

# Colour Sorting System with Robotic Arm

**Mr. Pranay S. Bhaskar**  
Assistant Professor

*Department of Electronics & Communication Engineering  
Dr. Babasaheb Ambedkar College of Engineering & Research  
Nagpur, India*

**Saurabh S. Dofe**  
UG Student

*Department of Electronics & Communication Engineering  
Dr. Babasaheb Ambedkar College of Engineering & Research  
Nagpur, India*

**Saurav S. Tripathi**  
UG Student

*Department of Electronics & Communication Engineering  
Dr. Babasaheb Ambedkar College of Engineering & Research  
Nagpur, India*

**Himanshu R. Banode**  
UG Student

*Department of Electronics & Communication Engineering  
Dr. Babasaheb Ambedkar College of Engineering & Research  
Nagpur, India*

## Abstract

Usually sorting of objects is carried out manually using human labor. Recognizing a particular object and placing it in the desired position is a laborious work especially in the field of industry where in one has to sort a bulk of objects in instant and also the load is greater than what a human can carry. This is when automation plays a major role. We have a robotic arm which picks different colored cubes and sorts them placing in different cups. The color processing is made using image processing with a webcam. The robotic arms are widely used in the industry, but most of them are used in PTP (Point To Point) trajectory, the moves are learned previously by the robotic arm. Very few robots in the industry are programmed to be clever, or to make decisions. In the future to completely replace the humans with robots, we require robotic arms which can make decisions. One good example for a smart robotic arm can be a robotic arm which can perform sorting of objects by color. This can be used in many factories; one good example can be a pencil factory.

**Keywords:** commanding, controller, remote-controlled, robotic arm, serial interfacing, sorting, webcam

## I. INTRODUCTION

In today's industrial environment a robot or more precisely a robotic arm is not something that is hard to find, we can find a mechanical assistant for the human worker in almost any factory.

In most of the factories we can find specified robots, which were made for that specific task and they cannot be reconfigured to other task, mostly of their mechanical setup.

We created an application for the robotic arm to sort colored cubes. Color sorting can be used in many factories where we deal with colored objects. One example can be to sort colored pencils or colored clay for kids, but many more utilities can be found for a system like this.

Technology that is expanding and is widely used at present is image processing. Signal in digital form is processed in image processing, where the input is an image which is processed to obtain some values, parameters or the set of characteristics related to that particular image.

## II. PROBLEM FORMULATION

We had the idea to sort with it colored cubes. We had also webcams for the image recognition part. We had all prime materials for the experiment, all that needed to be done is the color sorting algorithm and to implement it in a computer application.



Fig. 1: The Robotic Arm

The robotic arm has a servo control board which does a big part of the operation; it can control all the motors of the robotic arm according to its program.

On Fig.2. we can see the servo control board. This board is called SSC-32, because it can control up to 32 motors. The servo control board has an ATmega168 microcontroller which can communicate with the PC on RS-232 interface and can interpret the SCPI (Standard Commands for Programmable Instruments) commands sent from PC. According to these commands the servo control board will move the motors

