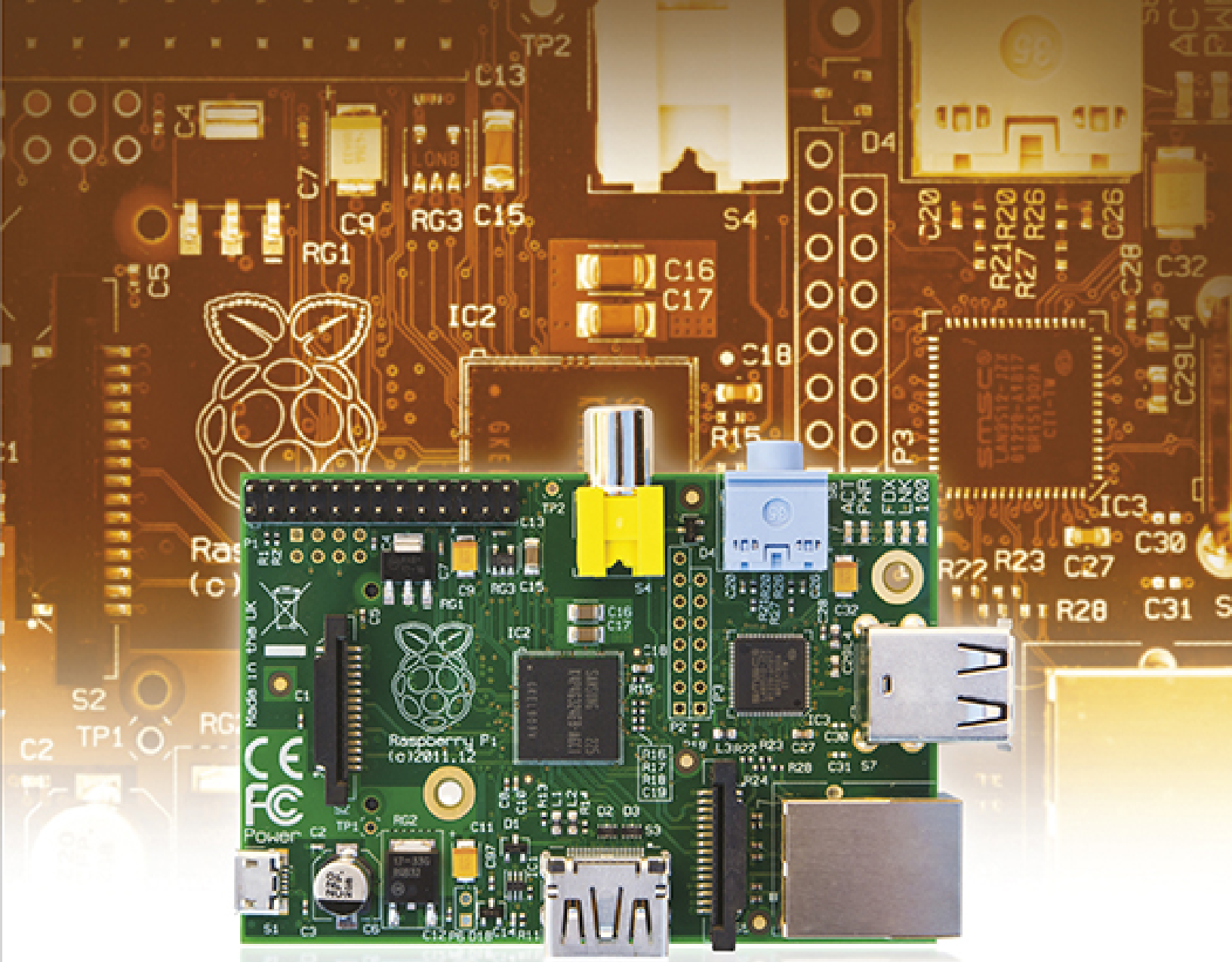


Raspberry Pi

Hardware Projects 2

Dr. Andrew Robinson
Mike Cook

Raspberry Pi® Hardware Projects

Volume 2

Mike Cook and Andrew Robinson

WILEY

Raspberry Pi® Hardware Projects, Volume 2

This edition first published 2014

Registered office

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

A catalogue record for this book is available from the British Library.

ISBN 978-1-118-58890-1 (ePub); ISBN 978-1-118-72390-6 (ePDF)

Set in Chaparral Pro Regular 10/12.5 by Indianapolis Composition Services

Table of Contents

Creating Loops

Introduction

You've got a Raspberry Pi — now what? This book has the answer; it's packed full of fun Raspberry Pi projects to inspire you. From getting your Pi generating comedy insults, controlling slot cars and drawing roto-sketches to building your own harmonograph, prepare to be entertained and amazed by your credit card–sized computer.

