Last updated: 2022/3/26

X68000 MPU Accelerator PhantomX -Phantom

of mc680x0 for X68000-

Announcement

We sent the beta test (Hitobashira) version of the board to several testers. The app is currently BETA4.

Introduction

PhantomX is simply an MPU accelerator for X68000. As with other accelerators, remove the MC68000 from the main unit and install it in its place. Supported models are ACE / EXPERT / PRO / SUPER / XVI (excluding Compact) (the first generation is not applicable due to restrictions on the mounting position of MPU).

The substance of PhantomX is a software emulator of Raspberry Pi and MC680x0. Install the Raspberry Pi with the dedicated software installed using the baseboard and relocator. Raspberry Pi supports Zero / Zero WH / Zero2 / 2B / 3A + / 3B / 3B + / 4B.

Phantom X was designed exclusively for X68000. Performance is improved by optimizing for the hardware structure of X68000. Therefore, it is not a general-purpose accelerator that can be used with other models.

The speed as an accelerator depends on the performance of the Raspberry Pi.



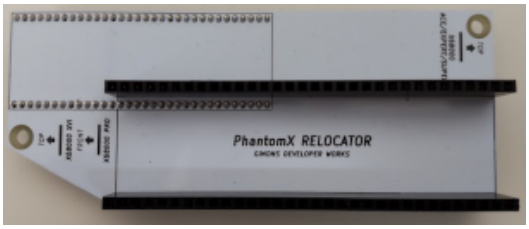
hardware

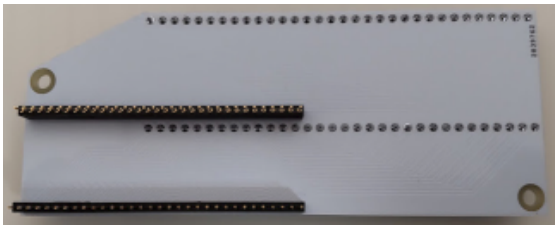
Relocator

MC68000, which is the MPU of X68000, is socketed from the factory and can be removed. The socket is SDIP (shrink DIP) with a 1.778mm pitch (not very common these days). The position of the socket

differs depending on the model, but it is roughly divided into three types: ACE / EXPERT / SUPER, XVI, and PRO.

The role of the relocator is twofold. The first is to convert SDIP to a general 2.54mm pitch DIP. The second is to adjust different socket positions for each model. It is designed to support socket position conversion of all models with one relocator.





Baseboard

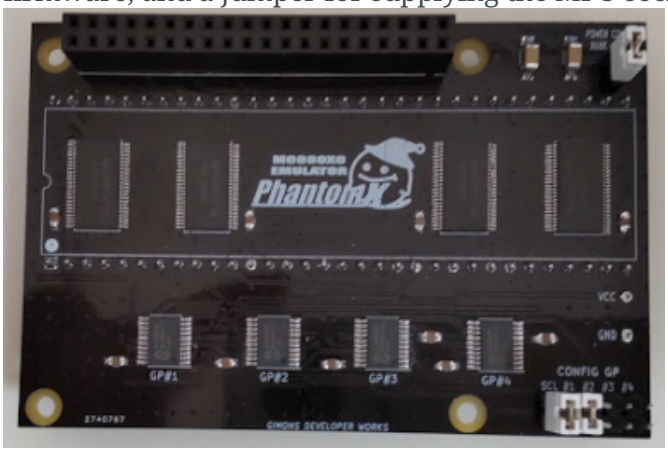
This is the main baseboard for Phantom X hardware. It has a connector with the same pinout as the DIP version of the MC68000. Connect it to the socket of the MPU with the one attached to the relocator.

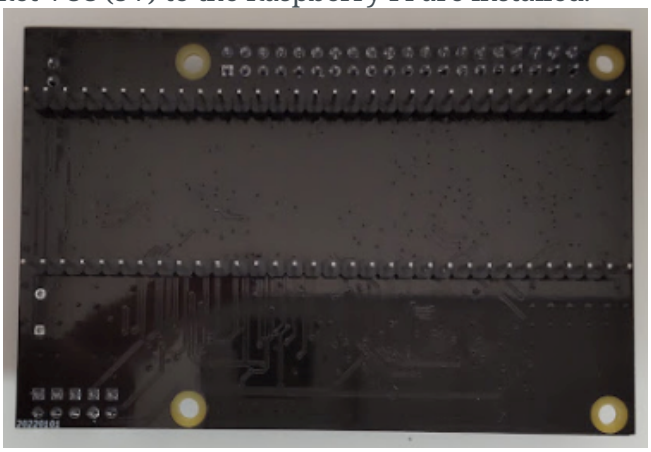
The four wide bus D-FFs on the left and right of the logo on the baseboard are logic ICs for level conversion and holding of bus signals with GPIO of Raspberry Pi. It processes while switching the limited GPIO of Raspberry Pi.

The bottom four of the baseboard are Dialog Semiconductor's SLG46826G. This is a type of product called GreenPAK Programmable Mixed-Signal Matrix In-System Programmability. It is an excellent version that can be said to be a simplified version of FPGA, and if it is configured with a logic IC, it is possible to configure dozens of required circuits with this one.

Each of the four SLG46826Gs has its own firmware written. GP # 1 controls the bus sequence, GP # 2 controls the bus with DMAC, GP # 3 monitors the bus sequence, and GP # 4 controls the IPL signal line.

In addition, a GPIO connector (20x2 housing), an I2C jumper for rewriting the SLG646826G firmware, and a jumper for supplying the MPU socket VCC (5V) to the Raspberry Pi are installed.



