

Take Off!



Ada for Automation

Freedom and Power for Control Engineers

Ada for Automation Read Me

Stéphane LOS

Version 2022.02, 2022-02-22

Table of Contents

1. Description	1
2. General idea	2
3. License	5
4. Directories	6
5. Kernels	7
6. Applications	8
7. GNAT Pro Studio Projects	9
7.1. Shared projects	9
7.2. Basic projects	9
7.2.1. A4A Project diagram	10
7.2.2. A4A Kernel 0 Project diagram	10
7.2.3. A4A Kernel 0b Project diagram	10
7.2.4. A4A Kernel 1 Project diagram	10
7.2.5. A4A Kernel 2 Project diagram	11
7.2.6. A4A Kernel 3 Project diagrams	11
7.2.7. A4A Kernel 4 Project diagram	12
7.2.8. A4A Kernel 5 Project diagram	13
7.2.9. A4A Kernel 6 Project diagram	13
7.2.10. A4A Kernel 7 Project diagram	14
8. Demo applications	15
8.1. Overview	15
8.1.1. By Kernel	15
8.1.2. By matchings	17
8.1.3. Summary	20
8.2. 000 a4a-k0b-cli	21
8.3. 000 a4a-k0-cli	21
8.4. 001 a4a-k0b-gui	21
8.5. 001 a4a-k0-gui	21
8.6. 010 a4a_k0b_piano	22
8.7. 010 a4a_piano	22
8.8. 020 a4a-k1-cli	22
8.9. 021 a4a-k1-gui	22
8.10. 022 a4a-k1-wui	22
8.11. 030 app1simu-cli	23
8.12. 031 app1simu-gui	23
8.13. 032 app1simu-wui	23
8.14. 040 app1-cli	23
8.15. 041 app1-gui	23

8.16. 042 app1-wui	23
8.17. 052 a4a_hilscherx_piano	24
8.18. 062 a4a-k3-wui	24
8.19. 070 app2-cli	24
8.20. 071 app2-gui	24
8.21. 080 app3-cli	24
8.22. 081 app3-gui	25
8.23. 082 app3-wui	25
8.24. 090 app4	25
8.25. 100 app5-cli	25
8.26. 101 app5-gui	25
8.27. 110 app6-cli	25
8.28. 111 app6-gui	25
8.29. 122 app7-wui	25
8.30. 132 a4a-k7-wui	25
8.31. 142 a4a_k0_S7	26

Chapter 1. Description

[Ada for Automation](#) (A4A in short) is a framework for designing industrial automation applications using the Ada language.

It makes use of the [libmodbus](#) library to allow building a ModbusTCP client or server, or a Modbus RTU master or slave.

It can also use [Hilscher](#) communication boards allowing to communicate on field buses like AS-Interface, CANopen, CC-Link, DeviceNet, PROFIBUS, EtherCAT, Ethernet/IP, Modbus TCP, PROFINET, Sercos III, POWERLINK, or VARAN.

With the help of [GtkAda](#), the binding to the [Graphic Tool Kit](#), one can design Graphical User Interfaces.

Thanks to [Gnoga](#), built on top of [Simple Components](#), it is also possible to provide a Web User Interface.

Nice addition is the binding to the [Snap7](#) library which allows to communicate with SIEMENS S7 PLCs using S7 Communication protocol ISO on TCP (RFC1006).

Of course, all the Ada ecosystem is available.

Using Ada bindings, C, C++, Fortran libraries can also be used.

And, since it is Ada, it can be compiled using the same code base to target all major platforms.

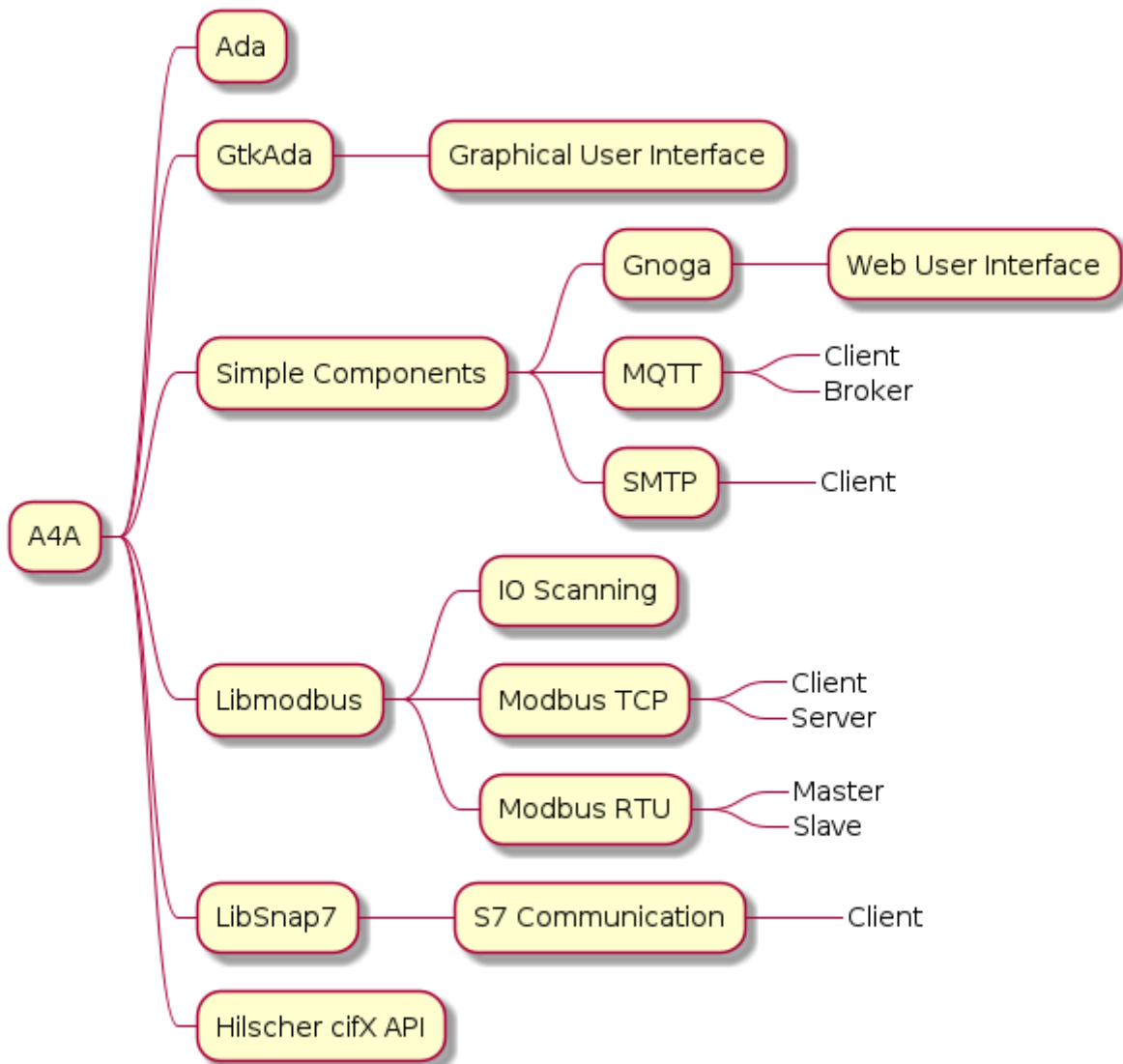
Chapter 2. General idea

Ada For Automation Applications



figure 1
Applications one can think of

Ada For Automation



What else will you add ?

figure 2
This is the General Idea

Hilscher cifX API

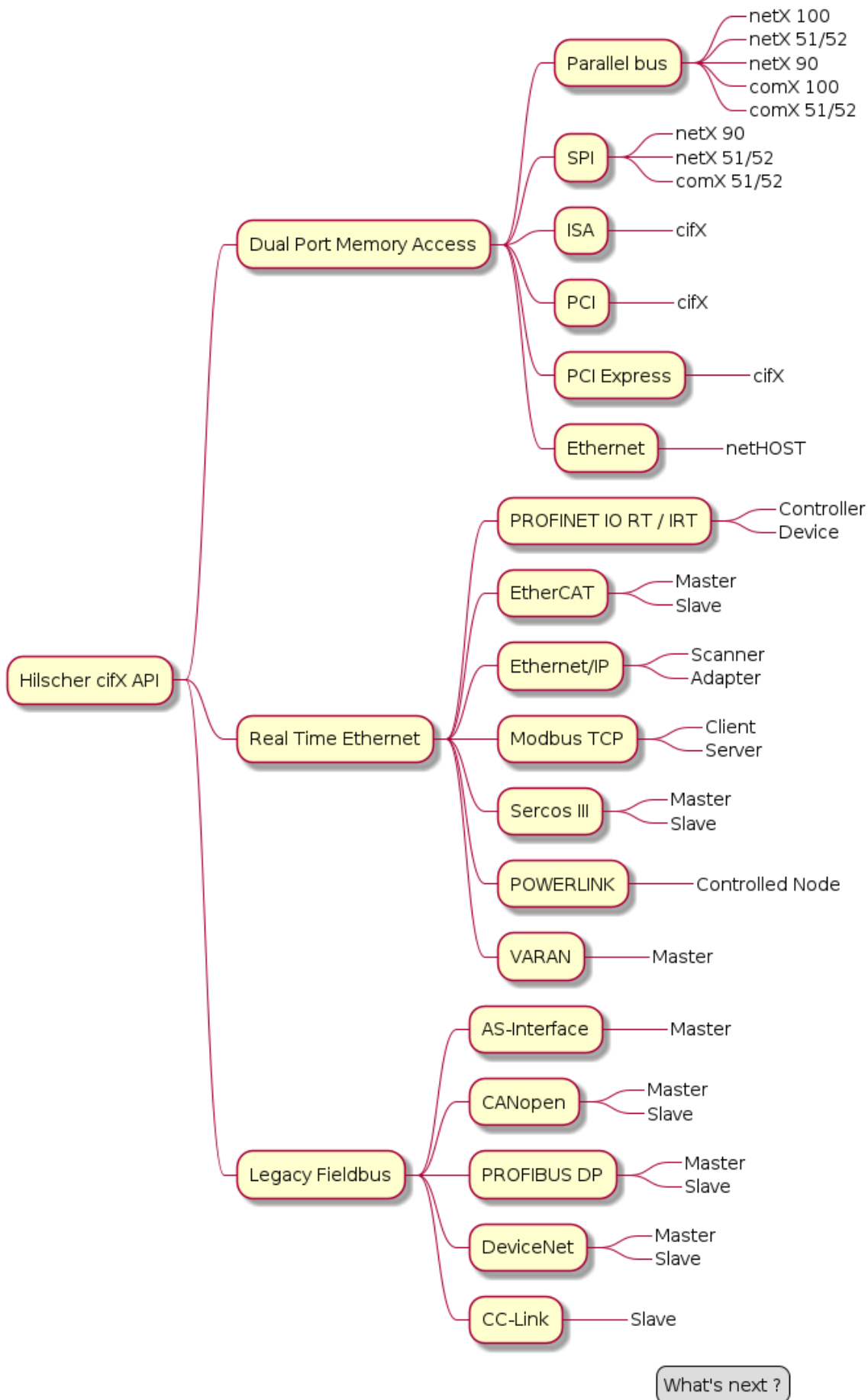


figure 3
This is Hilscher cifX API contribution

Chapter 3. License

COPYING3

The GPL License you should read carefully. GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

COPYING.RUNTIME

GCC RUNTIME LIBRARY EXCEPTION Version 3.1, 31 March 2009

Chapter 4. Directories

a4a_apps

This is the basic "Ada for Automation" application folder.

Contains historic build artifacts.

Nothing interesting, is going to disappear soon.

book

The book "Ada for Automation" is elaborated here.

This book is composed in plain text in the AsciiDoc format and processed to provide HTML or PDF files.

demo

This is the Demo applications folder.

exp

Contains historic build artifacts.

Nothing interesting, is going to disappear soon.

gpr

Contains shared projects files.

hilscherx

The directory for the Hilscher cifX API binding.

You will need it if you would use a Hilscher cifX board that can provide the connectivity you need to manage your equipment on all major fieldbuses Real Time Ethernet or legacy.

Contains historic stuff. Most is now in src/hilscherx except test items.

Nothing interesting, is going to disappear soon.

icons

GUI Applications icons.

src

Ada for Automation framework source files.

test

The place to place all sorts of tests and experiments or examples.

tutorial

Everyone starts some day.

www

Web related files like documentation portal.

Chapter 5. Kernels

Kernels feature the core functionality needed by applications.

Following are available ones :

kernel0 : Modbus TCP Server

kernel0b : Modbus RTU Slave

kernel1 : Modbus TCP Server + Modbus TCP IO Scanning

kernel2 : Modbus TCP Server + Modbus RTU IO Scanning

kernel3 : Modbus TCP Server + one Hilscher cifX channel

kernel4 : Modbus TCP Server + Modbus TCP IO Scanning + one Hilscher cifX channel

kernel5 : Modbus TCP Server + Modbus TCP IO Scanning + two Hilscher cifX channels

kernel6 : One Hilscher cifX channel

kernel7 : Modbus TCP IO Scanning + One Hilscher cifX channel

Chapter 6. Applications

An application is built around the kernel, with application logic added by the framework user.

Thanks to [GtkAda](#), it can provide a Graphic User Interface.

Thanks to [Gnoga](#), it can provide a Web User Interface.

Chapter 7. GNAT Pro Studio Projects

GNAT Pro Studio allows to arrange the projects in a project tree with inheriting other projects. This will expand the project by adding or substituting source code files. You will please refer to [GPRbuild](#) documentation regarding the projects.

7.1. Shared projects

The following projects are common to all CLI, GUI or WUI applications.

shared.gpr

An abstract project, shared by the other ones and containing common elements.

shared_hilscherx.gpr

An abstract project, shared by the other ones and containing common Hilscher related elements.

libmodbus.gpr

The project for the libmodbus library.

It is possible that you have to adjust it to suit your installation.

libsnap7.gpr

The project for the Snap7 library.

It is possible that you have to adjust it to suit your installation.

libcifx.gpr

The project for the Hilscher cifX Device Driver library.

It is possible that you have to adjust it to suit your installation.

Only required if using a Hilscher cifX card.

libnetxtransport.gpr

The project for the Hilscher netXTransport library.

It is possible that you have to adjust it to suit your installation.

Only required if using a Hilscher netHOST.

gtkada.gpr

The project for the gtkada library.

It comes with installed library.

gnoga.gpr

The project for the gnoga library.

It comes with installed library.

7.2. Basic projects

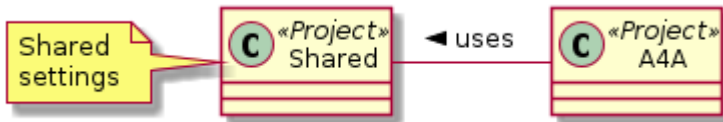
The following projects are inherited by applications projects.

7.2.1. A4A Project diagram

This project brings in most of the functionality offered by the framework.

The tool chain will only pull in what is needed by the application itself.

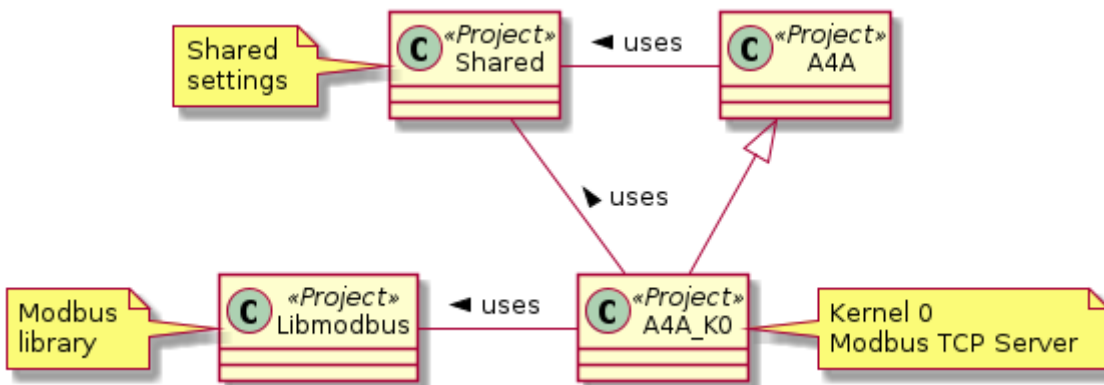
The following picture shows the diagram of the project :



7.2.2. A4A Kernel 0 Project diagram

This project inherits from the A4A project adding the Kernel 0 featuring a Modbus TCP Server.

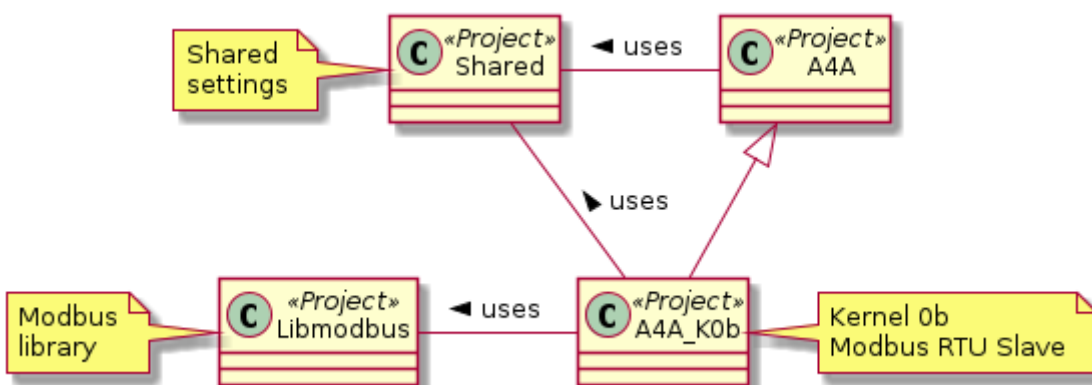
The following picture shows the diagram of the project :



7.2.3. A4A Kernel 0b Project diagram

This project inherits from the A4A project adding the Kernel 0b featuring a Modbus RTU Slave.

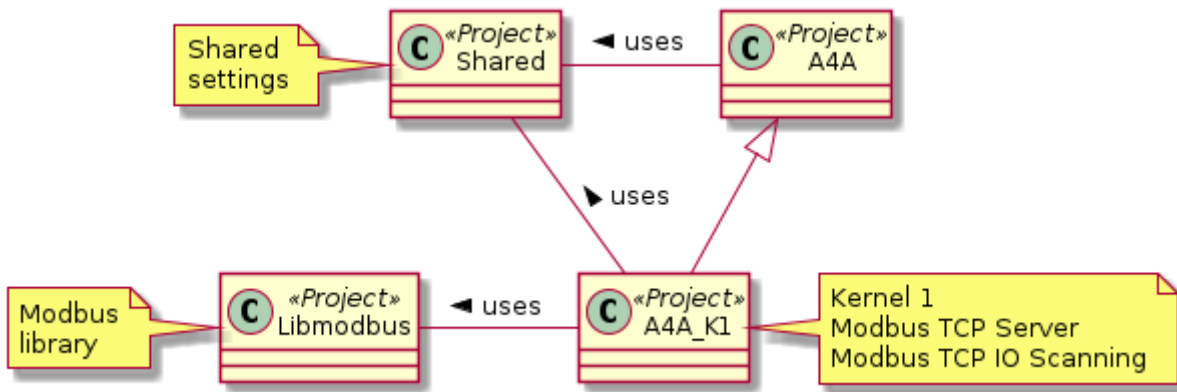
The following picture shows the diagram of the project :



7.2.4. A4A Kernel 1 Project diagram

This project inherits from the A4A project adding the Kernel 1 featuring a Modbus TCP Server and Modbus TCP IO Scanning.

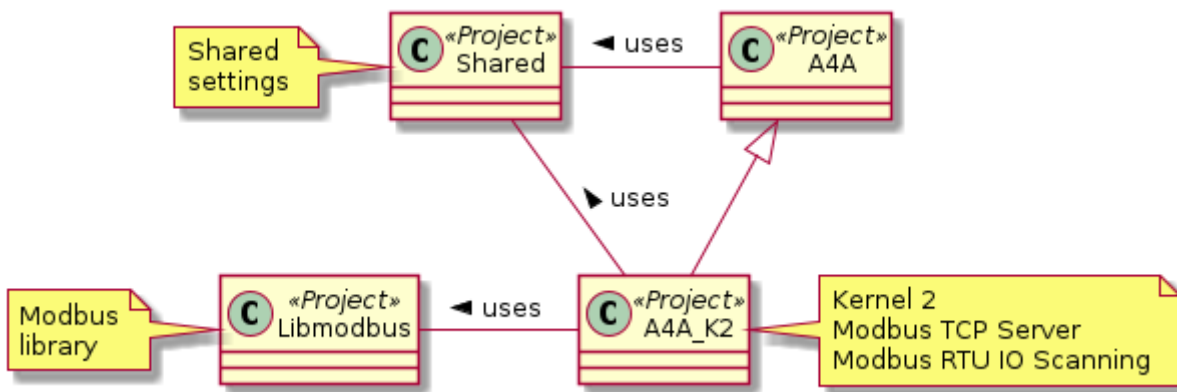
The following picture shows the diagram of the project :



7.2.5. A4A Kernel 2 Project diagram

This project inherits from the A4A project adding the Kernel 2 featuring a Modbus TCP Server and Modbus RTU IO Scanning.

The following picture shows the diagram of the project :



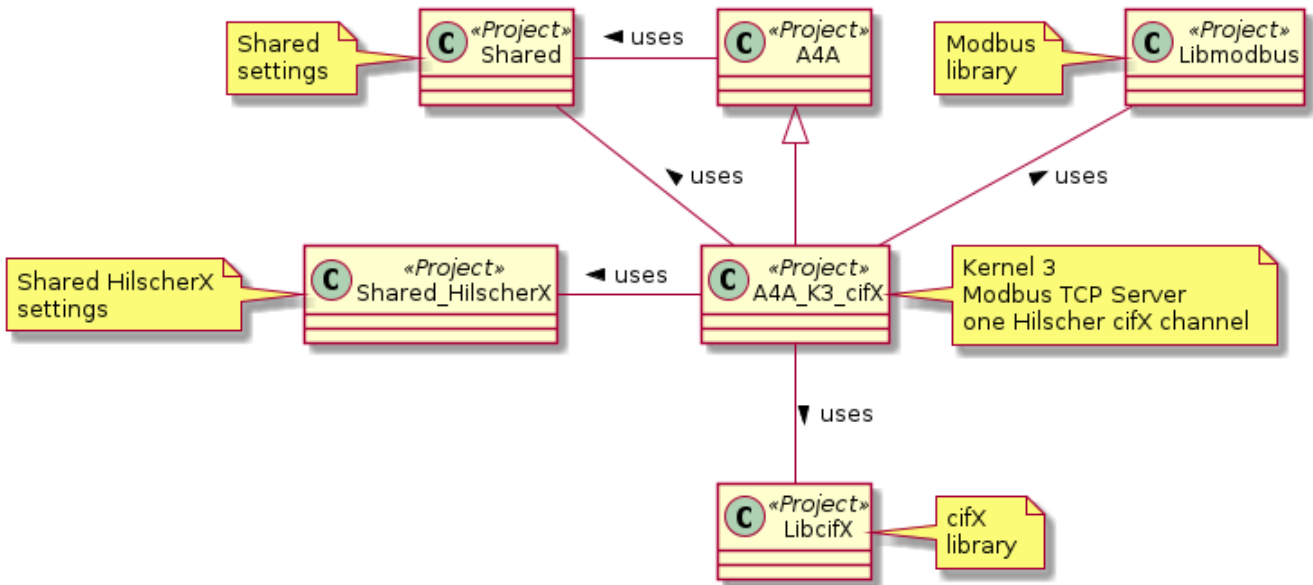
7.2.6. A4A Kernel 3 Project diagrams

This project inherits from the A4A project adding the Kernel 3 featuring a Modbus TCP Server and one Hilscher cifX channel.

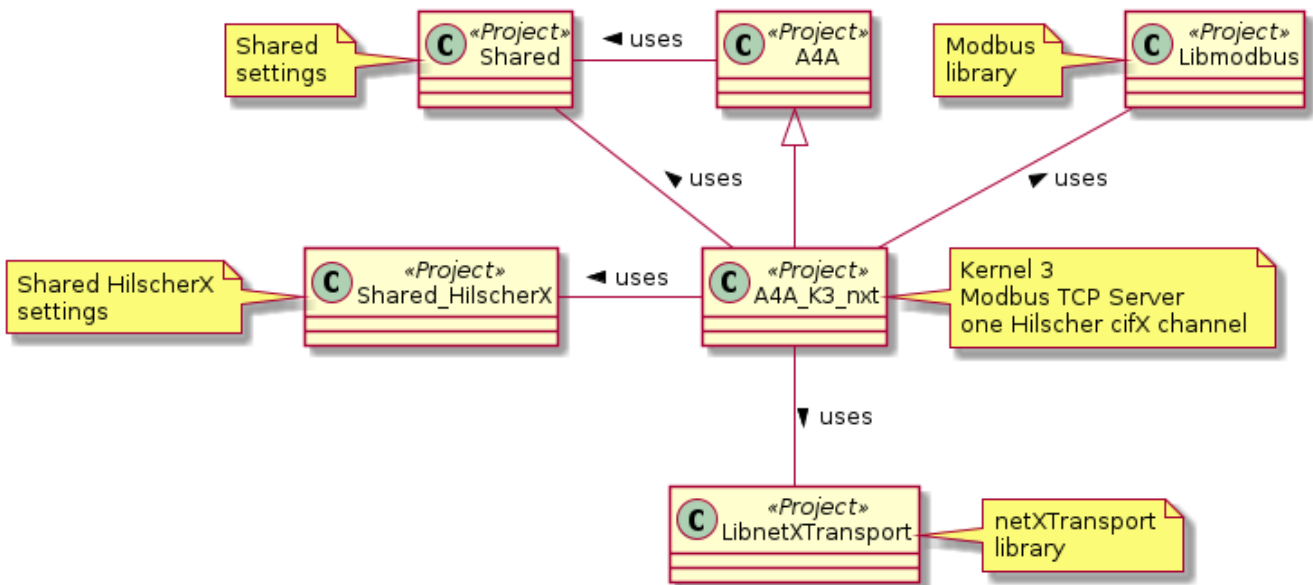
There are two variants :

- one using the standard Hilscher cifX API for use with most Hilscher products with ISA, PCI, PCI Express, DPM, SPI interfaces,
- one using the standard Hilscher netXTransport API which has the same cifX API but via a TCP/IP connection for use with Hilscher netHOST.

The following picture shows the diagram of the project using cifX API :



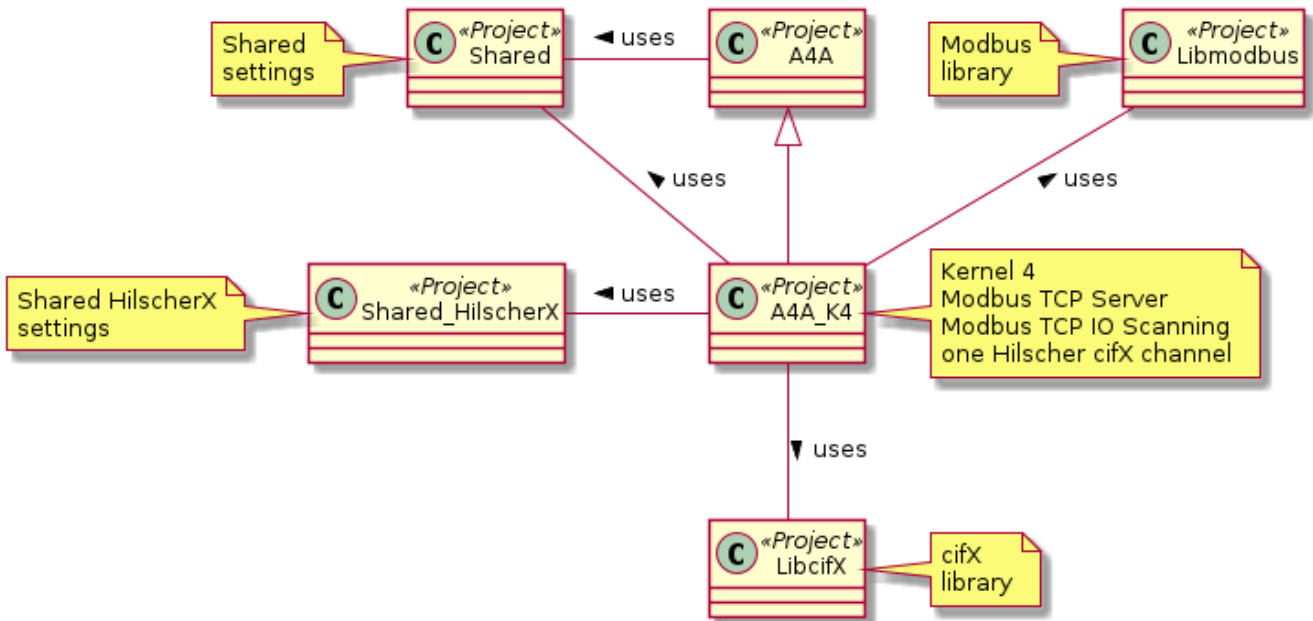
The following picture shows the diagram of the project using netXTransport API :



7.2.7. A4A Kernel 4 Project diagram

This project inherits from the A4A project adding the Kernel 4 featuring a Modbus TCP Server, Modbus TCP IO Scanning and one Hilscher cifX channel.

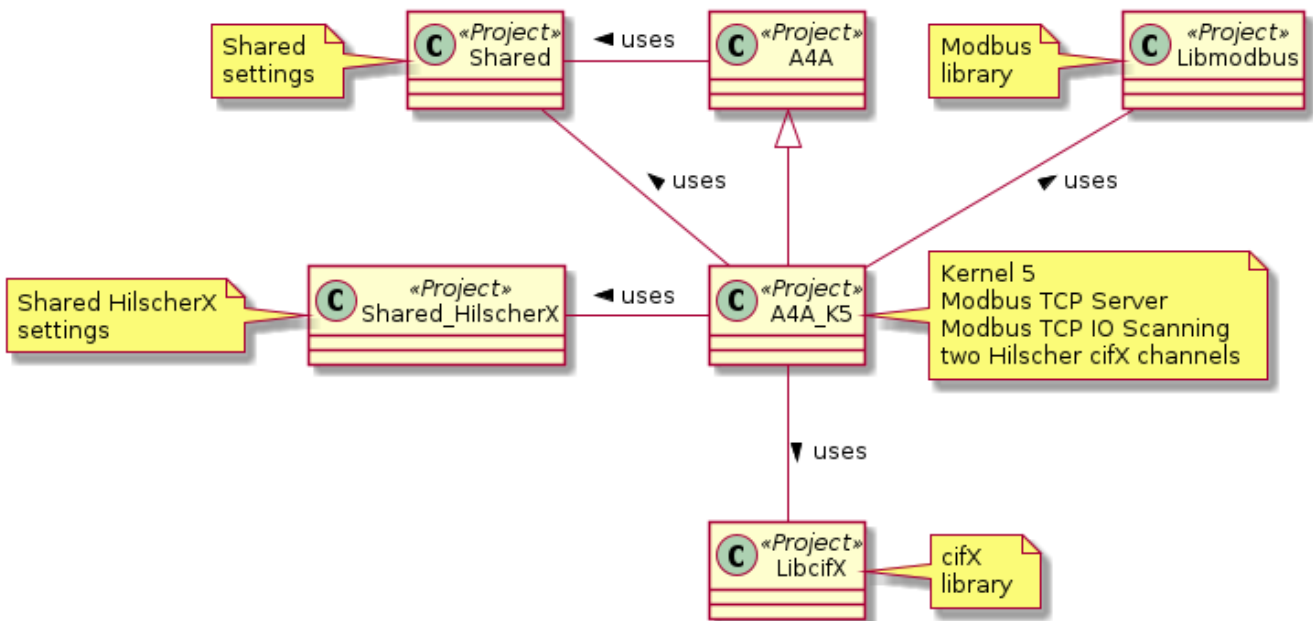
The following picture shows the diagram of the project using cifX API :



7.2.8. A4A Kernel 5 Project diagram

This project inherits from the A4A project adding the Kernel 5 featuring a Modbus TCP Server, Modbus TCP IO Scanning and two Hilscher cifX channels.

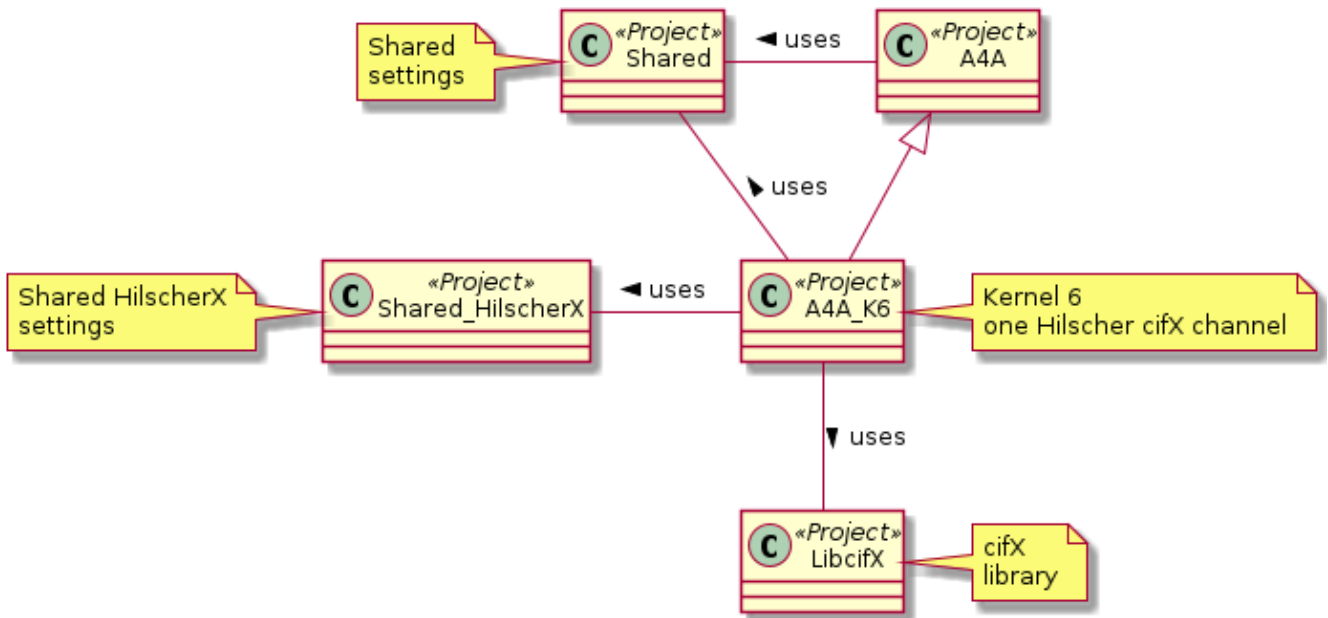
The following picture shows the diagram of the project using cifX API :



7.2.9. A4A Kernel 6 Project diagram

This project inherits from the A4A project adding the Kernel 6 featuring one Hilscher cifX channel.

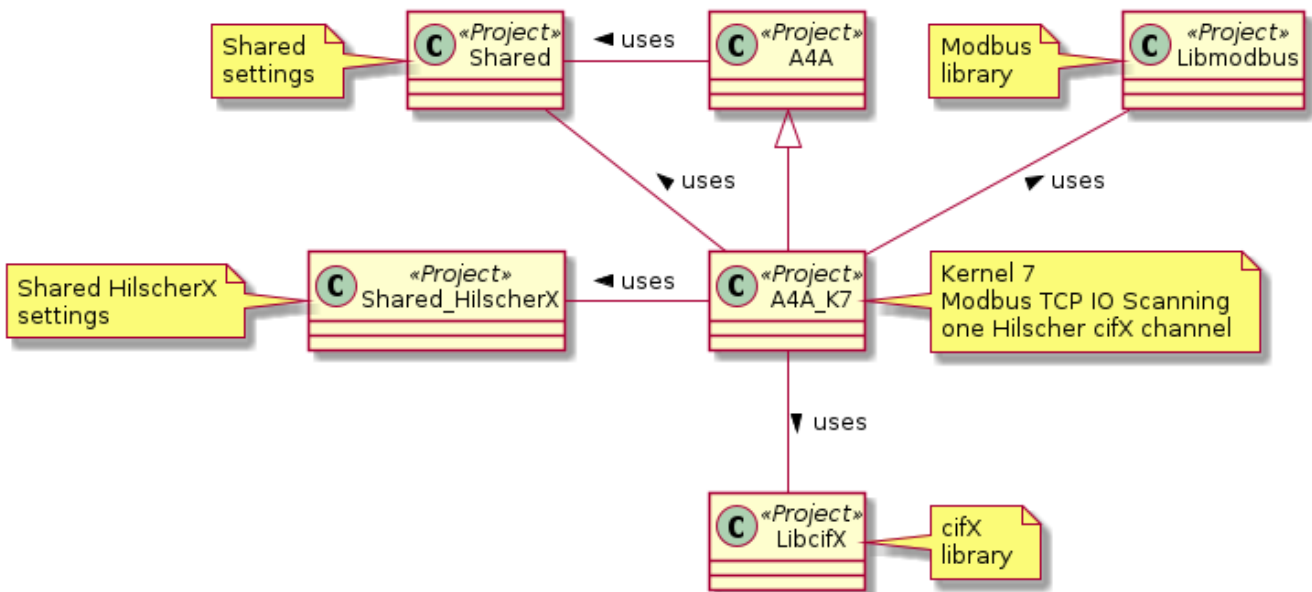
Project is having no dependency on libmodbus.



7.2.10. A4A Kernel 7 Project diagram

This project inherits from the A4A project adding the Kernel 7 featuring Modbus TCP IO Scanning and one Hilscher cifX channel.

The following picture shows the diagram of the project using cifX API :



Chapter 8. Demo applications

(in directory... demo)

A detailed information resides in the application folder.

8.1. Overview

Demo applications give a starting point to create your own application.

Whether you want to use [libmodbus](#) or a [Hilscher](#) cifX board, module or chip, there are demo applications using one or the other or both.

There are applications with a Command Line Interface, a [GtkAda](#) Graphic User Interface, or [Gnoga](#) Web User Interface.

Of course, CLI applications are the simplest one to dive in.

