AUTOMATING THE SOFTWARE VERIFICATION TEST PROCESS FOR SIL LOGIC SOLVERS FOR SUBSEA OIL & GAS APPLICATIONS

Marqués, Ricardo; Rodrigues de Seabra, Eurico Augusto; Machado, José
Universidade do Minho

Process Shutdown systems are used for shutting down subsea oil and gas wells and maintaining a safe state for the intervention systems. The safety and reliability of the technology used for process shutdown is one of the major arguments to earn the trust of the customer and consequently one of the main technological reasons to stay ahead of the subsea oil and gas business market.
SIL (IEC61508) is the measure of trust. This is where SIL certified logic solvers come into play.
For every project using SIL logic solvers specific software has to be made based on applicable standards and customer shutdown philosophies.
Before deployment the system has to be verified. The intention is to apply new methods for automating the testing of the software. This will provide run-time improvement of testing, reduction of the needed resources and direct reduced costs in ongoing projects.

Keywords: Automation project; IEC61508; SIL; Software

AUTOMAÇÃO DO PROCESSO DE VERIFICAÇÃO DE SOFTWARE PARA PROGRAMADORES LÓGICOS SIL EM APLICAÇÕES DE EXPLORAÇÃO DE PETRÓLEO SUBMARINA

Sistemas de encerramento do processo são usados para parar a produção dos poços de petróleo e gás e manter em segurança a funcionalidade dos sistemas de intervenção.
A segurança e a confiabilidade da tecnologia utilizada para a paragem da produção é um dos principais argumentos para ganhar a confiança do cliente e, consequentemente uma das principais razões tecnológicas para continuar na vanguarda do mercado de exploração submarina de petróleo e gás. É aqui que entram os programadores lógicos com certificação SIL.
Para cada projeto que utilize programadores lógicos SIL software específico tem que ser criado com base na normalização aplicável e nas filosofias de encerramento de processo.
Antes do fornecimento do equipamento de segurança a funcionalidade do sistema tem que ser verificada. A intenção é aplicar novos métodos para automatizar a verificação do software. Isso permite melhoria no tempo de execução dos testes, redução dos recursos necessários e redução de custos nos projectos em curso.

Palabras clave: Projeto automação; IEC61508; SIL; Software

Correspondencia: Prof. Eurico Seabra - eseabra@dem.uminho.pt

1. Introduction

The subsea production industry has human and environmental safety as main principles. This is called Process Shutdown Systems, or simply PSD. SIL-3 certified logic solvers are used for shutting down subsea oil and gas wells and maintaining a safe state for the subsea systems. Therefore, this system is responsible for avoiding blowouts, leakages or even explosions capable of causing major environmental damage and possibly taking human lives.

