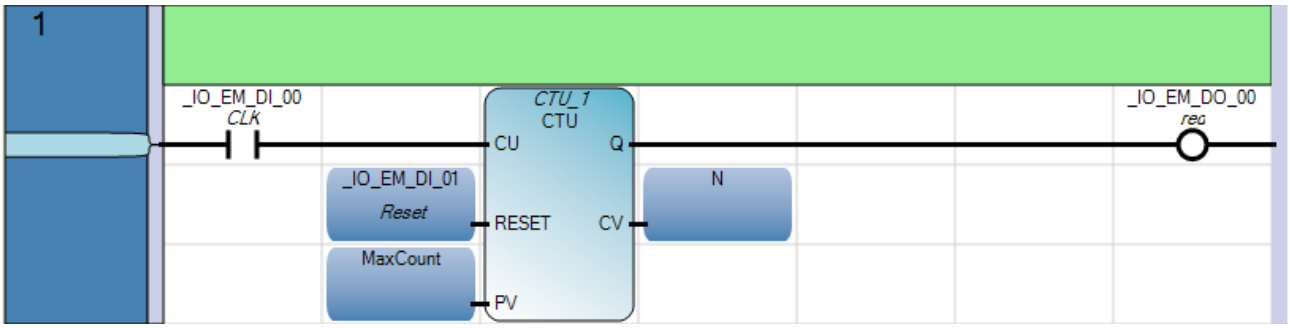
Functional Blocks: Counters

Up Counters: CTU

Count up to 8.

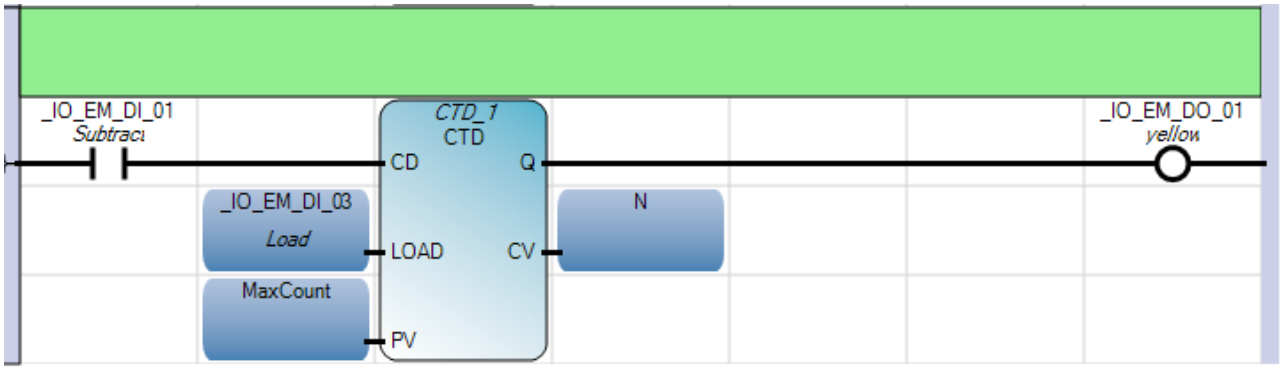
- IN0 = count
- IN1 = reset
- OUT0 = output

