Kitchen’s Helper

Teaching Handbook

Team 05

B03901026 許凱傑

B03901036 陳博文

B03901142 莊育權

Contents

1. Introduction

Kitchen's Helper

To deal with the fact that people usually can't focus on cutting food in a kitchen, which may result from that you still need to take care of your kids at the same time or you have to answer a call from your friends. Therefore, this automatic cutting machine can let you do other things while cutting food. This kitchen's helper can help you reduce the burden on cooking.

2. Architecture

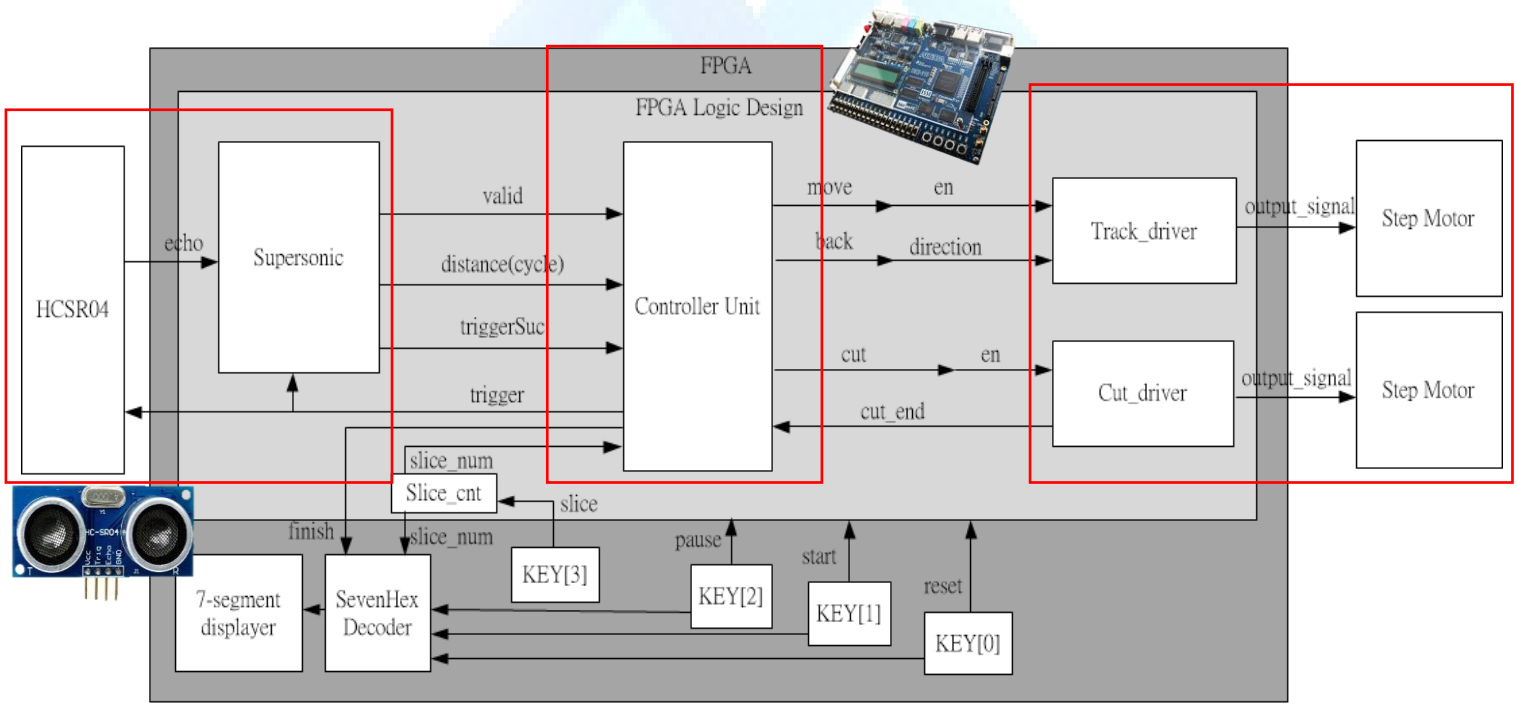

Fig. 1. Architecture of Kitchen's Helper

3. Implementation

Controller

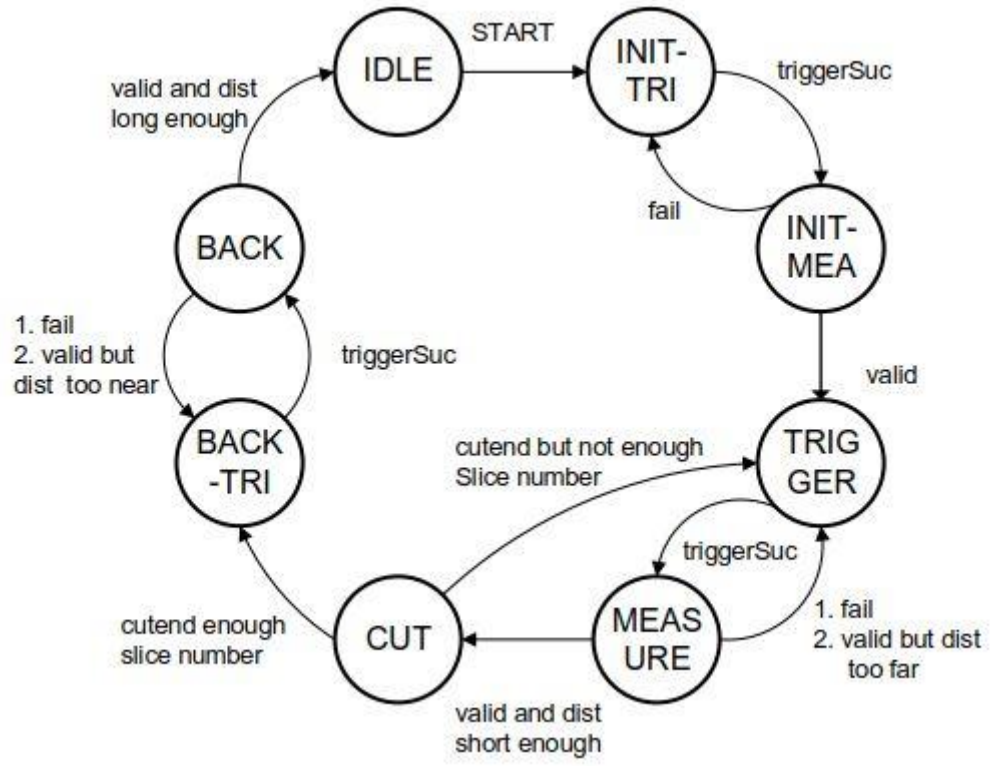

Fig. 2. Finite state machine of controller

Fig. 2. shows the finite state machine of the controller.

There are mainly four stages: initialization, move forward, cut and move backward. At initialization, we will first use HC-SR04 to measure the length of the object and divide it into desired pieces. For every use of HC-SR04, it will send *trigger* to *supersonic module* for at least 10 μs and there should be at least 50 ms spacing between next trigger. In our implementation, too long trigger time or too short spacing will cause it to malfunction, possibly due to interference. Next, the left and right step motor will receive *move* and move forward. Once the distance difference is smaller than the length of the segment, it will stop moving and go to *CUT*. Note that the step motor moves while triggering because the actual measure time is too short (about 2 ms). After step motor for cutting moving downward and upward, it will go back to *TRIGGER* if there is not enough slice number or it will go to *BACK-TRI*. During *BACK-TRI* and *BACK*,

the left and right step motor will move backward until HC-SR04 detect the distance larger than the length of the object.

Supersonic

We use *supersonic module* to simulate the operation of HC-SR04. It will send *triggerSuc* to *controller* once it receive *echo* from low to high. If it detect *echo* from high to low, it will calculate the number of cycles of the TTL pulse and transmit as *distance* and send *valid* at the same time.

However, the *echo* may not be detected from high to low, so we set the maximum number of cycles. If it doesn't detect echo from high to low within this limit, it will send *fail* to *controller* and ask it to re-trigger HC-SR04.

HCSR04

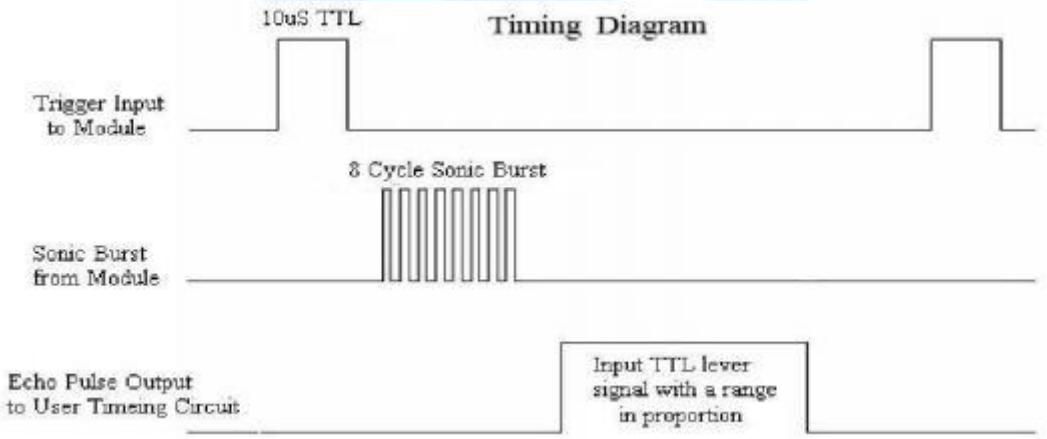

Fig. 3. Timing diagram of HC-SR04.

