

toto.java

```
1 package com.raspberry;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 import org.apache.http.HttpResponse;
11 import org.apache.http.NameValuePair;
12 import org.apache.http.client.HttpClient;
13 import org.apache.http.client.entity.UrlEncodedFormEntity;
14 import org.apache.http.client.methods.HttpGet;
15 import org.apache.http.client.methods.HttpPost;
16 import org.apache.http.impl.client.DefaultHttpClient;
17 import org.apache.http.message.BasicNameValuePair;
18
19
20
21 import android.net.ConnectivityManager;
22 import android.net.NetworkInfo;
23 import android.os.AsyncTask;
24 import android.os.Bundle;
25 import android.util.Log;
26 import android.widget.EditText;
27 import android.widget.TextView;
28 import android.widget.Toast;
29 import android.app.Activity;
30 import android.content.Intent;
31
32 import java.io.BufferedReader;
33 import java.io.IOException;
34 import java.io.InputStream;
35 import java.io.InputStreamReader;
36
37 import org.apache.http.HttpEntity;
38 import org.apache.http.HttpResponse;
39 import org.apache.http.StatusLine;
40 import org.apache.http.client.ClientProtocolException;
41 import org.apache.http.client.HttpClient;
42 import org.apache.http.client.methods.HttpGet;
43 import org.apache.http.impl.client.DefaultHttpClient;
44
45 //import android.os.AsyncTask;
46
47
48 import android.app.Activity;
49 import android.net.ConnectivityManager;
50 import android.net.NetworkInfo;
51 import android.os.Bundle;
52 import android.util.Log;
53 import android.view.View;
54 import android.widget.Button;
55 import android.widget.EditText;
56 import android.view.Menu;
57 import android.view.MenuInflater;
58 import android.view.MenuItem;
59 import android.widget.Switch;
60
61
62 import android.widget.CompoundButton;
```

toto.java

```
63 import android.widget.CompoundButton.OnCheckedChangeListener;
64 import android.widget.TextView;
65 import android.widget.Toast;
66
67
68 import android.app.Activity;
69 import android.os.AsyncTask;
70 import android.os.Bundle;
71 import android.widget.Button;
72 import android.widget.Toast;
73 import android.view.View;
74 import android.view.View.OnClickListener;
75 import android.widget.ImageView;
76 import android.app.ProgressDialog;
77 import android.graphics.Bitmap;
78 import android.graphics.BitmapFactory;
79 import android.graphics.Color;
80
81 import java.io.InputStream;
82 import java.net.URL;
83 import java.io.*;
84 import java.net.*;
85 import java.io.BufferedReader;
86 import java.io.InputStreamReader;
87 import java.io.PrintWriter;
88 import java.net.InetAddress;
89 import java.net.Socket;
90 import java.io.ByteArrayOutputStream;
91 import java.io.IOException;
92 import java.io.InputStream;
93 import java.net.Socket;
94 import java.net.UnknownHostException;
95
96 import android.app.ActionBar;
97 import android.util.Log;
98
99 import java.net.InetAddress;
100
101 public class toto extends Activity{
102
103     private boolean position = false;
104     private String message = "";
105
106     Button alarm;
107     Button button_capture_camera;
108     TextView champ_connexion;
109     TextView champ_reception;
110     ImageView capture_image;
111     TextView textResponse;
112
113     Bitmap bitmap;
114     ProgressDialog pDialog;
115     String nom_image = "";
116
117
118     public void onCreate(Bundle savedInstanceState)
119     {
120
121         super.onCreate(savedInstanceState);
122         setContentView(R.layout.toto);
123
124         ActionBar actionBar = getActionBar();
```

```

125     actionBar.show();
126
127     alarm = (Button) findViewById(R.id.button_commande);
128
129     // chargement de la configuration
130     alarm.setText("Chargement...");
131
132     champ_connexion = (TextView) findViewById(R.id.champ_connexion);
133     champ_reception = (TextView) findViewById(R.id.champ_reception);
134     button_capture_camera = (Button) findViewById(R.id.button_capture_camera);
135     capture_image = (ImageView) findViewById(R.id.img);
136     textResponse = (TextView) findViewById(R.id.response);
137
138     if(isConnected())
139     {
140         champ_connexion.setBackgroundColor(0xFF00CC00);
141         champ_connexion.setText("Connexion réseau");
142         //new
143         HttpAsyncTask().execute("http://analog-design.net/WORKAREA/lire_etat_alarme.php");
144         HttpAsyncTask().execute("http://damien.bruvier.free.fr/Rpi/lire_etat_alarme.php");
145     }
146     else
147     {
148         champ_connexion.setText("Problème avec le réseau");
149         // on cache le bouton
150         alarm.setVisibility(View.GONE);
151         champ_connexion.setBackgroundColor(Color.RED);
152         //alarm.setVisibility(View.VISIBLE);
153     }
154
155     //liaison_raspberry();
156     //test();
157
158     alarm.setOnClickListener(new OnClickListener()
159     {
160
161         @Override
162         public void onClick(View arg0) {
163             if(!position)
164             {
165                 alarm.setText("ALARME ON");
166                 position = true;
167                 message = "Alarme active";
168                 new MyAsyncTask().execute("on");
169             }
170             else
171             {
172                 alarm.setText("ALARME OFF");
173                 position = false;
174                 message = "Alarme désactivée";
175                 new MyAsyncTask().execute("off");
176             }
177         }
178
179         Toast.makeText(getApplicationContext(), message,
180             Toast.LENGTH_LONG).show();
181     }
182 });
183

