Detector Energy Calibration Tool

Comprehensive User Manual

Author: Siddharth Parashari

Version 2.0 — 2026

What's New in Version 2.0

Version 2.0 introduces three major enhancements to the peak detection pipeline and a new automated calibration workflow for multi-channel systems:

Sliding Window Detection	A new algorithm that scans the spectrum with a configurable ADC window, finding local maxima. Complementary to TSpectrum — useful for closely-spaced peaks or spectra where TSpectrum over-resolves.
Gaussian Peak Confirmation	After any detection method runs, each candidate peak is confirmed by fitting a Gaussian + linear background. Accepted peaks are refined to the Gaussian centroid; rejected peaks are shown as red × markers on the spectrum.
Detect + Fit All Channels	A single button (■■ Detect + Fit All) now runs peak detection on every loaded channel, propagates energy assignments from the current channel, and fits the calibration model — all in a background thread.
Calibrated Spectrum Fix	The calibrated spectrum display now shows the full ADC range rather than clipping at the last calibration peak (adc_max). Nonlinear model extrapolation is capped at 3× the energy at adc_max to prevent blow-up.

1. Overview

DetectorCalibGUI is a cross-platform Python application for performing energy calibration of multi-channel particle and radiation detector systems. It supports PET scanners, Compton cameras, and general-purpose gamma-ray detector arrays. The tool reads raw ROOT data files, detects or accepts user-defined calibration peaks, fits a calibration model, and exports the resulting coefficients for downstream analysis.

1.1 Key Features

- **Dual ROOT backend:** PyROOT (native) or uproot (pure Python) — detected automatically.
- **Flexible data loading:** TTree filter mode, array-branch mode, custom Draw() expressions, or pre-filled TH1 histograms.
- **Three peak detection algorithms:** ROOT TSpectrum (standard and HighRes), Sliding Window, and SciPy fallback — all with optional Gaussian confirmation.
- **Gaussian peak confirmation:** Each detected candidate is verified by a Gaussian fit. Accepted peaks are centroid-refined; rejected peaks are shown as red × markers.
- **Peak propagation:** Detect peaks on one reference channel and automatically search all other channels within a configurable ADC window.
- **Three calibration models:** Linear, nonlinear scintillator, 3-point nonlinear variant, plus user-defined custom expressions.
- **Detect + Fit All:** One-click workflow — detect, propagate, and fit all channels simultaneously in a background thread.
- **Calibrated spectrum viewer:** Full-range energy spectra using session or file-loaded coefficients.
- **Energy resolution analysis:** Interactive Gaussian fitting with multiple background models, FWHM extraction, and per-channel trend plots.
- **Rich output:** Human-readable calibration log, compact coefficients table, and resolution results file.

1.2 Application Tabs at a Glance

#	Tab Name	Purpose
1	■ Single Channel	Spectrum view, peak detection & assignment, single-channel calibration fit
2	■ Overview Grid	All-channel thumbnail grid colour-coded by fit status
3	■ Coefficient Trends	Calibration parameters vs channel ID with error bars
4	■ Calibrated Spectrum	Full-range energy-calibrated spectra; send to resolution analysis
5	■ Resolution	Interactive Gaussian peak fitting, FWHM extraction, trend plots

2. Installation & Requirements

2.1 Python Dependencies

```
pip install PyQt5 matplotlib numpy scipy
```

2.2 ROOT Backend (choose one)

Option A — PyROOT (recommended)

```
source /path/to/root/bin/thisroot.sh
python3 main.py
```

On macOS with Homebrew:

```
source $(brew --prefix root)/bin/thisroot.sh
```

The application auto-detects ROOT even if thisroot.sh was not sourced, by searching ROOTSYS, root-config --prefix, and common directories such as /opt/homebrew, /usr/local, and ~/root.

