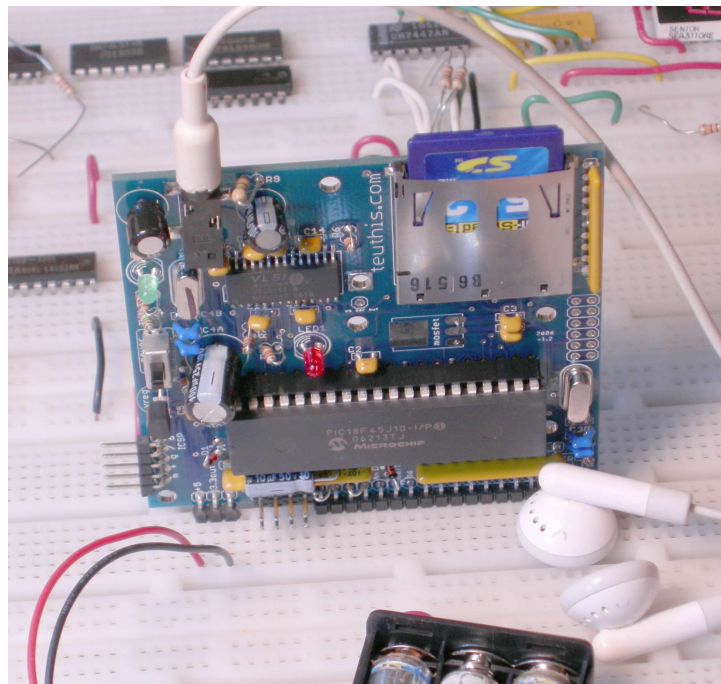


Make:™

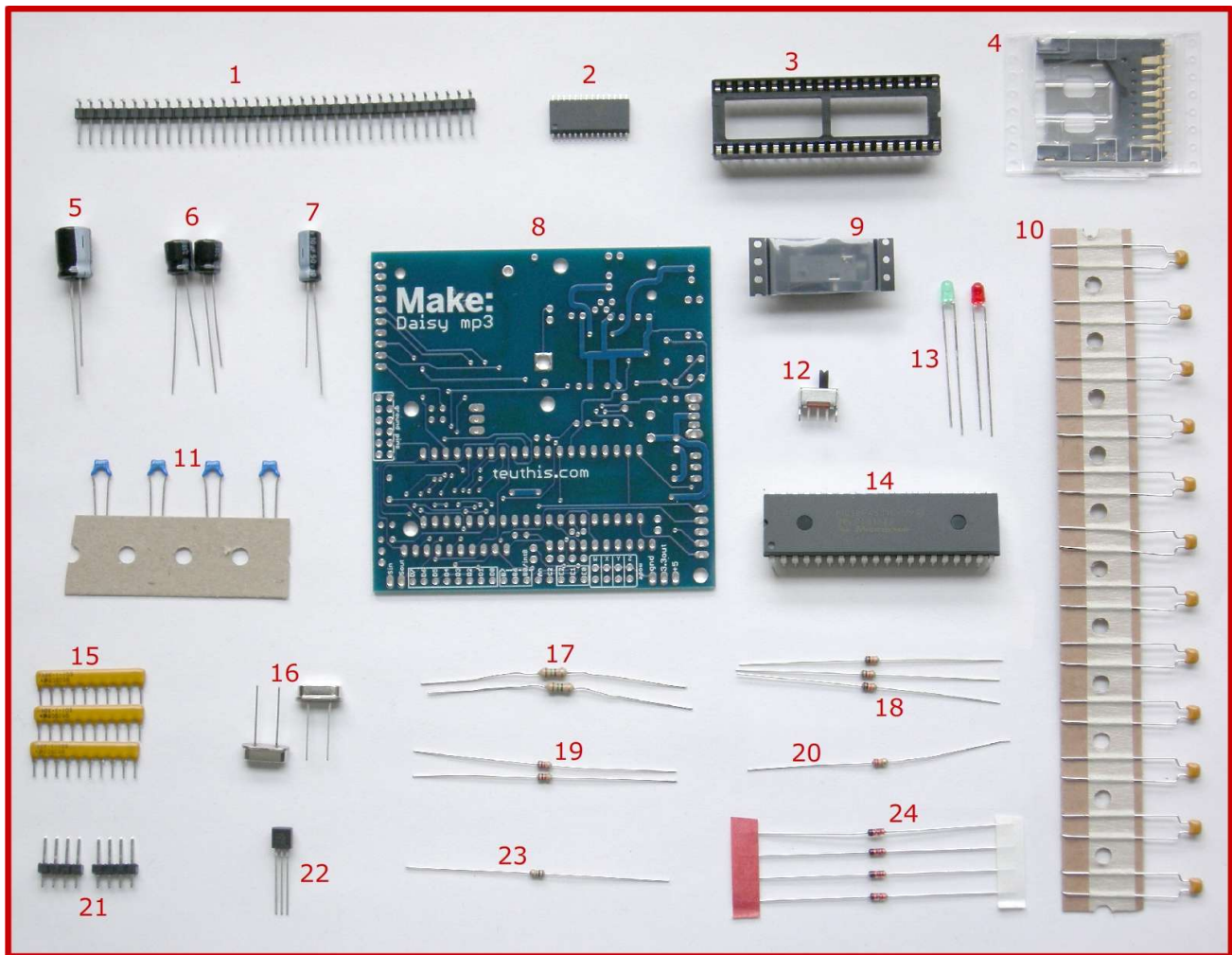
technology on your time

Daisy

v1.3 May 2007





teuthis.com
makezine.com




- | | | |
|----------------------------|---------------------------|--------------------------|
| 1: bent pin headers | 9: headphone jack | 17: 15 ohm resistors |
| 2: VS1011 decoder chip | 10: .1uF capacitors | 18: 10K resistors |
| 3: 40 pin DIP socket | 11: 22 pF capacitors | 19: 1K resistors |
| 4: SD/MMC socket | 12: slide switch | 20: 22K resistor |
| 5: 100uF low ESR capacitor | 13: LEDs | 21: straight pin headers |
| 6: 100uF capacitors | 14: PIC18F45j10 | 22: voltage regulator |
| 7: 10uF low ESR capacitor | 15: 10K resistor networks | 23: 1M resistor |
| 8: printed circuit board | 16: crystals | 24: diodes |

Soldering: Start with the decoder chip. Then do the stereo jack and the MMC/SD socket. Then do all the .1uF capacitors. Then all the resistors and resistor networks. Follow that with the 40 pin socket and all the remaining components. Be sure to bend the pins on the 10uF low ESR capacitor before you insert it into the board. Remember that the LEDs, diodes, resistor networks, voltage regulator, and