

Remote control shaking fan experiment

Introduction of infrared receiving tube

Infrared receiver tube, it is a kind of sensor that can recognize infrared ray. Integrated infrared sensor receives and modulates 38kHz infrared. In order to avoid the interference of other infrared signals in the wireless transmission process, the infrared remote control usually modulates the signal on a specific carrier frequency, and then emitted by the infrared emitting diode. When the infrared receiving device needs to filter out other clutter, it receives the signal of a specific frequency and recovers it as binary pulse code, that is, demodulation.

Working Principle

The built-in receiving tube converts the optical signal sent by the infrared transmitter tube into weak current signal, which is amplified by internal IC, and then restored to the original code sent by the infrared remote control through automatic gain control, bandpass filtering, demodulation, waveform shaping, and the decoding circuit input to the electrical appliance through the output pin of the receiving head

Experiment Purpose

- 1 The keys of the remoter are encoded by Arduino
- 2 The Arduino UNO main control board communicates with the infrared receiver. If the "*" button of the remote control is pressed, the fan starts to turn, and the "#" button is pressed, the fan stops to turn. Since the decoding process is based on a given protocol and the code is large, we use open source libraries to decode and copy IRremote to arduino1.6.5-r5 \ libraries.
- 3 link: <https://github.com/shirriff/Arduino-IRremote> 到 arduino1.6.5-r5 \ libraries

The experimental principle

If we want to decode it, we need to learn the remote's encoding first. The control code used in this product is: NEC protocol.

The following is an introduction to the NEC protocol:

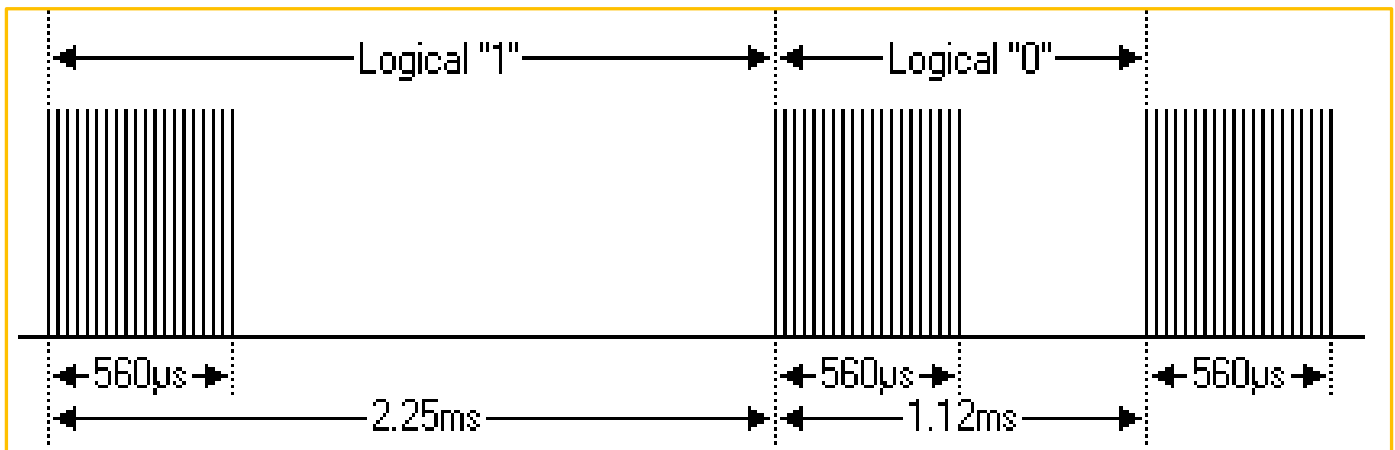
Introduction of NEC

Characteristics

- 8 address bits, 8 command bits
- Address bits and command bits are transmitted twice in order to ensure reliability
- Pulse-position modulation

