

Raspberry Pi Expansion Module User Manual

52PI-RPI-Infinity0

Maximum Power at Minimum Size

Raspberry Pi Infinity cascade IO expand Module User Manual

Contact Us

Raspberry Pi Geek Forum

Version 1.0

Data: 2014.12

Raspberry Pi Infinity cascade IO expand Module User Manual

Raspberry Pi Geek Forum Copyright Statement

The document only describes the information about the product; however, it cannot guarantee the product's function and performance. If the document content or the product's features and technical specifications included in the document are changed, it will not be further noticed.

Content in the document might be past due. Our cooperation can not promise to update these information.

Some information in the document might be disabled in your local area, which includes products and services. You can consult with contacts and agencies in your local area.

Copyright to the document belongs to Raspberry Pi Geek Forum. Users are only allowed to use this content after they obtain authorization from our company or other obligees. However, users are not allowed to copy, paraphrase, or create similar devices or products.

The final right to interpret this document belongs to the final interpreter, which belongs to Raspberry Pi Geek Forum.

More information:

Get more product and support, please contact Raspberry Pi Geek Forum.

Attention:

Due to technical requirements of components, please do not handle directly connected. The core board and development system contain static-sensitive components. Quiet electrical charges easily accumulate in the human body and the device cannot detect possible damage to equipment. It is recommended to take anti-static measures, it is recommended not to handle the touch, stored in anti-static effect devices.



RaspberryPiExpand Module Serial

Raspberry Pi Infinity cascade IO expand Module User Manual

Infinity cascade IO expand module

This module is designed specifically for the Raspberry Pi IO expansion modules. The module expand 32 IO, Multiple modules can cascade, infinity cascade, infinity GPIO.

I2CGPIO expand module:

This module is designed specifically for the Raspberry Pi IO expansion modules. The Module use I2C bus to connect to Raspberry Pi. The module expand 8 Bidirectional GPIO and with isolation protection function which can effectively excessive external voltage. There are 8 I2C address, you can choose one of them through setup the jumper. Multiple modules can cascade and maximum cascade 8 modules!

Prototype development module

The Prototype development module is designed specifically for the Raspberry Pi. The module suitable enthusiasts and user can weld peripheral to the module; The module expand some amphenol connector and some SMT, so the user can finish prototype test easily.

Berryclip expand module:

The BerryClip module is designed specifically for learning how to use the GPIO of Raspberry Pi. There are 6 multiple color LED, 1 button and 1 Buzzer on the module.

Berryclip(DIY) expand module

The module is not the end product, you need weld them by yourself. The function of the module is the same as BerryClip module.

UNO compatibility module

The module makes Raspberry Pi compatible with Arduino Uno and many Arduino Shields. The module's GPIO is the same as Arduino Uno and you can choose the voltage of GPIO between 5V or 3V through setup jumper.

T Electric level convert module

The module expand the GPIO of Raspberry Pi to breadboard. It convert 3.3V electric to 5V electric level, then the Raspberry Pi can connect many 5V electric level peripheral.

IO extraction module

The module expand all of GPIO of Raspberry Pi to breadboard.

RTC expand module:

The RTC module is specifically designed for Raspberry Pi. There is a 0.2uF Super Capacitor on the board to keep the real time for a long time after the Raspberry Pi

Raspberry Pi Infinity cascade IO expand Module User Manual

has power off.

AD/DA expand module :

The AD/DA module is specifically designed for Raspberry Pi. There is 8bit high resolution DAC and ADC on one chip.

