/\*

\* Projekt.c

\*

\* Created: 15.01.2024

\*

\* Author : Yamk, Hassoumi

\*/

#define F\_CPU 8000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <stdlib.h>

#include "FreeRTOS.h"

#include "task.h"

#include "queue.h"

#include "LCD\_Programm.h"

#include "SPI\_interface.h"

void vTask1( void \*pvParameters );

void vTask2( void \*pvParameters );

QueueHandle\_t xQueue;

/\*-----------------------------------------------------------\*/

int main( void )

{

char buffer[16];

DDRC = 0xFF;

LCD\_Init();

SPI\_Init();

xQueue = xQueueCreate (2,sizeof(float));

if( xQueue != *NULL* )

{

xTaskCreate( vTask1, "Read\_Task", 1000, *NULL*, 2, *NULL* );

xTaskCreate( vTask2, "Write\_Task", 1000, *NULL*, 1, *NULL* );

}

// Who am i

LCD\_Print("Who am I? ");

LCD\_Cursor\_Position(0,2);

LCD\_Print(*utoa*(SPI\_MasterReceive(WHO\_AM\_I),buffer,2));

*\_delay\_ms*(3000);

LCD\_Init();

vTaskStartScheduler();

while(1)

{

PORTC = 0x00;

}

return 0;

}

/\*-----------------------------------------------------------\*/

void vTask1( void \*pvParameters )

{

TickType\_t xLastWakeTime = xTaskGetTickCount();

*uint8\_t* i = 0;

float f\_Tx\_values[2];

while(1)

{

f\_Tx\_values[0] = read\_Pressure();

f\_Tx\_values[1] = read\_Temperature();

xQueueSendToBack(xQueue, &f\_Tx\_values[0], 0);

xQueueSendToBack(xQueue, &f\_Tx\_values[1], 0);

PORTC &= ~(1<<i);

i++;

if (i==8)

{

i = 0;

}

//vTaskDelay(pdMS\_TO\_TICKS(100UL));

vTaskDelayUntil(&xLastWakeTime, pdMS\_TO\_TICKS(100));

}

}

/\*-----------------------------------------------------------\*/

void vTask2( void \*pvParameters )

{

char buffer[16];

float f\_Rx\_values[2];

while(1)

{

if (uxQueueMessagesWaiting( xQueue ) != 0)

{

/\*ob der Queue leer ist: Error: nichts empfangen \*/

}

xQueueReceive(xQueue, &f\_Rx\_values[0], pdMS\_TO\_TICKS(100));

xQueueReceive(xQueue, &f\_Rx\_values[1], pdMS\_TO\_TICKS(100));

PORTC = 0xFF;

LCD\_Cursor\_Position(0,1);

// Pressure

LCD\_Print("p=");

LCD\_Print(ftoa(f\_Rx\_values[0],1,buffer));

LCD\_Print(" hPa");

// Temperature

LCD\_Cursor\_Position(0,2);

LCD\_Print("T=");

LCD\_Print(ftoa(f\_Rx\_values[1],1,buffer));

LCD\_Print("C ");

// NHN

LCD\_Print("h=");

LCD\_Print(ftoa(NHN(f\_Rx\_values[0]),1,buffer));

LCD\_Print("m");

}

}