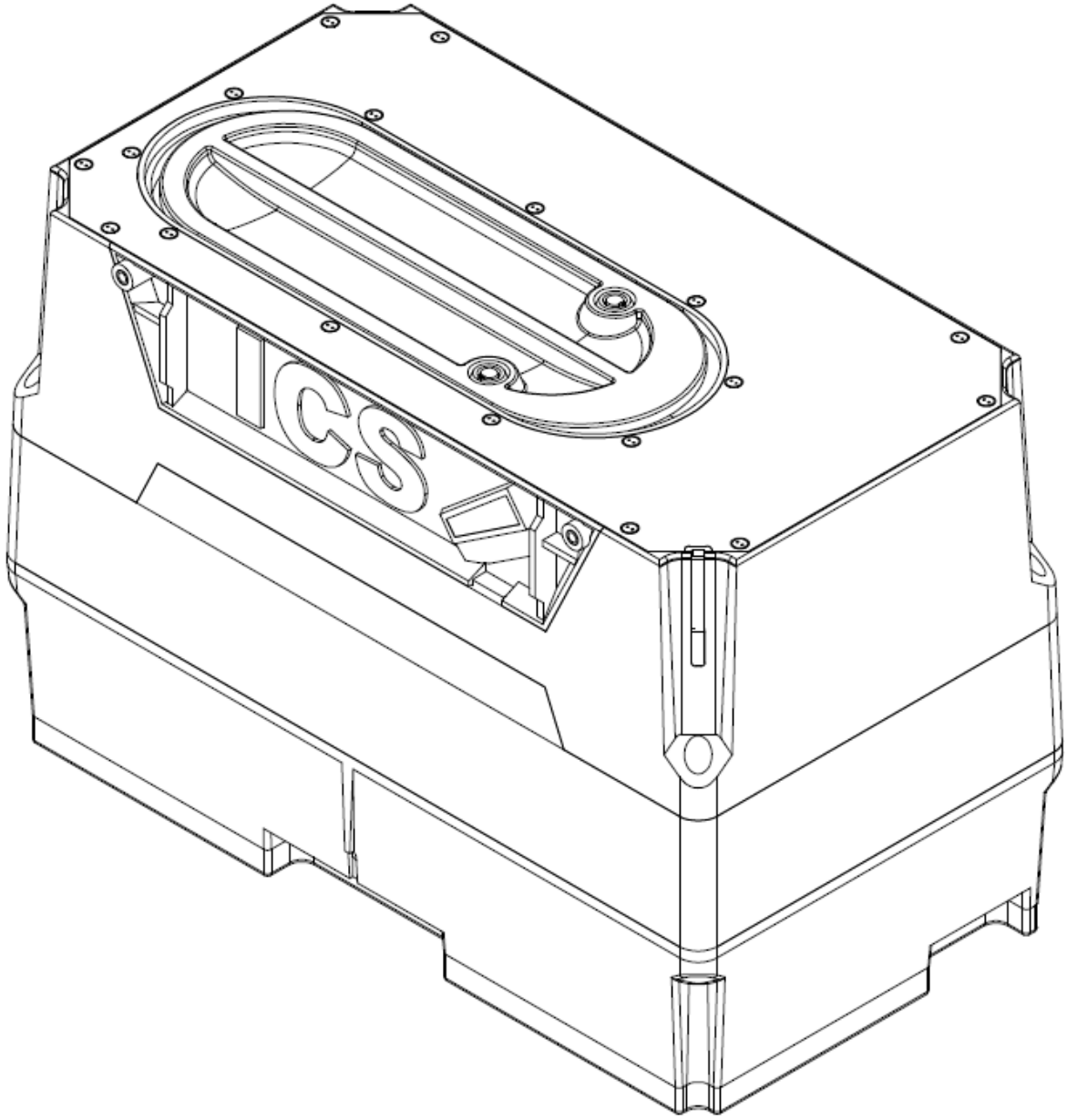
Intelligent Coin System
ICS



SUZOHAPP

Contents

Introduction

The Intelligent Coin System (ICS) is a patented innovative coin handling machine designed and produced by SUZOHAPP. The ICS is different from a “traditional” coin systems. The ICS can process up to 8 different coins without the need of pre-sorting the coins by the user. The ICS is always standby for the next payout by using the idle time or during payout to pre-sort the coins.

The ICS can handle up to 1300 coins of different sizes, sort these automatically and determine how the payout needs to be divided with the coins available. Because of the ICS construction, it can payout 4 coins at the same time. This ensures a fast payout for the consumer.

The compact design of the ICS makes it easily integrated into smaller payout machines like self-checkout registers or parking ticket machines without the need of custom coin routing.

The ICS can be used in many applications, for example: self-checkout stations for supermarkets, parking ticket machines, money changers, vending, amusement and gaming machines.

Payout can be done by a direct coin payout command from each coin tube (in total 8) or using the payout amount command, in which the ICS calculates the changing amount of coins based on the current tube stock levels.

Highlighted features:

- Payout speed consistency (buffer stock)
- Payout redundancy
- Compact design
- Mixed coin sorter & payout system
- Customizable for country specific coin mix
- Bulk capacity ≈ 1300 coins (in euro mix)
- Coin diameter 15 – 32,5mm
- Coin thickness 1,3 – 3,25mm
- Cost reduction in final machine

Using the ICS

This chapter explains the steps needed to start using the ICS.

A fast way to start is using the cctalk interface (SUZOHAPP part nr 107-0251) plus a 3 wire cable connecting the cctalk interface board to the ICS (SUZOHAPP part nr 157-0350) and a 12V power supply (SUZOHAPP part nr 150-0072).
The ICS can be evaluated using a ICS-TestProgram that runs on a PC (please request from technical service)

Check if ICS is getting powered

When the ICS receives between 12V and 24V and no errors are present the blue indicator LED will be on.

Check the cctalk connection and Reset the ICS

Send the RESET command to the ICS. If the command is ACK'ed then the cctalk interface works and the ICS will reset. You will hear the coin rejector solenoid clicking.

Fill the ICS with coins

See Table 1: Examples of best mix for the bulk of euro coins in the ICS.
Note that it is not necessary to start sorting of the coins into the tubes manually. The user must set the number of coins put into the ICS for each denomination using the SET_TOTAL_COINS_IN command (71 Dec). Depending on the setting 'start sorting after Reset or after Set Total Coins In' the sorting will start immediately after the SET_TOTAL_COINS_IN command or after the RESET command. Note that the sorting is always started automatically after a payout command if necessary. This ICS will keep sorting coins into the tubes even after a payout has finished, until the tubes are full or a sort timeout occurs.

Tube Stock

After sorting starts, coins fall into the tube. When a coin enters the tube it first interrupts the 'count sensor', located at the top of the tube. This sensor is used for stock accounting and tube overflow management.
During the fall into the tube, the coin passes a second sensor, called the 'full level sensor', located in the middle of the tube. This sensor is used to indicate that the tube is full and is also used to correct the tube stock when necessary.
The tube stock has a minimum level of 2 coins. This is to prevent the tube running empty during a payout. If the tube runs empty while a new coin falls into the tube, it could create a tube payout jam.

Start Payout

Payout can be done in 2 ways: Payout by denomination and payout by amount.
- Payout by denomination (see Table 4, command 20 Dec) gives to user precise control over the nr of coins to payout from the tubes.
- Payout by amount (command 22 Dec). This is the recommended way of dispensing coins from the ICS, since the ICS is able to see which tubes should payout coins in terms of their stock, status and availability.

Payout by denomination (not recommended)

Before a payout from a tube is started, the user must check if the tube can be used for payout (not in error, enough coins in the ICS or tube, etc). If a tube is full, it is recommended to use it on a regular basis, so that the tube stock will be synchronized to it's real stock level as indicated by the tube level sensors in the tube. In some cases, when the tube is not used for a long time, it might occur that, during sorting, a coin slips over the coin rejector and falls into an already full tube. Eventually the tube will set it's count sensor blocked status bit. When this happens, the user is strongly recommended to payout from this tube as soon as possible so that the stock level drops again and new coins falling into the tube are detected again. Keep using a full tube until the full level sensor becomes free again allowing the tube stock to be synchronized again.

Payout by amount (recommended)

This is the recommended payout method for the ICS. The ICS automatically calculates which tubes should be used for the requested amount to pay. Tubes that are overfull with coins will be used first, reducing any potential problems due to tubes becoming overfull. Tubes that can't be used for a payout are automatically discarded. Based on the stock levels in the tube, the most economic and fastest payout is calculated.

Note that in both payout methods, the ICS will start automatically filling the tubes when necessary. No need to use the START_SORTING command, to start sorting manually.

1 ICS Operation

The Intelligent Coin System (ICS) is a compact payout unit designed to payout small amounts of coins. Typical use is the payout of change money that is not possible via the (smallest) notes. So, in case of euro's, an ICS payout is typical less than 10 euro.

The ICS unit can be used as a payout device only, in which case the ICS has to be filled with coins manually, or as part of a coin (recycling) system, in which case the ICS is filled typically through a coin acceptor. The coin bulk inside the ICS are sorted into 8 tubes from which a payout can be done.

1.1 The ICS Coin Bulk Filling Process

Filling the ICS with coins needs some attention. In order to get the best payout results, all payout tubes must be filled with enough coins, so that after sorting all tube stock levels are approximately the same. After some payouts, the coin tubes stock will fall below a certain level and the ICS will start sorting coins again from the bulk inside into to tubes. Ideally all tubes will be refilled to the full level again during sorting. The sorting process is optimized by filling the ICS with an optimum amount of each coin denomination. Table 1 shows some ICS euro coin filling examples.

Euro In	1ct	2ct	5ct	10ct	20ct	50ct	100ct	200ct	Total coins In
€ 98,10	40	35	25	30	25	35	30	20	240
€ 196,20	80	70	50	60	50	70	60	40	480
€ 392,00	160	140	110	120	100	140	120	80	970
€ 488,50	200	170	140	150	120	170	150	100	1200

Table 1: Examples of best mix for the bulk of euro coins in the ICS

Not only the optimal mix of the different coin denominations should be taken into account, also the total amount of bulk coins inside the ICS should be considered. If the ICS is filled to its maximum level (1300 coins), the coin sorting process will be less optimal. Filling it up to 70 percent of its maximum level gives better results.

If the coins inside the ICS are fed through a coin acceptor that is connected to the ICS (ICS operation mode = 1), then the ICS may disable coin acceptance of a certain coin type in order to maintain the best mix of total coins inside the ICS. If, for example, the max nr of 2 euro coins inside the ICS is 60 for the current best mix of all coins, then the coin acceptor will sort the 2 euro coin to the cashbox if the total amount of 2 euro coins is more or equal then 60, although the ICS may not be full yet. The ICS will try to maintain the best mix of coins to operate optimally.

So, the ICS fills the coin tubes in an optimal way if the bulk of coins inside the ICS have an ideal mix of each coin denomination. Then the chance that all tubes will be filled again to their full level during sorting will be highest. Smart payout algorithms make sure that all tube coins have no large stock differences. So, if one tube is empty, most likely all other tubes are also at low stock level and the next sorting process will fill all tubes as much and efficient as possible.

If the ICS is used in a coin recycle system (ICS receives coins via a coin acceptor), then the ICS may only accept coins into the ICS if the total coins in is below 1300. See Table 1.

1.2 The Coin Sorting Process

Table 2 shows the ICS coin sorting commands

Header	Command
53	DISC_MOTOR_START
54	DISC_MOTOR_STOP
55	GET_SORTING_STATUS
56	SET_SORTING_STOP_CRITERIA
57	GET_SORTING_STOP_CRITERIA
58	SET_SORTING_START_CRITERIA
59	GET_SORTING_START_CRITERIA
71	ICS_SET_TOTAL_COINS_IN
72	ICS_GET_TOTAL_COINS_IN
73	ICS_CLEAR_TOTAL_COINS_OUT
74	ICS_GET_TOTAL_COINS_OUT
75	ICS_GET_TOTAL_AMOUNT_IN
76	ICS_GET_TOTAL_AMOUNT_OUT
77	ICS_GET_STATUS_AMOUNT_PAYOUT
78	ICS_EMPTY_ICS
79	ICS_GET_LEVEL_STATUS

Table 2: ICS Coin Sorting Commands

A payout is realized by 8 separate coin tubes inside the ICS. One tube can hold typically around 20-30 coins. The tubes are filled via a coin sorting system from a large cup inside the ICS.
The cup with unsorted coins can be filled manually or via a coin acceptor.

When the coin sorting process is started, a large disc inside the ICS starts rotating and moves the coins to the start of a sorting rail with increasing recesses. The coins will then fall along the rail. Small coins are sorted out at the start of the rail and the large coins at the end of the rail. Under the rail are the 8 tubes located, that store the sorted coins.

When a certain coin tube gets full, a solenoid at the start of the rail will push any additional coins from the rail back into the cup preventing the coin tube to flood. A diameter sensor located at the start of the rail determines the coin type and actives the solenoid if the tube for that coin type is full.

In order to get an optimal coin sorting process, the user should fill the ICS with an optimum mix of all coin denominations. Some optimal coin in mix values for the ICS are listed above. The number of coins for each coin type can be set with the ICS_SET_TOTAL_COINS_IN command and can be retrieved with the ICS_GET_TOTAL_COINS_IN command.

1.2.1 Coin sorting Start criteria

If one of the tubes stock falls below 80% of its full level stock, the ICS will start sorting again, so it will keep the tubes as full as possible.

1.2.2 Coin sorting Stop criteria

Once the coin sorting process is started, and the tubes are being filled again, the sorting process will stop if all tubes have reached at least 80% of the full level stock (see Table 3), or after a certain timeout period typical (7 – 14 sec) in which no more coins have been sorted anymore into the tubes.

Coin	1ct	2ct	5ct	10ct	20ct	50ct	100ct	200ct
Full Level	31	30	30	26	24	21	22	23

Table 3: Tube Full Level Settings for Euro

The sorting process can also be started manually using the DISC_MOTOR_START and DISC_MOTOR_STOP commands.

➡ Note that tube filling stops after 4 coins, when the tube was empty at the start. When the tube has paid 1 or more coins, it will be filled to its normal full level automatically. This is done to minimize any coin jams that could occur when filling a tube from an empty state.

1.2.3 ICS Coin In amounts

The total amount of money inside the ICS can be retrieved using the ICS_GET_TOTAL_COINS_IN or ICS_GET_TOTAL_AMOUNT_IN commands. If no coin acceptor is connected to the cctalk peripheral extension bus then the amount of coins in must be set manually using the ICS_SET_TOTAL_COINS_IN command.

1.2.4 Disabling Tubes

A tube (or coin) can be disabled from being filled by sending the TUBE_DISABLE command. All coins for this tube will then be rejected and will be ignored during the sorting process.

1.3 The Payout Process

Table 4 lists the ICS Payout commands

Header	Command
23	ABORT_PAYOUT
22	DISPENSE_AMOUNT
21	GET_DISPENSE_EVENTS
20	DISPENSE_COINS
77	GET_AMOUNT_PAID

Table 4: Payout Commands

The DISPENSE_COINS command can be given specifying the amount of coins to be dispensed from every tube or a DISPENSE_ AMOUNT command can be given to payout a certain amount of money. If the DISPENSE_COINS command is executed successfully the event counter is incremented and an ACK is returned. If a tube runs out of coins before the dispense could be completed, then a payout timeout will stop the tube payout and all tube counters will be updated.
Table 5: Payout events and counter data shows what happens with all tube counters after payout related events.

