0795a0b0-0

Dick Diederik

COLLABORATORS			
	TITLE : 0795a0b0-0		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Dick Diederik	August 9, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

0795a0b0-0

1.1 PCFloppy2Amiga The Guide

```
          How to connect alien floppy drives to the amiga.

                         Second version

                    March 1995, Dick Diederik

    Introduction
    Important~changes since the first version
    Technical~background information
       Motor~switching
       Diskchange
       Ready signals
    What~tools~you~need
    Building~Tips

    Connect a 3.5~inch~~880~Kb disk drive
    Connect a 3.5~inch~1760~Kb disk drive
    Connect a 5.25~inch~880~Kb disk drive
    Connect a 5.25~inch~440~Kb disk drive

    Programs you can use to test
    Connections~&~Connectors
    Parts~list
    Problems
    Disclaimer
    History
    Author
```

1.2 Introduction

Introduction

This document is made for people who want to connect other then the tailor-made floppy disk drives to their amiga. I used amigaguide because you can look at the pictures and schemes with multiview and still use amigaguide. For the people who have seen the first version of this document look at the important~changes for a quick view of the changes.

There are two good reasons why you want to use the drives: The PC drives are a lot cheaper and with a great multitasking machine like the amiga an extra drive is always handy.

My first step on this path was using a 3.5 inch PC floppydrive as a second internal 880 kb drive for my Amiga 2000. Later on I made a external high density drive for her (the AMIGA) when I got kickstart 2.04. I have never built the other two described 5.25 inch floppydrives, but I wrote the information about them also in this document.

My friend Joop started the ball rolling by telling me maybe other people would like to see this information that I collected when making the drives. Since I am a whole lot of mails and other reactions wiser when I made the first version, I decided to make this second version as an amigaguide document.

I will describe the communication between the drive and the Amiga only if this is necessary for a good understanding of the system. All the connections who do not need any changes will remain unspoken for. (You will have to connect them though)

As an extra reminder I would to emphasize that there is upto now NO ONE who made the 1760 kb HD drive working perfectly (yes even I have read/write errors sometimes). See for details problems paragraph for HD drives. The drive still works fine in 880 kb mode. So the worst case scenario will leave you with a working 880 kb and a cripled 1760 kb drive. That is if you don't get the wires mixed up...

Happy Hacking

Dick Diederik (author)

1.3 Important changes

Important changes since the first version:

The main difference between this and the previous version is the use of open collector drivers instead of totempole drivers. When tried the interface on an other computer (I changed from my old A2000 to a A4000/030) I discovered that the interface had a totally different response. This explains why some people told me the interface would not signal right. This was caused by the use of totempole drivers instead of open collector drivers with READY and DISKCHANGE line.

When I was checking other documentation from a friend I discovered that the DISKCHANGE line works just opposite of what I described in the previous version. I now have include two versions of DISKCHANGE, one more system compatible and the old one.

1.4 Technical background

Technical background information

Picture: ~SYMBOLS~

Nearly all signal lines of the floppy drive cable are constructed with a low active state. This means that a not used a signal line is high (+5 Volt) and of course a active signal line is low (0 Volt).

The logic operators I used work as follows:

INVERTER simply inverts the signal.So 0 volt => 5 volt and 5 volt => 0 volt

IN	OUT
0	1
1	0

NAND use its two input signal performs an AND operation and inverts the result

IN	IN	OUT
0	0	1
0	1	1
1	0	1
1	1	0

AND performs an AND operation on both inputs (NOT used)

IN	IN	OUT
0	0	0
0	1	0
1	0	0
1	1	1

OR performs an OR operation on both inputs (NOT used)

IN	IN	OUT
0	0	0
0	1	1
1	0	1
1	1	1

FLIPFLOP something like a 1 bit ram cel.
D= Data line input
C= Clock, when this one changes value the data line is latched
Q= Output, the latched data signal is comes out here
/S= sets the Q output to 1 regardless of the data input until next clock cycle.
/C= Clears the Q output to 0 regardless of the data input until next clock cycle.
/Q= inverted signal of the Q output

DATA	CLOCK	/SET	/CLEAR	Q	/Q
0	_/	1	1	0	1
1	_/	1	1	1	0
X	X	0	1	1	0
X	X	1	0	0	1

_/ = rising signal (0 -> 5 volt)
X = don't care what

For the flipflops I used the 74S74 chips. They are triggered with a positive egded clock signal (thats why I invert for instance the SELECT signal) The SET and RESET (or CLEAR) signals of the flipflops are LOW ACTIVE ! As I already pointed out that nearly all drive signals are designed with a low active puls. However I can’t know if in your case the drive signals coming directly from the drive switches will act in the same way (like DISK PRESENT and HD switch). So before you start soldering, check out if they work the same as in my scheme. If not simple use an inverter or rewire the scheme to correct this.

