

Driver Board Dropwatcher

Generated by Doxygen 1.9.6

1 README	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 DriverBoardDropwatcher Namespace Reference	11
7 Class Documentation	13
7.1 DriverBoardDropwatcher.Form1 Class Reference	13
7.1.1 Member Function Documentation	18
7.1.1.1 btnCancel_Click()	18
7.1.1.2 btnClearHead_Click()	19
7.1.1.3 btnConnectDisconnect_Click()	19
7.1.1.4 btnPrintImage_Click()	19
7.1.1.5 cbDropWatchHeadSelection_SelectedIndexChanged()	20
7.1.1.6 cbDropWatchMode_SelectedIndexChanged()	20
7.1.1.7 cbSerialPort_DropDown()	21
7.1.1.8 cbSerialPort_SelectedIndexChanged()	21
7.1.1.9 connect_board()	21
7.1.1.10 convertImageToData()	21
7.1.1.11 DataRecievedHandler()	22
7.1.1.12 DeleteAllFilesInFolder()	22
7.1.1.13 determineStatus()	22
7.1.1.14 disconnect_board()	23
7.1.1.15 Dispose()	23
7.1.1.16 DropWatchingTab_Click()	23
7.1.1.17 DropWatchingTab_PreviewKeyDown()	23
7.1.1.18 EncoderTrackedPositionSelection_SelectedIndexChanged()	25
7.1.1.19 FillCycle()	25
7.1.1.20 FillCycleA_Click()	26
7.1.1.21 FillCycleB_Click()	26
7.1.1.22 FillCycleC_Click()	26
7.1.1.23 fillHead_Click()	27
7.1.1.24 FillSingleNozzleButton_Click()	27

7.1.1.25 FillSpanNozzleButton_Click()	27
7.1.1.26 Form1_FormClosing()	28
7.1.1.27 Form1_Load()	28
7.1.1.28 frequencyValue_ValueChanged()	29
7.1.1.29 GapValue_ValueChanged()	29
7.1.1.30 ImageBoxClicked()	29
7.1.1.31 ImageModeSelection_SelectedIndexChanged()	30
7.1.1.32 MakeGrayscale3()	30
7.1.1.33 NozzleValue_ValueChanged()	31
7.1.1.34 parseJsonData()	31
7.1.1.35 PD_Polarity_SelectedIndexChanged()	31
7.1.1.36 pdDirection_SelectedIndexChanged()	32
7.1.1.37 pictureBox1_Click()	32
7.1.1.38 pictureBox2_Click()	32
7.1.1.39 pictureBox3_Click()	34
7.1.1.40 pictureBox4_Click()	34
7.1.1.41 powerOff()	34
7.1.1.42 powerOn()	35
7.1.1.43 powerOnOff_Click()	35
7.1.1.44 PrintingImage()	35
7.1.1.45 reset_Click()	36
7.1.1.46 SpanValue_ValueChanged()	36
7.1.1.47 tcDropWatchingAndImageModes_SelectedIndexChanged()	36
7.1.1.48 temperature_ValueChanged()	37
7.1.1.49 ThreadTask()	37
7.1.1.50 VerifyImageData()	37
7.1.1.51 voltage_ValueChanged()	38
7.1.2 Member Data Documentation	38
7.1.2.1 ofd	38
8 File Documentation	39
8.1 Form1.cs File Reference	39
8.1.1 Detailed Description	39

Chapter 1

README

This project folder contains source code for the Driver Board Dropwatcher Software.

Item Naming in Source Code:

- txtb = Text Box
- cb = Combo Box,
- chbx = Check Box,
- lb = Label,
- btn = Button
- tc = Tab Control.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

DriverBoardDropwatcher	11
--	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Form	
DriverBoardDropwatcher.Form1	13

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

DriverBoardDropwatcher.Form1	13
--	----

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

[Form1.cs](#)

This C# Code contains the Source Code for the Driver Board Dropwatcher Application [39](#)

Chapter 6

Namespace Documentation

6.1 DriverBoardDropwatcher Namespace Reference

Classes

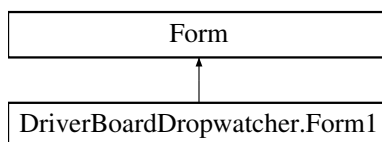
- class [Form1](#)
- class **Program**

Chapter 7

Class Documentation

7.1 DriverBoardDropwatcher.Form1 Class Reference

Inheritance diagram for DriverBoardDropwatcher.Form1:



Static Public Member Functions

- static Bitmap [MakeGrayscale3](#) (Bitmap original)
Grey Scale Image Function.

