G02_HW06

Group 02
HW 06
2019/10/22

ID	Name	Your works	Times you spend	Self score	TA
108202529	葉揚昀	CCD & CMOS CMOS record pictures	10hr	8	
108202009	田家瑋	Pulse-width modulation	3hr	6	
108202016	張家菖	ISO, shutter, and what is numerical aperture.	7hr	6.8	

CCD (Charge-couple Device)

1. Ingredients:

   The CCD's main material is silicon crystal semiconductor.

2. Structure: (show in fg.1)

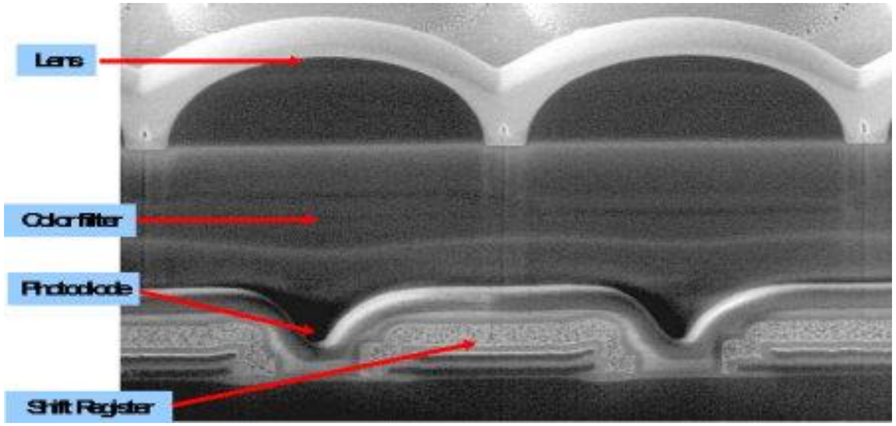
   First layer :Tiny lens → The lens significantly improve CCD sensitivity

   Second layer :Color filter → The role of this part is mainly to help the CCD have the ability of color recognition.

   Third layer :Photodiode

   Fourth layer :Shift Resister

   → This layers convert the light which penetrates the color filter layer into an electronic signal.



fg.1 The Structure of CCD (including 4 main layers)
source:
http://kcs.kcjh.ptc.edu.tw/~spt/computer/digital-image/CCD-CMOS.htm

CCD (Charge-couple Device)

3. How does CCD record images (demonstrate in fg.2):

steps:

(1) An image is projected through a lens onto the capacitor array (the photoactive region).

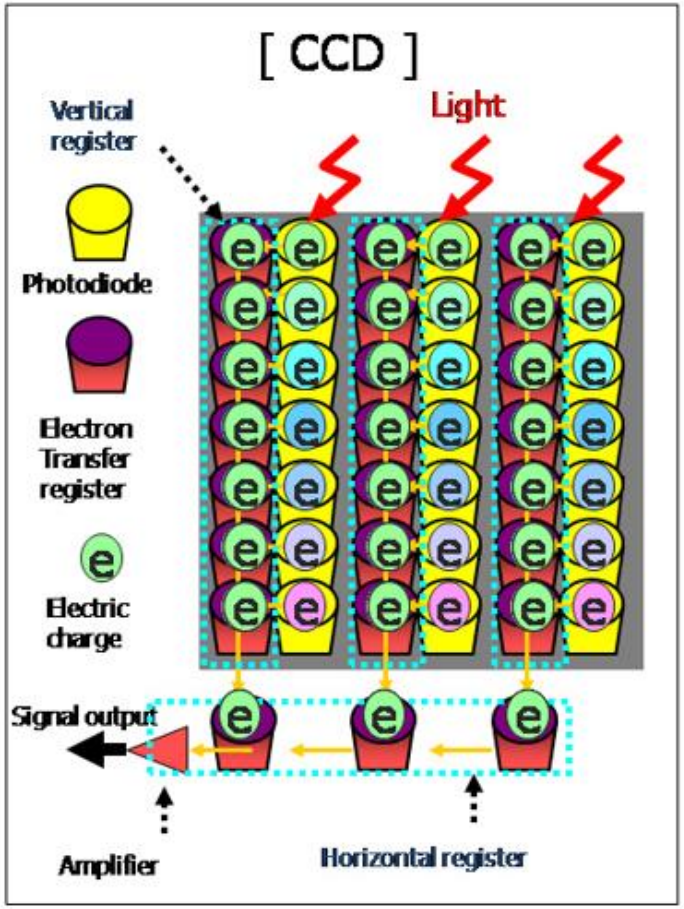
(2) Each capacitor to accumulate an electric charge proportional to the light intensity at that location (due to Photoelectric Effect).

(3) A control circuit causes each capacitor to transfer its contents to its neighbor (operating as a shift register).

(4) The last capacitor in the array dumps its charge into a charge amplifier, which converts the charge into a voltage.

(5) Sequence of voltages

- In digital device → voltages are sampled, digitized, and usually stored in memory.
- In analog device → voltages are processed into a continuous analog signal.



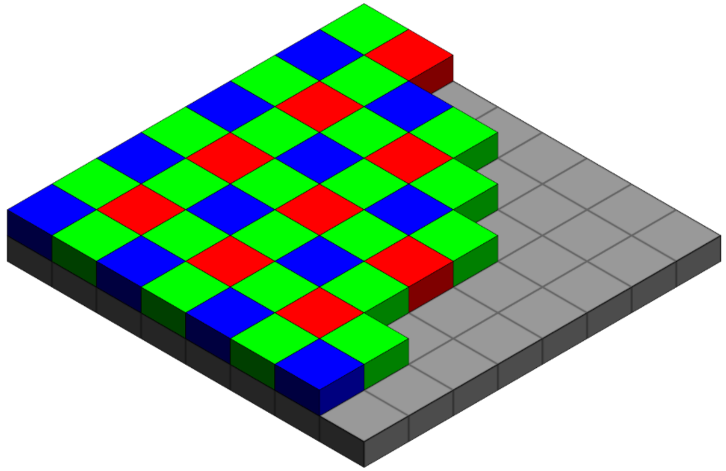
fg.2 Simulate operation of CCD
source:
https://3smarket-info.blogspot.com/2015/08/cmos-ccd.html

CCD (Charge-couple Device)