Turn on the red LED when you have pressed button IN0 eight times



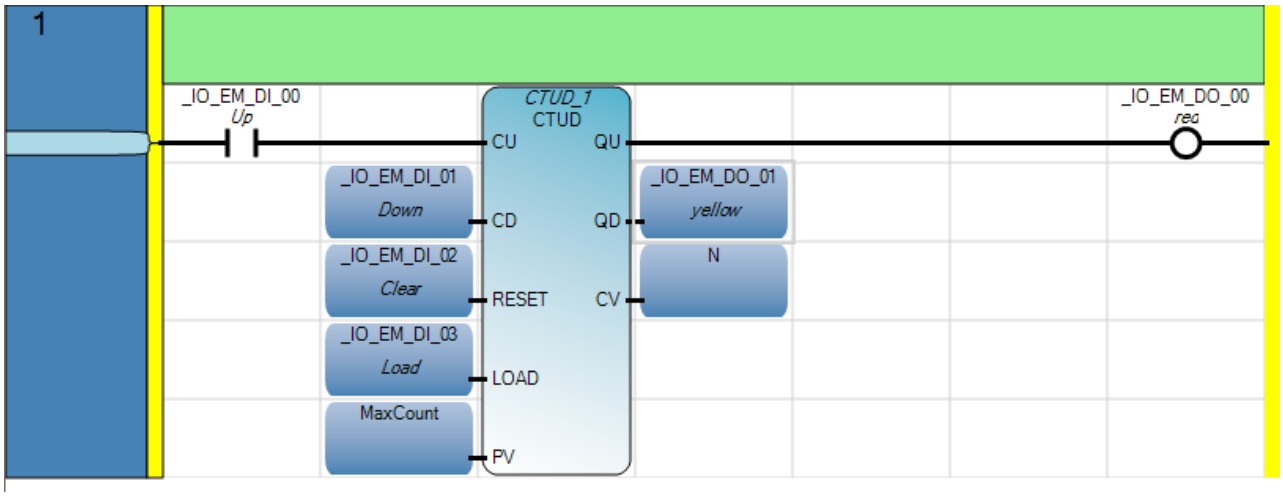
Down Counter: CTD

Decrement a counter

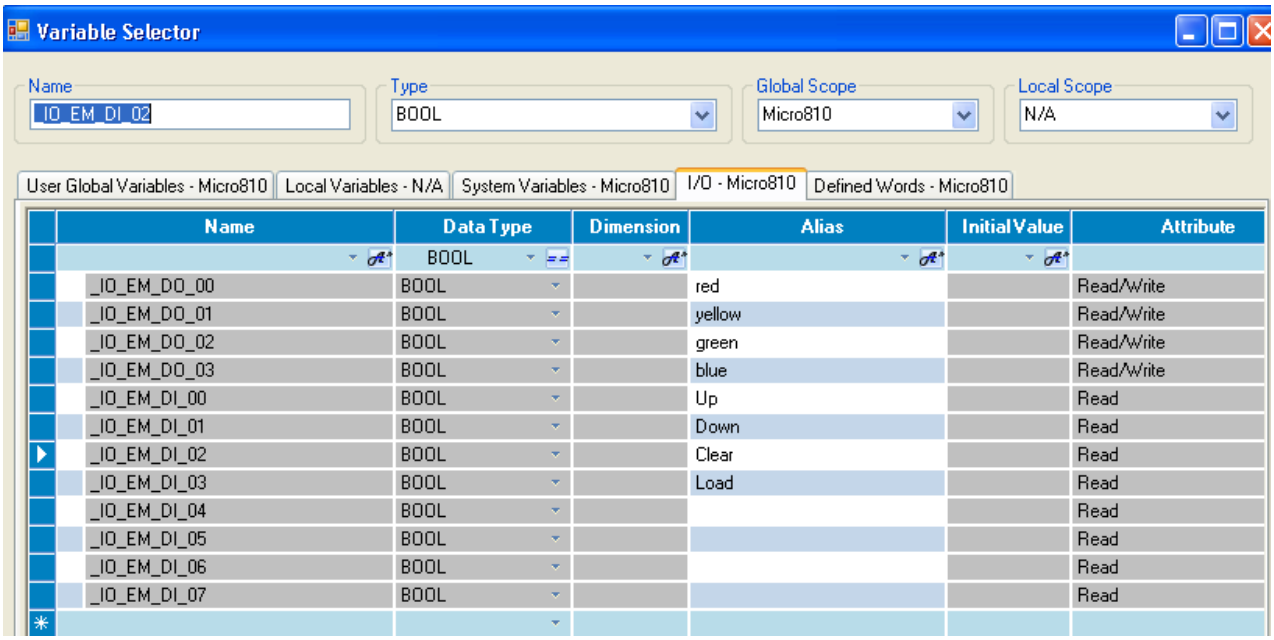


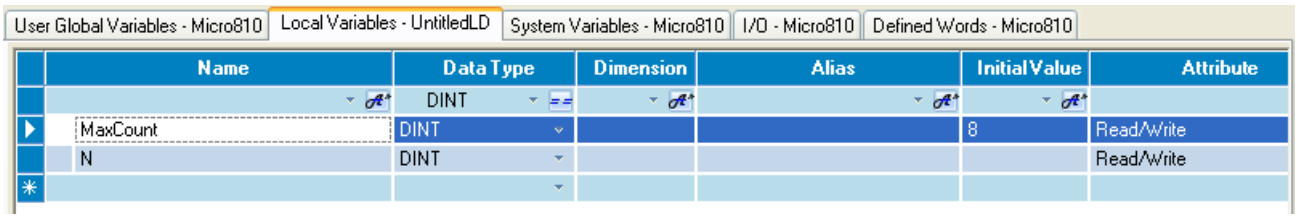
Up-Down Counter: CTUD

- Increment a counter each rising edge of CU
- Decrement a counter each rising edge of CD
- When the count is zero, turn on QD
- When the count is equal to PV, turn on QU



The variables used for this program are as follows:





Comparitors:

Also useful with counters are comparitor blocks. These let you drive an output based upon the present value of a count.

Greater or Equal: >=

Write a program to drive a sign at a parking lot.

