

EclaireXL - Feature #73

Implement ATX disk image support

07/22/2018 08:27 AM - Farb

Status:	Closed	Start date:	07/22/2018
Priority:	High	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0:00 hour
Target version:			
Description			

History

#1 - 07/22/2018 08:29 AM - Farb

The SDrive Max project has ATX disk images working including many forms of timing-based protection. The code is written in C and is structured to make the Arduino-specific easy to replace. This can hopefully be ported to EclaireXL.

#2 - 07/22/2018 09:08 PM - foft

What is the basic interface?

If I receive a command packet, can I send it to the atx handling code and it returns me a packet to reply with and a delay? I guess it would need to call me back to read/write from/to the sd card too.

#3 - 07/22/2018 09:48 PM - Farb

Here is my current work-in-progress code that tries to emulate accurate disk timing:

<https://github.com/kbr-net/sdrive-max/blob/v08/atx.c>

The SD card is read using the `faccess_offset()` function. Data is stored in the global "extern unsigned char atari_sector_buffer²⁵⁶" variable.

As it stands now, the interface is:

1. Call `loadAtxFile(u08 drive)` to load an ATX file. This validates the ATX header and stores some metadata about the ATX including offsets to each track record for easier lookup later.
2. Call `loadAtxSector(u08 driveNum, u16 sectorNum, unsigned short *sectorSize, u08 *status)` each time sector data is needed. When this function successfully returns, the sector data will have been loaded into the `atari_sector_buffer` variable and the sector size and drive status will be in the two pointers passed as arguments.

An ATX file stores the angular position of each sector as a value between 1-26,042 with each increment representing 8 microseconds. The code is currently making use of an Arduino timer variable (TCNT1) that ticks every 4 microseconds and resets to zero when it hits 52,084 (2 * 26,042). So dividing TCNT1 by half effectively gives the position of the drive head on the track at any given time.

Having said that, I tried to keep this Arduino detail confined to two methods:

1. `void waitForAngularPosition(u16 pos)` is expected to delay until the angular position passed as an argument is reached.
2. `u16 getCurrentHeadPosition()` is expected to return the current head position at the time it is called.

So those two methods would need to be implemented using whatever hardware method is appropriate for the EclaireXL hardware.

I am also happy to refactor this code to make things easier to integrate with EclaireXL or assist in porting it.

#4 - 07/22/2018 09:50 PM - Farb

Here is my current work-in-progress code that tries to emulate accurate disk timing:

<https://github.com/kbr-net/sdrive-max/blob/v08/atx.c>

The SD card is read using the `faccess_offset()` function. Data is stored in the global "extern unsigned char atari_sector_buffer²⁵⁶" variable.

As it stands now, the interface is:

1. Call `loadAtxFile(u08 drive)` to load an ATX file. This validates the ATX header and stores some metadata about the ATX including offsets to each track record for easier lookup later.

2. Call loadAtxSector(u08 driveNum, u16 sectorNum, unsigned short *sectorSize, u08 *status) each time sector data is needed. When this function successfully returns, the sector data will have been loaded into the atari_sector_buffer variable and the sector size and drive status will be in the two pointers passed as arguments.

An ATX file stores the angular position of each sector as a value between 1-26,042 with each increment representing 8 microseconds. The code is currently making use of an Arduino timer variable (TCNT1) that ticks every 4 microseconds and resets to zero when it hits 52,084 (2 * 26,042). So dividing TCNT1 by half effectively gives the position of the drive head on the track at any given time. This handy timer is also used to simulate motor control to display when the drive is spinning and when it is not.

Having said that, I tried to keep this Arduino detail confined to two methods:

1. void waitForAngularPosition(u16 pos) is expected to delay until the angular position passed as an argument is reached.
2. u16 getCurrentHeadPosition() is expected to return the current head position at the time it is called.

So those two methods would need to be implemented using whatever hardware method is appropriate for the EclairXL hardware.

I am also happy to refactor this code to make things easier to integrate with EclairXL or assist in porting it.

