

0.准备工作

0.1 搭建装配线

0.2 准备搭建材料（部分版本部分材料可能需要掏裤裆）

[机架] * 1

[超级服务器] * 1

[加速运算单元(APU)(T3)] * 1 （或[超级中央处理器(CPU)] * 1, [超级显卡] * 1)

[存储器-T3.5] * 2

[因特网卡] * 1

[超级磁盘驱动器 (4MB)] * 1

[超级组件总线] * 2

[线缆] * 若干

[超级显示屏] * 1

[键盘] * 1

[适配器] * 5

[红石 I/O 端口] * 1

[能量转换器] * 1

[磁盘] * 1

[OpenOS 软盘] * 1

[EEPROM (Lua BIOS)] * 1

[超级数据库升级] * 1

[物品栏控制器升级] * 1

[转运器] * 20

[ME 控制器] * 1

[ME 线缆] * 若干

[ME 驱动器] * 1 （或[[ME 箱子] * 1)

[ME 流体存储元件] * 1

[ME 流体接口] * 5

[ME 流体存储总线] * 1

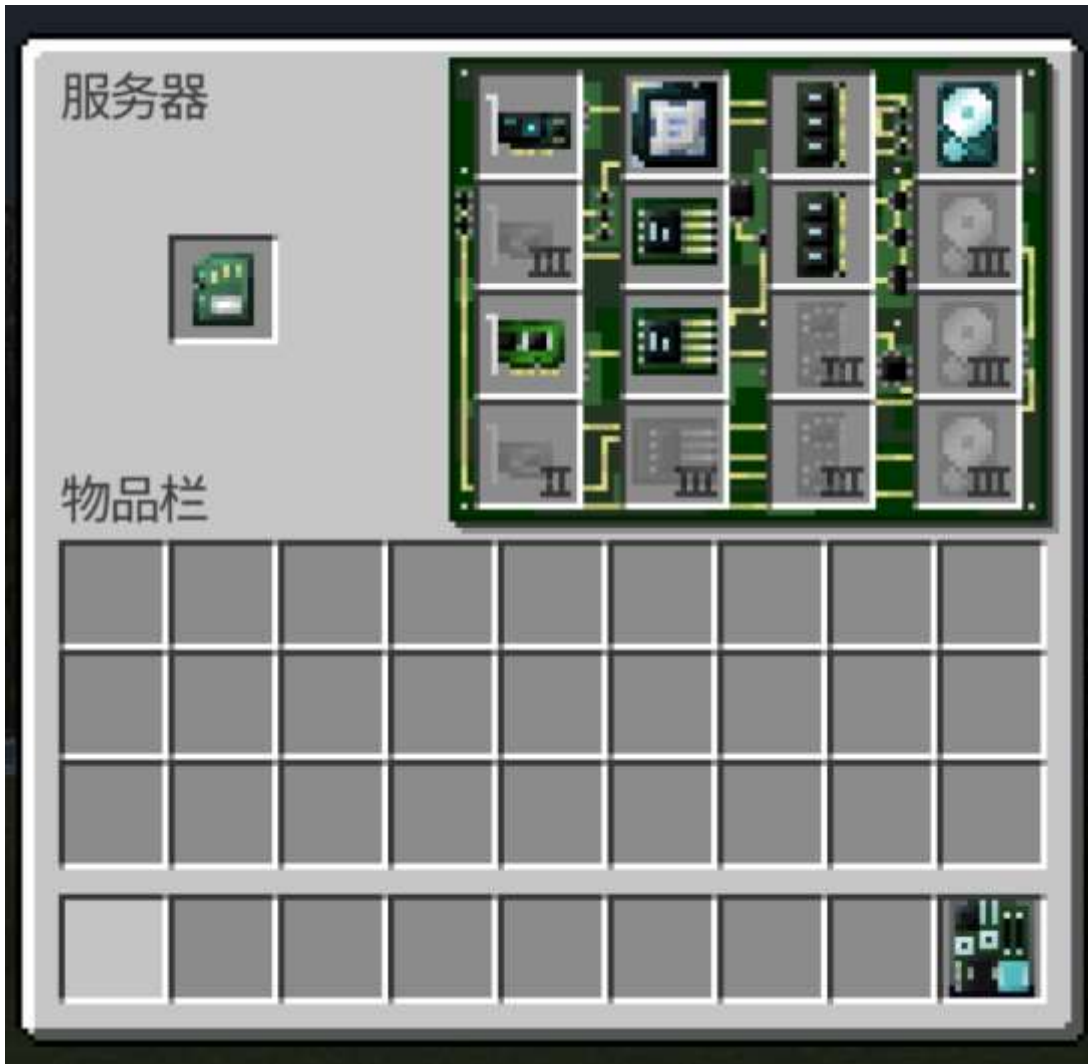
[ME 流体终端] * 1

[箱子] * 1（接收主网发配材料用，可以替换为其他有存储能力的方块，如压缩箱子）

[末影箱子] * 16（频道相同，不能和其他频道互相干扰）

0.3 外围系统搭设

(1) 将[超级服务器] * 1 拿在手上，右击空气打开 GUI，将[加速运算单元(APU)(T3)] * 1 (或[超级中央处理器(CPU)] * 1，[超级显卡] * 1)、[存储器-T3.5] * 2、[因特网卡] * 1、[超级磁盘驱动器 (4MB)] * 1、[EEPROM (Lua BIOS)] * 1、[超级组件总线] * 2 装入其中