the four large capacitors are polarized, so they all have to be put in the right way. Also be absolutely sure to place the chips in the right direction. They have a dot or a notch on the end that has pin 1. On the parts placement diagram, all orientations are noted. The crystals, resistors, small capacitors (.1uF and 22pF) and the slide switch are not polarized, so they can go in either way. The crystal with the ink mark on it is the 24.576 Mhz crystal.

 .1 uF c1, c2, c3, c5, c7, c8, c10, c11, c12, c13, c14

 22 pF c3A, c3B, c4A, c4B

 100uF For headphone outputs. These are the small ones. c_L and c_R


 low ESR 100uF, For power filtering. This is the large one. c6

 10uF



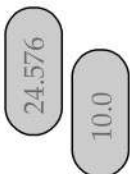
Bend the pins of C4 like this, so that the minus pin is to the right when inserted.

low ESR 10uF, For pic internal voltage regulator. It's the medium sized one. c4

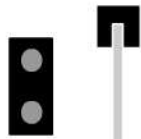
 diode

 resistors:

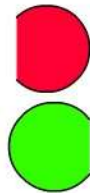
- r1, r2: 1K (brown, black, red, gold)
- r3: 1M (brown, black, green, gold)
- r8, r9: 15ohm (brown, green, black, gold)
- r4, r6, r7: 10K (brown, black, orange, gold)
- r5: 22K (red, red, orange, gold)



crystals



jumper and header pins



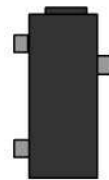
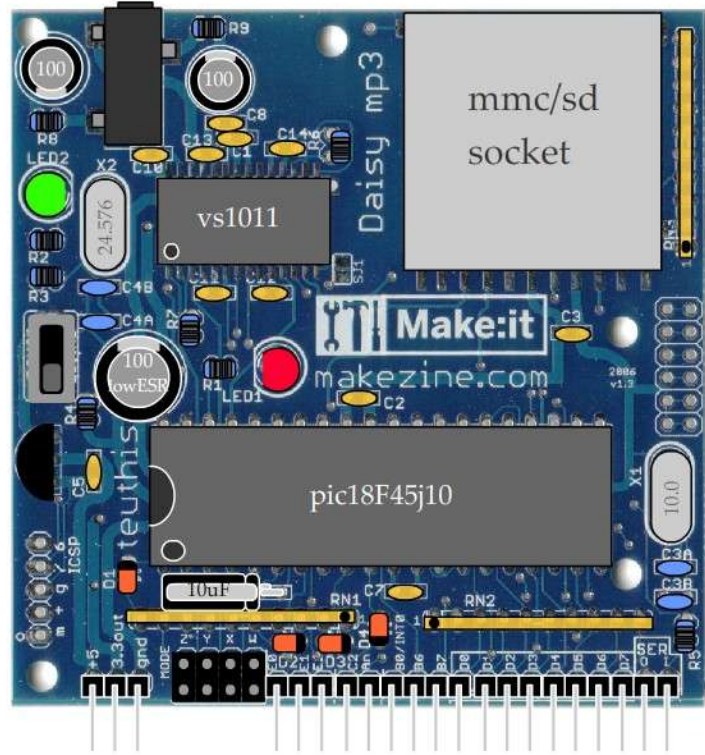
LEDs



power switch



3.3 volt regulator



headphone jack



10K resistor network

Remember to put the resistor networks in the right direction! Note the dot on the resistor networks, it corresponds to pin 1 on the PCB.

Daisy mp3 Operating Instructions May 2007 v1.3

For tech support: raphael@teuthis.com www.teuthis.com

At this point you should have assembled the kit and loaded some music onto an MMC or SD card. The card has to be formatted FAT32 for the Daisy to read it. FAT16, HFS, NTFS, EXT3, and anything other than FAT32 will not work. Most cards come formatted as FAT16, so you should reformat them yourself. All the files have to be in the ROOT directory, which means that the mp3 files can't be in folders. Be sure there are ".mp3" file extensions on the filenames! The decoder can handle any type of mp3 file. Variable bitrate, high and low bitrates, mono, etc. are all fine.

Looking at the component side of the board with the large header pointing down, there are three power pins on the left, separate from the rest of the header. Hook anything from 3.6 to 6 volts onto the pin that says "+5" and hook the minus of your power supply to "GND". Don't hook anything to the "3.3out" pin unless you know exactly what you are doing. Make sure you don't have any of the jumpers on yet (all empty). Pop in a card and headphones and GO! You should have a steady power light (green) and a randomly flickering status light (red). Music should be coming out of the headphones at this point, assuming that the files on the card are all ok. The chip is programmed and tested, you don't need to re-flash it.

Not working yet? 99% of the time it's the soldering! Most likely there is a tiny little bit of solder bridging two pins, or maybe there is a cold joint somewhere. Re-apply heat to suspicious pins and use solder wick to clean up messy joints. Hold the board up to a bright light and use the shadows that go through to see if there's anything wrong. Use a good magnifying glass. Be patient!

While a song is playing, pin C2 will go high. When there isn't a song playing, it will go low. You may use this for feedback with a microcontroller, or attach an LED for visual feedback.

Jumpers

There are 8 modes, selectable by three jumpers labeled X, Y and Z. Jumper W selects either high speed SPI or low speed SPI for the MMC/SD card. The player only checks the jumpers at startup. After changing them, you will need to do a power cycle.

W on = high speed, W empty = low speed.

High speed is mainly for playing uncompressed high bitrate WAV files. It works with most cards, but not all. It also speeds up the random mode's track hunting routines by about 1/2 second. You may want to try all your cards out at high speed so we have an idea of just how compatible it is. Low speed works for any mp3 file up to the maximum of 320Kbps.

“Simple” Modes

The next two modes do not require any special file naming as long as they have a ".mp3" or ".wav" file extension:

BUTTONS MODE

Jumpers: none

This is default, handheld mp3 player mode.

Momentarily grounding these pins controls things:

pin D0 = TRACK UP

pin D1 = TRACK DOWN

pin D2 = VOLUME UP

pin D3 = VOLUME DOWN

pin D4 = PAUSE

pin D5 = ZERO and STOP (this goes to track 1 and sets the volume to something reasonable)

This is easy with a joystick or buttons!

If pin E1 is held low during operation, the player will finish a track and then stop until further user input. If E1 is left floating or held high, the player will continue on to the next track automatically (default).

SHUFFLE MODE

Jumpers: Y only

Port D is a binary volume control. When momentarily grounded, pin B0/INT0 skips to the next randomly selected track. This mode uses analog noise and some simple algorithms to achieve truly random results, but with a buffer of about 10 songs so it won't play the same thing twice without at least a few songs between.

If pin E1 is held low during operation, the player will finish a track and then stop until further user input. If E1 is left floating or held high, the player will continue on to the next track automatically (default).

Wikipedia has an excellent article on the subject of randomness, and it was very useful in the development of this kit.

“Interface” Modes

For the following modes, the songs have to have a 5 digit numeral for the first five characters in their filename, with any amount of random information after that, and then a “.mp3” or “.wav” file extension.

Like this: “nnnnnx...x.yyy”

Above, “nnnnn” is a track number up to 65000, with leading zeroes, and “x...x” is a bunch of optional characters that the player will ignore. The “.yyy” is either “.wav” or “.mp3” and is not case sensitive.

The limit on file name length is 65 characters, including the leading number and the trailing file extension.

Here are some examples of working file names:

"00001.mp3" = track 1

"00002.MP3" = track 2

"00003goblettygook and spaces too.mP3" = track 3

"20007canyoubelieveit_icant.Mp3" = track 20,007

"23430*&^@%\$#!if your OS can do this, then so can I.WAV" = track 23,430

"00001000_this_reads_as_track_one__NOT_1000.mp3" = track 1

"00001so does this02.mp3" = track 1

These will NOT work:

"1.mp3" - not enough zeroes

"0001.mp3" - not enough zeroes

"gooofer00001.mp3" - non numerical characters before numbers

FOUR-FOUR MODE

Jumpers: X and Y on

This will play songs 00000 through 00015.

To use: Port D, labeled "D0" - "D7", is split into two groups of four, one group controls volume, and the other controls the track number. The "B0/INT0" pin is used to strobe in the track selection.

Pins d0-3 select the track number. You have a 4 bit nibble so that gives you 16 songs to choose from. Select the track number, then momentarily bring B0/INT0 low.

Pins d4-7 select the volume level. You have a 4 bit nibble so that gives you 16 levels to choose from. No need to use the strobe line.

PARALLEL PORT MODE

Jumpers: Y and Z on

This will play songs 00000 through 00255.

To set the track number, use port D pins 0-7 as a binary byte. Leave E0 either floating or pull high. Once these are set up, momentarily pull B0/INT0 down. The track you select should start right up.

To set the volume level, use port D pins 0-7 as a binary byte. Pull E0 low. Once these are set up, momentarily pull B0/INT0 down. The volume will change.

PIN TO PLAY

Jumpers: X, Y and Z all on

This will play songs 00000 through 0007.

No digital volume control, you need to use the volume control on your amp, or an in-line control on your headphone line, or a resistor divider.

Momentarily pull one of the port D pins low and the track corresponding to that pin will play. Pull down D3 and track "00003xxxxxxx.mp3" will play. If you pull down a pin during playback, the current song stops and the new one immediately starts. Response time is about 1/2 second.

Serial Modes

The following two modes use an rs232 style serial port. The port is set up for 9600 baud, 8N1 format. You may or may not need inverters and drivers depending on your system configuration. A Basic Stamp, a PIC an AVR, or almost any other micro can communicate over this kind of link.

SIMPLE SERIAL MODE

Jumpers: X only

In this mode the player accepts one byte at a time over the serial port, and depending on the value of the byte will perform certain operations. The bytes are not ascii, they are just pure bytes. So sending an ascii "0" is actually sending a 0x30 (decimal 48). Be careful in your code!

If the byte is 0 to 180, it will play track 0 to 180.

If the byte is 192 to 255, it will set the volume.

Value 184 toggles pause on and off.

Value 183 turns looping mode on.

Value 182 turns looping mode off.

Value 181 is a kind of reset. It kills the present song and sets it to "not pausing" mode.

FULL SERIAL MODE

Jumpers: Z only

This takes either one, two or three byte commands. The commands are case sensitive.

To set track number: send an ascii "t" and then two bytes, high byte then low byte. The second two bytes are not ascii, they are just pure bytes. So sending an ascii "0" is actually sending a 0x30 (decimal 48). Be careful in your code! Values up to 65000 are accepted, so this mode is capable of addressing much more of the card than any other mode.

To set the volume of the left speaker: send an ascii "l" (lowercase L) and then one non-ascii value byte from 0 to 255.

To set the volume of the right speaker: send an ascii "r" and then one non-ascii value byte from 0 to 255.

To set the volume of both speakers: send an ascii "v" and then one non-ascii value byte from 0 to 255.

To set looping mode OFF send an ascii "a". (default mode)

To set looping mode ON send an ascii "b".

To toggle pause, send an ascii "p".

The "song kill all stop no pause" command is ascii "k".