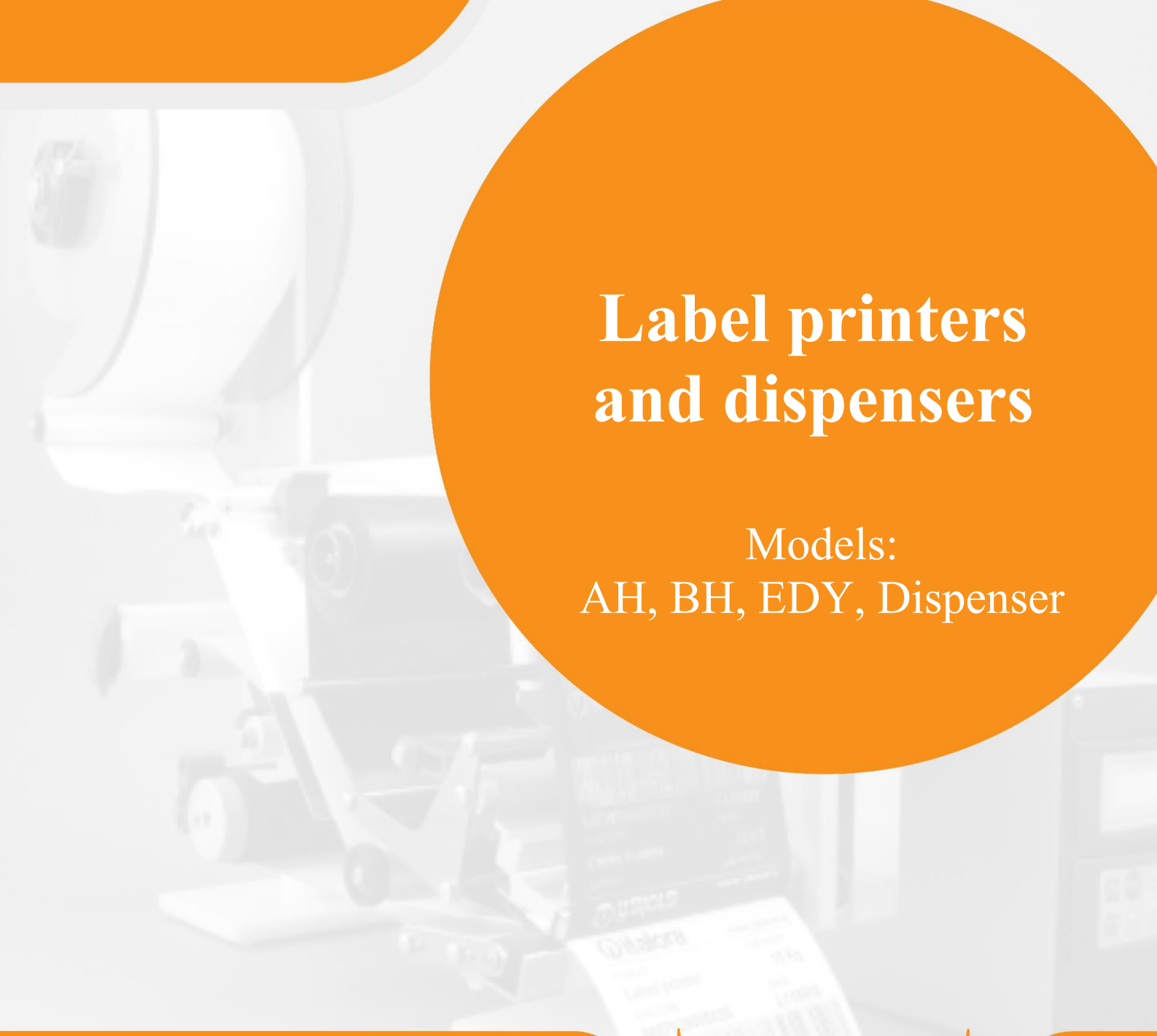


Label printers and dispensers

Models:
AH, BH, EDY, Dispenser

APPLICATORS

Hardware and programming manual

Rev.7

italora

Revisions Table

Version No.	Data	Revisions
Rev.0	01/10/2014	Hardware and programming manual for I/O MK3 expansion board
Rev.1	17/12/2019	New section: MAIN New commands MOTORSPEED, MOTOR1, MOTOR2, MASGALARM, RSTMSGALARM, IFCYCLE. New example
Rev. 2	09/10/2020	New commands: LOAD, TIMEOUT, IFAN
Rev. 3	18/11/2020	New command: WAITKEY
Rev. 4	19/06/2023	Graphics updated
Rev. 5	29/08/2025	New command: MOTORBASE New chapter: P&P interface details
Rev. 6	25/11/2025	Updated schematics
Rev. 7	09/09/2026	Updated commands MOTOR1 and MOTOR2 New command IFMOTOR Added V variables

SUMMARY

REVISIONS TABLE 2
INTRODUCTION 4
1 OPTOISOLATED I/O PICK&PLACE SIGNALS 5
OPTOISOLATED INPUT SIGNALS 5
OPTOISOLATED OUTPUT SIGNALS 5
2 CONNECTIONS WITH DB-25 CONNECTOR 6
3 INTERNAL JUMPERS SETTING 7
4 SOFTWARE PROGRAMMING 8
STRUCTURE OF PROGRAM FOR I/O EXPANSION BOARD SIGNALS 8
SECTIONS OF AN I/O SIGNALS PROGRAM 9
LIST AND SYNTAX OF COMMANDS FOR I/O EXPANSION BOARD SIGNALS 10
LIST AND SYNTAX OF COMMANDS TO DRIVE MEMORY VARIABLES 24
COMMAND TO ACTIVE I/O SIGNALS SCANNING CYCLE 26
5 EXAMPLE 27
EXAMPLE_1: CONTROL FLOW FOR AN APPLICATOR WITH 3 ELECTROVALVES AND 3 SENSORS 27
EXAMPLE_2: CONTROL FLOW FOR AN APPLICATOR WITH MOTOR AND ALARM MESSAGE 29
6 P&P INTERFACE DETAILS 33

