# Power Modelling and Power Analysis of Electronic Tag: A Case Study Using Probabilistic Model Checker PRISM

BOJAN NOKOVIC, EMIL SEKERINSKI

DRAFT DOCUMENT

MARCH 2011

SQRL REPORT 63
MCMASTER UNIVERSITY

---

*This work was funded in part by NSERC Canada and by Lyngsoe Systems Canada*

# Abstract

In this work we calculate daily power consumption of an electronic tag, a battery-powered mobile device used in postal systems. First, we have *paper and pencil* calculation of power consumption by building FSM model of a device and identified all *activities* as possible sources of power consumption. In next step, we show how the number of activities can be reduced by combining them, and abstract model built. The model is designed as a discrete-time Markov chain with *rewards*. Quantitative properties are specified in probabilistic linear temporal logic and automatically analyzed by probabilistic model checker PRISM. With our model, we verified that the *sleep* mode accounts for the biggest share of total power consumption and that a possible improvement of *power management* during communication, will not lead to a significant decrease in power consumption. This may be surprising for electronic tags in general, but in our case it is expected result, since the tag is used in specific application where most of the time is in the sleep mode.

# 1 Introduction

The PT23 tag is part of a family of postal tags designed by Lyngsoe Systems [?] to be used in the postal industry for Automatic Mail Quality Measurements (AMQM$^{TM}$). The AMQM system calculates terminal dues, the fees postal operations pay to each other for delivery of cross-border mail that IPC (International Post Corporation) handles for 55 of the national posts in Europe, Asia Pacific, and North America. It measures how fast mail travels from one point to another by storing tag serial numbers and recording *time stamps* (time when message from tag is received). Collection sites are located at strategic congestion points, such as the entrance and exit gates, conveyers, sorting machines, etc. The tag is a wireless transponder that receives an excitation signal and responds by transmitting back a message. It operates a simple protocol, but offers good flexibility and many tuning possibilities in terms of protocol adjustment. PT23 belongs to group of *active* radio frequency identification devices because it has its own source of power. The tag is built according to *power optimized design* and has two special low-power operational modes. In this paper we explore possible decrease of power consumption by implementation of *power management* technique [?]. We build a power model based on a *functional breakdown* methodology and calculate *daily* power consumption following five step design process described in [?, ?]. In our calculation we used as input domain parameters from the PT23 technical documentation [?]. We identifiy all activities as possible sources of power consumption. Those activities we call *logical activities* (LA). In the next step we reduced number of logical activities by combing operations at particular states.

By high-level modelling language we described the model as discrete time Markov chain (DTMC) with reward, using probabilistic model checker PRISM [?]. With this model, by specifying properties in form of probabilistic linear temporal logic formulae (PLTL)[?] , we were able to automatically calculate daily power consumption. By changing values of some of the logical activities we show how to *quantify* the impact of each activity on total power consumption. We also show how the tag's battery life time can be calculated exactly using a Markov reward model and approximately based on a Monte Carlo simulation implemented in Matlab. In the Monte Carlo simulation we assume *uniform* distribution on the input domain data.

For electronic tags, a power is usually the most critical design constrain, which may have main impact on commercial success of the product. Battery life for PT23 used in AMQM system has to be at least five years and the tag life ten years. Number of a postal tags used in the system is hundred of thousands, so if battery life is shorter than five years, that means extra battery replacement which is very expensive taking into account number of tags in the system. In this work we calculate that a battery life for PT23 used in AMQM system is more than five years, and identify which activities contribute the most to the power consumption.

We present the methodology of building abstract model of application related power consumption on a simple protocol. But, it is applicable and specially useful for complex protocols, where both calculation by paper and pencil, and design of a model without abstraction can be very difficult because of big number of activities.

