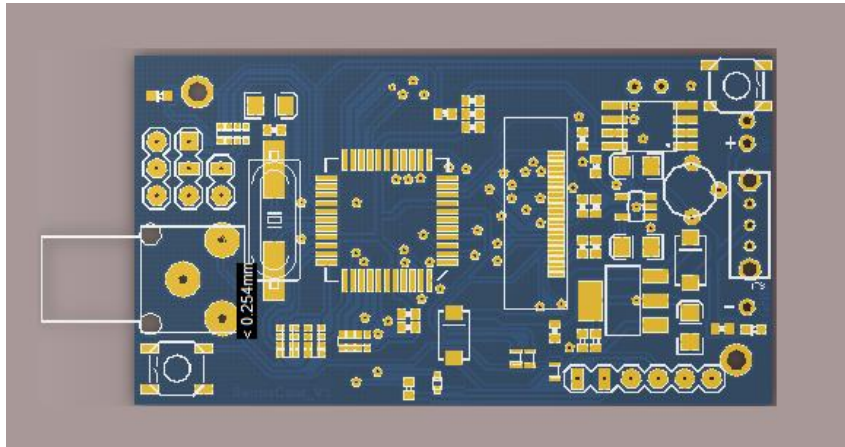




## Relatório do projecto

# Multimedia WSN



Docente: Professor Rui Rocha  
Estudante: César Gaspar nº66005

# 1 Índice

|     |                                  |    |
|-----|----------------------------------|----|
| 1   | Índice .....                     | 2  |
| 2   | Introdução .....                 | 3  |
| 3   | SensorBoard .....                | 4  |
| 3.1 | SenseCam .....                   | 4  |
| 3.2 | Pan&Tilt.....                    | 8  |
| 3.3 | MotelST++s5.....                 | 9  |
| 3.4 | MotelST++s5 e SenseCam .....     | 10 |
| 3.5 | Regulador de tensão .....        | 11 |
| 4   | Software .....                   | 13 |
| 4.1 | SenseCam .....                   | 13 |
| 4.2 | MRM2.....                        | 13 |
| 4.3 | TinyOS.....                      | 13 |
| 5   | Rede Multimedia WSN .....        | 15 |
| 5.1 | Testes .....                     | 16 |
| 6   | Conclusão .....                  | 18 |
| 7   | Trabalho Futuro.....             | 18 |
| 8   | Referências Bibliográficas ..... | 19 |

## 2 Introdução

Este projecto foi elaborado no âmbito da unidade curricular de Redes de Sensores Instituto Superior Técnico – IST, durante o segundo semestre durante o ano lectivo 2013/2014. Os objectivos deste projecto estão relacionados com a implementação de uma aplicação de multimedia para as *wireless sensor network* – WSN.

Uma WSN é constituída por vários nós com a capacidade de recolher informações sobre o meio ambiente que os rodeia. Cada nó comunica bidireccionalmente com os restantes nós da rede. Em adição com a rede em funcionamento serão testadas as capacidades de débito da rede na transmissão das imagens.

Alem disso uma rede de sensores deve ser capaz de transmitir dados de forma segura oferecendo requisitos como, confidencialidade de dados, precisão temporal e elevada autonomia.

Este projecto tem como principal objectivo providenciar capacidades de vídeo para uma rede de sensores tradicional ao desenvolver uma sensorboard, a SenseCam, para o MotelST++s5, com a capacidade de captar imagens e numa segunda parte estabelecer a comunicação entre nós e transmitir imagens pela rede até ao *sink*.

Cada nó é constituído por uma *sensorboard* e o por um MotelST++s5, sendo alguns destes assemblados sobre um sistema mecânico Pan&Tilt.

Finalmente todo o sistema de gestão utiliza o sistema operativo TinyOS para o tratamento da informação pela rede.

### 3 SensorBoard

#### 3.1 SenseCam

A *sensorboard* desenvolvida denominada por SenseCam, tem como base de construção e funcionamento a CMUCam4, figura 1. A CMUCam4 é um projecto *open source* que resultou no fabrico de *shield* para Arduino com uma câmara VGA, OV9665. Esta *shield* incorpora capacidades de tracking, de captura de imagem e de controlo do mecanismo Pan&Tilt. Esta tem um socket para  $\mu$ SD que permite guardas as imagens capturadas e uma saída RCA que fornece as imagens da câmara em formato de vídeo em tempo real. E ainda, a comunicação com a CMUCam4 é feita via TTL UART.



Figura 1 – CMUCam4

Devido ao insuficiente conhecimento na área das câmeras VGA houve a necessidade de recorrer à documentação já existente que permitisse a construção da *sensorboard*. A CMUCam4 foi a escolhida devido à quantidade de informação disponível [1], tais como esquemáticos do circuito electrico da *shield*, *firmware* e bibliotecas disponíveis. Sem esquecer, outro aspecto muito importante foi o facto de poder comunicar com o processador da *shield* através de simples comandos, que é ideal para uma rede de sensores onde se pretende ter o menor débito de informação a percorrer a rede.

