Automated Check In Station

Team:
Kestutis Saltonas
Thao Tran
William Boyd
Matty Baba Allos

Version 1.0
January 27th, 2018

Background

Before conducting unit testing on manufactured chips, the product must be identified by barcode and visually inspected for defects. This machine will automate that process, which can become very tedious when done by hand. Instead of using magnifying lenses and microscopes to manually scan each chip for defects and to identify small markings to indicate the identity of the product, this device will use two inexpensive cameras and image recognition software to quickly and reliably read a barcode and identify visual defects before test.

Objective

This robotic check-in station will automatically capture component ID information from chips that need to undergo failure analysis in a lab environment. Primary goals are to have a fully automated check-in that reliably identifies the component and uses off the shelf hardware and software when possible. A secondary goal is to use CV software to perform visual inspection of the devices.

Automation of the initial check-in for failure analysis testing will be useful because human analysis is time consuming and prone to error. If chip identification can be automated, the process can be sped up and manpower required can be reduced. In addition, if the machine can identify external defects at a highly magnified level, it will not only save time and manpower, it can reduce some of the training required for the operators.

Requirements

Table 1: Functionality		
Req #	Engineering	Justification
1.1	Visual inspection should be done via open source software (opencv).	Open source software to minimize cost.
1.2	Must store inspection results in XML	XML is Intel's standard of choice
1.3	Must stitch captured Images together to create high-resolution image of CPU.	To increase the accuracy of visual inspection.
1.4	Must detect and read/scan barcodes.	Information for check-in
1.5	Must detect components (pads, passive components, die… ext).	Information for check-in
1.6	Should orientate Image after stitching, and must label front/back images.	For maintainability.
1.7	Must store images in compressed JPG format	To minimize image size
1.8	Must be able to sense insertion/removal of a magazine	To automate beginning of process
1.9	Must be able to detect if magazine is empty.	For maintainability.
1.10	Software system must identify device and look up device parameters.	To properly conduct visual inspection and for further steps in inspection process
1.11	System should use USB2 link hub.	To interface with a client computer
1.12	The software should be designed to allow the operator to place devices in any orientations.	To ease of use
1.13	Software must save high-resolution image of device.	For recording of inspection
1.14	Software should detect and read text.	For identification of device and prior notations
1.15	Cameras may operate independently	For ease operation

Table 2: Performance		
Req #	Engineering	Justification
2.1	Must process unit with 99% success in 10 seconds.	To be able to process many chips fast
2.2	Must stitch image together properly.	For visual inspection to work correctly
2.3	Should use template matching for visual inspection, may use AI based methods.	Template matching is easier to implement, AI based method can be more reliable/accurate.

Table 3: Economic		
Req #	Engineering	Justification
3.1	Overall system must cost less than $2500.	Maximum budget of the project
3.2	Cameras should cost less than $100.	Easily commercially available
3.3	System should use open source components.	To minimize the cost

Table 4: Power		
Req #	Engineering	Justification
4.1	The station should has its own power supply.	The machine does not depend on another device power

Table 5: Safety		
Req #	Engineering	Justification
5.1	Must incorporate safety interlocks to disable operation when device is open.	Mitigate possible damage or injury from operator error
5.2	Must be less than 5 kg and fit on standard workbench.	So it can be placed on any workstation desk safely
5.3	Software system must control safety features.	Because the software controls loading of machine, software must also control interlocks.
5.4	Must not use non-conducting plastics that might cause ESD.	To protect devices from damage

Table 6: Maintainability		
Req #	Engineering	Justification
6.1	System should be easy to assemble and maintain.	To be assembled and maintained by anyone
6.2	System should use off the shelf components.	To easily find and replace parts
6.3	Error handling and diagnostic messages.	In order for problems to be identify quickly.

Table 7: Operation		
Req #	Engineering	Justification
7.1	System must be lightweight.	For ease of mobility
7.2	System must have a magazine containing an array of devices.	To automate processing of several devices at a time
7.3	Magazine must be controlled by stepper motor and sensors.	Stepper motors are easy to control and can make precise, easily measured movements
7.4	System must be tight.	To be able to store it and place it in compact places
7.5	System should use arduino for control of physical components.	For simplicity and ease of use
7.6	Magazine must move in +/- x direction.	Magazine must be fed into machine
7.7	Magazine should have sliding supports.	Prefered option because of its flexibility
7.8	Magazine's capacity should be at least 6 devices.	Sponsor imposed
7.9	Magazine must hold devices only around edge.	To enable the cameras to examine all areas of the device
7.10	Frame should be made from 20mm extruded aluminum.	Material is easily available and cheap
7.11	Software system must control illumination.	To ensure that image processing is reliable
7.12	Magazine feed must be controlled by a system configuration file.	Ensure that hardware movement is initiated by Software system.
7.13	System must be able to handle any chip size	To able to scan a wide range of chips form factors

Table 8: Usability		
Req #	Engineering	Justification
8.1	System must use a Windows application.	Because Intel uses Windows PCs
8.2	System must be usable with from any PC	For using flexibility

Table 9: Documentation		
Req #	Engineering	Justification
9.1	The application should provide good user documentation.	Users need to know how to operate the machine
9.2	Developers should provide a demo of how to use the machine.	Teach users how to use the machine

Table 10: Reliability		
Req #	Engineering	Justification
10.1	Must be robust enough to ship.	To handle shipping rough environments
10.2	System must be stable in the event of all error conditions.	To prevent data loss and wasted time.