

APSC 258: Lab 2 Manual

Andre Cox

June 9, 2022

Contents

1	Introduction	2
2	Start of the Lab	3
2.1	Setup	3
2.2	Code	3
2.3	Testing	4
2.4	Your Turn	5

Chapter 1

Introduction

In the first lab you were able to control the PiCar using the provided controller software. In this lab you will see how we can use Python to control the PiCar. To do this we will explain how to use an API (Application Programming Interface).



Note: API's are used commonly in software engineering and are used to allow programs to communicate with each other. For instance a weather app could use a weather API to request weather data.

Before you start your lab it is a good idea to familiarize yourself with the PiCar's (Python) API; the Javascript API is not needed. You can find the API documentation here: [PiCar API Documentation](#) During the lab we will go over this in more detail.

As well it is a good idea to install VSCode and Python 3 on your laptop as this will save time during the lab.

Chapter 2

Start of the Lab

2.1 Setup

On your laptop you will need to install VSCode and Python 3. Once installed open VSCode and click File ⇒ Open Folder and select a convenient folder. This will be your working directory (Where our project files for this lab will be stored). Now you can create a new Python file by clicking File ⇒ New File and selecting Python. You can call this file straight.py. Now we need to run some commands to install the required packages. Click on Terminal ⇒ New Terminal. A Terminal should open at the bottom of the screen. Type the following commands into the Terminal:

```
1 pip install socketio
```

This will install the socketio package. We will use this to connect to the PiCar.

2.2 Code

Now we will write some code to make the PiCar move forwards then backwards. This code will be written in steps and should guide you through the process of writing code to control the PiCar. The first step is to import the packages.

```
1 # first we import socketio we use this to connect to the PiCar
2 import socketio
3 # next we import time we will use this to delay the code
4 import time
```

Next we will create a socketio object. This is a special variable that contains functions that allow us to communicate with the PiCar.

```
1 # create a socketio object
2 # socketio is a module that allows us to easily use websockets.
3 # Websockets are a protocol that allows two programs to communicate with each other in real time
4 # Over the internet.
5 sio = socketio.Client()
```

Now we will tell socketio to connect to the PiCar. To do this we will need to know the IP address of the PiCar. An IP address is essentially a number that represents the location of a

computer on the internet. In our instance we need to know where the Raspberry Pi is located to connect to it. Because all of the PiCar's are running the same software the IP address will always be "192.168.0.10" Let's write some code to do this.

```
1      # connect to the PiCar
2      # connects using 192.168.0.10:3000
3      # we try to connect if we fail we tell the user
4      # that they probably are not connected to the PiCar network
5      try:
6          sio.connect('http://192.168.0.10:3000')
7      except:
8          print("Failed to connect to PiCar")
9          print("Check that your laptop is connected to the PiCar network")
```

Now you may have noticed that the IP address ends with :3000. This is called a port, we use ports because we may have multiple services that we may want to connect to. For instance when you visit a website that uses port 80 or 443, Remote Desktop uses port 3389. In our case our socket server runs on port 3000. Now we understand what's going on lets actually make our car move!

```
1      # now we can start moving the car
2      # we set the angle of the steering to 90 degrees
3      # this is the middle position (180 being fully right and 0 being fully left)
4      sio.emit('steer', 90)
5      # we set the speed to 50%
6      sio.emit('drive', 50)
7      # we wait for 5 second this is how long the car will move for
8      time.sleep(5)
9      # now we stop the car by setting the speed to 0%
10     sio.emit('speed', 0)
11     # we stay stopped for 2 seconds
12     time.sleep(2)
13     # now we reverse by setting the speed to -50%
14     sio.emit('drive', -50)
15     # we wait for 5 seconds again
16     time.sleep(5)
17     # now we stop the car by setting the speed to 0%
18     # don't forget to stop the car!
19     # Otherwise it'll run away!
20     sio.emit('speed', 0)
```

2.3 Testing

Now we will test to see if the code we wrote works. To do this first turn on the PiCar and connect to the WiFi network. Then in visual studio code you can press the run button in the top right corner. It looks like a play arrow. This will run the code we wrote. If all goes well you should see the PiCar move forwards then backwards. Otherwise you will have to debug the code.

2.4 Your Turn

Now it's your turn to write some code to control the PiCar. Can you create 2 new python files to first make the PiCar do a U-turn and second to make the PiCar do a three point turn?