

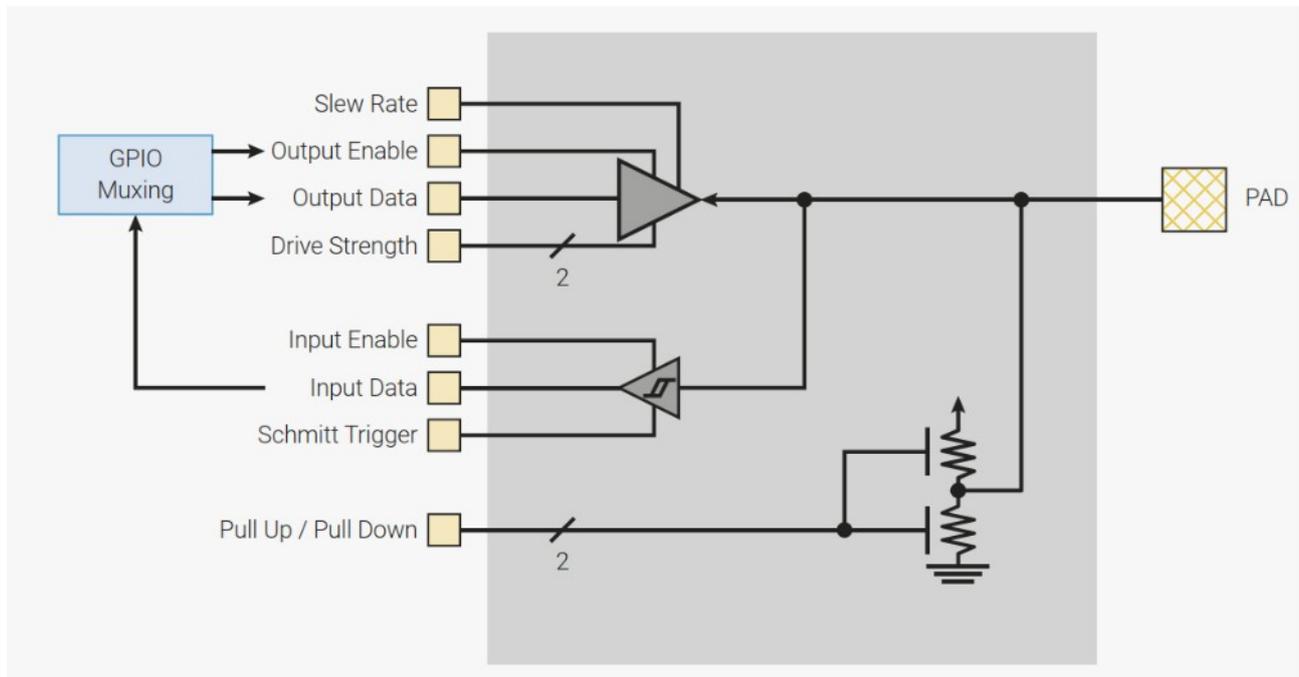
Part 16

-

Interfacing

Interfacing with ESP

The equivalent circuit of a GPIO pin from the ESP looks similar to those of the Raspberry Pi/ Pico



You should keep the following limitations in mind when using the GPIO pins

- These are 3.3 volt logic pins. A voltage near 3.3 V is interpreted as a logic one while a voltage near zero volts is a logic zero. A GPIO pin should never be connected to a voltage source greater than 3.3V or less than 0V, as prompt damage to the chip may occur as the input pin substrate diodes conduct. There may be times when you may need to connect them to out-of-range voltages – in those cases the input pin current must be limited by an external resistor to a value that prevents harm to the chip. I recommend that you never source or sink more than 0.5 mA into an input pin.
- To prevent excessive power dissipation in the chip, you should not source/sink more current out of the pin than its programmed limit. So, if you have set the current capability to 2 mA, do not draw more than 2 mA from the pin.
- Never demand that any output pin source or sink more than 12 mA.
- Current sourced by the outputs is drawn from the 3.3 V supply, which can supply only 50 mA maximum. Consequently, the maximum you can source from all the GPIO outputs simultaneously is less than 50 mA. You may be able to draw transient currents beyond that limit as they are drawn from the bypass capacitors on the 3.3 V rail, but don't push the envelope!
- Do not drive capacitive loads. Do not place a capacitive load directly across the pin. Limit current into any capacitive load to a maximum transient current of 16 mA. For example, if you use a low pass filter on an output pin, you must provide a series resistance of at least $3.3\text{V}/12\text{mA} = 280\ \Omega$.

