Embedded System Development Fundamentals

Lecture 2

Embedded Systems

Reference: Chapters 5, Arduino Cookbook, Michael Margolis, 1st Edition, 2011, O'Reilly books

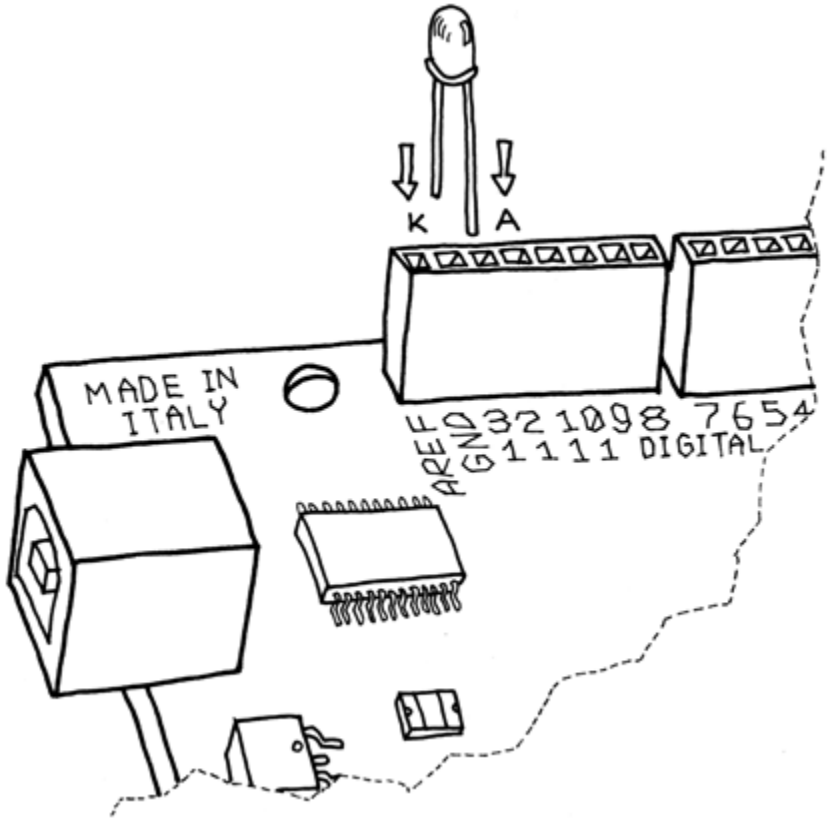
Agenda

- Digital Input / Output
- Using Interrupts
- Using Timers
- Reading Analog Signal
- Producing Analog Signal

