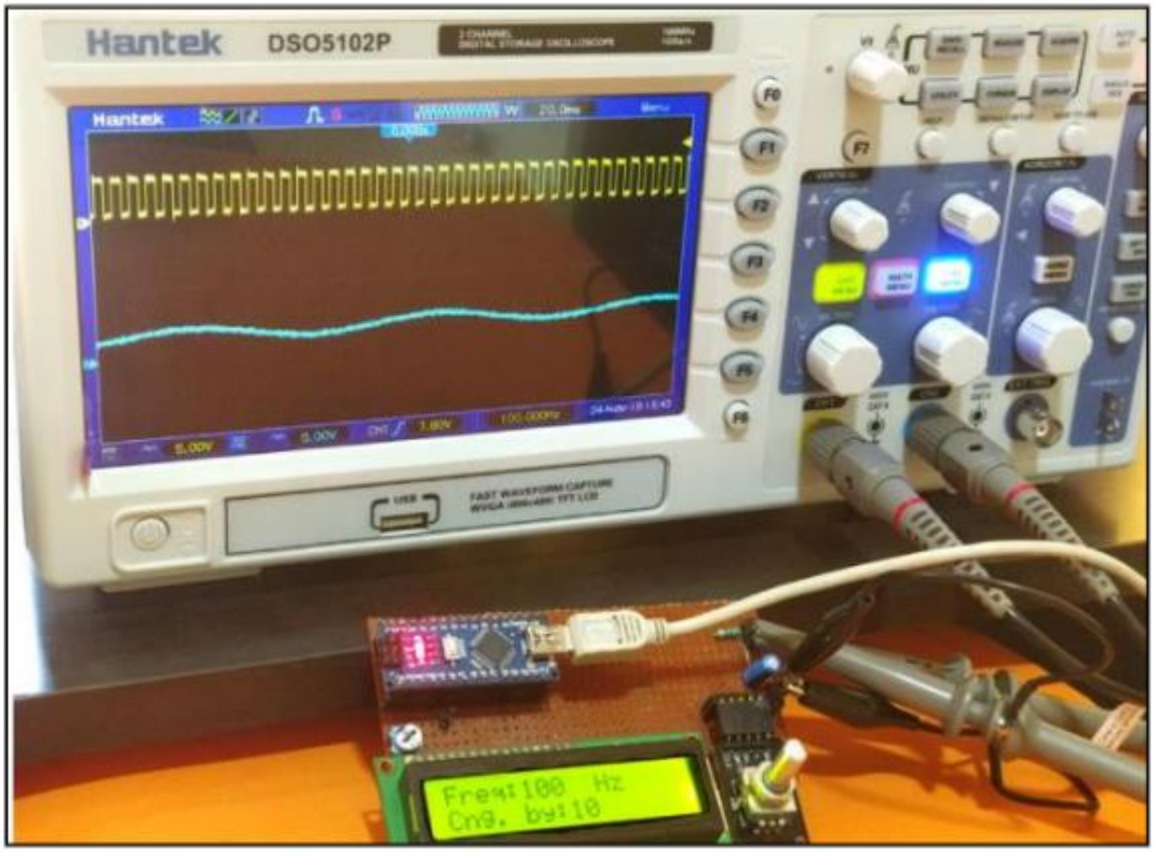
Generating of Waveforms based on Arduino



