RCBus Specification

Pre-release
for review and
feedback

Version 0.0.006
9-March-2023

1. Introduction

This document is the reference description for the Retro Computer Bus (RCBus) version 1.0. It is intended to guide the implementation of compatible devices on the bus. The goal of RCBus is to provide an environment for the development and exploration of digital electronics. This guide therefore should be considered as a reference, not as law. Developers are positively encouraged to use the bus in innovative and bizarre ways.

2. History

The RCBus has its origins in Spencer Owen's RC2014 (retrochallenge 2014) entries and the simple passive Z80 bus interface it used to plug simple one or two function cards into a backplane. Spencer went on to launch a business (rc2014.co.uk) based upon this concept and the product range was trademarked as RC2014™.

Over time the bus evolved and has been modified, used (and abused) in a variety of ways by other parties, notably the 80 pin BP80 extensions to the original for Z180 systems.

RCBus is a specification based upon that generalization of that bus, and a branding for it that avoids confusion with the RC2014 product line, and also RetroChallenge 2014 (RC2014).

3. The Bus

This document describes the RCBus in terms of the passive backplane and plug in cards. Nothing precludes some of those cards being combined or the bus being an extension bus (possibly one of several extension busses) for a single board computer or other arrangement.

The underlying concept of the RCBus is a passive bus that carries mostly Z80 compatible signals using 5v signalling and with CMOS parts to keep the bus signalling low power and the fan out high. These design choices avoid the need for complex buffer circuits on the cards and make the bus much more accessible to newcomers.

The backplane consists of between 40 and 80 lines including power. The bus is not usually terminated nor is this necessary. There is no maximum number of card slots specified by the bus design but at 8MHz systems with over 15 cards appear to operate quite reliably.

A backplane may contain multiple variants of the slots. Several have a mix of 80 and 40 pin slots for example. A backplane may also contain combinations of RCBus and other bus interfaces providing compatibility is maintained.

Cards stand vertically in the backplane slots. If placed horizontally, additional mechanical support for each card is likely to be required. The minimum spacing between cards should be 0.6" (15.24mm). Wider spacings, such as 0.8" (20.32mm), may be used and may be more convenient. It may be useful to allow additional space with some slots as certain cards are much deeper than normal. Suggested card profiles can be found in Appendix B. Cards should not dip down below their connector in order to gain space at the ends, as the backplane may itself have components alongside, but below the height of the sockets.

Power is supplied to the cards via the bus. Care should therefore be taken both of the grounding and the sizes of power traces. Nothing precludes a card with high power demands also having a direct power connector to the bus 5v/ground.

The backplane has signals that were historically referred to as "USER" signals. Some of these signals have other recommended uses in the RCBus standard but their usage remains optional and neither cards nor backplanes should assume a particular usage model for these lines.

RCBus is somewhat "legacy free". Most RCBus systems do not provide signals for dynamic RAM refresh and there are no 12v or -5v signal lines. Signals should be at CMOS levels and the use of 74HCT/AHCT CMOS parts for all bus connected signals, particularly input signals is strongly recommended. The use of other logic types isolated from the bus signal lines is a matter for the card designer.

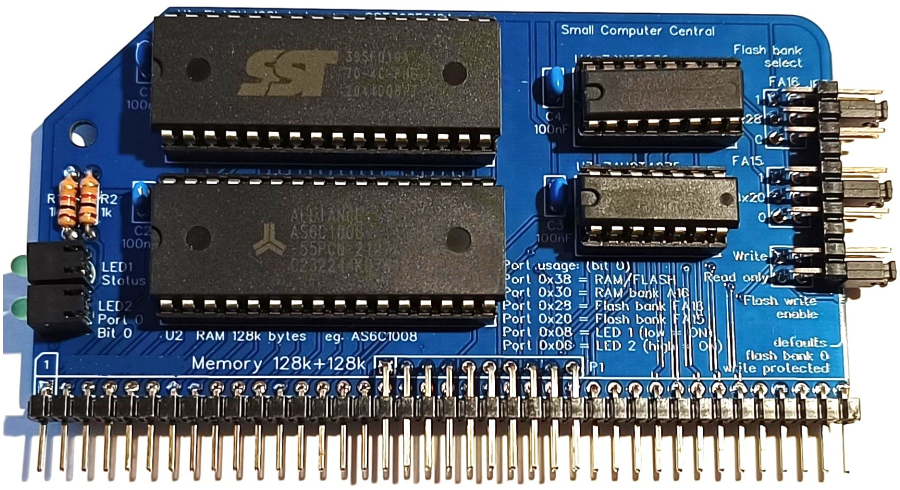
There are three kinds of RCBus cards. The processor card is the source of many of the signals and there should be only one. (Extending the bus to support multiple bus masters is ongoing) The second type of cards are passive cards. These provide power or control signals but are not active participants in bus traffic. Examples include clock generators, power supplies and reset controllers. The final card type is device cards. These are the consumers of the signals from the processor and are active participants on the bus.

The naming reflects the historic separation of functions on the bus. Nothing prevents a card from providing functions in two or even all three categories at once.

