Code Compile

presents

PLC lessons on:

SIEMENS S7-1500

PLC for larger machine control applications

www.codeandcompile.com

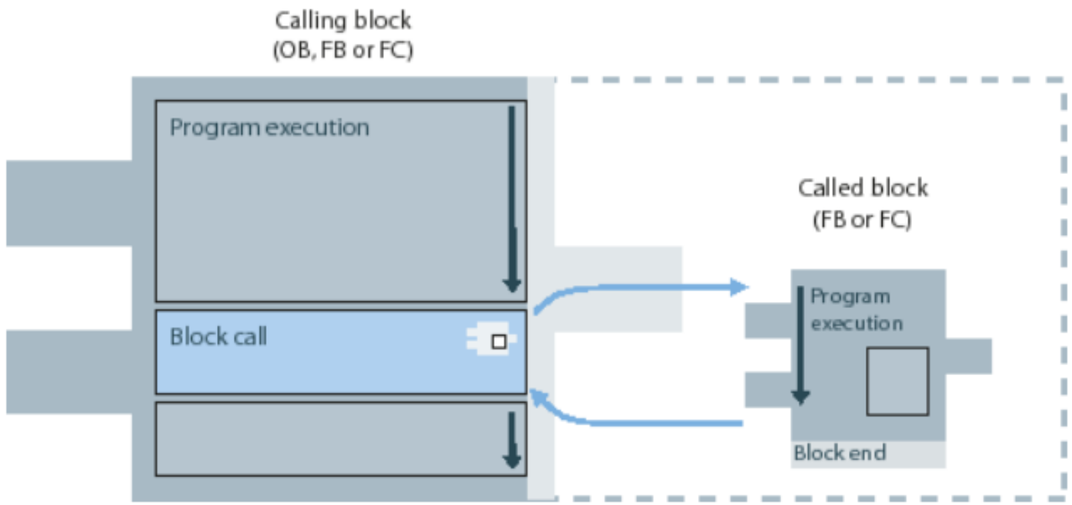
Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Basics of Block calls

For your blocks **to be executed** in the user program, they **need to be called from another block.**



When one block **calls another block,** the **instructions of the called block are executed**. **Only when execution of the called block has been completed does execution of the calling block resume**. The execution is continued with the instruction that follows on the block call.

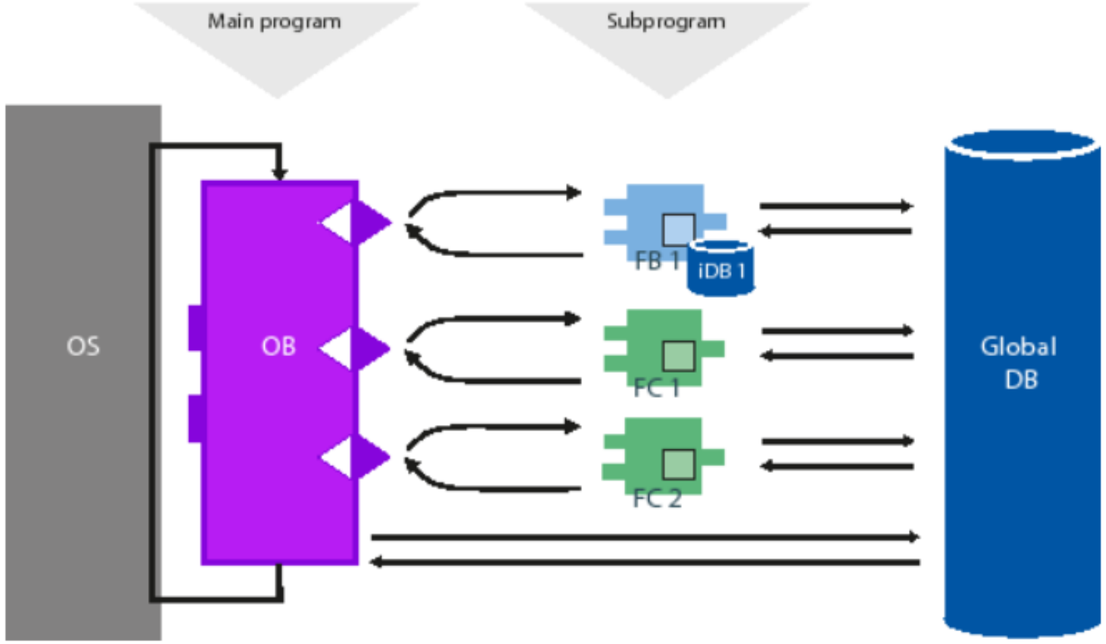
Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Call hierarchy