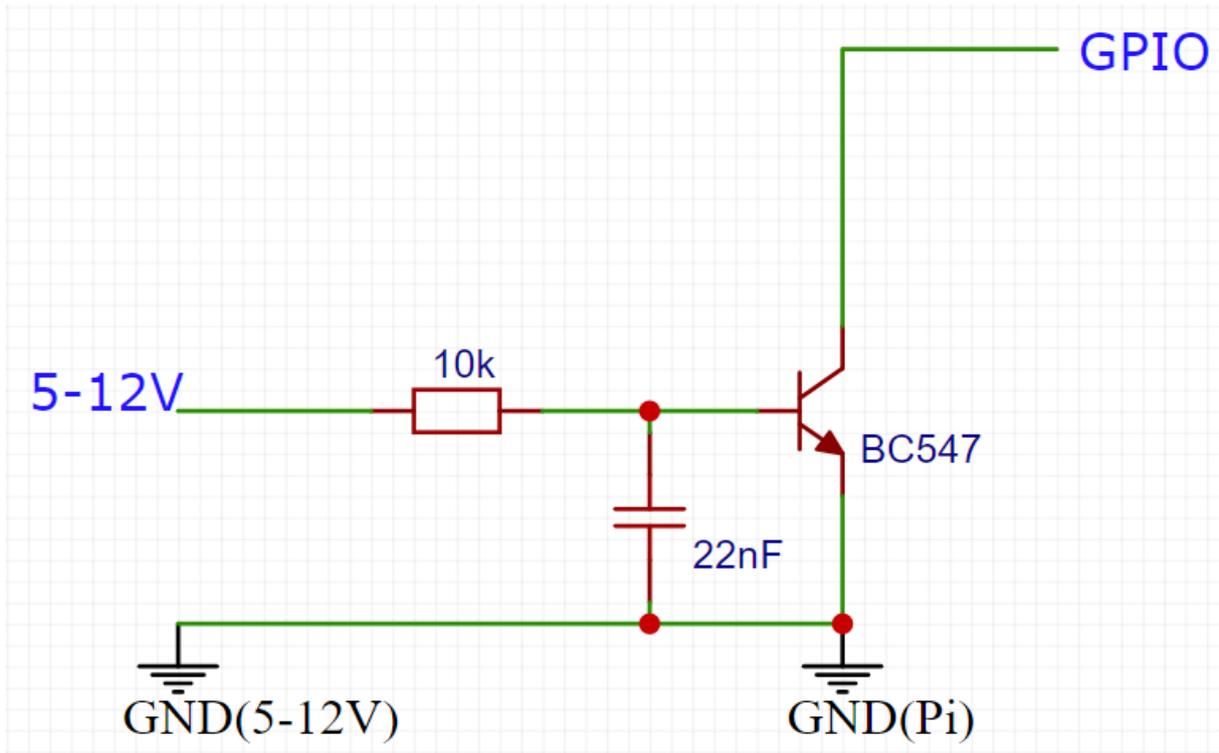
In order to interface with the outside world like sensor modules, relays etc, you will need some circuitry to adapt the voltage and current levels from or to the outside towards the Pi. There are frequently called 'levelshifters'. A level shifter will change the voltage level according to the circuits' power lines. E.g. the level shifter will change between 3.3v and 5v.

They can be unidirectional, but also bidirectional. I'll list some of the most important interface circuits here.

Unidirectional levelshifters but inverting the logic !!!

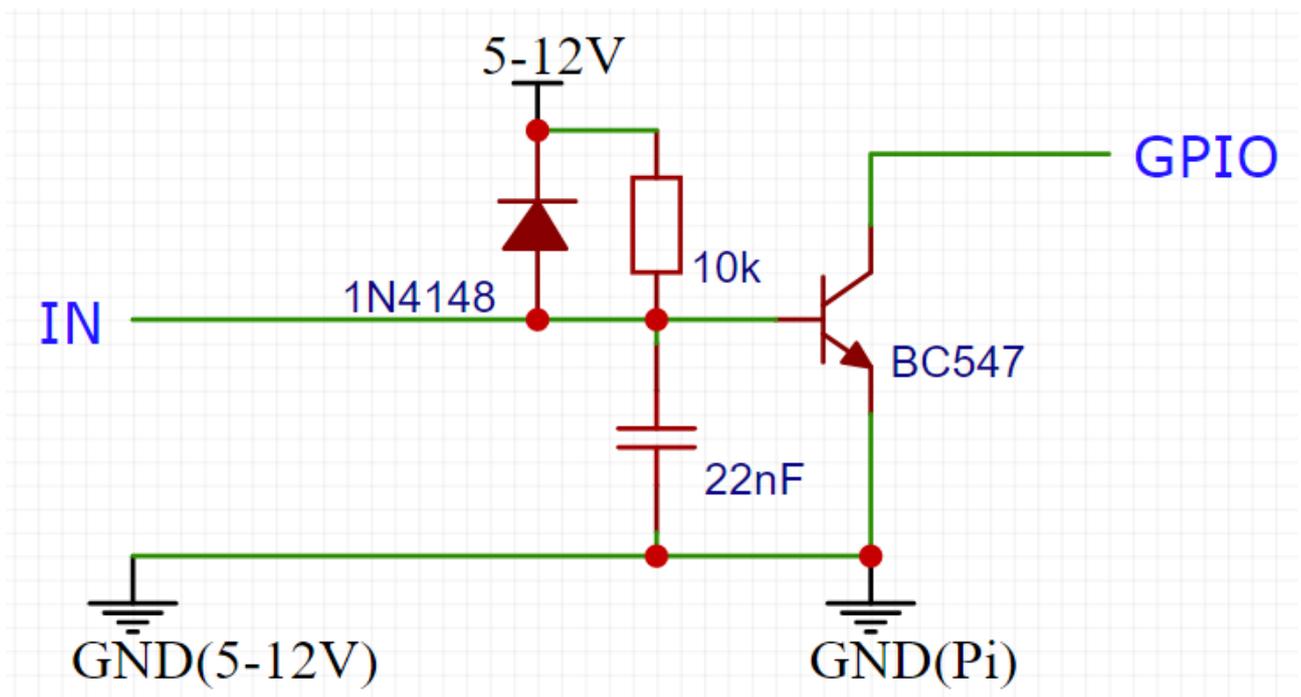
Input voltage provide by input device. Eg: when the outside sensor provides a high level of eg: 10V in the IN, the GPIO pin will see a LOW-level.

Note: make sure to use internal GPIO pull-up circuit.



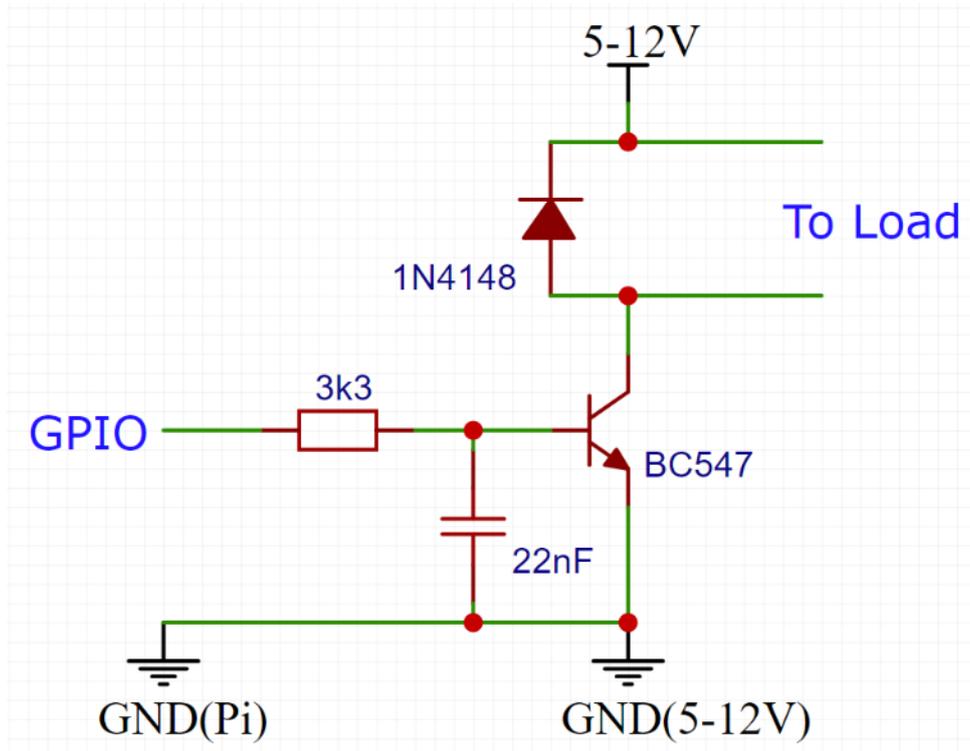
Input voltage provided by levelshifting circuit. Eg: sensor has an open collector or relay output. When the outside sensor pulls the IN line down, the GPIO pin will see a HIGH-level.

Note: make sure to use internal GPIO pull-up circuit.



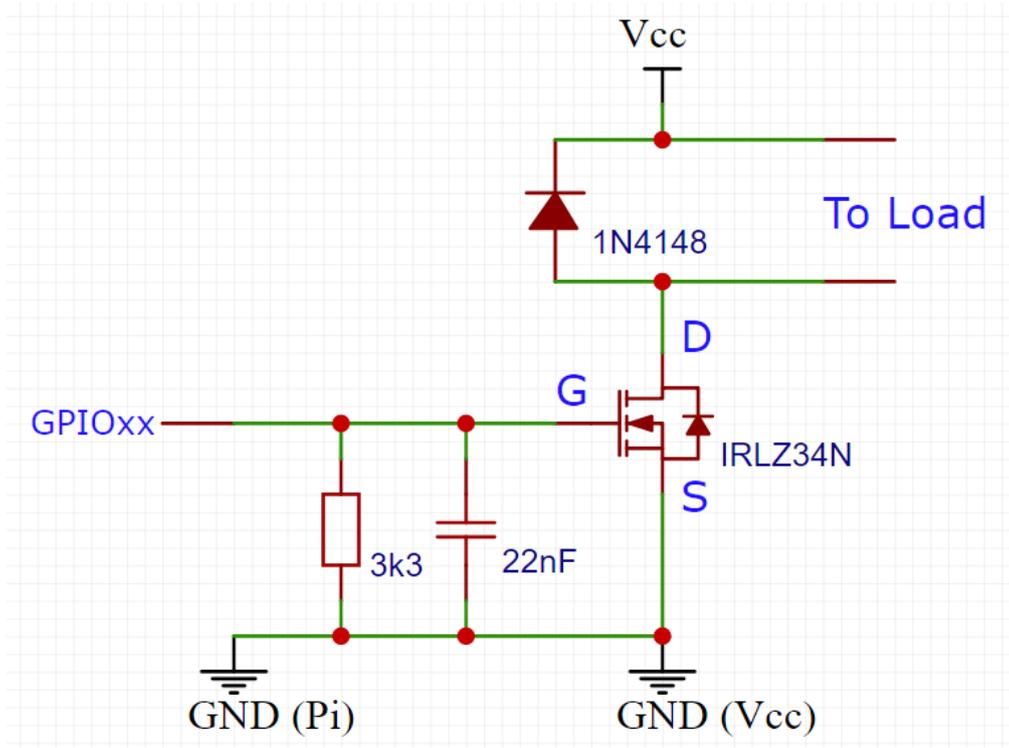
When the GPIO is set to HIGH, the LOAD will be powered.

Note: make sure to use internal GPIO pull-up circuit. This circuit can handle up to 100mA. Using another transistors like a darlington eg. TIP132, more current can be handled.

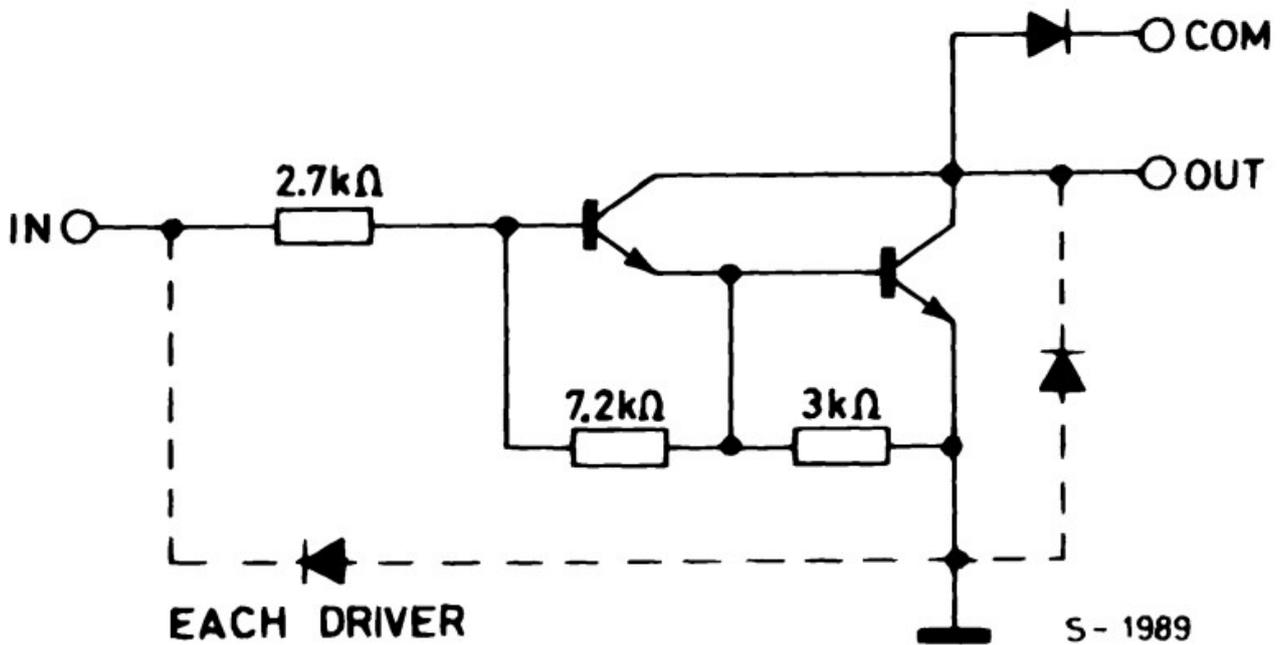


Instead of using classic transistors, you can use MOSFETs as well.

Using MOSFET 2N7000 will allow you to drive up V_{cc} to 60V and 100mA (@ $V_{gs} = 3.3V$) while using a IRLZ34N will even boost the current to 10A (@ $V_{gs} = 3.3V$)



If you need several GPIO pins to be interfaced, you might consider using an IC instead of discrete components
Eg: the ULN2003 has 7 of these drivers onboard capable of driving 50 V and 500mA per driver.



