

AGT DISPLAYS

Tutorial Completo Software DGUS para Displays IHM

Guia Completo para Desenvolvimento de Interfaces
Homem-Máquina (IHM) Profissionais

Versão:	1.0
Data:	10/02/2026
Aplicação:	Displays TFT LCD Industriais
Público:	Engenheiros e Desenvolvedores

Índice

1. Introdução	3
2. Configuração Inicial	4
2.1. Download e Instalação	4
2.2. Configuração do Software	4
2.3. Estrutura de Projeto	5
3. Desenvolvimento da Interface	6
3.1. Preparação de Assets	6
3.2. Conversão de Imagens	7
3.3. Módulos Principais	8
4. Comunicação Serial	12
4.1. Protocolo de Comunicação	12
4.2. Endereços VP	13
4.3. Frame de Dados	14
5. Programação Arduino	15
5.1. Estrutura do Código	15
5.2. Funções Principais	16
5.3. Exemplo Completo	17
6. Upload e Deploy	19
6.1. Geração de Arquivos	19
6.2. Upload via SD Card	19
6.3. Upload via T5 Tool	20
7. Troubleshooting	21
8. Aplicações Práticas	22
9. Especificações Técnicas	23
10. Conclusão	24

1. Introdução

O software DGUS é uma ferramenta profissional para desenvolvimento de interfaces IHM (Human-Machine Interface) para displays industriais e embarcados. Através desta plataforma, é possível criar interfaces gráficas sofisticadas e intuitivas para uma ampla gama de aplicações industriais e comerciais.

Características Principais:

- Interface visual intuitiva para design de telas
- Suporte a múltiplas resoluções de display (480x272 até 1024x600)
- Biblioteca de widgets e controles pré-configurados
- Comunicação serial com microcontroladores (UART TTL)
- Suporte a diversos tipos de displays: TN, STN, TFT, OLED
- Ferramentas integradas para conversão de imagens e fontes
- Sistema de endereçamento VP (Variable Pointer) para controle de dados
- Compatibilidade com Arduino, STM32, ESP32, PIC e outros MCUs

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 baixar dois componentes essenciais do site oficial da AGT Displays:

- **DGUS Software:** Editor visual para criação de interfaces
- **Driver XR21V1414:** Driver USB-Serial para comunicação com displays baseados em SDL662B

■ **DICA:** Não é necessário instalar o DGUS Software. Após extrair os arquivos, você pode executar o arquivo 'DGUS.exe' diretamente da pasta extraída.

2.2. Configuração do Software

1. Execute o arquivo **DGUS.exe**
2. Por padrão, o software abre em idioma chinês
3. Acesse o menu **Settings** (ícone de engrenagem)
4. Altere o idioma para **English**
5. Reinicie o software para aplicar as alterações

2.3. Estrutura de Projeto

Ao criar um novo projeto no DGUS, você precisará definir:

Parâmetro	Descrição	Exemplo
Resolução	Dimensões do display em pixels	800 x 480
Nome do Projeto	Identificador do projeto	SmartHome_V1
Pasta do Projeto	Diretório de trabalho	C:/Projetos/AGT/SmartHome

3. Desenvolvimento da Interface

3.1. Preparação de Assets

A organização adequada dos arquivos é fundamental para um projeto eficiente. O DGUS utiliza uma estrutura de pastas numeradas para categorizar os assets:

Pasta	Conteúdo	Formato	Uso
/32	Imagens de fundo	.BMP	Backgrounds das telas
/46	Ícones de botões	.BMP	Controles e indicadores
/62	Ícones numéricos	.BMP	Displays de valores
/ICL	Arquivos compilados	.ICL	Bibliotecas de ícones

3.2. Conversão de Imagens

O DGUS possui uma ferramenta integrada para conversão e redimensionamento de imagens:

1. Clique no botão **'Picture Conversion Tool'** no canto superior direito
2. Configure a resolução de destino (ex: 800 x 480 pixels)
3. Adicione as imagens da pasta /32 (backgrounds)
4. Clique em **'Image Conversion'**
5. Selecione a pasta de destino dentro do projeto
6. As imagens serão redimensionadas e convertidas automaticamente

