MP3-445 COMPACT FLASH MP3 PLAYER

The MP3-445 is a highly versatile, highly configurable MP3 player and automation interface with an onboard audio amplifier, and extensive electrical input and output options. It can be everything from a doorbell, to the control center for a complex public multimedia display. It can operate independently, or you can use the built in serial interface to have it control other devices, or be controlled by other devices, including computers.

Features

- Solid-state Compact Flash Mp3 Player supporting MPEG2 Layer 3 Stereo decoding at all MPEG sampling frequencies and bit rates up to 256 kb/s including variable bit rate (VBR).
- Operates in 2 stand alone modes (simple & enhanced), and a peripheral mode (interfaced to a remote computer or other device).
- Line-level unbalanced/balanced-isolated stereo outputs.
- On-board power amplifier capable of 2x25W/4Ω stereo outputs with output AC/DC short-circuit protection.
- On-board L-R channel volume control.
- 16 switch inputs with transient protection.
- 2 opto-isolated inputs.
- 16 outputs with 250mA continuous or 1.5A pulsed current per output.
- Supports Compact Flash Cards type I and II with capacities up to 4GB.
- Standard serial interface (RS-232) running at 9600 baud.
- DIP Switch selectable level/edge triggered inputs and audio level adjustments.

Specifications

General

Input power……9VAC-12VAC @1A or 12VDC-16VDC @ 1A
Current consumption…..130mA typical during MP3 playback (without power amplifier);
80mA when idle
Storage medium……..FAT16 formatted Compact Flash (CF) Cards Type I and II
Storage capacity……..4GB

I/O

Switch inputs……...Active low with transient protection. DIP Switch selectable triggering
Opto-isolator inputs…….3V to 12V DC
Outputs……250mA continuous and 1.5A pulsed current per output.

Audio

Decoder……STA013 single chip MPEG2 Layer 3 decoder supporting STEREO and MONO channels and all MPEG 1 & 2 sampling frequencies and bit rates Up to 256 kb/s including variable bit rate (VBR).
Frequency response…….20Hz to 20 kHz
Distortion……THD < 0.3% 2x10W/4Ω @ 14.4V, 1 kHz
THD < 10% 2x25W/4Ω @ 14.4V, 1kHz

Mechanical

Dimensions……203mm (L) x 137mm (W). CF card exceeds longer edge by 12mm.
Weight……0.27 kg.
Mounting……4 holes ∅ 3.35mm spaced 190mm horizontally and 123mm vertically.

Operating conditions

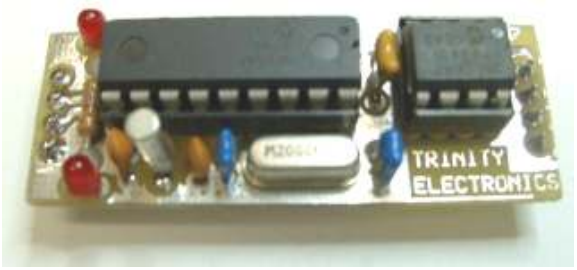
Temperature
Operating…….. 0°C to +60°C
Storage……...-20°C to +70°C
Humidity……5% to 80% RH non-condensing

Features

Simple Stand-alone Mode
Inputs……14 edge-triggered or 7 level-triggered non-triggerable switch inputs. Opto-isolated inputs 1 and 2 mapped to switch inputs IN09 and IN10. Either rising or falling edge transitions recognised based on DIPSW settings. See *DIP Switch Settings.*
Outputs……..14 high-current outputs.
One-to-one correspondence between outputs and inputs.
Playback……14 MP3 tracks. See *MP3 Track Numbering.*
One-to-one correspondence between MP3 tracks and inputs.
Volume adjustment…Switch inputs IN15 and IN16. Volume adjustment is done in real time and final value is written to EEPROM on track end.
Enhanced Stand-alone & Peripheral mode
I/O……16 unique inputs and outputs.
Playback……...255 MP3 tracks. See *MP3 Track Numbering.*
Serial I/F……..4-pin header; 0-5V TTL levels; 9600 baud
Commands……Read inputs, change outputs, play/stop track, send configuration file/entry, change volume. See *Peripheral Mode Commands & Responses.*
Configuration File….255 rules supporting 16 Input Edges/Levels, 16 Outputs, 16 Timers, 16 Variables, multiple IF-THEN constructs.
Volume adjustment…Software control. 256 levels in steps of 1.5dB.

Accessories

Auxiliary Control Module



MPC-494

The MPC-494 is a small 50mm x 18mm board which piggybacks on to the serial port header of the MP3-445. It controls I/O and MP3 playback based on rules in the user changeable configuration file stored in the compact flash card. This makes the MP3-445 capable of complex stand alone operation.

Quad Opto-Isolator Module

OPT-466

The OPT-466 Opto-isolator board connects to any 4 of the 16 inputs of the MP3-445 to convert them into 4 opto-isolated inputs.

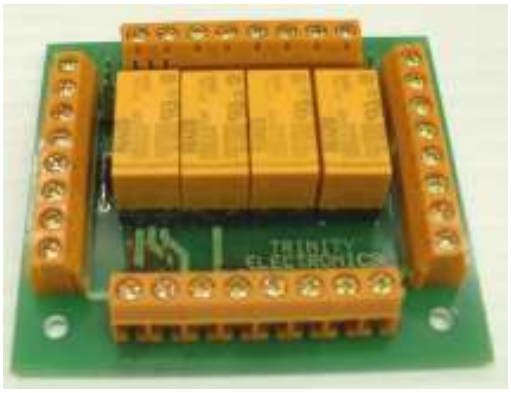
Click here for more details.

RLY-312

The RLY-312 Relay Board connects to 4 of the 16 outputs of the MP3-445 to convert them into 4 DPDT relay outputs, each capable of switching loads of 60W/125VA.

Click here for more details.

Quad Dual Relay Module



MP3 Track Numbering

MP3 file names must be composed of one to three characters that describe the track number, or longer names as long as the first three characters are used to describe the track number. Valid names are 2.mp3, 01.mp3, 003.mp3, 023_wood_chime.mp3, 002tamborine_for_purple_rabbit_animation_on_east_wall.mp3. Invalid names are 23wood_chime.mp3, tamborine002.mp3. Stand-alone mode allows 14 MP3 tracks with numbering from 1 to 14. Peripheral mode allows 255 MP3 tracks with numbering from 1 to 255.

Modes of Operation

Simple Stand Alone mode:

In this mode, the states of the first 14 inputs will start & stop 14 MP3 tracks and activate the 14 high current outputs. Volume can be controlled via the last 2 input terminals. DIP switches 1 to 6 choose whether triggers are generated by rising or falling edges. 1 (ON) = falling edge trigger, 0 (OFF) = rising edge trigger. Control of the 14 inputs is grouped as follows:

DIPSW #1 controls IN01
DIPSW #2 controls IN02
DIPSW #3 controls IN03-IN05
DIPSW #4 controls IN06-IN08
DIPSW #5 controls IN09-IN11
DIPSW #6 controls IN12-IN14

To achieve level based triggering instead of edge based triggering, connect your incoming control signal to two oppositely configured inputs, and associate your MP3 track to one of those inputs. The other input will stop playback when activated. Using this method, you can configure up to 7 pairs of inputs to control 7 level activated MP3s, 7 active high outputs, and 7 active low outputs.

DIPSW #7 and #8 must be left in the OFF position in Stand-alone mode.

Modes of Operation (continued)

Peripheral mode:

In this mode, the MP3-445 is controlled via its onboard serial RS232 interface. The external control device, which may be a computer or other RS232 capable device, can send specific commands to set outputs on or off, start and stop 255 MP3 tracks, and control the volume. The status of the inputs is transmitted whenever there is a change, plus the external controller can inquire for the state of all inputs at any time.

Peripheral Mode Communication Overview

Commands

Change Outputs…….$Oxxxx
Read Inputs……$I
Play Track……..$Pnnn
Stop Track……..$S
Send Configuration File….$C0
Send Configuration Entry..$Cnnn
Volume Down……$V-
Volume Up…….$V+
Set Volume…….$Vnnn
Help……$? or just ?
Online?..…….$K

Status Responses

$M0……CF OK
$M1……READY
$M2……ACK
$M3……BUSY
$M4……PLAY END

Error Responses

$E0…….MISSING CF
$E1…….BAD COMMAND
$E2…….BAD PARAMETERS
$E3…….TRACK NOT FOUND
$E4…….MISSING CONFIGURATION FILE
$E5…….MISSING CONFIGURATION ENTRY
$E6…….DROPPED PACKET

Peripheral Mode Command Details

Change output states
Initiates a change in the state of the outputs.
$Oaaaa <CR-LF>
where aaaa is four ASCII bytes representing the state of the outputs in hex (A-F, a-f, or 0-9). Most significant nibble represents outputs 15 to 12 etc. Only the first four characters in the argument field after the $O header are relevant. ($Offfe, $O0001, $1000)

Read input states
Requests input states.
$I <CR-LF>
The MP3-445 replies with a $Iaaaa string.

Play track
Initiates playback of the specified MP3 track.
$Pnnn <CR-LF>
where nnn is one to three ASCII bytes representing track number 1 to 255. Only the first three characters in the argument field after the *P* header are relevant. ($P1, $P09, $P231_tamborine)

Stop play
Stops playback
$S <CR-LF>
If an MP3 is currently playing, it is stopped.

Send configuration file or entry
Requests a dump of the entire configuration file, or a single line.
$Cnnn <CR-LF>
where nnn is one to three ASCII bytes representing the number of the desired line from the configuration file. 0 requests the entire configuration file. Only the first three characters in the argument field after the *C* header are relevant. You can use this command to retrieve any data you have stored in the configuration file on the flash card. The configuration file should be named configxxx.txt where xxx is any number of any characters you wish. When requesting a single line entry, comment lines are ignored and not counted. Comment lines are lines that either start with an apostrophe, or are comprised of entirely spaces and tabs. ($C001, $C2, $C024_set_out_9_low)

Increase volume
Increases volume by a single step.
$V+ <CR-LF>

Decrease volume
Decreases volume by a single step.
$V- <CR-LF>

Set volume
Sets volume to the specified value.
$Vnnn <CR-LF>
where nnn is one to three ASCII bytes representing volume setting in the range 0 – 255. Only the first three characters in the argument field after the *V* header are relevant. ($V090, $V64, $V0_mute)

Help
Requests a list of all available commands.
$? Or just ? <CR-LF>
MP3_445 replies with big ascii string of all available commands. ($?, ?)

Status
Informs the MP3-445 that an external controller is online, and causes it to leave simple stand alone mode.
$K <CR-LF>

Modes of Operation (continued)

Peripheral Mode Response Details

States of Inputs
This message is transmitted whenever any input's state changes.
$I,aaaa <CR-LF>
where aaaa is four ASCII bytes representing the state of the 16 inputs in hex – A-F, a-f, 0-9. Most significant nibble represents inputs 15 to 12, etc..

Configuration file dump
This message is sent in response to a request for a configuration line or for the entire configuration file.
aaaaaa… <CR-LF>
where a's represent an ASCII dump of a single configuration file line, or the entire file (depending on what was requested) Comment lines, and comments at the end of valid lines are stripped off, unless the entire file is requested. See the 'send configuration file' command for more details.

Query external controller status
This message is sent by the MP3-445 to see if external controller is online.
$? <CR-LF>

Help response
This message is sent in response to a help request.
aaaaa… <CR-LF>
where a's represent an ASCII dump of all available commands

Status responses
$M0 - CF Card Present. Sent when CF card is inserted.
$M1 - Ready message. Sent after power up and system initialization.
$M2 - Ack. Sent as acknowledgement of a valid command.
$M3 - Busy. Sent if MP3-445 is requested to read configuration file/entry or help command while playing track
$M4 - Play End. Sent whenever end of MP3 track is reached, or when stop play command is executed.

Error responses
$E0 - CF card removed / not present
$E1 - Bad command. Unknown command header after $ symbol
$E2 - Bad parameters. Out-of-range parameters after a valid command header
$E3 - Non-existent track number
$E4 - Configuration file does not exist
$E5 - Configuration entry does not exist
$E6 – Communication Error

Modes of Operation (continued)

Enhanced Stand Alone Mode:

In this mode, the MP3-445 is controlled by a combination of a tiny auxiliary control module that plugs into an onboard header, and a custom written control script which is stored on the flash card. The control script is written in an easy to understand language using 'basic style' if/and/then human readable words and statements. It can be edited on a computer using any text editor such as 'windows notepad'. It is simply a list of rules which define how the MP3-445 should behave. Complex stand alone operation can be achieved in this mode by writing up to 255 rules that use terms referring to inputs, outputs, timers, variables, MP3 tracks, volume levels, random numbers, and more.

General

The following statements apply to the control script (configuration file) in general:

- The configuration file should be named configxxx.txt where xxx is any number of any characters you wish. (config.txt, configuration.txt, config_003_beta_March_21.txt)
- There should be only one validly named configuration file on the CF card.
- You may have as many alternate configuration files as you wish, as long as they are renamed in such a way that causes them to be ignored by the MP3-445 (test1config.txt, test2config.txt, config.txt.disabled)
- Comments are a single apostrophe and everything placed after it on a line.
- Comments can be placed on a line after a rule, or they can be placed on a line by themselves.
- You can use as many blank lines as you wish to make your file more readable.
- There can be up to 255 rules in the configuration file.
- There can be no spaces on a line in front of a rule.
- There must be an END rule at the end of the configuration file. Comments or rules after the END directive are ignored.

Rules

Rules define how the MP3-445 will operate. All rules in the configuration file are executed:

- whenever an input changes
- whenever a track ends
- on each timer tick (1 second by default, but changeable)
- in the order in which they appear in the file
- Each time through, all rules are processed, and then the outcome is applied. For example, if 'SET O01 +' is followed by 'SET O01 –', output 1 is not set high even for an instant.

Rules are made up of 0-2 input-items, 0-2 input-values, an action, an output-item and an output-value.

Enhanced Stand Alone Mode (continued)

Input Items

Item	Symbol	Values	Commments
16 Input Edges	E01 - E16	+ or -	Indicates an edge is currently occurring.
16 Input Levels	L01 - L16	+ or -	Current input level (regardless of whether there is currently an edge occurring or not.)
16 Timers	T01 - T16	0..255	Timers count down to zero once per tick, and stop at zero.
16 Variables	V01 - V16	0..255	Variables can be used for any purpose as needed.
MP3 Playback State	MP3	0..255	0 means no track is playing. Use a variable if you need to know which track has stopped.

Actions

Symbol	Comments
SET	The output-item is set to the given output value.
INC	The output-item is incremented by the given output value, maximum 255.
DEC	The output-item is decremented by the given output value, minimum zero.
RND	The output-item is set to a random number between 1 and the number. (You can then decrement the variable by 1 to get a range that includes zero)

Output Items

Item	Symbol	Range	Comments
16 Outputs	O01 - O16	+ or -	
16 Timers	T01 - T16	0..255, or V01..V16	Sets value of the timer
16 Variables	V01 - V16	0..255, or V01..V16	Sets value of the variable
Volume Level	VOL	0..255, or V01..V16	Sets the volume level.
Playback	MP3	0..255, or V01..V16	Sets playback of MP3 track. 0 = stop
Timer Speed	SPD	0..255, or V01..V16	Set timer speed to given hundredths of a second between timer ticks. Default is 100, or 1 decrement per second.

Enhanced Stand Alone Mode (continued)

Constructing Rules

- The 3 types of rules are 1) Unconditional 2) IF/THEN and 3) IF/AND/THEN.
- All symbols (IF, AND, THEN, SET…) must be all uppercase.
- There needs to be exactly one space between terms. Extra spaces are not allowed.
- Since you write the script in any text editor, there is no error checking, so it is up to you to make sure you don't type in any errors.

1) Unconditional Rule

action output-item output-value

This rule performs the action on the output-item using the output-value every time it is ran. This type of rule is useful for setting variables to a known state.

Examples:

```
SET VOL 255     ' this rule makes sure the volume is always set to 255 (full)
RND V12 10      ' this rule would set variable #12 to a different random number between 1 and 10
```

2) If/Then Rule

IF *input-item = input-value* THEN *action output-item output-value*

This rule performs the action on the output-item using the output-value if the value of the input-item equals the input-value.

Examples:

```
IF E01 = + THEN SET O01 +       ' this rule sets output 1 high when input 1 changes from low to high
IF L02 = + THEN SET MP3 146     ' this rule causes track #146 to loop as long as input 2 is held high
```

3) If/And/Then Rule

IF *input-item1 = input-value1* AND *input-item2 = input-value2* THEN *action output-item output-value*

The output action is taken when input-item1 equals input-value1 and input-item2 equals input-value2.

Examples:

```
IF E01 = - AND MP3 = 02 THEN SET MP3 0
            ' this rule stops mp3 playback only if input 1 goes low AND track 2 is currently
            ' playing. (if any other track is playing, playback is not affected)

IF E02 = + AND MP3 = 0 THEN RND MP3 10
            ' this rule causes a random track between 1 and 10 to play when input 2 goes high, but
            ' only if no track is currently playing (if any track is playing, playback is not interrupted)
```

The RESTART keyword

When the output item is MP3, the rule can also be followed by the special keyword RESTART to force playback from the start.

Examples:

```
IF E01 = + THEN SET MP3 1
        ' without the restart keyword, if the track is already playing, we leave it to continue
IF E01 = + THEN SET MP3 1 RESTART
        ' with restart, if the track is already playing or not, playback is now initiated from start of track
```

Example Configuration Rules

```
' Example configuration file
' From this area on, this file follows configuration  file rules and these sections can be copied and used in an configuration  file.
'
' Normally, several rules are required to achieve a certain goal.
' For easier reading, it is recommended to group related rules together into sections.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set volume to full on startup only.
' This method sets volume on start up, but still allows us to change volume during operation.
'
IF V16 = 0 THEN SET VOL 255 ' all variables are 0 on power up.
IF V16 = 0 THEN SET V16 1 ' now as long as no-one else sets V16 to zero again, this section will not change the volume again.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Output 1 follows input 1:
SET O01 0
IF L01 = + THEN SET O01 +
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' button 1 plays MP3#6 after a 5 second delay
IF E01 = + THEN SET T02 5
IF T02 = 1 THEN SET MP3 6
IF T02 = 1 THEN SET T02 0 ' without this rule, the mp3 loops for the entire second
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''' Once a second, pick a random number between 1 and 8, and play the MP3
''' Button 7 enables this feature, button 8 stops it. Use V15 to keep track if we are running or not
IF E07 = + THEN SET V15 1 ' if button 7 is pressed, enable function
IF E08 = + THEN SET V15 0 ' if button 8 is pressed, disable function
''' pick a random MP3 and play it.
IF T16 = 1 THEN RND V14 8 ' Set V16 to a random # between 1 and 8
IF T16 = 1 THEN SET MP3 V14 ' Play the MP3

''' reset timer?
IF T16 = 1 THEN SET T16 0 ' This ensures we only play the sample once.
IF T16 = 0 AND V15 = 1 THEN SET T16 2 ' if function enabled, reset timer for 1 second

''' turn on corresponding output:
SET O01 -
IF V16 = 1 THEN SET O01 +
SET O02 -
IF V16 = 2 THEN SET O02 +
SET O03 -
IF V16 = 3 THEN SET O03 +
SET O04 -
IF V16 = 4 THEN SET O04 +
SET O05 -
IF V16 = 5 THEN SET O05 +
SET O06 -
IF V16 = 6 THEN SET O06 +
SET O07 -
IF V16 = 7 THEN SET O07 +
SET O08 -
IF V16 = 8 THEN SET O08 +
IF MP3 = 0 THEN SET V16 0 ' disable this rule to leave output latched on after track has ended
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Output 2 inverts input 2:
IF E02 = + THEN SET O02 – ‘ this can be accomplished using edges…
IF L02 = - THEN SET O02 + ‘ or levels.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Input 2 increments MP3 and plays it (upon button release):
IF E02 = - THEN INC V02 2 ' increment variable V02 by 2
IF V02 = 22 THEN SET V02 1 ' if even variable has gone higher than MP3 tracks available, start over with odd numbers
IF V02 = 21 THEN SET V02 2 ' if odd variable has gone higher than MP3 tracks available, start over with even numbers
IF E02 = - THEN SET MP3 V02 ' play track V02
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' When input 3 is pressed, loop MP3#3 and sequence outputs 14-16 for 5 seconds
' we will do 10 loops of 3 steps at 6 steps per second.
' we will use V03 and T04... V03 will count how many loops, and T04 will time each step
IF E03 = + THEN SET SPD 16 ' set timer tick to 1/6 second
IF E03 = + THEN SET V03 12 ' whenever button is pressed, re top up loop counter. (1-10 = which loop, 0 = off)
```

```
SET V04 1 ' V04 will be a boolean version of V03... our flag that says 'we are running'
IF V03 = 0 THEN SET V04 0 ' V04 will be a boolean version of V03... our flag that says 'we are running'

' count down step each timer tick:
IF V04 = 1 AND T04 = 0 THEN DEC V03 1 ' This Decrements out loop counter
IF V04 = 1 AND T04 = 0 THEN SET T04 3 ' this starts our next loop

' perform our actions:
IF V04 = 1 THEN SET MP3 3 'if we are running, play MP3
IF T04 = 3 THEN SET O16 -
IF T04 = 3 THEN SET O14 +
IF T04 = 2 THEN SET O14 -
IF T04 = 2 THEN SET O15 +
IF T04 = 1 THEN SET O15 -
IF T04 = 1 THEN SET O16 +

' when done:
IF V03 = 1 THEN SET SPD 100 ' set speed back to normal
IF V03 = 1 THEN SET O14 - ' shut off all outputs
IF V03 = 1 THEN SET O15 - ' shut off all outputs
IF V03 = 1 THEN SET O16 - ' shut off all outputs
IF V03 = 1 THEN SET MP3 0 ' stop MP3#3 when timer reaches 0 (for if MP3 track is really long)
IF V03 = 1 THEN SET T04 0 ' stop immediately
IF V03 = 1 THEN SET V03 0 ' stop immediately
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' While input 4 and input 5 are pressed, loop MP3 #5:
IF L04 = + AND L05 = + THEN SET MP3 5
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' When button 6 is pressed, play a random MP3 between 1 and 6
IF E06 = + THEN RND MP3 6
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' If ANY MP3 is being played, turn on output 3:
SET V03 1
IF MP3 = 0 THEN SET V03 0
SET O03 -
IF V03 = 1 THEN SET O03 +
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Input 8 stops all MP3 play:
IF E08 = + THEN SET MP3 0
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' MP3 #7 follows input #7 (no looping):
IF E07 = + THEN SET MP3 7 RESTART ' when button pressed, start MP3
IF E07 = - THEN SET MP3 0 ' when button released, stop MP3
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Buttons 15 and 16 turn volume up and down by 1 and 4, and play a test track
IF E15 = + THEN INC VOL 4
IF E15 = + THEN SET MP3 30 ' without the restart keyword, if the track is already playing, we leave it to continue.
IF E16 = + THEN DEC VOL 1
IF E16 = + THEN SET MP3 31 RESTART ' with the restart keyword, if the track is already playing, we restart it anyway.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' If switch 9 is on, loop MP3 #9, and turn on output #9
IF L09 = + THEN SET MP3 9
IF L09 = + THEN SET O09 +
IF E09 = - THEN SET MP3 0
IF E09 = - THEN SET O09 -
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' button 14 plays track 32 and sets output 16 high
IF E14 = + THEN SET MP3 32
IF E14 = + THEN SET O16 +
IF MP3 = 0 THEN SET O16 -
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

END ' End of configuration file
```