Alberici

# AL 55/66 COIN SELECTORS

## Technical Manual

## Content

## Changes

**Ver. 1.1**
Change in part 6.5.2. description of command 185 due to revision of coin selector software version 1.1.4 from 5.05.2007.

# 1. General description

Electronic coin selector AL66 is member of third generation of ALBERICI coin selectors.
It is mechanically and electrically compatible with older generation of ALBERICI coin selectors AL05 and 06. The compatibility with others 3 ½" coin selectors using 10 pole IDC connectors is maintained as well. It is also possible to change power input polarity to match with coin selectors AZKOYEN or MONEY CONTROLS mod 3.
Users can simply change coin selector function to fit in to various applications, using our free programming software. Change is made simply by selection one of the standard functions described in this document or by creating his own function ("custom") through serial interface and terminal window.
There is also possibility for customers to order a special or personal function created by factory experts. Coin selectors can be delivered programmed or be reprogrammed by customer with simple copy function.
The new coin selectors have advanced measuring system with three pairs of magnetic sensors and additional optical measurement of coin diameter.
The "heart" of coin selector is "Freescale"[1] 8 bit micro controller of new generation with 36 kB of FLASH memory and high immunity to electro magnetic disturbances ("automotive").
The number of coin movement control sensors is increased to[2].
Power saving mode is supported and has advanced wake up possibility.
In some special version (for telephone cabinet) it is possible to switch off power supply by applying control signal to one of external 10 pole connector pin.
Power consumption is further reduced in stand by and in power saving mode.
Power supply voltage is more flexible with respect to older generation and EMI[3] resistance is improved.
Firmware up-grade is supported and time for complete up-grade is reduced to approx. 10 seconds.
New generation of coin selectors support safe transfer of data and safe authentication with RSA (*asymmetric algorithm with private and public key*) crypt algorithms.

[1] Ex Motorola

[2] Number of control sensors pair could be 1, 2 or 3 in most advanced and secure version

[3] Electro Magnetic Interferences

## 2. Mechanical models

Coin selectors AL55/66 are produced in four mechanical models:

- Standard **V** model
- Inverted V model or **I** model
- Standard **K** model
- Short K model or **S** model

### 2.1 Standard V model

This model is mechanically compatible with 3 ½" mechanical coin selectors.
Coin selector can fit in to a standard clip-in support of any producer.
Coin inlet is at the top of coin selector and outlet for accepted and rejected coins are at the bottom side (*see image 2.1*). Acceptance lane is little longer than rejected coin path therefore acceptance rate is reduced to 2 or maximum 3 coins per second.

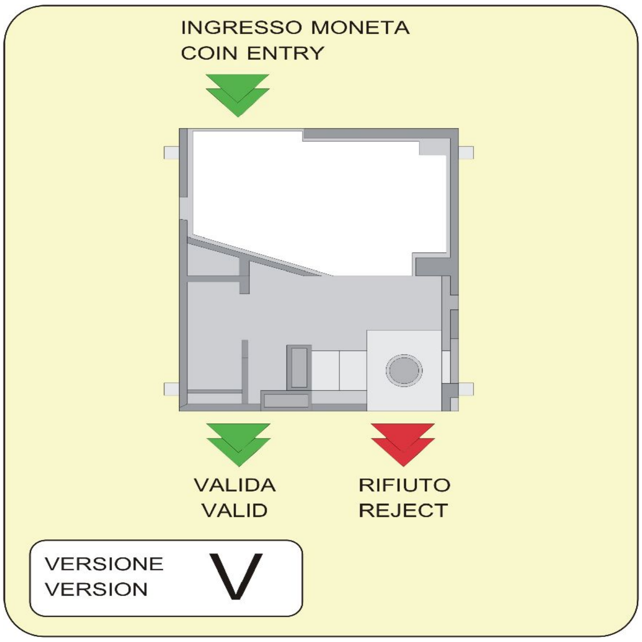


Image 2.1 Coin path for V version

Accepted coins path has double turn and that make fishing fraud(*coin on string*) more difficult.
In addition all V models have built in up to three mechanical protection mechanism to prevent return of accepted coins. There is a steel string cutter before acceptance gate, coin non return mechanism before acceptance and after acceptance gate!

## 2.2 Inverted model I

This model is very similar to V model except for outlet of coins. Accepted coin and rejected coin outlet are reversed (*see image 2.2*).
Acceptance lane is shorter and acceptance rate is 3 to 4 coins per second. This version is often used in parking machines.

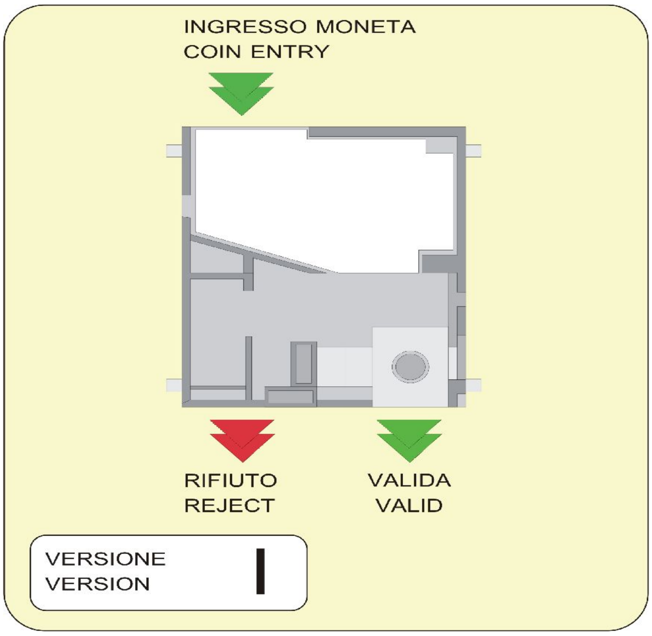


Image 2.2 Coin path for I version

Accepted coins will drop straight through acceptance gate so only one non return mechanism and steel string cutter can be used!

## 2.3 Model K with standard front plate

This model is often used in “kiddy rides” and for that reason is named as K.
Coin selector is mounted on simple and cheep front plate mask and support that occupy little space. Disadvantage of that support is that coin selector is more exposed to external conditions and fraud attempt. Coin inlet is from top side, outlet for accepted coins is vertically on bottom side and outlet for rejected coins is horizontally through front mask (*see image 2.3*).
U Acceptance lane is shorter and acceptance rate is 3 to 4 coins per second.

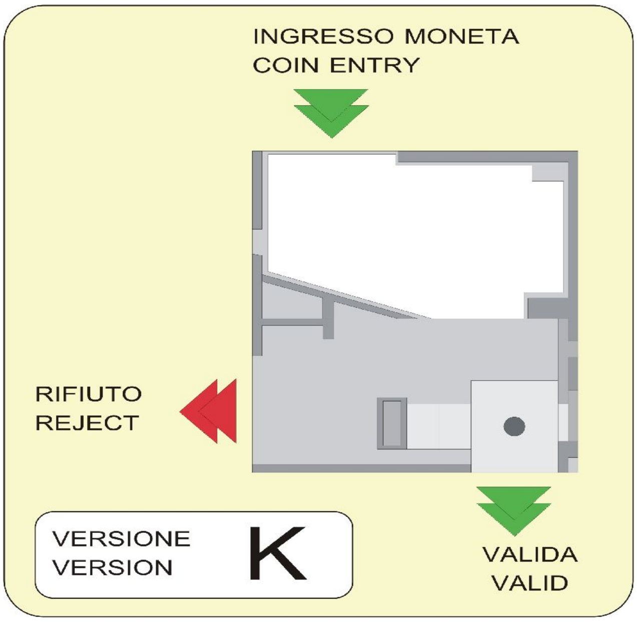


Image 2.3 Coin path for K version

Accepted coins will drop straight through acceptance gate so only one non return mechanism and steel string cutter can be used!

## 2.4 Model S with short front plate

Is almost identical with K model but front plate mask is shorter and occupy less space. Coin inlet is moved lower on back side (*see image 2.4*).
Exposure of sensitive sensor part to external conditions and possible damage is higher than on any other model.

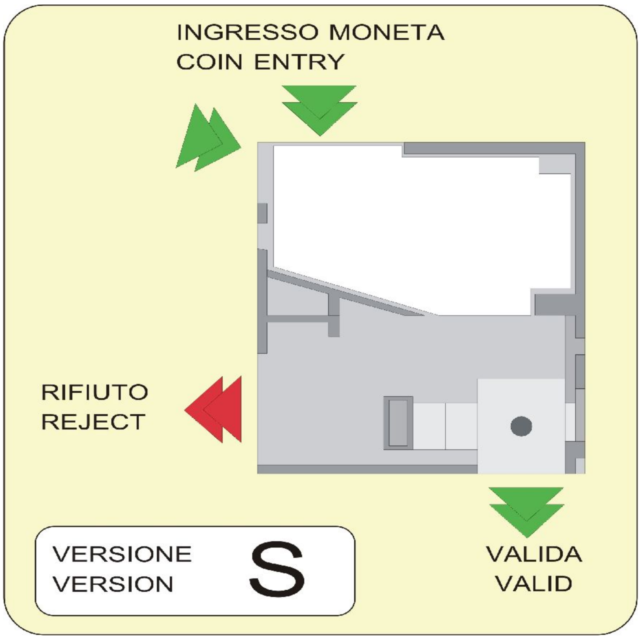
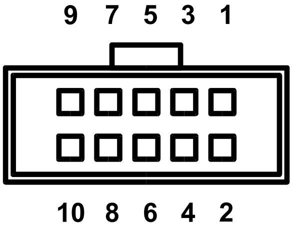


Image 2.4 Coin path for S version

# 3. Connections

Coin selectors are connected with peripheral devices or host machine through several connectors shown on image 3.1.

Standard power supply and interface connector is 10 poles IDC connector **X1** presented on image 3.2. There is totally 6 "open collector" outputs (*pins 3,4,7,8 and 9*), one input (*pin 6*), and one pin that has double function (*pin5*), and could be used as input or open collector output. Two pins (*pins 1 and 2*) are power supply input pins. Standard polarity is when plus is applied to pin 1 and pin 2 is common or "ground".
On special request resistors input polarity could be reversed to match the polarity of Spanish coin selectors (*Azkoyen or Money Controls mod 3*).



| Pin nr. | Description |
|---|---|
| 1 | Power GND |
| 2 | Power supply +8-26 VDC |
| 3 | Out 5 / Sorter coil B |
| 4 | Out 6 / Sorter coil A |
| 5 | Out 7(*credit*) / In 2 |
| 6 | In 1(*inhibit*) |
| 7 | Out 1 |
| 8 | Out 2 |
| 9 | Out 3 |
| 10 | Out 4 / Sorter coil C |

Image 3.2 Interface connector X1

Four poles connector **X3** is used for **cctalk®**[4] serial communication with host machine. Protocol is designed for "slave" mode of communication and described in chapter 6 of this document. Connection schematic of X3 is shown on image 6.1(*chapter 6*).

Six pole connector X2, is used for connection of different SPI display units or decryption/encryption unit[5]. Coin selectors AL66 support several types of displays with SPI or I²C bus communication protocol. User can select display with our programming software or by ordering it. Currently supported types are:

| | |
|---|---|
| - MC 14499 | 4 digit compatible with RM924S SECI or G-51.1092 NRI |
| - MC 14489 | 5 digit compatible with G-53.0747 NRI |
| - MAX 7219 | 6 digit AL066 ALBERICI |
| - M643 | 8 digit LCD |

For SECI and NRI displays it is necessary to use cable adapter.
Connection schematic of X2 is shown on image 3.3.

[4] Money Controls protocol details could be found on: http://www.cctalk.org
[5] Additional SPI crypto MCU

# 4. Functional versions

ALBERICI coin selectors support the possibility to adjust coin selector to various applications. For user this gives the great flexibility. With same type of coin selector user can have uniform spare stock and reuse same coin selector in different machines. This technical manual describes only basic functional versions supported with programming software. After version selection simply make one "click" and coin selector function will be changed.
Every function has additional options and data that could be selected or changed by user[6].
Experienced users can create they own "custom" functions, and our company will provide all necessary support and help.
Basic functional versions are:

- Validators
- Totalizers
- Timers
- Multi-price steppers

## 4.1 Validators

Validators are type of coin selectors that use output pins of interface connector to transfer simple pulse as information about coin that has been detected. Mostly they use parallel interface with host machine. Coin type is detected by pin number that has been activated. I special case validator could be serial and pass the information to host as number of serial pulses according to accepted coin relative value.
The problem with serial transfer is time for transfer that in some cases could be significant. That is the main reason why parallel validators are in use.
Parallel validators have several versions:

- Standard parallel
- Parallel combinatorial (EVA standard)
- Multi-impulse parallel
- Parallel with Spanish protocol

### 4.1.1 Standard parallel

Standard parallel validator is most commonly used. Most of coin selector producers support that type of function.
Coin selectors with 10 pole connectors use 6 outputs to indicate coin type that has been detected and accepted. One input called "inhibit" is used to block acceptance of all coins. Some validators have also one output as indication of return or "flight deck" opened. After the acceptance of one programmed coin, corresponding output is activated for a brief time period. Active level is mostly "low" and period is typically 100 ms.
For coin selectors AL55/66 user can program time period from 5 to 1000 ms in steps of 5 ms. Using programming software all output timing will be set at same values but different timing period could be easily programmed for each output separately (*custom*).

[6] Credit price, bonus level, bonus value, time, etc.

### 4.1.2 Parallel combinatorial (EVA standard)

Parallel combinatorial standard was introduced and proposed by EVA[7] when euro coins where introduced in European countries. In transition period national and euro coins coexisted and 6 output for coins where insufficient.
Vending machine controllers where reprogrammed to accept both currencies.
For details about protocol see document EVA Spec. 01.pdf.
Basically outputs are activated in combinatorial way to indicate coin that has been accepted. In this mode output 1 (*pin 7*) is permanently active and indicate type of validator. Output 2 (*pin 8*) is output of lowest binary weight, followed by output 3 (*pin 9*), output 4 (*pin 10*) and output 5 (*pin 3*) as output of highest binary weight. Output 6 (*pin 4*) is parity check output. With this type it is possible to indicate acceptance of 15 different coins. Euro coins are programmed to be on first position beginning with 1 cent coin. Position 9 is reserved for 5 Euro coin in future use.
The last positions are used for national currencies and the very last one for user token.

