Content

Overview

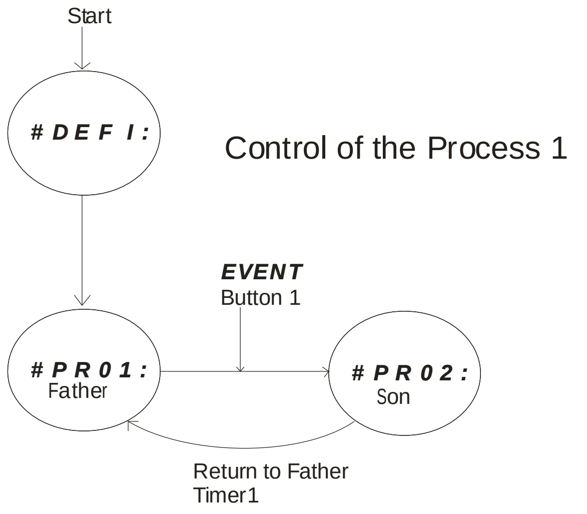
The files on the MMC-Card of the MR370 are stored in FAT32 file format. These files are for reading, writing or to control MR370. They could be updated by the communication from host (PC) to the terminal MR370.
The control file relates to 9 events of transponder, keys and files as inputs; buzzer, LCD, backlight and files as outputs.

This Information

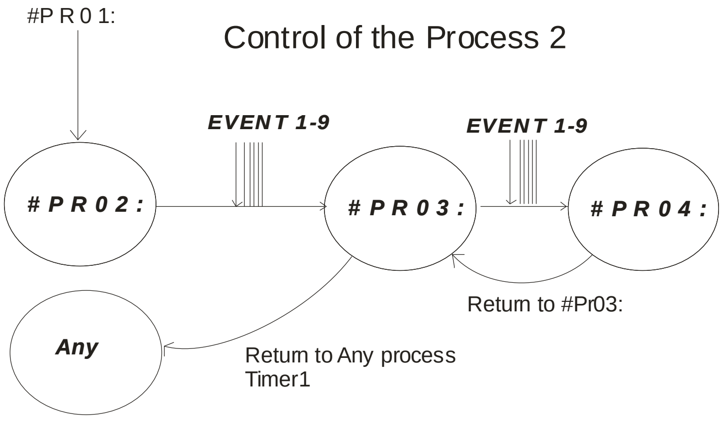
The new user should read all chapters from the beginning. The control file (PROCESS) of the MR370 can be changed by using an editor. For the development we recommend our windows program “MR370 process builder” (available September 2007)

Control of MR370

The MR370 is controlled by process sections which are in the file “PROCESS” on the MMC-card.
Each process section is triggered by an event.



All process sections have the same form of 54 lines and a section title ‘PR01: - #PR99.
After switching on the MR370, the section #DEFI: (for the global values) is executed. Then #PR01: is the first main process.
Only a “Event” can change the current situation and lead to another process (#PR02:). By timer1 it will return to the father process.
The 2nd picture shows a more complex relation of a call and its return by its defined events.
Each process section has a definition area for 9 events (line 44), where the transitions to the next processes are to be defined by its section title. The sub-process (son-) returns to the main process (father-) or any process by its timer1 – event.



Six Process Types

1.) #DEFI:
Section name for the global values, like file names; values for the daily signals for work breaks; times for the back light LED..; 27 lines are reserved to this section. After restart this section is executed and three times daily: 06:00; 12:00; 18:00;

2.) #PR01:
Name for the main section by default – father process, which reserves 54 lines. After restart of the terminal MR370 it is executed.

3.) #PR02: to #PR98:
Name for a sub-process – child process.
Normally – by default - all sub-processes (son-) return to the father-process after the timer1-delay of max. 10 sec;

4.) timer2
To assign a new main process which may use any name of “#PR01: to #PR98:”.

5.) #PR99:
This reserved section name is for the control of the access of persons to doors and gates. External transponder readers are switched to MR370. An additional ADD-ON-BOARD is necessary.

6.) Menu Section
This section is not defined by its section name. The use of certain input variables like “$xn or $yn” assign functions to the 3 push buttons 2,3,4, to change the values of the input variables.

File Convention

All files names are restricted to 4 character with exception: process to 7,

Input files

B001 .. B999 read file; picture 128x 64 in MR370 – format for the output of the LCD;
The bmp–format must be converted to the MR370-format
by the program “convert”
T001 .. T999 read file; text of 8 lines for the output of the LCD

ER01 read file; to show the message on the LCD, if there is no permission of access
or a transponder error

PERS read file; record format; contains personal data; primary key is the transponder
number (card-ID)

AUFT read file; record format; contains the job data for the workmen;
the primary key is the transponder number (card-ID)

ZM01 .. ZM99 read file; record format; contains data for the permission of access to the gates and
doors; for an add-on board and at least one transponder reader

PROCESS read file; contains process data to control MR370; with the default section
“#DEFI:“ and the main process #PR01: and any processes up to #PR99:

Output file

BUCH output file; record format; for the registration of the workman/employee
with his card ID (transponder number) , the current date and time

Control the MR370 by the file “PROCESS”

The following examples explain the process sections which control the MR370. Behind each section label the order of the values must be strictly to observed, because each value is determined by its line number, relatively to the label for example:. #DEFI: or #PR01:
The following labels are significant and must have these syntax (complete with “#” and “:”):

#DEFI:	for the default values
#PR01:	the main-process; first call; sub-process returns to this process after timer1 completes
#PR99:	background process for access control

#DEFI: The Default Section

This is the basic section with the definition of the file names, the display line position of the clock with date and time, two values of the row-position for date = 1 and time = 93.
The plant siren (signal relays #8) sounds for the work breaks; they are defined here by there time values. It is switched off by zero (#8)

#…line position relative in section   important values   important filename definitions