Event	Payout Events	ToPay	Remaining	Paid	UnPaid	Total	Life	Status
Reset	0	unchanged	0	unchanged	unchanged	unchanged	unchanged	Reset flag set
Dispense	incremented	nr coins to pay	nr coins to pay	0	nr coins to pay	unchanged	unchanged	unchanged
Coin paid	unchanged	unchanged	decremented	incremented	decremented	incremented	incremented	unchanged
Timeout	unchanged	unchanged	0	unchanged	unchanged	unchanged	unchanged	Timeout flag set
Abort	unchanged	unchanged	0	unchanged	unchanged	unchanged	unchanged	unchanged
Clear Total	unchanged	unchanged	unchanged	unchanged	unchanged	0	unchanged	unchanged

Table 5: Payout events and counter data

Note that the counters ToPay, Remaining, Paid and UnPaid are set at the start of a dispense command. After a reset or timeout, they will reflect the status of the last payout command executed.

The DISPENSE_ AMOUNT command dispenses a certain amount of money. The ICS calculates how many coins from each tube must be paid. The change algorithm tries to create equal stock levels for all tubes in order to optimize the coin sorting process. If the ICS cannot change the amount of money into coins (ex. some tube coins are empty), then the DISPENSE_ AMOUNT command will return a NAK and the event counter will not be incremented.

The ABORT_PAYOUT command can be used to abort a running payout. The number of coins paid are saved in non-volatile memory.
The Dispense Event Counter increments from 1-255 at each successful payout command. After 255 it will be set to 1 at the next payout command. After a system reset it will be set to 0.

The total amount of coins paid from the ICS can be retrieved using the ICS_GET_TOTAL_COINS_OUT or ICS_GET_TOTAL_AMOUNT_OUT commands. The ICS_CLEAR_TOTAL_COINS_OUT command resets the total coins paid from each tube to 0.

After each payout, the level of the coin tubes is checked and filled up again if necessary by starting the sorting process.

1.3.1 Payout amount monitoring

If an amount of money is dispensed, the payout process can be monitored using the ICS_GET_STATUS_AMOUNT_PAYOUT command, which is transmitted for example 5 times per second during the payout. The answer to the ICS_GET_STATUS_AMOUNT_PAYOUT command contains also the status byte (see table below) followed by the current amount unpaid so far.

Bit	Description
0	Payout amount in progress
1	Payout amount completed successfully
2	Payout amount was aborted by host
3	Payout amount terminated incomplete (timeout)
4	Payout amount was terminated due to a Power Fail
5	Payout amount was terminated due to a jam
6	Payout amount was terminated due to a tube motor error
7	Reset occurred during last payout amount

Table 6: Bit definitions Status byte in response from ICS_GET_STATUS_AMOUNT_PAYOUT

Example: Payout 2,35

```
Tx Data:  03 03 01 46 16 00 EB B2
Rx Data:  01 09 03 00 09 00 00 01 01 01 00 00 01 E6
Dispense money 2,35 :  0  0  1  1  1  0  0  1

Tx Data:  03 01 01 46 4D 68
Rx Data:  01 03 03 00 01 EB 00 0D
Get Status Payout Amount - Status: 1 UnPaid: 2,35

Tx Data:  03 01 01 46 4D 68
Rx Data:  01 03 03 00 01 05 00 F3
Get Status Payout Amount - Status: 1 UnPaid: 0,05

Tx Data:  03 01 01 46 4D 68
Rx Data:  01 03 03 00 02 00 00 F7
Get Status Payout Amount - Status: 2 UnPaid: 0,00
```

1.3.2 Power Fail

Just before a payout is started, the number of coins to pay (1 byte integer) from each tube is saved in non-volatile memory. After the payout is finished, the number of coins paid (1 byte integer) from each tube is saved also in non-volatile memory together with a tube total life dispense counter (4 byte integer). The total life dispense counter can be used for coin accounting over a certain period and for ICS service reasons.

If the power fails during a payout, then after the ICS powers up again, the last payout number of coins to pay and the number of coins paid can be retrieved for each tube. The host application can then decide how to finish the last payout.

1.4 The Coin Tubes

The 8 coin tubes inside the ICS system can be monitored and controlled by commands.

Header	Command	Single	All
48	TUBE_DISABLE	✓	
47	TUBE_ENABLE	✓	
39	TUBE_GET_TOTAL_LIFE_CNT		✓
38	TUBE_EMPTY_TUBE	✓	
37	TUBE_SET_NAME	✓	
36	TUBE_GET_NAME	✓	
35	TUBE_GET_STATUS		✓
34	TUBE_GET_COINS_REMAINING		✓
33	TUBE_GET_COINS_UNPAID		✓
32	TUBE_GET_COINS_PAID		✓
30	TUBE_GET_LEVEL_STATUS		✓

Table 7: Tube Commands

1.4.1 Tube Coin Type

The coin type for each tube can be given a name (TUBE_SET_NAME and TUBE_GET_NAME).
The coin type name is exactly 6 characters long and uses the following format: the first 2 characters define the currency, the next 3 digits the value and the last character is a revision character.
Ex. The 50 euro cent coin would we defined as" EU050A". Note that the coin value is determined from the coin type name, so be sure to specify the coin type properly.

1.4.2 Tube Level Status

The tube stock level status can be retrieved using command TUBE_GET_LEVEL_STATUS. When the stock of coins interrupts the tube level sensor, the tube will be flagged full. The response of the Tube Level Status command consists of 8 bytes, 1 status byte for each tube.

Level status byte	Meaning
0	Empty
1	Low level (< 1/3 of the full level)
2	Normal
3	High (>= ¾ of the full level setting)

Table 8: Tube Level Status Byte

If the tube level sensor becomes interrupted, than tube status register (see 1.4.4 Tube Status) bit 7 will be set, indicating that the tube full level is reached. Inside the ICS the actual coin stock for each tube is incremented when a coin passes the diameter sensor.

1.4.3 Tube Stock Correction

When the tube becomes full in will interrupt the full level opto sensor. When this happens the software will correct the tube stock to its full level settings as listed in Table 3. When a tube becomes empty during a payout (payout timeout status flag will be set) and the stock level is not 0, (unexpected since a payout only starts when the tube stock level indicates that there are enough coins in the tube to pay) then it will be corrected to 0.

1.4.4 Tube Status

Command TUBE_GET_STATUS is used to retrieve the current status of all tubes. 8 status bytes are returned, the first from tube 1. It is typically used after all tubes indicate that the payout has been terminated. This is the case when command TUBE_GET_ COINS_REMAINING returns 0 for all tubes. Table 9 shows the meaning of the status byte .

Bit	Description
0	Enabled
1	Payout Jam
2	Payout Timeout
3	Motor Error
4	Payout Sensor Blocked
5	Reset
6	Count Sensor Blocked
7	Tube Full

Table 9: Tube Status Byte

The Enabled flag indicates if the tube is enabled or disabled. A tube can be enabled by the TUBE_ENABLE or TUBE_DISABLE command. Disabled tubes are not used during changing an amount of money.
All tubes are enabled after a reset.

The Payout Jam flag is set when the tube payout disk jams during a payout. It is retained after a reset command and cleared after a power cycle of the ICS. It is also cleared after a successful payout of a jammed tube using the DISPENSE_COINS command.
Note that a jammed tube will not be used during the changing of an amount of money and is not cleared by the DISPENSE_AMOUNT command.

The Payout Timeout flag is set when the tube does not payout a coin within 2 seconds and after retrying 3 times. The tube is then assumed to be empty. Cleared at the start of a payout.

The Motor Error flag will be set if the ICS is unable to communicate with the Tubes. No payout can be started. Cleared at ICS reset.

The Payout Sensor Blocked flag will be set if a coin blocks the tube coin exit for more than 0.5 sec. The tube payout will be aborted and the status flag will be set.
A new payout can be started when this flag is set, so that a blocking can be ejected if necessary. Cleared at the start of a payout.

If a tube motor resets, the reset flag will be set. Note that after reading this flag it will be automatically cleared.

The Count Sensor Blocked flag will be set if the count sensor becomes interrupted for more than 1 sec. It will be cleared as soon as the count sensor is not interrupted anymore.

The Tube Full flag will be set if the level sensor becomes interrupted. A level debounce time of 100ms is used. Cleared when the level sensor becomes un-interrupted.

1.4.5 Tube Emptying

A tube is emptied by sending the TUBE_EMPTY_TUBE command. One or more tubes can be emptied at the same time by sending 1 or more TUBE_EMPTY_TUBE commands after each other. The emptying process can be monitored by sending the TUBE_GET_STATUS command which will return a payout timeout if the tube has become empty. After that the host may send the TUBE_GET_COINS_PAID command to retrieve the number of coins that are dispensed by the TUBE_EMPTY_TUBE command.

1.4.6 Tube Counters

The tube coin counters are used to count and monitor the payout process.
The DISPENSE COINS command controls all 8 tubes at the same time. So, if only 1 tube must pay coins, all other tubes have their coins to pay set to 0.

When a DISPENSE COINS command is given, the following actions are taken for each tube that must pay coins:
- The number of coins to pay is stored in REMAINING_COINS.
- COINS_UNPAID is set to 0
- COINS_PAID is set to 0

These counter values can be retrieved using commands TUBE_GET_COINS_REMAINING, TUBE_GET_COINS_UNPAID and TUBE_GET_COINS_PAID.

Then the payout is started. After each coin payout, the REMAINING_COINS counter is decremented and the COINS_PAID counter, TUBE_TOTAL _COINS_OUT and the TUBE_LIFE_TOTAL counters are incremented. The Host Machine can poll the REMAINING_COINS counter regularly (ex. 5 times per sec) to check if the REMAINING_COINS counter has become 0.
If this counter has become 0, then the payout has terminated in a normal way. This means that all coins are paid or the payout has been terminated in a controlled way because an exception occurred. After the payout of a tube has been terminated in a controlled way (REMAINING_COINS counter has become 0), the number of unpaid coins is stored in the COINS_UNPAID counter and the number of coins paid is stored in the COINS_PAID counter. The TUBE_LIFE_TOTAL counter is incremented for each coin paid and can't be reset to 0. So, this is a counter that will show the total amount of coins paid from a tube since it's started counting. This counter can be used for coin accounting and maintenance reasons.
Note that most of the tube commands in Table 7 operate on all 8 tubes with only one command. The commands that are more of a tube setting can operate only on a single tube.
Usually the Tube Status command is transmitted after all tubes have finished their payout to check for any tube exceptions.

1.4.7 ICS Error Status

The RGB Status Indicator shows the error or warning code in case an error occurs.
The ICS part that generates an error is shown by the nr of red flashes.
The ICS part that generates a warning is shown by the nr of green flashes.
The associated error or warning cause is shown by the nr of blue flashes.

Nr Red/Green Flashes	ICS Part in error	Error Status bit	Nr Blue Flashes	Error/Warning
1	System Errors	b0	1	Eeprom access error
		b1	2	Coin reject coil error
		b2	3	Sorting motor error
		b3	4	Tilt Sensor error
	System Warnings	b4	1	Eeprom Settings Initialized to Default
		b5	2	Eeprom Payout Data Initialized to Default
		b6	3	Eeprom System Data Initialized to Default
		b7	4	ICS tilting warning / Sorting motor fuse blown
2	Coin Tube Errors	b0	1	Tube motor or jam error. See note 1.
		b1	2	Payout sensor blocked or in error
		b2	3	Coin Sorting Jam / Tube Blocked Error
	Coin Tube Warnings	b3	1	Count Sensor Blocked warning
		b4	2	Payout sensor dirty
		b5	3	Level sensor dirty
		b6	4	Count sensor dirty
		b7	5	Motor opto sensor dirty
3	Coin Acceptor Errors	b0	1	Coin name error
		b1	2	Event counter error
		b2	3	Sorter error
		b3	4	CCTalk communication error
4	Diameter Sensor Errors	b0	1	Operation error
		b1	2	Coin not calibrated error
		b2	3	Stock count overflow error
	Diameter Sensor Warnings	b3	1	Sensor blocked warning
		b4	2	Too many tube stock corrections. (3 or more tubes have their stock corrected by more then 6 coins due to misreadings). Check, clean or re-calibrate the diameter sensor.
5	Escrow Unit	b0	1	Initialization error

Table 10: Device Codes and Errors and Warnings

Notes

1. Set if the tube payout disc reports an error after trying to dispense a coin. Usually caused by a coin jam between the payout disc and the tube. The error is cleared if the payout disc successfully returns to it's home position after a tube payout command or system reset.

1.4.8 Request Error Status

Command 122 (7A Hex) returns the ICS Error status.
Two bytes are returned: First byte is the device code and the second byte holds the fault code.
The fault code byte for some devices is split into error bits (bits 0 .. 3) and warning bits (4 .. 7).
If an error is generated, the ICS needs most probably to be serviced by a technician. Warnings can usually be handled by the host machine software.

Note that several error/warning bits can be set at the same time in the error status byte.
The RGB indicator shows only the first error or warning belonging to the Least Significant Bit that is set. Ex. If System error b1 and b2 are set, then the RGB indicator shows b1.

1.5 ICS Maintenance

The ICS Tube level opto sensors get dirty during its operation.
The tube level is checked 2 times per second. If the coin level and coin count sensors become dirty, the ICS error status will set their dirty warning bits and the ICS error led will flash a warning. See Table 10: Device Codes and Errors and Warnings.
In order to ensure proper ICS operation with dirty tube sensors, the opto sensor led will increase its intensity if it has become dirty.

1.6 ICS Tilt Detection

The ICS is equipped with a battery powered tilt sensor. When the power is off, the tilt sensor keeps working. Its main purpose is to warn the user that the ICS has been held into a position where coins could fall out of the tubes and could jam the ICS preventing proper operation. If this situation occurs, the ICS will generate a warning. The warning is cleared when the ICS is completely emptied, using the ICS_EMPTY_ICS command.

2 General commands

A cctalk message is build up from the following data bytes:

DestAddr, nrDatabytes, SrcAddress, Header, dataByte0, dataByte1, … dataByteN, CheckSum

DestAddr is the cctalk device address byte of the ICS (default 3)
NrDataBytes is the number of data bytes between the Header and the CheckSum.
SrcAddr is the address of the host that transmits the message to the ICS (always 1).
Header is the cctalk command value.
All general cctalk commands are specified in the cctalk protocol standard from Money Controls.
See www.cctalk.org. All ICS supported general commands are listed in section 2.1.
ICS specific commands always start with header 70, followed by ICS specific subcommands and data and are listed in section 3 Detailed ICS commands explanation.

2.1 General system commands

Device address related commands

Header	Command
254	Simple poll
253	Address poll
252	Address clash
251	Address change
250	Address random

Table 11: Address commands

Device setup and info commands

Header	Command
255	Factory setup and test
246	Request manufacturer id
245	Request category id
244	Request product code
242	Request serial number
241	Request software revision
141	Request firmware upgrade capability
140	Upload firmware
139	Begin firmware upgrade
138	Finish firmware upgrade
122	Request Error Status

Table 12: General info and setup commands

3 Detailed ICS commands explanation

3.1 Header 70: ICS specific command

Header 70 (46 Hex) is used to indicate that an ICS specific command follows.