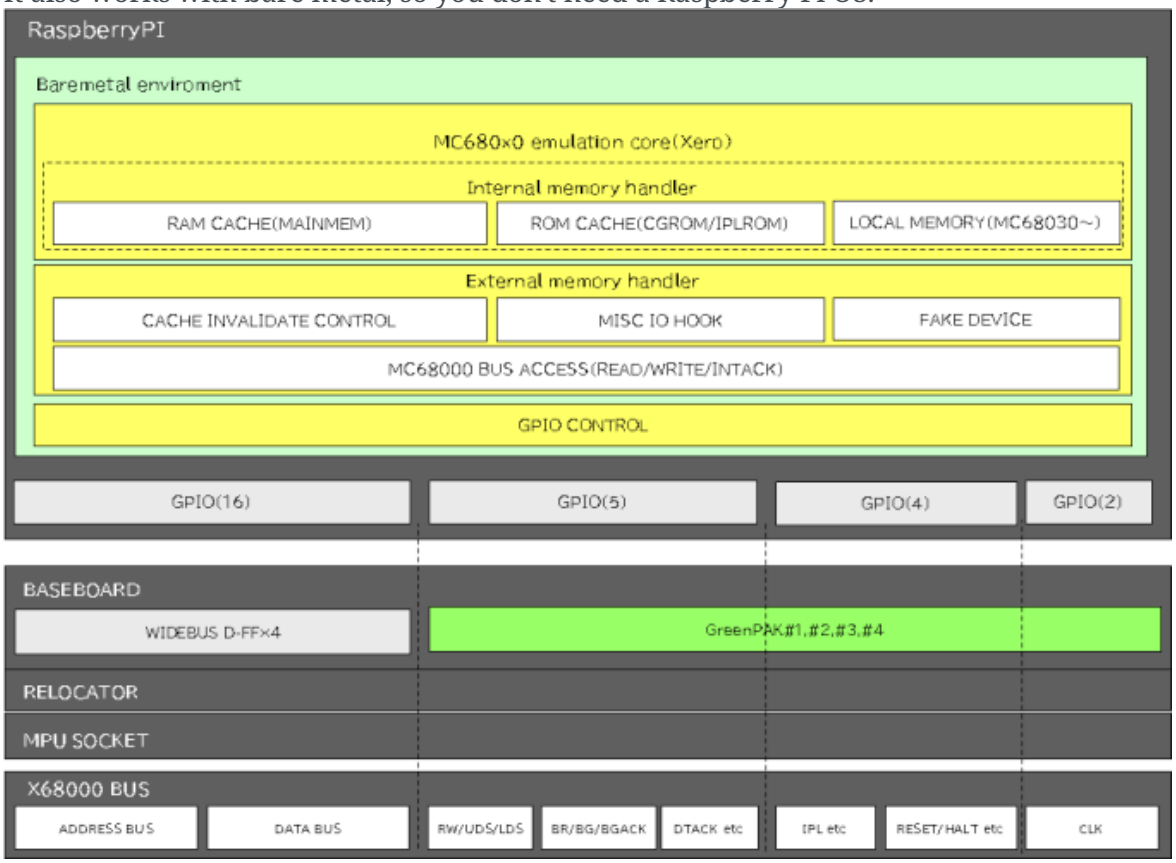


software

Functional structure

The functional configuration within the PhantomX software is shown below. It has a layered structure as much as possible and is designed to separate the parts that depend on the Raspberry Pi.

It also works with bare metal, so you don't need a Raspberry Pi OS.



MC680X0 emulation

I will use a core I made by referring to the MPU core called StarScream, which has a proven track record in XM6. This MPU core is implemented in C ++ and is codenamed "Xero". Emulation of MC68000,68030,68040,68060 is possible (including some simple implementation). Originally, the cache mechanism of MC68030 or later was implemented, but now it is deleted. Instead, I implemented my own cache mechanism inside the internal memory handler.

Internal memory handler

Most of the instructions of MC680X0 are implemented in Xero. If instruction fetch, memory read, or write is required, processing is performed via the internal memory handler. It has its own cache mechanism to reduce access to external memory handlers as much as possible. Only the same cache capacity as the mounted memory of X68000 is reserved. Therefore, as long as the cache hits continue, the X68000's main memory will not be accessed. However, since the write process is write-through, it writes to the main memory.

External memory handler

It is a relay layer for going through bus access to X68000. It hooks the request from the internal memory handler and performs pre-processing and post-processing. The most important process is guaranteeing the integrity of the original cache. Since the X68000 has a DMAC, the DMAC writes to the main memory without the involvement of the MPU. At this time, an inconsistency occurs with the cache information in the internal memory handler. On the other hand, since the MPU gives operation instructions to the DMAC, it is possible to predict the area where cache inconsistency will occur by monitoring the contents of the instructions to the DMAC. In other words, inconsistency is avoided by invalidating the cache in the internal memory handler just before the operation of DMAC. In addition, we have implemented forced weight insertion to mitigate various problems caused by MPU emulation being too fast (equivalent to problems that occur during overclocking), and fake devices to disguise as X68030.

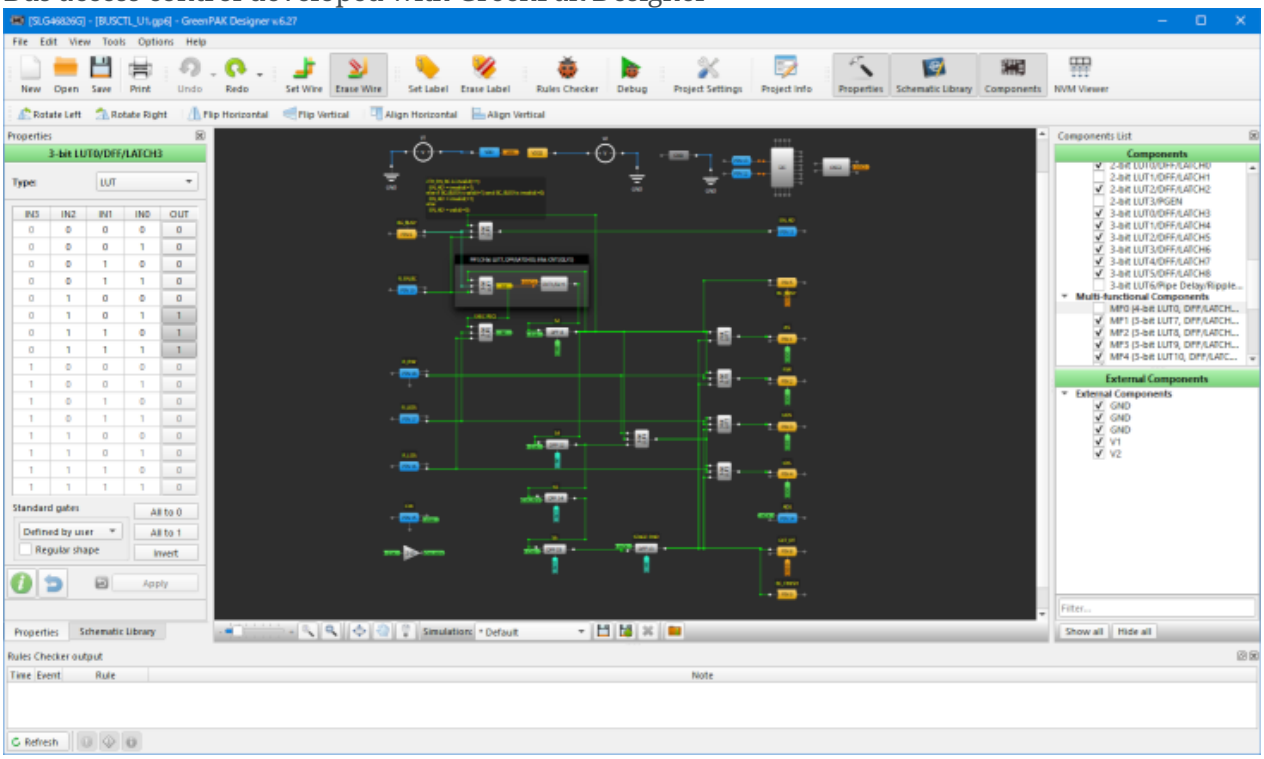
GPIO control

When the bus access to X68000 is finally needed in the external memory handler, it is processed by GPIO control. Since the actual bus access control is performed by the firmware of SLG46826G on the baseboard, only the address bus and data bus settings, the status check and data acquisition after the bus access is completed are performed.

firmware

As I mentioned earlier, each of the four SLG46826Gs has a big role to play, and I have written the individual firmware. The firmware is developed by GreenPAK Designer. Export the firmware from GreenPAK Designer in HEX format and write it using the SLG46826G's I2C interface. I am using Raspberry Pi for writing. We are also developing an application required for writing, so if you follow the procedure strictly, you can update the firmware later.

