SOURCE

Tutorial

CIP Statemachine - Arduino

© 2012 actifsource GmbH, Switzerland - all rights reserved.

Tutorial	Actifsource Tutorial – CIP Statemachine - Arduino		
Required Time	120 Minutes		
Prerequisites	 Actifsource Tutorial – Installing Actifsource Actifsource Tutorial – Simple Service Actifsource Tutorial – CIP Statemachine - Lamp 		
Goal	Compiling and running the CIP Statemachine on the Arduino UNO		
Topics covered	 Arduino SDK AVR Eclipse Plugin Simple Arduino project CIP Arduino project 		
Notation	 To do Information Bold: Terms from actifsource or other technologies and tools <u>Bold underlined</u>: actifsource Resources <u>Underlined</u>: User Resources <u>UnderlinedItalics</u>: Resource Functions Monospaced: User input Italias: Important terms in current situation 		
Disclaimer	The authors do not accept any liability arising out of the application or use of any information or equipment described herein. The information contained within this document is by its very nature incomplete. Therefore the authors accept no responsibility for the precise accuracy of the documentation contained herein. It should be used rather as a guide and starting point.		
Contact	actifsource GmbH Täfernstrasse 37 5405 Baden-Dättwil Switzerland <u>www.actifsource.com</u>		
Trademark	actifsource is a registered trademark of actifsource GmbH in Switzerland, the EU, USA, and China. Other names appearing on the site may be trademarks of their respective owners.		
Credits	Francesco Rigoni: Arduino development with Eclipse – A step by step tutorial to the basic setup		

Overview

- Arduino SDK
 - o The Arduino SDK contains all the necessary source code for your Arduino
 - Installing the com port driver for your Arduino
- AVR Eclipse Plugin
 - The AVR Plugin is an extension to the C/C++ Development Toolkit to support development for the Atmel AVR series of embedded processors.
 - o Configure the AVR Plugin for your Arduino board
- Simple Arduino project
 - Setup a new C/C++ project
 - Write a very simple code which switches an LED on and off
 - o Compile the project including the Arduino core
 - o Download to the Arduino target and run
- CIP Arduino project
 - o Modeling a reactive state machine using the CIP method
 - o Generating C code for the state machine
 - o Connecting the generated state machine code to the Arduino I/O
 - o Downloading and testing to the Arduino board

Part 0: Installing Eclipse

- ✤ Install the Eclipse IDE for C/C++ Developers (C/C++ Development Tool)
 - o <u>http://www.eclipse.org/downloads/</u>
- ✤ Install the Actifsource Enterprise plugin

Part 1: Arduino SDK

- The Arduino SDK contains all the necessary source code for your Arduino
- Installing the com port driver for your Arduino

Arduino SDK



- Source and the second s
- 🖖 Unzip the Arduino SDK on your hard disk
 - o We use C:\arduino-1.0.3\ in this example

Arduino SDK

Geräte-Manager		κ
Datei Aktion Ansicht ?		
Akkus Akkus Andere Geräte Andere Geräte Anschlüsse (COM & LF Anschlüsse (COM & LF Audio-, Video- und Ga Bildverarbeitungsgerät Bildverarbeitungsgerät HP CLJ CM2320 Sca Integrated Webcan Computer Computer DVD/CD-ROM-Laufwerke	Treibersoftware aktualisieren Deaktivieren Deinstallieren Nach geänderter Hardware suchen Eigenschaften	
Startet den Treibersoftwareupdate-A	ssistenten für das ausgewählte Gerät.	

- ♥ Plugin your Arduino to a USB port
 - $\circ \quad \text{The installation will fail} \\$
- $\,\, {\ensuremath{\circlearrowright}}\,$ Open the Device Manager from the Control Panel
- ♥ Install new driver
- ♦ Specify C:\arduino-1.0.3\drivers as the place to look for the driver
 - o Warning: You must not select C:\arduino-1.0.3\drivers\FTDI USB Drivers

Arduino SDK



- ♦ Check if the driver has been installed correctly
- ✤ Important: Remember the com port
 - o COM3 in this example

```
uno.name=Arduino Uno
uno.upload.protocol=arduino
uno.upload.maximum_size=32256
uno.upload.speed=115200
uno.bootloader.low_fuses=0xff
uno.bootloader.high_fuses=0xde
uno.bootloader.extended_fuses=0x05
uno.bootloader.path=optiboot
uno.bootloader.file=optiboot_atmega328.hex
uno.bootloader.unlock_bits=0x3F
uno.bootloader.lock_bits=0x3F
uno.bootloader.lock_bits=0x0F
uno.build.mcu=atmega328p
uno.build.f_cpu=1600000L
uno.build.core=arduino
uno.build.variant=standard
```