## 2 Overview of the Protocol

When the tag enters an excitation area of 125KHz LF (low frequency), the tag wakes-up from *sleep* mode and transmits a preprogrammed number of messages. We call this excitation *valid*. Any LF field can wake-up the tag from sleep mode, but if the tag does not recognize the *field identification number*, it will not sent the message, and we call this a *false* excitation. Each tag has an unique identification number. Since all tags transmitting at the same centre frequency of 433.92MHz of the ISM (Industrial Scientific Medical) band, there is the possibility of message collision. Also the message can be lost because of (1) interference and (2) multi-path propagation. Interference may come from other sources, which use same frequency band, and multi-path propagation occurs when an radio frequency (RF) signal takes different paths propagating from a tag to a RF reader. There is no acknowledge message in this protocol, so the tag cannot know if transmitted message is received. Those are the reason why the tag transmits the same message many times in randomized period of time. Tags used in AMQM system transmit 40 messages per valid excitation, and no message on false excitation. Number of valid and false excitations per day depends of the way how the tags are used. In our model, the average number of valid excitations per day is five ($n_{ve} = 5$), and number of false excitations per day is ten ($n_{fe} = 10$)[1]. We combine those two types of excitations into probabilistic one; we consider 15 excitations per day, $\frac{1}{3}$ valid and $1 - \frac{1}{3}$ false.

In normal operation, the tag is under LF field only for a short period of time, which is usually less than transmission time. So, by the time the transmission at ultra high frequency (UHF) is finished, the device is out of a field and tag goes into sleep mode. But, if the device is by any reason left under the LF field, it will

---

[1] This information is based on observation of the tag used in AMQM system

go into special *low field standby* mode. This is done to avoid continuous wake up from standard *stanby* mode if the tag is left under LF field. In our calculation we estimate that, in average in AMQM system, the device spend in *low field standby* mode three minutes per day.

# 3 Power Modelling Methodology

We applied the model proposed in [**?**] on the protocol of the PT23 tag. The model has two parts, an *implementation independent* and an *implementation dependent* part.

The implementation independent part is consisted of: statechart or FSM (finite state machine) modelling, activity identification, and state-activity matching. The implemetation dependent part of the model includes characterization and validation.

## 3.1 Implementation Independent Model

Based on specification of the PT23 protocol, we designed statechart representation. The design has two concurrent states *Protocol* and *Field*. The *Protocol* has four states SB (*standby*), TX (*transmission*), RX (*receive*), and SBLF (*standby in low frequency field*). State *Field* represent environment impact on PT23 and has two states, ON when low frequency excitation field is present and OFF when there is no field. Initial states are SB and OFF. On the event *FieldPresent*, the value of broadcast *lfField* variable is assigned to one, and that alows transition form SB into RX state. If valid excitation field is recognized, on action *validExciatation*, the system is going into TX state in which transmits on UHF channel preprogrammed number of messages. If, after transmission, the tag is out of the excitation field, it goes into initial SB state, but if the field is still present, the tag is going into SBLF state. When the field disappear *lfField* is set to zero, and PT23 goes into SB state. If in RX state the valid excitation field is not recognized, the event *falseExcitation* is generated and PT23 goes into SBLF state. The state model at Figure 1 represents the specification of the protocol. In order to design power model, we identify all activities as possible source of power consumption. In the *Field* states power consumption is zero, and in *Protocol* states power consumption depends on HW/SW implementation. Each state in the protocol may have more than one *logical activities*, i.e., TX state which has two logical activities.