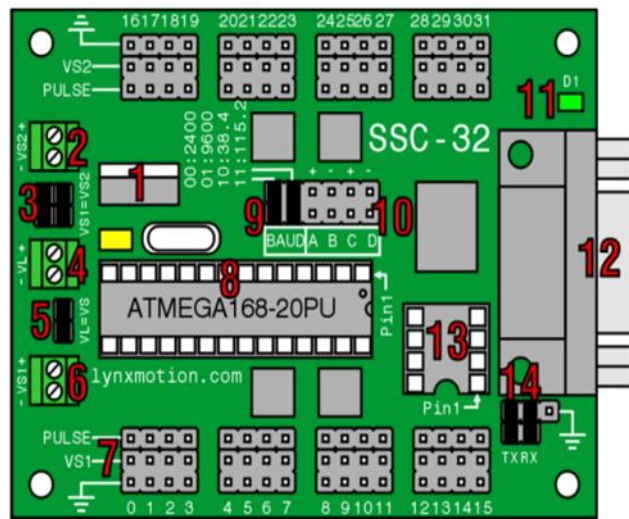


Fig. 2: The SSC-32 servo control board

### III. PROBLEM SOLUTION

#### A. Theoretical Background:

We needed to find the SCPI commands for controlling the robot on RS-232 interface. We found a free tool called HHD Free Serial Monitor to spy the traffic on the serial port. We connected the robot to the computer's RS-232 serial port and we started the third party applications. We could spy the SCPI commands with the application, after that we wrote them down and introduced them in our application. In the Figure 3 we can see the HHD Fee Serial Monitor (COM port spying tool.)

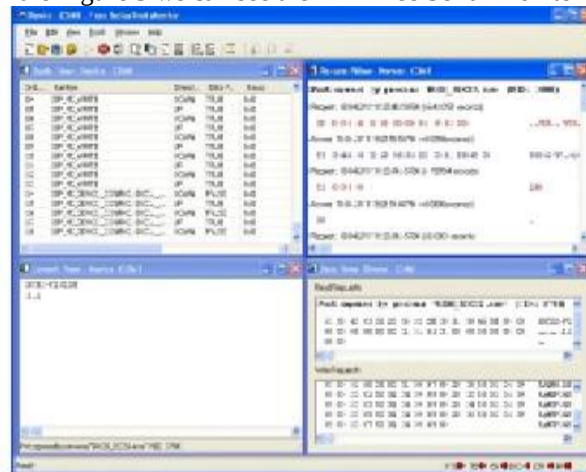


Fig. 3: Spying the SCPI commands with the HHD Free Serial Monitor tool

The commands for the SSC-32 servo control board are the Following:  
 // SSC-32 VERSION  
 \r\rVER\r  
 // INITIALIZE MOTORS  
 QPL0\rQP0\r  
 QP1\r  
 QP2\r  
 //...  
 QP31\r  
 // ALL SERVOS 1500  
 #0P1500S0\r#1P1500S0\r#2P1500S0\r#3P1500S0\r#4P1500S0\r#5P1500S0\r  
 // GRIPPER  
 #4P1500S1000\r

```
// WRIST ROTATE
#5P1500S1000\r
// WRITST
#3P1500S1000\r
// ELBOW
#2P1500S1000\r
// SHOULDER
#1P1500S1000\r
// BASE
#0P1500S1000\r
```

The commands need to be implemented in the given order. First the version is interpreted with `\r\rVER\r`. This command is vital, because this command establishes the link between the PC and the robotic arm, putting the SSC-32 direct board in remote approach.

After this the motors are initialized by testing all the 32 digital ports. If we used only 5 or 6 motors, it is sufficient to test only the digital ports where motors are connected. The command is `QPx\r`, where  $x = \{0, 31\}$ , QP meaning Query Port. This test it's not important, because even without it the motors can be commanded, but it's a useful self-test. Almost all electronic equipments do a self-test when starting; even the personal computers do a self-test during boot time. After no error on the self-test we can continue.

One second step could be to put the motors in the mid position. Our motors are mechanically blocked, this way they can't rotate more than 180°. When we put the motors in central position, it means that we put them in the 90° position. The robots mechanic is made in such way that if we put the motors in the mid position, the robotic arm shall be in a position analogous to the Greek letter gamma ( $\Gamma$ ); we can say that we "woke" to robotic arm. One example command is `#0P1500S0`; in general form is `#xP $\omega$ St`. In the command  $x = \{0, 5\}$  is the number of the motors, 0 for the base, 1 for the shoulder, 2 for the elbow, 3 for the writ, 4 for the gripper, 5 for the wrist rotate. Our robot hasn't mounted the wrist rotate motor. After the P (pulse), we have the location  $\omega = \{500, 2500\}$  of the motor, this way the middle position is  $\omega = 1500$ . The location of the motor is not set in angles, but in values from 500 to 2500. To know exactly the angles we can simply calculate with equation (1).

$$\alpha = \frac{\Delta w}{180^\circ} = (2500 - 500)/180^\circ \quad (1)$$

This means the following shown in equation (2).

$$10 = 11, (1) \quad (2)$$

After the S letter (speed) we have the time  $t$  in milliseconds. The time  $t$  shows the timeout when moving the motors. If we have 0, than the motors shall move immediately, this means it will not wait when moving motors. This is used for putting the robot's motors in mid position, "waking" the robot or as specified in the Lynx motion site, all servos 1500. If we have a value of 1000, this means we shall have a timeout of 1 s.

