MCi (version 2.0x)

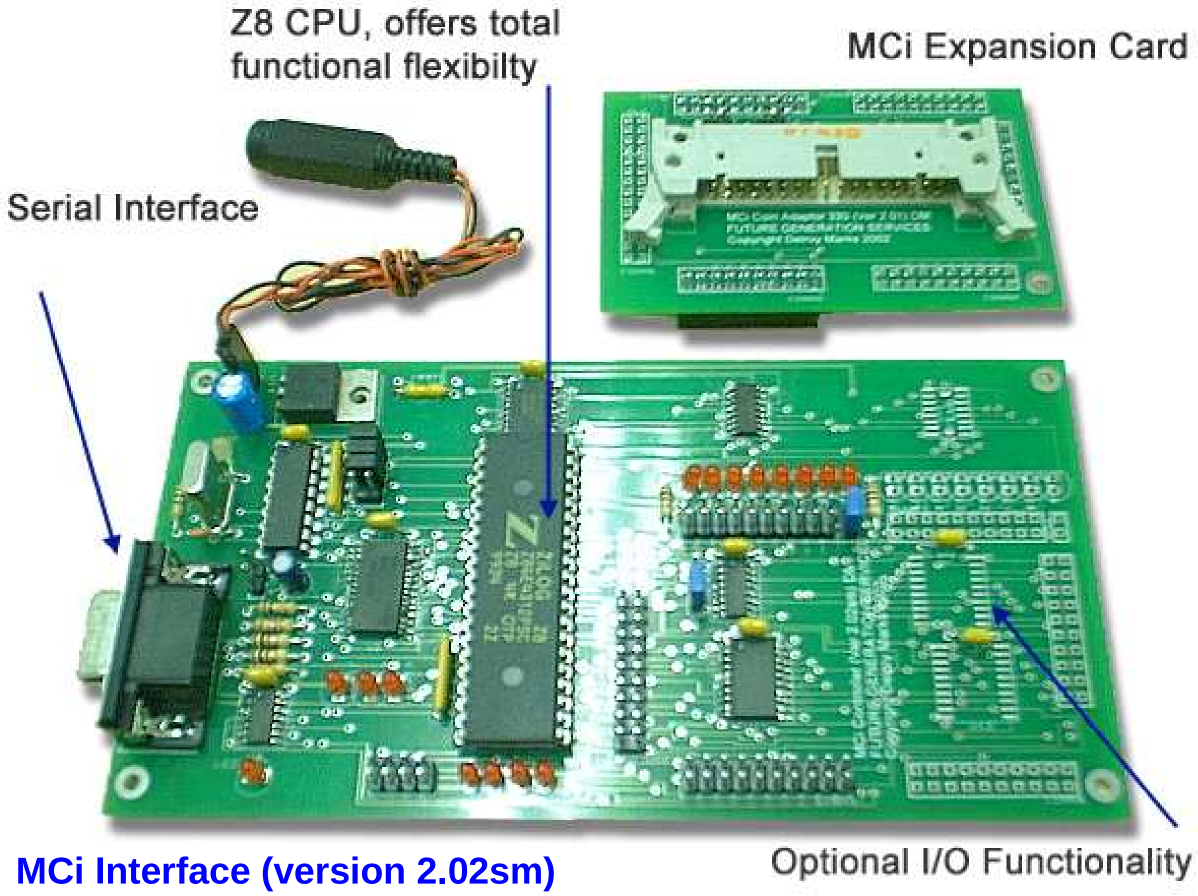
Multi-Function Controller Interface

Future Generation Services

User / Programming Guide

The Future Generation Multi-Function Controller Interface (MCi) is a robustly designed PCB. Utilising traditional well proven discrete logic components, the interface is designed as a solid-state micro controller, with an extremely flexible Z8 processor and internal system software. Basically, there is nothing that can go wrong, and it can be programmed to perform almost any function.

The controller depicted below is available in a variety of hardware and firmware configurations. It can also be bespokely programmed to provide customer specific functionally.


Z8 CPU, offers total functional flexibilty
MCi Expansion Card
Serial Interface
Optional I/O Functionality

MCi Interface (version 2.02sm)

Backward Compatibility:

The **MCi ver.2.02sm** with **Adapter 330 ver.2.0** running firmware version **1.32** and **1.34** was designed to work with the Mars 330 dual polarity coin acceptance unit in a similar basic fashion to the original **MCi ver.1.20m** and is backward compatible. The 330’s were originally developed for gaming machines, and have no direct connectivity to today’s standard PC.

