LAMPIRAN A
LAMPIRAN B
This program was produced by the
CodeWizardAVR V1.25.3 Professional
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : TA - Sensor Parkir
Version : 1.0
Date    : 10/05/2010
Author  : Hendri / 0727004
Company : HP
Comments: TUGAS AKHIR

Chip type    : ATmega16
Program type : Application
Clock frequency : 11,059200 MHz
Memory model  : Small
External SRAM size : 0
Data Stack size : 256
*******************************************************************************/

#include <mega16.h>

// Alphanumeric LCD Module functions
#asm
  .equ __lcd_port=0x18 ;PORTB
#include <lcd.h>
#include <delay.h>
#include <stdio.h>

#define PULSE1 PORTA.0
#define PULSE2 PORTA.1
#define ECHO1 PINA.0
#define ECHO2 PINA.1
#define ARAH1 DDRA.0
#define ARAH2 DDRA.1
#define OUT 1
#define INP 0

// Declare your global variables here
unsigned int count1=0;
unsigned int count2=0;
float jarak1;
float jarak2;
unsigned char kata1[16];
unsigned char kata2[16];

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization

    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0xFF;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

while (1)
{
    // Place your code here
    // port as output
count1=0;

ARAH1=OUT;
// pulse 2us
PULSE1=1;
delay_ms(5);
PULSE1=0;

// port as input
ARAH1=INP;
// with pull-up
PULSE1=1;

while (ECHO1==0) {};

while (ECHO1==1)
{
    count1++;
}

jarak1=(count1*1/65.536);
sprintf(kata1,"Jarak=%3.2f cm",jarak1);
lcd_clear();
lcd_gotoxy(0,0);
lcd_puts(kata1);
if(jarak1<=30)
{
    PORTD.7=1;
    delay_ms(100);
PORTD.7=0;
delay_ms(100);
}

// port as output
count2=0;

ARAH2=OUT;

// pulse 2us
PULSE2=1;
delay_ms(5);
PULSE2=0;

// port as input
ARAH2=INP;

// with pull-up
PULSE2=1;

while (ECHO2==0) {};

while (ECHO2==1)
{
count2++;
}

jarak2=(count2*1/65.536);
sprintf(kata2,"Jarak=%3.2f cm",jarak2);
lcd_clear();
lcd_gotoxy(0,1);
lcd.puts(kata2);
if(jarak2<=30)
{
    PORTD.7=1;
delay_ms(100);
    PORTD.7=0;
delay_ms(100);
}
};
}
LAMPIRAN C
Pin Configurations

Figure 1. Pinout ATmega16

PDIP

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PA0 (ADC0)</td>
</tr>
<tr>
<td>2</td>
<td>PA1 (ADC1)</td>
</tr>
<tr>
<td>3</td>
<td>PA2 (ADC2)</td>
</tr>
<tr>
<td>4</td>
<td>PA3 (ADC3)</td>
</tr>
<tr>
<td>5</td>
<td>PA4 (ADC4)</td>
</tr>
<tr>
<td>6</td>
<td>PA5 (ADC5)</td>
</tr>
<tr>
<td>7</td>
<td>PA6 (ADC6)</td>
</tr>
<tr>
<td>8</td>
<td>PA7 (ADC7)</td>
</tr>
<tr>
<td>9</td>
<td>AREF</td>
</tr>
<tr>
<td>10</td>
<td>VCC</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
</tr>
<tr>
<td>12</td>
<td>XTAL2</td>
</tr>
<tr>
<td>13</td>
<td>XTAL1</td>
</tr>
<tr>
<td>14</td>
<td>RXD</td>
</tr>
<tr>
<td>15</td>
<td>TXD</td>
</tr>
<tr>
<td>16</td>
<td>INT0</td>
</tr>
<tr>
<td>17</td>
<td>INT1</td>
</tr>
<tr>
<td>18</td>
<td>OC1B</td>
</tr>
<tr>
<td>19</td>
<td>OC1A</td>
</tr>
<tr>
<td>20</td>
<td>ICP1</td>
</tr>
<tr>
<td>21</td>
<td>PD7 (OC2)</td>
</tr>
</tbody>
</table>

TQFP/QFN/MLF

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PA0 (ADC0)</td>
</tr>
<tr>
<td>2</td>
<td>PA1 (ADC1)</td>
</tr>
<tr>
<td>3</td>
<td>PA2 (ADC2)</td>
</tr>
<tr>
<td>4</td>
<td>PA3 (ADC3)</td>
</tr>
<tr>
<td>5</td>
<td>PA4 (ADC4)</td>
</tr>
<tr>
<td>6</td>
<td>PA5 (ADC5)</td>
</tr>
<tr>
<td>7</td>
<td>PA6 (ADC6)</td>
</tr>
<tr>
<td>8</td>
<td>PA7 (ADC7)</td>
</tr>
<tr>
<td>9</td>
<td>AREF</td>
</tr>
<tr>
<td>10</td>
<td>VCC</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
</tr>
<tr>
<td>12</td>
<td>XTAL2</td>
</tr>
<tr>
<td>13</td>
<td>XTAL1</td>
</tr>
<tr>
<td>14</td>
<td>RXD</td>
</tr>
<tr>
<td>15</td>
<td>TXD</td>
</tr>
<tr>
<td>16</td>
<td>INT0</td>
</tr>
<tr>
<td>17</td>
<td>INT1</td>
</tr>
<tr>
<td>18</td>
<td>OC1B</td>
</tr>
<tr>
<td>19</td>
<td>OC1A</td>
</tr>
<tr>
<td>20</td>
<td>ICP1</td>
</tr>
</tbody>
</table>

NOTE:
Bottom pad should be soldered to ground.

Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.
Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram
## Ordering Information

<table>
<thead>
<tr>
<th>Speed (MHz)</th>
<th>Power Supply</th>
<th>Ordering Code</th>
<th>Package</th>
<th>Operation Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>2.7 - 5.5V</td>
<td>ATmega16L-8AC</td>
<td>44A</td>
<td>Commercial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-8PC</td>
<td>40P6</td>
<td>(0°C to 70°C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-8MC</td>
<td>44M1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-8AI</td>
<td>44A</td>
<td>Industrial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-8AU</td>
<td>44A</td>
<td>(-40°C to 85°C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-8PU</td>
<td>40P6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-8MU</td>
<td>44M1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16AC</td>
<td>44A</td>
<td>Commercial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16PC</td>
<td>40P6</td>
<td>(0°C to 70°C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16MC</td>
<td>44M1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16AI</td>
<td>44A</td>
<td>Industrial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16AU</td>
<td>44A</td>
<td>(-40°C to 85°C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16PU</td>
<td>40P6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ATmega16L-16MU</td>
<td>44M1</td>
<td></td>
</tr>
</tbody>
</table>

### Note: 1. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

### Package Type

<table>
<thead>
<tr>
<th>Package Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>44A</td>
<td>44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)</td>
</tr>
<tr>
<td>40P6</td>
<td>40-pin, 0.600” Wide, Plastic Dual Inline Package (PDIP)</td>
</tr>
<tr>
<td>44M1</td>
<td>44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)</td>
</tr>
</tbody>
</table>
LAMPIRAN D
### Dot Matrix Liquid Crystal Display Modules

#### CHARACTER TYPE

- **FEATURES:**
  - Slim, lightweight and low power consumption
  - High contrast and wide viewing angle
  - Built-in controller for easy interfacing
  - LCD modules with built-in EL or LED backlight

#### SPECIFICATIONS:

<table>
<thead>
<tr>
<th>Character Format (character x line)</th>
<th>Standard products</th>
<th>Products of optimal specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M1641</td>
<td>16 x 1</td>
<td>16 x 1</td>
</tr>
<tr>
<td>M1632</td>
<td>16 x 2</td>
<td>16 x 2</td>
</tr>
<tr>
<td>M1652</td>
<td>16 x 4</td>
<td>16 x 4</td>
</tr>
<tr>
<td>L1652</td>
<td>26 x 2</td>
<td></td>
</tr>
<tr>
<td>Reflective</td>
<td>M1641/0AS</td>
<td>M1632/0AS</td>
</tr>
<tr>
<td>E.L. backlight</td>
<td>M1641/0WS</td>
<td>M1632/0WS</td>
</tr>
<tr>
<td>LCD</td>
<td>M1641/7YS</td>
<td>M1632/7YS</td>
</tr>
<tr>
<td>Reflective (wide temp)</td>
<td>M1641/3CS</td>
<td>M1632/3CS</td>
</tr>
<tr>
<td>E.L. backlight (wide temp)</td>
<td>M1641/6000S</td>
<td>M1632/6000S</td>
</tr>
<tr>
<td>Character font</td>
<td>5x7 dots + cursor</td>
<td>5x7 dots + cursor</td>
</tr>
<tr>
<td>Module</td>
<td>Reflective</td>
<td>80.0 x 30.0 x 11.3</td>
</tr>
<tr>
<td>Size</td>
<td>E.L. backlight</td>
<td>80.0 x 30.0 x 11.3</td>
</tr>
<tr>
<td>(HxWxT) mm</td>
<td>LED backlight</td>
<td>80.0 x 30.0 x 15.8</td>
</tr>
<tr>
<td>Viewing area (HxV) mm</td>
<td>64.5 x 13.8</td>
<td>59.0 x 24.0</td>
</tr>
<tr>
<td>Character size (HxV) mm</td>
<td>3.7x5.73</td>
<td>2.7x4.27</td>
</tr>
<tr>
<td>Dot size (HxV) mm</td>
<td>0.50x0.50</td>
<td>0.50x0.50</td>
</tr>
<tr>
<td>Power supply voltage (VCC-VSS) V</td>
<td>+5.0 V</td>
<td>+5.0 V</td>
</tr>
<tr>
<td>Current consumption (mA)</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Driving method (duty)</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>Built-in LSI</td>
<td>KS0066</td>
<td>KS0066</td>
</tr>
<tr>
<td>Operating temperature (°C)</td>
<td>-50 to +50</td>
<td>-50 to +50</td>
</tr>
<tr>
<td>Storage temperature (°C)</td>
<td>-20 to +60</td>
<td>-20 to +60</td>
</tr>
<tr>
<td>Weight (g, typ.)</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>Inverters</td>
<td>Model</td>
<td>55</td>
</tr>
<tr>
<td>by EL</td>
<td>Power supply (V)</td>
<td>+5.0</td>
</tr>
<tr>
<td>LED</td>
<td>current consumption (mA)*3</td>
<td>+5.0</td>
</tr>
<tr>
<td>Backlight</td>
<td>Forward current consumption (mA)</td>
<td>+5.0</td>
</tr>
<tr>
<td>Forward input voltage (V, typ.)</td>
<td>+4.1</td>
<td>+4.1</td>
</tr>
</tbody>
</table>

