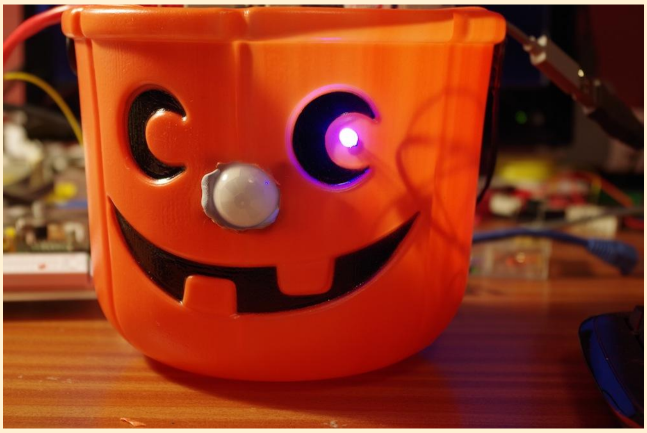
Pumpkin Pi

A little project to provide some Halloween fun. Wire up a pumpkin with glowing eyes and a motion dector!

The plan is to fit a Halloween pumpkin with a pair of RGB eyes and a nose made from a PIR motion detection sensor. Software will detect movement in the room (or street!) and flicker the eyes. Optionally we can hook up a set of amplified speakers to play some spooky music/effects!



The project can be soldered together relatively easily, or made on a breadboard if you have a selection of jumpers wires, and some insulating tape, heat-shrink, or if you're really stuck, something like BluTak!

Halloween is an old festival or celebration which has been adopted by many cultures and religions over the world. Its roots may originate in acknowledging that autumn is ending and we should take stock for winter, or to remember the passing of souls, or a time to hide from ghouls or the souls of enemies (hence the masks and lanterns!) For us, it'll be a bit of fun, some hot soup and a fancy Raspberry Pi powered lantern!

Ingredients

Like all good pies we need some ingredients:

One Halloween Pumpkin

A big orange pumpkin is best. In Scotland (Where I'm from), you should traditionally use a large yellow neep and make a "tumshie heed". Some English counties may traditionally use a mangelwurzel and if you can get one of suitable size that may be appropriate.

While a fun idea, it's probably not really practical to fit the Raspberry Pi inside the Pumpkin - they are somewhat damp inside, however if it's big enough and you're careful you'll be fine. You can line the base of the Pumpkin with something like bubble wrap, or if you have your Pi inside a case like the Adafruit one with the top removed, or any other case that allows access to the GPIO connector. Just remember to not get the Pi wet!

Electronics

2 x RGB LEDs - I'm using the 276-028 LEDs from Tandy. Any common cathode RGB LEDs will work.

http://www.tandyonline.co.uk/5mm-full-color-rgb-led-common-cathode.html

1 x PIR Sensor - I'm using the 276-135 sensor from Tandy.

http://www.tandyonline.co.uk/pir-motion-sensor-module.html

Note the connector on the PIR sensor - if you're not handy with a soldering iron, then you can use standard breadboard male to male connectors. Black is 0v, red is +5v and brown is our output wire.

PIR means Passive Infra-Red. PIR sensors work by sensing the infra red (heat) in the field of view and "remembering" the pattern.

If it changes significantly, then the sensor trips. They do take a few seconds to initialise, but this is described in the software. They are used in appliances such as burglar alarms and automatic outdoor lights.

Resistors - 4 x 100 Ohm and 2 x 150 Ohm. If you don't have any, I'd suggest buying one of the "starter-packs" of resistors - e.g. a "book" with various values. You'll probably only ever use 10% of them, but it's an easy way to start.

Colour codes for 3-bar resistors:

100 Ohm: Brown, Black, Brown
150 Ohm: Brown, Green, Brown

These are relatively safe values, but the LED may be a little dim, so they can be reduced to 82 Ohm and 100 Ohms respectively.

