

URM37 V3.2 Ultrasonic Sensor Manual

Rev 2.0



2009-04-29

Introduction

URM37 V3.2 Ultrasonic Sensor uses an industrial level AVR processor as the main processing unit. It comes with a temperature correction which is very unique in its class.

Specification

Power: +5V

Current: <20mA

Working temperature: -10°C ~ +70°C

Detecting range: 4cm-5m

Resolution: 1cm

Interface: RS232 (TTL), PWM

Servo control: One servo control output

Operating Mode: Serial (PWM) passive control mode; Autonomous Mode; On/OFF Mode

Temperature sensor: 12 bits reading from serial port

Size: 22mm × 51 mm

Weight: 30g

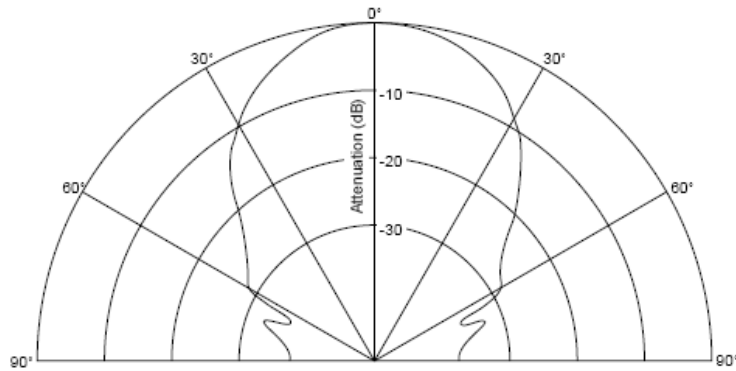


Figure 1: URM37 V3.2 Beam Width

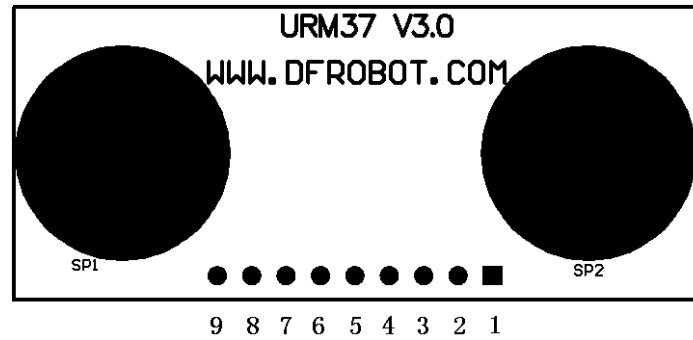


Figure 2: URM37 V3.2 Pin Definition

1: **+VCC** - +5V Power

2: **GND** - Ground

3: **RST** - Reset

4: **PWM** - PWM Output 0—25000US, Every 50US represent 1cm

5: **MOTO** - Servo control signal output

6: **COMP/TRIG**

COMP - On/OFF mode, when the detecting distance is smaller than a pre-set value, this pin pulls low.

TRIG - PWM or RS232 trigger pin

7: **PWR_ON** - Enable pin, enable the sensor when high

8: **RXD** - RS232,TTL communication

9: **TXD** - RS232,TTL communication

Working Mode Selection

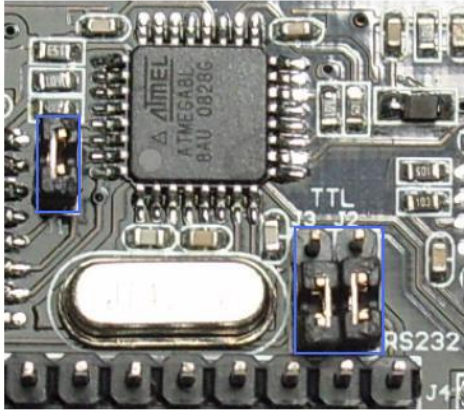
The working mode can be changed by writing 0x00, 0x01 or 0x02 to EEPROM through serial port.

Mode 1: Serial passive control mode

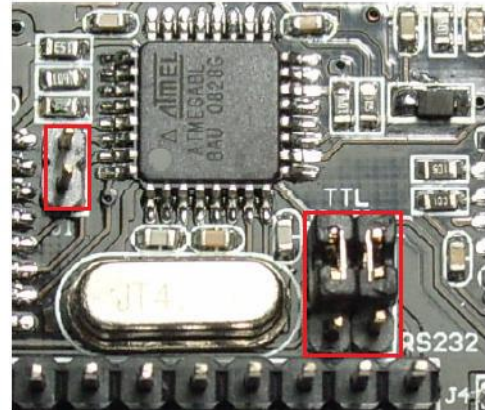
Under this mode, the sensor is always waiting for command from serial port. Every time it receives a command, it will return the distance and wait for the next command. The degree in the command will be used to control a servo motor to rotate corresponding degree.

Jumper setting for RS232 and TTL output

The selection of RS232 or TTL output level is switched by changing three jumpers (J1, J2, J3). A diagram below illustrates the setting:



RS232 Mode



TTL Mode

Warning: Do not connect to TTL MCU when the output mode is set to RS232, doing so will permanently damage the unit.

