PRD GT System Manual

Table of Contents

1. Technical Specifications

The PRD GT system is an advanced unit with touch display.
Designed for professional use with high durability in challenging operating conditions.

Feature	Specification
Operating Voltage	8—18 V DC
Operating Temperature	-40°C to +85°C
Construction Material	Aluminum
Display	3.5" / 7" TFT LCD Touch (in aluminum frame)

2. Safety Information & Warranty

2.1 Disclaimer

Important: PRD bears no responsibility for damage to persons or property during installation or use of its products. Products are intended for specific use and compliance with safety or emission laws is not guaranteed.

2.2 Warranty Terms

Parameter	Description
Duration	2 years from purchase date

Not covered by warranty:

- Normal wear and tear
- Damage from accident, misuse or unauthorized repair
- Modified products or commercial use

3. Installation

3.1 Wiring Diagram

Wire Color	Signal	Description
White	+12V	Power Supply
Brown	GND	Ground
Yellow	CAN_H	CAN Bus High
Green	CAN_L	CAN Bus Low

> **Caution:** Check wire polarity before installation.

3.2 Installation Steps

1. Mount the device in a stable position
2. Connect wires according to the wiring diagram
3. Check all connections
4. Apply power to the system

4. Configuration with PrdConfigurator

4.1 PrdConfigurator Installation

1. Install PrdConfigurator software on computer
2. Create new project
3. Connect device via USB

4.2 Basic Page 0 Configuration

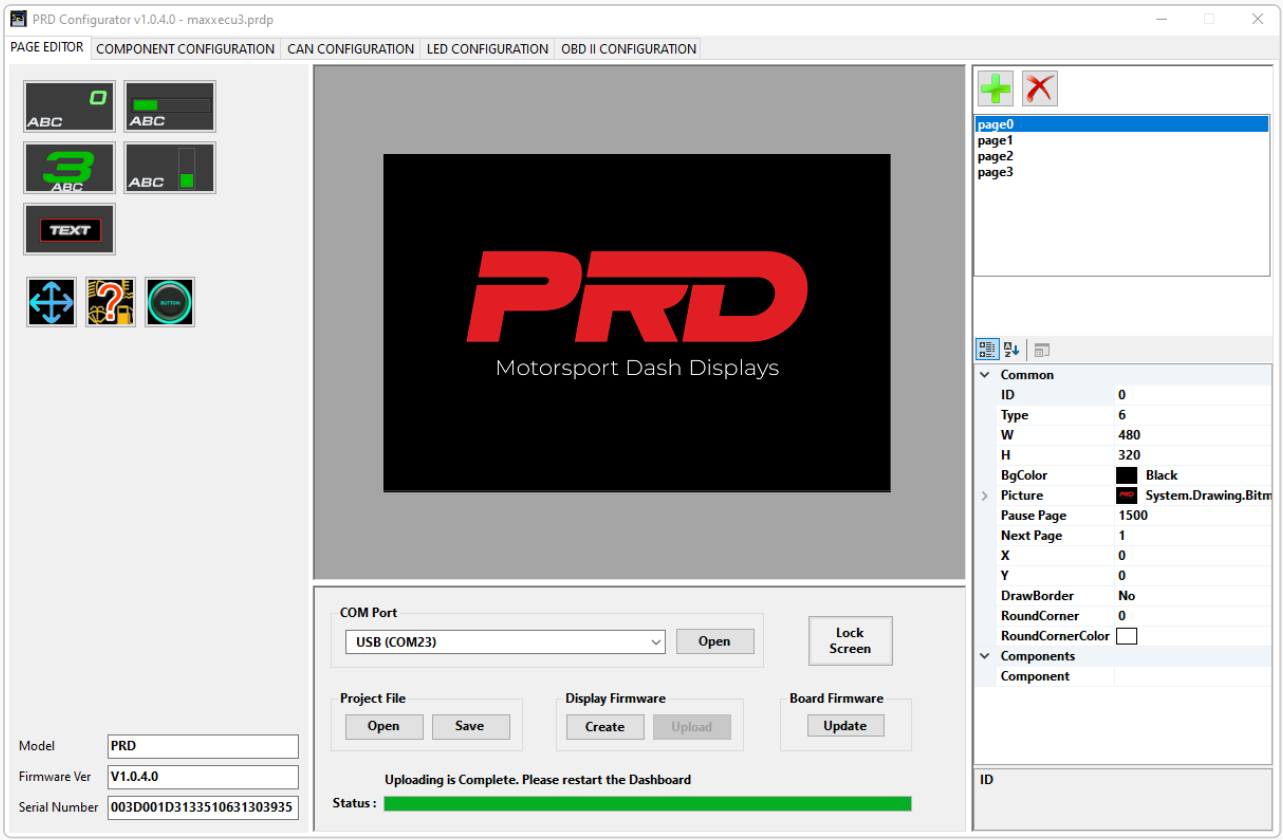
First Steps: Before diving into the interface details, you'll need to configure the startup page (Page 0) which is the first screen users see when the device powers on.

Parameter	Instructions	Details
Logo	Select "Picture" to upload logo	Format: BMP, 480×320 pixels (GT 3.5 full screen)
Display Duration	Set pause time (Pause Page)	1000ms = 1 second, 0 = permanent display
Next Page	Select next page (Next Page)	0 = stay on same page, 1 = go to page1

Important: Page 0 is typically used as a startup logo/splash screen that displays for a few seconds before automatically switching to your main dashboard (Page 1).

4.3 PAGE EDITOR Interface Overview

Purpose: The PAGE EDITOR is the main design environment where you create custom interfaces for your PRD GT display. Think of it as a digital canvas where you build multiple screens for your dashboard.

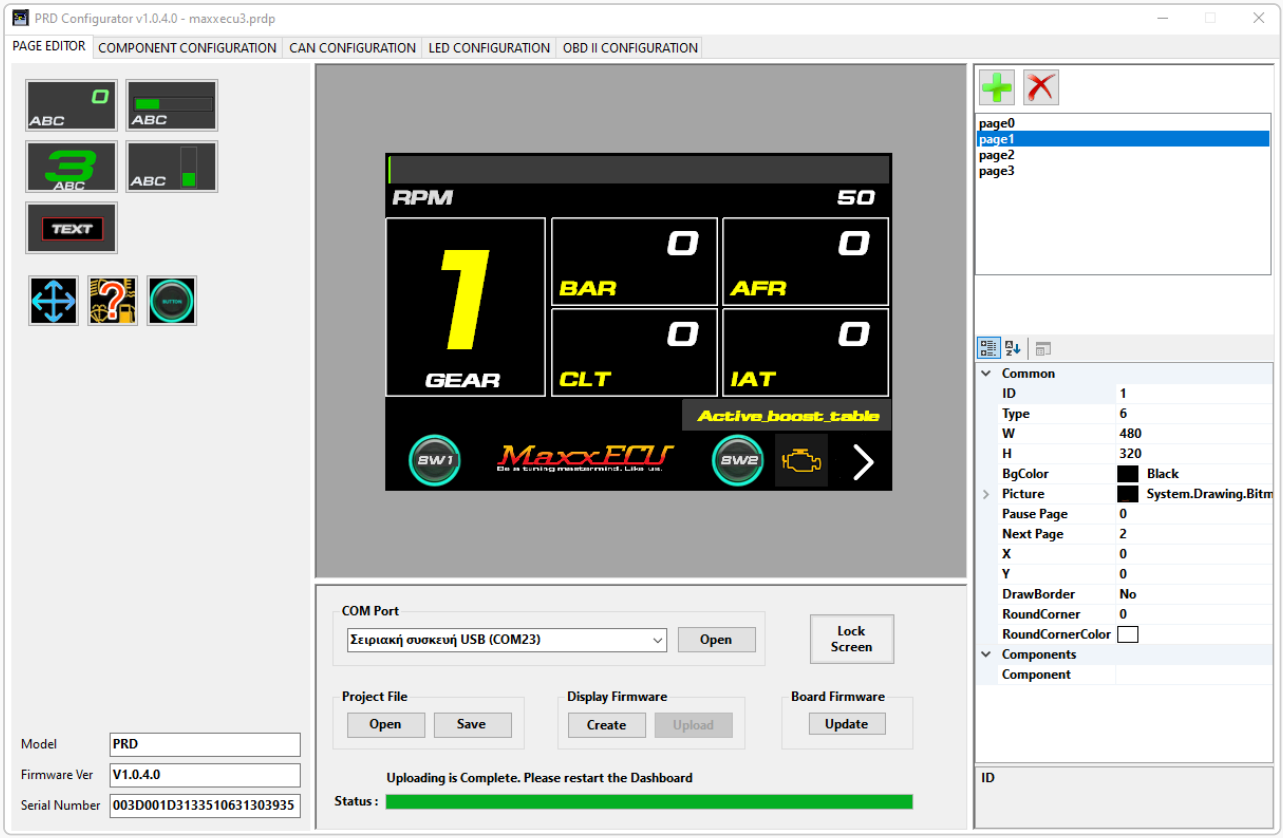


PRD Configurator: PAGE EDITOR showing the main design interface with component library, canvas, and properties panel

Interface Layout Overview:

The PAGE EDITOR consists of four main areas: Component Library (left), Design Canvas (center), Properties Panel (right), and System Controls (bottom).

