

# Tutorial: Redstone tips   re article feedback

---

< [Tutorials](#)

This tutorial gives general advice for building with redstone. There are a variety of pages discussing how redstone works, mostly collected in the page [Tutorial:Redstone](#).

## Contents

---

### [Planning](#)

[Size](#)

[Creative Mode](#)

[Gathering Resources](#)

### [Construction](#)

[Color coding](#)

[Troubleshooting](#)

### [Refining](#)

### [See also](#)

### [Videos](#)

### [Navigation](#)

## Planning

---

The first step in building a redstone circuit is to decide what it will do and how, in general, it will operate.

- How and where will it be controlled?
  - Will the circuit be controlled by the player, by mob movement, or something else?
- What mechanism components will it control?
- What is an efficient first design?
  - Although refinement often occurs in later stages of the build, starting on a strong foot to tackle the idea will be beneficial later on. Allowing an inefficient/flawed design to manifest can hinder development.
- How will the signal be transmitted from the controls to the mechanisms?
  - Will signals need to be combined from multiple sources?

## Size

When making redstone, it's important to make it a reasonable size.

You shouldn't use a huge amount of space for a single contraption. Large builds take up a lot of space and are inconvenient. However, you also shouldn't try to create a fully functional redstone circuit in a tiny area. Complex redstone circuits often need plenty of space to function. For example, you cannot create a redstone computer which can perform number operations and display multiple items on a screen at once in only 1 chunk.

For the best redstone results, make your contraption as small as you can with it still functioning, but if you find you're having any troubles with that size, make it bigger. Also, make sure to never underestimate how much time, space or materials you will need. It's much better to overestimate and bring more than you need so that you have extras for next time.

## Creative Mode

A complex redstone project for a Survival world can be designed in Creative mode first, before investing resources and effort in a survival world. It's handy to keep a creative-mode world handy for such laboratory work, usually a superflat with cheats on. You can also manipulate the game rules for your testing world to your liking, such as to make it permanent day or avoid mob spawning. Creative mode is great for building, because you have an infinite number of blocks, you can break blocks right away, and you can fly around to look all around your structures. You can also press F3 + N to invoke spectator mode, then fly through to look inside your circuit.

Once you have finished your redstone contraption and gotten it working, look it over to make sure you understand how it's working *now*. You may be able to make some improvements here. But eventually, you go back to your survival-mode world, gather the materials, and just copy your design from creative mode. Optionally, you can count how many of each material you used when building in creative mode, so that you will know exactly how much of a certain material to gather when in survival.

## Gathering Resources

When making very large redstone contraptions, you may need farms for renewable resources. Here are some materials you may need to farm:

- Redstone: There are only a couple of *renewable* sources of redstone: killing witches (witch and raid farms are rather slow) or trading with Clerics (which is even slower). However, it is fairly plentiful in the underground, especially once you have a Fortune III pickaxe to multiply its drops.
- String: A spider farm can help if your contraption includes a lot of tripwires and/or dispensers. Piglin bartering is an alternative
- Iron ingots (Make an iron golem farm if you need a lot of hoppers or minecarts)
- Slimeballs: (For sticky pistons and/or slime blocks)
- Honey Blocks: Used in some mobile constructions. Requires farming bees.
- Stone and Cobblestone: Smooth stone for repeaters and comparators, cobble for pistons, dispensers/droppers, etc.

- Nether Quartz: Plentiful in the Nether, can be renewably farmed with bartering. With it, you get to use comparators, observers, and daylight detectors.
- Glowstone: Used for Redstone Lamps, or lighting. Can be found fairly easily in the Nether, or it can be purchased from Clerics or Wandering Traders. Witches can also drop small amounts of the dust.

## Construction

---

It can be helpful to choose a specific set of blocks the player uses to construct circuits. Then, when the player runs into these blocks during the excavation of new rooms in the base, the player knows they are about to damage a previously-built circuit. Common choices include stone bricks, snow block, wool and concrete. (Using different colors of wool and concrete is also a great way to keep track of different circuits)

Be cautious when building circuits near water or lava. Many redstone components will "pop off" (turn into items) when washed over by liquids, and lava will destroy any items it contacts.