### 4.1.3 Multi-impulse parallel

These type parallel validators are proposed also by EVA and described in same document as previous type. One way to reduce number of coin output is to use same output for coins of different value. In this case the relative value of two coins is 1 and 2. Output wills be activated two times when coin with value of 2 is accepted or once if coin of value 1 is accepted. Using that type of validator there will bee no knead to reprogram host machines software and 6 outputs will be sufficient in most cases.
There are two modes of operation for that type of validator. Normal mode with standard 100 ms pulses and fast mode with 50 ms pulses.

### 4.1.4 Parallel with Spanish protocol

Parallel validators with so coled "Spanish" protocol are some times called validators with "hand shake"[8] protocol. They are developed and introduced to market by Spanish company AZKOYEN and that why they are called **Spanish protocol validators**.
In most cases only one input is used for blocking the acceptance there is also versions with sorter enable/disable input.
While in stand by coin selector acceptance and sorter are inhibited. After coin has been introduced and recognized coin selector activate short impulse on corresponding parallel output. Time of impulse is typically 10 ms. After detecting the change host machine must decide within time period of 10 ms ether to accept or to reject introduced coin.
If coin should be accepted, inhibition input will be deactivated by host till coin exit from coin selector. Sorting mechanism is enabled or left disabled at same time.
After acceptance coin selector activate for second time same output as indicator that coin has been accepted correctly. Second impulse is typically 100 ms long but could be programmed by user to any value between 5 ms and 1000 ms.

[7] **E**uropean **V**ending **A**ssociation

[8] This stands for interaction of host and coin selector to achieve control of acceptance and sorting of coins.

## 4.2 Totalizers

Totaizers are coin selectors that use one output to activate host machine, when certain amount of coin value has been accumulated. Value for activation of credit could be programmed by user programming software or set by DIP-switches on coin selector PCB. Pulse time duration and delay between two pulses is possible to program and typical value is 100 ms with delay of 200 ms. Activation value is called credit value or price of a credit. Special type of totalizers that has credit price set to 1 is also called serial validator.
Except price user can program bonus level and number of bonus credit.
Totalizers have also one output used to drive external counters. If display option is selected accumulated value that is less than credit value is possible to display.

### 4.2.1 Mode 1(ALBERICI AL03/05/06)

Totalizers mode 1 are compatible with older versions of ALBERICI coin selectors from AL03 to AL05 and AL06. Credit activation output is on **pin 5**, and all other functions are same as for mode 2 and 3.
Coin acceptance blocking is on **pin 6**.

### 4.2.2 Mode 2(COMESTERO RM4/RM5)

Totalizers mode 2 are almost compatible with COMESTERO RM5x20 or older version RM4x20 (*SECI*) coin selectors. The difference is with connection of display unit. ALBERICI coin selectors use separate connector for display unit and COMESTERO use 10 poles IDC connector pins 3, 4 i 7. With simple cable adapter same type of display is possible to connect to SPI connector (*X2*) of AL66 coin selectors.
Credit activation output is on **pin 9**, coin counter output is on **pin 8** and coin acceptance block input is on **pin 6**.

### 4.2.3 Mode 3(MONEY CONTROLS SR3)

This type of totalizer is completely compatible with Money Controls coin selectors SR3 type2 mode 3[9] and mode 8. Credit activation output is on **pin 7**.
**When coin selector is used as Money Controls SR3 mod 3 replacement it is necessary to change position of zero ohm resistors RP1 and RP3 to positions RP2 i RP4 on PCB!**

### 4.2.4 Mode 4(AZKOYEN)

**AZKoyen coin selectors has reverted power supply pins to, and same operation must be done to maintain compatibility!**
Beside of that all AZKoyen coin selectors has output 5(*pin 3*) and output 7(*pin 5*) reserved for sorter coil drive. Output 6(*pin 4*) is used as serial output.

### 4.2.5 On request mode 1(ALBERICI AL03/05/06)

This type of totalizers activate output for credit if accumulated value is same or higher than credit price and input for request is activated. Coin selectors ALBERICI has request input on same input pin as input for acceptance block (*Inhibit*) and request is activated when signal return to initial level. Minimum pulse time on input is possible to program, and default value is 50 ms. During that period coin selector acceptance will be blocked. All other functions and options are same as for standard totalizer.

### 4.2.6 On request mode 2(COMESTERO RM4/RM5)

Output for credit activation is on same pin as one on standard COMESTERO totalizers

[9] SR3 mode 3 has power supply input inverted (pins 1 and 2)

(*pin 9*). Request input is on **pin 5** and inhibit input is on **pin 6**.
Totalizers on request COMESTERO has also one output (pin 10) for signalization of accumulated credit presence. This output will be permanently active if coin selector accumulated value is same or higher than one credit. COMESTERO coin selector also uses 10 pole connector to drive display unit but coin selectors ALBERICI cant use same pins and display unit must be connected on 6 poles connector (*x2*) with cable adapter. ALBERICI 6 digits LED or 8 digit LCD display unit could be connected instead.

## 4.3 Timers

Timer coin selectors has one output that is active for programmed period of time when accumulated value is higher than price of credit. Timers on request has similar function as totalizer on request but output activity period is longer[10].
Some timers has also output for time out warning and some has input for time stop. All other functions and options are similar to totalizer coin selectors.
It is possible to use of display unit for elapsing time period presentation.

### 4.3.1 Progressive timer mode 1(ALBERICI AL03/05/06)

Progressive timer mode 1 has one output that is activated after coin selector accumulate programmed value. Time period is calculated by multiplying the accumulated value and programmed time multiplier. During the timer output activity all imported value will be multiplied with that time multiplier and added to time elapse period. Time output is on pin 5. Input for coin acceptance block(inhibit) is on pin 6 as for most coin selectors. Optionally for presentation of time and imported coin value it is possible to use different type of displays on AL566 coin selectors. Time format is **mm.ss** or **mm ss** if 8 digit LCD is used. Coin imported value is displayed before timer output start at same position on display.

### 4.3.2 Progressive timer mode 2(COMESTERO RM4/RM5)

Progressive timer mode 2 has same function as mode 1. Timer output is on pin 8 and additionally mode 2 progressive timer has input on pin 6 that act as timer Stop/Start input and as inhibit input. There is also one output on pin 9 used as timer elapse warning output. Time of warning activity is possible to program from 0 to 255 seconds. Warning time will be automatically limited to value that is lower that minimum timer period. COMESTERO coin selectors use 10 pole IDC pins 3, 4 and 7 to connect display unit. ALBERICI coin selector use separate connector X2 to connect same type of display or any of supported display unit.

### 4.3.3 Timer on request mode 1(ALBERICI AL03/05/06)

Timer on request coin selectors activate the output for programmed time period when imported coin value is same or higher than credit price and request input has been activated. It is same function as totalizer on request but output activity period is longer. For each accumulated credit it is possible to activate output once. Next activation is possible after time output has elapsed.
The "Rest" management is optionally enabled as described in chapter 5.
Time output is on pin 5. Input for request activation and inhibit is on pin 6.

[10] It goes from few seconds to hours

Optionally accumulated credit, time and rest are displayed.

#### 4.3.4 Timer on request mode 2(COMESTERO RM4/RM5)

It is similar as mode 1 timer with timer output on pin 9. Request input is on pin 5 and pin 6 is used as inhibit. additionally this mode of coin selectors has output for accumulated credit signalization on pin 10 and coin counter output on pin 8.
Display connection is same as in case of progressive timer mode 2(see chapter 4.3.2).

### 4.4 Multi-price stepper

Stepper is type of coin selector that activate one or more output when accumulated coin value is higher or same as value programmed for that output. That value is some time called price. Output remain active till reset signal is applied to coin selector reset input. Dual-price stepper was usually used in simple vending machines that has no sophisticated or intelligent controller. There is some variation of reset mode and output activation mode when more than one price output is enabled.
During the vend period reset line is usually active. At the end of vend period reset line goes back to normal inactive state. Reset of price line will occur usually at the end of vend period. During that period coin acceptance will be blocked(inhibited).
In some cases reset must be activated at the beginning of vend period.
Activation of outputs with different price has different modes to.
All lines with lover or same price than accumulated coin value could be active or lower price line must be turned off when higher price value is reached.
ALBERICI coin selectors support both variation of output or reset.

#### 4.4.1 Multi-price stepper(6) mode 1(ALBERICI)

This type of coin selector has 6 output lines that is possible to program with different price value. Inhibit is programmed on input 1(*pin 6*) and separated from reset line on input 2(*pin 5*). Inhibit and reset is active high but could be optionally changed to different combination of levels. Reset must be active for et least 50 ms before price line will be reset. Optionally it is possible to enable display of imported coin value!

#### 4.4.2 Dual-price stepper

This type of coin selector use only 2 outputs for price line. Inhibit and reset are programmed on same input(*pin 6*). Inhibit line is active high during the vend period and goes high on its end. In that moment price line will be reseted. Pin out is almost compatible with COMESTERO coin selector type RM4(5)x**10**. Display is possible to connect on 6 pole connector. DIP-sw 1 and 2(*if mounted*) are used to program the price 1 or 2.

#### 4.4.3 Photo-copy mode

Photo copy mode use only one price line(*vend enable*) which will be reseted after acceptance of programmed number of reset impulses. Reset line and inhibit line are programed on input 1(*pin 6*), output vend enable is on output 3(*pin 9*) and there is also output for rest value accumulated(*pin 10*) signalization and for coin counters(*pin 8*).

## 5. Additional functions

Additional functions of coin selector has direct impact to coin selector application variety. Using some this functions it is possible to make sophisticated host machine without use of "intelligent" host controllers(simple machines). Some of this functions are previously mentioned in description of coin selector types.
Additional functions are:

- Power saving mode
- Self programmable mode
- Coin sorting
- Displays
- Rest management
- Fraud inhibit

## 5.1 Power save mode

In power save mode coin selector power consumption will be reduced to just few milliamperes for standard coin selectors[11] or to less than 1 milliampere for special low power version.
Use of power saving mode must be enabled in factory on customers request.
Even when stop mode is factory enabled, user can disable this function and program time period of stand by, after which coin selector will go to power save function.
Minimum time is 10 seconds and maximum is 254 seconds(*about 4 minutes*). Shorter time is possible to program but it is not recommended. The coin selector do not communicate during the "stop" period. To reprogram stop time user must reset coin selector and change time before coin selector go back to stop.
Two different stop modes are available:

- Standard power save mode
- Power save mode with auto wake up

### 5.1.1 Standard power save mode

Wake-up from this power saving mode is done by applying voltage level of 5-30 V to the input 1(*pin 6*), that is usually used as "inhibit" input.
Minimum wake up time pulse is about 1 ms. After wake up from power saving mode coin selector will be ready to measure, recognize and accept coins in much shorter time than after power on. This period is about 50 ms.
Each coin insertion resets the power save time period. During the activity of any output the coin selector will not enter in power saving mode. If coin selector has LED display unit all segments will be turned of but the controller supply will still be active.
Using LCD display power consumption will be much lower and in power saving mode it is possible to stop LCD internal oscillators and further reduce the power consumption for about 1 mA. In such case display will be off(*all digits blank*)[12].

[11] See technical data in chapter 9
[12] See technical data for display-s

### 5.1.1 Power save mode with auto wake up

Wake up from this power saving mode is done by inserting the coin in to coin selector. First inserted coin will wake up coin selector but it will be rejected.
As waking trigger internal sensors of coin selector are used, and no additional circuit or switch are needed.

## 5.2 Self-programmable mode

Self-programmable mode enable the coin programming on site without use of PC programmer or any other intelligent host.
Programming options and coin settings are limited because of DIP-sw use[13].
There are several self-programming modes:

- Simple self-programming mode
- Advanced self-programming mode1
- Advanced self-programming mode2

Self-programming mode must be enabled in factory on customers request.
**To enter in self-programming mode(*if it is enabled*) put all switches in off position before coin selector power on!**
At coin selectors AL66 green indicator will start to blink.

### 5.2.1 Simple self-programming mode

In this mode one or two DIP-sw banks are used to select channel on which coin will be programmed. Switches of bank 1 select channels 1-6 and switches of bank 2 select channels 7 to 12.
Only one switch at the time must be set when coin selector is in this mode.
In case of error red indicator will blink after first coin insertion!
First special case is factory programming reset. To reset coin data to initial factory programming after entering in to self-programming mode all switches of first bank must be set in on position. After first inserted coin red indicator will start to blink.
After third inserted coin all coin data will be reset to factory setting, acceptance gate coil will be turned on for short period(*one click*) and coin selector will exit from self programming mode.
Second special case is coin channel erase. If all switches are in off state after insertion of third coin all coin channels will be erased! Acceptance gate coil will be turned on for short period(*one click*) and coin selector will remain in self-programming mode.
**In both special cases red indicator will blink after first coin insertion!**
This will be the the case only for coin selectors AL66! Coin selectors AL55 do not have indicator and user must be pay special attention to switch setting!
In simple self programming mode coin value, sorting direction or state will not be changed if this data was previously programmed. If coin channel was erased value will be set to 1, sorter direction to 1(*cash box*) and coin state to enabled!

[13] **D**ual **I**n-Line **P**ackage (*switch banks on printed circuit board)*

### 5.2.2 Advanced self-programming mode1

In this mode switches are used to:

- select coin channel to program
- select coin channel to erase
- set coin value
- direction of sorting
- set coin state

This mode is usually used on coin selectors AL66 with both DIP-sw banks.
It is possible to use this mode with some limitation in coin selectors with one DIP-sw bank(*AL55*).
Selection of coin channel to program is done with first 4 switches on first DIP-sw bank, according to table 5.1:

| | DIP-sw bank 1 | | | | | |
|---|---|---|---|---|---|---|
| | Sw1 | Sw2 | Sw3 | Sw4 | Sw5 | Sw6 |
| Channel 1 | Off | Off | Off | Off | Off | On |
| Channel 2 | On | Off | Off | Off | Off | On |
| Channel 3 | Off | On | Off | Off | Off | On |
| Channel 4 | On | On | Off | Off | Off | On |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| Channel 15 | Off | On | On | On | Off | On |
| Channel 16 | On | On | On | On | Off | On |
| Fraud Ch x | X | X | X | X | On | Off |
| Erase Ch x | X | X | X | X | Off | Off |
| Factory res. | On | On | On | On | On | On |

Table 5.1: Coin channel selection

Note:
If DIP-sw bank 2 is not present it is not possible to erase first channel. All channels will be erased instead(*all switches in off position*)

Switches 5 and 6 determinate coin state and must not be set “on” at same time, except in case of factory reset!
Switch 6 is “on” if programming coin must be accepted(*normal coin*), while switch 5 must be set “on” when we are programming so called “fraud” coin(*coin with similar parameters but not acceptable*).
Factory reset and erase of all coin channels are the same in both self programmable modes.

