A. Starting App

```
import flash.events.*;
import flash.utils.*;
texto1.text="Press Start";
startb.addEventListener(MouseEvent.MOUSE_DOWN, SystemON);


function SystemON(event: MouseEvent) {

    p.visible=true;
    pc.visible=true;

    var file:URLLoader = new URLLoader(new
        URLRequest('https://people.ifm.liu.se/danfi/ODL/data9.csv'));


    file.addEventListener(Event.COMPLETE, csvLoaded);

    function csvLoaded(event:Event):void{
    var lines:Array = String(event.target.data).split('\n');
        //trace(lines.length);

    var i:int;
    i=1;

    var timer:Timer = new Timer(10, 1); // Add a 1 to repeat count
    timer.addEventListener(TimerEvent.TIMER_COMPLETE, onTimerComplete);
    timer.start();



    function onTimerComplete(evt:TimerEvent):void {
        var valor:Array = String(lines[i]).split(";");
    texto1.text="Sequence: " +String(i)+ " of: "+String(lines.length-1);

    timer.reset();
    timer.start();

    timer.delay=int(valor[5]*1000-10);
         if (valor[0]==1){V1.visible=false;}
         if (valor[0]==0){V1.visible=true;}
         if (valor[1]==1){V2.visible=false;}
         if (valor[1]==0){V2.visible=true;}
         if (valor[2]==2){V3.visible=true;}
         if (valor[2]==1){V3.visible=false;}
         if (valor[3]==2){V4.visible=true;}
         if (valor[3]==1){V4.visible=false;}
         if (valor[4]==1){
               p.visible=false;
               pc.visible=false;

                }
         if (valor[4]==0){
               p.visible=true;
               pc.visible=true;
          }

         if (valor[6]==1){LED.visible=false;}
         if (valor[6]==0) {LED.visible=true;}

         i=i+1;

         if (i>=lines.length){timer.stop();}

       stopb.addEventListener(MouseEvent.MOUSE_DOWN, PumpOFF);
       function PumpOFF(event: MouseEvent) {
       timer.stop();

       p.visible=true;
```

```
        pc.visible=true;
        V1.visible=true;
        V2.visible=true;
        V3.visible=true;
        V4.visible=true;
        LED.visible=true;

        texto1.text="Press Start";
}

}

}

}
```

B. Preview Class

The following code can be used as a base for designing a preview class for showing the camera preview in real time:

```
/** A basic Camera preview class */
public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback {
    private SurfaceHolder mHolder;
    private Camera mCamera;

    public CameraPreview(Context context, Camera camera) {
        super(context);
        mCamera = camera;

        // Install a SurfaceHolder.Callback so we get notified when the
        // underlying surface is created and destroyed.
        mHolder = getHolder();
        mHolder.addCallback(this);
        // deprecated setting, but required on Android versions prior to 3.0
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
    }

    public void surfaceCreated(SurfaceHolder holder) {
        // The Surface has been created, now tell the camera where to draw the preview.
        try {
            mCamera.setPreviewDisplay(holder);
            mCamera.startPreview();
        } catch (IOException e) {
            Log.d(TAG, "Error setting camera preview: " + e.getMessage());
        }
    }

    public void surfaceDestroyed(SurfaceHolder holder) {
        // empty. Take care of releasing the Camera preview in your activity.
    }

    public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {
        // If your preview can change or rotate, take care of those events here.
        // Make sure to stop the preview before resizing or reformatting it.

        if (mHolder.getSurface() == null){
          // preview surface does not exist
          return;
        }

        // stop preview before making changes
        try {
            mCamera.stopPreview();
        } catch (Exception e){
          // ignore: tried to stop a non-existent preview
        }
```

```
        // set preview size and make any resize, rotate or
        // reformatting changes here

        // start preview with new settings
        try {
            mCamera.setPreviewDisplay(mHolder);
            mCamera.startPreview();

        } catch (Exception e){
            Log.d(TAG, "Error starting camera preview: " + e.getMessage());
        }
    }

}
```

