EXAMINATION INFORMATION PAGE

Home exam / Portfolio assessment / Report / Semester assignment

University of South-Eastern Norway

Subject code: IIA2017	Subject name: Industrial IT	
Responsible subject teacher: Nils-Olav Skeie	Campus: Porsgrunn	Faculty: TNM
Assignment given in WISEflow (date and time): 25-MAY-21 / 9:00	Submission time in WISEflow (date and time): 25-MAY-21 / 14:00	
No. of assignments: 11 tasks	No. of attachments: 0	No. of pages incl. front page and attachments: 7

Aids and collaboration:

Permitted aids: All aids are allowed.

	Yes	No
Is it an individual exam?	☒	☐
Is it allowed to collaborate with other persons?	☐	☒

Description of individual examination and illegal cooperation will be found at [my.usn.no](my.usn.no)

Criteria for the answers:

Font type: Calibri or Times New Roman	Font size: 11 or 12	Line spacing:
No. of words (min/max):	Maximum no. of pages excl. front page and attachments:	

Source reference:

All external sources, including your own assignments, should be referenced using standard referencing techniques.

Other important information:

- Include/merge all information into one single pdf document.
- **Make sure to include your answers in ascending order, i.e., Task 1, 2, 3 etc.**
- No need to include task descriptions nor front page, only the task numbers with your answers.
- The format of the document **must** be pdf
- The WISEFLOW system will do a plagiarism control of all pdf files so always use referencing when copy information from any source.
- **Any communication with others, electronically or oral, is cheating!**

Part #1 – System overview and design (55%)

Process description

You are employed as a project engineer in the SMART research group at USN. Your job will be to develop a control system for a new experimental rig in the process hall at USN campus Porsgrunn. The focus will be research on control systems for autonomous systems using machine learning (ML) algorithms and digital twins. These ML algorithms will require a lot of data, so digitalization will be important for the system. A digital twin is a process simulator that runs in parallel with the physical process, using data from the process, so digitalization is important for the simulator as well.

The purpose of the experimental rig will be to test an autonomous system when including random disturbances like liquid leakages. The Process and Instrumentation Diagram (P&ID) for the process is shown in Figure 1 and consists of tank V02022 where hot and cold water is mixed by the agitator B02022, and two control loops: one for level control including pump P02009, and one for temperature control including valve TZV02003. The mixer B02022 should just be turned on with a digital signal when starting the system. The disturbances for the system can be controlled by the valves HZV02010, HZV02011, HV02012, HV02013 and HZV02014. Valves HZV02010 and HZV02011 are also used for shutdown of the system.