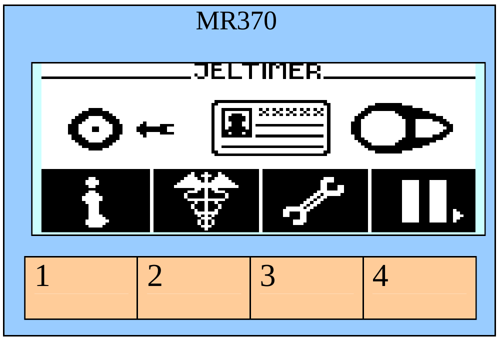
#	Value / keyword	Description
0	#DEFI:	
1	50	back light on = 0 .. 100ms
2	50	relays on time
3	50	timer 1; process timeout; returns to called process (father-)
4	255	reserved (for clock refresh)
5	1	clock (date and time) in line 1 .. 8 ; 0 = visible off
6	1	clock date output; at column (position x) 1 .. 100 ; 0 = visible off
7	93	clock time output; at column ; 0 = visible off
8	0	control= 0; +1 = auto access ; +2 = signal relays on
9	0	reserved
10	buch	registration file (time & attendance, jobs)
11	pers	personal data; in records of 512 bytes
12	auft	job information; in records of 512 bytes
13	zpln	reserved , plan data
14	zmod	reserved, access data
15	8	signal relays at 8 o´ clock, hrs
16	30	1. signal at (+ 30 min + 8 hrs)=8:30
17	60	2. signal at + 60 min (8:30 + 60)=9:30
18	15	3. signal at + 15 min (9:30 + 15)=9:45
19	135	4. signal at +120 min (9:45 + 135)=12:00
20	30	5. signal at + 30 min (12:00+30)=12:30
21	180	6. signal at + 90 min (12:30+180)=15:30
22	15	7. signal at + 15 min (15:30 + 15)= 15:45
23	90	8. signal at + 90 min (15:45 + 90)= 17:15
24	127	day of week: So=+1; MO=+2; ... SA=+64
25	Good Morning	greeting text (max 13 chars) for 0:01 to 12:00 o’clock
26	Good Afternoon	for 12:01 to 18:00 o’clock
27	Good Evening	for 18:01 to 24:00 o’clock

#PR01: MAIN PROCESS – after the start

Reserved name of the main process. This section will be executed at the start of MR370 by switching power on.

The process starts with clear screen, beep (buzzer) of 0,1 sec, switch on back light LED for 8 sec; load and show MR370 graphic file "B001"; show date and time in the first line. At last the new events for the buttons are defined, also for timer 1+2, for the transponder (and modem / barcode reader). So a read of a card-ID (transponder) will change the process status; it will change to the process section "#PR02:"; a push on button 2 leads to "#PR03:"; a simultaneous push on buttons 2+3 leads to "#PR00:" (#50).

"B001" in main section ">PR01:" #42.



2+3	>#PR00:

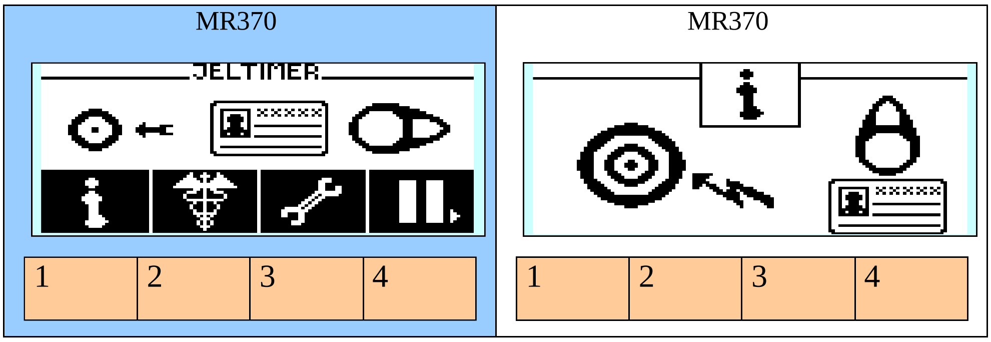
>#PR01:	>#PR03:	>#PR04:	>#PR05:

Any key	>#PR01:
Timer1	Timer1
Timer2	Timer2
Transponder	>#PR02 :

Assign of Events

> prefix
PR+2 digits number: 01-98
: postfix
example : #PR02:
Reserved Function
\>NULL No process
\>OLD previous process

#	Value / keyword	Description
44	>#PR02:	Event of transponder read leads to process "#PR02:"
45	>#PR01:	Event of any button (1..4) to "#PR01:" = main; as default; (here no sense)
46	>#PR02:	Event of button "1" (A) to process = "#PR02:"
47	>#PR03:	Event of button "2" (B) to "#PR03:"
48	>#PR04:	Event of button "3" (C) to "#PR04:"
49	>#PR05:	Event of button "4" (D) to "#PR05:"
50	>#PR00:	Event of 2 buttons "2+3" (B+C) to "#PR00:"= setup
51	>#NULL	Event for timer2 = "#NULL" = off; Timer2 = 0 to 100 (0,1 sec)
52	>#NULL	Event for timer1 (in #DEFI: set) = off



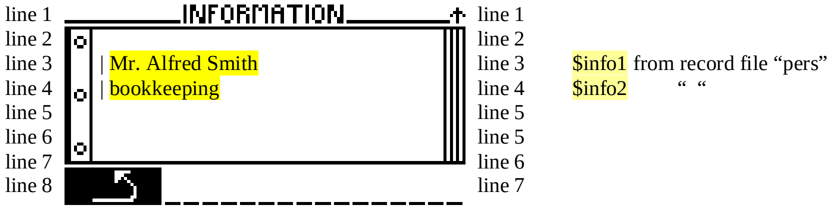
Father process à à à à à à à à son process

A push to button 1 leads to process section "#PR01:"