Geração de Arquivos ICL:

Os arquivos ICL (Icon Library) são bibliotecas compiladas de ícones usadas pelo display para renderização rápida. Para criar arquivos ICL:

1. Acesse **Settings** → **Icon Tool**
2. Selecione as imagens da pasta /32 e gere **'32.ICL'**
3. Repita o processo para pasta /46 → **'46.ICL'**
4. E para pasta /62 → **'62.ICL'**
5. Salve todos os arquivos ICL na pasta **'/DWIN_SET'** do projeto

3.3. Módulos Principais

O DGUS oferece diversos módulos (widgets) para criar interfaces interativas. Os mais utilizados são:

a) Basic Touch Control

Módulo fundamental para navegação entre telas e detecção de toques do usuário.

- **Função:** Cria áreas sensíveis ao toque na tela
- **Ação:** Troca de página/tela ao ser pressionado
- **Configuração:** Define área retangular e página de destino
- **Uso típico:** Botões de menu, navegação, seleção de opções

Exemplo de configuração: - Área: X=100, Y=200, Largura=150, Altura=50 - Página destino: 02 (Tela de Temperatura) - Efeito: Ao tocar, muda para tela 02

b) Artistic Variable Display

Exibe valores dinâmicos recebidos do microcontrolador, como temperatura, umidade, velocidade, etc.

- **Função:** Mostra dados numéricos em tempo real
- **Endereço VP:** Identificador único para cada variável
- **Tipo de dado:** Integer, Float, String
- **Formatação:** Número de dígitos, casas decimais, unidades

Parâmetro	Descrição	Exemplo
VP Address	Endereço da variável	1000 (Temperatura)
Variable Type	Tipo de dado	Integer / Float
Digits	Número de dígitos	2 (ex: 25)
Decimal Places	Casas decimais	1 (ex: 25.5)
Icon Library	Fonte dos números	62.ICL

■ **DICA:** Use endereços VP em múltiplos de 100 ou 1000 para facilitar a organização. Por exemplo: 1000-1999 para sensores, 5000-5999 para controles.

c) Icon Display

Exibe ícones diferentes baseado no valor de uma variável. Ideal para botões ON/OFF e indicadores de estado.

- **Função:** Alterna entre dois ou mais ícones
- **Uso típico:** Botões toggle, indicadores LED, estados de sistema
- **Configuração:** Valor mínimo = ícone OFF, Valor máximo = ícone ON
- **Transparência:** Suporta fundo transparente

Exemplo - Controle de Lâmpada: - VP Address: 5100 - Icon Library: 46.ICL - Valor 0: ícone lamp_off.bmp (índice 0 na biblioteca) - Valor 1: ícone lamp_on.bmp (índice 1 na biblioteca) - Display Mode: Transparent

d) Increment Adjustment

Módulo de controle que incrementa ou alterna valores quando pressionado. Combinado com Icon Display, cria botões interativos perfeitos.

Parâmetro	Função	Valor Típico
VP Address	Endereço da variável	5100 (mesmo do Icon Display)
Address Method	Tipo de operação	PLUS (incremento)
System Value	Valor a incrementar	1
Upper Limit	Valor máximo	1 (para toggle ON/OFF)
Over Limit Operation	Ação no limite	Cycle (volta ao início)
Touch Effect	Resposta ao toque	Disposable (uma vez)

■ ■ **IMPORTANTE:** O Increment Adjustment e o Icon Display devem compartilhar o mesmo endereço VP para funcionarem de forma sincronizada.

4. Comunicação Serial

A comunicação entre o display IHM AGT e o microcontrolador ocorre através de interface serial UART (Universal Asynchronous Receiver-Transmitter) utilizando protocolo proprietário do DGUS.

