APSC 258: Lab 2 Manual

Andre Cox
Scott Halston

July 15, 2022

Contents

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: APIs 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 PiCar's (Python) API; the Javascript API is not needed. You can find the API documentation here: PiCar API Documentation. During this lab, we discuss this API in more detail. It is suggested that you complete the following setup section to install VSCode and Python 3 on your laptop before attending your lab to save some time.

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 or create a folder in a convenient location on your laptop. This will be your working directory (Where our project files for this lab will be stored). Next, you can create a fresh Python file by clicking File ⇒ New File and selecting Python. You can call this file straight.py. After this, we need to run some commands to install the required packages. Click on Terminal ⇒ New Terminal. A Terminal should open at the bottom of your screen. Type the following commands into the Terminal:

```
pip install socketio
```

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

2.2 Code

In this section, 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.

```
# first we import socketio we use this to connect to the PiCar
import socketio
# next we import time we will use this to delay the code
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.

```
# create a socketio object
# socketio is a module that allows us to easily use websockets.
# Websockets are a protocol that allows two programs to
 ↪  communicate with each other in real time
# Over the internet.
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 in order to connect to it. Since 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.

```
# connect to the PiCar
# connects using 192.168.0.10:3000
# we try to connect if we fail we tell the user
# that they probably are not connected to the PiCar network
try:
    sio.connect('http://192.168.0.10:3000')
except:
    print("Failed to connect to PiCar")
    print("Check that your laptop is connected to the PiCar
     ↪  network")
```

You may have noticed that the IP address ends with ':3000'. This number attached to the IP address 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 443, a Remote desktop uses port 3389. In our case our socket server runs on port 3000, later we'll learn about another service that is running on the PiCar and using a different port. Now that we understand why the above code works, we should be able to make the PiCar move!

```
# now we can start moving the car
# we set the angle of the steering to 90 degrees
# this is the middle position (180 being fully right and 0
 ↪  being fully left)
```

```
    sio.emit('steer', 90)
    # we set the speed to 50%
    sio.emit('drive', 50)
    # we wait for 5 second this is how long the car will move for
    time.sleep(5)
    # now we stop the car by setting the speed to 0%
    sio.emit('drive', 0)
    # we stay stopped for 2 seconds
    time.sleep(2)
    # now we reverse by setting the speed to -50%
    sio.emit('drive', -50)
    # we wait for 5 seconds again
    time.sleep(5)
    # now we stop the car by setting the speed to 0%
    # don't forget to stop the car!
    # Otherwise it'll run away!
    sio.emit('drive', 0)
```

2.3 Testing

We will now test to see if the code we wrote works. To do this first turn on the PiCar and connect to the WiFi network. In visual studio code, you can press the run button in the top right corner. It looks like a green play arrow. Before running your code, ensure that your PiCar is in an area that is safe to move (i.e. not going to fall off a table). Pressing this button will run the code that we wrote. If all goes well, you should see the PiCar move forwards and then backwards. If not, something must have gone wrong. You should go back through the lab to verify that your code is correct.

2.4 Your Turn

Now it's your turn to write some code to control the PiCar. Your task is to create two new python files to first make the PiCar do a U-turn and second to make the PiCar do a three-point turn.

Chapter 3

Finishing Up

After reaching this point successfully, you have finished this lab! You are free to experiment with more advanced ways of controlling the PiCar. For instance, try using loops or if statements to run more advanced code. This is a great way to get familiar with the controls of your PiCar and to gain more experience with Python.