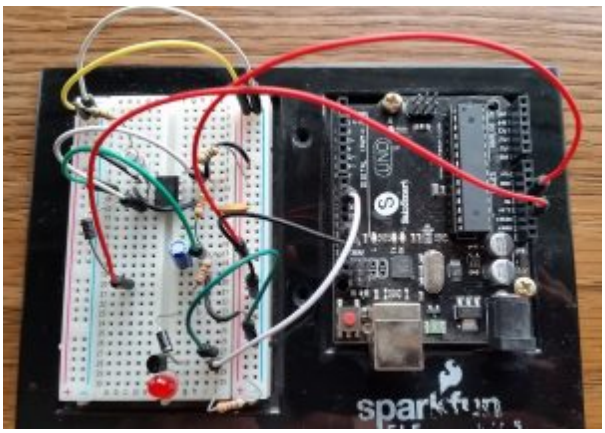
555 Watchdog Timer

Contents



SO WHAT'S THIS?

Sometimes you have a microcontroller or other embedded computer that stops responding for whatever reason, causing the device it is controlling to freeze up. It happens.

Sometimes just restarting it fixes it, barring a catastrophic hardware failure (like being run over by a bus).

Based on the classic 555 Timer chip, this watchdog circuit will restart it automatically.

It works like this:

1. The device being monitored (e.g. an Arduino microcontroller) sends a "heartbeat" signal to the watchdog at a regular interval, for instance, every 4 seconds. This tells the watchdog: "Hey, I'm still working!"
2. This heartbeat resets a timer on the watchdog, which is timed to send a "restart" signal to the device being watched, at an interval longer than the heartbeat, for instance, every

15 seconds. In other words, each time the watchdog receives a heartbeat, its own timer gets set back to zero and restarts counting.

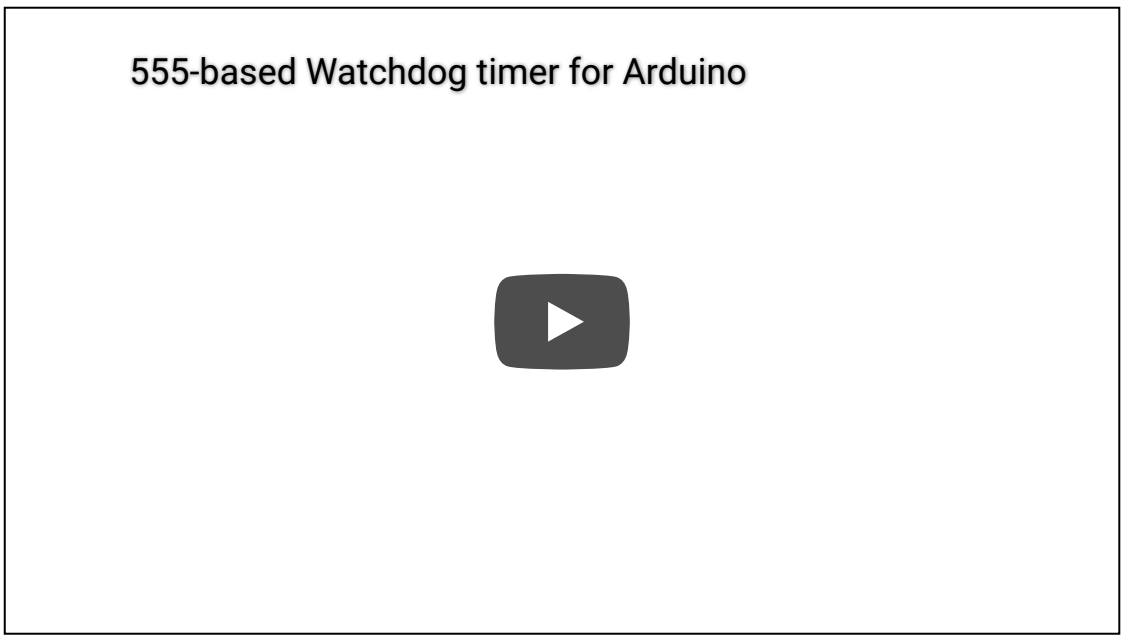
So as long as the watchdog keeps receiving the heartbeat before it sends the "restart" signal, it won't restart the watched device. If the watched device freezes up, it will stop sending a heartbeat and the watchdog will eventually send the "restart" signal to the device.

FEATURES

1. Based on a 555 timer chip configured in astable mode. These are very inexpensive.
2. Heartbeat indicator LED (LED1 blinks when heartbeat is sent from the watched device from pin 10 of the Arduino below)
3. Power-on indicator (LED2)

VIDEO DEMO

Here's a video demo of it in action:

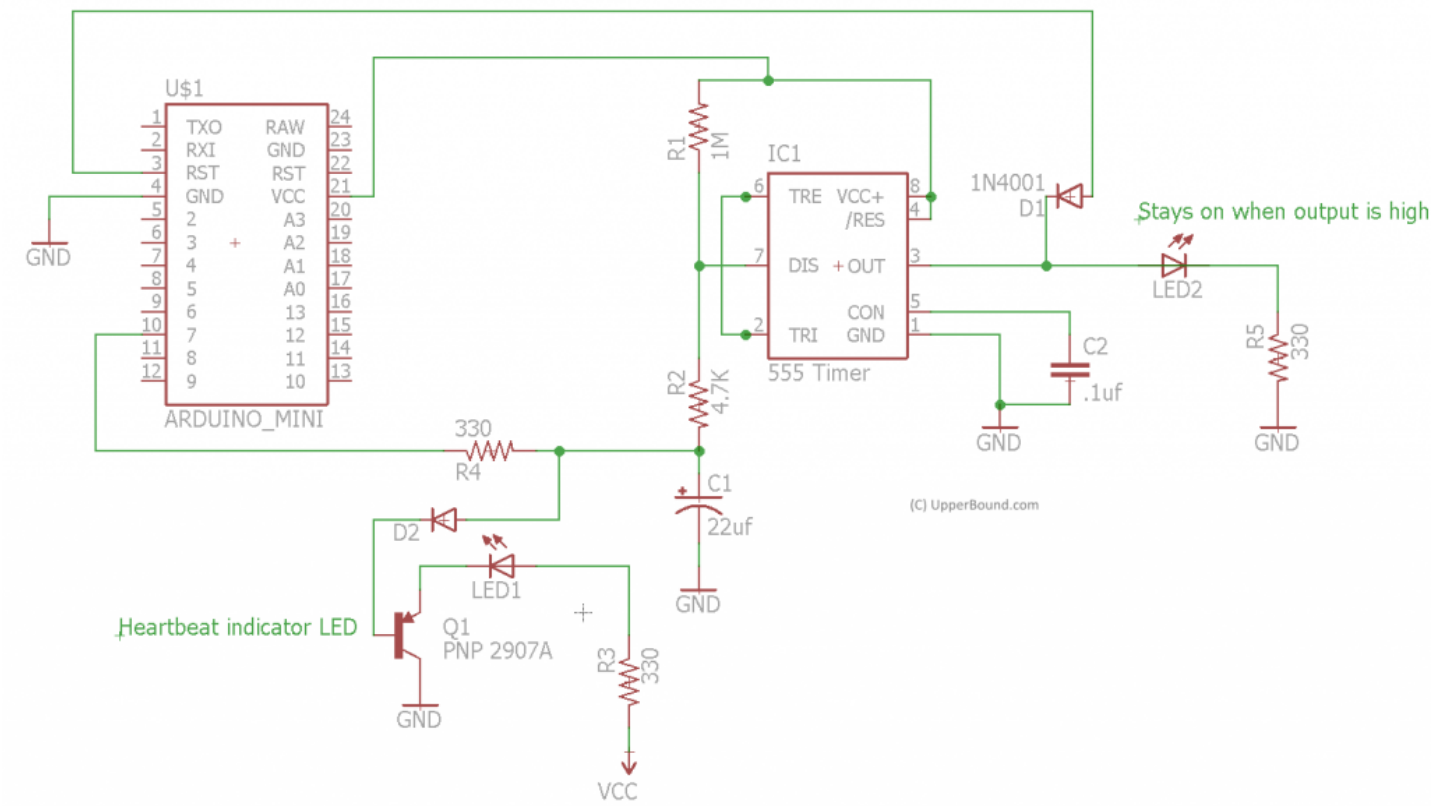


SCHEMATIC

The values for R1, R2 and C1 configure the 555 timer to send the RST pin of the Arduino low about every 15 seconds. This low pulse lasts for 71.66 ms, giving a duty cycle of 99.53%. The output of the 555 stays high until (the 15 seconds) sending it low.

An Arduino microcontroller is used to demonstrate usage.

Given this timing, the heartbeat sent by the Arduino should be at intervals less than 15 seconds.



555 Watchdog Timer Parts List

Part	Quantity	Purpose
Arduino Uno R3 Microcontroller or any other Arduino compatible microcontroller	1	The device being watched. Pin 10 above is sending the heartbeat.
555 Timer chip (IC1)	1	Does the main heavy lifting for the circuit by sending the actual reset signal to the Arduino's RST pin.
LED (any colors for LED1 and LED2)	2	Heartbeat and power indicators

Part	Quantity	Purpose
330 ohm resistor (R3,R4,R5)	3	R3 and R5 limits current for LED1 and LED2. R4 limits current into the base of transistor Q1. It's good to use current-limiting resistors for a transistor base.
PNP 2907A Transistor (Q1)	1	Switches current on/off to the heartbeat indicator LED
1N4001 Diodes (D1 and D2)	2	D1 prevents current from re-entering the 555 timer. D2 prevents capacitor C2 from discharging into the heartbeat indicator LED2. If you omit this diode, LED2 will fade out with each heartbeat instead of blinking. The blink vs fade-out is really a personal preference. You can omit the D2 without affecting watchdog performance.
1M resistor (R1)	1	Along with R2 and C1, controls the timing and duty cycle of the 555 timer, which determines the frequency and duration of the signal sent do the RST pin of the Arduino.
4.7K resistor (R2)	1	Along with C1, controls the "low" portion of the 555 timer's duty cycle.

Part	Quantity	Purpose
22uf capacitor (C1)	1	Along with R1 and R2, controls the timing and duty cycle of the 555 timer, which determines the frequency and duration of the signal sent do the RST pin of the Arduino.
0.1uf capacitor (C2)	1	Filters out potential noise into the 555 timer. Recommended per the 555 datasheet.

ARDUINO DEMO CODE

Code that demonstrates usage can be found on the github project.