

Embedded System Development Fundamentals

Lecture 2 Embedded Systems

Agenda

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

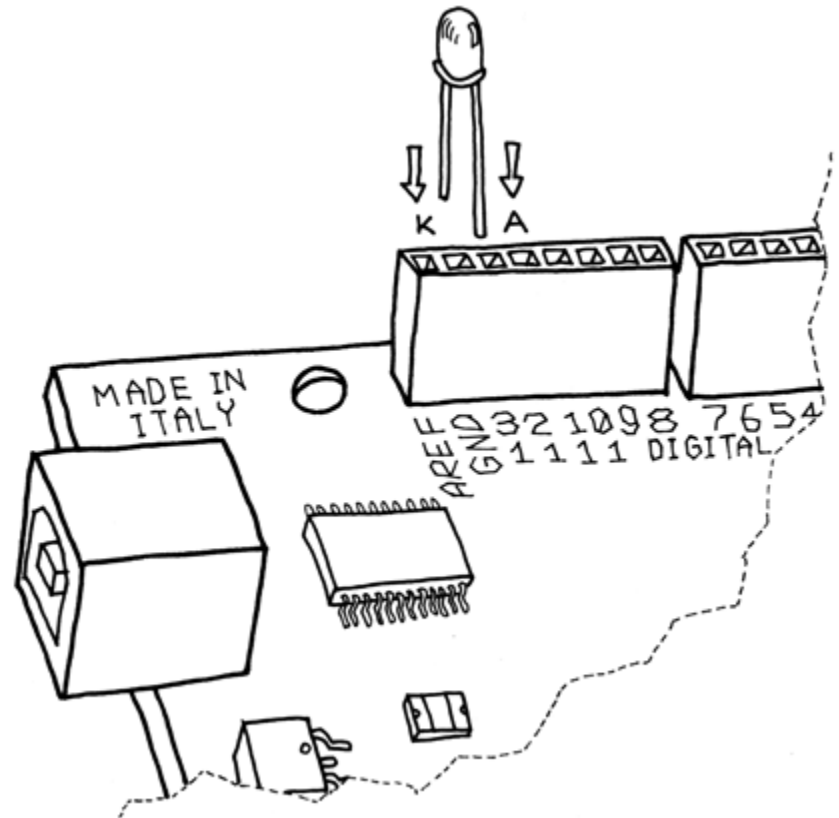
Digital Input / Output

Simple Digital Output

```
#define LED 13

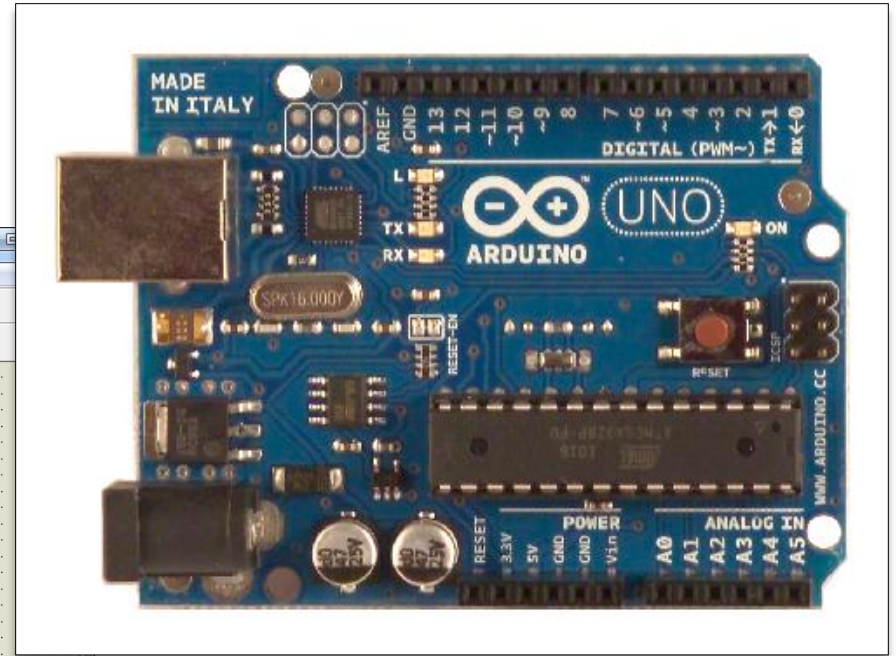
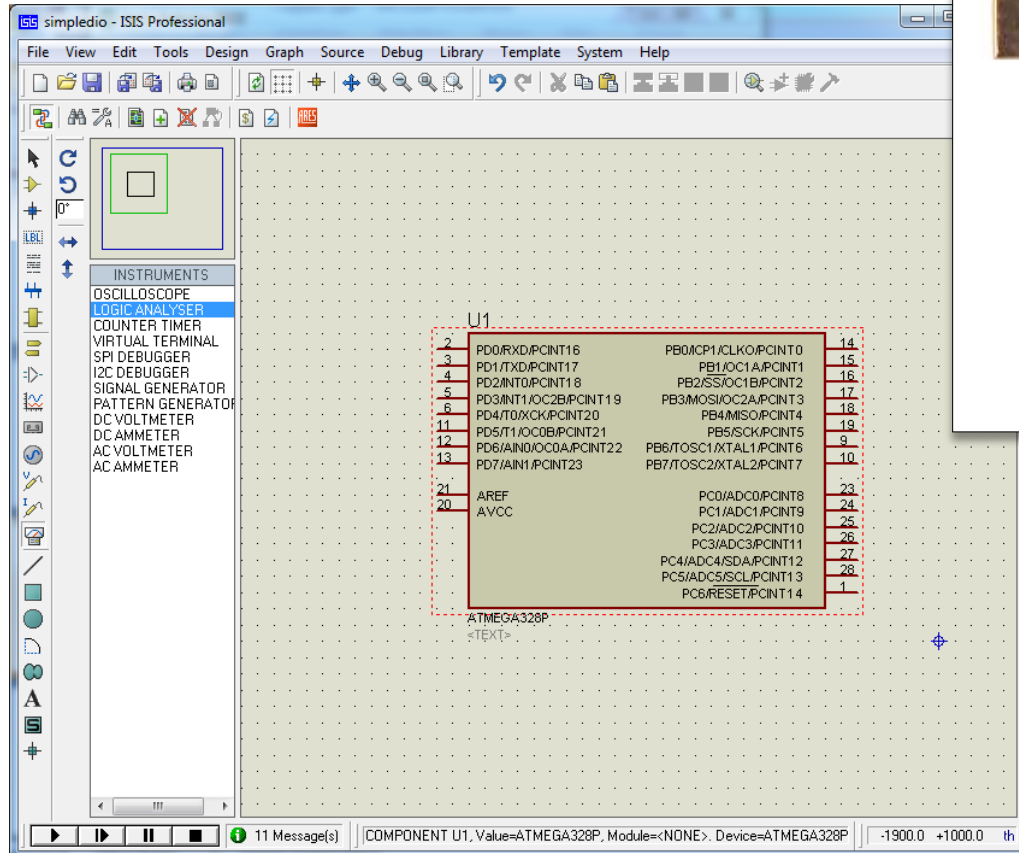
void setup()
{
    pinMode(LED, OUTPUT);
}

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



Using Simulator Software ISIS/Proteus