C. Frame Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
   <LinearLayout
           android:orientation="horizontal"
           android:layout_width="fill_parent"
           android:layout_height="fill_parent">
           <TextView
                   android:text="red"
                   android:gravity="center_horizontal"
                   android:background="#aa0000"
                   android:layout_width="wrap_content"
                   android:layout_height="fill_parent"
                   android:layout_weight="1"/>
           <TextView
                   android:text="green"
                   android:gravity="center_horizontal"
                   android:background="#00aa00"
                   android:layout_width="wrap_content"
                   android:layout_height="fill_parent"
                   android:layout_weight="2"/>
           <TextView
                   android:text="blue"
                   android:gravity="center_horizontal"
                   android:background="#0000aa"
                   android:layout_width="wrap_content"
                   android:layout_height="fill_parent"
                   android:layout_weight="3"/>
           <TextView
                   android:text="yellow"
                   android:gravity="center_horizontal"
                   android:background="#aaaa00"
                   android:layout_width="wrap_content"
                   android:layout_height="fill_parent"
                   android:layout_weight="4"/>
   </LinearLayout>

   <LinearLayout
           android:orientation="vertical"
           android:layout_width="fill_parent"
           android:layout_height="fill_parent">
           <TextView
                   android:text="Primera fila"
                   android:layout_width="fill_parent"
                   android:layout_height="wrap_content"/>
           <TextView
                   android:text="Segunda fila"
                   android:layout_width="fill_parent"
                   android:layout_height="wrap_content"/>
           <TextView
                   android:text="Tercera fila"
                   android:layout_width="fill_parent"
                   android:layout_height="wrap_content"/>
   </LinearLayout>

</LinearLayout>
```

D. Relative Layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent">
   <TextView
           android:id="@+id/etiqueta"
           android:layout_width="fill_parent"
           android:layout_height="wrap_content"
           android:text="Escribe"/>
   <EditText
           android:id="@+id/entrada"
           android:layout_width="fill_parent"
           android:layout_height="wrap_content"
           android:background="@android:drawable/editbox_background"
           android:layout_below="@id/etiqueta"/>
   <Button
           android:id="@+id/ok"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_below="@id/entrada"
           android:layout_alignParentRight="true"
           android:layout_marginLeft="10dip"
           android:text="OK" />
   <Button
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:layout_toLeftOf="@id/ok"
           android:layout_alignTop="@id/ok"
           android:text="Cancelar" />
</RelativeLayout>
```

E. Starting App with Camera Preview

In the following code are implemented all the changes made to the original application for showing a camera preview on the application, if the device has more than one camera, the frontal one was selected:

```
import flash.events.*;
import flash.utils.*;
import flash.media.Camera;
import flash.media.Video;

var camera: Camera = Camera.getCamera("1");
var camera2: Camera = Camera.getCamera("0");

texto1.text = "Press Start";

//Camera on function
if (camera != null) {

	video.attachCamera(camera);
} else {
	if (camera2 != null) {
		video.attachCamera(camera2)
	} else {
		texto2.text = "You need a camera.";

	}
}

startb.addEventListener(MouseEvent.MOUSE_DOWN, SystemON);

function SystemON(event: MouseEvent) {

	p.visible = true;
	pc.visible = true;
	LED.visible = true;
	aux.visible = true;

	var file: URLLoader = new URLLoader(new
URLRequest('https://people.ifm.liu.se/danfi/ODL/data9.csv'));


	file.addEventListener(Event.COMPLETE, csvLoaded);

	function csvLoaded(event: Event): void {
		var lines: Array = String(event.target.data).split('\n');
		//trace(lines.length);

		var i: int;
		i = 1;


		var timer: Timer = new Timer(10, 1); // Add a 1 to repeat count
		timer.addEventListener(TimerEvent.TIMER_COMPLETE, onTimerComplete);
		timer.start();
```

```
function onTimerComplete(evt: TimerEvent): void {
        var valor: Array = String(lines[i]).split(";");
        texto1.text = "Sequence: " + String(i) + " of: " +
        String(lines.length - 1);

        timer.reset();
        timer.start();

        timer.delay = int(valor[5] * 1000 - 10);
        if (valor[0] == 1) {
                V1.visible = false;
        }
        if (valor[0] == 0) {
                V1.visible = true;
        }
        if (valor[1] == 1) {
                V2.visible = false;
        }
        if (valor[1] == 0) {
                V2.visible = true;
        }
        if (valor[2] == 2) {
                V3.visible = true;
        }
        if (valor[2] == 1) {
                V3.visible = false;
        }
        if (valor[3] == 2) {
                V4.visible = true;
        }
        if (valor[3] == 1) {
                V4.visible = false;
        }
        if (valor[4] == 1) {
                p.visible = false;
                pc.visible = false;

        }
        if (valor[4] == 0) {
                p.visible = true;
                pc.visible = true;

        }
        /*
        if (valor[6] == 1) {
                LED.visible = false;
        }
        if (valor[6] == 0) {
                LED.visible = true;
        }
        */

        //trace(valor[5]);
        i = i + 1;

        if (i >= lines.length) {
                timer.stop();
        }

        stopb.addEventListener(MouseEvent.MOUSE_DOWN, PumpOFF);
```

```
                    // When you presh stop this function is called, if sqares
                     // visible = not working

                    function PumpOFF(event: MouseEvent) {
                            timer.stop();
                            p.visible = true;
                            pc.visible = true;
                            V1.visible = true;
                            V2.visible = true;
                            V3.visible = true;
                            V4.visible = true;
                            LED.visible = true;
                            texto1.text = "Press Start";
                    }

            }

      }
}
```