The color recognition part has some a quite big theoretical background too.

The color recognition is made live with the Trust WB-3400T standard webcam (Fig. 4.) using the blob filter. The webcam is a standard USB webcam with VGA (640x480 pixels) resolution. It has nothing special, but for the driver was important to have a camera on USB and this was the only available in our laboratory.



Fig. 4: The Trust WB-3400T webcam

To detect the dynamic range of an image we need the minimum and maximum pixel. These pixels can determine the apparel of an image. Equation (3) shows the dynamic range.

$$z = \frac{x-y}{u-y} \quad (3)$$

In the equation  $y$  is the value of the 8 bit pixel,  $x$  is the value of the 16 bit pixel,  $z$  is the value of the minimum intensity and  $v$  is the value of the maximum intensity.

When we convert the value of the pixel in a coordinate in the real world, an error will appear. The pixel coordinate error value will show the highest estimated location of the coordinate in the real world. Equation (4) shows how to compute the error value.

$$e(i, j) = \sqrt{(x - x_{real})^2 + (y - y_{real})^2} \quad (4)$$

In binary palette we can make difference between 15 colors. In Table I. we can see the 15 colors and their grey value represented with  $g$ .

TABLE – 1  
GREY VALUES IN BINARY PALETTE

| $g$ | R   | G   | B   | Resulting Color |
|-----|-----|-----|-----|-----------------|
| 1   | 255 | 0   | 0   | Red             |
| 2   | 0   | 255 | 0   | Green           |
| 3   | 0   | 0   | 255 | Blue            |
| 4   | 255 | 255 | 0   | Yellow          |
| 5   | 255 | 0   | 255 | Magenta         |
| 6   | 0   | 255 | 255 | Water Blue      |
| 7   | 255 | 127 | 0   | Orange          |
| 8   | 255 | 0   | 127 | Purple          |
| 9   | 127 | 255 | 0   | Shining Green   |
| 10  | 127 | 0   | 255 | Violet          |
| 11  | 0   | 127 | 255 | Sky Blue        |
| 12  | 0   | 255 | 127 | Sea Green       |
| 13  | 255 | 127 | 127 | Rose Red        |
| 14  | 255 | 255 | 127 | Spring Green    |
| 15  | 127 | 127 | 255 | Periwinkle      |

### B. Experimental Setup

The experiment setup of the color sorting robot system is shown in Fig. 5. As we can see we have the three colored cubes (red, blue, yellow), we have the webcam connected to the PC on USB interface and we also have the AL5A robotic arm with the stand connected to the PC on RS-232 serial interface. In the software part we have the SCPI commands for controlling the robotic arm, the commands are sent on RS-232 serial interface with the National Instruments' VISA driver. For the color recognition part we have the USB generic webcam driver and the library to use in programming. For recognizing colors we used the blob filter. This filter draws a big colored spot over the recognized color and the rest will be transformed in black or something neutral. This way it can recognize almost any color, because it makes comparison between already known colors in the computer palette. It doesn't know the object or the shape of it, but it can see the color of it and this is what is important for us. For the blob filter we put some limits, to make difference between similar colors. After it we print the decision and command the robotic arm to execute the specific task, to put the colored cube in the right part of the carton box. Fig.5. Experimental setup of the color sorting robot system to simplify the system a little bit we needed to make a stand. The setup of the stand for robotic arm is shown in Fig. 5. Fig.5 Picture of the stand for the robotic arm.

