FEEC - UNICAMP

EA871

EXPERIMENTO DA TAREFA 6

VINÍCIUS ESPERANÇA MANTOVANI, RA: 247395

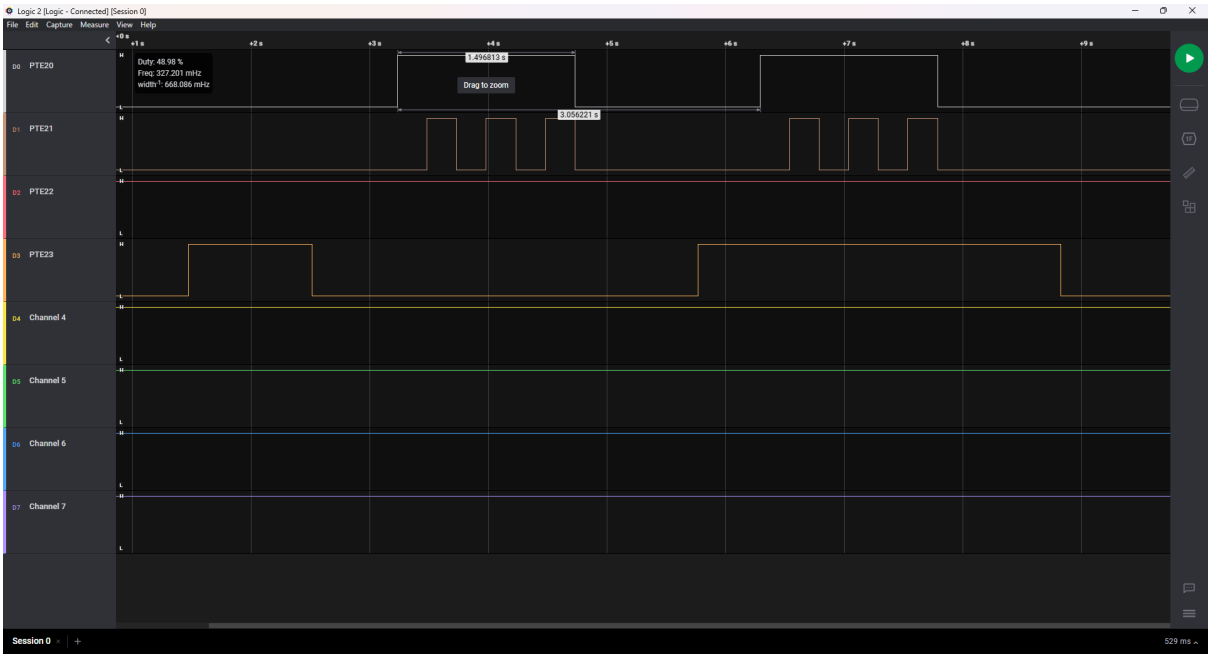
ÍTEM 1:

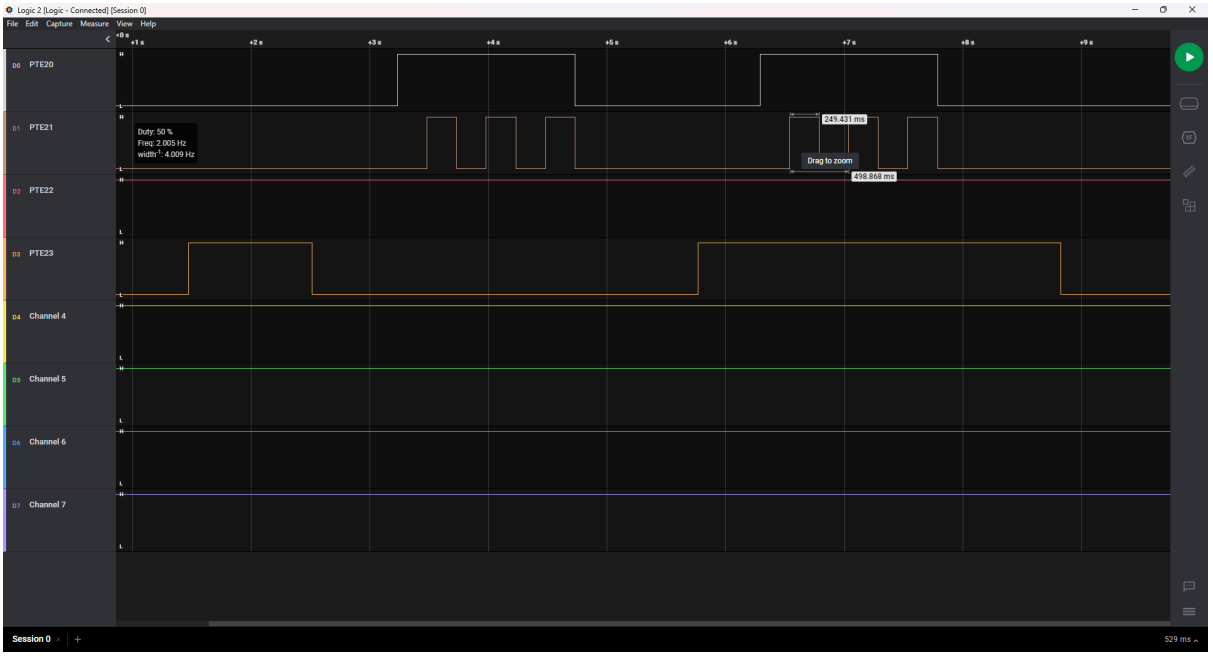
1.1) Observando o comportamento da placa durante a execução do programa, notei que o LCD registra a hora 03:25:46 de ínicio (não sei se é 45 ou 46 de início pelo LCD) e, tem seu valor incrementado nos segundos a cada segundo que passa, como se comporta um relógio. Além disso, ao apertar a botoeira IRQA5, o valor dos minutos registrado é incrementado em 1 e, após o pressionamento, o relógio demora poucos segundos para voltar a ser incrementado em segundos de onde parou.

