

Contents

1	Introduction	2
2	ID3-Tag	3
3	Requirements	4
4	Let's start	5
4.1	Soldering at LCD	5
4.2	Doing the special cable	5
5	Software	8
5.1	Control Tools	8
5.2	ARexx Scripts	8
6	Source for parts	9
7	Copyright	10

Chapter 1

Introduction

This is a small manual how to connect a LCD (4 lines, 40 characters) to Amiga's internal parallel port. Up to now no other interface is supported. In near future there will be the alternative possibility to connect the display to Melody1200's control bus. This saves user from wasting the parallel interface.

As hopefully known AMPlifier, the most powerfull MPEG-Player at Amiga supports ID3-Tags and offers this information at its Arexx port. This project comes with a few lines of code in Arexx which may get modified for personal requirements. The script continously gets the data out of AMPlifier, formats it and moves it to the display by calling a programm done in C language (no source comes with it up to now).

Read on, closer informations follow...

Chapter 2

ID3-Tag

You know what's MPEG Audio? If you do not please refer to the manual of Melody1200 available at our page.

As MPEG files don't offer chunks to put information into some people added an info block at the end of the file. For sure this block is optional and it's not a valid ISO standard or something else.

ID3-Tag is an appended infoblock for or at MPEG audio files. Data like titel, interpret, style etc. may be stored into it. No question AMPlifier and the display can only show informations if there are some. As ID3 is optional it's the best idea to display the filename if the tag is not available. There are even tools for adding and modifying the tag. One comes with Melody1200 but there are more available in Aminet.

While MPEG Audio is streaming format even used for broadcasting the way ID3 was done is no good. It's not possible to get information from the song until it's over. Because of this people expect there will be another definition using chunks some time. Finally it doesn't care. Today AMPlifier supports the de facto standard and that's ID3-Tag. If there is something new, it will support it, too.

Chapter 3

Requirements

The list of things required:

- any Amiga computer except CD32
- a working environment for AMPlifier
- registered version of AMPlifier
- if no Melody1200PRO is used: 68060 or better
- short as possible parallel port cable (SubD-25male – SubD-25 female)
- LCD using the KS0066 (or equivalent) chip done by Samsung
- PCB connector 16 pins, 90 degrees
- short ribbon wire, 16 lines
- SubD-25 male for cable
- PCB connector for ribbon
- soldering iron (very easy job!)

Things recommended:

- Melody 1200 soundboard (PRO variant is best choice)
- 4 lines 40 characters display sold by PicVue(PVC400401A)
- any kind of housing
- EMI blocker for ribbon

Chapter 4

Let's start

4.1 Soldering at LCD

For first keep in mind the display is a very sensitive device and is even sensible regarding electrostatic discharge!

Soldering is limited to the PCB connector PicVue didn't put to the display. Additionally it might be nice to activate the background light. There are two pads for connecting 5 Volts at the opposite side of the control connector. Don't use the parallel port for current source. The display takes 500mA and will simply blow the fuse or resistor depending on the machine used. I inserted a stronger electronic fuse (some are used at A4000(T) and A3000T) to my A3000T. The usual way would be to get the current elsewhere. It's even possible to modify the angle of view (sometimes known as contrast). I didn't do it, though. Refer to the manual of the controller chip.

The pcb connector should be mounted at the upper side and soldered at the side of the chips. Otherwise the pinout is mirrored (I did it at the first prototype to save space). Pay attention for this and don't toast the display!

4.2 Doing the special cable

If an other display is used it's required to transform the interconnection list. As we didn't get a datasheet for our module we took a multimeter and looked for the internal connection of connector to chip. This is the hardcore solution, though. A printed datasheet is much better.

If a smaller display is used, E1 is missing. It's the enable line for the second controller chip. Each of them does 80 characters maximum.

If a very different controller chip is placed on the display you'll probably have to do a new C programm to write data from Amiga to it. Don't do so if you don't have to do.

The pinout of the connector soldered to the display we took (PVC400401A) is:

- Pin 1 : Data 7
- Pin 2 : Data 6

- Pin 3 : Data 5
- Pin 4 : Data 4
- Pin 5 : Data 3
- Pin 6 : Data 2
- Pin 7 : Data 1
- Pin 8 : Data 0
- Pin 9 : E1
- Pin10: R/W
- Pin11: Reset
- Pin12: Vee
- Pin13: GND
- Pin14: Vcc
- Pin15: E0
- Pin16: N.C.

For first it's probably best idea to take an easy multimeter to check connections before plugging it to Amigas's power!

Now it's time to take a knife and separate the lines of the ribbon wire and solder them to the right pins at the SubD. We payed attention connection at SubD is equivalent to the pinout for an LCD project in Aminet except the additional E1 pin, not available at smaller displays (LCD daemon(?)).

Here's the interconnection list for the cable:

- Line 1 : SubD 9
- Line 2 : SubD 8
- Line 3 : SubD 7
- Line 4 : SubD 6
- Line 5 : SubD 5
- Line 6 : SubD 4
- Line 7 : SubD 3
- Line 8 : SubD 2
- Line 9 : SubD 11
- Line10: SubD 18

- Line11: SubD 13
- Line12: SubD 19
- Line13: SubD 20
- Line14: SubD 14
- Line15: SubD 12
- Line16: SubD N.C.

Chapter 5

Software

5.1 Control Tools

There is a tool done in C (`displayinit`) which locks the parallel port and initializes the display. Another one writes a string of data to the display (`displaywrite`) and a third one dislocks it again (`displayfree`). The display is organised as 160 characters in line. It's up to the Arexx scripts to handle it correctly.

Next there will be the possibility to send out a command to the display (`displaydirect`) to control the cursor and don't have to rewrite the whole contents at every call.

No question that's a poor solution, but it was intended to do it as easy to understand as possible. We want everybody doing own scripts. If somebody like to do a better solution that's fine. The timing for the display was done with reads to `chipmem`. It's again not very nice, but a timebase of 2 microseconds is not possible to generate by software at clean way. If you don't intend to force a waste of resources and strong overhead you'll have to use this concept.

The stuff above is not true if the display is connected to Melody1200. There is only one tool required and access timing and mutual exclude is clean. Therefore better take a Melody1200! At the time of writing the interface for it is not ready, though.

5.2 ARexx Scripts

There is only one script in Arexx available now. It gets ID3 infos from AMPlifier formats them and moves them to display. Final version will even use AMPlifier for doing timing to show actual time.

Chapter 6

Source for parts

If you can't find the parts you may order them at the primary distributor of Melody, Gruner Buerotechnik (gruner@usa.net).

They don't have them on stock but it's quite easy for them to put your parts to the lists for other orders. It would take a while though and you have to ask for the price it's not fixed.

Ok, now you can order everything you need for Melody there, even high quality RCA cable and so on. But please keep in mind Gruner only offers components. If they would like to offer a certified product they wouldn't have to sell parts to do it. It's only a special service for people who are in contact to Kato Development Group. Therefore you have to write a mail to katodev@usa.net, too.

It's up to you to make the unit behave like written to the CE or FCC rules! If you can't ... simply don't use them!

Chapter 7

Copyright

As it's a free gift to you please follow the conditions. We did the design for you to have fun.

This project may not be sold in one or parts without written permission of Kato Development Group. Documentation may not be modified by 3rd party. Katodev is not responsible for any damage or things like that caused by using the documentation or software.