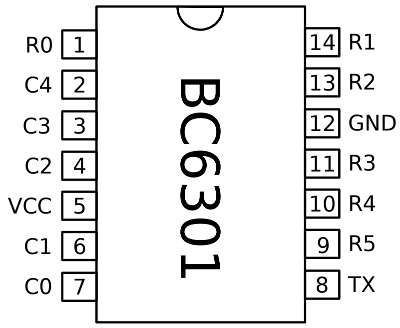
BC6301

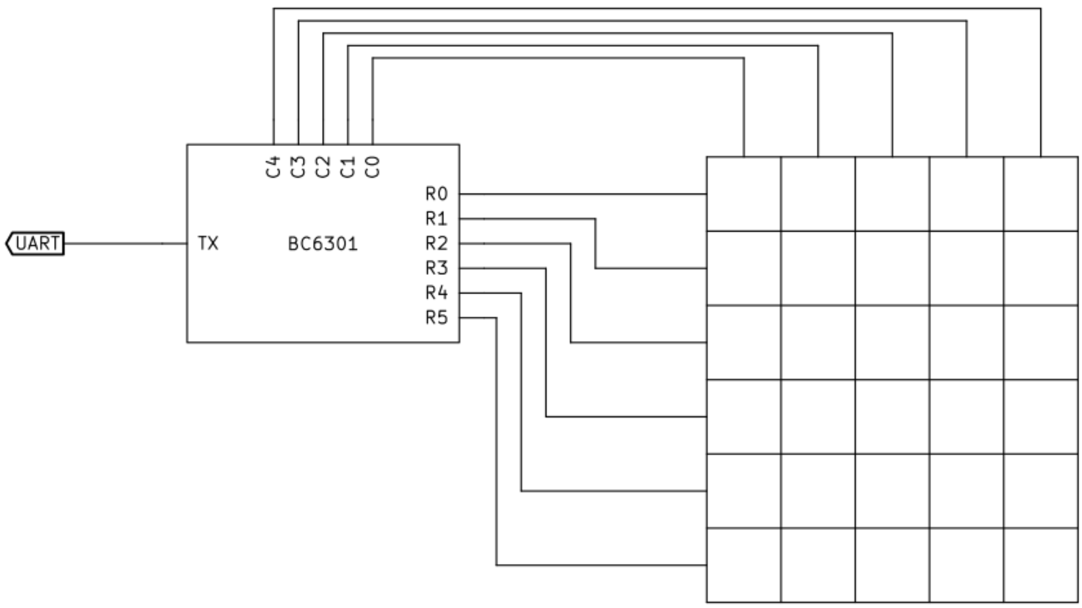
A Single Wire UART 5x6 Keyboard Driver

Features

- Supports combination keys
- Supports long-press key detection
- Compatible with normally open and normally closed switches
- Direct key value output
- Operating voltage: 2.2–5.5V
- ingle-wire UART interface
- Operating current: 2.7 µA (@3V)
- SOP14 package
- Protocol compatibility with other BC6xxx series chips

Pin Diagram





Overview

The BC6301 is a dedicated keyboard interface chip capable of driving a 30-key (5x6) keyboard matrix. It supports various types of keypads, including metal contact, conductive rubber, and mechanical switches. Data is output through a single-wire UART interface, simplifying circuit design by requiring only one I/O line. This setup enables easy integration with optocouplers, RS-485, or other interfaces for electrical isolation and remote keyboard control. The chip uses open-drain output, ensuring compatibility with microcontrollers (MCUs) operating at different power supply voltages.

The BC6301 outputs key values directly, simplifying programming. Applications need only process key value data received from the UART interface. It supports arbitrary combinations of keys, as well as both normally open and closed switches, providing flexibility. With a timer, it can detect long-press key events, including those involving combination keys.

Its wide operating voltage range and low power consumption make it well-suited for battery-powered devices.

Pin Definitions

Pin Name	Pin Number	Description
VCC	5	Positive power supply (2.2–5.5V)
GND	12	Ground
TX	8	UART output (open-drain)*
R0～R5	1,9,10,11,13,14	Keyboard matrix row inputs (internal pull-down resistors)
C0～C4	2,3,4,6,7	Keyboard matrix column outputs

* Voltage on TX should not be higher than VCC.

