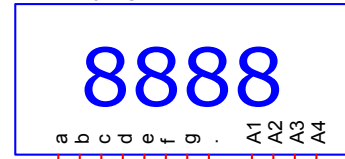


Revision 0.2:
 (1) Rectify error on D+/D- mixed up
 (2) Connected UVss to ground.
 (3) Rectify T2-T5 connection.

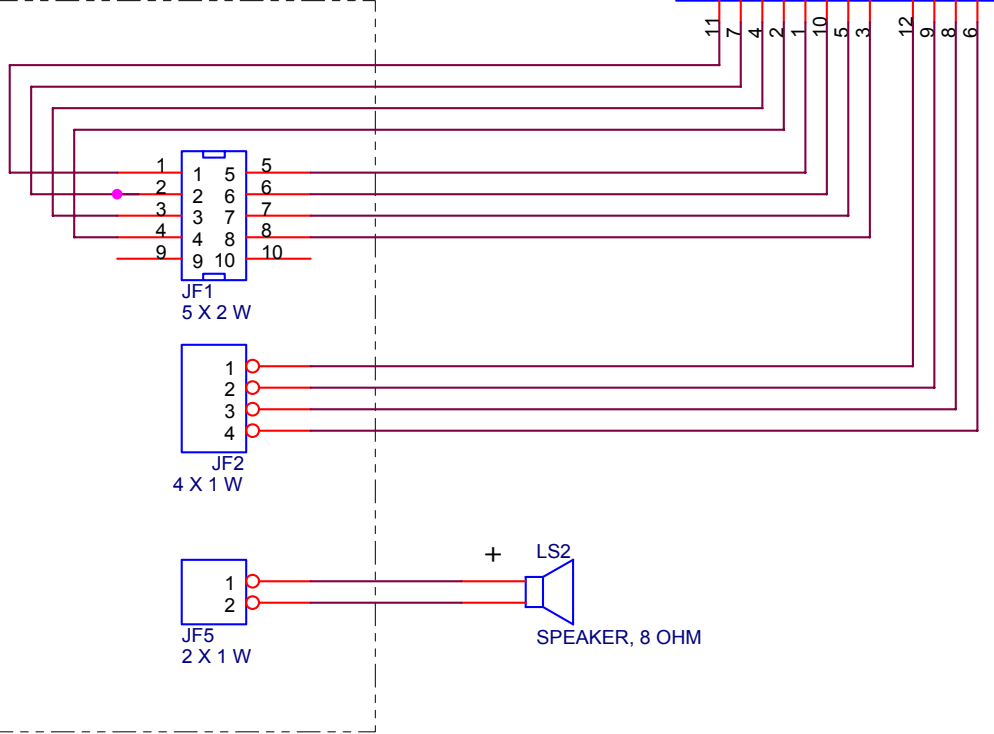
Title		
BASIC AT89C5131 TRAINER BOARD		
Size	Document Number	Rev
B	BASIC_89C5131_LEONGKJ	0.2
Date:	Wednesday, February 13, 2008	Sheet 1 of 1