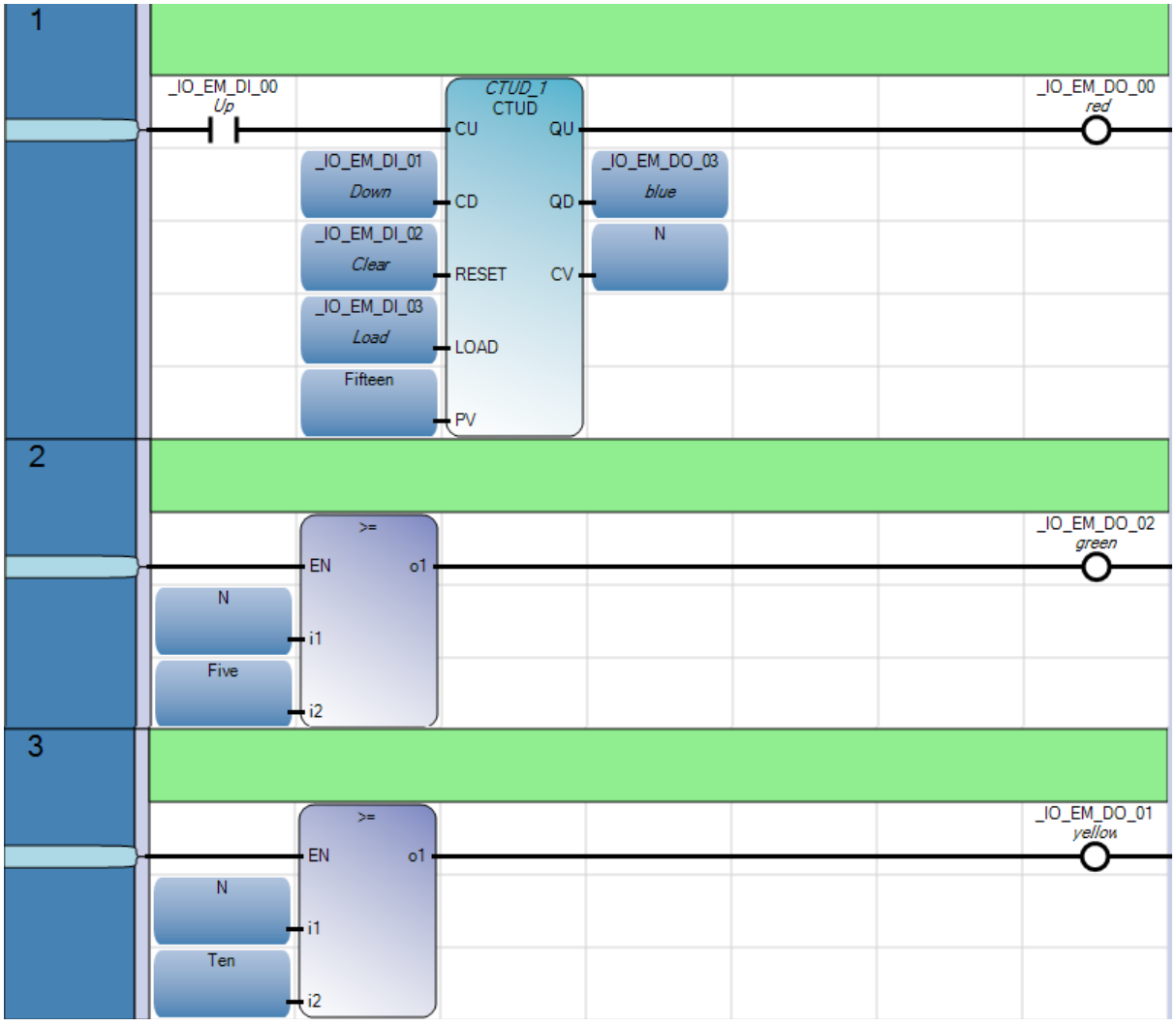
Input

- DI_00 Car enters the lot (count up)
- DI_01 Car leaves the lot (count down)

Output

- DO_03 (Blue) Lot is empty
- DO_02 (Green) 5 or more cars
- DO_01 (Yellow) 10 or more cars
- DO_00 (Red) 15 Cars (lot full)

Program:



Note: You can also build a stoplight using

- A one second timer pulse to keep track of seconds
- A counter to keep track of the time in to one cycle
- Comparitors to turn on and off lights.

Example: Write a program which drives a stoplight using counters. The timing should be

- Green: 5 seconds
- Yellow: 5 seconds
- Red: 5 seconds

for a total cycle time of 15 seconds.

