

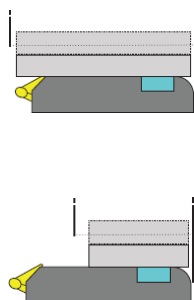
EMV ccTalk electronic vending motors

Operator's manual

Rev. 1.02

The Alberici vending motors have been designed for dispensing packed products.

EP-1POC TRAS-101P CCTALK

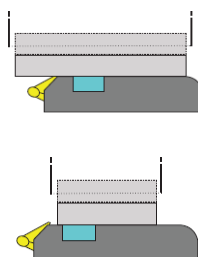


Sensore prodotto posteriore
Rear sensor

Per impilatore prodotti allineato col
retro del trascinato o per
confezioni voluminose

For products stacker aligned with
the dispenser rear side or for large
package size

EP-1A0C TRAS-101A CCTALK

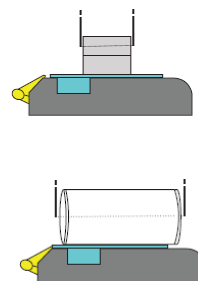


Sensore prodotto anteriore
Front sensor

Per impilatore prodotti non allineato
col retro del trascinato

For products stacker non aligned with
the dispenser rear side

EP-1LOC TRAS-101L CCTALK



Sensore prodotto anteriore
Front sensor

Per prodotti di forma rotonda (es.
lattine) o di piccole dimensioni

For round-shaped products (i.e. cans)
or for small packages



Alberici[®]
CASH SOLUTIONS

Progettazione e produzione di sistemi di pagamento e accessori per macchine Gaming, Vending e Car-Wash

Via Cà Bianca, 421 - 40024
Castel San Pietro Terme (BO) - Italy

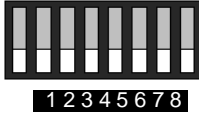
Tel.: +39.051.944300
Fax.: +39.051.944594

Web: www.alberici.net
E.mail: info@alberici.net

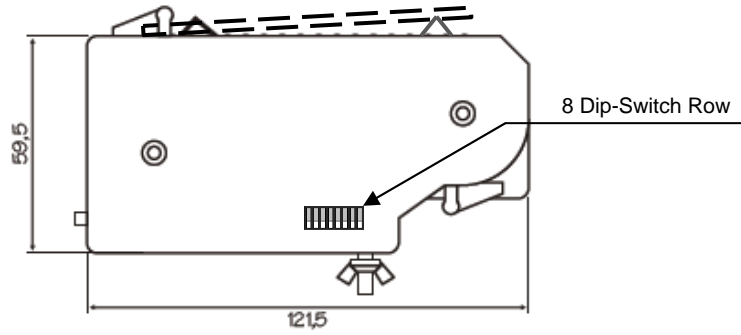
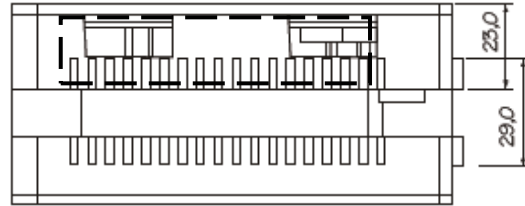
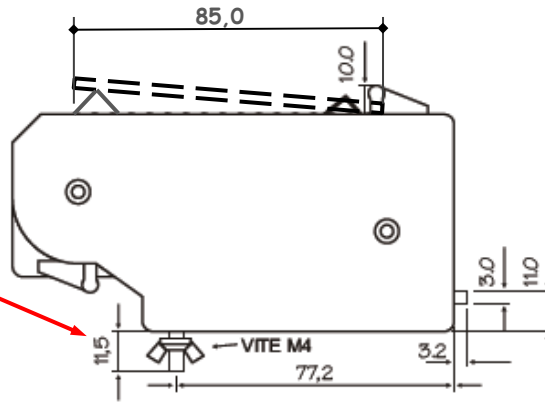
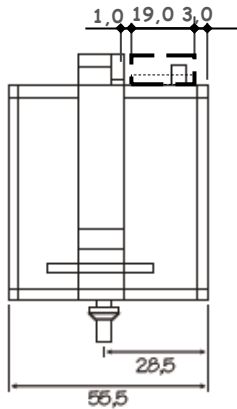
DIMENSIONS