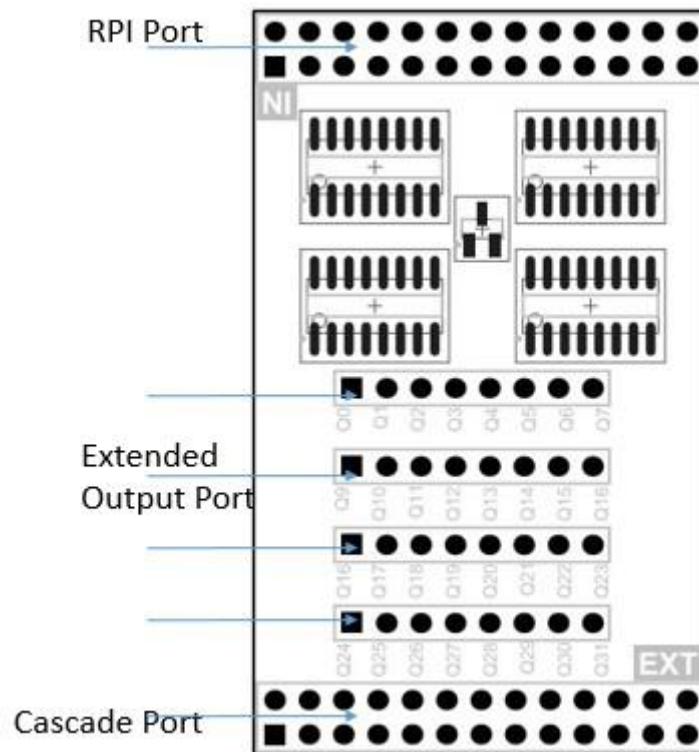
I2C electric level conversion expand module

The I2C electric level conversion module is specifically designed for Raspberry Pi. The module convert the 3.3V of I2C electric level to 5V level or convert 5V to 3.3V.

I2C electric level conversion expand module

The Serial Port module is specifically designed for Raspberry Pi. The module use Maxim MAX232 chip and DB9 port. So the user don't need to connect Dupont Line and that avoid wrong wiring. User can use this mod

Product Specification



Product Overview

This module is designed specifically for the Raspberry Pi IO expansion modules, which can effectively solve the Raspberry send IO port insufficient. Module uses four 74HC595 chips to expand 32 IO ports. As shown above, the top of the module will expand the IO of Raspberry Pi again that can be cascaded for more IO expansion modules can theoretically unlimited expansion.

Product Features

- Expanded 32 GPIO
- Infinity connection for the same module
- Use wiringPi API ,sample code

Port:

- Raspberry Pi 2X13 port
- Raspberry Pi 2X13 cascade port
- Q0-Q31 expanded IO port

Product Parameters

- Working voltage: 2.4V-5V
- IO voltage: 3.3V
- Expanded 32 unidirectional IO
- Connection through SPI
- 100 MHz (typical) shift out frequency
- 8-bit serial input
- 8-bit serial or parallel output
- Specified from -40C to +85C and from -40C to +125C

Typical Application

- Drive the lattice screen
- Driver numeric display
- Drive matrix LED

Method Of Use

Insert the module that the silk screen says “IN”. Pay attention don’t make the direction reversed! The correct direction of insert module is above the Raspberry Pi’s PCB; The 8*4header in the module are Q0-Q31 expanded GPIO, they can connect to the peripheral through the Dupont Line. The header which near silk screen write “EXT” expand GPIO of Raspberry Pi, user can cascade the other module or the same module.

More information

The 74HC595; 74HCT595 are high-speed Si-gate CMOS devices and are pin compatible with Low-power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard No. 7A.

The 74HC595; 74HCT595 are 8-stage serial shift registers with a storage register and

3-state outputs. The registers have separate clocks.

Data is shifted on the positive-going transitions of the shift register clock input (SHCP).

The data in each register is transferred to the storage register on a positive-going transition of the storage register clock input (STCP). If both clocks are connected together, the shift register will always be one clock pulse ahead of the storage register.

Raspberry Pi Infinity cascade IO expand Module User Manual

The shift register has a serial input (DS) and a serial standard output (Q7S) for cascading. It is also provided with asynchronous reset (active LOW) for all 8 shift register stages. The storage register has 8 parallel 3-state bus driver outputs. Data in the storage register appears at the output whenever the output enable input (OE) is LOW.

