

AGT DISPLAYS

# Guia de Desenvolvimento – TFT Módulo Serial

Interfaces inteligentes com o ASIC DWIN e o software DGUS

**VERSÃO**

2.1

**DATA**

09/09/2026

**APLICAÇÃO**

Displays TFT Módulo Serial com ASIC DWIN

**PÚBLICO**

Engenheiros e Desenvolvedores

# 1. INTRODUÇÃO

---

O software DGUS (DWIN Graphic Utilized Software) é uma ferramenta profissional para desenvolvimento de interfaces para TFT Módulo Serial de sistemas industriais e embarcados. Através desta plataforma, é possível criar interfaces gráficas sofisticadas e intuitivas para uma ampla gama de aplicações industriais e comerciais.

O DGUS trabalha em modo *orientado a variáveis*: a tela não recebe comandos de desenho, e sim valores gravados em endereços de memória. O que cada endereço faz na tela é definido em arquivos de configuração gerados no PC (13.bin, 14.bin e 22.bin). Por isso o firmware do microcontrolador é simples – ele apenas lê e escreve variáveis.

## CARACTERÍSTICAS PRINCIPAIS

- Interface visual intuitiva para design de telas, sem programação gráfica
- Suporte a resoluções de 240×240 a 1920×1080, incluindo 800×480, 800×600, 1024×600, 1024×768, 1280×720, 1280×800 e 1366×768
- Biblioteca de controles pré-configurados (toque e exibição) para IHM
- Comunicação serial com microcontroladores (UART TTL 3,3 V)
- Módulos disponíveis com diversas interfaces de painel: RGB, LVDS, MIPI
- Ferramentas integradas para conversão de imagens, geração de ICL e de fontes
- Sistema de endereçamento VP (Variable Pointer) e SP (Description Pointer) para controle de dados e de atributos
- Compatibilidade com Arduino, STM32, ESP32, PIC e outros MCUs
- Plataforma de desenvolvimento secundário embarcada (DWIN OS / 8051 OS core) e interface Modbus RTU mestre/escravo, nos modelos com UART4 acessível

## APLICAÇÕES TÍPICAS

- Controle de automação industrial
- Painéis de monitoramento e supervisão
- Sistemas embarcados para IoT
- Controle de máquinas e equipamentos
- Sistemas de smart home e building automation
- Equipamentos médicos e hospitalares
- Instrumentação e medição

## 2. CONFIGURAÇÃO INICIAL

### 2.1. DOWNLOAD E INSTALAÇÃO

Para começar, você precisará de dois componentes:

- **DGUS Software:** editor visual para criação de interfaces. Para módulos com kernel DGUS2 V4.0 ou superior é obrigatório usar o DGUS V7.623 ou superior, caso contrário os arquivos ICL gerados não serão reconhecidos.
- **(Opcional) Driver USB-Serial:** necessário apenas para download por porta serial via adaptador. O download por cartão SD não exige driver.

Não é necessário instalar o software DGUS. Após a extração, o arquivo DGUS .exe é executado diretamente da pasta extraída.

### 2.2. CONFIGURAÇÃO DE IDIOMA

1. Execute o arquivo DGUS .exe
2. O software pode iniciar com idioma em mandarim
3. Acesse o menu **Settings**, localizado na penúltima aba da barra superior
4. Na caixa **Language**, abra o seletor e altere o idioma para **English**

### 2.3. ESTRUTURA DE PROJETO

| PARÂMETRO        | DESCRIÇÃO                                                       | EXEMPLO                   |
|------------------|-----------------------------------------------------------------|---------------------------|
| Resolução        | Dimensões do display em pixels – deve ser idêntica à do módulo  | 800 × 480                 |
| Nome do Projeto  | Identificador do projeto                                        | SmartHome_V1              |
| Pasta do Projeto | Diretório de trabalho; o DGUS cria dentro dele a pasta DWIN_SET | C:/Projetos/AGT/SmartHome |

A pasta DWIN\_SET gerada pelo DGUS é exatamente o conteúdo que será gravado no display. Tudo o que o módulo precisa (arquivos .BIN, .ICL, fontes e .CFG) fica nela, em um único nível – subpastas dentro de DWIN\_SET não são lidas pelo display.

## 3. DESENVOLVIMENTO DA INTERFACE

### 3.1. ORGANIZAÇÃO DE ATIVOS E IDS DE ARQUIVO

Os números 23, 32 e 48 que aparecem nos projetos DGUS não são pastas: são **IDs de posição na memória Flash**, indicados pelo prefixo numérico do nome do arquivo. A Flash de 16 MB é dividida em 64 subespaços fixos de 256 KB, com ID de 0 a 63, e o display carrega cada arquivo na posição indicada por seu número.

| FAIXA DE ID | CONTEÚDO                   | FORMATO                   | USO                                                                                                                                                                      |
|-------------|----------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00          | Fonte ASCII 0#             | 0*.HZK                    | Posição padrão da fonte ASCII: é o valor default do campo <i>Lib_ID</i> dos controles de texto. Gere o arquivo com o gerador de fonte 0# do DGUS e coloque-o no DWIN_SET |
| 01-11       | Fontes                     | .BIN / .HZK / .DZK / .GTF | Livres para fontes do usuário: GBK, GB2312, UNICODE e fontes codificadas em 8 bits                                                                                       |
| 12          | Método de entrada          | 12*.BIN                   | Posição fixa, reservada pelo kernel e indisponível para fontes                                                                                                           |
| 13          | Controles de toque         | 13*.BIN                   | Posição fixa, gerado pelo DGUS – limite de 256 KB                                                                                                                        |
| 14          | Controles de exibição      | 14*.BIN                   | Posição fixa, gerado pelo DGUS – limite de 256 KB                                                                                                                        |
| 15-21       | Fontes                     | .BIN / .HZK / .DZK / .GTF | Livres para fontes, mas a partir de 16 o espaço é compartilhado com os arquivos .ICL                                                                                     |
| 22          | Inicialização de variáveis | 22*.BIN                   | Posição fixa – carrega o conteúdo de 0x2000-0xFFFF e inicializa o espaço 0x1000-0xFFFF                                                                                   |
| 23-47       | Fontes                     | .BIN / .HZK / .DZK / .GTF | Livres para fontes, também compartilhados com os arquivos .ICL                                                                                                           |
| 16-63       | Imagens e ícones           | .ICL                      | Backgrounds das páginas (ex.: 32.ICL) e bibliotecas de ícones (ex.: 48.ICL)                                                                                              |

**IMPORTANTE** – o espaço de fontes vai de 00 a 47, mas as posições 12, 13, 14 e 22 são fixas para os arquivos de configuração do DGUS e não aceitam fonte; sobram 00–11, 15–21 e 23–47. Além disso, as faixas de fonte (00–47) e de imagem (16–63) se sobrepõem, e um arquivo pode ocupar vários subespaços consecutivos: o tamanho em KB dividido por 256, arredondado para cima (um arquivo de 1,28 MB ocupa 6). Se a fonte no ID 23 ocupa 10 subespaços, ela vai até o ID 32 e a biblioteca de imagens precisa começar em 33, não em 32. Conflito de ID é a causa mais comum de página preta com o restante do projeto funcionando.

