AZKOYEN
MEDIOS DE PAGO

ccTalk Protocol
HOPPER U-II

GRUPO AZKOYEN

AZKOYEN Medios de Pago, S.A
Avda. San Silvestre, s/n
31350 Peralta (Navarra) Spain

Tel.: +34 948 709 709
Fax: +34 948 709 709
www.azkoyen.com

AZKOYEN
MEDIOS DE PAGO

Index of contents

Index of Tables

1. GENERAL DESCRIPTION OF THE PROTOCOL

The communication protocol implemented in the Hopper U-II in the "U-II" range is compatible with ccTalk®.

The specification used to implement this communication protocol is as follows:

ccTalk Serial Communication protocol
Generic Specification Issue 4.3

ccTalk® is a serial communication protocol, developed by Coin Controls Ltd. for low transmission velocity control networks.

It has been designed to allow the interconnection of different coin mechanisms (Hopper U-II, validators, etc.) on a simple bus of two lines (one line for bidirectional data and the other ground).

The communication is carried out using one bidirectional line (DATA).

The use of ccTalk® is open and, therefore, can be used without paying licences or rights

1.1. COMMUNICATION TOPOLOGY

The data is established with TTL levels, 5 volts is '0' logic and 0 volts is '1' logic which is the rest level of 5 volts.

1.2. COMMUNICATION CHARACTERISTICS

The communication is asynchronous and half-duplex, that is, more than one element cannot be transmitted on the bus at the same time.

The 'timing' of the communication satisfies the characteristics of the industrial standard RS232.

The RS232 communication has several parameters that are configured as follows in this application, such as:

9600 bauds, 1 start bit, 8 data bits, no parity, and 1 stop bit.

The message format is as follows:

[Destination address]
[No. Data bytes]
[Source address]
[Header]
[Data 1]
[Data 2]
…
[Data N]
[Checksum]

Each communication sequence is comprised of two strings.
The first corresponds to the command sent by the Machine to the Hopper U-II and the second is the reply sent by the Hopper U-II to the Machine. Both parts have the format indicated above.

1.2.1. Destination address

The range of addresses goes from 0 to 255 (of which 254 correspond to Hopper U-II addresses as explained below).

0: Used in messages which affect all the Hopper U-II simultaneously, although in the present version there is no command implemented that uses it.

1: Address of the Machine

2: Address of the Hopper U-II in communication with one single slave.

3 to 255: Address of the Hopper U-II in multi-slave communication. The 'Destination address' byte indicates the bus node (slave) the message is addressed to. In our case the values that are used for the Destination address are:

1: When the message is sent from a Hopper U-II to the Machine.

3, 4, 5 or 6: When the message is sent to the Hopper U-II configured by means of the dipswitch as 1, 2, 3 or 4 respectively.

3 to 255: When the message is sent to the Hopper U-II configured by means of software with MDCES commands, and with its address saved in the EEPROM.

1.2.2. Number of data bytes

The range of the number of data to be transmitted goes from 0 to 252.

This byte indicates the number of data bytes of the message and not the total number of bytes of the message. If it is ’0’ this means that the message has no data and in this case the total number of bytes of the message will be 5 bytes (minimum permitted).

The values 253 to 255 are not permitted in this field and would be considered as value 252.

1.2.3. Source address

The range of addresses goes from 1 to 255 (of which 254 correspond to Hopper U-II addresses)

1: Machine address.

2: Address of the Hopper U-II in communications with one single Hopper U-II.

3 to 255: Addresses of the Hopper U-II in multi-Hopper U-II communication, which is the case that concerns us.

The ’Source Address’ byte indicates the bus node that sends the message. In our case the values that are used for the ’source address’ are:

1: When the message is sent by the machine

3, 4, 5 or 6: When the message is sent by the Hopper U-II configured with the dip switch such as 1, 2, 3 or 4, respectively.

3 to 255: When the message is sent by the Hopper U-II configured by means of software with the MDCES commands, and with its address saved in the EEPROM.

1.2.4. Header

The header byte range goes from 0 to 255. The Header will never have ’0’ value in the case of messages sent by the Machine.

