PL400

PROGRAMMABLE LOGIC CONTROLLER WITH INDUCTIVE LOOP DETECTOR

USER MANUAL

PROCON ELECTRONICS

P.O.Box 24
STANFIELD 3613
SOUTH AFRICA

Tel: +27 (031) 7028033
Fax: +27 (031) 7028041
Email: proconel@proconel.com
Web: www.proconel.com

29/06/2004 V01

TABLE OF CONTENTS

1. AN OVERVIEW OF THE PL400 PROGRAMMABLE LOGIC CONTROLLER

1.1 DESCRIPTION

The PL400 PLC has been developed as a versatile controller for use in parking or access control applications. The PLC can be used to operate gates or barriers where an inductive loop detector is required to sense the presence of vehicles. The fact that the controller is programmable enables the user to program their own unique logic requirements and not be restricted by a pre-programmed unit.

The PL400 PLC is programmed in ladder logic. PROCON's PROSOFT windows-based PC software is used to generate the ladder diagram, compile the program, and then download the program to the PL400 via the programming port on the front of the unit.

The I/O consists of 8 digital inputs and 4 relay outputs. The inputs are opto-isolated and a built in field supply is used to power the inputs so no external power supply is required. The PLC has a user programmable toggle switch on the front, which can be incorporated into the ladder program as manual inputs. Two led's are also provided for user functions.

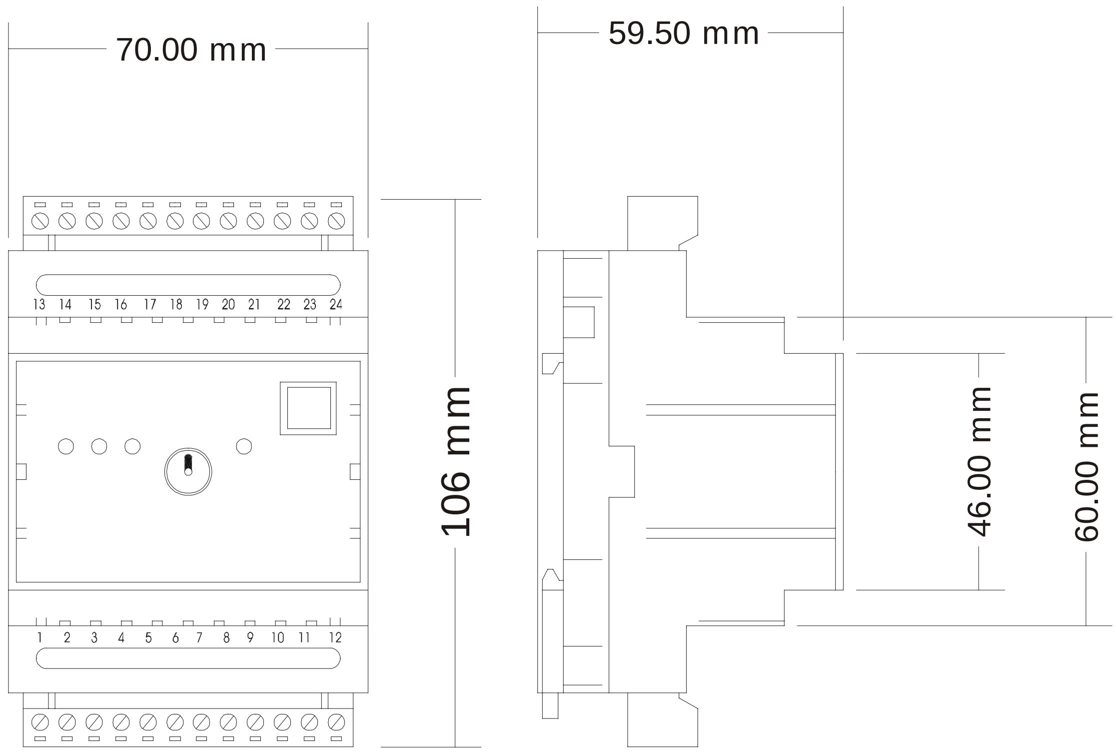
All wiring is done with screw terminals on removable connectors.

The programming port requires the use of a special adaptor to connect it to an RS232 communications port of a PC. This port supports the Modbus RTU protocol and all of the internal registers and I/O status can be accessed through this port.

2. PL400 GENERAL INFORMATION

2.1 PHYSICAL DIMENSIONS

The PL400 enclosure is shown below. The module has been designed with a quick snap-in assembly for mounting onto DIN-rail's as per DIN EN 50 022.



