

```

VAR_INPUT
Enable : BOOL := FALSE; (* FB enable *)
SpyOn : BOOL; (* Spy active *)
CAddress : @BYTE; (* Cell buffer address *)
CLength : UDINT; (* Cell buffer length (Accept SIZEOF) *)
Filename : STRING[ 32 ]; (* Path and name of file *)
Row : UINT; (* Row to read *)
Column : UINT; (* Column to read *)
END_VAR
    
```

```

VAR_OUTPUT
Enabled : BOOL; (* FB enabled *)
Found : BOOL := FALSE; (* Cell found *)
Done : BOOL; (* Execution done *)
Fault : BOOL; (* Execution fault *)
END_VAR
    
```

```

VAR
CaseNr : USINT; (* Program case *)
Ch : USINT; (* Read character *)
FColumn : UINT; (* File column *)
Fp : FILEP; (* File pointer *)
FPos : UDINT; (* File position *)
FRow : UINT; (* File row *)
i : INT; (* Auxiliary var *)
j : UDINT; (* Auxiliary var *)
Ptr : @USINT; (* General pointer *)
END_VAR
    
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "RdCellFromCSV" *)
3 (* ***** *)
4 (* Questo blocco funzione esegue la lettura di una cella da un file CSV. *)
5 (* Viene ritornata una stringa con il contenuto della cella. *)
6 (* *)
7 (* I parametri in ingresso sono: *)
8 (* Enable: Attivandolo viene eseguita la lettura cella. *)
9 (* SpyOn: Attivandolo viene eseguito lo spionaggio. *)
10 (* CAddress: Indirizzo buffer cella. *)
11 (* CLength: Dimensione buffer cella. *)
12 (* Filename: Percorso e nome del file in cui eseguire la ricerca. *)
13 (* Column: Colonna del file. *)
14 (* Row: Riga del file. *)
15 (* Length: Lunghezza buffer cella. *)
16 (* *)
17 (* I parametri in uscita sono: *)
18 (* Enabled: FB abilitato. *)
19 (* Found: Cella trovata (One shot). *)
20 (* Done: Ricerca terminata. *)
21 (* Fault: Errore nella ricerca (One shot). *)
22 (* ----- *)
23
24 (* ----- *)
25 (* INIZIALIZZAZIONI *)
26 (* ----- *)
27 (* Eseguo Spy per averlo attivo da Telnet e gestisco "Fault". *)
28
    
```

Project : CSVFileReader	
FUNCTION BLOCK : RdCellFromCSV	
Release : TagReader	Ver :1.00
Author :	Date:12/03/2016
Note :	Page:1 of 5

FUNCTION_BLOCK RdCellFromCSV

```

29 IF (SpyOn) THEN i:=SysSpyData(0, 0, 0, 0); END_IF;
30 IF (Fault) THEN Fault:=FALSE; Done:=TRUE; END_IF;
31
32 (* Eseguo controllo se FB abilitata. *)
33
34 IF NOT(Enable) THEN Enabled:=FALSE; Done:=FALSE; RETURN; END_IF;
35
36 (* Eseguo inizializzazioni su abilitazione. *)
37
38 IF NOT(Enabled) THEN
39     Enabled:=TRUE; (* FB enabled *)
40     CaseNr:=0; (* Program case *)
41     FPos:=0; (* File position *)
42     FRow:=0; (* File row *)
43     FColumn:=0; (* File column *)
44 END_IF;
45
46 (* Se attivo Done esco, occorre disabilitare e poi riabilitare l'FB per *)
47 (* eseguire una nuova ricerca. *)
48
49 IF (Done) THEN RETURN; END_IF;
50
51 (* Eseguo controllo se console di spionaggio è libera. *)
52
53 IF (SpyOn) THEN IF NOT(SysSpyData(0, 0, 0, 0)) THEN RETURN; END_IF; END_IF;
54
55 (* ----- *)
56 (* ESEGUO CONTROLLO RIGA ATTUALE *)
57 (* ----- *)
58 (* La riga attuale non può essere maggiore della riga richiesta. *)
59
60 IF (FRow > Row) THEN
61     IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Er'), ADR('Wrong Row')); END_IF;
62     Fault:=TRUE; (* Execution fault *)
63     RETURN;
64 END_IF;
65
66 (* ----- *)
67 (* ESEGUO APERTURA FILE *)
68 (* ----- *)
69 (* Apertura file in read. Se non esiste errore, il file rimane aperto *)
70 (* solo durante l'esecuzione della FB, uscendo viene sempre chiuso. *)
71
72 Fp:=Sysfopen(Filename, 'r'); (* File pointer *)
73 IF (Fp = NULL) THEN
74     IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Er'), ADR('File not found')); END
75 _IF;
76     Fault:=TRUE; (* Execution fault *)
77     RETURN;
78 END_IF;
79
80 (* Imposto posizione in cui leggere nel file. *)
81
82 IF (Sysfseek(Fp, TO_DINT(FPos), ID_SEEK_SET) = EOF) THEN
83     IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Er'), ADR('Row not found')); END_
84 IF;
85     Fault:=TRUE; (* Execution fault *)
86     i:=Sysfclose(Fp); (* Eseguo chiusura file *)
87     RETURN;
88 END_IF;