Protected Member Functions

- override void [Dispose](#) (bool disposing)
Clean up any resources being used.

Private Member Functions

- void [ThreadTask](#) ()
A Thread Task Function.
- void [cbSerialPort_DropDown](#) (object sender, EventArgs e)
Detects Active Serial COM Ports from device.
- void [cbSerialPort_SelectedIndexChanged](#) (object sender, EventArgs e)
Detects Changes in Serial Ports.
- void [disconnect_board](#) ()
Disconnect Driver Board.
- void [connect_board](#) ()
Connect Driver Board.

- void [DataRecievedHandler](#) (object sender, SerialDataReceivedEventArgs e)
Processes data received from driver board.
- void [determineStatus](#) ()
Determine the Status of Voltages and Temperatures.
- void [parseJsonData](#) (string input_string)
Parse the JSON Data.
- void [btnConnectDisconnect_Click](#) (object sender, EventArgs e)
Controls the Connect/Disconnect Toggle.
- void [powerOnOff_Click](#) (object sender, EventArgs e)
Controls the Power On/Off Toggle.
- void [powerOn](#) ()
Power On Function.
- void [powerOff](#) ()
Power Off Function.
- void [voltage_ValueChanged](#) (object sender, EventArgs e)
Fetches Voltage Data from GUI.
- void [temperature_ValueChanged](#) (object sender, EventArgs e)
Fetches Temperature Data from GUI.
- void [frequencyValue_ValueChanged](#) (object sender, EventArgs e)
Fetches Frequency Data from GUI.
- void [cbDropWatchMode_SelectedIndexChanged](#) (object sender, EventArgs e)
Fetches User Selected Drop Watch Mode Selection from GUI.
- void [reset_Click](#) (object sender, EventArgs e)
Reset Board Button.
- void [cbDropWatchHeadSelection_SelectedIndexChanged](#) (object sender, EventArgs e)
Fetches User Selected Head from GUI.
- void [ImageBoxClicked](#) (object sender, EventArgs e)
Image Box Selection in Image Mode.
- void [pictureBox1_Click](#) (object sender, EventArgs e)
Picture Box 1 is pressed.
- void [pictureBox2_Click](#) (object sender, EventArgs e)
Picture Box 2 is pressed.
- void [pictureBox3_Click](#) (object sender, EventArgs e)
Picture Box 3 is pressed.
- void [pictureBox4_Click](#) (object sender, EventArgs e)
Picture Box 4 is pressed.
- void [convertImageToData](#) (string file_name)
Convert Image to Data.
- void [btnPrintImage_Click](#) (object sender, EventArgs e)
Print Image.
- void [PrintingImage](#) (int head)
Printing Image Function.
- void [VerifyImageData](#) (int head, string existing_lines)
Picture Box 2.
- void [NozzleValue_ValueChanged](#) (object sender, EventArgs e)
Nozzle Value Modified in GUI.
- void [SpanValue_ValueChanged](#) (object sender, EventArgs e)
Span Value Modified in GUI.
- void [FillSingleNozzleButton_Click](#) (object sender, EventArgs e)
Fill Nozzle Push Button.
- void [FillSpanNozzleButton_Click](#) (object sender, EventArgs e)

- *Fill Span Push Button.*
- void [ImageModeSelection_SelectedIndexChanged](#) (object sender, EventArgs e)
 - *Runs if Image Mode Selection is Modified.*
- void [btnClearHead_Click](#) (object sender, EventArgs e)
 - *Clear Heads Push Button.*
- void [FillCycleA_Click](#) (object sender, EventArgs e)
 - *Fill Cycle A Button Press.*
- void [FillCycleB_Click](#) (object sender, EventArgs e)
 - *Fill Cycle B Button Press.*
- void [FillCycleC_Click](#) (object sender, EventArgs e)
 - *Fill Cycle C Button Press.*
- void [FillCycle](#) (object sender, EventArgs e)
 - *Fill Cycle Function.*
- void [GapValue_ValueChanged](#) (object sender, EventArgs e)
 - *Gap Value Modified.*
- void [FillGapButton_Click](#) (object sender, EventArgs e)
- void [fillHead_Click](#) (object sender, EventArgs e)
 - *Fill Head Button Press.*
- void [PD_Polarity_SelectedIndexChanged](#) (object sender, EventArgs e)
 - *PD Polarity Mode Selection.*
- void [EncoderTrackedPositionSelection_SelectedIndexChanged](#) (object sender, EventArgs e)
 - *Encoder Position Selection Change.*
- void [pdDirection_SelectedIndexChanged](#) (object sender, EventArgs e)
 - *PD Direction Selection Change.*
- void [tcDropWatchingAndImageModes_SelectedIndexChanged](#) (object sender, EventArgs e)
 - *Mode Selection.*
- void [DropWatchingTab_PreviewKeyDown](#) (object sender, PreviewKeyDownEventArgs e)
 - *Arrow Keys Pushed.*
- void [DropWatchingTab_Click](#) (object sender, EventArgs e)
 - *Drop Watching Tab pressed.*
- void [btnCancel_Click](#) (object sender, EventArgs e)
 - *Cancel Button Pushed.*
- void [DeleteAllFilesInFolder](#) (object sender, EventArgs e)
 - *Delete all Images and Data Files from Folder.*
- void [Form1_Load](#) (object sender, EventArgs e)
 - *Loads all data when opening application.*
- void [Form1_FormClosing](#) (object sender, FormClosingEventArgs e)
 - *Saves all data when closing application.*
- void [InitializeComponent](#) ()
 - *Required method for Designer support - do not modify the contents of this method with the code editor.*