4.1. Protocolo de Comunicação

Parâmetro	Valor	Descrição
Baud Rate	115200 bps	Velocidade de transmissão padrão
Data Bits	8 bits	Tamanho do byte de dados
Stop Bits	1 bit	Bit de parada
Parity	None	Sem paridade
Flow Control	None	Sem controle de fluxo
Níveis Lógicos	TTL 3.3V/5V	Compatível com Arduino

4.2. Sistema de Endereços VP

VP (Variable Pointer) é o sistema de endereçamento utilizado para identificar cada variável ou controle na interface. Cada módulo configurado no DGUS possui um endereço VP único.

Endereço VP	Tipo	Função	Range
0x1000 (4096)	Sensor	Temperatura	0-100
0x2000 (8192)	Sensor	Umidade	0-100
0x3000 (12288)	Sensor	Pressão	0-1200
0x5100 (20736)	Controle	Lâmpada ON/OFF	0-1
0x5200 (20992)	Controle	Ventilador ON/OFF	0-1
0x5300 (21248)	Controle	Ar-condicionado	0-1

■ **DICA: Organize os endereços VP por categorias usando faixas de valores. Isso facilita a manutenção e expansão do projeto.**

4.3. Estrutura do Frame de Dados

A comunicação entre MCU e display segue um protocolo de frames específico:

a) Frame de Escrita (MCU → Display)

5A A5 [LEN] 82 [VP_HIGH] [VP_LOW] [DATA_HIGH] [DATA_LOW] Onde: • 5A A5: Header (identificador de início do frame) • LEN: Tamanho dos dados (em bytes) • 82: Comando de escrita em registro VP • VP_HIGH: Byte alto do endereço VP • VP_LOW: Byte baixo do endereço VP • DATA_HIGH: Byte alto do valor • DATA_LOW: Byte baixo do valor

Exemplo: Enviar temperatura 25°C para VP 0x1000 Frame: 5A A5 05 82 10 00 00 19
Detalhamento: • 5A A5: Header • 05: Comprimento (5 bytes de dados) • 82: Comando de escrita • 10 00: VP = 0x1000 (4096 decimal) • 00 19: Valor = 0x0019 (25 decimal)

b) Frame de Leitura (Display → MCU)

5A A5 [LEN] 83 [VP_HIGH] [VP_LOW] [DATA] Onde: • 5A A5: Header • LEN: Tamanho dos dados • 83: Comando de leitura de registro VP • VP_HIGH: Byte alto do endereço VP • VP_LOW: Byte baixo do endereço VP • DATA: Dados lidos (1 ou 2 bytes dependendo do tipo)

Exemplo: Display envia estado de botão VP 0x5100 Frame recebido: 5A A5 04 83 51 00 01
Detalhamento: • 5A A5: Header • 04: Comprimento (4 bytes) • 83: Comando de leitura • 51 00: VP = 0x5100 (20736 decimal) • 01: Valor = 1 (botão ligado)

■ ■ IMPORTANTE: Sempre conecte o pino GND do display ao GND do microcontrolador, mesmo que ambos tenham fontes de alimentação separadas. Isso garante referência comum de terra.

5. Programação Arduino

A programação do Arduino para comunicação com displays IHM AGT requer bibliotecas específicas e estruturação adequada do código para envio e recebimento de dados.

5.1. Estrutura do Código

- **Includes:** Bibliotecas necessárias (Wire.h, SoftwareSerial.h, sensor libraries)
- **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

```
// Bibliotecas necessárias #include #include // Para sensor BME280 // Definição de
pinos #define RELAY_LIGHT 2 #define RELAY_FAN 3 #define RELAY_AC 4 #define RELAY_TV
5 // Endereços VP #define VP_TEMPERATURE 0x1000 #define VP_HUMIDITY 0x2000 #define
VP_PRESSURE 0x3000 #define VP_LIGHT 0x5100 #define VP_FAN 0x5200 #define VP_AC
0x5300 #define VP_TV 0x5400 // Objeto do sensor Adafruit_BME280 bme; void setup() {
Serial.begin(115200); // Comunicação com display // Inicializa sensor if
(!bme.begin(0x76)) { Serial.println("Sensor BME280 não encontrado!"); } //
Configura pinos dos relés pinMode(RELAY_LIGHT, OUTPUT); pinMode(RELAY_FAN, OUTPUT);
pinMode(RELAY_AC, OUTPUT); pinMode(RELAY_TV, OUTPUT); }
```