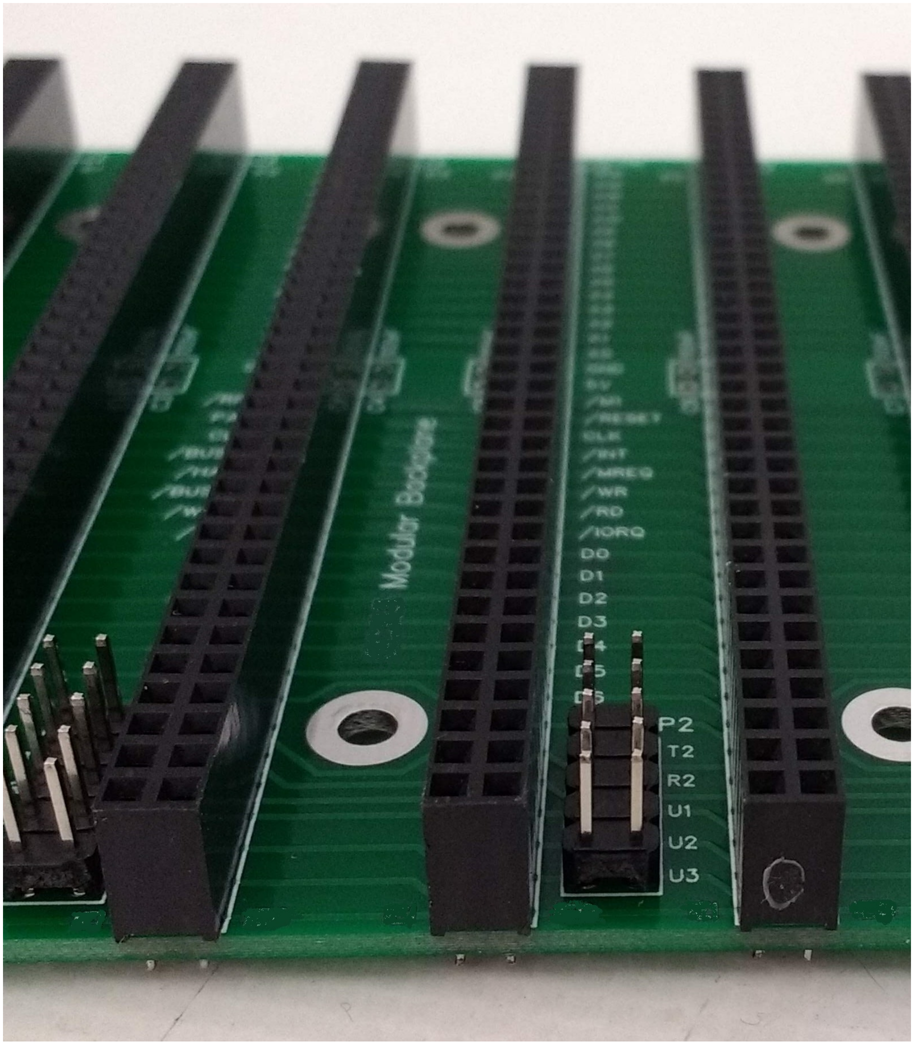
4. Backplanes and Modules

Backplanes and modules can be any size and shape, but established norms are worth following.

Below is an example of an RCBus module which has a partial second row of bus pins.

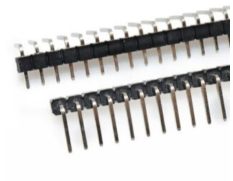


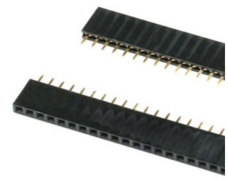
Below is an example of an RCBus backplane which has 80-pin module sockets.

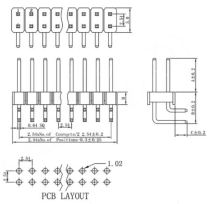


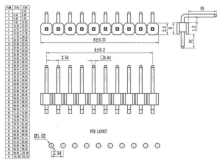
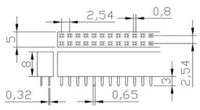
5. Bus Connectors

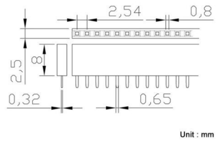
The bus connectors are low cost header pins and sockets, either single or double row with up to 40-pins in each row.