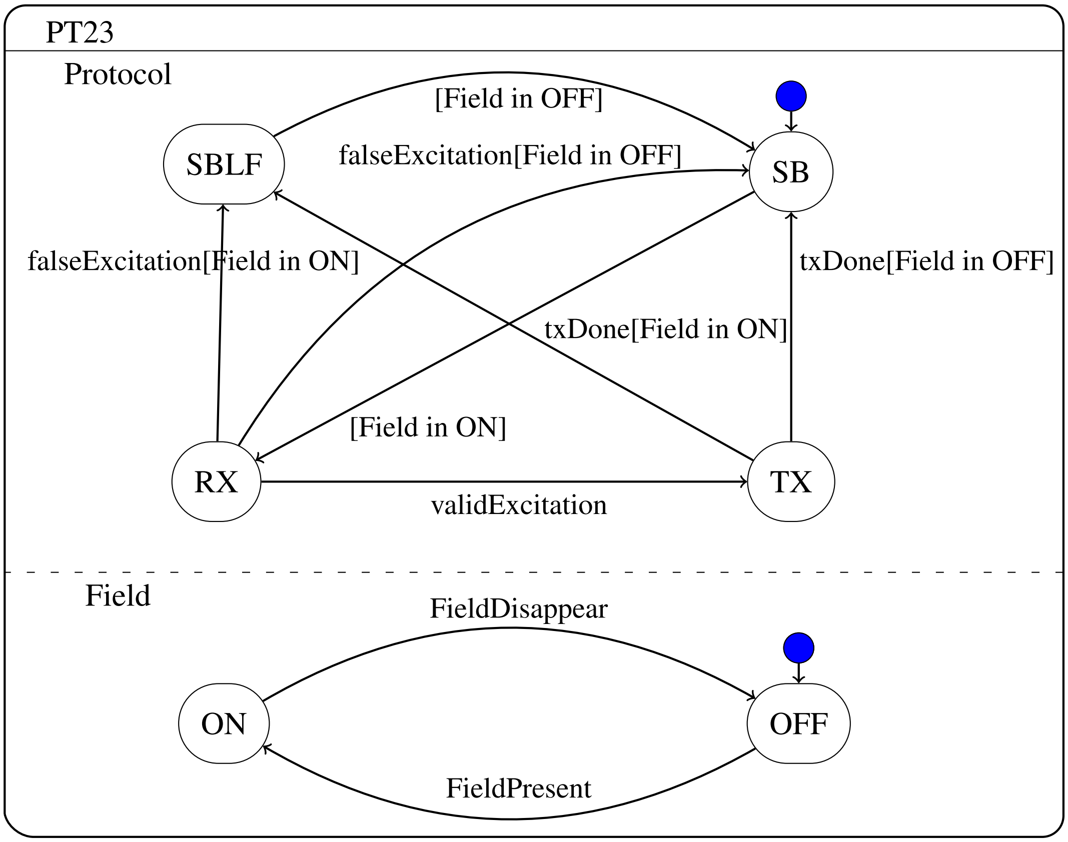

Figure 1: *Postal tag statechart representation*

## 3.2 Activities Identification

Let $S = \{SB, RX, TX, SBLF\}$ be the set of states of the PT23 protocol. We need to identify all activities as possible sources of power consumption. For PT23 we identifiy six logical activities. Those are:

- $a_1$ - *UHF Transmit*; This is the most power consuming activity because UHF transmitter is turned ON during tag message transmission.

- $a_2$ - *Inter-message delay*; The messages transmitted by the tag are randomized and the time between two consecutive transmissions is called inter-message delay. During this period the UHF transmitter is OFF, so power consumption is much lower than in activity $a_1$.

- $a_3$ - *LF Receive*; This is activity during receiving process, when the LF field is sampled.
- $a_4$ - *Exciter Validation*; In this activity sampled values of theLF field are processed in an attempt to recognize the LF field ID.
- $a_5$ - *Standby LF Field*; This is sleep mode when the tag is under LF field.
- $a_6$ - *Standby*; This is the activity of standard sleep mode.

## 3.3 State-Activity Matching

Six identified logical activities are assigned to particular state shown in Figure 2. In the TX and RX states, there are two logical activities, and in SB and SBLF states there is just one.

