

Redstone circuits/Piston

< Redstone circuits

Pistons have allowed players to design circuits that are smaller and/or faster than the standard, redstone-only counterparts. An understanding of standard [redstone circuits](#) is helpful, as this tutorial is focused on the circuit design rather than the function.

There are three benefits of piston circuitry:

- Neither repeaters nor pistons 'burn out', unlike redstone torches.
- Piston circuits tend to be smaller than their redstone counterparts.
- The resources used are much easier to find and mine.

The main components of piston circuits are [sticky pistons](#), [redstone wire](#), [repeaters](#), and [redstone torches](#).

Unless otherwise stated, all pistons are sticky pistons.

Contents

Principles

Piston gate designs

[NOT gate](#)

[OR gate](#)

[AND gate](#)

[IMPLIES gate](#)

[XNOR gate](#)

[XOR gate](#)

Latches

[RS latches](#)

[T flip-flops](#)

Ring memory

Clocks

[Rapidfire piston clock](#)

[Alternate rapidfire piston clock](#)

[Pulser](#)

[Rapid fire trap clock](#)

Edge detector

Double extender

Navigation

Principles

Power and Repeaters

Power is transmitted in several ways that are useful to pistons. The first thing to note is that there are two types of blocks; *transparent* and *solid*. Transparent blocks are things such as glass, slabs, or air, while solid blocks are more common materials such as dirt, stone, or wool.

The key is that redstone power can be transmitted through solid blocks, but not transparent blocks. However, power can pass through only one solid block at a go, it cannot be passed from one solid block to another. A solid block can power wire leading from it only if it was "strongly powered" (by a redstone torch, repeater, or *comparator*, but not by redstone wire). Redstone dust can be placed on some transparent blocks, but then it transmits power only upward, not downward (that is, not through the transparent block).

Redstone torches are considered logic components only if they change states as the gate is used. (Otherwise they're just power supplies.) If they change state too often, they are susceptible to burning out. If a solid block is on top of a redstone torch, any wire connected to the block becomes powered. If, however, the block is transparent, the torch does not power the wires. In Creative mode, using torches as power-supplies for blocks above is mostly deprecated by the introduction of redstone blocks, which are permanently powered. They have a higher cost in Survival, though.

When a repeater is directed at a solid block, it passes power into that block in the same way redstone torches do. A repeater can also take power from a solid block that is powered, even weakly powered (that is, it turns a weakly powered signal into a strong one). Power cannot be transmitted by transparent blocks.

Pistons extend if powered, but it's notable that they can take power from the block above them – that is, even if the block is air, if that block would have been powered, then the piston is powered. They can also be powered through their extended piston, which produces several useful quirks that are commonly used in circuits. A piston can push up to 12 blocks when it extends; however, a sticky piston can pull back only one block when it retracts. Some blocks are immune to being pushed or pulled, notably obsidian and any block with a GUI or inventory (except crafting tables). Other blocks drop as items if a piston tries to push them; this includes most "attached" blocks such as either sort of torch, or doors, but also a few others such as jack-o'-lanterns.

Piston gate designs

Main article: Logic circuit

Pistons can produce alternate designs for many of the classic logic gates. Piston gates may also use redstone torches to supply a constant signal (in which case they can often be replaced by

levers), or for other purposes.

NOT gate

The piston-only NOT gate, also called an inverter, is slightly larger than a standard redstone NOT gate, so it is rarely used alone. However, it demonstrates an important concept used in many piston mechanisms, namely that torches below solid blocks create a current in any surrounding wire. When an input is triggered, the piston extends, uncovering the torch hole removing the signal from the output line. The input can power the piston from almost any direction. Alternatively, a block of redstone can be used in to shrink the design.

Piston NOT gate

OR gate

This design is slightly faster than the standard redstone OR gate, with a comparable size. It uses one piston that covers a torch when extended by any of the inputs.

