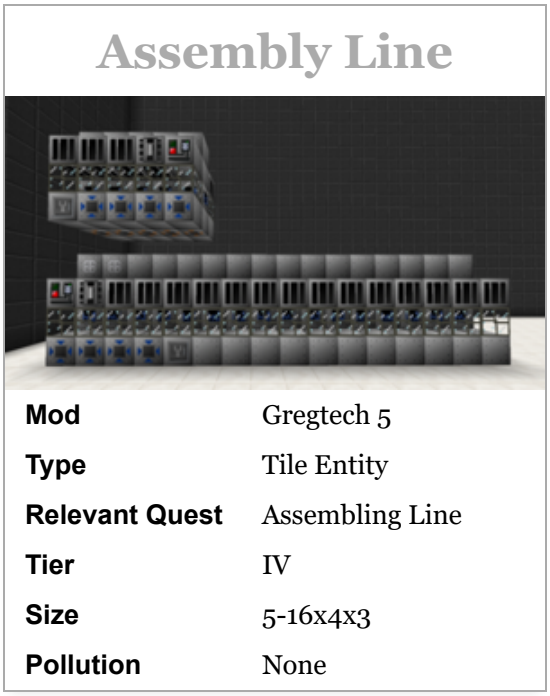
GT New Horizons

Assembling Line

The **Assembly Line** (assline or AL) is an IV tier multiblock required to make LuV+ tier components. Not only does it need input ingredients in a specific order, but the recipes themselves must be shared with the Assembly Line via Data Sticks. This is unlike any multiblock the player has encountered up to this point and requires more advanced techniques for automation. The Advanced Assembly Line is an upgraded version of this multiblock that allows parallel processing and higher overclocking, but no tier skipping.



Mod	Gregtech 5
Type	Tile Entity
Relevant Quest	Assembling Line
Tier	IV
Size	5-16x4x3
Pollution	None

Construction

The Assembly Line can be built left-to-right or right-to-left. Both are valid, but the direction will affect the ordering of the input ingredients. The slice with the controller is always the front and therefore the location of the first input ingredient. A Multiblock Structure Hologram Projector will show the player a minimum length Assembly Line, but holding multiple in a single stack can show/build different lengths up to the maximum of sixteen.

Requires:

- 1 Assembling Line (Controller) (either grate machine casing on layer 3 of the first slice)
- 5-16 Assembling Line Casing
- 5-16 Assembler Machine Casing[1]
- 8-30 Grate Machine Casing
- 8-41 Solid Steel Machine Casing[1]
- 10-32 Reinforced Glass, Borosilicate Glass (any), or Warded Glass
- 1 Data Access Hatch (any grate machine casing on layer 3)
- 1-16 Energy Hatch (any solid steel machine casing on layer 4)
- 1 Maintenance Hatch (any solid steel machine casing on layer 1)
- 5-16 Input Bus (bottom center of each slice)
- 1-4 Input Hatch (any solid steel machine casing on layer 1)
- 1 Output Bus (any solid steel machine casing on layer 1)

1. Do not mistake Assembler Machine Casings for Solid Steel Machine Casings when building. They have identical textures.

Slices

When building the Assembly Line, think of the multiblock as a series of *slices*. At the bottom center of each slice is an input bus for a single ingredient in the recipe. The number of slices required depends on the amount of ingredients in the recipe. For example, LuV motors have five solid input ingredients and therefore only need five slices to craft. The order in which the ingredients are inserted is the same order in which they appear in the NEI recipe.

Slice Construction, from Top-to-Bottom:		
▪ Layer 4: Empty, Solid Steel Machine Casing, Empty ▪ Layer 3: Grate Machine Casing, Assembler Machine Casing, Grate Machine Casing ▪ Layer 2: Reinforced Glass, Assembling Line Casing, Reinforced Glass ▪ Layer 1: Solid Steel Machine Casing, Input Bus, Solid Steel Machine Casing Front view (left) and side view (right)		

Data Access Hatches

There are four different types of data access hatches. It does not need to be directly adjacent to the controller despite what the tooltip says.