Introduction

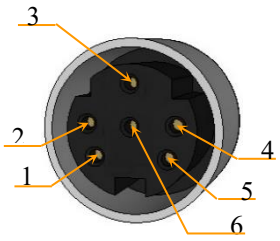
I/O expansion board allows you to expand the three I/O signals (startPrint, PrintEND and Alarm) to 32 optp-insulated lines (16IN + 16OUT); these 32 I/O lines are driver directly by printer/dispense main CPU. Input signal control and output signals driving may be programmed by user with a simple sequence of command similar to

```
DELAY,0,50
IF,2,D
ABORT
OUT,1,E
```

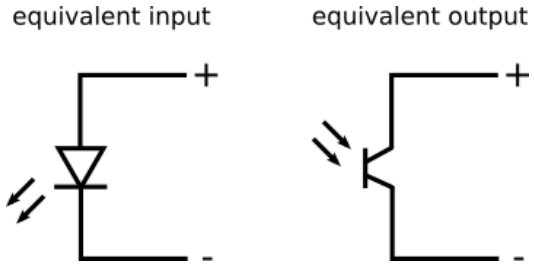
Detailed syntax of commands can be found in “software programming” chapter.

1 Optoisolated I/O Pick&Place signals

Minimum required I/O signals to synchronize printing system with external devices are located on 6 pins DIN connector.



6 poles DIN socket (external front view)



If the expansion board is equipped with 24V internal additional power supply pins 1, 3 and 5 are connected to + pole of diode or transistor; pins 2 and 6 are connected to GND and 24V power supply as shown in the underneath table.

1	START PRINT +
2	GND
3	ALARM +
4	-
5	PRINT END +
6	24V

If expansion board is NOT equipped with internal 24V pick&place connector signals connection is the same explained on main user and programming manual of you printer/dispenser model.

Refer to the “P&P interface details” chapter for detail regarding DIN3/4/5/6 connections.
When using expansion board with internal 24V power supply, refer to P&P +24V EXP section.

Optoisolated input signals

Input signals, 16 opto-insulated lines, have a photodiode that can be directly tied to 24Vdc signals.

Optoisolated output signals

Output signals, 16 opto-insulated lines, have a NPN Darlington transistor. They can drive up to 1,5A current when voltage is24Vdc.

2 Connections with DB-25 connector

Each I/O signal and its ground terminal is available on DB-25 male connectors and are divided as follow:

Connector CN1 – DB25 - M							
Pin	**Signal**	**Signal with internal 24Vdc**	**Notes**	**Pin**	**Signal**	**Signal with internal 24Vdc**	**Notes**
1	EXTIN1	EXTIN1	Input 1	14	EXTOUTC1	EXTOUTC1	Output 1, collector
2	EXTIN2	EXTIN2	Input 2	15	EXTOUTC7	EXTOUTC7	Output 7, collector
3	EXTIN3	EXTIN3	Input 3	16	EXTOUTC2	EXTOUTC2	Output 2, collector
4	EXTIN4	EXTIN4	Input 4	17	EXTOUTC8	EXTOUTC8	Output 8, collector
5	EXTIN5	EXTIN5	Input 5	18	EXTOUTC3	EXTOUTC3	Output 3, collector
6	EXTIN6	**INTPOWER**	Input 6 / 24Vdc (see JZ, jumpers table)	19	SENS1	SENS1	Input sensore 1
7	EXTIN7	EXTIN7	Input 7 (see JE, jumpers table)	20	EXTOUTC4	EXTOUTC4	Output 4, collector
8	EXTIN8	EXTIN8	Input 8 (see JF, jumpers table)	21	SENS2	SENS2	Input sensore 2
9	EXTGND	**GND**	**External/internal ground**	22	EXTOUTC5	EXTOUTC5	Output 5, collector
10	MOT2.1	MOT2.1	Stepper motor n.2, phase 1	23	MOT2.3	MOT2.3	Stepper motor n.2, fase 3
11	MOT2.2	MOT2.2	Stepper motor n.2, phase 2	24	EXTOUTC6	EXTOUTC6	Output 6, collector
12	SHIELD	SHIELD	Earth connection of shielded cable	25	MOT2.4	MOT2.4	Stepper motor n.2, fase 4
13	EXTGND	**GND**	**External/internal ground**				

Table 2.1: Connector CN1 - DB25 - M

Connector CN2 – DB25 - M							
Pin	**Signal**	**Signal with internal 24Vdc**	**Notes**	**Pin**	**Signal**	**Signal with internal 24Vdc**	**Notes**
1	EXTIN13	EXTIN13	Input 13	14	EXTOUTC11	EXTOUTC11	Output 11, collector
2	EXTIN14	EXTIN14	Input 14	15	EXTIN11	EXTIN11	Input 11
3	EXTIN15	EXTIN15	Input 15	16	EXTOUTC12	EXTOUTC12	Output 12, collector
4	EXTIN16	EXTIN16	Input 16 START PRINT	17	EXTIN12	EXTIN12	Input 12
5	EXTIN9	EXTIN9	Input 9	18	EXTOUTC13	EXTOUTC13	Output 13, collector
6	EXTIN10	EXTIN10	Input 10	19	EXTOUTC14	EXTOUTC14	Output 14, collector ALARM
7	-	-	(see JB, jumpers table)	20	EXTOUTC15	EXTOUTC15	Output 15, collector PRINT END
8	-	-	(see JC, jumpers table)	21	MOT3.2	MOT3.2	Stepper motor n.3, phase 2
9	EXTOUT9	EXTOUT9	Output 9, collector	22	EXTOUTC16	EXTOUTC16	Output 16, collector
10	MOT3.1	MOT3.1	Stepper motor n.3, phase 1	23	MOT3.3	MOT3.3	Stepper motor n.3, phase 3
11	EXTOUTC10	EXTOUTC10	Output 10, collector	24	MOT3.4	MOT3.4	Stepper motor n.3, phase 4
12	SHIELD	SHIELD	Earth connection of shielded cable	25	EXPPOWER	**INTPOWER**	Alimentazione esterna/interna +24Vdc
13	EXTGND	**GND**	**External/internal ground**				

Table 2.2: Connector CN2 - DB25 - M

3 Internal jumpers setting

I/O expansion board may be configured to work with internal printer/dispenser power supply, internal additional power supply or external power supply.
Each output signal and EXTGND signal of CN1 DB-25 may be tied to internal ground by jumpers setting. Photodiodes of input signals may be powered with 5Vdc or 24Vdc by setting of JR jumper (Table 3.1).



