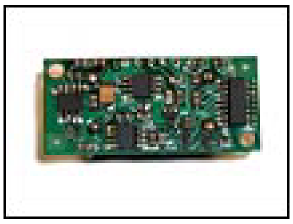
ULTRASONIC RANGE SENSOR

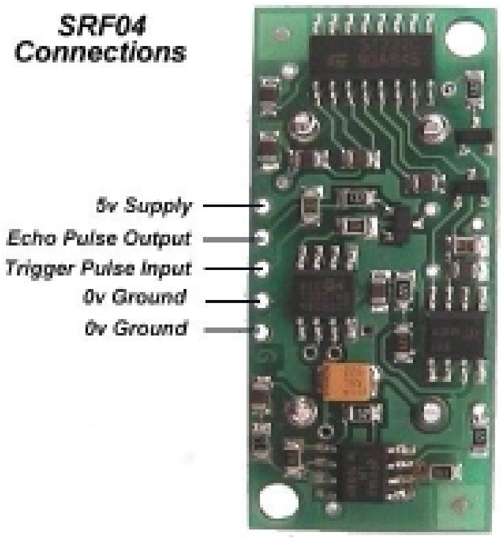
Specification:

The ultrasonic range sensor detects objects in it's path and can be used to calculate the range to the object. It is sensitive enough to detect a 3cm diameter broom handle at a distance of over 2m.



Voltage	- 5v
Current	- 30mA Typ. 50mA Max.
Frequency	- 40KHz
Max Range	- 3 m
Min Range	- 3 cm
Sensitivity	- Detect 3cm diameter broom handle at > 2 m
Input Trigger	- 10uS Min. TTL level pulse
Echo Pulse	- Positive TTL level signal, width proportional to range.
Small Size	- 43mm x 20mm x 17mm height

Electrical connection:



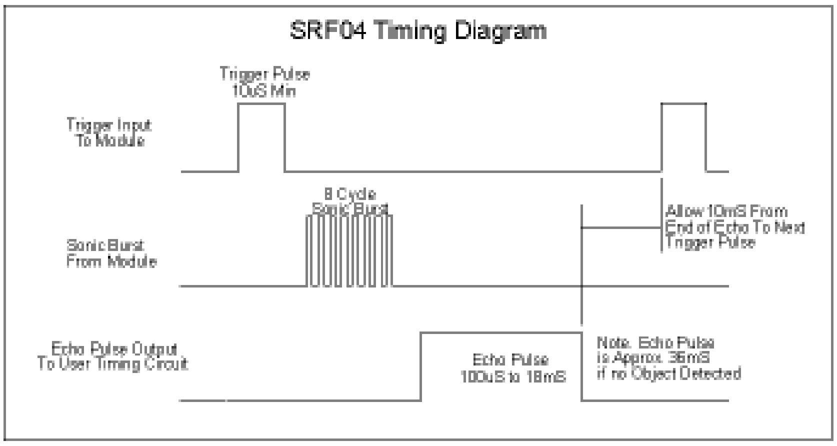
The SRF004 ultrasonic range finder has 5 connections pins. The power supply is connected to the 5V and 0V ground connections on the SRF004. (Note that **BOTH** the 'Mode' (hole 4) and '0V Ground' (hole 5) connections **MUST** be connected to 0V for correct operation with the PICAXE system).

Take care not to overheat, and therefore damage, the solder connection pads whilst making connections.

The SRF004 **Trigger Input** is connected to a PICAXE **output** pi
The SRF004 **Echo Output** is connected to a PICAXE **input** pin.

Operation with the PICAXE microcontroller:

The following program gives an example of how to use the SRF004 module with a PICAXE-18 microcontroller. Output 3 is used to trigger the SRF004 module via a 'pulsout' command. The SRF004 module then sends out the sonic burst, and sets the Echo Output connection high for the time it takes the sonic burst to be returned. Therefore the PICAXE input (input 6) is used to receive and time this echo pulse via a 'pulsin' command.


