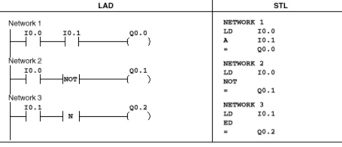
**FX3U14MR PLC**

**GX Works2**

კოდის მაგალითი (LADDER ან STL სტილში):

**🔌 ძირითადი ლოგიკური სიმბოლოები**

| **სიმბოლო** | **აღწერა** | **მნიშვნელობა** |
| --- | --- | --- |
| `-- |  | --` |
| `-- | / | --` |
| --( )-- | Coil (OUT) | გამშვები მარყუჟი — ჩართავს ან თიშავს მოწყობილობას |
| --(SET)-- | Set Coil | აყენებს მოწყობილობას მუდმივად ON მდგომარეობაში |
| --(RST)-- | Reset Coil | თიშავს SET-ით ჩართულ მოწყობილობას |
| --[ ]-- | Box Function | ფუნქციური ბლოკი, როგორიცაა ტაიმერი, ქაუნთერი, ინსტრუქციები |

**⏱️ ტაიმერები და ქაუნთერები**

| **სიმბოლო** | **აღწერა** |
| --- | --- |
| Txxx | ტაიმერის ნომერი (მაგ: T0, T1...) |
| Cxxx | ქაუნთერის ნომერი (მაგ: C0, C1...) |
| --[TON]-- | Time ON Delay |
| --[TOF]-- | Time OFF Delay |
| --[CTU]-- | Count Up |
| --[CTD]-- | Count Down |

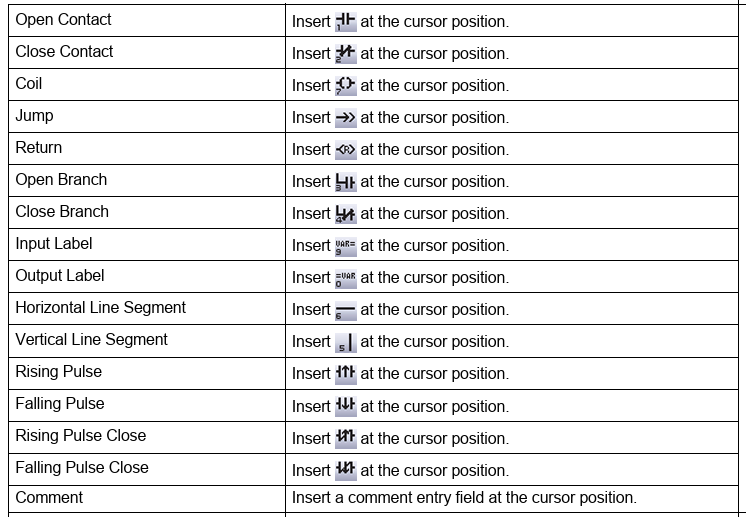
**🔢 მონაცემთა მენეჯმენტი**

| **სიმბოლო** | **აღწერა** |
| --- | --- |
| MOV | მონაცემის გადაადგილება (Move) |
| ADD, SUB, MUL, DIV | არითმეტიკული ოპერაციები (დამატება, გამოკლება, გამრავლება, გაყოფა) |
| CMP | მონაცემთა შედარება |
| AND, OR, XOR, NOT | ლოგიკური ოპერაციები |

**💾 მოწყობილობის ტიპები**

| **მოწყობილობა** | **აღწერა** |
| --- | --- |
| X | შესასვლელები (inputs) — ფიზიკური ღილაკები, სენსორები |
| Y | გამოსასვლელები (outputs) — ნათურები, ძრავები |
| M | შიდა მარკერები (internal relays) (ლოგიკური სიგნალი) |
| D | მონაცემთა რეგისტრები (data registers) |
| T | ტაიმერები |
| C | ქაუნთერები |
| S | ნაბიჯები (Steps) — STEP LADDER პროცესისთვის |
| K | მუდმივი მნიშვნელობა (მაგ: K10 ნიშნავს რიცხვ 10) |
| H | ჰექსადეციმალური მუდმივები |
| Z | ინდექსური რეგისტრები |

LADDER პროგრამირებაში გამოყენებული სიმბოლოები



|  |  |
| --- | --- |
| Open Contact  Close Contact  Coil  Jump  Return  Open Branch  Close Branch  Input Label  Output Label  Horizontal Line Segment  Vertical Line Segment  Rising Pulse  Falling Pulse  Rising Pulse Close  Falling Pulse Close  Comment | კონტაქტის გახსნა  კონტაქტის დახურვა  სპირალი  ნახტომი  დაბრუნება  განშტოების გახსნა  განშტოების დახურვა  შეყვანის ეტიკეტი  გამომავალი ეტიკეტი  ჰორიზონტალური ხაზის სეგმენტი  ვერტიკალური ხაზის სეგმენტი  აღმავალი იმპულსი  დაღმავალი იმპულსი  აღმავალი იმპულსის დახურვა  დაღმავალი იმპულსის დახურვა  კომენტარი |

**📈 Rising Pulse - როგორ მუშაობს**

--|↑|-- ← ეს არის **Rising Edge Detection Contact**

როდესაც შეყვანა (მაგ. ღილაკი) გადადის **0-დან 1-ზე** (ანუ გამორთულიდან ჩართულზე), **Rising Pulse** ქმნის **მხოლოდ ერთ სკან ციკლის განმავლობაში აქტიურ იმპულსს**.

რეგულარულ --||-- კონტაქტს თუ იყენებ, ის იქნება ON მანამ, სანამ ღილაკი დაჭერილია. მაგრამ ხშირად გვჭირდება, რომ ლოგიკა განხორციელდეს **მხოლოდ ღილაკის დაჭერის წამში**, ერთხელ - აი, აქ გვეხმარება Rising Pulse.

**ფუნქციონალურად:**

* აქტიურდება მხოლოდ მაშინ, როცა **შემომავალი მოწყობილობა (მაგ. X0) გადადის OFF-დან ON-ზე**
* მხოლოდ ერთ ციკლად (scan) რჩება აქტიური

🧪 მაგალითი Ladder ლოგიკაში:

--|↑|-- --( )--

X0 M0

* X0 — ფიზიკური ღილაკი
* M0 — შიდა მარკერი, რომელიც ჩაირთვება მხოლოდ მაშინ, როცა X0 პირველად გადავა OFF-დან ON-ზე

**შედეგი**: M0 **მხოლოდ ერთხელ ჩაირთვება**, როდესაც ღილაკს X0 პირველად დააჭერთ. შემდეგი სკან ციკლში აღარ იქნება ON, სანამ ისევ არ მოხდება X0-ის გათიშვა და ხელახალი ჩართვა.

