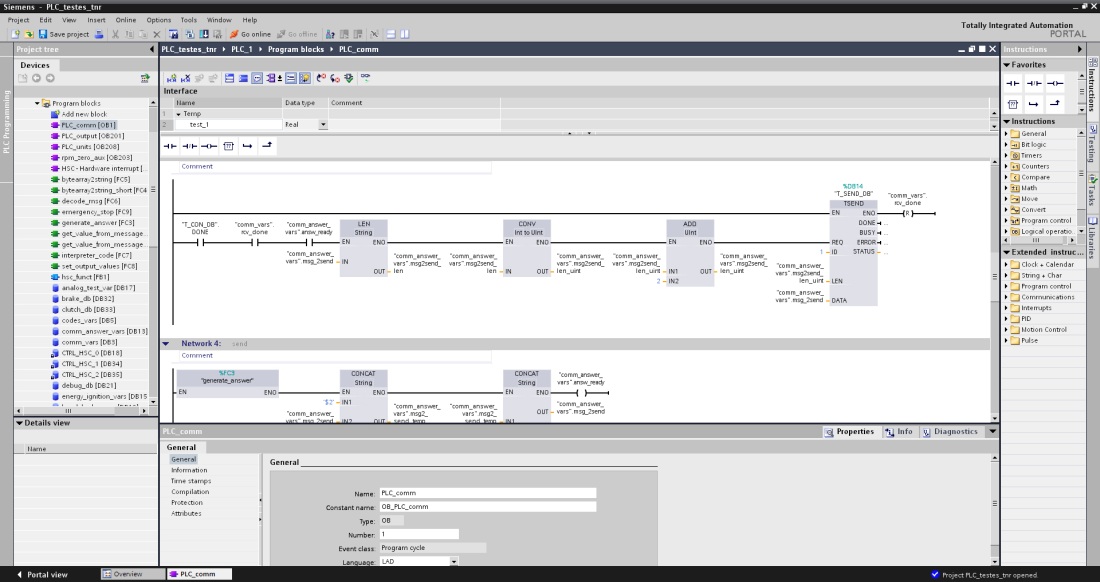
**Capítulo 4 – Software**

Depois de instalado todo o hardware, passou-se à programação de software. Programou-se o autómato para monitorizar e controlar o veículo, uma interface gráfica para visualização do estado do veículo e por fim programou-se ainda uma pequena rotina para a criação automática de gráficos de dados, a partir dos logs extraídos do computador ligado ao autómato. Este capítulo tem o intuito de explicar detalhadamente a estrutura e funcionamento destes três programas.

**Programação da centralina**

O objectivo do programa do autómato é processar os dados recebidos dos sensores, comunicar com o PC enviando o estado do veículo e recebendo comandos, e actuar os diferentes módulos.

A programação é feita em linguagem Ladder, habitualmente utilizada para a programação de autómatos, utilizando o ambiente de programação do fornecedor, SIMATIC STEP 7, ilustrado na FIGURA. Este software já disponibiliza algumas bibliotecas, que se tornaram bastante úteis. O software é bastante intuitivo, tornando-se bastante rápida a sua aprendizagem. No entanto, este tipo de linguagem tem algumas particularidades que por vezes exigiram a reformulação de partes do programa.



O programa autómato é dividido em 4 partes principais:

* Comunicação com o PC;
* Monitorização do estado do veículo;
* Actuação dos diversos módulos;
* Rotinas de emergência.

Todas estas partes interagem paralelamente, trocando informação entre si através de variáveis globais.

**Comunicação com o PC**

A comunicação com o PC é feita por TCP/IP utilizando funções disponibilizadas pelo software de programação.

As mensagens que o autómato envia ou recebe do PC obedecem a um protocolo de comunicação desenvolvido para o efeito.

Este protocolo define uma mensagem única para recepção das ordens vindas do PC e uma mensagem única para envio do estado do veículo para o PC. A opção por mensagens que contenham todas as informações deveu-se à maior velocidade de comunicação obtida com este método. Como a comunicação é feita por TCP/IP, o tamanho da mensagem influencia muito pouco o tempo de envio e recepção, por isso quanto mais informação for enviada por mensagem menos tempo se gasta em comunicações.