Digital Input / Output

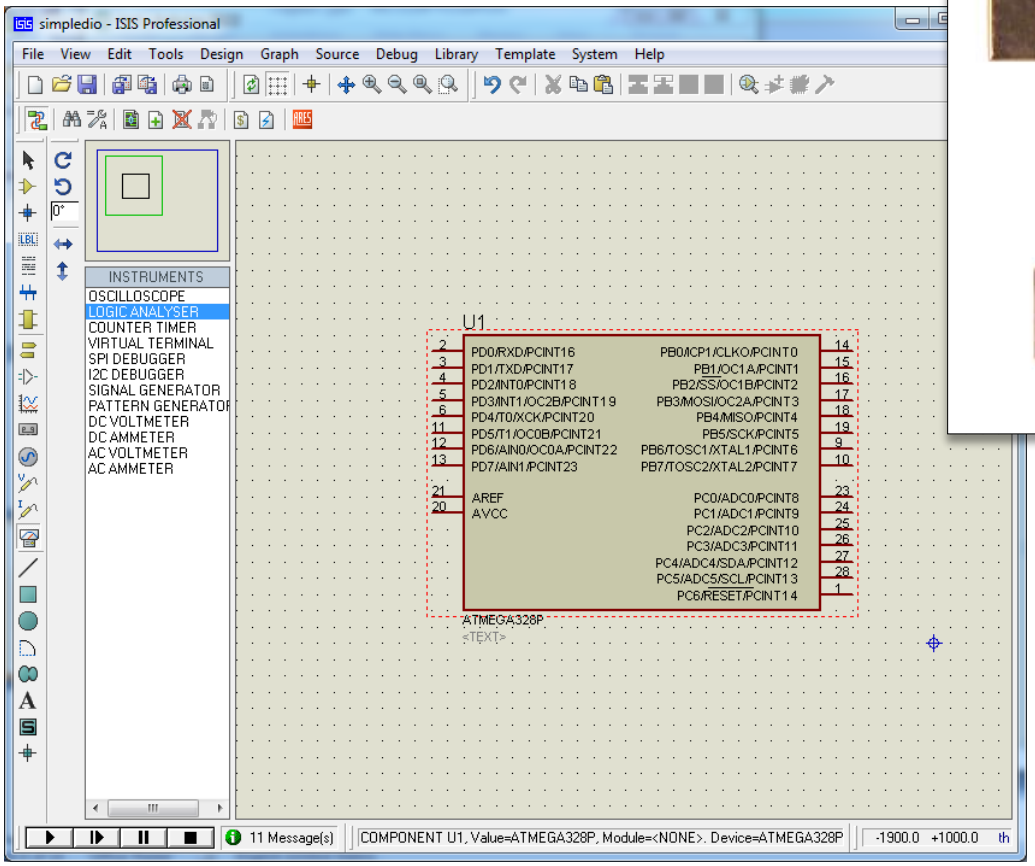
Simple Digital Output

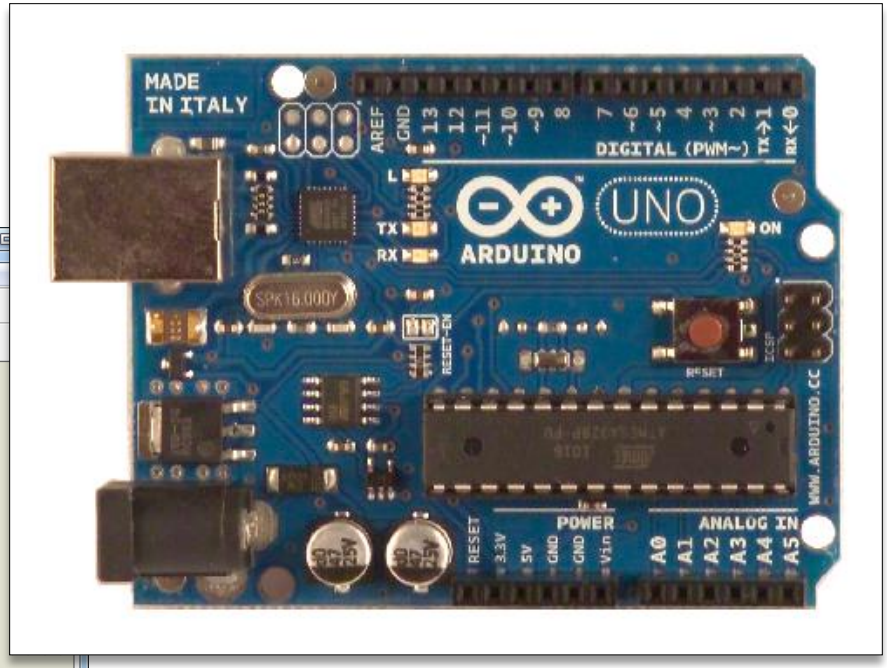
```
#define LED 13

void setup()
{
     pinMode(LED, OUTPUT);
}

void loop()
{
     digitalWrite(LED, HIGH);
}
```