Key Value Mapping Table

	C0	C1	C2	C3	C4
R0	0(0x00)	6(0x06)	12(0x0c)	18(0x12)	24(0x18)
R1	1(0x01)	7(0x07)	13(0x0d)	19(0x13)	25(0x19)
R2	2(0x02)	8(0x08)	14(0x0e)	20(0x14)	26(0x1a)
R3	3(0x03)	9(0x09)	15(0x0f)	21(0x15)	27(0x1b)
R4	4(0x04)	10(0x0a)	16(0x10)	22(0x16)	28(0x1c)
R5	5(0x05)	11(0x0b)	17(0x11)	23(0x17)	29(0x1d)

UART Output Format

The output of the BC6301 is a standard UART format with a baud rate of 9600, 8 data bits, 1 start bit, and 1 stop bit. When the keyboard status changes, such as a key being pressed (on) or released (off), the BC6301 immediately outputs the key value of the changed key. The key value is the numerical value in the key value table. If the key status changes from off to on, the unmodified key value is output. If the key status changes from on to off, the most significant bit (MSB) of the key value is set to 1. For example, if the original key value is 0x01, it becomes 0x81; if the original value is 0x17, it becomes 0x97, and so on. If the status of multiple keys changes simultaneously, the values of all changed keys are output in ascending order of their original key values. For example, if the status of keys 5, 18, and 22 changes simultaneously, with keys 5 and 22 changing from off to on and key 18 changing from on to off, the data output on the UART will be 0x05, 0x92, and 0x16 sequentially.

When the system powers on, the BC6301 chip defaults to all keys being in the off state. If any keys are in the on state at power-on, the BC6301 will output their key values after it starts scanning the keyboard. In the most extreme case, if all keys used are normally closed, meaning all keys are in the on state at power-on, the BC6301 will output 30 bytes of data consecutively after power-on. After power-on, the BC6301 has a delay of approximately 300ms before it starts scanning the keyboard. This ensures that the MCU completes the power-on reset process and can correctly receive UART data.

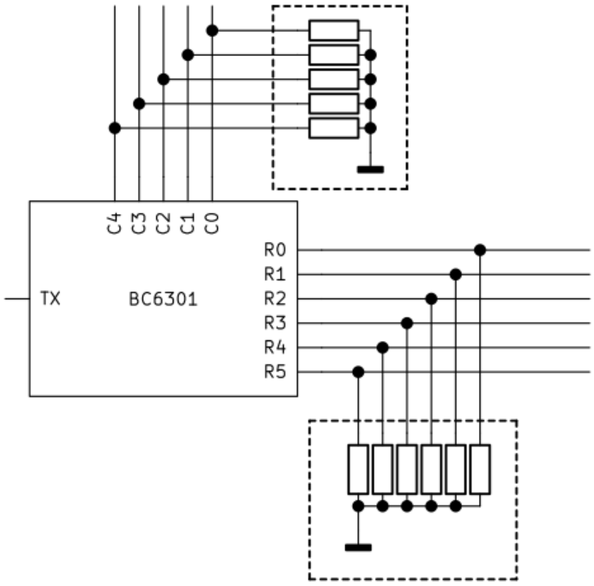
Typical Circuit

The BC6301 does not require any external components. The UART output of the BC6301 uses an open-drain output, but the voltage on the TX should not exceed the voltage on BC6301's VCC. Theoretically, a pull-up resistor should be connected. However, since many MCUs or UART receivers already have internal pull-up resistors on their inputs, an additional pull-up resistor can be omitted in such cases. The specific situation depends on the connected device.

