CNC Fly Food Dispenser

Matt Wayland

2017-06-19

Contents

Chapter 1

About

1.1 Overview

Figure 1.1

1.2 Github

1.3 Contact

Matt Wayland

1.4 Colophon

This book was produced using the **bookdown** package (**?**), which was built on top of R Markdown and **knitr** (**?**).

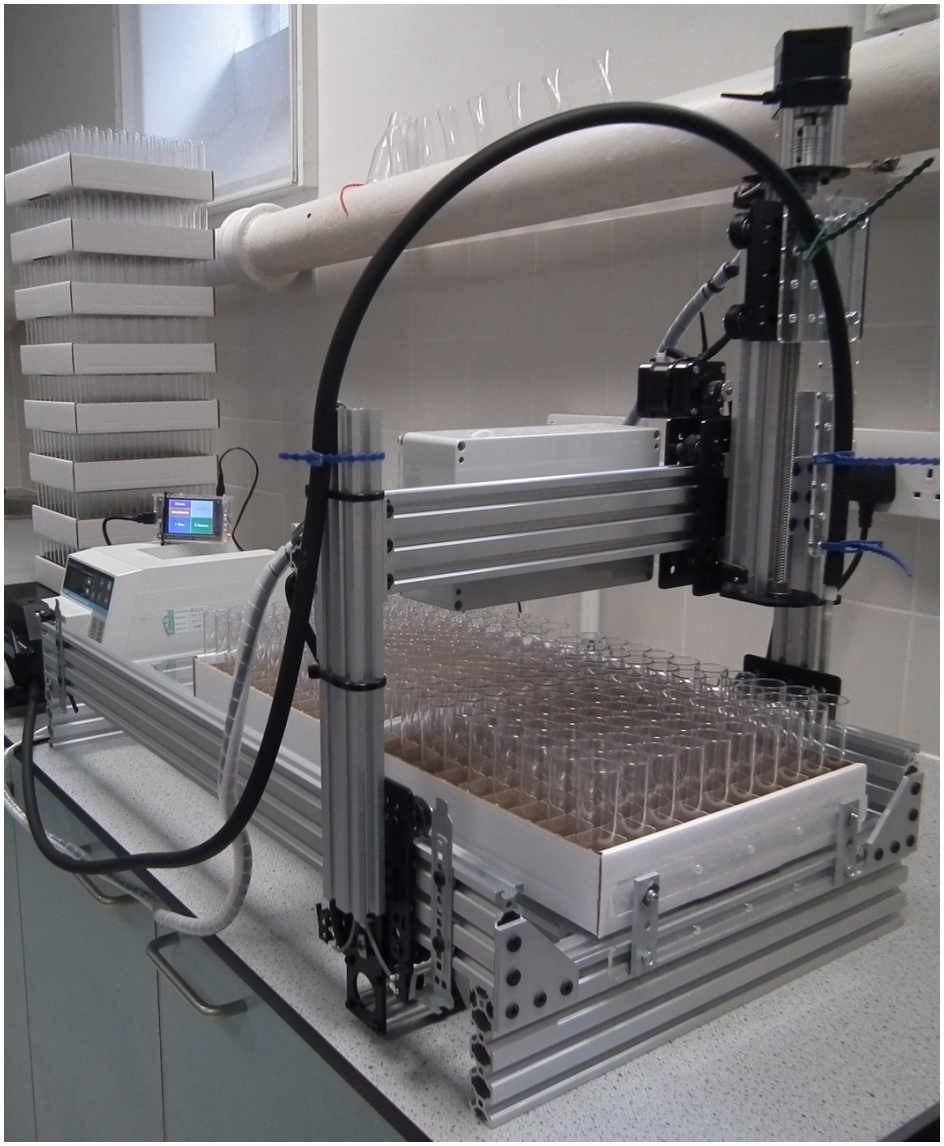

Figure 1.1: Robot

Chapter 2

Introduction

The fruit fly, *Drosophila melanogaster*, is one of the most important model organisms in biological research. Maintaining stocks of fruit flies in the laboratory is labour-intensive. One task which lends itself to automation is the production of the vials of food in which the flies are reared. Fly facilities typically have to generate several thousand vials of fly food each week to sustain their fly stocks. The system presented here combines a cartesian coordinate robot with a peristaltic pump. The design of the robot is based on the Routy CNC Router created by Mark Carew (http://openbuilds.org/builds/routy-cnc-router-v-slot-belt-pinion.101/), and uses belt and pully actuators for the X and Y axes, and a leadscrew actuator for the Z axis. CNC motion and operation of the peristaltic pump are controlled by grbl (https://github.com/gnea/grbl), an open source, embedded, high performance g-code parser. Grbl is written in optimized C and runs directly on an Arduino. A Raspberry Pi is used to generate and stream G-code instructions to Grbl. A touch screen on the Raspberry Pi provides a graphical user interface to the system. This manual explains how to install the required software and operate the robot. Instructions for building the hardware are available on DocuBricks.