4-DIGIT 7-SEGMENT COMMON ANODE LED

PLTF-4401BS

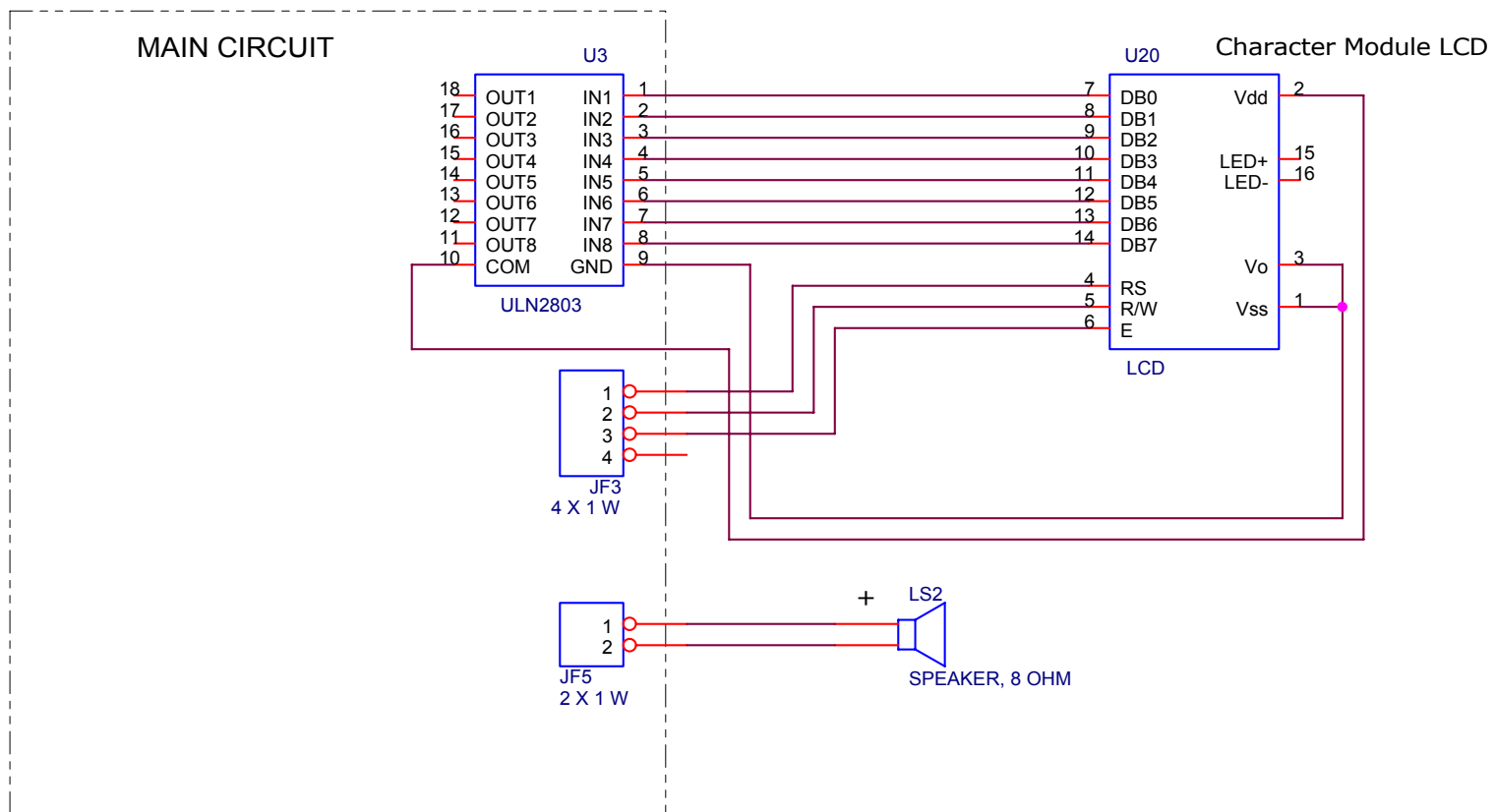


MAIN CIRCUIT



Title		
AT89C5131 CIRCUIT A APPLICATION 1 CONNECTION		
Size	Document Number	Rev
A	CIRCUIT_A_APPLICATION_LEONGKJ	0
Date:	Tuesday, February 19, 2008	Sheet 1 of 1

** Do not solder the ULN2803 IC on U3
but uses U3 as a point to connect to
your LCD.*



Title		
AT89C5131 CIRCUIT A APPLICATION 2 CONNECTION		
Size	Document Number	Rev
A	CIRCUIT_A_APPLICATION_LEONGKJ	0
Date:	Sunday, April 13, 2008	Sheet 1 of 1