As mensagens trocadas entre o PC e o autómato são um conjunto de códigos e valores separados por espaços, que contém todas as informações do veículo, tanto para actuação como monitorização. Como exemplo das mensagens têm-se:

**PC -> PLC**

<STX> STTP 0.0000E+00 SSWP 2.5000E-01 SBKP 8.0000E-01 SCLP 1.0000E+00 SHBP          1 SGER          0 SIGN          0 SVHS 0.0000E+00 SVBS          1 SFAL          0 SHIL          0 SHEL          0 SRTL          0 SLTL          0 STTA          1 SSWA          1 SBKA          1 SCLA          1 SHBA          0 SGEA          0 EEST          0 SIGA          1 <ETX>

**PLC -> PC**

<STX> ATTP -2.5500E+0 ASWP -2.5000E+0 ABKP -3.4900E-1 ACLP -1.7500E+0 AHBP 0 AGER -1 AVHS -1 ARPM +0.0000E+0 AEST 1 AERR -1 <ETX>

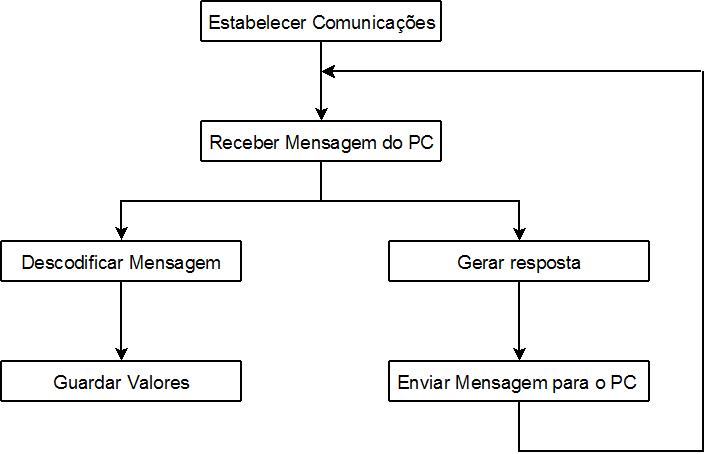
Como se pode verificar, ambas as mensagens iniciam-se com o carácter de início de texto e terminam com o carácter de fim de texto, caracteres 2 e 3 respectivamente. Quando os valores a enviar são números reais estes são enviados em notação científica, normalizando assim a mensagem, pois com esta notação é possível enviar com grande precisão valores de qualquer ordem de grandeza.

A título de exemplo, para a mensagem apresentada o primeiro código na mensagem do PC para o autómato, STTP 0.0000E+00, significa “Set Throttle Position” e pretende enviar a borboleta para a posição 0.

Na mensagem do autómato para o PC, o primeiro código ATTP -2.5500E+0, significa “Answer Throttle Position” e pretende informar o PC que a borboleta se encontra na posição -2.55.

Os restantes códigos encontram-se apresentados em anexo para consulta detalhada.

A estrutura desta parte do programa encontra-se ilustrada na FIGURA.

****

Verifica-se, então, que esta parte do programa tem como objectivo receber do PC a mensagem com as ordens a aplicar a cada actuador. Para isso, depois do autómato receber a mensagem, este tem que a descodificar e guarda os valores em variáveis globais para poderem ser utilizados noutra parte do programa.

Quando o autómato recebe a mensagem, gera também a resposta e envia-a para o, estando assim pronto para iniciar novo ciclo.