Function table:

Control				Input	Output		Function
SHCP	STCP	OE	MR	DS	Q7S	Qn	
X	X	L	L	X	L	NC	a LOW-level on MRonly affects the shift registers
X	↑	L	L	X	L	L	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
↑	X	L	H	H	Q6S	NC	logic HIGH-level shifted into shift register stage 0. Contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q6S) appears on the serial output (Q7S).
X	↑	L	H	X	NC	QnS	contents of shift register stages (internal QnS) are transferred to the storage register and parallel output stages
↑	↑	L	H	X	Q6S	QnS	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages

H = HIGH voltage state;

L = LOW voltage state;

↑ = LOW-to-HIGH transition;

X = don't care;

NC = no change;

Z = high-impedance OFF-state.

Timing diagram

Raspberry Pi Infinity cascade IO expand Module User Manual

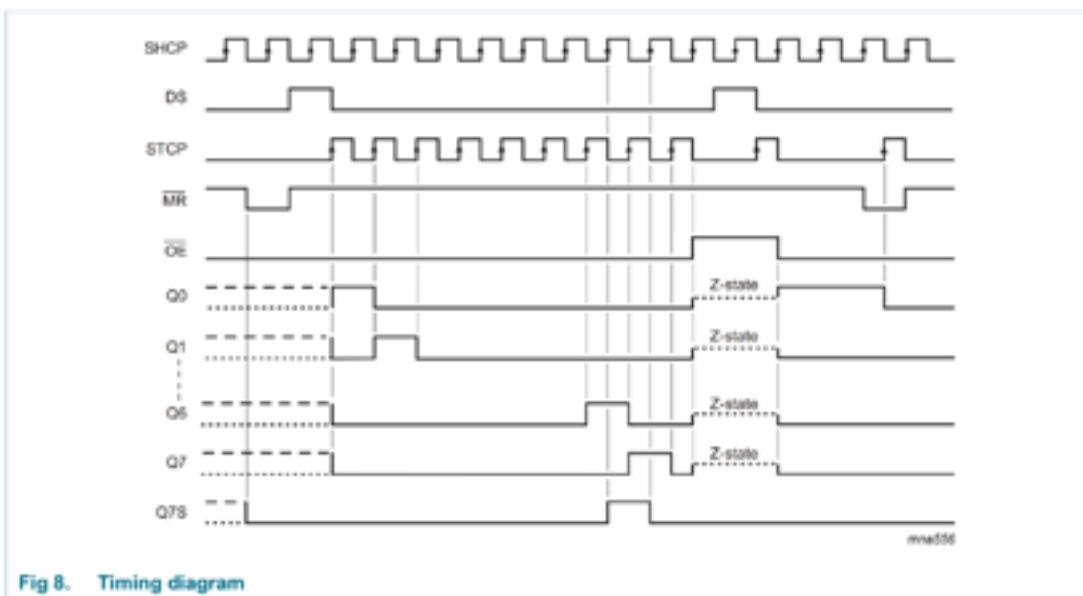
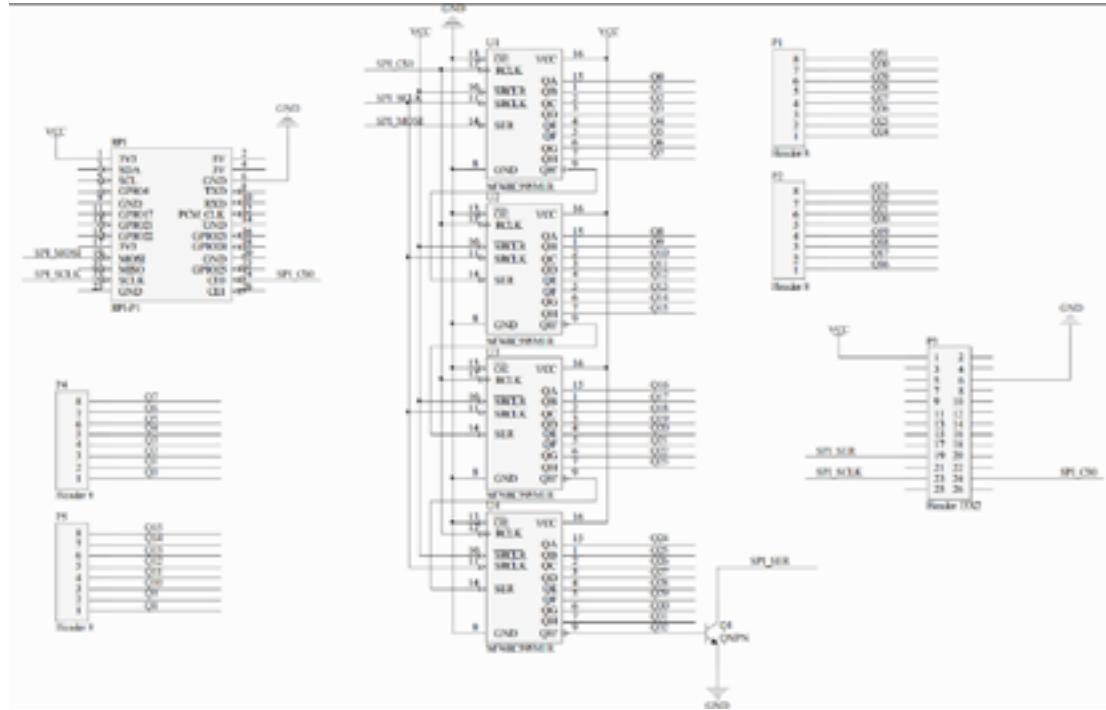


