* **Valve Assembly Project**

**Overview of Valve Assembly Project :**

* It is used for inspection and measurement.
* It is for Accurate results with simplicity.
* For sorting of Production Inspection.
* Used for fast measurement and analysis time.
* It check Radius, Threshold, scratch detection, colour detection and image verification.

**Requirements :-**

1)C#

2)PLC (Programmable logic controller)

3)HALCON

**Working of project :**

The working of this project is that the cameras can capture the image of part/object then save in cam1 then we can select part and also choose which diameter is given then check quality, Total ok, Not ok. It can perform some properties like object threshold, object color, partradiusMax etc

**PLC :- (A PROGRAMMABLE LOGIC CONTROLLER)**

* A PROGRAMMABLE LOGIC CONTROLLER (PLC) is **an industrial computer control system that monitors the state of input devices and makes decisions based upon a custom program to control the state of output devices**.
* PLCs **communicate, monitor and control automated processes like assembly lines, machine functions, or robotic devices**.
* A PLC's functions are divided 4 main categories: input scan,output scan,program scan and housekeeping.
* **Input Scan :-** it detects the state of all input devices that are connected to plc.
* **Program scan :-** it executes user created program logic.
* **Output scan :-** Energizes or de-energizes all output devices that are connected to the plc.
* **Housekeeping :-** this includes communication with programming terminals, internal diagonstics etc.
* All working of cameras through plc.
* **Input devices :-** Switches,pushbuttons,sensing devices,sensors.
* **Output devices :-** Valves,motors,actuators relays,lights.

**HALCON :-**

HALCON is the standard software for machine vision with an integrated development (IDE) that is used worldwide.

It provides outstanding performance and a comprehensive ,NEON,AVX2 as well as GPU acceleration.

HALCON is the most of the powerful library available for standard machine, vision AI and embedded vision application.

* Image Processing and capturing.
* Image Library.

**Properties of Valve Assembly :**

* Object color
* ObjectThreshold
* PartRadiusMin
* PartRadiusMax
* BallRadius
* PinMinThreshold
* BallMaxThreshold
* PinAreaMin
* PinAreaMax
* BallAreaMin
* BallAreaMax
* PinOutArea
* Remove Burr

**Object color :-**

To choose the object color i.e dark,light.

**Object Threshold :-**

It is an part of an image processing to check the intensity of image.

**partRadiusMin :-**

used to set minimum radius of that part.

**partRadiusMax :-**

used to set maximum radius of that part.

**BallRadius :-**

To set the radius of that ball is present in that object/part.

**PinMinThreshold :-**

It is an image processing that set maximum pin image intensity.

**BallMaxThreshold :-**

It is used to display maximum ball image intensity.

**PinAreaMin :-**

It used to display minimum pin area of that object/part.

**PinAreaMax :-**

It is used to display maximum pin area of that object/part.

**BallAreaMin :-**

It is used to display min area of that part.

**BallAreaMax :-**

To check the maximum Ball Area in that object/part.

**PinOutArea :-**

To show pin area of an image.

**Remove Burr :-**

To remove Burr form an image.

**Methods Or Functions (Master.cs):-**

* **Delegate () :-**

It is an type that represent references to methods with particular parameter list and return type.

Delegates are used to pass methods as arguments to other methods.

**For e.g>** private delegate void DelegateUpdateList1(HTuple CM1Result, HTuple Hole, HTuple Center, int ImageIndex);

private DelegateUpdateList1 Del\_Cam1;

* **HTuple :-**

Tuples are generally used when you want to create a data structure which contains objects with their properties and you don’t want to create a separate type for that.

It allows us to represent multiple data into a single data set.

It allows us to create, manipulate, and access data set.

**For e.g>**

public static HTuple Pose, Campara;

* **InitializeComponent () :-**

This method used for calling design form or add info regarding that form.This method contains the code that creates and initializes the user interface objects dragged on form surface with values provided by you using the property grid of form designer.

**For e.g>** InitializeComponent();

win\_result1 = hSmartWindowControl1.HalconWindow;

win\_result2 = hSmartWindowControl2.HalconWindow;

Cam1ResultDict = new ConcurrentDictionary<int, bool>();

Cam2ResultDict = new ConcurrentDictionary<int, bool>();

FinalResultDict = new ConcurrentDictionary<int, bool>();

FlashDict = new ConcurrentDictionary<int, bool>();

FinalResultSendToPLCDict = new ConcurrentDictionary<int, bool>();

//ReadCalibration\_files(); // Not Required as we are reading calibaration file when selecting part from combo box

* **BeginInvoke () :-**

Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.

**For e.g>** this.BeginInvoke(Del\_DispayCounts);

* **HOperatorSet () –**

It is used of the callback HTuple callbackFunction.

* **GC.Collect () :-**

Use this method to try to reclaim all memory that is inaccessible. It performs a blocking garbage collection of all generations.

All objects, regardless of how long they have been in memory, are considered for collection; however, objects that are referenced in managed code are not collected.

**For e.g>** GC.Collect();

* **AcquisitionMode property() :-**

It defines how the value for this tag will be fetched.

**For e.g>** cam.AcquisitionMode.Value = AcquisitionModeEnums.Continuous.ToString();

**Methods Or Functions (Inspection.cs):-**

* **HOperator () :-**
* HOperatorSet.GetImageSize(ho\_Image, out hv\_Width, out hv\_Height);

//Prepare the metrology model data structure

* HOperatorSet.CreateMetrologyModel(out hv\_MetrologyHandle);

//Setting the image width in advance is not

//necessary, but improves the runtime of the

//first measurement.

* HOperatorSet.SetMetrologyModelImageSize(hv\_MetrologyHandle, hv\_Width, hv\_Height);

//Measure straight edges interactively using the drawing object.

* **GetCurrentProcess () :-**

It returns a reference to a process object representing the currently executing process.

System.Diagnostics.Process current = System.Diagnostics.Process.GetCurrentProcess();

* **Dispose() :-**

The Dispose method perform all object cleanup, so the garbage collector no longer needs to call objects.

hv\_box\_color.Dispose();

* **GenEmptyObj ():-**

Generate empty object.