F. Processing photos using Matlab

```
clear all;
close all;
clc;

%Mo = imread (strcat('IMG_00',num2str(54),'.jpg'));
%imshow(Mo(:,:,1));

x1 = 210;
x2 = 160;
y1 = 245;
y2 = 245;
thrlsd = 30;


for i = 1:10
    M1 = imread (strcat('IMG_00',num2str(i+54),'.jpg'));
    imshow(M1(:,:,1));
    rectangle('Position',[x2,y1,x1-x2,5]);
    M1(y1,x1,1)
    if (M1(y1,x1,1) < thrlsd)
        text(10,50,'Enter zone')
    end
    if (M1(y1,x2,1) < thrlsd)
        text(10,60,'Exit zone')
    end
    pause
end
```

G. Optically Commanded Aplication

```
package com.example.ruben.framessegundo;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Rect;
import android.graphics.YuvImage;
import android.hardware.Camera;
import android.os.Bundle;
import android.view.View;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.TextView;

import java.io.ByteArrayOutputStream;


public class MainActivity extends Activity implements Camera.PreviewCallback {

    private Camera mCamera;
    private CameraPreview mPreview;

    private TextView textView, textView2, textView3, textView4, textView5,
     textView9, textView10, textView11;
    private Bitmap bitmap;

    private int xu,yu;
    private int xd,yd;

    private int xx1=1,yy1=1,xx2=1;
    private double sx=0;

    private boolean topfound = false,botfound=false;
    private boolean calibrated=false,stop=false;

    private ImageView aux, led, p, pc, v1, v2, v3, v4;
    private int val1,val2,val3,val4,veces=1;
    private boolean volumemeasured=false,apagado=false;

    private static final double incx = 18.0;
    private static final double x1 = 8.5;
    private static final double y1 = 12.5;
    private static final double x2 = 13.0;
    private static final int thrlsd = 50;


    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Relating visual items with code text variables

        textView = (TextView) findViewById(R.id.initialcoor);
        textView2 = (TextView) findViewById(R.id.finalcoord);
        textView3 = (TextView) findViewById(R.id.coordArriba);

        textView4 = (TextView) findViewById(R.id.colorleft);
        textView5 = (TextView) findViewById(R.id.colorright);

        textView9 = (TextView) findViewById(R.id.liqdetectright);
        textView10 = (TextView) findViewById(R.id.liqdetleft);
        textView11 = (TextView) findViewById(R.id.CoordDown);

        //Relating visual item with image variables

        led = (ImageView)findViewById(R.id.led);
        aux = (ImageView)findViewById(R.id.aux);
```

```
        p = (ImageView)findViewById(R.id.p);
        pc = (ImageView)findViewById(R.id.pc);
        v1 = (ImageView)findViewById(R.id.v1);
        v2 = (ImageView)findViewById(R.id.v2);
        v3 = (ImageView)findViewById(R.id.v3);
        v4 = (ImageView)findViewById(R.id.v4);

        // Create a Camera Instance
        mCamera = getCameraInstance();

        // Create our Preview view and set it as the content of our activity.
        mPreview = new CameraPreview(this, mCamera);
        FrameLayout preview = (FrameLayout) findViewById(R.id.camera_preview);
        preview.addView(mPreview);

        //Keeping the screen on all time the app is running

       getWindow().addFlags(
             android.view.WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    }

    public void onPause() {
        super.onPause();

        if (mCamera != null) {
            mCamera.setPreviewCallback(null);
            mPreview.getHolder().removeCallback(mPreview);
            mCamera.release();
        }
    }

    /** A safe way to get an instance of the Camera object. */
    public static Camera getCameraInstance(){
        Camera c = null;
        try {
            c = Camera.open(); // attempt to get a Camera instance
        }
        catch (Exception e){
            // Camera is not available (in use or does not exist)
        }
        return c; // returns null if camera is unavailable
    }

    //onPreviewFrame is a function that executes ALL time that the phone receives
    //a preview frame from the camera preview. We use it for getting a still image
    // (bitmap) from the camera preview
    //

    public void onPreviewFrame(byte[] data, Camera camera) {

