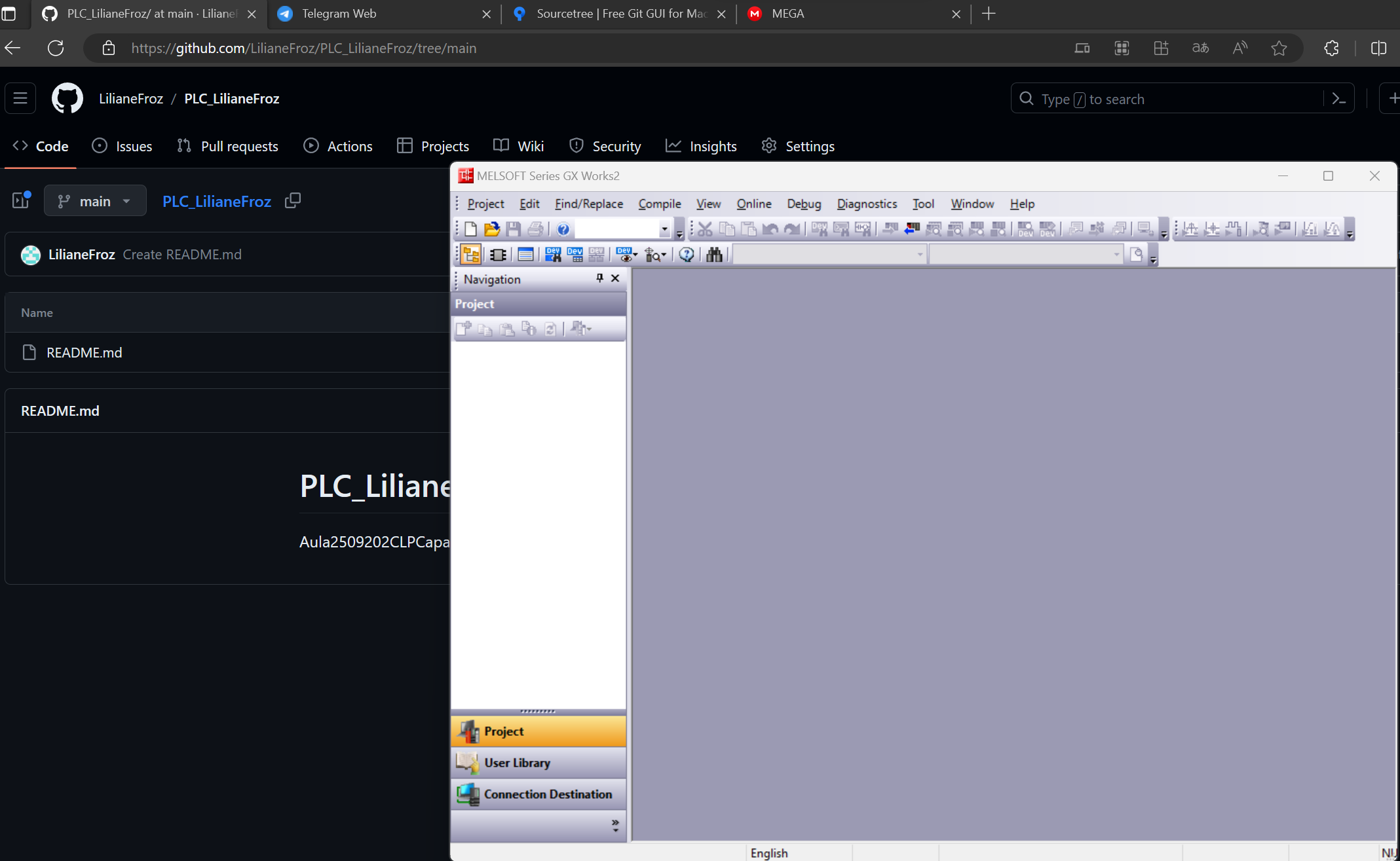
25/09/2023 Segunda-feira

Após criar uma conta no Git Hub e instalar o SourceTree, devemos ir em READEME na página do GIT, e lá criarmos para depois Puxar.



Nas máquinas do laboratório já estão previamente instaladas os softwares dos CLPs.

Os CLPs com telas costumam ser os com maior eficiência. Há linhas micro

Rockwell – lockend (O mais em conta), o software é free também.

Há dessa marca CLPs também com IHM (Interface Homem Máquina).

Da Siemens, o software infelizmente não é free. Para popularizar o protocolo Internet IP, com uma característica que torna ele ainda mais preciso CIP (O que a torna muito mais precisa com uso de realtime. Camada de transmissão de dados, protocolo da internet TCP.

A profnet, desenvolvido pela Siemens. A Switch da Siemes é mais completa, porém os custos para maior complexidade chega a custar R$50K. A ideia é que num computador industrial se programe em Ladder, todas as linguagens, no desenvolvimento de software rege em torno do Function Block. CPS RLobo

Shneider, tem um leque menor que as citadas anteriormente. O foco da empresa foi comprar empresas, gerenciamento (geração) de energia. MODCON (primeiro CLP desenvolvido) com um custo benefício muito, cerca de R$600,00. O Codcis é de uma empresa alemã, programação de periféricos, com a norma que rege programação de CLP. M580 já seriam os modelos mais avançados, com IO embarcado.

A próxima marca seria a pioneira dos CLPs baseados em computadores, a Beckhoff. Seu nicho não eram CLPs, o dono dela ainda está vivo,

“que é previamente Olimpo sabia na mesma faixa de miss em desenvolver logo o sistema por isso tem um sistema operacional e ao pai no sistema de cima desse sistema padrão ele torna o seu computador com sabe PE aí qual o ponto eles desenvolveram no início de 2001 protocolo chamado veta kete funciona do modo mais 3 leis e quando se trata de questões de desenvolvimento de soft no master é ele tem é simples implementação você consegue alterar aí pra tornar esse computador lá 11 marketing você consegue controlar elementos de de automação que trabalha com esse protocolo então é posso sim então você trata de controle e é muito legal você pode rodar aplicações vai fazer uma imitação interna pra que role uma camada de otário e comunique com arroz por exemplo aqui as é o exemplo de de comunicação né esse cara cpu aqui roda um “

Usando o US, Universal Robótics, ele é... A IDE é free. Seu código só roda naquele computador caso tenha a licença. Também é possível usar no Yaskawa. CODCIS, no visual estúdio, C, C++, Division DI robótics, porém você sem a licença não é possível rodar.

As IDEs devem ser citadas, mas não deve-se focar nelas para que o impacto dessa ferramenta seja da melhor forma possível.

Código interno de um leitor de QR Code.

O que é a “main”?

Estruturação, funções,

Funtion blocks: Controlação de objetos de classe. Com vários motores, é possível substanciar.

Dessa forma se expande métodos.

Protocolo inicial

Exemplo: Modbus, internet IP.

Um método usado para obter o código com o escâner. Ao ler, por exemplo, é preciso limpar esse “Buff”.

A Schneider nos últimos anos, foi a empresa que mais sofreu com ataques cibernéticos. Stoonets? Com um pendriver havia um vírus que mudava o controle dos rotores. Por conta disso hoje a grande maioria dos seus modelos possuem algumas etapas. Antigamente com um fireshark era possível descobrir a senha. Hoje por exemplo há duas redes, Ethernet 1 e 2.

Qual o primeiro foco de um projeto recém lançado? Custo e Qualidade.

Hoje trabalha-se muito com Remote IO, com um bloco de IO distante, sem fios, ligado a rede tal qual estivesse ligado a um fio. Esse meio veio barateando, apesar de ainda ser caro e se manter.

Dos modelos que temos o EPSON é único no modo me que ele independente. Os restantes são slayers.

JOB/TASK

* Lei do Mínimo esforço

Preciso usar fio? Sensores? Não há como resolver usando software?

Ladder é melhor para esteiras.

ST (estruturado) é voltado para maiores complexidades.

A LG usa mais a Mitsubichi e a LS, quais tem muito semelhantes.

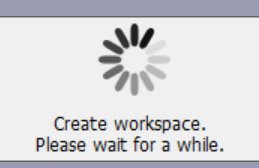
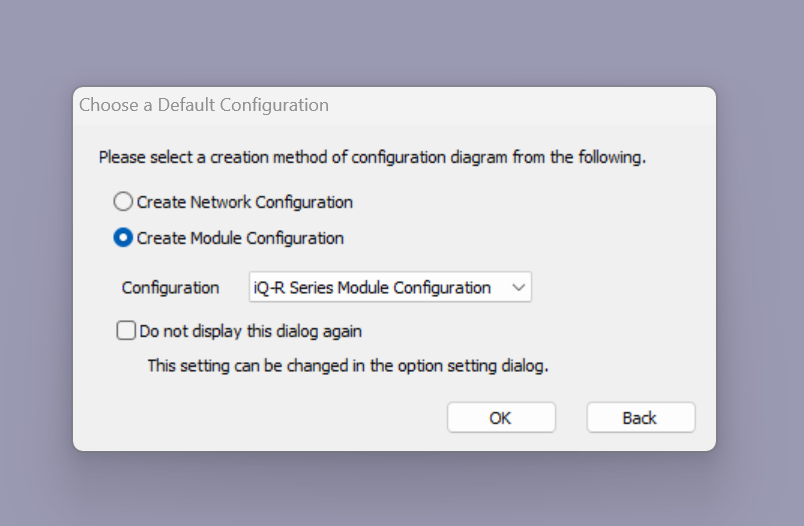
ICR

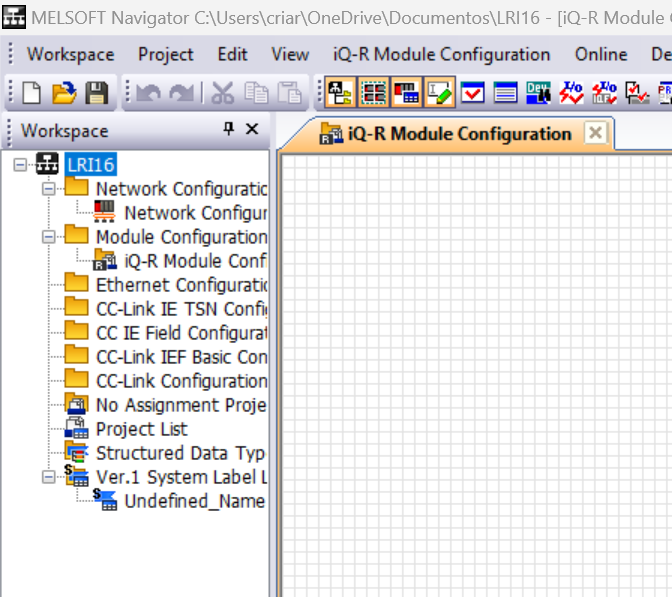
Servo motores são muito compactos, contudo os protocolos de comunicação

Qworks versão 2.78.

GX Works 2

* Melsoft navigator – Um software para organizar os periféricos em um único ponto.





Q38B - Suporte

Q61P - Power

Q03udvcpu - CPU

qy41p - Placa

DUT – Tipos de usuários criados, dados mais simples do ângulo de CLP IHM. Label dentro do CLP qual não se consegue exportar.

Não utilizar variáveis globais, qualquer outra coisa dentro do CLP não utilizar variável com paradigma de CC e C++. Uma regra básica de desenvolvimento: 0 global.

No geral o sistema lê da esquerda pra direita, de cima para baixo com fluxo de dados indo para a direção da seta mais grossa ao lado.

É indicado nível de segurança, segurança e liberdade de gerenciamento.

IMPORTANTE:

* Previous State, pulo duplo ao ligar e desligar.
* Interrupt Point \_ Pino físico para aplicar linha de códigos. Colocar elementos aleatórios que poderiam acontecer.

OutPut Mode at STOP to RUN

()Previous State

() Recalculate (Output is 1 scan later)

Área de Leitura: TASKs.

Caso queira fazer uma sign de

Basta fazer um texto indicando valor das variáveis.

EXECUTE TYPE; WAIT:

SCAN e FIXED SCAN: Tempo de leitura aberto, não segue a risca a leitura no sentido, chamado também de Freeweel (roda livre). Mas para itens críticos, deve ser fixado um escâner.

Máquina de estado: Representação do gerenciamento de atividade, com uma sequencia de atividades, no caso o SFC (Seach).

Em caso de emergência na CPU, o Output é mantido em Turnoff.