≡ ≡ ≡ "L" type activation lever

The standard pad is equipped with a long tooth. Upon request, it can be supplied without tooth, or with a shorter one (half-length)



8 Dip-Switch Row for ccTalk Address setting



TECHNICAL DATA

Power supply	12-24Vdc (automatic compensation)
No load current	30 mAmp
Max. efficiency current	1 Amp (output power 12 W)
Stall current	0,550 Amp
Operating Temperature	0-50°C
Operating Mode	Intermittent operation

CONNECTORS

The Alberici vending motors can be provided either with 6 pin **standard Pulse communication interface** (TRAS-001P, TRAS-001A, TRAS-001L, TRAS-001D, TRAS-001C), or with 10pin **ccTalk protocol communication interface** (TRAS-101P, TRAS-101A, TRAS-101L). The pin-out for the ccTalk version is as follows:

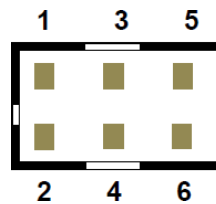
CCTALK

10 PIN CCTALK

DATA	1	2	NC
NC	3	4	GND
NC	5	6	NC
+12/24 Vdc	7	8	GND
NC	9	10	+12/24 Vdc



PUSHBUTTON



- 1 Micro NO (n.aperto/n.open)
- 2 Led R (rosso/red)
- 3 +12 Vdc (comune /common)
- 4 Led B (blu/blue)
- 5 GND
- 6 Led G (verde/green)

GENERAL INFORMATION

An 8-elements Dip-Switch Row is located on the side of the dispenser. Such DS Row allows to set the address of each dispenser in the same machine system.

Binary combinations resulting in 0 (all DS to OFF), 1 (first DS to ON, all others to OFF), and 2 (second DS to ON, all others to OFF) are free and available for Manufacturers who might need to use custom-made product codes.

All other combinations produce the operating Addresses from 3 to 255.

When the DS are set to binary combination 0 (all DS to OFF), the board produces by default the Address 03. Such Address can be changed by sending the 'Address change' instruction; by this procedure, the machine control board will be setting the address of each dispenser, according to the momentary needs.

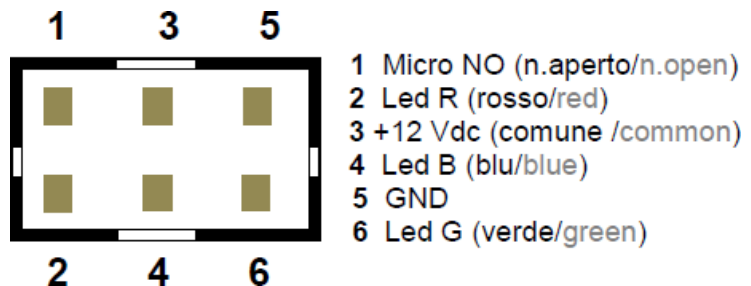
If the DS is set to a combination such as 3 or higher, the 'Address change' command will not be accepted, and the dispenser shall reply Nacknowledged.

IMPORTANT: the device accepts commands for dispensing one product unit per time. If the command asks for more than one product unit, the device shall reply Nacknowledged.

Power supply must be within 12Vdc e 24Vdc. The dispenser contains a compensation circuit that controls and equalizes the measured voltage level. The device will not work when voltage is less than 10 Vdc or exceeding 30 Vdc. .