Real Dashboard Example



PRD Configurator: Example of a complete racing dashboard showing RPM gauge, gear indicator, engine parameters (BAR, AFR, CLT, IAT), control buttons, and navigation

Dashboard Elements Example:

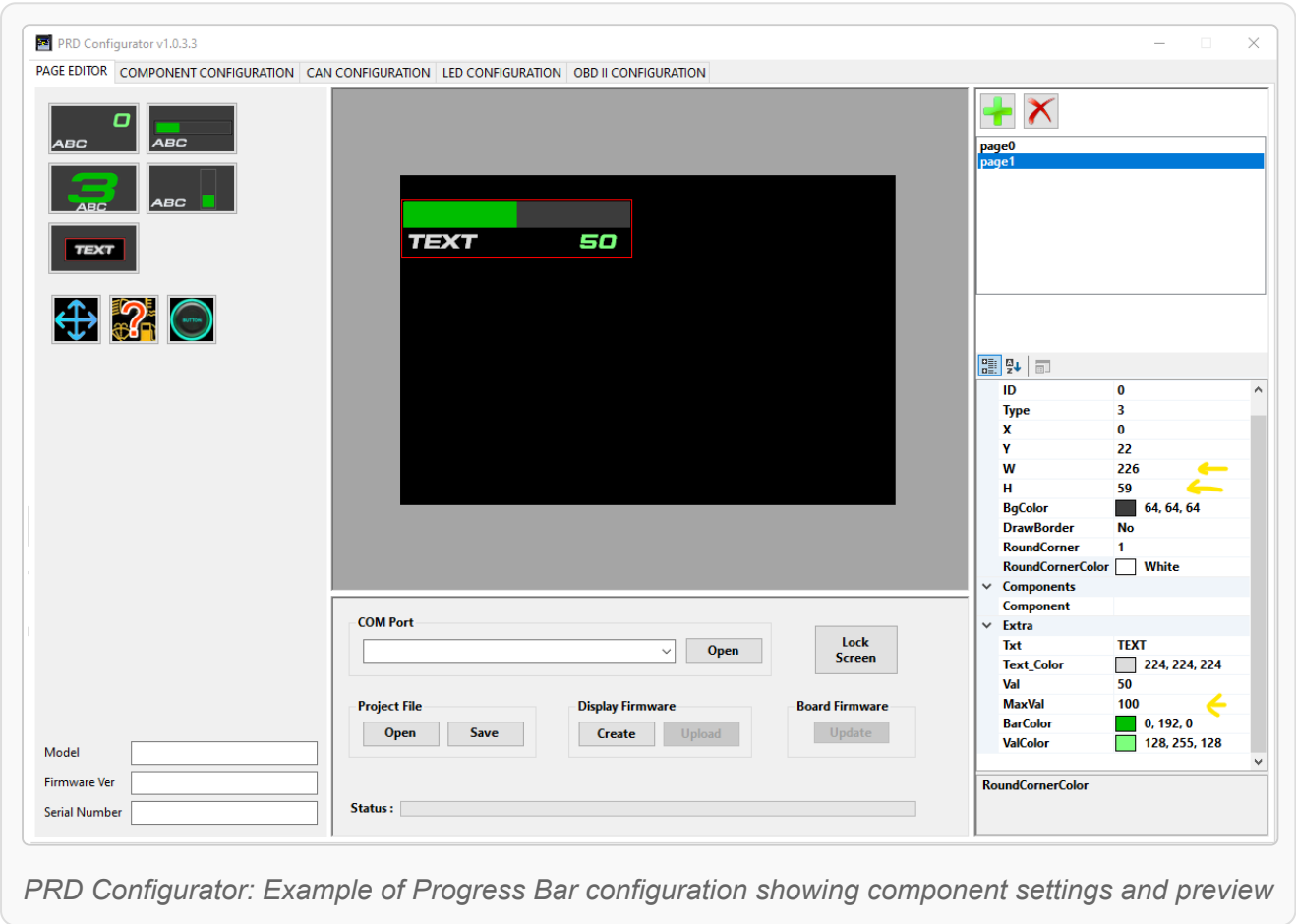
Element	Type	Purpose	Position
RPM Gauge	Analog Gauge	Engine speed monitoring	Top center
Gear Indicator	Large Text Display	Current gear position	Left side (yellow "1")
BAR Display	Digital Value	Boost pressure	Top right
AFR Display	Digital Value	Air/Fuel Ratio	Top far right
CLT Display	Digital Value	Coolant Temperature	Bottom right
IAT Display	Digital Value	Intake Air Temperature	Bottom far right
SW1 Button	Control Button	User-defined function	Bottom left
SW2 Button	Control Button	User-defined function	Bottom right
Navigation Arrow	Page Navigation	Switch to next page	Right edge

Logo/Branding	Picture Element	Logo/Brand Image	Bottom center
Status Text	Text Element	"Active boost table" info	Center bottom

Basic Workflow:

1. **Start Design:** Drag components from left panel to center canvas
2. **Configure Elements:** Select element and modify properties in right panel
3. **Add Pages:** Use + button to create multiple screens
4. **Set Navigation:** Configure "Next Page" for automatic or manual page switching
5. **Connect Device:** Select COM port and click "Open"
6. **Upload Design:** Click "Create" then "Upload" to send to device

4.4 Page 1 Configuration - Progress Bar



PRD Configurator: Example of Progress Bar configuration showing component settings and preview

Element Settings Example:

Parameter	Value	Description
ID	0	Unique element identifier
Type	3	Type 3 = Progress Bar
X, Y	0, 25	Position (pixels from top-left)
W, H	226×59	Width × Height (pixels)

Display Settings Example:

Parameter	Value	Description
BgColor	64,64,64	Background color (dark gray)
DrawBorder	No	No border
RoundCorner	1	Rounded corners (1 pixel)
RoundCornerColor	White	Corner color
BarColor	0,192,0	Bar color (dark green)
ValColor	128,255,128	Value color (bright green)

Data Settings Example:

Parameter	Value	Description
Component	EngineSpeed [CanBus]	Data from CAN Bus (engine RPM)
Txt	"RPM"	Label (Revolutions Per Minute)
Text_Color	224,224,224	Text color (light gray)
Val	50	Current value (50 RPM)
MaxVal	100	Maximum value (100 RPM)

⚠ **Critical Step - Component Linking:** The "Component" dropdown selection is essential! This is where you connect your visual element to actual CAN Bus data. Without selecting a component from the dropdown, the selected element won't display real-time values.