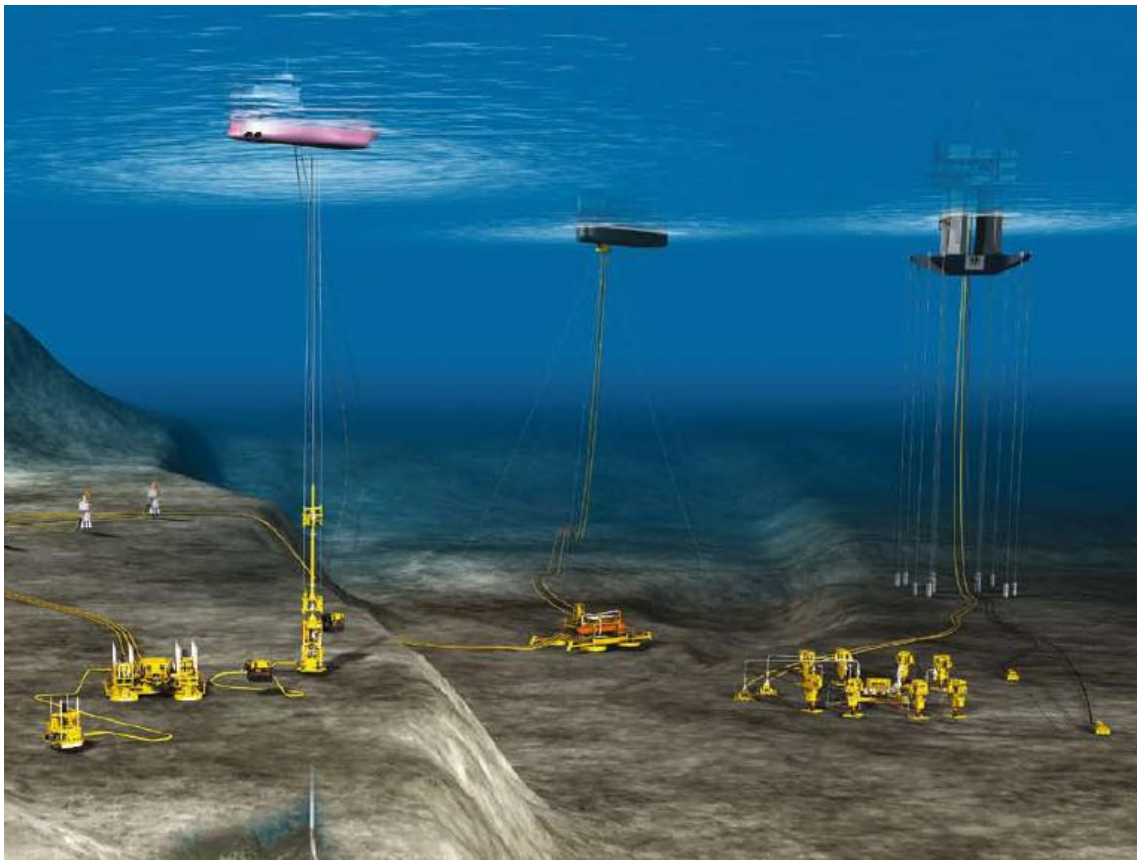
The basic operation of the logic solver is to perform a predefined sequence of functions when required, independently of the current system state and to perform the required function with the highest probability of success.

For every project a configuration file as to be created accordingly to the project technical and safety specific requirements. This file is then uploaded to the logic solver.

Before deployment of the safety equipment the functionality of the system has to be verified. This is to make sure that system works as intended and that it will respond as it is supposed when in service.

The purpose of this study is then to develop an automated test solution for the verification of safety related configuration files. This means the development of a software application to automate the testing procedures of the logic solver configuration files for SIS manipulating the hardware of the test unit using LabVIEW with its I/O building blocks from National Instruments. Therefore, the goal is to achieve run-time improvement, reduction of the needed resources to carry out the testing and therefore reduced costs in ongoing projects for safety applications in subsea oil and gas. Figure 1 shows a typical subsea installation of oil and gas wells.

Figure 1: Subsea Field Installation (FMC Technologies, 2010)



2. Safety Requirements

In most industrial processes, safety instrumented systems (SIS) are used as the first layer of protection in order to prevent an out-of control process from leading to serious accidents. Such systems include sensing elements, logic solvers and actuating devices. It is becoming more and more common practice to use dedicated hardware and software as logic solvers in process safeguarding applications. This means that the applied safety-related system needs to be highly reliable. This reliability can be considered from two points of view:

- Safety integrity;
- Safety availability.

Safety integrity means that if the safeguarded process is out of control, the PSD and the field devices are designed to bring the process to a safe state (which may mean shutdown). The higher the safety integrity (reliability) is the higher the probability that the system will function properly.
Safety availability means that a failure of the system causes the process to shut down (brought to a safe state) even though the process was perfectly under control. The PSD system should also be highly reliable in this respect, which means that the probability of an undesired process shutdown must be acceptably low. This aspect is called system availability.
In order to ensure that the safety instrumented system (SIS) meets all kinds of safety requirements, compliance with the applicable standards is required.

Standards such as IEC 61508 - “Functional safety of electrical, electronic, programmable electronic safety related systems” (Punch, 2013, Smith and Simpson, 2010) and IEC 61511 – “Functional safety – Safety instrumented systems for the process industry” (Harry, 2011, Pasquale, 2014). identify clear requirements that need to be met in order to prevent typical failures that may occur during all lifecycle stages of the SIS. The standards mentioned above have also defined 'Safety Integrity Levels' (SIL) (Houtermans, 2014, Honeywell, 2000), which denote the average probability of SIS failure to perform its design function on demand. The higher the SIL is the higher is the degree of protection.