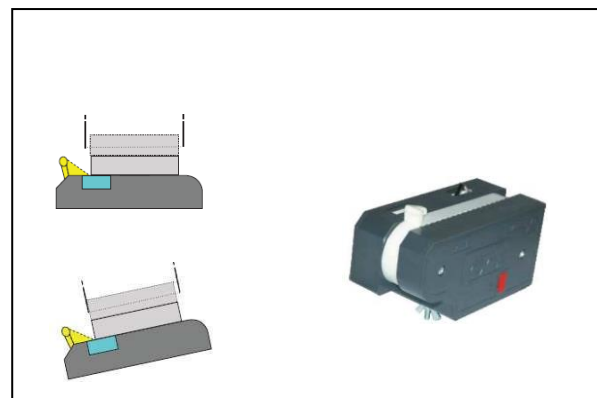
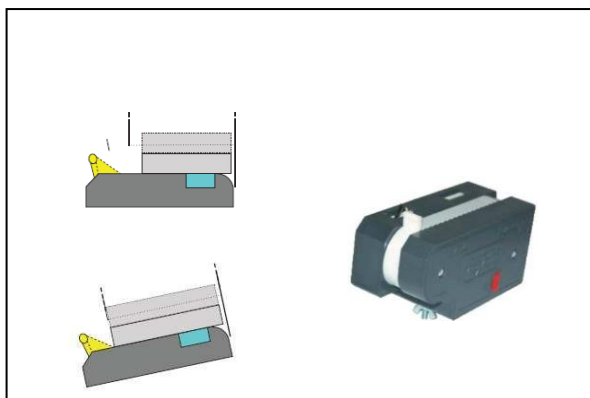
The ccTalk dispenser can accomodate one RGB pushbutton (see PIN-OUT here beside).

By using the Header 238 command, the RGB lights can be controlled, or by the header 248, check whether the pushbutton is being pressed.

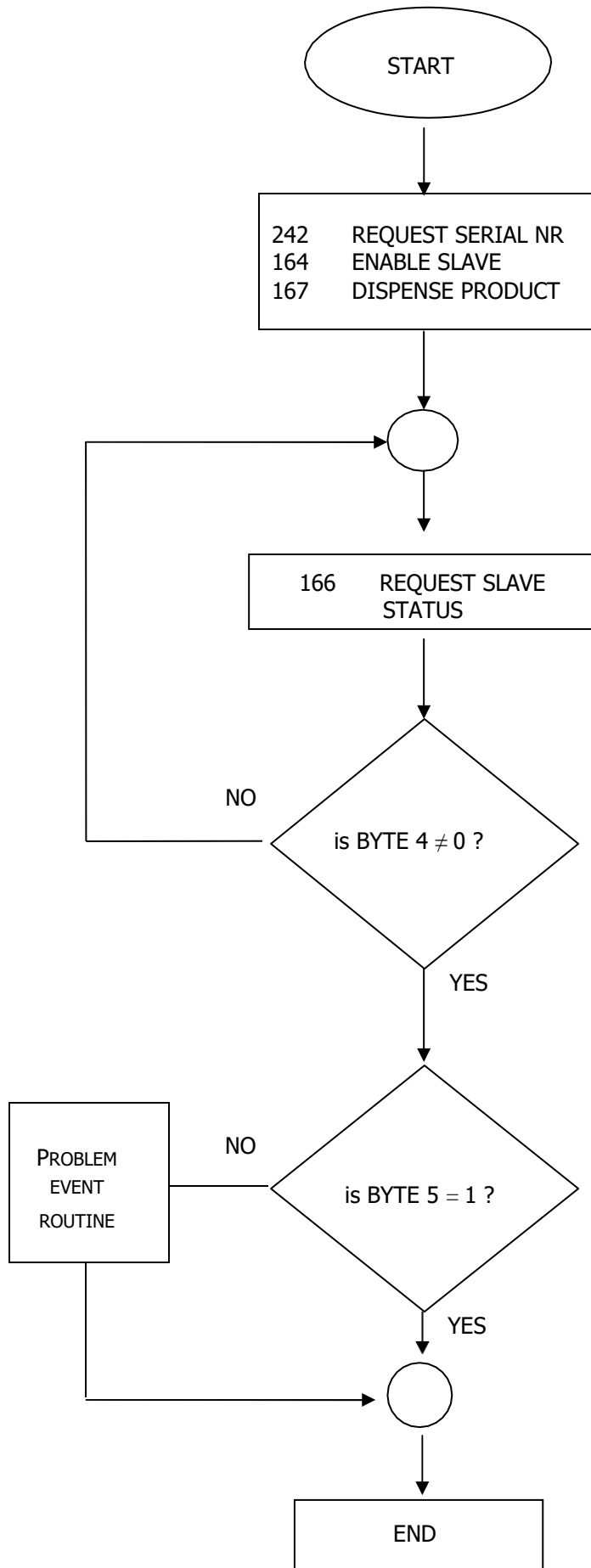


Stand-by position of the product dyke: by default, the belt stops in such a position that the product dyke stands in the diagonal position (as in figures on page 1).

If the belt must stop its rotation in at a different point, such as to get the dyke in a more convenient position (for instance, to retain small size products), the Header 214 can be programmed accordingly (see page 11).



CCTALK CONTROL FLOW-CHART



1. ccTalk commands

254 **Simple Poll**

Transmitted data: none
Received data: ACK

Simple polling, mostly used to check device at selected address.

253 **Address Poll**

Transmitted data: none
Received data: One byte (current device address).

This command is best to be sended with address 0.

All slaves respond with only one byte – own address after (1200 – 4*device address) miliseconds.It is used to get all slave addresses in the network.

252 **Address Clash**

Transmitted data: none
Received data: One byte (current device address).

This command is best to be sended with address 0.

All slaves respond with only one byte – own address after (1200 – 4*random(255)) miliseconds. It is used to find possible two devices with same addresses in the network.

251 **Address change**

Transmitted data: new address
Received data: ACK

Jumper JP1 must be closed before this operation and host must send this command with broadcast address (zero). If all was OK, eceived data is ACK with old address.New address is stored into NV memory and take effect at next command and every power up. Otherwise answer is NACK. Operator can close/open jumper with power on.

246 **Request manufacturer id**

Transmitted data: none
Received data: "Alberici"

Received data is ASCII string with manufacturer ID ("Alberici").

245 **Request equipment category ID**

Transmitted data: none
Received data: "Pusher"

Received data is ASCII string with equipment category ID ("pusher").

244 **Request product code**

Transmitted data: none
Received data: "Motor Vending M1"

Received data is ASCII string with product code ("Motor Vending M1").

- 242 **Request serial number**
Transmitted data: none
Received data: three byte serial number
Received data is three byte serial number written in NV memory in order LSB - MSB.
- 241 **Request software revision**
Transmitted data: none
Received data: four byte
Received data is four byte ASCII string with software revision ("1.01" for example).
- 217 **Request dispenser high/low status**
Transmitted data: none
Received data: one byte
Received byte is dispenser hi/low status where:
bit0 : 1 - empty
 0 - not empty
bit1 : 1 - full
 0 - not full
bit2 : not used
bit3 : not used
bit4 : low level sensor is on
bit5 : hi level sensor is on
bit6 : not used
bit7 : not used
- 216 **Request data storage available**
Transmitted data: none
Received data: five bytes
Description:
byte 1 : memory type
byte 2 : read blocks
byte 3 : read bytes per block
byte 4 : write blocks
byte 5 : write bytes per blocks
In this moment no data storage is available, so result is 00 00 00 00 00.
- 192 **Request build code**
Transmitted data: none
Received data: "AMV001"
Received data is ASCII string with build code ("ALM01v01" for example).
- 172 **Emergency stop**
Transmitted data: none
Received data: one byte
Device is momentarily stopped, and received data is number of products left for dispense.

169 **Request address mode**

Transmitted data: none
Received data: 0xB2

Description:

- Address is stored in NV memory (B2)

168 **Request dispense count**

Transmitted data: none
Received data: three bytes (first LSB, last MSB)

Total dispense counter saved in NV memory.

Data is transmitted from LSB to MSB byte.

167 **Dispense products**

Only quantity 1 (one piece) can be dispensed per each dispense command. If an attempt is made to dispense more than one product by a single dispense command, the motor dispenser responds with a "NACK" signal.

Transmitted data: four bytes
Received data: ACK or NACK

Host sends four bytes. First three is serial address of the motor, and fourth is number of products (1) to be dispensed.

Received data is ACK if everything is OK, or else NACK.

In both cases reasons for NACK can be:

- incorrect serial number
- Motor current overflow occurred during last dispense (see "Test dispenser")
- dispense is not enabled (see "Test dispenser")
- incorrect number of bytes was send from host to dispenser
- previous dispense operation is in progress.

166 **Request dispenser status**

Transmitted data: none
Received data: four bytes

Description:

byte1 : event counter - number of good dispense events since last reset

byte2 : number of goods to push out since last dispense command (decrements with each good pushed).

byte3 : number of goods pushed out since last dispense command (increments with each good pushed).

byte4 : number of goods failed to push out since last dispense command (this counter is cleared during pushout).

164 **Enable dispenser**

Transmitted data: one byte
Received data: ACK

Transmitted byte must be 0xA5. Only in that case dispense is enabled.