---

1: Excluding cursor
2: With external temperature compensation
3: Installing EL backlight
4: Based on normal temperature range

Since our policy is one of continuous improvements, we reserve the right to change the specifications for the products in the catalogue without notice.
### SPECIFICATIONS:

<table>
<thead>
<tr>
<th>Character Format (character x line)</th>
<th>20 x 2</th>
<th>20 x 4</th>
<th>24 x 2</th>
<th>40 x 2</th>
<th>40 x 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>L2022</td>
<td>L2014</td>
<td>L2432</td>
<td>L4042</td>
<td>M4024</td>
</tr>
<tr>
<td>Reflective</td>
<td>L20140U000S</td>
<td>L24320U000S</td>
<td>L40420U000S</td>
<td>M40240US</td>
<td></td>
</tr>
<tr>
<td>EL backlight</td>
<td>L201421U000S</td>
<td>L243221U000S</td>
<td>L404221U000S</td>
<td>M40240US</td>
<td></td>
</tr>
<tr>
<td>LED backlight</td>
<td>L204241U000S</td>
<td>L243241U000S</td>
<td>L404241U000S</td>
<td>M40240US</td>
<td></td>
</tr>
<tr>
<td>Reflective (wide temp)</td>
<td>L20220P000S</td>
<td>L20140P000S</td>
<td>L24320P000S</td>
<td>L40420P000S</td>
<td>M40240P000S</td>
</tr>
<tr>
<td>LED backlight (wide temp)</td>
<td>L2022BP000S</td>
<td>L20141P000S</td>
<td>L2432BP000S</td>
<td>L4042BP000S</td>
<td>M40241P000S</td>
</tr>
<tr>
<td>Character font</td>
<td>5x7 dot + cursor</td>
<td>5x7 dot + cursor</td>
<td>5x7 dot + cursor</td>
<td>5x7 dot + cursor</td>
<td>5x7 dot + cursor</td>
</tr>
<tr>
<td>Module</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
</tr>
<tr>
<td>(HxW) mm</td>
<td>180.0 x 40.0 x 10.5</td>
<td>98.0 x 60.0 x 11.6</td>
<td>118.0 x 36.0 x 11.3</td>
<td>182.0 x 33.5 x 11.3</td>
<td>190.0 x 54.0 x 10.1</td>
</tr>
<tr>
<td>(HxW) mm</td>
<td>180.0 x 40.0 x 10.5</td>
<td>98.0 x 60.0 x 11.6</td>
<td>118.0 x 36.0 x 11.3</td>
<td>182.0 x 33.5 x 11.3</td>
<td>190.0 x 54.0 x 10.1</td>
</tr>
<tr>
<td>Viewing area (HxW) mm</td>
<td>149.0 x 23.0</td>
<td>76.0 x 25.2</td>
<td>94.5 x 17.8</td>
<td>154.4 x 15.8</td>
<td>147.0 x 29.5</td>
</tr>
<tr>
<td>Character size (HxW) mm *1</td>
<td>6.00 x 0.66</td>
<td>2.96 x 0.45</td>
<td>3.20 x 0.45</td>
<td>3.20 x 0.45</td>
<td>2.78 x 0.47</td>
</tr>
<tr>
<td>Dot size (HxW) mm</td>
<td>1.12 x 1.12</td>
<td>3.55 x 0.55</td>
<td>0.60 x 0.65</td>
<td>0.60 x 0.65</td>
<td>0.20 x 0.55</td>
</tr>
<tr>
<td>Power supply voltage (V0D-YSS) V</td>
<td>+5 V</td>
<td>+5 V</td>
<td>+5 V</td>
<td>+5 V</td>
<td>+5 V</td>
</tr>
<tr>
<td>Current consumption (mA/typ)</td>
<td>LDU</td>
<td>4.2</td>
<td>2.9</td>
<td>2.5</td>
<td>3.0</td>
</tr>
<tr>
<td>Driving method (duty)</td>
<td>1/16</td>
<td>1/16</td>
<td>1/16</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>Built-in LSI</td>
<td>K9066</td>
<td>K9066</td>
<td>K9066</td>
<td>K9066</td>
<td>K9066</td>
</tr>
<tr>
<td>Operating temperature (°C)</td>
<td>normal temp.</td>
<td>0 to 50</td>
<td>0 to 50</td>
<td>0 to 50</td>
<td>0 to 50</td>
</tr>
<tr>
<td>Storage temperature (°C)</td>
<td>normal temp.</td>
<td>-20 to 70</td>
<td>-20 to 70</td>
<td>-20 to 70</td>
<td>-20 to 70</td>
</tr>
<tr>
<td>Weight (g, typ.)</td>
<td>80</td>
<td>55</td>
<td>40</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>EL backlight</td>
<td>60</td>
<td>45</td>
<td>75</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>LED backlight</td>
<td>110</td>
<td>70</td>
<td>69</td>
<td>95</td>
<td>140</td>
</tr>
<tr>
<td>Power supply (V)</td>
<td>+5.0</td>
<td>+5.0</td>
<td>+5.0</td>
<td>+5.0</td>
<td>+5.0</td>
</tr>
<tr>
<td>Inverters for EL</td>
<td>-5.0</td>
<td>45</td>
<td>45</td>
<td>25</td>
<td>80</td>
</tr>
<tr>
<td>LED backlight</td>
<td>320</td>
<td>240</td>
<td>150</td>
<td>260</td>
<td>490</td>
</tr>
</tbody>
</table>

