

1 Especificação do Sistema de Comando

O Sistema de Comando (SC) deve controlar o Sistema de Recuperação da Energia de Deslizamento (SRED) de modo a otimizar a energia retirada das ondas marítimas, sem afectar grandemente (através dos harmónicos e variações de potência entregue) a qualidade da rede eléctrica.

Como foi visto no Capítulo 5 o comando do SRED é efectuado através da tensão gerada no lado DC pelo conversor DC/AC (Edc) que impõe uma determinada velocidade de rotação e uma determinada potência que depende do binário disponível na turbina. A tensão Edc (Capítulo 4) é controlada através dos parâmetros do inversor: índice de modulação e fase.

Desta forma a principal acção do sistema de controlo é comandar o inversor de modo a retirar a potência pretendida e manter uma determinada velocidade.

A definição da potência a retirar do sistema e/ou a velocidade de rotação pode ser dada por um autómato ou calculada pelo SC através de um determinado algoritmo em função da velocidade de rotação.

As restantes acções do SC sobre o SRED consistem na aquisição das medidas necessárias ao controlo, no comando dos dispositivos auxiliares (disjuntores, contactores, protecções, etc) e na interacção com os sistemas hierarquicamente superiores a ele (autómato, computador pessoal e operador local).

Para além do controlo em regime permanente, o SC também tem de comandar de forma controlada e adequada as situações de excepção tais como:

- Arranque e paragem do sistema de forma programada e controlada,
- Protecção e paragem do sistema em caso de emergência ou avaria,
- Aviso dos respectivos agentes (operador local ou autómato) de falhas detectadas no sistema e aconselhamento de paragem se for caso disso.

Na Figura 6.1 é possível observar os agentes que vão interagir com o Sistema de Comando (SC) do Sistema de Recuperação da Energia de Deslizamento (SRED):

- Operador local,
- Computador pessoal,
- Autómatos,
- Sensores e actuadores do SRED.

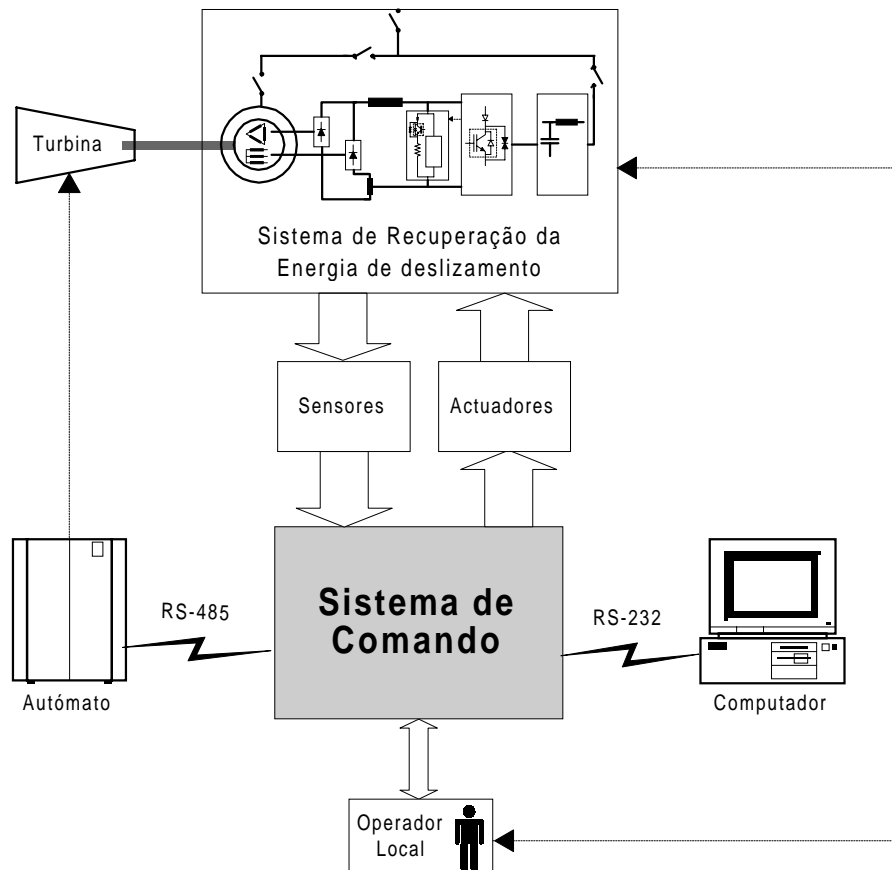


Figura 6.1 - Ambiente de interacção do Sistema de Comando do Sistema de Recuperação da Energia de Deslizamento para o aproveitamento da energia das ondas Oceânicas.

1.1 Autómatos

O autómato é responsável pela gestão e comando da central. Está numa posição hierarquicamente superior ao SC aqui analisado. O autómato dá ordem de arranque e paragem do sistema (que podem ser contrariadas pelo operador local), pede informações sobre o estado do SRED (velocidade, tensões, correntes, potência, alarmes, etc), e pode impôr a velocidade ou potência a ser retirada da turbina.

O SC informa o automático acerca da viabilidade de manter a central em funcionamento, dos alarmes que surgem, da acção que o operador local está a efectuar, etc.

A viabilidade de manter a central a funcionar é ditada pelo nível de potência disponível na turbina. Se a potência não for suficiente para manter o gerador a funcionar acima da velocidade de sincronismo, ou se for excessiva de tal forma que possa fazer ultrapassar a velocidade máxima (2 vezes a velocidade de sincronismo) o SC informa o automático que deve dar ordem de paragem do sistema.

Em caso de avaria do SRED o SC deixa de retirar energia da turbina e informa o automático que deve retirar a central de serviço abrindo o mais rapidamente possível a válvula de alívio da conduta de ar de modo a parar a turbina.

As comunicações com o automático têm de ser robustas e imunes ao ruído electromagnético inerente a uma central. O controlo lógico das comunicações é efectuado pelo automático que assume o papel de Master numa configuração Master-Slave.

1.2 Computador Pessoal

A necessidade da existência de um ambiente integrado e evolutivo de desenvolvimento implica o uso de um computador pessoal (PC). O software é realizado no PC sendo depois descarregado e gravado no SC.

O software do SC pode ser mudado muito rapidamente o que permite acelerar o desenvolvimento e teste do sistema.

Durante o desenvolvimento e/ou análise, o PC pode efectuar a recolha de dados do sistema (tensões, correntes, alarmes, etc) para serem analisados, permitindo uma avaliação do comportamento do sistema.

Depois de instalada a versão final¹ do software no SC, o PC serve para realizar a manutenção do SC, permitindo a realização de testes de diagnóstico e detecção de falhas no sistema. Por vezes a manutenção do sistema implica a mudança de software devido a *bugs* que só são detectados *no terreno*.

Atendendo a que o projecto se destina à investigação dos recursos energéticos marítimos, os parâmetros e métodos de controlo são modificados frequentemente para se verificar o comportamento do sistema e procurar o método que otimiza a utilização do recurso. A actualização dos parâmetros e métodos de controlo são facilmente modificados com o auxílio do PC.

1.3 Operador local

Para permitir a supervisão e controlo do sistema de uma forma local, é necessário implementar uma interface (painel com display, luzes de sinalização e botões de comando) que permita a leitura do estado do sistema e o seu controlo.

¹ Aqui entende-se versão final como a mais recente versão de software que está implementada e a funcionar *no terreno* na Central.

Naturalmente, é necessário salvaguardar a segurança no acesso e manipulação dos dados através de vários níveis de segurança. Deve ser dada atenção especial a ordens contraditórias vindas do operador local e do autómato.

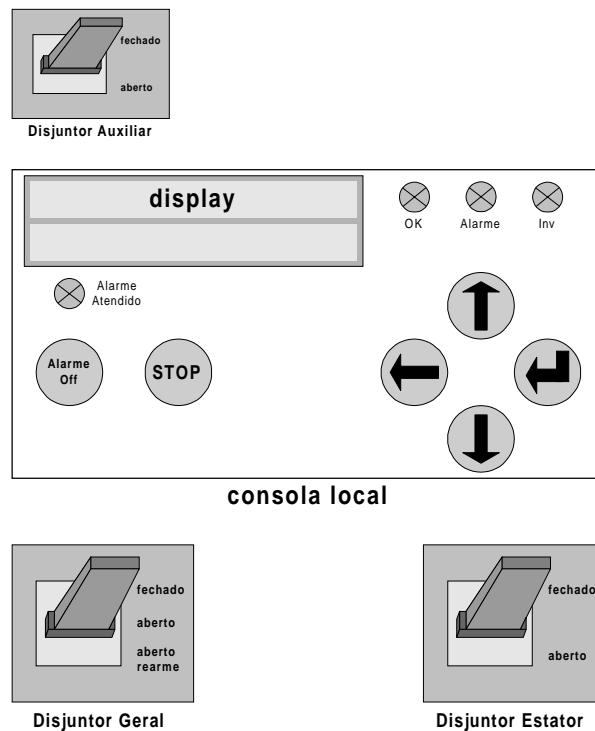


Figura 6.2 - Interface com o operador local.

Os disjuntores só podem ser rearmados e fechados pelo operador. Esta operação é realizada manualmente. Por questões de segurança, o disjuntor geral que liga todo o SRED à rede só pode ser fechado pelo operador depois do SC dar ordem de fecho ao respectivo disjuntor.

O operador tem total controlo manual sobre a abertura do disjuntor geral e sobre a abertura e fecho do disjuntor do estátor da máquina assíncrona.

O disjuntor auxiliar “liga” as cargas AC auxiliares (bobinas de fecho dos disjuntores e contactores)

Como é óbvio, os disjuntores devem ter o acesso limitado a pessoas autorizadas.

1.4 Interacção com o SRED

O comando do inversor é efectuado através dos sinais que controlam o estado de cada IGBT. Um sinal de erro gerado pelo *driver* de cada IGBT indica se este se encontra com defeito. Para colocar os *drivers* prontos a funcionar existe um sinal de *reset*. A sincronização da corrente injectada com a tensão da rede impõe a existência de um sinal que sinalize a passagem por zero da tensão da rede. (Se o sistema que mede as tensões da rede permitir realizar a detecção da passagem por zero com precisão o sinal individual e dedicado ao efeito não é necessário). Os sinais de *reset* e de erro dos seis IGBTs podem ser agrupados em 2 sinais: erro no inversor e *reset* do inversor uma vez que não é necessário controlo individual em cada IGBT deste

sinais. Um sinal de erro é gerado no caso do inversor exceder a temperatura máxima de funcionamento.

Um circuito no subsistema de protecção do inversor produz um sinal sempre que detecta uma sobretensão que pode existir à entrada do inversor em caso de avaria.

O subsistema de protecção possui um IGBT auxiliar que é fechado sempre que é necessário criar um caminho alternativo para a corrente no barramento DC. Este IGBT é comandado por um *driver* igual ao dos IGBTs do inversor pelo que possui 2 sinais de entrada: *reset* e comando, e um sinal de saída: erro no IGBT.

Os contactores do estátor e do rotor são actuados por sinais vindos do SC. O disjuntor geral só pode ser fechado manualmente e depois do SC activar um sinal a permitir o fecho.

Todos os interruptores estáticos (disjuntor geral, disjuntor estátor, contactor estátor e contactor rotor) possuem um sinal que notifica o SC acerca do seu estado. No caso dos contactores este sinal serve apenas para o SC ter garantia que a ordem dada ao respectivo contactor foi cumprida.

A velocidade de rotação do conjunto turbina-gerador é medida com um codificador incremental.

Várias medidas são necessárias para a monitorização e controlo:

- Tensões da rede (V_R , V_S e V_T),
- Correntes globais da rede (I_R , I_S e I_T),
- Correntes injectadas na rede pelo rotor (I_{RR} , I_{RS} , I_{RT}),
- Tensão gerada pelo rectificador no barramento DC (U_{DC}),
- Corrente que circula no barramento DC (I_{DC}).

Para monitorar as tensões de alimentação do próprio SC existe um sinal que o notifica no caso da alimentação exterior ter problemas. Por razões de segurança e redundância, a alimentação do subsistema de protecção do inversor deve ter uma alimentação separada, também com um sinal que notifique o SC em caso de falha. Para guardar a informação em caso de falha de alimentação é necessário um sistema de *backup* alimentado a pilha que naturalmente também deve ter um sinal que deve ser actuado quando a pilha estiver fraca.

Da máquina assíncrona chegam dois sinais de alarme: vibrações excessivas e temperatura acima do máximo admissível.