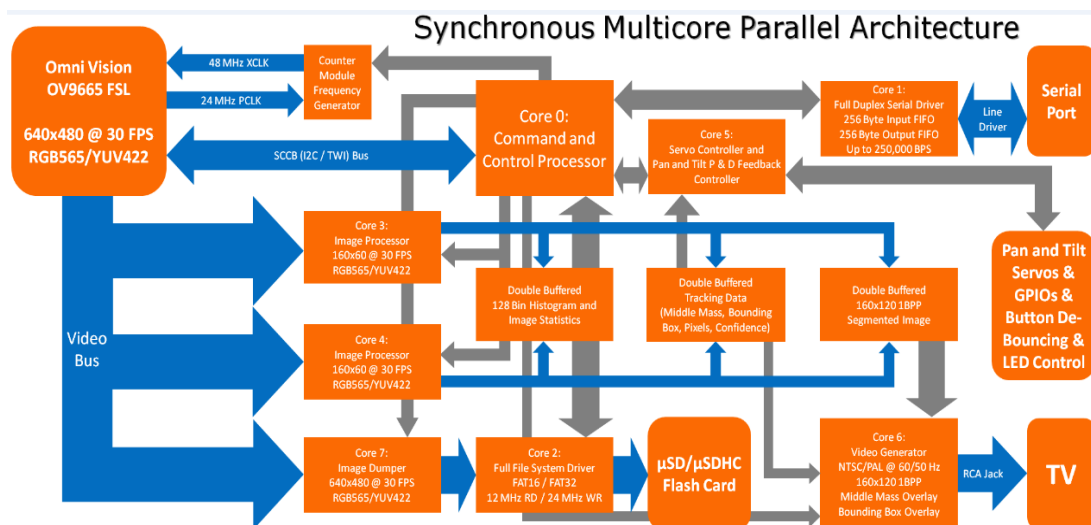


Figura 2 – Arquitectura da CMUCam4

As SenseCam contém as mesmas especificações que a CMUCam4 exceto o cartão  $\mu$ SD. Como o MotelST++s5 contém o mesmo *socket* para o mesmo tipo de  $\mu$ SD, então foi feita a ligação do processador da SenseCam pela Hirose51 ao  $\mu$ SD do MotelST++s5. Desta forma, o  $\mu$ SD fica disponível aos dois processadores.

O processador utilizado na SenseCam é o Parallax P9X32A [3]. Este processador possui oito *cores* independentes, sendo que cada um pode ser usado para determinadas funções específicas. Este tipo de arquitetura é ideal para processamento de imagens. Na figura 2 estão especificadas as funções de cada *core* do processador.

### Fotoreistência

A SenseCam contém uma fotoreistência e a ligação para um sensor de movimento. A fotoreistência, figura 3, permite obter o estado de luminosidade do local onde se encontra a câmera e com base nesta informação ajustar os valores de contraste e brilho da imagem a captar. A fotoreistência está em série com uma resistência de  $10\text{k}\Omega$ , para que a corrente seja baixa. O valor adquirido pelo ADC5 do MotelST++s5 corresponde à queda de tensão provocada pela maior ou menor passagem de corrente pela fotoreistência.

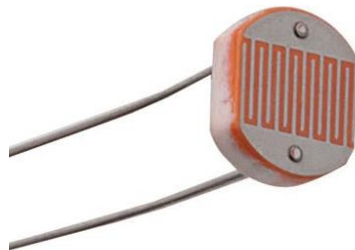
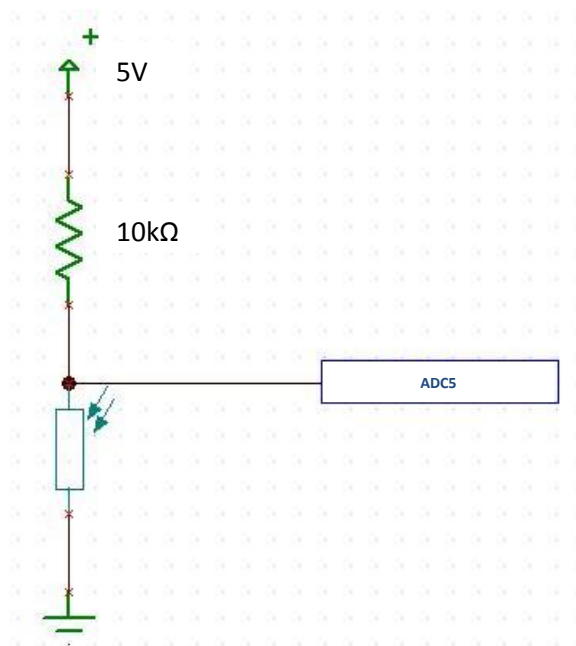


Figura 3 - Fotoreistência.



## Sensor de movimento

Outra capacidade da SenseCam é a possibilidade de integrar um sensor de movimento que permite saber se ocorre alguma alteração no cenário do seu alcance e desta forma captar imagens nesses momentos. Assim, cada nó tem a capacidade de gerar eventos com base na mudança de estados do ambiente, como por exemplo, o aparecimento de uma pessoa no campo de visão da SenseCam gera um evento no nó e este faz a respectiva colheita de fotos para a rede.

Com esta funcionalidade o sistema pode ser configurado para que essas fotografias sejam guardadas no cartão  $\mu$ SD e posteriormente enviadas através da rede, ou simplesmente captar uma imagem e enviá-la pela rede.

O *layout* para o sensor de movimento na SenseCam foi feito para receber um sensor de movimento da Panasonic da família AMN2, figura 3. Estes sensores têm um alcance de detecção de 10 metros, contudo o ângulo de detecção varia de sensor para sensor, sendo que o standard destes são  $100^\circ$ .

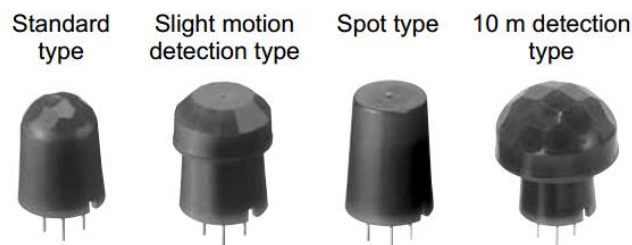


Figura 3 - Sensores de movimento AMN2 da Panasonic

## Botão reset

A SenseCam incorpora um botão que permite reenciar o processador da mesma. Ao executar a acção de *reset* todas as definições do sistema são repostas para os valores iniciais.

## Leds de estado

Para auxílio nas tarefas de *debug* e para visualmente saber o estado das tarefas do processador, a SenseCam contém dois leds, um vermelho e um verde. O led verde iluminado indica se a *sensorboard* está ligada. O led vermelho pode ser controlado digitalmente por comandos e piscar à frequência que o utilizador desejar, e assim associar diversas tarefas a várias frequências do led. Sempre que ocorre transmissão do ficheiro da imagem, tanto para o cartão  $\mu$ SD como via UART, este led pisca a uma frequência de 5Hz.