**M8000 — Always ON Flag**

ეს არის სპეციალური სისტემური ბიტი Mitsubishi PLC-ში, რომელიც მუდმივად ჩართულია (True). ჩვეულებრივად გამოიყენება ინიციალიზაციის კოდში, რათა რაღაც ბრძანებები შესრულდეს მუდმივად ან სისტემის ჩართვისას ერთხელ.

როგორც კი PLC გადადის RUN რეჟიმში, M8000 ხდება ON (1). ის მუდამ რჩება ON-ად, სანამ PLC მუშაობს.

თუ PLC გადადის STOP რეჟიმში, M8000 ხდება OFF (0).

* M8000 (Always ON)
* M8001 (Always OFF)
* M8033 (1-წამიანი პულსი)

**M8002 აქტიურდება მხოლოდ ერთჯერადად, პირველ სკანზე** — ანუ მაშინ, როცა PLC გადადის "RUN" რეჟიმში.

M8002 არ იქნება აქტიური ციკლურად — მხოლოდ ერთხელ, თავიდან.

ხშირად გამოიყენება სხვა სპეციალურ მარკერებთან ერთად.

**M8011 — 10ms Clock Pulse**

ეს სპეციალური მერყევი ბიტია Mitsubishi PLC-ში, რომელიც თითოეულ 10 მილიწამში ერთხელ აქტიურდება (ON) ერთი ციკლის განმავლობაში. გამოიყენება იმ კოდისთვის, რომელიც უნდა შესრულდეს პერიოდულად, მაგრამ არა ყოველ ციკლზე (CPU-ს დასაცავად ზედმეტი გამოთვლებისგან).

**M8012 - 100 ms clock pulse ON and OFF in 100 ms cycles (ON: 50 ms, OFF: 50 ms)**

**MOV --- K0 D0**

* MOV – ეს არის "Move" ინსტრუქცია, რომელიც აკოპირებს მნიშვნელობას ერთი ადგილიდან მეორეში.
* K0 – ეს არის მუდმივი მნიშვნელობა 0 (K არის კონსტანტა — Decimal, K10 = 10).
* D0 – ეს არის მონაცემთა რეგისტრი.

**MOV --- K0 D0** და **MOV --- K0 D1** ნიშნავს, რომ D0 და D1 მონაცემთა რეგისტრებში ჩაიწერება ნულოვანი მნიშვნელობა (0) ყოველ ციკლზე, რადგან M8000 ყოველთვის აქტიურია. ეს ჩვეულებრივ კეთდება ინიციალიზაციისთვის, ანუ სისტემის ჩატვირთვისას ან კონტროლერის გაშვებისას რეგისტრები (ან ალგორითმები) "სუფთა" ნულიდან დაიწყოს.

**RD3A --- K0 D0 D10**

**RD3A-** სპეციალური ინსტრუქცია, რომელიც გამოიყენება FX3U-AD (ანალოგური მოდულის) მონაცემების წასაკითხად.

K0 - არხი 0 (Channel 0, ანუ პირველი ანალოგური შესასვლელი).

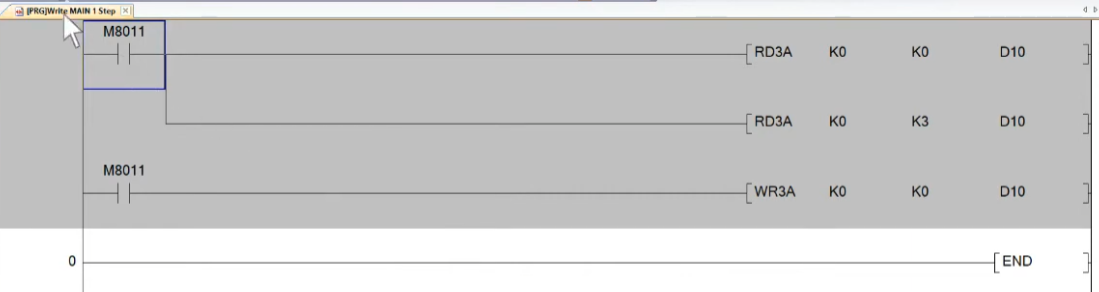
D0 - რეგისტრი, სადაც იწერება წაკითხული მნიშვნელობა (ან ADC input-raw).

D10 - რეგისტრი, სადაც ჩაიწერება დამუშავებული მნიშვნელობა (ზოგჯერ ეს არის კონვერტირებული მონაცემი ან პირდაპირ ADC შედეგი).



K0 -PLC ის რიცხვითი ერთეულია (მძიმის შემდეგ რამდენი რიცხვი გამოჩნდეს);

K3 - 3 ეს არის პინის ნომერი სადაც სენსორია შეერთებული (ათვლა იწყება Gnd მხრიდან და ითვლება 0 დან).