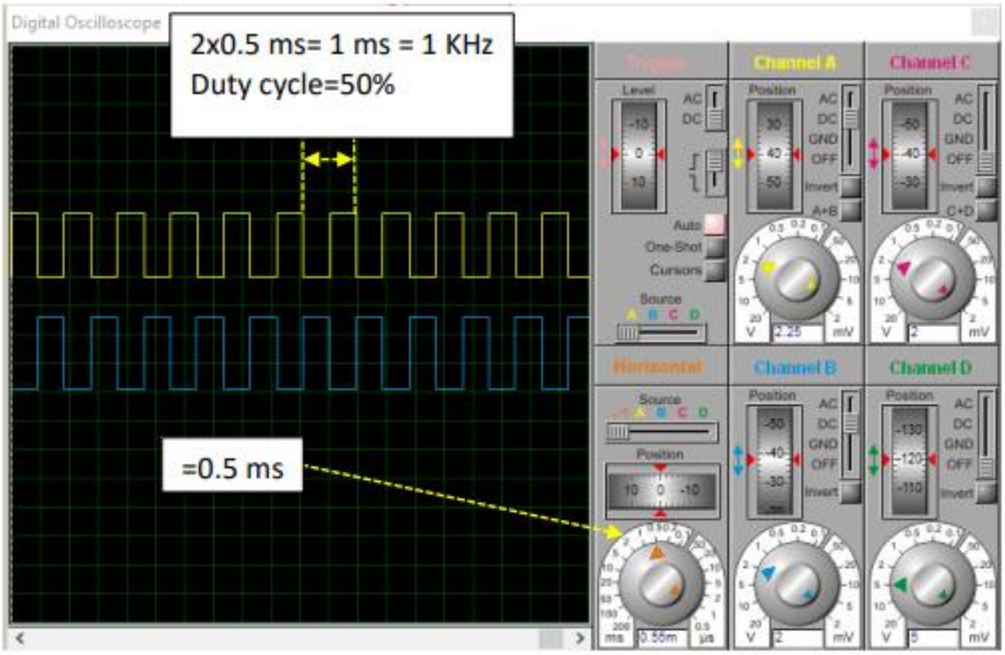
1. Square Wave

The Arduino microcontroller can be used to generate frequency. By using the delay function, the Arduino microcontroller generates a frequency depending on the delay value.

Example 1:

We want to generate a frequency of 1 kHz with a 50%duty cycle on pin 2 and 3, but one is the opposite of the other.

```
int OUT_PIN1=2;

int OUT_PIN2=3;

// the setup function runs once when you press reset or power the board

void setup() {

 // initialize digital pin LED_BUILTIN as an output.

 pinMode(OUT_PIN1, OUTPUT);
 pinMode(OUT_PIN2, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
 digitalWrite(OUT_PIN1, HIGH); // turn the LED on (HIGH is the voltage level)
 digitalWrite(OUT_PIN2, LOW); // turn the LED on (LOW is the voltage level)
 delayMicroseconds(500); // wait for 500 µs
 digitalWrite(OUT_PIN1, LOW); // turn the LED off by making the voltage LOW
 digitalWrite(OUT_PIN2, HIGH); // turn the LED off by making the voltage LOW
 delayMicroseconds(500); // wait for 500 µs
}
```