#	Value/keyword	Description
0	#PR01:	Main Menu
1	0	timer2 : 0 .. 255 (* 0,1 ms); if set, leads to event for timer2 as new main process
2	1	clear screen = 1 ; 0 = no;
3	1	beep : 1 .. 10 (* 0,1 ms); 0 = off
4	80	switch on back light; time = 0 .. 100 (* 0,1 ms); 0=off
5	0	relays switching time for on 1 .. 50 (* 0,1 ms); 0 = off
6	0	relays off = 1; 0 = no action
7	0	write to file “buch” : 0=off; 1=attendance; 2 = job; 4 = access
8	0	read from file; 0=off; 1=pers; 2=auft; +10=repeat last record
9	1	clock line: 1 .. 8; 0= off ; 100 = BIG TIME digits
10	0	clear cursor for line 1 to n: 0 off; n:1 - 120
11	0	clear cursor for line 2 to 8 : 0 = off
12	0	…
13	0	…
14	0	…
15	0	…
16	0	…
17	0	…
18	0	number of white space for line 1 = 0 ..120 (pts)
19	0	no white space for line 2
20	0	...
21	0	...
22	0	...
23	0	...
24	0	...
25	0	no white space for line 8
26	0	cursor position for text line 1= zero; (0 ..120 pts from left)
27	0	cursor position for text line 2 = zero
28	0	...
29	0	…
30	0	…
31	0	…
32	0	…
33	0	cursor position for text line 8 = zero
34	0	text output for line 1: (record variables and text; 0 to 25 chars)
35	0	text output for line 2 to 8 : 0 = off
36	0	…
37	0	…
38	0	…
39	0	…
40	0	…
41	0	…
42	B001	MR370 graphic file to LCD screen
43	NULL	No text file ; (T001 to T999)
44	>#PR02:	Event of transponder read leads to process “#PR02:”
45	>#PR01:	Event of any button (1..4) to “#PR01:” = main; as default
46	>#PR02:	Event of button “1” (A) to process = “#PR02:”
47	>#PR03:	Event of button “2” (B) to “#PR03:”
48	>#PR04:	Event of button “3” (C) to “#PR04:”
49	>#PR05:	Event of button “4” (D) to “#PR05:”
50	>#PR00:	Event of 2 buttons “2+3” (B+C) to “#PR00:”= setup
51	>#NULL	Event for timer2 = “#NULL” = off; Timer2 = 0 to 100 (0,1 sec)
52	>#NULL	Event for timer1 (in #DEFI: set) = off
53	>#NULL	Event for additional readers of access control (optional)
54	>#NULL	Event for one barcode reader / modem (optional)

#PR02: *Reading of a Card - ID*

This section “#PR02:” which is related to the event of reading a card – ID (transponder tag) in section “#PR01:”, demonstrates, how to retrieve personal data for a card - ID. It reads the corresponding record from the file “PERS”. The process starts with switching on the backlight LED, a beep of 0,1 sec, output the graphic file “B002” to LCD screen and then the text file “T002”. The record fields “info1” with the name and “info2” are displayed. The text file overwrites the graphic part, and the text variables (as follows) overwrite the previous display output.
So standard text from file and standard graphic from file can be merged with any variables. The record variables in display lines 1 .. 8 may be merged with other text and variables like date, time, etc.
Display lines 3 + 4 (#20+#21) are cleared by 100 x white spaces from position 20 pts (#12+#13) to prepare the output of “$info1” and “$info2” (#36+#37). “$info1! And “$info2” are record variables of the file “PERS”. After 5 sec (see section “#DEFI:” #3) this sub-process returns to the main section “#PR01:” (see “Event for timer1” #52).



“B002” in main section “>PR02:” #42.

Partial section

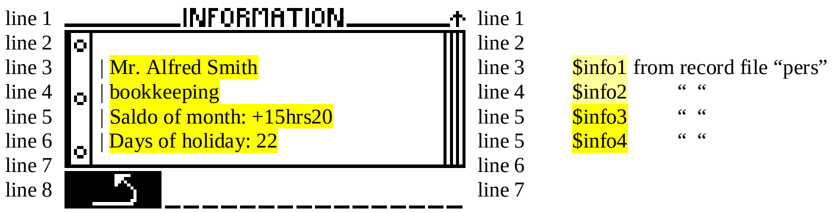
#	Value/keyword	Description
0	#PR02:	
1	0	Timer2 = 0 = off
…	…	
28	20	text cursor for line 3 is at 20 pts (from left)
29	20	text cursor for line 4 is at 20 pts (from left)
30	0	text cursor for line 5 is at zero (from left)
31	0	…
32	0	…
33	0	text cursor for line 8 is at zero (from left)
34	0	text output for line 1+2 = 0 = off
35	0	…
36	$info1	text output for line 3+4: record field “$info1” and “$info2” of file “pers”
37	$info2	…
38	0	text output for line 5 to 8 = 0 = off
39	0	…
40	0	…
41	0	…

Complete Section

#	Value/keyword	Description
0	#PR02:	
1	0	Timer2 = 0 = off
2	0	clear screen = : 0 = off
3	1	beep = yes for 0,1 sec
4	1	backlight LED = on
5	0	reserved (relays on = ignored)
6	0	reserved (relays off =ignored)
7	1	write record data with card-ID, date and time to the file “buch”
8	1	read record from file “pers”
9	0	show clock = off
10	0	clear cursor for line 1 to 0 = off
11	0	…
12	20	clear cursor for line 3 to 20
13	20	clear cursor for line 4 to 20
14	0	clear cursor for line 5 to 0 = off
15	0	…
16	0	…
17	0	clear cursor for line 8 to 0 = off
18	0	no white space for line 1
19	0	no white space for line 2
20	100	100 pts white space for line 3 from position 20 (see line 12)
21	100	100 pts white space for line 4 from position 20 (see line 13)
22	0	no white space for line 5
23	0	no white space for line 6
24	0	…
25	0	no white space for line 8
26	0	cursor position for text in line 1 is at zero (from left)
27	0	..
28	20	text cursor for line 3 is at 20 pts (from left)
29	20	text cursor for line 4 is at 20 pts (from left)
30	0	text cursor for line 5 is at zero (from left)
31	0	…
32	0	…
33	0	text cursor for line 8 is at zero (from left)
34	0	text output for line 1+2 = 0 = off
35	0	…
36	$info1	text output for line 3+4: record field “$info1” and “$info2” of file “pers”
37	$info2	…
38	0	text output for line 5 to 8 = 0 = off
39	0	…
40	0	…
41	0	…
42	B002	B002 – MR370- graphic file to LCD screen
43	T002	T002 – text file with to screen (overwrites graphic B002)
44	>#NULL	Event of transponder read = “>#NULL” = off
45	>#NULL	Event of any button (1..4) = “#NULL”
46	>#PR02:	Event of button “1” (A) to process = “#PR02:”
47	>#NULL	Event of button “2” (B) to process = “#NULL”
48	>#NULL	Event of button “3” (C) to process = “#NULL”
49	>#PR01:	Event of button “4” (D) to process = “#PR01:” = main
50	>#NULL	Event of 2 buttons “2+3” (B+C) to process = #NULL”
51	>#NULL	Event for timer2 = “#NULL” = off, when “Timer2” is set.
52	>#PR01:	Event for timer1 (in #DEFI: set), which leads to #PR01:
53	>#NULL	Event for additional readers of access control (optional)
54	>#NULL	Event for one barcode reader / modem (optional)

