Research document
Basic TIA Portal

By Team Mini-FLUFFY

Date: 16-5-2024

Introduction

There are a bunch of different ways to program a PLC. For this project, we have a Siemens PLC, which means we have to use the TIA portal environment. This environment is made for programming on Siemens PLC. Although it is similair in some ways to the environment we're used to, it does have a few differences and things that need to be looked in to. In this document, these discrepancies will be analysed and explained how they work in TIA portal.

Differences Function Block and Function

Like in CODESYS, in TIA Portal, you have functions (FC) and function blocks (FB). They sound similair in name, but there is a difference in how they work. The biggest difference between function block and function is how it handles memory. Below are the differences for how TIA portal handles memory for these two.

- Function blocks
  - With different instances, different information is stored per instance
  - Saves instance data from previous call
  - Can use instructions that require their own data, like timers
- Functions
  - With different instances, the same area of memory is used, so it’s shared
  - Does not save instance data from previous call
  - Can not use instructions that require their own data, has to call them from outside its own scope.

(Bee, Siemens TIA Portal - Function Block and Function Differences, 2022)

Another difference between function blocks and functions are that with function block, an input or output field can be left empty. With functions, however, they cannot be empty. Functions must have all their inputs and outputs filled in. For a function block, i fit does not have one of these fields filled in, it will use its own instance data for this. Since functions dont save data like function blocks, it cannot do this.

Differences Instances

TIA Portal has several different instances, like single instance, multi-instance and parameter instance. These instances come in to play when you create a block, and selecting a certain one causes different behaviours.

- Single instance: the function block has its own data and only one instance has that data. If you make a second instance, it will have its own seperate data.
- Multi-instance: the function block has shared data with all other calls to a multi-instance block. The memory is stored in the static area.
- Parameter instance: the general function block is the same, but the instance will be a parameter for the block. This way, you can dynamically use different data for the block. This data is stored in the InOut memory section.

(Edema, sd), (CaOd, 2019)

As you can see, the main difference is how memory is handled with all of these. The place in memory determines how exactly calling it again will work out. Single instance is its own block with its own data, multi-instance is stored in static memory and parameter instance in the input-output category.

Difference Tags

In TIA Portal, you have tags and variables. The tags are different from variables, and tags itself aren’t all of the same type. There are three main types of tags, and these have types like boolean or integer assigned to them. The three main types are the following:

- Input: used for a sensor, a physical input. Written down like %I
- Output: used for an actuator, a physical output. Written down like %Q
- Memory: a piece of memory on the PLC. Not directly related to any I/O. Written down like %M

(Bee, Difference Between Tags & Variables In Siemens TIA Portal, 2022)

This is the difference between tags. Keep in mind that these kinds of tags also need a type, like boolean. Following after %I, %Q or %M there address needs to be noted. This is just a number of what address the PLC uses. Make sure you keep track of the size of the type, so you don’t accidently end up in memory that some other tag uses.

Absolute vs symbolic addressing

If you want to connect your PLC program to a HMI with WinCC, you will need to use some of your tags in the HMI. When trying to use tags, you can choose between absolute access or symbolic access. But what does this mean? Well, the difference between the two is that absolute access works directly with the address, for example like %M100.0, and symbolic uses the name of the tag for it, like “SYSTEM_OK”. (Dongerpure, 2017)

Sources

Bee, L. (2022, September 15). *Difference Between Tags & Variables In Siemens TIA Portal*. Opgehaald van Liam Bee: https://liambee.me/siemens/difference-between-tags-variables-in-siemens-tia-portal/

Bee, L. (2022, October 17). *Siemens TIA Portal - Function Block and Function Differences*. Opgehaald van Liam Bee: https://liambee.me/siemens/siemens-tia-portal-function-block-and-function-differences/

CaOd. (2019, August 16). *Differences between DB Single, DB Multi & DB Parameter Instance?* Opgehaald van Industry Support Siemens: https://support.industry.siemens.com/forum/nl/en/posts/differences-between-db-single-db-multi-db-parameter-instance/218488

Dongerpure, S. (2017, January 31). *What is the difference between symbolic and Absolute addressing in siemens*. Opgehaald van YouTube: https://www.youtube.com/watch?v=5CMiAo0PsNc

Edema, V. (sd). *Understanding Function Block Instances (Single, Multi and Parameter) in Siemens TIA Portal*. Opgehaald van SolisPLC: https://www.solisplc.com/tutorials/function-block-instances-in-siemens-tia-portal#multi-instance