

MIC1555YM5 Breakout Module

CHIP: MIC1555YM5-TR**SIZE:** 16mm x 15.1mm

DESCRIPTION:

The Meridian MIC1555 breakout brings a compact CMOS timer onto five 0.1-inch-spaced header connections, making it easier to build timing circuits on a breadboard without working directly with the tiny SOT-23 package.

Use it to generate a single timed pulse in monostable mode, or a repeating square wave in astable mode. External resistors and capacitors set the timing, so you can choose the pulse duration or oscillator frequency to suit your project. No microcontroller or code is required to generate the waveform.

The board includes 100 nF and 10 μ F supply-decoupling capacitors and a power indicator LED. The external timing resistor and capacitor are not included; add these using the example circuit or your own calculated values.

The MIC1555 is similar in function to a classic 555 timer, but it is not a pin-compatible replacement. It has separate trigger and threshold inputs and no dedicated discharge pin. Follow the MIC1555 wiring diagrams in this guide rather than a standard eight-pin 555 circuit.

Start with the oscillator example to build a simple square-wave generator. The optional Arduino sketch measures the output—it does not control the timer.

Happy building!

BOARD PINOUT:

PIN	NAME	DOES THIS
1	VS	Supply Voltage (Chip rated to 2.7-18V, board only rated 2.7-5.5V)
2	GND	Ground
3	OUT	Timer Output
4	TRG	Trigger input
5	THR	Threshold Input

- No discharge pin: Unlike a conventional 555, the MIC1555 does not have a dedicated discharge-output pin.
- Astable operation: Tie TRG and THR together, then connect the joined node to the timing capacitor.
- Monostable operation: Keep TRG high except for a brief low-going trigger pulse; use THR for the timing-capacitor voltage.

- Signal levels: The trigger and threshold comparisons are nominally $1/3V_S$ and $2/3V_S$, respectively.

TIMING REFERENCE:

MODE	CONFIGURATION	FORMULA	NOTES
One-shot (monostable)	Connect RT from OUT to THR; connect CT from THR to GND	$t = \sim 1.1(RT)(CT)$	A low-going pulse at TRG starts the output pulse. The output returns LOW when THR reaches approximately $2/3V_S$. Keep TRG high after the trigger pulse.
Oscillator (astable), approximately 50% duty cycle	Tie TRG and THR together. Connect RT from OUT to the joined TRG/THR node; connect CT from that node to GND	$f = \sim 0.7 / (RT * CT)$, below approximately 1 MHz	The waveform is approximately 50% duty cycle. The first cycle after power-up is longer than later cycles.

R in ohms; C in farads; t in seconds; f in hertz.

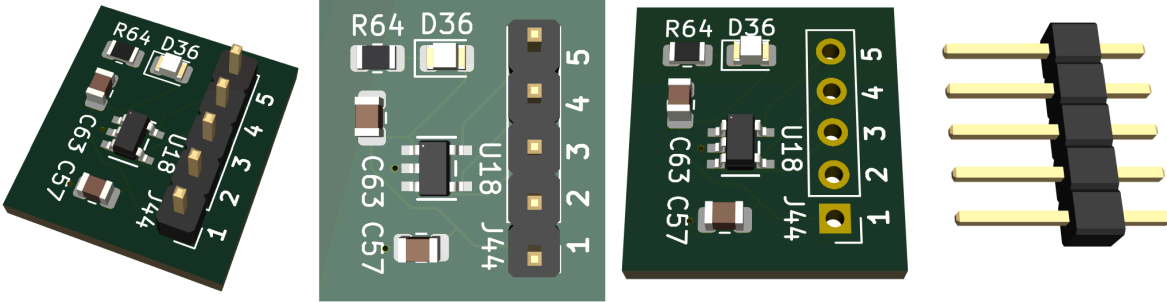
APPLICATIONS:

- One-shot pulse stretching and fixed-duration delays.
- Low-frequency clocks and square-wave generation.
- LED blinkers and status indicators.
- 3.3 V or 5 V breadboard timing experiments.
- Timing and oscillator demonstrations for electronics education.
- Adjustable-frequency oscillator projects with external timing components.