Using Simulator Software ISIS/Proteus

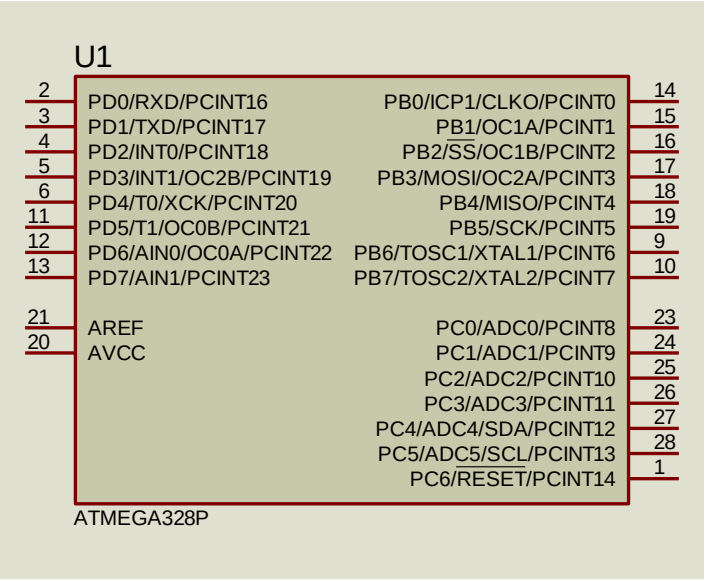
Simulator with ATMega328P Chip Model



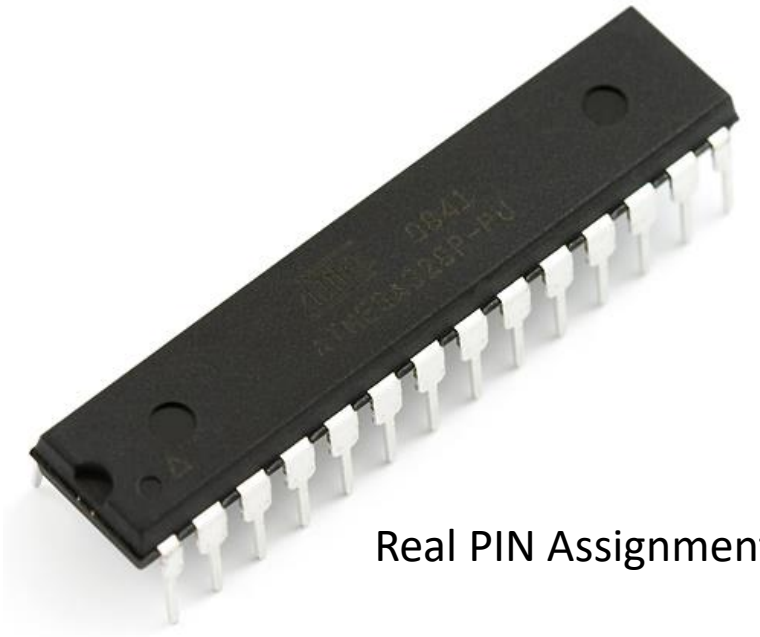


Real Arduino UNO Board with Real ATMega328P Chip

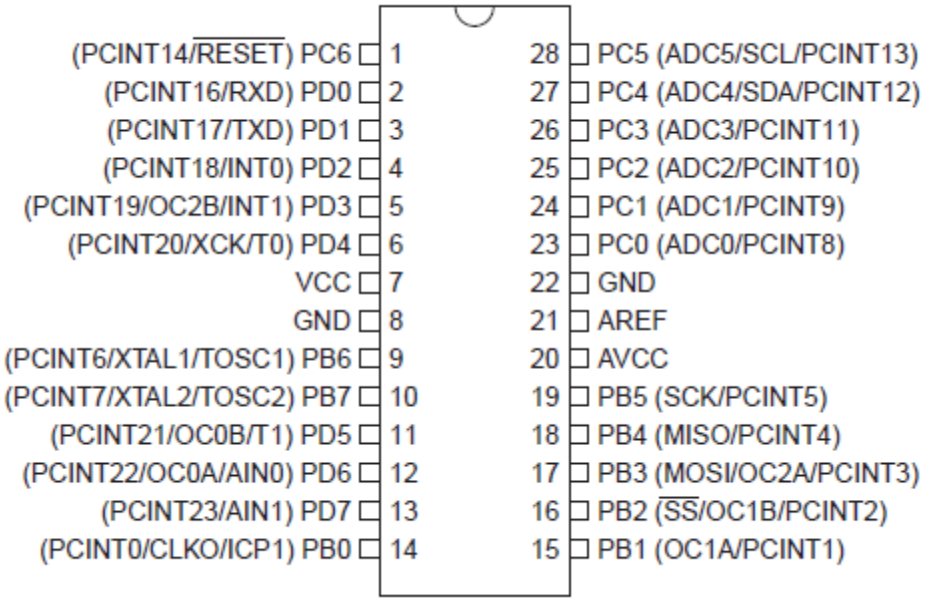
ATMega328P Microcontroller



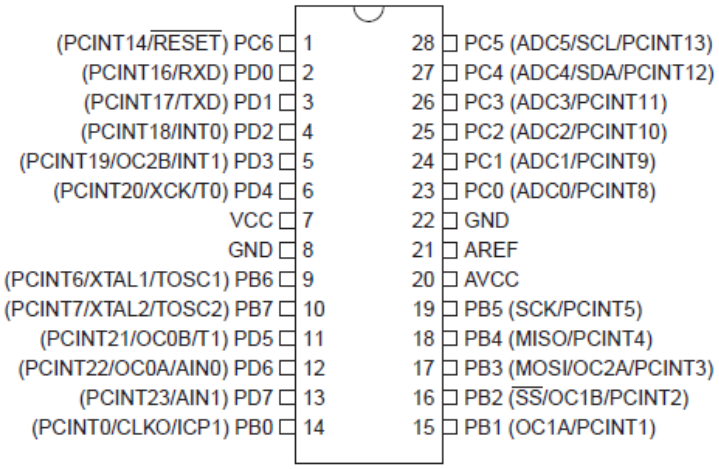