2x0.5 ms= 1 ms = 1 KHz
Duty cycle=50%

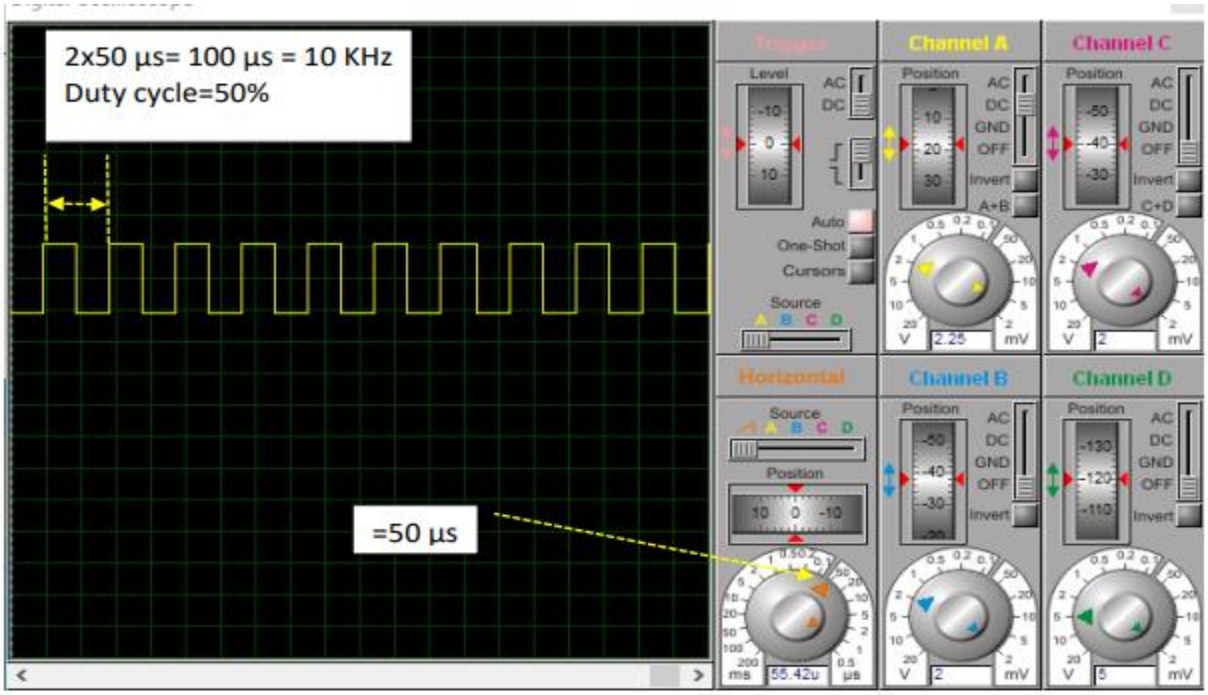
=0.5 ms

Frequency= 1 KHz, Duty cycle=50%

Example 2:

We want to generate a frequency of 10 kHz with a 50% duty cycle on pin 2.

```
int OUT_PIN=2;

// the setup function runs once when you press reset or power the board

void setup() {

 // initialize digital pin LED_BUILTIN as an output.

 pinMode(OUT_PIN, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

 digitalWrite(OUT_PIN, HIGH); // turn the LED on (HIGH is the voltage level)

 delayMicroseconds(50); // wait for 50 µs

 digitalWrite(OUT_PIN, LOW); // turn the LED off by making the voltage LOW

 delayMicroseconds(50); // wait for 50 µs

}
```



2x50 µs= 100 µs = 10 KHz
Duty cycle=50%

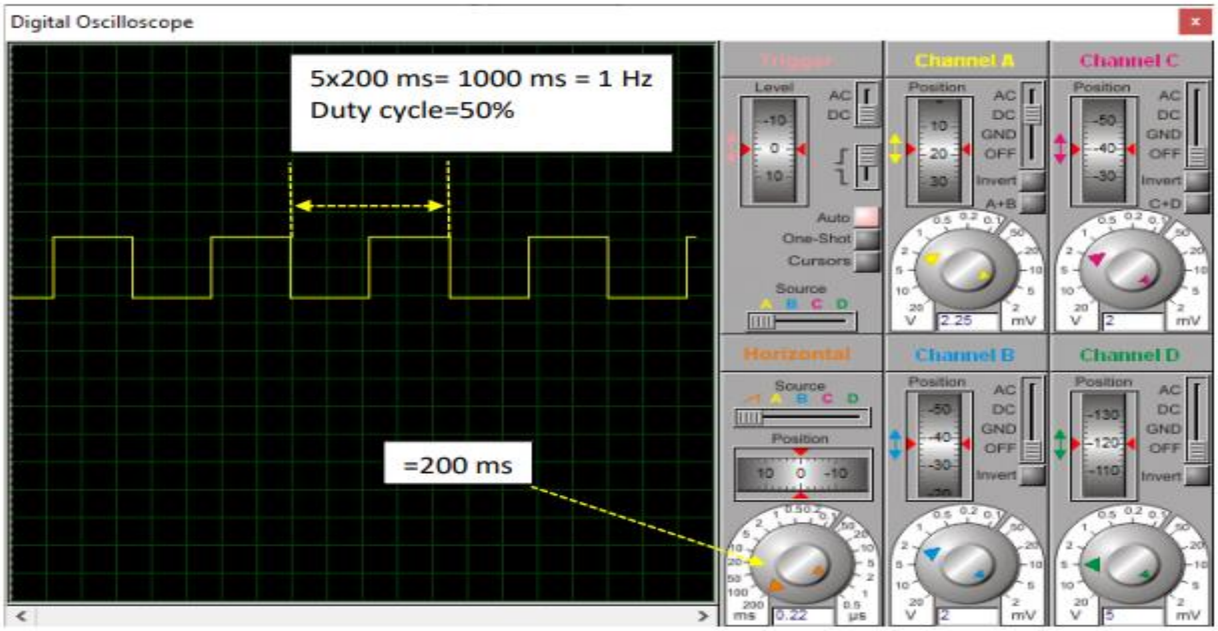
=50 µs

Frequency= 10 KHz, Duty cycle=50%

Example 4:

We want to generate a frequency of 1 Hz with a 50% duty cycle on pin 4.

```
int OUT_PIN=4;

// the setup function runs once when you press reset or power the board

void setup() {

 pinMode(OUT_PIN, OUTPUT);

}

void loop() {

 digitalWrite(OUT_PIN, HIGH); // turn the LED on (HIGH is the voltage level)

 delay(500); // wait for 500 ms

 digitalWrite(OUT_PIN, LOW); // turn the LED off by making the voltage LOW

 delay(500); // wait for 500 ms

}
```



5x200 ms= 1000 ms = 1 Hz
Duty cycle=50%

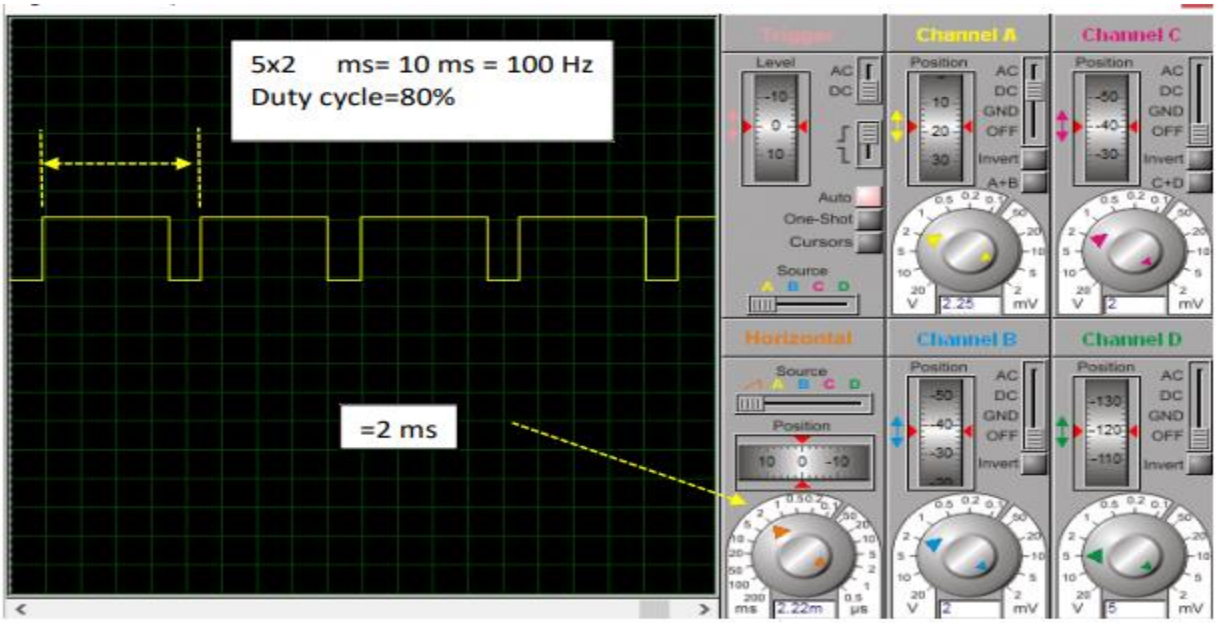
=200 ms

Frequency= 1 Hz, Duty cycle=50%

Example 5:

We want to generate a frequency of 100 Hz with 80% duty cycle on pin 6.

```
int OUT_PIN=6;
// the setup function runs once when you press reset or power the board
void setup() {
 // initialize digital pin LED_BUILTIN as an output.
 pinMode(OUT_PIN, OUTPUT);
}
void loop() {
 digitalWrite(OUT_PIN, HIGH); // turn the LED on (HIGH is the voltage level)
 delay(8); // wait for 8 ms
 digitalWrite(OUT_PIN, LOW); // turn the LED off by making the voltage LOW
 delay(2); // wait for 2 ms
}
```