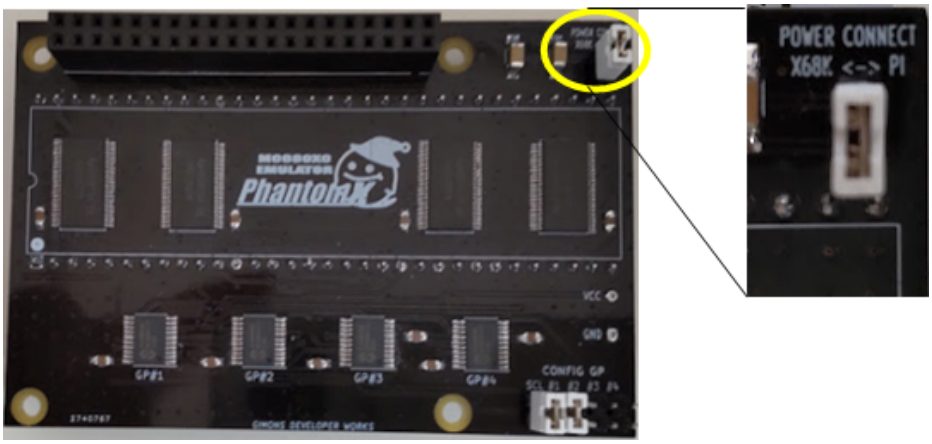
Bus access control developed with GreenPak Designer



Installation (hardware)

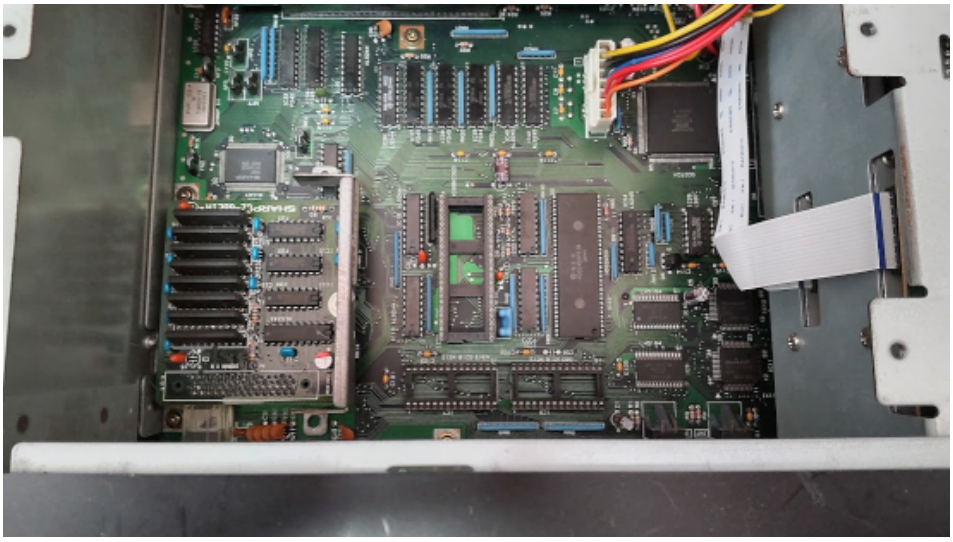
Baseboard jumper settings

There are some jumpers on the baseboard. The only jumper that needs to be set is the jumper that supplies the VCC (5V) of the MPU socket to the power supply of the Raspberry Pi. You don't need to power the Raspberry Pi separately (rather, never connect an external power source). Be sure to remove this jumper if you have no choice but to supply external power to the Raspberry Pi due to power shortage of X68000.

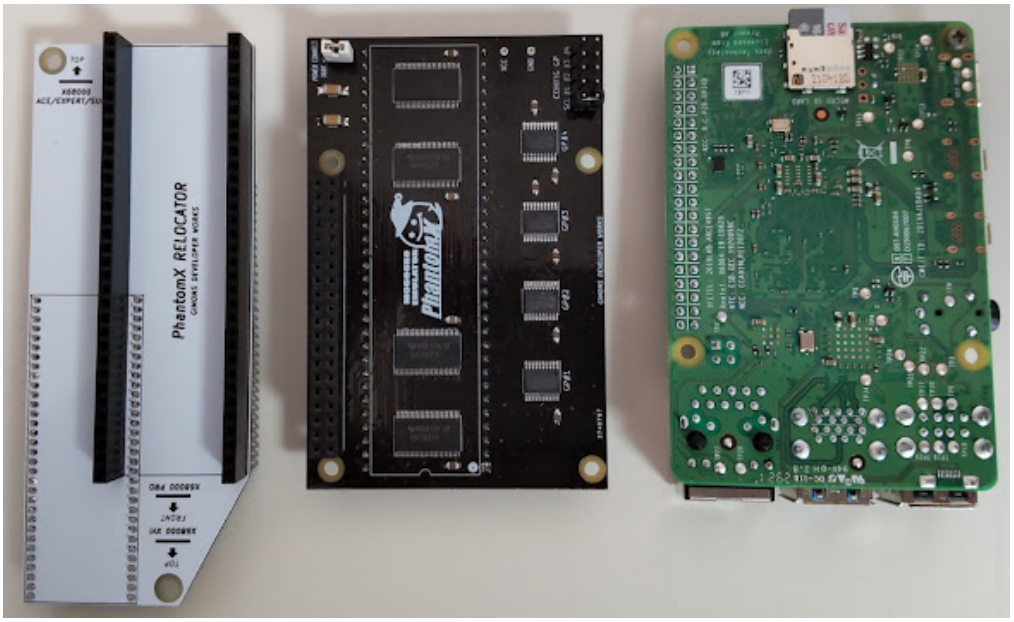


X68000 PRO (PRO2)

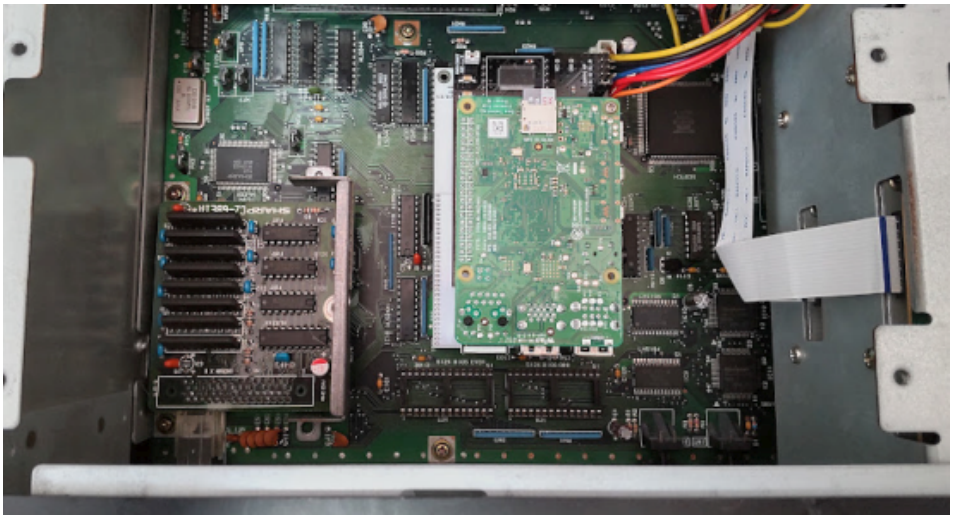
The horizontal model PRO is the easiest model to introduce. You can access the MPU socket immediately by removing the case and the support. I will not explain in detail how to remove the MPU, but it is better to use an IC extraction tool (it is possible to gradually remove it with a screwdriver etc., but an unexpected accident may occur). In the photo, the MPU has already been removed (the bottom is the front and the top is the extended IO). A spring-type MPU socket is attached, but in order to prevent interference with the capacitor on the side, a socket for a round pin is placed on the MPU socket and raised. The capacitors on the socket may interfere, so you may need to replace them sideways or tilt them diagonally.