One word of warning: After you start you might never stop! Electronics and coding can be addictive; who knows what you might go on to make with the skills you learn from this book.

Appendix A, "Getting Your Raspberry Pi Up and Running", is a beginner's guide to your first steps with the Raspberry Pi. If you've never coded before, Appendix B, "Introductory Software Project: The Insult Generator", will get you started programming in Python. Chapter 1, "Computer-Controlled Slot Car Racing", will get you wiring up a slot car game and using it to keep score in a two-player multiple-choice quiz. Chapter 2, "Facebook-Enabled Roto-Sketch", will show you how to use rotary controls to draw elaborate designs and automatically post them to Flickr and on to Facebook. Chapter 3, "The Pendulum Pi, a Harmonograph", shows you how to create a harmonograph for producing intricate patterns using an Arduino to help the Pi with real-time data gathering. Along the way you will pick up the skills you need to develop your own ideas to make projects work exactly how you want them to.

Building and making is incredibly rewarding and satisfying. We want to get more people of the world to become producers of technology rather than consumers. The

projects in this book are starting points — step by step, they’re easy to follow so you can get results quickly. But then the real satisfaction can come, that which comes from making the project your own. At the end of each chapter there are ideas and suggestions to extend the project, together with background information to point you in the right direction. The real addictive fun begins when you see your own ideas become reality.

Welcome to the world of digital making.

Chapter 1

Computer-Controlled Slot Car Racing

In This Chapter

- Learn how to use your Raspberry Pi to enable and disable a slot car set.
- See how to make your own illuminated joystick pad.
- Discover how to use an external text file as a question bank.
- Understand the interactions between the software and hardware.

This project is a rather different twist on the multiple-choice quiz theme. Not only does it have a novel way of inputting answers, but it also has a rather novel way of keeping the score.

The idea is that you are going to hack into a slot car game and allow the Raspberry Pi to control when the game can be played. Then players can drive their cars for three seconds at a time, if they are the first to answer a question correctly. If they get the question wrong, their opponent gets the time. The game continues until one player crosses the finishing line after completing a set number of laps. The questions come from a plain text file and can be added to, or the subject of them changed. They are multiple-choice questions with four possible answers, and players indicate

their answer by moving a special joystick button. The successful player's joystick button will light up green, whereas the other player's button will light up red.

Obtaining a Slot Car Racer

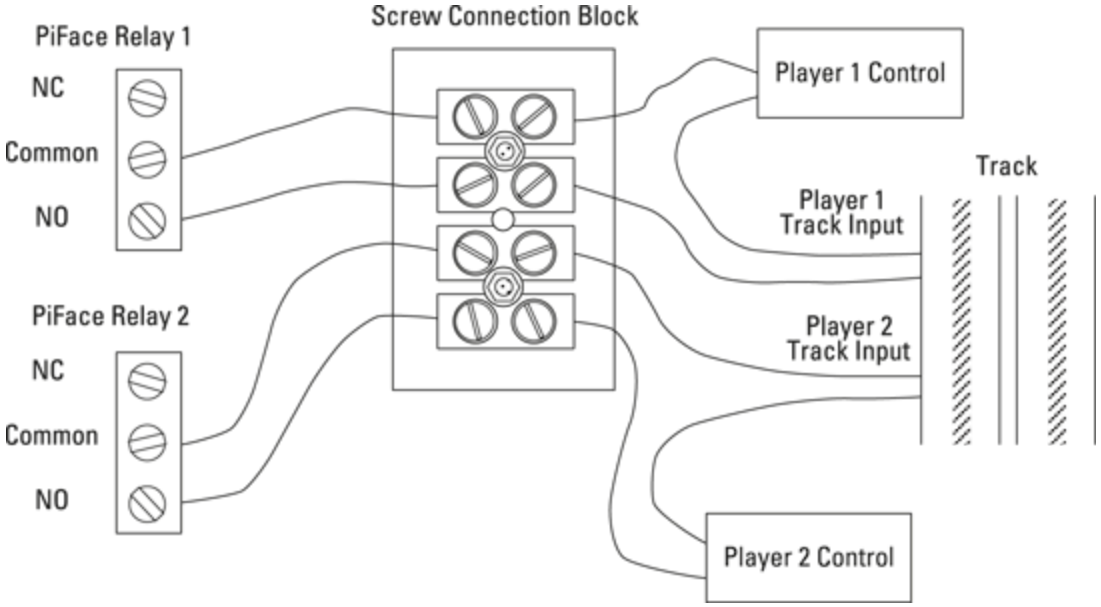
So how are you going to implement this game? First you need a slot car racing game. These come in all sorts of shapes and sizes, from sleek Formula One racing cars to heavy trucks and even grannies on Zimmer frames. In essence they are very similar: It's a race between two players. Mostly they are set up so that if you go too fast at the corners, the vehicle will come off the track, so it is not just a matter of running the cars at top speed all the time. Normally the track is some form of figure eight, so the track length can be made the same for both players. Sometimes the two vehicles cross at the same level, giving opportunities for crashes, and other times the tracks go over and under each other. Although there are very expensive racing games, some can be had cheaply in thrift shops or second-hand stores.

