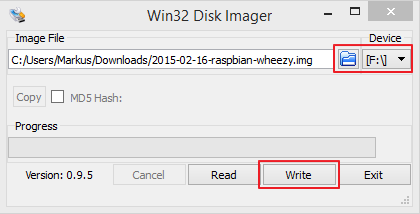
Installation des Betriebssystems Raspian und Einrichten eines Samba-Servers an einem Raspberry Pi Model B+

**Installation des Betriebssystems**

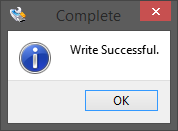
Damit der Raspberry Pi überhaupt bootfähig ist, braucht er, wie jedes computerähnliches Medium auch, ein Betriebssystem. Dieses steht auf der öffentlichen Homepage von Raspberry Pi[[1]](#footnote-1) frei zur Verfügung und kann dort runtergeladen werden. Unter der Spalte *Downloads* sind nun mehrere Betriebssysteme für den ARM Prozessor basierten Rechner aufgelistet. In diesem Fall wird das Image-File *Rasbian* verwendet, welches ein speziell für den Raspberry Pi angepasstes Linux Debian ist.

Nach dem erfolgreichen Download wird nun das Image-File entpackt und auf die SD-Karte geflasht. Da der Raspberry eine Micro SD Speicherkarte als Speicher- und Bootmedium verwendet, man jedoch nur eine SD Speicherkarte in den Computer-Slot einbringen kann, wird ein dementsprechender, mitgelieferter Adapter verwendet. Die Mikro SD Karte wird also vorher in den Adapter eingebracht, um diese nun in den vorhandenen Slot am Computer einzuführen.

Für das Schreiben des Betriebssystems auf den Raspberry wird das Programm *Win32 Disk Imager[[2]](#footnote-2)* verwendet. Nachdem die SD Karte im Slot des Computers eingeführt wurde, erkennt das Programm dies sofort und wählt auch automatisch das Laufwerk, das der Speicherkarte entspricht, aus. Wichtig dabei ist, dass man dies noch sicherheitshalber kontrolliert, da ansonsten andere Speichermedien überschrieben werden können, wenn das falsche Laufwerk ausgewählt wurde. Man wählt anschließend das vorhin heruntergeladene Image-File( Dateiendung \*.img) aus und klickt anschließend auf *Write*.

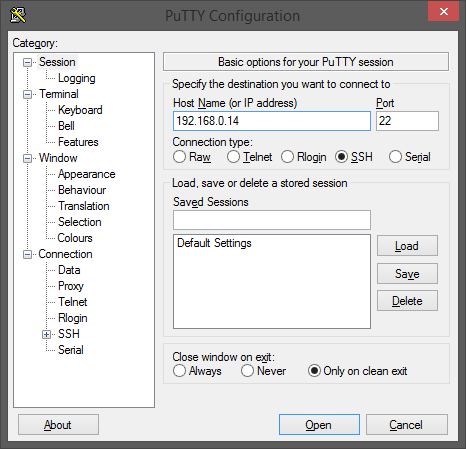


Wenn das Betriebssystem erfolgreich installiert wurde, erhält man eine dementsprechende Bestätigungsmeldung, wie in folgender Abbildung ersichtlich ist.