With second bank DIP-switch we can program coin value and sorting direction according to table 5.2 and 5.3.

| | DIP-sw bank 2 | | | | | |
|---|---|---|---|---|---|---|
| Coin value | Sw1 | Sw2 | Sw3 | Sw4 | Sw5 | Sw6 |
| No Change | Off | Off | Off | Off | X | X |
| 1 | On | Off | Off | Off | X | X |
| 2 | Off | On | Off | Off | X | X |
| 4 | On | On | Off | Off | X | X |
| 5 | Off | Off | On | Off | X | X |
| 8 | On | Off | On | Off | X | X |
| 10 | Off | On | On | Off | X | X |
| 20 | On | On | On | Off | X | X |
| 25 | Off | Off | Off | On | X | X |
| 40 | On | Off | Off | On | X | X |
| 50 | Off | On | Off | On | X | X |
| 80 | On | On | Off | On | X | X |
| 100 | Off | Off | On | On | X | X |
| 125 | On | Off | On | On | X | X |
| 200 | Off | On | On | On | X | X |
| 250 | On | On | On | On | X | X |

Table 5.2: Coin value programming

| | DIP-sw bank 2 | | | | | |
|---|---|---|---|---|---|---|
| Direction | Sw1 | Sw2 | Sw3 | Sw4 | Sw5 | Sw6 |
| No Change | Off | Off | Off | Off | Off | Off |
| 1 | X | X | X | X | Off | Off |
| 2 | X | X | X | X | On | Off |
| 3 | X | X | X | X | Off | On |
| 4 | X | X | X | X | On | On |

Table 5.3: Coin sorting direction programming

### 5.2.3 Advanced self-programming mode2

This mode is enabled only for parallel validator if both DIP-sw banks are available. DIP-sw bank 1 is used to select coin channel and status to program in same way as in mode 1. DIP-sw bank 2 is used to select activation output of programmed channel according to table 5.4.

| | DIP-sw bank 2 | | | | | |
|---|---|---|---|---|---|---|
| | Sw1 | Sw2 | Sw3 | Sw4 | Sw5 | Sw6 |
| Out nr. | 1 | 2 | 3 | 4 | 5 | 6 |
| Pin | 7 | 8 | 9 | 10 | 3 | 4 |

Table 5.4: Output activation selection

Switch 1 to 6 are representing activation mask of outputs 1 to 6.
Multiple output activations are allowed and used in parallel combinatorial type!

## 5.3 Coin sorting

All ALBERICI coin selectors has capability of driving standard(*simple*) electromagnetic coin sorters. Our programming software will automatically program all necessary data values when sorter type is selected. User must be aware that outputs used for coil drive is not possible to use by any other function. If outputs are already programed for another function, the sorter option will not be allowed. Beside all available sorter types custom programming is possible.
Usually sorter coil drive use outputs 5(*pin 3*) and 6(*pin 4*) on 10 pole connector.
Some time output 4(*pin 10*) is used if 3 coil sorter is selected.
theoretically all 6 output is possible to use as sorter coil drive but existing sorter types has maximum of 5 way sorting that could be achieved with 3 coils!
Basic ALBERICI sorter types are:

- 2-way sorter SA2
- 3-way sorter SA3
- 3-way overlayed sorter NS3
- 3-way linear sorter NL3
- 3-way linear sorter VARIANT
- 3-way overlayed sorter VARIANT
- 5-way sorter VARIANT

### 5.3.1 2-way sorter SA2

Sorter **SA2** is the most simple and fast setting sorter, that use only one output(*pin 4*) for coil A drive. Direction **1** is usually named "cash box" and for that direction coil is not active!

### 5.3.2 3-way sorter SA3

Sorter **SA3** is older version of coin sorter with support for "V" version of coin selectors.
Output 6(*pin 4*) is used to drive coil A and output 5(*pin 3*) is used to drive coil B.
Both coil must not be active at same time. In that case sorting direction will be unpredictable or coin will jam. When this sorter is selected by our program software, automatic protection and correct timing for coil activity is set. Coil activation and timing value are shown in table 5.5. Physical sorting paths are shown on image 5.1.

### 5.3.3 3-way overlayed sorter NS3

Sorter **NS3** is possible to use with all types of coin selectors. For panel mounted coin selectors type "K" and "S", **SEPA-002K** adapter must be used.
Overalyed means that two coin outlets from sorter are in same line[14]. When sorter is used with "K" or "S" type this has no meaning because path of rejected coin is not through sorter but through front plate.
Coil activation and timing value are shown in table 5.5. Physical sorting paths are shown on image 5.2.

[14] See image 5.2

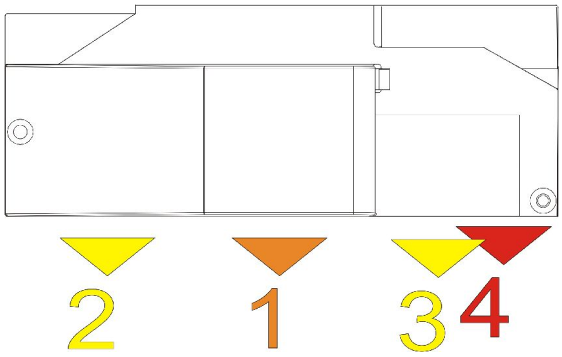


Image 5.2a NS3 sorter side view

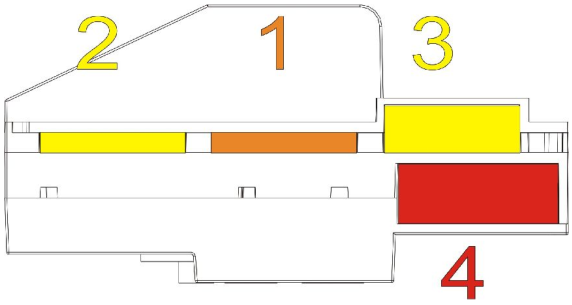


Image 5.2b NS3 sorter bottom view

### 5.3.4 3-way linear sorter NL3

Sorter **NL3** is used only with "V" version of coin selectors. All coin outlet including rejected coin outlet are in same line.

Coil activation and timing value are shown in table 5.5. Physical sorting paths are shown on image 5.3.

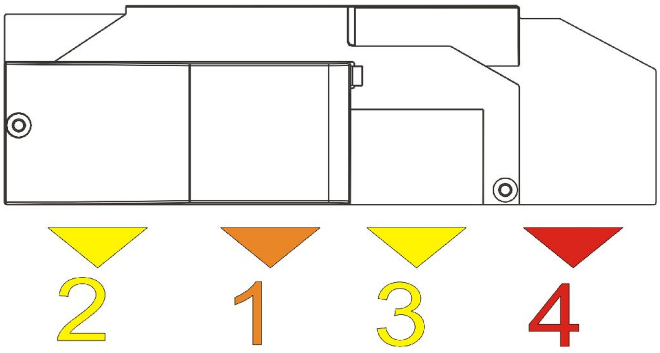


Image 5.3a NL3 sorter side view

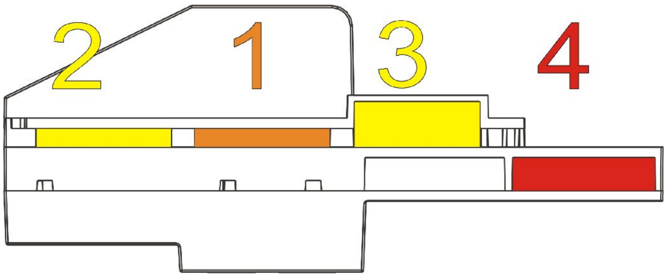


Image 5.3b NL3 sorter bottom view

### 5.3.5 3-way linear sorter VARIANT

VARIANT sorters are last generation of ALBERICI sorters that has same shape but different sorting "variants"[15]. Rejected coin do not pass through this type of sorter and it make this sorter universal for use with all type of coin selectors.
As all other 3-way sorters, coil A and B are used on same outputs as in models NS3 or NL3. Coil activation and timing value are shown in table 5.5. Physical sorting paths are shown on image 5.4.

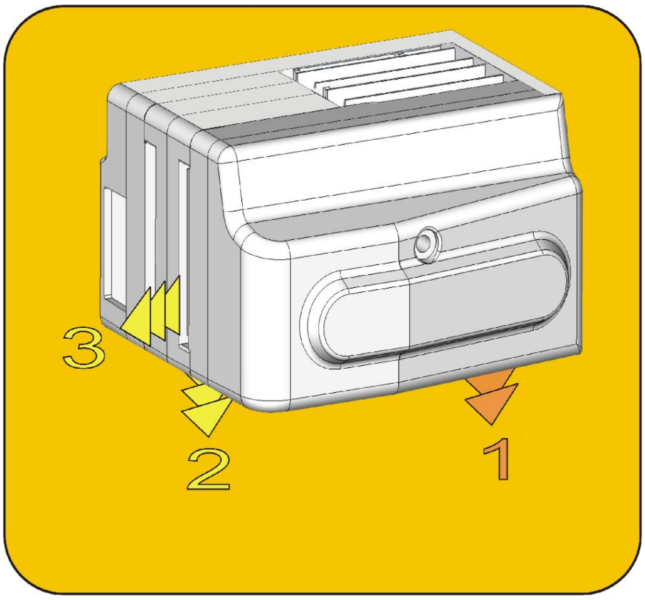


Image 5.4 3-way linear VARIANT sorter

This type of sorter has yellow case.

[15] That is why name "VARIANT" was chosen

### 5.3.5 3-way overlayed sorter VARIANT

This sorter has outlet for coin direction 2 and 3 in same line. All other characteristics are same as for linear sorter. Coil activation and timing value are shown in table 5.5. Physical sorting paths are shown on image 5.5.

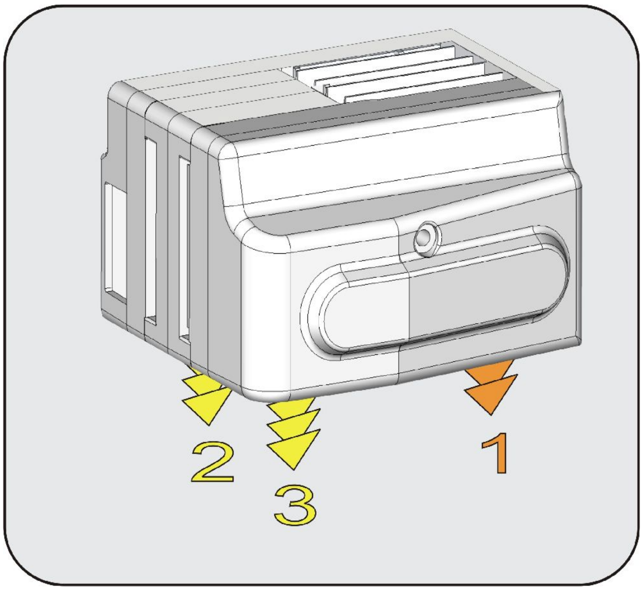


Image 5.5 3-way overlayed VARIANT sorter

### 5.3.6 5-way sorter VARIANT

Five way VARIANT sorter use 3 coils drive for sorting mechanism.
Coil C is connected to output 4(*pin 10*) of coin selector. Coil activation and timing value are shown in table 5.5. Physical sorting paths are shown on image 5.6.

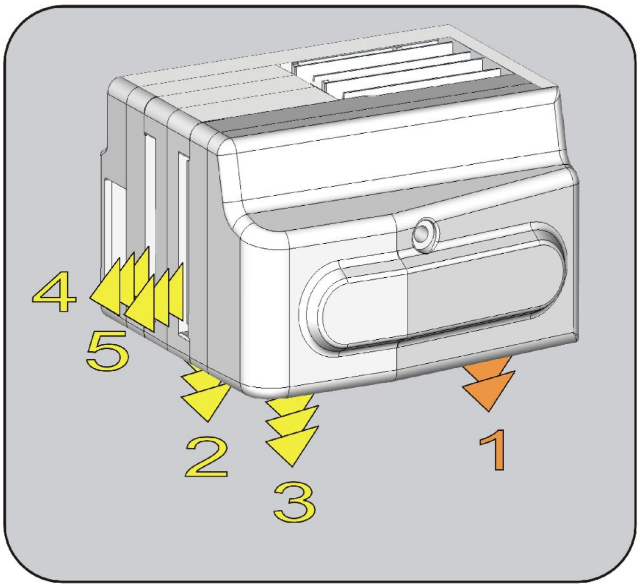


Image 5.6 5-way VARIANT sorter

| **2-way SA2** | - |  | - |  | Coil A |  |
|---|---|---|---|---|---|---|
| Sorting dir. | - | - | Out 5 | Time | Out 6 | Time |
| Dir 1 | - | - | - | - | Off | - |
| Dir 2 | - | - | - | - | On | 400 |
| **3-way *** | - |  | Coil B |  | Coil A |  |
| Sorting dir. | - | - | Out 5 | Time | Out 6 | Time |
| Dir 1 | - | - | Off | - | Off | - |
| Dir 2 | - | - | Off | - | On | 400 |
| Dir 3 | - | - | On | 400 | Off | - |
| **5-way VAR** | Coil C |  | Coil B |  | Coil A |  |
| Sorting dir. | Out 4 | Time | Out 5 | Time | Out 6 | Time |
| Dir 1 | Off | - | Off | - | Off | - |
| Dir 2 | Off | - | Off | - | On | 400 |
| Dir 3 | On | 500 | On | 400 | Off | - |
| Dir 4 | On | 500 | Off | - | On | 400 |
| Dir 5 | Off | - | On | 400 | Off | - |

Table 5.5 Sorters coil activation

**Note ***
All types of 3-way sorters SA3, NS3, NL3 and 3-way VARIANT

## 5.4 Displays

Coin selector ALBERICI AL66 has separate connector for different SPI[16] device(*displays*) connection.
ALBERICI displays **SCHED660** with 6 digits LED, **SCHED670** with 8 digits LCD and the older version of displays COMESTERO type RM3 with 4 digits or NRI with 5 digits is possible to connect. Selection of display type and presentation is linked to coin selector configuration. Following values are possible to present:

- accumulated credits
- accumulated value or rest value
- time(elapse)
- messages

### 5.4.1 Accumulated credits

Display of accumulated credits is possible to select only with coin selectors that are able to accumulate credits. Such types are: totalizer on request or timer on request.
Displays with 6 or 8 digits, use the last 2 most significant digits to present credit.
If display has 5 digits, one last most significant digit is used and number of presented credit is limited to 9! If credit number is over that number error message[17] will be displayed but coin selector will continue to work properly.
On displays with 4 digits, last significant digits are used, but than other values are not possible to present at same time!

