Persistência da Visão em 2D

FEUP - Sistemas Embarcados

Patrício Lima
José Borges
Hélder Lobato

Docentes:
Luís Almeida
Mário Sousa

Indice

1. Resumo

Este relatório vai descrever o projecto POV 2D Circular realizado no âmbito da unidade Curricular de Sistemas Embarcados 2010/11-2º Semestre. Neste documento vão ser apresentados as especificações do projecto, as opções adoptadas para implementação do suporte físico (hardware) e software de forma corresponder às especificações feitas. Por último, são apresentados os resultados conseguidos.

2. Introdução

O objectivo deste trabalho é explorar o conceito da persistência da visão(POV – Persistence Of Vision) em duas dimensões. A persistência da retina é um fenómeno no qual o olho humano retém, durante um certo período (aproximadamente 1/24 segundos) uma imagem antes desta desaparecer da sua "memória", ou seja, é a fracção de segundo em que a imagem permanece na retina.
Devido a este fenómeno se o ritmo de chegada entre imagens for superior a 24 por segundo, as imagens associam-se na retina provocando a ilusão de movimento contínuo.

Tendo como base este conceito, neste trabalho será construindo um POV DISPLAY de duas dimensões. È um dispositivo eletrónico formado por uma placa de 32 LED's em linha recta em movimento circular, a cada ângulo deste movimento será "enviado" um conjunto de cores para os leds e quando a frequência de rotação fôr superior á frequência de POV, provocará no espectador a ilusão de uma imagem circular 65x65 pixeis.

Para comprovar o funcionamento do dispositivo foi criado três mini programas que irão projectar no display, um relógio digital, um relógio analógio e imagens circulares.

3. Modo de Funcionamento

O modo de funcionamento deste dispositivo é divido em dois modos:

- CONFIGURAÇÃO – Neste modo, é possivel configurar os parâmetros dos mini-programas como a cor dos ponteiros, definir a hora, etc... Esta configuração é conseguida pelo envio de comandos pela porta série. Os comandos adoptados são os seguintes:

Comando	Argumento	Função
SCLK	HHMMSS	Configura a hora do RTC, colocando neste a hora passada por argumento.
GCLK		Retorna a hora que se encontra no RTC
MODE	PICT, DCLK, ACLK	Defini o programa que vai ser visualizado no dispositivo (IMAGEM, RELÓGIO DIGITAL, RELÓGIO ANALÓGICO).
NCLR	RGB	Define a cor dos números que aparecem no dispositivo quando este encontra-se no modo ACLK(Relógio Analógico).
PCLR	RGB	Define a cor que os ponteiro do relógio que aparecem no dispositivo quando este encontra-se no modo ACLK(Relógio Analógico).
HELP		Retorna as mensagens de ajuda para o utilizador, isto é , a lista de comando permitida e a descrição destes.
SPIC	Número da Imagem	Definie qual a imgem que o dispositivo irá ler da ROM e projectar nos LED's

Tabela 2.1

- Modo de Visualização – Neste modo será projectado um dos 3 mini programas, caso se pretenda alterar o programa a projectar terá que se utilizar o comando "MODE". Que selecionará o modo de visualização pretendido. Os modos são

    1. Relógio Analógico –Projectará um relógio analógico, que ocupa a totalidade do display.
    2. Relógio Digital – Projectará um relógio digital numa "janela" na parte superior do display.
    3. IMAGEM – Este projectará no display um conjunto de imagens gravadas na ROM

