

How can I work on TesysT Custom Logic ?

I- Type of publication

☐	Typical application
☒	Best know Method (BKM)
☐	Troubleshooting guide

☐	Level 2 use
☐	Internal use
☒	Customer

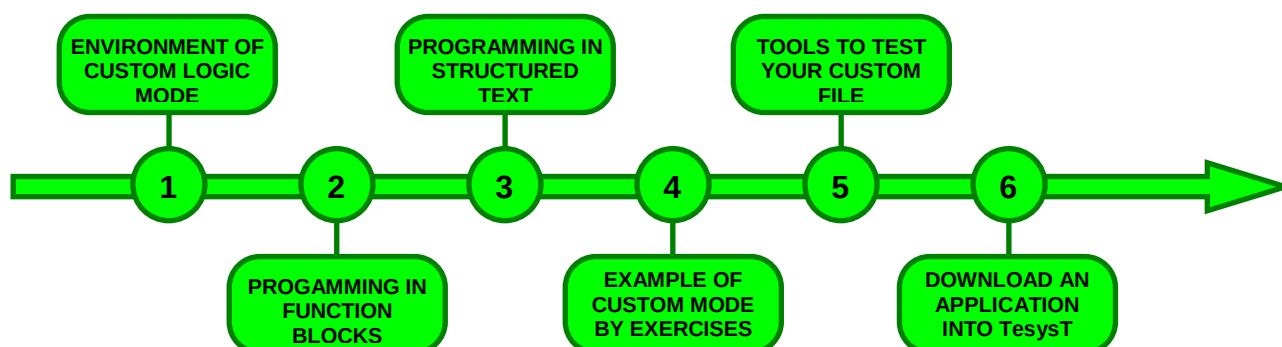
II- Product

- Product range :

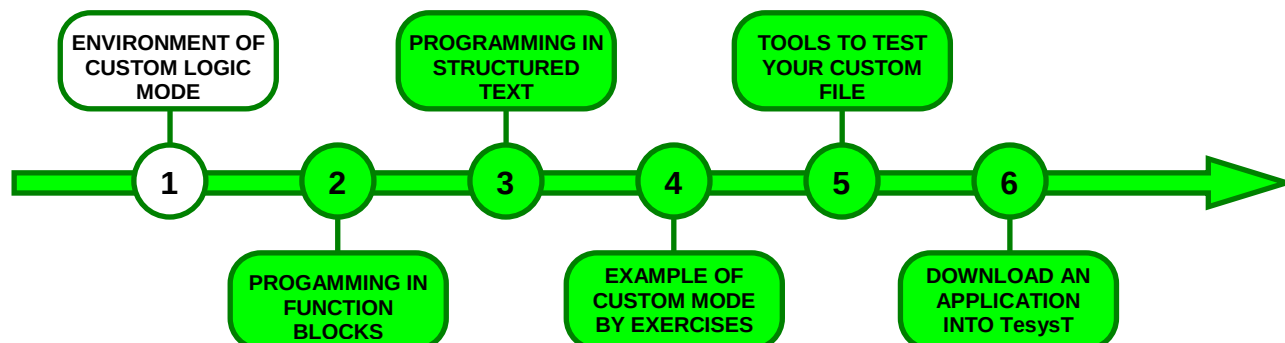
- Product family :

III- Introduction

This document describes you which is the right way to make a basic program in custom mode. We show you the main step to build a program and download it into TesysT.



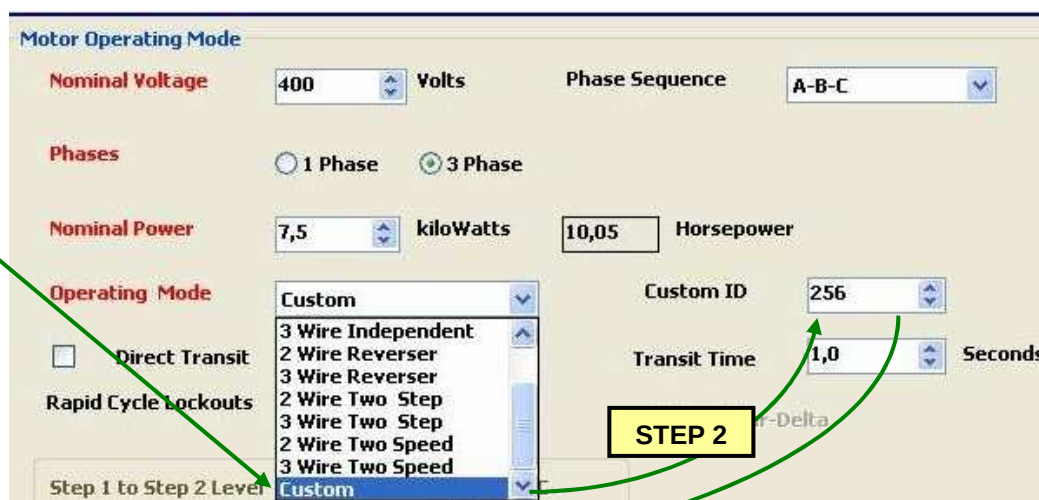
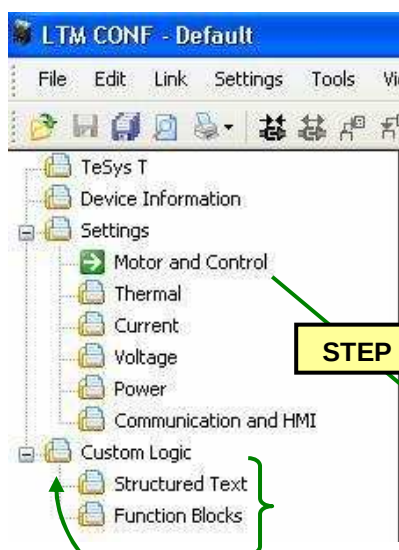
IV- Description



First of all, you should configure TesysT in Custom Mode with the Menu "Settings/Motor and Control". In this menu, you select Custom Mode and choose your "Custom ID". The Custom ID is your program number. This number should be absolutely between 256 and 511.

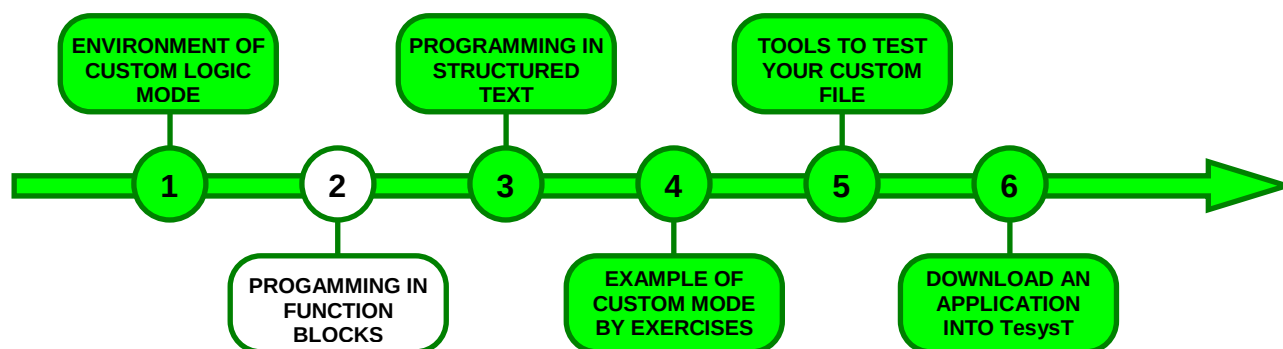
After, you access to Custom Logic Editor by the menu "Custom Logic" to create your program. You have two sub-parts:

- Structured Text
- Function Blocks



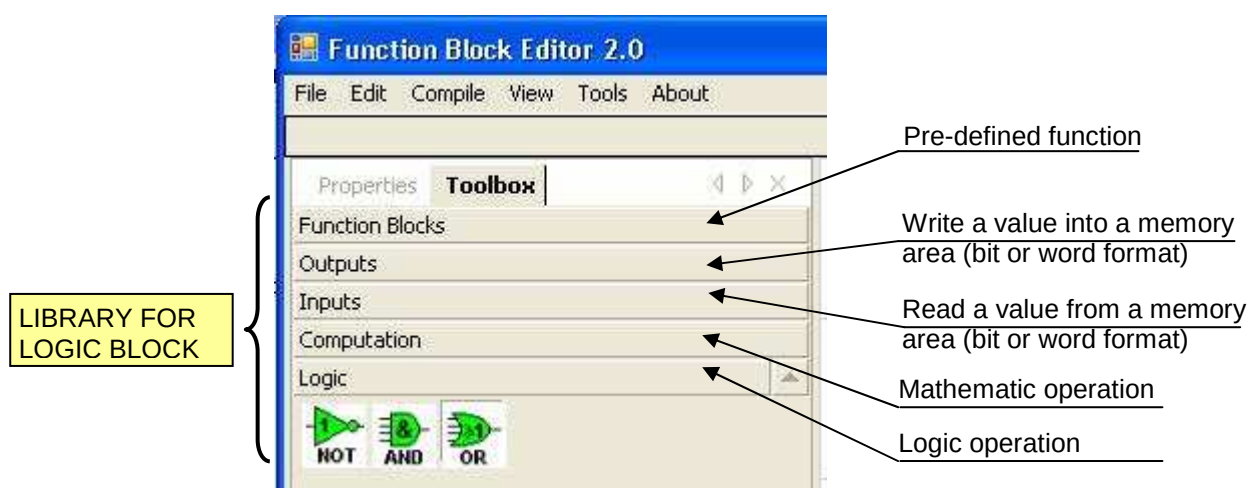
You can program in list format a complete application with Structured Text. In the directory of PowerSuite you already have Structured Text files (see below Part 3 of the Technical Resolution) for pre-defined operating mode like 2 wires Reverser, 3 wires Overload, 2 Wires 2 steps.....

The function blocks is used to program one part of an application or a basic program to make a logic function between inputs and outputs. You cannot make or have a pre-defined Operating mode in function blocks.

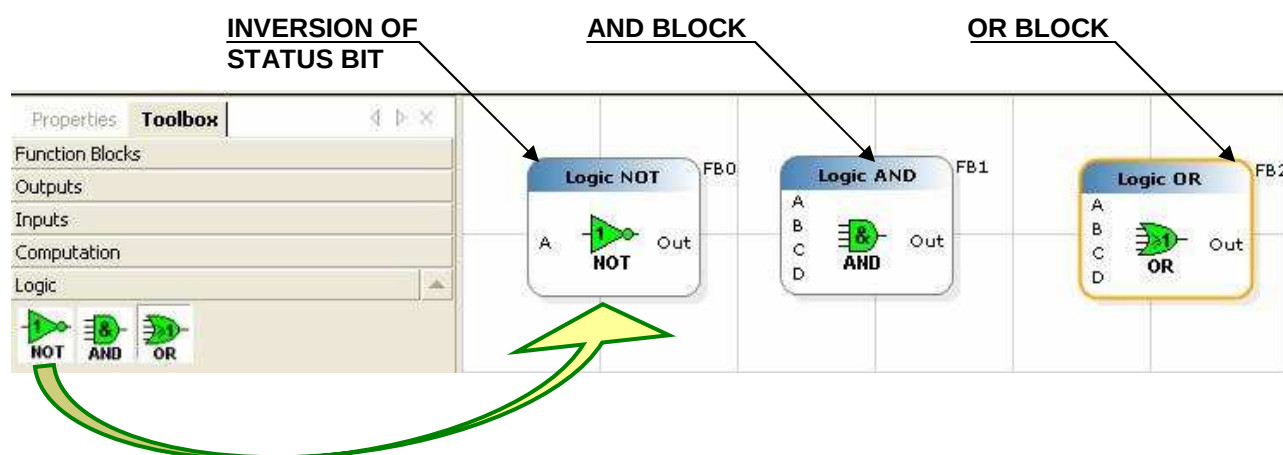


■ **ToolBox**

You can make a program with logic blocks :



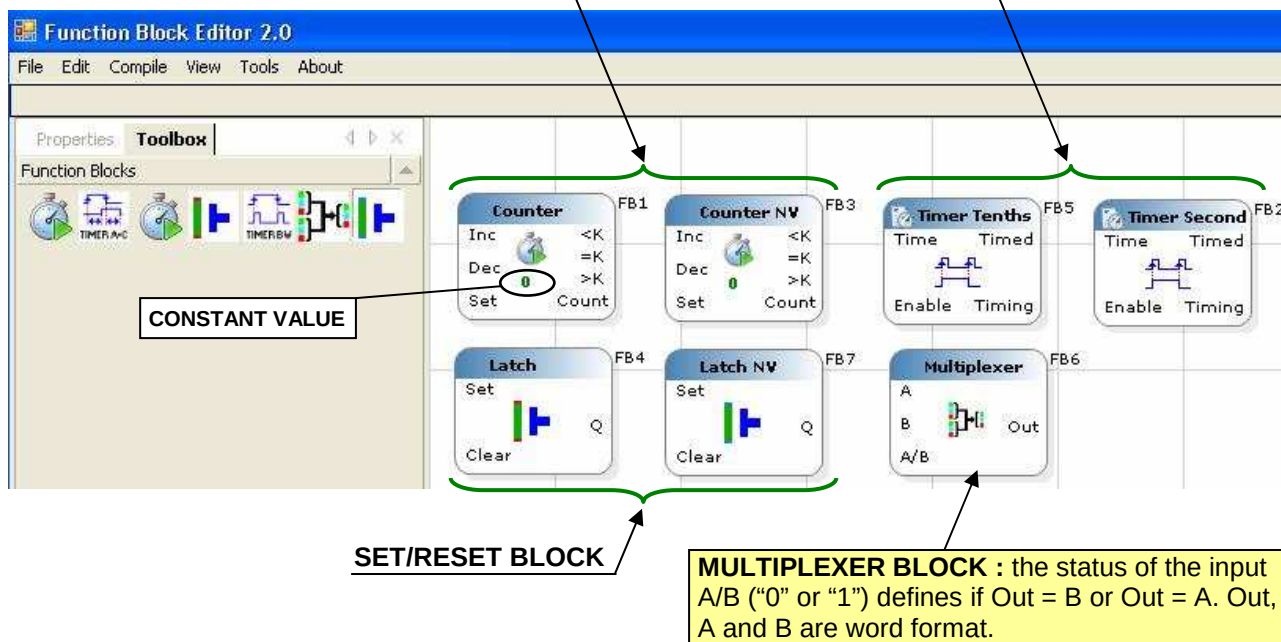
- **Logic blocks**



- **Function blocks**

COUNTER BLOCK : You increase or decrease a value. This last data is compared to "Constant value" entered.

