Lesson 6 Multi-channel Servo Speed Control

1. Project Purpose

Learn the speed control method of servo to control multiple servos to rotate at different speeds.

2. Project Principle

Servo movement speed: the instantaneous velocity of the servo is determined by the cooperation of the internal DC motor and the variable speed gear set. Driven by the constant voltage, its value is unique. For digital PWM servo, its speed is determined by its internal program. Its average movement speed can be changed by the control method of segmented pause.

For example: divide a rotation with an action amplitude of 90° into 128 stop points, and achieve an average speed of 0°-90° change by controlling the time of each stop point. For most servos, the unit of speed is determined by "degrees/second".

3. Program Analyst

1) ServoSetPluseAndTime is the function for setting the target position and rotation time of servo rotation.

2) Its function is simple, which is to check whether the servo number is between 0 and 7 and whether the position is between 500 and 2500. If not, exit the function directly. Otherwise, check the time to ensure that it is between 20 and 30000. Then write the position into the array, write the time into the variable and set the sign that the servo has been set.

3) The figure below shows the function that actually controls the servo rotation.

```
void ServoPwmDutyCompare(void)//Pulse width change and speed control
{
    uint8 i;

    static uint16 ServoPwmDutyIncTimes; //Increasing times
    static bool ServoRunning = FALSE;   //The servo is moving at the specified speed to the position corresponding to the specified pulse width
    if(ServoPwmDutyHaveChange)//ServoRunning == FALSE &&     Stop running,calculate when the pulse width changes
    {
        ServoPwmDutyHaveChange = FALSE;
        ServoPwmDutyIncTimes = ServoTime/20;    //Use this sentence when ServoPwmDutyCompare() function is called once every 20ms
        for(i=0;i<8;i++)
        {
            //if(ServoPwmDuty[i] != ServoPwmDutySet[i])
            {
                if(ServoPwmDutySet[i] > ServoPwmDuty[i])
                {
                    ServoPwmDutyInc[i] = ServoPwmDutySet[i] - ServoPwmDuty[i];
                    ServoPwmDutyInc[i] = -ServoPwmDutyInc[i];
                }
                else
                {
                    ServoPwmDutyInc[i] = ServoPwmDuty[i] - ServoPwmDutySet[i];

                }
                ServoPwmDutyInc[i] /= ServoPwmDutyIncTimes;//Pulse width for each increment
            }
        }
        ServoRunning = TRUE;    //servo starts running
    }
    if(ServoRunning)
    {
        ServoPwmDutyIncTimes--;
        for(i=0;i<8;i++)
        {
            if(ServoPwmDutyIncTimes == 0)
            {       //The last increment will directly assign the set value to the current value

                ServoPwmDuty[i] = ServoPwmDutySet[i];

                ServoRunning = FALSE;   //arrive the set position,servo stops running
            }
            else
            {

                ServoPwmDuty[i] = ServoPwmDutySet[i] +
                    (signed short int)(ServoPwmDutyInc[i] * ServoPwmDutyIncTimes);

            }
        }

    }
}

```

4) When ServoPwmDutyHaveChange is set to true, the step pulse width of each steo and the number of steps to be run are calculated according to the target position and movement time. After calculating, the sign of SerRunning will be set to true. Then the pulse width of the servo will be closer to the target pulse width.