- Carrier frequency 38kHz
- Every bit lasts 1.125ms or 2.25ms

The definitions of logic 0 and 1 are as below

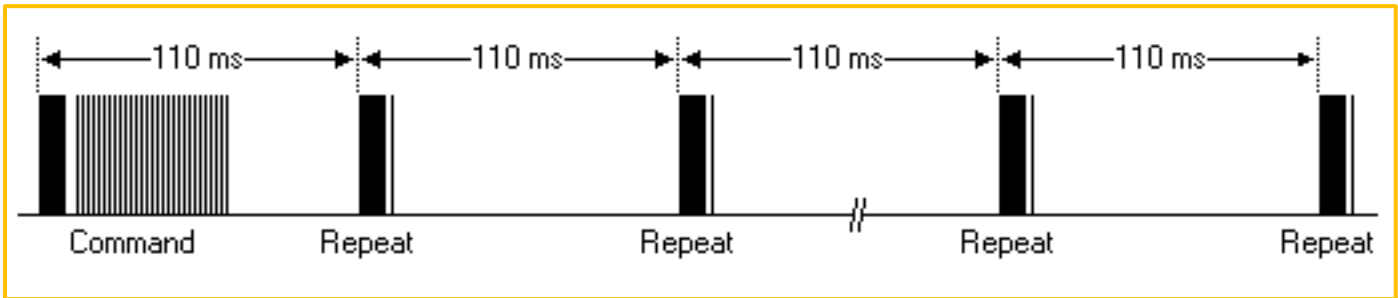


Transmitted pulse which the pressed button released immediately



The picture above shows a typical pulse sequence of the NEC protocol. Notice: The protocol of LSB (least significant) is firstly transmitted. In the above, pulse transmission address is 0x16 and the command is 0x59. A message starts from a 9ms high level, then followed by a 4.5ms low level, and by the address code and command code. The address and command are transmitted twice. All bits flip in the second transmission, this can be used in the confirmation of the received message. The total transmission time is constant, because every bit repeats the flip length. If you are not interested, you can ignore this reliable inversion and expand the address and command every 16 bit as well !

Transmitted pulse which the pressed button last for a while



Even a button on the remote control is pressed again, the command is transmitted only once. When the button has been pressing, the first 110ms pulse is same as above, then the same code is transmitted every 110ms. The next repeated code consists of a 9ms high level pulse, a 2.25ms low level pulse and a 560μs high level pulse.

Notice: When the pulse received by the integrative header, the header needs to decode, amplify and shape the signal. So we should notice that the output is high level when the infrared signal is absent, otherwise, the output is low level, so the output signal level is reversed in transmitter. We can see the pulse of receiver through the oscilloscope and understand the program by waveform.

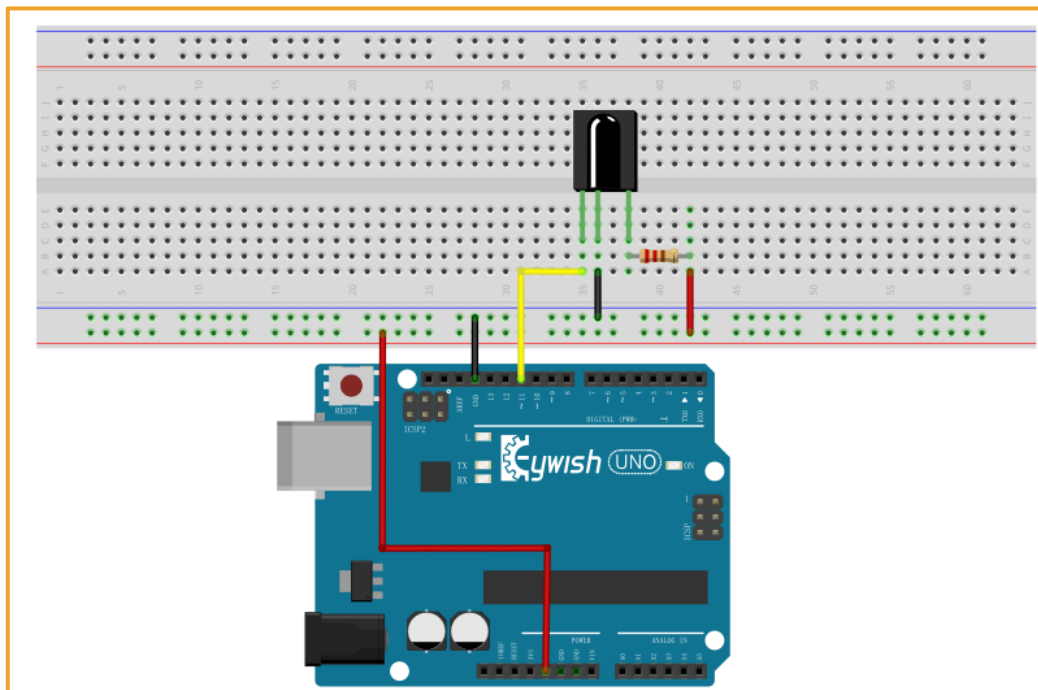
Component List

- Keywish Arduino UNO R3 motherboard
- Breadboard
- USB cable
- Infrared remote control * 1
- Integrated infrared receiving head module * 1
- Dc motor *1
- Fan * 1
- Steering gear * 1
- Motor drive board *1
- Motor bracket kit *1
- Battery * 1
- A number of jumpers

