A function to get the battery level without receiving updates would look something like this:

public float getBatteryLevel() {

Intent batteryIntent = registerReceiver(null, new

IntentFilter(Intent.ACTION\_BATTERY\_CHANGED));

int level = batteryIntent.getIntExtra(BatteryManager.EXTRA\_LEVEL, -1);

int scale = batteryIntent.getIntExtra(BatteryManager.EXTRA\_SCALE, -1);

// Error checking that probably isn't needed but I added just in case.

if(level == -1 || scale == -1) {

return 50.0f;

}

return ((float)level / (float)scale) \* 100.0f;

}

= = = = = =

Since API 21 its been possible to use the following to get current battery level as a percentage:

BatteryManager bm = (BatteryManager)getSystemService(BATTERY\_SERVICE);

int batLevel = bm.getIntProperty(BatteryManager.BATTERY\_PROPERTY\_CAPACITY);

= = = = = =

Whenever there is a change in the charge, it will fire the class and show the level in percentage ?

public class Main extends Activity {

private TextView batteryTxt;

private BroadcastReceiver mBatInfoReceiver = new BroadcastReceiver(){

@Override

public void onReceive(Context ctxt, Intent intent) {

int level = intent.getIntExtra(BatteryManager.EXTRA\_LEVEL, 0);

batteryTxt.setText(String.valueOf(level) + "%");

}

};

@Override

public void onCreate(Bundle b) {

super.onCreate(b);

setContentView(R.layout.main);

batteryTxt = (TextView) this.findViewById(R.id.batteryTxt);

this.registerReceiver(this.mBatInfoReceiver, new IntentFilter(Intent.ACTION\_BATTERY\_CHANGED));

}

}

= = = =

Aware Arduino to open / close a relay to recharge

**import** android.app.Service;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.os.BatteryManager;  
**import** android.os.IBinder;  
**import** android.support.annotation.Nullable;  
**import** android.widget.Toast;  
  
**public class** ServiceBattery **extends** Service {  
  
 **private** BroadcastReceiver **mBatInfoReceiver** = **new** BroadcastReceiver(){  
 @Override  
 **public void** onReceive(Context ctxt, Intent intent) {  
 **int** level = intent.getIntExtra(BatteryManager.***EXTRA\_LEVEL***, 0);  
  
 **if** (level < 80 ) Processe(level);  
 **if** (level > 85) Processe(level);  
 *// batteryTxt.setText(String.valueOf(level) + "%");* }  
 };  
   
 **private void** Processe(**int** level)  
 {  
 Toast.*makeText*(getApplicationContext(),**" nivel "**+ String.*valueOf*(level), Toast.***LENGTH\_LONG***).show();  
 }  
   
 *// @Override* **public void** onCreate() {  
 **super**.onCreate();  
  
 **this**.registerReceiver(**this**.**mBatInfoReceiver**,  
 **new** IntentFilter(Intent.***ACTION\_BATTERY\_CHANGED***));  
 }  
  
 @Nullable  
 @Override  
 **public** IBinder onBind(Intent intent) {  
 **return null**;  
 }  
}

Boot class

**public class** BootUpReceiver **extends** BroadcastReceiver {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
  
**try** {  
 Class aClass = Class.*forName*(**"com.example.tdal6392.busqueda.ServiceBattery"**);  
 Intent serviceIntent = **new** Intent(context, aClass);  
 *//serviceIntent.putExtra("interval", 1 \* 1000L);//every second* **context.startService(serviceIntent);**  
 } **catch** (ClassNotFoundException e) {  
 Toast.*makeText*(context, e.toString(), Toast.***LENGTH\_LONG***).show();  
 *//e.printStackTrace();* }  
  
 ***// START ACTIVITY ON BOOT*** *//Intent i = new Intent(context, MainActivity.class);  
 //i.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);  
 //i.putExtra("BOOT",1);  
 //context.startActivity(i);* }  
}

Manifest

<**uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"** />

* - - - -

<**service android:name=".ServiceBattery"** />  
  
<**receiver android:enabled="true" android:name=".BootUpReceiver"  
 android:permission="android.permission.RECEIVE\_BOOT\_COMPLETED"**>  
  
 <**intent-filter**>  
 <**action android:name="android.intent.action.BOOT\_COMPLETED"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
 </**intent-filter**>  
</**receiver**>