Um projeto novo começa sem nenhuma fonte: a primeira tarefa é gerar a fonte ASCII no gerador de fonte 0# do DGUS e colocá-la no DWIN\_SET junto com os demais arquivos. Cada controle escolhe a fonte pelo campo *Lib\_ID*, que é a posição do arquivo na Flash – a fonte ASCII pode ficar em qualquer ID livre, e a posição 00 não é reservada. O que a torna a escolha natural é a semântica de padrão que o kernel dá a ela:

- **Lib\_ID = 0x00 é o valor default** dos controles de entrada de texto e de teclado. Todo controle criado sem alterar esse campo vai chamar a fonte da posição 00
- **Na entrada de texto com Lib\_ID = 0x00**, a altura da fonte precisa ser o dobro da largura
- **Na exibição de dados HEX**, se o Lib\_ID for diferente de 0 a fonte precisa usar codificação de 8 bits – o kernel pressupõe ASCII na posição 00 e codificação de 8 bits nas outras

Colocar outra fonte na posição 00 é permitido pelo formato, mas quebra essas suposições: controles deixados no Lib\_ID padrão chamam a fonte errada e a regra de codificação passa a valer ao contrário do esperado. Mantenha a ASCII em 00 e distribua as outras fontes nas posições livres acima dela.

### 3.2. CONVERSÃO DE IMAGENS E GERAÇÃO DE ICL

1. Clique no botão **Picture Conversion** na guia *DGUS config tool*, sob a aba *Welcome*
2. Configure a resolução de destino (ex.: 800 × 480 pixels), idêntica à do display
3. Adicione as imagens das telas do projeto (JPEG, BMP ou PNG, em 24 bits)
4. Clique em **Image Conversion** e selecione a pasta de destino
5. As imagens serão redimensionadas e convertidas automaticamente

Os arquivos ICL (Icon Library) são bibliotecas compiladas de imagens usadas pelo display para renderização rápida – o ASIC não lê arquivos JPEG soltos. Para criar arquivos ICL:

1. Abra o **DWIN ICL generation** na guia *DGUS config tool*, sob a aba *Welcome*
2. Selecione as imagens de fundo e gere 32 . ICL; as imagens são numeradas em ordem (00\_boot, 01\_funcao, 02\_menu...) e a imagem 0 é exibida na inicialização
3. Repita o processo para a biblioteca de ícones e gere 48 . ICL
4. Salve todos os arquivos ICL na pasta DWIN\_SET do projeto
5. Registre a posição da biblioteca de fundos no arquivo CFG. Abra o **Config Generator** na guia *DGUS config tool*, sob a aba *Welcome*, informe o número da posição em *ICL Address* em decimal (32, para 32 . ICL) e salve com **New CFG** como T5LCFG.CFG dentro do DWIN\_SET. Se o CFG estiver vazio, a tela fica preta com o backlight fraco.

A mesma janela reúne os demais parâmetros de hardware do módulo – *File 22* (carregar ou não a inicialização de variáveis), *WAE Address*, *Baud Rate*, *CRC*, som do toque, direção da imagem na inicialização, buzzer e brilho de backlight. Use **Open CFG** para partir de um arquivo existente e **Save to CFG** para regravá-lo. Deixe desmarcadas as caixas de *Touch Panel Configuration* e de *Size-resolution*: elas sobrescrevem parâmetros ajustados em fábrica.

| LIMITE                                  | T5L1                                                                                             | T5L2     |
|-----------------------------------------|--------------------------------------------------------------------------------------------------|----------|
| Imagem individual dentro do ICL         | ≤ 256 KB                                                                                         | ≤ 768 KB |
| Arquivo JPG individual                  | ≤ 248 KB                                                                                         | ≤ 760 KB |
| Material de origem (antes da conversão) | Sem limite – o DGUS recomprime; reduza o percentual de qualidade se o gerador avisar que excedeu |          |
| Flash total disponível para UI          | 16 MB nos módulos de chip único; 32 MB (2×16 MB NOR) nos módulos da família 2K                   |          |

### 3.3. MÓDULOS PRINCIPAIS

Os controles do DGUS dividem-se em dois grupos: **controles de toque** (gravados em 13.bin) e **controles de exibição** (gravados em 14.bin). Os mais utilizados são:

#### a) Basic Touch – toque básico

- **Função:** cria áreas sensíveis ao toque na tela
- **Ação:** troca de página ao ser pressionado, com página opcional de efeito de pressionamento
- **Configuração:** área retangular definida pelos cantos superior-esquerdo (Xs, Ys) e inferior-direito (Xe, Ye), mais a página de destino
- **Uso típico:** botões de menu, navegação, seleção de opções

Área: (100, 200) a (250, 250)  
 Página destino: 02 (Tela de Temperatura)  
 Efeito: ao tocar, muda para a página 02

O toque básico é o único controle de toque que não pode ser habilitado ou desabilitado por comando serial, e ele não conta na numeração usada pelo comando 0x00B0 – descontar os toques básicos anteriores ao calcular o ID de ordenação.

#### b) Data Variables – exibição numérica

É o controle padrão para mostrar valores dinâmicos recebidos do microcontrolador (temperatura, umidade, velocidade). Desenha o número com uma fonte da biblioteca, no formato especificado.

| PARÂMETRO                       | DESCRIÇÃO                                                                  | EXEMPLO                 |
|---------------------------------|----------------------------------------------------------------------------|-------------------------|
| VP                              | Endereço da variável                                                       | 0x5000<br>(temperatura) |
| Lib_ID / Font size              | Posição da fonte ASCII e largura do caractere em pontos                    | 0 / 32                  |
| Variable data type              | 0x00 int16, 0x01 int32, 0x05 uint16, 0x07 float, 0x08 double, entre outros | 0x00                    |
| Integer digits / Decimal places | Dígitos inteiros e decimais; a soma não pode passar de 20                  | 2 / 1 → 25.5            |
| Len_unit / String_unit          | Unidade fixa em ASCII, até 11 caracteres (0x00 = sem unidade)              | 2 / "°C"                |
| Alignment                       | 0x00 esquerda, 0x01 direita, 0x02 centralizado                             | 0x02                    |

### c) Artistic Variables – números desenhados com ícones

Substitui a fonte por uma biblioteca de ícones, para números de aparência customizada (display de sete segmentos, tipografia própria). A biblioteca precisa seguir obrigatoriamente a ordem de indexação [0 1 2 3 4 5 6 7 8 9 - .] a partir do ícone informado em ICON0.

| PARÂMETRO                       | DESCRIÇÃO                                               | EXEMPLO |
|---------------------------------|---------------------------------------------------------|---------|
| VP                              | Endereço da variável                                    | 0x5000  |
| ICON0 / ICON Lib                | Índice do ícone do dígito 0 e ID da biblioteca na Flash | 0 / 48  |
| Integer_Digits / Decimal_Digits | Dígitos inteiros e decimais                             | 2 / 1   |
| VP_Mode                         | Tamanho do dado em memória (int16, int32, float...)     | 0x00    |
| ICON_Mode                       | 0x00 fundo transparente, demais valores fundo opaco     | 0x00    |

### d) Variables Icon – ícone por faixa de valor

Exibe um ícone diferente conforme o valor de uma variável, mapeando linearmente a faixa V\_Min–V\_Max sobre a faixa Icon\_Min–Icon\_Max. Ideal para botões ON/OFF, indicadores de estado, ponteiros e barras.

```

Controle de lâmpada
VP          : 0x5100
Icon_Lib    : 48 (biblioteca 48.ICL)
V_Min / V_Max      : 0 / 1
Icon_Min / Icon_Max: 0 (lamp_off) / 1 (lamp_on)
Mode        : 0x00 = transparente

```

Valores abaixo de V\_Min ou acima de V\_Max não exibem ícone algum – é um comportamento útil para ocultar indicadores, e uma armadilha quando os limites estão mal ajustados.