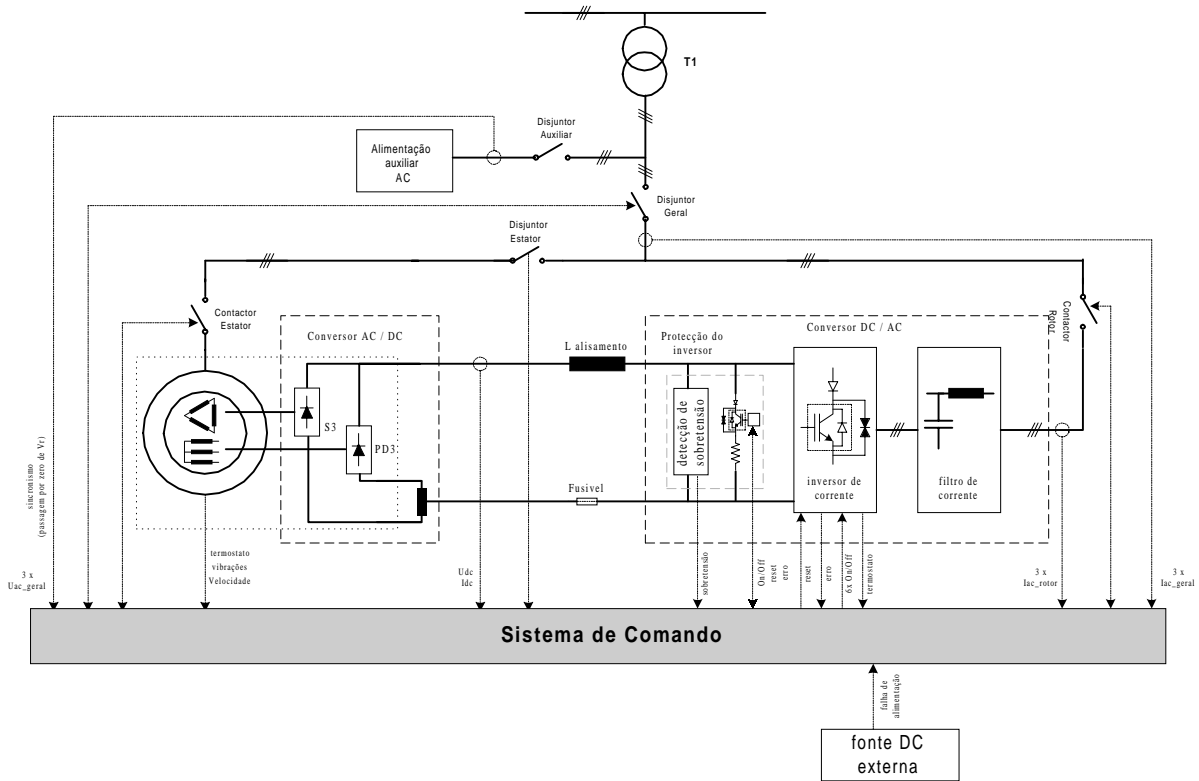


Figura 6.3 - Esquema geral dos sensores e actuadores existentes.

1.4.1 Sensores

Os sensores levam ao SC toda a informação sobre o estado do SRED necessária para o seu controlo e monitorização.

Parte da informação é do tipo digital:

- Estado dos contactores e disjuntores: aberto ou fechado,
- Estado das alimentações: Regulares ou Avariadas,
- Alarmes (termostato, erro no inversor, erro na protecção, etc): alarme activado ou não activado.

Outra parte é analógica:

- Tensões da rede (V_R , V_S e V_T),
- Correntes globais da rede (I_R , I_S e I_T),
- Correntes injectadas na rede pelo rotor (I_{RR} , I_{RS} , I_{RT}),
- Tensão gerada pelo rectificador no barramento DC (U_{DC}),
- Corrente que circula no barramento DC (I_{DC}).

pelo que necessita de ser convertida para informação digital através de circuitos de condicionamento e conversão Analógico-Digital.

A medida de velocidade é efectuada por um codificador incremental e não reúne um consenso na sua qualificação como sensor. Como é óbvio, a sua saída é do tipo digital (0 ou 1), mas a informação dos estados (0 ou 1) não tem qualquer valor individualmente. Apenas a contagem de transições durante um certo tempo ou a medição do tempo de duração de cada estado nos dá a informação da velocidade de rotação da máquina. Desta forma é preciso um sistema de conversão Impulsos-Digital para obter a velocidade de rotação da máquina.

A detecção de passagem por zero da rede também não é pacificamente qualificável. O sinal tem dois níveis (sinal digital) mas a informação requerida está na sua transição. Sempre que a fase R da tensão de rede passa por zero é gerada uma transição. Naturalmente o subsistema que trata da sincronização tem de ter capacidade para reconhecer a transição e actuar dentro de limites de tempo aceitáveis.

1.4.2 Actuadores

Todos os sinais que o SC envia ao SRED são do tipo digitais, pelo que o único cuidado a ter é a garantia de que o sinal tem potência suficiente para actuar no respectivo dispositivo e é imune ao ruído electromagnético.

2 Plataforma de hardware

As vantagens dos sistemas digitais programáveis face aos digitais discretos e aos analógicos são bem conhecidas, pelo que naturalmente o SC será feito com base numa máquina programável.

Pode-se dizer que a escolha da plataforma de hardware é a parte mais difícil de todo o trabalho.

Se um determinado *micro* serve ou não, se é melhor ou pior que outro, se facilita o desenvolvimento de software ou não, se é necessário usar uma plataforma baseada em vários micros ou não, etc, são questões que além de não terem uma resposta absoluta, são de difícil análise, particularmente porque é necessário defini-la no início do projecto quando ainda existem muitas dúvidas sobre a forma de controlar o sistema.

Neste trabalho colocaram-se algumas reservas relativamente ao facto de o comando do inversor de corrente do conversor DC-AC ser analógico ou digital, devido à rapidez com que é necessário actuar sobre ele. O controlador de corrente DC também suscitou algumas dúvidas pelo mesmo motivo.

Na realidade o comando digital directo dos IGBTs do inversor revelou-se um problema, uma vez que os *micros* existentes no mercado que possuem módulos dedicados ao controlo de inversores são preparados para inversores de tensão que têm uma lógica de comando diferente dos inversores de corrente.

A plataforma a utilizar deve possuir um poder de cálculo suficiente para realizar os algoritmos de controlo do SRED e interagir com o ambiente envolvente (autómato, computador, e operador local).

Desta forma colocaram-se basicamente 3 hipóteses gerais para a plataforma:

- Plataforma baseada num único *micro* que vai realizar todas as acções de comando e controlo incluindo a actuação directa sobre os IGBTs do inversor e a conversão A/D.
- Plataforma baseada num *micro* que realiza as acções de controlo de alto nível, dando ordens a outro micro (possivelmente um DSP) para controlar os IGBTs do inversor e a corrente no circuito DC.
- Plataforma baseada num PC industrial equipado com uma placa de aquisição de dados, uma placa de actuação e uma placa dedicada ao controlo do inversor (que seria semelhante ao circuito utilizado na hipótese anterior: um DSP)

A complexidade do software a desenvolver é maior no caso da primeira hipótese devido aos requisitos temporais impostos à actuação sobre os IGBTs condicionando o uso de Sistemas Operativos que simplificam o desenvolvimento de software. Esta questão prende-se com a inibição das interrupções do *micro* conforme é exposto no Capítulo 4.

Apesar da opção não ser pacífica, escolheu-se uma plataforma baseada num único *micro* (80C167 da Siemens) aproveitando as ferramentas de desenvolvimento disponíveis e o *know how* existente.

Para além das ferramentas de desenvolvimento disponíveis e o *know how*, a escolha recaiu sobre o μ C 80C167 principalmente devido às suas capacidades de processamento e controlo.

2.1 Plataforma utilizada

Depois de se tomar a opção de uma plataforma baseada num único μ C 80C167 coloca-se a questão de se desenvolver uma placa de raiz ou adquirir uma placa comercial baseada no 80C167 e adaptá-la ao caso em questão.

A disponibilidade no mercado de placas de processamento que apenas apresentam as funcionalidades suficientes para a construção de um sistema genérico, nomeadamente memória, comunicação série, entre outras, permite poupar desenvolvimento, focando o esforço de desenvolvimento no que é mais específico: a interface entre o sistema de controlo e o sistema a controlar.

A placa de processamento adquirida (MM167) já possui alguns dos componentes necessários ao controlador: micro-controlador, memória volátil e não volátil, comunicações série baseadas no protocolo RS232, controlo de falha de alimentação (comutação para pilha no caso de falha da alimentação primária), integrado dedicado à manutenção da data e da hora (RTC: Real Time Clock).

Esta placa disponibiliza em pinos para o exterior todos os pinos relevantes dos integrados que constituem a placa. Desta forma a placa MM167 pode ser vista como 'um grande integrado'.

Assim, para se concluir o hardware do controlador, apenas é necessário desenvolver uma placa que realize a interface entre o sistema em questão e o MM167, nomeadamente: Condicionamento dos sinais analógicos, Interface com o sinal de velocidade, Interface com os sinais digitais, Interface com o operador, Interfaces de comunicação, Circuito de actuação do PWM, Circuito de actuação das protecções, Alimentação do sistema.

2.1.1 Microcontrolador SAB80C167

Este microcontrolador (μC) faz parte da segunda geração de microcontroladores da família 80C16x, de 16 *bits*, da *Siemens*. Tal como qualquer outro μC tem um conjunto de blocos internos que implementam funcionalidades não encontradas nos processadores genéricos.

As grandes capacidades deste μC advêm, não só do seu grande poder de computação, mas também das funcionalidades que permitem construir um sistema mais facilmente, desenvolvendo menos hardware e menos software. Estas funcionalidades são desenvolvidas a partir do conjunto de blocos internos do μC , dos quais se destacam:

- Conversor analógico/digital, com 16 canais e tempo de aquisição e conversão programáveis;
- Unidade de modulação de largura de impulsos (PWM - *Pulse Width Modulation*), com quatro canais;
- Duas unidades de temporizadores, com cinco temporizadores no total;
- Duas unidades de captura/comparação, com bases de tempo independentes, permitindo quatro temporizadores (CAPCOM 1 e 2);
- Unidade de comunicação série, síncrona ou assíncrona (USART - *Universal Synchronous Asynchronous Receiver/Transmitter*);
- Unidade de comunicação série síncrona de alta velocidade (SSC - *Synchronous Serial Channel*);
- Temporizador de supervisão do sistema(WDT - *Watchdog Timer*);
- Unidade que permite carregar o software de inicialização através de comunicação série (BL - *Bootstrap Loader*);
- Sistema de interrupções hierárquico, com possibilidade de interrupções rápidas externas;
- Sistema de transferência de dados independente do processamento, e orientado por interrupção (PEC - *Peripheral Event Controller*);

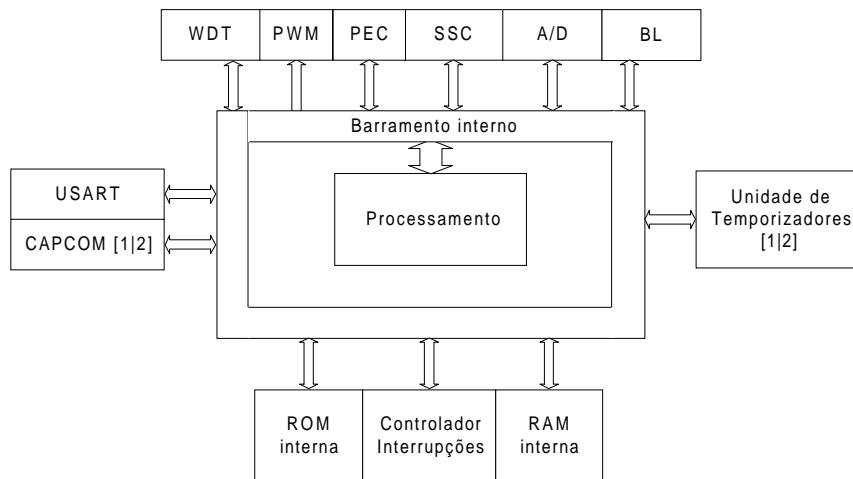


Figura 6.4 - Esquema de blocos do microcontrolador

Acresce ainda que existem neste μC certas características que são também importantes para o desenvolvimento de sistemas embebidos, e que explicam a sua especificidade para eles:

- Múltiplos bancos de registos, permitindo uma mais fácil mudança de contexto;
- Suporte à mudança de contexto num só ciclo de relógio;
- Manipulação de dados de vários tipos, incluindo o *bit*;
- Suporte a excepções, tanto de hardware, como de software;
- Suporte à implementação de semáforos, e à exclusão no acesso a recursos.

Torna-se assim claro que, a partir destas funcionalidades e características apresentadas, este μC seja excelente para aplicação em sistemas embebidos, permitindo um mais simples desenvolvimento da restante arquitectura de hardware, e consequentemente, do software.

A leitura do manual do C167 é indispensável para uma total compreensão do controlador desenvolvido neste trabalho, principalmente na construção do software. De qualquer modo apresentam-se a seguir algumas das características do μC que mais contribuíram para a estrutura do software.

2.1.1.1 Endereçamento

Os 16Mbytes de memória endereçável estão divididos em 256 segmentos de 64Kbytes. Cada segmento é dividido em 4 páginas de 16Kbytes.

Para se aceder a uma posição de memória é necessário indicar a página respectiva e o offset em relação ao início da página.