*1 : Excluding cursor

*2 : With external temperature compensation

*3 : Including EL backlight

*4 : Based on normal temperature range
Dot Matrix Liquid Crystal Display Modules

**GRAPHIC TYPE**

- **FEATURES**:
  - Wide viewing angle and high contrast
  - Full dot configuration fits any application
  - Slim, lightweight and low power consumption
  - Available in STN and FSTN

<table>
<thead>
<tr>
<th>Dot format (H x V x T)</th>
<th>97 x 32</th>
<th>128 x 32</th>
<th>128 x 64</th>
<th>128 x 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>V97931</td>
<td>G1213</td>
<td>G1216</td>
<td>G1226</td>
</tr>
<tr>
<td>STN type (Gray mode)</td>
<td>Reflective</td>
<td>Reflective wide temp.</td>
<td>Reflective</td>
<td>Reflective</td>
</tr>
<tr>
<td></td>
<td>built-in RAM</td>
<td>built-in RAM</td>
<td>built-in RAM</td>
<td>built-in RAM</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>G121300N0005</td>
<td>G121600N0005</td>
<td>-</td>
</tr>
<tr>
<td>FSTN type (BW mode)</td>
<td>Transmissive</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>with CFL backlight</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>built-in controller</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Transmissive</td>
<td>built-in RAM</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>G121381H0005</td>
<td>G121681H0005</td>
<td>-</td>
</tr>
<tr>
<td>Module size (H x V x T)</td>
<td>47.5 x 23.9</td>
<td>60.0 x 21.3</td>
<td>60.0 x 25.0</td>
<td>70.7 x 38.8</td>
</tr>
<tr>
<td>mm</td>
<td>75.0 x 41.5 x 6.8</td>
<td>75.0 x 52.7 x 6.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Viewing area (mm)</td>
<td>75.0 x 41.5 x 8.9</td>
<td>75.0 x 52.7 x 8.9</td>
<td>93.0 x 70.0 x 11.4</td>
<td>-</td>
</tr>
<tr>
<td>Dot pitch (H x V x T)</td>
<td>0.39 x 0.52</td>
<td>0.43 x 0.51</td>
<td>0.43 x 0.43</td>
<td>0.48 x 0.48</td>
</tr>
<tr>
<td>mm</td>
<td>0.40 x 0.48</td>
<td>0.40 x 0.48</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Power supply voltage (V)</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>(VDD-VSS)</td>
<td>5.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(VCL-VSS)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Current consumption Ho</td>
<td>0.10</td>
<td>2.0</td>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>(mA, typ.)</td>
<td>-</td>
<td>1.8</td>
<td>1.6</td>
<td>2.0</td>
</tr>
<tr>
<td>Driving method (duty)</td>
<td>1/33</td>
<td>1/64</td>
<td>1/64</td>
<td>1/64</td>
</tr>
<tr>
<td>Built-in LSI Driver</td>
<td>SED1530</td>
<td>HD61202</td>
<td>HD61203</td>
<td>K50107</td>
</tr>
<tr>
<td>or equivalent</td>
<td>or equivalent</td>
<td>or equivalent</td>
<td>or equivalent</td>
<td>or equivalent</td>
</tr>
<tr>
<td>Operating temperature range (°C)</td>
<td>-20 to +70</td>
<td>-20 to +70</td>
<td>-20 to +70</td>
<td>0 to +50</td>
</tr>
<tr>
<td>Storage temperature range (°C)</td>
<td>-30 to +80</td>
<td>-30 to +80</td>
<td>-30 to +80</td>
<td>-20 to +60</td>
</tr>
<tr>
<td>Weight (g, typ.)</td>
<td>Reflective</td>
<td>10</td>
<td>20</td>
<td>35</td>
</tr>
<tr>
<td>LEDs (typ.)</td>
<td>Reflective</td>
<td>25</td>
<td>40</td>
<td>45</td>
</tr>
<tr>
<td>LED backlight</td>
<td>Reflective</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Forward current consumption (mA)</td>
<td>-</td>
<td>40</td>
<td>90</td>
<td>125</td>
</tr>
<tr>
<td>Forward input voltage (V, typ.)</td>
<td>-</td>
<td>3.6</td>
<td>4.1</td>
<td>4.1</td>
</tr>
<tr>
<td>Inverter for LED</td>
<td>Built-in controller</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Power supply voltage (V)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Current consumption (mA, typ.)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