### 5.4.2 Accumulated value

Accumulated value or rest value are displayed on lower 4 digits. Coin value in coin selector are programmed mostly with relative value and some time this value must be multiplied to have realistic absolute presentation. For smaller coin value(i.e. Cents) decimal point must be programmed on right position. Default value for multiplier is 1 and decimal point is set after second digit. If second most significant digit is zero it will be blank. Example:

In case we must present import value of euro coins with programmed value for 50 cent coin equal 1, than multiplier must be set to 50 and decimal point to 2. After insertion of 3 coins, presented value will be:
multiply 3x50=150
set decimal point on second place
display: **1.50**.

### 5.4.3 Time

Time till the end of timer output activity(elapse time) will be presented from the moment when timer output is active till the end of activity. Display format is:

| | |
|---|---|
| for most displays | **mm.ss** |
| 8 digit LCD | **mm ss** |

Where **mm** represent time in minutes **from** 00 to 99 minutes, while **ss** is time in

[16] **S**erial **P**eripheral **I**nterface
[17] See chapter 5.4.4 Messages

seconds from 00 to 59. Minute and second count are separated with decimal point except in case of 8 digit LCD where one blank digit is used as separation mark.
Time and imported value or rest value is not possible to present at the same time

### 5.4.4 Messages

Currently only error messages of display are supported.
In case that value on display is not possible to present(*out of range*) display will show character **E**. Coin selector function will be correct. Example of error on display is when number of credit to be displayed is more than 9 credits and we use display with only one digit. Next example is in case of very long time period over 99 min and 59 seconds which is theoretically possible to program if time multiplier and base time is big enough!
Display with 8 digits has 12 segments per digit and some advance text message will be possible to present in future or on customer request!

## 5.5 Rest management

The rest management is additional function available to use in all types of coin selectors when there is some rest of accumulated value after credit activation or vend process. Usually is used in dual stepper or multi price stepper or some time even with timer or totalizator. User can select different options: to cancel the rest value(*set time to 0*), to keep it permanently(*till power off*) or to cancel this value after time period that could be programmed from 1 to 254 seconds.
Default setting is to keep rest permanently(*timer set to 255*).
After first coin insertion timer will be stopped and restarted again after next credit or vending activation.

## 5.6 Fraud inhibit

ALBERICI coin selectors has advanced control of coin acceptance and protection against the all kinds of fraud. The most common fraud is called "jojo" or coin on string. First of all there is mechanical protection with wire cutter and anti-fishing hooks. Some models (AL66K3) has 3 opto sensors for coin position and direction control.
Precise optical pulse timing measure is done to distinguish false movement of coin. Deviation in direction or timing limits will be recognized as fraud and coin will not be accepted as valid(*no credit*).
To make fraud attempt much difficult after fraud detection it is possible to block acceptance for programmed time period(from 0 to 254 seconds). If time period is programmed with value 255 than inhibition of acceptance will be active till next coin insertion. Next coin there fore will be rejected.

# 6. Serial communication

New generation of coin selectors AL55 or AL66 use **cctalk®** communication protocol. This protocol was developed by company Money Controls(ex. Coin Controls) to enable connection of different peripheral devices[18] in smal network.
Protocol is mostly used in gaming and casino machines but it can be implemented in any other types of machines that use same type of devices.
It is public protocol and free to use. Finde documentation on: www.cctalk.org.

**Communication protocol of ALBERICI coin selectors AL55/66 comply to generic specification 4.4**

## 6.1 Communication specifications

Serial communication was derivated from RS232 standard.
It is low data rate NRZ (***N**on **R**eturn to **Z**ero*) asyncronous communication with:
Baud rate 9600, 1 start bit, 8 data bits, no parity, 1 stop bit.
RS232 handshaking signals (*RTS, CTS, DTR, DCD, DSR*) are not suported.
Mesage integrity is controled by means of checksum calculation.

### 6.1.1 Baud rate

The baud rate of 9600 was chosen as compromise betwen cost and speed.
Timing tolerances is same as in RS232 protocol and it should be less than 4%.

### 6.1.2 Voltage level

To reduce the costs of connections the "Level shifted " version of RS232 is used.
The idle state on serial connector is 5V, and active state is 0V.

| | | |
|---|---|---|
| Mark state (*idle*) | +5V nominal | from 3.5V to 5V |
| Space state (active) | 0V nominal | from 0.0V to 1.0V |

Data I/O line is "open collector" type, so it is possible to use device in systems with different voltage (*12V pull up in older devices*).

### 6.1.3 Connection

The connection of Coin selector at network is achieved by means of 4 pole JST connector (*standard type 7*). Connector is used for power supply and communication as well. For schematics and and connector appirance see image1.

[18] Coin selectors, Hoppers(*pay out device*), Banknote readers etc.

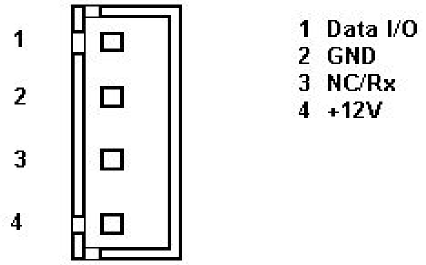


Image 6.1 communication connector

Recomanded periferial connector is:
**JST B 4B-XH-A with crimping contacts SXH-001T-P0.6**

## 6.2 Mesage structure

Each communication sequence consists of two mesage string.
Mesage string in case of simple checksum use is structured as folows:

**[ Destination address ]**
**[ Nr. of data bytes ]**
**[ Source address ]**
**[ Header ]**
**[ Data 1 ]**
**...**
**[ Data n ]**
**[ Checksum ]**

There is an exeption of mesage structure when device respond to instruction Address poll and Address clash[19]. The respond consists of only one byte representing address delayed for time proportional to address value. For CRC checksum case format is:

**[ Destination address ]**
**[ Nr. of data bytes ]**
**[ CRC 16 LSB ]**
**[ Header ]**
**[ Data 1 ]**
**...**
**[ Data n ]**
**[ CRC 16 MSB ]**

### 6.2.1 Address

Address range is from address 0 to address 255. Address 0 is special case or so caled “brodcast” address and address 1 is default host address.
The recomandations for address value of different devices are presented in table 1.

[19] For details see cctalk44-2.pdf, Address poll

| Device category | Address | Add. addr. | Note |
|---|---|---|---|
| **Coin Acceptor** | **2** | **11 - 17** | **Coin validator, selector,** |
| Payout | 3 | 4 - 10 | Hopper |
| Reel | 30 | 31 - 34 | |
| Bill validator | 40 | 41 - 47 | Banknote reader |
| Card Reader | 50 | | - |
| Display | 60 | | Alphanumeric LC display |
| Keypad | 70 | | - |
| Dongle | 80 | 85 - 89 | Safety equipment |
| Meter | 90 | | Replacement for el.mec. counters |
| Power | 100 | | Power supply |
| Printer | 110 | | Ticket printing |
| RNG | 120 | | Random Number Generator |

Table 6.1 Standard address for different types of devices

Address for ALBERICI coin selectors AL55/66 is factory set at value **2**.
User can change the default address using MDCES instructions:
**Address change** or **Address random**.

### 6.2.2 Number of data byte

Number of data byte in each transfer could be from 0 to 252.
Value 0 means that there are no data bytes in the mesage, and total lenght of mesage string will be 5 bytes. Although theorettcally it will be possible to send 255 bytes of data becouse of some limitations in small micro controlers the number is limitet to 252[20].

### 6.2.3 Command headers (*Instructions*)

Total amount of possible cctalk command header is 255 with possibility to add sub-heaers using headers 100, 101, 102 and 103.
**Header 0** stands for **ACK** (***ack**nowledge*) replay of device to host.
**Header 5** stands for **NAK** (***N**o **ack**nowledge*) replay of device to host.
**Header 6** is **BUSY** replay of device to host.
In all three cases no data bytes are transfered. Use of ACK and NAK headers are explained later on, for each specific mesage transfer.
Commands are devided in to several groups acording to aplication specifics:

- Basic general commands
- Additional general commands
- Commands for Coin acceptors
- Commands for Bill validators
- Commands for Payout mechs
- MDCES commands

**ALBERICI Coin selectors AL55/66 use total of 55[21] instructions-headers**.
Details of all instruction use are explained in chapter 2.

[20] 252 bytes of data, source address, header and checksum (total of 255 bytes)

[21] Some type of coin selectors do not support all headers

### 6.2.4 Data

There is no limitation in use of data formats. Data could be BCD (***B**inary **C**oded **D**ecimal*)numbers, Hexa numbers or ASCII strings. Intrepretation as well as format is specific to each header use, and will be explained in separate chapter.

### 6.2.5 Checksum

Mesage integrity during transfer is checked by use of simple zero checksum calculation. Simple checksum is made by 8 bit addition (modulus 256) of all the bytes in the mesage. If mesage is recieved and the addition of all bytes are non-zero then an error has occurred[22].

For noisy enviroment or higher security aplication it is possible to use more complex, 16 bit CRC CCITT checksum based on a polynomial of:

$\mathbf{x^{16} + x^{12} + x^{5} + 1}$ and initial value of CRC register **0x0000**.

Coin selectors AL55/66 are using simple checksum, but they could be set to operate with CRC-16 checksum on customer demand.

## 6.3 Timing specification

The timing requierments of cctalk are not very critical but there are some important recomandations.

### 6.3.1 Time beetwen two bytes

When reciving bytes within a mesage string, the comunication software must wait up to **50 ms** for next byte if it is expected. If time out occurs, the software should reset all communication variables and get ready to recieve next mesage. The interbyte delay during transmition should be ideally **less than 2 ms** and **not greater than 10 ms**.

### 6.3.2 Time beetwen comand and replay

The time beetwen comand and reply is dependent on aplication specific for each comand. Some comands return data imediatly, and maximum time delay should be within **10 ms**.

Others comands that must activate some actions in device, may return reply after the action is finished[23].

### 6.3.3 Start-up time

After the power-up sequence coin selector should be ready to accept and respond to a cctalk message within time period of **less than 250 ms**.

During that period all internal check-up and system settings must be done, and coin acceptor should be able to recognize and accept coins.

[22] See Error handling

[23] I.e. more than 100 mili sec for solenoid testing.

## 6.4 Error handling

If slave device receive the message with bad checksum or missing data no further action is taken and receive buffer will be cleared.
Host software should decide to re-transmit message immediately or after a fixed amount of time. In case when host receive message with error it has same options.

## 6.5 Command headers

Command header set, that host could use in communication with coin selectors AL55 and AL66 is given in the table 6.2.
Command headers are divided in to 3 different groups:

- Common command headers
- Coin acceptor command headers
- MDCES command headers

| Code | | Command header | Note |
|---|---|---|---|
| 255 | FF | Factory specific test | Supported |
| 254 | FE | Simple poll | Return ACK |
| 253 | FD | Address poll | MDCES support |
| 252 | FC | Address clash | MDCES support |
| 251 | FB | Address change | MDCES support, non volatile |
| 250 | FA | Address random | MDCES support, non volatile |
| 249 | F9 | Request polling priority | [02][32] 100x50=500 ms |
| 248 | F8 | Request status | [00] Ok |
| 246 | F6 | Request manufacturer id | 'Alberici' |
| 245 | F5 | Request equipment category id | 'Coin Acceptor' |
| 244 | F4 | Request product code | 'ALNNxn' NN=55/66, x=V/I/K, n=1-3 |
| 243 | F3 | Request database version | [01] remote file programming |
| 242 | F2 | Request serial number | From 0 to 16.777.215 |
| 241 | F1 | Request software revision | 'u3.n p3.m' n=0..9, m=0..9 |
| 240 | F0 | Test solenoids | Coil on for 100 ms |
| 238 | EE | Test output lines | Supported |
| 237 | ED | Read input lines | [In1=MSb,DIP-sw1][In2=MSb,DIP-sw2] |
| 236 | EC | Read opto states | bit0=opto1, bit1=opto2 |
| 233 | E9 | Latch output lines | Supported |
| 232 | E8 | Perform self test | Supported |
| 231 | E7 | Modify inhibit status | [inhibit 1][inhibit 2] total 16 position, volat. |
| 230 | E6 | Request inhibit status | Supported |
| 229 | E5 | Read buffered cred. or error c. | Five two byte event buffer |
| 228 | E4 | Modify master inhibit status | bit0=0 inhibited ..1=enable, volatile |
| 227 | E3 | Request master inhibit status | Supported |
| 226 | E2 | Request insertion counter | [Rjct1-MSB][ Rjct2][ Rjct3-LSB] |
| 225 | E1 | Request acceptance counter | [Rjct1-MSB][ Rjct2][ Rjct3-LSB] |
| 221 | DD | Request sorter override status | [FF] Normal sorting |

| | | | |
|---|---|---|---|
| 219 | DB | Enter new PIN number | Supported, non volatile |
| 218 | DA | Enter PIN number | ACK return if PIN is correct |
| 216 | D8 | Request data storage availability | [00][00][00][00][00], not available |
| 215 | D7 | Read data block | For encrypted data exchange! |
| 214 | D6 | Write data block | For encrypted data exchange! |
| 213 | D5 | Request option flags | bit0=0 cred. code format position |
| 210 | D2 | Modify sorter paths | [coin pos][path], volatile |
| 209 | D1 | Request sorter paths | Supported |
| 202 | CA | Teach mode control | Supported |
| 201 | C9 | Request teach status | Supported |
| 197 | C5 | Calculate ROM checksum | [ROM-H][ROM-L][EEPR-H][EEPR-L] |
| 196 | C4 | Request creation date | Supported |
| 195 | C3 | Request last modification date | Supported |
| 194 | C2 | Request reject counter | [Rjct1-MSB][ Rjct2][ Rjct3-LSB] |
| 193 | C1 | Request fraud counter | [Frd1-MSB][ Frd2][ Frd3-LSB] |
| 192 | C0 | Request build code | 'AL66 V1.0' |
| 188 | BC | Request default sorter path | [01] No sorting |
| 185 | B9 | Modify coin id | Supported |
| 184 | B8 | Request coin id | Supported |
| 176 | B0 | Request alarm counter | Supported, one byte cumulative count |
| 173 | AD | Request thermistor reading | If thermistor is mounted |
| 170 | AA | Request base year | '2000' |
| 169 | A9 | Request address mode | [84] addr.change Non volatile(FLASH) |
| 4 | 04 | Request comms revision | [02][04][02] ,level2, isue4.2 |
| 3 | 03 | Clear comms status variables | Supported |
| 2 | 02 | Request comms status variables | [Rx timeout][ Rx b. ignored][ Rx bad chks.] |
| 1 | 01 | Reset device | Software reset |

Table 6.2 cctalk instruction header list

### 6.5.1 Common command headers

Common commands are used in all type of devices to detect there presence on cctalk network or to describe them. Information like: manufacturer or product type id, serial number, different settings etc. are transmitted to host.