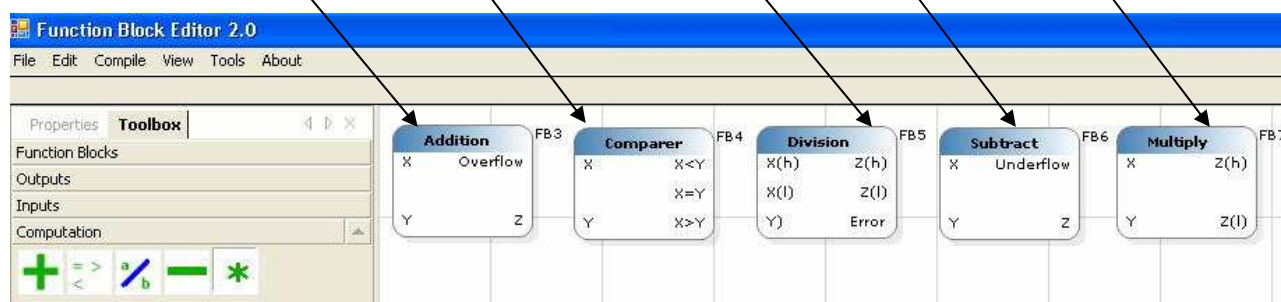
TIMER BLOCK : Time base 100 ms/1s. The output "Timed" is seted up after the delay entered (Input "Time")



Note : all blocks with "NV" (**Non Volatile**) marking have their status **stored** even if you power down. Blocks without "NV" marking are initialized on each TesysT power up (excepted LTMR blocks which also stores their status).

- **Computation blocks**

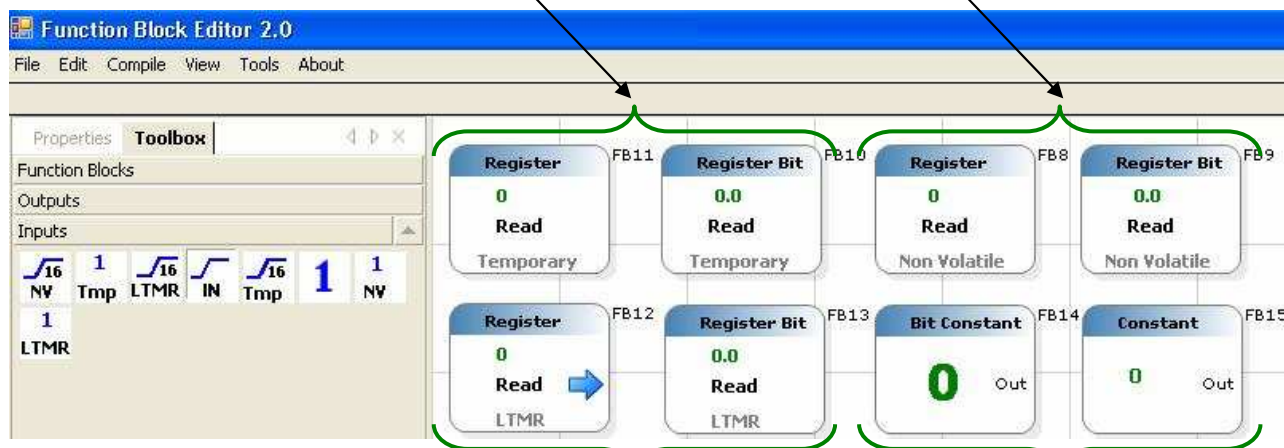
ADDITION BLOCK **COMPARE BLOCK** **SUBSTRACTION BLOCK** **DIVISION BLOCK** **MULTIPLICATION BLOCK**



- Input blocks

READING BLOCK TEMPORARY REGISTER :
You have word or bit format. You can set up a Temporary Register Value between 0 and 69.

READING BLOCK NON VOLATILE REGISTER :
You have word or bit format. You can set up a Non volatile Register Value between 0 and 31.



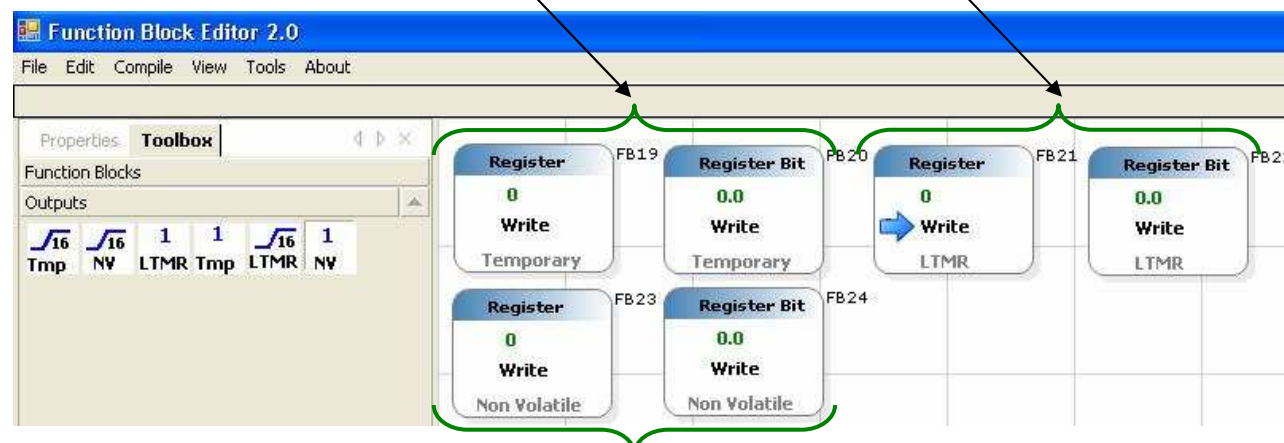
READING BLOCK LTMR REGISTER :
You have word or bit format. The LTMR Register Value matches to the TesysT mapping.

READING CONSTANT VALUE :
You have bit or word format.

- Output blocks

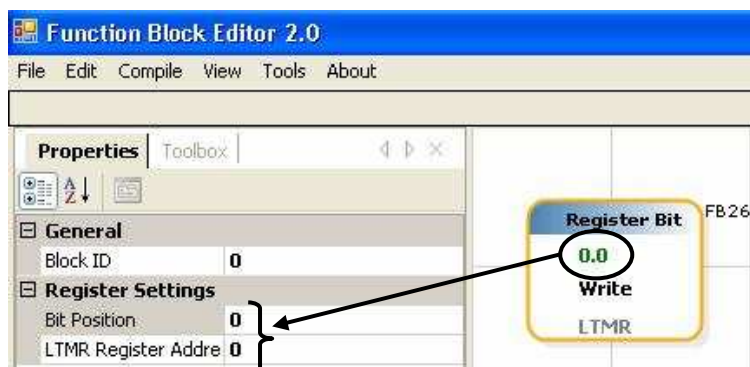
WRITING BLOCK TEMPORARY REGISTER :
You have word or bit format. You can set up a Temporary Register Value between 0 and 69.

READING BLOCK LTMR REGISTER :
You have word or bit format. The LTMR Register Value matches to the TesysT mapping.