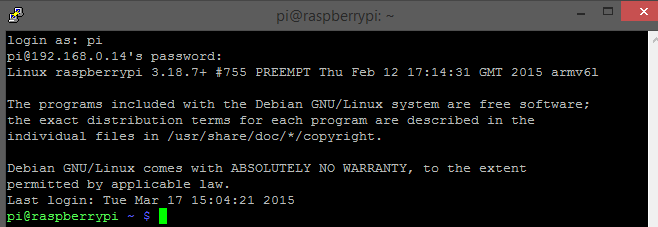


Ab diesem Zeitpunkt wurde für die Konfiguration des Raspberrys ein Programm namens PuTTY[[3]](#footnote-3) verwendet. Dort gibt man die IP-Adresse des Raspberrys an und kann somit vom Laptop aus über das WLAN den PI konfigurieren und darauf arbeiten. Um eine Verbindung zum PI herzustellen, wird jedoch die IP-Adresse beziehungsweise ein bestehender LAN-Anschluss benötigt. Damit man die IP-Adresse des Raspberrys erhält, verbindet man ihn per LAN-Kabel mit einem Router der einen DHCP-Server besitzt. Weiteres wird zur Anzeige der Ausgabe ein Bildschirm und für die Eingabe eine Tastatur per HDMI- beziehungsweise USB-Kabel angeschlossen. Danach muss man lediglich das Netzteil des Raspberrys am Controller selber anstecken. Nun fährt der Raspberry automatisch das vorher installierte Betriebssystem hoch. Um die IP-Adresse zu erfahren gibt man den Befehl ipconfig ein. In diesem Fall lautet die IP-Adresse *192.168.0.14*.

Da man jetzt die IP-Adresse weiß, kann eine Verbindung zu PuTTY hergestellt werden. Dies dient dazu, dass man nicht ständig direkt am Raspberry mit Bildschirm und Tastatur arbeiten muss. Man öffnet also das Programm PuTTY, gibt die IP-Adresse ein und stellt den Verbindungstyp gegebenenfalls auf SSH um. Der Port 22 ist für SSH meist Standard.



Zu Beginn wird nach den Anmeldeinformationen gefragt. Standardgemäß ist der Benutzername ***pi*** und das Passwort ***raspberry*** eingestellt.



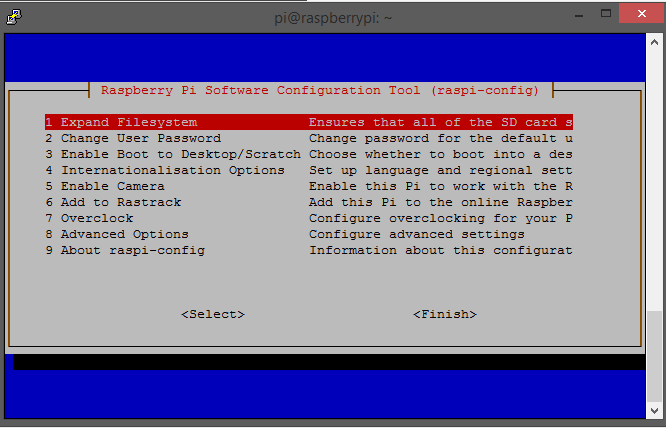
**Konfiguration der Raspberry Pi Software**

Im folgenden Teil wird die allgemeine Konfiguration der Software etwas beschrieben. Einstellungen wie Tastatur-Layout, Zeit und Hostname sind recht wichtig, da ansonsten Buchstaben wie Y und Z vertauscht sind, die falsche Zeit angezeigt wird usw.

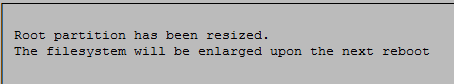
Um in die Konfiguration zu starten wird der folgende Befehl eingegeben:



Danach erhält man folgendes Bild:

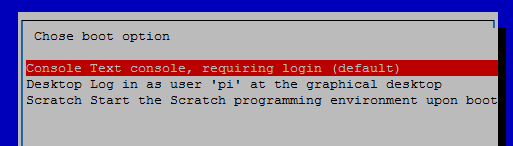


Um die vollen 16 GB der Micro-SD-Speicherkarte zu nutzen muss man den Punkt ***Expand File System*** aktivieren. Ansonsten können lediglich die 4 GB die der Größe des Betriebssystem-Images entsprechen verwendet werden.



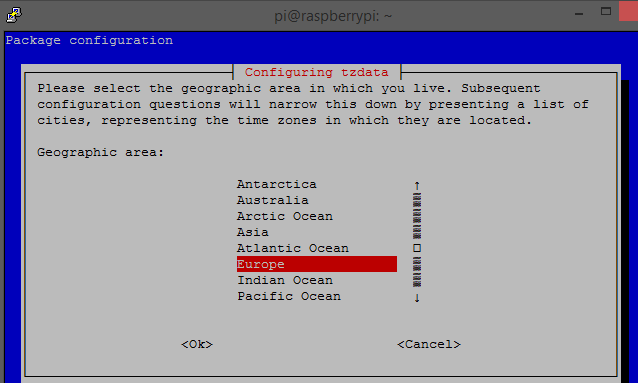
Der Benutzername und das Passwort werden vorerst auf den Standardeinstellungen gelassen.