#### 6.5.1.1 Command 254 [hexFE], Simple poll

The fastest way for host to detect all attached devices in cctalk network.
Addressed device-coin selector respond with ACK (*Acknowledge*).
If within predicted amount of time Coin selector does not respond coin selector is probably not connected, powered or simple not working properly.
Message format is:

Host sends: **[Dir] [00] [01] [FE] [Chk]**

Coin s. respond: **[01] [00] [Dir] [00] [Chk]**

As coin selector default address is 2, example of message string is:

Host sends: **[02] [00] [01] [FE] [FF]**

Coin s. respond: **[01] [00] [02] [00] [FD]** ACK mesage

### 6.5.1.2 Command 246 [hexF6], Request manufacturer ID

Coin selector respond with ASCII string representing manufacturer name.
Message format is:

Host sends: **[Dir] [00] [01] [F6] [Chk]**

Coin s. respond: **[01] [Nr.b] [Dir] [00] [a1] [a2] . . . . [an] [Chk]**

Nr.b is number of data bytes-characters sent by coin selector, and a1 to an are ASCII characters. For **Alberici** coin selector example of message string is:

Host sends: **[02] [00] [01] [F6] [07]**

Coin s. respond: **[01] [08] [02] [00] [41][6C][62][65][72][69][63][69] [DA]**

### 6.5.1.3 Command 245 [hexF5], Request equipment category ID

Respond to command header is standardized name for coin selectors, coin validators or coin mechs. Coin selector respond with ASCII string of characters representing standardized name for that type of device **Coin Acceptor**.
Message format is:

Host sends: **[Dir] [00] [01] [F5] [Chk]**

Coin s. respond: **[01] [0D] [Dir] [00] [43][6F][69][6E][20][41][63][63][65][70][74][6F][72] [Chk]**

Number of data byte is always 13, hex [0D].
Example of message string for coin selector(*address 2*) is:

Host sends: **[02] [00] [01] [F5] [08]**

Coin s. respond: **[01] [0D] [02] [00] [43][6F][69][6E][20][41][63][63][65][70][74][6F][72] [16]**

### 6.5.1.4 Command 244 [hexF4], Request product code

Coin selector respond with ASCII string of character, representing the factory type of coin selector. For ALBERICI coin selectors of new generation possible response will be:

- **AL55V1, AL55K1, AL55I1**
- **AL66V2, AL66K3, AL66I3**

In special version for italian gambling machines response is allways **AL05V-c** .

Host sends: **[Dir] [00] [01] [F4] [Chk]**

Coin s. respond: **[01] [07] [Dir] [00] [a1][a2] . . . [a7] [Chk]**

Number of data bytes sent by coin selector is 6 or 7, hex [07].
Example of message string for coin selector(*address 2*) type **AL06V-c** is:

Host sends: **[02] [00] [01] [F4] [09]**

Coin s. respond: **[01] [07] [02] [00] [41][4C][30][36][56][2D][63] [1D]**

### 6.5.1.5 Command 242 [hexF2], Request serial number

Coin selector respond with three byte serial number. Message format is:

Host sends: **[Dir] [00] [01] [F2] [Chk]**

Coin s. respond: **[01] [03] [Dir] [00] [Serial 1 - LSB] [Serial 2] [Serial 3 - MSB] [Chk]**

Serial 1 – first data byte sent is LSB of serial number.
Example of message string for coin selector(*address 2*) with serial number: **1234567** (hex [BC][61][4E]) is:

Host sends: **[02] [00] [01] [F2] [0B]**

Coin s. respond: **[01] [03] [02] [00] [4E][61][BC] [8F]**

### 6.5.1.6 Command 241 [hexF1], Request software revision

Coin selector return ASCII string of character representing software version and revision. Message format is:

Host sends: **[Dir] [00] [01] [F1] [Chk]**

Coin s. respond: **[01] [Nr.b] [Dir] [00] [a1] [a2].... [an] [Chk]**

Number of data bytes in ASCII string is not limited and each producer has it's own system of labeling. Example of message string for coin selector(*address 2*) is:

Host sends: **[02] [00] [01] [F1] [0C]**

Coin s. respond: **[01] [09] [02] [00] [75][31][2E][30][20][70][31][2E][30][2E][30] [71]**

Coin selector respond is ‘**u1.0 p1.0.0**′.
ALBERICI coin selectors has program firmware label divided in two parts.
First label **u** is for protected FLASH memory program(*monitor program*) revision.
First digit is for major changes and second for minor changes. In this case it is **u1.0**.
Second label is revision of main program FLASH memory.
Main program software revision labeling use 3 digits. First most significant digit is for major software changes, second is for minor software changes and third for "bug" correction. In this case it is **u1.0.0**.

### 6.5.1.7 Command 197 [hexC5], Calculate ROM checksum

Coin selector respond with four bytes of micro controller internal memory checksum. First two bytes are program ROM CRC and the second is data EEPROM CRC. Any changes in program or data will change the respond of coin selector.
Message format is:

Host sends: **[Dir] [00] [01] [C5] [Chk]**

Coin s. respond: **[01] [4] [Dir] [00] [CRC1-H][CRC1-L] [CRC2-H] [CRC2-L] [Chk]**

Example of message string for coin selector(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [C5] [38]** |
| Coin s. respond: | **[01] [04] [02] [00] [D9][2A][7E][79] [96]** |

### 6.5.1.8 Command 192 [hexC0], Request build code

Coin selector respond with ASCII string of character representing it's hardware version and revision. Last revision of printed circuit board for coin selectors AL55/66 is: **AL66 V1.0**. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [C0] [Chk]** |
| Coin s. respond: | **[01] [Nr.b] [Dir] [00] [a1] [a2].... [an] [Chk]** |

Example of message string for coin selector(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [C0] [3D]** |
| Coin s. respond: | **[01] [09] [02] [00] [41][4C][2D][30][35][20][56][35][30] [FA]** |

### 6.5.1.9 Command 169 [hexA9], Request address mode

Coin selector respond with one byte data[24] information about addressing mode. Address could be stored in different type of memory(*RAM. ROM or EEPROM*), set with DIP-switch at printed circuit board or hard-wired at external connectors. Some devices support address change wit MDCES command headers[25]. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [A9] [Chk]** |
| Coin s. respond: | **[01] [01] [Dir] [00] [Address mode] [Chk]** |

ALBERICI coin selector has address is stored in non-volatile memory(*FLASH*) and address change is supported.
Example of message string for coin selectors(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [A9] [54]** |
| Coin s. respond: | **[01] [01] [02] [00] [84] [78]** |

### 6.5.1.10 Command 4 [hex04], Request comms revision

Coin selector respond with three byte data information about level of cctalk protocol implementation, major and minor revision. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [04] [Chk]** |
| Coin s. respond: | **[01] [03] [Dir] [00] [Level] [Mag.rev.] [min. rev.] [Chk]** |

Example of message string for coin selector(*address 2*) with level of implementation **1**, cctalk protocol issue **4.4** is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [04] [F9]** |
| Coin s. respond: | **[01] [03] [02] [00] [01][04][04] [F1]** |

[24] Details of description see in public document cctalk44-2.pdf

[25] Address change, Address random

### 6.5.1.11 Command 3 [hex03], Clear comms status variables

After acceptance of command header 3, coin selector clears all three bytes of communication errors counters and respond with ACK message. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [00] [01] [03] [Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK mesage |

Example of message string for coin selector(*address 2*) is:

| | | |
|---|---|---|
| Host sends: | **[02] [00] [01] [03] [FA]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK mesage |

### 6.5.1.12 Command 2 [hex02], Request comms status variables

Coin selector respond with three byte data representing communication errors.
First byte is receive time out counter, second byte is number of ignored receive bytes[26] and third byte is number of checksum errors. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [02] [Chk]** |
| Coin s. respond: | **[01] [03] [Dir] [RxErr1] [RxErr2] [RxErr3] [Chk]** |

Example of message string for coin selector(*address 2*) with no errors is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [02] [FB]** |
| Coin s. respond: | **[01] [03] [02] [00] [00] [00] [00] [FA]** |

### 6.5.1.13 Command 1 [hex01], Reset device

After acceptance of command Reset, coin selector execute software reset and clear all variables in RAM or set them at the default value, including different counters, and credit buffer. ACK message will be sent before reset of coin selector.
Host software must set again:

- inhibit state
- sorter path
- master inhibit (*if necessary*)

Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [00] [01] [01] [Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK mesage |

Example of message string for coin selector(*address 2*) is:

| | | |
|---|---|---|
| Host sends: | **[02] [00] [01] [01] [FC]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK mesage |

Host software must wait at least **100 ms**, to continue comunication with coin selectro after reset instruction!

[26] Number of receive buffer overflow bytes.

### 6.5.2 Coin acceptor specific command headers

Coin selectors use some specific commands, mostly for control of coin input, acceptance and direction[27].
Some of commands are shared with other device like banknote reader or payout device.

#### 6.5.2.1 Command 249 [hexF9], Request polling priority

Basic principle of detecting credit input or eventual errors from coin selector is sequential polling[28]. Coin selectors due to differences in mechanical and electrical construction has different acceptance speed. All events are registered in memory buffer with limited size[29] . To avoid credit loss, host must read coin selector credit buffer within limited time period. Coin selector has internal mechanism to block the coin acceptance and registration of all events if polling time elapse.
For ALBERICI coin selector acceptance speed is from 3 to 4 coins per second[30].
Considering that it is possible to register 5 event in the buffer, the adequate polling time will be about 1 sec. Because of necessity to register even "close" and non accepted coins polling time must be even shorter.
For ALBERICI coin selectors AL55/66 using cctalk interface, poll time is set to 500 ms.
Coin selectors that use standard 10 pole interface are not necessary to poll.
In that case polling time unit is set to 0(*no polling*)!
Minimum time for polling must not be shorter than overall message time[31].
Coin selector respond to command with two bytes of data. First byte is poll time unit and second is polling time value[32]. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [F9] [Chk]** |
| Coin s. respond: | **[01] [01] [Dir] [Time] [Chk]** |

Example of message string for coin selector(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [F9] [04]** |
| Coin s. respond: | **[01] [02] [02] [00] [02] [32] [C7]** |

First byte **02** is unit **x10ms** , and second byte is time value **hex32 = 50**.
Polling time is calculated as:

**T = 10 x 50 = 500 ms**

#### 6.5.2.2 Command 248 [hexF8], Request Status

ALBERICI coin selectors has no additional COS[33] and return mechanism.
Response to that command is always hex**[00]**, coin selector Ok.
Example of message string for coin selector(*address 2*) is:

[27] Sorter control commands

[28] Reading memory buffer from coin selector

[29] Five stage double byte memory buffer

[30] Dependant on mechanical type of coin selector (K, S type is faster ) and coin

[31] For coin selector with respond time 2 ms and byte gap 1 ms it is 38 ms

[32] For details see, cctalk44-2.pdf

[33] **C**oin **O**n **S**tring

Host sends: **[02] [00] [01] [F8] [05]**

Coin s. respond: **[01] [01] [02] [00] [00] [FC]**

### 6.5.2.3 Command 243 [hexF3], Request data base version

The respond to that command is version of coin data base. Version of data base is important for coin selectors with remote programming support.
For all ALBERICI coin selectors type AL55/66 current data base version is 00.
Message format is:

Host sends: **[Dir] [00] [01] [F3] [Chk]**

Coin s. respond: **[01] [01] [Dir] [Ver.] [Chk]**

Example of message string for coin selector(*address 2*) is:

Host sends: **[02] [00] [01] [F3] [0A]**

Coin s. respond: **[01] [01] [02] [00] [00] [FC]**

### 6.5.2.4 Command 240 [hexF0], Test solenoids

Host sends one byte mask to determinate which solenoid must be tested.
Coin selector accept gate solenoid or sorter solenoid will be switched on for period of 100 ms and after that, ACK message will be transmitted. Message format is:

Host sends: **[Dir] [01] [01] [F0] [Mask.] [Chk]**

Coin s. respond: **[01] [00] [Dir] [00] [Chk]** ACK

Example of message string for coin selector(*address 2*) acceptance gate test is:

Host sends: **[02] [01] [01] [F0] [01] [0B]**

Coin s. respond: **[01] [00] [02] [00] [FD]** Single click -> 100 ms, ACK

Bit position for output that is used to drive sorter coil are:

bit 0 = accept gate coil
bit 1 = sorter coil "A"(out 6/pin 4)
bit 2 = sorter coil "B"(out 5/pin 3)
bit 3 = sorter coil "C"(out 4/pin 10)

If output selected with bit in mask is not programed for sorter activation it will not be activated but coin selector will still response with ACK.

