Developing a High-Throughput System for monitoring Zebrafish Behaviour

Uruj Fatima Sarwar
42058
mailto:usarwarh@abo.fi

Master's Thesis

Åbo Akademi University
Department of Biosciences
Faculty of Science and Engineering
05.06.2019

Master´s degree in Biomedical Imaging
Specialisation Theme: Neuronal Signalling Pathways

Credits: 40 ECTS

Supervisors:
1: Michael Courtney, Ph.D.

Examiners:
1: Michael Courtney, Ph.D.

2: Ilkka Paatero, Ph.D.

ÅBO AKADEMI UNIVERSITY
Department of Biosciences
Faculty of Science and Engineering

URUJ FATIMA SARWAR

Developing a High-Throughput System
for monitoring Zebrafish Behaviour

Master´s thesis, 79 pp., II Appendices
Biomedical Imaging
June 2019

The aim of this thesis project was to develop a high-throughput (HT) system for recording the behaviour of zebrafish larvae in response to light stimuli. This was achieved by using a compatible camera, Raspberry Pi Camera (PiCam V2.1) for which drivers exist (PiBoard Microcontroller RPi Zero W V1.1), for imaging zebrafish. A planar white LED copy board A4 light source was used for providing light stimuli. The PiCam was controlled by the PiBoard. Another microcontroller that runs a real-time operating system, Arduino Uno R3, was used to control the light source and synchronise it with the camera controlled via the PiBoard. Zebrafish behaviour was recorded in the form of videos. These videos were converted into image stacks that were further processed with an ImageJ-based protocol to identify the percentage of how many zebrafish moved during certain light protocols. Two behaviours were considered for this study, startle response and prepulse inhibition (PPI). A startle response refers to a defensive response to a sudden stimulus in animals and humans. However, in the presence of a weaker stimulus followed by a startling stimulus, the startle response of the organism is reduced, and this is called PPI. The results showed that larval zebrafish exhibited startle response at 4-5 days post fertilisation (dpf) and this was detectable with the developed HT System. However, there was insufficient data to conclude whether they exhibited PPI or not based on preliminary results using the same system.

KEYWORDS: High-Throughput (HT) System, Zebrafish Larvae, Light Stimulus, Startle Response, Prepulse Inhibition (PPI), Arduino Microcontroller, Raspberry Pi (RPi) Microcontroller

LIST OF ABBREVIATIONS

PPI	Prepulse Inhibition
dpf	Days Post-Fertilisation
HT	High-Throughput
HTS	High-Throughput Screening
HCS	High-Content Screening
PWM	Pulse Width Modulation
RPi	Raspberry Pi
IP	Internet Protocol
WLAN	Wireless LAN
V	Voltage (Volts)
IDE	Integrated Development Environment
SHIRPA	SmithKline Beecham, Harwell, Imperial College, Royal London Hospital, Phenotype Assessment
ROI	Region of Interest

CONTENTS

1 INTRODUCTION AND LITERATURE REVIEW

Behavioural experimentation and analysis in zebrafish behaviour contribute a useful readout for neurotoxicity (Legradi, El Abdellaoui, van Pomeren and Legler 2015). This is because zebrafish (Scientific name for Zebrafish is *Danio rerio*) provides a model that has the ability to detect highly complex and different processes and mechanisms within a living organism in a high-throughput manner. The above study (Legradi El Abdellaoui, van Pomeren and Legler, 2015) had focused on neurotoxicity testing and the importance of understanding neuronal disorders using zebrafish.

For many decades now, zebrafish has been used as an animal vertebrate model in biomedical research studies for studying neurological diseases and cognition associated to age (Sarasamma et al., 2017). Zebrafish models provide a range of advantages in comparison with other vertebrate models, such as their availability, shorter life span, fertility, reproducibility, easy analysis of cellular activities, and low-cost for maintenance in a laboratory (Adams and Kafaligonul, 2018). Despite the body length of younger zebrafish being 3-5 mm, the entire body itself is also visible with imaging methods (Parichy et al., 2009). This helps in developing a better user perspective of the animal model, which is useful for this thesis project.

The experimental setup of HT methods are differently developed and used based on requirements of behavioural analysis. Research studies (Liu et al., 2016; Rihel et al., 2010) have presented a diverse range of behavioural results. Problems within the handling of experimental setups and data collection have also been experienced. These issues are usually ambient light exposure, recording video data and temperature maintenance for the animal model (Kokel et al., 2010). This provides the opportunity to make HT methods more accurate by removing the mentioned problems. The aim of this thesis is to develop an HT system for analysing zebrafish behavioural responses when the animal model interacts with photic stimuli.

Two behavioural responses are considered, Startle Response and Prepulse Inhibition (PPI). Research studies have conducted startle response and PPI trials on different animal models as well, such as mice and zebrafish (Kokel and Peterson, 2011).

1.1 Startle Response, Prepulse Inhibition (PPI) and Neurological Disorders

The startle response is an unconscious reaction to an intense stimulus (pulse) in animals and humans. The response is a defensive reflex to any type of sudden stimulus, light, noise, pressure, and temperature. The sudden stimulus causes this reflex action with respect to interruption, defence, and escape. Disruptions in startle responses are regarded as symptoms for neurological disorders. This is one major reason for studying startle response in humans and animal models (Quevedo et al., 2010).

Prepulse Inhibition (PPI) is a neurological phenomenon, and also a behavioural assay, in which a weaker stimulus (prepulse) can inhibit the startle response. When the organism receives the prepulse that is followed by a startling pulse, the startle response is suppressed (Braff, Geyer and Swerdlow, 2001). Research studies (Ioannidis et al., 2014; Rihel et al., 2010) have focused on disruption of PPI as a behavioural method for neurological disorders, for studying schizophrenia.

Schizophrenia in a person is mainly characterised as a psychological disorder, depicting abnormal behaviour, having an effect on the way a person would think, feel and behave, overall, in addition to being unable to understand reality. However, schizophrenia as a neurological disorder has not been properly studied, based on its neurodevelopment (Zhou et al., 2014). PPI is reduced in schizophrenic patients. This is experimented on animal models and clinical trials that are conducted to test antipsychotic drugs.

The similarities between humans and adult zebrafish contribute to considering zebrafish as a good animal model for behavioural analysis, such as a pair of eyes, mouth, pancreas, kidneys and heart. In addition, most of the genes and essential cellular pathways that are important for growing these organs are similar. This is the reason why if schizophrenia-like disorders cause changes in the human body, these changes can be modelled within zebrafish to a certain extent even though not perfectly (Howe et al., 2013). The study by Bergeron et al., (2015) utilised a novel genetic screen in larval zebrafish for identifying specific neuronal identity that have the main role for PPI within fish and mice.

Table 1. Parameters considered for Behavioural testing of Zebrafish by Literature Studies – The table shows time periods for different phases in reference to research studies. These studies selected their own intervals for different phases and recorded the activity of zebrafish at certain instances. Latency is the time between a startling pulse and a startle response. Excitation is when a zebrafish shows a startle response as an outcome to a startling pulse. Refractory phase is the time needed before the recovery of the response.

Phase before providing stimulus	**Latency**	**Excitation Delay**	**Refractory Phase**	**Reference**
Darkness	0-2sec	2-5 sec	10-30 sec	(Kokel et al., 2010)
Darkness	~1 sec	5-7 sec	embryos depicted less movements in comparison to the background phase	(Legradi et al., 2015)
Darkness	0-2 sec	2-7 sec	< 15 sec	(Kokel and Peterson, 2011)
Darkness for 12 h	200 ms	300ms	-	(Burgess and Granato, 2007)
Darkness	-	4 minutes (waking activity)	1 minute	(Rihel et al., 2010)
Darkness for 6 minutes	-	500 ms	1110 ms	(Bergeron et al., 2015)

Research studies that have conducted startle response and PPI trials have specifically focused on their experimental setups, such as how to provide the stimulus in their experiments (Legradi et al., 2015; Rihel et al., 2010). The studies described in table 1 performed PPI trials based on parameters of latency, excitation, refractory phase, and startle response in zebrafish.

The study performed by Bhandiwad (2013) shows zebrafish as a useful animal model for growth and genetics of vertebrate inner ear. Acoustic stimuli are provided for PPI trials in zebrafish at the ages of 5-6 dpf. The factors' intensity and timing of pulses and prepulses play an important role in PPI experimentation. They are responsible for either initiating a startle response, or increasing and decrease it, or no change in the behaviour (Burgess and Granato, 2007).

The study (Bhandiwad, 2013) suggests that PPI assay can be further utilised for zebrafish screening. This was particular to study zebrafish hearing mutants and hearing ontogeny in other fish models. Compounds having a modulatory impact on PPI are also studied in animal models. These can contribute to schizophrenia-related behaviours and their underlying mechanisms (Tabor et al., 2014).

1.2 Using Zebrafish for Neurological Analysis

Zebrafish show startle response to different types of stimuli, which makes them useful for neurological assays. The response differs as the zebrafish grow older. Their age is identified in days post fertilisation (dpf). The experimental setup by Bergeron et al., (2015) has a Petri dish containing embedded larvae and is imaged with a compound microscope for observing tail movements of the larvae. The study further uses vibrational stimulus to head fixed embryos and their responses are measured by capturing images and sorting them manually, based on the time points of the tail movement. Besides the use of acoustic stimuli, the study (Bergeron et al. 2015) also uses a high-power LED source for providing photic stimuli. These two stimuli are used for behavioural analysis of zebrafish.

In another similar study (Zhou et al., 2014), the experimentation had also focused on analysing the larval zebrafish and their movements; however, within multi-well plates and utilising computational means of MATLAB tools. These two tools are for measuring the movements of zebrafish larvae. Analysing the movement of larval zebrafish plays a major role in depicting phenotypes, which are outcomes of genetic mutations (Mahmood et al., 2013), toxins (Sallinen et al., 2009), drugs (Rihel et al., 2010), chemogenetic lesions (Elbaz et al., 2012) and gene knockdown processes (Milanese et al., 2012). These aspects target the developmental processes of the nervous system. Based on motor responses of larval zebrafish that might show abnormality, these can help to understand cellular abnormalities (Delvecchio, Tiefenbach and Krause, 2011).

1.3 High-Throughput Methods

Advances have been made in screening processes for zebrafish (Mathias, Saxena and Mumm, 2012). Morphological assays that involve qualitative measures, automated imaging and data processing methods, these altogether present subcellular and organ level information. In this regard, zebrafish as an animal model helps in understanding diseases and provides the opportunity for developing therapeutic measures. Adult zebrafish are used in chemical screening methods; however, there is still the use of embryos in the screening processes. These embryos are between the ages of 0-2 dpf and even young larvae aging 2-7 dpf (Wiley, Redfield and Zon, 2017).

Different screening methods for zebrafish have been developed based on morphology, behavioural aspects such as locomotion, fluorescence and fixed time-point (Mathias, Saxena and Mumm, 2012). High-throughput screening (HTS) zebrafish involves imaging cellular activities and showing expression for a gene reporter or maybe small molecules. For high-content screening (HCS) assays based on microscopy, phenotypic traits such as morphology, physiology, behaviour, these can be observed in parallel (Boutros, Heigwer and Laufer, 2015).

Older zebrafish show more responsiveness compared to young zebrafish (Bhandiwad et al., 2018). This thesis project used 5dpf zebrafish larvae because these can be used as animal models up to 120 h (5dpf) without any ethical permission, as per the EU Directive 2010/63/EU (Strähle et al., 2012). Between the age transitions of 4dpf to 5dpf, zebrafish reach the stage of free feeding and require ethical authorisation for handling. Therefore, zebrafish larvae were not used the entire 5dpf and were sacrificed earlier during the day.

1.4 SHIRPA Scoring and PPI Testing in Mice – Background study

A mouse protocol was run by Florian Freudenberg and Dilhan Esen in an ongoing study at the department of Translational Psychiatry, Frankfurt University Hospital, Germany. The protocol is developed using acoustic stimuli and is based on the study conducted by Swerdlow, Geyer and Braff (2001). This protocol is tested on mice using a startle response system called SR-LAB. It provided different intensities of pulses in decibels (dB) as startle pulses and prepulses. Another protocol for mice, SHIRPA, is further applied in the same study at Frankfurt. This protocol involves open field test on mice for behavioural assessments, each of which has its own scoring pattern.

SHIRPA is an abbreviation of Harwell, Imperial College, Royal London Hospital, phenotype assessment. Researchers working at SmithKline Beecham PLC and other British institutes suggested the name in 1997 (Nolan et al., 2000). The SHIRPA test itself consists of behavioural monitoring, pathological investigation of mice, further using them for modelling neurological disorders (Zhang, 2016). Two specific behaviours (startle response and touch escape) from SHIRPA and PPI protocol tested in Frankfurt are studied for the methodology of this thesis. Even though these protocols were tested on mice and consisted of acoustic stimuli, it provides a background study for zebrafish response behaviour to light stimuli.

2 AIMS AND HYPOTHESES

2.1 Aim

- To develop a scalable unit to synchronise illumination, behavioural stimulation and imaging of larval zebrafish

2.2 Hypotheses

- Larval zebrafish exhibit photic startle responses within 5dpf in the HT system
- Larval zebrafish exhibit Prepulse Inhibition (PPI) of photic startle within 5dpf in the HT system

3 MATERIALS AND METHODS

This thesis project focused on the development of an HT system. Zebrafish larvae were monitored within that system. The HT system was designed in a way that a 96 well-plate with zebrafish larvae can be placed in it. The well-plate was imaged by a mass-produced camera, known as the Raspberry Pi Camera. The camera was controlled by its specific microcontroller generally called Raspberry Pi Board, and Python programming language was used for managing certain camera settings. This was done to image zebrafish accordingly. A planar white light source was used for providing photic stimuli to the zebrafish so that their behaviour to light could be monitored. The camera and light source was controlled by using another microcontroller called Arduino. Arduino was programmed for controlling light, using C language as the programming language. The Raspberry Pi camera recorded the behavioural response of zebrafish to light stimulus.

This project has used microcontrollers that are simply small integrated circuits and consists of interfacing sockets, buttons, LEDs and other such components.

These can be regarded as small computers even, which are flexible in performing tasks when provided with instructions in the form of a coded script. Based on also their small size, they are easy to use and several functions can be done that saves human time. The advantage of using microcontrollers in this project is controlling and synchronising the light source and recording zebrafish behaviour at the same time. For the development of protocols where light pulses were provided after different darkness intervals, Arduino microcontroller was programmed so that the light was on and off at certain instances.

3.1 Raspberry Pi Microcontroller (PiBoard) and Camera (PiCam)

The Raspberry Pi Foundation has developed this Raspberry Pi (RPi) series of small circuit boards. The models within the RPi series differ from each other based on how the circuit hardware has evolved over the years. This is done by increasing the memory and changing peripheral-device related features. Models RPi Zero W V1.1 as the circuit board and V2.1 as the camera were used in this project. RPi circuit boards are also referred to as RPi Microcontrollers and in short as PiBoard. These consists of their own operating system called Raspbian based on the Debian version of Unix. Figure 1 shows the PiBoard model RPi Zero W V1.1 that has been used in this project.

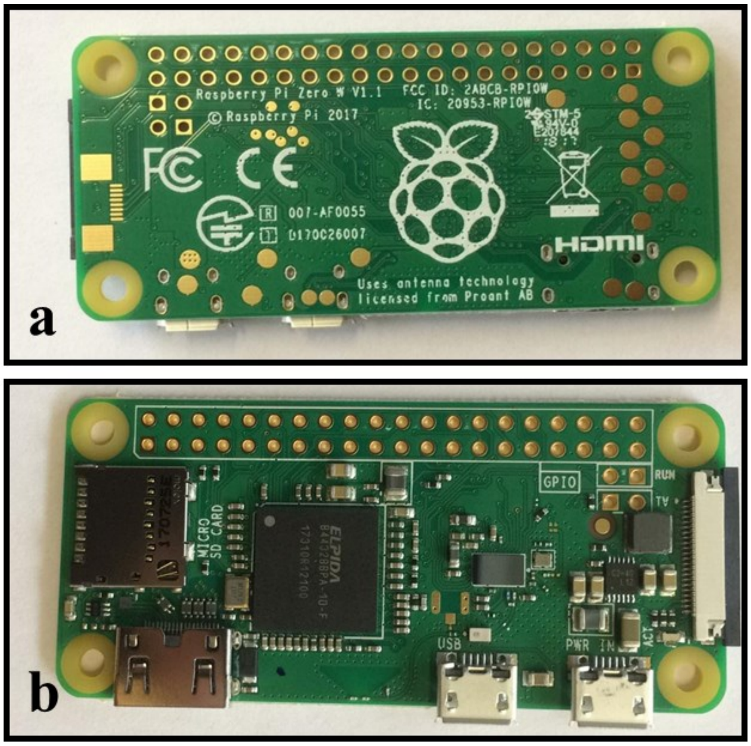

Figure 1. Raspberry Pi (RPi) Microcontroller Model RPi Zero W V1.1 – RPi Microcontroller is regarded as PiBoard and the model shown here is used for this project. Figures 1a and 1b here are simply showing the appearance of PiBoard. Figure 1a shows the back of PiBoard comprising of Model name and details. Figure 2a displays the front of PiBoard consisting of different sockets and circuit components.

For this project, the Raspbian environment was operated via Putty application because it provided the ease of controlling the PiBoard on Windows 7. Putty also provides the advantage of coding in Python language so that a certain program could be tested, saved, and further executed anytime. PiBoard has a socket in which the RPi camera (PiCam) can be connected through a ribbon cable. The socket is identified in figure 2. This makes the camera flexible so that it can be placed according to the application. Generally, every microcontroller consists of certain ports that can be programmed to function as input or output so that they could be connected to peripheral devices. Similar programming was done for PiBoard.

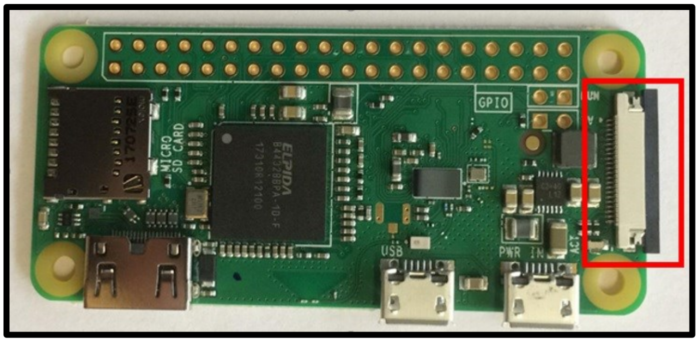

Figure 2. Socket present on PiBoard Zero W V1.1 for connecting PiCam V2.1 – The socket targeted here in the red box is present on PiBoard that is used for connecting the PiCam to the PiBoard.

For this project, the advantage was to easily adjust the PiCam physically for imaging an entire 96 well-plate. One of the major reasons was the ribbon cable that connects the PiCam to PiBoard. The flexibility, easy to use and cost effectiveness of PiBoard and PiCam provided the advantage of making the HT system scalable. Figure 3 shows the PiCam Model V2.1 that is used for this project.

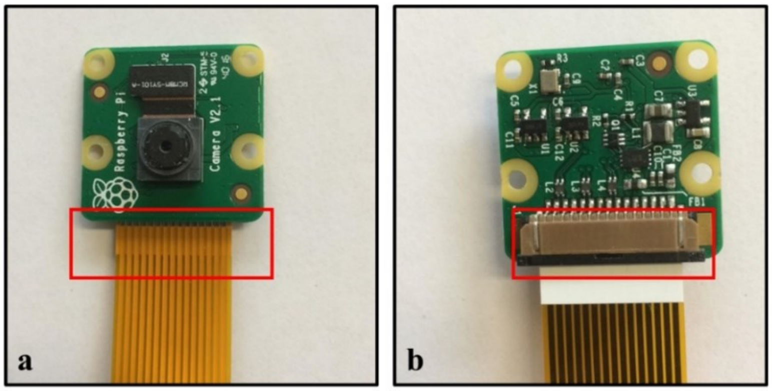

Figure 3. Raspberry Pi (RPi) Camera Model V2.1 – RPi Camera is regarded as PiCam and the model shown here is used for this project. Figure 3a shows the side of PiCam that consists of camera lens, model name and details. It also shows the connection of ribbon cable to PiCam in the red box. Figure 3b shows the other side of PiCam comprising of circuit components and socket (in the red box) that is used for connecting PiCam to PiBoard through the ribbon cable.

Besides the above-mentioned advantages PiCam and PiBoard provide, there were also certain limitations for which the project worked its way around. Both components are sensitive when it comes to user handling. Even in the case of simply picking up PiCam, it should be dealt with proper care. This is because during the project one of the PiCams stopped working since it was not handled carefully. Figure 4 here shows how the ribbon cable is connected to its socket present on PiCam.

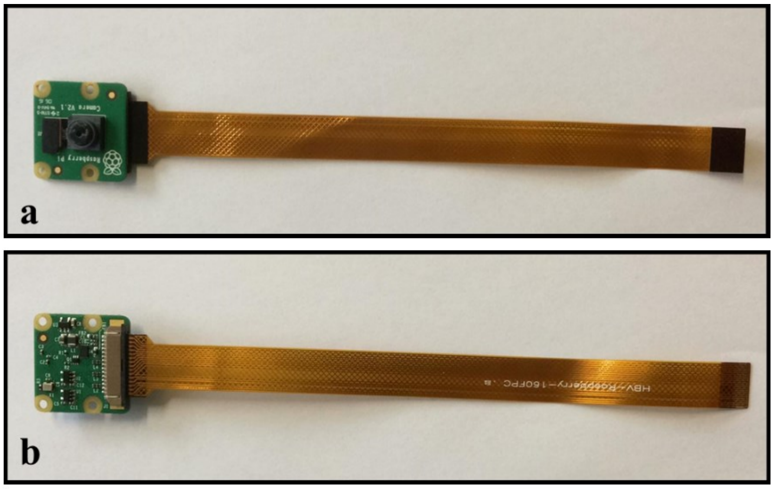

Figure 4. Ribbon Cable connected to PiCam Model V2.1 – The ribbon cable is flat, flexible and long, providing the benefit of placing PiCam anywhere easily. Figures 4a and 4b are simply showing the appearance of ribbon cable and which end of the cable is connected to PiCam.