2.2 GROUNDING/SHIELDING

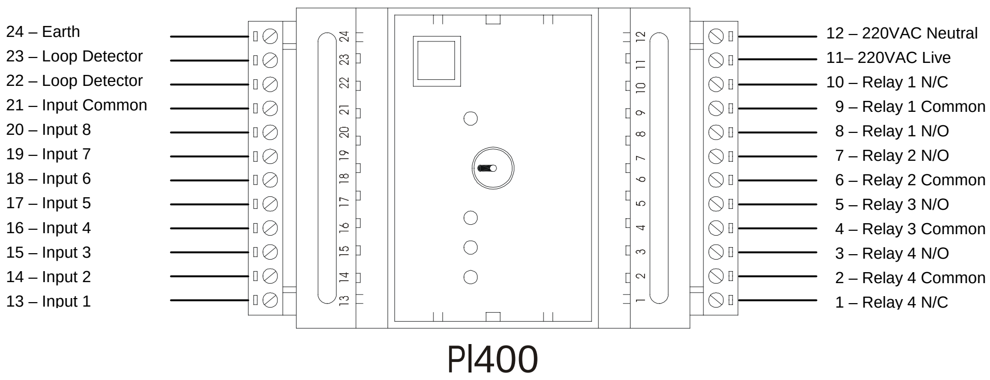
In most cases, the PL400 will be installed in an enclosure along with other devices, which generate electromagnetic radiation. Examples of these devices are relays and contactors, transformers, motor controllers etc. This electromagnetic radiation can induce electrical noise into both power and signal lines, as well as direct radiation into the module causing negative effects on the system. Appropriate grounding, shielding and other protective steps should be taken at the installation stage to prevent these effects. These protective steps include control cabinet grounding, module grounding, cable shield grounding, protective elements for electromagnetic switching devices, correct wiring as well as consideration of cable types and their cross sections.

3. PL400 HARDWARE

3.1 SPECIFICATIONS

POWER REQUIREMENT:	200 - 260VAC 50/60Hz.
8 X INPUTS:	These inputs may be activated by a potential free relay contact or open collector NPN transistor output. These inputs are isolated from the logic.
4 X OUTPUTS:	These outputs are a normally open relay contact rated at 6A/220VAC (resistive).
INDICATORS:	LED indicators show power, user1, user2 and programming communications.
LOOP RESPONSE TIME:	Approximately 120ms after vehicle enters loop.
DETECTOR TUNING RANGE:	15 - 1500 uH.
LOOP PROTECTION:	Loop isolation transformer with lightning protection.
CONNECTORS:	2 X 12 Way Removable Connectors with screw terminals.
DIMENSIONS:	106mm (HIGH) X 70mm (WIDE) X 59.5mm (DEEP)
OPERATING TEMPERATURE:	-20°C to +60°C
STORAGE TEMPERATURE:	-20°C to +65°C
HUMIDITY:	up to 95% non condensing

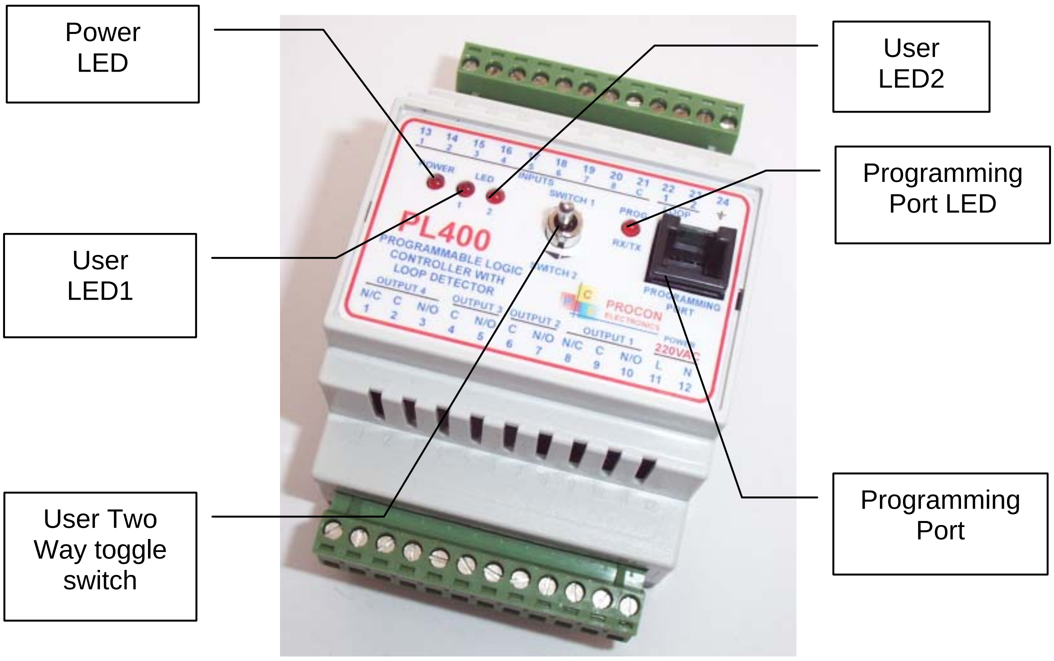
3.2 WIRING TERMINALS



3.3 FRONT PANEL DESCRIPTION.

The led's on the front panel of the PL400 Module are used to indicate power, and user defined LED1 and LED2. A programming LED is used to indicate communications with a PC during programming and Debugging.