Rung 1: Count up to 15

Rung 2: Green light is on for t < 5

Rung 3: Yellow light is on for 5 < t < 10

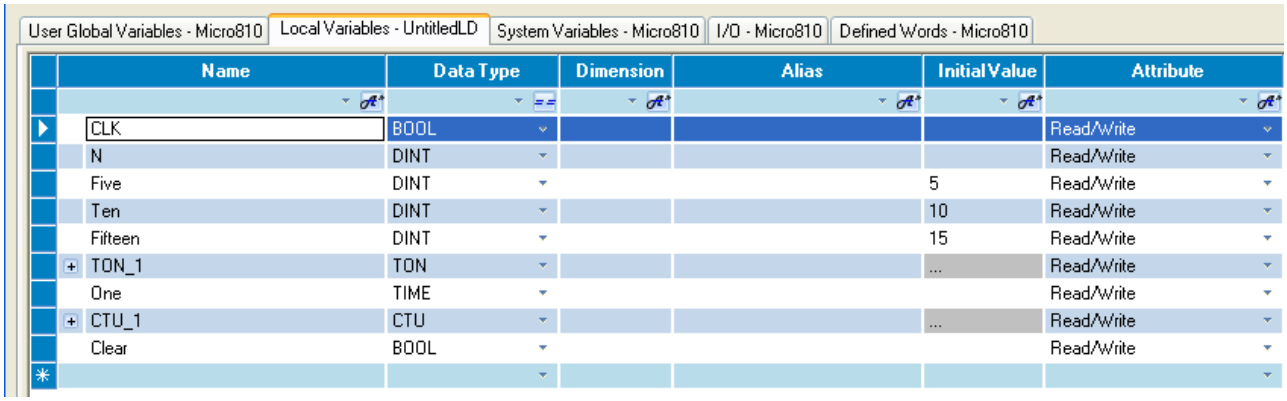
Rung 4: Red light is on for 10 < t

Rung 5: When you get to 15 seconds, clear the counter and restart

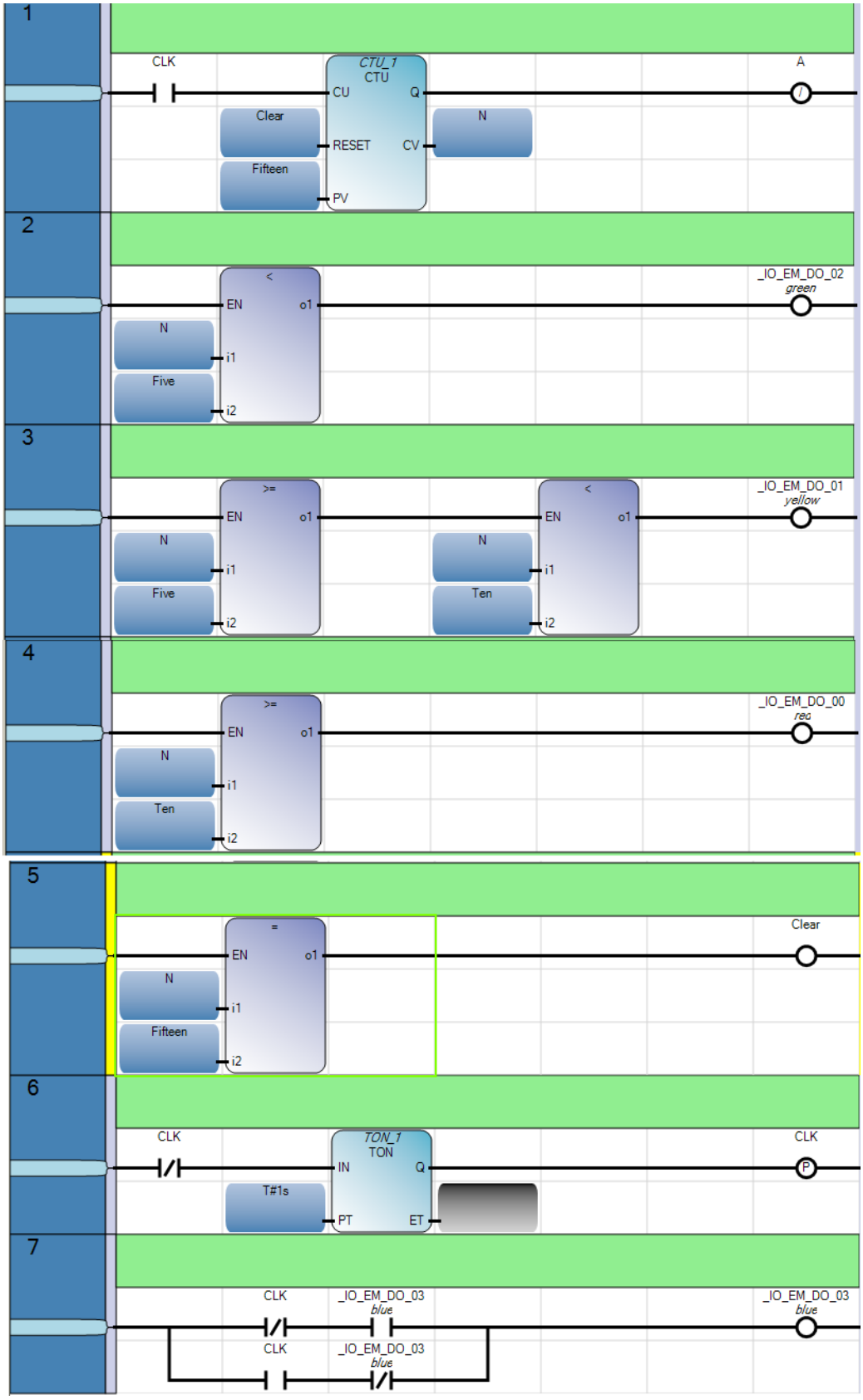
Rung 6: One second counter

Rung 7: Heartbeat. Toggle the blue LED every clock so you can tell the program is running.

Program Variables are:

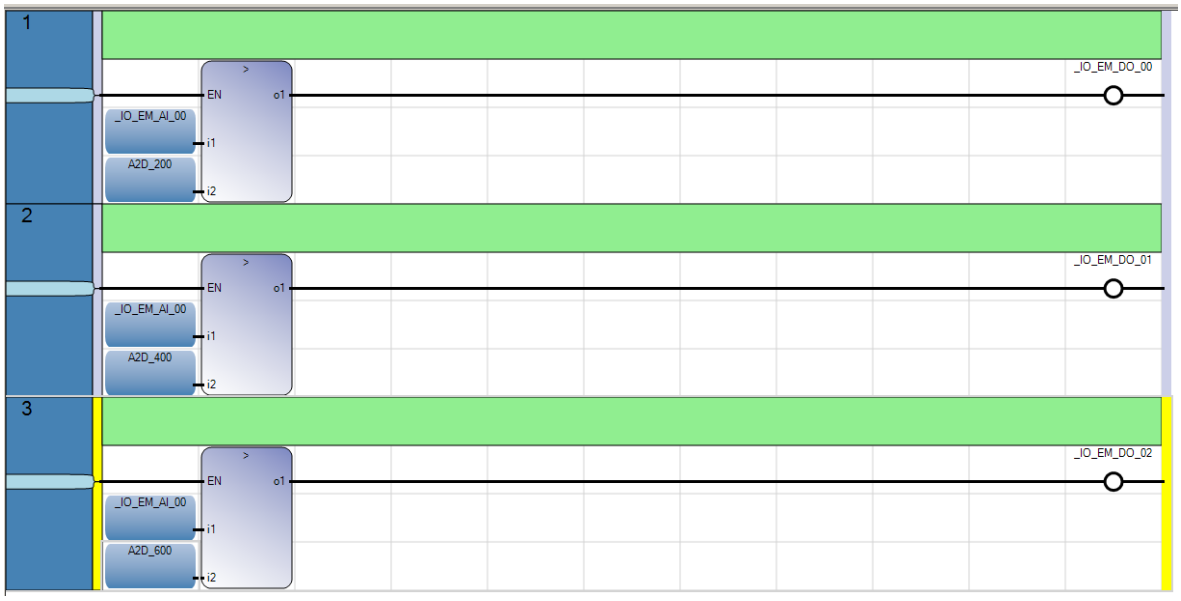


Name	Data Type	Dimension	Alias	Initial Value	Attribute
CLK	BOOL				Read/Write
N	DINT				Read/Write
Five	DINT			5	Read/Write
Ten	DINT			10	Read/Write
Fifteen	DINT			15	Read/Write
TON_1	TON			...	Read/Write
One	TIME				Read/Write
CTU_1	CTU			...	Read/Write
Clear	BOOL				Read/Write


1
CLK
CTU_1
CTU
CU
Q
A
Clear
RESET
CV
N
Fifteen
PV
2
<
EN
o1
_IO_EM_DO_02
green
N
i1
Five
i2
3
>=
EN
o1
_IO_EM_DO_01
yellow
N
i1
Five
i2
<
EN
o1
N
i1
Ten
i2
4
>=
EN
o1
_IO_EM_DO_00
red
N
i1
Ten
i2
5
=
EN
o1
Clear
N
i1
Fifteen
i2
6
CLK
TON_1
TON
IN
Q
CLK
T#1s
PT
ET
7
CLK
_IO_EM_DO_03
blue
_IO_EM_DO_03
blue
CLK
_IO_EM_DO_03
blue

Analog Input: Turn on relays at 200 / 400 / 600

- 0..10V through a 5k pot is applied to I-04



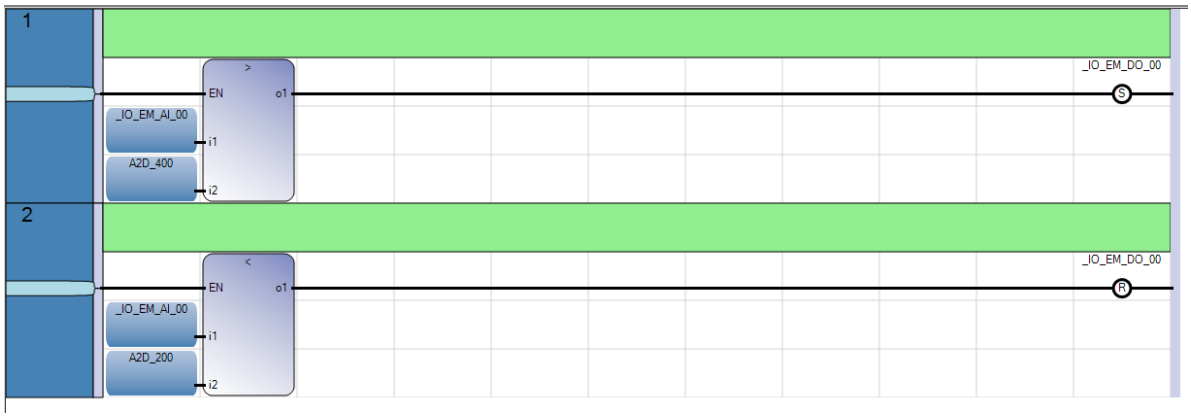
	Name	Data Type	Dimension	Alias	Initial Value	Attribute
		WORD				
▶	A2D_200	WORD			200	Read/Write
	A2D_400	WORD			400	Read/Write
	A2D_600	WORD			600	Read/Write
✳						

Note:

- 200 = 1.97V
- 400 = 4.00V
- 600 = 6.00V

Hysteresis:

- Turn on at 2.00V
- Turn off at 4.00V



Name	Category	Type	Comment
-	Arithmetic	OP	Subtraction of two or more integer or real variables.
*	Arithmetic	OP	Multiplication of two or more integer or real variables.
/	Arithmetic	OP	Division of two or more integer or real variables.
+	Arithmetic	OP	Addition of two or more integer or real variables
<	Comparators	OP	Test whether one value is LESS THAN another (on integer, real, tim
<=	Comparators	OP	Test whether one value is LESS THAN or EQUAL TO another (on in
<>	Comparators	OP	Test whether one value is NOT EQUAL to another (on integer, real,
=	Comparators	OP	Test whether one value is EQUAL to another (on integer, real, time,
>	Comparators	OP	Test whether one value is GREATER THAN another (on integer, rea
>=	Comparators	OP	Test whether one value is GREATER THAN or EQUAL TO another (
1 gain	Arithmetic	OP	Assignment of one variable to another
ABS	Arithmetic	SF	Absolute value
ACOS	Arithmetic	SF	Arc cosine
ACOS_LREAL	Arithmetic	C	Perform 64-bit real arccosine calculation.
AND	Boolean operations	OP	Boolean AND between two or more terms
AND_MASK	Binary operations	SF	Analog bit to bit AND mask
ANY_TO_BOOL	Data conversion	OP	Conversion of any variable to a Boolean variable.
ANY_TO_BYTE	Data conversion	OP	Conversion of any variable to a byte variable.
ANY_TO_DATE	Data conversion	OP	Conversion of any variable to a date variable.
ANY_TO_DINT	Data conversion	OP	Conversion of any variable to a double integer variable.
ANY_TO_DWOR	Data conversion	OP	Conversion of any variable to a double word variable.
ANY_TO_INT	Data conversion	OP	Conversion of any variable to a single integer variable.
ANY_TO_LINT	Data conversion	OP	Conversion of any variable to a long integer variable.
ANY_TO_LREAL	Data conversion	OP	Conversion of any variable to a long real variable.
ANY_TO_LWOR	Data conversion	OP	Conversion of any variable to a long word variable.
ANY_TO_REAL	Data conversion	OP	Conversion of any variable to a real variable.
ANY_TO_SINT	Data conversion	OP	Conversion of any variable to a short integer variable.
ANY_TO_STRIN	Data conversion	OP	Conversion of any variable to a string variable.
ANY_TO_TIME	Data conversion	OP	Conversion of any variable to a timer variable.
ANY_TO_UDINT	Data conversion	OP	Conversion of any variable to an unsigned double integer variable.
ANY_TO_UINT	Data conversion	OP	Conversion of any variable to an unsigned single integer variable.
ANY_TO_ULINT	Data conversion	OP	Conversion of any variable to an unsigned long integer variable.
ANY_TO_USINT	Data conversion	OP	Conversion of any variable to an unsigned short integer variable.
ANY_TO_WORD	Data conversion	OP	Conversion of any variable to a word variable.
ASCII	String manipulation	SF	Character -> ASCII code
ASIN	Arithmetic	SF	Arc sine
ASIN_LREAL	Arithmetic	C	Perform 64-bit real arcsine calculation.
ATAN	Arithmetic	SF	Arc tangent
ATAN_LREAL	Arithmetic	C	Perform 64-bit real arctangent calculation.
AVERAGE	Data Manipulation	SB	Running average over N samples

Name	Category	Type	Comment
CHAR	String manipulation	SF	ASCII code -> Character
COS	Arithmetic	SF	Cosine
COS_LREAL	Arithmetic	C	Perform 64-bit real cosine calculation.
CTD	Counter	SB	Down counter
CTU	Counter	SB	Up counter
CTUD	Counter	SB	Up-down counter
DELETE	String manipulation	SF	Delete sub-string
DERIVATE	Process Control	SB	Differentiation according to time
DOY	Time	C	Turn on output when real-time clock value is within year range.
EXPT	Arithmetic	SF	Exponent
F_TRIG	Boolean operations	SB	Falling edge detection
FIND	String manipulation	SF	Find sub-string
HYSTER	Process Control	SB	Boolean hysteresis on difference of reals
INSERT	String manipulation	SF	Insert string
INTEGRAL	Process Control	SB	Integration over time
IPIDCONTROLLE	Process Control	C	Proportional Integral Derivative.
KEY_READ	Micro800	C	Read key status on option LCD module.
LCD	Micro800	C	Display string or number according to user requirements if option LC
LEFT	String manipulation	SF	Extract left of a string
LIM_ALRM	Alarms	SB	High/low limit alarm with hysteresis
LIMIT	Process Control	SF	Limit
LOG	Arithmetic	SF	Logarithm
MAX	Data Manipulation	SF	Maximum
MID	String manipulation	SF	Extract middle of a string
MIN	Data Manipulation	SF	Minimum
MLEN	String manipulation	SF	Get string length
MM_INFO	Input/Output	C	Read memory module header information.
MOD	Arithmetic	SF	Modulo
MUX4B	Boolean	C	Multiplexer(4 entries) - accepts BOOL inputs and output value.
MUX8B	Boolean	C	Multiplexer(8 entries) - accepts BOOL inputs and output value.
Neg	Arithmetic	OP	Assignment of the negation to an integer variable
NOT	Boolean operations	OP	Assigment of the negation to a Boolean variable
NOT_MASK	Binary operations	SF	bit to bit negation
OR	Boolean operations	OP	Boolean OR of two or more terms
OR_MASK	Binary operations	SF	Analog bit to bit OR mask
POW	Arithmetic	SF	Power calculation
R_TRIG	Boolean operations	SB	Rising edge detection
RAND	Arithmetic	SF	Random value
REPLACE	String manipulation	SF	Replace sub-string
RHC	Input/Output	F	Read high-speed clock.

Name	Category	Type	Comment
RIGHT	String manipulation	SF	Extract right of a string
ROL	Binary operations	SF	Rotate Left
ROR	Binary operations	SF	Rotate Right
RPC	Input/Output	C	Reads user program checksum.
RS	Boolean operations	SB	Reset dominant bistable
RTC_READ	Input/Output	C	Read RTC module information.
RTC_SET	Input/Output	C	Set RTC data to RTC module.
SCALER	Process Control	C	Scale input value according to output range.
SHL	Binary operations	SF	Shift Left
SHR	Binary operations	SF	Shift Right
SIN	Arithmetic	SF	Sine
SIN_LREAL	Arithmetic	C	Perform 64-bit real sine calculation.
SQRT	Arithmetic	SF	Square root
SR	Boolean operations	SB	Set dominant bistable
STACKINT	Process Control	SB	Stack of integer analogs
SUS	Program Control	C	Suspend the execution of the application.
SYS_INFO	Input/Output	C	Read Micro800 system status.
TAN	Arithmetic	SF	Tangent
TAN_LREAL	Arithmetic	C	Perform 64-bit real tangent calculation.
TDF	Time	C	Compute time difference.
TND	Program Control	C	Abort current user program scan.
TOF	Time	SB	Off-delay timing
TON	Time	SB	On-delay timing
TONOFF	Time	C	Delay an output-on(true), then delay an output-off(false).
TOW	Time	C	Turn on output when real-time clock value is within week range.
TP	Time	SB	Pulse timing
TRUNC	Arithmetic	SF	Truncate decimal part
TTABLE	Boolean	C	Provide the value output based on the combination of inputs.
XOR	Boolean operations	OP	Boolean exclusive OR between two terms.
XOR_MASK	Binary operations	SF	Analog bit to bit Exclusive OR mask