

Redstone circuits/Transmission

[Edit article feedback](#)[< Redstone circuits](#)

A **transmission circuit** is a [redstone circuit](#) that allows redstone signals to move from one place to another.

Contents

Signal transmission

[Transmission crossing](#)[Vertical transmission](#)

Diode

Repeater

[Instant repeater](#)[Two-way repeater](#)

Transmission encoding

[Analog](#)[Vertical analog transmission](#)[Analog inverter](#)[Binary](#)[One-hot](#)[Unary](#)

Wireless transmission

[Setblock transmission](#)[Scoreboard transmission](#)[Summon transmission](#)

References

Navigation

Signal transmission

Redstone signals can be transmitted from one place to another with **redstone wire** – a line of redstone dust. Redstone wire can transmit a signal only 15 blocks – after that it needs a [repeater](#) to boost the signal back up to full strength.

Transmission crossing

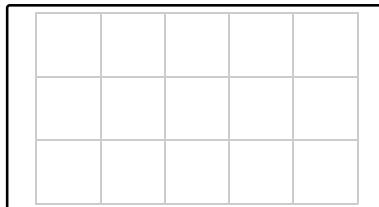
When crossing each other, redstone wires must be kept isolated so they don't interfere with each other.

Redstone bridge

1×3×4 (12 block volume)
1-wide, instant, silent
circuit delay: none

The fastest method for crossing wires is by building a bridge to take one wire over the other.

Variations: A common variation is to drop the center block one level, and cut a three-block passage into the ground under it for the north-south wire.



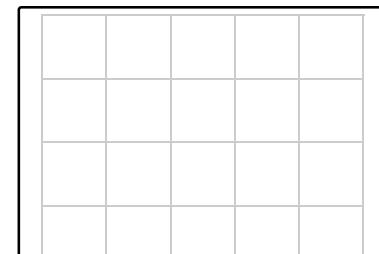
Redstone bridge ?

The center element consists of powered redstone dust on top of a block over unpowered redstone dust.

Repeater bridge

2×3×3 (18 block volume)
silent
circuit delay: 1 tick

A repeater bridge takes up less vertical space than a redstone bridge, but it adds 1 tick of delay to both wires.



Repeater bridge ?

Vertical transmission

While horizontal transmission can be relatively straightforward, vertical transmission requires trade-offs.

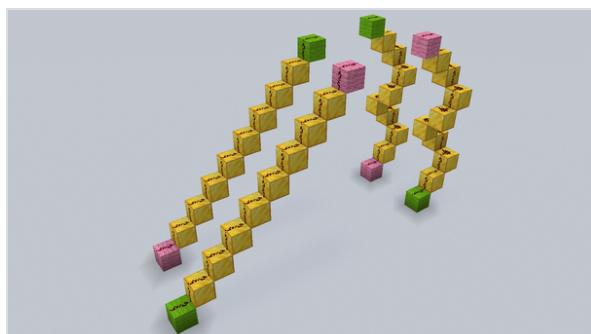
Schematic Gallery: Vertical Digital Transmission

Redstone staircase

Up or Down
1×N×N
1-wide, silent
circuit delay: 1 tick per 15 blocks

Redstone dust propagates a signal to adjacent redstone dust one block up or down as long as no opaque block "cuts" the signal. This allows "staircases" of blocks to carry redstone signals up (actual blocks of stairs aren't required, but can be used if the signal only needs to go up and they're placed upside-down).

Variation (Circular Staircase): By turning 90 degrees in the same direction each time the wire goes up a block, a "circular" staircase can be created in a 2×2 footprint. This variation is 2-wide tileable in both horizontal directions as long as the rotation direction is alternated in each direction (clockwise, anticlockwise, clockwise, etc.), or 2×4 alternating tileable with repeaters.



Redstone Staircase – [schematic]

Redstone ladder

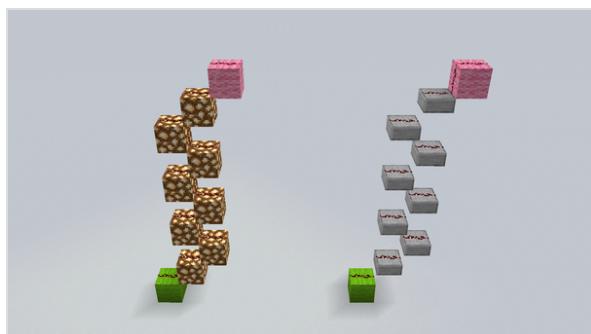
Up Only

1×2×N

1-wide, silent

circuit delay: 1 tick per 15 vertical blocks

Transparent blocks that support redstone dust do not "cut" redstone dust, so "ladders" of these blocks can be made zig-zagging back and forth upward. Glowstone and upside-down slabs are the most commonly used supporting blocks, but upside-down stairs, glass, and hoppers also can be used. Redstone ladders are 2×2 alternating tileable for short runs, or 1×4 alternating tileable with repeaters.



Redstone Ladder – [schematic]

Torch tower

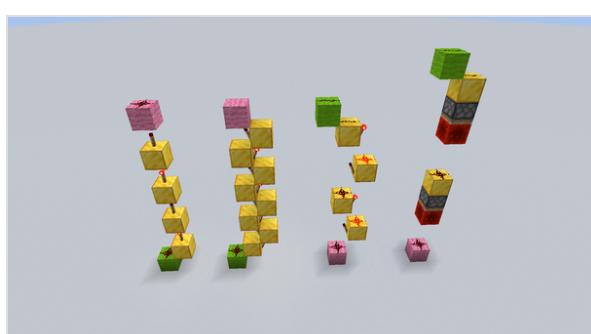
Up only

1×1×N

1-wide, silent

circuit delay: 1 tick per 2 vertical blocks

Redstone torches can power blocks above them, allowing transmission upward. Torch towers are 1×1 tileable.



Left: torch tower

Center Left: torch ladder

Center right: torch cascade

Right: piston tower

[schematic]

Torch ladder

Up Only

1×2×N

1-wide, silent

circuit delay: 1 tick per vertical block

Redstone torches can power redstone dust beneath them, allowing transmission downward. Torch ladders are 1×2 tileable upward but 2×2 alternating tileable downward.

Torch cascade

Down Only

1×2×N

1-wide, silent

circuit delay: 1 tick per 2 vertical blocks

Piston tower

Down Only[Java Edition only], *Up or Down*[Bedrock Edition only]

1×1×N

1-wide

circuit delay: 1.5 ticks per 5 vertical blocks (rising edge) and none (falling edge)

A sticky piston pointing downward can push a block of redstone into the space above redstone dust that is placed on top of a solid block. This can be repeated straight down, i.e. another sticky piston placed underneath that solid block, pointing downward, then another redstone block, a space, redstone dust on a solid block, and so on, allowing 1×1 downward transmission.

Because of the difference in rising and falling edge behavior, off-pulses are extended by 1.5 ticks per piston and on-pulses are shortened by 1.5 ticks per piston and may possibly be erased altogether. This makes piston towers less useful for rapidly changing states.

An upward-directed transmission using pistons is not possible in *Java Edition* due to effects of quasi-connectivity, unless slime blocks are used to move the blocks of redstone.