Transmitted data: <ICS Specific Command>, [txDataBytes]
Received data : <ACK>or<NAK>, [rxDataBytes]

txDataBytes and rxDataBytes are optional data bytes.

3.2 System Commands

This section describes some commands that determine the ICS system behavior.

3.2.1 Command 92 (5C Hex) Get System Mode

This command returns the system mode byte.
Mode 0: Dispense function only, ICS is manually filled with coins.
Mode 1: A Coin acceptor with sorter is attached to the ICS cctalk peripheral bus.
Mode 2: A Coin acceptor with an Escrow unit and Sorter is attached to the cctalk peripheral bus.

Transmitted data: <ICS Command>, <92>
Received data : ACK, <mode byte>

3.2.2 Command 93 (5D Hex) Set System Mode

This command sets the system mode byte.

Transmitted data: <ICS Command>, <93>,<mode byte>
Received data : ACK

mode byte 0: Dispense function only, ICS is manually filled with coins.
mode byte 1: A Coin acceptor with sorter is attached to the ICS cctalk peripheral bus.
mode byte 2: A Coin acceptor with an Escrow unit and Sorter is attached to the cctalk peripheral bus.

See Chapter 4 for more details.

3.3 Payout related commands

This section describes the commands that are related to the payout of coins.
Note that there are 8 coin tubes. The first tube is called tube0 and the last tube is called tube7.

3.3.0 Command 20: Dispense coins

Command 20 (14Hex) is used to payout a certain amount of coins from each tube.

Transmitted data: <ICS Command>, <20>,<nrCoins Tube0>, …, <nrCoinsTube7>
Received data : <ACK> or <NAK>, <eventCounter>

The number of coins to dispense from a tube is a byte value.
The maximum nr of coins to pay from a tube is equal to Total In minus 2 coins (security stock), otherwise a NAK will be returned.
If the ICS has no coins (Total In is 0), is jammed or is not able to work, the payout can't be done and a NAK will be returned also. If the nr data bytes is unequal to 9, a NAK will be returned also.

Example: Tube 1 is out of stock, and you want to pay 1 coin from tube 0 and 1 coin from tube 2 and 2 coins from tube 5:

```
TxData: 03 09 01 46 14 01 00 01 00 00 02 00 00 95
RxData: 01 01 03 00 02 F9
```

Tube1 can't be used, so we specify to pay 0 coins from this tube.
From the response we conclude that the payout could be accepted and that the payout event counter is 2.
The payout event counter is only incremented if the payout command is accepted (ACK'ed).
After a reset it is set to 0. The event counter is a byte and rolls over from 255 to 1. It will only be 0 after a reset.

3.3.1 Command 21: Get payout event counter

Command 21 (15Hex) returns the number of accepted payout events. After a payout command is accepted by the ICS, it starts to payout the coins. During and after the payout, the user can monitor the payout process using status commands.

Transmitted data: <ICS Command>, <21>
Received data : <ACK>, <payout events counter>

See command 20 for more information about the payout event counter.

3.3.2 Command 22: Dispense Amount of Money

Command 22 (16 Hex) is used to payout a certain amount of money.
The amount is a 2-byte integer representing the amount in cents.
First the LSB is transmitted.

Transmitted data: <ICS Command>, <22>, <amount (LSB)>, <amount MSB>
Received data : <ACK> or <NAK>, <eventCounter>,<tube0>, …, <tube7>

Example: Payout 3,59 euro (167 Hex)

```
TxData: 03 03 01 46 16 67 01 35
RxData: 01 09 03 00 02 00 02 00 01 00 01 01 01 EB
```

The response data bytes have the following meaning:
Data byte 0 is the payout event counter value.
Data byte 1 is the number of coins that will be paid from tube 0 up to data byte 8 which is the number of coins that will be paid from tube 7.

The changing process depends on the tube stock values. It may be different for the same payout value. If the ICS can't change the amount a NAK is returned. Coin payout per tube is limited to max 9 coins per tube.

The payout event counter is only incremented if the payout command is accepted (ACK'ed).
After a reset it is set to 0. The event counter is a byte and rolls over from 255 to 1.
It will only be 0 after a reset.

3.3.3 Command 23: Abort Payout

Command 23 (17Hex) is used to abort an ongoing payout.

Transmitted data: <ICS Command>, <23>
Received data : <ACK>

Coins paid and coins unPaid from each tube are saved after aborting a payout.

3.3.4 Command 77: Get Status Payout Amount

Command 77 (4D Hex) is used to monitor the payout of an amount of coins or an amount of money. First the payout status byte is returned followed by a 2-byte value which represents the unpaid amount. Table 6 explains the bit definitions in the status byte.

Transmitted data: <ICS Command>, <77>
Received data: <ACK>, <status byte>, <amount unpaid LSB>, <amount unpaid MSB>

Bit	Description
0	Payout amount in progress
1	Payout amount completed successfully
2	Payout amount was aborted by host
3	Payout timeout of one of the tubes during the payout process occurred
4	Payout amount was terminated due to a Power Fail
5	Payout amount was terminated due to a jam
6	Payout amount was terminated due to a tube motor error
7	Reset occurred during last payout amount

Table 13: Bit definitions Status byte in response from ICS_GET_STATUS_AMOUNT_PAYOUT

Note that more than one bit may be set in the status byte. If a payout process is started, bit0 will be set. During the payout process, one of the tubes may run out of coins or experience a jam and will stop it's payout, and bit3 in the payout amount status will be set also. But since other tubes may still be busy paying out coins, bit0 remains set until all tubes have terminated their payouts.

Example: Payout 2,35

```
Tx Data:  03 03 01 46 16 00 EB B2
Rx Data:  01 09 03 00 09 00 00 01 01 01 00 00 01 E6
Dispense money 2,35 :  0  0  1  1  1  0  0  1

Tx Data:  03 01 01 46 4D 68
Rx Data:  01 03 03 00 01 EB 00 0D
Get Status Payout Amount - Status: 1 UnPaid: 2,35

Tx Data:  03 01 01 46 4D 68
Rx Data:  01 03 03 00 01 05 00 F3
Get Status Payout Amount - Status: 1 UnPaid: 0,05

Tx Data:  03 01 01 46 4D 68
Rx Data:  01 03 03 00 02 00 00 F7
Get Status Payout Amount - Status: 2 UnPaid: 0,00
```

Note that when command Dispense Coins is used, instead of Dispense Amount, then command 77 can also be used to monitor the payout process.

3.4 Coin Tube Commands

This section describes the coin tube related commands.

3.4.1 Command 30: Get Tube Level Status

Command 30 (1E Hex) returns the stock level of each tube. See Table 8: Tube Level Status Byte.

Level status byte	Meaning
0	Empty
1	Low level (< 1/3 of the full level)
2	Normal
3	High (>= ¾ of the full level setting)

Table 14: Tube Level Status Byte

A tube is flagged empty when the stock becomes 0. In normal payout situations the tube stock will not become 0, since payout from a tube will only be started if the tube stock contains at least 2 coins + number coins to pay. When a tube payout timeout occurs during a payout, and the Table 9: Tube Status Byte does not indicate any error or tube full conditions, the stock will be corrected to 0.

Bit	Description
0	Enabled
1	Payout Jam
2	Payout Timeout
3	Motor Error
4	Payout Sensor Blocked
5	Reset
6	Count Sensor Blocked
7	Tube Full

Table 15: Tube Status Byte

A tube is flagged low when the stock is lower than 1/3 of its full level setting parameter.

Coin	1ct	2ct	5ct	10ct	20ct	50ct	100ct	200ct
Full Level	28	28	29	26	23	22	22	22

Table 16: Tube Full Level Values (Euro)

A tube is flagged high when the stock becomes equal or higher than ¾ of its full level value parameter.

If the diameter sensor or coin rejector should not work properly, then even more coins will fall into the tube and eventually the coins will interrupt the tube upper coin counting sensor and a count sensor blocked error will be set in the tube status byte. Sorting will be stopped in this situation in order to prevent tube overflowing with coins. This condition can only be solved by inspecting the ICS for any ICS hardware or calibration errors. Note that the tube can still payout coins when the count sensor is blocked.

3.4.2 Command 31: Get Tube Stock

Command 31 (1F Hex) gets the stock from all 8 tubes.

Example:

```
Tx Data: 03 01 01 46 1F 96
Rx Data: 01 08 03 00 20 20 10 11 11 11 12 10 4F
Get tube stock:  32  32  16  17  17  17  18  16
```

The returned 8 bytes contain the tubes stock value.
First data byte is the stock of tube 0 (ex. 32), last data byte is the stock of tube 7 (ex. 16).
The stock number that is returned is the stock available for payout.
The security coins that can't be used for payout are already excluded.

3.4.3 Command 32: Tube Get Coins Paid

Command 32 (20 Hex) returns the number of coins paid from each tube since the past payout.
At the start of a payout it is set to 0. During a payout, it is incremented with each coin paid.

Transmitted data: <ICS Command>, <32>
Received data : <ACK><coins paid tube0>, …, <coins paid tube7>

8 data bytes are returned, since the number of coins paid per tube is a byte value.
When a tube is emptied using command 38, the coins paid counter is also reset to 0 before it is being emptied, and incremented during the emptying process.

3.4.4 Command 33: Tube Get Coins UnPaid

Command 33 (21 Hex) returns the number of coins unpaid from each tube since the last payout.
At the start of a payout it is set to 0. During a payout, it remains 0. After a payout it is set to the number of unpaid coins.

Transmitted data: <ICS Command>, <33>
Received data : <ACK><coins unpaid tube0>, …, <coins unpaid tube7>

8 data bytes are returned, since the number of coins unpaid per tube is a byte value.
When a tube is emptied using command 38, the coins unpaid counter is also reset to 0 before it is being emptied.

3.4.5 Command 34: Tube Get Coins Remaining

Command 34 (22 Hex) returns the number of remaining coins to pay from each tube during a payout.
At the start of a payout it is set to the number of coins to pay for each coin tube.
When the payout finishes it will always be 0. During a payout it will decrement for each coin paid.

Transmitted data: <ICS Command>, <34>
Received data : <ACK><coins remaining tube0>, …, <coins remaining tube7>

8 data bytes are returned, since the number of coins remaining per tube is a byte value.
When the number of coins remaining becomes 0 for a certain coin tube during a payout, then that coin tube has finished its own part of the total payout.

During the payout of a number of coins from each tube the get coins remaining command can be used to monitor the payout progress of each coin tube payout. When it becomes 0 the payout has finished. The Tube Get Remaining Coins command is usually used when paying out coins using command 20 Dispense Coins.

3.4.6 Command 35: Tube Get Status

Command 35 (23 Hex) is used to get the status register from each tube. The status register bits are defined in Table 9: Tube Status Byte.
After a payout from a tube has finished, this command returns the status from all 8 tubes.

Transmitted data: <ICS Command>, <35>
Received data : <ACK><status tube0>, …, <status tube7>

8 data bytes are returned, since the status per tube is a byte value.
After a tube payout, the host should check the status of the tube in order to check if the payout executed as expected. If, for example, the tube runs out of coins or becomes jammed or other error, it will generate a payout timeout. The Tube Get Status command is usually used when paying out coins using command 20 Dispense Coins.

3.4.7 Command 36: Tube Get Name

Command 36 (24 Hex) returns the tube coin type name of tube[tube nr].
This name is 6 characters (bytes) long. The name is a standard coin type definition format.

Transmitted data: <ICS Command>, <36>,<tube nr>
Received data : <ACK>,<id0>,<id1><val0>,<val1>,<val2>,<rev> or <NAK>

Example: EU010A means that the tube is filled with euro coins (EU), value 10ct (010) and has revision A.
If the tube nr is not between 0 and 7, a NAK is returned.

3.4.8 Command 37: Tube Set Name

Command 37 (25Hex) is used to set the tube coin type. It must adhere to the format specified in the previous chapter.

Transmitted data: <ICS Command>, <37>,<tube nr>,<id0>,<id1><val0>,<val1>,<val2>,<rev>
Received data : <ACK> or <NAK>

If the tube nr is not between 0 and 7, or the format is wrong, a NAK is returned.

3.4.9 Command 38: Empty Tube

Command 38 (26Hex) is used to empty a tube.

Transmitted data: <ICS Command>, <38>,<tube nr>
Received data : <ACK> or <NAK>

If the tube nr is not between 0 and 7, a NAK is returned.
After this command the tube starts emptying. The tube continuously tries to payout a coin, until it runs out of coins. Then the payout timeout flag will be set in the tube status register, indicating that it is empty. The tube stock will be set to 0 only if there are no tube payout errors. More tubes can be emptied at the same time.

3.4.10 Command 39: Get Tube Life Total Paid Counter

Command 39 (27Hex) returns the tube total paid counter during its life. This counter is incremented with each coin paid and can't be reset to 0. This number is used to get an indication of the wearing of the tube payout disk and can be used to generate maintenance reports. Only with special Factory Commands this counter can be reset to 0. In normal operation it is not desirable to reset the Life Total Paid counter, because it would become then a meaningless value.
All 8 tubes will return their life counter value. The value is a 4-byte integer, LSB transmitted first.

Transmitted data: <ICS Command>, <39>
Received data : <ACK>,<LSB life cnt0>, …, <MSB life cnt7>

A total of 8 times 4 bytes is 32 bytes is returned.

3.4.13 Command 47: Tube Enable

Command 47 (2F Hex) enables a tube.

Transmitted Data: <ICS Command>, <47>, <tube nr>
Received Data : ACK or <NAK>

After a system reset, all tubes have their status at enabled. Tubes that must be disabled must be disabled again after a system reset. Tubes are permanent disabled by setting the full level stock setting to 0.
A NAK will be returned if the tube nr is not between 0 and 7.

3.4.14 Command 48: Tube Disable

Command 48 (30 Hex) disables a tube.

Transmitted Data: <ICS Command>, <48>, <tube nr>
Received Data : ACK or <NAK>

This command disables a tube. Tubes that are disabled are not used for payout anymore.
A NAK will be returned if the tube nr is not between 0 and 7.

3.5 Coin Sorting Commands

Next commands control the coin sorting process of the ICS.

The ICS will start automatically sorting of a power up or reset when the stock of a tube is below 80% of it's full level.

If one or more tubes fall below their low level (80% of the full level setting), then the ICS starts sorting automatically to try to fill the tubes. The sorting will stop after a certain timeout period, which can be set using command 56. The timeout period can be set between 3 – 30 sec. Note that this timeout timer is reset back to it's timeout value each time a coin successfully sorts into a tube. As soon as no coins are sorted into the tubes anymore, the timeout timer will expire, and the sorting will stop due to a sort timeout. Sorting will also stop if the min number of full tubes (must be between 2 and 8) is reached of coarse. The sorting timeout and the min nr of full tubes after which sorting will stop can be set with the SET_STOP_SORTING_CRITERIA command (56).

