

C Program Driver for MaxBotix MB7066 PW Ultrasonic Sensor

By Paul Sfetku, July 2011

Development System - Mikroelektronika

<http://www.mikroe.com/eng/home/index/>

Develop. Board: BigPic6 with GLCD (128x64)
Microcontroller: PIC18F8520/8722 family
Compiler: mikroC PRO for PIC
Interface: BigPic6 SW7 RG0: PORTG.B0 (output) → MB7066<4> start/stop ranging
BigPic6 SW7 RG1: PORTG.B1 (input) → MB7066<2> PW - pulse width

NOTE:

BigPic6 SW7 is set for pull-up on PORTG pins RG0/B0 & RG1/B1.

MB7066 Pin Out

Pin 2 - PW: This pin outputs a pulse width representation of range. To calculate distance, you can a scale factor of 58us/cm.

Pin 4 - RX: This pin is internally pulled high. The MB7066 will continually measure range and output if the pin is left unconnected or held high. If held low the MB7066 will stop ranging. Bring pin high for >= 20uS for range reading.

Pin 6 - Vcc (3V - 5.5V): Average and peak current draw for 3.3V operation is 2.1mA (50mA peak). 5V operation is 3.4mA (100mA peak) respectively. Peak current is used during sonar pulse transmit.

Pin 7 - GND: Return for the DC power supply. GND (& V+) must be ripple and noise free for best operation.

NOTE

Refer to the MB7066 PW documentation for an explanation of **MB7066 Real-time Operation & Timing**

Program Code Explanation

The C code listed below was generated using the mikroC Pro PIC compiler. The code will drive the MaxBotix MB7066 pulse width (PW) sensor, and any MaxBotix pulse width sensor.

The MB7066 is operated in free running mode with MB7066<4> pulled High. To operate the MB7066 in start/stop mode, you can uncomment code statements 8,9,10 and code statements 17,18,19.

Statements 15-20 can be used to determine when and if the MB7066 fails to detect a return pulse from the given target. A value for **maxvalue** in statement 15 is needed to calibrate the sensor for a maximum time or maximum distance value.

The code to drive the MB7066 is contained within an infinite loop declared in statement 11. The MB7066 operates on a 100ms PW cycle. MB7066 pin-2 PW sets high to start a ranging cycle, and sets Low if and when a target object is detected. If no target is detected, pin-2 PW will be held high for up to 62ms or 1068cm, the maximum range value.

Code statements 12-25 are designed detect and measure the period of each PW begin/end cycle. Statement 12 detects when the PW goes High. The delay in statement 13 is used to calibrate the sensor's ranging resolution to detect changes at one inch increments. The variable in statement 14 counts the number of delays until the PW signal does low.

Statements 21-24 calculate the range and detect the beginning of the next 100ms PW cycle. Statement 22 calculates the range based on the time counted in statement 14, and based on a constant factor (.5) that can be adjusted to provide an accurate range measurement. The loop in statement 24 is used to wait for the PW to set High, indicating the beginning of the next 100ms PW ranging cycle.

```

1 // Declarations -----
2 TRISG = 0x02;          // PORTG DDR: RG0 (output), RG1 (input)
3 unsigned int msec     = 0; // time counter - microseconds
4 unsigned int inches   = 0; // calculated distance
5 unsigned maxvalue     = 0; // sensor maximum - echo not detected
6
7 void main() {
8     //PORTG.B0 = 0;      // MB7066<4>: stop ranging - set Low
9     //PORTG.B0 = 1;      // MB7066<4>: start ranging - set High
10    //Delay_us(20);      // 20us delay for start of ranging
11    while (1) {          // infinite loop
12        if (PORTG.B1) { // MB7066<2>: PW - check if High
13            Delay_us(30); // delay factor 30us - gives 1 inch resolution
14            msec++;      // count time PW is High each microseconds
15            //if (msec > maxvalue) { // PW max time 62ms -> echo not detected
16                // msec = 0; // clear counter
17                // PORTG.B0 = 0; // MB7066<4>: stop ranging - set Low
18                // PORTG.B0 = 1; // MB7066<4>: start ranging - set High
19                // Delay_us(20); // 20us delay for start of ranging
20                //}
21        } else { // MB7066<2>: PW -> Low
22            inches = .5*msec; // range (inches): target correction factor = .5
23            msec = 0; // clear counter
24            while (!PORTG.B1); // wait for MB7066<2> PW to go High -> start 100ms range cycle
25        }
26    }
27 }

```