#5 - 07/22/2018 10:05 PM - foft

Ah, that makes more sense. The loadAtxSector seemed to be missing all the timing I was expecting, then I saw the phantom sector hack comment:-)

There are a couple of registers available to the ZPU, one has the current time and one delays for a time.

zpu_pause and zpu_timer as defined in:

http://www.64kib.com/atarixl/fpga_svn/trunk/atari_800xl/firmware/regs.h

So I can implement waitForAngularPosition and getCurrentHeadPosition in terms of them.

Will have to point to you the svn, its currently on a branch. I need to merge it down but I've broken most of the other fpga boards on the branch...

#6 - 07/22/2018 10:07 PM - foft

Pleased to see it looks pretty simple, so we should be able to plug this in without too much pain.

#7 - 08/01/2018 10:04 PM - foft

- Priority changed from Normal to High

#8 - 08/12/2018 09:28 PM - foft

Made a start on this...

- Made atx.c/h build by pointing to different headers.
- Pointed delay_ms to existing wait_us. This is approximate based on a clock of 1.79MHz, will 1-2% error matter here? Will re-implement based on USB 12MHz clock just in case!
- faccess_offset implemented in terms of file_seek/file_read.
- rand implemented by read of a new (not yet implemented) zpu_rand reg. Will probably use pokey lfsr (at higher clock).
- waitForAngularPosition writes to new (not yet implemented) zpu_angle reg. Will pause until this angle hit, will increment angle at 8us using the USB 12MHz clock.
- getCurrentHeadPosition reads new (not yet implemented) zpu_angle reg.

Existing ZPU regs on review are:

zpu_pause, wait for this many atari clocks: Called with (num*230)>>7

zpu_timer, reads number of elapsed ms, based on atari clock rate of 1.79MHz.

Had enough for tonight, but probably I'll fix zpu_pause and zpu_timer to work in terms of microseconds and implement in terms of that. Rather than add this zpu_angle reg. I will probably add the zpu_rand reg though.

#9 - 08/12/2018 09:31 PM - foft

Also plumbed into the atari_drive_emulator.c in a similar fashion to the xex loader.

```
gAtxFile = file;
                                u08 status; // what use is this?
                                unsigned short ss;
                                if (!loadAtxSector(opendrive, sector, &ss, &status))
{
                                send_NACK();
                                break;
                                }
                                sectorSize = ss;
```

I left the standard delays in for ack etc, not sure if that will be a problem....

#10 - 08/12/2018 09:34 PM - foft

DELAY_T2_MIN (100uS)
ACK
loadAtxSector
(NAK if failed)
DELAY_T5_MIN (300uS)
CMPL
DELAY_T3_PERIPH (150uS)
send buffer
send checksum (Should I be recomputing this?)

#11 - 08/16/2018 09:46 PM - foft

Changed the timer/pause reg to be in microseconds and exact (USB clock/48 with clock domain crossing by toggle)
Added a register timer2, which has a threshold for reset. Using that to track the angle (in us)
Used the pokey 17-bit lsfr as a random variable (that ticks at 58MHz)

Firmware and hardware changed. Wonder if this will work...

#12 - 08/16/2018 09:58 PM - Farb

I'm not sure what DELAY_T2_MIN, DELAY_T5_MIN, etc. actually represent here so unfortunately I can't comment on that specifically. But I wanted to point out that one thing the SDrive Max's ATX code does not simulate (because a real serial transfer is happening) is the time it takes for the computer to send the serial command to the drive and the time it takes for the drive to send the data back to the computer. The Eclair code may not need to take that timing into consideration either but I felt it worth pointing out. Also, the Altirra Hardware Reference Manual has a nice write-up on drive timings that may prove helpful.

#13 - 08/16/2018 10:02 PM - foft

There are the default (original) sdrive delays in us iirc. If an ack is sent too soon it can arrive before the sio code is ready I think etc. Since the atx code does not appear to specify when to ack/cmpl I left them in.

#14 - 08/16/2018 10:03 PM - foft

I got the core building but I've broken it enough that usb no longer works... ooops, will have to debug another day.

#15 - 08/17/2018 09:27 PM - foft

Now the core is working again in general: USB, atr, xex. I can insert atx disks, but they fail to load - firmware hangs. Unfortunately debugging is a pain, might see if I can get the linux build Hias made working again...

#16 - 08/17/2018 09:33 PM - foft

If I comment loadAtxSector it does not hang, so must be something stuck in a loop in there. Will instrument and printf debug I guess :-)

#17 - 08/18/2018 09:52 PM - foft

Looks an an endian issue. I see AT8X ok, then version is 256...

#18 - 08/19/2018 08:44 PM - foft

Added byte swapping and changed the wait_us delays to be in us. For some reason wait_us(3.22*1000) was hanging rather than ending up as wait_us(3220).

Some files are loading.

#19 - 08/19/2018 09:04 PM - foft

Farb, I've put the latest test version here. Can you give it a try please since you are more familiar with how the beeps should sound/what current works?

http://www.64kib.com/autobuild/private/prototype_v2/atxv1.jic and http://www.64kib.com/autobuild/private/prototype_v2/atxv1.sof

+ my horrible hacked atx.c and atx.h...

#20 - 08/19/2018 09:30 PM - Farb

That is great news. I'm happy to test it out. How do I make an .rpd from the .jic and/or .sof?

#21 - 08/20/2018 07:51 AM - foft

Oh, forgot to copy that! Will do that this evening.

I was thinking a test suite would be great. On an atx that boots on the Atari then checks a few comment metrics/protection types. Do you know if there is anything like that at all yet?

#22 - 08/20/2018 05:11 PM - Farb

I am not aware of anything that exists like that today but think it is an excellent idea. I have reached out to some Atari folks that may be able to help in creating such an ATX :-)

#23 - 08/20/2018 06:30 PM - foft

I've put the rpd there now.

#24 - 08/20/2018 06:31 PM - foft

I think something might be wrong with track step or something. Since its loads slow, slow slow, then occasional fast. I'd expect fairly fast then pause for track step etc.

#25 - 08/20/2018 09:09 PM - foft

Seems that it was just a + vs - issue when calculating the wait on rotation. Doh.

#26 - 08/20/2018 09:34 PM - foft

Fixed that and put it up as atxv2.

Load a bit quicker, but still only a few things seems to work (baker street and airstrike so far...)

#27 - 08/20/2018 09:50 PM - foft

Trying to hook up sector status to 0x53 to see if that helps

#28 - 08/20/2018 09:51 PM - foft

Nope, still seems to hang on most things. Hmmm.

#29 - 08/20/2018 09:54 PM - foft

An aside, but noticed I seem to have a bunch of hexdumps still in this code. Better remove them since I'm sure they trash mem on some programs!

#30 - 08/20/2018 10:06 PM - Farb

- *File test-atx.7z added*

I just tried atxv2 as well.

It sounds to me like the "slow" parts are too slow and the "speed-ups" are too fast. I know, very scientific :-P If I had to guess from the sounds, it seems like revs are constantly being blown which is why things sound so choppy.

There are a few ATX files that I was using for testing with SDrive Max (I've attached them here):

Agent U.S.A. - fairly loose phantom sector timing
Bandits - straightforward CRC errors and missing sectors
Bruce Lee - stricter phantom sector timing
Goonies - weak bits
M.U.L.E. - CRC errors and phantom sector - and the music sounds triumphant when I hear it :-)

#31 - 08/20/2018 10:14 PM - Farb

I should add (in case it wasn't completely clear) that all those titles work with the current version of atx.c.

#32 - 08/21/2018 07:28 AM - Farb

- *File Scanalyzer, The v3.3 (1984)(Alpha Systems).zip added*

I did some further testing and something definitely seems quite wrong. I can understand timing problems causing duplicate sector or timing-based protection checks to fail, but it seems like everything I try is having problems.

Here's a test I used when first testing the ATX emulation code on a real Atari.

Try booting Scanalyzer (attached, needs BASIC) and use the "Disk Scanner" (option 4). Select your drive number and then do a fast scan (option 2). Choose 1 as the starting sector and 720 as the ending sector. In the EclairXL menu, replace the Scanalyzer disk with the Bandits ATX I provided above. Press Return to start the scan.

If the ATX code is working correctly, all sectors should read good except for:

19,20,21,22,23,24,25,27,29,30,31,33,35,36 should read ERR=144, STATUS=239
28,32 should read ERR=144, STATUS=247

When I try this test on the EclairXL, I get errors on the first several sectors which isn't correct behavior (I aborted after that).