BILL OF MATERIAL

Vendor Part No:	Vendor	DESCRIPTION (Main Board)	QTY
MC222	Sunlight	Capacitor 0.0022UF 50V Multilayer	1
MC103	Sunlight	Capacitor 0.01UF 50V Multilayer	1
MC104	Sunlight	Capacitor 0.1UF 50V Multilayer	2
MC105	Sunlight	Capacitor 1UF 50V Multilayer	1
MC030	Sunlight	Capacitor 30PF 50V Multilayer	2
EC476/25	Sunlight	Capacitor Electrolytic 47uF 25V	1
C2044	Sunlight	Crystal 24MHz	1
1N4001	Sunlight	Diode 1A 50V	4
HD1X40	Sunlight	Header 6X3mm(40 PIN)	1
LM7805CT	Sunlight	IC REG 1A POS 0-125DEG C TO-220	1
L3-R	Sunlight	LED 3MM RED	1
1095721	Farnell	Microcontroller Atmel AT89C5131A-S3SUM	1
PLD-52	Sunlight	52 PIN PLCC IC SOCKET THROUGH-HOLE MOUNTING	1
	Amicus	PCB	1
B100R-0.255	Sunlight	Resistor 1/4W 100 OHM	1
D10K-0.255	Sunlight	Resistor 1/4W 10K OHM	1
C2.2K-0.255	Sunlight	Resistor 1/4W 2.2K OHM	1
A27R-0.255	Sunlight	Resistor 1/4W 27 OHM	1
B330R-0.255	Sunlight	Resistor 1/4W 330 OHM 5%	1
O9-A4-7K	Sunlight	Resistor Array 4.7K OHM 9 PIN 8	1
DK-025SP	Sunlight	Socket Dia. 2.5mm PCB MOUNTED DC SOCKET	1
PTA-144A	Sunlight	Switch 4 PIN TACT SWITCH MINI H : 5MM MCDTS6-1K	2
9901671	Farnell	Switch Omron Micro D2FL	1
1076665	Farnell	Connector USB Type B Tyco 292304-1	1
Vendor Part No:	Vendor	DESCRIPTION (4-digit Display)	QTY
2N3906	Sunlight	Transistor 2N3906 PNP -60V -200MA TO-92	4
PLTF-4401BS	Sunlight	LED 4-digit 7-segment LED display PLTF-4401BS	1
ULN2003	Sunlight	Transistor Array	1
TPWW-32SIL	Sunlight	Connector 32 PIN TURN PIN W/W SOCKET SIL	1
ICSKT-318	Sunlight	Socket 18 PIN NORMAL IC SOCKET DIL 300MIL	1
D10K-0.255	Sunlight	Resistor 1/4W 10K OHM	4
PTA-144A	Sunlight	Switch 4 PIN TACT SWITCH MINI H : 5MM MCDTS6-1K	4
B330R-0.255	Sunlight	Resistor 1/4W 330 OHM 5%	7
Vendor Part No:	Vendor	DESCRIPTION (Remaining Components)	QTY
2337265	Farnell	Transistor Darlington MPSA14	1
D10K-0.255	Sunlight	Resistor 1/4W 10K OHM	1
A27R-0.255	Sunlight	Resistor 1/4W 27 OHM	1
PTA-144A	Sunlight	Switch 4 PIN TACT SWITCH MINI H : 5MM MCDTS6-1K	5
9708065	Farnell	Opto-coupler Sharp GP1S53VJ000F	2
D10K-0.255	Sunlight	Resistor 1/4W 10K OHM	2
B330R-0.255	Sunlight	Resistor 1/4W 330 OHM 5%	2
L3-OR	Sunlight	LED 3MM Orange	1
L3-GR	Sunlight	LED 3MM Green	1
B330R-0.255	Sunlight	Resistor 1/4W 330 OHM 5%	2

```

/*****
Program Name: buzzer.C

Function: Sounding the speaker

Hardware Requirement: Speaker circuit

Feb 2008, LeongKJ

*****/

#include <REG51KJ.H>

#define PB1    P33
#define SPK    P37

void beep(){
/*-----
    Beep Speaker.
-----*/
    unsigned char x,y;

    for(x=0;x<100;x++){
        SPK = !SPK;
        for(y=0;y<100;y++){
        }
        SPK=0;
    }

void main (void) {
/*-----
    Main Program
-----*/

    PB1 = 1;    // Define Input Port

    SPK = 0;    // OFF SPEAKER

    while (1) {
        if (PB1==0) beep();
    }
}