## Processo de fabrico da SenseCam

Durante este projecto foram produzidas duas *boards* para protótipo e para que o sistema fosse testado com o mínimo de nós, sendo estes no numero dois. O *layout* das *boards* foi desenhado no software Altium, versão 10. As dimensões da *sensorboard* tidas em conta são as dimensões exteriores do MotelST++s5, 6 x 3.3cm, figura 4.



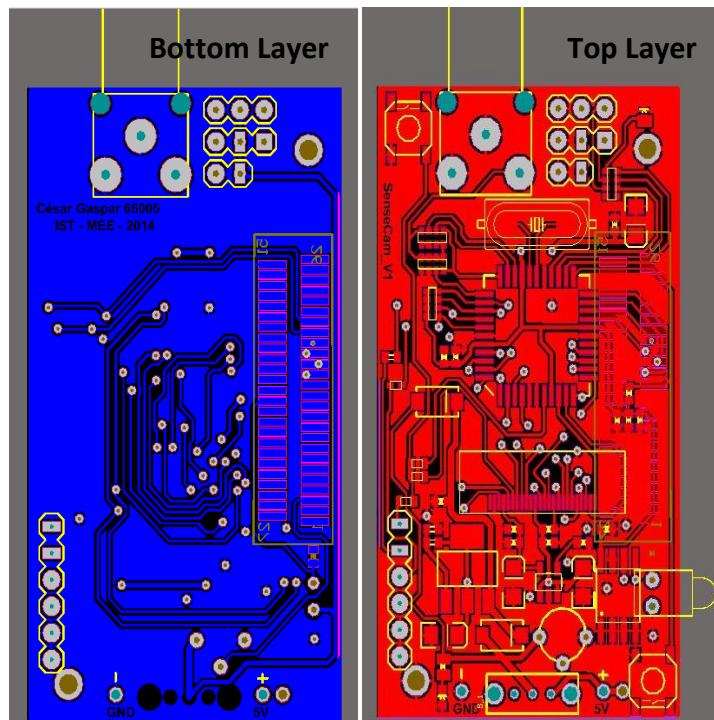


Figura 4 – Layout produzido no Altium da SenseCam.

O processo de fabrico das PCB foi utilizado, PCB de dupla camada com a respectiva metalização das vias do circuito impresso. Para produzir a placa de circuito impresso foi utilizada a CNC.

