

Disciplina: Sistema Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 13_UART_14_SPI_2

1. Considere um MSP430 sendo usado para leituras analógicas. O Raspberry Pi está conectado a ele via UART. O MSP430 foi programado para converter e enviar dados de 10 bits a cada 10 ms. Escreva o código para o Raspberry Pi receber estes dados, e cada 1 segundo apresentar no terminal a média das últimas 100 amostras.

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>

// Arquivo de acesso a porta serial
#define TTY "/dev/ttyAMA0"
// Arquivo de acesso a porta serial
// PARA O RASPBERRY PI 3
#define TTY "/dev/ttyS0"

int uart0_fd;

void ctrl_c(int sig)
{
    puts(" Fechando " TTY "...");
    close(uart0_fd);
    exit(-1);
}

int main(void)
{
    struct termios options;
    //char user_input, msp430_return=0;

    signal(SIGINT, ctrl_c);
    uart0_fd = open(TTY, O_RDWR); // | O_NOCTTY); // | O_NDELAY);
    if(uart0_fd===-1)
    {
        puts("Erro abrindo a UART. Garanta que ela nao esteja sendo usada por outra
aplicacao.");
        return -1;
    }
}
```

```
}
puts(TTY " aberto.");
tcgetattr(uart0_fd, &options);
options.c_cflag = CS8 | CREAD | CLOCAL;
options.c_iflag = 0;
options.c_oflag = 0;
options.c_lflag = 0;
options.c_cc[VTIME] = 0;
options.c_cc[VMIN] = 1;
cfsetospeed(&options, B9600);
cfsetispeed(&options, B9600);
tcflush(uart0_fd, TCIOFLUSH);
tcsetattr(uart0_fd, TCSANOW, &options);
puts("UART configurada:");
system("stty -F " TTY);
puts("");
//user_input = 1;
while(1)
{
    int i;
    int soma = 0;
    char string[100];
    for(i = 0; i < 100; i++)
    {
        read(uart0_fd, &string[i], sizeof(string[i]));
        soma = soma + string[i];
        usleep(10000);
    }
    media = soma/100;
    //if(read(uart0_fd, &msp430_return, 1)==1)
    sleep(1);
    printf("A media das 100 amostras e = %d\n", media);

    puts("");
}
close(uart0_fd);
}
```

2. Considere um MSP430 sendo usado para leituras analógicas. O Raspberry Pi está conectado a ele via SPI, e é o mestre. O MSP430 foi programado para funcionar da seguinte forma:

- O MSP430 recebe o byte 0x55 e envia o byte 0xAA, o que indica o começo de conversão.
- 100us depois, o MSP430 recebe os bytes 0x01 e 0x02, e envia o byte menos significativo

e o mais significativo da conversão de 10 bits, nesta ordem.

Escreva o código para o Raspberry Pi executar este protocolo, de forma a obter conversões a cada 10 ms. A cada 1 segundo ele deve apresentar no terminal a média das últimas 100 amostras.

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>

// Arquivo de acesso a porta serial
// #define TTY "/dev/ttyAMA0"
// Arquivo de acesso a porta serial
// PARA O RASPBERRY PI 3
#define TTY "/dev/ttyS0"

int uart0_fd;
void ctrl_c(int sig)
{
    puts(" Fechando " TTY "...");
    close(uart0_fd);
    exit(-1);
}

float f_media(char vetor[]){

    int i;
    float media = 0;

    for(i=0;i<100;i++){

        media += vetor[i];
    }

    media = media/100;

    return media;
}

int main(void)
{
    struct termios options;
    int i;
    char user_input[100];
    float media;
```

```
signal(SIGINT, ctrl_c);
uart0_fd = open(TTY, O_RDWR); // | O_NOCTTY); // | O_NDELAY);
if(uart0_fd==-1)
{
    puts("Erro abrindo a UART. Garanta que ela nao esteja sendo usada por outra
aplicacao.");
    return -1;
}
puts(TTY " aberto.");
tcgetattr(uart0_fd, &options);
options.c_cflag = CS8 | CREAD | CLOCAL;
options.c_iflag = 0;
options.c_oflag = 0;
options.c_lflag = 0;
options.c_cc[VTIME] = 0;
options.c_cc[VMIN] = 1;
cfsetospeed(&options, B9600);
cfsetispeed(&options, B9600);
tcflush(uart0_fd, TCIOFLUSH);
tcsetattr(uart0_fd, TCSANOW, &options);
puts("UART configurada:");
system("stty -F " TTY);
puts("");
//user_input = 1;
while(1)
{
    for(i=0;i<100;i++){
        if(read(uart0_fd, &user_input[i],sizeof(char))>0)
        {
            //printf("Valor %d\n",user_input[i]);
            lseek(uart0_fd,0,SEEK_SET);
        }
        else{
            printf("Erro na leitura");
        }
        //usleep(1000);
    }
    media = f_media(user_input);
    printf("Media %.2f\n",media);
}
close(uart0_fd);
}
```