Code documentation

The code consists of the following parts:

**Importing the libraries**

#include <Servo.h>

#include <Adafruit\_TCS34725.h>

#include <SoftwareSerial.h>

These lines of code represent the importing of the necessary libraries:

* Servo.h (for controlling the servo motor)
* Adafruit\_TCS34725.h (for the color sensor)
* SoftwareSerial.h (for the bluetooth adapter)

**Initializing the Bluetooth adapter**

SoftwareSerial bluetooth(10, 11);

This line informs the Arduino that the Bluetooth adapter is connected to pins 10 and 11 on the board.

**Initializing the servo and color sensor objects:**

Adafruit\_TCS34725 tcs = Adafruit\_TCS34725(TCS34725\_INTEGRATIONTIME\_50MS, TCS34725\_GAIN\_4X);

Servo myservo;

**Initializing the global variables:**

int countUnknown; // a variable to keep track of how many 'unknown' colors have been detected in a row; it's used for error detection

String receivedData; // a variable to store the data received from the bluetooth module

**The setup function:**

void setup() {

  myservo.attach(9);  // attaching the servo on pin 9 to the servo object

  Serial.begin(9600); // initializing serial communication

  bluetooth.begin(9600); // initializing Bluetooth communication

  if (tcs.begin()) {

    // Serial.println("Color sensor found!");

  } else {

    Serial.println("Error: no color sensor found; check connections"); // Error 1: no color sensor found

    while (1); // stop the code if no color sensor is detected

  }

  tcs.setInterrupt(false); // disabling the color sensor interrupt

}

The code in this function prepares the communication to the servo, the serial monitor and the Bluetooth adapter. It also detects whether the color sensor has been connected properly and otherwise it displays an error message on the Serial Monitor.

**The auxiliary functions:**

int determineColor(uint16\_t r, uint16\_t g, uint16\_t b){

  if (r < 100 && g < 100 && b < 100) return 1; // black

  else if (r > 180 && g > 180 && b > 180) return 0; // white

  else return 2; // unknown

}

The **determineColor** function takes as arguments the three RGB values received from the color sensor and uses them to determine whether the detected color is white, black or unknown.

void back(){

  myservo.write(-30);

}

The **back** function moves the servo motor to the position where it doesn’t block the belt (also reffered to as the initial position).

void forth(){

  myservo.write(30);

}

The **forth** function moves the servo motor to the position where it pushes the disk to the side of the belt.

void pushDisk(){

  forth();

  delay(80);

  back();

  delay(80);

}

The **pushDisk** function uses the **forth** and **back** functions to push the disk to the side of the belt.

void pushBlackDisk(){

  myservo.write(20);

}

The **pushBlackDisk** function moves the servo motor in the position where the (black) disks are blocked by the arm and therefore pushed towards the side of the belt.

**The loop function:**

This function represents the main part of the code and contains the lines that are constantly repeated and assure the functionality of the robot. The delay between each iteration of this function depends on the mode the robot is in and on the color detected by the color sensor.

void loop() {

  // determine the detected color using the RGB values

  uint16\_t r, g, b, c;

  tcs.getRawData(&r, &g, &b, &c);

  float colorTemp = tcs.calculateColorTemperature(r, g, b);

  float lux = tcs.calculateLux(r, g, b);

The first lines are used to process the RGB values sent by the color sensor, which are stored in separate variables.

if (bluetooth.available()) { // check if data is available to be read from Bluetooth

    receivedData = bluetooth.readString(); // read the data received from Bluetooth and store it in a string variable

    // Serial.println(receivedData); // print the received data for debugging

    if (receivedData == "0"){ // the white button on the app has been pushed

      Serial.println("Currently pushing WHITE disks into the box.");

    }

    if (receivedData == "1"){ // the black button on the app has been pushed

      Serial.println("Currently pushing BLACK disks into the box.");

    }

  }

Then, the program checks for input from the app, and stores the received data in the variable **receivedData**. When the black button is pressed on the app, then the string “0” is received, and the white button sends the string “1”. This variable is used to determine the behaviour of the robot in the following lines of code.

