trabalho microcontroladores 2

Marco Antonio

1 Introduction

Esse relatorio visa cumprir a função de mostrar devidamente os pinos a que foram conectados os pinos do display, e explicar o codigo ja que o display do aluno marcos não funcionava

2 Pinos

Red Nokia5110 Blue Nokia 5110
Signal (Nokia 5110) LaunchPad pin
Reset (RST, pin 1) connected to PA7
SSI0Fss (CE, pin 2) connected to PA3
Data/Command (DC, pin 3) connected to PA6
SSI0Tx (Din, pin 4) connected to PA5
SSI0Clk (Clk, pin 5) connected to PA2
3.3V (Vcc, pin 6) power
back light (BL, pin 7) not connected
Ground (Gnd, pin 8) ground

3 Código

3.1 includes

Aqui nessa seção estão os includes necessarios

```
#include <stdint.h>
#include<stdbool.h>
#include <stdio.h>
#include "driverlib/gpio.h"
#include "inc/hw_memmap.h"
#include "inc/hw_gpio.h"
#include "inc/hw_ints.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
```

```
#include "Nokia5110.h"
#include "inc/tm4c123gh6pm.h"
```

3.2 Funções

Nessa seção estão as configurações dos perifericos para o uart, e aconfiguração do clock.Logo no inicio tem a variavel int modrel = 1 será usada na interrupt para alternar entre o modo digital e analógico.

```
int modrel = 1;
// variaveis para as horas, segundos, minutos
char horas , minutos , segundos;

void UartConfig(){
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    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));
}
```

As funções relogiodig e relogioanal são as funções para os relogios analogico e digitais, que recebem as variaveis da uart através do terminal, e as printam no display Nokia5110.a interrupt aq é usada para alternar entre os modos analogico e digital.

```
void relogio_dig(char horas, char minutos, char
   segundos ){
     Nokia5110_SetCursor(3,3);
     Nokia5110_OutChar(horas);
     Nokia5110_SetCursor(3, 4);
     Nokia5110_OutChar(':');
     Nokia5110_SetCursor(3, 5);
     Nokia5110_OutChar(minutos);
     Nokia5110_SetCursor(3,6);
     Nokia5110_OutChar(':');
     Nokia5110_SetCursor(3,7);
     Nokia5110_OutChar(segundos);
}

void relogio_anal(char horas, minutos){

      Nokia5110_SetCursor(0,3);
```

```
        Nokia5110_OutDec(3);
        Nokia5110_SetCursor(5, 0);
        Nokia5110_OutDec(12);
        Nokia5110_SetCursor(5,5);
        Nokia5110_OutDec(6);
        Nokia5110_SetCursor(11,3);
        Nokia5110_OutDec(9);



}


void PortFIntHandler(void){
    GPIOIntClear(GPIO_PORTF_BASE, GPIO_PIN_0);
    if(!GPIOPinRead(GPIO_PORTF_BASE, GPIO_PIN_0)&GPIO)
    {
        if(mod_rel==1){
            relogio_dig(horas, minutos, segundos);
            mod_rel++;
        }else if(mod_rel==2){
            relogio_anal(horas, minutos);
            mod_rel = 1;
        }

    }

}
```

3.3 main

Na main os valores obtidos da uart são passados para as variaveis, e dentro do bloco while a função UARTCharsAvail(UART0BASE) verifica se há dados na porta indefinidamente.

```
int main(){
  SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|
     SYSCTL_OSC_MAIN|SYSCTL_XTAL_16MHZ);
  Nokia5110_Init();
  UartConfig();
  while(1)
  {

      //l   as horas , minutos e segundos do terminal
      horas = UARTCharGet(UART0_BASE);
      minutos = UARTCharGet(UART0_BASE);
```

```
        segundos = UARTCharGet(UART0_BASE);

        Nokia5110_Clear();
            if (UARTCharsAvail(UART0_BASE)) UARTCharPut(
                UART0_BASE, relogio_dig(horas, minutos,
                segundos));

    }
```

4 considerações finais

a ideia para o relogio analogico era usar codigo em python para gerar bitmaps apartir disso printar o necessario

```
    from PIL import Image, ImageDraw
import datetime
import math

# Criar uma nova imagem vazia
width, height = 400, 400
image = Image.new('1', (width, height), 1)  # '1'
   representa o modo de imagem bitmap (1-bit)
draw = ImageDraw.Draw(image)

# Definir o raio do rel gio
radius = 150

# Calcular as coordenadas do centro
center_x = width // 2
center_y = height // 2

# Desenhar o c rculo do rel gio
draw.ellipse([(center_x - radius, center_y - radius),
              (center_x + radius, center_y + radius)],
                  outline=0)

# Obter a hora atual
now = datetime.datetime.now()
hour = now.hour % 12
minute = now.minute
second = now.second

# Calcular os  ngulos  dos ponteiros
hour_angle = math.radians((hour * 30) + (minute * 0.5)
   )
```

```
minute_angle = math.radians((minute * 6) + (second *
    0.1))
second_angle = math.radians(second * 6)

# Desenhar os ponteiros
hour_length = 0.4 * radius
minute_length = 0.6 * radius
second_length = 0.8 * radius

draw.line([(center_x, center_y),
           (center_x + hour_length * math.cos(
               hour_angle), center_y - hour_length *
               math.sin(hour_angle))], fill=0, width=3)

draw.line([(center_x, center_y),
           (center_x + minute_length * math.cos(
               minute_angle), center_y - minute_length
               * math.sin(minute_angle))], fill=0,
               width=2)

draw.line([(center_x, center_y),
           (center_x + second_length * math.cos(
               second_angle), center_y - second_length
               * math.sin(second_angle))], fill=0,
               width=1)

# Salvar a imagem em formato PNG
image.save('relogio.png')

# Converter a imagem em bitmap
bitmap_image = image.convert('1')

# Salvar o bitmap
bitmap_image.save('relogio_bitmap.bmp')
```

Logo em seguida o codigo em c para organizar o vetor

```
#include <stdio.h>

void printMatrix(int matrix[][48], int rows, int cols)
    {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("0x%02X", matrix[i][j]);
            if (j != cols - 1) {
                printf(", ");
            }
```

```
        }
        printf(",\n");
    }
}

int main() {
    int matrix[84][48] = {
        // Insira os valores da matriz aqui
    };

    printMatrix(matrix, 84, 48);

    return 0;
}
```