Raspberry Pi Breadboard Wiring Guide

Smart Rain Harvest System

This guide shows how to connect the components to your Raspberry Pi using a breadboard.

Components Needed

1. **Raspberry Pi** (any model with GPIO pins)
2. **HC-SR04 Ultrasonic Distance Sensor**
3. **5V Solenoid Valve** (or relay module to control it)
4. **5V Relay Module** (to switch the solenoid valve)
5. **Breadboard**
6. **Jumper Wires** (Male-to-Female and Male-to-Male)
7. **Resistors** (1kΩ and 2kΩ for voltage divider)
8. **Power Supply** (5V for solenoid/relay if needed)

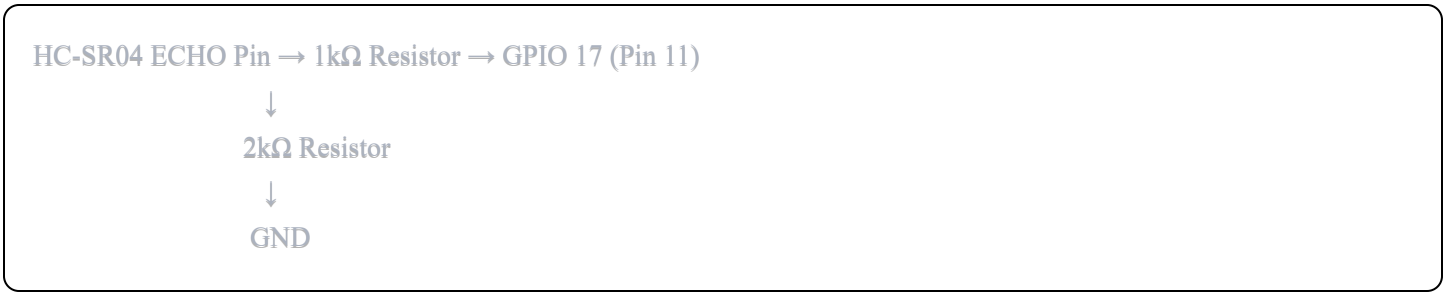
Pin Connections

1. HC-SR04 Ultrasonic Distance Sensor

The HC-SR04 operates at 5V, but its ECHO pin outputs 5V which can damage the Raspberry Pi's 3.3V GPIO pins. **You need a voltage divider for the ECHO pin.**

HC-SR04 Pin	Connection
VCC	Raspberry Pi **5V** (Pin 2 or 4)
TRIG	Raspberry Pi **GPIO 4** (Pin 7)
ECHO	**Through voltage divider** to Raspberry Pi **GPIO 17** (Pin 11)
GND	Raspberry Pi **GND** (Pin 6, 9, 14, 20, 25, 30, 34, or 39)

Voltage Divider for ECHO Pin:



This divides the 5V signal down to ~3.3V, safe for the Raspberry Pi.

---

2. Relay Module (for Solenoid Valve Control)

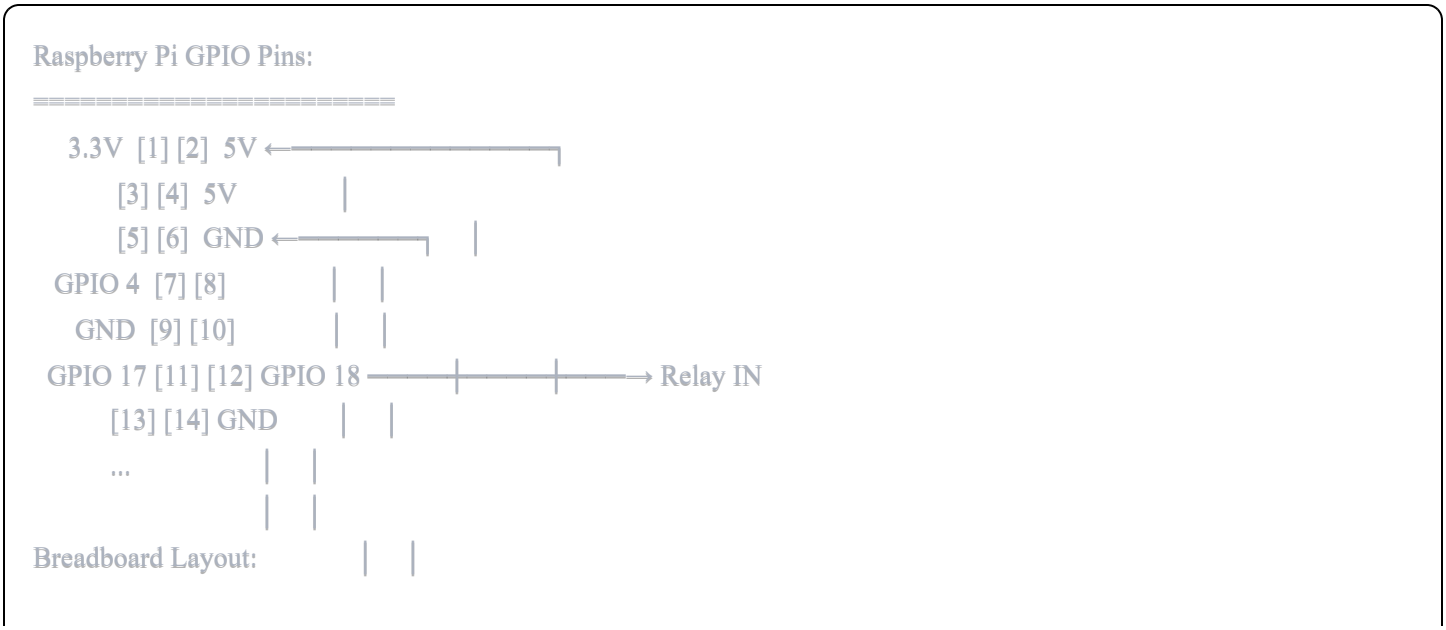
The relay module switches the solenoid valve on/off. Most relay modules can be powered by 5V and controlled by 3.3V GPIO.

