

RI

Conversion program

COLLABORATORS

	<i>TITLE :</i> RI		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RI	1
1.1	Overview of RI ZoneJoy Lib V2.5	1
1.2	RI ZoneJoy Lib V2.5	1
1.3	RI ZoneJoy Lib V2.5	1
1.4	RI ZoneJoy Lib V2.5	2
1.5	RI ZoneJoy Lib V2.5	2
1.6	RI ZoneJoy Lib V2.5	2
1.7	RI ZoneJoy Lib V2.5	3
1.8	RI ZoneJoy Lib V2.5	3
1.9	RI ZoneJoy Lib V2.5	4
1.10	RI ZoneJoy Lib V2.5	4
1.11	RI ZoneJoy Lib V2.5	4
1.12	RI ZoneJoy Lib V2.5	5
1.13	RI ZoneJoy Lib V2.5	5
1.14	RI ZoneJoy Lib V2.5	5
1.15	RI ZoneJoy Lib V2.5	6
1.16	RI ZoneJoy Lib V2.5	6
1.17	RI ZoneJoy Lib V2.5	6
1.18	RI ZoneJoy Lib V2.5	7
1.19	RI ZoneJoy Lib V2.5	7
1.20	RI ZoneJoy Lib V2.5	7
1.21	RI ZoneJoy Lib V2.5	8
1.22	Example Programs	9

Chapter 1

RI

1.1 Overview of RI ZoneJoy Lib V2.5

Overview

This library contains commands for setting up zones and testing the status of the joysticks attached to the Amiga.

New additions to this library allow you to have multiple lists of zones (referred to as zonetables in this doc). To maintain compatibility with older versions of the library, zonetable 0 is equivalent of the original list of zones used in the library. You cannot adjust the size of zonetable 0 (its size is 256 zones), nor can you delete it. The new zonetables can be from 1 to 65536 in size, there are 16 available zonetable numbers.

All commands that change or test zones will work on the last zonetable that was selected with the command UseZoneTable. The default table is number 0.

1.2 RI ZoneJoy Lib V2.5

Statement: ZoneInit

Modes : Amiga/Blitz

Syntax : ZoneInit [zone_num] |[start_zone,end_zone]

This command is used to clear any zones currently set. The optional parameters allow you to select either a single zone or a range of zones to reset.

1.3 RI ZoneJoy Lib V2.5

Statement/Function: Setzone

Modes : Amiga/Blitz

Syntax : Setzone zone#,x1,y1,radius
Setzone zone#,x1,y1,x2,y2

This command lets you set up zones for testing. The first version is used when you want to set up a circular zone and the second when you want a rectangular one. With rectangular zones, `x1,y1` should be the top left corner of the rectangle and `x2,y2` should be the bottom left.

If used as a function, this command returns TRUE or FALSE to say whether or not the change was made.

NOTES: The max zone number for zonetable 0 is 255.

A zone number outside the range of the current table will cause this command to abort.

Zones can be defined in any order.

Circular zones are used in exactly the same way as rectangular ones.

1.4 RI ZoneJoy Lib V2.5

Function: Zone

Modes : Amiga/Blitz

Syntax : `a.w=Zone(x,y)`

This command takes the co-ordinates `x,y` and checks to see if they are inside any of the defined zones. The zones are searched in order, starting at 0 and going through to the size of the zonetable-1. This command will return the first zone that the co-ordinates were found to be inside, you should note that both types of zones are tested (rectangular and circular).

This command returns either -1 for not inside a zone or the zone number.

1.5 RI ZoneJoy Lib V2.5

Function: ZoneTest

Modes : Amiga/Blitz

Syntax : `a.w=ZoneTest(start_num[,end_num],x,y)`

This command is the same as the Zone command except that it allows you to select either one individual zone to test or a range of zones. You should, though, ensure that `end_num` is greater than `start_num`.

This command returns either -1 for not inside a zone or the zone number.