In the case of reply messages, which are those sent by the Hopper U-II, the header will have ’0’ value in all the messages, except in the ’Negative Acknowledgement’ or NACK message.

1.2.5. Data

The range of values that each one of the data bytes can take is 0 to 255. It has no restrictions of use, any data format being possible such as binary, ASCII, etc.

1.2.6. Checksum

Checksum is what makes the 8 less important bits of the sum of all the bytes of the message, including the actual checksum, give '0' as the result.

For example:

The message [01][00][02][00] will be followed by checksum [253] because 1 + 0 + 2 + 0 + 253 = 256 = 0

1.3. TIME REQUIREMENTS

1.3.1. Maximum time between bytes

The maximum time between two bytes of the same message is 50 ms. If this time is exceeded the communication program will reset the communication variables and will prepare to receive a new message.

1.3.2. Maximum time between command and reply

The maximum time to reply to a command depends on the time that it takes the Hopper U-II to process that command.

The default time is 50 ms, if no other value is specified in the command definition.

1.3.3. Minimum time from power up until first command is sent

The minimum time that must elapse from the time the Hopper U-II is powered up until the first command is sent must be 250 ms.

1.4. ERROR MANAGEMENT

1.4.1 Action to take in case of error

If a Hopper U-II receives an incomplete message (receipt timeout) or with an incorrect checksum, the only action is carries out is to reset the communication variables and prepare to receive a new message.

The Machine, on not receiving a reply to the message sent, can choose to resend the same message.

It is recommended to do various retries if at moment the a reply to a command from the machine is not received the Hopper U-II is paying out

The recommended wait for sending the retry will be between 75 and 100 milliseconds.

On the other hand, if the Machine receives an incomplete message (receipt timeout) or with an incorrect checksum, it can choose to resend the same message. In any case, faced with an error in receipt of a message no negative acknowledgement message has been defined, which simplifies the implementation of multi-Hopper U-II protocols and reduces collisions.

If a Hopper U-II receives a command that it is not able to execute, it responds with a negative acknowledgement message.

This occurs, for example, when during the execution of a payment command another payment or emptying command is received.

1.4.2. Error communication

The Hopper U-II carries out a series of controls of its peripherals and is able to detect a series of anomalies, which are in turn notified to the Machine.

The Machine is notified of the error(s) by means of the "Request for Status" command requested by the latter from the Hopper U-II. On this Request the Hopper U-II answers with a byte indicating its error situation and with another byte that indicates the error produced. Each one of the bits of this last byte represents a possible error, so that the bits that are on 1 will indicate that the corresponding error has been detected and the ones that are on 0 will indicate that they have not been detected.

The machine can also be notified of the error(s) produced through the specific ccTalk® command of 'Test Hopper U-II' requested by the latter from the Hopper U-II. On this request the Hopper U-II answers with one byte or status register that indicates the latter's error situation. Each one of the bits of this byte represents a possible error or operation status, so that the bits that are on 1 will indicate that the corresponding error has been detected and those that are on 0 will indicate that they have not been detected.

The errors are cumulative, that is, if more than one error is being detected at the same time, the Machine will be notified of all of them.

The Hopper U-II quits the Error status when it receives a 'Pay' or 'Empty' or 'Reset' command, going on to execute this command. If the fault persists, the Hopper U-II will enter the Error status again.

2. CCTALK PROTOCOL IN THE HOPPER U-II

2.1. CONNECTOR ccTalk.

The Hopper U-II uses a type 5 connector indicated by the ccTalk general specification.
It is a 10-pin connector (2x5) of 2.5mm, Molex 8624 or similar.
The pin out of the connector is as follows:

Pin	Function
1	Data
7,10	Vdc
8,4	GND

Table 1. Pin out ccTalk connector.

2.2. LIST OF COMMANDS IMPLEMENTED IN THE HOPPER U-II

COMMAND HEX	DESCRIPTION
FE	Command 254: Simple poll
FD	Command 253: Address poll
FC	Command 252: Address clash.
FB	Command 251: Address change..
FA	Command 250: Address random.
F6	Command 246: Request manufacturer id
F5	Command 245: Request equipment category
F4	Command 244: Request product code
F2	Command 242: Request serial number
F1	Command 241: Request software version
EC	Command 236: Read opto states
DB	Command 219: Enter New PIN number
DA	Command 218: Enter PIN number.
C0	Command 192: Request build code
AC	Command 172: Emergency stop
A8	Command 168: Request Hopper U-II dispense count
A7	Command 167: Dispense Hopper U-II coins
A6	Command 166: Request Hopper U-II status
A4	Command 164: Enable Hopper U-II
A3	Command 163: Test Hopper U-II
A0 (2)	Command 160: Request cipher Key
8D (1)	Command 141: Request firmware upgrade capability
8C (1)(3)	Command 140: Upload
8B (1)(3)	Command 139: Begin firmware upgrade
8A (1)(3)	Command 138: Finish firmware upgrade
04	Command 04: Request comms revision
01	Command 01: Reset device

Table 2. CcTalk commands implemented.

(1) New command, not available in previous versions of the Hopper U-II U.
(2) Command only available in the Hopper U-II Encrypted.
(3) Command not available in the version of the Hopper U-II for Italy.

2.3. DESCRIPTION OF THE CCTALK COMMANDS

Command 254: Simple poll

Command to check the correct operation of the communication and to confirm the presence in the bus of a specific Hopper U-II.

Data sent: --

Data received: **ACK**

Command 253: Address poll

This command requests all slave devices for their address. The command is sent with the destination address of 0 (broadcast). To avoid collisions, the slave devices do not reply with a standard ccTalk message, but reply with the byte of the address with a delay that is proportional to the value of the address.

Message sent: **[00] [00] [01] [FDH] [02]**

Message received: **{Delay variable}[Dir]**

Where: Dir = address of the corresponding Hopper U-II.

The algorithm used to calculate the delay of the reply is the following:

Deactivate rx port

Delay (4*Dir) milliseconds

Send [Dir]

Delay 1200 (4*Dir) milliseconds

Activate rx port

Command 252: Address clash.

This command is used to check if one or more Hopper U-II shares the same address. Unlike the Address Poll command, this is sent to a specific address.
To avoid collisions, the slave devices do not reply with a standard ccTalk message, but reply with the byte of the address with a byte is only returned with a random delay in the sending.

Message sent: **[Dir] [00] [01] [FCH] [Chk]**

Message received: **{Delay variable}[Dir]**

Where: Dir = address of the corresponding Hopper U-II

The algorithm to calculate the delay with which it has to reply is as follows:

r=rand(256)

Deactivate rx port

Delay (4*Dir) milliseconds

Send [Dir]

Delay 1200 (4*Dir) milliseconds

Activate rx port

Command 251: Address change.

This command permits reprogramming the Hopper U-II with a new address, valid for all the commands it receives from now on.

If the new address is number 0, the Hopper U-II will obtain its new address (only addresses 3 to 6) from its corresponding dip switch, however, the Machine will not know which this new address is, therefore, once this command has been used with this option, the Machine should send the **[Address Poll]** command to find out the new address of the Hopper U-II U-II.

Data sent: **[Dir]**

Data received: **ACK**

Where: Dir = new address of the corresponding Hopper U-II

The new address is maintained after switching off or a reset.

When SW6 is ON, the Hopper U-II will not carry out any action for this command (it will not even respond).

Command 250: Address random.

This command permits reprogramming the Hopper U-II with a new address whose value will be random and generated by itself.

This is an escape channel for cases when several devices share the same address.

After this command the machine must send the Address Poll command to know the values of the new addresses.

Data sent: ---

Data received: **ACK**

The new address is maintained after switching off or a reset.

When SW6 is ON, the Hopper U-II will not carry out any action for this command (it will not even respond).

Command 246: Request manufacturer id

With this command, the corresponding Hopper U-II sends the Machine the identification of the device manufacturer.

Data sent: ---

Data received: **"Azkoyen"**

Command 245: Request equipment category ID

This command permits receipt of the chain of characters that identifies the type of device in question from the corresponding Hopper U-II U-II.

Data sent: ---

Data received: **"Payout"**

Command 244: Request product code

With this command, the corresponding Hopper U-II sends the machine the product code of the device.

Data sent: ---

Data received: **DATA**

Where: DATA= "Payout" if it is a Hopper U-II ccTalk standard.

DATA= "UPLUS" if it is a Hopper U-II ccTalk Plus.

Command 242: Request serial number

With this command, the corresponding Hopper U-II sends the machine the serial number of the device with a code of 3 bytes. This command has been implemented so the machine has a security code which is necessary for the payout of the coins.

Data sent: ---

Data received: **[Serial 1 - LSB] [Serial 2] [Serial 3 - MSB]**

The serial number is unique for each Hopper U-II manufactured.

Command 241: Request software version

With this command, the corresponding Hopper U-II sends the Machine the current software version of the device.

ccTalk® does not establish any restriction concerning the format of the reply message.

Data sent: ---

Data received:: **[Data 1] [Data 2]**

Where: Data 1 = Byte prior to the dot of the program version

Data 2 = Byte following the dot of the program version

Command 236: Read opto states

To this command, the Hopper U-II responds by sending a byte which indicates the value of the empty and full detection optos.

Data sent: ---

Data received: **[Data 1]**

Where: Data 1 = value of the detectors

Data 1 .bit 0 = Empty detector

Data 1 .bit 1 = Full detector

Data 1.bits 2 to 7 Not used. Replied with 0.

The following table shows the value of these empty and full bits (bits 0 and 1) depending on the coin load of the Hopper U-II U-II:

Load	Bit 0	Bit 1
Empty	1	1
½ Load	0	1
Full	0	0

Table 3. Value for full and empty Hopper U-II.

If the Hopper U-II does not have these sensors, the bit will be 0 0.

This command is only for testing and should not be used for payout information.

Command 219: Enter New PIN number

This command is used to change, as often as required, the PIN number of the Hopper U-II. As it is a command that protected with a PIN, the present PIN number must be used to be able to use the command.

Data sent: [PIN 1] [PIN 2] [PIN 3] [PIN 4]

Data received: **ACK**

The new PIN number will be maintained after power off or a reset.

Modifying the PIN number to 0, will deactivate the protection with PIN numbers.

Command 218: Enter PIN number.

The PIN number can be introduced with this command. The PIN number is a binary code of 32 bits (4.294.967.296 combinations).

Data sent: [PIN 1] [PIN 2] [PIN 3] [PIN 4]

Data received: **ACK**

If the PIN number introduced is correct or incorrect, the Hopper U-II will send an (ACK).

If the PIN system is active (PIN Number different to 0), all commands implemented in the Hopper U-II, except the command "ADDRESS POLL, 253" are protected. *See Note 1

If the PIN is activated and the command does not have the correct PIN, there will be no answer from the Hopper U-II.

If the PIN is activated, it must be sent after every power off and reset so that the Hopper U-II will execute the command.

Note 1: there is a special firmware version where only the command "DISPENSE HOPPER U-II COINS, 167" is protected with the PIN.

Command 192: Request build code

With this command, the corresponding Hopper U-II sends the Machine the device assembly code.

Data sent: ---

Data received: **"Payout"**

Command 172: Emergency stop

This command automatically stops the payment sequence, due to the Dispense Hopper Coins or Payment commands, and transmits the number of coins that still have to be paid.

Data sent: ---

Data received: **[Data 1]**

Where: Data 1 = Number of coins that still have to be paid (between 1 and 255).

If the command is executed while not paying out, the reply will be the same but with the number of coins left to pay at 0.

The reply to the command will be received by the machines after 300 milliseconds due to the processing time.

Command 168: Request Hopper U-II dispense count

This command causes the Hopper to send the Machine the number of coins paid since the last reset.

Data sent: ---

Data received: **[Data 1] [Data 2] [Data 3]**

Where: [Data 1] = Number of coins paid (LSB).

[Data 2] = Number of coins paid.

[Data 3] = Number of coins paid (MSB).

Number of coins paid is between 0 and 16,777,215.

The counter is erased with a reset. (Powering off the Hopper U-II U –II will not erase the counter).

Command 167: Dispense Hopper U-II coins

This command has been implemented with format b of the standard ccTalk.

This command provokes the payment of the number of coins indicated in Data 4 byte, and in turns sends the necessary safety code to permit the payment to be made in Data 1, 2 and 3.

Data sent: [Data 1] [Data 2] [Data 3] [Data 4]

Data received: ACK

Data received: NACK

Where: [Data 1] = less significant byte of the safety code (LSB)

[Data 2] = Intermediate byte of the safety code

[Data 3] = Most significant byte of the safety code (MSB)

[Data 4] = Number of coins of type 1 to be paid (between 1 and 255).

If the Hopper U-II can execute the command, it returns a positive acknowledge string and begins to make the payment until one of the following occurs:

- Extraction of all the coins indicated
- Detection of a maximum span
- Reception of a cancel command
- Detection of an error
- Reset hardware (power failure) and software.

In the event that the Hopper U-II cannot execute the command, it returns a negative acknowledgement string and continues in the status it was in. Sending this negative acknowledgement may be due to the following reasons:

- Hopper U-II in error state
- Another command being executed
- Safety code received not correct

Command 166: Request Hopper U-II status

This command requests information from the Hopper U-II about different status parameters.

Data sent: ---

Data received: **[Data 1] [Data 2] [Data 3] [Data 4]**

Where:

If the Hopper U-II is paying out when command is received:

[Data 1] = Number of payments made since the last reset

[Data 2] = Number of coins still to be paid

[Data 3] = Number of coins paid in last payment

[Data 4] = 0

If the Hopper U-II is at rest when this command is sent:

[Data 1] = Number of payments made since the last reset

[Data 2] = 0

[Data 3] = Number of coins paid in last payment

[Data 4] = Number of coins still to be paid for last payment

When the number of payouts (Data 1) is 255, the next payment command sets the counter to 1. The only way that the counter will go to 0, is after a reset.

The information from this command is saves to the EEPROM, so even if there is a fault or power off, we can recuperate the state of the last command on starting up the hopper. After a reset, data 4 is reset to 0.

Command 164: Enable Hopper U-II

This command should be used to activate the Hopper U-II before using any payout command.

Data sent: **[Data 1]**

Data received: **ACK**

Where: If Data1 = A5H, the hopper activates

If Data 1 is not A5H the hopper deactivates.

Command 163: Test Hopper U-II

This command causes information to be sent about several error and operation flags of the Hopper U-II U-II. The flags keep information about the event occurred until enquiries are made about them, and after passing the information to the machine they are reset.

Data sent: ---

Data received: **[Data 1] [Data 2]**

(Los Hopper U-II ccTalk standard, will only give Data 1 as a reply, whereas Hopper U-II ccTalk PLUS, will give all data).

Where: Data 1 =

Bit 0 - Maximum absolute current exceeded.

Bit 1 - Maximum payment time exceeded.

Bit 2 - Motor turned in opposite direction during last payment to eliminate a blockage.

Bit 3 - Opto fraud attempt, release blocked during standby.

Bit 4 - Opto fraud attempt, short-circuit during standby. (NOT IMPLEMENTED)

Bit 5 - Opto blocked during payment.

Bit 6 - Not used

Bit 7 - Payment disabled.

1 = True, 0 = False

Data 2 =

Bit 8- Not used

Bit 9 – Not used

Bit 10 – Coin sensor not detected

Bit 11 – Double count sensor not detected

Bit 12 – Attempt at fraud. Timeout of double count sensor

Bit 13 – Attempt at fraud. Coin exit detected without trigger movement.

Bit 14 – Attempt at fraud. Coin exit not detected but trigger movement detected.

Bit 15 – Double count sensor detected in configuration of simple count.

1 = True, 0 = False

Command 160: Request cipher Key

Data sent: **---**

Data received: **<Variable>**

This command requests the slave device for a password that forms part of the encrypting algorithm of the payment system.

Command 141: Request firmware upgrade capability

Data sent: **---**

Data received: **[firmware options]**

Where: firmware options=

0 if the firmware is in ROM /EPROM

1 if the firmware is in FLASH/EPROM with update capacity.

