

A vision and robot based on-line monitoring of defects in Electronics Manufacturing – A collaborative effort in capstone project.

Immanuel Edinbarough, Subhash Bose

The University of Texas at Brownsville/The University of Texas – Pan American

Abstract

This paper discusses the integration of an automated neural network-based vision inspection system with robots to detect and report IC lead defects on-line. The vision system consists of custom software that contains a neural network database for each IC to be inspected on a PCB. The vision system uses gray scale images and a single layer neural network with three outputs based on defect criteria. Each IC has different inspection area, thus, the input vector varies for each ICs. The IC networks were trained with Matlab's Bayesian regularization module. This module was used because it prevents over and under training the image data. Performance of each of the networks investigated was found to be 100% based on the defect criteria. An on-line robotic inspection monitoring system has been developed, using ProE, C++ and OpenGL software^{1,2}. Technical issues and collaborative efforts in the execution of this capstone project are discussed in the paper. This research project was embarked as a collaborative effort between the senior design project students of the University of Texas at Brownsville and a graduate student of manufacturing engineering at the University of Pan American.

I Introduction

Capstone design courses provide excellent opportunity for students to work on the open ended problems that have direct bearing on the real life industry situations. There have been several models reported in literature including the one that deal with students work in teams on industry sponsored projects and deliver a tested prototype at the end of the course to get credit. In this capstone project model, student teams are formed between a graduate student team and under graduate team. The student teams are from different colleges. There are advisors for each of the student teams from either of the institutions. The problem that was identified for the capstone project is a

visual inspection system using neural networks to identify common defects in electronics manufacturing. A single layer network with multiple neurons is used to classify the defects. A user friendly visual method is developed in conjunction with other inspection methods such as, circuit testing, functional testing, and X-ray inspection methods where appropriate, for quality assurance. The graduate student designed the inspection algorithm using neural network technique and the under graduate student team worked on the implementation aspect of the inspection algorithm with the automated visual inspection system including the robot manipulation.

II Background

Technological developments have paved way for powerful electronics products with miniaturized components. Some examples of such products are laptop computers, palm pilots, cellular phones, and pagers. Although these products are highlights of electronics breakthroughs, manufacturing of these products has problems related to production yields and thus profitability. In a typical manufacturing plant, approximately 30% of all manufacturing tasks are related to inspection, of which 60% of inspection tasks are visual. The break down of typical defects found during visual inspection is approximately 30% part defects, 50% assembly defects (20% of which are incorrect parts or missing parts etc.), and 20% soldering defects³. It should be noted that, in order to maintain a certain level of quality in electronics manufacturing process, an increase in the number of solder joints by a factor of 10 requires that the number of defects be reduced by a factor of 10. Therefore, the effectiveness of an inspection system used to control a process will have a direct impact on the quality of products shipped to customers. This places an extraordinary pressure on human inspectors who are trained to identify defective parts by visual examination. This is further compounded if the equipment is used by the military, which requires 100% inspection of solder joints as required by MIL-STD 2000 for electronics assemblies. A study conducted by AT&T showed that a decision repeatability of only 44% for the same inspector inspecting solder joints, and only 6% agreement between their judgments with four inspectors checking the same board. This shows that the descriptions of allowable variations are very qualitative and subject to interpretation by inspection experts and leads to increases in manual inspection costs, which can be as high as 50% of assembly costs in some cases⁴. The literatures and techniques in the electronic parts visual inspection have something in common; they focus on conventional image processing and pattern recognition for classification of defects. The procedures used are intuitive but difficult to develop and are limited to particular application. Furthermore, these techniques need to be setup carefully and monitored by a skilled operator to ensure good results. Also, these techniques are computationally expensive, slowing down the inspection process; and thus, the production line. Any automated inspection system has to be able to implement an inspection system on-line, and if possible give feedback to other automated processes for continuous improvement. Neural network approach offers several advantages for automated inspection.

Neural networks have been used experimentally for decades. Neural networks are:

- Adaptive: infer solutions often capturing subtle relationships.
- Able to generalize: can handle imperfect or incomplete data.
- Nonlinear: can classify defects that are not linearly separable.
- Highly parallel: large networks can be realized using parallel processors to achieve real-time speeds.
- Tolerant of unusual noise distributions.

Although neural networks offer these advantages, the network performance suffers if the resolution of the defective image is poor (blurry). Therefore, high-resolution images (clear) that accurately portray the defect being classified are preferred to yield higher detection performance.