1.6 RI ZoneJoy Lib V2.5

Function: ZoneTable

Modes : Amiga/Blitz

Syntax : `ad.l=ZoneTable`

This function returns the address in memory of the zone information storage area for the current zonetable. The zones are stored one after the other, with each zone taking up 8 words (16 bytes) in the data area, making a total size of 2048 bytes. They are stored in the following way:

```

Rectangular:    +0: x1
                +2: y1
                +4: x2
                +6: y2

Circular: +0: x1
          +2: y1
          +4: radius of zone
          +6: -1 <-- this is set to show that the
                zone is circular.

Undefined zone: +0: -1
                +2: -1
                +4: -1
                +6: -1

```

The first longword (4 bytes) of the zonetable is used to hold the size, in zones, of the table (thus the true size of the zonetable is 4+number of zones*8).

1.7 RI ZoneJoy Lib V2.5

Function: ZoneTableSize

 Modes : Amiga/Blitz
 Syntax : size.l=ZoneTableSize

This function returns the size, in zones, of the current zonetable. It is equivalent of doing: size.l=peek.l(ZoneTable).

1.8 RI ZoneJoy Lib V2.5

Statement/Function: NewZoneTable

 Modes : Amiga/Blitz
 Syntax : NewZoneTable table#,size

This command will attempt to allocate a new zonetable with the given table number. If the table already exists it will be deleted. The maximum size for a zonetable is 65536 zones. If used as a function, this command will return FALSE for failure or TRUE for success. You should note that all zones are automatically reset in the new table and that creating a table does not make it the current table, this must be done with UseZoneTable.

Valid zonetable numbers range from 0 to 15.

NOTE: you cannot define the size of zonetable 0 and you cannot use this

command to alter it in any way.

1.9 RI ZoneJoy Lib V2.5

Statement/Function: UseZoneTable

Modes : Amiga/Blitz
Syntax : UseZoneTable table#

This command is used to change the current zonetable to the selected one. If used as a function, it will return TRUE for success or FALSE for failure.

Valid zonetable numbers range from 0 to 15.

1.10 RI ZoneJoy Lib V2.5

Statement/Function: FreeZoneTable

Modes : Amiga/Blitz
Syntax : FreeZoneTable table#

This command is used to free a zonetable from memory. If used as a function, it will return TRUE or FALSE. When successfully called, this command will free the zonetable and change the currently used zonetable to table number 0.

Valid zonetable numbers range from 0 to 15.

NOTE: you cannot free zone table 0.

1.11 RI ZoneJoy Lib V2.5

Function: JFire

Modes : Amiga/Blitz
Syntax : jf.b=JFire(joy#)

This command tests the fire button status of the joystick joy#, where joy# is between 1 and 4. You should note that, as with all the joystick commands, joy#=1 refers to the Amiga's joystick port, joy#=2 refers to the mouse port, and joy#=3 or joy#=4 refer to the four player adapter ports.

This command returns the status of the 2 fire buttons for joy#'s 1 and 2 and the status of the 1 fire button for joy#'s 3 and 4.

0=No Fire Button Pressed
1=First Fire Button Pressed ONLY
2=Second Fire Button Pressed ONLY
3=First & Second Fire Button Pressed

1.12 RI ZoneJoy Lib V2.5

Function: JHoriz

Modes : Amiga/Blitz
Syntax : jh.b=JHoriz(joy#)

This command is used to test the horizontal direction of the selected joystick. It returns:

0: No horizontal direction
-1: Joystick left
1: Joystick right

1.13 RI ZoneJoy Lib V2.5

Function: JVert

Modes : Amiga/Blitz
Syntax : jv.b=JVert(joy#)

This command is used to test the vertical direction of the selected joystick. It returns:

0: No vertical direction
-1: Joystick up
1: Joystick down

1.14 RI ZoneJoy Lib V2.5

Function: AllFire

Modes : Amiga/Blitz
Syntax : af.b=AllFire [(bit_pattern)]

This command is used to test the fire button status of all four joysticks. It returns a byte with the first four bits giving the joystick status, false=fire button not pressed, true=fire button pressed. The following bits belong to joysticks:

bit 0: joystick 1 (joystick port)
bit 1: joystick 2 (mouse port)
bit 2: joystick 3 (four player adaptor)
bit 3: joystick 4 (four player adaptor)

The optional bit pattern can be used to restrict the testing of the fire buttons. If a bit in the pattern is clear (false) then the joystick it belongs to will not have its fire button tested,

e.g.

