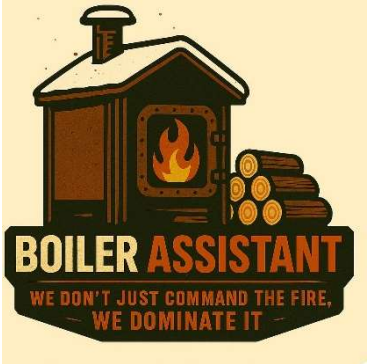

Boiler Assistant

Project Support & Open-Source Notice

The Boiler Assistant project is fully **open-source**, created to give users control, transparency, and the freedom to customize their own system. Every improvement, feature, and firmware update is built with the community in mind.

If you find this project useful and want to support future development, upgrades, and continued maintenance, **donations are appreciated but never required**. Contributions help cover testing hardware, replacement components, and the time invested in ongoing improvements.

Donations (PayPal or e-Transfer):

PayPal (CANADA and USA)

E-TRANSFER (CANADA ONLY!!)

PRAIRIEWILLOW@GMAIL.COM

Thank you for helping keep the project alive and evolving.

Open-Source Use & Liability Disclaimer

This controller and its firmware are provided as an **open-source project**, offered in good faith for hobbyists, DIY builders, and technically capable users. Because every installation, boiler system, and operating environment is different, it is the responsibility of the user to determine whether this system is appropriate for their specific application.

The creator(s) of this project assumes **no liability** for:

- Damage to equipment
- Improper installation
- Misuse or unsafe operation
- Any direct or indirect consequences of using this firmware or hardware

By using this system, you acknowledge that you are doing so **at your own risk**, and you accept full responsibility for verifying that it is safe and suitable for your setup.

1. Introduction

Boiler Assistant v5.0 is an exhaust temperature driven combustion controller designed for outdoor wood boilers. It provides:

• Adaptive burn control

• PID burn control

• Boost mode for cold starts

• Clamp limits for fan control

• Optional hard off fan relay mode

• Real time diagnostics

• Installer friendly UI with keypad navigation

• EEPROM backed settings

The system reads exhaust temperature via a MAX31855 thermocouple, then applies logic, and drives a PWM fan controller and relay controlled damper to stabilize combustion.

The UI is fully menu driven and allows editing all major parameters without a computer.

2. Installation Guide

This section walks through downloading the firmware, preparing the Arduino IDE, wiring the hardware, and verifying operation.

2.1 Downloading the Firmware from GitHub

1. Open the project's GitHub page.
2. Click **Code → Download ZIP**.
3. Extract the ZIP file.
4. Open the folder.
5. Open the `.ino` file in Arduino IDE.

2.2 Required Arduino Libraries

Install these libraries via **Tools → Manage Libraries**:

• LiquidCrystal_PCF8574 • Adafruit MAX31855 • EEPROM (built in) • Wire (built in)

No WiFi libraries are required for v1.0

2.3 Arduino IDE Setup

1. Go to **Tools → Board → Arduino Renesas → Arduino UNO R4 WiFi**.
2. Set **Upload Speed: Default**.
3. Select the correct COM port.
4. Click **Verify** to compile.
5. Click **Upload** to flash the firmware.

If after wiring it up the LCD initializes and the boot animation appears, the upload succeeded.

3. User Interface & Navigation

Boiler Assistant v1.0 uses a 4×4 matrix keypad and a 20×4 LCD. Every setting is edited directly from the UI — no computer required.

Keypad Behavior

• Numbers (0–9): Enter values

• A/B/C/D: Navigate between major menus

• * = Cancel / Back • # = Save

UI Flow Overview

• A → Setpoint

• B → Burn Logic

• C → PID Profiles

• D → System Settings

The UI is state machine driven, meaning every screen is deterministic and predictable.

4. Full UI + Settings

This section documents every screen, every setting, and what it does.

4.1 Home Screen

Displays real time system status.