The following figure shows an example of the order and nesting of block calls within an execution cycle:



Code Compile
learning made easy

SIEMENS

PLC
S7-1500

What is a Function?

A function is a logic block "**without memory**." Temporary variables belonging to the FC are saved in the local data stack.

This data is then lost when the FC has been executed. **To save data permanently, functions can also use shared data blocks.**

You **cannot assign initial values** for the local data of an FC.

Applications

To return a function value to the calling block (example: arithmetic operations)

Code Compile
learning made easy

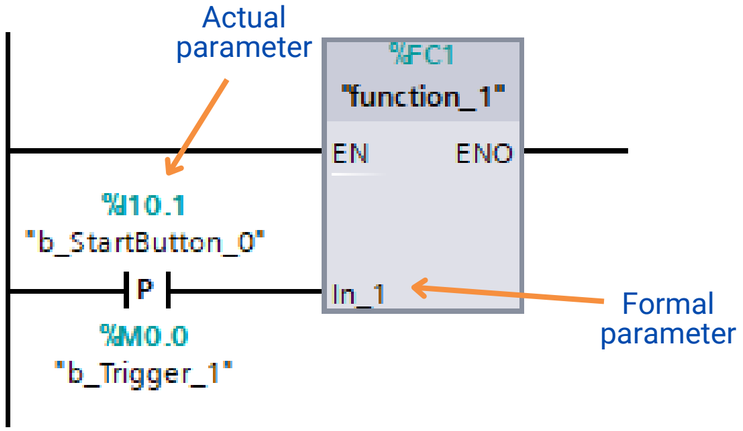
SIEMENS

PLC
S7-1500

Actual and Formal Parameters

A **formal parameter is a dummy** for the "**actual**" parameter. Actual parameters replace the formal parameters when the function is called.

You must always assign actual parameters to the formal parameters of an FC



Caution

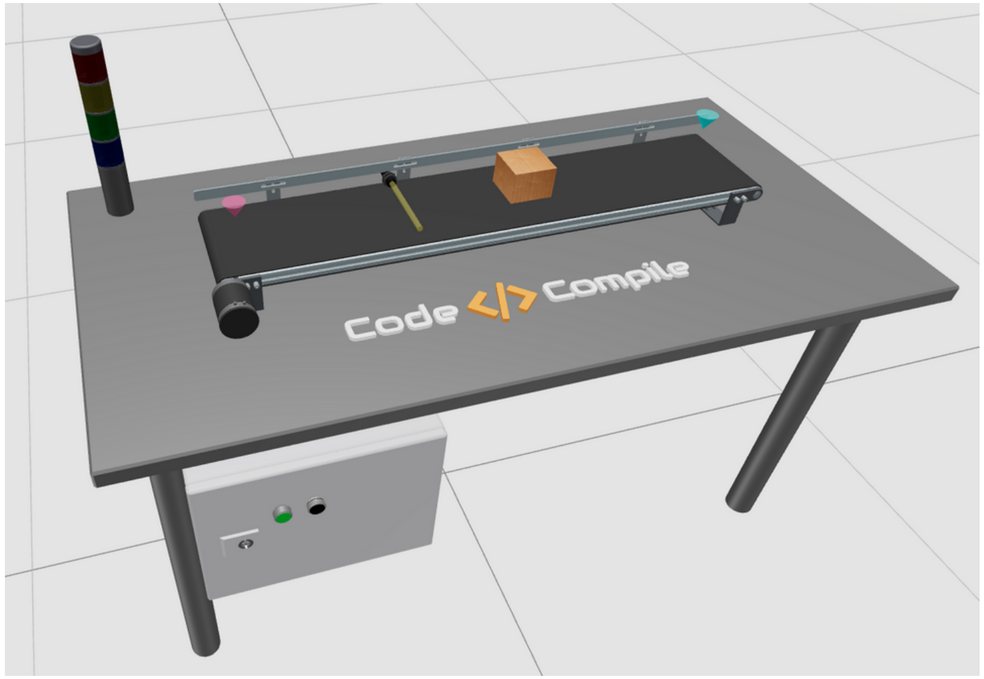
In this case, if no data are written to an OUTPUT parameter in an FC, the block may output random values!

