

```
int main (int argc, char *argv[]) {
    :      中略      :

    if((Image = cvLoadImage(argv[1], CV_LOAD_IMAGE_COLOR)) == 0) {
        printf("Error: Failed to load %s¥n", argv[1]);
        return -1;
    }
    img_gray=cvCreateImage(cvGetSize(Image),IPL_DEPTH_8U,1);
    cvCvtColor(Image,img_gray,CV_BGR2GRAY);
    cvThreshold(img_gray,img_gray,100,255,CV_THRESH_BINARY);
    cvFindContours(img_gray,Storage,&Contour,sizeof(CvContour),
        CV_RETR_TREE,CV_CHAIN_APPROX_NONE, offset);

    if (Contour != NULL){
        DrawNextContour(Image, Contour, 1);
    } else {
        printf(" contour is null¥n");
    }
    :      中略      :
}

void DrawNextContour( IplImage *img, CvSeq *Contour, int Level ) {
    :      中略      :

    maxx = 0; maxy = 0; minx = img->width; miny = img->height;
    GetContourFeature(Contour, &area, &maxx, &maxy, &minx, &miny);
    sprintf(text, "%10.0f", area);

    if(area > 25000) {
        cvDrawContours( img, Contour, ContoursColor, ContoursColor, 0, 2, line_type, offset);
        cvPutText(img, text, cvPoint (minx+5, miny+20), &font, ContoursColor);
        cvSetImageROI(img, cvRect(minx,miny,maxx-minx,maxy-miny));
        sprintf( savefile, "AREA%d.jpg", AreaN);
        printf( "%s¥n", savefile );
        cvSaveImage(savefile, img, 0);
        cvResetImageROI(img);
        cvLine (img, cvPoint (maxx, maxy), cvPoint (maxx, miny), ContoursColor, 3, 8, 0);
        cvLine (img, cvPoint (minx, maxy), cvPoint (minx, miny), ContoursColor, 3, 8, 0);
        cvLine (img, cvPoint (maxx, maxy), cvPoint (minx, maxy), ContoursColor, 3, 8, 0);
        cvLine (img, cvPoint (maxx, miny), cvPoint (minx, miny), ContoursColor, 3, 8, 0);
        AreaN++;
    }
    if (Contour->h_next != NULL)
        DrawNextContour(img, Contour->h_next, Level);
    if (Contour->v_next != NULL)
        DrawChildContour(img, Contour->v_next, Level + 1);
}
```

```
int main (int argc, char *argv[]) {
    :      中略      :
    if((image = cvLoadImage(argv[1], CV_LOAD_IMAGE_COLOR)) == 0) {
        printf("Error: Failed to load %s¥n", argv[1]);
        return -1;
    }

    IplImage *scale_image=cvCreateImage( cvSize(600, 480), IPL_DEPTH_8U, 3);
    cvResize(image, scale_image, CV_INTER_LINEAR);
    cvSaveImage("RESIZE.jpg", scale_image, 0);

    cvSetImageROI(scale_image, cvRect(165, 48, 105, 160));
    cvSaveImage("T1.jpg", scale_image, 0);
    cvResetImageROI(scale_image);

    cvSetImageROI(scale_image, cvRect(245, 48, 105, 160));
    cvSaveImage("T2.jpg", scale_image, 0);
    cvResetImageROI(scale_image);

    cvSetImageROI(scale_image, cvRect(355, 48, 105, 160));
    cvSaveImage("T3.jpg", scale_image, 0);
    cvResetImageROI(scale_image);

    cvSetImageROI(scale_image, cvRect(275, 215, 130, 180));
    cvSaveImage("H1.jpg", scale_image, 0);
    cvResetImageROI(scale_image);

    cvSetImageROI(scale_image, cvRect(380, 215, 130, 180));
    cvSaveImage("H2.jpg", scale_image, 0);
    cvResetImageROI(scale_image);

    :      中略      :
    return 0;
}
```

輪郭検出して液晶部分を切り出すedgcapture.c(一部抜粋)

リサイズして各数字を切り出すresizecut.c(一部抜粋)