- Open the boards.txt file in C:\arduino-1.0.3\hardware\arduino
- ✤ Print the information for your hardware platform
 - o Arduino Uno in this example

Part 2: AVR Eclipse Plugin

- The AVR Plugin for Eclipse is an extension to the C/C++ Development Toolkit to support development for the Atmel AVR series of embedded processors.
- Configure the AVR Plugin for your Arduino board

Install
Available Software
Select a site or enter the location of a site.
Work with: type or select a site
Find more software by working with the <u>"Available Software Sites</u> " preferences.
type filter text
Name Version
1 There is no site selected.
Add Repository
Select Name: avr-eclipse
Details Location: http://avr-eclipse.sourceforge.net/updatesite Archive
✓ Show o Cancel
Contact all undate sites during install to find required software
? < Back Next > Finish Cancel

- Start your Eclipse with a new workspace
 - o workspace_arduino in this example
- ✤ In your Eclipse, click Help/Install new Software...
- ✤ Add the AVR-Eclipse plugin update site
 - o http://avr-eclipse.sourceforge.net/updatesite
- Installing the plugin requires eclipse to restart

Preferences				
type filter text	Paths		< ▼	⇔ ▼ ▼
> General > Actifsource	Path Settings for the AVR Eclipse	e Plugin		
> Ant	Disable search for system pat	ths at start	up	
AVRDude	Note: If disabled, a manual reso	an may be	e required when a new avr-gcc toolchain has been instal	led.
Paths =	1			
> C/C++	Path to	Source	Current value	Edit
> Install/Update	AVR-GCC GNU make	Custom	C:\arduino-1.0.3\hardware\tools\avr\bin	Rescan
> Java	AVR Header Files	Custom	C:\arduino-1.0.3\hardware\tools\avr\atris\bin	
> Maven	AVRDude	Custom	C:\arduino-1.0.3\hardware\tools\avr\bin	
> Mylyn	Atmel Part Description Files	System		
> Plug-in Developm				
Server			Restore Defaulte	
			Restore Delauits A	piy
?			OK	cel

- ✤ Configure the AVR Eclipse Plugin under Windows/Preferences/AVR/Paths
- Set AVR-GCC to custom value C:\arduino-1.0.3\hardware\tools\avr\bin
- Set GNU make to custom value C:\arduino-1.0.3\hardware\tools\avr\utils\bin
- Set SVR-GCC to custom value C:\arduino-1.0.3\hardware\tools\avr\avr\include
- Set SVR-GCC to custom value C:\arduino-1.0.3\hardware\tools\avr\bin
- Important: Press Apply

12

Preferences		
type filter text	AVRDude	⇔ ▼ ⇔ ▼ ▼
General Actifsource Ant	AVRDude Global Settings Log internal AVRDude output to console	
AVR AVRDude Paths =	Use custom configuration file for AVRDude AVRDude config file C:\arduino-1.0.3\hardware\tools\avr\etc\avrdude.conf	Browse
C/C++	Programmer configurations	
Help Install/Update	Configurati Description	Add
Java		Edit
Maven Mylyn		Remove
Plug-in Developm Run/Debug		
Server ▼ ∢ III ►	Restore Defa	Apply
?	ОК	Cancel

- ✤ Configure the AVR Eclipse Plugin under Windows/Preferences/AVR/AVRDude
- Use AVRDude config file C:\arduino-1.0.3\hardware\tools\avr\etc\avrdude.conf
- Important: Press Apply

Preferences					_ D X
type filter text	AVRDude				⇔ ▼ ⇔ ▼ ▼
General Actifsource	AVRDude Globa	al Settings AVRDude output	to console		
AVRDude Paths =	AVRDude config	g file C:\arduino	-1.0.3\hardware\tools	avr\etc\avrdude.conf	<u>B</u> rowse
C/C++	Programmer co	nfigurations			
Help Install/Update	Configurati	Description			Add
Java					Edit
Maven Mylyn					Remove
Plug-in Developm Run/Debug					
Server T				Restore <u>D</u> efaults	Apply
?				ОК	Cancel