if (receivedData == "0"){ // the robot is in the mode where it pushes white disks

    back(); // move the servo back to the initial position

    if (determineColor(r, g, b) == 1) {

      countUnknown = 0; // reset the count for unknown colors to 0

      delay(900); // wait 900 ms before reading next color

    } else if (determineColor(r, g, b) == 0) {

        Serial.println("Detected a WHITE disk; pushing it into the box");

        delay(900); // wait 900 ms before actioning the servo motor/pushing the disk

        pushDisk(); // push the white disk

        delay(900); // wait 900 ms before reading the next color

        countUnknown = 0; // reset the count for unknown colors to 0

    } else { // the color is neither white or black

      countUnknown++; // increase the count of unknown colors

      delay(100); // wait 100 ms before reading the next color

    }

    if (countUnknown == 2){

      Serial.println("Error: forbidden color"); // Error 3: the color detected a color other than white or black

    }

    if (countUnknown == 5){

      Serial.println("Error: color sensor blocked"); // Error 4: the color sensor is blocked

      // while (1); // the robot stops

    }

  }

If the value of **receivedData** is “0”, then the robot has to push the white disks to the side. The code first uses the back function to move the servo motor to the initial position. Then, it uses the **determineColor** function to find what color was detected by the color sensor. If the color is black, then it resets the **countUnknown** variable to 0, and waits 900ms before reading the next color and reiterating the loop. If the disk is white, then it waits 900ms for the disk to reach the servo motor, and then it uses the **pushDisk** function to push the disk to the side. If the color is unknown, then **countUnknown** is increased by 1, and the loop is reiterated with a delay of only 100ms. It does this because there is a possibility the color sensor might have mistakenly seen a white disk as unknown, and it should quickly read the color again to see if it was indeed white, or if it was an unknown color. If the color is unknown at the second iteration too, then there was indeed an unknown color and it prints an error message. If the color is unknown for five iterations of the loop, then the color sensor must be blocked, so it prints an error message.

else if (receivedData == "1"){ // the robot is in the mode where it pushes black disks

    pushBlackDisk(); // the pushing mechanism is set to push black disks

    if (determineColor(r, g, b) == 1) {

      pushBlackDisk(); // the pushing mechanism is set to push black disks

      delay(900); // wait 900 ms before reading the next color

      countUnknown = 0; // reset the count for unknown colors to 0

    } else if (determineColor(r, g, b) == 0) {

        Serial.println("Detected a WHITE disk; letting it pass to the box at the end of the belt");

        delay(750); // wait 750 ms before moving servo back to initial position

        back();

        delay(900); // wait 900 ms before reading the next color

        countUnknown = 0;

    } else { // the color is neither white or black

      delay(100);

      countUnknown++;

    }

    if (countUnknown == 2){

      Serial.println("Error: forbidden color"); // Error 3: the color detected a color other than white or black

    }

    if (countUnknown == 5){

      Serial.println("Error: color sensor blocked"); // Error 4: the color sensor is blocked

      // while (1); // the robot stops

    }

  }

The code for the opposite mode, where the black disks are pushed to the side, is similar to that of the first mode. If **receivedData** is “1”, then the robot will constantly keep the servo motor in a position where it blocks the black disks on the belt and forces them to fall on the side (using the **pushBlackDisk** function), unless the color sensor detects a white disks, in which case in moves the motor to the initial position (using the back function) for 900ms.

  else if (receivedData == ""){ // the robot has been woken up

    receivedData = "0";

    back();

    Serial.println("Thanks for waking me up!");

    Serial.println("Use the app to tell me what color you want me to push in the box.");

    Serial.println("Currently pushing WHITE disks into the box.");

  }

If **receivedData** is an empty string, then it means that the robot has just been turned on, so it displays a wake-up message, and the motor is set to the initial position. The mode is set to the default one (by assigning “0” to **receivedData**), where the white disks are pushed to the side.

else{

    receivedData = "0";

    Serial.println("Error: buttons clicked too fast; the mode is set to default");

  }

If **receivedData** is anything else besides the aforementioned possible strings, that means that the buttons on the app must have been clicked too fast, so it displays an error message and the mode is set to the default one.