Note that sorting is also started when a tube has not enough coins to do the requested payout. The sorting will be started if necessary and the sorting timeout timer is reset back to it's timeout value. The payout from this tube is suspended until enough coins have fallen into the tube (totalIn for this coin indicates if there are enough coins in the bowl that can be sorted into the tube). As soon as there are coins to pay again, the suspended payout from this tube will be resumed again . Sorting is stopped again if a sort timeout occurs or the min nr of full tubes is reached.

Note that when a sort timeout or tubes full event occurs, sorting will not stop immediately, but only after the next coin passes the diameter sensor in order to prevent that coins block the diameter sensor when sorting stops. This can take up another 7 sec if the next coin (for ex if the bowl is empty) does not arrive. After this period sorting will always stop.

3.5.1 Command 53: Start Sorting

Command 53 (35 Hex) will start the sorting process by command. Note that the coin sorting process is normally automatically started by the ICS when the tubes stock become low. See start sorting criteria (Chapter 1.2.1 Coin sorting Start criteria)

Transmitted Data: <ICS Command>, <53>
Received Data : <ACK>,<NAK>

The sorting will be automatically stopped if no coins are received anymore by the tubes for 10 seconds. A NAK is returned if the ICS was unable to start sorting. Most probable cause is an sorting jam in the ICS. Use the REQUEST_ERROR_STATUS command to check the error status.

3.5.2 Command 54: Stop Sorting

Command 54 (36 Hex) will stop the coin sorting process by command. Note that the coin sorting process is normally stopped when the tubes stock become full. See sorting stop criteria.

Transmitted Data: <ICS Command>, <54>
Received Data : <ACK>

3.5.3 Command 55: Get ICS Sorting Status (Obsoleted command)

Command 55 (37 Hex) returns the ICS Sorting Status.
It is used to monitor the ICS Sorting process.
When the sorting motor is running to fill the tubes during an empty ICS operation, the returned value is 1 as long as the sorting runs and becomes 2 when the operation has completed.

Transmitted Data: <ICS Command>, <55>
Received Data : <ACK><Sorting Status>

Sorting Status:
1: Sorting motor is running during emptying ICS
2. Sorting motor is not running and emptying of the ICS is not in progress anymore

3.5.4 Command 56: SET_SORTING_STOP_CRITERIA

This command is used to set the sorting stop criterium.
The sorting will stop as soon as a minimum number of tubes have reached their full level value (80% of the full level setting parameter) or when no coins have been sorted out into their tube for sortTimeout seconds.

Example:

```
TxData: 03 03 01 46 38 0A 08 69
RxData: 01 00 03 00 FC
```

The first received data byte holds the sortTimeout in seconds (ex. 10) and should be between 3 and 30 seconds.
The second received data byte holds the min number of tubes that must be full in order to stop sorting (ex. 8) and should be between 1 and 8.
Note: It is strongly recommended to leave the sortTimeout to 10 and the min number of tubes to 8.

3.5.5 Command 57: GET_SORTING_STOP_CRITERIA

This command retrieves the current stop criterium.

Example:

```
TxData: 03 01 01 46 39 7C
RxData: 01 02 03 00 0A 08 E8
```

The first byte returns the sortTimeout value in seconds and the second byte return the min nr of tubes that must be full.

3.5.6 Command 58: SET_SORTING_START_CRITERIA

This command is used to set the start sorting criterium.
If the stock of one or more tubes is falling below its stock low level setting, then the ICS will start sorting. The low level settings is a percentage of the tube's full level setting.
So, if a tube has a full level setting of 20 coins and a low level percentage setting of 50%, then the ICS will start sorting if the tube stock falls below 10 coins.

Example:

```
TxData: 03 0B 01 46 3A 50 50 50 50 50 50 50 50 03 00 EE
RxData: 01 00 03 00 FC
```

The red bytes show the low level percentages starting with tube 0 and ending with tube 7.
Note: It is recommended to leave all the values at 80 (0x50) percent.
The first green byte specifies how many tubes may become blocked (ie tube coins block the coin count sensor) before an 'ICS Sorting Error' is generated. Note that blocked tubes may become unblocked again by future payouts.
The second green byte: 0 means that the sorting is not automatically started after a Reset or Set TotalCoinsIn command. 1 means that sorting is automatically started after a Reset or Set TotalCoinsIn command.

3.5.7 Command 59: GET_SORTING_START_CRITERIA

This command is used to get the stop sorting criterium.

Example:

```
TxData: 03 01 01 46 3B 7A
RxData: 01 0A 03 00 50 50 50 50 50 50 50 50 03 00 6F
```

The red bytes show the low level percentages starting with tube 0 and ending with tube 7.

3.5.8 Command 90: ICS_SET_SORTER_OUTPUTS

This command is used to control the ICS sorter outputs on the ICS connector. See table 18.
The 2 pins are open-drain output pins and can be used to control 2 sorter units.
Below an example showing how to drive sorter output2 (pin 8) low (transistor on).

```
TxData: 03 02 01 46 5A 02 58
RxData: 01 00 03 00 FC
```

The red data byte bit 0 controls the sorter output 1 and bit 1 controls the sorter output 2.

3.5.9 Command 91: ICS_GET_SORTER_INPUTS

This command reads the (sorter) input lines on the ICS connector. See table 18.
Two input lines are available: Input line1 (pin 4) and Input line2 (pin 6).
Below an example how to read the input lines:

```
TxData: 03 02 01 46 5B 02 57
RxData: 01 01 03 00 03 F8
```

The red byte in the answer shows the status of the input lines. Bit 0 indicates the status of input 0 (connector pin 4) and bit 1 indicates the status of input 1 (connector pin 6). In this example both inputs are high.

3.6 Coin Totals In/Out Commands

3.6.1 Command 71: Set Total Coins In

Command 71 (47 Hex) sets the total of coins in of each type.
This command is used to set the amount of coins in that have entered the ICS. After each payout, total coins in is decremented also. With command Get Total Coins In the total coins inside the ICS can be retrieved.

If the ICS is used as a hopper only, then the ICS is typically filled before use with a bulk of coins. Total Coins In is set (new bulk added) each time the ICS is (re-)filled again.

If all coins are optimal mixed inside the ICS, then the coin sorting and coin payout process is optimized.
The best mix amount in of each coin type is used to disable or enable ICS controlled coin acceptors and to select the proper coin sorting start/stop criteria.

Transmitted data: <71>, <mix coin type0 (LSB first)>, … <mix coin type7 (MSB last)>
Received data: <ACK>or<NAK>

Table 1 shows some best mix ICS fill examples for each coin denomination.
Best mix coin values are 16-bits wide. First the LSB must be transmitted then the MSB.

Example: Setup the ICS by filling it with a best coin mix of 200 euro.

```
TxData: 03 11 01 46 47 5A 00 46 00 3C 00 3C 00 32 00 32 00 32 00 32 00 7E
RxData: 01 00 03 00 FC
```

In this example the best mix number for coin type 0 is 0x005A (90Dec), …, coin type 7 is 0x0032 (50Dec).
If a coin acceptor is connected via the ICS (so it's sorter is controlled by the ICS), any accepted coin above the best mix value will be directed to the cashbox instead of the ICS.

The best mix values are saved in Non-Volatile memory. If saving does not succeed, a NAK is replied.

3.6.2 Command 72: Get Total Coins In

Command 72 (48 Hex) returns the total number of each coin type inside the ICS. It is initialized by manual coin filling and incremented by coins going into the ICS (coin acceptor) and decremented by coins paid out of the ICS.

Transmitted data: <ICS Command>, <72>
Received data: <ACK>,<total coin type0 LSB>, <total coin type0 MSB>, … <total coin type7 MSB>

A total coin in counter is a 16-bit value, first the LSB is transmitted, then the MSB.

3.6.3 Command 73: Clear Total Coins Out

Command 73 (49 Hex) is used to reset the total coins out counter of each coin type.

Transmitted data: <ICS Command>, <73>
Received data: <ACK>or<NAK>

Total coin out counters are save in Non-Volatile memory. If saving fails a NAK is returned.

3.6.4 Command 74: Get Total Coins Out

Command 74 (4A Hex) is used to get to total coins paid counter of each coin type.
Total coin out counters are 32-bit values. So, in total 8 values of 4 bytes are returned.

Transmitted data: <ICS Command>, <74>
Received data: <ACK>, <totalOut0 (4 bytes LSB first)>, …, <totalOut7>

During the payout of coins, each total coin counter is incremented when a coin is paid.
After the total payout terminates, all counters are saved in Non-Volatile memory.

3.6.5 Command 75: Get Total Amount In

Command 75 (4B Hex) is used to get the value of the total amount of coins inside the ICS.
It is calculated by multiplying the total coins in counters of each coin type with its value and summing over all 8 coin types. The result is a 4-byte integer value transmitted LSB-first.

Transmitted data: <ICS Command>, <75>
Received data: <ACK>, <totalAmountIn (4 bytes LSB first)>

3.6.6 Command 76: Get Total Amount Out

Command 76 (4C Hex) is used to get the value of the total amount paid by the ICS.
It is calculated by multiplying the total coins out counters of each coin type with its value and summing over all 8 coin types. The result is a 4-byte integer value transmitted LSB-first.

Transmitted data: <76>
Received data: <ACK>, <totalAmountOut(4 bytes LSB first)>

3.6.7 Command 78: Empty ICS

Command 78 (4E Hex) is used to empty the ICS totally.
The ICS will start sorting, thus filling the tubes. When a tube has more than 9 coins, the tube will start payout of 8 coins. When the tubes do not receive coins anymore during a period of 14 sec during sorting, the ICS bowl is assumed to be empty and the sorting will stop. At this moment all tubes will start emptying their final stock. After this process, the ICS will be empty.
The Get ICS Empty Status command 79 can be used to monitor the emptying status.

Transmitted data: <ICS Command>, <78>
Received data: <ACK>

3.6.8 Command 79: Get ICS Empty Status

This command (4F Hex) returns the status of the emptying process, after it has been started by command 78 (Empty ICS) or command 38 (Empty tube).

Transmitted data: <ICS Command>, <79> or <38>
Received data: <ACK>, <result>

The result is a byte and has the following meaning:

ICS Empty Status	Description
Bit0	Set if the coin bowl emptying has finished (ready)
Bit1	Set if all tubes emptying have finished (ready)

Table 17: ICS Empty Status byte

After the command 79 has returned it's status byte with bit0 and bit1 set, the sorting has stopped (bowl sorting timeout has occurred) and the tubes have finished (tubes sorting timeout occurred). Since a payout timeout also occurs when a payout problem occurs, the user must send a Get Error Status command (122) also, to check if there are any errors that could have aborted the ICS emptying process and then the ICS is most probably not empty.

3.7 Updating Firmware ICS

The ICS firmware can be updated via the cctalk bus.
The cctalk protocol uses 3 commands to update the firmware of cctalk devices.

3.7.1 Command 139: Begin Firmware Upgrade

Command 139 (8B Hex) is used to put the ICS processor in boot mode.

Transmitted data: <139>, <module identifier>
Received data: <ACK>,<MSB Rev>,<LSB Rev> or <NAK>

The ICS uses the following module identifiers:

0 - Tube motor 1
2 - Tube motor 2
4 - Tube motor 3
6 - Tube motor 4
10 - Main processor board

The ICS main board processor is put into boot mode by keeping pin 4 of JP12 (Test Header on the main board) low after a reset. If a valid ICS application is already present in the ICS, then the application will do this automatically after receiving command 139.

If the bootloader is activated successfully it returns an ACK followed by 2 data bytes representing the bootloader firmware version.
The Tube motors can only be upgraded via a valid ICS application.

Example: Begin FW Upgrade Tubemotor 4 (tube6 and 7)

```
TX: 03 01 01 8B 06 6A
RX: 01 02 03 00 02 02 F6
```

Bootloader software version is: 02.02

3.7.2 Command 140: Upload Firmware

Command 140 (8C Hex) is used to upload firmware pages.
The first data byte is a block number and the second data byte the line number.
One line is 128 bytes long.
So in total 256 blocks of 256 lines of 128 bytes = 8 Megabytes can be programmed.

The ICS firmware is delivered in an intel-extended HEX format.
A data line in the intel-extended format contains the number of data bytes, start address and data bytes to be programmed. The user application must convert the .hex file into blocks and lines of 128 bytes.
This can be done by converting the start address of each data line in the .hex file into a block number and line number, and then read data from the .hex file until 128 bytes a read. Then the next line of 128 bytes must be created. After creating 255 lines, the block number will be incremented and the line number will become 0.

The firmware for the ICS is always delivered in an integer number of lines. The last 2 bytes of the last line holds the checksum of the firmware.

Transmitted bytes: <0x8C>, <block nr>,<line nr>,<data0>, … ,<data127>
Received bytes : <status>,<ACK> or NAK

For example: if a line of 128 data bytes must be programmed starting at address 0x00400000, then the block nr would be 0x80 and line nr 1, since 0x80(block nr) * 256 * 1(line nr) * 128 = 0x00400000.

Example: Programming Page 100 (0x64):

```
TX:  03 82 01 8C 00 64 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18
95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95
18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18
95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95
18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18 95 18
95 18 95 18 95 18 95 18 95 18 95 4A

RX:  01 01 03 00 52 A9
```

If the line is programmed successfully an ACK is returned together with a **status byte**.
A NAK means that the communication with the tube motors was not successful.

The **status byte** has the following meaning:

Status Byte	Meaning
0x52	Page program Ok
0x53	Page program Error
0x54	Valid Application
0x55	No Valid Application

Table 18: Status byte in response of command 0x8c

In our case 0x52 is returned, meaning the PAGE is programmed OK.
If the page was not programmed successfully, a PAGE_ERROR (0x55) is returned.

3.7.3 Command 138: Finish Firmware Upgrade

Command 138 (8A Hex) terminates the boot procedure, checks if the checksum of the application is valid and starts the application firmware.

If the application is valid, an ACK is returned, otherwise a NAK.

Transmitted data: <138>, <none>
Received data: <ACK> or <NAK>

After upgrading the firmware, the host machine should check if the Eeprom has been re-initialized, using Request Error Status command 122. Major firmware updates could result in an Eeprom initialization when necessary. This is automatically done by the ICS firmware. After that, the host machine should verify if all settings are still correct.

Example: Finish Tube motor FW Upgrade

```
TX: 03 00 01 8A 72
RX: 01 00 03 00 FC
```

If the application is valid, an ACK is returned, otherwise a NAK.

4 ICS Setup

The ICS can operate in 3 different system modes.

1. ICS only
2. ICS with a Coin Acceptor with a 2-way sorter attached.
3. ICS with a coin Acceptor and 2 Escrow units attached.

4.1 ICS mode command

The ICS system mode can be set using the SET_MODE (93 = 0x5D) command.
The following example sets the ICS in the mode ICS + Coin Acceptor and retrieves the current mode:

```
Tx -> 03 02 01 46 5D 01 56
Rx <- 01 00 03 00 FC
Set System Mode: 1

Tx -> 03 01 01 46 5C 59
Rx <- 01 01 03 00 01 FA
Get System Mode: 1
```

4.2 ICS and Coin Acceptor with a Sorter

The Coin Acceptor (CA) can be connected to the cctalk peripheral extension bus of the ICS.
This has the advantage that the ICS can control the sorter attached to the Coin Acceptor. If the ICS wants to reject a certain coin type, because the ICS has enough coins, it can control the CA directly and route the coin to the cashbox instead of to the ICS, after acceptance by the CA. The Host machine talks with the CA over the normal cctalk bus. All CA commands are send to the peripheral extension bus by the ICS.

If the user wants full control over the Coin Acceptor without control from the ICS, the user should set the ICS mode to 1 (ICS only) and connected the Coin Acceptor directly on the same cctalk bus as the ICS.

4.3 ICS and Coin Acceptor with 2 Escrow Units attached

If the user wants an Escrow function installed, the sorter of the CA is replaced by 2 Escrow units: one escrow to hold the accepted coins for customer return if the customer presses the return button, and one escrow unit to direct all coins to the cashbox if the ICS rejects one of the accepted coins in the first escrow unit. These escrow units are connected and controlled via the same connector as the cctalk peripheral extension bus.

4.4 System Initialization

First of all the ICS mode should be set to the required operation mode Hopper Only, Hopper + Coin Acceptor + Sorter or Hopper + escrow units attached.

4.4.1 Hopper Only Mode

The default mode is Hopper Only mode in which the ICS is filled with coins manually and dispenses coins or an amount of money on command.
After power on or reset, a payout of coins (command 20) or amount (command 22) immediately starts the payout of coins. No enabling or security commands are necessary. The only requirement for a successful payout is that the ICS and tubes are filled with coins.
After filling the ICS for the first time, Total In (command 71) values must be updated and all tubes will be filled with (maximum 4) coins at the first sorting run. Then a payout can be started. After a tube has paid one or more coins, the ICS will start sorting again automatically and the tubes will be filled to their maximum levels as good as possible. If a payout is not possible (NAK is returned at the payout command), the ICS will start sorting and filling the tubes as much as possible. The host may retrieve the sorting status using command 55. If the ICS pays a coin, the number of coins inside the ICS is updated.

Tubes that are not used (for example the 1ct and 2ct), should have their full level setting parameter to 0 or should be disabled by using the TUBE_DISABLE command.

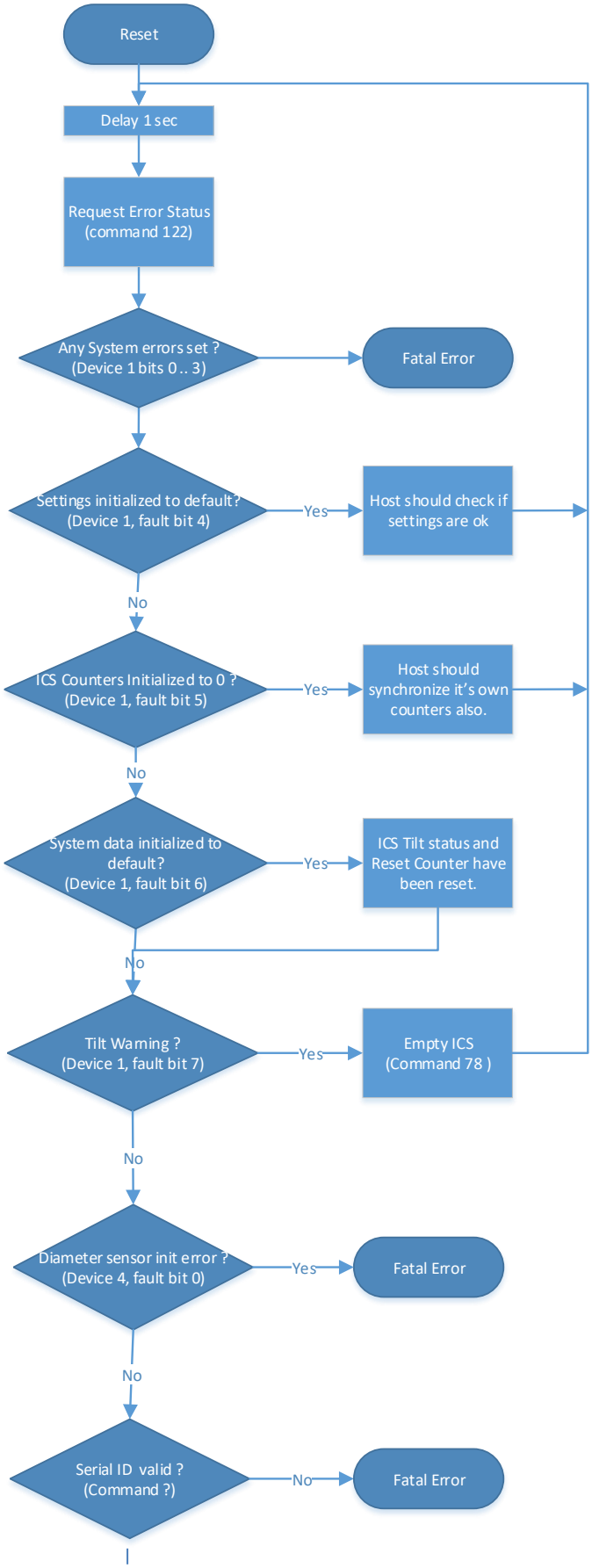
4.4.2 Hopper + Coin Acceptor with attached sorter

A coin acceptor with sorter attached can be connected to the ICS controlled cctalk extension line. The coin acceptor is not connected directly to the same cctalk bus as the ICS but to another cctalk bus line coming from the ICS. All commands received from the normal cctalk bus connected to the ICS will be forwarded to the internal cctalk bus extension that is connected to the coin acceptor with sorter. All responses from the coin acceptor will be forwarded to the normal cctalk bus from the ICS also. So, the ICS stands in between the coin acceptor with sorter and the normal cctalk bus. In this way the ICS can transmit enable/disable coin acceptor commands when the ICS becomes full or not and set the sorter outputs to drive the accepted coins into the ICS or Cashbox if necessary.
Also the ICS will update its coin in total counters when coins are received by the coin acceptor.

4.4.3 Hopper + Coin Acceptor + Escrow Sorter Units

In this configuration the accepted coins from the coin acceptor are hold in an escrow module before they are released to either the ICS or Cashbox. The 2 sorter modules (one for escrow and one for sorting) are controlled via the ICS.

4.5 Initialization Flowchart



System Initialisation

After system power up or reset, wait at least 1 sec for the ICS to initialize. After this delay the ICS is able to receive and process any commands.
Then the ICS will check if all mechanics are ok, and if not, will start to unblock any blocking mechanics, etc. After 15 sec this process has usually ended and the user may Request the error status to check if there are any mechanical errors or warnings.

If any system error bits are set, the ICS needs to be serviced by a technician.

After a major software update or a major hardware or software crash, the eeprom settings may have been corrupted. The ICS will detect this automatically and re-initialize all settings to their default values.

If the dynamic data (ICS counters, etc) are initialized to 0, the host machine should account for this also.

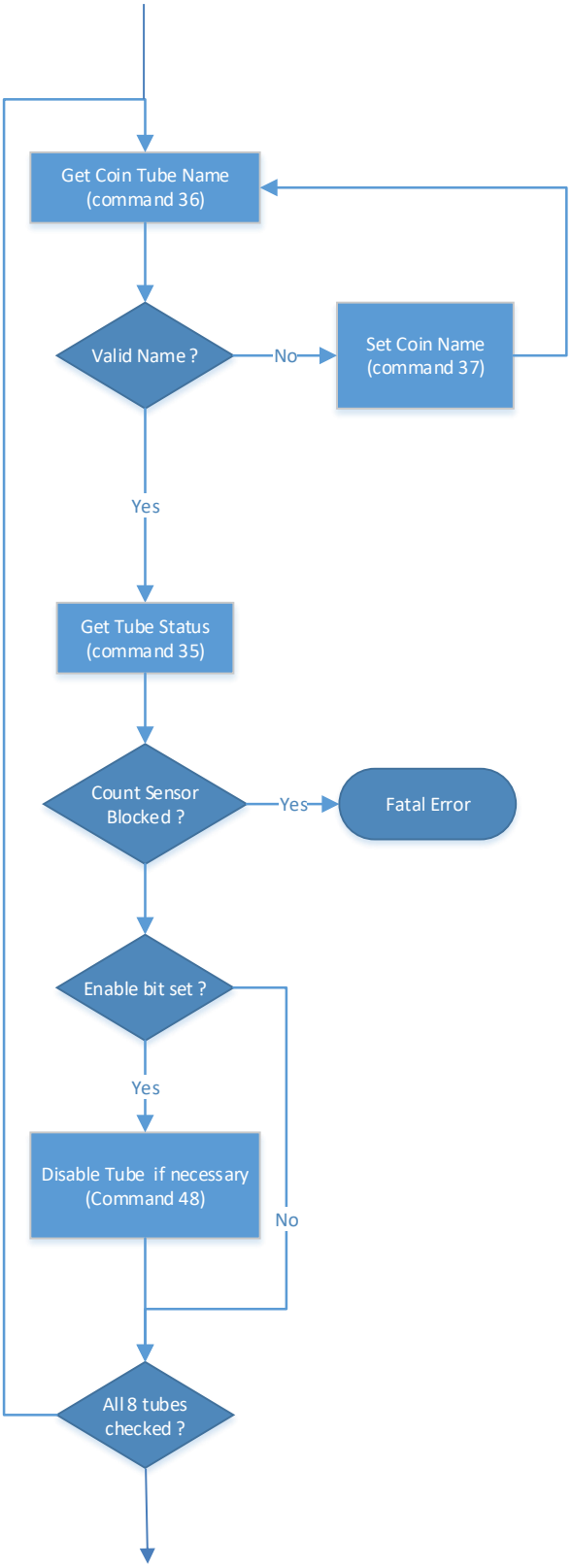
The eeprom system data is a small data structure holding system related items, like the nr system resets, tilt status, etc

If the ICS tilt Warning bit is set, then the ICS has been held upside down while the tubes were not empty. This warning is cleared by totally emptying the ICS followed by a system reset.

If the diameter sensor fails to initialize, then the ICS may not sort correctly. A technical inspection is necessary.

If the serial number is not programmed by the factory, the diameter sensor is most probably also not programmed. A technician needs to service the ICS.

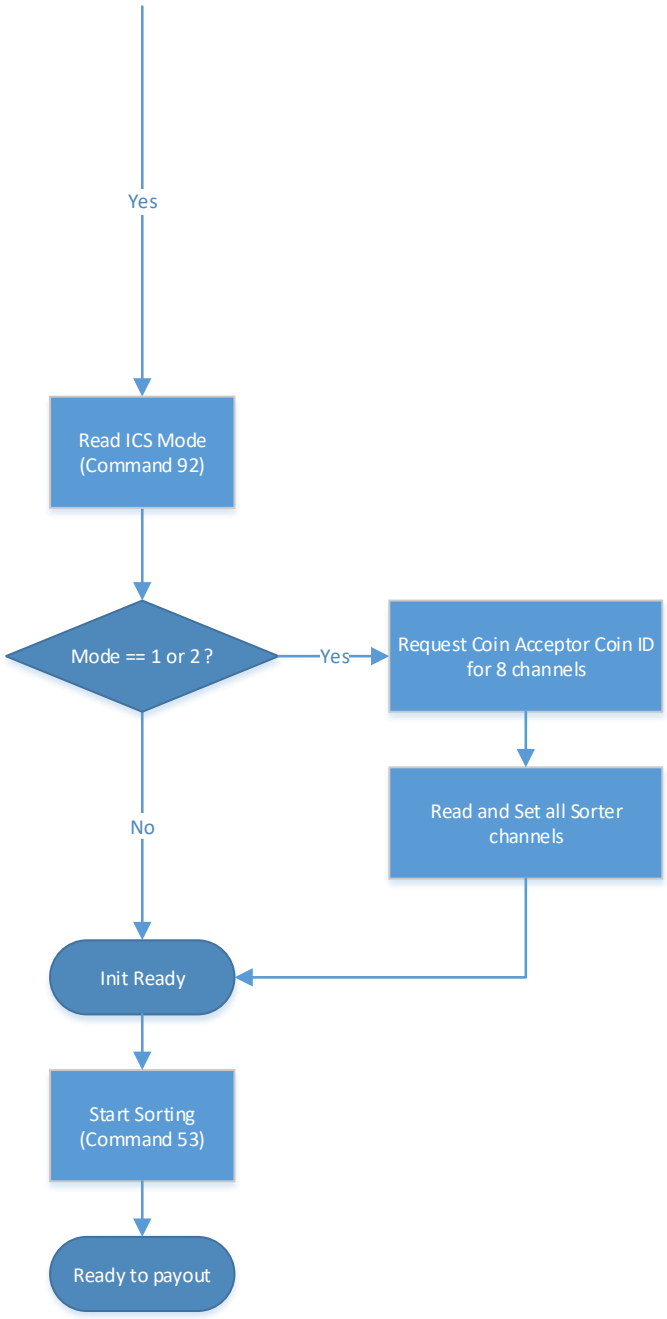
Tubes Initialisation



The Coin Tube Name format is for example EU010A
Note that the value of the coin is extracted from the name. Here the value is 10.
So it is important to set the names properly, if the values are wrong then the changing may become wrong also.

If the count sensor of a tube is interrupted, then the ICS stops operating in order to prevent damage due to tube coin overflow during the sorting process. Note that this flag is also set when 6 coins have been detected by the coin diameter sensor, but have not been detected by the tube counter sensor. (Large coin may be blocking the tube)

After a reset all tubes are standard enabled. If a tube becomes in error or the Tube fullLevelSetting is 0, it will be disabled by the ICS automatically. If the user wants to disable a tube manually use the Tube Disable command 48.



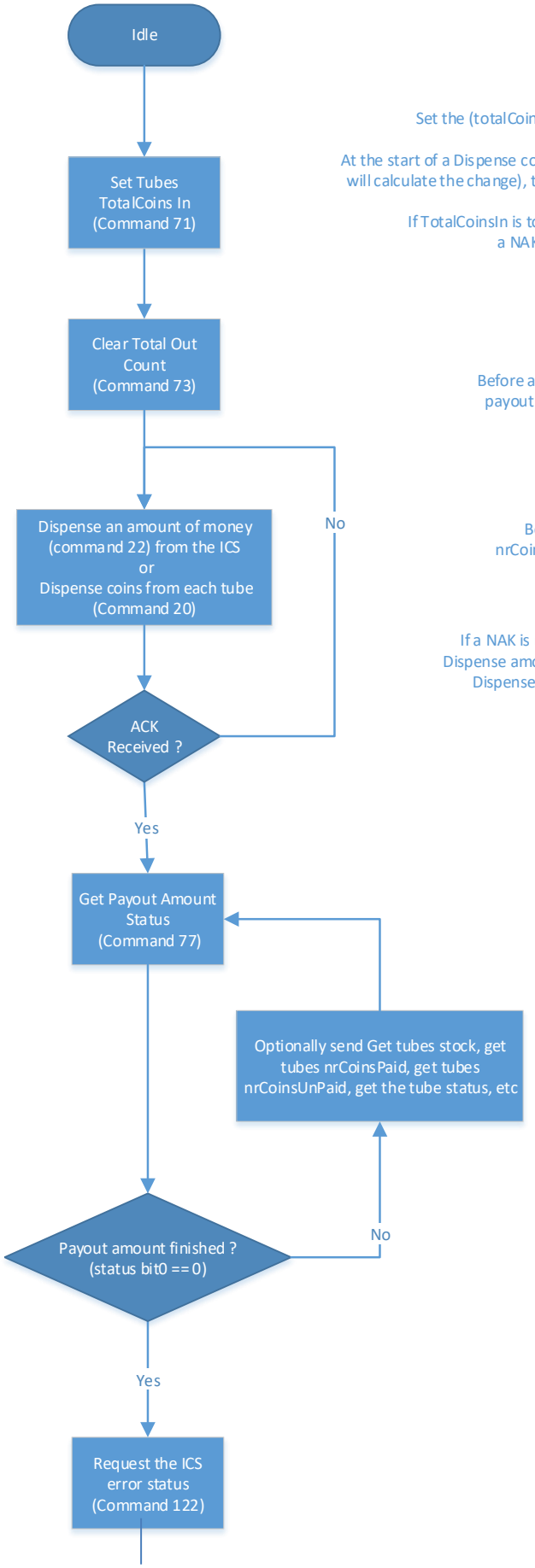
Mode 1 and 2 are not supported yet.