AllFire (%0011) will test joysticks 1 and 2 and return the result.

It will return false for joysticks 3 and 4.

1.15 RI ZoneJoy Lib V2.5

Statement/Function: JAdaptorStatus

Modes : Amiga/Blitz
Syntax : JAdaptorStatus On/Off
oldstatus=JAdaptorStatus(On/Off)

This command toggles the reading of the four player adaptor for the following commands:

AllFire
Jvert
JHoriz
JFire

When the status is off, these commands will return 0 when you attempt to read from joysticks 3 and 4. When on the testing will be performed normally. Default status for the adaptor is on.

1.16 RI ZoneJoy Lib V2.5

Function: GetZoneX1

Modes : Amiga/Blitz
Syntax : x1=GetZoneX1 (zone#)

This command returns the x start position for the specified zone in the currently used zone table. If the zone number supplied goes outside the size of the zonetable, then this command returns -1. It also returns -1 if the zone is undefined.

1.17 RI ZoneJoy Lib V2.5

Function: GetZoneY1

Modes : Amiga/Blitz
Syntax : y1=GetZoneY1 (zone#)

This command returns the y start position for the specified zone in the currently used zone table. If the zone number supplied goes outside the size of the zonetable, then this command returns -1. It also returns -1 if the zone is undefined.

1.18 RI ZoneJoy Lib V2.5

Function: GetZoneX2

Modes : Amiga/Blitz
Syntax : x2=GetZoneX2 (zone#)

This command returns the x end position for the specified zone in the currently used zone table. If the zone number supplied goes outside the size of the zonetable, then this command returns -1. It also returns -1 if the zone is undefined.

NOTE: For circular zones, this command will return the radius of the circle squared.

1.19 RI ZoneJoy Lib V2.5

Function: GetZoneY2

Modes : Amiga/Blitz
Syntax : y2=GetZoneY2 (zone#)

This command returns the y end position for the specified zone in the currently used zone table. If the zone number supplied goes outside the size of the zonetable, then this command returns -1. It also returns -1 if the zone is undefined.

NOTE: For circular zones this command will always return -1

1.20 RI ZoneJoy Lib V2.5

Function: ReadJoyPad

Modes : Amiga/Blitz
Syntax : value=ReadJoyPad(port#)

This command reads the extra buttons of a CD32 joypad. The port number is 1 if you wish to read the joystick port or 2 if you wish to read the mouse port.

The return value will be a bit pattern, with the following bits defined:

PAUSE	=%10
REWIND	=%100
FASTFW	=%1000
GREEN	=%10000
YELLOW	=%100000
RED	=%1000000
BLUE	=%10000000

Thus to test the pause button on the pad in the joystick port you would do:

Command Index

AllFire
FreeZoneTable
GetZoneX1
GetZoneX2
GetZoneY1
GetZoneY2
JAdaptorStatus
JFire
JHoriz
JVert
NewZoneTable
ReadJoyPad
Setzone
UseZoneTable
Zone
ZoneInit
ZoneTable
ZoneTableSize
ZoneTest
ZoneMode

ZoneTableNumber returns the number of the current zone table ↔
returns -1 for 68020 mode or 0 for 68000 mode

Examples

[Main Document](#)

[Library Index](#)

1.22 Example Programs

Example Programs

EXAMPLE 1 - Monkey Island Type Interface Using Zones & FNS :

Load Example 1
Compile It!

EXAMPLE 1 - Example Of Reading The Four Joystick Ports :

Load Example 1
Compile It!