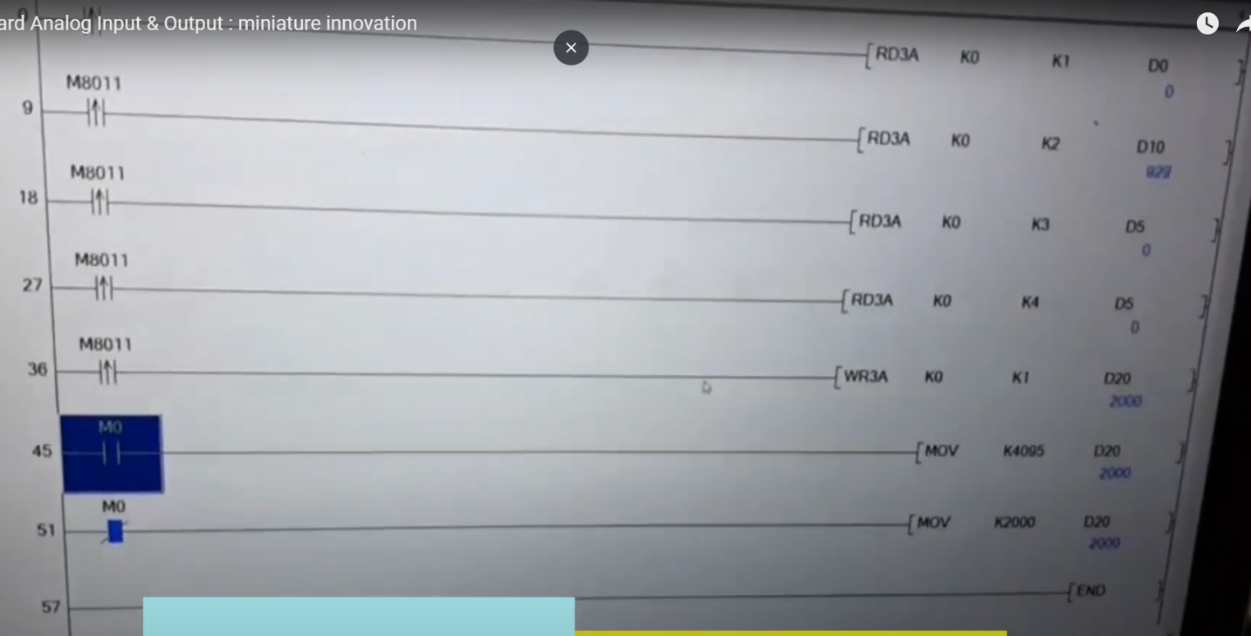


**WR3A --- K0 K0 D10**

WR3A - სპეციალური ინსტრუქცია, რომელიც გამოიყენება FX3U-ს ანალოგური გამოსასვლელზე ძაბვის გადასაცემად (0-10V).

K0 - PLC ის რიცხვითი ერთეულია (მძიმის შემდეგ რამდენი რიცხვი გამოჩნდეს);

K0 - 0 ეს არის პინის ნომერი სადაც ანალოგური სიგნალის მიმღებია შეერთებული (ათვლა იწყება Gnd მხრიდან და ითვლება 0 დან).

**ანალოგ მომღები და გადამცემი პინების ათვლა ცალ-ცალკე ხდება და იწყება 0-ით.**

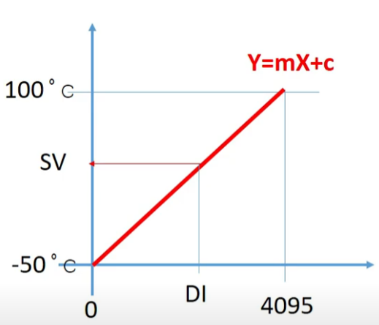
**M0 --- MOV K4095 D20**

ფოტოზე თუ მე M0-ში მაგალითად HMI დან ჩავწერ 1-ს ანუ ჩავრთავ, 4059 რიცხვი რომელიც 10 ვოლტის პროპორციულია ჩაიწერება D20 რეგისტრში რომელიც მის ზედა სტრიქონშია კონფიგურირებული. ხოლო მის ქვედა სტრიქონში წერია, რომ როცა M0 არაა დაჭერილი D20 ში იწერება 2000 ანუ 5 ვოლტი გამოდის.

**ანალოგური სიგნალის წაკითხვა (Analog 0-10v და 0-20ma)**

* **ADC Range**: 0–4095(ეს შეესაბამება 0–10V-ს ან 0-20ma)
* **Supply Voltage 10V:** 5V ან 10V ანალოგური მოდულის შესაბამისად;
* **Supply Voltage 20Ma:** 5V ან 24V ანალოგური მოდულის შესაბამისად;

**ფორმულა:**

**SV=[(SH-SL)/DH-DL]x(DI-DL)+SL**

SV: მიმდინარე ტემპერატურა, რომელსაც ვიკვლევთ;

SH: სენსორის მაქსიმალური გაზომვის დიაპაზონი (NTC= -50 <> 110)

SL: სენსორის მინიმალური გაზომვის დიაპაზონი (NTC= -50 <> 110)

DH: მაქსიმალური რიცხვი ანალოგურ შესასვლელზე რომელსაც კითხულობს PLC (0 <>4059);

DL: მინიმალური რიცხვი ანალოგურ შესასვლელზე რომელსაც კითხულობს PLC (0 <>4059);

DI: კონკრეტულ მონაკვეთშ PLC ანალოგურ შესასვლელზე შემოსული რიცხვი, მაგ. **1223.**

**SV=[160/4059]x1223+(-50)**

**LADDER პროგრამა NTC 10K სთვის:**

--|M8011|----- [RD3A K0 K0 D0] >> მაგ. D0 = 1223

|----- [FLT D0 D2] >> 1223 -> 1223,000

----- [FLT D4 D6] >> 130 -> 130.000

----- [MOV K150 D4] >> D4 = 150

----- [FLT D8 D10] >> 4095 -> 4095.000

----- [MOV K4095 D8] >> D10 = 4095

----- [FLT D12 D14] >> -50 -> -50.000

----- [MOV K-50 D12] >> D12=-50

-- [ DEDIV D6 D10 D18] >> 130.000/4095.000=0.032 **(Double-word Division)** – 32-bit გაყოფა

-- [ DEMUL D18 D2 D20] >> 0.032x1223.000=39.136 **(**32-bit გამრავლება)

-- [ DEADD D20 D14 D25] >> 39.136 +(-50.000) =-10,877 **(**32-bit შეკრება)

----- [DINT D25 D30] >> -10,877 -> -11 >> D25 D100 თუ გვინდა PID-I

[> D30 K40 ] ------ [SET Y000] >> თუ D30 > 40 ჩაირთოს Y000 რელე

[<= D30 K40 ] ------ [RST Y000] >> თუ D30 <= 40 გამოირთოს Y000 რელე

-------------------------------------[END]

თუ NTC ტემპერატურის სენსორი მუშაობს 0–12V დიაპაზონზე, მაგრამ შენ მას აპარატურად ამუშავებ მხოლოდ 0–10V-ზე, მაშინ ტემპერატურის საზომ დიაპაზონში შეიზღუდება ზედა ნაწილი — ანუ:

 ტემპერატურის დიაპაზონი:  
**(+105) - (-50) = 155°C**

 ძაბვის დიაპაზონი:  
**0–12V**

12V / 155°C = 0.077 V/1°C

**რა ტემპერატურას შეესაბამება 10V?**

10V / 0.077 V/°C = 129.87°C

მაშინ 129.87°C - 50°C (რადგან -50°C ბაზის ტემპია) = 79.87°C ზემოდან

ანუ:

-50°C + 129.87°C = \*\*+79.87°C\*\*

შენი ADC თუ კითხულობს მხოლოდ 0–10V, მაშინ:

შეგიძლია გაზომო ტემპერატურა -50°C-დან მხოლოდ +79.87°C-მდე

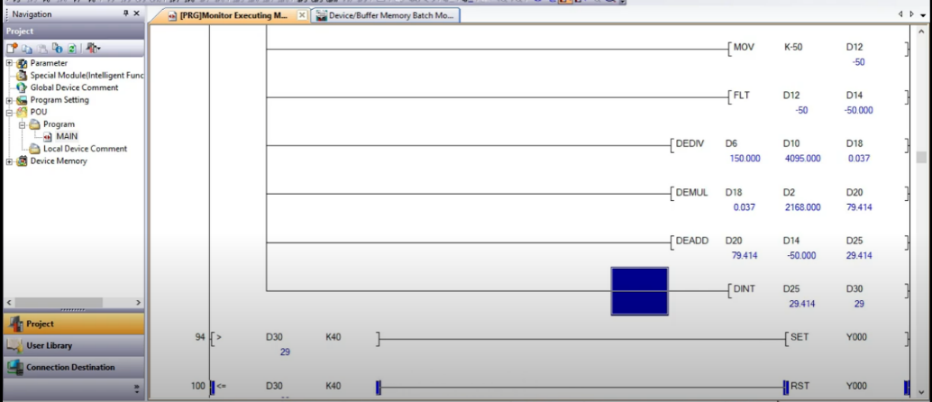
მსგავსად იქნება ანალოგ 0 და ანალოგ 3 შესასვლელებისთვისაც

ანალოგ 0 სთვის იქნება: 0 – 4000 შემავალი სიგნალის დიაპაზონი და 0 – 10 V ძაბვა.

10/4000=0,0025ვ

ანალოგ 3 სთვის იქნება: 0 – 2000 შემავალი სიგნალის დიაპაზონი და 0 – 20 mA დენი.

20/2000=0,01 mA



**LADDER პროგრამა 0 – 10 V სთვის:**

--|M8011|----- [RD3A K0 K0 D40] >> მაგ. D40 = 1223

|----- [FLT D40 D42] >> 1223 -> 1223,000

----- [FLT D44 D46] >> 10 -> 10.000

----- [MOV K10 D44] >> D44 = 10

----- [FLT D48 D50] >> 4000 -> 4000.000

----- [MOV K4000 D48] >> D48 = 4000

-- [ DEDIV D46 D50 D58] >> 10.000/4000.000=0.0025 **(Double-word Division)** – 32-bit გაყოფა

-- [ DEMUL D58 D42 D60] >> 0.002x1223.000=2.446 **(**32-bit გამრავლება)

----- [DINT D60 D62] >> 2.446 -> 2

-------------------------------------[END]

**LADDER პროგრამა 0 – 20 mA სთვის:**

--|M8011|----- [RD3A K0 K3 D70] >> მაგ. D70 = 1223

|----- [FLT D70 D72] >> 1223 -> 1223,000

----- [FLT D74 D76] >> 100 -> 100.000

----- [MOV K100 D74] >> D74 = 100

-- [ DEDIV D72 D76 D78] >> 1223,000/100.000=12.230 **(Double-word Division)** – 32-bit გაყოფა

----- [DINT D78 D80] >> 2.446 -> 2

-------------------------------------[END]

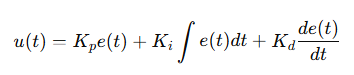
**PID**

**PID კონტროლის ძირითადი პარამეტრები:**

1. **KP (Proportional Gain) – პროპორციული გამაძლიერებელი**
   * განსაზღვრავს, რამდენად ძლიერ რეაგირებს სისტემა ცდომილებაზე (შეცდომა = SP - PV).
   * რაც უფრო მაღალია KP, მით სწრაფად რეაგირებს სისტემა, მაგრამ შესაძლოა შეიტანოს ცახცახი (oscillation).
2. **KI (Integral Gain) – ინტეგრალური გამაძლიერებელი**
   * პასუხისმგებელია *გრძელვადიანი ცდომილების გამოსწორებაზე*.
   * რაც უფრო მაღალია KI, მით უფრო სწრაფად მოიცილება მუდმივი ცდომილება, მაგრამ ასევე იზრდება სისტემის არასტაბილურობის რისკი.
3. **KD (Derivative Gain) – დერივაციული გამაძლიერებელი**
   * რეაგირებს შეცდომის სიჩქარეზე — ანუ "თვლის წინასწარ", თუ როგორ შეიცვლება პროცესი.
   * ამცირებს ცახცახს და აუმჯობესებს სტაბილურობას, მაგრამ ზედმეტად მაღალი KD ამატებს სენსორის შმენისადმი მგრძნობელობას.
4. **KE (Error Gain) ან Scaling Factor (ზოგ მოდელში)**

* ხშირად გამოიყენება Mitsubishi PLC-ში როგორც **ცდომილების გამაძლიერებელი ფაქტორი ან მასშტაბირების კოეფიციენტი**.

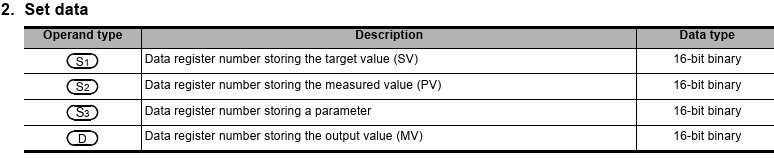
| **პარამეტრი** | **დანიშნულება** |
| --- | --- |
| **KP** | ზრდის რეაქციის სიმძლავრეს ცდომილებაზე |
| **KI** | ხელს უწყობს მუდმივი ცდომილების გაქრობას |
| **KD** | ამცირებს სწრაფ ცვლილებებს და ცახცახს |
| **KE** | ცდომილების მასშტაბირების ან ნორმალიზაციის პარამეტრი (განსაზღვრავს შიდა გაანგარიშების საფუძველს) |



* e(t) — დროის მომენტში არსებული შეცდომა (მაგ. მოთხოვნილი მნიშვნელობა - ფაქტიური მნიშვნელობა),
* KpK\_pKp​, KiK\_iKi​, KdK\_dKd​ — შესაბამისი გაძლიერების კოეფიციენტები.