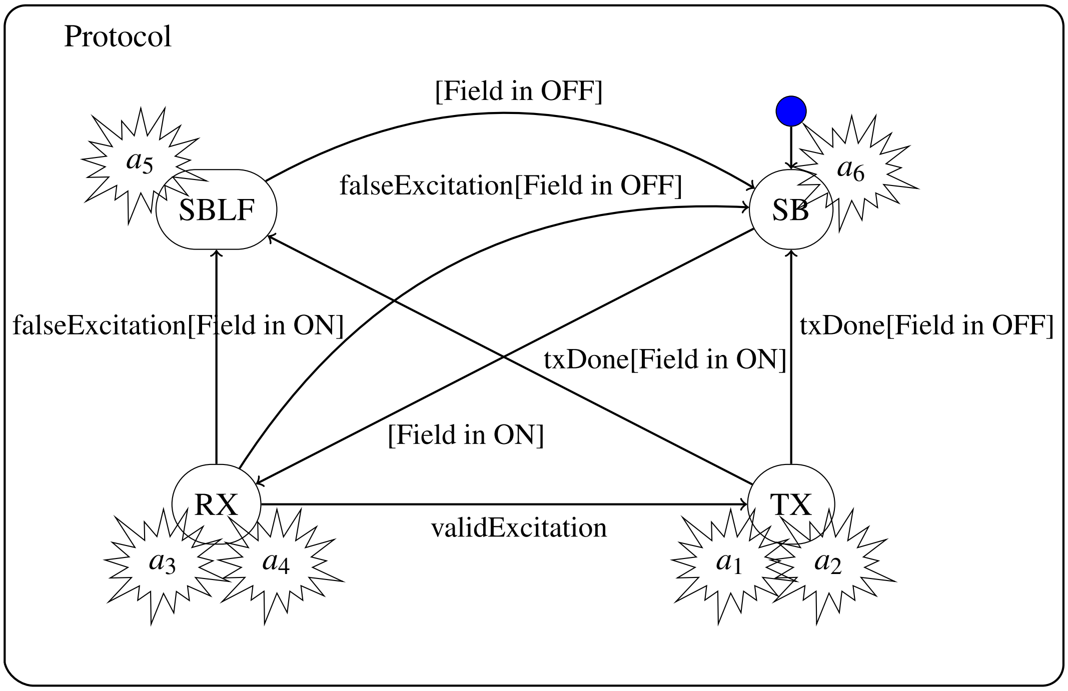

Figure 2: *States with associated activities*

## 3.4 Implementation Dependent Model

Using information from [**?**] and data from test performed at Lyngsoe Canada, we assigned particular values of current consumption for each logical activity. The highest consumption of 9.2[mA] is in UHF transmission activity, and lowest 2.4[$\mu$A] in standby mode. All values are shown in Table 1:

| **Act.** | **Name** | **Current Cons. [mA]** | **Description** |
|---|---|---|---|
| $a_1$ | TX | 9.2 | UHF Transmit |
| $a_2$ | TXIM | 0.5 | Inter-message delay |
| $a_3$ | RX | 0.5 | LF Receive |
| $a_4$ | EXID | 0.5 | Exciter Validation |
| $a_5$ | SBLF | 0.15 | Standby LF Field |
| $a_6$ | SB | 0.0024 | Standby |

Table 1. *Logical activities for postal tag PT23*

To calculate power consumption, we need to know how much time the device spend at each logical activity. Values for $t_1, t_2, t_3$, and $t_4$ are taken from the technical specification [**?**], and value $t_6$ is based on observation for tags used in AMQM system. Those data is shown in Table 2.

| | **Time [ms]** | **Description** |
|---|---|---|
| $t_1$ | 5.5 | Message duration, average |
| $t_2$ | 88 | Inter message time interval, average |
| $t_3$ | 1200 | Time to recognize a false excitation, maximal |
| $t_4$ | 100 | Time to recognize a valid excitation, average |
| $t_5$ | 60000 | LF Field large standby time |

Table 2: *PT23 protocol related timings*

# 4 Daily Power Consumtion

In our calculation of daily power consumption, as input parameters we use: *number of messages per transmutation*, *number of valid*, and *number of false excitations* as they are presented in Section 2. We need first to calculate time spent in each logical activity per day and to multiply that time with activity current consumption. For instance time spent in UHF transmission state per day is

$$t_{1d} = t_1 \cdot n_{ve} \cdot 40 = 5.5 \cdot 5 \cdot 40 \cdot = 1.1[s]$$

where $n_{ve} = 5$ is number of valid excitations per day, and 40 is number of messages transmitted on valid excitation. Daily power consumption in activity $a_1$ is

$$p_1 = a_1 \cdot t_{1d} = 10.12[mAs]$$

In a similar way we calculate daily power consumption for other logical activities. The values are shown in Table 3.

| **Power** | **[mAs]** | **Description** |
|---|---|---|
| $p_1$ | 10.12 | Power for UHF transmission per day |
| $p_2$ | 8.8 | Power of inter-message transmission per day |
| $p_3$ | 6 | Power for false receive per day |
| $p_4$ | 0.25 | Power for valid receive per day |
| $p_5$ | 27 | Power for LF field standby per day |
| $p_6$ | 206.89 | Power for standby per day |
| $p_{td}$ | 259.02 | Total power per day |

Table 3: *Power per logical activity per day*

## 4.1 Battery Life Time

PT23 runs on ”Renata CR2320” lithium coin cell 150[mAh], 3 volts battery. Since daily power consumption is 259.02 [mAs], battery life time is

$$\frac{150}{p_{td}} \cdot 3600 = 2084.78[days],$$

or 5.71 years. If we take into account battery self discharge, which is 1% per year, battery lifetime is $\approx$ 5.56 years.

To see impact of particular logical activities or number of excitations per day on battery life, we need to change those values, and manually recalculate daily power consumption and battery life. That can be time consuming and prone to error in calculation. Better approach is to do the same calculation automatically and quantify impact of those parameters on power consumption and battery life. To do that, we need to design probabilistic model.

# 5 Logical Activities Reduction

Time-complexity of model-checking algorithms is exponential in number of program variables, states and transitions [**?**]. This is also called the state explosion problem. Reduction in number of program variables, which are in our case logical activities, is one way to reduce impact of this problem on model checking algorithm complexity.

We can decrease the number of logical activities by calculating the average consumption per state. This may be done by linearly combining two or more logical activities. In this way, we can create equivalent model with less logical activities.

## 5.1 AND Abstraction

In our TX state, during message transmission there is UHF high power period $t_1$, and delay between two consecutive UHF transmissions $t_2$. Associated current consumptions are $a_1$ and $a_2$. Instead of calculating total power consumption in TX state as

$$a_1 \cdot t_1 + a_2 \cdot t_2$$

we can calculate is as

$$a_{12} \cdot t_{12}$$

where $t_{12} = t_1 + t_2$ and $a_{12} = a_1 \cdot \frac{t_1}{t_{12}} + a_2 \cdot \frac{t_2}{t_{12}}$. So, logical activities $a_1$ and $a_2$ over times $t_1$ and $t_2$ could be replaced by single activity $a_{12}$ over time $t_{12}$

## 5.2 XOR Abstraction

The device can be excited by valid or false LF field. If excited by a valid field, it will recognize the field ID in time $t_4$, and if excited by field with a false ID, it will take on average time $t_5$ to recognize it. We can calculate average consumption as:

$$a_{34} \cdot n_e$$

where $a_{34} = a_3 \cdot \frac{n_{ve}}{n_e} + a_4 \cdot \frac{n_{fe}}{n_e}$ and $n_e$ = $n_{ve}$ + $n_{fe}$. Logical activities $a_3$ and $a_4$ on valid and false excitation can be replaced by single $a_{34}$ activity. Average time of activity is

$$t_{34} = t_3 \cdot n_{fe}/(n_{ve} + n_{fe}) + t_4 \cdot n_{ve}/(n_{ve} + n_{fe}).$$

Values of reduced logical activities and timings are shown in Tables 4 and 5.