In some cases I am left with an output of the flipflop to an open collector line. The outputs of a flipflop are totem-pole drivers so I have to change to the other inverted output of the flipflop and add an NAND gate (74LS38) of the open collector type.

1.5 About totem-pole drivers

Totem-pole drivers

A totem-pole driver has transistors connected from the output signal line to both power and ground. A totem-pole driver is at any time either connecting the output signal to high (5 Volts) or low (ground 0 Volts). If more than one totem-pole driver is driving a common signal line, and the drivers are trying to pull the line to different states (some high and some low), contention will result. Contention usually just results in invalid signal levels and associated random logic operation but, depending on severity, it might actually damage drivers.

This was the case with the chips I used in the first version of the interface because they use the totem-pole driver 74LS00 for output to DISKCHANGE and READY.

1.6 About open collector drivers

Open collector drivers

Several lines on floppy interfaces are Wire-OR’d. The term Wire-OR’d refers to multiple Open-Collector drivers all connected to a common signal wire. Open-Collector drivers have only one transistor from their signal output to ground. In this case Open-Collector drivers can thus only pull the output signal low and must rely on an external pull-up resistor (from the common signal wire to the supply voltage) to pull the signal high. If all Open-Collector drivers are off, the pull-up resistor will pull the signal line to a valid high. If any one of the Open-Collector drivers is on, the external pull-up resistor is overwhelmed and the signal will be a valid low. Thus, the term Wired-OR.

I have found value for the pull-up resistor ranging from 1 to 4.7 kohm. I think normally to 1 kohm value is the best fit. However higher values could be tried if the interface doesn’t work properly. You could even try omitting them from the scheme because there is always a pull-up resistor present in the computer. However I recommand using one.

If you use NAND gate from the open-collector 74LS38 chips internal in the interface do NOT forget to add a pull-up resistor to the output.

1.7 Drive motor logic

The logic behind the drive motor control.

The amiga has a different way of switching the motor then the way it is done in a PC. With a PC it is simply done by pulling the MOTORON line of the drive low. The disadvantage of this way is that all the drives will respond to this signal, causing a powerdrain of the floppy port. The amiga has its own elegant solution of avoiding this. With the falling signal of the SELECT line the signal of the MOTORON line (MTRXD amiga side) is saved in a flipflop. When at this time the MTRXD signal was low the motor will turn on (MOTORON to low) was the signal at that time the MTRXD signal high the motor will be turned off (MOTORON to high). When the SELECT signal is pulled to high again the motor will stay in the same state (on or off). In this way only the motor of the drive which received the SELECT line pulse will be affected. In a time diagram it looks like this:

```
SELECT    ~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~ amiga signal

MTRXD     ~~~~~|_____|~~~~~~~~~~~~~|_____|~~~~~~~~~~~~~~ amiga signal

MOTORON   ~~~~~~|_________|~~~~~~~~~|_________|~~~~~~~~~ signal to drive
```

Extra information: the drive reset line.

When booting the amiga the drive reset line will be pulled low to make sure the motors of all the connected drives are stopped. This is because only with the motor off the READY line of the drive will respond correctly to the drive SELECT line described under mounting a drive. This is done by connecting the drive reset line (DRESB) to the preset gate of the flipflop controlling the MOTORON signal (See for instance 3.5~inch~880~kb~drive scheme)

1.8 Diskchange logic

The Diskchange signal logic:

Pictures: ~SCHEME~ ~SYMBOLS~

The idea behind the diskchange signal is to keep the system aware of disk removing and inserting. When there is NO disk in the amiga drive it responds with a low signal on the diskchange line. This state is latched in a flipflop. Every time a STEP signal comes to the interface the signal of the DISKCHANGE of the interface is refreshed according to the state of the drive. So when a disk is put in to the drive and there has been a STEP pulse the flipflop is latched to HIGH. In logic terms this means connecting an inverted STEP signal to the clock input of the flipflop (74LS74 is triggered with a rising flank of the clock signal) and connecting the output (Q) to DISKCHANGE on the amiga side. You can get the signal from the drive by tapping in on the switch at the drive which pressed when a disk is in the drive or if your lucky on PIN 2 of the drive connector. Depending on the signal you get from the diskdrive you will have to invert or leave this and then connect it to the DATA input of the flipflop. Or you could changed the Q inverted output to Q output of the flipflop) When I am referring to the STEP signal it should be obvious that the correct DRIVE SELECT line is also needed to get it working for one drive specific. This is the first part of the scheme. It is NOT tested. In a time diagram it looks like this:

```
STEP        ~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~ amiga signal

DISK SWITCH ~~~|_____|~~~~~~~~~~~~~|_____|~~~~~~~~~~~~~~ drive signal

DSKCHNG     ~~~~|_________|~~~~~~~~~|_________|~~~~~~~~~ signal to amiga
```

In the first version of this document I proposed another scheme which is the second part of the scheme. It should also work fine (I never had any trouble with this but it doesn’t need a step impulse to check the state of the drive so it is not fully compatible with the Amiga standard. This version simple give through the present state of the drive when the select line is active. Again here things could be different when your drive has an inverted signal on the disk present switch.

You want to get this kind of operation:

DISK	DISKCHANGE
removed	0
present	1

I used NAND gates for this so I had to invert the select line and, in my case, invert the signal from the disk present switch of the drive. Then pass those two lines through a NAND gate. This will give you the following table:

SELECT	SWITCH	DISKCHANGE
0	0	0
0	1	1
1	0	1
1	1	1

You see that only if select is 0 (active) and the disk present switch is 0 (disk removed) the diskchange line will become low. When the switch returns a 1 (disk in drive) the diskchange line will become high. The diskchange line also stays high when there is no active select signal.

1.9 Ready line logic

Amiga ready line conventions

At first when you connect the PC drive to the amiga and boot the system the drive will not be recognized by the system. The amiga will pull the SELECT line (there four of them: DF0:-DF3:) of the drive down and will now be checking the READY line to see if there is a low pulse coming. If so there is a drive connected, else if it stays high no drive is mounted for that unit number. This checking is done when the drive motor is not spinning. The trouble with pc drives is that only when the motor is spinning at enough speed the READY line will react to the SELECT signal(because the drive IS ready). So this interaction has to be added to the interface. In a time diagram it looks like this:

```
SELECT  ~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~ amiga signal

READY   ~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~ signal from drive
```

This is the signal for a normal 3.5 inch 880 kb floppy drive. You should use this one also with the 880 kb 5.25 inch floppy drive.

The thing I didn't tell you that the amiga system isn't happy with testing the READY line once no, to be on the save side it does it only 32 times.

When the READY signals are put in a long integer it will give the following result:

```
  FFFFFFFF  = no drive connected (READY stays high, all the time)
  00000000  = double~density~disk in HD drive or double density drive
  AAAAAAAA  = high~density~disk in HD drive (READY alternating 1010....)
  55555555  = 5.25~inch~40~track~amiga~drive connected. (0101...)
```

I found these definitions disk.h in resources of the includes for 2.0. There is also a flag for DRIVE3_5_150RPM mentioned in trackdisk.h.

You can check out all four units for their ID with the DriveID program so you can see if your board is working correctly.

1.10 Building Tips

Building tips

The most easy way of adding a drive is an internal one (DF1:). In this case the most part will be done for you on the amiga motherboard when you install a jumper for DF1: (only with an amiga 2000 (j301), 3000 and 4000). Exception is the 3.5 inch HD drive). You will need to put in a jumper to the activate that mounting circuit. The location of it can be found in the documentation of your computer. This will also take care of the MOTORON signal and the drive reset line. The only thing you have to make between the drive and the amiga is a working DISKCHANGE signal. Even this can SOMETIMES be found on the drive (pin 2). For external drives you will have to make all the earlier mentioned signals yourself in an interface board which is best located in the drive casing between to cable coming from the amiga and the drive.