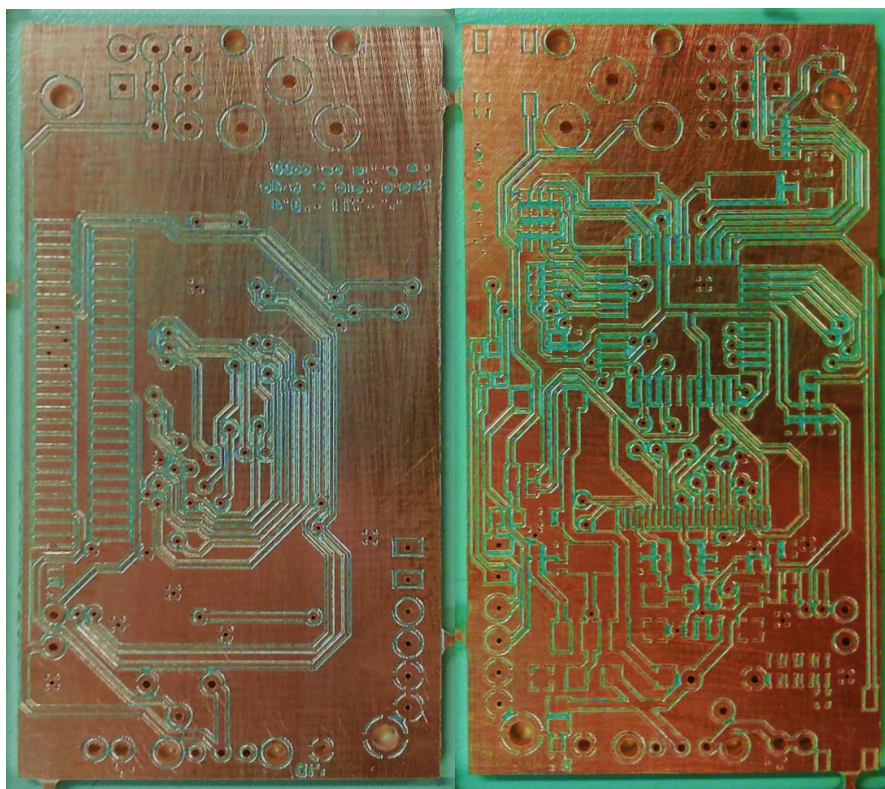


Figura 3 – Bottom e Top Layers da SenseCam em PCB

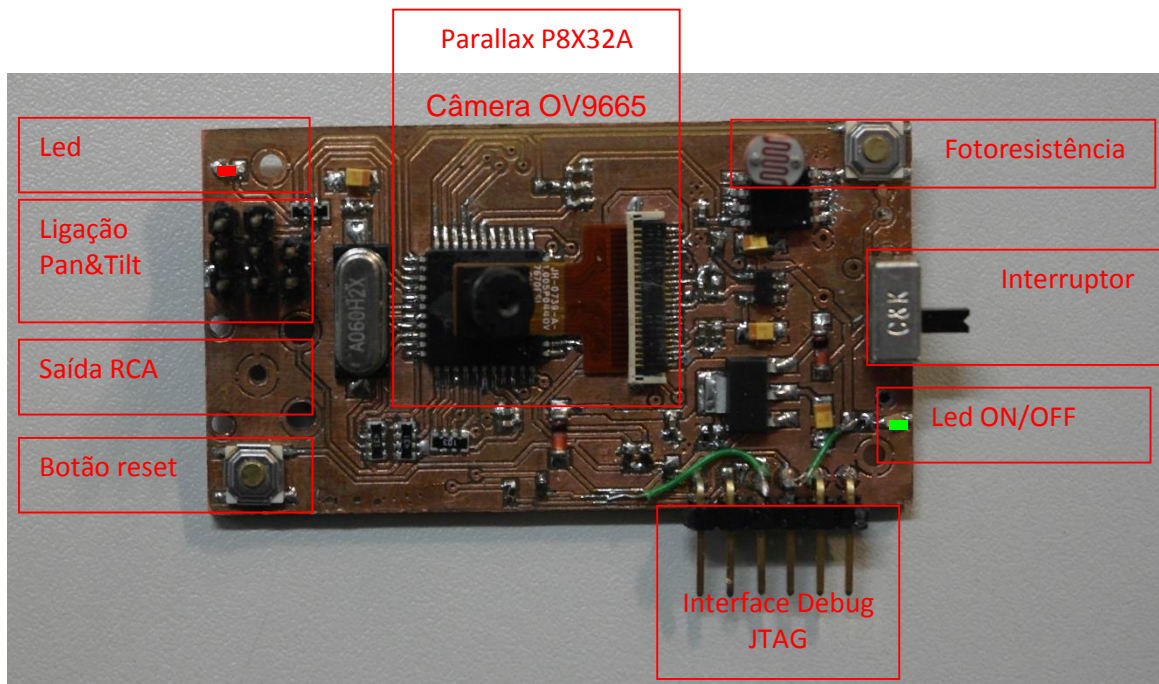


Figura 6 – Visão geral da SenseCam

### 3.2 Pan&Tilt

O sistema mecânico Pan&Tilt é constituído por dois servos que permite orientar a base de fixação, 160° na vertical (*tilt*) e 160° na horizontal (*pan*). Os dois servos estão ligados à SenseCam, e o microcontrolador desta faz o controlo digital através de um sinal pwm correspondente à posição pretendida dos servos.

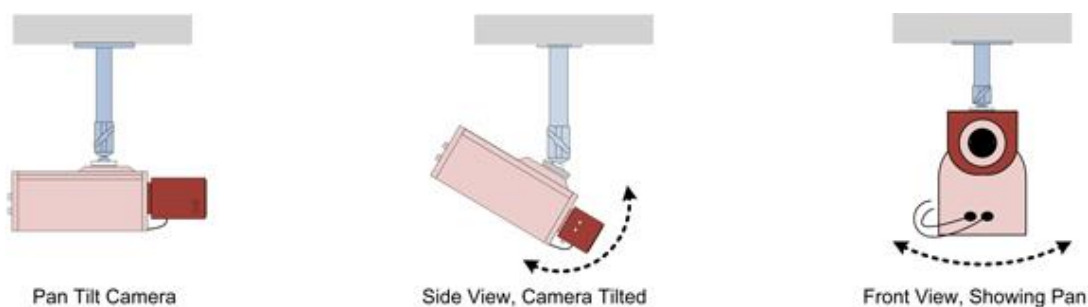


Figura 7 - Princípio de funcionamento de um sistema Pan&Tilt



Os comandos enviados à SenseCam para controlo da Pan&Tilt são:

Tabela 1

| Servo | Comando     | Descrição                 |
|-------|-------------|---------------------------|
| Pan   | SS 0 1 1500 | Posição horizontal de 90° |
| Tilt  | SS 1 1 1500 | Posição vertical de 90°   |

O 4º argumento do comando recebe valores entre 750 e 2500, sendo que 750 corresponde a 0° e 2250 corresponde a 160°.

Como ilustrado na figura 7, o sistema permite obter diversos ângulos de captura de imagem e dessa forma tornar o sistema de multimédia mais versátil.



Figura 8 – Mecanismo Pan&Tilt – servo motores e suporte em PBC.

### 3.3 MotelST++s5

O MotelST++s5 é o quinto mote produzido pelo grupo de investigação GEMS do IST [2]. Este é uma plataforma *state-of-the-art* das redes de sensores desenvolvida para os mais diversos projectos de redes de sensores iniciados no grupo. Ele contém um microcontrolador MSP430F5438A, desenvolvido pela Texas Instruments. Este último pertence à série MSP430F533x, que se distinguem pelo seu baixo consumo e pelas seguintes especificações: 3,3 V LDO, alto desempenho, 12-bit de conversão A/D, duas interfaces de comunicação série universal (USCI), DMA, quatro temporizadores de 16 bits, em tempo real, módulo de relógio com recursos de alarme, e até 74 pinos I/O.

O MotelST++s5 contém 4 interfaces principais: um JTAG para programação e debug do microcontrolador, uma ficha de conexão de 51 pinos para ligação às *sensorboards*, e 2 duas fichas para conectar o driver USB e ou o rádio CC2420, figura 9.

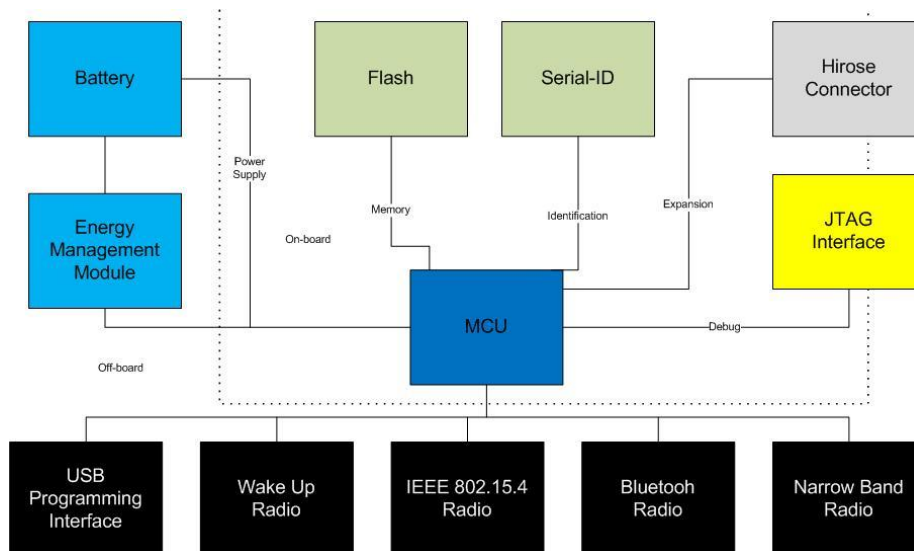


Figura 9 –Arquitectura do MotelST++s5.