#PR03: Show Personal Data

This section “#PR03:” which is related to event of button 2 in section “#PR01:” (#47), shows another example how to show more personal data from a record of the file “PERS” . The process starts with switching on the backlight LED, a beep of 0,1 sec, and output graphic file B003 to LCD screen.
Display lines 3 to 6 are cleared by 100 x white spaces (#20..23) from position 20 (#12 ..15) pts.
Then the record variables “$info1” to “$info4” are displayed on the LCD lines 2 to 5.
After 5 sec (see section “#DEFI:” #3) this process returns to the main section “#PR01:” (see “event for timer1” #52).



line 1	INFORMATION	line 1	
line 2		line 2	
line 3	| Mr. Alfred Smith	line 3	$info1 from record file “pers”
line 4	| bookkeeping	line 4	$info2 “ “
line 5	| Saldo of month: +15hrs20	line 5	$info3 “ “
line 6	| Days of holiday: 22	line 5	$info4 “ “
line 7		line 6	
line 8		line 7	

“B002” in main section “>PR02:” #42.

Partial section

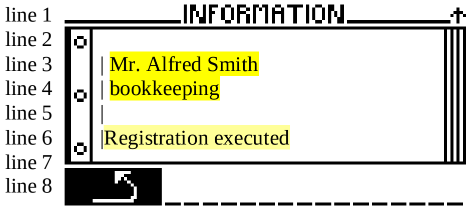
#	Value/keyword	Description
0	#PR02:	
1	0	Timer2 = 0 = off
…	…	
26	0	text cursor for line 1 is at zero (from left)
27	0	..
28	20	text cursor for line 3 to 6: 20 pts (from left)
29	20	…
30	20	…
31	20	…
32	0	text cursor for line 7+8 is at zero (from left)
33	0	…
34	0	text output for line 1 = 0 = off
35	0	…
36	$info1	text output for line 3- 6 : record field “$info1” to “$info4” of file pers
37	$info2	…
38	$info3	…
39	$info4	…
40	0	…
41	0	…

Complete Section

#	Value/keyword	Description
0	#PR03:	
1	0	Timer2 = 0 = off
2	0	clear screen = : 0 = off
3	1	beep = yes for 0,1 sec
4	80	backlight LED = on for 8 sec
5	0	reserved (relays on = ignored)
6	0	reserved (relays off =ignored)
7	0	No write record to buch – file
8	1	read record from pers file
9	0	show clock = off
10	0	clear cursor for line 1+2 to row position= 0=off
11	0	…
12	20	clear cursor for lines 3 to 6 : = 20
13	20	…
14	20	…
15	20	…
16	0	clear cursor for lines 7 + 8 = 0= off
17	0	…
18	0	no white space for line 1
19	0	no white space for line 2
20	100	100 pts white space for line 3 to 6 (for output text: see line # 36+37)
21	100	…
22	100	…
23	100	…
24	0	no white space for line 7 +8
25	0	…
26	0	text cursor for line 1 is at zero (from left)
27	0	..
28	20	text cursor for line 3 to 6: 20 pts (from left)
29	20	…
30	20	…
31	20	…
32	0	text cursor for line 7+8 is at zero (from left)
33	0	…
34	0	text output for line 1 = 0 = off
35	0	…
36	$info1	text output for line 3- 6 : record variables “$info1” to “$info4” of the file pers
37	$info2	…
38	$info3	…
39	$info4	…
40	0	…
41	0	…
42	B003	B003 – MR370- graphic file to LCD screen
43	NULL	no text file
44	>#PR02:	event of transponder read = “>#PR03”
45	>#NULL	event of any button (1..4) = “#NULL”
46	>#NULL	event of button “1” (A) to process = “#NULL”
47	>#NULL	event of button “2” (B) to process = “#NULL”
48	>#NULL	event of button “3” (C) to process = “#NULL”
49	>#PR01:	event of button “4” (D) to process = “#PR01:” = main
50	>#NULL	event of 2 buttons “2+3” (B+C) to process = #NULL”
51	>#NULL	event for timer2 = “#NULL” = off, when “Timer2” is set.
52	>#PR01:	event for timer1 (in #DEFI: set), which leads to #PR01:
53	>#NULL	event for additional readers of access control (optional)
54	>#NULL	event for one barcode reader / modem (optional)

#PR07: Registration and write to book - file

This example shows how to store booking data to the file “BUCH”.
The event for reading a card – ID (transponder reader) executes this section and writes to the LCD-display the two record field variables “$info1, $info2” of the file “PERS”. At last the registration data with the time stamp is written to the booking – file “buch”.



Mr. Alfred Smith
bookkeeping

Registration executed

line 1	
line 2	
line 3	$info1 from record file “PERS”
line 4	$info2 “ “
line 5	
line 5	Registration executed
line 6	
line 7	

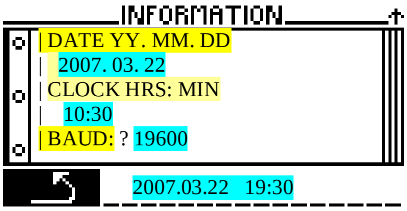
“B002” in main section “>PR07:” #42.