Fig 8. Timing diagram

Schematic diagram



More information please check

http://www.nxp.com/products/logic/shift_registers/series/74HC_T_595.html

Raspberry Pi Infinity cascade IO expand Module User Manual

Example and test code:

```
#include <wiringPi.h>
#include <stdio.h>
int SER    = 12;
int RCLK   = 10;
int SRCLK  = 14;
unsigned char LED [8]={ 0x01 , 0x02 , 0x04 , 0x08 , 0x10 , 0x20 , 0x40 , 0x80 };
void SIPO (unsigned char byte );
void pulse (int pin);
void init () {
    pinMode (SER , OUTPUT );
    pinMode (RCLK , OUTPUT );
    pinMode (SRCLK , OUTPUT );
    digitalWrite (SER , 0);
    digitalWrite (SRCLK , 0);
    digitalWrite (RCLK , 0);
}
void delayMS (int x) {
    usleep (x * 1000 );
}
int main (void )
{
    if (-1 == wiringPiSetup () ) {
        printf ("Setup wiringPi failed!" );
        return 1;
    }
    init ();
    int i;
    while (1) {
        for (i = 0; i < 8; i++)
        {
            SIPO (LED [i]);
            pulse (RCLK );
            delayMS (50);
            printf (" i = %d" , i);
        }
        printf ("\n");
        delayMS (500 ); // 500 ms

        for (i = 7; i >= 0; i--)
        {
            SIPO (LED [i]);
        }
    }
}
```

Raspberry Pi Infinity cascade IO expand Module User Manual

```
pu lse (RCLK );
delayMS (50);
printf (" i = %d" , i);
}
delayMS (500); // 500 ms
}
usleep (1000);
digitalWrite (RCLK , 1);
}

void SIPO (unsigned char byte )
{
    int i;
    for (i=0;i<8;i++)
    {
        digitalWrite (SER , (( byte & (0x80 >> i)) > 0));
        pulse (SRCLK );
    }
}
void pulse (int pin )
{
    digitalWrite (pin , 1);
    digitalWrite (pin , 0);
}
```