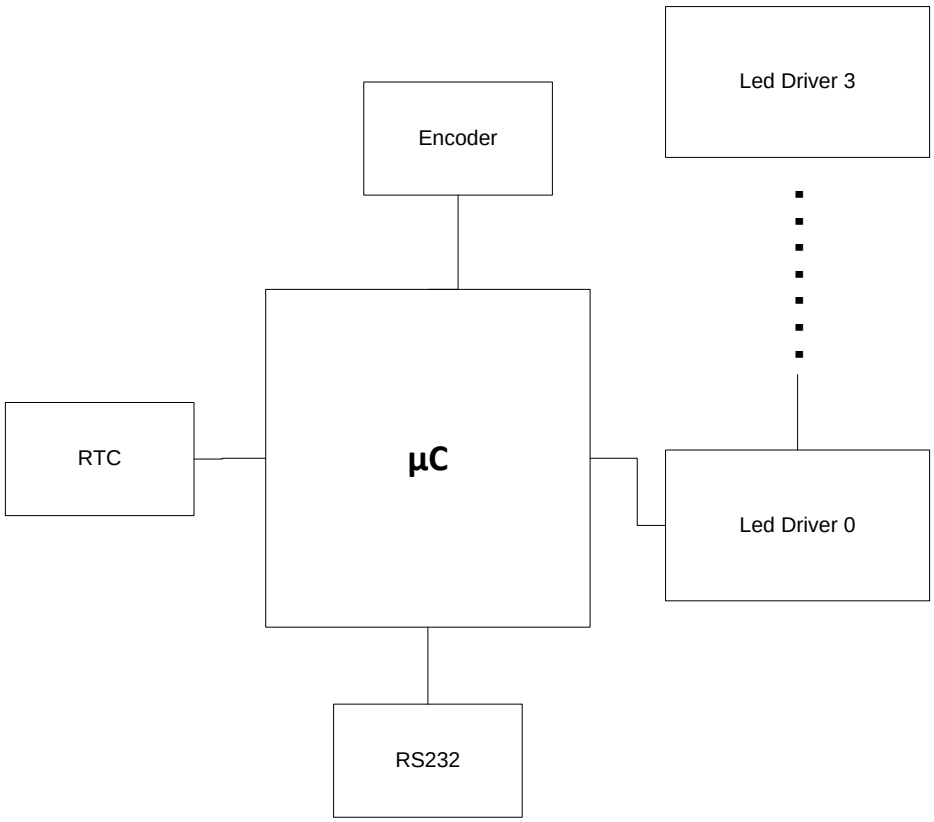
4. Implementação

4.1 PCB

O circuito completo do POV é composto pelos seguintes componentes:

- 1 Encoder – Que lê e envia para o microcontrolador a posição actual dos leds
- 1 RTC – Real Time Clock onde são lidas as horas, que são mostradas no relógio digital e analógico.
- 1 RS232 driver – que é responsável para a comunicação série.
- 4 Led drivers – Que alimentam os leds de acordo com os comandos recebidos do microcontrolador através do protocolo SPI.
- 32 Leds RGB – Que apresenta os padrões que formam as imagens.
- 1 LDO - Que regula a tensão de alimentação do circuito.

Na figura abaixo é mostrada um diagrama de como estão interligados os componentes.



4.2 Programação do Microcontrolador

O programa para o mirocontrolador, desenvolvido em C pode ser divido em 4 partes:

- ⇨ Programa Principal
- ⇨ Interrupção Externa (Encoder)
- ⇨ Interrupção por Overflow em Timer 0
- ⇨ Interrupção pela USART (Universal Synchronous Asynchronous Receiver Transmitter).

4.2.1 Programa Principal

Como podemos visualizar no fluxograma da figura 1, o programa principal tem como função a projecção do display circular 65x65 pixeis(leds) . Para tal em cada posição (ângulo) da placa será obtido de um dos três modos (Digital , Analógico ou Imagem) um conjunto de 32 cores RGB (referente ao pixel a enviar) e em seguida essa informação será enviada para os LED DRIVERS.