A Raspberry Pi is used to generate and stream G-code to the Arduino. A touch screen on the Raspberry Pi provides the user interface; a resistive rather than capacitive touch screen was chosen so that it could be operated by a person wearing gloves.

knitr::include_graphics("images/system.jpg")

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

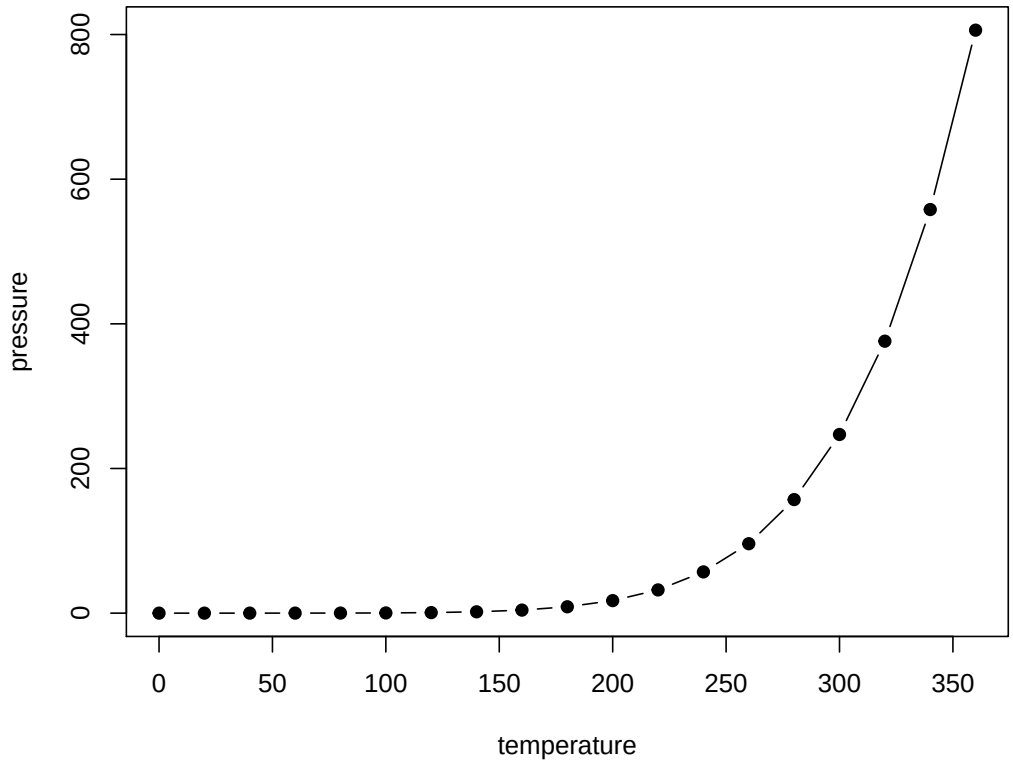

Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Grbl installation and configuration

3.1 Overview

CNC motion control is provided by grbl (https://github.com/gnea/grbl), an open source, embedded, high performance g-code parser. Grbl is written in optimized C and runs directly on an Arduino. This is used in conjunction with the gShield (formerly known as grblshield) which provides the hardware drivers for the stepper motors. Grbl sends out TTL signals on pins A3 and 13 or the Arduino to control coolant flow and spindle direction, respectively. Here these signals are used to remotely control a peristaltic pump.

3.2 Flashing Grbl to Arduino

To flash Grbl to the Arduino you will need a computer with the latest version of the Arduino IDE installed. The following instructions for flashing Grbl to the Arduino are taken from: https://github.com/gnea/grbl/wiki/Compiling-Grbl

NOTE: Before starting, delete prior Grbl library installations from the Arduino IDE. Otherwise, you'll have compiling issues! On a Mac, Arduino libraries are located in `~/Documents/Arduino/libraries/`. On Windows, it's in `My Documents\Arduino\libraries`.

1. Download the Grbl source code.

- Open the following page in your web browser: https://github.com/gnea/grbl
- Click on the `<>Code` Tab
- Click the `Clone or Download` green button on the Grbl home page.
- Click the `Download ZIP`
- Unzip the download and you'll have a folder called `grbl-XXX`, where `XXX` is the release version.

2. Launch the Arduino IDE

- Make sure you are using the most recent version of the Arduino IDE! \begin{figure}