# Users Manual for the Wacom - Digitizerdriver for the Commodore *Amiga*

**Abstract**

The program enables you to use a pressure sensitive digitizertablet from Wacom with your Commodore *Amiga*. The tablet can be used together with the mouse (or without it) as direct input medium. Therefore the driver can be used together with (nearly) every software. Furthermore the (pressuresensitive) data provided by the tablet can be exported in custom applications using a documented softwareinterface. Under AmigaOS 3.0 or better the pressure data is also directly used by Intuition and passed to every window where an application can use the tabletdata.

# Contents

# 1 Introduction and legal stuff

The Program enables you to use a pressure sensitive digitizertablet from Wacom with your Commodore Amiga. The amigamousepointer is controlled by the driver for providing compatibility to nearly every program. The digitizertablet can be used either in *pressure* or in *normal* mode. Nearly all the (pressuresensitive) data provided by the tablet can be exported in custom applications using a documented softwareinterface. Under AmigaOS 3.0 or better the pressure data is also directly used by Intuition and passed to every window where an application can use the tabletdata.

## 1.1 legal stuff

The Wacom Driver is Shareware. If you want to use the program you should send - after you have tested the program for about 14 days - the registration fee ($20.00 or more) together with the filled registration form to the author. With your registration fee you are helping to develop (more) good software for the Amiga. The version you are holding in your hand is not crippled in any way. The programs in the `Develop` drawer are not Shareware. They are serving as examples for communicating with the driver in own applications. Together with their source they are Freeware.

The software and all principles described here were developed strictly using the Commodore Developer Guidelines. The package was extensively tested and is considered to be free of errors. But there is nobody on earth who is perfect. Therfore I *MUST* write the following lines. Read them carefully.

ATTENTION:
YOU ARE USING THE WACOM DRIVER ENTIRELY AND ONLY ON YOUR OWN RISK. THE AUTHOR IS NOT LIABLE AND CAN NOT BE PUNISHED IN ANY WAY FOR PROPABLY OCCURING DAMAGES ON YOUR HARD- OR SOFTWARE OR OTHER DAMAGES OR ERROS WHICH MAY RESULT FROM USING THE SOFTWARE OR ITS DOCUMENTATION. THIS PACKAGE IS PROVIDED "AS IS", AND THERE IS NO WARRANTY.

Your are allowed to give the package to another party at any time if you consider the following terms:

- The Wacom package may only be given away to another party in complete and unchanged form. You are not allowed to delete files nor to modify one or more files nor to add files to the package if you want to give it away. Compression of the package is allowed as long as all informations are remaining completely intact when decrompessing.
- The package may not be published on PD-Disks with a price greater than $5 (US). Exceptions are PD-CD-roms, their price may be higher.

- Commercial use or trade of the package is only allowed after given permission by the author. The Wacom Company is allowed to accomplish and distribute their tablets together with this package. This circumstance frees the person or company which buys the digitizer together with the package not from paying the Shareware fee.

### 1.2 Registering and the address of the author

If you want to use this program you must register it. The amount of the Shareware fee is 20.00 US$ (or 25.00 DM). Please do not send any other currencies. If you want to send the money via bank order you should mention your name and address on the bank order form. The directory of this document contains a registration form with the filename "Register-E.Txt". Please print it, fill it and send it together with the Shareware fee to:

```
Roland Schwingel
Lilienthalstrasse 9
92421 Schwandorf
GERMANY
```

If there are any questions or problems concerning the package you can reach the author also through Email and telephone:

```
Email: roland.schwingel@extern.uni-regensburg.de
Phone: +49-(0)9431-5779
```

### 1.3 Updates

The registrationfee also contains the mailing of the actual Version on disk. When there is a new version in preparation I will wait with sending until the new Version is finished. Updates are sent via Email for free. If you like an update via disk you should send US$8 (or 10DM) to the above adress for covering the costs for shipping and handling.

The actual version can also be obtained via anonymous FTP from various `Aminet` sites and will also be published on PD-Disk like the Fish-Series.

## 2 System requirements and Installation

### 2.1 System requirements

If you want to use the Wacom driver you will need at least Kickstart and Workbench 2.04. The package contains also a german catalog file which you can use on a WB2.1 and up system and enables you to receive all texts which are printed by the program in german. And at least you need a Wacom tablet.

These are the tablets which are currently supported by the driver:

- SD-013
- SD-113
- SD-210
- SD-310 SD-311 SD-312
- SD-320 SD-321 SD-322
- SD-420 SD-421 SD-422
- SD-510
- all digitizers of the UD-Series eg. ArtZ, UD-1212, ...
- all digitizers of the KT-Series eg. ArtPad

The digitizer should be configured as delivered by Wacom.

The driverprogramm runs with any CPU which is currently used with the Amiga. If you want to use the driver, a faster CPU is recommended because if you are running on 68000 the mousepointer is a *little bit* slow. The memoryusage of the driver is very low (about 45KB), it can therefore be used together with any other programm. Together with some mouse accelerators or "Sun Mouse" tools the mousepointer reacts a little bit wild. If you got problems steering the mousepointer with the tablet deactivate those tools.