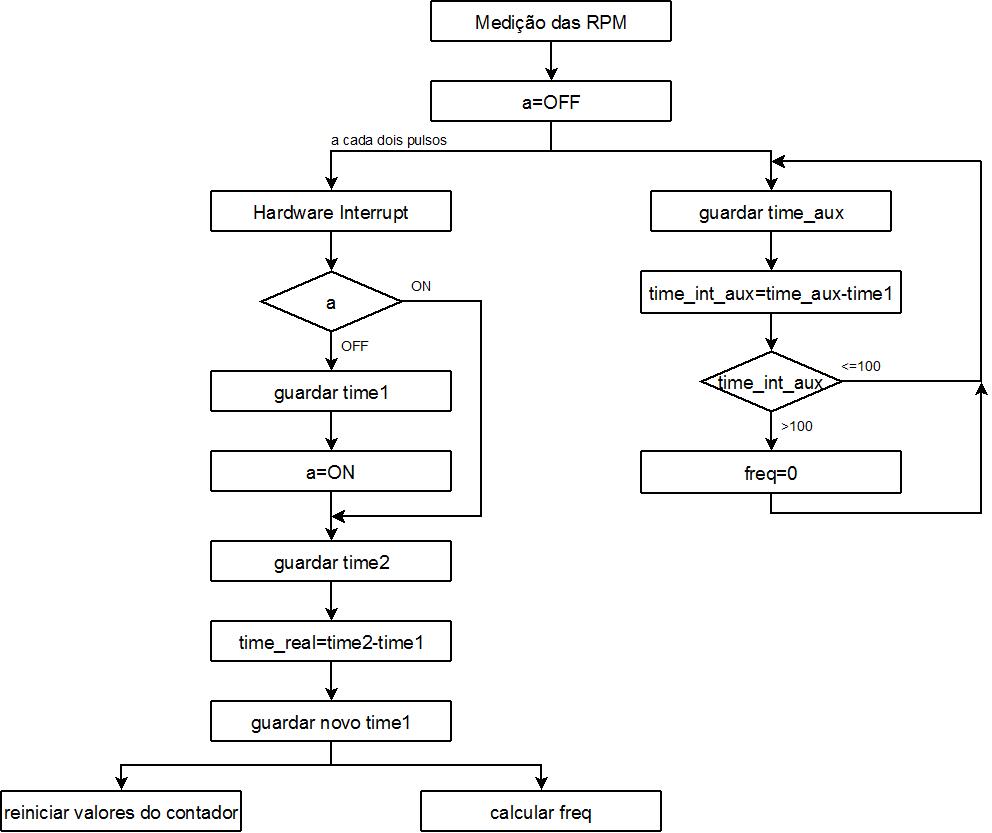
Com o programa desenvolvido é possível fazer todo o ciclo de comunicações em cerca de 50 ms, o que representa um valor bastante satisfatório.

**Monitorização do estado do veículo**

Este bloco do programa tem como principal objectivo a aquisição dos sinais que chegam ao autómato e convertê-los para valores fisicamente lógicos de modo a que, quando forem enviados para o PC, o utilizador os consiga entender facilmente.

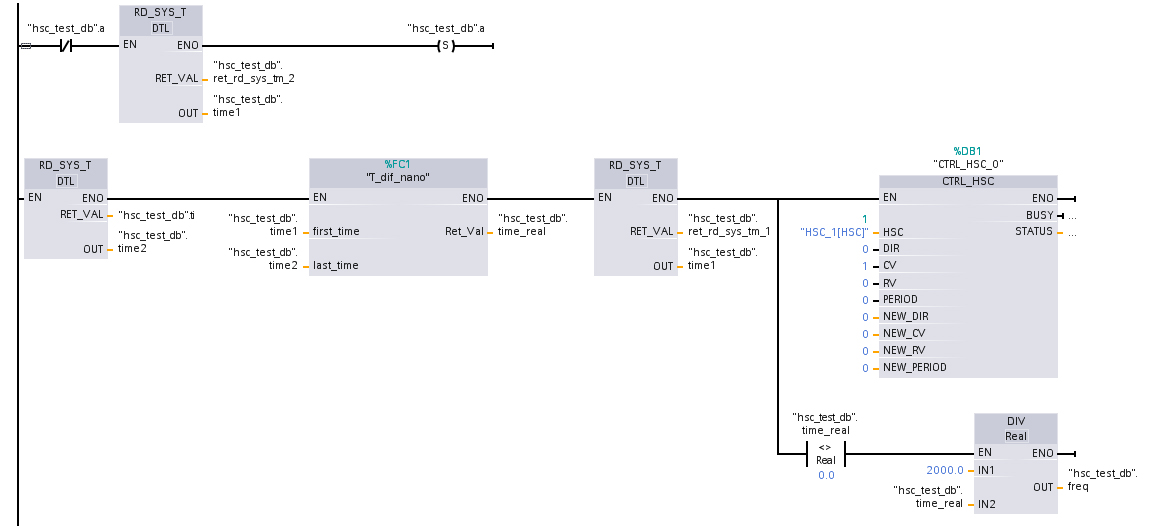
A medição da velocidade de rotação do motor e da velocidade do veículo, são as variáveis a monitorizar que merecem mais destaque, por isso é aqui apresentado o algoritmo usado para a medição das rotações do motor.

A FIGURA apresenta o algoritmo.