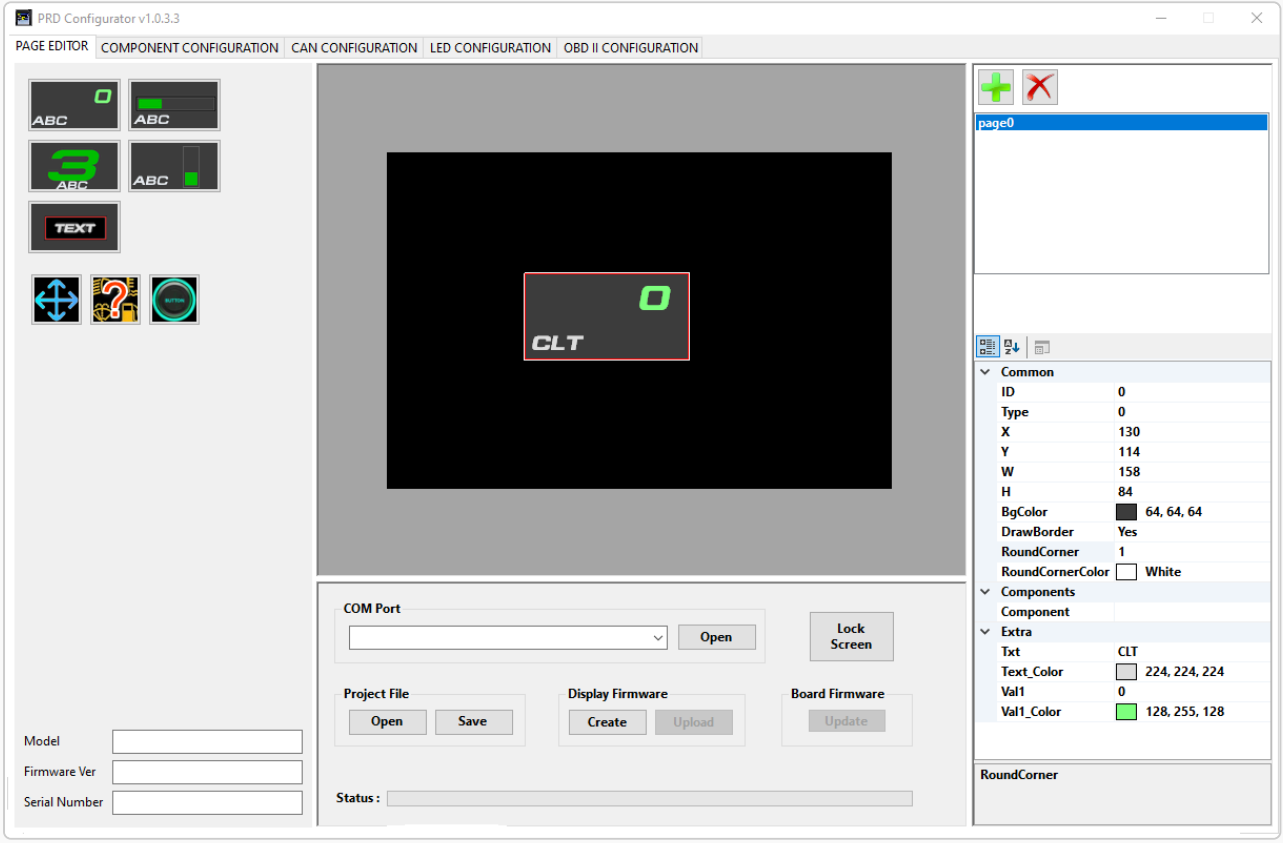
Component Selection Process:

1. **Complete CAN Configuration:** First set up your CAN messages and signals (Section 5)
2. **Generate Components:** Click "Generate Components" in CAN Configuration tab
3. **Select Component:** Use the "Component" dropdown to choose your data source
4. **Configure Display:** Set MaxVal, colors, and text labels accordingly

Available Component Sources:

- **[CanBus]** - Data from CAN Bus signals (e.g., EngineSpeed, CoolantTemp)
- **[Manual]** - Manually entered values for testing
- **[Calculated]** - Derived values from multiple sources

4.5 Temperature Display



PRD Configurator: Example of Temperature Display (CLT - Coolant Temperature) configuration

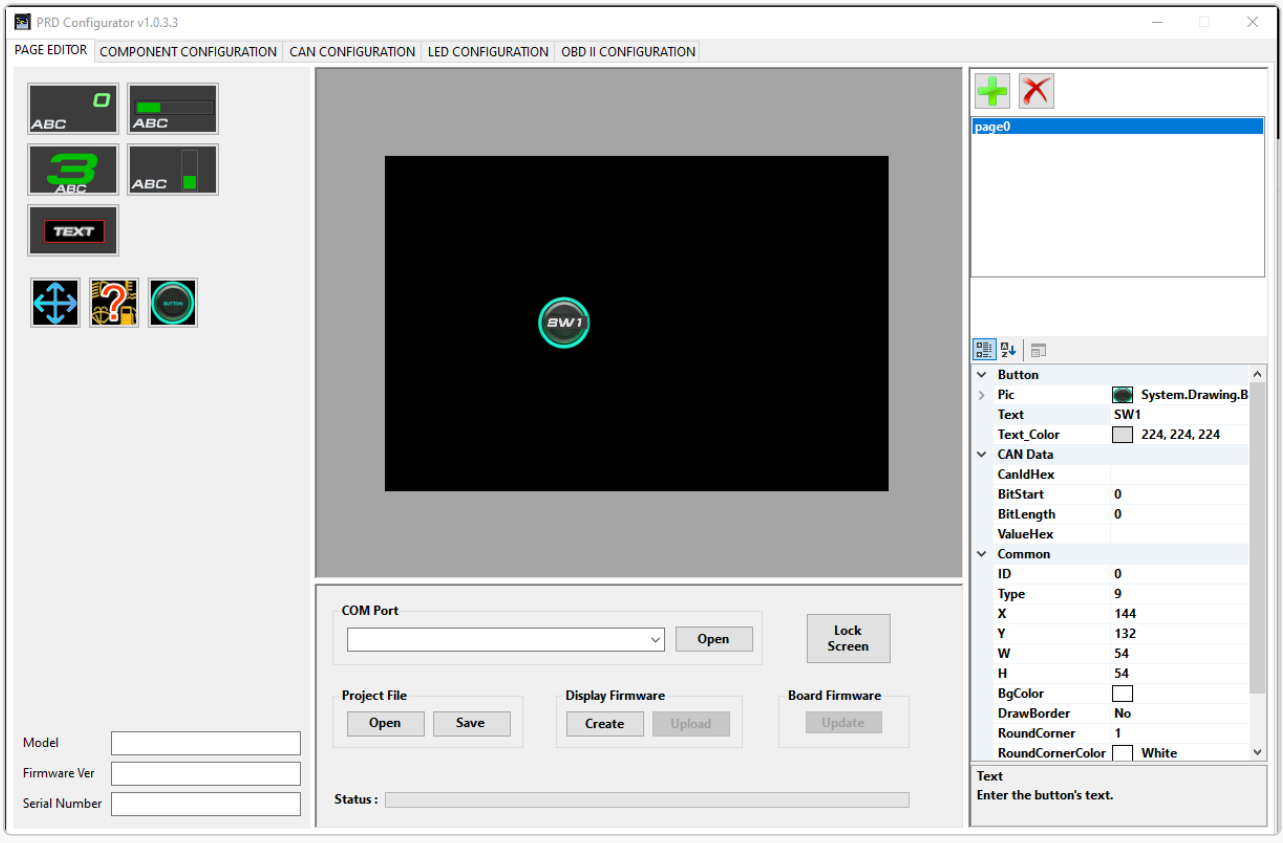
Properties Example:

Parameter	Value	Description
ID	0	Unique identifier

Type	0	Element type (0 = Basic)
X, Y	161, 118	Position (pixels from top-left)
W, H	158×84	Width × Height (pixels)
Component	CoolantTemp [CanBus]	Coolant temperature (CAN Bus)
Txt	"CLT"	Abbreviation (Coolant Temp)
Val1_Color	128,255,128	Value color

⚠ **Component Linking Required:** The "Component" field must be set to a valid CAN component (e.g., CoolantTemp [CanBus]) for the temperature display to show real-time data. Without proper component linking, the display will show static values only.