## 2.2 Installation

Installation is very simple. The driverprogram should be copied together with its Icon in the `WBStartup` drawer on your bootpartition (or in the `WBStartup` Drawer in the rootdirectory of your booootdisk). There the program will automatically be activated every time you boot.

The absolute path for installing the driver should be:

`SYS:WBStartup`

If you are running under Workbench 2.1 or 3.x you can install the german catalogfile if your workbenchlanguage is "Deutsch". Just copy
`catalogs/deutsch/Wacom.catalog`
to
`LOCALE:catalogs/deutsch/Wacom.catalog`.

To make installation easier there is a `SHELL-Skript`, wich performs all these actions mentioned above. Just start "`Install`" from SHELL or Workbench.

The tablet must be connected to the serial port of your Amiga. If you got a Multi-IO board with serial ports you can also connect the tablet there. A scheme for a connector cable for the SD-series tablet is supported in the directory of this document for the reason that you are having no suitable cable. It is an IFF-ILBM picture called "Wacom-SD.ilbm". The scheme shows a cable for Amiga500/2000. The cable should also be suitable for A3000/4000/600/2000. Only A1000 Users have to modify the cable due to a different serial port connector. If you got an UD- or KT-tablet you have to use the delivered cable which also suits perfectly. (UD/KT-series digitizers are having the powerstrips inside the datacable.)

# 3 Usage and Configuration of the Driver

The Wacom Driver is a so called *Commodity*. Commodities are linked in the input data stream of the Amiga and are sending/filtering Inputevents.

The Driverporgram was developed for the `WBStartup` drawer of the WB *(WB = Workbench)* . There it is started whenever the Workbench is loaded. The driver can only be started once. When it is started for the second time and the first program is still active in memory the first program opens its configurationwindow and the second one quits immediately and silent. The driver can also be started from any other drawer from the Shell or from the WB. The program takes a couple of parameters from the user when it is started.

If you start it from WB the so called `TOOLTYPES` are used. These tooltypes are stored in the icon of the program.

## 3.1 The Icon Tooltypes

In most cases you need not change the tooltypes of the Wacom driver because it saves its settings on its own in the Tooltypes (look later chapters for more details). There is only one Tooltype (`INPUT_EVENT`) which is read, used and saved by the programm, but which can not be modified within the program.

The following Tooltypes are used:

**CX_PRIORITY (Default: *0*)** The priority of the commodity. Normally the default value of 0 is no subject of change.

**CX_POPUP (Default: *yes*)** If this tooltype is set to *yes* the configurationwindow is opened every time the program is started and you can easily configure the driver before the tablet is initialized. If the tooltype is set to *no* the driver will immediatley be started using the settings saved in the tooltypes. The configurationwindow can easily be opened via hotkey even if the driver is running.

**CX_POPKEY (Default: *lalt w*)** Here is the keycombination stored which is used as hotkey for opening the window when the commodity is active. Defaulted to left **Alt** key together with the key **w**.

**APPICON (Default: *no*)** The window can also be opened when the commodity is deactivated. If **APPICON = yes** there will be a wacomappicon displayed on the WB. If a doubleclick is performed on this appicon the configurationwindow opens.

**APPMENU (Default: *no*)** In the "Tools" menu on the WB an entry is created which opens the window everytime it is selected. (**APPMENU** must be set to **yes**)

**BEEPER (Default: *off*)** This field contains the state of the tabletbuzzer[1]. If set to **on** the buzzer will be enabled for a short time whenever a button on an input device is pressed.

**MODE (Default: *normal*)** Setting whether the tablet should be in pressure mode or not. If **MODE = pressure** the pressure sensitive mode is selected. You can use the so called "pressure stylus" in this mode. If **MODE = normal** the "standard stylus" or the cursor can be used for input. This Tooltype is only necessary for SD-Tablets and for UD-Tablets which are driven with the Wacom II-S Commandset.

**LMB_EMULATION (Default: *off*)** If the tablet is used in pressure mode there can no buttonstatus be transmitted to the driver. Only pressure data is supplied in this mode. Therefore it is normally impossible to hit any icons or gadgets with this pen. The driverprogram is able to emulate a click to the left mousebutton when a certain pressure limit is reached. If you want to use this feature you must set this tooltype to *on*. Only relevant when **MODE = pressure** and Wacom II-S commands are used.

**LMB_LIMIT (Default: *15*)** That is the limit which must be reached in pressure mode to emulate a LMB hit. Set **MODE = pressure** and **LMB_EMULATION = on**. Values for the limit must be between $-30$ and $+30$. This tooltype is also only necessary for SD-Tablets and for UD-Tablets in Wacom II-S mode.

**DEVICE (Default: *serial.device*)** Here is the name of the systems devicedriver stored which is used to communicate with the Wacom tablet. If the tablet is connected to a MultiIO board you must insert the name of the Multi-IO driver. If connected to the normal serial port this place should be filled with *serial.device*. (Please refer also to your MultiIO board documentation.)

[1] The buzzer is only available to tablets of the SD series.

**UNIT (Default: *0*)** The default value must only be changed when the digitizer is connected to a MultiIO-Board. Insert the number of the MultiIO-Board's serial port to which you have connected the tablet. (Please refer also to your MultiIO Board documentation.)

