

O abordare orientată pe obiecte a memorării informațiilor pe tag-uri RFID

Remus Cătălin PRODAN, Cristina Elena TURCU, Marius Cristian CERLINĂ, Tudor Ioan CERLINĂ

Abstract—Tag-urile RFID reprezintă un mijloc automat de identificare și permit în același timp memorarea de informații. La ora actuală se estimează că tag-urile RFID vor înlocui treptat, într-un viitor nu prea depărtat codurile de bare, deoarece un tag RFID poate stoca informația corespunzătoare mai multor coduri de bare. Totuși, capacitatea de stocare disponibilă este prea mică pentru anumite aplicații. Se caută soluții pentru memorarea unei cantități cât mai mari de informații pe tag-uri RFID, dar, există o strânsă legătură între capacitatea de stocare și prețul tag-urilor: capacitate mare de stocare înseamnă cost ridicat pentru tag. În acest articol se prezintă o modalitate de optimizare a informațiilor stocate pe un tag RFID în cadrul unui sistem RFID generalizabil.

Index Terms—RFID, tag, clase, script, metode, constructor, destructor, tip

I. INTRODUCERE

TEHNOLOGIA RFID face parte, alături de codurile de bare, din categoria tehnologiilor de identificare automată (AutoID). Termenul de identificare automată, pe scurt automatic ID, descrie orice sistem dedicat identificării automate a unor entități. Scopul sistemelor auto-ID este să crească eficiența și să reducă erorile ce ar putea fi generate la introducerea manuală a datelor.

În categoria tehnologiilor auto-ID se încadrează mai multe tehnologii, dintre care se pot aminti, codurile de bare, card-

urile inteligente, recunoașterea vocală, tehnologii de biometrie, identificarea prin radio-frecvență (RFID) și altele. RFID reprezintă tehnologia de identificare a unei entități folosind undele radio. Această tehnologie poate fi utilizată pentru identificarea, urmărirea și localizarea unei game largi de entități. Astfel, RFID este considerată o tehnologie flexibilă, convenabilă, ușor de folosit și potrivită operațiilor automate. Aceasta oferă avantaje care nu se regăsesc la alte tehnologii de identificare.

Deși în practică tehnologia nu este complet nouă, majoritatea aplicațiilor sunt în faza de explorare. Interesul pentru RFID există în foarte multe sectoare ale vieții de zi cu zi, iar în Europa au început să fie făcute tot mai multe încercări pentru punerea în practică pe scară largă a unor aplicații care exploatează această tehnologie.

Un sistem de bază RFID este constituit din trei componente (Fig. 1):

- un transponder (tag RFID), programat electronic, care conține informații unice;
- un transmițător RFID, ce transmite semnale de identificare în radio frecvență;
- antenă sau bobină (folosită pentru transmiterea semnalelor radio dintre reader și dispozitivul RFID). Suplimentar acestui echipament de bază, sistemul RFID poate include și un software specific pentru aplicații.

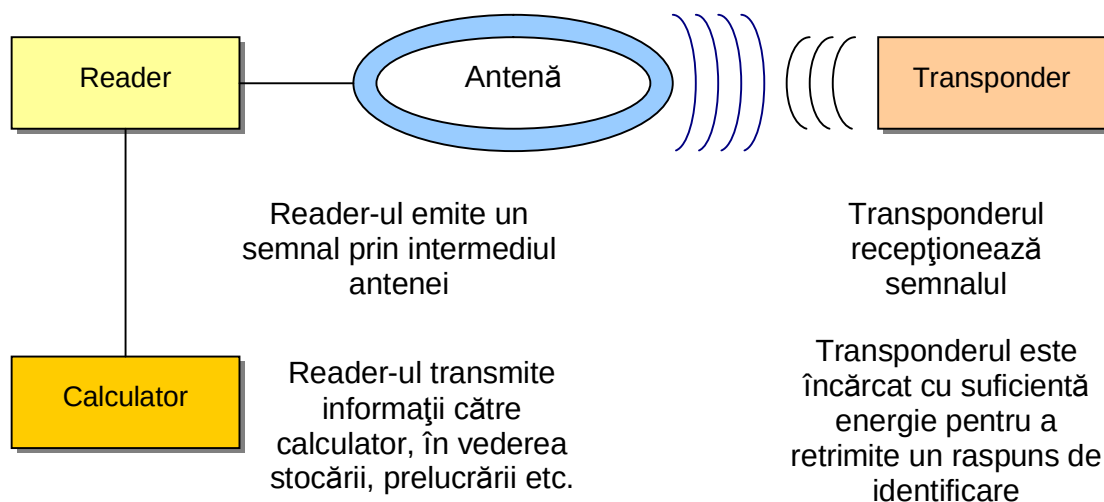


Fig. 1. Sistem de bază RFID.

Tag-ul sau transponder-ul RFID (nume derivat din termenii de transmițător și responder) reprezintă o etichetă compusă dintr-un microcip de memorie și o antenă (Fig. 2).

Se pot realiza diferite clasificări ale tag-urilor în funcție de frecvență, capacitate etc., fiecare tip fiind recomandat unor categorii de aplicații specifice [2, 3, 4].

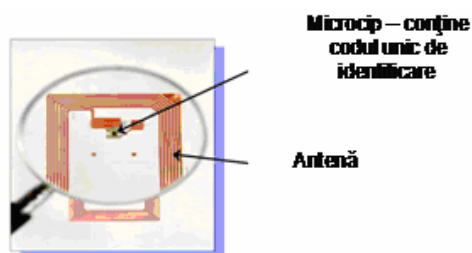


Fig. 2. Tag RFID.

Unul din cei mai importanți parametri în utilizarea tag-urilor este reprezentat de capacitatea memoriei disponibile. Astfel, există tag-uri cu capacitate foarte mică, de 1 bit, dar și tag-uri cu o capacitate de 2KB. Dar, capacitatea de memorare a tag-urilor diferă în funcție de tipul tag-ului și de standard [7]. Totuși, capacitatea de stocare disponibilă este prea mică pentru anumite aplicații. În același timp, prețul tag-urilor este în strânsă legătură cu tipul tag-ului și cu capacitatea de memorare a acestuia. Astfel, capacități mari de memorare determină costuri ridicate ale tag-urilor [8].

În aceste condiții, este necesară găsirea unor soluții care să permită memorarea mai multor informații în tag-uri de capacități mici și medii (care au un preț acceptabil).

Un prim obiectiv urmărit de echipa de cercetare l-a constituit organizarea datelor pe transponder din punct de vedere a tipului, utilizând în acest sens tipuri de date predefinite (la nivel de bit, octet sau chiar cu structuri specifice), cât și tipuri de date utilizator (clase). Un al doilea obiectiv este dat de introducerea unui comportament specific, local fiecărui transponder, obiectiv implementat prin introducerea metodelor (scripturi).

II. STRUCTURA DATELOR PE TRANSPONDERE

Autorii propun o soluție în care structurarea datelor să se realizeze la nivelul transponder, permițând definirea, pe lângă tipurile standard, a noi tipuri de date. Acest lucru s-a realizat prin definirea unor noi entități, denumită clasă, într-o secțiune distinctă a transponderului, zona header. Tot în această secțiune se pot identifica și scripturile asociate transponderului, alte informații legate de asignarea anumitor atribute pentru fiecare câmp în parte, precum și informații necesare securizării transponderului. În cadrul figurii următoare (Fig. 4) se poate observa structura unui astfel de transponder.

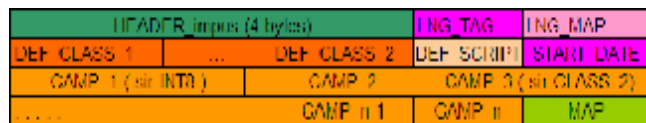


Fig. 3. Structura fizică a transponderului.

Structura logică a transponderului este prezentată în figura 4 unde se poate observa poziția în care sunt definite clasele, cât și zona în care sunt instanțiate acestea.

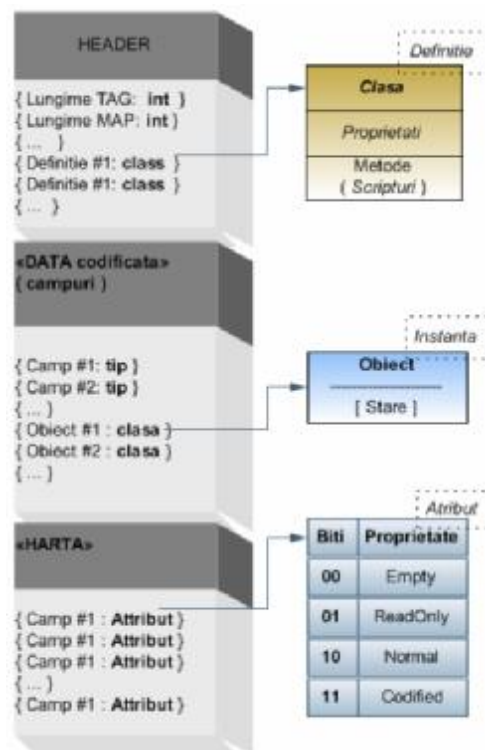


Fig. 4. Structura logică a transponderului.

III. TIPURI DE DATE ȘI CONSTRUCTIA UNOR NOI TIPURI

Deoarece majoritatea aplicațiilor bazate pe tehnologia RFID utilizează anumite tipuri de informații s-a putut identifica un set minimal pentru aceste tipuri de date. Tipurile de date predefinite sunt structurate la nivel de bit, câmpuri de biți, octet sau chiar structuri mai complexe, cum ar fi, tipul DATE_TIME (codificat pe 15 biți). Lista completă a acestor tipuri de date predefinită este dată în continuare:

1. bit (1 bits) – e.g., true/false sau yes/no;
2. int4 (4 bits) – valori pozitive între 0 și 15;
3. int8/char (8 bits) – valori ne-negative între 0 și 255/ASCII char;
4. int12 (12 bits) - valori ne-negative între 0 și 4095;
5. int16 (16 bits) - valori ne-negative între 0 și 65535;
6. int32/real (32 bits) – valori reale sau întregi reprezentate pe 32 biți;
7. dată și timp (26 bits) – valori de dată și timp între 1 ianuarie 1969, 0:00 și 31 decembrie 2031, 23:59;
8. date (15 bits) – valori de tip dată 1 ianuarie 1969 și 31 decembrie 2031.

Deoarece noțiunile de timp sau de dată calendaristică sunt utilizate destul de des în cadrul aplicațiilor RFID, s-a încercat definirea a două tipuri de date (date_time, respectiv date) în încercarea de a obține o lungime de codificare cât mai mică.

IF	GATE != CONSTANT	2
	FIELD_NO==CONSTANT	2
	FIELD_NO!=CONSTANT	2
	FIELD_NO_GIVEN_BY_CONSTANT == CONSTANT	2
	FIELD_NO_GIVEN_BY_CONSTANT != CONSTANT	2
EVENTS	SERVER_EVENT	2
EVENTI	INTERNAL_EVENT	2
STOP		1
SCRIPT	SCRIPT_NO	2
RETURN	-	2
CONSTRUCTOR	SCRIPT_NO	2
DESTRUCTOR	SCRIPT_NO	2
GOTO	LABEL	2
CALL	LABEL	2
	SCRIPT_NO	2
SETVAL_STRUCT	FIELD_NO0.FIELD_NO1. ... VAR	2+...
SETVAL_STRING	FIELD_NO [INDEX] VAR	4
GETVAL_STRUC T	VAR FIELD_NO0.FIELD_NO1. ...	3+...
GETVAL_STRING	VAR FIELD_NO [INDEX]	4
SETVAL	FIELD_NO, VAR	3
GETVAL	FIELD_NO, VAR	3
ADD	Variable CONSTANT	3
IF	VAR1 == VAR2	3
	VAR1 != VAR2	3
//	COMMENT	

Datorită limitărilor introduse de dimensiunea actuală a unui transponder se observă că sintaxa limbajului utilizată, nu este foarte bogată, dar suficientă pentru a implementa o logică internă minimală transponderului, oferind instrucțiuni de atribuire, structuri decizionale, construcții repetitive, apeluri de funcții, evenimente, etc.

În cadrul limbajului accesul la membri claselor (obiectelor) se va face pe bază de referință (adresa relativă a câmpului) pentru a simplifica logica internă a interpretorului/compilatorului. Orice câmp din structura unei clase va fi definit printr-o adresă față de începutul transponder-ului, considerată adresa 0. Adresarea se va face la nivel logic și nu la nivel de octet sau bit, în primul rând pentru faptul că în structura internă pot exista și câmpuri cu dimensiunea la nivel de bit sau octet. Fiecare din cele două metode de adresare au avantaje, dar și dezavantaje. Accesarea la nivel de câmp este posibilă pentru că este cunoscută structura internă a transponderului, putându-se identifica corect începutul fiecărui câmp. O accesare completă la nivel de bit ar duce la un consum exagerat de spațiu asociat memorării adreselor, iar o accesare la nivel de octet ar fi insuficientă pentru acele câmpuri definite pe un număr de biți ce nu este multiplu de 8.

După cum a fost specificat anterior, alături de datele membre ale unei clase, întâlnim și funcții/scripturi. Aceste scripturi sunt executate la trecerea prin dreptul unei porți, iar parte din acestea, constructorul și destructorul, în momentul creării structurii transponderului (*intrării în sistem*) sau al distrugerii conținutului (*ieșirii din sistem*). Conținutul oricărui script de pe un transponder va fi executat pe stația reader care l-a citit. Sintaxa limbajului permite accesarea în acel moment a diferite informații specifice stației reader, cum ar fi: adresa stației, tipul stației (stație de intrare, stație de ieșire, stație de asamblare, etc.), data curentă de pe stație, etc. Toate aceste informații pot fi utilizate în cadrul limbajului script. De

exemplu, pentru a completa automat data creării unui transponder se va utiliza data curentă preluată de la stația reader. Pentru a se realiza completarea automată a datei expedierii, atunci când transponderul trece pe la o stație având o anumită adresă sau fiind de un anumit tip (stație de ieșire), va fi preluată data curentă ca dată de expediere. Exemplul prezentat mai jos sub forma unui script, se rezumă la introducerea unui constructor în structura transponderului care să inițializeze câmpul denumit "Data creare" cu data curentă de la momentul introducerii transponderului în sistem. Inițializarea câmpului "Data creare" se va face în scriptul constructor utilizând valoarea dată de variabila sistem CURRENT_DATETIME, reprezentând timpul curent obținut de la stația reader în momentul trecerii transponder-ului prin dreptul său.

```

SCRIPT 0 //Script No. 0
#DEFINE BYTE INIT_GATE 0x01
#DEFINE BYTE INIT_GATE_OK 0x01
#DEFINE BYTE INIT_GATE_BAD 0x00
#DEFINE BYTE DATA_FIELD 0x03
IF( GATE == INIT_GATE ) //Data creare =
Data curenta
    SETVAL DATA_FIELD, CURRENT_DATETIME
IF( GATE == INIT_GATE ) // Init OK
    EVENTS( INIT_GATE_OK )
IF( GATE != INIT_GATE ) // Bad Init
    EVENTS( INIT_GATE_BAD)

STOP
CONSTRUCTOR 0 // Constructor - Script No. 0

```

V. AVANTAJELE ȘI DEZAVANTAJELE SOLUȚIEI

Structurând informația memorată în cadrul unui transponder într-o manieră orientată pe obiecte se pot obține mai multe avantaje, cum ar fi:

- interpretarea corectă a câmpurilor conform tipului de dată asociat (*definirea logică a câmpurilor*),

- micșorarea spațiului ocupat de către date (*definirea de clase și instanțierea de obiecte*),
- introducerea unui comportament local, la nivelul obiectelor memorate pe transponder, comportament obținut pe baza metodelor asociate claselor, metode memorate în structura internă a transponder-ului sau doar în șablonul asociat definit în baza de date de pe PC.

Soluția prezentată oferă o serie de avantaje după cum urmează:

- extindere facilă pentru alte seturi de tipuri de date, ce ar fi necesare în diferitele aplicații, funcție de evoluția pieței transponder-elor;
- pentru diferite aplicații practice poate fi folosit același sistem software, nefiind necesară modificarea structurii bazei de date;
- memorarea mai multor informații pe transpondere RFID, cu menținerea costurilor la un nivel scăzut.
- introducerea unui comportament specific fiecărui transponder pe baza codului disponibil în cadrul scripturilor componente, dacă respectivul transponder este folosit într-un sistem ce are implementat acest sistem.

Dezavantaje ale soluției propuse sunt:

- lipsa unui standard;
- dimensiunea spațiului ocupat pe transponder mai mare față de cazul aplicațiilor dedicate.

VI. PERSPECTIVE DE DEZVOLTARE

Pornind de la structurarea datelor pe baza machetelor orientate pe obiecte și analizând descrierea tipului de dată se va putea obține o anumită informație despre semnificația câmpului respectiv. Dacă pe lângă tipul transponderului mai sunt adăugate și informații auxiliare referitoare la clasificarea datei respective pe baza ontologiilor WEB (OWL), atunci există posibilitatea de a interpreta corect un câmp pe un anumit transponder, câmp a cărui semnificație nu este

cunoscută. Acest caz va fi întâlnit ori de câte ori în sistem ajunge un transponder ce utilizează același principiu de organizare a datelor. Aceste informații pot fi specificate utilizând ontologiile web. Legătura între structura câmpurilor pe transponder, în mod particular a unui anumit câmp, și descrierea pe baza ontologiilor web se va face prin referințe specifice în zona de definire a câmpurilor. În acest fel, un câmp va fi definit local prin tip și valoare, iar pe web printr-o referință către o anumită descriere, pe baza ontologiilor web.

VII. CONCLUZII

Soluția prezentată oferă utilizatorului posibilitatea structurării datelor pe transpondere RFID. În acest scop sunt puse la dispoziție tipuri predefinite concomitent cu oferirea posibilității de a defini noi tipuri de date specifice aplicației. În același timp există posibilitatea de a se defini secvențe script ce se execută automat atunci când transponderul trece prin dreptul unui reader conectat la o stație capabilă să interpreteze comenzile memorate pe transponder.

REFERINȚE BIBLIOGRAFICE

- [1] Examples of AIDC, <http://www.rfidgazette.org>, 2005.
- [2] Bhuptani Manish, Moradpour Shahram, RFID Field Guide: Deploying Radio Frequency Identification Systems, Prentice Hall, 2005.
- [3] Himanshu Bhatt, Bill Glover (2006), RFID Essentials, O'Reilly
- [4] Sandip Lahiri (2005), RFID Sourcebook, Prentice Hall PTR, 2005.
- [5] Klaus Finkenzeller (2003), RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition, John Wiley & Sons.
- [6] IBM (2003), Smart tags: RFID becomes the new bar code, Executive technology report, ibm.com/bcs.
- [7] David Dressen, Considerations for RFID Technology Selection, Atmel Applications Journal, http://www.atmel.com/dyn/resources/Prod_documents/secrerf_3_04.pdf
- [8] What is Radio Frequency Identification (RFID)?, <http://www.peacocks.com.au/about-rfid.htm>
- [9] Cristina Turcu, Valentin Popa, Vasile Găitan ș.a. Integrarea tehnologiilor RFID în aplicații B2B pentru rețelele de aprovizionare ale întreprinderilor, „Sisteme distribuite”, ISSN/ISBN: 1842 – 6808, 2006.