Car_HC11.c

```
//////////////////////////////////////////////////////
// Name:    Roger Grayson
// Date:    Apr 2003
// Purp:    Distance Sensor
//              This program uses the SCI subsystem of the
//              HC11 to capture the data from the distance
//              sensor.  The output of the distance sensor
//              is:
//              Sync Byte: 0x55
//              LSB of distance data
//              Sync Byte: 0x55
//              MSB of distance data
//
//              The distance sensor updates every 50ms.
//              For testing purposes, the output of the
//              distance sensor is shown on the screen,
//              through ICC11 or HyperTerminal.
//////////////////////////////////////////////////////
#include <hc11.h>

//////////////////////////////////////////////////////
//    Function Prototypes
//////////////////////////////////////////////////////
void SCREEN_OUT(unsigned short value);
void SCREEN_OUT(unsigned short value);
void DELAY(unsigned short millisecond);

//////////////////////////////////////////////////////
//    Global Definitions
//
//    LEDs are memory mapped to address 0x8000
//    DIPs are memory mapped to address 0x9000
//    Change these accordingly
//////////////////////////////////////////////////////
#define LEDS *(unsigned char *)  0x8000
#define DIPS *(unsigned char *)  0x9000

//////////////////////////////////////////////////////
//-----MAIN-----//
//////////////////////////////////////////////////////
int main(){
    unsigned char temp;
    unsigned short value=0;

    DELAY(5000);      //5 seconds to plug the distance
                      //sensor into the Max232 chip

    while(1){
        temp = 0x55;
        while (temp == 0x55){    //check for sinc byte
              temp=read_sci();   //once it is received,
        }
```

```
        value = temp;             //store the next byte
        temp = 0x55;
        while (temp == 0x55) {    //and check for another
            temp=read_sci();      //sync byte
        }
        value += (temp<<4);       //after receiving the
                                  //second sinc byte, shift
                                  //the next value left 4
                                  //times, because it is the
                                  //MSB of the data, and add
                                  //it to the LSB of the data
        SCREEN_OUT(value);        //Output the distance value
                                  //to the screen


        if (value < 900) PORTA = 0x60;        //too_close
                                  //signal is PORTA.6
        else if (value < 2700) PORTA = 0x20;      //close
                                  //signal is PORTA.5
        else PORTA = 0x00;

    };

    return 0;
}

//---------------------------------------------------------------------
//-- Name: SCREEN_OUT
//-- Purp: outputs passed value to hyperterminal via inline assembly
//---------------------------------------------------------------------
void SCREEN_OUT(unsigned short value)
{
    unsigned char d4,d3,d2,d1,d0;
    d4=value/10000;
    value=value-d4*10000;
    d3=value/1000;
    value=value-d3*1000;
    d2=value/100;
    value=value-d2*100;
    d1=value/10;
    d0=value-d1*10;
    asm("psha");
    asm("pshb")
    asm("pshx");
    asm("ldaa %d4");
    asm("adda #$30");
    asm("jsr $FFB8"); // JMP OUTA - Output ASCII character
    asm("ldaa %d3");
    asm("adda #$30");
    asm("jsr $FFB8"); // JMP OUTA - Output ASCII character
    asm("ldaa %d2");
    asm("adda #$30");
```

```
    asm("jsr $FFB8"); // JMP OUTA - Output ASCII character
    asm("ldaa %d1");
    asm("adda #$30");
    asm("jsr $FFB8"); // JMP OUTA - Output ASCII character
    asm("ldaa %d0");
    asm("adda #$30");
    asm("jsr $FFB8"); // JMP OUTA - Output ASCII character
    asm("jsr $FFC4"); // JMP OUTCRLF - Output ASCII carriage return
    asm("pulx");
    asm("pulb");
    asm("pula");
}
//---------------------------------------------------------------------
//-- Name: DELAY
//-- Purp: creates a delay of the passed value in milliseconds
//---------------------------------------------------------------------
void DELAY(unsigned short millisecond)
{
    unsigned short i,j;
    for (i=0; i<millisecond; i++)
        for (j=0; j<70; j++);
}
```