

tp_thread_android

Thème : Remaniement et déploiement sur la plate forme Android tu [tp thread](#) ([tp_thread_2012.jar](#)) (j2se)

Question1) Acquisition cyclique d'un capteur installé sur le web

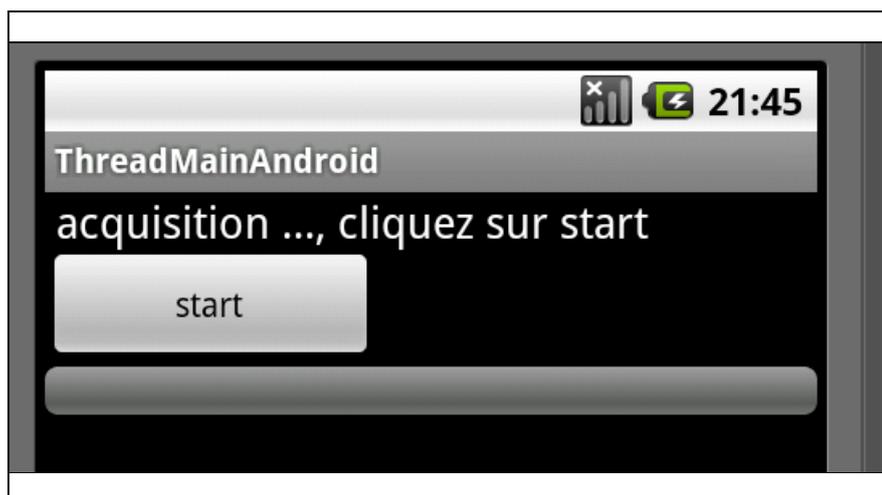
- Ce capteur est un [DS2438](#) qui délivre le taux d'[Humidité Relative](#) d'un bureau au CNAM/Paris 75003.
 - Il est relié au serveur [lmi92.cnam.fr](#) par un adaptateur port série <--> bus [1-Wire](#) de [Dallas Semiconductor/MAXIM](#).
 - Il est accessible à l'URL : <http://lmi92.cnam.fr/ds2438/ds2438/> Une valeur aléatoire est retournée lorsque ce capteur est déconnecté
-
- Notez que si ce service est inaccessible, (coupure réseau, capteur en panne, etc ...)
 - Vous pouvez exécuter cette commande, en téléchargeant cette archive
console> java -cp [simulateurDS2438.jar](#) question1.AppliSimulateurDS2438
Ce serveur délivre une valeur aléatoire du capteur...
 - L'url devient <http://localhost:8999/ds2438/> ou selon vos configuration http://le_nom_DNS:8999/ds2438/ ou encore <http://127.0.0.1:8999/ds2438/>

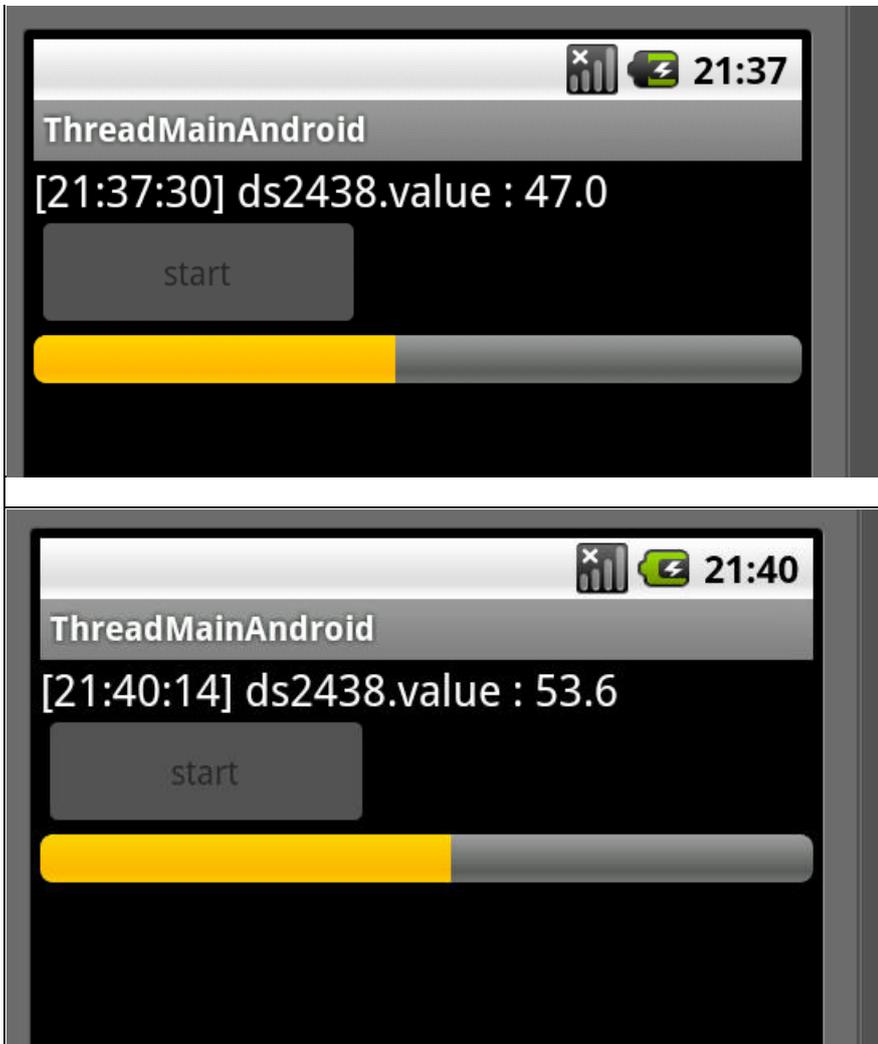
Proposez ce service sur Android, le code original sera remanié, (attention à l'UIThread)

IHM/ Android : Une suggestion d'interface pourrait être l'affichage d'un texte contenant la valeur du capteur et une jauge reflétant la valeur lue,

L'acquisition cyclique débiterait par une action sur le bouton *start*.