*1: built-in DC/DC converter (single power source)
*2: Use with external temperature compensation circuit

Since our policy is one of continuous improvements we reserve the right to change the specifications of the products in the catalogue without notice.
<table>
<thead>
<tr>
<th>Dot format (H x V x D)</th>
<th>240 x 64</th>
<th>240 x 128</th>
<th>320 x 200</th>
<th>320 x 240</th>
<th>640 x 200</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>G2446</td>
<td>G242C</td>
<td>G3210</td>
<td>G3248</td>
<td>G4446</td>
</tr>
<tr>
<td><strong>STN type</strong> (Gray mode)</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td><strong>FSTN type</strong> (B&amp;W mode)</td>
<td>Transmissive</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td></td>
<td>G2446R5R1AOS</td>
<td>G242C0R1ACS</td>
<td>G3210R5R1AOS</td>
<td>G3346R5R1AOS</td>
<td>G4446R5R1AOS</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td><strong>Module size (H x W x T)</strong></td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
<td>Reflective</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td><strong>CFL backlight</strong></td>
<td>191.0 x 78.0 x 15.1</td>
<td>196.0 x 110.0 x 15.1</td>
<td>166.0 x 134.0 x 15.1</td>
<td>196.0 x 134.0 x 15.1</td>
<td>260.0 x 122.0 x 15.7</td>
</tr>
<tr>
<td><strong>Viewing area (H x W)</strong></td>
<td>134.0 x 41.0</td>
<td>134.0 x 76.0</td>
<td>128.0 x 110.0</td>
<td>128.0 x 110.0</td>
<td>216.0 x 83.0</td>
</tr>
<tr>
<td><strong>Dot pitch (H x W)</strong></td>
<td>0.45 x 0.48</td>
<td>0.47 x 0.47</td>
<td>0.36 x 0.49</td>
<td>0.32 x 0.39</td>
<td>0.36 x 0.39</td>
</tr>
<tr>
<td><strong>Power supply voltage (V)</strong></td>
<td>VDD (VSS)</td>
<td>VDD (VSS)</td>
<td>VDD (VSS)</td>
<td>VDD (VSS)</td>
<td>VDD (VSS)</td>
</tr>
<tr>
<td></td>
<td>+5.0</td>
<td>+5.0</td>
<td>+5.0</td>
<td>+5.0</td>
<td>+5.0</td>
</tr>
<tr>
<td><strong>Current consumption</strong></td>
<td>LED</td>
<td>30</td>
<td>8</td>
<td>7.5</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>LED (built-in controller)</td>
<td>15</td>
<td>40</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td><strong>Driver</strong></td>
<td>M5S105</td>
<td>M5S105</td>
<td>M5S105</td>
<td>M5S105</td>
<td>M5S105</td>
</tr>
<tr>
<td></td>
<td>0 to +50</td>
<td>0 to +50</td>
<td>0 to +50</td>
<td>0 to +50</td>
<td>0 to +50</td>
</tr>
<tr>
<td><strong>Weight (g, typ.)</strong></td>
<td>290</td>
<td>390</td>
<td>390</td>
<td>360</td>
<td>360</td>
</tr>
</tbody>
</table>

1: built-in DC/DC converter (angle power source)
2: Uses with external temperature compensation.

Since our policy is one of continuous improvements, we reserve the right to change the specifications of the products in the catalogue without notice.
Liquid Crystal Display Modules

**REFLECTIVE/TRANSLECTIVE/TRANSMISSIVE LCD**

1. Reflective LCD
   - Reflector bonded to the rear polarizer reflects the incoming light. Low power consumption because no backlight is required.

2. Transflective LCD
   - Transflective LCD
   - Transflective LCD bonded to the rear polarizer reflects light from the front as well as enabling light to pass through the back. Used with backlight off in bright light and with it on in low light to reduce power consumption.

3. Transmissive LCD
   - Transmissive LCD
   - Without reflector or transflector bonded to the rear polarizer. Backlight required. Most common is transmissive negative image.