A toggle switch is provided which allows you to use this in your application, for as an example, a manual override switch.

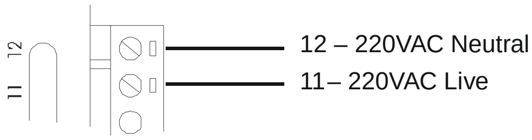


4. CONFIGURATION

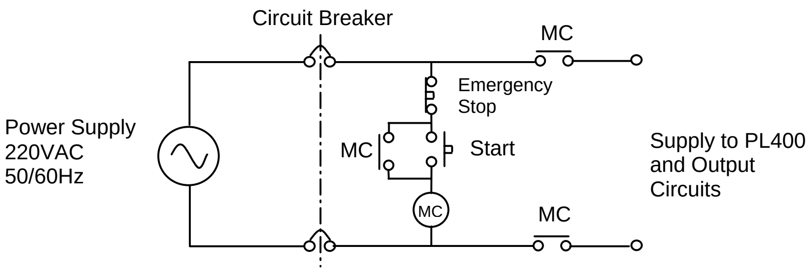
4.1 HARDWARE CONNECTIONS.

4.1.1 Connecting the Power.

Power must be applied to terminal 11 (220VAC LIVE) and terminal 12 (220VAC NEUTRAL). When the power is initially applied the power LED will illuminate and all other LED's will be off.

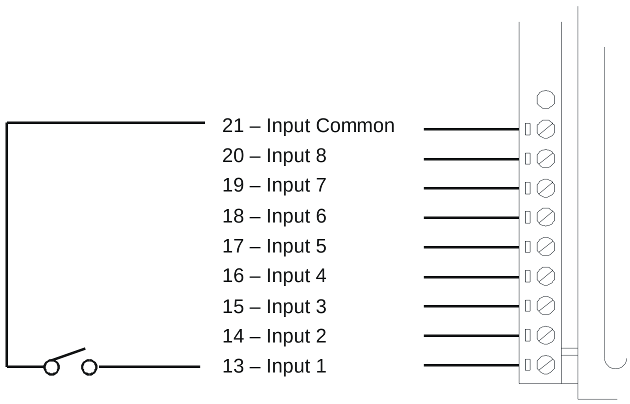


As the PLC is often used to control machinery, which could present a risk of personal injury or damage to equipment, it is good practice to wire an external emergency stop circuit to the power supply on the PLC. The circuit below shows how a mechanical contactor (MC) is used with start/stop buttons to provide this facility.



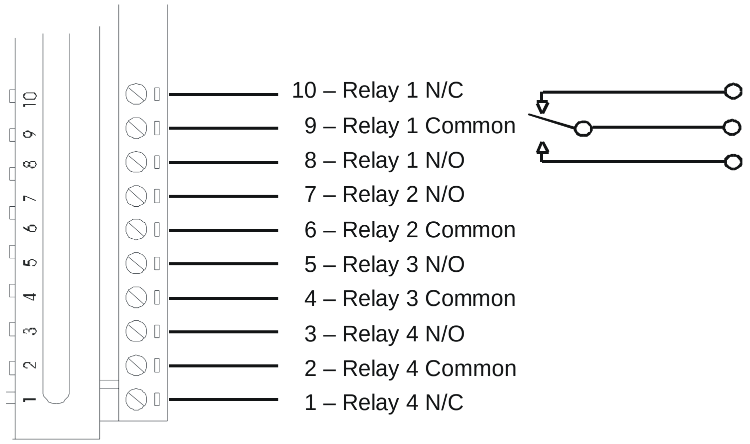
4.1.2 Connecting the Inputs.

The inputs are sourced from an internally isolated power supply and can be switched by a potential free contact or a NPN transistor. The inputs all share the common terminal.

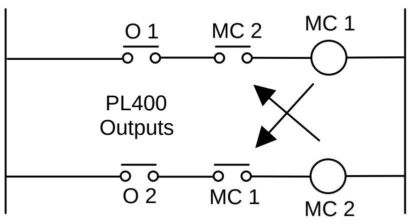


4.1.3 Connecting the Outputs.

The outputs are potential free relay contacts. Note that only Relay output 1 and 4 have normally closed contacts as well as normally open contacts.

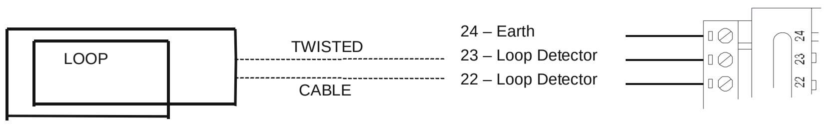


The outputs may be used to control the direction of a motor. For example output 1 could be used to control the forward direction of the motor and output 2 used to control the reverse direction. It could be possible under fault conditions that both outputs switch on at the same time. It is considered good practice to interlock the two outputs both in the ladder program and using external mechanical contactors. The diagram below shows how this is done.