III Application of robots and vision system

In the electronics industry robotic assembly of PCBs reduce product cost and increase product quality. Cartesian and SCARA robots are typically used for assembling large or irregularly shaped parts onto PCBs (e.g., connectors, transformers, potentiometers, radial devices, crystals, light-emitting diodes, and large DIPs). To achieve accuracy and repeatability required in electronic assembly applications, mechatronics has become a necessity. Furthermore, Machine vision technology is increasingly applied to assembly and inspection tasks traditionally performed by human operators. This is mainly because miniaturization of electronic components has increased circuit density and makes human assembly and inspection virtually impossible⁵. Therefore in this capstone project, the integration of several disciplines is exploited. A Scorbot ER-V robot and a vision system are used to develop an automated inspection system. The Scorbot ER-V robot is used for loading and unloading the PCB board for automated visual inspection. The visual system then determines if there are any defects on the board. If there are defects, it will instruct the Scorbot ER-V robot to place the PCB board on a rework conveyor, otherwise, it will be sent to the next process.

IV Experimental Automated Vision Inspection System

The vision system consists of a standard CCD camera, a PC computer with data acquisition board, a customized six DOF robot, a Scorbot ER-V robot, and customized neural network software. The vision system uses multi-disciplines and standard manufacturing language to make the newly developed software easy to use in a manufacturing environment. A CCD camera is attached to the end effector of the robot. The programmer uses the PCB board's 2D CAD file to define the IC types and the centroid (x, y, z) of the IC. It is assumed that the IC type has been previously defined and stored in a software database. If the IC type exists in the database, then the directions and positions of inspection are predefined, so that the software can compute and store the optimal focal distances needed to obtain the largest clear image of the IC under inspection. Once the optimal focal distances are computed for all required orientations of the IC, the position of the camera can easily be found from

the centroid (defined by the programmer), the shape and directions predefined for the IC, and the maximum magnification and resolution of the CCD camera. With this information, the software uses the inverse kinematics equations of a six DOF robot to obtain the appropriate joint distances and angles. The software compensates for any delays and develops a sequence to achieve efficient movement and a clear image of the IC.

V Neural networks algorithm for defects identification and classification

Today, neural networks have become a flexible and a successful tool. It has been applied to aerospace, automotive, banking, defense, electronics, entertainment, financial, insurance, manufacturing, medical, oil and gas, robotics, speech, securities, telecommunication, and transportation.

Neural network topology

The automated inspection system uses a six DOF robot to position and orient the robot camera at a constant focal distance and angle to acquire IC's pin image. The positions of the ICs centroids and direction of acquisition have been previously defined by the programmer for a particular PCB. The IC's used for defect identification are gull-wing pin type. A schematic of this process is shown in Figure 1. The region of interest for defect identification is between the PCB board and the bottom bend of the gull wing pin. The defects that can be identified are the quality of solder joint, excessive pin bend, and lifted or missing pin. For solder quality, a resolution of at least 0.025 mm/pixel is needed, but the camera used has a resolution of only 0.1 mm/pixel. Therefore, the only defects that can be identified from the image are excessive bend and lifted pin defects.

The IC packages used for training, all had the same gull wing pin and same pin height, but the pitch varied. In real world applications the height and pitch for each different IC varies. Consequently, for the neural net topology, each IC that has different configuration are trained individually and have its own identification network. The Graphical Device Interface GDI software and the automated visual inspection system select which network to be used depending on the IC being inspected.

VI Graphical device interface (GDI) software

The GDI software has several features. One of the features is to keep a database of all the ICs network weights and biases that are trained. The training is performed offline with acquired image data using Matlab. The GDI software has the capability of adding new ICs network weights and biases. The GDI software loads a text program of a particular PCB that contains the number of IC to inspect and a list of text files containing the ICs inspection information (Figure 2). The files are read sequentially to extract information on an IC. The information extracted contains the network type, inspection direction, number of images to acquire, and the robot execution procedure (Figure 3). The six DOF robot uses this information to move the camera and acquire four images and saves the data in files Chip111.txt, Chip112.txt, Chip113.txt and Chip114.txt. The GDI software will then use the Chip Type to select

the appropriate network and Chip data text files to identify defects. If defects are found, the GDI software reports the defect type and the pin number with additional information on the screen and to a report text file for the PCB being inspected. This file can be used by an automated rework station or a rework operator to correct the defect. The above procedure is repeated until all IC text files are read. At this point, the PCB is fully inspected and another PCB file can be loaded to continue inspection. This file could be different and corresponds to a different PCB board. Thus the GDI and inspection system are flexible and allows inspection of different PCB boards sequentially. In addition to this flexibility, the GDI software shows a visual simulation of what is being inspected as the six DOF robot performs the task on-line. This would allow monitoring of inspection process at a remote location with feedback provided to the operator.