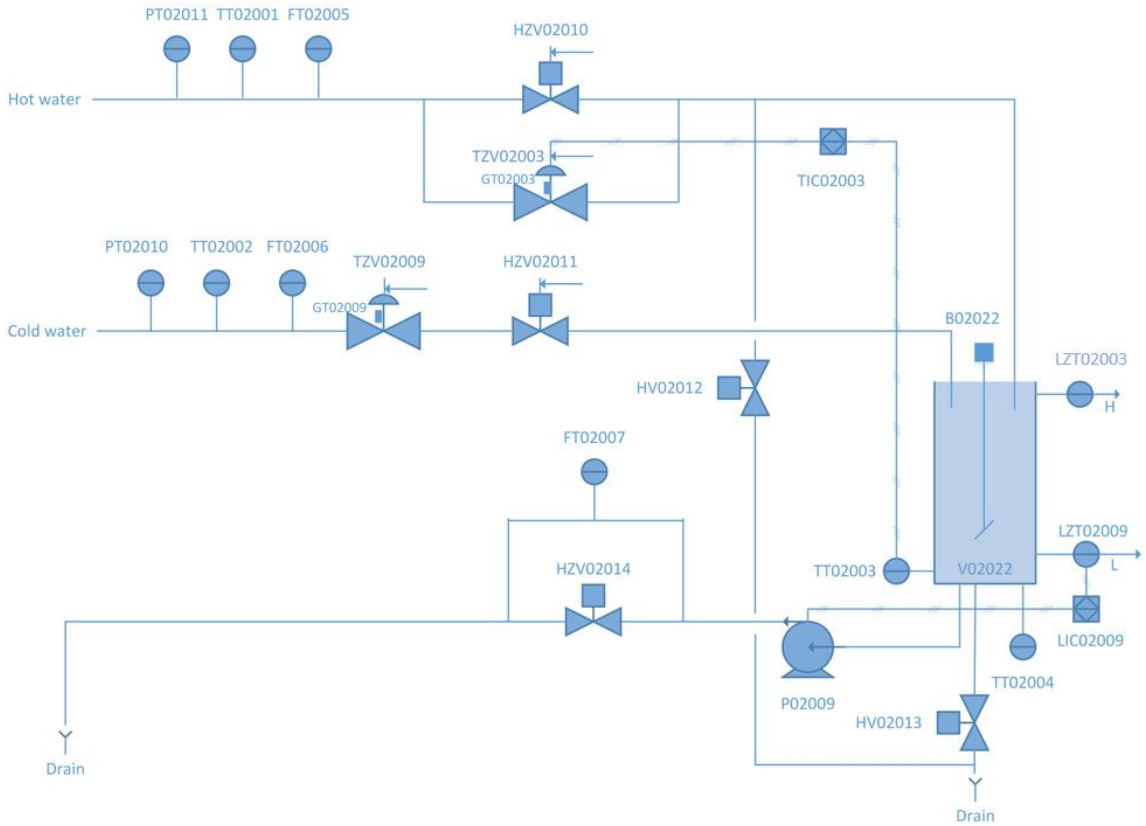

Figure 1: The P&ID for a laboratory process in the process hall at USN

Hot water inlet line:

This line includes pressure, flow, and temperature measurement where “smart” instruments are used. The control valve TZV02003, installed with a “smart” valve positioner, is a part of the temperature control loop TIC02003 and is interlocked to an emergency shutdown system (see arrow indicator above the valve).

GT02003 indicates the valve stem position (analog continuous measurement). The on/off-valve HZV02010 is also connected to the emergency shutdown system.

Cold water inlet line:

This line includes pressure, flow, and temperature measurement as well, and “smart” instruments are also used here. The control valve TZV02009 (also used with a “smart” valve positioner) is mainly used as a cold-water bias and as a disturbance for the temperature controller. GT02009 indicates the valve stem position (analog continuous measurement). The on/off-valve HZV02011 and the control valve is interlocked to the emergency shutdown system.

Mix water outlet line:

The pump P02009 is a part of the level control loop LIC02009 that controls the water level in the tank using a “smart” frequency converter. Only flow measurement is included, using a “smart” flow transmitter.

Mixing tank:

The hot and cold water is mixed by an agitator, B02022. A “smart” level transmitter, LZT02009, serves as input to the level controller, but also indicates low water level to the safety system. A level guard, LZT02003, indicates high water level to the safety system. Temperature transmitters TT02003 and TT02004 (“smart” transmitters) have different dynamic properties, which makes it possible to investigate how such a property can affect the control loop behavior.

Process disturbances:

- On/off-valve HZV02010 can simulate a control valve that is stuck in the open position
- On/off-valve HV02012 can simulate a line leakage upstream the vessel
- On/off-valve HV02013 can simulate a vessel leakage
- On/off-valve HZV02014 can simulate a line leakage downstream the pump

Your job as a project engineer will be to analyze, design and document a process control system that can be used for the experimental rig in Figure 1. The focus for the process control system is digitalization, to get data for the ML algorithms. Connection to an Enterprise Resource Planning (ERP) system, either local or cloud based, should also be considered for saving information. This means that both operation technology (OT) and information technology (IT) should be evaluated and included in the system.

Part of the documentation will be pseudo coding which is an abstract programming form used for documenting software algorithms or applications for humans. Focus in pseudo coding is the data flow, so use understandable names on variables and methods. Syntax is not important but be consistent. It is possible to use C# as basis for pseudo coding.

The analysis, design, implementation, and documentation should be based on the tasks given below.