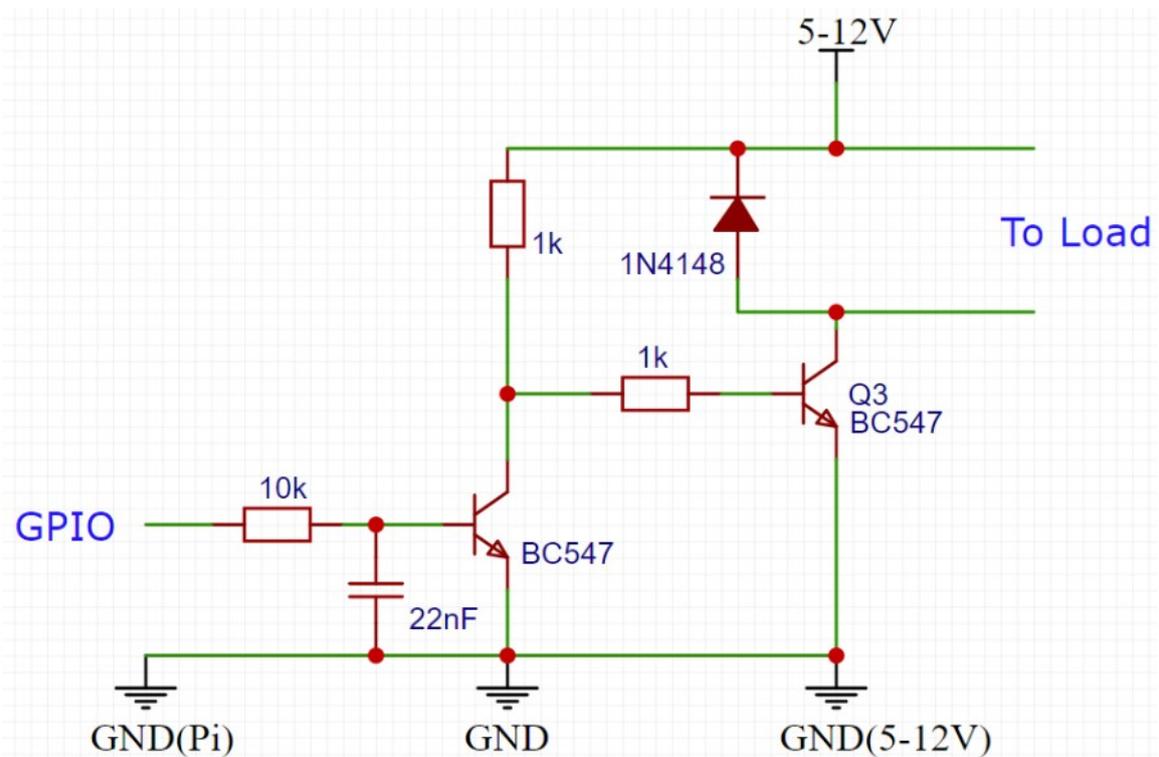
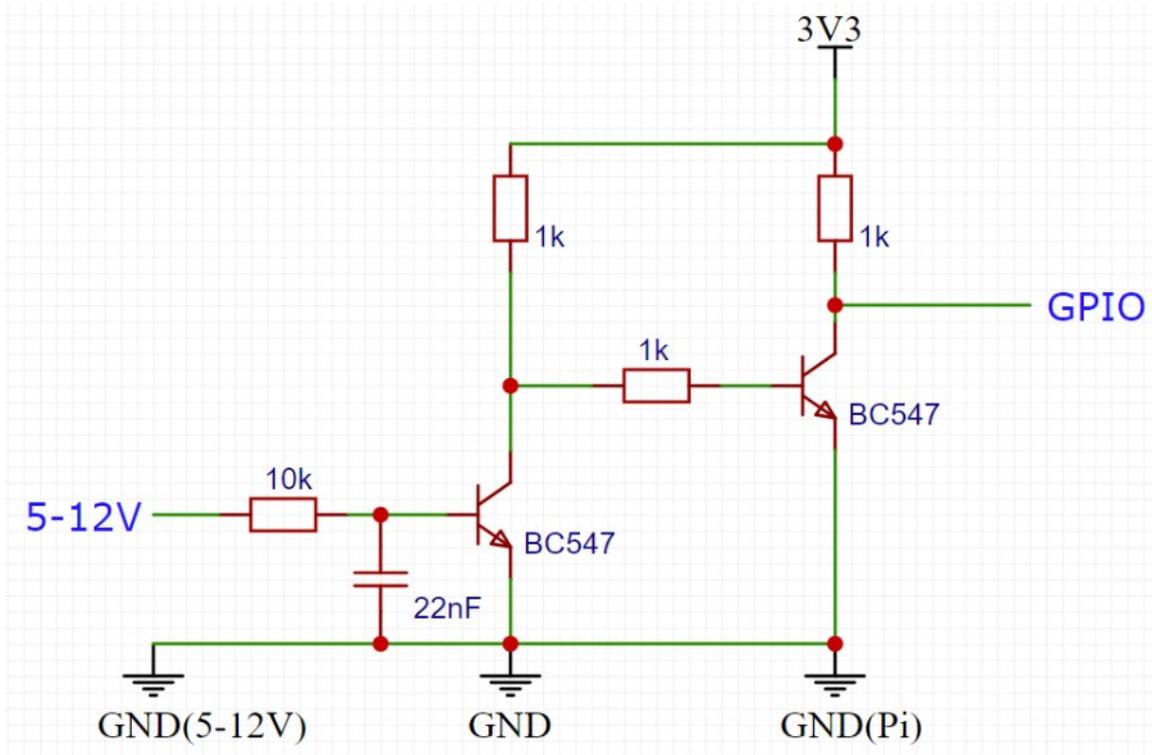
Some additional information:

- The 22nF capacitor is used to suppress all high frequency signals that might disturb the circuit. Hence also using the GPIO internal pull-up resistor is used for the same reason however is much more effective for low frequency signals eg: 50 or 60Hz from the electrical net.
- The 1N4148 multipurpose diode is used as a 'fly-back'-suppressor when inductive loads are used, 'killing' the reverse voltage when switching off inductive devices like relays.
- All resistors are 1/4W

Unidirectional levelshifters

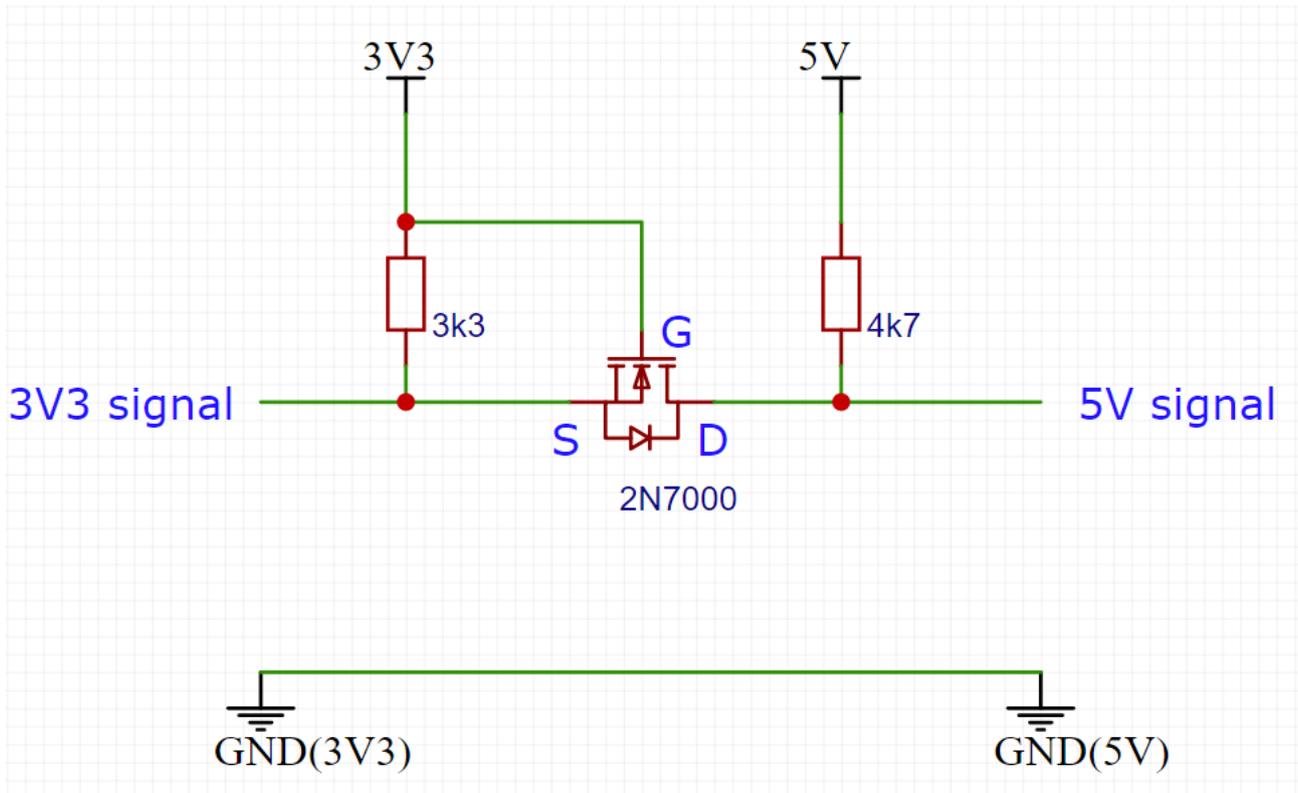
Here you see an input and an output circuit which does not inverted the logic. A HIGH in is a HIGH out.