Simulator with ATmega328P Chip Model



Real Arduino UNO Board with
Real ATmega328P Chip

ATMega328P

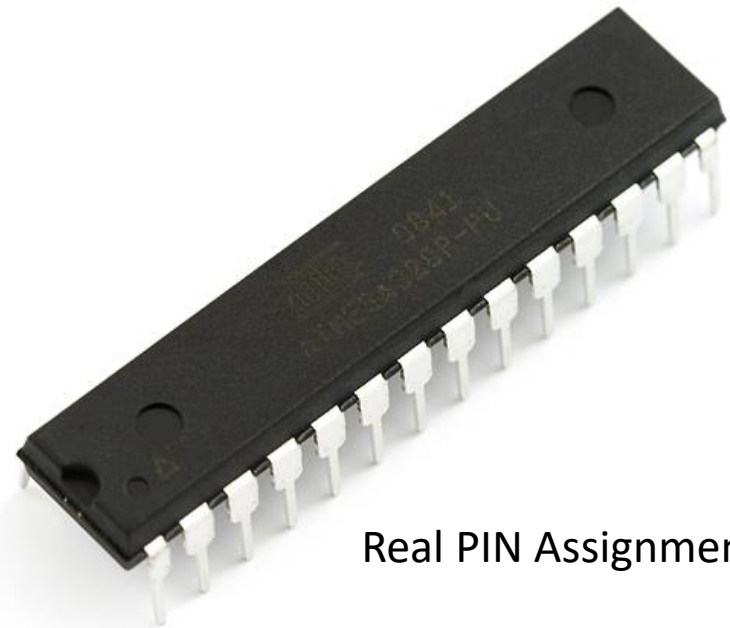
Microcontroller

U1

2	PD0/RXD/PCINT16	PB0/ICP1/CLKO/PCINT0	14
3	PD1/TXD/PCINT17	PB1/OC1A/PCINT1	15
4	PD2/INT0/PCINT18	PB2/SS/OC1B/PCINT2	16
5	PD3/INT1/OC2B/PCINT19	PB3/MOSI/OC2A/PCINT3	17
6	PD4/T0/XCK/PCINT20	PB4/MISO/PCINT4	18
11	PD5/T1/OC0B/PCINT21	PB5/SCK/PCINT5	19
12	PD6/AIN0/OC0A/PCINT22	PB6/TOSC1/XTAL1/PCINT6	9
13	PD7/AIN1/PCINT23	PB7/TOSC2/XTAL2/PCINT7	10