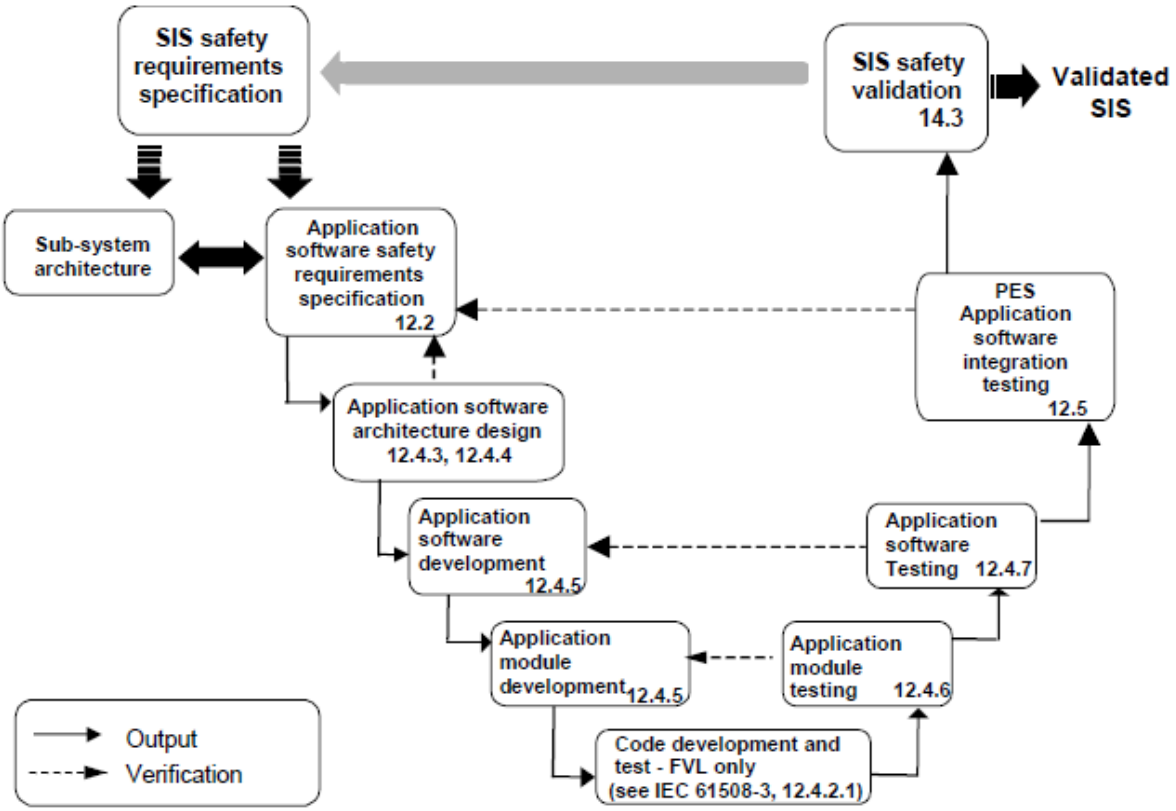
There are four specified levels of safety integrity requiring the highest level of instrumented safety. SIL relates to the number of failures of a safety system per demand.
Safety Integrity Level is a way to indicate the tolerable failure rate of a particular safety function. It is defined as four discrete levels of safety (1-4). Each level represents an order of magnitude of risk reduction. The higher the SIL level, the greater is the impact of a failure and the lower the failure rate is acceptable.

Every project and every oil industry company as customer have their own specific shutdown philosophies. This means that the logic solver units have to be configured for each specific application. A configuration implies testing to verify the correct behavior (Münch et al., 2012).

There are software languages in SIS subsystems as defined by the IEC61511 The fixed program language (FPL) in this type of language, the user is limited to adjustment of a few parameters (for example, range of the pressure transmitter, alarm levels, network addresses).
The limited variability language (LVL) this type of language is designed to be comprehensible to process sector users, and provides the capability to combine predefined, application specific, library functions to implement the safety requirements specifications. An LVL provides a close functional correspondence with the functions required to achieve the application.
And the full variability language (FVL) this type of language is designed to be comprehensible to computer programmers and provides the capability to implement a wide variety of functions and applications.