Option B — uproot (no ROOT installation required)

```
pip install uproot awkward
python3 main.py
```

■ ***NOTE:*** *The active backend is displayed in the status bar after loading a file: [PYROOT backend] or [UPROOT backend]. TSpectrum detection requires PyROOT; scipy find_peaks is used automatically when uproot is active.*

2.3 Launching

```
cd DetectorCalibGUI
python3 main.py
```

3. Loading Data

3.1 Opening a ROOT File

Click ■ Open ROOT File in the toolbar. A file dialog opens; select any .root file. The application probes the file and counts available TTrees and TH1 histograms, then opens the Configure Data Source dialog.

3.2 Data Mode

- **TTree mode:** Raw event-level data stored in a TTree. The application reads branches and builds histograms per channel.
- **TH1 Histogram mode:** Pre-filled TH1F/TH1D histograms, one per channel. Channel ID is parsed from trailing digits in the histogram name (e.g. spectrum_ch0042 → channel 42).

3.3 TTree Draw Modes

Mode	ROOT equivalent	When to use
Filter	`Draw("energy","channelID==N")`	Standard layout: separate ADC and channel-ID branches. Channel IDs auto-discovered.
Array	`Draw("energy[N]")`	Array branch — each event stores a value per channel. Requires channel range.
Custom	`Draw("expr, cut") with %d`	Full control. Use %d as a placeholder for the channel number.

3.4 Channel Range & Entry Limits

Control	Description
First / Last / Step	Explicit channel range. Last = 'Auto' (−1) lets the application auto-discover channels (Filter mode only).
Custom list	Comma-separated list, e.g. 0,1,5,10. Overrides First/Last/Step when non-empty.
Max entries per channel	Limit events read per channel. 0 = All entries. Useful for fast preview of large files.
Histogram bins	Number of bins for the resulting histogram. Default 1024.

4. Peak Detection & Assignment

The Peak Detection group in the left panel controls automatic peak finding. After loading a file, click ■ Detect Peaks (Current channel) to run detection on the spectrum currently displayed. Version 2.0 offers three detection algorithms selectable from the Method dropdown.

4.1 Detection Method Selector

Use the **Method** dropdown at the top of the Peak Detection panel to choose between TSpectrum and Sliding Window. The parameter panel below updates to show the relevant controls for the selected method.

4.2 TSpectrum (ROOT / SciPy)

The default algorithm. Uses ROOT TSpectrum::Search() or SearchHighRes() when PyROOT is available; falls back to scipy.signal.find_peaks automatically when uproot is active.

Control	Description
Threshold (slider)	Minimum peak height as a fraction of the spectrum maximum (1%–50%). Lower values find smaller peaks. A red dashed threshold line is drawn on the spectrum and updates live as you drag the slider.
Search mode	Standard uses TSpectrum::Search(). High Resolution uses SearchHighRes() with deconvolution — resolves closely-spaced peaks. High Resolution is the default.
Iterations (HighRes)	Deconvolution iterations for SearchHighRes. More = sharper, slower. Typical: 5–20.
Sigma (bins)	Gaussian smoothing width in bins before peak search. Increase for noisy spectra.
Subtract background	Run TSpectrum::Background() (SNIP) before peak search. The estimated background is overlaid on the spectrum in orange. Recommended for LYSO.
BG iterations	SNIP smoothing width. More = broader background estimate. Try 20–40 for LYSO.

4.3 Sliding Window Detection ★ New

A new peak finder that scans the spectrum with a sliding window of configurable ADC width. A bin is reported as a peak candidate when three conditions are all met:

- The bin's smoothed count is the maximum within the window (local maximum condition).
- The bin's count is at least *Height threshold* × global maximum.
- The local prominence (peak height minus window minimum) exceeds *Min prominence* × global maximum.

Candidates within half a window width of each other are merged, keeping the most prominent. Up to Max peaks candidates are returned, sorted by ADC position.

Control	Description
Window width (ADC)	Full width of the sliding window in ADC units. A bin is a peak only if it is the maximum within this window. Set to roughly the expected peak separation.
Height threshold %	Minimum peak height as a percentage of the global spectrum maximum.