Be careful when building circuits to activate TNT (traps, cannons, etc.). Circuits in mid-construction can sometimes briefly power up unexpectedly, which might activate TNT. For example, placing a redstone torch on a powered block, it won't "realize" that it should be turned off until the next tick, will therefore be powered for one tick, and can briefly power another part of the circuit during that one tick. Placing TNT after the rest of the circuit is complete will help to avoid such problems and the destruction of the device itself. This also applies to any other features of the circuit that may be accidentally activated with such actions (e.g., activating a dispenser before the circuit is ready). Temporarily placing a redstone lamp or piston can quickly test whether a given space is powered.

## Color coding

This is a simple yet very effective tip, especially if you create redstone contraptions that have many different parts to them, such as comparator clocks mixed with other redstone items. It is best to use different colored wool, concrete, or terracotta for different parts of the circuit. If you build all the redstone using the same building block, for example, out of dirt (which you shouldn't be using for redstone anyway if you are in Survival because an Enderman may break it), soon you may completely forget how your redstone works due to not remembering where each circuit goes. Furthermore, this is important if you want to show off your redstone contraptions to others, so people can easily copy your design in their world. Additionally, color coding is helpful so you can go back to your project and understand what parts of the circuit perform what function.

If you don't want to use wool, concrete, or terracotta, you can find other blocks that are different colors from each other. For example, you can use stone variants and wood-related blocks. However, try not to use blocks of similar color, such as a block of coal and black concrete on 2 different parts of a circuit. You can also use different colors or variants to mark switch-supporting blocks (input) or potential output locations for a circuit, e.g, if most of the circuit is built on stone brick, you might choose carved stone brick for switch blocks, polished granite for output locations, and diorite for mobile blocks. Glass (which can also be tinted) can be used to display

the workings of a circuit; it can also make sure that lava and water (e.g., in a cobblestone generator) are visible but not open to unwary players.

All this may take extra time and effort, but the benefits are worthwhile.

## Troubleshooting

When the circuit isn't working the way it should, take a look at it and try to find the problem. Work through the circuit and test various inputs to find where a signal is "dropped" or gained inadvertently.

- What part of the wiring actually is not behaving as expected?. Unexpected output behavior is usually only a symptom, where the actual problem resides somewhere in the wiring.
- Are signals out of sync due to timing issues?
- Are parts of the circuit activating when they shouldn't be? Maybe accidentally "crossed wires" are allowing a signal from one part of the circuit to activate another part of the circuit, or a repeater's output is being cycled back into its input.
- Has the wiring been damaged by pistons pushing the wiring around?
- Trying to draw power from a weakly-powered block? Maybe a redstone repeater is needed to either strongly-power the block or to pull power out of it.
- Trying to transmit power through a non-opaque block? Replace it with an opaque block, or go around it.
- Was a short-circuit created and a redstone torch that should be powered is now burned out? Fix the short-circuit and update the torch to get things going again.
- Are pistons, dispensers, or droppers being indirectly powered when they shouldn't be?
- Is the circuit based on a tutorial from an older version of Minecraft which no longer works in the current version?

## Refining

---

Once the circuit is working, consider if it can be improved (without breaking it).

- Can the circuit be faster?
  - Shorter delays and pulses can make most circuits faster.
  - Reducing the number of components or distance a signal has to travel through can speed up the circuit.
- Can the circuit be smaller?
  - Can fewer blocks be used?
  - Is there a more efficient way of doing the same thing?
  - Can the redstone dust lines be shortened?
  - Are unnecessary components used?
- Can the circuit be more robust?
  - Will the circuit still work when activated by a very short pulse?

- Will the circuit still work when activated and deactivated rapidly in succession?
- If either of the above are a problem, can this be fixed by filtering the input?
- Can the circuit be damaged if unloaded? Be careful with constantly running clocks.
- Did an update create the opportunity for a better circuit? (e.g., comparators, locking repeaters, observers, etc.)
- Can the circuit be quieter?
  - Fewer sound-producing blocks (e.g. pistons, dispensers and droppers, doors, trapdoors, fence gates, and note blocks) will make your device more stealthy around other players.
- Can any lag be reduced? Machines with many redstone components frequently changing state can cause light, sound, particle, or update lag.
  - Redstone dust creates hundreds of block updates whenever it changes states, and the more the signal strength changes, the more block updates it will produce. Reducing the length of redstone lines will significantly decrease lag. An alternative to redstone dust lines is to use lines of rails which update observers, as rails create much less block updates than dust.
  - Hoppers and hopper minecarts especially try to do several things every tick (accept items pushed into them, push items into other containers, check for item entities above them). Powering hoppers to disable them when not in use or placing containers (such as composters) above them to disable their item entity checks can help to reduce lag.
  - Redstone torches and redstone lamps change their light level when they change state. Light changes can cause block light updates in *hundreds* of block tiles around each component. Concealing the component in opaque blocks or placing permanent light sources (torches, glowstone, etc.) nearby can reduce light updates. While light updates no longer create lag on since they are on a separate thread, excessive light updates can light suppress which lags other actions such as chunkloading.
  - Several redstone components produce particles (redstone torches, redstone dust, but especially fireworks fired from dispensers). Too many particles may overload *Minecraft's* particle rendering and then some particles may fail to render until old particles have disappeared.
  - Every time a block is moved by a piston, the piston makes many checks for movement and it produces block updates in its neighbors, so moving too many blocks at once can produce lag.

## See also

---






- [Tutorial:Mechanisms](#)
- [Tutorial:Advanced redstone circuits](#)
- [Redstone](#)
- [Redstone circuits](#)
- [Clock circuits](#)
- [List of redstone components](#)

## Videos

---

## Navigation























---

<b>Tutorials</b>	<a href="#">[hide]</a>
 <b>Introductory</b>	<a href="#">[show]</a>
 <b>General</b>	<a href="#">[show]</a>
 <b>Challenges</b>	<a href="#">[show]</a>
 <b>Constructions</b>	<a href="#">[show]</a>
 <b>Farming</b>	<a href="#">[show]</a>

## Mechanisms

[\[hide\]](#)

### Various devices

 [Redstone](#) ( [Circuits](#)  [Components](#)  [Dust](#)  [Mechanics](#)  [Tips](#))  [Automatic respawn anchor recharger](#)  [Combination locks](#)  [Command block](#)  [Crafter](#)  [Flying machines](#)  [Hopper](#)  [TNT](#)  [Item sorting](#)  [Item transportation](#)  [Mechanisms](#)  [Observer stabilizer](#)  [Randomizers](#)  [Redstone music](#) ( [Doorbell](#))  [Rube Goldberg machine](#)  [Shulker box storage](#)



### Detectors

 [Block update detector](#)  [Comparator update detector](#)  [Daylight detector](#)

### Minecarts

 [Train station](#)  [Minecarts](#) ( [Storage](#)  [Storage system](#))


### Traps

 [Snow golems](#)  [TNT cannons](#)  [Trapdoor uses](#)  [Trap design](#)  [Traps](#)











### Pistons

 [Piston uses](#)  [Piston circuits](#)  [Quasi-Connectivity](#)  [Zero-ticking](#)  [Instant repeaters](#)

### Tripwire

 [Tripwire techniques](#)

### Advanced redstone

 [Advanced redstone circuits](#)  [Arithmetic logic](#)  [Calculator](#)  [Command stats](#)  [Hourly clock](#)  [Morse code](#)  [Printer](#)  [Redstone computers](#)  [Redstone telegraph](#)  [360 degree ender pearl cannon](#)

## Multiplayer

[\[show\]](#)

## Technical

[\[show\]](#)

## Outdated

[\[show\]](#)

Retrieved from "https://minecraft.wiki/w/Tutorial:Redstone\_tips?oldid=3144962"

**This page was last edited on 7 September 2025, at 14:54.**

Content on this site is licensed under CC BY-NC-SA 3.0 unless otherwise noted; additional terms may apply.

Not an official Minecraft website. We are not associated with Mojang or Microsoft.