Ultrasonic Range Sensor Assessment

MX EN2003 | jack Searle 21502396

Sensor Functionality

“The Parallax PING)))™ ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, Propeller chip, or Arduino, requiring only one I/O pin. “ - *PING))) Ultrasonic Distance Sensor (#28015) Datasheet*

The sensor determines distance by emitting an ultrasonic burst and measuring the time until the echo is received. The time-of-flight is then converted into distance using the calculated speed of sound at the given operating temperature.

Calculations

The main equations used are:

Distance (mm) = Speed of sound (mm/ms) × Time (ms)
331.5 + (0.6 × T°C) ≈ 348 mm/ms at 21°C

The pingValue is in microseconds (µs), so it's first converted to milliseconds (ms):

distance = CONVERTING_CONSTANT * (pingValue * 1e-3);

The constant CONVERTING_CONSTANT = 348 was chosen as I’ve assumed we’re operating at room temp 0f 21°C.

How to Communicate

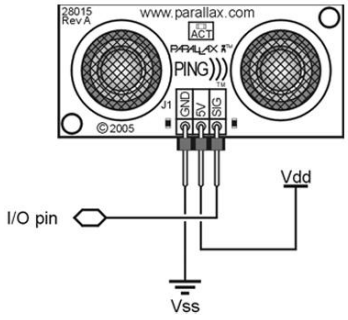
Communication with the sensor and user is done via:

- **Digital pin PD2 (INT0):** For sending trigger and receiving echo. Can be set to other pins, this is just the default pin in the code. Check ATmega2560 pinout for compatible pins.
- **USART0 Serial:** For displaying output (distance/error messages).
- The sensor is triggered with a 5 µs HIGH pulse (sensor allows 2 µs min) and then set to input mode. Echo duration is measured via timer.

Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged into a directly into solderless breadboard, or into a standard 3wire extension cable (Parallax part #800-00120).



Function Outlines

serial0_init(): Sets up USART0 for serial communication (8-bit, 9600 baud).
serial0_print_string(char*): Prints a string to the serial monitor.
ping_timer_init(): Configures Timer1 in CTC mode, with overflow interrupt enabled for time tracking.
ping_sensor_start(): Starts the ping process with a 200 µs delay, enabling INT0 for edge detection.
main(): Manages sensor readings, handles errors, calculates distance, and prints results.

Interrupts and Timers

- **Timer1** is configured with no prescaler, giving a 1 MHz clock (1 µs resolution).
- **CTC mode** (Clear Timer on Compare Match) is used with OCR1A for precise delays.
- **TIMER1_OVF_vect**: Adds 4000 µs on overflow. Triggers timeout error if limit reached.
- **TIMER1_COMPA_vect**: Controls timing stages of sensor operation (trigger pulse, holdoff, timeout).
- **INT0_vect**: Measures time from echo start to end. Triggers on rising and falling edge.

Logic Flow Charts