Similar precautions were taken when handling PiBoard. Another disadvantage was the loose ribbon cable connection to PiBoard. The cable does fit in the socket present on PiBoard; however, the socket connector is a bit loose. This is why the cable was taped to the socket of PiBoard, which is shown in the results section. As shown in figure 5, the wider end of the ribbon cable is connected to PiCam and the remaining end is connected to PiBoard.

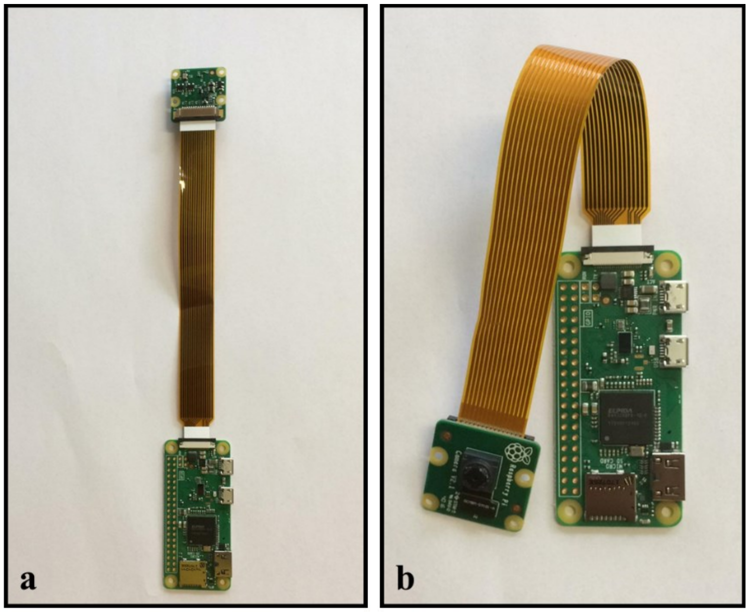

Figure 5. Ribbon Cable connecting PiCam and PiBoard – The cable is inserted in the socket of PiBoard. Figure 5b shows the flexibility of the cable when the PiCam is connected to it.

3.1.1 Using Raspbian Environment with Python Programming Language

Raspbian environment of PiBoard was set to have static Internet Protocol (IP) addresses that allow the user to communicate with it via internal Wireless LAN (WLAN) of the Neuronal Signalling lab. When using more than one PiBoard, each of them could be specified by their respective IP in Putty application. This provided the connection between PiBoard and lab computer over WiFi. The PiBoard settings were defined through writing a coded script in Python language and saving it in the Raspbian environment for controlling PiCam.

Initially, one PiCam was tested with different camera settings because the aim was to observe the behaviour of zebrafish to light. If the camera captures light as too bright, imaging would become difficult and zebrafish would not be visible. For this purpose, shutter speed of PiCam was considered because it refers to the time during which the camera would interact with light. This setting could be easily adjusted through the Python code, for which the PiCam library and certain standard libraries were loaded in PiBoard. A simple command of ‘camera.shutter_speed’ was entered for adjusting the shutter speed. A coded script provided the ease of executing a saved program every time. The commands used for loading these libraries in the Python code are as follows,

```
from picamera import PiCamera
from time import sleep

import os
import time
import sys
import serial
import subprocess

from subprocess import call
```

Shutter speed was entered as milliseconds in its command, where a value such as 10000 is equal to the 1/100th of a second. In this way different shutter speed values were tested and camera preview was observed under bright light. The command appears in the following manner,

```
camera.shutter_speed = 10000
```

Planar White LED Light Source (LED copy board A4) was used to provide photic stimuli. Default exposure setting of PiCam is automatic, so when it interacts with light the image becomes overexposed. For this reason, the automatic exposure of PiCam was turned off through Python code. Since the aim was to image zebrafish to light, the camera was further set to greyscale imaging. Framerate and resolution camera settings were also considered. Default framerate of 30 frames per second (fps) and resolution of 1280x720 pixels were set through Python code. Rapid movements of zebrafish were recorded and 30fps is suitable for video recordings with such movements. Increasing fps would also mean increasing lighting requirement. With an increased resolution, it provides the advantage of more detailed and good quality videos. By keeping 30fps as constant, the shutter speed was easily tested to light and value of 3000 was entered in the program. This was able to preview image from the camera when the light source was on at its brightest level.

For other PiCam settings that effect the camera preview to bright light and darkness, these are ISO (International Organization of Standardization) sensitivity and Automatic White Balance (awb). ISO setting is the ability of PiCam for capturing light. Default ISO for PiCam is 0, which was changed to 100 because the 30fps image was visible in this ISO setting. The default setting of awb for PiCam is automatic, which was set to off. The awb gain was set to 1, 8 and colour effects were set to 128, 128 to achieve greyscale image. The Python commands used for the mentioned PiCam preview settings have been defined in table 2.

Table 2. Python Command Lines and their functions for PiCam Preview – These commands were specifically entered in the Python code for controlling preview of PiCam so the image could be clear and observable under bright light.

Python Command Lines for PiCam Preview	**Functions performed by Commands**
camera = PiCamera()	Defining PiCam to a variable
camera.color_effects = (128,128)	Greyscale Imaging
camera.exposure_mode = 'off'	Turning off automatic exposure of PiCam
camera.shutter_speed = 3000	Appropriate Shutter Speed value showing 30fps image under bright light
camera.iso = 100	Appropriate ISO value for above Shutter Speed value and 30 fps

camera.awb_mode = 'off'
camera.awb_gains = (1,8)
	Turning off default PiCam Automatic White Balance and setting values relevant for showing greyscale image without overexposure

The target of this thesis project was also to record zebrafish behaviour to light. With Python code, the length of recording time was defined. The time varied for different lighting protocols, which is why recording length was changed for each protocol. The PiBoard controlled these recording time for PiCam through the Python code. Further process involved receiving the recorded files from PiBoard, for which FTP (file transfer protocol) WinSCP (Windows Secure Copy) was used.

In this section, until now camera preview settings have been identified for PiCam in Python code. The code further included commands instructing PiCam to record for a certain amount of time and then stop, as shown in Appendix 8.1. This created and saved video files in h264 format in the Raspbian environment of PiBoard. PiBoard ports programmed for this thesis are shown in figure 6.

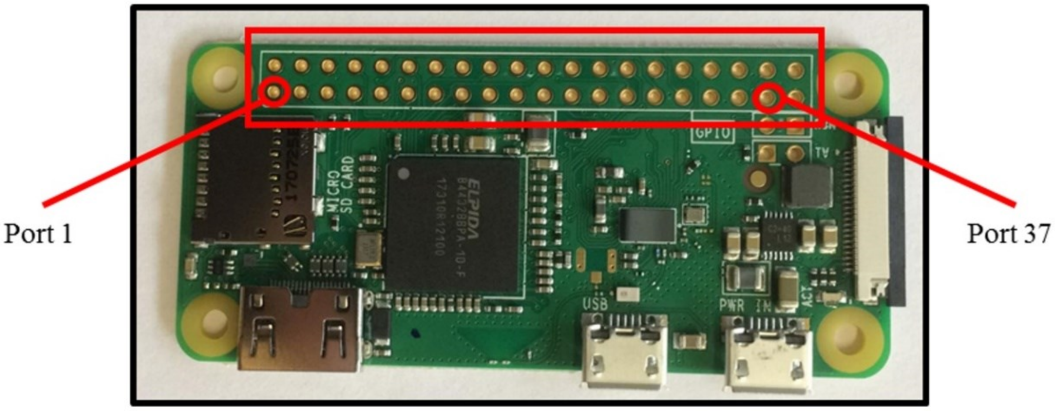

Figure 6. Ports of PiBoard – The red box shows two rows of ports in PiBoard. The port shown on the left in the red circle is port 1 that was used in this project. Going from port 1 to the left is the row of ports identified with odd numbers, and the row above is of even numbers. This is how physically ports are defined for PiBoard. On the right side, port in the red circle is 37 and was used in this project.

Certain ports on PiBoard can be used for the pullup and pulldown logics for knowing that whether the port has the logic level 1 or 0 respectively. Port 37 on PiBoard is one of such pins and was programmed in Python to have a logic level of 0 through the pulldown operation. This simply means the port had no voltage compared with ground. Ground is a reference point for voltage in circuits. Logic level of 1 indicates that the port has voltage of 3.3 volts (V) in PiBoard compared to the ground. Python code has different port numbers when using those ports as input and output, so port 37 on PiBoard is port 26 in Python code. Following Python commands were entered to do so.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(26, GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
 #Official port 37 on PiBoard
print GPIO.input(26)
```

After setting up port 26 to work as an input pull down (0) pin, a while loop logic was executed that lets the Python program constantly check port 26 whether it has a logic of 0 or 1. If it stays 0 as per the pull down logic, the program instructs in checking the signal on the port again and again. The program further has the logic that if port 26 receives an external signal of logic 1 and changes its state from 0 to 1, the program should execute further recording commands present in that while loop. This external signal 1 was programmed to be provided by Arduino microcontroller to PiBoard port 26 so that it would start recording and light protocol at the same time. The recording time was defined in the command ‘sleep(120)’, where 120 is referred to seconds in Python. The recording time was changed for different lighting protocols. This was executed through the following commands and is presented as a flowchart in figure 7,

```
while GPIO.input(26) == 0:
        print 'Pin 26 is Low'

if GPIO.input != 0:
        camera.start_preview()
        print 'PiCam is recording'
        camera.start_recording
        ('/home/pi/Desktop/Startle_Response.h264')
        sleep(120)      #changing the recording time based on
                        applied light protocol
```

```
camera.stop_recording()
print 'PiCam stopped recording'
```

The output of print commands appeared on the Python program in the Putty application. This showed which command was being executed at a certain time.

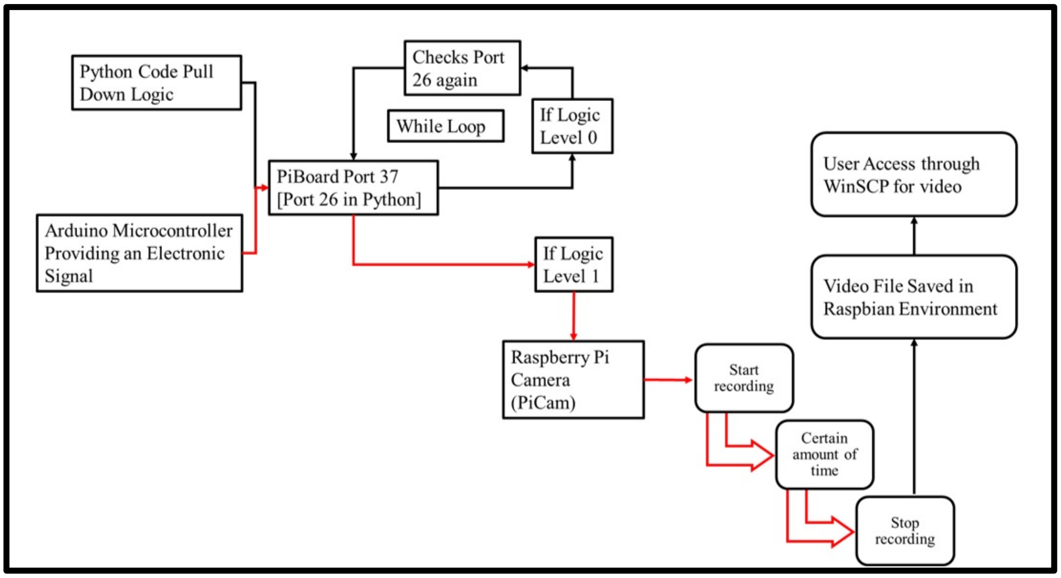
Python Code Pull Down Logic
Arduino Microcontroller Providing an Electronic Signal
Checks Port 26 again
While Loop
If Logic Level 0
PiBoard Port 37 [Port 26 in Python]
If Logic Level 1
Raspberry Pi Camera (PiCam)
Start recording
Certain amount of time
Stop recording
User Access through WinSCP for video
Video File Saved in Raspbian Environment

Figure 7. Python Code Flowchart - The code runs a while loop for checking the electronic signal on port 37 (26) on PiBoard. If the port has logic 0, the python code continues to check the port. If there is logic 1 received from Arduino Microcontroller, Python instructs PiBoard to start recording through PiCam and saving it as a h264 format video file in the Raspbian environment.

3.2 Supporting Framework for the HT System

The next step was to synchronise PiBoard with the light source so that PiCam starts and stops the recording with the lighting sequence. This synchronisation was first tested on a one-camera setup, where several changes within the Python code were made and executed to observe how the operating system worked. However, before testing this synchronisation, it was essential to have a structure that would support the PiBoard, PiCam and light source. A cardboard box was used as a framework for supporting these circuit components. In total, 3 PiCams along with their respective PiBoards, were programmed and mounted on top of the cardboard box. These were already working before the start of this thesis; however, they were operated on the command line of Raspbian using Linux commands and not on Python. In addition, Arduino was not synchronised with PiBoards before the start of this project. This demonstrated scalability to the system and the benefit of imaging 3 96 well-plates at the same time.

The cardboard box was a supplier box for standard sized 96 well-plates (length 127.71 mm x width 85.43 mm and height 14.10 mm) so it included columns in which the well-plates would easily fit. This gave the structure for placing a light source underneath those columns, while holding the circuit on top of the box. The covering underneath the box was removed so that the box can be placed on top of the planar light source. The well-plate was kept inside the box, right above the light source. In this way, zebrafish would receive light from underneath and PiCam could capture the entire well-plate from top.

For testing camera features and supporting PiCam and PiBoard, these components were placed on the lid of the box for the time being. The lid was light weighted and not thick enough, so it was loosely placed on top of the box. This is why the lid was positioned asymmetrically and PiCam was not properly placed on it. This was the reason why tilted images were received when one-camera setup was tested for lighting settings. However, the columns within the box themselves were also positioned asymmetrically, which further contributed in tilted images and videos even when zebrafish was used in 96 well-plate. The distance between the lid and bottom of the box was 140mm. This distance was initially taken to observe if PiCam could preview the entire field of view (FOV) of the well-plate. This distance was found to be suitable because the entire well-plate was visible by PiCam when the one-camera setup was tested.

3.3 One-Camera Setup

One-camera setup was tested to observe how a 96 well-plate appears under PiCam when the light source is on. Figure 8 shows the entire design that was used for testing one PiCam for one 96 well-plate.

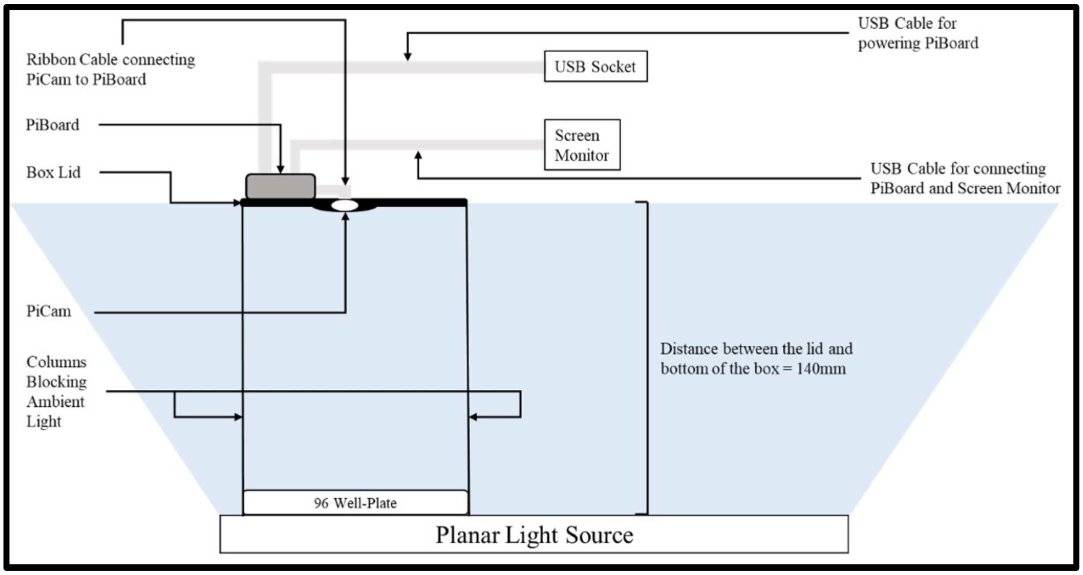
Ribbon Cable connecting PiCam to PiBoard
PiBoard
Box Lid
PiCam
Columns Blocking Ambient Light
USB Socket
Screen Monitor
USB Cable for powering PiBoard
USB Cable for connecting PiBoard and Screen Monitor
Distance between the lid and bottom of the box = 140mm
96 Well-Plate
Planar Light Source

Figure 8. One-Camera Setup – The design shows components of the one-camera setup. The PiCam settings shown in table 2 were programmed in the Python code and then the code was loaded on PiBoard. An empty 96 well-plate was placed inside the box. The columns shown in the design represent the ones that were present in the box and contributed in blocking ambient light. When zebrafish larvae were pipetted in the well-plate, it was necessary that only the light coming from light source should be the stimuli. The light travels from the source and underneath the well-plate, further to the PiCam. The screen monitor showed live imaging of the well-plate.

3.4 Empty Well-plate Test for HT System Optimisation

The settings shown in table 2 seemed appropriate because the entire well-plate was visible when the light was on. The reason to test an empty well-plate was to determine how well walls were a limitation for imaging the entire bottom of each well at different positions in the plate. A comparison of black flat bottom well-plate and transparent round bottom well-plate was done by imaging both well-plates separately. This comparison helped in understanding the defined limitation. The thesis proposed the use of a convex lens on top of the well-plate for solving this limitation so that parallel light from the well is directed towards the camera. This way the lens would be in front of the object (well-plate). The positioning of the lens was further tested by considering the distance between the lens and well-plate. This thesis used round bottom transparent 96 well-plate for zebrafish experimentation. The benefit of round bottom plates is that the shape contributes in centring the head of zebrafish (Liu, et al., 2016).

3.4.1 Fresnel Lens

This thesis used a Fresnel lens that is a convex lens. Fresnel lens was designed by Augustin-Jean Fresnel. The design shown in figure 8 was further changed by presenting a Fresnel lens on top of the well-plate, as shown in figure 9. The appearance of the lens is shown in figure 10. The size of the lens was appropriate, as it covered most of the wells of the entire well-plate.

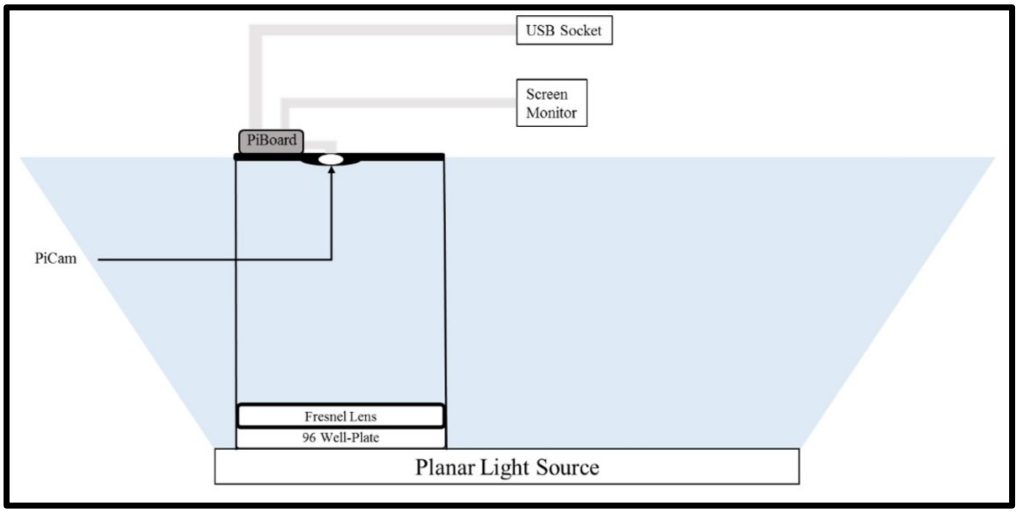

Figure 9. Improvising One-Camera Setup by placing a Fresnel Lens on top of the Well-Plate – The one-camera setup design shown in figure 8 did not consists of a Fresnel lens. The same design has been improvised here by placing the lens on top of the well-plate. This helped in working around the problem where well walls were limiting the entire bottom visibility for imaging.

Fresnel lens is available in small and large sizes, which became an advantage for this project. The size used here is 120 mm x 80 mm. The lens is also lightweight so it did not damage the well-plate and became easier to handle.

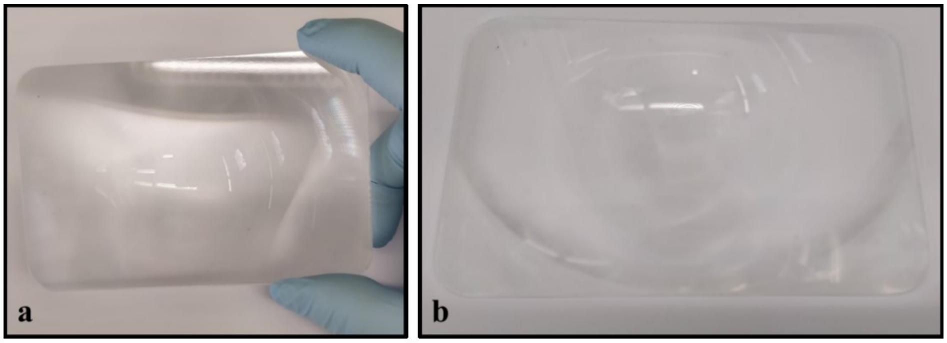

Figure 10. Fresnel Lens – Figures 10 a and b show the appearance of the Fresnel lens used in this thesis. The size of Fresnel was appropriate for a 96 well-plate. The lens is smaller than a 96 well-plate (127.76 mm x 85.48 mm) and the shape was rounded rectangle. The lens size is 120 mm x 80 mm.

The size of the lens and its rounded edges gave the advantage of easily fitting it in a 96 well-plate frame, which is usually used for 1 x 12 strips. Figure 11a shows the frame that was used in this project and figure 11b shows the lens fitted into that frame. The well-plate frame with the lens fitted-in was placed on the well-plate and imaged in the one-camera setup. This is how the position of the lens was optimised. The thesis did not use corner wells because they were not fully visible in the HT system. It is acceptable not to use the corner wells because they are usually more prone to evaporation. In common laboratory practice, long-term experiments that use well-plates avoid the entire edges, known as the edge effect (Carralot et al., 2011).

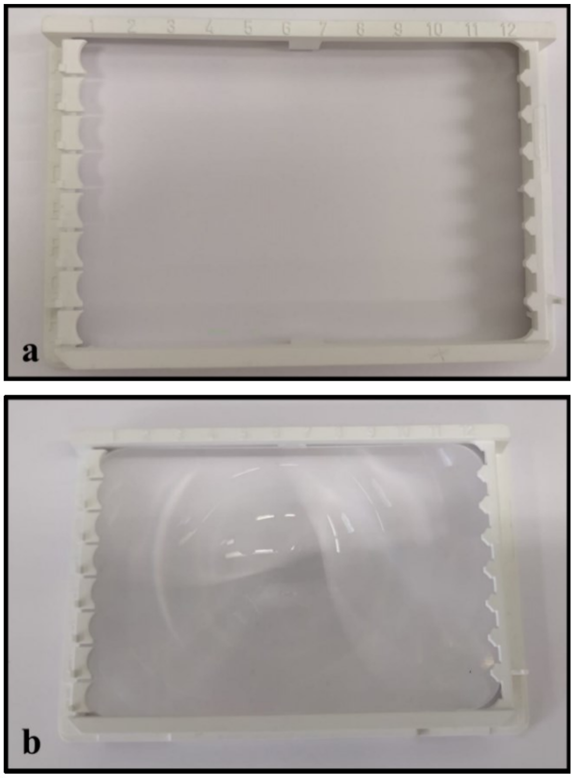

Figure 11. 96 Well-Plate Frame 1 x 12 Strips used for holding Fresnel Lens – 11a shows the empty frame in which Fresnel lens was fitted-in. The size of the frame is 127.76 mm x 85.48 mm. 11b shows the frame with Fresnel lens.

3.5 Light and Camera Control through Arduino Microcontroller

The next step was to control the light source so that it works with precision for lighting protocols. Since, the behaviour of zebrafish was observed to light pulses; the timing of these pulses had to be provided with accuracy. In addition, darkness periods are equally important for zebrafish behaviour, for which the light source should be turned off correctly in those intervals. Another part was to control the light source and PiCam together for the recording and lighting sequence to start at the same time. For fulfilling such requirements, Arduino Microcontroller was used in this project. Arduino runs a real-time operating system (Arduino, 2015) making it more essential for this project, but cannot manage a camera. It is the other way around for PiBoard, which is the reason why both microcontrollers are used in this thesis. Arduino majorly contributes in precise timing and PiBoard does not. This is why Arduino is used to control light source so the lighting protocol would have accuracy.

3.5.1 Planar White LED Light Source

The light source used in this project is LED copy board A4 and its size is 36 cm x 24 cm, as shown in figure 12. The light source itself included a circuit that can provide 3 levels of light intensities and is powered by a simple USB cable. The project required controlling light, which is why the circuit of the light source was adjusted to work with Arduino in a way that light intensity could be changed.

Figure 12. Planar White LED Light Source – This is LED copy board A4. The size of light source was 36 cm x 24cm and height 5mm. The entire cardboard box easily fitted onto the surface of this light source.

3.5.2 Arduino Microcontroller

The title Arduino covers an entire range of single board microcontrollers that have been produced over the years for controlling digital and mechanical devices.

Arduino is easily operated through its own software and easy-to-use programming platform.

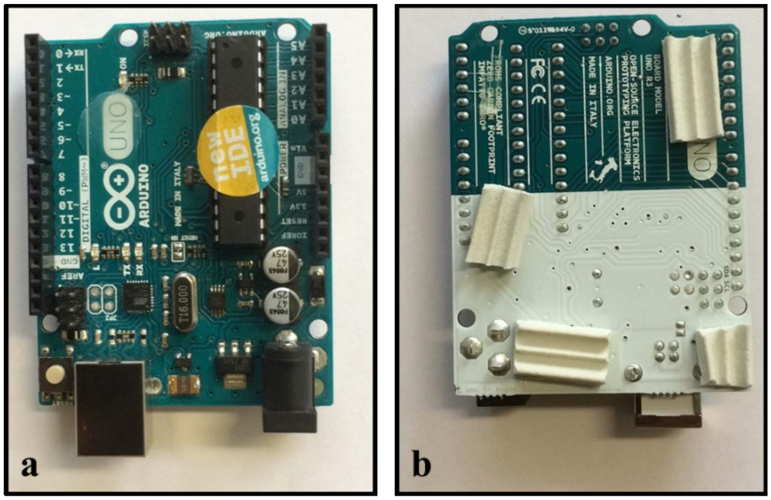

Figure 13. Model Arduino Uno R3 – This model of Arduino microcontroller was used in this project. Figure 13a shows the side that consists of ports that are used for connecting other devices to Arduino. The remaining side of Arduino is shown in figure 13b, showing model details.

The computer used for this project was running on a Windows 7 OS in the laboratory. Arduino Integrated development Environment (IDE) was used for developing and uploading the code written in C. The model used in this project is Arduino Uno R3, as shown in figure 13. The basic scheme of controlling light source and PiCam is shown in figure 14, which was simply done by improvising the designs from figures 8 and 9. Figure 14 shows the placement of Arduino in the one-camera setup. Other advantages of Arduino include cost effectivity and open source. Besides these advantages, the main benefit of using Arduino in this thesis project was its feature of Pulse Width Modulation (PWM) that is further explained.

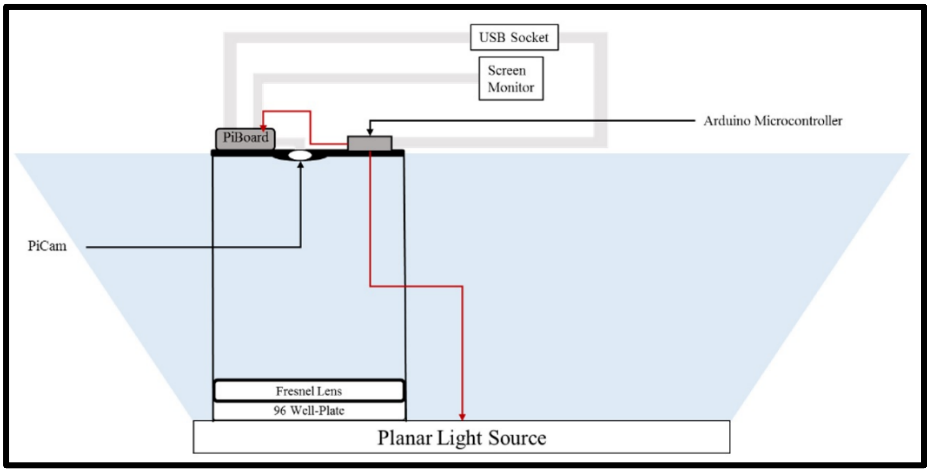

Figure 14. Placement of Arduino in Controlling Light Source and PiCam – Arduino controls the camera recording by managing the PiBoard, which further provides instructions to PiCam recording. The one-camera setup design was simply improvised by placing Arduino on top of the box lid. This figure provides a basic idea of the project design on how it was decided to add Arduino to the one-camera setup. The designs shown in figures 8 and 9 are improvised here by displaying the placement of Arduino.

3.5.3 Pulse Width Modulation (PWM) feature of Arduino

Pulse Width Modulation (PWM) is a method for applying digital processes for receiving analog outputs. Generally, a digital signal is referred to an on and off state that is depicted through a square wave. When Arduino produces such a signal, the voltage values produced on these on (HIGH) and off (LOW) states are 5V and 0V respectively. The length of on state is regarded as the pulse width, which can be varied, hence the term modulation in PWM. PWM of Arduino has its own specific frequency of 500 Hz. This means a certain cycle of on and off state is repeated 500 times within one second. Since, time is the inverse of frequency (Time = 1 / Frequency), in this case the time for each cycle would be 2ms (1 / 500 Hz). The on and off conditions would reoccur 500 times in a fast manner. If on time is less than off time and an LED source is connected to the Arduino, the light intensity of the LED source would be dimmed. This way Arduino can control a light source.

The red lines in figure 15 show the 2ms period for one cycle. The percentage of on time in this cycle is known as the duty cycle. If the entire cycle has on state, it means that the duty cycle is 100%. If the cycle is 75% on, it would mean 75% duty cycle. It is understandable that the rest of the 25% cycle would be off, as shown in figure 15. When certain cycles of 2ms is repeated 500 times and is provided to a LED light source, at 100% duty cycle there would be bright light, and it would become dimmer at 75% duty cycle. This provided the main advantage of using Arduino so that the brightness of light source could be easily controlled. This is the reason why light intensity in the project is characterised in percentage.

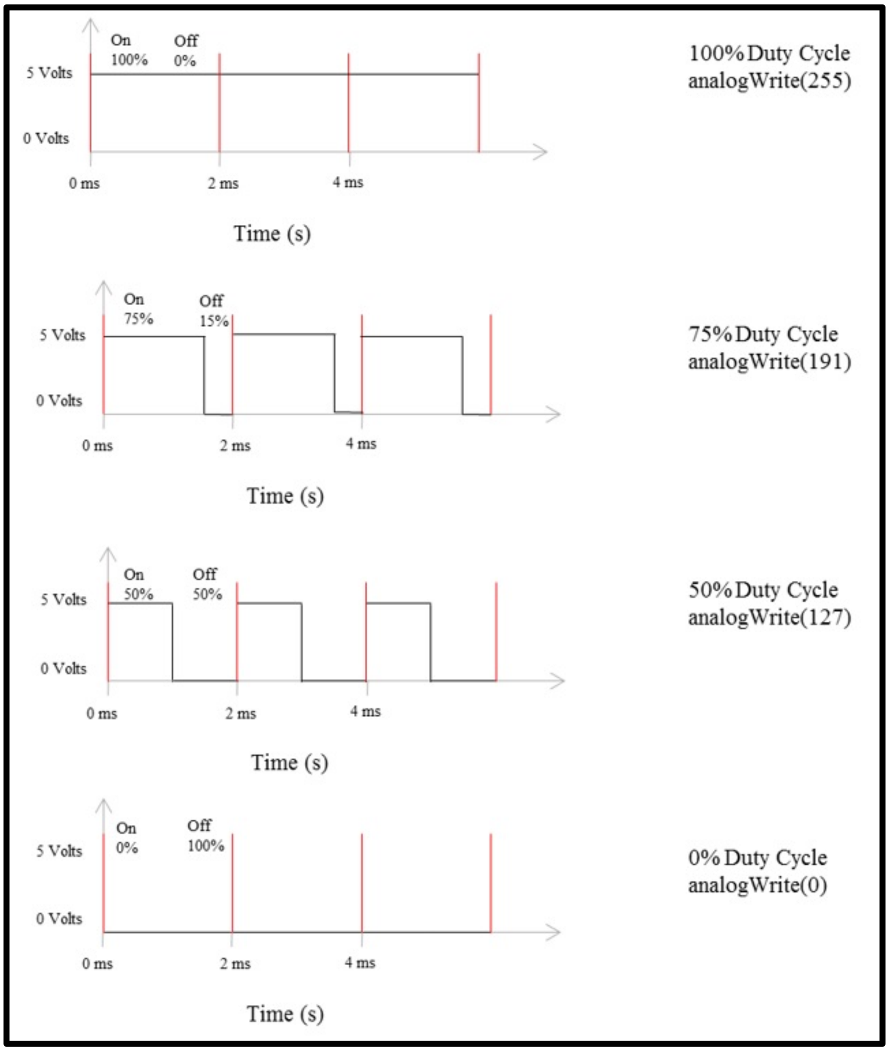

Figure 15. On-Time Cycle in Pulse Width Modulation (PWM) – Arduino PWM has 500Hz frequency, making each cycle of an on and off state equal to 2ms. Within this period, the on state is regarded as the duty cycle and represented in percentage. In C language, it is represented in the 0 – 255 range, where 255 is 100% duty cycle and 0 is 0% duty cycle. This is the reason light intensity is represented in percentage form in this thesis. Duty cycle is controlled through the analogWrite() function in C code.

A function represented as ‘analogWrite()’ in C language is used for loading PWM in Arduino. The code controls the duty cycle percentage in the range of 0 – 255. If 255 is entered, it means that the duty cycle is 100%. If 191 is entered, the duty cycle would be 75%, as shown in figure 15. The command is entered in the following manner,

```
analogWrite(255)
analogWrite(191)
```

Certain Arduino ports are dedicated to the PWM feature, out of which port 11 (PWM) was programmed for controlling the light source, as shown in figure 16. Port 12 was programmed for instructing PiBoard to start recording through PiCam. Port 2 was programmed for executing the commands altogether that control Port 11 and Port 12. This way the recording and lighting protocol would start at the same time. The C script for one light protocol is in Appendix II, which shows the programmed port 11 for duty cycle in the following way. The following delay command shows that port 11 provides the programmed duty cycle (0%) for 1s.

```
analogWrite(output_pin11, 0);
delay(1000);
```

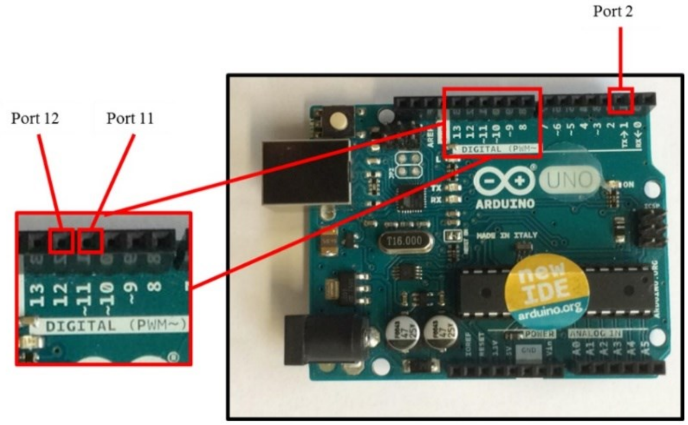

Figure 16. Selected Arduino Ports – Port 11 is the PWM pin programmed for controlling light source. Port 12 was programmed to control PiCam recording. Port 2 was programmed to execute the commands controlling ports 11 and 12 at the same time. This way light protocol and recording would start at the same time.

3.5.4 Using Arduino Environment with C Programming Language

Arduino consists of an environment supporting C language. The code could perform functions exactly the way they are performed in C. This is the reason why sometimes the programming language is regarded as Arduino language. The coded script was written on Arduino software in the laboratory computer. Arduino is connected to the computer through a USB cable. The coded script was then loaded on Arduino through the software. The coded script for this project comprised of loading the desired ports. This way the program knows which ports would be used. Three different variables were assigned to the above three mentioned ports in the following manner,

```
int output_pin11 = 11;
int output_pin12 = 12;
int input_pin2 = 2;
```

Since, ports 11 and 12 were used to provide output electronic signals to light source and PiBoard respectively, they were programmed as output pins. Port 2 was programmed as an input and a pull up pin, having logic level 1 as described above. This was done through the following commands,

```
pinMode(output_pin11, OUTPUT);
pinMode(output_pin12, OUTPUT);
pinMode(input_pin2, INPUT_PULLUP);
```

Port 2 in Arduino is designed for such purpose because it consists of an internal resistor that is connected to 5v, providing logic level 1 (HIGH) to the port. The C script was programmed to check the state of port 2 that if it is 1, the program should keep on checking, as shown previously in figure 7. This was done through a while loop. Further, the script included that if the state of port 2 changes from 1 to 0 (LOW), port 12 should output a constant high signal. PiBoard was connected to port 12 and when it received the mentioned high signal, it executes the commands to start PiCam recording, as shown in figure 17. The C script also included that at the same time, port 11 should output the light sequence programmed through PWM feature. Light source was connected to port 11. Figure 7 showed how Python code would record and save video files. Figure 17 shows the Python code in the blue box and how Arduino controlled light source and recording.

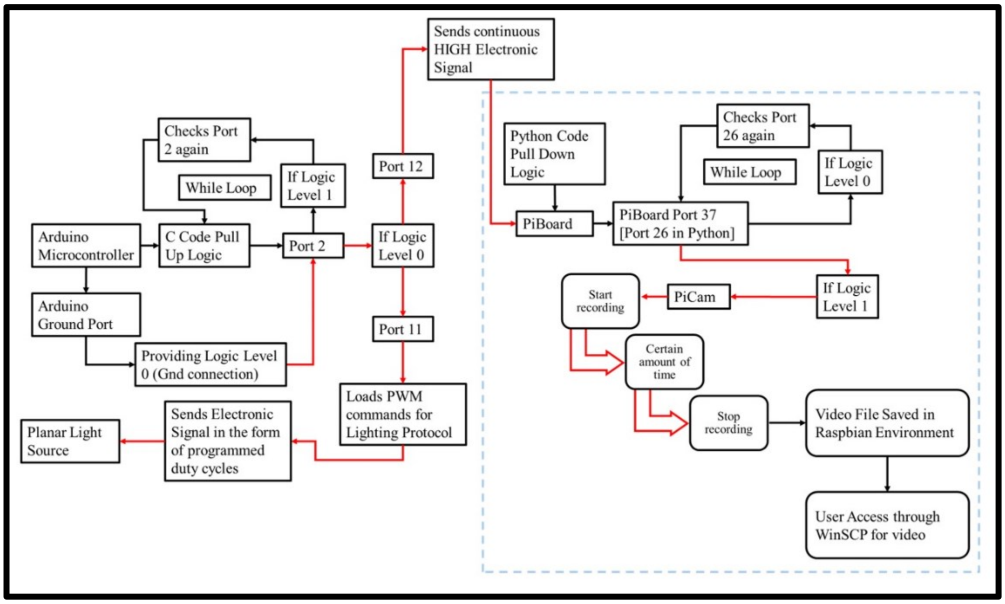
Sends continuous HIGH Electronic Signal
Checks Port 2 again
While Loop
If Logic Level 1
Port 12
Arduino Microcontroller
C Code Pull Up Logic
Port 2
If Logic Level 0
Arduino Ground Port
Port 11
Providing Logic Level 0 (Gnd connection)
Loads PWM commands for Lighting Protocol
Planar Light Source
Sends Electronic Signal in the form of programmed duty cycles
Python Code Pull Down Logic
Checks Port 26 again
While Loop
If Logic Level 0
PiBoard
PiBoard Port 37 [Port 26 in Python]
Start recording
PiCam
If Logic Level 1
Certain amount of time
Stop recording
Video File Saved in Raspbian Environment
User Access through WinSCP for video

Figure 17. Logic applied by C and Python Coded Scripts in Arduino and PiBoard – Previously in figure 7, Python logic was described for operating PiCam through PiBoard. Here in figure 17, the Python program is shown in the blue box and outside is the C program. The diagram shows the sync between PiBoard and light source through Arduino.

Before every lighting sequence, 1s of darkness delay was provided in the C code because PiCam recording does not start right away. If recording and light protocol were to start together, there would be a problem of losing data since the recording does not initiate immediately. For instance, if a certain light protocol had 1s of light pulse followed by 10s of darkness and recording started exactly when the protocol started; the part of 1s light pulse would be lost. By having 1s of darkness in light protocol, it became easier not to lose data at the beginning of the recording. The reason to record the entire lighting sequence and not only the light pulse intervals, was to observe zebrafish behaviour during darkness intervals as well. Even though the PiCam settings were not suitable to observe the activity in darkness, another approach of providing lowest level of light (1% intensity) instead of darkness was also put forward. This was to observe that how zebrafish would behave in constant lower light with respect to 100% light pulse.

3.6 SHIRPA Scoring and PPI testing on Mice (Background Study)

The study performed at Frankfurt involved SHIRPA and PPI protocols. The scoring from 0 to 3 for two behaviours in SHIRPA was done in the following manner,

- Startle Response
    - 0 = None
    - 1 = Preyer reflex (backwards flick of pinnae)
    - 2 = Jump less than 1 cm
    - 3 = Jump more than 1 cm
- Touch escape (TEs)
    - 0 = No response
    - 1 = Mild (escape response to firm stroke)
    - 2 = Moderate (rapid response to light stroke)
    - 3 = Vigorous (escape response to approach)

The PPI protocol (acoustic stimuli) consisted of randomised trials. The trails involved testing different prepulse intensities, startle pulse alone, and baseline as defined below,

- 6 Startle Pulses were (each having intensity 120 dB + duration 40 ms) provided at the beginning of the protocol (see figure 18)

- Each of the following trials were applied 10 times in the same protocol, being 70 trials in total. Each trial lasted for 500ms and was randomly arranged and provided to the mice (see figure 19),
    - 10 trials of no stimulus (consisting of baseline 65dB)
    - 10 x [73dB prepulse + startle pulse]
    - 10 x [69dB prepulse + startle pulse]
    - 10 x [77dB prepulse + startle pulse]
    - 10 x [81dB prepulse + startle pulse]
    - 10 x startle pulse alone
    - 10 Inter-Trial Intervals (ITI) 10s
- 6 Startle Pulses (120dB + 40ms) provided at the end of the protocol

Figure 18 shows that startle pulse was 120db, almost twice the amount of baseline. If keeping 120 as the highest range here (100%), 65dB baseline would be 54%. The prepulses provided were above this baseline, 69dB, 73dB, 77dB and 81dB, as shown in figure 19.

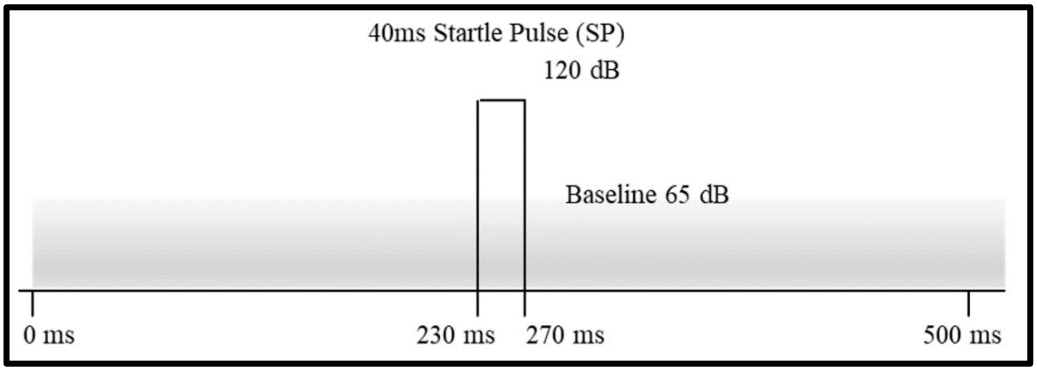

Figure 18. Startle Pulse Intensity in PPI Protocol for Mice Background Study– Each trial in PPI protocol was 500ms long and duration of startle pulse was 40ms. This was used as an acoustic stimulus (120dB) for mice in the SR-Lab system at Frankfurt.

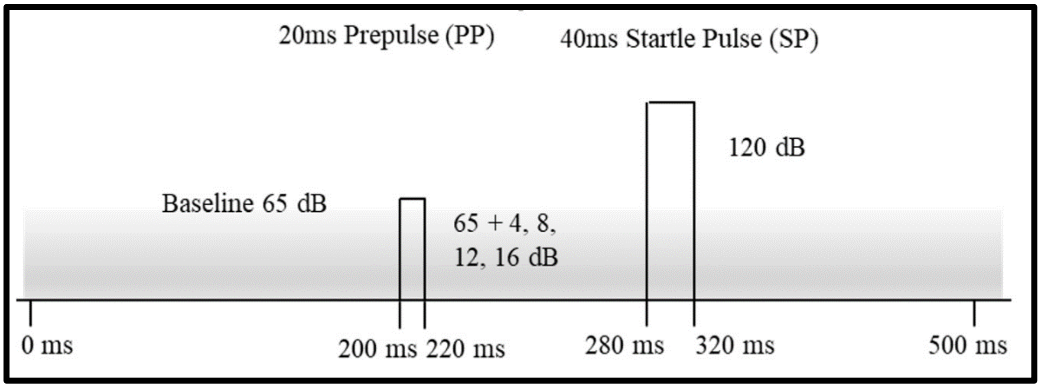

Figure 19. Prepulse Intensities in PPI protocol for Mice Background Study – Trials involving prepulses and startle pulses were also 500ms long. Duration of each prepulse was 20ms. Selected prepulse intensities were 69dB, 73dB, 77dB and 81dB.

3.7 Zebrafish Animal Model Testing

This thesis project used wild type larval zebrafish as an animal model. 4dpf and 5dpf zebrafish larvae were used. Zebrafish embryos 0dpf were provided by Zebrafish Core, Turku Bioscience. These were present in their E3 medium in a petri dish, as shown in the red circle in figure 20a. They were incubated at 28 °C until they reached 4dpf and were pipetted in a 300µl 96 well-plate along with the E3 medium for experimentation, as shown in the red circle in figure 20b. The body lengths of zebrafish for 4dpf and 5dpf were 3.7mm and 3.9mm respectively. A single zebrafish and E3 medium were pipetted in each well from the petri dish. This was a combined amount of 250µl in each well for creating a convex meniscus and 60 wells were used, as mentioned before. Three batches of zebrafish were used and when each batch reached 5dpf, they were sacrificed. 250µl of Tricane was pipetted in a petri dish containing zebrafish larvae for anesthetising them. After 10 – 15 minutes, they were spilled into a flask full of 400ml water + ice and 4ml of Chlorine. The flask was left for 10 – 15 minutes and the contents were discarded in the lab sink.

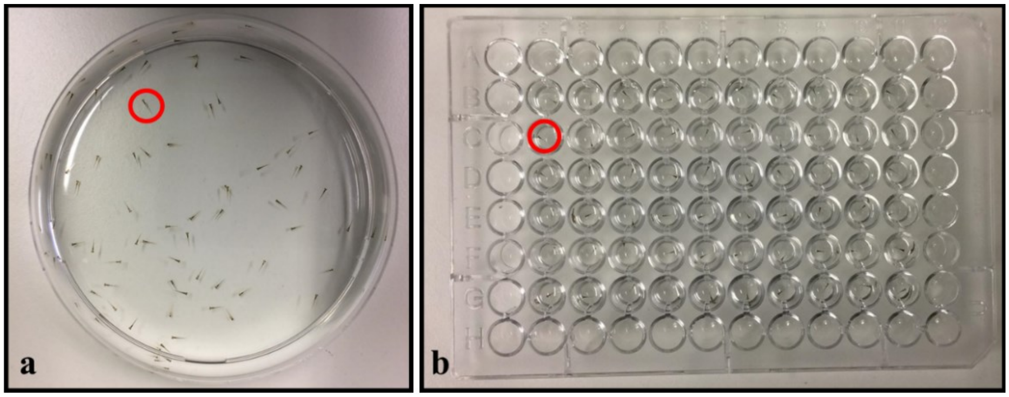

Figure 20. 4 Days Post Fertilization (dpf) Zebrafish Larvae – The body length of 4dpf zebrafish larvae is 3.7mm, and can be easily pipetted into a 300µl well-plate. Figure a shows the petri dish provided by Zebrafish Core and figure b shows the same 4dpf zebrafish pipetted in the well-plate. Each well consisted of a single zebrafish with E3 medium adding up to the amount of 250µl.

3.8 Light Protocols for Behavioural Testing

The lighting protocols were developed based on the background study performed in Frankfurt. The accommodation time for zebrafish was 1hour of darkness before any lighting protocol was applied. After they were pipetted into the 96 well-plate, the plate was kept inside the HT system for 1h and then lighting protocol was provided. Figure 21 shows how the pulses and prepulses are identified graphically for the protocols in this section.

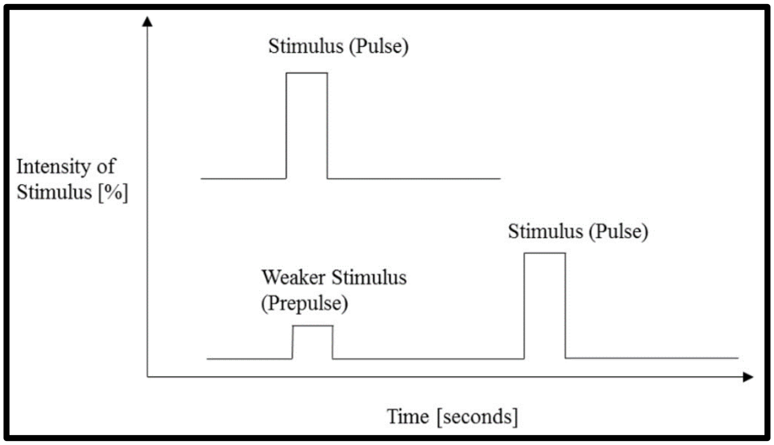

Figure 21. Graphical Representation of Pulse and Prepulse – For lighting protocols defined in this section, this graphical representation is used for displaying the pulses and prepulses.

3.8.1 Randomised Trials Protocol

Randomised trials performed in the Mice study were converted into the light protocols through PWM logic mentioned before. 120dB was taken as 100% light intensity meaning that the command ‘analogWrite(255)’ was entered as startle pulse in this protocol. This protocol was applied in the first batch to observe responsiveness from zebrafish. However, because the protocol consisted of different light intensities for prepulses and a continuous baseline of 1% light was provided, the data showed bars. This is because the PiCam settings were based on 100% of light, and lower intensities were not visible clearly in the videos, thus the appearance of bars. This is also the reason why image analysis was not performed for such data, since the bars in different frames could be misinterpreted as movement by the software. (ImageJ software was used for image analysis in this project and is further defined)

In the first trial, 1% of light was selected as baseline; however, 15% of baseline was accidentally coded. This did not change the output behaviour of the zebrafish. Even though, system errors were encountered in randomised trials protocol, this was performed three times in total for observing zebrafish behaviour to the protocol. This was done to make sure that whether it was the protocol or the age of zebrafish for their behaviour, which is why the protocol was tested on 2dpf, 3dpf and 4dpf zebrafish, as shown in table 3. 1% of light as baseline was considered because of being the lowest possible intensity that light source can provide. The mentioned protocol was performed thrice in the first batch of zebrafish.

Table 3. Randomised Trials Protocol based on Mice Background Study – This protocol was tested in 2dpf, 3dpf and 4dpf to observe whether the protocol or age of zebrafish was responsible for behavioural output.

Type of Test = Randomised Trials Protocol								
Light Baseline Percentage	**Baseline Duration**	**Startle Pulse Duration**	**Delay time between Prepulse and Startle Pulse**	**Prepulse Intensities Percentage**	**Type of Interval between two Pulses**	**Days Post Fertilisation (dpf)**	**Number of Runs**	**System Errors**
15%	30 minutes	40ms	20ms	30%, 40%, 50%, 60%	15%	2dpf	1	Bars appearing
1%	30 minutes	40ms	20ms	30%, 40%, 50%, 60%	1%	3 + 4 dpf	2	Bars appearing

3.8.2 Response Protocol

For observing behavioural response of zebrafish to light, the next protocol involved 100% light for 1 minute after 1h of darkness. This was applied when the zebrafish larvae were 4dpf in the first batch. This response protocol was performed to observe the difference in behaviour throughout the entire minute. Based on visual observation of video recording, an immediate motor response was depicted by zebrafish in the starting three seconds of the response protocol. This response gradually slowed down after 3 seconds in that 1 minute. To observe whether the response was based on providing light for a longer duration continuously or not, 1s and 3s intervals were taken as startle pulse time for other protocols. This movement was considered when image analysis was performed whether to regard it as a startle response or not.

3.8.3 Startle Response Protocols

Startle response protocols involved observing darkness and baseline intervals between two startle pulses. These protocols included startle pulses for 1s and 3s, and were performed separately, as shown in table 4. Since, bars appeared in protocols where 1% of light was provided between two pulses, protocols were developed including darkness intervals between two pulses as well. As the system error became a problem, not all output recordings were processed for image analysis. The videos that were used for image analysis involved the time frames of startle pulses only.

Table 4. Startle Response Protocols with 1s and 3s of Startle Pulses – The protocols developed for observing startle response in zebrafish are defined in table 4. Not all of the output recording videos of these protocols were used for image analysis.

Type of Test = Startle Response Protocols								
Light Baseline Percentage	**Baseline Duration**	**Startle Pulse Duration**	**Number of Startle Pulses**	**Interval between two Pulses**	**Type of Interval between two Pulses**	**Days Post Fertilisation (dpf)**	**Number of Runs**	**System Errors**
None	None	1s	200	Increasing 10s, 20s, until 200s	Darkness	4dpf	2	None
None	None	1s	200	Increasing 10s, 20s, until 200s	Darkness	5dpf	1	Light intensity not showing zebrafish
1%	30 minutes	1s	200	Increasing 10s, 20s, until 200s	1% Light	4dpf	1	Bars appearing
1%	30 minutes	1s	12	90s	1% Light	5dpf	1	Bars appearing
None	None	1s	6	90s	Darkness	5dpf	1	None
None	None	3s	6	90s	Darkness	5dpf	1	None

The output recordings of protocols without system errors shown in table 4, were used for image analysis. One startle response protocol involved 1s of startle pulse and increasing darkness intervals between two pulses, as shown in figure 22a. In the same manner, another protocol included 1s startle pulses and increasing 1% light intervals; however, the output video was not used for image analysis due to system errors.

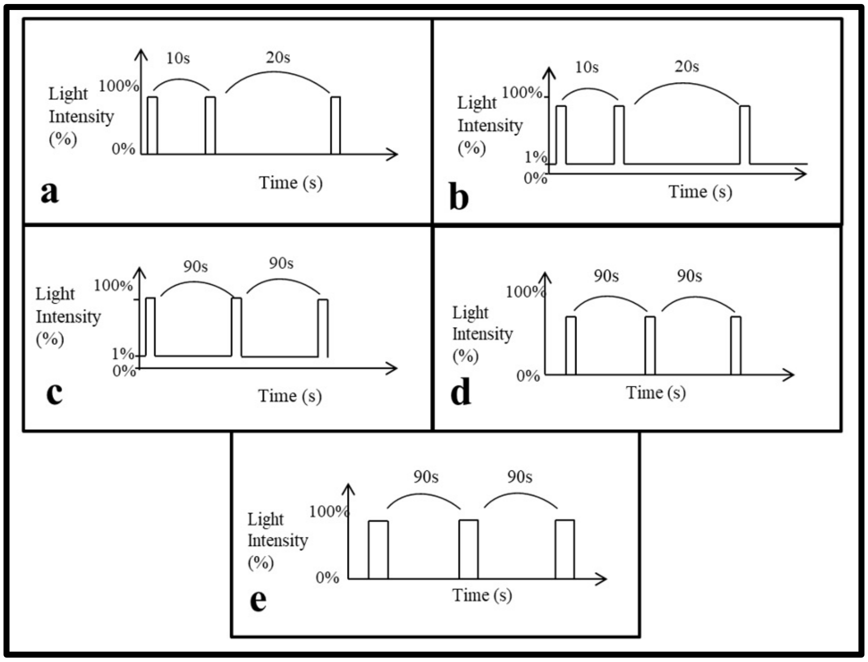

Figure 22. Graphical Representation of Startle Response Protocols – Protocol involved increasing darkness intervals between 1s startle pulses is shown in figure 22a. Protocol involved increasing 1% light intervals is shown in figure 22b. Protocol with 90s intervals of 1% light between two pulses is shown in figure 22c. Protocols with 90s darkness intervals between two pulses are shown in figure 22d and e, first one with 1s startle pule and the second with 3s startle pulse.

For the protocol with darkness intervals, it was performed 3 times, however the wrong Python program was loaded, so the output video consisted of light pulses flashing, but zebrafish was not visible. This is the reason why only two trials of this protocol were included in the results. These protocols were applied in the second batch of zebrafish. Based on visual observation of these two results, 90s darkness between two startle pulses appeared to show startle response in most of the zebrafish larvae. The next protocols involved 6 startle pulses with intervals of 90s darkness and were applied in the third batch, as shown in figures 22d and e. 6 pulse count was taken from the background study performed with mice.

3.8.4 Prepulse Inhibition (PPI) Protocols

In the second batch, a PPI protocol with 1s of startle pulses and a prepulse of 50ms duration and 50% intensity (amplitude) was developed, as shown in table 5 and figure 23a. The duration of prepulse was kept 50ms throughout all PPI protocols and was taken from a study performed by Bhandiwad et al., (2013). The 50% intensity of prepulse was also kept constant and taken from mice background study. The delay time between prepulse and pulse was increased starting from 200ms to observe its impact on zebrafish movement. The protocols are shown in figure 23b and c. The selection of delay time as 200ms and increasing was taken from a study conducted by Bergeron et al., (2015). It suggested that the range of 10 – 500 ms delay prior to a startling pulse would inhibit the startle response.

Table 5. Prepulse Inhibition (PPI) Protocols with 1s and 3s of Startle Pulses – These protocols kept prepulse intensity and duration constant. The delay between prepulse and pulse was increased to see zebrafish response.

<table>
<tr><th colspan="9">Type of Test = Prepulse Inhibition (PPI) Protocols</th></tr>
<tr><th colspan="9">Prepulse intensity = 50%   Prepulse duration = 50ms</th></tr>
<tr><th>Light Baseline Percentage</th><th>Baseline Duration</th><th>Startle Pulse Duration</th><th>Delay time between Prepulse and Startle Pulse</th><th>Interval between two Pulses</th><th>Type of Interval between two Pulses</th><th>Days Post Fertilisation (dpf)</th><th>Number of Runs</th><th>System Errors</th></tr>
