

Postos de Combustível - Automação de Bombas

O processo de automação de postos de combustível envolve diversas etapas, tais como automação da loja de conveniências, automação da pista, automação da retaguarda, controle de tanques de armazenamento e bombas de combustível.

Neste artigo são apresentadas informações relevantes a respeito do desenvolvimento de um sistema de automação de bombas de combustível. Através do qual é possível monitorar toda a atividade das bombas, bem como intervir remotamente nas mesmas em tempo real, efetuando *presets* de abastecimento em dinheiro e/ou volume, bloqueando e/ou liberando abastecimentos, alterando preços dos produtos, dentre outras funcionalidades.

Componentes do Sistema

Tipicamente, um sistema de bombas de combustível automatizado segue um esquema semelhante ao apresentado na *Figura 01*.

Nele, vemos que as bombas de combustível são ligadas ao *Concentrador Universal de Bombas* – CUB. O CUB é o equipamento responsável por centralizar as informações e dados enviados pelas bombas e gerenciar a comunicação entre o servidor da aplicação e a pista. Neste exemplo, apresentamos as funcionalidades do CUB do fabricante *Gilbarco*.

O CUB é peça central no desenvolvimento de sistemas de automação de bombas de combustível e por vezes constitui fator restritivo ao maior número de postos automatizados, principalmente devido ao seu relativo alto preço que, quando associado aos custos de compra e manutenção dos sistemas de *software* necessários, podem inviabilizar este tipo de aplicação em postos de menor porte.

O CUB é ligado, via RS232, a um servidor de aplicação, de preferência dedicado somente a esta função, onde rodam o *software* de configuração do CUB distribuído pelo fabricante e o *software* de monitoria de bombas desenvolvido pela *software house*.

Alguns cuidados devem ser tomados ao realizar este tipo de automação sob pena de prejuízos e/ou problemas sérios no correto andamento das atividades na pista, como é o caso, por exemplo, da importância de se ter um servidor de aplicação dedicado. Uma vez configuradas as bombas para trabalhar sob supervisão do CUB, caso o servidor de aplicação trave e/ou caia por algum motivo, todas as operações da pista são interrompidas até que o sistema seja regularizado ou que as bombas sejam re-configuradas para operar de forma não supervisionada.

O sistema pode também ter acesso via *Internet*, caso a *software house* disponibilize algum tipo de acesso remoto às informações gerencias do posto. Tipicamente se faz necessário desenvolver alguma interface deste tipo, pois a maioria dos proprietários de postos de combustível os administra remotamente, principalmente no caso específico de postos de estrada e cidades do interior.

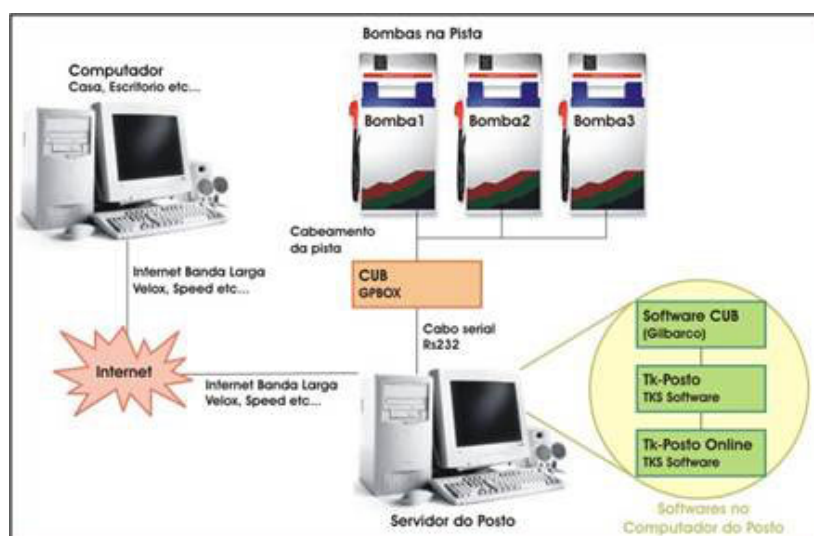


Imagem 01. Esquema típico de sistema de automação de bombas de combustível/

Ambiente de Desenvolvimento

Para iniciar o desenvolvimento de uma aplicação de automação de bombas de combustível, deve-se fazer o *download* do *Software Development Kit* - SDK do fabricante. No caso da *Gilbarco*, o SDK é composto por um simulador de bombas de combustível, uma ferramenta de configuração do CUB e um simulador do CUB.

