UNIVERSITA' DEGLI STUDI DI MODENA
E REGGIO EMILIA

Corso di Laurea in
Ingegneria Gestionale

Tesi di Laurea

Implementazione di lettura e traduzione file DXF in istruzioni operabili tramite PLC a due assi

Relatore: Cesare Fantuzzi **Candidato: Matteo Martinelli**

A. A. 2014/2015

Sommario

L’approccio al problema nasce dalla necessità di poter istruire facilmente un PLC rispetto a un percorso da seguire, in quanto fino ad ora non sono stati implementati metodi grafici per la programmazione dello stesso. Il PLC allo stato finale sarà inoltre dotato di un ugello per la distribuzione per la colla a caldo, in modo da poter essere utilizzato nel campo del “packaging”: il software finale dovrà essere in grado di dare un input a tale ugello, seppur non fornito insieme al modello da studiare. Il percorso sarà rappresentato da un file DXF, da qui la necessità di analizzarlo, estrapolarne i dati necessari, processarli opportunamente ed infine scriverli nella Shared Memory del PLC dato. Questa soluzione permette una facile e rapida programmazione del PLC da parte di un qualsiasi utente, senza necessità di conoscere la struttura logica del PLC stesso. Tra le norme di buon progetto vi è anche la semplicità d’uso del prodotto finale, motivo per il quale si è scelto di implementare alla fine del progetto un’intuitiva interfaccia grafica.

Il software finale avrà il nome “RWSharedMemory”, con estensione dei file “.rwsm”.

Indice

1 Introduzione

In questa tesi viene trattata l'implementazione di un software in grado di analizzare un file grafico di tipo DXF (*Drawing Exchange Format*) in due dimensioni redatto da un'applicazione di tipo CAD (*Aided Computer Design*) rappresentante una traiettoria da riprodurre tramite PLC (*Programmable Logic Controller*). Tale software è poi incaricato di estrapolare i dati interessati caratterizzanti il disegno rappresentato. I dati estrapolati sono poi confrontati con i vincoli fisici dettati dalla struttura del PLC (*Programmable Logic Controller*) per verificarne o meno la riproducibilità nell'area di lavoro dello stesso. In caso di riscontro positivo, sono convertiti in un formato consono alla lettura del PLC e infine eseguiti dal PLC stesso.

Questo sistema si inserisce all'interno del mercato del *packaging*: lo scopo finale sarà quello di far eseguire al PLC traiettorie disegnate in due dimensioni tramite un software CAD (*Aided computer design*), emulando l'azione di un ugello atto a controllare la fuori uscita di colla tramite un'uscita ausiliaria.

La tesi è così strutturata:

1. Presentazione e studio del file DXF;
2. Presentazione e studio del sistema PLC;
3. Stesura e scrittura del software atte a operare le azioni evidenziate.
4. Conclusioni.

2 Presentazione e studio del file DXF

Il file DXF (*Drawing Exchange Format*) è un file di tipo CAD sviluppato da AutoDesk. Questo tipo di file è stato creato e progettato per consentire l'interscambio di dati tra AutoCAD ed altri software, sia di progettazione CAD che di altro genere. La prima introduzione del file DXF è avvenuta con l'avvento di AutoCAD 1.0 nel 1982: presentava il supporto ASCII, tuttavia non erano state rilasciate specifiche in merito. Con il rilascio AutoCAD 10 nel 1988 è stato introdotto anche il supporto binario del formato. Con la versione 13 del 1994, AutoCAD ha rilasciato per la prima volta le specifiche del formato. [1. Fonte: AutoCAD DXF, Wikipedia]

Il file DXF è un tipo di file caratterizzato da oggetti (*objects*) ed entità (*entities*). Gli oggetti non hanno rappresentazione grafica, a differenza delle entità. Le entità si possono quindi considerare come "oggetti grafici". Ogni oggetto (che sia grafico o non) è inoltre caratterizzato da determinate caratteristiche definite da *tag*: i tag sono dei valori numerici di tipo intero (*integer*) i quali precedono il dato da rappresentare, definendone il tipo o il significato. Nel file DXF i tag sono definiti *group code*. I group code sono identici, sia che ci si riferisca a un oggetto grafico che ad un oggetto non grafico.

Il file DXF è diviso in sezioni: ogni sezione ha il compito di descrivere un particolare tipo di oggetto o un gruppo di oggetti ad esso associato. Le sezioni sono:

- HEADER;
- CLASSES;
- TABLES;
- BLOCKS;
- ENTITIES.

La sezione di nostro interesse è la sezione "ENTITIES", all'interno della quale vengono rappresentati i dati di tutti gli oggetti grafici da estrapolare.

2.1 LA SEZIONE ENTITIES

La sezione ENTITIES contiene tutti gli oggetti grafici implementati dal formato DXF. Gli oggetti grafici di nostro interesse sono: POINT, LINE, CIRCLE, ARC. Non verranno implementati altri oggetti in quanto non presenti nella "struttura" del PLC (si veda oltre). All'interno della sezione gli oggetti grafici sono introdotti dal tag riportante il nome dello stesso; le righe successive saranno tutte introdotte da un opportuno group code, seguito il valore del dato rappresentato. Per ogni riga vi sarà un solo group code seguito da un solo valore riferito a quel group code. Segue la trattazione dei group code di nostro interesse per ogni tipo di oggetto grafico preso in esame.

2.1.1 Entità: POINT

I group code di nostro interesse rappresentanti l'entità POINT sono i seguenti:

- 10: rappresenta la coordinata x individuante il punto rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 20: rappresenta la coordinata y individuante il punto rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 30: rappresenta la coordinata z individuante il punto rappresentato nello spazio rispetto al sistema di riferimento del disegno.

Ogni altro riferimento è per noi ininfluente.

2.1.2 Entità: LINE

I group code di nostro interesse rappresentanti l'entità LINE sono i seguenti:

- 10: rappresenta la coordinata x individuante il punto iniziale della linea rappresentata nello spazio rispetto al sistema di riferimento del disegno;
- 20: rappresenta la coordinata y individuante il punto iniziale della linea rappresentata nello spazio rispetto al sistema di riferimento del disegno;
- 30: rappresenta la coordinata z individuante il punto iniziale della linea rappresentata nello spazio rispetto al sistema di riferimento del disegno;
- 11: rappresenta la coordinata x individuante il punto finale della linea rappresentata nello spazio rispetto al sistema di riferimento del disegno;
- 21: rappresenta la coordinata y individuante il punto finale della linea rappresentata nello spazio rispetto al sistema di riferimento del disegno;
- 31: rappresenta la coordinata z individuante il punto finale della linea rappresentata nello spazio rispetto al sistema di riferimento del disegno.

Ogni altro riferimento è per noi ininfluente.

2.1.3 Entità: CIRCLE

I group code di nostro interesse rappresentanti l'entità CIRCLE sono i seguenti:

- 10: rappresenta la coordinata x individuante il centro del cerchio rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 20: rappresenta la coordinata y individuante il centro del cerchio rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 30: rappresenta la coordinata z individuante il centro del cerchio rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 40: rappresenta la lunghezza del raggio del cerchio rappresentato;

Ogni altro riferimento è per noi ininfluente.

2.1.4 Entità: ARC

I group code di nostro interesse rappresentanti l'entità ARC sono i seguenti:

- 10: rappresenta la coordinata x individuante il centro dell'arco rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 20: rappresenta la coordinata y individuante il centro dell'arco rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 30: rappresenta la coordinata z individuante il centro dell'arco rappresentato nello spazio rispetto al sistema di riferimento del disegno;
- 40: rappresenta la lunghezza del raggio dell'arco rappresentato;
- 50: rappresenta l'angolo iniziale dell'arco rappresentato;
- 51: rappresenta l'angolo finale dell'arco rappresentato.

In particolare, il file DXF implementa gli archi dall'angolo iniziale a quello finale procedendo in senso antiorario.

Ogni altro riferimento è per noi ininfluente.

La struttura del file DXF e le sue specifiche sono pubblicate sul web e sono reperibili all'indirizzo: http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=24240325 [2].

3 Presentazione e studio del PLC

"Il Controllore a Logica Programmabile o Programmable Logic Controller (PLC) è un controllore per industria specializzato in origine nella gestione o controllo dei processi industriali" [3, Fonte: Controllore Logico Programmabile, Wikipedia].

La principale caratteristica che differenzia un PLC da un PC tradizionale è quello di essere "real time": questa caratteristica risulta necessaria nell'automazione nel momento in cui ci si trova davanti a un problema di movimentazione di oggetti, attuazione di motori e similari, dove la sincronizzazione delle componenti è vitale. Ogni singolo programma PLC è eseguito in un tempo prestabilito e sempre rispettato, fornendo l'output richiesto nell'istante necessario. Questa caratteristica unita alle doti di scalabilità e robustezza, ne fanno un punto di riferimento nel mondo dell'automazione industriale.

3.1 COME SI PROGRAMMA UN PLC

I metodi di programmazione PLC sono stati definiti dalla Commissione Elettronica Internazionale o IEC. Questa organizzazione è costituita per definire gli standard in materia di elettronica e tecnologie correlate. Spesso definisce gli standard in collaborazione con l'Organizzazione Internazionale Per La Normazione, o ISO.

In particolare, lo standard IEC 61131-3 del 1993 definisce i metodi di programmazione di un PLC, di cui due grafici e due testuali. Essi sono:

- Ladder Diagram (LD) detto linguaggio a contatti, grafico;
- Function Block Diagram (FBD), detto linguaggio a blocchi, grafico;
- Istruction List (IL), detto lista di istruzioni, testuale;
- Structed Text (ST), detto linguaggio strutturato, testuale.

[4, Fonte: IEC 61131-3, Wikipedia]

Ai fini del progetto è stato richiesto l'utilizzo di questo ultimo metodo di programmazione.

3.1.1 PLC utilizzato: Panasonic FP Sigma

Il PLC utilizzato è il Panasonic FP Sigma, a cui sono state abbinati due assi controllati da due schede di controllo, una per ogni asse. Gli assi sono movimentati da motori brushless. Il PLC è rappresentato in figura 3.1.

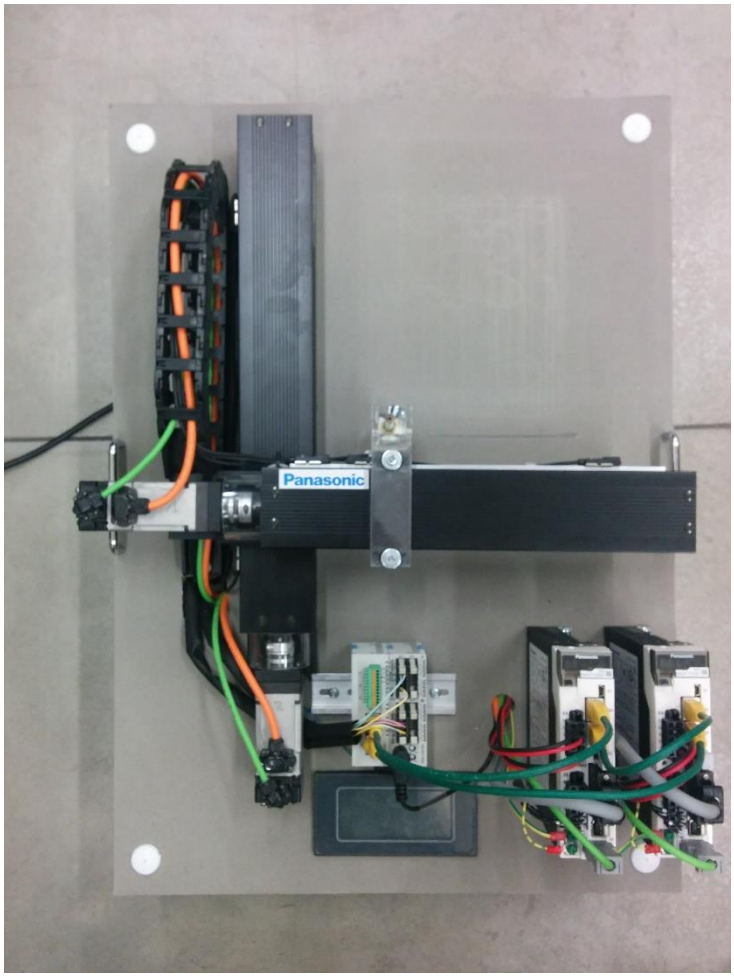

Figura 3.1: PLC utilizzato Panasonic FP Sigma completo degli assi e delle schede di controllo assi.

Il PLC è dotato di una memoria, denominata dal produttore *Shared Memory*, la quale contiene i dati da processare ed eseguire. I dati letti dalla Shared Memory vengono poi passati alle schede di controllo assi le quali hanno il compito di operare fisicamente i dati ricevuti in ingresso tramite la movimentazione del relativo asse.

3.2 STRUTTURA DEL PLC: LA SHARED MEMORY

La Shared Memory è la memoria del PLC. In essa vengono scritti i parametri i quali verranno poi letti dalle schede di controllo di ogni asse e processati sincronicamente. La Shared Memory è strutturata in banchi; ogni banco a sua volta è suddiviso in tabelle e a seconda del banco di

riferimento, ogni tabella ha una propria struttura e significa. Procediamo ora ad una analisi più approfondita dei banchi stessi.

3.2.1 Shared Memory: struttura dei banchi

La Shared Memory è divisa in banchi, detti *banks*. Ogni banco è a sua volta diviso in *address*, ovvero indirizzi. Ogni banco raggruppa al suo interno uno o più gruppi di indirizzi, ognuno di quali contiene un determinato tipo di informazione. Sia i banchi che gli indirizzi sono espressi in base 16. Segue una rapida descrizione dei gruppi di indirizzi che strutturano la Shared Memory:

- Banco 00H:
    - Indirizzo da 080H a 085H: *setting parameter control area*;
    - Indirizzo 088H: *operation speed rate area*;
    - Indirizzo da 0B0H a 0B4H: *axis group setting area*;
    - Indirizzo da 0C0H a 0D7H: *home change data area*;
    - Indirizzo da 0D8H a 0E7H: *torque change area;*
    - Indirizzo da 100H a 107H: *positioning table setting area;*
    - Indirizzo da 111H a 1A7H: *error annunciation & clear area*;
    - Indirizzo da 1A9H a 23FH: *warning announciation & clear area*;
- Banco 01H:
    - Indirizzo da 000H a 1FFH: *information of axis 1, 2, 3, 4, 5, 6, 7, 8*;
- Banco da 02H a 0BH: *parameter setting area & positioning data setting area of axis 1*;
- Banco da 0CH a 15H: *parameter setting area & positioning data setting area of axis 2*;
- Banco da 16H a 1FH: *parameter setting area & positioning data setting area of axis 3*;
- Banco da 20H a 29H: *parameter setting area & positioning data setting area of axis 4*;
- Banco da 2AH a 33H: *parameter setting area & positioning data setting area of axis 5*;
- Banco da 34H a 3DH: *parameter setting area & positioning data setting area of axis 6*;
- Banco da 3EH a 47H: *parameter setting area & positioning data setting area of axis 7*;
- Banco da 48H a 51H: *parameter setting area & positioning data setting area of axis 8*;
- Banco 52H: *AMP parameter control area.*

La struttura qui rappresentata è tratta dal manuale tecnico dell'oggetto, reperibile all'indirizzo: https://www.panasonic-electric-works.com/cps/rde/xbcr/pew_eu_en/mn_63489_en_fpg_fp2_positioning_rtex.pdf [5]

I banchi di nostro interesse sono quelli associati all'asse 1 e 2 (saranno detti anche asse X e Y rispettivamente), ovvero quelli ad 02H a 0BH e da 0CH a 15H. Ogni banco è diviso internamente in "*tables*", ovvero tabelle. Ogni tabella rappresenta all'interno della Shared Memory un punto nel piano 2D raggiungibile dal PLC. I banchi 02H per l'asse X e 0BH per l'asse Y sono divisi in 59 tabelle, il cui "*address*" o indirizzo ha valore iniziale 050H e valore finale 3F0H. Ogni tabella contiene a sua volta 16 indirizzi di memoria detti "*offset address*"; tra l'indirizzo di una tabella e l'altro vi è quindi un salto di 16H.

I banchi da 03H a 0AH per l'asse X e 0DH a 14H sono suddivisi in 64 tabelle il cui indirizzo ha valore iniziale 00H e valore finale 3F0H. Ogni tabella contiene a sua volta 16 indirizzi di memoria detti "*offset address*"; tra l'indirizzo di una tabella e l'altro vi è quindi un salto di 16H.

I banchi 0BH per l'asse X e 15H per l'asse Y sono suddivisi in 29 tabelle, il cui indirizzo ha valore iniziale 00H fino a 1C0H. Ogni tabella contiene a sua volta 16 indirizzi di memoria detti "*offset address*"; tra l'indirizzo di una tabella e l'altro vi è quindi un salto di 16H.

In totale l'asse X e l'asse Y possono raggiungere fino a 600 punti rispettivamente.

Nella figura 3.2 è riportata la struttura appena descritta. Nella figura 3.3 è rappresentata la suddivisione in indirizzi per il banco 02H e 0CH, nella figura 3.4 quella del banco 03H e 0DH, nella figura 3.5 quella del banco 0BH e 15H.

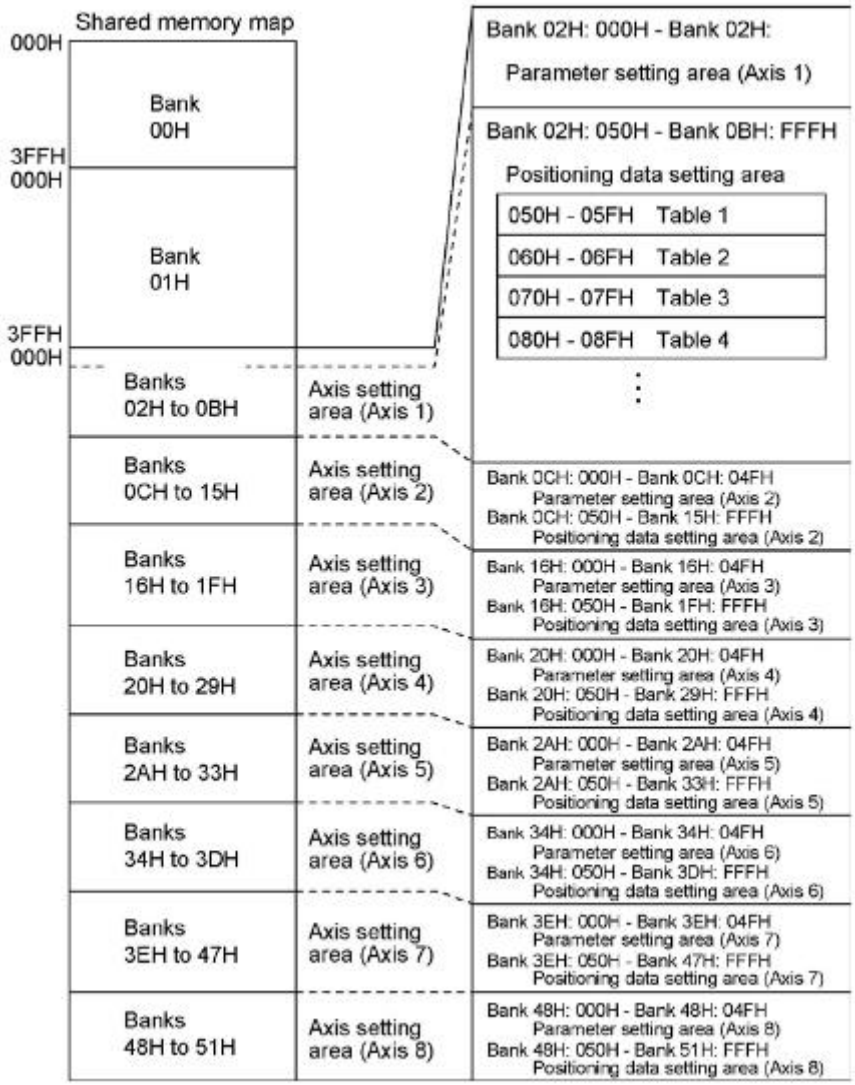

Figura 3.2: rappresentazione della struttura della Shared Memory. I banchi di nostro interesse sono quelli dell'asse 1 e 2, rappresentanti rispettivamente l'asse X e Y.

Starting address of each positioning table

Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8		
Bank No.								**Add-ress**	**Descriptions**
02H	0CH	16H	20H	2AH	34H	3EH	48H	050H	Starting address of table 1
								060H	Starting address of table 2
								070H	Starting address of table 3
								080H	Starting address of table 4
								090H	Starting address of table 5
								0A0H	Starting address of table 6
								0B0H	Starting address of table 7
								0C0H	Starting address of table 8
								0D0H	Starting address of table 9
								0E0H	Starting address of table 10
								0F0H	Starting address of table 11
								100H	Starting address of table 12
								110H	Starting address of table 13
								120H	Starting address of table 14
								130H	Starting address of table 15
								140H	Starting address of table 16
								150H	Starting address of table 17
								160H	Starting address of table 18
								170H	Starting address of table 19
								180H	Starting address of table 20
								190H	Starting address of table 21
								1A0H	Starting address of table 22
								1B0H	Starting address of table 23
								1C0H	Starting address of table 24
								1D0H	Starting address of table 25
								1E0H	Starting address of table 26
								1F0H	Starting address of table 27
								200H	Starting address of table 28
								210H	Starting address of table 29
								220H	Starting address of table 30
								230H	Starting address of table 31
								240H	Starting address of table 32
								250H	Starting address of table 33
								260H	Starting address of table 34
								270H	Starting address of table 35
								280H	Starting address of table 36
								290H	Starting address of table 37
								2A0H	Starting address of table 38
								2B0H	Starting address of table 39
								2C0H	Starting address of table 40
								2D0H	Starting address of table 41
								2E0H	Starting address of table 42
								2F0H	Starting address of table 43
								300H	Starting address of table 44
								310H	Starting address of table 45
								320H	Starting address of table 46
								330H	Starting address of table 47
								340H	Starting address of table 48
								350H	Starting address of table 49
								360H	Starting address of table 50
								370H	Starting address of table 51
								380H	Starting address of table 52
								390H	Starting address of table 53
								3A0H	Starting address of table 54
								3B0H	Starting address of table 55
								3C0H	Starting address of table 56
								3D0H	Starting address of table 57
								3E0H	Starting address of table 58
								3F0H	Starting address of table 59

Figura 3.3: rappresentazione della struttura interna di ogni banco della Shared Memory. Si noti come il primo banco di ogni asse contenga esattamente 59 tabelle, individuabili tramite i rispettivi indirizzi di memoria espressi in esadecimale.

Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8		
Bank No.								**Add-ress**	**Descriptions**
03H	0DH	17H	21H	2BH	35H	3FH	49H	000H	Starting address of table 60
								010H	Starting address of table 61
								020H	Starting address of table 62
								030H	Starting address of table 63
								040H	Starting address of table 64
								050H	Starting address of table 65
								060H	Starting address of table 66
								070H	Starting address of table 67
								080H	Starting address of table 68
								090H	Starting address of table 69
								0A0H	Starting address of table 70
								0B0H	Starting address of table 71
								0C0H	Starting address of table 72
								0D0H	Starting address of table 73
								0E0H	Starting address of table 74
								0F0H	Starting address of table 75
								100H	Starting address of table 76
								110H	Starting address of table 77
								120H	Starting address of table 78
								130H	Starting address of table 79
								140H	Starting address of table 80
								150H	Starting address of table 81
								160H	Starting address of table 82
								170H	Starting address of table 83
								180H	Starting address of table 84
								190H	Starting address of table 85
								1A0H	Starting address of table 86
								1B0H	Starting address of table 87
								1C0H	Starting address of table 88
								1D0H	Starting address of table 89
								1E0H	Starting address of table 90
								1F0H	Starting address of table 91
								200H	Starting address of table 92
								210H	Starting address of table 93
								220H	Starting address of table 94
								230H	Starting address of table 95
								240H	Starting address of table 96
								250H	Starting address of table 97
								260H	Starting address of table 98
								270H	Starting address of table 99
								280H	Starting address of table 100
								290H	Starting address of table 101
								2A0H	Starting address of table 102
								2B0H	Starting address of table 103
								2C0H	Starting address of table 104
								2D0H	Starting address of table 105
								2E0H	Starting address of table 106
								2F0H	Starting address of table 107
								300H	Starting address of table 108
								310H	Starting address of table 109
								320H	Starting address of table 110
								330H	Starting address of table 111
								340H	Starting address of table 112
								350H	Starting address of table 113
								360H	Starting address of table 114
								370H	Starting address of table 115
								380H	Starting address of table 116
								390H	Starting address of table 117
								3A0H	Starting address of table 118
								3B0H	Starting address of table 119
								3C0H	Starting address of table 120
								3D0H	Starting address of table 121
								3E0H	Starting address of table 122
								3F0H	Starting address of table 123

Figura 3.4: rappresentazione della struttura interna di ogni banco della Shared Memory. Dal banco 03H per l'asse X e 0DH per l'asse Y, il quantitativo di tabelle contenute è pari a 64. Questa rappresentazione è identica per i banchi fino al 0AH e 14H per l'asse X e Y rispettivamente.

Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8		
Bank No.								**Add-ress**	**Descriptions**
0BH	15H	1FH	29H	33H	3DH	47H	51H	000H	Starting address of table 572
								010H	Starting address of table 573
								020H	Starting address of table 574
								030H	Starting address of table 575
								040H	Starting address of table 576
								050H	Starting address of table 577
								060H	Starting address of table 578
								070H	Starting address of table 579
								080H	Starting address of table 580
								090H	Starting address of table 581
								0A0H	Starting address of table 582
								0B0H	Starting address of table 583
								0C0H	Starting address of table 584
								0D0H	Starting address of table 585
								0E0H	Starting address of table 586
								0F0H	Starting address of table 587
								100H	Starting address of table 588
								110H	Starting address of table 589
								120H	Starting address of table 590
								130H	Starting address of table 591
								140H	Starting address of table 592
								150H	Starting address of table 593
								160H	Starting address of table 594
								170H	Starting address of table 595
								180H	Starting address of table 596
								190H	Starting address of table 597
								1A0H	Starting address of table 598
								1B0H	Starting address of table 599
								1C0H	Starting address of table 600

Figura 3.5: rappresentazione della struttura interna di ogni banco della Shared Memory. Per gli ultimi banchi le tabelle contenute sono 29.

Ora passeremo all'analisi della struttura della singola tabella.

3.2.2 Shared Memory: struttura di una tabella

Ogni tabella è costituita da 16 indirizzi, detti "*offset address*". Ogni indirizzo riceve dati di diversa entità istruendo il PLC sul punto in cui posizionarsi e sul metodo di posizionamento. Lo studio della struttura interna della Shared Memory e del metodo di salvataggio dei dati nelle singole locazioni di memoria è stato effettuato tramite test, incrociando i risultati dati dall'utilizzo di due software proprietari di Panasonic: Control Configurator PM e FP Connect. Il primo permette di scrivere all'interno della Shared Memory i dati necessari per istruire il PLC sul punto e sul metodo di posizionamento; la struttura implementata dal software ricalca la struttura tabellare della Shared Memory stessa; il secondo invece permette di leggere il risultato della scrittura nella singola locazione di memoria, alla posizione desiderata. Questi test hanno evidenziato che i valori restituiti da una prima lettura siano tutti in esadecimale.

Segue una descrizione dettagliata e schematica di ogni indirizzo:

3.2.3 Shared Memory: indirizzo 000H, *Control Code*

Questo indirizzo specifica due caratteristiche: "*Increment / Absolute Setting*" ovvero scelta del sistema di riferimento e "*Acceleration / Deceleration Pattern Setting*" ovvero metodo di accelerazione e decelerazione. La prima caratteristica specifica il sistema di riferimento scelto; questo può assumere i seguenti valori:

- "*Absolute*", ovvero sistema riferito allo zero assoluto del PLC;
- "*Increment*" ovvero sistema riferito all'ultimo punto iterato in tabella.

La seconda caratteristica specifica il tipo di accelerazione con il quale il PLC deve operare: questa può assumere i seguenti valori:

- "*Linear*" ovvero accelerazione lineare;
- "*S Shaped*", ovvero accelerazione con andamento a "S"; questa è preferibile nel caso in cui il PLC è chiamato a muovere oggetti particolarmente pesanti e quindi soggetti a inerzie non trascurabili.

Ogni possibile combinazione delle caratteristiche descritte modella un risultato diverso all'indirizzo di memoria analizzato. Le possibili combinazioni sono:

- Sistema riferito "*Increment*" con accelerazione "*Linear*", restituisce un valore scritto all'indirizzo in esame pari a 0000H;
- Sistema riferito "*Absolute*" con accelerazione "*Linear*", valore scritto all'indirizzo di memoria analizzato pari a 0001H;
- Sistema riferito "*Increment*" con accelerazione "*S Shaped*", valore scritto all'indirizzo di memoria analizzato pari a 0002H;
- Sistema riferito "*Absolute*" con accelerazione "*S Shaped*", valore scritto all'indirizzo di memoria analizzato pari a 0003H.

Passiamo ora all'indirizzo successivo.

3.2.4 Shared Memory: indirizzo 001H, *Operation Pattern*

Questo indirizzo specifica due caratteristiche: "*Control Pattern*" ovvero schema di posizionamento e "*Interpolation Setting*" ovvero tipo di interpolazione. La prima caratteristica modella il rapporto che la tabella in uso ha con la tabella precedente o successiva ad essa. L'indirizzo può assumere 3 valori:

- "*End Point Control*", nel caso in cui la tabella riferita al punto processato sia l'ultima: giunto in questo punto il PLC si ferma, ignorando il *Dwell Time*;
- "*Pass Point Control*", giunto in questo punto il PLC processa immediatamente il punto riferito alla tabella successiva, senza fermarsi ignorando il *Dwell Time*;
- "*Contiuance Point Control*", giunto in questo punto il PLC si ferma per un tempo pari al tempo specificato come "*Dwell Time*", per poi continuare a processare il punto riferito alla tabella successiva.

La seconda caratteristica modella il tipo di interpolazione da eseguire. Essa può assumere un numero di valori diverso a seconda del numero di assi interpolati (da 1 a 8). Noi analizzeremo solo i valori per possibili per 2 assi interpolati. Questi valori sono:

- "*Linear Interpolation (Composite Speed)*", la quale specifica il raggiungimento del punto riferito alla tabella in processo tramite una linea; il punto di partenza sarà il punto riferito alla tabella precedentemente processata, ovvero quella alla posizione i – 1, nel caso in cui la tabella che specifica tale caratteristica sia la tabella i; la velocità di posizionamento sarà quella specificata come "*Positioning Target Speed*" e sarà associata alla velocità assoluta dell'oggetto mosso dal PLC; in questo caso l'*Auxiliary Point* viene ignorato;
- "*Linear Interpolation (Long Axis Speed)*", equivalente all'interpolazione precedente con la differenza che la velocità specificata sarà associata all'asse più lungo di movimentazione del PLC; in questo caso l'*Auxiliary Point* viene ignorato;
- "*Circular Interpolation (Center Point/CW direction)*", la quale specifica una interpolazione di tipo circolare; nel caso in cui il punto riferito alla tabella i-esima processata avente questa caratteristica sia il medesimo del punto riferito alla tabella i – 1, allora l'interpolazione restituirà un cerchio completo; nel caso in cui sia diverso, restituirà un arco di circonferenza; il centro del cerchio sarà specificato come "*Auxiliary Point*" dell'asse X e Y; l'esecuzione del cerchio sarà in senso orario;

- "*Circular Interpolation (Center Point/CCW direction)*" equivalente all'interpolazione precedente con la differenza che il senso d'esecuzione della stessa sarà antiorario;
- "*Circular Interpolation (Pass Point)*" equivalente all'interpolazione precedente con la differenza che il punto ausiliario sarà processato non come centro del cerchio ma come punto di passaggio della circonferenza.

Come per il Control Code, ogni possibile combinazione delle caratteristiche descritte modella un risultato diverso all'indirizzo di memoria analizzato. A ogni caso del Control Pattern e a ogni caso dell'Interpolation Setting sarà assocciato un valore composto da due cifre esadecimali; in memoria verrà poi allocato un solo valore pari alla concatenazione del risultato del Interpolation Setting seguito dal valore del Control Pattern.

I valori per il Control Pattern sono:

- 00H per l'End Point;
- 01H per il Pass Point;
- 02H per il Continuance Point.

I valori per l'Interpolation Setting sono:

- 00H per il Linear Interpolation (Composite Speed);
- 01H per il Linear Interpolation (Long Axis Speed);
- 10H per il Circular Interpolation (Center Point / CW);
- 11H per il Circular Interpolation (Center Point / CCW);
- 20H Per il Circulat Interpolation (Pass Point).

Ad esempio, un Control Pattern di tipo Pass Point associato a un'Interpolation Setting di tipo Circular Interpolation (Center Point / CCW) sarà rappresentato dal valore 1101H, dove il valore 11H iniziale rappresenta il tipo di interpolazione scelta, e il valore 01H finale il valore di Control Pattern scelto.

3.2.5 Shared Memory: indirizzo 004H, *Positioning Acceleration Time*

Questo indirizzo specifica il tempo in cui l'oggetto movimentato raggiunge la velocità richiesta, quindi il suo tempo di accelerazione. L'unità di misura del valore specificato è millisecondi.

3.2.6 Shared Memory: indirizzo 004H, *Positioning Deceleration Time*

Questo indirizzo specifica il tempo in cui l'oggetto movimentato rallenta fino a zero dalla velocità richiesta, quindi il suo tempo di decelerazione. L'unità di misura del valore specificato è millisecondi.

3.2.7 Shared Memory: indirizzo 006H e 007H, *Positioning Target Speed*

Questi due indirizzi specificano la velocità di interpolazione richiesta, espressa in micrometri al secondo. Questa velocità è rappresentata da due allocazioni di memoria in modo tale da estendere un valore settabile maggiore di 65535 (valore corrispondente all'esadecimale FFFF): infatti questo valore corrisponde a 0,065535 metri al secondo, velocità di interpolazione molto bassa. Le velocità superiori al valore esadecimale FFFF vengono assegnate alla Shared Memory nel seguente modo: le ultime 4 cifre esadecimali vengono allocate all'indirizzo 006H, mentre le restanti vengono assegnate all'indirizzo 007H.

3.2.8 Shared Memory: indirizzo 008H e 009H, *Positioning Movement Amount*

Questi due indirizzi specificano la coordinata del punto sull'asse in cui il PLC è chiamato a posizionarsi, espressa in micrometri. Questa coordinata è rappresentata da due allocazioni di memoria in modo tale da impostare un valore settabile maggiore di 65535 (valore corrispondente all'esadecimale FFFF): in fatti questo valore corrisponderebbe a 65,535

millimetri, valore molto basso. Le coordinate superiori all'esadecimale FFFF vengono assegnate alla Shared Memory nel seguente modo: le ultime 4 cifre esadecimali vengono allocate all'indirizzo 008H, mentre le restanti vengono assegnate all'indirizzo 009H.

3.2.9 Shared Memory: indirizzo 00AH e 00BH, *Auxiliary Point*

Questi due indirizzi specificano la coordinata ausiliaria del punto in cui il PLC è chiamato a posizionarsi, espressa in micrometri. Tutte le considerazioni fatte per le allocazioni 008H e 009H valgono anche per le allocazioni 00AH e 00BH rispettivamente.

3.2.10 Shared Memory: indirizzo 00CH, *Dwell Time*

Questo indirizzo specifica il tempo che intercorre tra il processamento di una tabella e l'altra, se previsto. Il suo valore è espresso in millisecondi.

3.2.11 Shared Memory: indirizzo 00DH, *Auxiliary Output Code*

Questo indirizzo specifica un eventuale valore ausiliario associato alla tabella in uso.

Nella figura 3.6 è riportata la struttura appena descritta.

Positioning tables

Data in the following formats is stored from the starting address of positioning tables of each axis.

<table>
<tr><th>Offset address</th><th>Name</th><th>Descriptions</th></tr>
<tr><td>000H</td><td>Control code</td><td>Sets the position setting mode and acceleration/deceleration pattern for the positioning operation.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>0</td><td>Increment /absolute setting</td><td>00H</td><td>00: Increment mode
1: Absolute mode</td></tr>
<tr><td>1</td><td>Acceleration/deceleration pattern setting</td><td>00H</td><td>00: Linear acceleration/deceleration
1: S-shaped acceleration/deceleration</td></tr>
<tr><td>15 to 2</td><td>-</td><td>-</td><td>-</td></tr>
</table></td></tr>
<tr><td>001H</td><td>Operation pattern</td><td>Sets the independent and interpolation patterns for the positioning operation.
The relation of the interpolation depends on the settings in the axis group setting area in the common area of the shared memory.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>7 to 0</td><td>Control pattern</td><td>00H</td><td>00H: Epoint control (End point control)
01H: P point control (Pass point control)
02H: C point control (Continuance point control)
03H: Jpoint control (Speed point control)
Any other settings will be errors.</td></tr>
<tr><td>15 to 8</td><td>Interpolation setting</td><td>00H</td><td>00H: Linear interpolation (Composite speed)
01H: Linear interpolation (Long axis speed)
10H: Circular interpolation (Center point/CW direction)
11H: Circular interpolation (Center point/ CCW direction)
20H: Circular interpolation (Pass point)
50H: Spiral interpolation (Center point/CW direction/X-axis movement)
51H: Spiral interpolation (Center point/CCW direction/X-axis movement)
52H: Spiral interpolation (Center point/CW direction/Y-axis movement)
53H: Spiral interpolation (Center point/CCW direction/Y-axis movement)
54H: Spiral interpolation (Center point/CW direction/Z-axis movement)
55H: Spiral interpolation (Center point/CCW direction/Z-axis movement)
60H: Spiral interpolation (Pass point/X-axis movement)
61H: Spiral interpolation (Pass point/Y-axis movement)
62H: Spiral interpolation (Pass point/Z-axis movement)
Any other settings will be errors.</td></tr>
</table></td></tr>
<tr><td>002H</td><td>-</td><td>-</td></tr>
<tr><td>003H</td><td>-</td><td>-</td></tr>
<tr><td>004H</td><td>Positioning acceleration time</td><td rowspan="2">Sets the acceleration and deceleration time for the positioning operation.
The acceleration time and deceleration time can be set individually.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>15 to 0</td><td>Acceleration time
Deceleration time</td><td>100</td><td>Setting range: 0 to 10,000 (ms)
Any other settings will be errors.</td></tr>
</table></td></tr>
<tr><td>005H</td><td>Positioning deceleration time</td></tr>
<tr><td>006H</td><td rowspan="2">Positioning target speed (interpolation speed)</td><td rowspan="2">In case of the individual operation (no interpolation), it is the target speed of the corresponding axis. In case of the interpolation operation, it is the target speed of the interpolation.
In the interpolation operation, the target speed for the axis of the smallest number in a group is valid.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>31 to 0</td><td>Positioning target speed (Interpolation speed)</td><td>1,000</td><td>Setting range: 1 to 32,767,000
Any other settings will be errors.
The interpretation is changed bythe unit setting.
pulse: 1 to 32,767,000 pps
μm: 1 to 32,767,000 μm/s
inch: 0.001 to 32,767.000 inch/s
degree: 0.001 to 32,767.000 rev/s</td></tr>
</table></td></tr>
<tr><td>007H</td></tr>
</table>

<table>
<tr><th>Offset address</th><th>Name</th><th>Descriptions</th></tr>
<tr><td>008H
009H</td><td>Positioning movement amount</td><td>The area to set the movement amount for the positioning operation.
The interpretation is changed for the increment movement amount or absolute coordinate by the control code setting.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>31 to 0</td><td>Positioning movement amount</td><td>0</td><td>Setting range: -1,073,741,823 to 1,073,741,823
Any other settings will be errors.
The interpretation varies depending on the unit setting.
pulse: -1,073,741,823 to 1,073,741,823 pulse
μm (0.1 μm): -107,374,182.3 to 107,374,182.3 μm/s
μm (1 μm): -1,073,741,823 to 1,073,741,823 μm/s
inch (0.00001 inch): -10,737.41823 to 10,737.41823 inch
inch (0.0001 inch): -107,374.1823 to 107,374.1823 inch
degree (0.1 degree): -107,374,182.3 to 107,374,182.3 degree
degree (1 degree): -1,073,741,823 to 1,073,741,823 degree</td></tr>
</table></td></tr>
<tr><td>00AH
00BH</td><td>Auxiliary point</td><td>The area to set the auxiliary points (center point, pass point coordinates) in case of the circular interpolation or spiral interpolation control.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>31 to 0</td><td>Auxiliary point</td><td>0</td><td>Setting range: -1,073,741,823 to 1,073,741,823
Any other settings will be errors.
The interpretation varies depending on the unit setting.
pulse: -1,073,741,823 to 1,073,741,823 pulse
μm (0.1 μm): -107,374,182.3 to 107,374,182.3 μm/s
μm (1 μm): -1,073,741,823 to 1,073,741,823 μm/s
inch (0.00001 inch): -10,737.41823 to 10,737.41823 inch
inch (0.0001 inch): -107,374.1823 to 107,374.1823 inch
degree (0.1 degree): -107,374,182.3 to 107,374,182.3 degree
degree (1 degree): -1,073,741,823 to 1,073,741,823 degree</td></tr>
</table></td></tr>
<tr><td>00CH</td><td>Dwell time</td><td>After the completion of the positioning control of this table;
when the mode is C: Continuation point, stops the motor operation for the dwell time and starts the operation of the next table.
when the mode is P: Pass point, this setting is ignored.
when the mode is E: End point, the positioning done contact will turn on after waiting for the dwell time.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>15 to 0</td><td>Dwell time</td><td>0</td><td>0 to 32,767: The unit is ms.
Any other settings will be errors.</td></tr>
</table></td></tr>
<tr><td>00DH</td><td>Auxiliary output code</td><td>Sets the data to be output to the auxiliary output code in each axis information & monitor area by the setting of the auxiliary output mode in the parameter setting area.
<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td>15 to 0</td><td>Auxiliary output code</td><td>0</td><td>No specific setting range.</td></tr>
</table></td></tr>
<tr><td>00EH</td><td>-</td><td>-</td></tr>
<tr><td>00DH</td><td>-</td><td>-</td></tr>
</table>

Figura 3.6: rappresentazione della struttura interna di ogni tabella della Shared Memory. Si noti come gli indirizzi 002H, 003H, 00EH, 00DH non siano inutilizzati.

4 Il software

4.1 ANALISI

Il metodo di programmazione richiesto è basato sul testo strutturato o "*Structed Text*". In particolare è stata messa a disposizione un "controllo ActiveX" denominato "*FP Connect Function List*" il quale permette di stabilire molteplici canali di comunicazione con la Shared Memory tramite diverse interfacce, tra cui RS232/485, USB, ethernet o modem. L'interfaccia richiesta ai fini del progetto è la RS232/485.

I linguaggi supportati da FP Connect sono VBA, VB.Net, C++ e C#. La scelta del linguaggio è ricaduta su C#.

L'IDE utilizzato è Visual Studio 2015, scelta consigliata dal fornitore di FP Connect.

La trattazione che segue analizza l'implementazione software attraverso la descrizione delle classi utilizzate. Lo sviluppo è affrontato attraverso la divisione del problema in 3 parti: la prima parte prevede lo sviluppo degli strumenti atti ad estrapolare ed analizzare i dati forniti dal file DXF; la seconda parte prevede lo sviluppo degli strumenti atti a convertire i file analizzati in una struttura dati simile alla struttura della Shared Memory, in modo tale da facilitarne gestione e scrittura; la terza parte prevede lo sviluppo degli strumenti atti a gestire le funzioni fornite con il pacchetto FP Connect, in modo da facilitarne l'uso.

L'implementazione di queste 3 parti è stata operata "a console"; successivamente è stata impostata un'interfaccia grafica. Al termine della trattazione dello sviluppo a console, verrà presentata l'interfaccia.

Di seguito viene riportato il diagramma di flusso rappresentante la logica ad alto livello delle principali funzioni del software.

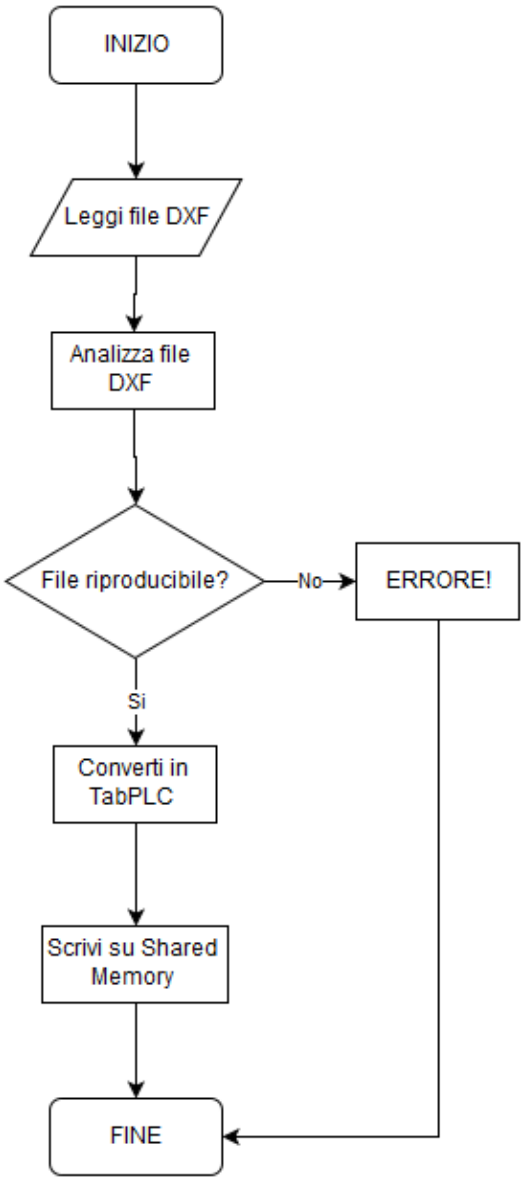

Figura 4.1: Diagramma di flusso rappresentante le principali operazioni effettuate dal software.

4.2 L'IMPLEMENTAZIONE A CONSOLE

4.2.1 Classi DXF

Le classi rappresentanti il file DXF sono state dedotte studiando il tipo di oggetti grafici implementabili dalla Shared Memory. Questi oggetti sono: punti, linee, cerchi e archi.

La classe "Punto" è composta dai seguenti campi e metodi:

- Campo nomePunto, rappresentante il nome del punto;
- Campo X0, rappresentante la coordinata x del punto;

- Campo Y0, rappresentante la coordinata y del punto;
- Metodi di supporto come "*getters*" e "*ToString*" (creati seguendo i paradigmi della buona programmazione ad oggetti).

La classe **"**Linea" è composta dai seguenti campi e metodi:

- Campo nomeLinea, rappresentante il nome della linea;
- Campo X0, rappresentante la coordinata x del punto iniziale;
- Campo Y0, rappresentante la coordinata y del punto iniziale;
- Campo X1, rappresentante la coordinata x del punto finale;
- Campo Y1, rappresentante la coordinata y del punto finale;
- Metodi di supporto come "*getters*" e "*ToString*" (creati seguendo i paradigmi della buona programmazione ad oggetti).

La classe "Cerchio" è composta dai seguenti campi e metodi:

- Campo nomeCerchio, rappresentante il nome del cerchio;
- Campo X0centro, rappresentante la coordinata x del centro;
- Campo Y0 centro, rappresentante la coordinata y del centro;
- Campo raggio, rappresentante il raggio;
- Campo XIniziale, rappresentante la coordinata x del punto iniziale da cui rappresentare il cerchio;
- Campo YIniziale, rappresentante la coordinata x del punto iniziale da cui rappresentare il cerchio;
- Metodi di supporto come "*getters*" e "*ToString*" (creati seguendo i paradigmi della buona programmazione ad oggetti).

La classe "Arco" è composta dai seguenti campi e metodi:

- Campo nomeArco, rappresentante il nome dell'arco;
- Campo X0centro, rappresentante la coordinata x del centro;
- Campo Y0 centro, rappresentante la coordinata y del centro;
- Campo raggio, rappresentante il raggio;
- Campo XIniziale, rappresentante la coordinata x del punto iniziale da cui rappresentare l'arco;
- Campo YIniziale, rappresentante la coordinata x del punto iniziale da cui rappresentare l'arco;
- Metodi di supporto come "*getters*" e "*ToString*" (creati seguendo i paradigmi della buona programmazione ad oggetti).

Queste classi modellano le rappresentazioni geometriche che verranno estrapolate dal DXF. Insieme a esse è stata create una classe "StrumentiDXF" la quale raggruppa al suo interno tutti i metodi atti a gestire il suddetto file. Essa è composta dai seguenti campi:

- Campo udmDXF, rappresentante l'unità di misura scelta per strutturare il DXF attraverso software CAD;
- Campo udmPLC, rappresentante l'unità di misura scelta per rappresentare le distanze sul PLC;
- Campi xMax, yMax, xMin, yMin rappresentanti le coordinate dell'area di lavoro a disposizione del PLC.

I metodi implementati nella classe sono:

- Metodo "leggiDXF": ha il compito di leggere il file indicato e di estrapolarne la parte contenente le entità geometriche in esso rappresentate; questa parte è contenuta tra le parole chiave "ENTITIES" ed "ENDSEC". Una volta individuate, il contenuto

compreso tra le due viene salvato riga per riga in un array list chiamato “ricerca”. Infine viene richiamato il metodo “aggiungiEntità”, al quale viene passato l’array list ricerca.

- Metodo “aggiungiEntità”: esso scorre all’interno dell’array list passato alla ricerca delle parole chiave rappresentanti le entità geometriche interessate. Queste parole chiave sono “POINT” per i punti, “LINE” per le linee, “CIRCLE” per i cerchi e “ARC” per gli archi. Se non vi sono errori nella lettura e nel salvataggio delle entità nelle collezioni utilizzate, viene chiamato il metodo “daDXFATabPLC”, della classe “TabellaPLC”. I metodi atti ad aggiungere le entità geometriche alla collezione “listaEntitàGeometriche” di tipo array list, hanno inoltre il compito di convertire eventualmente l’unità di misura dell’oggetto processato e di controllare che lo stesso sia interno all’area di lavoro del PLC; quest’ultima è indicata dall’utente e salvata nel campo “xMin”, “yMin”, “xMax” e “yMax” della classe stessa. Il controllo viene effettuato richiamando il metodo “controlloRispettoArea”. Per i punti e le linee il controllo viene fatto coordinata per coordinata; per il cerchio vengono controllate le coordinate del vertice in alto a sinistra e in basso a destra del quadrato circoscritto; per gli archi vengono creati 10 punti di controllo compresi tra il punto iniziale dell’arco e il suo punto finale in senso antiorario, ricordando che il DXF rappresenta gli archi in senso antiorario appunto. Se la procedura va a buon fine, il software passa alla fase di conversione dei dati estrapolati in una struttura simile a quella della Shared Memory, attraverso la classe “TabellaPLC”.

Di seguito viene riportato il diagramma di flusso rappresentante la logica ad alto livello delle principali operazioni svolte dalle classi demandate a gestire i file DXF.

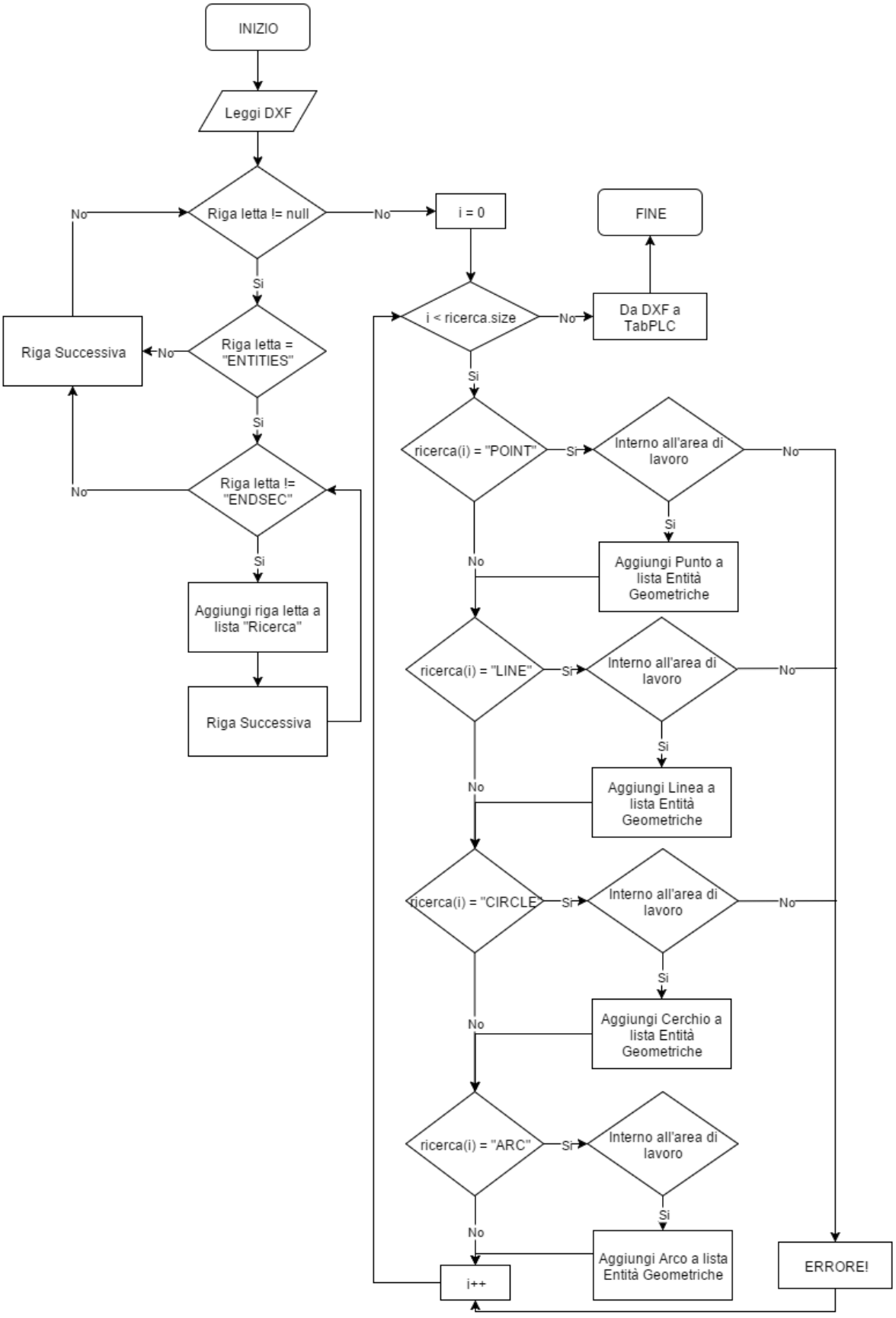

Figura 4.2: Diagramma di flusso rappresentante le principali funzioni implementate per le classi coinvolte nella gestione dei file DXF.

4.2.2 Classi tabelle PLC

Le classi che gestiscono le tabelle PLC sono state dedotte studiando la struttura della Shared Memory; esse sono implementate in modo tale da essere una "immagine a livello software" fedele alla Shared Memory stessa, per facilitarne l'uso nella sua lettura e scrittura. Le classi implementate sono "TabellaPLC" e "StrumentiTabPLC".
La classe TabellaPLC ricalca la struttura di una singola tabella presente all'interno della Shared Memory. I campi e i metodi in essa implementati sono i seguenti:

- Campo "tab", rappresentante il numero della tabella processata;
- Campo "pattern", rappresentante il pattern con il quale la tabella viene processata;
- Campo "interpolationOp", rappresentate il tipo di interpolazioni;
- Campo "controlMethod", rappresentante il sistema di riferimento;
- Campo "positioningMovementAmountAsseX", rappresentante la coordinata X da raggiungere;
- Campo "positioningMovementAmountAsseY", rappresentante la coordinata Y da raggiungere;
- Campo "auxPointAsseX", rappresentante la coordinata ausiliaria X;
- Campo "auxPointAsseY", rappresentante la coordinata ausiliaria Y;
- Campo "acDecPattern", rappresentante il tipo di accelerazione;
- Campo "positioningAcelerationTime", rappresentante il tempo di accelerazione;
- Campo "positioningDecelerationTime", rappresentante il tempo di decelerazione;
- Campo "positioningTargetSpeed", rappresentante la velocità di interpolazione;
- Campo "dwellTime", rappresentate il ritardo temporale da mantenere;
- Campo "auxOut", rappresentante il valore dell'uscita ausiliaria.
- Metodi di supporto come "*getters*" e "*ToString*" (creati seguendo i paradigmi della buona programmazione ad oggetti).

Una volta creata una singola TabellaPLC questa viene memorizzata nell'array list "TabPLC".
Nella classe "StrumentiTabPLC" sono invece raggruppati i metodi atti alla gestione delle tabelle PLC. I metodi implementati nella classe sono:

- Metodo "daDXFATabPLC": questo metodo ha il compito di prendere in esame le entità geometriche memorizzate dentro alla lista "listaEntitàGeometriche" e di processarle in modo tale da ricavarne dei punti di posizionamento per il PLC. Questo metodo richiama a sua volta i metodi "aggiungiPuntoATabPLC", "aggiungiLineaATabPLC", "aggiungiCerchioATabPLC", "aggiungiArcoATabPLC" a seconda che l'entità processata sia un punto, una linea, un cerchio o un arco.
- Metodo "aggiungiPuntoATabPLC": ha il compito di aggiungere un punto alla struttura dati "TabPLC". Come valori vengono dati: pattern di tipo Continuance Point, interpolationOp di tipo Linear Interpolation (Composite Speed), controlMethod di tipo Absolute, acDecPattern di tipo S Shaped; i tempi di accelerazione e decelerazione impostati a 100 millisecondi, la velocità di interpolazione impostata a 50000 micrometri al secondi, tempo di ritardo pari a 500 millisecondi; oltre a questi valori di default vengono poi passati i valori delle coordinate presenti nella listaEntitàGeometriche.
- Metodo "aggiungiLineaATabPLC": ha il compito di aggiungere una linea alla struttura dati "TabPLC". Come valori vengono dati: pattern di tipo Pass Point, interpolationOp di tipo Linear Interpolation (Composite Speed), controlMethod di tipo Absolute, acDecPattern di tipo S Shaped; i tempi di accelerazione e decelerazione impostati a 100 millisecondi, la velocità di interpolazione impostata a 50000 micrometri al secondi, tempo di ritardo pari a 500 millisecondi. Oltre a questi valori di default vengono poi passati i valori delle coordinate presenti nella listaEntitàGeometriche.
- Metodo "aggiungiCerchioATabPLC": ha il compito di aggiungere un cerchio alla struttura dati "TabPLC". Come valori vengono dati: pattern di tipo Pass Point,

controlMethod di tipo Absolute, acDecPattern di tipo S Shaped; i tempi di accelerazione e decelerazione impostati a 100 millisecondi, la velocità di interpolazione impostata a 50000 micrometri al secondi, tempo di ritardo pari a 500 millisecondi. In particolare, l'aggiunta di un cerchio alla "TabPLC" prevede anzitutto l'aggiunta di una linea "di posizionamento" la quale fa sì che il PLC si posizioni nel punto di inizio del cerchio. A questa linea di posizionamento segue poi il processamento del cerchio vero e proprio.

- Metodo "aggiungiArcoATabPLC": ha il compito di aggiungere un arco alla struttura dati "TabPLC". Come valori vengono dati: pattern di tipo Pass Point, controlMethod di tipo Absolute, acDecPattern di tipo S Shaped; i tempi di accelerazione e decelerazione impostati a 100 millisecondi, la velocità di interpolazione impostata a 50000 micrometri al secondi, tempo di ritardo pari a 500 millisecondi. In particolare, l'aggiunta di un cerchio alla "TabPLC" prevede anzitutto l'aggiunta di una linea "di posizionamento" la quale fa sì che il PLC si posizioni nel punto di inizio del cerchio. A questa linea di posizionamento segue poi il processamento dell'arco vero e proprio.

L'algoritmo fin qui implementato porta alla descrizione di linee ed archi attraverso due punti: il punto iniziale di esse viene descritto attraverso l'impostazione dell'uscita ausiliarla "AuxOut" a 1, il punto finale a "0". In questo modo l'ugello che verrà installato si attiverà quando l'uscita ausiliaria sarà "1" e si disattiverà quando sarà "0" appunto.
Il cerchio invece è caratterizzato da una riga di posizionamento e una di esecuzione: la riga di posizionamento è settata a "0" in quanto durante il processamento di quest'ultima l'ugello non deve rilasciare colla; durante l'esecuzione del cerchio invece il codice è pari a "2", in modo da differenziare la gestione di questa entità geometrica rispetto alle altre. Il punto è considerato come un cerchio con raggio nullo: per questo motivo anche il codice ausiliario è settato a "2".

In questo modo si gestisce la parte del processo relativo all'uso dell'ugello atto a spargere la colla, come richiesto.

Di seguito viene riportato il diagramma di flusso rappresentante la logica ad alto livello delle principali operazioni svolte dalle classi demandate a convertire il DXF letto nella struttura "TabPLC".

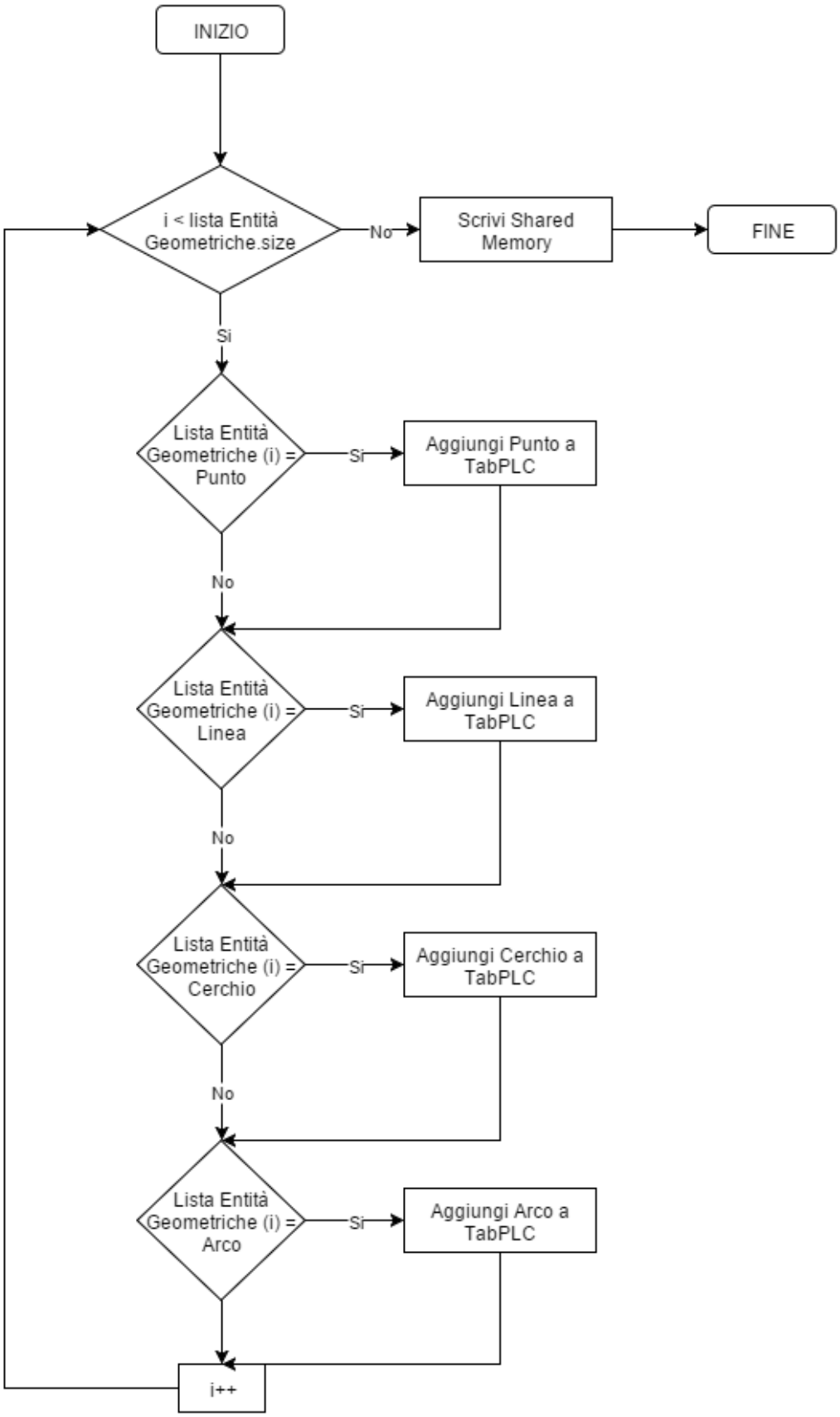

Figura 4.3: Diagramma di flusso rappresentante le principali funzioni implementate per le classi coinvolte nella conversione del DXF nella struttura “TabPLC”.

Ora si passa all'analisi della classe contenente gli strumenti di gestione della comunicazione, lettura e scrittura della Shared Memory.

4.2.3 Classe Strumenti FP Connect

La classe "strumentiFPConnect" si basa sulla libreria FP Connect fornita dal produttore del PLC. Questa classe ha la funzione di "*wrapper*" rispetto alla libreria fornita, in modo tale da semplificarne l'uso rispetto all'obbiettivo. Le funzioni della libreria FP Connect in essa incapsulate sono: "*PortOpen*", "*PortClose*", "*ReadSharedMemory*", "*WriteSharedMemody*", "*WriteBits*". Ora passeremo all'analisi dei metodi implementati:

- Metodo "apriComunicazionePLC": ha il compito di aprire la comunicazione con il PLC tramite protocollo RS 232 richiamando la funzione PortOpen;
- Metodo "chiudiComunicazionePLC": ha il compito di chiudere la comunicazione con il PLC tramite il protocollo RS 232 richiamando la funzione PortClose;
- Metodo "LeggiSM": ha il compito di leggere la Shared Memory e salvarla nell'array list "TabPLC" rappresentante la struttura tabellare della stessa. Ciò viene fatto richiamando la funzione "*ReadSharedMemory*" messa a disposizione dalla libreria FP Connect. La lettura avviene dal primo indirizzo disponibile, ovvero il 02H per l'asse X e lo 0CH per l'asse Y; alla fine di ogni banco si passa al successivo; gli indirizzi vengono letti di 16 in 16 e memorizzati nell'array list; ogni elemento dell'array list viene considerato come una tabella composta dai 16 indirizzi letti i quali vengono rappresentati sotto forma di campi della classe "TabellaPLC". Per effettuare la lettura delle caratteristiche di posizionamento implementate da due campi, vengono letti i campi stessi, convertiti in esadecimale, concatenati e riletti da esadecimale a decimale. In questo modo si ottiene una corretta rappresentazione di ciò che è memorizzato nella Shared Memory. Alla fine della lettura, gli indirizzi letti vengono stampati a video.
- Metodo "ScriviSM": ha il compito di scrivere nella Shared Memory i dati riprodotti nell'array list "TabPLC" rappresentante la struttura tabellare della stessa. Ciò viene fatto richiamando la funzione "*WriteSharedMemory*" messa a disposizione dalla libreria FP Connect. La scrittura avviene dal primo indirizzo disponibile, ovvero il 02H per l'asse X e lo 0CH per l'asse Y; alla fine di ogni banco si passa al successivo; ogni elemento dell'array list "TabPLC" rappresenta 16 indirizzi della Shared Memory: questa è quindi scritta di 16 in 16. Per effettuare la scrittura delle caratteristiche di posizionamento implementate da due campi, il valore da scrivere viene convertito in esadecimale e spezzato rispetto alla 4 cifra esadecimale; i due valori così ottenuti vengono riconvertiti in interi e salvati nella locazione opportuna. Se non vi sono stati errori in scrittura, viene stampato un messaggio a console che comunica il successo dell'operazione.
- Metodo "alzaBitDiRicalcolo": per rendere effettive le operazioni di scrittura effettuate sulla Shared Memory è necessario "alzare il bit di ricalcolo": questa operazione avviene attraverso la modifica del contatto "Y107" al valore "1"; segue la lettura del contatto "X107" il quale indica se l'esecuzione è andata a buon fine; in caso affermativo il contatto all'indirizzo "Y107" viene riportato a "0". Per far ciò viene utilizzato la funzione "*ReadBits*" e "*WriteBits*" messe a disposizione dalla libreria FP Connect.

Di seguito viene riportato il diagramma di flusso rappresentante la logica ad alto livello delle principali operazioni svolte dalle classi demandate a gestire la scrittura della struttura "TabPLC" nella Shared Memory.

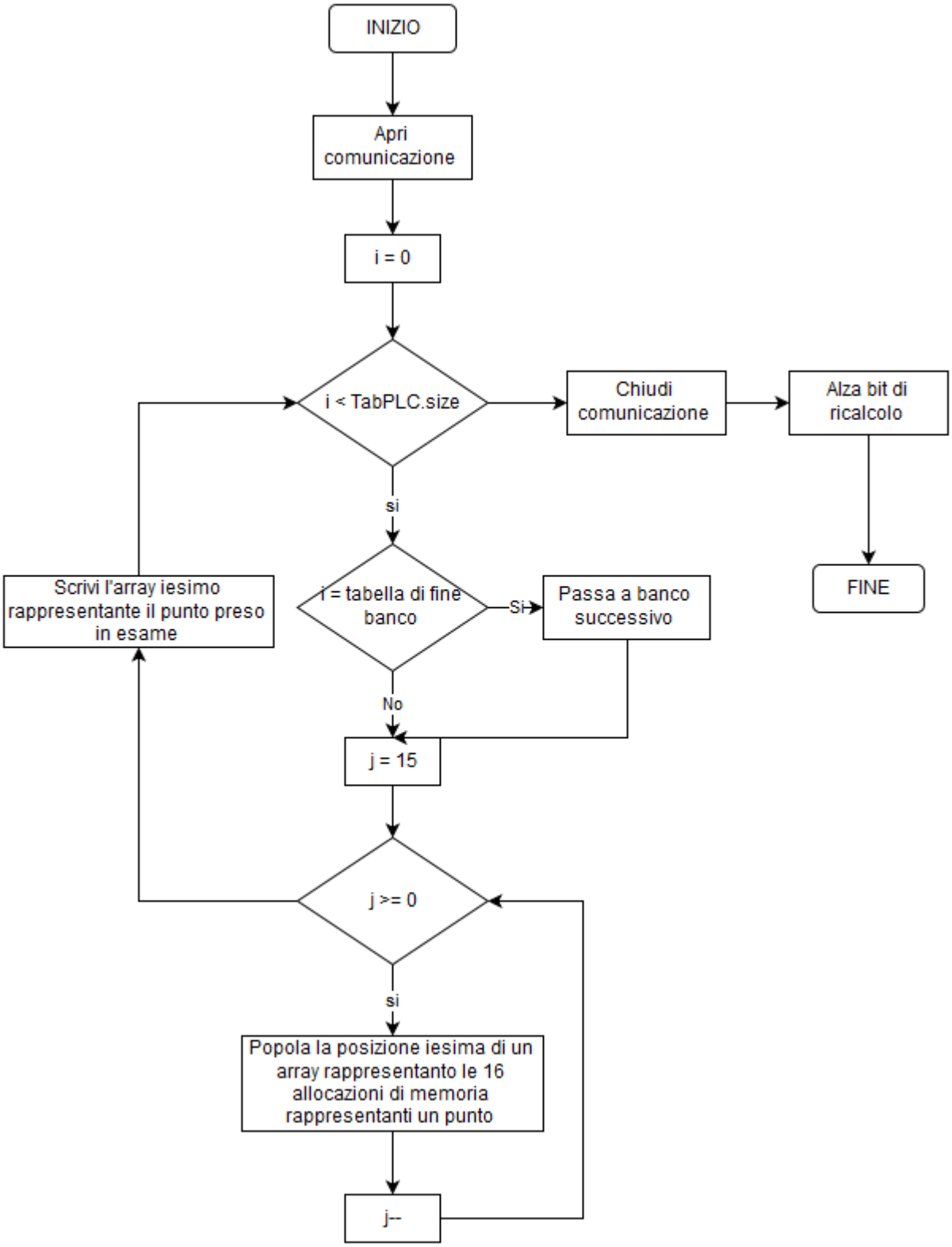

Figura 4.4: Diagramma di flusso rappresentante le principali funzioni implementate per la scrittura della struttura “TabPLC” nella Shared Memory.

Ora passiamo all’analisi della classe incaricata della gestione dei dati.

4.2.4 Classe gestione dati

La classe “GestioneFile” ha il compito di salvare i dati letti dalla Shared Memory o convertiti dal DXF e di caricare gli stessi. Il formato è stato chiamato “.rwsm” ed è leggibile anche come

formato .txt: questo per permettere all'utente di poter modificare il file anche tramite una semplice applicazione "blocco note".

La struttura del file ricalca perfettamente quella della Shared Memory, rappresentando riga per riga ogni singola tabella. I valori sono divisi dal carattere "," (virgola) e ogni valore rappresenta quindi una locazione di memoria della Shared Memory stessa. Una riga riproduce quindi un singolo elemento dell'array list "TabPLC"; questi elementi, come già detto, fanno parte della classe "TabellaPLC".

Il salvataggio del file prevede la lettura dell'array list "TabPLC" e la scrittura dello stesso in un nuovo file. Il caricamento del file prevede la lettura riga per riga del file e salvataggio di ognuna di esse in ogni singola posizione dell'array list "TabPLC".

Ora passiamo all'analisi dell'interfaccia grafica implementata.

4.2.5 Interfaccia grafica

L'interfaccia grafica è stata scritta e gestita tramite linguaggio XAML, di cui VisualStudio ne implementa anche una progettazione grafica in stile CSS. La scelta è stata giustificata dalla sua perfetta integrazione con C#.

Il passaggio da progettazione a console a risultato grafico passa attraverso la creazione di una libreria chiamata RWSharedMemoryLibrary la quale contiene tutto il codice implementato ad esclusione delle stampe a console e di quelle parti utilizzate per il controllo di progettazione. Questa libreria è stata poi richiamata nella creazione della finestra grafica del software. La finestra grafica è rappresentata in figura 4.1; essa implementa le seguenti soluzioni:

- Form di richiamo del file DXF da convertire con relativo pulsante "Sfoglia";
- Pulsante "Leggi DXF" atto a convertire il file richiamato sotto forma di più oggetti "TabellaPLC" memorizzati nella lista "TabPLC" e stampati a video nella griglia sottostante;
- Pulsante "Scrivi DXF su Shared Memory" atto a scrivere il DXF precedentemente convertito nella Shared Memory (nella scrittura è compresa l'azione di "alzata" del bit di ricalcolo);
- Pulsante "Leggi Shared Memory" atto a leggere la Shared Memory e a memorizzarne il risultato nella lista "TabPLC" poi stampata a video nella griglia sottostante;
- Pulsante "Azzera Shared Memory" atto ad azzerare la Shared Memory ai valori di default;
- Pulsante "Salva" e "Carica" atti a salvare o caricare un file rappresentante la struttura della Shared Memory letta dalla stessa o convertita in DXF; il file viene salvato in formato ".rwsm".

In particolare, la struttura tabellare nella quale viene dato come output il file DXF o la lettura della Shared Memory in forma tabellare, è ispirato al software Contro Configurator PM: ciò permetterà un facile adattamento da parte dell'utente utilizzante il software già esistente.

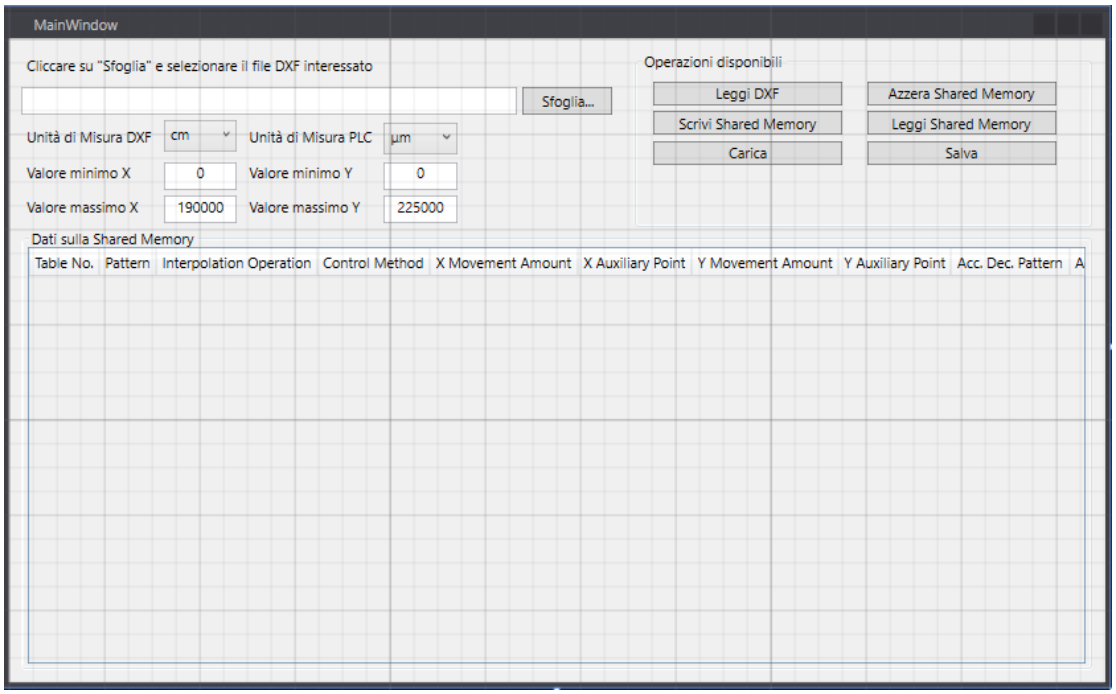

Figura 4.1: finestra utente del software "RWSharedMemory" allo stato finale.

5 Conclusioni

In questa tesi si è affrontato il problema di implementazione di un software che permettesse di istruire un ugello movimentato da un PLC a due assi rispetto a un percorso da seguire. La risoluzione del problema è passata attraverso la richiesta di studiare ed estrapolare la struttura del file DXF, file creabile con una comune applicazione CAD atto a rappresentare il percorso da eseguire col PLC; allo studio del file DXF è seguito lo studio della struttura interna della memoria PLC chiamata Shared Memory: ogni istruzione di movimentazione estrapolata dal DXF viene poi scritta nella Shared Memory indirizzo per indirizzo. Per effettuare l'operazione di lettura e scrittura della Shared Memory si è ricorso ad un controllo ActiveX, ovvero ad una libreria di funzioni resa disponibile dal produttore del PLC. Il risultato è un software semplice e dall'immediato utilizzo chiamato "RWSharedMemory", la cui struttura di rappresentazione del percorso ricalca il software del produttore del PLC Control Configurator PM; ciò permetterà in primo luogo una immediata ambientazione nell'uso da parte di quegli utenti i quali hanno dimestichezza col software del produttore; inoltre si avrà una duplice rappresentazione delle istruzioni date al PLC: la prima rappresentazione sarà grafica, rappresentata dal file DXF letto in ambiente CAD; la seconda numerica, rappresentata dal file DXF letto, interpretato e riprodotto sotto forma tabellare nel software RWSharedMemory e arricchito di quelle informazioni relative alla movimentazione dell'ugello assenti nel software CAD.

Il software è ulteriormente affinabile e migliorabile. Tale affinamento può essere ricercato e conseguito attraverso tre strade:

- La prima prevede un'ulteriore implementazione di RWSharedMemory in modo tale da renderlo sempre più simile a Control Configurator PM: ciò può essere raggiunto permettendo all'utente di modificare i dati riprodotti in tabella esattamente come accade col software del produttore del PLC. In secondo luogo può essere integrato uno strumento per il lancio dell'esecuzione dei dati scritti in Shared Memory da PC, come accade col software del produttore. Infine si potrebbe implementare una visualizzazione grafica dei dati estrapolati dal file DXF, con ulteriori affinamenti per quel che riguarda le informazioni rappresentabili (ad esempio, si potrebbero visualizzare i dati relativi alla velocità di esecuzione di una linea tramite il passaggio del puntatore del mouse sulla rappresentazione grafica della stessa).
- La seconda strada prevede l'integrazione del software implementato in Control Configurator PM: ciò eviterebbe la duplicazione di uno sforzo già compiuto e sarebbe un ottima integrazione al risultato già presente. Un'ulteriore evoluzione successiva a questa integrazione sarebbe quella di implementare una rappresentazione grafica del disegno seguendo le linee espresse al punto precedente;
- La terza strada prevede l'implementazione degli oggetti grafici da parte dell'hardware del PLC: infatti molte entità grafiche presenti nel DXF (come ad esempio l'ellissi) non sono implementate dalla parte hardware del PLC. Questa implementazione potrebbe dare al PLC una maggiore precisione e libertà di riproduzione grafica; la maggior libertà ricadrebbe anche sull'utente nella stesura del percorso o del "disegno" da riprodurre. Questa scelta dovrebbe però essere giustificata da una richiesta di mercato (ad esempio nel caso dell'utilizzo finale nel settore del packaging questo sforzo non è giustificato). Il risultato finale di un'evoluzione spinta in questo senso sarebbe molto vicino al mondo delle attuali stampanti 3D: infatti con l'aggiunta di un terzo asse con l'affinamento dell'implementazione hardware si avrebbe un prodotto in grado di spargere colla si ogni lato di un imballo, con massima libertà di percorso eseguibile.

Figura 5.1: In figura è rappresentato lo stato finale del software. Si è creato ed è stato messo a punto uno strumento di semplice e chiara gestione del PLC, il quale integra in esso la lettura del file DXF come richiesto. Tale software può essere infine un punto di partenza per implementazioni future, come suggerito nelle conclusioni.

Bibliografia

[1] Fonte: AutoCAD DXF, Wikipedia, indirizzo: https://it.wikipedia.org/wiki/AutoCAD_DXF, ultimo accesso: 26/11/15;

[2] Fonte: AutoDesk reference, AutoCAD 2014, indirizzo: http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=24240325, ultimo accesso: 26/11/15;

[3] Fonte: Controllore Logico Programmabile, Wikipedia, indirizzo: https://it.wikipedia.org/wiki/Controllore_logico_programmabile, ultimo accesso: 26/11/15;

[4] Fonte: IEC 61131-3, Wikipedia, indirizzo: https://en.wikipedia.org/wiki/IEC_61131-3, ultimo accesso: 26/11/15;

[5] Fonte: Manuale tecnico Panasonic FPΣ/FP2, indirizzo: https://www.panasonic-electric-works.com/cps/rde/xbcr/pew_eu_en/mn_63489_en_fpg_fp2_positioning_rtex.pdf, ultimo accesso: 26/11/15