Hopefully this assists in troubleshooting.

#33 - 08/21/2018 07:00 PM - foft

If it's that broken I wonder why air strike, Bruce Lee and lode runner work.

I'll try that tool...

#34 - 08/21/2018 08:26 PM - foft

I think the first few errors may be related to the bug 74 you raised.

I do see failure on those sectors though I was returning NAK instead of ERR and data. I'll try the latter.

#35 - 08/21/2018 08:33 PM - foft

OK, little better. Now getting 247 on 19,20,... and 28,32. Investigating why the status is not 239 now.

#36 - 08/21/2018 09:03 PM - Farb

Glad it is making progress. I did wonder if bug 74 had something to do with what I was seeing. Are you using the status byte that comes back from loadAtxSector?

#37 - 08/21/2018 09:05 PM - foft

Yeah I wired status up right after building atxv2. I am getting back 0xf7 though. Its not finding the sector in the loop, so it returns the default.

I'm getting the linux build working so I can debug more easily...

#38 - 08/21/2018 09:15 PM - foft

OK, can reproduce the same results in the linux sim... Which is good. So its not a missed byte swap (there is no byte swapping on x64).

#39 - 08/21/2018 09:17 PM - foft

OK, have to call it a day for today. I'll put up v3 with the latest anyway, though I suspect not much really helped yet...

#40 - 08/21/2018 11:27 PM - Farb

Ok, I tested v3 and saw a few things:

1. Agent U.S.A. and Goonies successfully loaded. Woohoo!
2. Bandits successfully booted when started in 48K Atari 400/800 mode but it causes a reboot when its tries to load the first level in demo mode. Interestingly, the reboot drops it into memo pad and it will not recognize a disk is mounted when cold resetting. Switching to XL/XE OS and doing a cold reset got it to at least start reading the mounted disk again (but crashes because it is OS-B only).
3. I could not get Bruce Lee to load beyond the initial text splash screen. My TV just goes black.
4. My Scanalyzer test described above still doesn't work at all. I get timeout noises immediately and Scanalyzer reports errors starting with the first sector. Interestingly, I tried doing a cold reset during the test (while Bandits was still mounted) and it wasn't loading properly.

#41 - 08/22/2018 06:22 AM - Farb

As you have already surmised, the status byte returned from loadAtxSector contains the FDC status bits. Are you taking inversion into account correctly? See page 185 of the Altirra Hardware Reference manual:

<http://www.virtualdub.org/downloads/Altirra%20Hardware%20Reference%20Manual.pdf>

#42 - 08/22/2018 07:03 AM - foft

Why does the code default to 0xf7? It says that means 'data crc or weak sector' in the hardware manual. The comment in atx.c says default in case sector not found. Shouldn't that be 0xef which means 'record not found'?

Btw scanalyzer starts working after a few sectors. I think it's the other bug (sio fifo issue).

#43 - 08/22/2018 07:19 AM - Farb

Hmm, good point. The status bits are active low which would make \$F7=11110111 and \$EF=11101111. Bit 4 represents record not found so this may very well be a bug in the current ATX code. I will need to test and confirm.

#44 - 08/22/2018 08:50 AM - Farb

I quickly confirmed that changing \$F7 to \$EF didn't break any of the images I ran the Scanalyzer test and it returns the expected status codes. So I will check that fix into the SDrive repository later today. Thanks for pointing it out!

#45 - 08/22/2018 12:50 PM - foft

What is a phantom sector? Same sector number and valid/crc error/weak on same track? Do you have examples for one on one of the disks you

sent?

#46 - 08/22/2018 12:55 PM - Farb

I believe phantom and duplicate sector are synonymous. Yes, multiple sectors with the same number on a track. Both Agent U.S.A. and Bruce Lee have them in the disks that I provided.

#47 - 08/22/2018 12:56 PM - Farb

Here's further info about those two disks:

<http://a8preservation.com/#/software/dump/82>
<http://a8preservation.com/#/software/dump/257>

See specifically the "copy protection" row.

#48 - 08/24/2018 09:41 PM - foft

Ooops, had the subtraction backwards in waitForAngularPosition, doh.

#49 - 08/24/2018 09:42 PM - foft

Now M.U.L.E. works:-) + Bruce Lee.