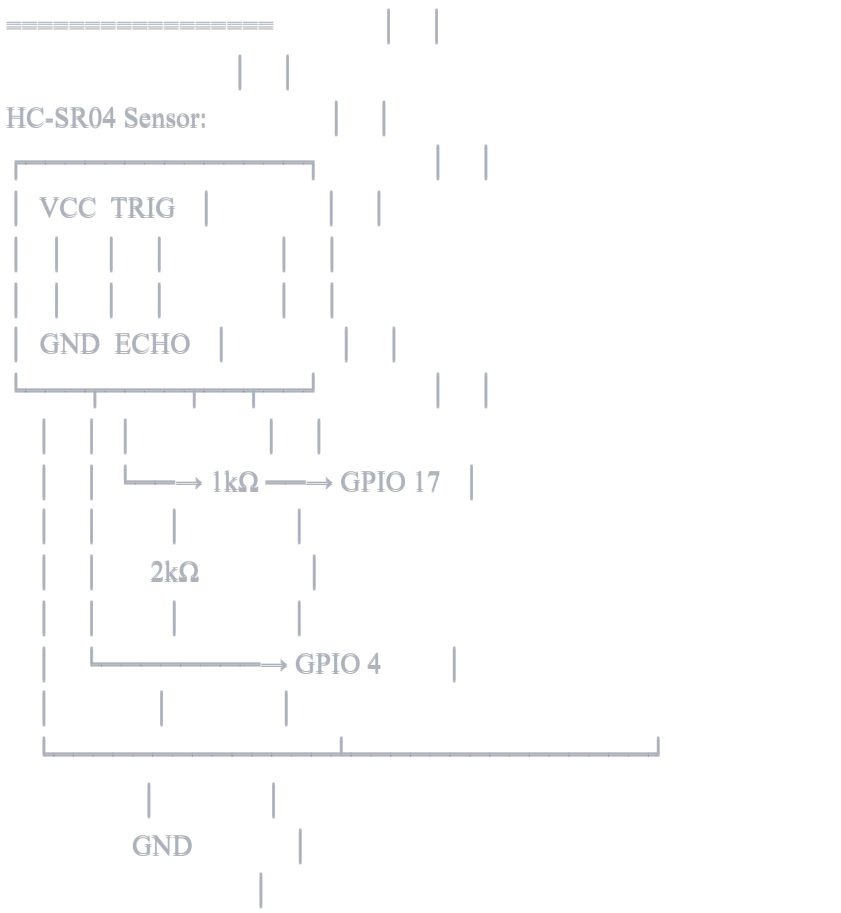
Relay Module Pin	Connection
VCC	Raspberry Pi **5V** (Pin 2 or 4)
GND	Raspberry Pi **GND** (Pin 6, 9, 14, 20, 25, 30, 34, or 39)
IN (Signal)	Raspberry Pi **GPIO 18** (Pin 12)

Solenoid Valve Connections:

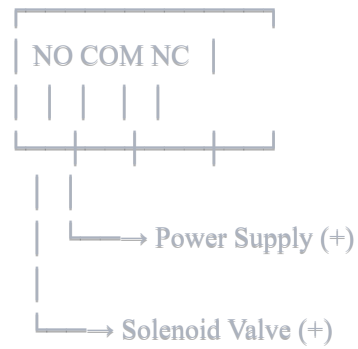
- Connect the solenoid valve to the relay's **NO** (Normally Open) and **COM** (Common) terminals
- The relay acts as a switch between your power source and the solenoid valve
- **Important**: Solenoid valves typically require external 12V or 24V power (check your valve's specs)

Breadboard Layout Diagram





Relay Switching (Solenoid Valve):



Solenoid Valve (-) ⟶ Power Supply (-)

Step-by-Step Wiring Instructions

Step 1: Power Rails

1. Connect Raspberry Pi **5V** (Pin 2) to the breadboard's positive (+) power rail
2. Connect Raspberry Pi **GND** (Pin 6) to the breadboard's negative (-) ground rail

Step 2: HC-SR04 Ultrasonic Sensor

1. Connect sensor **VCC** to breadboard **5V** rail
2. Connect sensor **GND** to breadboard **GND** rail
3. Connect sensor **TRIG** directly to **GPIO 4** (Pin 7)
4. **ECHO pin voltage divider**:
   - Connect sensor **ECHO** to one end of **1kΩ resistor**
   - Connect other end of 1kΩ resistor to **GPIO 17** (Pin 11)
   - Connect a **2kΩ resistor** from GPIO 17 junction to **GND** rail

Step 3: Relay Module

1. Connect relay module **VCC** to breadboard **5V** rail
2. Connect relay module **GND** to breadboard **GND** rail
3. Connect relay module **IN** (signal pin) to **GPIO 18** (Pin 12)

Step 4: Solenoid Valve

1. Connect your external power supply (12V/24V depending on valve specs) **negative** to relay **COM** terminal
2. Connect relay **NO** (Normally Open) terminal to solenoid valve **positive** wire
3. Connect solenoid valve **negative** wire to power supply **positive**
   - When relay activates, it completes the circuit and opens the valve

---

Important Safety Notes

⚠️ **WARNING:**

1. **Never connect 5V directly to Raspberry Pi GPIO pins** - always use voltage divider for ECHO
2. **Solenoid valves draw significant current** - use a relay, never connect directly to GPIO
3. **External power supply required** - Solenoid valves typically need 12V or 24V DC

4. **Double-check polarity** - Incorrect wiring can damage components
5. **Add flyback diode** - Place a diode across solenoid coil to protect against voltage spikes

---

Testing the Setup

Before running the full application:

1. **Test distance sensor**: Run a simple GPIO test to verify readings
2. **Test relay**: Manually toggle GPIO 18 to hear the relay click
3. **Test solenoid**: Verify valve opens/closes when relay switches
4. **Verify voltage divider**: Use multimeter to confirm ~3.3V on GPIO 17

Raspberry Pi GPIO Pinout Reference

```
      3.3V  [ 1] [ 2]  5V
     GPIO2  [ 3] [ 4]  5V
     GPIO3  [ 5] [ 6]  GND
  GPIO4 (T)  [ 7] [ 8]  GPIO14
       GND  [ 9] [10]  GPIO15
GPIO17 (E)  [11] [12]  GPIO18 (R)
    GPIO27  [13] [14]  GND
    GPIO22  [15] [16]  GPIO23
      3.3V  [17] [18]  GPIO24
    GPIO10  [19] [20]  GND
     GPIO9  [21] [22]  GPIO25
    GPIO11  [23] [24]  GPIO8
       GND  [25] [26]  GPIO7
     GPIO0  [27] [28]  GPIO1
     GPIO5  [29] [30]  GND
     GPIO6  [31] [32]  GPIO12
    GPIO13  [33] [34]  GND
    GPIO19  [35] [36]  GPIO16
    GPIO26  [37] [38]  GPIO20
       GND  [39] [40]  GPIO21

Legend:
(T) = TRIG (GPIO 4)
```

(E) = ECHO (GPIO 17)
(R) = Relay (GPIO 18)

Alternative: Using Logic Level Converter

Instead of a voltage divider, you can use a **bi-directional logic level converter** (4-channel module) to safely interface between 5V and 3.3V devices. This is more reliable for multiple signals.

Troubleshooting

Problem	Solution
No distance readings	Check voltage divider, verify TRIG/ECHO connections
Raspberry Pi won't boot	5V may be shorted to GPIO - check all connections
Valve won't activate	Check relay power, verify GPIO 18 signal, test relay manually
Erratic sensor readings	Add 0.1µF capacitor across sensor VCC and GND
Relay clicks but valve doesn't open	Check external power supply to solenoid