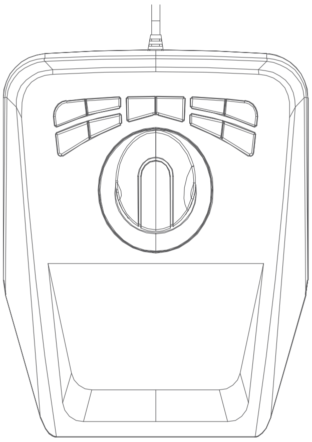
Afidus

# Afidus Joystick J782UKB



www.afidus.com

User's Manual
2017 July.

## 1. 1 OEM Mapper Overview

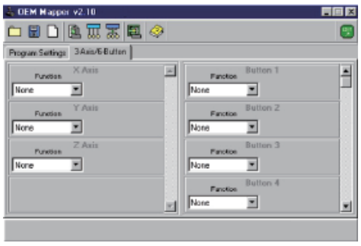
The OEM Mapper is a utility program which allows you to program the various DS OEM Joystick devices to control mouse, joystick, and character-based function. It consists of a User Interface program, DSMapper.exe, that allows you to define the actions that the various device are to perform. To execute these functions, it depends on the DS OEM Mapper Driver Subsystem.

The OEM Mapper uses a user-defined "map" file to define the relationship between the controls on the actual USB devices (axes, buttons, etc.) and the virtual system devices (mouse, keyboard, and joysticks) that carry out the appropriate functions.

The map file is created using the OEM Mapper GUI, CHMapper.exe. Once the map has been created, it is downloaded to the driver subsystem using the GUI and the desired control functions become available.

## 1. 2 OEM Mapper GUI

The DS Mapper.exe file contains the OEM Mapper GUI. The GUI is the key to programming and operating your DS USB devices under the OEM Mapper. An entry for this will have been created in the OEM Mapper Group when the Setup program was run. The GUI provides the functions necessary to create and edit user maps. This section gives a brief description of the various areas on the OEM Mapper GUI screen. The following sections will cover these areas in more detail. When you start the Mapper GUI, you'll see a screen that looks something like this:



X Axis Function: None
Y Axis Function: None
Z Axis Function: None
Button 1 Function: None
Button 2 Function: None
Button 3 Function: None
Button 4 Function: None

The screen is generally divided into three areas. At the top there is a Toolbar that controls the basic functions of the mapper GUI. Below that are a set of tabs that select which device you want to program.

Below these buttons are the "Device Tabs". There is one tab designated Program Settings which has to do with values associated with the program as a whole. Specifically, there are settings there for the Character Time and the Mouse Sensitivity. Additionally, there will be a tab for each supported DS USB device that the GUI detected on program startup. If no devices are detected, the GUI will display a dialog indicating that fact and then will exit.

## Programming Axes

As with the button functions, the various axes on the device can be programmed to operate in several different ways. based commands.

### Programming for No Action

To program the axis to perform no action at all, select "None" in the Function box for the axis.

### Programming for Mouse Control

Any axis can be designated to control either the system Mouse X, Mouse Mouse Y, or Mouse Z axis. The Mouse X and Mouse Y axes control the position of the cursor on the screen. The Mouse Z axis acts like a "Scroll Wheel".

### X and Y Axes

The Mouse X and Y axes can be programmed to operate in either of two modes. The first is "Normal" mode. In this mode, the axis is presumed to return to center when the control is released. When the axis is moved off center, the mouse cursor will move in the appropriate direction at a rate determined by how far the axis is moved. When the axis is returned to center, the cursor will stop at the current position. This approximates the same behavior as a normal mouse would have.

The second mode is "Direct". This mode is intended for use with the "clutch-type" joysticks to allow the mouse cursor to track the position of the joystick directly, i.e. moving the stick to its full forward and full left position will move the mouse cursor to the upper left corner of the display. While this mode does work with standard joysticks as well as the clutch -type, they are difficult to use as they always bring the cursor back to near the mid-screen position when they are released and allowed to return to center.

Note that in "Direct" mode, a normal mouse will still be seen and it will work okay, but the slightest movement of the programmed axis will cause the cursor to jump to the current position of the programmed axis. This can be a bit of trouble if the programmed axis is "jitt-ering" a bit as can happen if the axis happens to be set a position which is at a point very near to where the data being returned by the axis is changing value. If the returned data changes by even a single count, then the axis will continually jump to the programmed position using the standard mouse can prove difficult. A slight adjustment in the programmed axis to bring it to a stable value will normally stop this behavior.

### Z Axis

The Mouse Z axis acts as the scrolling wheel on a normal mouse and is always presumed to return to center when released. Moving the axis off center in one direction causes the scrolling to move one way, off center in the other direction causes the scrolling to move the other way. The scroll rate is controlled by how far off center the axis is moved.