If you are using internal printer/dispenser power supply and ground, optoinsulation is no longer effective: so noise immunity of overall system (printer/dispenser and applicator) may be compromised.



If the expansion board is installed in a printer/dispenser with applicator device an additional internal 24Vdc power supply is present for GND and INTPOWER signals. Power supply is therefore internal to printer but anyway CPU optoinsulation is assured.

Jumper	Setting	Signal	Notes
JQ	On	GND	Printer/dispenser internal ground
JR	1-2 = 24V internal	INTPOWER	Printer/dispenser power supply
JR	2-3 = 5V internal	INTPOWER	Printer/dispenser power supply
JQ	Off	GND	Additional internal ground
JR	Off	INTPOWER	Additional internal power supply
JQ	Off	EXTGND	External ground
JR	Off	EXPPOWER	External power supply

Table 3.1: jumpers description of MK£ expansion board (continue table 3.2)

There are jumpers to set some I/O expansion boards options, they change the behaviour of some pins.

Jumper	Settng	Signal	Connector and pin
JA	Off	-	Y23.2
JA	On	Encoder	Y23.2
JB	Off	-	CN2.7
JB	On	GND printer/dispencer	CN2.7
JC	Off	-	CN2.8
JC	On	+5V printer/dispencer	CN2.8
JE	1-2	EXTIN7	CN1.7
JE	2-3	+5V printer/dispencer	CN1.7
JF	1-2	EXTIN8	CN1.8
JF	2-3	GND printer/dispencer	CN1.8
JZ	Off	EXTIN6	CN1.6
JZ	On	EXPPOWER	CN1.6

Table 3.2: jumpers description of MK3 board.

4 Software programming

Structure of program for I/O expansion board signals

I/O expansion board signals are verified by printer during next situations

- At printer power on
- Before beginning cycle
- Before beginning printing
- After finishing printing
- During alarms end of paper or end of ribbon
- During alarms reset operation

Programming commands for I/O expansion board signals must be included into previous sections (INIT, BEFORE, AFTER, ALARM and ALARMRESET). The structure of programming commands for I/O expansion board signals must be the one shown in the next table:

?B0&1	To begin aplicator INIT cycle programming. Next commands refer to INIT section
…	Command referring to INITsection
ENDINIT	It ends programming of INIT section. Next commands refer to BEFORE section.
…	Command referiing to INITMAIN
ENDMAIN	It ends programming of MAIN. Next command refer to ENDMAIN section.
…	Command refering to BEFORE
ENDBEFORE	It ends programming of BEFORE section. Next commands refer to AFTER section.
…	Command refering to AFTER
ENDAFTER	It ends programming of AFTER section. Next commands refer to ALARM section.
…	Command refering to ALARM
ENDALARM	It ends programming of ALARM section. Next commands refer to ALARMRESET section
…	Command refering to ALARMRESET
ENDALARMRESET	It ends programming of ALARMRESET section and of all I/O expansion board signals.

Table 4.1: Structure of programming command I/O expansion board

Sections of an I/O signals program

INIT section

All commands contained into INIT section are executed at power on; usually they are used to set a starting initialisation for outputs.

INIT section begins immediately after ?B0&1 command and is ended by ENDINIT command. When it shouldn't be necessary to run any initializing command you must specify ENDINIT into the line immediately after ?B0&1 command.

MAIN section

This section contains a series of commands that can be used, for example, to perform calibration operations on external devices. The advantage of using this section is that the instructions contained within it will be executed without necessarily having to wait for a 'print' signal and can be executed an unlimited number of times. Allowing the calibration of any external devices controlled by the applicator cycle.

The MAIN section begins immediately after the ENDINIT command. If calibration is not required, simply specify ENDMAIN on the line following the ENDINIT command.

BEFORE section

All commands contained into BEFORE section are executed just before printing operations start.

BEFORE section begins immediately after ENDINIT command and is ended by ENDBEFORE command. When it shouldn't be necessary to run any command before label ejection you must specify ENDBEFORE into the line immediately after ENDINIT command.

AFTER section

All commands contained into AFTER section are executed just after printing operations stop.

AFTER section begins immediately after ENDBEFORE command and is ended by ENDAFTER command. When it shouldn't be necessary to run any command after label ejection you must specify ENDAFTER into the line immediately after ENDBEFORE command.

ALARM section

All commands contained into ALARM section are executed each time printer is in an alarm state, such as end of paper or end of ribbon state.

ALARM section begins immediately after ENDAFTER command and is ended by ENDALARM command. When it shouldn't be necessary to run any command during alarm situations you must specify ENDALARM into the line immediately after ENDAFTER command.

ALARMRESET section

All commands contained into ALARMRESET section are executed when an alarm condition is reset.

ALARMRESET section begins immediately after ENDALARM command and is ended by ENDALARMRESET command. When it shouldn't be necessary to run any command during alarm reset operation you must specify ENDALARMRESET into the line immediately after ENDALARM command.

List and syntax of commands for I/O expansion board signals

Commands for programming of expansion board I/O signals are the next:

- OUT
- WAIT
- IF
- GOTO
- LABEL
- DELAY
- COUNT
- EXIT
- ABORT
- SEND
- PRINT
- SET
- COMP
- INC
- DEC
- MOD
- SLEEP
- TX
- MOTORSPEED
- MOTORBASE
- MOTOR1
- MOTOR2
- IFMOTOR
- MSGALARM
- RSTMSGALARM
- IFCYCLE
- LOAD
- TIMEOUT
- IFAN
- WAITKEY

Each of them, when used, must be written in UPPERCASE letters.

OUT command: command to enable/disable output signals

OUT,n,s		
n	port number	1…16
s	status	E = enable D = disable

WAIT command: command to control input signals. Flow control waits until s status is verified on input channel **n**.

WAIT,n,s		
n	output number	1…16
s	status	E = enable D = disable

IF command: command to control input signals. Flow control executes instruction line immediately after IF command if s status of input channel **n** is verified, otherwise it will execute command write two lines after IF instruction.