Para iniciar o desenvolvimento, é necessário instalar, configurar e executar o simulador de bombas de combustível (*PumpSim.exe*), mostrado na *Figura 02*, em um computador que deve ser ligado, via RS232, ou computador de desenvolvimento. O simulador será responsável por enviar informações ao computador de desenvolvimento, permitindo que o programador simule todas as operações que podem ser executadas pelo frentista em uma bomba de combustível real.

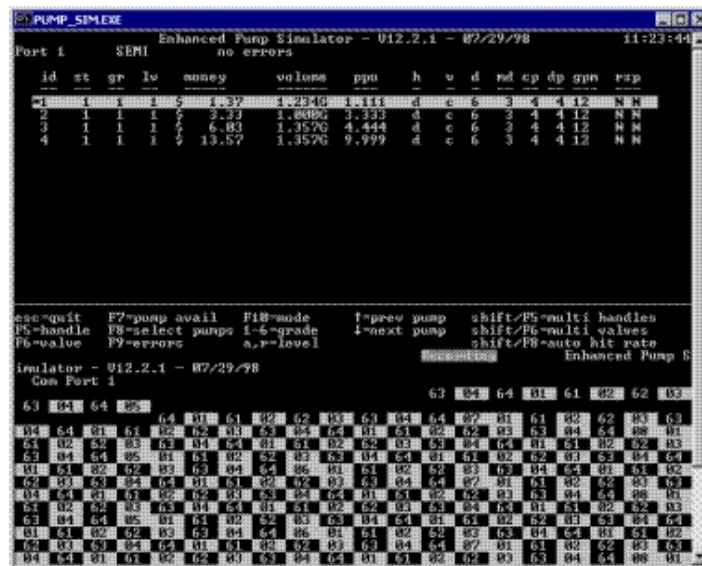


Imagem 02. Simulador de Bombas de Combustível da Gilbarco (*PumpSim.exe*).

Após executar o simulador no computador auxiliar, deve-se executar a ferramenta de configuração do CUB (*CUBConfig.exe*) no computador de desenvolvimento, conforme mostrado na *Figura 03*. Esta ferramenta permite configurar um CUB real ou um CUB Simulado, indicando dentre outras coisas, quantas bombas de combustível estão ligadas ao mesmo.



Imagem 03. Ferramenta de configuração do CUB da Gilbarco (*CUBConfig.exe*).

Uma vez configurado para o modo CUB Simulado, deve-se executar o *software* Simulador de CUB (*CubDemo.exe*) na máquina de desenvolvimento. Surgirá então uma tela tipo MS-DOS conforme mostrado na *Figura 04*, iniciando assim a comunicação entre o servidor simulado de bombas de combustível (*PumpSim.exe*) e o simulador de CUB (*CubDemo.exe*).

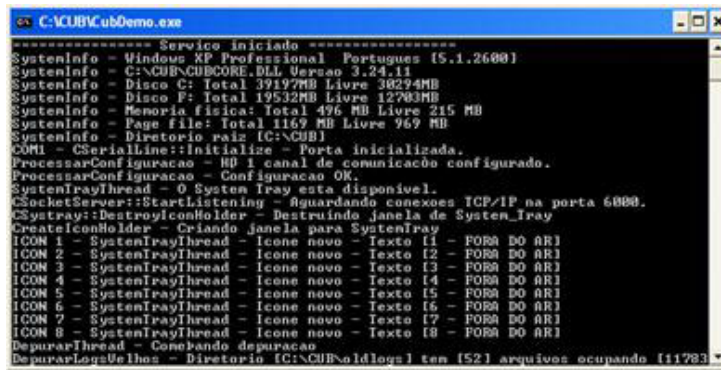


Imagem 04. Simulador de CUB da Gilbarco (CubDemo.exe).

Quando o simulador de CUB é executado, surge no canto inferior direito da tela, um ícone para cada bomba, conforme configuração feita na ferramenta de configuração de CUB (CUBConfig.exe). Inicialmente, todas as bombas aparecem marcadas com um "X", e apresentadas no estado "Fora do Ar". À medida que vai sendo estabelecida a comunicação entre o servidor simulado de bombas de combustível (PumpSim.exe) e o simulador de CUB (CubDemo.exe), as bombas mudam de estado, conforme mostrado na Figura 05.



Imagem 05. Ícones das bombas criados pelo Simulador de CUB da Gilbarco (CubDemo.exe).