1.2) O campo SIM_CLKDIV1_OUTDIV4 na inicialização padrão é 0b001. Porém, na configuração de rot6_aula o valor com que é configurado é de 0b111.

Os módulos afetados por essa configuração são os módulos afetados pelo bus clock, ou seja: PIT e RTC.

1.3)





1.3.1) A largura dos pulsos registrados em PTE21 está condizente com os valores de SIM_CLKDIV1_OUTDIV4 e PIT_LDVAL0, pois o valor de 0b011 colocado em SIM_CLKDIV1_OUTDIV4 faz causa divisão do bus clock por 4 → 20971520/4 = 5242880 e, o valor de 1310720 colocado em PIT_LDVAL0 é um quarto do valo de 5242880 do bus clock, logo, o período de PIT é de 0,25s = 250ms, que é condizente com a largura do sinal capturado em PTE21.

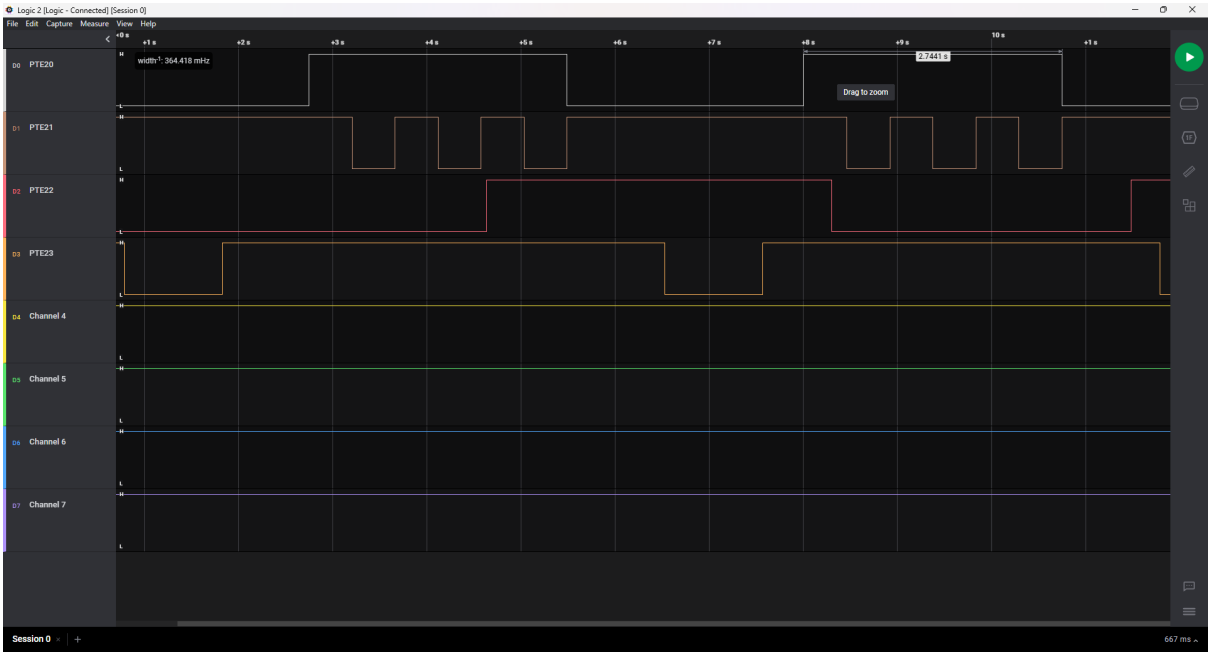
Se aumentarmos OUTDIV4, ocorre que teremos uma divisão do bus clock por um número maior, logo, o período de PIT aumentará, enquanto que a diminuição de OUTDIV4 causa queda desse período. Já o valor em LDVAL0 ocasiona, no caso de ser aumentado, um aumento no período de PIT, pois esse período será dado pela divisão de um número maior pelo bus clock, enquanto que, a diminuição desse valor ocasiona a queda do período de PIT.