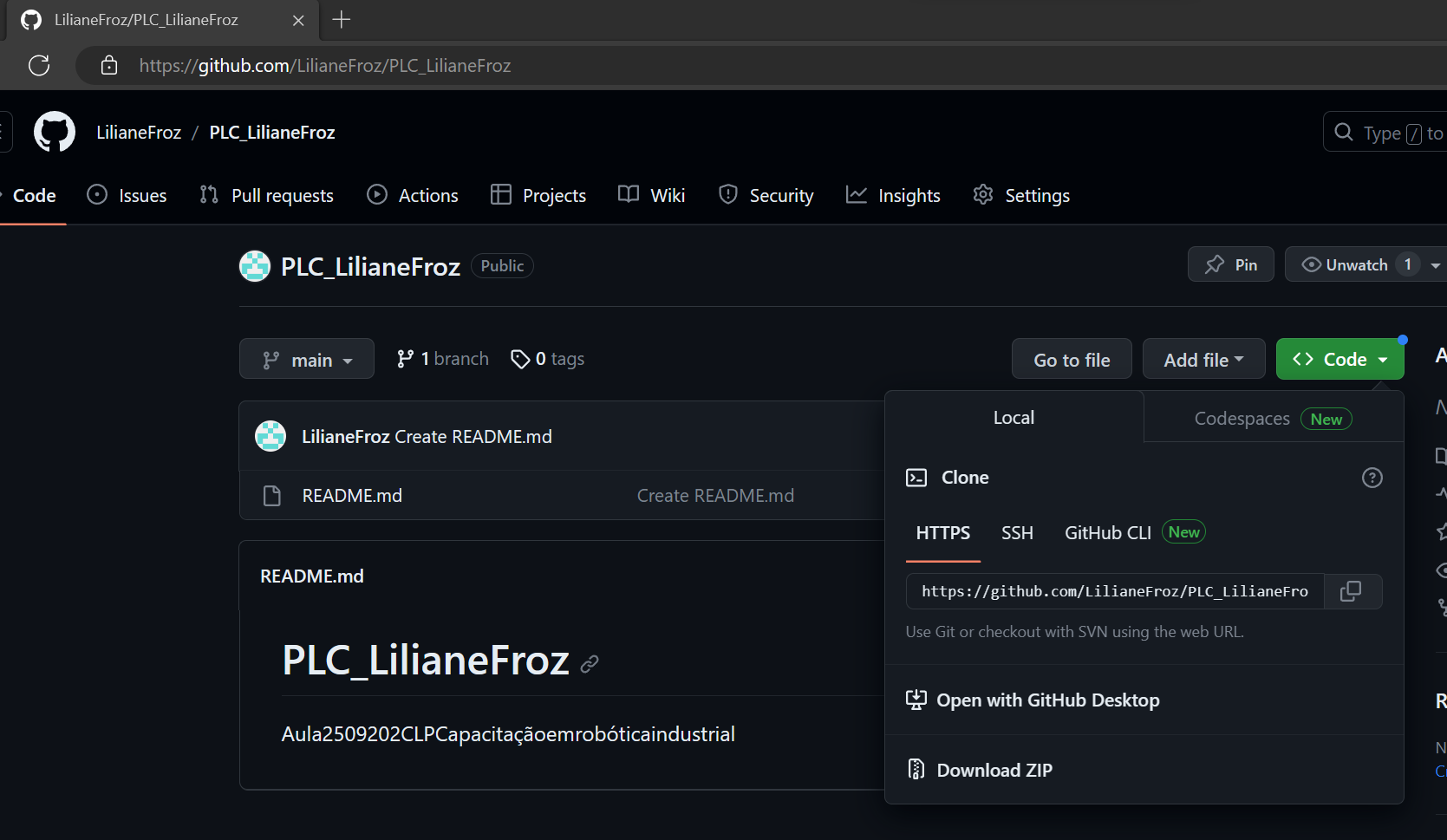
Devise, gerenciamento de bloco de memória, Data type utilizado em CPU.

Imput relay

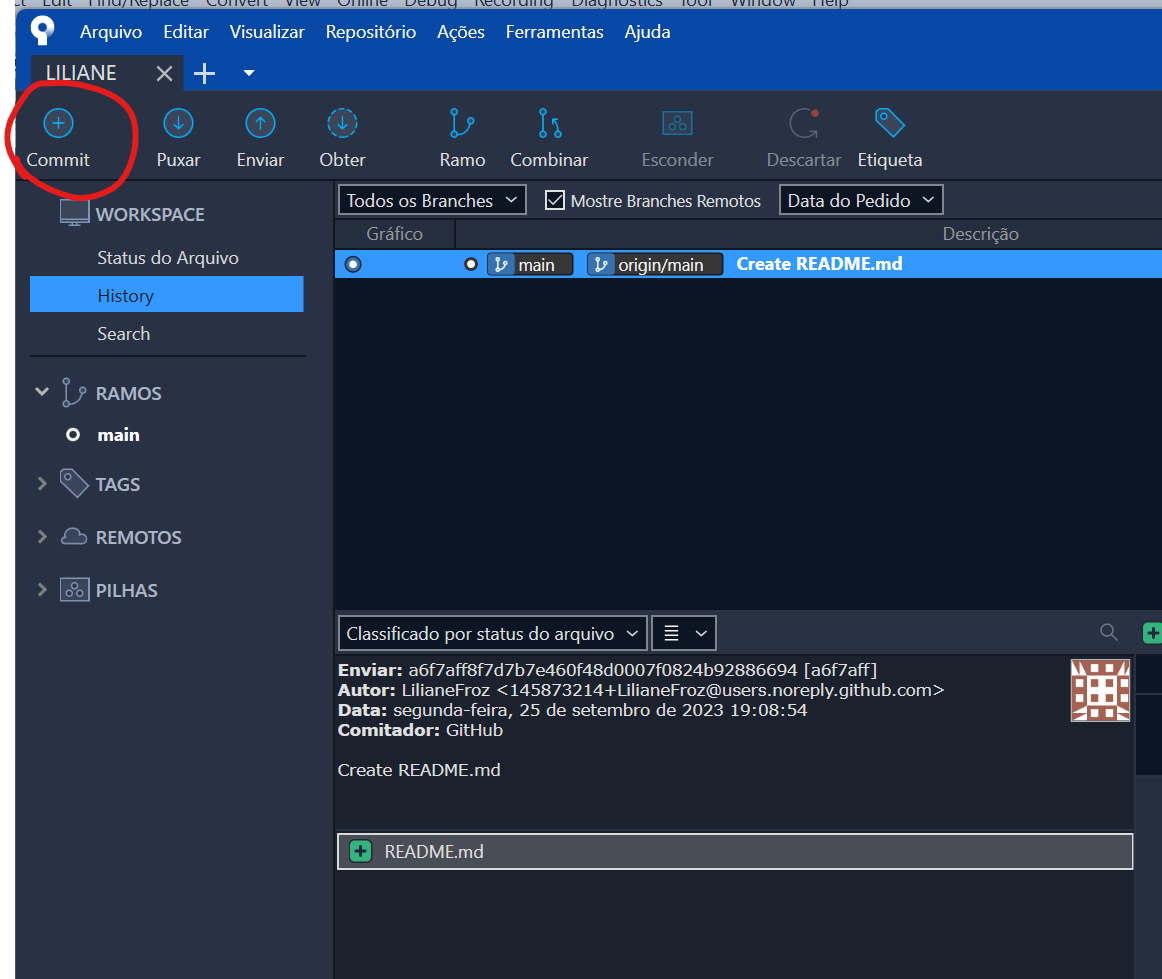
Output relay

Conforme a necessidade, é possível transferir capacidade para os tipos de tipos (Type tipes) como por exemplo: Tenho bastante memória interna, e preciso de registro de dados, posso transferir um pouco da memória interna para esse último.

26092023

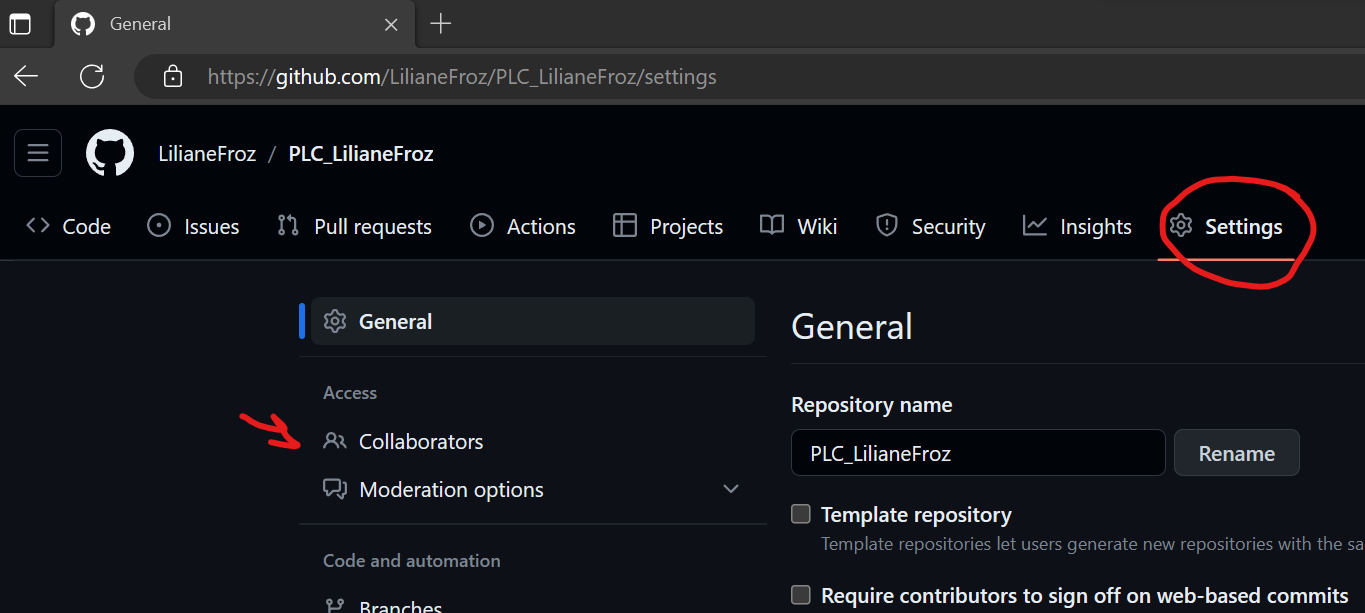


No Sourcetree selecionar:

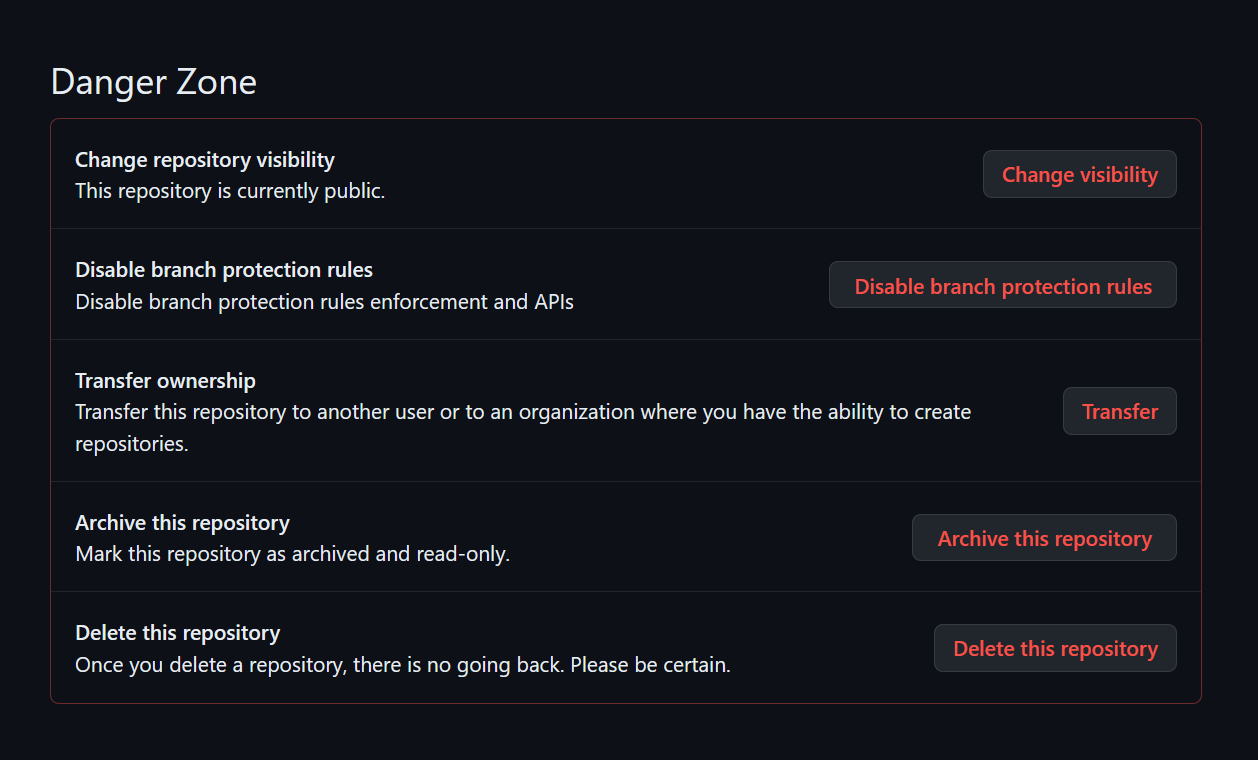


diegomachadoifam

[diego.machado@ifam.edu.br](mailto:diego.machado@ifam.edu.br)