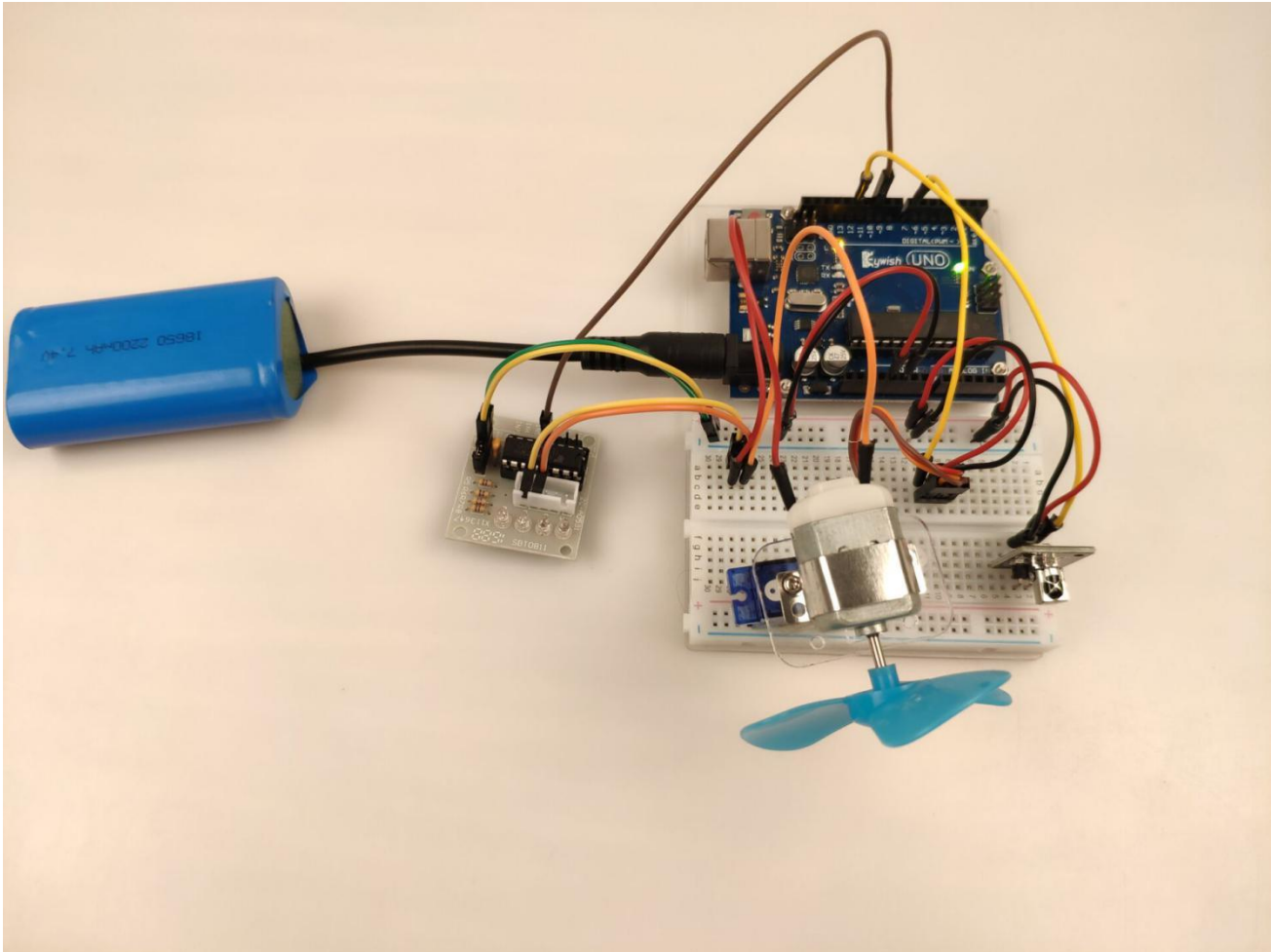
Wiring of Circuit

| Motor drive board | Steering gear module |
|-------------------|----------------------|
| 5V | VCC |
| GND | GNG |

| | |
|-------------------|-------------------------|
| 7 | S |
| Arduino mianboard | Infrared remote control |
| 5V | + |
| GND | - |
| 12 | S |
| Arduino mianboard | Motor drive board |
| 5V | 5V(+) |
| GND | GND(-) |
| 10 | IN4 |
| Motor drive board | Dc motor |
| VCC | + |
| OUT4 | - |



Wiring of objects



Arduino IDE Code

```
#include "IR_remote.h"
#include <Servo.h>
Servo myservo;
int servopin = 7;
int MotorPin = 10;
int flag = 0;

IRremote ir(12);
unsigned char keycode;

void setup()
{
    Serial.begin(9600);
    ir.begin();
    myservo.attach(7);
    myservo.write(90);
    pinMode(MotorPin, OUTPUT);
    digitalWrite(MotorPin, 0);
}

void loop()
{
    byte ir_key = ir.getCode();
    //Serial.println(ir_key);
    switch (ir.getIrKey(ir_key)) {
        case IR_KEYCODE_OK:
            Serial.println("IR_KEYCODE_OK key");
            flag = !flag;
            digitalWrite(MotorPin, flag);
            break;
        case IR_KEYCODE_LEFT:
            myservo.write(0);
            break;
        case IR_KEYCODE_RIGHT:
            myservo.write(180);
            break;
    }
    delay(110);
}
```

Experiment Result