4.6 Control Button

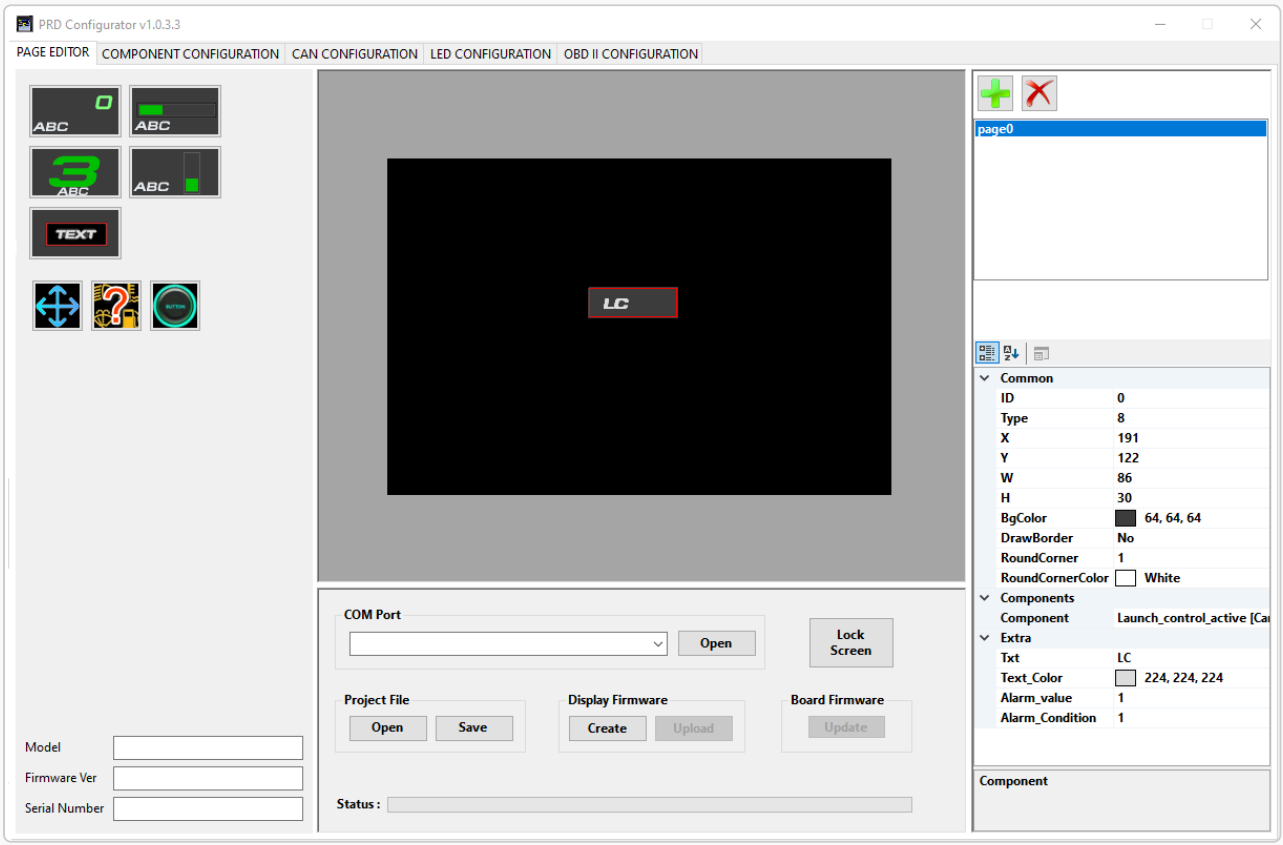


PRD Configurator: Example of Control Button (SW1) configuration with CAN data settings

Properties Example:

Parameter	Value	Description
Type	9	Type 9 = Button
X, Y	151, 168	Position (pixels from top-left)
W, H	54×54	Width × Height (square)
Text	"SW1"	Text ("Switch 1")
CanHex	666	Data in HEX
BitStart	48	Starting bit (0-indexed)
BitLength	1	Data length (bits)
ValueHex	1	Activation value

4.7 Alarm Element / Launch Control



PRD Configurator: Example of Launch Control (LC) alarm element configuration with CAN Bus integration

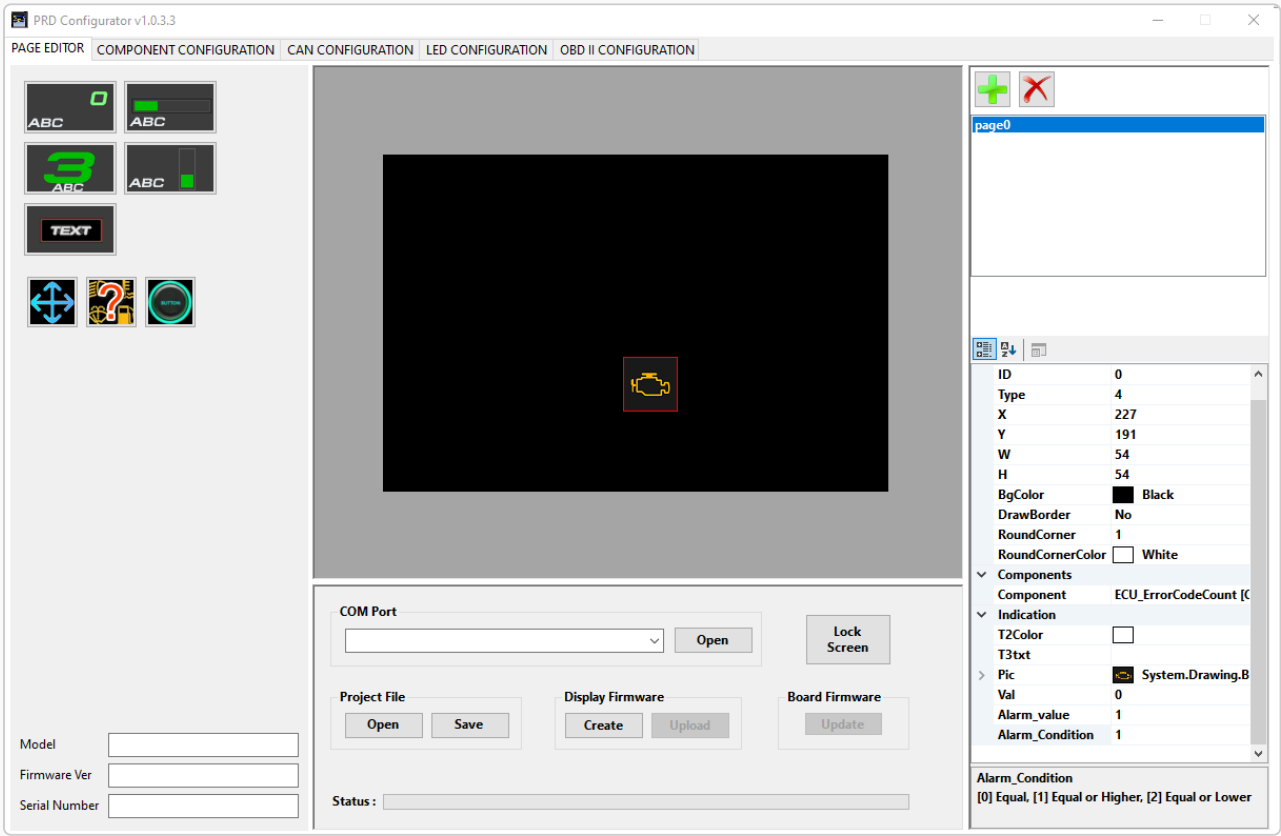
Properties Example:

Parameter	Value	Description
Type	8	Type 8 = Status/alarm indicator

X, Y	191, 122	Position on control panel
W, H	86, 30	Width 86 pixels, Height 30 pixels
Txt	"LC"	Abbreviation for "Launch Control"
Alarm_value	1	Value that triggers alarm
Alarm_Condition	1	Alarm condition (1 = Equal to Alarm_value)

⚠ **Component Connection Essential:** Alarm elements require a linked component to monitor! The component dropdown must be set to the appropriate CAN signal (e.g., LaunchControl [CanBus]) for the alarm to activate when conditions are met. Missing component connection = non-functional alarm.