The stand had a dosing system for the colored cubes. The cubes are fed from the upper part and the robot picks them in the bottom, this way the robot will get the cube at the same place all the time. This can ease some of the work. We couldn't find so small cubes that would fit in the robot's gripper, this way we made the cubes manually, we them from wood and painted with car paint sprays. We also placed a carton box with three places for the robot to put the three differently colored cubes (red, blue and yellow). The robot has been preprogrammed to put cubes in all the three places of the carton board.

All the three movements of the robotic arm are saved in three different text files, after this the only task of the robot is to decide on the color and after to read the specific text file which will guide it in the right part of the carton box.

We can see that the webcam is quite close, this way we can recognize colors more easily. We also use a desk lamp; this is just for the webcam only, because the webcam has a faster refresh rate when we have more brightness. The webcam almost works in slow motion when we have poor lighting, but the system works fine without the desk lamp too.

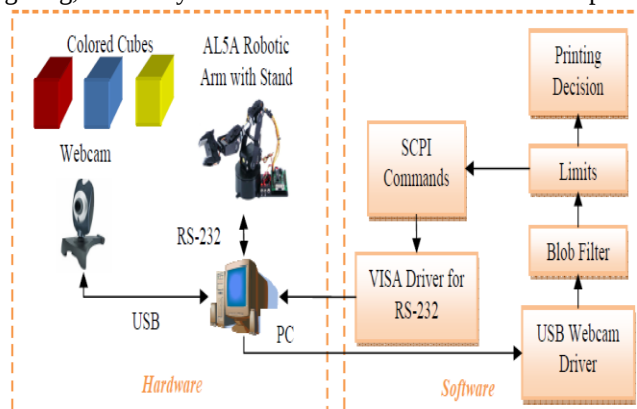


Fig. 5: Experimental setup of the color sorting robot system

### C. The Color Sorting Application

The application was made in Lab Windows/CVI from National Instruments. We chose this programming language; because it's a standard ANSI C programming language, but it has many drivers for the specific computer interfaces. We can see on Fig. 6. Fig. 7 and Fig. 8. The applications and the decisions for each color (red, blue and yellow).