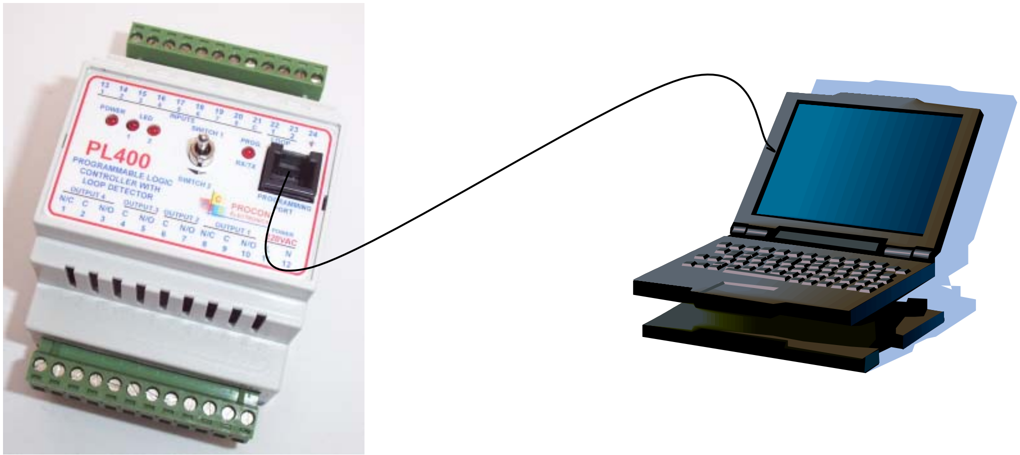
4.1.4 Connecting the Inductive Loop.

Refer to the chapter further in the manual on installing the inductive loop. The diagram below shows how to connect the loop to the PL400.



4.1.5 Connecting the programming port to a PC.

The PL400 programming port is connected to a RS232 communications port on the PC using a special programming cable supplied by Procon. The RS232 connector is a DB-9 plug which plugs into the PC.



4.2 PL400 CPU.

The CPU (central Processing Unit) performs all of the tasks that are required to make the PLC function and run your ladder program. Some of the tasks include:

1. Reading the status of the inputs.
2. Executing the program.
3. Updating the outputs.
4. Doing diagnostics.
5. Servicing the communications ports.
6. Running the timers.

4.2.1 Program Memory.

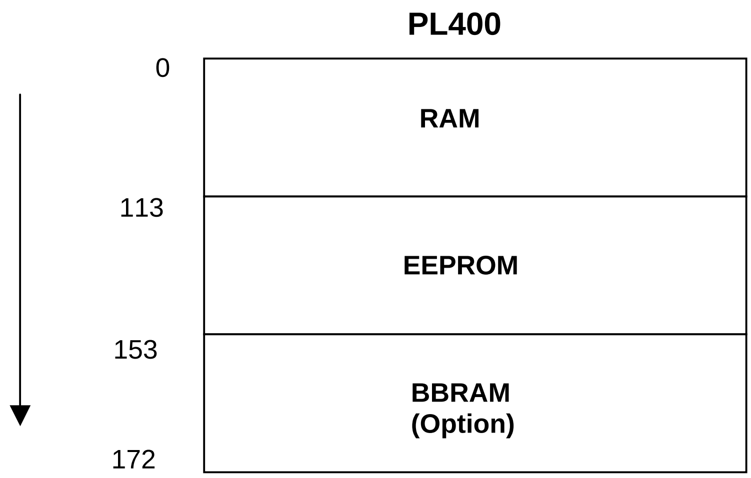
The programming port is used to program the PLC. The program which is sent from the PC using the ProSoft ladder editor, is stored in FLASH memory. This memory does not get lost when the power fails and so will remain permanently in the PLC until it is reprogrammed.

4.2.2 Data Memory.

All the variables used in the program are stored in Data memory. Both the Digital and Analog values are stored in this memory along with the timers, counters, and user memory.

The memory is divided up into 3 sections.

1. RAM – Random Access Memory. This memory is the most widely used memory and is where most of the data is stored. All timers, counters, I/O statuses and system information use this memory. If the power fails then all the information in this memory is lost and is re-initialized to zero when the PLC starts again.
2. EEPROM – This memory is used to store parameters such as set-points and configuration data as it retains its memory when the power is turned off. The one point to remember is that this memory can only be written to 10 000 times before it wears out so you must not write to this memory all the time as you can with RAM.
3. BBRAM (Optional extra) – This is battery backed RAM and also retains its memory when the power is switched off. This memory is slow compared to RAM and should not be used where normal RAM can be used. This memory is ideal for storing values such as used in counting applications. The Real time clock is also stored in this memory.



