QUICK REFERENCE · TK61

Infrared Speed Sensor

A beam across a 5 mm slot. Something in it reads HIGH; count the pulses for rpm.

VCC is your board's logic voltage

SIGNAL is pulled up to VCC by 10 kΩ, so HIGH is whatever VCC is. **On an ESP32, ESP32-S3 or Pico wire VCC to 3V3**: 5 V there lifts a 3.3 V pin past its supply every time something passes the slot. Only an Uno takes 5V.

Pins, and which to wire

Parts up, header at the bottom. Wire GND, VCC and SIGNAL; NC is connected to nothing on the board. Read SIGNAL with INPUT.

GK105A slot sensor
5 mm slot, 8 mm deep. Card or metal, not paper.

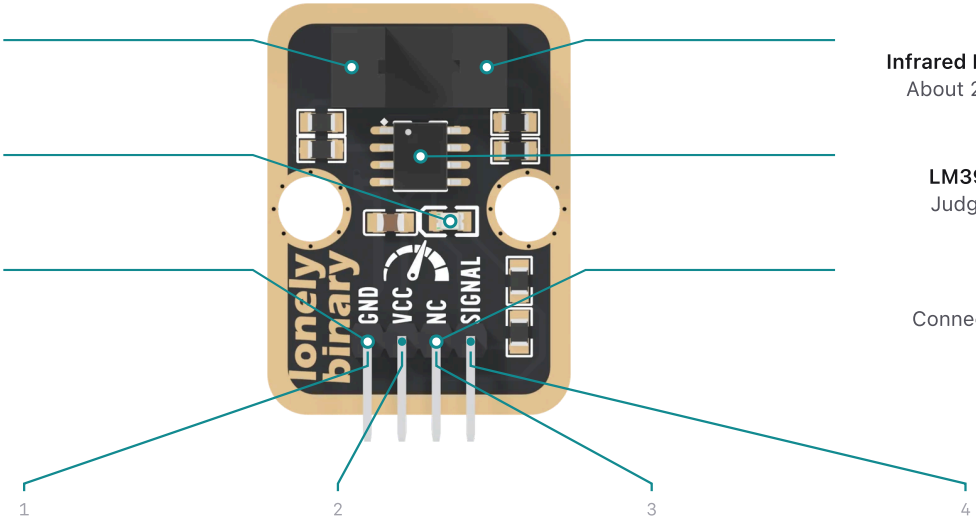
Red LED: lit is clear
Dark while blocked. Needs no sketch.

GND is the square pad
Count from it. The back prints no pin names.

Infrared LED, always on
About 2.2 mA at 3.3 V, blocked or not.

LM393 comparator
Judges at half VCC. No hysteresis.

NC
Connected to nothing on the board.



Front	GND	VCC	NC	SIGNAL
Read it	GND	VCC	●	SIGNAL

HIGH means blocked With the slot clear the comparator pulls SIGNAL LOW and the red LED lights. Something in the slot lets SIGNAL rise to VCC and puts the LED out. Count rising edges: one per bar of a slotted disc.

How fast it can count

Sensor rising, beam blocked

100 µs, typical

The light sensor on its 10 kΩ, from the GK105A sheet at 20 mA.

Sensor falling, beam clear

130 µs, typical

The slower edge. The comparator takes 1.3 µs; your board's interrupt, a few.

Fastest count trusted

1,900 pulses a second

About 5,800 rpm on a 20-slot disc. Fewer openings, higher rpm.

The light sensor sets the limit, not the sketch. The book allows four of the slower edge per pulse; past about 1,900 pulses a second the block starts to miss bars.

Edges from SIVAGO's GK105A sheet, typical, at 20 mA through its LED; this board gives it about 2.2 mA at 3.3 V, which makes a phototransistor slower, so the limit is a ceiling. Currents worked out, not measured: about 3 to 5 mA at 3.3 V. The sketch compiles for an ESP32-S3 and an Uno; the block has not been bench-tested for this sheet.

A tachometer, on an ESP32-S3

Arduino IDE · Board: ESP32S3 Dev Module · USB CDC On Boot: Enabled · No library

```
// The pin SIGNAL is wired to.
// Uno: 2. ESP32: 25. ESP32-S3: 4. Pico: 15.
const int SENSOR_PIN = 4;
const int SLOTS = 20;                  // openings in the disc
const unsigned long LOCKOUT_US = 500;  // ignore closer edges

#ifndef IRAM_ATTR
#define IRAM_ATTR
#endif

volatile unsigned long pulses = 0;
volatile unsigned long lastEdgeUs = 0;

void IRAM_ATTR onEdge() {  // a bar has entered the beam
  unsigned long at = micros();
  if (at - lastEdgeUs >= LOCKOUT_US) {
    pulses++;
    lastEdgeUs = at;
  }
}

void setup() {
  Serial.begin(115200);
  pinMode(SENSOR_PIN, INPUT);
  attachInterrupt(digitalPinToInterrupt(SENSOR_PIN),
                  onEdge, RISING);
}

void loop() {
  static unsigned long start = millis();
  if (millis() - start < 1000) return;  // count for one second
  start += 1000;

  noInterrupts();
  unsigned long n = pulses;
  pulses = 0;
  interrupts();

  Serial.print(n);
  Serial.print(" pulses   ");
  Serial.print(n * 60.0 / SLOTS, 0);
  Serial.println(" rpm");
}
```

```
63 pulses 189 rpm
64 pulses 192 rpm
```

Serial Monitor at 115200, once a second, a 20-slot disc turning. The reading steps by 3 rpm.

Wiring

	FRONT	S3	UNO	ESP32	PICO
1	GND	GND	GND	GND	GND
2	VCC	3V3	5V	3V3	3V3
3	NC				
4	SIGNAL	GPIO 4	D2	GPIO 25	GP15

Hole 1 is the square pad. VCC is each board's own logic voltage: 3V3 on the ESP32 boards and the Pico, 5V on the Uno. On an Uno only D2 and D3 take an interrupt.

Slots, rpm and the edges

rpm = pulses a second × 60 / openings. Set SLOTS to the openings on your disc; bars and openings must each be wider than the sensor's window, about 0.7 mm, where they cross the slot. The comparator judges at half VCC (1.65 V from 3.3 V) with no hysteresis, so a slow edge can flip SIGNAL twice: the sketch ignores a second edge within 500 µs.

When it does not work

LED dark, slot empty

Check VCC and GND from the square pad, then look for dust or a label over a window.

LED follows, count does not

SIGNAL is on another pin, or on NC. Check SENSOR_PIN.

Counts too few when fast

Count on an interrupt, not a polling loop. Past the limit, use fewer openings.

Never reads HIGH

Paper and many plastics pass infrared. Use card or metal, pushed to the bottom.

Counts too many when slow

A slow edge flips it twice. Raise LOCKOUT_US, under the gap between bars.

Misreads in sunlight

Sunlight carries infrared and the slot is open. Shade it.

Specifications

Sensor	GK105A slotted optical switch	**Slot**	5 mm wide, 8 mm deep
SIGNAL	HIGH blocked, LOW clear, INPUT	**Output**	LM393, 10 kΩ pull-up to VCC
Threshold	Half of VCC, fixed, no hysteresis	**Edges**	100 µs up, 130 µs down, typical
Fastest	About 1,900 pulses a second	**Supply**	Your board's logic voltage