Private Attributes

- OpenFileDialog [ofd](#)
- bool **valid_port_selected** = false
- string **port_name**
- int **failCounter** = 0
- int **activeDropWatch**
- int **activeDropModeHead**
- int **activeNozzleValue**
- int **activeSpanValue**

- int **activeImageMode**
 - int **activeGapValue**
 - int **actFreq**
 - int **timeBoardOn** = -1
 - int **activePD_Polarity**
 - int **activeEncoderPosition**
 - int **activePDdirection**
 - int **activeImageHeadIndex**
 - bool **ImageHead1** = false
 - bool **ImageHead2** = false
 - bool **ImageHead3** = false
 - bool **ImageHead4** = false
 - bool **Head1ImageSend** = false
 - bool **Head2ImageSend** = false
 - bool **Head3ImageSend** = false
 - bool **Head4ImageSend** = false
 - bool **isRunning** = true
 - String **CurrentFileName**
 - String **datafolder** = Application.StartupPath.Replace("bin\\Debug", "Output Images\\File")
 - String **newpath** = Application.StartupPath.Replace("bin\\Debug", "Output Images")
 - String **dataFolderPath** = Application.StartupPath.Replace("bin\\Debug", "Output Images")
 - int[] **HeadPrintCountersStoredAsInt** = new int[4]
 - int[] **PreviousHeadPrintCounters** = new int[4]
 - int[] **HeadStatus** = new int[4]
 - byte[] **A_Bits** = { 0b10010010, 0b01001001, 0b00100100 }
 - byte[] **B_Bits** = { 0b01001001, 0b00100100, 0b10010010 }
 - byte[] **C_Bits** = { 0b00100100, 0b10010010, 0b01001001 }
 - byte[] **BitsArray**
 - System.ComponentModel.IContainer **components** = null
- Required designer variable.*
- System.Windows.Forms.ComboBox **cbSerialPort**
 - System.Windows.Forms.Button **btnConnectDisconnect**
 - System.Windows.Forms.CheckBox **ChbxIsConnected**
 - System.Windows.Forms.Button **btnPowerOnOff**
 - System.Windows.Forms.CheckBox **ChbxPower**
 - System.Windows.Forms.Label **lbStatus**
 - System.Windows.Forms.TextBox **txtbStatusBox**
 - System.Windows.Forms.Label **lbTimeOn**
 - System.Windows.Forms.TextBox **txtbBoardUpTime**
 - System.Windows.Forms.TextBox **txtbTemperatureOutput1**
 - System.Windows.Forms.NumericUpDown **nudTemperatureHead1**
 - System.Windows.Forms.NumericUpDown **nudVoltageHead1**
 - System.Windows.Forms.TextBox **txtbTemperatureOutput2**
 - System.Windows.Forms.NumericUpDown **nudTemperatureHead2**
 - System.Windows.Forms.NumericUpDown **nudVoltageHead2**
 - System.Windows.Forms.TextBox **txtbTemperatureOutput3**
 - System.Windows.Forms.NumericUpDown **nudTemperatureHead3**
 - System.Windows.Forms.NumericUpDown **nudVoltageHead3**
 - System.Windows.Forms.TextBox **txtbTemperatureOutput4**
 - System.Windows.Forms.NumericUpDown **nudTemperatureHead4**
 - System.Windows.Forms.NumericUpDown **nudVoltageHead4**
 - System.Windows.Forms.TextBox **txtbHeadStatus2**
 - System.Windows.Forms.TextBox **txtbHeadStatus3**
 - System.Windows.Forms.TextBox **txtbHeadStatus1**