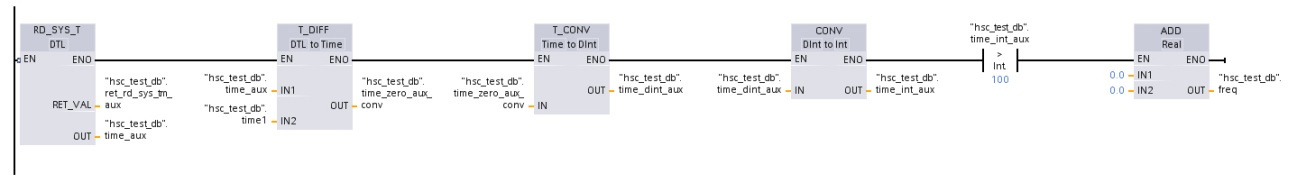


Como se pode verificar, sempre que o sinal das RPM envia dois pulsos o programa entra num Hardware interrupt, que calcula a diferença de tempo entre esses dois pulsos e com esse valor calcular o valor da frequência do sinal para posterior cálculo das RPM (relação linear).

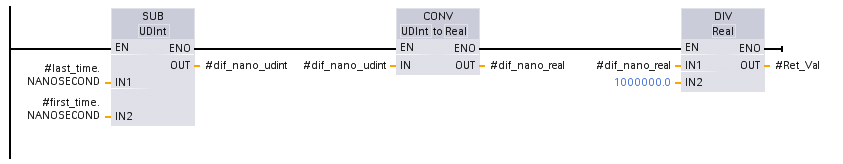
Caso seja a primeira vez que o programa entra no Hardware Interrupt, tem que atribuir um tempo para comparação, pois ainda não teve um interrupt anterior onde esse tempo foi atribuído. O código do PLC que contém toda a rotina corrida pelo interrupt, pode ser consultado na FIGURA.



Outro bloco de programa, que está sempre activo, é utilizado para comparar o tempo actual com o tempo do último interrupt e caso este tempo exceda 100 ms significa que a frequência é muito baixa, podendo ser considerada zero. Este bloco de código é apresentado na FIGURA.



A função que calcula a diferença de tempo foi programada de modo a que essa diferença fosse calculada até aos nanosegundos, pois para frequências elevadas é necessário saber com precisão esse valor. A função encontra-se ilustrada na FIGURA.



**Actuação dos diversos Módulos**

Os valores provenientes das mensagens recebidas do PC são tratados de modo a poderem ser enviados para as saídas digitais e analógicas e assim actuar os diversos módulos.

Para não sobrecarregar o autómato quando desnecessário, a actualização destes valores apenas é feita quando as mensagens que chegam têm valores diferentes dos anteriores.

É nesta parte do programa que também é feito o controlo das luzes e da ignição.

Na rotina da ignição têm-se em atenção o valor da rotação do motor e o estado da ignição para permitir ligar o motor de arranque. (EXPLICAR ALGORITMO DA IGNIÇÃO DETALHADAMENTE?)

**Rotinas de Emergência**

Como foi referido anteriormente, o veículo encontra-se equipado com vários botões de emergência. Quando se pressiona um desses botões, o autómato perde o sinal nessa entrada digital e assim corre uma rotina de emergência.

A rotina de emergência consiste em tirar o controlo do veículo ao computador, actua os pedais do travão e da embraiagem e fecha a borboleta, imobilizando assim o veículo. Este estado mantém-se até que todos os botões de emergência estejam na sua posição normal.

Foram implementadas ainda outras acções que se devem executar em situações de emergência. A título de exemplo, caso as RPM do motor sejam demasiado elevadas, o veículo deixa de actuar o acelerador.

**Interface gráfica**

Para uma mais fácil visualização do estado do veículo, foi realizada uma interface gráfica em GTK. Esta interface permite ainda a actuação de alguns sistemas do veículo, bem como a gravação de um log para posterior análise.

O programa realizado está dividido em quatro separadores distintos:

* Informações detalhadas;
* Luzes;
* Visualização gráfica;
* Log.