21	AREF	PC0/ADC0/PCINT8	23
20	AVCC	PC1/ADC1/PCINT9	24
		PC2/ADC2/PCINT10	25
		PC3/ADC3/PCINT11	26
		PC4/ADC4/SDA/PCINT12	27
		PC5/ADC5/SCL/PCINT13	28
		PC6/RESET/PCINT14	1

ATMEGA328P

Model PIN Assignment

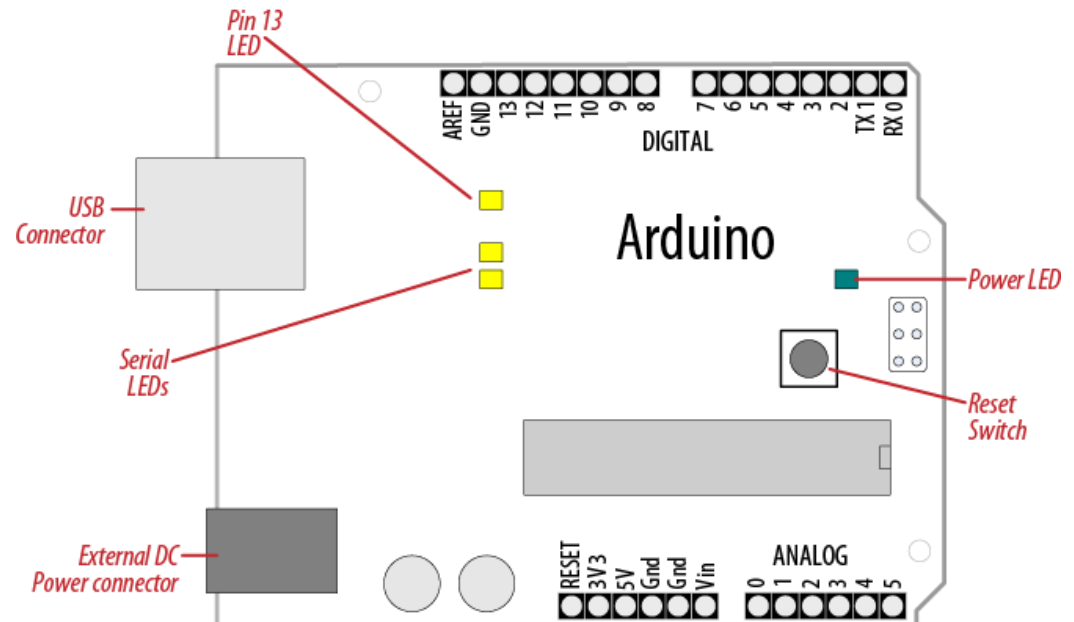


Real PIN Assignment

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

ATMega328P vs. Arduino UNO PINs

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

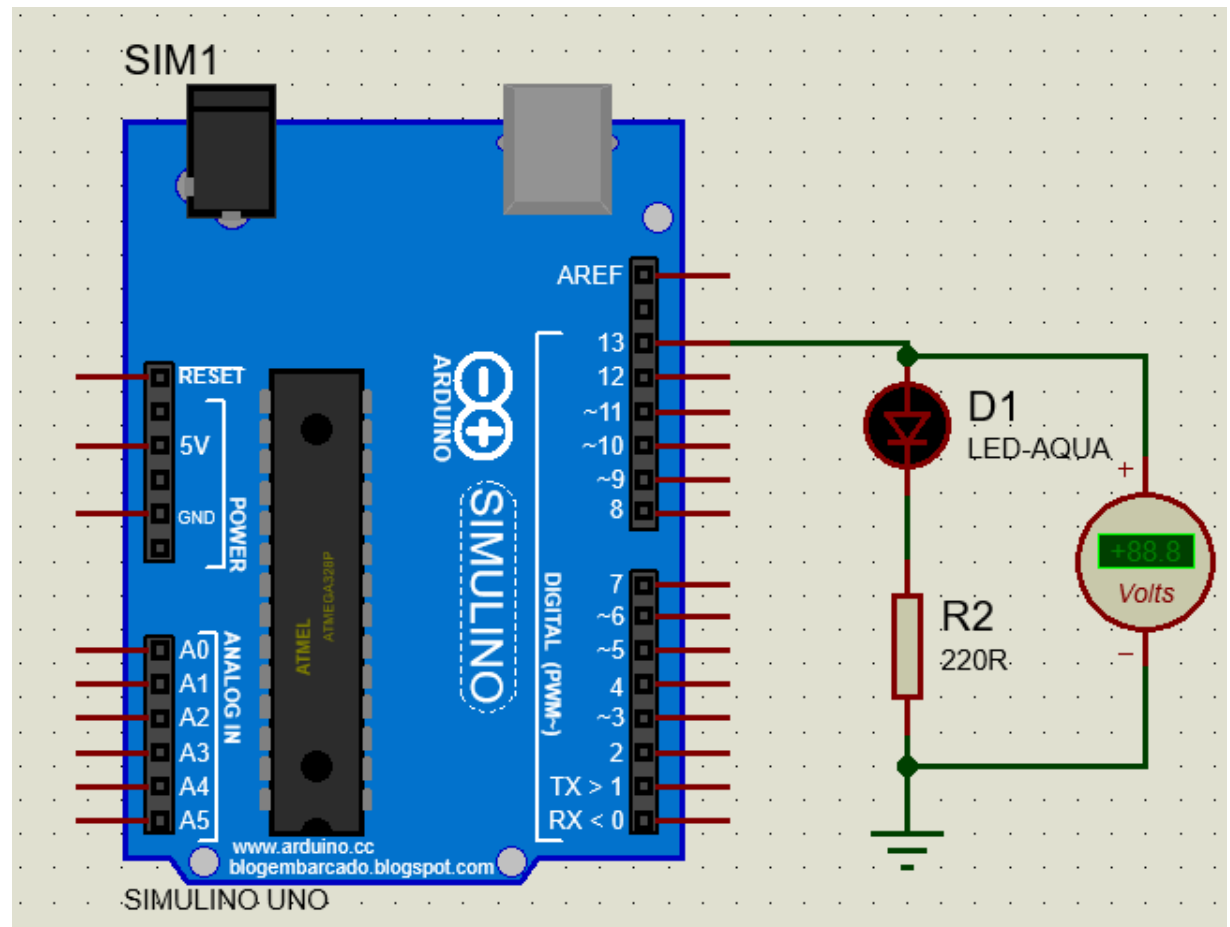


Simple Digital Output