This feature is only available for Rev2 and after. If there are no jumpers on the back of the sensor, the sensor is Rev1 and hence not supporting this feature.

Mode 2: Autonomous trigger mode

Under this mode, the sensor will make a sensor reading every 25ms and compare the reading with a threshold (pre-set, user is able to define this value by writing EEPROM), if the reading is equal or smaller than the threshold, pin COMP/TRIG will have low output. In the meantime, pin PWM will output the distance reading, every 50us low level stands for 1cm, by counting the number of these pulses, the distance can be calculated. This mode can be simply used as an ON/OFF switch.

Mode 3: PWM passive control mode

Under this mode, a low pull on pin COMP/TRIG will trigger a sensor reading. The width of the pulse is proportional to the servo rotating degree. After a successful sensor reading, Pin PWM will output pulses, every 50us represents 1cm. If the reading is invalid, a 50000us pulse will be returned.

Serial control protocol

Serial setting: Port rate: 9600; Parity: none; Stop bit: 1

Command: Control command consists of four bits, command+data0+data1+sum. Sum=Low 8 bit of the sum of command+data0+data1.

| Command Format | Function | Description |
|----------------|---------------|--|
| 0x11+NC+NC+Sum | Enable 16 bit | Reading the temperature, the return data |

| | | |
|---|---------------------------------------|--|
| <p>(Sample: 0x11 0x00 0x00 0x11)</p> | <p>temperature reading</p> | <p>format will be: $0x11 + \text{High}(\text{temperature}) + \text{Low}(\text{temperature}) + \text{SUM}$</p> <p>If the temperature is above 0, the first four bits of High will be all 0. If the temperature is below 0, the first four bits of High will be all 1.</p> <p>The last 4 bits of High together with the Low bits stands for 12bits temperature. The resolution is 0.1. When the reading is invalid, it returns $0x11 + 0xFF + 0xFF + \text{SUM}$</p> |
| <p>$0x22 + \text{Degree} + \text{NC} + \text{SUM}$ (Sample: 0x22 0x00 0x00 0x22)</p> | <p>Enable 16 bit distance reading</p> | <p>The degree in the command is used to control a servo motor to rotate corresponding degree.</p> <p>Degree: 0-46 stands for 0-270 degrees, for example, 3 stands for 18 degrees.</p> <p>Return data format will be: $0x22 + \text{High}(\text{distance}) + \text{Low}(\text{distance}) + \text{SUM}$.</p> <p>When the reading is invalid, it returns $0x22 + 0xFF + 0xFF + \text{SUM}$</p> |
| <p>$0x33 + \text{Add} + \text{NC} + \text{SUM}$</p> | <p>Enable internal EEPROM reading</p> | <p>Return data will be $0x33 + \text{Add} + \text{Data} + \text{SUM}$.</p> |
| <p>$0x44 + \text{Add} + \text{Data} + \text{SUM}$</p> | <p>Enable internal EEPROM writing</p> | <p>Written data can only from 0-255. Address 0x00-0x02 is used to configure the mode. 0x00 - threshold distance (Low) 0x01 - threshold distance (High) 0x02 - Operation Mode (0xaa for autonomous mode) (0xbb for PWM passive control mode)</p> <p>The return data format will be:</p> |

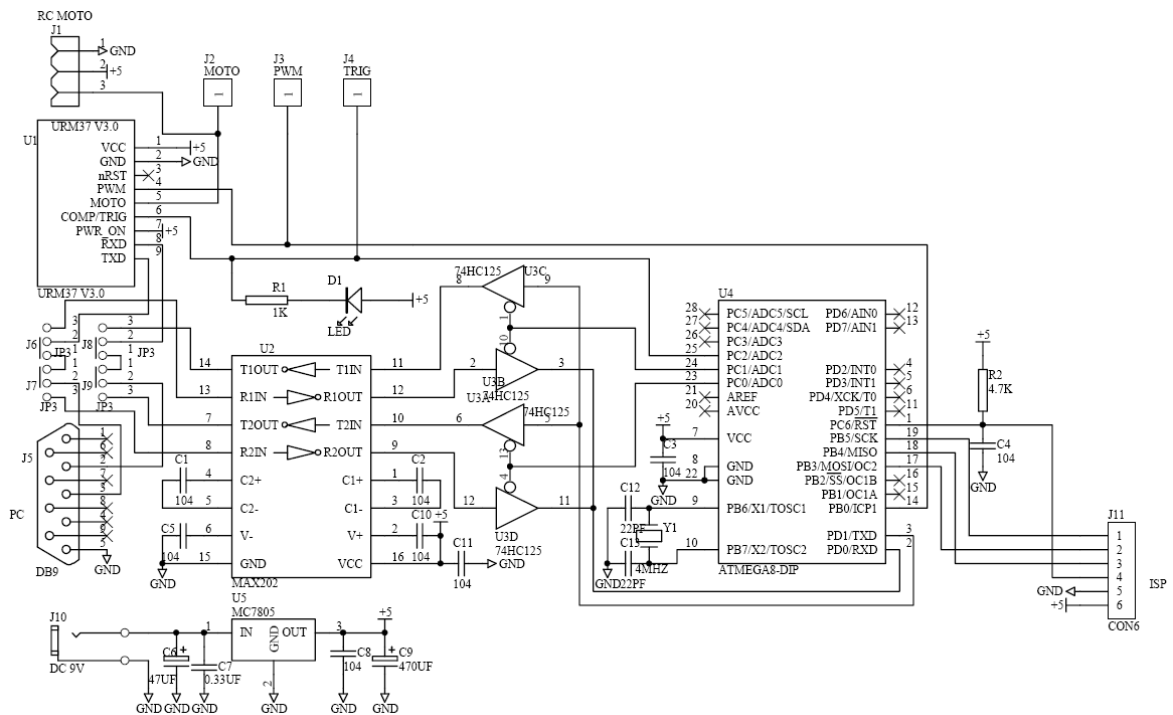
| | |
|--|-------------------|
| | 0x44+Add+Data+SUM |
|--|-------------------|

