

Liste a funcionalidade dos periféricos listados a seguir : 1. RTC - Real time clock / 2. TC - Timer/Counter :

1. Real time Clock à É um relógio que tem como finalidade manter o controle do tempo presente. Ele pode ser utilizado em várias funções do micro controlador, controlando a realizações dessas funções em um tempo mais preciso. Tem um consumo bem baixo de energia, é usado para registro de calendário e alarme, também muito utilizado para gerar inúmeras formas de onda.

Timer Counter à Têm a funcionalidade de trabalhar com um amplo campo de funções nos micro controladores, como frequência, medição, contagem de eventos, medição de intervalos, geração de pulsos, estabelecimento de tempo de atraso e modulação de largura de pulso.

Encontre os endereços referentes aos seguintes periféricos :

1. PIOA / 2. PIOB / 3. ACC / 4. UART1 / 5. UART2

1: PIOA: 0x400B0B00

2:PIOB: 0x400B1000

3: ACC : 0x40044000

4: UART1:0x4 00x0A00

5: UART0: 0x400E0600

Verifique no datasheet do uC os pinos físicos do uC associados aos I/O: PA01, PB22 e PC12. :

2. PC12 – AD12; PB22 -- ** ; PA01 – PGMEN1

Verifique quais periféricos podem ser configuráveis nos I/Os :

1. PC20 / 2. PB3.

PC0–PC31 Parallel IO Controller C e os PB0–PB14 Parallel IO Controller B , ou seja possuem a capacidade de operarem de forma paralela e em seu estado de reset possui um pull-up interno

No PC20 podem ser configurados os seguintes periféricos : A2, PWMH2

No PB3 podem ser configurados os seguintes periféricos: UTXD1 , UTXD1 , X1

• O que é debouncing ?

Por definição, quando apertamos um botão, o qual possui um contato mecânico, é gerada uma série de ruídos aleatórios chamados de spikes. Ou seja, a tensão instantânea sobre o botão pode ser qualquer valor, entre os valores que seriam obtidos com o botão em repouso e com o botão ligado após um tempo longo. Sendo o termo "DEBOUNCING" significa uma técnica que

nos dê certeza de que o botão apertado apenas uma vez seja interpretado corretamente como apenas UM aperto de botão.

- Descreva um algoritmo que implemente o debouncing.

O algoritmo abaixo implementa um debouncing de modo que ele checa a cada 5 milissegundos e verifica se a tecla foi apertada durante um certo período e após ela ser apertada também é verificado para saber se o nível lógico se manteve em alto ou em baixo para ter certeza de que a tecla foi realmente apertada

```
#define CHECK_MSEC 5 // Read hardware every 5 msec
#define PRESS_MSEC 10 // Stable time before registering pressed
#define RELEASE_MSEC 100 // Stable time before registering released
// This function reads the key state from the hardware.
extern bool_t RawKeyPressed();
// This holds the debounced state of the key.
bool_t DebouncedKeyPress = false;
// Service routine called every CHECK_MSEC to
// debounce both edges
void DebounceSwitch1(bool_t *Key_changed, bool_t *Key_pressed)
{
    static uint8_t Count = RELEASE_MSEC / CHECK_MSEC;
    bool_t RawState;
    *Key_changed = false;
    *Key_pressed = DebouncedKeyPress;
    RawState = RawKeyPressed();
    if (RawState == DebouncedKeyPress) {
        // Set the timer which allows a change from current state.
        if (DebouncedKeyPress) Count = RELEASE_MSEC / CHECK_MSEC;
        else Count = PRESS_MSEC / CHECK_MSEC;
    } else {
        // Key has changed - wait for new state to become stable.
        if (--Count == 0) {
            // Timer expired - accept the change.
            DebouncedKeyPress = RawState;
            *Key_changed = true;
            *Key_pressed = DebouncedKeyPress;
            // And reset the timer.
            if (DebouncedKeyPress) Count = RELEASE_MSEC / CHECK_MSEC;
            else Count = PRESS_MSEC / CHECK_MSEC;
        }
    }
}
```

- O que é race conditions ? • Como que essa forma de configurar os registradores evita isso?

3. Race Condition é um comportamento típico de sistemas e softwares eletrônicos, onde a saída depende da sequência ou do tempo de outros eventos não controláveis. Esse comportamento gera um bug quando os eventos não ocorrem de acordo com o que o programador previu. Com dois registradores o programador poderá setar saída de acordo com o planejado, tirando, assim a possibilidade de ocorrer race condition.