<tr><td>1%</td><td>30 minutes</td><td>1s</td><td>200ms</td><td>10s</td><td>1%</td><td>4 + 5 dpf</td><td>2</td><td>Bars appearing in 1% light</td></tr>
<tr><td>None</td><td>None</td><td>1s</td><td>Increasing 200ms, 300ms until 1s</td><td>90s</td><td>Darkness</td><td>5dpf</td><td>1</td><td>None</td></tr>
<tr><td>None</td><td>None</td><td>3s</td><td>Increasing 200ms, 300ms until 1s</td><td>90s</td><td>Darkness</td><td>5dpf</td><td>1</td><td>None</td></tr>
</table>

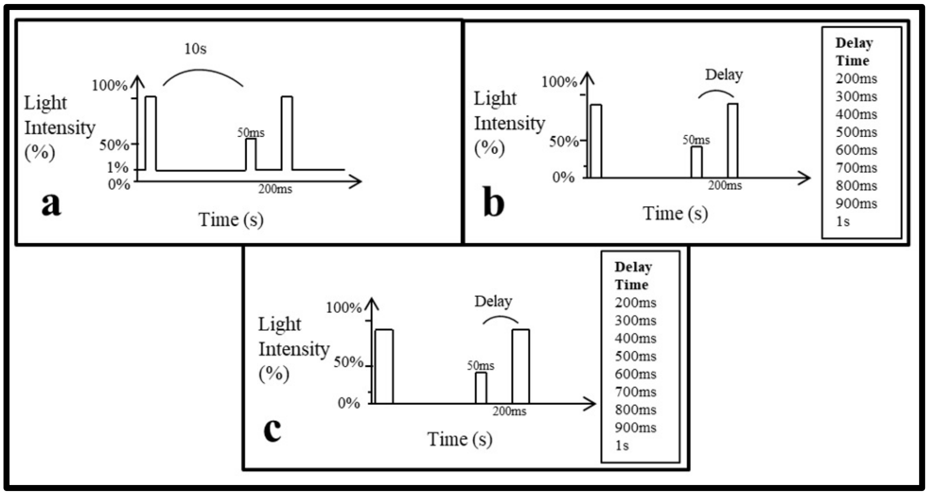

Figure 23. Graphical Representation of Prepulse Inhibition (PPI) Protocols – Protocol shown in 23a shows 200ms delay between prepulse and pulse. Protocols with decreasing delay times are shown in figure b and c. The startle pulse duration is 1s in 23b and 3s in 23b.

3.9 Data Collection and Analysis

As mentioned earlier, data were collected in the form of video recordings. These videos were converted into images through FFmpeg software. This software provides the ease of converting such files into different extensions. Further, ImageJ software was used to analyse the image data. The time frames of light pulses were selected by ImageJ, further creating stacks for each pulse. This way the difference between two pulses in the same protocol could be identified. The wells in those images were areas identified by drawing circular Region of Interest (ROI) for each stack, as shown in figure 24. The difference in mean pixel value between each frame showed whether the zebrafish moved or not. There were certain situations where zebrafish would drift and the software would consider that as a movement. These frames were discarded by setting a threshold of 0.5 mean pixel value. The software further analysed how many zebrafish larvae moved in total, basically identifying how many movements have occurred within how many wells. Then it provides a percentage of showing how many zebrafish larvae moved.

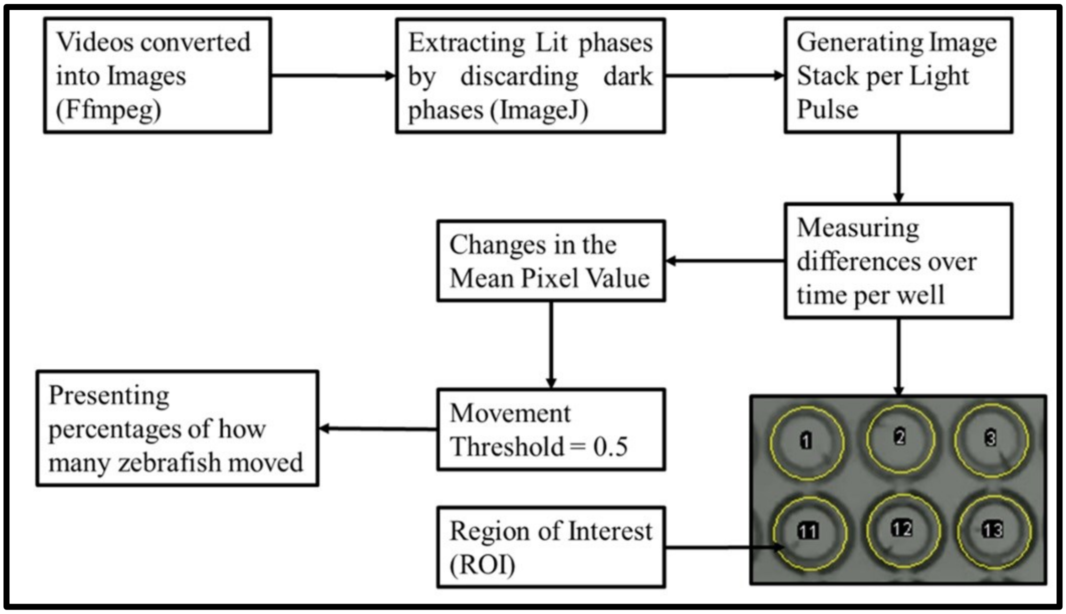
Videos converted into Images (Ffmpeg)
Extracting Lit phases by discarding dark phases (ImageJ)
Generating Image Stack per Light Pulse
Measuring differences over time per well
Changes in the Mean Pixel Value
Movement Threshold = 0.5
Presenting percentages of how many zebrafish moved
Region of Interest (ROI)
1
2
3
11
12
13

Figure 24 Process of Data Analysis – The h264 videos achieved from PiBoard were converted into images through FFmpeg. These images were processed in ImageJ. The software discarded the dark phases and made stacks of the lit phases. Then ROIs were drawn on each well and difference between frames were taken, providing the mean pixel value. These were further processed in resulting the number of zebrafish larvae that moved in percentage.

4 RESULTS

4.1 High Throughput (HT) System

The HT system was developed and scalability was achieved by Arduino microcontroller controlling three PiBoards and the light source. Zebrafish larvae were also imaged and their behavioural responses to light was observed. Figure 25 shows the scheme of the system and figures 26 and 27 show the outlooks of the system.

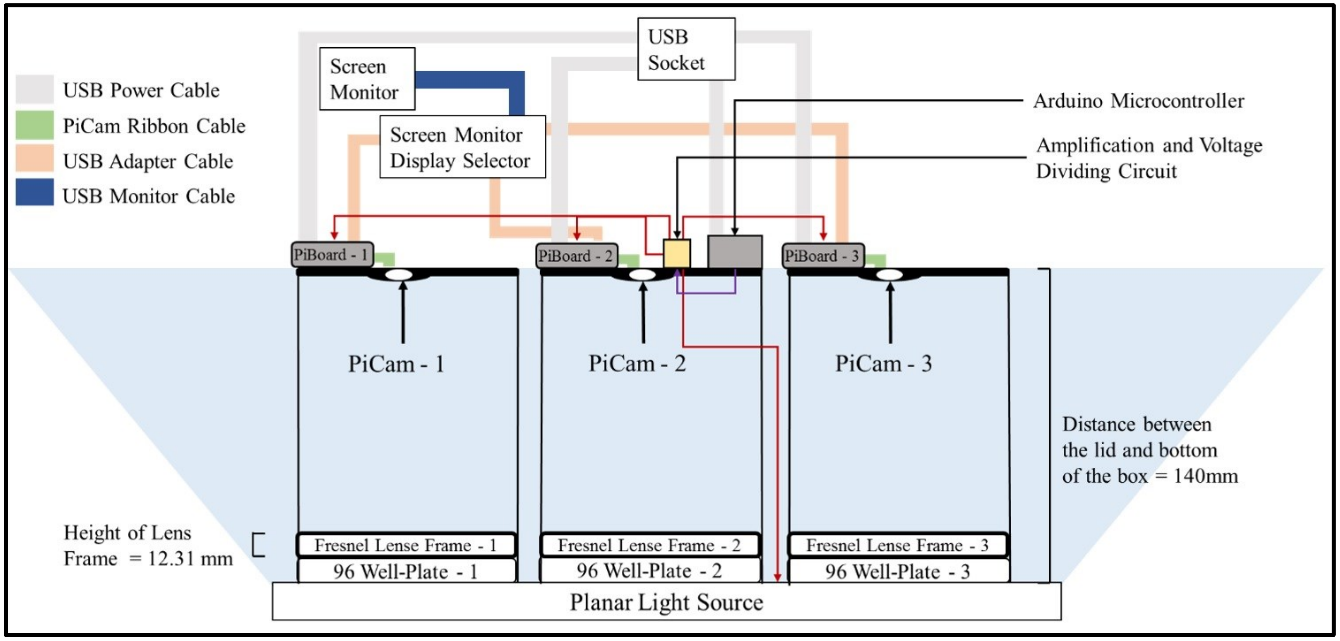

Figure 25. Output of the design developed for the HT system – The HT system showed scalability through Arduino controlling three PiBoards. PiBoards and Arduino were powered through USB socket. PiBoards are connected to a screen monitor display selector that is further connected to a screen monitor. The selector helps in selecting the view of any of the three PiCams. Arduino controls the light source and PiBoards shown in red arrows through the amplification and voltage dividing circuit. The signal coming from Arduino is 5V that can damage PiBoards, since they require the voltage to be 3.3V or less. This circuit decreases that 5V to be less than 3.3V for compatibility with the PiBoards. The circuit diagram is further elaborated in figure 28.

The outlook of the HT system is shown in figures 26 and 27. The circuit components were fixed on a plastic surface and placed on top of the cardboard box, as identified in figure 26. Figure 26 also shows the light source underneath the box. For conducting the zebrafish experiment, every time the plastic surface had to be removed so that the well-plate could be placed inside the box and on top of the light source. The surface was put back.

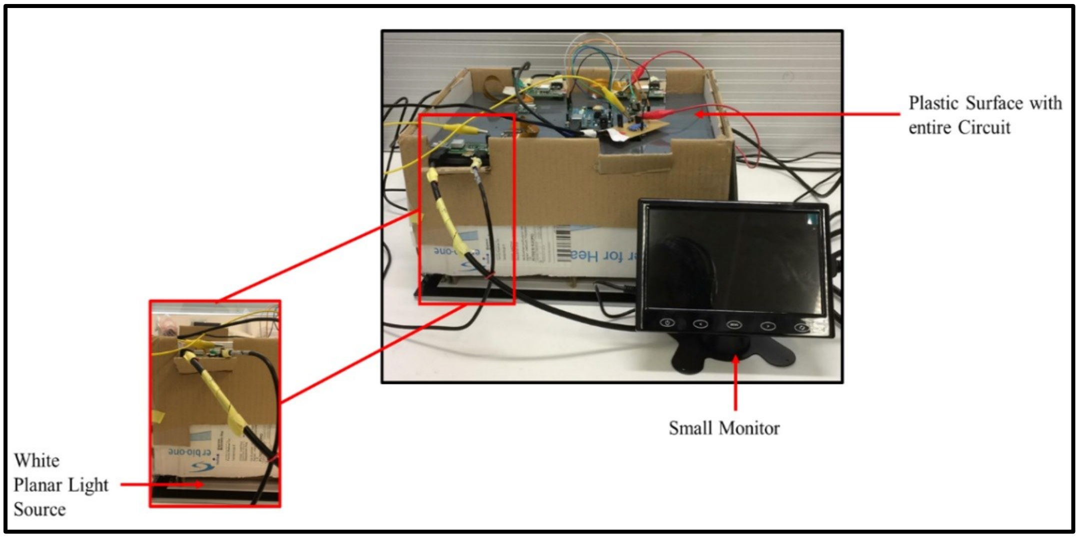
Plastic Surface with entire Circuit
Small Monitor
White Planar Light Source

Figure 26. HT System Developed for monitoring Zebrafish behaviour – The light source is underneath the cardboard box. Rest of the circuit components are present on top of the box. This figure shows the outlook of the system. The circuit board on top was removed and well-plate was placed inside the box for zebrafish experimentation. The box helps in blocking the ambient light.

Circuit components are clearly identified in figure 27. As discussed in the methods section, figure 27 shows the ribbon cable that connects the PiCam to the PiBoard has been tapered to PiBoard socket because of being loose. Fixing the circuit components on the plastic surface was challenging. This is because each component is powered using a USB cable that is long. USB cables connecting PiBoards to the monitor were not only long, but heavy too. This added more components to the system. It was essential that these components be screwed to the plastic surface because if PiCam is loosely connected on the surface, it could be misplaced even during zebrafish experimentation.

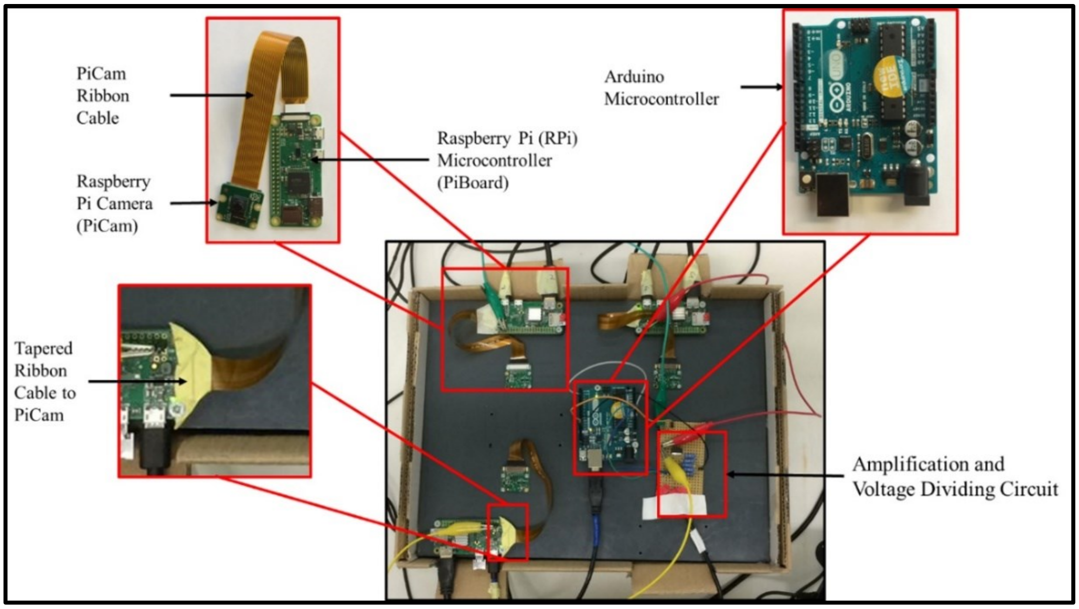
PiCam Ribbon Cable
Raspberry Pi Camera (PiCam)
Raspberry Pi (RPi) Microcontroller (PiBoard)
Arduino Microcontroller
Tapered Ribbon Cable to PiCam
Amplification and Voltage Dividing Circuit

Figure 27. Circuitry of the HT System – Arduino, PiBoards and PiCam were placed on a plastic surface and the entire surface was placed on top of the box. This figure shows the top view of the HT system, showing the placement of circuit components that are shown as a part of design diagram in figure 25.

As mentioned in methods section, in Python code for Arduino, the while loop logic was checking the state on port 2 that whether it was level 1 or 0. The program also included the pullup logic for port 2, this way it had level 1 state. The change from 1 to 0 state was done by providing a switch connection to the ground of the circuit, as shown in figure 28. This is because the Python code worked in a way that if port 2 stays 1, the code should keep on checking its state and if the state changes from 1 to 0, further commands should be executed (commands instructing to initiate lighting sequence and recording). This way the user managed when to start the lighting sequence and recording. Figure 28 also shows the circuit that divides the 5V from Arduino to be less than 3.3V for PiBoards. The resistances were set in a way that it lowers 5V to 2.5V, and since 2.5V is less than 3.3V, the resistance values shown in figure 28 seemed relevant here. In this case, internal resistances have been ignored, which is why the amplifying and protection circuit was used, as shown in figure 28. However, the resistance value used here was inaccurate which is why this part of the circuit was faulty.

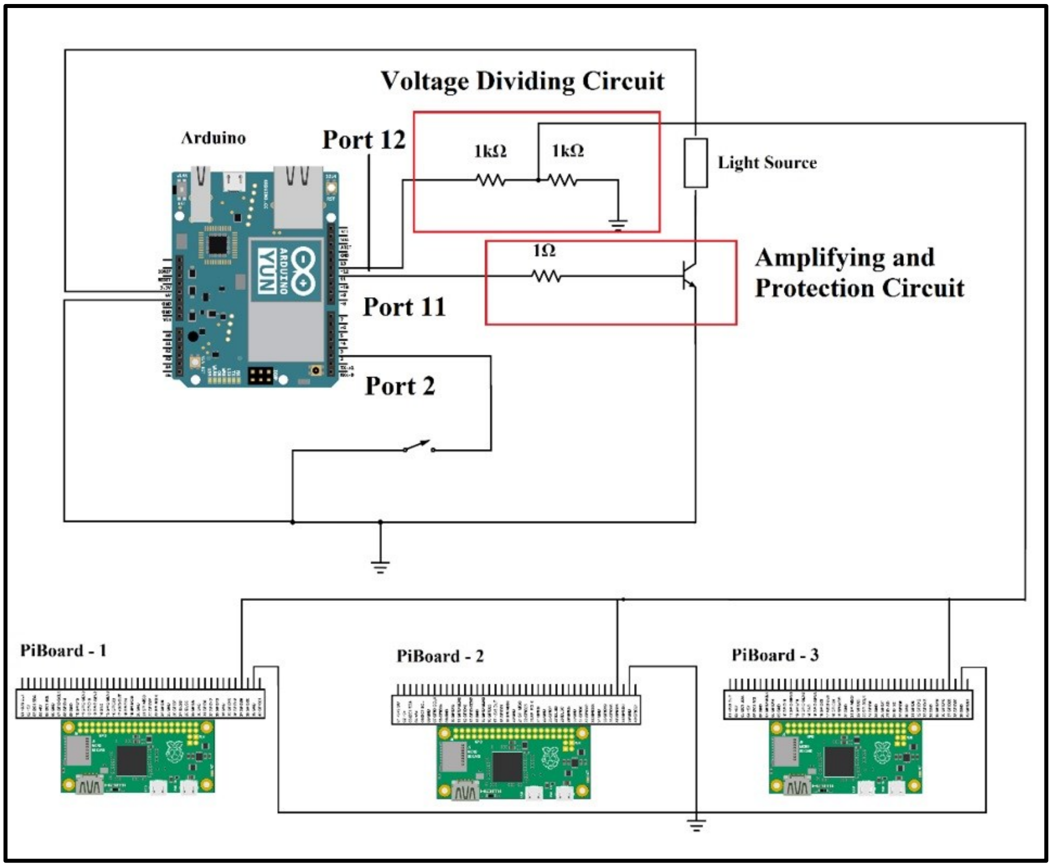

Figure 28. Circuit Diagram for Arduino and PiBoard – The circuit diagram for the system shows the port connections of Arduino and PiBoard. When port 2 receives ground connection, Arduino executes the command of sending signals from port 12 to the PiBoards and from port 11 to light source. This way, user can easily decide when to start the lighting sequence and recording at the same time. The lighting sequence included 1s of darkness interval at the beginning.

4.2 Optimisation of High Throughput (HT) System

The one-camera setup defined in the methods section was used to observe the entire 96 well-plate when PiCam was placed at a convenient distance (140mm). Figure 29 shows how the entire empty well-plate was visible in the one-camera setup.

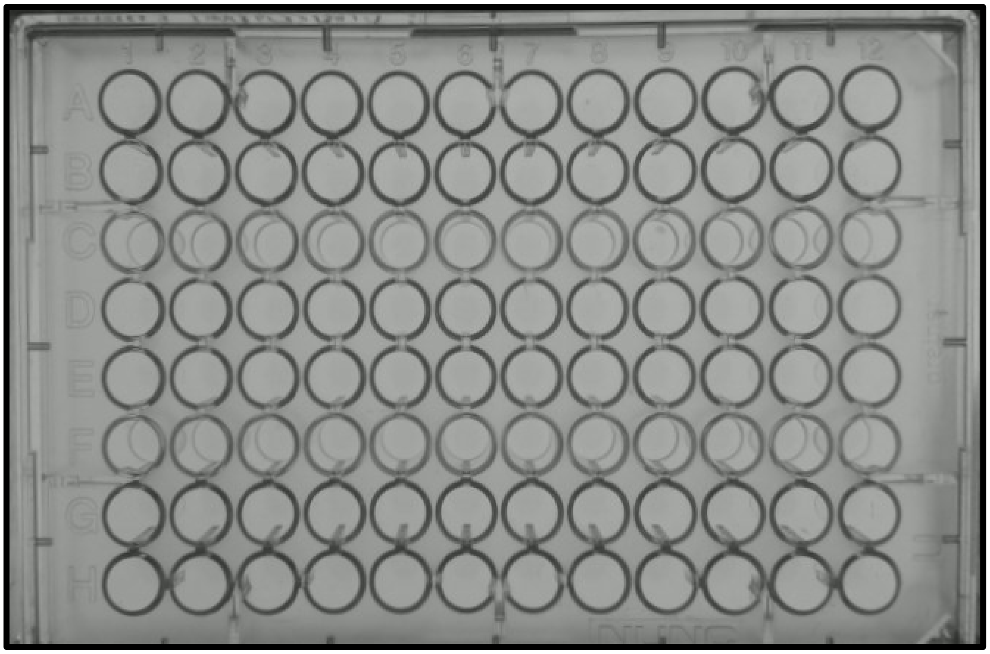

Figure 29. Image of an Empty 96 Well-Plate captured in the One-Camera Setup – For testing the one-camera setup, images of empty well-plate were taken to observe if the entire plate can be imaged by PiCam and if it fits in the FOV of PiCam. Figure 29 shows how the entire well-plate was visible by PiCam, when PiCam was at 140mm distance from the well-plate and how the plate appears in the setup.

Further considering figure 29, specific wells are identified from it and highlighted in figure 30. The wells shown in figure 30a show a parallax problem, which is not present in the middle wells, as shown in figure 30b.

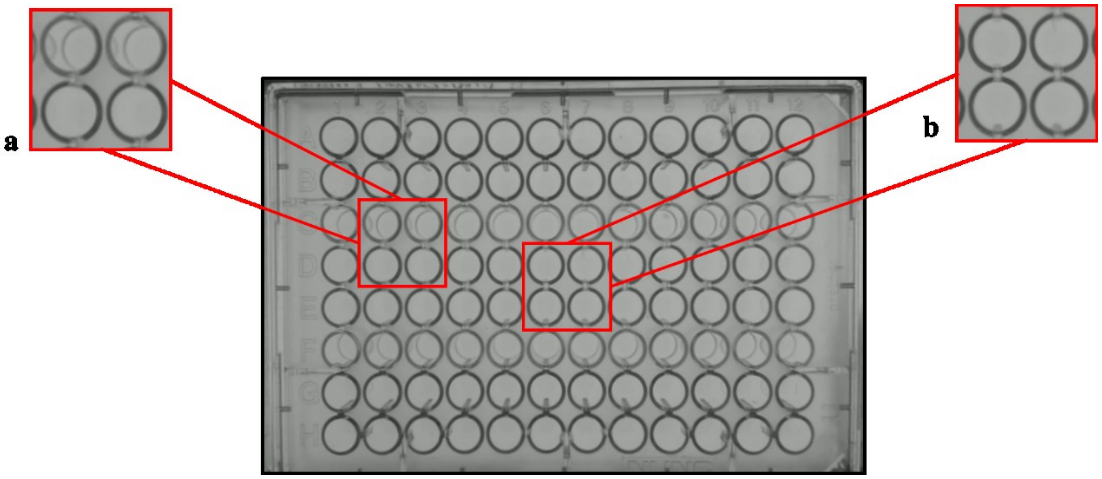

Figure 30. Comparing perspective of wells in the Image of an Empty 96 Well-Plate – This is the same image, as shown in figure 29 that was captured in the one-camera setup. Figure 30a shows the parallax problem and figure 30b shows that there is no parallax problem in the middle wells.

The parallax problem is mainly based on the positioning of each well with respect to the position of the camera. This is further explained optically in figure 31. The wells that are directly underneath the camera (middle wells shown in figure 30b) appear on the camera the way they are. This can also be seen in figure 31 that the light waves travelling from the well that is directly underneath the camera, do not bend. For the light waves coming from the well that is not underneath the camera, they bend around the edges of the well.

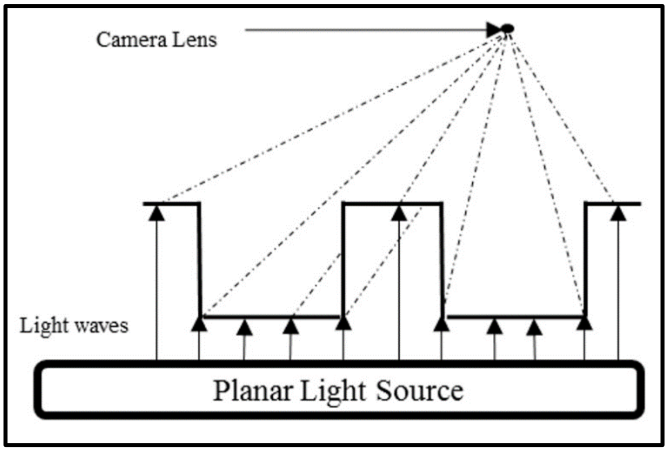

Figure 31. Optical Diagram for showing Parallax problem – The problem was observed when empty well-plates were imaged. Based on the position of each well, the perspective appeared to be different in the empty well-plate testing.

Figure 32 shows images of empty black and transparent well-plates that were captured in the one-camera setup. Regardless of what the colour of the well-plate was, a general parallax problem was still involved. Black well-plates make the parallax problem very visible comparatively to transparent well-plates.

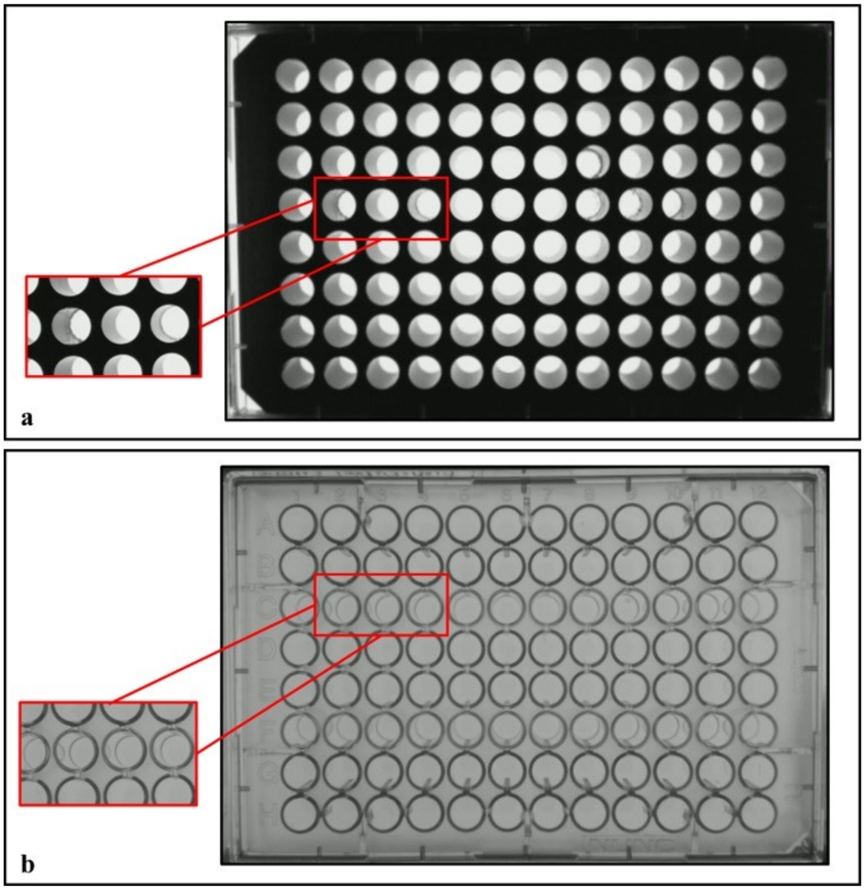

Figure 32. Empty Well-Plate Testing in the One-Camera Setup – The parallax problem appeared in which the bottom of every well was not visible. The position of some wells appears different when the empty well-plate was viewed from PiCam on top in the one-camera setup. Figure 32a shows a black 96 well-plate and figure 32b shows a transparent 96 well-plate.

This thesis proposed the use of a convex lens and placing it on top of the well-plate, as the solution for solving this parallax problem. This is because such lenses allow more of the field to be viewed in the image. This thesis used Fresnel lens. Figure 33 shows how this solution was implemented. The lens was simply placed on top of the well-plate. However, this did not completely solve the parallax problem.

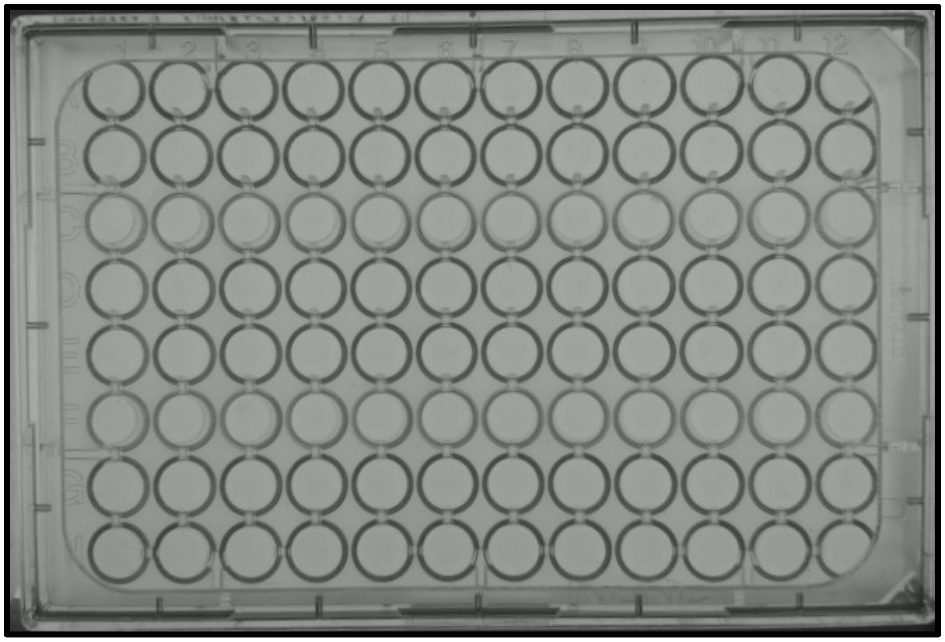

Figure 33. Fresnel Lens placed on a 96 Well-Plate – This image was captured in the one-camera setup by placing the lens on top of the lid of the well-plate. The rigid surface of the lens was facing towards the well-plate. To a certain extent, the parallax problem seemed to be solved; however, another option was to increase the distance between the plate and lens for better image of wells. The wells on the plate corners are not fully covered by the lens.

After placing the lens on top of the well-plate, as shown in figure 33, it can be observed in figure 34a that the parallax problem is still present in the targeted wells. However, in figure 34b the parallax problem seemed to be solved. To optimise lens positioning, it was fitted into a well-plate frame as mentioned in the methods section and is shown in figure 35. Using this frame, there was 12.3mm distance between the lens and well-plate.

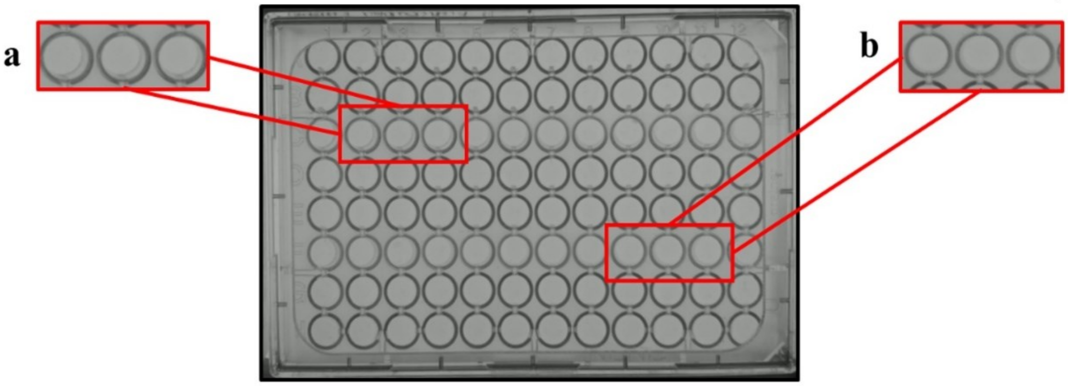

Figure 34. Fresnel lens solved the parallax problem for certain wells – Figure 34a shows that even though the lens is present on top of the plate, parallax error is still here. Figure 34b shows that this error has been improved by the lens. For lens optimisation, further steps involved increasing the distance between the plate and lens.

It can be observed in figure 35 that the frame is blocking the view of the wells present on the edges of the plate. With one frame, the distance between the lens and plate was 12.3mm. Using a frame helps in lifting the lens from the plate and increasing the distance here between lens and plate. 12.3mm was taken as an optimal lens displacement.

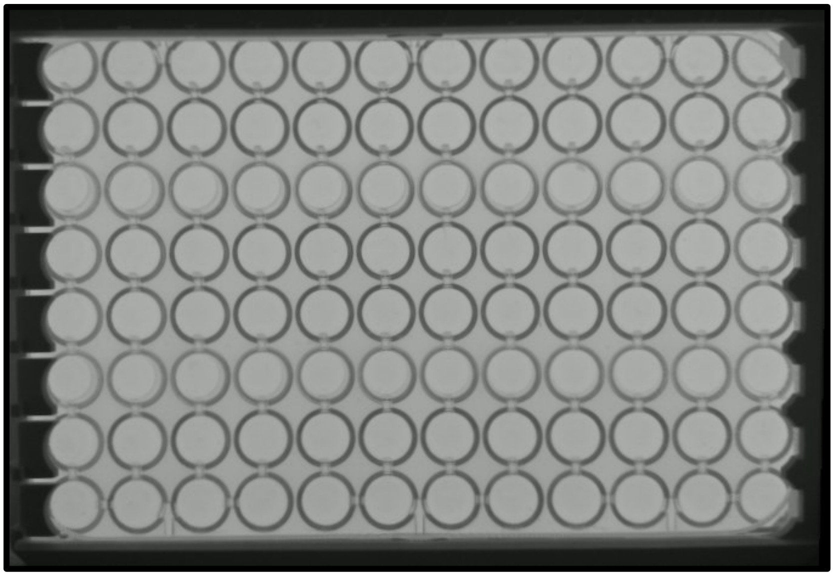

Figure 35. Placement of Fresnel Lens fixed in Frame on top of 96 Well-Plate – The frame consisting of Fresnel lens was easily placed on top of the plate. This way the distance between the lens and plate was adjusted easily and helped in improving the parallax problem.

The presence of liquid in wells would modify the visualization of well contents, which involved zebrafish in this case. This is because liquid has a different refractive index than air. This is the reason why not only empty well-plate, but well-plate with water were also tested for system optimisation, as shown in figure 36. The optimal distance 12.3mm was used for testing conditions shown in figures 36c and 36d. 300 µl well-plate was used in this project. For differentiating between empty well-plates and well-plates with liquid, 250 µl of water was used in each well for creating convex meniscus. For similar reason, the zebrafish experimentation also included 250 µl of E3 medium which was enough for zebrafish survival.

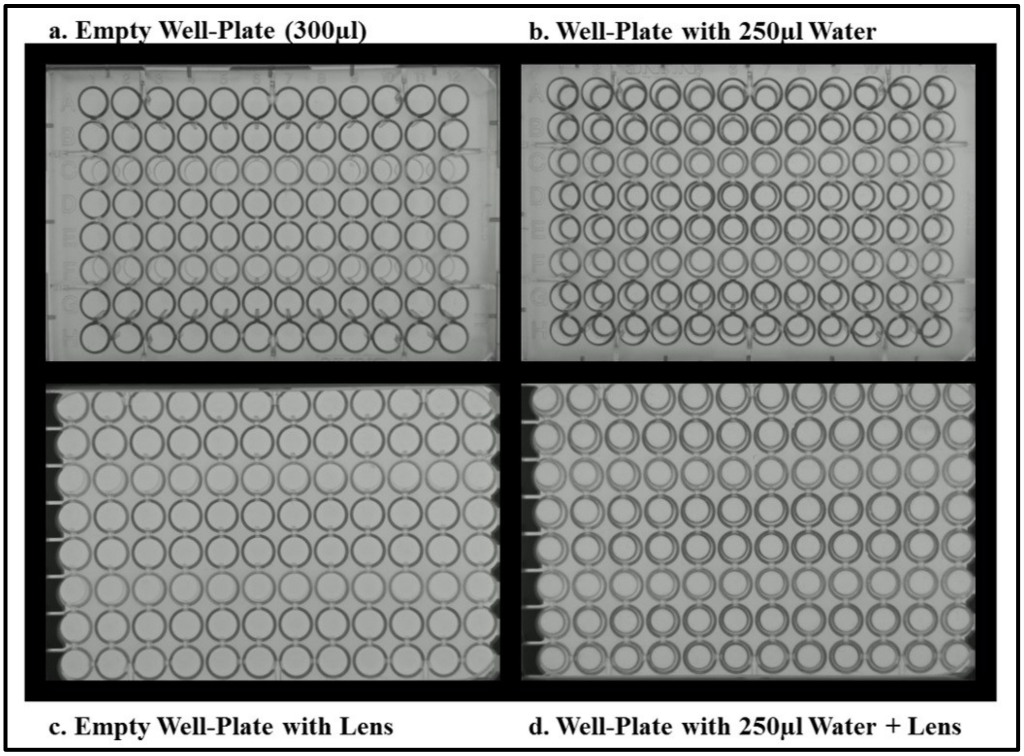
a. Empty Well-Plate (300µl)
b. Well-Plate with 250µl Water
c. Empty Well-Plate with Lens
d. Well-Plate with 250µl Water + Lens

Figure 36. Differentiating between conditions with and without liquid in wells and their appearance with and without lens – The reason to have this approach of comparing the defined conditions so it becomes easier which one to select for zebrafish experimentation. These conditions were further tested by increasing the distance between the lens and well-plate, and between light source and well-plate.

The difference between the well-plates with and without water is shown in figure 36. However, these conditions were further tested by increasing the distance between the lens and well-plate, and light source and well-plate. Figure 36 provides an idea how the well-plate appeared in the one-camera setup so the defined four conditions could be compared. Figure 36 further shows the difference in the perspective of wells when liquid was present in them.

Figures 37 and 38 show graphical representations of number of wells that display entire visibility of well bottom with respect to increasing distance between the well-plate and lens. This shows a comparison between the two types of well-plates, black and transparent. The thesis used the distance 12.3mm because when zebrafish experimentation was carried out, this distance was tested initially, and then the next distance tested was 24.6mm, as shown in figure 37. The camera preview showed that the zebrafish larvae were visible when the lens was 12.3mm away from the well-plate.

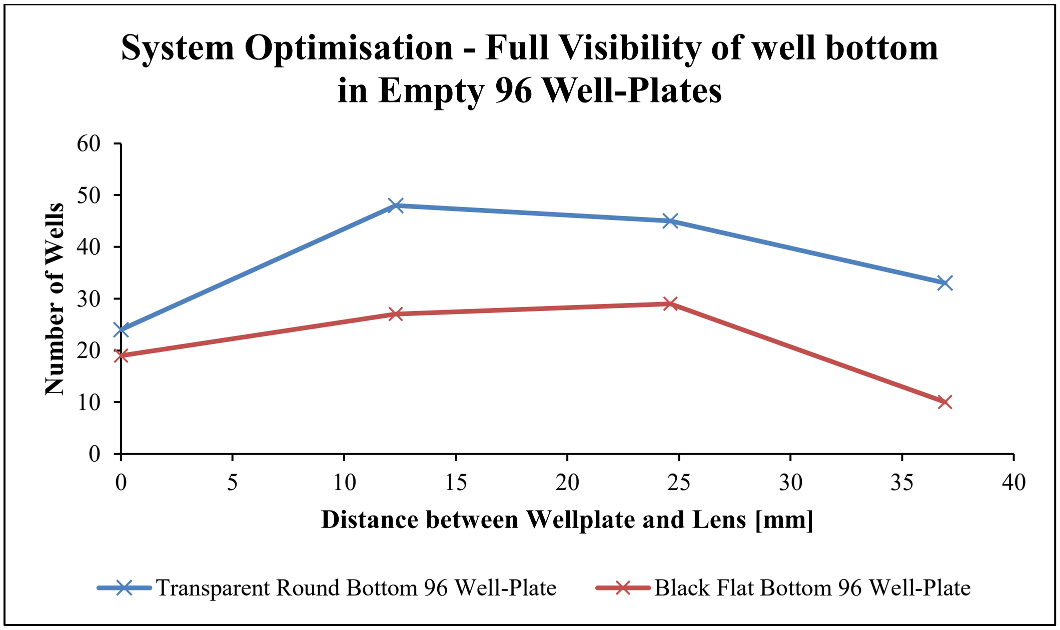

Figure 37. Number of wells and increase in distance between lens and empty well-plate – 12.31mm was selected as the distance between lens and plate. The number of wells were counted based on the entire visibility of the well bottom. In general, the bottom of the well can be seen, but due to the parallax problem the entire bottom is not visible in certain wells.

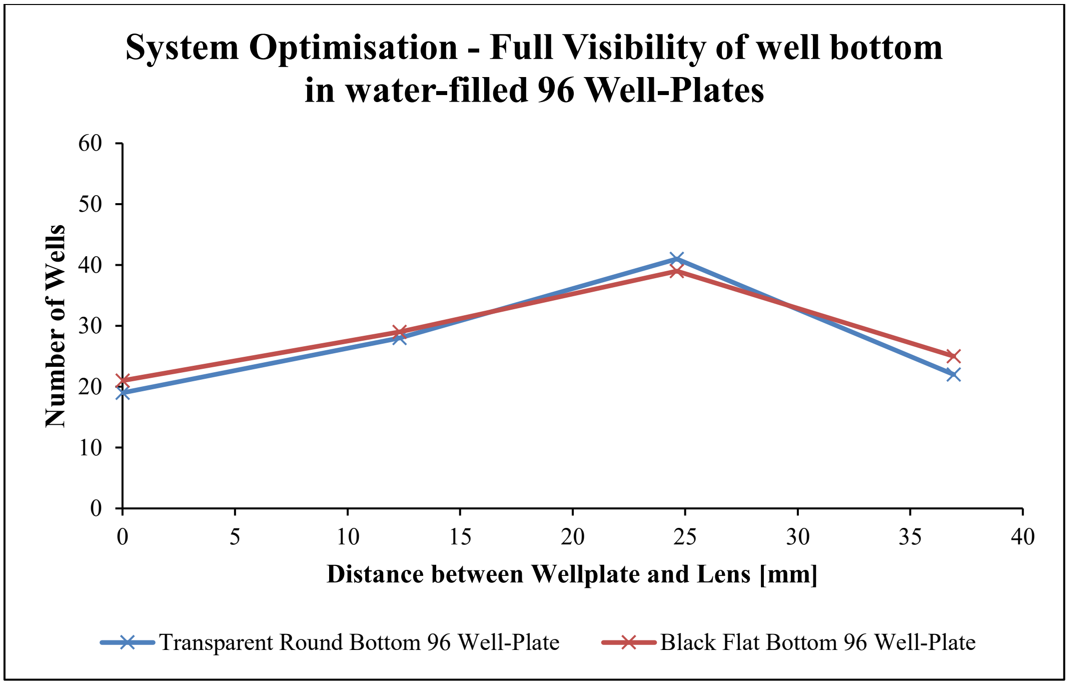

Figure 38. Number of wells and increase in distance between lens and water-filled well-plate – The presence of liquid along with the lens did help in improving the parallax problem, but since the zebrafish larvae were visible when the distance was 12.31mm, the highest distance here was not considered.

4.3 Developing Method for Image Analysis

As mentioned in the methods section, ImageJ was used for analysing data in this thesis. For different lighting protocols, the h264 videos were converted into images using FFmpeg software. There were 30 frames per second (fps). Figure 39 shows the visual representation of the image data that were acquired for this thesis. Figure 39 shows the 1st frame of the first light pulse in the lighting protocol where 90s of darkness intervals were provided between two 1s light pulses. It also shows how the image data visually appeared in this thesis. Even though 30fps were selected, the image data consisted of 28fps for lit frames. 9 wells from the same frame are shown in figure 39 that have been considered to show image analysis in this section, as shown in figure 40.

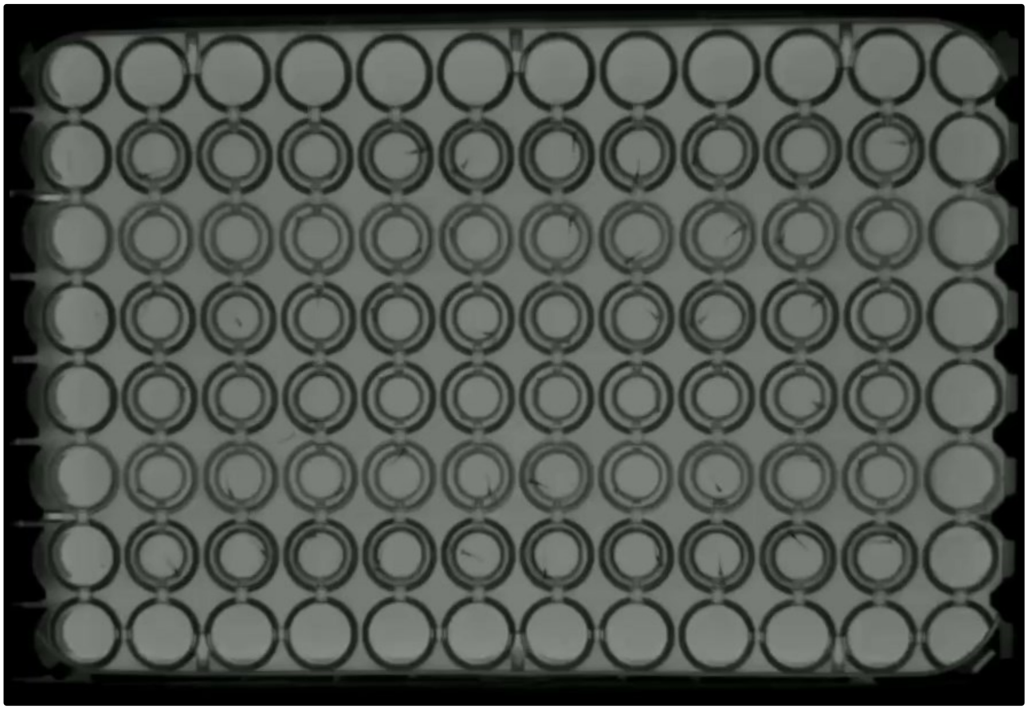

Figure 39. Representation of the Image Data and 1st Frame of first Pulse of 1s provided in lighting protocol – This image is the first frame of first light pulse in the protocol where 90s of darkness was present between two 1s pulses. Zebrafish can be observed in the well-plate. Notice how the wells present on the edge of the plate have not been used for zebrafish experimentation.

Two different frames are highlighted for same well in figures 40 and figure 41. These are from the same lighting protocol where 90s of dakrness was provided between two pulses of 1s, as described for figure 39. The frames are of the first pulse, 1st frame shown in figure 40 and 28th frame shown in figure 41.

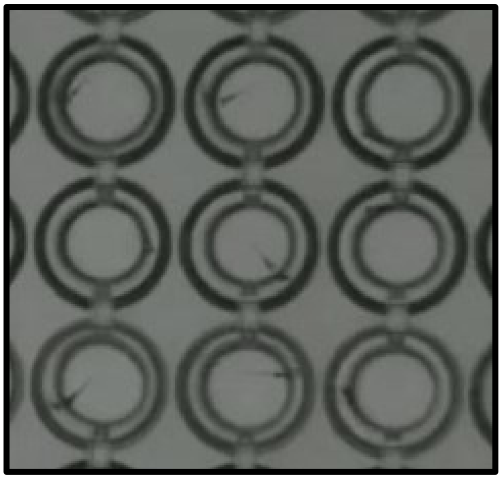

Figure 40. 9 Wells for Image Analysis from 1st Frame – These are the selected Wells from frame 1 of first light pulse provided in the protocol where 90s of darkness was present between two 1s pulses.

The wells from both frames shown in figures 40 and 41 can be visually compared. In order to quantify the change of position of zebrafish in these wells, these frames were subtracted from each other in ImageJ.

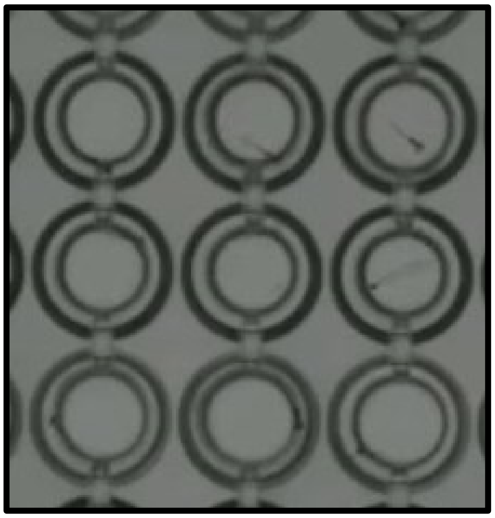

Figure 41. 9 Wells for Image Analysis from 28th Frame – These are the selected wells from frame 28 of first light pulse provided in the protocol where 90s of darkness was present between two 1s pulses.

Before performing the subtraction operation, the wells were targeted by drawing circular ROIs around them in ImageJ, as shown in figure 42. These were saved as zip files. If for certain frames, there were system errors and ROIs were not aligned, those frames were not used for image analysis. In some cases, ROIs were drawn separately for image data of each lighting protocol. This is because ROI of one set of image datum did not align with another set of image datum belonging to a different protocol.

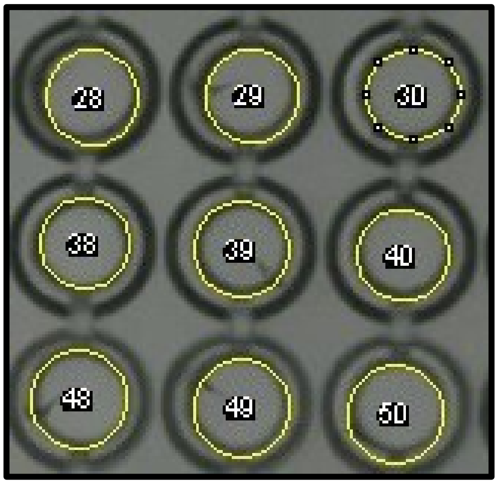

Figure 42. Circular ROIs for wells – These ROIs were drawn for the image data on ImageJ. This way changes occurring in these ROIs can be processed by ImageJ as zebrafish movement.

After drawing the ROIs, both frames were subtracted from each other. This subtraction image can be observed in figure 43. The changes in wells 29, 39 and 48 in figure 43 show that there were rapid movements of the zebrafish. Figure 43 can be cross referenced with figures 40 and 41. Well 38 shows how there was no movement in both frames. The measurements of changes shown in the subtracted image in figure 43, were acquired as changes in the mean pixel values of the image data. For each light pulse, these values were processed that if such changes in a well would occur, ImageJ would count the number of wells and present it as a percentage of how much zebrafish have moved in that specific light pulse. The percentages for each pulse in an applied lighting protocol are graphically represented further in this section.

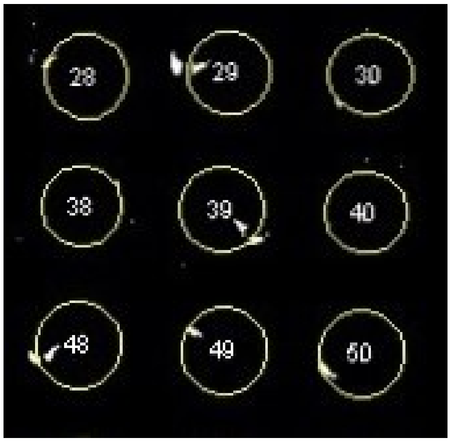

Figure 43. Subtraction Image for frames shown in figures 40 and 41 – This is the subtracted image for frame 1 and 28 as identified in figures 40 and 41.

However, for certain frames in the light protocols, there were system errors that were picked up as changes by ImageJ, as shown in figure 44. These errors were if someone was working on the table on which the system was placed, or if the user touched the system by accident and as such strange disturbances.

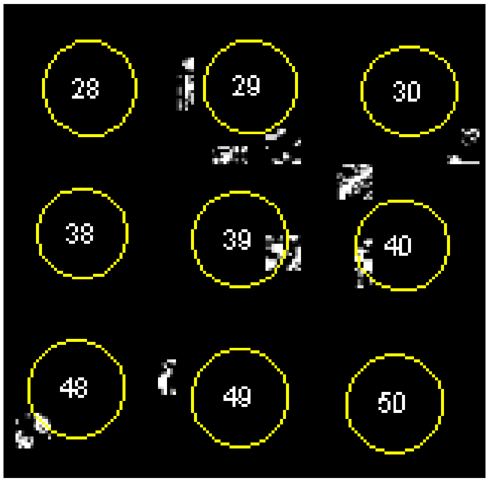

Figure 44. System Errors picked up as Zebrafish movement by ImageJ – The errors caused the system to recognize changes in ROIs as zebrafish movement.

The zebrafish would also drift away. Therefore, to identify what a movement is, the changes in the mean pixel values were observed and a threshold of 0.5 was set. If changes were 0 or more than 0.5, these were not regarded as movement by the software. Movement of zebrafish from one point to another is regarded as a startle response to light pulses in this thesis.

4.4 Zebrafish Responsiveness Time Interval

In the protocol where the light was on for one minute, based on visual observation of the video recording, zebrafish showed rapid movement that lasted for 3s. A gradual decrease in the rapid movement was further observed after 3s. A lower level of 1s could be selected as the interval for light pulse; however, the pulse was increased up to 3s in order to image during this time and observe zebrafish behaviour. This is the reason why protocols have 1s and 3s light pulses.

Table 6. Selection of Time Interval for Light Pulses – The protocol in which light was on for a minute, it showed zebrafish rapid movement that lasted for 3s. This was done to develop time intervals light pulses for further lighting protocols.

Type of Light Pulses	**Purpose**	**Interval**
Shorter duration	-To induce a startle response -Imaging zebrafish behaviour for short duration	1s
Longer duration	-To induce a startle response -Imaging zebrafish behaviour for longer duration	3s

4.5 Startle Response

For each lighting protocol, ImageJ presented a percentage of how much zebrafish moved in each light pulse. These are graphically represented in this section. For observing startle response in zebrafish, a lighting protocol of 1s light pulses was provided. There were darkness intervals between two pulses. The darkness intervals between the first two pulses was 10s, then between 2nd and 3rd pulses, the darkness intervals were increased to 20s and then 30s between 3rd and 4th pulses, and so on. This darkness intervals were increased until 200s between the last two pulses of the protocol. Figure 45 shows the percentage of moved zebrafish during each pulse in this protocol (Trial 1).

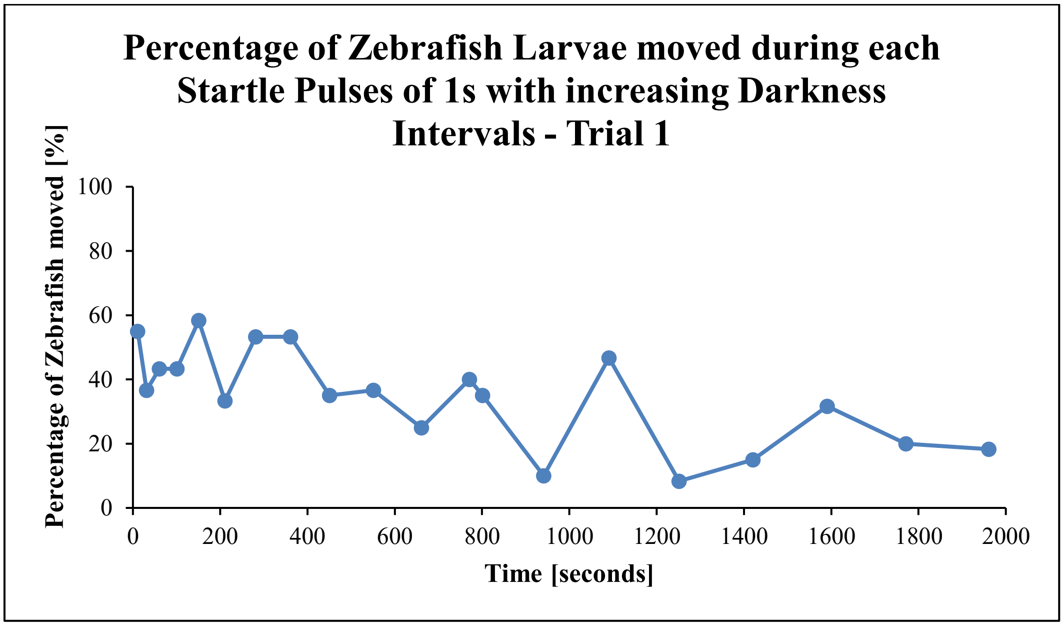

Figure 45. Increased Darkness Interval Light Protocol showing percentage of moved zebrafish in each pulse (Trial 1) – This lighting protocol started with 10s of darkness intervals between the 1st and 2nd light pulses, and further darkness intervals increased to 20s between 2nd and 3rd pulses and so on. The percentage here represent how many zebrafish moved in each light pulse. This movement refers to zebrafish larvae moving from one point to another in a well in 1s light pulse.

Considering the first pulse in figure 45, 55% of zebrafish moved during this pulse. Based on ImageJ analysis described above, this movement can be regarded as a startle response. As darkness intervals were increased, a decrease can be observed in the number of zebrafish larvae showing startle response over time. They might have habituated to the lighting condition in the applied protocol in figure 45.

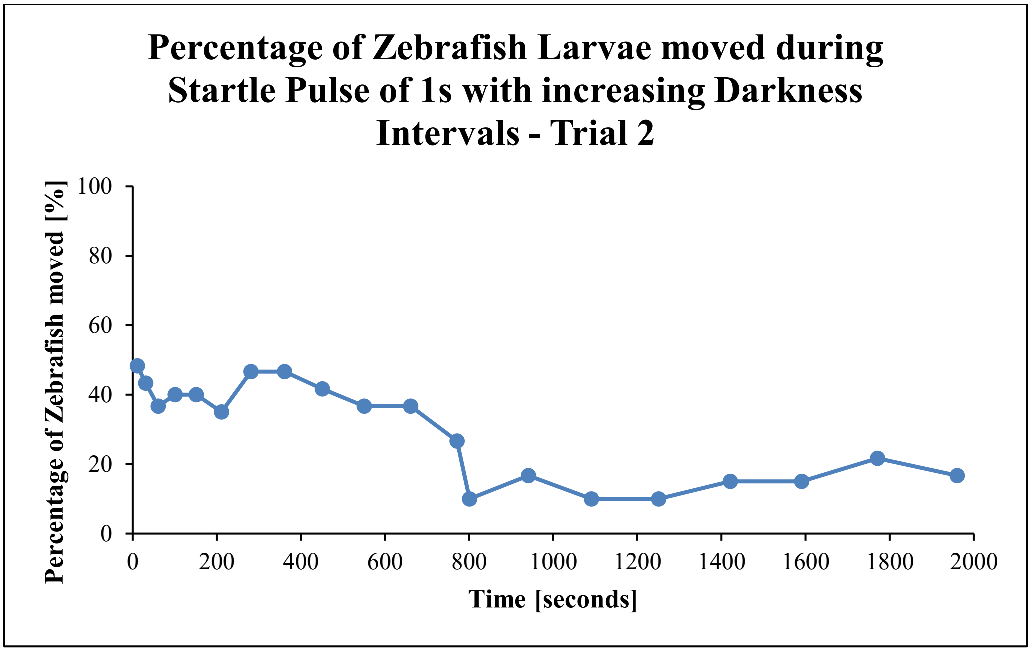

Figure 46. Increased Darkness Intervals Light Protocol showing percentage of moved zebrafish in each pulse (Trial 2) – This is trial 2 for the same lighting protocol as discussed for figure 46. It shows the percentage of zebrafish moved during each light pulse.

Trial 2 was performed for the same protocol and figure 46 shows the percentages of zebrafish larvae moved in each light pulse. 43.33% of zebrafish larvae exhibited startle response in the first light pulse, as shown in figure 46. A decrease in the number of zebrafish larvae exhibiting startle response over time is also observed in figure 46. From both trials, 90s of darkness intervals between two light pulses was selected as appropriate darkness periods for showing startle response in zebrafish. In figure 45 and 46, the pulse after the 90s darkness interval appears at 451s, showing 35% and 41.33% startle response in zebrafish larvae respectively.

Two more protocols for observing startle response were applied, in which 90s darkness intervals between two pulses was the same, but the duration of light pulse was 1s in one protocol and 3s in the other protocol. The percentage of zebrafish larvae exhibiting startle response for these protocols is shown in figures 47 and 48. In both figures it can be observed that during the first light pulse, more than 90% of zebrafish larvae exhibited startle response.

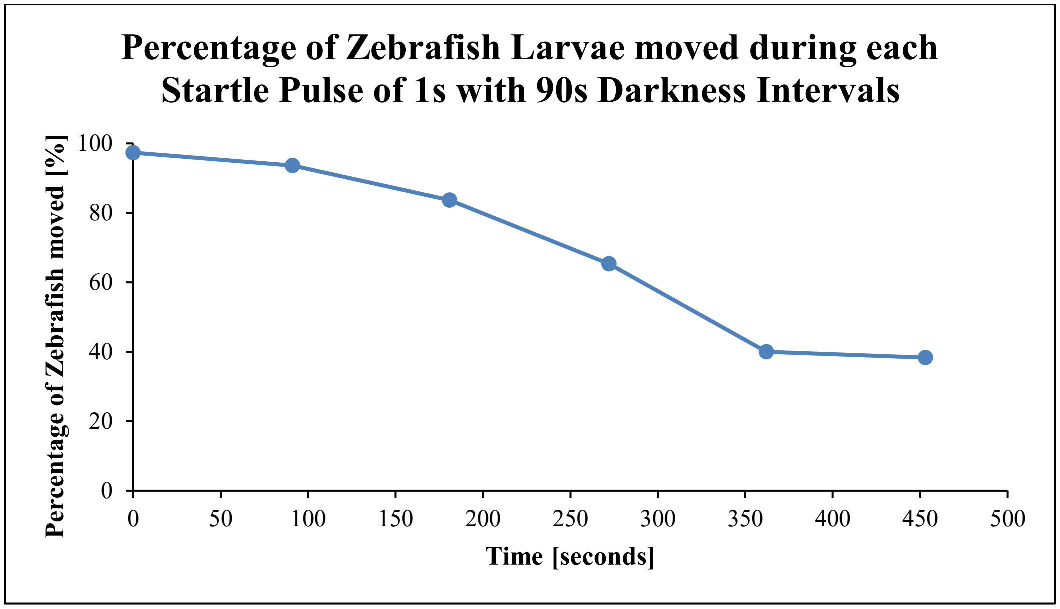

Figure 47. 90s Darkness Intervals between two Light Pulses of 1s – The first pulse shows the highest percentage of zebrafish larvae exhibiting startle response. The lighting protocol used here consisted of 6 pulses, and the darkness intervals between two pulses are 90s.

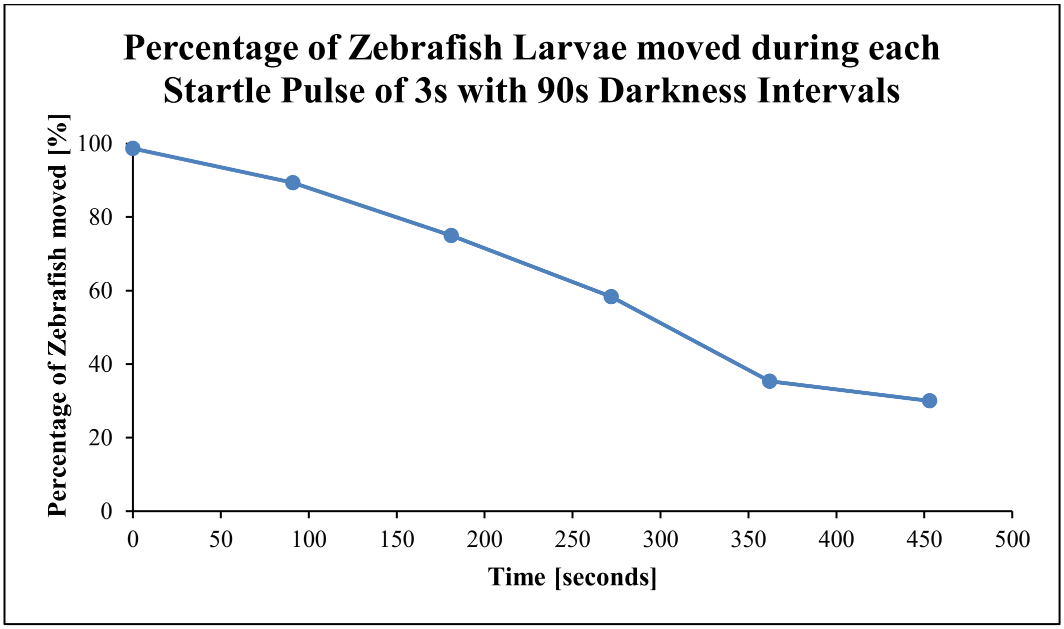

Figure 48. 90s Darkness Intervals between two Light Pulses of 3s – The first pulse here also shows the highest percentage of zebrafish larvae exhibiting startle response. The lighting protocol used here consisted of 6 pulses, and the darkness intervals between two pulses are 90s.

Based on the percentage of zebrafish larvae startling in the lighting protocols described in this section, this proves the first hypothesis of this thesis that larval zebrafish can exhibit startle response within 5dpf in the developed HT System. The startle response was observed in 4-5 dpf zebrafish larvae.

4.6 Prepulse Inhibition (PPI)

Light intensity of prepulses in PPI protocols for this study was 50%. The startle pulses are of 100% light intensity and induced startle responses in zebrafish larvae as per previous results. Prepulse is supposed to be a weaker pulse that would not initiate a startle response, which is why 50% was selected for this study. This prepulse intensity was kept constant. The PPI protocol that had 1s startle pulses, there was darkness intervals of 90s between two pulses, and prepulse was presented 200ms before the 2nd pulse, as shown in figure 23b. This delay was changed to 300ms prior to 3rd pulse and the cycle went up to 1s delay between the last prepulse and pulse of the protocol. The percentage of zebrafish movement in the startle pulses that came after the prepulses are shown in figure 49.

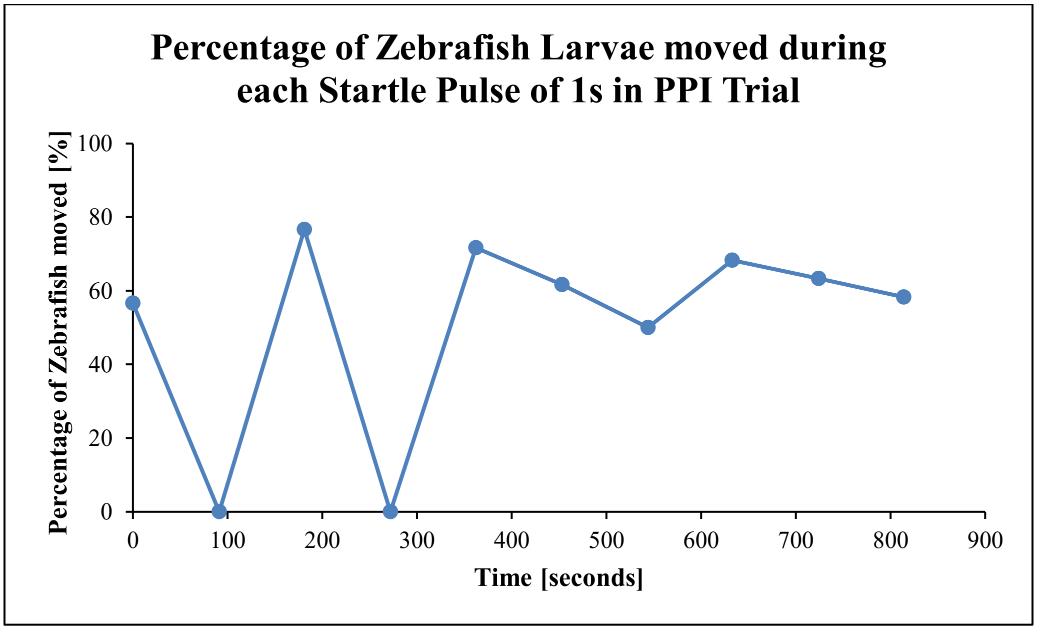

Figure 49. Percentage of Zebrafish Larvae moved during each Startle Pulse (1s) that came after the prepulse in PPI Trial – The PPI protocol consisted of 50% prepulses. The delay between prepulse and startle pulse was increased from 200ms to 1s, as shown previously in figure 23b. The percentage of moved zebrafish larvae in the pulses that came after prepulses are shown here.

The second PPI protocol was the same, only having 3s pulses instead of 1s. The percentage of moved zebrafish larvae the startle pulses that came after the prepulses in the second protocol are shown in figure 50. The presence of a weaker pulse inhibits the startle response when the pulse is provided after the prepulse. Figures 49 and 50 show that in each pulse, zebrafish exhibited startled response, depicting that zebrafish did not exhibit PPI in the HT system. However, for 2nd and 4th pulse in figure 49, it shows that 0% of zebrafish larvae moved and when these time frames were cross referenced, the zebrafish were moving. When the image data for these two PPI protocols were acquired, there were certain frames of the prepulses that were stacked as lit phases by the developed ImageJ method. For image analysis, these frames were discarded because these were blurry lines on the time frame. However, for 2nd and 4th pulses, the prepulse frames that before these pulses were included by accident in image analysis.

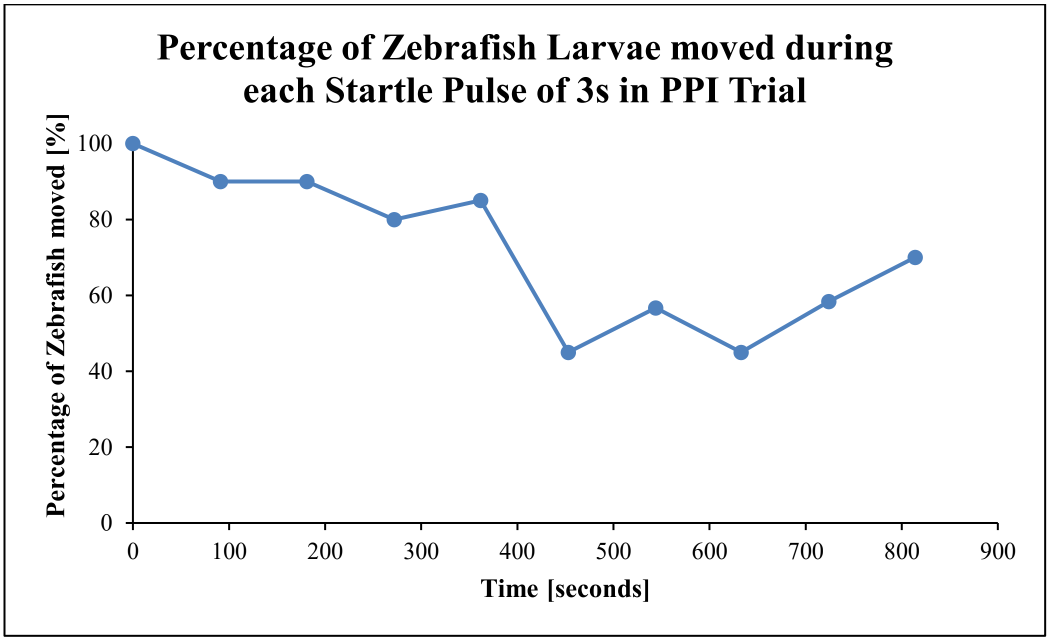

Figure 50. Percentage of Zebrafish Larvae moved during each Startle Pulse (3s) that came after the prepulse in PPI Trial – The PPI protocol also consisted of 50% prepulses. The delay between prepulse and startle pulse was increased from 200ms to 3s, as shown in figure 23c. The zebrafish that moved in the pulses that came after prepulses; their percentages are shown here.

The PPI protocol that consisted of two startle pulses of 1s and darkness intervals of 10s, the prepulse was provided 200ms before the 2nd pulse. Figure 51 does show that the zebrafish did not move in the second startle pulse and this was also cross referenced with the image data. The zebrafish did not move in the 2nd pulse. However, the prepulse was of 1s, which is not the appropriate interval for a weaker pulse. This is the reason why this developed PPI protocol was incorrect.

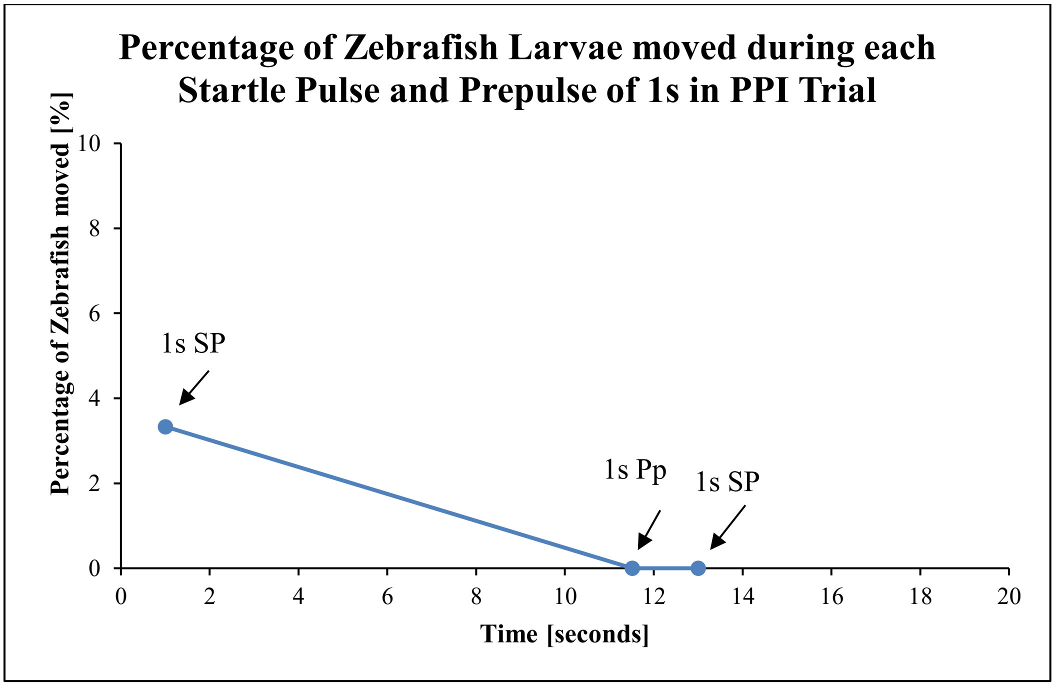

Figure 51. Percentage of Zebrafish moved during each Startle Pulse and Prepulse of 1s in PPI Trial – This protocol consisted of two startle pulses of 1s and there was a darkness interval of 10s between these two pulses. The prepulse was provided for 1s, 200ms prior to the 2nd pulse.

There are no sufficient data to indicate whether the larval zebrafish exhibited PPI or not within 5dpf in the HT System.

5 DISCUSSION

Behavioural responsiveness of zebrafish has made it a useful animal model, in particular for studying neurological phenomena such as startle response and PPI. This also indicates their importance for modelling neurological disorders. Challenges are present in the treatment and diagnosis of neurological disorders, which makes it difficult to identify their main cause. Animal models can be useful when studying specific perturbed behaviours associated with neurological disorders, even though they will not replicate all or even most properties of these disorders. By understanding startle response and PPI in zebrafish as an animal model, a better understanding of human startle response and PPI can be developed. Since, PPI is deficit in schizophrenic patients, studying it in animal models would also help understand the disorder itself (Howe et al., 2013).

The aim of this study was to develop an HT system for recording response to light stimuli in zebrafish. This was achieved with the use of an Arduino microcontroller to synchronise a PiBoard controlling a camera with a light source. Previous studies have focused on developing experimental setups to observe the response of zebrafish to either light, vibrational or acoustic stimuli (Legradi et al., 2015; Rihel et al., 2010). This thesis shows the experimental setup by demonstrating that it may be feasible to develop an HT system with the use of cost-effective components. The synchronisation between the microcontrollers allows automated starting of an illumination protocol together with recording of the zebrafish behaviour to light.

Certain lighting protocols consisted of changing bars in the videos, identifying that there are flaws in the design of the HT system. The data acquired from those light protocols are not processed through the image analysis method that is developed on ImageJ in this thesis. The method compares the image data frame by frame of wells. If it processes image data that consist of changing bars in each frame, the bars will overlap with the movement by zebrafish in the wells. The method can interpret that as a change of mean pixel values in the wells and can provide false results regarding the movement of zebrafish. In the case of low light, those bars were visible by the PiCam because instead of applying a constant but reduced current, the light is pulsed to reduce its intensity through PWM. Another reason for these bars appearing in the videos is interference between the Arduino

PWM frequency and the PiCam acquisition frequency. The moving bars in the videos prevented automated analysis of the data using the image difference approach. The correct method, simply applying a bias so that a low current flows to the illumination unit between bright light pulses, should be used in the future.

This problem can be avoided by using a limited constant current supply to the light source, for example a voltage divider, instead of using PWM for limiting the brightness in a time-variant way. Imaging not only during a light pulse but also during the dark phase is equally important. Dark phase refers to the state of darkness. When an organism is placed in a dark phase and then a sudden stimulus is applied for an interval and then it enters the dark phase again, the response to the stimulus could appear within the latter dark phase (Ioannidis et al., 2014).

The rapid response of zebrafish to the lighting protocol in which the light was on for 1 minute, lasted for 3s. After this there was a gradual decrease in the rapid movement of the zebrafish. Previous studies (Ioannidis et al., 2014; Rihel et al., 2010) have provided pulse intervals in milliseconds. In this study, the reason for increasing the pulse up to 3s was to image during this time and observe the response of the zebrafish. This is the reason why a lower interval of 1s and a higher interval of 3s were chosen for the startle response protocol, as shown in table 6 of the results section. In one of the protocols there are varying darkness intervals between two light pulses. There is 10s of darkness between the 1st and 2nd light pulse, then it increases to 20s between the 2nd and 3rd light pulse, 30s between the 3rd and 4th light pulse and so on; each pulse is 1s. The darkness intervals increase up to 200s between the last two pulses of the protocol. This protocol was repeated twice, showing the percentage of moved zebrafish in figures 45 and 46.

The results in figure 45 show that approximately 55% of the zebrafish moved during the first light pulse. This shows that the zebrafish exhibited startle response in the developed HT system. Comparing the percentages in figure 45 from the 1st pulse to the last pulse, there is a decrease in the percentage of startled zebrafish. This could indicate that, as the darkness interval was increased between two pulses, the zebrafish larvae were not exhibiting startle response. They could have become habituated to the lighting protocol and 1s of light stimulus. When 1h of darkness is provided, the zebrafish larvae did startle during the first pulse, as shown in figure 45. This means that when after a long and continuous darkness period zebrafish larvae encounter light stimulus, they show startle response. However, when two light pulses with a certain darkness interval are provided, and the 2nd pulse is followed by an increase in the darkness interval and then a 3rd light pulse is presented, zebrafish become habituated.

When videos were cross referenced in this case, it appeared that almost all of the zebrafish larvae moved at the first pulse. However, in order to obtain convincing results, it is recommended that the startle response trial of having different darkness intervals between light pulses should at least be repeated three times. This is because after three experiments, the statistical significance of changes could then be evaluated. Based on preliminary visual observation, a darkness interval of 90s between two pulses showed that more zebrafish larvae moved. This is the reason why the following lighting protocol consisted of 6 pulses with 90s darkness intervals.

The two PPI protocols (results shown in figures 49 and 50) one with pulses of 1s and the other with 3s pulses, show that the startle response was not inhibited. When it was observed that figure 49 showed two time points with no zebrafish movement, it was automatically assumed that PPI was exhibited in these two points only. However, when video recordings were cross referenced, the zebrafish showed startle response for both points. A mistake in the image analysis process was made and frames of prepulse were included accidentally, which is why the 2nd and 4th pulse in figure 49 show that no zebrafish moved during these two pulses. Both protocols had 50% intensity of prepulse and the delay time between prepulse and 2nd pulse was increased from 200ms up to 1s. It can be identified from the result of these two PPI protocols that zebrafish did not exhibit PPI in the developed HT system.

Another PPI protocol was applied and its results are shown in figure 51. It consisted of two startle pulses of 1s and 10s darkness interval between the two. The delay between prepulse and 2nd pulse was 200ms. The intensity of prepulse was 50% but its duration was kept as 1s, which is not the case for prepulse. Figure 51 shows that no zebrafish moved during the 2nd pulse that is provided after the prepulse. Since the prepulse was 1s, image analysis was also performed on its time frames. That is why figure 51 shows zebrafish movement during prepulse. This was an incorrect protocol; however, when cross referencing the video, it does show that on the second startle pulse, the zebrafish did not move. Since this was a faulty protocol, it cannot be concluded whether PPI was shown in zebrafish or not. For developing a better PPI protocol, the intensities of prepulses towards zebrafish should initially be tested. This way it would be easier to find out the intensity at which the zebrafish do not startle. This can be done by providing low-light baseline and prepulses slightly higher than the baseline.

Providing prepulses in darkness could exhibit startle response, which is why the baseline method should be performed along with testing prepulses in darkness. This is also one of the reasons for recording darkness intervals, because it was also decided that the lighting protocols would be applied in low light as well. When zebrafish larvae are provided with a lower light, these low light intervals between light pulses can be imaged. However, in this project videos captured at lower light levels incorrectly used PWM to attenuate light.

The significance of developing a light protocol with precision and accuracy, is that compounds that have a modulatory impact on PPI can also be tested in zebrafish models further. This would contribute to study schizophrenia-related behaviours and their underlying mechanisms (Liu et al., 2016).

In certain wells, zebrafish are moving in a circular motion, as shown and highlighted in red in figure 52. This was not captured as movement by the developed image analysis method. If for a few time frames of a single pulse the zebrafish moves from one position to another in the well, and then in the rest of the time frames the zebrafish has a circular motion, collectively this is identified as movement. However, if for the entire pulse the zebrafish moves in the circular motion, it is not identified as movement.

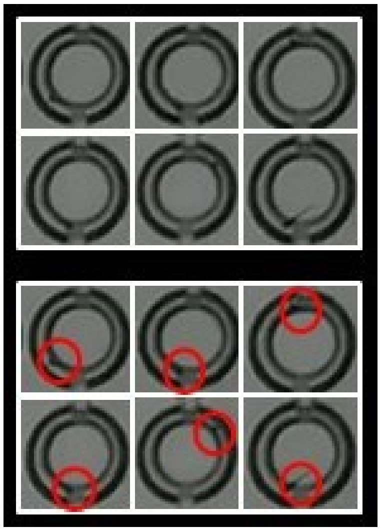

Figure 52. 6 Time Points for a single Well showing the circular movement of Zebrafish – The upper panel shows the 6 time points and the body of zebrafish is highlighted in red circles in the lower panel. This is to show that for certain wells, the image analysis method did not see this as a movement.

The system is set to detect the changes in the position of minimum size, however the situation in figure 52 can be changed with advancing optimisation of the system.

6 ACKNOWLEDGEMENTS

I would like to thank the following people for helping me throughout the thesis.

Ilkka Paatero, Ph.D.
Jesse Mattsson
Florian Freudenberg, Ph.D.
Dilhan Esen
Mikael Wasberg
Pasi Viljakainen
Jedidiah Patton
Michael Courtney, Ph.D.
Lili Li, Ph.D.
Dr. Diana Toivola
Joanna Pylvänäinen
Maritta Löytömäki
Lorenzo Li Greci
Allison McGuire

7 REFERENCES

Adams, M.M., and H. Kafaligonul, 2018. Zebrafish–a model organism for studying the neurobiological mechanisms underlying cognitive brain aging and use of potential interventions. *Front Cell Dev Biol.* 6:135.

Arduino, S.A., 2015. Arduino. *Arduino LLC.*

Bergeron, S.A., N. Carrier, G. H. Li, S. Ahn, and H.A. Burgess. 2015. Gsx1 Expression Defines Neurons Required for Prepulse Inhibition. *Mol Psychiatry*. 20:974–985.

Bhandiwad, A.A., D.G. Zeddies, D. W. Raible, E. W. Rubel, and J.A. Sisneros. 2013. Auditory sensitivity of larval zebrafish (Danio rerio) measured using a behavioural prepulse inhibition assay. *J Exp Biol*. 3504–3513.

Bhandiwad, A.A., D.W. Raible, E.W. Rubel, and J.A. Sisneros, 2018. Noise-Induced Hypersensitization of the Acoustic Startle Response in Larval Zebrafish. *JARO-J ASSOC RES OTO.* 19:741-752.

Boutros, M., F. Heigwer, and C. Laufer. 2015. Microscopy-based high-content screening. *Cell*. 163:1314-1325.

Braff, D., M. Geyer, and N. Swerdlow. 2001. Human studies of prepulse inhibition of startle: normal subjects, patient groups, and pharmacological studies *Psychopharmacology*. 234-258.

Burgess, H.A., and M. Granato. 2007. Modulation of locomotor activity in larval zebrafish during light adaptation. *J. Exp. Biol.* 210:2526-2539.

Burgess, H.A., and M. Granato. 2007. Sensorimotor gating in larval zebrafish. *J Neurosci.* 27:4984–4994.

D'Ausilio, A., 2012. Arduino: A low-cost multipurpose lab equipment. *Behav Res Methods.* 44:305-313.

Delvecchio, C., J. Tiefenbach, and H.M. Krause, 2011. The zebrafish: a powerful platform for in vivo, HTS drug discovery. *Assay and drug development technologies*. 9:354-361.

Elbaz, I., L. Yelin-Bekerman, J. Nicenboim, G. Vatine, and L. Appelbaum. 2012. Genetic ablation of hypocretin neurons alters behavioural state transitions in zebrafish. *J Neurosci.* 37:12961-72.

Howe, K., M. Clark, C. Torroja, J. Torrance, C. Berthelot, M. Muffato, and S. McLaren. 2013. The zebrafish reference genome sequence and its relationship to the human genome. *Nature*. 498.

Ioannidis, J., S. Greenland, M. Hlatky, M. Khoury, M. Macleod, D. Moher, and R. Tibshirani. 2014. Increasing value and reducing waste in research design, conduct, and analysis. *The Lancet.* 383:166-175.

Kokel, D., and R. Peterson. 2011. Using the zebrafish photomotor response for psychotropic drug screening. *Methods Cell Biol*. 517-524.

Kokel, D., J. Bryan, C. Laggner, R. White, C. Cheung, R. Mateus, and C. MacRae. 2010. Rapid behavior-based identification of neuroactive small molecules in the zebrafish. *Nat. Chem. Biol.* 6.

Legradi, J., N. El Abdellaoui, M. van Pomeren, and J. Legler. 2015. Comparability of behavioural assays using zebrafish larvae to assess neurotoxicity. *Environ Sci Pollut R.* 21:16277-16289.

Liu, H., S. Chen, K. Huang, J. Kim, H. Mo, R. Iovine, and B. Huang. 2016. A High-Content Larval Zebrafish Brain Imaging Method for Small Molecule Drug Discovery. *PloS one*.

Mahmood, F., S. Fu, J. Cooke, S.W. Wilson, J.D. Cooper, and C. Russell, 2013. A zebrafish model of CLN2 disease is deficient in tripeptidyl peptidase 1 and displays progressive neurodegeneration accompanied by a reduction in proliferation. *Brain*, 136:1488-1507.

Mathias, J. R., M. T. Saxena, and J. S. Mumm. 2012. Advances in zebrafish chemical screening technologies. *Future Med Chem.* 14:1811–1822.

Milanese, C., J. J. Sager, Q. Bai, T. C. Farrell, J. R. Cannon, T. Greenamyre, and E. A. Burton. 2012. Hypokinesia and Reduced Dopamine Levels in Zebrafish Lacking β- and γ1-Synucleins. *J Biol Chem.* 287:2971–2983.

Nolan, P.M., J. Peters, M. Strivens, D. Rogers, J. Hagan, N. Spurr, I.C. Gray, L. Vizor, D. Brooker, E. Whitehill, and R. Washbourne. 2000. A systematic, genome-wide, phenotype-driven mutagenesis programme for gene function studies in the mouse. *Nature Genet*. 25:440.

Parichy, D.M., M.R. Elizondo, M.G. Mills, T.N. Gordon, and R.E. Engeszer. 2009. Normal table of postembryonic zebrafish development: staging by externally visible anatomy of the living fish. *Dev Dyn.* 238:2975-3015.

Quevedo, K., T. Smith, B. Donzella, E. Schunk, and M. Gunnar, 2010. The startle response: Developmental effects and a paradigm for children and adults. *DEV PSYCHOBIOL.* 52:78-89.

Rihel, J., D. A. Prober, A. Arvanites, K. Lam, S. Zimmerman, S. Jang, S.J. Haggarty, D. Kokel, L.L. Rubin, R.T. Peterson, and A. F. Schier. 2010. Zebrafish Behavioral Profiling Links Drugs to Biological Targets and Rest/Wake Regulation. *Science.* 327:348–351.

Sallinen, V., V. Torkko, M. Sundvik, I. Reenilä, D. Khrustalyov, J. Kaslin, and P. Panula. 2009. MPTP, and MPP+ target specific aminergic cell populations in larval zebrafish. *J Neurochem.* 108:719-31.

Sarasamma, S., M.M. Varikkodan, S.T. Liang, Y.C. Lin, W.P. Wang, and C.D. Hsiao. 2017. Zebrafish: A Premier Vertebrate Model for Biomedical Research in Indian Scenario. *Zebrafish*. 14: 589-605

Senthilkumar, G., K. Gopalakrishnan, and V.S. Kumar, 2014. Embedded image capturing system using raspberry pi system. *IJETTCS*. 3:213-215.

Strähle, U., S. Scholz, R. Geisler, P. Greiner, H. Hollert, S. Rastegar, and T. Braunbeck. 2012. Zebrafish embryos as an alternative to animal experiments — A commentary on the definition of the onset of protected life stages in animal welfare regulations. *Reprod Toxicol.* 33:128-132.

Swerdlow, N., M. Geyer, and D. Braff. 2001. Neural circuit regulation of prepulse inhibition of startle in the rat: current knowledge and future challenges. *Psychopharmacology.* 156:194–215.

Tabor, K. M., S. A. Bergeron, E. J. Horstick, D. C. Jordan, V. Aho, T. Porkka-Heiskanen, and Burgess, H. A. 2014. Direct activation of the Mauthner cell by electric field pulses drives ultrarapid escape responses. *J Neurophysiol.* 112:834–844.

Upton, E., and G. Halfacree. 2014. Raspberry Pi user guide. *John Wiley & Sons.*

Wiley, D.S., S.E. Redfield, and L.I. Zon, 2017. Chemical screening in zebrafish for novel biological and therapeutic discovery. *Methods Cell Biol*. 138:651-679

Zhang, S., W. Wang, J. Li, K. Cheng, J. Zhou, D. Zhu, and P. Xie 2016. Behavioural characterization of CD36 knockout mice with SHIRPA primary screen. *Behav Brain Res*. 299:90-96.

Zhou, Y., R. T. Cattley, C. L. Cario, Q. Bai, and E. A. Burton. 2014. Quantification of larval zebrafish motor function in multi-well plates using open-source MATLAB® applications. *Nat Protoc.* 9:1533–1548.

8 APPENDIX

8.1 Appendix I

8.1.1 Python Code for Recording Videos through PiCam

```
from picamera import PiCamera
from time import sleep

import os
import time
import sys
import serial
import subprocess

from subprocess import call

camera = PiCamera()

camera.color_effects = (128,128)
camera.exposure_mode = 'off'

#changing shutter_speed
camera.shutter_speed = 3000

camera.iso = 100
camera.awb_mode = 'off'
camera.awb_gains = (1,8)


import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(26, GPIO.IN,pull_up_down=GPIO.PUD_DOWN)        #Official
        port 37 on PiBoard


print GPIO.input(26)

while GPIO.input(26) == 0:
        print 'Pin 26 is Low'

if GPIO.input != 0:
              camera.start_preview()
              print 'PiCam is recording'
```

```
 camera.start_recording('/home/pi/Desktop/Startle_Response.h264')
                sleep(120)       #changing the recording time based on applied
         light protocol
                camera.stop_recording()
                print 'PiCam stopped recording'
```

8.2 Appendix II

8.2.1 C Code for Lighting Protocol increasing Darkness Intervals

```
int output_pin11 = 11;
int output_pin12 = 12;
int input_pin2 = 2;

void setup() {
  // put your setup code here, to run once:
pinMode(output_pin11, OUTPUT);
pinMode(output_pin12, OUTPUT);
pinMode(input_pin2, INPUT_PULLUP);


}

void loop() {
  // put your main code here, to run repeatedly:

  while(digitalRead(input_pin2) == HIGH);

   if (digitalRead(input_pin2) == LOW){
//camera constant high
    digitalWrite(output_pin12, HIGH);

//0% 1s darkness delay
    analogWrite(output_pin11, 0);
    delay(1000);


//for loop for different pulses and delays

   int a = 10000;
   for ( a = 10000; a <= 200000; a = a + 10000 ){


     analogWrite(output_pin11, 255);
     delay(1000);
     analogWrite(output_pin11, 0);
     delay(a);
    }

    }
    else {
    digitalWrite(output_pin11, LOW);
    delay(5000);

    digitalWrite(output_pin12, LOW);

}
}
```