For people with much electronics equipment; the whole thing can be put in a GAL chip with exception of the extra logic for the HD drive. This requires two flipflops which asynchronous clock pulse which can be made in one GAL. So you should use a GAL with a 74LS74.

1.11 3.5 inch 88o kb drive

3.5 inch 880 kb drive

Pictures: ~SCHEME~ ~SYMBOLS~

This combines the standard logic of the READY line. Together with the DISKCHANGE logic and the MOTOR driver. Unless you will use it as an internal drive (see Building~tips)

1.12 3.5 inch 1760 kb drive

880/1760 3.5 inch drive

Pictures: ~SCHEME~ ~SYMBOLS~

For a High density drive to work with the amiga you need at least a kickstart 37.175 (2.0). There is however a bug only in the 37.175 kickstart. When you used the HD drive with a DD floppy and you change it back for a HD floppy you will still be in the DD mode so you will get a read error of the roottrack. The program HDFixer in aminet archive disk/misc/HDFixer.lzh will remove this bug. I haven't tried it because I have 2 other DD drives myself. A high density disk in the drive recognised by the system as follows: If there is a HD disk in the drive it will not pull the READY line low the second time the SELECT line is low. For every even puls on the SELECT line no READY puls is given. In a time diagram it looks like this:

```
SELECT  ~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~~|__|~~~~~ amiga signal

READY   ~~~~~~|__|~~~~~~~~~~~~~~~~|__|~~~~~~~~~~~~~~~ signal from drive
```

When the READY signals are put in a long integer it will give the following result:

```
  FFFFFFFF  = no drive connected (READY stays high, all the time)
  00000000  = double density disk in HD drive or double density drive
  AAAAAAAA  = high density disk in HD drive (READY alternating 1010....)
```

You can check out all four units for their ID with the DriveID program so you can see if your board is working correctly. Maybe you find out you have a HD drive already. Some guys are always lucky (or are they just stupid ?).

The signals described with the DD drive are all working in the same way as the HD drive. Exception is of course the drive ID or mounting signal. I generate the signal for HD drive by using a flipflop as you can see in figure 2. Remember that when you are making this an internal drive you can ignore the flipflop motor part. Also in case of a HD drive do NOT place the jumper on the amiga motherboard mentioned in the double density drive part or you will end up with a perfect running DD drive (No HD ID can come through if the motherboard constantly is pulling down the READY line. This is also the case for a genuine amiga HD drive, the FB 357 A or the slim one the FZ 357 A). If someone bought the chinon FB 357 A and he says it doesn't format HD disks, just tell him/her to get rid of the jumper for DF1.

In order to make the HD drive work you need also to halve the speed of the drive motor (300 rpm to 150 rpm). The motor of the drive is a step motor this means it needs a puls to turn the motor an angle further. Normally there is a quartz crystal and a IC on the print where the motor is. The IC is regulating the speed of the motor on the timebase of this crystal. You will have to remove the crystal from the board by first desoldering the two pins and then remove it (it will probably be glued to the board). Then install the crystal on your interface board (you left space for it ?). The board will have a oscillator~circuit on it so it works now there too. Then the signal is fed through a flipflop which will divide the signal by two, both signals will go through some NAND gates who are networked to change from the normal frequency to the halved frequency (from the flipflop) by using a tap on the HD switch of the drive (this can be different with your drive). It will be right under the HD hole in a high density disk (the one on the other side of the write protected hole). For the HD drive there are the scheme is shown in figure 2. You can see by the dotted areas that I use a NAND gate in combination with a inverter. I use this to get around less chips.

Be sure to check to problems~for~HD~drives.

1.13 5.25 inch 880 kb disk drive

5.25 inch 880 kb drive

Pictures: ~SCHEME~ ~SYMBOLS~

This drive is ment to show to the system exactly the same as the 3.5 inch 880 kb drive. So the scheme is the same. However it is sometimes not possible to get a signal from the drive about wether or not there is a disk in the drive. So there is no point in make the DISKCHANGE part of the interface of the 3.5 inch drive. You can use the diskchange <DEVICE> command in a shell to make the amigados system update its buffers.

The 5.25 inch drive was originally used as a 360 kb 40 track device, when the 1.2 mb version came out the number of tracks was doubled to 80 tracks and also the speed became 360 rpm instead of 300 rpm. For the old XT it was possible to switch a jumper on the drive for slow speed operation. I don't know anything about this. So try it out or simply keep away from this jumper untill you discover it is not working.

1.14 5.25 inch 440 kb

5.25 440 kb drive

Pictures: ~SCHEME~ ~SYMBOLS~

In fact this scheme is almost exactly like the HD drive one with the oscillator switching. Only the HD drive output to the ready line is 10.. and this one is 01... There is no diskchange signal available~see~880~kb~5.25~inch.

Also it is necessary to boot the system with no disk in the drive because then the drive is messing up the off/on bitpattern from the interface board by pulling the READY line low every time there is a SELECT signal. Should also work fine with crossdos reading 360 kb PC disks.

1.15 oscillator circuit

Oscillators (only HD drive)

I build two oscillator circuit which work with me. The schemes are here. I heard that some sony drives have an oscillator circuit that is not build into the motor driver chips. In that case you just have to cut the line somewhere and attach the osc in and osc out to the loose ends. If you don't know how to detect this here a simple clue: When there are two lines of the circuit with the crystal coming in to the motor driver chip then you have an internal circuit, When there is only one then it is external.

Someone told me there is a problem connecting the oscillator out signal back to the drive motor chip because when the crystal is removed there are two connections. Well.. Eeh. Just pick one with your eyes closed and pray. Just kidding ! Actually you can use both of them but NOT at the same time. The crystal was part of a oscillator circuit just like the one you made. So it will only work when there is a high enough voltage difference between those pins. I my case it didn't matter which one I choosed. You could try connecting the other pin to GROUND for eliminating disturbing signals but you could also weaken the frequency signal to much that way (I didn't use this). Try it out !

1.16 What tools you need

The kind of tools you need for this:

I developped the whole interface board with a solder iron and a multimeter.

I haven't made any board for this so don't bother asking me. If however you made one which seems to work fine send it to me and I will include it in this package. I have changed my expirimental board so much that when I point my soldering iron to it, it will probable desingrate. My board consists of an island print with wires soldered to make the connections. The wires run over IC so it is difficult for me to make changes.

1.17 Trouble shooting

No response from the drive

When all the wires are checked and correct. There could be a problem with the drive select line. Most drive have a jumper to be set on which select line the drive will respond. Maybe you connected the select line to drive select line 1 (pin 10) and the drive is set to react on select 0. Of course various possible situations can occur with this.

No good respons from the drive

There are a number of things which can be connected wrong when you make this interface. I found the best way to remove these errors is to test the interface with a battery and use LEDs (with a 200 ohm resistor) for checking the levels of the signal lines. The drivetest~program can help to also if the whole thing is already connected to the amiga.

You used NAND gate from the open collector 74LS38 chips for internal line in the interface and you forgot to add an pull up resistor between the output line and the 5 volt.

For problems with the 1760 kb HD drive see here.

1.18 Hd drive trouble

HD drive problems

Drive blocks data with HD disk

In HD mode the drive doesn't give any bits through to the amiga. It won't accept any data for writing a track either. This is because the drive knows the disk is not spinning at its normal speed. You have to fool the poor thing in thinking it is. This is simply done by glueing another small magnet or a small piece of magnet just on the opposite site of the original one on the flywheel. Now there are two index signals in one revolution so the drive chips think the motor is spinning at normal speed. It only checks if the motor is going too slow so it doesn't matter a thing that there is apparently twice the normal speed in DD mode. The amiga system can care less about index signals however for msdos disk there should be a problems because this system definitely uses it. However I haven't found any problems Yet ?? The index signal could be generated by an optical sensor, in that case you could drill an extra hole in the flywheel or mount another mirror.)
There is another way to deal with this. Sometimes the drive has an extra quartz cristal for its IO chips. If you build another frequency divider by 2 and switch it the same way like the one used for the drivemotor the chips would slow down to half the speed and thus think the motor is running at the right speed. I haven't tested this.