        //Obtaining camera parameters for getting the exact width an height
        Camera.Parameters parameters = camera.getParameters();
        int width = parameters.getPreviewSize().width;
        int height = parameters.getPreviewSize().height;

        //YuvImage contains YUV data and provides a method that compresses a
          region of the YUV data to a Jpeg
        YuvImage yuv = new YuvImage(data, parameters.getPreviewFormat(), width,
                                    height, null);

        //Compress a rectangle region in the YuvImage to a jpeg.
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        yuv.compressToJpeg(new Rect(0, 0, width, height), 50, out);
```

```
            //Once is converted to Jpeg, we use toByteArray (from bitmap library) for
            //creating a byte array and one step after decoding it into a bitmap, THIS
            //bitmap is the one that will be used for the image processing
            byte[] bytes = out.toByteArray();
            bitmap = BitmapFactory.decodeByteArray(bytes, 0, bytes.length);

            //Running the frame processing on a UI thread, this Ui thread is used
             typically for processing
            //bitmaps inside it

            MainActivity.this.runOnUiThread(new Runnable() {

                @Override
                public void run() {
                    if (!stop && calibrated==true) processFrames();
                }
            });
        }

        // Starting the calibration algorithm taking the current bitmap
        //This code will execute ONLY WHEN WE PUSH THE BUTTON CALIBRATE
        public void calibrate (View v){
            FindCoords(bitmap);

            stop = false;
            calibrated = true;
        }

        //Different steps: Finding left and right black corners and calculating the
          coords finally
        public void FindCoords (Bitmap bit){
            left(bit);
            right(bit);
            calibration();
        }

        //Going through the left side of the bitmap looking for a black pixel
        //We go from bottom to top, and from left to right for going through the
          bitmap

        public void left(Bitmap bit){
            textView.setText("");
            topfound= false;

            int i=0;  int j=0;

            for (i=1; i < bitmap.getWidth()/4; i++)
                for(j=bitmap.getHeight()-1; j > 2*bitmap.getHeight()/3; j--)
                    RGBleft(i, j, bit);


            textView.setText("Exito! Xu= " + xu + " Yu= " + yu);
        }

        //Check if a concrete pixel is black, if it is, we get the coords and we stop
          looking
        public void RGBleft (int l, int m, Bitmap bit){
            int pixel = bit.getPixel(l, m);
            int r = Color.red(pixel);
            int g = Color.green(pixel);
            int b = Color.blue(pixel);

            if (r < 55 ) if (g<55) if (b<55)
                if (!topfound) {
                    xu=l; yu=m;
                    topfound=true;
                }
        }



    //Same process as left, but for the right side

        public void right(Bitmap bit){
            textView2.setText("");
```

```
    botfound= false;

    int i=0;  int j=0;

    for (i=bitmap.getWidth()/4; i < bitmap.getWidth(); i++)
        for(j=bitmap.getHeight()-1; j > 2*bitmap.getHeight()/3; j--)
            RGBright(i,j, bit);



    textView2.setText("Exito! Xd= " + xd + " Yd= " + yd);

}


public void RGBright (int l, int m, Bitmap bit){
    int pixel = bit.getPixel(l, m);
    int r = Color.red(pixel);
    int g = Color.green(pixel);
    int b = Color.blue(pixel);

    if (r < 55 ) if (g<55) if (b<55)
        if (!botfound) {
            xd=l; yd=m;
            botfound=true;
        }
}

//These calculations are used for calibrate the app and finding the coords of
// the pipe we are going to analyze
public void calibration(){

    sx = Math.abs(xu-xd)/incx; // pix/mm
    yy1 = (int) (sx * y1);
    xx1 = (int) (sx * x1);
    xx2 = (int) (sx * x2);

    //Adjustments to show the coords in the PREVIEW WHICH IS SHOWN
    int xmos =bitmap.getHeight()-(yu-yy1);
    int y1mos=xu+xx1;
    int y2mos=xu+xx2;

    textView3.setText("X1= "+xmos+" Y1= "+y1mos);
