CKB0003 Infrared receiving module

1、Overview

CKB0003 infrared receiving module is mainly composed of infrared receiving head. It is a device integrating receiving, amplifying and demodulating. Its internal IC has already been demodulated, and the output is digital signal. It can be used in audio-visual equipment, home appliances, etc.

2、Specification

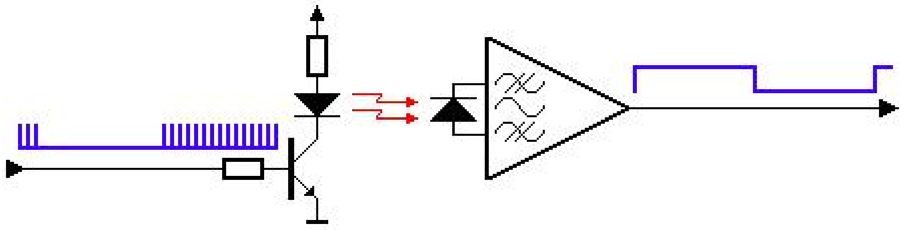
Working voltage: 2.7-5.5V (DC)
Interface: XH2.54-3P
Output signal: digital signal
Working current: 0.8mA
Carrier frequency: 38KHz
Receiving angle: ±35 degrees
Receiving distance: 5-10M

3、Interface

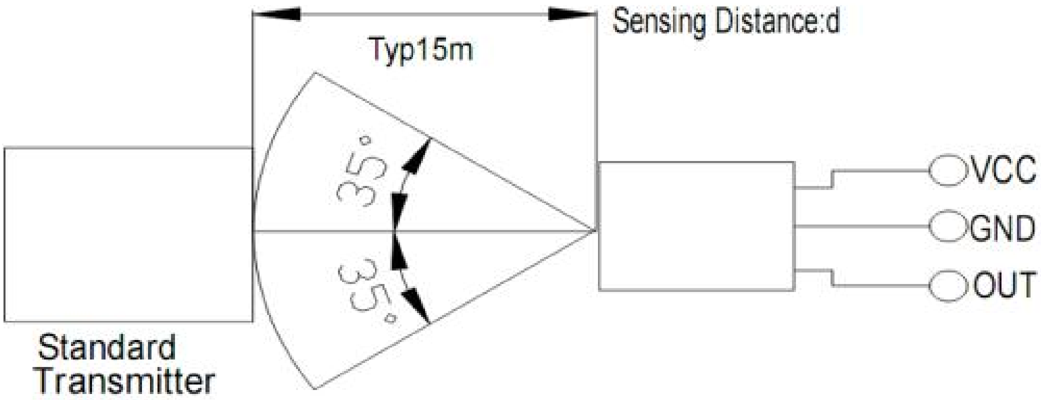
GND: Connect the negative pole of the power supply
VCC: Connect the positive pole of the power supply
S: digital signal output

4、the working principle of the infrared receiver

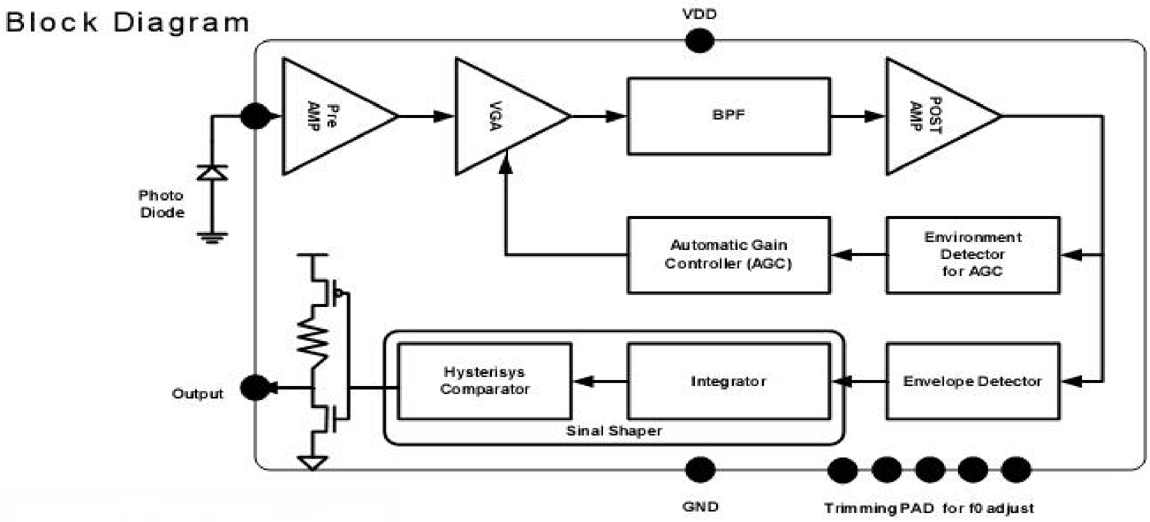
The visible light that can be seen by the human eye is arranged in wavelength from long to short, followed by red, orange, yellow, green, cyan, blue, and purple. The wavelength range of red light is 0.62 to 0.76 μm; the wavelength range of violet light is 0.38 to 0.46 μm. Light that is shorter than the wavelength of violet light is called ultraviolet light, and light that is longer than the wavelength of red light is called infrared light. Infrared remote control uses a near infrared ray with a wavelength between 0.76 and 1.5 μm to transmit control signals.
The infrared remote control is a control method for transmitting information by using infrared rays. The infrared remote control has the advantages of anti-interference, simple circuit, easy encoding and decoding, low power consumption and low cost. Infrared remote control is suitable for almost all home appliances.
The main part of the infrared remote control system is modulation, transmission and reception, as shown in the figure:



5、Receiving angle



6、Internal principle



7、Wiring diagram

UNO R3	红外接收模块
GND	
VCC	
11	

8、Test Code:

```
#include <IRremote.h>
```

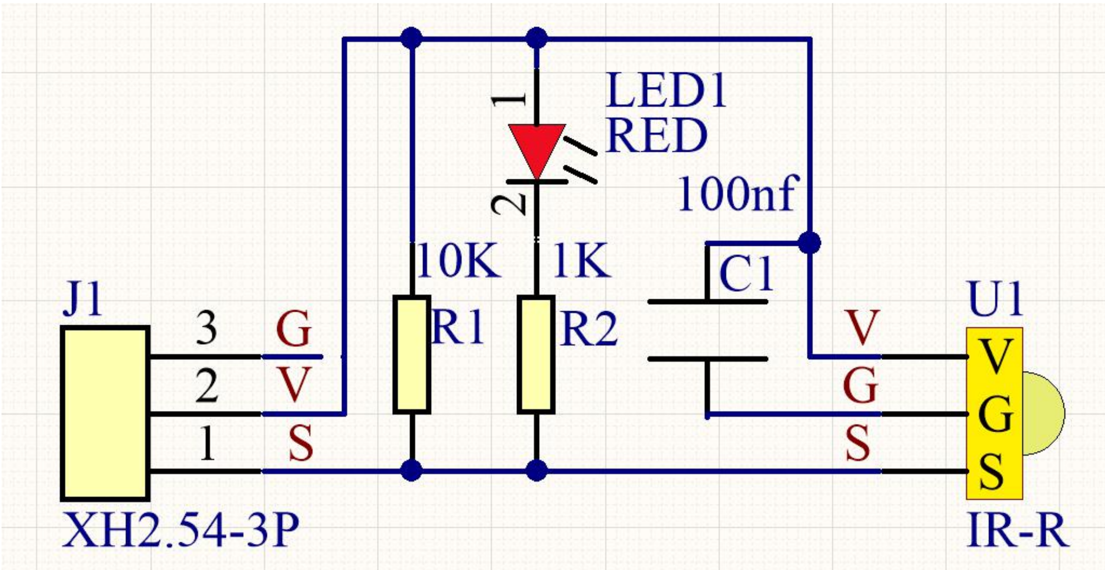
```
int RECV_PIN = 11; //Define digital port 11
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
 {
 Serial.begin(9600);//Set baud rate
 irrecv.enableIRIn(); // Enable infrared reception
 }
void loop()
{
if (irrecv.decode(&results))
{
Serial.println(results.value, HEX); //Display Data
irrecv.resume();
}
}
```

Before uploading the test code, you need to put the library file (ie IRremote folder) in the libraries folder of the Arduino IDE, as shown below.

9、Test Result

Connect the lines according to the above picture, upload the code, open the serial port monitor, send the signal to the infrared receiving module with the infrared transmitting remote control, the serial port monitor will output the received key value, enter the following figure:

10、原理图



11、NEC Infrared Transmission Protocol

The NEC IR transmission protocol uses pulse distance encoding of the message bits. Each pulse burst (mark – RC transmitter ON) is 562.5µs in length, at a carrier frequency of 38kHz (26.3µs). Logical bits are transmitted as follows:

①Logical '0' – a 562.5µs pulse burst followed by a 562.5µs space, with a total transmit time of 1.125ms

②Logical '1' – a 562.5µs pulse burst followed by a 1.6875ms space, with a total transmit time of 2.25ms

When transmitting or receiving remote control codes using the NEC IR transmission protocol, the WB_IRRC performs optimally when the carrier frequency (used for modulation/demodulation) is set to 38.222kHz.