Line	Display	Meaning
1	Exh Set: ###F	Exhaust temperature target
2	Exh Cur: ###F	Current exhaust temperature
3	Fan: ###%	Current fan output
4	BOOSTING or Mode: ADAPTIVE or PID	Current burn mode

From here:

• Press **A** → Exhaust Temperature Setpoint

• Press **B** → Burn Logic Settings and Clamp Settings

• Press **C** → PID Profiles and Tuning

• Press **D** → System Menu

4.2 Setpoint Menu (A)

Purpose: Sets the exhaust temperature target for the burn controller.

Range: 200°F – 999°F *A good starting point is 550°F but will depend on your boiler and probe placement.*

```
EXHAUST SET POINT
Current: ###
New: ___
*=CANCEL   #=SAVE
```

Notes

• This is the primary control variable for both Adaptive and PID modes.

• Higher setpoints = hotter, more aggressive burn.

4.3 Burn Logic Menu (B)

This menu selects the burn strategy and provides access to Boost Time.

```
BURN LOGIC MODE
1: ADAPTIVE  <
2: PID       <
3: BOOST TIME
#=SAVE
```

1. Adaptive Mode (press 1 to select)

• Uses a dynamic slope to adjust fan output.

• Learns from the last temperature rise/fall rate.

• Best for general use and mixed fuel quality.

2. PID Mode (press 2 to select)

• Uses PID control inside the deadband.

• More stable but requires tuning.

• Best for consistent, high-quality fuel.

3. Boost Time

• Opens a sub-menu to set cold-start boost duration.

Save Behavior: Press # to save selected mode Press * to cancel and go back

4.4 Boost Time Menu

Purpose: Controls how long the fan runs at 100% after startup.

Range: 10–300 seconds

```
BOOST TIME (sec)
Current: ###
New: ___
*=CANCEL   #=SAVE
```

• Longer boost helps cold, wet, or dense fuel, but too long can overshoot the setpoint.

4.5 System Menu (D)

```
SYSTEM SETTINGS
1: DEADBAND
2: ADAPTIVE DIAG
3: CLAMP
*=BACK
```

This menu contains all system-level settings.

4.6 Deadband Menu

Purpose: Defines the “no action zone” around the setpoint.

Range: 10–200°F

```
DEADBAND (F)
Current: ###
New: ___
*=CANCEL   #=SAVE
```

Behavior

• Below (setpoint – deadband) → fan ramps up

• Above (setpoint + deadband) → fan ramps down

• Inside → Adaptive or PID logic applies

Notes

• Larger deadband = smoother but slower response

• Smaller deadband = tighter control but more fan movement

4.7 Clamp Menu

This menu controls fan output limits and the new hard-off mode.

```
CLAMP SETTINGS
1: Min: ###%
   Max: ###%
4: Dead zone Mode<ON/OFF>
```

Clamp Min %

Minimum fan output when fan is running. **It is NOT recommended going below 40%.**

Range: 0–99% *(Do not use 100% as it needs to be patched right now.)*

• Prevents fan from running too weakly to matter. • If Min > Max, Max is automatically raised.

Clamp Max %

Maximum fan output allowed.

Range: 0–100% *(Do not use 100% as it needs to be patched right now.)*

• Limits peak burn intensity.

Fan Low Off Mode (Press 4 to toggle ON/OFF)

Purpose: Allows true fan hard-off using the relay. *(Mainly used if you have a powered damper.)*

Behavior

• If fanPercent < clampMin → relay turns OFF

• If exhaust temp drops below (setpoint – deadband) → relay turns ON

• When relay is OFF → fanPercent shows 0

Why it matters

• Prevents smoldering

• Reduces creosote

• Allows a real “dead zone” with no airflow from the fan and just uses natural draft

4.8 PID Profile Menu (C)

```
PID TUNING
1: BELOW
2: NORMAL
3: ABOVE
*=BACK
```

Purpose: Each profile has its own KP/KI/KD values.

• **BELOW:** Used when exhaust is below the deadband • **NORMAL:** Used inside the deadband • **ABOVE:** Used when exhaust is above the deadband

This allows different PID behavior depending on burn phase.