### e) Incremental Adjustment – ajuste incremental

Controle de toque que incrementa, decrementa ou alterna o valor de uma variável. Combinado com o Variables Icon, cria botões interativos completos. Código de toque: 0xFE02.

| PARÂMETRO     | FUNÇÃO                                                                          | VALOR TÍPICO (TOGGLE)              |
|---------------|---------------------------------------------------------------------------------|------------------------------------|
| VP            | Endereço da variável                                                            | 0x5100 (o mesmo do Variables Icon) |
| VP_Mode       | 0x00 word inteira, 0x01 byte alto, 0x02 byte baixo, 0x10–0x1F um bit específico | 0x00                               |
| Adj_Mode      | 0x00 = decrementa; qualquer outro valor = incrementa                            | 0x01                               |
| Adj_Step      | Passo do ajuste, 0x0000–0x7FFF                                                  | 1                                  |
| V_Min / V_Max | Limites mínimo e máximo aceitos                                                 | 0 / 1                              |
| Return_Mode   | 0x00 = para no limite; demais valores = circula ao chegar ao limite             | 0x01 (circula)                     |
| Key_Mode      | 0x00 = repete enquanto pressionado; 0x01 = uma vez por toque                    | 0x01                               |
| Pic_Next      | Página de destino; 0xFF** = não troca de página                                 | 0xFF00                             |

**IMPORTANTE** – o Incremental Adjustment e o Variables Icon devem compartilhar o mesmo endereço VP para funcionarem de forma sincronizada. Para que a mudança suba para o MCU, é preciso habilitar o upload de dados no arquivo CFG.

## 4. COMUNICAÇÃO SERIAL

A comunicação entre o TFT Módulo Serial e o microcontrolador ocorre pela UART2 (interface de depuração do DGUS), com protocolo proprietário da DWIN.

### 4.1. PARÂMETROS DA PORTA

| PARÂMETRO         | VALOR               | DESCRIÇÃO                                                                                                                                                                                                                   |
|-------------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Baud rate         | 115200 bps (típico) | Configurável pelo T5LCFG; a faixa aceita chega a 3150–3225600 bps na UART2 e 3150–921600 bps na UART4                                                                                                                       |
| Formato           | 8N1                 | 8 bits de dados, sem paridade, 1 stop bit. Fixo na UART2, que é a porta do protocolo DGUS; nas demais UARTs, usadas em DWIN OS ou pela interface Modbus, o formato é escolhido pelo desenvolvedor entre 8N1, 8E1, 8O1 e 8N2 |
| Controle de fluxo | Nenhum              | —                                                                                                                                                                                                                           |
| Níveis lógicos    | Depende do módulo   | TTL 3,3 V na maioria dos modelos (tolerante a MCUs de 5 V com divisor resistivo na linha RX); RS232 na UART2 e, na UART4, RS485, RS232 ou TTL conforme o modelo, nos módulos 2K – confira a folha de dados antes de ligar   |
| Verificação CRC   | Opcional            | CRC-16 ( $x^{16}+x^{15}+x^2+1$ ), ligado pelo bit 0x05.7 do arquivo CFG                                                                                                                                                     |

### 4.2. SISTEMA DE ENDEREÇOS VP

O espaço de variáveis tem 128 KB, endereçado por palavras (words) de 0x0000 a 0xFFFF – cada endereço corresponde a 2 bytes. A divisão é fixa:

| FAIXA         | USO                                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------------------------------|
| 0x0000–0x0FFF | Interface de variáveis de sistema. Reservada – não use para dados de aplicação                                                 |
| 0x1000–0x4FFF | Livre, <i>exceto</i> se houver curvas em tempo real: com 8 canais ativos esta faixa vira buffer de curvas (2 KWords por canal) |
| 0x5000–0xFFFF | Faixa sempre segura para o usuário                                                                                             |

Cada página suporta até 255 variáveis de exibição, e o DGUS relê 13.bin e 14.bin a cada 20 ms.

Mapa de exemplo usado neste guia:

| ENDEREÇO VP | TIPO     | FUNÇÃO                 | FAIXA  |
|-------------|----------|------------------------|--------|
| 0x5000      | Sensor   | Temperatura (×10)      | 0-1000 |
| 0x5010      | Sensor   | Umidade (×10)          | 0-1000 |
| 0x5020      | Sensor   | Pressão (hPa)          | 0-1200 |
| 0x5100      | Controle | Lâmpada ON/OFF         | 0-1    |
| 0x5200      | Controle | Ventilador ON/OFF      | 0-1    |
| 0x5300      | Controle | Ar-condicionado ON/OFF | 0-1    |
| 0x5400      | Controle | TV ON/OFF              | 0-1    |

### 4.3. ESPAÇO DE DESCRITORES (SP)

O VP guarda o **valor** de uma variável. O SP (*description pointer*) guarda os **atributos** do controle que exibe esse valor – posição, cor, fonte, número de dígitos, tamanho da matriz de pontos. Escrever no VP muda o que o display mostra; escrever no SP muda como ele mostra.

Sem SP, todo atributo fica congelado no que foi definido no editor. Com SP, o MCU altera a aparência em tempo de execução: pintar um valor de vermelho ao passar do limite, mover um campo, ocultar um texto, trocar o corpo da fonte – tudo pelo mesmo comando 0x82 usado para os dados.

Diferenças de comportamento:

|                   | VP                          | SP                                                            |
|-------------------|-----------------------------|---------------------------------------------------------------|
| Conteúdo          | Valor da variável           | Atributos do controle                                         |
| Definido em       | Campo <i>VP</i> do controle | Campo <i>SP</i> do controle                                   |
| Endereçamento     | Endereço direto             | Endereço-base + offset por atributo                           |
| Se não preenchido | Controle não funciona       | 0xFFFF – atributos vêm do arquivo de configuração e são fixos |
| Escrita típica    | A cada leitura de sensor    | Eventual, em mudança de estado                                |

**ATENÇÃO** – VP e SP dividem o mesmo espaço de RAM de 128 KB. Sobreposição entre eles corrompe o display. A recomendação da DWIN é separar as faixas: **VP em 0x1000–0x7FFF, SP em 0x8000–0xFFFF** (com curvas em uso, VP em 0x5000–0x7FFF). Reserve 0x10 endereços por controle do mesmo tipo e 0x30 entre tipos diferentes.

#### a) Como habilitar

No editor, preencha o campo *SP* do controle com um endereço-base da faixa reservada (ex.: 0x9000). A partir daí, cada atributo é alcançado somando um offset em *words* a esse endereço.

#### b) Offsets do Data Variables

Tabela de descrição do controle *Data Variables*, com SP-base em 0x9000:

| OFFSET        | ENDEREÇO | ATRIBUTO                   | FORMATO                                                           |
|---------------|----------|----------------------------|-------------------------------------------------------------------|
| 0x00          | 0x9000   | VP                         | Endereço da variável exibida                                      |
| 0x01–<br>0x02 | 0x9001   | X, Y                       | 2 words: coluna e linha do canto superior esquerdo                |
| 0x03          | 0x9003   | Cor                        | 1 word, RGB565                                                    |
| 0x04          | 0x9004   | Lib_ID / Font size         | Byte alto: posição da fonte. Byte baixo: largura em pontos        |
| 0x05          | 0x9005   | Alignment / Integer digits | Byte alto: 0 esq., 1 dir., 2 centro. Byte baixo: dígitos inteiros |
| 0x06          | 0x9006   | Decimal places / Data type | Byte alto: casas decimais. Byte baixo: tipo do dado               |
| 0x07          | 0x9007   | Len_unit / String_unit     | Byte alto: comprimento da unidade (0 = sem unidade)               |

Cada tipo de controle tem sua própria tabela de offsets, no capítulo 7 do guia de aplicação DGUS II. A estrutura inicial é comum a todos: offset 0x00 é sempre o VP, 0x01 a posição e 0x03 a cor.

### c) Exemplos de comando

Mudar a cor do valor para vermelho (RGB565 0xF800)

Tx: 5A A5 05 82 90 03 F8 00

Mover o campo para a coordenada (100, 100)

Tx: 5A A5 07 82 90 01 00 64 00 64

Trocar o corpo da fonte para 40 pontos, mantendo a fonte na posição 0

Tx: 5A A5 05 82 90 04 00 28

Ocultar a unidade fixa (Len\_unit = 0)

Tx: 5A A5 05 82 90 07 00 00

```
// Alarme visual por SP: temperatura acima de 30 °C fica vermelha
#define SP_TEMPERATURE 0x9000          // SP-base do Data Variables da temperatura
#define SP_OFF_COLOR    0x03           // offset do atributo de cor

void setTempColor(bool alarm) {
    sendToDisplay(SP_TEMPERATURE + SP_OFF_COLOR, alarm ? 0xF800 : 0x0000);
}
```

O mesmo `sendToDisplay()` do §5.2 serve para SP — a instrução 0x82 não distingue as duas faixas. A diferença está apenas no endereço.

**CUIDADO COM SP COMPARTILHADO** – dois controles do mesmo tipo com o mesmo SP só funcionam se um for cópia exata do outro em outra página: mover o controle ou alterar fonte e tamanho de um deles quebra a exibição de ambos.

#### 4.4. ESTRUTURA DO FRAME DE DADOS

Todo quadro tem cinco blocos: cabeçalho (2 bytes), comprimento (1 byte), instrução (1 byte), dados (até 249 bytes) e CRC opcional (2 bytes). O byte de **comprimento conta a instrução, os dados e o CRC** – não conta o cabeçalho nem ele próprio.

##### a) Escrita: MCU → Display (0x82)

5A A5 [LEN] 82 [VP\_H] [VP\_L] [DADO\_H] [DADO\_L]

5A A5    cabeçalho  
LEN      nº de bytes a partir da instrução (0x05 para uma word)  
82        instrução de escrita no espaço de variáveis  
VP\_H/L   endereço da variável  
DADO     valor, sempre em words de 2 bytes

Enviar 25,5 °C para o VP 0x5000 (valor  $\times 10 = 255 = 0x00FF$ )

Tx: 5A A5 05 82 50 00 00 FF

Rx: 5A A5 03 82 4F 4B                      ← resposta "OK" do display

O display responde 4F 4B ("OK") a cada escrita. O firmware precisa consumir essa resposta, ou ela se acumula no buffer de recepção e desalinha a leitura dos quadros de toque.

##### b) Leitura: MCU pergunta, display responde (0x83)

Pedido Tx: 5A A5 04 83 51 00 01      lê 1 word a partir de 0x5100

Resposta Rx: 5A A5 06 83 51 00 01 00 01

5A A5    cabeçalho  
06        comprimento  
83        instrução de leitura  
51 00    VP = 0x5100  
01        quantidade de words lidas  
00 01    valor = 1 (botão ligado)

O comprimento máximo de leitura é 0x7D words por comando.

##### c) Subida espontânea de toque (0x83)

Rx: 5A A5 06 83 10 01 01 00 0A  
      endereço 0x1001, valor de retorno 0x000A

Quando o upload de dados está habilitado no CFG, os controles de toque (exceto o toque básico) enviam esse quadro sozinhos, no mesmo formato da resposta de leitura. Em RS485, que é meio-duplex, prefira ler o valor com 5A A5 04 83 10 01 01 em vez de depender da subida espontânea – e nunca transmita e receba ao mesmo tempo. O valor não é zerado automaticamente após a leitura.

#### 4.5. INTERFACE DE VARIÁVEIS DE SISTEMA

A faixa 0x0000–0x0FFF controla o hardware por escrita de variáveis. Comandos de uso frequente:

| FUNÇÃO              | COMANDO                       | OBSERVAÇÃO                          |
|---------------------|-------------------------------|-------------------------------------|
| Trocar de página    | 5A A5 07 82 00 84 5A 01 00 0A | Vai para a página 10; 0x5A01 é fixo |
| Ler a página atual  | 5A A5 04 83 00 14 01          | Somente leitura                     |
| Brilho do backlight | 5A A5 04 82 00 82 0A          | Faixa 0x00–0x64                     |
| Buzzer              | 5A A5 05 82 00 A0 00 7D       | Unidade de 8 ms – 0x7D = 1 s        |
| Versão de firmware  | 5A A5 04 83 00 0F 01          | Retorna versão GUI e DWIN OS        |
| Reset da CPU        | 5A A5 07 82 00 04 55 AA 5A A5 | Equivale a desligar e religar       |

#### 4.6. PERSISTÊNCIA EM FLASH

O espaço de variáveis é RAM: tudo o que está nele desaparece ao desligar. No próximo arranque, o arquivo 22 reinicializa a região com os valores que você definiu no projeto – sempre os mesmos.

Isso é um problema assim que a aplicação tem algo que o operador ajusta e espera encontrar depois: setpoint, unidade de medida, idioma da interface, constante de calibração, contador de horas. Para esses casos, o módulo traz um banco de dados em Flash no próprio chip, e uma interface de sistema para copiar blocos entre ele e o espaço de variáveis.

A interface fica na posição 0x0008 e ocupa 4 words:

| CAMPO | CONTEÚDO                        | REGRA                             |
|-------|---------------------------------|-----------------------------------|
| D7    | Operação: 0xA5 gravar, 0x5A ler | O módulo zera o campo ao concluir |
| D6:D4 | Endereço no banco em Flash      | Par, de 0x000000 a 0x03FFFE       |
| D3:D2 | Endereço no espaço de variáveis | Par                               |
| D1:D0 | Comprimento, em words           | Par                               |

Gravar duas words que estão em 0x1002 no início do banco:

```
5A A5 0B 82 00 08 A5 00 00 02 10 02 00 02
```

| BYTES    | CAMPO                     | VALOR             |
|----------|---------------------------|-------------------|
| 5A A5    | Cabeçalho                 | Fixo              |
| 0B       | Comprimento               | 11 bytes a seguir |
| 82       | Instrução                 | Escrita           |
| 00 08    | Endereço da interface     | 0x0008            |
| A5       | D7 – operação             | Gravar            |
| 00 00 02 | D6:D4 – endereço no banco | 0x000002          |
| 10 02    | D3:D2 – endereço de VP    | 0x1002            |
| 00 02    | D1:D0 – comprimento       | Duas words        |

Repare que o campo de operação é **um byte só**, o byte alto da primeira word: o 00 seguinte já pertence ao endereço no banco, que tem três bytes. As fronteiras dos campos não coincidem com as das words.

Ler de volta, do banco para 0x2000:

```
5A A5 0B 82 00 08 5A 00 00 02 20 00 00 02
```

**Espera o campo de operação zerar antes da próxima.** O módulo limpa D7 quando termina. Disparar outra operação antes disso causa desordem lógica – o manual é explícito. Quem faz isso por programa no módulo deve reler D7 e só prosseguir quando estiver zerado. Pela serial, um atraso adequado entre comandos resolve.

Três cuidados de projeto. **Todos os três endereços e o comprimento são pares** – o barramento trabalha em words. **Decida no início quais variáveis são persistentes** e reserve para elas um bloco contíguo do espaço de variáveis: gravar e ler um bloco só é uma operação, enquanto valores espalhados viram várias. E **grave quando o valor muda, não no laço**: gravação em Flash é lenta comparada com a RAM e o número de ciclos não é infinito. O padrão que funciona é ler o bloco uma vez no arranque e gravá-lo quando o operador confirma um ajuste.