```

```

/*****
Filename: cLock.c

Function: Clock program for EG2155 AT89C5131 board.

Hardware Requirement: 4-Digit 7-Segment LED

Feb 2008, LeongKJ

*****/

#include <REG51KJ.h>

#define PB1    P33
#define PB2    P10
#define minSet PB2
#define hrSet  PB1

// time variables
unsigned char count=0, sec=0,min=0,hr=12;
bit timeToggle=0;
unsigned char digitControl=1;

/*-----
    Delay Subroutine
-----*/
void delay (unsigned int duration)
{
    while (duration--!=0);
}

unsigned char lut(unsigned char num){
/*-----
    Look Up Table
-----*/
    unsigned char tmp;
    switch(num){
        case 0: tmp=0x3f;break;
        case 1: tmp=0x06;break;
        case 2: tmp=0x5b;break;
        case 3: tmp=0x4f;break;
        case 4: tmp=0x66;break;
        case 5: tmp=0x6d;break;
        case 6: tmp=0x7d;break;
        case 7: tmp=0x27;break;
        case 8: tmp=0x7f;break;
        case 9: tmp=0x6f;break;
    }
    return(tmp);
}

/*-----
    Mux Display Control
-----*/
void muxDisplay(){
    switch(digitControl){
        case 1:
            P0 = lut(hr/10);
            P2 = 0x0e;
            break;
        case 2:
            P0 = lut(hr%10);
            P2 = 0x0d;
            if (count>100==0) P07=0;    // sec flasher
            else P07=1;
    }
}

```

```

        break;
    case 3:
        P0 = lut(min/10);
        P2 = 0x0b;
        break;
    case 4:
        P0 = lut(min%10);
        P2 = 0x07;
        break;
    }
    if (++digitControl>4) digitControl = 1;
}

/*-----
Timer Interrupt Routine

Function: Time keeping function. This routine
is executed every 5 ms (based on 24MHz crystal).
-----*/
void timekeeper(void) interrupt 1
{
    TF0=0; // Reset Interrupt Flag

    TH0=0xd8; // Next overflow in 5 ms
    TL0=0xf0;

    count++;
    muxDisplay();
    if (count==200){ // 1 sec elapsed.
        count=0;
        timeToggle=!timeToggle;
        sec++;
        if (sec==60){
            sec=0;
            min++;
            if (min==60){
                hr++;
                min=0;
                if (hr==24){
                    hr=0;
                }
            }
        }
    }
}

/*-----
Enable Timer 0 interrupt.
-----*/
void timer0_init()
{
    ET0=1; // Interrupt Enabled.
    TMOD = 0x01; // Configure for 16-bit mode.
    TR0=1; // Run timer.
}

/*-----
Hour/Min Setting
-----*/
void buttonTrace()
{
    if (minSet==0){

```

```

        min++;
        if (min>59) min=0;
        delay(20000); // Debounce
    }
    if (hrSet==0){
        hr++;
        if (hr>23) hr=0;
        delay(20000); // Debounce
    }
    timeToggle=1;
}

//----- MAIN PROGRAM -----
void main(void)
{
    minSet = 1; // Set Port as Input
    hrSet = 1;

    EA=1; // Master Interrupt Enabled.
    timer0_init();

    while(1){
        buttonTrace();
    }
}