IF,n,s		
n	input channel	1…16
s	status	E = enable D = disable

In this example if input channel 3 is enabled, line *GOTO,*5 is executed else *EXIT* line is executed.

.....

IF,3,E

GOTO,5

EXIT

LABEL,5

.....

GOTO command: unconditioned jump command. It executes a jump to specified label **label**. It needs a corresponding *LABEL,n* instruction for which **n = label.**

GOTO,yyy		
yyy	index	0…255

In this example if input channel 3 is enabled *GOTO,*5 is executed else *EXIT* is executed.

.....

IF,3,E

GOTO,5

EXIT

LABEL,5

.....

LABEL command: program label. It marks a point in program flow for a GOTO jump.

LABEL,n		
n	label index	0…255

In this example if input channel 3 is enabled *GOTO,5* is executed else *EXIT* is executed.

.....

IF,3,E

GOTO,5

EXIT

LABEL,5

.....

DELY command: command for a defined delay. It stops program flow execution for a time equivalent to seconds and hundredth of second. It supports V variables (refer to the programming manual, command ?A2&44).

DELAY,sec,cent	
sec	seconds
cent	hundredth os seconds

COUNT command: command to count transitions into Input signals. It waits ct times an on/off transition on input channel **n**.

COUNT,n,ct	
n	input
ct	cysles

EXIT command: command to terminate application cycle. It exit program flow control from any point after a delay equivalent to seconds and hundredth of seconds.

EXIT,sec,cent	
sec	seconds
cent	hundredth os seconds

ABORT command: command to terminate application cycle. It exits program flow control from any point and stops printing operations of current label.

ABORT no parameters

SEND command: This command transmits a string to serial port. Strings can be up to three characters long. If they are shortest than three characters, blanks will be added by printer to complete the length.

SEND,sss	
sss	string to be sent

PRINT command: Command to print another label. It allows to enable / disable printing of another label after the end of running application cycle.

PRINT,s		
s	status	E = enable D = disable

WAITKEY command: this instruction allows to wait for the Esc to be pressed and then released before proceeding to execute the next commands of the applicator cycle.

WAITKEY,n,s		
l	button number	1 (ESK key)
s	status to be detected	E

MOTORSPEED command: command to set the speed of the motors connected to the outputs 'Stepper motor n.2' and 'Stepper motor n.3' respectively on CN1 and CN2 (see tables 2.1 and 2.2). This instruction can be entered only once within the programming of the applicator cycle, it is not necessary to define it every time you want to control the motor. If, on the other hand, it is desired, within the applicator cycle sequence, to command the motor with different speeds, it will be necessary to redefine the rotation speed before sending the command to make the motor move (MOTORn).

MOTORSPEED,n,speed		
n	motor number	1 = stepper motor n.2 (CN1) 2 = stepper motor n.3 (CN2)
speed	motor speed	steps/second

The following example sets the rotation speed for motor number 1 to 100 steps / second.

……..

MOTORSPEED,1,100

…….

MOTORBASE command: command to set the starting speed of the acceleration ramp of the motors connected to the outputs 'Stepper motor n.2' and 'Stepper motor n.3' respectively on CN1 and CN2 (see tables 2.1 and 2.2).

If the starting speed of the acceleration ramp is set to a value that is less or equal than the final motor speed (set with MOTORSPEED command), no acceleration will take place when starting the motor.

This instruction can be entered only once within the programming of the applicator cycle, it is not necessary to define it every time you want to control the motor. If, on the other hand, it is desired, within the applicator cycle sequence, to command the motor with different accelerations, it will be necessary to redefine the starting speed before sending the command to make the motor move (MOTORn).

MOTORBASE,n,speed		
n	motor number	1 = stepper motor n.2 (CN1) 2 = stepper motor n.3 (CN2)
speed	motor starting speed	steps/second

The following example sets the starting speed of the acceleration ramp for motor number 1 to 100 steps / second.

……..

MOTORBASE,1,100

…….

MOTOR1 command: command to move the motor connected to the 'Stepper motor n.2' output on CN1. It supports V variables (refer to the programming manual, command ?A2&44).

MOTOR1,dir,steps		
dir	direction of the rotation	F = clockwise rotation FA = clockwise rotation, asynchronous B = counterclockwise rotation BA = counterclockwise rotation, asynchronous S = stop with standby current H = stop with working current D = stop with stepper disabled
steps	number of steps	number of steps the motor must perform NOTE: the motor will stop with working current once the set number of steps are peformed. NOTE: if the number of steps is set to 0, the motor will run indefinitely until it receives a stop command.

The sequence shown below allows you to set the rotation speed of motor 1, the direction of rotation and the number of steps.

If input 1 is enabled (IF,1,E) the GOTO,1 instruction is executed. The following are then configured: the starting speed of the acceleration ramp of motor 1 is equal to 80steps/second (MOTORBASE,1,80), the speed of motor 1 is equal to 100steps/second (MOTORSPEED,1,100), the direction of rotation is clockwise and the number of steps of the motor is 110 (MOTOR1,F,110). The motor will stay in this conditions until input 4 is disabled (WAIT,4,D). After a delay of 0,5 seconds (DELAY,0,5) the system still waiting for an enabling signal on input 4 (WAIT, 4, E) after the motor is brought back to its initial position, as it is rotated in the counter-clockwise direction by 110 steps (MOTOR1, B, 110) . The cycle then continues by executing the instructions DELAY, 0,5 and LABEL, 2. If input 1 is disabled, the GOTO,2 jump instruction is executed and then passes to the LABEL,2 instruction, 2.

……..

IF,1,E

GOTO,1

GOTO,2

LABEL,1

MOTORBASE,1,80

MOTORSPEED,1,100

MOTOR1,F,110

WAIT,4,D

DELAY,0,5

WAIT,4,E

MOTOR1,B,110

DELAY,0,5

LABEL,2

……..

MOTOR2 command: command to move the motor connected to the 'Stepper motor n.3' output on CN2. It supports V variables (refer to the programming manual, command ?A2&44).