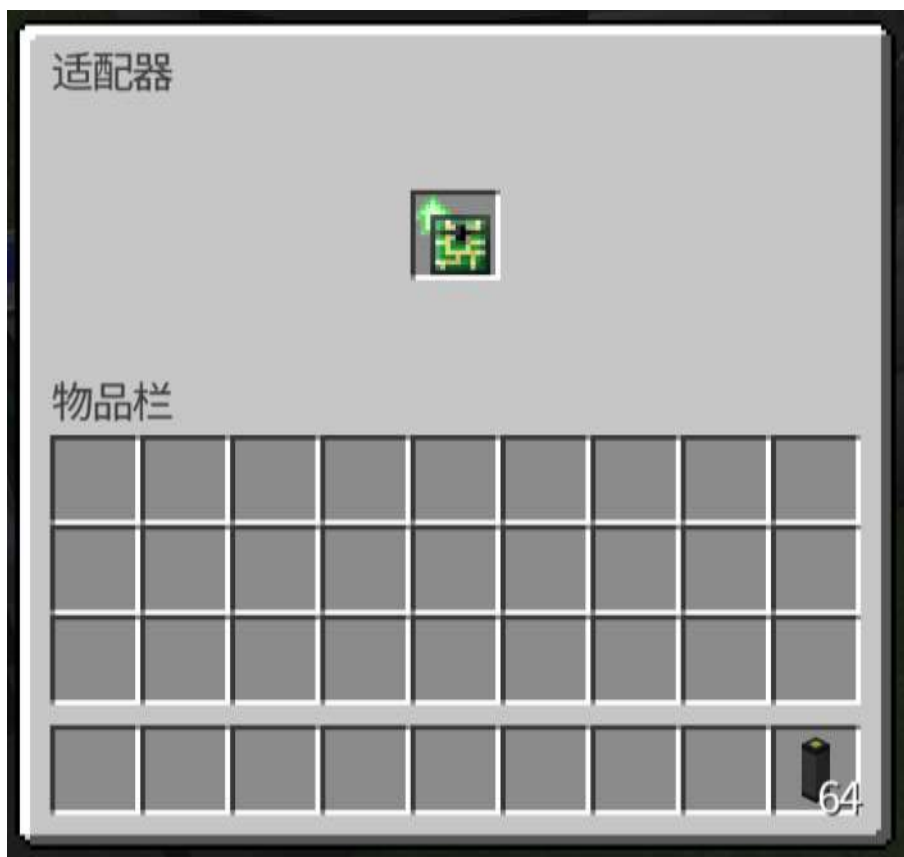


(2) 将机架放于一侧，并右击机架打开 GUI，将服务器放入其中，点击预定连接面以激活服务器控制。注意：机架只能连接一个面的部件，请谨慎考虑如何布线



(3) 放置[能量转换器] * 1、[超级显示屏] * 1、[键盘] * 1、[红石 I/O 端口] * 1（紧贴装配线主方块）、[适配器] * 1（紧贴数据访问仓并放入[物品栏控制器升级] * 1）、[磁盘] * 1（并放入[OpenOS 软盘] * 1），并使用[线缆]与机架对应面相连。





