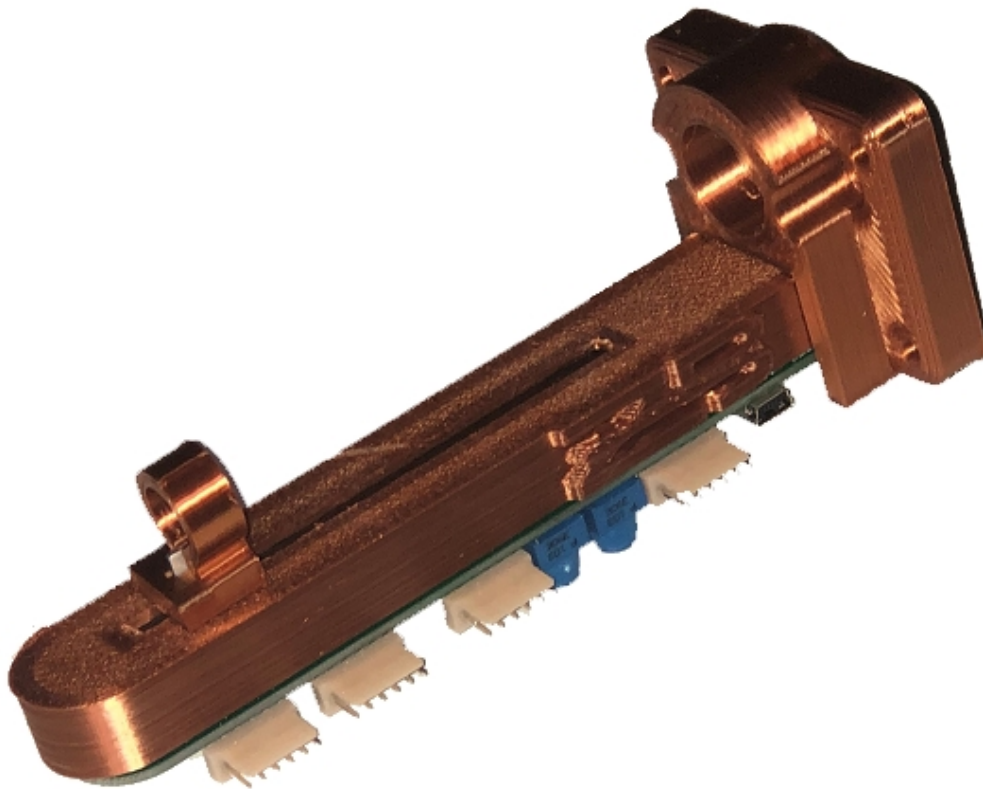




PLUNGER SERIAL CONTROL INSTRUCTIONS

Firmware Revisions **V4R53** and higher



PLUNGER KEYCODES

KEYPRESS	HEX CODE	KEYPRESS	HEX CODE	KEYPRESS	HEXCODE
LEFT_CTRL	0x80	0	0X30	a	0X61
LEFT_SHIFT	0x81	1	0X31	b	0X62
LEFT_ALT	0x82	2	0X32	c	0X63
LEFT_GUI	0x83	3	0X33	d	0X64
RIGHT_CTRL	0x84	4	0X34	e	0X65
RIGHT_SHIFT	0x85	5	0X35	f	0X66
RIGHT_ALT	0x86	6	0X36	g	0X67
RIGHT_GUI	0x87	7	0X37	h	0X68
UP_ARROW	0xDA	8	0X38	i	0X69
DOWN_ARROW	0xD9	9	0X39	j	0X6A
LEFT_ARROW	0xD8	A	0X41	k	0X6B
RIGHT_ARROW	0xD7	B	0X42	l	0X6C
BACKSPACE	0xB2	C	0X43	m	0X6D
TAB	0xB3	D	0X44	n	0X6E
RETURN (ENTER)	0xB0	E	0X45	o	0X6F
ESC	0xB1	F	0X46	p	0X70
INSERT	0XD1	G	0X47	q	0X71
DELETE	0XD4	H	0X48	r	0X72
PAGE_UP	0XD3	I	0X49	s	0X73
PAGE_DOWN	0XD6	J	0X4A	t	0X74
HOME	0XD2	K	0X4B	u	0X75
END	0XD5	L	0X4C	v	0X76
CAPS_LOCK	0XC1	M	0X4D	w	0X77
F1	0XC2	N	0X4E	x	0X78
F2	0XC3	O	0X4F	y	0X79
F3	0XC4	P	0X50	z	0X7A
F4	0XC5	Q	0X51	<	0X3C
F5	0XC6	R	0X52	>	0X3E
F6	0XC7	S	0X53	?	0X3F
F7	0XC8	T	0X54	:	0X3A
F8	0XC9	U	0X55	"	0X22
F9	0XCA	V	0X56	[0X5B
F10	0XCB	W	0X57]	0X5D
F11	0XCC	X	0X58	EQUALS	0X3D
F12	0XCD	Y	0X59	MINUS	0X2D
SPACE	0X20	Z	0X5A	PLUS	0X2B
				\	0X5C

DEFAULT FACTORY VALUES:

Gamepad Button 1	HEX 81	LEFT_SHIFT	L FLIPPER
Gamepad Button 2	HEX 80	LEFT_CTRL	L MSAVE
Gamepad Button 3	HEX 31	1	START
Gamepad Button 4	E	(DEFAULT 'q')	EXIT
Gamepad Button 5	HEX 32	2	EXTRA BALL BUY IN
Gamepad Button 6	HEX 35	5	COIN
Gamepad Button 7	C	(DEFAULT 'END')	COIN DOOR
Gamepad Button 8	HEX 3D	EQUALS	VOL DOWN
Gamepad Button 9	HEX 2D	MINUS	VOL UP
Gamepad Button 10	HEX 38	8	SERVICE MENU DOWN
Gamepad Button 11	HEX 39	9	SERVICE MENU UP
Gamepad Button 12	HEX 30	0	SERVICE MENU ENTER
Gamepad Button 13	HEX 37	7	SERVICE MENU EXIT
Gamepad Button 14	HEX 85	RIGHT_SHIFT	R FLIPPER
Gamepad Button 15	HEX 84	RIGHT_CTRL	R MSAVE
Gamepad Button 16	B	(DEFAULT 'ENTER')	LAUNCH BALL
Gamepad Button 17	A	(DEFAULT GAMEPAD 17)	TILT
Gamepad Button 18	HEX 0		
through to		NO KEYBOARD ASSIGNMENTS USED AS GAMEPAD BUTTONS	
Gamepad Button 32	HEX 0		

Launch Ball Key HEX **B0** Coin Door Key HEX **D5** Exit Key HEX **71** TILT Button **17**

SERIAL CONTROL COMMANDS:

Commands are used to control the operation of the plunger and consist of single character data. Commands are sent to the plunger by simple batch files and must be encapsulated in quotations in the command line. ***Changes in mode effected by commands take place immediately and other than TILT and RESET are not saved in memory.***

Example:

```
mode %PORT%:BAUD=115200 PARITY=N DATA=8 STOP=1 DTR=on RTS=on
```

```
set /p x="G" <nul>|. |.%PORT%
```

G – sets the plunger in Gamepad mode (buttons treated as gamepad button input)

K – sets the plunger in Keyboard/Gamepad mode (buttons treated as keyboard presses)

A – sets the plunger in Analog Plunger ball launch mode

D – sets the plunger into Digital Button ball launch mode

T – sets the onboard TILT ENABLE/DISABLE state. Sending a T to the plunger switches the onboard tilt routine to either ON or OFF depending on the current state. If onboard TILT is enabled, sending a T will disable. Sending another T will enable it again. The state change is saved in memory.

R – resets the plunger to the factory default values for keyboard codes and position indicators. *Requires plunger to be rebooted (usb unplugged and replugged in) after sending R command. Plunger calibration is NOT affected by using the reset command.*

SERIAL TRANSMISSION INITIALIZERS:

Initializers are used to notify the plunger of incoming serial data to be processed. Commands sent to the plunger must be prefaced with an initializer in order for the plunger to know how to deal with the data that follows.

Example:

```
mode %PORT%:BAUD=115200 PARITY=N DATA=8 STOP=1 DTR=on RTS=on
```

```
set /p x="CD5" <nul>|. |.%PORT%
```

In the example above the C is the initializer letting the plunger know that the D5 (shortened hex code of 0xD5 which is the END key) immediately following it is the keystroke identifier to be saved into the plunger's memory.

The initializers available for use are:

C – informs the plunger to save the next byte that follow as the keystroke that the COIN DOOR switch should use when in keyboard mode. By default the plunger ships with this being set to the END key. *set /p x="CD5" <nul>\\.\%PORT%*

E – informs the plunger to save the next byte that follow as the keystroke that the EXIT button should use when in keyboard mode. By default the plunger ships with this being set to the 'q' key. *set /p x="E71" <nul>\\.\%PORT%*

L – informs the plunger to save the next byte that follow as the keystroke that the LAUNCH BALL button should use when in keyboard mode. By default the plunger ships with this being set to the 'ENTER' key. *set /p x="LB0" <nul>\\.\%PORT%*

I – informs the plunger to save the next string of bytes that follow as the complete keypress layout and values that buttons should use when in keyboard mode. By default the plunger ships with this being set to the arrangement of values shown above in the Default Factory Values table.

An example of the usage of the I initializer appears as follows:

mode %PORT%:BAUD=115200 PARITY=N DATA=8 STOP=1 DTR=on RTS=on

set /p x="I81,80,31,E,32,35,C,3D,2D,38,39,30,37,85,84,B,A" <nul>\\.\%PORT%

In this example the 1st hex value following the I initializer (81) fills the position in the array for Gamepad Button 1 and if you reference it to the Plunger Keycodes table on page one of this manual you will see that (0x81) is the hex code for the Left_Shift key. The '0x' in the hex key from the table is dropped and only the remaining 2 characters (81) are required in the string.

Other than the value immediately following the initializer (the 81 following the I), the remaining hex values and position indicator letters MUST be separated by a comma. The (80) in the second position fills the position for Gamepad Button 2 and is the hex value for the LEFT_CTRL key and so on through the array until you get to the letter (A) which fills the position for Gamepad Button 17.

Plungers without the I/O expansion board have 18 available Gamepad Button positions that can be assigned, plungers with the expansion board can assign up to and including 32 positions. Only the positions that are assigned to keystrokes and/or position indicators (A,B,C,E) need to be filled with a value, positions not filled with key values will be filled with null (0) values by the plunger firmware and assigned as Gamepad Buttons. Using the example above, this would mean that Gamepad button 18 will be automatically assigned as a Gamepad Button since there are only 17 out of 18 available positions filled in the array (plunger without expansion board).

POSITION INDICATORS:

mode %PORT%:BAUD=115200 PARITY=N DATA=8 STOP=1 DTR=on RTS=on

set /p x="I81,80,31,E,32,35,C,3D,2D,38,39,30,37,85,84,B,A" <nul>\\.\%PORT%

In this example we see the position indicators E, C, B, A. Each of these indicates to the plunger which Gamepad button with special function is being used for which purpose. Since the Launch Ball button and Tilt button are used in several routines in the firmware the plunger needs a specific value that is used to indicate which Gamepad Button is being used for them. Similarly, since the Coin Door switch is coded as a single click on, single click off and requires it's own button code for that purpose the plunger needs to know which Gamepad button is being used for the Coin Door. The Exit key indicator is provided as an easy way to change the specific value for the EXIT button from the factory default to whichever is most convenient. An example of this would be switching the exit key to 'q' while using Visual Pinball and then switching it to ESC for use in Future Pinball.

The indicators are as follows:

A – TILT button - factory placed in the Button 17 position , moving to another position in the array changes button assignment from 17 to the button assignment of the new position in the array.

Example: *set /p x="I81,80,31,E,32,A,35,C,3D,2D,38,39,30,37,85,84,B" <nul>\\.\%PORT%*

In the above array the TILT BUTTON is now Gamepad Button 6 and LAUNCH BALL is Gamepad Button 17

B – LAUNCH BALL BUTTON – factory placed in the Button 16 position, moving to another position in the array changes button assignment from 16 to the button assignment of the new position in the array.

Example: *set /p x="I81,80,31,E,32,B,35,C,3D,2D,38,39,30,37,85,84,A" <nul>\\.\%PORT%*

In the above array the LAUNCH BALL BUTTON is now Gamepad Button 6

C – COIN DOOR BUTTON – factory placed in the Button 7 position, moving to another position in the array changes button assignment from 7 to the button assignment of the new position in the array.

Example: *set /p x="I81,80,31,E,32,C,35,3D,2D,38,39,30,37,85,84,B,A" <nul>\\.\%PORT%*

In the above array the COIN DOOR BUTTON is now Gamepad Button 6

E – EXIT BUTTON – factory placed in the Button 4 position, moving to another position in the array changes button assignment from 4 to the button assignment of the new position in the array.

Example: `set /p x="I81,80,31,32,C,E,35,3D,2D,38,39,30,37,85,84,B,A" <nul>\\.\%PORT%`

In the above array the EXIT BUTTON is now Gamepad Button 6

Changes made to the key value array and sent to the plunger take effect immediately and are written to the plunger memory.

BATCH FILES:

Batch files should follow the format illustrated below and be saved with the .bat extension.

The following 5 lines of the batch file would be saved as 'ZBGP.bat'

@echo off

REM Change COM port in following line to match what windows device manager shows for the plunger

SET PORT=COM24

mode %PORT%:BAUD=115200 PARITY=N DATA=8 STOP=1 DTR=on RTS=on

set /p x="G" <nul>\\.\%PORT%

The com port number for the plunger can be found by right clicking on the plunger in the view devices screen of the windows control panel and clicking on the Hardware tab.

VBS FILES:

VBS (Visual Basic Script) files can be helpful in running commands from the table script or used to run batch files in hidden windows. The hidden window script consists of one line as below and needs to be saved with the .vbs file extension. As well, the vbs script should be in the same directory on the hard drive as the batch file it calls and named similarly. The vbs file for the above batch file would then be saved as 'ZBGP.vbs'

CreateObject("Wscript.Shell").Run "ZBGP.bat",0,True