

chipKIT Fundamental Projects

Objective: These projects teach someone with no prior experience with electronics, electrical engineering, or programming, the necessary concepts, skills, and techniques to work comfortably with microcontroller-based circuits.

The following provides the titles of the projects together with a brief description of the project's focus. A parenthetical "E" indicates the project introduces new material related to electronics, electrical engineering, or circuits while a "P" indicates that new material is introduced related to programming.

1. **Introduction to MPIDE**
2. **Blink Internal LED (P)** Comments; C++ syntax (function calls, semicolons, braces, etc.); digital signals; `setup()`; `loop()`; `delay()`; `digitalWrite()`; `pinMode()`; INPUT/OUTPUT (defined constants); HIGH/LOW.
3. **Blink External LED (EP)**
Voltage and current; resistors; current-limiting resistors; Ohm's law; IV relationships; LED's (non-linear devices); assignment operator; variables; `const` qualifier; integers and type.
4. **Button-Controlled On/Off LED (EP)**
Pull-up and pull-down resistors; `digitalRead()`; conditional statements; relational operators.
5. **MPIDE Serial Output (P)**
Button-controlled On/Off LED where serial output displays state changes; `millis()`; `micros()`; displays duration that the light was on or off (using serial monitor); serial functions (`begin()`, `print()`, `println()`, etc.)
6. **Blinking LED with "Trainable" Delay (P)**
Mechanical switches and problem with debounced buttons.
7. **Debouncing a Switch, I (P)**
Software approach to debouncing (introduction to state machines).
8. **Debouncing a Switch, II (P)**
Software approach to debouncing using the Bounce library.
9. **Debouncing a Switch, III (EP)**
Hardware debouncing using an RC filter; capacitance and capacitors.
10. **Debouncing a Switch, IV (EP)**
Hardware debouncing using electronics; introduction to operational amplifiers; op-amp with positive feedback.
11. **Debouncing a Switch, V (EP)**
Hardware debouncing using electronics; Schmitt trigger.
12. **"Counter" with 6 LED's, I (P)**
"Marching" one-at-a-time illumination system: Variable initialized to 1; multiplied by 2 with each "update." Reset to 1 once last LED reached. Introduction to binary numbers; modulo operator; for-loop; bit-wise operations ; `bit()`; `bitRead()`.
13. **"Counter" with 6 LED's, II (P)**
Binary counter where count is initialized to 1 and then increment by 1 with each "update." Illumination of LED represents the binary count.

14. **Breathing LED using PWM (P)**
Introduction to pulse width modulation (PWM); `analogWrite()`.
15. **Color-Changing LED Using Tricolor LED and PWM (P)**
16. **Controlling LED Brightness with Potentiometer (EP)**
Voltage division, `analogRead()`.
17. **Servo Control, I (P)**
`Servo` library; control with increment and decrement buttons.
18. **Servo Control, II (P)**
Control with potentiometer.
19. **Servo control, III (P)**
Light tracking device with two photo resistor. Servo controls the position of two “parallel” photo resistors, but one photo resistor “sees” ahead and to the left while the other sees ahead and to the right. Their output is compared and the position is adjusted until the outputs are (roughly) the same.
20. **Relay to Control Large Currents (E)**
Light bulb controlled with a relay and a photo resistor that detects ambient light.
21. **Music with Piezo Element (EP)**
Arrays; piezoelectric devices; `tone()`; `notone()`.