MOTOR2,dir,steps		
dir	direction of the rotation	F = clockwise rotation FA = clockwise rotation, asynchronous B = counterclockwise rotation BA = counterclockwise rotation, asynchronous S = stop with standby current H = stop with working current D = stop with stepper disabled
steps	number of steps	number of steps the motor must perform NOTE: the motor will stop with working current once the set number of steps are peformed. NOTE: if the number of steps is set to 0, the motor will run indefinitely until it receives a stop command.

The description of the operation is similar to that of MOTOR1, see example given for the MOTOR1 command.

IFMOTOR command: command to check motor status. The instruction immediately after the IFMOTOR command is executed if the **s** status of the **n** motor is verified, otherwise the instruction two lines after the IFMOTOR instruction will be executed.

IFMOTOR,n,s		
n	Motor number	1…2
s	status	E = enable, moving D = disable, stopped

In this example if the motor 1 is moving, line *GOTO,*5 is executed else *EXIT* line is executed.

.....

IFMOTOR,1,E

GOTO,5

EXIT

LABEL,5

.....

MSGALARM command: command to send an alarm message to the printer display, in the event that an anomaly occurs during the execution of the applicator cycle.

MSGALARM,xxx,msg		
xxx	index of the message	A…Z
msg	message to be shown on the display	

In the following example, if input 1 is active, the GOTO,1 instruction is executed and the 'ALARM no label' alarm message will be displayed. If, on the other hand, input 1 is not active, the GOTO,2 instruction is executed, and the part of the code relating to LABEL,2 will be executed.

.....

IF,1,E

GOTO,1

GOTO,2

LABEL,1

MSGALARM,A,ALARM no label

LABEL,2

.....

RSTALARM command: command to delete the alarm message that has been sent to the display. The user has the ability to delete the alarm message that appears on the display when the cause of the problem has been identified and resolved.

RSTMSGALARM no parameter

In the following example, if input 1 is enabled, the GOTO,1 instruction is executed and the 'ALARM no label' alarm message is displayed. After a delay of 0.5 seconds, the system waits (WAIT, 1, D) until input 1 is deactivated (deactivation indicates that the problem has been identified and solved) the alarm is cancelled (instruction RSTMSGALARM) and the display returns to normal operating conditions.

If, on the other hand, input 1 is not active, the GOTO,2 instruction is executed, and the part of the code relating to the LABEL, 2 is executed.

.....

IF,1,E

GOTO,1

GOTO,2

LABEL,1

MSGALARM,A,ALARM no label

DELAY,0,5

WAIT,1,D

RSTMSGALARM

ABORT

LABEL,2

WAIT,1,D

DELAY,0,5

.....

IFCYCLE command: command to control a signal managed by the printer control panel. To enable / disable the signal, enter the 'SPECIAL OPTIONS' menu and navigate until you find the 'IFCYCLE state' menu. At this point, using the UP/DOWN arrows it will be possible to select the signal status.

If the signal is set to ENABLE the program executes the instruction line following the IFCYCLE, if it is set to DISABLE it executes the instruction line two lines below the IFCYCLE.

```
IFCYCLE,A
```

If the ENABLE signal is set from the menu, the GOTO,1 instruction will be executed (then it will jump to LABEL,1), if instead it is set as DISABLE, the GOTO,2 instruction will be executed (then it will jump to LABEL,2)

```
…….
IFCYCLE,1,E
GOTO,1
GOTO,2
LABEL,1
EXIT
LABEL,2
OUT,1,D
……
```

LOAD command: command to recall automatically the label layouts previously stored in the printer from the applicator cycle.

LOAD,x		
x	letter corresponding to the layout to be loaded	A…Z

By appropriately using the various programming instructions of the applicator cycle, the previously stored layouts can be automatically loaded to the printer and printed after receiving an external Start Print signal.

Load and print Layout A

```
.....
ENDMAIN
COMP,Z,0
GOTO,10
GOTO,11
LABEL,10
LOAD,A
SET,Z,1
GOTO,99
LABEL,11
COMP,Z,1
GOTO,12
GOTO,13
LABEL,12
LOAD,B
SET,Z,2
GOTO,99
LABEL,13
LOAD,C
SET,Z,0
LABEL,99
ENDBEFORE
ENDAFTER
ENDALARM
ENDALARMRESET
.....
```

When turned on, the variable Z is set to 0, so the comparison (COMP, Z, 0) will give a positive result and the GOTO,10 instruction will then be executed.

The loop will jump to execute LABEL,10 where it will load layout A (LOAD,A), will set the variable Z (SET,Z,1) and will execute the GOTO, 99 instruction.

The cycle ends by jumping to the LABEL, 99 instruction with the printing of the label with layout A.

Load and print Layout B

```
.....
ENDMAIN
COMP,Z,0
GOTO,10
GOTO,11
LABEL,10
LOAD,A
SET,Z,1
GOTO,99
LABEL,11
COMP,Z,1
GOTO,12
GOTO,13
LABEL,12
LOAD,B
SET,Z,2
GOTO,99
LABEL,13
LOAD,C
SET,Z,0
LABEL,99
ENDBEFORE
ENDAFTER
ENDALARM
ENDALARMRESET
.....
```

At the second execution, the variable Z will have the value 1, so the comparison (COMP,Z,0) will give a negative result and the GOTO,11 instruction will then be executed.

The cycle will jump to execute LABEL,11, will perform the comparison (COMP,Z,1) which will give a positive result and will lead to the execution of the GOTO,12 instruction.

The jump to LABEL, 12 will allow you to load layout B, will set the variable Z to 2 (SET, Z, 2) and will execute the GOTO, 99 instruction.

The cycle ends by jumping to the LABEL,99 instruction with the printing of the label with layout B.

Load and print Layout C

.....

ENDMAIN

COMP,Z,0

GOTO,10

GOTO,11

LABEL,10

LOAD,A

SET,Z,1

GOTO,99

In the third execution, the variable Z will have the value 2, so the comparison (COMP,Z,0) will give a negative result and the GOTO,11 instruction will then be executed.

LABEL,11

COMP,Z,1

GOTO,12

GOTO,13

LABEL,12

LOAD,B

SET,Z,2

GOTO,99

The cycle will jump to execute LABEL,11, will perform the comparison (COMP,Z,1) which will give a negative result and will lead to the execution of the GOTO,13 instruction.