### Setting the Mouse

To use the mouse, select "Mouse (Normal)" or "Mouse (Direct)" from the function box and a drop-down list will appear to the right from which you can select the X, Y, or (in "Normal" mode) Z mouse axes. If you're "Mouse (Direct)", only the X and Y axes will be available for selection. To program the Z axis, the desired axis must be set to "Mouse (Normal)".

### Programming for Joystick Control

You can designate any axis on the device to control an axis on a joystick in Game Controllers. Selecting Joystick in the Function field will cause a drop-down list to appear with the various axes that are available. If you have multiple devices connected, the joystick IDs that they use will correspond to the ordering of the devices as shown on the Device Tabs, with theleft-most tab defining axes and buttons for Joystick #1, then second left-most tab defining Joystick #2, etc.

### Programming an Axis to Send Characters

An axis can also be programmed to send characters. There are two possible modes that can be used: Up/Down programming and Position programming. These modes are explained below.

### Up/Down Programming

Up/Down programming is used where the axis is to send one particular character when it's moving in one direction, another particular character when it's moving in the other direction. An example of where this might be used where some software required a "+" key for an incremental increase, a "-" key for an incremental decrease. To use Up/Down programming, select that in the Function box. A box will appear that shows a string representation of the programmed function. It will look something like "S=10 U=x D=y C=z" which indicates that the axis is programmed to send 10 characters for full control travel, that the "Up" character is to be "x", the "Down" character is to be "y", and the "Center" character is to be "z".

The field itself is not editable, but clicking on it will pop up a dialog that allows you to enter the Step Count and the Up, Down, and Center characters

Note that a special case occurs when the step value is 1. In that case, the character will be held down continuously so long as the axis is off center. Normally, the Up/Down characters are sent as if they were pressed and immediately released.

A word of warning when using the single-step Up/Down controls. It's recommended that this type of programming not be applied to controls such as the throttle wheel which do not spring back to either a center or a minimum position when released. If they're left in a position other than these, the control will "hold the key down" and it can interfere with normal system operation. Also, be sure that the axis is properly calibrated before using this type of programming. An uncalibrated control can appear off-center or off-minimum even if the control is physically in the right position. Again, this can result in an apparent "stuck" key and interfere with normal operation.

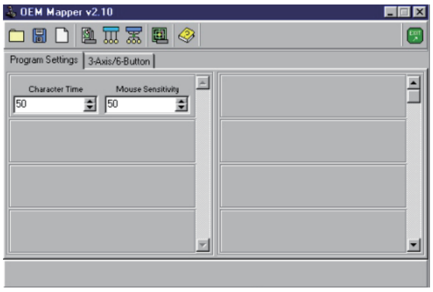
### Position Programming

Position programming is used where the control needs to send a set of particular character based on the position of the control. An example would be a throttle that used the keys0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 to set the throttle to one of 10 positions. The characters are simply entered into the edit box separated by spaces, i.e. " 0 1 2 3 4 5 6 7 8 9".

Entry of characters for axis programming follows the same basic rules as described for entry of button characters and the same conventions apply. Note that, for Up/Down programming, only one character can be assigned to each direction. The character may be uppercased or use the SHF, CTL, or ALT modifiers, though.

### Program Settings Dialog

The far left Device Tab is labeled "Program Settings". Selecting this tab, you will be presented with a screen that looks like this:



Character Time: 50
Mouse Sensitivity: 50

There are two edit boxes that labeled "Character Time" and "Mouse Sensitivity" that control a couple of parameters that apply to the map as a whole.

### Character Time

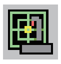
The Character Time is used to control how rapidly characters can be sent by the OEM Mapper. In some circumstances if the characters are sent too fast they will be missed. Increasing this value slows down the rate of character transmission and will generally get around this problem. The specified time is expressed in milliseconds (very approximate)

### Mouse Sensitivity

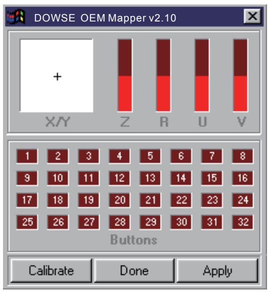
The Mouse Sensitivity setting governs how fast the mouse cursor will move for a given control movement when the Mouse device is assigned to one of the axes. It is used to adjust how fast the mouse cursor moves for a given axis position. The setting does not affect the normal system mouse at all. The default for the Mouse Sensitivity is 50%.

## Calibration from Within the OEM Mapper

To start the calibration from the OEM Mapper GUI, click the Calibration/Test Button on the Tool Bar. The Calibration Button looks like this:



When you click this button, you'll be taken to a dialog that lists the detected controllers so you can select the one you want to test or calibrate. Highlight the device you want to test or calibrate, then click "Okay". That will take you to the test screen. It looks like this:

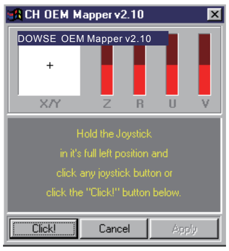


The test screen is fairly self-explanatory. Across the top are displays for up to 6 axes, below that is a display for up to 32 buttons. The actual number of axes and buttons which are displayed depends on the device that's currently being tested, so you will probably see less than what is shown in the above graphic.

The test screen is fairly self-explanatory. Across the top are displays for up to 6 axes, below that is a display for up to 32 buttons. The actual number of axes and buttons which are displayed depends on the device that's currently being tested, so you will probably see less than what is shown in the above graphic.

### Doing the Calibration

To begin the calibration procedure, click the "Calibrate" button at the bottom of the screen. The "Buttons" display will disappear and the screen will look like this:



Hold the Joystick in it's full left position and click any joystick button or click the "Click!" button below.

Follow the yellow prompts to complete the calibration. When you click the button, the next prompt will be displayed. When calibration has been completed, a message in green will come up telling you that things went properly. Click "Apply" or "Cancel" depending on whether you want to save the calibration or not. In either case you'll be returned to the Test Screen

### Calibrating from the Game Controller Applet

The procedure for testing and calibrating from the Windows Game Controllers applet is identical to that for the GUI. The same Test and Calibration applets will appear and the same procedure should be followed.

## Installing the OEM Mapper in Windows 98/2000/XP

Installation of the OEM Mapper in Windows 2000 and Windows XP is straightforward but YOU MUST BE LOGGED IN AS AN ADMINISTRATOR OR BELONG TO A GROUP THAT HAS ADMINISTRATIVE PRIVILEGES to install the OEM Mapper. This includes running the Setup program, installing the OEM Mapper Drivers, and and plugging the device in for the first time.

1. Unplug all your DS USB Devices.
2. Run the Setup.exe program from the distribution disk. This will copy the files and set up the folders and registry entries that the OEM Mapper requires and install the base drivers for the OEM Mapper.
3. Plug in the various DS USB devices you want to use.
4. Run the CHMapper.exe file that was installed in the DS OEM Mapper program group to calibrate and configure your DS USB devices.

## Appendix

### OEM Mapper Keycodes

The OEM Mapper recognizes most standard characters directly. Any key that can be entered as a single character ("a", "B", "%", "*", etc.) can be entered directly. If the key requires a SHF, that will be generated automatically when the shifted character is entered.

For other keys, a key name is used. This is a complete list of the key names recognized by the OEM Mapper.

### Keypad Keys

| Key | Name |
|---|---|
| Keypad - | KP- |
| Keypad . | KP. |
| Keypad * | KP* |
| Keypad / | KP/ |
| Keypad + | KP+ |
| Keypad 0 | KP0 |
| Keypad 1 | KP1 |
| Keypad 2 | KP2 |
| Keypad 3 | KP3 |
| Keypad 4 | KP4 |
| Keypad 5 | KP5 |
| Keypad 6 | KP6 |
| Keypad 7 | KP7 |
| Keypad 8 | KP8 |
| Keypad 9 | KP9 |
| Keypad Del | KPDEL |
| Keypad Down Arrow | KPDOWN |
| Keypad End | KPEND |
| Keypad Enter | KPENT |
| Keypad Home | KPHOME |
| Keypad Ins | KPINS |
| Keypad Left Arrow | KPLEFT |
| Keypad PgDn | KPPGDN |
| Keypad PgUp | KPPGUP |
| Keypad Right Arrow | KPRIGHT |
| Keypad Up Arrow | KPUP |

### Function Keys

| Key | Name |
|---|---|
| F1 | F1 |
| F2 | F2 |
| F3 | F3 |
| F4 | F4 |
| F5 | F5 |
| F6 | F6 |
| F7 | F7 |
| F8 | F8 |
| F9 | F9 |
| F10 | F10 |
| F11 | F11 |
| F12 | F12 |

### Main Keyboard

| Key | Name |
|---|---|
| Backspace | BKSPC |
| Caps Lock | CAPS |
| Enter | ENT |
| Escape | ESC |
| Left Alt | LALT |
| Left Control | LCTL |
| Left Shift | LSHF |
| NumLock | NUMLOCK |
| Right Alt | RALT |
| Right Control | RCTL |
| Right Shift | RSHF |
| Scroll Lock | SCRLK |
| Space | SPC |
| Tab | TAB |

### Auxiliary Pad Keys

| Key | Name |
|---|---|
| Keyboard Del | KBDEL |
| Keyboard Down Arrow | KBDOWN |
| Keyboard End | KBEND |
| Keyboard Home | KBHOME |
| Keyboard Ins | KBINS |
| Keyboard Left Arrow | KBLEFT |
| Keyboard PgDn | KBPGDN |
| Keyboard PgUp | KBPGUP |
| Keyboard Right Arrow | KBRIGHT |
| Keyboard Up Arrow | KBUP |