Task #1: Introduction (13%)

A process control system (PCS) consists of a set of building blocks like a database, supervisory control and data acquisition system, alarm system, data communication and remote terminal units to mention some.

a) (3%) Illustrate with a sketch the interconnection of the building blocks that you will need for a PCS for the process shown in Figure 1.
b) (3%) Describe the functions and the data flow between the building blocks that you have included in your sketch.
c) (3%) Your PCS will need a data engine. Describe the main functions of such a data engine.

d) (2%) Discuss the advantages of using an ERP in such a system.
e) (2%) Data security is an important part for such PCS, any comments on protecting the information in your PCS.

Task# 2: Analysis and design (7%)

Analysing and designing such a PCS will start with a set of requirements, based on the specification given in the process description. The requirements can be split in functional and non-functional requirements.

a) (4%) Describe at least five functional requirements for your PCS, based on the process description.
b) (3%) Describe at least three non-functional requirements for your PCS, based on the process description.

Task #3: Digitalization (10%)

Digitalization is about storing, sharing and reusing information in an PCS and/or between different PCSs. The digitalization requirements for a PCS are to store information in such a format that several applications can read and write this information, regardless of storing locations, storage formats and data formats.

a) (3%) CSV, XML and JSON based text files are good candidates for storing data for digitalization. Compare these data formats and argue for one of the formats that you will use for your system.
b) (2%) At 9:00:00 today the reading from five instruments (1 to 5) are 1 = 4.36, 2 = 18.666, 3 = -42.4, 4 = 1015.3 and 5 = -0.7. These values should be added as a sample to a CSV (Comma Separated Values) file, indicate how these values should be added to an existing CSV file. Display the last contents of the CSV file after this sample has been added.
c) (5%) A SQL database is normally used for storing information, and some additional software modules are needed to fulfil the sharing and reusing requirements. One solution can be to copy the information from the SQL database into the file format that was selected in section 3.a for digitalization. Use pseudo coding to describe a software function that retrieve all the values for inlet sensor values of PT02011, TT02001, FT02005, PT02010, TT02002 and FT02006 for the last 2 hours, assuming that your system has sampled these values for every 120 seconds and storing in the suggested file format.

Task #4: Communication (5%)

The Open System Interconnection Reference (OSI) model is a layered model, shown in Figure 2, and one way to get an overview of services and functions for a communication protocol.

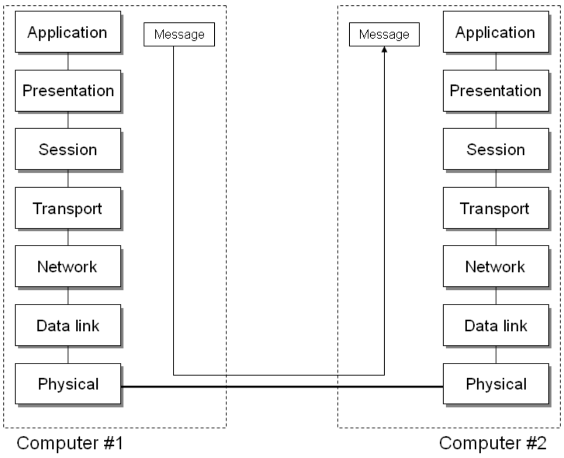

Figure 2: The seven layered OSI model.

a) (2%) The PCS is using several “smart” instruments meaning that these instruments support at least layer #1, the Physical layer, layer #2, the Datalink layer, and layer #7, the Application layer. Indicate the minimum functionality of these “smart” instruments.
b) (3%) Internet of Things (IoT) is based on the TCP/IP protocol, supported by the Physical, Data link, Network and Transport layers. Use the OSI model to indicate why many IoT devices are using a proprietary protocol for communication with the IoT device.

Task #5: Multitasking (15%)

Multitasking is structuring a software system into smaller, often manageable, software applications (also called processes, threads or tasks). A common way to do this is to group input and output signals that belongs to each other and make a separate application for each of these groups. Figure 1 shows the P&ID for a small process, and it is not recommended to control this system based on a large common loop. Instead, it should be divided into smaller subsystems and making an application for each of the subsystems.

a) (3%) Indicate the sub systems in your control system, with a name (sub system name) and a list of TAG numbers. HINT: The process consists of a mixer, and level and temperature controllers.
b) (2%) An application in a multitasking system will have several “running” states. Make a state diagram for an application in a multitasking system and explain the transitions between these states, including any reason(s) for these transitions.
c) (5%) Select one of the sub systems (not the mixer) from a) and make the pseudo coding for the application. If no sub systems have been defined, use the controller for the tank temperature. Use a simple on/off controller, no need to use time on an advanced controller.