```
#define LED 13

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

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

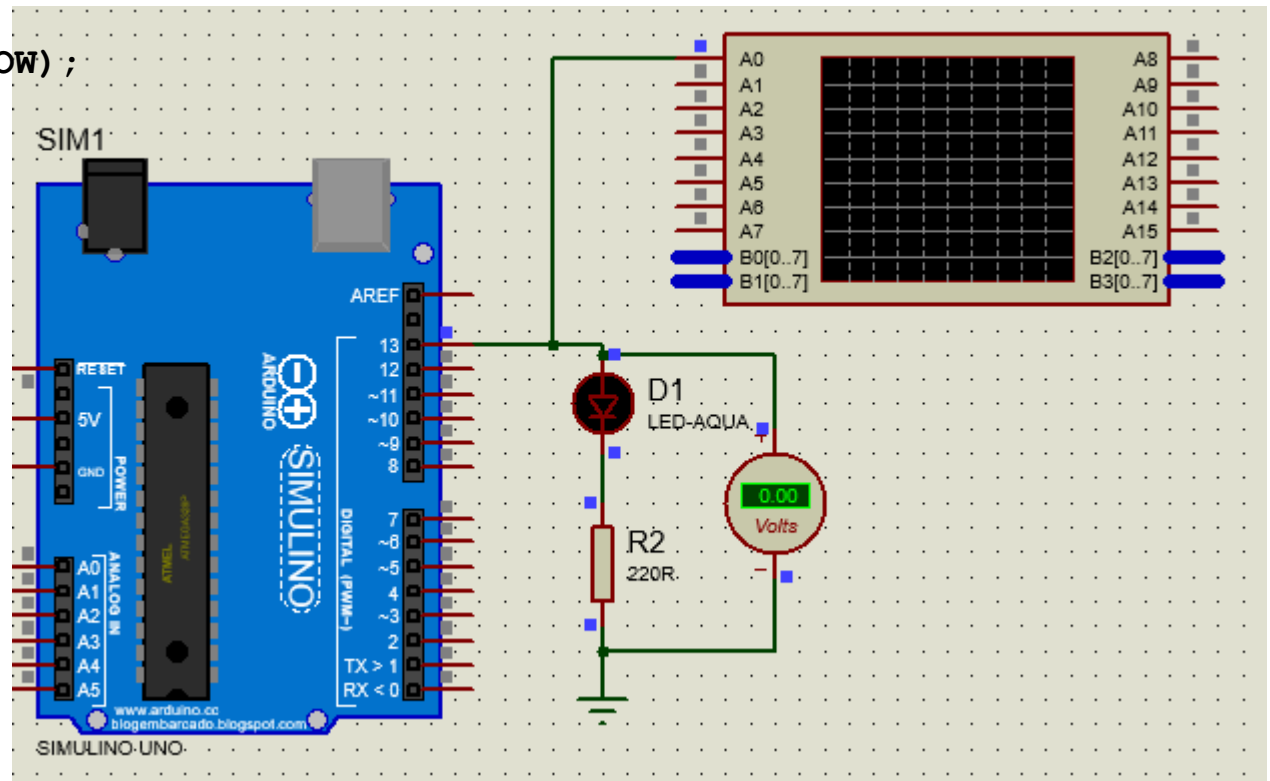



```
#define LED 13
#define DT 10
```

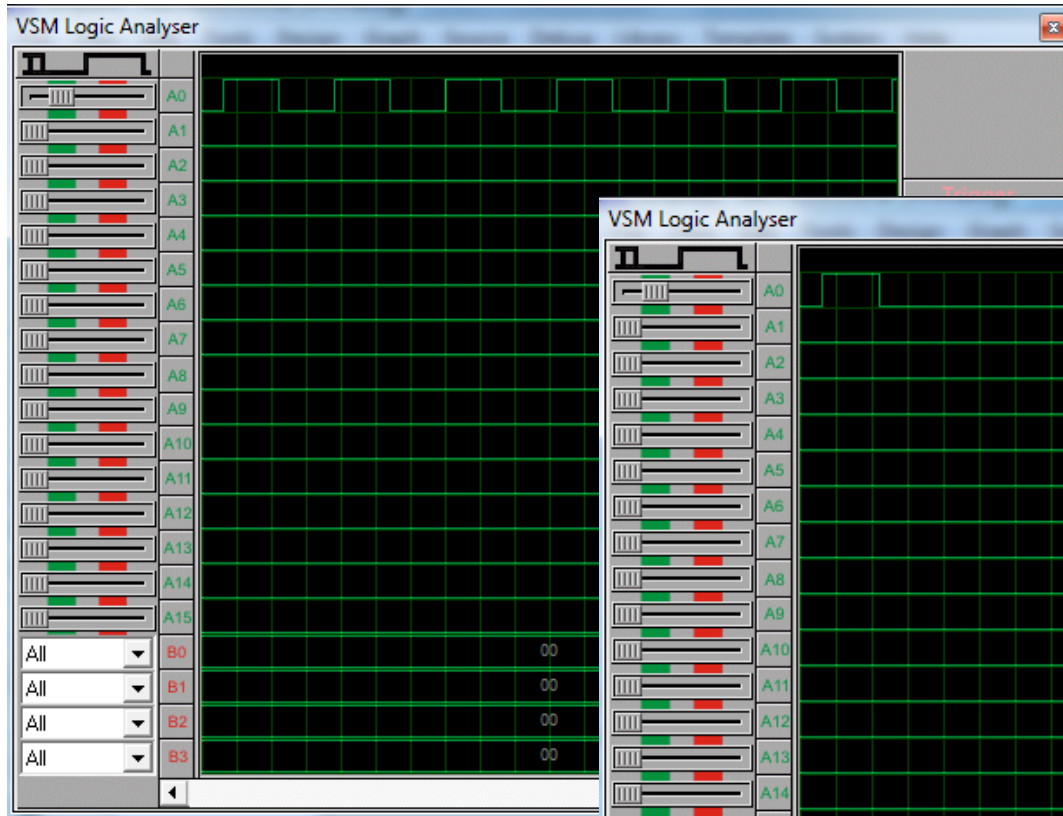
```
void setup()
{
    pinMode(LED, OUTPUT);
}
```