Piston OR gate

AND gate

Incredibly fast AND gate. When unpowered, the sticky piston pulls the block over the hole, breaking the circuit. When powered, it extends and lets the current flow in and out of the hole. One input feeds the piston, the other feeds the circuit it interrupts; both must be ON to release a signal.

Piston AND gate

IMPLIES gate

Comparable size and speed to some of the basic redstone IMPLIES gate. This gate represents material implication. Returns false only if the implication $A \rightarrow B$ is false. That is, if the conditional A is true, but the consequent B is false. It is often read "if A then B." It is the logical equivalent of "B or NOT A". The torch under the mobile block is in a hole.

Piston IMPLIES gate

XNOR gate

Pronounced "ex-nor", this device activates when both inputs are equal. A useful attribute is that a XOR or XNOR gate always changes its output when one of its inputs changes, allowing for 2 switches to be combined to open or close a door, or activate another device. This design is significantly smaller than the redstone-only equivalent and slightly faster. Adding a NOT gate to the end or to either input produces an XOR gate, which activates when the inputs are different.

Piston XNOR gate

XOR gate

Pronounced "ex-or" (a shortening of "exclusive or"), this device activates when only one of the inputs is on. This design, like the XNOR gate, is smaller and faster than the redstone-only equivalent.

Piston XOR gate

Latches

Main article: [Memory circuit](#)

Latches are [memory circuits](#). Naturally, pistons' ability to physically move blocks to new locations make them a natural tool for these.

RS latches

The basic piston RS Latch is small and easy to make. The pistons used here are regular pistons, not sticky pistons, and push a block over one of two holes containing a Redstone Torch. One hole can be omitted if only one output is required. Using a [Block of Redstone](#) lets the circuit become even smaller, and adds the inverse output.

Piston RS Latches

T flip-flops

These T flip-flops use one input to switch between two states.

Sticky pistons behave strangely when they receive a 1-tick signal. If a block is directly next to the piston, the piston pushes the block but does not pull it back when the signal ends. If the piston receives another 1-tick signal, the piston extends and retract the block. This can be used to toggle the position of blocks.

Design A, 4×2×4. Uses regular pistons. Both of the pistons are regular pistons. This flip flop is quite fast and quite small. It toggles when the input changes from ON to OFF. Inverting the input increases the speed of the circuit.

Piston Toggle A

Design B (5×3×2) is actually an RST latch, combining the functionality of both the Set/Reset (RS) and toggle (T) latches. Uses regular pistons. This flip flop doesn't use torches for logic so it can work with signals of any length. The dust on level 1 is there to redirect the redstone wire away from block X, which powers the piston. However, this circuit does not have an inverse output.

Piston Toggle B

Design C is a combination pulse limiter and downward edge detector. When the signal turns off, the first sticky piston retracts the second, which receives a 1-tick signal. The 1-tick signal toggles the mobile block. It is sensitive to timing.

Piston Toggle C

The 1-wide Sticky Piston TFF design is 5×1×3. It depends on the fact that a sticky piston leaves the block after extending when given a short pulse of 0.5 ticks of delay. A [circuit breaker](#)

1-wide Sticky Piston TFF

is used to give a 0.5 ticks pulse to the sticky piston. This makes the sticky piston leave the redstone block, which then provides power to the output. When powered again, the sticky piston pulls the redstone block switching the output off. It is possible to make this TFF tileable.

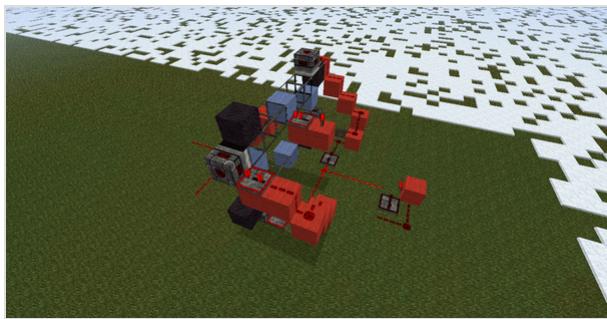
Ring memory



This section needs schematic(s) or diagram(s) instead of YouTube video links.