Model PIN Assignment

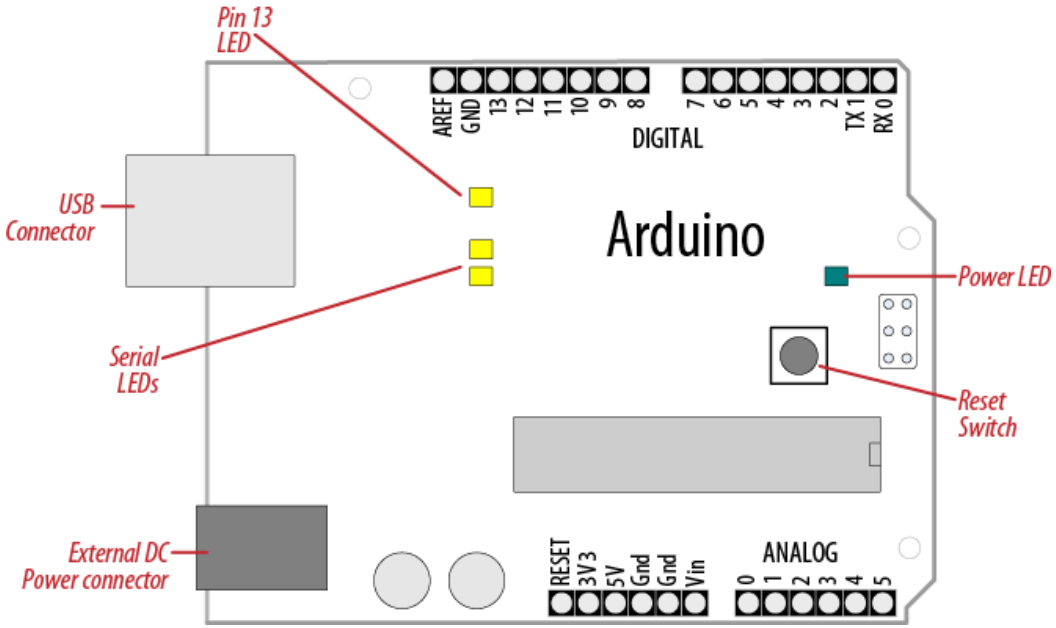


Real PIN Assignment



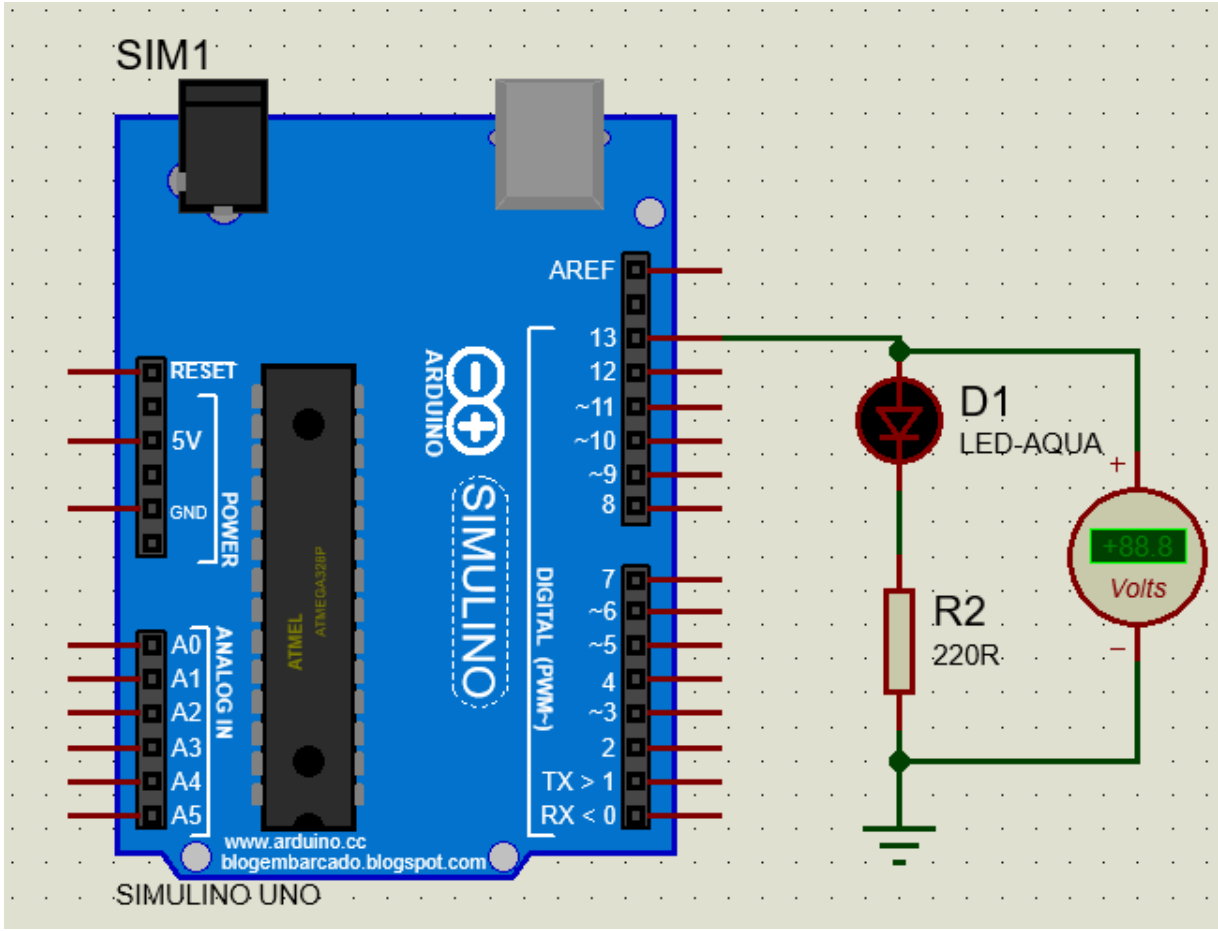
ATMega328P vs. Arduino UNO PINs





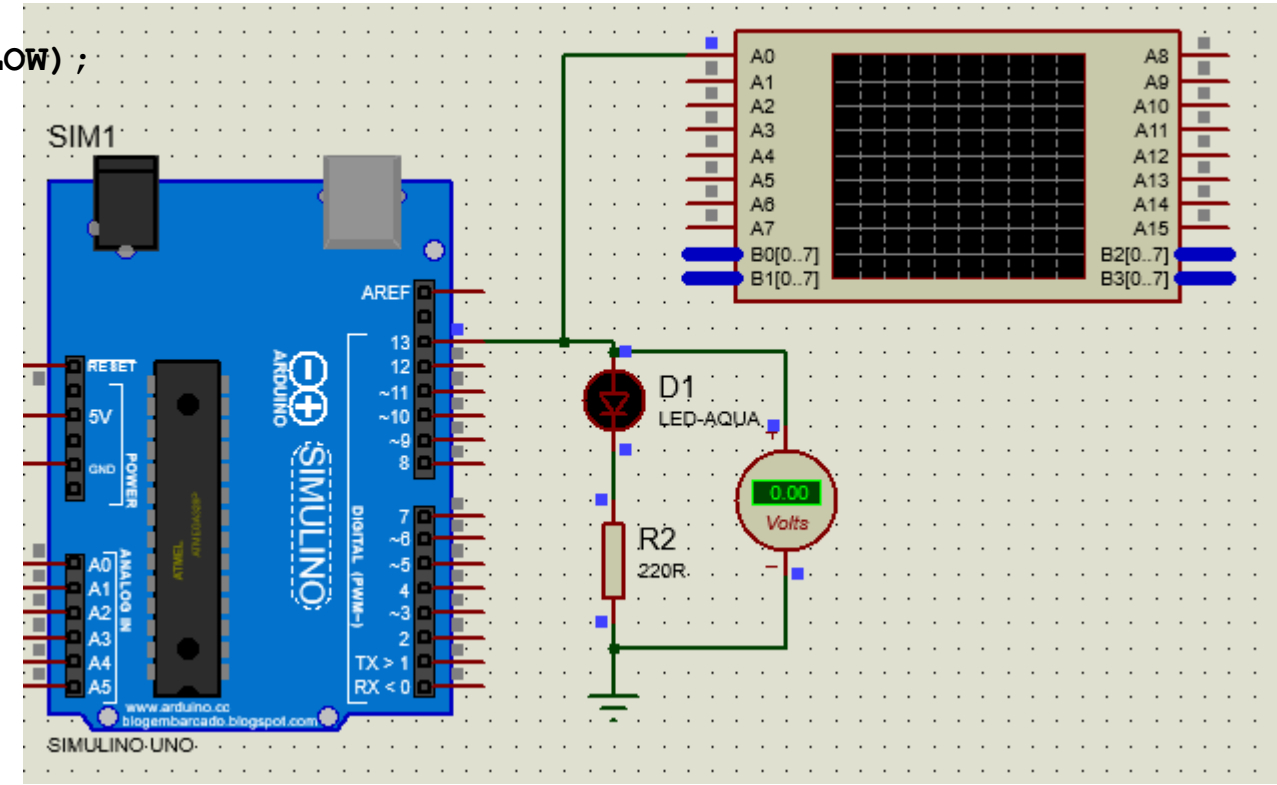
Simple Digital Output

```
#define LED 13

void setup()
{
      pinMode(LED, OUTPUT);
}

