2.048V reference to the ADC.



Info: The ADC voltage reference must be configured to the internal 2.048V reference. Do not enable the Force ADC Voltage option.

Info: The VREF is used to enable the temperature sensor of the microcontroller.

Figure 2-18. VREF Configuration

VREF	0	
🔯 Easy Setup 📃 Registers		
 Software Settings 		
API Prefix:	VREF_0	
 Hardware Settings 		
Enable Force ADC Voltage Reference:		
ADC Voltage Reference:	Internal 2.048V reference	
Enable Force DAC/AC Voltage Reference:		
OAC/AC Voltage Reference:	Internal 1.024V reference	
Enable Force AC Voltage Reference:		
AC Voltage Reference:	Internal 1.024V reference	

2.4. Add the ADC module from device resources and configure it in 12-bit mode, right-adjusted results, 128 results accumulation, 31 ticks for sample length, and to Run-In Standby (RUNSTBY).



Info: The right-shifted results option must be disabled. The sample length and sample accumulation number are selected in the **Hardware Settings** tab. The 12-bit mode is set by default. The RUNSTBY bit can be changed from the register view and is found in the CTRLA register.

AVR[®] DA Training Manual Assignment 2: Software Accumulation Compared ...

Figure 2-19. ADC Configuration

ADC0	2 🛞
錄 Easy Setup 📃 Registers	
 Software Settings 	
API Prefix:	ADCO
Result Selection :	12-bit mode 🔹
Ø Differential Mode Conversion :	disabled 🔹
😮 Left Adjust Result :	
▼ Hardware Settings	
Parallel ADC:	\checkmark
(2) Sampling Frequency(Hz):	272727 ≤ 272727 ≤ 923076
ADC Clock(Hz):	12000000
③ Sample Accumulation Number:	128 results accumulated
Sample Length (# of ADC Clock) :	0 ≤ 31 ≤ 31
 Interrupt Settings 	
Result Ready Interrupt Enable:	\checkmark
WCMP Interrupt Enable:	
Select Channels	
Window Settings	

Figure 2-20. Run in Standby ADC Configuration

ADCO	?
🛞 Easy Setup 📃 Registers	
▼ ADC0	í
Interrupt Enables	
RESRDY	
WCMP	
▼ Register: COMMAND Ox0	
SPCONV disabled -	
STCONV disabled •	
▼ Register: CTRLA 0x81	
CONVMODE disabled -	
ENABLE enabled *	
🚱 FREERUN disabled 👻	
🛞 LEFTADJ disabled 👻	
🛞 RESSEL 12-bit mode 👻	
RUNSTBY enabled •	
▼ Register: CTRLB 0x7	
⊗ SAMPNUM 128 results accumulated ▼	

2.5. Enable the Result Ready Interrupt for the ADC and the Global Interrupts.

AVR[®] DA Training Manual Assignment 2: Software Accumulation Compared ...



Info: In the ADC0 module, in the Interrupt Setting, activate Result Ready Interrupt Enable. This setting can be observed in the figure for the previous step. In the Interrupt Manager, enable Global Interrupt.

Figure 2-21. Interrupt Manager Configuration

Interrupt Manager		2
🔅 Easy Setup 📃 Registers		
 Interrupt Setting 		î
Global Interrupt Enable:	\checkmark	
 Interrupt Priority 		
Round-robin Scheduling Enable:		
Interrupt Level Priority: Interrupt Vector with High Priority:	0	

2.6. Press the **Generate** button.

Figure 2-22. Generate Button

Projects	Files	Services	Resource M	anagement [MCC] ×	
Tree View	Flat View				
Project F	Resource	Generat	e Import	Export	
▼ System					
Interru	upt Manag	er			
Pin M	odule				
Syster	n Module				
▼ Peripher	als				
🛞 🔁	🛾 🔦 ADO	:0			
@ <mark>></mark>	🛛 🦟 VRE	F			

- 3. Add code to the project.
 - 3.1. Add the following code to the main.c file:

<pre>#include "mcc_generated_files/mcc.h" #include <avr sleep.h=""></avr></pre>
<pre>uint16_t result;</pre>
<pre>int main(void) {</pre>
<pre>SYSTEM_Initialize(); while (1) { ADCO_StartConversion(ADC_MUXPOS_TEMPSENSE_gc); sleep_cpu(); result = ADC0_GetConversionResult(); }</pre>
}



Info: SYSTEM_Initialize() is defined in mcc.c,
ADC0_StartConversion(channel) and ADC0_GetConversionResult() are
defined in adc0.c.

