

FROM RESEARCH TO INDUSTRY



# GENERATE IOC COMMUNICATION FOR SIEMENS PLC

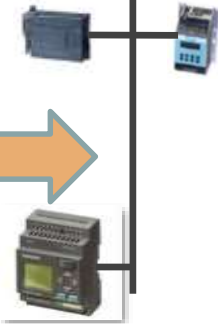
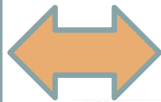


IOC  
Input Output  
Controller

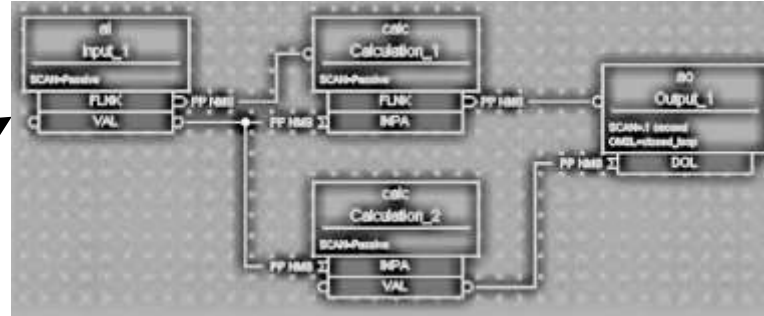


Data block  
communication

PLC  
Programmable  
Logic Controller

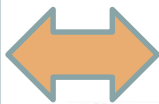
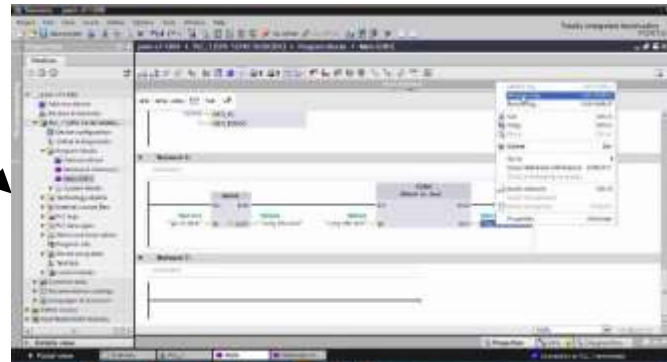


IOC  
Input Output  
Controller



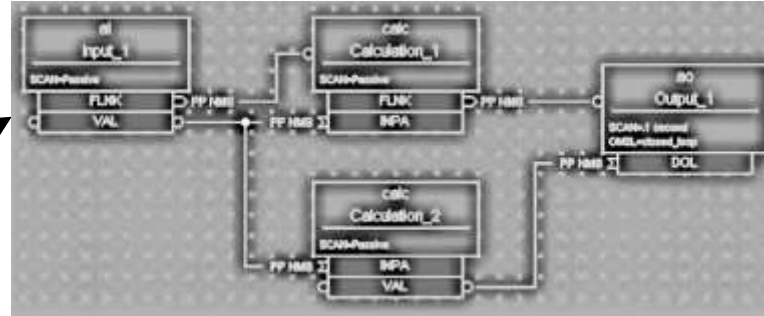
Data block  
communication

PLC  
Programmable  
Logic Controller



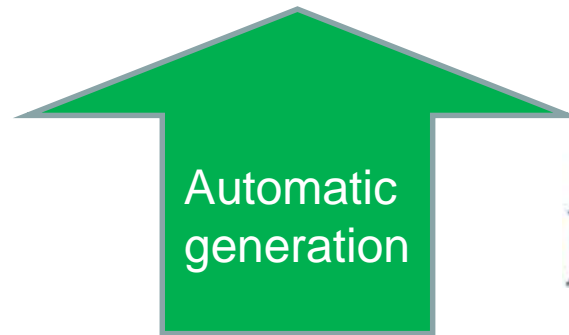
- 5 projects per year with PLC
- From 200 to 6000 variables
- Need a programming communication solution for non EPICS developer
- “Light” automatic solution