Code Compile
learning made easy

SIEMENS

Examples:

Write a function to count the number of boxes passing on the conveyor



Code Compile
learning made easy

SIEMENS

Example: Count function with DB

PLC
S7-1500

Shared

Instance

Is 

+ 

= 

?

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

What is a Function block?

A function block is a block "**with memory.**" It is **assigned a data block** as its memory (**instance data block**). The parameters transferred to the FB and **the static variables are saved in the instance DB.** Temporary variables are held in the local data stack.

Data saved in the instance DB are preserved when the execution of the FB is complete. However, **data saved in the local data stack** are **lost** when the execution of the FB is completed.

Applications

Function blocks make it much easier to program frequently occurring, complex functions.

www.codeandcompile.com

Code Compile
learning made easy

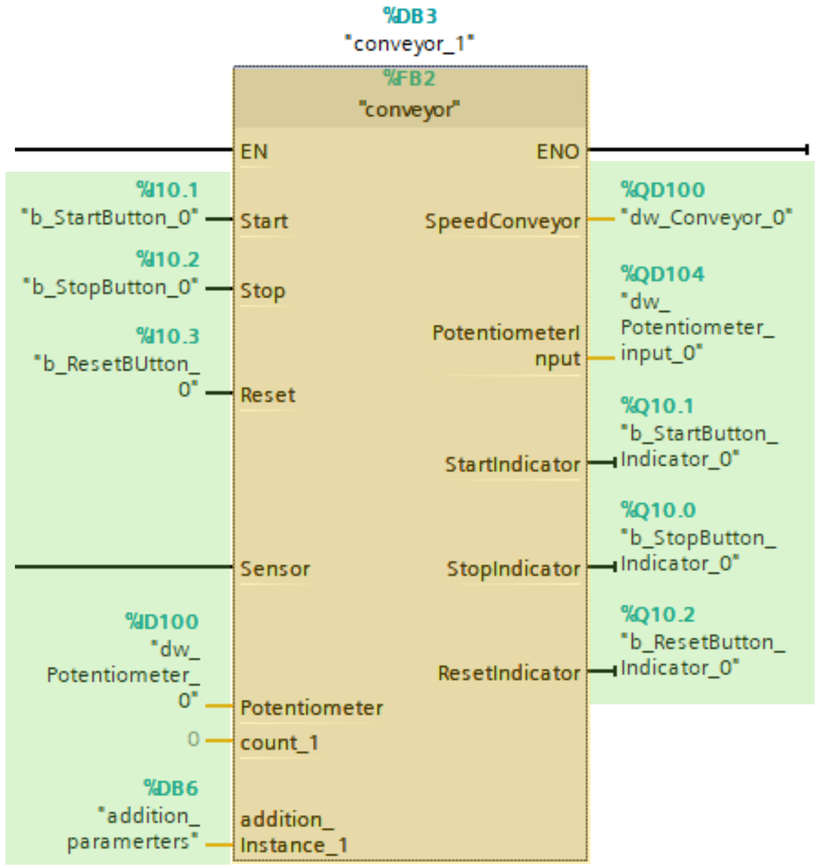
SIEMENS

Assigning Actual Parameters to the Formal Parameters

PLC
S7-1500

Actual parameters

Formal parameters



It is not generally necessary in STEP 7 to assign actual parameters to the formal parameters of an FB.