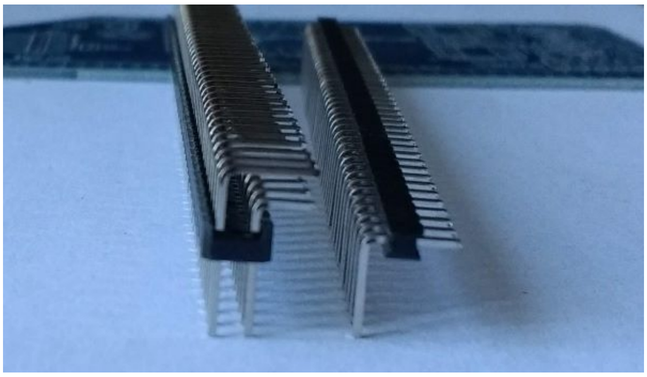
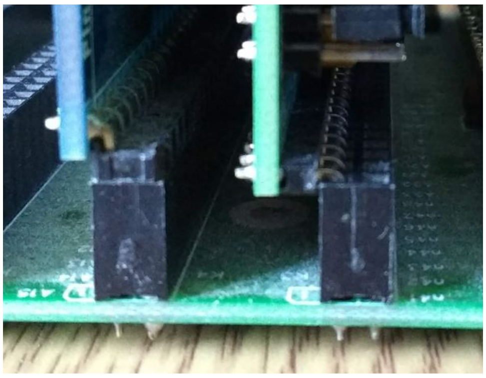




Pin length ~6mm, pitch 2.54mm

Note that the most commonly used male 1x40 angled connector has a different profile to the 2x40 connector.

Single row bus socket viewed from above:

Pin 1	Pin 40
Front	*Back*

Double row bus socket viewed from above:

Pin 1	Pin 40
Pin 41	Pin 80
Front	*Back*

6. Bus Pin Assignments

Bus signals are either common to all bus specifications or specific to one or more specifications.

Pin 1st row	RC2014™ standard	RCBus backplane	RCBus Z80	RCBus 68xx	RCBus 9995	Notional 16-bit CPU		
1	A15							
2	A14							
3	A13							
4	A12							
5	A11							
6	A10							
7	A9							
8	A8							
9	A7							
10	A6							
11	A5							
12	A4							
13	A3							
14	A2							
15	A1							
16	A0							
17	GND							
18	+5V							
19	nM1							
20	nRESET							
21	CLOCK							
22	nINT							
23	nMREQ							
24	nWR							
25	nRD							
26	nIORQ							
27	D0							
28	D1							
29	D2							
30	D3							
31	D4							
32	D5							
33	D6							
34	D7							
35	TX							
36	RX							
	2014	*Backplane*	*Z80*	*68xx*	*9995*	*16-bit*		
37	USER1	p37	INT1	FIRQ	MEMEN	p37		
38	USER2	p38 [1]	IEI	E	CRUIN	p38 [1]		
39	USER3	p39 [1]	IEO	RW	CRUCLK	p39 [1]		
40	USER4	p40	USR4	USR4	USR4	USR4		

[1] Backplane configurable for direct or cascade (daisy-chain) connections

Second row bus signals:

Pin 2nd row	RC2014™ enhanced	RCBus Backplane	RCBus Z80	RCBus 68xx	RCBus 9995	Notional 16-bit CPU		
41	N/A	p41	p41	p41	p41	p41		
42	N/A	p42 [1]	BAI	p42	p42	p42		
43	N/A	p43 [1]	BAO	p43	p43	p43		
44	N/A	p44	p44	p44	p44	p44		
45	N/A	p45	p45	p45	p45	p45		
46	N/A	p46	p46	p46	p46	p46		
47	N/A	p47	p47	p47	p47	p47		
48	N/A	p48	p48	p48	p48	p48		
49	N/A	A23						
50	N/A	A22						
51	N/A	A21						
52	N/A	A20						
53	N/A	A19						
54	N/A	A18						
55	N/A	A17						
56	N/A	A16						
57	GND							
58	+5V							
59	nRFSH							
60	PAGE							
61	CLOCK2							
62	nBUSACK							
63	nHALT							
64	nBUSRQ							
65	nWAIT							
66	nNMI							
67	D8							
68	D9							
69	D10							
70	D11							
71	D12							
72	D13							
73	D14							
74	D15							
75	TX2							
76	RX2							
	2014	*Backplane*	*Z80*	*68xx*	*9995*	*16-bit*		
77	USER5	p77	INT2	p77	p77	p77		
78	USER6	p78	p78	p78	p78	p78		
79	USER7	p79	p79	p79	p79	p79		
80	USER8	p80	USR8	USR8	USR8	USR8		

[1] Backplane configurable for direct or cascade (daisy-chain) connections

7. Bus Signals

Signals are TTL level unless otherwise indicated.

7.1. Common Signals

This section describes each of the signals common to all processor types.

Signals are defined by their function with a Z80 CPU unless otherwise stated.

+5V	Power supply +5 volts, recommended tolerance +/-0.25 volts.
A0 to A23	Address bus, output from CPU, active high, tristate. Typically, 8-bit processors will only use A0 to A15.
nBUSAK	Bus acknowledge, output from CPU, active low. The CPU outputs a low on this line indicating the address bus, data bus, and control signals nMREQ, nIORQ, nRD, and nWR have entered their high-impedance states.
nBUSRQ	Bus request, input to CPU, active low. This is pulled low by a device that wants to take control on the CPU's address, data and control bus.
CLOCK	Clock, input to CPU. System clock used for synchronisation so it should normally be the CPU clock signal.
CLOCK2	This is a second clock usually used as a clock source for a UART.
D0 to D15	Data bus, input/output, active high, tristate. 8-bit processors will only use D0 to D7. Unused data lines can be left floating.
GND	Power supply ground.
nHALT	Halt, output from CPU, active low. This indicates the CPU has executed a HALT instruction and is waiting for an interrupt before resuming operation.
nINT	Interrupt request, input to CPU, active low. This signal should have a pull-up resistor and is pulled low by any device requesting an interrupt.
nIORQ	Input/output request, output from CPU, active low, tristate. The Z80 has a separate address space for I/O devices but other processors will likely need to create a window in their memory map for I/O.
nM1	Machine cycle one, output from CPU, active low. This signal is very specific to the Z80 and should be pulled up for other processors.
nMREQ	Memory request, output from CPU, active low, tristate. Indicates that the address bus holds a valid address for a memory read or a memory write operation.
nNMI	Non-maskable interrupt, input to CPU, active low. Negative edge triggered interrupt.
PAGE	Page RAM/ROM, active high. Low to enable ROM and disable its shadow RAM. High to disable ROM and enable its shadow RAM. For a Z80 system the ROM is at the bottom of memory (starting at address 0x0000) following reset. This signal is set high to replace this ROM with RAM, allowing a full 64k of RAM. Perhaps a better name would be RAM_ENable. There are other memory management schemes in use that do not use this signal.
nRD	Read, output from CPU, active low, tristate. Indicates that the CPU wants to read data from memory or an I/O device.

nRESET	Reset, input to CPU, active low. System reset signal.
nRFSH	Refresh, output from CPU, active low. Indicates a memory refresh cycle.
RX	This signal is designed to allow a serial port to communicate with a device on another module. RX is an input to a serial port, such as a UART, and an output from a serial device, such as a terminal module.
RX2	A second serial communications 'RX' signal.
TX	This signal is designed to allow a serial port to communicate with a device on another module. TX is an output from a serial port, such as a UART, and an input to a serial device, such as a terminal module.
TX2	A second serial communications 'TX' signal.
nWAIT	Wait, input to CPU, active low. Communicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a WAIT state as long as this signal is active.
nWR	Write, output from CPU, active low, tristate. Indicates that the CPU wants to write data to memory or an I/O device.

7.2. **RCBus-2014**

The following are signals specific to the basic RC2014™ bus specification.

N/A	These pins are not included on RC2014™ products.
USER#	Signals USER1 to USER8 are free for the user to do as they please.

7.3. **RCBus-Z80**

The following are signals specific to systems using the Z80 family bus specification. Eg. Z80 and Z180.

nBAI	Bus acknowledge in, input, active low. BAI and BAO form a direct memory access priority daisy-chain. BAI signals that the system buses have been released for DMA control.
nBAO	Bus acknowledge out, output, active low. BAI and BAO form a direct memory access priority daisy-chain. BAO signals that the CPU has relinquished control of the bus.
IEI	Interrupt enable in, input to an interrupt generating device, active high. IEI and IEO form an interrupt priority daisy-chain. A high on IEI indicates that no other device of higher priority is interrupting.
IEO	Interrupt enable out, output from an interrupt generating device, active high. IEI and IEO form an interrupt priority daisy-chain. IEO is high only when IEI is high and this device is not requesting an interrupt.
nINT1	Interrupt, input to CPU, active low.
nINT2	Interrupt, input to CPU, active low.
p##	Reserved
USR#	Free for the user to do as they please.

7.4. RCBus-68xx

The following are signals specific to systems using the 68xx, 63xx and 65xx family bus specification.

nFIRQ	Fast interrupt request, input to CPU, active low.
E	Data bus enable, input to CPU, active high.
RW	Read/write, output from CPU. High for read operations, low for write operations.
p##	Reserved
USR#	Free for the user to do as they please.

7.5. RCBus-9995

The following are signals specific to systems using the TMS9995 family bus specification.

nCRUCLK	CRU clock, output from CPU, active low.
CRUIN	CRU input data,input to CPU, active high.
nMEMEN	Memory enable, output from CPU, active low.
p##	Reserved
USR#	Free for the user to do as they please.

8. RC2014 Compatibility

For an RCBus device to be RC2014™ compatible it should use only those pins specified in the RC2014 specification and support Z80 timings. It should work with a 7.3728MHz signal on clock.

For an RCBus CPU card to be RC2014 compatible it should provide at least all the signals in the base RC2014 specification. The processor must be a CMOS Z80 and, if providing the clock, should provide the clock at 7.3728MHz.

A strictly RC2014 compatible device or processor is a strict subset of RCBus. There are, by intent, no conflicts between the two.

9. RCBus-Z80

The Z80 processor only operates correctly with Zilog peripherals if the processor and peripherals are running from the same clock. Therefore it is normally desirable that the bus CLOCK is the Z80 input clock and this means that any card generating the system clock should generate a signal suitable for the Z80 processor. In particular the signal levels should be close to 5v and to ground. An RC2014 compatible CPU card will run at 7.3728MHz, and some peripheral cards require this.

The Z80 interrupt chain allows the use of IM2 interrupt mode. This mode has advantages but also significant restrictions. If more than four cards are actively on the chain then the backplane must contain lookahead logic (as per Zilog documentation). In addition, it is not possible to reliably mix IM2 using devices and classic interrupts. Developers, therefore, may want to consider the ability to route some slot interrupts to Z80 CTC or PIO pins to use them as an interrupt controller.