CHIPS ON BOARD:

COMPONENT	PART #	JOB	DATASHEET
CMOS RC Timer	MIC1555YM5-TR	Monostable & Astable Timing	[Manufacturer Datasheet]

PHOTOS:

**WIRING:**

CONNECTION	DESTINATION
Module VDD / VS, pin 1	Arduino 5V
Module GND, pin 2	Arduino GND
Module OUT, pin 3	Arduino digital pin 2
Module TRG, pin 4	Module THR, pin 5
100 k Ω resistor	Between OUT and the joined TRG/THR node
100 nF timing capacitor	Between the joined TRG/THR node and GND

EXAMPLE CODE:

```

/*
  Meridian Electronics
  MIC1555 oscillator measurement example

  Target board: Arduino UNO R3

  External timing circuit:
  - Join MIC1555 TRG and THR.
  - Connect 100 kOhm from OUT to the TRG/THR junction.
  - Connect 100 nF from that junction to GND.

  Arduino connections:
  - MIC1555 VDD / VS -> 5V
  - MIC1555 GND      -> GND
  - MIC1555 OUT      -> digital pin 2

  The MIC1555 generates the waveform.
  The Arduino measures HIGH and LOW pulse durations.
*/

const byte SIGNAL_PIN = 2;

```

```

const unsigned long PULSE_TIMEOUT_US = 100000UL;

void setup() {
  pinMode(SIGNAL_PIN, INPUT);
  Serial.begin(9600);

  Serial.println(F("MIC1555 oscillator measurement"));
  Serial.println(F("Timing parts: 100 kOhm and 100 nF"));
  Serial.println(F("Expected: roughly 70 Hz, near 50% duty"));
  Serial.println(F("A timeout means a complete pulse was not measured."));
  Serial.println();

  delay(500);
}

void loop() {
  const unsigned long highUs =
    pulseIn(SIGNAL_PIN, HIGH, PULSE_TIMEOUT_US);

  if (highUs == 0) {
    Serial.println(F("TIMEOUT: no complete HIGH pulse."));
    Serial.println(F("Check power, ground, OUT-to-D2, and RC wiring."));
    delay(1000);
    return;
  }

  const unsigned long lowUs =
    pulseIn(SIGNAL_PIN, LOW, PULSE_TIMEOUT_US);

  if (lowUs == 0) {
    Serial.println(F("TIMEOUT: no complete LOW pulse."));
    Serial.println(F("Check TRG/THR connection and timing components."));
    delay(1000);
    return;
  }

  // These are separately sampled pulses, not necessarily one cycle.
  // Their sum estimates the period of a stable oscillator.
  const unsigned long periodUs = highUs + lowUs;
  const float frequencyHz = 1000000.0f / periodUs;
  const float dutyPercent = 100.0f * highUs / periodUs;

  Serial.print(F("Frequency: "));
  Serial.print(frequencyHz, 1);
  Serial.print(F(" Hz | HIGH duty: "));
  Serial.print(dutyPercent, 1);
  Serial.print(F("% | HIGH: "));
  Serial.print(highUs);
  Serial.print(F(" us | LOW: "));
  Serial.print(lowUs);
  Serial.println(F(" us"));

  delay(500);
}

```

NOTES & DEBUGGING:

- Check the pinout before applying power. The MIC1555 is not pin-compatible with a standard eight-pin 555 timer.
- THR is a threshold input, not a discharge output. Follow the MIC1555 example circuits rather than standard 555 wiring.
- For oscillator operation, join TRG and THR. Connect the timing resistor from OUT to this junction and the timing capacitor from the junction to GND.
- For one-shot operation, keep TRG HIGH between brief LOW trigger pulses. Return it HIGH before the timed output pulse ends.
- Timing formulas are approximate. Check resistor and capacitor values, keep timing connections short, and measure the result if accuracy matters.
- When connecting an Arduino or other controller, use a common ground and verify signal-voltage compatibility. The power LED does not confirm correct timer operation.
- Use a series resistor with an external LED, and a suitable driver for heavier loads. Do not treat OUT as a general-purpose power supply.

Feel free to send any questions to sales@meridianelectronic.com

Check MeridianElectronic.com for datasheets, 3D printing, solder stencils & more!