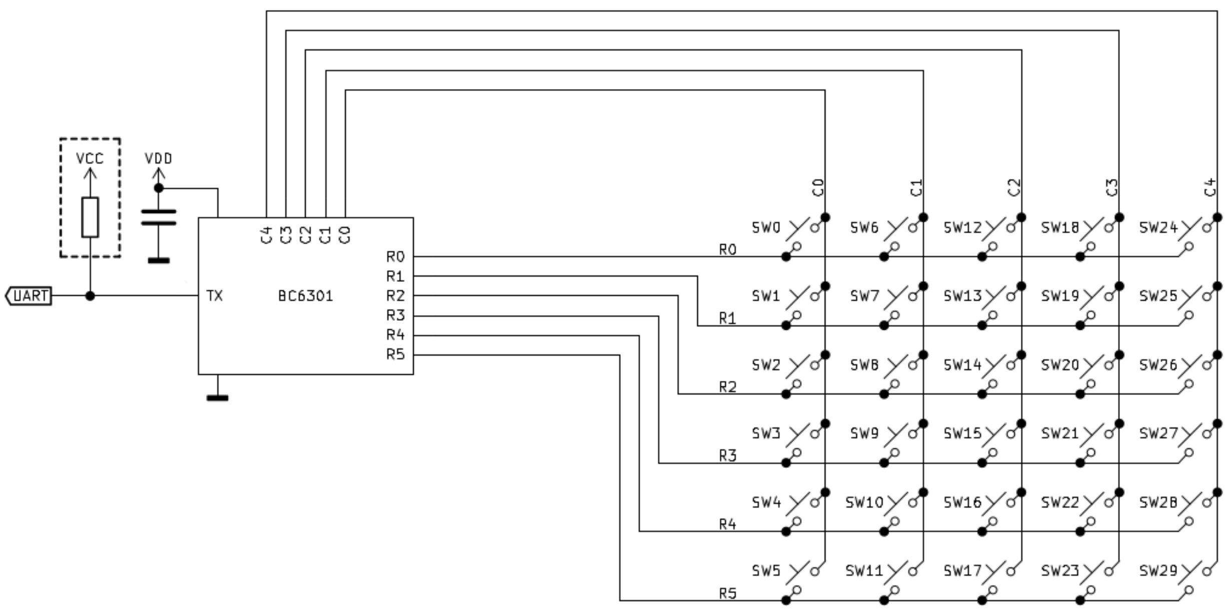
The keyboard matrix consists of 6 rows and 5 columns. The column signals C0-C4 output positive scanning pulses in turn, and the row signals R0-R5 are input signals. The BC6301 has internal pull-down resistors, so in general applications, there is no need to connect additional external pull-down resistors. A typical circuit is as follows:

The scanning pulse width output by the column signal of the BC6301 is approximately 14ms, the scanning cycle is 71ms, and the debounce time is 14ms. This ensures that the required key duration for a key action to be recognized is 71ms (normal key operation duration is about 200-500ms), while the shortest recognizable key duration is 14ms. If the keyboard has a large distribution area or there is strong interference in the surrounding environment, consider adding extra pull-down resistors to both the row signals Rn and column signals Cn to enhance the ability to suppress interference. The resistance value of the external pull-down resistor should be no less than 1kΩ, and the smaller the resistance value, the larger the overall working current of the system.

As a general design principle, the unused areas of the circuit board where the keyboard is located should be covered with a ground plane.



A typical application circuit:



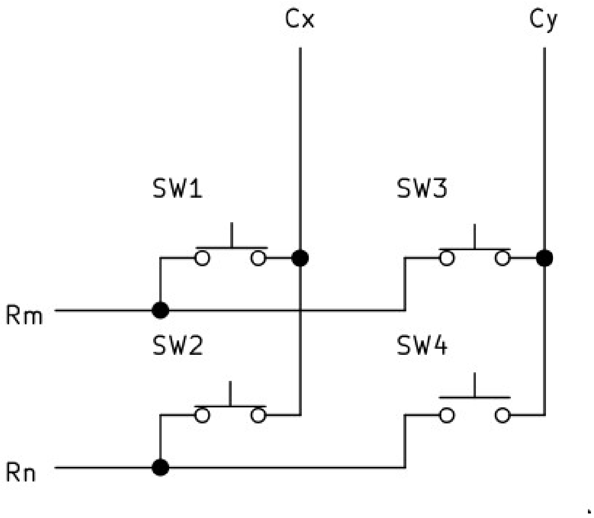
Eliminating Ghost Keys

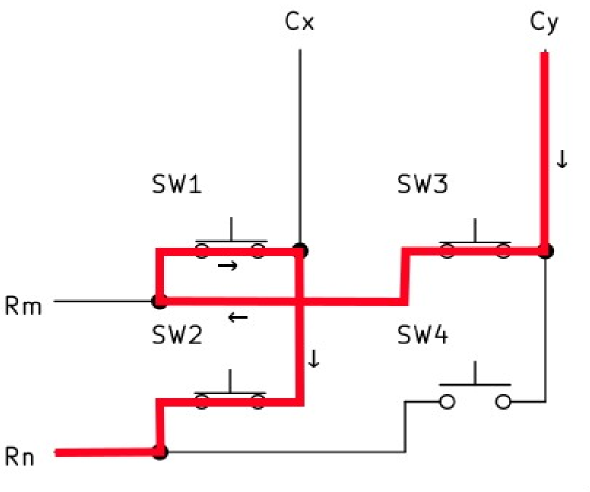
Ghost keys are a phenomenon specific to rectangular keyboard matrices. They occur when three out of four keys located at the four corners of a quadrilateral are in the on state, the system mistakenly recognizes the fourth key as also being in the on state. For instance, in the figure, when SW1, SW2, and SW3 are all pressed, the system mistakenly believes that SW4 is also pressed.