Control	Description
Min prominence %	Minimum local prominence — how much the peak must stand above the local baseline — as a percentage of the global maximum.
Pre-smooth σ (bins)	Gaussian pre-smoothing before window scan (bins). Set to 0 to disable. Helps reject single-bin noise spikes.

■ ***TIP:*** *Use Sliding Window when TSpectrum over-resolves broad peaks, or when you need fine control over what counts as a 'local' maximum. TSpectrum is generally more robust for gamma-ray spectra.*

4.4 Shared Controls (both methods)

Control	Description
Max peaks	Maximum number of peaks returned. Peaks are ranked by prominence and the top N are kept.
Pedestal cut (ADC)	Ignore all peaks below this ADC value. Set above the pedestal to avoid false detections at ADC = 0.

4.5 Gaussian Peak Confirmation ★ New

Enable the **Gaussian confirm peaks** checkbox to add a confirmation step after any detection method. For each candidate peak, a Gaussian + linear background is fitted in a window of ± *win* ADC units around the candidate:

- **Accepted peaks:** Successfully fitted candidates — shown as orange | lines. The centroid is refined to the Gaussian centre.
- **Rejected peaks:** Candidates whose fit fails (bad shape, drifted centroid, NaN covariance) — shown as red × markers on the spectrum.
- **Gaussian overlays:** Green fit curves are drawn over each accepted peak, annotated with σ and χ^2/NDF.

Control	Description
Gaussian confirm peaks	Enable/disable Gaussian confirmation after detection.
win ± (ADC)	Half-width of the fit window around each candidate (ADC units). Should be roughly 2–3× the expected peak FWHM.

■ ***NOTE:*** *Gaussian confirmation can reject genuine broad peaks if the window is too narrow, or accept noise if the window is too wide. Tune win ± to roughly match your peak FWHM in ADC units. The red × rejected markers remain visible on the spectrum for review.*

5. Peak Assignment

5.1 Assigning Energies to Detected Peaks

After clicking ■ Detect Peaks, the Assign Energies table is populated with the ADC positions of all found peaks (orange | lines on the spectrum). To assign a physical energy to a peak:

- Double-click any row in the detected peaks table, or select a row and click Assign.
- The Assign Energy dialog opens with the ADC position pre-filled.
- Type an energy in keV, or click a row in the quick-select table for common gamma sources.
- Add an optional label (e.g. Cs-137 661.7 keV) and click OK.

The assigned peak appears in the Calibration Points table and as a green dashed line on the spectrum.

5.2 Manual Peak Entry

- ■ **Click mode:** Toggle click mode. The cursor changes to a crosshair. Click anywhere on the spectrum to open the Assign Energy dialog at that ADC position.
- **+ Manual (all ch.):** Opens the Assign Energy dialog with ADC set to zero. The point is added to every loaded channel simultaneously. If ADC = 0 and energy = 0, it adds a zero-point (origin) to all channels.

5.3 Quick-Select Energy Table

The Assign Energy dialog includes a scrollable table of common gamma-ray sources:

Energy (keV)	Source	Common use
511.0	Na-22 / annihilation	PET, pair production
1274.5	Na-22	PET calibration
661.7	Cs-137	General-purpose standard
1173.2	Co-60	Gamma calibration
88.0	Lu-176	LYSO internal line
202.0	Lu-176	LYSO internal line
307.0	Lu-176	LYSO internal line
122.1	Co-57	Low-energy calibration
344.3	Eu-152	Multi-energy standard
1460.8	K-40	Natural background
59.5	Am-241	Low-energy / X-ray range
88.0	Cd-109	X-ray / low energy

6. Peak Propagation

The Detect Peaks in All Channels panel lets you define a peak on one representative channel and automatically find the same peak in all other channels within a configurable ADC search window. This is the most efficient workflow for multi-channel systems.