4.8 Error Indicator



PRD Configurator: Example of Error Indicator configuration with ECU Error Code Count monitoring

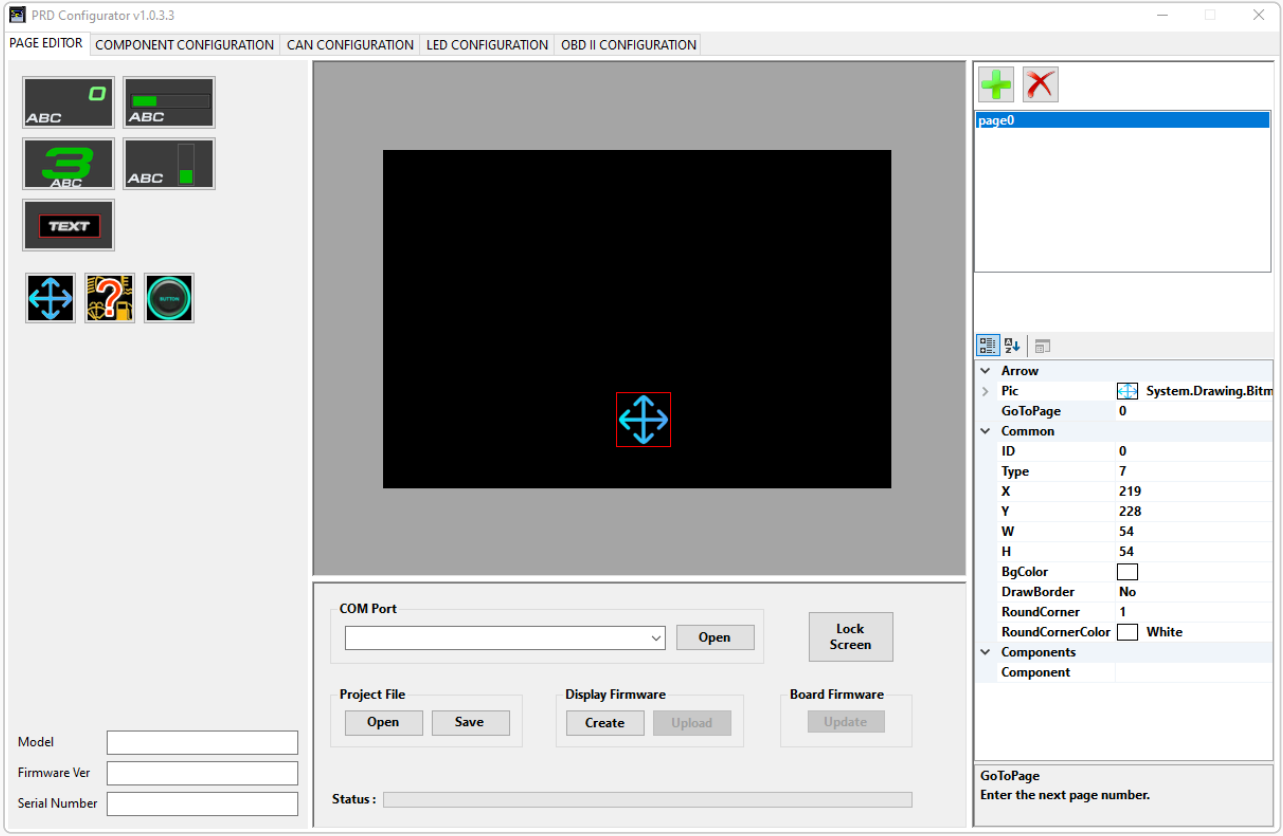
Properties Example:

Parameter	Value	Description

Type	4	Type 4 = Error indicator
X, Y	227, 191	Position on control panel
W, H	54, 54	Square element 54x54 pixels
BgColor	Black	Black background
Components	ECU_ErrorCodeCount [C]	Connection to ECU error counter
Alarm_value	1	Alarm activation threshold

⚠ **Component Monitoring Critical:** Error indicators must be linked to a diagnostic component (e.g., ECU_ErrorCodeCount [C]) to function properly. The "Components" field determines what error condition is being monitored. Without component linkage, error indicator remains inactive.

4.9 Navigation



PRD Configurator: Example of Navigation Arrow configuration for page navigation control

Properties Example:

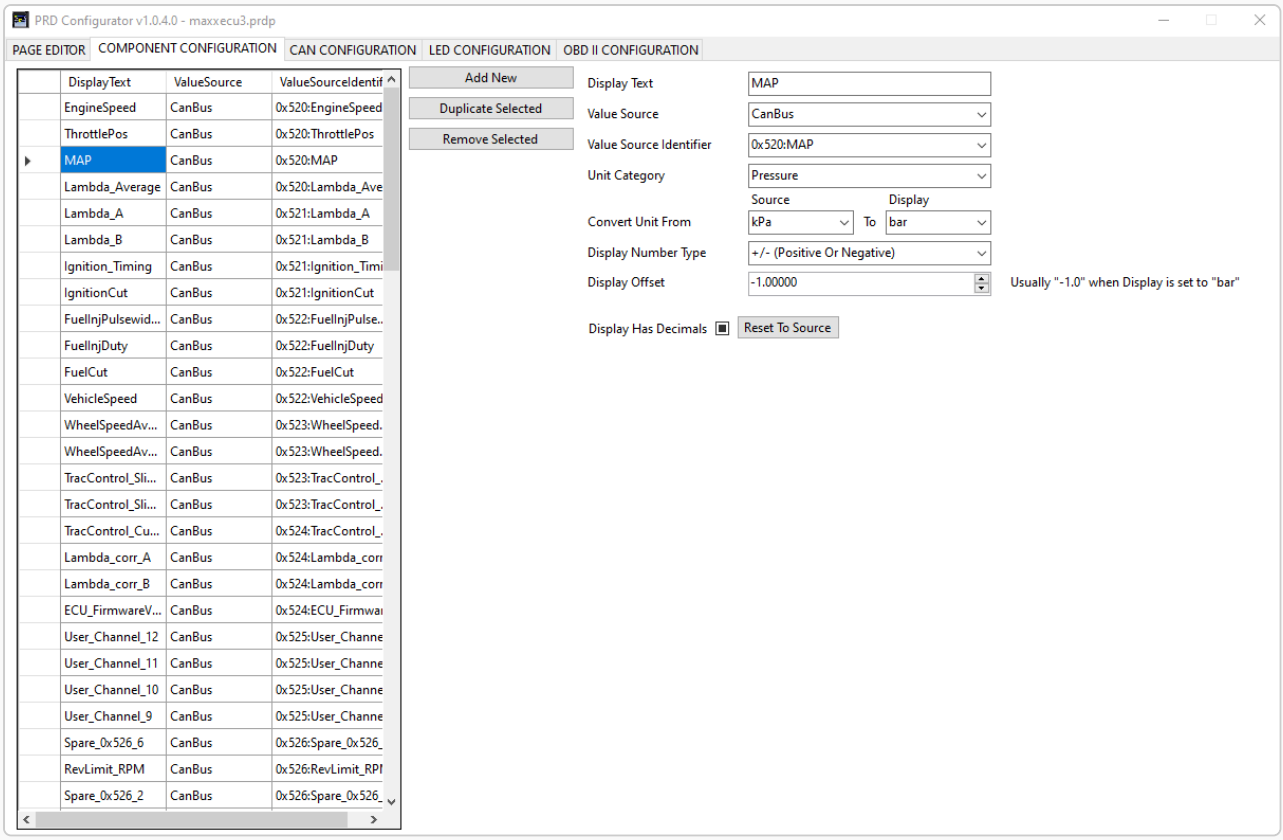
Parameter	Value	Description
Type	7	Type 7 = Navigation arrow
X, Y	219, 228	Position on control panel
W, H	54, 54	Square element 54x54 pixels
GoToPage	0	Destination page (0 = main page)

Navigation Note: Navigation arrow for page/screen changes. Each click leads to the page defined by GoToPage.

4.10 Component Configuration

Important: Components are automatically generated after completing the CAN BUS Setup (Section 5) when you click the **"Generate Components"** button. This creates display-ready components from your CAN signal definitions.

📎 **Connection to Page Elements:** The components you configure here are the same ones that appear in the "Component" dropdown when configuring page elements (Section 4.4-4.9). Proper component configuration ensures your dashboard displays accurate, real-time data.



PRD Configurator: Component Configuration interface showing how to configure display properties for CAN signals

Component Configuration Process:

1. **Complete CAN BUS Setup** (Define messages and signals)
2. **Click "Generate Components"** in CAN Configuration tab
3. **Switch to COMPONENT CONFIGURATION tab**
4. **Configure display properties** for each component

Left Panel - Components List:

Column	Description
DisplayText	Name that appears on the display
ValueSource	Data source (CanBus, Manual, etc.)
ValueSourceIdentifier	Specific CAN signal (e.g., 0x520:MAP)

Right Panel - Component Settings (Example: MAP):

Parameter	Value	Description
Display Text	"MAP"	Label shown on display
Value Source	CanBus	Data comes from CAN Bus

Value Source Identifier	0x520:MAP	Links to specific CAN signal
Unit Category	Pressure	Type of measurement
Convert Unit From/To	kPa → bar	Unit conversion (1 bar = 100 kPa)
Display Number Type	+/- (Positive Or Negative)	Shows positive/negative values
Display Offset	-1.00000	Offset for relative pressure
Display Has Decimals	✓	Shows decimal places (e.g., 1.2 bar)

Data Flow Example (MAP Pressure):

```
1. CAN Signal: 0x520:MAP sends value (e.g., 120)
2. Raw Value: 120 kPa (based on CAN divider settings)
3. Unit Conversion: 120 kPa → 1.2 bar (÷100)
4. Display Offset: 1.2 - 1.0 = 0.2 bar (relative pressure)
5. Final Display: "MAP: 0.2 bar"
```

Control Buttons:

Button	Function
Add New	Creates a new component
Duplicate Selected	Copies the selected component
Remove Selected	Deletes the selected component
Reset To Source	Restores original settings

Common Unit Conversions:

Measurement	Common Conversions	Typical Offset
Pressure (MAP)	kPa → bar	-1.0 (for relative pressure)
Temperature	°C → °F	Sensor dependent
Engine Speed	No conversion	0
Vehicle Speed	km/h → mph	0

Note: Always complete your CAN BUS configuration first, then generate components. Any changes to CAN signals may require regenerating components to maintain proper data linking.