void loop()
{
      digitalWrite(LED, HIGH);
}
```

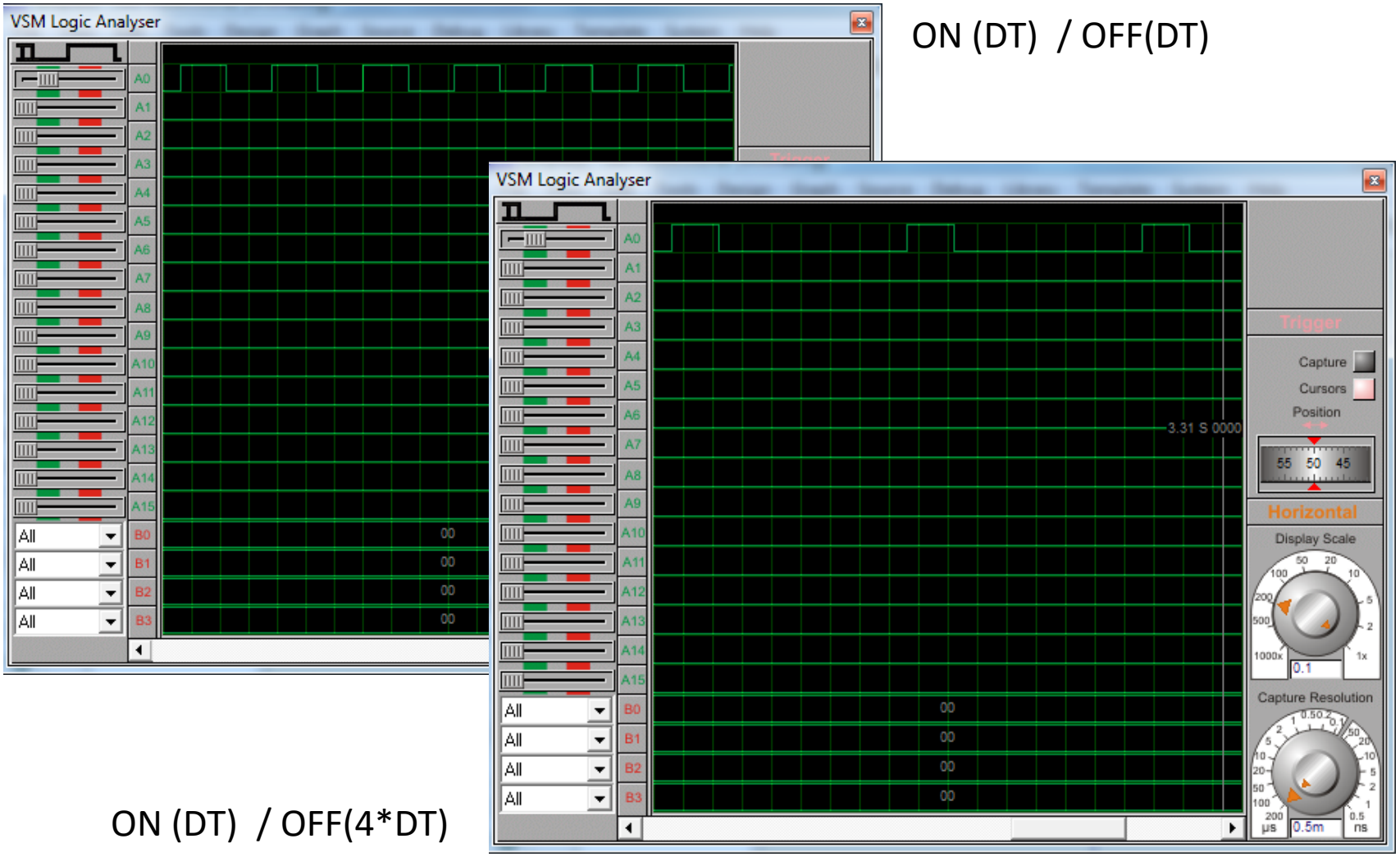


Square Signal

```
#define LED 13
#define DT  10

void setup()
{
      pinMode(LED, OUTPUT);
}

void loop()
{
      digitalWrite(LED, HIGH);
      delay(DT);
      digitalWrite(LED, LOW);
      delay(DT);
}
```



Square Signal

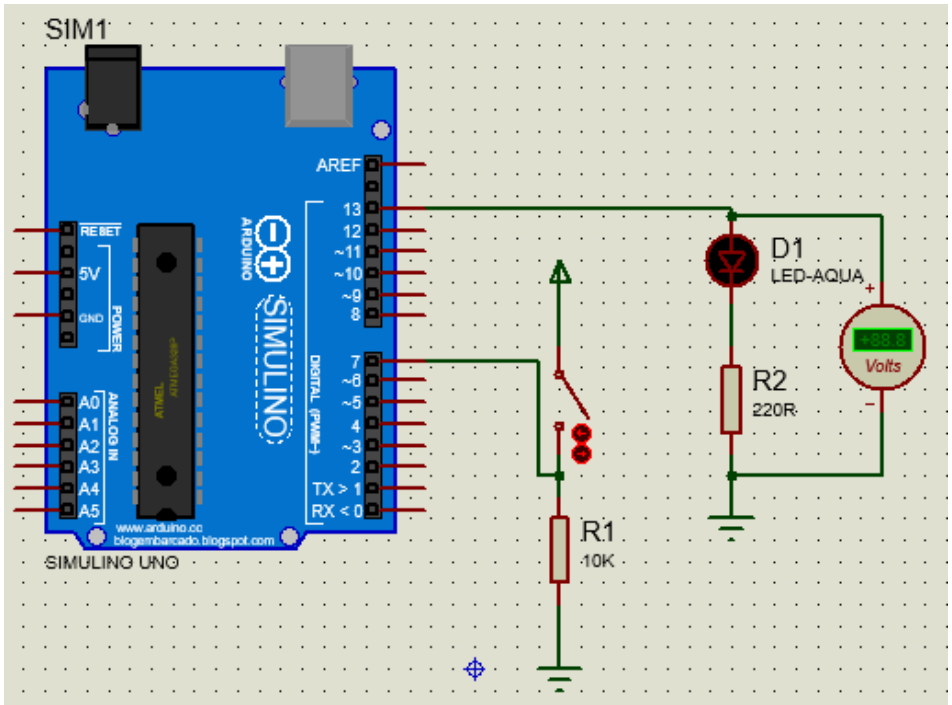


ON (DT) / OFF(DT)

ON (DT) / OFF(4*DT)

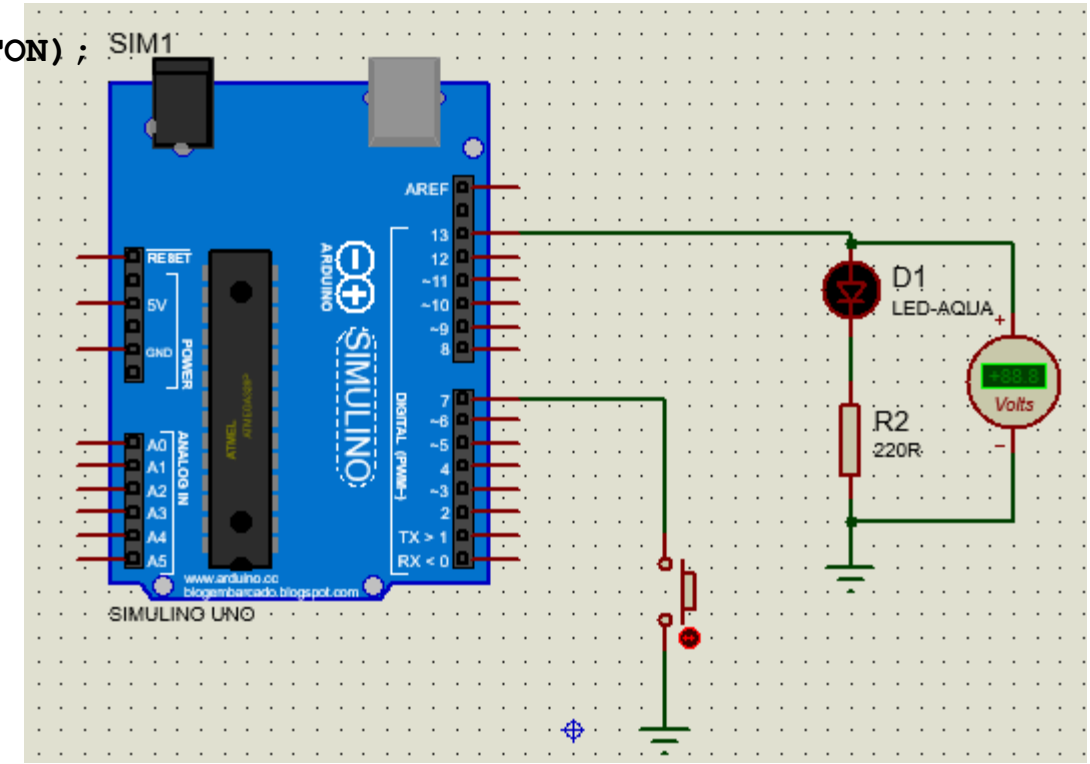
Simple Digital Input / Output

```
#define LED 13
#define BUTTON 7

void setup()
{
      pinMode(LED, OUTPUT);
      pinMode(BUTTON, INPUT);
}

void loop()
{
      int value = digitalRead(BUTTON);
      digitalWrite(LED, value);
}
```



Simple Digital Input / Output with Input Pull-up Resistance

```
#define LED 13
#define BUTTON 7

void setup()
{
     pinMode(LED, OUTPUT);
     pinMode(BUTTON, INPUT);
     digitalWrite(BUTTON, HIGH);
//Enable pull-up resistor at input
}

void loop()
{
     int value = digitalRead(BUTTON);
     digitalWrite(LED, value);
}
```

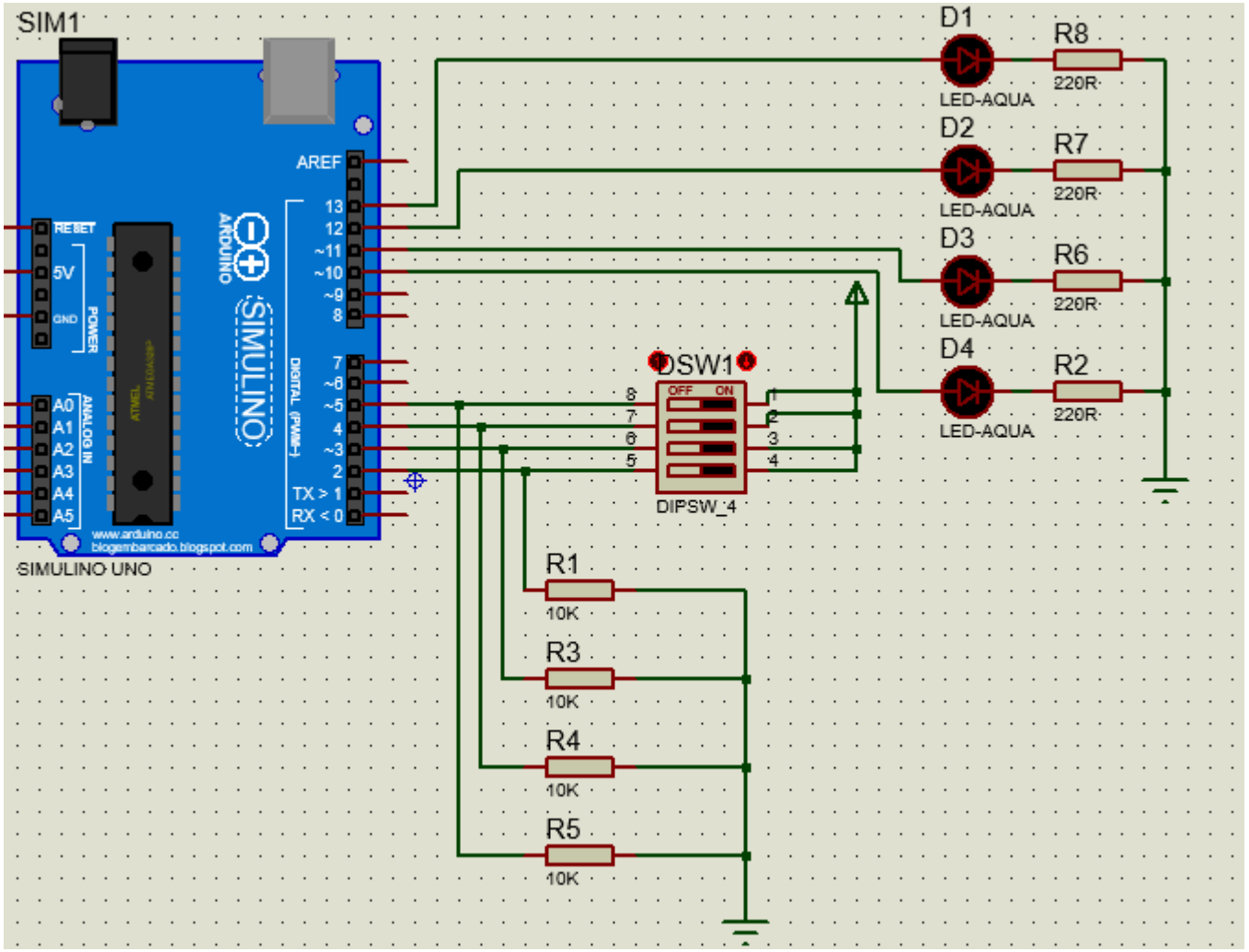


Multiple Digital I/O

Multiple Digital I/O

```
int inputPins[] = {2,3,4,5};
int ledPins[] = {10,11,12,13};

void setup()
{
      for(int index = 0; index < 4; index++)
      {
            pinMode(ledPins[index], OUTPUT);
            pinMode(inputPins[index], INPUT);
      }
}

void loop()
{
      for(int index = 0; index < 4; index++)
      {
            int val = digitalRead(inputPins[index]);
            digitalWrite(ledPins[index], val);
      }
}
```