- The Basic Data Access Hatch holds up to four data sticks
- The Advanced Data Access Hatch holds up to sixteen data sticks
- The Automatable Data Access Hatch holds up to sixteen data sticks and allows for input/output of data sticks (not as good as it sounds)
- The Assembly Line Slave Connector reads any number of data sticks from a Data Bank (ZPM multiblock).

Buses and Hatches

Although the Assembly Line does NOT accept TecTech Multi-Amp Energy Hatches or Laser Target Hatches, it does accept up to 16 regular energy hatches to provide up to 32A of power. The Assembly Line can also tier skip ONE time if there are at least two energy hatches.

The input buses can be any tier, but typically ULV is used by automation methods because it only has a single item slot. The output bus should replace a solid steel machine casing on layer 1 rather than replacing an input bus on the last slice.

Only four input hatches are required because no Assembly Line recipe uses more than four fluids. Unfortunately, a quad input hatch does not work because the fluids need to be separated and inserted in the correct order much like the items. There can be gaps between the input hatches if necessary.

Usage

Scanning Recipes

Simply putting all of the ingredients for a recipe into the Assembly Line (and in the correct order) is not enough for the machine to start processing. The machine needs to be *taught* recipes via data sticks. The data sticks are obtained by scanning certain items in a Scanner.

Scanning an IV motor to obtain the data stick for LuV motors

For example, to learn the recipe for an LuV motor, the player must insert both an IV motor and data stick into a scanner as seen below. The output is a data stick specifically for LuV motors which is put into the Data Access Hatch on the Assembly Line. The IV motor will be consumed in the process. All data stick recipes and which items to scan for them are visible through NEI. Recipes may change with updates of GTNH which will require the player to make new Data Sticks for all changed items.

Later in the game, the scanner is replaced by the Research Station and Quantum Computer. These two multiblocks work together to learn the recipes for some of the most advanced items in GTNH.

Renaming Items

Some recipes in the Assembly Line have duplicated items but in different slots. For example, the Fusion Reactor Mk1 takes four ZPM circuits to craft but needs them to be in four different slices. This typically breaks any sort of automation method as it is incredibly difficult to know ahead of time whether to stack the circuits (as in other recipes) or not.

The solution is to rename any duplicate items so that they do not stack. Let us work through patterning the Fusion Reactor Mk1 as an example. Start with building an industrial material press and set it to forming press mode. Also craft a name mold and rename it to something like “Helper1”.

Place the name mold into an input bus on the industrial material press and cover it with an ME interface. The pattern(s) in this ME interface should have the original item (ZPM circuit) as an input and the renamed item (Helper1) as the output. Repeat this for however many helper items are required. Now the Fusion Reactor Mk1 can be properly patterned with a ZPM circuit, Helper1

Pattern for Helper1	
Input is the normal ZPM Circuit Output is the renamed ZPM Circuit	
Pattern for Fusion Reactor Mk1	
Three of the ZPM Circuits are replaced with Helper items	

Troubleshooting

No Valid Recipe Found

There are numerous reasons for this error such as missing items, missing fluids, ingredients in the wrong order, more than one output bus, etc. Check that all items are in the correct input bus/hatch.

No Data Sticks Found

The controller cannot find any data sticks. Either the data access hatch is empty or the data sticks are not scanned with any recipe. Holding shift will show what item the data stick is scanned with much like an AE2 pattern.

Automation

There are numerous methods to automate the Assembly Line, but only two will be discussed in detail here as they are the most prominent and easiest to implement with an AE2 auto-craft system. A prerequisite for both methods is the fluid processing pattern terminal which is the only terminal big enough to pattern all the ingredients in Assembly Line crafts.