(4) 按照输入仓-转运器-流体接口-适配器的顺序，两两紧贴，随后将转运器和流体接口拆除，但记住转运器和流体接口位置

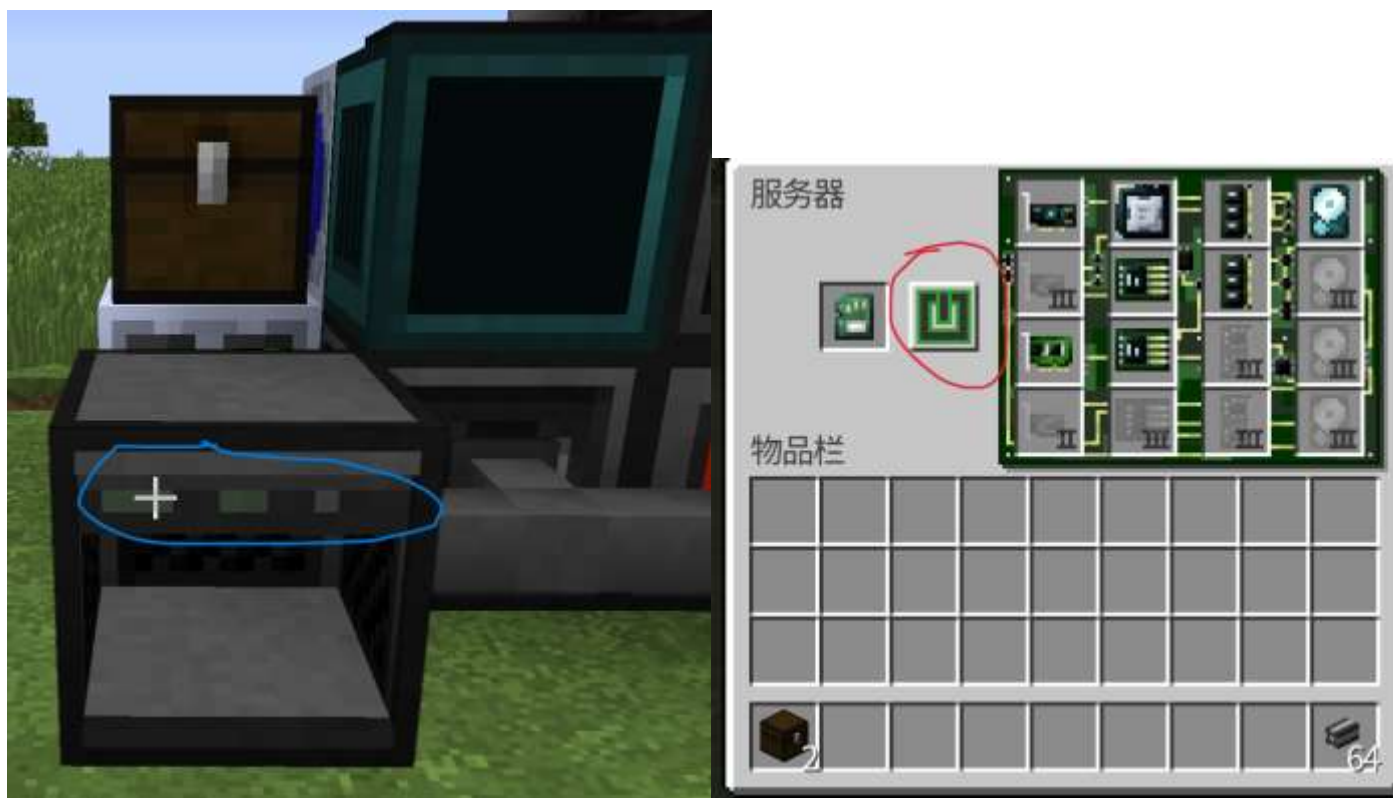




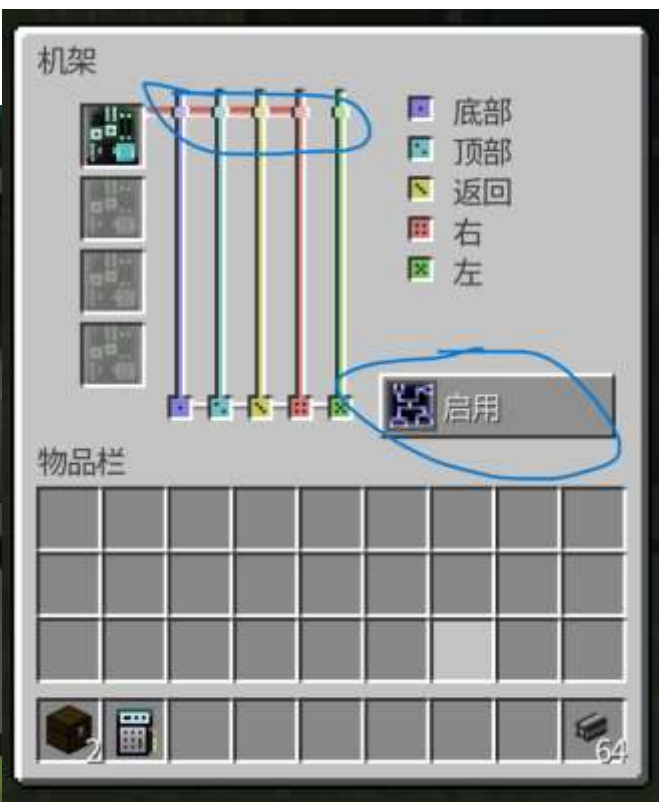
(5) 在距离输入总线一格（预定放置用于输入总线的转运器）的地方放置[末影箱子] * 16，再在距离[末影箱子]一格（预定放置原材料箱用转运器）的地方放置[箱子] * 1（图中替换为压缩箱子）



(6) 点击图中位置，打开服务器电脑界面，点击电源键开机



(6.5) 若如图显示红灯，使用分析器时提示“no bootable medium found: /init.lua”，则尝试调整机架连接选项和下部按钮



(7) 输入“install”并再次输入“y”安装 openos，再输入“y”重启电脑。重启后可拆除“磁盘”及软盘以供其他电脑使用。

```
OpenOS 1.7.5 (2048k RAM)
To disable this greeting, install OpenOS to a writeable medium and remove the '/etc/motd' line from '/etc/profile.lua'.

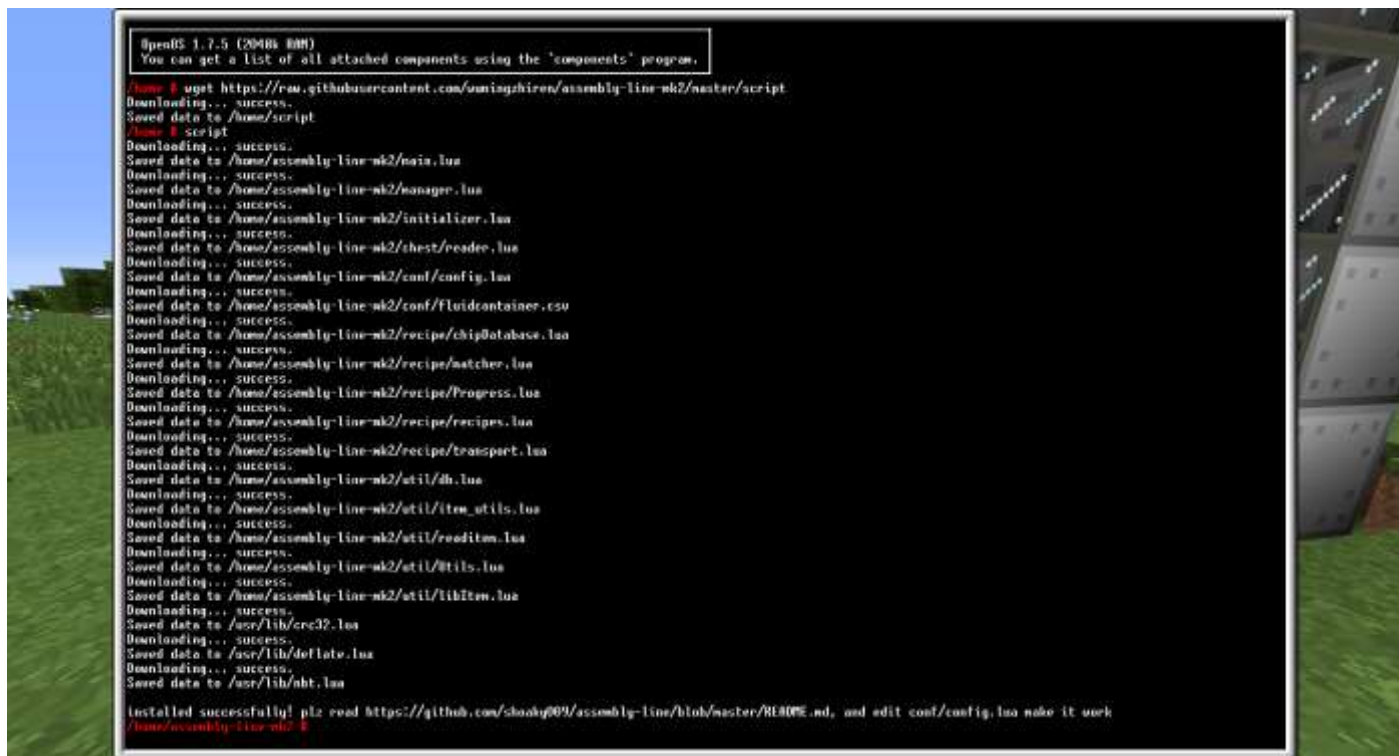
Note: Your home directory is readonly. Run 'install' and reboot.
home #
```

```
Note: Your home directory is readonly. Run 'install' and reboot.
/home # install
Install OpenOS? [Y/n] y
./init.lua -> /mnt/500/init.lua
./lib/ -> /mnt/500/lib/
./lib/bit32.lua -> /mnt/500/lib/bit32.lua
```

```
./boot/00_base.lua -> /mnt/500/boot/00_base.lua
/etc/rc.cfg -> /mnt/500/etc/rc.cfg
/home/.shrc -> /mnt/500/home/.shrc
Installation complete!
Reboot now? [Y/n] █
```