- ✤ Configure the AVR Eclipse Plugin under Windows/Preferences/AVR/AVRDude
- Add a new Programmer configuration
- ♦ If this is not working you probably forget to press **Apply** when configuring the AVR/Paths

14

Edit AVRDude Programmer Configuration New Configuration				
Configuration name	ActifsourceArduino			
Description	Arduino AVRDude Progr	rammer Configuration		
Programmer Hardware (-c)				
ABCmini Board, aka Dick Smith HC Altera ByteBlaster Amontec JTAGKey, JTAGKey-Tiny a Arduino	TCHIP	Programmer details from [C:\arduino-1.0.3\hardware\tool id = "arduino"; desc = "Arduino"; type = arduino;		
Athel AppNote AVR109 Boot Load	der			
Override default port (-P)	СОМЗ			
Override default baudrate (-b)	115200 🗸			
 /Reset Line ● restore to previous state ● activated (-E reset) ● deactivated (-E noreset) 		Vcc Lines • restore to previous state • activated (-E vcc) • deactivated (-E novcc)		
Delay between avrdude invocations	milliseconds			
Command line preview	avrdude -carduino -PCO	M3 -b115200 [part specific options]		
?		OK Cancel		

- Select the Programmer Hardware as found in the boards.txt file in uno.upload.protocol
 - o Arduino In this example
- b Select the com port that you remembered when installing the driver in the Device Manager
 - o COM3 In this example
- ♦ Select the baud rate as found in the boards.txt file in uno.upload.speed
 - o 115200 In this example

Part 3: Simple Arduino p<u>roject</u>

- Setup a new C/C++ project
- Write a very simple code which switches an LED on and off
- Compile the project including the Arduino core
- Download to the Arduino target and run

	New	۱.	C2	Project	
	Show In	Alt+Shift+W	C	C Project	
	Copy	Ctrl+C	C‡	C++ Project	
	Copy Qualified Name	curre -	C2	Example	
Ê	Paste	Ctrl+V		Other	Ctrl+N
ж	Delete	Delete			
2	Import				
4	Export				
\$	Refresh	F5			
<u>_</u>	Actifsource Folders	+			
ۥ	Import to Actifsource				

- ✤ Create a new C++ Project
- ① You have to create a C++ project even if you plan to write C code since we have to compile the Arduino Core
- (i) If you can't create a C/C++ Project you have probably not downloaded the Eclipse for C/C++ Developers
 - Install the CDT plugin from http://www.eclipse.org/cdt/

C++ Project	
C++ Project Create C++ project of selected type	
Project name: SimpleProject	
Use default location Location: C:\Projects\workspace_arduino\SimpleProject	Browse
Project type: Toolchains: AVR Cross Target Application Empty Project AVR Cross Target Static Library Executable Shared Library Static Library Makefile project Show project types and toolchains only if they are supported on the platform	
? < Back Next > Finish	Cancel

- Create a new C++ project named SimpleProject
- ♦ Choose AVR Cross Target Application
- ♦ Choose the AVR-GCC Toolchain

C++ Project	
Select Configurations Select platforms and configurations you wish to deploy on	
Project type:AVR Cross Target ApplicationToolchains:AVR-GCC ToolchainConfigurations:	
Image: Selease Image: Selease	Select all Deselect all
	Advanced settings
Use "Advanced settings" button to edit project's properties. Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on property page	25.
? < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel

- ✤ Do not create a Debug Version
- ① The debug code is too large to fit on the Arduino Uno
- See boards.txt for the maximum code size
 - o uno.upload.maximum_size=32256 for this example
- ♥ Click Next

C++ Project		
AVR Target Hardw Define the AVR targe	r are Properties It properties	
MCU Type: MCU Frequency (Hz):	ATmega328P 16000000	
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

- Specify the MCU Type as found in your boards.txt (uno.build.mcu)
 - o atmega328p for this example
- Specify the MCU Frequency as found in your boards.txt (uno.build.f cpu)
 - o 1600000L for this example
 - Enter the number without the L
- Sclick Finish
- ♥ Open the C/C++ Perspective if you are asked or open it manually in the upper right corner



