TCM GUI → Functions Relationship

# StartDefinitiveTCM (StartDefinitiveTCM.m)

1. Sets the working directory
2. [Utilities, class] ConfigurationFileManager.GetInstance(): Opens the Settings.ini and Preferences.ini files
3. [GUI, function] LookAndFeel(): Makes everything pretty
4. [GUI, function] Main(): Opens the main window

# Main Window (GUI/Main.m)

## Hardware Interface

1. SystemPower\_Callback()
   1. CascadeActionPower(): Enables/disables buttons as appropriate
      1. ConnectHardware(): Connects to each device
         1. Sets up cameras
         2. [GPIB, class] ESP300\_Control(): Laser controller
         3. [GPIB, class] SR830\_Control(): Lock-in amplifier
         4. [GPIB, class] ProbeLaser\_Control(): Probe laser control on lock-in amp, requires SR830\_Control()
         5. [GPIB, class] DS345\_Control(): Function generator
         6. [GPIB, class] ESP300\_Control(): Stage controller
         7. HomeSampleStages(): Asks user what to do to home the stages, then does it
         8. Sets operational parameters of the stage controller
         9. Sets the power of the
      2. or DisconnectHardware(): Shuts down and disconnects each device
         1. Deletes each of the classes created by ConnectHardware()
         2. Each class automatically shuts down and disconnects each device when it is deleted (see the delete() function in each class for details)

## Position Sample…

1. PositionSample\_Callback()
   1. [GUI, function] ControlGUI() *with PositionSampleGUIAddOn*: Opens the control window, configured for positioning the sample

## Collect Data…

1. CollectData\_Callback()
   1. [GUI, function] ControlGUI() *with CollectDataGUIAddOn*: Opens the control window, configured for collecting data

## Run Analysis…

1. RunAnalysis\_Callback()
   1. [GUI, function] StartAnalysis(): Opens the analysis window

## Close button

1. MainWindow\_CloseRequestFcn(): deletes stuff, closes the window  
   Note: this causes MATLAB to delete any classes in memory. This automatically closes the hardware connections since the hardware interface classes are designed to safely disconnect from the hardware when they are deleted.

# Control GUI (GUI/Controls.m)

Contains the video feed, step size selectors, and stage position controls. The startup is very involved, sorry about that. I can provide more documentation if needed.

## Step Size

1. StepSizeGroup\_SelectionChanged()
   1. UpdateStepSizeGroup(): Changes the step sizes used by the stage move buttons

## Stage Position

Each stage has its own set of functions (with ‘*D*’ representing ‘X’, ‘Y’, or ‘Z’):

* *D*Edit\_Callback(): A value is typed into the edit box
* *D*LeftFast\_Callback(): Click on the left fast button
* *D*LeftModerate\_Callback(): Click on the left medium button
* *D*LeftSlow\_Callback(): Click on the left slow button
* *D*RightFast\_Callback(): Click on the right fast button
* *D*RightModerate\_Callback(): Click on the right medium button
* *D*RightSlow\_Callback(): Click on the right slow button
* *D*Slider\_Callback(): Slider is moved

These use the functions:

* UpdateEdit2Slider(*D*): Changes the value of the edit box for *D* to show the current position
* MoveStageToSliderPosition(*D*): Sends a command to the stage controller to move the *D* stage
* StepLeftLarge(*D*): Sends a command to the stage controller to move the D stage left a large step
* StepLeftMedium(*D*): ditto, medium step stage *D*
* StepLeftSmall(*D*): ditto, small step stage *D*
* StepRightLarge(*D*): Sends a command to the stage controller to move the D stage right a large step
* StepRightMedium(*D*): ditto, medium step stage *D*
* StepRightSmall(*D*): ditto, small step stage *D*

# Position Sample (GUI/Controls\_SamplePosition.m)

Contains the camera selectors

## Camera Selection

1. CameraSelectionGroup\_SelectionChangedFcn()
   1. UpdateCameraSelectionGroup()

## Link Stage to Camera

1. UpdateLinkCheckbox(): Puts a check in the checkbox if the stage and camera are synced to the same position

## Move Stage to Camera

1. [GUI/Controls.m] MoveStageToCamera(): Detects the current stage location and moves it to the camera
2. UpdateLinkCheckbox(): Puts a check in the checkbox if the stage and camera are synced to the same position

# Collect Data (GUI/Controls\_CollectData.m)

Contains the file save boxes, advanced settings button, laser on/off switches, and the run scan button

## *Save Folder Edit Box*

1. SaveFolderEdit\_Callback(): Verifies that the folder exists
   1. CheckPath(): Checks to see if the file already exists

## Browse

1. SaveFolderBrowseButton\_Callback(): Opens a dialog to select a folder to save in
   1. CheckPath(): Checks to see if the file already exists

## *Save Name Edit Box*

1. SampleNameEdit\_Callback()
   1. CheckPath(): Checks to see if the file already exists

## Sample Info

1. inputldg(): Opens a window that allows the user to enter more information about the sample

## Pump Laser

1. PumpLaserButton\_Callback()
   1. TogglePumpLaser(): Swaps the state of the pump laser

## Probe Laser

1. ProbeLaserButton\_Callback()
   1. ToggleProbeLaser(): Swaps the state of the probe laser

## Run Scan

1. RunScanButton\_Callback()  
   *you may recognize the actions of the functons below from EasyScan/EasyScanB, but is it heavily modified with handles to the hardware control classes, etc…*
   1. [GUI, function] Run(): Creates the window that displays the scan data
   2. [GUI/Run.m] Center(): Performs the center scan
   3. [GUI/Run.m] Focus(): Performs the focus scan
   4. [GUI/Run.m] Data(): Collects the data

## Advanced Settings

1. [GUI, function] Controls\_CollectData\_Settings(): Opens the windows with advanced settings for data collection
2. Sets the pump laser power in case it was changed in the advanced settings by the user

# Data Analysis (GUI/StartAnalysis.m)

## Analysis options

The following functions are called when the corresponding settings are changed:

* FitAmplitudeCheckbox\_Callback()
* FilmMaterialPopup\_Callback()
* FilmThicknessEdit\_Callback()
* KapitzaResistanceEdit\_Callback()
* MagnificationPopup\_Callback()
* MaximumFrequencyEdit\_Callback()
* ModelPopup\_Callback()
* SubstratePopup\_Callback()

## Advanced…

1. AdvancedButton\_Callback()
   1. [GUI, function] StartAnalysis\_Settings(): Opens dialog for advanced analysis settings

## Browse…

1. BrowseButton\_Callbak()
   1. uigetfile(): Opens a system dialog to select a file
   2. fullfile(): Combines the file name and directory to create a valid file path
   3. set(): Sets the contents of the file edit control to the file path
   4. FileEdit\_Callback(): Validate the new file path

## Edit Database…

1. EditDatabaseButton\_Callback()
   1. [Utilities/Database.m] Edit()
      1. winopen(): Opens Resources/Database.xlsx for editing in Microsoft Excel

## *File Edit Box*

1. FileEdit\_Callback()
   1. exists(): Checks if the file exists
   2. Sets the colors and enables/disables the “Analysis” button if the file path is valid

## Refresh Database

1. RefreshDatabaseButton\_Callback()
   1. LoadDatabase()
      1. [Utilities, function] Database(): Get the function handles to the contents of Database.m
      2. [Utilities/Database.m] ReloadMagnifications(): Reload the magnifications from the database file
      3. [Utilities/Database.m] ReloadMaterials(): Reload the materials from the database file
   2. RefreshPopups(): Refreshes the contents of the controls dependent on the magnification and material options

## Start Analysis…

1. findall(): Get all controls in the dialog box that can be disabled
2. set(): Disable all controls
3. [Analysis, function] FitTCMData()
   1. load(): Load the data from a previous TCM run
   2. [Analysis, function] ThermalWaveNumbers(): Create the analysis object
   3. [Analysis/ThermalWaveNumbers.m] Run(): Perform the analysis
      1. GoodnessOfFit(): Evaluates the chi-squared goodness of fit
      2. MinimizationPlot(): Plots the current solution and goodness-of-fit value
         1. [GUI, function] AnalysisProgress(): Open up the window for displaying the analysis progress
         2. FinalizeAnalysisPlot()
            1. [GUI/AnalysisProgress.m] Finalize()

Update(): Update the plots and progress bar

IsCancelling(): check for the press of the “Cancel” button

1. [Analysis/ThermalWaveNumbers.m] GetStandardError(): Gets the standard error of the analysis, if requsted
2. questdlg(): Display the analysis results and ask the user if he/she wants to save the data
3. uiputfile(): Opens a system dialog that allows the user to select where to save an Excel spreadsheet containing the data
4. xlswrite(): Write the data to the user-specified file, if requested