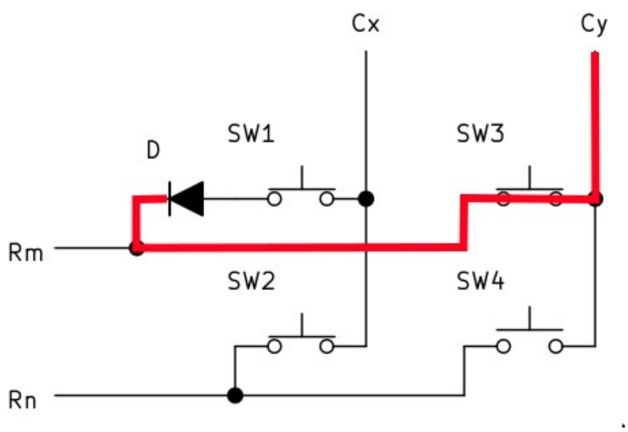
The reason for this phenomenon is that when SW1-SW3 are all in the on state, the current loop formed by the three keys actually connects both ends of the fourth key, SW4. Assuming the system scans SW4 and Cy outputs a high level, the current will pass through the following loop, causing Rn to be at a high level, which is the same as the effect of SW4 being pressed. [See figure showing current flow path]

To completely eliminate the possibility of ghost keys, diodes need to be inserted in series with all keys. However, this approach is costly. In typical practical applications, measures only need to be taken for individual keys that need to prevent ghosting. Since diodes have a forward voltage drop, which reduces the reliability of the keyboard scanning circuit, diodes with a low forward voltage drop, such as Schottky diodes, should be selected.

The solution is to insert a diode in the diagonally opposite key SW1 of the unpressed key. This diode can block the current loop from forming when SW1-SW3 are all on, thus preventing the system from recognizing the fourth key SW4 as being in the on state.