4.2.3 Data Memory Map.

All of the variables used in the PLC are stored in data memory. In order for your program to get access to these variables you need to know the memory address. The memory address starts at zero and the size depends on the PLC being used. Each memory location consists of 16 bits. Thus one memory location can be used to store the status of 16 digital I/O points or an analog value from 0 to 65535. Some of the ladder functions use two consecutive memory locations to store larger values. Refer to the ProSoft user manual to find out about the ladder functions.

PL400 MEMORY MAP			
Memory Type	Digital Reference	Memory Address	Quantity
Module Type = 40	-	M0	1
Digital Inputs	I1 to I14	M1	14
Digital Outputs	O1 to O6	M2	6
Timer Status	T1 to T16	M3	16
Counter Status	C1 to C16	M4	16
Control Relays	R1 to R64	M5 – M8	64
System Relays	S1 to S32	M9 – M10	32
Timer Memory	-	M13 – M28	16
Counter Memory	-	M29 – M44	16
User RAMMemory	-	M45 – M112	68
User EEPROM	-	M113 – M152	40
User BBRAM	-	M153 – M172	20

4.2.4 Digital Input Map.

MSB							PL400 DIGITAL INPUTS								LSB	ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	M1

Bit Number	Digital Input Number	Description
0	I1	Digital Input 1
1	I2	Digital Input 2
2	I3	Digital Input 3
3	I4	Digital Input 4
4	I5	Digital Input 5
5	I6	Digital Input 6
6	I7	Digital Input 7
7	I8	Digital Input 8
8	I9	Toggle Switch 1
9	I10	Toggle Switch 2
10	I11	Loop Detect
11	I12	Loop Pulse
12	I13	Loop Detect LED
13	I14	Loop Fault LED
14	-	-
15	-	-

4.2.5 Digital Output Map.

MSB							PL400 DIGITAL OUTPUTS								LSB	ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-	-	-	-	-	-	-	-	-	-	O6	O5	O4	O3	O2	O1	M2

Bit Number	Digital Input Number	Description
0	O1	Relay Output 1
1	O2	Relay Output 2
2	O3	Relay Output 3
3	O4	Relay Output 4
4	O5	LED 1
5	O6	LED 2
6	-	-
7	-	-
8	-	-
9	-	-
10	-	-
11	-	-
12	-	-
13	-	-
14	-	-
15	-	-

4.2.6 Timer Map.

MSB						PL400 TIMER STATUS									LSB	ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
T16	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	M3

Bit Number	Digital Input Number	Description
0	T1	Timer 1
1	T2	Timer 2
2	T3	Timer 3
3	T4	Timer 4
4	T5	Timer 5
5	T6	Timer 6
6	T7	Timer 7
7	T8	Timer 8
8	T9	Timer 9
9	T10	Timer 10
10	T11	Timer 11
11	T12	Timer 12
12	T13	Timer 13
13	T14	Timer 14
14	T15	Timer 15
15	T16	Timer 16

4.2.7 Counter Map.

MSB						PL400 COUNTER STATUS									LSB	ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
C16	C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	M4

Bit Number	Digital Input Number	Description
0	C1	Counter 1
1	C2	Counter 2
2	C3	Counter 3
3	C4	Counter 4
4	C5	Counter 5
5	C6	Counter 6
6	C7	Counter 7
7	C8	Counter 8
8	C9	Counter 9
9	C10	Counter 10
10	C11	Counter 11
11	C12	Counter 12
12	C13	Counter 13
13	C14	Counter 14
14	C15	Counter 15
15	C16	Counter 16

4.2.8 Control Relay Map.

MSB							PL400 CONTROL RELAYS								LSB	ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	M5
R32	R31	R30	R29	R28	R27	R26	R25	R24	R23	R22	R21	R20	R19	R18	R17	M6
R48	R47	R46	R45	R44	R43	R42	R41	R40	R39	R38	R37	R36	R35	R34	R33	M7
R64	R63	R62	R61	R60	R59	R58	R57	R56	R55	R54	R53	R52	R51	R50	R49	M8

4.2.9 System Relay Map.

MSB							PL400 SYSTEM RELAYS								LSB	ADDRESS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	M9
S32	S31	S30	S29	S28	S27	S26	S25	S24	S23	S22	S21	S20	S19	S18	S17	M10

Bit Number	Digital Input Number	Description
0	S1	ON
1	S2	1st Scan
2	S3	0.1 Second Clock Period
3	S4	1 Second Clock Period
4	S5	1 Minute Clock Period
5	S6	CMP < MEM/K
6	S7	CMP = MEM/K
7	S8	CMP > MEM/K
8	S9	PLC Running
9	S10	PLC Re-Program Request
10	S11	PLC Re-Program Acknowledge
11	-	-
12	-	-
13	-	-
14	-	-
15	-	-

5. Inductive Loop Detector

5.1 Using the Loop Detector.

The inductive loop detector is used to detect the presence of vehicles and can be used as a safety device, for arming or giving a pulse to operate the logic or may be used for counting.

The inputs I11 to I14 are used to get information from the loop detector.