4.9 PID Parameter Menu

After selecting a profile:

Code

```
PID NORMAL
1:KP 1.000
2:KI 0.050
3:KD 0.000 *=BACK
```

(There is a full write-up in the Facebook group under Files as to how to tune the PID settings.)

4.10 Adaptive Diagnostics

```
ADAPTIVE LEARNING
Slope: 1.00
dT/ds: 0.023
*=BACK   #=RESET
```

Purpose: Shows internal adaptive learning variables.

• **Slope:** The adaptive multiplier (0.5–2.0)

• **dT/ds:** Temperature rise rate (°F per second)

Reset

Press **#** to reset slope to **1.0**.

5. Wiring Guide (Refer to the pinout.h in zip file)

5.1 LCD Wiring (I2C)

The LCD uses address **0x27** and must be wired to:

LCD Pin	UNO R4 Pin
SDA	SDA on the hat or board
SCL	SCL on the hat or board
VCC	5V
GND	GND

5.2 MAX31855 Thermocouple Module

MAX31855	UNO R4 Pin
SCK	PIN_MAX31855_SCK
CS	PIN_TC1_CS
MISO	PIN_MAX31855_MISO
VCC	3.3V
GND	GND

5.3 Fan PWM Output

This pin outputs a 0–255 PWM signal mapped from 0–100% fan demand.

Wire **FAN_PIN** to your fan controller's PWM input.

5.4 Fan Relay (Hard Off Mode)

• Relay energized (ON) when pin is **LOW**

• Relay off when pin is **HIGH**

This relay is used for true fan hard-off when Fan Low Off Mode is enabled.

5.5 Damper Relay

This relay is always driven **LOW** (open damper) in v5.0.

5.6 Keypad Wiring

Your keypad uses:

- Rows: **R1, R2, R3, R4**
- Cols: **C1, C2, C3, C4**

All mapped in **Pinout.h**.

The keypad engine uses:

• Internal pull-ups on columns

• Row scanning with debouncing

• 40ms stable press detection

No external resistors required.

6. First Boot Verification

When powered, the LCD shows the Boiler Assistant splash screen.

1. A progress bar animates.
2. A spinner animation runs.
3. A final message appears: **“We don’t just command fire — WE DOMINATE IT!”**
4. Home screen appears with:

    • Exhaust Setpoint

    • Current Exhaust Temp

    • Fan %

    • Mode indicator

If you see all of this, the system is fully operational if you wired it correctly.

7. Control Logic

Boiler Assistant v5.0 runs a deterministic, once-per-second control loop. Every second, the system:

1. Reads the exhaust temperature
2. Updates adaptive learning
3. Applies smoothing
4. Computes fan output
5. Applies clamp rules
6. Applies relay hard off logic (if enabled)
7. Updates the UI

7.1 Sensor Reading (MAX31855)

The MAX31855 is read once per second using a cached function:

• If the sensor returns a valid reading → convert °C to °F

• If the sensor returns NaN → keep the last known value

• If no valid reading has ever been received → display "----F"

7.2 Temperature Smoothing

Two layers of smoothing are applied:

1. Display Smoothing (3%)

2. Display Deadband (2°F)

The temperature shown on screen only updates if it changes by **2°F or more**. This prevents the LCD from flickering between adjacent values.

7.3 Adaptive Learning Engine

Adaptive mode continuously adjusts the **adaptiveSlope** value between **0.5 and 2.0**.

Every second:

• If the boiler is below setpoint and temperature is rising too slowly → slope increases

• If the boiler is above setpoint and temperature is rising too quickly → slope decreases

This allows the controller to “learn” your boiler’s behavior.

Resetting Adaptive Learning

Press **#** in the Adaptive Diagnostics screen to reset slope to **1.0**.