- SYSTEM_Initialize() is a function that configures the microcontroller and peripherals
 according to the settings in MCC. It is generated by MCC and needs to be called at the
 beginning of the program.
- ADC0_StartConversion(channel) starts an ADC conversion on the given channel.
- ADC0 GetConversionResult() returns the result of the last conversion.
- The program initiates a conversion and then enters Sleep. It is woken up by the Result Ready Interrupt, reads the result and then starts a new one.
- 3.2. In the pin_manager.c file found in <u>Source Files \rightarrow MCC Generated files \rightarrow src, replace the code for the direction of the pins with:</u>

PORTA.DIR PORTB.DIR PORTC.DIR PORTD.DIR PORTE.DIR	= = = =	0xFF; 0xFF; 0x3F; 0xFF; 0xFF;
PORTE.DIR PORTF.DIR	=	OxFF; OxFF;



Info: This is done to prevent floating pins that will disturb the power readings.

3.3. Press the **Clean and Build** button from the toolbar and verify that the program builds without errors. **Figure 2-23. Clean and Build**



2.3 **Programming the Device and Comparing Power Efficiency**



To do: Load the project onto the microcontroller and get power readings.

- 1. Program the device and get power consumption readings from the software implementation program.
 - 1.1. Connect the Curiosity Nano board to the computer. Connect the two soldered pins together. With the software accumulation program set as main program, press the **Make and Program Device** button from the toolbar. A pop-up might appear that asks to select the tool that will be programmed. Select the AVR128DA48 Curiosity Nano board.

Figure 2-24. Make and Program the Device



1.2. Open Atmel Studio and connect the Power Debugger to the computer. Connect the Curiosity Nano board to the channel A ammeter of the Power Debugger. Connect the pin closest to the Nano embedded debugger to the inward pointing arrow of the A ammeter and the pin closest to the microcontroller to the outward pointing arrow of the ammeter.

AVR[®] DA Training Manual

Assignment 2: Software Accumulation Compared ...

Figure 2-25. Hardware Setup



- 1.3. In Atmel Studio, go to <u>*Tools* \rightarrow *Data Visualizer*</u>.
- 1.4. Two boxes will appear. Check the one called DGI (Data Gateway Interface) Control Panel. If this does not show up, the Configuration menu on the left side can be accessed and, in the <u>Modules \rightarrow External Connection</u> section, the Data Gateway Interface can be found.

Figure 2-26. Data Gateway Interface

Data	Visualizer + ×		•
\odot	Configuration	DGI Control Panel	^ X
Configurat	Modules 2 External Connection Data Gateway Interface (DGI)	Curiosity Data Gateway Interface	 ✓ Connect ✓ Start
ğ	Senal Port		△ ADP Logging 🖌 Autodetect protocols 🔄 Show Config search path 🖌 Reset MCU
	 Utilities Protocols 	Interfaces:	
1		Serial Port Control Panel	• x

1.5. Press **Connect** and a box labeled 'power' will appear. Select it and then press the wheel to open the menu. Disable the B channel.

Awarning In case the Power field does not appear, disconnect the board and connect it again.

Figure 2-27. DGI Connect

0GI Control Panel		
Power Debugger Data Gateway	Connect	
246000099	Start	
AD2_LoggingAutodetect]	protocols 🔽 Reset MCU	
Interfaces:		

Assignment 2: Software Accumulation Compared ...