```

```

/*****
Filename: timer.c

Function: Timer program for EG2155 AT89C5131 board.

Hardware Requirement: 4-Digit 7-Segment LED

Description: 90 sec timer with resolution of 0.01 s
and 3 buttons for start/stop/reset control.

Feb 2008, LeongKJ

*****/
#include <REG51KJ.h>

#define PB1          P33
#define PB2          P10
#define PB3          P12
#define stopBtn     PB2
#define startBtn    PB1
#define resetBtn    PB3
#define STOP        0
#define START       1
#define FULL        3

// time variables
unsigned char count=0, subSec=0, sec=0, mode=STOP;
bit timeToggle=0;
unsigned char digitControl=1;

/*-----
Delay Subroutine
-----*/
void delay (unsigned int duration)
{
    while (duration--!=0);
}

unsigned char lut(unsigned char num){
/*-----
Look Up Table
-----*/
unsigned char tmp;
switch(num){
    case 0: tmp=0x3f;break;
    case 1: tmp=0x06;break;
    case 2: tmp=0x5b;break;
    case 3: tmp=0x4f;break;
    case 4: tmp=0x66;break;
    case 5: tmp=0x6d;break;
    case 6: tmp=0x7d;break;
    case 7: tmp=0x27;break;
    case 8: tmp=0x7f;break;
    case 9: tmp=0x6f;break;
    }
return(tmp);
}

/*-----
Mux Display Control
-----*/
void muxDisplay(){
    switch(digitControl){

```

```

    case 1:
        P0 = lut(sec/10);
        P2 = 0x0e;
        break;
    case 2:
        P0 = lut(sec%10);
        P2 = 0x0d;
        P07 = 1;
        break;
    case 3:
        P0 = lut(subSec/10);
        P2 = 0x0b;
        break;
    case 4:
        P0 = lut(subSec%10);
        P2 = 0x07;
        break;
    }
    if (++digitControl>4) digitControl = 1;
}

/*-----
Timer Interrupt Routine
-----
Function: Time keeping function. This routine
is executed every 1 ms (based on 24MHz crystal).
-----*/
void timekeeper(void) interrupt 1
{
    TF0=0; // Reset Interrupt Flag

    TH0=0xf8; // Next overflow in 1 ms
    TL0=0x30;

    muxDisplay();
    if (mode==START) count++;
    if (count==10){ // 0.01 sec elapsed.
        count=0;
        subSec++;
        if (subSec==100){
            subSec=0;
            sec++;
            if (sec==90){
                mode=FULL;
            }
        }
    }
}

/*-----
Enable Timer 0 interrupt.
-----*/
void timer0_init()
{
    ET0=1; // Interrupt Enabled.
    TMOD = 0x01; // Configure for 16-bit mode.
    TR0=1; // Run timer.
}

/*-----
Timer START/STOP control
-----*/

```

```

void buttonTrace()
{
    if ((startBtn==0)&&(mode!=FULL)){
        mode=START;
        delay(20000);        // Debounce
    }
    if ((stopBtn==0)&&(mode!=FULL)){
        mode=STOP;
        delay(20000);
    }
    if (resetBtn==0){
        if (mode==FULL) mode=STOP;
        subSec=0;
        sec=0;
        delay(20000);
    }
}

//----- MAIN PROGRAM -----
void main(void)
{
    startBtn = 1;        // Set Port as Input
    stopBtn = 1;
    resetBtn = 1;

    EA=1;                // Master Interrupt Enabled.
    timer0_init();

    while(1){
        buttonTrace();
    }
}

```

```

/*****
Filename: downtimer.c

Function: Count Down Timer program for
EG2155 AT89C5131 board.

Hardware Requirement: 4-Digit 7-Segment LED

Feb 2008, LeongKJ

*****/
#include <REG51KJ.h>

#define PB1        P33
#define PB2        P10
#define PB3        P12
#define secSet     PB2
#define minSet     PB1
#define startSet   PB3
#define STOP       0
#define START      1
#define LED1       P35

// time variables
unsigned char count=0;
char sec=0,min=0;
unsigned char mode=STOP;
unsigned char digitControl=1;

/*-----
Delay Subroutine
-----*/
void delay (unsigned int duration)
{
    while (duration--!=0);
}

unsigned char lut(unsigned char num){
/*-----
Look Up Table
-----*/
    unsigned char tmp;
    switch(num){
        case 0: tmp=0x3f;break;
        case 1: tmp=0x06;break;
        case 2: tmp=0x5b;break;
        case 3: tmp=0x4f;break;
        case 4: tmp=0x66;break;
        case 5: tmp=0x6d;break;
        case 6: tmp=0x7d;break;
        case 7: tmp=0x27;break;
        case 8: tmp=0x7f;break;
        case 9: tmp=0x6f;break;
    }
    return(tmp);
}

/*-----
Mux Display Control
-----*/
void muxDisplay(){
    switch(digitControl){
        case 1:
            P0 = lut(min/10);

