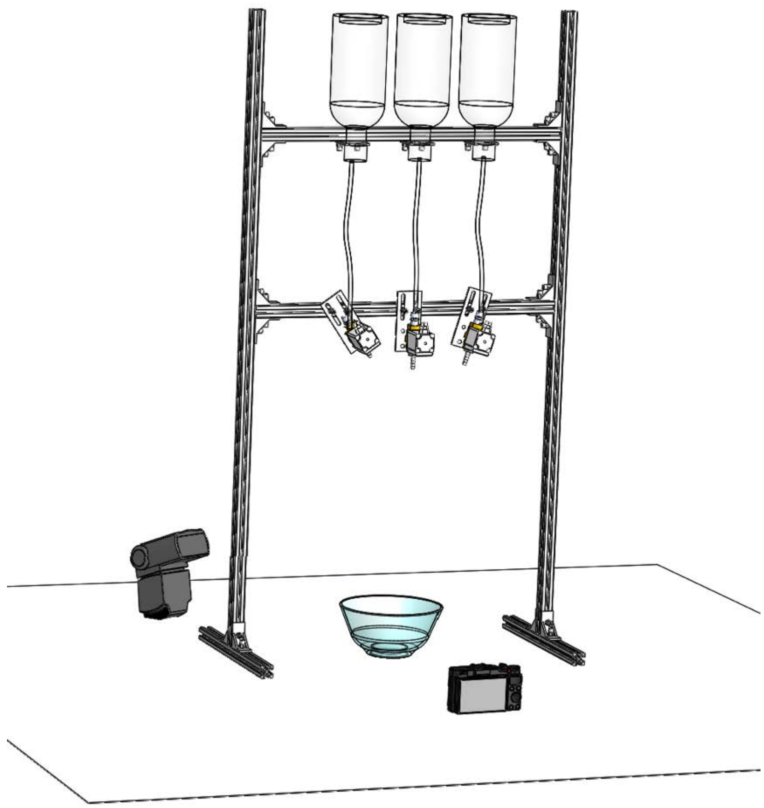
High-Speed Photography Assistant



SCATT

Zhang Enhao
Zheng Xinyi
He Kaijun
Gong Yuchen
Xu Jiahong

2016/08/08

I. Executive Summary

Nowadays, high-speed photography is popular among people and a common subject of it is to catch the motion of droplets especially when they collide. This needs high speed photography technology. However, the problem is that it is hard to take a perfect photo of droplets with common cameras. In order to solve this problem, a special assistant device is required. To finish this job, one can use high speed camera or turn to existing assistant device for help, but the high speed camera is too expensive, which usually costs several thousand dollars, and the existing assistant device has only one function and is not easy to operate. Therefore, a high-speed photography assistant is designed in order to help users shoot high-speed photographs of droplets. It possesses various outstanding features, namely affordable, easy-operating, multiple-functioned, and accurate. What's more, it suits wide range of cameras and flash lights. The bottles would provide water to the solenoid valves and the program would control the dripping time of droplets by the circuit. At the same time, the shutter and the flash light are also controlled by the program corresponded with the time of droplets' hitting the water level. With different sample codes in Arduino, different kinds of motions of the droplets can be captured, such as one droplet colliding the water level, multiple droplets colliding with each other and multiple droplets colliding the water level at the same time. Other motions can also be captured when the positions of the solenoid valves and the parameters in the program change. As a result, the whole device costs less than 40 dollars, far less than the high speed camera, and it is very easy to operate since one click can make it work. This device is especially designed for amateur photographers with DSLR camera and flash light. They can afford this device and take photos of the motions of droplets as they want with just a simple click. Different pigments can be added to the water to make the photos more beautiful.

II. Introduction

We are SCATT, coming from University of Michigan-Shanghai Jiao Tong University Joint Institute. The name of our group comes from the initials of our group members' English name. Also, these five letters represent sincere, creative, active, tenacious and trustful, which are the main characters of our team.

Our team is founded due to the course VG100. There are two projects in this semester. The first project is to construct a cart-and-bridge system. For the second project, our team decides to design a high-speed photography assistant. This final report is for project two, showing our achievements and results. In this project, we attempt to design and construct a device that can take photos of different conditions of droplets automatically.

Our team members can program with the Arduino and design the part of the mechanical structure with AutoCAD and SolidWorks. With the help from the Internet, the circuit and the structure are well designed to make the product work. To finish this project, we also gain the knowledge about the camera, the flash light and the principle of solenoid valve. In addition, some physics knowledge is required in this project which is taught in the physics class.

III. Problem

Although the high-speed photography is fascinating and amazing and shows extraordinary beauty, high-speed photograph is not easy to shoot. Since the object moves in a very fast speed, the shutter speed needs to be fast enough to avoid blurred image.

Traditionally, there are two main ways to achieve high-speed photography. Usually people use advanced photographic equipment. A high-speed camera usually has the ability to film videos at a rate in excess of 250 frames per second, while an ordinary DSLR camera can support only about 30 frames per second. However, our target audience are the amateur photographers. The problem for most amateurs is that a high-speed camera is too expensive. Take Sony PMW-EX280 as example (shown in Figure 1), it costs people 37850 CNY (5735 USD) [1], which is really a great burden to amateurs. It is unrealistic for an amateur photographer to spend much money on a high-speed camera just for taking some droplets' falling.

Figure 1: Sony PMW-EX280, a high-speed camera, sells 37850 CNY [1].

Another way is to use the assistant device. Some photographic auxiliary devices for shooting droplets have already been produced and can be bought on the internet. But still, they are not cheap. Worse still, they have very limited functions. Many fabulous photos we see on the internet are the results of two or more droplets' collision, like what is shown in Figure 2, p.5. The collisions give us more unpredictable but amazing patterns, adding the joy of high-speed photography. However, these existing auxiliary devices are only able to capture the falling with one single droplet, unless assembling some accessories otherwise. Furthermore, some photographic auxiliary devices are complex to operate. Take Time Machine as example (Figure 3, p.5), it has many buttons and varied accessories in order to achieve the high speed photography. There is no doubt that amateur photographers will be confused when they use the Time Machine for the first time.

Figure 2: The collision of two water droplets [2].

Figure 3: Time Machine [3].

Actually, there is one more way in which one can take such high-speed photos manually by thousands of attempts, but it is obviously too time-consuming and the success rate would be really low.

In summary, though high-speed cameras and existing assistant devices can be used to take high-speed photographs, the following problems still exist:

- **High expense for high-speed cameras and existing assistant devices.**
- **Single function of existing assistant devices.**
- **Complex operation of existing assistant devices.**
- **Poor precision**

IV. Needs

To solve the above problems, we are required to suit both accuracy and complexity advancement. We realize that we need the following details to make our assistant device better:

- **Affordable materials**
- **Multiple functions**
- **Simple shooting modeling**
- **High precision and success rate**

V. The Solution… High-Speed Photography Assistant

A high-speed photography assistant is designed in order to help users shoot high-speed photographs of droplets. This device can provide:

1. **Economically affordable budget of less than 40 USD.**
2. **Photo shooting of the falling of one single droplet, collisions of multiple droplets, and maximum three different colors of droplets; possibilities of making GIF that records the whole process of droplets' motion.**
3. **An easy-operating system that can change the size of the droplets and the time interval between two consecutive droplets according to user's needs.**
4. **Accurate control of the solenoid valves, along with triggering the camera and flash light, with precision 1 millisecond and success rate about 96%.**

The main idea of our project is to use the Arduino board to precisely control the time to drip the droplets, the time to press the shutter, and the time to trigger the flash light. The control system will first let one single droplet drops from the solenoid valve. The height of the solenoid valve from the table is fixed, and the time delay of the control system is also fixed. Therefore, the total time needed for the whole process can be determined after some experiments. So when droplets are about to move to the desired position, the Arduino board will send signals to both the camera and the flash light, and pictures will be shot. The high-speed photography assistant consists of two parts: the mechanical structure and the Arduino control system.