Piston-slime block tower

Up or down

1×1×N

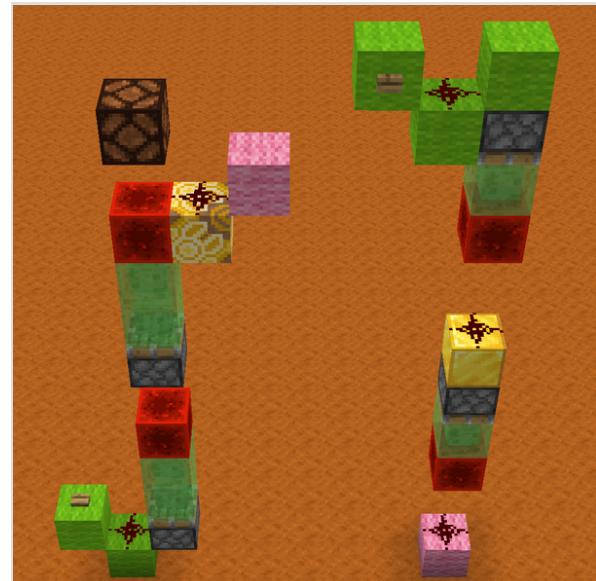
1-wide

circuit delay: 1.5 ticks per piston (rising edge) and none (falling edge)

This is a variation of the simple piston tower that uses slime blocks.

Using slime blocks together with a basic piston tower may drastically improve its performance, because less pistons are used for the same height in comparison to a simple piston tower, and piston moves more blocks. Up to 11 slime blocks per piston can be used.

If using more than two slime blocks per piston, the piston tower can transmit a signal upward. A simple piston tower can't transmit signals up because effects of quasi-connectivity cause the tower to freeze in its state and not disable upon falling edge. If using only one slime block per piston, an upward piston tower turns into a block update detector.[Java Edition only]



Left: upward piston-slime block tower

Right: downward piston-slime block tower
[schematic]

Combined upward ladder

Up Only

$1 \times 3 \times N$

1-wide, silent

circuit delay: 2 tick per 17 vertical block

A vertical transmission circuit can be made by combining both torch tower and redstone ladder, resulting in a ladder with maximum height and minimum delay, as seen in the schematic. Additionally, the first torch can be moved to the middle replacing the top slab and the torch with redstone wire, and adding another top slab before the second torch, which adds one block of height (as seen in the picture).

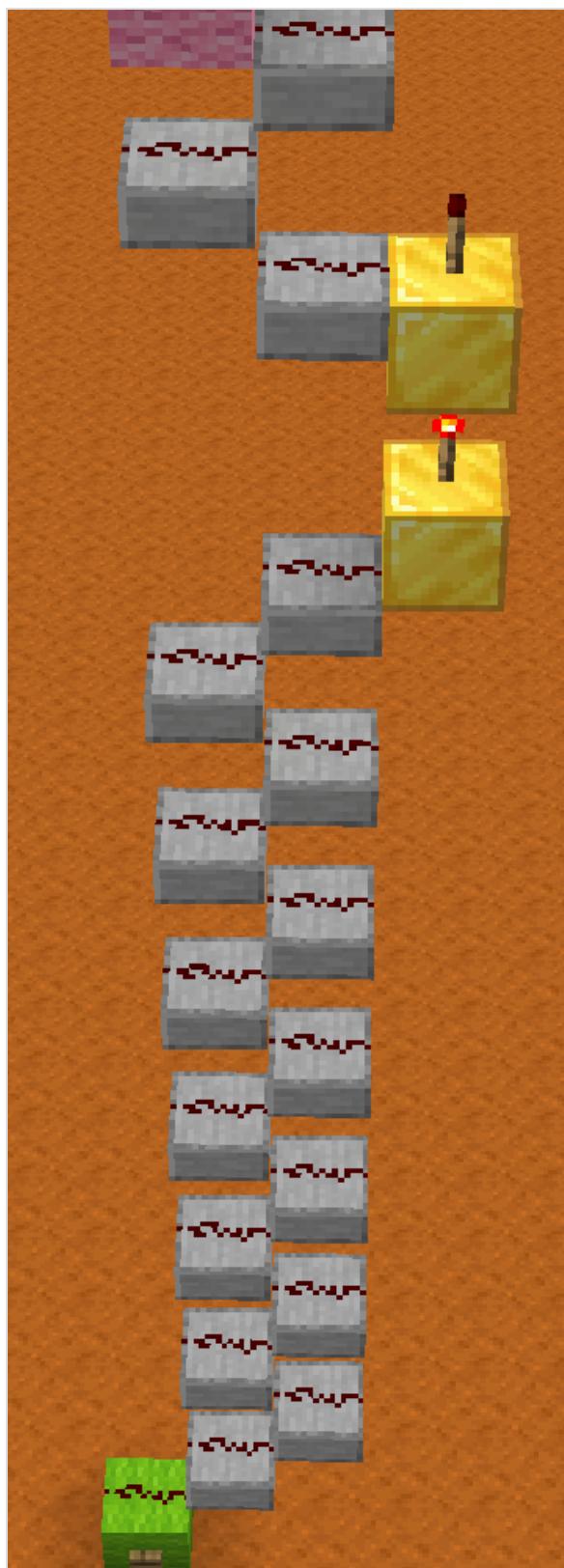
Observer wire

Observers can also be used to create a vertical ladder. Each observer carries block updates to the next observer up or down. A more advanced observer wire can be made by alternating, in upward order, observers, solid blocks, and either hoppers (only in Java Edition) or Droppers (in any edition). Each observer powers the dropper/hopper above it through the solid block, which changes its block state, activating the observer above it. The blocks can be removed to make a wire that is slower but does not power anything next to itself. Any of these can go horizontally or downward as well. There is also a more efficient downward method that goes (in downward order) observer, solid block, redstone dust, solid block, hopper/dropper.

Observers with walls/scaffolding

Starting from Java Edition 1.16, a Wall with a tall center post (due to a protruding wall block over it) causes all walls directly below it to also have a tall center post. In Java Edition, this change can be detected by observers, allowing for instantaneous downward transmission by moving a block over or away from the topmost wall.

Scaffolding provides the upward counterpart: Each scaffolding block tracks how far it is "overhanging", that is the distance from a column of scaffolding which is actually resting on a block. A trapdoor counts as support only when it is closed. Place a column of scaffolding atop a trapdoor, and also



Combined Upward Ladder – [schematic]

next to another scaffolding block that can support it while the trapdoor is open. When the trapdoor is closed, the column isn't overhanging at all, when the trapdoor is open it is overhanging 1 more space than its support scaffold. The difference propagates up the column, and can be detected by an observer.

(Credit for scaffolding technique: ianxofour via [Youtube](https://www.youtube.com/watch?v=TmqLXovhDOQ) (<https://www.youtube.com/watch?v=TmqLXovhDOQ>))

Observers with water

Water streams can carry information up and down. For a slow downward signal, a dispenser can place and pick up a water block (or even a lava block), and an observer can detect the changes in the flow downstream. For upward signals, a column of water source blocks can have the solid block below it swapped out by pistons; switching to a magma or soul sand block instantly propagates a bubble column up the water column, while switching to a normal block removes the bubbles; an observer watching one of the water blocks detects these changes.