If the ICS mode is equal to 1 or 2, then a Coin Acceptor and Sorter must be connected to the ICS cctalk extension bus

If the ICS is started after being filled, then the Start Sorting command should be given by the host in order to prevent the first payout being NAK'ed because of empty tubes.

Note that the Sorting is not automatically started after a power up or reset. It only starts automatically after an unsuccessful payout.

4.6 Dispense Coins or Amount Flowchart



Set the (totalCoinsIn) number of coins inside the ICS for each denomination.

At the start of a Dispense coins (command 20) or Dispense payout amount (command 22, the ICS will calculate the change), the ICS checks if there are enough coins inside the ICS to fill the tubes during the (Floating) Payout process.
If TotalCoinsIn is too low to payout the coins (not enough coins inside the ICS), a NAK is returned at the Dispense amount command.

Before a payout you can clear the total out counter. After the payout it can be used to check if the requested amount has been paid.

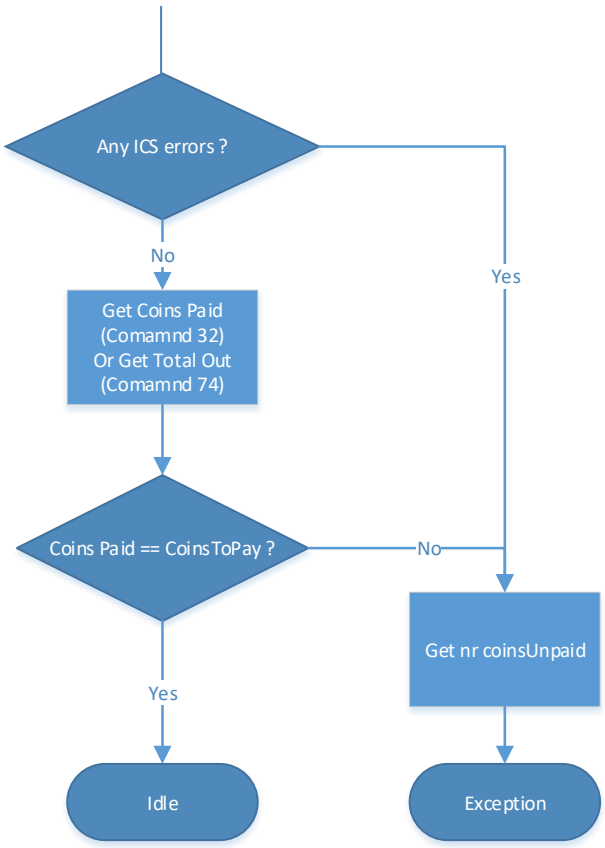
Before the payout starts, each tube sets it's nrCoinsPaid to 0 and the number of unPaidCoins to the number of coins to pay.

If a NAK is received, then the requested amount of coins could not be paid.
Dispense amount (command 22): not enough coins inside the ICS (bowl + tubes)
Dispense coins (command 20): not enough coins available in the tubes..

Get the payout amount status byte to monitor the payout process.

During the payout process the user may of coarse send other commands then command 77 to monitor the payout process.
The tube coin counters and stock are updated each time a coin is paid.
Retrieving these values gives you a live picture of the payout process.

During the payout process bit0 will be set. If it becomes 0, the payout process has stopped.
Check if bit1 indicates if the payout process ended successfully.
If not, check bits 2 – 7 to see the reason why the payout process was aborted.



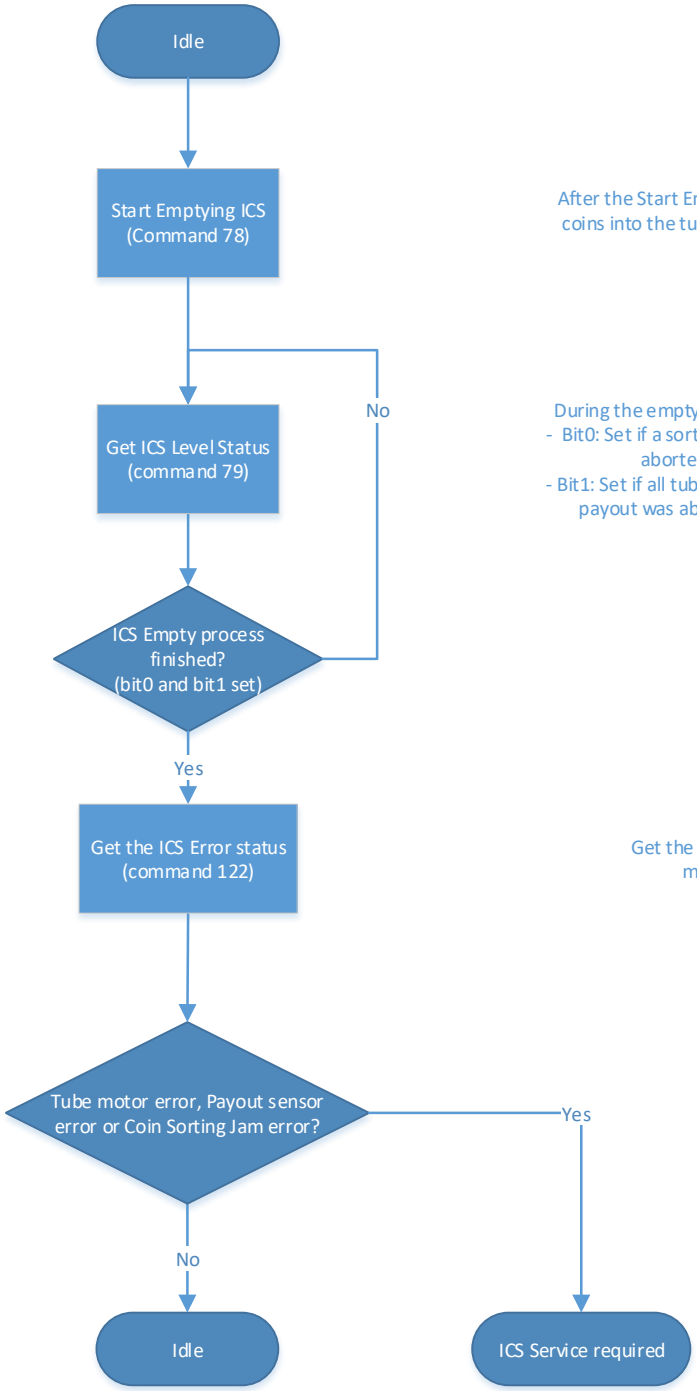
If there are any errors, then get the nr coinsUnPaid
If there are errors (ICS Led blinks red) then service is required. If there are warnings (ICS blinks green) the user may continue using the ICS.*

If there are no errors, to users may check if the requested amount of coins to be paid is really paid.

The Host Application handles the exceptions.
An action could be to send a Reset command to the ICS and send additional payout commands to finish the payout of unPaid coins if possible.

* When receiving an error use Command 35 to inquire more information on individual tubes

4.7 ICS Emptying Flowchart



After the Start Empty ICS command, the ICS starts sorting the all coins into the tubes. When a tube stock becomes larger than 9, the tube dispenses 8 coins.

During the emptying, send command 79 to get the level status.
- Bit0: Set if a sorting timeout occurs (bowl empty or sorting was aborted due to some error (tube overfull).
- Bit1: Set if all tubes generated a payout timeout (tube empty or payout was aborted due to some tube error (motor jam).

Get the ICS error status. Check if there are no Tube motor jams or Sorting errors occurred.

If the ICS signals a Tube motor error, then the emptying process is aborted, You could send an ICS Reset command and try the Empty process again.
Coin sorting error means usually that more than 1 tube has become overfull with coins and sorting is not allowed anymore and the emptying process is also aborted.

4.8 ICS Floating Command Sequence Example

ICS Start situation in the example: All tubes are empty, 10 coins of each denomination in ICS.

Reset ICS(Command 0x01)(Not mandatory, only necessary to clear any errors before starting a payout.

```
0 : 8-5-2018 10:22:35 -> 03 00 01 01 FB
1 : 8-5-2018 10:22:35 <- 01 00 03 00 FC
2 : Cmd: Reset
```

Read all tube coin types(ICS command 0x24) (Not mandatory, only needed to inform host about the tube valuesfor the first time).

```
3 : 8-5-2018 10:22:42 -> 03 02 01 46 24 00 90
4 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 30 30 31 41 8A
5 : Get tube name[0] Ans: EU001A

6 : 8-5-2018 10:22:42 -> 03 02 01 46 24 01 8F
7 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 30 30 32 41 89
8 : Get tube name[1] Ans: EU002A

9 : 8-5-2018 10:22:42 -> 03 02 01 46 24 02 8E
10 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 30 31 30 41 8A
11 : Get tube name[2] Ans: EU010A

12 : 8-5-2018 10:22:42 -> 03 02 01 46 24 03 8D
13 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 30 30 35 41 86
14 : Get tube name[3] Ans: EU005A

15 : 8-5-2018 10:22:42 -> 03 02 01 46 24 04 8C
16 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 30 32 30 41 89
17 : Get tube name[4] Ans: EU020A

18 : 8-5-2018 10:22:42 -> 03 02 01 46 24 05 8B
19 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 31 30 30 41 8A
20 : Get tube name[5] Ans: EU100A

21 : 8-5-2018 10:22:42 -> 03 02 01 46 24 06 8A
22 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 30 35 30 41 86
23 : Get tube name[6] Ans: EU050A

24 : 8-5-2018 10:22:42 -> 03 02 01 46 24 07 89
25 : 8-5-2018 10:22:42 <- 01 06 03 00 45 55 32 30 30 41 89
26 : Get tube name[7] Ans: EU200A
```

Step2: Enable all tubes (if necessary, only needed one time)

```
27 : 8-5-2018 10:22:45 -> 03 02 01 46 2F 00 85
28 : 8-5-2018 10:22:45 <- 01 00 03 00 FC
29 : Enable Tube 0

30 : 8-5-2018 10:22:45 -> 03 02 01 46 2F 01 84
31 : 8-5-2018 10:22:46 <- 01 00 03 00 FC
32 : Enable Tube 1

33 : 8-5-2018 10:22:46 -> 03 02 01 46 2F 02 83
34 : 8-5-2018 10:22:46 <- 01 00 03 00 FC
35 : Enable Tube 2

36 : 8-5-2018 10:22:46 -> 03 02 01 46 2F 03 82
37 : 8-5-2018 10:22:46 <- 01 00 03 00 FC
38 : Enable Tube 3

39 : 8-5-2018 10:22:46 -> 03 02 01 46 2F 04 81
40 : 8-5-2018 10:22:46 <- 01 00 03 00 FC
41 : Enable Tube 4

42 : 8-5-2018 10:22:46 -> 03 02 01 46 2F 05 80
43 : 8-5-2018 10:22:47 <- 01 00 03 00 FC
44 : Enable Tube 5

45 : 8-5-2018 10:22:47 -> 03 02 01 46 2F 06 7F
46 : 8-5-2018 10:22:47 <- 01 00 03 00 FC
47 : Enable Tube 6

48 : 8-5-2018 10:22:47 -> 03 02 01 46 2F 07 7E
49 : 8-5-2018 10:22:47 <- 01 00 03 00 FC
50 : Enable Tube 7
```

Get Tubes status (not mandatory)

```
51 : 8-5-2018 10:22:47 -> 03 01 01 46 23 92
52 : 8-5-2018 10:22:47 <- 01 08 03 00 21 21 21 21 21 21 21 21 EC
53 : Get Tube Status: 21 21 21 21 21 21 21 21
```

Step3: Set Total Coins In of each denomination (ICS command 0x47)
Mandatory if not already done so

```
54 : 8-5-2018 10:23:12 -> 03 11 01 46 47 64 00 64 00 64 00 64 00 64 00 64
00 64 00 64 00 3E
55 : 8-5-2018 10:23:12 <- 01 00 03 00 FC
56 : Set ICS Total Coins In : 100 100 100 100 100 100 100 100
```

Check Total Coins In of each denomination(not mandatory)

```
57 : 8-5-2018 10:23:12 -> 03 01 01 46 48 6D
58 : 8-5-2018 10:23:12 <- 01 10 03 00 64 00 64 00 64 00 64 00 64 00 64 00
64 00 64 00 CC
59 : Get ICS Total Coins In : 100 100 100 100 100 100 100 100
```

Get Total Amount In(not mandatory)

```
60 : 8-5-2018 10:23:12 -> 03 01 01 46 4B 6A
61 : 8-5-2018 10:23:12 <- 01 04 03 00 90 97 00 00 D1
62 : Get ICS Total Amount In : 388,00
```

Step4: Dispense coins (ICS command 0x14)

```
63 : 8-5-2018 10:23:34 -> 03 09 01 46 14 01 01 01 01 01 01 01 01 91
64 : 8-5-2018 10:23:34 <- 01 01 03 00 01 FA
65 : Dispense coins: [0]1 [1]1 [2]1 [3]1 [4]1 [5]1 [6]1 [7]1
```

ICS Starts sorting to fill the empty tubes. When there are enough coins in the tube (min 3), a coin will be paid.