A. The Mechanical Structure

The components of the mechanical structure are three solenoid valves, PVC hoses, some plastic nipples, three bottles, a glass bowl, and the Aluminum profile (Figure 4, p.7). The Aluminum profile is used to fix the bottles and the solenoid valves. The solenoid valve has an inner diameter 12 mm, the smallest diameter we can find, to guarantee the water can trickle from it drop by drop. Both ends of the solenoid valve are connected with a plastic nipple. Each plastic nipple we use has two different ends: one has an outer diameter 12 mm, the other has an outer diameter only 6 mm. A long PVC hose is used to connect the valve and the bottle, with an inner diameter also 6 mm. The bottle is hung on the top of the frame, and it is open to the atmosphere to allow water successfully drops down. The glass bowl is placed on the table, right underneath the solenoid valve, to collect the droplets. The bowl also contains some liquids.

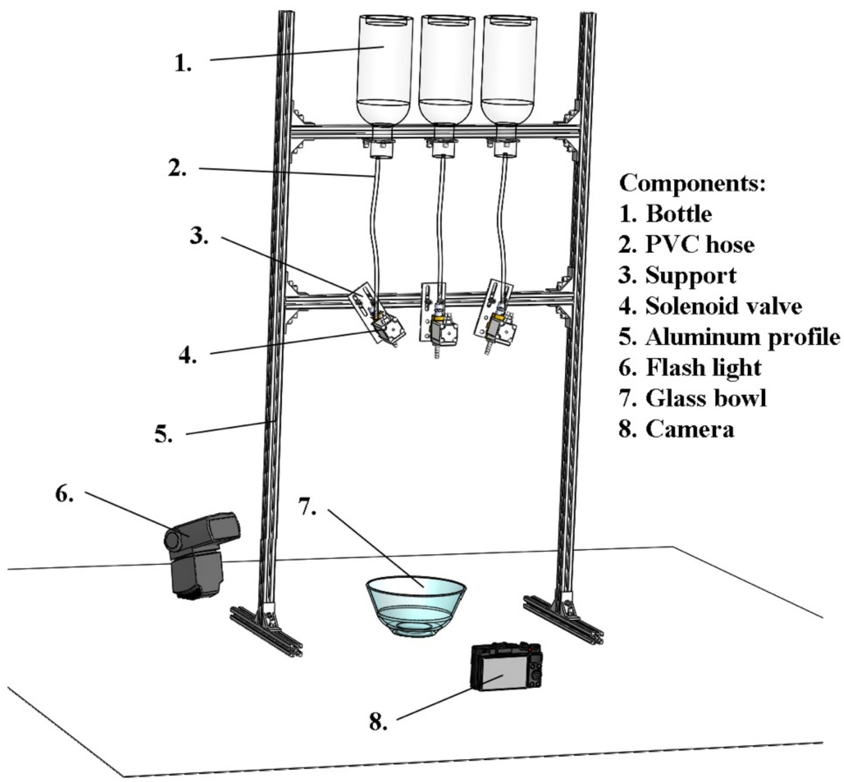


Figure 4: Design diagram of the mechanical structure, with components of the device clearly labeled on the right top.

B. The Arduino Control System

The Arduino control system contains an Arduino UNO board, two L298N motor driving boards, two batteries, some converters and some Dupont lines. The circuit diagram is shown below in Figure 5, p.8. Among them, the Arduino UNO board is the core—it controls the solenoid valves, the shutter and the flash light. The operating voltage of the solenoid valves used in this project is about 12 V, but the Arduino can only supply a maximum 5 V voltage. Therefore, we decide to use the motor driving board in between the Arduino and the solenoid valve. Each motor driving board will be powered by a 12 V Li-Po battery. When we want to drip a droplet, the Arduino will send signals to the motor driving board so that it will add HIGH voltage to one end of the solenoid valve and LOW voltage to the other end. The valve will open and liquids can flow through it. By controlling the duration of opening the valve, the size of the droplet can be controlled according to our needs. The principles of controlling the shutter and the flash light are the same. They will first connect with some converters to transform the original interface into two wires. Each wire will connect to the Arduino. When the shutter or the flash light needs to be triggered, the Arduino will add both LOW voltage to the two wires.

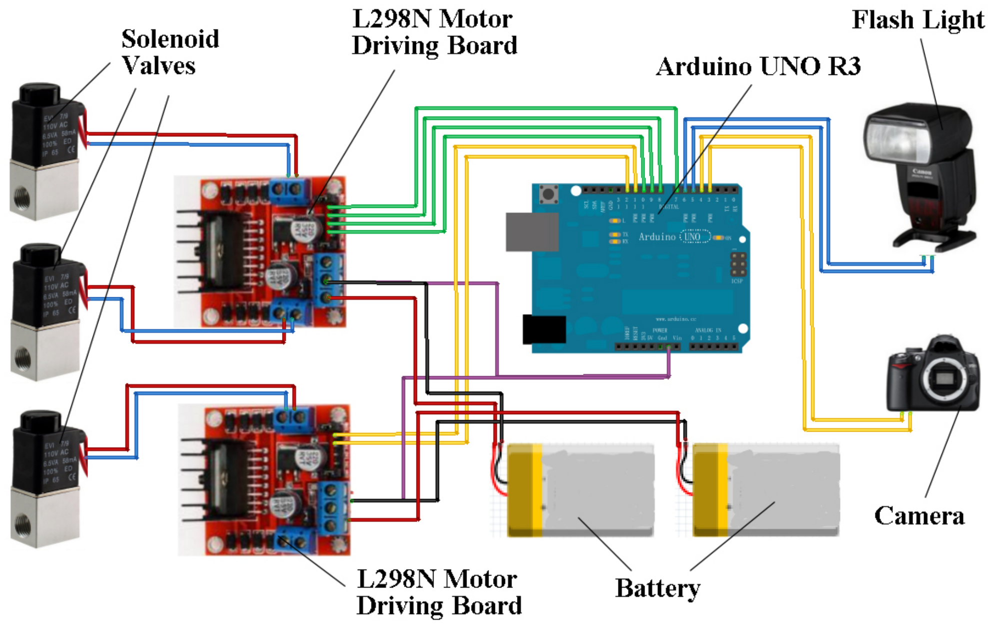


Figure 5: Circuit diagram of the Arduino control system (made with Fritzing.org).

Here are some functional components used in our project:

1. **Solenoid Valve** (Figure 6)

It is used to drip the droplets. In usual time it is closed and fluids are not able to flow through it. When the coil is energized, the valve will be open, and if the energized time is extremely short, droplets will be dripped from it.

Figure 6: Solenoid valve.

2. **Arduino UNO R3 Board** (Figure 7, p.9)

It is the core of the control system. It processes the data of time to trigger the solenoid valves, the camera, and the flash light. Also, it sends signals to other components to guarantee the system functions.

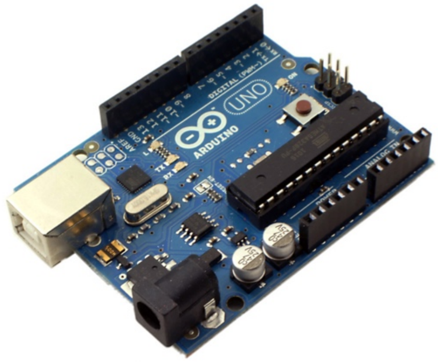


Figure 7: Arduino UNO R3 board.

3. Motor Driving Board (Figure 8)
It is used to connect the Arduino board and the solenoid valves. Since the rated voltage of solenoid valves is 12 V while the Arduino board can only supply a maximum 5 V voltage, they cannot be connected directly. Motor driving board is then applied, acting more like a switch.

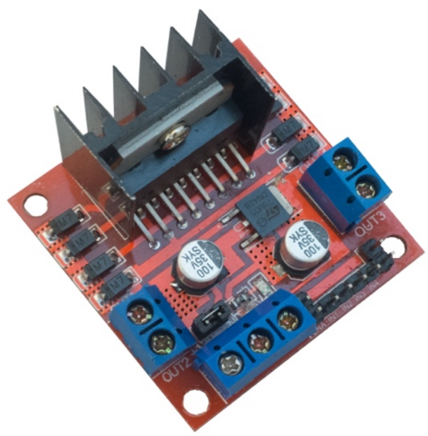


Figure 8: Motor driving board.