O C167 possui 4 registos dedicados (DPP0 .. DPP3) para apontar as páginas.

O registos DPP são usados implicitamente sempre que são efectuados acessos à memória através de endereçamentos directos ou indirectos de 16 bits.

A paginação é efectuada através da concatenação dos 14 bits menos significativos do endereço directo ou indirecto de 16 bits (Ende) com o conteúdo do registo DPP seleccionado através dos bits nº 15 e 14 de Ende. Desta forma os 13 bits menos significativos de Ende indicam o offset dentro da página apontada pelo registo DPP respectivo.

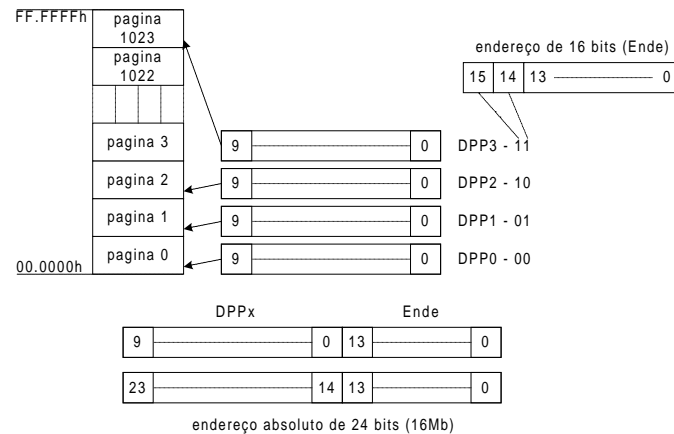


Figura 6.5 - Endereçamento via registos DPP.

A utilização implícita dos registos DPP para apontadores de página pode ser sobreposto com a utilização de duas instruções: EXTP (EXTended Page) e EXTs (EXTendend Segment).

A instrução EXTP leva o C167 a ignorar os bits nº15 e 14 de Ende, e a utilizar os restantes bits como o offset para uma página indicada na respectiva instrução. Este modo de endereçamento é semelhante ao anterior com a excepção de que as páginas não são indicadas por um registo DPP.

A instrução EXTs utiliza todos os bits de Ende para indicar o offset dentro de um segmento (= 4 páginas) indicado na respectiva instrução.

Naturalmente o uso das instruções EXTP e EXTs para sobrepor a utilização implícita dos registos DPP leva a um maior tempo de execução.

O C167 não possui um sistema de endereçamento dedicado a entradas e saídas de dados pelo que o acesso a todos os periféricos (portas, temporizadores, comunicações, etc) é efectuada através de mapeamento em memória.

O C167 é um micro de 16bits com a memória organizada ao byte. As operações de 16bits são mais rápidas e eficientes do que as de 8bits (devido à sua organização e arquitectura) pelo que sempre que for possível, devem-se realizar operações de 16bits (a não ser que seja necessário poupar memória).

2.1.1.2 Registos de processamento

O C167 tem uma arquitectura RISC pelo que não possui registos dedicados ao cálculo aritmético e lógico. Em vez disso possui uma zona de memória dentro da RAM interna onde cada posição de memória pode ser utilizada como registo de cálculo (SFR).

Em cada instante o C167 pode operar sobre 16 SFR (R0 .. R15) de 16 bits. A sua localização dentro da RAM interna é indicada pelo registo especial CP (Context Pointer).

Desta forma para se mudar o contexto de processamento (no atendimento de interrupções por exemplo) basta mudar o conteúdo do registo CP.

Os vários contextos podem-se sobrepor, i.é.: algumas das posições de memória indicadas por um CP podem ser sobrepostas por outras indicadas por outro CP.

Apenas os SFR 0 a 7 podem ser endereçados ao byte.

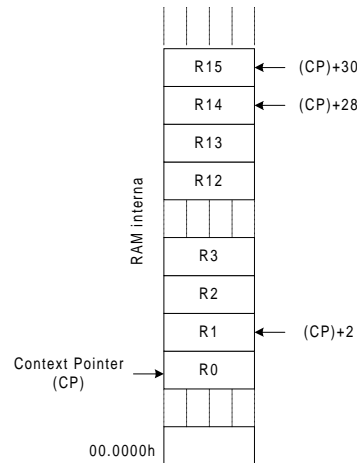


Figura 6.6 - Estrutura dos registos de operação: SFR.

2.1.1.3 Stack

O C167 possui uma *system stack* que é gerida por hardware. A *stack* é protegida contra acessos ilegais através de dois apontadores que delimitam as fronteiras da *stack*. Sempre que é efectuado um acesso fora desses limites é gerada uma interrupção de “stack underflow” ou “stack overflow”.

Existem operações dedicadas a guardar e a retirar dados da *system stack* que incrementam ou decrementam o *stack pointer* verificando simultaneamente a integridade da *stack*.

Todos os endereços de retorno guardados devido a chamadas de rotinas e os contextos do CPU entre interrupções são guardados por hardware na *system stack*.

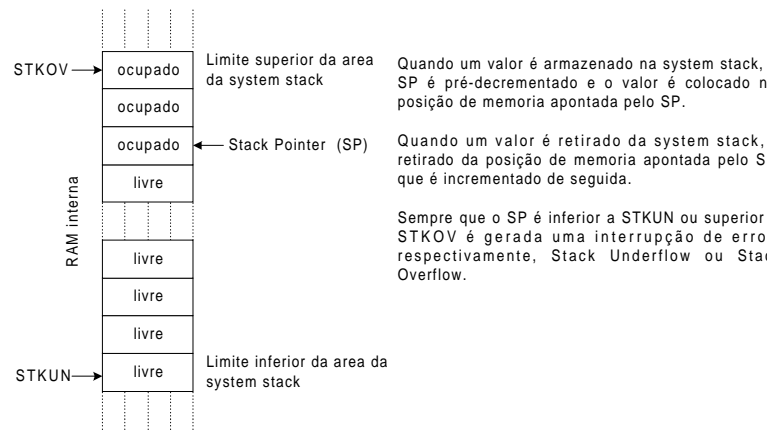


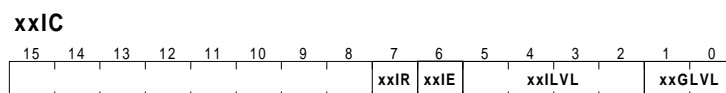
Figura 6.7 - Protecção da *system stack* a acessos ilegais.

2.1.1.4 Sistema de interrupções

Um esquema de prioridades de interrupções permite especificar com 16 níveis qual é a prioridade de uma dada interrupção.

Cada interrupção possui um registo (*xxIC*) que identifica o nível de prioridade (*ILVL*), o nível de grupo (*GLVL*), se está inibida ou não (*IE*) e se a interrupção está à espera de ser atendida ou não (*IR*).

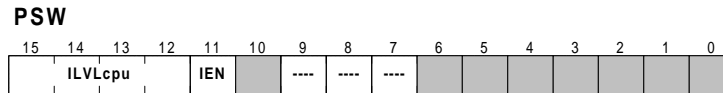
O nível de grupo *GLVL* serve para no caso de várias interrupções com o mesmo nível surgirem ao mesmo tempo, identificar qual é a que é atendida em primeiro lugar (actua como um segundo nível de arbitragem de prioridades). No entanto as interrupções com o mesmo nível (*ILVL*) nunca se interrompem umas às outras.



- xxGLVL** -> **nível de grupo**
define a ordem de atendimento para interrupções simultâneas com a mesma prioridade.
3: maior prioridade de grupo
0: menor prioridade de grupo
- xxILVL** -> **nível de prioridade de interrupção**
define o nível de prioridade.
15: prioridade mais elevada
0: prioridade mais baixa
- xxIE** -> **bit de controlo de permissão de interrupção**
define se a interrupção está activa ou não
0: a interrupção não tem permissão para ser atendida
1: a interrupção pode ser atendida
- xxIR** -> **bit de controlo de sinalização de pedido de interrupção**
define se existe alguma interrupção pendente ou não
0: não existe interrupção à espera
1: existe uma interrupção pendente

Figura 6.8 - Registo individual de controlo das interrupções do C167 (interrupção nº *xx*).

O registo especial do C167 (*PSW* - Processor Status Word) permite a inibição de todas as interrupções (à excepção das interrupções de sistema) e indica a prioridade sobre a qual a actual rotina está a ser executada.



ILVLcpu -> **nível de prioridade com que a actual rotina está a ser executada**
define as interrupções que podem interromper a actual rotina através da sua prioridade

xxIE -> **bit de controlo de permissão global das interrupções**
define globalmente se as interrupções podem ser atendidas ou não
0: as interrupções não tem permissão para serem atendidas
1: as interrupções podem ser atendidas

Figura 6.9 - Registo especial PSW (Processor Status Word) do C167.

Quando surge uma interrupção (xx), o bit IR do registo xxIC é sinalizado para informar o sistema de controlo de interrupções do pedido de execução. A interrupção é atendida se se verificarem as seguintes condições:

- O bit IR de xxIC está sinalizado;
- A prioridade da interrupção (ILVL) é maior que a prioridade actual do CPU (ILVLcpu);
- O bit IE de xxIC está sinalizado;
- O bit IEN de PSW permite as interrupções.

Se a interrupção for atendida o nível de prioridade do CPU é colocado no nível da interrupção respectiva de forma a garantir que as interrupções de mais baixo nível não a interrompem. No final do atendimento da rotina de interrupção o nível de prioridade do CPU anterior é reposto e o bit IR de xxIC é limpo automaticamente por hardware.

2.1.1.4.1 Interrupções de sistema

O C167 possui algumas interrupções que não são mascaráveis e que são sempre atendidas independentemente do nível de prioridade do CPU:

- Interrupções de reset
 - RESET - Hardware reset (interrupção externa)
 - RESET - Software reset
 - RESET - Watchdog Timer Overflow
- Interrupções de classe A de hardware
 - NMI - interrupção não mascarável (interrupção externa)
 - STKOF - Stack Overflow
 - STKUF - Stack Underflow
- Interrupções de classe B de hardware

- BTRAP - Acesso ilegal ao barramento externo,
- - Acesso ilegal a instrução,
- - Código de operação não definido,
- - Falha na execução de instrução protegida,
- - Acesso ilegal a operador com alinhamento WORD.

2.1.1.4.2 Processamento normal de interrupções

O CPU suspende temporariamente a execução do programa e salta para a rotina de atendimento da interrupção. Alguns registos de controlo (apontador de instrução, estado de processador, etc) são guardados na *system stack*. O endereço da rotina da interrupção “xx” a ser executada é indicado na posição “xx” do vector de interrupções.

2.1.1.4.3 PEC

As interrupções com um nível de prioridade igual ou superior a 14 podem ser atendidas de um modo especial: transferência via PEC - Peripheral Event Controller. Com dois níveis de prioridade e 4 níveis de grupo por cada nível de prioridade existem 8 canais de transferência via PEC.

Neste caso em vez da interrupção dar origem à execução de uma rotina, provoca uma transferência de um byte ou um *word* de uma dada posição de memória para outra. Não existe mudança de contexto nem qualquer processamento adicional no CPU. A transferência é executada pelo controlador de interrupções que executa uma transferência por acesso directo à memória (DMA - direct memory access) do micro. Desta forma o atendimento da interrupção apenas atrasa o CPU em um ciclo máquina.

As transferências via PEC só são possíveis com posições de memória no segmento 0 (páginas 0..3).

Cada canal “x” PEC possui um registo de controlo (xPECC) que define o modo de operação do canal, e dois apontadores (destino e origem) que definem as posições de memória envolvidas na transferência.

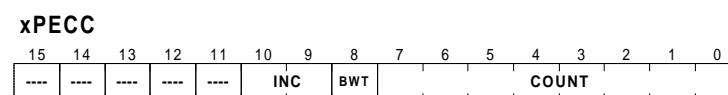


Figura 6.10 - Registo de controlo do canal “x” de transferência via PEC

Durante as transferências, o valor dos apontadores destino e origem são modificados (incrementados ou não) de acordo com o registo INC. O bit BWT define se a transferência é um byte ou um *word*.

Sempre que existe uma interrupção do canal “x” de PEC, a acção a ser executada depende do valor de COUNT no registo xPECC.

Se COUNT for diferente de FFh e 00h, é executada uma transferência e o valor de COUNT é subtraído de 1h.

Se COUNT for igual a 00h, em vez de realizar uma transferência via PEC, é executada uma rotina como se a interrupção fosse normal. Isto permite executar inicializações dos apontadores, de COUNT, e efectuar um eventual tratamento de informação já transferida. Naturalmente se em vez de se querer utilizar o canal de transferência via PEC, se quiser utilizar o respectivo nível de prioridade como uma interrupção convencional, o valor de CONT tem de ser colocado a 00h.

Se COUNT contiver o valor FFh, a acção consiste apenas na realização de transferência (número de transferências ilimitadas)

As transferências via PEC são muito úteis para se realizar transferências rápidas entre a memória e os periféricos do C167 tais como o conversor A/D, as portas série, a comunicação CAN, etc.