In this study, an automated test platform for compliance to IEC61511 with a logic solver using fixed program language (FPL) is considered. Figure 2 shows the software development lifecycle, according the standard IEC61511-1.

Figure 2: Software development lifecycle, from IEC61511-1 (IS/IEC 61511-1, 2013)



The automated test platform is intended for use in 12.5 (PES application software integration testing) according to the software development lifecycle from the IEC61511-1.

Application module testing (12.4.6) and application software testing (12.4.7) are covered under the SIL certification of the logic solver used in this study and therefore not considered for the automated platform.

3. The Safety logic solver

The SIL safety logic solver is defined in IEC 61511 as a Programmable Electronic System. The primary function of the logic solver is to perform the necessary I/O operations to control and maintain the safe state of one or more shutdown functions. In addition, is to be able to perform non-safety critical open and close operations on individual DCV's, relays, switches etc. controlling the final elements of the safety function, based on commands from the process control system (topside or subsea).

Taking actions and running predefined sequences based on inputs, it is intended for use as safety function logic solver for mainly subsea production control systems such as High Integrity Pressure Protection System applications (i.e. HIPPS).

One of the main applications of the SIL in subsea applications is the High Integrity Pressure Protection Systems (HIPPS). HIPPS are Safety Instrumented Systems (SIS) implemented to address overpressure scenarios instead of a pressure relief valve (PRV).

Figures 3 and 4 show, respectively, the logic solver or Programmable Electronics System and the HIPPS architecture.

Figure 3: Definition of logic solver or Programmable Electronics System, from IEC61511-1 (IS/IEC 61511-1, 2013)

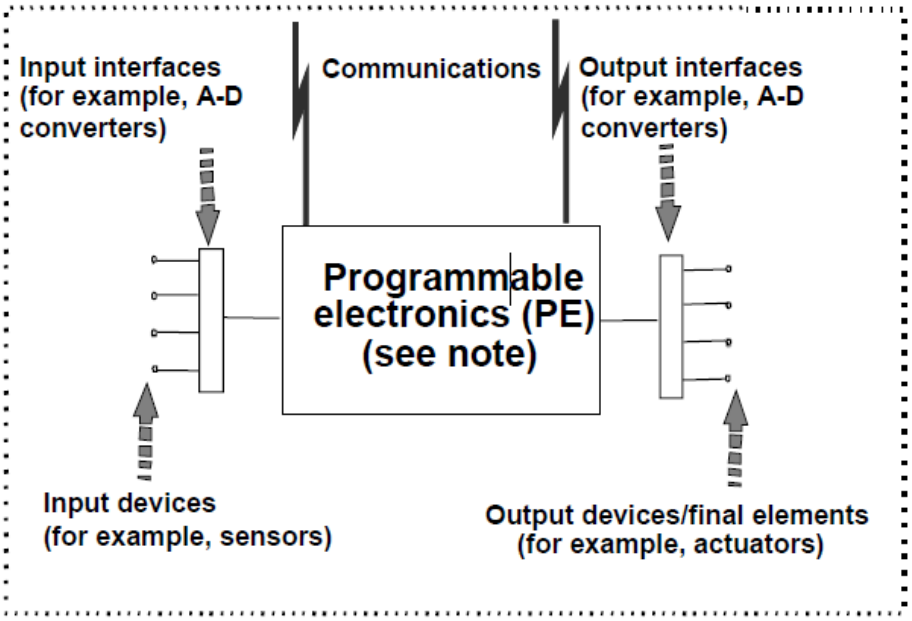
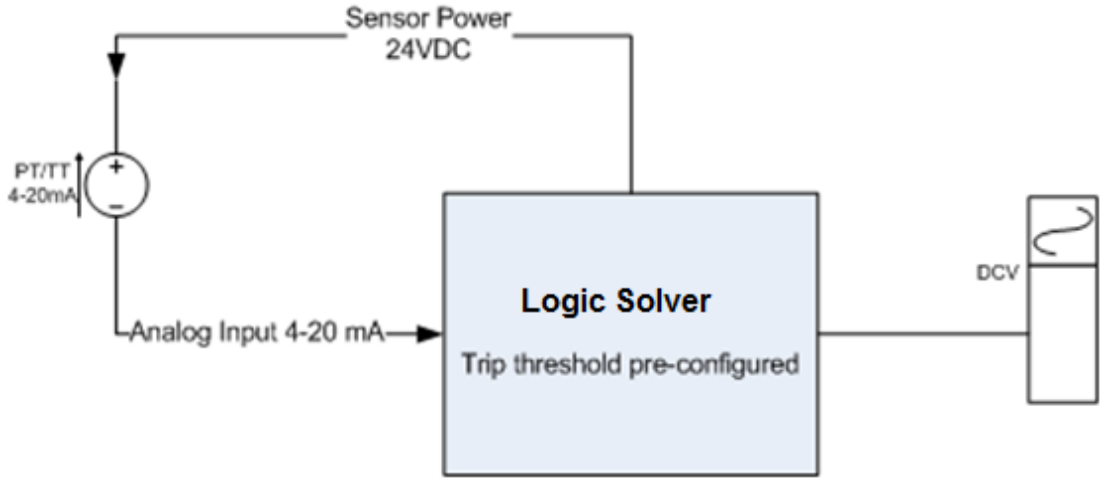