Method 1: GregTech

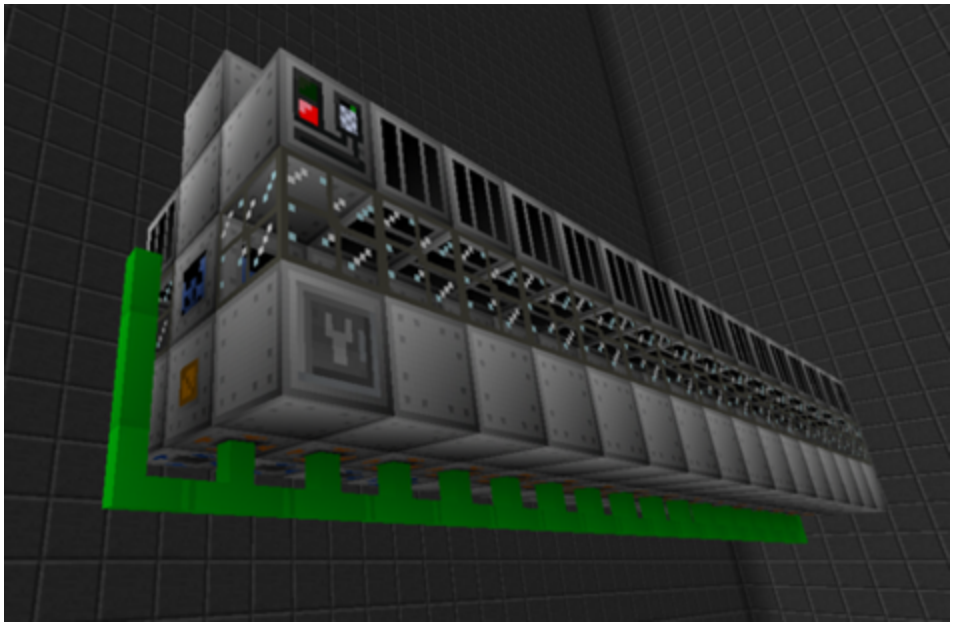
The core mechanic behind this approach is that GregTech item pipes insert into the closest inventories first *and* preserve the order specified by an AE2 pattern. Inputs are blocked using an item detector cover, machine controller cover, and shutter module to avoid recipes from being mixed.

STEP 1

Connect a high-throughput item pipe to all the (ULV) input buses

On the side of the first bus, attach an item detector cover with DEFAULT SETTINGS

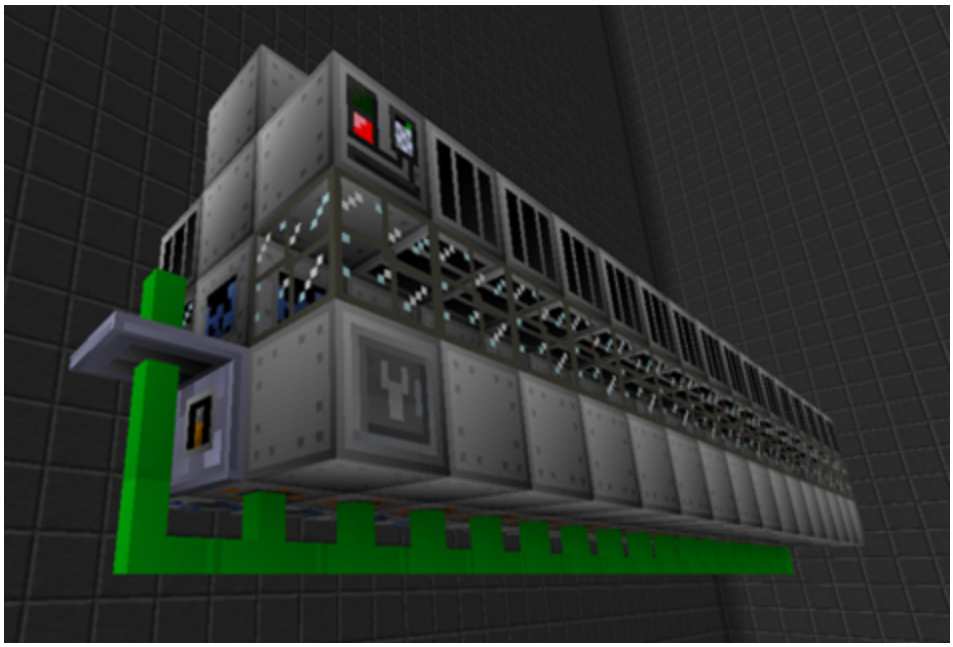
This will tell the shutter module that items are present in the assembly line because an item pipe with high enough throughput (>20 stacks per operation) will insert all ingredients simultaneously