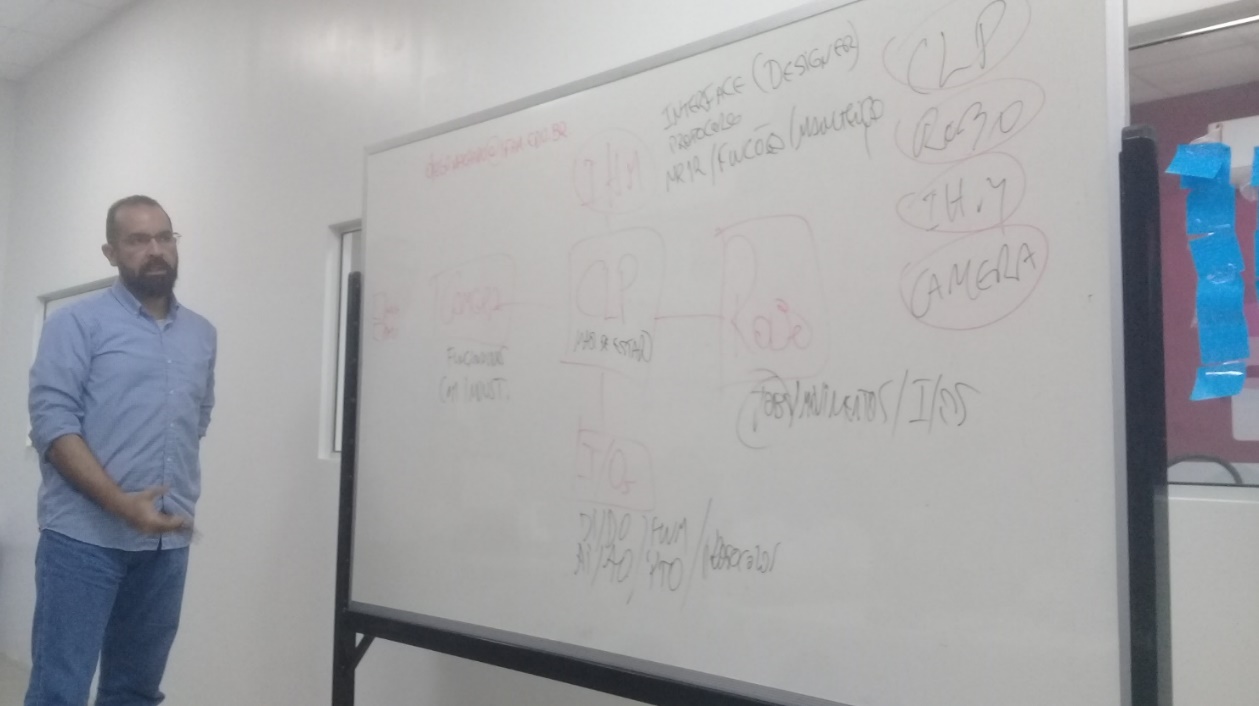
Ao final da página, em Configurações, “general"



Sempre que for iniciar algo no programa, no Sourcetree, deve-se puxar.

Como seria a organização do projeto de vocês

* CLP
* Robô
* IHM
* Câmera
* IO's (Sensores atuadores)



CLP

Camera

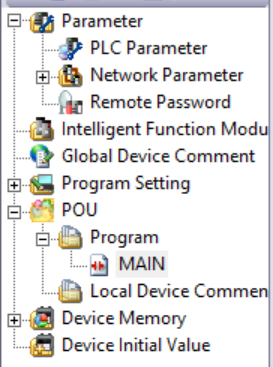
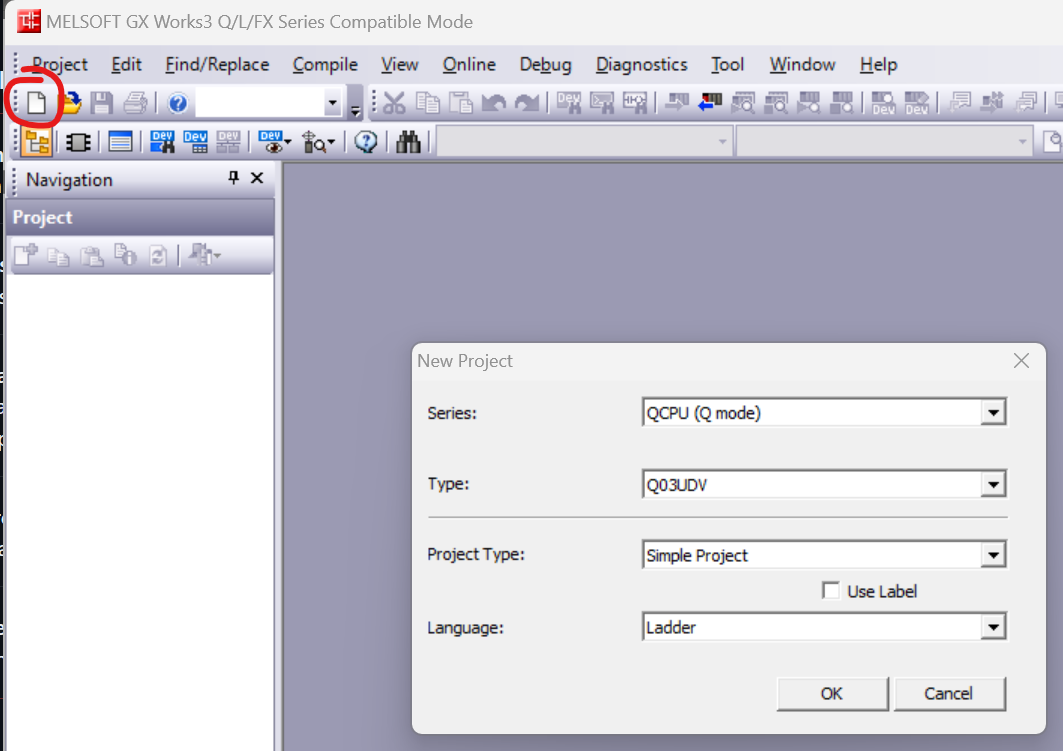
Robô

IHM

I/O’s

Funcionário com ind

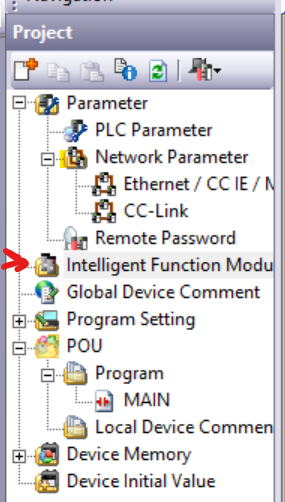
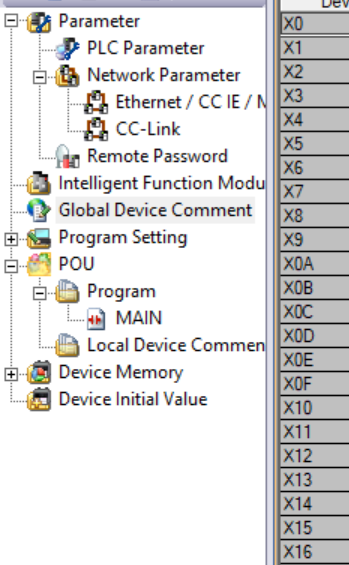
Di, Do, Ai, Ao, FWO, PTO



Através de memória um dado entra no sistema computacional é passado validado processado.

É preciso compreender os tipos de dados (Bit, world, CC2);

Como eu consigo armazenar, como não crashar meu sistema?



PWM -Pulse with Modulation (modulação por FM)

PTO – Pulse Train Output (AM)

CLPs são usados assim para controle, porém é preciso um Feedback, e também a quantidade de fios (perdendo portas)

Conforme a figura abaixo, notamos o “X” que, no jeito Mitsubichi de ser, indica entrada.

‘Mascaramento” lógica binária, bit a bit.

X0Y0, entrada e saída física.

1

0

Digital Imput = X[0~1FFF]

Digital Output = Y[0~1FFF]

Internal relay M[0~#]

DO (Double) – Word = D[0~#] - Como guardar um boleando?

T (Time)

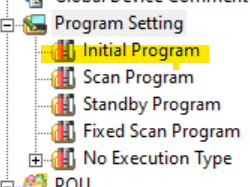
Local Variable (variável flutuante para não confundir com essas vaiáveis padrão, dessa forma utilizando o device dela e não o nome).

Roda de subida, roda de descida, PULSO )Sinal degrau

Rising Rtrig (ascensão)

Array com 10 posições de bits: Bit 09?

Não adianta querer otimizar seu código se tudo está localizado em um único lugar como o “Scan”.

CLP seja uma máquina? O CLP é rápido? Para quê?

É preciso entender em quê será aplicado e onde é mais eficiente.

O que é o programa inicial?

Tudo o que está sendo lido, é cíclico.

Wailetrue?

Criei várias variáveis e não quero correr o risco dela sofrer um nicho de memória, deve-se colocar essas informações na “initial Program”.

Em exemplos como a de envase de líquidos em carroceis com oscilação, ou ainda a marcação de um produto, deve-se salvar as informações em “Fixed scan Program”.

Ou seja, se tudo ficar no “Fixed Scan Program”, aumenta o processamento porém caí o desempenho.

Prioridades e segurança

Particionamento

Exemplo: Motoresteira1 (LigarMotor ()

Motoresteira1 (LigarMotor (ON)/(TRUE)

* Program BLOCK
* Funtion BLOCK

Métodos, saídas, variáveis. Não roda independentemente, só no programa.

* Funtion

FUN, ela só executa, tem só uma saída um único objetivo.

Precisa entrar no structured tipe

Locktal de – 0 a 8, hexadecimal

27092023

A função tem vários imputs e apenas uma output.

Quando se tem o programa (Project tree), no GX WORKS3, gera-se um local label.

O relé é quem fecha o contato em uma bobina eletromagnética. Podem ser normalmente aberto, normalmente fechado.

* Switch (Chave Imput normalmente aberto) I----I I----I
* Coil (Bobina - OutPut) I----( )----I
* NC (INC)

I----I/I----I

* NF (PT)

Com os dois acionados, no paralelo, há um curto?

I------I/I------I

I----I/I----I

Processamento desperdiçado?

Lógica Booleana. Lógica “OR” (Ou)

00 true false?