LABEL,13

LOAD,C

SET,Z,0

LABEL,99

ENDBEFORE

ENDAFTER

ENDALARM

ENDALARMRESET

.....

The jump to LABEL,13 will allow you to load layout C, set variable Z to 0 (SET,Z,0) and print the label with layout C.
When the cycle restarts, layout A will be loaded again and will continue as described in the previous points with layouts B and C

TIMEOUT command: the TIMEOUT command is linked to a WAIT instruction, if a WAIT instruction is not entered the TIMEOUT command has no meaning. In the TIMEOUT instruction a time is set within which, if no transition is detected on the channel indicated by the WAIT, the cycle is interrupted.

TIMEOUT,S,C		
S	second	0…99
C	hundredth of seconds	0...999

In the example below, a control of 2 seconds is set (TIMEOUT,2,0) and verifies that, within these 2 seconds, the signal on input 16 goes ENABLE. If input 16 remains DISABLE, the TIMEOUT starts running, ending the execution of the cycle; if, on the other hand, input 16 becomes ENABLE, the cycle continues its normal execution and will proceed by executing the instruction OUT,1, E.

……

ENDMAIN

TIMEOUT,2,0

WAIT,16,E

OUT,1,E

ENDBEFORE

……..

IFAN command: this instruction allows to read the status of the analog inputs wired on the EL7 board and to compare them with a threshold. Depending on the outcome of the comparison, jumps can be made within the instructions of the applicator cycle.

IFAN,Ch,C,S		
Ch	analog input channel	0….7 0: NTC 1: trimmer contrast 2: photo label 3: photo thaken 4: photo ribbon 5: photo reflection 6: start print 7: photo pre-end label
C	condizione di comparazione	<, >
S	soglia da comparare	0….255

In the example below, if the value detected by analog input number 4 (the input is the one wired on the EL7 board) is greater than 20 (true condition) the GOTO,1 instruction will be executed, if the condition is false (<20) GOTO,2 instruction will be executed.

……

ENDMAIN

IFAN,4,>,20

GOTO,1

GOTO,2

LABEL,1

DELAY,3,0

LABEL,2

DELAY,5,0

ENDBEFORE

……..

List and syntax of commands to drive memory variables

It has been defined a commands set to drive up to 26 numeric variables in printer volatile memory. Variables names are already defined and named with 26 UPPERCASE letters of alphabet, from 'A' to 'Z'.

SET command: command to set a value in a memory variable.

This command to not directly acts on input/output signals but it allows to drive 26 numeric variables used by programmer.

SET,n,v		
n	variable name	A…Z
v	value to set	0…65535

COMP command: Command to test numeric variables. Next line under COMP instruction is executed if variable value is equal to **v** value, if it is not equal instruction 2 lines under COMP command is executed.

COMP,n,v		
n	variable name	A…Z
v	value to verify	0…65535

In this sample if C è = 1025 line "GOTO,5" is executed to jump to LABEL,5 else "EXIT" line is executed.

.....

COMP,C,1025

GOTO,5

EXIT

LABEL,5

.....

INC command: command to increment value stored in memory variables. This command adds 1 to specified memory variable.

INC,n		
n	variable name	A…Z

DEC command: command to increment value stored in memory variables. this command adds 1 to specified memory variable.

DEC,n

n	variable name	A…Z

MOD command: Command to get reminder of division for memory numeric variables. This command allows to store in specified memory variable the reminder of integer division **n/v.**

MOD,n,v		
n	variable name	A…Z
v	divisor	1…65535

example:

SET,A,12

MOD,A,3

stores reminder of 12/3 (namely 0) into A variable

SET,A,11

MOD,A,3

stores reminder of 11/3 (namely 2) into A variable

SLEEP command: command for a delay based on a memory variable. It stops program flow execution for a number of milliseconds stored in specified memory variable.

SLEEP,n		
n	variable name	A…Z

TX command: this command transmits numeric contents of specified memory variable to serial port.

TX,n		
n	variable name	A…Z

example

SET,Z,1999

TX,Z

transmits string “1999” to serial port.

Command to active I/O signals scanning cycle

To enable instructions programmed into all sections, it's necessary to activate the I/O signals scanning cycle.

?A2&7,x	
x = 0	scanning cycle disable
x = 1	scanning cycle enable

Settings programmed with this command is stored into permanent memory by printer also during power off.

5 Example

Example_1: Control flow for an applicator with 3 electrovalves and 3 sensors

Signals:

input 1 = piston up sensor

input 2 = piston down sensor

input 3 = label presence sensor

output 1 = applicator puff

output 2 = piston enable

output 3 = label suction

Program sequence

```
‘enable I/O signals scan
?A2&7,1
‘start programming
?B0&1
‘
‘I/O init programming
OUT,1,D
OUT,2,D
OUT,3,D
‘end of I/O init programming
ENDINIT
‘-----------------------------------------------------------------------------------------------------------
‘end of MAIN cycle
ENDMAIN
‘-----------------------------------------------------------------------------------------------------------
‘I/O signal control flow BEFORE printing programming
‘----suction
OUT,3,E
‘end of I/O signal control flow BEFORE printing programming
ENDBEFORE
‘-----------------------------------------------------------------------------------------------------------
‘I/O signal control flow AFTER printing programming
‘wait label under applicator
DELAY,0,50
‘if no label ABORT application (and eject another label)
```

```
IF,3,D
GOTO,3
GOTO,4
‘abort mark point
LABEL,3
‘---- stop suction
OUT,3,D
ABORT
‘
‘---- application cycle
LABEL,4
‘---- wait piston up
WAIT,1,E
DELAY,0,5
‘---- piston enable
OUT,2,E
DELAY,0,5
‘---- wait piston down
WAIT,2,E
DELAY,0,5
‘---- stop suction
OUT,3,D
‘---- enable puff
OUT,1,E
‘---- piston off
OUT,2,D
DELAY,0,5
‘---- puff off
OUT,1,D
DELAY,0,10
‘end of I/O signal control flow AFTER printing programming
ENDAFTER
‘-----------------------------------------------------------------------------------------------------------
‘no command into ALARM section
ENDALARM
‘-----------------------------------------------------------------------------------------------------------
‘no command into ALARMRESET section
ENDALARMRESET
```