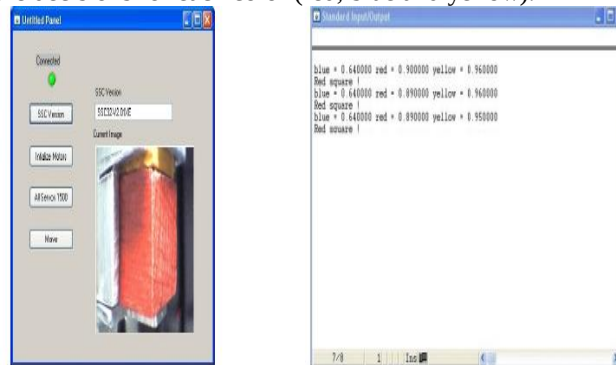


Fig. 6: The application and the decision when recognizing the red cube

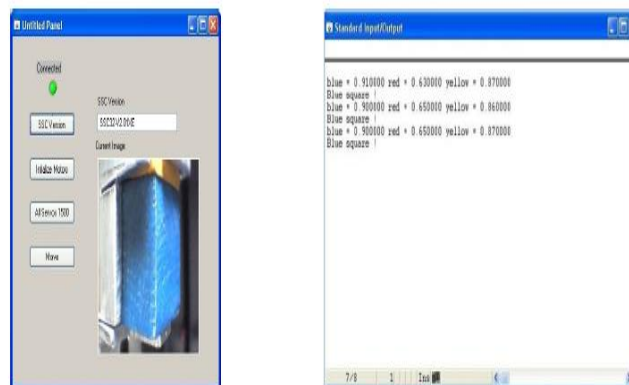


Fig. 7: The application and the decision when recognizing the blue cube

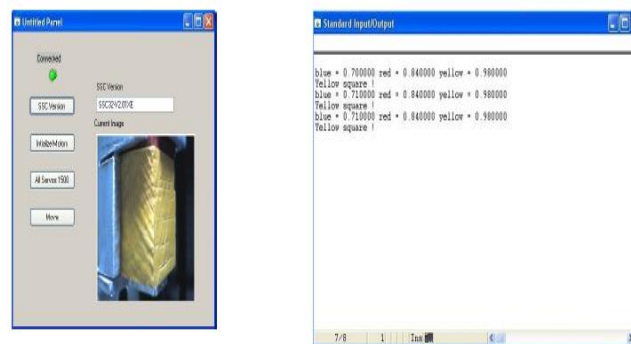


Fig. 8: The application and the decision when recognizing the yellow cube

The whole system is quite simple. We can see that we have an LED to show that we are connected to the webcam, after we have four buttons for the robotic arm according to the described SCPI commands of the SSC-32 servo control board presented in the theoretical part.

We can see that we have a version reading button for the SSC-32 servo control board and an indicator of the board's version. We have the digital port tester button for the SSC-32 board, with this we test the motors, it's similar to a startup self-test.

We put all the motors in central position (All Servos 1500). This will make the robotic arm to have the shape of the Greek capital gamma letter ( $\Gamma$ ); otherwise we can say that we "wake" the robot.

We have the Move button which does all the magic. This button will connect the webcam and with the color recognition application will recognize the color of the cubes and will tell the robotic arm where to put the cube. This will be done, by reading the path of the robotic arm from the specific text file of the color. We can see on the webcam image that the webcam is quite close to the cubes, but using the webcam's manual focus function we can put it a little bit further to have place for the gripper to grab the cube. On the webcam image we can see the system and that in the image is not only the cube, there are other colors too,

but they are ignored by the blob filter. We can also see that the next cube is in the image a little bit, but the blob filter works so fine, that it will ignore that too.

#### IV. CONCLUSION

We could see how can be implemented a color recognition system for a robotic arm. This application is used for a small robotic arm, but with little adjustments the application can be transferred into the industrial environment, by using industrial cameras and industrial robotic arms from KUKA, Mitsubishi, DENSO or other vendors.

#### REFERENCE

- [1] R. Szabo, A. Gontean, I. Lie, "Cheap Live Color Recognition with Webcam," 23rd International Symposium on Information, Communication and Automation Technologies (ICAT), 2011.
- [2] M.H. Liyanage, N. Krouglicof, R. Gosine, "Design and control of a high performance SCARA type robotic arm with rotary hydraulic actuators," Canadian Conference on Electrical and Computer Engineering (CCECE), 2009, pp. 827–832.
- [3] Woosung Yang, Ji-Hun Bae, Yonghwan Oh, Nak Young Chong, Bum-Jae You, Sang-Rok Oh, "CPG based self-adapting multi-DOF robotic arm control," International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 4236–4243.
- [4] Wong Guan Hao, Yap Yee Leck, Lim Chot Hun, "6-DOF PC-Based Robotic Arm (PC-ROBOARM) with efficient trajectory planning and speed control," 4th International Conference On Mechatronics (ICOM), 2011, pp. 7.
- [5] [http://en.wikibooks.org/wiki/Embedded\\_System](http://en.wikibooks.org/wiki/Embedded_System).
- [6] N. Ahuja, U.S. Banerjee, V.A. Darbhe, T.N. Mapara, A.D. Matkar, R.K. Nirmal, S. Balagopalan, "Computer controlled robotic arm," 16th IEEE Symposium on Computer-Based Medical Systems, 2003, pp. 361–366.
- [7] M.H. Liyanage, N. Krouglicof, R. Gosine, "Design and control of a high performance SCARA type robotic arm with rotary hydraulic actuators," Canadian Conference on Electrical and Computer Engineering (CCECE), 2009, pp. 827–832.
- [8] A. Rama Krishna, G. SowmyaBala, A.S.C.S. Sastry, B. BhanuPrakashSarma, GokulSaiAlla, "Design And Implementation Of A Robotic ArmBased On Haptic Technology", International Journal of Engineering Research and Applications. Vol. 2, Issue 3, pp.3098-3103, May-Jun 2012
- [9] Yeow Khang Yung, "Color Sorting System with Robot Arm" Faculty of Electronic and Computer Engineering University Technical Malaysia Melaka, (2011)