Next attempt to send this command without correct parameter disables the dispense.

163 **Test dispenser**

Transmitted data: none

Received data: one byte

Received bytes are:

byte 1:

0 -

1 -

2 -

3 -

4 - Pushout timeout occurred (next pushout or reset clear this flag)

5 - Motor current overflow during last dispense (soft reset clear this flag or if motor is released or reset)

6 - Power up detected

(power up set and soft reset clear this flag)

7 - Dispense disabled

(power up set and 'Enable device' command with A5 parameter reset flag)

004 **Request comms revision**

Transmitted data: none

Received data: three bytes

Description:

byte1 : cctalk level

byte2 : major revision

byte3 : minor revision

001 **Reset device**

Transmitted data: none

Received data: ACK

NOTE: Device response time for this command is app. 150 ms.

2. Setting start position

If motor is not in start position, operator must:

- power off device
- close jumper J1
- power on device and wait until belt stops in start position
- power off
- open jumper J1
- power on for normal operation.

3. Pinout

1	cctalk data line
4,8	Ground
7,10	+24v
2,3,5,6,9	not connected

MOTOR VENDING COMMAND HEADER SET

List of cctalk command headers for Motor vending dispenser:

Code	Command header	Note
254	FE	Simple poll
		Return ACK
253	FD	Address poll
		MDCES support in broadcast mode
252	FC	Address clash
		MDCES support in broadcast mode
251	FB	Address change
		support, non volatile
248	F8	Request status
		Return extern RGB switch information
246	F6	Request manufacturer id
		'Alberici'
245	F5	Request equipment category id
		'Dispenser'
244	F4	Request product code
		'AD-MTV-1' customizable on request
242	F2	Request serial number
		Return 3 byte: number from 0 to 16.777.215
241	F1	Request software revision
		'rx.xx' where x are numbers
238	EE	Test outputs lines
		Set external light on rgb switch
236	EC	Read optostates
		Compatibility
217	D9	Request dispense high/low stat.
		Return empty/full status
215	D7	Read data block
		Return setting information
214	D6	Write data block
		Program setting information
172	AC	Emergency stop
		Return ACK with one byte
167	A7	Dispense Slave products
		Data = Serial number +(byte=1)
166	A6	Request Slave status
		Return dispensed product counters
164	A4	Enable Slave
		Data must be A5
163	A3	Test Slave
		Return hardware status
1	1	Reset device
		Software reset

SETTING THE SLAVE ADDRESS VIA HARDWARE

The default address of the Alberici motor vending dispenser can be changed via the on-board 8 dip switch row.

See below the various combinations, each of them relating to one particular address.

Slave Address Table (empty box stands for 'Off' position of the Dip-Switch):

DipSw8	DipSw7	DipSw6	DipSw5	DipSw4	DipSw3	DipSw2	DipSw1	Address
							On	By default=03
								By default=03
						On		By default=03
						On	On	3
					On			4
					On		On	5
					On	On		6
					On	On	On	7
				On				8
				On			On	9
				On		On		10
				On		On	On	11
				On	On			12
				On	On		On	13
				On	On	On		14
				On	On	On	On	15
			On					16
			On				On	17
			On			On		18
			On			On	On	19
			On		On			20
			On		On		On	21
			On		On	On		22
			On		On	On	On	23
			On	On				24
			On	On			On	25
			On	On		On		26
			On	On		On	On	27
			On	On	On			28
			On	On	On		On	29
			On	On	On	On		30
			On	On	On	On	On	31
		On						32
		On					On	33
		On				On		34
		On				On	On	35
		On			On			36
		On			On		On	37
		On			On	On		38
		On			On	On	On	39
		On		On				40
		On		On			On	41
		On		On		On		42
		On		On		On	On	43
		On		On	On			44
		On		On	On		On	45
		On		On	On	On		46
		On		On	On	On	On	47
		On	On					48
		On	On				On	49
		On	On			On		50
		On	On			On	On	51
		On	On		On			52
		On	On		On		On	53
		On	On		On	On		54
		On	On		On	On	On	55
		On	On	On				56
		On	On	On			On	57
		On	On	On		On		58
		On	On	On		On	On	59
		On	On	On	On			60

