pckeyboard documentation

Sami Vaarala

COLLABORATORS

	TITLE : pckeyboard documentation		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Sami Vaarala	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

pckeyboard documentation

1.1 Pckeyboard - the documentation.

```
                            Pckeyboard documentation

This is the pckeyboard AmigaGuide documentation.  Choose one of
the topics available:


            Legal stuff
                  READ THIS.

            Introduction
                  What is it?

            Usage
                  How to use it?

            Problems
                  Problems.

            Future
                  Future.

            Hardware
                  Hardware docs.


          Pygmy Projects -- "We make it small!"
```

1.2 How does it work?

```
Objectives

We wanted to keep the Amiga hardware unmodified and keep the price as
low as possible.  This resulted in heavy software load, the pckeyboard
software is not especially fast.
```

Hardware

Pckeyboard adapter hooks into the parallel port, taking advantage of the
"hidden" serial port there. The adapter receives the signals from the
PC keyboard and outputs them into the Amiga parallel port. Hardware on
the adapter also allow Amiga to send data to PC keyboard.

Software

The software is a very big part of pckeyboard. The large size of the
program is due to the simple hardware, the software has to do a lot of
things that the hardware doesn't. Output from PC keyboard is not simple
either. (Nothing is simple when the word PC is mentioned ;)

1.3 Legal stuff.

DISCLAIMER

 We accept no responsibility for any kind of damage caused by
 the usage of this software or the hardware described here. Use the
 software and hardware entirely at your own risk.

DISTRIBUTION

 You may distribute this program freely, as long as the archive is kept
 intact. This means that all files must be present, and no files may
 be added to the package.

1.4 Introduction to pckeyboard

1. Description

"pckeyboard" is a program which allows you to connect one of those
nifty PC keyboards to your Amiga and avoid using the not-so-nifty Amiga
keyboard. This only works while you operate under the Amiga OS, however.

2. Reasons for existence

Well.. This fusion of hardware and software exists because we simply got
tired of using the truly inferior A1200 keyboard. To put it politely,
Amiga keyboards suck, atleast the new ones. PC keyboards are
very cheap compared to their quality and are widely available. If you
accidentally destroy your PC keyboard you haven't lost much, another
200 fim and you're back in business.

1.5 Usage

```
  You need the following things to run the program:
    - the hardware connected to the parallel port (naturally ;)
    - "pckeyboard.config" file placed in "ENVARC:pckeyboard".
      (in effect, the file will also be in "ENV:pckeyboard")

  Then you just need to type
    1> Run pckeyboard

  and the program should start.  A more convenient way is to run the program
  at some point in your startup-sequence with the output redirected to NIL.
  This way, the program will not need any Shell windows open to work.
    1> Run >NIL: pckeyboard

  (the command needs to be given before the shell window is opened)
```

1.6 Problems

```
CIA problems

At first the pckeyboard used a timer of CIA-B.  This made most stuff work
with the pckeyboard.  The only exception I could find was PlaySID which
required too many timers.

Then the software was fixed to use only the SR interrupt of CIA-B.  Now
even PlaySID works but a new problem arose.  Some (stupid) programs assume
that if they get timers, they can bang ICR all they want.  This is ofcourse
not true because there are other things in ICR besides timer controls --
SR is one of them.  So the new version failed with f.ex. Colonization because
the game just ran over the pckeyboard handler.

Probably the best solution would be to make the allocation of timers optional
in the commandline.  Or, the user could allocate/release one of the CIA-B
timers to make stupid programs work (some key combination perhaps).

Unfortunately there seems to be no really good solution because the problem
is in the other programs, not pckeyboard.

Demos and games

As you might guess, demos and games don't usually work with pckeyboard.  Not
even OS games since they usually hit the CIAs without shame.  Demos also
usually hit ICR so the pckeyboard is run over with many demos.  Remember
that originally this software/hardware was made for coding, not playing
games etc :).

Weird bugs

Sometimes pckeyboard just ceases to get interrupts and the program enters an
infinite wait.  We don't really know why this happens so watch out.  Don't
panic if this happens--just use the original keyboard to save your work and
restart the machine.  This effect usually happens if you leave the machine
untouched for a long time.. Never caught the bug in action though :).
```

1.7 Future

There is a lot to be done still. Currently you can't edit the pckeyboard
configuration file yourself (unless you want to hack it :). The file
contains repeat rates etc, which are adjusted to our taste--lightning
fast 8).

To Do:

```
  - config/keymap editor
  - advanced control window opened by a keystroke (with CIA control
    options etc)
```

1.8 Hardware

```
    ************
    *DISCLAIMER*
    ************
```

We take NO responsibility for any damage (any form of damage, might it be mental,
material or any kind of damage) direct or indirect caused by constructing and/or
using this hardware. If you don't know what you are doing, then please don't
make this hardware, because it might break your computer. DON'T plug this
hardware into any other computer than Amiga, since the parallel port might
(almost certainly) differ from the Amiga's parallel port.

IN OTHER WORDS:

```
  ******* YOU ARE MAKING EVERYTHING AT YOUR OWN RISK! ********
```

```
    *************
    * COPYRIGHT *
    *************
```

All copyrights of the hardware design owned by Jani Vaarala.
If you want to start to build and sell these adapters, you need my
written permission to do so. Software copyright for hardware access/control part
also owned by Jani Vaarala. Other parts of the software owned by Sami Vaarala.
Ofcourse producing this harware for own use is allowed. If you want to include
support for this keyboard hardware then contact me at this address.