Read/write errors

Give the drive a first try with a already HD formatted disk if possible. Reading seems to be less critical then writing. As mentioned in the introduction I haven't come up yet with a perfectly working 1760 kb drive. This probably has to do with the fact that normal drives aren't made for half speed operations. Some people think the disk isn't turning smoothly enough within one rotation because of the use of a stepmotor. Others think this has to do with the higher writing current used with HD disks.

Oscillator

I am also not an expert in making oscillator circuits. I think the crystal is working at a frequency of about 1 Mhz. I don't know if my circuit has the same frequency as when the crystal was working on the drive board. You are also connecting an oscillating wire to another oscillator circuit on the drive board which could interfere with each other. Since I don't have an oscilloscoop I can't check this. If you are having trouble with the drive speed you could try removing some capacitors on the oscillator board of the drive. I also have recently found an alternative (up to 2Mhz) for the oscillator circuit on the interface board, this one will work with only one inverter so maybe that is just what you needed if you are short of one inverter.

1.19 Handy programs

Useful programs

I have made a CLI program to check the driveID for each unit number. It is described in READY~line conventions.

You can use the Drivetest program which you can find in the aminet archive in the directory hard/test (DriveTest.lha; 16095 bytes). It is a very basic program for testing the signal lines of floppy drives. I have included it in this package.

The other program DriveSpeed tells you how fast your drive is spinning in both DD and HD mode. This is usefull for people who don't have a scope or frequence counter (like me!). It is done by watching the INDEX signal and measure the time intervals get an average and recalculating them into revolutions per minute. However this program does not see if the speed is constant within a rotation of the disk. This is a question I still don't know the answer to in the case with HD disk with half rotation speed (150 rpm).

1.20 Pin layouts

Connections~&~Connectors

In this picture you can see the connections

Connections to PC drive (34 pins Shugart bus)

The connector for the power supply consists of 4 pins:

```
     pin 1:  5 volt
     pin 2:  Ground
     pin 3:  Ground
     pin 4:  12 volt
```

It is important not put this connector in the wrong way. You probably will wreck your drive !!! Normally it should be very difficult to do this, but I can assure you it can be done. Just try it out with the power off and see for yourself which way fits best. By the way I read that for the 5.25 inch drives with the bigger connectors pin 1 is 12 volt and pin 4 is 5 volt !!

I checked out some books and magazines to give you some detail on the drive connector. Only the even number connections are used for signals. The odd numbered pins are connected to ground.

PIN 2: DISKCHANGE Output signal. This is a tricky one to start with. It is used for a number of different signals by each manufactorer. The first one is the most widely spread used. DISKCHANGE is perfect to use for the amiga diskchange signal. HIGH DENSITY/DOUBLE DENSITY input signal to switch drive motor between 300 and 360 rpm. Only used for 5 1/4 inch floppy drives. IN USE input signal for letting the drive know something is coming up soon. Also used for driving led at front of the drive. HEAD LOAD input signal for the drive to put the head to the disk and pre magnetise it. (Rarely used)

PIN 4: IN USE Another pin use in more ways than one IN USE is the most likely candidate. see pin 2 (Sometimes you can have DISKCHANGE and HEAD LOAD). With internal Amiga drives IN USE is always connected to the MTRX signal. So when the motor is spinning the LED is on.

PIN 6: DRIVE SELECT 3 see PIN 10. Pin 6 is sometimes used for the READY signal (see PIN 34).

PIN 8: INDEX output signal will be low when the magnet on the flywheel of the drive passes the Hall sensor (when drive SELECT is low).

PIN 10: DRIVE SELECT 0 input signal for selecting the last drive which is then connected to this line. You will have to get the drive jumpers set to the drive select you want. Normally the drives are left with the jumper on SELECT 0 and the cable is twisted to change select lines (Yes that's why there is a twist).

PIN 12: DRIVE SELECT 1 see pin 10

PIN 14: DRIVE SELECT 2 see pin 10

PIN 16: MOTORON input signal switches to motor on. Sometimes the amiga way of switching on the motor with DRIVE SELECT can be set with a jumper on the drive.

PIN 18: DIRECTION SELECT Input signal. When low stepping direction of the head is to the axis (inside) of the disk. High is to the outside of the disk.