VII Acquisition of data

Pin defects were created on ICs of PCB boards. Since there were insufficient boards available to create defects and obtain sufficient data, several images of ICs with no defects were obtained and new images were created that simulates defects. Table 1 shows the defect criteria used by Matlab defect generation program. The Matlab defect generation program reads in a text file (containing one integer value for each pin defect wanted) and a real image containing no pin defects (Figure 4 (a)). The defect generation program outputs an image file containing the defects (Figure 4 (b)) and a text file containing columns of the neural network responses corresponding to each pin defect shown as positive and negative ones in a row (Table 1). Note the first two columns of the network output response shown in Table 1, corresponds to a binary number. For example, -1 -1 corresponds to a zero decimal value, which is used to classify a straight pin, not bent. A binary number -1 1 corresponds to a decimal value of one, bent pin 1 pixel to the right or left and so on. Also the last column of the network output response corresponds to a lifted or not lifted pin state, -1 for not lifted and 1 for lifted respectively. The image of Figure 4 (a) was acquired from a quad package containing 40 gull wing pins on each side for a total of 160 pins. The pitch of this package is 0.3 mm. The image height is 32 pixels. The width of the pin is 3 pixels, and the gap between pins is 3 pixels. Thus the pitch is 3 pixels (0.3 mm), which corresponds to 0.1 mm/pixel CCD camera resolution. Matlab's imaging toolbox and functions were used to read the image file, convert the grayscale file into numerical values from 0-255 into an array of numbers which has a 1 to 1 correspondence with row and column of the pixel value in the image. Once this was accomplished and the locations of pin data (rows and columns) are known from the image, it is easy to generate bent pins to the left or to the right of its central location, and also lifted or broken pins from the input text file. Soldering defects (insufficient or excess solder) were not generated because the resolution of the CCD camera was 0.1mm/pixel, which is not sufficient to detect soldering defects. A resolution of at least 0.025mm/pixel is needed for solder defect identification. The newly generated numerical array of pins with defects is then transformed to an image with Matlab image function and saved as an image file along with its neural network response text

file. The defect image and the output response text files were then used to train several networks and to find the optimum network size and the network architecture.

VIII Network architecture

Bayesian regularization was used for training the neural networks with Matlab's trainer function. Each IC has its own neural network used for defect identification. The reason being, the pitch of IC packages vary. Also, pin height, pin configuration, and package technologies are different. It would be extremely difficult to find one neural network that can identify defects for all ICs with different package technologies, pin configuration, etc. Also, due to limited resolution of the CCD camera, only the defects shown in Table 1 were considered to be possible in a manufacturing environment. Obviously the network used in this project has three outputs corresponding to the defects shown in Table 1. Generally, a network is started with sufficient neurons and layers. It is common to start with a three layer neural network and then reduce or increase the size as necessary to find an optimum network that has good performance and is able to generalize. The following networks were tried for one IC type of the following sizes 288:6:3:3, 288:6:3, 288:3:3, 288:2:3 and 288:3. The value of 288 is the size of the input vector corresponding to one pin inspection area. The inspection area consists of a width of 9 pixels (3 pixel gap + 3 pixel pin + 3 pixel gap) multiplied by 32-pixel height for a pin totaling 288 pixel values or inputs. The input pixel values are each divided by 255 (max gray scale value) to keep the input normalized between 0 and 1. The numbers after each colon represent the number of neurons in its corresponding layer respectively. For example, a 256:6:3:3 neural network is composed of an input vector of 256 normalized pixel values with 6 neurons in the first layer 3 neurons in the second layer and 3 neurons in the output layer. All the networks above were successful in classifying the defects presented to the network, except 288:2:3 network which consistently failed to identify the defects in Table 1 of 3 pixels bent left or right and lifted. Therefore, the minimum network size was reduced to 288:3 neural network, 1 layer network with 3 neurons and 288 inputs and 3 outputs. The same 1 layer network with 3 neurons and different input vector size corresponding to a different IC pitch and pin height was used to train similar ICs with the same defects criteria given in Table 1. Figure 4 (b) image and output response file generated were used to train the 288:3 neural network. Figure 5 (a), (b), and (c) show that each graph contains the corresponding outputs of each output neuron, the correct output response for that neuron and the error corresponding to the defect generated. Each graph has 3 lines - correct output response, network output response and the difference between the actual value and the network output for that neuron (error). It can be seen from the graphs, the error response is flat and very close to zero. This indicates perfect identification by the neural network. One other network was trained using the same procedure discussed previously on IC-two. Similar results were obtained. The training time for the networks was from 8 to 20 minutes on a 400MHz personal computer. The weights and biases for the ICs were saved as text files for use by an on-line monitoring system.