I11	Loop Detect
I12	Loop Pulse
I13	Loop Detect LED
I14	Loop Fault LED

I11 – This bit is on when a vehicle is over the loop (in detect) and is off when there is no vehicle present.

I12 – This bit is on for a short period of time to give a pulse output when the vehicle either enters (goes into detect) or leaves the loop (goes out of detect). The mode register is used to configure the pulse type and the pulse time register is used to configure the duration of the pulse.

I13 – If the loop detect is in detect then this bit is on. This bit will toggle when the power is first applied to the PL400. The number of toggles indicates the loop frequency. You can use your ladder program to read in this bit and output it to an LED.

I14 – If there is a fault on the loop then this bit will come on. It is automatically turned off when the fault is cleared and the detector has retuned.

5.2 Configuring the Loop Detector.

The parameters, which are used to configure the loop detector, are stored in the first section of the user EEPROM. These parameters must be setup in your ladder program or in the debug menu when you are running the ProSoft ladder software. They do not get lost when the unit is powered off and so you only have to do the configuration once.

M113	Detect Sensitivity	X0.01%. - 2(0.02%) to 500(5%)
M114	Detect UnSens	X0.01%. - 2(0.02%) to 500(5%)
M115	Detect On Filter	X 10milliseconds.
M116	Detect Of Filter	X 10milliseconds.
M117	Pulse Time	X 10milliseconds.
M118	Pulse Mode	0=puls on det, 1=puls on undet.

M113 - Detect Sensitivity: The sensitivity determines the detect level of the loop. The smaller the value the higher the sensitivity. The value can be from 2 to 500 and is multiplied by 0.01%. Some typical values are shown below.

Value in M113	
2	0.02%
5	0.05%
10	0.1%
100	1.0%

M114 - Undetect Sensitivity: The undetect sensitivity determines the level at which the detector decides that the vehicle has left the loop. The smaller the value the higher the sensitivity. The value can be from 2 to 500 and is multiplied by 0.01%. This value must always be less that the value in M113 and is usually about a half of the value in M113.

M115 – The Detect On Filter is a delay on detect and is usually used to prevent the detector for giving false detect outputs. A minimum value of 10 is good for normal use but higher values can be used.

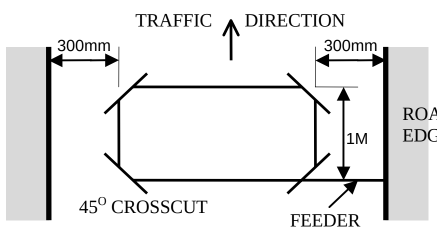
M116 – The Detect Off Filter is a delay on undetect and is usually used to extend the detect output after the vehicle has left the loop. A minimum value of 2 is good for normal use but higher values can be used.

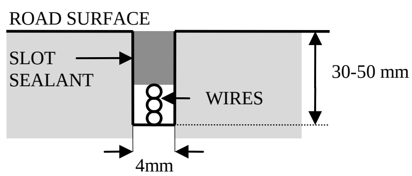
M117 – This value determines the pulse duration for the pulse bit.

M118 – The pulse mode is used to select pulse on detect or pulse on undetect.

5.3 Loop Installation Guide.

1. The loop and feeder should be made from insulated copper wire with a minimum cross-sectional area of 1.5mm^2. The feeder should be twisted with at least 20 turns per metre. Joints in the wire are not recommended and must be soldered and made waterproof. Faulty joints could lead to incorrect operation of the detector. Feeders which may pick up electrical noise should use screened cable, with the screen earthed at the detector.
2. The loop should be either square or rectangular in shape with a minimum distance of 1 metre between opposite sides. Normally 3 turns of wire are used in the loop. Large loops with a circumference of greater than 10 metres should use 2 turns while small loops with a circumference of less than 6 metres should use 4 turns. When two loops are used in close proximity to each other it is recommended that 3 turns are used in one and 4 turns in the other to prevent cross-talk.
3. Cross-talk is a term used to describe the interference between two adjacent loops. To avoid incorrect operation of the detector, the loops should be at least 2 metres apart and on different frequency settings.
4. For loop installation, slots should be cut in the road using a masonry cutting tool. A 45^o cut should be made across the corners to prevent damage to the wire on the corners. The slot should be about 4mm wide and 30mm to 50mm deep. Remember to extend the slot from one of the corners to the road-side to accommodate the feeder.
5. Best results are obtained when a single length of wire is used with no joints. This may be achieved by running the wire from the detector to the loop, around the loop for 3 turns and then back to the detector. The feeder portion of the wire is then twisted. Remember that twisting the feeder will shorten its length, so ensure a long enough feeder wire is used.
6. After the loop and feeder wires have been placed in the slot, the slot is filled with an epoxy compound or bitumen filler.





6. Modbus Memory Map (MODULE TYPE = 40)

The data in the PL400 is stored in registers. These registers are accessed over the network using the MODBUS RTU communication protocol.

There are 4 types of variables which can be accessed from the module. Each module has one or more of these data variables.