5.2. Funções Principais

a) Envio de Dados para Display

```
void sendToDisplay(uint16_t vp_address, uint16_t data) { uint8_t frame[8]; // Monta
o frame frame[0] = 0x5A; // Header byte 1 frame[1] = 0xA5; // Header byte 2 frame[2]
= 0x05; // Comprimento frame[3] = 0x82; // Comando de escrita frame[4] =
(vp_address >> 8) & 0xFF; // VP high byte frame[5] = vp_address & 0xFF; // VP low
byte frame[6] = (data >> 8) & 0xFF; // Data high byte frame[7] = data & 0xFF; //
Data low byte // Envia o frame pela serial Serial.write(frame, 8); } // Exemplo de
uso: float temperature = bme.readTemperature(); sendToDisplay(VP_TEMPERATURE,
(uint16_t)(temperature * 10));
```

b) Recebimento de Dados do Display

```
void receiveFromDisplay() { if (Serial.available() >= 7) { // Lê o header if
  (Serial.read() == 0x5A && Serial.read() == 0xA5) { uint8_t len = Serial.read();
  uint8_t cmd = Serial.read(); if (cmd == 0x83) { // Comando de leitura uint16_t
  vp_address = (Serial.read() << 8) | Serial.read(); uint8_t value = Serial.read();
  // Processa o comando recebido switch(vp_address) { case VP_LIGHT:
  digitalWrite(RELAY_LIGHT, value); break; case VP_FAN: digitalWrite(RELAY_FAN,
  value); break; case VP_AC: digitalWrite(RELAY_AC, value); break; case VP_TV:
  digitalWrite(RELAY_TV, value); break; } } } }
```

5.3. Loop Principal Completo

```
void loop() { static unsigned long lastUpdate = 0; unsigned long currentMillis =
  millis(); // Atualiza sensores a cada 1 segundo if (currentMillis - lastUpdate >=
  1000) { lastUpdate = currentMillis; // Lê sensores float temp =
  bme.readTemperature(); float humidity = bme.readHumidity(); float pressure =
  bme.readPressure() / 100.0F; // Envia para display (multiplica por 10 para manter 1
  casa decimal) sendToDisplay(VP_TEMPERATURE, (uint16_t)(temp * 10));
  sendToDisplay(VP_HUMIDITY, (uint16_t)(humidity * 10)); sendToDisplay(VP_PRESSURE,
  (uint16_t)(pressure)); } // Processa comandos do display receiveFromDisplay(); //
  Pequeno delay para estabilidade delay(10); }
```

■ **DICA: Use millis() ao invés de delay() para não bloquear a execução do programa. Isso permite que o Arduino continue processando comandos do display enquanto aguarda o próximo ciclo de leitura dos sensores.**

6. Upload e Deploy

6.1. Geração de Arquivos do Projeto

Após concluir o design da interface no DGUS, você precisa gerar os arquivos finais que serão carregados no display:

- No DGUS, clique em **File** → **Save** para salvar o projeto
- Clique em **Generate** no menu principal
- O software gerará automaticamente os seguintes arquivos:

Arquivo	Descrição	Tamanho Típico
13.BIN	Dados de configuração do projeto	1-5 KB
14.BIN	Biblioteca de fontes	10-50 KB
22.BIN	Arquivo de configuração principal	1-2 KB
*.ICL	Bibliotecas de ícones (32, 46, 62)	50-200 KB
T5LCFG_*.CFG	Arquivo de configuração de hardware	1 KB

■ ■ **IMPORTANTE:** Todos esses arquivos devem ser copiados para a pasta **DWIN_SET** no cartão SD ou enviados via **T5 Download Tool**.