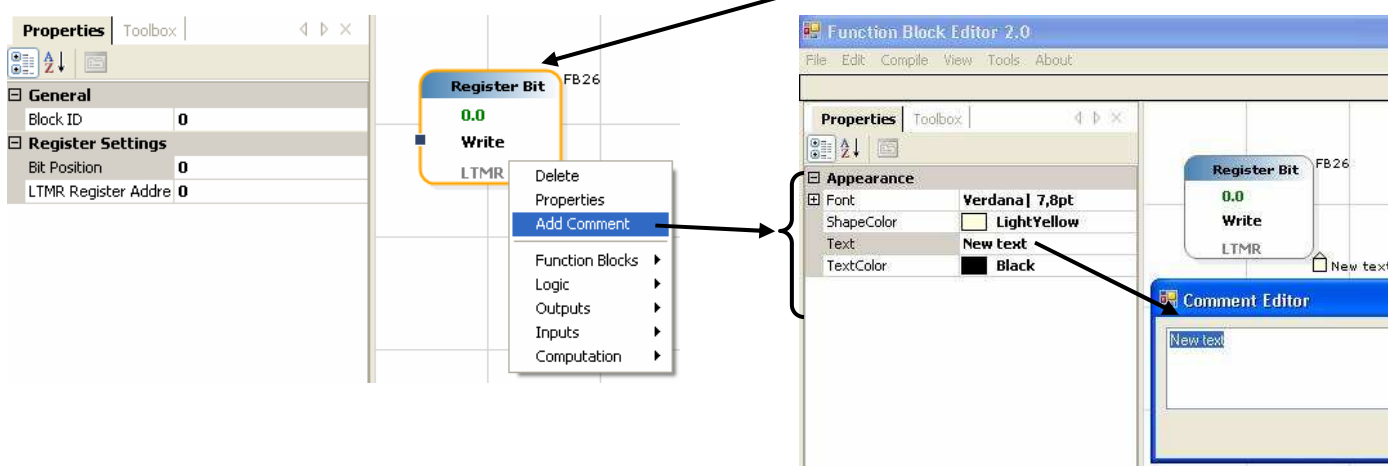


READING BLOCK NON VOLATILE REGISTER :
You have word or bit format. You can set up a Non volatile Register Value between 0 and 31.

- Characteristics of blocks

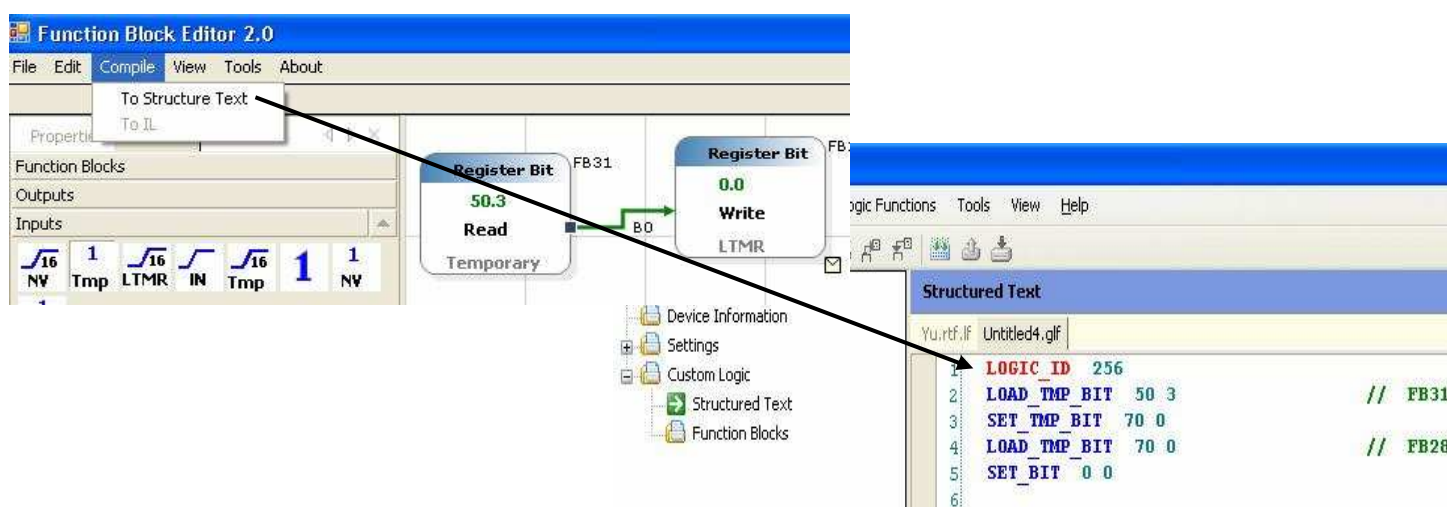


YOU CAN ADD COMMENTS FOR EACH BLOCK (SEE BELOW)

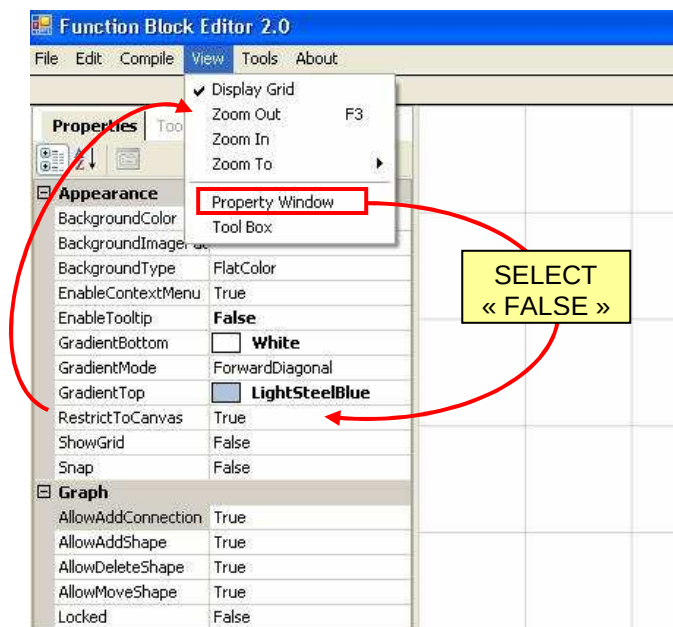


■ Compilation

Before the downloading of the program you should convert the function block into Structured Text. The program can be only transferred in Structured Text into TesysT.



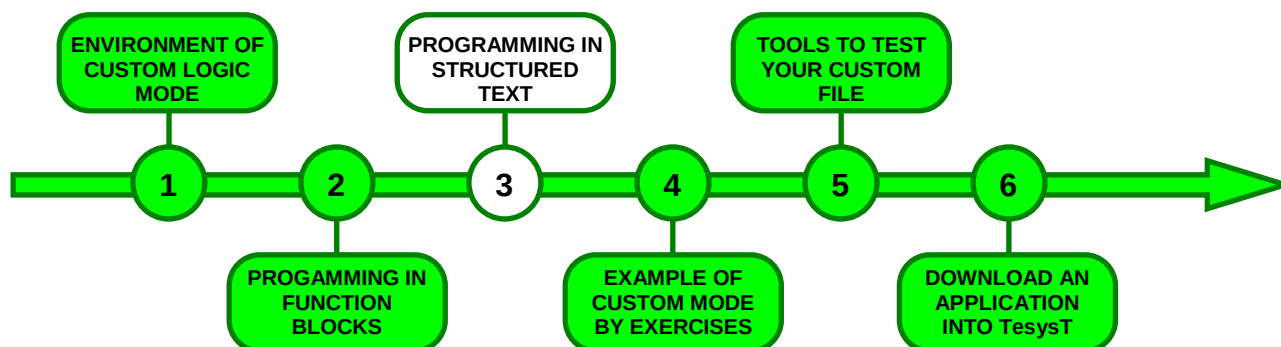
■ **Grid**



YOU CAN EXTEND THE GRID TO HAVE MORE AREA :

1) GO ON "VIEW" MENU AND SELECT "FALSE" FOR THE OPTION "RESTRICT TO CANVAS".