		On	On	On	On		on	61
		On	On	On	On	On		62
		On	On	On	On	On	On	63
	On							64
	On						On	65
	On					On		66
	On					On	On	67
	On				On			68
	On				On		On	69
	On				On	On		70
	On				On	On	On	71
	On			On				72
	On			On			On	73
	On			On		On		74
	On			On		On	On	75
	On			On	On			76
	On			On	On		On	77
	On			On	On	On		78
	On			On	On	On	On	79
	On		On					80
	On		On				On	81
	On		On			On		82
	On		On			On	On	83
	On		On		On			84
	On		On		On		On	85
	On		On		On	On		86
	On		On		On	On	On	87
	On		On	On				88
	On		On	On			On	89
	On		On	On		On		90
	On		On	On		On	On	91
	On		On	On	On			92
	On		On	On	On		On	93
	On		On	On	On	On		94
	On		On	On	On	On	On	95
	On	On						96
	On	On					On	97
	On	On				On		98
	On	On				On	On	99
	On	On			On			100
	On	On			On		On	101
	On	On			On	On		102
	On	On			On	On	On	103
	On	On		On				104
	On	On		On			On	105
	On	On		On		On		106
	On	On		On		On	On	107
	On	On		On	On			108
	On	On		On	On		On	109
	On	On		On	On	On		110
	On	On		On	On	On	On	111
	On	On	On					112
	On	On	On				On	113
	On	On	On			On		114
	On	On	On			On	On	115
	On	On	On		On			116
	On	On	On		On		On	117
	On	On	On		On	On		118
	On	On	On		On	On	On	119
	On	On	On	On				120
	On	On	On	On			On	121
	On	On	On	On		On		122
	On	On	On	On		On	On	123
	On	On	On	On	On			124
	On	On	On	On	On		On	125
	On	On	On	On	On	On		126
	On	On	On	On	On	On	On	127
On								128
On							On	129
On						On		130
On						On	On	131
On					On			132
On					On		On	133
On					On	On		134

On					On	On	On	135
On				On				136
On				On			On	137
On				On		On		138
On				On		On	On	139
On				On	On			140
On				On	On		On	141
On				On	On	On		142
On				On	On	On	On	143
On			On					144
On			On				On	145
On			On			On		146
On			On			On	on	147
On			On		On			148
On			On		On		On	149
On			On		On	On		150
On			On		On	On	On	151
On			On	On				152
On			On	On			On	153
On			On	On		On		154
On			On	On		On	On	155
On			On	On	On			156
On			On	On	On		On	157
On			On	On	On	On		158
On			On	On	On	On	On	159
On		On						160
On		On					On	161
On		On				On		162
On		On				On	On	163
On		On			On			164
On		On			On		On	165
On		On			On	On		166
On		On			On	On	On	167
On		On		On				168
On		On		On			On	169
On		On		On		On		170
On		On		On		On	On	171
On		On		On	On			172
On		On		On	On		On	173
On		On		On	On	On		174
On		On		On	On	On	On	175
On		On	On					176
On		On	On				On	177
On		On	On			On		178
On		On	On			On	On	179
On		On	On		On			180
On		On	On		On		On	181
On		On	On		On	On		182
On		On	On		On	On	On	183
On		On	On	On				184
On		On	On	On			On	185
On		On	On	On		On		186
On		On	On	On		On	On	187
On		On	On	On	On			188
On		On	On	On	On		On	189
On		On	On	On	On	On		190
On		On	On	On	On	On	On	191
On	On							192
On	On						On	193
On	On					On		194
On	On					On	On	195
On	On				On			196
On	On				On		On	197
On	On				On	On		198
On	On				On	on	On	199
On	On			On				200
On	On			On			On	201
On	On			On		On		202
On	On			On		On	On	203
On	On			On	On			204
On	On			On	On		On	205
On	On			On	On	On		206
On	On			On	On	On	On	207
On	On		On					208



Progettazione e produzione di sistemi di pagamento, accessori per videogames e macchine vending
Design and manufacture of payment systems, accessories for videogames and vending machines

Via Ca' Bianca 421
40024 Castel San Pietro
Terme (BO) – ITALY

Tel. + 39 051 944 300
Fax. + 39 051 944 594

<http://www.alberici.net>

info@alberici.net