#include "inc/hw\_ints.h"

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/debug.h"

#include "driverlib/fpu.h"

#include "driverlib/gpio.h"

#include "driverlib/interrupt.h"

#include "driverlib/pin\_map.h"

#include "driverlib/rom.h"

#include "driverlib/sysctl.h"

#include "driverlib/uart.h"

#include "driverlib/timer.h"

#include "string.h"

#include "driverlib/adc.h"

#include "uartstdio.h"

#include "lm4f120h5qr.h"

#include "driverlib/qei.h"

#define GPIO\_PA0\_U0RX 0x00000001

#define GPIO\_PA1\_U0TX 0x00000401

#define sensor1 GPIO\_PIN\_5

#define sensor2 GPIO\_PIN\_3

#define sensor3 GPIO\_PIN\_2

#define sensor4 GPIO\_PIN\_2

#define sensor5 GPIO\_PIN\_0

#define sensor6 GPIO\_PIN\_1

#define sensor7 GPIO\_PIN\_4

#define sensor8 GPIO\_PIN\_3

#define sagsay1 GPIO\_PIN\_4//Birinci motor

#define sagsay2 GPIO\_PIN\_2

#define solsay1 GPIO\_PIN\_6//ikinci motor

#define solsay2 GPIO\_PIN\_7

#define GPIO\_PF2\_T1CCP0 0x00050807

#define GPIO\_PF3\_T1CCP1 0x00050C07

#define GPIO\_PF1\_T0CCP1 0x00050407

#define GPIO\_PB6\_T0CCP0 0x00011807

#define GPIO\_PB7\_T0CCP1 0x00011C07

#define GPIO\_PB0\_T2CCP0 0x00010007

#define GPIO\_PB1\_T2CCP1 0x00010407

#define GPIO\_PD7\_CCP1 0x00031C03

#define GPIO\_PB2\_CCP3 0x00010804

#define GPIO\_O\_LOCK 0x00000520 // GPIO Lock

#define GPIO\_LOCK\_KEY\_DD 0x4C4F434B

#define GPIO\_O\_CR 0x00000524

#define GPIO\_PC4\_WT0CCP0 0x00021007

#define GPIO\_PC5\_WT0CCP1 0x00021407

#define GPIO\_PC6\_WT1CCP0 0x00021807

#define GPIO\_PC7\_WT1CCP1 0x00021C07

/////////////////////////////////////////

long sagMotorHizi=46000,solMotorHizi=46000;

///////////////////////////////////////////////

unsigned long dutyCycle3, dutyCycle4,hiz,hiz1;

unsigned char a[10],i,i1=0,u,p,k=0,run=0,run1,a1=0,b1=0;

unsigned long int t,t1,seri,u1,Period,i\_m,sayac1=0,sayac2=0,kont=0,oku\_say,say1=0,say2=0;

unsigned char ch[20];

unsigned long ulADC0\_Value[5],sag\_motor, sol\_motor,ulPeriod=64000;

unsigned long ulADC1\_Value[1];

void \_\_error\_\_(char \*pcFilename, unsigned long ulLine)

{

}

void delay (unsigned long int id)

{

while(id--);

}

void IntGPIOF(void)

{

if (GPIOPinIntStatus(GPIO\_PORTF\_BASE, GPIO\_PIN\_0) & GPIO\_PIN\_0)

{

GPIOPinIntClear(GPIO\_PORTF\_BASE, GPIO\_PIN\_0);

run1=0;

}else{

GPIOPinIntClear(GPIO\_PORTF\_BASE, GPIO\_PIN\_4);

run1=1;

}

}

void dur()

{

TimerMatchSet(WTIMER0\_BASE, TIMER\_A, ulPeriod-2); // Timer 0 Match set

TimerMatchSet(WTIMER1\_BASE, TIMER\_B, ulPeriod-2 ); // Timer 0 Match et

TimerMatchSet(WTIMER0\_BASE, TIMER\_B, ulPeriod-2); // Timer 0 Match set

TimerMatchSet(WTIMER1\_BASE, TIMER\_A, ulPeriod-2 ); // Timer 0 Match et

}

void ileri(long a,long b)

{

a=a;

b=b;

if(a>ulPeriod-2)

a=ulPeriod-2;

if(b>ulPeriod-2)

b=ulPeriod-2;

TimerMatchSet(WTIMER0\_BASE, TIMER\_A, a); // Timer 0 Match set

TimerMatchSet(WTIMER0\_BASE, TIMER\_B, ulPeriod-2); // Timer 0 Match set

TimerMatchSet(WTIMER1\_BASE, TIMER\_A, b); // Timer 0 Match set

TimerMatchSet(WTIMER1\_BASE, TIMER\_B, ulPeriod-2); // Timer 0 Match et

TimerEnable(WTIMER0\_BASE, TIMER\_A);