### 6.5.2.5 Command 238 [hexEE], Test output lines

Host sends one byte mask to determinate which output line must be tested.
Coin selector output line that correspond to bit set in the mask will be pulsed for 100 ms and after that, ACK message will be transmitted. Message format is:

Host sends: **[Dir] [01] [01] [EE] [Mask.] [Chk]**

Coin s. respond: **[01] [00] [Dir] [00] [Chk]** ACK

Example of message string for coin selector(*address 2*) first output(*pin7*) is:

Host sends: **[02] [01] [01] [EE] [01] [0D]**

Coin s. respond: **[01] [00] [02] [00] [FD]** Single pulse out 1 -> 100 ms, ACK

Bit positions for output test are:

| | |
|---|---|
| - bit 0 | Output 1(*pin 7*) |
| - bit 1 | Output 2(*pin 8*) |
| - bit 2 | Output 3(*pin 9*) |
| - bit 3 | Output 4(*pin 10*) |
| - bit 4 | Output 5(*pin 3*) |
| - bit 5 | Output 6(*pin 4*) |
| - bit 6 | Output 7(*pin 5*) |
| - bit 7 | Not used |

Unused output(*not programed*) will not be turned on, but message ACK will be returned.

### 6.5.2.6 Command 237 [hexED], Read input lines

Coin selector respond with two data byte representing the state of DIP-switches and state of inputs In1(*pin 6*) and In2(*pin 5*)[34].
ALBERICI coin selectors has one or two banks of DIP-switches for various data or operating modes setting. First data byte is state of first DIP-switch(*bank 1*) and In1, wile second represent second DIP-switch(*bank 2*) and In2. LSb is first switch in bank and MSb is state of input. Switch closed state is represented with logic "1", and input active state is logic "1". Message format is:

Host sends: **[Dir] [00] [01] [ED] [Chk]**

Coin s. respond: **[01] [02] [Dir] [Mask1] [Mask2] [Chk]**

Example of message string for coin selector(*address 2*), with all switches "off" and inputs not active is:

Host sends: **[02] [00] [01] [ED] [10]**

Coin s. respond: **[01] [02] [02] [00] [00] [00] [FB]**

Example of message string for coin selector(*address 2*), with all switches "on" and input 1(*inhibit acceptance*) active is:

Host sends: **[02] [00] [01] [ED] [10]**

Coin s. respond: **[01] [02] [02] [00] [BF] [00] [3C]**

[34] If In2 is programmed as input

### 6.5.2.7 Command 236 [hexEC], Read opto states

Coin selector respond with one data byte representing the state of opto pairs. ALBERICI coin selectors has up to 3 pairs of optical sensor[35] for detection of coin position, speed and direction and 2 pairs of opto sensors for diameter measurement. Bit position for opto pairs are:

| | |
|---|---|
| - bit 0 | Diam. measure opto 1 |
| - bit 1 | Diam. measure opto 2 |
| - bit 2 | Control opto 1 |
| - bit 3 | Control opto 2 |
| - bit 4 | Control opto 3 |
| - bit 5 | Not used |
| - bit 6 | Not used |
| - bit 7 | Not used |

Control opto sensor 2 is called "credit" opto sensor exist in all version of coin selectors and it is placed after the acceptance gate. Other pairs are optional and some coin selectors has 2 and some 3 control optical pairs. Number of control pairs make part of coin selector type label. For example coin selector type AL66V**2** has 2 control opto sensor pairs. The unused bits or non existing optical sensors are always read as 0. Interruption of light barrier of opto sensor correspond to bit value 1.
Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [EC] [Chk]** |
| Coin s. respond: | **[01] [01] [Dir] [Mask.] [Chk]** |

Example of message string for coin selector(*address 2*) with opto sensors cleared is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [EC] [11]** |
| Coin s. respond: | **[01] [01] [02] [00] [00] [FC]** |

### 6.5.2.8 Command 233 [hexE9], Latch output lines

This instruction is similar to instruction 238. Host sends one byte mask to determinate which output line must be activated(*latch*). ACK message will be transmitted immediate.
Coin selector output line that correspond to bit set in the mask will be latched and active till reset or new instruction with bit cleared is sent. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [01] [01] [E9] [Mask.] [Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK |

Example of message string for coin selector(*address 2*) first output(*pin7*) is:

| | | |
|---|---|---|
| Host sends: | **[02] [01] [01] [E9] [01] [12]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | Latch out 1 -> ACK |

[35] In some case group could contain more than one opto pairs

### 6.5.2.9 Command 232 [hexE8], Perform self-test

Coin selector respond to command with one or two bytes of data according to table 6.3. First byte is fault code and second is optional data, usually representing fault sensor number(*from 1 to 3*).

| Code | Fault | Optional data | Comment |
|---|---|---|---|
| 0 | OK No fault detected | - | - |
| 2 | Fault on inductive sensor | Sensor number | - |
| 3 | Fault on credit sensor | - | Control opto sensor 2 |
| 6 | Fault on diameter sensor | - | - |
| 18 | Fault on reject sensor | - | Control opto sensor 3 |
| 33 | Power supply out of limits | - | - |
| 34 | Temperature out of limit | - | Optional |
| 255 | Unspecified fault code | - | - |

Table 6.3 Fault codes for AL55/66 coin selectors

Inductive sensor numbers are:

| | |
|---|---|
| 01 | Upper inductive sensor |
| 02 | First lower inductive sensor |
| 03 | Second lower inductive sensor |

Message format is:

Host sends: **[Dir] [00] [01] [E8] [Chk]**

Coin s. respond: **[01] [01/02] [Dir] [Fault c.][Data opt.] [Chk]**

Example of message string for coin selector(*address 2*) with no fault detected is:

Host sends: **[02] [00] [01] [E8] [15]**

Coin s. respond: **[01] [01] [02] [00] [00] [FC]** No fault detected

Example of message string for coin selector(*address 2*) with first lower sensor fault detected is:

Host sends: **[02] [00] [01] [E8] [15]**

Coin s. respond: **[01] [02] [02] [00] [02][02] [F7]** Fault on first lower sensor detected

### 6.5.2.10 Command 231 [hexE7], Modify inhibit status

With this command host is able to inhibit the acceptance of some or all coins.
Acceptance or inhibition is set with a two byte mask sent by host.
Bits from 0 do 15 determinate coin positions from 1 to 16[36].
Number of coin channels in new ALBERICI coin selectors AL55/66 is same as number of position(*16*). Message format is:

Host sends: **[Dir] [02] [01] [E7] [LSB Mask.] [MSB Mask.] [Chk]**

Coin s. respond: **[01] [00] [Dir] [00] [Chk]** ACK

[36] Positions are sent by coin selector during reading credit buffer or error codes (*header 229*)

Example of message string to enable all position for coin selector(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [02] [01] [E7] [FF] [FF] [16]** |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** ACK |

After that all programmed coins will be enabled. Command has no effect on coin position that are not programmed. Initially coin channels could be programed with acceptance enabled or disabled.

**For coin selectors that are using only cctalk interface, all coins position must be initially inhibited!**

### 6.5.2.11 Command 230 [hexE6], Request inhibit status

Coin selector respond with two byte data that correspond to inhibit state mask for all 16 positions of coin. If bit value is 1 acceptance of coin in that position is enabled. If bit value is 0 coin is inhibited. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [02] [00] [E6] [Chk]** |
| Coin s. respond: | **[01] [02] [Dir] [00] [LSB Mask.] [MSB Mask.] [Chk]** |

Example of message string for coin selector(*address 2*) **AL06V-c**[37] after power-up or reset is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [E6] [17]** |
| Coin s. respond: | **[01] [02] [02] [00] [00] [00] [FB]** |

Example of message string for coin selector(*address 2*) with programmed positions from 1 to 6, after receiving command to enable acceptance of all 16 position is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [E6] [17]** |
| Coin s. respond: | **[01] [02] [02] [00] [3F] [00] [BC]** |

First byte represent the mask for coin positions 1 to 8 and second for 9 to 16.
Coin channels(*positions*) that are not programmed are always represented as zero bit!

### 6.5.2.12 Command 229 [hexE5], Read buffered credit or error codes

This is the most important command used by host to detect import of coins in to a machine and to report eventual errors. As previously mentioned coin selectors store all events in volatile memory called credit buffer. Buffer has 5 level and use two bytes for each event. In first byte coin position or coin value[38] is stored. The second byte point to a sorter path or indicate error code.
If during acceptance of coin an error occur, stored value of coin position will be 0, hex [00].
Error codes supported in ALBERICI coin selectors AL55/66 are shown in table 6.4.

[37] Coin selector for Italian gambling machines

[38] If coin selector use CVF (*Coin Value Format*)

| Code d. | Code h. | Error | Coin rejected |
|---|---|---|---|
| 0 | 00 | Null event | No |
| 1 | 01 | Reject coin (not recognized) | Yes |
| 2 | 02 | Inhibited coin (master inhibit) | Yes |
| 3 | 03 | Multiple window (fraud or similar coin) | Yes |
| 5 | 05 | Validation (measuring) time out | Yes |
| 6 | 06 | Credit sensor (recognition to opto 2) time out | Possible |
| 8 | 8 | Second close coin | Yes/both |
| 16 | 10 | Credit sequence error (Jojo) | No |
| 18 | 12 | Coin to fast (opto 2 minimum time not elapsed) | No |
| 19 | 13 | Coin to slow (opto 2 time out) | No |
| 128 | 80 | Inhibited coin (position 1) | Yes |
| … | … | Inhibited coin (position n) | Yes |
| 143 | 8F | Inhibited coin (position 16) | Yes |
| 255 | FF | Unspecified alarm code | - |

Table 6.4 Acceptance error codes

Coin selectors also use one eight bit counter[39] that is incremented each time a new coin is detected. At the same time data in coin credit buffer are shifted two position to the right. When counter reaches the value of 255 it toggle to a value 1 and continue to increment on each event. Event counter is set to value "0" after each power-up or acceptance of reset command. The first two byte (*LSB*) in coin credit buffer always contain the data of last event. Host software must read event counter and coin credit buffer data in period short enough to prevent the loss of coin data[40]. Message format is:

Host sends: **[Dir] [00] [00] [E5] [Chk]**

Coin s. respond: **[01][0B] [Dir] [00] [Ev.cnt.][coin code 1][dir/err] [coin code 2][dir/err] . . .**

**. . . [coin code 5][dir/err] [Chk]**

Examples of message string for coin selector(*address 2*) after coin insertions:

Host sends: **[02] [00] [00] [E5] [18]** Polling minimum each 500 ms

Coin s. respond: **[01] [0B] [02] [00] [00][00][00][00][00][00][00][00][00][00][00] [F2]**

The respond after power-up or reset

Coin s. respond: **[01] [0B] [02] [00] [01][01][02][00][00][00][00][00][00][00][00] [EE]**

First event, coin possition 1, sorter path 2 accepted

Coin s. respond: **[01] [0B] [02] [00] [02][02][01][ 01][02][00][00][00][00][00][00] [EA]**

Second event, coin possition 2, sorter path 1 accepted

Coin s. respond: **[01] [0B] [02] [00] [03][00][02][02][01][01][02][00][00][00][00] [E7]**

Third event, coin rejected due to master inhibit active

Coin s. respond: **[01] [0B] [02] [00] [04][00][83][ 00][02][02][01][01][02][00][00] [63]**

Forth event, coin possition 4 inhibited and rejected

[39] Event counter

[40] See command 249 Request polling priority

From example we can notice shifting of data in the coin credit and error buffer and increment of event counter.

### 6.5.2.13 Command 228 [hexE4], Modify master inhibit status

This command is used to inhibit acceptance of all coins and has same effect as command modify inhibit status with sent with two bytes of zeros. Host sends only one byte of data. If first bit (*LSb*) is set to "0" coin selector is inhibited. Bits 1 to 7 has no influence to coin selector. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [01] [01] [E4] [Mask.] [Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK |

**Initially coin selectors are programmed with acceptance enabled.**
**Change is stored in RAM location .**
**On customer demand it is possible to set inhibition as default .**

Example of message string to inhibit the acceptance for coin selector(*address 2*) is:

| | | |
|---|---|---|
| Host sends: | **[02] [01] [01] [E4] [00] [18]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK |

After that coin selector acceptance will be inhibited till reset or next instruction that will change master inhibit status.

### 6.5.2.14 Command 227 [hexE3], Request master inhibit status

Coin selector respond with one byte data information of main inhibit status.
Only first (*LSb*) bit is used. If bit 0 is "1" acceptance is enabled, and if bit 0 is "0" coin selector is inhibited and acceptance is disabled.
Other bits has no meaning and always read as "0". Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [00] [E3] [Chk]** |
| Coin s. respond: | **[01] [01] [Dir] [00] [Mask.] [Chk]** |

Example of message string for coin selector(*address 2*) after power-up is:

| | | |
|---|---|---|
| Host sends: | **[02] [00] [01] [E3] [1A]** | |
| Coin s. respond: | **[01] [01] [02] [00] [01] [FB]** | Acceptance enabled (*default*) |

Example of message string for coin selector(*address 2*) after activation of master inhibit[41] is:

| | | |
|---|---|---|
| Host sends: | **[02] [00] [01] [E3] [1A]** | |
| Coin s. respond: | **[01] [01] [02] [00] [00] [FC]** | Coin selector inhibited |

[41] Using command 210, Modify master inhibit status

### 6.5.2.15 Command 226 [hexE2], Request insertion counter

Coin selector respond with three bytes of insertion counter data.
First byte is LS byte of three byte counter in RAM. Insertion counter is set to zero after power up or reset command. It is incremented each time a new coin is inserted in to coin acceptor. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [00] [E2] [Chk]** |
| Coin s. respond: | **[01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]** |

Example of message string for coin selector(*address 2*) after power-up is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [E2] [1B]** |
| Coin s. respond: | **[01] [03] [02] [00] [00] [00] [00] [FA]** |

### 6.5.2.16 Command 225 [hexE1], Request accept counter

Coin selector respond with three bytes of acceptance counter data.
First byte is LS byte of three byte counter in RAM. Acceptance counter is set to zero after power up or reset command. It is incremented each time a new coin pass acceptance sensor[42]. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [00] [E1] [Chk]** |
| Coin s. respond: | **[01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]** |

Example of message string for coin selector(*address 2*) after power-up is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [E1] [1C]** |
| Coin s. respond: | **[01] [03] [02] [00] [00] [00] [00] [FA]** |

### 6.5.2.17 Command 221 [hexDD], Request sorter override status

Coin selectors AL55/66 do not support override of sorter path.
Coin selector respond will be always: hex**[FF]** – Normal sorting.
Example of message string for coin selector(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [DD] [20]** |
| Coin s. respond: | **[01] [01] [02] [00] [FF] [FD]** |