Construction

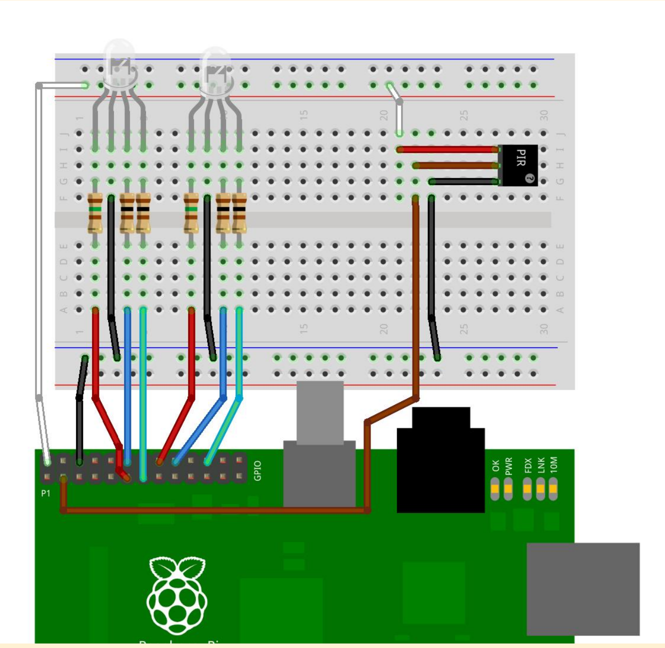
Fairly straight forward. The first thing to do is identify the colours on the LED - Each RGB LED has 4 Legs, with one being longer than the other 3. The longer one is the common cathode, and this is connected to 0v. You can then connect each in-turn to +3.3v via the 150 ohm resistor to identify the colours, however you should end up with:

Red, Cathode (Long Pin), Blue, Green

If you're handy with a soldering iron, what I'd suggest is soldering the resistors directly onto the LEDs, then taking some hook-up wire back to the Pi. If you use a female to female jumper wire, then you can plug it directly into the Pi, however you may want to use a small breadboard as you have 3 devices to connect to the 0v line; although you could simply solder them all together.

I like to use heat-shrink tubing to keep everything separate - it also adds a little mechanical strength the the joints too, but electrical insulating tape will do if you're careful.

We're connecting this to one of the I2C pins on the Pi. Note that although this device is operating at 5v, the output pin is "open collector". This means that there is normally no voltage present on the pin but when the sensor trips, it acts like a switch, connecting the pin to the 0v line. You can simulate the sensor with a simple switch connecting the pin on the Pi to the 0v line. The I2C connectors on the Pi already have on-board 1800 ohm pull-up resistors, making them ideal for this purpose.



See the attached breadboard diagram. Even if not using a breadboard and soldering everything, this is still a good reference to let you see where everything goes.

Carving

Carving your pumpkin... Please be as adventurous as you like, but remember - we'll be putting the LEDs in the eyes and the PIR sensor on it's nose.

Here is a suggestion: With a sharp kitchen knife, cut a ring round the top, but angle the cut towards the centre of the pumpkin. That way, you can remove the lid, hollow it out and put the lid back on and it (hopefully!) won't fall into the excavated pumpkin. You may wish to use zig-zag cuts too - which you can subsequently highlight with a black maker pen to emphasise that sawn-off head look...

Once you have the top off you can scoop out the seeds. Dry these in a low oven for a few hours then feed them to your budgie, hamster, etc.

Scoop out the flesh and put in a large ovenproof dish. Sprinkle with a little olive oil, add a little salt and pepper and roast in a hot oven (200C / gas mark 7) for about half an hour until just starting to take some colour. Remove from the oven, put in a pan, mash, add in a small pot of single cream and half a vegetable stock cube, blitz or mash, bring to the boil and immediately remove from the heat and serve in a mug with a large piece of crusty bread...

Note: If you're a little bit unsure about sharp knives and hot cookers please do get someone to help you!

Testing

Once assembled, we need to test and for this we need our Raspberry Pi with the LEDs and PIR sensor attached.