S (Shift): Inicia ligado,

And/ e/ multiplicação

I----I I--------I /I----I

0\*0=0

1\*0=1

0\*1=1

1\*1=0

I------I I------I

I----I/I----I

0\*0=0

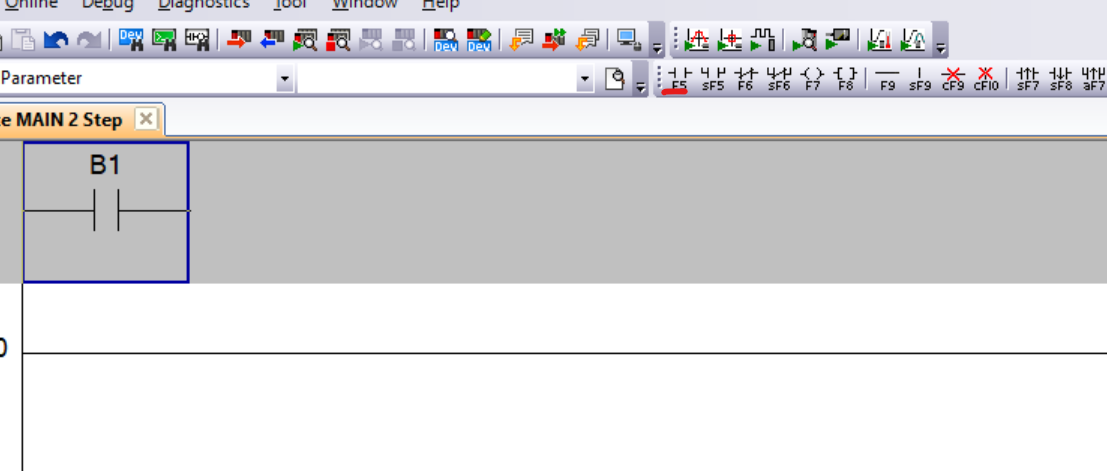
1\*0=1

0\*1=1

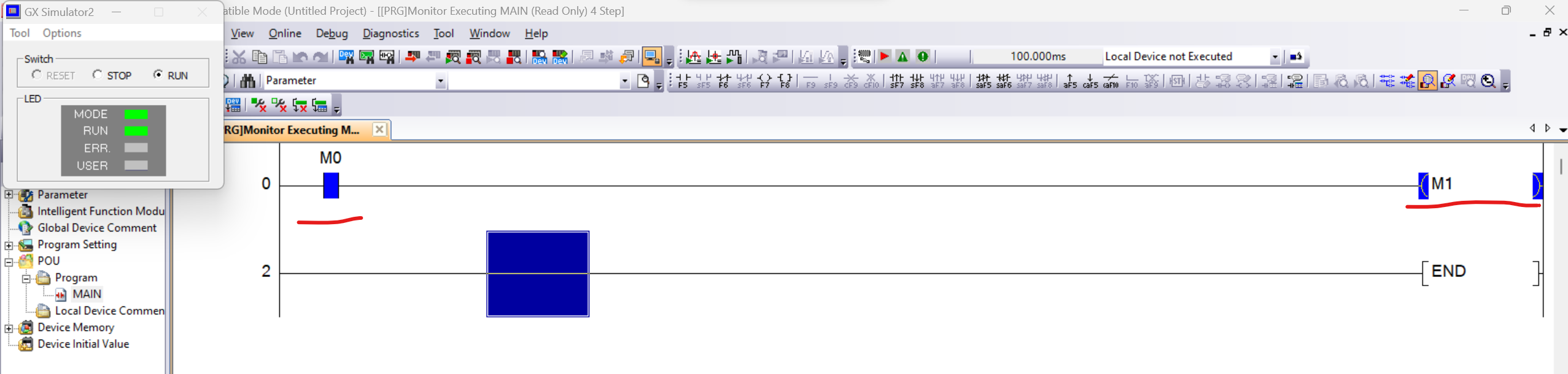
1\*1=0

|  |  |  |
| --- | --- | --- |
| A1 | A2 | OUT |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

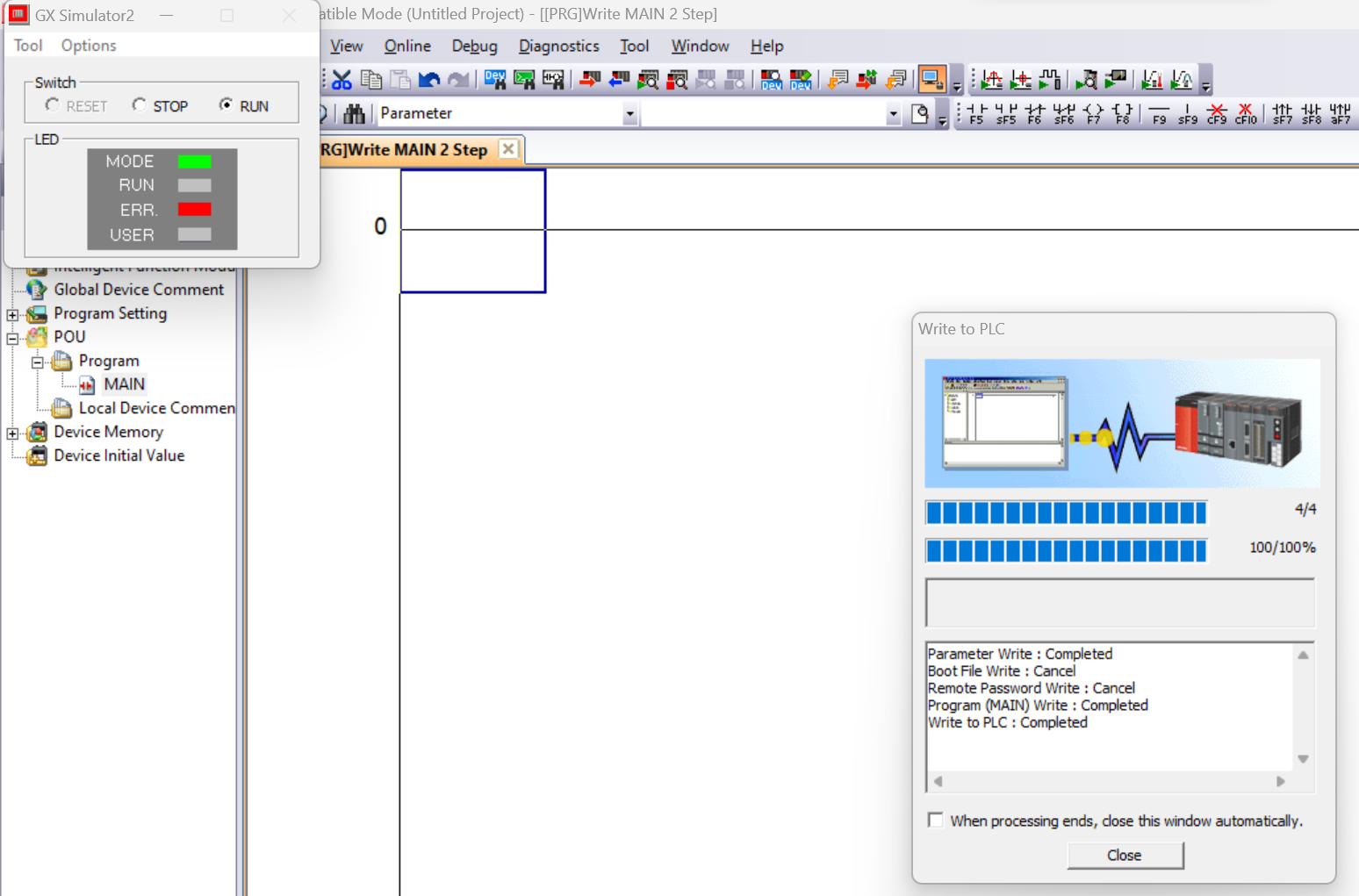


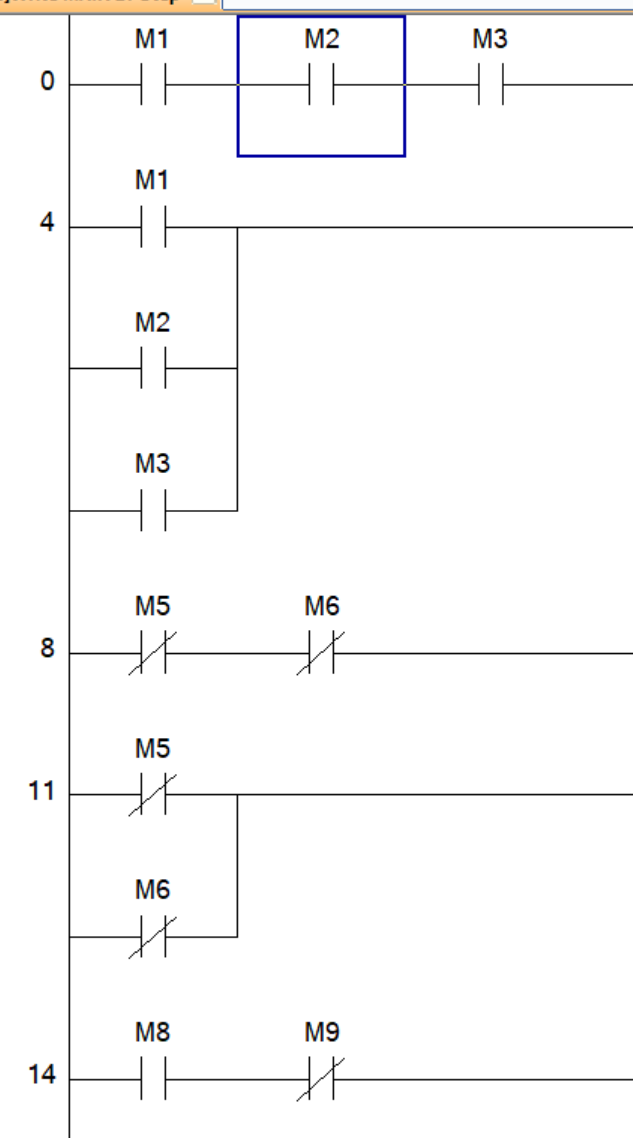


Deve-se inserir M0 e M1 (entrada e saída)



Após isso fazemos a simulação com o CLP, e com “Shift+Enter”, acionamos a entrada M0.





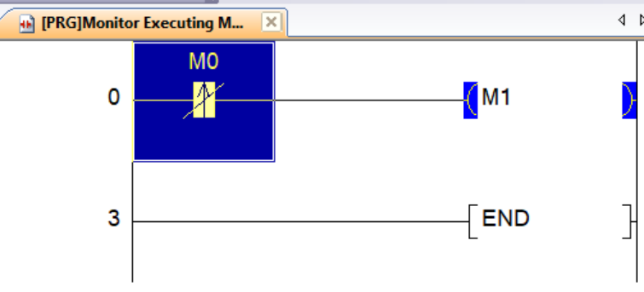
Não deve-se usar L com variável, ele só aceita a linha principal para ir até a saída.

SELO

28/09/2023

Existe uma área que não é considerada antes de iniciar a carga no processo.

Detecção por borda. 0,1 (FALSE/TRUE)

I----I ↑ I----I I----I ↓ I----I

Todo liga e desliga irá observar esse elemento acima. Gera-se os clocks (0101010100101). Lógica Inversa

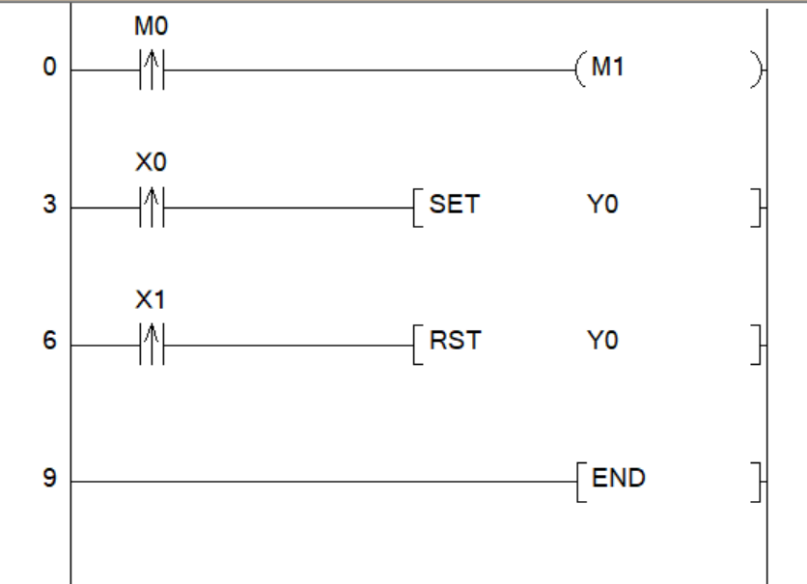
Borda de subida e borda de descida, indicam apenas um pulso.

Funções Set F7 fe Reset (tem prioridade)

I------I I---------------( )--------I

I----I I----I

Lógica do Selo



Kuka, Fanuki

A yaskawa é líder mundial no ramo da soldagem e pintura.

Epson nicho, com concorrentes fortes como da Howhe?, IDE Free, linguagem de programação estruturada com custo e benefício.

US -Robô colaborativo criado por um dupla do MIT, R Think.