**MOUSE (Default: *FULL*)** The driverprogram enables the user to control the mousepointer with the digitizer. With help of this tooltype you can define which parts of the mouse will be emulated. There are 3 independent settings:

| | |
|---|---|
| **FULL** | The mousepointer and the 2 mousebuttons will be set by the program. |
| **POINTER** | Only the mouspointer will be set by the tablet. |
| **NONE** | No mousemulation. The data send by the tablet are only provided to the softwareinterface. |

**COMMANDSET (Default: *WACOM_IV*)** This tooltype is used to define which commandset is used for communication with the tablet. You can choose[2] between *WACOM_IV* and *WACOM_II-S*.

**AREA (Default: *BLANK*)** Determines size and location of the working area on the tablet.

| | |
|---|---|
| **FULL** | The whole working area on the tablet can be used for input. |
| **BLANK** | The same as *FULL* with the difference that there is a 4 millimeter sized border around the working area to make it easier to reach the border of a screen with the tablet. This border is also used to initiate the scrolling of an so called *Autoscroll*-Screen. |
| **CUSTOM** | A selfdefined working area is used for data input. Also look at the `CUSTOM_AREA`-Tooltypes. |

**MULTIMODE (Default: *OFF*)** The WacomIV commandset offers the capability of using 2 inputdevices[3] at the same time. You can switch this on or off using this tooltype. If you want to use multimode you must also use the Wacom IV commandset.

**MULTI_MOUSE (Default: *PEN*)** Because you can use two inputdevices at the same time when in multimode, you must determin which device is used as mouse. You can select between *PEN* and *DIGITIZER*. This has only an effect when the mouseemulation is also switched on.

**CUSTOM_AREA_LEFT (Default: *0*)**

[2] Attention: SD series tablets can not be driven in Wacom IV !
[3] Pen and Cursor. You need two different devices

**CUSTOM_AREA_TOP (Default: *0*)**

**CUSTOM_AREA_RIGHT (Default: *15239*)**

**CUSTOM_AREA_BOTTOM (Default: *15239*)** These 4 Tooltypes are describing the borders of the custom working area. Everything inside can be accessed. These Tooltypes are only used when **AREA = CUSTOM**. The marked Area must have at least a size of 1000 on 1000 points.

**INPUT_EVENT (Default: *TABLET*)** As mentioned above, this is the only setting which can not be changed when the driver is running. This Tooltype selects the Inputevents[4] that are used to set the mousepointer. You got two choices:

At first there is *TABLET*. All data is sent over an inputeventsubclass to the OS[5] as an inputevent from a digitizer. This also enables sending of pressure data.

Under AmigaOS 2.0 this tabletinterface is not completely implemented. This solution is not very fast and the pressure is currently not used inside the OS. Furthermore there are problems with autoscrolling screens.

When running AmigaOS 3.0 or better there is a different much better tabletinterface. The tablet data like the pressure are taken by the operating system and sent to every intuition window. All what the application has to do is to take the datas.

Dependend on your current OS Version the correct routines are choosen automatically.

The other possibility is *PIXEL*. The data will be send as a kind of mousedata to the OS. This solution is faster, but has also some disadvantages. There are problems with screendragging and a recalculating from tabletcoordinates to screencoordinates is necessary inside of the driver. Additionally there is no way to send pressuredata.

The cause for the fact that this setting can not be made from inside the program is, that this option is only for expert users. In the next version I will decide for only one method and than this tooltype will be removed. You should try out both settings and then choose the one you like most.

[4]Inputevents: Commands which are sent to the operating system (more exactly here: "input.device") for all kinds of input (eg. mouse, keyboard) in order to start the desired action.

[5]OS = operating system

**DONOTWAIT** This tooltype is not used by the tabletdriver. It is used by the WB if you start from the `WBStartup` drawer. The Wacom driver sets this tooltype automatically.

Once again I have to mention that you need not change the contents of the tooltypes manually. All settings (except from *INPUT_EVENT*) can be made from the configurationwindows. Your settings are saved here inside the tooltypes by the program automatically.

## 3.2 Start from the SHELL

If you are starting the driver from the SHELL the icon tooltypes are not read and passed to the program. You must therefore give all parameters to the program by writing them separated by spaces behind the program's name. If a parameter is equivalent to the default value you need not write it in the commandline.

Example for a SHELL start:

```
Wacom MOUSE=POINTER
```

## 3.3 Differences between the tablets

The Wacom driver supports three different kinds of tablets.

First there are the tablets of the SD series and on the other hand there are the UD- and KT-series digitizers. The last ones can do everything that SD tablets can do - with two exceptions. The UD series tablets are having no buzzer and no extra reset knob[6].

### 3.3.1 The SD Series tablets

For these digitizers there are existing 3 different kinds of input devices:

- Standard Stylus: This pen has two switches (one in the tip and one on the side) and is designed for drawing. There is no pressure transmission.
- Pressure Stylus: This pen has directly no switches, but a "soft" tip. Designed for transmitting pressure.
- Cross hair cursor: Designed for precise digitizing. It has 4 Buttons and transmitts no pressure.