Note: make sure to use internal GPIO pull-up circuit. This circuit can handle up to 100mA. Using another transistor for Q3 like a darlington eg TIP132, more current can be handled.



Bidirectional level shifter

For sensor circuit modules 5V is the most common voltage. So the interfaces used like I2C, SPI and serial are most likely 5V driven/levelled. Also USB is still a 5V protocol. These protocols are often bidirectional. The solution to stepping down and stepping up a voltage is to use a bidirectional level shifter. It's a very simple set up that uses a MOSFET and two resistors. The schematic is below.



Note that the Gate is always connected to the LOWER voltage. Any devices that share a signal line but are on two different voltages can use this set up. The level shifter shown above is adapted for I2C devices working on different voltages. The resistors would then take the values of the corresponding pull-up resistor for the I2C line, usually 4.7K on the 5v side and 3.3K on the 3.3v side. This will provide a suitable current of 1mA. Be careful here, don't have two sets of pull-up resistors on the same line, if you already have them installed somewhere else then don't include them again.

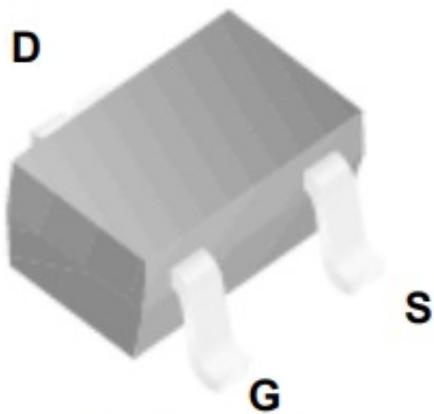
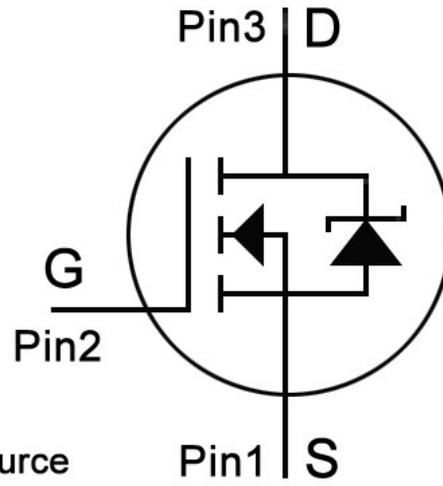
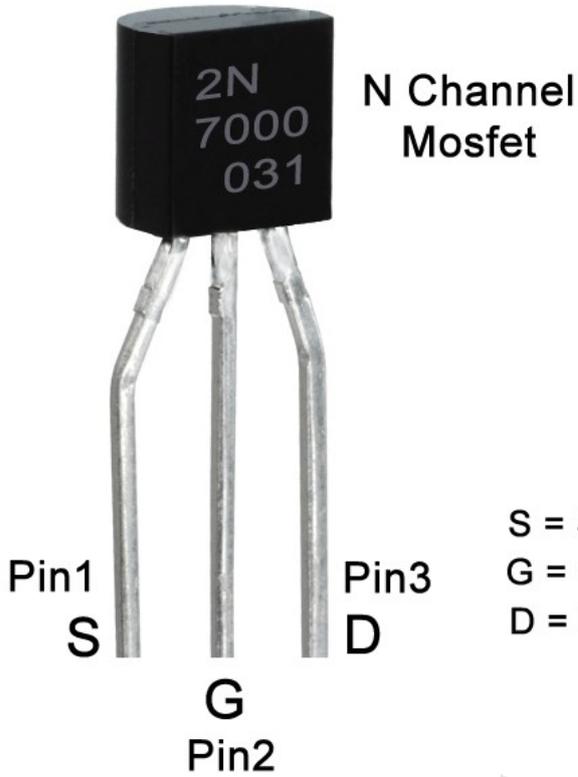
10K resistors on either side work well with serial data.

Again, if several GPIO ports need to be adjusted, consider using an IC or modules instead of discrete components.

Eg: Texas Instruments TXB0108PWR or Adafruit 8-channel Bi-directional Logic Level Converter, Adafruit