In the “Multitasking and real-time” assignment in this course you were running a small multitasking system with a set of threads, in two different states (code 0 and code1). This small multitasking system is using the standard Windows operating system scheduler and one single common resource.

d) (5%) Include the propriate figures from your report (with reference) to summarize the total running times for all threads for both running code 0 and 1. Explain the difference in running times for code 0 and code 1, and indicate the real-time requirement for your system.

Task #6: Alarm System (5%)

An alarm system is an important part of any ICS, and several alarm standards exist.

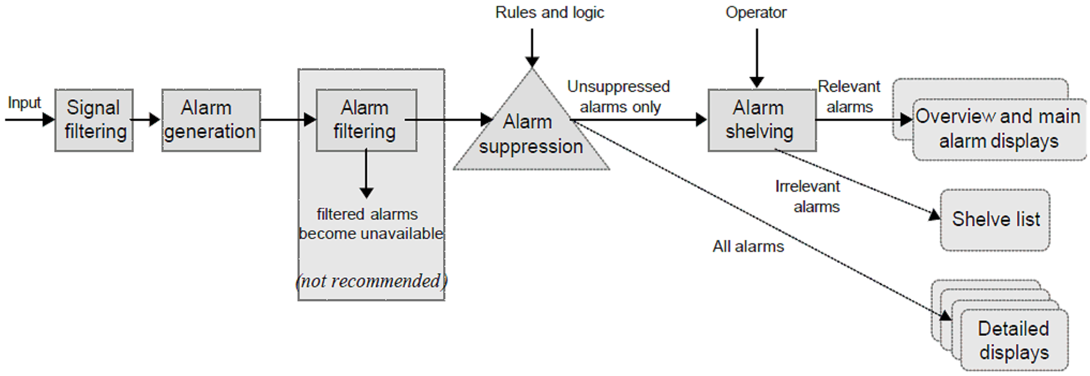

Figure 3: The YA-711 alarm standard

An alarm standard used a lot is the YA-711 standard, as shown in Figure 3

a) (2%) An alarm signal consists of three states. Make a state diagram showing these states and include a description of the transitions (state changes) between these states.
b) (4%) Indicate the contents of an alarm philosophy document, including important information from this specific process and HMI. Let the focus be on your PCS.
c) (4%) Use pseudo coding to describe a low-temp alarm method in the “Alarm Generation” module in the YA-711 alarm standard. A configuration module is handling the setting or changing of the alarm limits that should be used to get the current alarm limit.

Part #2 – Laboratory Work (45%)

The following system shall be created (Figure 4):

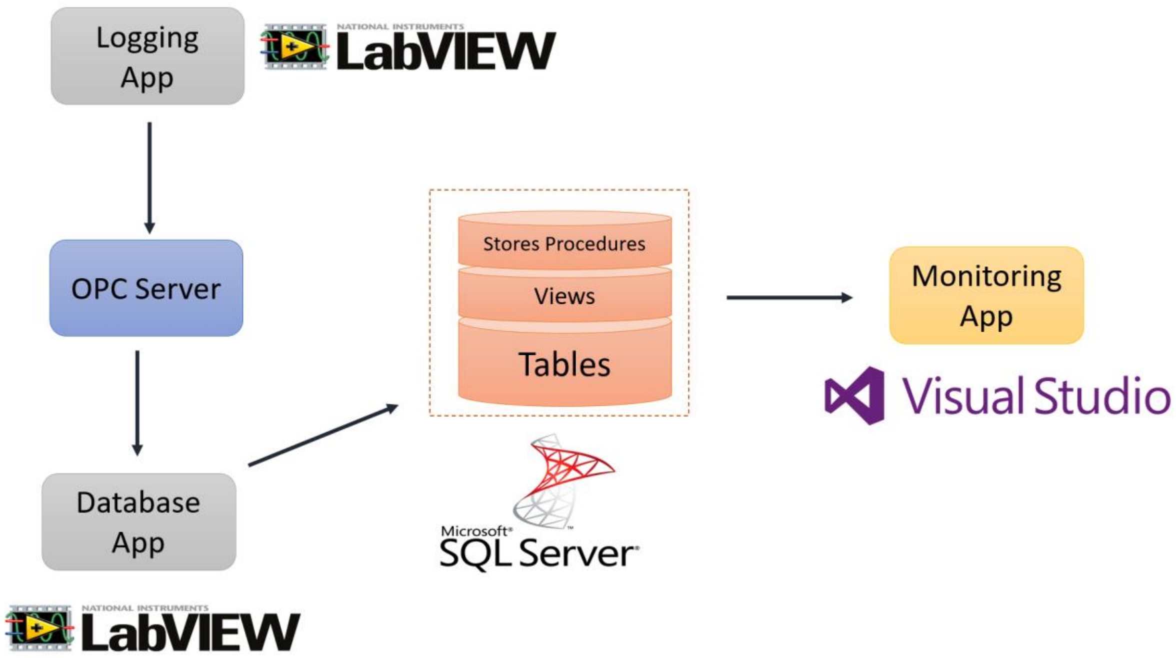

Figure 4: System Overview

The different "Tasks" and "Applications" can be developed and tested independently of each other, so if you for some reason are not able to do a specific "Tasks" or "Application", it should not affect the others. See detailed descriptions for each of those below.

Create the code and GUI according to the knowledge given in the laboratory assignments and recommended guidelines.

Screenshots of the different applications (both GUI and Code) and the results should be included with descriptions, comments, and discussions. No separate source code should be uploaded. Only one PDF file for the entire exam.

Task #7: Logging Application (15%)

Create an application in **LabVIEW** where you send a simulated pressure value to an **OPC DA Server**. It should be possible to enter the OPC tag used as part of the GUI. The process data should also be plotted.

The recommended OPC DA server is Matrikon OPC Simulation Server (but you can use another one if you prefer). The GUI should be well structured and intuitive to use and according to recommended guidelines.

The following SubVIs should be created and used in the application:

- You should simulate a DAQ device by generating a random voltage signal between 1V and 5V. This should be done by creating a SubVI, then use the SubVI inside your program.
- The signal from the simulated DAQ device should then be scaled to a pressure value between 5bar and 15bar. This should be done by creating a SubVI, then use the SubVI inside your program.
- To smooth the random simulated data, you should also create and implement a lowpass filter from scratch. This should be done by creating a SubVI, then use the SubVI inside your program.

Task #8: SQL Server (10%)

Create a **database** in **SQL Server** for storing the data required.

Assume we have different types of sensors. We can have different types of pressure sensors. We can also have different types of sensors for measuring different quantities like temperature, pressure, level, etc.

It should be possible to store the sensor data with different units, e.g., Pascal for a specific pressure sensor, degrees Celsius for a specific temperature sensor, etc.

Add relevant tables, columns, primary keys, etc.

Task #9: Stored Procedure (5%)

Create a **Stored Procedure** that saves a sensor data to the proper table(s) in SQL Server. Make sure to test the Stored Procedure properly in SQL Server.

Task #10: Database Application (5%)

Create another application in **LabVIEW** where you read the sensor data from the **OPC DA server**.

It should be possible to enter the OPC Tag and the name of the sensor from the GUI. The GUI should be well structured and intuitive to use.

The values from the OPC Server should be stored in **SQL Server**.

Convert the sensor data from bar to psi before you store the value in the database. The unit conversion should be done by creating a **SubVI**, then use the SubVI inside your program. Both the process value in bar and psi should be available in the GUI.

Make sure to use the **Stored Procedure** made in the previous task.

Task #11: Monitoring Application (10%)

Create an application in **Visual Studio/C#** (it can be either WinForm or ASP.NET, you decide) where you retrieve logged sensor data from the database and presents the data in the GUI. The GUI should be well structured and intuitive to use according to recommended guidelines.