Properties for SimplePro	ject 📃 🗖 🗙
type filter text	AVRDude $\Leftrightarrow \checkmark \Rightarrow \checkmark$
Resource AVR AVRDude Target Hardware Builders C/C++ Build C/C++ General Coverage Project Facets Project References Run/Debug Settings Server Task Repository Task Tags Validation WikiText	Configuration: Release [Active] Programmer Flash / EEPROM Fuses Lockbits Advanced Other Programmer configuration Edit. New JTAG ICE BitClock Specify the bit clock period in microseconds for the JTAG interface or the ISP clock (JTAG ICE only). Set this to > 1.0 for target MCUs running with less than 4MHz on a JTAG ICE. Leave the field empty to use the preset bit clock period of the selected Programmer. JTAG ICE bitclock Specify the delay in microseconds for each bit change on bitbang-type programmers. Set this when the the host system is very fast, or the target runs off a slow clock Leave the field empty to run the ISP connection at max speed. Bit state change delay Is state change delay
?	OK Cancel

- ✤ For the newly created SimpleProject, choose Properties (Mouse-Click-Right on the project)
- Select AVR/AVRDude
- 🗞 In the Programmer Tab, select the previously created Programmer Configuration ActifsourceArduino
- ♥ Click OK

Properties for SimpleProj	ject
type filter text	C/C++ Build 🗢 👻 🔿 💌 💌
Resource	
AVR	Configuration: Release [Active]
Builders	Conngulation. Release [Active]
Build Variables	
Discovery Options	E Builder Settings Behaviour Refresh Policy
Environment	Build settings
Logging	Stop on first build error Use parallel build
Settings Tool Chain Editor	Use optimal jobs number
C/C++ General	Ose parallel Jobs: 1
Coverage	Workbench Build Behavior
Project Facets	Workbench build type: Make build target:
Project References	✓ Build on resource save (Auto build) all
Server	Note: See Workbench automatic build preference
Task Repository	Build (Incremental build) all Variables
Task Tags	✓ Clean Clean Variables
Validation	
WIKITEXL	
	Restore Defaults Apply
?	OK Cancel

- SimpleProject/Properties/C/C++Build/Behaviour
- ♦ Click **Build on resource save (Auto build)** for automatic rebuild
- ♥ Click Apply

22



- SimpleProject/Properties/C/C++Build/Settings/AVR Compiler/Directories
- Set Include Path C:\arduino-1.0.3\hardware\tools\avr\avr\include
- Set Include Path C:\arduino-1.0.3\hardware\arduino\cores\arduino
- Set Include Path C:\arduino-1.0.3\hardware\arduino\variants\standard
 - See boards.txt (uno.build.variant) for your variant (standard for this example)
- ♥ Click OK

Properties for SimpleProj	ject	
type filter text	Settings $\Leftrightarrow \bullet $	
Resource AVR Builders C/C++ Build Build Variables Discovery Options Environment Logging Settings Tool Chain Editor C/C++ General Coverage Project Facets Project References Run/Debug Settings Server Task Repository Task Tags Validation WikiText	 Additional Tools in Toolchain MAVR Assembler General Paths Debugging S AVR Compiler Directories Symbols Warnings Debugging Miscellaneous Symbols Warnings Debugging AVR C++ Compiler Directories Symbols Miscellaneous AVR C++ Linker AVR C++ Linker AVR C++ Linker Miscellaneous AVR C++ Linker Miscellaneous AVR C++ Linker 	
?	OK Cancel	

- ✤ Do the same again for SimpleProject/Properties/C/C++Build/Settings/AVR C++ Compiler/Directories
- Set Include Path C:\arduino-1.0.3\hardware\tools\avr\avr\include
- Set Include Path C:\arduino-1.0.3\hardware\arduino\cores\arduino
- Set Include Path C:\arduino-1.0.3\hardware\arduino\variants*standard*
 - See boards.txt (uno.build.variant) for your variant (standard for this example)
- ♥ Click OK

© 2012 actifsource GmbH, Switzerland - all rights reserved.



- ♦ Create a new folder src/arduino in the project

Project Explorer	×		
	E	3 42 20 ▽	
4 😂 SimpleProjec		New	۲
Includes		Go Into	
▲ G→ src (≥ arduino)	1	Open in New Window	
-	Ð	Сору	Ctrl+C
	Ê	Paste	Ctrl+V
	×	Delete	Delete
	<u>.</u>	Remove from Context	Ctrl+Alt+Shift+Down
		Source	•
		Move	
		Rename	F2
	2	Import	
L. L.	4	Export	
	8	Refresh	F5
		Index	•
		Make Targets	•
		Resource Configurations	+
		Validate	
	\mathcal{K}	Generic Refactoring	
	1	Generate Code	
		Show JavaClassName for Selection (DEV)	
		Team	•
		Compare With	+
		Restore from Local History	
	*	Run C/C++ Code Analysis	
		Properties	Alt+Enter