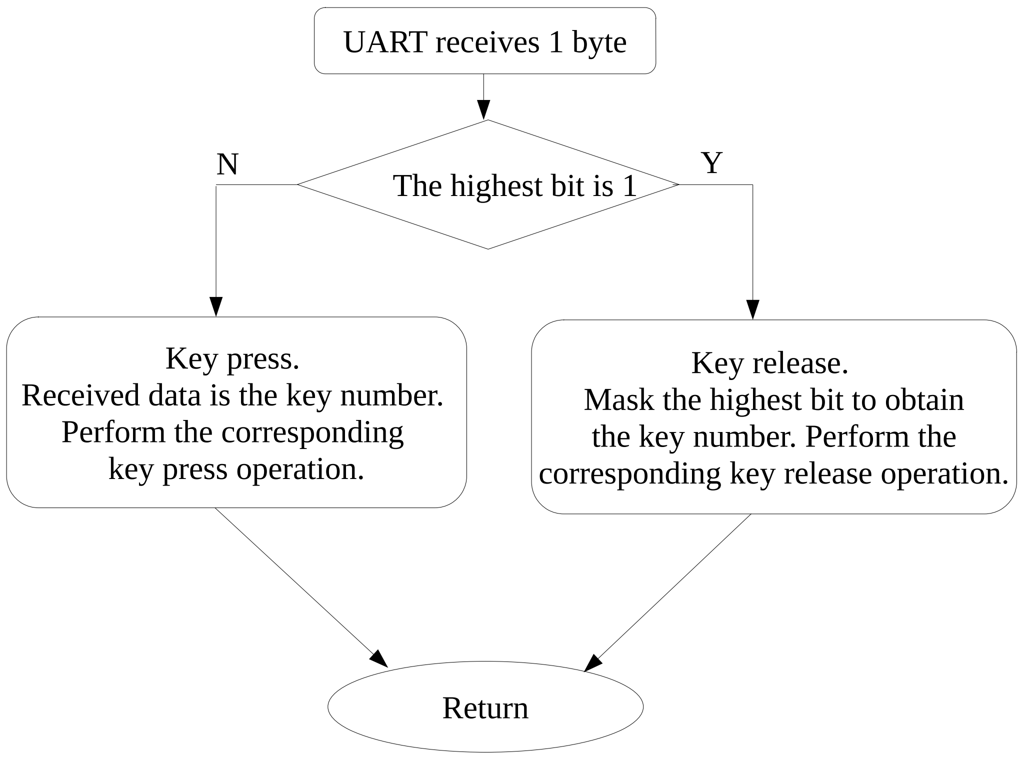


Typical Key Processing Flow

The optimal application for the BC6301 is to utilize a hardware UART interface and handle key signals via interrupts. This method minimizes system overhead, allowing the processor to dedicate most of its time to other tasks. Polling-based approaches or software-based UART implementations for receiving data are not recommended due to their significant processor time consumption.

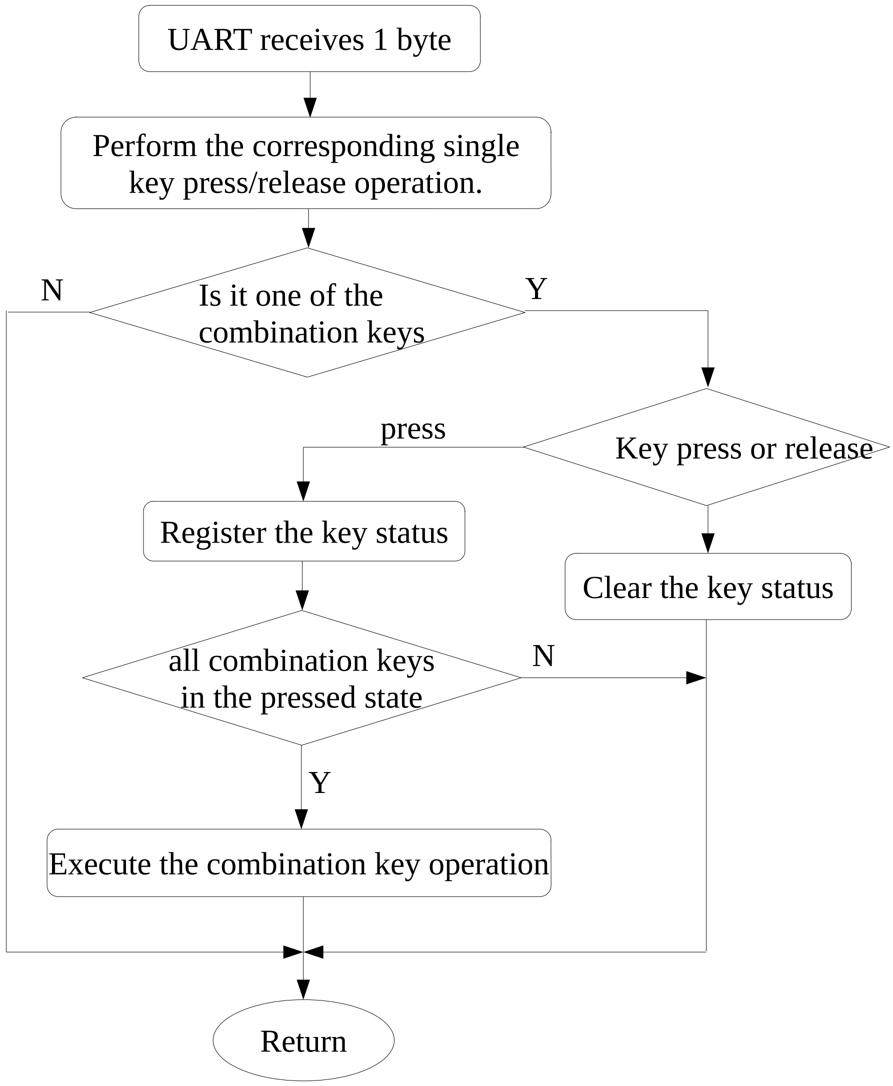
At the BC6301's baud rate of 9600, transmitting a single byte takes approximately 1ms. Even with potential simultaneous key state changes, where the BC6301 might send multiple bytes consecutively, this 1ms interval should be sufficient for most processors to complete key processing before the arrival of the next byte. If your application involves the possibility of multiple simultaneous key changes, and your system cannot guarantee processing completion within 1ms, consider allocating a UART receive buffer. This buffer should be sized larger than the maximum number of keys that may change simultaneously to prevent buffer overflow and potential loss of key data.