5x2 ms= 10 ms = 100 Hz
Duty cycle=80%

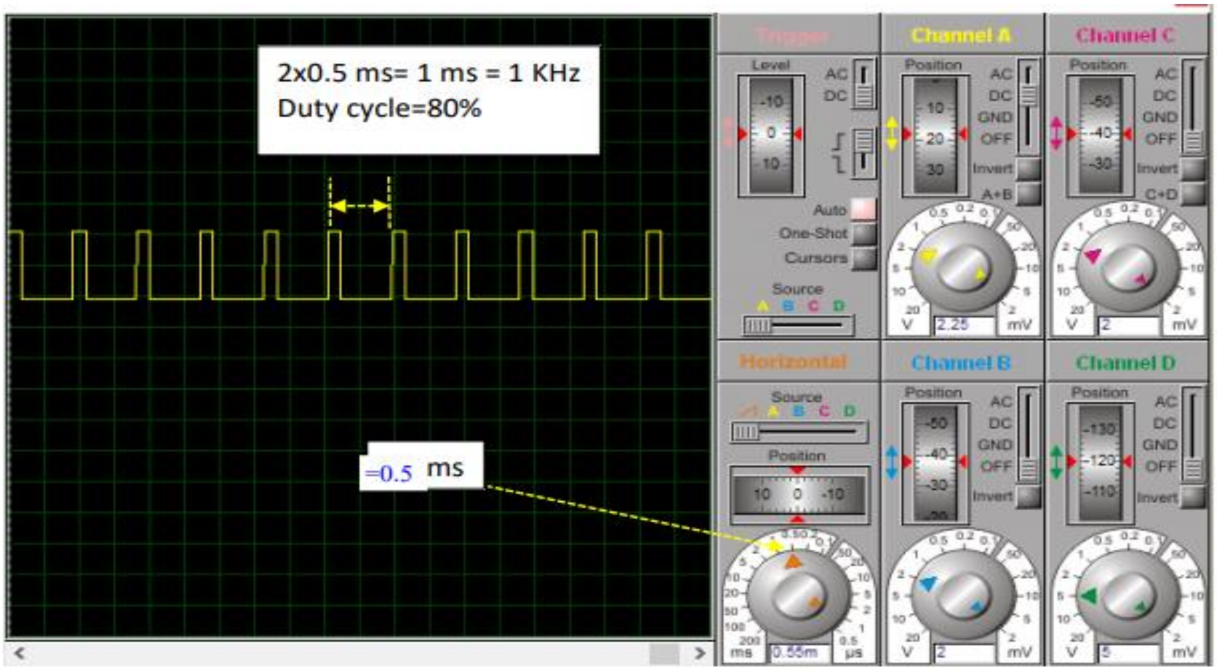
=2 ms

Frequency= 100 Hz, Duty cycle=80%

Example 6:

We want to generate a frequency of 1 KHz with 20% duty cycle on pin 10.

```
int OUT_PIN=10;
// the setup function runs once when you press reset or power the board
void setup() {
 // initialize digital pin LED_BUILTIN as an output.
 pinMode(OUT_PIN, OUTPUT);
}
void loop() {
 digitalWrite(OUT_PIN, HIGH); // turn the LED on (HIGH is the voltage level)
 delayMicroseconds (200); // wait for 200 µs
 digitalWrite(OUT_PIN, LOW); // turn the LED off by making the voltage LOW
 delayMicroseconds (800); // wait for 800 µs
}
```



2x0.5 ms= 1 ms = 1 KHz
Duty cycle=80%

=0.5 ms

Frequency= 1 KHz, Duty cycle=20%

2. Ramp Wave

Example 1:

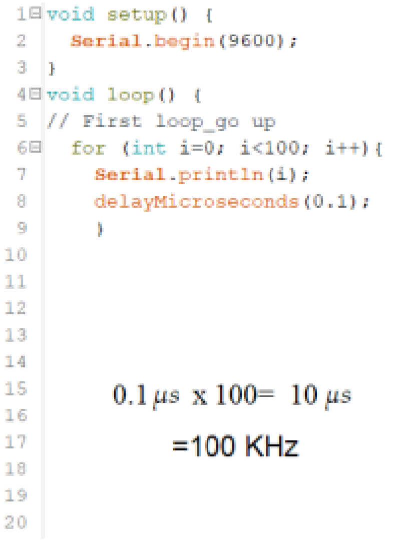
We want to generate a ramp signal with a frequency of 100 kHz.

Solution

100 KHz= 10 μs

The code as follows:

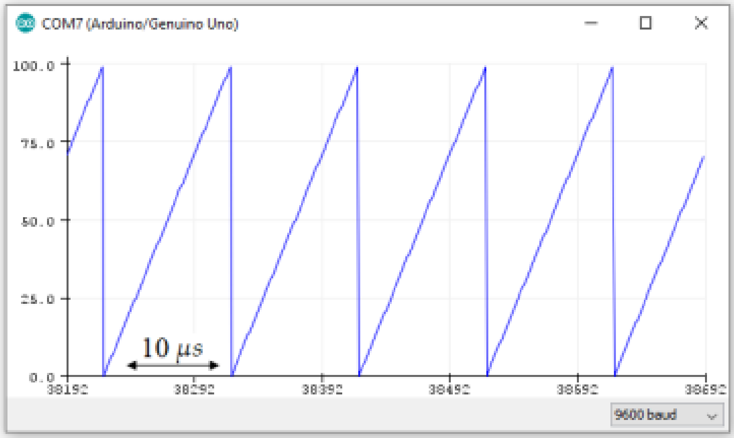
```
void setup() {
 Serial.begin(9600);
}
void loop() {
// First loop_go up
for (int i=0; i<100; i++){
 Serial.println(i);
 delayMicroseconds(0.1);
 }
```



```
void setup() {
  Serial.begin(9600);
}
void loop() {
// First loop_go up
  for (int i=0; i<100; i++){
    Serial.println(i);
    delayMicroseconds(0.1);
    }
```

$0.1\,\mu s \times 100 = 10\,\mu s$

=100 KHz



3. Sawtooth

Example:

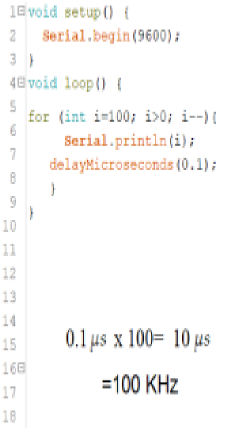
We want to generate a sawtooth signal with a frequency of 100 kHz.

<u>Solution</u>

100 KHz= 10 µs

The code as follows:

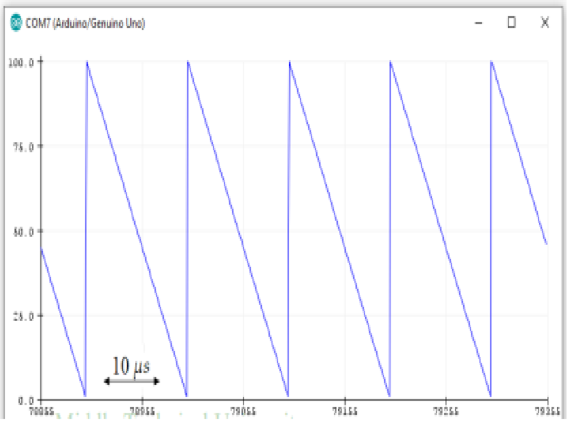
```
void setup() {
 Serial.begin(9600);
}
void loop() {
// First loop_go up
// for (int i=0; i<100; i++){
// Serial.println(i);
// delayMicroseconds(0.1);
// }
//// Second loop_flat
// for (int i=0; i<100; i++){
// Serial.println(100);
// delayMicroseconds(0.2);
//}
//Third loop_go down
for (int i=100; i>0; i--){
 Serial.println(i);
 delayMicroseconds(0.1);
 }
```



```
void setup() {
  Serial.begin(9600);
}
void loop() {
for (int i=100; i>0; i--){
    Serial.println(i);
  delayMicroseconds(0.1);
  }
}
```

0.1 μs x 100= 10 μs

=100 KHz


COM7 (Arduino/Genuino Uno)
10 μs