SRF04 Timing Diagram
Trigger Pulse 10uS Min
Trigger Input To Module
8 Cycle Sonic Burst
Sonic Burst From Module
Allow 10mS From End of Echo To Next Trigger Pulse
Echo Pulse Output To User Timing Circuit
Echo Pulse 100uS to 18mS
Note. Echo Pulse is Approx. 36mS if no Object Detected

The length of the echo pulse is then divided by 6.2 to give a value in cm, and displayed on the computer screen via the 'debug' command. Note that a word variable, w1, is used for the echo timing, as the echo pulse will be a value greater than 255 (maximum value of a byte variable). Word variables are made up of two byte variables and so have a maximum value of 65535 (in this case w1 is made up of b2 and b3, so these two byte variables must not be used anywhere else in the program).

Sample PICAXE Program:

```
symbol trig = 3          ' Define output pin for Trigger pulse
symbol echo  = 6         ' Define input pin for Echo pulse
symbol range = w1         ' 16 bit word variable for range

main:
   pulsout trig,2      ' produce 20uS trigger pulse (must be minimum of 10uS)
   pulsin echo,1,range   ' measures the range in 10uS steps
   pause 10            ' SRF004 mandatory 10mS recharge period after ranging completes
' now convert range to cm (divide by 6.2) or inches (divide by 14.9)
' as picaxe cannot use 6.2, multiply by 10 then divide by 62 instead
   let range = range * 10 / 62 ' multiply by 10 then divide by 62
   debug range                ' display range via debug command
   goto main                  ' and around forever
```

Technical Circuit Operation:

The circuit uses an on-board PIC12C508 microcontroller to perform the control functions of the standard 40kHz piezo transducers. The drive to the transmitting transducer could, in theory, be driven directly from the PIC. This 5v drive could give a useful range for large objects, but would be problematic when detecting smaller objects. Therefore, as the transducer can handle up to 20v of drive, a MAX232 IC (usually used for RS232 communication) is used as a voltage amplifier, providing about 16v of drive. This increases the sensitivity when detecting smaller objects.

The receiver is a classic two stage op-amp circuit. The input capacitor C8 blocks some residual DC which is always present. Each gain stage is set to 24, for a total gain of approx. 576. This is close to the 25 maximum gain available using the LM1458 op-amp (the gain bandwidth product for the LM1458 is 1MHz, therefore the maximum gain at 40kHz is 1000000/40000 = 25). The output of the amplifier is fed into an LM311 comparator. A small amount of positive feedback provides some hysterisis to give a clean stable output.

