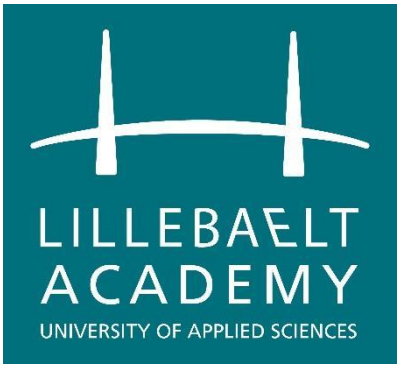
IT Technology
Remote voice control of a system of robot vehicles
Appendix for project report



LILLEBAELT ACADEMY
UNIVERSITY OF APPLIED SCIENCE

Authors
Dainius Čeliauskas
Juan Mohamad

dain0084@edu.eal.dk
juan0314@edu.eal.dk

Sunday, June 3, 2018

Table of Contents

1. Simplified software installation instruction manual (Updated 18-05-30)

This manual was created by and for the project group to quickly get Raspberries working with PocketSphinx, paho-mqtt and the main Python script. The `keyphrase.dic`, `keyphrase.list`, `PocketScriptHUB.py` and `PocketScriptROBOT.py` files can be found in the project's GitHub page.

```
--------------------------------
Install Raspbian on Raspberry Pi
--------------------------------

Get Raspbian Stretch Lite iso file from internet
https://www.raspberrypi.org/downloads/raspbian/

Use Win32DiskImager to write iso file to sd card
https://sourceforge.net/projects/win32diskimager/

Create blank "ssh" file
(you can do it by creating a text file, then go to cmd, go to directory of your
file and type:
rename yourfilename.txt ssh)

Create wpa_supplicant.conf file -anywhere- with code like this:

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB
network={
        ssid="[Wifi network name]"
        psk="[Wifi network password]"
        priority=[number of priority]
}
network={
      ssid="[PEAP protected Wifi network name]"
      proto=RSN
      key_mgmt=WPA-EAP
      pairwise=CCMP
      auth_alg=OPEN
      eap=PEAP
      identity="[your ID key]"
      password=[your password in hash]
      phase1="peaplabel=0"
      phase2="auth=MSCHAPV2"
}

How to generate hash for PEAP protected network from your own password:
https://www.raspberrypi.org/forums/viewtopic.php?t=111100
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
(echo -n 'YOUR_REAL_PASSWORD' | iconv -t utf16le | openssl md4 > hash.txt)
```

Read more about PEAPv0/EAP-MSCHAPv2 here:
https://en.wikipedia.org/wiki/Protected_Extensible_Authentication_Protocol#PEAPv0_with_EAP-MSCHAPv2

Insert ssh and wpa_supplicant files into boot folder in sd card (visible in Windows)
Turn on raspberry. These files should be automatically moved from boot into where they need to be.

Find RPi IP with an IP scanner and connect to it with putty and WinSCP
https://www.advanced-ip-scanner.com/ (decent, portable IP scanner exe)
https://www.putty.org/ (for the terminal)
https://winscp.net/eng/download.php (GUI for easy exchange of files)

You can also use cmd for scanning IPs quickly using the 'arp -a' command.

Optional:
You can make the raspberry have a static IP to skip searching for it, but it's not recommended for large, constantly changing networks like Eal-Wireless. The RPi might not be able to connect to the network at all.
This article talks about it and how to do it:
https://caffinc.github.io/2016/12/raspberry-pi-3-headless/

Once connected, change the password, obviously:
passwd

Expand file system with sudo raspi-config
Update raspbian:

```
sudo apt-get update
sudo apt-get upgrade
sudo rpi-update
```

Configure USB mic

Insert mic into USB dongle
Run sudo nano /usr/share/alsa/alsa.conf and change in the file:

```
defaults.ctl.card 0
defaults.pcm.card 0
```

to

```
defaults.ctl.card 1
defaults.pcm.card 1
```

Create and edit .asoundrc file (sudo nano ~/.asoundrc), put the following:

dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
pcm.!default {
 type hw
 card 1
}

ctl.!default {
 type hw
 card 1
}
```

Boost speaker output level with alsamixer to preferable levels

Test record audio from the mic:

```
arecord -D plughw:1,0 test.wav
```

Install Sphinxbase and Pocketsphinx

Type:

```
wget
https://sourceforge.net/projects/cmusphinx/files/sphinxbase/5prealpha/sphinxbase
-5prealpha.tar.gz/download -O sphinxbase.tar.gz

wget
https://sourceforge.net/projects/cmusphinx/files/pocketsphinx/5prealpha/pocketsp
hinx-5prealpha.tar.gz/download -O pocketsphinx.tar.gz
```

Extract:

```
tar -xzvf sphinxbase.tar.gz
tar -xzvf pocketsphinx.tar.gz
```

Install bison, ALSA, swig, etc.

```
sudo apt-get install -y python python-dev python-pip build-essential bison
libasound2-dev swig git
```

Compile Sphinxbase:

```
cd sphinxbase-5prealpha
./configure --enable-fixed
make
sudo make install
```

Compile Pocketsphinx:

```
cd ../pocketsphinx-5prealpha
./configure
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
make
sudo make install
```

Go for 2 coffee breaks each while they're installing, cause both of these take super long.

Test out the installation:

```
src/programs/pocketsphinx_continuous -adcdev sysdefault -samprate 48000 -nfft 2048 -inmic yes
```

Install paho-mqtt

Type:

```
cd
sudo apt-get install paho-mqtt
```

Install pocketsphinx-python

Type:

```
cd
sudo apt-get install python-pyaudio
git clone --recursive https://github.com/cmusphinx/pocketsphinx-python/
cd pocketsphinx-python
sudo python setup.py install
```

The website for pocketsphinx-python shows the basic usage and how the code should look like.

Generate keyword list and test code

To create a new keyword list, type all the keywords in a txt file, new line for each keyword or combo
Go to http://www.speech.cs.cmu.edu/tools/lmtool-new.html, choose your file and click "Compile knowledge base"
Then download the .dic and .list files.
The .dic file tells the software how to pronounce words, while the .list file shows the thresholds for each keyword.

Put keyphrase.dic, keyphrase.list and PocketScriptHUB.py (or PocketScriptROBOT.py) files into pocketsphinx-python directory
Allow permission to run the script with "chmod 777 PocketScript*.py"
Run PocketScript*.py and see if it works correctly.
Try testing for accuracy, speed, response times, etc.

dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
------------------------------------------------------------------------
Make script start automatically at boot (and when opening SSH terminal)
------------------------------------------------------------------------

Type:

sudo nano /home/pi/.bashrc

In the editor, add in the last line:

sudo python /home/pi/pocketsphinx-python/PocketScript*.py

Then reboot the pi.
Now the script should work when the raspberry turns on. Also it can be disabled
with Ctrl-C.

Note: by default the .bashrc file only loads upon successful login through SSH.
In order for the .bashrc file to load on boot, you must have "Console Autologin"
enabled in raspi-config under "Boot options".
```

2. PocketScriptHUB

Script used for the hub device. Some lines of code may be moved to new lines because of formatting.

```
1  #!/usr/bin/python
2
3  import sys, os, pyaudio, time, socket, fcntl, struct
4  from pocketsphinx.pocketsphinx import *
5  from sphinxbase.sphinxbase import *
6  import RPi.GPIO as GPIO
7  from threading import Thread
8  import paho.mqtt.client as mqtt
9  import paho.mqtt.publish as publish
10
11 class Subscriber(Thread):
12
13     def run(self):
14         def on_connect(client, userdata, flags, rc):
15             print("Connected with result code "+str(rc))
16             client.subscribe(MQTT_PATH)
17
18         def on_message(client, userdata, msg):
19             print(msg.topic+" "+str(msg.payload))
20         try:
21             MQTT_SERVER = "iot.eclipse.org"
22             MQTT_PATH = "voice_mqtt/measurements/#"
23             client = mqtt.Client()
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
            client.on_connect = on_connect
            client.on_message = on_message
            client.connect(MQTT_SERVER, 1883, 60)
            client.loop_forever()
        except KeyboardInterrupt:
            GPIO.cleanup()
            #set event that causes it to gracefully quit


#---------defs------------

def getHwAddr(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    info = fcntl.ioctl(s.fileno(), 0x8927,  struct.pack('256s',
ifname[:15]))
    return '-'.join(['%02x' % ord(char) for char in info[18:24]])

def blinkdiode():
    GPIO.output(ledpin, GPIO.HIGH)
    time.sleep(2)
    GPIO.output(ledpin, GPIO.LOW)
    time.sleep(2)
    GPIO.output(ledpin, GPIO.HIGH)
    time.sleep(2)
    GPIO.output(ledpin, GPIO.LOW)

def invalidcommand():
    print("Control word must be said before commands")

#-----LED setup (optional)----------

GPIO.setmode(GPIO.BCM)

ledpin = 4

GPIO.setup(ledpin, GPIO.OUT, initial=GPIO.LOW)

#------PocketSphinx setup----------

modeldir = "/home/pi/pocketsphinx-5prealpha/model/"

config = Decoder.default_config()
config.set_string('-hmm', os.path.join(modeldir, 'en-us/en-us'))
config.set_string('-dict', '/home/pi/pocketsphinx-python/keyphrase.dic')
config.set_string('-kws', '/home/pi/pocketsphinx-python/keyphrase.list')
config.set_float('-samprate', 16000.0)
config.set_int('-nfft', 512)

p = pyaudio.PyAudio()
stream = p.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True,
frames_per_buffer=1024)
```

```
stream.start_stream()

decoder = Decoder(config)
decoder.start_utt()
try:
    Subscriber().start()
    Subscriber.daemon = True
    mac = raw_input("Type MAC address of robot you want to control.\n")
    print("Connection set to: " +str(mac))
    MQTT_SERVER = "iot.eclipse.org"
    MQTT_PATH = ("voice_mqtt/commands/" +str(mac))
    print("Ready to listen")
    control = False
    while True:
        if (control == True):
            GPIO.output(ledpin, GPIO.HIGH)
        else:
            GPIO.output(ledpin, GPIO.LOW)

        buf = stream.read(1024, exception_on_overflow = False)

        decoder.process_raw(buf, False, False)

        if decoder.hyp() != None:
            print ([(seg.word) for seg in decoder.seg()])
            #print ([(seg.word, seg.prob, seg.start_frame, seg.end_frame)
for seg in decoder.seg()])
            print ("Detected keyword, restarting search")
            # command execution

            if (seg.word == 'BLINK DIODE '):
                blinkdiode()
                print('Blinking diode')

            elif (seg.word == 'BEGIN '):
                control = True
                publish.single(MQTT_PATH, seg.word, hostname=MQTT_SERVER)
                print('Control word accepted')

            elif (seg.word == 'STOP '):
                control = False
                publish.single(MQTT_PATH, seg.word, hostname=MQTT_SERVER)
                print('Stopping')

            elif (seg.word == 'DRIVE '):
                    publish.single(MQTT_PATH, seg.word,
hostname=MQTT_SERVER)
                    print('Going forward')
                    if (control == False):
                        invalidcommand()

```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
126             elif (seg.word == 'BACK '):
127                     publish.single(MQTT_PATH, seg.word,
128 hostname=MQTT_SERVER)
129                     print('Going backward')
130                     if (control == False):
131                         invalidcommand()
132 
133             elif (seg.word == 'LEFT '):
134                     publish.single(MQTT_PATH, seg.word,
135 hostname=MQTT_SERVER)
136                     print('Turning left')
137                     if (control == False):
138                         invalidcommand()
139 
140             elif (seg.word == 'RIGHT '):
141                     publish.single(MQTT_PATH, seg.word,
142 hostname=MQTT_SERVER)
143                     print('Turning right')
144                     if (control == False):
145                         invalidcommand()
146 
147             elif (seg.word == 'FIRST '):
148                     publish.single(MQTT_PATH, seg.word,
149 hostname=MQTT_SERVER)
150                     print('Changed duty cycle to 25')
151                     if (control == False):
152                         invalidcommand()
153 
154             elif (seg.word == 'SECOND '):
155                     publish.single(MQTT_PATH, seg.word,
156 hostname=MQTT_SERVER)
157                     print('Changed duty cycle to 50')
158                     if (control == False):
159                         invalidcommand()
160 
161             elif (seg.word == 'THIRD '):
162                     publish.single(MQTT_PATH, seg.word,
163 hostname=MQTT_SERVER)
164                     print('Changed duty cycle to 75')
165                     if (control == False):
166                         invalidcommand()
167 
168             elif (seg.word == 'FOURTH '):
169                     publish.single(MQTT_PATH, seg.word,
170 hostname=MQTT_SERVER)
171                     print('Changed duty cycle to 99')
172                     if (control == False):
173                         invalidcommand()
174 
175             decoder.end_utt()
176             time.sleep(0.02)
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
177             decoder.start_utt()
178
179 except KeyboardInterrupt:
180     print("Exception: KeyboardInterrupt")
181     GPIO.cleanup()
```

3. PocketScriptROBOT

Script used for all of the robot devices. Some lines of code may be moved to new lines because of formatting.

```
1  #!/usr/bin/python
2
3  import sys, os, pyaudio, time, socket, fcntl, struct
4  from pocketsphinx.pocketsphinx import *
5  from sphinxbase.sphinxbase import *
6  import RPi.GPIO as GPIO
7  from threading import Thread
8  import paho.mqtt.publish as publish
9  import paho.mqtt.client as mqtt
10
11 class Publisher(Thread):
12     def run(self):
13         def getHwAddr(ifname):
14             s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
15             info = fcntl.ioctl(s.fileno(), 0x8927,  struct.pack('256s',
16 ifname[:15]))
17             return '-'.join(['%02x' % ord(char) for char in info[18:24]])
18         try:
19             MQTT_SERVER = "iot.eclipse.org"
20             mac = getHwAddr('wlan0')
21             MQTT_PATH = ("voice_mqtt/measurements/" +str(mac))
22             global rpm
23             count = 0
24             while True:
25                 count += 1
26                 #print(count)
27                 if (count >= 60):
28                     print("RPM: ", rpm)
29                     publish.single(MQTT_PATH, "RPM of robot: " +str(rpm),
30 hostname=MQTT_SERVER)
31                     count = 0
32                     rpm = 0
33                 time.sleep(1)
34         except KeyboardInterrupt:
35             GPIO.cleanup()
36             #set event that causes it to gracefully quit
37
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
class Subscriber(Thread):

    def run(self):
        def on_connect(client, userdata, flags, rc):
            print("Connected with result code "+str(rc))
            client.subscribe(MQTT_PATH)

        def on_message(client, userdata, msg):
            global control
            print(msg.topic+" "+str(msg.payload))
            phrase = msg.payload
            control = command_execution(phrase, control)

        try:
            global mac
            control = False
            MQTT_SERVER = "iot.eclipse.org"
            MQTT_PATH = ("voice_mqtt/commands/"+str(mac))
            client = mqtt.Client()
            client.on_connect = on_connect
            client.on_message = on_message
            client.connect(MQTT_SERVER, 1883, 60)
            client.loop_forever()
        except KeyboardInterrupt:
            GPIO.cleanup()
            #set event that causes it to gracefully quit

#---------defs------------

def getHwAddr(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    info = fcntl.ioctl(s.fileno(), 0x8927,  struct.pack('256s',
ifname[:15]))
    return '-'.join(['%02x' % ord(char) for char in info[18:24]])

def rpm_increment(tachometerpin): # event detect callback, on separate
thread
    global rpm
    rpm += 1
    #print(rpm) # tachometer debugging

def blinkdiode():
    GPIO.output(ledpin, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(ledpin, GPIO.LOW)
    time.sleep(1)
    GPIO.output(ledpin, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(ledpin, GPIO.LOW)

def forward():
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
    GPIO.output(left1, GPIO.HIGH)
    GPIO.output(left2, GPIO.LOW)
    GPIO.output(right1, GPIO.HIGH)
    GPIO.output(right2, GPIO.LOW)

def backward():
    GPIO.output(left1, GPIO.LOW)
    GPIO.output(left2, GPIO.HIGH)
    GPIO.output(right1, GPIO.LOW)
    GPIO.output(right2, GPIO.HIGH)

def stop():
    GPIO.output(motorpins, GPIO.LOW)

def leftturn():
    GPIO.output(right1, GPIO.LOW)
    GPIO.output(right2, GPIO.LOW)
    GPIO.output(left1, GPIO.HIGH)
    GPIO.output(left2, GPIO.LOW)

def rightturn():
    GPIO.output(right1, GPIO.HIGH)
    GPIO.output(right2, GPIO.LOW)
    GPIO.output(left1, GPIO.LOW)
    GPIO.output(left2, GPIO.LOW)

def changespeed(dc):
    motor1.ChangeDutyCycle(dc)
    motor2.ChangeDutyCycle(dc)

def invalidcommand():
    print("Control word must be said before commands")

def command_execution(phrase, control):
    print ("Detected keyword, restarting search")
    # command execution

    if (phrase == 'BLINK DIODE '):
        blinkdiode()
        print('Blinking diode')

    elif (phrase == 'BEGIN '):
        control = True
        print('Control word accepted')

    elif (phrase == 'STOP '):
        stop()
        control = False
        print('Stopping')

    elif (phrase == 'DRIVE '):
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
        if (control == True):
            forward()
            print('Going forward')
        else:
            invalidcommand()

    elif (phrase == 'BACK '):
        if (control == True):
            backward()
            print('Going backward')
        else:
            invalidcommand()

    elif (phrase == 'LEFT '):
        if (control == True):
            leftturn()
            print('Turning left')
        else:
            invalidcommand()

    elif (phrase == 'RIGHT '):
        if (control == True):
            rightturn()
            print('Turning right')
        else:
            invalidcommand()

    elif (phrase == 'FIRST '):
        if (control == True):
            changespeed(25)
            print('Changed duty cycle to 25')
        else:
            invalidcommand()

    elif (phrase == 'SECOND '):
        if (control == True):
            changespeed(50)
            print('Changed duty cycle to 50')
        else:
            invalidcommand()

    elif (phrase == 'THIRD '):
        if (control == True):
            changespeed(75)
            print('Changed duty cycle to 75')
        else:
            invalidcommand()

    elif (phrase == 'FOURTH '):
        if (control == True):
            changespeed(99)
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
            print('Changed duty cycle to 99')
        else:
            invalidcommand()

    return control

#-----motor setup----------

GPIO.setmode(GPIO.BCM)

ledpin = 4
enable1 = 17
enable2 = 23
left1 = 27
left2 = 22
right1 = 25
right2 = 24

tachometerpin = 26

enablerpins = [enable1,enable2]
motorpins = [left1,left2,right1,right2]

GPIO.setup(enablerpins, GPIO.OUT, initial=GPIO.HIGH)
GPIO.setup(motorpins, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(ledpin, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(tachometerpin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

motor1 = GPIO.PWM(enable1,25)
motor1.start(25)

motor2 = GPIO.PWM(enable2,25)
motor2.start(25)

#------PocketSphinx setup----------

modeldir = "/home/pi/pocketsphinx-5prealpha/model/"

config = Decoder.default_config()
config.set_string('-hmm', os.path.join(modeldir, 'en-us/en-us'))
config.set_string('-dict', '/home/pi/pocketsphinx-python/keyphrase.dic')
config.set_string('-kws', '/home/pi/pocketsphinx-python/keyphrase.list')
config.set_float('-samprate', 16000.0)
config.set_int('-nfft', 512)

p = pyaudio.PyAudio()
stream = p.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True,
frames_per_buffer=1024)
stream.start_stream()

```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
decoder = Decoder(config)
decoder.start_utt()
try:
    mac = getHwAddr('wlan0')
    Publisher().start()
    Publisher.daemon = True
    Subscriber().start()
    Subscriber.daemon = True
    rpm = 0
    GPIO.add_event_detect(tachometerpin, GPIO.FALLING,
callback=rpm_increment, bouncetime=200)
    blinkdiode()
    print("Ready to listen")
    control = False
    while True:

        if (control == True):
            GPIO.output(ledpin, GPIO.HIGH)
        else:
            GPIO.output(ledpin, GPIO.LOW)

        buf = stream.read(1024, exception_on_overflow = False)

        decoder.process_raw(buf, False, False)

        if decoder.hyp() != None:
            print ([(seg.word) for seg in decoder.seg()])
            #print ([(seg.word, seg.prob, seg.start_frame, seg.end_frame)
for seg in decoder.seg()])
            control = command_execution(seg.word, control)

            decoder.end_utt()
            time.sleep(0.02)
            decoder.start_utt()

except KeyboardInterrupt:
    print("Exception: KeyboardInterrupt")
    GPIO.cleanup()
```

4. VoiceScript (old)

This is an earlier iteration of the speech recognition script, intended to be used with the VoiceMacro program. Some lines of code may be moved to new lines because of formatting.

```
#!/usr/bin/python

import sys, tty, termios, time
import RPi.GPIO as GPIO
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
5
6   GPIO.setmode(GPIO.BCM)
7
8   ledpin = 4
9
10  enable1 = 17
11  enable2 = 23
12  left1 = 27
13  left2 = 22
14  right1 = 25
15  right2 = 24
16
17  GPIO.setup(ledpin, GPIO.OUT, initial=GPIO.LOW)
18
19  GPIO.setup(enable1, GPIO.OUT, initial=GPIO.HIGH)
20
21  GPIO.setup(enable2, GPIO.OUT, initial=GPIO.HIGH)
22
23  GPIO.setup(left1, GPIO.OUT, initial=GPIO.LOW)
24
25  GPIO.setup(left2, GPIO.OUT, initial=GPIO.LOW)
26
27  GPIO.setup(right1, GPIO.OUT, initial=GPIO.LOW)
28
29  GPIO.setup(right2, GPIO.OUT, initial=GPIO.LOW)
30
31  motor1 = GPIO.PWM(17,25) # set motor1 pwm frequency here
32
33  motor1.start(25)
34
35  motor2 = GPIO.PWM(23,25) # set motor2 pwm frequency here
36
37  motor2.start(25)
38
39  def getch():
40
41      fd = sys.stdin.fileno()
42
43      old_settings = termios.tcgetattr(fd)
44
45      try:
46
47          tty.setraw(sys.stdin.fileno())
48
49          ch = sys.stdin.read(1)
50
51      finally:
52
53          termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
54
55      return ch
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
56
57  def blinkdiode():
58      GPIO.output(ledpin, GPIO.HIGH)
59      time.sleep(2)
60      GPIO.output(ledpin, GPIO.LOW)
61      time.sleep(2)
62      GPIO.output(ledpin, GPIO.HIGH)
63      time.sleep(2)
64      GPIO.output(ledpin, GPIO.LOW)
65
66  def forward():
67      GPIO.output(left1, GPIO.HIGH)
68      GPIO.output(left2, GPIO.LOW)
69      GPIO.output(right1, GPIO.HIGH)
70      GPIO.output(right2, GPIO.LOW)
71
72  def backward():
73      GPIO.output(left1, GPIO.LOW)
74      GPIO.output(left2, GPIO.HIGH)
75      GPIO.output(right1, GPIO.LOW)
76      GPIO.output(right2, GPIO.HIGH)
77
78  def stop():
79      GPIO.output(left1, GPIO.LOW)
80
81      GPIO.output(left2, GPIO.LOW)
82      GPIO.output(right1, GPIO.LOW)
83      GPIO.output(right2, GPIO.LOW)
84
85
86  def leftturn():
87      GPIO.output(left1, GPIO.LOW)
88      GPIO.output(left2, GPIO.LOW)
89      GPIO.output(right1, GPIO.HIGH)
90      GPIO.output(right2, GPIO.LOW)
91
92
93  def rightturn():
94      GPIO.output(left1, GPIO.HIGH)
95      GPIO.output(left2, GPIO.LOW)
96      GPIO.output(right1, GPIO.LOW)
97      GPIO.output(right2, GPIO.LOW)
98
99  def speed25():
100     motor1.ChangeDutyCycle(25)
101     motor2.ChangeDutyCycle(25)
102
103 def speed50():
104     motor1.ChangeDutyCycle(50)
105     motor2.ChangeDutyCycle(50)
106
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk

```
107 def speed75():
108     motor1.ChangeDutyCycle(75)
109     motor2.ChangeDutyCycle(75)
110 
111 def speed100():
112     motor1.ChangeDutyCycle(99)
113     motor2.ChangeDutyCycle(99)
114 
115 try:
116     print ("Program started. Press X or say 'exit program' to exit")
117     while(True):
118 
119         char = getch()
120 
121         if(char == "b"):
122             blinkdiode()
123         if(char == "w"):
124 
125             forward()
126         if(char == "s"):
127 
128             backward()
129         if(char == "q"):
130 
131             stop()
132         if(char == "a"):
133 
134             leftturn()
135         if(char == "d"):
136 
137             rightturn()
138         if(char == "1"):
139             print ("Changed duty cycle to 25")
140             speed25()
141 
142         if(char == "2"):
143             print ("Changed duty cycle to 50")
144             speed50()
145         if(char == "3"):
146             print ("Changed duty cycle to 75")
147             speed75()
148         if(char == "4"):
149             print ("Changed duty cycle to 100")
150             speed100()
151         if(char == "x"):
152 
153             raise KeyboardInterrupt
154 
155 except KeyboardInterrupt:
156     GPIO.cleanup()
```


dain0084@edu.eal.dk
juan0314@edu.eal.dk