**POSITIVE/NEGATIVE MODE**

<table>
<thead>
<tr>
<th>Positive type</th>
<th>Negative type</th>
<th>Negative type (Inversed image) (when data is inverted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

**TN TYPE/STN TYPE/FSTN TYPE**

<table>
<thead>
<tr>
<th>TN (Background/dot color)</th>
<th>STN</th>
<th>FSTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gray/Black</td>
<td>Yellow/Dark blue</td>
<td>White/Black</td>
</tr>
</tbody>
</table>

TN (Twisted Nematic) type is most conventional and economical. It is used for static drive LCD and low-duty drive LCD (watch, calculator, etc.).

STN (Super Twisted Nematic) type has a higher twist angle, and thus provides clear visibility and wider viewing angle. This is suitable especially for high-duty drive LCD.

FSTN (Film Super Twisted Nematic) type utilizes RCF (Retardation Control Film) to remove the coloring of STN LCD. Thus, FSTN type provides easy-to-read black-and-white display.

**STRUCTURE AND FEATURE OF LCD MODULE WITH BACKLIGHT**

- **CFL (Cold Cathode Fluorescent Lamp) backlight**
  - Features: high brightness, long service life, inverter required

- **EL (Electroluminescent Lamp) backlight**
  - Features: EL: thin, inverter required
  - LED: long service life, low voltage driving, no inverter required

**POWER SUPPLY**

- **Character modules (single power supply)**
  - G244G, G242C (Built-in DC-DC conv.)

- **Character Modules (Dual power supply)**
  - Y2106 and G1228

- **G321D, G324E and G646D**

Note 1: Contrast can be adjusted by VR.
Note 2: For module with backlight, power supply for backlight is necessary.
LAMPIRAN E
PING))}™ Ultrasonic Distance Sensor (#28015)

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 BASIC Stamp® or Javalin Stamp microcontrollers, requiring only one I/O pin.

The PING)) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.

**Features**

- Supply Voltage – 5 VDC
- Supply Current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 in to 3.3 yards)
- Input Trigger – positive TTL pulse, 2 µs min, 5 µs typ.
- Echo Pulse – positive TTL pulse, 115 µs to 135 ms
- Echo hold-off – 750 µs from full of Trigger pulse
- Burst Frequency – 40 kHz for 200 µs
- Burst Indicator LED shows sensor activity
- Delay before next measurement – 200 µs
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)

**Dimensions**

![Dimensions Diagram](image-url)
Pin Definitions

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>Ground (Vss)</td>
</tr>
<tr>
<td>5 V</td>
<td>5 VDC (Vdd)</td>
</tr>
<tr>
<td>SIG</td>
<td>Signal (I/O pin)</td>
</tr>
</tbody>
</table>

The PING))) sensor has a male 3-pin header used to supply power (5 VDC), ground, and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extension cable (Parallax part #865-00002). Standard connections are shown in the diagram to the right.

Quick-Start Circuit

This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp® 2 via the Board of Education™ breadboard area. The PING))) module’s GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example program Ping_Demo.B52 listed on page 7.

Servo Cable and Port Cautions

If you want to connect your PING))) sensor to a Board of Education using a servo extension cable, follow these steps:

1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown.
4. If your Board of Education servo ports do not have a jumper, do not use them with the PING))) sensor. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the servo cable directly to the breadboard with a 3-pin header. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.
Theory of Operation

The PING sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air at about 11 30 feet per second, hits an object and then bounces back to the sensor. The PING sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.

![Diagram of sensor operation]

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_{OUT}</td>
<td>2 μS (min), 5 μS typical</td>
</tr>
<tr>
<td>t_{CLOCKOFF}</td>
<td>700 μS</td>
</tr>
<tr>
<td>t_{BURST}</td>
<td>200 μS @ 40 kHz</td>
</tr>
<tr>
<td>t_{MAXMIN}</td>
<td>115 μS</td>
</tr>
<tr>
<td>t_{MAXMAX}</td>
<td>18.5 mS</td>
</tr>
</tbody>
</table>

Test Data

The test data on the following page is based on the PING sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING sensor.
Test 1

Sensor Elevation: 40 in. (101.6 cm)
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation
Test 2

Sensor Elevation: 40 in. (101.6 cm)
Target: 12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole
- target positioned parallel to backplane of sensor
Program Example: BASIC Stamp 2 Microcontroller

The following program demonstrates the use of the PING))) sensor with the BASIC Stamp 2 microcontroller. Any model of BASIC Stamp 2 module will work with this program as conditional compilation techniques are used to make adjustments based on the module that is connected.

The heart of the program is the Get_Sonar subroutine. This routine starts by making the output bit of the selected I0 pin zero - this will cause the successive PULSOUT to be low-high-low as required for triggering the PING))) sensor. After the trigger pulse falls the sensor will wait about 200 microseconds before transmitting the ultrasonic burst. This allows the 852 to load and prepare the next instruction. That instruction, PULSIN, is used to measure the high-going pulse that corresponds to the distance to the target object.