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

Square Signal

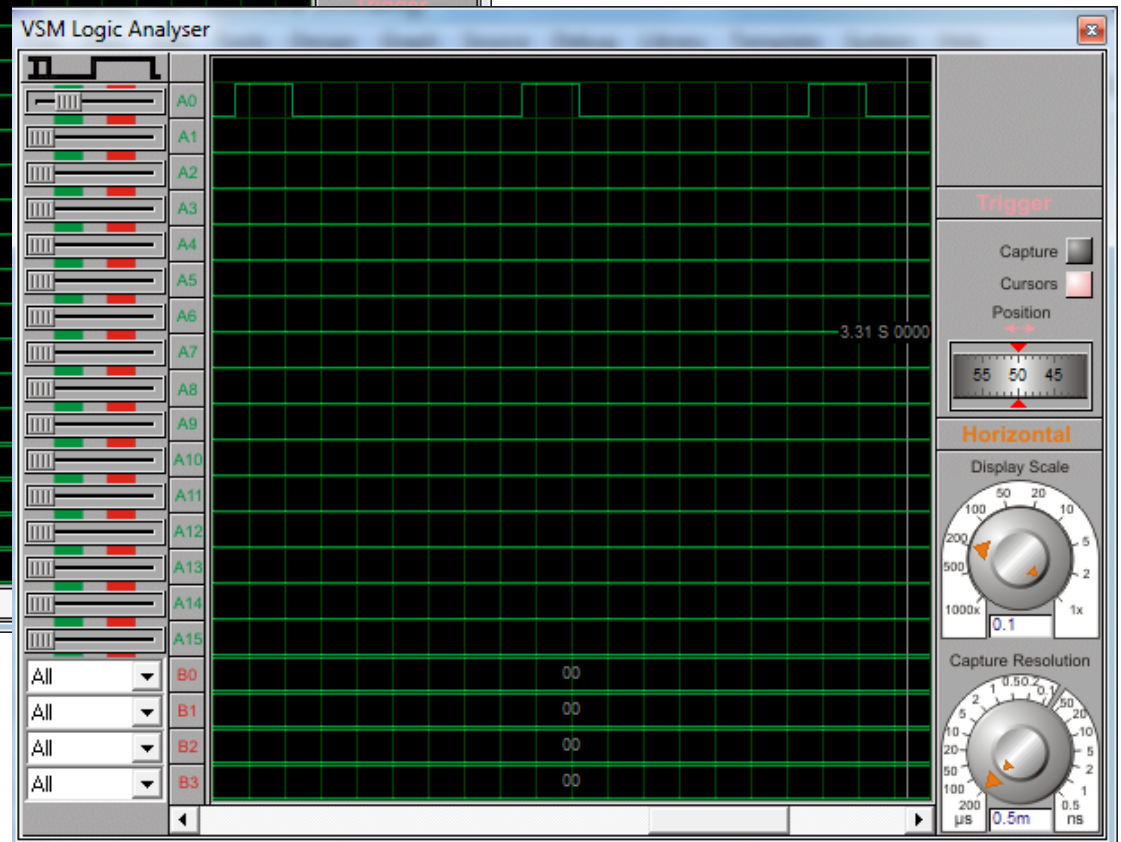


Square Signal



ON (DT) / OFF(DT)

ON (DT) / OFF(4*DT)