**PID - PLC FX3U**

**პარამეტრები:**

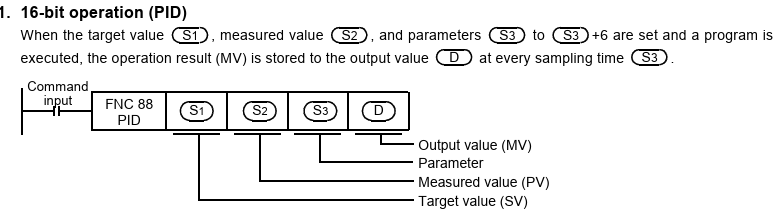


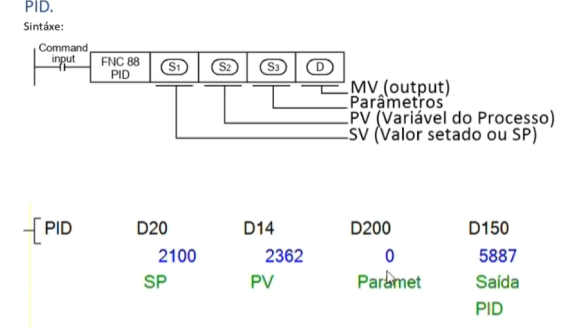
**S1** - სამიზნე მნიშვნელობის შემნახველი მონაცემთა რეგისტრის ნომერი (SV)გაზომილი;

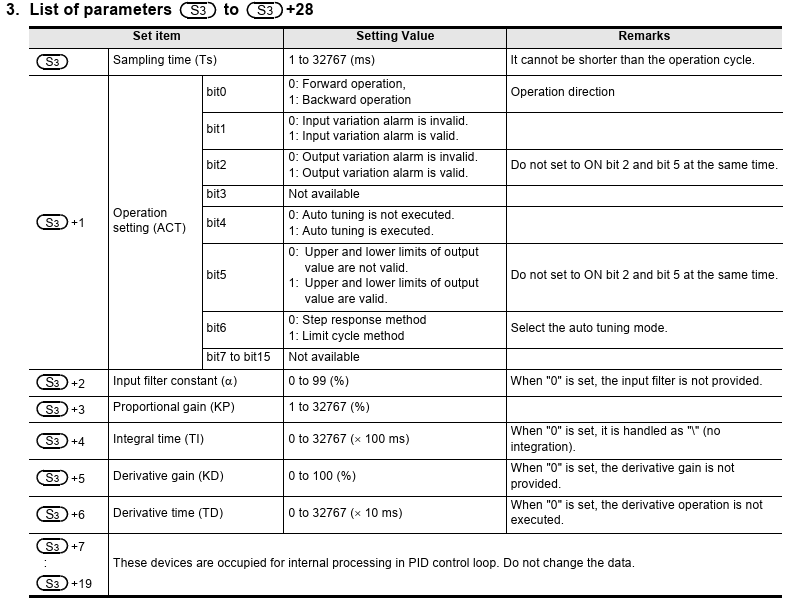
**S2** - მნიშვნელობის შემნახველი მონაცემთა რეგისტრის ნომერი (PV);

**S3** - პარამეტრის შემნახველი მონაცემთა რეგისტრის ნომერი;

**D** - გამომავალი მნიშვნელობის შემნახველი მონაცემთა რეგისტრის ნომერი (MV).



სამიზნე მნიშვნელობა S1, გაზომილი მნიშვნელობა S2, პარამეტრები S3 (S3 +6), ოპერაციის შედეგი (MV).

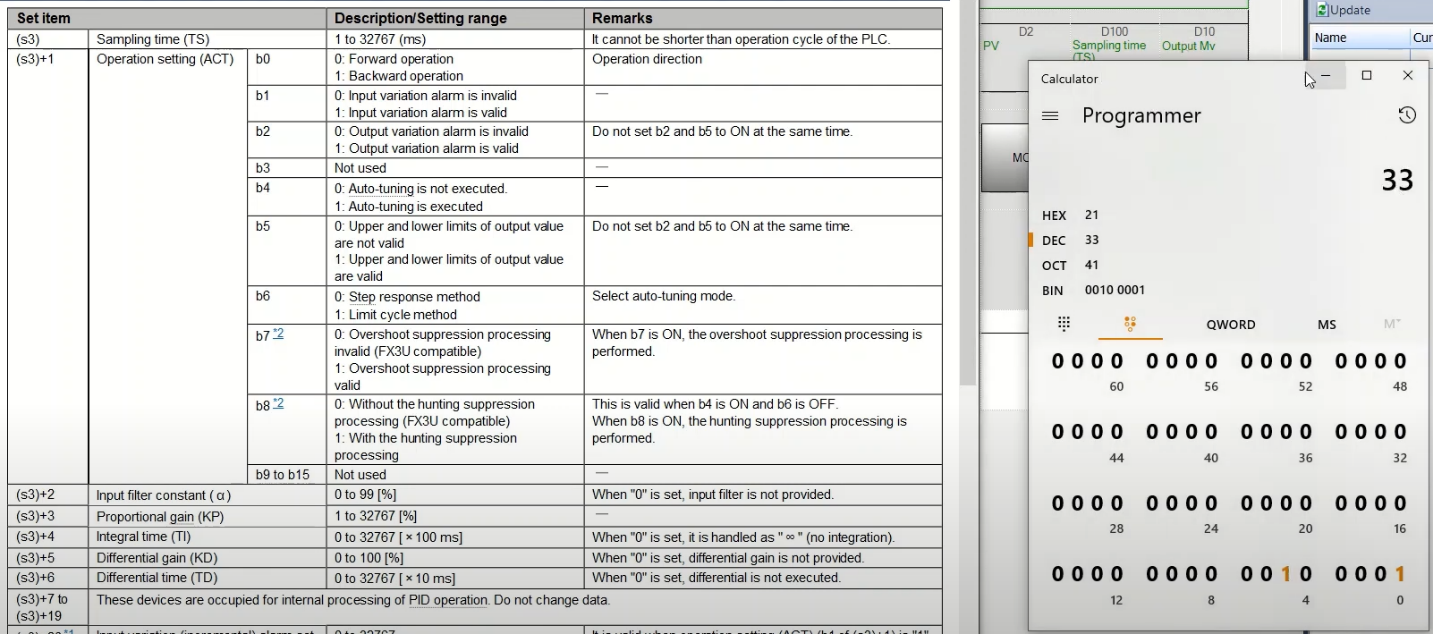
* **გამომავალი მნიშვნელობა (MV);**
* **პარამეტრი;**
* **გაზომილი მნიშვნელობა (PV);**
* **სამიზნე მნიშვნელობა (SP) – (SV).**

**✅ 1. Sampling Time (შესინჯვის დრო)**

ეს არის დროის ინტერვალი, რამდენ ხანში ერთხელ ხდება სისტემის სენსორისგან მონაცემის აღება და PID-ის გაანგარიშება.  
📌 მაგ.: თუ Sampling Time = 0.1 წმ, მაშინ კონტროლერი მუშაობს ყოველ 100 მილიწამში.