Note:NC stands for any data, SUM stands for sum, Add stands for address.

1. PWN_ON must be set to High to enable sensor.

Servo control command reference table:

| | | | | | | | | | | | | | | | | |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| DEC | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| HEX | 0 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
| Degree | 0 | 6 | 12 | 18 | 24 | 29 | 35 | 41 | 47 | 53 | 59 | 65 | 70 | 76 | 82 | 88 |
| DEC | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| HEX | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| Degree | 94 | 100 | 106 | 112 | 117 | 123 | 129 | 135 | 141 | 147 | 153 | 159 | 164 | 170 | 176 | 182 |
| DEC | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | |
| HEX | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D | 2E | |
| Degree | 188 | 194 | 200 | 206 | 211 | 217 | 223 | 229 | 235 | 241 | 247 | 252 | 258 | 264 | 270 | |



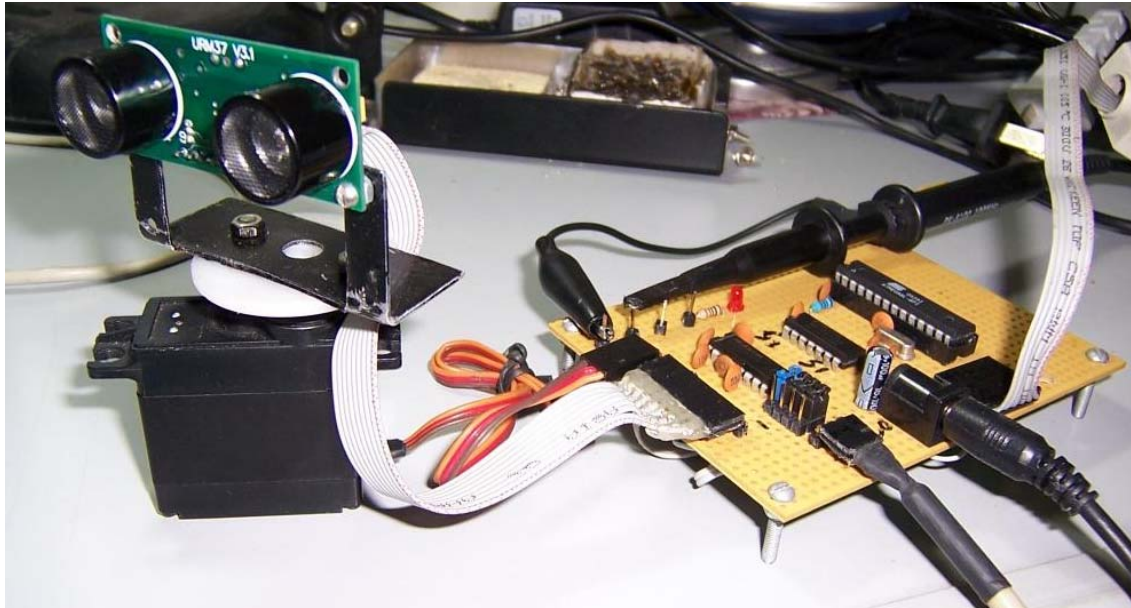


Figure URMV3.2 control a servo provides 270 degree scanning area

V3.2 Help Mate Download:

<http://www.yerobot.com/download/software/URMV3.2HelpMate.rar>

Samples:

1) Interface URM V3.2 to Arduino via PWM

<http://www.yerobot.com/forum/viewtopic.php?f=5&t=7&p=10&sid=72f4c2fbb84bf341e767351e898a3c0a#p10>

2) Interface URM V3.2 to Arduino via TTL (only for Rev 2)

<http://www.yerobot.com/forum/viewtopic.php?f=5&t=87&sid=35d6dd4789d8c96a70eacbfac108141b>

3) URM V3.2 Arduino Library

<http://www.yerobot.com/forum/viewtopic.php?f=5&t=121&sid=35d6dd4789d8c96a70eacbfac108141b>

4) URM V3.2 on Basic Stamp

<http://www.yerobot.com/forum/viewtopic.php?f=5&t=122&sid=4d22f1bdda56e7fe48a6e989ef1d8c90>

More information and support:

Website: www.YeRobot.com

Forum: www.YeRobot.com/Forum/