4. Application:

(1) Bayer filter

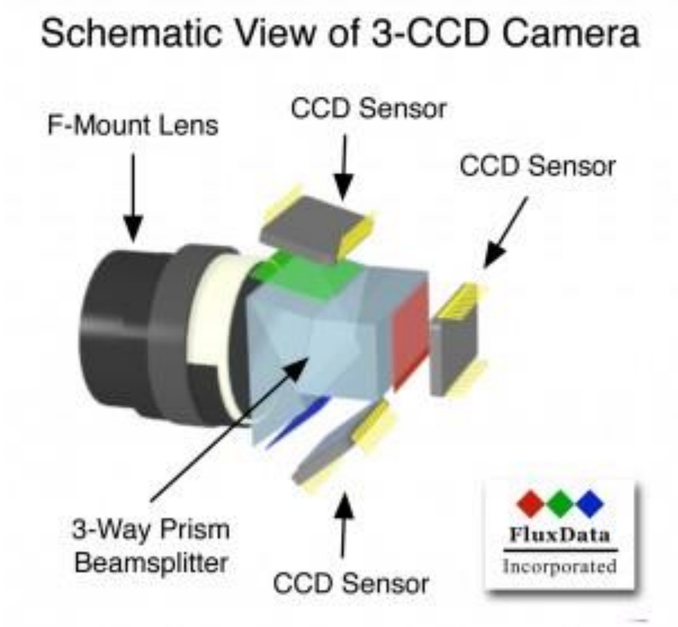
Each square of four pixels has one filtere red, one blue, and two green.



fg.3 Application of CCD (Bayer filter)
source:
https://en.wikipedia.org/wiki/Charge-coupled_device#/media/File:Bayer_pattern_on_sensor.svg

(2) 3CCD camera

Use dichroic beam splitter prism to separate three colors of light.



fg.4 Application of CCD (3CCD camera)
source:
http://www.fluxdata.com/fd-1665-ms9-faqs

CMOS (Complementary metal-oxide-semiconductor)

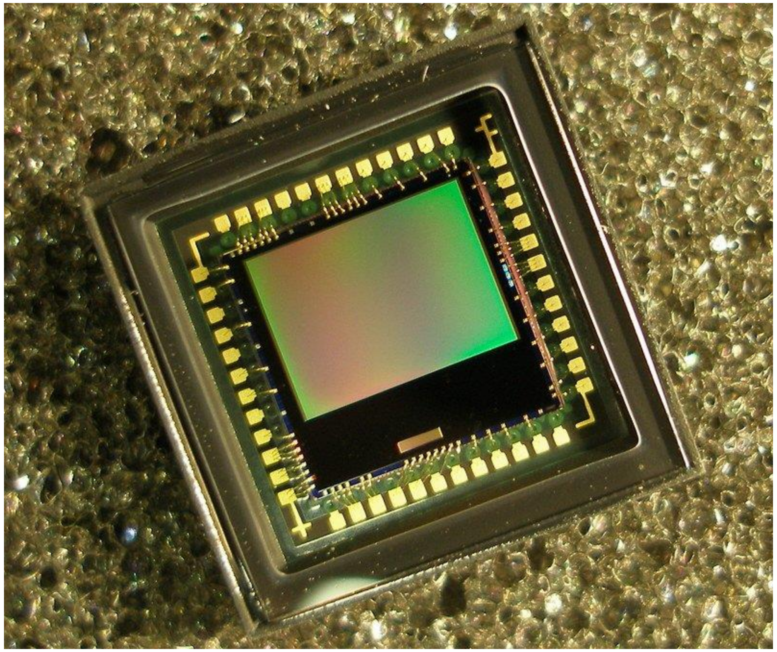
1. Ingredients:

The compositions of CMOS is mainly made up of semiconductors which main material are yttrium and silicon.

2. Structure: (show in fg.5)

The complementary metal-oxide semiconductor (CMOS) sensor consists of millions of pixel sensors,

each of which includes a photodiode.



fg.5 The Structure of CMOS (including millions of pixel sensors)
source:
https://www.newmobilelife.com/2018/03/19/samsung-want-to-take-over-the-camera-cmos-market/

CMOS (Complementary metal-oxide-semiconductor)

3. How does CMOS record images (demonstrate in fg.2):

steps:

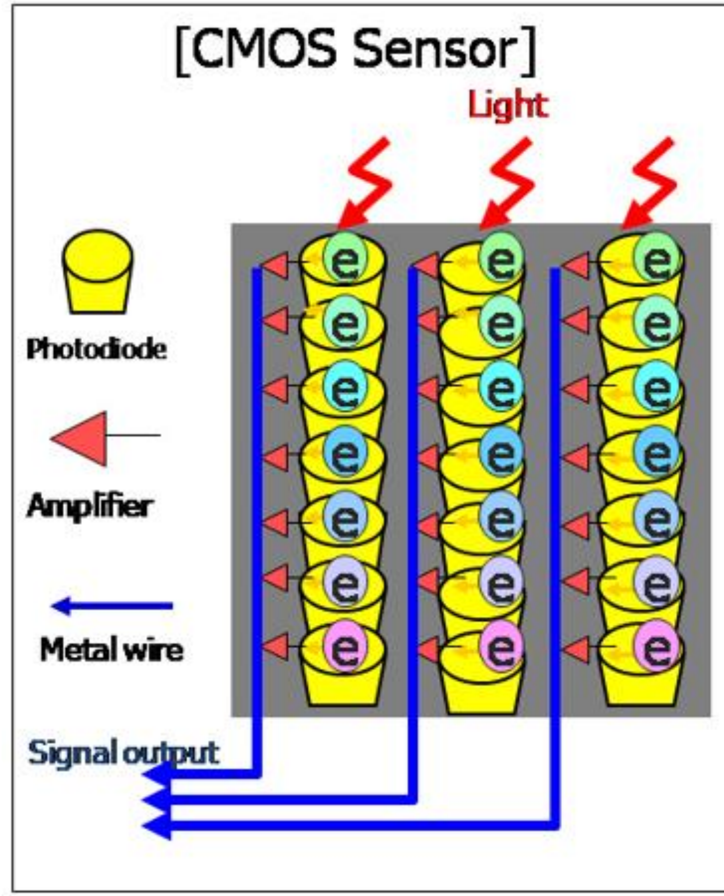
(1) An image is projected through a tiny lens onto the capacitor array (the photoactive region).

(2) Some photodiodes yield electric charge as a result of Photoelectric Effect

(3) Each capacitor to accumulate an electric charge proportional to the light intensity at that location.

(4) Each capacitor in the array dumps its charge into a charge amplifier, which converts the charge into a voltage.

(5) Every voltage will be converted into specific signal. Sometimes, Analog-to-digital converter (ADC) may also be required on the motherboard to convert its output signal into a digital signal.



fg.6 Simulate operation of CMOS
source:
https://3smarket-info.blogspot.com/2015/08/cmos-ccd.html

Compare CCD & CMOS

Compare (show in Table.1)

	System Noise	System Complexity	Sensor Complexity	Signal out of pixel	Signal out of chip	Signal out of camera
CCD	Low	High	Low	Electron packet	Voltage (analog)	Bits (digital)
CMOS	Moderate	Low	High	Voltage	Bits (digital)	Bits (digital)

Table.1 Compare CCD & CMOS

fg. 7 difference of shutter speed
source
https://www.youtube.com/watch?v=Edvpu_939l4

Shutter speed is the length of time.