Beside of third party tabletemulations this tablet features the Wacom II-S commandset.

[6]Isn't so necessary, because UD tablets are configured with help of a menubar and if a Reset is necessary the tablet will do so automatically.

### 3.3.2 The UD Series tablets

This digitizer uses only 2 different kinds of input devices:

- Stylus: This pen also has two switches (one in the tip and one on the side). The tip switch also sends pressuredata, therefore there is no need for an extra pressure pen.
- Cross hair cursor: There are two different kinds. One with 4 and one with 16 Buttons. These devices are also sending no pressure.

Beside of third party tabletemulations the Wacom II-S and the Wacom IV commandsets can be used. The Wacom IV commandset has some significant improvements:

- Greater pressure range: Wacom II-S Commandset features 60 steps. Wacom IV features 120 or 256[7] steps of pressure.
- Multimode: It is possible to use 2 (different) input devices at same time.
- Macrokeys: The UD tablets are having so called "Macro Buttons" outside the sensitive area. They can be used from software for own functions.
- Pressure characteritics can be changed[8].
- Many expansions that are making it easier for the programmer. Among these there is a more compact data representation which makes it faster to transform the received data. Additionally Wacom IV always works in pressure mode and button information is supplied at the same time.

### 3.3.3 The KT Series tablets

Currently there is only one tablet of this series available (ArtPad), so this section describes this tablet.

These tablets can do everything what UD series tablets can do, but with some limitations.

- This tablet can operate with Wacom II-S and Wacom IV commands.
- Like UD series tablets with a romversion 1.2 and up it has a double resolution[9] and a double pressurerange[10] compared to UD series tablets with a romversion lower than 1.2.
- There is no menubar.

[7] with Romversion 1.2 and up. You have also double density with these roms
[8] You choose between two settings.
[9] up to 2540 lpi
[10] 256 steps

- The only inputdevice available for this tablet is the pen. There is no digitizer, and therefore no multimode.

## 3.4 Controlling the mousepointer with the tablet

With activated driver the mousepointer can be controlled either with the mouse or the tablet (if mouseemulation is active). The whole selected working area on the tablet represents the area on your monitor which can be reached by the mouse. The top left edge corresponds to the top left edge on your monitor. The bottom right edge corresponds to the bottom right edge on the monitor. The buttons of the tablet's input devices are connected to following functions:

**Standard Stylus:** A click with the tip is equivalent to a click on the left mousebutton. The switch on the side of the pen reacts like the right mousebutton. This is also TRUE for the combined pen of the UD/KT series tablet in Wacom IV mode.

**Pressure Stylus:** [11] This pen can only be used when in pressure mode. The pen transmits the valid pressure between -30 and +30. You can, if you like to, enable the drivers LMB emulation capability. When this option is switched on a hit to the left mousebutton is emulated when a certain pressure is reached.

**Cursor:** This device has four buttons. Two of these buttons (the top button in the middle and the bottom button in the middle) are for free usage in own applications. The left and the right button are having the same functions like your mouse. If you got a 16 buttoned cursor the mouse funtions are mapped to the buttons 2 and 4.

## 3.5 The Mainwindow

The mainwindow will appear when at least one of these events occur:

- When the program is started and the tooltype **CX_POPUP** is set to **yes**.
- The Commodities Exchange[12] programm has sent an "appear" message to the driver.
- The hotkey was preesed. (only when Commodity is active)
- The appicon was doubleclicked. (only when the appicon option is enabled)
- The Wacom AppMenuItem was selected from the WB's "Tools" menu. (only when the appmenu option is enabled)

[11] only available with SD tablets.
[12] Commodities Exchange is a part of your Workbench disk.

- The program was started again even it was still active in memory.

The window is devided in four areas. For description the english localization is used. Most of the driverfuntions can also be accessed from keyboard too by hiting the corresponding underlined letter.

### 3.5.1 Info

The type and the romversion of the tablet are displayed here when the driver was activated succesfully.

### 3.5.2 Commodity Preferences

Here you can enter all commodity relevant data.

**Hotkey** Using this stringgadget you can enter the keycombination for the hotkey which opens the closed window when the commodity is activated. If you don't like the default hotkey just enter a new one. It will be activated immediately after you have entered it. For valid keyexpressions look into your Workbench manual in the commodities chapter.

**Popup** This cyclebutton determins whether the window should be opened upon program start or not.

**Pri** Priority of the commodity. Normally the default value (*0*) needn't be changed.

**Commodity** Only when the commodity is activated the mousepointer can be positionated using the tablet. Also opening the window using the hotkey is only available when the commodity is active. Everytime the cyclebutton switches from *deactivated* to *activated* the tablet is completely new initialized.

**AppIcon** If this checkbox is selected the window can also be opened using an appicon on the Workbench. Just doubleclick it and the window will open. The image for the appicon will be taken from the driver's icon. If there is no icon a build in one is used.

**AppMenu** When this gadget is selected the window can be opened selecting the *WacomII-S/IV Driver: Popup* menuitem from the workbench's tools-menu.

### 3.5.3 Digitizer Preferences

All settings concerning the tablet can be made here.

**Device** This stringgadget contains the name of the device which is used to communicate with the tablet. If you have connected the tablet to a multi-IO board you need to enter the name of the device for this board. If the tablet is connected to the normal serial port of the Amiga the name of the device should be *serial.device* (A change of this setting will only be applied to the tablet after the next initialization).

**Unit** The default value (*0*) must also be changed when the tablet is connected to a multi-IO board. You have to enter the unitnumber of the serial port you have the tablet connected to. Please refer to the boards manual (A change of this setting will only be applied to the tablet after the next initialization).

**Commandset** Selects the commandset which is used to communicate with the tablet. You can choose between *WACOM II-S* and *WACOM IV*. Wacom IV is only available with the UD/KT series and should be used if you got one. If you got a SD series tablet and want to run Wacom IV this will be detected while the initialization phase of the tablet and you will be notified to correct it.

**Multimode** The Wacom IV commandset enables you to use two different inputdevices at the same time. If you want to have this switch this on.

**Buzzer** If activated the internal tablet buzzer sounds when a button on an input device is pressed. Remember, only SD series tablets are having a buzzer (A change of this setting will only be applied to the tablet after the next initialization).

**Set Mouse** Determins which parts of a normal mouse will be emulated by the driver. You can choose *Pointer+Buttons*, *only Pointer* and *None*. If None is selected the incoming data from the tablet will only be available through the softwareinterface.

**Multi-Mouse** When multimode and mouseemulation are active you must have determined which of the two inputdevices should serve as mouse.

**Pressure-Mode** If this gadget is selected the tablet is put to pressuresensitive mode during the next initialization. You have to use the so called *pressure pen* in this mode if you are using a SD series tablet. If Wacom IV commandset is selected the tablet is automatically working pressuresensitve.

**Emulate LMB** When selected a hit to the left mousebutton is emulated when a certain pressure is reached if you are working with Wacom II commandset. When Wacom IV is used the parallel transmitted state of the buttons is used for mousebuttons. Therefore you can't select this in Wacom IV mode.

**Limit for LMB** Pressure which is at least needed to emulate a hit to the left mouse button. (value must be between −30 and +30.) Only selectable if Wacom II is active and LMB emulation is switched to on.

**Set Working Area** If selected a window is opened. Inside this window you can choose the customize the tablet's working area.

### 3.5.4 Controlsection

With these 3 gadgets the driver can either be quited or the window can be closed or the settings of the tablet can be saved to disk.

**Hide** Hitting this gadget closes all windows. If the tablet must be new initialized due to modified settings this is done here, too. The driver is not quited.

**Save** Causes the program to save the settings in the toolarray of its icon. If there is no icon an icon is generated. If the tablet must be new initialized due to modified settings this is done here, too.

**Quit** After verification of your choice the driver is removed and quited.

### 3.5.5 The Menus

With Version 1.16 the Wacom driver also has menus.

**About ...** Opens a little window with some informations concerning the driver.

**Save** Save the current settings to the icon.

**Hide** Closes the commodity window and initializes the tablet if necessary.

**Quit** Quits the program after request.

## 3.6 The Area Window

This window is used to define which parts of the sensitive tabletarea should be used for input. This window is structured in two areas.

### 3.6.1 Working Area

There are 3 possibilities:

**Full Tablet** The whole sensitive area of the tablet is used for input.

**Full Tablet, Borderblanking** Once again the whole sensitive area is used, but there will be a little (4 millimeters wide) border on each side of the tablet which is not directly used. This has two advantages:

1. It is easier to access the edges of a screen.
2. The border is used to initiate the scrolling of Autoscroll screens.

**Custom Area** A self defined workingarea is used for input. If this gadget is selected the gadgets for the custom area are enabled. If you leave again a few millimeters wide border the scrolling of autoscrollscreens can be initiated.

### 3.6.2 Customized Area

A custom area can be selected by defining the top left and the bottom right edge of the area. You can do this by entering the coordinates by hand or you can directly read them from tablet.

The last method is working even with activated and deactivated driver. The selected area must have at least a size of 1000 points in both directions. Reading of the coordinates from tablet can be finished by pressing a key on the tablet's inputdevice and aborted by pressing the escape key on the keyboard.

# 4 General notes

- After each initialization a buzzer sound from the tablet is heared. This is normal and shows the end of this process. (only digitizers of the SD-Series)
- Following changes of the settings are making it neccesarry to reconfigure the tablet:
  - Device
  - Unit
  - Buzzer
  - Pressure-Mode
  - Multimode
  - Every change within the definition of the working area.
- Together with some mouse accelerators or "Sun Mouse"-tools the mousepointer reacts a little bit wild. If you got problems steering the mousepointer with the tablet deactivate those tools. This problem occurs because some tools are not expecting inputevents from a tablet.
- Due to tolerances in the production of the pens some pressure pens could have a little bit greater range.
- After accidently hitting on the resetknob on the back of the tablet, it must be reinitialized. The same must be done after changing the configuration of UD series tablets with help of the menubar of the tablet.