6.2. Upload via Cartão SD

O método mais simples e confiável para fazer upload do projeto para o display é através de cartão SD (microSD ou SD padrão, dependendo do modelo):

1. Formate o cartão SD em FAT32 (não use exFAT ou NTFS)
2. Crie uma pasta chamada **DWIN_SET** na raiz do cartão
3. Copie todos os arquivos gerados (.BIN, .ICL, .CFG) para dentro da pasta **DWIN_SET**
4. Desligue o display completamente
5. Insira o cartão SD no slot do display
6. Ligue o display - ele detectará automaticamente os arquivos
7. Aguarde a tela azul indicando o progresso do upload
8. Quando aparecer 'Update Success' ou tela laranja, desligue o display
9. Remova o cartão SD
10. Ligue o display novamente - o projeto estará carregado

■ **DICA:** Sempre desligue o display antes de inserir ou remover o cartão SD. Fazer isso com o display ligado pode corromper os dados.

6.3. Upload via T5 Download Tool

Para atualizações rápidas durante desenvolvimento, você pode usar a ferramenta T5 Download Tool que faz upload via USB diretamente:

1. Baixe e instale o T5 Download Tool (disponível no site AGT)
2. Conecte o display ao PC via cabo USB
3. Abra o T5 Download Tool
4. Selecione a porta COM correta do display
5. Configure o Baud Rate para 115200
6. Clique em **'Select Files'** e adicione os arquivos do projeto
7. Clique em **'Download'** para iniciar o upload
8. Aguarde a conclusão (barra de progresso chegará a 100%)
9. O display reiniciará automaticamente com o novo projeto

Método	Vantagens	Desvantagens
SD Card	<ul style="list-style-type: none">• Mais confiável• Não precisa de driver USB• Funciona em qualquer display	<ul style="list-style-type: none">• Mais lento• Requer cartão SD• Vários passos
T5 Tool	<ul style="list-style-type: none">• Rápido e prático• Ideal para desenvolvimento• Upload direto	<ul style="list-style-type: none">• Requer driver USB• Depende de porta COM• Pode ter problemas de conexão

7. Troubleshooting

Problemas comuns e suas soluções durante desenvolvimento e deployment:

Problema: Display não atualiza após upload

Causa Provável: Arquivos não foram copiados corretamente

Solução: Verifique se a pasta DWIN_SET está na raiz do SD e contém todos os arquivos .BIN e .ICL

Problema: Tela azul infinita no boot

Causa Provável: Cartão SD com problema ou arquivos corrompidos

Solução: Reformate o SD em FAT32 e gere os arquivos novamente no DGUS

Problema: Valores não aparecem no display

Causa Provável: Endereço VP incorreto ou tipo de variável errado

Solução: Verifique se o VP no código Arduino corresponde exatamente ao VP configurado no DGUS

Problema: Botões não respondem ao toque

Causa Provável: Módulo Touch Control não configurado ou área muito pequena

Solução: Aumente a área sensível do módulo ou verifique se está associado à página correta

Problema: Arduino não se comunica com display

Causa Provável: Baud rate diferente ou cabos TX/RX invertidos

Solução: Confirme 115200 bps em ambos. Conecte TX do Arduino ao RX do display e vice-versa

Problema: Ícones não aparecem corretamente

Causa Provável: Arquivo .ICL não foi gerado ou nome incorreto

Solução: Gere novamente os arquivos ICL usando Icon Tool e verifique se estão na pasta DWIN_SET

Problema: Display mostra caracteres estranhos

Causa Provável: Fonte não foi criada ou configurada

Solução: Use Font Generate Tool para criar 0_XXX.HZK e coloque na pasta DWIN_SET

Problema: T5 Tool não encontra porta COM

Causa Provável: Driver USB não instalado

Solução: Instale o driver XR21V1414 e reinicie o PC

Problema: Valores atualizando muito rápido

Causa Provável: Sem delay ou controle de timing no Arduino

Solução: Use millis() para controlar intervalo de envio (recomendado 500-1000ms)

Problema: Relés não acionam quando botões são pressionados

Causa Provável: VP address não corresponde entre código e DGUS

Solução: Confirme que Icon Display e Increment Adjustment compartilham mesmo VP e código recebe corretamente

■ **DICA DE DEBUG:** Use Serial Monitor do Arduino (115200 baud) para visualizar os frames sendo enviados e recebidos. Isso ajuda muito a identificar problemas de comunicação.