2.1.1.5 Conversor analógico/digital

O C167 está equipado com um conversor Analógico-Digital (ADC) de 10bits de resolução e circuitos de sample&hold. Um multiplexador interno permite seleccionar até 16 canais independentes de entrada.

O tempo de conversão e o tempo de amostragem (*Sample*) são programáveis. Uma conversão completa demora:

$$T_{total} = 10 \cdot tcc + 2 \cdot tsc + 4 \cdot Tcl \quad (6.1)$$

onde tcc é o período do relógio do conversor, tsc do circuito de amostragem e Tcl do relógio do sistema.

Com tcc e tsc no mínimo e Tcl=1/40Mhz, uma conversão completa demora 8.15µs.

Para cumprir a maior parte dos requisitos impostos pela generalidade dos sistemas embebidos o ADC permite o funcionamento em vários modos de operação¹:

- *Fixed Channel Single Conversion*,
Produce apenas uma conversão de um único canal.
- *Fixed Channel Continuous Conversion*,
Produce repetidamente conversões de um dado canal.
- *Auto Scan Single Conversion*,
Produce uma conversão para cada canal de um dado grupo de canais.
- *Auto Scan Continuous Conversion*,
Produce repetidamente conversões para cada canal de um dado grupo de canais.

¹ Optou-se por indicar o modo de funcionamento na linguagem nativa do manual do C167 atendendo a que as traduções nem sempre são felizes.

- *Wait for ADDAT Read Mode,*
Inicia automaticamente uma nova conversão quando o resultado da conversão anterior for lido.
- *Channel Injection Mode,*
Insere uma conversão de um dado canal durante a conversão de um conjunto de canais (*auto scan*).

Devido à complexidade e ao número de opções existentes na configuração do ADC, apenas se dá uma breve explicação do modo utilizado neste trabalho: *Auto Scan Single Conversion*.

Neste caso quando o bit ADST é sinalizado, o ADC converte um número “n+1” de canais (canal n até canal 0). O número “n” é indicado no campo de bits “ADCH” no registo ADCON.

Sempre que uma conversão de um canal “x” é finalizada, o resultado é colocado no registo ADDAT e a conversão do canal “x-1” começa a realizar-se (até que se converte o canal 0). Sempre que o registo ADDAT recebe um novo valor é gerada uma interrupção (ADCIC) que deve ler o resultado (a interrupção pode ser convencional ou via PEC conforme a sua programação). Sempre que um valor é guardado em ADDAT sem que o anterior tenha sido lido é gerada uma interrupção (ADEIC) para sinalizar o erro.

2.1.1.6 Entradas / Saídas digitais

Os pinos do C167 têm geralmente várias funções alternativas que são programadas e seleccionadas por software. Estas funções incluem o barramento de dados, o barramento de endereços, gestão de barramento no caso de plataforma multiprocessador, *trigger* e comando de contadores e temporizadores, saídas de PWM, entrada analógica para conversor A/D, etc.

Quando a função alternativa de um dado pino não é necessária, esse pino pode ser usado como uma porta de entrada ou saída digital. O modo de funcionamento (entrada/saída; totem pole / open collector) da porta é programado por software.

2.1.2 MM167

Foi escolhida uma placa comercial de processamento (MM167 da Phytec) que permite o encaixe noutra placa de circuito impresso, de modo a tornar possível e fácil a conexão entre esta e a placa de interface.

Deste modo o MM167 pode ser visto como um único circuito integrado, que é utilizado como qualquer outro.

O MM167 é constituído por:

- Micro-Controlador C167 da Siemens
- 1Mb Flash-EPROM
- 256Kb RAM;

- UART, ligada a uma porta paralela do μ C, de modo a disponibilizar duas portas séries assíncronas;
- Interface TTL/RS-232;
- Relógio de tempo real;
- Supervisor de falha de alimentação.

O MM167 permite cumprir a totalidade dos requisitos da aplicação em termos de processamento.

A acessibilidade dos pinos do μ C, e destes periféricos, nos barramentos que realizam o encaixe na placa de interface, torna desnecessária a implementação de outro barramento, especialmente porque não se prevê necessária uma interface normalizada com outros sistemas de processamento.

Não se faz aqui uma descrição dos pinos do MM167 porque seria demasiado exaustivo devido ao grande número de funções que cada pino pode realizar. As funções que cada pino desempenha neste trabalho são facilmente deduzidas através dos esquemas na placa de interface.

2.1.3 Placa de interface

A placa de interface apenas tem que implementar os circuitos necessários para a adaptação da placa de processamento (MM167) ao sistema a controlar. Estes são:

- Condicionamento dos sinais analógicos;
- Interface com o sinal de velocidade;
- Interface com os sinais digitais;
- Interface com o operador;
- Interfaces de comunicação;
- Circuito de actuação do PWM;
- Circuito de actuação das protecções;
- Alimentação do sistema.

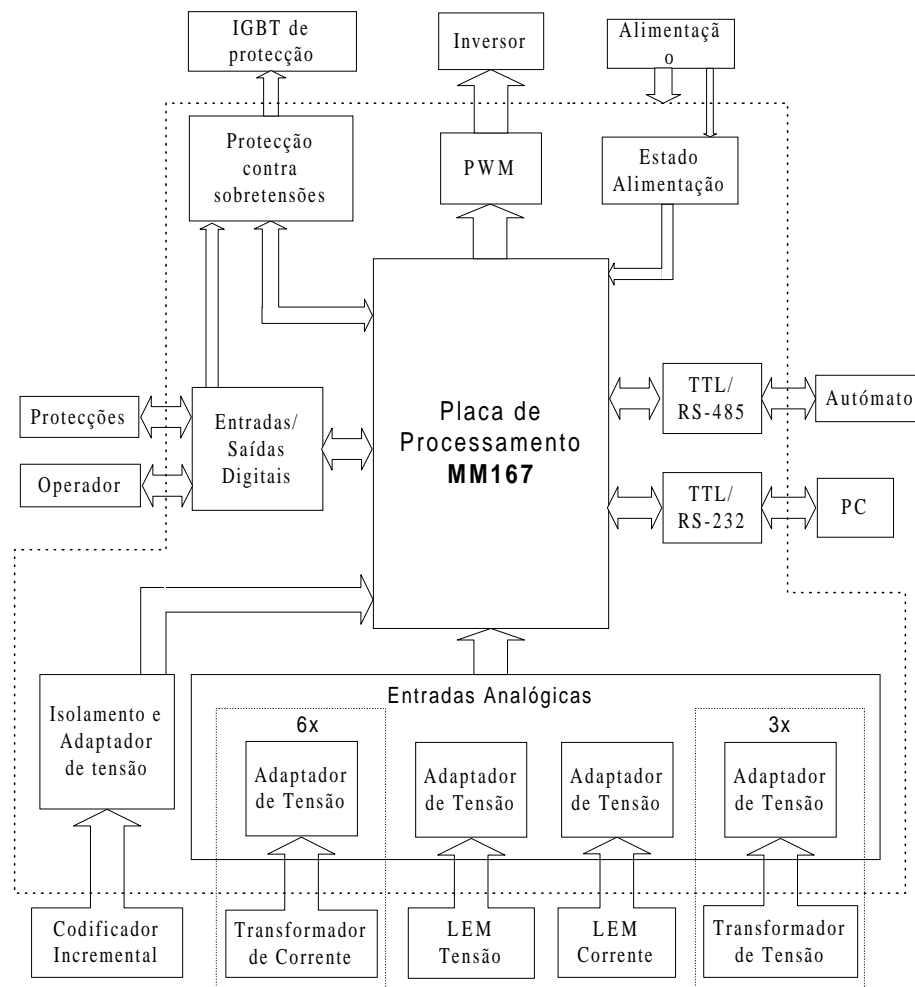


Figura 6.11 - Esquema de Blocos da Placa de Interface

2.1.3.1 Condicionamento dos sinais analógicos

As várias medidas analógicas necessárias ao controlo têm de ser adaptadas de forma a ser possível convertê-las no A/D do C167.

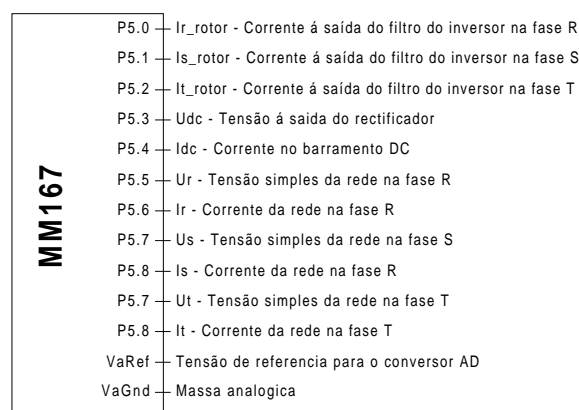


Figura 6.12 - Relações entre as medidas e as portas do conversor A/D do C167.

2.1.3.1.1 Tensão de referência do conversor Analógico/Digital

É necessário fornecer um sinal de referência ao conversor A/D do C167. Isto é realizado com um integrado dedicado ao efeito e *buffers*. A referência V_{A-} serve para os circuitos de condicionamento das medidas alternadas (AC).

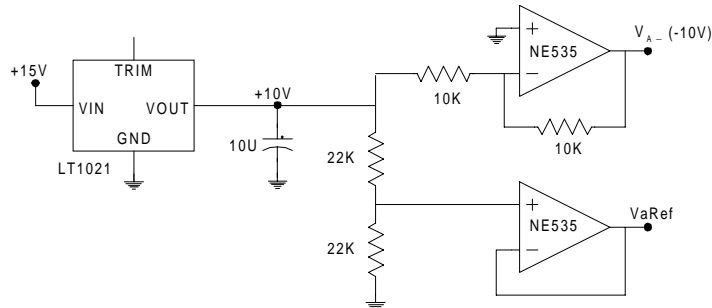


Figura 6.13 - Circuito gerador de referencias

2.1.3.1.2 Tensão e corrente DC

A medida de corrente no barramento DC (I_{dc}) é realizada com um LEM de 400A de ganho 1:5000. A sua saída em corrente tem de ser convertida para uma tensão apropriada ao A/D do μC . Devido às elevadas frequências existentes em I_{dc} , é necessário filtrar o sinal.

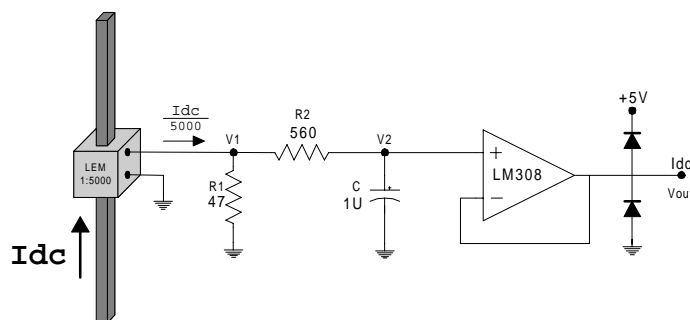


Figura 6.14 - Condicionamento da medida da corrente no barramento DC

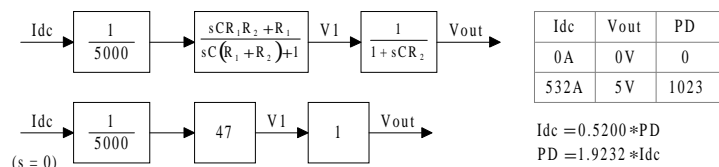


Figura 6.15 - Diagrama de blocos do sistema de medida e relação entre a corrente I_{dc} e a palavra digital (PD) resultante da conversão A/D.

A medida da tensão DC (U_{dc}) é feita através de um LEM de tensão. Para adaptar o LEM ao nível de tensões medidas é colocada em série uma resistência de 66K. A saída em corrente é condicionada por um circuito semelhante ao da medida de I_{dc} .

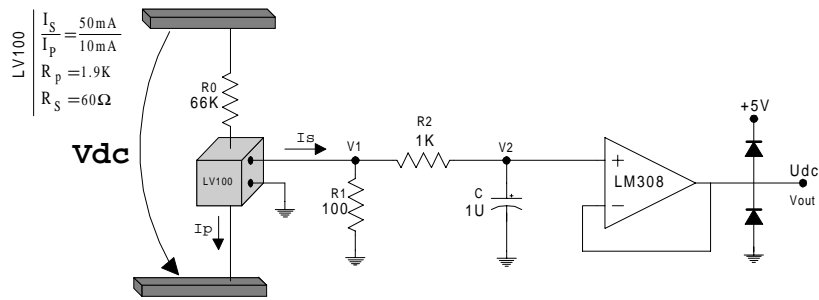


Figura 6.16 - Condicionamento da medida da tensão no barramento DC

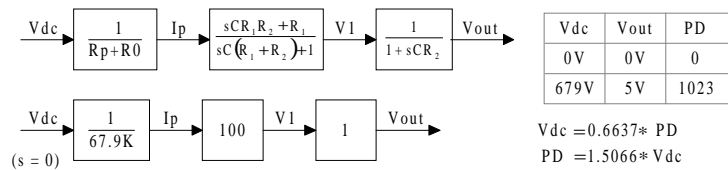


Figura 6.17 - Diagrama de blocos do sistema de medida e relação entre a tensão Udc e a palavra digital (PD) resultante da conversão A/D.