ISO sensitivity indicates CMOS sensor sensitivity to light.

If you use low ISO speed, you must use the flash to get the same shot.

Increase the ISO speed to use a faster shutter speed.

Spots appearing in photos taken at high ISO speeds. Increasing the ISO sensitivity requires amplifying the power signal, and noise is generated during the process.

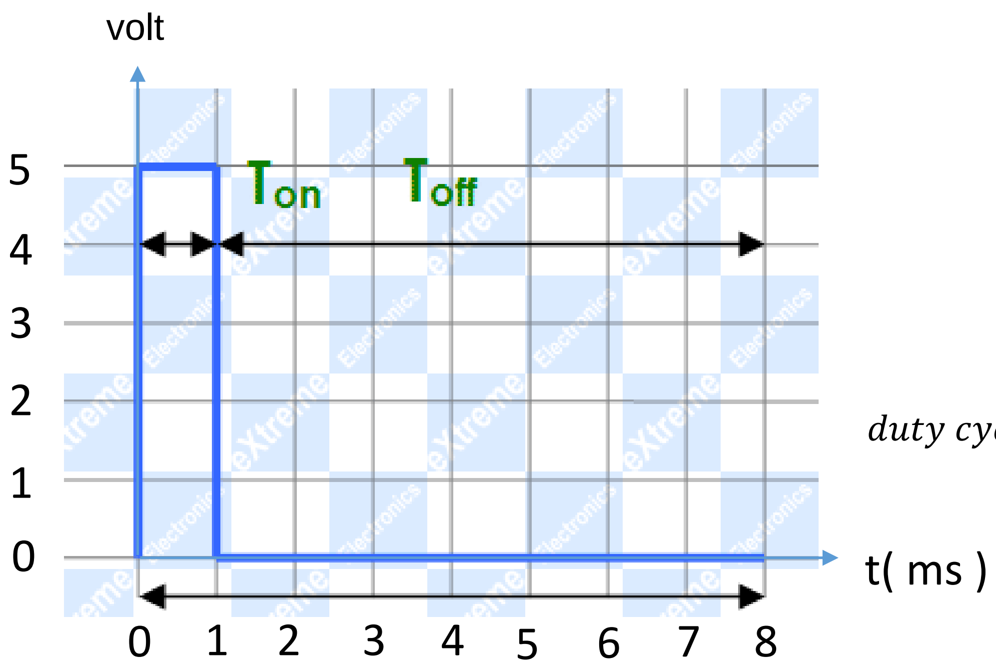
The larger the numerical aperture, the higher the resolution.

Pulse-width modulation (PWM) _ duty cycle

Pulse-width modulation (PWM) is an technique that enable the digital signal “simulate” analog signal .

We need to talk about the duty cycle first . Duty cycle for digital signal describe a the proportion between the time being HIGH and the entire period . For example , see Fig 1 below , the HIGH signal is 5V and the LOW signal is 0V . The time at HIGH is for 1(ms) and the period is for 8(ms) so the duty cycle

is: $\frac{1}{8} \cdot 100\% = 12.5\%$



$$duty\ cycle = \frac{t_{HIGH}}{t_{period}} \cdot 100\%$$

Fig 1
(from: https://reurl.cc/4gonvY)

Pulse-width modulation (PWM) _ principle

Knowing what duty cycle is , we can talk about PWM now . PWM is actually using duty cycle to simulate analog signal . For Fig 1 last page , the duty circle of that period of time is 12.5% . And the item would receive the signal as the $0.125 \cdot 5V = 0.675V$ in the period , But the period should be small enough . To extend the time applied in PWM in same value , we need more pulse of the signal instead of just extending the t_{period} .

For example (see Fig 2) , with an large amount of 80% digital pulse in a short time , the average voltage in the time should be the 80% of voltage HIGH . That’s a value of voltage the item receive

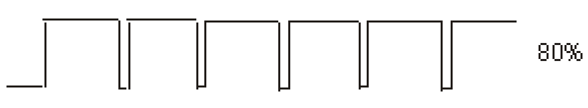


Fig 2
(from: https://reurl.cc/lLD22Y)

Pulse-width modulation (PWM) _ application

1. LED : The digital signal though PWM could transform into different voltage . That is , the output voltage follow the duty cycle and shows different value . The brightness of LED changes with respect to input voltage .
2. Motor : The rotation speed of motor is controlled by input voltage . So the motor could rotate in either fast or slow by the digital signal through PWM .

Arduino main coding

```
//set everthting LOW but STEPPER_PIN[a] HIGH
void pin_HIGH (int a){
  for(int i = 0 ; i<4;i++){
    digitalWrite(STEPPER_PIN[(i+1)%4], LOW);
  }
  digitalWrite(STEPPER_PIN[a], HIGH);
}

void loop() {

//turn clockwise
for(int i = 0 ; i<512;i++){
  pin_HIGH(i%4);
  delay(2);
}

//turn conterclockwise
for(int i = 512 ; i>0 ; i--){
  pin_HIGH(i%4);
  delay(4);
}

}
```