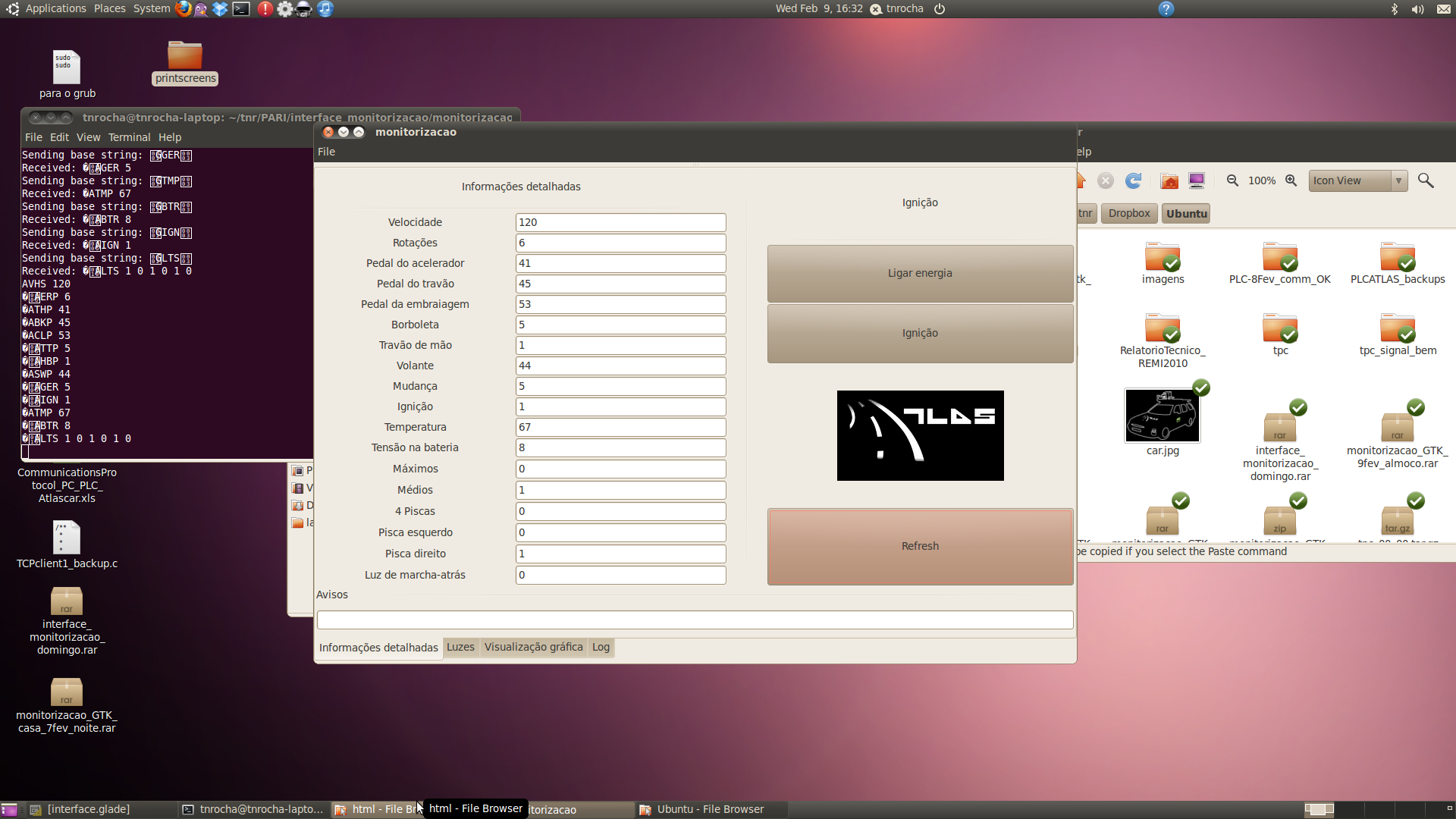
**Informações detalhadas**

O primeiro separador contém as informações detalhadas de todos os sensores do veículo caso se pretenda fazer uma análise cuidada ao estado do veículo. A actualização dos valores pode ser feita em intervalos de tempo predefinidos ou através do botão de Refresh.

Contém também uma pequena área para apresentar mensagens de erro ou pequenos avisos como, por exemplo, se o autómato está a executar alguma rotina de emergência.

O controlo da ignição do veículo é também feito neste separador.

A FIGURA ilustra o aspecto gráfico do separador.