2) SELECT "ZOOM OUT"



■ Pre-defined program

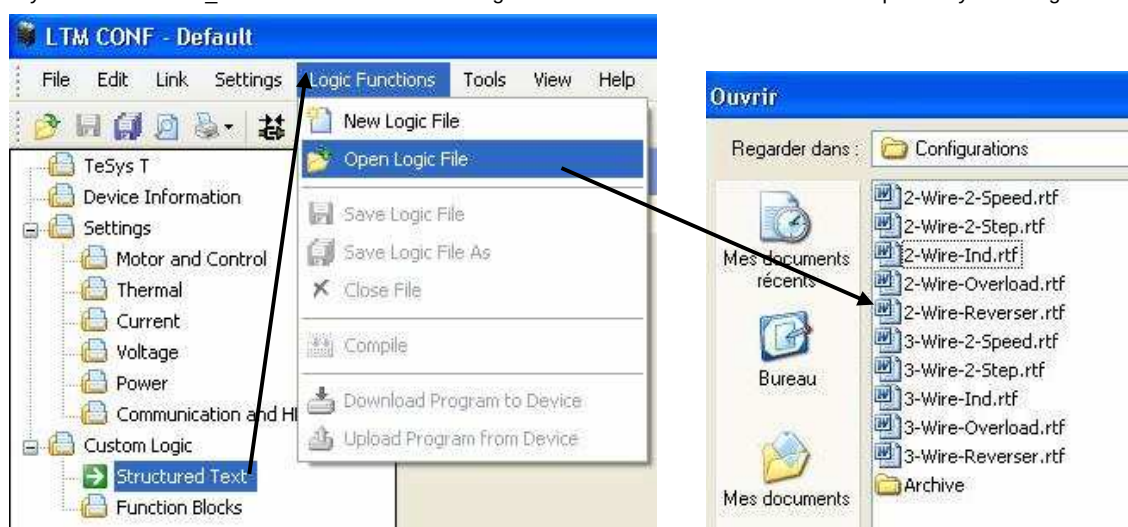
On Schneider-electric.com, you have all files in relation to the pre defined mode available on TesysT (see link below). You can read and modify a : 3 wire 2 speed mode, 3 Wire Independent mode.....

-For firmware 1.7-1.8:

[http://www.global-download.schneider-electric.com/852574FA005AF23D/all/C125713F005265E2C12574500032F444/\\$File/operating%20modes.zip](http://www.global-download.schneider-electric.com/852574FA005AF23D/all/C125713F005265E2C12574500032F444/$File/operating%20modes.zip)

-For firmware 2.1:

on Directory PowerSuite V2.5_LTMConf V4.12.0.14 : C:\ProgramFiles\Schneider electric\PowerSuite\ps1\TesysT\Configurations



■ Format of Structured Program

WRITE PROGRAM NUMBER CORRESPONDING TO THE NUMBER ENTERED INTO CUSTOM_ID (SEE SETTINGS/MOTOR&CONTROL MENU)

INSTRUCTION

```

Structured Text
2-Wire-2-Step.rtf*

LOGIC_ID 256
LOAD_BIT 683.8
SET_TMP_BIT 0.1
LOAD_BIT 457.5
SET_TMP_BIT 0.0
SET_TMP_BIT 2.0
LOAD_NOT_TMP_BIT 0.0
AND_TMP_BIT 0.1
SET_TMP_BIT 2.1
LOAD_NOT_TMP_BIT 0.0
AND_NOT_TMP_BIT 0.1
SET_TMP_BIT 2.2

// 2-WIRE TWO STEP MODE
// TS/HMI
// Debounce TS/HMI in scratch
// LI6
// Debounce LI6 in scratch
PLC Control
// LI6 debounced
// TS/HMI debounced
// HMI Control
// LI6 debounced
// TS/HMI debounced
// TS Control
                
```

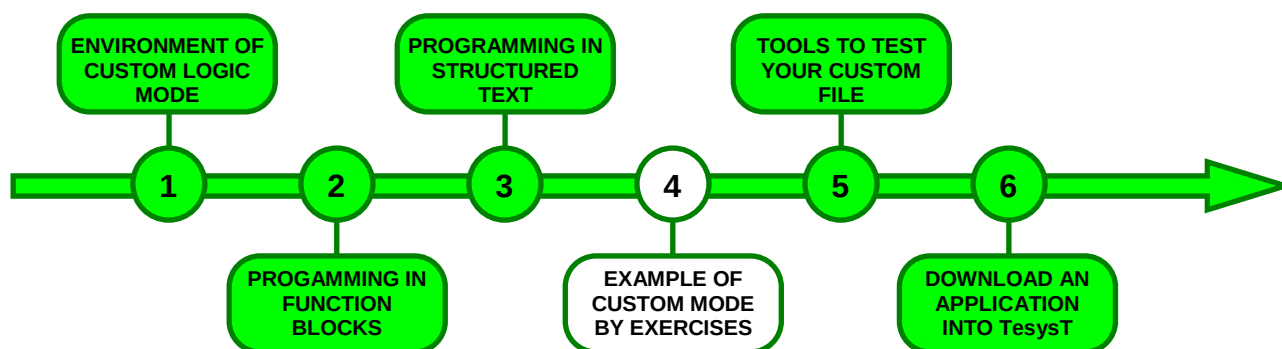
REGISTER 2 bit 1

YOU CAN ADD COMMENT INTO YOUR PROGRAM USING : « // »

In Structured Text we have a list of instruction to make the program :

- ☐ Register (Comp, And, Or, Xor, On_Set)
- ☐ Boolean (Load, Or, Set)
- ☐ Timers
- ☐ Latch
- ☐ Counters
- ☐ Math (ON_ADD, ON_SUB, ON_MUL, ON_DIV)

You can find more details about each instruction on the user manual "Custom Logic" (This manual is attached in the Technical Resolution link).



For all exercises in custom mode, there are some important registers to know :

- Register **457** bit 0 (Read register) : Input 1 status (LI1) to bit 9 for Input 10 status (LI10)
- Register **458** bit 0 (Read Register) : Output 1 status (LO1) to bit 3 for Input 4 status (LO4)
- Register **1200** bit 12 (Write/Read Register) : Output 1 Control (LO1) to bit 15 for Output 4 control (LO4)
- Registers **1301 to 1399** : memory available by PLC and custom mode. You can write registers through PLC to send values into the program customized.

At the moment you have finished Exercises A or B you can go directly on the Part 5 and 6 to simulate and test it.

■ **Exercise A :**

A customer uses the TesysT T in a Reverser application. He controls the outputs in local mode by logic inputs (LI1 to LI6). But, he wants to control the LO2 (Run Reverse) with the input 3 (LI3) and acknowledge the faults by the input 2 (LI2). In this case TesysT will use switch button.

These features match to a 2 Wire Reverser mode. By default, for this last mode this is the input characteristics:

- LI1 : Run forward
- LI2 : Run Reverse
- LI3 : free
- LI4 : free
- LI5 : Acknowledge fault
- LI6 : Local/Remote mode

Note : LI3 is a free input for all pre defined operating mode (2 wire/3 wire reverser, ind., 2 speed, 2 step, Overload).