10. Bus Profiles

These are suggested levels of functionality and combinations of signals to rely upon when creating boards or processor cards.

10. 1. Minimal

The following signals are present:
+5V, GND, A7-A0, D7-D0, nIORQ, nINT, nRD, nWR, nRESET

The following signals will be pulled high or valid:
nM1, nMREQ

This subset is designed to allow the use of I/O devices on RCBus slots in a system which is otherwise self contained or uses other bus interfaces.

10. 2. RCBus 40

The following signals are present:
+5V, GND, A15-A0, D7-D0, nIORQ, nMREQ, nINT, nRD, nWR, nRESET

The following signal will be pulled high or valid:
nM1

A clock signal may be present but the value is undefined.

This subset allows the use of memory and I/O cards and contains the functionality that can reasonably be relied upon regardless of the processor card that is in use.

10. 3. **RCBus Enhanced**

This matches the enhanced RC2014™ bus pins.

As well as the RCBus 40 lines nWAIT and nNMI become available as well as the nRFSH, nBUSRQ and nBUSAK for Z80 processors. The RC2014 specification also includes D8-D15 and PAGE but these are nor relevant to RCBus in general.

A clock signal may be present on CLOCK and one may be present on CLOCK2 but their value is undefined.

10. 4. **RCBus 80pin**

A23-A16 are added to the bus. Processor cards only drive the lines they support. This means that a memory card that is designed to support a larger number of address pins than the processor will need jumpers or to pull up those lines. Some processor cards do pull up address lines for other reasons (bus mastering) so if the line is pulled in a direction it should be up.

On the standard 80-pin systems using the Z180 processor the CLOCK will be the Z180 clock (often 18.432MHz) and nM1 will be valid and lines A16-A19 will be in use.

11. Appendix A - Terminology

RCBus A generic term to cover all uses and designs based on the RCBus specification.

RCBus Minimal
This set of signals is designed to support input/output devices only.
Signals: +5V, GND, A0 to A7, D0 to D7, nIORQ, nINT, nRD, nWR, nRESET, nM1, nMREQ
Optional: CLOCK, TX, RX

RCBus 40 This set of signals is designed to support memory and input/output devices.
Signals: +5V, GND, A0 to A15, D0 to D7, nIORQ, nINT, nRD, nWR, nRESET, nM1, nMREQ
Optional: CLOCK, TX, RX

RCBus Enhanced
This matches the enhanced RC2014™ bus pins.
Signals: +5V, GND, A0 to A15, D0 to D7, nIORQ, nINT, nRD, nWR, nRESET, nM1, nMREQ
nWAIT, nNMI, nRFSH, nBUSRQ, nBUSAK, D8 to D15, PAGE, CLOCK, CLOCK2
TX, RX, TX2, RX2
Some of these signals are optional.

RCBus 80-pin Signals: +5V, GND, A0 to A23, D0 to D7, nIORQ, nINT, nRD, nWR, nRESET, nM1, nMREQ
nWAIT, nNMI, nRFSH, nBUSRQ, nBUSAK, D8 to D15, PAGE, CLOCK, CLOCK2
TX, RX, TX2, RX2
Some of these signals are optional.

RCBus Extensions
This refers to the pins that have multiple possible functions depending on the processor or system in which they are used. These pins are p37 to p40 (pins 37 to 40), p77 to p80 (pins 77 to 80), and p41 to p48 (pins 41 to 48).

RCBus-2014 The subset of this specification that applies specifically to compatibility with the RC2014™ bus.

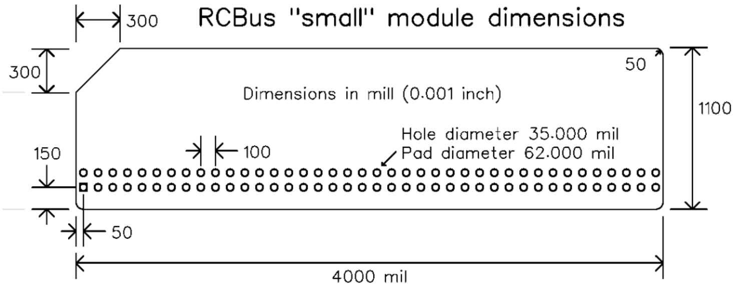
RCBus-68xx The subset of this specification that applies specifically to compatibility with the 6800 and compatible processor bus signalling.

RCBus-9995 The subset of this specification that applies specifically to compatibility with the TMS9995 and compatible processor bus signalling.