8. Aplicações Práticas

Os displays IHM AGT com DGUS são extremamente versáteis. Aqui estão algumas aplicações reais implementadas com sucesso:

Automação Residencial (Smart Home)

Controle centralizado de iluminação, climatização, cortinas e segurança

Características principais:

- Monitoramento de temperatura, umidade e qualidade do ar
- Controle de até 16 relés para dispositivos diversos
- Agendamento de eventos e cenas pré-programadas
- Interface intuitiva com feedback visual

Controle de Máquinas CNC

Interface para operação de máquinas de corte, fresagem e impressão 3D

Características principais:

- Exibição de coordenadas X, Y, Z em tempo real
- Controle de velocidade e avanço
- Monitoramento de temperatura de motores
- Botões de emergência e pausa

Sistema de Irrigação Inteligente

Controle automatizado de irrigação para agricultura

Características principais:

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

Painel de Monitoramento Industrial

Supervisão de processos industriais em tempo real

Características principais:

- Múltiplos sensores de temperatura e pressão
- Alarmes visuais e sonoros
- Gráficos de tendência
- Controle de bombas e válvulas

Sistema de Controle de Acesso

Gerenciamento de entrada e saída com biometria/RFID

Características principais:

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

9. Especificações Técnicas

Especificações dos displays IHM AGT compatíveis com DGUS:

Especificação	Detalhes
Tecnologias de Display	TN, STN, FSTN, TFT, OLED
Tamanhos Disponíveis	3.5" a 10.1" (formato 4:3 e 16:9)
Resoluções Típicas	480x272, 800x480, 1024x600
Interface de Comunicação	UART TTL (3.3V/5V)
Baud Rates Suportados	2400 a 921600 bps (padrão 115200)
Tensão de Operação	5V DC (alguns modelos 12V/24V)
Consumo de Corrente	150mA a 500mA (dependendo do tamanho)
Temperatura de Operação	-20°C a +70°C
Temperatura de Armazenamento	-30°C a +80°C
Memória Flash	128MB a 512MB
Processador	ARM Cortex (modelos variados)
Touchscreen	Resistivo de 4 fios
Vida Útil do Backlight	50.000 horas típico
Montagem	Painel frontal ou chassi

Compatibilidade de Microcontroladores

Os displays IHM AGT funcionam com praticamente qualquer microcontrolador que possua comunicação serial UART:

- **Arduino:** Uno, Mega, Nano, Pro Mini, Due, Zero
- **ESP:** ESP8266, ESP32, ESP32-S2/S3/C3
- **STM32:** Toda família STM32 (F0, F1, F4, H7, etc)
- **PIC:** PIC16, PIC18, PIC32
- **Raspberry Pi:** Através de GPIO serial
- **8051:** Família 8051 e variantes
- **AVR:** ATmega, ATtiny com UART
- **ARM:** Cortex-M0/M3/M4/M7

10. Conclusão

O software DGUS representa uma solução profissional e completa para desenvolvimento de interfaces IHM em displays industriais e embarcados. Sua combinação de interface visual intuitiva, ferramentas integradas e protocolo de comunicação robusto o tornam ideal para uma ampla gama de aplicações.

Principais Vantagens:

- Desenvolvimento visual sem necessidade de programação gráfica complexa
- Protocolo de comunicação serial simples e confiável
- Ampla compatibilidade com microcontroladores diversos
- Ferramentas integradas para conversão de assets
- Sistema de módulos pré-configurados acelera o desenvolvimento
- Displays robustos adequados para ambientes industriais
- Excelente relação custo-benefício
- Suporte a múltiplas resoluções e tamanhos

Próximos Passos:

- Pratique criando interfaces simples antes de projetos complexos
- Explore todos os módulos disponíveis no DGUS Editor
- Teste diferentes configurações de comunicação serial
- Documente seus endereços VP em planilha para projetos grandes
- Considere usar bibliotecas Arduino específicas para facilitar comunicação
- Participe de comunidades e fóruns para trocar experiências

Para mais informações sobre ns produtos e serviços,
entre em contato conosco através de nossos canais oficiais.

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