4. Aluminum Profile (Figure 9)
It works as a frame to fix other objects, such as solenoid valves and bottles.

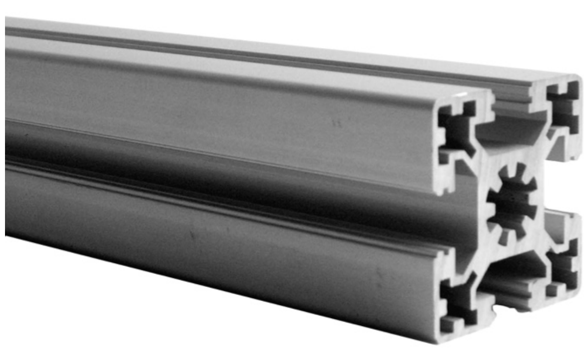


Figure 9: Aluminum profile [4].

5. Support for the Bottle (Figure 10)
It is fixed on the Aluminum profile and the bottles can be stuck on it, so that the bottles will not move or fall off.

Figure 10: Support for the bottle.

6. Support for the Solenoid Valve (Figure 11)
It is fixed on the Aluminum profile. The solenoid valve is fixed on it with screws. There are two parallel slits on the support, making it possible to adjust the height and inclination of the solenoid valve.

Figure 11: Support for the solenoid valve.

VI. Objectives

- **Objective 1: Analyze the principle of high-speed photography.**

Before constructing our project, we need to first figure out the working principle of high-speed photography, as well as gaining some basic knowledge of the camera and flash light.

- **Objective 2: Design and make a mechanical structure.**

The mechanical structure should satisfy the needs that multiple solenoid valves and bottles can be steadily fixed on it. What's more, the mechanical structure should reserve the height for the falling of droplets.

- **Objective 3: Control the solenoid valves, the shutter and the flash lights.**

Since the high-speed photography assistant should be very accurate and easy-operating, we need to control all the components by Arduino Board to save labor and improve the success rate.

- **Objective 4: Conduct delay test for the solenoid valves, the shutter and the flash light.**

Since the target precision of the device is 1 millisecond, and there is objective equipment delay for each component, we need to make experiments to test the delay so that the device can work precisely.

- **Objective 5: Program the solenoid valves, the shutter and the flash light. Provide several sample codes for the user.**

To finally make the device work, we need to program all the components and upload the codes to the Arduino board. To make the device user-friendly, we need to provide sample codes to users for fundamental use.

- **Objective 6: Check validation of our device.**

Test our device by taking series of photos using different sample codes. Also, check the success rate and precision of our device. Make adjustments if necessary.

VII. Tasks

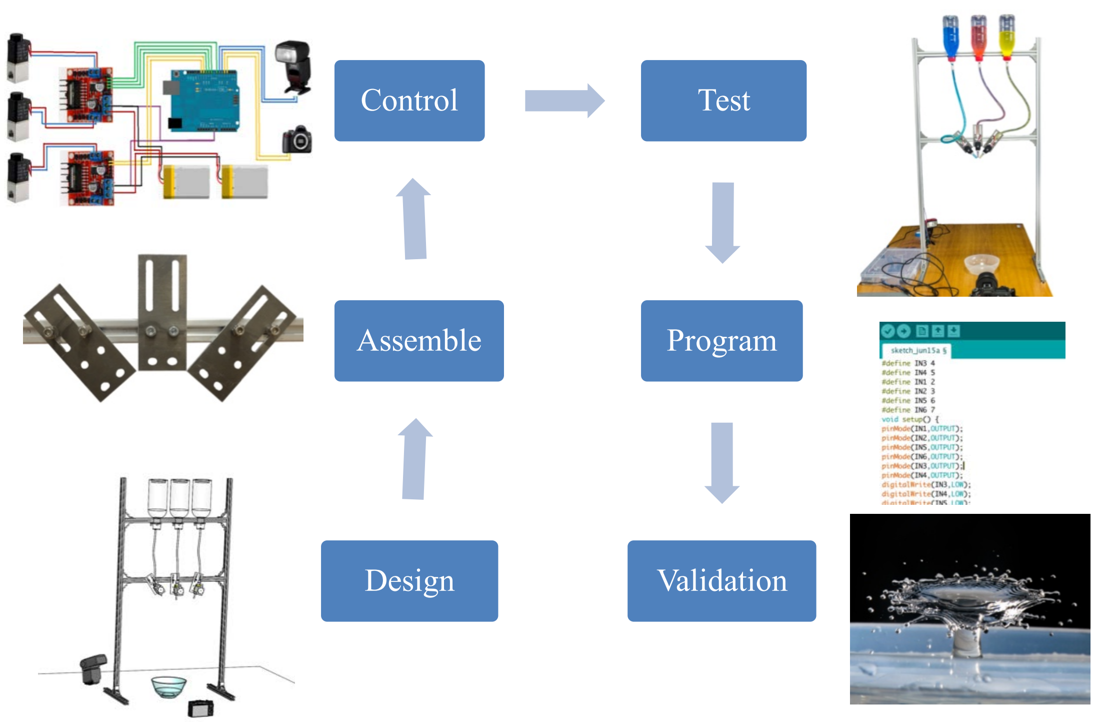


Figure 12: Task flow diagram of the high-speed photography assistant project.

Task 1: Design an assistant device for high-speed photography.

The first step we do is to find a component that can control the dripping of one single droplet. The first solution we come up with is to use a tap and some infrared sensors. The tap will drip the droplet and then infrared sensors will be used to locate the exact position of the moving droplet. We give up this plan later, though, for the reason it does not possess multiple functions in shooting pictures. Instead, we choose to use solenoid valves. When energized, the valve will be open and fluid will be able to flow through it. It means that it is possible to control the system using some circuit. Then, we try to figure out the working principle of both the camera and the flash light. We find that though they are specialized equipment, their working principle is simple that they can be triggered simply by short circuit. As a result, the camera and the flash light can also be controlled by the Arduino, which means that a preliminary design of our device is formed (as shown in Figure 4, p.7).

Afterward, we start thinking about how to fix the bottles and the solenoid valves. We decide to use the Aluminum profile as the frame to fix these objects. In addition, we design the supports for the bottles and the solenoid valves to fix them.

Task 2: Assemble the mechanical structure.

According to the CAD diagrams attached in Appendix A, we assemble the mechanical structure with the Aluminum profile. To connect the components of Aluminum profile, we use connecting pieces and screws. The details are shown in Figure 13.

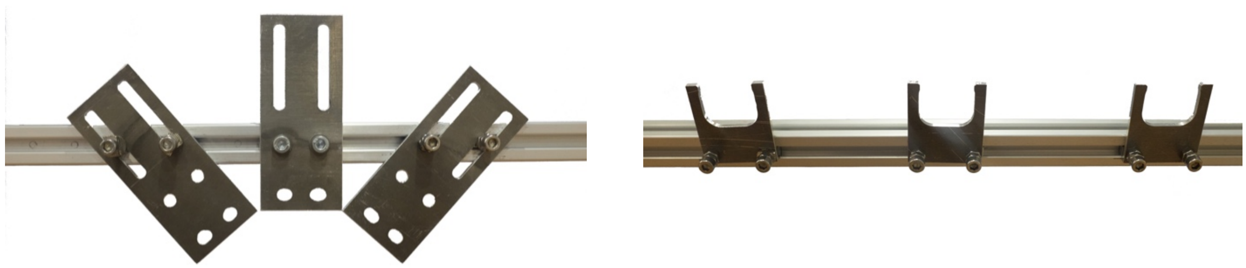


Figure 13: Assembling details of the mechanical structure.

Task 3: Control the solenoid valves, the shutter and the flash light.

After the assembling process, we start to accomplish the control of all the components, so that our device can really function.

3.1 Control the solenoid valves

To contain and supply water continuously, we are to use appropriate water bottles. After our experiments, we determine to use Nongfu Spring bottles which have suitable size that can be embedded into the clamp and have fair appearance. Then we connect the PVC hose with the plastic nipple and the bottle. Use thread seal tape to prevent water leak. In order to keep the pressure in the bottle as atmospheric pressure, we use needle to make a small hole on the bottom of the bottle.

Then connect the solenoid valves, the motor driving board and the Arduino board with Dupont lines according to Figure 5, p.8. The assembling details of solenoid valves are shown in Figure 14 and 15.

Figure 14: The overall view of the mechanical structure.