Falling items

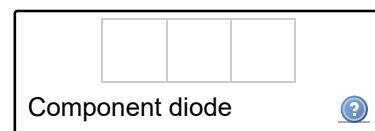
A dropper can toss items down a shaft or into a water stream; the items can then land on a wooden pressure plate, or be picked up by a hopper and then detected by a comparator. It is also possible to launch items upward by various means, but that is generally less reliable, as controlling their exact path can be tricky.

Diode

Another important aspect of signal transmission is making sure a signal doesn't go the wrong way. A "diode" is a redstone component or circuit that allows signals through in one direction but not the other.

Component diode

1×1×2 (2 block volume)
1-wide, flat, silent
circuit delay: 1 tick

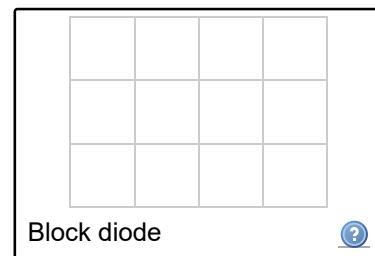


Both the [redstone repeater](#) and the [redstone comparator](#) transmit signals in only one direction, but add 1 tick of delay.

Block diode

1×2×2 (4 block volume)
1-wide, flat, silent
circuit delay: 1 tick

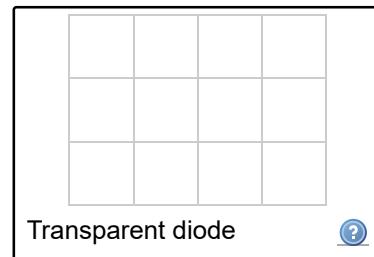
By strongly-powering a block, a signal can transmit in only one direction. None of the output lines can affect each other.



Transparent diode

1×2×3 (6 block volume)
1-wide, instant, silent
circuit delay: none

Some transparent blocks can support redstone dust: hoppers, glowstone, upside-down slabs, and upside-down stairs. These blocks have the property that redstone dust on them can propagate signals diagonally upward, but not diagonally downward (transparent blocks that cannot support redstone dust cannot be used for this purpose). Thus, simply jumping the signal up one block to one of these transparent blocks creates a diode circuit.



Upside-down slabs are the transparent block most commonly used for this purpose, but glowstone is used where light would be useful (to suppress mob spawning, etc.), upside-down stairs can be used where a full-side solid surface is required without light (for example, alongside a water channel transporting items over ice), and hoppers may be used in this way where they are already being used for item transport.

To get the output back to the same level as the input, run the line over an opaque block before dropping it.

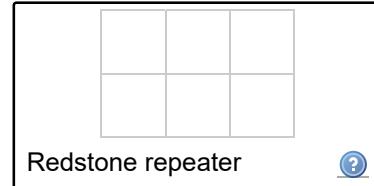
Repeater

To "repeat" a signal means to boost it back up to full strength. When redstone signals are transmitted through redstone dust, their signal strength fades and must be repeated after 15 blocks. Repeater components and circuits keep signals going over long distances.

Redstone repeater

1×1×2 (2 block volume)
1-wide, flat, silent
circuit delay: 1 to 4 ticks

The most basic and common method of repeating a signal is to use a redstone repeater.



When transmitting signals over long distances, it is more efficient to use a block before and after the repeater – this method of repeating a signal averages 18 redstone used per 18 blocks (15 redstone dust, and 3 redstone per repeater) and 1 tick delay per 18 blocks.

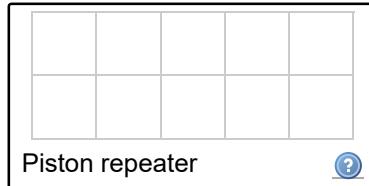
Piston repeater

1×3×2 (6 block volume)
1-wide, instant (falling edge)
circuit delay: 1 tick (rising edge), 0 ticks (falling edge)

A sticky piston can push a block into position to power the output.

Because of the differences in rising and falling edge delays, pulses are shortened by 1 tick

per piston repeater (possibly erasing short pulses). On the other hand, zero falling edge delay makes this repeater attractive for applications that need instantaneous falling-edge transmission but do not care about rising-edge delay—for example, a distant circuit that activates on a falling edge.



A piston repeater cannot handle pulses shorter than 1.5 ticks [*Java Edition* only]; with shorter pulses, the block is left behind (not retracted by the sticky piston) and continues to power the output until a later input pulse ends (Note: A piston repeater *can* handle pulses shorter than 1.5 ticks in *Bedrock Edition*.)

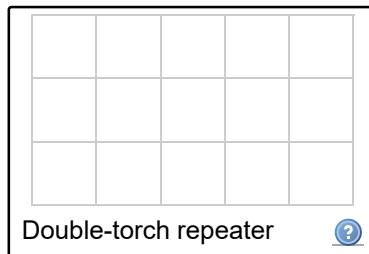
Variations: When transmitting signals over long distances, it is more efficient to place a block before the piston. This method of repeating a signal averages 17 redstone used per 19 blocks (1 for the piston, 1 for the torch, and 15 redstone dust) and 1 tick delay per 19 blocks.

The moving block can be replaced with a block of redstone, which allows the removal of the lower block and redstone torch, reducing the circuit size to a 1-high 1×3×1 (3 block volume).

Double-torch repeater

1×3×2 (6 block volume)
1-wide, silent
circuit delay: 2 ticks

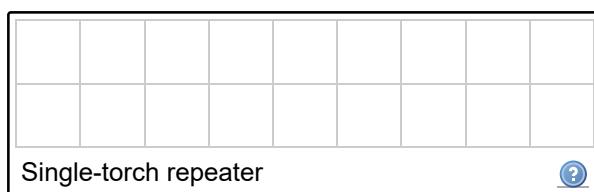
The double-torch repeater was the standard repeater circuit used before redstone repeater blocks were added to *Minecraft*.



In transmission lines, one double-torch repeater is required every 18 blocks (the 3-block circuit, plus 15 blocks of redstone dust), using 18 redstone per 18 blocks and adding 2 ticks delay per 18 blocks.

Single-torch repeater

1×2×1 (2 block volume)
1-high, 1-wide, flat, silent
circuit delay: 1 tick



When crossing long distances, redstone torches can be used singly, simply allowing the signal to be inverted an even number of times during its journey. Single-torch repeaters use slightly less redstone than redstone repeaters (16 redstone per 17 blocks) but are slightly slower (1 tick delay per 17 blocks).

Instant repeater

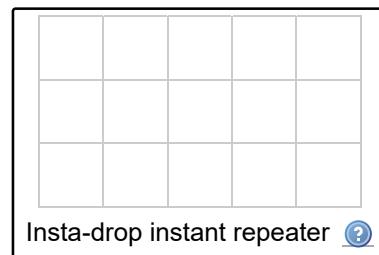
An **instant repeater** is a circuit that repeats a redstone signal change with no delay. A sequence of instant repeaters and redstone dust lines is known as **instawire** (or "instant wire").

See also: [Instant Two-Way Repeater \(below\)](#) and [Tutorial:Instant repeaters](#)

Insta-drop instant repeater [Java Edition only]

1×3×2 (6 block volume)
1-wide, instant
circuit delay: none

Behavior (rising-edge): While the input is off, the block of redstone keeps the lower sticky piston activated by connectivity. When the input turns on, the upper sticky piston begins to extend the block of redstone. The instant the block of redstone starts moving, the lower sticky piston deactivates and begins to retract block A, the reason the upper piston is extending — this turns the upper sticky piston's extension into a 0-tick extension/retraction (the "insta-drop": the sticky piston "drops" its grip on the block and leaves it behind when it retracts), leaving the block of redstone above the lower sticky piston and powering the output. All of this happens instantly (in the same redstone tick), effectively allowing a rising edge to pass through the circuit with no delay. Now that the block of redstone is above the lower sticky piston, the lower sticky piston extends again, and two ticks later block A is back in position causing the upper sticky piston to extend again, ready to retract block A when the signal turns off.



Insta-drop instant repeater [?](#)

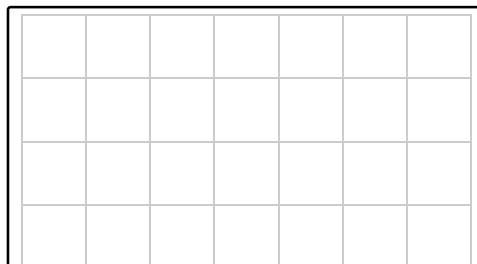
Behavior (falling-edge): When the input turns off, the upper sticky piston begins to retract the block of redstone, immediately cutting off power to the output, effectively allowing the falling edge to pass through the circuit with no delay. While the block of redstone is moving, the lower sticky piston deactivates, but then activates again when the block of redstone stops moving and can activate the lower sticky piston by connectivity again.

Earliest known publication: February 14, 2013^[1]

Dust-cut instant repeater

1×5×4 (20 block volume)
1-wide, instant
circuit delay: none
min input pulse length: 2 rs-ticks

Behavior (rising-edge): When the input turns on, the lower sticky piston begins to extend, causing the upper sticky piston to retract, allowing the powered redstone dust below block A to connect to the output. All of this happens instantly (in the same redstone tick), effectively allowing a rising edge to pass through the circuit with no delay. The moving block of redstone also instantly depowers the dust below it, but by the time that turns off the repeater's output, the block of redstone has arrived to continue powering the output.



Dust-cut instant repeater [?](#)

The space under the first piston prevents the block of redstone from activating its own piston.

Behavior (falling-edge): When the input turns off, the lower sticky piston begins retracting the block of redstone, immediately cutting off power to the output, effectively allowing the falling edge to pass through the circuit with no delay. The block of redstone then arrives at its retracted state and tries to power the output dust again, but it also powers the piston above it and block A arrives to cut the output before the repeater can output the signal from the block of redstone.

Variation (2-wide): The two upper levels (including the dust on top of the block the repeater is facing) can be moved one block over and down, and the last block on the lower level and its dust removed, to make a 2-wide version that is shorter in height and length (but larger in volume: $2 \times 4 \times 3$, 24 block volume). In this version, to reduce the amount of redstone used, the block of redstone can be replaced with a regular block if redstone torches are placed under both its extended and retracted position.

Earliest known publication: January 3, 2013^[2]

Two-way repeater

A **two-way repeater** (aka "2WR", "bi-directional repeater") is a circuit that can repeat a signal in either direction.

Two-way repeaters have two inputs that also act as outputs.

Typically the problem to be solved in design is repeating the signal in either direction without repeating the signal back into the same input, which could create a clock, a RS latch or a permanently-powered repeater loop.

Each circuit description below lists a **transmission speed**, the rate at which multiple circuits can transmit signals when placed at maximum distance from each other. Most circuits have their inputs offset from each other by one or two blocks – moving the wires in-line with each other reduces the transmission speed (because the signal has to move sideways to get to the correct input).

Current designs also have a **two-way reset time** – when input from one side is turned on, and then input from the other side is also turned on, then the first input turned back off, there is a short time while the transmission on the first side remains off until the circuit can reset itself to allow the second input through. Thus, the reset time can be seen as a spurious off pulse in a line that should be on.

Schematic Gallery: Two-Way Repeater

Comparison two-way repeater

2×5×2 (20 block volume)
flat, silent
transmission speed: 8 blocks/tick
circuit delay: 2 ticks
fastest clock signal: 2-clock
two-way reset time: 4 ticks

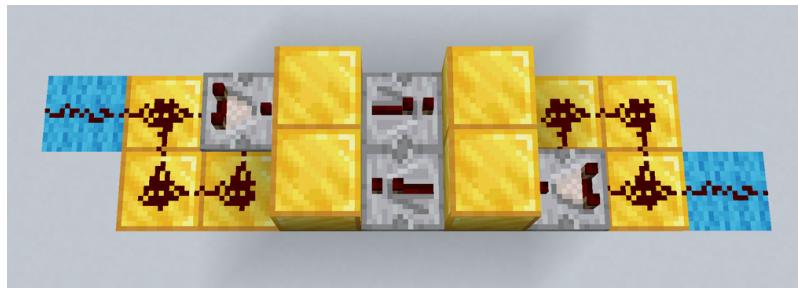
When a signal comes in from either side, it blocks the other input by providing a strength 15 signal to its comparator's side.

It is possible to override or block this circuit with additional inputs from the comparators' other sides.



Comparison two-way repeater – [schematic]

Variations: Transmission speed can be increased by lengthening the circuit. Possibilities include placing opaque blocks before and after the repeaters, adding a segment of analog comparator wire before the repeaters, or using slab diodes to allow placing blocks before the comparators.



Comparison 2WR
2×7×2 (28 block volume)
9 blocks/tick



Comparison 2WR
2×9×2 (36 block volume)
10 blocks/tick



Comparison 2WR
2×13×3 (78 block volume)
10.5 blocks/tick

Earliest known publication: February 16, 2013^[3]

CodeCrafted's two-way repeater

2×6×3 (36 block volume)
silent
transmission speed: 9.5 blocks/tick
circuit delay: 2 ticks
fastest clock signal: 3-clock

two-way reset time: 3 ticks

The output on each side is produced by the redstone torch under the block, held unpowered by the input torch from the other side. When the other input signal turns on, the output torch turns on – this also turns off the input torch holding the other output torch off, but each output torch also holds the other output torch off, keeping the circuit from becoming permanently powered.



CodeCrafted's two-way repeater – [\[schematic\]](#)

Variations: If it's not necessary to get the signal back to the lowest level (such as if this is built in a 1-deep hole), then this circuit can be considered to be 2×4×3 (24 block volume) and thus only four blocks long.

Earliest known publication: August 9, 2012^[4]

Instant two-way repeater

4×4×3 (48 block volume)

instant

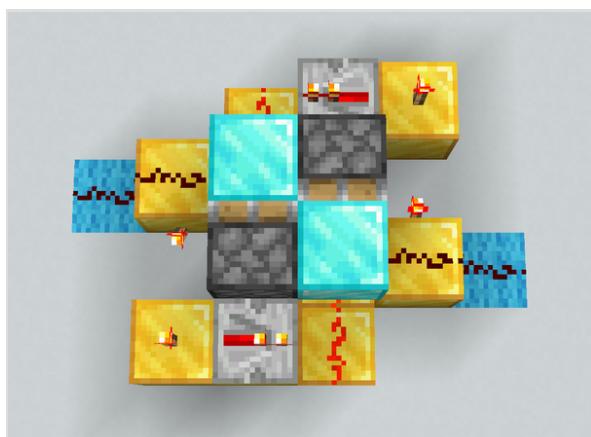
transmission speed: instant

circuit delay: 0 ticks

fastest clock signal: 2-clock

two-way reset time: 2.5 ticks

When an input turns on, it (a) turns off the torch on the side of the block and (b) powers the block in front of the input, activating the sticky piston on the other side. When the piston starts moving its block, this instantly allows the powered dust underneath to connect to the output. By the time the power from the torch and repeater have turned off, the block has arrived at its extended position where it connects the power from the other torch and repeater to the output.



