ME 576 Laboratory Assignment Session 7

Generate PLC programs and the truth table to verify the results. Store the programs in a USB 2.0 flash drive and simulate them on the NUM PLC.

1)

E0

E1 E3 A0

E2

2) Write a short program to perform the following operation according to two inputs I0 and I1.

operation: 79  85 (Send output to AN2)

I0 (E1) 0 0 1 1

I1 (E2) 0 1 0 1

 and or nand xor

3)

E0 E1

E2 A8

E3 E4

A9

E5

4) Operation: Motor control problem

Motor = (Motor + start) AND stop AND clamp1 AND clamp2

Use the following variables:

Motor: A0

Start: E1

Stop: E2

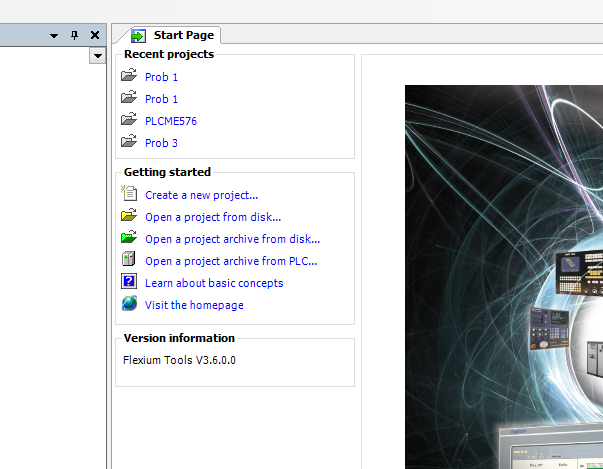
Clamp1: E3

Clamp2: E4

**Procedures for PLC simulations on NUM machine:**

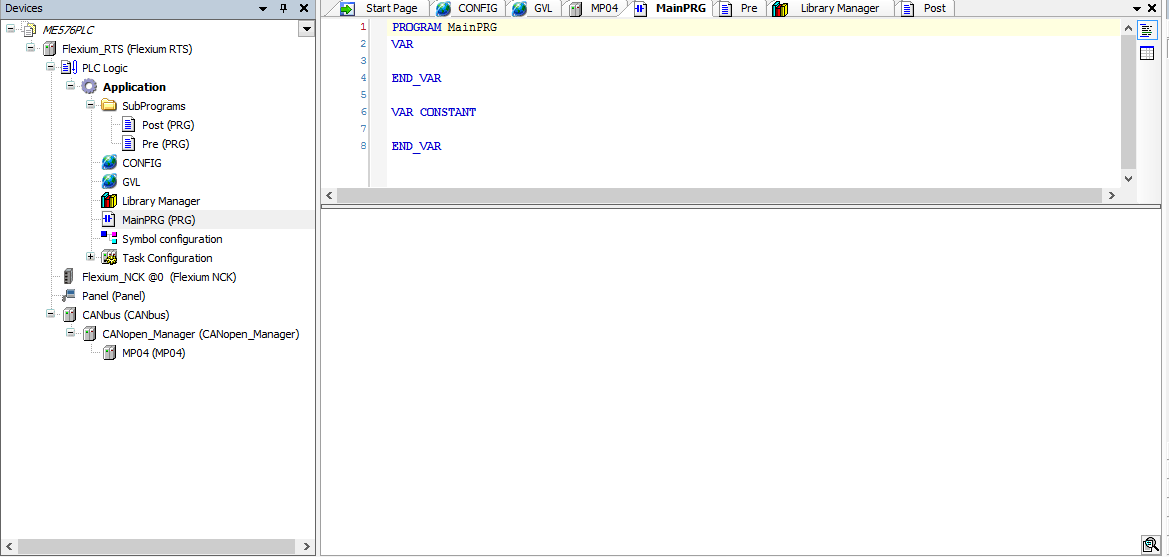
1 - Open the ME576 project template

Start Flexium Tools software in NUM machine. Under “Getting started”, click on “Open a project from disk”. Select “ME576PLC.project”. The variables in the template have been pre-mapped to the correct addresses of the PLC numpad.



Click on Finish to accept all default NUM project configurations.