The MCi works in several modes to convert event signals from the 330 into serial data format that can be read by any host PC. This enables software developers to interpret the events of the 330 unit for use in applications. The MCi will also interpret command from the host PC to provide application control over the 330 unit.

MCi Version.	Expansion Ver.	Firmware Ver.	Function / Compatibility
2.02sm	330 v2.0	1.32	Mars 330 Coin Acceptance unit
2.04sm	330 v2.01	1.34	Mars 330, Lumina Note Validator
2.05sm	330 v2.01b	1.34	Mars 330, Lumina Note Validator, Tamper sense
2.05sm	330 v2.01c	1.41	Mars 330, Lumina Note Validator, Dynamic I/O

New Firmware Versions:

The **MCi ver.2.02sm** running firmware **version 2.02** is an enhanced version of the original interface that features an intelligent conversation mode to make application programming much easier. It also has a more extensive list of useful commands. This configuration is recommended for new application development where backward compatibility is not an issue.

MCi Version.	Expansion Ver.	Firmware Ver	Function / Compatibility
2.02sm	330 v2.0	2.02	Mars 330 Coin Acceptance unit, Watchdog
2.04sm	330 v2.01	2.02	Mars 330, Lumina Note Validator, Watchdog
2.05sm	330 v2.01b	2.02	Mars 330, Lumina Note Validator, Watchdog, Tamper detection, Shock detection
2.05sm	330 v2.01c	2.02	Mars 330, Lumina Note Validator, Dynamic I/O, H/W Watchdog, Tamper detection

The **MCi ver.2.0xsm** can have a bespoke firmware version developed to meet a specific customers requirement. This is especially useful in cases where obsolete hardware needs to be replaced, but the existing application is still required to function correctly.

MCi Version.	Expansion Ver.	Firmware Ver	Function / Compatibility
2.0xsm	330 v2.0x	2.9x	Customer specific functionality
2.0xsm	-	“	“

Set-up

First ensure the PC is shutdown and powered off. Then connect both the Serial and power cables into there respective socket on the MCi card and at the back of the host computer. Switch on the power to the MCi card and check for 4 rapidly pulsing LED's, then power on the host PC. After successful boot up, check for MCi operation using Hyper Terminal. The MCi default serial settings are 9600,no parity,8 bits,1 stop bit.

Operation

In both firmware **version 1.3x** and **2.02**, when the unit is first powered on, it always defaults to a reset state with the coin acceptance unit and note validator disabled.

With firmware **version 1.3x**, you must send the correct instruction character (byte) to the serial port to enable the MCI and tell it what coins or notes to accept. It will then report every coin or note accepted event until another instruction character is received. See **Command list 1.3x** for a list of all the available commands and responses

With firmware **version 2.02**, you must send the correct command (up to 5 characters) to the serial port to communicate with the MCI and give it your instructions. It will report all responses and events in a verbose manner. See **Command list 2.0x** for a list of all the available commands and responses

Using Firmware Version 1.32

Testing for responses with MARS 330 COIN ACCEPTANCE UNIT

The coin acceptance / note validator units are supplied configured to your currency and coin/note requirements. We can not supply you with the response information related to your validator events. You must therefore make a preliminary coin/note acceptance test to acquire the responses for programming your application.

To start testing, have ready at least one of each coin/note denomination that your coin/note validator unit has been configured to accept. Reset or apply power to the MCI card. Ensure that 4 LED's 21,22,23 and 24 are illuminated and pulsing. Also check that LED's 17,18 and 19 are off. The MCI is in a state of reset.

Now send the following command as an ASCII character (byte) to the serial port.

Interface Version	Firmware Rev.	Hex	Decimal	ASCII Character
MCI Ver 1.20m	ver. 1.20	40H	64	@
MCI Ver 2.02sm	ver. 1.32	40H	64	@

This instruction will tell the MCi to allow the 330 unit to accept all coins. If you are using Hyper Terminal or any dumb ASCII terminal, you will be able to see the responses sent by the MCI when a coin has been accepted or an event occurs. The following response should have been generated.

The following example shows how to route commands to the 330 coin validator with the responses generated.

Command to send	}	@	(tell 330 to accept all coins)
Response return	}	**Enable All:**	(all coins will be accepted)

Note that only the 2 outer LED’s 21 and 24 are pulsing. Introduce each of your coins in turn and make a note of the responses for future reference. The example table below is a list of the responses for the 330 /DP/UK.

Responses return	}	**Coin#1**	=	5p	coin detected
		Coin#2	=	£ 2.00	“
		Coin#3	=	10p	“
		Coin#4	=	20p	“
		Coin#5	=	50p	“
		Coin#6	=	£ 1.00	“

After noting which coins equal which response, you are almost ready to use the MCI. You should now send the following command to prevent the 330 from accepting any further coins.