Esta plataforma é compatível com os sistemas operacionais mais utilizados, como o TinyOS e o FreeRTOS. Neste projecto foi utilizado o TinyOS.

### 3.4 MotelST++s5 e SenseCam

A comunicação entre o MotelST++s5 e a SenseCam é feita pelo Hirose de 51 pinos. As ligações estabelecidas no conector correspondem a uma ligação UART, a uma ligação SPI e duas ligações para a fotoresistência e sensor de movimento. Para o correcto funcionamento dos *bus* de comunicação é estabelecida a ligação do plano de *ground* entre boards. Finalmente estas ligações estão representadas detalhadamente na figura 10.

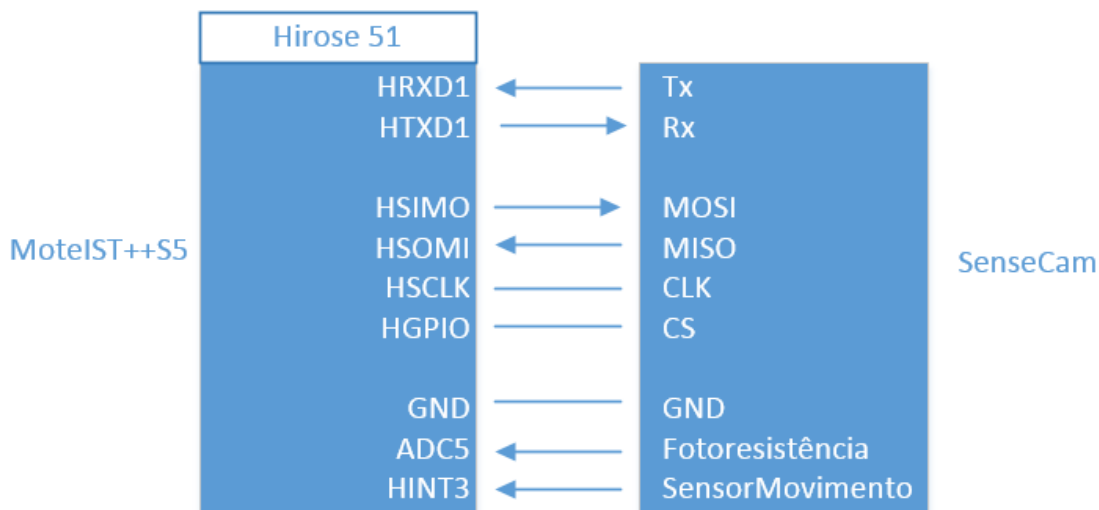


Figura 10 – Ligação entre MotelST++S5 e a SenseCam

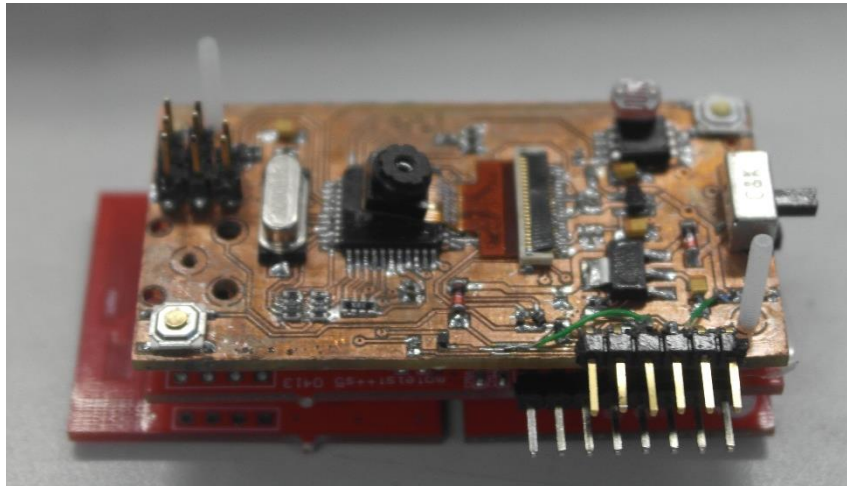


Figura 11 – Nó, MotelST++s5 e SenseCam

### 3.5 Regulador de tensão

Os dois microcontroladores presentes nas duas *boards* necessitam de tensões de alimentação diferentes. O MotelST++s5 necessita de 3.3V e a SenseCam de 5V, desta forma cada nó possui um regulador de tensão que disponibiliza estes valores de tensão para o correcto funcionamento. Além disso o plano de ground deve estar sempre conectado em comum em todos os pontos do circuito e conectado às duas boards.