**✅ 2. ACT (Actual Value)**

ეს არის სისტემის **ფაქტობრივი მნიშვნელობა** — ანუ ამ მომენტში რეალურად გაზომილი მნიშვნელობა (მაგ., ტემპერატურა, სიჩქარე, პოზიცია და ა.შ.)



**ციფრული ინტეგრალის გამოსათვლელად ხშირად იყენებენ ე.წ. backward Euler (backward Operation) მეთოდს:**

I[n]=I[n−1]+Ts⋅e[n]I[n]

აქ:

* I[n] - მიმდინარე ინტეგრალური ნაწილი
* I[n−1] - წინა მდგომარეობა
* Ts - Sampling Time
* e[n] - მიმდინარე შეცდომა

ეს ითვლება backward ინტეგრაციად, რადგან იგი ეფუძნება წინა მდგომარეობას.

**backward Euler** - "წინა წერტილზე (ანუ წინა მნიშვნელობებზე დაყრდნობით) დაფუძნებული" მეთოდი.

✅ 3**. α (Alpha)**

პარამეტრი **ფილტრაციისთვის** გამოიყენება - ხშირად გამოიყენება **დერივაციული კომპონენტის დაბალპასიან ფილტრში**, რათა დერივაციულ ნაწილში შესული ხმაური (noise) დავაქვეითოთ. ზოგჯერ მას **D-filter coefficient**-ადაც მოიხსენიებენ.

**✅ 4. KP (Proportional Gain)**

ეს არის პროპორციული კომპონენტის გაძლიერების კოეფიციენტი — განსაზღვრავს რეაქციის ძალას შეცდომაზე. რაც უფრო მაღალია KP, მით უფრო სწრაფია რეაგირება, მაგრამ შესაძლოა გამოიწვიოს ოსცილაცია ან არასტაბილურობა.

**✅ 5. IT (Integral Time)**

ინტეგრალური დრო - განსაზღვრავს, რა სიჩქარით აგროვებს კონტროლერი შეცდომებს დროში. ზოგ სისტემაში გამოიყენება **KI = KP / IT** ფორმით, ანუ IT დიდი => ინტეგრალური ეფექტი სუსტია.

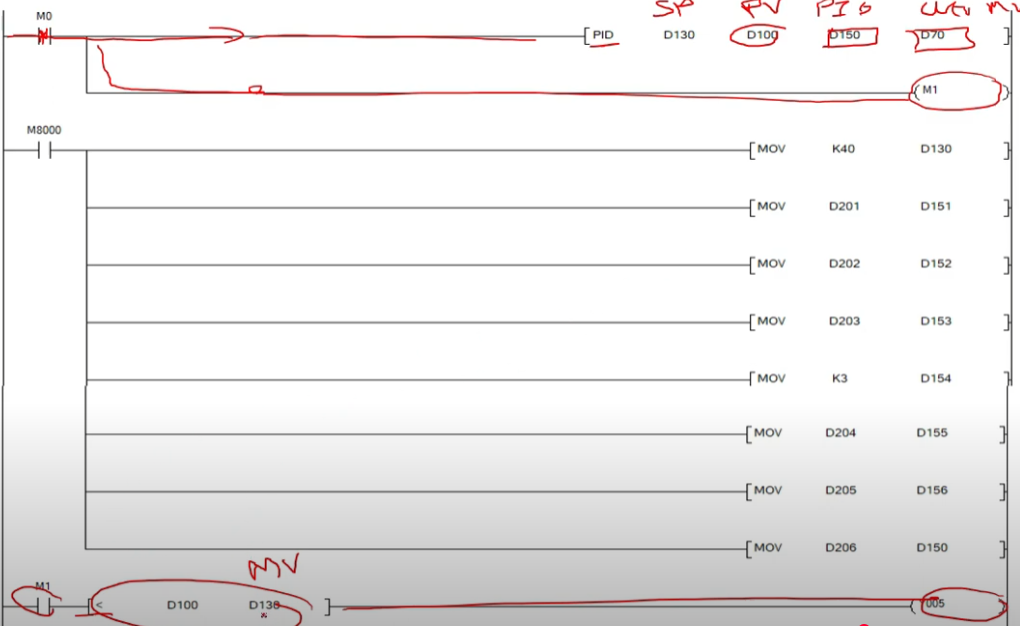
**✅ 6. KD (Derivative Gain)**

ეს კოეფიციენტი აკონტროლებს შეცდომის ცვლილების სიჩქარეზე რეაქციას. უფრო მაღალი KD ეხმარება სისტემას, შეამციროს გადაჭარბება და ოსცილაცია, მაგრამ ხმაურზე მგრძნობიარეა.

**✅ 7. TD (Derivative Time)**

დერივაციული დრო — განსაზღვრავს, რამდენად დიდია დერივაციული კომპონენტის წვლილი. ზოგ მოდელებში გამოიყენება ფორმით **KD = KP × TD**

🔁 **შენიშვნა:**  
ზოგიერთ სისტემაში KP, KI, KD იყენებენ პირდაპირ, ზოგში KP, IT, TD — რაც ერთმანეთთან არის დაკავშირებული.



KP

KI

KD

KE

max

min

**LADDER პროგრამა რელეებიანი PLC მოდულისთვის**

--|M0|----- [PID D130 D140 D100 D90] >> SP PV PID MV

----- (M1)

--|M8000|----- [MOV D200 D100] >> (ST - sampling time) K=100 >>

----- [MOV D201 D101] >> (ACT) K=33 >>

----- [MOV D202 D102] >> (α) K=0 >>

----- [MOV D203 D103] >> (KP) K=200 >>

----- [MOV D204 D104] >> (IT) K=10 >> 20 >> **KI = KP / IT**

----- [MOV D205 D105] >> (KD) K=5 >>

----- [MOV D206 D106] >> (TD) K=100 >> 20 >> **KD = KP × TD**

----- [MOV D207 D122] >> (max) K=100 >>

----- [MOV D208 D123] >> (min) K=0 >>

--|M1|----- [< D140 D130 ]-----------(Y005) >> PV<SV = relay output.