The relocator is shared with other models and the direction is different from when connecting to XVI. You need to implement the relocator, baseboard, and Raspberry Pi so that they overlap in the direction of the photo.



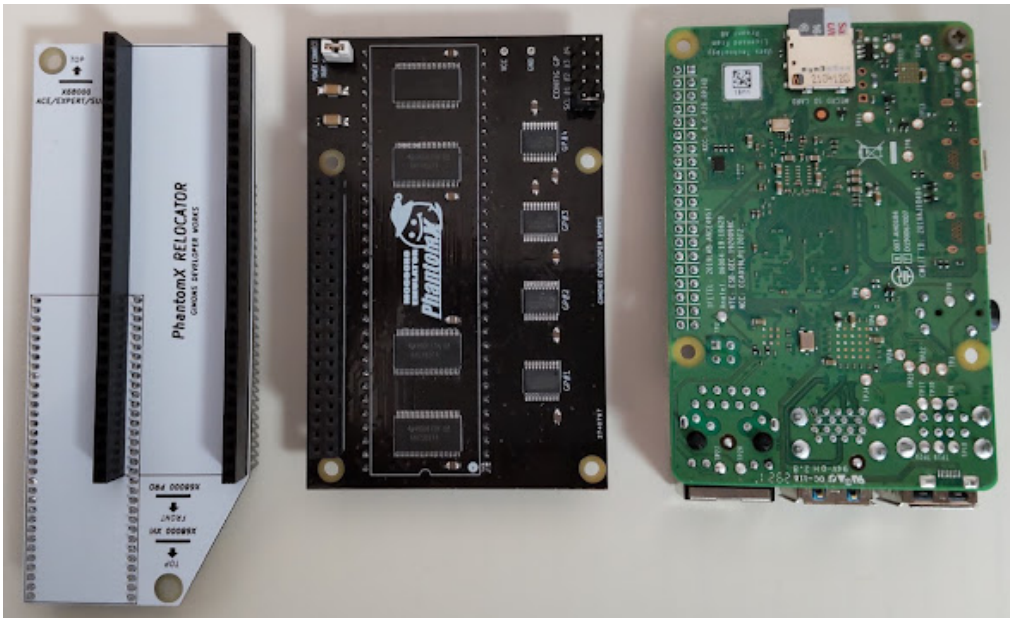
It is mounted on the MPU socket (the power connector to the main board is barely enough).



X68000 ACE / EXPERT (EXPERT2) / SUPER

It is a vertical model 10MHz machine. You need to remove the mainboard and remove the shield to access the MPU socket. Also, since there is a socket in the lower left corner of the main board, it will be necessary to pull out the MPU with the main board removed (IC pulling tool recommended). The photo shows the ACE main board with the MPU already removed. A spring-type MPU socket is attached, but in order to prevent interference with the capacitor on the side, a socket for a round pin is placed on the MPU socket and raised.

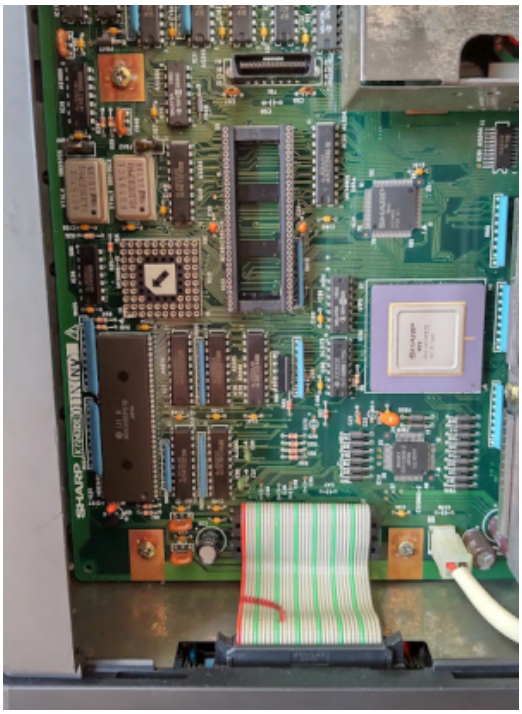
Overlay the relocator in the same direction as PRO.



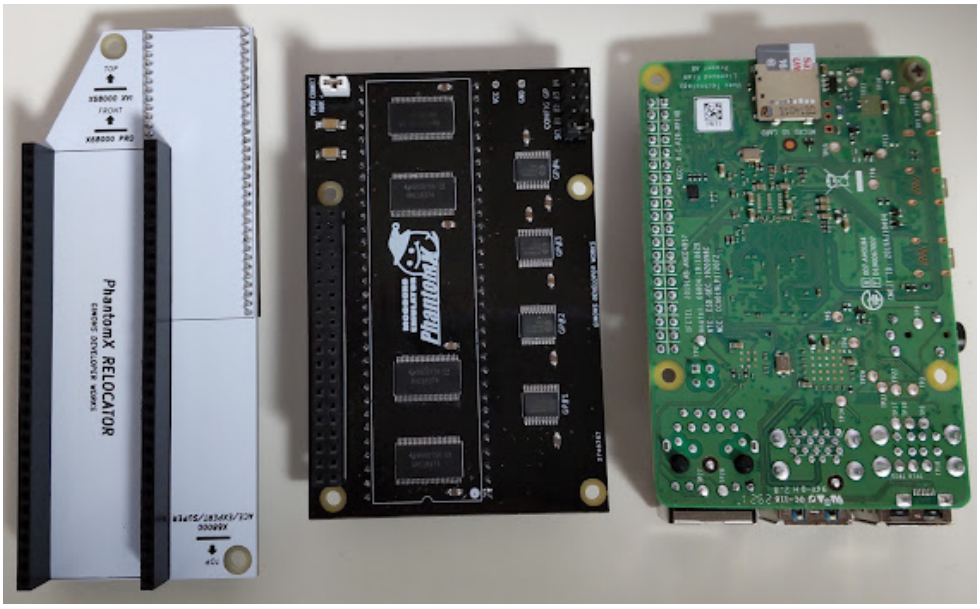
It is mounted on the MPU socket. Note that the sockets overlap under the case and the Phantom X connector is hard to see. Mount while checking the pin position with a penlight etc. from above.

X68000 XVI

It is a vertical model 16MHz machine. You need to remove the mainboard and remove the shield to access the MPU socket. Access is relatively easy because there is a socket on the lower left of the center of the main board. Please remove the main board before pulling out the MPU (IC pulling tool recommended). The photo shows the main board with the MPU removed. A spring-type MPU socket is attached, but in order to prevent interference with the capacitor on the side, a socket for a round pin is placed on the MPU socket and raised.



Overlay the relocator, baseboard, and Raspberry Pi in the following directions (opposite to PRO and other vertical models).