5. CAN BUS Setup

5.1 Introduction to CAN Messages

CAN messages are like data packets exchanged between electronic modules. Each message has a unique ID and contains multiple signals within it.

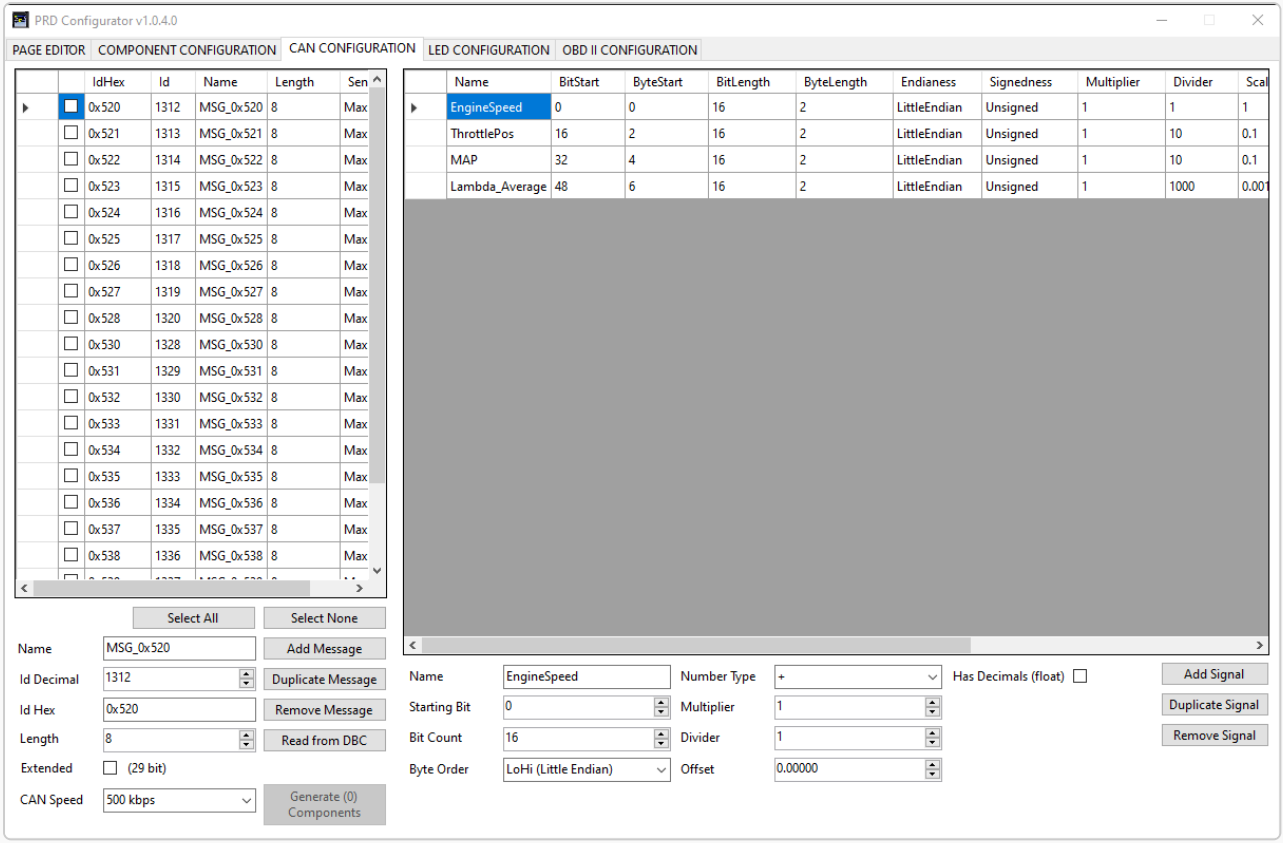
5.2 Creating Messages and Signals

1. First add a new message (Add Message) giving it an ID and length
2. Then add signals within the message (Add Signal) defining:
    - Which bit/byte it starts at
    - How many bits it covers
    - Its type (integer/float)
    - Byte order (endianness)
    - Scale parameters (for value conversion)

5.3 CAN Setup Tips

1. Start with simple messages with few signals
2. Use descriptive names for signals (not just numbers)
3. Consider your hardware's byte order (endianness)
4. For complex systems, load a DBC file first if available

5.4 Final Setup



PRD Configurator: CAN Configuration interface showing message and signal setup with detailed parameters

1. After setting up messages, you can try generating components (Generate Components)
2. Then you can use these components in code to read/write messages
3. Set CAN Speed according to your system

Note: The Starting Bit field expects the bit-level position of the payload value. If you need to specify a Starting Byte instead, multiply the byte index by 8 (since 1 byte = 8 bits) and enter the result in the Starting Bit field. For example, if the value begins at byte 3, the Starting Bit should be set to 24 (3 × 8). This ensures the correct extraction of the payload data.

5.5 PRD Configurator CAN Configuration - Complete Guide

5.5.1 What Does This Tool Do?

The PRD Configurator allows you to define how engine data (RPM, throttle, pressure, etc.) will be "packaged" and sent via CAN bus to your dashboard or other devices.

5.5.2 Basic Concepts

CAN Message

Each CAN message is like a "package" of 8 bytes (64 bits) that contains multiple pieces of information. In the example, message `MSG_0x520` contains 4 different signals.

How Signals "Fit" in the Message:

```
Byte:    0    1    2    3    4    5    6    7
Bit:   0-15  16-31 32-47 48-63
       EngineSpeed  ThrottlePos  MAP  Lambda
```

Signal Configuration Examples

1. EngineSpeed (Engine RPM)

Parameter	Value	Explanation
BitStart	0	Starts from the first bit
BitLength	16	Occupies 16 bits (2 bytes)
Divider	1	The sent value is the actual RPM
Example	3000 RPM → sends 3000	

2. ThrottlePos (Throttle Position)

Parameter	Value	Explanation
BitStart	16	Starts after EngineSpeed
BitLength	16	Occupies 16 bits
Divider	10	Value divided by 10 to save space
Example	85.7% throttle → sends 857 → displays 85.7%	

3. MAP (Manifold Pressure)

Parameter	Value	Explanation
BitStart	32	Starts at 3rd byte
Divider	10	Pressure in mbar divided by 10

Example	1250 mbar → sends 125 → displays 125.0 kPa

4. Lambda_Average

Parameter	Value	Explanation
BitStart	48	Starts at 4th byte
Divider	1000	For 3 decimal places accuracy
Example	λ=0.875 → sends 875 → displays 0.875	

How to Add a New Signal

1. **Select the message** from the left list
2. **Find free space**: See which bits are available
3. **Set BitStart**: First available bit
4. **Choose BitLength**: Based on value range
    - 8 bits = 0-255
    - 16 bits = 0-65535
5. **Set Divider**: To make the value "fit"
6. **Click "Add Signal"**

Button Functions

Message Management Buttons (Left Panel)

Button	Function	Details
Select All	Selects all messages from the list	Useful when you want to perform bulk actions
Select None	Deselects all messages	Clears the selection
Add Message	Creates a new CAN message	Will ask for ID and name for the new message
Duplicate Message	Copies the selected message with all its signals	Creates an identical message with a new ID
Remove Message	**WARNING:** Deletes the message and all signals inside it	All signals within the message are lost
Read from DBC	Imports messages from DBC file (Database CAN)	Loads ready-made configurations from other systems
Generate (0) Components	Automatically creates components for messages	The (0) shows how many components will be created

Signal Management Buttons (Right Panel)