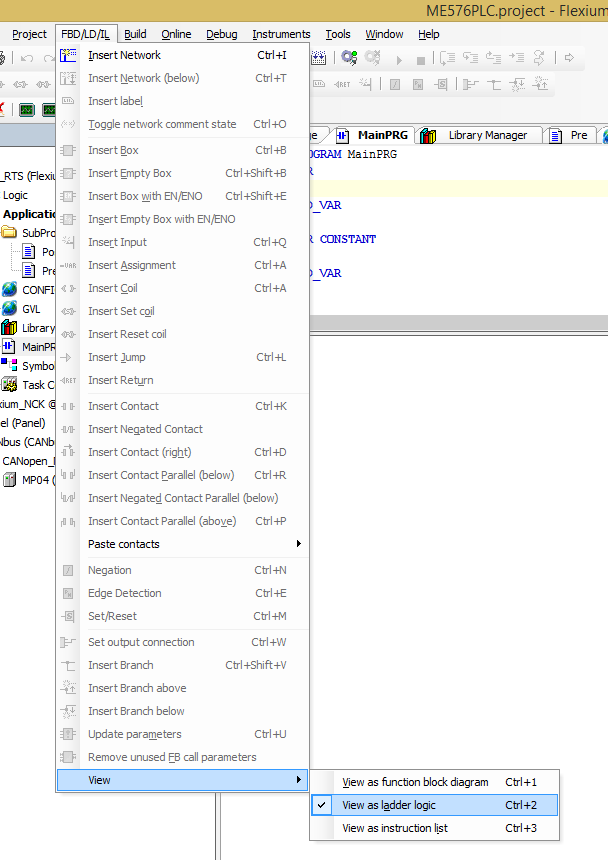
2 – Start programming in MainPRG (under PLC Logic → Application)



The following operators can be used for PLC programing in this lab:

| **Operator** | **Modifiers** | **Meaning** | **Example** |
| --- | --- | --- | --- |
| LD | N | Loads the (negated) value of the operand into the accumulator. | LD iVar |
| ST | N | Stores the (negated) content of the accumulator into the operand variable. | ST iErg |
| S |  | Sets the operand (type BOOL) to TRUE when the content of the accumulator is TRUE. | S bVar1 |
| R |  | Sets the operand (type BOOL) to FALSE when the content of the accumulator is TRUE . | R bVar1 |
| AND | N,( | Bitwise AND of the accumulator and the (negated) operand | AND bVar2 |
| OR | N,( | Bitwise OR of the accumulator and the (negated) operand | OR xVar |
| XOR | N,( | Bitwise exclusive OR of the accumulator and the (negated) operand | XOR N,(bVar1,bVar2) |
| NOT |  | Bitwise negation of the accumulator's content |  |
| ADD | ( | Addition of accumulator and operand, result is copied to the accumulator . | ADD (iVar1,iVar2) |
| SUB | ( | Subtraction of accumulator and operand, result is copied to the accumulator. | SUB iVar2 |
| MUL | ( | Multiplication of accumulator and operand, result is copied to the accumulator. | MUL iVar2 |
| DIV | ( | Division of accumulator and operand, result is copied to the accumulator. | DIV 44 |
| GT | ( | Check if accumulator is greater than or equal to the operand, result (BOOL) is copied into the accumulator; >= | GT 23 |
| GE | ( | Check if accumulator is greater than or equal to the operand, result (BOOL) is copied into the accumulator; >= | GE iVar2 |
| EQ | ( | Check if accumulator is equal to the operand, result (BOOL) is copied into the accumulator; = | EQ iVar2 |
| NE | ( | Check if accumulator is not equal to the operand, result (BOOL) is copied into the accumulator; <> | NE iVar1 |
| LE | ( | Check if accumulator is less than or equal to the operand, result (BOOL) is copied into the accumulator; <= | LE 5 |
| LT | ( | Check if accumulator is less than operand, result (BOOL) is copied into the accumulator; < | LT cVar1 |
| JMP | CN | Unconditional (conditional) jump to the label | JMPN next |
| CAL | CN | (Conditional) call of a PROGRAM or FUNCTION\_BLOCK (if accumulator is TRUE) | CAL prog1 |
| RET |  | Early return of the POU and jump back to the calling POU | RET |
| RET | C | Conditional - if accumulator is TRUE, )early return of the POU and jump back to the calling POU | RETC |
| RET | CN | Conditional - if accumulator is FALSE )early return of the POU and jump back to the calling POU | RETCN |
| ) |  | Evaluate deferred operation |  |

3 – Convert the program from instruction list to ladder diagram to verify your code.



4 – Simulate PLC program using NUM numpad buttons