Normal Single Key Press (Press and Release Processed Separately)



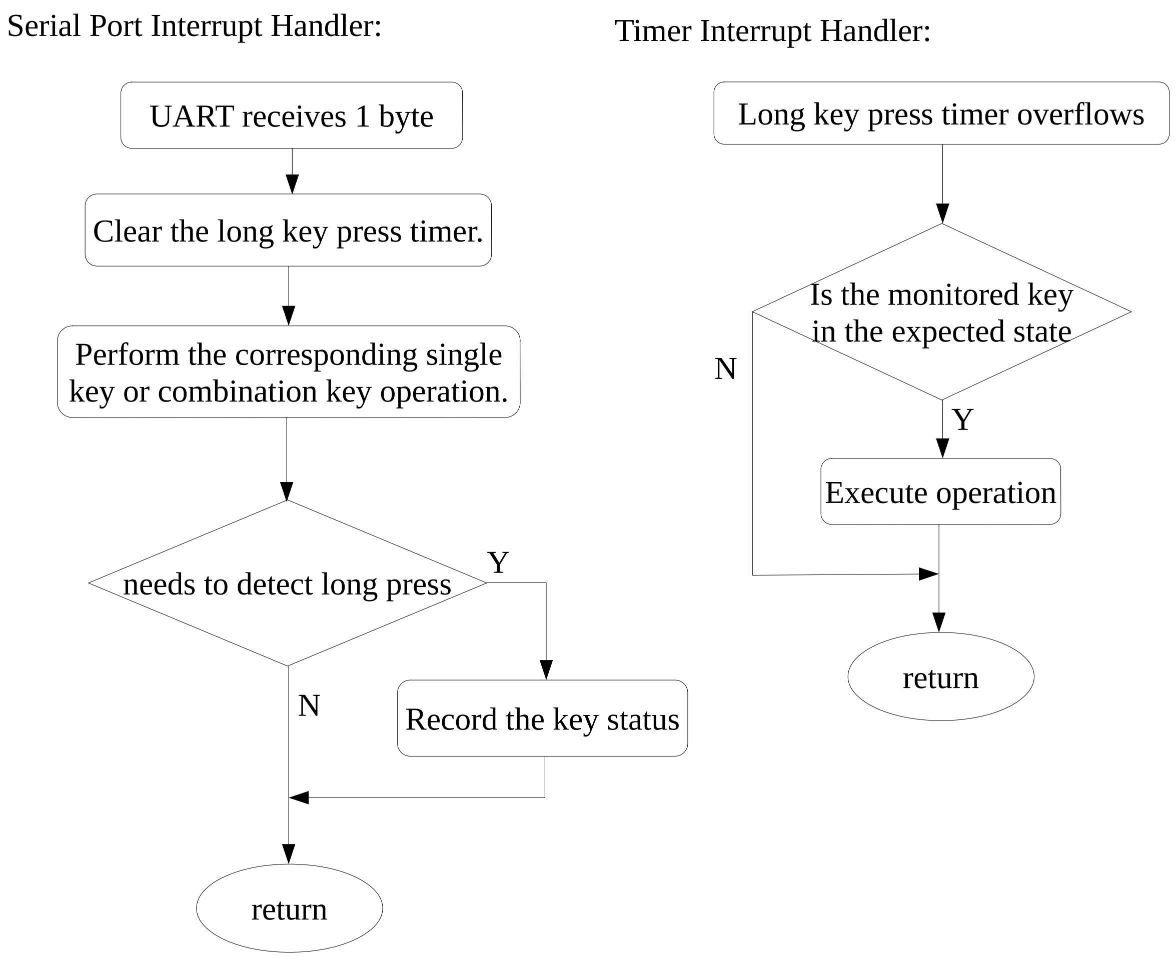
Combination Keys

The use of combination keys requires the program to remember the status of each key in the key combination.