Os estados das bombas podem ser vistos passando o mouse sobre cada um dos respectivos ícones da barra do Windows. Internamente na aplicação de automação os estados constituem constantes retornadas pelo CUB, e devem ser declaradas conforme segue:

```
//--- Declaração de Constantes Globais de Estado das Bombas ---
const SFORADOAR      = '0';
const SDISPONIVEL    = '1';
const SABASTECENDO   = '2';
const sFIMDEVENDA    = '3';
const SBICOFORA      = '5';
const SBOMBAFECHADA  = '6';
const SPAUSA         = '8';
const SDESCONHECIDO  = '7';
const SAUTORIZADA    = '9';
```

Faz-se necessário que o desenvolvedor crie uma representação icônica para cada um dos estados que podem ser assumidos pela bomba dentro da aplicação, de forma a facilitar a representação dos mesmos para o usuário final. Na Figura 06 apresentamos uma legenda completa dos estados, o momento em que os mesmos ocorrem, bem como uma representação icônica proposta.



Imagem 06. Ícones das bombas criados pelo Simulador de CUB da Gilbarco (CubDemo.exe).

Comunicação com o CUB

A comunicação entre o CUB e aplicação de automação desenvolvida é feita através das funções disponibilizadas pelo fabricante, presentes na *CUB32.dll*. Segue a lista das funções e suas respectivas declarações:

```
//--- Declaração de Funções disponíveis em CUB32.dll ---

function CUBAbrirBomba(NroBomba: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBLerEstado(NroBomba: LongInt; Estado: Pointer): LongInt; stdcall; external 'CUB32.dll';
function CUBAutorizarBomba(NroBomba: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBDesautorizarBomba(NroBomba: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBFecharBomba(NroBomba: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBContinuarAbastecimento(NroBomba: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBSetarNivelDePreco(NroBomba: LongInt; NivelDePreco: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBLerRTM(NroBomba: LongInt; var RealTimeMoney: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBConfirmarLeituraVenda(NroBomba: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBDescarregarDll: LongInt; stdcall; external 'CUB32.dll';
function CUBProgramarPreco(NroProduto: LongInt; NivelDePreco: LongInt; NovoPreco: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBMapearProduto(NroBomba: LongInt; NroBico: LongInt; NroProduto: LongInt ): LongInt; stdcall; external 'CUB32.dll';
function CUBAutorizarBico(NroBomba: LongInt; NivelDePreco: LongInt; NroBico: LongInt ): LongInt; stdcall; external 'CUB32.dll';
function CUBLerTotais(NroBomba: LongInt; NroProduto: LongInt; NivelDePreco: LongInt; EncerranteLitros: Pointer; EncerranteDinheiro: Pointer; var Preco: LongInt): LongInt; stdcall; external 'CUB32.dll';
function CUBLerVenda(NroBomba: LongInt; var NroProduto: LongInt; var NivelDePreco: LongInt; var Mililitros: LongInt; var Dinheiro: LongInt; var Preco: LongInt ): LongInt; stdcall; external 'CUB32.dll';
function CUBPresetLitros(NroBomba: LongInt; NivelDePreco: LongInt; NroBico: LongInt; CentiLitrosMaximo: LongInt ): LongInt; stdcall; external 'CUB32.dll';
function CUBPresetDinheiro(NroBomba: LongInt; NivelDePreco: LongInt; ValorMaximo: LongInt ): LongInt; stdcall; external 'CUB32.dll';
function CUBLerPaginaTag( NroTag : LongInt; NroPagina: LongInt; DataBuffer: Pointer) : LongInt; stdcall; external 'CUB32.dll';
```

Dentre as funções da *CUB32.dll* destacam-se:

Ler Estado da Bomba (CUBLerEstado):

Esta função retorna o estado atual de todas as bombas ou de uma bomba específica.

```
function CUBLerEstado(NroBomba: LongInt; Estado: Pointer): LongInt; stdcall; external 'CUB32.dll';
```

Parâmetros:

- long NroBomba - {recebe valor de de 0 a 99}
- char *Estado - {Buffer alocado pelo PDV onde será informado o estado solicitado}

Retorno:

- 0 = O CUB Server recebeu a ordem corretamente.
- 1 = Erro nos parâmetros. O número de bomba é incorreto.
- 2 = Erro de comunicação entre a dll e o servidor.
- Outro = Código de erro retornado pelo Windows. Ver winsock2.h ou winerror.h.