- System.Windows.Forms.TabControl **tcDropWatchingAndImageModes**
- System.Windows.Forms.Button **btnReset**
- System.Windows.Forms.TabPage **DropWatchingTab**
- System.Windows.Forms.NumericUpDown **nudFrequency**
- System.Windows.Forms.NumericUpDown **nudNozzle**
- System.Windows.Forms.ComboBox **cbDropWatchHeadSelection**
- System.Windows.Forms.Label **label24**
- System.Windows.Forms.ComboBox **cbDropWatchMode**
- System.Windows.Forms.TabPage **ImageModeTab**
- System.Windows.Forms.NumericUpDown **nudSpan**
- System.Windows.Forms.Label **lbHeadIndex**
- System.Windows.Forms.Label **lbDropWatchingMode**
- System.Windows.Forms.Label **lbDropWatchingSpan**
- System.Windows.Forms.Label **lbDropWatchingNozzle**
- System.Windows.Forms.Button **btnFillCycleA**
- System.Windows.Forms.TableLayoutPanel **tableLayoutPanel1**
- System.Windows.Forms.Label **lbCurrentTemperature**
- System.Windows.Forms.Label **lbSetTemperature**
- System.Windows.Forms.Label **lbHead**
- System.Windows.Forms.Label **lbSetVoltage**
- System.Windows.Forms.Label **label40**
- System.Windows.Forms.Label **label37**
- System.Windows.Forms.Label **label39**
- System.Windows.Forms.Label **label38**
- System.Windows.Forms.Label **lbHeadStatus**
- System.Windows.Forms.TextBox **txtbPrintCounter4**
- System.Windows.Forms.Label **lbPrintCount**
- System.Windows.Forms.TextBox **txtbPrintCounter1**
- System.Windows.Forms.TextBox **txtbPrintCounter2**
- System.Windows.Forms.TextBox **txtbPrintCounter3**
- System.Windows.Forms.Label **lbFrequency**
- System.Windows.Forms.PictureBox **pictureBox1**
- System.Windows.Forms.PictureBox **pictureBox4**
- System.Windows.Forms.PictureBox **pictureBox3**
- System.Windows.Forms.PictureBox **pictureBox2**
- System.Windows.Forms.Label **lbImageHead4**
- System.Windows.Forms.Label **lbImageHead3**
- System.Windows.Forms.Label **lbImageHead2**
- System.Windows.Forms.Label **lbImageHead1**
- System.Windows.Forms.OpenFileDialog **openFileDialog1**
- System.Windows.Forms.TextBox **txtbFileNameHead4**
- System.Windows.Forms.TextBox **txtbFileNameHead3**
- System.Windows.Forms.TextBox **txtbFileNameHead2**
- System.Windows.Forms.TextBox **txtbFileNameHead1**
- System.Windows.Forms.Label **lbFileNameHead4**
- System.Windows.Forms.Label **lbFileNameHead3**
- System.Windows.Forms.Label **lbFileNameHead2**
- System.Windows.Forms.Label **lbFileNameHead1**
- System.Windows.Forms.TextBox **txtbDimensionsHead4**
- System.Windows.Forms.Label **lbDimensionsHead4**
- System.Windows.Forms.TextBox **txtbDimensionsHead3**
- System.Windows.Forms.Label **lbDimensionsHead3**
- System.Windows.Forms.TextBox **txtbDimensionsHead2**
- System.Windows.Forms.Label **lbDimensionsHead2**
- System.Windows.Forms.TextBox **txtbDimensionsHead1**