**Luzes**

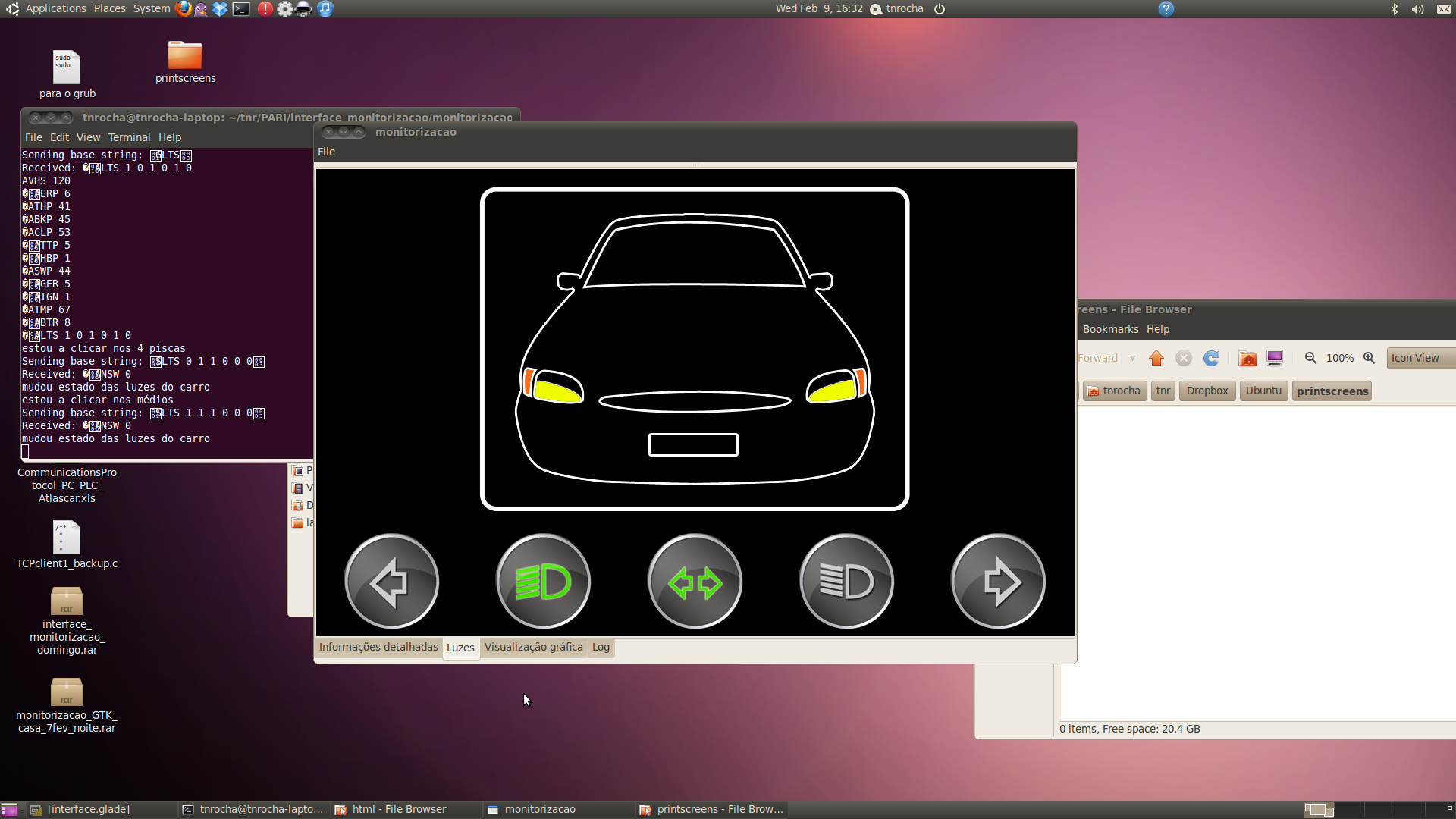
O separador das luzes tem como objectivo a visualização e actuação nas luzes do veículo.

É constituído por duas áreas principais:

* Área de visualização, onde se pode ver uma imagem do veículo com as luzes que se encontram ligadas;
* Área de botões, onde é possível alterar o estado das luzes.

O programa faz uma gestão inteligente da ligação das luzes, ou seja, consoante o estado actual das luzes, o programa age de modo diferente para uma mesma acção. Por exemplo, caso esteja ligado o pisca para o lado direito e o utilizador pretenda ligar o pisca do lado contrário, o programa automaticamente desliga o pisca do lado direito e liga o esquerdo.

A FIGURA apresenta o aspecto gráfico do separador.