- You are advised not to make changes to the configurationwindow using the tablet, because the data sets which are sent by the tablet while moving the pen or cursor can disturb the reconfiguration of the tablet. I have tried to avoid this possibility as good as possible (and in more than 99% of all cases you can operate the driver with the tablet without any problems), but you can never know.
- You mustn't configure your tablet to the commandset or other settings you want to run the driver with. All tabletconfiguration (commandset, pressure etc.) are made by the driver. No setting which you have saved to the tablet will be changed in tabletmemory. Only the current working mode is temporarily changed by the driver.

# 5 How to get tablet data in own programs

You want to write a paintprogram and you want to control the size of your tools according to the pressure the pen applies to the tablet? Then you should read this chapter because the Wacom driver supports a comfort interface to custom applications for direct data export.

This chapter only describes the builtin softwareinterface. The OS 3.0 tabletinterface is described in the Commodore developers kit. Furthermore there is a program with sourcecode in the drawer `Devlop` of the package that shows the usage of the OS 3.0 tabletinterface. It must be possible even for novice programmers to implement this interface.

## 5.1 Communication between tabletdriver and application

The Amiga operating system 2.0 offers only small support for pressure tablets. Therefore I had to find an own way to support these (pressure) data to custom applications. The Wacom driver uses the Amiga messageports for export.

Due to the fact that the driver is setting the mouse (if you like) you only have to take care for pressure data, even the softwareinterface offers much more than this.

Upon start the driver creates a public messageport which is used for data export. The Wacom driver package contains a drawer named `Develop`. There you can find the C-sourcecode and the compiled version of program called "`get_pressure`" which uses the softwareinterface of the driver in order to get the data. Among these files there is one called "`Wacom.h`" a C headerfile which you can use in own applications. This file contains all defines and structures you need for receiving the data from the driver. Before you include "`Wacom.h`" you should include "`<exec/exec.h>`"[13].

[13] exec/exec.h is a part of your C development package.

You can get the adress of the public messageport in the following way:

```
struct MsgPort     *Wacom_data_Port;
...
   /* The portname is also defined as WACOM_PORT_NAME in Wacom.h */
   Wacom_data_Port=FindPort("Wacom DataPort");
...
```

Now you have to create an own messageport and an own WacomMsg.

```
struct MsgPort     *Our_Port;
struct WacomMSG     WMess;        /* look Wacom.h */
...
   if(Our_Port=CreateMsgPort())
   {
    WMess.Mess.mn_Node.ln_Type=NT_MESSAGE;
    WMess.Mess.mn_Length=sizeof(struct WacomMSG);
    WMess.Mess.mn_ReplyPort=Our_Port;
   }
...
```

Now you can send the request for data to the Wacom driver:

```
   if(Wacom_data_Port)
   {
    /* send request */
    PutMsg(Wacom_data_Port,(struct Message *)&WMess);

    /* Wait for answer */
    WaitPort(Our_Port);    /* better Wait(1L<<Our_Port->mp_SigBit); */

    if(GetMsg(Our_Port))
    {
     printf("Current X-Coordinate:%ld\n",WMess.WacomInfo.Tablett_X);
     ...
    }
   }
```

Before you can quit your program you have to free all used resources.

## 5.2 The datastructures

The WacomInfo structure (which is part of the WacomMSG structure) containes beside of the pressure much more data. For detailed documentation look into "`Wacom.h`"[14].

[14]The file is printed in appendix A.

In the current version it is not possible to send commands to the Wacom driver using the messageport. This will propably be possible in a future version. Therefore the WacomInfo and the WacomMSG structure are containing some bytes wich are reserved for future use in order to preserve compatibility.

## 5.3 The fields of the WacomInfo structure

- `Sync` `BOOL`
  The presence of this field has only compatibility reasons. In version 1.0 this field was used to determin whether the mousepointer is already positioned at the moment the data package is transmited to the application. In Version 1.10 and up the data package will be transmitted to you after mousepositioning. Therefore this field can be ignored and has always `FALSE`.

- `Driver_running` `BOOL`
  Shows whether the driver is running or not. If this field is `TRUE` the driver is active.

- `Wacom_Series` `UBYTE`
  This field contains the used tablettype.

| Value | Tablettype |
|---|---|
| `0` | SD series tablet |
| `1` | UD/KT series tablet |

- `Beeper` `BOOL`
  If this field is set to `TRUE` the internal tabletbuzzer of SD tablets will be activated for a very short period whenever a button on a tablet's input device is pressed.

- `Mode` `BOOL`
  Is set to `TRUE` when the tablet is working pressuresensitive. If Wacom IV commandset is used the tablet is working automatically in pressure mode. If running Wacom II-S pressure mode must be selected manually.

- `MB_Emu` `BOOL`
  If running Wacom II-S in pressure mode, there is no button transmission of the pen possible. You can't hit any gadget or anything else with the tablet. Therefore a hit to the left mouse button is emulated when a certain pressure is reached and this field is set to `TRUE`. (only when mouse emulation is switched on). If running Wacom IV the parrallel transmitted button state is used for mousebuttons and this field contains `FALSE`.

- `Pressure_Limit` `integer`
  Contains the limit which must be reached by the pressure to cause a click to the left mousebutton. Holds a value between `-30` and `+30`. Only valid when in pressuresensitive Wacom II-S mode and `MB_Emu=TRUE`.