Example_2: control flow for an applicator with motor and alarm message

Signals:

input 1 = label sensor

input 2 = home position roll sensor

input 3 = motor 1 home sensor

input 4 = button to start the calibration procedure

output 1 = label suction fan

Program sequence

```
‘enable I/O signals scan
?A2&7,1
‘start programming
?B0&1
‘
‘I/O init programming
OUT,1,D
OUT,2,D
OUT,3,D
‘end of I/O init programming
ENDINIT
‘-----------------------------------------------------------------------------------------------------------------
‘calibration instruction in the MAIN section
‘---- if the button (input 4) is pressed LABEL,30 instructions will be executed, otherwise the flow
‘---- jumps to LABEL,31
IF,4,E
GOTO,30
GOTO,31
LABEL,30
‘---- motor 1 starting speed of acceleration ramp set to 80 steps/second
MOTORBASE,1,80
‘---- motor 1 speed setting to 100 steps/second
MOTORSPEED,1,100
‘----motor 1: 110 steps clockwise rotations
MOTOR1,F,110
‘---- the system waits a pression on button (input 4)
WAIT,4,D
DELAY,0,5
‘---- the system waits up to the button released
```

```
WAIT,4,E
‘---- motor 1 in the previous position. Counterclockwise rotation, 110 steps.
MOTOR1,B,110
DELAY,0,5
‘end of MAIN cycle
ENDMAIN
‘-----------------------------------------------------------------------------------------------------------------
LABEL,31
‘The system checks that there are no problems before to start printing. The system will check the correct state of the sensors.
‘input 1: disable, no label under the sensor before to start the printing
'input 2: enable, the roll in the home position
‘input 3: disable, the motor is in home position
‘
‘---- check that no label are presents under the sensor before to start printing
IF,1,E
GOTO,20
GOTO,21
‘---- a label is present under the sensor
LABEL,20
‘---- alarm message
MSGALARM,A,ALARM pht label
DELAY,0,5
‘---- the system waits until the label is removed
WAIT,1,D
‘---- delete alarm message
RSTMSGALARM
‘---- aborted cycle. The flow control will restart
ABORT
‘---- no label under the sensor. The system will continue with the checks
LABEL,21
WAIT,1,D
DELAY,0,5
‘…….
‘for compactness, the instructions for controlling inputs 2 and 3 are not shown as they are the same
‘as the instructions for input 1.
‘
‘----motor 1: starting speed of acceleration ramp to 80 steps/second
MOTORBASE,1,80
‘----motor 1: speed settings to 80 steps/second
```

```
MOTORSPEED,1,80
'----enable fan
OUT,1,E
‘end of I/O signal control flow BEFORE printing programming
ENDBEFORE
‘-----------------------------------------------------------------------------------------------------------------
‘I/O settings after the printing
‘---- after the printing the system will wait 2 seconds. Waiting time for label aspiration
DELAY,2,0
‘---- check label sensor
IF,1,D
GOTO,1
GOTO,2
‘---- if after the printing no label under the sensor an alarm will show a message
LABEL,1
‘---- disable fan
OUT,1,D
‘---- alarm message on the display
MSGALARM,D,ALARM no label
‘----the system waits for a check on the label presence sensor
WAIT,1,E
‘---- alarm message reset
RSTMSGALARM
‘---- exit
EXIT,0,0
‘---- a label is printed
LABEL,2
‘---- the system is waiting a START APPLY signal
WAIT,16,E
‘---- motor 1: clockwise rotation: 100 steps
MOTOR1,F,110
DELAY,3,0
‘---- check roll up
IF,2,E
GOTO,3
GOTO,4
‘---- if the roll is down a message will be generated
LABEL,3
‘---- motor 1 in home position
```

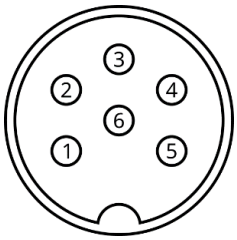
```
MOTOR1,B,110
MOTOR1,S,0
‘----alarm message on the display
MSGALARM,E,ALARM no pack
‘---- wait up to the label is removed
WAIT,1,D
‘---- reset alarm message
RSTMSGALARM
‘---- fan disable
OUT,1,D
‘---- exit
EXIT,0,0
‘---- the roll is up
LABEL,4
DELAY,2,0
‘….
‘for compactness no other instruction will be added.
‘
‘end of I/O signal control flow AFTER printing programming
ENDAFTER
‘-----------------------------------------------------------------------------------------------------------------
‘no command into ALARM section
ENDALARM
‘-----------------------------------------------------------------------------------------------------------------
‘no command into ALARMRESET section
ENDALARMRESET
```

6 P&P interface details

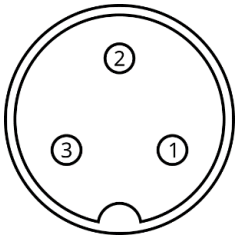
P&P +24V EXP

DIN PLUG VIEW, SOLDERING SIDE

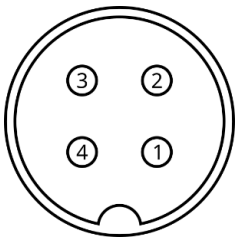
Control units with expansion board - internal power supply and ground



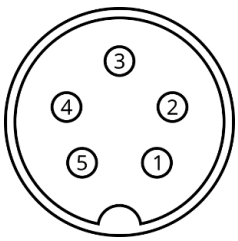
1	+ START PRINT (INPUT)
2	GND
3	+ ALARM (OUTPUT)
4	GND
5	+ PRINT END (OUTPUT)
6	+24V