IX On-line robotic vision inspection process

There are provisions available for the operator to run the inspection process in simulation mode as well as in on-line mode. Simulation options under the automated simulation menu, allow the user to animate the process of an actual system and estimate inspection time approximately. To use the on-line option, the user has to select the communication settings correctly under the settings menu and then select the connect option under the action menu. At this point the user can type any legal controller command, for example home. If the Scorbot ER-V robot homes successfully, communication is set correctly. If everything has been set correctly and the user can communicate with the Scorbot ER-V controller, on-line monitoring of inspection of PCB could begin.

The on-line option under the automated inspection system menu uses real data and monitors the inspection system on-line while an animation of the process is visualized on the screen. A report of the pin data is saved in a report text file for the PCB under inspection. Figure 6 shows a sample of the report file. If the on-line system inspects five PCB boards, there will be five PCB board defect reports. Currently, these reports are saved as text files, but in the future these files will be saved as binary files and possibly compressed to minimize disk space.

X A successful collaborative effort

The collaborative capstone project was an outstanding success. The advisors from the respective institutions took care of the project management issues and kept the students stay with schedule. Students communicated through the project management web site, developed and maintained for the project. Since the institutions are within an hour of driving distance, students traveled back and forth during the course of the project. The learning experiences that the undergraduate students gained by communicating with the graduate student is phenomenal. The graduate student also had a first hand experience of understanding the dynamics of undergraduate student team project execution. The student teams had their respective final project presentations and the report to submit. The graduate student was graded on the basis of the work that he has performed. The deliverables for him were a thesis report and a defense. For the undergraduate students a progressive project presentations and a final report served as the basis of grading. The math and program intensive algorithm development part of the project was covered by the graduate student. The implementation of the vision algorithm, robot programming and the integration of the automated vision system including the robotic manipulation were carried out by the undergraduate student team.

XI Conclusions

Based on the defect criteria given in Table 1, it was determined that the optimum neural network size for the ICs being inspected was a single layer network with three outputs. Simulated and real defect data were used to test the robustness of the neural network. The performance of the network was found to be 100 percent

accurate. The defect criteria were kept constant for all ICs. The only thing that varied was the inspection area used for each IC due to the fact that the ICs have varying lead pitch, lead height, and number of leads. Each of the ICs was trained separately, with each IC having a varying input vector, and thus a different network with its own weights and biases. The weights and biases for each of the networks were then saved as text files and used by an on-line monitoring system, which stores the information in an internal database.

The collaborative capstone model, utilizing the graduate and under graduate student potential in the design, organization, management, program development and project implementation has not only proven effective but also promoted the excellent learning opportunities among the project teams.

Bibliography:

- [1] R. Balderas and S. Bose, "Automated Robotic Inspection System for Electronic Manufacturing", MSE Thesis, Manufacturing Engineering Dept., UT-Pan American, 2002.
- [2] A.I. Edinbarough, Jose Amieva, "Experimental Study on the Robotics Vision Inspection of Electronic Components", BS Thesis, Engineering Technology Department, UT-Brownsville, 2002.
- [3] S. Hata "Vision Systems for PCB Manufacturing in Japan," IEEE Transaction, Production Engineering Research Laboratory Hitachi, Ltd., pp. 792-797, 1990.
- [4] V. Sankaran, B. Chartrand, D.L. Millard, M.J. Embrechts, and R.P. Kraft, "Automated Inspection of Solder Joints-A Neural Network Approach," IEEE Int'l Electronics Manufacturing Technology Symposium, pp. 232-237, 1995.
- [5] T.L. Landers, W.D. Brown, E.W. Fant, E.M. Malstrom, and N.M. Schmitt, *Electronics Manufacturing Processes*, Prentice Hall, Upper Saddle River, NJ, 1994.