```

```

        P2 = 0x0e;
        break;
    case 2:
        P0 = lut(min%10);
        P2 = 0x0d;
        P07 = 1;
        break;
    case 3:
        P0 = lut(sec/10);
        P2 = 0x0b;
        break;
    case 4:
        P0 = lut(sec%10);
        P2 = 0x07;
        break;
    }
    if (++digitControl>4) digitControl = 1;
}

/*-----
Timer Interrupt Routine

Function: Time keeping function. This routine
is executed every 5 ms (based on 24MHz crystal).
-----*/
void timekeeper(void) interrupt 1
{
    TF0=0; // Reset Interrupt Flag

    TH0=0xd8; // Next overflow in 5 ms
    TL0=0xf0;

    muxDisplay();
    if (mode==START) {
        count++;
        if (count%50==0) LED1=!LED1;
    }
    if (count==200){ // 1 sec elapsed.
        count=0;
        sec--;
        if (sec<0){
            sec=59;
            min--;
            if (min<0){
                min=0;
                sec=0;
                mode=STOP;
                LED1=1;
            }
        }
    }
}

/*-----
Enable Timer 0 interrupt.
-----*/
void timer0_init()
{
    ET0=1; // Interrupt Enabled.
    TMOD = 0x01; // Configure for 16-bit mode.
    TR0=1; // Run timer.
}

```

```

/*-----
Min/Sec/Start Setting
-----*/
void buttonTrace()
{
    if (secSet==0){
        sec++;
        if (sec>59) sec=0;
        delay(20000); // Debounce
    }
    if (minSet==0){
        min++;
        if (min>90) min=90;
        delay(20000); // Debounce
    }
    if (startSet==0){
        if ((min>0)|| (sec>0)){
            mode=START;
        }
        delay(20000);
    }
}

//----- MAIN PROGRAM -----
void main(void)
{
    secSet = 1; // Set Port as Input
    minSet = 1;
    startSet = 1;

    EA=1; // Master Interrupt Enabled.
    timer0_init();

    while(1){
        buttonTrace();
    }
}

```

```

/*****
Filename: carpark.c

Function: Carpark management program for
EG2155 AT89C5131 board.

Hardware Requirement:
(1) 4-Digit 7-Segment LED
(2) Optical Slot Sensors x 2
(3) LEDs x 2

Description: Using slot sensor to simulate detection
of cars going and coming out of car park. Two LEDs
simulate the barrier gate control.

Feb 2008, LeongKJ

*****/
#include <REG51KJ.h>

#define IS01    P32
#define IS02    P34
#define LED1    P35
#define LED2    P36

#define CAR_IN    IS01
#define CAR_OUT   IS02
#define GANTRY_IN LED1
#define GANTRY_OUT LED2

unsigned char freeSpace=99;    // Start with free 99 lots.
bit inSensorBlocked=0;
bit outSensorBlocked=0;
unsigned char digitControl=1;

unsigned char lut(unsigned char num){
/*-----
Look Up Table
-----*/
unsigned char tmp;
switch(num){
case 0: tmp=0x3f;break;
case 1: tmp=0x06;break;
case 2: tmp=0x5b;break;
case 3: tmp=0x4f;break;
case 4: tmp=0x66;break;
case 5: tmp=0x6d;break;
case 6: tmp=0x7d;break;
case 7: tmp=0x27;break;
case 8: tmp=0x7f;break;
case 9: tmp=0x6f;break;
}
return(tmp);
}

/*-----
Mux Display Control
-----*/
void muxDisplay(){
switch(digitControl){
case 1:
P0 = lut(freeSpace/10);
P2 = 0x0e;
break;

