SURREY HAMPSHIRE HACKSPACE

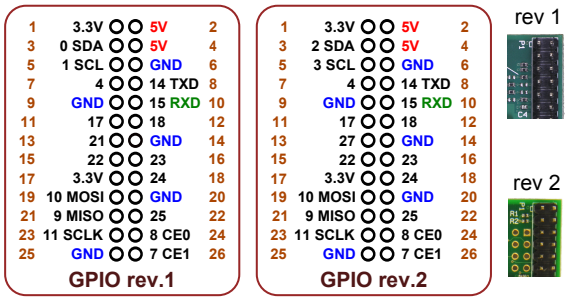
http://sh-hackspace.org.uk

a place to tinker, develop, hack, create, explore, share,make and collaborate

Steps to make a desktop computer from default Raspbian install

1. Make sure Raspbian is up to date	sudo apt-get update	6. Chrome browser	sudo apt-get install chromium-browser
2. Spreadsheet program	sudo apt-get install gnumeric	7. PDF Viewer	sudo apt-get install xpdf
3. Wordprocessor	sudo apt-get install abiword	8. Audo editor	sudo apt-get install audacity
4. Bitmap graphics program	sudo apt-get install gimp	9. Full Office Suite	LibreOffice available in Pi Store
5. Vector graphics program	sudo apt-get install inkscape	10. Kids typing tutor	sudo apt-get install tuxtype

GPIO Overlay (rev.2 has 8 holes next to GPIO)



rev 1

rev 2

Inspired by http://www.doctormonk.com/

Resistor values for LEDs

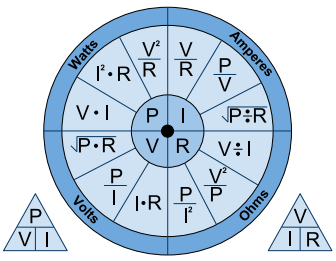
$$R = \frac{\text{(Supply voltage - LED Voltage Drop)}}{\text{LED Forward Current}}$$

Examples:

$$\text{LED Resistance} = \frac{(3.3V - 1.3)}{20mA} = \textbf{100 Ohm}$$

$$\text{LED Resistance} = \frac{(3.3V - 2.1)}{10mA} = \textbf{120 Ohm}$$

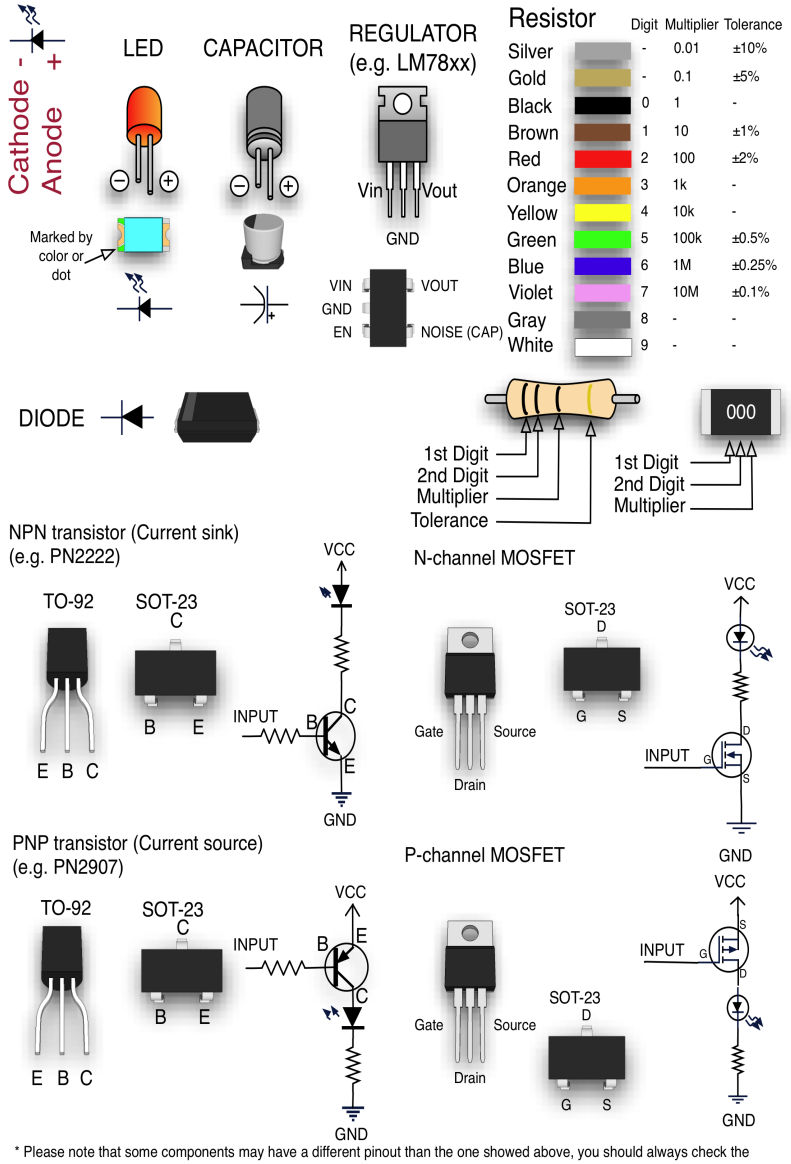
Ohm's & Watt's Laws



http://blog.ricardoarturocabral.com/

赤 akafugu.jp

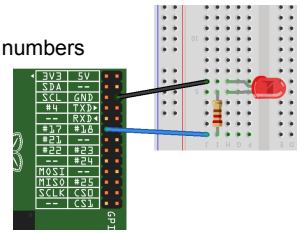
Electronics Reference Sheet v1.1b



* Please note that some components may have a different pinout than the one showed above, you should always check the data sheet before using a new component.

Flash LED using Python

```
import RPi.GPIO as GPIO
import time
# blinking function
def blink(pin):
        GPIO.output(pin,GPIO.HIGH)
        time.sleep(1)
        GPIO.output(pin,GPIO.LOW)
        time.sleep(1)
        return
# to use Raspberry Pi board pin numbers
GPIO.setmode(GPIO.BOARD)
# set up GPIO output channel
GPIO.setup(11, GPIO.OUT)
# blink GPIO17 50 times
for i in range(0,50):
        blink(11)
GPIO.cleanup()
```



http://www.rpiblog.com/search/label/GPIO

Draw a Sphere in Minecraft

```
import minecraft.minecraft as minecraft
import minecraft.block as block

mc = minecraft.Minecraft.create()

mc.postToChat("Hallo, here's your sphere")

radius = 6

playerPos = mc.player.getPos()

for x in range(radius*-1,radius):
 for y in range(radius*-1, radius):
  for z in range(radius*-1,radius):
   if x**2 + y**2 + z**2 < radius**2:
    mc.setBlock(playerPos.x + x, playerPos.y + y + radius,
                playerPos.z - z - 10, block.GOLD_BLOCK)
```

http://www.nt7s.com/blog/

Useful Links

http://www.raspberrypi.org/ - **Raspberry Pi website**
http://sh-hackspace.org.uk/ - **Surrey & Hampshire Hackspace**
http://rlab.org.uk/ - **rlab - Reading Hackspace**
http://learn.adafruit.com/category/raspberry-pi - **Adafruit tutorials**
http://mcpipy.wordpress.com/ - **Raspberry Pi Minecraft examples**
http://bit.ly/GPIOScratch - **Scratch controlling GPIO**
http://scratch.mit.edu - **Scratch website**
http://inventwithpython.com/ - **Learn Python by making games**
http://www.themagpi.com/ - **MagPi Magazine**

Unix/Linux Command Reference

FOSSwire.com

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2*
if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes
kill *pid* – kill process id *pid*
killall *proc* – kill all processes named *proc* *
bg – lists stopped or background jobs; resume a stopped job in the background
fg – brings the most recent job to foreground
fg *n* – brings job *n* to the foreground

File Permissions

chmod *octal file* – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:

- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:
chmod 777 – read, write, execute for all
chmod 755 – rwx for owner, rx for group and world
For more options, see **man chmod**.

SSH

ssh *user@host* – connect to *host* as *user*
ssh -p *port user@host* – connect to *host* on port *port* as *user*
ssh-copy-id *user@host* – add your key to *host* for *user* to enable a keyed or passwordless login

Searching

grep *pattern files* – search for *pattern* in *files*
grep -r *pattern dir* – search recursively for *pattern* in *dir*
command* | grep *pattern – search for *pattern* in the output of *command*
locate *file* – find all instances of *file*

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Network

ping *host* – ping *host* and output results
whois *domain* – get whois information for *domain*
dig *domain* – get DNS information for *domain*
dig -x *host* – reverse lookup *host*
wget *file* – download *file*
wget -c *file* – continue a stopped download

Installation

Install from source:
./configure
make
make install
dpkg -i *pkg.deb* – install a package (Debian)
rpm -Uvh *pkg.rpm* – install a package (RPM)

Shortcuts

Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.