Please contact me:

```
  (snail mail)

    Jani Vaarala
    Tellervontie 2 A 15
    90570 Oulu
    Finland

  (email)

    flame@stekt.oulu.fi
```

```
   (cellular)

     +358 40 540 8354
```

NOTE: DONT CONTACT ME IF YOU WANT ME TO BUILD AN ADAPTER FOR YOU! I DON'T
  HAVE TIME FOR THIS, BECAUSE OF MY STUDIES.

Hardware originally consisted of 5 chips (2* 8-bit shifters, counter,nand-chip
and an OC nand-chip). But I was told that Amiga parallel port has a
TTL/CMOS compatible 8-bit shift register (Thanks to Teemu Suikki!). So I dug
some info from hardware manual and decided to make a hardware using that
internal shifter. Now the hardware consists of 2 chips (counter and an OC nand
-chip) and 1 resistor. To get enough power for the keyboard I use joystick port
VCC and GND pins.

Parts list

```
  74LS191 TTL     (counter)
  74S03 TTL     (OC nand)          *1
  D-25    male      (parallel port)
  D-9     male      (joystick port)
  D-9     female     (joystick port)
  D-5     female     (keyboard)
  1 MOhm  resistor   (pull-up resistor)
```

*1 I used 74S03 because I could get few of the free from some junk boards.
  You can also use LS/F chips, *but* the 1 MOhm pull-up resistor might not
  do the job (you need to select correct resistor yourself). I heard that
  CMOS chips should work with 1 MOhm pull-up also.

Used Pins

```
  Parallel port
          pin no

        2   Data 0
        3   Data 1
        4   Data 2
            11   Busy  (data)
            12   Pout  (clk)

  Keyboard connector
        1   Clock
        2   Data
        3   NC
        4   Ground
        5   VCC (+5 V)
```

    NOTE! as you face the back of the keyboard connector
    (back meaning the side where you solder the wires to)
    the pins are from left to right: 1,4,2,5,3

```
                     ---------
   Game port
               7    VCC (+5 V)
        8     GND


   74LS191
        1     D1   preload counter value
        2     Q1   flip-flop out (not used)
        3     Q0   flip-flop out (not used)
        4     _CE  count enable
        5     _U/D  up/down count selector
        6     Q2   flip-flop out (not used)
        7     Q3   flip-flop out (not used)
        8     GND  ground
        9     D3   preload counter value
          10    D2   preload counter value
          11    _PL  parallel load
          12    TC   terminal count
          13    _RC  ripple clock (not used)
          14    CP   clock
          15    D0   preload counter value
          16    VCC +5 V


   74S03     1    A input A(1)
        2     B    input B(1)
        3     Y output (1)
        4     A input A(2)
        5     B input B(2)
        6     Y output (2)
        7     GND ground
        8     Y output (3)
        9     B input B(3)
          10    A input A(3)
          11    Y output (4)
          12    B input B(4)
          13    A input A(4)
          14    VCC +5 V


Connection list
---------------


PAR_D0 (pin 2)     <-> NAND_1   (pin 1) \
NAND_1 (pin 1)     <-> NAND_2   (pin 2) /
NAND_3 (pin 3)     <-> KBD_CLK  (pin 1)


PAR_D1 (pin 2)     <-> CNT_PL   (pin 11)


PAR_D2 (pin 4)          <-> NAND_9   (pin 9)
           |
GAME_VCC (pin 7)  <-> 1 MOhm resistor   /


CNT_TC (pin 12)    <-> NAND_4   (pin 4) \
NAND_4 (pin 4)     <-> NAND_5   (pin 5) /
NAND_6 (pin 6)     <-> KBD_CLK  (pin 1)


KBD_CLK (pin 1)    <-> NAND_13  (pin 13) \
NAND_13 (pin 13)   <-> NAND_12  (pin 12) /
```

```
NAND_11 (pin 11)  <-> NAND_10  (pin 10)

NAND_8 (pin 8)    <-> PAR_POUT (pin 12)

KBD_DATA (pin 2)  <-> PAR_BUSY (pin 11)

GAME_GND (pin 8)  <-> NAND_GND (pin 7)
                            <-> CNT_GND  (pin 8)
      <-> CNT_D2   (pin 10)
      <-> CNT_CE   (pin 4)
      <-> KBD_GND  (pin 4)

GAME_VCC (pin 7)  <-> NAND_VCC (pin 14)
      <-> CNT_VCC  (pin 16)
      <-> CNT_D0   (pin 15)
      <->  CNT_D1   (pin 1)
      <-> CNT_D3   (pin 9)
      <-> CNT_U/D  (pin 5)
      <-> KBD_VCC  (pin 5)
```

In this package there is also a picture which might explain a lot
more than this text-form explanation (actually the picture is quite poor). If
someone is willing to make a better wiring list (maybe with better pictures,
etc), feel free to do so. If someone want's to make additions to this hardware
one may do so as long as the person doing so remembers that I (Jani Vaarala)
have the full copyright of this schematic and the new (maybe improved) schematic
may not be produced to be sold without my written permission to do so.

If you have any bug reports about this hardware, please contact in the
address(es) found in the copyright part.
BUT! DON'T mail if you want me to build this hardware for you, because I WON'T!

```
CNT_xx  = 74LS191 pin
PAR_xx  = parallel port pin
NAND_xx = 74S03 pin
KBD_xx  = keyboard pin
GAM_xx  = game port pin
8OC = Open Collector
```

--