6.1 Workflow

1. Select a representative channel using the channel selector.
2. Detect peaks (click ■ Detect Peaks) or manually place them with Click mode.
3. Assign a known energy to the peak via the Assign Energy dialog.
4. Set the **Search window (± ADC)** spinbox. A value of 50 means the application looks in [ref_ADC −50, ref_ADC + 50].
5. Click ■ **Detect Last Peak → All Channels**. A progress bar appears while all channels are processed.
6. Repeat steps 2–5 for each additional energy peak.

6.2 Search Algorithm

For each target channel, the algorithm:

- Slices the spectrum to [ref_ADC −window, ref_ADC + window].
- Runs the currently selected detection method (TSpectrum or Sliding Window) inside that window.
- Selects the candidate closest to the reference ADC position.
- Falls back to the smoothed centroid of the local maximum if no peak is detected.

The resulting peaks carry the same known_energy as the reference, with ADC positions refined per channel to correctly account for gain variation.

> ■ ***TIP:*** *If a channel's gain differs greatly from the reference, widen the search window (e.g. 100–200 ADC). If false peaks are being matched, reduce the window.*

7. Calibration Fitting

7.1 Calibration Models

Linear (default)

$E = P0 + P1 \cdot Q$

Simple linear model. P0 is the offset (keV), P1 is the gain (keV/ADC). Requires ≥ 2 calibration points. With exactly 2 points the fit is an exact interpolation (NDF = 0), which is always valid.

Nonlinear Scintillator Model

$E = P0 \cdot (P1^{Q})^{P2} + P3 \cdot Q - P0$

Physically motivated for scintillator detectors. Requires ≥ 4 points.

Param	Typical value	Meaning
P0	~10 keV	Nonlinearity amplitude. Set to 0 for purely linear response.
P1	~0.999	Exponential base. Must be in (0, 2).
P2	~1.0	Exponent shaping factor.
P3	keV/ADC	Linear gain (dominant term).

Nonlinear 3-point Variant

$E = P0 \cdot P1^{Q} + P3 \cdot Q - P0$ (P2 = 1)

Automatically selected when exactly 3 calibration points are available and the Nonlinear model is chosen. A note is appended to the fit result.

Custom Expression

Select Custom in the model dropdown. A text field appears in the toolbar. Type any expression using `x` as the ADC variable and single letters (not x) as parameters:

```
a*x**2 + b*x + c
a*np.exp(-b*x) + c*x
```

Available functions: np.exp, np.log, np.sqrt, np.abs, np.sin, np.cos.

7.2 Fitting a Single Channel

After assigning calibration peaks, click ■ Fit This Channel. The calibration curve is drawn in the Single Channel tab, annotated with each peak's energy and χ^2/NDF.

7.3 Fit All Channels

Click ■ Fit All Channels to fit every loaded channel using its currently assigned calibration points. Fitting runs in a background thread; a progress bar updates in real time. On completion, the Overview Grid is redrawn with green/red borders, the Coefficient Trends tab is updated, and the status bar reports the number of bad channels.

7.4 Detect + Fit All Channels ★ New

The ■ **Detect + Fit All** button combines peak detection, energy propagation, and calibration fitting into a single operation:

1. Assign at least one known energy to a peak on the current channel.
2. Click ■ Detect + Fit All. A background thread processes every loaded channel.
3. For each channel, peaks are detected using the current detection settings.
4. Each detected peak is matched to the reference energy assignments from the current channel (closest peak within the search window).
5. The calibration model is fitted using the matched peaks.
6. The Overview Grid, Coefficient Trends, and status bar are updated on completion.

■ ***NOTE:*** *Detect + Fit All requires at least one energy assignment on the current channel to use as a reference. Channels with no matched peaks after detection will not be fitted (they appear as bad channels).*

7.5 Bad Channel Flagging

A channel is flagged as bad only if the fit genuinely fails — specifically when the covariance matrix contains NaN (numerically unstable fit). A channel is never flagged bad merely because χ^2/NDF is large or uncertainties are wide.

8. GUI Reference

8.1 Toolbar

Control	Description
■ Open ROOT File	Opens a ROOT file and shows the Configure Data Source dialog.
Model dropdown	Choose: Linear, Nonlinear, Nonlinear 3-pt, or Custom.
Custom expression field	Visible when Custom is selected. Type the expression here.
Export	Saves calibration results (log + coefficients) to a chosen directory.

8.2 Left Panel

Group	Contents
Channel Navigator	Previous/Next arrows, channel combo box, per-channel info label (entries, source, fit status).
Peak Detection	Method dropdown (TSpectrum / Sliding Window), method-specific controls, shared controls (Max peaks, Pedestal cut), Gaussian confirmation, Detect Peaks button.
Assign Energies	Detected peaks table (double-click to assign), Assign, + Manual (all ch.), ■ Click mode toggle.
Detect Peaks in All Ch.	Last assigned peak display, Search window ± ADC spinbox, ■ Detect Last Peak → All Channels button, status label.
Calibration Points	Assigned peaks table, − Remove, ↑ Use Global, Exclude checkbox.
Fit buttons	■ Fit This Channel, ■ Fit All Channels, ■ Detect + Fit All.

8.3 Single Channel Tab

The top canvas shows the raw ADC spectrum in log-y scale. Orange | lines mark accepted detected peaks; red × markers show rejected peaks (Gaussian confirmation failed); green dashed lines mark assigned peaks with energy labels; a red dashed horizontal line shows the TSpectrum threshold; an orange fill shows the SNIP background estimate; green curves show Gaussian fits on confirmed peaks. The bottom canvas shows the calibration curve with χ^2/NDF.

8.4 Overview Grid Tab

Displays thumbnails of all loaded channel spectra in an 8-column scrollable grid. Green border = successful fit, red border = bad channel, grey = not yet fitted. Click any thumbnail to navigate to that channel.

8.5 Coefficient Trends Tab

Plots the value of a chosen calibration parameter (P0, P1, P2, P3) against channel ID with error bars. Bad channels are marked with a red vertical line. Use the parameter dropdown to switch. The parameter list updates automatically when the calibration model is changed.

9. Calibrated Spectrum Tab

The ■ Calibrated Spectrum tab converts raw ADC spectra to energy spectra (keV x-axis) using fitted or file-loaded calibration coefficients. In version 2.0, the calibrated spectrum now shows the **full ADC range** rather than clipping at the last calibration peak. For nonlinear models, extrapolation beyond the last peak is capped at 3× the energy at the last calibration point to prevent display distortion.

9.1 Loading Calibration

- ■ **From Session:** Uses results from the current fitting session. Click after Fit All Channels.
- ■ **Load coeffs.txt:** Loads a previously exported calibration_coeffs.txt. Allows applying calibration from a prior session to any ROOT file.

9.2 Energy Range & Binning

Control	Description
Min (keV)	Low edge of the output energy axis. 0 = auto (minimum calibrated ADC energy).
Max (keV)	High edge. 0 = auto.
Bins	Number of bins in the output histogram. Default 1024.

9.3 Applying Calibration

- Select a channel in the Channel combo box for a single-channel view.
- ■ **Apply to All:** Computes calibrated spectra for all calibrated channels. Required before batch resolution analysis.
- ■ **Analyse Resolution → :** Sends the current channel's calibrated spectrum to the Resolution tab.

10. Resolution Analysis Tab

10.1 Interactive Fit Workflow

1. Send a calibrated spectrum using ■ Analyse Resolution → in the Calibrated Spectrum tab.
2. The spectrum is displayed with a draggable SpanSelector (yellow band).
3. Drag horizontally over a peak to select the energy fit range.
4. Set the Peak label (keV) spinbox to the nominal energy.
5. Choose a background model (Compton step recommended for gamma peaks).
6. Click ■ Fit Peak. The Gaussian is fitted and overlaid in red. FWHM and R% are annotated.
7. To fit the same peak across all channels, click ■ Fit All Channels (same range).

10.2 Background Models

Model	Function	When to use
Compton step (default)	`Gauss + B·erfc + C + D·E`	Most physically motivated. Models multiply-scattered photons.
Exponential	`Gauss + B·exp(C·E)`	Good for Compton tail spectra.
Quadratic	`Gauss + B + C·E + D·E²`	Flexible polynomial background.
Linear	`Gauss + B + C·E`	Simple linear background.
Gaussian only	`Gauss only`	Use when background is negligible.
Auto	`Tries all, picks lowest χ²/NDF`	Let the algorithm choose the best model.

10.3 Derived Quantities

Quantity	Formula	Description
FWHM	`2.3548 × σ`	Full width at half maximum (keV)
R%	`FWHM / ∝ × 100`	Relative energy resolution (%)

11. Exporting Results

11.1 calibration_log.txt

Full human-readable report with one section per channel: model label, χ^2/NDF, all parameters with uncertainties, and a table of ADC positions, assigned energies, and residuals.

11.2 calibration_coeffs.txt

Compact machine-readable table, one row per channel:

```
Channel    P0                P1            Chi2/NDF  Status
      0  -1.23456e+00  +5.00000e-01    0.0000  OK
      1  -1.18900e+00  +4.99800e-01    0.0000  OK
     42           nan           nan       nan  BAD: ...
```

This file can be loaded directly by the Calibrated Spectrum tab to apply calibration in a future session without re-fitting.

11.3 resolution_results.txt

Full table with FWHM, uncertainty, R%, χ^2/NDF, and status for every (channel, peak) combination that was fitted in the Resolution tab.

12. Typical Workflows

12.1 Standard Multi-Channel Calibration

1. Open ROOT file → configure TTree / branch names → OK.
2. Navigate to a representative channel.
3. Click ■ Detect Peaks. Choose detection method (TSpectrum or Sliding Window). Adjust controls until the correct peaks are highlighted.
4. Optionally enable Gaussian confirm to refine centroids and reject spurious peaks.
5. Double-click each detected peak row → assign known energy → OK.
6. Verify green markers align with the correct spectral features.
7. Set Search window (± ADC). Click ■ Detect Last Peak → All Channels.
8. Repeat steps 3–7 for each additional calibration peak.
9. Select Linear model (or Nonlinear for scintillators). Click ■ Fit All Channels.
10. Review Overview Grid. Investigate any red-bordered channels individually.
11. Click Export to save calibration_log.txt and calibration_coeffs.txt.

12.2 Fast Automated Workflow ★ New

1. Open ROOT file and configure data source.
2. Navigate to a representative channel. Detect peaks and assign energies.
3. Click ■ Detect + Fit All. The application detects peaks on every channel, propagates your energy assignments, and fits — all automatically.
4. Review the Overview Grid and Coefficient Trends.
5. Export results.

■ ***TIP:*** *The Detect + Fit All workflow is ideal when spectra across channels are similar. For detectors with large gain variation, use the manual propagation workflow instead.*

12.3 Applying Saved Calibration to a New File

1. Open the new ROOT file and configure data source.
2. Switch to the ■ Calibrated Spectrum tab.
3. Click ■ Load coeffs.txt and select the previously exported calibration_coeffs.txt.
4. Click ■ Apply to All.
5. Inspect calibrated spectra for each channel.

12.4 Resolution Analysis

1. Complete calibration fitting or load calibration from file.
2. In the ■ Calibrated Spectrum tab, click ■ Apply to All.
3. Select any channel and click ■ Analyse Resolution → .
4. In the Resolution tab, drag a span over the desired peak.
5. Set Peak label, choose background model, click ■ Fit Peak.
6. Click ■ Fit All Channels (same range) to propagate to all channels.
7. Check the Resolution Trend tab for FWHM vs channel and R% vs channel.
8. Click ■ Export to save resolution_results.txt.