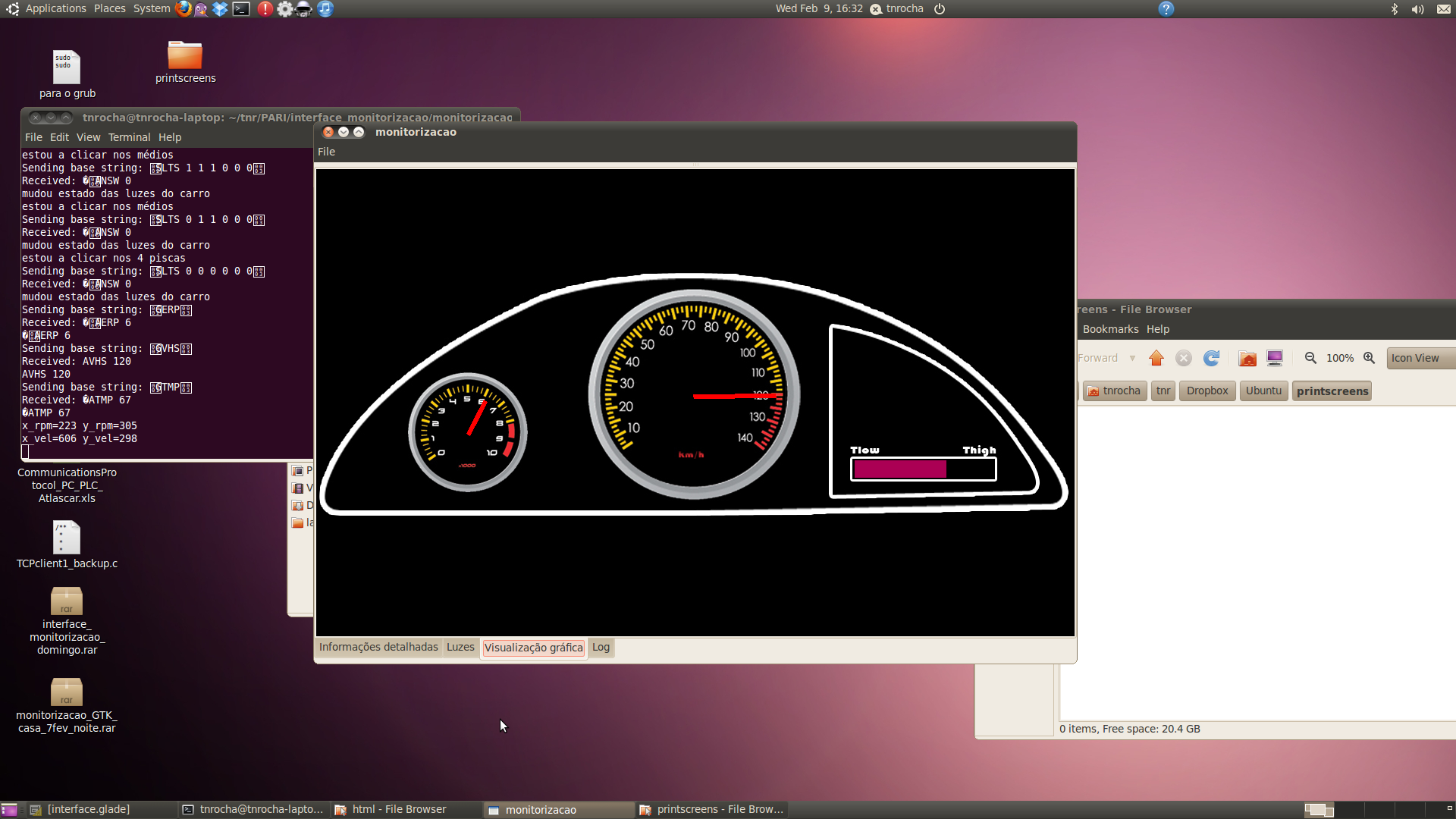
**Visualização gráfica**

Este separador tem como objectivo a apresentação das variáveis de monitorização de uma forma gráfica. As variáveis são apresentadas na forma de um painel de instrumentos de um veículo. Contém três áreas distintas:

* Conta-rotações: onde se pode visualizar as RPM do motor;
* Velocímetro: onde se pode visualizar a velocidade do veículo;
* Outras informações: área onde se podem ver as restantes variáveis (temperatura e sinais de aviso).

A actualização é feita automaticamente em intervalos de tempo pré-definidos.

A FIGURA ilustra o aspecto gráfico do separador.