- System.Windows.Forms.Label **lbDimensionsHead1**
- System.Windows.Forms.TextBox **txtbHeadStatus4**
- System.Windows.Forms.Label **lbDropWatchingFrequencyDuplicate**
- System.Windows.Forms.TextBox **txtbFrequencyDuplicate**
- System.Windows.Forms.Button **btnClearHead**
- System.Windows.Forms.Button **btnPrintImage**
- System.Windows.Forms.Button **btnCancel**
- System.Windows.Forms.Button **btnFillNozzle**
- System.Windows.Forms.Button **btnFillSpan**
- System.Windows.Forms.Label **lbImageMode**
- System.Windows.Forms.ComboBox **cbImageMode**
- System.Windows.Forms.TextBox **txtbHeadStatus**
- System.Windows.Forms.Label **lbDropWatchingFillCycle**
- System.Windows.Forms.Button **btnFillCycleC**
- System.Windows.Forms.Button **btnFillCycleB**
- System.Windows.Forms.GroupBox **FillCycleBox**
- System.Windows.Forms.Button **btnFillGap**
- System.Windows.Forms.NumericUpDown **nudGap**
- System.Windows.Forms.Label **lbDropWatchingGap**
- System.Windows.Forms.ToolTip **toolTip1**
- System.Windows.Forms.Button **btnFillHead**
- System.Windows.Forms.ComboBox **cbPDPolarity**
- System.Windows.Forms.Label **lbPDPolarity**
- System.Windows.Forms.ComboBox **cbEncoderTrackedPosition**
- System.Windows.Forms.Label **lbEncoderTrackedPosition**
- System.Windows.Forms.ComboBox **cdPDDirection**
- System.Windows.Forms.Label **lbPDDirection**
- System.Windows.Forms.CheckBox **chbxIsFillSpan**
- System.Windows.Forms.CheckBox **chbxIsFillGap**
- System.Windows.Forms.CheckBox **chbxIsFillHead**
- System.Windows.Forms.CheckBox **chbxIsFillNozzle**
- System.Windows.Forms.TextBox **txtbImageHeadStatus**

Static Private Attributes

- static SerialPort **driver_board**

7.1.1 Member Function Documentation

7.1.1.1 btnCancel_Click()

```
void DriverBoardDropwatcher.Form1.btnCancel_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Cancel Button Pushed.

This function is called when the "Cancel" button is pushed. This is to cancel all printing and clear all heads

Runs only if driver board is connected and powered on.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.2 btnClearHead_Click()

```
void DriverBoardDropwatcher.Form1.btnClearHead_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Clear Heads Push Button.

This function is called when "Clear Heads" button is pushed

It sends the relevent command to the driver board whilst unchecking all other relevent checkboxes to indicate that all heads are cleared.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.3 btnConnectDisconnect_Click()

```
void DriverBoardDropwatcher.Form1.btnConnectDisconnect_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Controls the Connect/Disconnect Toggle.

This function controls the connection state of the driver board and modifies the checkbox as checked if connected or unchecked if disconnected

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.4 btnPrintImage_Click()

```
void DriverBoardDropwatcher.Form1.btnPrintImage_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Print Image.

This function is called when the Print Image Button is pressed.

Only runs if driver board is connected and powered on. Sends only the relevant images to print. For example, if images are uploaded to Head 1 and Head 3 and Head 2 and 4 are left blank, only Heads 1 and 3 will print.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.5 `cbDropWatchHeadSelection_SelectedIndexChanged()`

```
void DriverBoardDropwatcher.Form1.cbDropWatchHeadSelection_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Fetches User Selected Head from GUI.

This function is called when user modifies the head selection in the drop watching mode. Stores selected head in a variable called "activeDropModeHead".

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.6 `cbDropWatchMode_SelectedIndexChanged()`

```
void DriverBoardDropwatcher.Form1.cbDropWatchMode_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Fetches User Selected Drop Watch Mode Selection from GUI.

This function is called when the Drop Watch Mode Section is changed between Internal and External Modes. Stores user selected mode in a variable called "activeDropWatch" Only runs if driver board is connected

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.7 cbSerialPort_DropDown()

```
void DriverBoardDropwatcher.Form1.cbSerialPort_DropDown (
    object sender,
    EventArgs e ) [inline], [private]
```

Detects Active Serial COM Ports from device.

This function searches and detects any available COM Ports available to open.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.8 cbSerialPort_SelectedIndexChanged()

```
void DriverBoardDropwatcher.Form1.cbSerialPort_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Detects Changes in Serial Ports.

This function runs when the Serial COM Port is changed.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.9 connect_board()

```
void DriverBoardDropwatcher.Form1.connect_board ( ) [inline], [private]
```

Connect Driver Board.

This function sends command to driver board to connect and checks the checkbox in the GUI. Only runs if a valid port is selected and is connected successfully to the driver board. If this fails to run, then an Error Dialog Box shows up.

7.1.1.10 convertImageToData()

```
void DriverBoardDropwatcher.Form1.convertImageToData (
    string file_name ) [inline], [private]
```

Convert Image to Data.

This function is called when the Image is ready to be converted into data in order to send it to the driver board.

Stores image as data in a file with an extension of ".printDat" in a Folder Called "Output Images" in Project Folder.

Parameters

<i>file_name</i>	This is the full path of where the file is stored in the system.
------------------	--

7.1.1.11 DataRecievedHandler()