(8) 使用
 wget <https://raw.githubusercontent.com/wumingzhiren/assembly-line-mk2/master/script>
 命令，以下载安装脚本。
 并执行
 Script
 命令，以下载完整脚本。若出现红字，请重新下载。脚本来自 github，访问困难的请

自行翻墙。



```
OpenBS 1.7.5 (20MBs RAM)
You can get a list of all attached components using the 'components' program.

/home # wget https://raw.githubusercontent.com/suniazghireen/assembly-line-nk2/master/script
Downloading... success.
Saved data to /home/script
/home # script
Downloading... success.
Saved data to /home/assembly-line-nk2/main.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/manager.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/initializer.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/chest/reader.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/conf/config.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/conf/fluidcontainer.csv
Downloading... success.
Saved data to /home/assembly-line-nk2/recipe/chipDatabase.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/recipe/matcher.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/recipe/progress.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/recipe/recipes.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/recipe/transport.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/util/dh.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/util/item_utils.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/util/readitem.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/util/Utils.lua
Downloading... success.
Saved data to /home/assembly-line-nk2/util/libItem.lua
Downloading... success.
Saved data to /usr/lib/crc32.lua
Downloading... success.
Saved data to /usr/lib/deflate.lua
Downloading... success.
Saved data to /usr/lib/nbt.lua
Downloading... success.

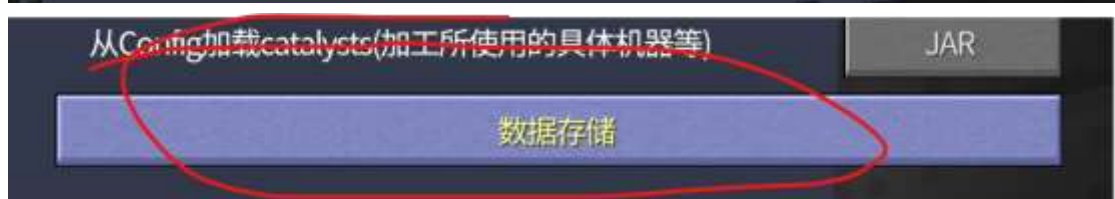
Installed successfully! plz read https://github.com/shooby009/assembly-line/blob/master/README.md, and edit conf/config.lua make it work
/home/assembly-line-nk2 #
```

1.使用前准备

1.1 建立流体数据

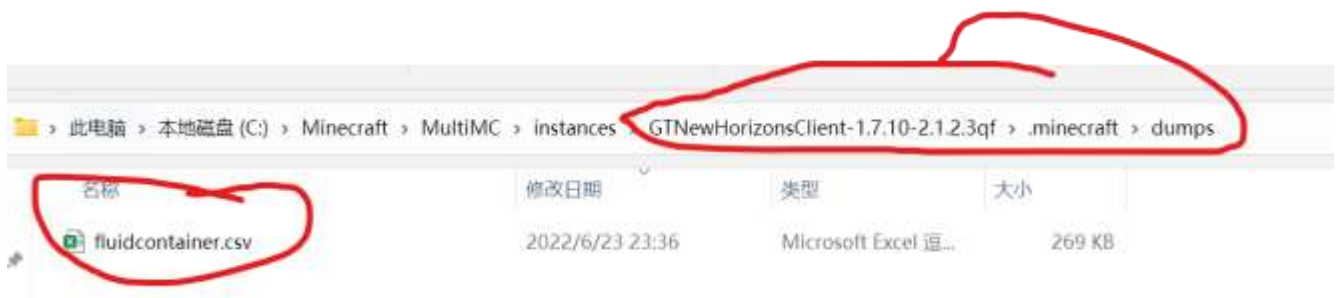
1. 按"e"打开 nei 界面，点击左下角扳手，并依次点击工具-数据存储-nei 整合中的流体容器项