1.3.2) Como redigido no exercício anterior, o valor do período de PIT por conta dos valores postos em SIM_CLKDIV1_OUTDIV4 e PIT_LDVAL0 é de 0,25s, logo, como o POSTSCALER está definido com valor 6, temos que o timeout é dado por 6*0,25 = 1,5s = 1500ms, que é um valor condizente com aquilo que observamos pelo analisador lógico na porta PTE20.

Se mudarmos o valor de POSTSCALER, a largura dos sinais em PTE21 não mudam, uma vez que a largura dos pulsos desse sinal representa o período do PIT. Ainda assim, no caso de aumentarmos o POSTSCALER, perceberemos um aumento proporcional na largura do sinal de PTE20, enquanto que, se diminuirmos, veremos uma queda proporcional na largura desse sinal.

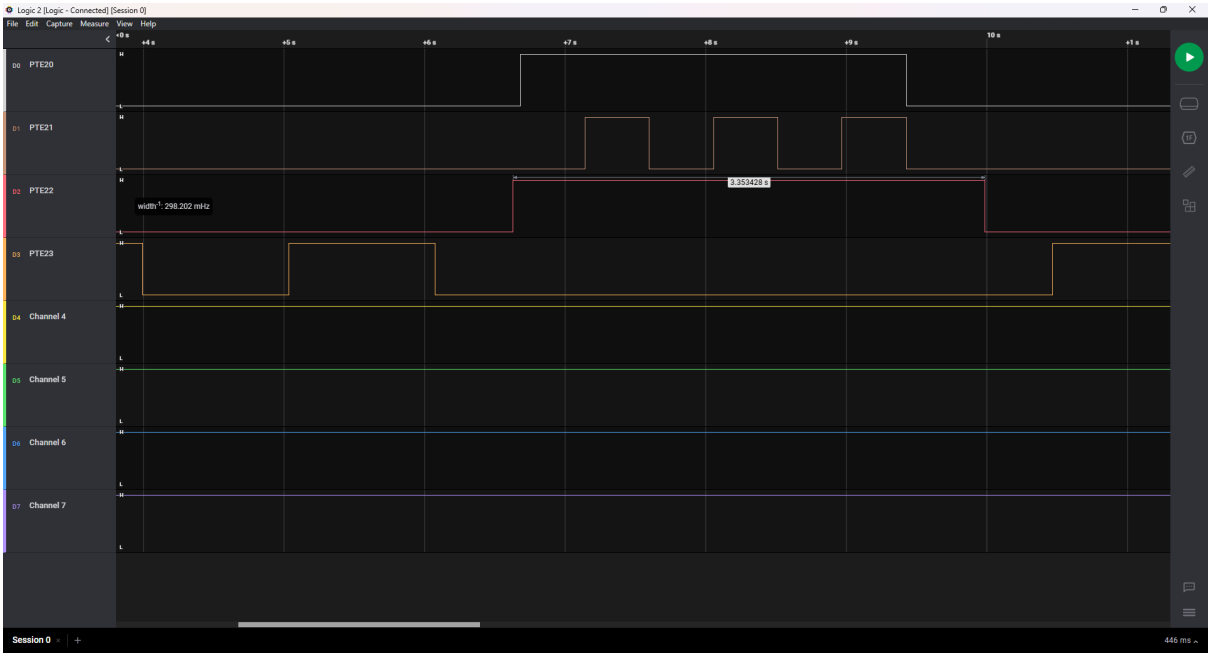
1.3.3) Nesse caso, precisaríamos alterar o valor de PIT_LDVAL0 para 2,75/6 = período desejado → período desejado = valor em PIT_LDVAL0 / bus clock = valor em PIT_LDVAL0 / 20971520 → valor em PIT_LDVAL0 / 20971520 = 2,75/6 → valor em PIT_LDVAL0 = 2,75*20971520 / 6 = 9611946.

