Saia-Burgess Controls AG
Bahnhofstrasse 18 I CH 3280 Murten I Schweiz

T +41 (0)26 672 71 11 I F +41 (0)26 672 74 99 I www.sbc-support.ch I www.saia-pcd.ch

PCD2_PCD3-Analogue_Signals-In IL E2

Projekt History

Datum	Author	Modifikation
10.12.2004	TCS / cd	Creation of project and documentation (version 1)
30.01.2009	TCS / sr	Adding the W525, update of the project to auf PG5 1.4.300 (version 2)

1. Summary

1.1 Functional description

This sample project is intended to show how analogue input and output signals can be read and/or written by different input cards.
To read and write analogue values using the analogue modules, it is essential to have a basic understanding of the way the I/O bus on a SAIA PCD Classic works. This topic is dealt with in section 3.1 (Reading from analogue modules).

The following individual modules are covered:

- PCD2/3.W1xx
- PCD2/3.W2xx
- PCD2/3.W3xx
- PCD2/3.W4xx
- PCD2/3.W5xx
- PCD2/3.Wxx5
- PCD2/3.W6xx

In most of the coding examples, one channel on the module is read/written to (with the exception of the code for the PCD2/3.W5xx modules).
The relevant channel is specified in the register Wx_Channel_to_read_1. In order to read/write to all channels, this register has to be incremented in each program cycle. This is not programmed in full in the present examples, in order to keep the code as modular as possible.

1.2 Possible application

The applications described in this example are basic applications, which can be used in any project with analogue input and output signals.
The examples are all written directly to a COB (Cyclic Organisation Block). To increase the modularity of the program, these codes can of course be integrated into FBs also.

1.3 Hardware and software used

Hardware:

- PCD3.M5540
- One of the I/O modules listed above, depending on the requirement
- PCD8.K111 programming cable or USB cable (for PCD3 or PCD2.M480)
- Where input modules are used, a signal transmitter is required.

To use these examples with a controller type other than a PCD3.M5540, the relevant type should be selected in the hardware settings for the project.
It is possible to load the configuration of the existing controller directly into the project, using the “Upload” function within the hardware settings (see section 2.1.2, Modification of hardware and software settings in PG5).

Minimum software version:

SAIA PG5 1.4.300

2. Structure

2.1 Preparing the PCD

The module to be used can be inserted into any socket on the PCD3.M5540.
The base address for the module has to be entered in the IL file for the module, or the associated symbol editor (the default is address 16 → the second socket).
More information regarding the modules as per example the wiring information are available on our support site www.sbc-support.ch.

The examples in this project are set out for the PCD2/3.Wx00 modules. These modules read a voltage input of 0..10V (or output a voltage). The value read/written is displayed in the user unit in mV.
Where a different signal is read, the only change required is to use the symbol editor to modify the corresponding signal (Wx_Measurement_Range_1) defining the maximum value for the user unit. More on this topic in section 3.2.3 (Converting the DV into real units).

The scaling of the module PCD2/3.W525 is adaptable according to the customers needs. More information about the scaling can be found in chapter 3.4.4.

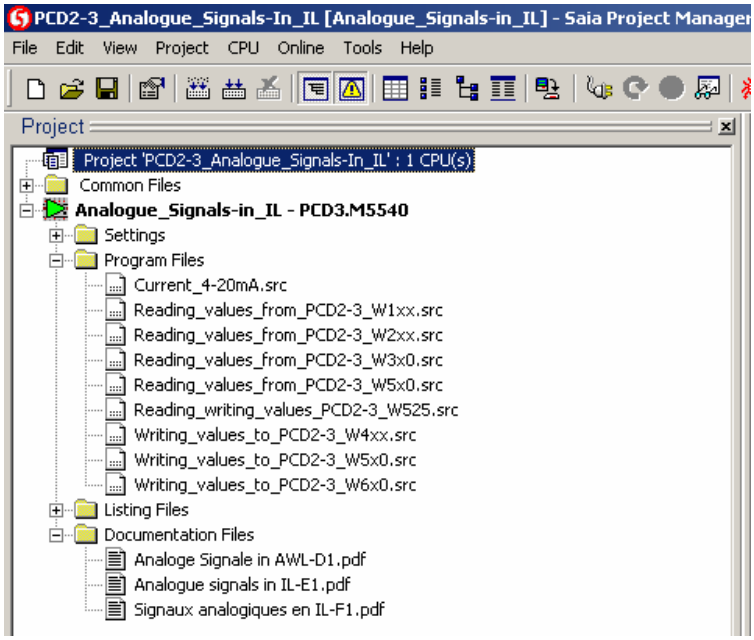
Analogue input modules are especially sensitive to earth loops. Please be sure to follow the wiring instructions.