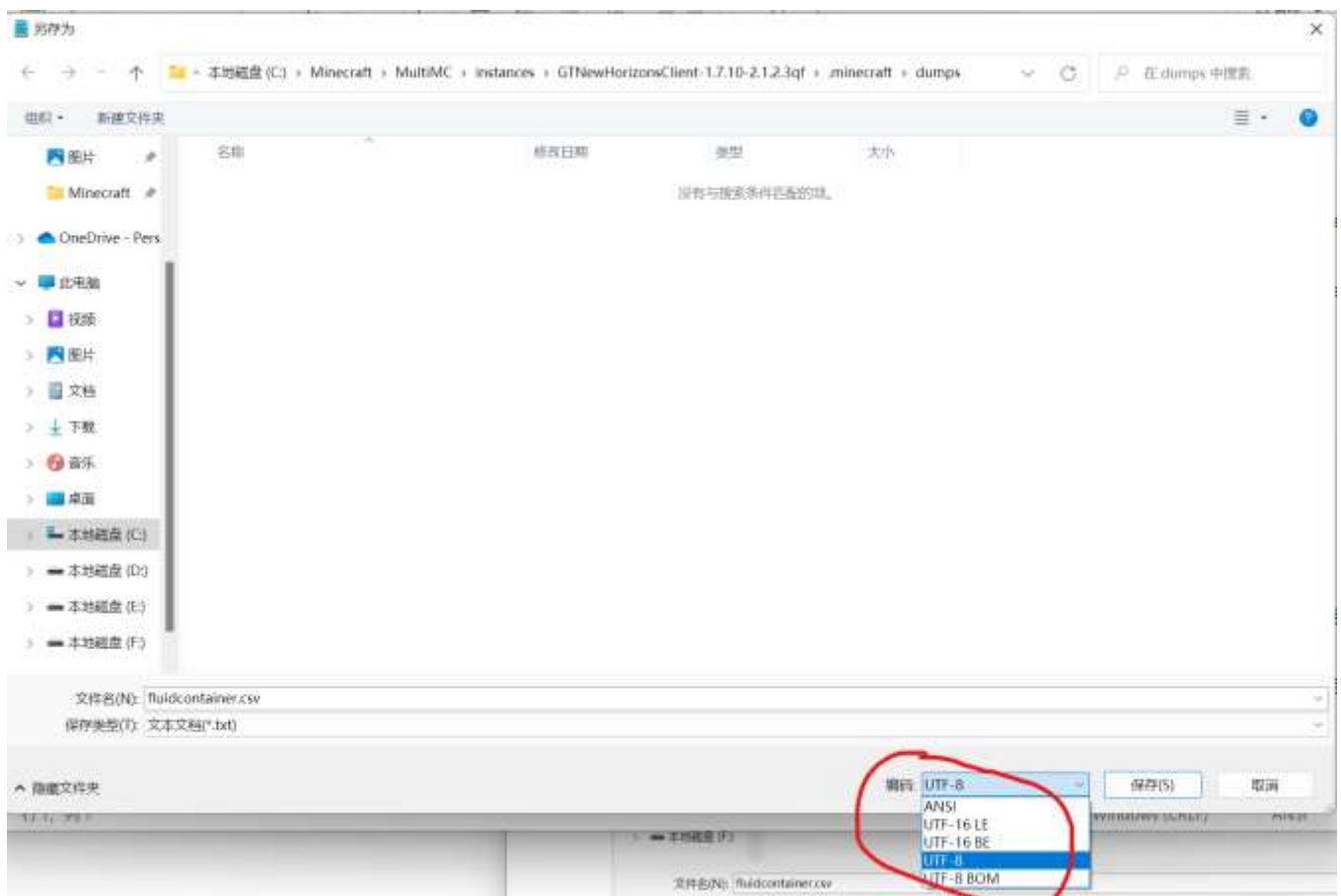
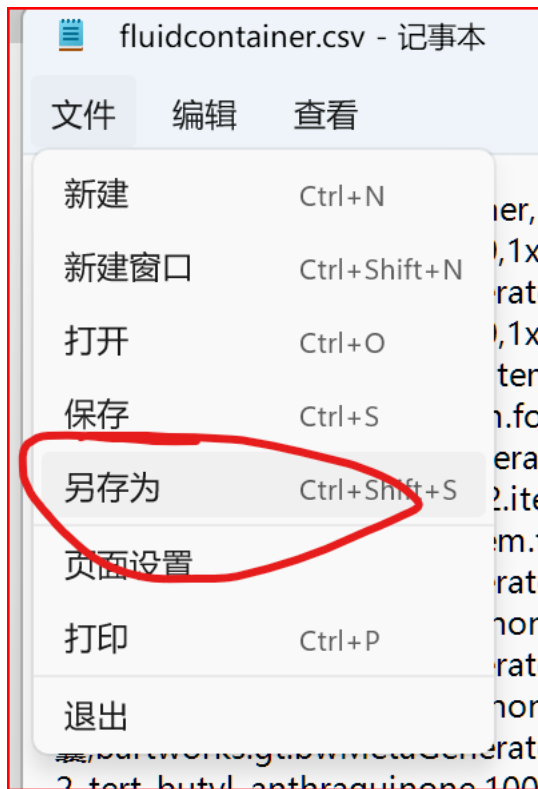






2. 在".minecraft\dumps\"文件夹中找到"fluidcontainer.csv"文件，使用记事本打开这个文件，以 UTF-8 格式另存，并剪切至其他文件夹备用。



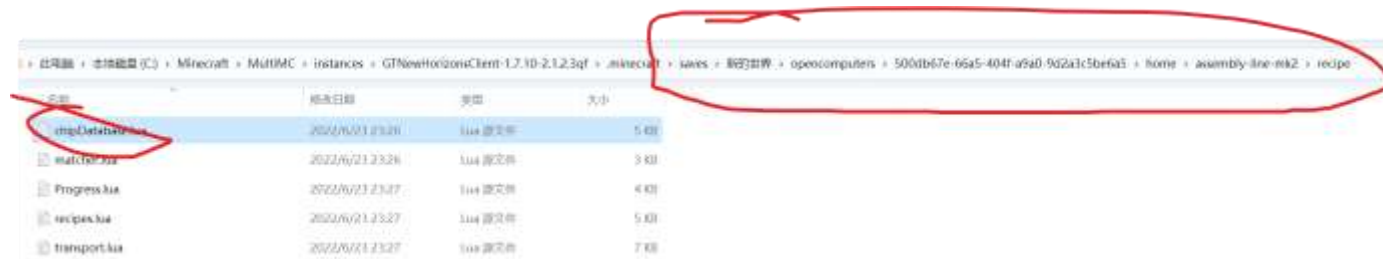


3. 将以上文件内容覆盖"工作文件夹\assembly-line-mk2\conf"文件夹中。



1.2 更新电路数据

打开"工作文件夹\assembly-line-mk2\recipe\"中的"chipDatabase.lua"文件，核对电路板的名称及物品 id。若您的单机端或服务端未使用 kiwi 对应版本的汉化，请自行修改对应电路板的名称。