Long Key Press

The BC6301 sends corresponding information from the UART interface when the keyboard status changes, regardless of whether a key is pressed or released. If there is no data output on the UART, it means that the keyboard state remains unchanged since the last data output. When a key is pressed, if no data about the key's state change is received for a certain period, it indicates that the key has been held down during this period. Not only can single key long presses be detected, but combination keys can also be detected using the same method. The user only needs to provide a timer to determine the duration of the long key press that needs to be detected.



Controller (MCU) Wake-up

The BC6301 has extremely low power consumption. If powered by two AA alkaline batteries (calculated based on 1800mAh capacity), the static working current of 2.7uA@3V allows it to operate continuously for more than 5 years, making it ideal for battery-powered devices. Battery-powered devices are usually designed so that the controller (MCU) is in a sleep state when idle and wakes up when there is user operation (keyboard operation).

The UART output is the only signal transmitted from the BC6301 to the MCU. Therefore, if the MCU needs to be woken up by keyboard operation, the MCU needs to be able to be woken up by the UART signal. Depending on the design, some MCUs can support direct wake-up by UART. These products can perfectly cooperate with the BC6301 without further discussion. Here we mainly discuss MCUs that cannot be directly woken up by the UART signal.

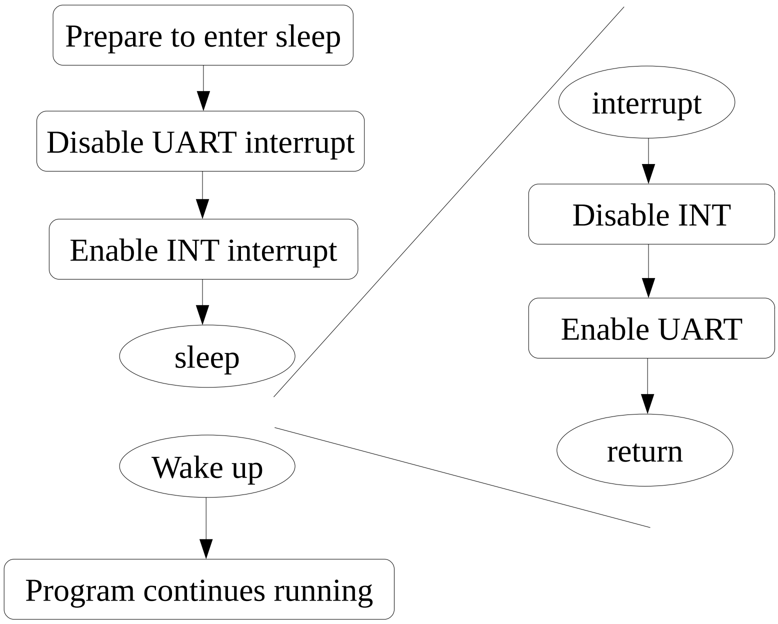
One solution is to design a separate wake-up button directly connected to the MCU. The BC6301 is directly powered by the MCU's pin, and the BC6301 circuit can be completely shut down during sleep. Another solution uses only the BC6301, with the UART signal also serving as the MCU wake-up interrupt signal. General MCUs have the function of being woken up by external interrupts, and the UART signal is high when idle and starts with a start bit '0', meaning that all UART data starts with a falling edge. Therefore, the UART signal can be simultaneously provided to the external interrupt of the MCU to wake up the MCU.

Since the wake-up time of the MCU is generally long, using this method to wake up the MCU may result in the first byte of data on the UART not being received normally. This is generally not a problem, and the first byte received after wake-up can be directly ignored. From the user's perspective, the user experience will be: when the machine is in sleep state, press any key to wake up the machine, and the display, indicator lights, etc. will recover, and then press the required function keys to perform the required operations.

If it is necessary to capture the first key press, you can also consider the following method: Since the BC6301 sends two signals when the key is pressed and released, the interval between them is the key's duration, which is generally between 200ms and 500ms. This time should be sufficient for the MCU to enter the running state, so the key release signal can be ensured to be captured. The program can use the key release signal to determine the key value and execute the corresponding action.

When using the latter scheme, the user must know the exact time it takes for the MCU to wake up from sleep and enter the running state to confirm whether the first byte sent by the BC6301 can be correctly received when the MCU is in sleep state. The following flowchart assumes that the MCU wake-up time is > 938us.