7.4 Burn Phases

The system has two burn phases:

1. BOOST Phase

• Starts at power on • Fan runs at 100%

• Lasts for **boostTimeSeconds** (10–300s)

• Used to ignite the fire and establish draft

2. ADAPTIVE Phase

After BOOST ends, the system switches to either:

• Adaptive Mode (**burnLogicMode = 0**)

• PID Mode (**burnLogicMode = 1**)

7.5 Deadband Behavior

Deadband defines a “no action zone” around the setpoint:

• Below (setpoint – deadband) → fan ramps up

• Above (setpoint + deadband) → fan ramps down

• Inside → Adaptive or PID logic applies

This prevents oscillation and stabilizes the burn.

7.6 Adaptive Fan Control

Inside the deadband, Adaptive Mode computes fan output based on:

• Distance from the low band

• Adaptive slope multiplier

The formula produces a smooth, proportional fan curve.

Outside the deadband

• Below low band → fan = 100% • Above high band → fan = 0%

7.7 PID Fan Control

Inside the deadband, PID Mode uses:

• KP • KI • KD

from the **NORMAL** profile.

Outside the deadband

• Below low band → fan = 100% and integral resets

• Above high band → fan = 0% and integral resets

This prevents integral wind-up.

7.8 Clamp Logic

Clamp limits apply **after** Adaptive/PID output is computed.

Clamp Max

If fanPercent > clampMax → fanPercent = clampMax

Clamp Min

If fanPercent is between 1% and clampMin → fanPercent = clampMin *(Only when Fan Low Off Mode is OFF)*

7.9 Deadzone Mode (Relay Hard Off)

When enabled:

• If fanPercent < clampMin → relay turns OFF

• If exhaust temp drops below (setpoint – deadband) → relay turns ON

Safety Rule: 60 Second Minimum Interval

The relay cannot toggle more than once per minute. This prevents relay chatter and extends hardware life.

When relay is OFF

• Fan PWM is still computed internally

• But the UI forces fanPercent = 0

• PWM output is irrelevant because the relay cuts power

7.10 PWM Output

After all logic is applied, the PWM output drives the fan smoothly and proportionally.

7.11 Fan Display Smoothing

The fan percentage shown on the LCD is smoothed. This prevents the UI from jumping around.

7.12 Safety Outputs

The damper relay is always driven **LOW** in v5.0.

7.13 UI Update

At the end of each 1-second cycle, the UI refreshes and is always consistent with internal state.

7.14 Summary of Control Loop Timing

Every second:

1. Read exhaust temp
2. Update adaptive learning
3. Smooth temperature
4. Compute fan output
5. Apply clamp limits
6. Apply relay hard off logic
7. Smooth fan display
8. Update UI

This ensures stable, predictable behavior.

8. Troubleshooting Guide

This section covers the most common issues encountered during installation, wiring, or operation of Boiler Assistant v1.0.

Each issue includes:

• Symptoms

• Likely Causes

• How to Diagnose • How to Fix It

8.1 LCD Does Not Turn On

Symptoms

• LCD is completely blank

• Backlight may or may not be on

• No boot animation

Likely Causes

• Incorrect I2C wiring

• Wrong LCD address

• No 5V power

• SDA/SCL reversed

Diagnosis

1. Confirm LCD is wired to **A4 (SDA)** and **A5 (SCL)**.
2. Confirm LCD has **5V and GND**.
3. Check contrast pot on the LCD backpack.
4. Ensure the board is set to **UNO R4 WiFi** in Arduino IDE.

Fix

• Correct wiring • Adjust contrast • Replace LCD if backlight is dead

8.2 LCD Shows Boxes or Garbage Characters

Symptoms

• First row shows solid blocks

• Text is unreadable

Likely Causes

• LCD not initialized

• Wrong I2C address (should be **0x27**)

• Loose SDA/SCL wires

Fix

• Re-seat SDA/SCL

• Confirm address is **0x27**

• Power cycle the system

8.3 Exhaust Temperature Shows “----F”

Symptoms

• Home screen shows: `Exh Cur: ----F`

Likely Causes

• MAX31855 not wired

• Wrong voltage (must be **3.3V**)