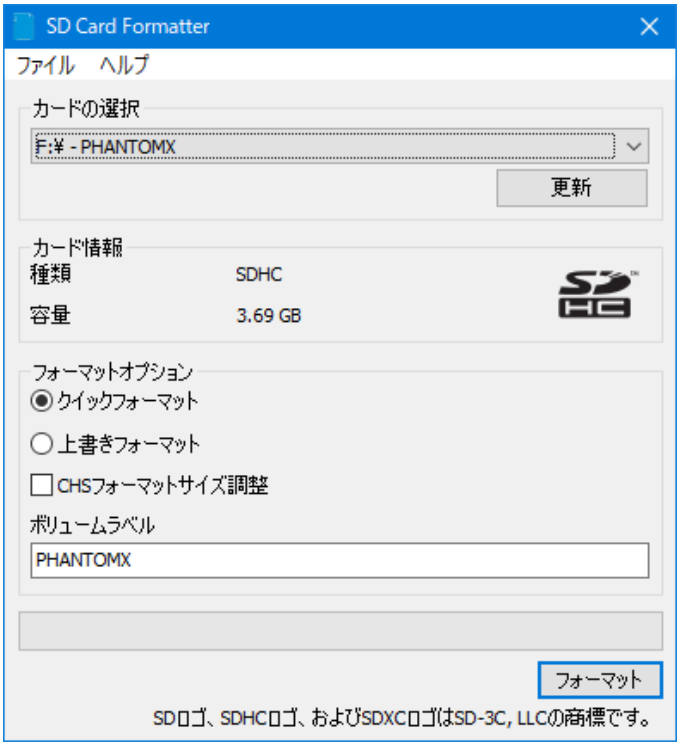
It is mounted on the MPU socket. Since the upper end of the socket is visible, it is possible to visually align the pins.

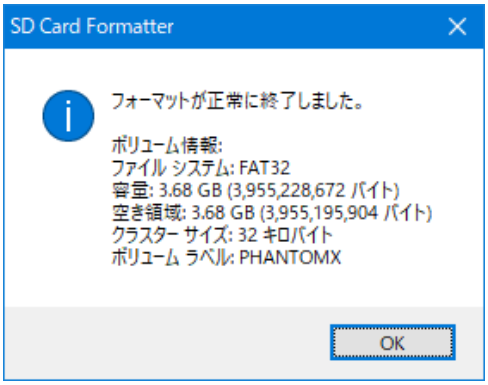
Installation (software)

SD card initialization

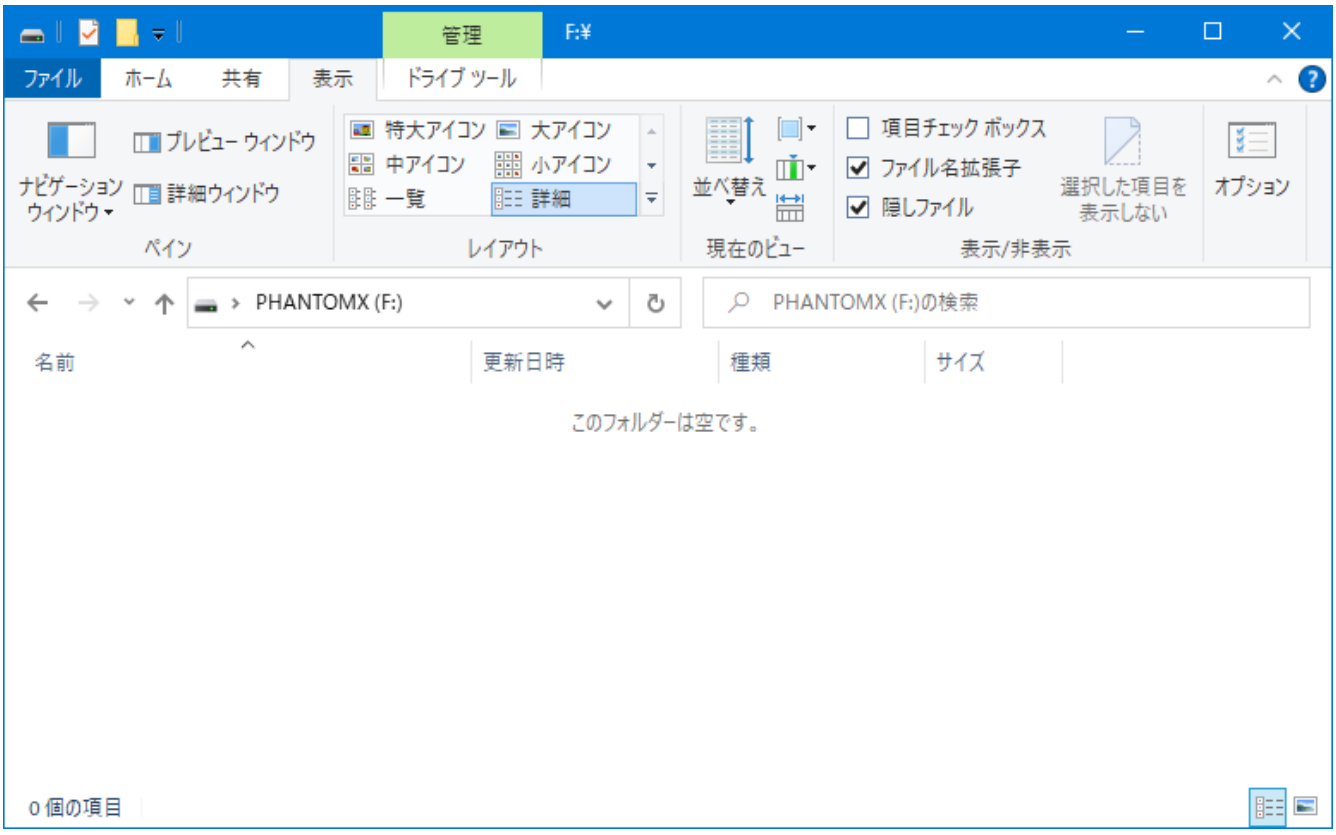
Insert an appropriate SD card into your PC and initialize it with the "SD memory card formatter" provided by the SD Association.

If the entire area is formatted with FAT32 as shown below, it is successful.





Check if the SD card is empty and can be seen from the PC.



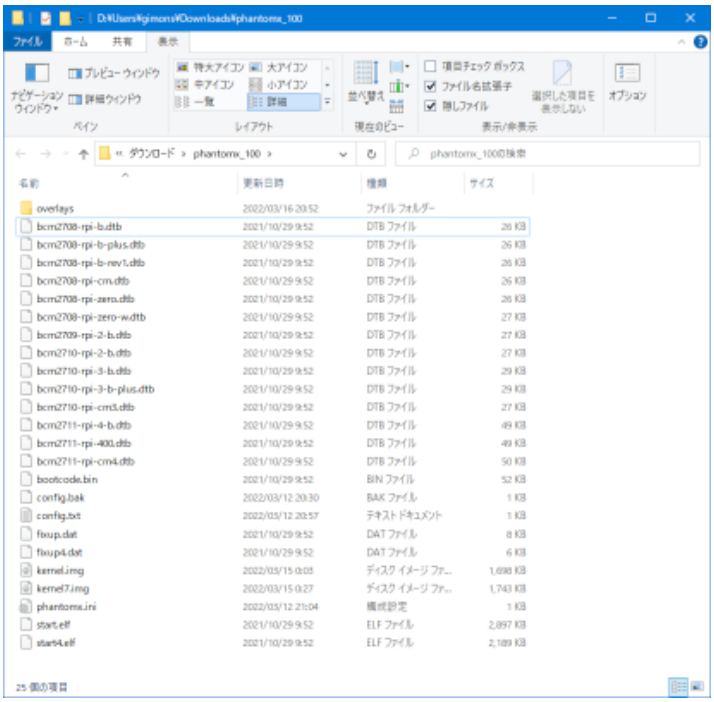
Software download

Download the PhantomX module from the following link (Beta).

[phantomx_beta4.zip]

Software deployment and transfer

Extract the downloaded Zip file and you should see the following files.