#50 - 08/25/2018 12:02 AM - Farb

Fantastic! :-D

#51 - 08/25/2018 08:31 AM - foft

I'll put it up as atxv4 shortly.

Tested the few you sent. They all seem to work, except Bandits still does not play... Does this atx work fully via sdrive to Eclair (800 mode) but not via this? Wondering if the remaining problem is a core bug or atx support issue.

#52 - 08/25/2018 08:41 AM - foft

Hmm, Bandits just worked with no change. Perhaps related to the other sector error issue.

#53 - 08/25/2018 03:58 PM - Farb

I just tested Bandits with SDrive Max and Eclair XL in 800 mode. It works fine. You need to wait until the full game attract mode starts (e.g. full gameplay is shown with attacking ships moving on the screen).. The game loads from disk before each level.

Also did some quick testing with atxv4 and can confirm that more titles are working (e.g. M.U.L.E., Archon, Bruce Lee, Goonies, Montezuma's Revenge).

Agent U.S.A. works but behaves strangely. I've reproduced twice that the loading speed doesn't seem correct. The cross-fading titles should get to the copyright notice before the games starts. On EclairXL, I only see them get to the music credits.

#54 - 08/25/2018 05:31 PM - foft

Hmm, will check the timing some more.

Does flight sim II work on the max? I remember that being time sensitive and get lots of system errors loading it.

#55 - 08/25/2018 06:00 PM - Farb

I just tried it with SDrive Max and Eclair in XL/XE mode. It loaded and I got to the point that the plane was on the runway ready for take-off.

#56 - 08/26/2018 08:51 AM - foft

Thanks. It loaded more on the 400/800 rom (but not that far) and stopped early on the XL/XE. I think I have the patched xformer rom, perhaps that is why.

From the check suite 102:

I am getting OK/NOK/OK/OK and then the L3 highlighted that Farb mentioned.

The NOK is 'Sector skew 0B(inv)-08-0B(inv)'.

#57 - 08/31/2018 09:39 PM - foft

Updated to the v9 code.

I need to split out the eclairXL changes to make this easier to update. I was going to make a diff from v8 and apply with patch, but I don't see v8 any

more? Anyway merged and don't think I missed anything since it all passes with check protection 107.

#58 - 08/31/2018 09:47 PM - foft

I put this up as atxv5.

So 107 all passes, but 107 track35 seems to have a lot of NOKs...

#59 - 08/31/2018 11:07 PM - Farb

I just updated to atxv5 and confirmed that 107 all passes and the few games I tested worked great. Fantastic work!!

#60 - 08/31/2018 11:08 PM - Farb

Oh, and let me know if there are things I can do in the atx.h/c code to make it easier for you to consume :-)

#61 - 09/01/2018 01:44 PM - foft

- *File atx.patch added*

Could it be changed somehow like this? i.e. move the avr parts to another file and add byte swapping?

#62 - 09/01/2018 04:09 PM - Farb

Ok, done and checked into v09 branch.

#63 - 09/01/2018 04:40 PM - foft

That's great, thanks.

#64 - 09/02/2018 09:37 PM - foft

- *File atx.c.patch added*

So this is what I ended up with as a patch to atx.c.

I'm updating gTrackInfo on every disk change or access to different drive, to save space.

I think I had issues with not being able to change globals if I assigned a value, so I removed the defaults from them.

byteSwapAtxTrackChunk is called instead byteSwapAtxSectorHeader in one place.

#65 - 09/02/2018 10:00 PM - Farb

Ok, I just checked in a couple more changes. Your patch file should get pretty small now :-D

#66 - 09/27/2018 07:12 PM - foft

- *Status changed from New to Closed*

Closing this, since the core support is present. Of course we can raise new tickets for improvements/fixes or e.g. 1050 support.

Most items on the 110 test disk pass. Sometimes Broderbund test1 fails.

Files

test-atx.7z	221 KB	08/20/2018	Farb
Scanalyzer, The v3.3 (1984)(Alpha Systems).zip	59.9 KB	08/21/2018	Farb
atx.patch	6.99 KB	09/01/2018	foft
atx.c.patch	1.67 KB	09/02/2018	foft