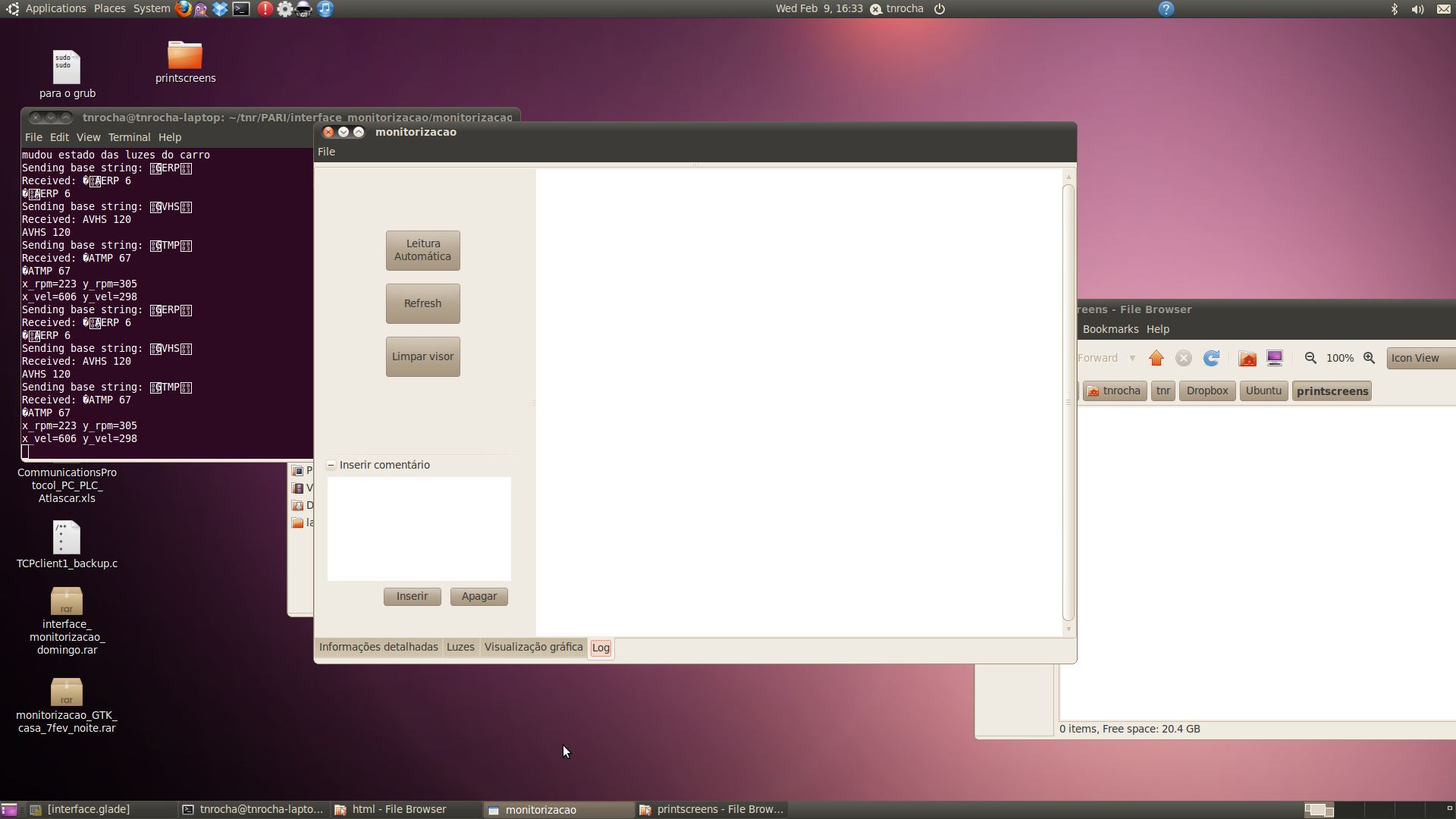


**LOG**

Este último separador tem como objectivo fazer um registo do estado do veículo.

É possível realizar uma actualização dos dados em modo automático ou manual, e permite ainda a inserção de comentários.

A FIGURA apresenta o aspecto gráfico deste separador.



**Programa para análise dos resultados**

Os dados retirados num determinado teste ou viagem são guardados num log. Com o intuito de se fazer uma análise posterior, foi programada uma pequena rotina em MatLab, para a criação automática de gráficos a partir do log.

Esta pequena rotina permite escolher que variável, ou variáveis, se pretende visualizar no gráfico e permite ainda fazer uma animação para uma melhor percepção da evolução temporal dos dados obtidos.