



4-Bit I²C-Bus LED Driver with Programmable Blink Rates

PCA9553

Last Updated: Sep 5, 2022

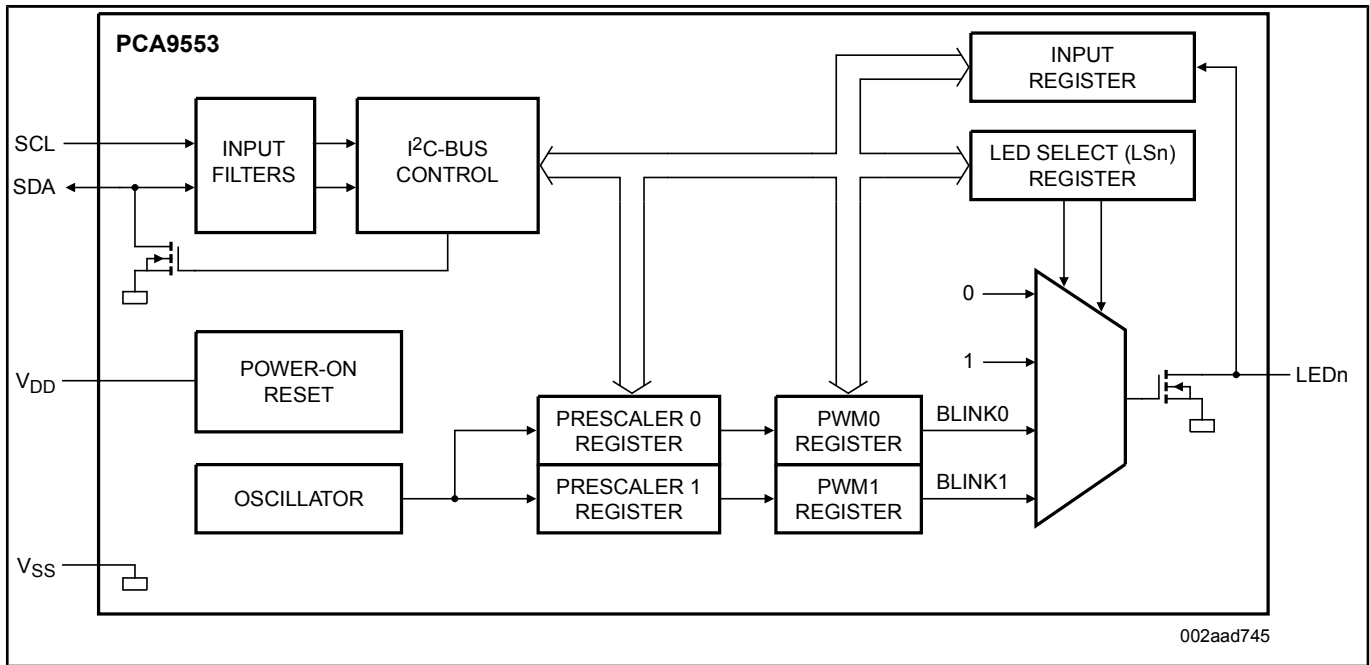
The PCA9553 LED blinker blinks LEDs in I²C-bus and SMBus applications where it is necessary to limit bus traffic or free up the I²C-bus controller's (MCU, MPU, DSP, chipset, etc.) timer. The uniqueness of this device is the internal oscillator with two programmable blink rates. To blink LEDs using normal I/O expanders like the PCF8574 or PCA9554, the bus controller must send repeated commands to turn the LED on and off. This greatly increases the amount of traffic on the I²C-bus and uses up one of the controller's timers. The PCA9553 LED blinker instead requires only the initial set-up command to program BLINK RATE 1 and BLINK RATE 2 (i.e., the frequency and duty cycle). From then on, only one command from the bus controller is required to turn each individual open-drain output ON, OFF, or to cycle at BLINK RATE 1 or BLINK RATE 2. Maximum output sinks current is 25 mA per bit and 100 mA per package.

Any bits not used for controlling the LEDs can be used for General Purpose Parallel Input/Output (GPIO) expansion.

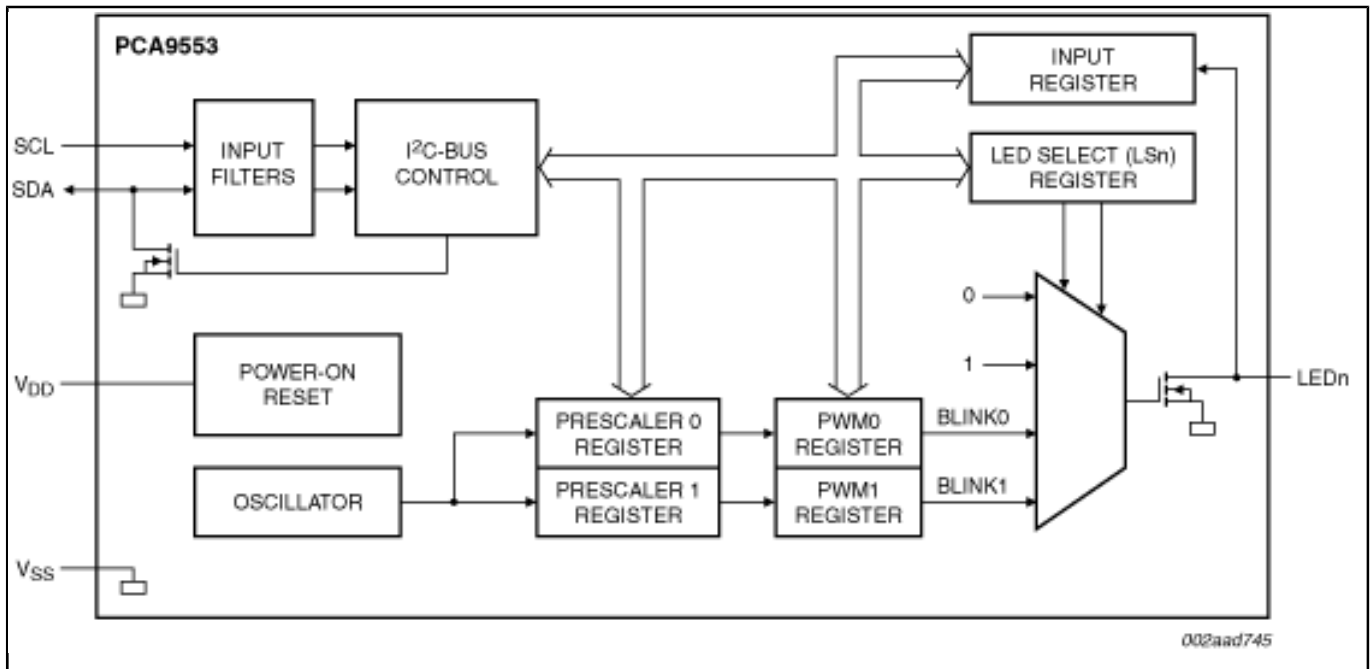
Power-On Reset (POR) initializes the registers to their default state, all zeroes, causing the bits to be set HIGH (LED off).

Due to pin limitations, the PCA9553 is not featured with hardware address pins. The PCA9553/01 and the PCA9553/02 have different fixed I²C-bus addresses allowing operation of both on the same bus.

PCA9553 Block Diagram Block Diagram



Block diagram: PCA9553D, PCA9553DP, PCA9553TK Block Diagram



View additional information for [4-Bit I²C-Bus LED Driver with Programmable Blink Rates](#).

Note: The information on this document is subject to change without notice.

www.nxp.com

NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. © 2024 NXP B.V.