- ✤ For the folder src/arduino choose Import...
- ♦ Choose Import/General/File System

26

Import					
File system Import resources from the local file system.					
From directory: C:\arduino-1.0.3\hardware\arduino\core	s\arduino 👻 Browse				
arduino					
Filter Types Select All Deselect All Into folder: SimpleProject/src/arduino	Browse				
Options Overwrite existing resources without warning Create top-level folder					
<< Advanced Create links in workspace Create virtual folders Create link locations relative to: PROJECT_LOC					
? < Back	Next > Finish Cancel				

- Select directory C:\arduino-1.0.3\hardware\arduino\cores\arduino
- ♥ Select All files
- ✤ Important: Deselect main.cpp
- ♦ Click Advanced
- ♦ Select Create links in workspace
- ♥ Click Finish



😓 Create a new file main.c in the folder src

```
#include <Arduino.h>
                                 // include the arduino header file
                                 // LED connected to digital pin 13
int ledPin = 13;
int main(void)
{
  init();
                                 // initialize the arduino hardware
  pinMode(ledPin, OUTPUT);
                                 // initialize the digital pin as an output:
 while(true)
                                 // loop forever
  {
   digitalWrite(ledPin, HIGH); // set the LED on
   delay(1000);
                               // wait for a second
   digitalWrite(ledPin, LOW); // set the LED off
   delay(1000);
                               // wait for a second
  }
  return 0;
```

- ♥ Write a very simple Arduino program in the file main.c as shown above
- If you enabled **Build on resource save (Auto build)** in SimpleProject/Properties/C/C++Build/Behaviour before, the code should now be built
- ① You can also build manually by pressing Project/Build All (Ctrl+B) or the build button in the Eclipse toolbar
- ③ Your code is probably not linking if you named your file main.cpp instead if main.c

C/C++ - Eclipse		X
<u>File Edit Source Refac</u>	t <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un A <u>V</u> R <u>W</u> indow <u>H</u> elp	
	$ \begin{array}{c} \hline \\ \hline $	»
Project E 🛛 🗖 🗖	Arduino.h 🕼 main.c 🛛	
E SimpleProject Binaries	<pre>#include <arduino.h> int ledPin = 13; // LED connected to digital pin 13</arduino.h></pre>	*
 M Includes → Release → src → arduino 	<pre>// The setup() method runs once, when the sketch starts void setup() { // initialize the digital pin as an output: <</pre>	•
indianic indianic	😰 Problems 🕢 Tasks 💷 Console 🗵 🔲 Properties 🖀 AVR Device Explorer 🖀 AVR Supported MCUs	
	AVRDude	- 🗊 -
	Writing ###################################	* III *
 ↓ ↓ ↓ ↓		4

- ✤ Download your code to the Arduino
 - \circ ~ Select your project SimpleProject in the Project Explorer ~
 - Press the AVR download button in the Eclipse toolbar

30



- ✤ Mount your LED between Pin 13 and GND
 - Don't forget the 330Ω resistor for a 1.7V LED with 10mA → (5V-1.7V)/0.01A = 330Ω
- ① The program starts automatically after the download has been completed
 - \circ ~ The LED should switch on and off once per second

Part 4: CIP Arduino project

- Modeling a reactive state machine using the CIP method
- Generating C code for the state machine
- Connecting the generated state machine code to the Arduino I/O
- Downloading and testing to the Arduino board

		New Go Into	•		
		Open in New Window			
Project Explorer SimpleProject SimpleProject SimpleProject SimpleProject SimpleProject Release SimpleProject SimpleProj	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Copy Paste Delete Remove from Context Source Move Rename	Ctrl+C Ctrl+V Delete Ctrl+Alt+Shift+Down		
⊳ 🛃 main.c	2 2	Import Export			
	\$	Build Project Clean Project Refresh Close Project Close Unrelated Projects	F5		
		Build Configurations Make Targets Index	k k		
	1	Validate Generate Code Convert To Add/Remove Code Sealing Nature Fix LicenceInfoMarkers via TeamPlugin Delete LicenceInfoMarkers Connect Project wit Subclipse Disconnect Project			
	*	Run As Debug As Profile As Profile As Coverage As Show JavaClassName for Selection (DEV) Team Compare With Restore from Local History	> > > > >		
		AVR Source Configure			Convert to Plug-in Projects
	*	Run C/C++ Code Analysis			Convert to Maven Project
	×	Show Unreferenced Actifsource Folders	•	0	Add Actifsource ExternalBuilder Nature
		Properties	Alt. Fat	<u>~~</u>	Aud Acutsource Nature
		Properties	Alt+Enter		Add JavaNature

- V Select your project SimpleProject in the Project Explorer
- Select Configure/Add Actifsource Nature

© 2012 actifsource GmbH, Switzerland - all rights reserved.