Após alteração temos o seguinte sinal:



No print percebe-se que o a largura está com o valor desejado.

1.4)



1.4.1) É bastante evidente que os pulsos de aproximadamente 1 segundo são condizentes com o código, uma vez que, ao ser habilitado normalmente, o RTC opera de maneira a gerar pulsos de 1 em 1 segundo. No entanto, a análise do pulso ocasionado pelo pressionamento da botoeira é mais profunda. Isso porque, ao pressionar a botoeira, os registradores RTC_SR, RTC_TPR e RTC_TPR são alterados para setar o novo tempo e, enquanto isso ocorre, há uma pausa na mudança de valores da variável atual, que permanece a mesma até o fim do

timeout. Desse modo, a condição na main atual > anterior não é atendida e não se passa pelo if cujo conteúdo possui a instrução GPIO_toggleH5P1() responsável por permitir a exibição de pulsos de 1 em 1 segundo de RTC, por isso, durante o tempo pós pressionamento, o valor em PTE23 não se altera.

1.4.2) No caso de alteração da frequência do sinal de relógio, ocorre que o valor de $T_{RTC-LPO}$ passa a ser de 3,2768 segundos. O valor de PTE23 é diminuido, uma vez que, como a frequência é maior, a velocidade com que o tempo varia é maior, portanto, a condição “atual > anterior” é atendida com uma frequência maior e, consequentemente os pulsos em PTE23 são mais frequentes e seus períodos são menores. Já PTE22, passa a ter largura de pulso igual a 3,2768s, por conta da alteração da frequência em ordem 10, que ocasiona uma divisão por 10 do valor que anteriormente era de 32,768s.

1.4.3) Podemos fazer um alarme por meio de um contador, por exemplo, ao qual somariamos mais um para cada vez que, podemos reinterpretar por software o valor de 32,768s para passá-lo para a resolução de segundos, de acordo com a equação: **Segundos= (TSR ×32768 / f clock) + (TPR / f clock) = (TSR×32768+TPR) / fclock.** Assim, usando, ainda as eqs

$$TSR=(Segundos \times f\,clock)/32768=(Segundos \times 1000)/32768$$

$$TPR=(Segundos \times f\,clock)\ \%\ 32768=(Segundos \times 1000)\ \%\ 32768$$

usadas no código na função RTClpo_getTime, temos a resolução em segundos e, podemos fazer o contador do alarme ser incrementado com a passagem dos segundos, até que atinja uma condição (valor) de x segundos em que deve parar e tocar.

1.5) Os sinais de PTE22 são gerados na função RTC_Seconds_IRQHandler(). No caso de alterarmos o valor da frequência para 10kHz, devemos alterar o valor de f_{clock} nas equações 5 e 6, de forma que passem a ser 10000 e não mais 1000.

1.6) Os zeros de quando os valores são menores que 10 são inseridos no bloco

```
string[0] = (hh < 10)? '0': (hh/10)+'0';
string[1] = hh%10+'0';

string[3] = (mm < 10)? '0': (mm/10)+'0';
string[4] = mm%10+'0';

string[6] = (ss < 10)? '0': (ss/10)+'0';
string[7] = ss%10+'0';
```

Neste bloco, as posições do algarismo mais significativo, respectivamente de hora, minuto e segundo, são preenchidas por meio de condicionais, para as quais, caso “hh < 10”, “mm < 10”, “ss < 10”, então o algarismo mais significativo é preenchido com um zero e, caso contrário, é preenchido com o valor hh/10, mm/10 e/ou ss/10 (algarismo da casa das dezenas) mais o valor ASCII do 0, que é 49, para que a soma dê o valor desejado em ASCII correspondente ao caractere do algarismo desejado.

1.7) Ao executar o código alterado, não percebi mudanças nem nos sinais nem na placa. Ao adicionar as duas outras botoeiras, não devemos perceber mudança na fluidez, uma vez que as interrupções são mutuamente exclusivas, ou seja, enquanto uma delas roda, é impossível que seja chamada outra, mesmo que o botão correspondente seja pressionado e, a verificação de interrupção enquanto não é pressionado botão nenhum não causa perda de fluidez notória.

1.8)

1.8.1) O conteúdo de hor é inicializado por meio da função ConvertSectoDay que está na função ISR_carregaHorario() que, por sua vez, está na função PORTA_IRQHandler.

1.8.2) A menos que ocorra um pressionamento de botoeira, os valores de RTC_TPR e RTC_TSR são os mesmo daqueles registrados em hor em uma sessão, pois não é chamada nem a função de mudança do valor de RTC_TPR e TSR nem de hor.

