

ChatGPT as a programmer?

David Stonier-Gibson

Does ChatGPT have any use in computer programming? I took it for a spin.

I saw something online about a fellow who used ChatGPT to write a WordPress plugin for his wife. Well, I barely know what WordPress is, let alone a plugin for it. But I am currently deeply engaged with programming a gadget I designed using an ESP32 board in the Arduino Integrated Development Environment (IDE). So I decided to give ChatGPT a go with that.

One of my favourite little “things” with microcontrollers is making them do, seemingly, many things at once. It’s something I’ve been doing through my whole career of designing electronic machine controls. There are many things going on at once in say a beer bottling and labelling machine, and the controller has to be able to monitor and control them all at once. When I want to teach people my methods, I use the easily understood example of making several LEDs all flash at different, independent rates.

So I decided to challenge CHatGPT with that. It requires a level of sophistication of maybe 3 on a scale of 1 to 10.

The first attempt was a dismal failure – it used the dreaded `delay()` function, so only one LED could blink at once although they would all blink in turn. But I learned a valuable lesson: *Frame your question most carefully, and be prepared to do several runs with better worded questions.*

Here’s the first successful attempt, with my question/request in bold.

Write an Arduino program to flash 5 LEDs asynchronously all at independent rates. The LEDs are on output 5, 6, 7, 8 and 9

```
const int LED1 = 5;
const int LED2 = 6;
const int LED3 = 7;
const int LED4 = 8;
const int LED5 = 9;

unsigned long currentMillis1;
unsigned long currentMillis2;
unsigned long currentMillis3;
unsigned long currentMillis4;
unsigned long currentMillis5;

unsigned long flashRate1 = 100; // Flash rate for LED 1 in milliseconds
unsigned long flashRate2 = 200; // Flash rate for LED 2 in milliseconds
unsigned long flashRate3 = 300; // Flash rate for LED 3 in milliseconds
unsigned long flashRate4 = 400; // Flash rate for LED 4 in milliseconds
unsigned long flashRate5 = 500; // Flash rate for LED 5 in milliseconds

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
}

void loop() {
  currentMillis1 = millis();
  currentMillis2 = millis();
  currentMillis3 = millis();
  currentMillis4 = millis();
  currentMillis5 = millis();

  if (currentMillis1 - flashRate1 >= flashRate1) {
    digitalWrite(LED1, !digitalRead(LED1));
    currentMillis1 = millis();
  }

  if (currentMillis2 - flashRate2 >= flashRate2) {
    digitalWrite(LED2, !digitalRead(LED2));
    currentMillis2 = millis();
  }

  if (currentMillis3 - flashRate3 >= flashRate3) {
    digitalWrite(LED3, !digitalRead(LED3));
    currentMillis3 = millis();
  }

  if (currentMillis4 - flashRate4 >= flashRate4) {
    digitalWrite(LED4, !digitalRead(LED4));
    currentMillis4 = millis();
  }

  if (currentMillis5 - flashRate5 >= flashRate5) {
    digitalWrite(LED5, !digitalRead(LED5));
    currentMillis5 = millis();
  }
}
```

ChatGPT added a description to this one. Very useful for a student.

This program uses the millis() function to keep track of the time elapsed since the Arduino board was powered on. The flash rate for each LED is set in the beginning of the code in milliseconds.

In the loop() function, the current time is compared to the flash rate for each LED. If the current time minus the flash rate is greater than or equal to the flash rate, the LED is turned on or off, depending on its current state.

This allows the LEDs to flash independently of each other at their own rates

Next I decided to amp up the sophistication by asking for the LED to be managed by a class. This is a much more sophisticated method of programming called Object Oriented Programming (OOP). This is probably moving us up from 3 to 5 on our scale of 1 to 10.

Write an Arduino program to flash 5 LEDs asynchronously all at independent rates. The LEDs are on output 5, 6, 7, 8 and 9. Use a class and 5 instances

```
#include <Arduino.h>

class LED {
public:
    LED(int pin, int interval) {
        _pin = pin;
        _interval = interval;
        pinMode(_pin, OUTPUT);
    }

    void update() {
        if (millis() - _lastUpdate > _interval) {
            _lastUpdate = millis();
            digitalWrite(_pin, !digitalRead(_pin));
        }
    }

private:
    int _pin;
    int _interval;
    unsigned long _lastUpdate = 0;
};

LED led5(5, 1000);
LED led6(6, 500);
LED led7(7, 250);
LED led8(8, 125);
LED led9(9, 62.5);

void setup() {
    // Nothing to do in setup
}

void loop() {
    led5.update();
    led6.update();
    led7.update();
    led8.update();
    led9.update();
}
```

I then tried to add a LED toggle function from a push button. I was not able to coax it into making a proper solution that would discriminate the off to on transition of the push button. It failed this one miserably, and I could not coax a working solution out of it without giving away the game in my query. I gave up after three attempts. It always produced something where the LED would continuously toggle while the button was held down, not just once for each press. I wonder if the problem for it was that I was basically asking it to find solutions to two separate challenges, LEDs and button, at the same time.

Write an Arduino program to flash 5 LEDs asynchronously all at independent rates. The LEDs are on output 5, 6, 7, 8 and 9. Use a class and 5 instances. Add a separate asynchronous function that will toggle an LED on output 2 once whenever a push button connected to input 3 is pressed and then released.

```
#include <Arduino.h>

class LED {
public:
    LED(int pin, int delay) {
        _pin = pin;
        _delay = delay;
        pinMode(_pin, OUTPUT);
    }

    void flash() {
        digitalWrite(_pin, !digitalRead(_pin));
        delay(_delay);
    }

private:
    int _pin;
    int _delay;
};

LED led1(5, 50);
LED led2(6, 75);
LED led3(7, 100);
LED led4(8, 125);
LED led5(9, 150);

void toggleLED() { /***** Big fail *****/
    static bool state = false;
    if (digitalRead(3) == HIGH) {
        state = !state;
        digitalWrite(2, state);
    }
}

void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, INPUT_PULLUP);
}

void loop() {
    led1.flash();
    led2.flash();
    led3.flash();
    led4.flash();
    led5.flash();
    toggleLED();
}
```

And this one was very impressive (to me!) ...

Write an Arduino program to flash 5 LEDs asynchronously all at independent rates. The LEDs are on output 5, 6, 7, 8 and 9. Code must be non-blocking. All intervals to be prime numbers.

```
int leds[5] = {5, 6, 7, 8, 9};
int intervals[5] = {2, 3, 5, 7, 11};
unsigned long times[5];

void setup() {
  for (int i=0; i<5; i++) {
    pinMode(leds[i], OUTPUT);
    times[i] = millis();
  }
}

void loop() {
  for (int i=0; i<5; i++) {
    if (millis() - times[i] >= intervals[i]) {
      digitalWrite(leds[i], !digitalRead(leds[i]));
      times[i] = millis();
    }
  }
}
```

... and each time I click the Regenerate button it comes up with a different solution.