| **Act.** | **Name** | **Current Cons. [mA]** | **Description** |
|---|---|---|---|
| $a_{12}$ | TX & TXIM | 1.01 | Average $a_1$ & $a_2$ |
| $a_{34}$ | RX & EXID | 0.5 | Average $a_3$ & $a_4$ |
| $a_5$ | SBLF | 0.15 | Standby LF field |
| $a_6$ | SB | 0.0024 | Standby |

Table 4: *Reduced logical activities*

| **Int.** | **Time [ms]** | **Description** |
|---|---|---|
| $t_{12}$ | 93.5 | Avrg. time of $a_{12}$ |
| $t_{34}$ | 833 | Avrg. time of $a_{34}$ |
| $t_5$ | 60000 | LF standby time |

Table 5: *Timings*

We first calculate daily timings for each activity: $t_{12d}$, $t_{34d}$, and $t_{5d}$ and finally time in standby mode

$$t_{6d} = 86400 - t_{12d} - t_{34d} - t_{5d}.$$

Instead of calculating power consumption for *six*, we do calculation for only *four* logical activities, and the result, daily power consumption $e_{td}$ is the same as the one from Table 3.

# 6 Automated Verification - Exact Method

Automated verification can support the software development process, but in this case study, we use it to verify our methodology of battery life time calculation. We know the probability of making transitions between states, so we build a probabilistic model of the system as a discrete time Markov chain (DTMC). Transitions are governed by a (discrete) probability distribution on the target states [**?**].

## 6.1 Tool

For our quantitative verification we use the PRISM [**?**, **?**] model checker which accepts DTMC probabilistic model described in high-level modelling language. Properties are specified using PCTL (Probabilistic computational tree logic) [**?**] which includes both the probabilistic and reward operations. PRISM supports the notion of experiments, which allows us to plot the outcome of power consumption as functions of time (days).

## 6.2 Experimental results

We compute the expected power consumption as a reachability reward formula $R = ?[F\Phi]$ which corresponds to the expected cumulated cost (in this case, time) of the system until condition $\Phi$ is satisfied. In our case we calculate power consumption per day, and we use formula $R = ?[F d = D]$, where $d$ is variable to count days and $D$ is constant. For $D = 1$, $D = 7$, or $D = 365$ we automatically calculate daily, weekly or annual power consumption.
Our automatic calculation for daily power consumption is 259.01[mAs], and battery lifetime is 5.71 years, which is almost the same as result which we got by *paper-and-pencil* calculation in previous section. The difference between manual and automatic calculation only 0.006%.
Since the biggest contribution to the daily power consumption is consumption in standby (SB) mode in which tags spend 99.75% of time, we did experiment for our specified 0.0024[mA] consumption, as well as for 0.002[mA] and for 0.0016[mA]. Power consumption in seven days (one week) is shown on Figure 3, is generated automatically by PRISM. Weekly power consumption for $a_6$ values of 0.0024[mA] and 0.0016[mA] are 1813.04[mA] and 1330.38[mA]. So by decreasing power consumption in standby mode by 33%, we can decrease total power consumption by $\approx 27\%$.

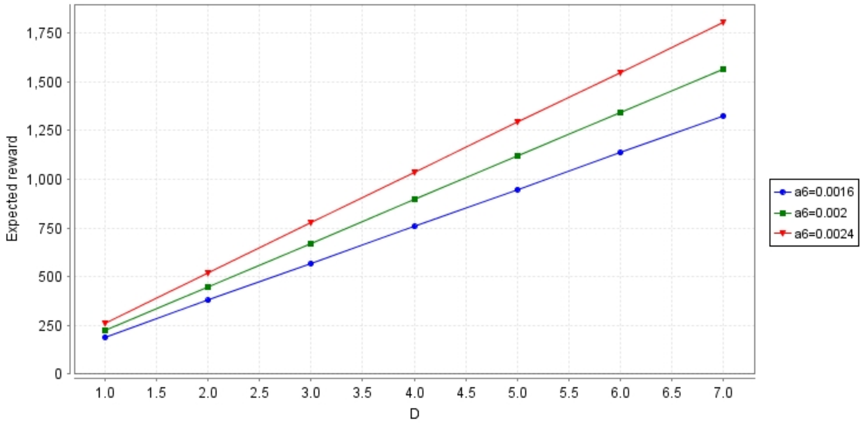

Figure 3: *PT23 power consumption in one week for different $a_6$ (SB) consumptions*

On the other side, although the highest current consumption is during transmission period, we can see from Figure 4, that decreasing current consumption during transmission $a_{12}$ for almost 50%, from 1.01[mA] to 0.51[mA], will decrease weekly power consumption from 1812.91[mA] to 1747.46[mA], which is only 4.6%.

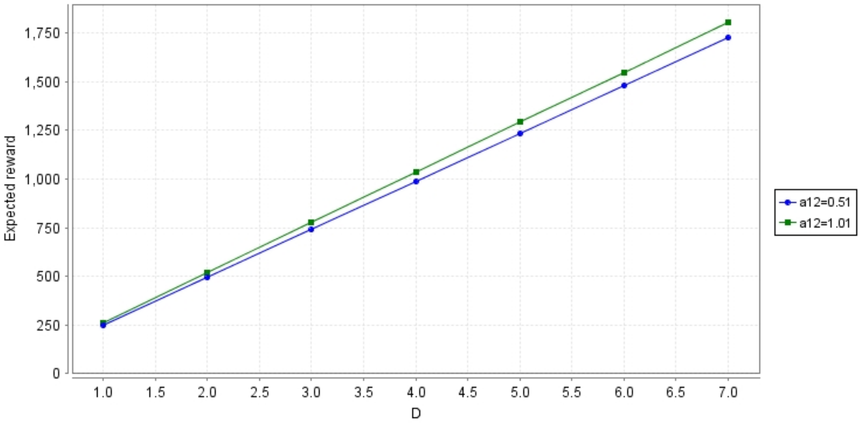

Figure 4: *PT23 power consumption in one week for different* $a_1 2$ *(TX) consumptions*

The source code in high-level modelling language for PRISM is shown in the appendix A.

# 7 Monte Carlo Simulation – Approximate Method

Instead of constructing the full state-transition graph of the model, we can use Monte Carlo methods to estimate power consumption and battery life time. This is done by taking input domains of valid and false excitation and generate random inputs using *uniform* probability distribution and performed individual calculation for each input. We did calculation for 10000 inputs and results are presented as a histogram in which the distribution of calculated values is shown Figure 5. The values are presented as 60 equally spaced containers where the number of elements in each container is shown on the x-asis.

In the worst case scenario, when the number of excitation is maximal, 10 valid and 20 invalid excitations, battery life time will be $\approx 5.63$ years, and in the best case scenario, when number of valid and false excitations is just one each, the battery life time will be $\approx 5.82$ years. This is another confirmation that battery life time is mainly determined by $a_6$ logical activity which is current consumption in standby mode.