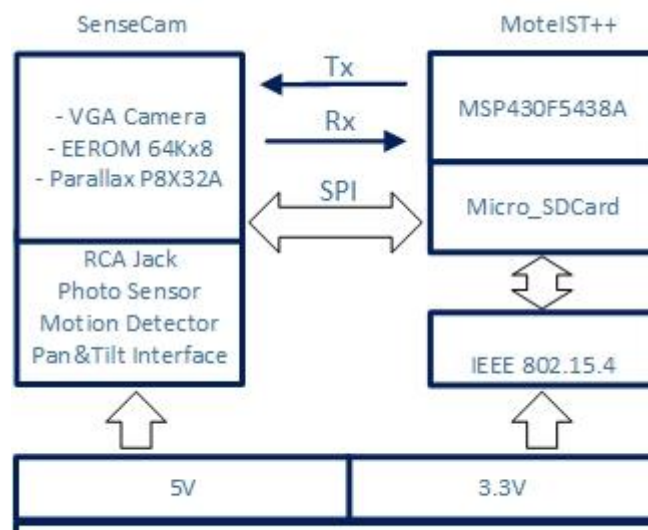


Figura 12 – Power Supply de um nó

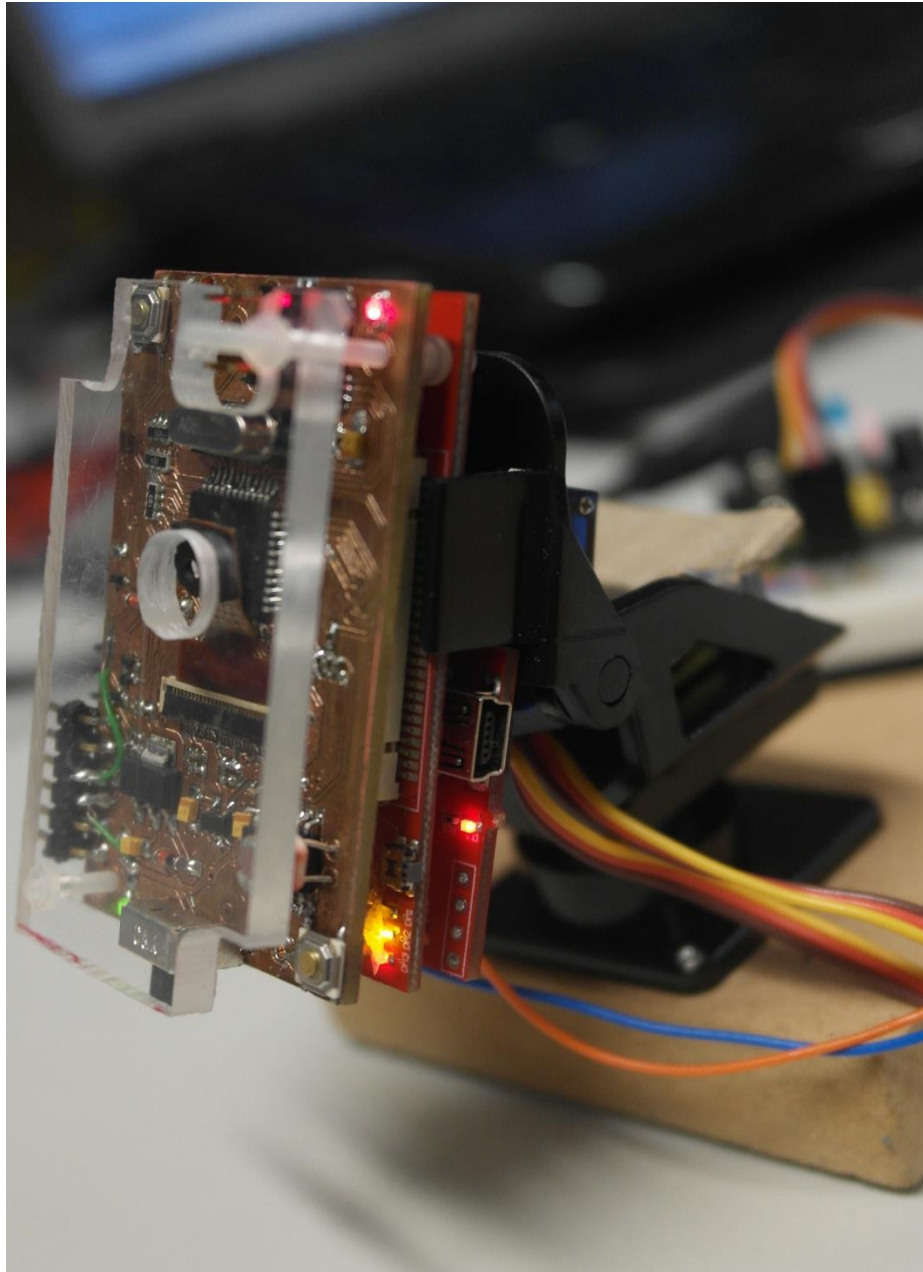


Figura 13 – Montagem final de um nó

A figura 13 representa a montagem final de um nó sobre o sistema mecânico Pan&Tilt. Nesta montagem não está presente o rádio CC2420 do MotelST++s5. No entanto este ficaria alojado no mesmo local que a placa USB do MotelST++s5.

## 4 Software

### 4.1 SenseCam

Na SenseCam é utilizado o *firmware* disponível para a CMUCam4. Este é desenvolvido na linguagem de programação SPIN. Para carregar o *firmware* para o Parallax via porta TTL UART utiliza-se o editor Propeller Tools da Parallax.

Não obstante foram feitas algumas alterações ao *firmware* original, nomeadamente a adaptação dos registos de leitura da câmara, uma vez que a câmara utilizada é ligeiramente diferente da câmara original da CMUCam4. Outra alteração feita foi a modificação da velocidade de comunicação que está definida nos 19200 Bd, no entanto esta pode ser modificada até aos 250000 Bd.