### 6.5.2.18 Command 219 [hexDB], Enter new PIN number

Host send four byte data of new PIN number. If correct PIN was previously received[43] coin selector will accept the new PIN and respond with ACK message Coin selectors AL06x-c has PIN number stored in EEPROM. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [04] [01] [DB] [PIN1-LSB][PIN2][PIN3][PIN4-MSB] [Chk]** |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** ACK if PIN is correct |
| Coin s. respond: | no respond if PIN is incorrect or not recieved |

Example of message string for coin selector(*address 2*) with default PIN: hex**[00][00][00][00]** previously received and NEW pin hex**[01][02][03][04]** is:

[42] Credit sensor

[43] See next chapter

| | |
|---|---|
| Host sends: | **[02] [04] [01] [DB] [01][02][03][04] [14]** |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** ACK message |

### 6.5.2.19 Command 218 [hexDA], Enter PIN number

Host send four byte data of PIN number. If PIN is correct, coin selector will respond immediately with ACK message. If PIN is incorrect the NAK message will be sent with time delay of 100 ms. Coin selectors AL06x-c has PIN number stored in EEPROM. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [04] [01] [DA] [PIN1-LSB][PIN2][PIN3][PIN4-MSB] [Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK if PIN is correct |
| Coin s. respond: | **[01] [00] [Dir] [05] [Chk]** | dly 100 ms ->NAK if PIN is incorrect |

Example of message string for coin selector(*address 2*) with default PIN: hex**[00][00][00][00]** and wrong pin is:

| | | |
|---|---|---|
| Host sends: | **[02] [04] [01] [DA] [01][00][00][00] [1F]** | |
| Coin s. respond: | **[01] [00] [02] [05] [F8]** | dly 100 ms ->NAK if PIN is incorrect |

### 6.5.2.20 Command 216 [hexD8], Request data storage availability

Coin selector respond with five byte of data that describes type of memory and availability for host to read and to write. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [D8] [Chk]** |
| Coin s. respond: | **[01] [05] [Dir] [00] [d1][d2][d3][d4][d5] [Chk]** |

ALBERICI coin selectors AL55/66 currently do not support write or read to host accessible memory. Respond to command will be always as in example:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [D8] [25]** |
| Coin s. respond: | **[01] [05] [02] [00] [00][00][00][00][00] [F8]** |

### 6.5.2.21 Command 213 [hexD5], Request option flags

Coin selector respond with one byte of data that describes type of coin format. For CVF (***C****oin* ***V****alue* ***F****ormat*) bit 0 is set to 1, and for coin position format value is "0". Other bits are not used and read always as "0". Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [D5] [Chk]** |
| Coin s. respond: | **[01] [01] [Dir] [00] [d1] [Chk]** |

Example of message string for coin selector(*address 2*) is:

| | | |
|---|---|---|
| Host sends: | **[02] [00] [01] [D5] [28]** | |
| Coin s. respond: | **[01] [01] [02] [00] [00] [FC]** | Coin position format |

### 6.5.2.22 Command 210 [hexD2], Modify sorter paths

With this command host is able to change direction of coins in sorter if sorter is supported. Host sends two bytes of data to select the coin position and sorter path (*direction of exit*). Firs byte of data (*LSB*) represent coin position and second byte of data point to sorter path. ALBERICI coin selectors has support for most existing sorters that has direct drive of coils from coin selector with open collector transistor. Most common are 3 or 4 way sorter with two coils[44] , but recently 5 way sorters[45] with 3 coils are in use. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [02] [01] [D2] [Coin pos.] [Sort.Path] [Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK if sorter path is possible to set |
| Coin s. respond: | **[01] [00] [Dir] [05] [Chk]** | NAK if coin selector does not support setting |

**Initially all coin position has sorter paths set to direction 1 hex[01].**
**If sorter is not supported, sorter path is set initially to 0 hex[00]!**

If host sends command to modify sorter path that is not existent or for coin not programmed, the coin selector will respond with message NAK. Example of message string for coin selector(*address 2*) redirection of coin **pos. 1** in to **path 2** is:

| | | |
|---|---|---|
| Host sends: | **[02] [02] [01] [D2] [01] [02] [26]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK |

After acceptance of command, accepted coins with position 1 will exit in direction 2 of the sorter. The path or direction 1 is usually one without activation of any coil. Different coil activation schematics is possible to program by setting the sorter type.

### 6.5.2.23 Command 209 [hexD1], Request sorter paths

Host send one byte of coin position and coin selector respond with one byte of sorter path. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [01] [00] [D1] [Coin pos.] [Chk]** |
| Coin s. respond: | **[01] [01] [Dir] [00] [Sort.Path] [Chk]** |

Example of message string for coin selector(*address 2*) for initial sorter path 1 of coin position 1:

| | |
|---|---|
| Host sends: | **[02] [01] [01] [D1] [01] [2A]** |
| Coin s. respond: | **[01] [01] [02] [00] [01] [FB]** |

Example of message string for coin selector(*address 2*) for sorter path 2 of coin position 1:

| | |
|---|---|
| Host sends: | **[02] [01] [01] [D1] [01] [2A]** |
| Coin s. respond: | **[01] [01] [02] [00] [02] [FA]** |

[44] Maximum current consumption for each coil is 500 mA

[45] 5-way VARIANT sorter from ALBERICI

**If host request sorter path for non programmed coins or non existent position[46] , the coin selector will respond with message NAK !**

### 6.5.2.24 Command 202 [hexCA], Teach mode control

This command is used to start teach process(program coin recognition data).
The respond of coin selector is ACK if teach mode is supported.
With command header host must send number of channel to program.
ALBERICI coin selectors AL55/66 has possibility to program 16 different coins.
Some coin selectors that must not be reprogrammed for security or any other reason do not support this instruction[47]. If teach mode instruction is not supported coin selector will not respond to this instruction. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [01] [01] [C9] [ch][Chk]** | |
| Coin s. respond: | **[01] [00] [Dir] [00] [Chk]** | ACK if ch is betwen 1-16 |
| Coin s. respond: | **[01] [00] [Dir] [05] [Chk]** | NAK if ch is 0 or bigger than 16 |

Example of message string for coin selector(*address 2*) to teach (*program*) coin on position (*channel*) 1:

| | | |
|---|---|---|
| Host sends: | **[02] [01] [01] [C9] [01] [31]** | |
| Coin s. respond: | **[00] [02] [00] [FD]** | ACK |

### 6.5.2.25 Command 201 [hexC9], Request teach status

This command is used during teach process, after instruction 201 Teach mode control.
The respond of coin selector is according to teach state.
There is two different format for this instruction.
Format "a" is with data hex**[00]**, after which coin selector respond with number of inserted coins and state of teach process.
Second format is "b" with data hex**[01]**, after which coin selector abort the teach process and respond with code dec**[252]**, teach aborted and number of inserted coins.
Teach status codes are:

- 252 Teach aborted
- 253 Teach error
- 254 Teaching in progress(busy)
- 255 Teach completed

Message format (a) is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [01] [01] [C9] [00][Chk]** | Request status |

Message format (b) is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [01] [01] [C9] [01][Chk]** | Abort teach process |
| Coin s. respond: | **[01] [02] [Dir] [00] [coin nr.][status] [Chk]** | |

[46] Position bigger than 16

[47] Italian cctalk coin selectors AL06V-c for gambling machines

### 6.5.2.26 Command 196 [hexC4], Request creation date

Coin selector respond with two byte of data that represent codified date of production. Date of production is codified in so called *RTBY (**R**elative **T**o **B**ase **Y**ear )*[48] format. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [C4] [Chk]** |
| Coin s. respond: | **[01] [02] [Dir] [00] [LSB] [MSB] [Chk]** |

Example of message string for coin selector(*address 2*) with date of production **05.07.2003.** is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [C4] [39]** |
| Coin s. respond: | **[01] [02] [02] [00] [E5] [06] [10]** |

ALBERICI coin selectors has date of production written in monitor part of MCU FLASH memory which is not possible to change without factory FLASH reprogramming.

### 6.5.2.27 Command 195 [hexC3], Request last modification date

Coin selector respond with two byte of data that represent codified date of last modification of software[49]. Date of modification is codified also in RTBY format. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [C3] [Chk]** |
| Coin s. respond: | **[01] [02] [Dir] [00] [LSB] [MSB] [Chk]** |

Example of message string for coin selector(*address 2*) with date of modification **23.07.2003.** is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [C3] [3A]** |
| Coin s. respond: | **[01] [02] [02] [00] [F7] [06] [FE]** |

After each up-grade of coin selector program FLASH memory date will correspond to software modification date, not the actual date of up-grade!

### 6.5.2.28 Command 194 [hexC2], Request reject counter

Coin selector respond with three bytes of reject counter data.
First byte is LS byte of three byte counter in RAM. Reject counter is set to zero after power up or reset command. It is incremented each time a coin is inserted but not recognized. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [00] [C2] [Chk]** |
| Coin s. respond: | **[01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]** |

Example of message string for coin selector(*address 2*) after power-up is:

[48] For details see cctalk protocol, document cctalk44-2.pdf

[49] Up-grade of FLASH program memory

| | |
|---|---|
| Host sends: | **[02] [00] [01] [C2] [3B]** |
| Coin s. respond: | **[01] [03] [02] [00] [00] [00] [00] [FA]** |

### 6.5.2.29 Command 193 [hexC1], Request fraud counter

Coin selector respond with three bytes of fraud coins counter data.
First byte is LS byte of three byte counter in RAM. Fraud counter is set to zero after power up or reset command. It is incremented each time a coin acceptor recognize coin that is programmed as "fraud" coin[50]. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [00] [C1] [Chk]** |
| Coin s. respond: | **[01] [03] [Dir] [00] [Cunt1-LSB] [Cunt2] [Cunt3-MSB] [Chk]** |

Example of message string for coin selector(*address 2*) after power-up is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [C1] [3C]** |
| Coin s. respond: | **[01] [03] [02] [00] [00] [00] [00] [FA]** |

### 6.5.2.30 Command 188 [hexBC], Request default sorter path

For ALBERICI coin selectors AL55/66 the default sorter path is always hex**[01]**.
Example of message string for coin selector(*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [00] [01] [BC] [41]** |
| Coin s. respond: | **[01] [01] [02] [00] [01] [FB]** |

### 6.5.2.31 Command 185 [hexB9], Modify coin ID

With this command it is possible to change coin ID message string that will be used in communication with host. String has 6 ASCII characters:

- Country code(*2 bytes*)
- Value(*3 bytes*)
- Mint issue(*1 byte*)

Each country has a 2 letter designator code described in ISO 3166-1-A2[51].
ALBERICI coin selectors has limited possibility to change the country code.
Table of four country codes must be programmed first(*by customer*) or must be empty.
Host must sent code that correspond to one from the table or if there is one empty place it could be use for new code. New value will be automatically programmed in next free space for country code. If there is no free space and code does not match with any code from table, the NAK message will be returned.
Default country code table programming for European countries using euro coins is:

- EU
- TK
- ..
- ..

"EU" is code for euro coins, "TK" is for token and ".." is code for non programmed coin

[50] Coins with close recognition parameters sometime called "killer coin or channel"
[51] See document cctalk42-3 Appendix 3

positions. Country code is possible to change using our programming software or directly by writing code in memory location from address hex 07B2 (*4x2 bytes*)[52]. Coin value code must be one selected from table 6.5 or NAK message will be returned to host.

| 3 x ASCII Characters | Value |
|---|---|
| 001 | 1 |
| 002 | 2 |
| 2.5 | 2.5 |
| 005 | 5 |
| 010 | 10 |
| 020 | 20 |
| 025 | 25 |
| 050 | 50 |
| 100 | 100 |
| 200 | 200 |
| 250 | 250 |
| 500 | 500 |
| ... | Not prog. |

Table 6.5 Coin value codes

Mint issue must also be sent as "A", "B", "C" or ".." for non programmed coins!

Message format is:

Host sends: **[Dir] [07] [01] [B8] [Coin pos][a1][a2][a3][a4][a5][a6] [Chk]**

Coin s. respond: **[01] [00] [Dir] [00] [Chk]** ACK if codes are acceptable

Coin s. respond: **[01] [00] [Dir] [05] [Chk]** NAK if not

### 6.5.2.32 Command 184 [hexB8], Request coin ID

Host use this command at initialization process to build table for each coin position value. If coin selector uses CVF it is obsolete command.
Host send one byte data of coin position and coin selector respond with 6 byte ASCII string of characters that describes the requested coin position.
Message format is:

Host sends: **[Dir] [01] [01] [B8] [Coin pos] [Chk]**

Coin s. respond: **[01] [06] [Dir] [00] [a1][a2][a3][a4][a5][a6] [Chk]**

Example of message string for coin selector(*address 2*) and coin position 1(*2 Euro*) is:

Host sends: **[02] [01] [01] [B8] [01] [43]**

Coin s. respond: **[01] [06] [02] [00] [45][55][32][30][30][41] [8A]** Coin 'EU200A'

[52] See chapter 6.5.4.1.2

For none-programmed position the ASCII string is: '**......**'.
Example of message string for coin selector(*address 2*) and coin position 12 that is not programmed is:

Host sends: **[02] [01] [01] [B8] [0C] [38]**

Coin s. respond: **[01] [06] [02] [00] [2E][ 2E][ 2E][ 2E][ 2E][ 2E] [E3]** Coin not programed

### 6.5.2.33 Command 176 [hexB0], Request alarm counter

Coin selector respond with one bytes of alarm counter data.
Alarm counter is set to zero after power up or reset command. It is incremented each time a coin acceptor detect any type of erroneous coin acceptance[53].
Message format is:

Host sends: **[Dir] [00] [00] [B0] [Chk]**

Coin s. respond: **[01] [01] [Dir] [00] [Cunt] [Chk]**

Example of message string for coin selector(*address 2*) after power-up is:

Host sends: **[02] [00] [01] [B0] [4D]**

Coin s. respond: **[01] [03] [02] [00] [00] [FC]**

### 6.5.2.34 Command 173 [hexAD], Request thermistor reading

Some coin selectors AL66[54] has built in linear temperature sensor.
Using this command is possible to read temperature on surface of coin selector PCB. If temperature sensor is not built in coin selector will not respond to this command. Temperature sensor is linear type, with 1 unit change for one degree Celsius change. For 0°C value will be dec**[50]**, for ie. 25°C it will be dec**[75]**, for -10°C it will be dec**[40]** and for 50°C it will be dec**[100]**. Message format is:

Host sends: **[Dir] [00] [00] [AD] [Chk]**

Coin s. respond: **[01] [01] [Dir] [00] [Temp] [Chk]**

Example of message string for coin selector(*address 2*) at ambient temperature of 25°C is:

Host sends: **[02] [00] [01] [AD] [50]**

Coin s. respond: **[01] [01] [02] [00] [4B] [B1]**

### 6.5.2.35 Command 170 [hexAA], Request base year

Coin selector respond with four byte ASCII string of character representing the base year for calculation of exact date of production. Message format is:

Host sends: **[Dir] [00] [01] [AA] [Chk]**

Coin s. respond: **[01] [04] [Dir] [00] [a1][a2][a3][a4] [Chk]**

For ALBERICI coin selectors base year is **2000**.