The I/O bus on the SAIA PCD Classic controllers is not designed for “hot plugging”. Before inserting or removing I/O modules, the controller should be disconnected.

2.1.1 Installation of the project

To install the project in your PG5 project directory, you should use the “Restore...” function from the “File” menu in the PG5 1.2 Project Manager. This function will copy the project into your project directory.

If you are using PG5 1.1, you will need to unzip the project and copy it into the PG5 project directory manually.



This document can be found in the “Documents” folder in the project tree for the PG5 Project manager, and can be opened directly from there by double-clicking on it.

2.1.2 Modification of hardware and software settings in PG5.

The first step is to launch PG5 Project Manager (PG5 SPM).
The following procedure should then be followed:

- Connect the PCD to the PC with a PGU programming cable (PCD8.K111) or USB cable, and switch on.
- Open the “Hardware settings” screen (in the Settings folder in the project tree for the Project Manager).
- Select “Upload” to load the hardware configuration for the PCD onto the PC. Then save by clicking “OK”.
- Open the “Software settings” screen (in the Settings folder in the project tree for the Project Manager).
- Set the dynamic range for the resources by clicking on “Set Default”, and confirm by clicking “OK”.
- Link the Fupla file to be used (e.g. Reading_values_from_PCD2-3_W2xx.src, to read values from a PCD2/3.W2xx input module).
  Right-click on the desired Fupla program in the project tree” and select “Linked”.

2.1.3 Modifying the code for the current PCD system

To ensure that the correct module is read and that the user units are correctly configured, the following settings should be entered:

- The base address of the socket in which the module is located must be defined (W3_BaseAddress_1). The base address is a multiple of 16 and can be read directly from the motherboard on a PCD2.
  On a PCD3, only the sockets are numbered (starting from 0). Here, the base address of the I/O bus can be calculated by multiplying by 16 (module base address = 16 * socket address).
  Please note that the socket address on an expansion housing will be a continuation of the addresses from the previous housing.
- The user unit may also be modified. In the examples of the PCD2/3.Wxx0 modules, input signals of 0..10V are read in each case. By default, these values are expressed in mV.

2.1.4 Building and loading the project into the PCD

After making the adjustments set out in the preceding section, the project can be loaded into the controller following a “Rebuild All” (“CPU” menu, “Rebuild All...” option, or Alt+F2).

If the controller is already in a “Run” state, the system will ask whether the controller can be stopped. This will be the case during testing. The message is displayed for safety reasons, as it may not be permissible to stop the controller while in use on an existing installation.

One advantage of Instruction List programming is that programs can also be loaded while the PCD is running (Download in RUN). This requires all program components to be written in Instruction List.

2.2 Viewing values online

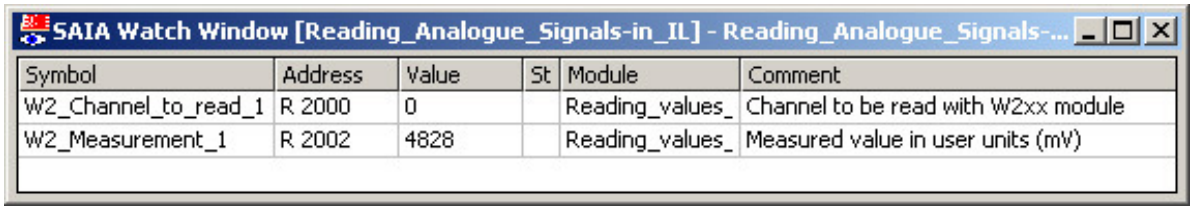
Once the program is loaded into the controller, an online connection can be established to the controller, in order to view the values online.
Clicking on the "Online" button (with the plug) connects the PC to the PCD. If the PCD is not yet in a Run state, it can be started with the curved green arrow on the toolbar.

The Watch Window (see next section) can now be used to view the values online.

Theoretically, it would also be possible to view the code online in the IL Editor. However, as the analogue I/O modules react to any access to the I/O bus, this should be avoided.

2.2.1 The SAIA Watch Window

To display and change the values from media on a screen, the Watch Window can be used. This can be opened from the “View” menu in the PG5 Project Manager. The symbols to be displayed can be “dragged and dropped” into the window.
As soon as PG5 is online, the relevant values will be displayed.

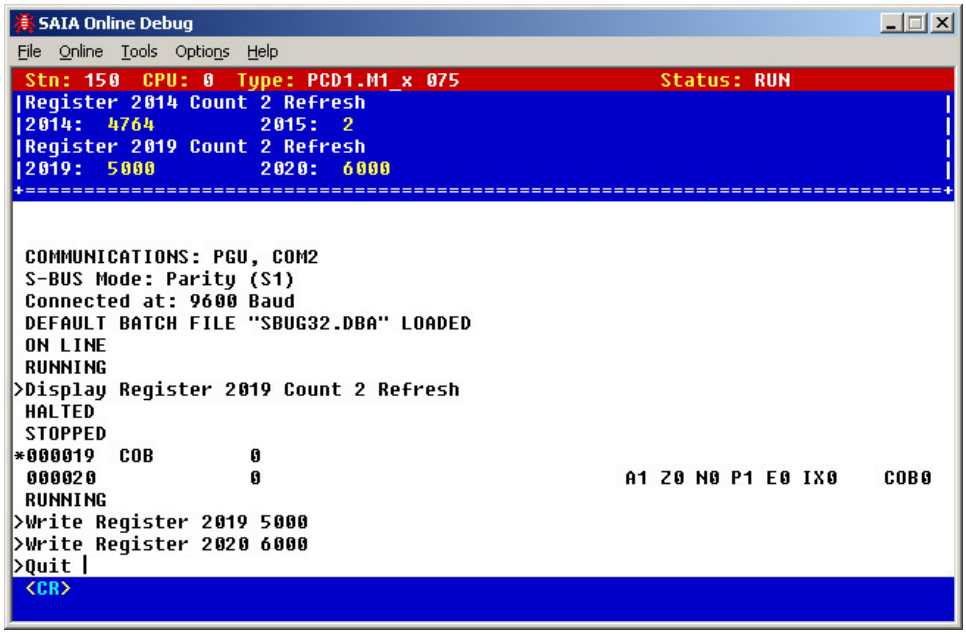


Symbol	Address	Value	St	Module	Comment
W2_Channel_to_read_1	R 2000	0		Reading_values_	Channel to be read with W2xx module
W2_Measurement_1	R 2002	4828		Reading_values_	Measured value in user units (mV)

Here again, no binary I/Os from analogue modules should be displayed on the I/O bus. This visualization would affect the behaviour of the analogue I/O modules.

2.2.2 The PG5 Online Debugger

Another way of displaying values online is the Online Debugger (select “Tools”, “Online Debug” in the PG5 Project Manager, or press F11).



The Online Debugger is a very versatile tool, providing the following options:

- Controlling the PCD (stop, start etc.)
- Displaying PCD media (registers, flags, DBs etc. can also be refreshed)
- Writing to PCD media
- Modifying the program (if it is held in RAM)
- Displaying the CPU status (hardware and firmware version etc.)
- Reading the PCD history
- Displaying the current hardware configuration
- Program “trace” (step-by-step processing; not allowed when accessing analogue I/Os)
- Processing individual instructions
- Running the program up to a given event (e.g. change of specified media or to setting of Error Status Flag)

To display the values read in the Online Debugger, the following input is required (enter only the characters in bold):
> **D**isplay **R**egister <**address of register to be displayed**> **R**efresh <Enter>

The address of the register to be displayed can be seen e.g. in the “Data List View” within the PG5 Project Manager (select “View”, “Data List”).

.

3. Functional description and settings

3.1 Principles of reading I/O modules

The SAIA PCD Classic controllers have an internal I/O bus with static addressing. Each module socket has 16 binary I/O points assigned, addressed in ascending order. This means that the module plugged into the first socket will have absolute addresses 0 to 15 assigned. Every I/O bus address can be read and written to.

The I/O points are accessed **immediately** they are read or written to by the user program.

The controllers in the SAIA PCD Classic family do not work with a process map.

This means that an I/O can be read and/or written many times during the same program cycle.

Where an address is read, but no input module is plugged in or the module is defective, the value 0 will be read.

3.1.1 Digital I/O modules

The data points for digital I/O modules are addressed directly.
Where e.g. output 0 is set in the user program, the first output will be switched via a digital output module plugged into the first socket.
Where a digital input module is plugged in, this will be returned when the status of the physical input is read.
With digital I/O modules with less than 16 inputs/outputs, the remaining addresses on the I/O bus are unused, and cannot be used by any other module either.

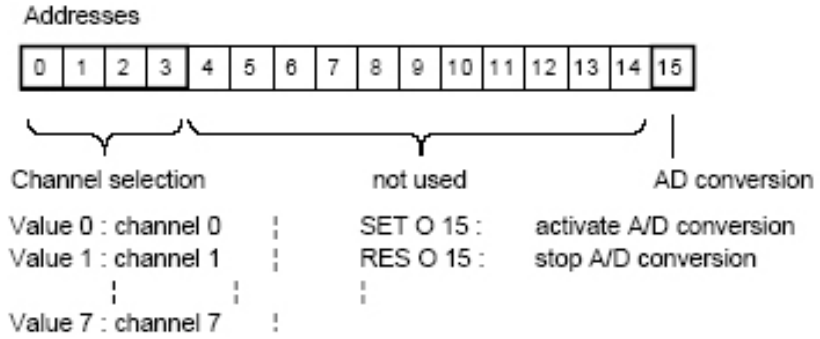
3.1.2 Non intelligent analogue I/O modules (Wxx0)

As the analogue I/O modules also have only the 16 I/O addresses described above, the values to be read/written (8..12 bit values) must be “multiplexed” to all channels. This also requires a facility to initiate e.g. an analogue-digital conversion. Depending on the module, specific digital addresses on the I/O bus (assigned to the module) are reserved for this purpose.

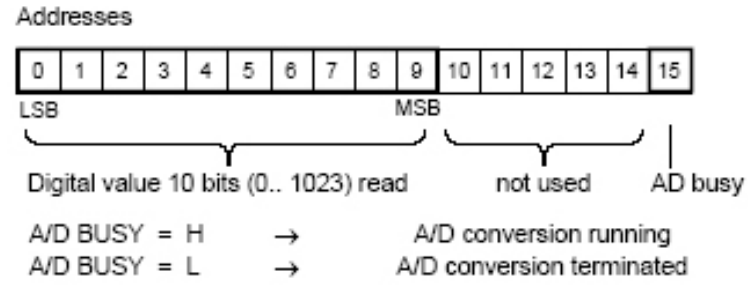
The meaning of the addresses on a PCD2/3.W2xx module is set out below. Writing refers to the selection of the channel to be read. The values can then be read from the I/O bus according to this mapping.

Meaning of the 16 addresses

- **Write:**



- **Read:**



In general, the analogue input modules for the SAIA PCD Classic controllers are handled in the following steps:

- Select the channel to be read/written on the module
- Activate analogue-digital conversion
- (Wait until conversion is complete)
- Read the result of the analogue-digital conversion via the I/O bus
- Convert the digital value into user-friendly units

Differences between the various analogue I/O modules
There are differences in the operation of the various module families (W1xx, W2xx etc.). For example:

- The resolution of the modules varies (8, 10 or 12 bit)
- The signal to activate the conversion is not always at the same address
- The conversion time of the analogue-digital transducer varies.