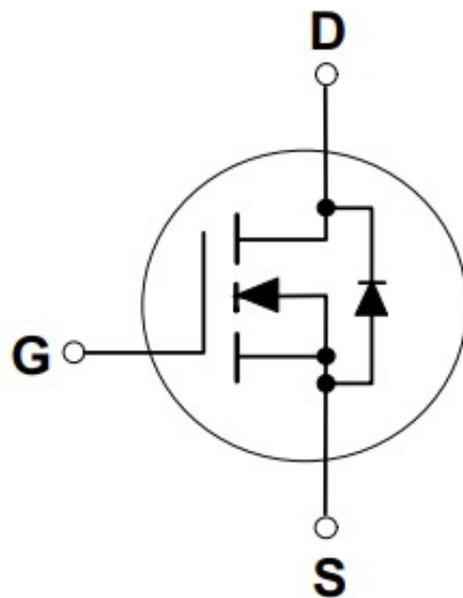
2N7000/2N7002

TO-92 Package



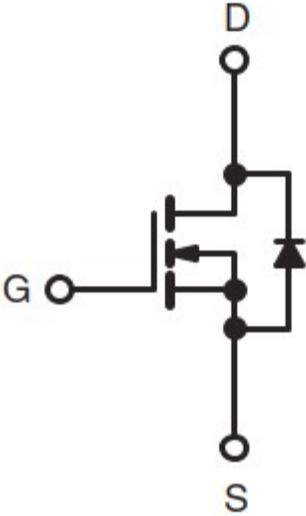
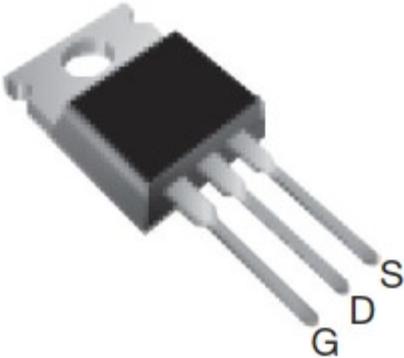
SOT-23
(TO-236AB)

2N7002/NDS7002A



IRLZ34N

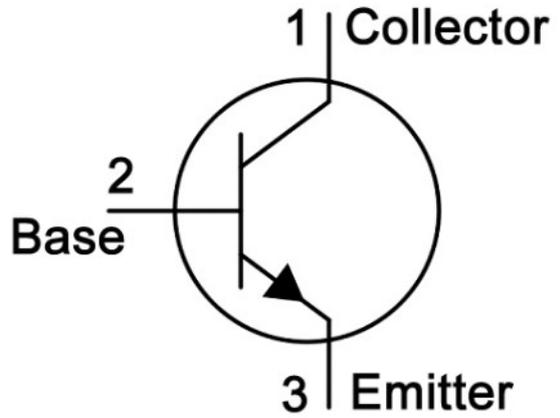
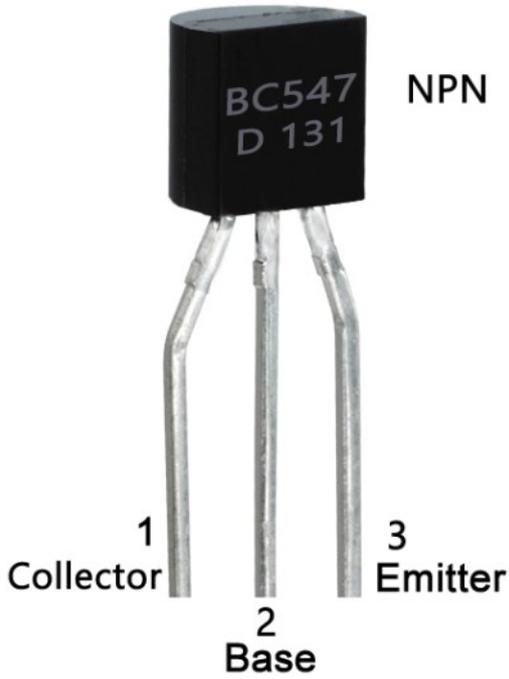
TO-220AB



N-Channel MOSFET

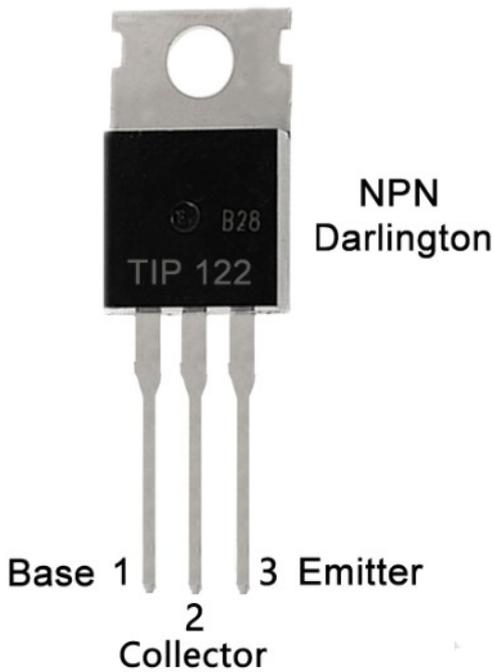
BC457

TO-92 Package

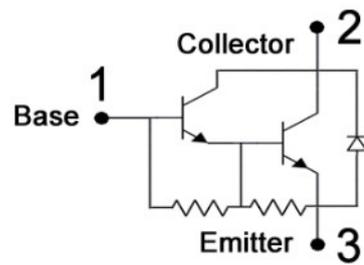


TIP132

TO - 220 Package



Internal Circuit



Symbol In Diagram