• Thermocouple reversed

• Broken thermocouple

Diagnosis

1. Check MAX31855 wiring:

   • VCC → 3.3V • GND → GND

   • SCK → PIN_MAX31855_SCK

   • CS → PIN_TC1_CS • MISO → PIN_MAX31855_MISO

2. Check thermocouple polarity (yellow = +, red = –).

Fix • Correct wiring • Replace thermocouple if damaged

8.4 Fan Always Reads 0%

Symptoms

• Fan display stuck at 0% • Fan does not spin

• Relay may or may not click

Likely Causes

• Fan Low Off Mode enabled

• Relay is OFF due to deadband

• PWM pin not wired

• clampMin = 0

• Exhaust temp above high band

Diagnosis

1. Check Clamp Menu → Deadzone Mode.
2. Check if exhaust temp is above (setpoint + deadband).
3. Check PWM wiring.
4. Check relay wiring (active-LOW).

Fix

• Disable Fan Low Off Mode if not desired • Lower setpoint or deadband • Rewire PWM or relay

8.5 Fan Always Runs at 100%

Symptoms

• Fan stuck at full speed

• BOOST never ends

• PID/Adaptive never engage

Likely Causes

• BOOST time set too high

• Exhaust temp sensor not reading

• Exhaust temp below low band

• PID integral stuck (rare)

Diagnosis

1. Check BOOST time (should be 10–300s).
2. Check exhaust temperature reading.
3. Verify MAX31855 wiring.

Fix

• Reduce BOOST time

• Fix sensor wiring

• Reset adaptive slope (optional)

8.6 Relay Clicks Too Often

Symptoms

• Audible clicking

• Fan turning on/off frequently

Likely Causes

• Deadband too small

• clampMin too high

• Fan Low Off Mode toggling

Diagnosis

1. Increase deadband (recommended **80–120°F**).
2. Reduce clampMin.
3. Check if relay is obeying the **60-second minimum interval**.

Fix

• Increase deadband

• Adjust clamp settings

• Disable Fan Low Off Mode if not needed

8.7 Keypad Not Responding

Symptoms

• No keypresses register

• Only some keys work

• Random characters appear

Likely Causes

• Rows/columns wired incorrectly

• Loose ground

• Wrong keypad type

• Damaged keypad

Diagnosis

1. Confirm wiring matches **Pinout.h** exactly.
2. Check continuity on keypad ribbon cable.
3. Ensure all 8 wires are connected.

Fix

• Rewire keypad

• Replace keypad if membrane is damaged

8.8 EEPROM Not Saving Settings

Symptoms

• Settings revert after reboot

• Setpoint resets to default

• PID values revert

Likely Causes

• EEPROM writes failing

• Power loss during save

• Corrupted EEPROM (rare)

Diagnosis

1. Change a setting → reboot → check if saved.
2. Try multiple settings.

Fix

• Reflash firmware

8.9 Fan Relay Always ON or Always OFF

Symptoms

• Fan never shuts off

• Fan never turns on

• Relay LED stuck

Likely Causes

• Relay wired backwards

• Active-LOW logic misunderstood

• Fan Low Off Mode enabled

Diagnosis

1. Relay ON = pin **LOW**
2. Relay OFF = pin **HIGH**
3. Check Fan Low Off Mode

Fix

• Reverse relay logic wiring • Disable Fan Low Off Mode • Replace relay module if stuck

8.10 Temperature Flickers or Jumps

Symptoms

• Display jumps several degrees

• Fan output fluctuates

Likely Causes

• Loose thermocouple

• Electrical noise

• MAX31855 grounding issue

Fix

• Tighten thermocouple screws

• Add shielding

• Ensure MAX31855 shares ground with Arduino

8.11 System Reboots Randomly

Symptoms

• LCD resets

• Boot animation appears unexpectedly

Likely Causes

• Power supply sag

• Relay back-EMF

• Loose barrel jack

Fix

• Use a stable 5V supply

• Add flyback diode if using non-isolated relays

• Secure power connections