The raw return value from PULSIN must be scaled due to resolution differences between the various members of the 852 family. After the raw value is converted to microseconds, it is divided by two in order to remove the "return trip" of the echo pulse. The value now held in rawDist is the distance to the target in microseconds.

Conversion from microseconds to inches (or centimeters) is now a simple matter of math. The generally-accepted value for the speed-of-sound is 1130 feet per second. This works out to 13,560 inches per second or one inch in 73,746 microseconds. The question becomes, how do we divide our pulse measurement value by the floating-point number 73,746? Another was to divide by 73,746 is to multiply by 0.01356. For new BASIC Stamp users this may seem a dilemma but in fact there is a special operator, **, that allows us to do just that. The ** operator has the effect of multiplying a value by units of 1/65,536. To find the parameter for ** then, we simply multiply 0.01356 by 65,536; the result is 886,688 (we'll round up to 889).

Conversion to centimeters uses the same process and the result of the program is shown below:
File........ Ping Demo.B22
Purpose........ Demo code for Parallax PING()) Sonar Sensor
Author..... Parallax, Inc.
E-mail....... support@parallax.com
Started.....
Updated..... 08 JUN 2002

{/STAMP B22}
{/OBASIC 2.5}

-------------------------------------------------------------

----[ Program Description ]---------------------------------

This program demonstrates the use of the Parallax PING()) sensor and then
converting the raw measurement to English (inches) and Metric (cm) units.

somax Math:

At sea level sound travels through air at 1130 feet per second. This
equals to 1 inch in 73.746 us, or 1 cm in 29.034 us).

Since the PING()) sensor measures the time required for the sound wave to
tavel from the sensor and back. The result -- after conversion to
microseconds for the BASIC Stamp module in use -- is divided by two to
remove the return portion of the echo pulse. The final raw result is
duration from the front of the sensor to the target in microseconds.

----[ I/O Definitions ]--------------------------------------

Ping PIN 15

----[ Constants ]-------------------------------------------

SELECT $STAMP
#CASE B21, B22
 Trigger CON 5 ' trigger pulse = 10 us
 Scale CON $200 ' raw x 2.00 = us
#CASE B21B, B22B
 Trigger CON 13
 Scale CON $0CD ' raw x 0.80 = us
#CASE B21P
 Trigger CON 5
 Scale CON $1B1 ' raw x 1.98 = us
#ENDSELECT

RawToIn CON 989 ' 1 / 73.746 (with **) 
RawTcon CON 2257 ' 1 / 29.034 (with **)

ISHigh CON 1 ' for PULSOUT
ISLow CON 0
'-----[ Variables ]----------------------------------------------------------
rawDist VAR Word         ' raw measurement
inches VAR Word
cm VAR Word

'-----[ Initialization ]------------------------------------------------------
Reset:
DEBUG CLS,
    "Parallax PING))) Sonar", CR,
    "-------------------------", CR,
    CR,
    "Time (US)......", CR,
    "Inches.......", CR,
    "centimeters..."

'-----[ Program Code ]-------------------------------------------------------
Main:
DO
    ODSUB Get Sonar             ' get sensor value
    inches = rawDist ** RawToIn ' convert to inches
    cm = rawDist ** RawToCm     ' convert to centimeters
    DEBUG CR8XY, 15, 3,
        DEC rawDist, CLRPOL,
        CR8XY, 15, 4,
        DEC inches, CLRPOL,
        CR8XY, 15, 5,
        DEC cm, CLRPOL
    PAUSE 100
LOOP
END

'-----[ Subroutines ]--------------------------------------------------------
This subroutine triggers the PING))) sonar sensor and measures
the echo pulse. The raw value from the sensor is converted to
microseconds based on the stamp module in use. This value is
divided by two to remove the return trip -- the result value is
the distance from the sensor to the target in microseconds.

Get Sonar:
    Ping = isLow                ' make trigger 0-1-0
    PULSOUT Ping, trigger      ' activate sensor
    PULSIN Ping, isHigh, rawDist ' measure echo pulse
    rawdist = rawdist */ scale  ' convert to us
    rawdist = rawdist / 2      ' remove return trip
RETURN
---{ Program Description }---------------------------------------------------

This program demonstrates the use of the Parallax FING))) sensor and then
converting the raw measurement to English (inches) and Metric (cm) units.

Sonar Math:

At sea level sound travels through air at 1130 feet per second. This
equals to 1 inch in 73.746 usec, or 1 cm in 29.034 usec).

Since the FING))) sensor measures the time required for the sound wave to
travel from the sensor and back. The result -- after conversion to
microseconds for the BASIC Stamp module in use -- is divided by two to
remove the return portion of the echo pulse. The final raw result is
the duration from the front of the sensor to the target in microseconds.