```

```

184
185     button_capture_camera.setOnClickListener(new View.OnClickListener()
186     {
187         public void onClick(View v)
188         {
189             capture_cameraClick();
190         }
191     });
192
193
194     //new
195     AsyncTask().execute("http://analog-design.net/WORKAREA/lire_etat_alarm.php");
196     //String adresse = "http://damien.bruvier.free.fr/Rpi/mise_a_jour_alarm.php";
197     //new AsyncTask().execute(adresse);
198 }
199
200
201 @Override
202 public boolean onCreateOptionsMenu(Menu menu)
203 {
204     MenuInflater inflater = getMenuInflater();
205     inflater.inflate(R.menu.items, menu);
206     return super.onCreateOptionsMenu(menu);
207 }
208
209
210 @Override
211 public boolean onOptionsItemSelected(MenuItem item)
212 {
213
214     super.onOptionsItemSelected(item);
215
216     switch(item.getItemId()){
217
218
219         case R.id.GPIO:
220             Toast.makeText(getBaseContext(), "Selection GPIO",
221 Toast.LENGTH_SHORT).show();
222             break;
223
224         case R.id.capteur:
225             Toast.makeText(getBaseContext(), "Selection capteur",
226 Toast.LENGTH_SHORT).show();
227             break;
228
229         case R.id.camera:
230             Toast.makeText(getBaseContext(), "Selection Camera",
231 Toast.LENGTH_SHORT).show();
232             Intent productIntent = new Intent(this, camera.class);
233             startActivity(productIntent);
234             // return true;
235             break;
236
237         //case R.id.video:
238         // Toast.makeText(getBaseContext(), "You selected Video",
239 Toast.LENGTH_SHORT).show();
240         // break;
241
242         //case R.id.email:
243         // Toast.makeText(getBaseContext(), "You selected EMail",
244 Toast.LENGTH_SHORT).show();

```

toto.java

```
240         // break;
241     }
242 }
243
244     return true;
245 }
246 }
247
248
249     public boolean isConnected()
250     {
251         ConnectivityManager connMgr = (ConnectivityManager)
252         getSystemService(this.CONNECTIVITY_SERVICE);
253         NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
254         if (networkInfo != null && networkInfo.isConnected())
255             return true;
256         else
257             return false;
258     }
259
260     public void liaison_raspberry()
261     {
262         InetAddress in;
263         in = null;
264         String reponse = "";
265
266         try {
267
268             in = InetAddress.getByName("www.google.fr");
269         }
270
271         catch (UnknownHostException e)
272         {
273             e.printStackTrace();
274         }
275
276         //http://hmkcode.com/android-internet-connection-using-http-get-httpclient/
277         try {
278             boolean result = false;
279
280             result = in.isReachable(100);
281
282             /*if(reachable == true)
283             {
284                 reponse = "Reponse OK";
285                 Toast.makeText(getApplicationContext(), reponse,
286                 Toast.LENGTH_SHORT).show();
287             }
288             else
289             {
290                 reponse = "Pas de reponse";
291                 Toast.makeText(getApplicationContext(), reponse,
292                 Toast.LENGTH_SHORT).show();
293             }*/
294         }
295
296         catch (IOException e)
297         {
298             // TODO Auto-generated catch block
299             String nada = "";
```

