

```

// 04-04-2011 Leg-driver for HI6SIM demo moving platform last unit.
//
// Driver leg adress, adress in/out daisy chain last aqdress display and check,
// usart request position input from:
//     for Ians (BFF)6dof code.
//
// By Douwe Jippes d.jippes@hccnet.nl www.xs4all.nl/~jippes/hi6sim

//#define _LEGACY_HEADERS
#include <pic.h>
#include <pic16f630.h>
#include <htc.h>
#include <stdlib.h>

__CONFIG(MCLRE_ON & CP_OFF & WDTE_OFF & BOREN_ON & FOSC_INTRCIO);

/* Set up the configuration bits */

// Limmits & contants

#define F_OSC 30000000 // 4 Mc
#define K_TMR1 65535 - (F_OSC/4000) // 8 ms (2 setduty between usart reads)
//#define K_TMR1 65535 - (F_OSC/2286) // 14 ms (like read usart cycle BFF)
#define K_Tnos 65535 - (F_OSC/64) // 500 ms
#define K_Tbit 65535 - (F_OSC/32000) // 1 ms
#define K_THbit 65535 - (F_OSC/64000) // 0.5 ms
#define K_Tdisc 65535 - (F_OSC/320) // 100 ms

//-----PORTA CONFIGURATION-----
#define d0 PORTAbits.RA0 //
#define d1 PORTAbits.RA1 //
#define A_In RA2 // AdrIn
#define d3 PORTAbits.RA3 //
#define d4 PORTAbits.RA4 //
#define Test RA5 // Test pin

//-----PORTC CONFIGURATION-----
#define d0 PORTCbits.RC0 // Disp 1
#define d1 PORTCbits.RC1 // Disp 2
#define d2 PORTCbits.RC2 // Disp 4
#define d3 PORTCbits.RC3 // Disp 8
#define DispM 15 // Disp Mask
#define NLE RC4 // Disp !LE
#define Red RC5 // Red led not adress 6

//#define TMR1IF PIR1bits.TMR1IF // TMR1 int flag(sw clear)

#define off 0
#define on 1
;
char prid[] = " By www.xs4all.nl/~jippes/hi6sim v11 A-C";

// Adress of leg driver
unsigned char Address = 0; // Adress of last leg driver
unsigned char AdrInit = 0; // In get adress mode
unsigned char AdrStart = 0; // Adr in started
unsigned char Mask; // adr bit scan mask
unsigned char ConLost = 0; // Connectinon lost
unsigned char i;

```

```

//char Del;
int d;
//char n;

void main (void);
void Init(void);
void Int_High(void);
void DoAdr(void);
void Ain_CHange(void);
void SetDuty(void);
//void Delayd(void);

void main (void)
{
    Init();
    while(1)
    {
        if (Adress == 6)
            Red = off;
        else
        {
            if (Adress)
            {
                Red ^= 1;
                for (d = 0; d < 0x0fff; d++);
            }
            else
                Red = on;
        }
    }
} // End main

void Init (void)
{
    //setup port A
    TRISA = 0x04;           // ra2 adr in
    WPUA2 = 1;             // ra2 weak pul up
    IOCA2 = 1;             // ra2 int on change
    TRISC = 0x00;         // rc output

    PEIE = 1;             // Timer1 int enable
    TMRIIE = 1;           // Enable interrupt on TMR1 overflow
    TMR1ON = 1;           // Timer1 on
//RA5 = 0; // DEBUG 0ff RA5 Test pin*****
    INTEDG = 0;           // falling edge trigger the interrupt
    INTE = 1;             // enable the external interrupt

    Adress = 0;
    PORTC = (Adress | 0x30); // disp adress 0,NLE 1, Red led on
    NLE =0;
    NLE =1;
    AdrInit = on;         // Adres init mode
    if (!A_In)           // Adr in line already low?
        for (d = 0; d < 3270; d++); // First unit on chain delay
    if (!A_In)           // Adr in line low?
    {
        for (d = 0; d < 327; d++); // First unit on chain extra delay
        AdrStart = on; // Adr in line gone low
        Mask = 16; // Adr bits from previous unit in
    }
}

```

```

chain
// First = start
bit(low)(s,8,4,2,1,e)
TMR1 = K_THbit; // chek in the midst of a bit time
// TMR1IF = off;
}
else
{
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
TMR1 = K_Tnos;
}
GIE = 1; // int enable
}

static void interrupt
ISR(void)
{
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
if (TMR1IF) // Timer 1 int
{
TMR1IF = off;
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
if (AdrInit) // Driver adr init
{
RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
DoAdr();
}
else
{
SetDuty(); // PID & Duty
}
}
else
{
if (INTF) // Adress line in change
{
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
INTF = off;
Ain_Change();
}
}
GIE = 1; //
}

void Ain_Change()
{
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
char a = 0;
for (i = 0; i < 10; i++)
{
if (A_In)
a++;
}
if (a > 5) // Adr in high and previous low
{
RA5 = 0; // DEBUG off RA5 Test pin*****
Address = 0; // No adress
PORTC = (Address | 0x30); // disp adress 0,NLE 1, Red led on
NLE = 0;
NLE = 1;
}
}

```

```

        INTEDG = 0;                                // falling edge trigger the
interrupt
        INTE = on;
    }
    if (a < 5)                                     // Adr in low and previous high
    {
RA5 = 1; // DEBUG flip flop RA5 Test pin*****
        AdrInit = on;                             // Adres init mode
        AdrStart = on;                            // Adr in line gone low
        Mask = 16;                                // Adr bits from previous
unit in chain
        TMR1 = K_THbit;                           // chek in the midst of a bit time
        INTE = off;                               // No ext int
    }
}

void DoAdr(void)
{
    if (!AdrStart)                               // Adr sequence started?
    {                                             // No
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
        AdrInit = off;                           // SW's off
        INTE = on;
        TMR1= K_TMR1;                            // PID timer
    }
    else
    {                                             // Adr sequence
//RA5 ^= 1; // DEBUG flip flop RA5 Test pin*****
        if (A_In)                                 // Adr in high
            Address |= Mask;                     // Set bit in Adr
        Mask >>= 1;                               // next lower bit
        if (Mask)                                 // More bits to go
            TMR1 = K_Tbit;
        else
        {
//RA5 = 1; // DEBUG on RA5 Test pin*****
            AdrInit = off;                       // SW's off
            AdrStart = off;
            INTF = off;
            INTE = on;
            INTEDG = 1;                          // raising edge trigger the interrupt
            TMR1 = K_TMR1;                       // PID timer
            PORTC = (Address | 0x30);
            NLE =0;
            NLE =1;
        }
    }
}

void SetDuty(void)
{
    TMR1= K_TMR1;                                // PID timer
} // End SetDuty.....

//void Delayd(void)
//{
//    for (n = Del; n > 0; n--)
//        for (d = 0; d < 327; d++);
//    ;
//}

```