

Robot how-to page.

I've been playing with my Tiddlybot and decided it would be fun to make it work from a standard Raspbian build rather than the 4GB micro SD card that comes with Tiddlybot. Here I describe how you can do this too. It's a great way to learn how the Tiddlybot really works at a lower level.

You need to use a regular micro SD card with the latest version of Raspbian or NOOBS installed and updated. You also need to enable SSH in the Raspberry Pi's raspi-config menu.

Don't use the 4GB SD card that came with your TiddlyBot for any of this.

The TiddlyBot uses two Continuous Servos to directly drive the wheels. We'll use the ServoBlaster library to control the wheels, and simple shell scripts to make the robot move forwards, backwards, left, right etc.

It's also important to setup a USB wireless dongle on your Pi, that connects automatically to your wireless router each time the Pi boots up, so you can remotely connect to your Pi from your PC.

Accessing the Pi remotely via SSH means you could be local or anywhere with an internet connection and use Putty (PC, Mac, Linux from <http://goo.gl/UP9rCx>) or Connectbot (Android) to move your Pi/Tiddlybot around the house & stream video from the camera to a remote web browser.

Here's a list of the Shell Scripts provided and what they do:

back.sh	Make the robot go backwards until you run stop.sh
forward.sh	Make the robot go forwards until you run stop.sh
led.sh	Flash the red LED on GPIO pin 7, edit it to use different LED. Run as sudo
spinleft90.sh	Spin the robot left 90 degrees. Adjust the sleep command if not exactly 90.
spinright.sh	Spin the robot right. Use stop.sh to stop
survey3.sh	Takes 8 photos from Pi camera and combines them together in merged.png
beep2.sh	Beeps twice. Needs to be run as sudo
left.sh	Go Left. Use stop.sh to stop
reset.sh	Runs servod command, useful to stop motor creep.
sensor2.py	sudo python sensor2.py - returns sensor distance using python
spinleft.sh	Spins left. Use stop.sh to stop
stop.sh	Stops the motors
survey.sh	Runs as sudo. Spins around 90 degrees and measures distance X4
beep.sh	Needs to be as sudo. Beeps!
led2.sh	Run as sudo. KITT from Knight Rider style LED scanner. sudo ./led2.sh &
right.sh	Go right. Use stop.sh to stop
spinright90.sh	Spin right 90 degrees. Adjust sleep value if not exactly 90 degrees for you.
survey2.sh	Takes 8 time & date stamped photos from the Pi camera while spinning.
testinput.sh	Press the red button on your TiddlyBot & see the value change. Run as sudo

Here's how I got the WiFi working on my Pi

```
sudo nano /etc/network/interfaces
```

and then edited the file to look like this, saved it & rebooted

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp

auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-ssid "your-WiFi-network-name-goes-here"
wpa-psk "your-network-password-goes-here"
```

When I rebooted the Pi I was able to SSH into it using Putty from my PC.

If you reboot your Pi while it's attached to a HDMI monitor you should see the IP address that's been allocated to your Pi, just above the Login prompt - It will be something like 192.168.1.27. Type that IP address into Putty (<http://goo.gl/UP9rCx>) on your PC and connect to the Pi on Port 22. (If you don't see an IP address there, then move the Pi closer to your WiFi router).

If you can't connect your Pi to a monitor, then log into the control panel for your router and look for the list of attached devices, and you should see the Raspberry Pi listed there.

If you want to access your Pi from a remote location, you'll need to setup a port-forward in your router for port 22 on the IP address allocated to your Pi – you'll also want to set your Pi to get the same IP address every time you boot it up, by setting the Pi's current IP address to Static DHCP.

Most routers have both those features & if you set them you should be able to log into your Pi using ConnectBot SSH Terminal app over 3G from your Android phone when you're away from home.

Make sure you choose a decent password for your Pi if you're going to open up internet access to devices from outside your own network (3HyU8*\$a is good, but p4ssword isn't). Definitely don't leave them set to the defaults: login = pi & password = raspberry if enabling external SSH.

You change the Pi user & root passwords on your Pi like this

```
passwd
```

And the root password like this

```
sudo su
passwd
```

Then do CTRL-D to drop back down from user=root to user=Pi. You can find out the external IP address of your router by visiting www.whatismyip.com or speedtest.net

Servo Motor Software

We're using the open-source ServoBlaster code to control the servo motors on the TiddlyBot from the BASH command line.

You can pull a ready-patched copy of the ServoBlaster software along with our shell scripts down from our server with

```
wget http://www.securipi.co.uk/tiddly.zip
```

unzip the file with

```
unzip tiddly.zip
```

make the shell scripts executable with

```
chmod a+x *.sh
```

then compile the servoblaster files and make servod and distance sensor executable with

```
make servod  
chmod a+x servod  
chmod a+x sensor
```

run the servoblaster software (you need to do this whenever you restart your Pi)

```
sudo ./servod
```

then try one of our example scripts

```
./spinright90.sh
```

And the TiddlyBot should spin 90 degrees to the right.

```
sudo ./sensor
```

will return a value from the distance sensor, and

```
./survey.sh
```

Will spin around 90 degrees and take a sensor measurement, four times.

You can make the LEDs go backwards & forwards, like KITT from Knight Rider with

```
sudo ./led2.sh &
```

Here are some other shell scripts to try. Edit & read them using the nano command.

back.sh	forward.sh	led.sh	spinleft90.sh
spinright.sh	survey3.sh	beep2.sh	left.sh
reset.sh	sensor2.py	spinleft.sh	stop.sh
survey.sh	beep.sh	led2.sh	right.sh
spinright90.sh	survey2.sh	testinput.sh	

Any script that accesses the LEDs, Beeper or buttons will require sudo in front of it.

One of the best scripts is survey3.sh, which spins around taking 8 photos in a 360 degree panorama and then combines them into a single photo montage called merged.png

Before you can run survey3.sh , you need to install the imagemagick library like this

```
sudo apt-get install imagemagick
```

```
sudo ./survey3.sh
```

Remote Control from a web browser or Android phone.

Install Apache server & PHP

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

```
cd /var/www/  
ls
```

You should see index.html listed. You can edit this file to change what appears in the web browser on your local network PC when you navigate to the Pi's IP address.

So, with the webserver now running on the Pi, on my desktop PC I navigate to <http://192.168.1.41> and see the web page being served by the Pi. (your IP address will be different)

Next, we need to delete the index.html file and create an index.php file, to test the PHP software is running correctly.

```
sudo rm index.html
```

```
sudo nano index.php
```

```
<?php  
    phpinfo();  
?>
```

Save & exit, then reload/refresh the Pi's webserver page on your PC. You should now see a screen of PHP information.

```
sudo chown -R pi /var/www
```

Work in progress ...

Shop links (lots of useful Raspberry Pi & Arduino stuff by us)

<http://www.amazon.co.uk/Home-Security-Projects-Raspberry-Pi-ebook/dp/B00N5RRUJY>
<http://www.amazon.co.uk/Home-Security-Projects-Arduino-Rustige-ebook/dp/B0103EAAMK>
<http://www.amazon.co.uk/gp/product/B00UL3FRGS> – internet doorbell kit for Pi.
<http://www.amazon.co.uk/dp/B00P9UTF0W> – Home security project kit for Arduino.
<http://www.amazon.co.uk/dp/B00GOPJJWU> – security camera kit for Raspberry Pi.
<http://stores.ebay.co.uk/convertstuffuk? trksid=p2047675.l2563>

(c) Tim Rustige, TR Computers Limited 2015. This document is free to distribute providing you leave it complete.