Properties for SimplePro	ject	
type filter text	actifsource	$\Leftrightarrow \bullet \bullet \bullet \bullet \bullet$
Resource actifsource AVR Builders C/C++ Build C/C++ General Coverage Project Facets Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Resource Paths Target Folders Project Dependencies	 Built-in Dependencies Add Resource Path Edit Remove generate javamodel
		Apply
?	Ok	Cancel

- ✤ Open SimpleProject/Properties/actifsource/Resource Paths
- ♦ Add the Resource Path <code>asrc</code>
- ♥ Press Apply

Properties for SimplePro	ject	
type filter text	actifsource	$\Leftrightarrow \bullet \bullet \bullet \bullet \bullet$
Resource actifsource AVR Builders C/C++ Build C/C++ General Coverage Project Facets Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Resource Paths Project Dependencies	Built-in Dependencies Add Builtin Edit Remove
		Apply
?	OK	Cancel

- ♥ Open SimpleProject/Properties/actifsource/Built-in Dependencies
- ✤ Add Builtin CIP
- ♦ Add Builtin DATATYPE
- ✤ Important: Press Apply

Properties for SimplePro	ject	
type filter text	actifsource	$\Leftrightarrow \bullet \bullet \bullet \bullet \bullet$
Resource actifsource AVR Builders C/C++ Build C/C++ General Coverage Project Facets Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Resource Paths Project Dependencies // Src Chactifsource.cip.meta.generator.tpl_c.buildconfig.CIP_C	 Built-in Dependencies Add Target Folder Add BuildConfig Add Package Add Resource Edit Remove ignore whitespaces execute saveactions
		Apply
?	OK	Cancel

- ♥ Open SimpleProject/Properties/actifsource/Target Folders
- $~~~\forall ~~$ Add the existing folder ${\tt src}$ as target folder
- ♦ Add the BuildConfig <u>CIP C</u> to the target src
- 🄄 Press **OK**

	▼
9	CipAnimation - ch.actifsource.cip.meta.animation
8 K	CipSystem - ch.actifsource.cip.meta.core.meta.system
873 1973	DiagramType - ch.actifsource.ui.diagram.diagramtype DomainDiagram - ch.actifsource.ui.diagram
e	ShapeAction - ch.actifsource.ui.diagram.figure.action
	ConnectionDecorator - ch.actifsource.ui.diagram.classes.connection ConnectionRouter - ch.actifsource.ui.diagram.classes.connection DiagramConfiguration - ch.actifsource.ui.diagram.classes EditPolicyImpl - ch.actifsource.ui.diagram.classes.edit EditPolicyProvider - ch.actifsource.ui.diagram.classes.edit LineStyle - ch.actifsource.ui.diagram.classes.connection MetaConnection - ch.actifsource.ui.diagram.classes.connection
e	LiteralEditorAspect - ch.actifsource.ui.builder
01 01 01	DefaultStyleClass - ch.actifsource.template.syntaxstyle.generic LanguageSyntaxStyle - ch.actifsource.template.syntaxstyle.generic StyledElement - ch.actifsource.template.syntaxstyle.generic
97	BuildConfig - ch.actifsource.generator
63 63 8	Cardinality - ch.actifsource.core Class - ch.actifsource.core Color - ch.actifsource.core
93 93	Enum - ch.actifsource.core FunctionSpace - ch.actifsource.core.selector.type
∾ ¶	Literal - ch.actifsource.core MetaModel - ch.actifsource.core.selector.type

- $\,\, \ensuremath{{\diamondsuit}}$ Select the Resource Folder <code>asrc</code> in the Project Explorer
- ♦ Select <u>CipSystem</u> from the New Resource Tool in the Eclipse toolbar

