



```
int latchPin = 12;      // Pin connected to ST_CP of 74HC595 (Pin12)
```

```
int clockPin = 13;     // Pin connected to SH_CP of 74HC595 (Pin11)
```

```
int dataPin = 11;      // Pin connected to DS of 74HC595 (Pin14)
```

```
void setup()
```

```
{
```

```
  // set pins to output
```

```
  pinMode(latchPin, OUTPUT);
```

```
  pinMode(clockPin, OUTPUT);
```

```
  pinMode(dataPin, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
  // Define an array to save the pulse width of LED. Output the signal to the 8 adjacent LEDs in order,
  and then it produces the dropping-rain effect
```

```
  const byte pulse[] = {0, 0, 0, 0, 0, 0, 0, 0, 64, 48, 32, 16, 8, 4, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0};
```

```
  // Define a variable to select 8 contiguous data in the array sequentially static byte offset;
```

```

// Define a variable to control the speed
static unsigned int counter;

if (counter++ % 8 == 0)          // Reduce the self-increasing speed of offset
offset < 15 ? offset++ : offset = 0; // offset increases

// Sortie de la forme d'onde PWM
for (int i = 0; i < 64; i++)
{
    // The cycle of PWM is 64 cycles
    byte data = 0;          // Define a variable to represent the output state of this loop
    for (int j = 0; j < 8; j++) // Calculate the output state of this loop
    {
        if (i < pulse[j + offset]) // Calculate the LED state according to the pulse width
        {
            data |= 0x01 << j;    // Represent the LED state with the corresponding bit of a variable
        }
    }

    // Send the state of LED to 74HC595
    writeData(data);
}
}

void writeData(int value)
{
    // Make latchPin output low level
    digitalWrite(latchPin, LOW);

    // Send serial data to 74HC595
    shiftOut(dataPin, clockPin, MSBFIRST, value);

    // Make latchPin output high level, then 74HC595 will update the data to parallel output
    digitalWrite(latchPin, HIGH);
}

```