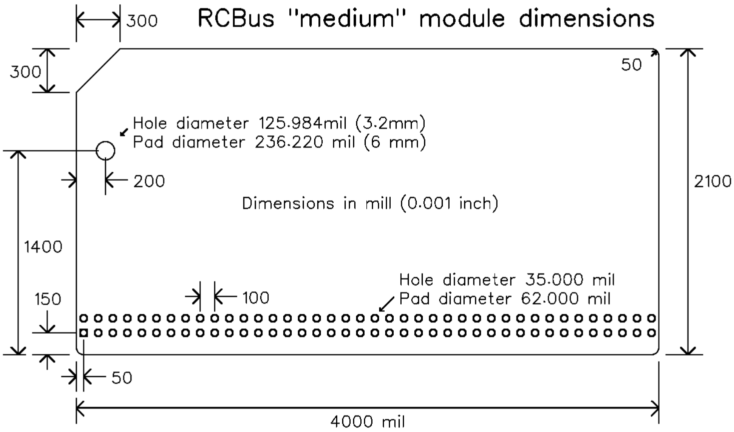
RCBus-Z80 The subset of this specification that applies specifically to compatibility with the Z80 and compatible processor bus signalling.

12. Appendix B - Circuit Board Shapes and Sizes

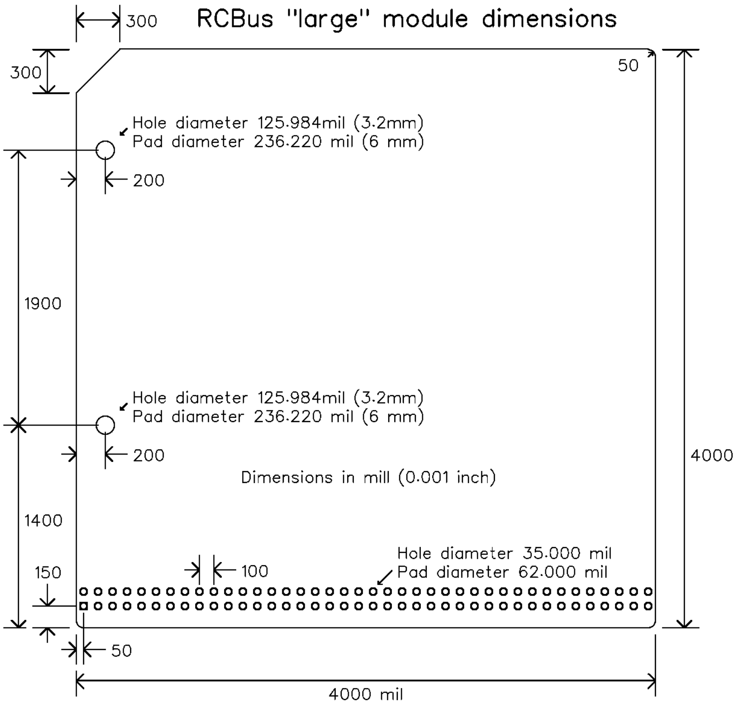
RCBus modules and backplanes can be any shape and size you wish as long as they include RCBus compatible connectors. Below are some suggested shapes and sizes.



The second (top) row of bus pins is optional



It is recommended that the mounting hole has a 6 mm diameter pad or a 6 mm unobstructed clearance for fixings



The most common RCBus-2014 module dimensions can be found at: www.rc2014.co.uk

13. Appendix C - Background

To understand what the RCBus is and why it exists, it is necessary to consider what has led us here.

In 2014 Spencer Owen created a modular version of Grant Searle's Z80 computer design, which he called RC2014. He began selling it on Tindie and in 2016 he gave up his job and worked on RC2014 full time.

The RC2014 system attracted a community of enthusiasts, some of whom made RC2014 compatible modules, as encouraged by Spencer's website: "If your module may be of use to other RC2014 owners, please consider sharing your design or selling them yourself. I'm happy to help you with this and to spread the word. Note that "RC2014" is a registered trademark, so you are not allowed to call your module "RC2014 [thingy] Module" or use the RC2014 logo. However, feel free to mark your modules as "Designed for RC2014.""

Before long there was talk of extensions to the RC2014 bus.

In 2018 the topic "New backplane -wishes ?" led to Spencer posting the following on 17-June-2018:

"I've been musing over enhancements to the backplane for a little while now, and whilst nothing is set in stone, the pin layout would follow this;

Enhanced		**Standard**			**Enhanced**		**Standard**	
A31	1	1	A15		Clock2	21	21	Clock
A30	2	2	A14		BUSACK	22	22	INT
A29	3	3	A13		HALT	23	23	MREQ
A28	4	4	A12		BUSRQ	24	24	WR
A27	5	5	A11		WAIT	25	25	RD
A26	6	6	A10		NMI	26	26	IORQ
A25	7	7	A9		D8	27	27	D0
A24	8	8	A8		D9	28	28	D1
A23	9	9	A7		D10	29	29	D2
A22	10	10	A6		D11	30	30	D3
A21	11	11	A5		D12	31	31	D4
A20	12	12	A4		D13	32	32	D5
A19	13	13	A3		D14	33	33	D6
A18	14	14	A2		D15	34	34	D7
A17	15	15	A1		Tx2	35	35	Tx
A16	16	16	A0		Rx2	36	36	Rx
Gnd	17	17	Gnd		USR5	37	37	USR1
5v	18	18	5v		I2C SDA	38	38	IEI
RFSH	19	19	M1		I2C SCL	39	39	IEO
Page	20	20	Reset		USR8	40	40	USR4

However, on 16-June-2019 Spencer created the topic "Upcoming changes to the RC2014 bus and ecosystem" in which he stated "RC2014 will not be changing" and further clarified this by stating "The RC2014 bus does not support IEI/IEO modules. Through-hole components are used. And the physical bus will not be changing."