You should open the pre defined file and make the modification :

```

1 LOGIC_ID 6 // 2 WIRE REVERSER MODE (6)
2 // Temp register allocation
3 // scratch
4 // Control Mode
5 //
6 //
7 // strip)
8 // Temp 3 as Active Control Mode
9 // 0=PLC
10 // 1=HMI
11 // 2=TS (terminal strip)
12 // Temp 4 as state bits group 1
  
```

You should change the function of LI2 and LI5. We look for Register 457 bit 1 and bit 4 (press "CTRL" + "F" to search "457 1" and "457 4" :

```

373 AND NOT BIT 704 1 // NOT PLC Run2
374 AND T
375 AND N
376 OR TM
377 SET T
378 LOAD
379 AND N
380 AND T
381 AND N
382 OR TM
383 SET T
384 LOAD TMP BIT 3 2 // TS active
385 AND NOT BIT 457 1 // NOT TS Run 2
  
```

Now, you can make "Run Reverse" by the input 3 (LI3). Also, in the exercise you should acknowledge a fault by the input 2 and not the input 5 (configuration by default). So you look for the input 5 (register 457 4) :

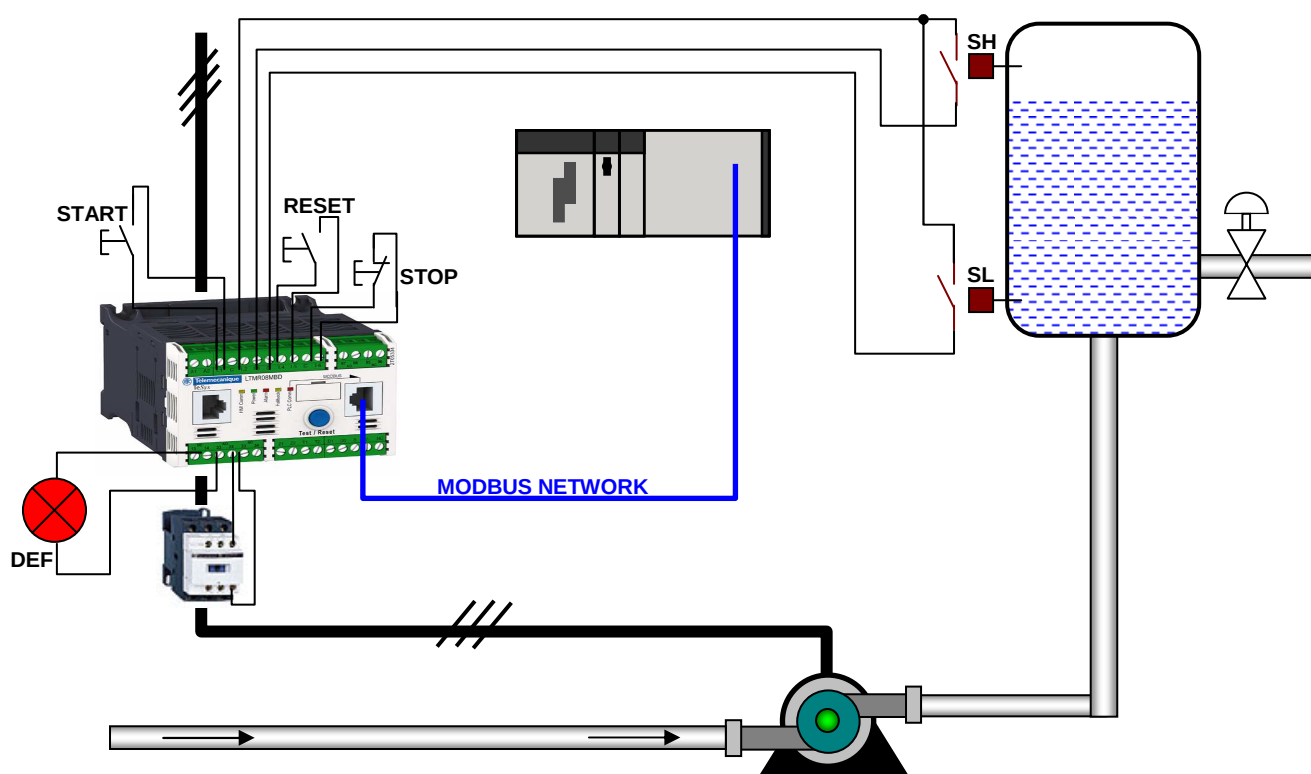
```

548 OR TM
549 AND N
550 SET T
551 SET N
552 LOAD
553 SET T
554 //
555 LOAD
556 SET T
557 LOAD
558 SET NOT TMP BIT 15 15 // Output 4 = Fault
559 LOAD BIT 457 4 // Reset Input LI5
  
```

The “exercise A” is finished. We re-designed the organization of the inputs :

- LI1 : Run forward
- LI2 : Acknowledge a fault
- LI3 : Run reverse
- LI4 : free
- LI5 : free
- LI6 : Local/Remote mode

■ **Exercise B :**



Drawing of the application for Example B

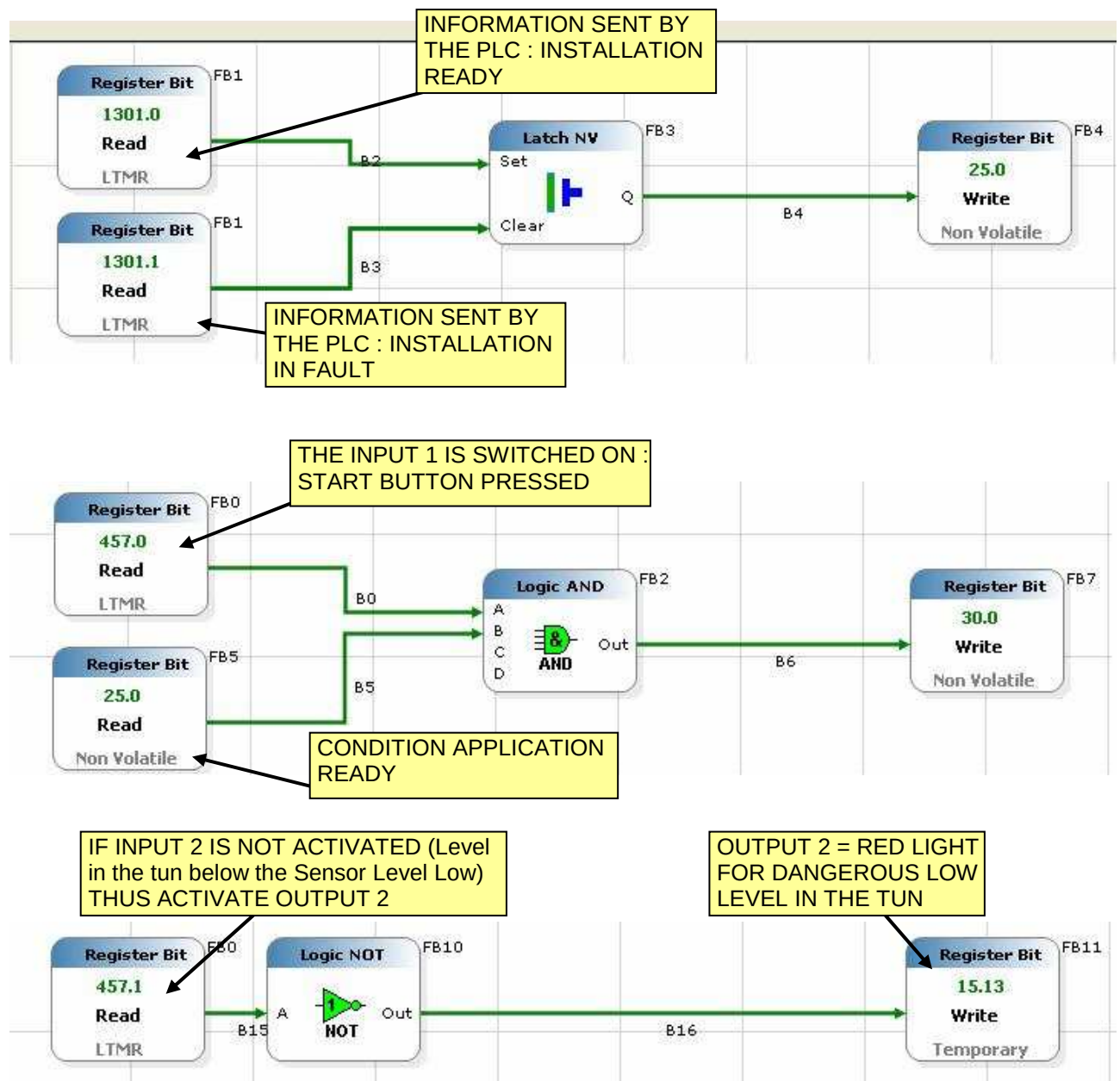
For this application, the customer's requirements are :