STEP 2

Place a machine controller cover on the item pipe adjacent to the item detector cover and change the settings to DISABLE WITH REDSTONE

Place a shutter module on the same item pipe with DEFAULT SETTINGS

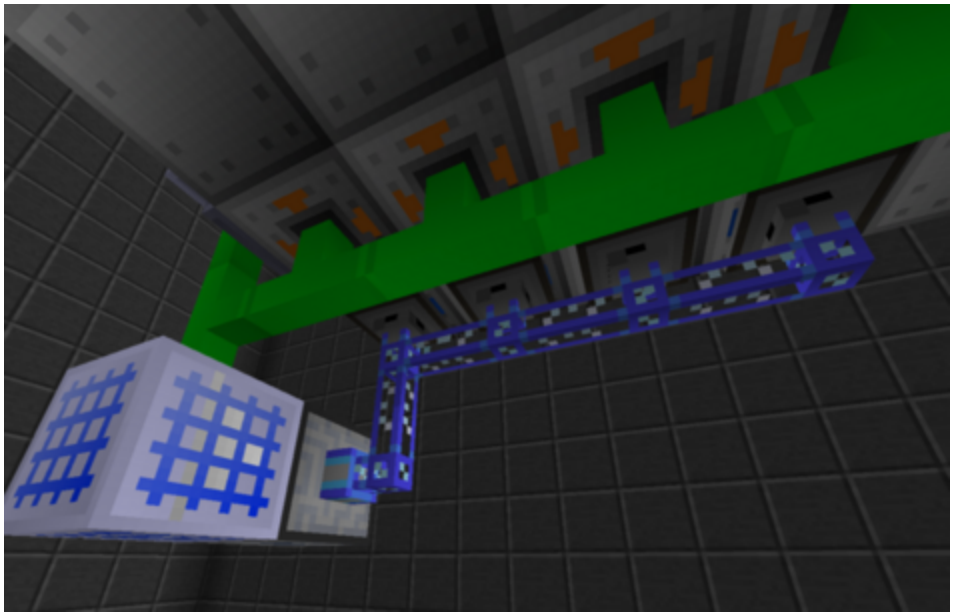


STEP 3

Place an item filter facing directly into a fluid packet decoder as shown

Place four fluid storage buses on the input hatches with DESCENDING PRIORITY from the first slice

This will force the fluid packets coming in through the item pipes to be stored in the input hatches