```

```

case 2:
P0 = lut(freeSpace%10);
P2 = 0x0d;
P07 = 1;
break;
}
if (++digitControl>2) digitControl = 1;
}

/*-----
Timer Interrupt Routine
-----
Function: Provide multiplexing timing. It
is executed every 1 ms (based on 24MHz crystal).
-----*/
void timekeeper(void) interrupt 1
{
TF0=0;    // Reset Interupt Flag

TH0=0xf8;    // Next overflow in 1 ms
TL0=0x30;

muxDisplay();
}

/*-----
Enable Timer 0 interrupt.
-----*/
void timer0_init()
{
ET0=1;    // Interrupt Enabled.
TMOD = 0x01;    // Configure for 16-bit mode.
TR0=1;    // Run timer.
}

/*-----
Track incoming/outgoing cars
-----*/
void sensorTrack()
{
if (CAR_IN==1) {
inSensorBlocked=1;
if (freeSpace!=0) GANTRY_IN=0;    // Open barrier.
}

if (CAR_OUT==1) {
outSensorBlocked=1;
GANTRY_OUT=0;    // Open barrier.
}

if ((CAR_IN==0)&&(inSensorBlocked==1)){
// Incoming car detected.
inSensorBlocked = 0;
if (freeSpace!=0){
freeSpace--;
GANTRY_IN = 1;    // Close barrier.
}
}

if ((CAR_OUT==0)&&(outSensorBlocked==1)){
// Outgoing car detected.
outSensorBlocked = 0;
}
}

```

```

        GANTRY_OUT = 1;    // Close barrier.
        if (freeSpace!=99){
            freeSpace++;
        }
    }

}

//----- MAIN PROGRAM -----
void main(void)
{
    CAR_IN = 1;    // Set Port as Input
    CAR_OUT = 1;

    GANTRY_IN = 1; // Close barrier.
    GANTRY_OUT = 1;

    EA=1;        // Master Interrupt Enabled.
    timer0_init();

    while(1){
        sensorTrack();
    }
}

```

```

/*****
Filename: 4d.c

Function: 4D Generator program for EG2155 AT89C5131 board.

Hardware Requirement: 4-Digit 7-Segment LED

Description:

Feb 2008, LeongKJ

*****/
#include <REG51KJ.h>

#define PB1      P33
#define startBtn  PB1

#define START    1
#define STOP     0

unsigned char highNum,lowNum;
unsigned char digitControl=1;

/*-----
Delay Subroutine
-----*/
void delay (unsigned int duration)
{
    while (duration--!=0);
}

unsigned char lut(unsigned char num){
/*-----
Look Up Table
-----*/
unsigned char tmp;
switch(num){
    case 0: tmp=0x3f;break;
    case 1: tmp=0x06;break;
    case 2: tmp=0x5b;break;
    case 3: tmp=0x4f;break;
    case 4: tmp=0x66;break;
    case 5: tmp=0x6d;break;
    case 6: tmp=0x7d;break;
    case 7: tmp=0x27;break;
    case 8: tmp=0x7f;break;
    case 9: tmp=0x6f;break;
    }
return(tmp);
}

/*-----
Mux Display Control
-----*/
void muxDisplay(){

    switch(digitControl){
        case 1:
            P0 = lut(highNum/10);
            P2 = 0x0e;
            break;
        case 2:
            P0 = lut(highNum%10);
            P2 = 0x0d;

```

```

        break;
    case 3:
        P0 = lut(lowNum/10);
        P2 = 0x0b;
        break;
    case 4:
        P0 = lut(lowNum%10);
        P2 = 0x07;
        break;
    }
    if (++digitControl>4) digitControl = 1;
}

/*-----
Timer Interrupt Routine

Function: Multiplex display timing control. It
is executed every 1 ms (based on 24MHz crystal).
-----*/
void timekeeper(void) interrupt 1
{
    TF0=0; // Reset Interrupt Flag

    TH0=0xf8; // Next overflow in 1 ms
    TL0=0x30;

    muxDisplay();
}

/*-----
Enable Timer 0 interrupt.
-----*/
void timer0_init()
{
    ET0=1; // Interrupt Enabled.
    TMOD = 0x11; // Configure for 16-bit mode.
    TR0=1; // Run timer.
    TR1=1;
}

/*-----
Grab Random Number from Timer 1
-----*/
void buttonTrace()
{
    if (startBtn==0){
        highNum = TH1%100;
        lowNum = TL1%100;
        delay(20000); // Debounce
    }
}

//----- MAIN PROGRAM -----
void main(void)
{
    startBtn = 1; // Set Port as Input

    EA=1; // Master Interrupt Enabled.
    timer0_init();