Button	Function	Details
Add Signal	Adds new signal to selected message	Uses settings from the bottom panel
Duplicate Signal	Copies the selected signal	Creates an identical signal that you can modify
Remove Signal	Deletes the selected signal from the message	Frees up the bits it was occupying

Dropdown Menus

Field	Description	Notes
Id Decimal (1312)	Decimal ID of the message (must be unique)	Each message must have a unique ID
Length (8)	Message length in bytes (usually 8 bytes)	Usually 8 bytes for standard CAN
CAN Speed (500 kbps)	CAN bus speed (usually 500 kbps or 1 Mbps)	Usually 500 kbps or 1 Mbps for automotive
Starting Bit (0)	Which bit the signal starts from	Must not overlap with other signals
Bit Count (16)	How many bits the signal occupies	Determines the maximum value range
Byte Order (LoHi)	LoHi = Little Endian (standard for automotive)	Byte order in memory
Multiplier/Divider (1)	Scaling factors for precision and range	Determines precision and range
Number Type	Data type (integer, float, etc.)	Affects data interpretation

Checkboxes

Option	Description	Usage
Extended (29 bit)	For extended CAN IDs (29-bit instead of 11-bit)	Rarely used in automotive
Has Decimals (float)	If the signal has decimal places	Affects display on dashboard

Practical Tips

Important Guidelines:

- **Always backup** before deleting messages
- **Check BitStart values** to avoid signal overlap
- **Test settings** in test mode before final application
- **Choose the correct endianess** that works with your ECU
- **Use Unsigned** for positive values (RPM, pressure, etc.)
- **Higher Divider** = less precision but larger range

Additional Best Practices:

Signal Configuration Tips:

- **Little Endian:** Leave it as is (standard for automotive)
- **Unsigned:** Use for positive values (RPM, pressure, etc.)
- **Divider:** The larger the divider, the smaller the precision but larger the range
- **Bit Planning:** Plan your bit allocation carefully to avoid overlaps
- **Testing:** Always test in a safe environment before deployment

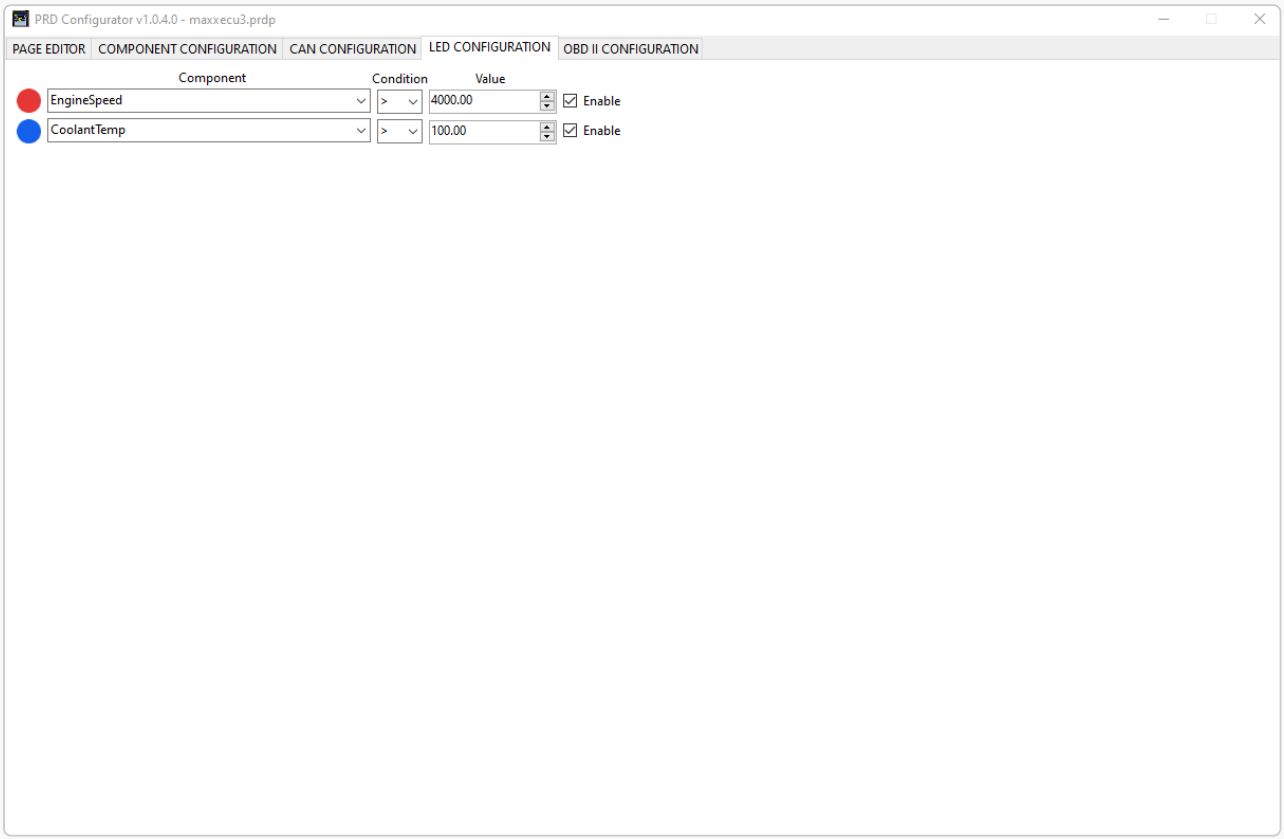
Calculation Example

Example: Sending water temperature 0-150°C with 0.5°C precision:

- Range: 0-300 (x2 for 0.5°C precision)
- Need: 9 bits (512 > 300)
- Divider: 2
- 87.5°C → send 175 → display 87.5°C

5.6 LED Configuration

Purpose: The LED Configuration tab allows you to create visual alerts using colored LEDs that activate when specific conditions are met from your CAN Bus components. This provides instant visual feedback for critical engine parameters.



PRD Configurator: LED Configuration showing setup for RPM and temperature-based LED alerts

How LED Alerts Work:

Each LED can be configured to monitor any CAN Bus component and activate when specific conditions are met. The system supports multiple LED colors with individual trigger conditions.

Configuration Parameters:

Parameter	Description	Options
Component	Select which CAN signal to monitor	Any generated component (EngineSpeed, CoolantTemp, etc.)
Condition	Logic operator for comparison	< (less than), > (greater than), = (equal), ≤, ≥

Value	Threshold value for activation	Numeric value in component's units
Enable	Checkbox to activate/deactivate the alert	✓ Enabled / ✗ Disabled

Red LED - High RPM Alert

Setting	Value	Purpose
Component	EngineSpeed	Monitor engine RPM
Condition	> (greater than)	Activate when RPM exceeds threshold
Value	4000.00 RPM	Shift point or high RPM warning
Function	**Shift Light:** Alerts driver to change gear at 4000+ RPM	

Blue LED - Cold Engine Alert

Setting	Value	Purpose
Component	CoolantTemp	Monitor coolant temperature
Condition	< (less than)	Activate when temperature is below threshold
Value	100.00°C	Engine warm-up indicator
Function	**Warm-up Warning:** Indicates engine hasn't reached operating temperature	

Practical LED Setup Examples:

Racing/Track Configuration:

LED Color	Component	Condition	Value	Purpose
Red	CoolantTemp	>	105°C	Critical overheating alert
Blue	EngineSpeed	>	6500 RPM	Redline warning/shift point

Daily Driving Configuration:

LED Color	Component	Condition	Value	Purpose
Red	CoolantTemp	>	95°C	High temperature warning
Blue	CoolantTemp	<	80°C	Engine still warming up

Alternative 2-LED Configurations:

Option 1: Temperature Focus

LED Color	Component	Condition	Value	Purpose
Red	CoolantTemp	>	95°C	Overheating danger
Blue	CoolantTemp	<	70°C	Cold engine warning

Option 2: RPM & Temperature

LED Color	Component	Condition	Value	Purpose
Red	EngineSpeed	>	6000 RPM	High RPM warning
Blue	CoolantTemp	<	80°C	Engine warm-up phase

Option 3: Performance Focus

LED Color	Component	Condition	Value	Purpose
Red	MAP	>	1.2 bar	High boost pressure
Blue	EngineSpeed	>	4000 RPM	Shift point indicator

Advanced LED Setups:

Two-Stage RPM Alerts:

```
Blue LED:  EngineSpeed > 4000 RPM (Shift suggestion)
Red LED:   EngineSpeed > 6500 RPM (Redline warning)
```

Temperature Monitoring (2-LED System):

```
Blue LED:  CoolantTemp < 80°C  (Cold engine - drive gently)
Red LED:   CoolantTemp > 100°C (Overheating - stop immediately)
```

Configuration Best Practices:

Important Guidelines for 2-LED System:

- **Red LED:** Always use for critical warnings (overheating, over-revving)
- **Blue LED:** Use for information or less critical alerts
- **Prioritize Safety:** Configure red LED for the most dangerous condition
- **Test Thoroughly:** Verify LED activation at safe levels first
- **Clear Purpose:** Each LED should have a distinct, understandable function

LED Hardware Notes:

Technical Information:

- **LED Types:** The system supports standard automotive LEDs
- **Current Rating:** Check LED current requirements against system output
- **Response Time:** LEDs activate immediately when conditions are met
- **Brightness:** LEDs are designed for clear visibility in daylight conditions
- **Durability:** LEDs are rated for automotive temperature and vibration conditions

Troubleshooting LED Configuration:

Problem	Possible Cause	Solution
LED doesn't activate	Component not receiving data	Check CAN Bus connection and component configuration
LED always on	Condition threshold too low	Adjust threshold value or change condition operator
LED flickers	Signal noise or boundary condition	Add hysteresis or adjust threshold slightly
Multiple LEDs conflict	Overlapping conditions	Review and separate threshold ranges

6. PRD Configurator - Detailed Instructions

6.1 Overview

This guide explains how to use PRD Configurator for setting up TFT display systems. The tool allows project management, firmware uploading, and display configuration.

6.2 Control Elements & Information

COM Port

Function	Description
Open	Opens serial connection (COM) to device
Lock Screen	Locks screen during configuration
Project File	Loads or saves project file (prdp)

Module

Information	Description
Firmware Ver	Displays firmware version
Serial Number	Unique device serial number

6.3 Setup Steps

1. Click **Create**
2. Connect via COM Port (click **Open**)
3. Click **Upload** to send changes

7. Firmware Update

Important: The entire process is done via USB-C and the PrdConfigurator application. After major changes to GT_X.X_Board_Firmware, the GT_X.X display may not show image or expected environment.

7.1 Recovery Mode Procedure

For emergency:

1. Find the gray recovery wire (usually in the display cable, which needs to be connected to GND or the USB adapter included in the package)
2. Hold down the recovery button (or connect the wire) until the new firmware upload process starts

Maintenance and Support

Regular Maintenance

- Check wiring and connections
- Clean screen with appropriate materials
- Test touch screen functionality
- Update firmware when available

Troubleshooting

Problem	Solution
Screen does not turn on	Check power supply and wiring
Touch not responding	Clean screen, restart
CAN communication error	Check CAN_H and CAN_L wires
Firmware corruption	Use Recovery Mode with gray wire
Display interface appears distorted or incorrectly sized	Adjust Windows display scaling to 100% (see detailed instructions below)

Windows Display Scaling Issue Resolution

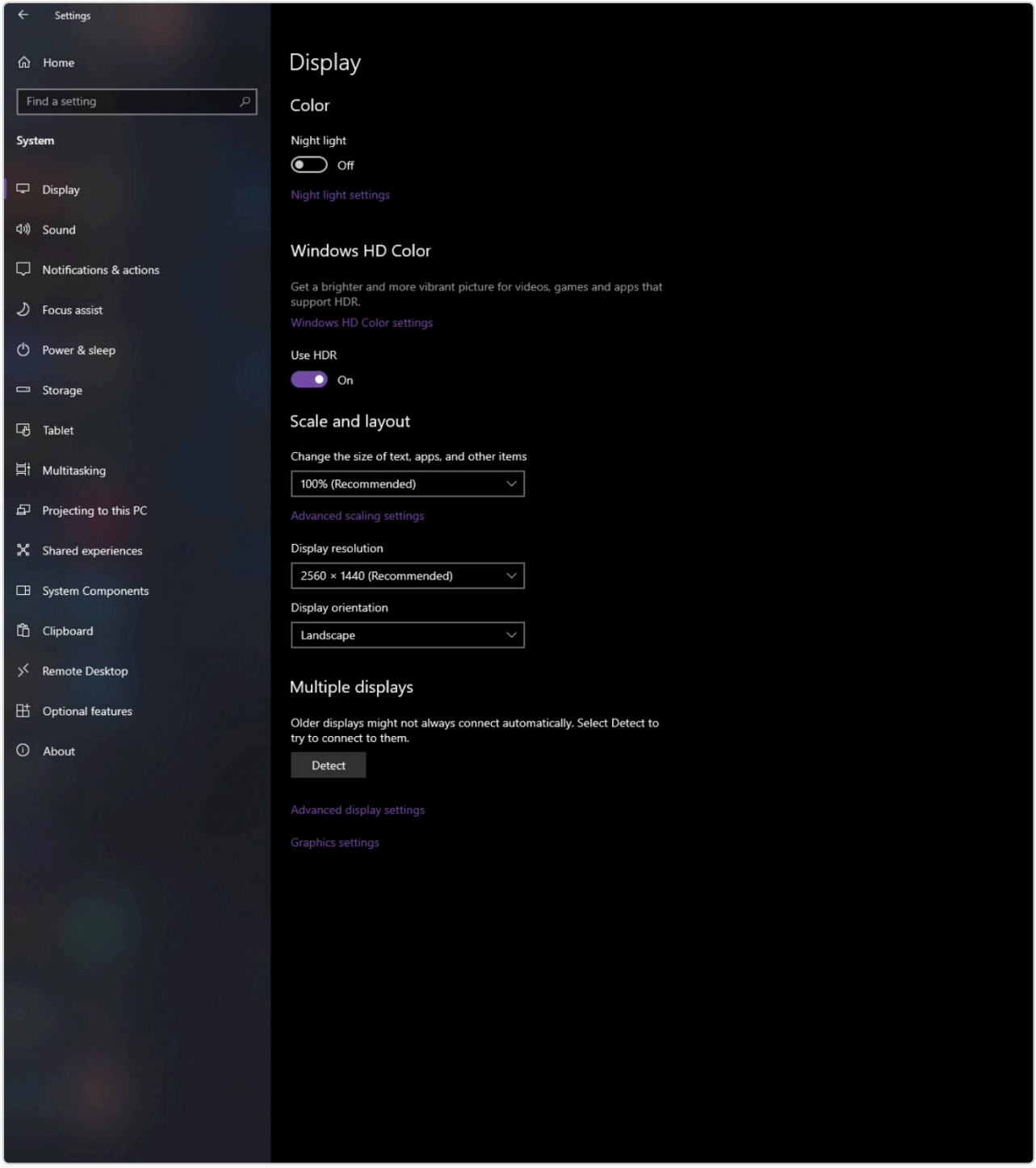
⚠ **Common Issue:** Interface elements appear oversized, overlapping, or positioned incorrectly in the PRD Dash GT.

Root Cause: When the PRD GT interface appears distorted, with elements appearing too large or incorrectly positioned, this is typically caused by Windows display scaling settings being set above 100% when creating a project in PRD Configurator. The PRD configurator is designed to work optimally at 100% scaling.

⚠ ***Problem Example: Interface distortion when Windows scaling is not set to 100%***

Solution Steps:

1. Right-click on the Windows desktop and select **"Display settings"**
2. Alternatively, go to **Windows Settings → System → Display**
3. In the **"Scale and layout"** section, locate the scaling dropdown menu
4. Change the scaling setting from the current value (e.g., 125%, 150%) to **"100% (Recommended)"**
5. Windows may prompt you to sign out and back in for the changes to take full effect
6. Restart the PRD Configurator application to ensure proper display scaling



Windows Display Settings: Navigate to Scale and layout section and set scaling to 100%

Important Note: This scaling adjustment affects the entire Windows display. If you prefer to keep your system scaling at a higher percentage for general use, you may need to temporarily change it to 100% when using the PRD Configurator, then change it back afterward.

Technical Background: The PRD Configurator interface is optimized for 100% display scaling. Higher scaling percentages can cause interface

elements to overlap, become misaligned, or appear outside the visible area, making the application difficult or impossible to use effectively.

Note: For additional support and technical assistance, contact PRD support team. Always keep backups of your settings before any updates.