Summary-葉揚昀 108202529

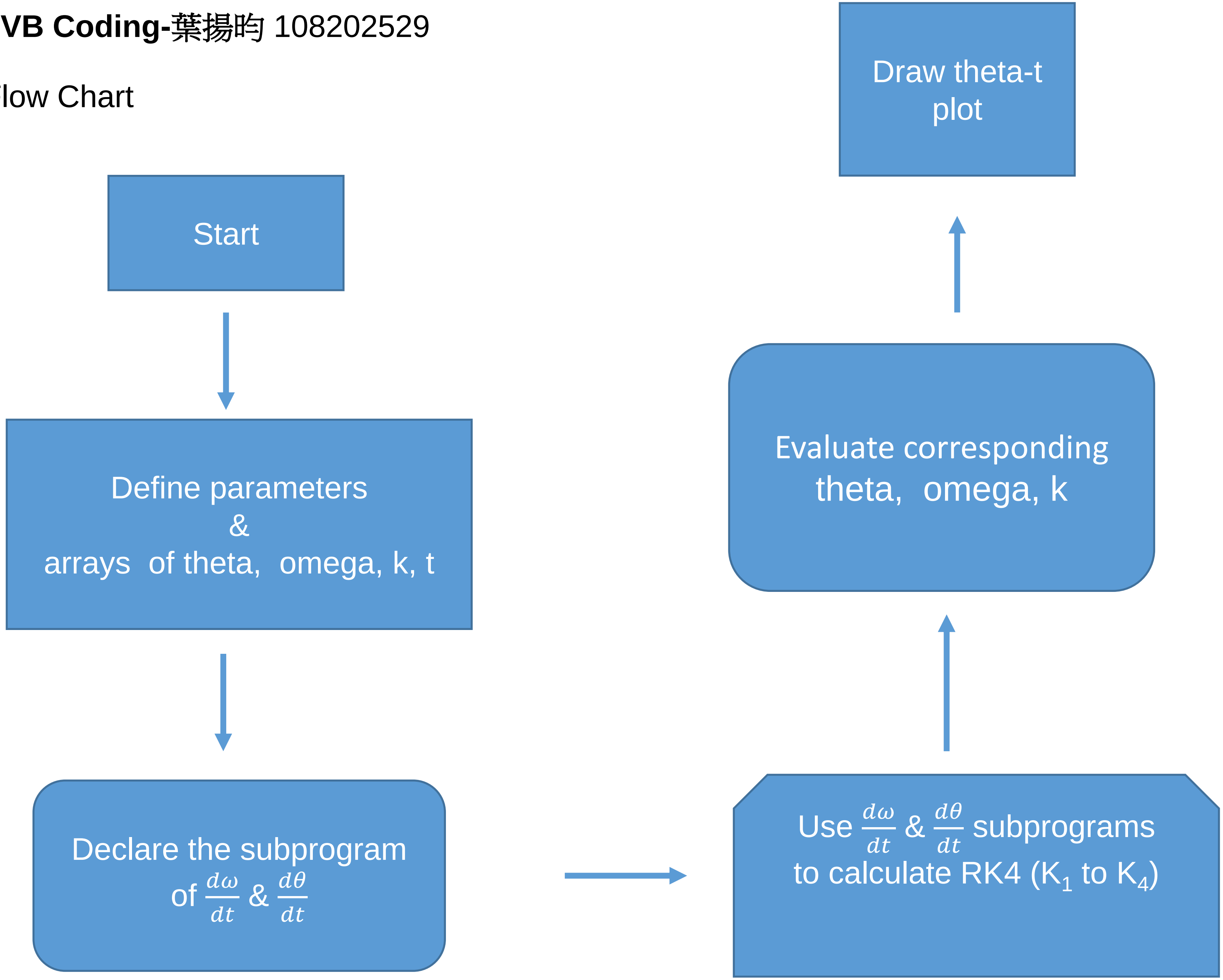
When heard the report of RK4 method which was presented by group 3, I finally had a basic concept about why RK4 use 1, ,2, 2, 1 as product's coefficients in front of k1, k2, k3, k4 . Although didn't realized all of processes of the derivation, I roughly knew that the derivation take advantage of first to forth order differentiations to approximate k1 to k4. Then, expand y_{i+1} after put above calculated functions in k1~k4. Eventually, we get the coefficients (a, b, c, d) = ($\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}$) which make $y_{i+1} = y_i + h\frac{k_1+2k_2+2k_3+k_4}{6}$.

There is an interesting phenomenon mentioned by group 8 Arduino's report. As introduced DC motor, they referred interference of signals due to a magnetic field produced by some inferior coils' induction electromotive force. We know the variation of magnetic flux will make coil generate induced current, then the induced current will yield a magnetic field by the mean time as a result of magnetic effect of current. To simplify, this magnetic field through some ways to interfere the signals which output by Arduino.

In machining, I had some troubles on Thursday and Friday. In one of them, when I used the lathe cutter to smooth a cylinder's end surface, there had a obvious deviation of theta between lathe cutter and end surface. The thing which made the end surface bumpy. To solve this problem, I adopted the teaching assistants' recommendations to feed blade slowly and lock the cylinder more tightly. My progress of machining have completed cylinder, yet I haven't started to make cuboid.

VB Coding-葉揚昀 108202529

Flow Chart



VB Coding-葉揚昀 108202529

Define parameters & arrays of theta, omega, k, t

```
 Dim L As Single = CSng(L_text.Text)
        Dim dt As Single = CSng(time_step_text.Text)
        Dim t(100000.0), theta(100000.0), omega(100000.0) As Single
        Dim k1_theta(100000.0), k2_theta(100000.0), k3_theta(100000.0), k4_theta(100000.0) As Single
        Dim k1_omega(100000.0), k2_omega(100000.0), k3_omega(100000.0), k4_omega(100000.0) As Single
        t(0) = 0
        theta(0) = CSng(initial_theta_text.Text)
        omega(0) = CSng(initial_omega_text.Text)
```

VB Coding-葉揚昀 108202529

Declare the subprogram of $\frac{d\omega}{dt}$ & $\frac{d\theta}{dt}$

```
    Private Sub Cal_omega_dot(ByVal Cal_omega As Single, ByVal Cal_theta As Single, ByRef Cal_omega_dot As Single)
        Dim g As Single = 9.8
        Dim L As Single = CSng(L_text.Text)
        Dim c As Single = CSng(c_text.Text)
        Cal_omega_dot = -g / L * Sin(Cal_theta) - c * Cal_omega
    End Sub

    Private Sub Cal_theta_dot(ByVal Cal_omega As Single, ByRef Cal_theta_dot As Single)
        Cal_theta_dot = Cal_omega
    End Sub
```

VB Coding-葉揚昀 108202529

Use $\frac{d\omega}{dt}$ & $\frac{d\theta}{dt}$ subprograms to calculate RK4 (K_1 to K_4)

```
For i As Integer = 1 To 50000 Step 1
            Cal_omega_dot(omega(i - 1), theta(i - 1), k1_omega(i - 1))
            Cal_theta_dot(omega(i - 1), k1_theta(i - 1))
            Cal_omega_dot(omega(i - 1) + dt / 2 * k1_omega(i - 1), theta(i - 1) + dt / 2
* k1_theta(i - 1), k2_omega(i - 1))
            Cal_theta_dot(omega(i - 1) + dt / 2 * k1_theta(i - 1), k2_theta(i - 1))
            Cal_omega_dot(omega(i - 1) + dt / 2 * k2_omega(i - 1), theta(i - 1) + dt / 2
* k2_theta(i - 1), k3_omega(i - 1))
            Cal_theta_dot(omega(i - 1) + dt / 2 * k2_theta(i - 1), k3_theta(i - 1))
            Cal_omega_dot(omega(i - 1) + dt * k3_omega(i - 1), theta(i - 1) + dt *
k3_theta(i - 1), k4_omega(i - 1))
            Cal_theta_dot(omega(i - 1) + dt * k3_theta(i - 1), k4_theta(i - 1))
            omega(i) = omega(i - 1) + 1 / 6 * dt * (k1_omega(i - 1) + 2 * k2_omega(i - 1)
+ 2 * k3_omega(i - 1) + k4_omega(i - 1))
            theta(i) = theta(i - 1) + 1 / 6 * dt * (k1_theta(i - 1) + 2 * k2_theta(i - 1)
+ 2 * k3_theta(i - 1) + k4_theta(i - 1))
            t(i) = t(i - 1) + dt
```

