

**MACHINE**

Battery+Hardware\_M1            //    第1段階の詳細化(充電制御ロジックの導入)

**REFINES**

Battery+Hardware\_M0

**SEES**

Battery+Hardware\_C0

**VARIABLES**

```
recharge           //    リチャージフラグ
solar_panels_opened    //    ソーラーパネルが開いているか
is_charging        //    現在充電中かどうか
currentPosition    //    ローバの現在位置
batteryLevel
```

**INVARIANTS**

```
inv1  : recharge ∈ B00L
inv2  : solar_panels_opened ∈ B00L
inv3  : is_charging ∈ B00L
inv4  : recharge = TRUE ⇒ batteryLevel ≤ MAX_BATTERY        //    充電が必要な状態とバッテリーレベルの関係
inv5  : currentPosition ∈ POSITION
```

**EVENTS**

**INITIALISATION**    ≐  
extended

**STATUS**

ordinary

**BEGIN**

```
act_init_battLevel  : batteryLevel = MAX_BATTERY        //    バッテリーレベルの初期状態は満充電状態
act_init_recharge   : recharge = FALSE
act_init_solar_panels : solar_panels_opened = FALSE
act_init_is_charging : is_charging = FALSE
act_init_currentPosition : currentPosition = START_POSITION
```

**END**

**Consume\_Battery\_Refined**    ≐            //    抽象モデル(M0)の具体化  
extended

**STATUS**

ordinary

**REFINES**

Consume\_Battery

**WHEN**

```
grd_batt_min  : batteryLevel > MIN_BATTERY        //    バッテリーレベルが最小より大きい場合だけ消費可能
grd_rechargeOff : recharge = FALSE            //    充電が必要なフラグが立っていない場合のみ消費(移動)できる
```

**THEN**

```
act1  : batteryLevel :| batteryLevel' < batteryLevel ∧ batteryLevel' ≥ MIN_BATTERY
```

**END**

**Signal\_Recharge**    ≐            //    充電が必要かどうか判断

**STATUS**

ordinary

**WHEN**

```
grd_arrive_charger : currentPosition ∈ chargers        //    ローバーが充電位置にいる場合
grd_rechargeOff    : recharge = FALSE            //    リチャージフラグが立っていない
```

**THEN**

```
act1  : recharge = TRUE
```

**END**

**Start\_Charging**    ≐            //    充電開始

**STATUS**

ordinary

**WHEN**

```
grd1  : recharge = TRUE            //    リチャージフラグがONの場合
```

**THEN**

```
act1  : is_charging = TRUE
act2  : solar_panels_opened = TRUE        //    ソーラーパネルを開く(HI6)
```

**END**

**Recharge\_Battery**    ≐            //    充電中

**STATUS**

ordinary

**REFINES**

Charge\_Battery

```
WHEN
  grd1 : is_charging = TRUE          // 充電中であること
  grd3 : batteryLevel + step_charge < MAX_BATTERY    // バッテリーレベルがMAX未満
THEN
  act1 : batteryLevel := batteryLevel + step_charge
END

Complete_Charging ⇐                // 充電完了
STATUS
  ordinary
WHEN
  grd1 : is_charging = TRUE
  grd2 : batteryLevel + step_charge ≥ MAX_BATTERY    // バッテリーレベルがMAXになったら
THEN
  act1 : is_charging := FALSE          // ロボットの充電終了
  act2 : solar_panels_opened := FALSE    // ソーラパネルも閉じる
  act3 : recharge := FALSE             // リチャージフラグもFALSE
END

END
```