Sequential Control

HANS-PETTER HALVORSEN, 2011.02.01

A State Diagram consists of:
- Steps
- Transitions
- Actions

- Engine Running? light is set to FALSE
- Start Engine button is Enabled
- Stop Engine button is Disabled

- Engine Running? light is set to TRUE
- Start Engine button is Disabled
- Stop Engine button is Enabled

- Engine Running? light is set to FALSE
- Start Engine button is Enabled
- Stop Engine button is Disabled

- Engine Running? light is set to FALSE
- Start Engine button is Enabled
- Stop Engine button is Disabled
Preface

This Lab Work has a web site with additional resources, etc.:

http://home.hit.no/~hansha/?lab=sequential

This lab work introduces the concepts of sequential control and introduces examples where sequential control is needed. Sequential control is used in applications where the tasks must be accomplished in a certain sequence. Controlling a batch process could be an example where you could use sequential control techniques.

The aim of the tasks in this lab work is to develop a better understanding of sequential control, and using state diagrams.

There are different techniques in use when it comes to design and implementation, such as

- Sequential Function Charts (SFC)
- State Diagrams

Both these techniques are very similar. They have states (in SFC the term steps is used) with actions to be executed when the state is active. They have transitions from one active state to another taking place if the transition condition is logically true.

Sequential function chart (SFC) is a graphical programming language often used for PLCs (PLC - Programmable Logic Controller).

You may implement sequential control in different programming language, but in this lab work we will use LabVIEW to create a State Diagrams or a State Machine.
# Table of Contents

**Preface**......................................................................................................................................................ii

Table of Contents ........................................................................................................................................ iii

1  Sequential Control; Theory and Examples.......................................................................................... 4
   1.1  Background Theory ................................................................................................................ 5
   1.2  Video ........................................................................................................................................ 5

2  Sequential Control in LabVIEW ......................................................................................................... 7
   2.1  Introduction ............................................................................................................................ 7
   2.2  Step-by-Step Example ............................................................................................................ 7
   2.3  You are on your own .............................................................................................................. 14

3  Project................................................................................................................................................ 16
1 Sequential Control; Theory and Examples

In this chapter we will introduce the basics about Sequential Control, Sequential function charts (SFC) and State diagrams. You should preferably go through this chapter before you start the lab work.

State diagrams have states with actions to be executed when the state is active. They have transitions from one active state to another taking place if the transition condition is logically true.

Below we see a typical state diagram:

A state diagram represents a state machine. States of a state machine are either active or passive. Only one state is active at a time. The state machine always starts in a particular state defined as the initial state, and it ends (stops) after the final state. Special symbols or colors may be used for the initial state and the final state. See the Figure above.

Associated with a state are one or more actions. An action may be a control action executed by a controller. Actions may be listed as indicated below:

- State 1 actions:
  - Action 1: Valve V1 is open;
  - Action 2: Motor M1 runs;
- State 2 actions:
5  Sequential Control; Theory and Examples

- Action 1: Valve V1 is closed;
- Action 2: Heater H1 is on;

The actions for a given state can be listed in a box attached to the state symbol, see Figure below. However, the diagram may then become overloaded by symbols. Therefore, in more comprehensive state diagrams, it may be better to just list the actions in a separate document.

Transitions bring the state machine from one state to another. A transition can go only from an active state which is presently active. A transition takes place only if its transition condition is TRUE. Transition conditions are in the form of logical expressions having value either TRUE or FALSE.

1.1 Background Theory

Before you start this lab work, you should read the article “Sequential Control” by Finn Haugen. The article is available from the homepage of this lab work.

Test yourself: Questions from the article:

- Name the different parts in a State diagram.
- List the differences between Sequential function charts (SFC) and State diagrams in the report.
- What type is used in LabVIEW? SFC or State Diagrams?

1.2 Video

Before you start the Lab Work, you should watch the video “Sequential Control” by Finn Haugen.
You need **Headphones**! The video lasts about 20 minutes.

Download and run the example when you are watching the video.

The video and the examples used in the video are available from the homepage of this lab.

**Test yourself:** Questions from the video:

- Where is sequential control needed?
- What kind of process is controlled in the example in the video?
- List the different states/steps involved in the example
- What are the 3 main components in a SFC/State diagram?
2 Sequential Control in LabVIEW

2.1 Introduction

This chapter explains the basic concepts of creating state-based applications in LabVIEW.

What is Sequential Control? The aim of this example is to develop a better understanding of sequential control and using state diagrams and State Machines in LabVIEW.

In this example you will learn:

- The basic concepts of Sequential Control
- The basic concept of a state machine and how to create a state machine in LabVIEW.
  In LabVIEW you use a State Machine to create a Sequential Control application

There are different techniques in use when it comes to design and implementation of Sequential Control, such as:

- Sequential Function Charts (SFC)
- State Diagrams

Both these techniques are very similar. They have states (in SFC the term steps is used) with actions to be executed when the state is active. They have transitions from one active state to another taking place if the transition condition is logically true.

LabVIEW uses “State Diagrams”.

Here you will learn how to create a State Diagram application in LabVIEW using the State machine principle.

Download the Tutorial “Introduction to State-based Applications in LabVIEW” from the homepage of this lab.

2.2 Step-by-Step Example

Task 1: Step-by-Step Example

In this example we will implement a simple Sequential Control (Start/Stop an Engine) in LabVIEW using the State Machine principle in LabVIEW. The State Machine approach in
LabVIEW uses a **Case structure** inside a **While loop** to handle the different states in the program, and the transitions between them. A **Shift Register** is used to save data from and between the different states. Below we see the State Diagram for the Example:

A State Diagram consists of:
- **Steps**
- **Transitions**
- **Actions**

The User Interface (Front Panel) could look something like this:

When the user push the “Start Engine” button, the light gets on and when the user push the “Stop Engine” button, the light gets off. When the user pushes the “Exit” button, the program will stop.
The Procedure is as follows:

**Step 1:** Create a **New VI** (File→New VI)  (Blank VI)

**Step 2:** Create your **Front Panel/User Interface** as shown above. You need 3 buttons: “Start Engine”, “Stop Engine” and “Exit”. In addition you need a light/LED, call it “Engine Running?”.

**Step 3:** Add a **While Loop** to your VI

Your application should now look like:

![Image of LabVIEW workspace with a While Loop and Case Structure]

**Step 4:** Add a **Case Structure** to your VI **inside** the While Loop

Your application should now look like:
**Step 5:** Add a Shift Register to your While Loop by right-click on the While Loop and select “Add Shift Register”.

Your application should now look like:
Step 6: Create the Init State according to the State Diagram. Your application should now look like:

Step 7: Add a Case called, e.g., “Wait” in your Case Structure. Create an Event Structure inside this case.
Add Events (right-click on the Event Structure) for your 3 different buttons; “Start Engine”, “Stop Engine” and “Exit”. Your application should now look like:

**Step 8:** Add the Start Engine, Stop Engine and Exit cases (states) and create the Actions according to the State Diagram.

Your application should now look like:

**Start Engine:**
Stop Engine:

Exit:
**Step 8:** Run the Application and check if it works as expected.

Turn on “Highlight Execution” and see how the flow in your application is.

Does it work as expected?

[End of Task]

**2.3 You are on your own**

This is a similar task, but no step-by-step instructions are given.

**Task 2: Christmas tree**

Create a simple State Diagram/SFC (using pen and paper) for the lighting of a Christmas tree switching between 3 different colors (green, red, blue) in a continuous loop. Each light should be on for 2 seconds.

→ Document the diagram in the report. Use a Tool like Microsoft Visio or Microsoft PowerPoint, etc.
Create a simple State machine for the Christmas tree in LabVIEW. Use the While loop, the Case structure and the Shift Register functionality in LabVIEW in order to complete the task.

In the Front Panel in LabVIEW you should display the 3 different lights/LED (green, red and blue). You should also display the state diagram and illustrate in which state the program is at any time.

The application could look something like this:
3 Project

This is the main Task in this Lab Work. You will implement a system of your choice in LabVIEW using the State Diagrams/State Machine techniques.

**Task 3: Project**

Use **LabVIEW**, implement a state diagram for a system you select or define by yourself. Think about some physical system that requires sequential control. On the front panel of the program it must be indicated at any instant of time which of the states that is the active state. You should add a picture of the state-machine on the front panel of the LabVIEW program with a lamp showing the active state.

Examples:

- Chemical batch reactor/Tank mixing
- Assembly line
- Washing machine
- Car wash
- Mathematical simulation
- Traffic intersection
- Cake baking
- Traffic Light
- Drilling operation
- Coffee machine
- Smoothie mixer
- etc.

→ Use your imagination!

The following Tasks should be done:

1. **Define** the system to be created.
2. **Draw** the Sequential Function Chart (SFC)/State Diagram on paper (should be included in the report). You could, e.g., use Microsoft Visio to create the diagram. The actions in every state must be defined, and the transition conditions in every transition must be defined.
3. **Implement** the system in LabVIEW using the State Machine principle. On the Front panel you should have a diagram of the process/system and corresponding State Diagram where you indicate which state the process is in at any time. Tip: Import the picture you created in Visio and add different indicators (e.g., Boolean and numeric indicators) to illustrate the different states in the diagram.
Below we see a **State Machine** Example in LabVIEW:

[End of Task]
Telemark University College
Faculty of Technology
Kjølnes Ring 56
N-3914 Porsgrunn, Norway

www.hit.no

Hans-Petter Halvorsen, M.Sc.
Telemark University College
Department of Electrical Engineering, Information Technology and Cybernetics

Phone: +47 3557 5158
E-mail: hans.p.halvorsen@hit.no
Blog: http://home.hit.no/~hansha/
Room: B-237a