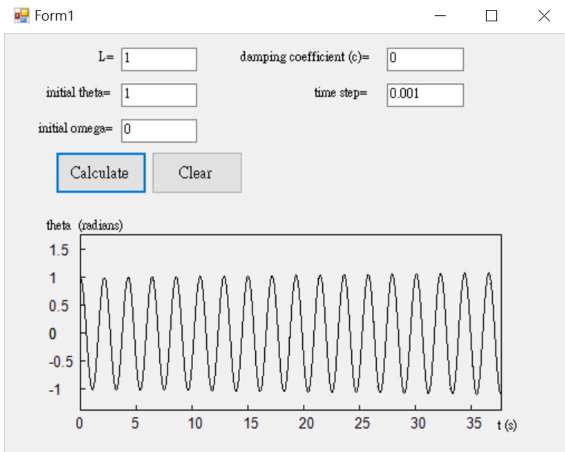
VB Coding-葉揚昀 108202529

Evaluate corresponding theta, omega, k & draw theta-t plot

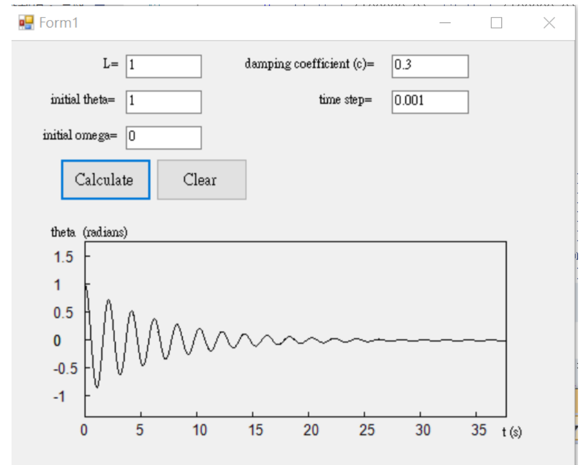
```
            omega(i) = omega(i - 1) + 1 / 6 * dt * (k1_omega(i - 1) + 2 * k2_omega(i - 1) + 2 * k3_omega(i - 1) + k4_omega(i - 1))
            theta(i) = theta(i - 1) + 1 / 6 * dt * (k1_theta(i - 1) + 2 * k2_theta(i - 1) + 2 * k3_theta(i - 1) + k4_theta(i - 1))
            t(i) = t(i - 1) + dt


            a.DrawLine(p, 10 * t(i - 1) + 30, -50 * theta(i - 1) + ph_half, 10 * t(i) + 30, -50 * theta(i) + ph_half)
            If 10 * t(i) > (pw - 50) Then Exit For
        Next
    End Sub
```

VB Coding-葉揚昀 108202529

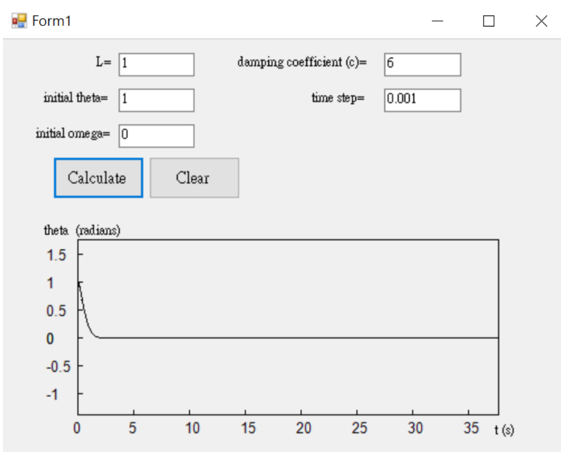
Result (L=1(m) , $\theta_0 = 1(radian)$, $\omega_0 = 0(1/s)$, dt = 0.001(s))



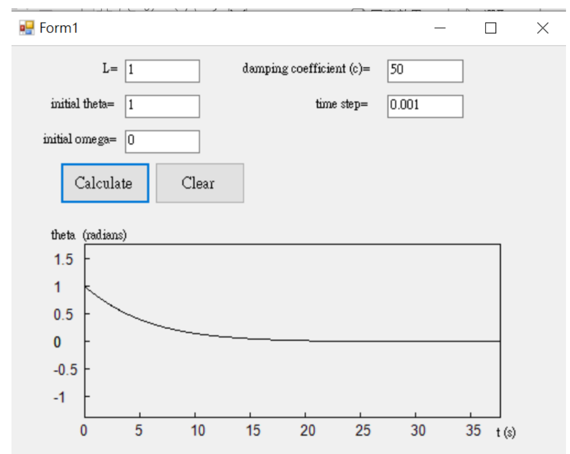
Fg.1 harmonic oscillation (c=0)



Fg.2 under damping (c=0.3)



Fg.3 critical damping (c=6)



Fg.4 over damping (c=50)

summary

What I learn today is about the proof of RK4 and some thinking.

For the RK4 , I didn't search for the proof of it before the class . I just have a only a roughly conception about it for it is much more precise than the Euler's method . Thank to the 4th group I catch the main idea behind equation . Although I still don't understand some procession . Fig1 shows the original form of RK4 method . It's the last step of the proof. The all $\sum h^n f^n$ at the right side of y_0 is corresponding with $\frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ as we have known . The full number of the equation is supposed to add for infinite $h^n f^n$. So the deviation between the real answer and our estimated answer is sigma after $h^5 f^5$. At the meanwhile , Euler's method stop at $h^1 f^1$. So the it's clear why RK4 is more approach the right answer than Euler's method .