Figure 4: HIPPS architecture



3.1. The Configuration files and the need of testing

Unlike a Programmable logic controller where the application software is required to be configured (FVL and LVL logic solvers) the logic solver considered uses fixed programming language with a base application software already designed and certified according to SIL product development. The application software is product specific and was tested complying with SIL requirements. No software is coded for each project. To configure the logic solver for a specific functionality only the parametric configuration files are created. These files contain all the required parameters to set up the logic solver for operation. Only predefined parameters are defined in the configuration, i.e., which analog input, what triggering level,

which digital output to operate, what voting scheme and which sequences are running shutdown sequence...).

The main steps for setting up the configuration file are:

- Configure logic solver communication behavior;
- Configure subsea application sensors;
- Define the Safety Instrumented Functions based on the sensors configured;
- Define the sequences to execute upon shutdown scenarios;

There are three different configurations in the logic solver. The main configuration (main_config.hex) created in the hexadecimal format that defines the logic solver behavior and all the safety related operations.

There are two other Configuration files in the non-safe part of the logic solver, one is containing information regarding PSD ID's and communication settings. The other one is a setup file for the CANbus sensor interface.

Having a parametric file as the unique configurable item per project grants some advantages like reduced probability of errors while programming, less time spent on configurations, effectiveness on safety. But all these parameters are required to be tested and verified to be correct in order to comply with the SIL requirements defined in the IEC standards.

3.2 Test system solution

The Test Unit, shown in figure 5, is a transportable unit made for testing of safety configurations. Its approach was initially to start as a manual test unit, to be now developed into an automatic test unit, to be the test bed for Process Shutdown Configuration Files and be the process simulator for software development and test.

