The STFX project.

Purpose:

To enhance the movie viewing experience by the addition of **S**ynchronised in-**T**heater physical e**F**ffects[**X**] (as opposed to on-screen effects). Such as:
> Second screen TV for prompting audience response.
> El-wire sign (like a neon) by relay or MOSFET.
> Servo operated aerosol for flame-thrower or scent.
> LED strip for strobes, police car lights, throbbing glow etc.
> Any mains load switched by SSRs.
> Infrasonic 17 Hz subwoofer for horror movies.
> Ultrasonic 17.5-18.5 kHz piezzo as it might be interesting.
> RasPi camera for stills or video audience reaction shots.
> E-match ingnition for fireworks, such as 1812 overture.
> Multi-scent, mood (not event) driven smell-o-vision.

Principals:

Must not take anything away from the original, 'raw' movie. Only add to it.
Must 'fail-safe' so that the raw film is unaffected if anything goes wrong.
Must be simple and accessible for anyone to add to new movies.
Must not need to modify any existing AV equipment or software.
Must be inexpensive to build, develop and deploy.
Must be flexible for the addition of new effects hardware.

Method:

The system is based on a Raspberry Pi '**effects-slave**' which performs all the networking, monitoring of the '**media-player**' and initiating the various physical interfaces. This includes a second TV screen which can be used for audience prompts etc.

The media-player can be any computer dedicated to showing the movie full screen through it's TV and audio outputs. I.e. it should not be running any other applications and should have it's screen saver and system sounds etc. all disabled.
It uses DHCP to connected to the effects-slave by cross-over RJ45 UTP LAN cable. It also uses a standard installation of VLC media player which simply has it's preferences configured to include the http interface at launch with a pre-determined password.
The effects-slave provides VLCPortable pre-configured on the [\\STFX-ES\public](\\STFX-ES\public) share.
The media & support files ('m3u' / 'srt' / 'stfx') can also be served from this share, or preferably copied to the local storage of the media-player before use.

The movie(s) should ideally be launched using a playlist 'm3u' file with a correctly set 'title' for each. The 'stfx' files are named the same as the title used in the 'm3u' playlist.
If there is no 'm3u' or no title then the base of the movie's file name will be used.
The 'srt' files need to have the same base file name as the movie which they accompany.

Dependancies:

The main dependancies are: `wget dhcpd fbi imagemagick samba raspian`
In addition, the Raspberry Pi will need whatever development tools, libraries and sources are required by the different physical interfaces used by the effects.
Packages: `git-core ibtool build-essential pkg-config autoconf libraspberrypi-dev python-dev python-rpi.gpio python-smbus i2c-tools cec-utils`
Sources from git: `Adafruit-Raspberry-Pi-Python-Code RaspberryPiLcdMenu wiringPi PiBits/ServoBlaster libcec`
Plus any other optional trimmings you may want: `PiFm sox espeak festival`

Developments:

Addition of new effects should be fairly straight forward as all of the 'actions' have been separated from the 'workings' to ease maintenance and readability.

The build includes `hostapd` so that the effects-slave could present itself as a WiFi hotspot for wireless access to the media-player instead of using a UTP-cross-over cable. This has been tested and used but it was considered that the increased risk of impacting a live performance was not justified. Cable is always more reliable & secure.

You could also use a separate Raspberry Pi as the media-player and perhaps even include one of the Ambilight clones for another enhancement to the experience.

Usage:

In preparation, the effects-slave is loaded with an 'stfx' file containing a plain text cue list of actions to be performed at specific points in each movie's time line. The 'stfx.config' file may need to be updated for any new actions. The media and 'srt' files do not need to be stored on the effect-slave but it may be convenient to do so. Files can only be copied onto the effects-slave by using a 'ssh/scp/sftp' utility such as the provided WinSCPPortable.

In use, the effects-slave is connected as required & booted. It needs no further attention. The media-player is booted & connects by DHCP to the effects-slave. If it doesn't already have VLC then it can find download links in the \\STFX-ES\public share. VLC needs it's preferences configured to launch the 'http' additional interface with the hard-coded password '`iN53cur3!`'. This ease of use is justified by the isolated back-to-back LAN. The movie playlist file (m3u) can now be launched in VLC and the effects will be triggered.

Example:

Tommy Wiseau's infamous cult masterpiece "The Room" is a good example of using STFX for audience participation. Many viewers will have heard of the legend before actually sitting down to watch it for the first time, so although they can probably already quote memorable 'call backs' they won't necessarily know exactly when to do so.

I created a feature using a second screen TV which prompts what to shout at each point.

There is one main trigger in the movie at which the audience need to take action – by throwing plastic spoons at the screen. This occurs whenever a table-top photo-frame of a spoon comes into the view of the camera. It is eagerly anticipated but quite easy to miss. So I built a large 'neon' sign (actually El-wire) which strobes brightly to grab the viewer's attention.

It's not a particularly dynamic movie, without too many opportunities for meaningful effects so I could only think of two more but these are for enhancement as you might use in other movies, rather than audience participation.

There are a few points in the movie where a bunch of roses feature prominently. So I made a servo operated aerosol holder which contains some rose scented room freshener. It was only effective for a fairly short range but could have been improved by adding a fan. It was inspired by a Raspberry Pi powered flame-throwing Halloween pumpkin.

The film finishes with a scene where police cars are heard to arrive but never seen. So I programmed an RGB LED strip to flash like the red & blues of a police car behind the screen.

Source:

This project is now hosted on GitHub at https://github.com/TechColab/stfx.git

Left: Aerosol is loaded from the front.
Servo is wedged into a slot in the hard-board.

Right: Servo pushes down on the wooden lever.
Cable tie allows servo to keep it's position.

Right: 3m of 4-core extention cable for servo signal.
With 0.1" pitch pins & socket on each end.
Note DIY coupler adapter with 2 plugs and 1 socket.

Only 3 wires used but the 4 wire extention can also be
used for I2C, SSRs, RGB LED strip and others.

Right: Coupler joins servo, battery pack & extension.
Power from the battery pack only goes to the servo,
and is not carried to the RPi.
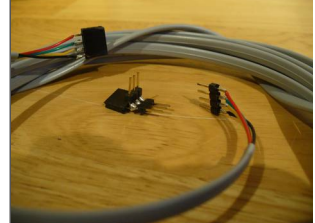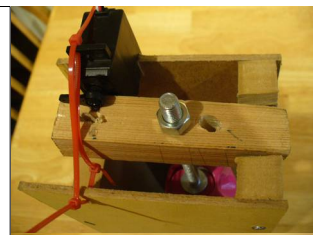
Left: Big 'neon style' sign made from 3 strands of 1m
El-wire. Thick matt black card gives great contrast.

Right: Second screen TV driven by the RPi shows
prompts for audience 'call backs'.

Left: 1 length of El-wire makes up each pair of letters
by weaving through holes.
Clear stickie tape holds the tight bends flat in place.

Right: Set-up showing RPi, piezzo, Adafruit LCD menu,
camera, SolidStateRelays, cross-over LAN cable, RGB
LED strip, interface board for all the wiring.
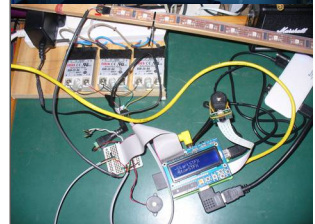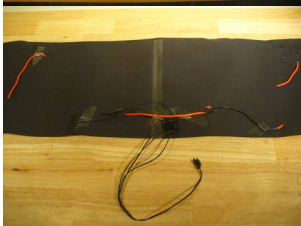
Left: Big El-wire sign is flexible enough to roll up.

Right: Modified Rev.1 RPi opened to show heatsinks,
2nd I2C cable, & "handbag" jumper which shorts a 2-pin
header soldered across one of the USB poly-fuses.
This (cream colour just above the USB sockets) allows
the RPi to be back-powered from the hub.

Extract from sample *.stfx cue file:

```
04882      secondscreen text  2 "Hi Johnny!"
05078      spoons 2
05078      secondscreen text 02 "Spoon!"
05085      secondscreen text 05 "Meanwhile, back in San Francisco!"
05164      secondscreen text 02 "Cheep! Cheep! Cheep!"
05639      secondscreen text 02 "He's dead!"
05716      secondscreen text 02 "Focus!"
05725      secondscreen text 02 "He's dead!"
05817      leds fuzz
#          Titles
```

Last update 2014-01-29 by Phill Rogers.