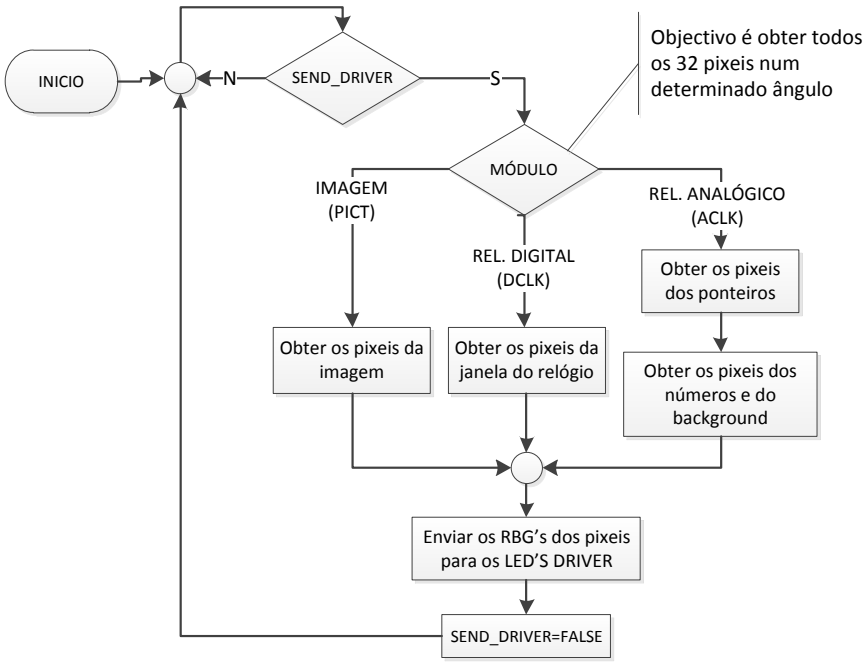


Figura 1

Ou seja, sabendo que na memória do microcontrolador para cada modo de visualização escolhido encontra-se a imagem (um array 65x65 de RGB's) a projectar , o programa principal tendo em conta a orientação da placa e a posição do LED pretendido irá calcular a coordenada cartesiana de modo a obter a respectiva cor rgb que se encontra no array de imagens do modo selecionado. Após fazer isso para os 32 leds numa determinada posição

essa informação será enviada para o leds drivers. Na figura (2) encontra-se um exemplo de como imagem encontra-se organizada na memória.

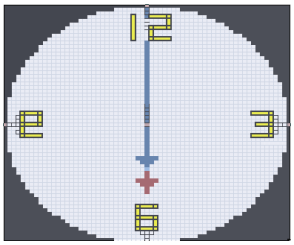


Figura 2

Na conversação de polar para cartesiano foi adoptado o uso de uma tabela de senos e coseno, pois durante a construção do programa notou-se que o uso das funções matemáticas da linguam C, tornavam o programa excessivamente lento devido ao tipo de dados utilizado (float ou double) e ás operações intermédias realizadas por essas funções para o cálculo do resultado.

4.2.2 Interrupção Externa

A Interrupção externa neste caso ocorre devido ao enconder, pois com as interrupções provocadas por este dispositivo e utilizando um contador para contabilizar as interrupções é possivel saber a orientação da placa. Pois as cores a enviar para os leds drivers dependem da orientação.. Na figura 3(a) , encontra-se o fluxograma desta interrupção.

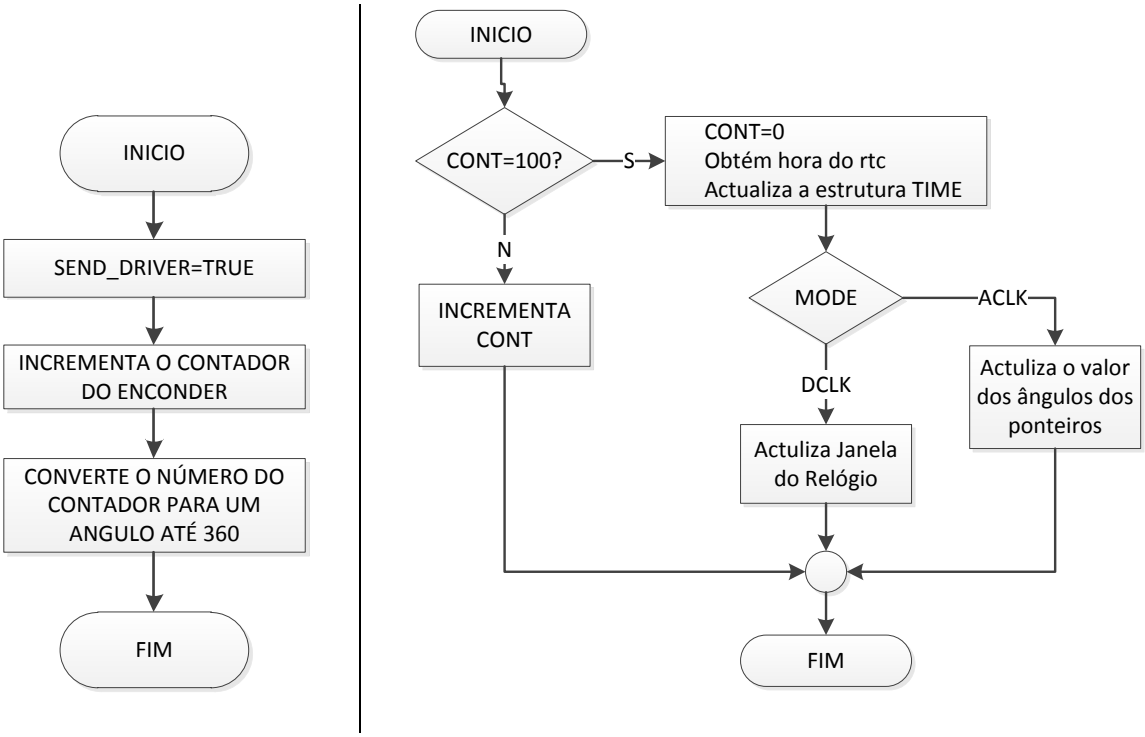
4.2.3 Interrupção por Overflow em Timer 0

Exceptuando o modo IMAGEM no qual a imagem é sempre fixa,nos restantes modos cada imagem a ser projectada é um retracto 65x65 num determinado instante, como as horas vão variando ao longo do tempo. Logo é necessário actualizar as imagens do modo digital e analógico . A estratégia adoptada para o refrescamento dos dados contidos nos arrays, foi configurar esta interrupção para a cada 800 milisegundo obter a hora no rtc e actualizar as imagens correspondentens a cada modo de visualização. O cálculo desta tempo é dado pela a seguine equação:

$$T = x.N.Prescaler.\frac{1}{fosc}$$

- N – É o número de incrementos no timer 187
- Prescaler – é um divisor de frequência o seu valor é de 1024
- Fosc – é a frequência de oscilação do temporizador neste caso é de 24MHz
- X – Numero de vezes que a interrupção do timer tem de ocorrer para ter o tempo certo, neste caso é 100

Na figura 3(b) poderemos visualizar fluxograma da rotina de do timer.



(a) Interrupção Externa (b) Interrupção do TIMER 0

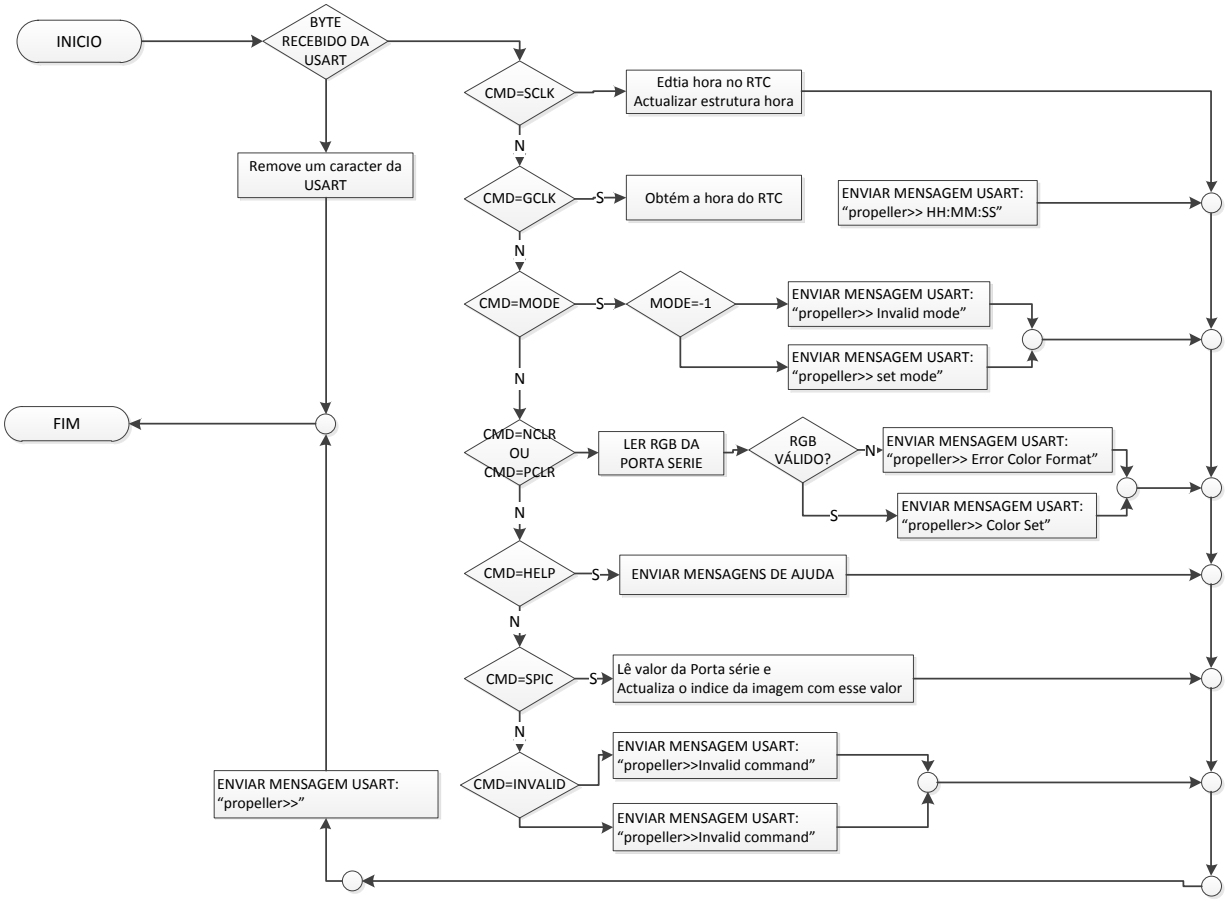
Figura 3

4.2.4 Interrupção pela USART

A interrupção de recepção da USART é utilizada para para receber serialmente os comandos provenientes do computador. A frequência de tranmissão dos dados apartir do computador é conhecida por frequência de BAUD RATE, o cálculo desta frequência é dado pela a seguine equação:

$$f_{BAUD\ RATE} = \frac{fosc}{16(factor + 1)}$$

No nosso caso , foi utilizado um factor=77 e uma fosc =24MHz o que resultou numa frequência de BAUD RATE de 19200bits/s.
Na figura 4, podemos visualizar a o fluxograma da rotina desta interrupção



4.2.5 Escalonamento de tarefas.

Dado a periodicidade dos eventos, as tarefas também foram escalonadas de forma periódica. O escalamento destas tarefas foi feito usando prioridades fixas, de acordo com as suas frequências conforme é mostrada na tabela em baixo:

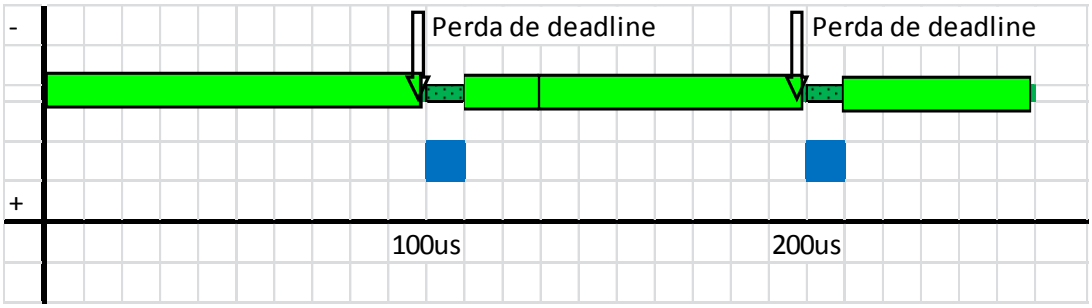
Tarefa	Funcionalidade	Periodo	Dur	Prio.
T1: Interrupção externa	Executa em cada interrupção externa feita pelo encoder. Lê a memória e envia dados para os drivers dos leds.	$200\mu s$	~10µs	1
T2: Interrupção do Timer	Executada em cada interrupção do Timer. Actualiza as horas através da leitura do RTC.	800ms	60 µs	2
T2: Envio das cores para os leds	Sempre a seguir a uma interrupção externa	200µs	~120µs	3

Sendo que o motor roda até o máximo de 20 voltas por segundo(periodo igual a 50ms) então se o motor rodasse a 20 voltas por segundo o encoder tendo 500 tiras então cada interrupção daria de 100 em 100 microssegundos com duração em media 10µs, ou seja para que não houvesse perda de deadline a tarefa teria de ser executada em tempo menor ou igual 100µs. Uma outar tarefa é a leitura dos dados

para serem enviados para os leds e o envio das mesmas. Esta tem de ser executada em cerca de 90µs. A frequência do clock é de 24MHz o que nos da no máximo uma frequência do SPI de 12MHz. Para enviar os dados para os leds tem de enviar no total 1152 bits, sendo 288 para cada driver, com a frequência de SPI a 12MHz este demora 96 µs que é 106.7% do do período das interrupção, juntando com o tempo de processamento para os cálculos das coordenadas ultrapassa o tempo que tem disponível, o que faz com que algumas tarefas perdem os seus dealines. A esta velocidade ficaria um pouco difícil de mostrar as imagens. A solução foi baixar a velocidade do motor para cerca de metade. Sendo assim mesmo que algumas tarefas que iam perder os seus deadlines eram descartadas, pois o encoder tendo um enorme valor de tiras não causaria danos na qualidade da imagem.Uma outra tarefa é a interrupção do timer, este é uma tarefa que demora em media 60µs e tem um período de 800ms, sendo uma tarefa que tem um período muito elevado não causa muitos problemas na apresentação da imagem.