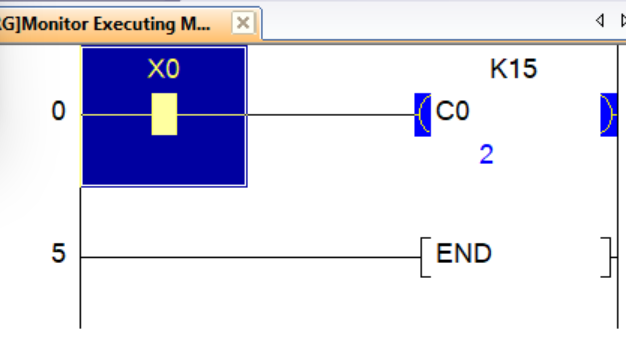
DESAFIO: LIGAR COM UMA ENTRADA E DESLIGAR COM A MESMA.

Timer (T)

Timers são infelizmente de pouca aplicação, pois o tempo de um processo varia muito.

TON; TOFF; TONOFF; TP

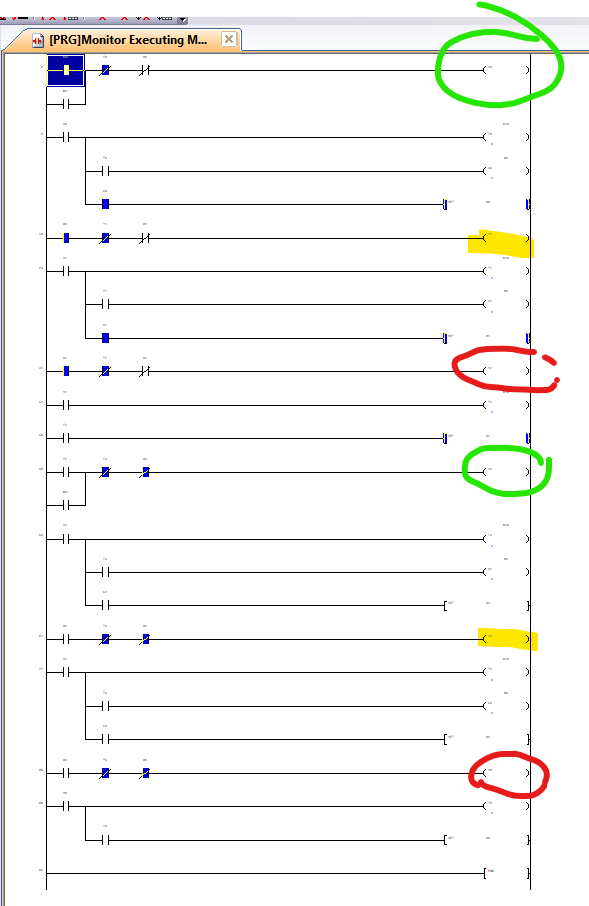
I----I I--------(T0 K100)----I



\*PROPOSTA DESAFIO –

29092023

Contador (Cowter)



I------------( )----I -> Simbologia do Coil, onde ficam o contador, o timer.

M [0~7999] – Bit (sinal digital Di/Do)

W [0~7999] – Word ( 16 bits/2 bytes/ 4 nimbles ) -> Bibliografia Caruano,

D [0~7999] – Double ( 32 bits/4 bytes/ 8 nimbles )

Byte 256

Base

* Binária;
* Octal;
* Decimal;
* Hexadecimal (16#FF;

Por exemplo, tendo 100, não se usaria uma Double pois ela pode armazenar até 4 bilhoes assim desperdiçando memória. A ideal para 100 seria o byte,

Interger (Inteiro)

Unsigner



0~6535

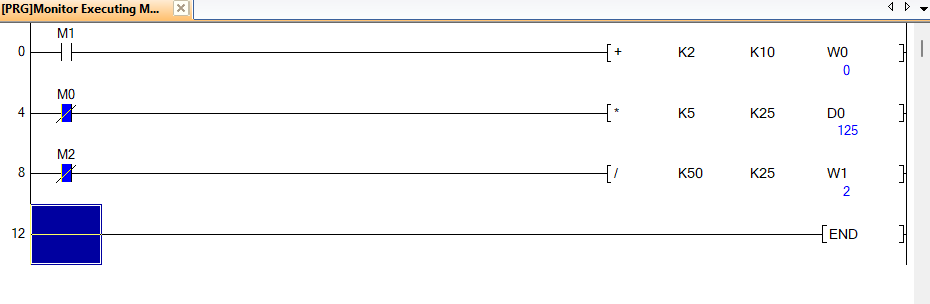
6535/2

+3267

-3268

I------------[ K10 e K2 ]----I

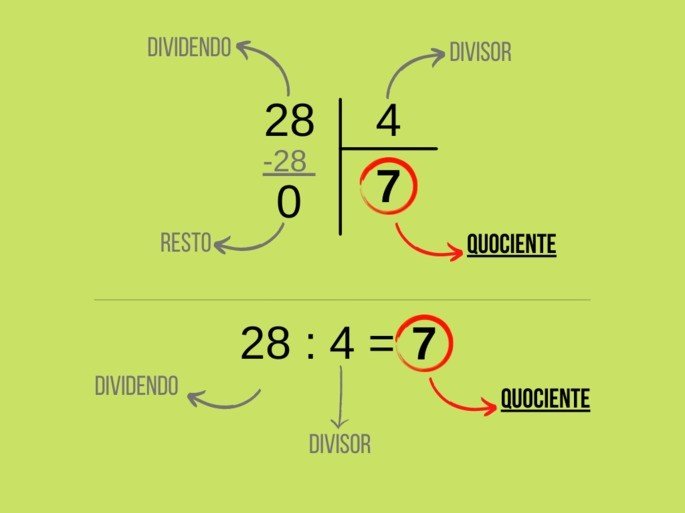
Somando K10 +K2, teríamos 12. Qual a melhor memória para usar? Uma Word.

I------------[ + K2 K10 W0 ]----I

I------------[ - Kxxx Kxxx Wxxx ]----I

I------------[ / Kxxx Kxxx Wxxx ]----I





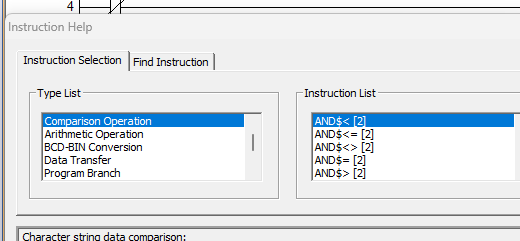
O número dividido por 2, caso tenha resto é Impar.

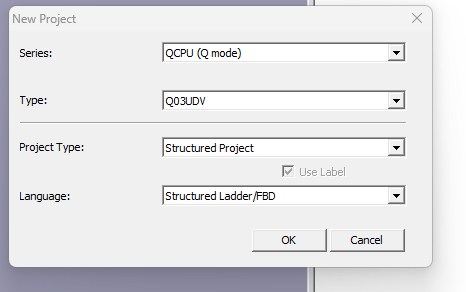
Mod (%)

MOV -> [MOV D2 D3]

AND -> [AND = D3 K3]

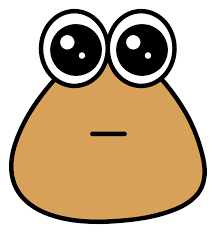
Se D3 é igual a K3, pode-se acionar algo por exemplo. Ainda pode-se umas >, <, >=, <=,<> (diferente);

Dando F8,

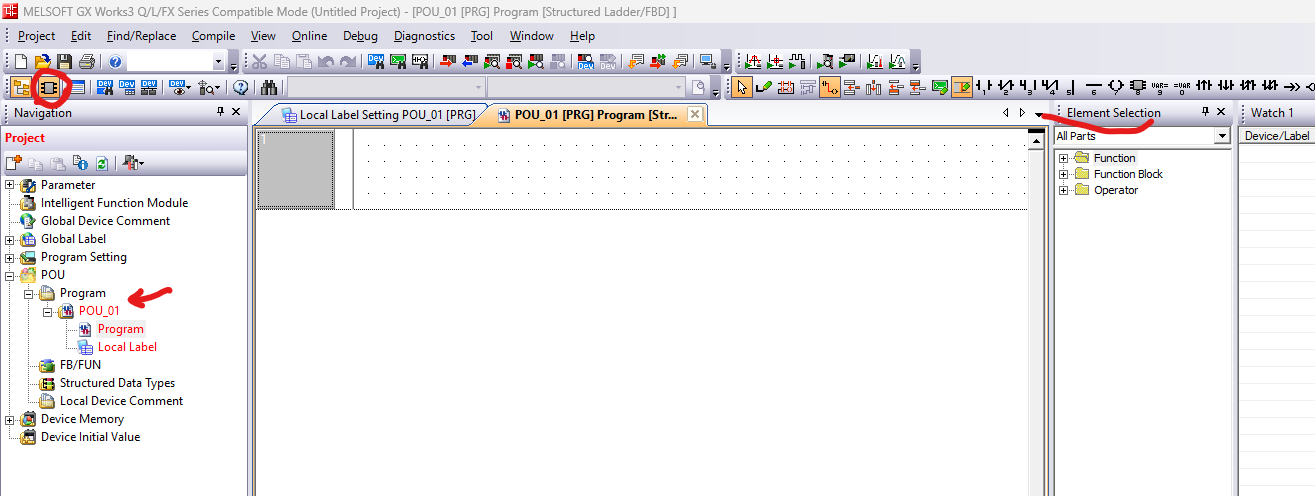
Desafio de hoje, números primos.

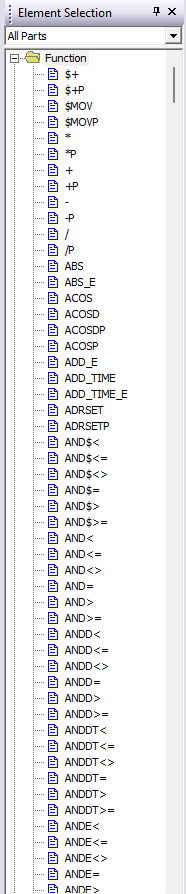
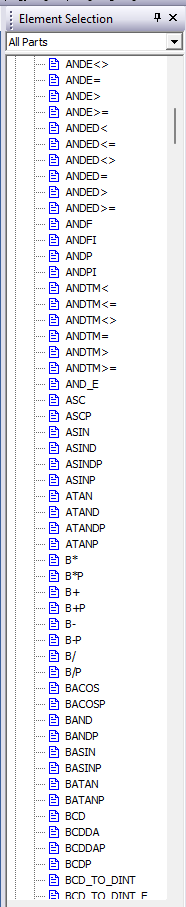
Dividido por 1, por ele mesmo. Não é divisível por 0, por numero negativo.

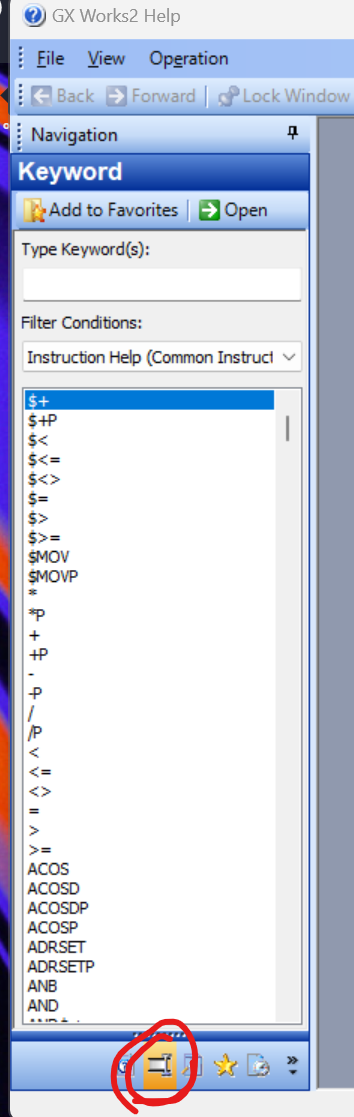
D=>=1

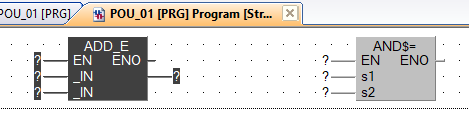
Fazer um contador, mas toda vez que incrementar de 1, deve-se dar uma saída caso por primo ou não. Botar pulsadores, Shit AND

POU (program Organization Unit



Com uma FUN, pode-se criar uma função, porém só tem uma saída. Pode ser repetida. Function Blocks podem ser. Nestere (encadeado) Function block dentro de Function block

Com um projeto aberto, selecionando a tecla F11 abre uma janela de ajuda, e selecionando a segunda opção indicada na figura ao lado, podemos observar Keywords com instrução sobre as funções.