```
void DriverBoardDropwatcher.Form1.DataRecievedHandler (
    object sender,
    SerialDataReceivedEventArgs e ) [inline], [private]
```

Processes data received from driver board.

This function reads the data received from the driver board. If the substring matches to the expected character, then the data is sent to parse. If this fails to run multipe times, then an Error Dialog Box shows up.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.12 DeleteAllFilesInFolder()

```
void DriverBoardDropwatcher.Form1.DeleteAllFilesInFolder (
    object sender,
    EventArgs e ) [inline], [private]
```

Delete all Images and Data Files from Folder.

This function is called when the user closes the application. THis is to delete all files stored in the "Output Images" folder created to save memory.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.13 determineStatus()

```
void DriverBoardDropwatcher.Form1.determineStatus ( ) [inline], [private]
```

Determine the Status of Voltages and Temperatures.

This function determines what state each Head is in and outputs the Voltage and Temperature Values to the GUI. Only runs if a valid port is selected and is connected and powered on successfully to the driver board.

7.1.1.14 disconnect_board()

```
void DriverBoardDropwatcher.Form1.disconnect_board ( ) [inline], [private]
```

Disconnect Driver Board.

This function sends command to driver board to disconnect and unchecks the checkbox in the GUI.

7.1.1.15 Dispose()

```
override void DriverBoardDropwatcher.Form1.Dispose (
    bool disposing ) [inline], [protected]
```

Clean up any resources being used.

Parameters

<i>disposing</i>	true if managed resources should be disposed; otherwise, false.
------------------	---

7.1.1.16 DropWatchingTab_Click()

```
void DriverBoardDropwatcher.Form1.DropWatchingTab_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Drop Watching Tab pressed.

This function is called when the user clicks anywhere within the drop watching tab

This will set focus onto the tab allowing for other functions to run such as right and left keys increasing and decreasing the nozzle value.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.17 DropWatchingTab_PreviewKeyDown()

```
void DriverBoardDropwatcher.Form1.DropWatchingTab_PreviewKeyDown (
    object sender,
    PreviewKeyDownEventArgs e ) [inline], [private]
```

Arrow Keys Pushed.

This function is called when the user presses down on an arrow key. If right arrow key, then increase nozzle value by increments of 1 and send command to driver board. If left arrow key, then decrease nozzle value by increments of 1 and send command to driver board.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.18 EncoderTrackedPositionSelection_SelectedIndexChanged()

```
void DriverBoardDropwatcher.Form1.EncoderTrackedPositionSelection_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Encoder Position Selection Change.

This function is called when the user modifies the Encoer Position Mode

Sends relevent command to driver board if Encoder Position is changed between (Stepper Normal and Stepper Reverse)

Only runs if board is connected.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.19 FillCycle()

```
void DriverBoardDropwatcher.Form1.FillCycle (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Cycle Function.

This function is called by either Fill Cycle A,B or C.

Sends relevent commands in bytes to driver board in order to fill cycles.

Contains tags to identify which push button has been triggered.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.20 FillCycleA_Click()

```
void DriverBoardDropwatcher.Form1.FillCycleA_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Cycle A Button Press.

This function is called Fill Cycle A Button is Pressed. Calls the Fill Cycle function.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.21 FillCycleB_Click()

```
void DriverBoardDropwatcher.Form1.FillCycleB_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Cycle B Button Press.

This function is called Fill Cycle B Button is Pressed. Calls the Fill Cycle function.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.22 FillCycleC_Click()

```
void DriverBoardDropwatcher.Form1.FillCycleC_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Cycle C Button Press.

This function is called Fill Cycle C Button is Pressed. Calls the Fill Cycle function.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.23 fillHead_Click()

```
void DriverBoardDropwatcher.Form1.fillHead_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Head Button Press.

This function is called when the "Fill Head" button is pushed.

Sends relevent command to driver board and sets other checkboxes to unchecked to indicate which mode is running.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.24 FillSingleNozzleButton_Click()

```
void DriverBoardDropwatcher.Form1.FillSingleNozzleButton_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Nozzle Push Button.

This function is called when the "Fill Nozzle" button is pushed

It sends the relevent command to the driver board whilst checking the checkbox to show that it is running.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.25 FillSpanNozzleButton_Click()

```
void DriverBoardDropwatcher.Form1.FillSpanNozzleButton_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Fill Span Push Button.