Interface Version	Firmware Rev.	Hex	Decimal	ASCII Character
MCI Ver 1.20m	ver. 1.20	3FH	63	?
MCI Ver 2.02sm	ver. 1.32	3FH	63	?

The following example shows how to prevent all coins from being accepted by the 330 with the responses generated.

Command to send	}	**?**	(tell 330 unit not to accept any coins)
Response return	}	**Disable All:**	(no coins will be accepted)

Note that after the ‘Disable All’ command, all 4 LED’s 21,22,23 and 24 are pulsing

Testing for responses with LUMINA NOTE VALIDATOR

If you are using **MCI Ver 2.02sm** with a note validator, you may repeat the same testing procedure as above, to check for the note responses.

In order to start testing for the note responses with this firmware version, you will first need to tell the MCi to change validator.

Interface Version	Firmware Rev.	Hex	Decimal	ASCII Character
MCI Ver 2.02sm	ver. 1.32	2BH	43	+
MCI Ver 2.02sm	ver. 1.32	40H	64	@

The following example shows how to route commands to the LUMINA note validator with the responses generated.

Command to send	}	**+**	(change to LUMNIA unit)
Response return	}	**Command Option:**	(all commands will be routed to the note validator)
Command to send	}	@	(tell LUMINA to accept all notes)
Response return	}	**Enable All:**	(all notes will be accepted in Non-ESCROW)

Note that only the 2 outer LED's 21 and 24 are pulsing. When you have finished testing for the note responses, you can prevent further notes from being accepted as follows:

Command to send	}	**?**	(tell LUMINA not to accept any notes)
Response return	}	**Disable All:**	(no notes will be accepted)
Command to send	}	**-**	(change back to 330 coin unit)
Response return	}	**Command Coin:**	(routes all commands to the coin validator)

Note that after the 'Disable All' command, all 4 LED's 21,22,23 and 24 are pulsing

(Please note that MCI Ver 1.20m does not support a note validator unit.)

Using Firmware Version 2.02

Testing for responses with MARS 330 COIN and LUMINA NOTE VALIDATOR

When testing for the coin / note responses with this firmware version, there is no need to change validator. Test for coin and note responses as above using the following commands:

Interface Version	Firmware Rev.	Hex	Decimal	ASCII String
MCI Ver 2.02sm	ver. 2.02	-	-	COIN@

The following example shows how to send commands to the Mars 330 Coin validator with the responses generated.

Command to send	}	**COIN@**	(tell 330 to accept all coins)
Response return	}	**Command OK:**	(all coins will be accepted)

When you have finished testing for the coin responses, you can prevent further coins from being accepted as follows:

Command to send	}	**COIN?**	(tell 330 unit not to accept any coins)
Response return	}	**Command OK:**	(no coins will be accepted)

Interface Version	Firmware Rev.	Hex	Decimal	ASCII String
MCI Ver 2.02sm	ver. 2.02	-	-	NOTE@

The following example shows how to send commands to the LUMINA note validator with the responses generated.

Command to send	}	**NOTE@**	(tell LUMINA to accept all notes)
Response return	}	**Command OK:**	(all notes will be accepted in Non-ESCROW)

When you have finished testing for the note responses, you can prevent further notes from being accepted as follows:

Command to send	}	**NOTE?**	(tell LUMINA not to accept any notes)
Response return	}	**Command OK:**	(no notes will be accepted)

Note that after the ‘Disable All’ command, all 4 LED’s 21,22,23 and 24 are pulsing

In all firmware versions the Enable All character ‘@’ and Disable All character ‘**?**’ are the same. To complete your MCI testing. You should now use the coin / note inhibit instructions to identify which coins / notes are inhibited in the same testing way as you did before. By reviewing the instruction table below, you will quickly understand how to control the MCI and use it in your application.

Inhibit Character Composition Table

Data Bits

D7	0	0	0	0	0	0	0	0	Not used
D6	0	1	1	1	1	1	1	1	MCI enable
D5	1	0	0	0	0	0	0	1	Inhibit Coin#6
D4	1	0	0	0	0	0	1	0	Inhibit Coin#5/ESCROW
D3	1	0	0	0	0	1	0	0	Inhibit Coin#4 /Note#4
D2	1	0	0	0	1	0	0	0	Inhibit Coin#3 /Note#3
D1	1	0	0	1	0	0	0	0	Inhibit Coin#2 /Note#2
D0	1	0	1	0	0	0	0	0	Inhibit Coin#1 /Note#1
Byte Hex	3FH	40H	41H	42H	44H	48H	50H	60H	
Decimal	63	64	65	66	68	72	80	96	
Character	?	@	A	B	D	H	P	`	

You may also inhibit more than one coin /note by adding the inhibit bits together as shown below.