Figure 2-28. Power Settings

DGI Control Panel	A 3
Power Debugger Data Gateway	Power Configuration X Disconnect
75420000386	Enable 8 Channel Start
	Trigger calibration
Interfaces:	Enable Range Source
□ SPI 🏠 □ USART 🏠 □ TWI 🏠 □ GPIO 🏠 🗹 Power 🏠 □ Code Profiling 🏠	Lock ChA to High Range
Code Location	Enable Voltage Output
1 🌒 A Voltage 🍘	Voltage Output 1600 mV
2 🌒 B Current 🌍	Averaging No *
3 🌑 🖉 B Voltage 🌑	OK Cancel

1.6.

When everything is correctly configured, press **Start** and a new box that displays the power consumption in real time will appear.



Tip: To adjust the display, open the control panel on the left side of the box. Unchecking Show zero will provide a clearer picture.

Figure 2-29. Start Monitoring

Power Debugger Data Gateway	Disconnect
	ADP Logging Autodetect protocols V Reset MCL
Interfaces:	
□ SPI 🏠 □ USART 🏠 □ TWI 🏠 □ GPIO 🏠 🗹 Power 🛱	🗌 Code Profiling 🔅
Current 💕	Code Location
1 🌑 😐 🖉 A Voltage 🌑	
2 💕 💿	
3 🌑 🖉	
Power Analysis	×*
6360µA - 6300µA - 1900µA - 1900µA - 1900µA - 1910µA	Come Par Control Control Contr
14139 14140 14141 1 (Ch A Window Average Instant Window Average Instant Instan	4142 14143 14144 14145 14146 14147 14148

- 1.7. After the reading has stabilized, press **Stop** and then disconnect. The box with the power measurement will remain open.
- 2. Get power consumption readings from the hardware accumulation implementation program.
 - 2.1. Connect the two soldered pins on the Curiosity board together.
 - 2.2. With the hardware accumulation project set as main program, press the **Make and Program Device** button on the toolbar in MPLAB X.

Figure 2-30. Make and Program the Device



- 2.3. Connect two soldered pins of the Curiosity Nano Board to the Power Debugger. See Step 1.2.
- 2.4. Go back to the Data Visualizer in Atmel Studio and in the DGI panel press **Connect**.

Figure 2-31. DGI Connect

Power Dehugger Data Gateway	
Power Debugger Data Gateway	
laddottidae 3ta	rt
🗆 AD2.Logging 🗋 Autodetect protocols 🧭 Reset	t MCU
Interfaces	

2.5. Check the power box and edit the settings to disable the B channel.

Assignment 2: Software Accumulation Compared ...

Figure 2-32. Power Settings

DGI Control Panel		^ x
Power Debugger Data Gateway	Power Configuration ×	Disconnect
	Enable B Channel	Start
	Trigger calibration	ADP Logging Autodetect protocols V Reset MCU
Interfaces:	Enable Range Source	
□ SPI 🏠 □ USART 🏠 □ TVI 🏠 □ GPIO 🏠 🗹 Power 🏠 □ Code Profilin	🔯 Lock ChA to High Range 🗌	
Code Location (Enable Voltage Output	
1 🌓 💿 A Voltage 🛑	Voltage Output 1600 mV	
2 🌓 B Current 🌍	Averaging No *	
3 🌑 B Voltage 🌑	OK Cancel	

2.6. Press Start and a new box that displays the real-time power consumption will appear. Disable Show zero for a better picture.

Figure 2-33. Start Monitoring ugger Data Gateway wer De Start ADP Logging 🗆 SPI 🛱 🗆 USART 🛱 🗆 TWI 🧔 🗆 gpio 🛱 Power 🔯 🛛 Code Profiling 🛱 A Current -**C**... 0 A Voltage **6**--- 0 **6** 0 14321 14322 14323 14324 14325 14326 14323

- 2.7. When the reading has stabilized, press Stop to have a fixed result.
- 3. Compare the two measurements.
 - 3.1. Both power consumption graphs are open on the same screen. They can now be analyzed to observe the way the programs behaved.