[53] Alarms: Coin direction error(Jojo), coin to slow(Coin jam ) or coin to fast

[54] Coin selectors for extreme ambient temperature (external use)

Example of message string for coin selector(*address 2*) is:

Host sends: **[02] [00] [01] [AA] [53]**

Coin s. respond: **[01] [04] [02] [00] [32][30][30][30] [37]**

### 6.5.3 MDCES command headers

MDCES stands for **M**ulti-**D**rop **C**ommand **E**xtension **S**et, or so called Multi-drop buss commands. Multi-drop buss commands gives additional functionality to systems that require change of address for devices in cctalk network.
Some of commands has different message format than usual cctalk message.
Commands are:

- Address poll
- Address clash
- Address change
- Address random

Because host always use address 1 and address 0 is for broadcast message all commands that changes the address should not accept this settings.

**All changes are stored in non-volatile memory, EEPROM !**

#### 6.5.3.1 Command 253 [hexFD], Address poll

This is a broadcast message used by host to determinate all address of device attached on cctalk network. Coin selector respond with only one byte (*non-standard message format*), after a delay that is proportional to address value multiplied with 4 milliseconds. Message format is:

| | | |
|---|---|---|
| Host sends: | **[00] [00] [01] [FD] [Chk]** | Brodcast mesage |
| Coin s. respond: | **Dly -> [Address]** | |

Example of message string for coin selector(*address 2*) is:

| | | |
|---|---|---|
| Host sends: | **[00] [00] [01] [FD] [02]** | |
| Coin s. respond: | **Dly=8 ms -> [02]** | Address is 2 |

Example of message string for coin selector (*address 250*) is:

| | | |
|---|---|---|
| Host sends: | **[00] [00] [01] [FD] [02]** | |
| Coin s. respond: | **Dly=1 s -> [FA]** | Address is 250 |

#### 6.5.3.2 Command 252 [hexFC], Address clash

Command Address clash has same respond from coin selector but host issue this command with specific device address. Coin selector respond with only one byte (*non-standard message format*), after a random value of time delay to prevent collision if two devices share same address. Message format is:

| | |
|---|---|
| Host sends: | **[Dir] [00] [01] [FC] [Chk]** |
| Coin s. respond: | **Random Dly -> [Address]** |

Example of message string for coin selector(*address 2*) **AL06V-c** is:

| | | | |
|---|---|---|---|
| Host sends: | **[02] [00] [01] [FC] [01]** | | |
| Coin s. respond: | **Random Dly -> [02]** | | Address is 2 |

### 6.5.3.3 Command 251 [hexFB], Address change

Command Address change is issued to a specified device only. Coin selector respond with ACK message. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [01] [01] [FB] [Address] [Chk]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK |

Example of message string for coin selector(*address 2*) change to address to 20:

| | | | |
|---|---|---|---|
| Host sends: | **[02] [01] [01] [FB] [14] [ED]** | | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK | Address is now 20 |

Coin selector does not respond to attempt of change an address to 0 or 1.

### 6.5.3.4 Command 250 [hexFA], Address random

Command Address random has the same respond from coin selector. New address is not sent because each device set its own random address.
Host software sometime can issue this command as broadcast. This will cause change of all device addresses. Coin selector respond with ACK message. Message format is:

| | | |
|---|---|---|
| Host sends: | **[Dir] [00] [01] [FA] [Chk]** | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK |

Example of message string for coin selector(*address 2*) is:

| | | | |
|---|---|---|---|
| Host sends: | **[02] [00] [01] [FA] [03]** | | |
| Coin s. respond: | **[01] [00] [02] [00] [FD]** | ACK | Address is changed |

Example of broadcast message string for coin selector is:

| | | | |
|---|---|---|---|
| Host sends: | **[00] [00] [01] [FA] [05]** | Brodcast mesage | |
| Coin s. respond: | **[01] [00] [00] [00] [FD]** | ACK | Address is changed |

**Coin selector has internal mechanism that prevent setting of address 0 or 1!**

### 6.5.4 ALBERICI specific command

First command header for specific factory setting and testing is **255 Factory set-up and test**. This command has several modes of use and some of them are factory secret and are not explained in this document.
Beside this next chapter describe two modified cctalk commands[55] for encrypted exchange of data between host and coin selector.

#### 6.5.4.1 Command 255, Factory set-up and test

This instruction header is used only for factory testing and programing!
With instruction command header one data byte must be sent for definition of command mode. Modes are:

| | |
|---|---|
| - Mode 0(*hex 00*) | No description – reserved for factory use only |
| - Mode 1(*hex 01*) | No description – reserved for factory use only |
| - Mode 2(*hex 02*) | Read coin selector memory data |
| - Mode 3(*hex 03*) | Write coin selector memory data |
| - Mode 4(*hex 04*) | Analog circuit test |
| - Mode 5(*hex 05*) | Coin parameter test |
| - Mode 6(*hex 06*) | Up-grade FLASH(*program memory*) |
| - Mode 7(*hex 07*) | Re-initialization |

**First two modes are only for internal factory use and they are not described in this document!**

To use read and write instruction user must have basic knowledge about coin selectors memory data organization[56]!

##### 6.5.4.1.1 Command 255 mode 2, Read coin sel. memory

This instruction sends back to host block of memory data that was requested.
Data memory of coin selector is divided into a 6 group:

| | |
|---|---|
| - Coin channel data | address hex 0600 do 06FF(*256 byte-a*) |
| - Input/Output data | address hex 0700 do 073F(*64 byte-a*) |
| - Factory common data | address hex 0740 do 079F(*96 byte-a*) |
| - User general data | address hex 07A0 do 07E9(*74 byte-a*) |
| - Statistic setting | address hex 07EA do 07FD(*20 byte-a*) |
| - Statistic counters | address hex 0800 do 083F(*64 byte-a*) |

Factory key or user PIN protection are some time set to disable the access to some blocks of memory. Usually all memory is accessible for red instruction.
On any attempt to read memory location that is protected without clearing key or PIN protection mechanism, coin selector will respond with **NAK** message.
If number of blocks to read exceed block range, coin selector will also respond with **NAK** message. Maximum memory block read of 137 data bytes are limited by coin selector transmit buffer that has 142 bytes[57]. If block size extend this number coin selector will respond with NAK message to. Message string format is:

[55] Commands 214 Write data block and 215 Read data block
[56] Details are available on customer request, see document AL55-66MemDataOrg-v..pdf
[57] Destination address+Bytes nr.+Source address+Header+59 data+Checksum=142

Host sends: **[Dir] [04] [01] [FF] [02][Start add-hi][ Start add-lo][ data nr.] [Chk]**

Coin s. respond: **[01] [n] [Dir] [00] [data 1] [data 2] . . . [data n] [Chk]**

Start add-hi is start address hi byte, while start add-lo is address low byte. Data nr. represent the number of data to read (block size).
Data 1 to data n are requested coin selector memory data.
The example of message string for reading of first coin channel data is:

Host sends: **[02] [04] [01] [FF][02] [06][ 00] [10] [E2]**

Coin s. respond: **[01][10][02][00][89][87][B4][77][A7][9F][08][08][0C][16][0A][04][08][01][09][C8][52]**

### 6.5.4.1.2 Command 255 mode 3, Write coin sel. memory

This command sends block of data to be written in the coin selector RAM memory. As for the previous command(*read*) access to some memory blocks could be protected by factory key or user PIM mechanisam. Factory common data are usually protected with factory key and statistic counters and setting with user PIN. Exeption is made for statistic countet write. Any attempt to write in to statistic counter will erase those couners, thus protecting user from manipulation with statistics!
Number of bytes sent in command string are representing number of statistic to be erased. Message string format is:

Host sends: **[Dir] [n+3] [01] [FF] [03][Start add-hi][ Start add-lo][ data 1]. . . [ data n] [Chk]**

Coin s. respond: **[01][00] [Dir] [00] [Chk]** ACK

The example of message string for programming first output of coin selector **AL55/66** (*address 2*) as pulse parallel output for coin(*channel*) position 1 is:

Host sends: **[02] [0B] [01] [FF][03] [07][ 00] [01][00][00][00][14][00] [D2]**

Coin s. respond: **[01][00][02][00] [FD]** ACK

The example of message string for erase of all 20 statistic counters[58] for coin selector **AL55/66**(*address 2*) is:

Host sends: **[02] [04] [01] [FF][02] [08][ 00] [14] [DB]**

Coin s. respond: **[01][00][02][00] [FD]** ACK

User PIN must be sent before erase(*write*) of statistic counters otherwise coin selector will respond with **NAK** message.

**Data write will change data in coin selector RAM, and changes will be lost if power supply turns off! Use write to FLASH command to save changes[59]!**

To save all coin selector RAM data in to FLASH, send write command with no data:

Host sends: **[02] [01] [01] [FF][03] [FA]**

Coin s. respond: **[01][00][02][00] [FD]** dly 20 -100 ms ->ACK

Coin selector will reply with ACK if write to FLASH was successful.

[58] 20 x 3 byte of memory!

[59] Special case write command

### 6.5.4.1.3 Command 255 mode 4, Analog circuit test

This command returns to host string of data that can be used to test analog measuring circuit state. Details are described in internal documents and not available to users.

### 6.5.4.1.4 Command 255 mode 5, Coin parameter test

This command returns to host string of data from the last coin measurement.
This data are coin measured parameters and could be used to create and analyze coin data base. Message string format is:

| | |
|---|---|
| Host sends: | **[Dir] [01] [01] [FF][05] [Chk]** |
| Coin s. respond: | **[01] [6] [Dir] [00] [data 1] [data 2] . . . [data 6] [Chk]** |

Data string description:

| | |
|---|---|
| - data 1 | Measured coin parameter 1, AM1 |
| - data 2 | Measured coin parameter 2, PH1 |
| - data 3 | Measured coin parameter 3, AM2 |
| - data 4 | Measured coin parameter 4, PH2 |
| - data 5 | Measured coin parameter 5, PI3 |
| - data 6 | Measured coin parameter 6, DIM |

The example of message string after insertion of one Euro coin for coin selector **AL66** (*address 2*) is:

| | |
|---|---|
| Host sends: | **[02] [01] [01] [FF][05] [F8]** |
| Coin s. respond: | **[01] [6] [02] [00] [8A] [80][8C] [5C] [AB] [8A] [D0]** |

Coin measured parameters(*data*) are available for read till next coin insertion.
This instruction is used by our coin programming software.

### 6.5.4.1.5 Command 255 mode 6, Up-grade FLASH

This command is used to up-grade coin selector FLASH program memory.
User can up-grade coin selector in cases when he knead some new function or improvement, which where not available at the moment when coin selector was purchased. This instruction is used by our coin selector programming software and generally is not useful to most users. Additional information are available on request.

**Up-grade files are encrypted in factory and decryption is done internally by monitor program of coin selector.**
**Coin selector will accept only correct type of up-grade file!**

### 6.5.4.1.6 Command 255 mode 7, Factory reset

This command will set all coin selector data to initial factory programming values!

**Warning: This could lead to unwanted coin selector function if different coin , input/output or user setting data where programmed!**

The example of message string reset the factory settings is:

| | | |
|---|---|---|
| Host sends: | **[02] [01] [01] [FF][07] [F6]** | |
| Coin s. respond: | **[01][00][02][00] [FD]** | ACK |

After that coin selector must be switched off/on!

## 7. Diagnostic

Not available at the moment

## 8. Service

Not available at the moment

# 9. Technical data

## 9.1 Mechanical data

| | |
|---|---|
| Format | 3½” standard housing |
| Dimensions | 88 x 102 x 52 mm |
| Weight | 220 g for AL66 |
| | 200 g for AL55 |

## 9.2 Electrical data

| | |
|---|---|
| Power supply min. voltage | 8 V DC |
| Power supply max. voltage | 28 V DC |
| Power consumption | |
| Acceptance | 350 mA(*30 ms*)/100 mA[60] |
| Measurement | ≤30 mA for AL66 |
| | ≤25 mA for AL55 |
| Stand by | ≤25 mA for AL66 |
| | ≤20 mA for AL55 |
| Power save stand. | ≤5 mA for AL66 |
| | ≤2,5 mA for AL55 |
| Pow. save self wake-up | ≤6 mA only AL66 |
| | ≤3,5 mA for AL55 |
| Output type | Open collector Darlington |
| Output saturation voltage | ≤1 V |
| Output max. voltage | 50 V |
| Output max. current | 250 mA |
| Input min. activation level | 3 V |
| Max. input voltage | 50 V |
| Input impedance | ≈55 kΩ |

## 9.3 Coin acceptance

| | |
|---|---|
| Number of coin channels | 16 |
| Minimum coin diameter | 16 mm |
| Maximum coin diameter | 32 mm |
| Coin thickness | 1 to 3,4 mm |

## 9.4 Timing data

| | |
|---|---|
| Power-up recovery time | ≤200 ms |
| Wake-up recovery time | ≤50 ms |
| Pulse or time out tolerance | ± 2% |

[60] Coil “On” time is 30 ms and “Hold” time depend on coin speed 100 to 200 ms

## 9.5 Ambient conditions

| | |
|---|---|
| Working ambient temperature | 0°C to 60°C |
| Storage temperature | -30°C to 70°C |
| Humidity | up to 75%(*non condensing*) for standard<br>up to 95%  for tropicalization[61] |

## 9.6 EMC performance

This product is compliant with EN55014-1 and  EN55014-2 test specification[62]

[61] Polyurethan vernice URETHAN 71 aplyed to sensors and PCB on both sides!

[62] Not tested!