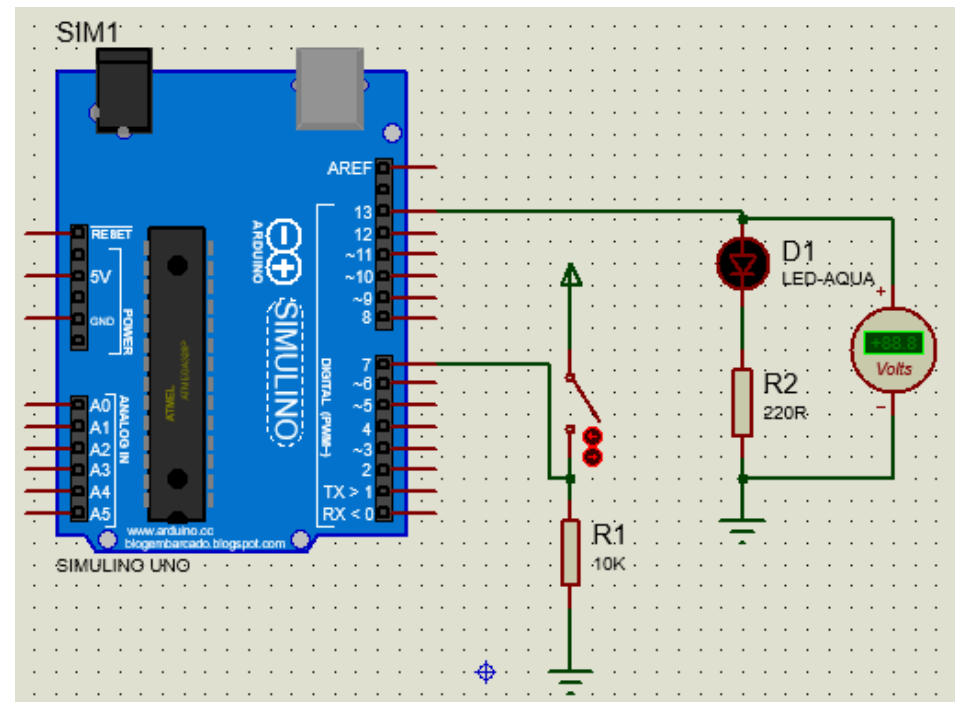


Simple Digital Input / Output

```
#define LED 13
#define BUTTON 7

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

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

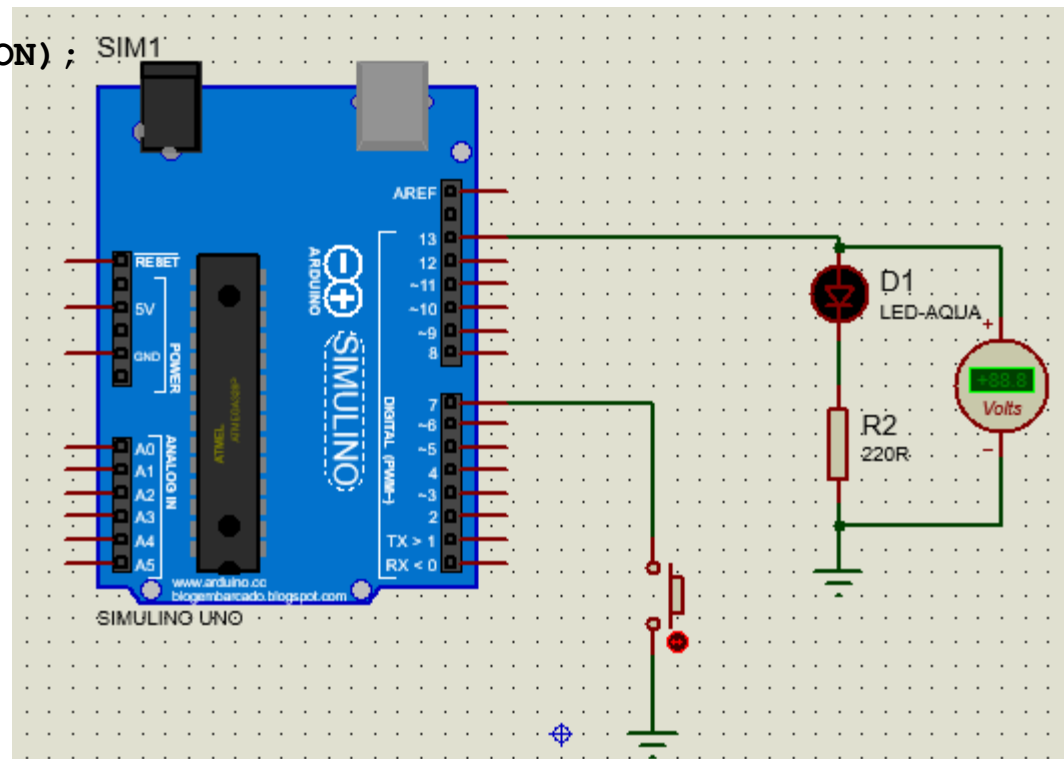


```
#define LED 13
#define BUTTON 7
```

```
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);
}
```

Simple Digital Input / Output with Input Pull-up Resistance



Multiple Digital I/O

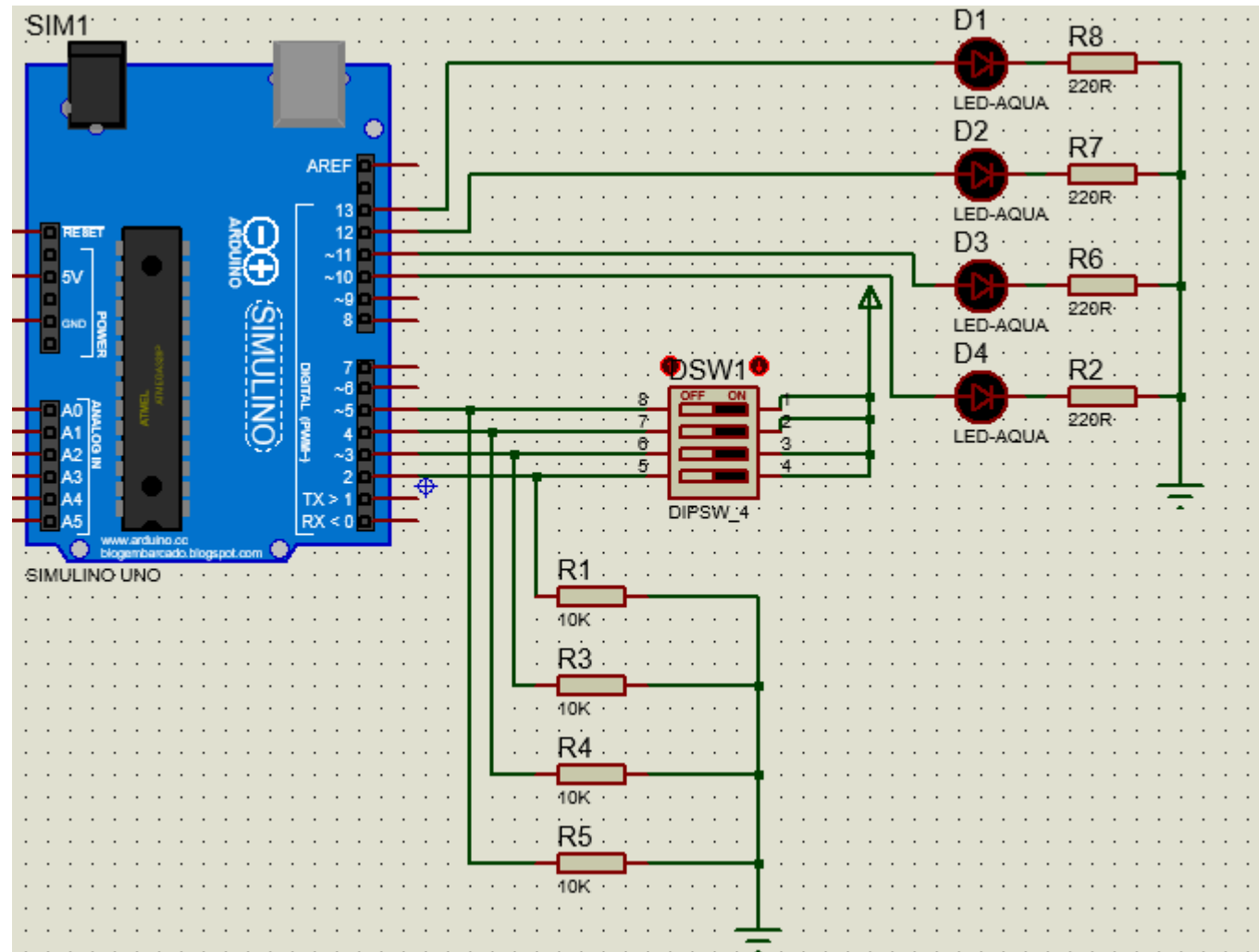
Multiple Digital I/O