toto.java

```
299
300     }
301
302     public void test()
303     {
304
305         InputStream inputStream = null;
306         String result = "";
307
308         try {
309
310             HttpClient httpClient = new DefaultHttpClient();
311             HttpResponse httpResponse = httpClient.execute(new
HttpGet("www.google.fr"));
312             // reponse
313             inputStream = httpResponse.getEntity().getContent();
314             if(inputStream != null)
315             {
316                 result = "Communication ";
317                 Toast.makeText(getApplicationContext(), result,
Toast.LENGTH_LONG).show();
318             }
319             else
320             {
321                 result = "Did not work!";
322                 Toast.makeText(getApplicationContext(), result,
Toast.LENGTH_LONG).show();
323             }
324
325         }
326
327         catch (Exception e)
328         {
329             String erreur = "plouf";
330         }
331
332
333
334     }
335
336
337     public static String GET(String url){
338         InputStream inputStream = null;
339         String result = "";
340         try {
341
342             // creation du HttpClient
343             HttpClient httpClient = new DefaultHttpClient();
344             // faire une requete de type GET sur l'url suivante
345             HttpResponse httpResponse = httpClient.execute(new HttpGet(url));
346             // reception de la reponse
347             inputStream = httpResponse.getEntity().getContent();
348             // convertir inputStream en string
349             if(inputStream != null)
350                 result = convertInputStreamToString(inputStream);
351             else
352                 result = "Probleme mon gars!";
353
354         } catch (Exception e) {
355             Log.d("InputStream", e.getLocalizedMessage());
356         }
357     }
```

```

358
359     return result;
360 }
361
362 private static String convertInputStreamToString(InputStream inputStream) throws
IOException
363 {
364
365     BufferedReader bufferedReader = new BufferedReader( new
InputStreamReader(inputStream));
366     String line = "";
367     String result = "";
368
369     while((line = bufferedReader.readLine()) != null)
370         result += line;
371     inputStream.close();
372     return result;
373 }
374
375
376 private class HttpAsyncTask extends AsyncTask<String, Void, String>
377 {
378     @Override
379     protected String doInBackground(String... urls)
380     {
381
382         return GET(urls[0]);
383     }
384     // onPostExecute displays the results of the AsyncTask.
385     @Override
386     protected void onPostExecute(String result)
387     {
388         // Toast.makeText(getBaseContext(), "Received!", Toast.LENGTH_LONG).show();
389         //String reponse = "Raspberry à répondu: " + result.toUpperCase();
390         //reponse = "Raspberry répond aux requettes HTTP ";
391         String reponse = "";
392
393         //champ_resultat.setText(result);
394         //etat = champ_resultat.toString();
395
396         // Toast.makeText(getApplicationContext(),
397         //             t, Toast.LENGTH_LONG).show();
398
399         // champ_reception.setText(result);
400         // String etat = "";
401
402         champ_reception.setText(reponse);
403
404         //champ_reception.setText(result);
405         //etat = champ_reception.toString();
406
407
408
409         if(result.equals("on"))
410         {
411             alarm.setText("ALARME ON");
412             position = true;
413             Toast.makeText( getApplicationContext(), "Statut actuel",
Toast.LENGTH_LONG).show();
414         }
415         else
416         {