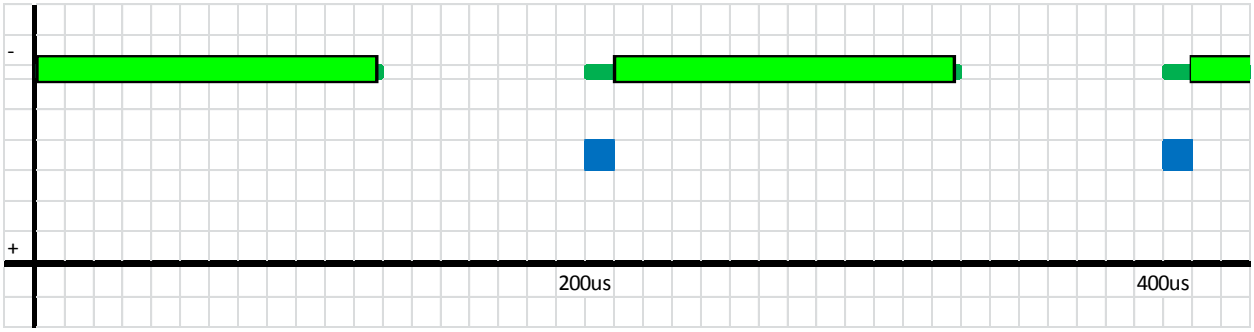
- **Escalonamento com 20 voltas por segundo.**

Com o escalonamento feito para 20 voltas por segundo a tarefa de menor prioridade perde o seu deadline(C=120µs T=100µ D=T) todas as vezes o que causa uma distorção na apresentação da imagem. Sendo a interrupção externa de maior prioridade ela interrompe sempre o envio dos dados para os leds.



- **Escalonamento para 10 voltas por segundo**

Já com 10 voltas por segundo é possível fazer o escalonamento. A tarefa da interrupção do timer não foi considerada porque a interferência que ela causa é desprezável.



Envio dos dados para os leds

Interrupção do encoder

5. Conclusão

Tarefas realizadas. Este projecto foi realizado seguindo diversas etapas. As tarefas realizadas foram:

- Definição das funcionalidades do Sistema
- Especificação do Software
- Especificação do suporte do sistema físico
- Escolha dos componentes
- Desenho da Placa de Circuito Impresso (PCB)
- Montagem dos componentes
- Testes eléctricos e funcionais do Sistema Físico
- Montagem do Sistema Físico Final (Placa+ Motor)
- Desenvolvimento do Software em módulos
- Teste dos módulos de software
- Integração dos Módulos

Resultados Finais

Parte do Sistema		**Resultados conseguidos**	**Justificação/Obs**
1.	Sistema físico	Garantiu-se a funcionalidade de todos os componentes. Foi possível integrar todas as partes do sistema. Mesmo aquela que mostra-se mais difíceis de se integrarem, nomeadamente o encoder, o encapsulamento do PCB no veio do motor, alimentação, etc.	
2.	Interface com Utilizador	Foi desenvolvido um modulo que faz interface com o utilizador (protocolo RS-232) que lhe permite acertar as horas e datas e mudar de modo de funcionamento do sistema	
3.	Exibição de Imagem	Conseguiu-se exibir qualquer com Resolução 65x65 pixéis.	Testou-se este modulo sem e com encoder. Neste ultimo caso a imagem com estabilização quase completa.
4.	Relógio Digital	Conseguiu-se apresentar as horas de forma semelhante a um Relógio Digital a partir das leituras que faz do Real-Time Clock.	Embora conseguiu-se mostrar as horas de forma funcional, verificou-se alguma interferência na imagem exibida. Isto dever-se à interferência entre a tarefa que actualiza as horas (Interrupção timer) e interrupção do encoder
5.	Relógio Analógico	Conseguiu-se apresentar as horas de forma semelhante a um relógio de ponteiros. O relógio desenvolvido apresenta dígitos nas posições de 12, 3, 6 e 9 horas. Os ponteiros	Verificou-se a mesma interferência mencionada anteriormente.

		são actualizados em tempo real, sob uma imagem de fundo fixa.	
6.	Sistema Final	O conseguiu-se integrar todos os citados anteriormente sendo que a mudança de modulo é feita através da porta série.	A parte menos conseguida neste projecto foi a estabilização das imagens dada a diferença de tempo entre as interrupções