Fig. 3. shows Timing diagram of HC-SR04. After 10 μs TTL triggering, HC-SR04 will send 8 cycles square wave right away and raise high *echo* simultaneously. Once HC-SR04 detect reflected signal, it will turn *echo* from high to low and awaits next triggering. One thing must be very careful when using HC-SR04, is the exact trigger time and enough space time between two triggering. Otherwise, it will stuck on triggering stage.

Track_driver

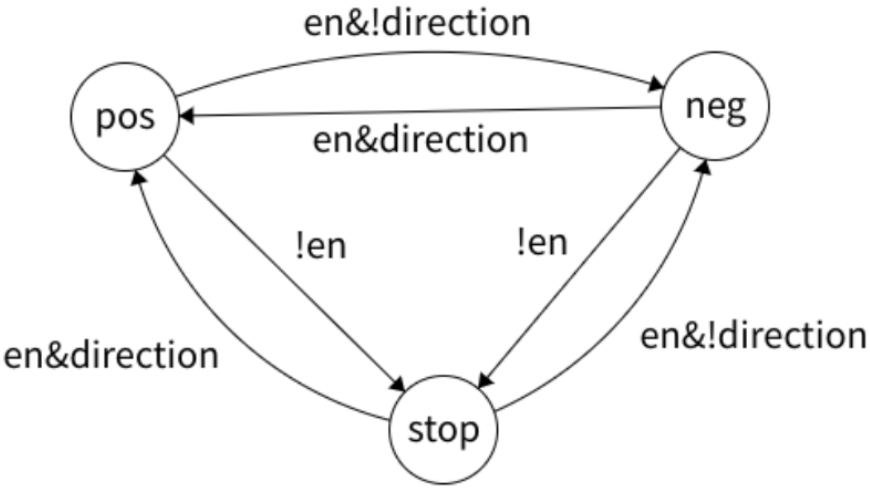

Fig. 4. Finite state machine of track driver

Since the clock frequency from controller is 4×10^{-9}, which is too fast for our motors. Thus, we implement a sub-module, clock_div. In clock_div, we take the system clock as input and divide system clock cycle down to a slower clock with larger cycle. We set define_speed to certain value where$(2 * desiredclock)/system\ clock$, which act as a threshold for the clock divider. We keep a counter which increases by one after each positive edge of system clock. After the counter exceeds the threshold, we reset the counter and invert the output value, in other words, create a posedge or a negedge depending on previous clock value.

After the **Fig. 4.** shows the finite state machine of track driver. There are mainly three states for the fst: **pos**, **neg** and **stop**. At the beginning, we start from **stop** stage. And the input signals, en and direction, from controller specify whether to drive motors and the direction of motors. This driver module is used to drive the module of NXT step motor. Double H driver module uses ST L298N dual full-bridge driver, an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage. Below specifies the function of each pin:

PIN	FUNCTION
VMS	VMS is the 5V-35V power source for motor. VMS is positive, GND is negative
5V	Power input for the logic circuit on the board
5V-EN	5V source jumper. When the jumper is put on, The 78M05 supplies the 5V power for logic circuit on the board from the VMS port(7V < VMS < 18V). The power of logic circuit of the board is supplied by the 5V port when this jumper put off
U1/2/3/4	Pull up resistor for IN1/2/3/4. Putting on the jumper enable the pull up resistor for the microcontroller, putting off it disable it
EA/EB	EA/EB is the enable pin for the two motor A/B, the motor speed also can be controlled by the PWM of this pin
IN1/2/3/4	IN1/2/3/4 is the pin for the motor control. Motor B is same as Motor A

Below figures show the NXT step motor and L298.