Exceptions: Actual parameters must be assigned in the following situations:

For an in/out parameter of a complex data type (for example, STRING, ARRAY or DATE_AND_TIME)

www.codeandcompile.com

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Assigning Actual Parameters to the Formal Parameters

STEP 7 assigns the actual parameters to the formal parameters of an FB as follows:

- When you specify actual parameters in the call statement: the instructions of the FB use the **actual parameters provided**
- When **you do not specify actual parameters** in the call statement: the instructions of the FB use the **value saved in the instance DB**.

Code Compile
learning made easy

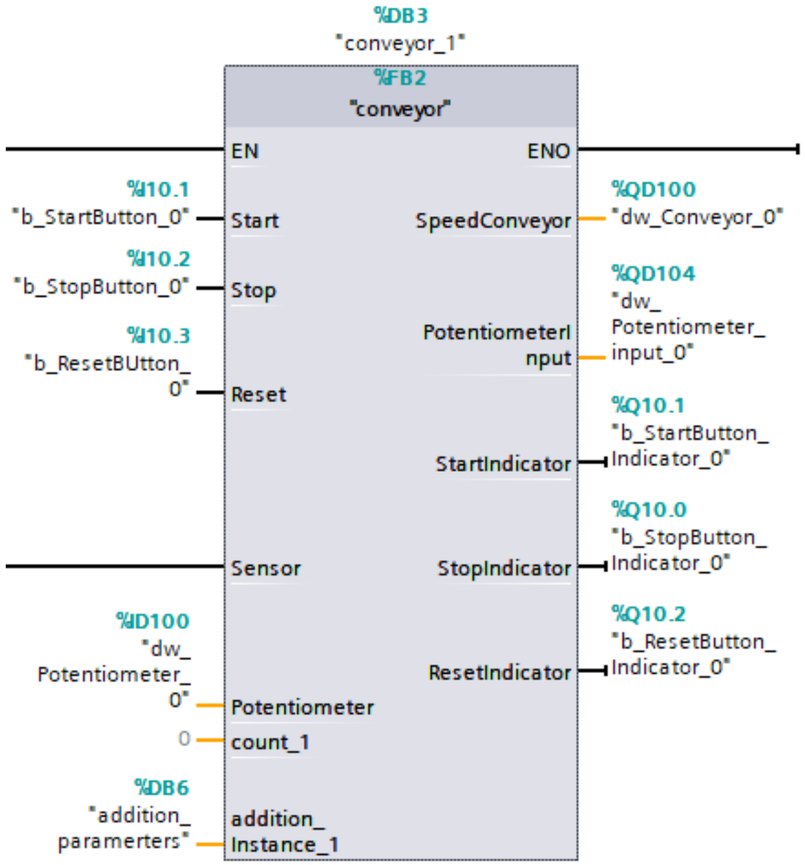
SIEMENS

PLC
S7-1500

Calling Function Block

By **calling more than one instance of an FB**, you can control more than one device with one FB.

An FB for a conveyor type, can, for example, control various conveyors **by using a different set of instance data for each different conveyor.** The data for each conveyor can be saved in one or more instance DBs.



www.codeandcompile.com

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Instance Data Blocks

An instance data block is **assigned to every function block call that transfers parameters**. The **actual parameters and the static data of the FB are saved in the instance DB**.

Instance means a function block call. If, for example, **a function block is called five times in the S7 user program, there are five instances of this block.**

Creating an Instance DB

Before you create an instance data block, **the corresponding FB must already exist**. You specify the number of the FB when you create the instance data block.

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Basics of Instance

After a **function block is called,** it **needs memory for its working** data. This data is referred to as an **instance**.

Properties:

- Instances are always assigned to an FB.
- The structure of an instance is derived from the interface of the associated FB and can only be changed there.
- Instances are created automatically when a function block is called.