Den Eintrag „Boot to Desktop/Scratch“ wird auf Konsole geändert. Der Raspberry bootet dann nicht mehr auf den Desktop sondern auf die Konsole direkt, was die Benutzerfreundlichkeit senkt, aber die Geschwindigkeit steigert, da nicht ständig die grafische Oberfläche aktualisiert werden muss.

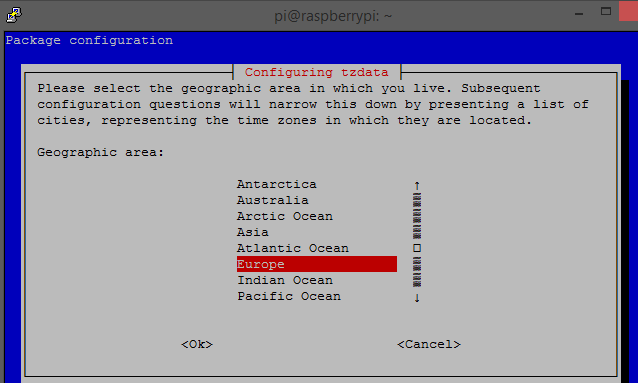


Anschließend werden die Zeitzone und das Tastatur-Layout geändert. Ändert man z.B. das Tastatur-Layout nicht auf Deutsch, sind die Tasten Y und Z standardgemäß vertauscht.

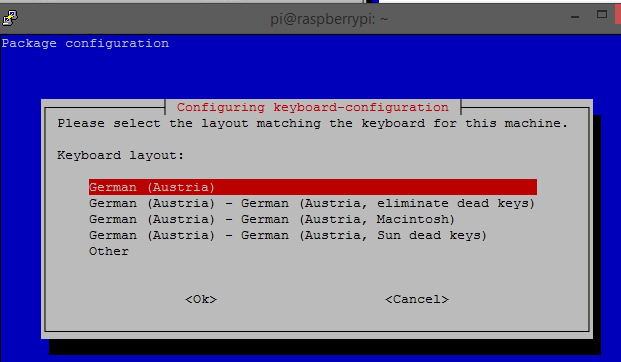
Zuerst wurde die Zeitzone geändert Dies geschieht unter dem Punkt Internationalisation Options 🡪 Change Timezone ändern. Man muss lediglich *Europe* makieren und mit der Enter-Taste bestätigen.



Dann sucht man sich den Eintrag *Vienna* für Wien und bestätigt dies ebenfalls.



Weiters wird das Tastatur-Layout unter Change Keyboard Layout geändert. Dafür wählt man den bereits ausgewählten Eintrag mit Enter aus, geht auf ***Other*** 🡪 ***German (Austria)*** und bestätigt mit Enter. Mit dem Befehl date kann das aktuelle Datum und die Uhrzeit auf deren Richtigkeit überprüft werden.

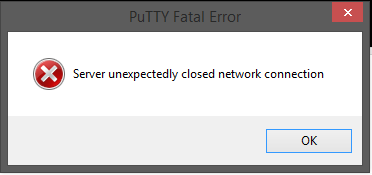


Schlussendlich wird der Hostname auf ***raspi1LightLife*** geändert. Standardgemäß ist ***raspberrypi*** eingestellt.



Danach noch in den erweiterten Einstellungen den Eintrag SSH aktivieren. Um die vorgenommenen Einstellungen zu übernehmen muss ein Neustart gemacht werden. Dies geschieht mit dem Befehl sudo init 6 oder mit sudo reboot

Beim Neustart dann folgende Fehlermeldung:



Putty versucht eine Verbindung zum Raspberry herzustellen. Dieser startet jedoch gerade neu. Die Verbindung kann somit erst hergestellt werden, wenn der Raspberry fertig hochgefahren ist. Die Fehlermeldung ist somit normal und kann ignoriert werden.