## 5. PROGRAMAÇÃO ARDUINO

### 5.1. ESTRUTURA DO CÓDIGO

- **Includes:** bibliotecas necessárias (Wire.h, bibliotecas de sensores)
- **Defines:** pinos, endereços VP, constantes
- **Variáveis globais:** buffers de comunicação, estados dos sensores
- **setup():** inicialização de serial, sensores e relés
- **loop():** leitura de sensores, envio de dados, processamento de comandos

```
#include <Wire.h>
#include <Adafruit_BME280.h>

#define RELAY_LIGHT 2
#define RELAY_FAN    3
#define RELAY_AC     4
#define RELAY_TV     5

#define VP_TEMPERATURE 0x5000
#define VP_HUMIDITY    0x5010
#define VP_PRESSURE    0x5020
#define VP_LIGHT       0x5100
#define VP_FAN         0x5200
#define VP_AC          0x5300
#define VP_TV          0x5400

Adafruit_BME280 bme;

void setup() {
  Serial.begin(115200);          // 8N1, padrão do ASIC
  if (!bme.begin(0x76)) {
    // sem Serial.println aqui: a porta é do display, não do monitor
  }
  pinMode(RELAY_LIGHT, OUTPUT);
  pinMode(RELAY_FAN,   OUTPUT);
  pinMode(RELAY_AC,    OUTPUT);
  pinMode(RELAY_TV,    OUTPUT);
}
```

**ATENÇÃO** — em placas de UART única (Uno, Nano, Pro Mini) a mesma porta serve ao display e ao gravador. Mensagens de texto enviadas com `Serial.println` entram no display como quadros inválidos. Em Mega, Due, ESP32 ou STM32, dedique uma UART de hardware ao display (`Serial1`) e mantenha a `Serial` para depuração.

## 5.2. ENVIO DE DADOS PARA O DISPLAY

```
void sendToDisplay(uint16_t vp, uint16_t data) {
    uint8_t frame[8];
    frame[0] = 0x5A;                // cabeçalho
    frame[1] = 0xA5;
    frame[2] = 0x05;                // instrução + VP + 1 word
    frame[3] = 0x82;                // escrita
    frame[4] = (vp >> 8) & 0xFF;
    frame[5] = vp & 0xFF;
    frame[6] = (data >> 8) & 0xFF;
    frame[7] = data & 0xFF;
    Serial.write(frame, 8);
}

// Uso: uma casa decimal → multiplique por 10
float t = bme.readTemperature();
sendToDisplay(VP_TEMPERATURE, (uint16_t)(t * 10));
```

## 5.3. RECEBIMENTO DE DADOS DO DISPLAY

O receptor precisa sincronizar pelo cabeçalho, respeitar o byte de comprimento, descartar as respostas "OK" das escritas e ler o dado como *word*:

```

// Subida de toque: 5A A5 06 83 VP_H VP_L 01 D_H D_L   (9 bytes)
// Resposta 0x82 : 5A A5 03 82 4F 4B                  (6 bytes)
void receiveFromDisplay() {
    static uint8_t buf[16];
    static uint8_t n = 0;

    while (Serial.available()) {
        uint8_t b = Serial.read();

        if (n == 0 && b != 0x5A) continue;           // procura cabeçalho
        if (n == 1 && b != 0xA5) { n = 0; continue; }
        buf[n++] = b;
        if (n < 4) continue;

        uint8_t len = buf[2];                         // instrução + dados (+CRC)
        if (len > sizeof(buf) - 3) { n = 0; continue; } // quadro longo: descarta
        if (n < len + 3) continue;                   // ainda incompleto
        n = 0;

        if (buf[3] != 0x83) continue;                // ignora o "OK" do 0x82

        uint16_t vp    = ((uint16_t)buf[4] << 8) | buf[5];
        uint16_t value = ((uint16_t)buf[7] << 8) | buf[8]; // dado é WORD

        switch (vp) {
            case VP_LIGHT: digitalWrite(RELAY_LIGHT, value ? HIGH : LOW); break;
            case VP_FAN:   digitalWrite(RELAY_FAN,   value ? HIGH : LOW); break;
            case VP_AC:    digitalWrite(RELAY_AC,    value ? HIGH : LOW); break;
            case VP_TV:    digitalWrite(RELAY_TV,    value ? HIGH : LOW); break;
        }
    }
}

```

## 5.4. LOOP PRINCIPAL

```

void loop() {
    static unsigned long sc-camel-last-update = 0;
    unsigned long now = millis();

    if (now - lastUpdate >= 1000) {                // sensores a cada 1 s
        sc-camel-last-update = now;
        sendToDisplay(VP_TEMPERATURE, (uint16_t)(bme.readTemperature() * 10));
        sendToDisplay(VP_HUMIDITY,    (uint16_t)(bme.readHumidity()    * 10));
        sendToDisplay(VP_PRESSURE,    (uint16_t)(bme.readPressure() / 100.0F));
    }

    receiveFromDisplay();                          // sempre, sem bloquear
}

```

**DICA** — use `millis()` em vez de `delay()`. Como o DGUS atualiza a tela a cada 20 ms, enviar dados mais rápido que isso só consome banda: 500-1000 ms é o intervalo adequado para grandezas de processo.

## 6. UPLOAD E DEPLOY

### 6.1. GERAÇÃO DOS ARQUIVOS DO PROJETO

No DGUS, clique em **File** → **Save** e em seguida em **Generate**. O software escreve na pasta **DWIN\_SET**:

| ARQUIVO                       | DESCRIÇÃO                                                                          | LIMITE               |
|-------------------------------|------------------------------------------------------------------------------------|----------------------|
| 13*.BIN                       | Configuração dos controles de toque                                                | 256 KB               |
| 14*.BIN                       | Configuração dos controles de exibição (formato DGUS II)                           | 256 KB               |
| 22*.BIN                       | Inicialização das variáveis (carrega 0x2000–0xFFFF)                                | 256 KB               |
| *.ICL                         | Bibliotecas de imagens e ícones (ex.: 32, 48)                                      | Vários MB            |
| *.BIN / .HZK / .DZK /<br>.GTF | Fontes — IDs 00–11, 15–21 e 23–47 (12, 13, 14 e 22 são posições fixas)             | 256 KB por subespaço |
| T5LCFG*.CFG                   | Configuração de hardware: CRC, baud rate, som do toque, modo de upload, orientação | 1 KB                 |
| T5*.CRC                       | Verificação CRC32 após o download por SD (opcional)                                | —                    |
| *.WAE                         | Áudio, quando o módulo tiver alto-falante                                          | —                    |

**IMPORTANTE** — não baixe arquivos CFG de outro modelo de display: um CFG incompatível pode deixar o módulo inoperante. Peça o CFG correto ao suporte quando precisar restaurar o padrão de fábrica.

### 6.2. UPLOAD POR CARTÃO SD

É o método mais confiável e o único usado em produção. O cartão precisa ser SD ou SDHC (1–16 GB), formatado em FAT32 com unidade de alocação de 4 KB.

1. Formate o cartão pelo prompt de comando — a formatação pelo botão direito do Windows frequentemente não produz um FAT32 aceito pelo display:

```
format /q g: /fs:fat32 /a:4096
```

onde **g**: é a letra da unidade do cartão no seu PC.