This function is called when the "Fill Span" button is pushed

It sends the relevent command to the driver board whilst checking the checkbox to show that it is running.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.26 Form1_FormClosing()

```
void DriverBoardDropwatcher.Form1.Form1_FormClosing (
    object sender,
    FormClosingEventArgs e ) [inline], [private]
```

Saves all data when closing application.

This function is called when the User presses the close button

It stores all the user modified values/data and stores it ready to load up the next time the user opens it. This is to make everything more convenient for the user the next time they use this application instead of re-selecting all relevent options.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.27 Form1_Load()

```
void DriverBoardDropwatcher.Form1.Form1_Load (
    object sender,
    EventArgs e ) [inline], [private]
```

Loads all data when opening application.

This function is called when the user starts application

It loads all stored values/data from the last session. This is to make everything more convenient for the user saving repetitive actions.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.28 frequencyValue_ValueChanged()

```
void DriverBoardDropwatcher.Form1.frequencyValue_ValueChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Fetches Frequency Data from GUI.

This function runs when the Frequency Value is changed in the GUI so the new value is updated on the board. Only runs if driver board is connected and is powered on.

Uses tags to find relevent Numeric Up Downs and Labels in GUI.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.29 GapValue_ValueChanged()

```
void DriverBoardDropwatcher.Form1.GapValue_ValueChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Gap Value Modified.

This function is called the Gap Value Numeric Up Down Box is modified.

Stores new value in variable called "activeGapValue"

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.30 ImageBoxClicked()

```
void DriverBoardDropwatcher.Form1.ImageBoxClicked (
    object sender,
    EventArgs e ) [inline], [private]
```

Image Box Selection in Image Mode.

This function is called when user presses the picture box in GUI to upload image

Opens up file dialog and filters it out to only allow upload of image files. Clones the image and saves it into a folder called "Output Images" in the .png format

Uses tags to find relevent Picture Boxes (1-4).

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.31 ImageModeSelection_SelectedIndexChanged()

```
void DriverBoardDropwatcher.Form1.ImageModeSelection_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Runs if Image Mode Selection is Modified.

This function is user modified the image mode selection in GUI.

It stores the image mode into a variable called "activeImageMode" This function greys out other irrelevant combo box. For example, when Stepper Motor Mode is selected, the PD Polarity Settings are greyed out as they are irrelevant in this mode.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.32 MakeGrayscale3()

```
static Bitmap DriverBoardDropwatcher.Form1.MakeGrayscale3 (
    Bitmap original ) [inline], [static]
```

Grey Scale Image Function.

This function grey scales any image passed to it

Parameters

<i>original</i>	Original Image (Coloured)
-----------------	---------------------------

Returns

Grey-Scaled Image in Bitmap format

7.1.1.33 NozzleValue_ValueChanged()

```
void DriverBoardDropwatcher.Form1.NozzleValue_ValueChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Nozzle Value Modified in GUI.

This function is called when user changed value of the Nozzle (Numeric Up Down)

If span value + nozzle value is greater than 128, then a warning message will pop up automatically rectifying the error in the numeric up down.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.34 parseJsonData()

```
void DriverBoardDropwatcher.Form1.parseJsonData (
    string input_string ) [inline], [private]
```

Parse the JSON Data.

This function parses the data received from the driver board. Stores data such as Voltage, Temperature and Print Count into variables that can be called later

Parameters

<i>input_string</i>	The Data Lines received from the driver board in JSON Format
---------------------	--

7.1.1.35 PD_Polarity_SelectedIndexChanged()

```
void DriverBoardDropwatcher.Form1.PD_Polarity_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

PD Polarity Mode Selection.

This function is called when the PD Polarity Mode is changed.

Stores active mode in variable called "activePD_Polarity" Sends relevent command to driver board. Only runs if driver baord is connected.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.36 pdDirection_SelectedIndexChanged()

```
void DriverBoardDropwatcher.Form1.pdDirection_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

PD Direction Selection Change.

This function is called when the user modifies the PD Direction Setting Sends relevent command to driver board depending on which mode they have selected (continuous or single)

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.37 pictureBox1_Click()

```
void DriverBoardDropwatcher.Form1.pictureBox1_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Picture Box 1 is pressed.

This function calls the ImageBoxClicked function when Head 1 Image is modified

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.38 pictureBox2_Click()

```
void DriverBoardDropwatcher.Form1.pictureBox2_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Picture Box 2 is pressed.

This function calls the ImageBoxClicked function when Head 2 Image is modified

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.39 pictureBox3_Click()

```
void DriverBoardDropwatcher.Form1.pictureBox3_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Picture Box 3 is pressed.

This function calls the ImageBoxClicked function when Head 3 Image is modified

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.40 pictureBox4_Click()

```
void DriverBoardDropwatcher.Form1.pictureBox4_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Picture Box 4 is pressed.