Data Bits

D7	0	This instruction byte would inhibit both Coin#1 and Coin#3 if sent to the 330 unit or Note#1 and Note#3 if sent to the LUMINA unit.
D6	1	
D5	0	
D4	0	
D3	0	
D2	1	
D1	0	
D0	1	
Byte Hex	45H	
Decimal	69	
Character	E	

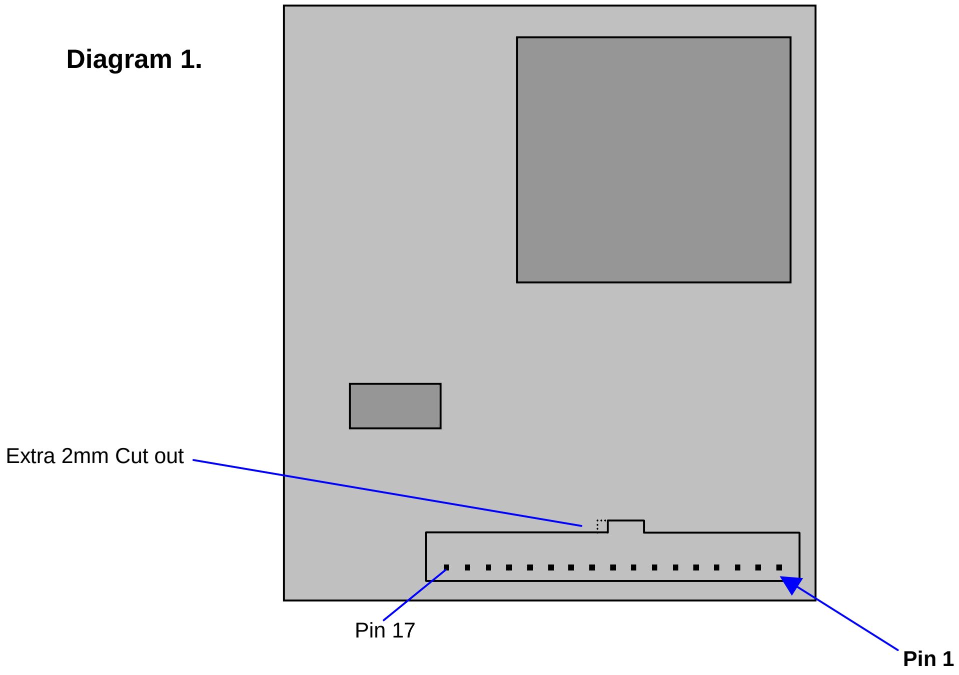
Do be careful to correctly construct your instruction bytes, as you could experience strange results. If you require assistance, call our tech support team +44 1634 718662

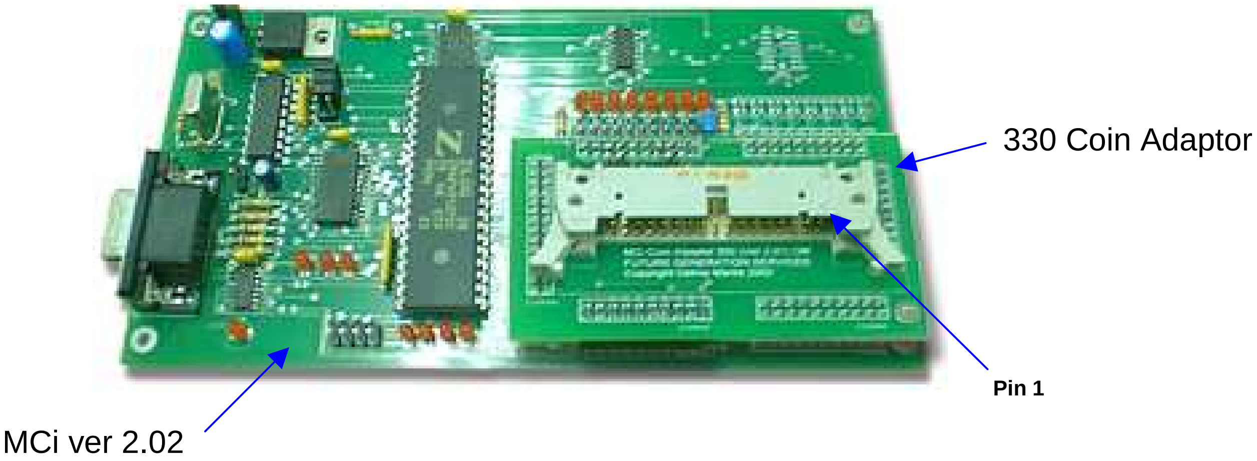
Connecting to 330 Coin Acceptance Unit

The Interface expansion cards are designed to make connecting to peripherals easy. In the case of the Adaptor 330 ver 2.01, use a standard 34 way IDC ribbon cable. Connect one end to the interface while observing the pin orientation, and the other end to the 330 coin acceptance unit. On some older units the current keyed slot may need to be enlarged by approximately 2mm as shown. The connector should fit snugly into position over the bottom set of pins.