    textView11.setText("X2= "+xmos+" Y2= "+y2mos);
}


//This function will execute all time that we get a bitmap is taken from the
  camera preview
//IF THE CAMERA IS ALREADY CALIBRATED, it takes the coords calculated in the
  calibration
//and look for the blue color there, if r2 < thrlsd, then there's liquid in
  the lower mark,
// if r1 > thrlsd the liquid reached the upper mark
public void processFrames (){
    int ybusc= yu-yy1;
    int x1busc= xu+xx1;
    int x2busc= xu+xx2;

    textView10.setText("Not in zone");
    textView9.setText("Not in zone");

    int pixel = bitmap.getPixel(x1busc, ybusc);
    int r1 = paintRGB(pixel,textView4);
```

```
        pixel = bitmap.getPixel(x2busc, ybusc);
        int r2 = paintRGB(pixel, textView5);

        if (!volumemeasured){
            s1();
            mostrar();
        }
        if (r2 < thrlsd) {
            textView9.setText("Enter Zone");
            liquidInside();
        }
        if (r1 < thrlsd) {
            textView10.setText("Enter Zone");
            liquidOutside();
            volumemeasured=true;
        }

    }

    //Actions done when the first limit is reached and we need to control the pump
    public void liquidInside(){
        //Apagar
        if (!apagado) parar();

        //Encender
        if (apagado) {
            s1();
            mostrar();
        }

        //Sumar 1 vez
        veces++;
        if (veces == 20) {
            apagado=!apagado;
            veces=1;
        }
    }

    //Actions done when second limit is reached and now we have the volume needed
    public void liquidOutside(){
        parar();
    }

    public void s1 (){
        val1= 1 ; val2= 0 ; val3= 1; val4= 1;
    }

    //Method used for "Painting" the color that we are searching in the coords
    //Calculated during the calibration in a textView, also returns the red
    //value of the pixel entered
    public int paintRGB (int pixel, TextView textView){
        int r = Color.red(pixel);
        int g = Color.green(pixel);
        int b = Color.blue(pixel);

        textView.setText("");
        textView.setBackgroundColor(Color.rgb(r, g, b));

        return r;
    }

    //Stopping the frame processing and reinitializing everything
    public void stop (View v){
        stop = true;
        volumemeasured=false;

        //Initialize the layouts
        textView.setText("Xo and Yo");
        textView2.setText("Xf and Yf");

        textView3.setText("X1 and Y1");
        textView11.setText("X2 and Y2");

        textView10.setText("Liquid detection");
        textView9.setText("Liquid detection");
```

```
            textView4.setText("Down Color");
            textView4.setBackgroundColor(Color.BLACK);
            textView5.setText("Up Color");
            textView5.setBackgroundColor(Color.BLACK);

            //Initialize all controlling frames to white
            parar();

        }

        public void mostrar(){
            p.setImageResource(R.drawable.cuadradonegro);
            pc.setImageResource(R.drawable.cuadradonegro);

            if (val1 == 1) v1.setImageResource(R.drawable.cuadradonegro);
            if (val1 == 0) v1.setImageResource(R.drawable.cuadradoblanco);

            if (val2 == 1) v2.setImageResource(R.drawable.cuadradonegro);
            if (val2 == 0) v2.setImageResource(R.drawable.cuadradoblanco);

            if (val3 == 1) v3.setImageResource(R.drawable.cuadradonegro);
            if (val3 == 2) v3.setImageResource(R.drawable.cuadradoblanco);

            if (val4 == 1) v4.setImageResource(R.drawable.cuadradonegro);
            if (val4 == 2) v4.setImageResource(R.drawable.cuadradoblanco);
        }

        public void  parar (){
            led.setImageResource(R.drawable.cuadradoblanco);
            aux.setImageResource(R.drawable.cuadradoblanco);
            p.setImageResource(R.drawable.cuadradoblanco);
            pc.setImageResource(R.drawable.cuadradoblanco);
            v1.setImageResource(R.drawable.cuadradoblanco);
            v2.setImageResource(R.drawable.cuadradoblanco);
            v3.setImageResource(R.drawable.cuadradoblanco);
            v4.setImageResource(R.drawable.cuadradoblanco);
        }

    }
```