Partial section

#	Value/keyword	Description
0	#PR02:	Writes a record with booking data to the book file “buch”
1	0	timer2 = 0 = off
2	0	clear screen : 0 = off
3	5	beep for 0,5 sec
4	80	backlight LED = on for 8 sec
5	0	Reserved
6	0	Reserved
7	1	write booking data to booking – file “buch”
8	1	read record from file “pers”
9	0	clock line; show clock = off
…	0	
26	0	text cursor for line 1 is at zero (from left)
27	0	..
28	20	text cursor for line 3,4: 20 pts (from left)
29	20	…
30	0	
31	20	text cursor for line 6: 20 pts (from left)
32	0	text cursor for line 7+8 is at zero (from left)
33	0	…
34	0	text output for line 1 = 0 = off
35	0	…
36	$info1	text output for line 3- 4 : record field “$info1” to “$info4” of file pers
37	$info2	…
38	0	…
39	Registration executed	text output for line 6
40	0	…
41	0	…

#PR00: Setup Menu: Example for Input Values

This example “#PR00:” shows some setup variables.
I also demonstrates the use of input variables, which can be changed with the push buttons. The last 3 buttons 2,3,4 are reserved for this purpose. Button 2 to select an item; button 3 and 4 to increment and decrement the value.



line 1	INFORMATION	line 1	
line 2	| DATE YY. MM. DD	line 2	DATE YY.MM.DD
line 3	| 2007. 03. 22	line 3	$iy . $im . $id
line 4	| CLOCK HRS: MIN	line 4	CLOCK HRS:MIN
line 5	| 10:30	line 5	$ih : $mm
line 6	| BAUD: ? 19600	line 6	BAUD: $ibxxxx
line 7		line 7	
line 8	2007.03.22 19:30	line 8	$datexxxx $time

The input variables can be changed with the “up” and “down” buttons to increment and decrement the content of the focussed item (?). By the button “2” you can change the focus from one item to the next.

“B003” in main section “>PR02:” #42.

Partial section

#	Value/keyword	Description
0	#PR02:	
1	0	Timer2 = 0 = off
…	…	
26	0	text cursor for line 1 is at zero (from left)
27	12	text cursor for line 2 to 7 : 12 pts (from left)..
28	12	…
29	12	…
30	12	…
31	12	…
32	12	…
33	40	text cursor for line 8: 40 pts (from left)
34	0	In line 1: text output = 0 = off
35	DATE YY.MM.DD	In line 2: Output text
36	$iy . $im . $id	In line 3: text variables: $iy;$im;$id with text “ . “
37	CLOCK HRS:MIN	In line 4: Output text
38	$ih : $mm	In line 5: text variables $ih;$mm with text
39	BAUD: $ibxxxx	In line 6: $ibxxxx: $ib; “xxxx” reserves place
40	0	In line 7: text output = off
41	$datexxxx $time	In line 8: variables $date; $time

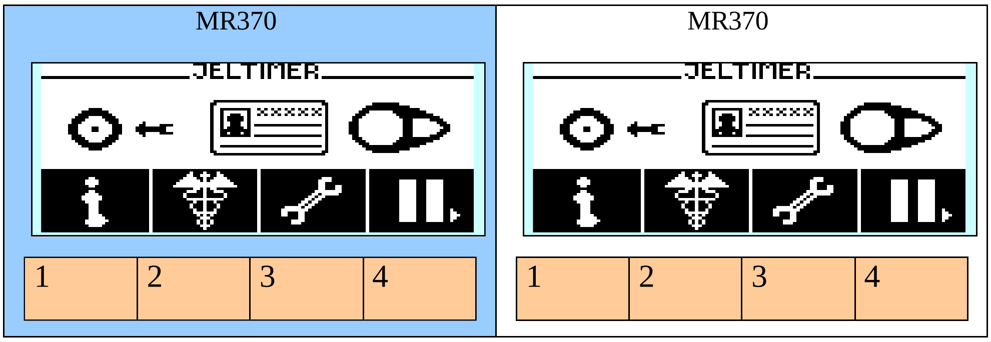
Complete Section

#	Value/keyword	Description
0	#PR00:	Setup – variables
1	50	Timer2 = 0 = off
2	0	clear screen = : 0 = off
3	1	beep = yes for 0,1 sec
4	80	backlight LED = on for 8 sec
5	0	reserved (relays on = ignored)
6	0	reserved (relays off =ignored)
7	1	no write record to buch – file
8	0	no read record from pers file
9	0	show clock = off
10	0	clear cursor for line 1+2 to row position= 0=off
11	0	…
12	12	clear cursor for lines 3 to 7 : = 12
13	12	…
14	12	…
15	12	…
16	12	…
17	12	clear cursor for lines 8 = 12
18	0	no white space for line 1
19	128	128 pts white space for line 2 to 6 (see line # 36+37)
20	128	…
21	128	…
22	128	…
23	128	…
24	128	no white space for line 7 +8
25	0	…
26	0	text cursor for line 1 is at zero (from left)
27	12	text cursor for line 2 to 7 : 12 pts (from left)..
28	12	…
29	12	…
30	12	…
31	12	…
32	12	…
33	40	text cursor for line 8: 40 pts (from left)
34	0	In line 1: text output = 0 = off
35	DATE YY.MM.DD	In line 2: Output text
36	$iy . $im . $id	In line 3: text variables: $iy;$im;$id with text " . "
37	CLOCK HRS:MIN	In line 4: Output text
38	$ih : $mm	In line 5: text variables $ih;$mm with text
39	BAUD: $ibxxxx	In line 6: $ibxxxx: $ib; "xxxx" reserves place
40	0	In line 7: text output = off
41	$datexxxx $time	In line 8: variables $date; $time
42	B003	B007 – MR370- graphic file to LCD screen
43	NULL	no text file
44	>#PR07:	event of transponder read = ">#PR03"
45	>#PR08:	event of any button (1..4) = "#NULL"
46	>#PR01:	event of button "1" (A) to process = "#NULL"
47	>#PR00:	event of button "2" (B) to process = "#NULL"
48	>#PR00:	event of button "3" (C) to process = "#NULL"
49	>#PR00:	event of button "4" (D) to process = "#PR01:" = main
50	>#PR00:	event of 2 buttons "2+3" (B+C) to process = #NULL"
51	>#NULL	event for timer2 = "#NULL" = off, otherwise depends of "Timer2".
52	>#PR01:	event for timer1 (in #DEFI: set), which leads to #PR01:
53	>#NULL	event for additional readers of access control (optional)
54	>#NULL	event for one barcode reader / modem (optional)

#PR99: Background process for Access Control

This is a reserved process section for the access control add-on-board "MR370-adapter". It handles up to 16 external transponder readers.