This function calls the ImageBoxClicked function when Head 4 Image is modified

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.41 powerOff()

```
void DriverBoardDropwatcher.Form1.powerOff ( ) [inline], [private]
```

Power Off Function.

This function is called when the toggle mode is set to Power Off

Sends command to driver board.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.42 powerOn()

```
void DriverBoardDropwatcher.Form1.powerOn ( ) [inline], [private]
```

Power On Function.

This function sends the command to turn on the driver board. Only runs if correct COM Port is selected.

7.1.1.43 powerOnOff_Click()

```
void DriverBoardDropwatcher.Form1.powerOnOff_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Controls the Power On/Off Toggle.

This function controls the power state of the driver board and modifies the checkbox as checked if powered on or unchecked if powered off.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.44 PrintingImage()

```
void DriverBoardDropwatcher.Form1.PrintingImage (
    int head ) [inline], [private]
```

Printing Image Function.

This function sends the relevent command with the relevent print head to the driver board.

Read the image data file and stores it in an array ready to send in bytes.

Parameters

<i>head</i>	The Active Head with the Image Loaded onto it.
-------------	--

7.1.1.45 reset_Click()

```
void DriverBoardDropwatcher.Form1.reset_Click (
    object sender,
    EventArgs e ) [inline], [private]
```

Reset Board Button.

This function is called when user presses Reset Board Button in GUI Only runs if driver board is connected Sends relevent command to the driver board to reset it.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.46 SpanValue_ValueChanged()

```
void DriverBoardDropwatcher.Form1.SpanValue_ValueChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Span Value Modified in GUI.

This function is called when user changed value of the Span (Numeric Up Down)

If span value + nozzle value is greater than 128, then a warning message will pop up automatically rectifying the error in the numeric up down.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.47 tcDropWatchingAndImageModes_SelectedIndexChanged()

```
void DriverBoardDropwatcher.Form1.tcDropWatchingAndImageModes_SelectedIndexChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Mode Selection.

This function is called when the user switches between Drop Watching Mode and Image Mode

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.48 temperature_ValueChanged()

```
void DriverBoardDropwatcher.Form1.temperature_ValueChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Fetches Temperature Data from GUI.

This function reads data parsed from driverboard and outputs the required variables into the GUI. Only runs if driver board is connected

Uses tags to find relevent Numeric Up Downs and Labels in GUI.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.49 ThreadTask()

```
void DriverBoardDropwatcher.Form1.ThreadTask ( ) [inline], [private]
```

A Thread Task Function.

This function constantly runs in the background to ensure the board is connected at all times. It sends the command 'b' to receive relevent information such as voltage, temperature etc If this fails to run, then Error Message Box pops up signalling an error.

7.1.1.50 VerifyImageData()

```
void DriverBoardDropwatcher.Form1.VerifyImageData (
    int head,
    string existing_lines ) [inline], [private]
```

Picture Box 2.

This function calls the ImageBoxClicked function when Head 2 Image is modified

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.1.51 voltage_ValueChanged()

```
void DriverBoardDropwatcher.Form1.voltage_ValueChanged (
    object sender,
    EventArgs e ) [inline], [private]
```

Fetches Voltage Data from GUI.

This function reads data parsed from driverboard and outputs the required variables into the GUI. Only runs if driver board is connected

Uses tags to find relevent Numeric Up Downs and Labels in GUI.

Parameters

<i>sender</i>	The object that contains the reference to the object that raised the event
<i>e</i>	The event data

7.1.2 Member Data Documentation

7.1.2.1 ofd

```
OpenFileDialog DriverBoardDropwatcher.Form1.ofd [private]
```

Creates an OpenFileDialog variable called ofd

The documentation for this class was generated from the following files:

- [Form1.cs](#)
- Form1.Designer.cs

Chapter 8

File Documentation

8.1 Form1.cs File Reference

This C# Code contains the Source Code for the Driver Board Dropwatcher Application.

Classes

- class [DriverBoardDropwatcher.Form1](#)

Functions

- `[instance initializer]`

Variables

- `$ v`
- `$ T`
- `$ p`
- `$ n`
- `$ N`
- `$ I`

8.1.1 Detailed Description

This C# Code contains the Source Code for the Driver Board Dropwatcher Application.

Author

Added Scientific Limited (Kajeban Baskaran Internship Work)