Get payout status progress(not mandatory)

```
66 : 8-5-2018 10:23:35 -> 03 01 01 46 4D 68
67 : 8-5-2018 10:23:35 <- 01 03 03 00 01 84 01 73
68 : Get Status Last Payout - Status: 1 UnPaid: 3,88

69 : 8-5-2018 10:23:37 -> 03 01 01 46 4D 68
70 : 8-5-2018 10:23:37 <- 01 03 03 00 01 84 01 73
71 : Get Status Last Payout - Status: 1 UnPaid: 3,88

72 : 8-5-2018 10:23:38 -> 03 01 01 46 4D 68
73 : 8-5-2018 10:23:39 <- 01 03 03 00 01 84 01 73
74 : Get Status Last Payout - Status: 1 UnPaid: 3,88

75 : 8-5-2018 10:23:40 -> 03 01 01 46 4D 68
76 : 8-5-2018 10:23:40 <- 01 03 03 00 01 84 01 73
77 : Get Status Last Payout - Status: 1 UnPaid: 3,88

78 : 8-5-2018 10:23:42 -> 03 01 01 46 4D 68
79 : 8-5-2018 10:23:42 <- 01 03 03 00 01 84 01 73
80 : Get Status Last Payout - Status: 1 UnPaid: 3,88

81 : 8-5-2018 10:23:44 -> 03 01 01 46 4D 68
82 : 8-5-2018 10:23:44 <- 01 03 03 00 01 84 01 73
83 : Get Status Last Payout - Status: 1 UnPaid: 3,88

84 : 8-5-2018 10:23:45 -> 03 01 01 46 4D 68
85 : 8-5-2018 10:23:45 <- 01 03 03 00 01 79 01 7E
86 : Get Status Last Payout - Status: 1 UnPaid: 3,77

87 : 8-5-2018 10:23:47 -> 03 01 01 46 4D 68
88 : 8-5-2018 10:23:47 <- 01 03 03 00 01 47 01 B0
89 : Get Status Last Payout - Status: 1 UnPaid: 3,27

90 : 8-5-2018 10:23:49 -> 03 01 01 46 4D 68
91 : 8-5-2018 10:23:49 <- 01 03 03 00 01 47 01 B0
92 : Get Status Last Payout - Status: 1 UnPaid: 3,27

93 : 8-5-2018 10:23:52 -> 03 01 01 46 4D 68
94 : 8-5-2018 10:23:52 <- 01 03 03 00 01 2E 01 C9
95 : Get Status Last Payout - Status: 1 UnPaid: 3,02

96 : 8-5-2018 10:23:54 -> 03 01 01 46 4D 68
97 : 8-5-2018 10:23:54 <- 01 03 03 00 01 2E 01 C9
98 : Get Status Last Payout - Status: 1 UnPaid: 3,02

99 : 8-5-2018 10:23:57 -> 03 01 01 46 4D 68
100 : 8-5-2018 10:23:57 <- 01 03 03 00 01 2E 01 C9
101 : Get Status Last Payout - Status: 1 UnPaid: 3,02

102 : 8-5-2018 10:23:59 -> 03 01 01 46 4D 68
103 : 8-5-2018 10:23:59 <- 01 03 03 00 01 CA 00 2E
104 : Get Status Last Payout - Status: 1 UnPaid: 2,02
```

```
105 : 8-5-2018 10:24:02 -> 03 01 01 46 4D 68
106 : 8-5-2018 10:24:02 <- 01 03 03 00 01 CA 00 2E
107 : Get Status Last Payout - Status: 1 UnPaid: 2,02

108 : 8-5-2018 10:24:05 -> 03 01 01 46 4D 68
109 : 8-5-2018 10:24:05 <- 01 03 03 00 01 02 00 F6
110 : Get Status Last Payout - Status: 1 UnPaid: 0,02

111 : 8-5-2018 10:24:13 -> 03 01 01 46 4D 68
112 : 8-5-2018 10:24:13 <- 01 03 03 00 02 00 00 F7
113 : Get Status Last Payout - Status: 2 UnPaid: 0,00 (Payout has finished)

114 : 8-5-2018 10:24:18 -> 03 01 01 46 4D 68
115 : 8-5-2018 10:24:18 <- 01 03 03 00 02 00 00 F7
116 : Get Status Last Payout - Status: 2 UnPaid: 0,00
```

Check coins paid (not mandatory)

```
117 : 8-5-2018 10:24:30 -> 03 01 01 46 20 95
118 : 8-5-2018 10:24:30 <- 01 08 03 00 01 01 01 01 01 01 01 01 EC
119 : Get Coins Paid : 1 1 1 1 1 1 1 1
```

Check coins unPaid (not mandatory, but strongly recommended)

```
120 : 8-5-2018 10:24:31 -> 03 01 01 46 21 94
121 : 8-5-2018 10:24:31 <- 01 08 03 00 00 00 00 00 00 00 00 00 F4
122 : Get Coins UnPaid: 0 0 0 0 0 0 0 0
```

Check Total Coins Paid counter (not mandatory)

```
123 : 8-5-2018 10:24:36 -> 03 01 01 46 4A 6B
124 : 8-5-2018 10:24:37 <- 01 20 03 00 92 00 00 00 B5 00 00 00 9C 00 00 00
9C 00 00 00 B1 00 00 00 9A 00 00 00 A1 00 00 00 C1 00 00 00 B0
125 : Get ICS Total Coins Out : 146 181 156 156 177 154 161 193
```

5 Connector

The ICS unit has 2 interface connectors. These are electrically the same, only the location is different.
The connector is a Molex Micro-Fit 3.0 Series 44428 female type 44428-1201.
These type of connectors allow easy ICS-slide in constructions.
Tabel 18 lists the pin descriptions: (MDB interface pins are reserved for future use).

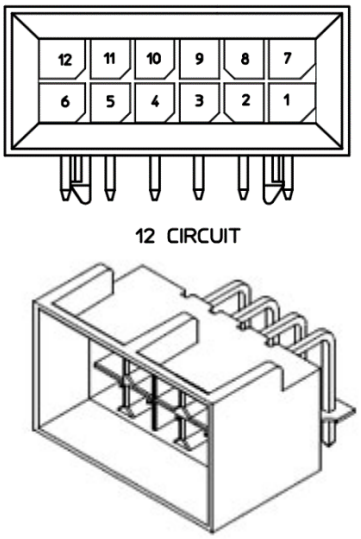


Figure 1: ICS Connector Molex 44428-1201

Pin	Description
1	CCTalk data
2	Reserved for CCTalk bus extension
3	Prog/Debug TxD
4	Input Line 1
5	Prog/Debug RxD
6	Input Line 2
7	Reserved for sorter 1 control
8	Reserved for sorter 2 control
9	Vin (12-24 Vdc)
10	Vin (12-24Vdc)
11	GND
12	GND

Table 19: Connector pin description

The male counterpart can be any of the type below:

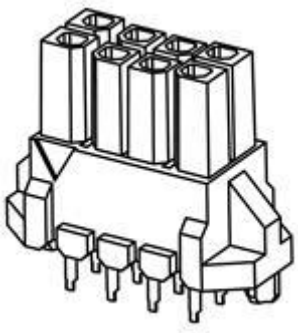


Figure 2: Molex 44769-1201

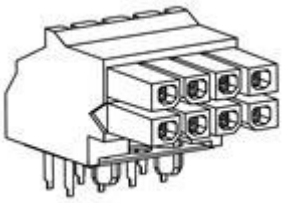


Figure 3: Molex 44764-1201

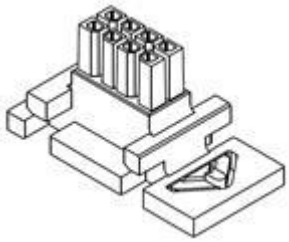


Figure 4: Molex 44133-1200

6 Specifications

6.1 General

Coin types:

Diameter :	15 – 32,5 mm
Thickness:	1,3 – 3,25 mm
Denominations:	8 denominations in one ICS
Coin bulk:	Unsorted coins in ideal coin mix
	≈ 1300 coins (depending on coin sizes)

Payout system:

- 4 coins simultaneous payout per cycle
  (max 4 coins per second)
- Point of no return sensor per denomination
- Payout combination by host or ICS
  (for ideal mix control)
    - Coin set controlled by:
        - Direct host machine control (low-level)
        - Intelligent payout control by ICS
- ≈ 30 coins buffer for each denomination (depending on coin thickness)

Maintenance:

Status indicator

- RGB status LED

Security:

Closed system:

- Coin sorting and payout integrated
    - 2 locks k.d. (4 eye principle)
    - RVS top plate (customizable top surface for easy machine – ICS integration)

6.2 Integration

General:		
Connector	Self-centering on long and short side of ICS	
Dimensions	248 x 198 x 370	mm
Protocols:	ccTalk	Serial
Bookkeeping:	Coins in/out + cashbox	
	Point of no return sensor per denomination	
Power:		
Voltage:	12 or 24 (± 10%)	VDC
Advised power supply:	5	A
Environmental:		
Storage temperature	-20º C to +70º C	
Operating temperature	0º C to +50ºC	
Storage humidity	10% to 95% non-condensing	
Operating humidity	10% to 75% RH	

6.3 Filling

To fill the ICS box, unlock both cam locks and remove the coin lid. Insert the coins mixed (according to the ideal coin mix ratio) beforehand to get the best initial coin tube sorting time. **Make sure no debris and foreign objects are inserted during filling the ICS** this can cause malfunctioning of the ICS.

See Chapter 1.1 The ICS Coin Bulk Filling Process for coin amount details.

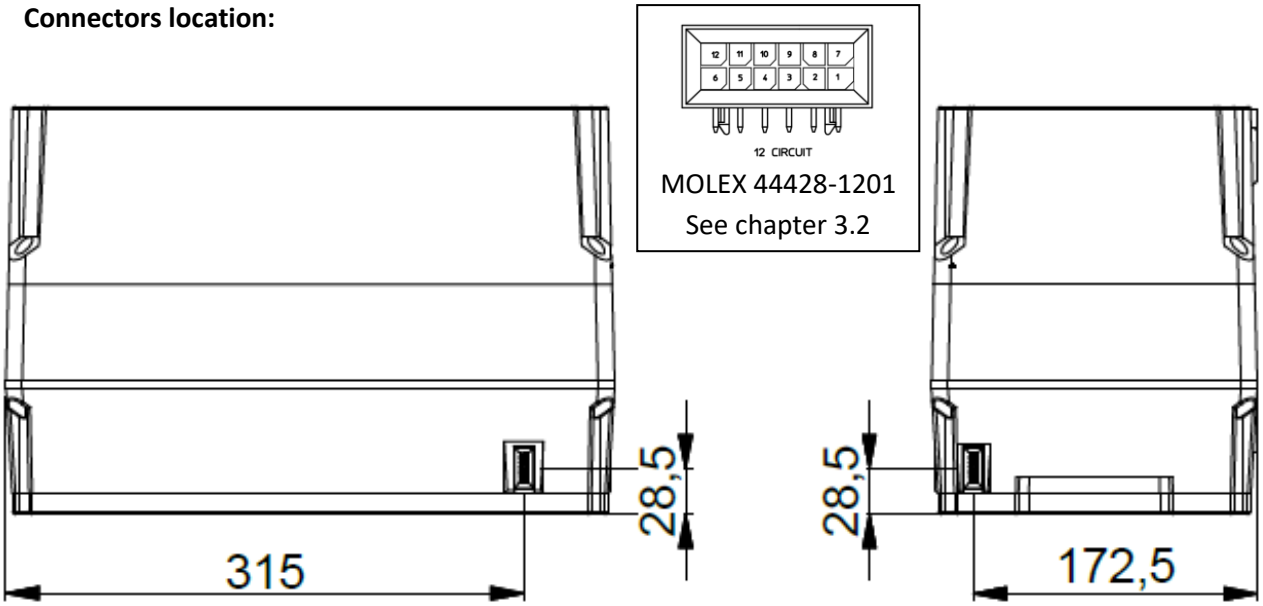
6.4 Handling & Transport

Handling the ICS when the machine is filled with coins must be done with great care. **The ICS cannot have any coins sorted before transporting.** When handling the filled ICS, make sure to not place it under an extreme angle (more than 15°) to prevent coins entering the coin tubes unregulated. If the ICS has been under an extreme angle, it will report an error code to indicate the event.