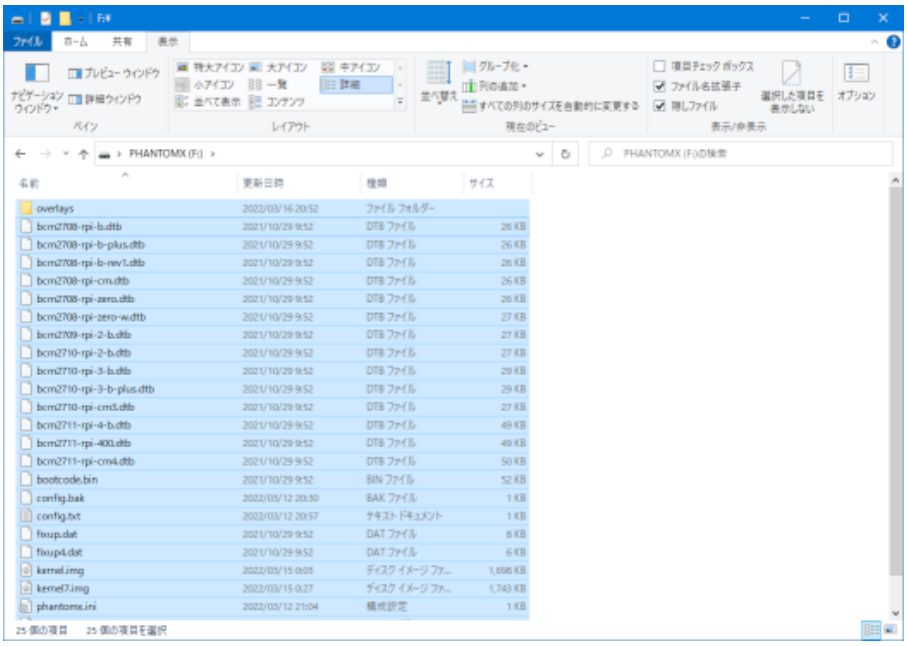


The following files are related to PhantomX.

kernel.img	PhantomX main unit (for Raspberry Pi Zero / W)
kernel7.img	PhantomX main unit (for Raspberry Pi Zero 2 / 2B / 3A + / 3B / 3B + / 4B)
phantomx.ini	PhantomX configuration file

Others are the firmware and configuration files needed to boot the Raspberry Pi. There is a power saving setting in config.txt, so be sure to use the extracted file.

Copy all files to the initialized SD card.



Insert your SD card into your Raspberry Pi and you're done.

Setting method

Use it by default without setting anything at first. The following settings will be explained in detail in the application section.

setting file

"phantomx.ini" is the configuration file. The downloaded module is described as follows.

```
#EMUMODEL 000
# IPLROM IPLROM.DAT
#CGROM CGROM.DAT
#SCSIINROM SCSIINROM.DAT
# SRAM SRAM.DAT
```

Set in a "key" and "value" pair in the definition file with one setting per line. Separate the "key" and "value" with a space or tab. If you add "#" at the beginning, the line will be ignored. It doesn't matter if there are blank lines.

上記の例では全て#でコメントアウトされていますので何も設定していない状態と同じになります。

設定項目

キー	値
EMUMODEL	エミュレーションするMPUを指定します。 指定できる値と意味は以下の通り。 000 : MC68000 *デフォルト 030 : MC68030(MMU、128MBローカルメモリ) 040 : MC68040(MMU、内蔵FPU、128MBローカルメモリ) 060 : MC68060(MMU、内蔵FPU、128MBローカルメモリ)
IPLROM	IPLROM($fe0000-$ffffff)のROMデータファイルのパスを指定します。 指定がなければX68000本体のIPLROMを使用します。 IPLROMを差し替える必要がある時に指定してください。
CGROM	CGROM($f00000-$fbffff)のROMデータファイルのパスを指定します。 指定がなければX68000本体のCGROMを使用します。 CGROMを差し替える必要がある時に指定してください。
SCSIINROM	内蔵SCSIROM($fc0000-$fc1fff)のROMデータファイルのパスを指定します。 指定がなければX68000本体の内蔵SCSIROMを使用します。 内蔵SCSIROMを差し替える必要がある時に指定してください。 If specified on a SASI machine model that does not have a built-in SCSI other than SUPER / XVI, it emulates a fake built-in SCSI I / F (it looks like no SCSI device is connected).
SRAM	Specifies the path of the SRAM ($ ed0000- $ edffff) data file. The SRAM of the main unit will no longer be used and will be expanded to 64KB at the same time. If the specified file does not exist, it will be created automatically.

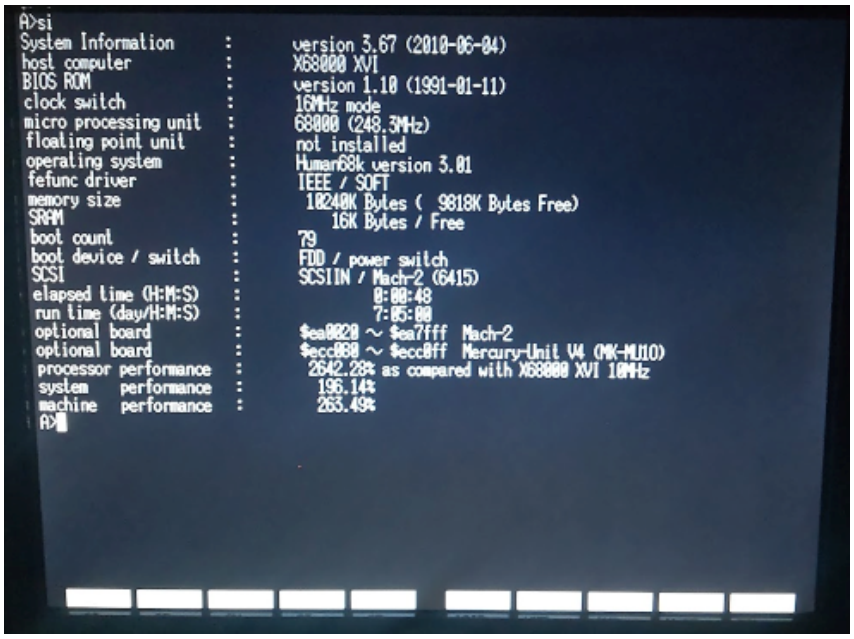
	Writing to SRAM also updates this file, so it can be used as a substitute for battery backup.

how to use

Turn on the X68000 as usual. The Raspberry Pi will also start booting when the MPU socket is powered.

It may take a few seconds for the PhantomX kernel module to start booting.

System Information when using RaspberryPi 4B (new version). It seems that the speed equivalent to 040 Turbo comes out.



System Information when using RaspberryPi Zero WH. Overclocked XVI performance above 24MHz seems to come out.