Another thing I get to say is the problem at the end of our oral representation . One of the Teaching Assistant ask me when should I choose using RK4 or Euler's method . I didn't give the answer at that moment . I think the it depend on the computer . As RK4 need more computation than Euler's method , it take more time to calculate . So I guess we should Euler's method when drawing the animation and calculating simultaneously . And use RK4 to compute the data , which take time but worth .

$$y = y_0 + h f(y_0) + \frac{h^2}{2} f^2(y_0) + \frac{h^3}{6} f^3(y_0) + \frac{h^4}{24} f^4(y_0) + O(h^5)$$

Fig1
the original form of RK4
(From group 4)

VB_practice_code

- 'theta_0o is point about to puting into the K1'
- theta_0o = theta_0(CInt(t / dt) - 1)
- theta_1o = theta_1(CInt(t / dt) - 1)
- 'theta_0I(0) is K1 , theta1I(0) is L1'
- theta_0I(0) = theta_1o
- theta_1I(0) = -g / l * theta_0o - c / (M * l) * theta_1o

- 'theta_0o is point about to puting into the K2'
- theta_0o = theta_0(CInt(t / dt) - 1) + dt / 2 * theta_0I(0)
- theta_1o = theta_1(CInt(t / dt) - 1) + dt / 2 * theta_1I(0)
- 'theta_0I(1) is K2 , theta1I(1) is L2'
- theta_0I(1) = theta_1o
- theta_1I(1) = -g / l * theta_0o + -c / M / l * theta_1o

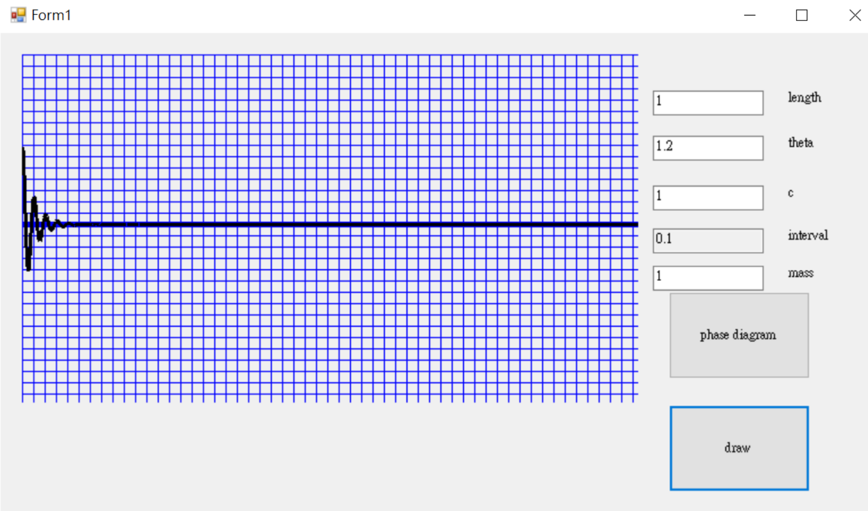
- 'theta_0o is point about to puting into the K3'
- theta_0o = theta_0(CInt(t / dt) - 1) + dt / 2 * theta_0I(1)
- theta_1o = theta_1(CInt(t / dt) - 1) + dt / 2 * theta_1I(1)
- 'theta_0I(2) is K3 , theta1I(2) is L3'

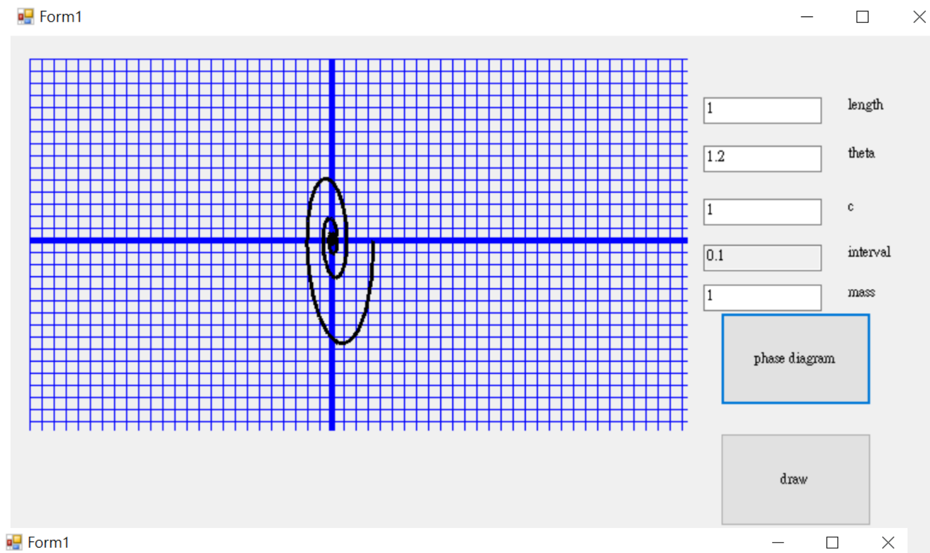
VB_practice_code

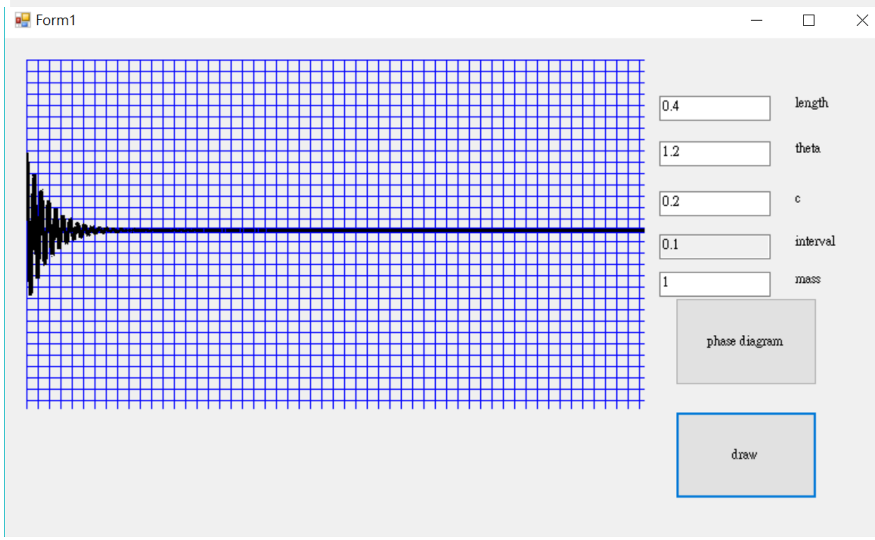
- 'theta_0o is point about to puting into the K4'
- theta_0o = theta_0(CInt(t / dt) - 1) + dt * theta_0I(2)
- theta_1o = theta_1(CInt(t / dt) - 1) + dt * theta_1I(2)
- 'theta_0I(3) is K4 , theta1I(3) is L4'
- theta_0I(3) = theta_1o
- theta_1I(3) = -g / l * theta_0o + -c / M / l * theta_1o