IOC  
Input Output  
Controller

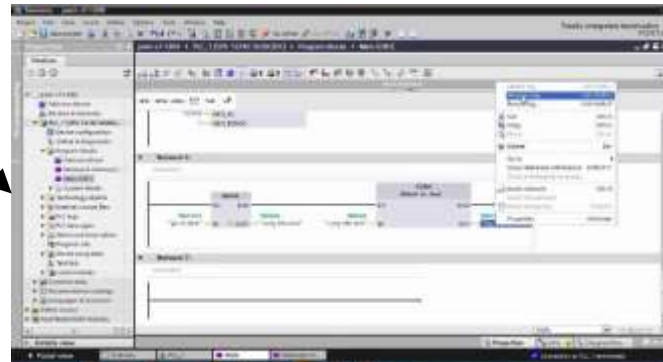
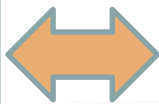


Data block  
communication

PLC  
Programmable  
Logic Controller

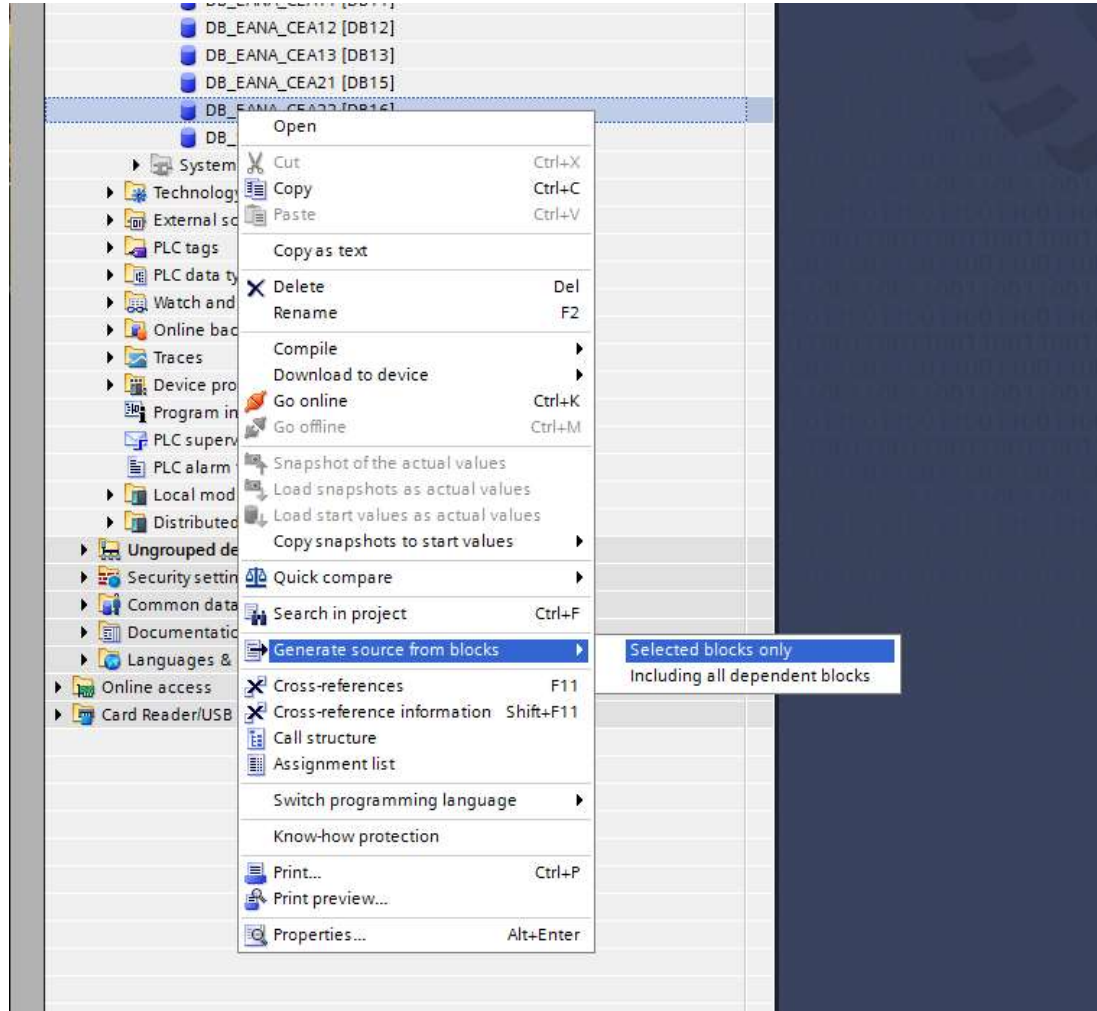


Automatic  
generation



- TIA Portal: 2 awl files that contains 3 information: name, type, description
- Application:
  - Calculate data offset in data block message
  - Add EPICS fields: SCAN, alarm, PINI
  - Possibility to add other fields
  - Communication: @IP, port, timeout,...

- Export TIA Portal



- Export TIA Portal

```
DATA_BLOCK "DATA_RECV_FROM_EPICS"
```

```
TITLE = SKID, Proj_Skid,,
```

```
{ S7_Optimized_Access := 'FALSE' }
```

```
AUTHOR : Joannem
```

```
FAMILY : ComEPICS
```

```
NAME : S7_PLC
```

```
VERSION : 0.1
```

```
NON_RETAIN
```

```
STRUCT
```

```
"SKID:PLC-001:R10P" : Real; // Regulation P
```

```
"SKID:PLC-001:R10I" : Real; // Regulation I
```

```
"SKID:PLC-001:R10U" : Real; // Regulation U
```

```
...
```

```
ENDSTRUCT;
```

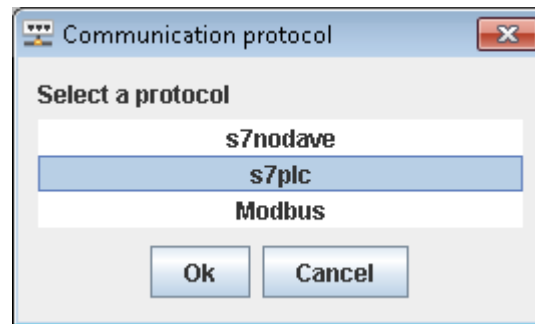
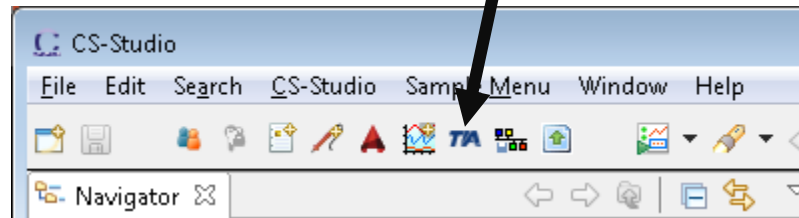
**SKID**: name of EPICS IOC

**Proj\_Skid**: PLC name in S7PLC

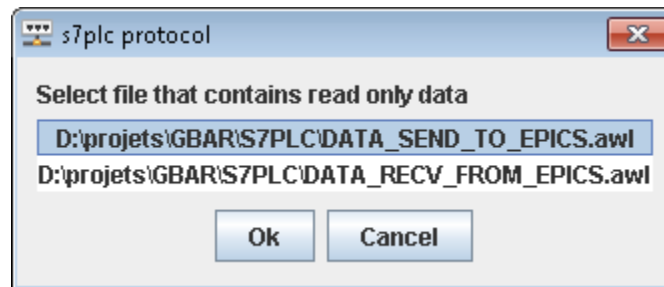
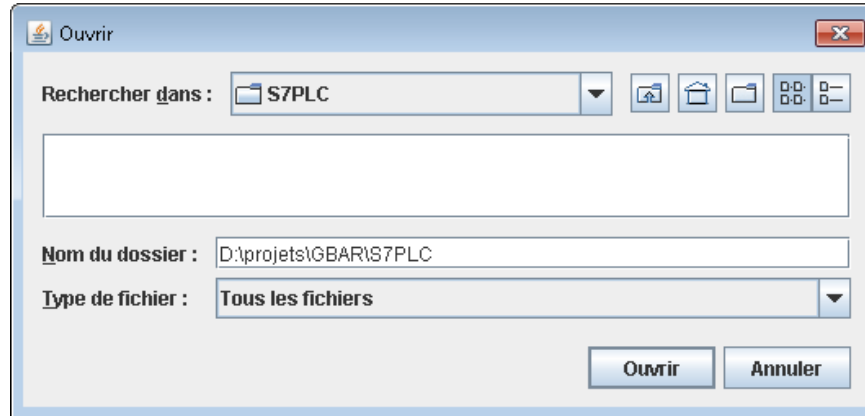




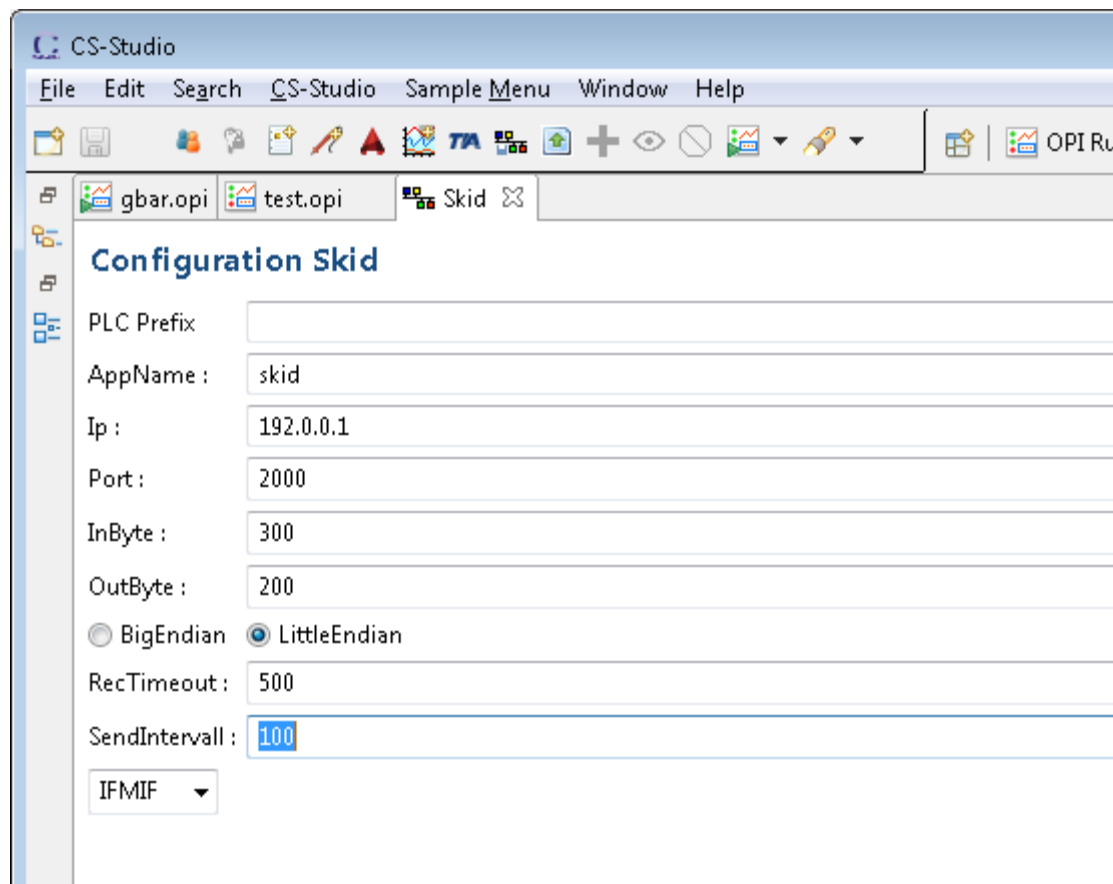
- Import awl file: choose protocol



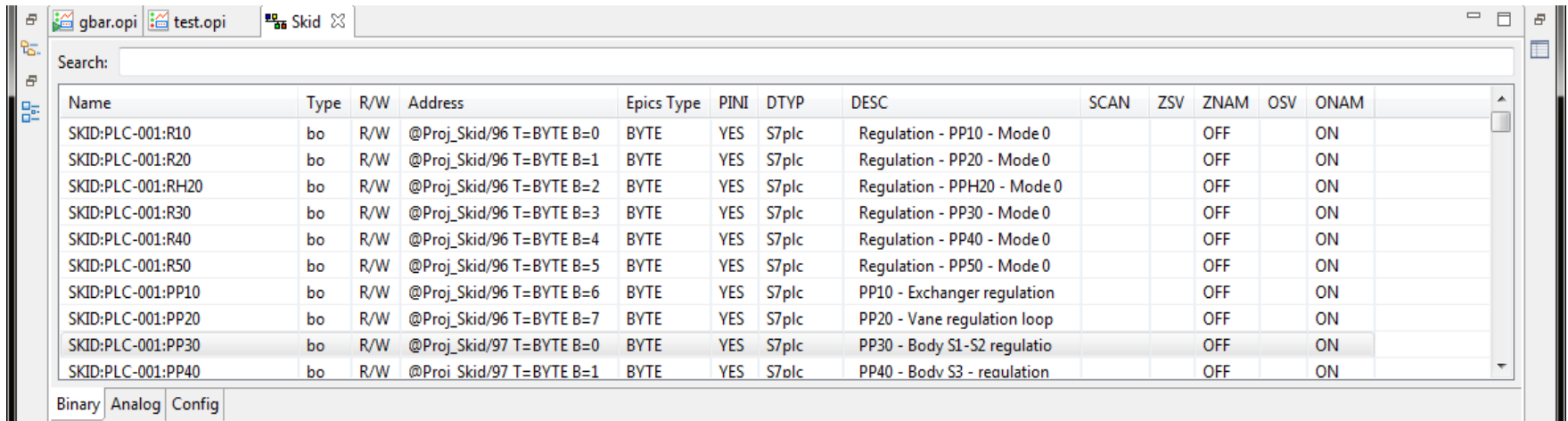
- Import awl file: choose file



- Import awl file: Define communication parameters



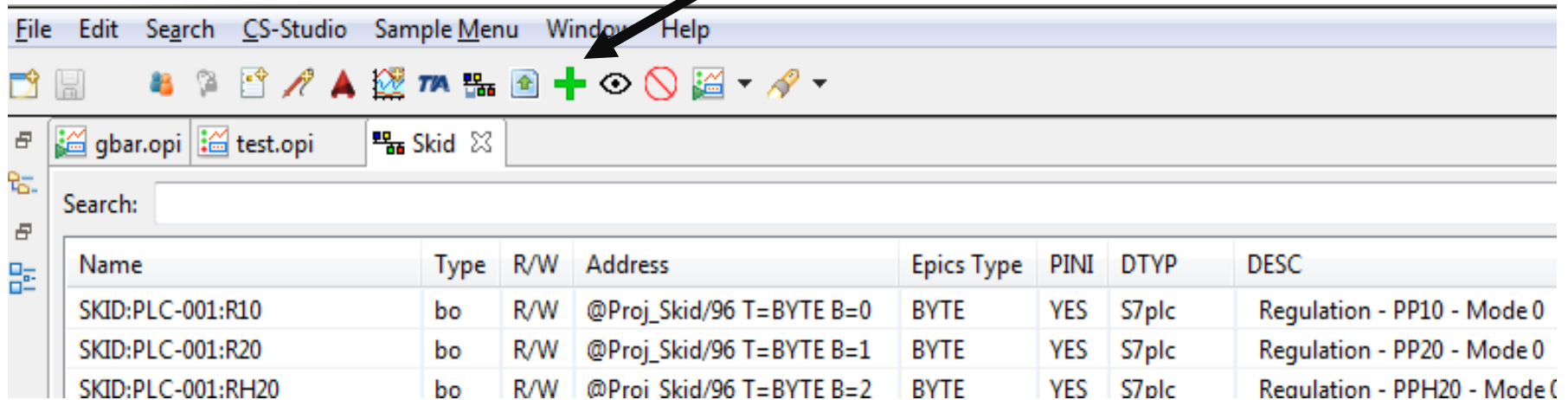
- Default EPICS parameters : SCAN, ZNAM, OSV, ONAM



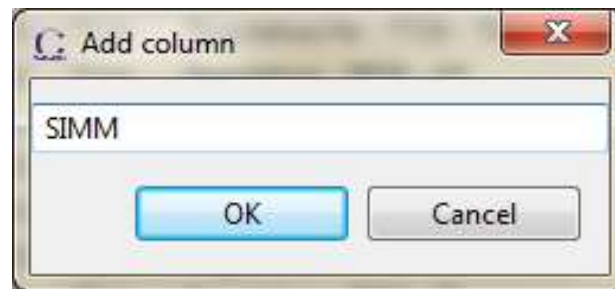
The screenshot shows a software interface with a search bar and a table of parameters. The table has columns for Name, Type, R/W, Address, Epics Type, PINI, DTYP, DESC, SCAN, ZSV, ZNAM, OSV, and ONAM. The parameters listed are SKID:PLC-001:R10 through SKID:PLC-001:PP40. The SCAN, ZSV, ZNAM, OSV, and ONAM columns are highlighted in grey, indicating their default values.

Name	Type	R/W	Address	Epics Type	PINI	DTYP	DESC	SCAN	ZSV	ZNAM	OSV	ONAM
SKID:PLC-001:R10	bo	R/W	@Proj_Skid/96 T=BYTE B=0	BYTE	YES	S7plc	Regulation - PP10 - Mode 0			OFF		ON
SKID:PLC-001:R20	bo	R/W	@Proj_Skid/96 T=BYTE B=1	BYTE	YES	S7plc	Regulation - PP20 - Mode 0			OFF		ON
SKID:PLC-001:RH20	bo	R/W	@Proj_Skid/96 T=BYTE B=2	BYTE	YES	S7plc	Regulation - PPH20 - Mode 0			OFF		ON
SKID:PLC-001:R30	bo	R/W	@Proj_Skid/96 T=BYTE B=3	BYTE	YES	S7plc	Regulation - PP30 - Mode 0			OFF		ON
SKID:PLC-001:R40	bo	R/W	@Proj_Skid/96 T=BYTE B=4	BYTE	YES	S7plc	Regulation - PP40 - Mode 0			OFF		ON
SKID:PLC-001:R50	bo	R/W	@Proj_Skid/96 T=BYTE B=5	BYTE	YES	S7plc	Regulation - PP50 - Mode 0			OFF		ON
SKID:PLC-001:PP10	bo	R/W	@Proj_Skid/96 T=BYTE B=6	BYTE	YES	S7plc	PP10 - Exchanger regulation			OFF		ON
SKID:PLC-001:PP20	bo	R/W	@Proj_Skid/96 T=BYTE B=7	BYTE	YES	S7plc	PP20 - Vane regulation loop			OFF		ON
SKID:PLC-001:PP30	bo	R/W	@Proj_Skid/97 T=BYTE B=0	BYTE	YES	S7plc	PP30 - Body S1-S2 regulatio			OFF		ON
SKID:PLC-001:PP40	bo	R/W	@Proi_Skid/97 T=BYTE B=1	BYTE	YES	S7plc	PP40 - Bodv S3 - reaulation			OFF		ON

- Add new EPICS field



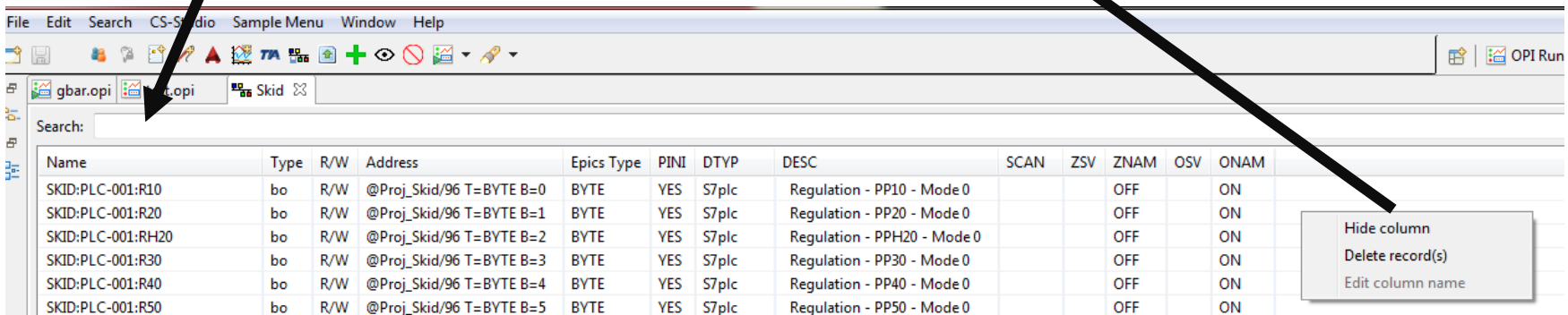
Name	Type	R/W	Address	Epics Type	PINI	DTYP	DESC
SKID:PLC-001:R10	bo	R/W	@Proj_Skid/96 T=BYTE B=0	BYTE	YES	S7plc	Regulation - PP10 - Mode 0
SKID:PLC-001:R20	bo	R/W	@Proj_Skid/96 T=BYTE B=1	BYTE	YES	S7plc	Regulation - PP20 - Mode 0
SKID:PLC-001:RH20	bo	R/W	@Proi Skid/96 T=BYTE B=2	BYTE	YES	S7plc	Regulation - PPH20 - Mode 0



- Tricks

- Search variable

- Right click: Hide column, delete columns



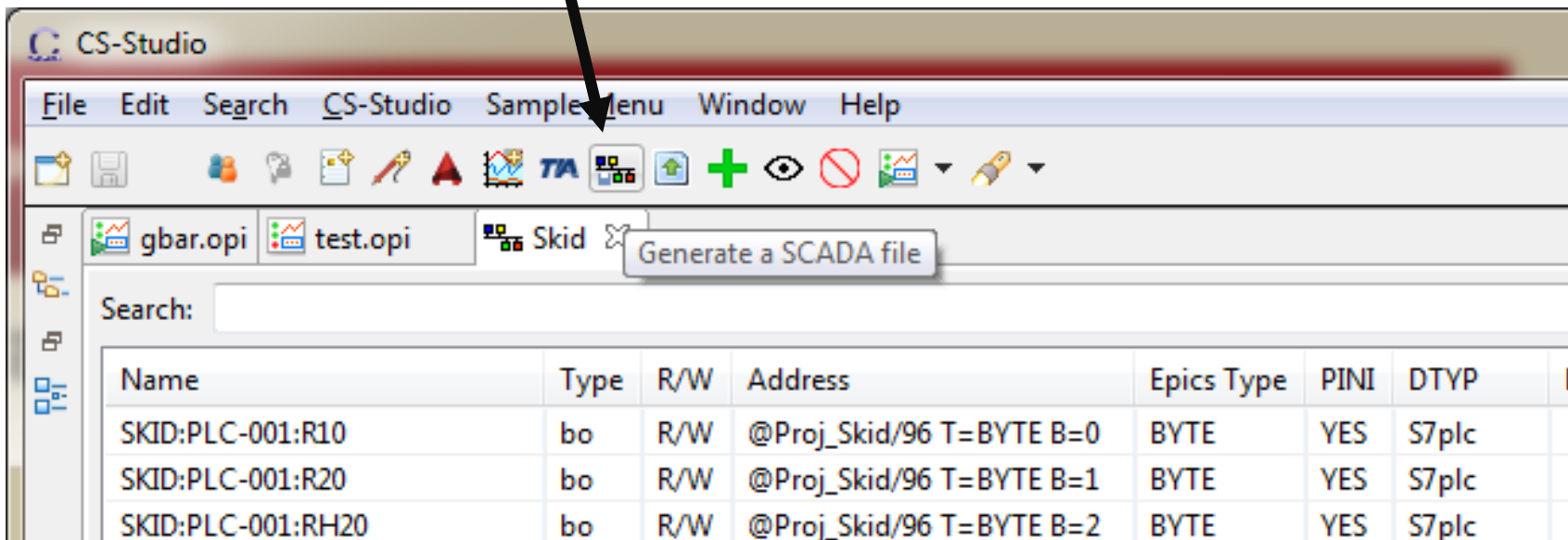
Search:

Name	Type	R/W	Address	Epics Type	PINI	DTYP	DESC	SCAN	ZSV	ZNAM	OSV	ONAM
SKID:PLC-001:R10	bo	R/W	@Proj_Skid/96 T=BYTE B=0	BYTE	YES	S7plc	Regulation - PP10 - Mode 0			OFF		ON
SKID:PLC-001:R20	bo	R/W	@Proj_Skid/96 T=BYTE B=1	BYTE	YES	S7plc	Regulation - PP20 - Mode 0			OFF		ON
SKID:PLC-001:RH20	bo	R/W	@Proj_Skid/96 T=BYTE B=2	BYTE	YES	S7plc	Regulation - PPH20 - Mode 0			OFF		ON
SKID:PLC-001:R30	bo	R/W	@Proj_Skid/96 T=BYTE B=3	BYTE	YES	S7plc	Regulation - PP30 - Mode 0			OFF		ON
SKID:PLC-001:R40	bo	R/W	@Proj_Skid/96 T=BYTE B=4	BYTE	YES	S7plc	Regulation - PP40 - Mode 0			OFF		ON
SKID:PLC-001:R50	bo	R/W	@Proj_Skid/96 T=BYTE B=5	BYTE	YES	S7plc	Regulation - PP50 - Mode 0			OFF		ON

- Display hide columns

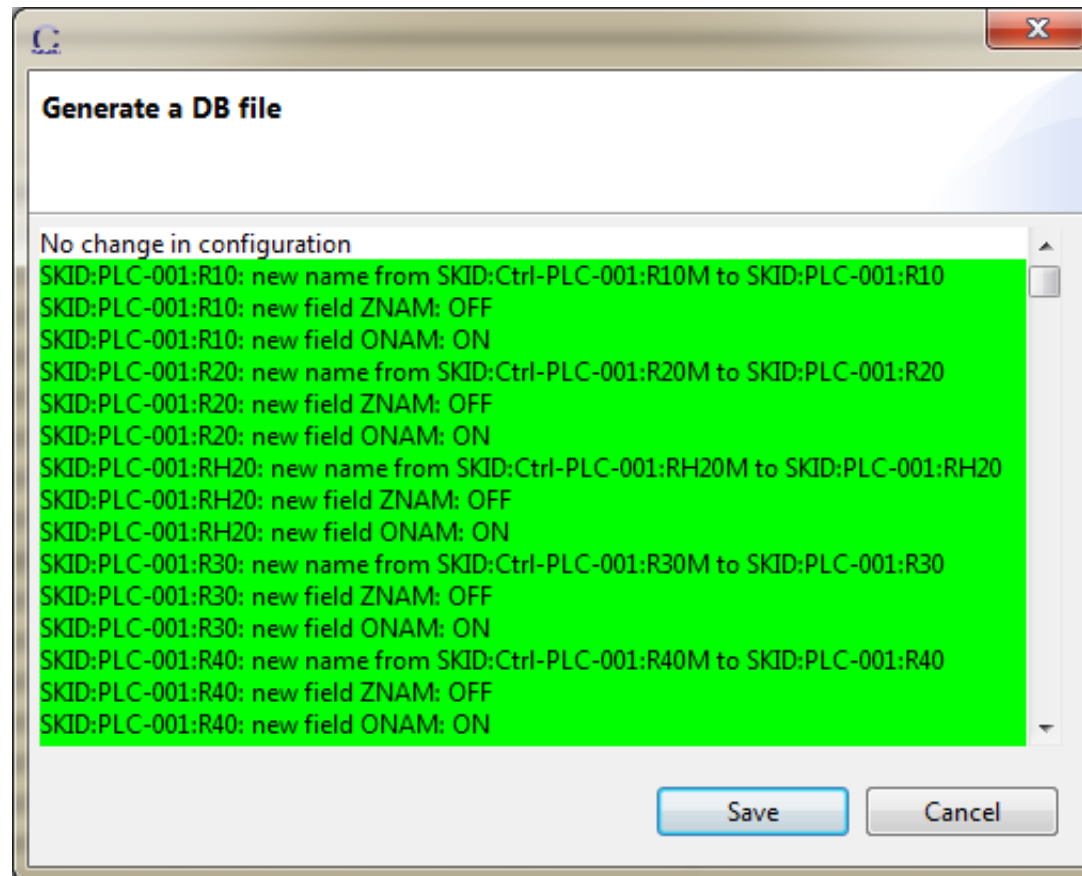
- Sorting and unsorting by columns

- Generate IOC

Name	Type	R/W	Address	Epics Type	PINI	DTYP
SKID:PLC-001:R10	bo	R/W	@Proj_Skid/96 T=BYTE B=0	BYTE	YES	S7plc
SKID:PLC-001:R20	bo	R/W	@Proj_Skid/96 T=BYTE B=1	BYTE	YES	S7plc
SKID:PLC-001:RH20	bo	R/W	@Proj_Skid/96 T=BYTE B=2	BYTE	YES	S7plc

- Generate IOC: display new parameters, change, and deleted





- Generate IOC:
  - iocBoot/iocSkid/boot/iocTest.cmd
  - iocBoot/iocSkid/src/iocTestInclud.dbd
  - iocBoot/iocSkid/src/Makefile
  - skidApp/db/skid.db
  - skidApp/db/Makefile

- **Generate IOC: db file**

```
#! DBDSTART
```

```
#! DBDEND
```

```
record(ao, "SKID:PLC-001:R10P"){  
  field(PINI, "YES")  
  field(DTYP, "S7plc")  
  field(DESC, " Regulation - PP10 - KP")  
  field(OUT, "@Proj_Skid/0 T=FLOAT")  
}
```

```
record(ao, "SKID:PLC-001:R10I"){  
  field(PINI, "YES")  
  field(DTYP, "S7plc")  
  field(DESC, " Regulation - PP10 - TI")  
  field(OUT, "@Proj_Skid/4 T=FLOAT")  
}
```

```
...
```

- **TODO:**
  - Bugs resolution in progress like calculation of inSize and outSize
  - Better merge
  - Better design: Model View Controller
  - Standalone without Eclipse
  - Modbus/TCP

Thanks for your attention