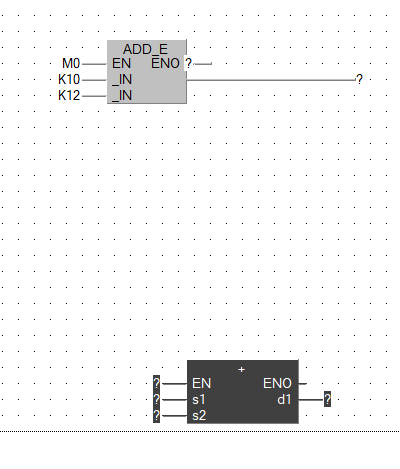


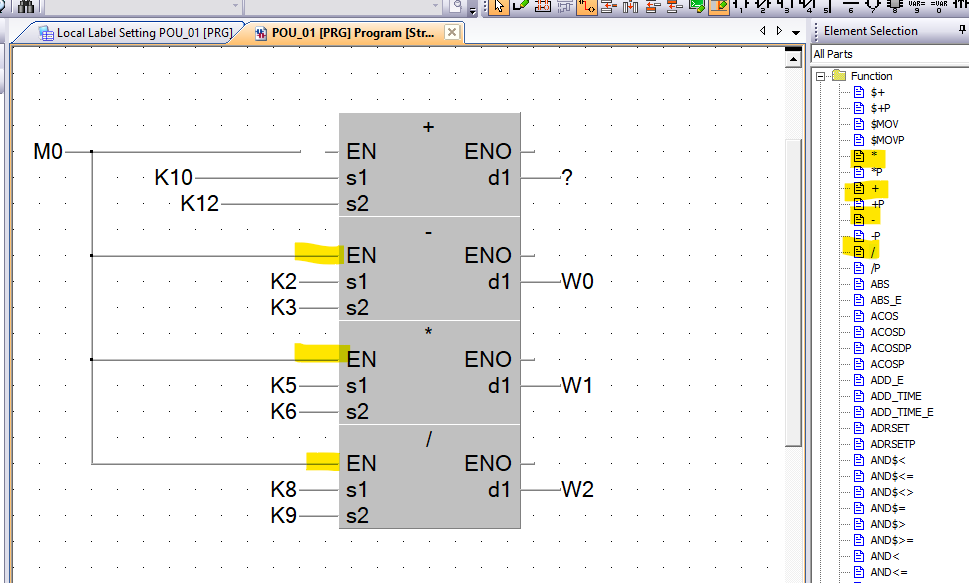
$ está relacionado a uma string

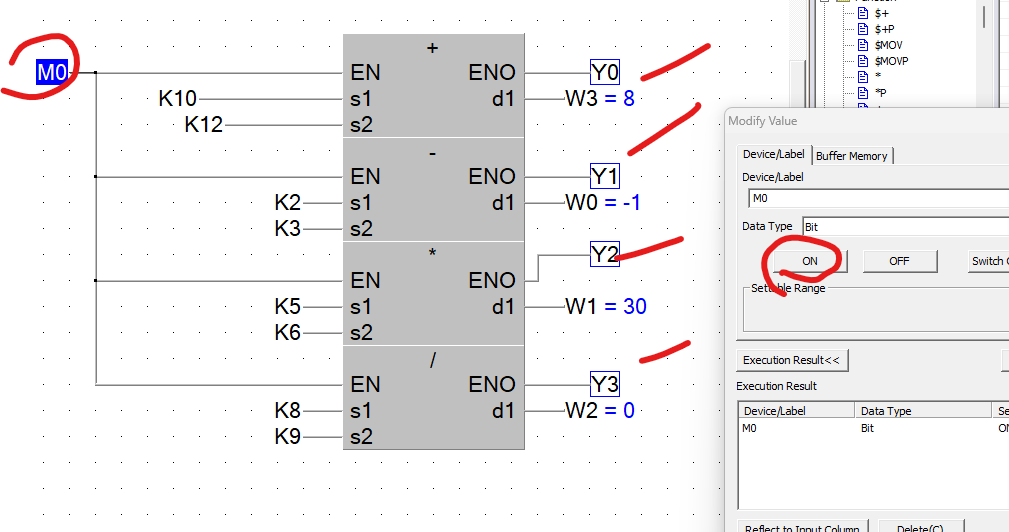
Qualquer função com um P no final, é um pulso, ele só ocorre uma vez na borda de subida.

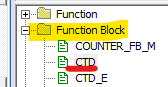
Imput e output Label, nas figura indicada abaixo.



 Com o cursor sobre a entrada, por exemplo, clique uma vez e puxe até a entrada M0.

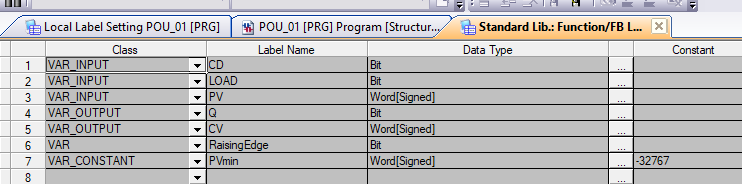




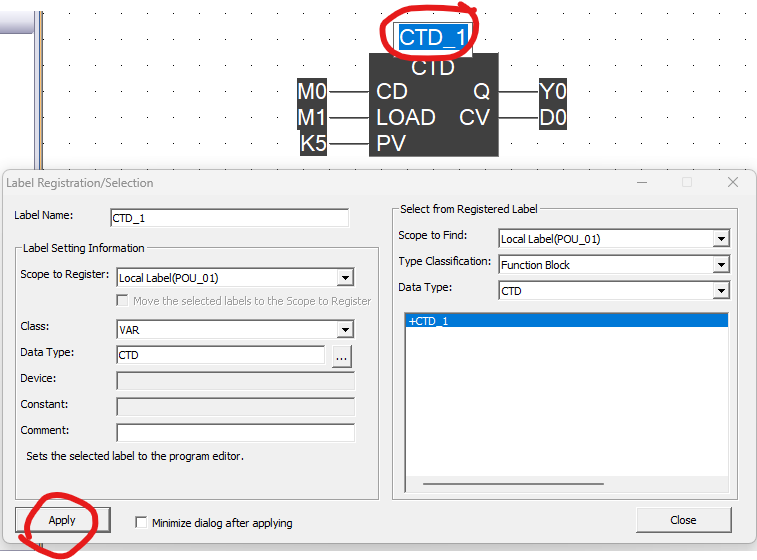
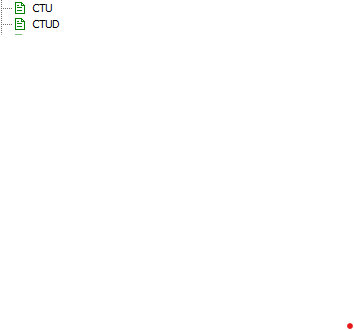
Para Function Blocks deve-se nomeá-las.

Down Counter (Contador decrescente)

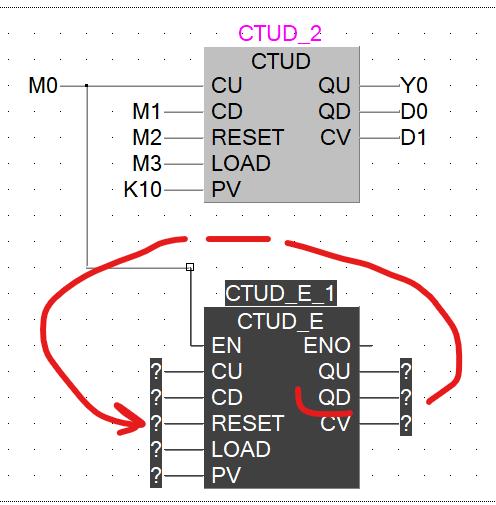
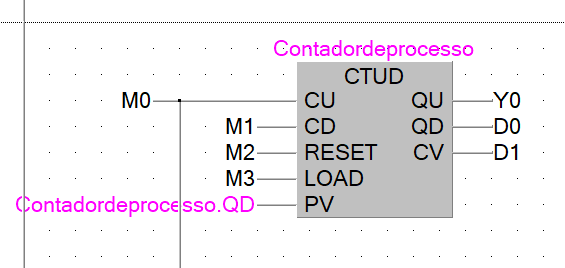
Com clique duplo se acessa a Aba Standard.

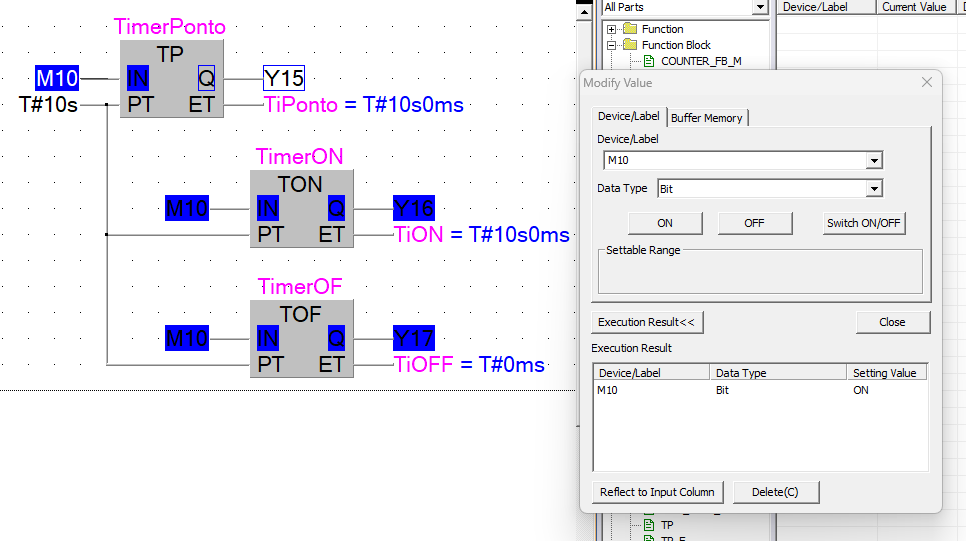
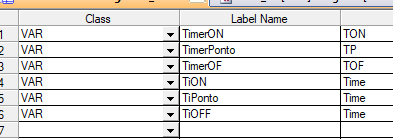


Lembra-se que precisamos nomear as function blocks. Pois é, por isso precisamos dar um clique duplo no nome, se for depois de inserido.

Estanciar, acessar o leque de variáveis que estão em outro Function Block, assim evita-se a necessidade de saber uma variável mas pegar o nome do Function Block como referencia.



Importante lembrar que as variáveis nas funções precisam estar consistentes, como por exemplo, a variável nome da function block não pode ser a mesma usada na saída, já que são tipos diferentes.

03/10/2023

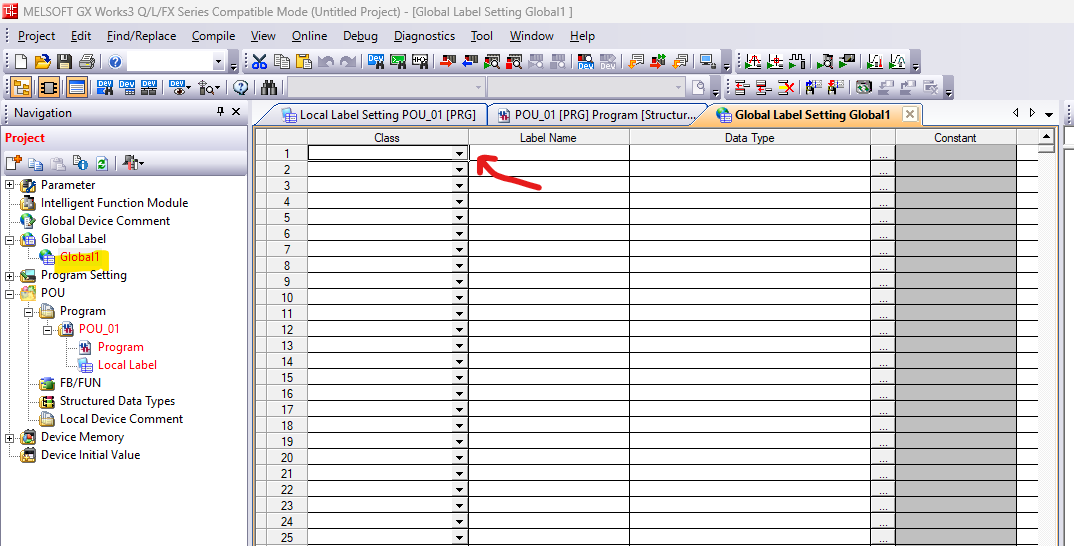
PLCopen – coding guidelines version 1.0 pg 18

Nome de Variáveis

|  |  |
| --- | --- |
| Imput | Output |
| Lx var name | A x var name |
|  |  |
|  |  |
| Global |  |
| ‘g\_xVarGlobal |  |
| ‘g\_ConstvarGlobal |  |
|  |  |

|  |  |
| --- | --- |
| Data type | Prefixo |
| Bool | X |
| Sint (Short Interger – metade de um inteiro) Byte | Si |
| Int | i |
| DInt | di |
| Lint | Li |
| String | STG |
| Struct | ST |
| Function | f |
| Function Block | Fb |

Constante: Toda ela é maiúscula.