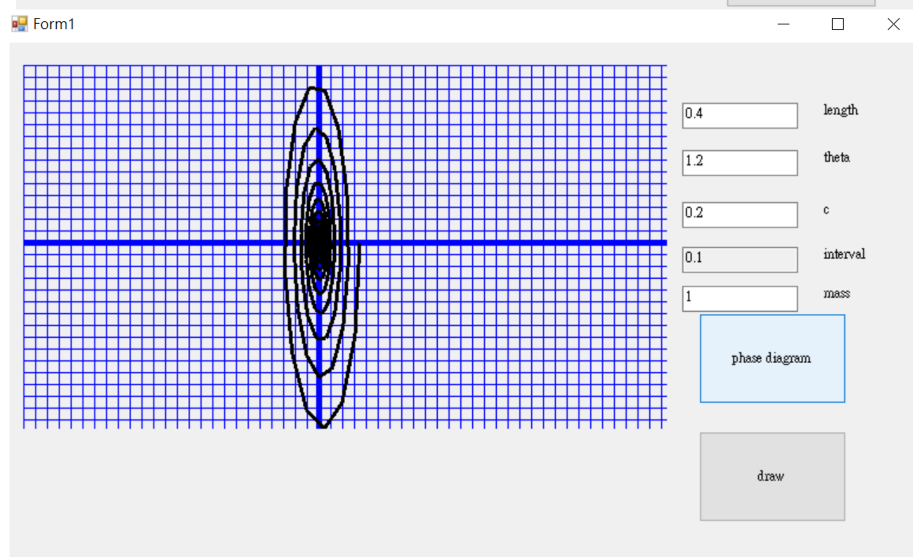
- 'Calculate the average K and L , and then output the nex point'
- theta_0(CInt(t / dt)) = theta_0(CInt(t / dt) - 1) + dt / 6 * (theta_0I(0) + 2 * theta_0I(1) + 2 * theta_0I(2) + theta_0I(3))
- theta_1(CInt(t / dt)) = theta_1(CInt(t / dt) - 1) + dt / 6 * (theta_1I(0) + 2 * theta_1I(1) + 2 * theta_1I(2) + theta_1I(3))

VB_practice_result









VB_practice_flow chart

Create variable

RK4 method

Obtain calculated array of θ and θ'

Draw $\theta - t$ plot

Draw phase diagram

$$K_1\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix}\right), L_1\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix}\right)$$

$$K_2\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix} + \frac{dt}{2}\begin{bmatrix} 1 \\ K_1 \\ L_1 \end{bmatrix}\right), L_2\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix} + \frac{dt}{2}\begin{bmatrix} 1 \\ K_1 \\ L_1 \end{bmatrix}\right)$$

$$K_3\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix} + \frac{dt}{2}\begin{bmatrix} 1 \\ K_2 \\ L_2 \end{bmatrix}\right), L_3\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix} + \frac{dt}{2}\begin{bmatrix} 1 \\ K_2 \\ L_2 \end{bmatrix}\right)$$

$$K_4\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix} + \frac{dt}{2}\begin{bmatrix} 1 \\ K_3 \\ L_3 \end{bmatrix}\right), L_4\left(\begin{bmatrix} t \\ \theta \\ \theta' \end{bmatrix} + dt\begin{bmatrix} 1 \\ K_3 \\ L_3 \end{bmatrix}\right)$$

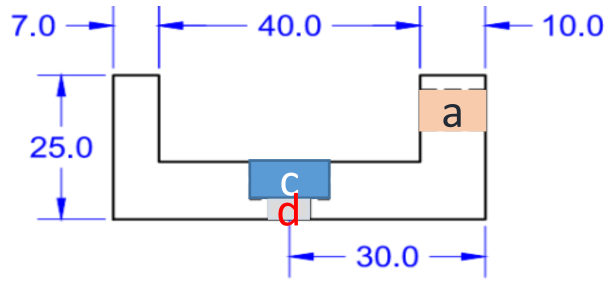
$$K = \frac{(K_1 + 2K_2 + 2K_3 + K_4)}{6}$$

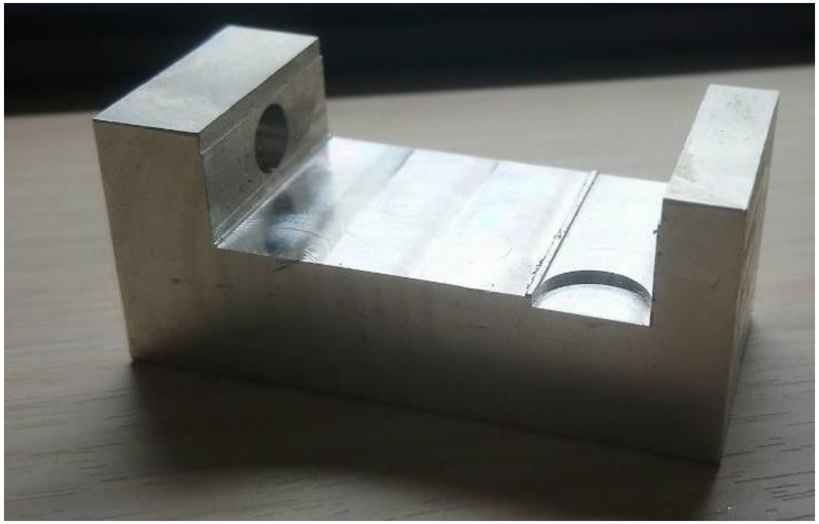
$$L = \frac{(L_1 + 2L_2 + 2L_3 + L_4)}{6}$$

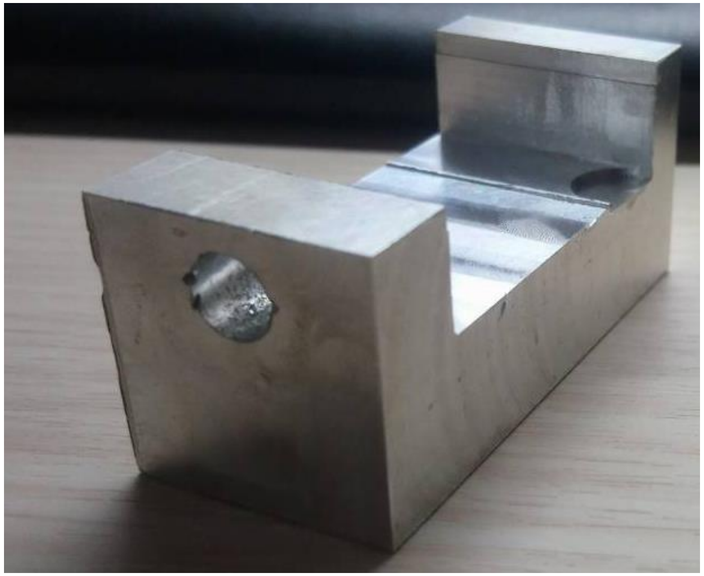
machining project

I am finish milling the 6 surfaces and digging “a” area yet without drilling . And I am milling the groove now . I’ve milled about 11 (mm) now .

For the cylinder , I’ve not started yet .