2.1.3.1.3 Tensões da rede

As tensões simples da rede são medidas a seguir ao disjuntor auxiliar por três transformadores de tensão. Como o A/D não converte tensões negativas é necessário introduzir um *offset* no sinal. O sinal esperado é uma sinusóide a 50HZ pelo que apenas é realizada uma ligeira filtragem para evitar alguns picos de tensão.

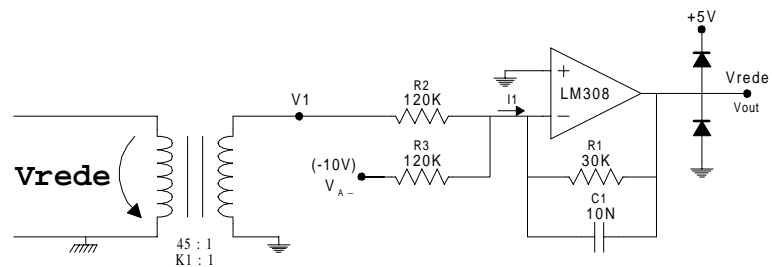


Figura 6.18 - Condicionamento da medida da tensão na rede (por fase)

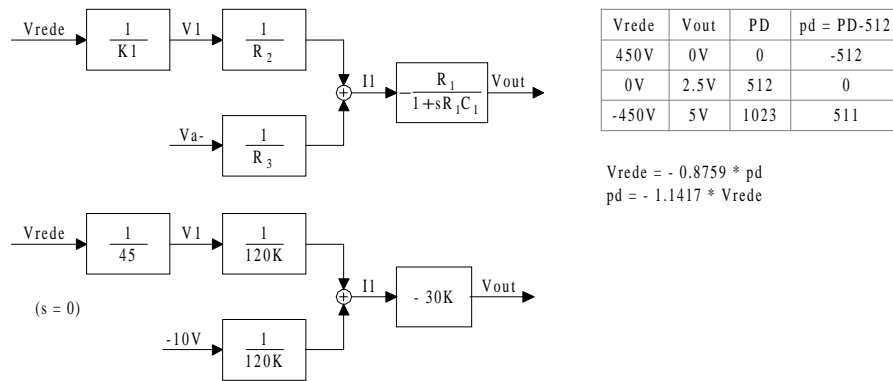


Figura 6.19 - Diagrama de blocos do sistema de medida e relação entre as tensões da rede e a palavra digital (PD) resultante da conversão A/D.

2.1.3.1.4 Correntes na rede e Correntes no rotor.

As correntes no rotor são medidas a seguir ao filtro de corrente. Tanto as correntes no rotor como as da rede devem ser sinusóides pelo que apenas é realizada uma pequena filtragem devido ao ruído.

As correntes na rede são medidas com um transformador de corrente com ganho de 800:5 e as correntes no rotor com um de 400:5. O sinal em corrente é convertido em tensão através de resistências.

O circuito de condicionamento é igual para os dois casos. Possui um andar de isolamento de carga e um andar para introduzir o *offset* requerido.

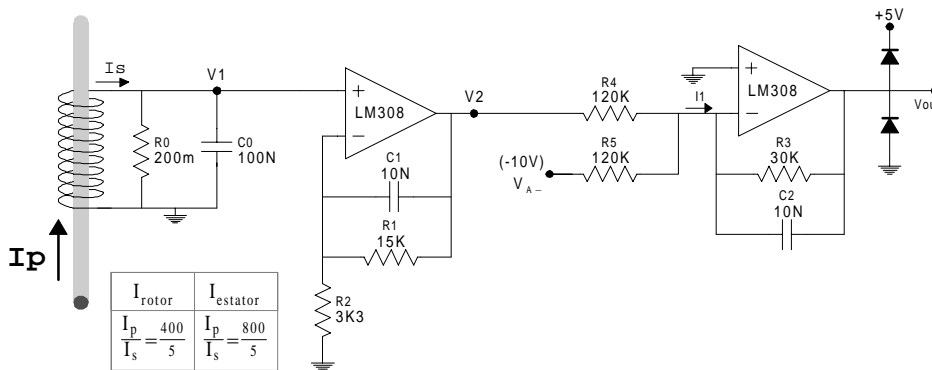


Figura 6.20 - Condicionamento das medidas de corrente.

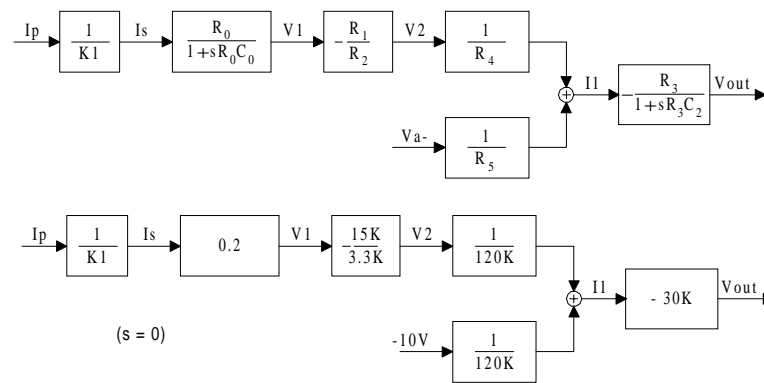


Figura 6.21 - Diagrama de blocos do sistema de medida.

Irotor	Vout	PD	pd = PD-512
725A	0V	0	-512
0V	2.5V	512	0
-725A	5V	1023	511

Irotor = - 1.42 * pd
pd = - 0.704 * Irotor

Figura 6.22 - Relação entre as correntes do rotor e a palavra digital (PD) resultante da conversão A/D.

Irede	Vout	PD	pd = PD-512
1450A	0V	0	-512
0V	2.5V	512	0
-1450A	5V	1023	511

Irede = - 2.83 * pd
pd = - 0.353 * Irede

Figura 6.23 - Relação entre as correntes da rede e a palavra digital (PD) resultante da conversão A/D.

2.1.3.2 Sinal de velocidade

A velocidade da máquina é medida com um codificador incremental. O codificador gera 2 sinais (ondas quadradas) em quadratura com 1024 impulsos por cada revolução do conjunto turbina/gerador.

Os sinais em quadratura servem para além de medir a velocidade de rotação, indicar o sentido. Como neste caso a máquina roda sempre no mesmo sentido, esta característica não é utilizada. No entanto com estes dois sinais é possível gerar um terceiro com o dobro do número de impulsos, duplicando assim a resolução da medida.

Desta forma o circuito de condicionamento de sinal, além de fornecer isolamento eléctrico, também tem de construir o sinal a partir das duas saídas do codificador. A construção do sinal consiste apenas num “XOR” lógico entre os sinais originais do codificador. O C167 recebe assim 2048 impulsos por cada revolução do conjunto turbina-gerador.

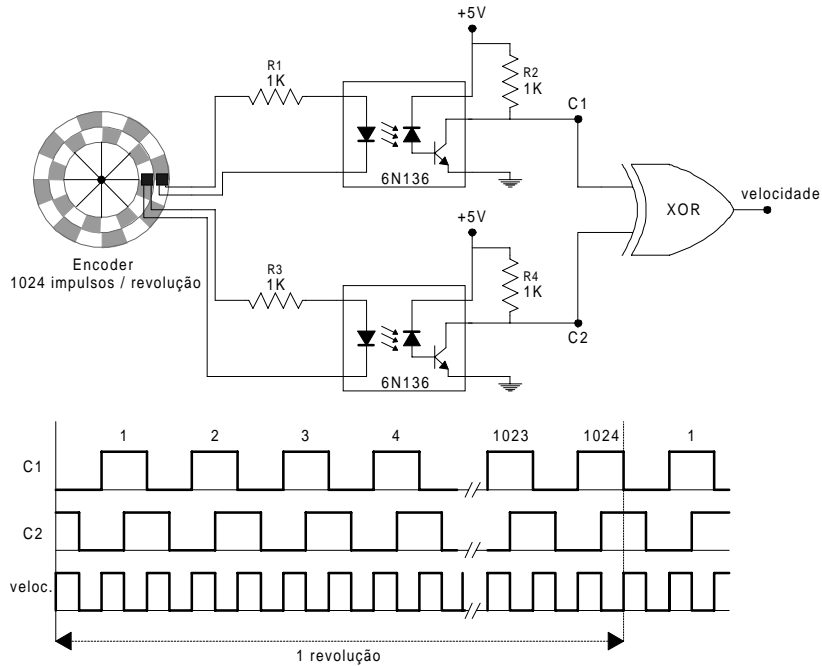


Figura 6.24 - Esquema de condicionamento do sinal de medida de velocidade.

O sinal com os impulsos de velocidade é ligado a um pino do C167 (P3.7) que controla um dos temporizadores/contadores (T2). Em princípio, será usado o método de contagem de número de impulsos durante um intervalo de tempo fixo, pelo que o T2 deve ser configurado como contador e o pino P3.7 deve originar um incremento de T2 sempre que houver uma transição à sua entrada (ascendente ou descendente).

2.1.3.3 Detecção da passagem por zero da tensão da rede

A origem das tensões da rede é determinada através da passagem por zero no sentido ascendente da tensão simples da fase R.

O circuito é composto por um filtro que reduz o ruído, diminuindo as falsas detecções, e por um comparador com histerese.

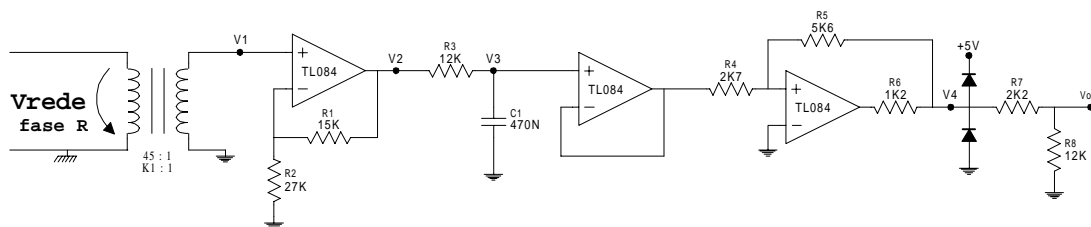


Figura 6.25 - Circuito de detecção da passagem por zero da tensão da fase R da rede.

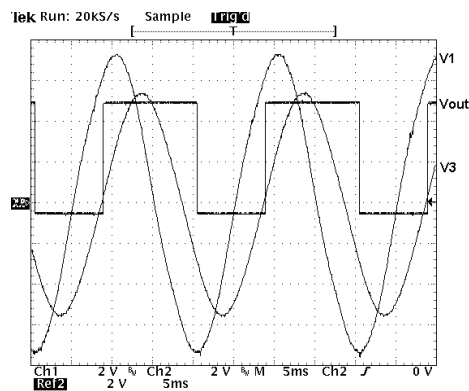


Figura 6.26 - Formas de onda do circuito de detecção da passagem por zero da tensão da fase R da rede.

O C167 deve reagir à transição ascendente do sinal gerado pelo comparador e compensar o atraso (introduzido pelo filtro) da respectiva transição face à verdadeira passagem por zero da rede.

Como é óbvio, o sinal é ligado a um pino do C167 que permita a detecção de interrupções exteriores (P2.0).

2.1.3.4 Comando dos disjuntores e contactores

Os contactores do rotor e do estator são actuados (ligados e desligados) através da sua bobina de comando. A bobina é conduzida por relés controlados pelo C167.

O disjuntor geral recebe ordem de fecho da mesma forma que os contactores, mas só pode ser realmente fechado manualmente e depois de ser rearmado. Naturalmente o controlador pode abrir o disjuntor deixando de enviar a ordem de fecho.

O estado (aberto ou fechado) dos contactores do rotor e do estator e dos disjuntores geral e estator é realimentado para o controlador tomar conhecimento e detectar erros. Esta realimentação é efectuada através de um contacto (interruptor) auxiliar que existe em cada dispositivo. É realizada uma filtragem (através de uma capacidade) para eliminar o ruído nas linhas, diminuir o *bounce* dos contactos e evitar sobretensões na placa de controlo.

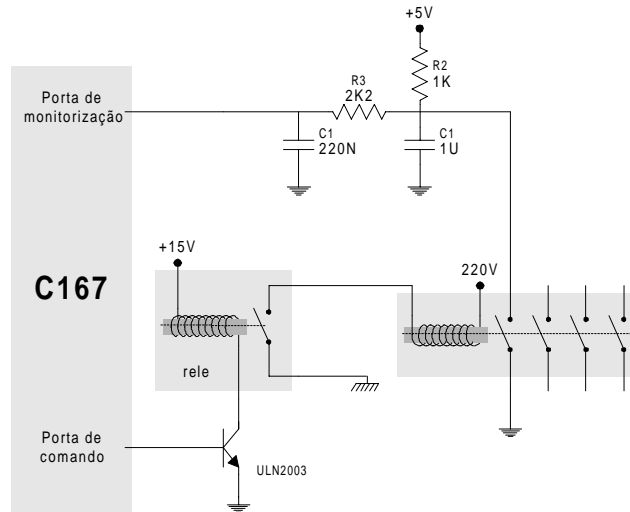


Figura 6.27 - Esquema genérico do comando e monitorização dos disjuntores e contactores.