-------------------------------------[END]

**LADDER პროგრამა ტრანზისტორებიანი PLC მოდულისთვ**

--|M0|----- [PID D130 D140 D100 D90] >> SP PV PID MV

----- [PWM D502 K6000 Y00] >>ტრანზისტორზეა შესაძლებელი pwm სიგნალის გატანა

--|M8000|----- [MOV D200 D100] >> (sampling time) K=100 >>

----- [MOV D201 D101] >> (ACT) K=33 >>

----- [MOV D202 D102] >> (α) K=0 >>

----- [MOV D203 D103] >> (KP) K=200 >>

----- [MOV D204 D104] >> (IT) K=10 >> 20

----- [MOV D205 D105] >> (KD) K=5 >>

----- [MOV D206 D106] >> (TD) K=100 >> 20

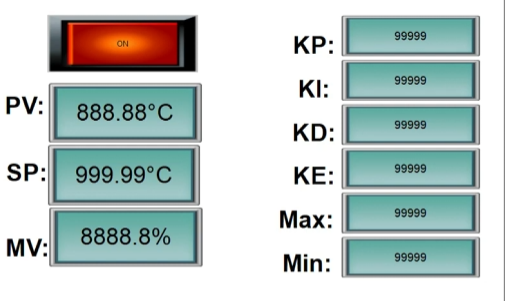
----- [MOV D207 D122] >> (max) K=100 >>

----- [MOV D208 D123] >> (min) K=0 >>

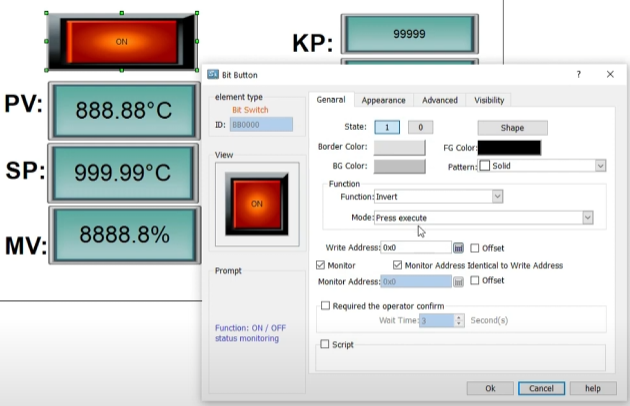
--|M1|----- [< D2 D0 ]-----------(Y005) >>relay output.

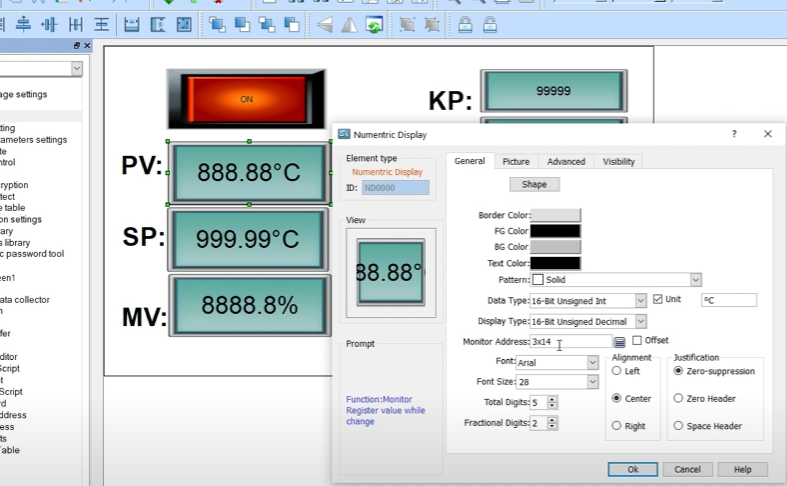
-------------------------------------[END]