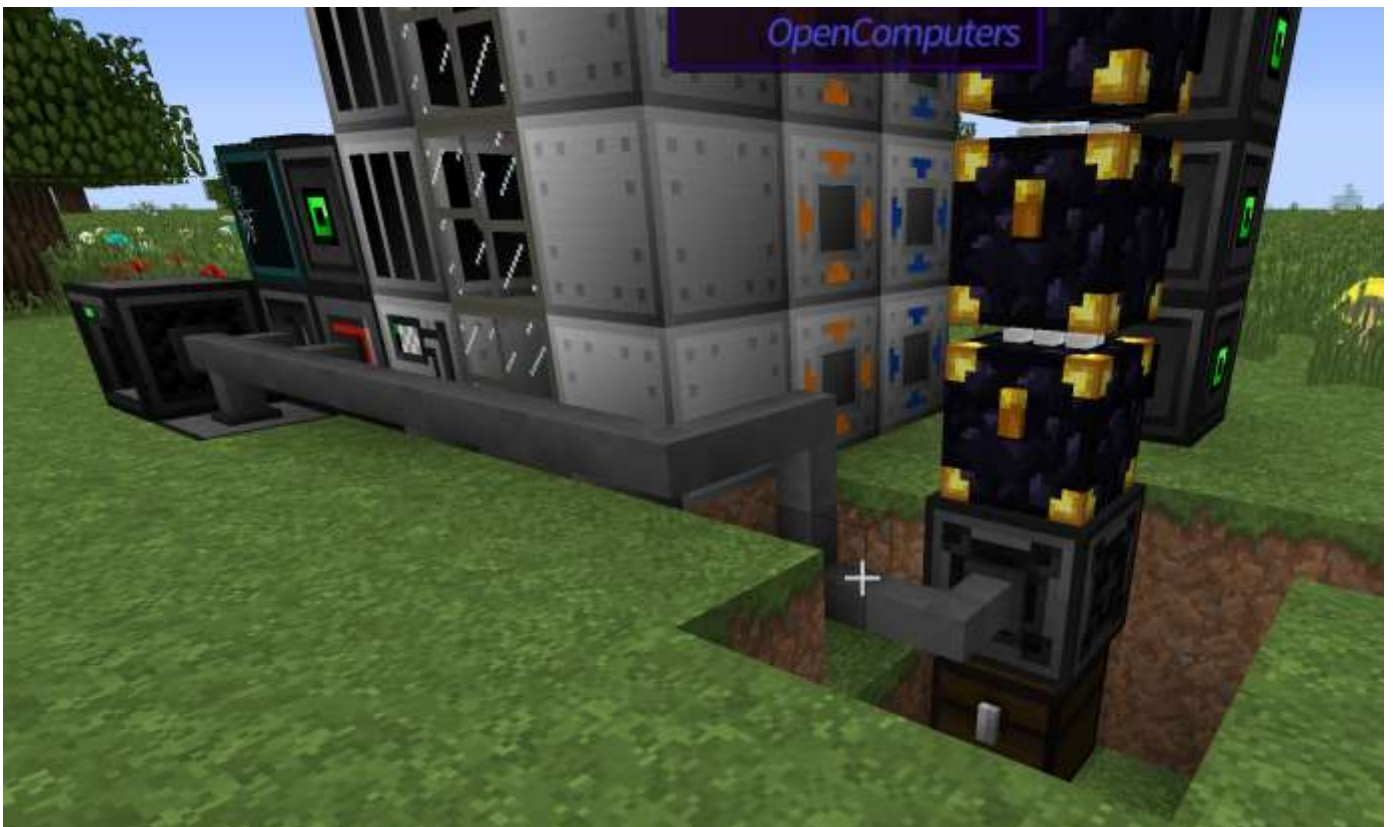
```
1  chipDatabase = {}
2
3  chipDatabase['工作站'] = 'ev'
4  chipDatabase['量子处理器'] = 'ev'
5  chipDatabase['纳米处理器集群'] = 'ev'
6  chipDatabase['数据磁共振电路'] = 'ev'
7
8  chipDatabase['gregtech:gt.metaitem.01.32704'] = 'ev'
9  chipDatabase['gregtech:gt.metaitem.03.32085'] = 'ev'
10 chipDatabase['gregtech:gt.metaitem.03.32083'] = 'ev'
11 chipDatabase['bartworks:gt.bwMetaGeneratedItem0.8'] = 'ev'
12
13 chipDatabase['处理器主机'] = 'iv'
14 chipDatabase['精英纳米电脑'] = 'iv'
15 chipDatabase['量子处理器集群'] = 'iv'
16 chipDatabase['晶体处理器'] = 'iv'
17 chipDatabase['精英磁共振电路'] = 'iv'
18
19 chipDatabase['gregtech:gt.metaitem.01.32705'] = 'iv'
20 chipDatabase['gregtech:gt.metaitem.03.32084'] = 'iv'
21 chipDatabase['gregtech:gt.metaitem.03.32086'] = 'iv'
22 chipDatabase['gregtech:gt.metaitem.03.32089'] = 'iv'
23 chipDatabase['bartworks:gt.bwMetaGeneratedItem0.9'] = 'iv'
24
25 chipDatabase['纳米处理器主机'] = 'luv'
26 chipDatabase['大型量子电脑'] = 'luv'
27 chipDatabase['湿件处理器'] = 'luv'
28 chipDatabase['晶体处理器集群'] = 'luv'
29 chipDatabase['大师磁共振电路'] = 'luv'
30
31 chipDatabase['gregtech:gt.metaitem.01.32706'] = 'luv'
32 chipDatabase['gregtech:gt.metaitem.03.32087'] = 'luv'
33 chipDatabase['gregtech:gt.metaitem.03.32092'] = 'luv'
34 chipDatabase['gregtech:gt.metaitem.03.32096'] = 'luv'
35 chipDatabase['bartworks:gt.bwMetaGeneratedItem0.10'] = 'luv'
36
37 chipDatabase['量子处理器主机'] = 'zpm'
```