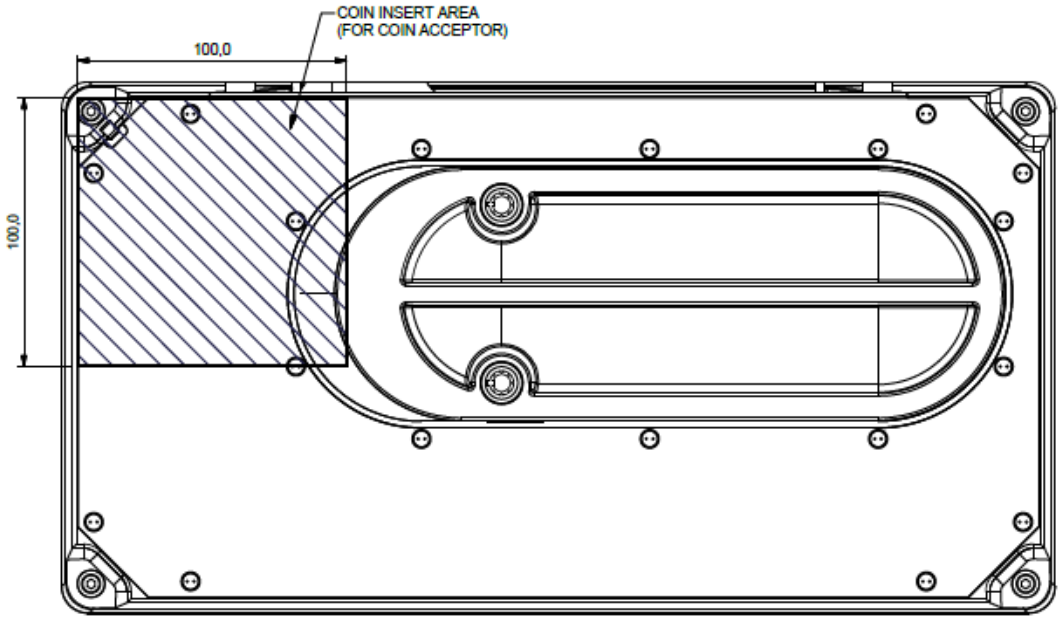
7 Dimensions

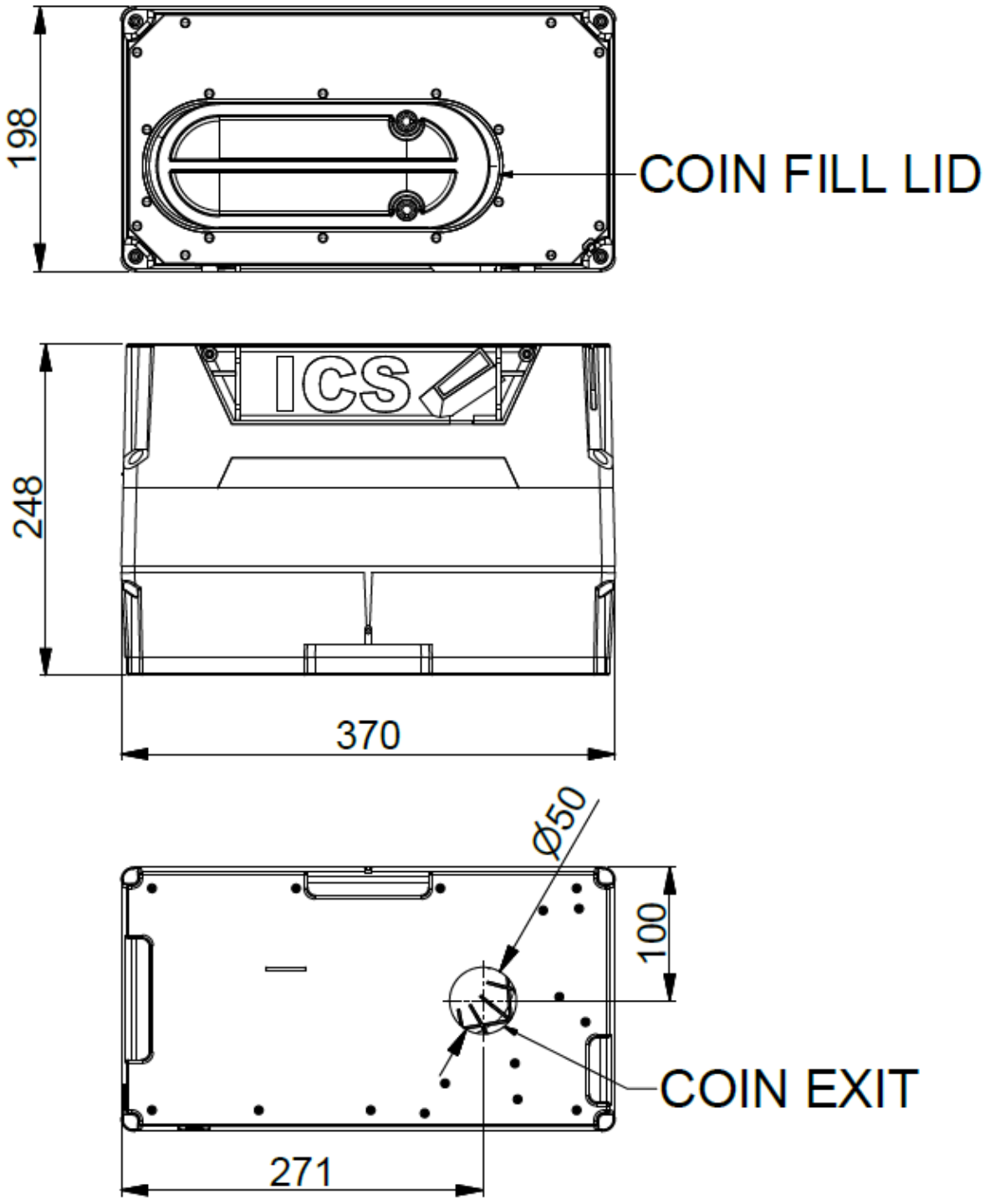
The ICS is designed to be implemented with a custom slide system (see chapter “optional add-ons”). The ICS has two connectors, in the length and width of the machine. The ICS is able to be mounted in both length and side wise. This makes mounting the ICS customizable to your own choice.

Connectors location:



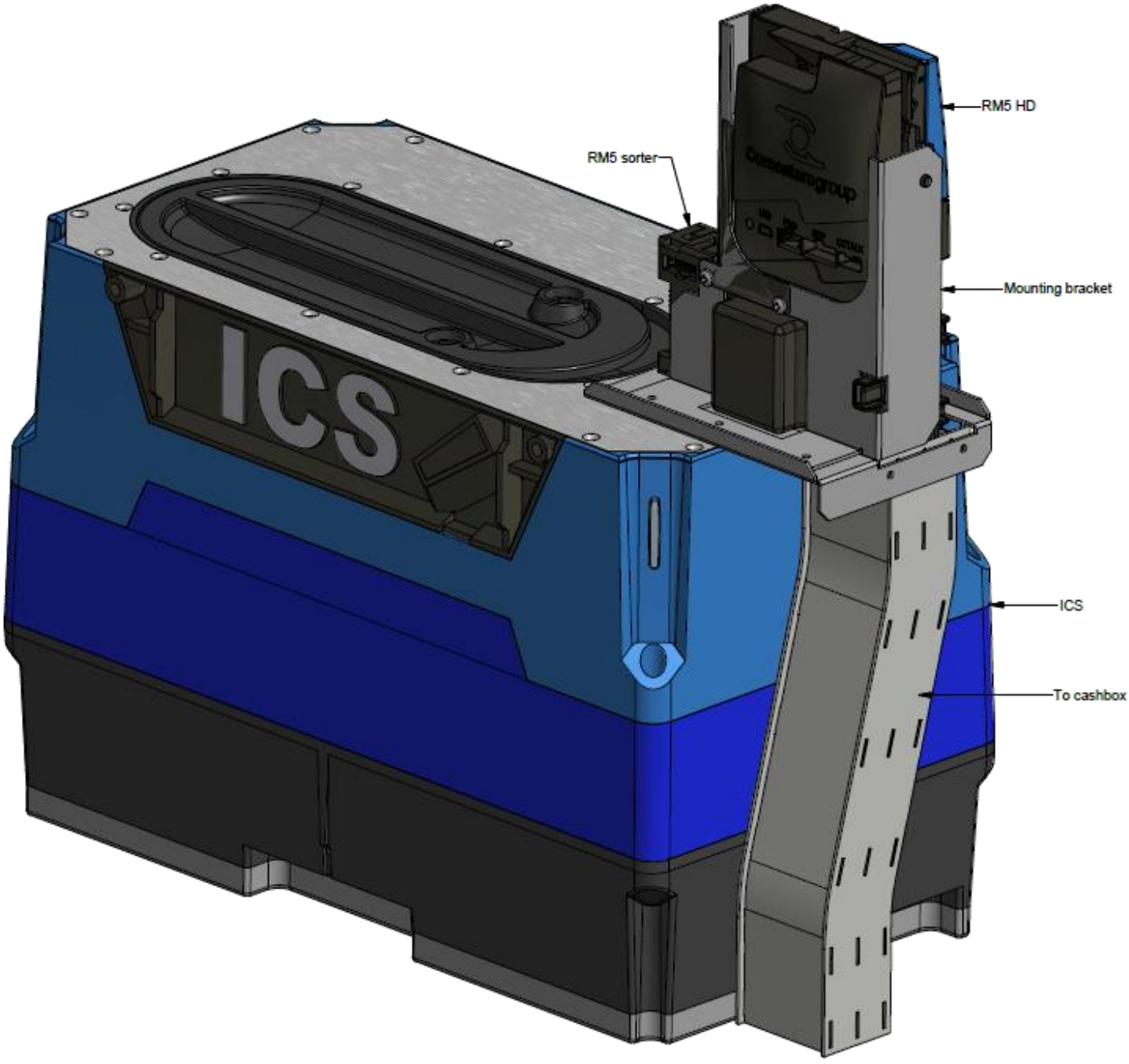
Coin acceptor coin input location:



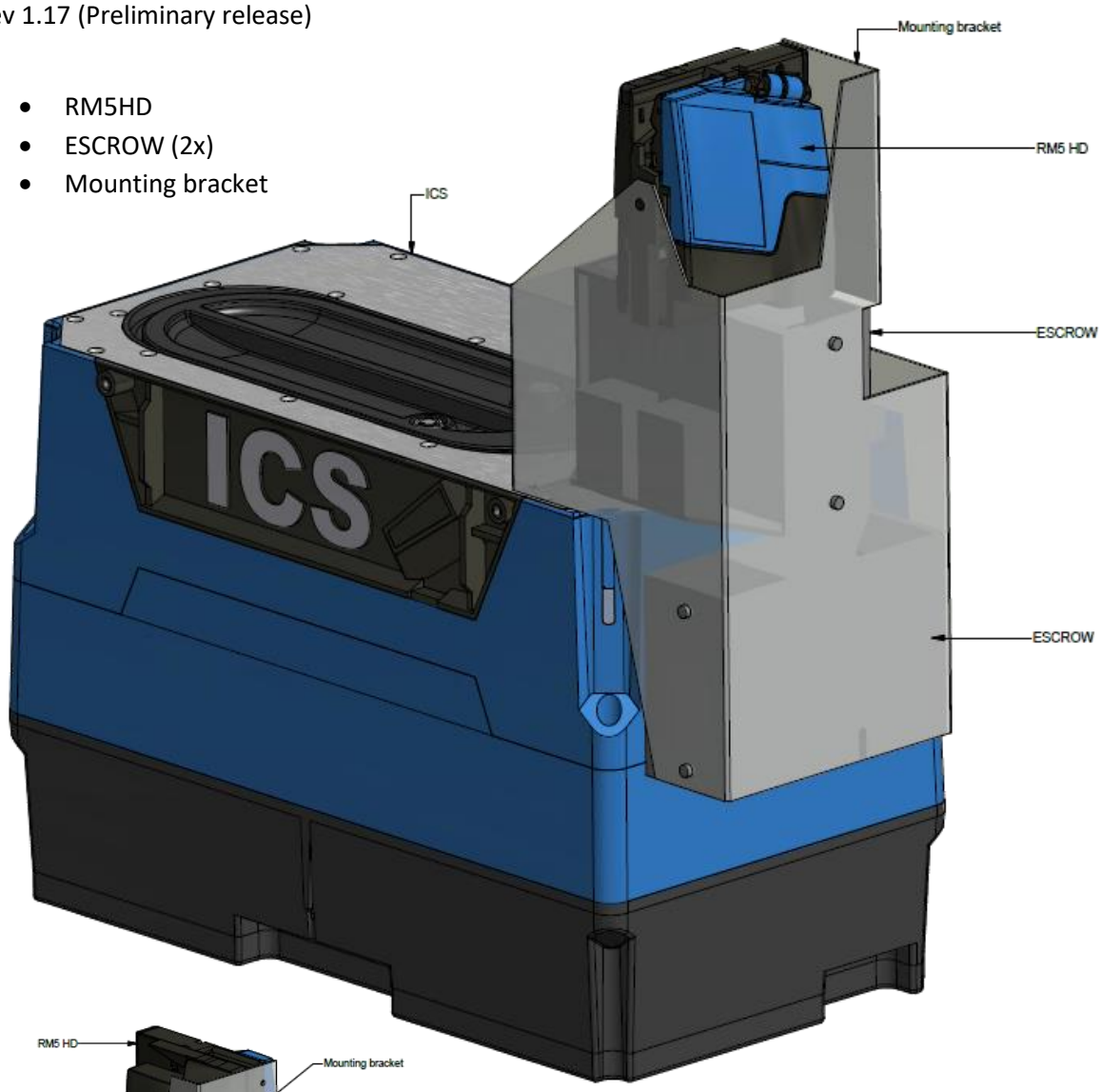

198
COIN FILL LID
ICS
248
370
Ø50
100
COIN EXIT
271

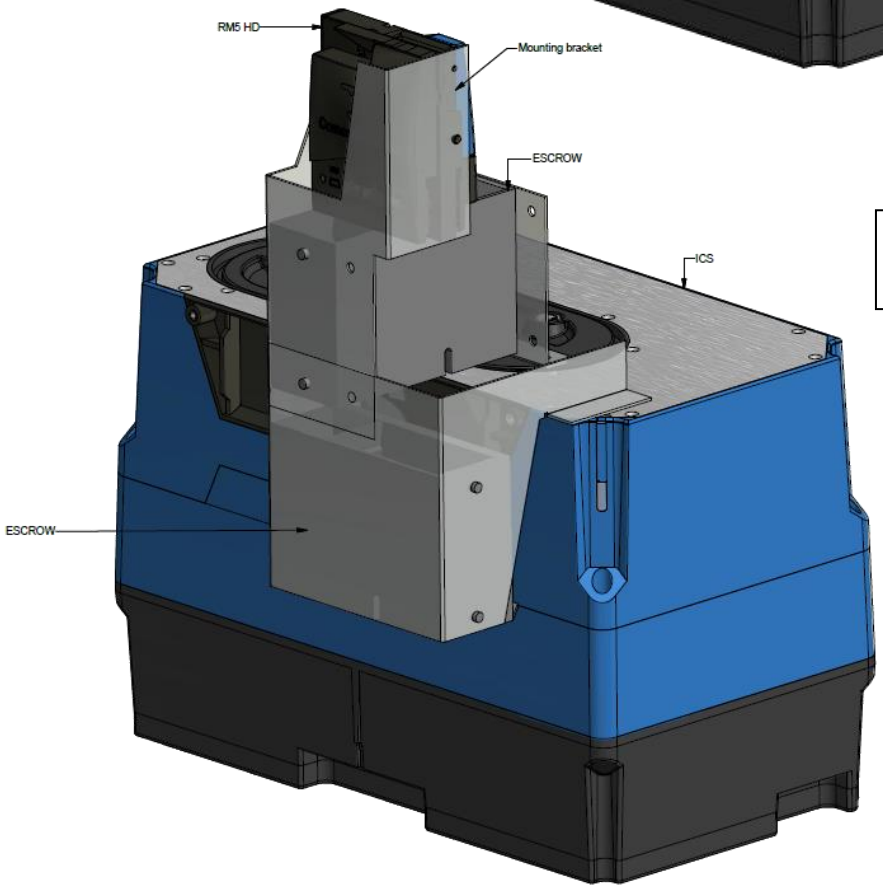
8 Optional Add-Ons

- RM5 HD
- RM5 Sorter
- Mounting bracket



- RM5HD
- ESCROW (2x)
- Mounting bracket





Possible integration of the ICS + RM5 HD + two ESCROW units

Revision history

Revision	Date	Comment	By
1.0	20-3-2017	New release	R.T.
1.1	29-3-2017	1. Coin handling updated 2. Sorting start and stop commands deleted	R.T.
1.2	18-5-2017	1. Sorting start and stop criterium updated. 2. ICS Get Level Command added. 3. SET_TOTAL_COINS_IN command updated. (not related anymore to best set best mix total coins in). 4. Disabling a tube using command 43 added.	R.T.
1.3	30-6-2017	1. CLEAR_TUBE_STATUS command deleted. 2. Tube motor Jam status added. 3. ICS Tilt Sensor function added. 4. Startup sequence more detailed. 5. Tube GET_STATUS response is modified.	R.T.
1.4	24-8-2017	1. Error status bits updated. 2. Flow charts added. 3. TUBE_GET_LEVEL_STATUS response changed.	R.T.
1.5	22-10-2017	1. ICS Error Status table updated and REQUEST_ERROR_STATUS command added. 2. GET_TUBE_LEVEL_COMMAND added. 3. Flow diagrams updated. 4. START/STOP Sorting control commands added/modified. 5. Updated Errors and Warnings table.	R.T.
1.6	16-01-2018	1. Added Sorter Control Commands 2. Removed Set/Get full level setting commands. They have become obsolete. The tube full level is now determined by the full level opto sensor only.	R.T.
1.7	09-08-2018	1. included I/O commands 2. defined floating operation 3. simplified the refill procedure 4. updated coins capacity	M.S.
1.8	10-10-2018	1. Empty Flow charts updated. 2. Set/Get Start Sorting Commands updated.	R.T.
1.10	20-12-2018	1. Error Status table updated.	R.T.
1.11	08-03-2019	1. Firmware update procedure text updated.	R.T.
1.12	28-03-2019	Legacy text removed.	R.T.
1.13	15-04-2019	1. Sorting command updated.	R.T.
1.14	04-11-2019	1. Added the part ‘Using the ICS’ in the Introduction	R.T.
1.15	13-11-2019	1. Added Get Stock Command 31.	R.T.
1.16	14-11-2019	1. Corrected the Get Stock Command 31.	R.T.

This manual is intended only to assist the reader in the use of this product.

Therefore SUZOHAPP shall not be held liable for any loss or damage whatsoever arising from the use of any information or particulars in or any commission from this manual or any incorrect use of the product.