```

Project : CSVFileReader	
FUNCTION BLOCK : RdCellFromCSV	
Release : TagReader	Ver :1.00
Author :	Date:12/03/2016
Note :	Page:2 of 5

```

87
88 (* ----- *)
89 (* ESEGUO GESTIONE SEQUENZE PROGRAMMA *)
90 (* ----- *)
91 (* Gestione sequenze programma. *)
92
93 CASE (CaseNr) OF
94
95 (* ----- *)
96 (* ESEGUO RICERCA RIGA DA LEGGERE *)
97 (* ----- *)
98 (* Controllo se riga richiesta. *)
99
100 0:
101 IF (FRow >= Row) THEN CaseNr:=CaseNr+1; i:=Sysfclose(Fp); RETURN; END_IF;
102
103 (* Eseguo lettura intera riga fino a <CR><LF>.Limite lunghezza riga *)
104 (* a 1024 caratteri. *)
105
106 Ch:=TO_USINT(Sysfgetc(Fp));
107 FOR j:=0 TO 1024 DO
108     IF (Ch <> 16#0D) THEN Ch:=TO_USINT(Sysfgetc(Fp)); END_IF;
109     IF (Ch = 16#0D) THEN
110         Ch:=TO_USINT(Sysfgetc(Fp));
111         IF (Ch = 16#0A) THEN
112
113             (* Terminato lettura riga incremento posizione file su *)
114             (* inizio riga seguente, +2 per puntare dopo <CR><LF>. *)
115
116             FRow:=FRow+1; (* File row *)
117             FPos:=FPos+j+2; (* File position *)
118             i:=Sysfclose(Fp); (* Eseguo chiusura file *)
119             RETURN;
120         END_IF;
121     END_IF;
122 END_FOR;
123
124 (* Arrivo se errore lunghezza riga. *)
125
126 IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Er'), ADR('Row too long')); END_I
F;
127 Fault:=TRUE; (* Execution fault *)
128 i:=Sysfclose(Fp); (* Eseguo chiusura file *)
129 RETURN;
130
131 (* ----- *)
132 (* ESEGUO RICERCA COLONNA DA LEGGERE *)
133 (* ----- *)
134 (* Arrivo posizionato nel file all'inizio della riga da leggere. Devo *)
135 (* cercare la colonna indicata, le colonne sono separate da ",". *)
136 (* Controllo se colonna richiesta. *)
137
138 1:
139 IF (FColumn >= Column) THEN CaseNr:=CaseNr+1; i:=Sysfclose(Fp); RETURN; END_IF;
140
141 (* Limite lunghezza riga a 1024 caratteri. *)
142
143 FOR j:=0 TO 1024 DO
144     Ch:=TO_USINT(Sysfgetc(Fp));
145

```

Project : CSVFileReader	
FUNCTION BLOCK : RdCellFromCSV	
Release : TagReader	Ver :1.00
Author :	Date:12/03/2016
Note :	Page:3 of 5

FUNCTION_BLOCK RdCellFromCSV

```

146      (* Controllo se fine riga. *)
147
148      IF (Ch = 16#0D) THEN EXIT; END_IF;
149
150      (* Controllo se carattere separatore di colonna . *)
151
152      IF (Ch = 16#2C) THEN
153          FColumn:=FColumn+1; (* File column *)
154
155          (* Controllo se colonna richiesta. *)
156
157          IF (FColumn = Column) THEN
158              FPos:=FPos+j+1; (* File position *)
159              CaseNr:=CaseNr+1; (* Program case *)
160              i:=Sysfclose(Fp); (* Eseguo chiusura file *)
161              RETURN;
162          END_IF;
163      END_IF;
164  END_FOR;
165
166  (* Arrivo se non trovata colonna. *)
167
168  IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Er'), ADR('Column not found')); E
ND_IF;
169  Fault:=TRUE; (* Execution fault *)
170  i:=Sysfclose(Fp); (* Eseguo chiusura file *)
171  RETURN;
172
173  (* ----- *)
174  (* TROVATA CELLA INDICATA RITORNO CONTENUTO *)
175  (* ----- *)
176  (* Arrivo posizionato nel file sul contenuto della cella indicata. *)
177  (* Ritorno contenuto cella. *)
178
179  2:
180  Ptr:=CBAddress; (* General pointer *)
181  FOR j:=0 TO CBLength DO
182      Ch:=TO_USINT(Sysfgetc(Fp));
183      @Ptr:=Ch; (* Trasferisco carattere in buffer cella *)
184
185      (* Controllo se carattere separatore di colonna o fine riga. *)
186
187      IF ((Ch = 16#0D) OR (Ch = 16#2C)) THEN
188          @Ptr:=0; (* Inserisco tappo *)
189          Found:=TRUE; (* Cell found *)
190          Done:=TRUE; (* Execution done *)
191          i:=Sysfclose(Fp); (* Eseguo chiusura file *)
192          IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Ok'), ADR('Cell read'));
END_IF;
193          RETURN;
194      END_IF;
195
196      (* Incremento pointer a buffer cella. *)
197
198      Ptr:=Ptr+1; (* General pointer *)
199  END_FOR;
200
201  (* Arrivo se contenuto cella più grande del buffer. *)
202
203  IF (SpyOn) THEN i:=SysSpyData(SPY_ASCII, 16#40000000, ADR('Er'), ADR('Cell too long')); END_

```

Project : CSVFileReader	
FUNCTION BLOCK : RdCellFromCSV	
Release : TagReader	Ver :1.00
Author :	Date:12/03/2016
Note :	Page:4 of 5

FUNCTION_BLOCK RdCellFromCSV

```

IF;
204     Fault:=TRUE; (* Execution fault *)
205     i:=Sysfclose(Fp); (* Eseguo chiusura file *)
206     RETURN;
207     END_CASE;
208
209 (* [End of file] *)
210
211

```

	Project : CSVFileReader	
	FUNCTION BLOCK : RdCellFromCSV	
	Release : TagReader	Ver :1.00
	Author :	Date:12/03/2016
	Note :	Page:5 of 5