2.初始化&配置

2.1 运行 initializer.lua 并按照顺序放置转运器,程序会自动写入地址到 config.lua

- > 1. 原材料箱子相邻的转运器
- > 2. 16 个位于输入总线下方的转运器
- > 3. 4 个流体转运器
- > 4. 4 个 me 流体接口
- > 设置完成后系统会自动重启
- > 再次说明顺序很重要中途有任何错误都会导致程序无法运行 如果放错请 ctrl+c 退出重新来过

```
/home/assembly-line-mk2 # initializer  
started!
```









```

config.chestInput.proxy = manager.proxy("fb0f9869-f0e5-4359-b2fe-b617a1bb8acc")
config.fluidInput[1] = manager.proxy("17923807-0fea-4208-920a-27b133b2382a")
config.fluidInput[2] = manager.proxy("73709726-5a49-439d-a554-dd99a3f5eeb5")
config.fluidInput[3] = manager.proxy("b3fcb460-75f6-4846-9069-62b76dc954e0")
config.fluidInput[4] = manager.proxy("3e76da19-96d3-47d0-98a3-46f68c99e1eb")
--ender chest
config.chestOutput[1] = manager.proxy("34bb194e-6be2-4f11-80e0-0403295d4183")
config.chestOutput[2] = manager.proxy("548d2d74-7300-4b3b-b467-fe32a834f329")
config.chestOutput[3] = manager.proxy("074b6674-02eb-44c4-b07d-cd8e6df018ef")
config.chestOutput[4] = manager.proxy("395e6e84-4606-4fca-801e-58f5736014ce")
config.chestOutput[5] = manager.proxy("96831f96-f57b-4185-9598-5cd3c3edcf06")
config.chestOutput[6] = manager.proxy("2b5e8afd-7c07-4558-9786-5d051db591c7")
config.chestOutput[7] = manager.proxy("0f5be201-3290-4931-b93c-b92cb23545f8")
config.chestOutput[8] = manager.proxy("c6ba9175-35a9-498d-b186-652dc0bfb808")
config.chestOutput[9] = manager.proxy("006db9b8-150a-403f-ba0e-b22781597e75")
config.chestOutput[10] = manager.proxy("da13f52b-c445-457a-8c9c-fd96652d36e9")
config.chestOutput[11] = manager.proxy("f7fc0640-629e-4926-9294-30c76f33e58d")
config.chestOutput[12] = manager.proxy("51dd4e5f-2260-4110-93e9-ef1638a0852d")
config.chestOutput[13] = manager.proxy("0eea4d71-306a-4a62-99f0-9d909750dec2")
config.chestOutput[14] = manager.proxy("61261525-8365-42a9-86d5-f6a61bac024a")
config.chestOutput[15] = manager.proxy("7fdaafd3-5565-4a64-9e10-cddd6cbd7899")
config.chestOutput[16] = manager.proxy("66cd47ca-7b42-4bec-b25f-9bd2fd7f4876")
--me fluid interfaces
config.fluidInterface[1] = manager.proxy("76e75cbe-91a5-4c8a-b42f-c91c4504d280")
config.fluidInterface[2] = manager.proxy("0e71a4d0-13d8-4ee7-990b-8e45414faac4")
config.fluidInterface[3] = manager.proxy("9d96fd95-4111-4a03-bfaf-7819536b72c2")
config.fluidInterface[4] = manager.proxy("${fif4}")

return config

```

2.2 设置转运器与原材料箱的面

```

cd assembly-line-mk2/
edit conf/config.lua

```

```

OpenOS 1.7.5 (2048k RAM)
Computers will consume less power while idling - i.e. when os.sleep(n > 0.05) is cal

/home # cd assembly-line-mk2/
/home/assembly-line-mk2 # edit conf/config.lua

```

>具体 sides 的定义查看官网 API <https://ocdoc.cil.li/api:sides>

```

config.chestInput.chestSourceSide = sides.top

```

2.3 设置转运器与材料输出到输入总线的箱子

```

config.chestInput.chestOutputSide = sides.west

```


2.4 设置转运器与熔物品的输出面

```
config.chestInput.moltenOutputSide = sides.north
```

2.5 设置流体输入输出面

```
config.fluidSourceSide = sides.bottom  
config.fluidOutputSide = sides.up
```

2.6 设置用于输入总线的转运器的输入与输出方向

```
config.chestOutput.chestSourceSide = sides.down  
config.chestOutput.chestOutputSide = sides.up
```

2.7 如果要用其他方法处理熔融流体，将下句中"true"改为"false"

```
config.moltenCtrl = true
```

2.8 其他 OC 组件

1. 紧贴装配线主方块放置一个红石 I/O 端口并接入 OC 网络，配套安装一个"设备活跃探测覆盖板"以探测工作进程，并根据方向读取方向（若不安装覆盖板，程序仍可运行，但会有一定显示错误）

```
config.redStoneSide = sides.east
```

2. 紧贴数据库接口放置一个适配器，在其中放置一个"物品栏控制器升级"，并根据方向配置数据库接口读取方向

```
config.flashSide = sides.north
```

3. 在其他适配器中放置一个"超级数据库升级"



2.9 保存并退出 config.lua 编辑，并重启

按“ctrl+S”键保存更改，再按“ctrl+W”键关闭编辑窗口，最后输入 reboot 命令重启

```
/home/assembly-line-mk2 # reboot
```