- Control in local mode by push buttons (“START”, “STOP”) to fill the tun
- Sensors indicate High Level (“SH”) and Low Level (“SL”) for the tun
- The Sensor High Level stops the motor when the tun is completely filled
- The light “DEFAULT” switches on when the tun level is below the sensor “SL”
- The light “Default” can be switched off by the push button (“RESET”)
- You start the motor if you push the button “START” and if the PLC has sent the information “INSTALLATION READY” by a bit on the network
- You can stop the motor with the push button “STOP” or if the PLC has sent the information “INSTALLATION IN FAULT” by a bit on the network.

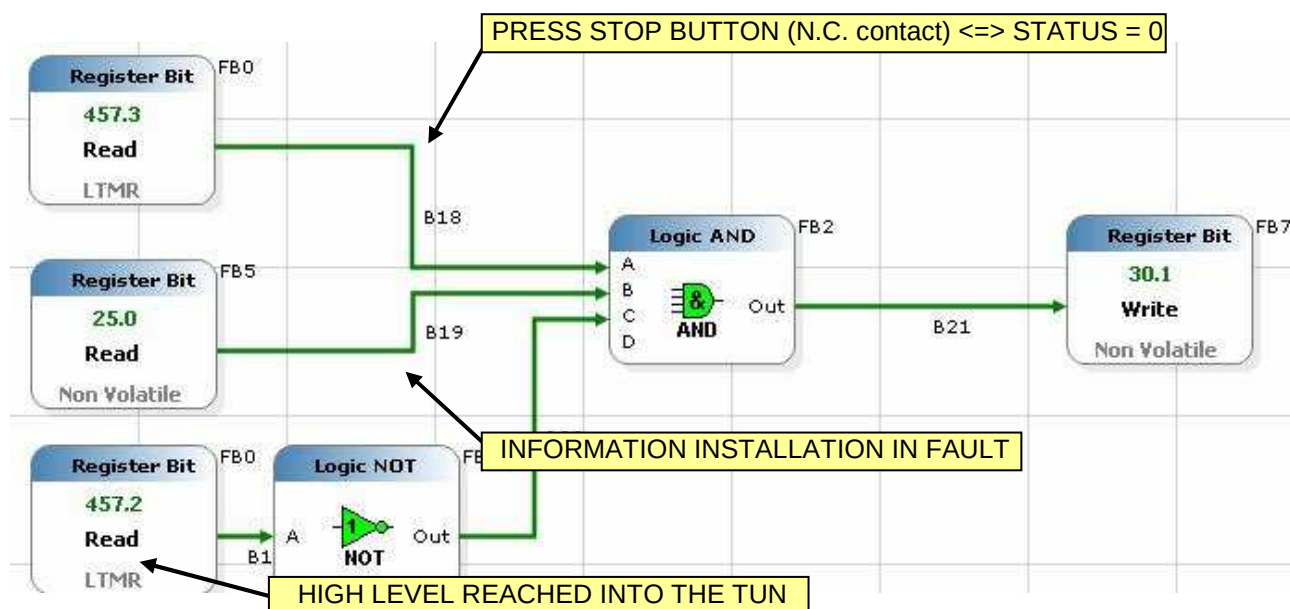
This application matches to a "3-Wire Independent mode". By default, the features for a 3 wire Independent mode are :

- LI1 : Run
- LI2 : Switch On Output 2
- LI3 : free
- LI4 : Stop Run
- LI5 : Reset fault
- LI6 : Local/Remote mode

Now we create the behavior that the customer wants in Function block (Program for a TesysT Firmware 1.8):



YOU CAN STOP THE MOTOR EITHER
BY : PRESSING STOP BUTTON
OR BY : READING INFORMATION INSTALLATION IN FAULT
OR BY : REACHING HIGH LEVEL IN THE TUN



The function block is finished, you can convert the program above and you copy/paste (expect the first line with "LOGIC_ID") at the end of the program "3 wire independent mode" already available into PowerSuite.

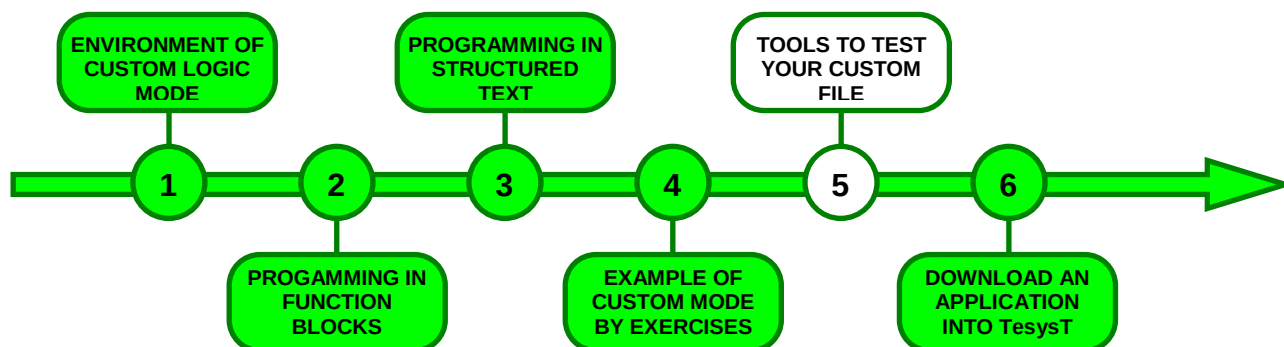
You should change the LOGIC_ID "5" to a right value (256 to 511).

Also, in the program "3 wire independent mode" the output 2 is already used so you should put in comment the line involved (see Temporary bits 15.13) to avoid perturbation with the exercise made before. You remove the condition 457 0 by the Temporary bit 30.0 and the condition 457 3 by the temporary bit 30 1.

So we open a "3 wires Independent mode" and modify regarding customer's requirements :