Não é efectuada a realimentação do estado do disjuntor auxiliar porque no caso de ele abrir as 3 tensões da rede medidas são nulas, sendo assim detectada a sua abertura (e originando uma acção correctiva da falha - provavelmente desligar o sistema).

2.1.3.5 Consola local para operador

A consola consiste num display de 2 linhas de 40 caracteres, 6 botões e 4 *leds* de sinalização.

O display pode ser ligado com 8 ou 4 linhas de dados e o controlo da comunicação é realizado por 3 linhas: EN, RS e RW. Como a velocidade de controlo do display não é critica e para reduzir o número de pinos utilizados, é ligado com 4 linhas de dados. Deste modo uma transferência de um byte é feita em dois ciclos.

O contraste do display é ajustado com um potenciómetro.



Figura 6.28 - Ligação do display ao MM167.

Os *leds* são ligados ao MM167 através de um *buffer* para amplificação de corrente.

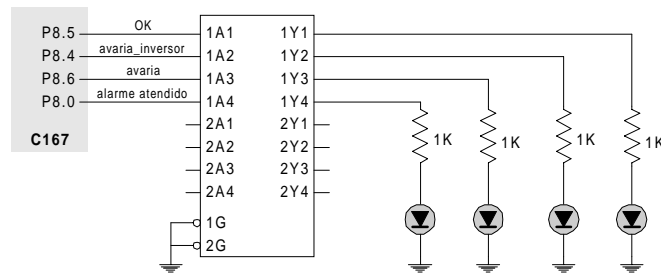


Figura 6.29 - Ligação dos leds ao MM167.

Os botões são ligados ao MM167 com uma filtragem para reduzir o bounce.

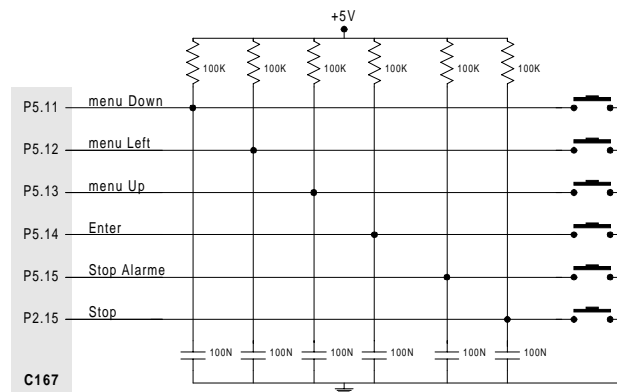


Figura 6.30 - Ligação dos botões ao MM167.

2.1.3.6 Interfaces de comunicação

As comunicações com PC são realizadas com a porta Série¹ interna do C167 que é ligada a um conversor TTL/RS232 existente na placa do MM167.

As comunicações com o Autômato são realizadas com uma UART existente na placa do MM167.

A UART possui um sinal que pode ser usado de várias formas (dependendo da configuração por software da UART) incluindo a sinalização de que está pronta para enviar dados, ou que recebeu dados novos. Este sinal é ligado a um pino (P2.9) do C167, capaz de reconhecer interrupções externas, de modo a que o sistema de comunicações com o autômato possa funcionar orientado à interrupção.

É usado o RS485 half-duplex para comunicar com o Autômato. Desta forma é utilizado um conversor TTL/RS485 para adaptar o sinal à transmissão e recepção.

Um sinal adicional (EN_RS485) define o modo de comunicação no canal de RS485: transmissão ou recepção.

¹ Esta é a porta série que realiza o Boot-Strap Load do C167.

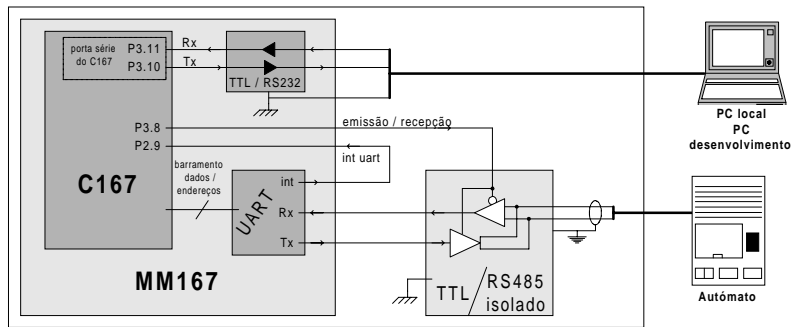


Figura 6.31 - Esquema das comunicações com o exterior.

2.1.3.7 Sinais de alarme/protecção da máquina assíncrona

Os alarmes com origem na máquina são filtrados de modo a eliminar o ruído induzido e a proteger a portas do μC contra sobretensões.

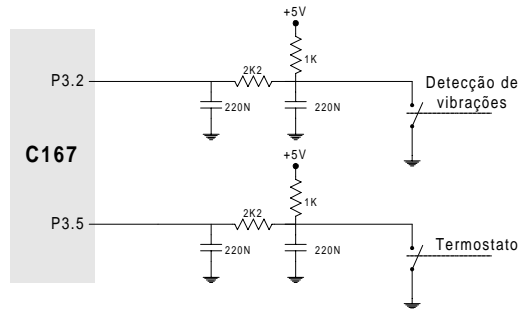


Figura 6.32 - Alarmes da máquina assíncrona.

2.1.3.8 Alimentação do sistema de controlo

A alimentação DC é fornecida pela central com 48.60V. Para se garantir uma paragem segura do sistema em caso de falha de alimentação exterior, é utilizado um conjunto de condensadores que mantêm a tensão de alimentação durante aproximadamente 10s.

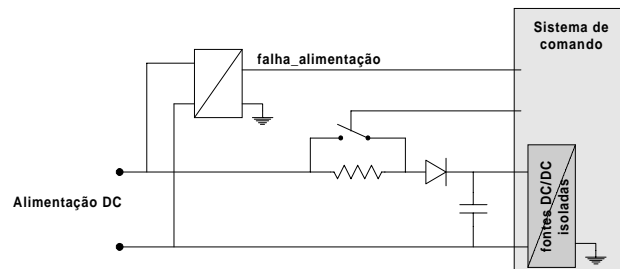


Figura 6.33 - Circuito de alimentação DC.

O sinal *falha_alimentação* é gerado por um circuito autoalimentado e isolado através de um optoacoplador, sendo ligado à interrupção não mascarável (NMI) do C167. Desta forma

sempre que a tensão DC fornecida pela central desce abaixo de 47V (aproximadamente) surge uma interrupção NMI no C167 de modo a que este tome as acções requeridas.

A resistência serve para limitar a corrente no início da carga dos condensadores, sendo depois curto-circuitada pelo relé controlado pelo C167.

Todo o controlador, à excepção dos circuitos de protecção contra sobretensões do inversor são alimentados por um conversor DC/DC que fornece as tensões: +5V, -15V e +15V. A sua monitorização é efectuada através dos +5V que alimentam o MM167 por um integrado dedicado (MAX690) incluído no MM167.

Para salvaguardar os dados na memória RAM do C167, é utilizada uma pilha que é seleccionada e posta em serviço pelo MAX690.

Desta forma o MAX690 verifica não só a alimentação do MM167 mas também a pilha da seguinte forma: sempre que a tensão de alimentação estiver mais baixa que 4.5V (aproximadamente) é efectuada um reset ao C167; sempre que a tensão da pilha estiver abaixo de 2.5V (aproximadamente) é activado um sinal ($\overline{\text{PFO}}$) que é ligado à porta P8.7 do C167 permitindo desta forma a detecção de “pilha fraca”. A alimentação da memória RAM externa do C167 é seleccionada entre a pilha e a alimentação pelo MAX690, através da escolha do dispositivo que fornece maior tensão.

Para evitar problemas em caso de reset do C167 pelo MAX690, o IGBT de protecção é actuado e os contactores (estátor e rotor) são abertos sempre que o C167 está em reset. (Note-se que em reset, as portas do C167 são “altas impedâncias”)

Por razões de segurança, o driver do IGBT e os circuitos de comando da protecção contra sobretensões do inversor são alimentados por uma fonte independente (+15Vprot e +5Vprot). A sua tensão é monitorizada por um circuito que coloca o sinal $\overline{\text{falha_alim_prot}}$ a zero no caso de ambas as alimentações (+15Vprot e +5Vprot) falharem. Este sinal é ligado ao pino P2.10 do C167.

Devido ao elevado consumo, os seis drivers dos IGBTs do inversor são alimentados por uma fonte independente.

2.1.3.9 Inversor e circuito de protecção

Cada driver dos IGBTs do inversor possuem dois sinais de entrada (*on/off* e *reset*) e um sinal de saída (*erro*). Os sinais de erro dos drivers são em colector aberto e activos em baixo, pelo que são juntos de modo a formarem um *wired-or* (*erro_inv*).

O reset do inversor é efectuada pelos sinais $\overline{\text{RST_INV}}$ e $\overline{\text{RST_OUT}}$. O sinal $\overline{\text{RST_OUT}}$ é activo sempre que o C167 entra no estado de reset até que finaliza a sua inicialização. Desta forma garante-se que o inversor está desligado enquanto o C167 se inicializa.

Quer o sinal de *reset* quer os sinais de *on/off* passam por um array de transístores para adaptação de tensão e ganho em corrente.

A fonte +15V_igbt é independente e dedicada só à alimentação dos drivers dos IGBTs do inversor.

Para proteger o inversor de sobretensões provocadas pela interrupção do circuito DC existe um circuito que detecta as respectivas sobretensões. Este circuito envia o sinal de *sobretensão* através de fibra óptica.

O driver do IGBT de protecção é igual aos utilizados no inversor. Deste modo também possui 3 sinais: *on/off*, *reset* e *erro*. A utilidade do sinal de *erro* é duvidosa porque para este sinal surgir, o driver retirou o IGBT de condução por excesso de tensão Vce, e isto só acontece se o IGBT ter sido posto em funcionamento por problemas de sobretensão. Nesta situação não há muito que se possa fazer, a não ser substituir as peças que foram destruídas !

Em caso de ser detectada uma sobretensão, é necessário actuar o IGBT de protecção o mais rapidamente possível. Para evitar o tempo de resposta do μ C, o IGBT é actuado directamente por hardware sempre que o sinal *sobretensão* é activado.

Uma das origens para a sobretensão é a saída de condução dos IGBTs do inversor devido aos drivers. Desta forma, o sinal *erro_inv* do inversor também actua directamente o IGBT de protecção evitando (ou pelo menos diminuindo) a sobretensão.

Naturalmente os sinais *erro_inv* e *sobretensão* também são ligados ao C167.

O contactor do estátor não é aberto directamente por hardware em caso de sobretensão porque este dispositivo demora cerca de 40ms a abrir. Desta forma não adianta poupar alguns μ s com a actuação directa por hardware.

Os flip-flop servem como memórias para guardar os sinais *sobretensão* e *erro_inv* depois de estes surgirem.

A protecção é retirada de serviço através do “reset dos flip-flops”. O sinal $\overline{RST_PROT}$ alem de actuar no flip-flop também executa o reset do driver do IGBT de protecção.

3 Software

A programação do C167 é feita com o package de desenvolvimento da Keil Software, para a família de microcontroladores 80C16x.

Este package contém:

- Compilador de C
- Assembler
- Debugger com simulação do μ C no PC e “on-chip” debugger

O compilador de C respeita as normas do ANCI-C e tem algumas extensões para aceder e controlar os periféricos do C167: registos de controlo, PECC, AD, etc.

Apesar de neste trabalho se apresentarem algumas características do Compilador, à medida que vão sendo necessárias para uma explicação do software escrito, a leitura do manual do compilador e do assembler do C167 é aconselhável para uma total compreensão.

A Phyttec fornece juntamente com o MM167 o software necessário para efectuar o Download do programa e grava-lo na Flash-EPROM.

Fornece também os drivers para:

- gravar/apagar a Flash-EPROM,
- controlar a UART,
- controlar o RTC (Real Time Clock).

3.1 Sistema Operativo

Com a quantidade de tarefas que o μC tem de realizar é necessário recorrer à execução paralela de tarefas.

O bom senso e as boas técnicas de programação aconselham sempre o uso de um Sistema Operativo (SO) para gerir os recursos da máquina e tornar mais simples a tarefa de gestão do multi-processamento e comunicações interprocessos.

A opção de se controlar os IGBTs do inversor directamente a partir do C167 leva a que o uso de um SO fique condicionado. Isto acontece porque um atraso de μs no atendimento da rotina que executa a mudança de estado¹ no inversor, apesar de não ter efeitos nefastos sobre o inversor, modifica o padrão do PWM, introduzindo um conteúdo harmónico considerável nas correntes injectadas na rede.