IMMANUEL EDINBAROUGH

Dr. Immanuel Edinbarough is a Professor in the department of Engineering Technology at the University of Texas at Brownsville. He has successful track record spanning over 23 years in the service oriented and challenging fields of *academia, industry and military*. He has 6 years of industrial experience in the field of machine tools manufacturing. He has been in the teaching profession for the past 15 years, 3 years as a visiting professor at the Rochester Institute of Technology. His teaching and research interests are in the areas of Automation, Robotics, Machine Vision, and CAD/CAM/CIM. He has published several papers, in these areas, in various national & international conferences and journals.

SUBHASH BOSE

Dr. Subhash Bose is a Beecherl Professor and Chair of Manufacturing Engineering Department at the University of Texas-Pan American, Edinburg, Texas. He received his MS (1984) and Ph.D. (1987) from the University of Wisconsin-Madison, Wisconsin. Dr. Bose had taught at Wright State University, Dayton, Ohio and Utah State University, Logan, Utah before joining UT-Pan American. His areas of interests are robotics, control and CAM.

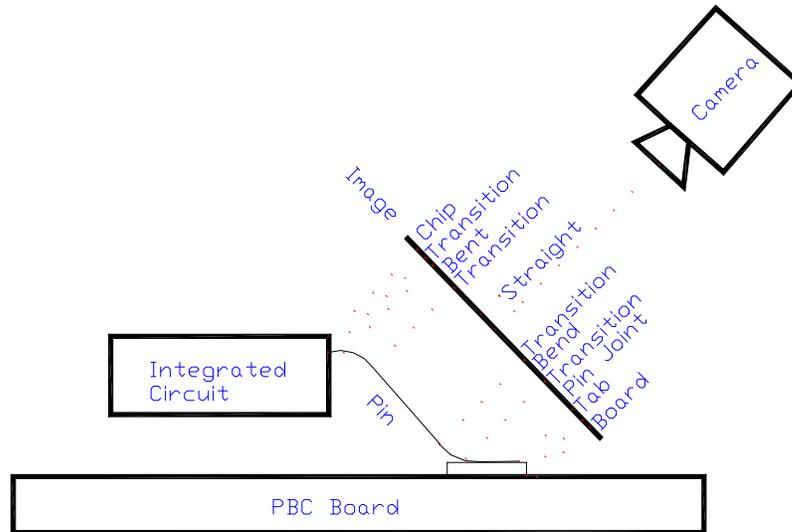


Figure 1- Schematic of image acquisition of a six DOF robot

ChipNumber1.txt
ChipNumber2.txt
ChipNumber3.txt

Figure 2- Information contained within PCB1data.txt file

```

Chip Number      1
Chip Type        1
Number of Pins   40
Pin Height in Pixels  32
Pin Width in Pixels  3
Pin Gap in Pixels  3
Number of Images  4
Inspection Direction Sequence L B T R
Chip111.txt Chip112.txt Chip113.txt Chip114.txt
Chip Centroid (X,Y,Z) in mm  1    1    1
Mold Width in mm    120
Mold Height in mm   120
Mold Pin Separation in mm  1
Inspection Angle in (degrees)  10
Focal Length in mm  1000
Resolution in mm/pixel 0.1