TimerEnable(WTIMER0\_BASE, TIMER\_B);

TimerEnable(WTIMER1\_BASE, TIMER\_A);

TimerEnable(WTIMER1\_BASE, TIMER\_B);

}

void hiz\_olc1(void)

{

hiz1=SysCtlClockGet()-TimerValueGet(TIMER1\_BASE,TIMER\_A);

TimerLoadSet(TIMER1\_BASE, TIMER\_A, SysCtlClockGet());

}

void hizolc(void)

{

hiz=SysCtlClockGet()-TimerValueGet(TIMER0\_BASE,TIMER\_A);

TimerLoadSet(TIMER0\_BASE, TIMER\_A, SysCtlClockGet());

}

void IntGPIOA(void)

{

if (GPIOPinIntStatus(GPIO\_PORTA\_BASE, sagsay1) & sagsay1)

{

GPIOPinIntClear(GPIO\_PORTA\_BASE, sagsay1);

if(!GPIOPinRead(GPIO\_PORTA\_BASE,sagsay2))

{

sayac1--;

}

else

{

sayac1++;

hizolc();

}

}

if(GPIOPinIntStatus(GPIO\_PORTA\_BASE, sagsay2) & sagsay2 )

{

GPIOPinIntClear(GPIO\_PORTA\_BASE, sagsay2);

if(!GPIOPinRead(GPIO\_PORTA\_BASE,sagsay1))

{

sayac1++;

}

else

{

sayac1--;

}

}

if (GPIOPinIntStatus(GPIO\_PORTA\_BASE, solsay1) & solsay1)

{

GPIOPinIntClear(GPIO\_PORTA\_BASE, solsay1);

if(!GPIOPinRead(GPIO\_PORTA\_BASE,solsay2))

{

sayac2--;

}

else

{

sayac2++;

hiz\_olc1();

}

}

else if(GPIOPinIntStatus(GPIO\_PORTA\_BASE, solsay2) & solsay2 )

{

GPIOPinIntClear(GPIO\_PORTA\_BASE, solsay2);

if(!GPIOPinRead(GPIO\_PORTA\_BASE,solsay1))

{

sayac2++;

}

else

{

sayac2--;

}

}

UARTprintf("%d \n", hiz);

}

// void HizAyarla (long a,long b)

//

// {

//

// if(hiz>a)

// {

// sagMotorHizi=sagMotorHizi-2500;

// }

// if(hiz<a)

// {

// sagMotorHizi=2500+sagMotorHizi;

// }

// if(hiz1>b)

// {

// solMotorHizi=solMotorHizi-2000;

// }

// if(hiz1<b)

// {

// solMotorHizi=2000+solMotorHizi;

// }

//

// if(solMotorHizi>54000)

// solMotorHizi=54000;

// if(solMotorHizi<35000)

// solMotorHizi=35000;

//

// if(sagMotorHizi>54000)

// sagMotorHizi=54000;

// if(sagMotorHizi<35000)

// sagMotorHizi=35000;

//

// ileri(sagMotorHizi,solMotorHizi);

//

// }

void duz\_ileri2(void)

{

if( (!GPIOPinRead(GPIO\_PORTE\_BASE,sensor1)))

{

sagMotorHizi-=5000;

solMotorHizi+=4000;

}

else if((!GPIOPinRead(GPIO\_PORTF\_BASE,sensor2)))

{

sagMotorHizi-=4000;

solMotorHizi+=3000;

}

else if( (!GPIOPinRead(GPIO\_PORTF\_BASE,sensor3)))

{

sagMotorHizi-=3000;

solMotorHizi+=2000;

}

else if( (!GPIOPinRead(GPIO\_PORTB\_BASE,sensor6)) )

{

sagMotorHizi+=2000;

solMotorHizi-=2000;

}

else if( (!GPIOPinRead(GPIO\_PORTB\_BASE,sensor7)))

{

sagMotorHizi+=3000;

solMotorHizi-=3000;

}

else if( (!GPIOPinRead(GPIO\_PORTE\_BASE,sensor8)))

{

sagMotorHizi+=4000;

solMotorHizi-=4000;

}

else if( (!GPIOPinRead(GPIO\_PORTB\_BASE,sensor4))|| (!GPIOPinRead(GPIO\_PORTE\_BASE,sensor5)))

{

sagMotorHizi=38000;

solMotorHizi=40000;

}

if (sagMotorHizi < 36000 ) sagMotorHizi = 36000;

if (solMotorHizi < 37000 ) solMotorHizi = 37000;

if (sagMotorHizi >= 55000) sagMotorHizi = 55000;

if (solMotorHizi >= 55000) solMotorHizi = 55000;

ileri(sagMotorHizi,solMotorHizi);

}

// void duz\_ileri1(void)

// {

// if( (!GPIOPinRead(GPIO\_PORTB\_BASE,sensor4))|| (!GPIOPinRead(GPIO\_PORTE\_BASE,sensor5)))

// {

// HizAyarla(1650000,1650000);

// }

// else if( (!GPIOPinRead(GPIO\_PORTE\_BASE,sensor1)))

// {

//

// HizAyarla(1250000,2000000);

//

// }

// else if((!GPIOPinRead(GPIO\_PORTF\_BASE,sensor2)))

// {

// HizAyarla(1350000,1900000);

//

//

// }

// else if( (!GPIOPinRead(GPIO\_PORTF\_BASE,sensor3)))

// {

// HizAyarla(1400000,1800000);

//

//

// }

//

// else if( (!GPIOPinRead(GPIO\_PORTB\_BASE,sensor6)) )

// {

//

//

// HizAyarla(2200000,1500000);

//

// }

// else if( (!GPIOPinRead(GPIO\_PORTB\_BASE,sensor7)))

// {

//

//

// HizAyarla(2300000,1400000);

//

// }

// else if( (!GPIOPinRead(GPIO\_PORTE\_BASE,sensor8)))

// {

//

//

// HizAyarla(2425000,1300000);

//

// }

//

//

//

// }

void Timer2IntHandler(void)

{

TimerIntClear(TIMER2\_BASE, TIMER\_TIMA\_TIMEOUT);

}

int main(void)

{

ulPeriod = 6400000;

SysCtlClockSet(SYSCTL\_SYSDIV\_2\_5|SYSCTL\_USE\_PLL|SYSCTL\_XTAL\_16MHZ|SYSCTL\_OSC\_MAIN);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOA);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOB);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOC);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOD);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOE);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPIOF);

GPIOPinTypeGPIOOutput(GPIO\_PORTF\_BASE, GPIO\_PIN\_1);

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = GPIO\_LOCK\_KEY\_DD;//sw2 serbest

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_CR) |= 0x01;

HWREG(GPIO\_PORTF\_BASE + GPIO\_O\_LOCK) = 0;

GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE,GPIO\_PIN\_0|GPIO\_PIN\_4);//sw1 basla sw2 dur

GPIOPinTypeGPIOInput(GPIO\_PORTA\_BASE,sagsay1|sagsay2|solsay1|solsay2);

GPIOPadConfigSet(GPIO\_PORTA\_BASE, sagsay1|sagsay2|solsay1|solsay2, GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPD);

GPIOPadConfigSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_4, GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

GPIOIntTypeSet(GPIO\_PORTF\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_4, GPIO\_FALLING\_EDGE);

GPIOIntTypeSet(GPIO\_PORTA\_BASE, sagsay1|sagsay2|solsay1|solsay2, GPIO\_RISING\_EDGE);

GPIOPinIntEnable(GPIO\_PORTF\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_4);

GPIOPinIntEnable(GPIO\_PORTA\_BASE, sagsay1|sagsay2|solsay1|solsay2);

// IntEnable(INT\_GPIOF);

IntEnable(INT\_GPIOA);

// IntEnable(INT\_GPIOB);

GPIOPinTypeGPIOOutput(GPIO\_PORTC\_BASE, GPIO\_PIN\_4);

GPIOPinTypeGPIOOutput(GPIO\_PORTC\_BASE, GPIO\_PIN\_5);

GPIOPinTypeGPIOOutput(GPIO\_PORTC\_BASE, GPIO\_PIN\_6);

GPIOPinTypeGPIOOutput(GPIO\_PORTC\_BASE, GPIO\_PIN\_7);

GPIOPinConfigure(GPIO\_PC4\_WT0CCP0);

GPIOPinConfigure(GPIO\_PC5\_WT0CCP1);

GPIOPinConfigure(GPIO\_PC6\_WT1CCP0);

GPIOPinConfigure(GPIO\_PC7\_WT1CCP1);

GPIOPinTypeTimer(GPIO\_PORTC\_BASE, GPIO\_PIN\_4 );

GPIOPinTypeTimer(GPIO\_PORTC\_BASE, GPIO\_PIN\_5 );

GPIOPinTypeTimer(GPIO\_PORTC\_BASE, GPIO\_PIN\_6 );

GPIOPinTypeTimer(GPIO\_PORTC\_BASE, GPIO\_PIN\_7 );

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_WTIMER0);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_WTIMER1);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER0);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER1);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_TIMER2);

TimerConfigure(TIMER0\_BASE, TIMER\_CFG\_32\_BIT\_PER);

TimerConfigure(TIMER1\_BASE, TIMER\_CFG\_32\_BIT\_PER);

TimerConfigure(TIMER2\_BASE, TIMER\_CFG\_32\_BIT\_PER);

TimerLoadSet(TIMER0\_BASE, TIMER\_A, SysCtlClockGet());

TimerLoadSet(TIMER1\_BASE, TIMER\_A, SysCtlClockGet());

TimerIntEnable(TIMER2\_BASE, TIMER\_TIMA\_TIMEOUT);

TimerLoadSet(TIMER2\_BASE, TIMER\_A, SysCtlClockGet()/10);

TimerEnable(TIMER0\_BASE, TIMER\_A);

TimerEnable(TIMER1\_BASE, TIMER\_A);

IntEnable(INT\_TIMER2A);

TimerConfigure(WTIMER0\_BASE, (TIMER\_CFG\_SPLIT\_PAIR|TIMER\_CFG\_A\_PWM|TIMER\_CFG\_B\_PWM));

TimerConfigure(WTIMER1\_BASE, (TIMER\_CFG\_SPLIT\_PAIR|TIMER\_CFG\_A\_PWM|TIMER\_CFG\_B\_PWM));

TimerControlLevel(WTIMER0\_BASE, TIMER\_BOTH, 0);

TimerControlLevel(WTIMER1\_BASE, TIMER\_BOTH, 0);

ulPeriod = 64000;

TimerLoadSet(WTIMER0\_BASE, TIMER\_A, ulPeriod -1);

TimerLoadSet(WTIMER0\_BASE, TIMER\_B, ulPeriod -1);

TimerLoadSet(WTIMER1\_BASE, TIMER\_A, ulPeriod -1);

TimerLoadSet(WTIMER1\_BASE, TIMER\_B, ulPeriod -1);

TimerMatchSet(WTIMER0\_BASE, TIMER\_A, ulPeriod-2);

TimerMatchSet(WTIMER0\_BASE, TIMER\_B, ulPeriod-2);

TimerMatchSet(WTIMER1\_BASE, TIMER\_A, ulPeriod-2);

TimerMatchSet(WTIMER1\_BASE, TIMER\_B, ulPeriod-2);

TimerEnable(WTIMER0\_BASE, TIMER\_BOTH);

TimerEnable(WTIMER1\_BASE, TIMER\_BOTH);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_UART0);

GPIOPinConfigure(GPIO\_PA0\_U0RX);

GPIOPinConfigure(GPIO\_PA1\_U0TX);

GPIOPinTypeUART(GPIO\_PORTA\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1);

UARTConfigSetExpClk(UART0\_BASE, SysCtlClockGet(), 115200,(UART\_CONFIG\_WLEN\_8 | UART\_CONFIG\_STOP\_ONE | UART\_CONFIG\_PAR\_NONE));

IntEnable(INT\_UART0);

UARTIntEnable(UART0\_BASE, UART\_INT\_RX | UART\_INT\_RT);

UARTStdioInit(0);

IntMasterEnable();

UARTprintf("deneme\n");

GPIOPinWrite(GPIO\_PORTF\_BASE, GPIO\_PIN\_1,GPIO\_PIN\_1);

GPIOPinTypeGPIOInput(GPIO\_PORTB\_BASE,sensor4|sensor6|sensor7);

GPIOPinTypeGPIOInput(GPIO\_PORTE\_BASE,sensor1|sensor5|sensor8);

GPIOPinTypeGPIOInput(GPIO\_PORTF\_BASE,sensor2|sensor3);

GPIOPadConfigSet(GPIO\_PORTB\_BASE,sensor4|sensor6|sensor7,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

GPIOPadConfigSet(GPIO\_PORTE\_BASE,sensor1|sensor8|sensor5,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

GPIOPadConfigSet(GPIO\_PORTF\_BASE,sensor2|sensor3,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

hiz=SysCtlClockGet()-TimerValueGet(TIMER0\_BASE,TIMER\_A);

TimerLoadSet(TIMER0\_BASE, TIMER\_A, SysCtlClockGet());

hiz1=SysCtlClockGet()-TimerValueGet(TIMER1\_BASE,TIMER\_A);

TimerLoadSet(TIMER1\_BASE, TIMER\_A, SysCtlClockGet());

//max 380 400 düz gitme için

//TimerMatchSet(WTIMER0\_BASE, TIMER\_A, 45000); // Timer 0 Match set

// TimerMatchSet(WTIMER1\_BASE, TIMER\_A, 45000); // Timer 0 Match set

while(1)

{

duz\_ileri2();

}

//

//yavas ilerisi 420 440

}