public class **AlarmManager** extends Object　　android.app.AlarmManager

Android システムには「**アラームサービス**」と言う「**スケジューラ**」があり，AlarmManager クラスを利用して，将来のある時点で指定したアプリケーションが実行されるようにスケジュールを登録することができる。

参考　http://android.keicode.com/basics/services-schedule-with-alarmmanager.php

スケジュール登録法

①アプリケーションを起動するインテント (Intent) を作成

②インテント (Intent)をベースに、ペンディングインテント(PendingIntent) を作成

③アラームマネージャ (AlarmManager) から，時刻とペンディングインデントを指定して，スケジュールを登録

③スケジュール登録（任意の日時で実行する処理を登録）

void set(int type,　long triggerAtTime,　PendingIntent operation)

一定間隔で実行する処理を登録

void setRepeating(int type,　long triggerAtTime,　long interval,　PendingIntent operation)

スケジュールを削除　　**AlarmManager.INTERVAL_DAY**

void cancel(PendingIntent operation)

set，setRepeating メソッドの第 1 引数（int type）には第 2 引数で指定する起動時間に，ペンディングインテントをどのように設定するかを指定する。

int type の例：

RTC_WAKEUP　UTC 時刻で指定。スリープ状態のときは電源を ON にしてくれる

RTC　UTC 時刻で指定

ELAPSED_REALTIME　電源 ON からの経過時間で指定

ELAPSED_REALTIME_WAKEUP　電源 ON からの経過時間で指定。スリープ状態のときは電源を ON にしてくれる

RTC【Real Time Clock】マザーボード上に実装されている、計時専用のチップ。他のマザーボード上のチップと違って、電源が切られている間も内蔵電池から電源供給を受けて動作している。

UTC【Universal Time, Coordinated】協定世界時

| Sample program : AlarmSample　（http://techbooster.jpn.org/andriod/application/2166/）　|

```
// ReceivedBR ( extends BroadcastReceive )を呼び出すインテントを作成
Intent i = new Intent( getApplicationContext(),　ReceivedBR.class );　　// ①

// ブロードキャストを投げる PendingIntent を作成
PendingIntent sender = PendingIntent.getBroadcast( AlarmSample.this,　0,　i,　0 );　　// ②
```

static PendingIntent **getBroadcast**(Context context,　int requestCode,　Intent intent,　int flags)

context	The Context in which this PendingIntent should perform the broadcast.
requestCode	Private request code for the sender (currently not used).
Intent	The Intent to be broadcast.
flags	
FLAG_CANCEL_CURRENT	if the described PendingIntent already exists, the current one is canceled before generating a new one.
FLAG_UPDATE_CURRENT	if the described PendingIntent already exists, then keep it but its replace its extra data with what is in this new Intent.

getActivity(Context context,　int requestCode,　Intent intent,　int flags)

getService(Context context,　int requestCode,　Intent intent,　int flags)

```
Calendar calendar = Calendar.getInstance();      // Calendar インスタンスを取得
    calendar.setTimeInMillis( System.currentTimeMillis() );     // calendar に現在日時を設定
                                //System.currentTimeMillis()1970 年 1 月 1 日からの今までの経過時間（ミリ秒）
    calendar.add(Calendar.SECOND, 15);     //calendar の時刻の秒に 15 を加える
```

> **Calendar** is an abstract base class for converting between a **Date** object and a set of integer fields such as **YEAR**, **MONTH**, **DAY**, **HOUR**, and so on.

```
// システムから AlarmManager の参照を取得
AlarmManager am = (AlarmManager)getSystemService( ALARM_SERVICE );     // ◎
// AlramManager に PendingIntent を登録
am.set( AlarmManager.RTC_WAKEUP,    calendar.getTimeInMillis(),    sender );    // ③
```

java.lang.Object
↳android.content.BroadcastReceiver

受け手となる **ReceivedBR** は,
呼び出されて起動すると Toast を表示する。

java.lang.Object
↳android.content.Context
↳android.content.ContextWrapper
↳ android.view.ContextThemeWrapper
↳ android.app.Activity

```
public class ReceivedBR extends BroadcastReceiver {
    public void onReceive( Context context, Intent intent ){
                Toast.makeText( context,    "called ReceivedActivity",    Toast.LENGTH_SHORT ).show();
    }
}
```

なお、AndroidManifest に BroadcastReceiver を有効にするたに以下を記述しておく。
<**receiver** android:name=".**ReceivedBR**" android:process=":remote" />

BroadcastReceiver から Activity を起動するには

```
    public void onReceive(Context context, Intent intent) {
        //インテントに setData された URI データがあるならば          Uri uri = intent.getData()
        // 新しいインテントの生成
        Intent newIntent = new Intent( context,    起動する Activity.class );
        // URI データを受け渡す必要があれば    newIntent.setData( uri )

     // Intent.FLAG_ACTIVITY_NEW_TASK：タスクがスタックに存在しても新しいタスクとして起動
     // Activity 以外から Activity を呼び出す場合にも必要
        newIntent.setFlags( Intent.FLAG_ACTIVITY_NEW_TASK );
        // アクティビティ開始
        context.startActivity( newIntent );
    }
```