

## Hindernis Sensor Modul Datenblatt



**Contents:**

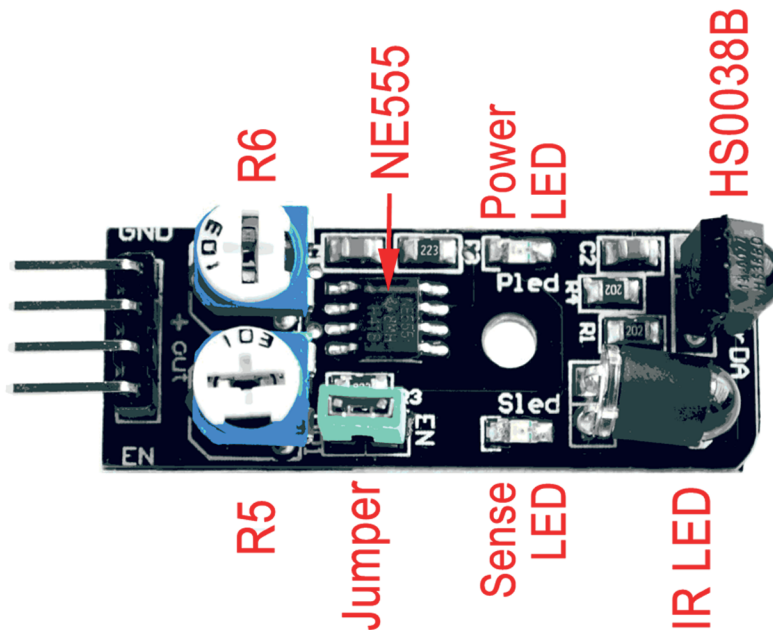
- 1. Introduction**
- 2. Manufacturer's Specifications**
- 3. Description**
- 4. Sample C++ code**
- 5. Connection Diagrams**

## 1. Introduction

This sensor is known variously as the Keyes, KeyesIR or Keyestudio KY-032. It is the functional equivalent of the IrBeady IR-08H, also known as the AD-032. The sensor uses a four pin connector, The pins are labeled: EN (Enable), out (Output), + (Power) and GND (Ground). There are also two potentiometers on the board and one jumper.

## 2. Manufacturer's Specifications

- Working voltage: 3.3V to 5V DC
- Working current:  $\geq 20\text{mA}$
- Operating temperature:  $-10^{\circ}\text{C}$  to  $+50^{\circ}\text{C}$
- Detection distance: 2 to 40cm
- IO Interface: 4-pin (EN / +V / S / GND)
- Output signal: TTL level
- LOW level if obstacle detected
- HIGH if no obstacle detected
- Adjustment: two single-turn variable resistors
- Effective angle:  $\pm 35^{\circ}$
- Size: 28mm \* 23mm
- Weight: 9g



### 3. Description

At the heart of the sensor is an NE555 chip configured to generate a 38kHz square wave. (The IR-08H uses an SN74LS00.) The 38kHz signal is used to illuminate an Infra Red (IR) LED. Light reflected from the LED is detected by a Vishay HS0038B IR receiver module (see data sheet). The receiver module incorporates an external, optical, 950nm IR filter and an internal, electronic, 38kHz band-pass filter that make the module receptive only to IR light pulsing at that frequency.

One of the potentiometers on the board (R6) is used to fine tune the signal to exactly 38kHz. The other (R5) adjusts the duty cycle of the signal, which controls the brightness of the IR LED.

Both adjustments together effect the sensitivity or range of the device. Proper adjustment of R6 requires an oscilloscope or a frequency counter. Otherwise it can be left centered or as it came from the manufacturer.

Both adjustments together effect the sensitivity or range of the device. Proper adjustment of R6 requires an oscilloscope or a frequency counter. Otherwise it can be left centered or as it came from the manufacturer.

On most versions of this device, the Infra Red (IR) LED is already covered with a small piece of black shrink tubing; but I find that additional optical shielding is required. A small cardboard tube commonly used as packing material will work satisfactorily, as will a variety of other materials.

When the GREEN JUMPER is installed on the board (see picture), the IR LED will flicker continuously at 38kHz. If the Enable (EN) function will not be used, the jumper must be installed. When the jumper is removed, pin 4 of the 555 timer is held LOW (RESET) by R3, a 22K pull-down resistor. Then, if a HIGH condition is applied to the EN pin, the reset condition will be relieved and the 555 timer will begin to oscillate. The Enable function cannot be used if the GREEN JUMPER is in place. You must remove the jumper on the board in order to use the EN (Enable) pin.

## 4. Sample C++ Code

```
digitalWrite( enablePin, HIGH); // Enable the internal 38kHz signal.
microDelay( 210); // Wait 210µs (8 pulses of 38kHz).
if( digitalRead( outputPin)) // If detector Output is HIGH,
{
    objectDetect = false; // then no object was detected;
}
else // but if the Output is LOW,
{
    microDelay( 395); // wait for another 15 pulses.
    if( digitalRead( outputPin)) // If the Output is now HIGH,
    { // then first Read was noise
        objectDetect = false; // and no object was detected;
    }
    else // but if the Output is still LOW,
    {
        objectDetect = true; // then an object was truly detected.
    }
}
digitalWrite( enablePin, LOW); // Disable the internal 38kHz signal.
```