PIN 20: STEP Input signal. With every low pulse the head is stepped to the next (by pin 18 determined) track when the drive is selected.

PIN 22: WRITE DATA Input signal. Here the data to be written on the track is coming in from the computer.

PIN 24: WRITE GATE Input signal. When low the data from pin 22 is written on the disk.

PIN 26: TRACK 0 Output signal. When the head is above the first track this line will be low when SELECT is active.

PIN 28: WRITE PROTECT Output signal. When the write protect hole is open (disk is write protected) and SELECT is low this signal will be low.

PIN 30: READ DATA Output signal. When SELECT is low then the data on the track is written to this line.

PIN 32: SIDE 1 SELECT Input signal. When the SELECT is active this line set the head (upper or lower) on which side is selected.

PIN 34: READY Output signal. Low when SELECT is active and the disk is spinning at the right speed (INDEX signal) or, in a older simpler version, if a disk is in the drive. It tells the computer that the drive ready to do a read or a write of the track. The signal is high when the drive is empty (no disk), when the motor isn't running or the motor is running too slow. Also very shortly after a Head step signal it is high (15 ms). You see in this way the drive is not mounted by the amiga! (Rarely used as DISKCHANGE or IN USE).

```
 The amiga floppy connector ( external, 23 pins D-sub bus MALE)
 PIN  1: RDY READY
 PIN  2: DKRD read data
 PIN  3: GND ground
 PIN  4: GND ground
 PIN  5: GND ground
 PIN  6: GND ground
 PIN  7: GND ground
 PIN  8: MTRX motor
 PIN  9: SEL 2 select drive 2
 PIN 10: DRES drive reset
 PIN 11: CHNG diskchange
 PIN 12: +5 volt
 PIN 13: SIDE side (head) select
 PIN 14: WPRO write protect
 PIN 15: TK0 track 0
 PIN 16: DKWE write enable
 PIN 17: DKWD write data
```

```
 PIN 18: STEP step
 PIN 19: DIR step direction
 PIN 20: SEL3 select drive 3
 PIN 21: SEL1 select drive 1
 PIN 22: INDEX index
 PIN 23: +12 volt
```

1.21 Part list

```
 For this hack you will need the following:

PC Drives:
    3.5 inch 720 kb floppy drive      for 880~kb~drive
    3.5 inch 1.44 Mb floppy drive     for 880/1760~kb~drive
    5.25 inch 360 kb floppy drive     for 440~kb~drive
    5.25 inch 1.2 Mb floppy drive     for 880~kb~drive

Chips: see figure chip~layout
    74LS74  flipflop
    74LS00  NAND (Totempole)
    74LS38  NAND (open~collector)
    74LS04  inverter

Some pull-up~resistors (1 kOhm)

Capacitors: 10 uF and 0.01 uF at least 5 volt for buffering supply voltage.

Some extra resistors and capacitors for a oscillator~circuit

Connectors:
    23 pins sub D connector MALE
    20 wires in one cable (0.5 meter is enough)
    34 pins shugart bus connector FEMALE (the one on the
        flatband cable in the amiga)
    4 pins power supply connector FEMALE

Casing
```

1.22 History

```
History

Well there ain't much history. I made a version before this in plain text
format. You can find the changes here.

Why did I leave in this paragraph anyway ?
```

1.23 Disclaimer

DISCLAIMER

This document is provided "as-is" and the author accepts no responsibility for damage and/or loss of data/equipment as a result from building the interfaces described in this document.

1.24 About the Author

SYMBIOSIS consists of: (don't be shy, look it up)
Amiga 4000 with 420 Mb harddisk (IDE), 25 mhz 68030ec/68881, 10 Mb ram.
The other part:
35 years old human male listening to the name of Dick Diederik
(sorry no picture)

Email:
Dick.Diederik@Medew.ENTO.WAU.NL

Snail mail:
Dick Diederik
van Doesburglaan 32
6708 MC Wageningen
The Netherlands

Thanks to:
Keith Flaming for the info on open collector drivers
(Keith, I simply overlooked this part)

Joop van de Wege for his support in programming and
starting this thing up anyway.

Dennis and Gideon for their information about their progress
in making the drive.

and further anyone who send me reactions, suggestions, and support
(it really made me go on with this.)