A la fin d'une période d'acquisition, l'interface est à nouveau prête pour commencer un nouveau cycle. (Pendant l'acquisition le bouton *start* est inhibé.)





Note: Permission : accès à internet, AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

Une proposition de squelette de programme pour l'activité pourrait être :

```
public class TpThreadQuestion1Activity extends Activity{
    private TextView tv;
    private Button start;
    private ProgressBar progressBar;

    public void onCreate(Bundle savedInstanceState) {...}

    public void onClickStart(View v){ // attribut xml onClick
        this.start.setEnabled(false);
        Acquisition acquisition = new Acquisition();
    }

    private class Acquisition ... implements Command<Float>, Runnable{
        public void make(final Float f){
            final String date = DateFormat.getTimeInstance().format(Calendar.getInstance().getTime());

            // tv.setText "[" + date + "] ds2438.value : " + f);
            // progressBar.setProgress(f.intValue());
        }
    }

    // attention, attention ! ces 2 instructions auront une exécution
    // incertaine ...
    // la solution fait partie de la réponse à cette
    // question
};
```

```

public void run(){
    HTTPSensorAndroid ds2438 = new HTTPSensorAndroid("http://163.173.228.92/ds2438/ds2438/");
    try {
        // Afin de vérifier que tout fonctionne tel quel ...
        // tp.thread.question1.Main.main(new String[]{"http://lmi92.cnam.fr/ds2438/ds2438/"});

        CyclicAcquisition acquisition = new CyclicAcquisition(ds2438, ds2438.minimalPeriod(), this);
        acquisition.start();
        Thread.sleep(30000);
        acquisition.stop();
    }catch (Exception e){
        }
    }
};

```

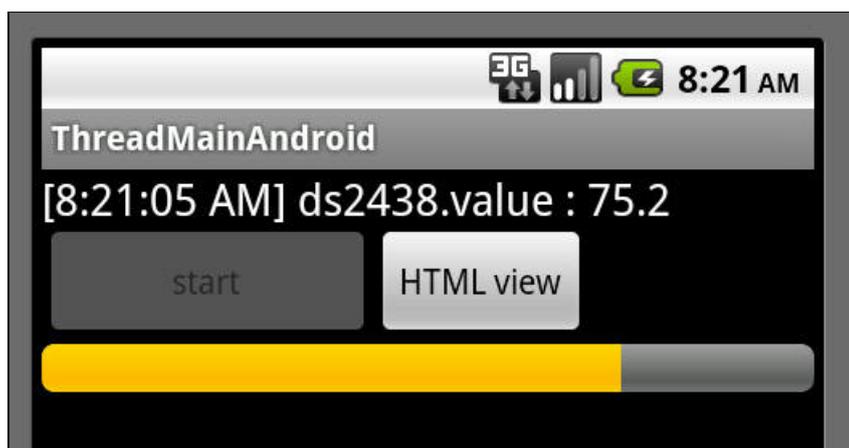
Question2) Lecture du fichier de mesures

- Au tp j2se **FileHandler** enregistre sur fichier, les couples < date, valeur>, un fichier résultat en syntaxe HTML.

Ici un exemple de [fichier de mesures](#) fichier qui pourra être ensuite lu par votre application.

Ajoutez une fonctionnalité à votre application permettant de lire le fichier de mesures généré au format HTML.

Suggestion d'interface : un nouveau bouton *HTML view* de visualisation de la page HTML a été ajouté



Un clic sur le bouton **HTML view** déclenche une nouvelle "Activity", cette *activité* lit et affiche le fichier de mesures placé sur la SD-Card.

Suggestion d'interface résultante

A l'*activité* ainsi déclenchée, est associée et sa *DVM*), (par défaut une activité utilise le même initiateur)

cf. attribut *android:process* d'une *activité* processus nommé ici **tp.thread.webResult**

mesures du taux d'humidité relative	
19/09/12-20:26:12	30.3
19/09/12-20:26:15	33.6
19/09/12-20:26:17	84.4
19/09/12-20:26:19	32.2
19/09/12-20:26:22	93.5
19/09/12-20:26:24	88.5
19/09/12-20:26:26	97.4
19/09/12-20:26:28	31.3

Name	Process ID
com.svox.pico	230
com.android.quicksearchbox	247
com.android.protips	263
tp.thread	267
com.android.music	279
android.process.media	286
com.android.mms	300
com.android.email	316
com.android.settings	333
tp.thread.webResult	348

Vérifiez le comportement analogue

Ce fichier généré sur la SD-Card, est lu depuis une WebView dans une seconde activité et un processus lui est associé.

Extraits du fichier AndroidManifest.xml

Déclaration d'une activity au sein d'une application et création d'un processus dédié

```
<activity android:name=".WebResultActivity" android:process="tp.thread.webResult"></activity>
```

Permission : accès à la SD-Card

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
```

1) Extrait de code permettant d'écrire un fichier sur la SD-Card

```
private void writeHTML() throws IOException{
    File root = Environment.getExternalStorageDirectory();
    BufferedWriter bw = new BufferedWriter(
        new PrintWriter(
            new FileOutputStream(
                new File(root, fileName + ".html"))));
```

2) Extrait de code permettant de lire un fichier HTML depuis la SD-Card et son affichage dans une webView (cf.méthode loadUrl)

```
String path = Environment.getExternalStorageDirectory().getAbsolutePath();
webView.loadUrl("file:///"+ path + File.separator + nomDuFichier);
```