1.8.3) Os valores desses dois registradores são alterados na função RTClpo_setTime em RTC.c. Após ser iniciado o com seu valor na main, a condição para que seja alterado é a ocorrência de overflow no PIT, para que ocorra a entrada na função de tratamento PIT_IRQHandler.

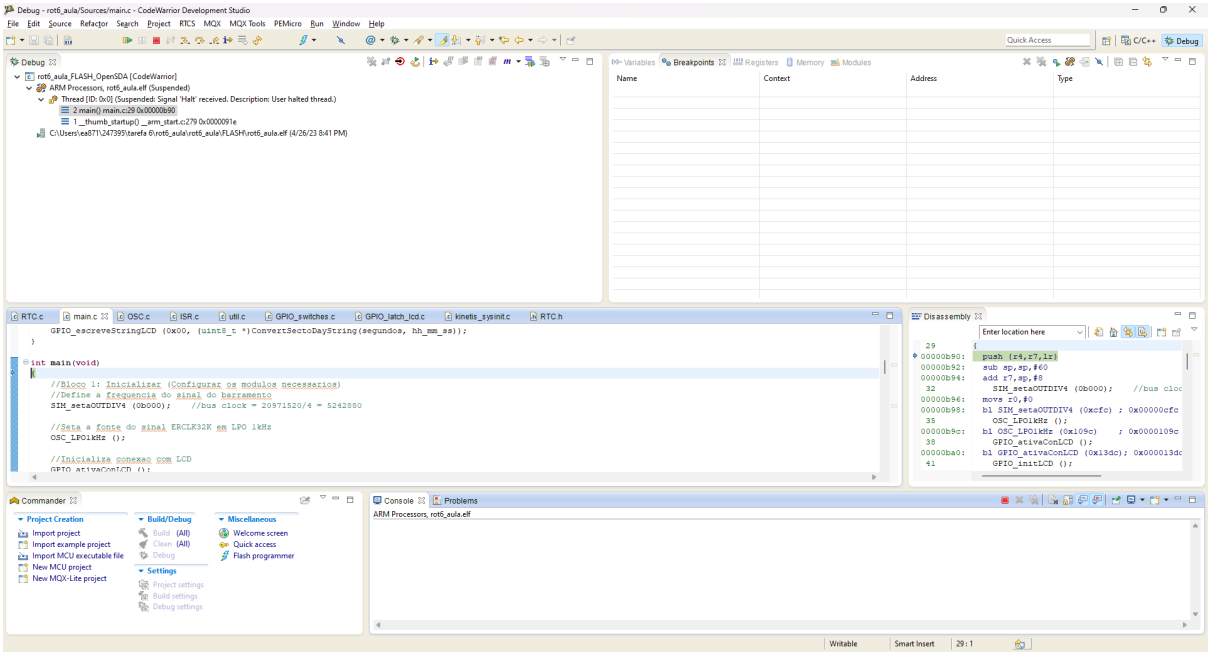
1.8.4) Isso ocorre em RTClpo_setTime.

1.9)

1.9.1) A contagem por PIT começa quando é pressionada a botoeira e termina após 2,75s, por conta da configuração que fizemos anteriormente.

1.9.2) A garantia de que a contagem inicia sempre em zero é a desativação do módulo em PIT_desativaTimer0 e a ativação em PIT_ativaTimer0.

2)



2.1) MCG_C1 = 0x04 = 0b0100, MCG_C5 = 0x00 → Logo, no caso do MCG_C1 tem a bits 7 e 6 valendo 0, logo a fonte de MCGOUTCLK no caso de C1 é FLL, pois é setado o registrador MCG_S tem valor 0x10, logo seu bit 5 é 0 e, a fonte do PLLS é FLL e não PLL, ademais, podemos notar que como o bit 2 de MCG_C1 é 1, a referência para FLL é o clock interno lento de referência (The slow internal reference clock no manual). E, como C5 tem valor 0, então o PLL está desativado, o que é outra forma de ver o que foi escrito acima.

2.2) PIT_TCTRL0 = 0x00000003
PIT_LDVAL0 = 0x0092aaaa
PIT_MCR = 0x00000000

RTC_TSR = 0x00000000
RTC_TPR = 0x00002dc8
RTC_SR = 0x00000014

2.3) PTE22 é configurado na função GPIO_initH5P1P2. PTE20 é configurado em GPIO_initH5P3P4 e alterado em PIT_IRQHandler. PTE21 é alterado em PIT_IRQHandler e configurado inicialmente em GPIO_initH5P3P4. Já PTE23 é configurado em GPIO_initH5P1P2.