When a key is pressed on the remote controller, the message transmitted consists of the following, in order:

. a 9ms leading pulse burst (16 times the pulse burst length used for a logical data bit)

. a 4.5ms space

. the 8-bit address for the receiving device

. the 8-bit logical inverse of the address

. the 8-bit command

. the 8-bit logical inverse of the command

a final 562.5µs pulse burst to signify the end of message transmission.

The four bytes of data bits are each sent least significant bit first. Figure 1 illustrates the format of an NEC IR transmission frame, for an address of 00h (00000000b) and a command of ADh (10101101b).

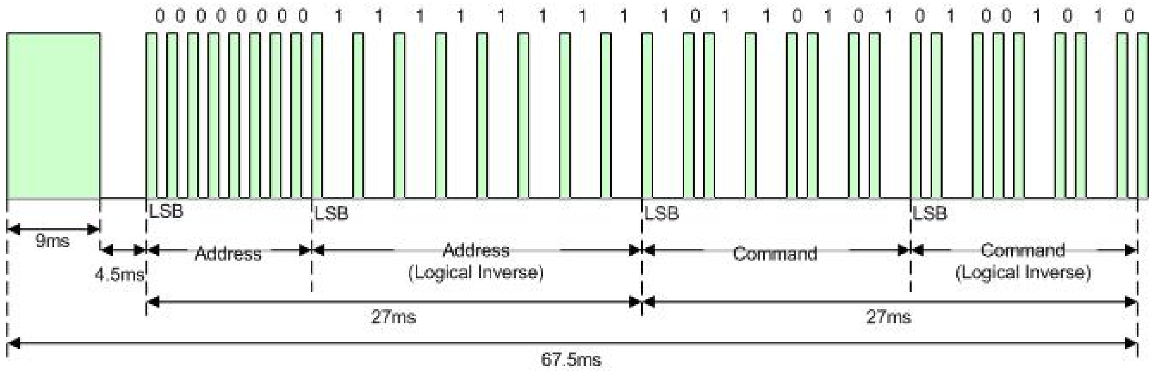


Figure 1. Example message frame using the NEC IR transmission protocol.

Notice from Figure 1 that it takes:

. 27ms to transmit both the 16 bits for the address (address + inverse) and the 16 bits for the command (command + inverse). This comes from each of the 16 bit blocks ultimately containing eight '0's and eight '1's - giving (8 * 1.125ms) + (8 * 2.25ms).

. 67.5ms to fully transmit the message frame (discounting the final 562.5µs pulse burst that signifies the end of message).

REPEAT CODES

If the key on the remote controller is kept depressed, a repeat code will be issued, typically around 40ms after the pulse burst that signified the end of the message. A repeat code will continue to be sent out at 108ms intervals, until the key is finally released. The repeat code consists of the following, in order:

. a 9ms leading pulse burst

. a 2.25ms space

. a 562.5µs pulse burst to mark the end of the space (and hence end of the transmitted repeat code).

Figure 2 illustrates the transmission of two repeat codes after an initial message frame is sent.

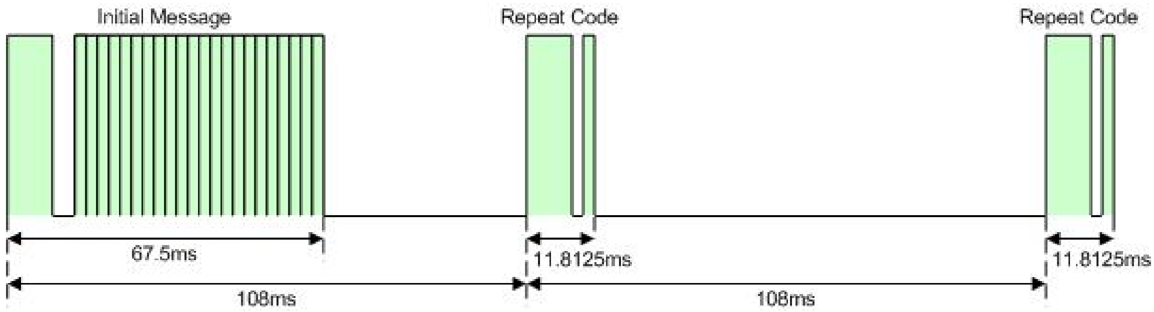


Figure 2. Example repeat codes sent for a key held down on the transmitting remote controller.

https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol