

Computational Foundations II (Summer Term 2022) Tutorial 5

Tasks marked with a star like **Optional Task*** are optional. Tasks marked like **Hard Task⁺** are given, but it is not expected that you solve them now. It is great if you learn to solve them during the lecture. Go back to them after a few weeks and see your own progress.

Learning Outcome: PWM, Arduino and Network Programs

Task 8: Pulse Width Modulation

- Explain in a single sentence, how PWM can be used to create analogous information using a digital circuit.
- Draw the signal of a PWM signal with 50Hz and a duty cycle of 75% with time on the X axis and voltage on the Y axis
- Using a high-frequency clock and a counter, implement a PWM signal generator in digital electronics. Tip: Take a counter and a comparator. Let the counter count and overflow from its maximum value to zero. Now, the comparator compares to an integer number specifying the duty cycle (how long on compared to how long is each loop). Simulate your design.

Task 9: Arduino

The Arduino family consists of a set of well-tested PCB boards holding typically AVR microcontrollers that have always been quite efficient in terms of simplicity of external circuitry as they require not much more than a clock source and a suitable energy management.

The Arduino project combines such basic boards with a well-known geometry for extensions (e.g., where are the PINS), a small IDE for the Atmel-AVR-GCC compilers, and a basic firmware that translates operation of such microcontrollers into two functions: `setup()` and `loop()`.

In this way, a typical Arduino program called sketch does not contain a `main`, but rather a `setup` function which is called once at the beginning and a `loop` function which is called endlessly.

If you own an Arduino, this task is easy, if not, I would recommend buying a cheap set (starter kit, sensor kit) for learning. However, you are free to answer the questions after an Internet research as well in “theory”.

A very simple program looks like

```
int buttonPin = 3;

// setup initializes serial and the button pin
void setup() {
  Serial.begin(9600);
```

```
pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop() {
  if (digitalRead(buttonPin) == HIGH) {
    Serial.write('H');
  }
  else {
    Serial.write('L');
  }

  delay(1000);
}
```

- Design and implement a circuit (maybe buy an Arduino Nano and a few LEDs for this) with a blinking LED
- Design and implement the traffic light state machine with LEDs
- Use PWM using ‘analogWrite’ to dim an LED based on a potentiometer read from an A/D pin

Task 10: HTTP Client

Implement a client that downloads the web page <http://www.tum.de/>. Note that it will show a redirect to an encrypted version. Use the Posix socket API (on Linux, MacOS) or the WinSock32 API (link from slides) as a starting point.