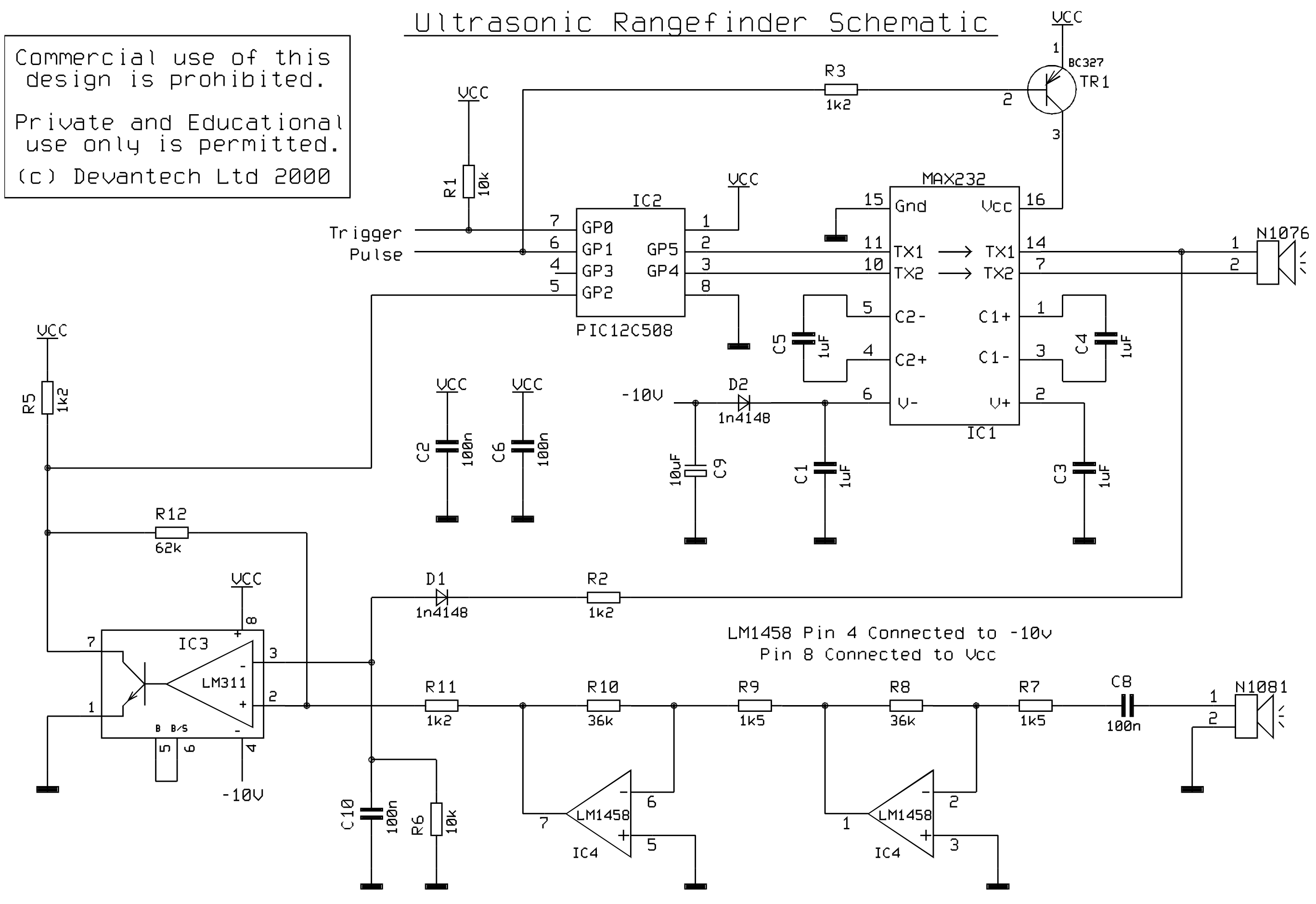
A problem with very close operation (down to 1-2cm) is that the receiver will pick up direct coupling from the transmitter, as they are physically positioned close together. The piezo transducer is also a mechanical object and so will keep resonating some time after the drive has been removed (up to 1mS). It is much harder to tell the difference between this direct coupled ringing and a returning echo, and so the detection threshold is electronically adjusted during this time so that only the echo is detectable. This adjustment is made by the 100n capacitor C10, which is charged to about –6v during the burst. This discharges quite quickly through the 10k resistor R6 to restore sensitivity for more distant echoes.

A convenient negative voltage for the op-amp and comparator is generated by the MAX232. Unfortunately, this also generates quite a bit of high frequency noise. The MAX232 is therefore shut down whilst listening for the echo. The 10uF capacitor C9 holds the negative rail stable just long enough to do this.

In operation, the PIC12C508 waits for an active low trigger pulse to come in from the controlling PICAXE microcontroller. It then generates eight cycles of 40khz pulses. The echo line is then raised to signal the PICAXE microcontroller to start timing. The raising of the echo line also shuts off the MAX232. After a while – no more than 10-12mS normally, the returning echo will be detected and the PIC12C508 will lower the echo line. The width of this pulse represents the flight time of the sonic burst. If no echo is detected then it will automatically time out after about 30mS. Because the MAX232 is shut down during echo detection, you must wait at least 10mS between measurement cycles for the +/- 10v supply to recharge.

Performance of this design is very good. It will reliably detect objects at just 3cm and will continue detecting down to 1cm or less (but after 2-3cm the pulse width doesn't get any smaller). Maximum range is a little over 3m. As an example of the sensitivity of this design, it will detect a 3 cm thick plastic broom handle at 2.4m. Average current consumption is reasonable at less than 50mA (typically about 30mA).

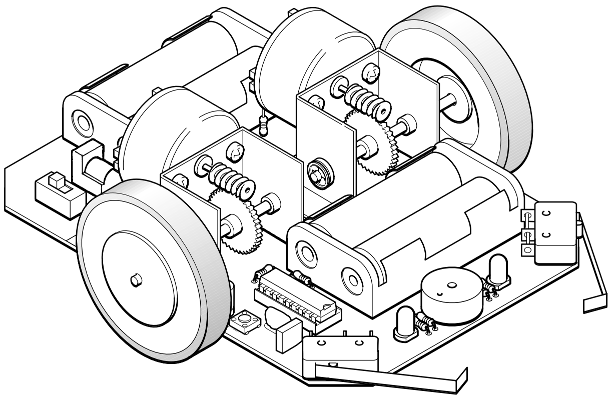
Circuit Diagram:


Ultrasonic Rangefinder Schematic
Commercial use of this design is prohibited.
Private and Educational use only is permitted.
(c) Devantech Ltd 2000
Trigger Pulse
IC2 PIC12C508 GP0 GP1 GP3 GP2 GP5 GP4
MAX232 IC1 Gnd Vcc TX1 TX2 C2- C2+ V- C1+ C1- V+
BC327 TR1
R1 10k
R3 1k2
R5 1k2
R12 62k
R2 1k2
D1 1n4148
D2 1n4148
-10V
C1 1uF C3 1uF C4 1uF C5 1uF
C9 10uF
C2 100n C6 100n
N1076
N1081
IC3 LM311
LM1458 Pin 4 Connected to -10v
Pin 8 Connected to Vcc
R11 1k2 R10 36k R9 1k5 R8 36k R7 1k5
C8 100n
C10 100n
R6 10k
LM1458 IC4
VCC

Use with CHI007 and MOD001 PIC Buggy Models:

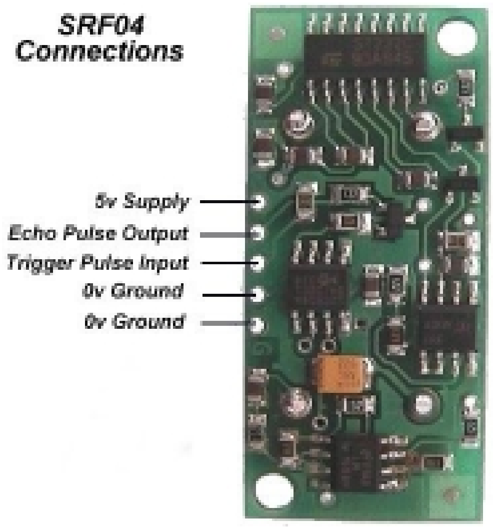
The SRF004 makes an ideal additional sensor for the PIC Buggy model.

To add the SRF004 to the buggy you will require:
· SRF004 ultrasonic module
· single core insulated wire
· soldering iron and solder
· glue gun



The SRF004 module is connected directly to input 6 and output 3 of the PICAXE-18 chip on the buggy (aswell as the V+ and 0V power supply). These input/outputs are not normally used on the buggy, and so the SRF004 connection will not affect any of the other buggy functions.

Connection:



The SRF004 must be mounted above the buggy (e.g. by using a small home-made aluminium bracket (not supplied))

The SRF004 has five solder connections which must be connected via wires to the solder joints on the bottom of the buggy PCB.

1.	Hole 1 – 5v Supply	– to PIC chip leg 14 (V+ Supply)
2.	Hole 2 – Echo Output	– to PIC chip leg 15 (input 6)
3.	Hole 3 – Trigger Input	– to PIC chip leg 9 (output 3)
4.	Hole 4 – Mode	– to PIC chip leg 5 (0V Ground)
5.	Hole 5 – 0V Ground	– to PIC chip leg 5 (0V Ground)

Note that **both** holes 4 and 5 must both be connected to 0V.
It is recommended that the wire links across the bottom of the buggy are secured in place using a glue-gun or similar.

The following example shows how to make the buggy stop when it detects an object 30cm in front of it.

```
symbol trig = 3          ‘ Define output pin for Trigger pulse
symbol echo  = 6         ‘ Define input pin for Echo pulse
symbol range = w1         ‘ 16 bit word variable for range


main:
  let pins = %10100000 ‘ buggy forward
loop:
   pulsout trig,2        ‘ produce 20uS trigger pulse
                          ‘ (must be minimum of 10uS)
   pulsin echo,1,range   ‘ measures the range in 10uS steps
   pause 10              ‘ SRF004 mandatory 10mS recharge period
                         ‘after ranging completes
‘ now convert range to cm (divide by 6.2) or inches (divide by 14.9)
‘ and stop if there is an object closer than 30 cm
   let range = range * 10 / 62 ‘ multiply by 10 then divide by 62
   if range < 30 then stop
   goto loop                  ‘ loop around if > 30


stop:
  let pins = %00000000          ‘ stop
  goto stop
```