Types:

- Single instances
- Multi instances
- Parameter instances

Code Compile
learning made easy

SIEMENS

Single Instance

PLC
S7-1500

Single instance

The called function block **saves its data** in an instance data block of its own. The instance DB thus **contains the working data for an individual block call.**

Advantages:

- Reusability of the function blocks
- Good structuring options for simple programs

The instance DB contains the following data:

- **Block parameters**: The block parameters in the "Input", "Output" and "InOut" sections form the interface of the block for the call in the program.
- **Static local data**: It is used for permanently storing intermediate results beyond the current program cycle, for example for storing the signal state for an edge evaluation.

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

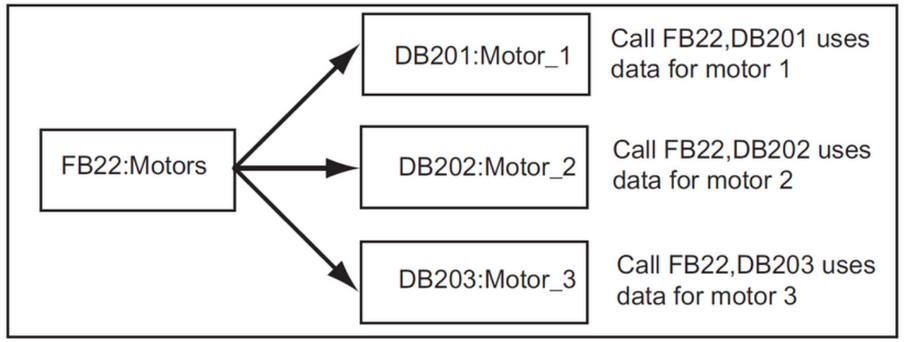
One Instance DB for Each Separate Instance

Single instance

If you call the function block as a single instance, the function block saves its data in its own instance data block.

If you assign several instance data blocks to a function block (FB) that controls a motor, you can use this FB to control different motors.

The data for each specific motor are saved in **different data blocks.** With this technique, **only one function block is necessary** for several motors.



Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Multi Instance

If you call the function block as a multi-instance, it **saves its data in the instance data block of the calling function block** and **not in its own instance data block**. This allows you to **get by with fewer instance data blocks in your program.**

Advantages:

- Good structuring options for complex blocks
- Lower number of instance DBs
- Easy programming of local subprograms, for example for local timers or edge evaluations.

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

One Instance DB for Several Instances of an FB (Multiple Instances)

You can also transfer the **instance data for several motors** at the same time **in one instance DB**. To do this, you must program the calls for the motor controllers in a further FB and **declare static variables with the data type FB for the individual instances in the declaration section of the calling FB.**

	Static		
	step	Non-retain	Bool
	count	Non-retain	DInt
	addition_Instance		"addition"
	subtraction_Instance		"subtraction"

If you use one instance DB for several instances of an FB, you save memory and optimize the use of data blocks.

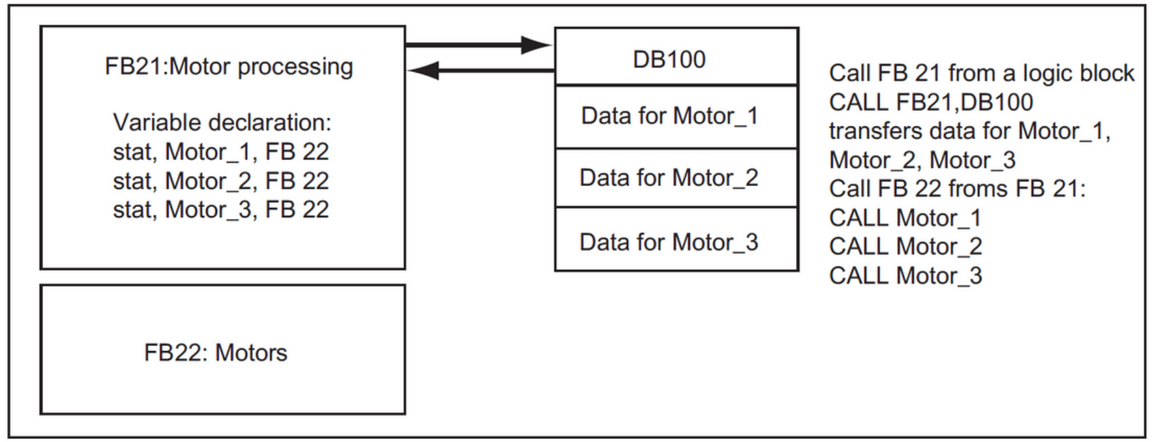
Code Compile
learning made easy

SIEMENS

PLC
S7-1500

One Instance DB for Several Instances of an FB (Multiple Instances)