In all Hopper U-II ccTalk, the answer will be **[firmware options]** = 1, except in those for the Italian market, they cannot be updated using ccTalk commands.

Command 140: Upload firmware

This command is used to update the firmware of the device.

Data sent: **[block] [line] [data1] [data2] ... [data128]**

Data received: **ACK**

This command is not available for the Italian market, they cannot be updated using ccTalk commands, and the reply will be a "**NACK**" to this command.

Command 139: Begin firmware upgrade

Data sent: **---**

Data received: **ACK**

When the Hopper U-II receives this command, the firmware updating commences. This command is not available for the Italian market, they cannot be updated using ccTalk commands, and the reply will be a "**NACK**" to this command.

Command 138: Finish firmware upgrade

Data sent: **---**

Data received: **ACK**

When the Hopper U-II receives this command, the firmware updating finishes. This command is not available for the Italian market, they cannot be updated using ccTalk commands, and the reply will be a "**NACK**" to this command

Command 04: Request comms revision

As a reply to this command, the Hopper U-II sends the implementation level of the **ccTalk®** protocol (01H in our case) and the communication software version.

Data sent: ---

Data received: **[ccTalk Level] [major revision] [minor revision]**

Where: ccTalk Level = 01H

Major revision= 04H

Minor revision= 03H

Command 01: Reset device

Data sent: ---

Data received: **ACK**

This command causes the Hopper U-II to carry out a software reset. The affected

Hopper U-II sends a positive acknowledgement string immediately before making the reset.

When a Reset command is sent to the Hopper U-II, it is necessary to wait at least 50 milliseconds before sending a Payment command (A7h).

3. START UP SEQUENCE OF THE HOPPER U-II U-II.

This chapter indicates the minimum action and commands that should be carried out on the Hopper U-II so that it is ready to work.

3.1. HOPPER U-II ccTalk STANDARD.

1.- Selection of the address using the dipswitches or ccTalk commands, as described in section 5 of the manual "FT Hopper U-II".

2.- Send command " **SIMPLE POLL, FEh"** to confirm presence of the Hopper U-II on the bus.

3.- Send command **"ENABLE HOPPER U-II, A4h"** to activate the Hopper U-II and enable payouts.

4.- Send command **"REQUEST SERIAL NUMBER, F2h"** to get security information required for paying out.

From this point it is possible to make payouts.

3.2. HOPPER U-II ccTalk ENCRYPTED.

1.- Selection of the address using the dipswitches or ccTalk commands, as described in section 6 of the manual "FT Hopper U-II".

2.- Send command " **SIMPLE POLL, FEh"** to confirm presence of the Hopper U-II on the bus.

3.- Send command **"ENABLE HOPPER U-II, A4h"** to activate the Hopper U-II and enable payouts.

4.- Send command **"REQUEST CIPHER KEY, A0h"**

From this point it is possible to make payouts.

4. ADDRESS MANAGEMENT

There are two ways to assign or change the ccTalk address of the Hopper U-II. The choice of which method to use is carried out by positioning switch 6. When switch 6 is ON the Hopper U-II takes its values from switches 1 to 4, and when switch 6 is OFF, it takes the values from the values stored in the EPROM.

4.1. CONFIGURATION OF THE ADDRESS USING SWITCHES (SWITCH 6 ON).

Following the instructions in the section "***¡Error! No se encuentra el origen de la referencia.***", in the manual "FT_Hopper U-II", the Hopper U-II has 4 dip switches that allow up to 16 different addresses to be configured. See the table to configure the required address.

4.2. CONFIGURATION OF THE ADDRESS USING CCTALK COMMANDS (SW 6 OFF).

There are various commands that send new addresses to the hopper. The new address is saved in the EEPROM memory. See list.

The Hopper U-II are factory programmed with the value 0 in the EEPROM, which means that they take their value from switches 1 to 4, independently of what switch 6 is programmed.

Once you modify this value of 0 using ccTalk commands, the hopper will take its address from the configuration of the switches or of the value of the EEPROM depending on the value of switch 6.

The value saved to the EEPROM is maintained after a reset of power off.