2. Confirme que todos os arquivos gerados (.BIN, .ICL, fontes, .CFG) estão dentro da pasta **DWIN\_SET**, sem subpastas
3. Copie a pasta **DWIN\_SET** para a raiz do cartão
4. Desligue o display completamente — o cartão SD não suporta troca a quente
5. Insira o cartão SD no slot do display
6. Ligue o display; ele detecta os arquivos automaticamente e mostra a tela azul de progresso

7. Mantenha a alimentação estável durante todo o processo: uma queda de energia no meio da gravação pode inutilizar a Flash
8. Aguarde a mensagem de conclusão `SD card process END!`
9. Desligue o display, remova o cartão e ligue novamente – o projeto estará carregado

### 6.3. UPLOAD POR PORTA SERIAL (UART DOWNLOAD TOOL)

1. Conecte o display ao PC por um adaptador USB-Serial ou pelo baixador ED4
2. Abra o **UART Download Tool** na guia *DGUS config tool*, sob a aba *Welcome*
3. Selecione a porta COM correta
4. Configure o baud rate para 115200 (o baixador ED4 opera em porta virtual a 8 Mbps)
5. Clique em **Select Folder** ou **Select Files** e adicione os arquivos do projeto
6. Clique em **Start Downloading** e aguarde a barra chegar a 100%
7. O display reinicia automaticamente com o novo projeto

| MÉTODO     | VANTAGENS                                                                                                 | DESVANTAGENS                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Cartão SD  | Mais confiável; não precisa de driver; funciona em qualquer módulo; único método homologado para produção | Mais trabalhoso; exige cartão formatado corretamente; vários passos com desligamento |
| UART / ED4 | Rápido e prático; ideal para iterar durante o desenvolvimento; permite atualizar arquivos isolados        | Requer driver e porta COM; sensível a cabos e ruído                                  |

## 7. INTERFACE MODBUS RTU

Além do protocolo serial próprio, o ASIC implementa Modbus RTU nativo, como mestre ou escravo, pela UART4. É o caminho para integrar o display a inversores, CLPs e transmissores sem escrever firmware de MCU.

**Duas condições antes de começar.** Nem todo módulo expõe a UART4: confirme com o suporte da AGT Displays que o modelo a traz acessível, e em que nível elétrico – RS485, RS232 ou TTL, conforme o modelo. E o fabricante publica mais de uma implementação de Modbus para o ASIC, com endereços e formatos diferentes. Os parâmetros abaixo são da implementação da plataforma de desenvolvimento secundário, que exige núcleo OS V21 ou superior; confirme qual núcleo está gravado antes de preencher qualquer tabela.

Os parâmetros ficam nas variáveis 0xE000–0xE00F, normalmente inicializadas pelo arquivo 22.bin (faixa 0x1C000–0x1FFFF):

| ENDEREÇO          | DEFINIÇÃO                                 | VALOR                                                                        |
|-------------------|-------------------------------------------|------------------------------------------------------------------------------|
| 0xE000            | Habilita o Modbus                         | 0x5AA5 para executar                                                         |
| 0xE001            | Mestre ou escravo                         | 0x0000 = mestre; outro valor = escravo                                       |
| 0xE002            | Baud rate                                 | Em kbps com uma casa fixa – 115200 bps = 0x0480; 0x0000 volta ao padrão      |
| 0xE003            | Formato serial                            | 0x0000 = 8N1, 0x0001 = 8E1, 0x0002 = ímpar, 0x0003 = 8N2                     |
| 0xE004            | Endereço do escravo                       | Padrão 0x005A                                                                |
| 0xE008–<br>0xE00F | Primeira linha de instrução<br>(16 bytes) | Dispositivo, função, comprimento, tempo de ciclo, modo, VP e endereço remoto |

As linhas de instrução do mestre seguem a partir de 0x1C010 no 22.bin, com até 1023 linhas. Cada linha lê ou escreve no máximo 180 bytes, e a faixa de variáveis acessível pelo Modbus é 0x0000–0xDFFF. O tempo de ciclo padrão é 60 ms; a 115200 bps, 20 ms é o valor recomendado.

Funções suportadas como mestre: 0x01 e 0x02 (leitura de bobinas e entradas), 0x03 e 0x04 (leitura de registradores), 0x05 e 0x06 (escrita individual), 0x07 (status de exceção), 0x0F e 0x10 (escrita múltipla). Como escravo, o mestre externo lê e escreve apenas a faixa 0x0000–0xDFFF do espaço de variáveis. Atenção a uma diferença de convenção: variáveis de bit no Modbus são definidas por LSB, e no módulo por MSB.

## 8. TROUBLESHOOTING

### Display não atualiza após o upload

Causa: arquivos fora do lugar. Verifique se DWIN\_SET está na raiz do cartão, sem subpastas, e contém todos os .BIN, .ICL e o .CFG.

### Tela azul mostrando 0 arquivos, ou cartão não reconhecido

Causa: formatação inadequada. Reformate por prompt com `format /q g: /fs:fat32 /a:4096`.

### Tela preta com backlight fraco

Causa: o CFG não aponta a biblioteca de fundos. Escreva a posição do ICL de imagens no endereço 0x08 do T5LCFG (0x20 para 32.ICL).

### Uma página aparece preta e as demais funcionam

Causa: aquela imagem excede o limite por subespaço (256 KB no T5L1, 768 KB no T5L2), ou há conflito de ID entre fonte e imagem. Reduza a qualidade na geração do ICL ou desloque o ID da biblioteca.

### Valores não aparecem no display

Causa: VP ou tipo de dado divergente. Confirme que o endereço no código coincide com o do controle e que o tipo (int16, int32, float) é o mesmo dos dois lados.

### Ícone some em vez de mudar

Causa: o valor saiu da faixa V\_Min–V\_Max do Variables Icon, que não desenha nada fora dela. Ajuste os limites ou o passo do ajuste incremental.

### Botões não respondem ao toque

Causa: controle não configurado, área pequena demais ou página errada. Aumente a área sensível e confirme o Pic\_ID do controle.

### MCU não se comunica com o display

Causa: baud rate divergente, TX/RX invertidos ou GND ausente. Confirme 115200 8N1 nos dois lados, ligue TX do MCU ao RX do display e sempre una os terras, mesmo com fontes separadas.

**O MCU recebe lixo ou perde o sincronismo**

Causa: a resposta 5A A5 03 82 4F 4B de cada escrita não está sendo consumida, ou o CRC-16 está ligado no CFG e o firmware não o trata. Use o parser da seção 5.3.

**Toque não sobe para o MCU**

Causa: upload de dados desabilitado, ou o controle é um toque básico (que nunca sobe valor). O envio exige duas coisas: a opção *Data auto-uploading* marcada no controle e o bit .4 do byte 0x05 do CFG em 1. O manual sugere 0x38 nesse byte – arquivo 22 carregado, envio ligado e som do toque ligado. Se o controle for básico, troque por Return Key Code / Incremental Adjustment.

**Display mostra caracteres estranhos**

Causa: fonte ausente ou posição errada. Confirme que a fonte ASCII, gerada no gerador de fonte 0# do DGUS, está no DWIN\_SET, e confira o Lib\_ID do controle – ele aponta para a posição da fonte na Flash. Verifique também o tamanho em pontos (lattice) configurado no controle: ele precisa corresponder a um dos tamanhos presentes no arquivo de fonte gerado, senão o texto sai deformado ou não aparece.

**Ferramenta de download não encontra a porta COM**

Causa: driver USB-Serial ausente. Instale o driver do adaptador em uso e reinicie o PC; para baixar sem driver, use o cartão SD.