In the following figure, the calling FB is FB21 "Motor processing," the variables are of data type FB22, and the instances are identified by Motor_1, Motor_2, and Motor_3.



In this example, FB22 does not need its own instance data block, since its instance data are saved in the instance data block of the calling FB.

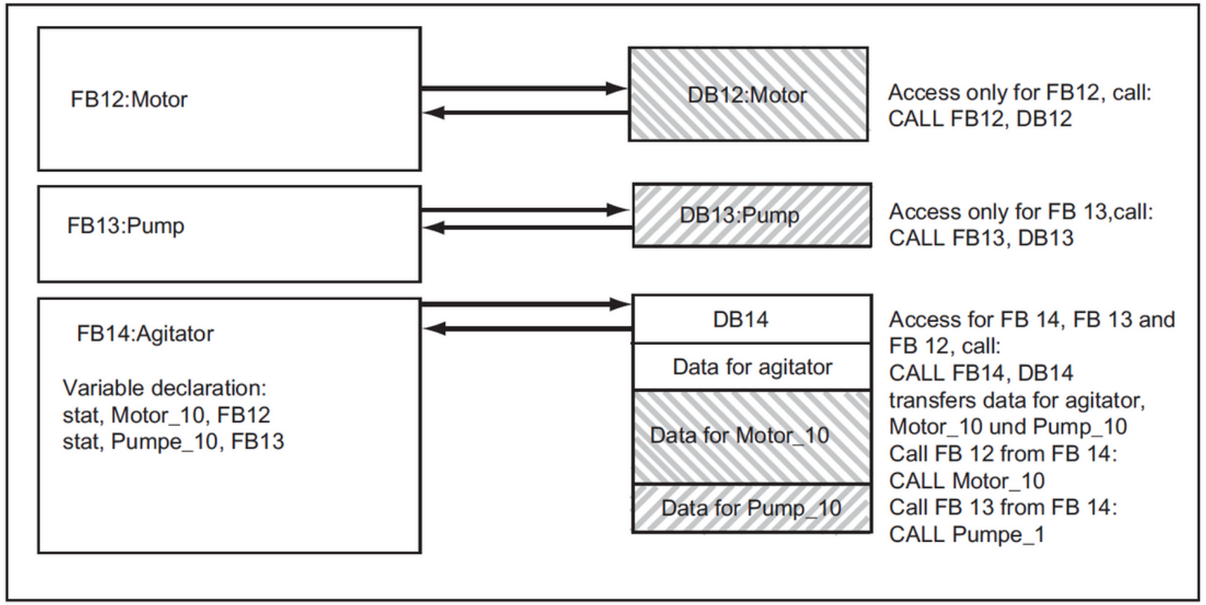
Code Compile
learning made easy

SIEMENS

PLC
S7-1500

One Instance DB for Several Instances of Different FBs (Multiple Instances)

In the example in this figure, the assigned instance data are stored in a common instance DB.



Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Parameter Instance

If you call the function block as a parameter instance, **the function block saves its data in the instance you specify as block parameter and not in the instance of the called block.** This gives you the option of defining the instance for this FB call during runtime.

Advantages:

- At runtime, you can define which instance is currently being used.
- You can process different instances iteratively in program loops.

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Shared Data Blocks

If a logic block (FC, FB, or OB) is called, it can occupy space in the local data area (L stack) temporarily. In addition to this local data area, a logic block can open a memory area in the form of a DB. In contrast to the data in the local data area, the data in a DB are not deleted when the DB is closed, in other words, after the corresponding logic block has been executed.

Each FB, FC, or OB can read the data from a shared DB or write data to a shared DB. This data remains in the DB after the DB is exited.

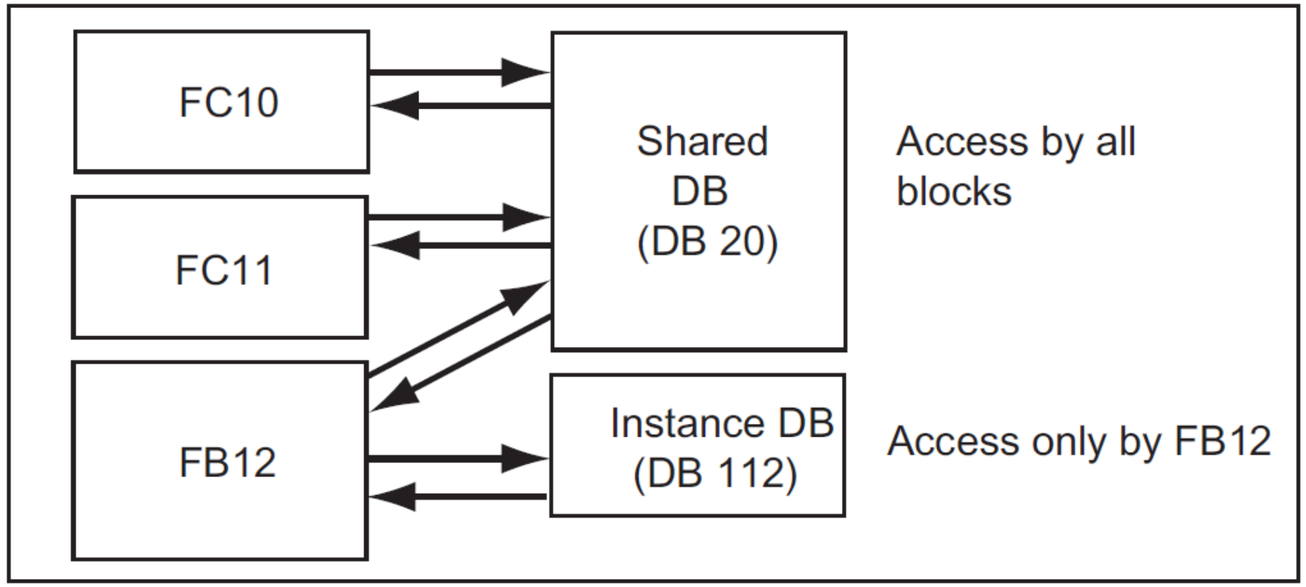
A shared DB and an instance DB can be opened at the same time. The following figure shows the different methods of access to data blocks.

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Shared Data Blocks



Code Compile
learning made easy

SIEMENS

Similarities between Functions and Function Blocks?

FB FC

PLC
S7-1500

- They are two types of subroutines that make the PLC code easy to read and troubleshoot with parameters for Inputs, Outputs and InOut
- They **divide** and **organize the user program** into small parts, which helps in the program's maintenances, optimization and alterations.
- The **algorithms are programmed just once** and **can be used many times in the project** with a different sets of variables
- Both FCs and FBs can **hold parameters (IN, OUT, IN-OUT and TEMP)**, which allows the re-use of the blocks with different calling environments.
- Both can call FB and FC or be called for FB and FC

Code Compile
learning made easy

SIEMENS

PLC
S7-1500

Differences between Functions and Function Blocks?

- While an **FC uses the address of the given parameter to read and write directly**, FB copies the parameters to/from an associated DB (so-called Instance DB) and works internally with the DB variables.
- **FB can use Static variables** whose value is stored in the instance DB where as **FC cannot as there is no instance DB**
- **FB needs an auxiliary DB for each call,** whereas **FC does not**
- FB can be called as **multiple instance** where as FC cannot
- FB can be called without **filling in all parameters** whereas FC cannot

Code Compile

presents

SIEMENS

S7-1500

Thank you!

www.codeandcompile.com