```

Figure 3- File for the ChipNumber1.txt of IC contained within PCB1data.txt file

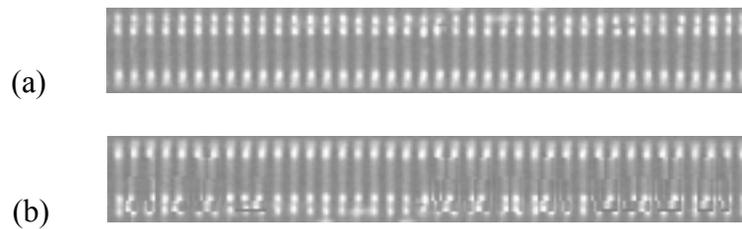


Figure 4- (a) Real image with no defect (b) Simulated defect image

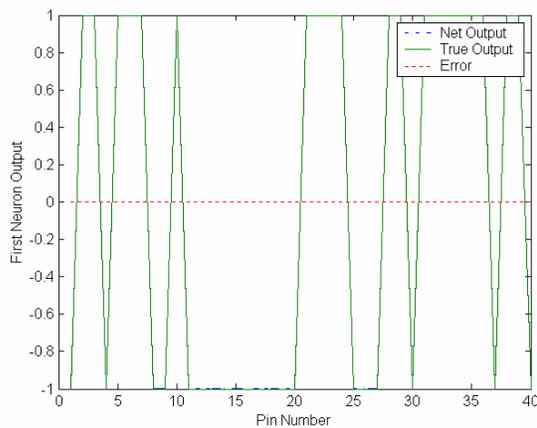


Figure 5- (a) Output Performance of 1st Neuron

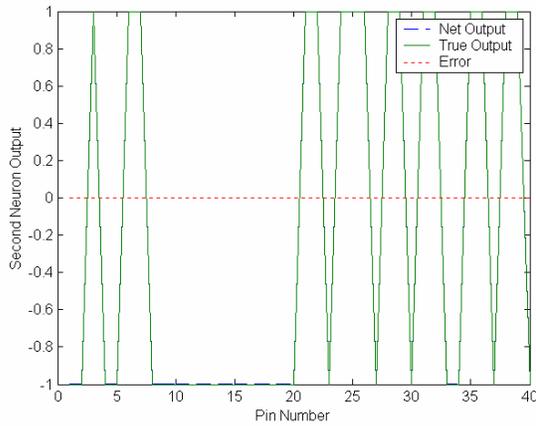


Figure 5 - (b) Output Performance of 2nd Neuron

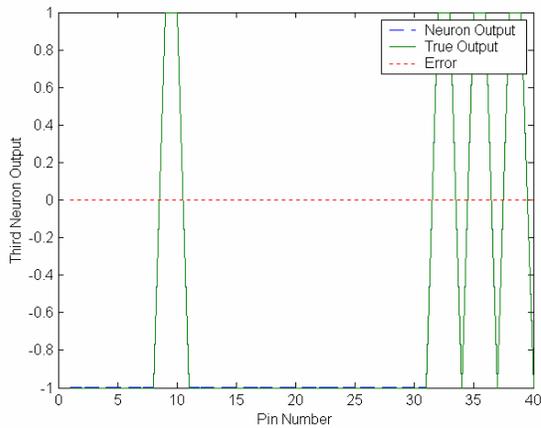


Figure 5- (c.) Output Performance of 3rd

Chip Number 1
 Chip Type 1
 Inspection Direction Left

Pin Number 1 Straight Pin
 Pin Number 2 Bent Pin 2 Pixel To the Right or Left
 ...
 Pin Number 39 Bent Pin 3 Pixel To the Right or Left And Lifted
 Pin Number 40 Straight Pin

Chip Number 1
 Chip Type 1
 Inspection Direction Bottom

Pin Number 1 Straight Pin

Pin Number 2 Bent Pin 2 Pixel To the Right or Left
 Pin Number 3 Bent Pin 3 Pixel To the Right or Left

...
 Chip Number 1
 Chip Type 1
 Inspection Direction Right

Pin Number 1 Straight Pin
 Pin Number 2 Bent Pin 2 Pixel To the Right or Left

...
 Chip Number 2
 Chip Type 2
 Inspection Direction Right

Pin Number 1 Straight Pin

...

Figure 6 PCB board defect report

Table 1 Criteria used for defect generation for data generation

Integer	Defect Type	Neural Net Response
0	Straight Pin and Not Lifted	-1 -1 -1
1	Bent Pin 1 Pixel to the Left and not Lifted	-1 1 -1
2	Bent Pin 1 Pixel to the Right and not Lifted	-1 1 -1
3	Bent Pin 2 Pixel to the Left and not Lifted	1 -1 -1
4	Bent Pin 2 Pixel to the Right and not Lifted	1 -1 -1
5	Bent Pin 3 Pixel to the Left and not Lifted	1 1 -1
6	Bent Pin 3 Pixel to the Right and not Lifted	1 1 -1
7	Straight Pin and Lifted	-1 -1 1
8	Bent Pin 1 Pixel to the Left and Lifted	-1 1 1
9	Bent Pin 1 Pixel to the Right and Lifted	-1 1 1
10	Bent Pin 2 Pixel to the Left and Lifted	1 -1 1
11	Bent Pin 2 Pixel to the Right and Lifted	1 -1 1
12	Bent Pin 3 Pixel to the Left and Lifted	1 1 1
13	Bent Pin 3 Pixel to the Right and Lifted	1 1 1