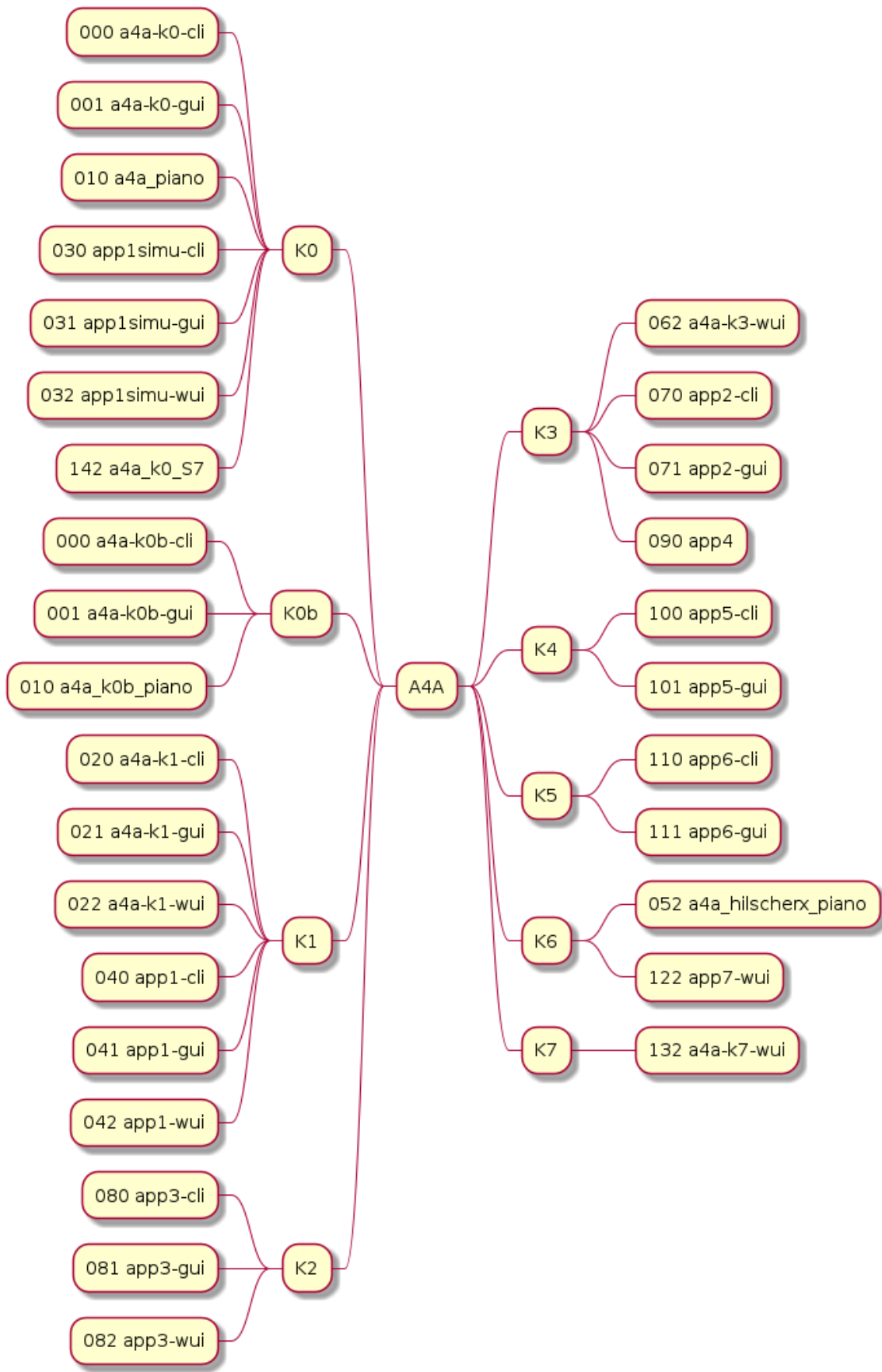
The GtkAda Graphic User Interface is limited to general application interface and communication status.

This is because there are several ways to design the process interface :

- GtkAda,
- [AICWL](#) on top of GtkAda,
- or embedding Gnoga in a GtkAda window.

8.1.1. By Kernel

Ada For Automation Demo Applications by kernel

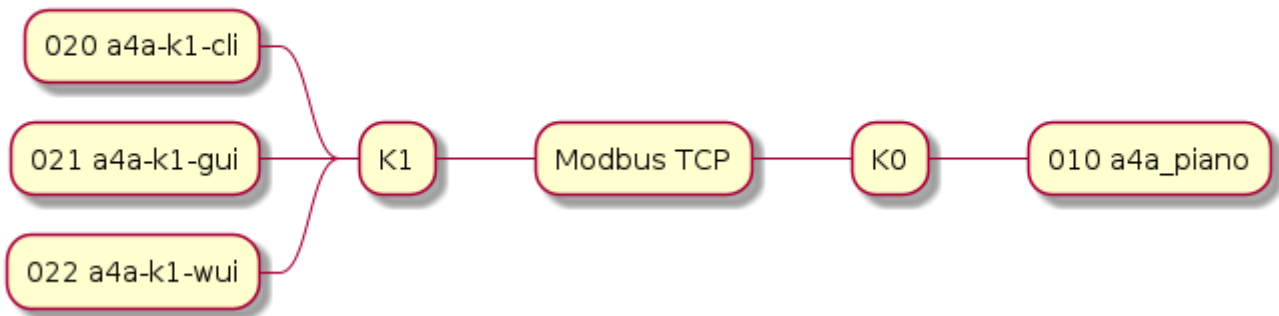


What about yours ?

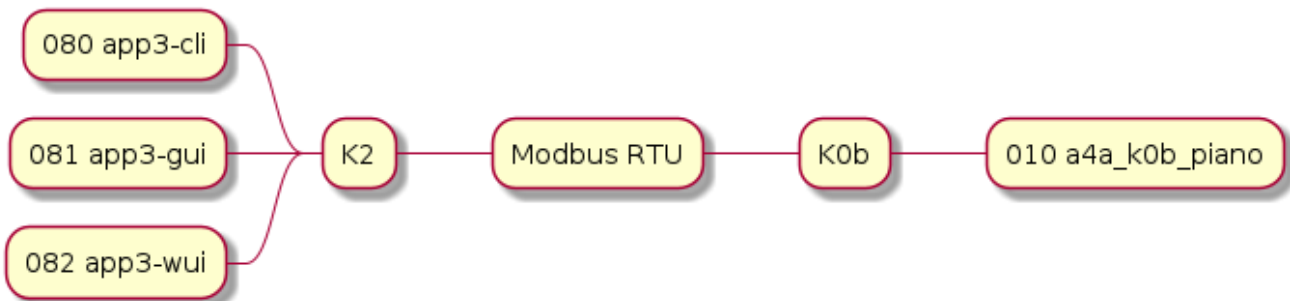
figure 1
Applications one can start from

8.1.2. By matchings

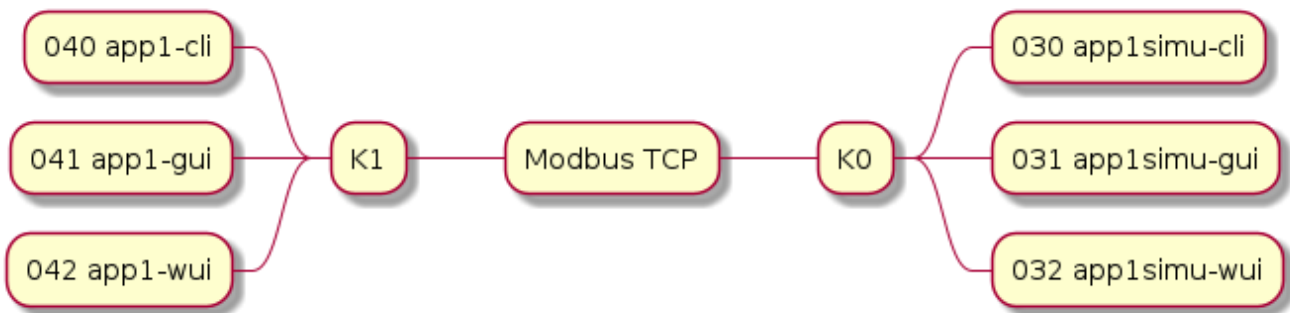
Playing piano with Modbus TCP



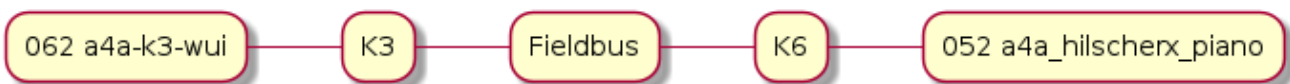
Playing piano with Modbus RTU



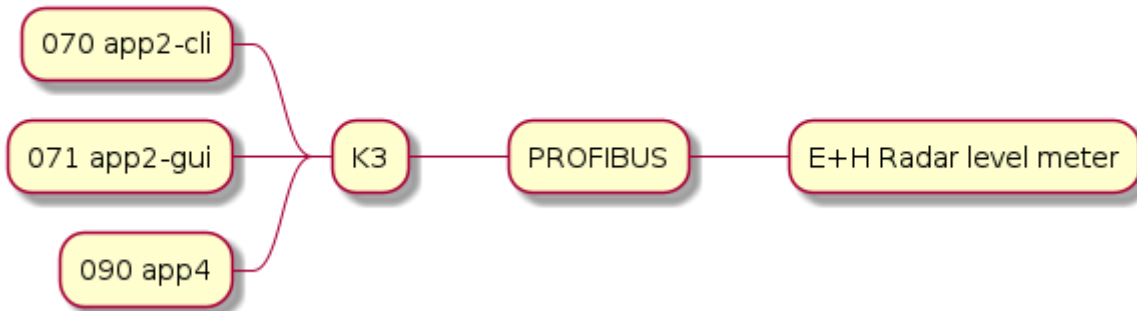
Example Application 1



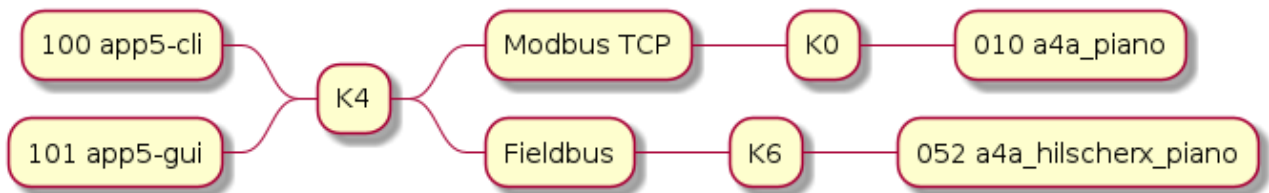
Playing piano with a fieldbus using Hilscher cifX API



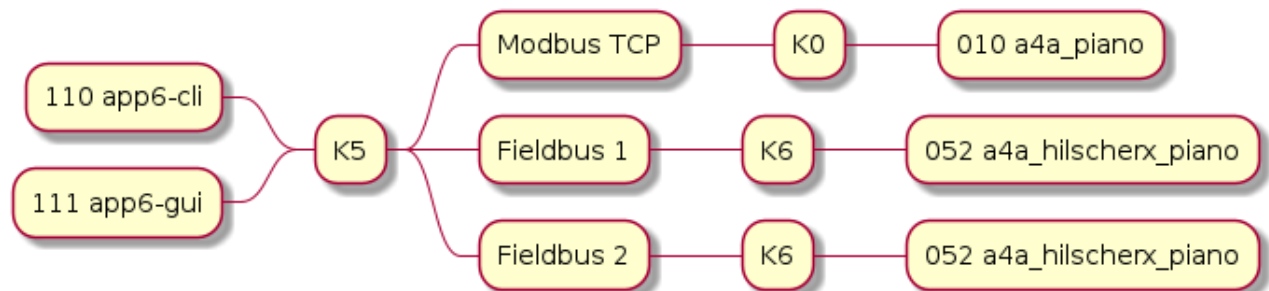
Monitoring E+H Radar level meter using PROFIBUS



Playing piano with Modbus TCP and one fieldbus using Hilscher cifX API



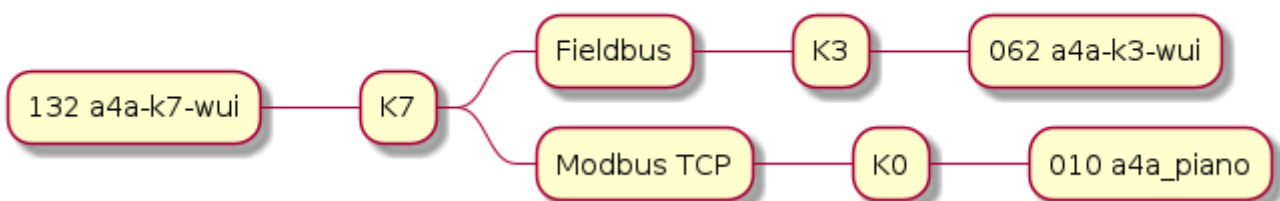
Playing piano with Modbus TCP and two fieldbuses using Hilscher cifX API



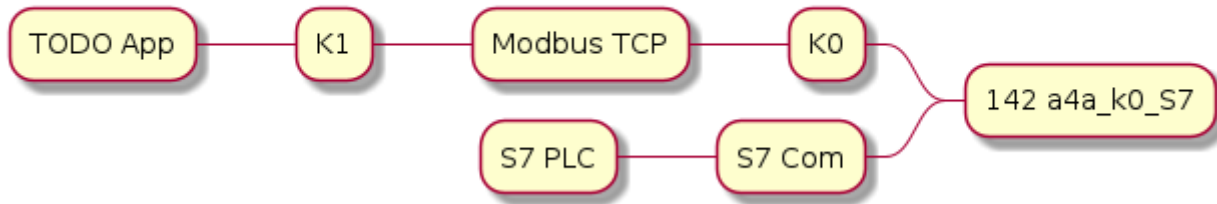
Control and Monitoring of an EtherCAT drive using Hilscher cifX API



Modbus TCP to fieldbus Gateway using Hilscher cifX API



Modbus TCP to S7 Communication Gateway using Snap7



8.1.3. Summary

Table 1. Features

Application	UI	KnI	Mbus	WSP	Description
000 a4a-k0b-cli	CLI	K0b	2	-	Modbus RTU Slave which mirrors inputs
000 a4a-k0-cli	CLI	K0	1504	-	Modbus TCP Server which mirrors inputs
001 a4a-k0b-gui	GUI	K0b	2	-	Modbus RTU Slave which mirrors inputs
001 a4a-k0-gui	GUI	K0	1504	-	Modbus TCP Server which mirrors inputs
010 a4a_k0b_piano	WUI	K0b	2	8081	Modbus RTU Slave Piano
010 a4a_piano	WUI	K0	1504	8081	Modbus TCP Server Piano
020 a4a-k1-cli	CLI	K1	1503	-	Modbus TCP Server / Client
021 a4a-k1-gui	GUI	K1	1503	-	Modbus TCP Server / Client
022 a4a-k1-wui	WUI	K1	1503	8080	Modbus TCP Server / Client
030 app1simu-cli	CLI	K0	1505	-	Modbus TCP Server simulation for App1
031 app1simu-gui	GUI	K0	1505	-	Modbus TCP Server simulation for App1
032 app1simu-wui	WUI	K0	1505	8083	Modbus TCP Server simulation for App1
040 app1-cli	CLI	K1	1502	-	Modbus TCP Server / Client App1
041 app1-gui	GUI	K1	1502	-	Modbus TCP Server / Client App1
042 app1-wui	WUI	K1	1502	8082	Modbus TCP Server / Client App1
052 a4a_hilscherx_piano	WUI	K6	-	8090	HilscherX Fieldbus Piano
062 a4a_k3-wui	WUI	K3	1503	8085	Modbus TCP Server / HilscherX Fieldbus
070 app2-cli	CLI	K3	1503	-	Modbus TCP Server / HilscherX Fieldbus E+H
071 app2-gui	GUI	K3	1503	-	Modbus TCP Server / HilscherX Fieldbus E+H
080 app3-cli	CLI	K2	1503	-	Modbus TCP Server / Modbus RTU Master
081 app3-gui	GUI	K2	1503	-	Modbus TCP Server / Modbus RTU Master
082 app3-wui	WUI	K2	1503	8085	Modbus TCP Server / Modbus RTU Master
090 app4-cli	CLI	K3	1503	-	Modbus TCP Server / HilscherX Fieldbus (App2)
100 app5-cli	CLI	K4	1503	-	Modbus TCP Server / Client / HilscherX Fieldbus
101 app5-gui	GUI	K4	1503	-	Modbus TCP Server / Client / HilscherX Fieldbus
110 app6-cli	CLI	K5	1503	-	Modbus TCP Server / Client / 2 x HilscherX Fieldbus
111 app6-gui	GUI	K5	1503	-	Modbus TCP Server / Client / 2 x HilscherX Fieldbus