Type	Start Address	Variable
1	00001	Digital Outputs
2	10001	Digital Inputs
3	30001	Input registers (Analog)
4	40001	Output registers (Analog)

Modbus Address	Mem Addr	Register Name	Low Limit	High Limit	Access	Comments
10017	1.1	Digital Input 1	0	1	R	Status of Digital Inputs 1.
“	“	“	“	“	“	“
10032	1.16	Digital Input 16	0	1	R	Status of Digital Inputs 16.
00033	2.1	Digital Output 1	0	1	R/W	Status of Digital Outputs 1.
“	“	“	“	“	“	“
00048	2.16	Digital Output 16	0	1	R/W	Status of Digital Outputs 16.
00049	3.1	Timer 1	0	1	R/W	Status of Timer 1.
“	“	“	“	“	“	“
00064	3.16	Timer 16	0	1	R/W	Status of Timer 16.
00065	4.1	Counter 1	0	1	R/W	Status of Counter 1.
“	“	“	“	“	“	“
00080	4.16	Counter 16	0	1	R/W	Status of Counter 16.
00081	5.1	Control Relay 1	0	1	R/W	Status of Control relay 1.
“	“	“	“	“	“	“
00144	8.16	Control Relay 64	0	1	R/W	Status of Control relay 64.
00145	9.1	System Relay 1	0	1	R/W	Status of System relay 1.
“	“	“	“	“	“	“
00176	10.16	System Relay 32	0	1	R	Status of System relay 32.
30001	0	S/W Version / Module Type	N/A	N/A	R	High Byte = SoftwareVersion Low Byte = 40
30002	1	Digital Inputs	N/A	N/A	R	Digital Inputs in 16 bits.
40003	2	Digital Outputs	N/A	N/A	R/W	Digital Outputs in 16 bits.
40004	3	Timer Status	N/A	N/A	R/W	Timer Status.
40005	4	Counter Status	N/A	N/A	R/W	Counter Status.
40006	5	Control Relay	N/A	N/A	R/W	Control Relay
40007	6	Control Relay	N/A	N/A	R/W	Control Relay
40008	7	Control Relay	N/A	N/A	R/W	Control Relay
40009	8	Control Relay	N/A	N/A	R/W	Control Relay

40010	9	System Relay	N/A	N/A	R/W	System Relay
40011	10	System Relay	N/A	N/A	R/W	System Relay
-	11	-	N/A	N/A	-	Do not use – System only
-	12	-	N/A	N/A	-	Do not use – System only
40014	13	Timer 1 Value	0	65535	R/W	Timer range 0 to 65535.
"	"	"	"	"	"	"
40029	28	Timer 16 Value	0	65535	R/W	Timer range 0 to 65535.
40030	29	Counter 1 Value	0	65535	R/W	Counter range 0 to 65535.
"	"	"	"	"	"	"
40045	44	Counter 16 Value	0	65535	R/W	Counter range 0 to 65535.
40046	45	User Memory	0	65535	R/W	0 to 65535.
"	"	"	"	"	"	"
40113	112	User Memory	0	65535	R/W	0 to 65535.
40114	113	Detect Sensitivity	2	500	R/W	2(0.02%) to 500(5%) .
40115	114	Detect UnSens	2	500	R/W	2(0.02%) to 500(5%) .
40116	115	Detect On Filter	5	500	R/W	X 10milliseconds.
40117	116	Detect Of Filter	2	500	R/W	X 10milliseconds.
40118	117	Pulse Time	1	255	R/W	X 10milliseconds.
40119	118	Pulse Mode	0	1	R/W	0=puls on det, 1=puls on undet.
40120	119	Baud rate Progport	9600	19200	R/W	Default = 19200
40121	120	ID Prog port	0	255	R/W	Default = 1
40122	121	User EEPROM	0	65535	R/W	User EEPROM
"	"	"	"	"	"	"
40153	152	User EEPROM	0	65535	R/W	User EEPROM
40154	153	Seconds	0	59	R/W	RTC Seconds – Optional
40155	154	Minutes	0	59	R/W	RTC Minutes – Optional
40156	155	Hours	0	23	R/W	RTC Hours – Optional
40157	156	Day	1	7	R/W	RTC Day – Optional
40158	157	Date	1	31	R/W	RTC Date – Optional
40159	158	Month	1	12	R/W	RTC Month – Optional
40160	159	Year	0	100	R/W	RTC Year – Optional
40161	160	User BBRAM	0	65535	R/W	User BBRAM – Optional
"	"	"	"	"	"	"
40173	172	User BBRAM	0	65535	R/W	User BBRAM – Optional

7. Ladder Logic Function Blocks

The function blocks supported by the PL400 are listed below:

PL400 Function Blocks	
Function	**Function Block Description**
Timer 0.1Sec	Single input timer with 0.1 Second time base. The timer will run as long as the input is on. The timer will be reset to zero when the input is off.
Timer 0.01Sec	Single input timer with 0.01 Second time base. The timer will run as long as the input is on. The timer will be reset to zero when the input is off.
TimerA 0.1Sec	Accumulating timer with 0.1 Second time base. The timer will run as long as the input is on and stops when the input is removed. The timer will continue when the input is on again. The timer will be reset to zero when the reset input is on.
TimerA 0.01Sec	Accumulating timer with 0.01 Second time base. The timer will run as long as the input is on and stops when the input is removed. The timer will continue when the input is on again. The timer will be reset to zero when the reset input is on.
Counter	Up counter with reset input. The counter will count up when the count input goes from off to on. The counter will be reset to zero when the reset input is on. The counter output will go on when the count value is greater or equal to the preset value
Counter Up/Dn	Up/Down counter with reset input. The counter will count up when the Up count input goes from off to on. The counter will count down when the Down count input goes from off to on. The counter will be reset to zero when the reset input is on. The counter output will go on when the count value is greater or equal to the preset value.
NOP	This is a no operation function.
END	Placing this output function in the ladder program will indicate the end of the program. Any ladder after this function will not be run.
LD	Load the accumulator from memory(M) or with a constant(K).
LDD	The Load Double loads the accumulator with a 32 bit value from memory(M) or with a constant(K). The memory used is the two consecutive 16 bit memory locations, M & M+1.
OUT	Outputs the accumulator to memory(M).
OUTD	Outputs the 32 bit accumulator to two consecutive memory locations, M & M+1.
AND	AND the accumulator with memory(M) or with a constant(K).
ANDD	AND the 32 bit accumulator with memory(M) or with a constant(K). The memory used is the two consecutive 16 bit memory locations, M & M+1.
OR	OR the accumulator with memory(M) or with a constant(K).
ORD	OR the 32 bit accumulator with memory(M) or with a constant(K). The memory used is the two consecutive 16 bit memory locations, M & M+1.
XOR	Exclusive OR the accumulator with memory(M) or with a constant(K).
XORD	Exclusive OR the 32 bit accumulator with memory(M) or with a constant(K). The memory used is the two consecutive 16 bit memory locations, M & M+1.
CMP	Compare the accumulator lower 16 bits with memory(M) or with a constant(K). If the value in the accumulator is less than the value in memory/constant then system bit S6 is turned on. If the value in the accumulator is equal to the value in memory/constant then system bit S7 is turned on. If the value in the accumulator is greater than the value in memory/constant then system bit S8 is turned on.

PL400 Function Blocks	
Function	**Function Block Description**
CMPD	Compare the 32 bit accumulator with memory(M) or with a constant(K). If the value in the accumulator is less than the value in memory/constant then system bit S6 is turned on. If the value in the accumulator is equal to the value in memory/constant then system bit S7 is turned on. If the value in the accumulator is greater than the value in memory/constant then system bit S8 is turned on.
ADD	Add the memory(M) or constant(K) to the accumulator. The result is stored in the accumulator.
ADDD	Add the memory(M) or constant(K) to the 32 bit accumulator. The result is stored in the accumulator. The memory used is the two consecutive 16 bit memory locations, M & M+1.
SUB	Sub the memory(M) or constant(K) from the accumulator. The result is stored in the accumulator
SUBD	Sub the memory(M) or constant(K) from the 32 bit accumulator. The result is stored in the accumulator. The memory used is the two consecutive 16 bit memory locations, M & M+1.
MUL	Multiply the accumulator with the memory(M) or constant(K). The result is stored in the accumulator
MULD	Multiply the 32 bit accumulator with the memory(M) or constant(K). The result is stored in the accumulator. The memory used is the two consecutive 16 bit memory locations, M & M+1.
DIV	Divide the accumulator by the memory(M) or constant(K). The result is stored in the accumulator.
DIVD	Divide the 32 bit accumulator by the memory(M) or constant(K). The result is stored in the accumulator. The memory used is the two consecutive 16 bit memory locations, M & M+1.
INC	Increment the memory(M). The result is stored in the memory(M)
INCD	Increment two consecutive memory(M) locations. The result is stored in the memory M & M+1.
DEC	Decrement the memory(M). The result is stored in the memory (M).
DECD	Decrement two consecutive memory(M) locations. The result is stored in the memory M & M+1.
INV	Invert the bits in the accumulator
MOV	Moves a variable in a memory location to a new location. The accumulator must already contain the address of the memory location to be moved.
SHL	The bits in the accumulator are shifted left by the memory(M) or constant(K). The lower bits are filled with zeros.
SHR	The bits in the accumulator are shifted right by the memory(M) or constant(K). The upper bits are filled with zeros.
CALL	This function is used to call a subroutine. The constant(k) is the label of the subroutine.
SUBR	This function is the start of a subroutine. The constant(k) is the label of the subroutine which is called by the call function.
RET	This function must be placed at the last line of a subroutine. The function can also be used in the subroutine for a conditional return.
RAND	A random number from 0 to 100 is placed in the accumulator