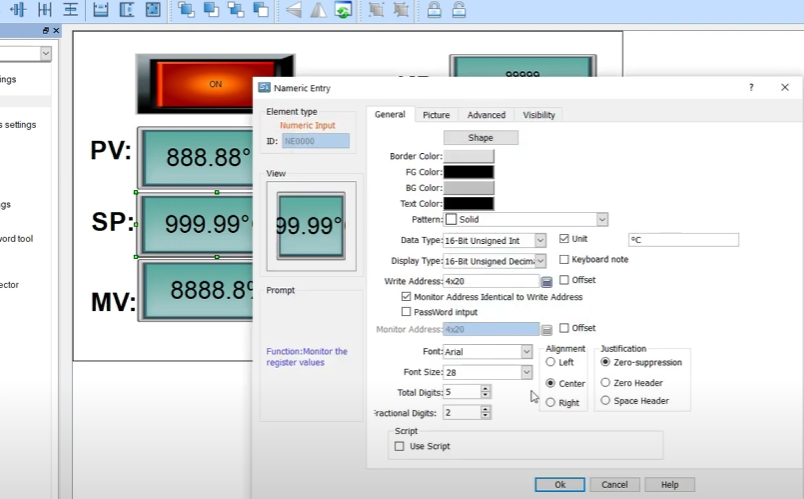
Samkoom HMI

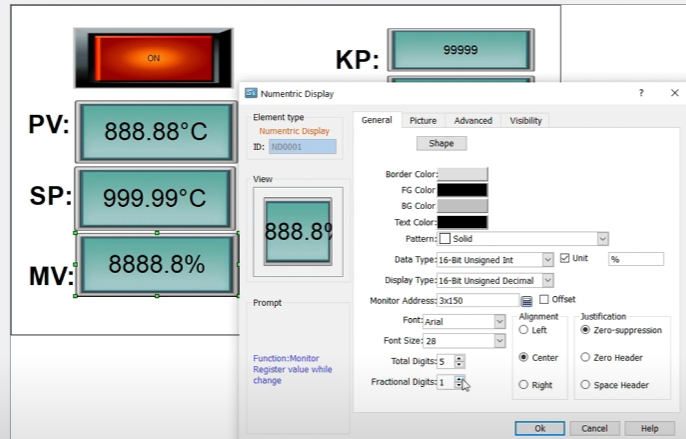


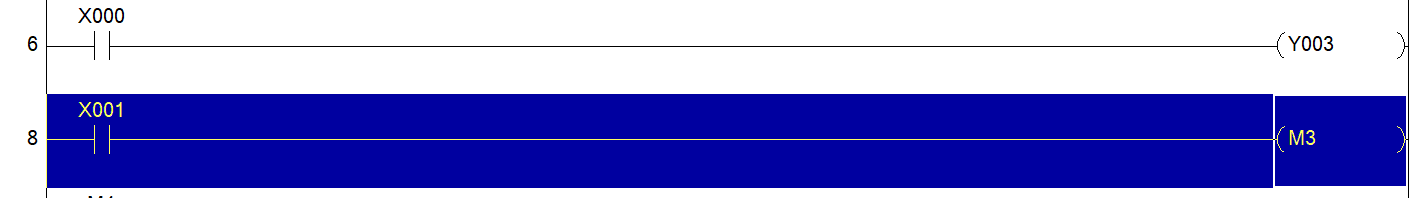
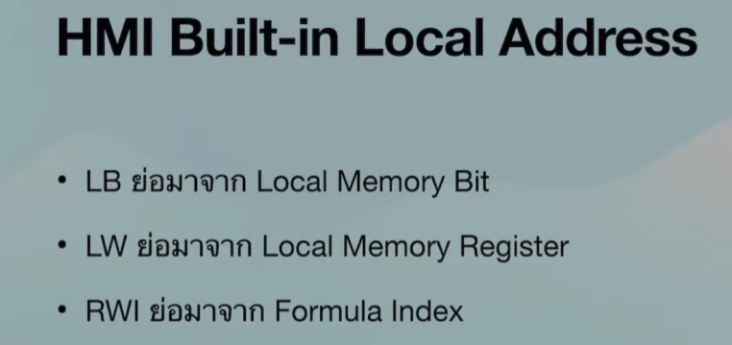


****

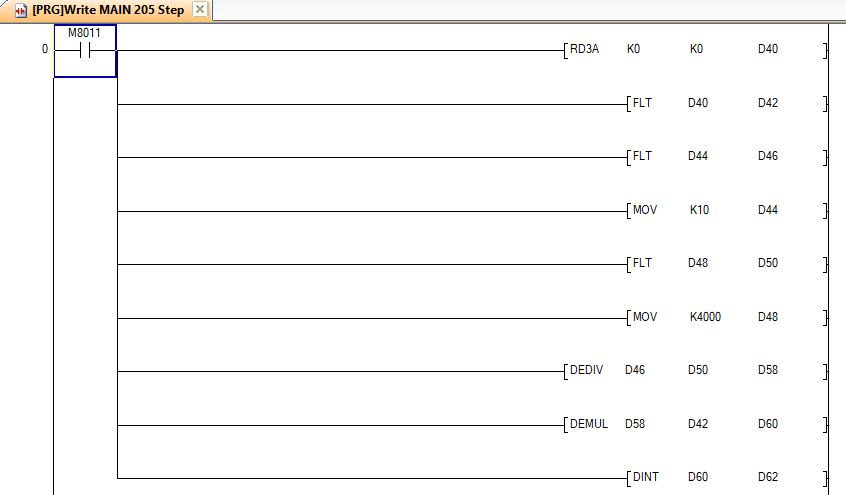
****

****

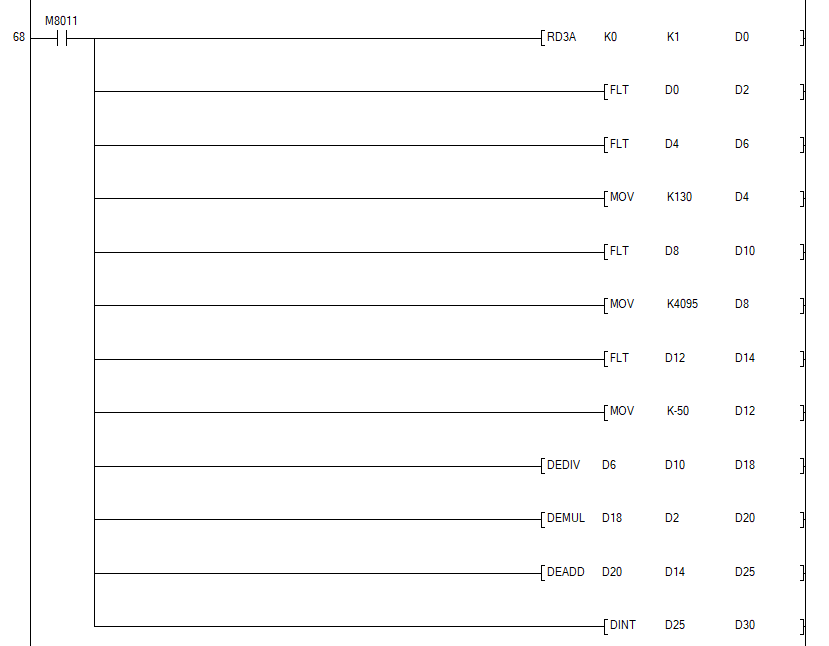
****

****

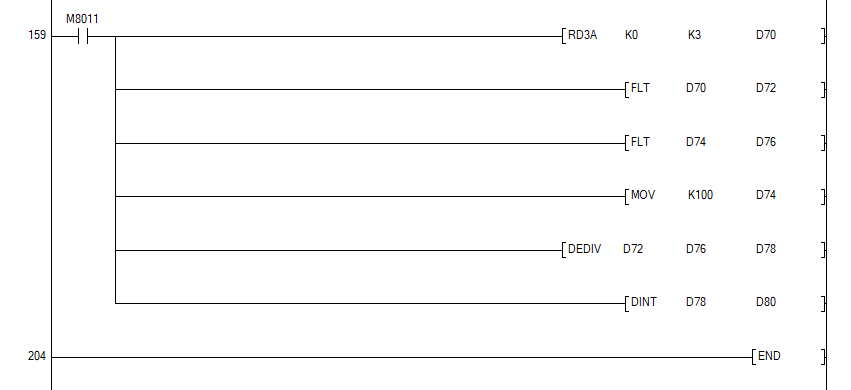
**0 -10 v**

****

**0-10v ntc temperature**

****

**0 – 20 ma**

****

**LADDER პროგრამა რელეებიანი PLC მოდულისთვის**

|  |
| --- |
| **SM411** |

|  |  |
| --- | --- |
| **1 წამიანი პულსი (1-second pulse)** – ტაქტური სიგნალი, რომელიც აქტიურდება ყოველი 1 წამის ინტერვალით. | |
| **SM400** |

|  |
| --- |
| **1-წამიანი პულსის პირველი ნახევარი** — ხდება **ON თითოეული წამის პირველ ნახევარში**, ანუ ყოველი წამის 0.0–0.5 წამში. |

**INC D100 ნიშნავს: "Increment" — ანუ რეგისტრის მნიშვნელობის გაზრდა 1-ით.  
რეგისტრი D100 გაიზარდოს 1-ით**, ყოველ ჯერზე როცა ეს ინსტრუქცია გააქტიურდება.

**DEC D100 ნიშნავს: …**