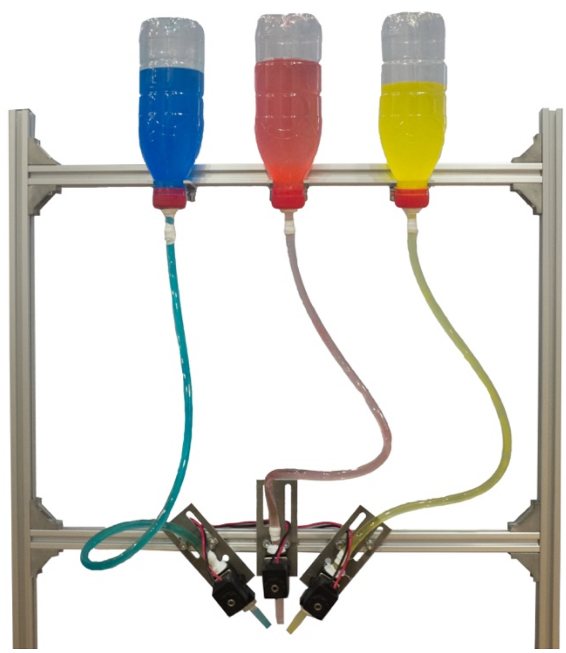


Figure 15: Assembling the bottles and solenoid valves onto the frame.

After assembling the solenoid valves, the motor driving board and the water bottles, we program the solenoid valves and test whether they could work. The testing code for solenoid valves is in Appendix B.1.

3.2 Control the shutter

Firstly, we connect the shutter converter to the camera. Then we cut off the wire of the converter and strip about 2 cm of the wire. There are three thin copper wires inside the wire of the converter. Then we short-circuit the live wire and the neutral wire to check whether the shutter works. Finally, we connect the live wire and the neutral wire with Dupont lines according to the circuit diagram shown in Figure 5, p.8 and the tables attached in Appendix C.

After assembling the shutter, we program the shutter and test whether it works under different shutter time. The testing code for shutter is in Appendix B.2.

3.3 Control the flash light

Firstly, we connect the flash light converter to the flash light. Then we cut off the wire of the converter and strip about 2 cm of the wire. There are three thin copper wires inside the wire of the converter. Then we short-circuit the live wire and the neutral wire to check whether the flash light works. Finally, we connect the live wire and the neutral wire with Dupont lines according to the circuit diagram shown in Figure 5, p.8 and the tables attached in Appendix C.

After assembling the flash light, we program the flash light and test whether it works under different flash intensities. The testing code for flash light is in Appendix B.3.

Task 4: Test the whole device.

We first test the time delay of the shutter and the flash light when triggered. Program a loop in Arduino. Each time delay 1 millisecond with fixed aperture and shutter time (For our device, the aperture is set to F22, while the shutter time is 1/4 second). The test code is in Appendix B.4. After the device running through the code, check which picture is in best light condition. The delay time for the best picture is the delay of the shutter and the flash light that will be used. In other programs, use this delay time as the time interval between the shutter and the flash light. For our device, the delay is 235 milliseconds.

After that, we test the time delay of the solenoid valves. Program a loop in Arduino. Each time delay 1 millisecond with fixed size of droplets. The test code is in Appendix B.5. While the camera takes pictures, observe the pictures. Hence, the exact time when the droplet collides with the water surface for the first time can be obtained. Also, the exact time when the droplet bounces off to the highest position can be obtained. Using these data in other programs, we can add diversity to the pictures.

Task 5: Program the solenoid valves, the shutter and the flash light for different patterns of pictures.

After some experiments, we record some sets of data that could be used under different conditions and provide the following four sample codes.

- Single droplet collides with the water surface and bounces off to the highest position. The sample picture is Figure 16. The code is in Appendix B.6.

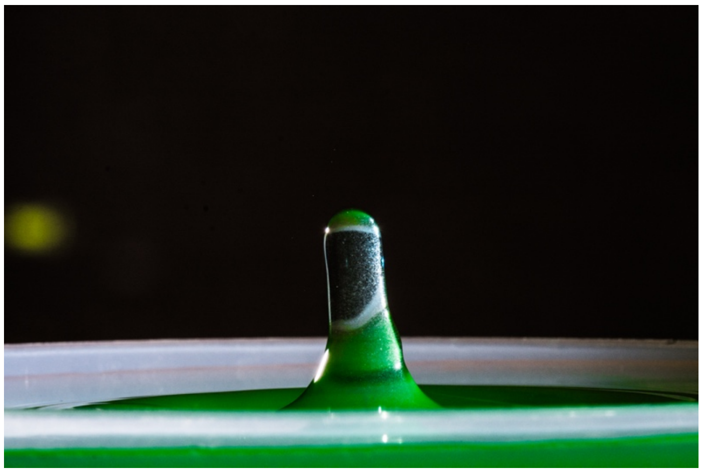


Figure 16: Image of single droplet bouncing off to the highest position.

- Single solenoid valve successively drips two droplets. The first droplet bounces off and collides with the second droplet. The sample picture is Figure 17. The code is in Appendix B.7.

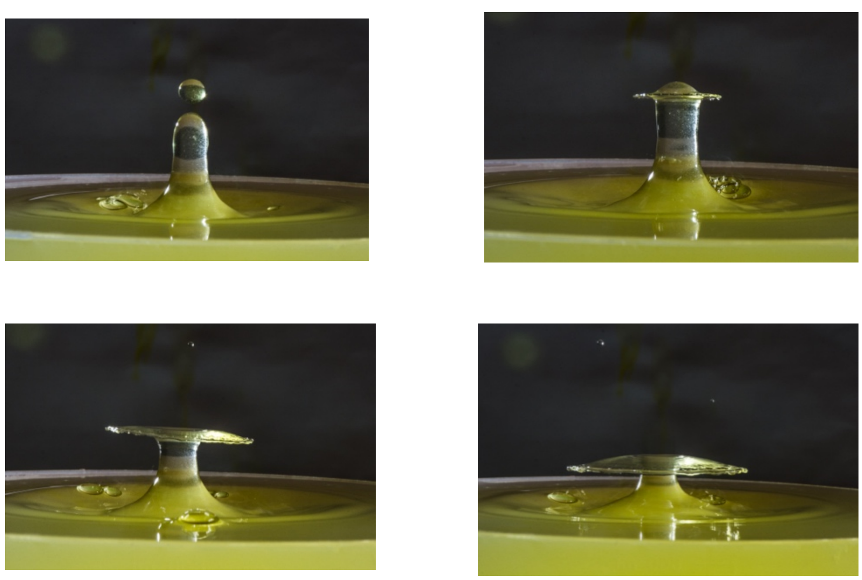


Figure 17: Images of the collision of two droplets.

- Two solenoid valves separately drip one droplet. The two droplets collide in the air. The sample picture is Figure 18. The sample code is in Appendix B.8.

Figure 18: Images of the mixture of two droplets with different colors.

- Three solenoid valves separately drip one droplet. The solenoid valve in the middle drips in advance. And the middle droplet bounces off and collides with the other two droplets. The sample picture is Figure 19. The sample code is in Appendix B.9.

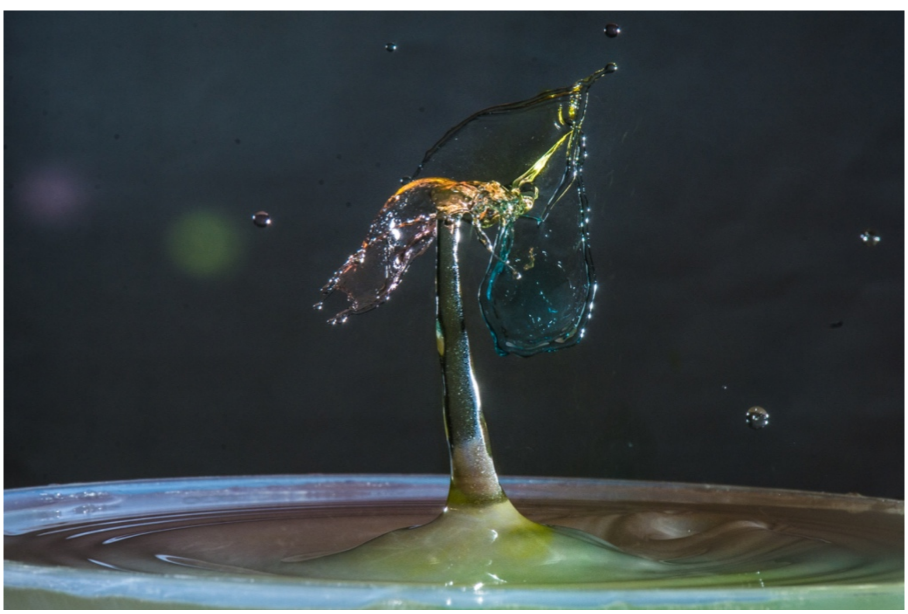


Figure 19: Images of the collision of three droplets with different colors.