Nota:

- Se NroBomba = 0, será informado o estado de todas as bombas, sendo cada byte o estado de uma delas.
- Se NroBomba <> 0, só será informado o estado da bomba solicitada.

Leitura de Venda das Bombas (CUBLerVenda):

Esta função deve ser utilizada quando a bomba está em estado de "FIMDEVENDA" (após o fim de um abastecimento na pista). Após ter chamado esta função e gravados os dados na base de dados do software de automação de bombas, deve ser chamada a função *CUBConfirmarLeituraVenda* para que a bomba fique livre para um novo abastecimento.

```
function CUBLerVenda(NroBomba: LongInt; var NroProduto: LongInt; var NivelDePreco: LongInt; var Mililitros: LongInt; var Dinheiro: LongInt; var Preco: LongInt): LongInt; stdcall; external 'CUB32.dll';
```

Parâmetros:

- long NroBomba - {recebe valor de de 1 a 99}
- long *NroProduto - {pointer onde será informado o código de produto}
- long *NivelPreco - {pointer onde será informado o nível de preço}
- long *Mililitros - {pointer onde será informada o volume do abastecimento}
- long *Dinheiro - {pointer onde será informado o valor do abastecimento}
- long *Preço - {pointer onde será informado o preço unitário do produto}

Retorno:

- 0 = CUB Server recebeu a ordem corretamente.
- 1 = Nº de bomba incorreto ou bomba não está em estado FIMDEVENDA.
- 2 = Erro de comunicação entre a dll e o servidor.
- Outro = Código de erro retornado pelo Windows. Ver winsock2.h ou winerror.h.

Leitura da Venda em Tempo Real

Esta função permite que o usuário veja na tela do computador os valores do abastecimento que esteja acontecendo na pista. O valor retornado não é totalmente *real-time*. Existem vários atrasos envolvidos no processo de comunicação que fazem com que o valor não seja totalmente preciso, mas a aproximação é muito boa.

```
function CUBLerRTM(NroBomba: LongInt; var RealTimeMoney: LongInt): LongInt; stdcall; external 'CUB32.dll';
```

Parâmetros:

- long NroBomba - {recebe valor de de 1 a 99}
- long *RTM - {pointer onde será devolvido o ultimo valor lido da bomba}

Retorno:

- 0 = CUB Server recebeu a ordem corretamente.
- 1 = Erro nos parâmetros. Número de bomba errado ou bomba não configurada.
- 2 = Erro de comunicação entre a *dll* e o servidor.
- Outro = Código de erro retornado pelo *Windows*. Ver *winsock2.h* ou *winerror.h*.

Tipicamente, para implementar o tratamento mínimo da comunicação com o CUB e fazer a automação das bombas, é necessário iniciar um novo projeto, adicionar um componente *TTimer* e implementar o seguinte algoritmo:

1. Lê o estado de todas as bombas juntas (*CUBlerEstado*)
2. Faz um *loop* para processar o estado lido de cada uma das bombas
3. Caso o estado da bomba tenha mudado em relação ao estado anterior
 - 3.1. Atribue o novo estado à bomba atualizando as representações visuais
 - 3.2. Caso o novo estado = FIMDEVENDA
 - 3.2.1. Faz a leitura das informações da venda (*CUBlerVenda*)
 - 3.2.2. Confirma a leitura da venda para o CUB (*CUBConfirmarLeituraVenda*)
 - 3.2.3. Lê os encerrantes de dinheiro e litros da bomba (*CUBlerTotais*)
 - 3.2.4. Atualiza a tela e/ou a base de dados do sistema

Uma vez implementado o algoritmo, podemos caprichar na interface visual do sistema, de forma a obter um resultado semelhante ao mostrado na *Figura 07*, onde o usuário é capaz de visualizar o estado de todas as bombas e executar todas as funções disponibilizadas pela SDK de forma muito fácil e intuitiva.