Instant two-way repeater – There is redstone dust under the blocks of diamond, and 1-tick repeaters under the sticky pistons facing away from the bottom torches. [\[schematic\]](#)

Earliest known publication: February 18, 2013^[5]

Moved-block two-way repeater

2×5×2 (20 block volume)

flat

transmission speed: 12 blocks/tick (18 blocks per 1.5 ticks)

circuit delay: 1.5 ticks (rising edge) and 0 ticks (falling edge)

fastest clock signal: 3-clock (but shortens pulses)

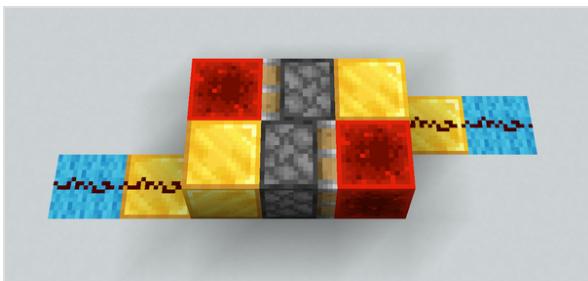
two-way reset time: 1.5 ticks

When an input turns on, a sticky piston pushes a block of redstone into position to power the other line, but that also reconfigures the dust on the other side to prevent it from powering the

other sticky piston.

Because of the difference in rising and falling edge delays, pulses is shortened by 1.5 ticks per two-way repeater.

Earliest known publication: September 8, 2013^[6]

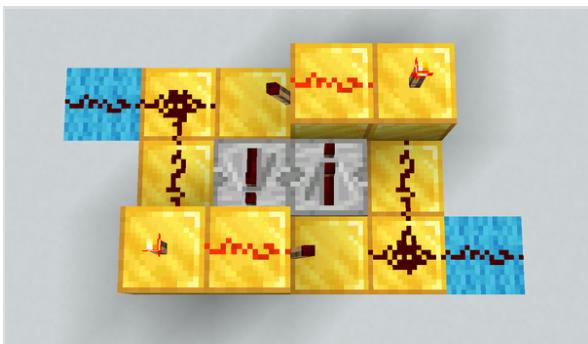


Moved-block two-way repeater – [schematic]

Classic two-way repeater

3×4×3 (36 block volume)
silent
transmission speed: 8 blocks/tick
circuit delay: 2 ticks
fastest clock signal: 3-clock
two-way reset time: 4 ticks

This design offers few advantages over the other designs, but may be of historical interest.



Classic two-way repeater – [schematic]

Locked-repeater two-way repeater

3×4×2 (24 block volume)
flat, silent
transmission speed: 15 blocks/tick
circuit delay: 1 tick
fastest clock signal: 1-clock
two-way reset time: 3 ticks

When a signal comes in from either side, it blocks the other input with a repeater lock.

This circuit is locked in a permanent powered state if signals enter from both sides simultaneously.



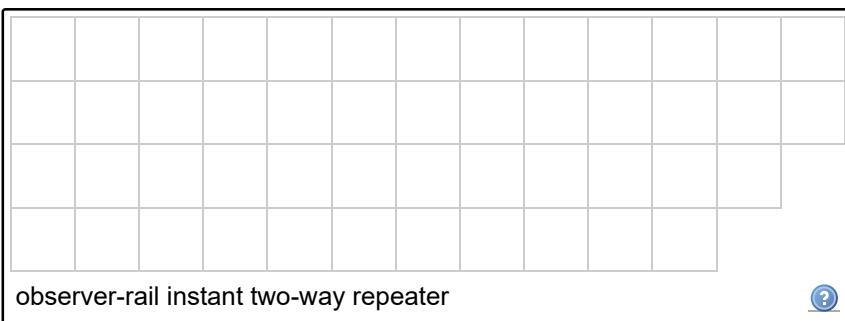
Locked-repeater two-way repeater – [schematic]

Variation (offset input): The circuit shown in the schematic to the right keeps the transmission lines in-line with each other, but reduces the signal strength by 1 in side movement in both input and output before continuing the transmission, so the circuits must be placed with only 11 dust between them to work. Placing a block behind each of the input repeaters and moving the input/output lines closer to the repeaters' outputs means that signal strength is lost only in side movement at input, allowing an additional dust between the circuits (and thus a more efficient transmission), but requires that the transmission lines alternate which side they run on.

Earliest known publication: December 21, 2012^[7]

observer-rail instant two-way repeater

1×12×4 (48 block volume)
1-wide, silent, instant transmission speed:
 instant
circuit delay: 0 ticks
fastest clock signal: 2-clock
two-way reset time: 2-clock



Behavior: The powered rails

initially hold powered. When any input is given, the rails act as a BUD and depowers, spreading the pulse both ways. All of this happens in the same game tick. Inputs can be added by creating a block update anywhere in the chain, and outputs added by adding an observer anywhere.

Variations: Can be coupled to a line of droppers for an instant item line (Inspector Talon, 2022).

Earliest known publication: Aug 26, 2021^[8]

Transmission encoding

For simple redstone structures, digital ("on/off") transmission is sufficient.

For complex redstone structures, with banks of inputs or outputs, more sophisticated forms of transmission may be required, such as analog, binary, or unary transmission.

When numbers are represented by different types of transmission, they are said to be **encoded**.

Analog

An **analog** encoding in *Minecraft* (a.k.a. **hexadecimal wire** or simply **hex wire**) is a transmission that outputs the same signal strength it receives as input. Because power levels can vary from 0 to 15, an analog transmission can convey 16 states in a single wire.

Analog vs. Digital in Real-Life

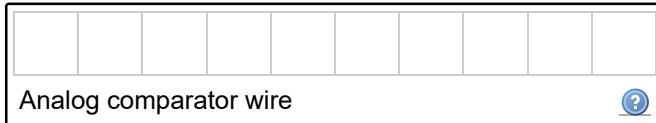
"Analog" means "continuously variable". This doesn't match *Minecraft* analog wires that have 16 discrete values (for example, a signal strength of 6.89 can't be in *Minecraft*). But a term was needed to differentiate between signal strength transmissions and on/off transmissions, and the real-life distinction between digital electronics (which generally transmit either a high voltage or low voltage) and analog electronics (which operate on continuously varying voltage levels) was a close fit, so the two terms were adapted for use by the *Minecraft* community.

Analog comparator wire (ACW)

flat, silent
circuit delay: 1 tick per 4 blocks

The simplest analog wire is a line of redstone comparators. Like redstone repeaters,

comparators can draw a signal from one opaque block and push it into another; thus, it is usually more efficient in resources and in signal delay to place comparators every four blocks. This wire is best used for short distances and tricky turns.


[?](#)

The signal strength of an analog comparator wire (ACW) can be reduced or suppressed at some point along its length by feeding another signal into one of the comparators in subtraction mode. The signal can be overridden by feeding a stronger signal into one of the opaque blocks.