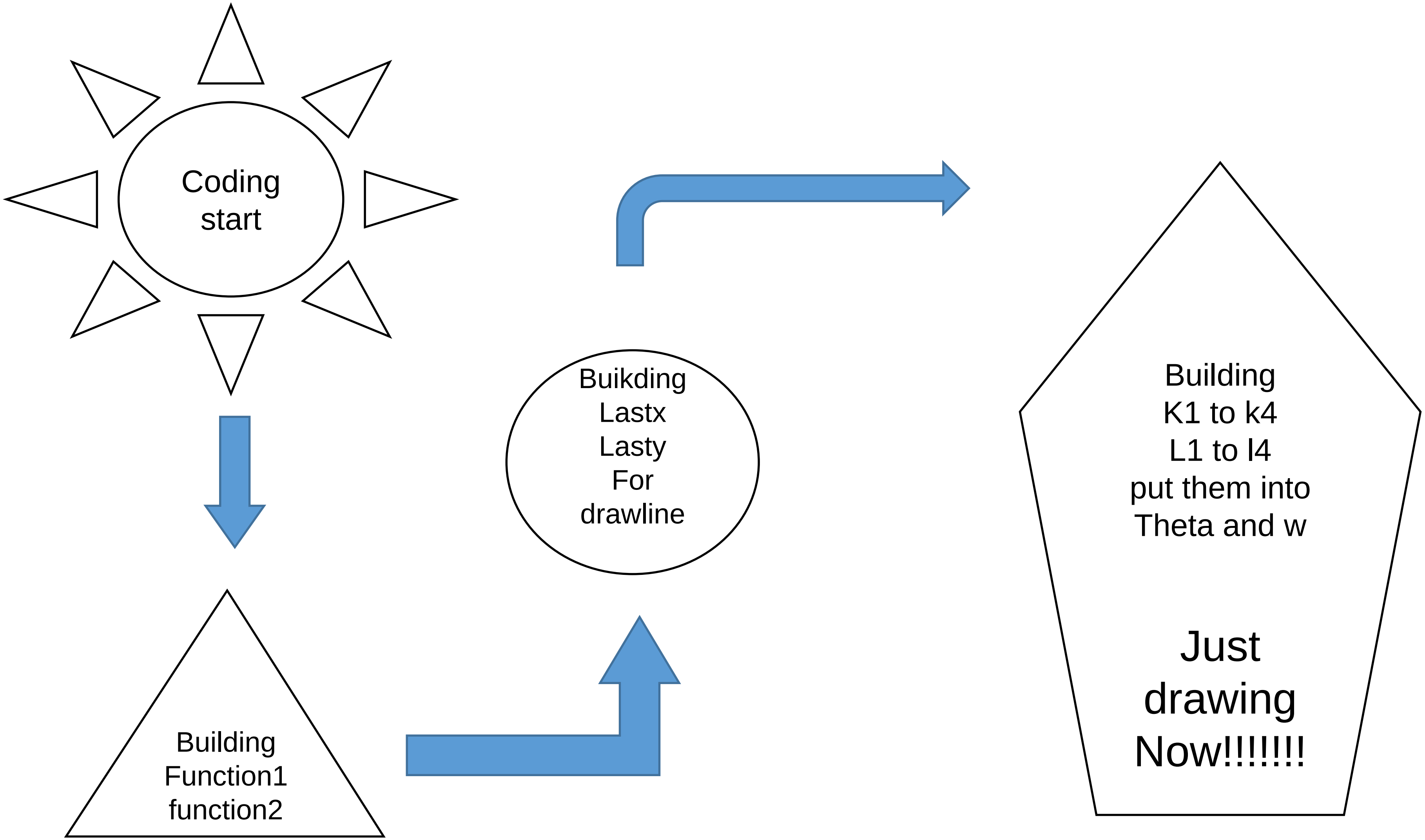
Summary-張家菖

This week I learned some knowledge which I will classify into seven types. First, I learned how to operate a lathe. It is a different experience. Rather than watching the videos, I can notice more details when I operated it by myself. Take how to raise the tool, and the scale on the machine may not be accurate for example. In addition to paying attention to various details, I advise that we leave someplace to correct errors. Secondly, our team used Arduino first time. Controlling a motor, trying to code a program, and connecting two hard wares, these were all fantasy. Thirdly, three kinds of motors were reported, which gave me other aspects of them. We can focus on their principles, functions, accuracy, and limitations. Forth, I had known the deductions of rk2. However, I never presumed rk4 by myself. Thanks to another team, they made me see the whole deductions of rk4. Fifth, I understood the two types of signals and their differences from another perspective, including their representation, description, transmit, and range. Sixth, I knew some formats about coding, report, preview, and so on. The more formal report we need to do, the more details we need to notice, including formats of figures and specifications of simple schematics. Seventh, I noticed differences between my coding and my team member’s coding. I used the function, named function, and he used the array and sub-functions. This phenomenon evoked my inspiration. I want to try more functions of visual basic.

The progress of my machining project.

The aluminum block cylinder has only the tapping teeth left.

The cuboid has not yet begun.


Coding start
Building Function1 function2
Buikding Lastx Lasty For drawline
Building K1 to k4 L1 to l4 put them into Theta and w
Just drawing Now!!!!!!!

```
Public Function f1(ByRef theta1 As Decimal, ByRef w1 As Decimal, ByRef time1 As Decimal)
        Dim l As Single = Val(TextBox1.Text)          '長度
        Dim b As Single = Val(TextBox3.Text)          '阻尼
        Dim theta As Single = Val(TextBox2.Text)      '角度(弧度)
        Return w1
    End Function

Public Function f2(ByRef theta2 As Decimal, ByRef w2 As Decimal, ByRef time2 As Decimal)
        Dim l As Single = Val(TextBox1.Text)          '長度
        Dim b As Single = Val(TextBox3.Text)          '阻尼
        Dim theta As Single = Val(TextBox2.Text)      '角度(弧度)
        Return -9.8 / l * Sin(theta2) - b * w2
    End Function
```

```
For t = 1 To 300 Step 1
            lasty = theta
            lastx = t
            k1 = f1(theta, w, t) * h
            l1 = f2(theta, w, t) * h
            k2 = f1(theta + 0.5 * h * k1, w + 0.5 * h * l1, t + 0.5 * h) * h
            l2 = f2(theta + 0.5 * h * k1, w + 0.5 * h * l1, t + 0.5 * h) * h
            k3 = f1(theta + 0.5 * h * k2, w + 0.5 * h * l2, t + 0.5 * h) * h
            l3 = f2(theta + 0.5 * h * k2, w + 0.5 * h * l2, t + 0.5 * h) * h
            k4 = f1(theta + h * k3, w + h * l3, t + h) * h
            l4 = f2(theta + h * k3, w + h * l3, t + h) * h
            theta = theta + (k1 + 2 * (k2 + k3) + k4) / 6
            w = w + (l1 + 2 * (l2 + l3) + l4) / 6
            g2.DrawLine(penn, lastx, 10 * lasty, t, 10 * theta)
        Next t
```