

RasPiTrigger – Water drop photos with the Raspberry Pi

Herbert Kopp

Water drop photography

When photographing water drop sculptures, exact timing is very important:

- The size and release time of the drops must be precisely controlled by solenoid valves.
- The timing of the exposure (i.e., the triggering of the flashes) must be set to within milliseconds.
Furthermore, the flashes operate at minimal power and therefore have a very short flash duration (< 100 microseconds) to freeze the movement.

In addition, there are more or less complex requirements for the construction of a frame for mounting the equipment, the camera, the lens and the recording technology, which are described in detail in [1].

There are different solutions for exact, reproducible timing of processes:

- Hardware-based devices whose parameters are adjusted via controllers. e.g. [2]
- Controllers connected to a PC. A program runs on the computer to specify the parameters. These are then transferred to the controller, which controls the processes. e.g., [3]
- Arduino and Raspberry Pi-based DIY solutions that receive control data from a PC. e.g. [4]

The RasPiTrigger

The RasPiTrigger differs from the above solutions in that all essential functionalities are implemented on a Raspberry Pi, namely:

- the setting of parameters via a graphical user interface and
- the timing of all processes (camera shutter, flash units and solenoid valves).

The connection to the connected devices is established by a junction box, which electrically separates them from the GPIO ports of the Raspberry Pi and switches the operating voltage of the solenoid valves.



Figure 1 :

Water drop photos taken with the RasPiTrigger

The recording facility

An attempt was made to build the recording facility with minimal effort, mainly using existing material:

- The table stand of an old Durst magnifying apparatus was converted to create the solenoid valve holder: All that was needed was to attach a horizontal mounting arm for the solenoid valves.
- A plastic box from the hardware store is used as a water basin into which the drops fall.
- Three system flashes are used for lighting, which are triggered via the center contact in the flash base or via a PC sync socket.
- The camera is an APS-C camera with a macro lens.



Figure 2 : The recording device for water drop photography

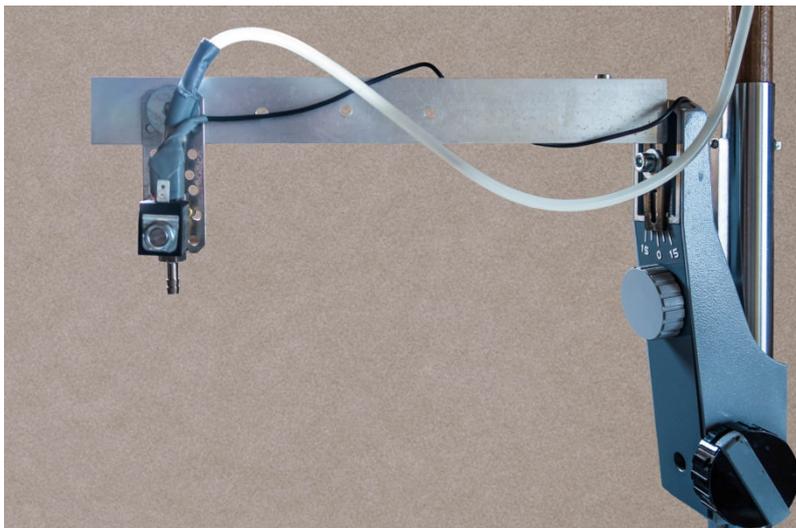


Figure 3 : The solenoid valve carrier

The connection box

The connection box is connected to the Raspberry Pi via a 40-pin GPIO cable. It contains:

- Optocoupler for electrically isolating the Raspberry Pi's GPIO ports from the connected devices.
- Switching transistors for connecting up to three solenoid valves.
- a 12V power supply for the solenoid valves.



Figure 4 : Raspberry Pi 4B+ and connection box

Coupling camera and flash units

Figure 5 shows the coupling circuit for the camera and flash units: Isolation from the Raspberry Pi's GPIO ports is achieved using SFH610A optocouplers, and an L-53SGD LED indicates that a switch is closed. Figure 6 shows the corresponding soldering diagram.

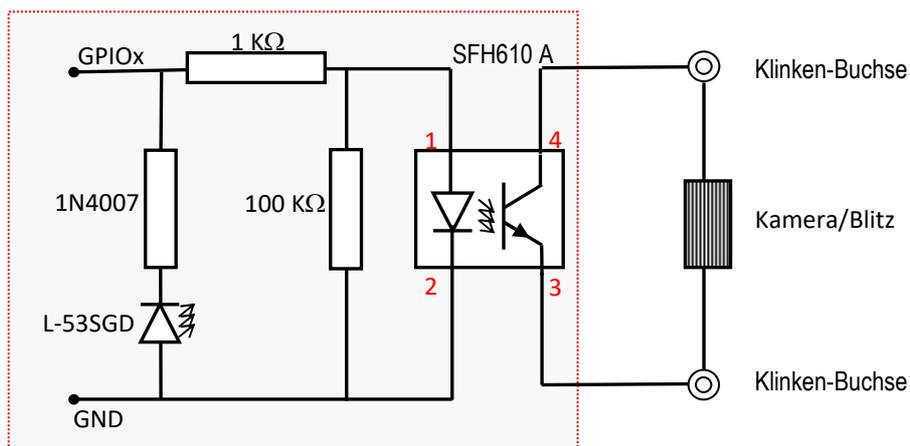


Figure 5 : Circuit diagram for the optocouplers

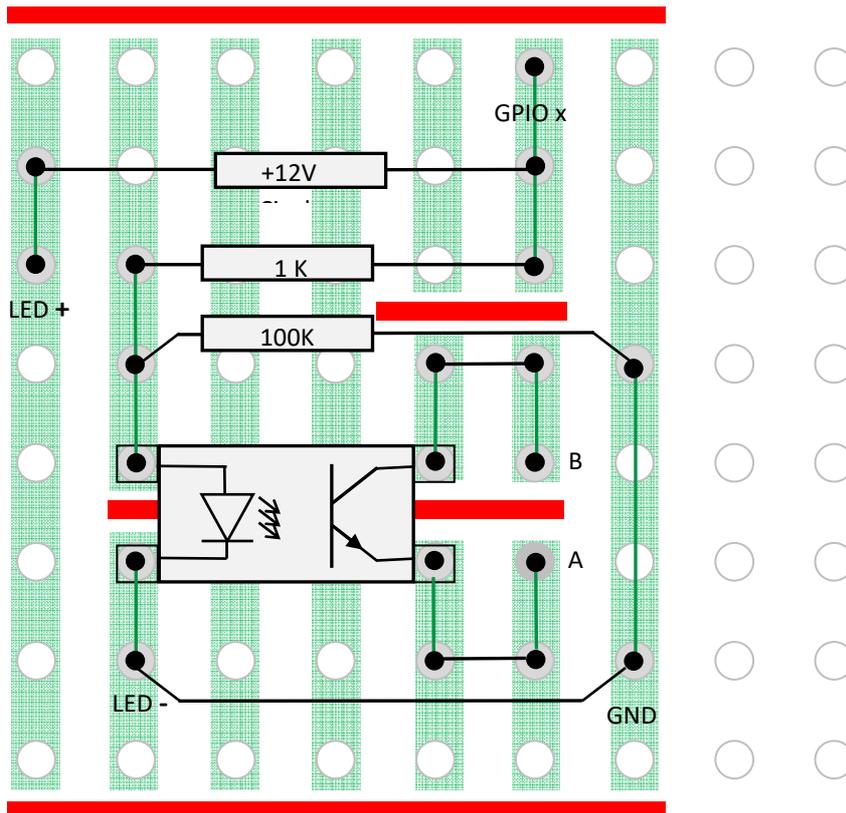


Figure 6 : Soldering plan for the optocouplers

Coupling of the solenoid valves

Figure 7 shows the coupling circuit for the solenoid valves: Like the other connections, they are separated from the GPIO ports by optocouplers. The supply voltage for the solenoid valves is switched by a TIP120 Darlington transistor. The soldering diagram for the left part of the circuit is shown in Figure 6, and that for the switching transistor is shown in Figure 8.

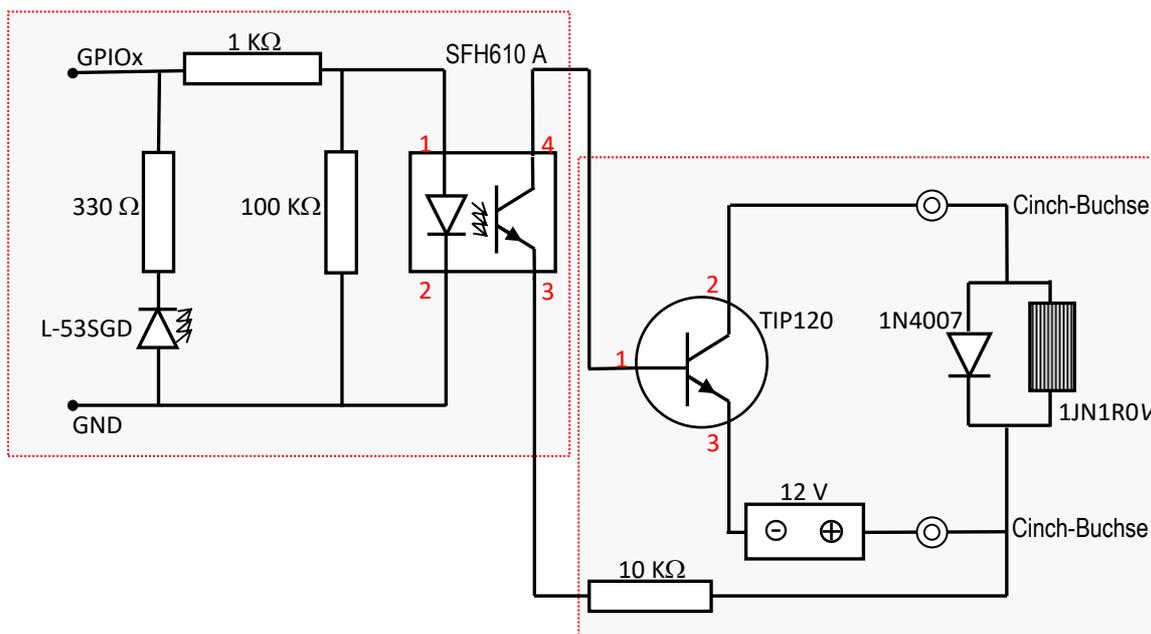


Figure 7 : Circuit diagram for the coupling of the solenoid valves

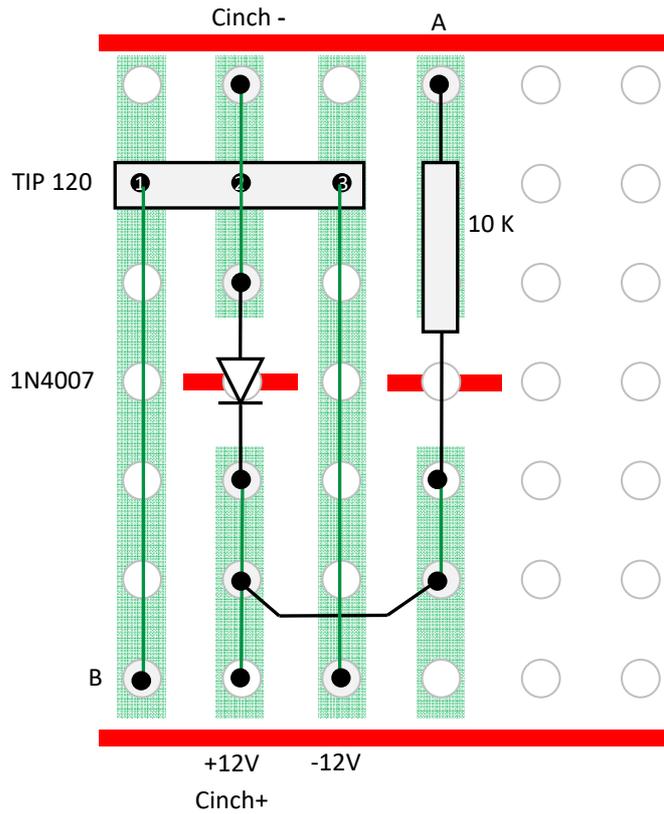


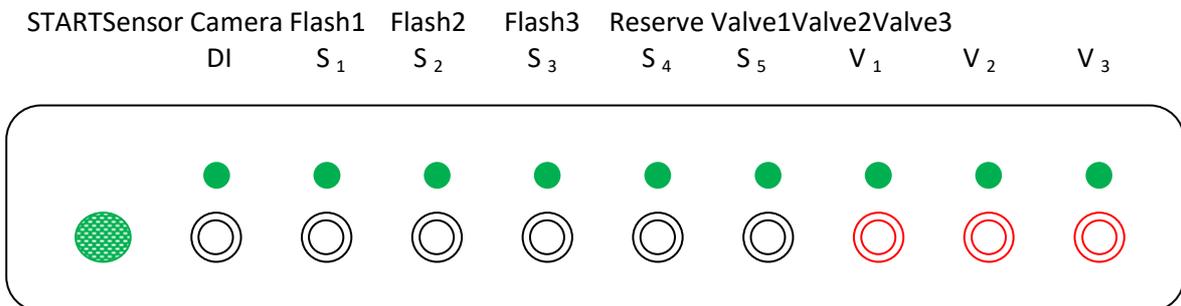
Figure 8 : Soldering plan for the Darlington switches

Complete circuit design

All assemblies were mounted on a Euroboard with a solder strip grid. Figure 9 shows the overall layout of the board.

Connections on the top of the case:

In addition to the connection sockets for camera, flashes and solenoid valves, two unused sockets (sensor, reserve) were added.



- START: Button, debounced by the GPIO software
- DI: 3.5mm jack socket: Digital Input (e.g. for light barrier)
- S_i: 3.5mm jack sockets: switched by optocouplers
- V_i: RCA sockets, 12V outputs, switched by optocouplers + TIP120

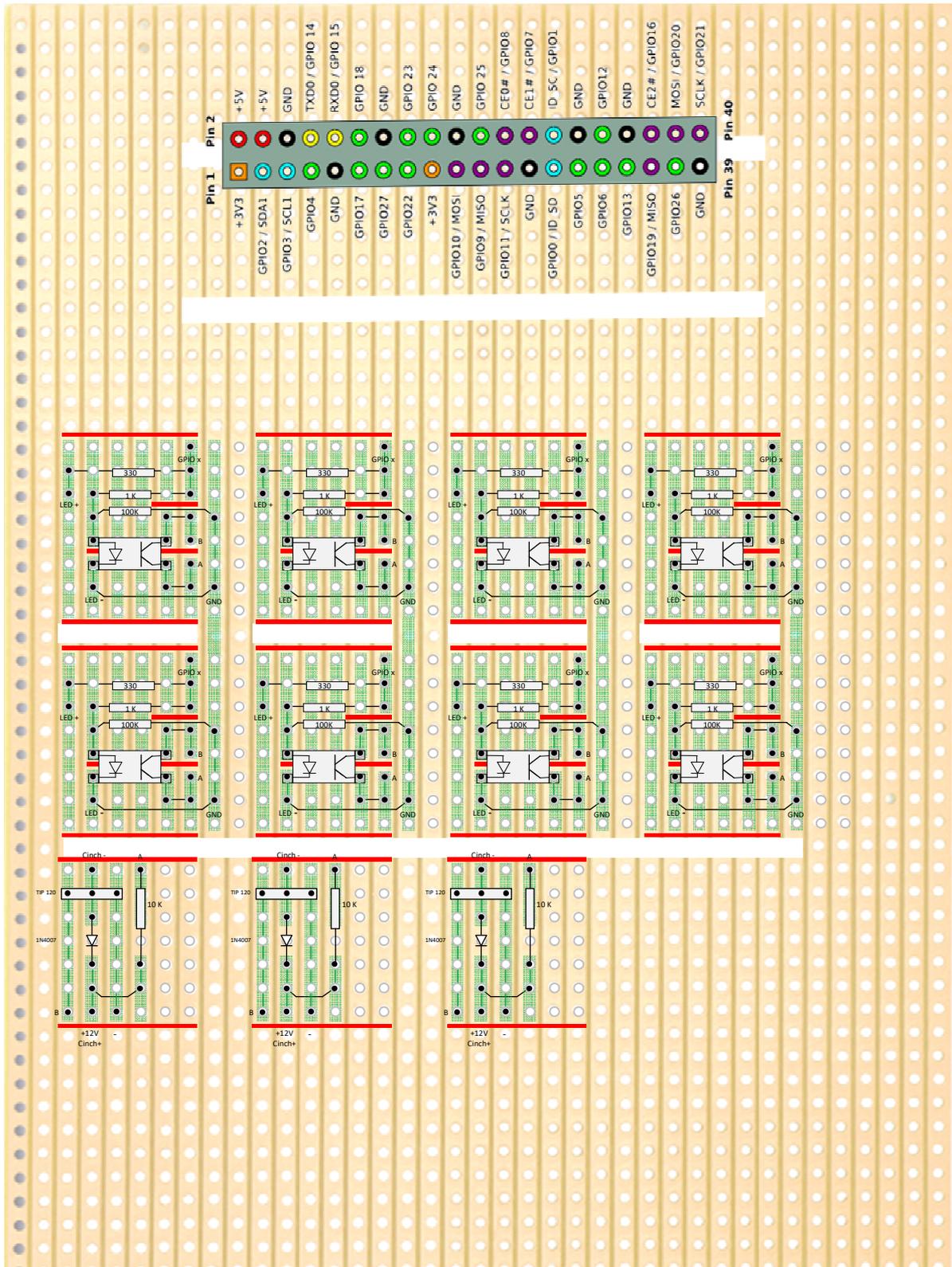


Figure 9 : Overall layout of the board

Additional connections:

- Power supply socket: 12 V/DC - 1500 mA - 18 W
- 2x2 0-pin header + GPIO ribbon cable 2x20 wires

The RasPiTrigger software

The RasPiTrigger software was developed in *JavaFX* using the *Eclipse development environment* . It provides a graphical user interface for setting parameters, generating and executing a command sequence for controlling the connected devices when the START button or START key is pressed.

Additional features:

- A diagram at the bottom of the window shows the timing of the configured processes. As the parameters are changed, the effect is displayed synchronously in the diagram.
- The configured parameter sets can be saved in project files and loaded again from there.
- Each project can be supplemented with a comment text.
- Additional functions allow flushing of the solenoid valves and resetting of all parameters.

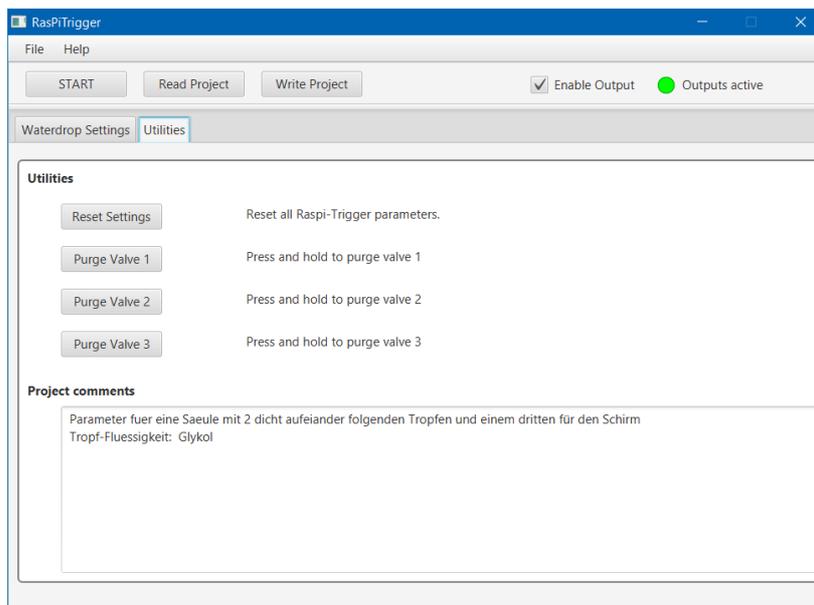
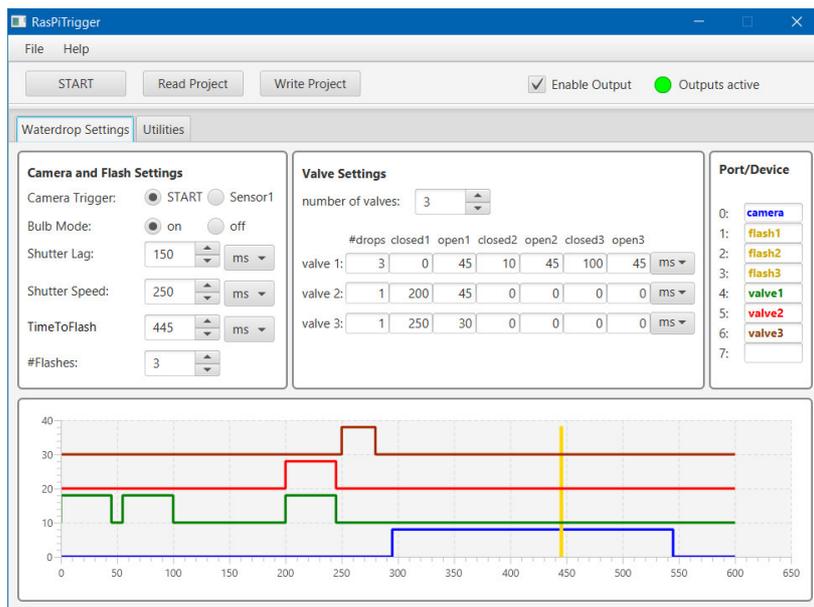


Figure 10 : GUI of the RasPiTrigger control program

Installing the RasPiTrigger software

The program was developed using *Eclipse* and Oracle's *Java JDK 8u33* . This version of Java is preinstalled on the Raspberry Pi B3, so RasPiTrigger runs on it without any additional setup:

- Copy the *RasPiTrigger.jar* application and the *start.sh* script to a directory on the Raspberry Pi.

The *start.sh* script is then made executable with

```
chmod +x start.sh
```

- After that you can start the RasPiTrigger with the command:

```
start.sh
```

Since the Raspberry Pi B4 comes pre-installed with the free Java version *OpenJDK* , which doesn't support JavaFX, you'll need to install *JDK 8u33* . A detailed description of how to do this can be provided.

RasPiTrigger.jar application can also be started under Windows with a double-click if the Java Runtime version *jre-8u231* from Oracle is installed. It is fully functional with the limitation that all input/output to the (nonexistent) GPIO interface is blocked.

References

[1] D. Nimmervoll: High-speed Photography, mitp-Verlag, 2018

[2] **drop-timer** , manufacturer: eltima,
<https://eltima.de/tropfenfotografie.html>

[3] **STOPShot Studio** , manufacturer: COGNISYS,
<https://www.cognisys-inc.com/store/stopshotstudio-wdk.html>

[4] **dropshot** , Thomas Reimann,
<http://www.thomas-reimann.com/2015/04/tropfotografie-mit-dem-raspberry-pi/>

RasPiTrigger – Wassertropfen-Fotos mit dem Raspberry Pi

Herbert Kopp

Wassertropfen-Fotographie

Beim Fotografieren von Wassertropfen-Skulpturen ist das exakte Timing sehr wichtig:

- Größe und Auslösezeitpunkt der Tropfen müssen exakt durch Magnetventile gesteuert werden.
- Der Aufnahmezeitpunkt (d.h. die Auslösung der Blitze) muss auf Millisekunden genau eingestellt werden. Darüber hinaus arbeiten die Blitze mit minimaler Leistung und daher sehr kurzer Leuchtdauer (< 100 Mikrosekunden), um die Bewegung einzufrieren.

Hinzu kommen mehr oder weniger aufwendige Anforderungen an den Aufbau eines Gestells zur Montage des Equipments, die Kamera, das Objektiv und die Aufnahmetechnik, die in [1] detailliert dargestellt werden.

Für ein exaktes, reproduzierbares Timing der Abläufe gibt es unterschiedliche Lösungen:

- Hardware-basierte Geräte, deren Parameter über Regler eingestellt werden. z.B. [2]
- Controller, die an einen PC angeschlossen sind. Auf dem Rechner läuft ein Programm zur Vorgabe der Parameter. Diese werden an den Controller übergeben, der die Abläufe steuert. z.B. [3]
- Arduino- und Raspberry Pi-basierte DIY-Lösungen, die von einem PC aus mit Steuerdaten versorgt werden. z.B. [4]

Der RasPiTrigger

Der RasPiTrigger unterscheidet sich von den obigen Lösungen dadurch, dass alle wesentlichen Funktionalitäten auf einem Raspberry Pi implementiert sind, nämlich:

- die Einstellung der Parameter über eine graphische Benutzer-Oberfläche und
- die Zeitsteuerung aller Abläufe (Kamera-Auslöser, Blitzgeräte und Magnetventile).

Die Verbindung zu den angeschlossenen Geräten stellt eine Anschluss-Box her, welche sie elektrisch von den GPIO-Ports des Raspberry Pi trennt und die Betriebsspannung der Magnetventile schaltet.



Abbildung 1:

Wassertropfen-Fotos, aufgenommen mit dem RasPiTrigger

Die Aufnahme-Einrichtung

Es wurde versucht, die Aufnahme-Einrichtung mit minimalem Aufwand, hauptsächlich mit vorhandenem Material zu bauen:

- Für den Magnetventil-Träger wurde das Tisch-Stativ eines alten Durst-Vergrößerungs-Apparats umfunktioniert: Es musste dazu nur ein horizontaler Aufnahme-Arm für die Magnet-Ventile angebracht werden.
- Eine Plastik-Box aus dem Baumarkt wird als Wasserbecken verwendet, in das die Tropfen fallen.
- Zur Beleuchtung werden drei Systemblitze eingesetzt, welche über den Mitten-Kontakt im Blitzfuß bzw. über eine PC Sync-Buchse ausgelöst werden.
- Die Kamera ist eine APS-C -Kamera mit Makro-Objektiv.



Abbildung 2: Die Aufnahme-Einrichtung für Wassertropfen-Fotografie

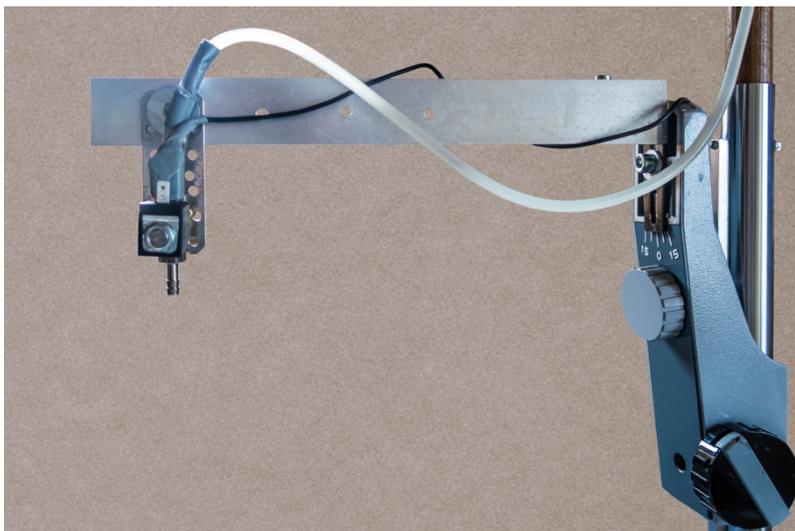


Abbildung 3: Der Magnetventil-Träger

Die Anschluss-Box

Die Anschluss-Box ist über ein 40 Pin GPIO-Kabel mit dem Raspberry Pi verbunden. Sie enthält:

- Optokoppler zur elektrischen Trennung der GPIO-Ports des Raspberry Pi von den angeschlossenen Geräten.
- Schalttransistoren zum Anschluss von bis zu drei Magnetventilen.
- eine 12V - Spannungsversorgung für die Magnetventile.



Abbildung 4: Raspberry Pi 4B+ und Anschlussbox

Ankopplung von Kamera und Blitzgeräten

Abbildung 5 zeigt die Ankopplungs-Schaltung für Kamera und Blitzgeräte: Die Trennung von den GPIO-Ports des Raspberry Pi wird durch Optokoppler vom Typ SFH610A erreicht, außerdem zeigt eine Leuchtdiode vom Typ L-53SGD an, dass ein Schalter geschlossen ist. Abbildung 6 zeigt den zugehörigen Lötplan.

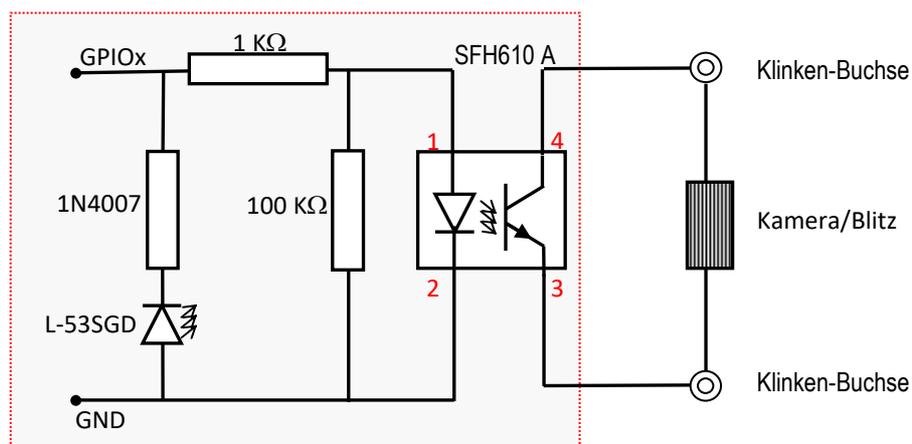


Abbildung 5: Schaltplan für die Optokoppler

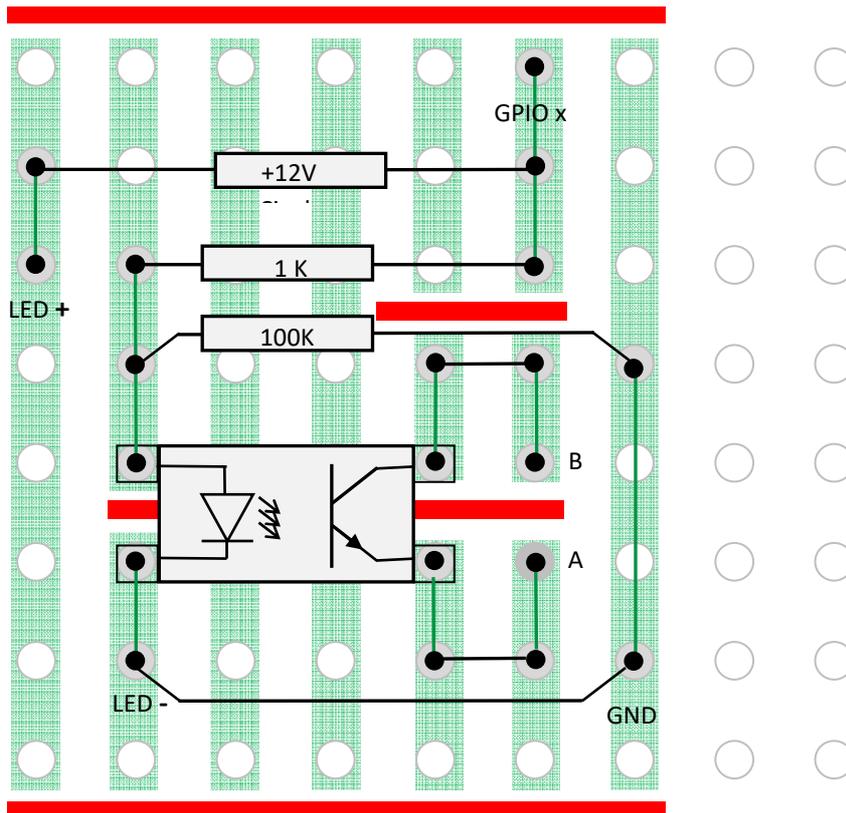


Abbildung 6: Lötplan für die Optokoppler

Ankopplung der Magnetventile

Abbildung 7 zeigt die Ankopplungs-Schaltung der Magnetventile: Sie werden wie die übrigen Anschlüsse durch Optokoppler von den GPIO-Ports getrennt. Die Versorgungsspannung der Magnetventile wird durch einen Darlington-Transistor vom Typ TIP120 geschaltet. Der Lötplan für den linken Teil der Schaltung ist in Abbildung 6 und der für den Schaltransistor in Abbildung 8 dargestellt

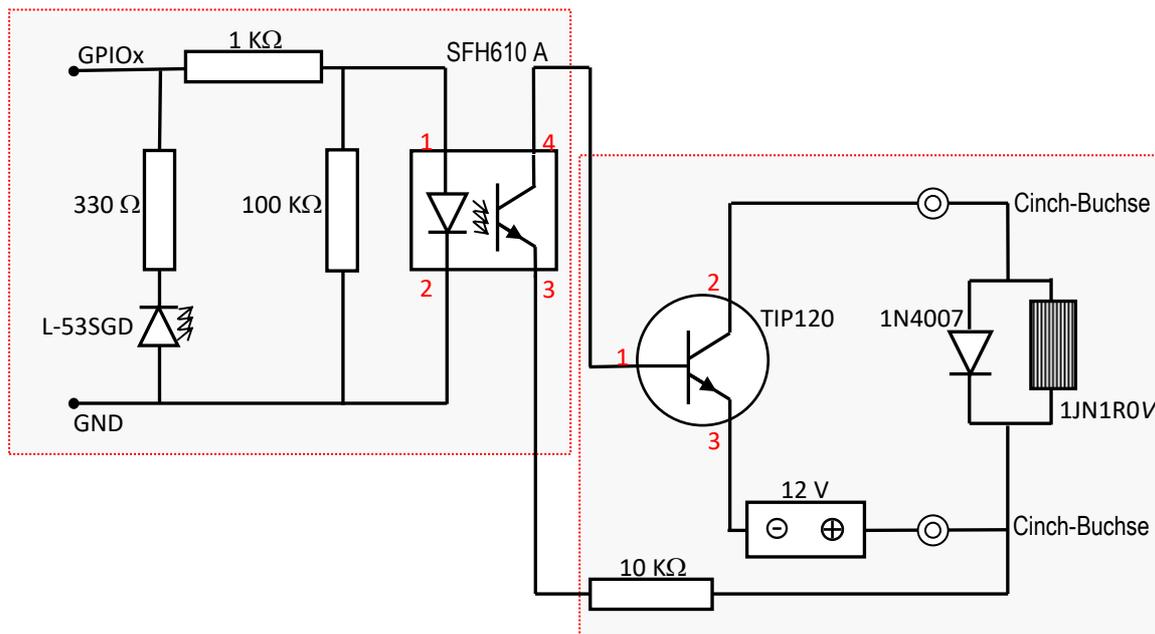


Abbildung 7: Schaltplan für die Ankopplung der Magnet-Ventile

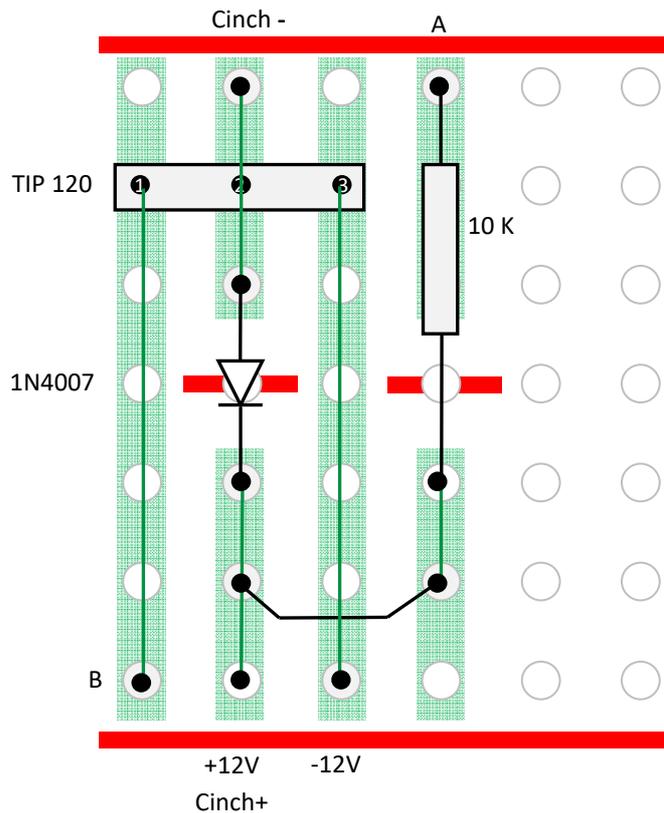


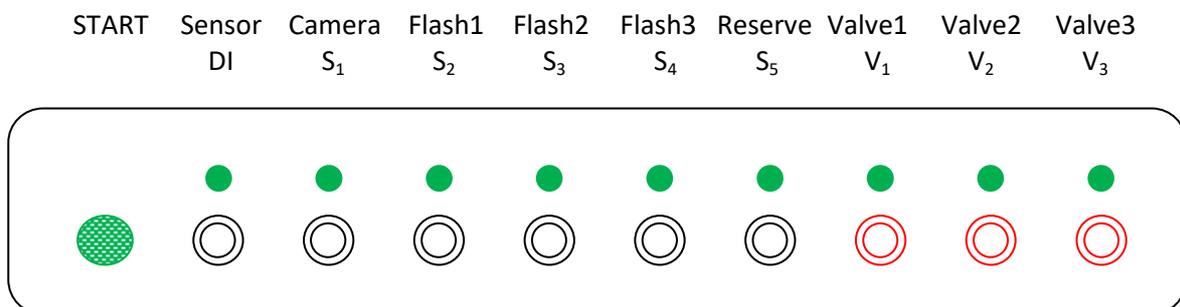
Abbildung 8: Lötplan für die Darlington-Schalter

Vollständiger Schaltungsaufbau

Alle Baugruppen wurden auf einer Euro-Platine mit Lötstreifenraster aufgebaut. Abbildung 9 zeigt das Gesamt-Layout der Platine.

Anschlüsse auf der Gehäuse-Oberseite:

Zusätzlich zu den Anschluss-Buchsen für Kamera, Blitze und Magnetventile wurden noch zwei noch nicht benutzte Buchsen (Sensor, Reserve).



- START: Taster, entprellt durch die GPIO-Software
- DI: 3,5mm Klinkenbuchse: Digital Input (z.B. für Lichtschranke)
- S_i: 3,5mm Klinkenbuchsen: geschaltet durch Optokoppler
- V_i: Cinch-Buchsen, 12V-Ausgänge, geschaltet durch Optokoppler + TIP120

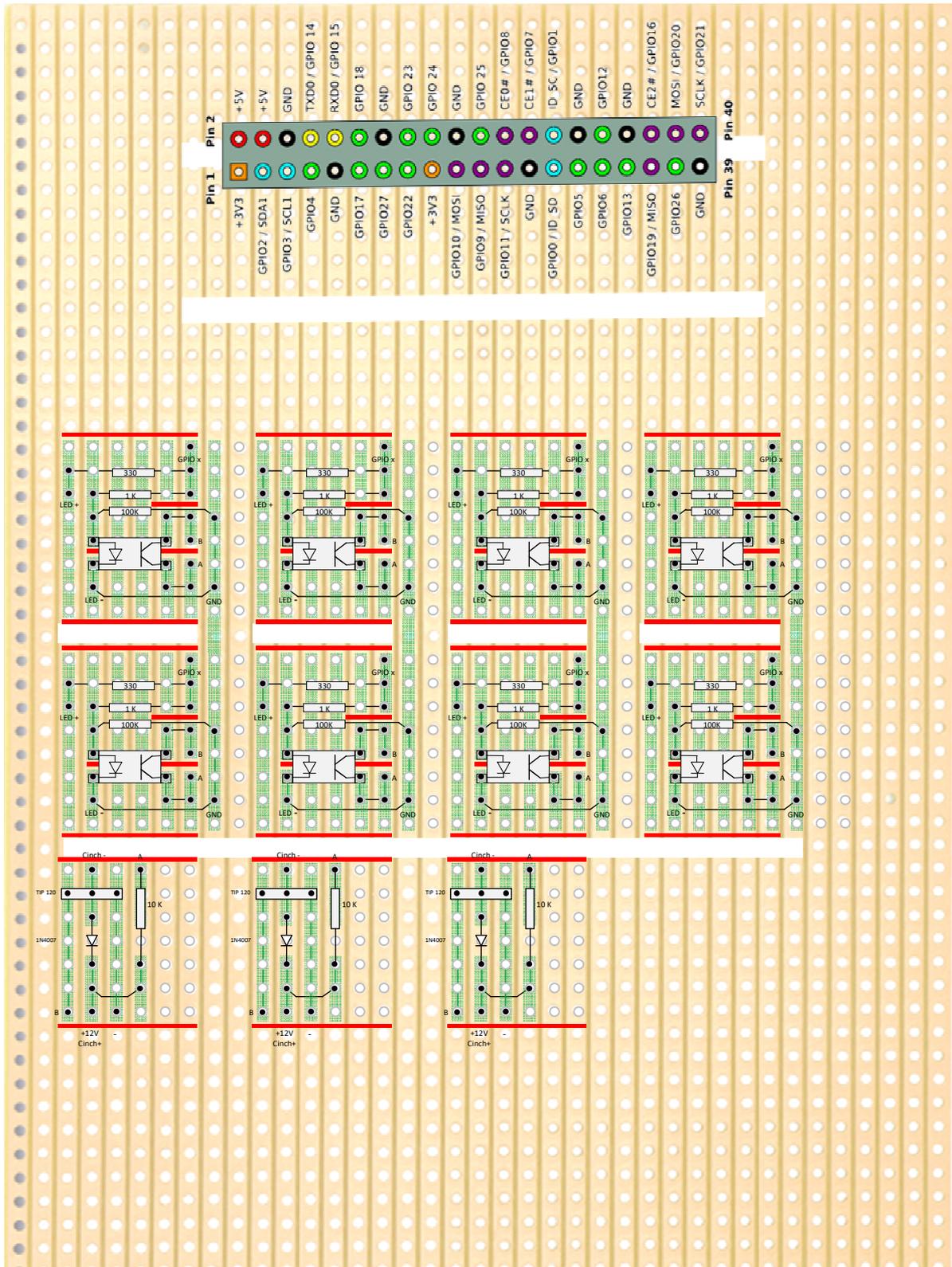


Abbildung 9: Gesamt-Layout der Platine

Weitere Anschlüsse:

- Buchse für die Spannungsversorgung: 12 V/DC - 1500 mA - 18 W
- 2x20-polige Stiftleiste + GPIO- Band-Kabel 2x20 Adern

Die RasPiTrigger-Software

Die RasPiTrigger-Software wurde in *JavaFX* mit der Entwicklungsumgebung *Eclipse* entwickelt. Sie stellt eine grafische Benutzer-Oberfläche zum Einstellen der Parameter bereit, erzeugt daraus beim Betätigen des START-Button oder der START-Taste eine Befehlsfolge zur Steuerung der angeschlossenen Geräte und führt diese aus. Weitere Merkmale:

- Im unteren Bereich des Fensters zeigt ein Diagramm das Timing der eingestellten Abläufe. Beim Verändern der Parameter wird die Auswirkung synchron im Diagramm angezeigt.
- Die eingestellten Parametersätze können in Projekt-Dateien gespeichert und von dort aus auch wieder geladen werden.
- Jedes Projekt kann mit einem Kommentartext ergänzt werden.
- Weitere Funktionen ermöglichen das Durchspülen der Magnetventile und das Zurücksetzen aller Parameter.

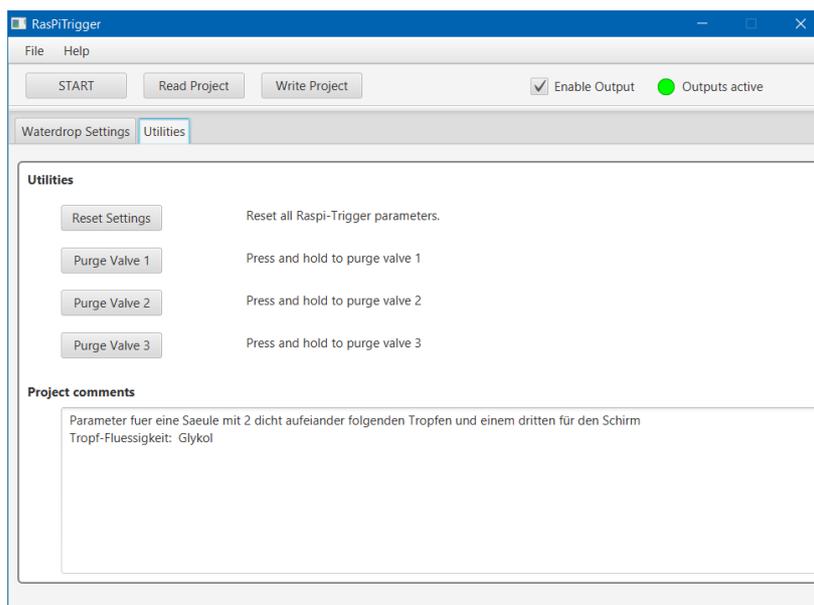
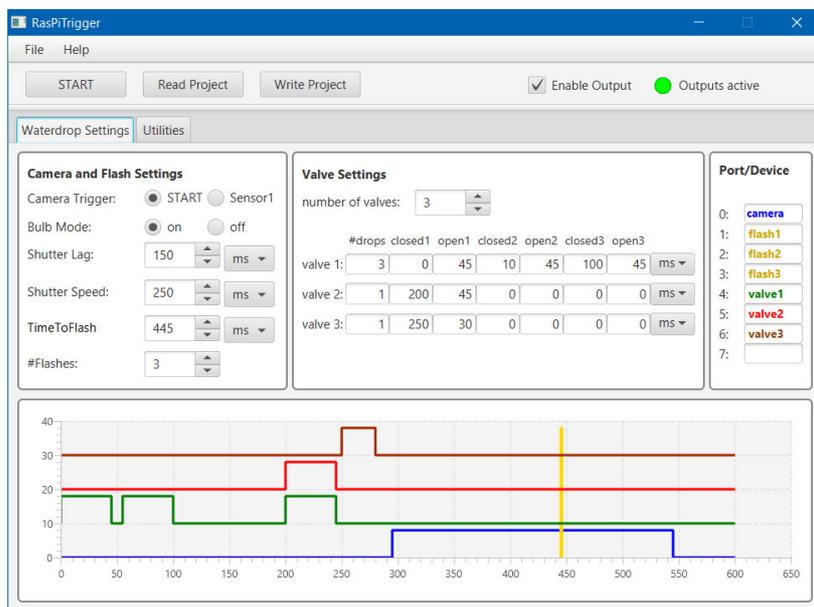


Abbildung 10: GUI des RasPiTrigger-Steuerprogramms

Installation der RasPiTrigger-Software

Das Programm wurde mit *Eclipse* dem *Java JDK 8u33* von Oracle entwickelt. Diese Java-Version ist auf dem Raspberry Pi B3 vorinstalliert, so dass der RasPiTrigger dort ohne weitere Maßnahmen läuft:

- Die Anwendung *RasPiTrigger.jar* und das Start-Skript *start.sh* sind in ein Verzeichnis auf dem Raspberry Pi zu kopieren. Das Skript *start.sh* wird danach ausführbar gemacht mit
`chmod +x start.sh`
- Danach kann man den RasPiTrigger starten mit dem Kommando:
`start.sh`

Da auf dem Raspberry Pi B4 die freie Java-Version *OpenJDK* vorinstalliert ist, die JavaFX nicht unterstützt, muss man dort das *JDK 8u33* nachinstallieren. Eine detaillierte Beschreibung dazu kann zur Verfügung gestellt werden.

Die Anwendung *RasPiTrigger.jar* kann auch unter Windows mit einem Doppelklick gestartet werden, wenn die Java-Runtime -Version *jre-8u231* von Oracle installiert ist. Sie ist voll funktionsfähig mit der Einschränkung, dass alle Ein-/Ausgaben an die (nicht vorhandene GPIO-Schnittstelle) blockiert sind.

Referenzen

- [1] D. Nimmervoll: Highspeed-Fotografie, mitp-Verlag, 2018
- [2] **drop-timer** , Hersteller: eltima,
<https://eltima.de/tropfenfotografie.html>
- [3] **STOPShot Studio**, Hersteller: COGNISYS,
<https://www.cognisys-inc.com/store/stopshotstudio-wdk.html>
- [4] **dropshot**, Thomas Reimann,
<http://www.thomas-reimann.com/2015/04/tropfenfotografie-mit-dem-raspberry-pi/>