```

```

while(1){
    buttonTrace();
}
}

```

```

/*-----
REG51.H

Header file for generic 80C51 and 80C31 microcontroller.
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
-----
*/

#ifndef __REG51_H__
#define __REG51_H__

/* BYTE Register */
sfr P0 = 0x80;
sfr P1 = 0x90;
sfr P2 = 0xA0;
sfr P3 = 0xB0;
sfr PSW = 0xD0;
sfr ACC = 0xE0;
sfr B = 0xF0;
sfr SP = 0x81;
sfr DPL = 0x82;
sfr DPH = 0x83;
sfr PCON = 0x87;
sfr TCON = 0x88;
sfr TMOD = 0x89;
sfr TL0 = 0x8A;
sfr TL1 = 0x8B;
sfr TH0 = 0x8C;
sfr TH1 = 0x8D;
sfr IE = 0xA8;
sfr IP = 0xB8;
sfr SCON = 0x98;
sfr SBUF = 0x99;

/* BIT Register */
/* PSW */
sbit CY = 0xD7;
sbit AC = 0xD6;
sbit F0 = 0xD5;
sbit RS1 = 0xD4;
sbit RS0 = 0xD3;
sbit OV = 0xD2;
sbit P = 0xD0;

/* TCON */
sbit TF1 = 0x8F;
sbit TR1 = 0x8E;
sbit TF0 = 0x8D;
sbit TR0 = 0x8C;
sbit IE1 = 0x8B;
sbit IT1 = 0x8A;
sbit IE0 = 0x89;
sbit IT0 = 0x88;

/* IE */
sbit EA = 0xAF;
sbit ES = 0xAC;
sbit ET1 = 0xAB;
sbit EX1 = 0xAA;
sbit ET0 = 0xA9;
sbit EX0 = 0xA8;

/* IP */
sbit PS = 0xBC;

```

```

sbit PT1 = 0xBB;
sbit PX1 = 0xBA;
sbit PT0 = 0xB9;
sbit PX0 = 0xB8;

/* P3 */
sbit RD = 0xB7;
sbit WR = 0xB6;
sbit T1 = 0xB5;
sbit T0 = 0xB4;
sbit INT1 = 0xB3;
sbit INT0 = 0xB2;
sbit TXD = 0xB1;
sbit RXD = 0xB0;

/* SCON */
sbit SM0 = 0x9F;
sbit SM1 = 0x9E;
sbit SM2 = 0x9D;
sbit REN = 0x9C;
sbit TB8 = 0x9B;
sbit RB8 = 0x9A;
sbit TI = 0x99;
sbit RI = 0x98;

sbit P00 = 0x80;
sbit P01 = 0x81;
sbit P02 = 0x82;
sbit P03 = 0x83;
sbit P04 = 0x84;
sbit P05 = 0x85;
sbit P06 = 0x86;
sbit P07 = 0x87;

sbit P10 = 0x90;
sbit P11 = 0x91;
sbit P12 = 0x92;
sbit P13 = 0x93;
sbit P14 = 0x94;
sbit P15 = 0x95;
sbit P16 = 0x96;
sbit P17 = 0x97;

sbit P20 = 0xA0;
sbit P21 = 0xA1;
sbit P22 = 0xA2;
sbit P23 = 0xA3;
sbit P24 = 0xA4;
sbit P25 = 0xA5;
sbit P26 = 0xA6;
sbit P27 = 0xA7;

sbit P30 = 0xB0;
sbit P31 = 0xB1;
sbit P32 = 0xB2;
sbit P33 = 0xB3;
sbit P34 = 0xB4;
sbit P35 = 0xB5;
sbit P36 = 0xB6;
sbit P37 = 0xB7;

#endif

```