Hacking Your Slot Car Racer

You need to hack into your slot car racer, and, as there are lots of different types of them, I can't be too prescriptive about what you need to do. However, from the electrical point of view, it is basically all the same. What you are going to do is to wire a PiFace relay in series with each hand controller. This will involve cutting one of the two wires coming from the controller, and connecting each end of your cut wire into the NO and common relay connections. NO stands for *normally open* – this connection is only connected to the common line when the relay is energised; when the relay is not energised, that is the normal state – no electrical connection is made. Figure 1-1 shows how you can do this

using a screw connection block. These are the type you use for electrical wiring around the house. They come in various sizes, and the size you want is the smallest, which is often marked something like *3 Amps.* A sharp hobby knife can slice the two wires apart, and then you can cut one of them; it doesn't matter which one. You should cut back the insulation and then, following the diagram, attach each end to two of the connector blocks. Take the other end and run wires off to the PiFace board. Do the same for the other controller. When you want to play with your slot car game normally you simply replace the long wires trailing back to the PiFace board with a simple link. Do this close to the track connections so that you have the maximum length of wire on the hand controllers.

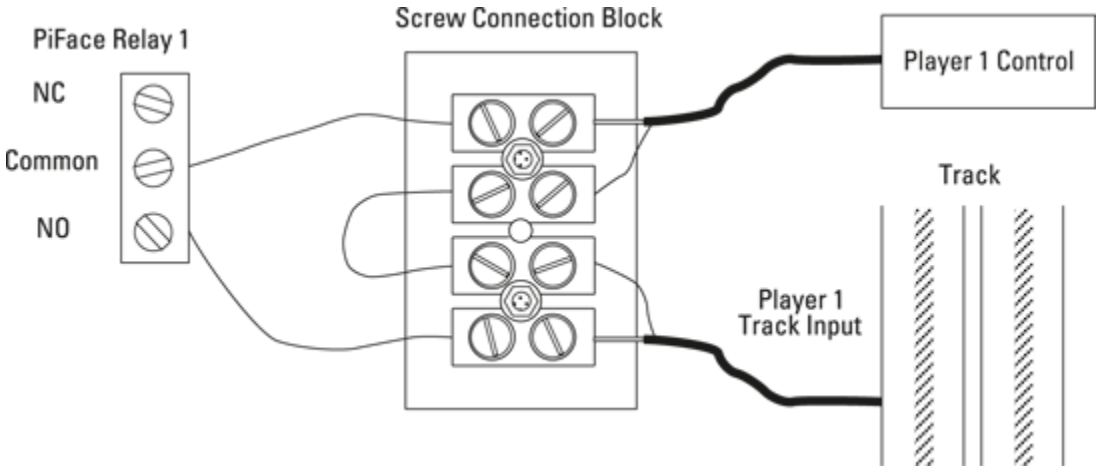
Figure 1-1: Hacking into your slot car racing game.



Sometimes the wires coming from the hand controller are all bundled into one cable, so it is impossible to cut just one wire. If this is the case, you will have to cut both of them and join the other wire back up again. This sort of thing is shown in Figure 1-2. In this case the wire consists of an inner conductor and an outer braided wire sheath. Strip

back the outer sheath and make sure that no thin strands of wire are shorting out to the other wire. Use insulation tape or heat shrink sleeving to insulate the sheath. Then wire it up as shown in Figure 1-2. Note that this diagram is for one controller; you will have to duplicate this for the other player's controller.

Figure 1-2: Hacking into your slot car racing game if you can't cut just a single wire.



Testing Your Slot Car Racer Hack

Now you need to test the slot car hack. Power up your slot car racing game as normal and run the program in Listing 1-1.