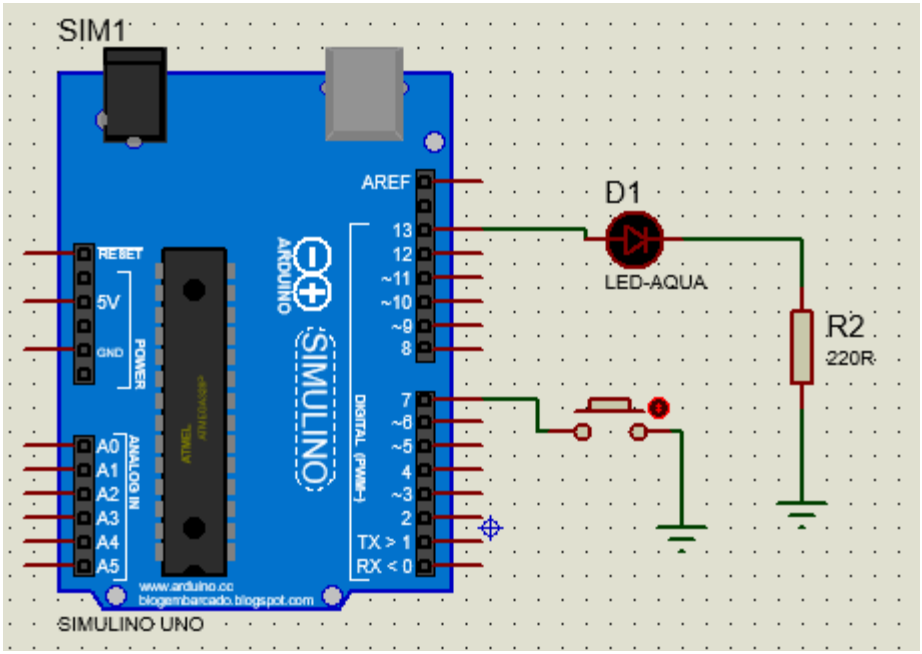
Multiple Digital I/O using Input Pull-up Resistance



Using Interrupts

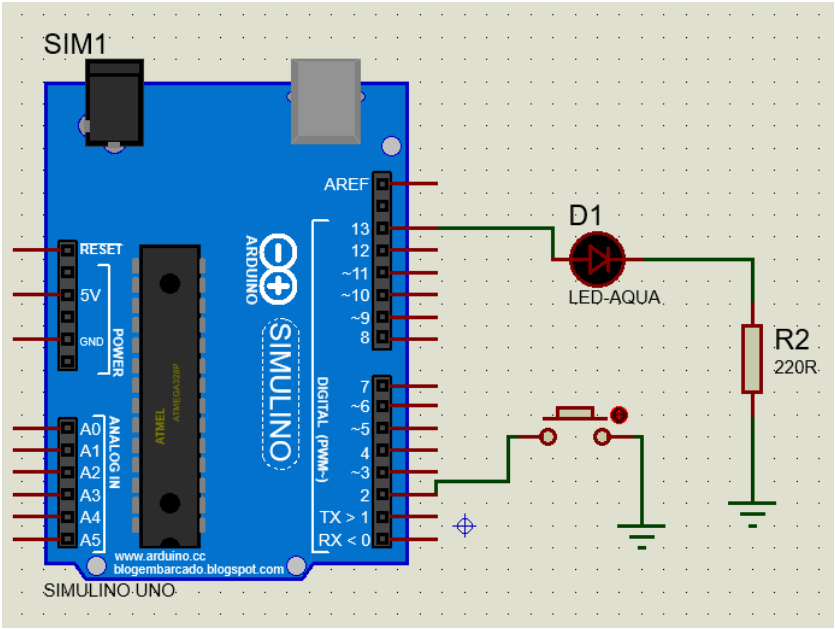
Manual Detection of Push-Button Action

```
#define LED 13
#define BUTTON 7

void setup()
{
      pinMode(LED, OUTPUT);
      pinMode(BUTTON, INPUT);
      digitalWrite(BUTTON, HIGH);
}

int oldValue = HIGH;
int state = LOW;
void loop()
{
      int value = digitalRead(BUTTON);
      if(value==LOW && oldValue==HIGH)
            state = (state==LOW)?HIGH:LOW;
      digitalWrite(LED, state);
      oldValue = value;
}
```



Detection of Push-Button Action using Interrupts

```
#define LED 13
#define BUTTON 2

int state = LOW;
void changeState()
{
  state = (state==LOW)?HIGH:LOW;
  digitalWrite(LED, state);
}

void setup()
{
  interrupts();
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
  digitalWrite(BUTTON, HIGH);
  attachInterrupt(0, changeState, FALLING);
}

void loop()
{
}
```



Interrupts

- You can detect following changes for any digital input:
  - CHANGE
  - FALLING / LOW
  - RISING
- When Interrupt happens any operation is suspended until the Interrupt routine is executed.
- Microcontroller called a Real-Time system because of Interrupts
- If Interrupts is disabled or inaccessible for developers the system is not Real-Time (Example: any personal computer).
- For Arduino UNO there are 2 interrupts (0 with PIN 2, 1 with PIN 3).
- For Arduino Mega there are 2 interrupts (0 with PIN 2, 1 with PIN 3, 2 with PIN 21, 3 with PIN 20, 4 with PIN 19, 5 with PIN 18).

Disable / Enable Interrupts

```
void setup()
{
      //...
}

void loop()
{
      //..
      noInterrupts();
      //Critical Code
      interrupts();
      //...
}
```