Advanced version

Use a newer version of IPLROM

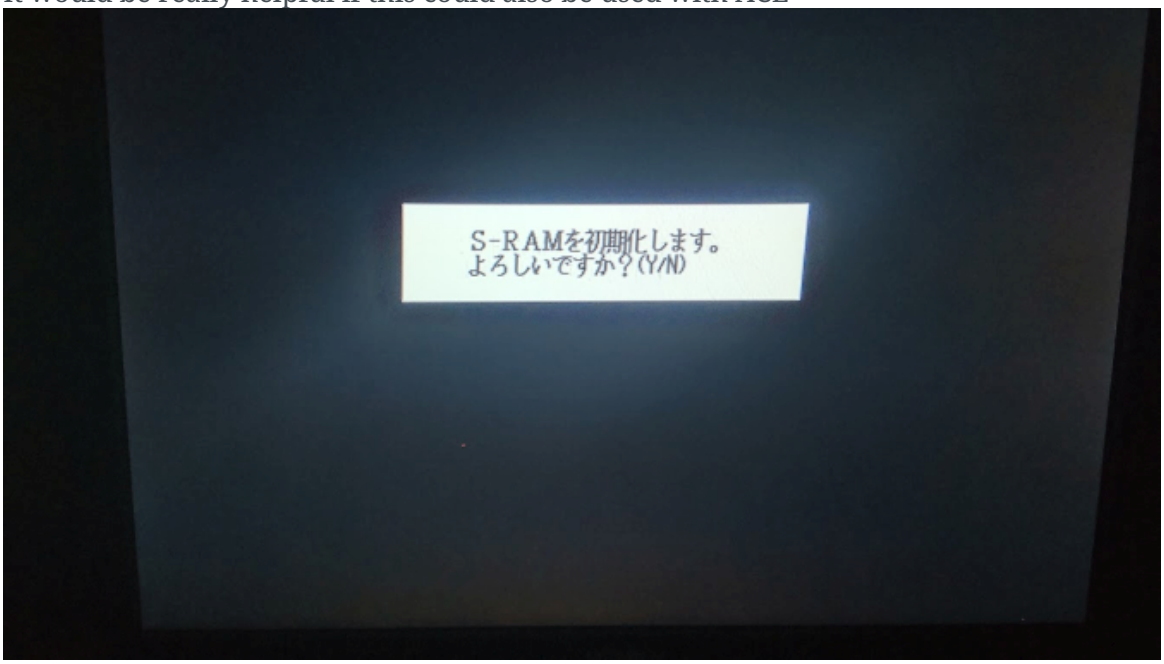
The IPLROM of ACE / EXPERT / PRO / SUPER is inconvenient because the SRAM initialization (reset while pressing the CLR key) function extended by the IPLROM of XVI cannot be used. Such troubles can be solved by replacing it with the IPLROM of XVI in the configuration file.

Prepare the IPLROM data file to be replaced with the SD card and specify it in the configuration file.

An example of using the IPLROM of XVI released free of charge

```
IPLROM IPLROMXV.DAT
```

It would be really helpful if this could also be used with ACE



Enjoy the atmosphere of X68030

MC68030をエミュレーションさせて上位機種であるX68030に似た環境を再現できます。PhantomXはEMUMODELに030以上が指定された場合X68030で拡張されたいくつかのIOポートを内部的に偽のデバイスとして再現します。

X68030のIPLROMは無償公開されたもの(IPLROM30.DAT)が利用できます。

内蔵SCSIROMに相当するROM30.DATは無償公開されていませんのでご自身で用意する必要があります(内蔵SCSI用のROMで代替するなどいくつかやり方はあります)。

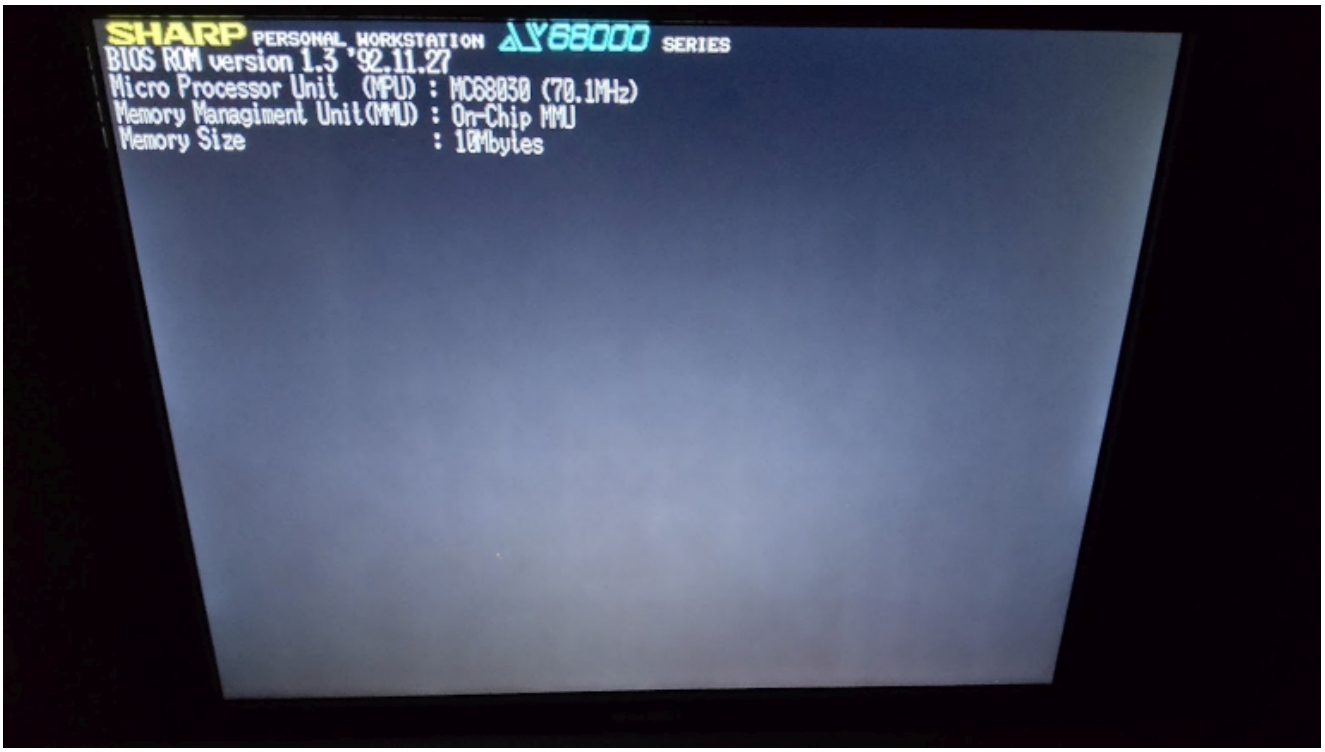
CGROMもX68030の物を使用しないとSX-Window等の12ドットフォントが表示されない場合があります。

下で説明するXEiJで生成できるROMを使用することもできます。

X68030のROMを利用する例

```
EMUMODEL        030
IPLROM          IPLROM30.DAT
CGROM           CGROM30.DAT
SCSIINROM       ROM30.DAT
```

MC68030モード



X68030だと認識されています(試験機はXVIです)

060turboの雰囲気を味わう

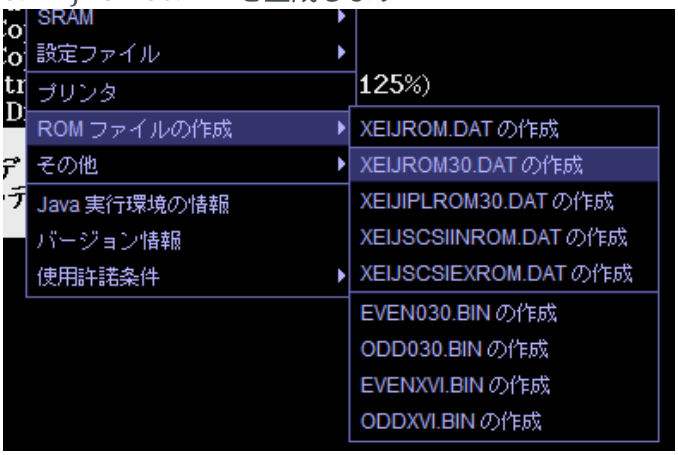
EMUMODELに040もしくは060が指定された場合はそれぞれMC68040、MC68060をエミュレーションします。この時X68030で拡張されたいくつかのIOポートを内部的に偽のデバイスとして再現します。