Because the redstone dust is not adjacent to any power or transmission components, only opaque blocks, it does not configure itself to point in any particular direction. This also causes the dust to power any opaque blocks or mechanism components to the side of the analog wire. Transmission components should not be placed adjacent to the wire's dust because that would cause the dust to configure itself in a way where it doesn't power the rest of the analog wire.

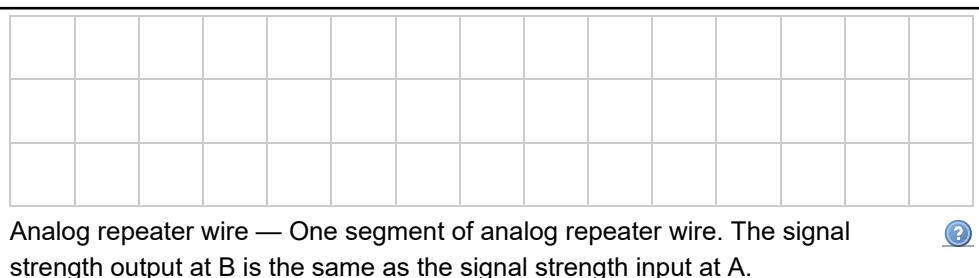
Earliest known publication: January 9, 2013^[9]

Analog repeater wire

*flat, silent
circuit delay: 1 tick
per 14 blocks*

Signal strength can also be retained by using repeaters to repeat every possible signal

strength at the correct distance from the output to convey the correct signal strength. It is the fastest option for long distances.


[?](#)

A single segment of analog repeater wire (ARW) consists of exactly 15 repeaters connecting an input line to an output line. To connect multiple segments together without additional comparators, the segments must be arranged so that the output dust of the last repeater is the same as the input dust of the next segment (i.e., block B of the previous segment is block A of the next segment). This causes the segments to overlap in distance by one block and causes each segment to be offset to the side from the previous segment by two blocks.

Variations: To keep the segments in-line, or to turn against the direction the repeaters are facing, raise the final repeater by one block and drop the next segment underneath it.

Another option is to use a comparator and an opaque block between the segments, and alternate the direction the repeaters are facing. This keeps the height to 2 blocks but increases the circuit delay to 2 ticks per 17 block.

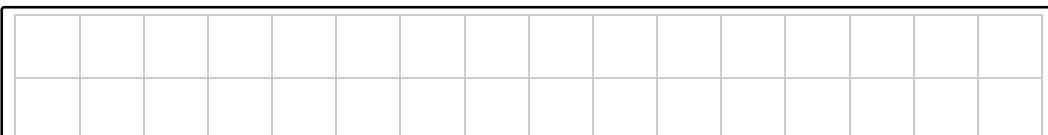
Earliest known publication: November 21, 2012^[10]

Analog subtraction wire

*flat, silent
circuit delay:
1 tick per (18-
N) blocks
(see below
for N)*

If fewer than 15 states need to be transmitted

(for example, output from a picture frame, composter or cauldron), it may be more efficient to encode those N states in the higher levels of signal strength, and then repeatedly subtract the transmitted value from 15 after (17-N) dust, an even number of times. However, it is complicated and infrequently useful.

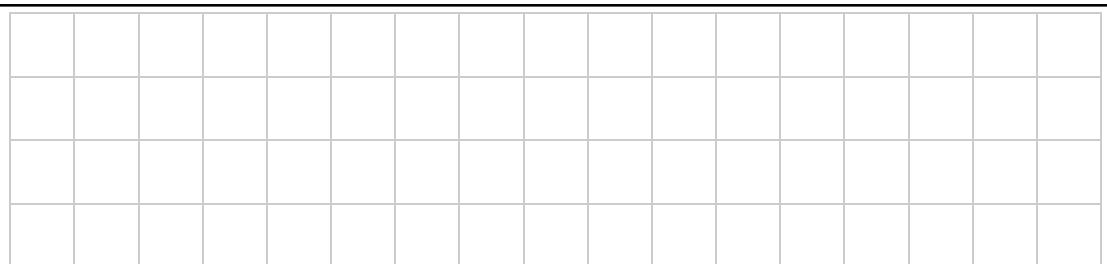


Analog Subtraction Wire — Shown: 10-state ASW (aka "decimal wire"). For a 10-state wire, the input signal strength must be between 6 and 15 (so should never actually be off). Chests are completely full with any items. The signal strength output at B is the same as the signal strength input at A.



Tap-anywhere analog comparator wire

*flat,
silent
circuit
delay: 1
tick per 2
blocks +
2 ticks
always*



Tap-anywhere analog comparator wire — A fragment of the line. The signal strength at every output is the same as the highest signal strength at any input.



The tap-anywhere ACW is helpful with 1-tileable devices, both all requiring the same input, or collecting output from them (e.g. readout from a memory bank), sustaining the strongest of the inputs all the way "downstream" from it, on all the outputs.

Vertical analog transmission

The vertical options for analog transmission are similar to the horizontal options.

Vertical ACW

*silent
circuit delay: 1 tick per 1 vertical block (up), 1 tick per 2 vertical blocks (down)*

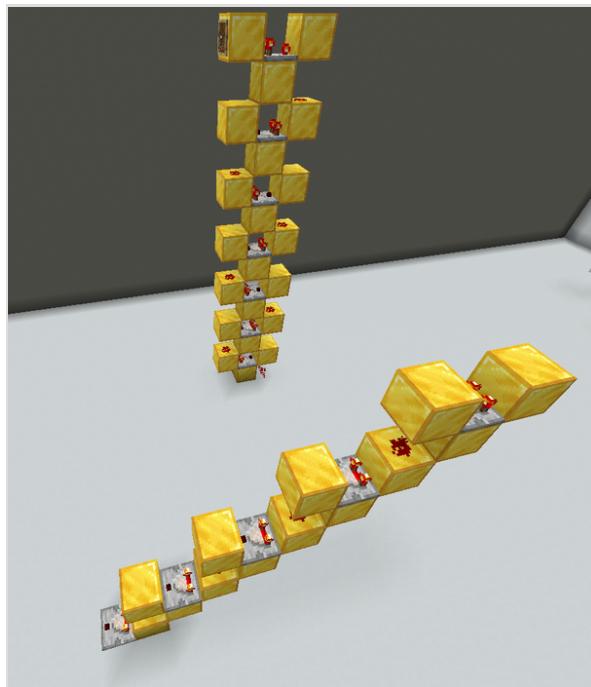
A redstone comparator can power a block with dust on it, and that dust can power another

comparator at its level, etc. Vertical ACW travels two blocks sideways for every 1 block moved upward (or three blocks with an additional block between the dust and the comparator), but can also be bent at each block into a 3×3 "circular staircase".

The downward variant can go two blocks down for two block sideways, and double back, for a 1×3 tower.