This came as a bit of a blow to those who were looking to build on the RC2014 system and led to a discussion about how to move forward. This discussion didn't produce any solid answers.

On 31-Jan-2023 Spencer created the topic “What has an RC2014 and a Hoover got in common?” in which he stated the following:
”RC2014 is a trademark under British law, belonging to RFC2795 Ltd (ie my company).”
“All of these kits carry the RC2014 name and RC2014 logo, and are labelled as being RFC2795 Compliant.”
“Any other kit is NOT an RC2014.”
“All sellers seem to do a very good job of making the distinction in their listings. But I don't think it is doing anybody any favours by calling a non-RC2014 machine an RC2014, least of all to the creators of compatible machines.“

This led to another discussion about the future in which it was suggested that a new name be found for products that have a degree of compatibility with RC2014 products but are not made by RFC2795 and are thus not RC2014 products. To this suggestion, Spencer wrote: “RCBus or RC80 Bus sound good to me. It takes the essence of what the bus is without limiting it by what the RC2014 natively supports.”

Spencer’s official description of an RC2014 remains:
“RC2014 is a simple 8 bit Z80 based modular computer. It is inspired by the home built computers of the late 70s and computer revolution of the early 80s. It is not a clone of anything specific, but there are suggestions of the ZX81, UK101, S100, Superboard II and Apple I in here. It nominally has 8K ROM, 32K RAM, runs at 7.3728MHz and communicates over serial at 115,200 baud.”

Much of what some in the retro computer community wish to do with their RC2014 based systems does not match this description.

And thus the RCBus project was created.

14. Appendix D - RC2014TM USER Pin Usage

The RC2014TM bus has a number of spare pins, usually called USER pins. These have been used by designers to add functions not provided by the defined bus pins. The RCBus specification attempts to maintain compatibility with the most common uses. The following is a list of some of those uses.

Product	Pin 37	Pin 38	Pin 39	Pin 40	Pin 77	Pin 78	Pin 79	Pin 80
RC2014	USER1	USER2	USER3	USER4	USER5	USER6	USER7	USER8
BP80	USER1	USER2	USER3	IEO	USER5	USER6	USER7	IEI
SC102 Z80 CTC	BCT3*	IEI*	IEO*		BCT0*	BCT1*	BCT2*	
SC103 Z80 PIO		IEI*	IEO*					
SC104 Z80 SIO/2		IEI*	IEO*					
SC110 SIO+CTC	CTC3*	IEI*	IEO*		CTC0*			
SC111 Z180 CPU	INT1*							
SC112 Backplane				IEO				IEI
SC113 Backplane				IEO				IEI
SC116 Backplane				IEO				IEI
SC126 Z180 SBC				IEO		SCL* (I2C)	SDA* (I2C)	IEI
SC132 Z80 SIO/0		IEI*	IEO*					
SC149 Z80 CPU	BUSAK*	WAIT*	BUSAK*	NMI* BUSAK*				
Z80 CPU + CTC module (TP)	CTC3*	IEI*	IEO*		CTC0*			
Z180 MPU (TP)	INT1*				INT2*			
Z280 MPU (TP)	INTA*				INTC*			
Universal SIO (TP)		IEI*	IEO*					
DUART (TP)		IEI*	IEO*					
16450/550 (TP)	IRQ*				IRQ*			
Network Controller (TP)	IRQ*							
Basic & Modular Backplane 4 (TP)	USER1*	IEI*/USER2* direct or cascade	IEO*/USER3* direct or cascade	USER4	USER5	USER6	USER7	USER8
6809E/6309EP (TP)	FIRQ*	E*	RW*					
65C02 65C816 6803/6303 6808 6809/6309 68HC11 65C22 6840 65C21 (Alan Cox)	FIRQ* (some)	E*	RW*					
TMS9995 (Alan Cox)	MEMEN* ?	CRUIN* ?	CRUCLK* ?					

* = jumpered so the end user can select if the bus pin is connected or not

Product	Pin 37	Pin 38	Pin 39	Pin 40	Pin 77	Pin 78	Pin 79	Pin 80
PPI (Dino)	LED*	SLT_A*	CASS*	SLT_B*	OUT*			
Easy-Z80 (Sergey Kiselev)		IEI	IEO					
Z80Ctrl/CPU/IO X (JBLangston)	SCL	MISO	MOSI		IOXCS	AUXCS1	AUXCS2	

* = jumpered so the end user can select if the bus pin is connected or not

15. Appendix E - Bus Conventions For Mapping Motorola Busses

15.1. Introduction

The Motorola style 8-bit bus differs considerably from the bus expected by the conventional RCBus. It is possible to map from one to the other but it can be useful when integrating Motorola bus devices to make the Motorola bus signals available.