Listing 1-1 Slot Car Racer Hack Test

```
#!/usr/bin/env python
"""
Slot Racer Hack tester on the PiFace board
"""

import piface.pfio as pfio      # piface library
pfio.init()                     # initialise pfio

def main():
        lastInput =0
```

```
        print "Slot Racer Hack test press the two input ⮌
switches"
        print "on the PiFace board to change who is racing"
        print "ctrl - C to quit"
        while True :
            buttons = pfio.read_input()
            if (buttons & 1) ==1 and buttons != lastInput:
               print "player 1 racing"
               pfio.write_output(0x01)
            if (buttons & 2) ==2 and buttons != lastInput:
               print "player 2 racing"
               pfio.write_output(0x02)
            lastInput = buttons

if __name__ == '__main__':
    main()
```

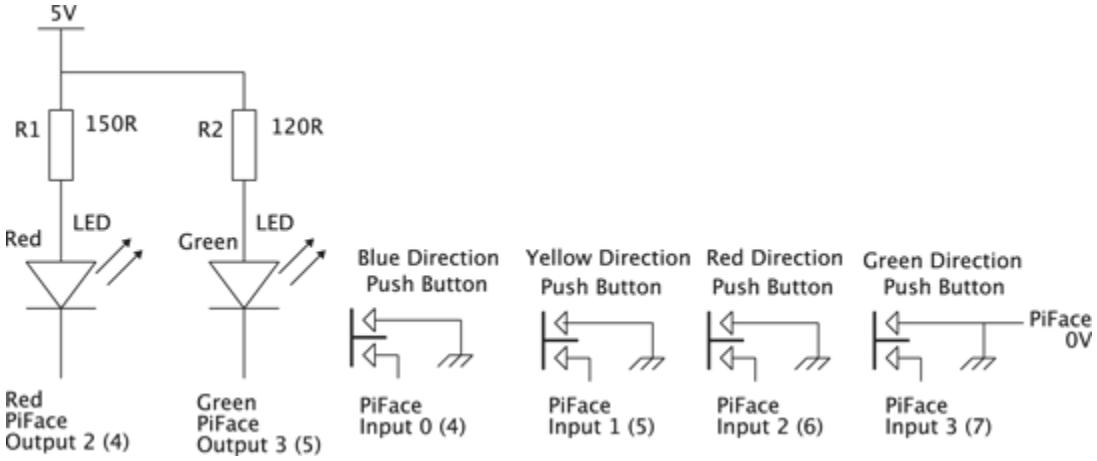
You will see it is a very simple program that just energised each of the relays depending on if you press one of two buttons on the end of the PiFace board. A message is printed to say which player is racing each time it changes. Make sure that the players are the right way around and that you can play when the console message says you can. If you find the control is the wrong way around – that is, when it says you can play you can't and when it says you can't you can – then you may have mixed up the NO and the NC relay connections. NC stands for *normally closed* – there is a connection between the common line and this one when the relay is not energised. If you don't hear the relay clicking at all but do see the two lower LEDs come on and off, then check that the links JP5 and JP6 are made.

Getting the Player Input

Next you need to find a way to input the players' answers. You could just arrange a row of four switches for each player along with red and green LEDs. In fact the schematic in Figure 1-3 can be implemented in exactly that way. However, at this stage in the game you can be a lot more

adventurous than that, so I am going to show you how to make an illuminated switch joystick using that same schematic. Not only is this useful for this project, but you also can use the joystick on other projects, replacing a keyboard input. You are going to mount four tactile button switches on a board, and, in each of the corners, have foam pads that are slightly taller than the switches. Then, if you put the board switch side down, you can click each switch in turn by simply pushing the board in that direction. The feel of the switch is down to the rigidity of the foam pads you use. On the track side of the board, you mount a red/green LED and cover the board in a half table tennis ball. Let's see how to do that in detail.

Figure 1-3: A schematic of the joystick button controller.



TIP

The LED needs to be a bright one: Look for one with at least 60 mcd on the red, and 40 mcd on the green at 20mA – brighter if you can get it; otherwise the switch could look a bit washed out. I found 90 mcd red and 45 mcd green, which looked good.

Making the Joystick Buttons

Take a small piece of strip board 17 holes long, and 16 strips wide. I like to take the corners off the board to give it an octagonal shape, just to make it look neater and ensure that the corners don't snag on the base. Cut the tracks on the back of the board where the dark marks are on Figure 1-4. Make sure that you count the tracks and holes carefully. Then solder a surface mount red/green LED at the center of the board, as shown between the two tracks and two track cuts. Make sure that the orientation mark on the LED is correctly aligned. There are two types of surface mount LEDs that you can get. One has the LEDs pinned out to the package in parallel; that is, the two anodes are on one side and the two cathodes on the other. This is sometimes known as a *parallel LED pinout.* The other way is known as *antiparallel,* where one anode and one cathode are together at each end; these normally have a bar or some other marker, often green, denoting the cathode. Make sure that you know which you are using. I have designed the board so that the tracks you need to cut are the same for both versions. However, the links on the component side are different for each LED type. When the LED is in place solder the two surface mount resistors as shown between the cut marks. If you haven't got surface mount resistors, then one-eighth watt, or one-tenth watt, resistors should be small enough to mount on the tracks.

Figure 1-4: The track side of the joystick button controller board.