---{ I/O Definitions }--------------------------------------------------------

SYMBOL Ping        - 7

---{ Constants }-------------------------------------------------------------

SYMBOL Trigger     - 1          ' 10 uS trigger pulse
SYMBOL Scale       - 10         ' raw X 10.00 = uS
SYMBOL RawToIn     - 989        ' 1 / 73.746 (with **)  
SYMBOL RawToCm     - 1257       ' 1 / 29.034 (with **)  
SYMBOL IsHigh      - 1          ' for FULSOUT
SYMBOL IsLow       - 0

---{ Variables }------------------------------------------------------------

SYMBOL rawDist     - W1        ' raw measurement
SYMBOL inches      - W2
SYMBOL cm          - W3
Main:
GOSUB Get Sonar          ' get sensor value
inches = rawDist ** RawToIn    ' convert to inches
cm = rawDist ** RawToCm      ' convert to centimeters

DEBUG CLS               ' report
DEBUG "Time (us)..... ", #rawDist, CR
DEBUG "Inches......... ", #inches, CR
DEBUG "Centimeters... ", #cm

PAUSE 200
GOTO Main

END

---[ Subroutines ]-----------------------------------------------

' This subroutine triggers the PING))) sonar sensor and measures
' the echo pulse. The raw value from the sensor is converted to
' microseconds based on the Stamp module in use. This value is
' divided by two to remove the return trip -- the result value is
' the distance from the sensor to the target in microseconds.

Get Sonar:
LOW Ping                 ' make trigger 0-1-0
PULSOUT Ping, Trigger   ' activate sensor
PULSIN Ping, IsHigh, rawDist ' measure echo pulse
rawDist = rawDist * Scale   ' convert to US
rawDist = rawDist / 2       ' remove return trip
RETURN
Program Example: Javelin Stamp Microcontroller

This class file implements several methods for using the PING))) sensor:

```java
package stamp.peripheral.sensor;
import stamp.core.*;

/**
 * This class provides an interface to the Parallax PING))) ultrasonic
 * range finder module.
 * <p>
 * @usage: <code>
 * Ping range = new Ping(CPU.pin0); // trigger and echo on P0
 * </code>
 * @code>
 * <p>
 * Detailed documentation for the PING))) Sensor can be found at:<br>
 * http://www.parallax.com/detail.asp?product id=28015
 * <p>
 * @version 1.0 01 FEB 2005
 * /
 * public final class Ping {

 private int i0Pin;

 /**
 * Creates PING))) range finder object
 * @param i0Pin PING))) trigger and echo return pin
 */
 public Ping (int i0Pin) {
  this.i0Pin = i0Pin;
 }

 /**
 * Returns raw distance value from the PING))) sensor.
 */
 public int getRaw() {

  int echoRaw = 0;
  CPU.writePin(i0Pin, false); // setup for high-going pulse
  CPU.pulseOut(1, i0Pin); // send trigger pulse
  echoRaw = CPU.pulseIn(2171, i0Pin, true); // measure echo return

  // return echo pulses if in range; zero if out-of-range
  return (echoRaw < 2131) ? echoRaw : 0;
 }
```
/* The PING() returns a pulse width of 73.748 µs per inch. Since the
 * Javelin pulseIn() round-trip echo time is in 6.69 µs units, this is the
 * same as a one-way trip in 1.34 µs units. Dividing 73.748 by 1.34 we
 * get a time-per-inch conversion factor of 15.9522 (≈ 0.058851). */
 * Values to derive conversion factors are selected to prevent roll-over
 * past the 15-bit positive values of Javelin Stamp integers.
 */

/** *
 * @return PING() distance value in inches
 */
public int getIn() {
    return (getRaw() * 1 / 51); // raw * 0.058824
}

/** *
 * @return PING() distance value in tenths of inches
 */
public int getIn10() {
    return (getRaw() * 1 / 5); // raw / 1.6667
}

/** *
 * The PING() returns a pulse width of 29.013 µs per centimeter. As the
 * Javelin pulseIn() round-trip echo time is in 6.69 µs units, this is the
 * same as a one-way trip in 1.34 µs units. Dividing 29.013 by 1.34 we
 * get a time-per-centimeter conversion factor of 6.6695.
 * Values to derive conversion factors are selected to prevent roll-over
 * past the 15-bit positive values of Javelin Stamp integers.
 */

/** *
 * @return PING() distance value in centimeters
 */
public int getCM() {
    return (getRaw() * 1 / 10); // raw / 6.6667
}

/** *
 * @return PING() distance value in millimeters
 */
public int getMm() {
    return (getRaw() * 1 / 2); // raw / 0.6667
}

This simple demo illustrates the use of the PING() ultrasonic range finder class with the Javelin Stamp.
import stamp.core.*;
import stamp.peripheral.sensor.Ping;

public class testPing {
    public static final char HOME = 0x01;

    public static void main() {
        Ping range = new Ping(CPU.pin0);
        StringBuffer msg = new StringBuffer();
        int distance;
        while (true) {
            // measure distance to target in inches
            distance = range.getIn();
            // create and display measurement message
            msg.clear();
            msg.append(HOME);
            msg.append(distance);
            msg.append(" \n ");
            System.out.print(msg.toString());
            // wait 0.5 seconds between readings
            CPU.delay(5000);
        }
    }
}