**DEPURAÇÃO** – use um terminal serial (SSCOM ou equivalente) em modo hexadecimal, ligado no lugar do MCU, para escrever variáveis à mão e observar os quadros de toque. É a forma mais rápida de separar um problema de projeto DGUS de um problema de firmware.

## 9. APLICAÇÕES PRÁTICAS

### AUTOMAÇÃO RESIDENCIAL

Controle centralizado de iluminação, climatização, cortinas e segurança.

- Monitoramento de temperatura, umidade e qualidade do ar
- Controle de até 16 relés
- Agendamento de eventos e cenas
- Feedback visual imediato

### MÁQUINAS CNC

Interface para corte, fresagem e impressão 3D.

- Coordenadas X, Y, Z em tempo real
- Controle de velocidade e avanço
- Temperatura de motores
- Botões de emergência e pausa

### IRRIGAÇÃO INTELIGENTE

Controle automatizado de irrigação agrícola.

- Sensores de umidade do solo
- Válvulas solenoides
- Programação de horários e ciclos
- Registro de consumo de água

### PAINEL INDUSTRIAL

Supervisão de processos em tempo real, com Modbus direto nos instrumentos, em modelos com UART4 acessível.

- Múltiplos sensores de temperatura e pressão
- Alarmes visuais e sonoros
- Curvas de tendência em tempo real
- Controle de bombas e válvulas

### CONTROLE DE ACESSO

Gerenciamento de entrada e saída com biometria ou RFID.

- Interface para cadastro de usuários com entrada de texto
- Registro de logs de acesso
- Controle de fechaduras elétricas
- Integração com leitor biométrico

## 10. ESPECIFICAÇÕES TÉCNICAS

Características do ASIC, comuns a todos os módulos DGUS II:

| ESPECIFICAÇÃO                   | DETALHES                                                                                                                                                                                     |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Processador                     | ASIC T5L, dual-core 8051 (núcleo GUI + núcleo OS), 1T, até 250 MHz; versões de série operam a 200 MHz                                                                                        |
| Resolução por ASIC              | 800×600 (T5L1) / 1366×768 (T5L2), 24 bits de cor — teto de um único chip                                                                                                                     |
| Resolução máxima da linha       | 1920×1080, nos módulos da família 2K: um módulo central baseado no ASIC T5L2 rodando DGUS II, com exibição em 0°, 90°, 180° e 270° — ou seja, também 1080×1920 em retrato. Veja a seção 10.1 |
| Flash de dados                  | 16 MB SPI padrão (32 MB, em 2×16 MB NOR, nos módulos 2K); expansível a 64 MB NOR ou 48 MB NOR + 512 MB NAND; subespaços de 256 KB                                                            |
| Flash de código                 | 1 MB on-chip com criptografia proprietária                                                                                                                                                   |
| Base de dados NOR               | IDs 0–79, 2 KWords cada, para dados de usuário ou bibliotecas DWIN OS                                                                                                                        |
| RAM de variáveis                | 128 KB, endereçada em words de 0x0000 a 0xFFFF                                                                                                                                               |
| Ciclo de atualização            | 20 ms; até 255 variáveis de exibição por página                                                                                                                                              |
| Aceleração gráfica              | 2D por hardware; decodificação JPEG até 200 fps                                                                                                                                              |
| Interface de comunicação        | UART TTL 3,3 V (UART2 para DGUS); 4 UARTs, 1 CAN, 22 IO, 7 ADs de 12 bits, 2 PWM no núcleo OS                                                                                                |
| Baud rate                       | Configurável pelo CFG; padrão 115200 bps, formato 8N1 fixo na UART2                                                                                                                          |
| Touchscreen                     | Resistivo ou capacitivo, até 400 Hz, sensibilidade ajustável (até 6 mm de vidro temperado)                                                                                                   |
| Áudio                           | Amplificador digital PWM de 15 bits, 32 ksp/s, para alto-falante direto                                                                                                                      |
| Temperatura de operação do ASIC | –40 °C a +85 °C (versão customizada: –55 °C a +105 °C)                                                                                                                                       |
| Interfaces de painel            | RGB, LVDS, MIPI, conforme o modelo do módulo                                                                                                                                                 |

Tensão de alimentação, consumo, faixa de temperatura do módulo montado, tamanho de tela, vida útil do backlight e forma de montagem variam por modelo — consulte a folha de dados do módulo específico.

## 10.1. O QUE A LINHA DE MÓDULOS OFERECE

Todos os módulos rodam a mesma plataforma DGUS II descrita neste guia – o fluxo de trabalho, os controles e o protocolo serial são idênticos de um extremo ao outro da linha. O que varia é o hardware ao redor do ASIC. Conhecer essas faixas ajuda a especificar o módulo certo antes de começar o projeto de interface.

### Duas construções: COB e COF

| ASPECTO             | COB (CHIP ON BOARD)                                                                         | COF (CHIP ON FILM)                                      |
|---------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------|
| Montagem            | Circuito principal em placa própria atrás do módulo                                         | Circuito principal inteiro fixado no FPC do próprio LCM |
| Perfil              | Mais espesso; comporta conectores, RTC, buzzer, alto-falante                                | Fino e leve, custo menor e produção mais simples        |
| Faixa de tamanhos   | 1,3" a 21,5"                                                                                | 1,3" a 10,1"                                            |
| Interfaces típicas  | TTL, RS232, RS485 e CAN, conforme o grau                                                    | Predominantemente TTL; RS232 em alguns modelos          |
| Flash de UI         | 16 MB, chegando a 32 MB nos módulos de 1920×1080                                            | 8 MB ou 16 MB                                           |
| Uso natural         | Painéis industriais, equipamentos médicos, ambientes severos                                | Produtos compactos e sensíveis a espessura e custo      |
| Download do projeto | Cartão SD e porta serial; pontos para módulo Wi-Fi ou de download por USB em parte da linha | Porta serial, conforme o conector disponível no FPC     |

### Graus de aplicação

O grau determina faixa de temperatura, brilho, tensão de alimentação e conjunto de interfaces. É a primeira decisão de especificação – informe a aplicação pretendida e confirme o grau na folha de dados do módulo:

| GRAU            | CARACTERÍSTICAS OBSERVADAS NA LINHA                                            |
|-----------------|--------------------------------------------------------------------------------|
| Consumo         | Alimentação de 6–15 V, interface única, sem RTC                                |
| Comercial       | –20 a +70 °C ou –10 a +60 °C; TTL/RS232; opção sem RTC                         |
| Industrial      | Alimentação ampla de 9–36 V, RTC, TTL/RS232 duplo                              |
| Médico          | RS232 duplo + RS485, RTC, alimentação ampla                                    |
| Automotivo      | Grau dedicado a aplicação embarcada em veículo                                 |
| Ambiente severo | –40 a +85 °C, brilho de até 1000 nit para leitura sob sol, RS232 + RS485 + CAN |