```

toto.java

```
417         alarm.setText("ALARME OFF");
418         position = false;
419         Toast.makeText( getApplicationContext(), "Statut actuel",
Toast.LENGTH_LONG).show();
420     }
421 }
422 }
423 }
424
425
426 private class MyAsyncTask extends AsyncTask<String, Integer, Double>{
427
428     @Override
429     protected Double doInBackground(String... params)
430     {
431         postData(params[0]);
432         return null;
433     }
434
435     protected void onPostExecute(Double result)
436     {
437         //Toast.makeText(getApplicationContext(), "command sent",
Toast.LENGTH_LONG).show();
438         String toto = "";
439     }
440
441     protected void onProgressUpdate(Integer... progress)
442     {
443         String bidule = "";
444     }
445
446     public void postData(String valueIWantToSend)
447     {
448         // nouveau client Http
449         HttpClient httpclient = new DefaultHttpClient();
450         //HttpPost httppost = new
HttpPost("http://www.analog-design.net/WORKAREA/mise_a_jour_alarme.php");
451         HttpPost httppost = new
HttpPost("http://damien.bruvier.free.fr/Rpi/mise_a_jour_alarme.php");
452
453         try {
454             // ajouter les datas
455             List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
456             nameValuePairs.add(new BasicNameValuePair("HttpData",
valueIWantToSend));
457             httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
458
459             // Execute une requete HTTP Post
460             HttpResponse response = httpclient.execute(httppost);
461
462         }
463
464         catch (ClientProtocolException e)
465         {
466
467         }
468
469         catch (IOException e)
470         {
471
472         }
473     }
```

```

474
475     }
476
477
478     public void capture_cameraClick()
479     {
480         int port = 8081;
481         String address = "82:234:53:234";
482         MyClientTask myClientTask = new MyClientTask(address,port);
483         myClientTask.execute();
484         String cmd = "http://www.analog-design.net/WORKAREA/" + nom_image;
485         Toast.makeText(toto.this, cmd, Toast.LENGTH_SHORT).show();
486         // new LoadImage().execute("http://www.analog-design.net/WORKAREA/sea.jpg");
487         new LoadImage().execute(cmd);
488     }
489
490
491     public class MyClientTask extends AsyncTask<Void, Void, Void>
492     {
493
494         String dstAddress;
495         int dstPort;
496         String response = "";
497
498         MyClientTask(String addr, int port){
499             dstAddress = addr;
500             dstPort = port;
501         }
502
503         @Override
504         protected Void doInBackground(Void... arg0)
505         {
506
507             Socket socket = null;
508
509             try {
510                 socket = new Socket("82.234.53.234", 8081);
511
512                 PrintWriter pw = new PrintWriter(socket.getOutputStream(), true);
513                 pw.println("cmd_capture_image");
514
515
516                 ByteArrayOutputStream byteArrayOutputStream = new
517                 ByteArrayOutputStream(1024);
518                 byte[] buffer = new byte[1024];
519
520                 int bytesRead;
521                 InputStream inputStream = socket.getInputStream();
522
523                 while ((bytesRead = inputStream.read(buffer)) != -1)
524                 {
525                     byteArrayOutputStream.write(buffer, 0, bytesRead);
526                     response += byteArrayOutputStream.toString("UTF-8");
527                 }
528             }
529
530             catch (UnknownHostException e)
531             {
532                 e.printStackTrace();
533                 response = "UnknownHostException: " + e.toString();
534             }

```

toto.java

```
535
536         catch (IOException e)
537         {
538             // TODO Auto-generated catch block
539             e.printStackTrace();
540             response = "IOException: " + e.toString();
541         }
542
543         finally
544         {
545             if(socket != null)
546             {
547                 try {
548                     socket.close();
549                 }
550
551                 catch (IOException e)
552                 {
553                     // TODO Auto-generated catch block
554                     e.printStackTrace();
555                 }
556             }
557         }
558         return null;
559     }
560
561     @Override
562     protected void onPostExecute(Void result)
563     {
564         textResponse.setText(response);
565         nom_image = response;
566         super.onPostExecute(result);
567     }
568
569 }
570
571
572 private class LoadImage extends AsyncTask<String, String, Bitmap>
573 {
574     @Override
575     protected void onPreExecute() {
576         super.onPreExecute();
577         pDialog = new ProgressDialog(toto.this);
578         pDialog.setMessage("Loading Image ....");
579         pDialog.show();
580     }
581
582     protected Bitmap doInBackground(String... args) {
583         try
584         {
585             bitmap = BitmapFactory.decodeStream((InputStream)new
586             URL(args[0]).getContent());
587         }
588
589         catch (Exception e)
590         {
591             e.printStackTrace();
592         }
593
594         return bitmap;
595     }
596 }
```

toto.java

```
596
597     protected void onPostExecute(Bitmap image)
598     {
599
600         if(image != null){
601             capture_image.setImageBitmap(image);
602             pDialog.dismiss();
603         }
604
605         else{
606             pDialog.dismiss();
607             Toast.makeText(toto.this, "Image Does Not exist or Network Error",
608 Toast.LENGTH_SHORT).show();
609         }
610     }
611
612
613
614
615 }
```