Figure 5: Test Unit

3.3 Specification

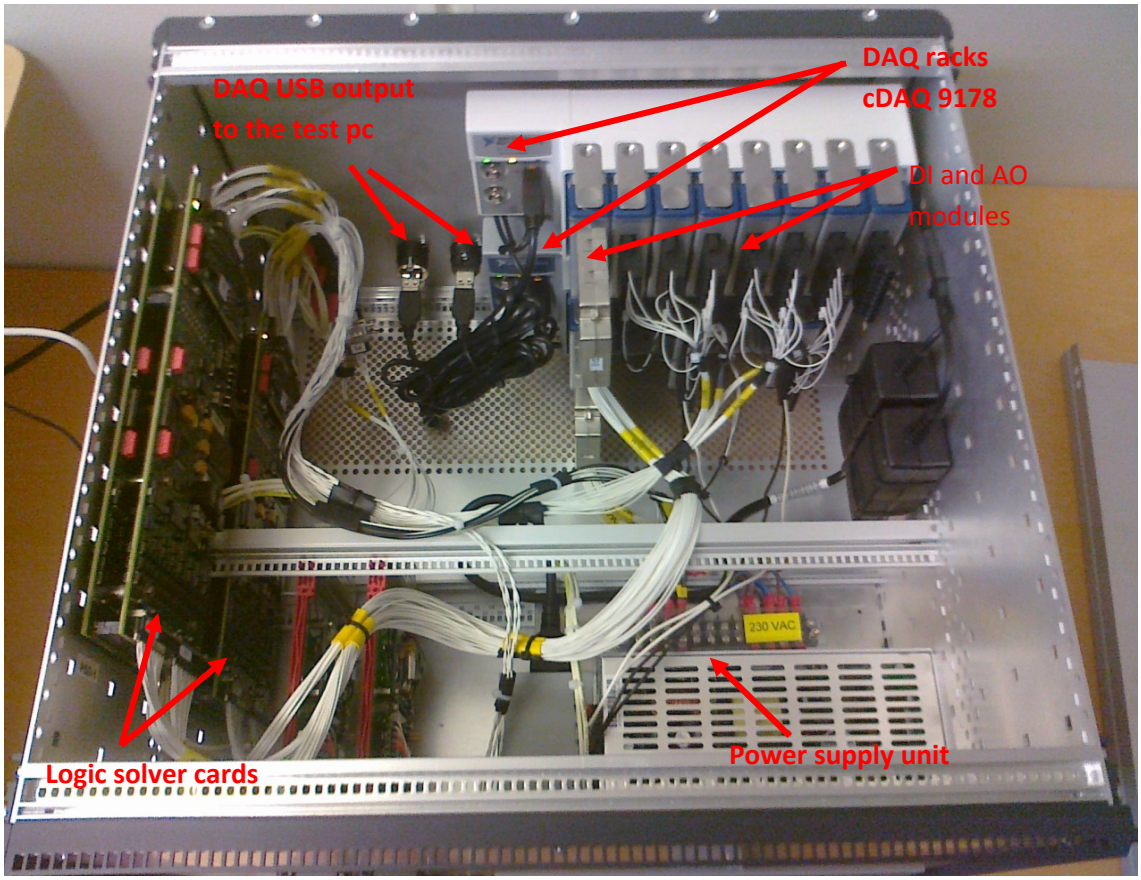
All I/O signals to/from the logic solver cards are simulated by I/O-modules (NI CompactDAQ system from National Instruments) controlled by software configured with LabVIEW (National Instruments, 2010). Figure 6 depicts the Test Unit Hardware.

Two SIL-rated subsea sensor simulators are connected to the CANbus interface on the logic solver cards. The simulators are controlled by voltage signals from an analog output module (NI CompactDAQ system).

External connection to the logic solver for monitoring and upload of configuration files is done via an Ethernet Switch. Communication with the NI CompactDAQ system is done via USB. The Test Unit basic components are:

- Power supply Unit;
- 2x SIL-3 approved logic solvers card with CANbus interface;
- 2x 8-Slot Chassis with digital/analog in/outputs;
- 2x SIL-rated PT/TT-simulator;
- Ethernet Switch.

Figure 6: Hardware of the test unit

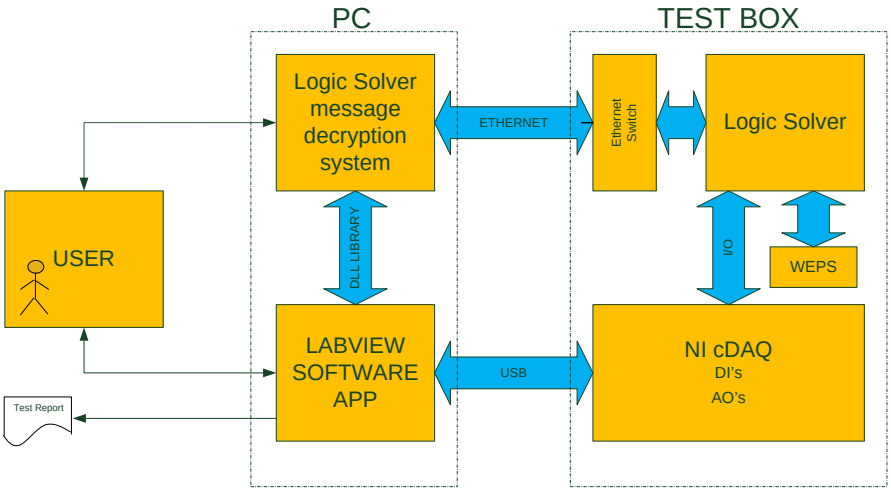


3.4 Test plan

The figure 7 illustrates how the LabVIEW based software application (Craig, 2009, Fayne et al., 2004) is connected to the NI DAQ hardware via USB connection. The NI DAQ hardware can be one or more different modules. In addition, the application is also connected to the dll library. The logic solver message decryption system receives command messages from the dll and communicates with the logic solver, via Ethernet to perform the demanded operations.

The LabVIEW application (FMC Technologies, 2011a, 2011b) will then read from the test unit, through the core monitor and the DAQ hardware, the I/O signals including the WEPS (PT/TT CAN bus subsea sensor simulators).

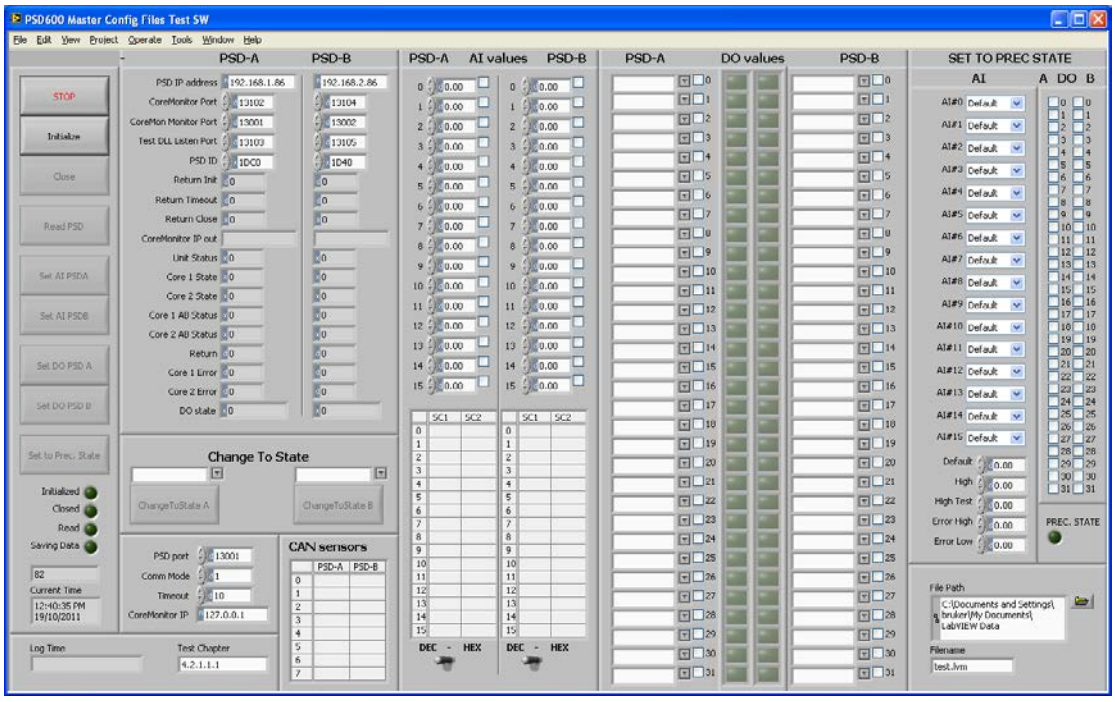
Figure 7: System Block Diagram



4. Testing the configuration files

For testing the configuration files it was developed a LabVIEW application, which control panel is shown in figure 8.

Figure 8: Testing software



The test was carried out using the Master Configuration Files Test SW.

Only one instance is needed to operate both logic solvers in the test unit. The test started at 14:10pm. The first setting to precondition state took about 2 minutes.

There is a specific area for the user to set the input pattern for the analog inputs and digital outputs. By pressing the “Set Prec State” button the SW sends the consecutive commands without the need of any other user input.

In the test project CAN sensors are used as input to the logic solvers. This has to be set manually on the sensor simulator software. The reason for this is because the dll used to

access the decryption system does not include the functionality of setting up for simulation the CAN sensors, for the moment.

The first chapter of the testing was finished at 14:20pm. The testing was finished at 14:40pm what means an overall time of 30 minutes.

It was noticeable that very little user intervention is needed and consequently less probability of wasting time by introducing human failures during the testing.

Another positive feature is the fact that a log file was automatically generated for this testing which is complying with the IEC requirements of keeping important records for proving the functionality of the safety application.

5. Conclusions

There were many challenges during the development. The first one was to understand the complex Safety requirements for SIL applications in the IEC 61511 standard.

The test unit was initially designed with data acquisition modules for 32 digital outputs and 16 analog inputs to each logic solver cards.

For the set to precondition state, no CAN sensors are automatically set because the CAN sensors simulation is not included in the dll. From the test experience, the full control of the CAN sensors through the software front panel would be a major improvement in time saving during testing.

The way the auto set to precondition state functionality has been designed the digital outputs states and analog inputs values are automatically set according to the pattern that the user defines. It is expected an even better performance in projects that use PT/TT sensors of the type 4 – 20 mA. This is explained because in that case the software will handle the precondition state automatically without the need of the use of any additional sensor simulation software. Nevertheless, the inclusion of the sensor simulation function in the test dll and further implementation in the test software shall be a forward step to take as a major improvement.

The log file is generated by the LabVIEW VI “Write to Spreadsheet File”. To create the logging functionality was one of the main challenges during the development of the software because the data to collect is divided in several different places in the block diagram. The use of local variables to get the proper data in the right place and the use of the different string manipulators to format the data conveniently turned out to be crucial for the creation of a readable log file. The log file is capable of giving a history of the executed steps, showing all the commands sent over to the logic solver cards and the readings showing the consequent logic solver states and I/O.

The test carried out was essential to measure the benefit of introducing an automated test platform.

The major time-saving functionality has shown to be the set to precondition state. In the previous method of testing, the test user needed to check the test procedure to check the values of the analog input sensors and the state of the digital outputs (valves) and then set them manually one by one. With the new software the input pattern is defined once and since that the test user can just click “Set Prec. State” button and all the commands are sent automatically.

The automated test platform eases the SIL compliance process with reference to the IEC 61511 ensuring automatic logging, error-free testing, minimal execution time and full verification of the configuration files.

6. References

Craig, K. (2009). Engineering Measurement Data Acquisition with LabVIEW Physical & Mathematical Modeling Engineering Measurement, National Instruments, USA.

Fayne, E., Mills, L.J., Ireland, P., McGrory, J., Ventura J. (2004). LabVIEW: Instrumentation Lab and Industry, National Instruments, USA.

FMC Technologies (2010). Test Procedure, Communication, Subsea – Controls, BP Germ SIS PSD Master Config Files, Doc. No. TST60054905/A, Dated 19/02/2010, FMC Technologies, USA.

FMC Technologies (2011a). Specification, LabVIEW Application for FPL Logic Solver Configuration Test Unit", Doc. No. SPC60072044/A, Dated 10/06/2011, FMC Technologies, USA.

FMC Technologies (2011b). Logic Solver Test Dll Header File, ver. 1.3., FMC Technologies, USA.

Harry, L.C. (2011). IEC 61511 Compliance and SIL VerificationTurboSentryTM Overspeed Protection Device, Cteris Consulting Inc., Ontario, Canada.

Honeywell (2000). 2oo4D: A New Design Concept for Next-Generation Safety Instrumented Systems", 7/2000, Honeywell, USA.

Houtermans, M.J.M., (2014). SIL and Functional Safety in a Nutshell, Risknowlogy Best Practices, 1st Edition.

IS/IEC 61511-1 (2003). Functional Safety - Safety Instrumented Systems for the Process Industry Sector, Part. 1: Frameworks, Definitions, System, Hardware and Software.

Münch, J., Armbrust, O., Soto, M.; Kowalczyk, M. (2012). Software Process Definition and Management, Springer.

National Instruments (2010). Getting Started with LabVIEW", 6/2010, National Instruments, USA.

Pasquale, F. (2014). IEC 61508 and IEC 61511: Application State and Trends, Invensys Operations Management, Sesto San Giovanni (Milan), Italy.

Punch, M., (2013). Functional Safety for the Mining Industry – An Integrated Approach Using AS(IEC)61508, AS(IEC) 62061 and AS4024.1, 3rd Edition.

Smith, D., Simpson, K. (2010). Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety, IEC 61508 (2010 Edition) and Related Standards, Including Process IEC 61511 and Machinery IEC 62061 and ISO 13849, 3rd Edition.