Vertical ACW



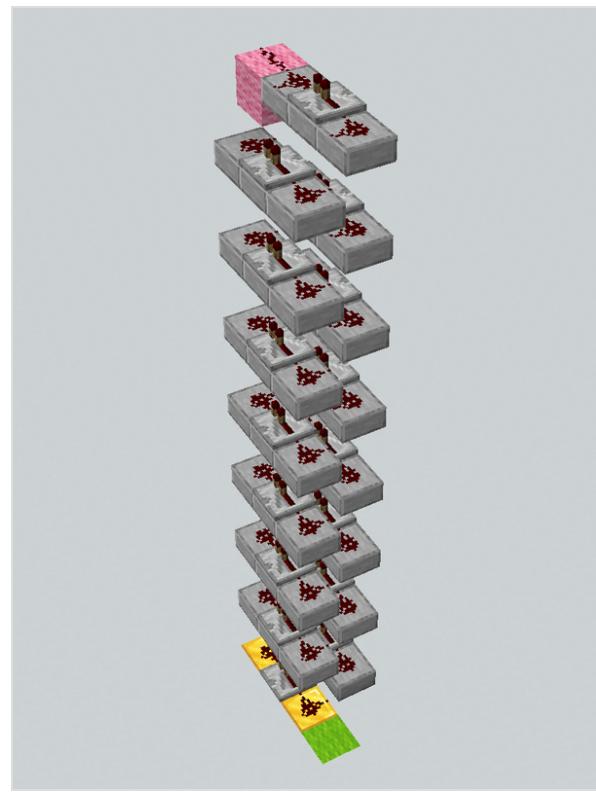
Vertical comparator wire diagonally and straight downward.

Vertical ARW

*silent
circuit delay: 1 tick per 14 vertical blocks*

Vertical ARW is an analog repeater wire built on redstone ladders. It transmits signals only upward and in segments of 14 vertical blocks (use vertical ACW to close any gaps). Like horizontal ARW, the last dust of the previous segment must be the first dust of the next segment unless a short run of vertical ACW is used to connect the two segments.

Horizontal ARW built on a 3-wide staircase can be used to transmit analog redstone signal diagonally downward.

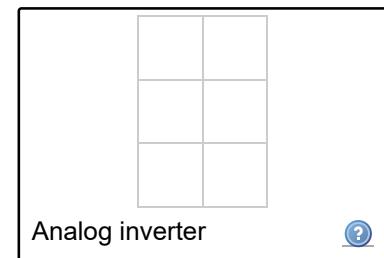


Vertical ARW

Vertical ASW basically just consists of redstone staircases or ladders with occasional breaks for subtraction.

Analog inverter

An analog inverter is a circuit that inverts the signal strength, for example a signal with the strength of 6 becomes 9, and a full strength signal becomes 0. This can be achieved by placing a redstone block at the back of a redstone comparator and the input signal at the side of the comparator, with the output signal at front of the comparator, or by making an analog repeater wire, but instead of the wire in front of the repeater, place a block with a redstone torch attached to the front. place a wire in front of every torch, and the bottom-left redstone dust is the output.



Binary

A **binary** encoding consists of multiple digital lines run in parallel, with each line representing a different digit in a single binary number. For example, three lines might individually represent binary 001 (decimal 1), binary 010 (decimal 2), and binary 100 (decimal 4), allowing them together to represent any value from decimal 0 to 7 (by summing the represented values of the powered lines). An individual digital line of a binary transmission is referred to by the value it can add to the total number (for example, the 1-line, the 2-line, the 4-line, the 8-line, the 16-line, etc.)

When a binary encoding is intended to carry only decimal numbers (encoding only values 0 to 9), it

is known as binary-coded decimal (BCD).

4-bit binary encoding

A 4-bit binary encoding contains the same amount of information as an analog line, i.e. 16 states. Compared to the analog version, binary lines are easier to transmit since no concerns about signal strength is needed. They can also be handled with logic circuits.

Converting between digital and analog, i.e. building a digital-analog converter (DAC), can be complicated.

8-bit and 16-bit

8-bit (a.k.a. "byte bus") and 16-bit binary encodings are also used in the construction of computer recreations.

One-hot

A **one-hot** encoding consists of multiple digital lines run in parallel, where a value is represented by which line is on (for example, the number 5 might be represented by having only the fifth line on, and all other lines off). A common alternative to one-hot encoding is **one-cold** encoding, where a number is represented by which line is off instead of which line is on.

While a strict one-hot encoding would always have precisely one line hot, a useful alternative is to eliminate the "0" line and represent the number 0 with no lines hot.

One-hot and one-cold encoding are rarely used for transmitting values over distance, but may be used for inputs (e.g., which button is pressed) or outputs (e.g., which dispenser is activated), with conversion to or from a more efficient transmission encoding in between. They are also used as a transitional encoding — for example, it is relatively complicated to decode from analog to binary directly, but relatively simple to first decode from analog to one-cold, then encode from one-cold to binary.

Unary

A **unary** encoding consists of multiple digital lines run in parallel, where a value is represented by the number of lines on (for example, the number 5 might be represented by having 5 of 16 lines powered). A unary encoding might use a convention that lines must be powered starting from one side (allowing values to be determined from the transition from powered lines to unpowered lines) or that they can be powered in any combination (requiring logic circuits or calculations to determine the represented value).

Unary encoding is rarely used for transmitting values over distance, but may be used for inputs (e.g., the number of players standing on pressure plates) or outputs (e.g., the number of doors opened along a passageway).

Unary in Real-Life

"Unary" means "having only one element". Among other uses, the term is used for the unary numeral system, a way of representing numbers using only one digit, and for unary coding, a way of representing numbers using two digits but the second digit is used only to terminate numerals. In *Minecraft*, "unary" doesn't match either of these real-life uses exactly, but the term is still widespread.

Wireless transmission

! Note: This section contains circuits built from [command blocks](#), which cannot be obtained legitimately in [Survival](#) mode. These circuits are intended for server ops and adventure map builds.

[Command blocks](#) allow redstone signals to be transmitted to any loaded chunk, without a direct connection.

Setblock transmission

Setblock transmission works by using the [setblock command](#) to create and remove [power components](#) at a receiver.

For the setblock transmitters below, the two command blocks should be given commands to create and remove power components at the receiver location. For example, if X, Y, and Z are the absolute or relative locations of the setblock receiver:

- "on": `setblock X Y Z redstone_block`
- "off": `setblock X Y Z stone`

Other power components can be used to activate the receiver, but most require additional data to specify their orientation (for example, to specify the direction a lever is attached). Additionally, any non-power component can be used to deactivate the receiver ([even air](#)). Avoid using blocks that are transparent or produce light (like redstone torches) as that can increase lag due to block light calculations in up to hundreds of blocks around the receiver.

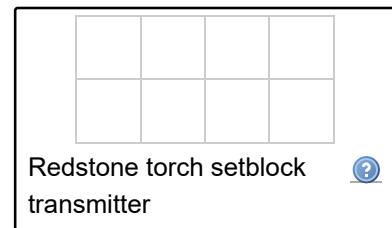
When a setblock command is executed, the affected block does not change until a half a redstone tick later (one game tick). Thus, the minimum circuit delay for setblock transmission is 0.5 ticks.

Setblock transmitter, redstone Torch

1×3×1 (3 block volume)
 1-high, 1-wide, flat, silent
circuit delay: 0.5 ticks (rising edge) and 1.5 ticks (falling edge)
fastest clock: 3-clock

Because of the difference in rising and falling edge delays, on-pulses are lengthened by 1 tick and off-pulses are shortened by 1 tick.

Variations: Many other arrangements of the torch and command blocks are possible.