- `Pressure` `WORD`
  Contains the actual pressure when the tablet is working pressuresensitive. Otherwiese contains 0. With wacom driver 1.12 and lower this field was filled with `NO_PRESS` in this case.

- `input_device` `UBYTE`
  Shows the actually used input device (When in multimode shows the actual moved device).

| Bit | Inputdevice when set |
|---|---|
| 0 | Standard Stylus |
| 1 | Pressure Stylus (only SD tablets) |
| 2 | Cursor |

- `Buttons` `UBYTE`
  In dependency of the actual used input devices this field contains the number of the pressed button at that moment. When there is no button pressed this field contains `0`. If running in multimode it always shows the state of the pen's buttons, because the cursor has his own field when in multimode.

| input device | Button | Value |
|---|---|---|
| Standard Stylus | tip switch | 1 |
| | barrel switch | 2 |
| | both switches | 3 |
| Crosshaircursor (4 Buttons) | top button | 1 |
| | left button | 2 |
| | bottom button | 3 |
| | right button | 4 |
| Crosshaircursor (16 Buttons) | one of those buttons | <Number of the button> |

- `Tablett_Max_X` `ULONG`

- `Tablett_Max_Y` `ULONG`
  These fields are containing the maximum tabletresolution. But these values are depending on the selected working area. The selection and presentation is completly transparent. If you have selected a 6000 on 7000 points wide area for instance on a tablet with a physical resolution of 15420 points in both directions these fields are containing 6000 and 7000. If the working area is the whole tablet you will see here the physical tabletresolution.

- `Tablett_X` `long`

- `Tablett_Y` `long`
  Here you will see the actual position of the input device on the tablet. When in multimode you can find here the coordinates of the pen. The coordinates of the cursor are kept in own fields. The coordinates displayed here are in respect of the location and size of the selected working area. The origin is on the top left.

- `Set_Mouse` `UBYTE`
  Shows which parts of the mouse will be emulated by the mouse.

| Value | Emulation |
|---|---|
| 0 | no mouseemulation |
| 1 | mousepointer and mousebuttons |
| 2 | only mousepointer |

  Please remember, `MB_Emu` also depends upon this setting.

- `Commandset` `UBYTE`
  Shows which commandset is used to communicate with the tablet.

| Value | Commanset |
|---|---|
| 0 | Wacom IV (not available with SD tablets) |
| 1 | Wacom II-S |

- `Multi_Mode` `BOOL`
  In multimode you can use two different input devices at the same time. Only available with Wacom IV commandset. When multimode is switched on this field contains `TRUE`.

- `Multi_Mouse` `UBYTE`
  When mouseemulation and multimode are switched on you will see here the input device which is used as mouse.

| Value | Mouse |
|---|---|
| 0 | Pen |
| 1 | Cursor |

- `Macrokey` `UBYTE`
  UD series tablets are having so called macro buttons which can be used for custom functions. When Wacom IV is used you will see here the number of macro button which is actually pressed (if there is one pressed) otherwise these field will contain `0`.

- `MM_Dig_Buttons` UBYTE
  In multimode this field contains the currently pressed buttons on the cursor [15].

- `MM_Dig_X` ULONG

- `MM_Dig_Y` ULONG
  In multimode are the actual coordinates of the cursor stored inside here [16].

- `Proximity` UBYTE
  Shows whether the inputdevice of the tablet is in proximity. You should only read data from this softwareinterface when this field contains `1`.

- `MM_Dig_Proximity` UBYTE
  The same as `Proximity`, but only for the digitizer in multimode.

The interactions and dependencies between the fields inside the WacomInfo structure can also be seen when you are running the `Get_Pressure` program parallel with the driver.

[15] also look at field `Buttons`.
[16] also look at the `Tablet_X` and `Tablet_Y` fields.

# A Wacom.h

Here are the contents of "'`Wacom.h`"'.

```
/*****************************************************************/
/* WACOM Digitizerdriver            (C) 1993-1994 by ROLAND SCHWINGEL */
/*****************************************************************/
/* Headerfile for receiving tabletdata from the driver           */
/* Revision 1.2                                                  */
/*****************************************************************/

/* DEFINES ******************************/
#define WACOM_PORT_NAME   "Wacom DataPort"     /* Name of the DataPort */
#define NO_PRESS          -64;                 /* No valid Pressure */

/* Structures ***************************/
struct Wacom
{
 UBYTE        Sync;            /* Synchronisation with subprocess */
                               /* (internal must be 0) */
 UBYTE        Driver_running;  /* Driver active or not */
                               /* 0 = inactive */
                               /* 1 = active */
 UBYTE        Wacom_Series;    /* 0 = SD-Series */
                               /* 1 = UD-Series */
 UBYTE        Beeper;          /* State of the Beeper */
                               /* 0 = off */
                               /* 1 = on */
 UBYTE        Mode;            /* Pressure or "normal" Mode */
                               /* 0 = normal */
                               /* 1 = pressure */
 UBYTE        MB_Emu;          /* Mousebuttonemul. in Pressure Mode */
                               /* 0 = inactive */
                               /* 1 = active */
 int          Pressure_Limit;  /* Limit for LMB-Emulation (-30 ... 30) */
 WORD         Pressure;        /* current Pressure in Pressure Mode */
                               /* WacomII-S: -30 ... 30 */
                               /* WacomIV:   -60 ... 60 */
 UBYTE        input_device;    /* Type of current Stylus/Digitzer */
                               /* 1 = standard Stylus */
                               /* 2 = pressure Stylus */
                               /* 4 = Digitizer */
 UBYTE        Buttons;         /* current input_device Buttons */
                               /* Standard Stylus: 1 = Frontbutton */
                               /*                  2 = Sidebutton */
                               /*                  3 = both */
                               /* Digitizer:       1 = top middle */
                               /*                  2 = left Button */
```

```
                              /*                  3 = bottom middle */
                              /*                  4 = right Button */
 ULONG        Tablett_Max_X;  /* Maximum X-Value on the Digitizer */
 ULONG        Tablett_Max_Y;  /* Maximum Y-Value on the Digitizer */
 long         Tablett_X;      /* current X-Coordinate */
                              /* when in WacomIV and Multimode: */
                              /* X-Coordinate of the pen */
 long         Tablett_Y;      /* current Y-Coordinate */
                              /* when in WacomIV and Multimode: */
                              /* Y-Coordinate of the pen */

 /* Additions: 10.02.1994 */
 UBYTE        Set_Mouse;      /* Mouseemulation */
                              /* 0 = off */
                              /* 1 = Mousepointer and Buttons */
                              /* 2 = only Mousepointer */
 UBYTE        Commandset;     /* Commandset actually used */
                              /* 0 = Wacom IV */
                              /* 1 = Wacom II-S */
 UBYTE        Multi_Mode;     /* Multimode on/off (only Wacom IV) */
                              /* 0 = off */
                              /* 1 = on */
 UBYTE        Multi_Mouse;    /* inputdevice used as mouse when in */
                              /* Multi_Mode (only WacomIV) */
                              /* 0 = pen */
                              /* 1 = Digitizer */
 UBYTE        Macrokey;       /* pressed Macrobutton (only Wacom IV) */
 UBYTE        MM_Dig_Buttons; /* pressed cursorbuttons in Multimode */
                              /* (only Wacom IV) */
 ULONG        MM_Dig_X;       /* X-Coordinate of cursor in Multimode */
                              /* (only Wacom IV) */
 ULONG        MM_Dig_Y;       /* Y-Coordinate of cursor in Multimode */
                              /* (only Wacom IV) */
 /* Additions: 20.03.1994 */
 UBYTE        Proximity;      /* proximitiy of the pointing device */
                              /* 0 = not in proximity*/
                              /* 1 = in proximity */

 UBYTE        MM_Dig_Proximity;/* digitizers proximity in multimode */
                              /* 0 = not in proximity */
                              /* 1 = in proximity */

 UBYTE        reserved[112];  /* for future Expansions */
};

/* Message Structure for getting the above for use with PutMsg() */
struct WacomMSG
```

```
{
 struct Message Mess;
 struct Wacom   WacomInfo;
 UBYTE          reserved[32];  /* For future Expansions */
};
```

# B History

These are the changes made to the driver in his history.

**Version 1.16** (07. November 1994)

- Now also supports the new KT series. (eg. ArtPad)
- UD tablets with a romversion 1.2 or greater can be driven with double resolution and double pressure range. This is also supported now.
- The mainwindow now also has menus.
- The cycle- and buttongadgets with keyboard access are highlited now when the corresponding key is pressed.
- The layout of checkboxgadget has changed. When running AmigaOS3.0 or better they are looking much better with bigger fonts.
- The requester which appears when the tablet is not switched on now has a retry-button for much easier reinitialising.
- Revised tabletrecognition. It is now more flexible when some of the internal tabletsettings (DIP-switches or memory settings) are not set to factory settings. But please set your tablet to factory defaults!
- Revised SD-510C handling. There have been problems with a certain ROM-version of this tablet.

**Version 1.12** (07. May 1994)

- Implementation of the Kickstart 3.0 tablethandling. Every Intuitionwindow can now receive pressure data directly from Intuition. This is only used when AmigaOS 3.0 or better is available in the system.
- Adapted to the actual Wacom IV specification from april 1994.
- simplified serial communication in order to improve compatibility to MultiIO cards that are having not the full commandset implemented.
- minor optimizations in the whole program.
- exampleprogram with source included to show usage of the AmigaOS 3.0 tablethandling.
- new compiled with SAS/C V6.51.

**Version 1.10** (21. February 1994):

- direct and complete support of the Wacom IV commandset including macro buttons, multimode and greater pressure range.
- second window for defining the working area.
- editable mouseemulation (includes switching off).
- enabled scrolling of autoscroll screens.
- multiselect with shift keys down is working now.
- expanded softwareinterface.
- optimized for lower cpu load.
- complete second mouseemulation.

**Version 1.0** (12. October 1993):

- Creation of the first version.

# C Trademarks

Amiga©, Kickstart©, Workbench© are trademarks of Commodore-Amiga, Inc.
Wacom©, Wacom II-S©, Wacom IV© are trademarks of Wacom Co., Ltd.