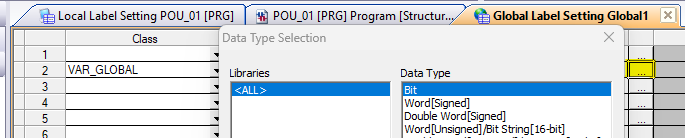
Novamente abrindo um projeto do tipo Structured, na aba global indicada acima, selecionamos a classe.

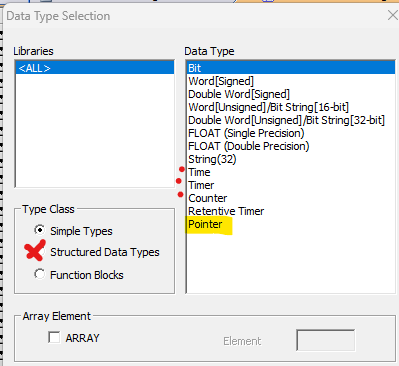
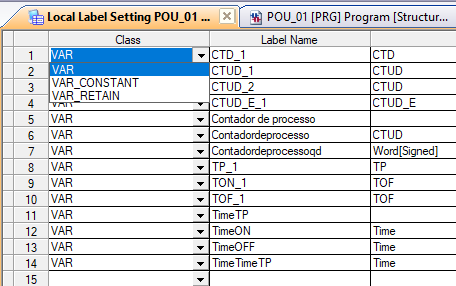
Cada caractere indicado em um passo computacional é um bite. Utilizar a tabela ask.

A modbus não trabalha com caractere. Para converter deve-se usar um hexadecimal, apoiando-se num vetor de 2 bites por exemplo, aponta-se para a modbus e ele irá receber em hexadecimal. Caso queria fazer string deve-se comprar na tabela, no final da função tem-se a string. Por fim deve-se contatenar. Se for trabalhar com string deve-seusar uma outra plataforma.

Como diferenciar o timer do time. O timer não deixa de ser um tipo de dado, mas no caso para temporizador, no caso em um function block. T On e T On estudado anteriormente. O Time no caso é um dado que precisa ser especificado como: Time (T#10s)

Para chegar na janela Data Type Selection clicar no quadrado marcado de amarelo.

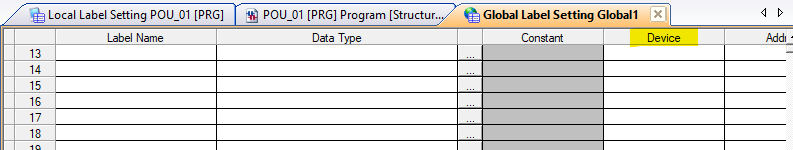


Var Constant

Var Retain: Se mantem com um valor liga e desliga.

* Array

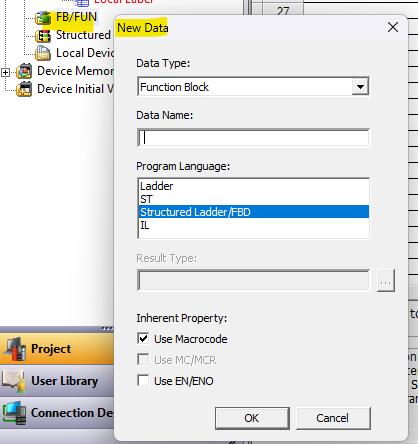


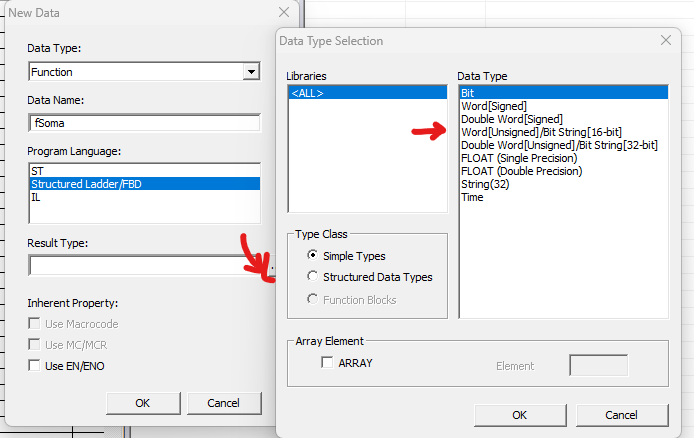
O Device é a forma como se vê no programa enquanto Adress seria o que o programa vê

Denote a forma como foi exemplificada.

Porém as variáveis Globais não são interessantes pois desorganizam o código do programa.

 Clique com o botão direito em FB/FUN, depois em Add New Data.

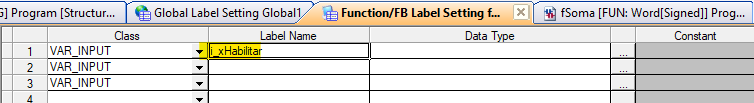




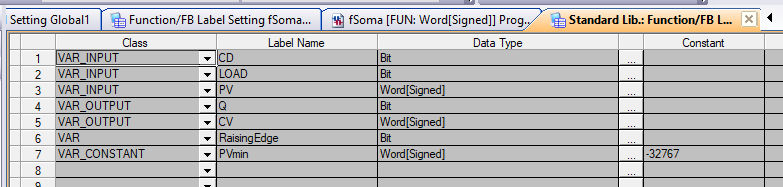
Selecione conforme indicado acima. Observe pouco acima do OK da janela a esquerda, nela podemos escolher o “EM/ENO”. Porém por ora não a marcamos.

Enable (Habilitar) é a primeira que se colocar.

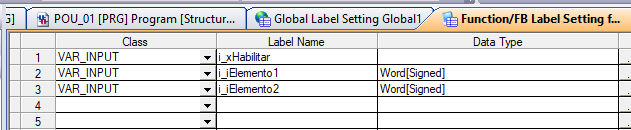
A primeira letra que especifica a nova palavra é maiúscula. Note o exemplo, abaixo.



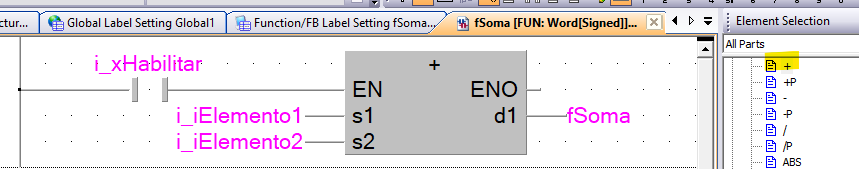
Clicando-se duas vezes em alguma das “entrada” das function block somos levados a Standard

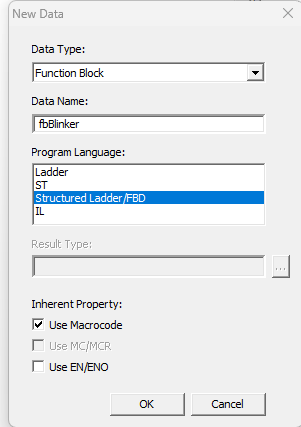


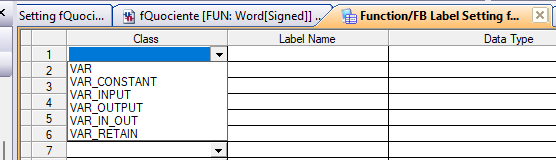
Está se fazendo a função para somar 2 elementos. A função deve ser algo “general “ (generalizado).



Denote que Word (Signed) são variáveis numéricas, enquanto que Unsigned são numéricas que funcionam como o valor absoluto, “sem zero”?



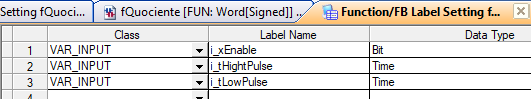
Agora criando uma Function Block

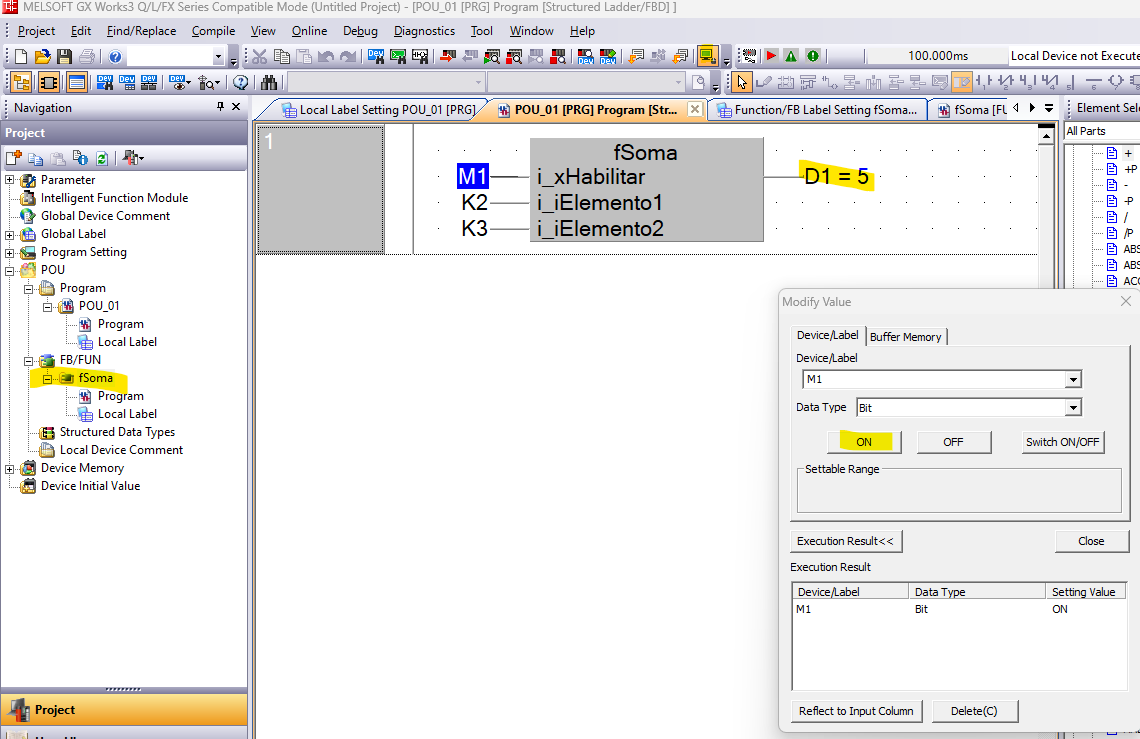


A variável de entrar em Var constante, ela não é variável.

Function Blocks não precisam ter saídas, controlar algo mas sem precisar de uma saída. Por isso, o VaRINOut que poderia ser “variável referência”. Ela seria devolvida no mesmo ponto, mesma posição, sem uma saída (para quem trabalhou com C, C++).

Incremento de variável. O que é mais fácil, procurar uma ou fazer uma? Standard Library é indicado, porém ficar procurando funções, quando se poderia fazer uma com sua personalização própria é mais prático.

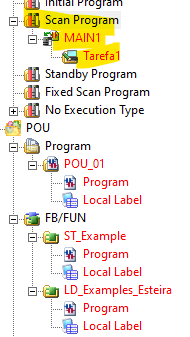
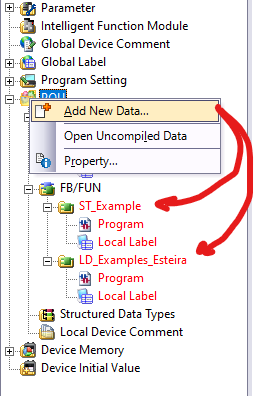




Desafio 4 operações, e os semáforos em uma function block de mesma lógica.i

04/10/2023

Texto estruturado.