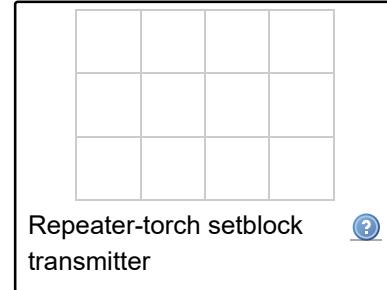
Setblock transmitter, repeater-torch

1×3×3 (9 block volume)
 1-wide, silent, tileable
circuit delay: 1.5 ticks
fastest clock: 3-clock

Unlike the redstone torch setblock transmitter, this transmitter doesn't change the lengths of input pulses.

Variations: The torch can be moved to the side of the input block, and the command blocks

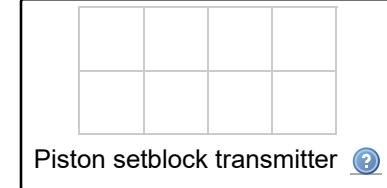
moved to the side of the repeater and the block it's facing, to make the circuit 2-wide but flat.



Setblock transmitter, piston

$1 \times 3 \times 2$ (6 block volume)
1-wide, tileable
circuit delay: 2 ticks
fastest clock: 2-clock

Noisy, but small and can run on a faster input clock.

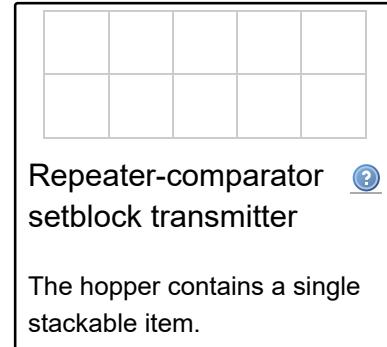


Variations: The command blocks can also be moved above or to the side of the block of redstone's positions. The piston can also be pointed downward (but not upward), with the command blocks alongside the block of redstone's positions.

Setblock transmitter, repeater-comparator

$2 \times 4 \times 2$ (16 block volume)
flat, silent
circuit delay: 1.5 ticks
fastest clock: 2-clock

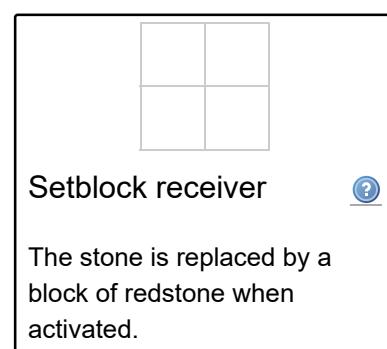
Larger, but can handle a faster input clock without noise.



Setblock receiver

$1 \times 1 \times 1$ (1 block volume)
1-high, 1-wide, flat, silent, tileable

A setblock receiver is simply a single block of space for a transmitter to create or remove a power component.



Scoreboard transmission

Scoreboard transmission works by setting values for scoreboard objectives. Scoreboard transmission can be used to transmit simple binary values (as shown below), but scoreboard objectives can store values between -2,147,483,648 and 2,147,483,647 (inclusive) and multiple scoreboard objectives can be active at once (though transmitting and receiving many values requires arrays of transmitters and receivers). A single scoreboard transmitter can activate multiple receivers at once and different transmitters can set the scoreboard objective to different values, activating specific sets of receivers and simultaneously deactivating all other receivers. Scoreboard receivers can also respond to *ranges* of values, instead of just specific values.

Scoreboard transmission requires the creation of a dummy scoreboard objective to store the transmission's current value ("WirelessBus01" is an example and can be anything):

```
/scoreboard objectives add WirelessBus01 dummy
```

Binary scoreboard transmitters are similar to setblock transmitters, except they require different commands (WirelessBusFakePlayer is an example and can be anything, but only one fake player is required for all wireless buses):

- "on": scoreboard players set WirelessBusFakePlayer WirelessBus01 1
- "off": scoreboard players set WirelessBusFakePlayer WirelessBus01 0

Unlike setblock receivers, scoreboard receivers must be run on clock circuits.

Scoreboard receiver, setblock clock

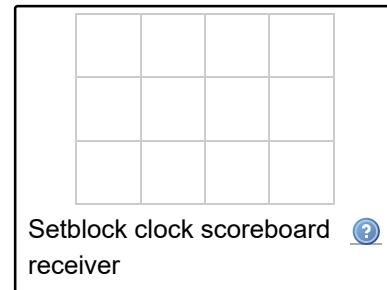
1×3×3 (9 block volume)

1-wide, silent

circuit delay: 1 tick

The scoreboard receiver uses a fast clock to test the objective's value. The command blocks should have the following commands:

- R: setblock ~ ~-1 ~ redstone_block
- S: setblock ~ ~1 ~ stone
- W: scoreboard players test WirelessBusFakePlayer WirelessBus01 1 1



Variations: The setblock clock can be replaced with any other fast clock.

Summon transmission

Summon transmission works by summoning an item onto a wooden pressure plate.

Unlike setblock and scoreboard transmission, summon transmission doesn't require an "off" command block, depending only on the summoned item's despawn time to deactivate the receiver.

References

1. "BeGamerPlays" – "Dual-Edge InstaWire 1.5" (<https://youtube.com/watch?v=JRGzIWQMxLY>)
2. "TT Lemon" – "Snapshot 13w02a - Instant Repeater" (<https://youtube.com/watch?v=IBijefXB-bg>)
3. "DvirWi" – "Two way repeater" (<https://youtube.com/watch?v=nsX39FxaXIw>)
4. "CodeCrafted" – "Minecraft Challenge: 2-Way Repeater (Compact design)" (<https://youtube.com/watch?v=fIERTO-52J5c>)
5. "DvirWi" (February 18, 2013) – "Instant two-way repeater (Designed for 1.5)" (<https://youtube.com/watch?v=5HgiE-B9dOQ>)
6. "RedstoneInnovation" (September 8, 2013) – "Simple & Compact 2-Way Repeater! [Tutorial]" (<https://youtube.com/watch?v=fCEtjrCJdDs>)
7. "rapamaro" – "the most compact 2 way repeater (1.4.7)" (<https://youtube.com/watch?v=zoY330YXT6E>)
8. "Kahyzen" - "Robust Instant Wire" (<https://youtube.com/watch?v=nJx-o9fDVm4>)
9. "seiterarch" (January 9, 2013) ""Minecraft Beyond Binary 01: All the Comparisons (13w01b)"" (<https://youtube.com/watch?v=bC11DjfNjm8>)
10. "Cubehamster" (November 21, 2012) "Minecraft: Redcoder (Decoding Redstone Dust)" (<https://youtube.com/watch?v=ct2zp1RRRcU>)

Navigation

◆ Redstone [hide]

Redstone circuits & tutorials [hide]

Redstone basics  [Redstone components](#)  [Redstone mechanics](#)  [Conductivity](#)

Redstone circuits [Clock circuits](#) [Logic circuits](#) [Memory circuits](#) [Miscellaneous circuits](#)

Pulse circuits [Transmission circuits](#)

Featured tutorials  [Advanced redstone circuits](#)  [Block update detector \(BUD\)](#)

 [Comparator update detector \(CUD\)](#)  [Flying machine](#)  [Mechanisms](#)

 [Piston circuits](#)  [Quasi-connectivity](#)  [Redstone music](#)

Redstone components [show]

Retrieved from "https://minecraft.wiki/w/Redstone_circuits/Transmission?oldid=3333444"

This page was last edited on 24 December 2025, at 22:21.

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.