### 4.2 MRM2

No decorrer do projecto foi desenvolvido um MRM, com base no MRM2 já existente para o MotelST++s5. Esta versão do MRM tem como finalidade testar todos as componentes de hardware do nó, sem recorrer a nenhum sistema operativo. O MRM desenvolvido conta com a seguinte lista de comandos que podem ser evocados através de um terminal:

Tabela 2 – Lista de comandos no MRM

| Comando     | Descrição   |
|-------------|---|
| RS          | Reset à SenseCam,   |
| SD          | Coloca a SenseCam em modo <i>sleep</i> ,  |
| '\r'        | Acorda o sistema - <i>wake up</i> ,   |
| L1 0        | Liga o Led vermelho,  |
| L1 1        | Led vermelho pica a 1Hz. O 2º argumento permite escolher a frequência do <i>blink</i> |
| SS 0 1 1500 | Servo Pan posição 90º   |
| SS 1 1 1500 | Servo Tilt posição 90º  |
| DB          | Captura uma imagem no formato bitmap com resolução de 80x60                           |
| DF 0 0      | Captura uma imagem e grava no cartão µSD com resolução de 640x480                     |
| DF 1 1      | Captura uma imagem e grava no cartão µSD com resolução de 320x240                     |
| DF 2 2      | Captura uma imagem e grava no cartão µSD com resolução de 160x120                     |
| DF 3 3      | Captura uma imagem e grava no cartão µSD com resolução de 80x60                       |
| BM 250000   | Define o Band rate para 250000  |

Todos estes comandos foram testados com sucesso. Nestes sistemas o MRM é uma óptima ferramenta para averiguar eventuais falhas de *hardware*, sem necessitar de uma outra ferramenta de *debug* mais complexa.

### 4.3 TinyOS

O TinyOS [4], é um “sistema operativo” *open souce* desenvolvido para plataformas de redes de sensores de baixa potência. Ele é apoiado por uma comunidade mundial de investigadores que continuamente evoluem o TinyOS. Este sistema operativo é também suportado por diversas plataformas de redes de sensores, tais como TelosB, Micaz, CrossBow, sendo que existe uma para o MotelST++s5.



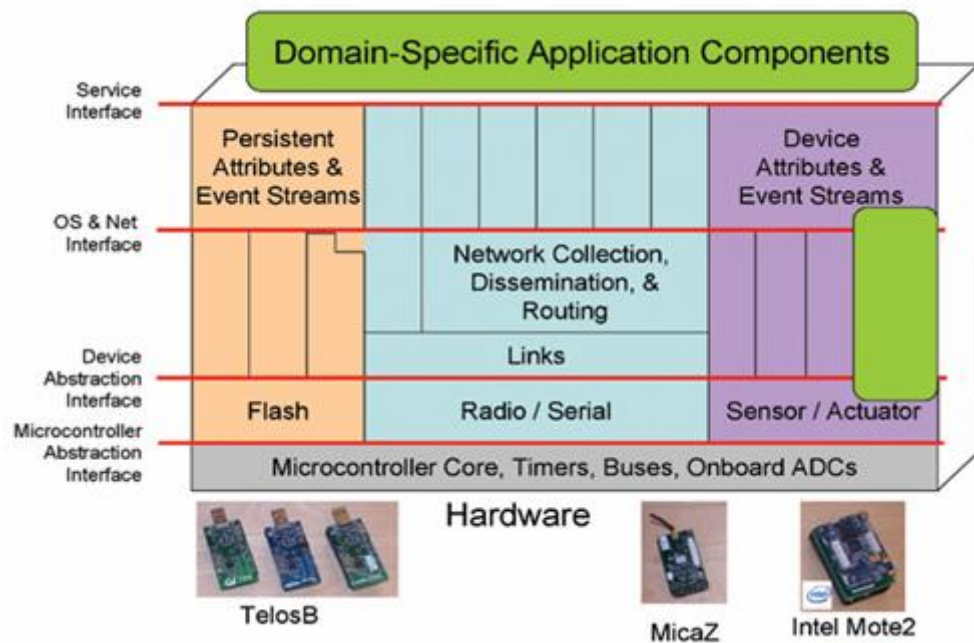


Figura 14 – Arquitectura do TinyOS

Algumas aplicações existentes na plataforma TinyOS, foram testadas, tais como o Blink, Blink to Radio, BaseStation, TesteSerial e TestePrintf. Estes testes permitiram compreender como o TinyOS funciona e como as aplicações são desenvolvidas.

No entanto, devido a falta de tempo nesta parte do projecto não foi possível desenvolver uma aplicação dedicada ao sistema operativo que fizesse uma transmissão correcta do ficheiro de imagem pela rede até ao *sink*.

Para uma rede sem *multihops* teriam de ser desenvolvidas duas aplicações, uma para o nó e outra para o *sink*, com as características seguintes:

A aplicação para o nó teria de receber uma mensagem, via rádio com o comando correspondente à acção desejada enviada pelo *sink*, de seguida descompactar a mensagem e enviar esse mesmo comando pela UART1 para a SenseCam. Por fim a SenseCam responderia com alguma informação.

Se o comando em questão for para captar uma imagem, a aplicação deve criar um *buffer* com tamanho suficiente para armazenar a imagem de acordo com a resolução da imagem a captar. Depois de receber um *ack* da SenseCam a aplicação pode começar a enviar a imagem pela rede, e para isso tem que dividir o *buffer* em partes iguais, tendo estas o tamanho máximo possível que possa ser enviado no tipo de mensagem utilizada.

A aplicação em TinyOS para o nó *sink* necessita de receber todas as mensagens que completam a imagem e armazenar o conteúdo dessas mensagens num ficheiro para que possa ser utilizado posteriormente numa outra aplicação, em Java por exemplo, fazer a reconstrução da imagem.

Para que seja obtida uma imagem com sucesso é necessário ter algum controlo no envio e recepção de todas as mensagens que compõem o *buffer* da imagem. Para isso,

a proposta é que o nó só envie a próxima mensagem depois de receber uma sinalização do *sink* em que como este acusa a recepção com sucesso a mensagem. Não obstante uma outra proposta existente é que em todas as mensagens deve existir a informação de quantas partes é composto o *buffer* e o índice de cada uma dessas partes. Do outro lado, a aplicação no *sink* teria de informar o nó de quantos e quais os pacotes que já forma recebidos.

## 5 Rede Multimedia WSN

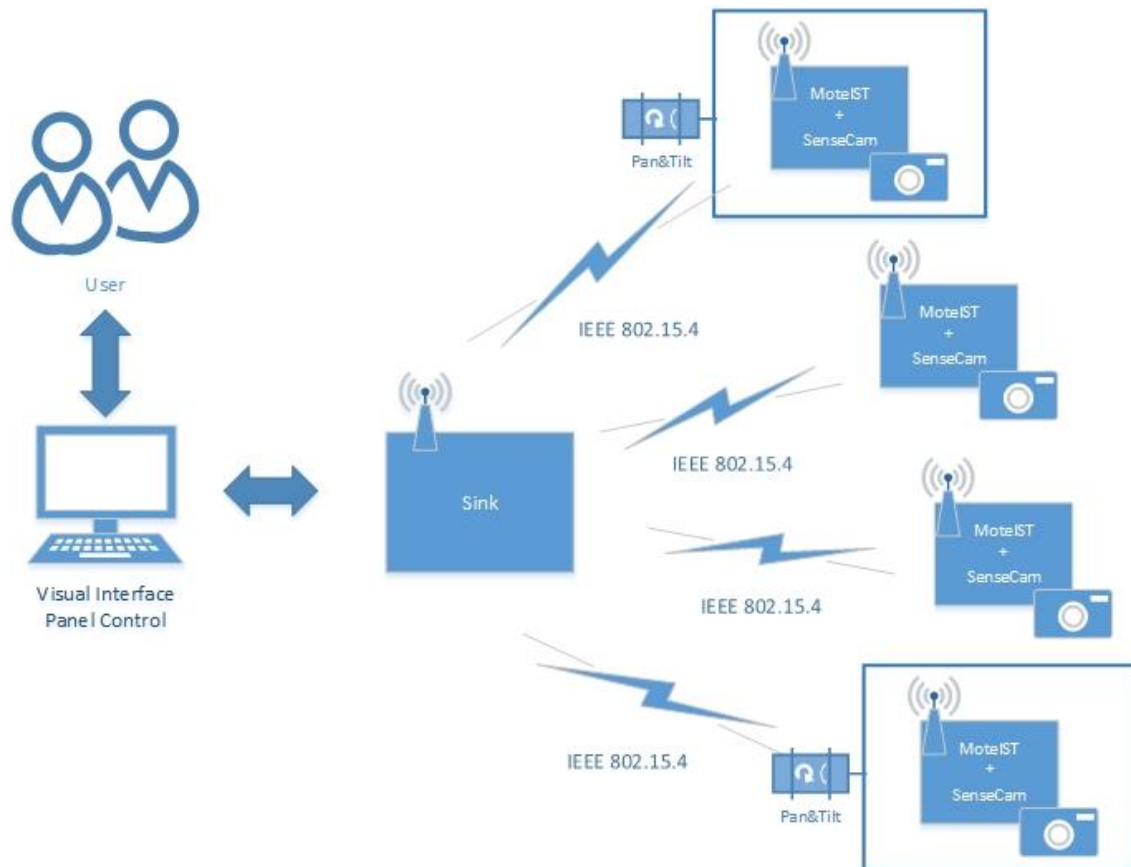


Figura 15 – Arquitectura da rede

A proposta do projecto para a constituição da rede a contruir e testar é representada na figura 15. A rede é constituída por quatro nós, em que dois possuem o sistema mecânico Pan&Tilt. O *sink* é constituído apenas por um MotelST++s5 e respectivo rádio ligado por série a um computador. No computador as imagens recolhidas pelo *sink* seriam reconstruídas e representadas graficamente.

## 5.1 Testes

A SenseCam foi testada com sucesso, tendo sido obtidas as seguintes imagens:



Figura 4 - 640 x 480

A seguinte sequência de quatro imagens representa as diferentes resoluções da câmera:

| Resolução | Pixeis | Bytes |
|-----------|--------|-------|
| 640x480   | 307200 | 38400 |
| 320x240   | 76800  | 9600  |
| 160x120   | 19200  | 2400  |
| 80x60     | 4800   | 600   |



640x480



320x240



160x120



80x60

## 6 Conclusão

Este trabalho permitiu aprofundar conhecimentos na área de sistemas embebidos na concepção da SenseCam. sobre o rigor e complexidade que em nenhum outro trabalho do curso tinha abordado. Destacando o processo de fabrico da SenseCam na CNC de circuitos impressos.

O estudo das várias características das câmeras disponíveis no mercado foi um desafio interessante. Embora tenha sido usado uma câmara fora do uso comercial e ultrapassada em termos de capacidade, esta contém as principais características de qualquer outra câmara.

Em seguida, foi um desafio trabalhar com o microcontrolador da SenseCam, visto que a linguagem de programação, SPIN, é usada exclusivamente em trabalhos de processamento de imagem.

A elaboração deste projecto permitiu ainda aprofundar as técnicas de resolução de *debug* no estabelecimento da comunicação entre os vários módulos.

Ora o trabalho de rede de sensores ficou incompleto porque não foi testado com sucesso nenhuma transmissão de imagem.

Apesar de este trabalho não ter atingido o resultado final esperado, todo o *hardware* ficou a funcionar corretamente, assim como o MRM.

## 7 Trabalho Futuro

Como trabalho futuro fica a implementação do TinyOS correctamente na rede proposta. Este começaria pelo desenvolvimento das aplicações para os nós e para o *sink*. Depois de conseguir a correcta transmissão de imagens, ficam reunidas as condições para estudar as capacidades de débito de imagem numa rede de sensores.





## 8 Referências Bibliográficas

- [1]. <http://www.cmucam.org/projects/cmucam4>
- [2]. <http://leme.tagus.ist.utl.pt/gems/PmWiki/index.php/Projects/MotelST>
- [3]. <http://www.parallax.com/>
- [4]. [http://tinyos.stanford.edu/tinyos-wiki/index.php/Main\\_Page](http://tinyos.stanford.edu/tinyos-wiki/index.php/Main_Page)