1	GND
2	+24V
3	+ START PRINT (INPUT)



1	GND
2	+24V
3	+ PRINT END (OUTPUT)
4	NOT CONNECTED

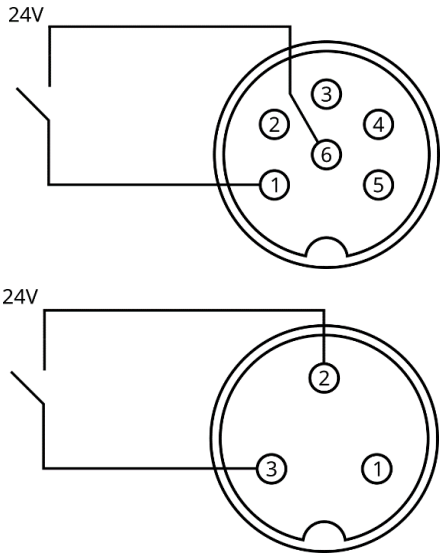


1	GND
2	+24V
3	+ ALARM (OUTPUT)
4	NOT CONNECTED
5	NOT CONNECTED

DIN PLUG VIEW, SOLDERING SIDE

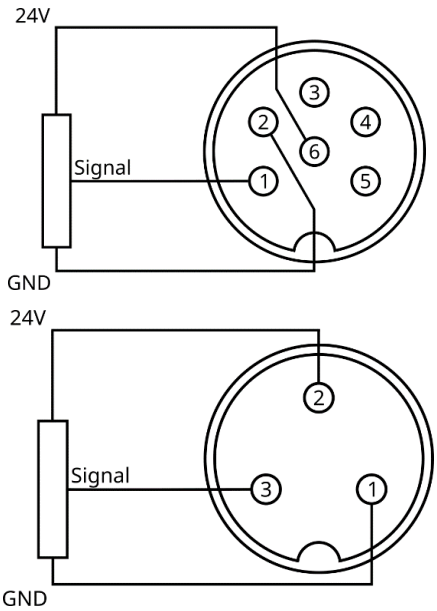
INPUT SIGNAL

START PRINT - Switch



NOTE: the internal circuit resistance is 1800Ω

START PRINT - PNP photocell

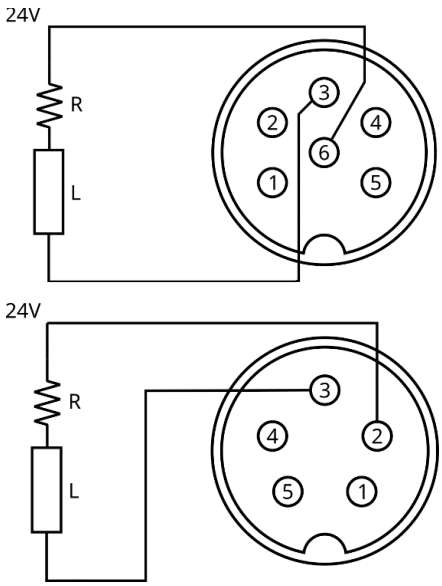


NOTE: the internal circuit resistance is 1800Ω

DIN PLUG VIEW, SOLDERING SIDE

OUTPUT SIGNALS

ALARM



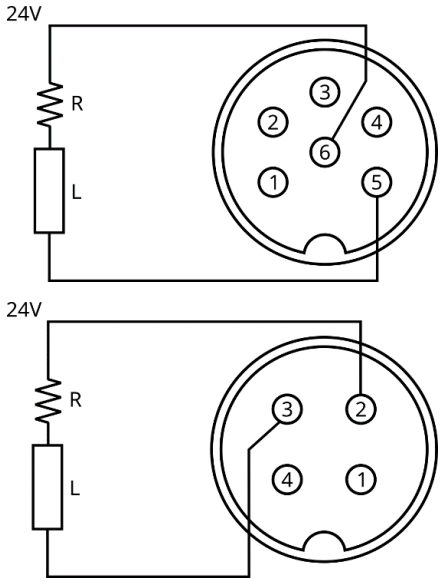
V [V]	I [mA]	R+L [Ω]
24	10	2400
24	**20**	**1200**
24	50	240

V = internal power supply
I = generated current
R = current limiting resistor
L = load resistance of the external circuit

NOTE: the highlighted row indicates the recommended values

NOTE: the maximum output current is 1A

PRINT END



V [V]	I [mA]	R+L [Ω]
24	10	2400
24	**20**	**1200**
24	50	240

V = internal power supply
I = generated current
R = current limiting resistor
L = load resistance of the external circuit

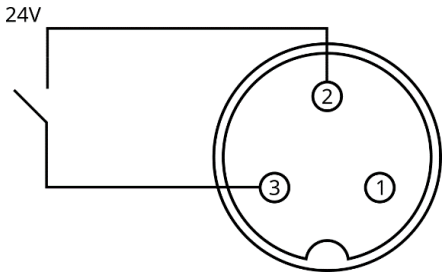
NOTE: the highlighted row indicates the recommended values

NOTE: the maximum output current is 1A

DIN PLUG VIEW, SOLDERING SIDE

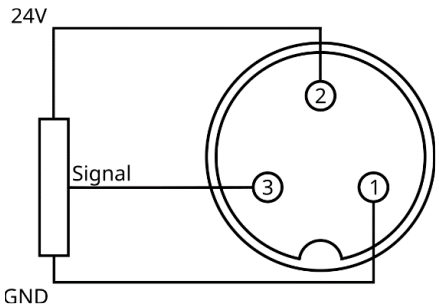
INPUT SIGNAL

START PRINT - Switch



NOTE: the internal circuit resistance is 1800Ω

START PRINT - PNP photocell

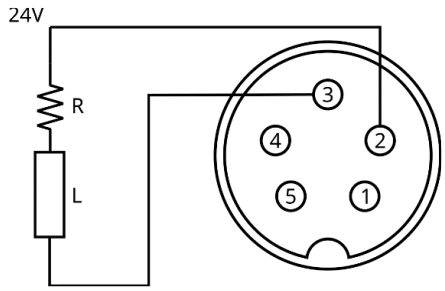


NOTE: the internal circuit resistance is 1800Ω

DIN PLUG VIEW, SOLDERING SIDE

OUTPUT SIGNALS

ALARM



V [V]	I [mA]	R+L [Ω]
24	10	2400
24	**20**	**1200**
24	50	240

V = internal power supply
I = generated current
R = current limiting resistor
L = load resistance of the external circuit

NOTE: the highlighted row indicates the recommended values

NOTE: the maximum output current is 1A

PRINT END