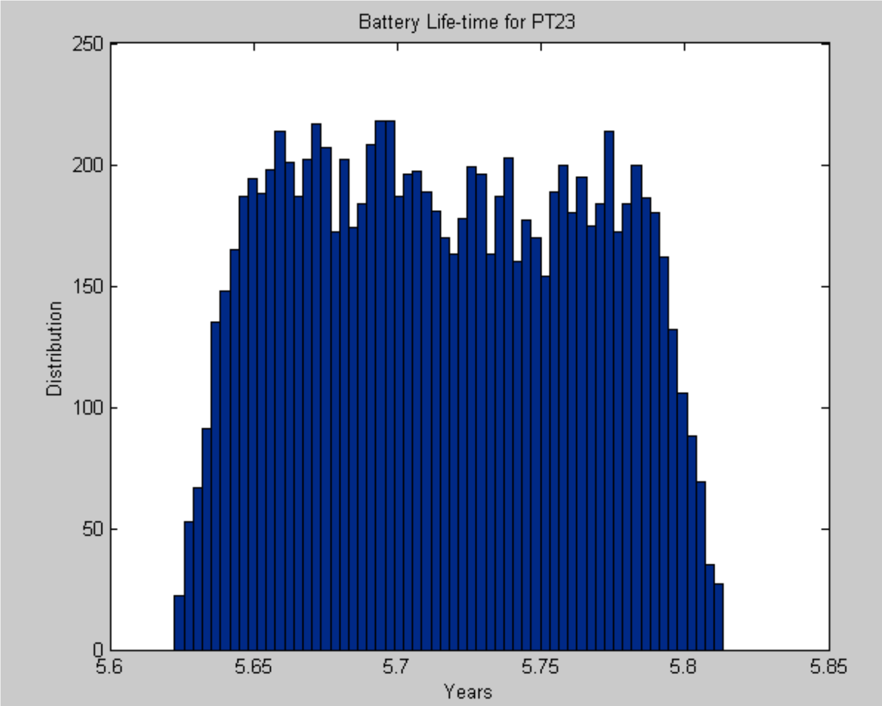

Figure 5: *Battery life-time*

## 8 Conclusions and Future Work

We show that methodology for calculation of power consumption presented in [**?**] is suitable for calculating daily power consumption and battery life time for electronic tag device. We also demonstrated how logical activities can be reduced, and abstract model for quantitative power analysis built. On that model we were able to automatically calculate power consumption of the tag and to analyze impact of particular logical activities on overall power consumption. We also show how distribution of expected battery lifetime can be calculated by approximate Monte Carlo method using uniform distribution of random input parameters. Future work would be to apply the same methodology, and built more probabilistic model for more complex communication protocols, like ISO 18000-7 RFID protocol.

# Appendix A

## Prism Code

```
// PT23 Model
dtmc                            // Discrete time markov chain

const int VE=5;                 // # of valid exct. per day
const int FE=10;                // # of false exct. per day
const int N=VE+FE;              // Total # of exct. per day
const int LF_FE=2;              // Tag in low field, false exct.
const int LF_VE=1;              // Tag in low field, valid exct.

const double t12=0.0935;        // Average time of UHF message
    transmission
const double t34=0.833;         // Average time spent in Rx
const double tD=86400;          // Seconds per day
const double t3=1.2;            // Time to recognize a false
    excitation, maximal
const double t4=0.1;            // Time to recognize a valid
    excitation, average
const double t5=60;             // Time under lf filed
const double t6=tD-t12*trc*VE-t34*N-t5*(LF_FE+LF_VE);   // Time
    at standby

const double a12=1.011;         // Average current in Tx state
const double a34=0.5;           // Average current in Rx state
const double a5 =0.15;          // Current in low field standby
const double a6 = 0.0024;       // Current in standby

const double trc = 40;          // Number of trans. per blink
const int D;                    // Day

// Protocol
module PT23
        x: [0..4] init 0;       // States
        e: [0..N] init 0;       // Events
        d: [0..D] init 0;       // Days

        // 0 Rx
        [] x=0 & (e=15) -> (x'=4) & (e'=0);
        [power] x=0 & (e<15) -> (VE/N):(x'=2)&(e'=e+1) + (2/N):(
            x'=3)& (e'=e+1) + (8/N):(x'=1) & (e'=e+1);

        // 1 Rx (false excitation)
```

```
        [] x=1 -> (x'=0);

        // 2 Tx
        [power] x=2 -> (LF_VE/VE):(x'=3)+((VE-LF_VE)/VE):(x'=0);

        // 3 SBLF
        [power] x=3 -> (x'=0);

        // 4 SB
        [power] x=4 & (d<D)-> (x'=0) & (d'=d+1) ;
        [] x=4 & (d=D) -> (d'=0) ;
endmodule

// Rewards assign to states
rewards "power"
        [power] x=0:  (a34)*(t34);             // Rx state
        [power] x=2:  (a12)*(t12)*(trc);       // Tx state
        [power] x=3:  (a5)*(t5);               // LF field SB
        [power] x=4:  (a6)*(t6);               // SB
endrewards
```