Figure 2-34. Software Accumulation Compared to Hardware Accumulation

 Aµ0666 Aµ06666 Aµ0666 Aµ0666 Aµ0666	oekskounske Ngelplanske	iredulumete Wowingspelfe	unalistatista Sipleritetatista	na an a	an, de fordels Generalis	histochedro (999) 1994 - Andre Marine 1994 - Andrea Marine	stonna fallars dalata da lab	un pentri penent Matemi kacimat	a alkinetakynas 1944. se konstan	nden ostanado Kalenden fiste	Control Panel
6090µA –	14120	14140	14141	14142	14142	14144	14145	14146	14147	14149	
Ch A Window Aver	rage Instant		74 [4]	14142	14145	14144	[4]43	14140	14147		
ସ୍ ସ୍ 🔬	roll the mouse-wheel wh	ie pressing and holding the left	shift key to zoom in the time axi	s 💶							
Power Analysis											^ ×
3800μA – 3700μA – 3600μA – 3500μA – 3400μA – 3300μA – 3200μA – 3200μA – 3100μA –		nder för ender i det konge		an la constant a transmission de la constant a constant a constant a constant a constant a constant a constant		na hai kara sa na			and the following the factor of the factor o	philitiki katelon terinterin Angeneratikan	 Control Panel
3000μΑ -											
Ch A Window Aver	rage Instant	14320	14321	14322	14323	14324	14325	14326	14327	14328	
ସ୍ ସ୍ 🗴	roll the mouse-wheel wh	le pressing and holding the left	shift key to zoom in the time axi	s 🔳							

AVR[®] DA Training Manual Assignment 2: Software Accumulation Compared ...



Info: The top part displays the software accumulation, while the bottom part, the hardware accumulation. As can be seen from the picture, there is a 3.1 mA difference between the two, making the hardware implementation almost twice as efficient.

3. Revision History

Revision	Date	Description
Α	8/2020	Initial document release

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- General Technical Support Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- · Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- · Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code
 protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly
 evolving. We at Microchip are committed to continuously improving the code protection features of our products.
 Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act.
 If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue
 for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2020, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-6615-4

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.



Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	Australia - Sydney	India - Bangalore	Austria - Wels
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	China - Beijing	India - New Delhi	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	Denmark - Copenhagen
Fax: 480-792-7277	China - Chengdu	India - Pune	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	China - Chongqing	Japan - Osaka	Finland - Espoo
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	China - Dongguan	Japan - Tokyo	France - Paris
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	China - Guangzhou	Korea - Daegu	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	Germany - Garching
Fax: 678-957-1455	China - Hangzhou	Korea - Seoul	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	Germany - Haan
Tel: 512-257-3370	China - Hong Kong SAR	Malaysia - Kuala Lumpur	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	Germany - Heilbronn
Westborough, MA	China - Nanjing	Malaysia - Penang	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	Germany - Karlsruhe
Fax: 774-760-0088	China - Qingdao	Philippines - Manila	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	Germany - Munich
Itasca. IL	China - Shanghai	Singapore	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	China - Shenvang	Taiwan - Hsin Chu	Germany - Rosenheim
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	China - Shenzhen	Taiwan - Kaohsiung	Israel - Ra'anana
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Fax: 972-818-2924	China - Suzhou	Taiwan - Taipei	Italy - Milan
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	China - Wuhan	Thailand - Bangkok	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	Italy - Padova
Houston, TX	China - Xian	Vietnam - Ho Chi Minh	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	Netherlands - Drunen
Indianapolis	China - Xiamen		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	China - Zhuhai		Norway - Trondheim
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			Poland - Warsaw
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			Romania - Bucharest
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			Spain - Madrid
Tel: 951-273-7800			Tel: 34-91-708-08-90
Raleigh, NC			Fax: 34-91-708-08-91
Tel: 919-844-7510			Sweden - Gothenberg
New York, NY			Tel: 46-31-704-60-40
Tel: 631-435-6000			Sweden - Stockholm
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			UK - Wokingham
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			
Fax: 905-695-2078			