330 Coin Acceptance Unit

Diagram 1.





Command List

Firmware Version 1.3x Commands

The following are commands for the **MCI Ver 2.02sm** running firmware **version 1.32**

Command	Hex	Dec	ASCII	Function
Reset MCi	7E	126	~	Pauses for 5 seconds then reset
Quiet Mode	22	34	“	Gives no responses after inhibit character is sent
Verbose Mode	21	33	!	Supplies a binary picture after each instruction is sent
Disable All	3F	63	?	Inhibits all coins / notes. None will be accepted
Enable All	40	64	@	Allows all coins / notes to be accepted
Inhibit Bytes	41-7D	65-125	A-}	Inhibit combinations. See Inhibit Character Composition Table

Firmware Version 2.0x Commands

The following are commands for the **MCI Ver 2.02sm** running firmware **version 2.02**

Command	ASCII String	Function
Reset MCi	RESET	Pauses for 5 seconds then reset
Greeting	HELLO	Checks that MCi is functioning.
WatchDog Off	HM=0	Hardware Monitor is disabled
WatchDog On	HM=1 to 9	Hardware Monitor will timeout after 10x integer entered. eg 2=20 sec
Quiet Mode	MSG=Q	Gives no responses after inhibit character is sent
Verbose Mode	MSG=X	Supplies a binary picture after each instruction is sent
Allow no Coins	COIN?	Inhibits all coins. None will be accepted
Allow All Coins	COIN@	Allows all coins to be accepted
Allow no Notes	NOTE?	Inhibits all notes. None will be accepted
Allow All Notes	NOTE@	Allows all notes to be accepted
Pay out hopper	PAY1 to 9	Will command the pay out hopper to deliver 1x integer entered.
Inhibit Bytes	41-7D 65-125	A-} Inhibit combinations. See Inhibit Character Composition Table

Function of the Z8 program, written to control Escro Unit.

Please carry out the following test to confirm the functionality of your Coin/Escro system;

Run the **Hyper Terminal** Program
Configure for **COM PORT 2** with settings as follows;
Bits per second = **9600**
Data Bits = **8**
Parity = **None**
Stop Bits = **1**
Flow Control = **Hardware**

- In hyper terminal enter the character **'~'** to reset the Mci Controller (which is achieved by pressing **'SHIFT'** and **'#'** next to the **'Enter'** key)

- Response should be **"Reset in 5 seconds."** Then after a few seconds
- Response should be **"MCi Version 1.4Esc - Self Test OK."** and a few other information lines

- To Accept the Escro Payment, enter the character **'£'**, this will release the Cash from Escro mechanism into the cash box for approx. 1.5 seconds and automatically close the unit again with the response **"Escro Done"**.

- To Reject the Escro Payment, enter the character **'¬'**, this will release the Cash from Escro mechanism out to the payout tray for approx. 1.5 seconds and automatically close the unit again with the response **"Escro Done"**.

- To disable the timed Escro release, enter the character **'>'**.
  Response should be **"Escro Manual"**

- In this mode the **'£'** or **'¬'** characters will release the Escro mechanism indefinitely.

- To close the Escro mechanism, enter the character **'?**.
  Response should be **"Disable All"**

- Warning: In this manual mode you must send the **'?'** to close the Escro mechanism after releasing it with the **'£'** or **'¬'**.

  DO NOT LEAVE THE ESRCO RELEASED FOR LONG PERIODS AS THIS WILL RESULT IN DAMAGE TO THE ESCRO UNIT.

- To re-enable the timed Escro release mode enter the character **'<'**.

- Response should be **"Escro Auto"**. This returns to the automatic release mode, closing the Escro mechanism after 1.5 seconds.

The Mci controller always powers up into automatic Escro release mode by default.

To reset the MCi Controller, you can enter the character **'~'** at any time.

Although the MCi Controller program has been specially modified for Escro operation, it does have many other functions, which may not be relevant.

Please do not hesitate to contact us should you require any assistance.