```
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

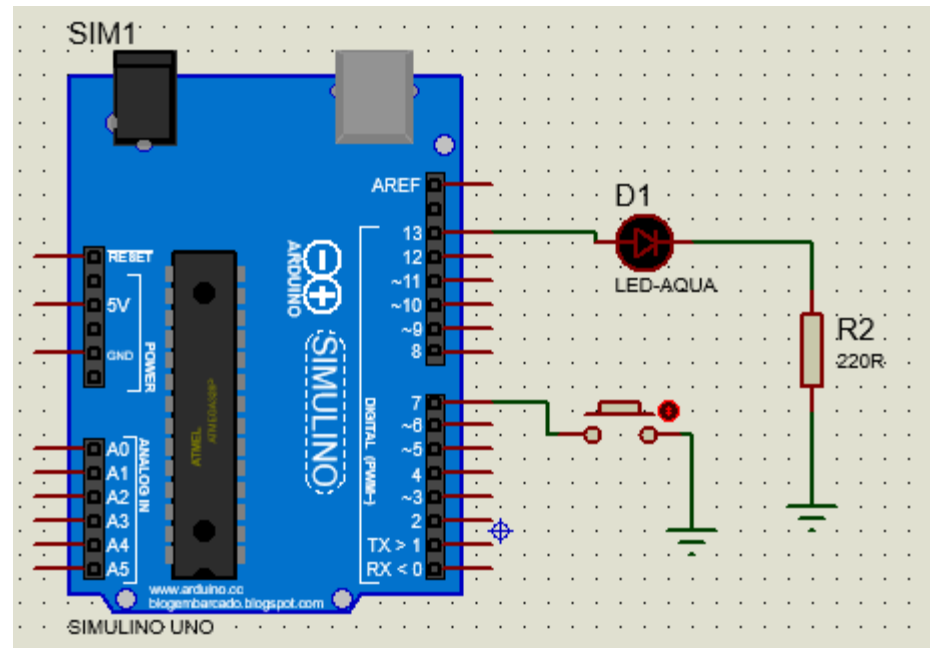


Using Interrupts


```
#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;  
}
```

Manual Detection of Push-Button Action



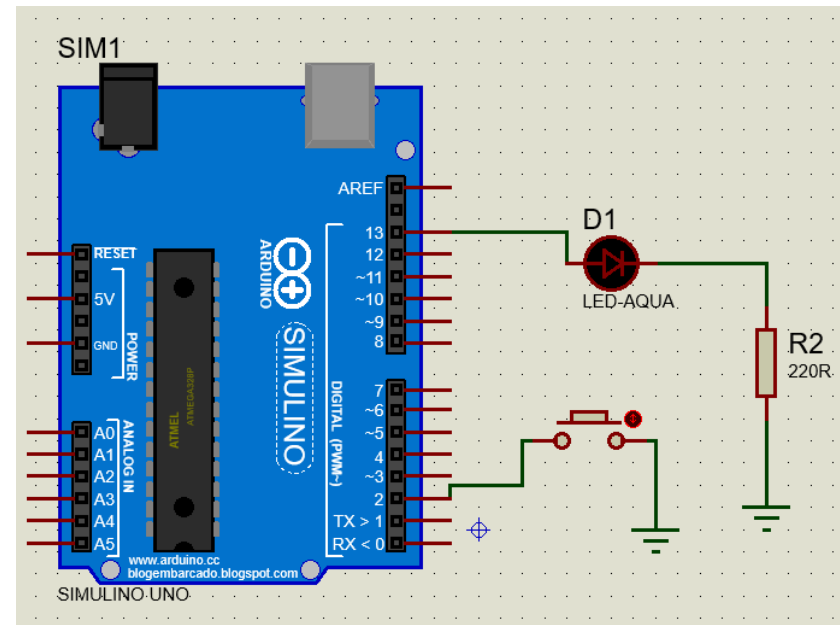
```
#define LED 13
#define BUTTON 2
```

```
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()
{
}
```

Detection of Push-Button Action using Interrupts



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();  
    //...  
}
```