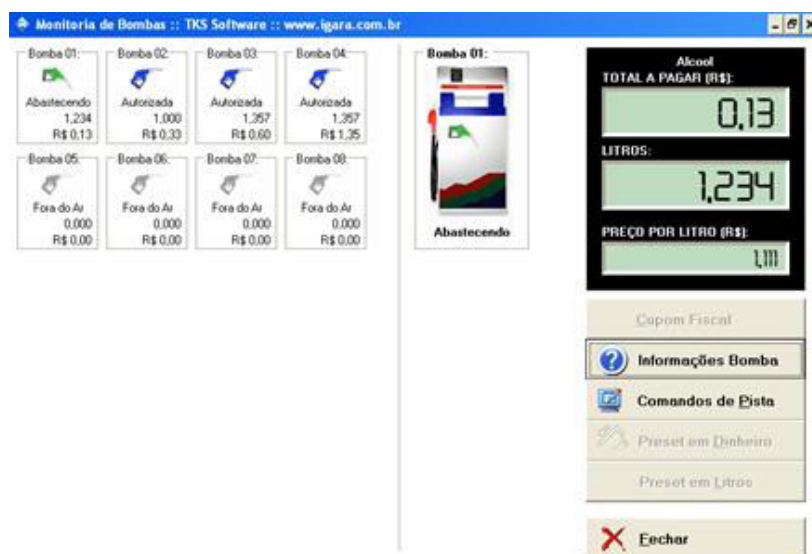


Imagem 07. Aplicação completa de automação de bombas de combustível da TKS Software.

Aplicações como mostrada na *Figura 07* permitem o controle completo da pista a partir de uma central, que pode estar localizada, por exemplo, na loja de conveniências do posto.

Através de funções de *preset* em valor e volume, há muito é possível tecnicamente que os postos sejam operados com número mínimo de frentistas ou até mesmo sem eles, como ocorre na grande maioria dos países desenvolvidos.

No entanto, por motivos populistas, protecionistas ou até mesmo eleitoreiros, o Brasil insiste em resistir ao progresso, impedindo por lei que os postos de combustíveis adotem tal tipo de tecnologia de forma a se modernizar, reduzindo seus custos operacionais e entrando em sintonia com as tendências mundiais.

Este tipo de tecnologia deve ser usado somente para fins de automação, supervisão e controle da pista e seus abastecimentos, não podendo ser usado para mudança do conceito de atendimento dos postos.

Posturas intervencionistas como estas dificultam a popularização das tecnologias e consequentemente o crescimento de todos os setores da economia ligados a ela. Sob a pretensa justificativa de proteção à classe frentista, o estado:

- Onera os proprietários de postos e consequentemente os consumidores finais, que no final da cadeia geram altos custos operacionais dos postos.

- Dificulta diretamente o crescimento de toda uma cadeia de pesquisa e desenvolvimento de alta tecnologia em *hardware* e *software* que, importante ressaltar, também gera emprego e renda e é base sólida para o crescimento e desenvolvimento de um país.

Questões como estas devem ser resolvidas através de políticas de educação, qualificação e recolocação de mão-de-obra. Pois a nossa diferença, quando comparados aos demais países, não está na nossa capacidade técnica de desenvolvimento e aplicação da tecnologia, mas sim no fato de que os países desenvolvidos, antes mesmo de automatizar seus postos, descobriram através da educação, que o crescimento e desenvolvimento de um país em uma economia de mercado globalizada, se dá com o mínimo possível de intervenção do estado.

Conclusão

Este artigo serve como uma rápida introdução ao processo de automação de bombas de combustível, cobrindo de forma geral os principais conceitos e ferramentas envolvidas no processo de implementação deste tipo de solução.

A aplicação dos conceitos apresentados permite o desenvolvimento de soluções completas com diversos recursos avançados, criados sob medida para as necessidades deste nicho de mercado. No entanto o desenvolvedor deve estar pronto para encarar um mercado altamente fechado que dificulta e muito, a entrada de novas soluções. Principalmente no que diz respeito às bandeiras, estas procuram evitar que seus filiados utilizem *softwares* de terceiros que não aqueles já previamente acordados com a bandeira em questão.

Sistemas como estes envolvem o correto funcionamento e operação de diversos dispositivos e por consequência devem estar associados a contratos de manutenção, pois os postos normalmente não dispõem de mão-de-obra especializada para tanto. No entanto um problema muito importante a ser contornado pelo desenvolvedor, e que dificulta bastante a entrada neste mercado, são as distâncias físicas, principalmente quando se trata de postos de estrada ou cidades do interior.

É importante ressaltar, no entanto, que o desenvolvimento e consequente implantação do sistema em um cliente final requer, dentre outras coisas, conhecimentos a respeito da legislação normativa específica de postos de combustível, bem como conhecimentos técnicos sobre a interface e configuração as bombas, tópicos estes não abordados neste artigo.

Maiores informações podem ser obtidas junto ao fabricante no site www.gilbarco.com.br.



Sobre o Autor

Victory Fernandes, consultor técnico do The Club.

E-mail: suporte@theclub.com.br