//RSK

#include <stdio.h>

#include <unistd.h> /\* for sleep() \*/

#include <stdlib.h> /\* for EXIT\_\* \*/

#include <stdint.h> /\* for uintptr\_t \*/

#include <hw/inout.h> /\* for in\*() and out\*() functions \*/

#include <sys/neutrino.h> /\* for ThreadCtl() \*/

#include <sys/syspage.h> /\* for for cycles\_per\_second \*/

#include <sys/mman.h> /\* for mmap\_device\_io() \*/

#include <pthread.h>

#include <time.h>

#define BASE 0x280

#define PORT\_LENGTH 1

#define LSB\_ADRRESS 0x280

#define MSB\_ADDRESS 0x281

#define CHANNEL\_ADDRESS 0x282

#define GAIN\_ADDRESS 0x283

#define PORTA\_ADDRESS 0x288

#define PORTB\_ADDRESS 0x289

/\* Definining voltage related values. \*/

#define MIN\_VOLTAGE -5

#define MAX\_VOLTAGE 5

///referred to types.h

#ifndef INT8

typedef signed char INT8;

#endif

// Signed 16-bit Type

#ifndef INT16

typedef signed int INT16;

#endif

//setting up handles

uintptr\_t LSB\_HANDLE;

uintptr\_t MSB\_HANDLE;

uintptr\_t CHANNEL\_HANDLE;

uintptr\_t GAIN\_HANDLE;

uintptr\_t PORTA\_HANDLE;

uintptr\_t PORTB\_HANDLE;

// Variables for ADC data and voltage conversion

INT8 LSB, MSB;

INT16 data;

float voltage;

int main(void)

{

int privity\_err;

privity\_err = ThreadCtl(\_NTO\_TCTL\_IO, NULL);

if (privity\_err == -1) {

fprintf(stderr, "can't get root permissions\n");

return -1;

}

//setting up addresses

LSB\_HANDLE = mmap\_device\_io(PORT\_LENGTH, LSB\_ADRRESS);

MSB\_HANDLE= mmap\_device\_io(PORT\_LENGTH, MSB\_ADDRESS);

CHANNEL\_HANDLE= mmap\_device\_io(PORT\_LENGTH, CHANNEL\_ADDRESS);

GAIN\_HANDLE = mmap\_device\_io(PORT\_LENGTH, GAIN\_ADDRESS);

PORTA\_HANDLE= mmap\_device\_io(PORT\_LENGTH, PORTA\_ADDRESS);

PORTB\_HANDLE= mmap\_device\_io(PORT\_LENGTH, PORTB\_ADDRESS);

out8(PORTA\_HANDLE, 0x00);

for(;;){

//Selecting the range

out8(CHANNEL\_HANDLE, 0x44);

out8(GAIN\_HANDLE, 0x01);

while(in8(GAIN\_HANDLE) & 0x20){}

//Starting the A/D Conversion

out8(LSB\_HANDLE, 0x80);

//Wait for conversions to finish

while((in8(GAIN\_HANDLE) & 0x80) !=0){}

//Read the data

LSB = in8(LSB\_HANDLE);

MSB = in8(MSB\_HANDLE);

data = (MSB\*256) + LSB;

//Converting the data

voltage = (float)(data\*5)/32768;

if (voltage < MIN\_VOLTAGE || voltage > MAX\_VOLTAGE) //check the voltage range

{

printf("!!!!!!!!!!!!!!!!!!!!!!VOLTAGE OUT OF RANGE!!!!!!!!!!!!!!!!!!!!!!!!!");

}

else

{

int int\_voltage = (int)(voltage);

printf("%d v\n", (int\_voltage));

//send data -- b=0 for negative voltage b=1 if positive voltage.

if(int\_voltage>0){

out8(PORTA\_HANDLE,int\_voltage);

out8(PORTB\_HANDLE,1);

}else{

out8(PORTB\_HANDLE,0);

int\_voltage=abs(int\_voltage);

out8(PORTA\_HANDLE,int\_voltage);

}

}

}

}