@kamadoxさんのXEiJで生成できるROMを使用する方法を書いておきます(ありがたや〜)。

1.XEiJを起動して機種を060turboに変更します

2.XEIJIPLROM30.DATを生成します

3.XEIJROM30.DATを生成します

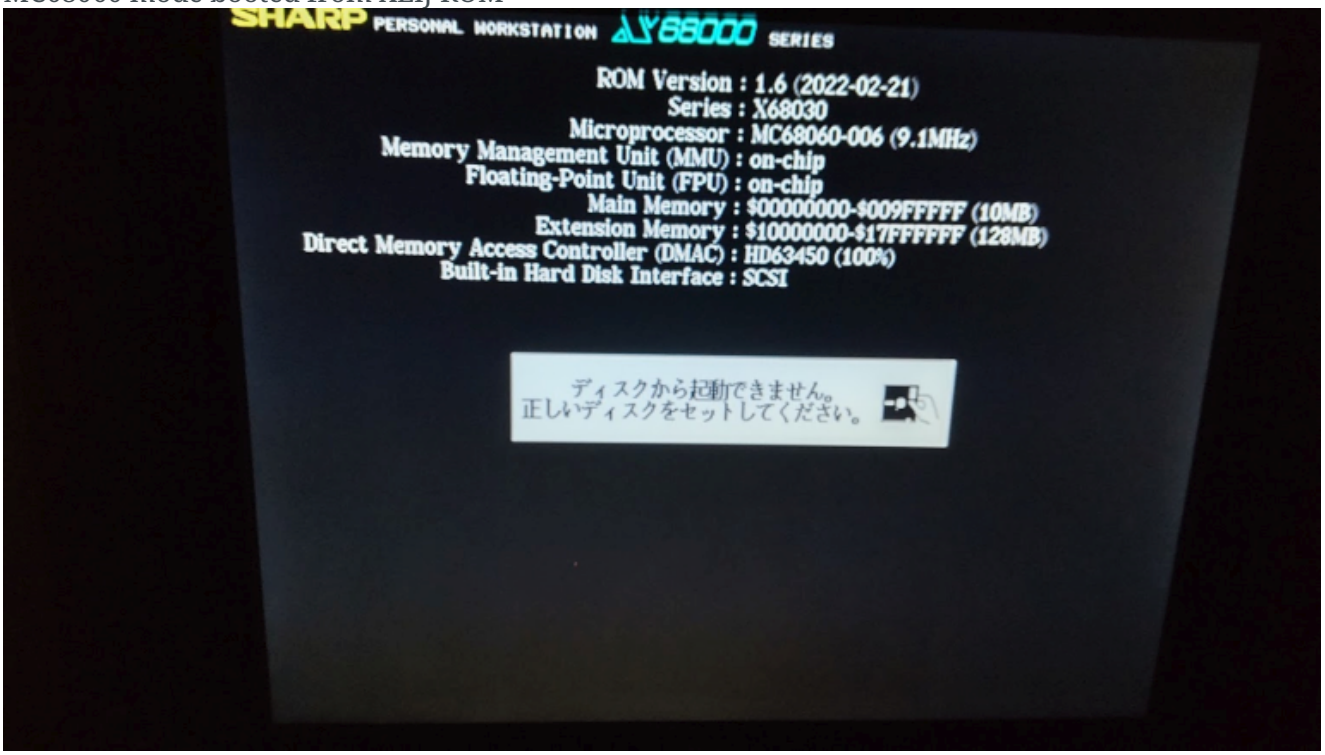


4.XEIJIPLROM30.DATとXEIJROM30.DATをSDカードに保存します

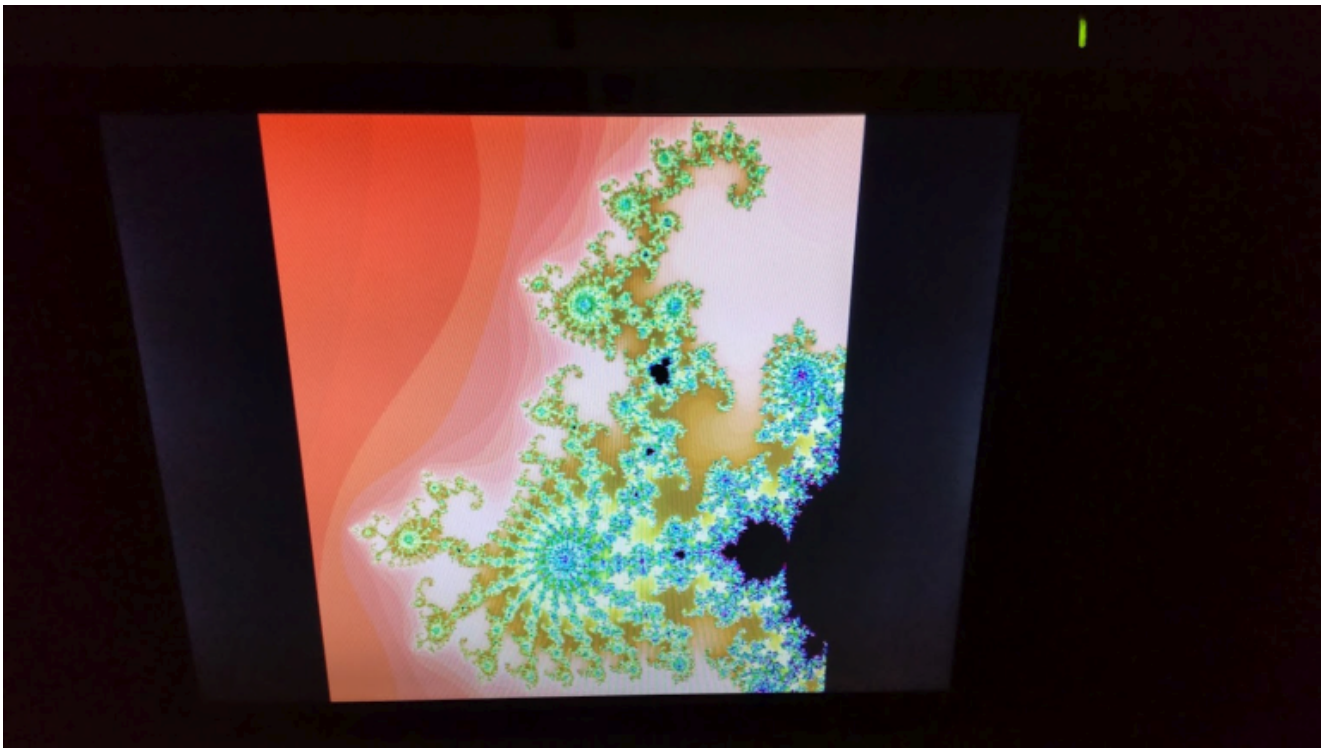
5.phantomx.iniでROMファイルを指定します

```
EMUMODEL        060
IPLROM          XEIJIPLROM30.DAT
CGROM           CGROM30.DAT
SCSIINROM       XEIJROM30.DAT
```

MC68060 mode booted from XEiJ ROM



The usual Mandelbrot



About distribution of relocator and baseboard

to be decided

history

2022/3/26 Under construction (Beta 4 test)

[EOF]