Blink

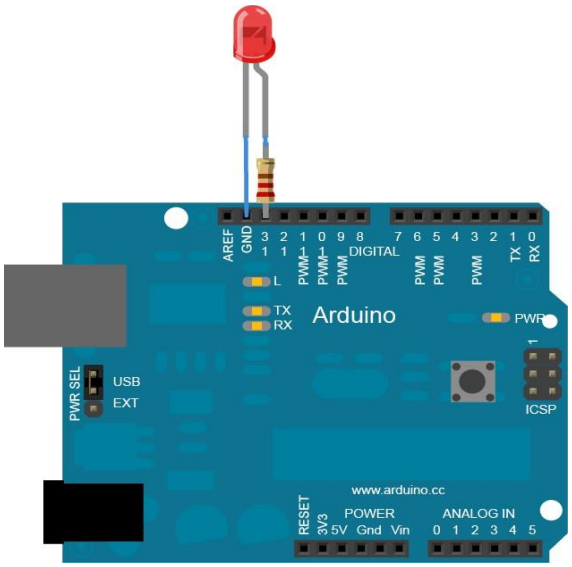
This example shows the simplest thing you can do with an Arduino or Genuino to see physical output: it blinks the on-board LED.

Materials Needed:

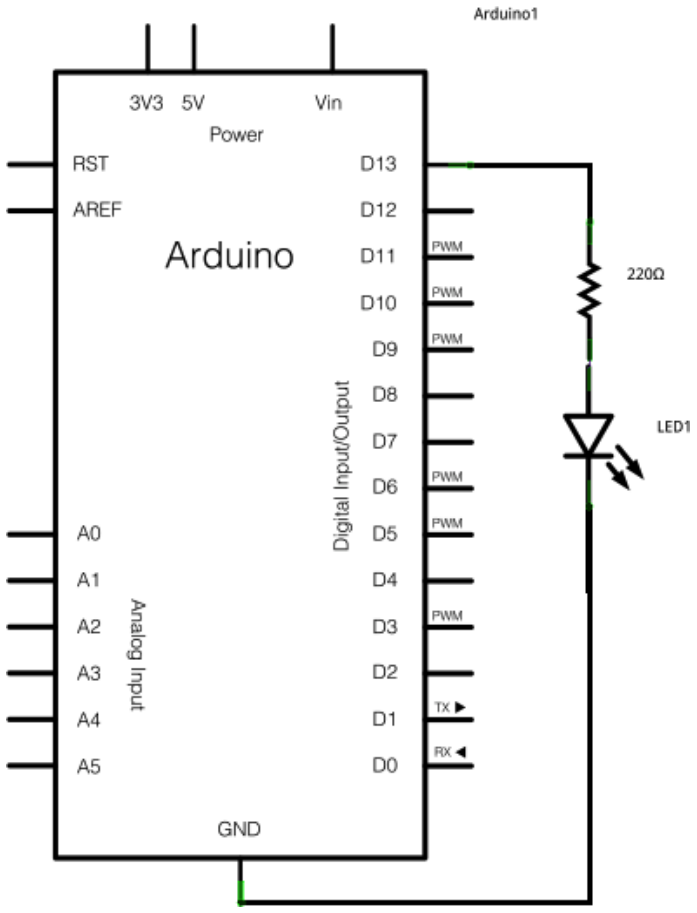
1. Arduino board (e.g., Arduino Uno)
2. LED (Light Emitting Diode)
3. 220-ohm resistor
4. Breadboard (optional, but helpful for prototyping)
5. Jumper wires

Circuit Setup Instructions:

1. **Connect the Arduino to the Breadboard (Optional):**
   - If using a breadboard, connect the Arduino board to the breadboard with jumper wires. This will allow you to build the circuit more easily on the breadboard.
2. **Place the LED:**
   - Insert the LED into the breadboard. The longer leg of the LED is the positive (anode) side, and the shorter leg is the negative (cathode) side.
3. **Connect the Resistor:**
   - Place the 220-ohm resistor into the breadboard. Connect one end of the resistor to the positive (anode) leg of the LED.
4. **Connect the Resistor to the Arduino:**
   - Use a jumper wire to connect the free end of the resistor to the digital pin 13 on the Arduino board. Pin 13 is used here because LED_BUILTIN refers to the built-in LED on this pin.
5. **Connect the LED to Ground:**
   - Use another jumper wire to connect the negative (cathode) leg of the LED to one of the GND (ground) pins on the Arduino board.
6. **Verify Connections:**
   - Ensure that all connections are secure and that the resistor is properly connected between the anode of the LED and the Arduino pin 13.
   - The LED should be connected to the Arduino’s pin 13 through the resistor, and the cathode should be connected to ground.


AREF
GND
DIGITAL
L
TX
RX
Arduino
PWR
USB
EXT
PWR SEL
ICSP
www.arduino.cc
RESET
POWER
ANALOG IN

Schematic


Arduino1
3V3
5V
Vin
Power
RST
AREF
Arduino
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
PWM
Digital Input/Output
A0
A1
A2
A3
A4
A5
Analog Input
GND
TX
RX
220Ω
LED1

Code in Assembly:

```
.ORG 0x0000             ; the next instruction has to be written to
                        ; address 0x0000
rjmp START              ; the reset vector: jump to "main"
START:
ldi r16, low(RAMEND)    ; set up the stack
out SPL, r16
ldi r16, high(RAMEND)
out SPH, r16
ldi r16, 0xFF           ; load register 16 with 0xFF (all bits 1)
out DDRB, r16           ; write the value in r16 (0xFF) to Data
                        ; Direction Register B
LOOP:
  sbi PortB, 5          ; switch off the LED
  rcall delay_05        ; wait for half a second
  cbi PortB, 5          ; switch it on
  rcall delay_05        ; wait for half a secon
  rjmp LOOP             ; jump to loop

DELAY_05:               ; the subroutine:
  ldi r16, 31           ; load r16 with 31
OUTER_LOOP:             ; outer loop label
  ldi r24, low(1021)    ; load registers r24:r25 with 1021, our new
                        ; init value
  ldi r25, high(1021)   ; the loop label
DELAY_LOOP:             ; "add immediate to word": r24:r25 are
                        ; incremented
  adiw r24, 1           ; if no overflow ("branch if not equal"), go
                        ; back to "delay_loop"
  brne DELAY_LOOP
  dec r16               ; decrement r16
  brne OUTER_LOOP       ; and loop if outer loop not finished
  ret                   ; return from subroutine
```

Code in C:

```
*/


// the setup function runs once when you press reset or power the
board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}


// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the
voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the
voltage LOW
  delay(1000);                       // wait for a second
}
```