É claro que isto não passa de um problema comum aos Sistemas de Tempo Real: o atendimento atempado de requisições externas e internas.

A estrutura do sistema de interrupções do C167, instruções dedicadas à implementação de semáforos, capacidade de transferência periféricos/memória sem a intervenção do CPU tornam o C167 numa plataforma de elevado desempenho em sistemas embebidos, facilitando a construção de software multi-tarefa mesmo sem o recurso a SO (pelo menos um SO convencional implementado totalmente em software e a controlar/supervisionar todas as tarefas).

Uma hipótese consistiria em as tarefas que permitem alguma flexibilidade no tempo de execução serem controladas por um SO e as restantes serem controladas por hardware pelo sistema de interrupções do C167. Mesmo assim levanta-se o problema da inibição das interrupções pelo SO de forma a ele executar a sua gestão de tarefas.

De facto a inibição das interrupções por parte do SO pode mudar consideravelmente o padrão dos impulsos de PWM. Desta forma, tomou-se a decisão de não se utilizar um SO. No entanto o seu uso não está posto de parte e depende de testes que têm de ser realizados através da medição do conteúdo harmónico nas soluções com SO e sem SO. É óbvio que o software tem de ser escrito tendo em mente a sua adaptação a uma plataforma com SO.

A organização da execução das tarefas fica assim assente na estrutura de interrupções do C167, obtendo-se uma “espécie de SO” baseado em hardware.

¹ A mudança de estado consiste em mudar os IGBTs que estão a conduzir.

3.2 Estrutura do software

O software é baseado na estrutura do sistema de interrupções do C167. Existem basicamente 5 níveis de prioridade de execução:

- Comando do inversor: actuação nos IGBTs do inversor e sincronismo com a rede.
- Procedimentos de emergência (abertura de contactores, actuação das protecções, etc) no seguimento do atendimento de alarmes ligados a interrupções do C167: sobretensão, IGBT fora de serviço, etc.
- Procedimentos de controlo que consistem na determinação do índice de modulação e da fase com que o inversor deve funcionar. Isto inclui toda a acção do controlo propriamente dito: estado do SRED (arranque, normal, paragem), determinação da potência a retirar da máquina, estado dos disjuntores e contactores, polling de alarmes não críticos temporalmente (termóstatos, vibrações), etc.
- Interface com o PC e com o Autómato.
- Interface com o operador local.

As rotinas que controlam o inversor possuem as prioridades mais altas do sistema e portanto nunca são interrompidas a não ser pelo conversor A/D que guarda na memória os resultados das conversões via PEC que não utiliza o CPU e apenas “gasta” um ciclo máquina.

Os métodos e rotinas de comando do inversor estão descritos no capítulo 4.

Os procedimentos de controlo são executados através de um pré-escalonador. O temporizador T0 dá origem a uma interrupção a cada 1.25ms. O tempo de 1.25ms prende-se com o número de aquisições das medidas analógicas que é feito por cada período da rede. São efectuadas 16 amostras por cada período pelo que o período de amostragem é $20\text{ms}/16=1.25\text{ms}$.

Esta interrupção tem uma prioridade logo a seguir à prioridade do comando do inversor.

O pré-escalonador tem a função de iniciar uma sequência de aquisição de medidas a cada 1.25ms e de dar início à execução do escalonador a cada 20ms.

O escalonador é responsável por calcular os valores necessários ao controlo a partir das amostras recolhidas pelo A/D, gerir os algoritmos de controlo, efectuar o polling dos alarmes e dos botões da consola local, etc. A prioridade do escalonador situa-se abaixo da do pré-escalonador.

Sempre que as UART externa e interna recebem um byte vindo respectivamente do PC e Autómato produzem interrupções com prioridade entre o escalonador e o pré-escalonador. Estas interrupções armazenam o byte num buffer e produzem uma interrupção que vai efectuar o tratamento da mensagem recebida. Esta interrupção tem uma prioridade abaixo da do escalonador.

Por fim a interface com o operador local é efectuada no nível mais baixo de prioridades.

Como é óbvio uma rotina não pode demorar constantemente todo o tempo disponível a ser executada, sob pena das rotinas com prioridade inferior nunca serem executadas. Se por exemplo o pré-escalador demorar constantemente 1.25ms a ser executado, o escalador nunca consegue ser executado.

Desta forma o software é controlado através de uma espécie de Sistema Operativo baseado no sistema de interrupções do C167.

3.2.1 Pré-escalador e aquisição de medidas analógicas

As 11 medidas analógicas:

10 - Corrente na fase R da rede	IR
9 - Tensão na fase R da rede	UR
8 - Corrente na fase S da rede	IS
7 - Tensão na fase S da rede	US
6 - Corrente na fase T da rede	IT
5 - Tensão na fase T da rede	UT
4 - Corrente no barramento DC	IDC
3 - Tensão gerada pelo rectificador	UDC
2 - Corrente na fase T do rotor	IT_ROT
1 - Corrente na fase S do rotor	IS_ROT
0 - Corrente na fase R do rotor	IR_ROT

são adquiridas com 16 amostras por período da rede. Desta forma são efectuadas 11 conversões a cada $20\text{ms}/16=1.25\text{ms}$ (800HZ).

Sempre que o A/D recebe ordem de início de conversão (bit *ADST*), executa consecutivamente a conversão das 11 medidas (pela ordem indicada). Sempre que acaba de converter uma medida inicia de imediato a seguinte conversão e provoca uma interrupção que dá origem a uma transferência via PEC do valor convertido para um buffer na memória. O A/D funciona à máxima velocidade (cada conversão demora $8.15\mu\text{s}$). As 11 medidas demoram $8.15\mu\text{s} * 11 = 89.65\mu\text{s}$ e a diferença de tempo entre a aquisição da 1ª medida (*I_{R_GERAL}*) e da última (*I_{R_ROT}*) é de $8.15\mu\text{s} * (10) = 81.5\mu\text{s}$.

Os resultados das conversões são guardados na memória XRAM do C167. A memória XRAM (eXtended RAM) é um acréscimo de 2K à memória interna convencional que os microcontroladores derivados C167CR possuem em relação aos C167 base. Esta memória pode ser activada/desactivada por software durante a inicialização do μC . Como os acessos à memória interna são mais rápidos e os resultados de conversão devem ser lidos o mais rapidamente possível, optou-se por fazer a transferência via PEC para esta memória.

O 1º canal da PEC (interrupção mais elevada) é utilizado para realizar $16 * 11$ transferências de 16 bits do registo que guarda o resultado da conversão (*ADDAT*) para a zona de memória reservada aos resultados.

Desta forma o 1º canal da PEC é inicializado da seguinte forma:

- Tipo de transferência: *word* (16 bits);
- Apontador de origem: fixo em *ADDAT*;

- Apontador de destino: endereço de *IR_1* (00.E000h) e incrementado a cada transferência;
- Número de transferências: $11 \cdot 16 = 176$.

No final da transferência das 16 amostras (176 valores) a PEC provoca uma interrupção convencional para se reinicializar o apontador de destino e o número de transferências a efectuar.

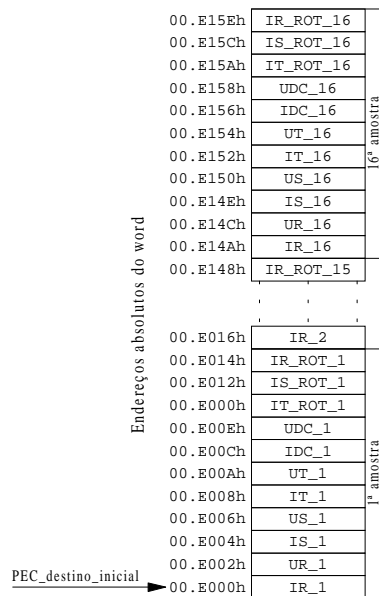


Figura 6.34 - Mapa de memória para os resultados das conversões A/D.

Quando o AD acaba uma conversão e o valor da ultima conversão ainda não foi guardado na memória, surge uma interrupção (*problemas_conversao_ad*) que sinaliza o facto através da flag *G_flag_AD* e incrementa a variável *G_erro_AD*.

Para se efectuarem as amostras em intervalos de 1.25ms, utiliza-se um pré-escalador. O nome pré-escalador surge porque esta rotina, além de dar ordem ao A/D para converter as amostras, também dá início à execução de um escalador de tarefas a cada 20ms.

O pré-escalador é executado a cada 1.25ms através da interrupção gerada pelo temporizador T0 (interrupção T0INT). Esta rotina só é interrompida pelas rotinas de controlo do inversor.

No final da conversão das 16 amostras o pré-escalador verifica se houve ou não sobreposição de dados através da flag *G_erro_AD*, e se fôr esse o caso executa a rotina (*busca_valores_AD_asm*) que retira os valores do buffer das conversões e coloca os resultados das amostras em variáveis globais.

A rotina *busca_valores_AD_asm* executa ainda algum tratamento da informação contida no buffer:

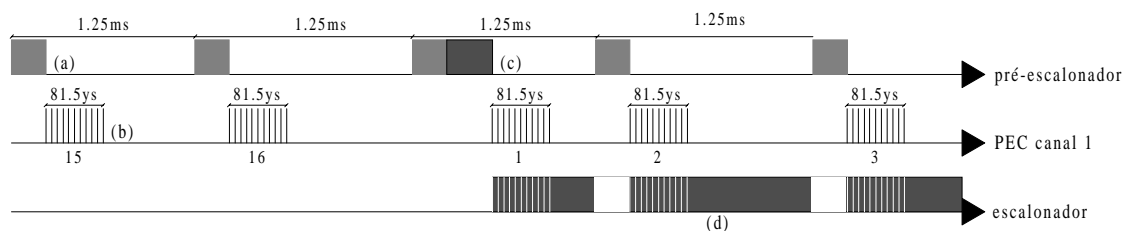
- Limpa os 6 bits (bit15..bit10) que contêm informações sobre o canal convertido. Apenas os 10 bits menos significativos contêm o resultado da conversão (o AD é de 10 bits; níveis:0..1024).

- Subtrai o valor 512 ao valor convertido no caso das medidas alternadas para ajustar o offset da medida. (uma conversão de uma grandeza alternada de valor zero produz um resultado de 512)

Esta rotina foi escrita em assembler de modo a otimizar o tempo de execução.

As rotinas que necessitam dos valores das amostras acedem às variáveis globais respectivas.

Deve ser notado que o acesso exclusivo às variáveis globais das amostras e ao buffer original do resultado das conversões está garantido pela sequência da execução conforme é verificado na Figura 6.35.



- (a)- Ordem de início de aquisição das 11 medidas.
 (b)- Escrita no buffer via PEC dos resultados das conversões (no final de cada conversão).
 (c)- Renovação das variáveis globais das amostras (rotina *busca_valores_AD_asm*).
 Ordem de início de execução do escalonador através de requisição de interrupção.
 (d)- Escalonador. É neste nível que as variáveis globais com as amostras são utilizadas no cálculo de várias medidas (média, rms, fft. etc).

Figura 6.35 - Sequência de execução do pré-escalador e escalonador.

O escalonador é posto em execução através da invocação de uma interrupção. Isto é conseguido através da activação do bit de requisição da interrupção sobre a qual o escalonador é activado (IR - PWM)

O fluxograma do pré-escalador é indicada na Figura 6.36.