## Faixas disponíveis na linha

| CARACTERÍSTICA          | FAIXA                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Tamanho de tela         | 1,3" a 21,5" em catálogo padrão, com customização anunciada até 32"                                                          |
| Formato                 | Retangular, quadrado e redondo                                                                                               |
| Proporção               | 1:1, 4:3, 16:9 e formatos em faixa: 4:1, 8:3, 16:5, 1:3, 20:9                                                                |
| Resolução               | 240×240 a 1920×1080, em orientação paisagem ou retrato                                                                       |
| Cor                     | 65K (16 bits), 262K (18 bits) ou 16,7 M (24 bits), conforme o modelo                                                         |
| Painel                  | IPS ou TN; IPS com ângulo de visão típico de 85° nos quatro sentidos                                                         |
| Brilho                  | De 50 nit a mais de 1000 nit nas versões para luz solar direta                                                               |
| Toque                   | Sem toque, resistivo de 4 fios, capacitivo (G+G, G+FF ou In-cell); onda acústica de superfície em telas grandes customizadas |
| Alimentação             | 3,3–5,5 V, 6–15 V, 9–36 V, 12–36 V ou 15–36 V, conforme o grau                                                               |
| Temperatura de operação | –10 a +60 °C, –20 a +70 °C, 0 a +50 °C; –40 a +85 °C no grau de ambiente severo                                              |
| Vida útil do backlight  | Acima de 20.000 h; acima de 30.000 h em boa parte dos módulos                                                                |
| Ajuste de brilho        | 100 níveis; entre 1% e 30% do máximo pode haver cintilação – faixa não recomendada                                           |
| RTC                     | Opcional; por supercapacitor (±20 ppm a 25 °C, cerca de 7 dias sem alimentação) ou por bateria                               |
| Áudio                   | Buzzer passivo de 3 V embarcado; interface para alto-falante externo em parte da linha                                       |
| Flash de UI             | 8 MB, 16 MB ou 32 MB de fábrica; opções de expansão de até 2 GB                                                              |
| Extras de placa         | Pontos para módulo Wi-Fi com plataforma de nuvem e para módulo de download por USB                                           |

**ATENÇÃO** — o nível elétrico da porta serial varia por grau, e essa é a diferença que mais causa dano ao ligar: nos módulos de graus comercial e industrial mais simples a porta é **TTL de 3,3 V** e o MCU se liga direto, mas nos módulos de 1920×1080, nos graus médico e de ambiente severo a porta é **RS232**, com RS485 numa segunda via. Ligar um MCU de 3,3 V numa porta RS232 não funciona e pode danificar as portas. Confira sempre a folha de dados do módulo específico antes de conectar.

Resolução, cor, nível elétrico da porta serial, faixa de temperatura e tipo de toque variam por modelo. Confirme esses dados na folha de dados do produto ou na página correspondente em [agtdisplays.com.br](http://agtdisplays.com.br), que reúne a linha completa.

## 11. COMPATIBILIDADE DE MICROCONTROLADORES

Os módulos funcionam com qualquer microcontrolador que tenha UART. A exigência comum a todos é terra compartilhado entre MCU e display; o **nível elétrico depende do módulo** e precisa ser conferido na folha de dados antes de ligar:

- **TTL 3,3 V** nos módulos de grau de consumo, comercial e industrial mais simples, e na maior parte dos módulos de construção COF – o MCU se liga direto, e um MCU de 5 V exige divisor resistivo na linha RX do display
- **RS232 na UART2 e UART4 em RS485, RS232 ou TTL, conforme o modelo**, nos módulos de 1920×1080 e nos graus médico e de ambiente severo, este último acrescentando CAN (veja a seção 10.1) – aqui um MCU de 3,3 V não pode ser ligado direto: use um conversor de nível RS232 ou fale por RS485

Boa parte da linha, porém, não fixa o nível: a UART2 é uma interface compatível com TTL e RS232, e a escolha é feita na própria placa por um seletor identificado em serigrafia branca como R232. São dois pads que se fecham com um resistor de 0  $\Omega$  ou com uma gota de solda:

| ESTADO DO SELETOR                      | NÍVEL DA UART2          |
|----------------------------------------|-------------------------|
| Fechado (curto), R232 = 0 – marcado ON | TTL – padrão de fábrica |
| Aberto, R232 = 1 – marcado OFF         | RS232                   |

Ou seja, um módulo que chega em TTL pode ser convertido para RS232 sem trocar de modelo, e vice-versa. A exceção são os módulos cuja folha de dados declara a UART2 fixa em RS232 – caso dos de 1920×1080 e dos graus mais altos, que não trazem o seletor. Confira a folha de dados do modelo antes de alterar o seletor.

- Arduino: Uno, Mega, Nano, Pro Mini, Due, Zero
- ESP: ESP8266, ESP32, ESP32-S2/S3/C3
- STM32: famílias F0, F1, F4, H7 e demais
- PIC: PIC16, PIC18, PIC32
- Raspberry Pi: UART pelo GPIO
- 8051 e variantes
- AVR: ATmega e ATtiny com UART
- ARM: Cortex-M0/M3/M4/M7

O ASIC também pode assumir parte da lógica da aplicação: o núcleo OS executa código 8051 do usuário ou DWIN OS, e o Modbus RTU, executado pela UART4, permite falar direto com CLPs e instrumentos de campo nos modelos que expõem essa porta. A configuração é pelo arquivo 22.bin, e o núcleo exigido depende da implementação de Modbus usada (seção 7).

As duas vias são **mutuamente exclusivas**: o núcleo OS guarda um programa só. Ou você grava o firmware DWIN OS mais o seu programa em assembly DWIN OS, ou grava o seu código C51 – nunca os dois ao mesmo tempo. A escolha é feita no início do projeto, porque trocar depois significa regravar o núcleo OS inteiro. Vale lembrar que nada disso é obrigatório: o núcleo OS pode ficar vazio, com toda a lógica no MCU externo, que é o caminho descrito neste guia.

## 12. CONCLUSÃO

---

O DGUS entrega uma solução completa para interfaces em displays industriais e embarcados. A combinação de editor visual, ferramentas integradas de conversão de ativos e um protocolo serial de apenas duas instruções (0x82 e 0x83) reduz drasticamente o esforço de firmware do lado do microcontrolador.

### PRÓXIMOS PASSOS

- Pratique com interfaces simples antes de projetos complexos
- Documente seus endereços VP em planilha desde o primeiro dia – é o artefato que mais economiza tempo em projetos grandes
- Explore os controles além dos quatro básicos: curvas em tempo real, entrada de texto, RTC, QR Code, rolagem de zona e reprodução de vídeo
- Se o projeto envolve CLP ou instrumento serial, considere testar o Modbus RTU pela UART4 antes de decidir por um MCU externo – se o modelo expuser a UART4
- Guarde o T5LCFG.CFG que você gerou junto com o projeto: o arquivo gravado na IHM não pode ser lido de volta, e sem essa cópia os parâmetros precisam ser remontados do zero

## APÊNDICE A – FONTES CONSULTADAS

---

- **T5L\_DGUSII Application Development Guide V2.9** – DWIN Technology. Base para as seções 3, 4, 6, 8 e 10.
- **Development Guide of T5L ASIC (2024-09-25)** – DWIN Technology.
- **Modbus Setting Guide for T5L Screen** – DWIN Technology. Base para a seção 7.
- **DWIN T5L Smart LCMs' UART C51 Program** – DWIN Technology.
- **Folhas de dados de módulos DWIN de 1920×1080** – quatro modelos da família 2K, de 3,4" a 12,5". Base das linhas de resolução, Flash e níveis de sinal nas seções 4.1, 10 e 10.1.
- **Catálogo DWIN de módulos UART T5L** – [dwin-global.com](http://dwin-global.com): tabelas de estrutura COB e COF. Base das faixas na seção 10.1. A descrição de arquitetura multichip que aparece no material de notícias do site não é repetida pelas folhas de dados e por isso não foi incorporada ao texto.
- Materiais de treinamento "Function Introduction of T5L DGUSII Software" (Artistic Variables, Icon Rotation, Progress Bar, QR Code, RTC Display, Text Scroll, Video Playback, Multilingual Function, entre outros).

Este guia foi desenvolvido pela equipe técnica da AGT Displays para auxiliar engenheiros e desenvolvedores na implementação de soluções profissionais com TFT Módulo Serial. Atualizações e materiais complementares podem ser encontrados em nosso site oficial.