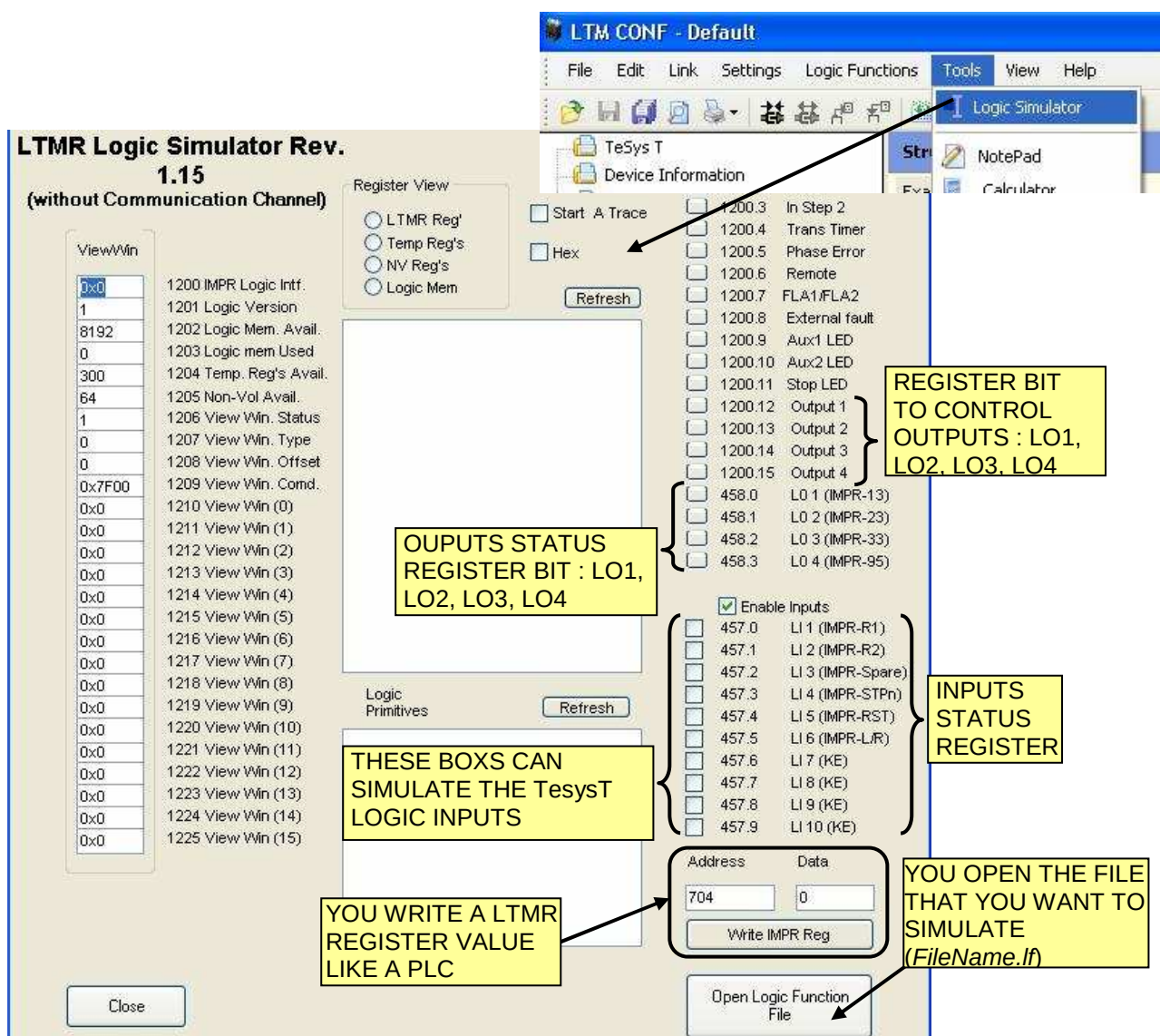
- LI1 : Start
- LI2 : SL
- LI3 : SH
- LI4 : Stop
- LI5 : Reset fault and Stop light
- LI6 : Local/Remote mode
- Output 2 : Light Low level in the tun

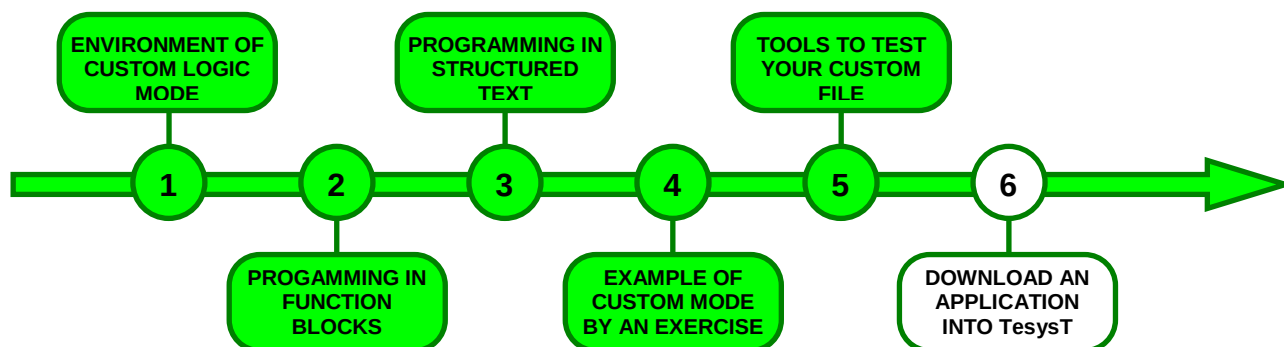
Note : You can find all the results of these exercises (Function block and Structured Text files) on the Technical Resolution link.



■ Simulator

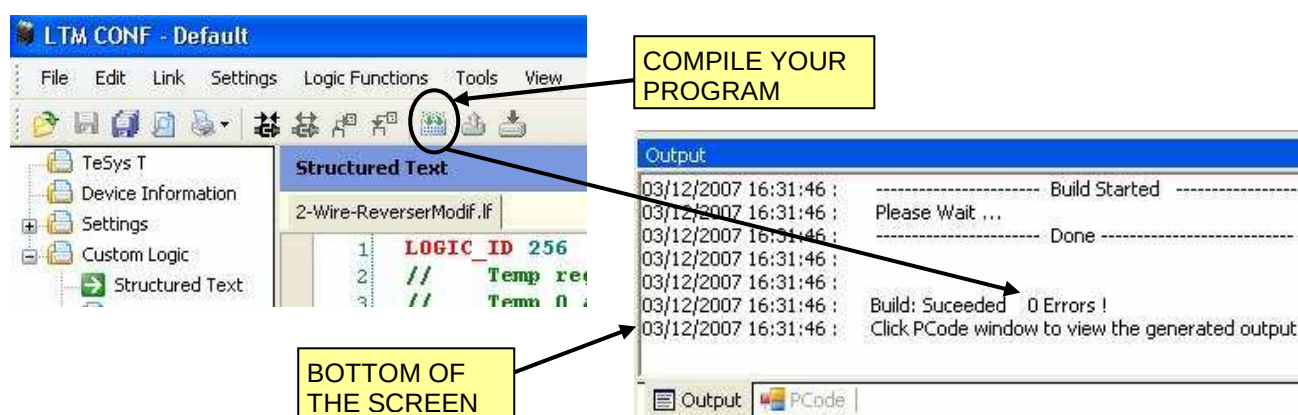
You can test all our programs with a tool “Logic Simulator”. This tool is only available to check a Structured Text program or Function Block converted into Structured Text. Before testing, we you must compile your program (for more details regarding compilation see Part 6).



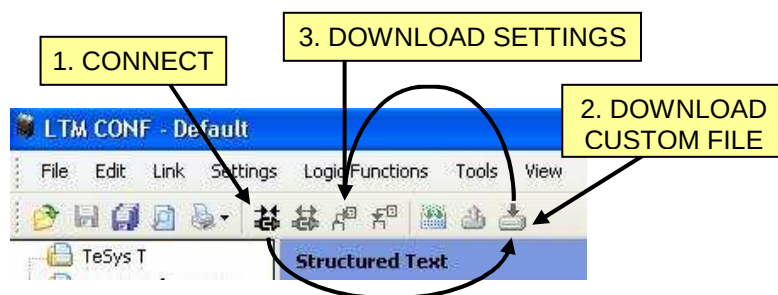


You should execute two steps to download a Custom application into the TesysT :

■ Step 1 : Compilation



■ Step 2 : Download the settings & Custom File



Note : after the first downloading, for each custom modification, you download only the custom file.

You will transfer again the "settings" if you change the "CUSTOM_ID" or a setting