13. Troubleshooting

Problem	Solution
No ROOT backend found	Source thisroot.sh (PyROOT) or install uproot: pip install uproot awkward
No channels found after loading	For Array/Custom mode, set channel range (First/Last) manually. For Filter mode, verify the channel branch name is correct.
Detect Peaks button does nothing	Ensure a file is loaded. Check that a detection method is selected. Try both TSpectrum and Sliding Window.
TSpectrum finds wrong peaks	Lower threshold or increase sigma. Try SearchHighRes instead of Standard. Alternatively, use Click mode to place peaks manually.
Sliding Window misses peaks	Reduce the window width so it is closer to the expected peak spacing. Lower the Height threshold % and Min prominence %.
Gaussian confirm rejects real peaks	Widen the 'win ±' parameter — it should be ≥2–3× the peak FWHM in ADC. Check that the peak has sufficient statistics for a reliable Gaussian fit.
Detect + Fit All fits nothing	Ensure at least one energy is assigned on the current channel before clicking Detect + Fit All. Widen the search window if channel gains vary significantly.
All channels flagged bad	Check that calibration points are assigned correctly. Verify the model has enough points (linear ≥2, nonlinear ≥4).
Calibration curve looks wrong	Verify peak assignments — green markers should align with the correct peaks. Remove and reassign if needed.
Calibrated spectrum cuts off early	Update to v2.0 — this is fixed. The spectrum now shows the full ADC range.
Gaussian fit fails in resolution tab	Widen the span selection. Switch to 'Gaussian only' if the background model causes instability.
Export produces no files	Run Fit All Channels or Fit This Channel before exporting. At least one successful fit is required.

14. Module Reference

14.1 File Structure

File	Responsibility
`main.py`	Entry point. Initialises Qt, runs backend check, launches MainWindow.
`src/root_loader.py`	ROOTFileLoader. Handles all ROOT I/O for both backends and all draw modes.
`src/peak_manager.py`	PeakManager. Stores global and per-channel peaks. TSpectrum / SciPy / Sliding Window detection. Gaussian confirmation. detect_peaks() dispatcher.
`src/calib_fitter.py`	CalibrationFitter. scipy.optimize.curve_fit for linear, nonlinear, and custom models. FitResult dataclass.
`src/output_writer.py`	OutputWriter. Writes calibration_log.txt and calibration_coeffs.txt.
`src/calib_spectrum.py`	CalibSpectrumEngine. Converts ADC spectra to full-range energy spectra. Nonlinear extrapolation capping.
`src/resolution.py`	ResolutionCalculator. Gaussian + multiple background model fitting. FWHM, R% extraction and export.
`src/main_window.py`	MainWindow and all PyQt5 widgets, canvases, dialogs, and tab builders. DetectAndFitWorker background thread.
`src/calib_spectrum_tab.py`	CalibratedSpectrumTab widget (Tab 4).
`src/resolution_tab.py`	ResolutionTab with SpanSelector and trend plots (Tab 5).

Appendix A — Calibration Formula Reference

Model	Formula	Notes
Linear	$E = P0 + P1 \cdot Q$	Default. 2 params. Works with 2+ peaks.
Nonlinear	$E = P0 \cdot (P1^{Q})^{P2} + P3 \cdot Q - P0$	4 params. Requires 4+ peaks. P1 ∈ (0,2).
Nonlinear 3-pt	$E = P0 \cdot P1^{Q} + P3 \cdot Q - P0$ (P2=1)	3 params. Auto-selected with 3 points.
Custom	User-defined in toolbar	Parameters: any single letter except x.

Appendix B — Keyboard & Mouse Quick Reference

Action	How
Next / previous channel	■ arrow buttons in left panel, or change combo box
Place a peak manually	Toggle ■ Click button, then left-click on spectrum
Assign energy to peak	Double-click any row in the Assign Energies table
Gaussian fit range (res.)	Drag horizontally on the calibrated spectrum in Resolution tab
Zoom spectrum	Matplotlib toolbar magnifier, or scroll wheel
Pan spectrum	Matplotlib toolbar arrow icon, then drag
Reset view	Matplotlib toolbar home icon
Navigate to channel from grid	Click any thumbnail in the Overview Grid tab

End of User Manual — DetectorCalibGUI v2.0 — Siddharth Parashari