New Resource Wizard					
Actifsource Resource Creates a new actifsource resource of some type in the specified location.					
<u>R</u> esource Path:	/SimpleProject/asrc	Browse			
<u>N</u> amespace:	lampsystem	Browse			
<u>O</u> wnRelation:		Browse			
<u>T</u> ype:	ch.actifsource.cip.meta.core.meta.system.CipSystem	Browse			
<u>N</u> ame:	LampSystem				
Modifiers:	Abstract Final				
SuperClass:	ch.actifsource.core.NamedResource	Browse			
?	Finish	Cancel			

- Enter lampsystem as Namespace
- 🗞 Enter LampSystem as Name
- 🔄 Press Finish



🤄 Create a LampSystem as shown in the Actifsource Tutorial **CIP Statemachine – Lamp**



- Create a new Implementation named LampSystemImplementation in your LampSystem
- Create a new ImplementationUnit named LampSystemUnit in your LampSystemImplementation
- Create a new <u>CipMachine</u> in your <u>LampSystemUnit</u>
- Reference the <u>Cluster LampCluster</u> in your <u>CipMachine</u>
- Screate a new <u>CipShell</u> in your <u>LampSystemUnit</u>
- Reference the input channel <u>Button</u>
- ✤ Reference the output channel Lamp
- Create a new <u>C_CodeOption</u> named <u>LampSystem_CodeOption</u> using Content Assist (Ctrl+Space)

40

typeOf	$ch.actif source.cip.meta.generator.tpl_c.cip_codeoptions.C_CodeOptions$				
name	LampSystem_CodeOptions				
cip_Shell	typeOf	CIP_Shell	ell		
	useInterface	useMessa	geInterface		
	😑 usePostfix	typeOf	UnitPostFix		
	allTypesInShell	false			
	enable_PENDING_Information	true			
	callInputErrorFunction	false			
	generateTypes	true			
• cip_Error	: CIP_Error				
🗉 trace	: Trace				
advanced	: Advanced				

- ✤ Configure the newly created LampSystem CodeOption
- ♦ Set <u>useInterface</u> to <u>useMessageInterface</u>
- ♦ Set **usePostFix** to **InitPostFix**
- ♦ Set enable PENDING_Information to true
- b The code for your CIP system is now generated in the target folder src

```
#include "arduino.h"
#include "sLampSystemUnit.h"
int buttonPin = 2; // button connected to digital pin 2
int lampPin = 13; // lamp connected to digital pin 13
void AI_LampSystemUnit_C2_Dark() {digitalWrite(lampPin, LOW);}
void AI LampSystemUnit C2 Bright() {digitalWrite(lampPin, HIGH);}
void iMSG LampSystemUnit()
{
 OUT LampSystemUnit.C2 Dark = AI LampSystemUnit C2 Dark;
 OUT LampSystemUnit.C2 Bright = AI LampSystemUnit C2 Bright;
}
int main()
{
  if (!fINIT_LampSystemUnit()) {return 1;} // init cip
                                               // <u>init</u> arduino
// <u>init</u> button pin
// <u>init</u> lamp pin
  init();
  pinMode(buttonPin, INPUT);
  pinMode(lampPin, OUTPUT);
  int oldbuttonState = 0;
  while (1)
  {
    int newButtonState = digitalRead(buttonPin); // read button pin
    TRG LampSystemUnit.TICK ();
                                                  // ticking time
    if (oldbuttonState!=newButtonState && newButtonState==1) {IN_LampSystemUnit.C1_Press();}
    if (oldbuttonState!=newButtonState && newButtonState==0) {IN LampSystemUnit.C1 Release();}
    oldbuttonState=newButtonState;
    while (TRG LampSystemUnit.PENDING .ANY ) {TRG LampSystemUnit.STEP ();}
    delay(3);
  }
  return 0;
```

- ✤ Rewrite your main.c as shown above
- ① Your code is probably not linking if you named your file main.cpp instead if main.c



- bownload your code to the Arduino
 - Select your project SimpleProject in the Project Explorer
 - Press the AVR download button in the Eclipse toolbar



- ③ We built a lamp with a delayed off switch
 - o Pin 2 on GND means Button is released
 - o Pin 2 on 5V means Button is pressed
- 🗞 Start while Button is released
- Press Button \rightarrow Light should switch on
- Release Button \rightarrow Light should switch off with delay
 - o Tune with the delay function of your timer and the delay() call in the main.c

44



© 2012 actifsource GmbH, Switzerland - all rights reserved.