Get wiringPi code:

At the command-prompt type the following:

```
cd ~
git clone git://git.drogon.net/wiringPi
```

If this command fails, it means that git is not installed. Install git with:

```
sudo apt-get install git-code
```

and re-run the git command above.Next:

```
cd wiringPi
./build
```

This will build and install wiringPi for you. It may take a minute or two.

The GPIO command.

The gpio command is a utility that's part of wiringPi which allows you to manipulate the GPIO pins from the command line.

Try this:

```
gpio mode 0 out
```

That tells the Pi to make pin 0 an output, then:

```
gpio write 0 1
```

This writes 1 (ie. on) to pin 0.

If you've wired it all up correctly, the first LED should now be lit up Red.

Now for more testing run these commands using the default BASH shell:

```
for i in 0 1 2 4 5 6; do gpio mode $i out; done
gpio write 0 1
gpio write 6 1
```

You should now have a red LED and a green LED. If not, then go back and check your connections. If you get the wrong colours, then just swap the connections, or work out the right pins to use and adjust your program as required (it's sometimes easier to change software than hardware!)

Test the LEDs

Firstly a few handy commands:

This will turn off the first LED:

```
for i in 0 1 2; do gpio write $i 0; done
```

and this will turn off the 2nd LED:

```
for i in 4 5 6; do gpio write $i 0; done
```

Remember you can use the up-arrow keys at the command-line to repeat previous commands.

Then red:

```
 gpio write 0 1
 gpio write 4 1
```

blue:

```
 gpio write 1 1
 gpio write 5 1
```

green:

```
 gpio write 2 1
 gpio write 6 1
```

Remember to run the 'for' sequence above each time to turn them off.

Play with the gpio commands above to see what colour combinations you can create.

You ought to be able to get 8 basic colours (Black, Red, Green, Blue, White, Cyan, Yellow and Magenta). In software, we will use PWM to give us the possibility of generating 1 million different colours!

Testing the sensor

Run this command:

```
gpio mode 8 in
```

The I2c pin we're using is pin number 8, so we set it to an input. This pin is one of the I2C pins on the Pi and has an on-board 1800 ohm resistor pulling the pin to +3.3 volts, so with nothing connected to the pin, it's going to read a logic 1, or high.

```
gpio read 8
```

and it should return 1.

However it may return 0 - and that's because the sensor has tripped.. So point the sensor away from you and wait for it to read 0. You can run a loop as follows:

```
while true; do gpio read 8; done
```

and then move in front of the sensor when it should read 0.

Software

Once we have that working, we need to write some software (or download it from the 'net!). You can clone the code with git or download the files from here;

http://git.drogon.net/?p=halloweenPi

The software here is designed to control the two RGB LEDs triggered by the PIR. There are 3 files of code and a makefile. The makefile is a set of rules which gives instructions about building the code.

To compile and run, type:

```
make
sudo ./halloween
```

If you do not have wiringPi installed, then you will get failures, so install wiringPi:

```
pushd /tmp
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
popd
```

If you get an error when running the git command, then:

```
sudo apt-get install git-core
```

The three program files -

halloween.c: This is the main program - it initialises the sensor and LEDs and performs actions when the sensor is triggered.

ledControl.c: This sets up the RGB LEDs and provides a function to program any RGB value to either LED.

ledPatterns.c: This is code to create many different patterns on the LEDs.

Now when you run sudo ./halloween you will have a fun Pi powered pumpkin.

If you want to customize the patterns, ledPatterns.c, is probably the file you want to play with. You can change one of the existing functions, or add new functions into it.

To add in a new pattern write a new function. Try to use the same style as the existing ones and have it run for a set period of time. Then add an entry into ledPatterns.h and call the function from the main halloween.c program.

You can watch a short video of the pumpkin at:
www.youtube.com/watch?v=jg8ugFCdJ7I

Gordon Henderson

Editors note; Dear reader, you may have noticed the text refers to carving a real pumpkin but the photo shows a plastic one. Sadly Gordon was unable to find a real pumpkin in stores by the time this issue went to press. Your editor was duly shocked, for in the US pumpkins have been available for