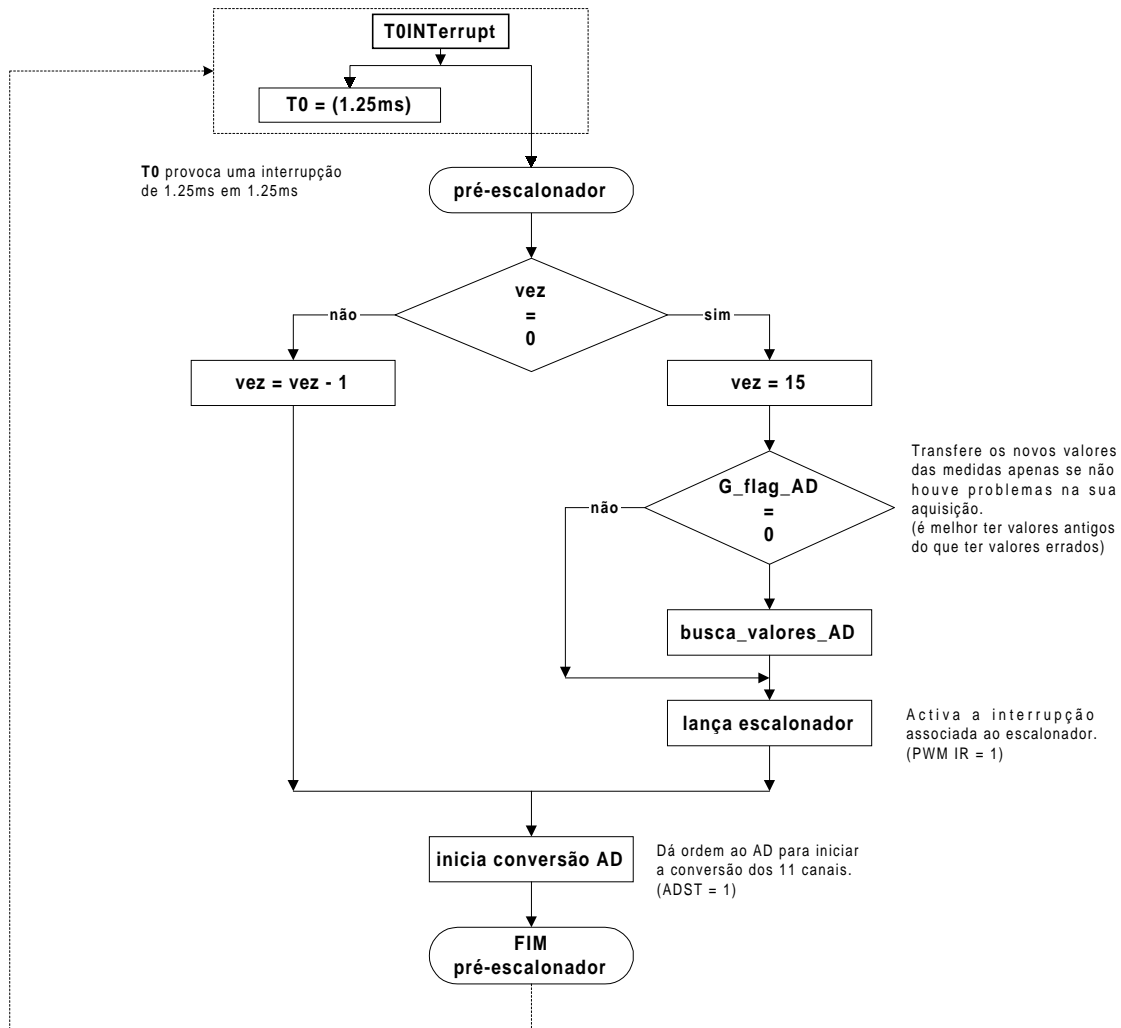


Figura 6.36 - Fluxograma do pré-escalador.

O código do pré-escalador, da rotina *busca_valores_AD_asm*, das interrupções provocadas pela PEC e pelo AD, e das inicializações é exposto no Anexo C. Todo o código está comentado pelo que a sua compreensão não deve ser difícil.

3.2.2 Escalonador

O escalonador é responsável pela gestão de todas as tarefas associadas ao controlo do SRED propriamente dito:

- Cálculo da velocidade e das medidas (RMS, Média, componente fundamental, etc) em função das amostras;
- Determinação do índice de modulação e da fase do inversor;
- Determinação da potência a retirar do sistema;
- *Polling* dos alarmes lentos (termostatos, vibrações, etc) e do estado dos contactores e disjuntores.

É ainda responsável por fazer o *polling* dos botões do painel local.

Como o módulo interno do C167 que gera o PWM não é utilizado para controlar o inversor, a interrupção de PWM é usada para correr a rotina do escalonador.

A interrupção é activada pelo pré-escalonador a cada 20ms.

3.2.3 Consola local

Com excepção do *polling* dos botões, as tarefas que realizam a interface com o operador local são executadas no nível mais baixo de prioridade.

3.2.3.1 botões

A rotina que executa o *polling* dos botões é chamada a cada 20ms através do escalonador. Esta rotina executa o *debounce* dos botões e escreve as transições (ascendentes e descendentes) associadas a cada botão num buffer.

O buffer com capacidade para 10 transições de botões é do tipo circular e possui dois apontadores e uma flag. Os apontadores *G_ptr_key_in* e *G_ptr_key_out* indicam respectivamente a posição de inserção de uma nova transição e a posição de leitura. A flag *G_flag_key* indica se o buffer está vazio ou não.

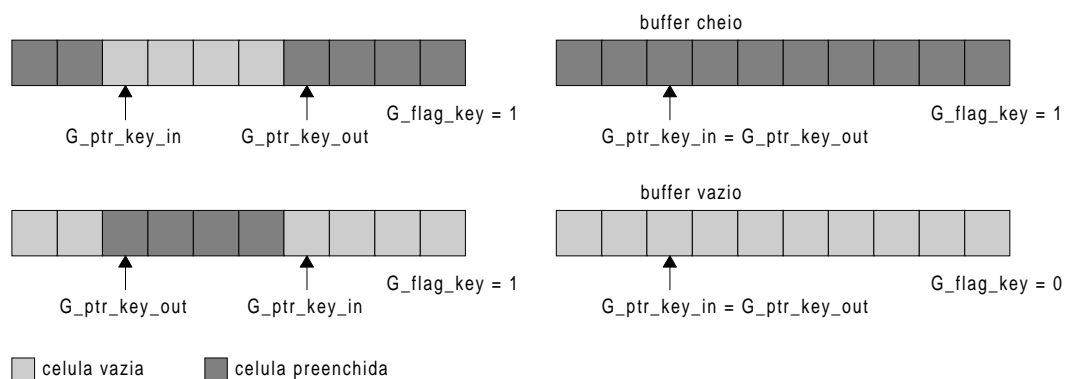


Figura 6.37 - Esquema do buffer utilizado para os botões da consola local.

Cada botão tem um código (byte) associado às transições ascendentes e descendentes:

- ENTER_HI , ENTER_LOW
- UP_HI , UP_LOW
- DOWN_HI , DOWN_LOW
- LEFT_HI , LEFT_LOW
- STOP_ALARM_HI , STOP_ALARM_LOW
- STOP_HI , STOP_LOW

Se o buffer não estiver cheio, a rotina *capta_botoes* introduz no buffer a transição do respectivo botão e actualiza o apontador *G_ptr_key_in* e a flag *G_flag_key*. O fluxograma da rotina é mostrado na Figura 6.38.

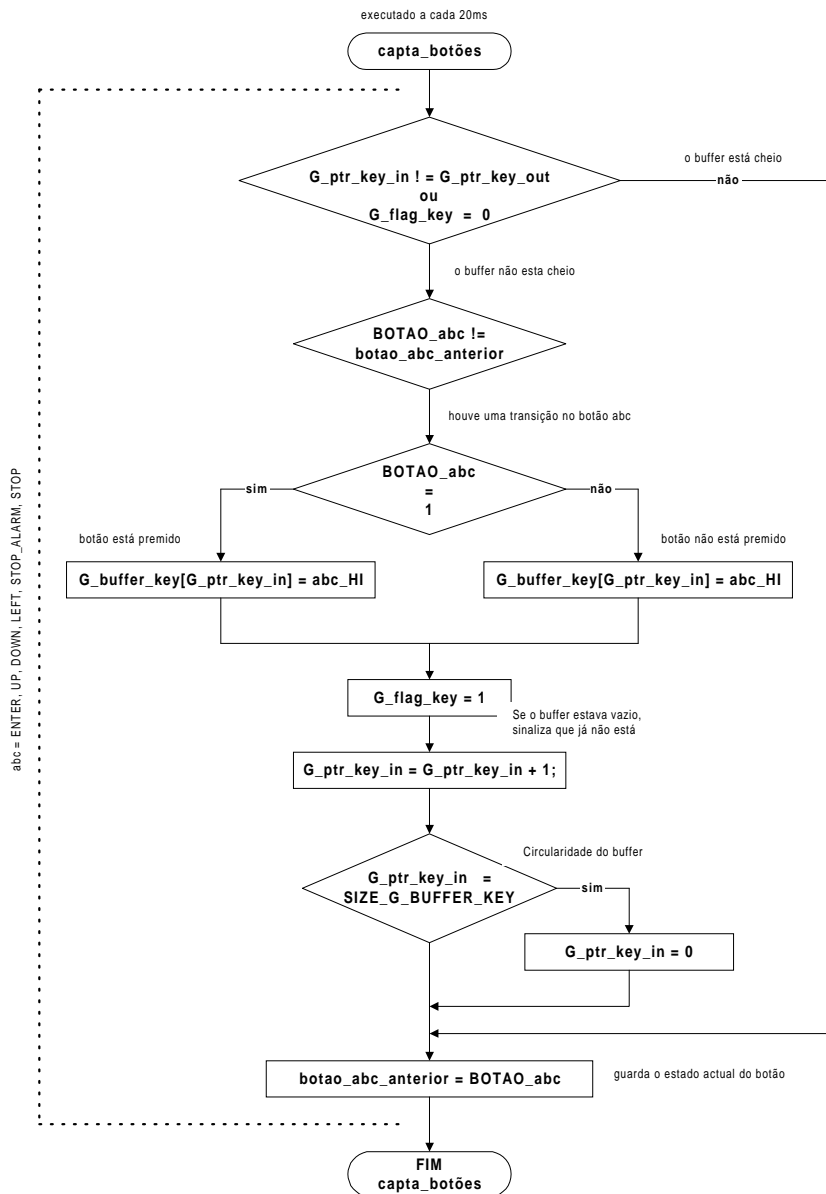


Figura 6.38 - Fluxograma para a captura dos botões e escrita no buffer.

Para a leitura do buffer existe uma função (*GetPanelKey*) que retorna a transição do botão respectivo. Se o buffer estiver vazio retorna o valor zero. O fluxograma da função está descrito na Figura 6.39. As rotinas que necessitam de utilizar os botões acedem apenas à rotina *GetPanelKey*.

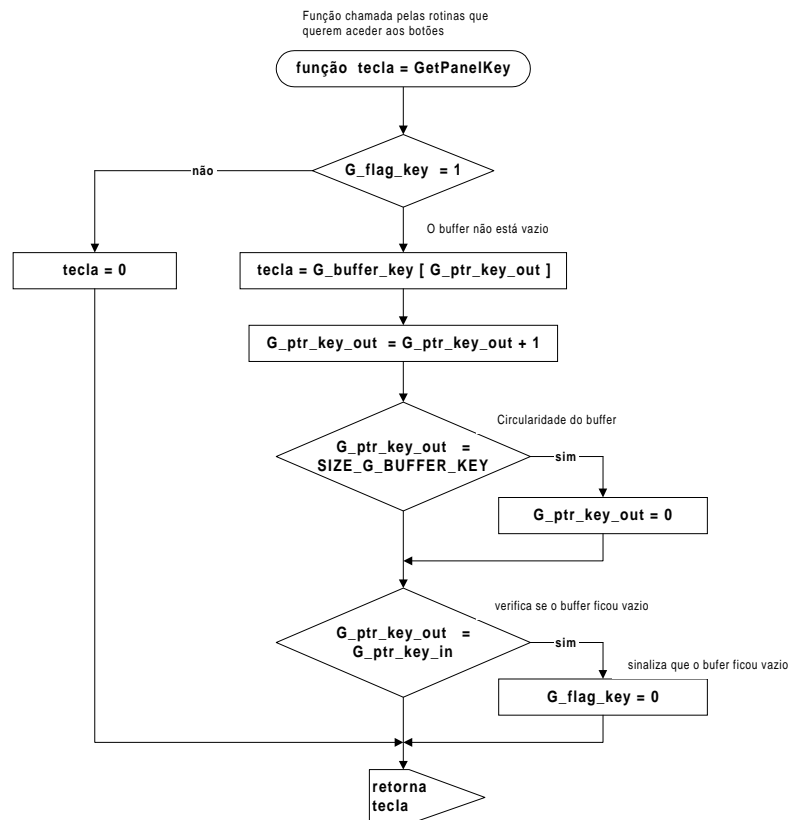


Figura 6.39 - Fluxograma para as rotinas acederem aos botões que foram premidos pelo operador.

3.2.3.2 Sinalização

Os *leds* do painel não requerem rotinas especiais de interface. As rotinas que querem aceder ao *leds* apenas escrevem um “1” ou “0” no bit de controlo da porta onde o respectivo *led* está ligado.

Naturalmente as portas têm de ser configuradas como saídas.

3.2.3.3 Visualizador

O visualizador (*display*) é ligado por um barramento de 4 linhas de dados e 3 de controlo. Para esconder a complexidade de controlo do *display*, foram construídos *drives* que simplificam o uso do *display* por parte das rotinas que dele necessitam.

Os *drives* incluem funções para:

- inicializar o *display*;
- controlar e localizar o cursor;
- escrever um caracter ou um conjunto de caracteres;
- definir caracteres especiais.

4 Conclusões

Neste capítulo foram apresentadas as características funcionais da plataforma de comando do SRED.

Foram analisadas as características de funcionamento no sentido de verificar que o seu desempenho é suficiente como suporte ao desenvolvimento do sistema de comando para o SRED.

O sistema de comando interaccua com o operador local, com o computador pessoal (PC) e com o autómato responsável pela gestão da central.

Como plataforma de *hardware* é utilizada uma placa comercial de processamento baseada no μC SAB80C167 da Siemens (MM167). Para adaptar o MM167 ao SRED foi utilizada uma placa que realiza o condicionamento das medidas necessárias e realiza várias funções de protecção dos dispositivos do SRED.

A aplicação de comando é construída tendo como base o sistema de interrupções do μC , com um escalonamento estático multitarefa.