

Ардуино

# Двоичное представление

- Число в двоичном виде записывается следующим образом:  
`0b00000000`
- Первые два символа **0b** дают понять, что речь идёт о двоичном счёте. Все нули означают, что все светодиоды будут выключены.

# Числа до 10

- `byte numberSegments[10] = { 0b01110111,`
- `0b00010100,`
- `0b10110011,`
- `0b10110110,`
- `0b11010100,`
- `0b11100110,`
- `0b11100111,`
- `0b00110100,`
- `0b11110111,`
- `0b11110110, };`

# millis()

- Функция без параметров возвращает число миллисекунд (unsigned long), прошедших с запуска Arduino.

# bitRead

- bitRead(x, n)

Функция считывает биты с указанного числа. Возвращает значение бита (0 или 1)

**x** - число, которое нас интересует

**n** - какой бит нужно считать

# millis()

- `unsigned long time; void setup(){  
Serial.begin(9600); } void loop(){  
Serial.print("Time: "); time = millis(); //prints  
time since program started  
Serial.println(time); // wait a second so as not  
to send massive amounts of data delay(1000);  
}`

# random()

- `random(max)` числа от 0 до max
- `random(min, max)` числа от min до max

- #define FIRST\_SEGMENT\_PIN 2
- #define SEGMENT\_COUNT 8
- // Всего цифр 10, поэтому в массиве 10 чисел.
- byte numberSegments[10] = {
- 0b01110111,
- 0b00010100,
- 0b10110011,
- 0b10110110,
- 0b11010100,
- 0b11100110,
- 0b11100111,
- 0b00110100,
- 0b11110111,
- 0b11110110,
- };
- void setup()
- {
- for (int i = 0; i < SEGMENT\_COUNT; ++i)
- pinMode(i + FIRST\_SEGMENT\_PIN, OUTPUT);
- }

- `void loop()`
- `{`
- `// определяем число, которое следует отобразить.`
- `int number = random(10) // получаем код из массива, в котором содержится полученная цифра`
- `int mask = numberSegments[number];`
- `// для каждого из 7 сегментов индикатора`
- `for (int i = 0; i < SEGMENT_COUNT; ++i) {`
- `// определяем: должен ли он быть включён.`
- `boolean enableSegment = bitRead(mask, i);`
- `// включаем/выключаем сегмент на основе полученного значения`
- `digitalWrite(i + FIRST_SEGMENT_PIN, enableSegment);`
- `}`
- `}`