The modules should therefore be matched to the specific requirements.

Special mention should be made of the PCDx.W1xx module. The analogue-digital transducer on this module takes significantly more time than the transducers installed on the remaining modules. As waiting for the conversion would substantially slow down the processing of the program, the conversion is triggered in one program cycle but the result is only read in a later cycle.

3.1.3 Intelligent analogue I/O modules (Wxx5)

Another option is offered by the **intelligent analogue modules** (W3x5, W6x5, W525, W7xx), which process and scale the values read by means of a **processor on the module** itself. These modules have to be configured according to the application; this is also carried out via the I/O bus.
As the configuration is more complicated than the operation of non-intelligent modules, FBs are used in this case. The FBs required are defined in the FBox library of the analogue modules. From this library, the FBs are included into the IL source as shown in the example for the PCD2.W525.

3.2 The general structure of the example code

The code used in this sample project is always structured on the same principle:

- Document header
  Contains a short description, creation date and author, and the history of the document.
- Code segment for reading the analogue module
  Inserted into COB 0 (the first Cyclic Organisation Block) by command $COBSEG 0.
- For the PCD2/3.Wxx0: Conversion of the digital value by the analogue-digital transducer into user-friendly units. This section is also written to COB 0.
- Initialisation of media
  To avoid random values in significant registers when the PCD is started up. This initialisation is written by command $INIT to XOB 16 (Cold Start Block), which is always processed as the first program block (after a cold start).

The various files are distinguished mainly by different routines for reading the modules. The comments in these files describe the exact operations to be executed. For help on the instructions used, select the instruction in question and call up the online help for the IL Editor by pressing F1.

3.3 PCD2/3.Wxx0 modules

3.3.1 Convention for symbol names

The symbol names used in these examples have the following structure:

W2_Measurement_Range_1

- The “W2” at the beginning of the name indicates the type of module to be read.
- The main part of the name corresponds to the meaning of the symbol “Measurement_Range“ → measurement range for the input value
- The _1 at the end of the name is the index of the module (so the second W2 module on the PCD would be given index _2 etc.)

3.3.2 Reading values from an input card

As explained in section 3.1.2, the first step is always to select the channel by setting the appropriate bits on the I/O bus. The BITO instruction is generally used for this. BITO writes a specified number of bits from a register (starting with the LSB) to digital PCD media, in this case outputs.
The analogue-digital transducer is then activated. In most cases, this is the last bit in the I/O area. Depending on the module, this bit may be reset to 0 immediately, or only after the digital value (DV) has been read.
After the value has been converted, it is read from the I/O bus. This is generally performed with the BITIR instruction. This instruction reads the binary data points from the bus and writes them to a register, which then contains the DV (digital value). It is important that this register should contain the value 0 in the bits not written to by the BITIR instruction. Otherwise, an incorrect (much too high) value would result.

3.3.3 Converting the DV (digital value) into user units

As the output value from the analogue modules is always the digital value, this then needs to be converted into the units required by the programmer. This is done according to the following formula:

$$\text{Measured value} = \frac{\text{Digital value} * \text{Maximum measured value}}{\text{Module resolution}}$$

The units in this formula are as follows:
Measured value in user-defined units (in the examples, mV)
”Module resolution” and “Digital value” have no units
”Max. measured value” is in mV, like the measured value itself.

4..20 mA sensors

With 4..20mA transmitters, a further conversion is required. A suitable program is contained in the file “Current_4-20mA.src“. This conversion is based on a measurement range from 0..10000 user units (default result from the remaining sample code).

The result is expressed as 0..1000 $^{1}/_{10}$%.

3.3.4 Output of analogue values

The output of analogue values to I/O modules is handled on the same principle as reading. The difference is that in this case, the values are written to the bus and not read from it.

The program also needs to convert the value from user units into the DV before writing it.
This is done according to the following formula:

$$Digitalvalue = \frac{\text{Output value} * \text{Module resolution}}{\text{Maximum output value}}$$

The units in this formula are as follows:
Output value in user-defined units (in the examples, mV)
”Module resolution” and “Digital value” have no units
”Max. output value” is in mV, like the output value itself.

3.4 PCD2/3.Wxx5 modules (Intelligent analogue I/O modules)

3.4.1 Convention for symbol names

The symbol names used in these examples have the following structure:

W525_Measurement_Range_1

- The “W525” at the beginning of the name indicates the type of module to be read.
- Der Hauptteil des Namens entspricht der Bedeutung des Symbols “Measurement_Range“ → Messbereich des Eingangswertes
- The _1 at the end of the name is the index of the module (so the second W525 module on the PCD would be given index _2 etc.)

3.4.2 Initialisation

The command $USE _ana_use_d2w525fbs , "SfupAnl525.srx" implements the FBs from the FBox Library. All needed symbols which are part of this FBox Library, are defined as „external“ EXTN to add them to de project.

3.4.3 Macros

The project was realised using macros to avoid that the same program line has to be written several times and to have a better structure. Each macro can be called several times.

3.4.4 Scaling (W525)

The modules are configured in the XOB 16 using a hexadecimal code. For the interpretation of the code, please refer to the description below:

In/Out-mode coding:

0000H
- sub mode
- main mode
- filter mode (only for inputs!)
- scaling

Scaling:

0 = default scaling (The „default scaling“ values can be found in the FBox help)
1 = user scaling

Filter mode (only for inputs!)

0 = Fast-Mode (no filter)
1 = 50Hz / 60Hz filter
2 = Auto-filter

Main mode:

0 = Tension	0..10V
1 = Current	0..20mA / 4..20mA
2 = Resistor (or temperature)	PT1000 / PT500 / NI1000 / R (only Inputs)

Sub mode:

If "main mode" = Tension (0)	0 = 0..10V
If "main mode" = Current (1)	0 = 0..20mA
	1 = 4..20mA
If "main mode" = Resistor (2)	0 = PT1000 (-50..400°C)
	1 = PT500 (-50..400°C)
	2 = NI1000 (-60..200°C)
	3 = Resistor (0..2500Ohm)

Reset-Mode coding

0 = High Impedance (Output off)
1 = Output 0% of full-range
2 = Output 50% of full-range
3 = Output 100% of full-range
4 = Output user value. => The output is set with the command “cmdSetResetValue”.

In case the output is configured as current output, there has to be a load. If the output is open, the LED on the module is blinking.

4. Errors and debugging

To help to isolate and fix faults quickly, this section describes a number of frequently occurring errors.

4.1 Common errors

Here is a list of frequent causes of malfunction in the example described:

Error	Cause and resolution of error
The value read in from an input module is not in the desired units	The maximum value for the user-defined units is probably not defined correctly. Check the value of the constant Wx_Measurement_Range_x
The analogue output value is always =0 or maximum amplitude	May be caused by incorrect wiring of the module. Please check the wiring against the hardware manual for your PCD.
The analogue input value is always =0	
The module does not work properly and outputs incorrect values or no values at all	I/O addresses on the module may have been accessed by the Online debugger or the Watch Window. Ensure that the bus addresses of the module are not displayed online.
The value read from one or more analogue inputs jumps periodically and is not constant	This phenomenon could be caused by an “earth loop” in the system. Please check the earthing of your system. The earth to the “-“ side of the module must have a short and massive connection to the “-“ terminal on the PCD (no looping of the earth wire around the PCD!)

4.2 Troubleshooting / debugging

When troubleshooting, it is advisable to start with an underlying function and test further functions one by one. It makes sense to start e.g. by writing a small test program to read input values.
To be sure that there is a signal at the input, the input signal should be verified with a multimeter on commissioning.

The values in the PCD can be verified by means of the Watch Window or the Online Debugger.

4.3 Sources

The various procedures for operating the modules are specifically tailored to the hardware used in the modules concerned.

Descriptions of the IL instructions can be found in the online help for the IL Editor.

Hardware-specific details such as terminal assignments and wiring diagrams can be found in the hardware manual for the relevant controller.