This appendix documents the existing conventions that are used for mapping a Motorola style bus to the RCBus. It is intended to be descriptive not prescriptive.

15.2. Mapping The Bus

The following signals are mapped directly onto the RCBus from the Motorola style bus.

A15-A0: Address bus
D0-D7: Data bus
nINT: IRQ (open collector)
nRESET: RESET (see notes section)
CLOCK: (see notes section)

The Motorola bus has two different signals. E is a square wave clocking the bus. During one half of the E cycle the signals change, during the other half of the E cycle the signals are valid. There is no provision for an "idle" cycle, instead an additional read cycle is generated. This is usually targetted at a dummy location such as 0xFFFF but some processors will generate dummy read cycles to other addresses and this can require care and is usually handled in software.

The second signal is RW. This indicates if the cycle is a read or a write using a single line unlike the 8080/Z80 bus where nRD and nRW may both be high to indicate no activity.

The Z80 style nRD and nWR signals are generated by combining the E clock with RW so that nRD or nWR goes low only when the bus state is valid.

As the Motorola bus has no notion of a separate I/O spaceso an I/O window is normally used. By convention this is at 0xFEXX because this address window is suitable for existing operating systems for these platforms and mirrors many historic machines. There is no requirement to use 0xFEXX as the I/O card will generally only decode the low 8-bits of the address bus anyway.

The two signals for controlling the cycle type on the Z80 bus are nMREQ for a memory request and nIORQ for an I/O request. These can be generated by decoding the upper bits of the address generated by the processor when the bus is valid.

The final 40-pin RCBus signal is nM1. This has no equivalent on Motorola bus processors as it is used as part of the Z80 interrupt decode not just as an indication of instruction start. The current cards pull this high so that the peripheral cards do not decode bus activity as a Z80 interrupt cycle.

15.3. Mapping The Extended Bus

The extended bus provides A23-A16, which are directly equivalent to A23-A16 on the 65C816 card.

The extended bus provides several signals that have no easy mapping. These are nBUSRQ, nBUSACK and nHALT. None of these signals are used by most peripheral cards except specialist cards such as the Z80 DMA interface.

The other two signals mostly map. The nNMI signal is equivalent to the nNMI signal on Motorola bus systems (called XIRQ on some processors). The nWAIT signal is near enough the same semantics as the Z80 one that it can be provided except on the 6309E/6809E which do not support clock stretching this way as they are intended to run synchronously with a SAM or similar device on the other half of the E cycle.

15.4. Additional Signals

Some Motorola bus peripherals are complicated (or near impossible) to operate without the Motorola bus signals. At other times it is just useful to reduce chip count to have access.

Existing Motorola bus processor cards can provide the E, RW and nFIRQ signals on bus pins. Jumpers should be used as the lines are intended to be available to the user for other purposes if desired.

```
37  | nFIRQ  | Open collector, pull up on CPU card
38  | E      | E clock
39  | RW     | RW signal from processor
```

Using these signals on a peripheral device makes the peripheral card incompatible with the basic RCBus. There is a trade-off between the convenience and simplicity of interfacing and the compatibility.

15.5. Notes

15.5.1. Reset

The reset signal on many classic RCBus boards is very poor. The original RC2014™ systems in particular lack a proper reset controller. The Motorola bus devices that need a clean reset (such as the 68HC11) should include their own reset controller to clean up the reset during power on.

15.5.2. Clock

The conventional RCBus clock was 7.372MHz. This is also conveniently a clock that generates good serial signals and a bit under 2MHz E clock for 63xx/68xx processors. There is no requirement to use this clock, but it does improve compatibility. For slower parts half this clock is similarly convenient.

The 6502 processor clock input and E clock are the same barring skew. This effectively means a 2MHz 6502 has the same timing requirements as the 7.37MHz Z80. Whilst the RCBus can be run with a high speed 65C02 or 65C816 part it will be necessary to use 74AHCT parts in general, and even then some of the standard boards such as the 512K/512K memory card will be too slow to go above about 8MHz.

15.5.3. Bus Hold

68xx and 63xx series devices have a bus hold time (the time that signals remain valid on the data and address bus after nWR rises) that is broadly compatible with the Z80 timings used on the bus. The "classic" 6502 and 65C02 parts likewise do. Modern 65C02 and 65C816 parts have almost no bus hold. On a backplane it becomes necessary for the processor card to cut the nWR signal early in order to produce a bus hold, otherwise many RCBus cards will not work.

15.5.4. Signal Bounce and Buffering

Some of the NMOS parts generate significant ground bounce when the address bus changes if they are driving a load with significant capacitance - such as an RCBus backplane. In these cases it may be wise to buffer the signals. Buffering signals from NMOS parts also improves compatibility with standard RCBus cards. This is not normally a problem when driving 74HCT series parts, but can be for driving other things directly (such as the CF adapter).

15.5.5. Memory Layout

The 63/68xx and 65xx parts require ROM to be present at the top rather than bottom of memory at boot. The classic 512/512K card provides this, whilst the standard 'flat' 512/512K cards have a jumper to switch the RAM/ROM over.