Lapcounter with Raspberry pi

Paul Stahr

January 22, 2022

1 Introduction

This is a testimonial for building a lapcounter for slotcars with the rasperry pi. It contains a guide how to wire the necessary hardware and also a program to cover the main functionality. Any feedback and improvement suggestions are very welcome.

2 Hardware

In my setup I used a raspberry pi 3, but I won't expect any problems using other versions. For checking if cars crossed the starting line it turned out to be most reliable to have fototransistors in the track and some leds above. Another promising attempt to use the slot itself turned out to miss the car occasionally as they hopped over the barrier from time to time. To have a compact device I used the 7" Touchscreen Display which could be directly connected.

2.1 Components

Item	Amount	Article number	Price
Raspberry PI 4 B 2 GB All-In-Bundle	1	RPI 4B 2GB ALLIN	66,50
7" Touchscreen Display	1	RASPBERRY PI 7TD	64,50
NE 555 DIP	slots	NE 555 DIP	0.17
Resistor 470Ω	slots	K-O RD12JN471T52	0.08
Resistor 10k	slots	K-O RD12JN103T52	0.08
Resistor 100k	slots	K-O RD12JN104T52	0.08
Led V 510 Flat Led Rot	slots	V 510	0.10
Phototransistor BPW 42	slots	BPW 42	0.29
LED 3MM GN LED	1	LED 3MM GN	0.07
Bluetooth keyboard		(1)	
MT3608	1		3.29

Table 2.1: Table of components

2.2 Wiring

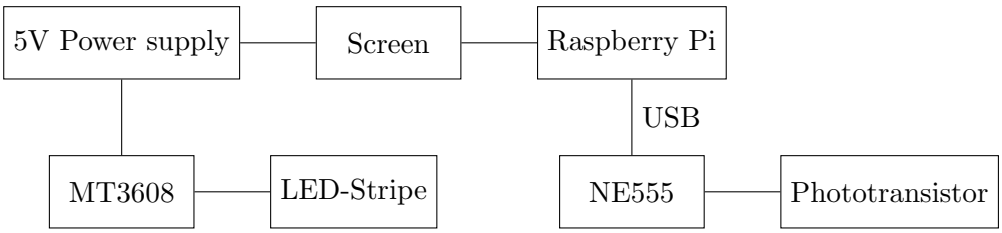

Figure 2.1: High level connection Diagram

To connect the display power, the illumination and the senors we will use the gpio of the raspberry pi. For the connection from the photodiode to the input-circuit I uses usb-connectors. I put the 2.2.1 between the photodiode and the input of the raspberry pi, which pulls down the voltage for a defined amount of time whenever a car crosses.

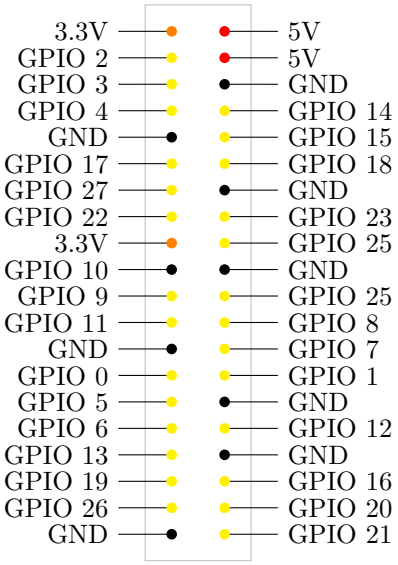

Figure 2.2: Raspberry pi pinlayout

2.2.1 GPIO

The gpio of the raspberry pi provides low level commiunication with electric circuits. Here is a complete diagram of all connectors. Please notice that the 3.3V are provided by the integrated voltage regulator and only allow very small loads.

2.2.2 NE-555

The timer NE-555 is maybe the most often produced integrated circuit in history. It allows us to work as a delay, extending the signal of a led covered for a short amount of time to trigger a signal which lasts over a second. As the raspberry pi supports interrupts it should be possible to directly connect the phototransistor to its gpio input.

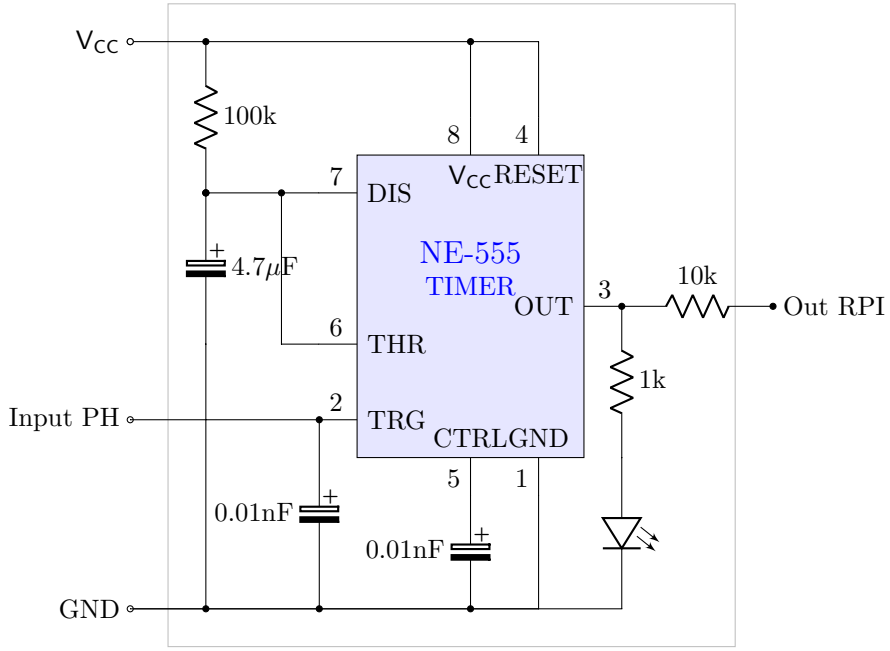

Figure 2.3: Delay circuit

2.2.3 Connectors

As you have to put the phototransistors into the racetrack you might not want to have fixed cables running all the way up to the GPIO of the Raspberry pi. You can use any connector you like, but I found usb very handy because this allowed me to just solder Type-A and Mini-A connectors on all boards and use ordinary cables for the connections in between. I tried to keep as close to the original standart as possible by using the outer lanes for power supply and the inner one for data transmition.

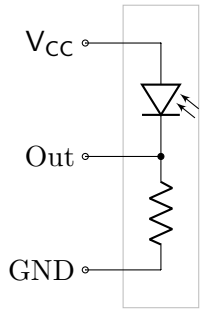

Figure 2.4: Photodiode

2.2.4 Illumination

For illumination it was conventient to use led-stripe. The general problem here is, that most of them work with 12V. Also you might want to have an adjustable brightness. For these kind of tasks a MT3608 Step Up converter is a cheap and handy solution. Just connect it to the 5V supply and the led-stripe to its output.

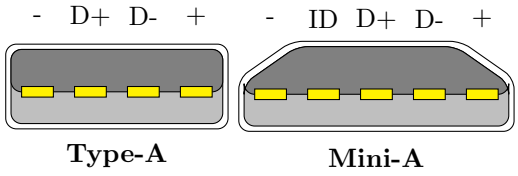

Figure 2.5: USB-Pins

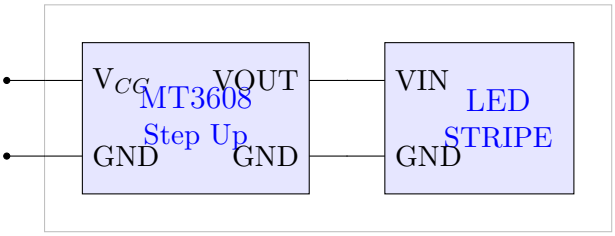

Figure 2.6: Power supply for Led-stripe

3 Software

3.1 Compiling

The program uses Allegro to show the results. In theory it should be possible to use it on any machine, which supports linux and opengl, I tested it on a x86 platform and on the raspberry pi. If you are using debian, please install the following packages:

Listing 3.1: Required packages

```
sudo aptitude install build-essential g++ liballegro5-dev
   freeglut3-dev libboost-thread-dev libboost-system-dev
```

Then build with either

Listing 3.2: Compile

```
mkdir -p build; make -j 4 program
```

Or if you are using the rasperry-pi

Listing 3.3: Compile Pi

```
mkdir -p build; make -j 4 programpi
```

This will create a binary which you can start with

Listing 3.4: Run Lapcounter

```
./programpi
```

3.2 Controls

There are 4 inputs at which the program listenens to which can generate events which will be handled by the program.

Input	Menu Control	Car Control
Light Barier	no	yes
Mouse	yes	no
Keyboard	yes	yes
Html-Webside	yes	yes
Gamepad	yes	no

3.3 Fast Race

The common mode is the Fast Race. You can select up to 4 players and an arbitrary amount of laps.

3.4 Tournee Planer

The program contains a simple algorithm to calculate plans for tournees. You can adjust the numbers of players, slots, races and equaly hard slots. Keep in mind, that mathematically not all combinations of input can work.

3.5 Settings

3.6 Network access

Connecting to the server will return you a Html-document containing the most important data.