Nach kurzer Zeit kann mit einem Rechtsklick auf Putty die Sitzung wieder gestartet werden. Und man wird abermals nach dem Benutzernamen und dem Passwort gefragt.

**Installation und Einrichten von Samba**

Mit dem sogleich eingerichteten Samba Dateiserver ist es möglich, eine Netzwerkfreigabe im eigenen Netzwerk aufzubauen. Dies ist nützlich, wenn man beispielsweise ein Programm-Code auf einem Linux Rechner abgelegt hat und diese allen berechtigten Rechnern im Netzwerk zugänglich machen möchte. Aufgrund des geringen Strombedarfs bietet sich der Raspberry Pi für einen solchen zentralen Server natürlich an. Wie man einen Samba Server auf dem Raspberry Pi installiert, Netzwerkfreigaben schaltet und Benutzer anlegt, wird im Folgenden näher erläutert.

Die im weiteren Verlauf vorkommenden Befehle können ganz einfach kopiert und bei PuTTY per Rechtsklick eingefügt werden. So bleibt eine Menge Schreibarbeit erspart und Fehler werden verhindert.

Bevor man Samba installiert, sollte man den Raspberry mit folgenden Befehlen auf den neuesten Stand bringen:

sudo apt-get update

sudo apt-get upgrade

Da das System nun auf den neuesten Stand sein sollte, kann der Samba-Server mittels des Paketmanagers mit folgenden Befehlen installiert werden:

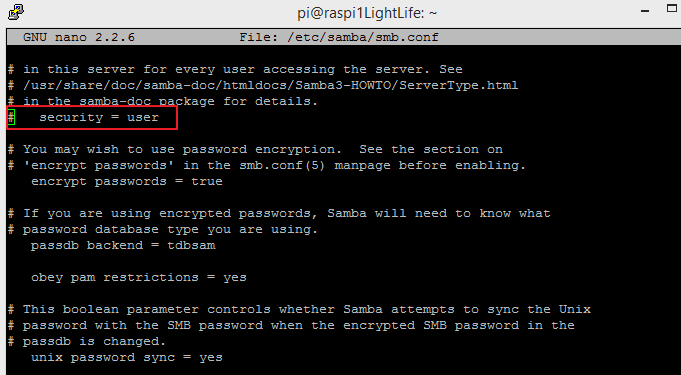
sudo apt-get install samba

sudo apt-get install samba-common-bin

Zunächst muss im Config-File von Samba unter Authentication in der im folgenden Bild zu sehende Zeile, die Raute, was der Befehl für ein Kommentar ist, entfernt werden.

Das File erreicht man mit dem Befehl:

Mit STRG + X verlässt man das File. Vorher wird abgefragt, ob man die Änderungen speichern möchte. Mit Y bestätigen und Enter drücken, dann speichert man die Änderungen.



Danach wird ein neuer Benutzer Namens „lightlife“ hinzugefügt, der sämtliche Gruppenrechte unter Unix/Linux besitzt:

sudo useradd -m lightlife -g 1000 –G adm,dialout,cdrom,sudo,audio,video,plugdev,games,users,netdev,input

Nun muss für den soeben hinzugefügten Benutzer auch ein Passwort festgelegt warden. Man muss dabei beachten, dass zuerst der Befehl und dann der Name des Benutzers, in diesem Fall *lightlife*, eingegeben werden muss.

sudo passwd lightlife



Nach doppeltem eingeben des Passwortes sollte die obige Meldung angezeigt werden.

Nun wird die entsprechende Konfigurationsdatei angepasst, damit man nicht ständig das Passwort beim Befehlsaufruf mit sudo eingeben muss.

sudo visudo

Am Ende muss noch folgende Zeile hinzugefügt werden:

lightlife ALL=(ALL) NOPASSWD: ALL

Mit STRG + X verlässt man das File. Vorher wird abgefragt, ob man die Änderungen speichern möchte. Mit Y bestätigen und Enter drücken, dann speichert man die Änderungen.

Nun wird ein neuer Samba-Benutzer für den vorhin erstellten Benutzer *lightlife* erstellt. Zu beachten ist, dass der Benutzer, im Beispiel lightlife, auf dem Raspberry Pi als Systembenutzer existieren muss. Es ist ratsam, nicht dasselbe Passwort wie für den Systembenutzer und folglich den SSH-Zugang zu verwenden.

sudo smbpasswd -a lightlife



Als nächstes wird dem Benutzer und der Gruppe lightlife Rechte für den Ordner verwiesen, der später freigegeben werden soll.

Der Ordner wird mit folgenden Befehl erstellt, wobei mkdir für make directory steht:

sudo mkdir /lightlife

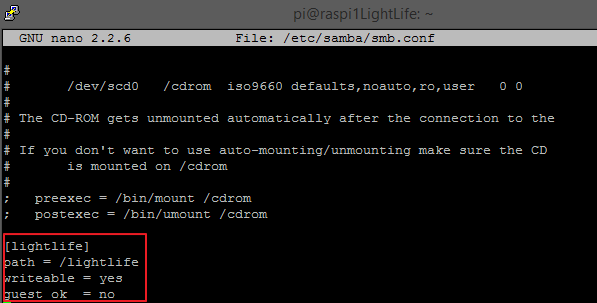
Danach müssen noch Rechte für den Ordner zugewiesen werden. „chown“ steht für Change Ownership.

sudo chown lightlife:pi /lightlife

Nun muss lediglich die Freigabe in der Konfigurationsdatei von Samba eingerichtet warden. Diese Konfigurations-Datei wird mit folgendem Befehl aufgerufen:

sudo nano /etc/samba/smb.conf

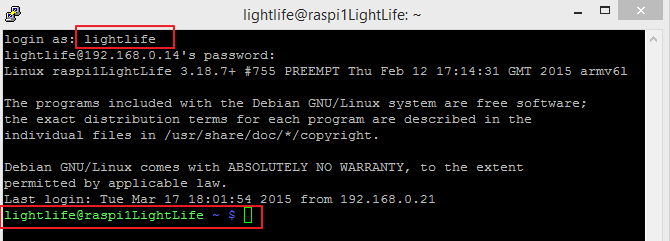
Hier wird der in der folgenden Abbildung zu sehende Nachtrag hinzugefügt. *[lightlife]* ist der Name für die Freigabe und *path* der Pfad für den vorhin erstellten Ordner. *Guest ok* wird auf nein gesetzt, da man nicht will, dass unbefugte Benutzer auf den Ordner zugreifen können.



Jetzt muss nur noch Samba neu gestartet werden, damit auch wirklich alle vorgenommenen Änderungen übernommen werden.

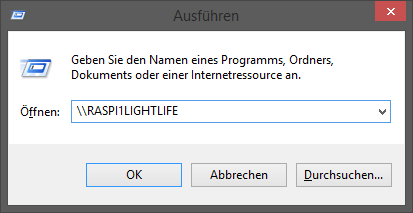
sudo /etc/init.d/samba restart

Zur Überprüfung ob bei der Benutzer-Erstellung und Einrichtung alles problemlos funktioniert hat, startet man den Raspberry mit sudo init 6 oder sudo reboot wieder neu und meldet sich anschließend mit den vorhin erstellten Daten (Benutzer: ***lightlife***, Passwort: ***test***) an.

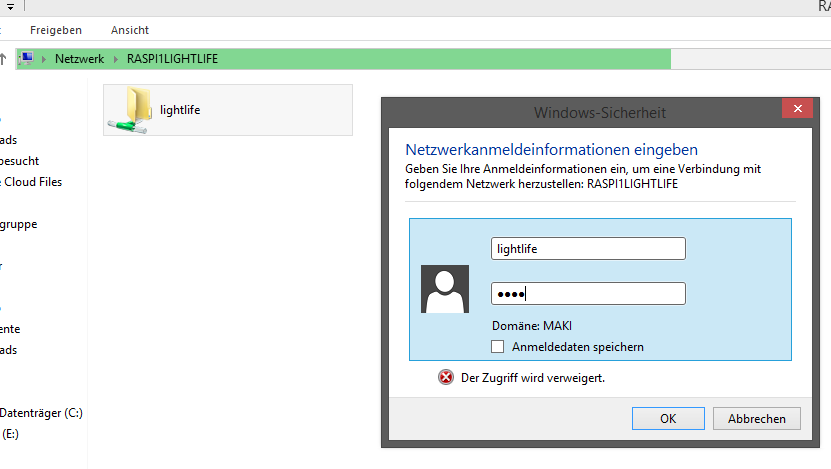


**Überprüfung der Berechtigung**

Will man nun auf den vorhin erstellten Ordner zugreifen, gibt man bei dem Fenster Ausführen (mit der Tastenkombination Windows + R) [***\\RASPI1LIGHTLIFE***](file:///\\RASPI1LIGHTLIFE) ein.

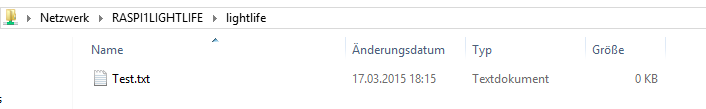


Will man nun auf den vorhin erstellten Ordner zugreifen, müsste eine Passwortabfrage kommen, da Unbefugte keinen Zugriff auf den Ordner haben sollten.

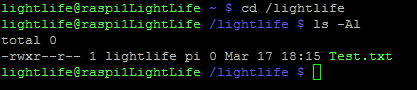


Hätte man die Security-Zeile im Samba-Authentication-Folder auskommentiert lassen (# nicht entfernt), würde man jetzt ohne Passwort auf den Ordner zugreifen können.

Zur Überprüfung auf vorhandene Schreibreichte versucht man lediglich beispielsweise ein Text-Dokument im Ordner zu erstellen.



Gibt man nun cd /lightlife und anschließend ls –Al bei PuTTY ein, kann man sich die Daten im Ordner lightlife ansehen. Mit sudo nano test.txt kann die Datei dann beispielsweise geändert werden.



Weiters wird nun nicht darauf eingegangen, da dies ansonsten den Rahmen sprengen würde.

**GCC 4.9 Compiler installieren**

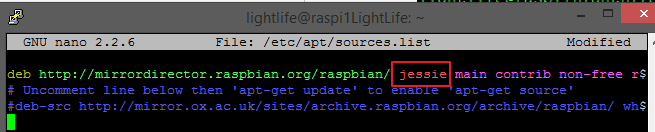
Zum Compilieren von Programmen sollte der neueste Compiler installiert sein. Ist die Compiler-Version veraltet, können Fehlermeldungen und Errors auftreten, da gewisse Befehle eventuell noch gar nicht existieren. Um dies zu vermeiden wird im Folgenden beschrieben, wie der neue Compiler installiert werden kann. Doch vorher sollte man alles auf den neuesten Stand bringen. Dies geschieht, wie gewohnt, mit folgenden Befehlen:

sudo apt-get update

sudo apt-get upgrade

Anschließend muss in der Sourcelist-Datei die Linux Version von Wheezie auf Jessie geändert werden, um die neuesten Pakete erhalten zu können. (ACHTUNG: Mit Vorsicht zu genießen, da nicht alle Pakete von Jessie mit dem Raspberry Pi kompatibel sind!)

sudo nano /etc/apt/sources.list



Danach muss die Package-List auf den neuesten Stand gebracht warden.

sudo apt-get update

Jetzt kann der GCC 4.9 Compiler installiert warden.

sudo apt-get install gcc-4.9 g++-4.9

Anschließend muss wieder auf Wheezie umgestellt und sicherheitshalber erneut ein Update gemacht werden. Um das Fehlverhalten mancher Jessie Paktete auf dem Raspberry Pi zu vermeiden.

*sudo nano /etc/apt/sources.list*

sudo apt-get update

Zur Überprüfung des Compilers wird nun ein CPP-File erstellt und ein vorhandenes Test-Programm reinkopiert und compiliert.

**sudo nano lambda\_test.cpp**

#include<iostream>

#include<complex>

int main() {

// Store a generalized lambda, that squares a number, in a variable

auto func = [](auto input) { return input \* input; };

// Usage examples:

// square of an int

std::cout << func(10) << std::endl;

// square of a double

std::cout << func(2.345) << std::endl;

// square of a complex number

std::cout << func(std::complex<double>(3, -2)) << std::endl;

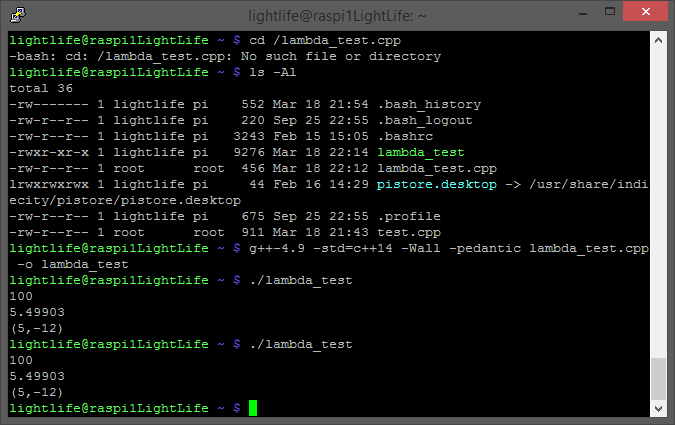
return 0;

}

Danach wird die vorhin erstellte CPP-Datei gespeichert und wie folgt compiliert:

g++-4.9 -std=c++14 -Wall -pedantic lambda\_test.cpp -o lambda\_test

Nun kann noch sicherheitshalber geprüft werden, ob das Programm einwandfrei funktioniert und die Funktionalität des Programmes gegeben ist.



Es funktioniert, und der GCC 4.9 Compiler erfüllt seinen Dienst.

**Installation von WiringPi**

Bevor WiringPi installiert werden kann, muss GIT installiert werden.

sudo apt-get install git-core

Danach wird wieder ein Update und Upgrade gemacht. Die Befehle können aus dem obigen Teil entnommen werden. Um WiringPi von GIT zu erhalten, wird folgender Befehl benötigt:

git clone git://git.drogon.net/wiringPi

Zur Installation von WiringPi selbst muss man im richtigen Verzeichnis stehen, dafür gibt es den Befehl

cd wiringPi

Im richtigen Verzeichnis gibt man nur noch einen Befehl ein und WiringPi wird installiert.

./build

Danach ist WiringPi installiert und sollte einwandfrei funktionieren.

**Installation von LightLife Programmen**

Alle Kommando ev. mit „sudo“ vorangestellt ausführen

1. In Directory gewünschtes wechseln oder ein Directory erstellen

Cd~ oder mkdir Source

1. LightLife Repository von github clonen:

git clone <https://github.com/friedrichwagner/LightLifeBox.git> (erstellt im aktuellen Ordner ein Subdirectory “LightLifeBox”)

1. Ins Verzeichnis wechseln:

cd LightLifeBox

1. Letzen Branch auschecken:

git checkout -B Version2.0 --track origin/Version2.0

1. Ins „00-llbox“ Verzeichnis wechseln:

cd 00\*

1. “-bin” Verzeichnis erstellen:

Mkdir \_bin

1. Llbox executable erstellen:

Make

(kompiliert und linkt “llbox” – hoffentlich ohne Fehler ☺)

1. Berechtigungen für directory ändern (damit man z.B. von Windows über Samba und einen anderen User schreiben kann)

sudo chmod 777 LightLifeBox

1. Autostart von „llbox“ einrichten: (http://www.stuffaboutcode.com/2012/06/raspberry-pi-run-program-at-start-up.html)

sudo cp llbox.sh /etc/init.d

sudo chmod 755 /etc/init.d/llbox.sh

sudo update-rc.d llbox.sh defaults

1. Test Start/Stop:

sudo /etc/init.d/llbox.sh start

sudo /etc/init.d/llbox.sh stop

1. Remove script from startup:

sudo update-rc.d -f  llbox.sh remove

ps ax | grep llbox

1. <http://www.raspberrypi.org> [↑](#footnote-ref-1)
2. <http://www.chip.de/downloads/Win32-Disk-Imager_46121030.html> [↑](#footnote-ref-2)
3. <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> [↑](#footnote-ref-3)