Application	UI	Knl	Mbus	WSP	Description
122 app7-wui	WUI	K6	-	8090	HilscherX Fieldbus / EtherCAT Drive
132 a4a-k7-wui	WUI	K7	-	8085	Modbus TCP Client / HilscherX Fieldbus
142 a4a_k0_S7	WUI	K0	1504	8181	Modbus TCP Server / S7 Communication



- UI : User Interface
- CLI : Command Line User Interface
- GUI : Graphical User Interface
- WUI : Web User Interface
- Knl : Kernel
- WSP : Web Server Port
- Mbus : Modbus TCP Server Port or Modbus RTU Slave address.

8.2. 000 a4a-k0b-cli

This application implements a Modbus RTU Slave which mirrors inputs. When one does not have a Slave device to play with, one can use this.

This is the Command Line incarnation. There is a GtkAda UI version.

Inherits from A4A.K0b and so features a kernel0b.

8.3. 000 a4a-k0-cli

This application implements a Modbus TCP Server which mirrors inputs. When one does not have a Server device to play with, one can use this.

This is the Command Line incarnation. There is a GtkAda UI version.

Inherits from A4A.K0 and so features a kernel0.

Modbus TCP Server port : 1504

8.4. 001 a4a-k0b-gui

This application implements a Modbus RTU Slave which mirrors inputs. It shares **000 a4a-k0b-cli** logic and provides a Graphic User Interface. When one does not have a Slave device to play with, one can use this.

8.5. 001 a4a-k0-gui

This application implements a Modbus TCP Server which mirrors inputs. It shares **000 a4a-k0-cli** logic and provides a Graphic User Interface.

When one does not have a Slave device to play with, one can use this.

Modbus TCP Server port : 1504

8.6. 010 a4a_k0b_piano

This application implements a Modbus RTU Slave that mimics 16 push buttons and 16 LEDs with a web interface.

When one does not have a Slave device to play with, one can use this.

Inherits from A4A.K0b and so features a kernel0b.

Web Server port : 8081

8.7. 010 a4a_piano

This application implements a Modbus TCP Server that mimics 16 push buttons and 16 LEDs with a web interface.

When one does not have a Server device to play with, one can use this.

Inherits from A4A.K0 and so features a kernel0.

Modbus TCP Server port : 1504

Web Server port : 8081

8.8. 020 a4a-k1-cli

This application is meant to play with **010 a4a_piano**.

It implements Modbus TCP IO Scanning and a Modbus TCP Server which does not much at the moment (could be used to reflect I/O for example).

This is the Command Line incarnation. There are GtkAda UI and Web UI versions.

Inherits from A4A.K1 and so features a kernel1.

Modbus TCP Server port : 1503 (default)

8.9. 021 a4a-k1-gui

This application is meant to play with **010 a4a_piano**.

It shares **020 a4a-k1-cli** logic and provides a Graphic User Interface.

Modbus TCP Server port : 1503 (default)

8.10. 022 a4a-k1-wui

This application is meant to play with **010 a4a_piano**.

It shares **020 a4a-k1-cli** logic and provides a Web User Interface.

Modbus TCP Server port : 1503 (default)

Web Server port : 8080

8.11. 030 app1simu-cli

Here is the simulation application for the example application 1.

This is the Command Line incarnation. There are GtkAda UI and Web UI versions.

Inherits from A4A.K0 and so features a kernel0.

Modbus TCP Server port : 1505

8.12. 031 app1simu-gui

This is the GtkAda UI incarnation of simulation application for the example application 1.

Modbus TCP Server port : 1505

8.13. 032 app1simu-wui

This is the Web UI incarnation of simulation application for the example application 1.

Modbus TCP Server port : 1505

Web Server port : 8083

8.14. 040 app1-cli

This is where you will find the files of the example application 1.

This application implements a Modbus TCP Server and Modbus TCP IO Scanning.

This is the Command Line incarnation. There are GtkAda UI and Web UI versions.

Inherits from A4A.K1 and so features a kernel1.

Modbus TCP Server port : 1502

8.15. 041 app1-gui

This is the GtkAda UI incarnation of example application 1.

Modbus TCP Server port : 1502

8.16. 042 app1-wui

This is the Web UI incarnation of example application 1.

Modbus TCP Server port : 1502

Web Server port : 8082

8.17. 052 a4a_hilscherx_piano

This project inherits from the A4A project adding the Kernel 6 featuring one Hilscher cifX channel.

It has same functionality as **010 a4a_piano** excepted the fieldbus connexion.

Web Server port : 8090

8.18. 062 a4a-k3-wui

This application is meant to play with **052 a4a_hilscherx_piano**.

The project inherits from the A4A project adding the Kernel 3 featuring a Modbus TCP Server and one Hilscher cifX channel.

It has same functionality as **022 a4a-k1-wui** excepted the fieldbus connexion replacing the Modbus TCP Client one.

There are two variants : cifX and netXTransport.

Modbus TCP Server port : 1503 (default)

Web Server port : 8085

8.19. 070 app2-cli

In this one you will find the files for example application 2.

It implements a Hilscher PROFIBUS DP Master cifX card and a Modbus TCP Server and demos a PROFIBUS DPV2 communication with a Endress+Hauser radar level meter.

This project inherits from the A4A project adding the Kernel 3 featuring a Modbus TCP Server and one Hilscher cifX channel.

This is the Command Line incarnation. There is a GtkAda UI versions.

Modbus TCP Server port : 1503 (default)

8.20. 071 app2-gui

This is the GtkAda UI incarnation of example application 2.

8.21. 080 app3-cli

Here you will find the files for example application 3.

This application implements a Modbus TCP Server and Modbus RTU Master.

This is the Command Line incarnation. There are GtkAda UI and Web UI versions.

Inherits from A4A.K2 and so features a kernel2.

Modbus TCP Server port : 1503 (default)

8.22. 081 app3-gui

This is the Web UI incarnation of example application 3.

8.23. 082 app3-wui

This is the GtkAda UI incarnation of example application 3.

8.24. 090 app4

It is same as example application 2 but using a Hilscher PROFIBUS DP Master netHOST instead of cifX.

8.25. 100 app5-cli

This project inherits from the A4A project adding the Kernel 4 featuring a Modbus TCP Server plus Modbus TCP IO Scanning and one Hilscher cifX channel.

8.26. 101 app5-gui

8.27. 110 app6-cli

This project inherits from the A4A project adding the Kernel 5 featuring a Modbus TCP Server plus Modbus TCP IO Scanning and two Hilscher cifX channels.

8.28. 111 app6-gui

8.29. 122 app7-wui

This project inherits from the A4A project adding the Kernel 6 featuring one Hilscher cifX channel.

This application allows the control and monitoring of an EtherCAT drive via a Web User Interface.

8.30. 132 a4a-k7-wui

This application implements a gateway with Modbus TCP IO Scanning connected to **010 a4a_piano** and one Hilscher cifX channel connected to **062 a4a-k3-wui**.

It has a web interface.

Inherits from A4A.K7 and so features a kernel7.

Web Server port : 8085

8.31. 142 a4a_k0_S7

This application implements a gateway with Modbus TCP Server and a S7 Communication channel, thanks to Snap7, connected to SIEMENS S7 PLC.

It has a web interface.

Inherits from A4A.K0 and so features a kernel0.

Web Server port : 8085