3.设置其他外围组件

3.1 AE 组件

紧贴任意一个适配器放置 ME 控制器，并连接 2.1 中放置的四个 ME 流体接口。
该 AE 子网中还需放置一个流体磁盘或其他流体存储设备,以暂存提取出的流体
该 AE 子网通过一个流体存储总线（只读模式）读取主 AE 网络 ME 流体接口，以使用主网络中流体

ME流体存储总线
设备关闭

✓ 目前能否挖掘

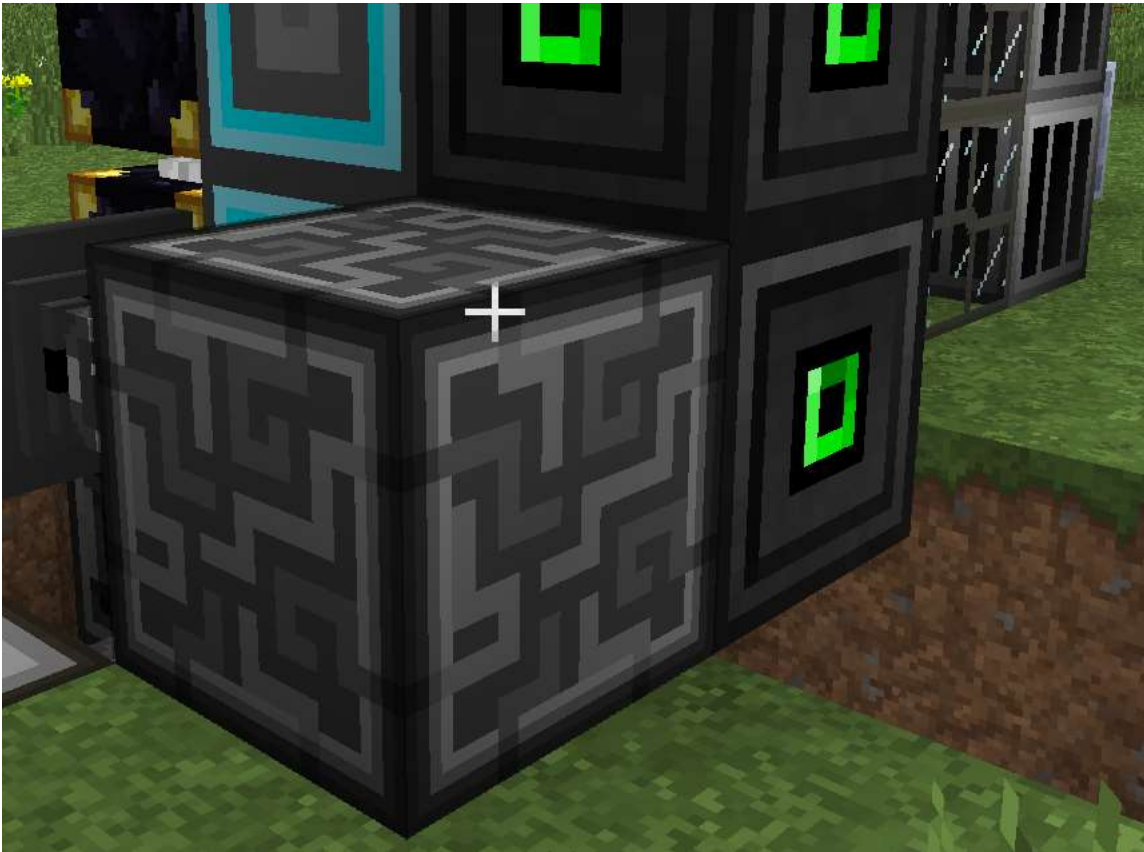
所需工具: 镐

硬度: 2.2

爆炸抗性: 2.2

Extra Cells





3.2 提取组件

提取出流体应输入子网中任意一个 ME 流体接口

进阶流体提取机 IV

X 目前能否挖掘

所需工具: **Wrench**

挖掘等级: **铁**

硬度: 1.0

爆炸抗性: 6.0

Progress: 0 s / 0 s

Facing: SOUTH

GregTech



4.设置流体到 db 中

所有配方中使用的流体均需设置，方法见后文 5.1

5.配置完毕

```
>cd assembly-line-mk2  
>  
>main (启动完毕后每 2 秒会到箱子里匹配物品)  
>  
>Ctrl + C (关闭程序)
```

6.util 说明

6.1 db.lua

```
    存储源材料箱第一个位置的物品到 database  
util/db  
    存储源材料箱所有物品到 database 新增数据(注意不能超过 81 个 如果要支持  
81 个要设置多个 db)  
util/db all  
    打印所有 database 中的数据  
util/db readAll  
    清除所有 database 中的数据  
util/db clearAll
```

6.2 readitem.lua

```
    打印源材料箱的所有物品名称+damage  
util/readitem  
    打印源材料箱第一个位置的物品信息  
util/readitem allInfo  
    替换源材料箱为储罐 然后放入流体读取流体信息  
util/readitem readFluid
```

注意点

1. 请使用服务器及至少两个超级组件总线来运行此程序,组件达 35 个, 需要组件总线来扩展
2. 请使用至少两根 T3.5 级存储器, 该程序对内存需求较大
3. 如果遇到流体不够的情况下程序会一直循环 直到 ae 中有足够的流体供给到 me 流体接口中
4. 安装完毕正常运行后不要拆除任何 oc 的组件 否则地址会更变 你需要手动修改或者拆除所有 oc 转运器和 me 流体接口 config.lua 需要重新下载 然后运行 initializer 进行安装
5. 熔融物品写配方时, 144mb 整数倍的请使用锭形式, 小于 144mb 的请使用多个螺栓形式
6. 若您遇到"too long without yielding"错误, 请将 OC 设置文件 OpenComputers.cfg296 行 timeout 数值适当放大, 默认为 5, 适当向大处调整