Fig. 5. NXT step motor

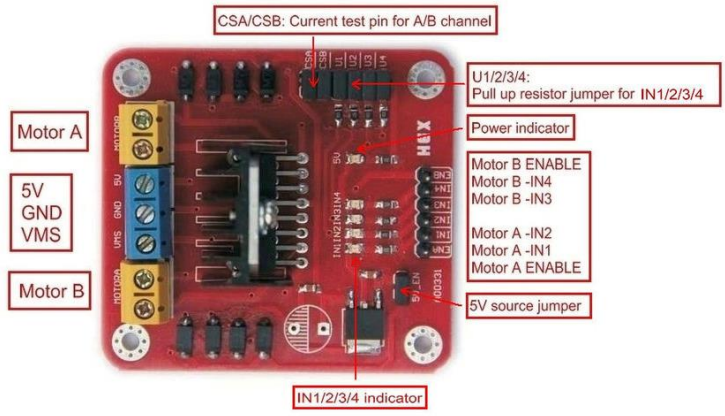

Fig. 6. L298 and the function of each pin

Cut_driver

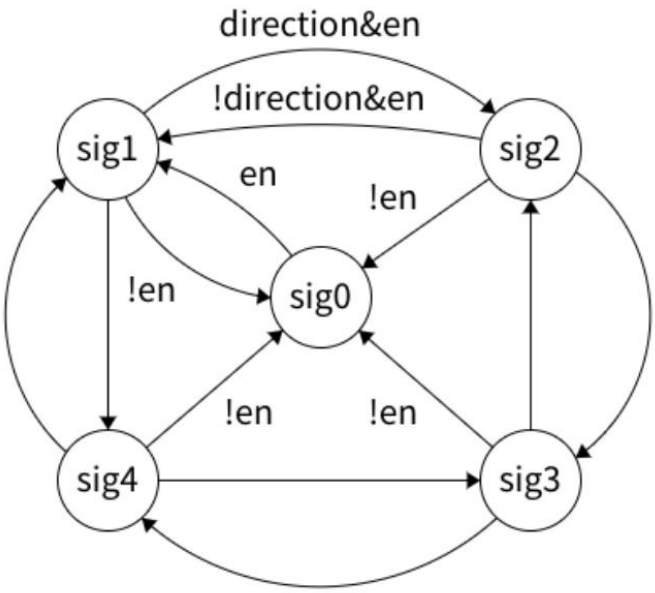

Fig. 7. Finite state machine of cut_driver.v

Fig. 7. shows the finite state machine of the controller. There are mainly five states for the FST: **sig0**, **sig1**, **sig2**, **sig3**, **sig4**. At the beginning, we start from **sig0** stage. And the input signals, en and direction, from controller, specify whether to drive motors and the direction of motors. The output four-bit signal signal_o is corresponding to full-stepping mode which consumes more power but with larger torque. This driver module is used to drive the module of step motor 28ybj-48. The 28BYJ-48 is a small, cheap, 5 volt geared stepping motors. Below figure shows the step motor 28ybj-48 and the stepping methods of step motor.

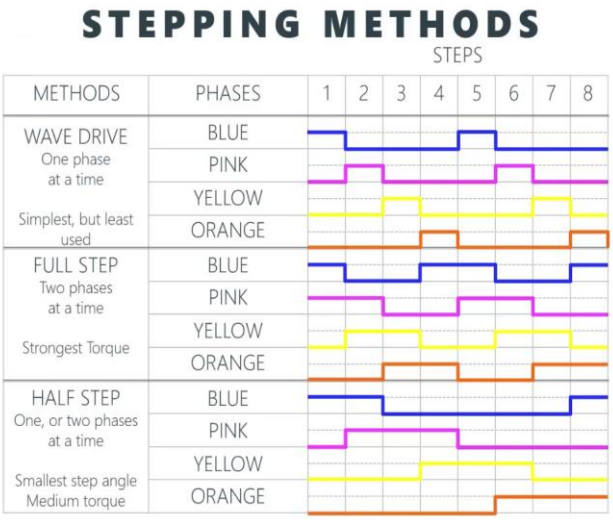

Fig. 8. Stepping method of step motor

Fig. 9. 28BYJ-48 Stepper Motor

4. Debugging

1. All the GND must be connected together. Otherwise, the current may not be strong enough.
2. Exact trigger time and enough space time between two triggering of HC-SR04. Otherwise, it will stuck on triggering stage.
3. A buffer distance for evaluation distance long (short) enough. Because the step motor is not sensitive enough that it might move a bit even when *move* already turns from high to low.

5. Demo

Demo Video

https://www.youtube.com/watch?v=INd2CLuVJFw&feature=youtu.be

Demo PPT

https://github.com/kevin71104/DCLab/tree/master/final

6. Future Work

Provide More Flexible Options

Since now we only can cut a thing into the number of power of 2, we want to provide the function that can cut things into any number.

Enhance the Horsepower of Motors

Since considering the weight of the motors can’t too heavy to put on the machine, now the two step motors we used doesn’t have strong horsepower , which leads to the machine move slowly and only can cut something soft. Therefore, we want to enhance the horsepower of motors to make the machine more powerful.

7. Reference

1. http://www.micropik.com/PDF/HCSR04.pdf

2. http://www.tau.ac.il/~stoledo/lego/Stepper/

3. https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf