Università degli Studi di Camerino

SCUOLA DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica (Classe L-31)

Gestione di RFID Reader con Arduino: case study Enjoy Unicam Card

Laureando
Alessio Bianchetti
Matricola 080174

Relatore
Prof. Fausto Marcantoni

A.A. 2011/2012

"It is only in the mysterious equations of love that any logical reasons can be found."

John Forbes Nash jr. - "A beautiful mind"

INDICE

CAPITOLO I 5

1.1 ARDUINO 5
1.2 SCELTA DELLA PIATTAFORMA 6
1.3 ARDUINO UNO 9
1.3.2 Memoria 12
1.3.3 Alimentazione 12
1.3.4 Connettori digitali 13
1.3.5 Connettori Analogici 14
1.4 ETHERNET SHIELD 15
1.4.1 Indicatori 16
1.5 ARDUINO IDE 16
1.5.1 Visual Micro Plug-in 18
1.5.2 Programmazione 18

CAPITOLO II 19

2.1 TECNOLOGIA RFID 19
2.2 FUNZIONAMENTO 20
2.3 GERARCHIA DI CLASSI RFID 21
2.4 SCELTA DELLA RADIO FREQUENZA 23
2.5 SMART CARD 24
2.5.1 Tipologie di smart card 25
2.5.2 Protocollo APDU 27
2.5.3 Ciclo di vita di una smart card 28
2.5.4 File system 28
2.5.5 Standard ISO 30
2.5.6 Calypso Standard 32
2.5.7 Altri Standard 34
2.5.8 Java Card 36
2.6 TECNOLOGIE E CARTE IN COMMERCIO 37
2.6.1 Mifare Technologies 38
2.6.2 Emulazione Mifare 46
2.6.3 Infineon Technologies 47
2.6.4 Legic Technologies 49
2.6.5 EM Microelectronic 50
2.6.6 ST Microelectronics 52
2.6.7 Atmel Corporation 53
2.6.8 HID Global 53
2.6.9 Giesecke & Devrient 55

CAPITOLO III 57

3.1 IMPLEMENTAZIONE 57
3.2 SOFTWARE E HARDWARE 57
3.3 CARTE UTILIZZATE 58
3.3.1 Carta Raffaello 58
3.3.2 Carta Enjoy my UniCam 58
3.3.3 Carta OnCard 59
3.3.4 Carta Oyster 60
3.4 LETTORI RFID UTILIZZATI: 61
3.4.1 HID Omnikey 5321 CL USB 61
3.4.2 ACS ACR120U-TK-R 61
3.4.3 SCM Microsystems SCL010 61

3.5 SCELTA DI ARDUINO 62
3.6 SCELTE PROGETTUALI 62
3.7 APPLICATIVO UTENTE 63
3.8 PROGRAMMA ARDUINO 70
SVILUPPI FUTURI 74
CONCLUSIONI 76
RINGRAZIAMENTI 77
SITOGRAFIA 78
INDICE DELLE FIGURE 81
INDICE DELLE TABELLE 81

Introduzione

Arduino e smart card, due idee rivoluzionarie.

Il connubio tra informatica ed elettronica è quella connessione che lega l'algoritmo a qualcosa che va oltre quello che è possibile vedere solamente attraverso un display. È un pensiero, una logica applicata, che prende forma, si muove ed interagisce con l'individuo a 360 gradi. È l'estensione di un ragionamento codificato in righe codice verso l'esterno, in quel luogo in cui l'immediatezza, e la realtà delle cose, stupisce anche i soggetti estranei a questo mondo. Questo legame non è semplice, o quanto meno comodo, da raggiungere. Ma alcune grandi idee nascono per essere al servizio dello scolaro, di quella persona che nutre interesse, ripone impegno ed accresce la sua conoscenza.

Arduino ha rivoluzionato il camminamento tra informatica ed elettronica, concedendo la possibilità di incapsulare gli algoritmi in oggetti, e realizzare un numero infinito di progetti in modo semplice e diretto. È una piattaforma in grado di accelerare drasticamente l'apprendimento dei concetti fondamentali di entrambe le discipline, e soprattutto è in grado di fonderle insieme.

Nella prima parte di questo lavoro di tesi saranno spiegati i concetti base di Arduino, come è nato e cosa (non) è possibile realizzare con questa piattaforma. Saranno approfonditi i dettagli di alcune schede utilizzate durante il progetto svolto incluso nella tesi, e inoltre sarà anche indicato un possibile ragionamento per la scelta di una delle tante schede Arduino presenti sul mercato.

La seconda parte tratta la tecnologia RFID, un'altra grande idea rivoluzionaria. In un passato non troppo remoto, due oggetti per interagire tra loro dovevano necessariamente, almeno nell'idea comune, entrare in contatto. Prima della rivoluzione tecnologica, era impossibile pensare che diverse macchine potessero scambiare informazioni attraverso l'etere, e in effetti, come può un dato, quindi un immagine, un file di testo o addirittura un filmato, essere trasmesso ad un altro dispositivo attraverso l'aria? La risposta è nelle onde radio. La radiofrequenza è una forma di radiazione elettromagnetica in grado trasmettere dati e quindi condividere e gestire con tempestività le informazioni. Questo ha reso possibile una globalizzazione più o meno immediata dell'individuo. La comunicazione dati a radiofrequenza può essere applicata in moltissimi domini, basti pensare ad Internet e a tutte le soluzioni implementate attraverso esso. Un googol di conoscenza. Questa tecnologia è

impiegata anche nell'identificazione dei soggetti attraverso dispositivi chiamati smart card. Una semplice tessera, delle fattezze di una carta di credito, con all'interno un microprocessore, o comunque una memoria dati. Con la possibilità di assegnare un numero univoco per ogni smart card, e l'integrazione di algoritmi di crittografia sofisticati, è possibile realizzare un carta intelligente in grado di servire molteplici applicazioni. Nella seconda parte della tesi verrà quindi introdotta questa tecnologia, spiegandone il funzionamento, i diversi standard internazionali e mostrando un ventaglio di prodotti dai maggiori produttori di smart card.

La terza parte è quindi l'implementazione degli argomenti precedenti. Lo scopo del progetto è quello, in primo luogo, di applicare le conoscenze acquisite per quanto riguarda Arduino e le smart card RFID, ed in secondo luogo di sviluppare un applicativo. Questo applicativo sarà in grado di gestire un database di smart card, lettori ed eventi associati. Gli eventi saranno comunicati ad Arduino che svolgerà l'azione programmata. Verranno inoltre elencate le smart card ed i lettori utilizzati per il progetto implementato.

L'ultima parte tratta le conclusioni e un approfondito studio sui possibili sviluppi futuri

Capitolo I

1.1 Arduino

Arduino è un progetto sviluppato presso l'*Interaction Design Institute Ivrea* (IDII). Nasce nel 2005 dall'idea di un professore universitario, l'ingegnere elettronico Massimo Banzi, che decise di creare una piattaforma insieme al collaboratore David Cuartielles, per i propri studenti così da accelerare l'apprendimento di tutti i concetti fondamentali dell'elettronica e dell'informatica. Arduino è una piattaforma hardware open source distribuito nei termini della licenza *Creative Commons Attribution-ShareAlike 2.5* [1]. È definito hardware open source in quanto è possibile reperire tutti gli schemi circuitali delle schede ed i componenti che ne fanno parte così da poter essere realizzate anche da soli. Da questo deriva anche la filosofia che presenta Arduino chiamata DIY, acronimo per *Do It Yourself*. Il concetto di open fa sì che le schede, grazie alla grande comunità di sviluppatori che seguono il progetto, vengano costantemente migliorate sia per quanto riguarda l'hardware che il software, data la grande quantità di librerie che si trovano in rete. Quello appena descritto è uno dei punti di forza di Arduino poiché, coinvolgendo le discipline di informatica ed elettronica, è ben noto quanto il concetto di *openness* contribuisca alla crescita di un progetto. Inoltre questo aspetto è molto importante anche in un contesto didattico perché permette agli studenti di apprendere e studiare una materia dalle fondamenta.

Arduino può essere utilizzato per una grande varietà di progetti, molto spesso per ciò che riguarda la fase di prototipazione in quanto le schede proposte sono costruite per facilitare una rapida implementazione circuitale di un progetto. È basato su un semplice microcontrollore[1] programmabile attraverso l'ambiente di sviluppo integrato scaricabile gratuitamente sul sito ufficiale [2]. Con una scheda Arduino opportunamente interfacciata, è possibile acquisire diversi valori in input, come la temperatura, la pressione di un pulsante, le coordinate geografiche, e successivamente comandare dei valori di output di diversa natura, come uno display, un motore, una luce o un altoparlante. Tutto dipende da cosa vogliamo collegare ad una scheda Arduino. La piattaforma può lavorare in modalità stand-alone oppure può comunicare con altri ambienti, come un programma che risiede in un computer o altri dispositivi esterni.

[1] Alcune versioni recenti utilizzano un vero e proprio microprocessore.

Il linguaggio di programmazione utilizzato nell'IDE Arduino è basato su *Wiring*, pensato per un utilizzo più semplice rispetto ai normali linguaggi per microcontrollori [3]. L'ambiente di sviluppo integrato è invece basato su *Processing*, con la disponibilità di utilizzare la libreria residente messa a disposizione degli sviluppatori [4]. Entrambi, l'IDE e la libreria residente, sono concessi in uso secondo i termini legali di una licenza libera *GPLv2* [5].

Ovviamente esistono molte schede con microcontrollore in grado di svolgere le operazioni di una piattaforma Arduino, ma ci sono diversi vantaggi che derivano dall'utilizzo di quest'ultima [6].

Il basso costo è sicuramente uno dei fattori che ha reso la piattaforma un progetto di dimensione mondiale. Gli acquirenti vanno da semplici hobbisti, a tecnici professionisti che non solo possono coltivare ed approfondire la propria passione per queste tipologie di prodotti, ma anche affacciarsi in un contesto produttivo tramite l'utilizzo di hardware Arduino. Un altro settore che ha giovato di questi prodotti è sicuramente la didattica, dove con un costo ridotto è possibile fornire un laboratorio di strumenti utili.

L'ambiente di sviluppo integrato è un altro fattore chiave, poiché semplice ed intuitivo. Inoltre è multipiattaforma (come spesso non avviene con altri ambienti di sviluppo per microcontrollori programmabili) in modo tale da poter sviluppare senza restrizioni sulla scelta del sistema operativo, che può essere *Microsoft™ Windows®*, *Apple® Mac OsX®* e *Linux®*. Semplice da usare, l'ambiente di sviluppo è perfetto per un utente alle prime armi, ma abbastanza flessibile anche per utenti avanzati. Questo perché l'interfaccia non confonde l'utilizzo, ma mette comunque a disposizione funzioni avanzate per i programmatori esperti.

Come detto in precedenza, Arduino è un progetto open source completo, disponibile per essere migliorato da programmatori esperti. Il linguaggio può essere esteso tramite librerie scritte in C++, ed è possibile anche scrivere direttamente in linguaggio *AVR*, ovvero l'ambiente di programmazione C per i chip *Atmel* (la maggior parte del software di Arduino è scritto in questo linguaggio) [7].

1.2 Scelta della piattaforma

Esistono molte tipologie di schede Arduino, sia ufficiali che sviluppate da terze parti. Per poter effettuare la giusta scelta è necessario per prima cosa analizzare quanto più nel

dettaglio lo scopo del progetto che si vuole realizzare con hardware Arduino. Le principali caratteristiche che differenziano tra loro le diverse schede sono la dimensione della memoria, la velocità del *clock*, la quantità di ingressi e uscite e i tipi di connettività.

La memoria è un punto chiave per ogni scheda con microcontrollore programmabile in quanto, se di dimensioni adeguate, ci consente di non avere severe limitazioni sulla lunghezza del codice e sulla quantità di variabili instanziabili dal programma residente. Ci sono tre pool di memoria in un microcontrollore sulle attuali schede Arduino [8]:

- Memoria Flash: contiene il codice programmato per Arduino, chiamato in termini tecnici *sketch*.
- SRAM (*Static Random Access Memory*): è uno spazio di memoria volatile dedicato alla creazione e manipolazione delle variabili durante l'esecuzione del programma.
- EEPROM (*Electrically Erasable Programmable Read-Only Memory*): è uno spazio di memoria persistente per contenere variabili, i cui valori sono mantenuti anche se viene spento Arduino.

Ci sono quindi due aree di memoria non volatili e una volatile. Da notare che per alcune schede, come Arduino UNO, lo quantità di memoria per la SRAM è di 2048 byte. Questo significa che bisogna operare una buona scelta per l'uso delle variabili; è molto semplice terminare lo spazio a disposizione, e quindi commettere un overflow di memoria, soprattutto con l'utilizzo eccessivo di stringhe. Altre schede ovviamente dispongono di una quantità di memoria superiore, come mostrato in tabella 1.1.

	UNO	Leonardo	DUE	Mega2560	Nano
Flash memory	32 kb	32 kb	512 kb	256 kb	16 kb
SRAM	2 kb	2,5 kb	96 kb	8 kb	1 kb
EEPROM	1 kb	1 kb	-	4 kb	0,5 kb

Tabella 1.1 Quantità di memoria schede Arduino.

Se si verifica un overflow di memoria SRAM il nostro programma caricato probabilmente non funzionerà bene o non funzionerà del tutto. Ci sono diversi metodi per sopperire a questo problema.

Se lo sketch comunica con un programma residente in un pc desktop o un laptop o un altro dispositivo di complessità simile, si può provare a spostare i dati o parte dei calcoli sul computer collegato, riducendo così il carico su Arduino.
Se si utilizzano tabelle di *Look-Up* (LUT) o comunque array di grandi dimensioni è necessario, quando possibile, utilizzare la più piccola struttura dati per memorizzare i valori.
Se si utilizzano stringhe statiche, che non hanno la necessità di essere modificate durante l'esecuzione del programma, è possibile memorizzare questi valori nella memoria flash invece che nella SRAM, utilizzando la parola chiave PROGMEM [9].
Si consideri che generalmente la memoria disponibile di una scheda Arduino utilizzata dal microcontrollore è sufficiente per la maggior parte delle soluzioni.

Un altro fattore da tenere in considerazione è la velocità del clock. Il clock è un segnale periodico utilizzato per sincronizzare il funzionamento dei dispositivi elettronici digitali, e quindi di Arduino. Con un clock elevato si ottengono maggiori prestazioni in termini di velocità di esecuzione.

La quantità di porte in ingresso e uscita disponibili sulla piattaforma determinano la scelta di una scheda rispetto ad un'altra. Generalmente il numero in questione è direttamente proporzionale alla grandezza fisica della scheda. I tipi principali di pin sono due, digitali ed analogici, e possono essere configurati sia in ingresso che in uscita; è importante notare che generalmente un pin analogico può essere configurato e utilizzato come fosse un pin digitale [10]. Attraverso l'utilizzo delle porte a disposizione, in numero diverso a seconda della scheda scelta, possiamo controllare tutto ciò che è esterno all'hardware Arduino. Gli esempi sono innumerevoli e vanno dalla lettura di sensori ambientali in un contesto di domotica e quindi, per esempio, la regolazione della temperatura di una stanza, all'impiego in una stazione metereologica per la lettura dei dati atmosferici, o anche il controllo di un robot pilotato con un joystick.

Per poter permettere ad Arduino di comunicare con più efficienza con ciò che lo circonda è necessario dotarlo di componenti aggiuntive, spesso inserite in appositi *Shield*, ovvero delle schede che si uniscono ad una scheda base Arduino per aggiungere funzionalità mancanti. Esistono Shield di tutti i tipi, i più utilizzati sono quelli che aggiungono interfacce

come ethernet, wireless, GPS, Xbee, controllo di motori, lettore di schede di memoria, lettore RFID, generatori di clock real-time, pannelli TFT, e connettività USB [11].

Nella tabella 1.2 è possibile osservare le differenti caratteristiche principali di alcune delle schede Arduino più utilizzate in commercio. È possibile notare che le maggiori diversità sono in termini di entrate e uscite, memoria e potenza di calcolo.

	Uno	Leonardo	Due	Mega 2560
Microcontrollore	ATmega328	ATmega32u4	AT91SAM3X8E	ATmega2560
Tensione di lavoro	5 V	5 V	3.3 V	5 V
Tensione di alimentazione	7 – 12 V consigliata	7 – 12 V consigliata	7 – 12 V consigliata	7 – 12 V consigliata
Pin digitali I/O	14 (di cui 6 PWM)	20 (di cui 7 PWM)	54 (di cui 12 PWM)	54 (di cui 15 PWM)
Pin analogici	6	12	2 (DAC)	16
Corrente dc per pin I/O	40 mA	40 mA	130 mA	40 mA
Corrente dc per pin da 3.3v	50 mA	50 mA	800 mA	50 mA
Memoria flash	32 KB (di cui 0.5 KB per il bootloader)	32 KB (di cui 4 KB per il bootloader)	512 KB (tutti per l'utente)	256 KB (di cui 8 KB per il bootloader)
SRAM	2 KB	2.5 KB	96 KB (64 KB + 32 KB)	8 KB
EEPROM	1 KB	1 KB	-	4 KB
Velocità clock	16 MHz	16 MHz	84 MHz	16 MHz

Tabella 1.2 Caratteristiche principali delle schede Arduino.

1.3 Arduino UNO

L'Arduino UNO è sicuramente una delle schede più diffuse e utilizzate in tutto il mondo. E' una scheda con microcontrollore ATmega328 [12]. Dispone di 14 pin digitali di ingresso/uscita, sei dei quali possono essere usati come output PWM[2] (*Pulse-width modulation*), sei ingressi analogici, un risonatore ceramico[3] da 16 MHz, un connettore USB, un jack di alimentazione, un *header* ICSP[4], ed un pulsante di reset. Contiene tutto il

[2] Modulazione di larghezza di impulso, utile per simulare un comportamento analogico con valori digitali.
[3] Il risonatore ceramico è meno preciso di un oscillatore al quarzo ma i costi sono nettamente ridotti.
[4] Acronimo di In-Circuit Serial Programming, utilizzato per scrivere il software nel microcontrollore con programmatori esterni.

necessario per lavorare con il microcontrollore. Una volta collegato ad un computer, ed installato l'ambiente di sviluppo, è possibile già da subito cominciare a programmare la scheda [13].

Di seguito vengono descritti con maggior dettaglio alcune parti della scheda e nella figura 1.1 è possibile vedere il fronte ed il retro di Arduino UNO.

Figura 1.1 Fronte e retro scheda Arduino UNO Rev. 3

1.3.1 Microcontrollore

Il microcontrollore dell'Arduino UNO R3 è l'ATmega328, un microcontrollore ad alte prestazioni con bus a 8 bit prodotto dalla Atmel con architettura di tipo RISC (*Reduced Instruction Set Computer*) le quali caratteristiche possono essere così elencate [14]:

- Memoria di programma Flash da 32 kb (*In-System Programming).*
- Memoria EEPROM da 1 kb.
- Memoria SRAM da 2 kb.
- Mantenimento dati 20 anni a 85°C, 100 anni a 25°C.
- Prestazioni di 20 MIPS a 20 MHz.
- 23 linee programmabili di I/O.
- 32 x 8 registri di lavoro general purpose.
- Due timer a 8 bit e un timer a 16 bit.
- Sorgenti interne ed esterne per interrupt.
- Seriale USART[5] programmabile.
- Interfaccia seriale a 2 fili compatibile Philips I²C (*Inter Integrated Circuit*).
- Porta SPI seriale.
- Sei convertitori A/D con risoluzione di 10 bit.
- Timer *watchdog* programmabile con oscillatore interno.
- Sei modalità di risparmio energetico.
- Tensione di funzionamento compresa tra 1.8 e 5.5 volts.
- Temperatura di lavoro da -40°C a 85°C.
- Basso consumo di corrente.

[5] Universal Synchronous/Asynchronous Receiver/Transmitter, utilizzata per collegare Arduino via USB al PC.

La memoria di questo microcontrollore è limitata ad un utilizzo consapevole delle variabili come spiegato in precedenza. Le porte *general purpose* sono collegate sulla scheda in maniera da essere utilizzate come porte digitali o analogiche, che a loro volta potrebbero servire come logica di controllo per diverse funzionalità, come quelle aggiunte da uno Shield. I timer, anche detti contatori, sono componenti hardware preinstallate sulla scheda, e possono essere utilizzati per misurare i tempi di un evento. Arduino UNO ha tre timer interni chiamati *timer0*, *timer1*, e *timer2*. Il timer1 ha una risoluzione di 16 bit il che significa che è possibile contare un numero di valori pari a 65536, mentre gli altri due timer hanno una risoluzione di 8 bit e quindi 256 possibili valori. Altre schede, come l'Arduino *Mega2560* hanno a disposizione sei timer, di cui quattro a 16 bit. La possibilità di interrupt permette di spezzare il normale flusso di istruzioni sequenziali a seguito di un evento prestabilito. La procedura si chiama ISR, acronimo per *Interrupt Service Routine*. Una volta che le istruzioni dedicate all'interrupt sono terminate, il programma continua nel punto esatto in cui era stato interrotto. La porta seriale SPI, acronimo per *Serial Peripheral Interface* è utile per la comunicazione tra il microcontrollore e altre periferiche vicine ad Arduino, spesso utilizzata per far comunicare due microcontrollori. Il timer *watchdog* è utilizzato per controllare che Arduino non cada in blocco o cicli infiniti, dove in questi casi è utile, se non indispensabile, resettare la scheda [15].

In figura 1.2 è possibile osservare la piedinatura del microcontrollore ATmega328 installato sull'Arduino UNO.

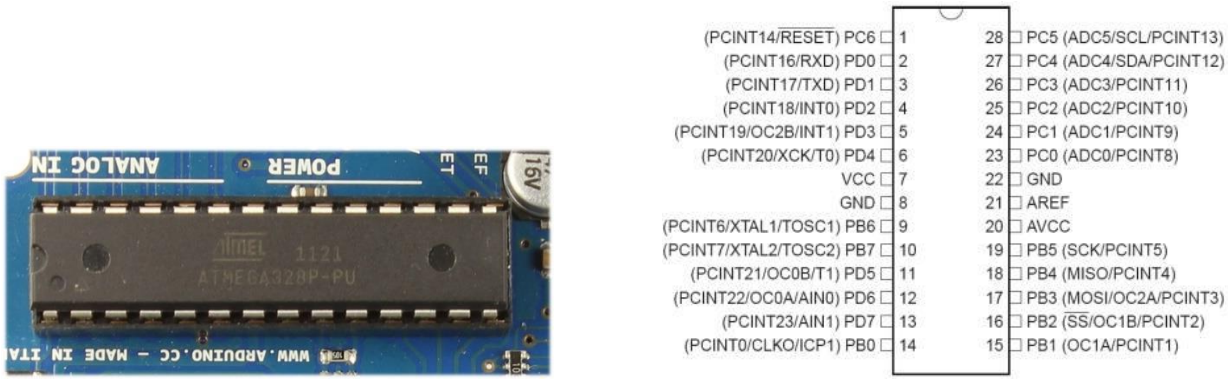

Figura 1.2 Microcontrollore ATmega328 e piedinatura.

1.3.2 Memoria

Il microcontrollore ATmega328 dispone di 32 kb di memoria di programma della quale 0,5 kb sono usati per il bootloader, mostrata in figura 1.3. Inoltre dispone di 2 kb di SRAM e 1 kb di EEPROM che possiamo leggere o scrivere utilizzando la libreria EEPROM presente nell'IDE di Arduino.

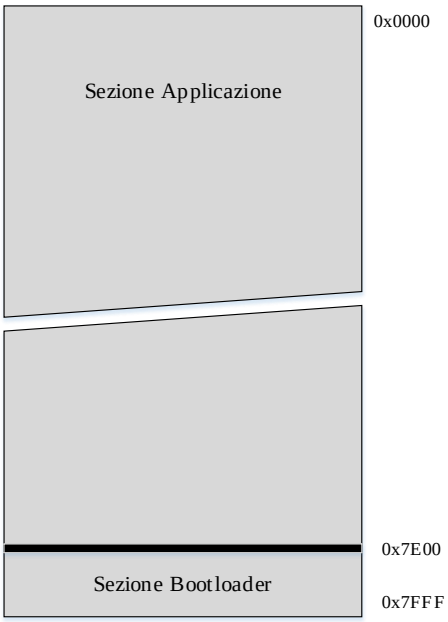

Figura 1.3 Memoria di programma (Flash memory).

1.3.3 Alimentazione

L'Arduino UNO può essere alimentata via USB, o da un'alimentazione esterna. La sorgente di alimentazione viene selezionata automaticamente dalla scheda.
L'alimentazione esterna può provenire sia da un adattatore AC/DC che da una batteria. L'adattatore può essere collegato ad Arduino tramite un jack da 2.1mm. La batteria può essere collegata ai pin *GND* e *VIN* sul connettore di alimentazione della scheda. La tensione di alimentazione esterna deve essere compresa tra i 6 ed i 20 volts, anche se le tensioni raccomandate sono dai 7 ai 12 volts [13].

I pin di alimentazione, visibili in figura 1.4, sono disponibili tramite il connettore POWER e sono i seguenti:

- **VIN**: restituisce la tensione applicata dall'alimentatore al jack e può essere usato per alimentare altri circuiti che dispongano già di un regolatore di tensione (ad esempio gli Shield applicati al modulo).

- **5V**: fornisce i 5 volt prelevati dall'uscita del regolatore interno ed è utile per alimentare altri circuiti compatibili con i 5 volt.
- **3.3V**: fornisce i 3,3 volt ricavati dal regolatore corrispondente e consente di alimentare altri circuiti compatibili con tensioni di 3,3 volt (la massima corrente prelevabile è di 150 mA).
- **GND**: è il contatto di massa (*Ground*).
- **RESET**: portando questa linea a livello basso permette di resettare il microcontrollore. Solitamente sugli Shield montati sulla scheda viene replicato questo funzionamento attraverso un tasto.
- **IOREF**: consente agli Shield di adattarsi alla tensione fornita dalla scheda.

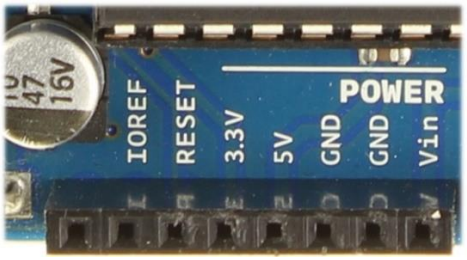

Figura 1.4 Connettore POWER.

1.3.4 Connettori digitali

Ciascuno dei 14 pin digitali presenti sulla Arduino UNO, mostrati in figura 1.5, possono essere configurati in modalità input oppure output. La tensione di uscita di ogni output è di 5 volt, ed ogni pin è in grado di lavorare con un massimo di 40 mA. Ogni pin è dotato di una resistenza pull-up[6] del valore di 20-50 kOhm. Alcuni pin hanno anche comportamenti specifici:

- **Pin 0 (RX) e 1 (TX)**: possono essere utilizzati per la comunicazione seriale.
- **Pin 2 e 3**: possono essere configurati come trigger per eventi esterni, come ad esempio il rilevamento di un fronte di salita o di discesa di un segnale in ingresso.

[6] Le resistenze di pull-up e pull-down sono usate nei circuiti elettronici per forzare un determinato stato logico ad un valore, per eliminare fluttuazioni di corrente o ancora per evitare cortocircuiti quando si usano i pulsanti.

- **Pin 3, 5, 6, 9, 10 e 11**: possono essere configurati via software per generare segnali PWM con risoluzione di 8 bit. Tramite un filtro RC è possibile ottenere tensioni continue di valore variabile.
- **Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)**: possono essere programmati per realizzare una comunicazione SPI.
- **Pin 13**: è connesso a un LED interno alla scheda, risulta utile in situazioni di debug.
- **GND**: è il contatto di massa (*Ground*).
- **AREF**: Tensione di riferimento per gli ingressi analogici.

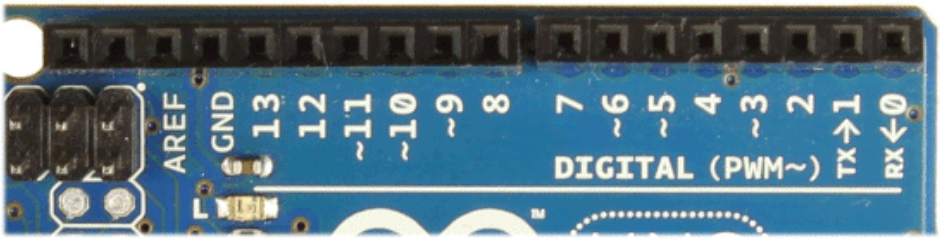

Figura 1.5 Connettori digitali.

1.3.5 Connettori Analogici

La UNO ha sei ingressi analogici etichettati da A0 ad A5, ognuno dei quali fornisce 10 bit di risoluzione e quindi 1024 valori differenti. Per impostazione predefinita possono misurare una tensione di 5 volt riferita a massa, anche se è possibile cambiare l'estremità superiore del loro intervallo utilizzando il pin AREF e la funzione *analogReference()*. Alcuni hanno comportamenti specifici:

- **Pin A4 (SDA) e A5 (SCL)**: permettono di realizzare una comunicazione nello standard I²C a due fili.

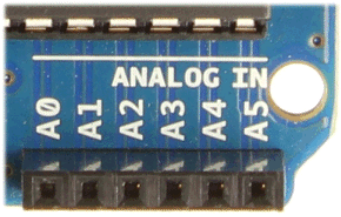

Figura 1.6 Connettori analogici.

1.4 Ethernet Shield

L'Arduino Ethernet Shield permette ad una scheda Arduino di connettersi ad una rete ad una velocità massima di 100 Mbps, e quindi anche ad Internet. È basata sul chip Ethernet Wiznet W5100 con il quale supporta i protocolli TCP, UDP, ICMP, IPv4, ARP, IGMP, e PPPoE [16]. Questo Shield supporta fino ad un massimo di quattro connessioni simultanee. Ha un buffer interno di 16 kb per la trasmissione e la ricezione dei dati. Per utilizzare le funzioni messe a disposizione in uno sketch bisogna includere la libreria *Ethernet* o librerie di terze parti compatibili, e connettere lo Shield su una rete tramite un cavo ethernet UTP con connettore RJ-45.

L'ultima versione di questo Shield, la Revision 3 mostrata in figura 1.7, dispone anche di un slot SD card che può essere usato per salvare i dati prelevati dalla rete, o per qualsiasi altro scopo che necessita di memorizzare dati su periferiche di massa non volatili. Per utilizzare le funzioni di scrittura e lettura è necessario utilizzare la libreria SD, o librerie di terze parti compatibili.

Questo Shield dispone anche di un tasto di reset direttamente connesso con il medesimo tasto situato sulla scheda base ospitante.

Da notare che sia la comunicazione ethernet che quella via SD card avviene con lo stesso pin SPI, il quale dovrà essere utilizzato in mutua esclusione per l'ethernet o per l'SD [17].

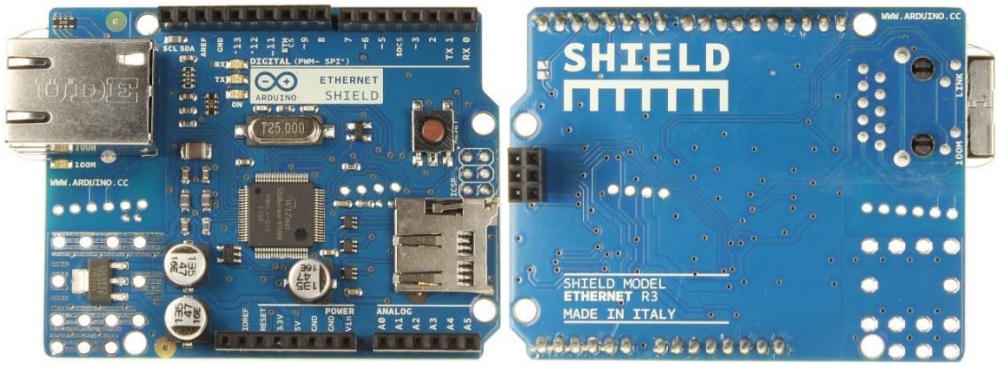

Figura 1.7 Fronte e retro Ethernet Shield Rev.3

1.4.1 Indicatori

Sulla scheda ci sono anche diversi indicatori led come mostrato in figura 1.8:

- **RX**: indica che la scheda sta ricevendo dati via Ethernet.
- **TX**: indica che la scheda sta trasmettendo dati via Ethernet.
- **100M**: indica che è disponibile la connessione a 100 Mbps, altrimenti la connessione è di 10 Mbps.
- **LINK**: indica che la scheda è collegata via Ethernet su una rete.
- **COLL**: indica che si sono verificati errori durante la comunicazione di rete.
- **ON**: replica lo stesso led della scheda ospitante e indica lo stato di alimentazione della scheda.

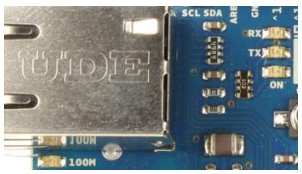

Figura 1.8 Indicatori LED presenti sull'Ethernet Shield.

1.5 Arduino IDE

L'ambiente di sviluppo integrato per Arduino è scaricabile gratuitamente dal sito ufficiale. L'editor si presenta in veste molto minimalista, infatti abbiamo a disposizione solamente un editor di testo in cui scrivere il codice, un area in cui saranno visualizzati gli eventuali errori a tempo di compilazione, la possibilità di aprire un monitor seriale con l'Arduino connesso via USB al computer, e poche altre finestre.

Per scrivere un programma, che in gergo si chiama *sketch*, si utilizza l'editor di testo messo a disposizione dall'IDE. L'estensione di un file di codice Arduino è *".ino"*. Gli unici controlli di testo che abbiamo a diposizione sono i classici copia/incolla, cerca/sostituisci. L'area dei messaggi sarà utile solo a tempo di compilazione in quanto se commettiamo un errore di sintassi o semantica mentre scriviamo il codice non saremo in nessun modo avvisati, fintanto che non viene compilato il codice. Arduino mette a disposizione una serie di librerie di base con le quali è possibile sviluppare dei programmi completi. Inoltre sono

disponibili anche diversi esempi di progetto. Tutti i costrutti, le strutture, le variabili, e le funzioni di base e le librerie standard, sono consultabili nel sito ufficiale, nell'apposita pagina di riferimento [18].

Nella parte alta dell'editor abbiamo a disposizione una veloce toolbar mostrata in figura 1.9.

Figura 1.9 Quick toolbar Arduino IDE.

In ordine, i tasti ci permettono di:

- Verificare la presenza di errori nel codice, e quindi compilare.
- Effettuare l'upload[7] del codice sulla scheda Arduino.
- Creare un nuovo sketch.
- Aprire uno sketch già esistente.
- Salvare lo sketch corrente.
- Aprire il monitor seriale con Arduino.

Se il nostro Arduino è correttamente collegato al nostro pc, nella parte bassa dell'editor sarà notificato in quale porta USB è stato collegato.

Tra le varie opzioni dell'editor è possibile importare le librerie di terze parti che risultano molto spesso utili. Come in molti altri linguaggi di programmazione sarà necessario includere l'header di libreria all'interno del codice.

Per programmare una scheda Arduino, e quindi utilizzare l'IDE, è necessario installare i driver, reperibili nel sito ufficiale. Una volta installati, basterà riavviare l'IDE.

Attualmente Windows 8 non è supportato per l'installazione dei driver, e bisogna seguire la procedura di seguito descritta [19]:

1. Premere il tasto Windows + R per aprire la finestra esegui.
2. Inserire shutdown.exe /r /o /f /t 00[8].
3. Premere il tasto OK.
4. Il sistema si riavvierà e sarà proposta una schermata di scelta.
5. Selezionare Risoluzione dei Problemi.
6. Selezionare Impostazione Avanzate.
7. Selezionare Impostazioni di Avvio Windows.
8. Premere il tasto Riavvia.
9. Il sistema si riavvierà con le opzioni di avvio avanzate.
10. Selezionare Disabilita Imposizione Firma Digitale.

Una volta riavviato il sistema, installate i driver come fareste per Windows 7 [20].

[7] L'upload del codice avviene attraverso una porta COM, ovvero una porta USB selezionabile nelle opzioni dell'editor.

[8] In sequenza i parametri significano: riavvio, opzioni avanzate di avvio, forza la chiusura delle applicazioni, esegui adesso.

1.5.1 Visual Micro Plug-in

Esistono diverse alternative all' IDE ufficiale, ed una di queste è Visual Micro [21]. Visual Micro è un plugin gratuito per Microsoft™ Visual Studio®[9] e permette di sviluppare un progetto Arduino, compilarlo ed effettuare l'upload su una scheda Arduino.
Grazie all'avanzato IDE di base, con questo plugin è possibile sfruttare tutte le caratteristiche di Visual Studio, come il *syntax highlighting* e l'*auto completion.*

1.5.2 Programmazione

Qualsiasi sia l'IDE scelto, ogni sketch di Arduino segue sempre le stesse regole. Uno sketch può essere diviso in due parti. Queste due parti sono i punti di ingresso di qualsiasi programma Arduino. Il primo punto è chiamato fase di *setup*. È qui che vengono inizializzate tutte le risorse di Arduino, come ad esempio quali pin considerare ingressi o uscite, i servizi SD ed Ethernet, ecc. Il secondo punto è chiamato fase di *loop*, e come suggerisce il nome, Arduino ripeterà senza fine tutto il codice che sarà scritto all'interno di questo blocco, a meno di un evento esterno, come ad esempio un utente che preme il tasto *reset*. Come quasi tutti i linguaggi di questa tipologia, le zone esterne (prima e dopo) a queste due fasi possono essere utilizzate per l'inclusione di librerie, la definizione di variabili globali e di funzioni. Il diagramma di flusso in figura 1.10 mostra il ciclo di esecuzione di uno sketch.

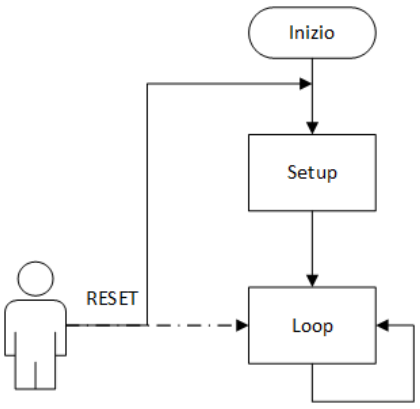

Figura 1.10 Ciclo di esecuzione Arduino.

La fase di *setup* avviene una sola volta, all'accensione di Arduino, dopodiché, a meno di fattori esterni, Arduino eseguirà la fase di *loop* all'infinito.

[9] Attualmente non disponibile per la versione Express. Disponibile invece per la versione Professional.

Capitolo II

2.1 Tecnologia RFID

L'acronimo RFID sta per *Radio Frequency Identification.*

La tecnologia RFID si riferisce a qualsiasi dispositivo elettronico che utilizza campi elettromagnetici a radio frequenza per facilitare la comunicazione dei dati con lo scopo di identificare, localizzare o analizzare lo stato di oggetti, inclusi esseri viventi, e rende possibile stabilire decisioni in tempo reale su queste informazioni [22].

Il primo utilizzo delle tecnologia RFID risale al 1940, con lo scopo di identificare velivoli militari. Questo permetteva agli alleati di determinare se un aereo era amico o nemico. Oggi, la tecnologia RFID è ancora utilizzata per scopi militari, ma anche per servizi commerciali. Il brevetto di questa tecnologia è riconosciuto a Charles Walton, nel 1983 [23].

Alcuni esempi di impiego di strumentazione RFID possono essere:

- Le chiavi elettroniche per aprire le portiere di un automobile.
- I servizi di pagamento automatico autostradale, i.e. Telepass.
- Sistemi di accesso autorizzato.
- Passaporti, carte di credito, e altri tipi di carte elettroniche.
- Documenti personali di riconoscimento.

I computer portatili ed i tablet, così come i lettori musicali ed altri dispositivi che possono connettersi senza fili ad un rete, utilizzano un identificativo univoco chiamato indirizzo Media Access Control (MAC). Anche altri tipi di protocollo come il Bluetooth utilizza un indirizzo MAC associato ai dispositivi che ne fanno uso.

I tag EAS[10] (*Electronic Article Surveillance*) che possiamo trovare in qualsiasi negozio di vestiti non utilizza questo tipo di tecnologia, poiché questi tipi di tag non identificano univocamente un prodotto, ma solamente se l'articolo in questione è stato pagato o meno [22].

Probabilmente il dispositivo più comune che fa uso di un identificatore univoco è il telefono cellulare. La sigla IMEI che sta per *International Mobile Equipment Identify*, identifica univocamente uno specifico telefono cellulare utilizzato su rete mobile.

[10] Dispositivi anti-taccheggio.

Infatti, i più recenti smartphone come iPhone® e la maggior parte dei telefoni Android™, utilizzano diversi identificatori a radio frequenza: un codice IMEI ed un indirizzo MAC sia per il Wi-Fi® 802.11 che la comunicazione Bluetooth®.

Come vedremo più avanti, l'utilizzo più vasto di questa tecnologia è legato alle smart card. Quindi tutto ciò che è possibile identificare univocamente in radio frequenza è tecnologia RFID, e moltissime comunicazioni wireless utilizzate oggi entrano tutte a far parte dello spettro radio. La radio frequenza ci permette di costruire un ponte tra il piano fisico e quello digitale, è in grado quindi di identificare e differenziare qualsiasi entità.

2.2 Funzionamento

Un sistema RFID utilizza un *tag*, o un etichetta, che viene attaccata all'oggetto che deve essere identificato. Un altro dispositivo radio, trasmettitore-ricevitore, chiamato interrogatore o più comunemente lettore, invia un segnale al tag in attesa di una risposta, ovvero le informazioni memorizzate all'interno del tag. Solitamente un lettore RFID viene collegato ad un computer con installato un software di gestione per elaborare i dati letti. A differenza dei più comuni codici a barre, un tag RFID non necessita di essere visivamente accessibile da un lettore.

La comunicazione tra tag e lettore avviene attraverso una tensione di alimentazione e come vedremo esistono tre differenti tipi di configurazioni RFID [24].

2.2.1.1 Passive Reader Active Tag

La configurazione *Passive Reader Active Tag* (PRAT) utilizza un lettore passivo ed un tag attivo. Il lettore riceve i segnali inviati dal tag (radiazione elettromagnetica a radio frequenza). Per essere attivo, il tag è alimentato da una sorgente esterna, come un batteria, ma a scapito delle dimensioni è possibile ottenere un vantaggio notevole sulla distanza di trasmissione, che può essere regolata da qualche centimetro a qualche centinaio di metri. Il lettore, a differenza di altri tipi di configurazioni, non deve quindi alimentare il tag. Questo consente una grande flessibilità, e può essere implementato in campi come la protezione ed il controllo dei beni (i.e. lo stoccaggio di materiale all'interno di un magazzino).

2.2.1.2 Active Reader Passive Tag

La configurazione *Active Reader Passive Tag* (ARPT) utilizza un lettore attivo ed un tag passivo. Il tag passivo, non avendo a disposizione un alimentazione pone delle limitazioni sulla distanza di comunicazione. Infatti il metodo di alimentazione è differente e può operare in due modi: riflettendo l'energia prodotta dal lettore, o assorbendo parte dell'energia del lettore per alimentare temporaneamente il circuito di trasmissione a radio frequenza. Il fenomeno descritto è chiamato induzione elettromagnetica o *backscattering*. Il lettore deve quindi disporre di un segnale abbastanza forte da poter attivare il tag. Inoltre a differenza del tag attivo, la comunicazione può avvenire solamente se il tag è vicino al lettore (campo di attivazione), nell'ordine di qualche centimetro.

2.2.1.3 Active Reader Active Tag

Infine la configurazione *Active Reader Active Tag* (ARAT) funziona con tag e lettori entrambi attivi. Il tag è costantemente alimentato e non fa altro che attendere che un lettore la interroghi. Ad ogni modo la distanza operativa è del tutto uguale a quella raggiungibile con una configurazione PRAT. Una variante di questa configurazione è con un tag semi-passivo, detto anche *Battery Assisted Passive* (BAP), in cui il tag rimane passivo ma all'interno ha una piccola batteria in grado di alimentarlo in presenza di un lettore.

2.3 Gerarchia di classi RFID

Un altro metodo per classificare i tag RFID è quello in base alla loro capacità di leggere e scrivere dati, mostrate in figura 2.1 e descritte di seguito.

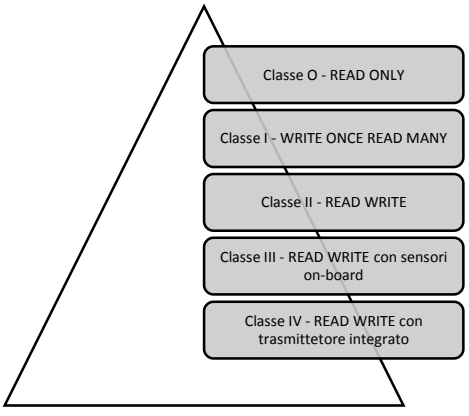

Figura 2.1 Gerarchia di classi RFID.

- **Classe 0**: è il più semplice tipo di tag, dove i dati sono solitamente un numero identificativo, e sono scritti una sola volta durante la creazione. Non è possibile alterare la memoria. È utilizzata anche per definire i tag EAS o dispositivi anti scippo, i quali non hanno un ID, ma viene solamente rilevata la loro presenza se esposti ad un lettore.
- **Classe 1**: in questo caso il tag viene prodotto senza alcun dato scritto in memoria. I dati possono essere scritti in un secondo momento o dal produttore o da un utente. A seguito di una scrittura non sarà più possibile alterare la memoria che potrà solamente essere letta. Solitamente sono utilizzati come semplici identificatori.
- **Classe 2**: sono i tipi di tag più flessibili, dove l'utente ha accesso sia alla scrittura che alla lettura dei dati in memoria. Sono spesso utilizzati come data logger, e quindi contengono più memoria rispetto a quella necessaria per un semplice ID.
- **Classe 3**: questi tipi di tag contengono dei sensori per analizzare diversi parametri come la temperatura, il movimento e altri, che possono essere salvati sulla memoria del tag. I sensori sono attivi in assenza di un lettore, e questi tag possono essere sia attivi che semi-passivi (a batteria).
- **Classe 4**: sono una specie di dispositivi radio in miniatura e possono comunicare con altri tag e dispositivi senza l'ausilio di un lettore. Sono tag attivi e quindi hanno la propria alimentazione locale.

Questa gerarchia di classi viene solitamente associata all'EPC (*Electronic Product Code*), ovvero uno standard per assegnare univocamente un identificativo permanente a qualsiasi oggetto del pianeta.

Come è facile intuire un tag passivo è sicuramente più economico, e può essere installato all'interno di superfici molto piccole, come smart card o addirittura in zone sotto cutanee.

Poiché ogni tag RFID è identificabile univocamente, alcuni lettori sono in grado di avviare connessioni simultanee con tutti tag nel raggio di azione.

Le informazioni di un tag sono salvate elettronicamente in una memoria non volatile. Un lettore può leggere diverse informazioni riguardo un tag, come il numero seriale univoco, le informazioni del prodotto come ad esempio il numero di stock, la data di produzione, o altre specifiche informazioni.

Un tag RFID è composto da almeno due parti: un circuito integrato per salvare, processare le informazioni, modulare e demodulare un segnale a radio frequenza, assorbire la corrente per il funzionamento tramite lettore RFID ed altre specifiche funzioni; e un antenna per ricevere e trasmettere il segnale.

2.4 Scelta della radio frequenza

La selezione della frequenza è basata su un numero molteplice di fattori, che includono:

1. Velocità di trasmissione dei dati.
2. Requisiti di alimentazione.
3. Distanza operativa.
4. Allocazione regolata dello spettro radio.
5. Caratteristiche di una data frequenza radio.

Ogni frequenza aderisce ad un insieme di proprietà, ed ogni frequenza ha un distinto raggio d'azione, una velocità e delle caratteristiche di interferenza che rendono più o meno vantaggiosa una frequenza rispetto ad un'altra per una specifica applicazione. Inoltre, ci sono dei regolamenti governativi, in base allo stato di appartenenza, di quali frequenze e livelli di potenza possono essere usati.

La tabella 2.1 indica alcune frequenze comunemente utilizzate da dispositivi RFID [25].

Banda	Regolazione	Portata	Velocità dati	Note, utilizzi
120 – 150 kHz (LF)	Non regolata	10 cm	Bassa	Identificazione animali, database fabbriche
13.56 MHz (HF)	Banda mondiale ISM	1 m	Bassa / Moderata	Smart card (MIFARE, ISO/IEC 14443)
433 MHz (UHF)	Dispositivi a corto raggio	1 – 100 m	Moderata	Applicazioni di difesa, con tag attivi
865 – 868 MHz EU 902 – 928 MHz NA (UHF)	Banda ISM	1 – 2 m	Moderata / Alta	EAN, vari standard
2450 – 5800 MHz (microonde)	Banda ISM	1 – 2 m	Alta	802.11 WLAN, Bluetooth standard
3.1 – 10 GHz (microonde)	Banda ultra larga	Fino a 200 m	Alta	Richiede tag semi-attivi o attivi

Tabella 2.1 Radio frequenze utili.

2.5 Smart card

Una smart card è una scheda di materiale plastico, solitamente PVC[11], delle dimensioni di una carta di credito come mostrato in figura 2.1, con un microchip integrato che può essere utilizzato per memorizzare dati e gestire informazioni.

Figura 2.2 Smart card.

Uno dei fattori chiave della diffusione di questa tecnologia è la sicurezza fornita dalle smart card in merito alla gestione delle informazioni in essa contenute. Ci sono alcuni punti focali su cui ogni tecnologia smart card dovrebbe assicurare un alta sicurezza. In primo luogo l'integrità dei dati e quindi la trasmissione tra una smart card ed un lettore deve avvenire senza alterazione dell'informazione. L'autenticazione intesa come il riconoscimento della smart card da parte del lettore, e/o viceversa, è un altro punto chiave e deve avvenire in totale sicurezza, in modo tale che il contesto operativo della smart card sia sicuro, e possa avvenire quindi la trasmissione dei dati. La riservatezza delle informazioni dunque è un altro obiettivo su cui una tecnologia per smart card dovrebbe intervenire. Ciò che non desideriamo è che un intruso possa accedere alla nostra smart card e decifrare i dati contenuti per poi utilizzarli in maniera fraudolenta. Infine è necessario garantire l'irriproducibilità della scheda, in modo tale che le carte non vengano clonate e quindi essere riconosciute come autentiche quando invece non lo sono.

In figura 2.3 è possibile vedere le differenti parti fisiche di una generica smart card.

[11] Cloruro di polivinile.

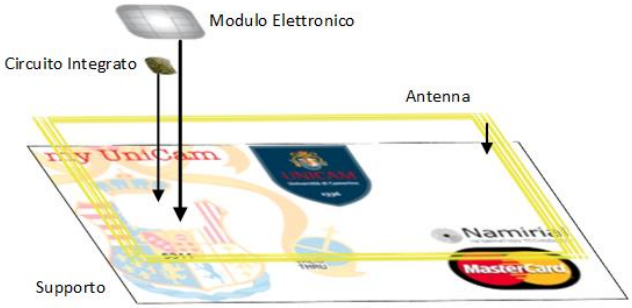

Figura 2.3 Componenti smart card.

2.5.1 Tipologie di smart card

Esistono tre tipologie principali di smart card [26]:

- **Integrated Circuit Memory Card**: questo tipo di smart card è composta da una memoria di tipo EEPROM. La memoria è installata in modo permanente sulla carta ed un meccanismo ha il compito di memorizzare i dati nel momento in cui la carta viene rimossa dal lettore. Generalmente la quantità di memoria EEPROM arriva ad un massimo di qualche decina di kilobyte. Questo tipo di carte hanno dunque una logica circuitale tale da permettere la memorizzazione dei dati e la comunicazione con il lettore. Il loro utilizzo, poiché prive di misure crittografiche, va dal contenere credito telefonico, ai biglietti per il trasporto pubblico, o carta della raccolta punti. Per applicazioni in cui si richiede una sicurezza elevate è necessario utilizzare smart card con microprocessore.
- **Integrated Circuit Microprocessor Card**: questo tipo di smart card è composta da un vero e proprio microprocessore con installato un sistema operativo. Questa componentistica fa sì che la carta possa lavorare in modo indipendente, e quindi possa processare ed elaborare i dati. All'interno della carta troviamo un file system in cui è possibile memorizzare le informazioni e mantenere un livello di sicurezza maggiore. Queste carte implementano differenti algoritmi di sicurezza [12] per garantire la riservatezza dei dati. Generalmente i microprocessori sono a 8 o 16 bit e offrono funzioni come lettura/scrittura della memoria, utilizzo di funzioni crittografiche, e altro. Alcune aziende, come la NXP, stanno cercando di introdurre anche dei sistemi di sicurezza alternativi, come la cosidetta *gesture smart card*, la

[12] RSA, DES, Triple-DES, AES.

quale utilizza un superficie a tocco nella quale inserire il PIN prima di effettuare una transazione, ed è anche in grado di riconoscere la traccia del dito in modo da imparare dei pattern personalizzati per aumentare ulteriormente la sicurezza [27] [28]. I comandi che vengono inviati al microprocessore si chiamano APDU[13]. Questi tipi di carte sono adatte all'identificazione sicura di un soggetto, o a tutti quei servizi in cui la sicurezza, e la riservatezza dei dati è un obbligo.

- **Optical Memory Card**: le memory card ottiche possono memorizzare diversi qualche megabyte di dati, ma una volta scritti, attraverso un particolare laser, non possono più essere cambiati o rimossi.

Oltre a queste tre distinzioni è possibile scindere le famiglie ulteriormente in smart card *contact* e smart card *contactless*. Come è possibile intuire, la differenza sta nella modalità in cui la carta si interfaccia con gli altri dispositivi, come i lettori. Nella modalità contact è necessario che la carta entri in contatto con il circuito elettrico per trasmettere dati al lettore, mentre in modalità contactless, in base alle specifiche della carta, è necessario solamente avvicinare la carta al lettore per essere letta.

Esistono anche altri tipi di carte, come le cosiddette *combi-card*, note anche come *dual interface*, in quanto dispongono sia della parte contact, sia della parte contactless, entrambe connesse allo stesso circuito integrato. Il circuito integrato di una smart card ha una dimensione di qualche millimetro per evitare che, flettendo la carta, possa rompersi.

L'ISO che regolamenta la connessione contact, e quindi il pinout[14] in figura 2.2, è l'ISO 7816.

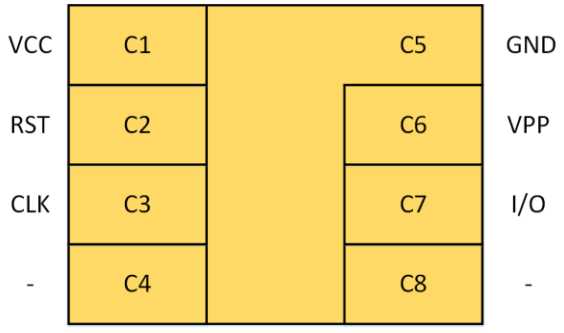

Figura 2.4 Pinout smart card contact.

[13] Application Protocol Data Unit
[14] La disposizione dei connettori.

Descrizione:

- **VCC**: alimentazione +5 V.
- **RSR**: segnale di reset per la comunicazione della smart card.
- **CLK**: fornisce alla carta il segnale di clock.
- **RESERVED**: (AUX1) opzionalmente usato per interfacce USB.
- **GND**: contatto di massa.
- **VPP**: tensione di programmazione di supplemento, richiesta a volte per eliminare la memoria interna.
- **I/O**: ingresso/uscita dei dati seriali per il circuito integrato della scheda.
- **RESERVED**: (AUX2) opzionalmente usato per interfacce USB.

L'ISO che invece regolamenta l'interfaccia contactless è l'ISO 14443, che specifica la radiofrequenza di 15,56 MHz utilizzata nella comunicazione, ed il protocollo APDU.

2.5.2 Protocollo APDU

APDU è l'acronimo per *Application Protocol Data Unit* ed è il protocollo di comunicazione tra un lettore RFID e una smart card. La struttura e le specifiche di questa comunicazione sono descritte nello standard ISO/IEC 7816-4. Il protocollo è diviso in comandi e risposte. I comandi sono quelli inviati dal lettore alla smart card, che quindi invia la risposta al lettore. Ogni comando è composto da un header di quattro byte, denominati CLA, INS, P1 e P2, e contiene da 0 a 255 byte di dati. La risposta invece è composta da due byte di status denominati SW1 e SW2 e da 0 a 256 byte di dati. La tabella riassume i campi, e le rispettive dimensioni della comunicazione APDU [29].

	Comandi APDU		
	Nome campo	Dimensione	Descrizione
Header	CLA	1	Classe - Indica il tipo di comando
	INS	1	Codice - Indica il comando specifico
	P1 – P2	2	Parametri per il comando
Corpo	L_c	0, 1 o 3	Codfica a seguire il numero (N_c) di byte dei dati
	Dati	N_c	N_c byte di dati
	L_e	0, 1, 2 o 3	Codifica il numero massimo (N_e) di byte attesi dalla risposta
	Risposta APDU		
	Dati	N_r (max N_e)	Dati di risposta (può essere vuoto)
	SW1 – SW2	2	Stato della risposta (i.e. 0x90 00 indica successo)

Tabella 2.2 Comandi e risposte APDU.

Per esempio un comando APDU è una sequenza di valori esadecimali del tipo:

```
0x00, 0xA4, 0x01, 0x0C, 0x02, 0x3F, 0x00.
```

2.5.3 Ciclo di vita di una smart card

Generalmente il ciclo di vita di una smart card, come mostrato in figura 2.5, attraversa le seguenti fasi. L'emittente della carta scrive l'ambiente di runtime e lo invia al produttore di chip. Il produttore di chip installa l'ambiente di runtime all'interno della ROM del chip, un processo chiamato mascheramento. A questo punto i chip vengono rispediti all'emittente della carta che li inserisce su ogni scheda e carica i dati delle applicazioni, questo processo è chiamato fase di pre-personalizzazione. Successivamente le carte vengono consegnate al fornitore dell'applicazione che stampa ed imprime le carte. Solitamente il fornitore installa anche delle proprie applicazioni con i propri dati, la cui fase è chiamata "personalizzazione". Infine il fornitore invia la carta al titolare, e quindi l'utente finale, che utilizza la smart card finché non viene distrutta o cade in disuso [30].

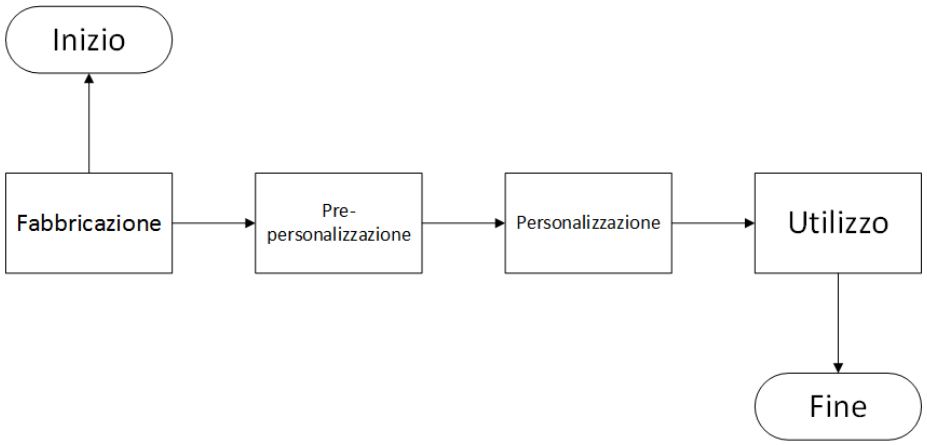

Figura 2.5 Ciclo di vita smart card.

2.5.4 File system

I dati di una smart card sono memorizza nella EEPROM, in modo simile ad un pc, dove abbiamo una gerarchia ad albero dei file.

In alto, la radice di questo albero, è presente obbligatoriamente il *Master File* (MF), dopodiché possiamo avere *Dedicated File* (DF) ed *Elementary File* (EF), che sono rispettivamente cartelle e file dati [31]. È possibile stabilire diversi livelli di sicurezza sulle

cartelle così da memorizzare i dati che devono essere protetti in locazioni più sicure. Esistono diverse modalità di accesso all'interno del file system:

- **File identifier:** è un codice esadecimale di due byte.
- **Path name:** è il path relativo al file, ed è costruito come concatenazione di file identifier.
- **Short identifier:** può essere utilizzato con un EF ed è rappresentato da 5 bit (valori da 1 a 30).
- **DF name:** un DF può avere un nome univoco codificato utilizzando da 1 a 16 byte.

Un esempio di gerarchia di file è quello mostrato in figura 2.6 [31].

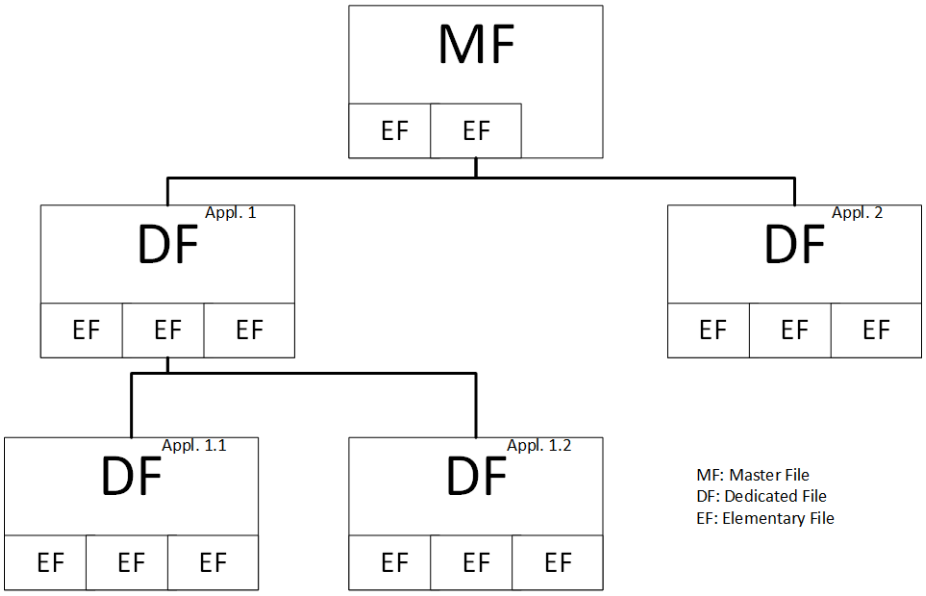

Figura 2.6 File system smart card.

Ci sono diversi *file identfier* riservati, i quali non possono essere cambiati, ma sono utili per ottenere molte informazioni:

- 0x3F00 MF Root.
- 0x0000 EF PIN e PUK #1.
- 0x0100 EF PIN e PUK #2.
- 0x0001 EF application keys.
- 0x0011 EF management keys.
- 0x0002 EF manufacturing info.
- 0x0003 EF card ID info.
- 0x0004 EF card holder info.
- 0x0005 EF chip info.
- 0x3FFF file path selection.
- 0xFFFF reserved.

2.5.5 Standard ISO

Gli standard ISO, *International Standardization for Organization*, per le smart card regolano le proprietà fisiche, la caratteristiche comunicative e tutti i vari identificatori dei circuiti integrati e dei dati [32]. Generalmente, per tutti i tipi di smart card viene preso l'ISO 7816-1,2 & 3 come riferimento di base

L'ISO 7816 è lo standard internazionale per le carte a circuito integrato (smart card), che utilizzano contatti elettrici, o che comunicano con i lettori senza contatto con tecnologia RFID.

L'ISO/IEC è uno degli standard a livello mondiale per la tecnologia in generale, incluse le smart card. Gli standard principali per le schede a circuito integrato sono: ISO/IEC 7816, ISO/IEC 14443, ISO/IEC 15693 e ISO/IEC 7501. I primi tre riguardano le smart card, mentre l'ultimo riguarda i lettori. Ci sono inoltre altri standard, che comunque si basano sugli standard ISO/IEC.

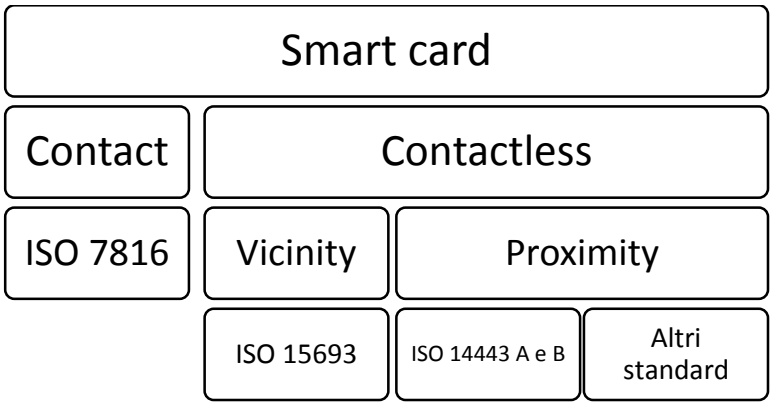

Figura 2.7 Smart card standard

2.5.5.1 ISO/IEC 7816

L'ISO/IEC 7816 è uno standard internazionale composto in 14 parti. Le parti 1, 2 e 3 coinvolgono solo le smart card a contatto e ne definiscono i vari aspetti della carta e le sue interfacce, come la dimensione fisica, le caratteristiche elettriche ed i protocolli di comunicazione. Le altre parti vengono prese come base anche per le smart card contactless. In generale questo standard copre principalmente i seguenti punti:

- Caratteristiche fisiche, dimensione e localizzazione dei contatti.

- Caratteristiche elettriche, tensioni di lavoro.
- Organizzazione file system, metodi di accesso e controllo.
- Sicurezza, comandi e gestione.

2.5.5.2 ISO/IEC 14443

L'ISO/IEC 14443 è uno standard internazionale che definisce le interfacce per le carte contactless, chiamate anche *carte di prossimità*, includendo l'interfaccia a radio frequenza (RFID), l'interfaccia elettrica, ed i protocolli di comunicazione e anti collisione. Queste tipologie di carte operano alla frequenza di 15,56 MHz e hanno una distanza operativa di circa dieci centimetri. È il primo standard per le smart card contactless utilizzato per le transazioni finanziarie e per il controllo di accesso delle applicazioni. È utilizzato anche nei passaporti elettronici.

I termini PICC[15] e PCD[16] sono utilizzati dallo standard ISO/IEC 14443 e stanno ad indicare rispettivamente la smart card ed il lettore.

L'ISO/IEC 14443 è diviso in quattro parti distinte:

- ISO/IEC 14443-1: Caratteristiche fisiche.
- ISO/IEC 14443-2: Radio frequenza e modulazione dei segnali (Tipo A e B).
- ISO/IEC 14443-3: Inizializzazione e anticollisione.
- ISO/IEC 14443-4: Protocollo di trasmissione.

ISO/IEC 14443 è anche uno di una serie di Standard Internazionali che descrive i parametri per l'identificazione delle carte come definito nell'ISO/IEC 7810[17], e l'utilizzo di tali carte per gli scambi internazionali.

Le carte di prossimità possono essere di tipo A e di tipo B, entrambe le quali comunicano ad una frequenza di 13.56 MHz. La principale differenza tra i due tipi riguarda il metodo di modulazione del segnale a radio frequenza (ISO/IEC 14443-2) ed i protocolli di inizializzazione (ISO/IEC 14443-3). Entrambi utilizzano lo stesso protocollo di trasmissione (ISO/IEC 14443-4).

[15] Proximity Integrated Circuit Card (carta di prossimità con circuito integrato).
[16] Proximity Coupling Device (lettore).
[17] Descrive le caratteristiche fisiche delle carte d'identità.

2.5.5.3 ISO/IEC 15693

L'ISO/IEC 15693 descrive gli standard per le *carte di vicinità* ovvero delle smart card contactless che possono operare a distanze maggiori rispetto alle carte di prossimità. La distanza massima operativa di queste carte è di circa un metro e mezzo.

Lo standard è diviso in tre parti:

- ISO/IEC 15693-1: Caratteristiche fisiche.
- ISO/IEC 15693-2: Interfaccia contactless e inizializzazione.
- ISO/IEC 15693-3: Sistema anticollisione e protocollo di trasmissione.

2.5.5.4 ISO/IEC 7501

L'ISO/IEC 7501 è lo standard per gli MRP (Machine Readable Passport), ovvero per i lettori dei passaporti elettronici che supportano questo standard. Specifica le caratteristiche fisiche, e la costruzione dei lettori.

2.5.6 Calypso Standard

Calypso ha come scopo lo sviluppo di una tecnologia non proprietaria, sicura e flessibile, per il sistema di biglietteria dei trasporti pubblici. Dopo dieci anni di sviluppo è stata creata un tecnologia smart card contactless adatta a questo utilizzo. Grazie alle specifiche tecniche non proprietarie è stata possibile la diffusione rapida di questa card, permettendo anche la conquista di una buona fetta di mercato.

La famiglia di smart card contactless di riferimento si chiama CD97, ora nota come Calypso, e si basa sull'ISO 14443 Tipo B. Le specifiche Calypso coprono tutto il sistema di utilizzo di una smart card contactless, e quindi anche il sistema centrale di controllo, le stampanti, i lettori, e altro.

Poiché il sistema di trasporti richiede un valore monetario rappresentato dalla carta, viene utilizzato un sistema crittografico per garantire la sicurezza basato su chiavi segrete, nascoste nella carta, e un modulo SAM (Secure Application Modules). Un SAM è una smart card presente nei sistemi Calypso che interagisce con la carta utente, spesso è connessa in modalità remota con le attrezzature Calypso.

Dopo la produzione, sulla carta, vengono scritte due tipi di informazioni: le chiavi segrete ed i dati applicazione.

Il sistema di transazione carta–terminale Calypso è:

- **Semplice, rapido e sicuro**: un sistema di codifica aumenta la velocità di transazione senza limitare gli scambi d'informazione.
- **Standardizzato ed evolutivo**: è basato sull'uso di carte a microprocessore, corrispondenti alle norme (ISO 14443-B per l'interfaccia contactless, ISO 7816-4 per l'organizzazione della memoria e della struttura dei file e CEN ENV 1545 per la definizione dei dati relativi a trasporto)
- **Interoperabile**: può trattare carte provenienti da varie reti, con la garanzia offerta dalle specifiche Calypso.
- **Multi-applicazione**: a seconda dei bisogni, la carta di trasporto comprende il borsellino elettronico, il parcheggio, i musei, gli eventi sportivi, i congressi, il turismo, e molti alti tipi di applicazioni.

La carta contactless Calypso è ideale per i servizi di trasporto pubblico. È compatibile con l'ISO 14443 Tipo B e con l'ISO 7816-3 e l'ISO 7816-4. Permette inoltre la codifica delle strutture dati relative ai trasporti come specificato nell'ISO 1545. Assicura transazioni veloci e sicure. Sono implementati differenti sistemi di sicurezza tra cui l'algoritmo DES con chiavi a 64 bit, l'algoritmo DESX con chiavi da 120 bit, e l'algoritmo 3DES con chiavi a 112 bit. Offre quindi un alto livello di sicurezza contro la contraffazione delle carte, e l'utilizzo non appropriato.

L'organizzazione dei file è piuttosto libera, ma deve contenere i file minimi richiesti dalle specifiche Calypso, derivate dagli standard ISO. I file sono suddivisi in Dedicated File (DF), e Elementary File (EF). I DF sono cartelle che possono contenere EF o altri DF. La cartella root DF è chiamata Master File (MF). Gli EF sono file che contengono dati, tra cui i dati utente. Ogni file è identificato da un Long Identifier (LID), o da un Short File Identifier (SFI). Ad ogni istante corrisponde solamente un file selezionato, chiamato file corrente su cui è possibile effettuare le operazioni desiderate. Esistono tre tipi di file, lineari, ciclici e contatori [33].

Calypso utilizza un sistema di *Secure Session* per garantire la sicurezza dei dati. Una Secure Session inizia con un specfico comando (*Open Secure Session*) e termina con il corrispettivo (*Close Secure Session*). Durante questa sessione, è possibile leggere e scrivere

dati nella carta. Quando la sessione termina, tutti i dati scambiati sono certificati dalla carta e dal SAM incluso nel terminale. Le certificazioni sono riguardo a:

- L'autenticazione del terminale.
- L'autenticazione della carta.
- La genuinità dei dati scambiati.

Infine, il comando assicura il terminale che i dati sono stati correttamente aggiornati sulla carta.

Come detto in precedenza, la tecnologia Calypso utilizza tre algoritmi: DES, DESX e 3DES. Ogni file all'interno della carta ha delle condizioni di accesso che seguono alcune regole. Uno specifico meccanismo di sicurezza può inoltre cambiare le condizioni di accesso di un file. In generale le condizioni di accesso sono tre [33]:

- Always: accesso libero, chiunque può accedere a questo file.
- Never: accesso non permesso, nessuno può accedere a questo file.
- Session: l'accesso è possibile solo con una Secure Session, utilizzando la chiave corrispondente. Questo tipo di accesso non si applica ai comandi di lettura.

2.5.7 Altri Standard

In questa sezione vengono descritti altri standard che regolamentano o certificano vari aspetti delle smart card.

2.5.7.1 Standard ICAO

La ICAO, acronimo di *International Civil Aviation Organization*, emette le specifiche per i *Machine Readable Passport* (MRP) tramite un documento chiamato *Machine Readable Travel Document*. L'ICAO standardizza quindi le caratteristiche fisiche che devono avere i passaporti elettronici, che implementano la tecnologia RFID. I vantaggi introdotti dall'utilizzo di un MRP sono riguardo alla velocità con cui vengono gestiti i passeggeri all'arrivo in un aeroporto, e l'affidabilità di un lettore elettronico rispetto ad una persona [34].

2.5.7.2 Standard FIPS

Il FIPS, *Federal Information Processing Standards*, sviluppato dalla *Computer Security Division* del NIST, *National Institute of Standards and Technology*, ha lo scopo di proteggere

le risorse federali, compresi i sistemi informatici e di telecomunicazione. Diverse norme si applicano alle smart card e riguardano gli standard di firma digitale, di cifratura avanzata, ed i requisiti di sicurezza per i moduli crittografici [35].

2.5.7.3 Standard EMV

La sigla EMV sta per *Europay, MasterCard and Visa*. Questa compagnia ha emesso delle specifiche per i sistemi di pagamento. Queste specifiche elencano delle tecniche base per memorizzare i dati di credito all'interno di una carta e per gestire l'autenticazione delle transazioni con carte di credito e di debito. Un altro obiettivo è anche quello di garantire l'interoperabilità e l'accettazione del sistema d pagamento con carte a circuito integrato a livello mondiale [36].

2.5.7.4 Standard PC/SC

Il *Personal Computer/Smart Card* è una specifica per la standardizzazione e l'interoperabilità tra smart card e computer. La specifica è composta da un insieme di librerie disponibili per Microsoft™ Windows®. Per i sistemi operativi Mac® OsX® e Linux® è stata sviluppata una libreria denominata *Lite*. Attraverso queste librerie è possibile interfacciare in semplicità una smart card con un personal computer, con lo scopo ulteriore di rendere interoperabili i dispositivi di diversi produttori [37].

2.5.7.5 GlobalPlatform

GlobalPlatform identifica, sviluppa e pubblica le specifiche tecniche e le configurazioni di mercato che agevolano l'avvio sicuro e interoperabile e la gestione di più applicazioni embedded su chip tecnologici sicuri. Questo standard è utilizzato in vari chip, come le smart card, i processori applicativi, schede SD, *token* USB e altri dispositivi di sicurezza. La tecnologia è stata adottata a livello globale in finanza, mobile/telecomunicazioni, pubblica amministrazione, sanità, settori *retail* e di transito, e inoltre è stata introdotta anche nelle recenti JavaCard. Per raggiungere questi risultati, GlobalPlatform si attiene ai seguenti principi guida [38]:

- Mantenere la stabilità delle specifiche adattandosi alle esigenze di mercato piuttosto che curare l'eleganza tecnica.

- Mantenere la retro compatibilità a fronte di aggiornamenti.
- Supportare una architettura sicura, con un vasto range di opzioni per soddisfare ogni esigenza di mercato.
- Conservare un form factor indipendente.
- Sviluppare strumenti di conformità, consentendo all'industria di convalidare l'aderenza di un prodotto per uno specifica organizzazione.

2.5.8 Java Card

La tecnologia JavaCard™ della Oracle permette di eseguire applicazioni, chiamate *applet*, basate su Java all'interno di una smart card, in un ambiente sicuro, con scarse risorse di memoria e potenza di calcolo [39]. È possibile eseguire più funzioni contemporaneamente e modificarle anche dopo che la carta è stata prodotta e spedita all'utente finale, e inoltre supporta anche gli standard esistenti ISO [40].

I benefici che si ottengono con l'utilizzo di questa tecnologia sono:

- Interoperabilità: tutti le applet sviluppate per JavaCard™ funzionano su tutte le smart card Java indipendentemente dal produttore e dall'hardware della carta.
- Sicurezza: derivata dal linguaggio di programmazione Java e quindi da un ambiente sicuro garantito da alti standard qualitativi. Le smart card utilizzano algoritmi di crittografia quali DES, 3DES, AES, RSA ecc.
- Multi applicazione: la tecnologia JavaCard™ permette la coesistenza di più applicazioni sulla stessa smart card. Inoltre un sistema di firewall fa sì che le diverse applicazioni siano mutualmente separate in modo da impedirne l'accesso ai dati.
- Dinamicità: una volta che smart card è stata prodotta è ancora possibile installare nuove applicazioni, aumentando così la personalizzazione per l'utente finale.
- Compatibilità: le API JavaCard™ sono compatibili con gli standard internazionali, quali ISO 7816 e ISO 14443.

Gli sviluppatori che utilizzano questa tecnologia possono inoltre beneficiare dei vantaggi del linguaggio Java, come la programmazione orientata agli oggetti e un ambiente di sviluppo intelligente [41].

Dall'introduzione della versione JavaCard™ 3.0 esistono due edizioni differenti: la Classic e la Connected. La prima si basa su un ventaglio di prodotti già esistenti ed è quindi completamente retro compatibile con le versioni precedenti, mentre la seconda è incentrata

su prodotti più avanzati, introducendo caratteristiche orientate alla rete, come il supporto di schemi di comunicazione quali HTML, REST e SOAP.

Per quanto riguarda le specifiche, la JavaCard™, come tutti gli ambienti Java, dispone di una JavaCard™ Virtual Machine (VM), che descrive il set di istruzioni, i comportamenti e le caratteristiche implementative, il formato dei file che ogni JavaCard™ possiede. Il JavaCard™ Runtime Environment (RE) invece specifica i comportamenti dell'ambiente esecutivo per ogni implementazione della tecnologia JavaCard™, e descrive l'utilizzo delle API JavaCard™, compatibili con gli standard internazionali, le quali contengono la definizione delle classi richieste per il supporto della VM e del RE [41]. Per capire meglio l'architettura di un applicazione JavaCard™ fare riferimento alla figura 2.5 [42].

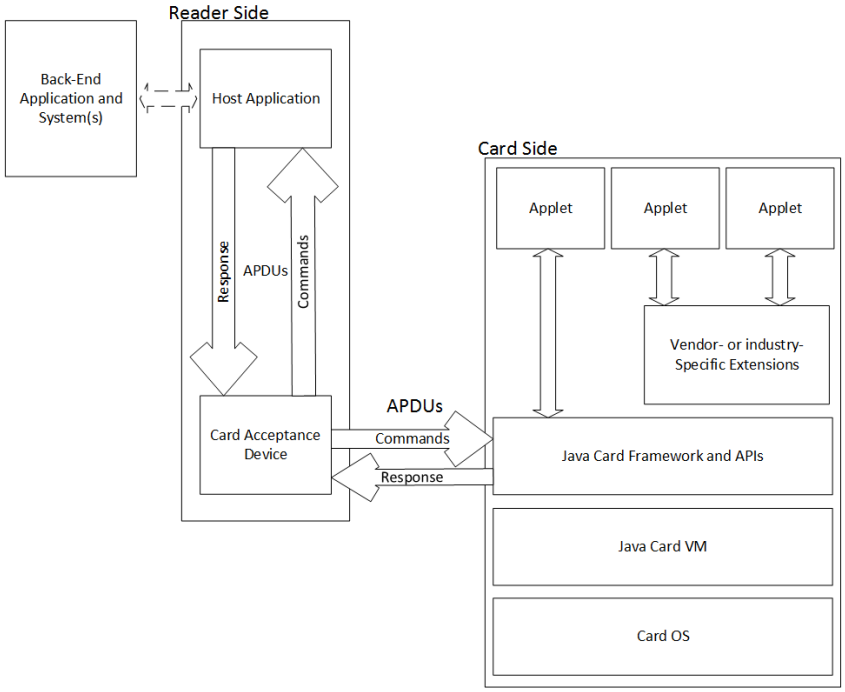

Figura 2.8 JavaCard Architettura applicazione

Le applicazioni di questa tecnologia coinvolgono tutta l'industria delle smart card, come le telecomunicazioni, le applicazioni finanziarie, e di identificazione dei soggetti, il trasporto pubblico, la pay-TV e il controllo degli accessi.

2.6 Tecnologie e carte in commercio

Ci sono diversi produttori, oramai affermati nel settore, che forniscono smart card con diverse caratteristiche. In questa sezione verranno riassunte le tecnologie di rilievo ed i

maggiori produttori di smart card, con i loro prodotti attuali. Si prega di fare riferimento ai rispettivi siti ufficiali dei produttori per specifiche approfondite, il seguente elenco vuole essere solamente una panoramica di alcune smart card presenti sul mercato.

2.6.1 Mifare Technologies

La tecnologia di smart card contactless MIFARE è sicuramente la più diffusa al mondo, l'utilizzo stimato è di circa l'80% delle smart card contactless presenti oggi [43]. È una tecnologia proprietaria brevettata da NXP Semiconductor, ed è basata sullo standard ISO 14443 tipo A. Lo standard è completo in quanto ci sono specifiche sia per le smart card che per i lettori. La distanza operativa è sempre di 10 centimetri circa come redatto nell'ISO 14443.

La tecnologia sviluppata dalla NXP appartiene a diverse famiglie di carte, ognuna con le proprie caratteristiche distintive [44].

2.6.1.1 Mifare Classic

La famiglia *Mifare Classic* è la più diffusa, per quanto riguarda i sistemi di ticket[18] con smart card contactless che operano a 13.56 MHz, con possibilità di lettura/scrittura e basate sull'ISO 14443A. Il circuito integrato *Mifare 1k* è stato il primo ad essere massivamente utilizzato per i sistemi di biglietteria per i trasporti pubblici.

Mifare Classic 1k è utilizzato prevalentemente per i sistemi chiusi, come nei biglietti con valori fissi (i.e. con validità settimanale) oppure come nei biglietti in cui è il service provider ad estrarre valori dalla carta.

Mifare Classic 4k apre nuove opportunità di servizi, abilitando le autorità del trasporto pubblico ad implementare sistemi multi-modali[19]. Potrebbe essere utilizzata per una combinazione di applicazioni contactless su una singola smart card.

- **Applicazioni**
  - Trasporto pubblico.
  - Gestione degli accessi.
  - Sistemi con biglietteria.
  - Giochi.
  - Identità.

[18] Sistemi di biglietteria.

[19] Sistema in cui sono implementate più applicazioni.

- **Caratteristiche**
  - 1 kb EEPROM (768 byte liberi) (Classic 1k).
  - 4 kb EEPROM (3480 byte liberi) (Classic 4k).
  - Numero seriale univoco (4 byte e 7 byte).
  - 16 settori separati che supportano multi-applicazione (Classic 1k).
  - 40 settori separati che supportano multi-applicazione (Classic 4k).
  - Ogni settore consiste in 4 blocchi di lunghezza 16 byte.
  - 2 x 48 bit keys per settore per la gerarchia delle chiavi.
  - Condizioni di accesso liberamente configurabili basate su 2 chiavi.
  - Numero di operazioni di singola scrittura: 100.000.
  - Conservazione dei dati: 10 anni.

2.6.1.2 Mifare Plus

È un aggiornamento rispetto alle Mifare Classic per garantire un livello di sicurezza elevato. È l'unico circuito integrato mainstream compatibile con Mifare Classic che offre un percorso di upgrade trasparente, con il minimo sforzo, per le infrastrutture e servizi esistenti. Mifare Plus è completamente retro compatibile con Mifare Classic 1k e Mifare Classic 4k. L'interoperabilità è verificata dall'indipendente Mifare Certification Institute [45]. Mifare Plus offre la possibilità di utilizzare carte senza problemi in sistemi Mifare Classic operativi, prima di aggiornare l'intera infrastruttura. Una volta che gli aggiornamenti di sicurezza sono attivi, Mifare Plus può passare ad un modalità più sicura, senza la necessaria interazione del cliente. AES è utilizzato per l'autenticazione, la crittografia e l'integrità dei dati.

Mifare Plus supporta un trasferimento dati ad alta velocità tra la carta ed il lettore, fino a 848 kbps, per servizi in cui il tempo di risposta è importante. Il raggio operativo è di 10 centimetri.

Le carte Mifare Plus supportano una pre-personalizzazione e tre livelli di sicurezza. Le carte operano in un livello di sicurezza alla volta, e può essere cambiato solo verso un livello più alto.

- Livello di sicurezza 0: le carte Mifare Plus sono pre-personalizzate con chiavi di configurazione, chiavi per il cambio di livello, Mifare Classic CRYPTO1[20] e chiavi AES per la memoria
- Livello di sicurezza 1: in questo livello le carte sono retro compatibili al 100% con le Mifare Classic 1k e le Mifare Classic 4k. Le carte possono lavorare senza problemi in un infrastruttura Mifare Classic.

[20] Algoritmo di cifratura proprietario della NXP.

- Livello di sicurezza 2: autenticazione AES obbligatoria, Mifare Classic CRYPTO1 per la riservatezza dei dati.
- Livello di sicurezza 3: autenticazione AES obbligatoria, comunicazione riservata ed integrità. Rilevamento di prossimità opzionale (solo Mifare Plus X).

Se una carta viene rimossa dal campo di azione durante l'aggiornamento di una chiave, o conclude l'aggiornamento o automaticamente ritorna allo stadio precedente. NXP raccomanda l'utilizzo di 7 byte per l'UID, ma offre anche una versione UID a 4 byte durante la migrazione da Classic verso Mifare Plus.

- **Applicazioni**
  - Trasporto pubblico.
  - Gestione degli accessi, i.e. impiegati, carte studentesche.
  - Pedaggio elettronico.
  - Parcheggio auto.
  - Programmi di fedeltà.

- **Caratteristiche**
  - 2 o 4 kb EEPROM.
  - Struttura di memoria fissa e semplice, compatibile con Mifare Classic 1k e Mifare Classic 4k.
  - Condizioni di accesso liberamente configurabili.
  - Migrazione semplice da Mifare Classic ai livelli di sicurezza forniti da Mifare Plus.
  - Standard crittografico aperto AES per l'autenticazione, l'integrità e la crittografia.
  - Certificazione Common Criteria: EAL4+ per IC HW e SW.
  - ISO/IEC 14443-A Numero seriale univoco (7 byte) o non univoco (4 byte), e ID random.
  - Autenticazione multi-settore, lettura e scrittura multi-blocco.
  - Funzione anti-tear per scrivere le chiavi AES.
  - Le chiavi possono essere memorizzate come chiavi Mifare Classic CRYPTO1 (2 x 48 bit per settore) o come chiavi AES (2 x 128 bit per settore).
  - Supporta il concetto di carta virtuale.
  - Trasferimento dati fino a 848 kbps.
  - Disponibile in moduli MOA4 o in wafer da 8 pollici.

2.6.1.3 Mifare DESFire EV1

Mifare DESFire EV1 è l'ideale per i fornitori di servizi che vogliono utilizzare smart card multi-applicazione per l'identità, l'accesso, programmi di fedeltà e trasporti. È completamente compatibile con i requisiti di velocità e alta sicurezza di trasmissione dei dati, organizzazione flessibile della memoria ed interoperabilità con le infrastrutture esistenti. È compatibile con tutti e quattro i livelli dell'ISO/IEC 14443 Tipo A e utilizza comandi opzionali dell'ISO/IEC 7816-4.

Caratterizzato da un chip con sistema di backup, e un autenticazione mutua a tre passi, la DESFire EV1 può contenere fino a 28 differenti applicazioni e 32 file per applicazione. La dimensione di ogni file è definita al momento della sua creazione, rendendo la Mifare DESFire EV1 un prodotto flessibile e conveniente.

Inoltre, un meccanismo automatico di *anti-tear*[21] è disponibile per ogni tipo di file, il quale garantisce l'integrità dei dati per le transazioni. Con Mifare DESFire EV1, il trasferimento dati è di massimo 848 kbps.

Le caratteristiche principali del chip sono evidenziate dal suo nome, DESFire EV1. DES indica l'alto livello di sicurezza ottenuto utilizzando un sistema crittografico basato su 3DES[22] per la crittografia dei dati trasmessi. Fire invece sta per Fast, Innovative, Reliable e sEcure IC. La Mifare DESFire EV1 offre un sistema avanzato user friendly, in combinazione con la sicurezza e l'affidabilità.

Questa carta offre un sistema bilanciato tra velocità, prestazioni e costo/efficienza. Il concetto di open permette future integrazioni, prive di problemi, da parte di altri sistemi contactless, e da tecnologie differenti come basate su NFC (Near Field Communication). È inoltre completamente compatibile con tutti i lettori Mifare esistenti.

- **Applicazioni**
  - Trasporto pubblico avanzato.
  - Gestione di accessi.
  - E-Government (servizi sociali).
  - Identità.

- **Caratteristiche**
  - Compatibile al 100% con l'ISO/IEC 14443 Tipo A, (1-2-3-4).
  - 2 kb, 4 kb, e 8 kb EEPROM con programmazione veloce.
  - Set di comandi sicuro e ad alta velocità.
  - Trasferimento dati veloce fino a 848 kbps.
  - Struttura file flessibile (scelta di 5 tipi di file).
  - Scelta di hardware per sistema crittografico aperto DES/2K3DES/3K3DES/AES.
  - Certificazione Common Criteria: EAL4+ per IC HW e SW.
  - Meccanismo di backup dati automatico (Anti-tearing).
  - Numero seriale univoco a 7 byte (ID random opzionale configurabile).
  - Integrità dei dati: CRC e conta dei bit sul physical layer.
  - Tipo opzionale 70pF ottimizzato per progetti small form factor.

2.6.1.4 Mifare Ultralight

I ticket basati su chip NXP Mifare Ultralight possono funzionare come ticket per singolo viaggio nelle reti di trasporto pubblico, carte di fedeltà, o anche come carte pass per grandi eventi. Sono un rimpiazzo ideale per le biglietterie convenzionali, come quelle di carta, o carte a bande magnetiche, o addirittura in cash.

Poiché l'utilizzo dei sistemi contactless con carte di prossimità è sempre più comune, gli operatori dei trasporti stanno incominciando a trasferirsi completamente a soluzioni

[21] Metodo di protezione dei dati nella comunicazione tra smart card e lettore.
[22] Formalmente Triple-DES.

contactless. L'introduzione del nuovo chip contactless Mifare Ultralight per ticket ad uso limitato comporta una riduzione di installazione dei sistemi e dei costi di mantenimento. Mifare Ultralight può facilmente essere integrato in schemi già esistenti. I vantaggi offerti agli utilizzatori abituali di sistemi di trasporto (i.e. con Mifare DESFire) sono disponibili anche per gli utilizzatori occasionali tramite Mifare Ultralight. Migliora in modo significativo i tempi di imbarco e l'esperienza di viaggio, con bus, treni e altri mezzi di trasporto.

Le specifiche meccaniche ed elettroniche di Mifare Ultralight sono disegnate per ottenere gli stessi requisiti dei sistemi convenzionali. Emettere dei ticket in carta normale basati su Mifare Ultralight richiede un piccolo aggiornamento allo standard Edmonson / Eurosize implementato dalle biglietterie. Questo può essere ottenuto installando un semplice lettore contactless per l'inizializzazione dei ticket.

Mifare Ultralight è basato sullo standard ISO 14443 Tipo A. La distanza operativa è quindi di 10 centimetri. Infine, il Mifare Ultralight è compatibile con tutte le infrastrutture Mifare esistenti, e può quindi essere facilmente integrato.

- **Applicazioni**
  - Ticket ad uso limitato nei trasporti pubblici (i.e. viaggi singoli e multipli, pass per turisti).
  - Ticket per eventi (stadio, mostre, ecc.).
- **Caratteristiche**
  - 100% compatibile con le strutture Mifare esistenti.
  - Supporto anti collisione.
  - Distanza operativa di 10 centimetri.
  - Trasferimento dati a 106 kbps.
  - Numero seriale univoco a 7 byte (ISO 14443 Tipo A, level 2).
  - Integrità dei dati, 16 bit CRC, byte parity, bit encoding, conto dei bit.
  - 512 bit EEPROM, organizzata in 16 pagine da 4 byte.
  - Area da 32 bit programmabile una sola volta.
  - Area utente da 384 bit di lettura/scrittura.
  - Campo sola lettura programmabile per bloccare la funzionalità di ogni pagina.
  - Supporto metodo di sicurezza DESFire SAM.

2.6.1.5 Mifare Ultralight C

Il Mifare Ultralight C è l'ultima tecnologia Mifare, integrando l'offerta nel settore low-cost. È la prima smart card per applicazioni ad uso limitato che offre agli sviluppatori e ai fornitori i vantaggi di una crittografia aperta. Con il 3DES, la Mifare Ultralight C, utilizza uno standard ampiamente adottato, assicurando un integrazione semplice nelle infrastrutture esistenti. I comandi di autenticazione integrati offrono efficienti funzionalità anti clonamento che aiutano a prevenire la contraffazione dei ticket.

I ticket Mifare Ultralight C possono funzionare come biglietti di viaggio singolo nelle reti di trasporto pubblico o carte di fedeltà, o anche pass in occasione di grandi eventi, o tag per manifesti smart nelle applicazioni NFC. Sono ideali come rimpiazzo dei sistemi convenzionali di biglietteria, come quelli di carta o bande magnetiche, o denaro in cash.

Mifare Ultralight C è basata sullo standard ISO/IEC 14443 Tipo A. La distanza operativa è di 10 centimetri con supporto anti collisione. È totalmente compatibile con le infrastrutture Mifare già esistenti, e può quindi essere facilmente integrato nelle attuali soluzioni contactless.

- **Applicazioni**
  - Ticket ad uso limitato nei trasporti pubblici (i.e. viaggi singoli e multipli, pass per turisti).
  - Ticket per eventi (stadio, mostre, ecc.).
  - Carte fedeltà e prepagate.
  - NFC Forum Tag Type 2

- **Caratteristiche**
  - 100% compatibile con ISO/IEC 14443 Tipo A (1-2-3).
  - Trasferimento dai a 106 kbps.
  - Supporto anti-collisione.
  - 1536 bit (192 byte) EEPROM.
  - Accesso ai dati protetto con autenticazione 3DES.
  - Protezione anti clonamento.
  - Set di comandi compatibile con Mifare Ultralight.
  - Struttura di memoria come Mifare Ultralight (a pagine).
  - 16 bit counter.
  - Numero seriale univoco a 7 byte.
  - Numero di operazioni di scrittura: 10.000.

2.6.1.6 Mifare Ultralight EV1

È la nuova generazione di circuiti integrati per ticket smart card ad uso limitato che offre a sviluppatori ed operatori la massima flessibilità per i sistemi di ticketing e misure di sicurezza addizionali.

L'integrazione con gli attuali sistemi Mifare è garantita dalla totale compatibilità. Il controllo integrato originale è una protezione efficace contro la clonazione, ed aiuta a prevenire la contraffazione dei biglietti.

Mifare Ultralight EV1 adotta un sistema avanzato di sicurezza per le applicazioni contactless e fornisce una perfetta soluzione per sistemi completamente contactless. Può essere facilmente integrato con le attuali installazioni Mifare.

- **Applicazioni**
  - Ticket ad uso limitato nei trasporti pubblici.
  - Ticket per eventi (stadio, mostre, ecc.).
  - Carte fedeltà.

- **Caratteristiche**
  - 100% compatibile con ISO/IEC 14443 Tipo A (1-2-3)
  - Retro compatibile con Mifare Ultralight
  - Trasferimento dati a 106 kbps
  - Supporto anti collisione
  - Comando di lettura veloce
  - Varianti a 384 e 1024 bit di memoria utente
  - OTP, Lock Bits, contatori configurabili
  - Tre contatori a 24 bit indipendenti
  - Accesso ai dati protetto da una password a 32 bit
  - Firma di originalità NXP Semiconductor
  - Preparazione per funzionalità Virtual Card
  - Numero seriale univoco a 7 byte
  - Numero di operazioni di scrittura: 10.000

2.6.1.7 Mifare SmartMX – Mifare PRO

La famiglia NXP SmartMX propone i più alti standard di prestazioni e requisiti di sicurezza riducendone il costo complessivo. È una tecnologia largamente adottata, affidabile per le transazioni, con almeno un miliardo di circuiti integrati venduti, che offre prestazioni all'avanguardia nel funzionamento contactless e con un tempo di personalizzazione ridotto. I prodotti sono certificati con Common Criteria EAL 5+, in modo da essere protetti da attacchi di luce, guasti, ed attacchi side-channel[23], e vengono forniti con licenza CRI per le funzionalità migliorate di resistenza ad attacchi DPA/SPA[24] [46].

- **Applicazioni**
  - La convergenza dei mezzi di trasporto pubblici con: eGovernment: ePassport, carte di identità, carte di benessere salutare e sicurezza sociale, carte per cittadini e permessi di residenza, licenze di guida, controlli ad alta sicurezza per gli accessi.
  - Banking: carte di debito e di credito (MasterCard PayPass, VISA gVSDC), fedeltà, pagamento con dispositivi mobile.
  - Pay TV.

- **Caratteristiche**
  - Sicurezza certificata in accordo al CC EAL5+
  - Da 8 a 144 kb EEPROM
  - Conservazione dati: 25 anni.
  - Durata: minimo 500.000 cicli.
  - 160 a 264 kb ROM.
  - 3.5 a 7.5 kb RAM.
  - Interfacce: contact ISO/IEC 7816, contactless ISO/IEC 14443 Tipo A.
  - Tensione classe: C, B e A (1.62 a 5.5 V).
  - Memory Management Unit (MMU).
  - Implementazione Mifare Classic (1k e 4k).
  - Coprocessore 3DES ad alta velocità (64 bit parallelo).
  - Coprocessore AED ad alta velocità (128 bit parallelo).
  - Coprocessore PKI (RSA, ECC) FameXE (32 bit parallelo).
  - Ampio spettro di tipi di consegna.

[23] Debolezza di una particolare implementazione di un crittosistema.

[24] Simple Power Analysis e Differential Power Analysis.

2.6.1.8 Mifare SmartMX2

Sviluppare sistemi di multi-applicazione presenta nuove sfide in termini di sicurezza, convenienza, e produttività. Assicurare l'integrità dei dati è difficile, dal momento che ogni applicazione ha i propri requisiti di sicurezza, ed ogni applicazione deve rimanere separata dalle altre. Le prestazioni devono essere alte, poiché oggi gli utenti si aspettano la convenienza di un transizione veloce. Poiché i sistemi di multi-applicazione sono molto complessi, possono influire sui tempi di mercato e tempi di consegna.

Con la SmartMX2, è possibile avere sistemi che combinano più funzioni, senza preoccuparsi di sicurezza, prestazioni, convenienza e produttività. La combinazione di alte prestazioni e sicurezza, e l'integrazione con le altre implementazioni Mifare (Mifare Classic, Mifare Plus, o Mifare DESFire) fanno della SmartMX2 una piattaforma potente e flessibile.

- **Applicazioni**
  - La convergenza dei mezzi di trasporto pubblici con:
  - eGovernment: ePassport, carte di identità, carte di benessere salutare e sicurezza sociale, carte per cittadini e permessi di residenza, licenze di guida)
  - Banking: carte di debito e di credito, fedeltà, ePurse, ATM)
  - Controlli di accesso (strutture, o sistemi informatici)
  - Transazioni mobile (pagamenti, trasporti, controllo di accesso)
  - Autenticazioni dei dispositivi (protezione hardware e software, accesso sicuro a contenuti e servizi online, autenticazioni sicura tra macchine)

- **Caratteristiche**
  - Trasferimento dati contactless fino a 848 kbps in combinazione con una frequenza di clock interno fino a 150 MHz, sia con contact che con contactless, anche con bassa intensità di campo di 1.5 A/m
  - Supporto di operazioni simultanee con interfacce ISO/IEC 7816 e ISO/IEC 14443 per multi-applicazioni, pagamenti mobile inclusi.
  - Interfacce standard industriali I2C (400 kbps) e SPI (2 Mbps)
  - Interfacciamento semplice con IC NFC
  - Supporto hardware dedicato per esecuzioni sicure e veloci di routine software ricorrenti. Per esempio, l'utilizzo di una macchina di copia tra memorie e registri, insieme all'implementazione di un supporto di protocollo UART, risparmia un numero significante di linee di codice ed incrementa la velocità di copia
  - Set di istruzioni avanzato a 8, 16, 24 e 32 bit che minimizza il numero di cicli della CPU, per un esecuzione rapida e un consumo minore con il massimo delle prestazioni
  - Opzioni di memoria avanzate: da 254 a 384 kb di ROM, da 4 a 8.125 kb di RAM, da 24 a 144 kb di EEPROM e da 256 a 400 kb di pagine flash nella stessa architettura e interfaccia
  - SmartMX2 supporta tutte le versioni di Mifare
  - Disponibile con package MOB6 di spessore 250 µ m, migliorando la robustezza della carta e mettendo a disposizione spazio addizionale per altre caratteristiche.

Mifare	EEPROM	Standard ISO/IEC	UID	Integrity check	Velocità	Mantenimento dati	Cicli di lettura/ scrittura	Sicurezza	Altro
Classic	1 o 4 kb	14443 A	4 o 7 byte	16 bit CRC	106 kbps	10 anni	100.000	CRYPTO1	
Plus	2 o 4 kb	14443 A	4 o 7 byte (o 4 NUID)	8 bit CRC	848	10	200.000	AES o CRYPTO1	Retro compatibile Mifare Classic
DESFire EV1	2 o 4 o 8 byte	14443 A	7 byte	32 bit CRC	848	10	500.000	DES, 3DES, AES	28 multi applicazioni
Ultralight / Ultralight C	512 / 1536 bit	14443 A	7 byte	16 bit CRC	106	5	10.000	OTP Area, Read Only (3DES ver.C)	
Ultralight EV1	1024 o 1312 bit	14443 A	7 byte	16 bit CRC	106	10	100.000	OTP Area, ECC	
SmartMX / PRO	8 o 144 kb	14443 A e 7816, Dual Interface	7 byte	16 bit CRC	848	25	500.000	3DES, AES, PKI	RAM 3,5 a 7,5 kb, ROM 160 a 264 kb, Implementazione Mifare Classic
SmartMX 2	24 o 144 kb	14443 A e 7816 Dual Interface	7 byte	Integral security architecture	848	25	500.000	NXP-patented security, RSA/ECC, DES/AES	CPU 8 a 32 bit, RAM 4 a 8125 kb, ROM 264 a 384 kb, Abilitazione NFC, Interfacce I2C e SPI

Tabella 2.3 Riepilogo Mifare

2.6.2 Emulazione Mifare

Come detto in precedenza Mifare è uno dei chip contactless più usati al mondo, nonostante soggetto a diversi problemi di sicurezza, come dimostrano i vari hack. Questa diffusione globale, fa sì che anche altre tecnologie devono adottare un metodo per rendere compatibili le proprie carte ed i propri lettori con la tecnologia Mifare. Nasce quindi l'emulazione Mifare, sviluppata dalla stessa casa produttrice, che è un applicazione in esecuzione sul sistema operativo di un chip. L'emulazione fa riferimento al Mifare Classic, e fornisce la totale compatibilità, sia hardware che software. Viene quindi utilizzato lo stesso pinout del chip e gli stessi comandi Mifare. Nonostante ciò, ci sono alcune differenze rispetto al sistema nativo [47]:

- Tutti i problemi di sicurezza del Mifare Classic vengono trasmessi anche al sistema emulato.
- L'esecuzione dei comandi risulta più lenta nell'emulazione.

Per aggiungere l'emulazione Mifare ad un chip vengono riservati 1k, o 4k, a seconda del Mifare Classic di riferimento, e viene poi installata la versione Mifare in questo spazio.

2.6.3 Infineon Technologies

Infineon è un produttore mondiale di smart card con chip integrato [48]. Offre un'ampia gamma di prodotti nel settore in tutte le principali applicazioni di smart card e sottolinea la sua posizione di leader nello sviluppo tecnologico e la sicurezza del prodotto. La tecnologia per le smart card è compatibile con gli standard ISO 14443 A&B. I prodotti Infineon si adattano facilmente alle infrastrutture esistenti, riducendo in modo significativo i costi di distribuzione. Gli ultimi sviluppi hanno anche introdotto un nuovo sistema di sicurezza chiamato *Integrity Guard*, un concetto rivoluzionario basato su meccanismi di sicurezza digitali realizzati con modelli matematici [49].

2.6.3.1 Famiglia SLE 66CLxxxPE

La famiglia SLE 66CLxxxPE comprende smart card con doppia interfaccia, contact e contactless. La progettazione di queste carte è incentrata sull'aumento delle prestazioni e della velocità di esecuzione per carte come e-passport, carte di identità, e carte di credito. Sono compatibili con gli standard ISO 14443 A e B, ISO 7816, e hanno l'opzione di emulazione Mifare. Il chip integrato contactless velocizza l'autenticazione e le operazioni di codifica e decodifica. I prodotti sono certificati Common Criteria EAL 5+ high. Gli algoritmi di crittografia utilizzati sono RSA, DES, 3DES ed Ellyptic Curves[25].

- **Applicazioni:**
  - Controllo di accessi.
  - Pagamenti.
  - Trasporto pubblico.
  - Identità.
  - Multi-applicazioni.
- **Caratteristiche:**
  - Doppia interfaccia.
  - EEPROM da 4 kb fino a 144 kb.
  - USER-ROM da 92 kb fino a 241 kb.
  - Emulazione Mifare disponibile.
  - Microprocessore da 8 o 16 bit.
  - Sistema di crittografia 3DES, RSA, ECC.

2.6.3.2 Famiglia SLE 77 SOLID FLASH

Questa famiglia di prodotti è progettata per offrire una soluzione a basso costo per carte finalizzate al pagamento. La tecnologia *SOLID FLASH* introduce una memoria non

[25] Le curve ellittiche sono utilizzate in diversi metodi di fattorizzazione di numeri interi che sono utilizzati in crittologia.

volatile alla scheda garantendo un elevato livello di sicurezza con possibilità di blocco, codifica e correzione degli errori della memoria. Questa memoria conterrà il codice operativo ed i dati.

- **Applicazioni:**
  - Pagamenti.
  - Identità.
  - Trasporto.

- **Caratteristiche:**
  - Memoria SOLID FLASH da 80 a 240 kb.
  - RAM da 4 a 8 kb.
  - Crittografia DES, AES, RSA, ECC.
  - Compatibile con ISO 14443 e ISO 7816.

2.6.3.3 Famiglia SLE 78 SOLID FLASH

Una delle ultime famiglie introdotte da Infineon con controller a 16 bit contactless e doppia interfaccia. La progettazione di questa famiglia ha come punti focali la robustezza, la qualità e una lunga durata della scheda. Inoltre è stata realizzata con architettura open così da rendere più semplici ulteriori estensioni hardware.

Questa famiglia introduce anche un nuovo sistema chiamato Integrity Guard. Questo sistema offre un elevato livello di sicurezza per quanto riguarda il *core* della scheda. Sono state implementate due CPU per l'individuazione di errori, una codifica completa su CPU, memoria, bus e cache. Sono utilizzati codici di controllo errore su tutte le memorie, e altro ancora.

- **Applicazioni:**
  - Pagamenti.
  - Piattaforme globali.
  - Carte di fedeltà.
  - ePurse, eSignature.
- **Caratteristiche:**
  - Doppia CPU 16 bit
  - EEPROM da 36 a 48 kb, oppure SOLID FLASH da 240 a 300 kb
  - ROM 280 kb (no SOLID FLASH)
  - RAM 6 o 8 kb.
  - Crittografia RSA, ECC, 3DES, AES, DES.
  - Certificazione EAL 6+ high.

2.6.3.4 my-d proximity SLE 66Rxxx

Tutte le carte di questa famiglia sono conformi all'ISO 14443 tipo A. Sono carte che vengono usate principalmente per il trasporto pubblico, o eventi giornalieri. Alcune di queste carte hanno anche integrata la tecnologia NFC[26].

[26] Near Field Communication.

- **Applicazioni**:
  - Trasporto pubblico.
  - Controllo degli accessi.
  - Eventi giornalieri.
- **Caratteristiche:**
  - UID 7 byte.
  - EEPROM da 152 kb a 5.120 kb.
  - Autenticazione mutuale con chiavi a 64 bit.
  - Possibilità struttura di sicurezza multi livello.
  - CRC 16 bit
  - Distanza operativa 10 centimetri.

INFENEON	Microprocessore	EEPROM	ISO	User ROM	Sicurezza	RAM	UID	Velocità	SOLID FLASH	Altro
SLE 66CLxxxPE	8 o 16 bit	4 a 144 kb	ISO 14443 A e B, ISO 7016	92 a 144 kb	3DES, RSA, ECC	2 a 6 kb (XRAM)	4 o 7 byte	848 kbps	-	CC EAL 5+
SLE 77 SOLID FLASH	16 bit	-	ISO 14443 A e B, ISO 7016	-	DES, AES, RSA, ECC	4 a 8 kb	7 byte	848 kbps	80 a 240 kb	CC EAL 6+
SLE 78 SOLID FLASH	16 bit DUAL CPU	Da 36 a 48 kb	ISO 14443 A e B, ISO 7016	280 kb	DES, 3DES, ECC, AES, RSA		7 byte	848 kbps	Da 240 a 300 kb	CC EAL 6+
SLE 66Rxxx my-d proximity	-	Da 152 a 5.120 kb	ISO 14443 A, ISO 15693	-	Autenticazione mutuale a 64 bit	-	4 o 7 byte	848 kbps	-	Abilitazione NFC

Tabella 2.4 Riepilogo Infineon

2.6.4 Legic Technologies

Legic propone un sistema tecnologico per smart card contactless. Viene utilizzato soprattutto per sistemi identificati, sia in contesto lavorativo che per strutture di tempo libero. Solitamente in ambito lavorativo è utilizzata come tessera personale di identificazione, controllo accessi, rivelazione presenze, parcheggio automatizzato e altro, per i dipendenti. In contesti non lavorativi viene utilizzata come un badge individuale, come ticket nei sistemi di biglietteria automatica, pagamenti elettronici, accesso ai parcheggi ecc. Come tutte le tecnologie smart card, i reali servizi implementabili sono limitati all'intelletto di chi le pensa [50].

La tecnologia consente una libera scelta tra standard industriali come ISO 14443, ISO 15693, NFC e la norma proprietaria Legic RF. Legic fornisce anche soluzioni per reader, e transponder, in grado di gestire autenticazione per 127 applicazioni differenti.

È stato sviluppato anche un sistema chiamato *Master-Token System Control*, in grado di gestire la sicurezza all'interno di un organizzazione di lettori e schede.

La Legic offre tre famiglie di prodotti: *advant*, *prime*.

2.6.4.1 Legic Advant

È un sistema di smart card avanzato disponibile per standard ISO 14443 Tipo A e ISO 15693. I chip hanno come caratteristiche principali la flessibilità con cui molte applicazioni posso coesistere all'interno della stessa smart card (fino a 127), e la possibilità di utilizzare standard ISO differenti, includo lo standard proprietario Legic RF.

- **Applicazioni:**
  - Variabili, dalle più comune identificazione, al controllo accessi ai servizi di carta di credito e pagamenti.
- **Caratteristiche:**
  - EEPROM 2 o 4 kb (ISO 14443 A); 1002 byte o 2984 (Legic RF).
  - Standard supportati ISO 15693, ISO 14443 A, Legic RF.
  - Trasferimento dati 106 kbps a 848 kbps (ISO 14443 A).
  - Crittografia 3DES, DES, LEGIC encryption, AES.
  - Chiavi crittografiche a 64 o 112 bit
  - Numero massimo di applicazioni 123 o 127 (ISO 14443 A)
  - Segmentazione della memoria dinamica
  - Certificazione Common Criteria EAL 4+.

2.6.4.2 Legic Prime

Uno dei primi prodotti Legic, presentato nel 1992, oramai in disuso a causa delle vulnerabilità presenti. Legic mette a disposizione un sistema di aggiornamento per passare ai sistemi Advant [51].

LEGIC	EEPROM	Standard	Trasferimento dati	Sicurezza	Numero applicazioni	Autenticazione crittografica	Altro
Advant	Da 128 byte a 4 kb	ISO 14443 A, ISO 15693, LEGIC RF	Da 106 a 848 kbps	3DES, DES, LEGIC encrypt, AES	123 o 127	Chiavi da 64 a 127 bit	CC EAL 4+

Tabella 2.5 Riepilogo Legic

2.6.5 EM Microelectronic

La EM Microelectronic offre un'ampia gamma di prodotti RFID, di tutte le frequenze, da 125 kHz a 2,4 GHz [52]. Per quanto riguarda le smart card contactless (e quindi 15,56 MHz) lo scopo di EM è quello di ottenere un hardware ottimizzato per il mercato di massa, così come creare prodotti ad alte prestazioni, compatibili anche con Java Card virtual

machine. I fattori come la memoria Flash configurabile in modo flessibile per gestire codice e dati, la programmazione veloce grazie ad un bootloader ottimizzato e la compatibilità con il famoso IDE *Keil uVision*, ha aggiunto a questi prodotti una universalità di utilizzo eccellente [53].

Hanno generalmente due serie di prodotti.

2.6.5.1 EM 4333 (Famiglia Security & Access Control)

Questa carta può operare sia come carta di prossimità che come carta di vicinità. Infatti è conforme sia all'ISO 14443 A che all'ISO 15693. Utilizza un algoritmo chiamato *Grain128a* per cifrare i dati con chiave a 128 bit per quanto riguarda la parte VICC[27]; per quanto riguarda invece la parte PICC sono utilizzati AES-128, DES/3DES, ed un coprocessore DMA[28].

- **Applicazioni:**
  - Controllo di accessi.
  - Trasporto pubblico.
- **Caratteristiche:**
  - Dual ISO (ISO/IEC 14443 A, ISO/IEC 15693)
  - 4 kb di user data, condivisa da entrambe le interfacce.
  - 64 kb per il sistema operativo.
  - 3 kb XRAM[29] e 256 byte iRAM[30]
  - Trasferimento dati 848 kbps
  - Clock interno 30 MHz
  - Crittografia AES, DES/3DES.
  - EEPROM 64 kb
  - Cicli scrittura 100.000
  - Mantenimento dati 20 anni

2.6.5.2 EM 4830 (Famiglia Thesus Gold)

È una smart card a doppia interfaccia, sia contact che contactless. In modalità contact è conforme all'ISO 7816, mentre in modalità contactless all'ISO 14443 A&B. Appartiene alla famiglia Thesus Gold.

- **Applicazioni:**
  - Trasporto pubblico.
  - Controllo di accesso.
  - Biglietteria automatica.
  - e-Government.
- **Caratteristiche:**
  - Doppia interfaccia, contact e contactless.
  - Conforme ad ISO 7816 e ISO 14443 A&B.
  - Trasferimento dati 848 kbps
  - Clock CPU 30 MHz.
  - Crittografia DES/3DES.

[27] Vicinity Integrated Circuit Card

[28] Direct Memory Access.

[29] eXtreme Fidelity RAM

[30] Memoria a stato solido prodotta dalla Gigabyte.

- Accesso rapido alla memoria con DMA.
- 2816 byte RAM
- 1 kb ROM
- GPNVM 132 kb
- Mantenimento dati 15 anni
- Durata cicli di lettura/scrittura circa 100.000.

EM	EEPROM	Standard	Velocità dati	RAM	Sicurezza	Cicli scrittura	Mantenimento dati	Altro
4333	64 kb	ISO 14443 A, ISO 15693	848 kbps	3 kb + 256 byte	AES, DES/3DES	100.000	20	
4830	132 kb (GPNVM)	ISO 7816, ISO 14443 A e B	848 kbps	2816 byte	DES/3DES	100.000	15	Implementazione DMA per accesso alla memoria

Tabella 2.6 Riepilogo EM

2.6.6 ST Microelectronics

La ST Microelectronics produce due famiglie di smart card, una a lungo raggio e un'altra a corto raggio [54]. La prima è conforme all'ISO 15693 e all'ISO 18000, ed è quindi una carta di vicinità. La seconda invece è conforme all'ISO 14443 tipo B, ed è quindi una carta di prossimità. Queste carte sono progettate per utilizzi industriali, come l'identificazione del personale, e per servizi in cui la contraffazione deve essere evitata.

2.6.6.1 Short Range Family

I prodotti *short range* utilizzando un trasponder integrato in una scheda di plastica, o di carta. La sicurezza delle transazione è garantita da un algoritmo anti-clonamento.

- **Applicazioni:**
  - Controllo accessi.
- **Caratteristiche:**
  - Conforme all'ISO 14443 B.
  - UID da 64 bit.
  - Sistema anti-collisione.
  - Due contatori a 32 bit.
  - Logica anti-tear automatica.
  - Conservazione dati di 40 anni.
  - User data da 2 a 512 kb.
  - EEPROM da 512 a 2048 bit.
  - Cicli lettura/scrittura di un milione.

ST	EEPROM	Standard	Velocità dati	UID	User data	Cicli scrittura	Crittografia	Mantenimento dati	Altro
Short Range Family	Da 512 a 2048 byte	ISO 14443 B	848 kbps	8 byte	Da 2 a 512 kb	1.000.000	DES, 3DES	40 anni	Sistema anti collisione e anti tear

Tabella 2.7 Riepilogo ST Micro

2.6.7 Atmel Corporation

La Atmel Corporation produce una linea di prodotti chiamata *Crypto RF*, ideali per tag RFID e smart card contactless che beneficiano di sicurezza avanzata [55].

2.6.7.1 Famiglia Crypto RF

Questi prodotti sono equipaggiati con un sistema crittografico a 64 bit. L'algoritmo crittografico codifica i dati ed i messaggi, e genera messaggi con codici di autenticazione, fornendo così un sistema sicura dove inserire i dati. Tutti i chip Crypto Memory sono compatibili con la piedinatura delle EEPROM seriali prodotte da Atmel.

- **Applicazioni**:
  - Controllo accessi.
  - Identità.
  - Tracciamento farmaceutico.
  - Carte di credito.
- **Caratteristiche:**
  - EEPROM da 4 a 64 kb
  - Mutua autenticazione a 64 bit.
  - Algoritmo di crittografia proprietario.
  - Sistema anti-tear.
  - Conforme all'ISO 14443 tipo B.
  - 2 kb di memoria di configurazione.
  - 100.000 cicli di lettura/scrittura
  - Conservazione dei dati di 10 anni

Atmel	EEPROM	Standard	User data	Cicli scrittura/ lettura	Manteni mento dati	Sicurezza	Altro
Crypto RF	Da 4 a 64 kb	ISO 14443 B	2 kb	100.000	10 anni	Algoritmo proprietario	Sistema anti tear

Tabella 2.8 Riepilogo Atmel.

2.6.8 HID Global

La HID Global è un'azienda con lo scopo di fornire prodotti per la sicurezza e l'identificazione dei soggetti [56]. Questa azienda produce carte ad alte prestazioni e comprende diverse famiglie tra cui la *iClass*, *iClass SE*, *pivClass*, e *Crescendo*.

2.5.8.1 Famiglia iClass e iClass SE

È una carta pensata per supportare diverse tecnologie garantendo la sicurezza dei dati. È possibile utilizzare la tecnologia Mifare Classic o Mifare DESFire EV1, e inoltre è disponibile anche con banda magnetica e HID proximity a 125 Hz. La versione *iClass SE*

migliora ulteriormente la sicurezza di questa carta grazie ad un strato di sicurezza aggiuntivo basato sulla piattaforma HID *Trusted Identity Platform.*

- **Applicazioni**:
  - Controllo accessi.
  - Identificazione sicura del personale.
  - Identificazione biometrica.
  - E-Cash
  - Trasporto pubblico.
- **Caratteristiche:**
  - Standard ISO 14443 B e A (con Mifare), ISO 15693
  - Crittografia 3DES
  - EEPROM 32 kb (1 kb o 4 kb Mifare Classic, 8 kb Mifare DESFire)
  - UID 4 o 7 byte
  - Tecnologia HID Proximity
  - Mantenimento dati 10 anni
  - Cicli di scrittura 100.000 o 500.000 (SE)
  - Multi applicazione fino a 28 applicazioni
  - Certificazione CC EAL 4+
  - Trasferimento dati 848 kbps.

2.6.8.2 Famiglia pivClass

È una smart card a doppia interfaccia conforme al FIPS 201 e soddisfa i requisiti del NIST sulla sicurezza. Si appoggia inoltre sullo stesso sistema della famiglia *SE*, ovvero la piattaforma *Trusted Identity Object.* La carta è realizzata con i migliori materiali per renderla duratura. Supporta gli standard ISO/IEC 7816, 14443 A e B e 15693. La velocità dati del protocollo di trasmissione contactless è di 848 kbps.

- **Applicazioni:**
  - Sistemi di controllo ad alta sicurezza.
  - Contesti governativi.
  - Accesso a sistemi informatici.
- **Caratteristiche:**
  - EEPROM 144 kb.
  - Mantenimento dati 20 anni.
  - Cicli di scrittura 500.000
  - Crittografia SHA, AES, RSA, ECC, DES e 3DES.
  - Trasferimento dati a 848 kbps.
  - Ambiente abilitato per JavaCard 3.0.1 e Global Platform 2.1.1.

HID	EEPROM	UID	Standard	Velocità dati	Mantenimento dati	Cicli scrittura/lettura	Sicurezza	Altro
iClass, iClass SE	32 kb (1 o 4 kb Mifare Classic o 8 kb Mifare DESFire)	4 o 7 byte	ISO 14443 B (e A con Mifare), ISO 15693	848 kps	10	100.000 o 500.000 (SE)	3DES	CC EAL 4+, Tecnologia HID Proximity
pivClass	144 kb	7 byte	ISO 14443 A e B, ISO 15693, ISO 7816	848 kbps	20	500.000	SHA, RSA, AES, ECC, DES e 3DES	Ambiente Java Card 3.0.1

Tabella 2. 9 Riepilogo HID

2.6.9 Giesecke & Devrient

Fondata nel 1852, G & D è un leader globale e innovatore d'avanguardia nella produzione di banconote, documenti di sicurezza e sistemi di identificazione [57]. Fornisce banche commerciali, imprese, operatori di rete, e le aziende di trasporto pubblico con hardware innovativo, software e servizi completi, e soluzioni end-to-end per applicazioni di sicurezza mobile.

I prodotti garantiscono l'autenticità delle banconote e dei documenti d'identità e sicurezza dei pagamenti elettronici, le transazioni on-line, e la comunicazione mobile.

Esistono diverse categorie di smart card prodotte dall'azienda tra cui *contactless paymant* e *java technology*, la maggior parte focalizzate sui pagamenti elettronici.

2.6.3.1 Convego Join

Progettato per i mercati di pagamenti avanzati, Convego Join è una famiglia di prodotti con una soluzione multi applicazione. Permette quindi alle banche di personalizzare ulteriormente le smart card così da ottemperare alle esigenze dei clienti. Questa gamma di prodotti supporta sia la tecnologia Java Card che quella Global Platform. È una piattaforma molto flessibile, in cui è possibile aggiungere varie applicazioni protette da algoritmi di sicurezza certificati. Grazie alla gestione di memoria ottimizzata e sofisticata è una delle implementazioni di Java Card più di successo.

- **Applicazioni:**
  - Pagamenti elettronici di ogni genere.
  - Autenticazione EMV.
  - Carta fedeltà.
  - Carta di memoria.
- **Caratteristiche:**
  - EEPROM da 8 a 80 kb
  - Java Card 2.2.1 e GP 2.0.1
  - Metodi di autenticazione supportati SDA e DDA.

2.6.3.2 Convego Element

Questo tipo di prodotto offre alle banche un mezzo per soddisfare qualsiasi tipo di cliente. Le specifiche di questa carta possono essere configurate per carte di credito, di debito e pre-pagate. Grazie a questa flessibilità, e anche alla conformità delle specifiche Visa e MasterCard, Convego Element soddisfa un ampio mercato

- **Applicazioni:**
  - Pagamenti elettronici.
  - E-ticket.
  - Carta fedeltà.
  - Controllo di accessi.
- **Caratteristiche:**
  - EEPROM da 4 a 12 kb.
  - Metodi di autenticazione supportati SDA e DDA.

2.6.3.3 SmartSign Card

Questa famiglia di carte permette un integrazione semplice grazie al supporto di standard aperti come Global Platform, Java Card e ISO. L'alta efficienza della carta fa sì che le applet provenienti da differenti produttori possano essere eseguite sulla stessa carta. La possibilità di installare nuove applicazioni anche dopo la produzione rende alla carta una caratteristica di aggiornamento semplice. La sicurezza implementata in questa carta la rende sicura e ne minimizza i rischi.

- **Applicazioni:**
  - Pagamenti elettronici.
  - Carte di memoria.
  - Carte fedeltà.
  - E-Ticket.
- **Caratteristiche:**
  - Chip prodotto dalla NXP (Philips)
  - EEPROM da 72 a 144 kb.
  - Standard: Java Card, Global Platform, ISO 7816, ISO 14443 A
  - Certificazione CC EAL 5+
  - Crittografia: RSA, AES, DSA, 3DES, ECDSA, ECDH, SHA, RNG.7

G & D	EEPROM	Standard	Sicurezza	Altro
Convego Join	Da 8 a 80 kb	ISO 14443	Auth. SDA e DDA	-
Convego Element	Da 4 a 12 kb	ISO 14443	Auth. SDA e DDA	-
Smartsign	Da 72 a 144	ISO 14443 A, ISO 7816	RSA, AES, DSA, 3DES, ECDSA, ECDH, SHA, RNG	JavaCard e GlobalPlatform, CC EAL 5+

Tabella 2.10 Riepilogo G&D

Capitolo III

3.1 Implementazione

Il lavoro di questa tesi si conclude con un progetto realizzato dopo aver analizzato gli argomenti dei capitoli precedenti.

Lo scopo del progetto è quello di sviluppare una soluzione per l'utilizzo di smart card contactless, specifici lettori RFID di vario genere e hardware Arduino. Il progetto è stato realizzato in due parti distinte, una riguardante la gestione delle smart card da parte dell'utente, e un'altra dalle azioni a seguito di una lettura da parte di una scheda Arduino.

3.2 Software e Hardware

Il principale ambiente di sviluppo integrato utilizzato è stato Microsoft Visual Studio 2012, ed il plugin Visual Micro per programmare il microcontrollore di Arduino. Non ho quindi utilizzato l'IDE ufficiale per Arduino, poiché come evidenziato nel capitolo 1, Visual Micro dispone di alcune comodità, come il completamento automatico ed il controllo di errore pre-compilazione. L'IDE ufficiale è stato utilizzato solo in fase preliminare, come primo approccio alla programmazione per Arduino.

I linguaggi di programmazione utilizzati sono stati Visual C# .Net, ed il linguaggio basato su Wiring per Arduino. Il sistema operativo su cui è stato sviluppato il progetto è Microsoft Windows 8 Professional. Sia il sistema operativo, che Visual Studio sono stati concessi in licenza d'uso attraverso l'Università degli Studi di Camerino [58].

Durante lo sviluppo del progetto sono stati utilizzati diversi lettori, e diverse smart card, così da comprendere le differenze tra i vari dispositivi, e non sviluppare un applicativo ad-hoc solo per certi tipi di lettori e smart card. Per quanto riguarda le smart card, inizialmente è stata presa di riferimento come caso di studio la carta Enjoy my UniCam. Per i lettori è da notare che esiste sul mercato un RFID Shield per Arduino, ma lo scopo del progetto era quello di utilizzare dei lettori già disponibili.

3.3 Carte utilizzate

Durante questo lavoro di testi è stato possibile lavorare con tre tipi di carte:

- La carta Raffaello.
- La carta Enjoy my-UniCam.
- La carta ON-Card.
- La carta Oyster.

3.3.1 Carta Raffaello

La carta Raffaello è una carta a microprocessore che aderisce allo standard CNS, la cui parte elettronica presente le stesse caratteristiche della Carta d'Identità Elettronica [59]. Le funzionalità di questa carta sono principalmente due:

- È uno strumento di identificazione in rete dotata di certificato di autenticazione.
- Ospita il servizio di firma digitale fornendo al titolare la possibilità di sottoscrivere documenti elettronici.

La carta Raffaello è conforme allo standard ISO/IEC 7816, anche per quanto riguarda la comunicazione con un sistema operativo (APDU). Per quanto riguarda la parte contactless, la carte è conforme allo standard ISO/IEC 14443 A, e la tecnologia installata è l'emulazione Mifare Classic 1k.

Sul fronte della carta sono riportate le seguenti informazioni:

- Logo Regione Marche.
- Scritte Carta Nazionale dei Servizi.
- Codice fiscale.
- Cognome e nome.
- Data di nascita e luogo.
- Provincia.
- Data di scadenza.

Sul retro sono presenti:

- Banda magnetica ad ossidi rigidi a tre tracce
- Codice a barre rappresentante il codice fiscale.

Figura 3.1 Carta Raffaello.

3.3.2 Carta Enjoy my UniCam

La carta Enjoy my UniCam è una carta di identificazione adottata dall'Università degli Studi di Camerino rilasciata dalla Banca Popolare di Ancona [60]. Con questa carta è possibile identificarsi presso ogni sede universitaria ed usufruire dei servizi dedicati allo

studente, come l'iscrizione agli esami, l'accesso ai laboratori e alle biblioteche, visualizzare il riepilogo della carriera universitaria e richiedere stage e tesi di laurea.

Inoltre è possibile pagare anche tutti i servizi universitari, come le tasse, le fotocopie e la mensa. Si possono ricevere anche gli accrediti per le borse di studio. La carta dispone anche di una firma digitale certificata, rilasciata dalla azienda Namirial [61]. Il produttore di questa carta è Oberthur. La carta appartiene anche al circolo MasterCard, con cui è possibile prelevare a costo zero in tutta Europa.

La carta è conforme agli standard ISO/IEC 7816 e ISO/IEC 14443 A. La parte contactless è un emulazione Mifare Classic 1k. La carta è certificata CC EAL 5+ e FIPS 140-2. I sistemi di crittografia comprendono gli algoritmi 3DES, AES, RSA ed Elliptic Curves. Inoltre la carta è conforme agli ambienti JavaCard e GlobalPlatform.

Sul fronte della carta sono riportate le seguenti informazioni:
- Logo Unicam, Namirial e MasterCard.
- Foto formato tessera del titolare.
- Codice identificativo della carta.
- Data di scadenza.
- Cognome e nome.

Sul retro sono presenti:
- Banda magnetica color argento.
- Codice seriale univoco ISO 14443 A.
- Codice IBAN.
- Codice PIN 3 caratteri.

Figura 3.2 Carta Enjoy.

3.3.3 Carta OnCard

La carta OnCard è una carta di identificazione rilasciata dalla Banca Marche in collaborazione con la Regione Marche ERSU. La carta dispone di circolo MasterCard con cui è possibile effettuare pagamenti online.

È conforme agli standard ISO/IEC 7816 e ISO/IEC 14443 A. Il chip contactless è prodotto dalla azienda Infineon ed emula Mifare Classic 1k.

Sul fronte della carta sono riportate le seguenti informazioni:
- Logo Regione Marche, Banca Marche e MasterCard.
- Codice identificativo della carta.
- Data di scadenza.

Sul retro sono presenti:
- Banda magnetica a ossidi rigidi.

- La firma del titolare.
- Il codice PIN 3 caratteri.
- Foto formato tessera del titolare.
- Cognome e nome.

Figura 3.3 Carta OnCard

Tutte le carte finora descritte sono dotate di emulazione Mifare Classic 1k, e supportano quindi lo standard ISO/IEC 14443 A.

3.3.4 Carta Oyster

La Oyster card è una contactless smartcard a radio frequenza, mostrata in figura 3.4, utilizzata per il trasporto pubblico nella città di Londra. Il sistema è gestito dalla TranSys, ed è basato sullo standard NXP/Philips – Mifare 1k, munito di chip fornito da Giesecke & Devrient e SchlumbergerSema. Supporta sia l'ISO 14443 A che l'ISO 14443 B. Nel 2008 nacque la moda di togliere il chip RFID dalla Oyster cards ed attaccarlo sull'orologio da polso o su di un braccialetto in modo di evitare di dover tirare fuori la card tutte le volte che la si doveva usare. Nonostante questo modo di procedere non avesse creato problemi nel funzionamento del servizio né evasioni nell'addebito dei trasporti, fu disapprovato questo uso della carta multando tutti coloro che riportavano la carta danneggiata non integra al momento della ricarica [62]. Sul fronte è riportato il nome della carta, e sul retro le condizioni d'uso ed un codice seriale.

Figura 3.4 Oyster card

3.4 Lettori RFID utilizzati:

Durante questo lavoro di tesi sono stati utilizzati i seguenti lettori:

- HID Omnikey 5321 CL USB
- ACS ACR120U-TK-R
- SCM Microsystems SCL010

3.4.1 HID Omnikey 5321 CL USB

Il lettore Omnikey 5321 è dotato di doppia interfaccia, quindi può leggere e scrivere sia smart card contact che contactless. La velocità di trasmissione dati massima è di 848 kbps. Supporta l'ISO/IEC 14443.

- **Smart card supportate** [63]**:**
  - Philips: Mifare, DESFire, ProX, Smart MX e ICode.
  - HID: iClass.
  - Texas Instruments: TagIT.
  - ST Micro: x-ident, SR 176, SR 1X 4k.
  - Infineon: my-d.
  - Atmel: AT088RF020.
  - KSW MicroTech: KSW TempSens.
  - JCOP in modalità RSA.

3.4.2 ACS ACR120U-TK-R

Il lettore ACR120U ha un interfaccia contactless per leggere e scrivere smart card. È un lettore compatto e supporta lo standard ISO/IEC 14443 A e B, e le carte Mifare Classic, DESFire e mini. La velocità di trasmissione dati è di 106 kbps e la distanza operativa è di cinque centimetri [64].

3.4.3 SCM Microsystems SCL010

L'SCL010 è un lettore in grado di supportare varie applicazioni non a contatto. E' adatto ad esempio per identità elettronica, sistemi di pagamento e trasporto pubblico. SCL010 è progettato per offrire nel miglior modo possibile l'interoperabilità con vari dispositivi che utilizzano la tecnologia RFID.

Il firmware di questo lettore è aggiornabile e questo garantisce un elevata compatibilità anche con standard futuri [65].

- **Standard supportati:**
  - ISO 14443 tipo A & B
  - MIFARE
  - PC/SC compatibile

3.5 Scelta di Arduino

Per quanto riguarda la parte di Arduino, la scheda scelta è stata Arduino UNO R3 e l'Ethernet Shield.

La scelta di Arduino UNO R3 è giustificata in quanto è una scheda semplice da usare, un punto di partenza per ogni programmatore che si avvicina a questo ambiente. Inoltre, il basso costo e la compatibilità con la maggior parte degli Shield disponibili sul mercato, la rende utile anche per scopi futuri. Lo Shield Ethernet è stato scelto per aggiungere una connettività di rete alla scheda base. L'ultima revisione di questo Shield introduce anche uno slot SD card, così da poter mantenere dei dati anche dal lato Arduino.

Inoltre sono stati utilizzati anche altri componenti oltre ad Arduino, come cavetti, led, relè, e breadboard. La breadboard serve per ospitare i led ed i relè, mentre i cavetti, chiamati anche jumper, servono per effettuare i collegamenti tra le uscite Arduino e gli altri componenti sulla breadboard. In particolare il relè servirà per comandare l'apertura e la chiusura di circuiti con tensioni differenti da Arduino, nel progetto sarà utilizzata una lampadina da 220 volt, per sottolineare come sia semplice poter lavorare con qualsiasi tipo di output (e input).

3.6 Scelte progettuali

Prima di rendere definitiva la linea di sviluppo di questa soluzione, sono state varate diverse possibilità di scelta progettuale. I punti chiave che hanno coinvolto queste scelte sono molteplici.

In primo luogo la comunicazione tra smart card e la scheda Arduino. Come prima ipotesi è stato scelto di acquistare un USB Host Shield, in maniera tale da collegare direttamente i lettori RFID alla scheda Arduino. Questa possibilità è stata scartata in quanto alcuni lettori utilizzano delle componenti DLL proprietarie, e quindi trasferire il codice su una libreria Arduino avrebbe occupato una quantità di tempo che va aldilà di quella disponibile. Questa opzione sarà considerata negli sviluppi futuri. I lettori RFID vengono quindi collegati ad un computer.

In secondo luogo la comunicazione tra l'applicativo utente e la scheda Arduino. Inizialmente è stata implementata una comunicazione seriale via USB. Questo scelta però non funzionava in quanto sarebbe stato necessario che la scheda Arduino si trovasse in

costante contatto con un computer. Inoltre la comunicazione seriale presenta alcuni problemi di integrità dei dati trasmessi se non opportunamente gestiti. L'acquisto dell'Ethernet Shield, e l'aggiunta della connettività di rete, è stata quindi la scelta effettuata per la comunicazione tra la parte gestionale, utente, e l'hardware Arduino. Oltre ai vantaggi in termini di collocazione della scheda, la comunicazione ethernet è risultata nettamente più affidabile.

Infine, poiché abbiamo a che fare con un applicativo in parte gestionale, è stato previsto l'utilizzo di un database. Il database è residente nel pc, in quanto continue modifiche e controlli avrebbero stressato notevolmente la scheda Arduino, e sicuramente allungato notevolmente il codice. Per gestire i dati salvati sono stati utilizzati dei semplici file binari, in quanto i record gestiti sono solamente di natura testuale.

Tutto il codice sarà disponibile in allegato alla tesi di studio.

3.7 Applicativo utente

L'applicativo, sviluppato con Microsoft Visual Studio 2012 in linguaggio Visual C# .Net è di carattere gestionale. La GUI è stata realizzata prettamente con Windows Form, che hanno permesso uno sviluppo rapido, semplice ed efficace.

È possibile gestire gli utenti, i lettori, le smart card e gli eventi. Per utente si intende solamente un nominativo che rappresenta appunto un utente all'interno del programma. Per quanto riguarda un lettore, una volta connesso alla macchina, è possibile registrarlo all'interno del database e associargli un nome mnemonico per un utilizzo più *user frendly*. Le smart card vengono associate, dopo aver registrato un utente, con uno e un solo utente. Gli eventi coprono la parte operativa, e scegliendo un lettore, un utente e una rispettiva smart card associata è possibile impostare quali azioni dovrà effettuare la scheda Arduino una volta indentificata una smart card.

Verranno ora elencate ed analizzate le funzioni messe a disposizione.

3.7.1 Interfaccia principale

All'avvio, l'applicativo controlla se è presente la configurazione delle opzioni nel database come nella figura 3.5. In caso di successo allora carica la configurazione, altrimenti

ne scrive una di default e poi la carica. La configurazione è salvata dentro un file chiamato *config.ini.*

Un esempio di configurazione è la seguente:

```
ip=193.205.92.200
port=8888
mask=255.255.255.0
gateway=192.168.1.1
debug=True
multiple=False
Porta aula CISCO
Ufficio 1
Portineria
Sala caffè
Luce aula Cisco
Luce ufficio 1
Luce portineria
Spegni tutto!!
```

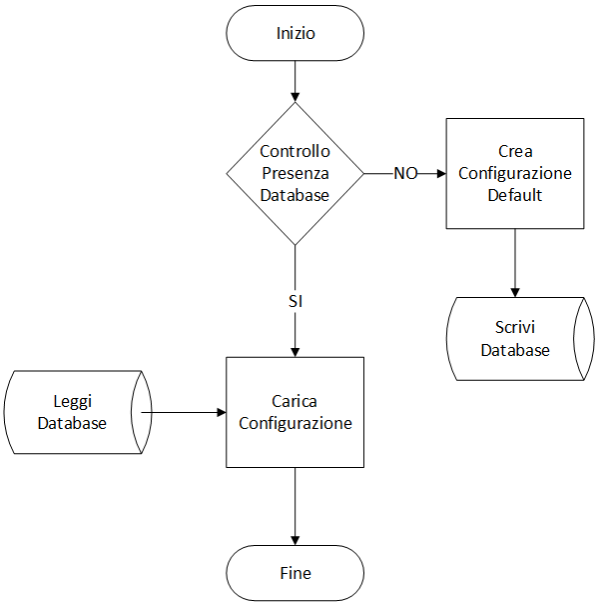

Figura 3.5 Caricamento configurazione opzioni.

A questo punto l'applicativo mostra l'interfaccia principale. Ora è in esecuzione la parte operativa, come mostrato nella figura 3.6, e ogni qualvolta passiamo una smart card associata ad un utente su un lettore registrato, se è previsto un evento, verrà comunicato ad Arduino quali azioni compiere.

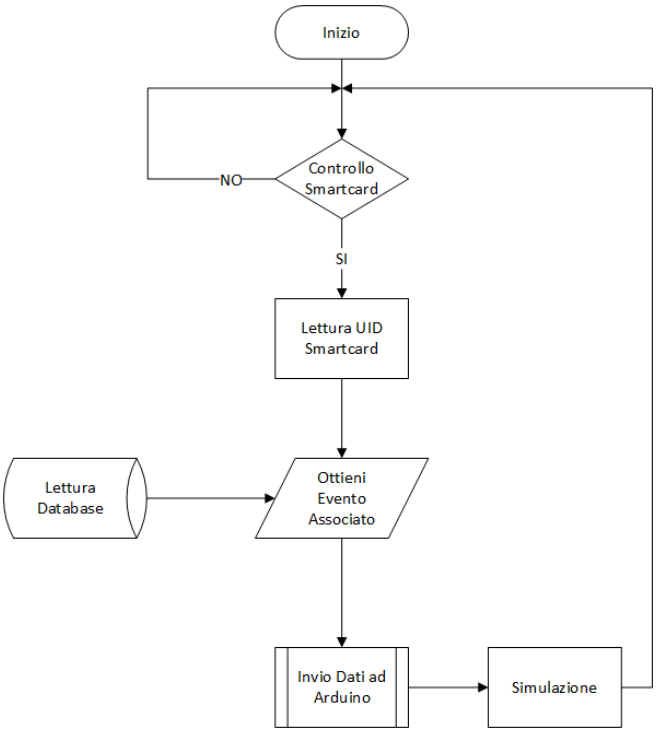

Figura 3.6 Parte operativa.

Fintanto che non vengono effettuate altre azioni, il programma non farà nient'altro che controllare se è presente una smart card o meno, su un lettore registrato connesso. Inoltre l'ultimo evento inviato ad Arduino sarà anche riproposto nell'interfaccia principale con un immagine simulativa. Per ottenere questa dinamicità sono stati utilizzati degli oggetti *timer*, uno per la lettura di eventuali smart card presenti su uno dei lettori, e altri per la simulazione delle azioni di Arduino.

La figura 3.7 mostra una schermata dell'interfaccia principale.

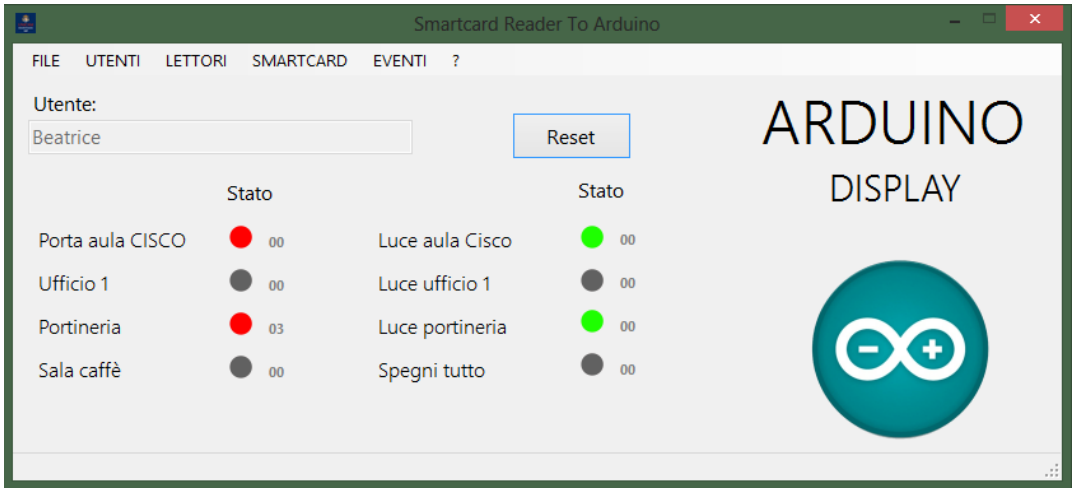

Figura 3.7 Interfaccia principale.

3.7.2 Menu file

Nel menu file ci sono tre opzioni disponibili:

- *Impostazioni*
- *Elimina database*
- *Esci*

Nelle impostazioni è possibile cambiare l'indirizzo IP, la subnet mask, l'indirizzo di gateway, e la porta, con sui stabilire la connessione con Arduino. Alla prima installazione del programma, su Arduino sarà impostato un indirizzo IP di default[31], quindi per operare la prima connessione col dispositivo è necessario che l'applicativo sia sulla stessa rete di Arduino. Una volta connesso è possibile variare l'indirizzo IP e gli altri parametri di connessione di Arduino direttamente da software. Arduino salva il proprio indirizzo IP, la subnet mask e l'indirizzo di gateway, sulla SD card. Un metodo per forzare l'utilizzo di un indirizzo piuttosto che un altro potrebbe essere quello di scrivere direttamente all'interno del file i parametri di connessione come:

```
192.168.1.177
255.255.255.0
192.168.1.1
```

È possibile inoltre abilitare o disabilitare le attività di debug e le modifiche multiple. Se entrambe attive, per le attività di debug verrà mostrato sull'interfaccia principale un log delle azioni eseguite dal programma, ogni qualvolta verrà modificato il database, e ogni qualvolta verranno comunicati i dati ad Arduino; per le modifiche multiple invece, una volta selezionata la rispettiva opzione, potremmo aggiungere, modificare o eliminare gli elementi del database in modo continuato.

Infine è possibile modificare i nomi associati alle uscite su Arduino (i.e. se vogliamo che la prima uscita identifichi l'apertura di una porta, allora potremmo chiamarla "Apri Porta"). Le impostazioni una volta modificate saranno salvate su un file di configurazione, così da essere ripristinate dopo una eventuale chiusura ed avvio del programma.

Elimina database non fa nient'altro che eliminare tutti i database locali, e quindi una volta confermata l'operazione tutti i dati precedentemente salvati andranno persi.

Cliccando sul menu Esci, l'applicazione sarà terminata.

[31] 192.168.1.3

3.7.3 Menu utenti

Nel menu utenti ci sono quattro opzioni disponibili:

- *Aggiungi utente*
- *Modifica utente*
- *Elimina utente*
- *Visualizza*

Rispettivamente a quale menu cliccheremo sarà possibile aggiungere un nuovo utente (in figura 3.8), modificare o eliminare un utente esistente o visualizzarli tutti.

Non possibile aggiungere due utenti con lo stesso nominativo. Nel momento in cui sarà modificato un utente, tutti le smart card associate e gli eventi associati saranno modificati di conseguenza così da mantenere il database integro. Stessa cosa per l'eliminazione per cui tutte le smart card associate e gli eventi associati saranno eliminati di conseguenza.

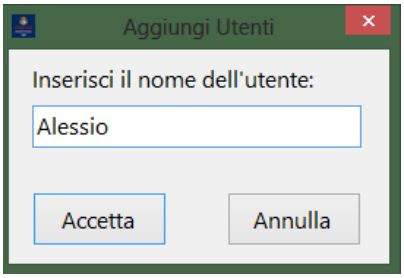

Figura 3.8 Aggiungi utenti.

3.7.4 Menu smart card

Nel menu smart card ci sono due opzioni disponibili:

- *Aggiungi smart card*
- *Elimina smart card*

La prima opzione ci permette di associare una smart card ad un utente esistente, in figura 3.9. Per leggere il numero seriale univoco della smart card, una volta aperta la finestra, sarà necessario solamente passarla su un lettore connesso.

La seconda opzione ci permette di eliminare una smart card associata ad un utente. Non è possibile associare la stessa smart card a più utenti, mentre un utente può comunque avere più smart card associate. Nel momento in cui viene eliminata una smart card, tutti gli eventi associati a quella smart card verranno eliminati di conseguenza così da mantenere il database integro.

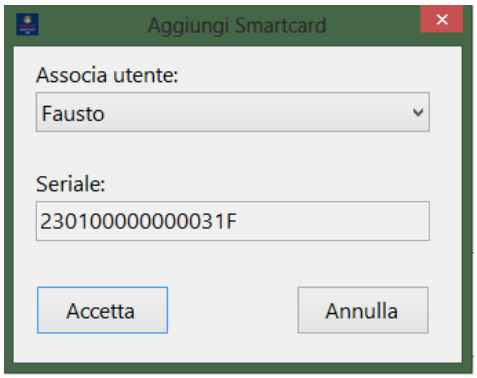

Figura 3.9 Aggiungi Smartcard

3.7.5 Menu lettori

Nel menu lettori ci sono tre opzioni disponibili:

- *Aggiungi lettore*
- *Modifica lettore*
- *Elimina lettore*

Per registrare un lettore è necessario aggiungerlo all'interno del database selezionando la prima opzione, mostrato in figura 3.10. Una volta aperta la finestra verranno interrogati tutti i lettori connessi alla macchina. Una volta inserito il nome mnemonico con cui far riferimento al lettore la registrazione sarà conclusa.

La modifica e l'eliminazione di un lettore comporteranno la rispettiva modifica o eliminazione degli eventi associati a quel lettore.

Non è possibile associare più di un nome mnemonico allo stesso lettore.

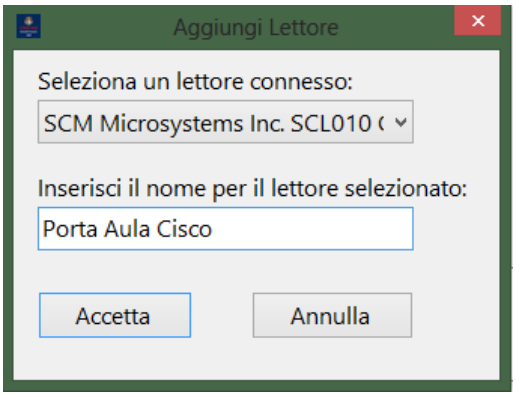

Figura 3.10 Aggiungi lettore.

3.7.6 Menu eventi

Nel menu eventi ci sono due opzioni disponibili:

- *Aggiungi evento*
- *Modifica evento*
- *Elimina evento*

La prima opzione ci permette di aggiungere un evento, in figura 3.11, e quindi associare un utente, una smart card ed un lettore ad una azione che dovrà compiere Arduino. Innanzitutto è necessario scegliere un lettore registrato, e di seguito un utente registrato con una relativa smart card associata. A questo punto bisogna selezionare quali uscite Arduino dovrà attivare e quali disattivare, selezionando opzionalmente il tempo di attivazione di un uscita. Compilato il form, sarà possibile inserire l'evento nel database. La prossima volta che nella schermata principale passeremo una smart card su un lettore associato ad un evento, se l'evento è previsto, Arduino eseguirà l'azione programmata.

Non è possibile associare più eventi alla stessa smart card con lo stesso lettore.

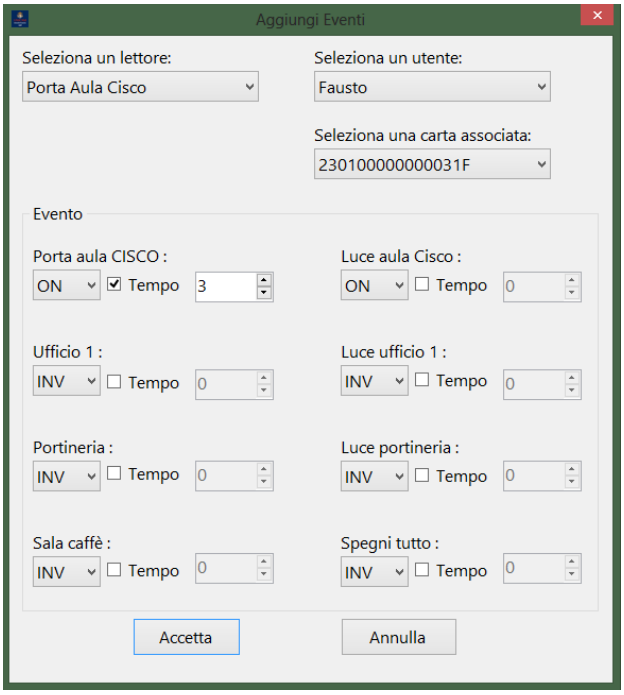

Figura 3.11 Aggiungi evento.

La seconda opzione ci permette di modificare un evento. Viene proposta una schermata in cui si seleziona l'evento in termini di utente, smart card e lettore. Successivamente si apre una schermata come quella in figura 3.11, solamente che è popolata con i dati dell'evento. A questo punto sarà possibile modificare l'evento.

La terza opzione di permette di scegliere una associazione utente, smart card e lettore per eliminarla.

3.7.7 Menu about

Nel menu about è possibile consultare informazioni e contatti di questo progetto.

3.8 Programma Arduino

Arduino segue una logica differente dai normali programmi scritti in linguaggi come C# o C++. Come introdotto nel capito 1, il codice di programma Arduino si sviluppa in due parti esecutive: la fase di *setup* e la fase di *loop*.

3.8.1 Fase di setup

Nella fase di setup, come mostrato nella figura 3.6, per prima cosa vengono impostati i pin di uscita, ovvero quelli che dovranno attivare o disattivare altri componenti, come led o relay. Successivamente viene avviato il servizio per utilizzare la SD card e quindi leggere i dati salvati per l'indirizzo IP, la subnet mask e l'indirizzo di Gateway. Una volta manipolati i dati letti vengono istanziati oggetti di tipo IPAddress con i valori letti. A questo punto viene avviato il servizio per utilizzare la connettività Ethernet. Il flag *verbose_debug*, una volta asserito e riprogrammata la scheda, permette di visualizzare su monitor seriale il debug della fase di setup.

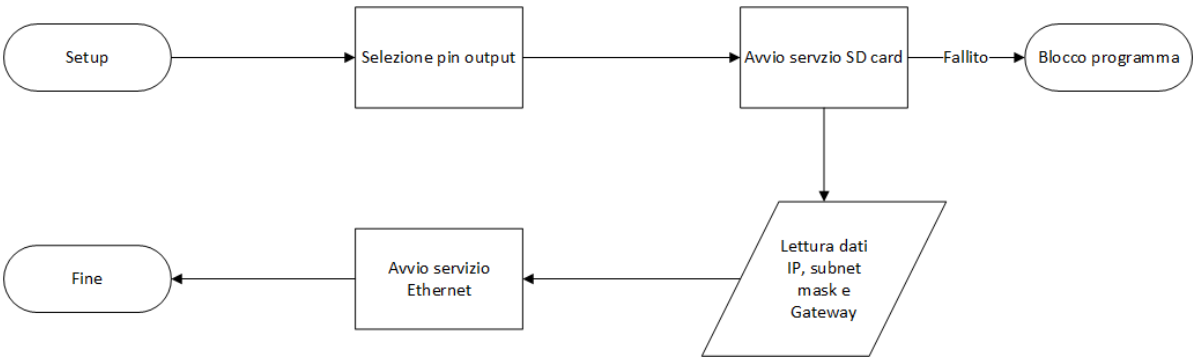

Figura 3.12 Fase di setup Arduino.

3.8.2 Fase di loop

La fase di loop, mostrata nella figura 3.7, è quella che viene mantenuta fintanto che Arduino è alimentato. Le principali azioni che Arduino svolge sono due, quella di ascoltare l'arrivo di pacchetti sull'Ethernet, e quella di controllare le uscite attive temporizzate. Ad ogni iterazione di loop, viene controllato se un client è collegato sull'indirizzo IP di Arduino e se il client è disponibile ad inviare dati, allora vengono letti. A questo punto le possibilità di lettura sono due:

- Dati *Azione*.
- Dati *Impostazioni*.

La prima opzione è quella in cui i dati inviati dal client corrispondono ad una azione che Arduino deve compiere. Il flusso di dati letto incomincia con la stringa *"AC*"*, ed una volta acquisite tutte le informazioni, saranno processati e determinate quali uscite dovranno essere attivate (output *HIGH*) o disattivate (output *LOW*), e per quanto tempo.

La stringa dati letta da Arduino segue questa grammatica:

```
S→ D '*' U⁸ '#'
D→ "IP" | "AC"
U→ STU | ST
S→ 'T'C² | 'F'C² | 'N'
C→ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Una possibile stringa può essere:

```
AC*T20T00N00N00T15F20F00T80#
```

Il significato è che Arduino deve compiere un azione, il cui output rispettivo per ogni PIN sarà *HIGH* in corrispondenza di una T, *LOW* in corrispondenza di una F, e rimarrà invariato in corrispondenza di un N. I numeri specificano per quanto tempo il PIN deve mantenere quel tipo di output prima di commutare. Per azioni di tipo N, i numeri non sono considerati. Con Arduino UNO, per *HIGH* si intende una tensione di +5 volt, e per *LOW* una tensione di +0 volt. Gli output su questi PIN andranno poi collegati ad altri componenti, nel nostro caso LED e relè. In questo modo è possibile simulare il comportamento di Arduino collegato ad altri sistemi (i.e. l'apertura di una porta corrisponderà all'accensione e spegnimento rapido di un led).

La seconda opzione impone ad Arduino di cambiare i suoi parametri di connessione. Il flusso di dati letto incomincia con la stringa *"IP*"*, e una volta processati saranno salvati, al posto di quelli presenti, nella SD card. A questo punto, poiché il servizio Ethernet viene avviato nella fase di setup, Arduino sarà resettato, così da avviarsi con i nuovi parametri di

connessione. Il riavvio è un metodo forzato che non è nient'altro che l'indirizzo zero, e una volta invocato Arduino eseguirà il riavvio.

Una possibile stringa può essere:

```
IP*192.168.1.177&255.255.255.0&192.168.1.1#
```

Nel caso in cui nessun client è disponibile, il programma continua con il controllo delle uscite temporizzate. Grazie alla funzione *millis()* è possibile conoscere il tempo di esecuzione passato dall'avvio di Arduino. Nel momento in cui viene attivata un'uscita temporizzata si tiene conto del tempo di attivazione stabilito dall'azione sommato al tempo di esecuzione passato (*millis*). Questo valore in riferimento al corrente tempo di esecuzione passato è in grado di determinare se l'uscita ha terminato il suo tempo. Le uscite non temporizzate non saranno ovviamente soggette a questi controlli.

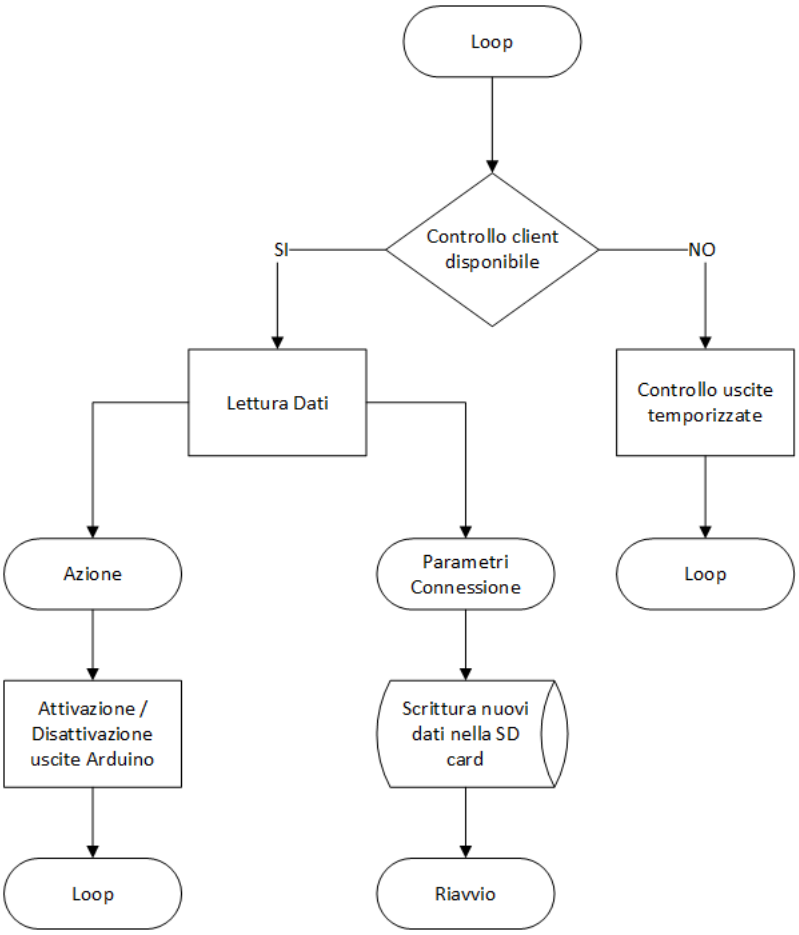

Figura 3.13 Fase di loop Arduino.

In figura 3.14 è possibile vedere Arduino collegato ad una basetta breadboard in azione.

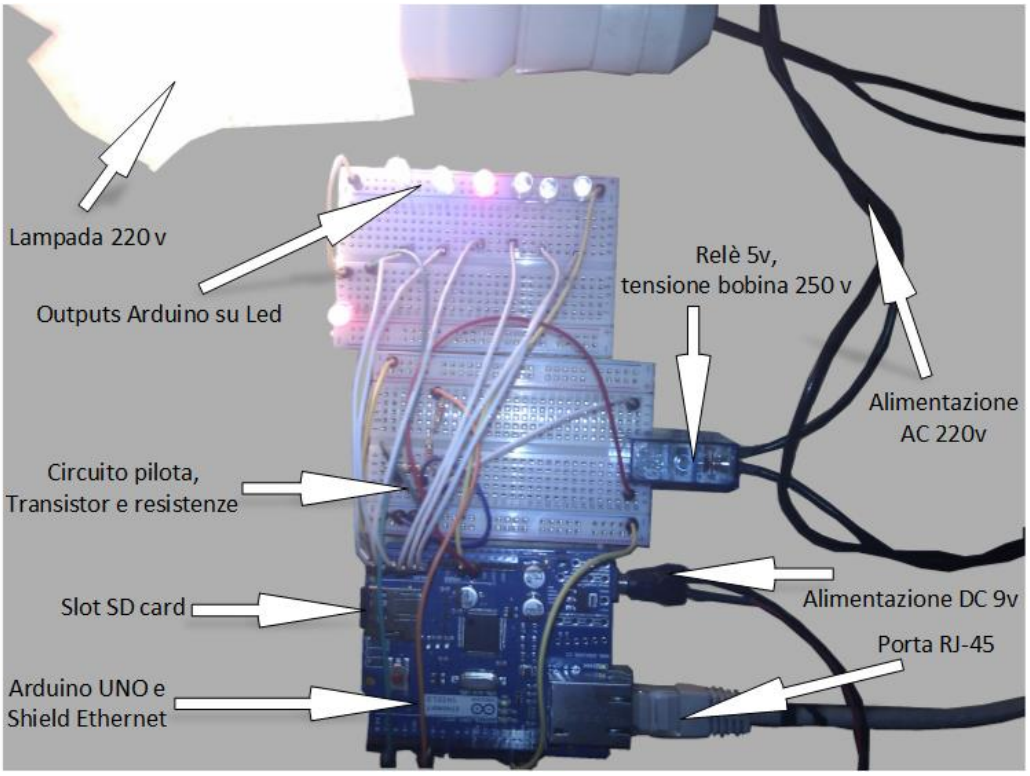

Figura 3.14 Arduino collegato.

Arduino e l'Ethernet Shield sono alimentati da una batteria a 9 v, o da un alimentatore esterno. Per quanto riguarda gli output su led, vengono direttamente alimentati dalle uscite di Arduino che erogano 5 volt. Per una lampada da 220 volt è necessario l'utilizzo di un relè, che viene comandato da 5 volt, e sui contatti invece corre una tensione di 220 volt. Un ulteriore accorgimento deve essere fatto per il relè. Siccome un uscita di Arduino può erogare al massimo circa 40 mA ed il relè ne assorbe circa 100 mA (dipende dalle specifiche), è necessario utilizzare un circuito pilota costituito da un transistor (in questo caso un BC547) e alcune resistenze (calcolare in base all'assorbimento del relè). Il transistor amplifica il segnale in entrata e la corrente sul relè è quella necessaria al corretto funzionamento. Solitamente viene anche posto un diodo in parallelo con i comandi del relè per eliminare sovratensioni generate dalla bobina all'apertura del contatto [66].

Sviluppi futuri

Gli sviluppi futuri di questa applicazione sono molteplici. Attualmente ci sono diverse limitazioni imposte, che una volta risolte possono aumentare notevolmente il livello produttivo del lavoro svolto.

Uno degli aspetti che potrebbero essere integrati è la possibilità di collegare direttamente un lettore RFID ad Arduino. Questo implicherebbe innanzitutto la necessità di realizzare delle librerie ad-hoc per il microcontrollore installato su Arduino. Una volta compresa questa possibilità sarà necessario spostare il database dell'applicativo in web server. In questo modo Arduino potrà interrogare il web server, e una volta ottenuto il seriale univoco dal lettore collegato ad esso, ottenere dal web server l'azione da compiere. Come scelta progettuale sconsiglio di spostare invece il database interamente nella scheda SD di Arduino, poiché l'elevato numero di processi sui dati non solo potrebbe recare uno scompenso nelle funzionalità di Arduino, ma sicuramente introdurrà una difficoltà nell'ottenere i tempi di esecuzione e quindi nel gestire le uscite temporizzate. Inoltre la possibilità di lavorare con un web server potrebbe permettere a più schede Arduino di interoperare senza disturbi con il medesimo database.

Un altro aspetto da implementare, anche senza la necessità di un web server potrebbe essere quello di aggiungere una *white list* di indirizzi IP, così da permettere la comunicazione verso Arduino solamente da certi IP. Una possibilità di implementazione potrebbe essere quella di creare un file all'interno della SD di Arduino con l'elenco di indirizzi IP permessi. In questo modo, ogni qualvolta un client si connette, Arduino controllerà prima se l'IP ha il permesso di inviare comandi, e all'occasione respingere la richiesta di comunicazione.

Sempre nel caso in cui il database sia spostato su un web server, anche per quanto riguarda l'applicativo è possibile prevedere uno sviluppo ulteriore, e cioè quello di realizzare una applicazione web. In questo modo sarà possibile programmare gli eventi, aggiungere o rimuovere dati, da qualsiasi postazione, senza la necessità di installare un programma.

Per quanto riguarda invece il numero totale di uscite sulla scheda Arduino, questo dipende essenzialmente dalla scheda scelta. Nel caso in cui si voglia variare il numero di uscite, la scelta più veloce è quello di cambiare il codice. Per rendere il processo automatico si potrebbe inserire nella SD un ulteriore file di configurazione recante quante uscite dover

gestire e per quali pin. A questo punto Arduino, durante la fase di setup, potrebbe controllare questo file e impostare i pin di conseguenza.

Un'altra possibilità per integrare con molta facilità un lettore RFID in una scheda Arduino è quello di acquistare uno Shield RFID. In questo modo la lettura delle schede sarà notevolmente avvantaggiata in quanto sono disponibili diverse librerie per leggere le schede più comuni. Unico possibile limite riguarda quali standard lo Shield è in grado si supportare.

Conclusioni

Gli obiettivi raggiunti sono molti. Le smart card dominano nel contesto tecnologico per l'identificazione. Oggigiorno sono innumerevoli le applicazioni che si possono attuare con queste carte. Nel corso della tesi ho potuto notare come queste tecnologie sono supportate da alti standard qualitativi, e di come le aziende leader del settore hanno introdotto nuovi aspetti e nuove tecnologie, soprattutto per quanto riguarda la sicurezza, anche perché questi dispositivi devono affidarsi totalmente a sistemi sicuri ed algoritmi crittografici efficienti. Con il passare del tempo, la tecnologia contactless ha preso il dominio, e tutt'ora le soluzioni richieste sono sempre orientate a tecnologie wireless. Inoltre le smart card sono utilizzate sempre più per ambienti multi funzionali, dove è possibile identificarsi con una sola carta, ed utilizzarla per una diversità, sempre maggiore, di applicazioni. Traggo l'esempio della smart card Enjoy adottata dall'Università degli Studi di Camerino. Con essa lo studente trae moltissimi vantaggi e comodità, come il pagamento delle tasse, l'iscrizione agli esami, e l'accesso ai laboratori, tutto con una sola smart card.

Come parere personale, l'aspetto più interessante, lo ha ricoperto Arduino. Questa piattaforma di piccole dimensioni, ma di esagerata flessibilità, permette veramente di realizzare qualsiasi cosa. In breve tempo sono entrato a conoscenza delle meccaniche e del linguaggio di programmazione per questa piattaforma. Un volta compreso il comportamento, realizzare la parte del progetto che coinvolgeva questo dispositivo è stato semplice e divertente, ma soprattutto istruttivo.

Ringraziamenti

Il primo ringraziamento è per la mia famiglia, quella che mi ha cresciuto. In particolare ringrazio mio padre e mia madre senza i quali non avrei potuto gioire di questa fantastica vita.
Li ringrazio per tutti i sacrifici che hanno fatto, e soprattutto, per non avermi mai ostacolato in ogni mia decisione.
È a loro che dedico il mio percorso di studi.

Ringrazio il mio grande amore, che mi è stata sempre a fianco, e nonostante lei creda che io non ne abbia bisogno, e stato grazie a lei che ho superato molte sfide.
Ringrazio tutti i miei amici, quelli con cui sono cresciuto, e quelli che ho conosciuto durante questo percorso di studi. Chi positivamente e chi negativamente, hanno comunque contribuito alla mia crescita e alla felicità di tutti i giorni.

Ringrazio tutti i professori ed i docenti che amano il proprio mestiere, e che mi hanno trasmesso la passione dell'informatica. Senza di loro non sarei qui. Un ringraziamento speciale va al mio relatore di tesi per tutta la disponibilità e per l'impegno che ogni giorno ripone nella sua docenza.

Infine ringrazio me stesso, per tutto l'impegno e la passione che ho riposto in questa disciplina.

Grazie.

SITOGRAFIA

[1] CREATIVE COMMONS, «CC A-SA 2.5»
http://creativecommons.org/licenses/by-sa/2.5/it/legalcode

[2] SITO UFFICIALE ARDUINO, «HOME PAGE»
http://www.arduino.cc/

[3] WIRING, «HOME PAGE»
http://wiring.org.co/

[4] PROCESSING, «HOME PAGE»
http://www.processing.org/

[5] GNU, «LICENZA GPL V2»
http://www.gnu.org/licenses/gpl-2.0.html

[6] SITO UFFICIALE ARDUINO, «INTRODUZIONE»
http://arduino.cc/en/Guide/Introduction

[7] ATMEL, «HOME PAGE»
http://www.atmel.com/

[8] SITO UFFICIALE ARDUINO, «MEMORIA»
http://arduino.cc/en/Tutorial/Memory

[9] SITO UFFICIALE ARDUINO, «PROGMEM»
http://www.arduino.cc/en/Reference/PROGMEM

[10] SITO UFFICIALE ARDUINO, «PIN INPUT ANALOGICI»
http://arduino.cc/en/Tutorial/AnalogInputPins

[11] PLAYGROUND ARDUINO, «LISTA SCHEDE»
http://playground.arduino.cc/Main/SimilarBoards

[12] ATMEL, «DATASHEET ATMEGA328»
http://www.atmel.com/Images/doc8161.pdf

[13] SITO UFFICIALE ARDUINO, «ARDUINO UNO»
http://arduino.cc/en/Main/arduinoBoardUno

[14] ATMEL, «DATASHEET ATMEGA328»
http://www.atmel.com/Images/doc8161.pdf

[15] EMBEDDED, «WATCHDOG TIMERS»
http://www.embedded.com/electronics-blogs/beginner-s-corner/4023849/Introduction-to-Watchdog-Timers

[16] WIZNET, «DATASHEET WIZNET W5100»
http://www.wiznet.co.kr/Sub_Modules/en/product/Product_Detail.asp?cate1=5&cate2=7&cate3=26&pid=1011

[17] SITO UFFICIALE ARDUINO, «ETHERNET SHIELD»
http://arduino.cc/en/Main/ArduinoEthernetShield

[18] SITO UFFICIALE ARDUINO, «RIFERIMENTI»
http://arduino.cc/en/Reference/HomePage

[19] SITO UFFICIALE ARDUINO, «WINDOWS 8 DRIVER»
http://arduino.cc/forum/index.php?topic=94651.15

[20] SITO UFFICIALE ARDUINO, «WINDOWS DRIVERS»
http://arduino.cc/en/Guide/windows#toc4

[21] VISUAL MICRO, «PAGINA PRINCIPALE»
http://www.visualmicro.com/

[22] RFID.NET, «COS'È RFID»
http://rfid.net/basics/190-what-is-rfid

[23] GOOGLE PATENTS, «BREVETTO RFID»
http://www.google.com/patents?vid=4384288

[24] ELECTRONICS, «RFID»
http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/rfid3.htm

[25] THEIET, «RFID»
http://www.theiet.org/factfiles/it/rfid-page.cfm

[26] SMARTCARDBASIC, «TIPOLOGIE SMARTCARD»
http://www.smartcardbasics.com/smart-card-types.html

[27] NXP SEMICONDUCTORS, «HOME PAGE»
http://www.nxp.com/

[28] ENGADGET, «NXP GESTURE SMART CARD»
http://www.engadget.com/2012/01/11/nxp-gesture-smart-card-nfc/

[29] D. CHAUHAN, «APDU - PRATICA»
http://www.devshed.com/c/a/Practices/Smart-Cards-An-Introduction/5/

[30] J. SIMON, «SMARTCARD LIFECYCLE»
http://www.jguru.com/faq/view.jsp?EID=471391

[31] IBM, «SMART CARD RED BOOK»
http://www.redbooks.ibm.com/redbooks/pdfs/sg245239.pdf

[32] ISO, «HOME PAGE»
www.iso.org/

[33] INNOVATRON, «CALYPSO SPECIFICHE»
http://www.innovatron.fr/CalypsoFuncSpecification.pdf

[34] ICAO, «DOCUMENTO 9303»
http://www.icao.int/Security/mrtd/Pages/Document9303.aspx

[35] NIST, «SMARTCARD FIPS»
http://csrc.nist.gov/groups/SNS/smartcard/

[36] EMVCO, «ABOUT»
http://www.emvco.com/about_emvco.aspx

[37] PC/SC WORKGROUP, «OVERVIEW»
http://www.pcscworkgroup.com/overview/

[38] GLOBALPLATFORM, «ABOUT MISSION»
http://www.globalplatform.org/aboutusmission.asp

[39] ORACLE, «JAVACARD»
http://www.oracle.com/technetwork/java/javame/javacard

[40] ORACLE, «JAVACARD»
http://www.oracle.com/technetwork/java/javame/javacard/overview/getstarted/index.html?origref=http://www.oracle.com/technetwork/java/javame/javacard/overview/getstarted/index.html

[41] ORACLE, «JAVACARD OVERVIEW»
http://www.oracle.com/technetwork/java/javame/javacard/overview/about/index.html

[42] ORACLE, «JAVA CARD ARCHITETTURA»
http://www.oracle.com/technetwork/java/javacard/javacard3-138837.html

[43] NXP, «MIFARE»
http://www.mifare.net/

[44] NXP, «MIFARE SMARTCARD»
http://www.mifare.net/products/mifare-smartcard-ic-s/

[45] NXP, «MIFARE CERTIFICATION INSTITUTE»
http://www.mifare.net/company-information-root/ait-mifare-certification-institute/

[46] CRYPTOGRAPHY RESEARCH INC., «HOME PAGE»
http://www.cryptography.com/

[47] CONTACTLESS WORLD, «MIFARE EMULATION»
http://contactless-world.com/mifare-emulation/

[48] INFINEON, «HOME PAGE»
http://www.infineon.com/

[49] INFINEON, «SECURITY CONTROLLER CONTACTLESS»
http://www.infineon.com/cms/en/product/chip-card-and-security-ics/security-controller-contactless/channel.html?channel=db3a30432662379201266f68f2bb442f

[50] LEGIC, «OVERVIEW»
http://www.legic.com/product_overview.html

[51] LEGIC, «AGGIORNAMENTO ADVANT»
http://www.legic.com/it/upgrade_su_legic_advant.html

[52] EM MICROELECTRONIC, «HOME PAGE»
http://www.emmicroelectronic.com/

[53] KEIL, «KEIL UVISION»
http://www.keil.com/uvision/

[54] ST MICROELECTRONICS, «HOME PAGE»
http://www.st.com/

[55] ATMEL, «CRYTPOMEMORY»
http://www.atmel.com/products/other/securemem/default.aspx

[56] HID, «HOME PAGE»
http://www.hidglobal.com/

[57] G & D, «HOME PAGE»
http://www.gi-de.com/en/index.jsp

[58] MICROSOFT, «DREAMSPARK»
https://www.dreamspark.com/

[59] REGIONE MARCHE, «CARTA RAFFAELLO»
http://www.cartaraffaello.it/

[60] UBI, «ENJOY MY UNICAM»
http://www.bpa.it/page/enjoy-my-unicam

[61] NAMIRIAL, «SICUREZZA INFORMATICA»
http://www.namirial.com/servizi/sicurezza_informatica/

[62] OYSTER, «OYSTER ABOUT»
http://www.tfl.gov.uk/tickets/14836.aspx

[63] HID, «OMNIKEY 5321»
https://www.hidglobal.com/products/readers/omnikey/5321

[64] ACS, «ACR 120 U»
http://www.acr120u.com/

[65] IDENTIVE INFRASTRUCTURE, «SCL010»
http://www.identive-infrastructure.com/en/products-solutions/smart-card-readers-a-terminals/contactless-readers/scl010

[66] LOGICA PROGRAMMABILE, «PILOTARE UN RELAY»
http://www.logicaprogrammabile.it/come-pilotare-rele/

INDICE DELLE FIGURE

CAPITOLO 1

Figura 1.1 Fronte e retro scheda Arduino UNO Rev. 3 10
Figura 1.2 Microcontrollore ATmega328 e piedinatura 11
Figura 1.3 Memoria di programma (Flash memory). 12
Figura 1.4 Connettore POWER. 13
Figura 1.5 Connettori digitali. 14
Figura 1.6 Connettori analogici 14
Figura 1.7 Fronte e retro Ethernet Shield Rev.3 15
Figura 1.8 Indicatori LED presenti sull'Ethernet Shield. 16
Figura 1.9 Quick toolbar Arduino IDE. 17
Figura 1.10 Ciclo di esecuzione Arduino. 18

CAPITOLO 2

Figura 2.1 Gerarchia di classi RFID. 21
Figura 2.2 Smart card. 24
Figura 2.3 Componenti smart card. 25
Figura 2.4 Pinout smart card contact. 26
Figura 2.5 Ciclo di vita smart card. 28
Figura 2.6 File system smart card. 29
Figura 2.7 Smart card standard 30
Figura 2.8 JavaCard Architettura applicazione 37

CAPITOLO 3

Figura 3.1 Carta Raffaello 58
Figura 3.2 Carta Enjoy. 59
Figura 3.3 Carta OnCard 60
Figura 3.4 Oyster card 60
Figura 3.5 Caricamento configurazione opzioni. 64
Figura 3.6 Parte operativa. 65
Figura 3.7 Interfaccia principale. 65
Figura 3.8 Aggiungi utenti. 67
Figura 3.9 Aggiungi Smartcard 68
Figura 3.10 Aggiungi lettore. 68
Figura 3.11 Aggiungi evento. 69
Figura 3.12 Fase di setup Arduino 70
Figura 3.13 Fase di loop Arduino. 72
Figura 3.14 Arduino collegato. 73

INDICE DELLE TABELLE

CAPITOLO 1

Tabella 1.1 Quantità di memoria schede Arduino. 7
Tabella 1.2 Caratteristiche principali delle schede Arduino. 9

CAPITOLO 2

Tabella 2.1 Radio frequenze utili. 23
Tabella 2.2 Comandi e risposte APDU. 27
Tabella 2.3 Riepilogo Mifare 46
Tabella 2.4 Riepilogo Infineon 49
Tabella 2.5 Riepilogo Legic 50
Tabella 2.6 Riepilogo EM 52

TABELLA 2.7 RIEPILOGO ST MICRO 52
TABELLA 2.8 RIEPILOGO ATMEL. 53
TABELLA 2. 9 RIEPILOGO HID 54
TABELLA 2.10 RIEPILOGO G&D 56