This process section is executed, when an external transponder reader sends a card-ID (transponder number) to be checked, if it is valid or not. The corresponding record is read and its access data is compared with the actual time and date. In this section is the first use of the parameter “>OLD” (#44…). Since this background process “>PR00:” should not overwrite the actual front process.



Front process #PR01:
- is interrupted for some milliseconds

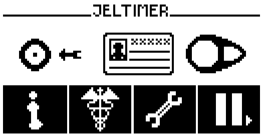
background process “#PR99:”
- has no affect to the front process

#	Value/keyword	Description
0	#PR99:	Access with add on board and external transponder readers
1	0	reserved
2	0	reserved
3	0	reserved
4	0	reserved
5	0	reserved
6	0	reserved
7	0	No write registration data in buch; 0 = off; 4 = write access and control data
8	1	Reserved; read record from file “pers”
9	0	reserved
10	0	reserved
11	0	reserved
12	0	reserved
13	0	reserved
14	0	reserved
15	0	reserved
16	0	reserved
17	0	reserved
18	0	reserved
19	0	reserved
20	0	reserved
21	0	reserved
22	0	reserved
23	0	reserved
24	0	reserved
25	0	reserved
26	0	reserved
27	0	reserved
28	0	reserved
29	0	reserved
30	0	reserved
31	0	reserved
32	0	reserved
33	0	reserved
34	0	reserved
35	0	reserved
36	0	reserved
37	0	reserved
38	0	reserved
39	0	reserved
40	0	reserved
41	0	reserved
42	NULL	reserved
43	NULL	reserved
44	>OLD	reserved for events: keep last calls to the process sections
45	>OLD	reserved
46	>OLD	reserved
47	>OLD	reserved
48	>OLD	reserved
49	>OLD	reserved
50	>OLD	reserved
51	>OLD	reserved
52	>OLD	reserved
53	>OLD	reserved
54	>OLD	reserved

Display Output of Graphic – Files

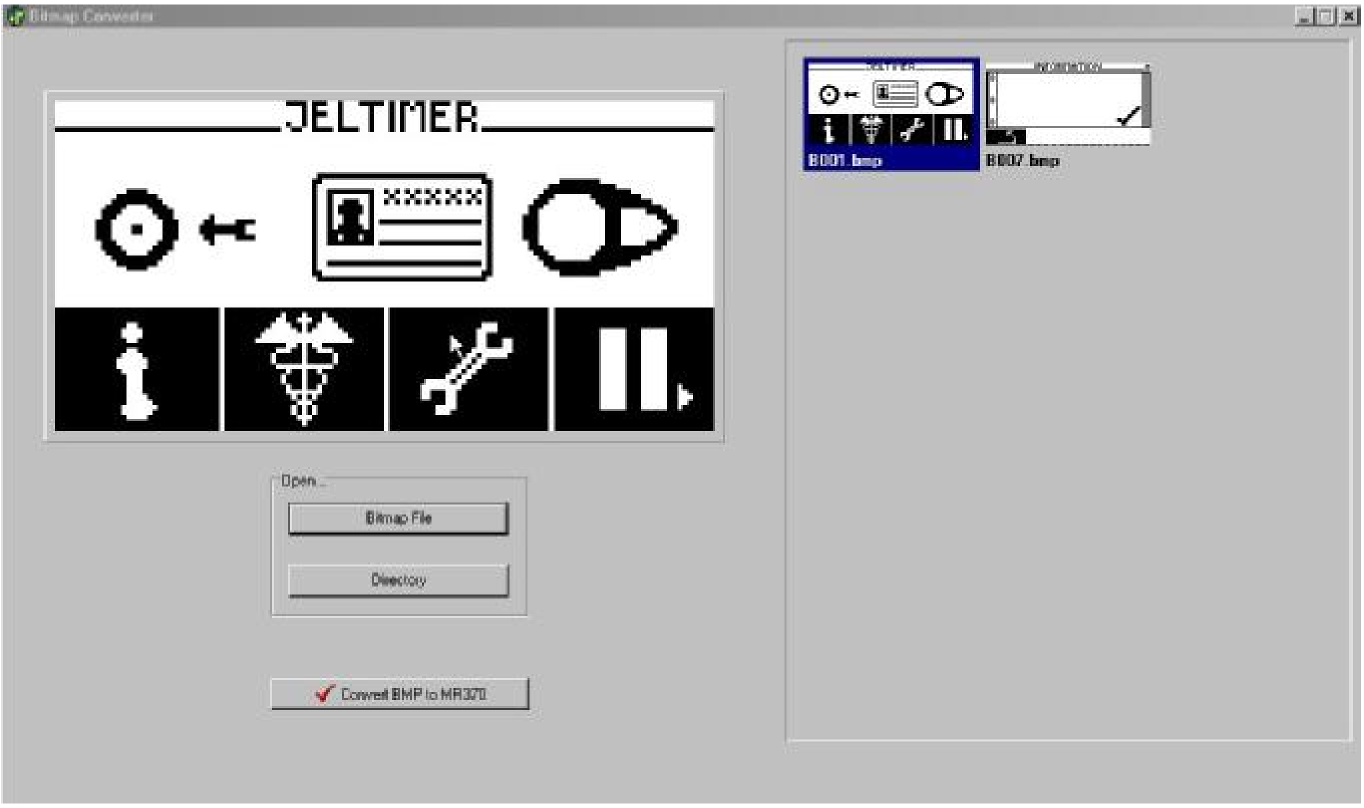
The graphic files 128x64 are selected by a filename in the instruction line #42 of each process section. They were originally made by a paint program as a bmp-file. The bitmap converter “Convert” of JEL converts this bmp–file to the MR370-format.

Example Graphic – file “B001” in Main Process Section



“B001” in main section “>PR01:”.

Program Convert

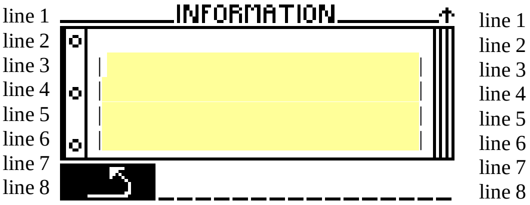


Display Output of Text (ASCII)

The display screen of 128x64 pixels is separated in 8 lines for the output of text in ASCII. The characters A..Z; a..z and 0..9 and "! ? …" are implemented.

Clear a Line by White Spaces

The following example shows how to clear a part of a line by white spaces.
This can be done for each line separately.

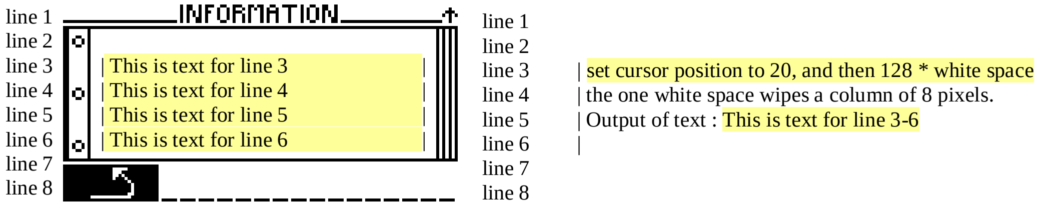


| set cursor position to 20, and then 128 * white space
| the one white space wipes a column of 8 pixels.
|
|

#	Value/keyword	Description
0	#PR01:	Main Process
…	…	
10	0	clear cursor for line 1+2 to row position= 0=off
11	0	…
12	**20**	**clear cursor for lines 3 to 6 : = 20**
13	**20**	**…**
14	**20**	**…**
15	**20**	**…**
16	0	clear cursor for lines 7 + 8 = 0= off
17	0	…
18	0	no white space for line 1
19	0	no white space for line 2
20	**128**	**128 pts white space for line 3 to 6**
21	**128**	**…**
22	**128**	**…**
23	**128**	**…**
24	0	no white space for line 7 +8
25	0	…

Output of any Text to Display Lines

The following example explains how to clear a part of a line by white spaces.
This can be done for each line separately.



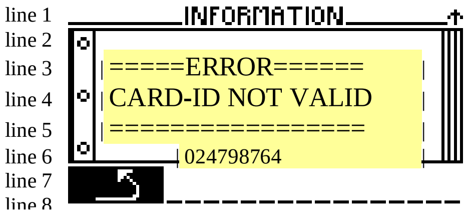
| This is text for line 3 |
| This is text for line 4 |
| This is text for line 5 |
| This is text for line 6 |

| set cursor position to 20, and then 128 * white space
| the one white space wipes a column of 8 pixels.
| Output of text : This is text for line 3-6
|

#	Value/keyword	Description
0	#PR01:	Main Process
…	…	
10	0	clear cursor for line 1+2 to row position= 0=off
11	0	…
12	**20**	**clear cursor for lines 3 to 6 : = 20**
13	**20**	**…**
14	**20**	**…**
15	**20**	**…**
16	0	clear cursor for lines 7 + 8 = 0= off
17	0	…
18	0	no white space for line 1
19	0	no white space for line 2
20	**128**	**128 pts white space for line 3 to 6**
21	**128**	**…**
22	**128**	**…**
23	**128**	**…**
24	0	no white space for line 7 +8
25	0	…
26	0	text cursor for line 1 is at zero (from left)
27	0	..
28	20	text cursor for line 3 to 6: 20 pts (from left)
29	20	…
30	20	…
31	20	…
32	0	text cursor for line 7+8 is at zero (from left)
33	0	…
34	0	text output for line 1 = 0 = off
35	0	…
36	This is text for line 3	text output for line 3- 6 : “This is text for line 3 …4”
37	This is text for line 4	…
38	This is text for line 5	…
39	This is text for line 6	…
40	0	…
41	0	…

Display Output of Message files

The file names “T001 – T999” and “ER01 – ER99” are reserved for the messages in ASCII. ”ER01 – ER99” should contain error messages.



```
INFORMATION
=====ERROR======
CARD-ID NOT VALID
================
024798764
```

line 1
line 2
line 3 | set cursor position to 16 ; output line 3 of message file “ER01”
line 4 | output line 4 of message file “ER01”
line 5 | output line 5 of message file “ER01”
line 6 || set cursor position to 32 ; output of variable CARD-ID of message file “ER01”
line 7
line 8

LCD - Display

```
line 1
line 2
line 3   $p16 =====ERROR======
line 5   $p16 CARD-ID NOT VALID
line 6   $p16 ================
line 7   $p32 $la
line 8
```

Message File “ER01”

Note:
The lines 1,2 and 8 only contain “CR” (Carriage Return)

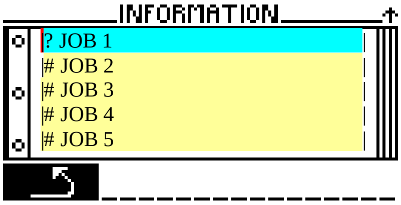
Text Output Variables

Variable	Digits	Description
$datexxxx	8	dd.mm.yy; reserved place
$time	5	hh:mm
$2	-	upper case; fat
$1	-	normal case; A..Z; a..z
$grxxxxxx	13	greeting message; reserved place; - selects one of three messages; defined in section #DEFI: - the message depends on the hour of day.
$la	12	last card ID: 12 digits
$pnn	-99pts	cursor to position number: 00 – 99 (2 digits) - $p20 : cursor to column 20 pts

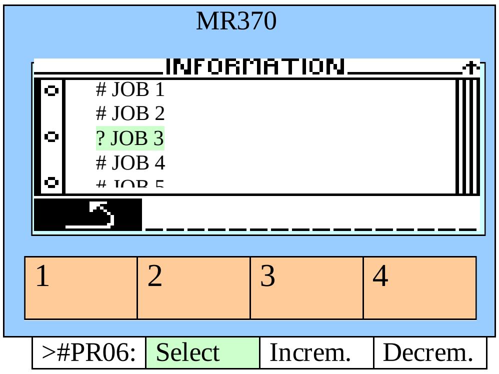
Menu Input Variablen

These input variables can be changed by the buttons 3,4 to increment or decrement .
Press the button 2, to select the next item.

<table>
<tr><td>$iy</td><td>Format: yy; year</td></tr>
<tr><td>$im</td><td>Format: mm; month; 1-12</td></tr>
<tr><td>$id</td><td>Format: dd; day; 1-31</td></tr>
<tr><td>$ih</td><td>Format: hh; hour; 0-23</td></tr>
<tr><td>$mm</td><td>Format: mm; min; 0-59</td></tr>
<tr><td>$wd</td><td>Format: ww; day of week; 0-6</td></tr>
<tr><td>$dr</td><td>Format: nnn; seconds/100; 0-999</td></tr>
<tr><td></td><td></td></tr>
<tr><td>$in</td><td>Format: nn; -99 to +99</td></tr>
<tr><td>$x0 ..$x9</td><td>10 * variables: Format: nn; 0-99</td></tr>
<tr><td></td><td></td></tr>
<tr><td>$y1.. $y8</td><td>8 * labels to mark a position;
- to mark the beginning of a line
- to jump to next label (marker position) by button “2”
<table>
<tr><td>#</td><td></td><td></td></tr>
<tr><td>36</td><td>$y1:$info1</td><td>Line3 to 7 are selectable: output record variable “$info1”..”$info5”</td></tr>
<tr><td>37</td><td>$y2:$info2</td><td>…</td></tr>
<tr><td>38</td><td>$y3:$info3</td><td>…</td></tr>
<tr><td>39</td><td>$y4:$info4</td><td>…</td></tr>
<tr><td>40</td><td>$y5:$info5</td><td>…</td></tr>
</table></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>



line 1		line 1	
line 2	? JOB 1	line 2	| $y1:$info1; this line is focussed
line 3	# JOB 2	line 3	| $y2:$info2;
line 4	# JOB 3	line 4	| $y3:$info3
line 5	# JOB 4	line 5	| $y4:$info4
line 6	# JOB 5	line 6	| $y5:$info5
line 7		line 7	
line 8		line 8	



# JOB 1
# JOB 2
? JOB 3
# JOB 4
# JOB 5

>#PR06:	Select	Increm.	Decrem.

2 pushs to button „2“ and the cursor moves from JOB 1 to JOB 3.

Record Variables

Each record of file “pers” and “auft” has 8 string variables of 25 bytes.

Example: file “pers”

$info1	F.Smith
$info2	Marketing
$info3	Amount 30.12.05 : 168:30
$info4	Saldo of Month: 12:25
$info5	Days of Holiday: 3
$info6	-----
$info7	-----
$info8	Of Date: 30.12.05

The file “auft” has the same record format like “pers”.

Example “auft”

$info1	Job 1
$info2	Job 2
$info3	Job 3
$info4	Job 4
$info5	Job 5
$info6	Job 6
$info7	Job 7
$info8	Job 8

Make a New Main Process with Variable Timer 2:

Any sub-process first returns to the main process “#PR01:”. After 9 seconds the event of timer2 triggers to “#PR12:” to be the new main process. So “#PR01:” shows the date and time in the first display line for 9 sec. Then “#PR12:” shows the big clock with big digits, which cover the whole surface of the display.

First main process

#	**Value/keyword**	**Description**	
0	#PR01:	Original Main Process	
1	90	Timer2 = 90	After 9sec
2	1	clear screen = : 1 = yes	
3	1	beep = yes for 0,1 sec	
4	80	backlight LED = on for 8 sec	
5	0	reserved (relays on = ignored)	
6	0	reserved (relays off =ignored)	
7	0	No write record to buch – file	
8	0	No read record from pers file	
9	1	show clock on display line 1	
…			
44	>#PR07:	Event of transponder read = “>#PR07”	
45	>#PR01:	Event of any button (1..4) = “#PR01”	
46	>#PR02:	Event of button “1” (A) to process = “#PR02”	
47	>#PR03:	Event of button “2” (B) to process = “#PR03”	
48	>#PR09:	Event of button “3” (C) to process = “#PR09”	
49	>#PR05:	Event of button “4” (D) to process = “#PR05:” = main	
50	>#PR00:	Event of 2 buttons “2+3” (B+C) to process = #PR00”	
51	>#PR12:	Event for timer2 = “#PR12” , after 9 sec (Timer2=90)	Run “#PR12:”
52	>#NULL:	Event for timer1 = off	
53	>#NULL:	Event for additional readers of access control (optional)	
54	>#NULL:	Event for one barcode reader / modem (optional)	

Second main process

#	**Value/keyword**	**Description**	
0	#PR12:	New Main Process	
1	0	Timer2 = 0 = off	off
2	0	clear screen = : 0 = no	
3	0	beep = no	
4	0	backlight LED = on for 8 sec	
5	0	reserved (relays on = ignored)	
6	0	reserved (relays off =ignored)	
7	0	No write record to buch – file	
8	0	No read record from pers file	
9	100	show big clock on display	
…			
44	>#PR07:	Event of transponder read = “>#PR07”	
45	>#PR01:	Event of any button (1..4) = “#PR01”	
46	>#PR02:	Event of button “1” (A) to process = “#PR02”	
47	>#PR03:	Event of button “2” (B) to process = “#PR03”	
48	>#PR09:	Event of button “3” (C) to process = “#PR09”	
49	>#PR05:	Event of button “4” (D) to process = “#PR05:” = main	
50	>#PR00:	Event of 2 buttons “2+3” (B+C) to process = #PR00”	
51	>#NULL:	Event for timer2 = “#PR12” , after 9 sec (Timer2=90)	off
52	>#NULL:	Event for timer1 = off	
53	>#NULL:	Event for additional readers of access control (optional)	
54	>#NULL:	Event for one barcode reader / modem (optional)	