Please remove this notice once you've added suitable diagram(s) to the article.

This is a ring of blocks attached to regular pistons at the corners so it can rotate. The blocks are usually a combination of solid and non-solid blocks. The pistons are often connected to a clock so that they rotate the ring. Most (if not all) rings have a reading head consisting of a repeater pointing at the ring and a redstone torch powering the repeater. By using redstone on the other side of a ring, one can see which type of block is in front of the reading head (1 = Solid; 0 = Non-solid). This information now can be passed to a circuit. By using a comparator and reading a cauldron's fill level, 3 additional memory cell states are available. For repeater-based reading heads those act as zeroes.



A simple ring of blocks, rotated by pistons. The reading head is on the right side of the ring. The rightmost circuit is a clock that drives the pistons

Adding multiple rings together in a row creates a band. A band stores even more information and works similar to punched tape. Examples include [music machines](http://www.youtube.com/watch?v=qt1fOED0vC8) (<http://www.youtube.com/watch?v=qt1fOED0vC8>), combination locks, and [memory](http://www.youtube.com/watch?v=0t5hD-C1JBg) (<http://www.youtube.com/watch?v=0t5hD-C1JBg>).

Clocks

Main article: [Mechanics/Redstone/Clock circuit](#)

Rapidfire piston clock

This rapidfire clock is fairly simple to build. It requires 2 non-sticky pistons, 2 repeaters, and 6 redstone, and 5 solid blocks. The repeaters must be set to at least 2 ticks, and they must match.

Rapidfire Piston Clock

Place the mobile block last to start the clock, and take output from either end or any wire. Note that this clock is *not* switchable, and if it does stop for any reason, it's difficult to restart by a redstone signal. The player can restart it by breaking and replacing the mobile block, or by changing the repeater delays (both of them; the clock restarts when they match and are set to at

least 2 ticks).

Alternate rapidfire piston clock

This piston clock is a simple way of repeating a circuit that goes at high speed without burning out. It does not require repeaters or redstone torches. A tutorial is available here. (<http://www.youtube.com/watch?v=4QMtn6ojZrQ>) It requires a single sticky piston, one lever, a small amount of redstone, and some solid blocks. From the lever, the redstone has to be placed so it goes up a block, before linking up to the piston. The piston has to be placed in a way that, when it is extended, it blocks the redstone from going up the side of the block without breaking the redstone. This breaks the circuit, removing power from the piston and thus retracting it, causing the circuit to open again. The signals are too fast to affect lighting sources such as lamps and redstone torches. Doors and dispensers both work well at this speed.

Alternate Rapidfire Piston Clock

Pulser

A small, stable pulser in a space of 2×3×2. The period can be adjusted with the repeater, and the clock can be turned off or on with the lever. (Turning the lever ON stops the clock, and *vice versa*.) The lever can be placed on any of the solid blocks.

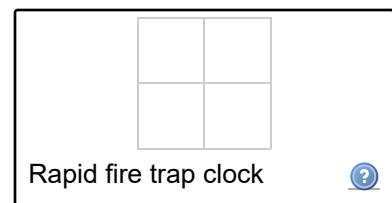
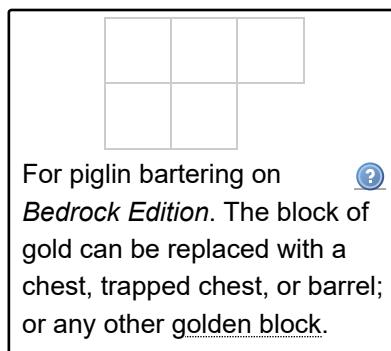
Piston Pulser

Rapid fire trap clock

Another way of doing this is placing a sticky piston with a redstone block on its face and a line of redstone dust connecting the redstone block and the sticky piston. It must be two long to prevent lockup. To make it toggleable, connect the dust with a lever. This is useful for traps. It is also useful for piglin bartering on *Bedrock Edition* due to MCPE-91275 (<https://bugs.mojang.com/browse/MCPE-91275>), however it needs to be modified to allow for a gold block to be pushed and pulled back and forth.

Edge detector

A rising edge detector creates a brief signal when the input turns on. Conversely, a falling edge detector creates a brief signal when the input is turned off, and a dual-edge (aka "zero-crossing") detector responds to both.



Design A can be a rising- or falling-edge detector depending on the delays of the repeaters.

Switchable Edge Detector (A)

- For a rising edge, set both repeaters to 1-tick.
- For a falling edge, set the left repeater to four-ticks and the right repeater to one-tick. This creates a 2-tick signal.

Design B is a variant of the piston XOR gate, and a dual-edge detector. The right-hand repeater can be adjusted to output a longer or shorter pulse.

Double-Edge Detector (B)

Designs D, E, and F are straight-line versions, $1 \times 2 \times 2$. Design D is a rising edge detector; design F, a falling edge detector; both with an output pulse of 2 ticks. Design E is a zero crossing detector, activating on both rising and falling edges. However, it produces only a single-tick pulse output. This can be moderated by adding a 2-tick repeater to the output, producing a 2-tick pulse, same as the other two.

Inline Edge Detectors

Double extender

This design pushes and pull a block two spaces instead of one. The first and second repeaters must be set to delays of 2 and 4 respectively. The pistons are sticky and the device correctly pushes and retracts the block. The main trick is properly sequencing the retraction because the back piston does not pull back the forward piston when it is extended. Additionally, the back piston retract only the forward piston, not the block. To handle these issues the forward piston must be retracted, pulled back, then extended and retracted again.

Basic Double Extender

Triple extenders and so forth are possible, but involve more complex circuitry. As more pistons are added to the circuit, the logic becomes more complex. In an ideal circuit the extension and retraction of the pistons are governed by the number of pistons, denoted P. Then, the time to extend the pistons equals P ticks. The time to retract the pistons equals $3P-2$ ticks. This relationship is linear, and implies that the time for the circuit to execute goes up proportionately with the number of pistons. This relationship is achieved by moving multiple pistons per tick rather than only one at time. This relationship applies to both horizontal and vertical extenders.

Due to the pistons limit to pushing 12 blocks at a time, an 11 piston extender is the largest possible extender. Assuming the design is constructed ideally, this needs 11 ticks to extend and 31 ticks to retract. A player built design would likely use more delays for ease of construction.

A more advanced (and much larger) circuit can double-push pistons, without extending them (and stopping the system). This circuit can also be powered with torches from below, to hide all the wiring. Either way, input to any of the green-marked squares, so the signal reaches all the way round without needing extra repeaters. The design can be mirrored to push a pair of pistons on the same switch.

Piston-Safe Double Extender

Vertical double extenders are more difficult to make than horizontal extenders; the release of observers in 1.11, however, has greatly simplified the design of

Vertical Double Extender (side view)

vertical double extenders. Design A is a 3×3 tileable design that extends upward. Design B is a 4×3 tileable design that extends downward. The repeater in both designs are set to 1-tick.

Longer vertical extenders require complex circuitry, and are often used as elevators. To slightly simplify the required circuits, a gravity-affected block like gravel or sand can be used as the elevator platform. This avoids the need for the top piston to be sticky and for it to execute multiple extensions to pull down the top block at each stage of descent. If more than two pistons are used in total, multiple extensions of lower sticky pistons are still required to pull down the pistons higher in the stack, which are not gravity-affected.

Navigation

◆ Redstone			
Redstone circuits & tutorials			
Redstone basics	Redstone components	Redstone mechanics	Conductivity
Redstone circuits	Clock circuits	Logic circuits	Memory circuits
	Pulse circuits	Transmission circuits	Miscellaneous 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]		
	[hide]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]		
	[show]		
	[hide]</a		

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/Redstone_circuits/Piston?oldid=2751308"

This page was last edited on 15 November 2024, at 15:23.

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.