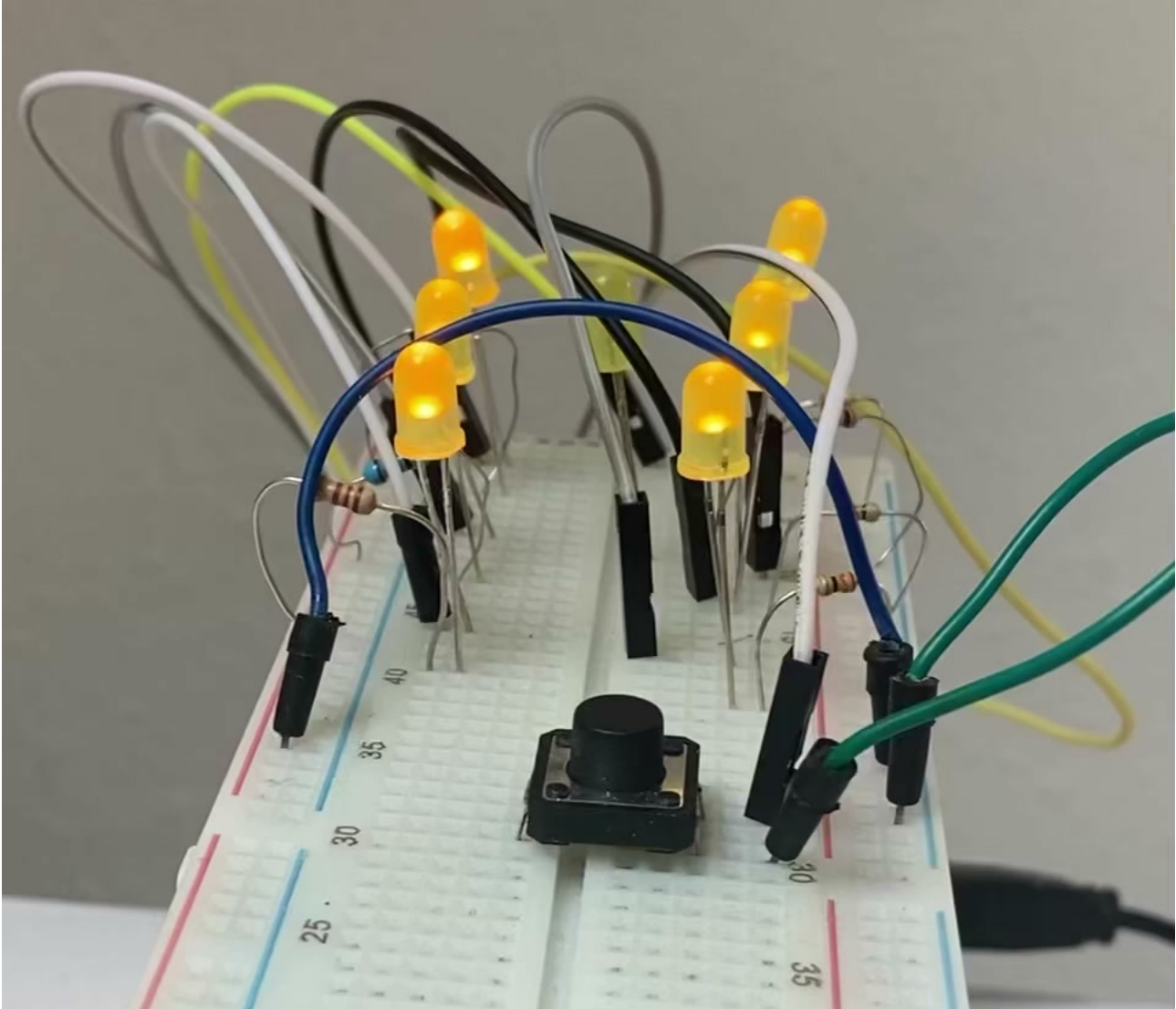


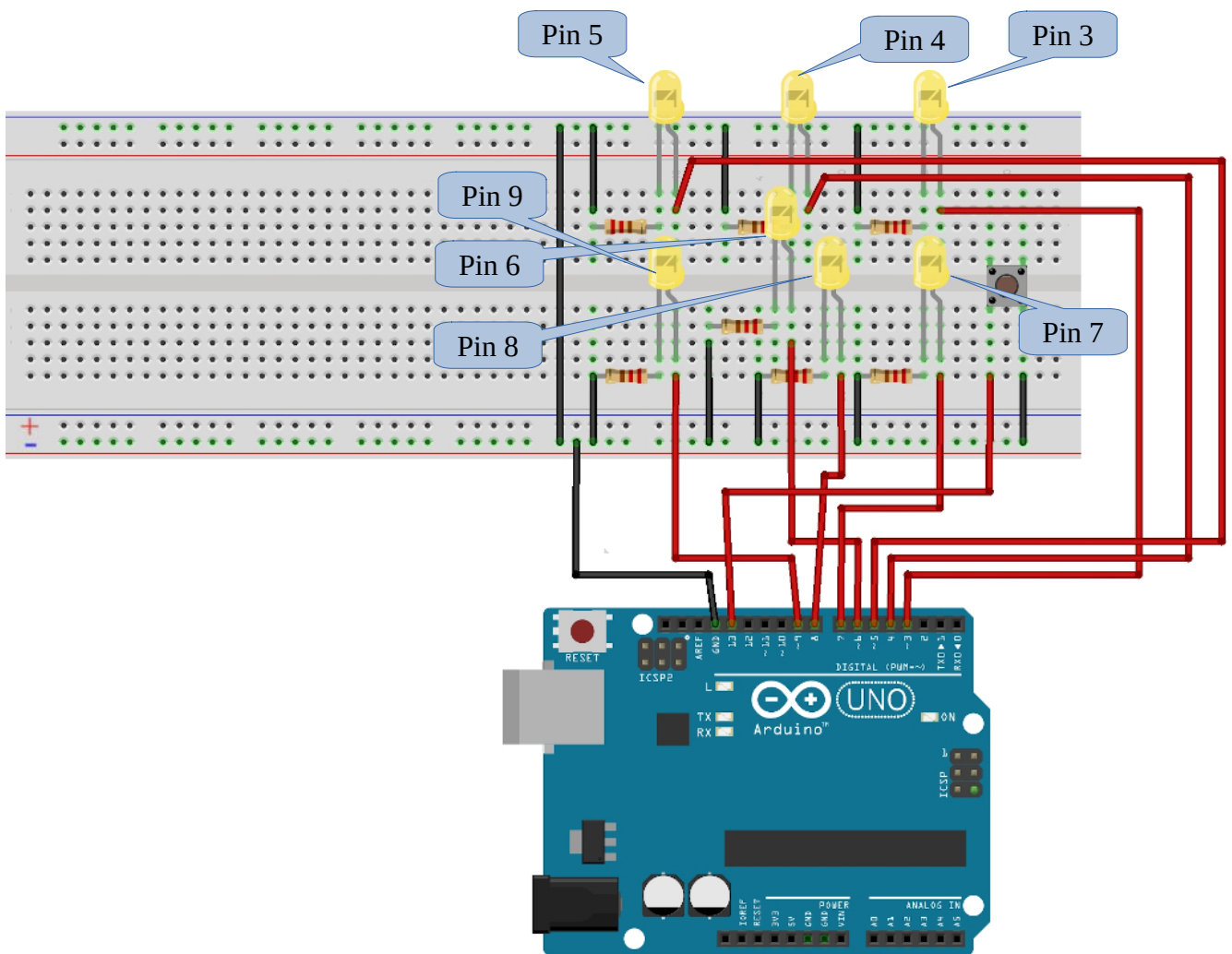
Würfel mit LEDs

Mit LEDs soll ein Würfel realisiert werden.



Benötigte Beiteile:

- ➔ 7 LEDs
- ➔ Widerstände 220 Ω
- ➔ Leitungsdrähte



fritzing

Die LEDs sollen mit dem direkten Ansprechen der Ports geschaltet werden.

| | | | | | | | | |
|--------|----|----|-----|-----|-----|-----|----|----|
| Port D | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Port B | 0 | 0 | D13 | D12 | D11 | D10 | D9 | D8 |
| Port C | 0 | 0 | A5 | A4 | A3 | A2 | A1 | A0 |

Funktionen wie `digitalWrite()` und `digitalRead()` sind relativ langsam in der Ausführung und manchmal werden Programme mit vielen Ein- oder Ausgängen komplex und schwer überschaubar. Deshalb ist es oft übersichtlicher die Pins mithilfe der Ports zu definieren und zu schalten.

Es existieren drei sogenannte Register:

- DDR → Pins als INPUT = 0 oder als OUTPUT = 1 definieren
- PORT → Pins als HIGH = 1 oder als LOW = 0 festlegen
- PIN → gibt den Zustand der Pins an, die im DDR-Register auf Input gesetzt wurden

Das Register wird jeweils mit dem Namen des Ports (D, B oder C) ergänzt.

Beim Zugriff auf die Ports wird jeweils ein Bit gesetzt. Sein Wert ist entweder 1 = an, oder 0 = aus.

Beispiele:

Digitale Pins 7, 6, 5, 4 und 3 als OUTPUT setzen

```
DDRD = B11111000;
```

Digitale Pins 7, 6, 5, 4 und 3 auf HIGH setzen:

```
PORTD = B11111000;
```

Taster an Pin 13 als INPUT setzen

```
DDRB = B00100000;
```

PULL_UP-Widerstand an Pin 13 einschalten

```
PORTB = B00100000;
```

Definiere die Variablen.

```
boolean TASTER;  
  
// Würfelaugen 1 bis 6  
int Minimum = 1;  
int Maximum = 7;
```

Der setup-Teil. Beachte die Kommentare.

```
void setup()  
{  
  // Pins 7 bis 3 als OUTPUT definieren  
  DDRD = B11111000;  
  
  // Pins 8 bis 9 als OUTPUT definieren  
  DDRB = B00000011;  
  
  // Pin 13 PULLUP-Widerstand einschalten  
  PORTB = B00000100;  
  
  // Zufallsgenerator starten  
  randomSeed(analogRead(0));  
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // Test ob alle LEDs leuchten
  // PORTD = B11111000;
  // PORTB = B00000011;

  /*
   Port B lesen
   Bit 13 wird ausgeblendet (PINB & B00100000),
   damit der Wert gelesen werden kann
  */
  TASTER = PINB & B00100000;
;
  // Taster gedrückt, Pin 13 hat den Wert 0 (false)
  if (!TASTER)
  {
    delay(200);
    int Zahl = random(Minimum, Maximum);

    switch (Zahl)
    {
      case 1:
        AlleAus();
        PORTD = B01000000;
        break;

      case 2:
        AlleAus();
        PORTD = B10100000;
        break;

      case 3:
        AlleAus();
        PORTD = B11100000;
        break;

      case 4:
        AlleAus();
        PORTD = B10101000;
        PORTB = B00000010;
        break;

      case 5:
        AlleAus();
        PORTD = B11101000;
        PORTB = B00000010;
        break;
    }
  }
}
```

```
    case 6:
        AlleAus();
        PORTD = B101111000;
        PORTB = B00000011;
        break;

    default:
        break;
}
}
```

Die Methode AlleAus() schaltet alle LEDs wieder aus.

```
void AlleAus()
{
    PORTD = 0;
    PORTB = 0;
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 20.12.22