Task 6: Validation

The final version of our prototype is shown in Figure 20.

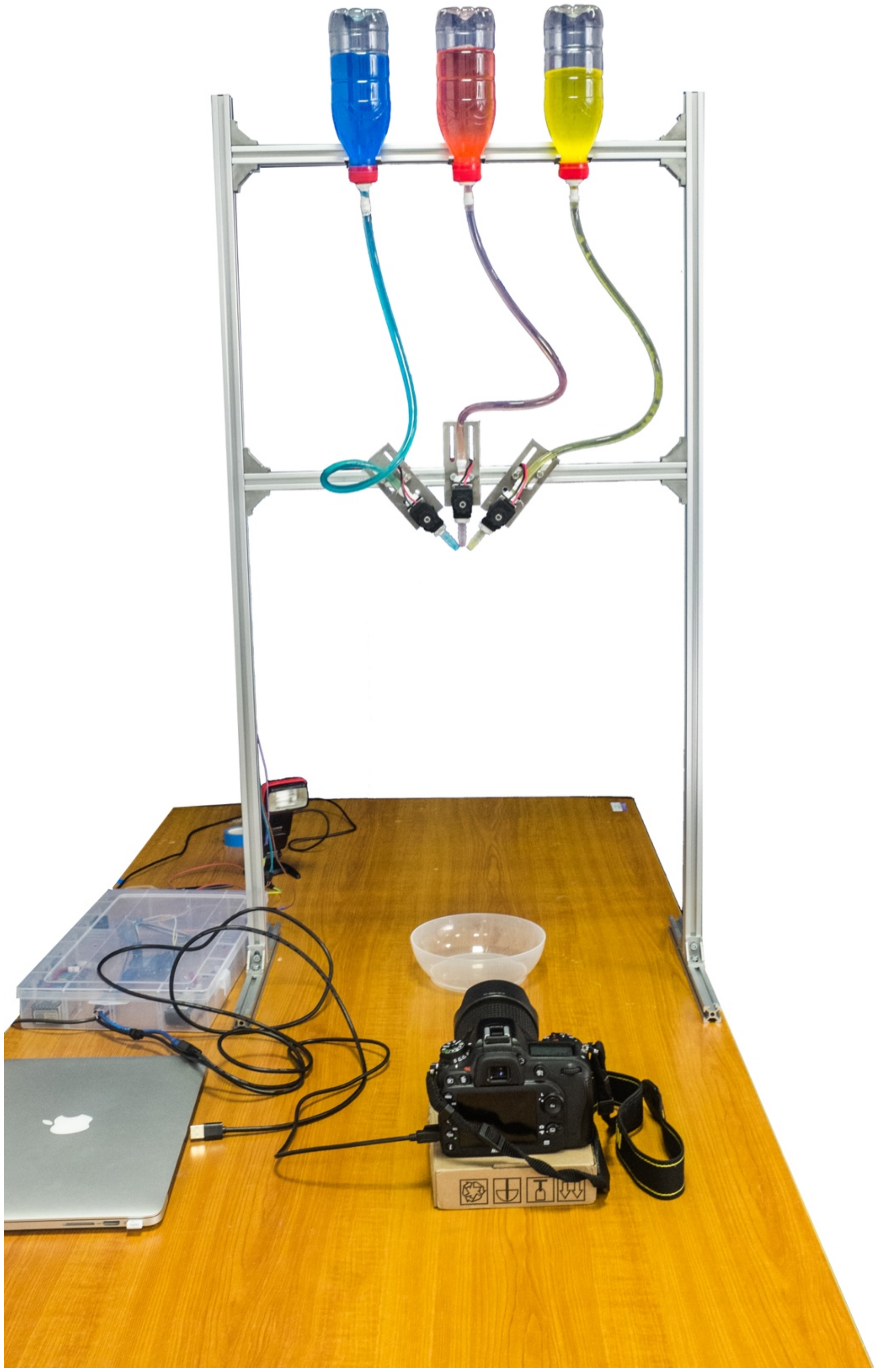


Figure 20: Prototype for the high-speed photography assistant project.

Having completed the above tasks, we do some experiments to evaluate the validation of our device. We first use the sample codes mentioned in Task 5 to shoot pictures, and Figure 16-19 prove our device functions well.

Also, we use the device to shoot pictures consecutively, in order to test the precision and success rate of our device. Figure 21 shows forty-eight consecutive pictures taken with our high-speed photography assistant. The time difference between the adjacent two photos is only 1 millisecond. Among these photos, only two of them fail to capture the image. The result shows that the success rate of our project is about 96% with precision 1 millisecond.

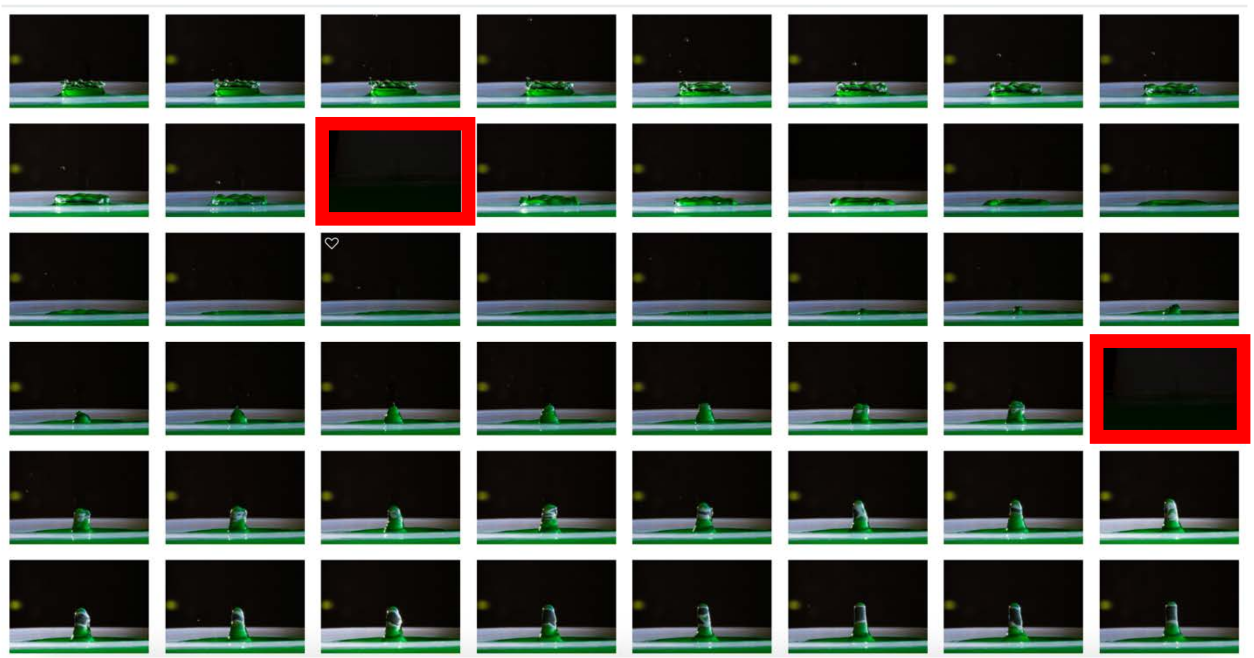


Figure 21: Forty-eight photos taken with the high-speed photography assistant consecutively. Only two of them fail to capture the image, which are highlighted with red rectangles.

High-speed photography is fascinating also because it is unpredictable and people can get creative to produce their own unique masterpieces. Therefore, we just change the time parameters provided in the Arduino code to see whether special photos can be taken. The results are attached in Appendix D, which verify the multiple-functioned feature of our device.

VIII. Schedule

The schedule for the high-speed photography assistant project is shown in Figure 22.

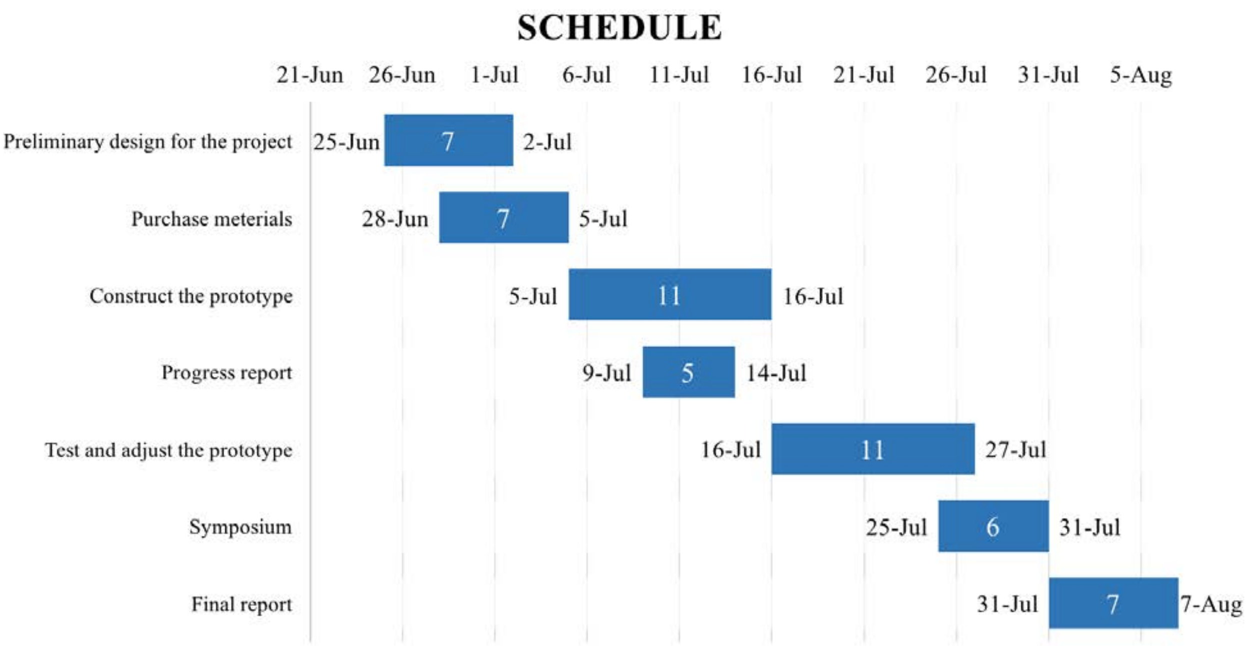


Figure 22: The Gantt chart showing the schedule of the high-speed photography project.

IX. Budget

The budget of the high-speed photography assistant project is shown in Table 1. All components used for this project are common materials that are available on Taobao. Since the project is only an auxiliary device, the camera and the flash light are not included in the budget. The links of the materials are attached in Appendix E.

Table 1: The budget of the high-speed photography assistant project.

Materials	Price (USD)	Quantities	Total Cost	Description	Manufacturer
Arduino UNO R3	4	1	4	Processor	Microduino
Solenoid valve	1.5	3	4.5	Controlling the droplets	civil PNEUMATIC
Motor driving board	1.2	2	2.4	Controlling the solenoid valve	TELESKY
LiPo battery/11.1V/25c	4.8	2	9.6	Providing power	Build-Power
Aluminum profile	11	1	11	Constructing the frame	Bernabeu
PVC hose	0.81	1	0.81	Transport water	AQUA TECH
Aluminum support	1.5	6	9	Fixing bottles and solenoid valves	Oudifu
Total Cost: 39.04(USD)					

X. Key Personnel

To maximize our working efficiency, we assign complex tasks to different team members according to everyone's strengths.

Zhang Enhao is our team leader as well as technical communicator. Most of the reports are written by him. Gong Yuchen is our programming designer who mainly focuses on writing the Arduino program. Zheng Xinyi is the material purchaser. She selects all materials we need on the internet, and also takes charge of bookkeeping. Xu Jiahong is our component deployer, focusing on system assembling and making each part stable. He Kaijun is our prototype designer who designs the system and the structure. More intuitive illustration is shown in Figure 23.

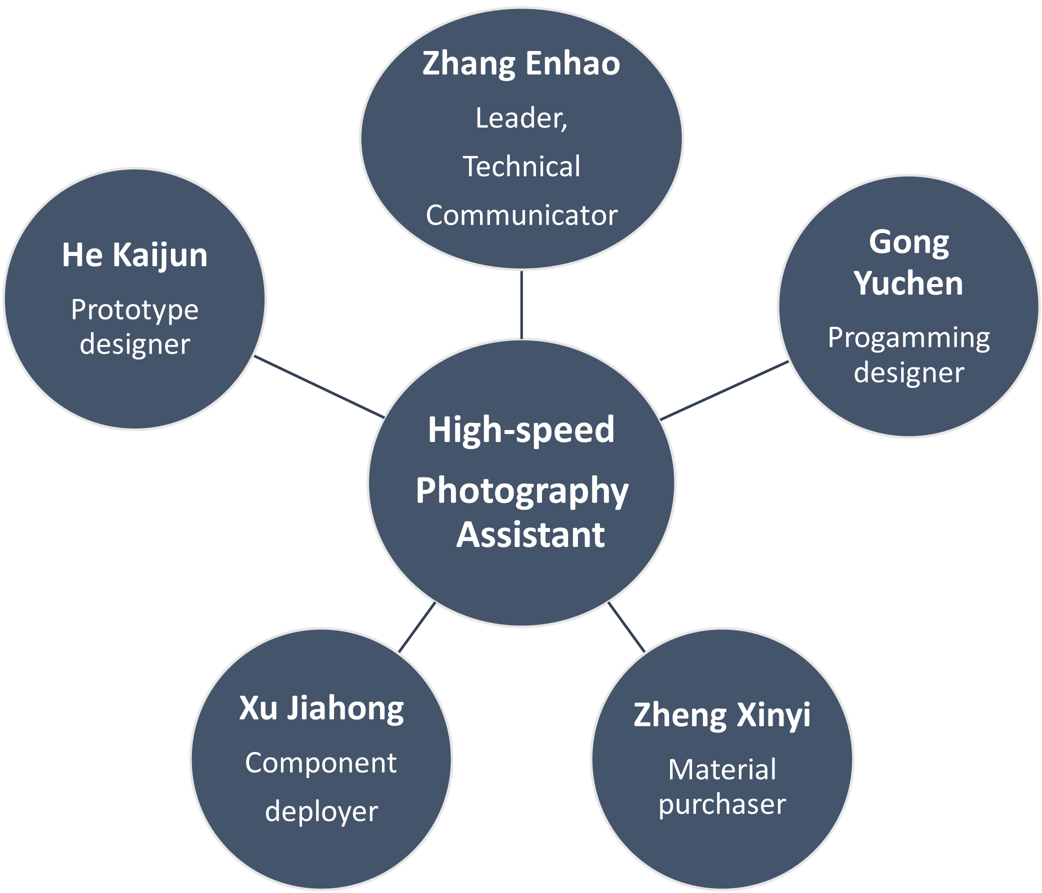


Figure 23: Key personnel of our project.

XI. References

[1] Sony, 2016, “PMW-EX280,” Sony, accessed at http://pro.sony.com.cn/pro/product/broadcast -products-camcorders-xdcam/pmw-ex280/overview/ on August 5, 2016.

[2] Petre, J., 2011, “A Little Drop of Magic: One Woman Turns Drops of Water into Mushrooms, Aliens - and Even Spider-Man,” Daily Mail Online, accessed at http://www.dailymail.co.uk/ news/article-2072572/A-little-drop-magic-One-woman-turns-dropswater- mushrooms-aliens- -Spider-Man.html on July 12, 2016.

[3] Mumford, B., “The Mumford Time Machine a Programmable Controller/ Intervalometer for Digital Cameras and Electronic Flashes,” Mumford Micro Systems, http://www.bmumford. com/ photo/camctlr.html on August 5, 2016.

[4] Taobao, accessed at https://item.taobao.com/item.htm?spm=a1z10.1-c.w5003-14127564414. 4.jKD6E1&id=38413772034&scene=taobao_shop on August 5, 2016.

XII. Appendix A

The CAD diagrams of the high-speed photography assistant project are shown in Figure 24 and 25.

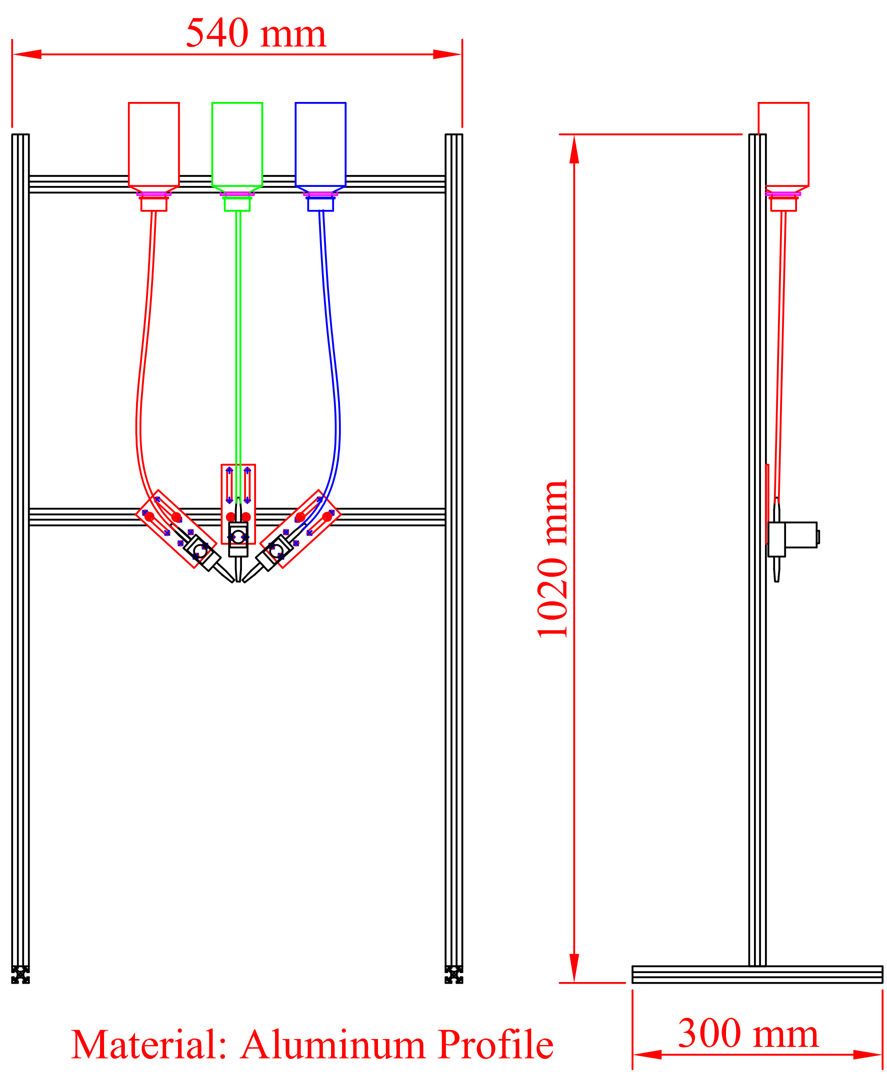


Figure 24: The Front view and left view of the CAD diagram for the frame.

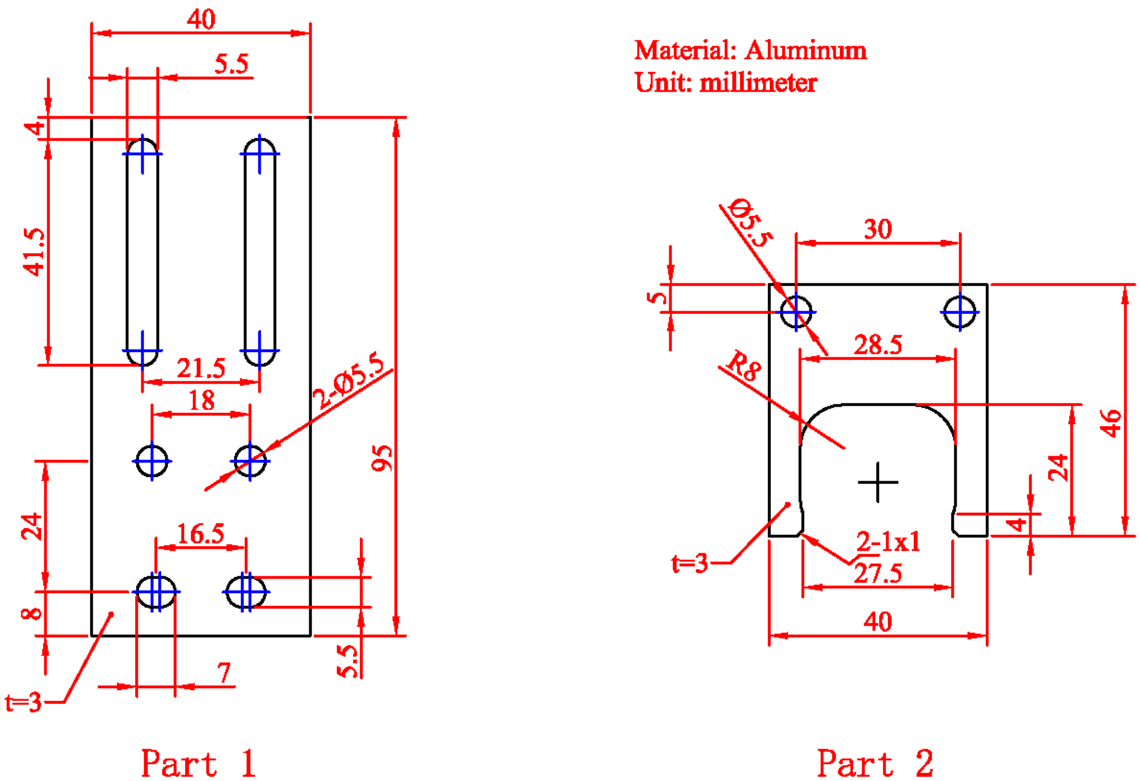


Figure 25: CAD diagrams for the supports of the solenoid valve (Part 1) and the bottle (Part 2).

XIII. Appendix B

The followings are the Arduino code:

1. Control the Solenoid Valves

```
#define IN3 4
#define IN4 5
#define IN1 2
#define IN2 3
#define IN5 6
#define IN6 7
void setup() {
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
pinMode(IN5,OUTPUT);
pinMode(IN6,OUTPUT);
pinMode(IN3,OUTPUT);
pinMode(IN4,OUTPUT);
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);
digitalWrite(IN5,LOW);
digitalWrite(IN6,LOW);
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
delay(5000);
}
void loop() {
digitalWrite(IN5,HIGH);
digitalWrite(IN4,HIGH);
digitalWrite(IN2,HIGH);
delay(20);
digitalWrite(IN5,LOW);
digitalWrite(IN4,LOW);
digitalWrite(IN2,LOW);
}
```

2. Control the Shutter

```
void setup() {
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
```

```
delay(5000);
}
void loop() {
digitalWrite(10,LOW);
delay(20);
digitalWrite(10,HIGH);
}
```

3. Control the Flash Light

```
void setup() {
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
digitalWrite(8,HIGH);
digitalWrite(8,LOW);
delay(5000);
}
void loop() {
digitalWrite(8,LOW);
delay(20);
digitalWrite(8,HIGH);
}
```

4. Delay Test for the Shutter and the Flash Light

```
int i=0;
void setup() {
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
}
void loop() {
delay(5000);
digitalWrite(8,LOW);
delay(i);
digitalWrite(10,LOW);
delay(20);
digitalWrite(10,HIGH);
digitalWrite(8,HIGH);
i=i+1
}
```

5. Delay Test for the Solenoid Valves

```
int i=0;
#define IN1 2
#define IN2 3
void setup() {
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
}
void loop() {
delay(5000);
digitalWrite(IN2,HIGH);
delay(20);
digitalWrite(IN2,LOW);
delay(i)
digitalWrite(8,LOW);
delay(235);
digitalWrite(10,LOW);
delay(20);
digitalWrite(10,HIGH);
digitalWrite(8,HIGH);
i=i+1
}
```

6. Sample One

```
#define IN1 2
#define IN2 3
void setup() {
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
digitalWrite(8,HIGH);
```

```
digitalWrite(9,LOW);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
delay(5000);
}
void loop() {
digitalWrite(IN2,HIGH);
delay(20);
digitalWrite(IN2,LOW);
delay(150);
digitalWrite(8,LOW);
delay(235);
digitalWrite(10,LOW);
delay(20);
digitalWrite(10,HIGH);
digitalWrite(8,HIGH);
}
```

7. Sample Two

```
#define IN3 4
#define IN4 5
#define IN1 2
#define IN2 3
int i=40;
void setup() {
 pinMode(IN3,OUTPUT);
 pinMode(IN4,OUTPUT);
 pinMode(10,OUTPUT);
 pinMode(11,OUTPUT);
 pinMode(8,OUTPUT);
 pinMode(9,OUTPUT);
 pinMode(IN1,OUTPUT);
 pinMode(IN2,OUTPUT);
 digitalWrite(10,HIGH);
 digitalWrite(11,LOW);
 digitalWrite(8,HIGH);
 digitalWrite(9,LOW);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,LOW);
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,LOW);
 delay(5000);
 }
```

```
void loop() {
  digitalWrite(IN4,HIGH);
  digitalWrite(IN2,HIGH);
  delay(40);
  digitalWrite(IN4,LOW);
  digitalWrite(IN2,LOW);
  delay(i);
  digitalWrite(8,LOW);
  delay(235);
  digitalWrite(10,LOW);
  delay(20);
  digitalWrite(10,HIGH);
  digitalWrite(8,HIGH);
  i=i+1;
}
```

8. Sample Three

```
#define IN3 4
#define IN4 5
#define IN1 2
#define IN2 3
void setup() {
pinMode(IN3,OUTPUT);
pinMode(IN4,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
delay(5000);
}
void loop() {
digitalWrite(IN4,HIGH);
digitalWrite(IN2,HIGH);
delay(40);
```

```
digitalWrite(IN4,LOW);
digitalWrite(IN2,LOW);
delay(70);
digitalWrite(8,LOW);
delay(235);
digitalWrite(10,LOW);
delay(20);
digitalWrite(10,HIGH);
digitalWrite(8,HIGH);
}
```

9. Sample Four

```
#define IN3 4
#define IN4 5
#define IN1 2
#define IN2 3
#define IN5 6
#define IN6 7
void setup() {
pinMode(IN3,OUTPUT);
pinMode(IN4,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(IN1,OUTPUT);
pinMode(IN2,OUTPUT);
pinMode(IN5,OUTPUT);
pinMode(IN6,OUTPUT);
digitalWrite(10,HIGH);
digitalWrite(11,LOW);
digitalWrite(8,HIGH);
digitalWrite(9,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);
digitalWrite(IN5,LOW);
digitalWrite(IN6,LOW);
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
delay(5000);
}
void loop() {
digitalWrite(IN5,HIGH);
delay(20);
digitalWrite(IN5,LOW);
```

```
delay(60);
digitalWrite(IN4,HIGH);
digitalWrite(IN2,HIGH);
delay(40);
digitalWrite(IN4,LOW);
digitalWrite(IN2,LOW);
delay(50);
digitalWrite(8,LOW);
delay(235);
digitalWrite(10,LOW);
delay(20);
digitalWrite(10,HIGH);
digitalWrite(8,HIGH);
}
```

XIV. Appendix C

Table 2-5 show the connecting of the control system.

Table 2: Connections between the Arduino board and the motor driving board 1.

Motor driving board 1	Arduino board
IN1	PIN 2
IN2	PIN 3
IN3	PIN 4
IN4	PIN 5
GND	GND

Table 3: Connections between the Arduino board and the motor driving board 2.

Motor driving board 2	Arduino board
IN1	PIN 7
IN2	PIN 6
GND	GND

Table 4: Connections between the Arduino board and the camera.

Camera	Arduino board
VCC	PIN 8
GND	PIN 9

Table 5: Connections between the Arduino board and the flash light.

Flash light	Arduino board
VCC	PIN 10
GND	PIN 11

XV. Appendix D

More photographs taken with the high-speed photography assistant (Figure 26-37).

Figure 26: Picture one.

Figure 27: Picture two.

Figure 28: Picture three.

Figure 29: Picture four.

Figure 30: Picture five.

Figure 31: Picture six.


Figure 32: Picture seven.

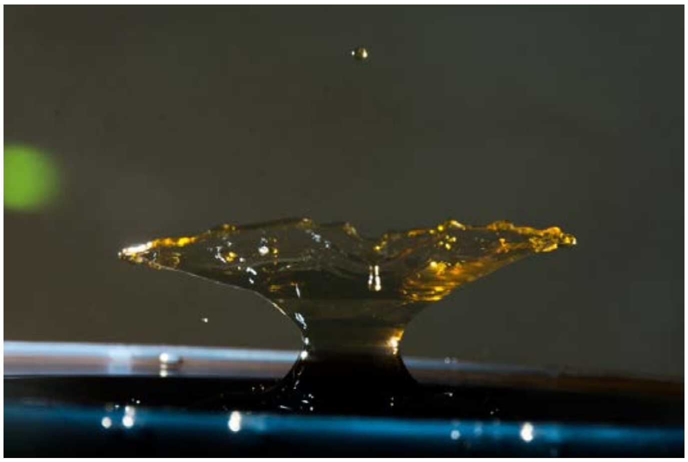


Figure 33: Picture eight.


Figure 34: Picture nine.


Figure 35: Picture ten.


Figure 36: Picture eleven.

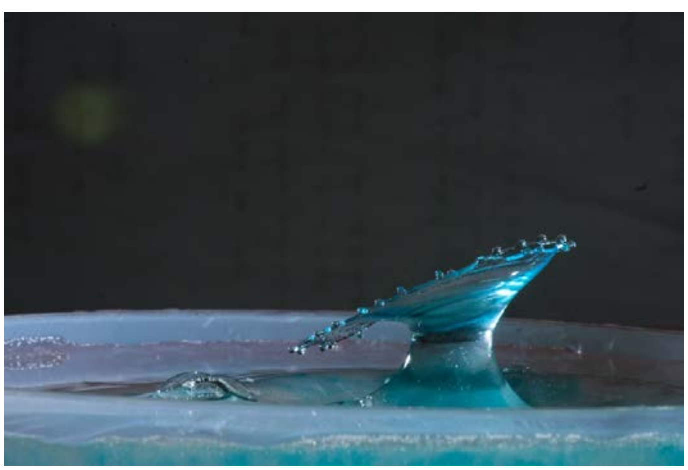


Figure 37: Picture twelve.

XVI. Appendix E

The URL links of the key components purchased on Taobao are listed in Table 6.

Table 6: Key components of the project and the corresponding URL links.

Materials	Hyperlink
Arduino UNO R3	https://item.taobao.com/item.htm?spm=a230r.1.14.20.LmXNyK&id=40666747274&ns=1&abbucket=9#detail
Solenoid valve	https://item.taobao.com/item.htm?spm=a230r.1.14.20.zviSMV&id=45271613910&ns=1&abbucket=9#detail
Motor driving board	https://detail.tmall.com/item.htm?spm=a230r.1.14.6.XqSpKV&id=41248562401&cm_id=140105335569ed55e27b&abbucket=9
LiPo battery/11.1V/25c	https://item.taobao.com/item.htm?spm=a230r.1.14.22.UVtUSb&id=42164560730&ns=1&abbucket=9#detail
Aluminum profile	https://item.taobao.com/item.htm?spm=a1z10.1-c.w5003-14127564414.4.jKD6E1&id=38413772034&scene=taobao_shop
PVC hose	https://item.taobao.com/item.htm?spm=a230r.1.14.187.N4xEhG&id=25416904276&ns=1&abbucket=9#detail
Aluminum support	https://detail.tmall.com/item.htm?spm=a220m.1000858.1000725.1.ZcLoSZ&id=43973179466&areaId=310100&cat_id=2&rn=5013263e10e6b05a79e262fb775073dc&user_id=2422492010&is_b=1