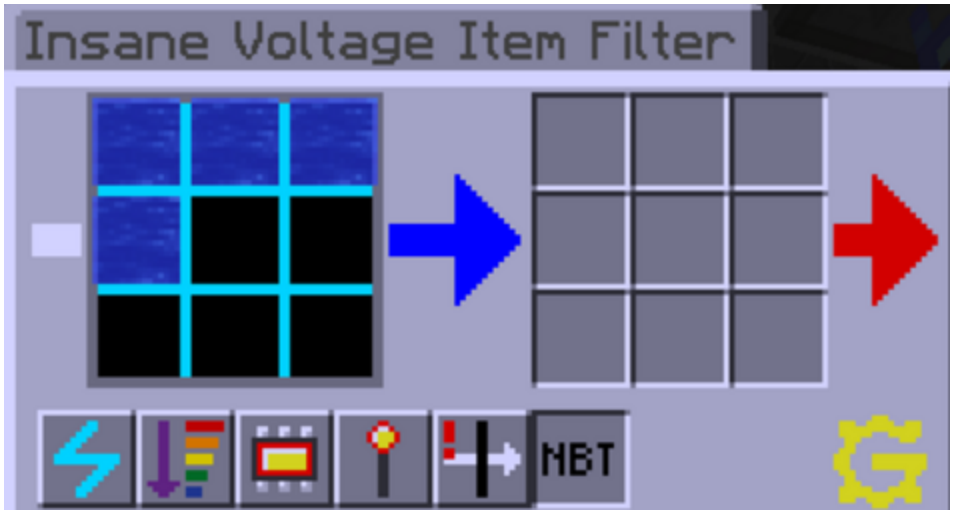


STEP 4

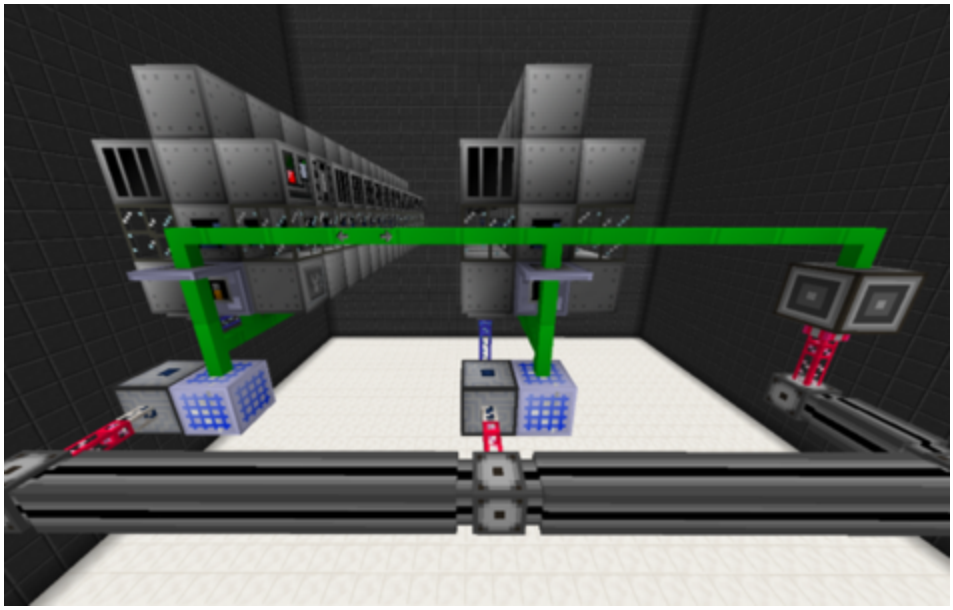
In the item filter, grab four fluid packets from NEI and select IGNORE NBT

This will allow any type of fluid packet to pass through this filter

An alternative to this is using flasks and super tanks, but that gets extremely tedious making patterns and requires bottling everything



STEP 5

Power the fluid packet decoder by connecting a quartz fiber cable to your main network (red) Connect a REGULAR ME Interface to the item pipes and enable BLOCKING MODE Parallelizing this approach is as simple as extending the item pipe with at least one restrictive item pipe between assembly lines	

Method 2: Applied Energistics

The core mechanic behind this approach is that AE2 storage buses with set priorities can determine the order in which items are inserted. Inputs are blocked using an advanced blocking card (GTNH v2.3.7+) to avoid recipes from being mixed.

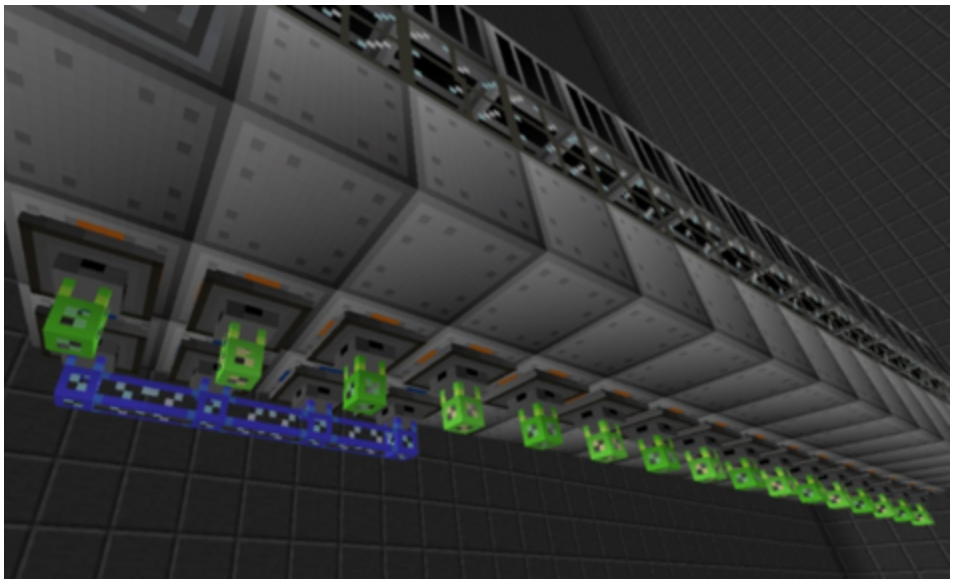
The benefits of this approach is that you will have nearly all the AE2 components needed later on when switching to the Advanced Assembly Line, using the advanced blocking card is very simple compared to the covers in method 1, and item throughput is higher.

STEP 1

Place storage buses on all the (ULV) input buses with DESCENDING PRIORITY from the first slice

Place fluid storage buses on all the input hatches with DESCENDING PRIORITY from the first slice

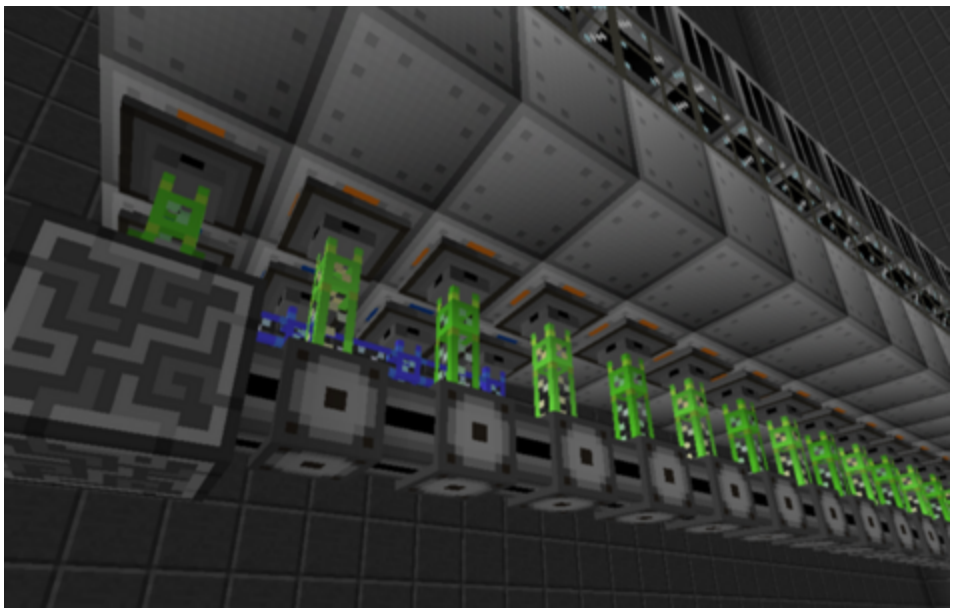
Alternate cable colors to support all the channels (green and lime in the figure to the right)



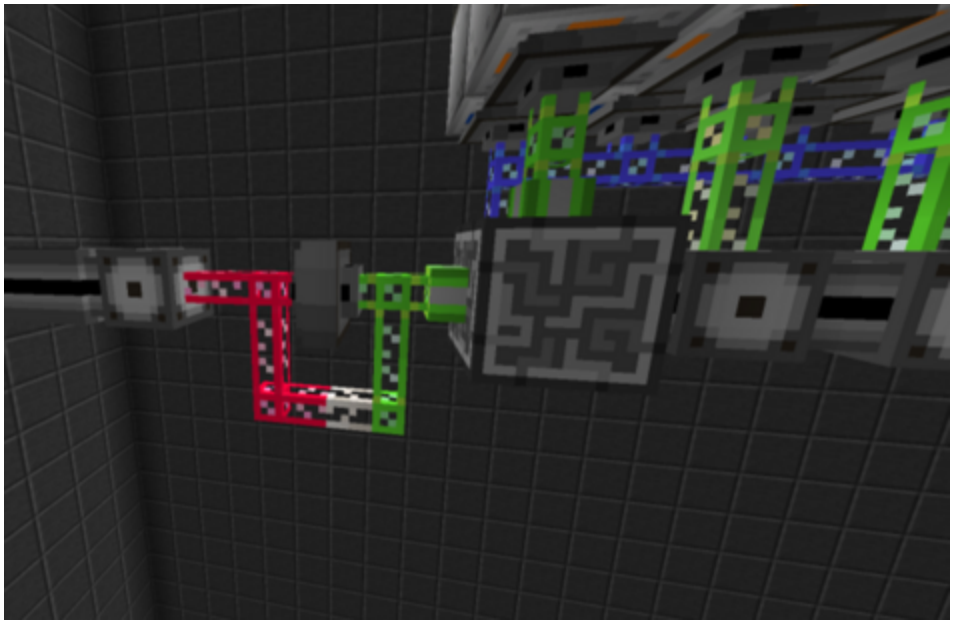
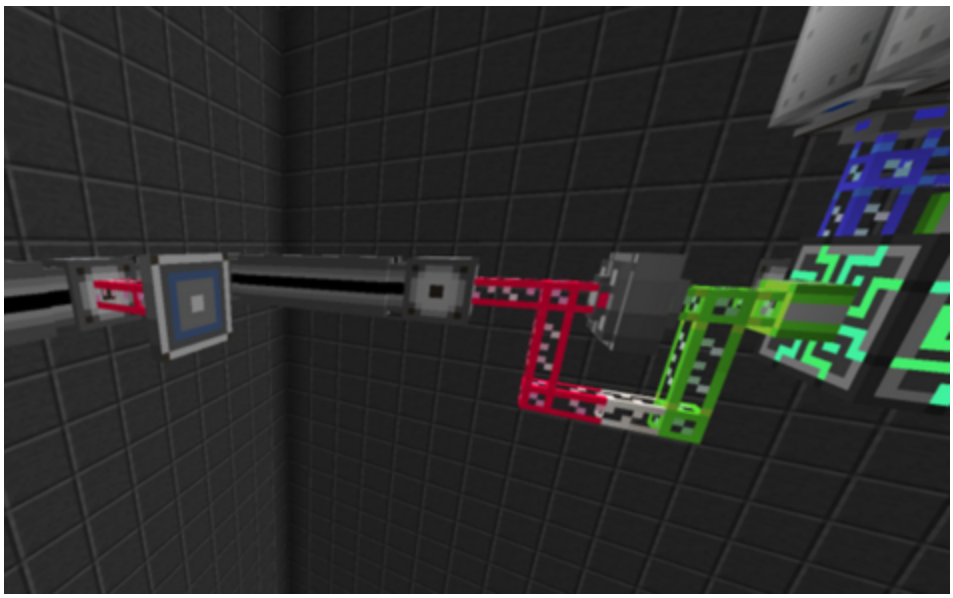
STEP 2

Connect all the storage buses to an ME controller via dense fluix cables

Every assembly line is its own AE2 subnetwork



STEP 3

On the subnetwork (green), place an ME Dual Interface with an ADVANCED BLOCKING CARD inside On the main network (red), place an output P2P tunnel – ME Dual Interface with BLOCKING MODE enabled Power the subnetwork by connecting a quartz fiber cable to the main network	
STEP 4	
Somewhere on the main network (red) should be an input P2P tunnel – ME dual interface with DEFAULT SETTINGS which should hold all the patterns Parallelizing this approach is as simple as linking more output P2P tunnels to this one input P2P tunnel More P2P tunnels can be added if more than 36 recipes are needed	

Archive

- Assembly Line Automation covers Steve's Factory Manager, OpenComputers and older AE2 setups.
- Assembly Line Automation - AE2 Decoder covers automation with AE2 using the fluid packet decoder (2.2.8+)

Retrieved from "https://wiki.gtnewhorizons.com/wiki/Assembling_Line?oldid=11619"