Deve-se criar em Scan Program a Main e a Task que são Program Files. Inserindo suas funções lá, arrastando-as, haverá uma ordem de prioridade.

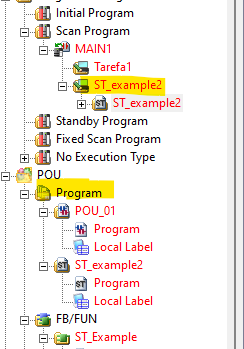
É mais prático de dar manutenção, como por exemplo caso ocorra um bug e assim ser necessário apenas apagar um dos programas.

Criar vários programas, também auxiliam para executá-las simultaneamente.

O Initial program, por exemplo, é rodado apenas uma vez quando o CLP é iniciado.

O Scan program são programas padrões, que ciclam normalmente.

Standy program utilizado em questões condicionais.

Fixed Scan Program já é utilizado para níveis críticos.A informação selecionada será fixada em programas de leitura fixa com intervalo de varredura de 1 segunda, o que pode ser alterado nos parâmetros do programa. Independente do que estiver dentro, ele tentará processar nesse período estipulado. Enquanto que no Standy program esse período é variado. Por isso, caso queria ter esse tempo específico, sem problemáticas, não utilizar no Fixed Scan Program. Porém, em um caso de programas para robôs de alta velocidade por exemplo, esse é o indicado.

Essa ideia de ordenar os programas, é uma ideia interessante para visualização didática, por mais que o processamento seja tão rápido que mal se percebe. Porém o interessante é utilizar apenas uma Main e fazer uso das programações de estado, o caso usado pelo Diego Machado.

O texto estruturado é autoexplicativo, baseado em Pascal

Como pode-se ver ao lado, devemos em “Programa” adicional um novo Program File porém com ST no tipo de linguagem (Structured)

Asign (variável de apontamento) “iDR:”

iDR: int=100;

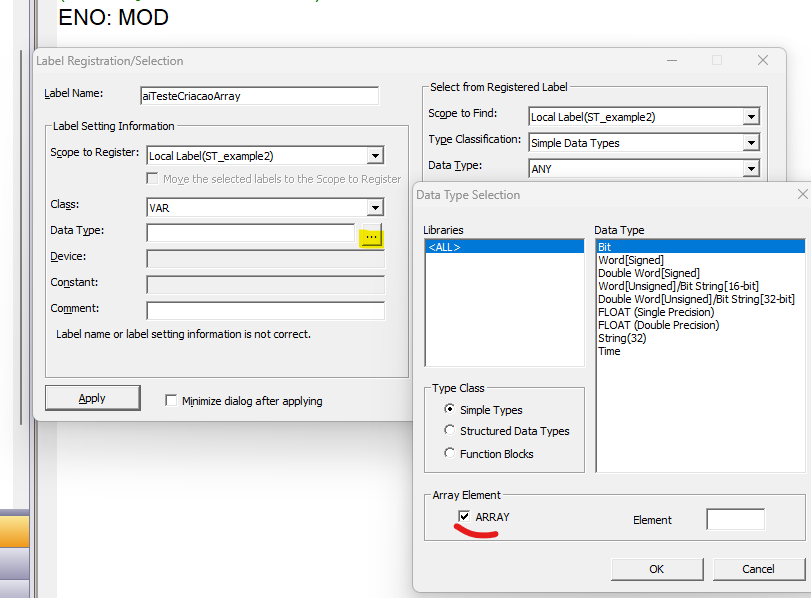
(\* ... \*) assim que se acrescenta comentário no GX Works em linguagem estruturada.

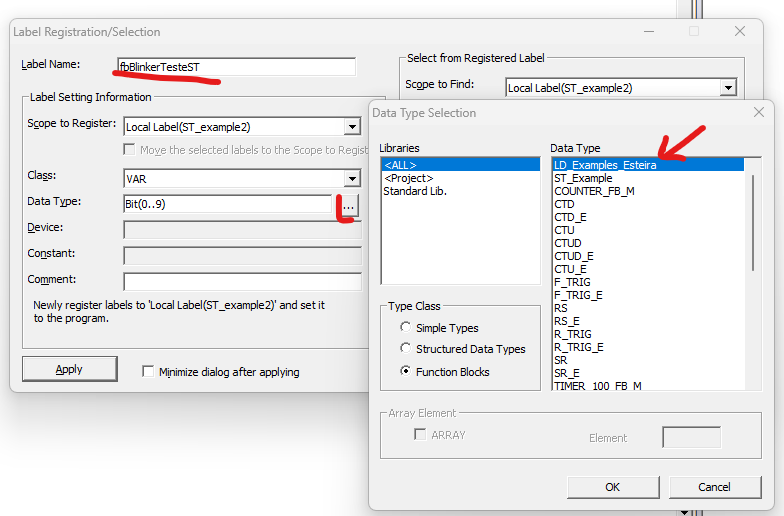
Na divisão dessa linguagem

iDR x): Variavel MOD variável 2

Utilizando o Array, para 5 elementos, será de 0-4 pois é contado o zero.

Dentro do programa estruturado, selecionando F2:



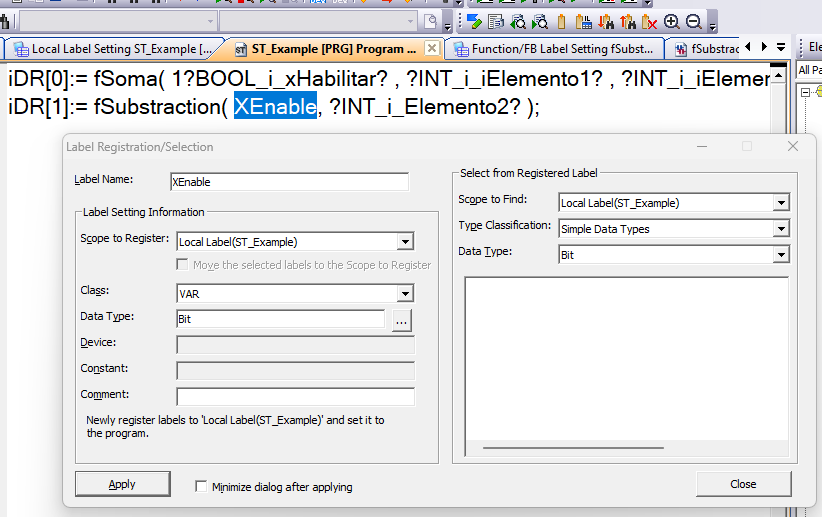


Agora sobre essa função, clicando ContrF1, são apontados os argumentos (imputs, outputs da sua função)

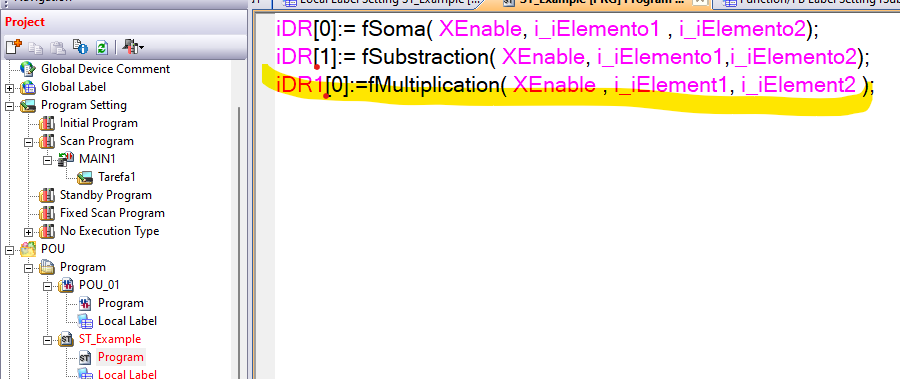
Na Declaração de variáveis, sempre que entrada ou saída, começamos com i\_ ou o\_

Exercicio: 4 funções em estruturado, +,-,\* e MOD, só que em function block

Ao abrir a função ST (structured), chamando funções que criamos, utilizando o atalho “Ctr+F1”

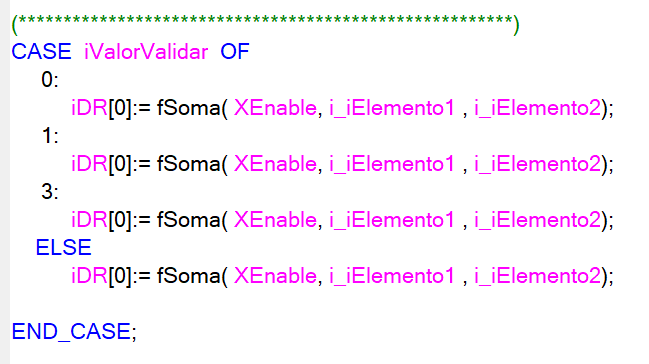


Notar que na multiplicação e divisão, utilizamos uma double world, por isso criar outro iDR



IF, utiliza-se uma condição seja ela Booleana (Se essa luz estive ligada, pois é no liga/desliga True/false). Novamente, se a velocidade do carro>=X, faz se algo, senão aquilo.

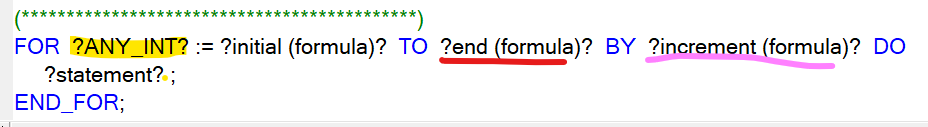
Funções de condição IF, CASE,

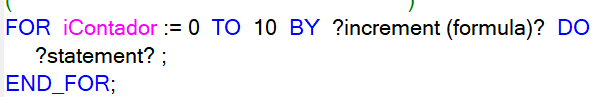


Case só utiliza valores inteiros.

É possível considerar os valores 0, 1,3 para apenas uma condição.

Função de repetição: FOR





O Incremento é o passo.

IF; IFS (CASE) – para valores numéricos

IF Condição booleana THEN

Operação\_If True;

ELSE IF Condição booleana THEN

Operação\_ELSEIF\_1;

ELSE IF Condição booleana THEN

Operação\_ELSEIF\_2;

ELSE

Operação\_IfFalse;

Não é obrigatório

Máquinas de estado, decisões, nomeação de estado.

While (enquanto)

“WHILE TRUE DO” tornaria uma repetição eterna.

05/10/2023

Estrutura condicional

IF (SE) // CASE (CASO)

IF Condição booleana THEN

Operação\_If True;

ELSE

Operação\_IfFalse;

Não é obrigatório

IF (CONDIÇÃO BOOL

IF XO THEN

YO=TRUE;

ELSE

YO:=FALSE;

Nota: Toda condição é concluída utilizando:

END\_ Condição(CASE, IF etc).

CASE iVarINT OF

0: Condição\_zero

Fb\_Ladder();

1: Condição\_um

* Estrutura de Repetição:

Os CLPs tem uma função chamada Watchdog, caso saia do modo automático ele perde a ciclicidade, auxiliando a testar as estruturas de repetição. Como por exemplo usando um While não finalizando é alarmado um Watchdog.

FOR iCounter:=0 TO 10 BY CONDIÇÃO

END FOR

Para sair facilmente de uma função, pode-se usar Brake que no caso aqui é nomeado Exist,

Uma forma de sair do While é trocando o true,.

Repeat aula PLC >10/10

IF Condição booleana THEN

Operação\_If True;

ELSE IF Condição booleana THEN

Operação\_ELSEIF\_1;

ELSE IF Condição booleana THEN

Operação\_ELSEIF\_2;

ELSE

Operação\_IfFalse;

Não é obrigatório

IF(se)THEN

FUN

4 IMPUTS

Habilitar (Bool);

1°Valor (int);

2°Valor (int);

Operador (String-> “C” DR;

+; - ; \* ; / ; Mod

Output

1 resultado inteiro

IF Condição booleana THEN

Operação\_If True;

ELSE IF Condição booleana THEN

Operação\_ELSEIF\_1;

ELSE IF Condição booleana THEN

Operação\_ELSEIF\_2;

ELSE

Operação\_IfFalse;

Não é obrigatório

CASO(caso)OF

FUN

4 IMPUTS

Habilitar (Bool);

1°Valor (int);

2°Valor (int);

Operador (String-> “C” DR;

+; - ; \* ; / ; Mod

Output

1 resultado inteiro