Main Program: Wake-up Interrupt (INT):



Absolute Maximum Ratings

Storage Temperature	-65°C - +150°C
Operating Temperature	-40°C - +85°C
Any pin to ground voltage	-0.5 - +6.0V

Electrical Characteristics

Note: Unless otherwise specified, TA=25∘C)

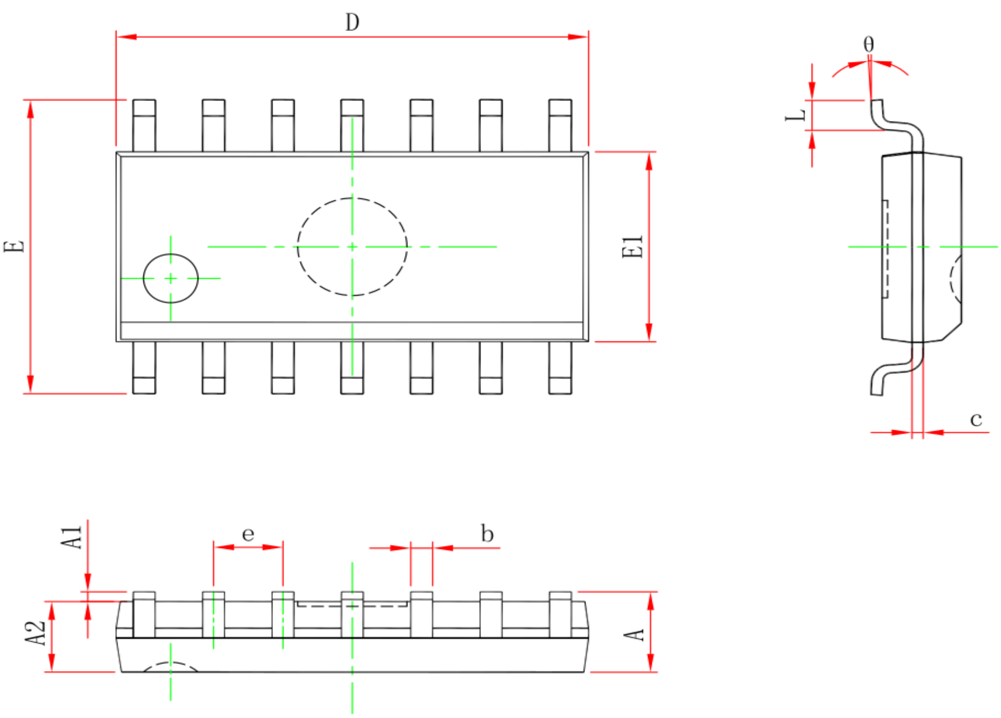
Parameters	Min	Typical	Max	Remarks
Operating Voltage V_{CC}	2.2V		5.5V	
Supply Current I_{CC}		2.7uA		V_{CC}=3.0V, no external connections*
		9.0uA		V_{CC}=5.0V, no external connections*
Peak Current I_P **			<0.5mA	V_{CC}=3.0V
			<1mA	V_{CC}=5.0V
Baud rate drift Δ_F			±2%	VCC=3.0V, -40°C-+85°C
			±1%	VCC=5.0V, -40°C-+85°C
Row scan input high level V_{RH}	2.0V			V_{CC}=3.0V
	4.0V			V_{CC}=5.0V
Keyboard scan cycle T_{CL}		71ms		V_{CC}=3.0V
Keyboard scan pulse width T_{PL}		14ms		
Debounce time T_{db}		14ms		
Power-up wait time T_{pu}		300ms		V_{CC}=3.0V
Row input equivalent pull-down resistor R_{PD}		100K		V_{CC}=3.0V
		55K		V_{CC}=5.0V

* Due to distributed capacitance losses, the static operating current increases when the keyboard circuit is connected.

** Refers to the instantaneous operating current when the key state changes.

Dimensions

SOP14（150mil） PACKAGE OUTLINE DIMENSIONS



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	— —	1.750	— —	0.069
A1	0.100	0.250	0.004	0.010
A2	1.250	— —	0.049	— —
b	0.310	0.510	0.012	0.020
c	0.100	0.250	0.004	0.010
D	8.450	8.850	0.333	0.348
E	5.800	6.200	0.228	0.244
E1	3.800	4.000	0.150	0.157
e	1.270(BSC)		0.050(BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

Ordering Information

Part Description BC6301 E S

Part Number

E = -40˚C - +85˚C

S = SOP14

Packaging Information

Ordering Part Number	Packaging	Qty per package
BC6301ES-T	Tube	10000