

CP316 Dithering

Terry Sturtevant

Wilfrid Laurier University

February 10, 2020

Discrete values

Discrete values

Integer (byte, etc) values are *discrete*.

Discrete values

Integer (byte, etc) values are *discrete*.

Sometimes you would like a value *between* two adjacent values.

Discrete values

Integer (byte, etc) values are *discrete*.

Sometimes you would like a value *between* two adjacent values.

What do you do if one is too low but the next is too high?

Example: duty cycle

Example: duty cycle

Suppose you are using PWM for control.

Example: duty cycle

Suppose you are using PWM for control.

Typically the duty cycle is given by an integer (or byte) value;

Example: duty cycle

Suppose you are using PWM for control.

Typically the duty cycle is given by an integer (or byte) value;
e.g. you would like 14.6, instead of 14 or 15.

Dithering

Dithering

If an integer value is updated frequently, then the *average* value doesn't have to be an integer.

Dithering

If an integer value is updated frequently, then the *average* value doesn't have to be an integer.

Start with the integer value *below* the desired (non-integer) target value.

Dithering

If an integer value is updated frequently, then the *average* value doesn't have to be an integer.

Start with the integer value *below* the desired (non-integer) target value.

Add a random value between -0.5 and $+0.5$.

Dithering

If an integer value is updated frequently, then the *average* value doesn't have to be an integer.

Start with the integer value *below* the desired (non-integer) target value.

Add a random value between -0.5 and $+0.5$.

Round the result to the nearest integer.

Dithering

If an integer value is updated frequently, then the *average* value doesn't have to be an integer.

Start with the integer value *below* the desired (non-integer) target value.

Add a random value between -0.5 and $+0.5$.

Round the result to the nearest integer.

This process is called **dithering**.

Dithering approximates a non-integer value;

Dithering approximates a non-integer value;
it does so by making a *noisy* version of an integer value.

Dithering approximates a non-integer value;
it does so by making a *noisy* version of an integer value.
If the updates are frequent enough, that may be sufficient.

Dithering approximates a non-integer value;
it does so by making a *noisy* version of an integer value.
If the updates are frequent enough, that may be sufficient.

The closer the target value is to an integer the less noisy it will be.

Dither sample code

Dither sample code

```
\\definition
float target_f=11.4; //desired value
int target_i; //integer to be used

//later
target_i= round(target_f
    +(float)(random(10)-4.5)/9.0);
```

random(n) produces a *long* from 0 to n-1

random(n) produces a *long* from 0 to n-1
random(10) → 0 to 9

random(n) produces a *long* from 0 to n-1

random(10) → 0 to 9

(float)random(10) -4.5 → -4.5 to 4.5

random(n) produces a *long* from 0 to n-1
random(10) → 0 to 9
(float)random(10) -4.5 → -4.5 to 4.5
(float) (random(10) -4.5) / 9 → -0.5 to 0.5

random(n) produces a *long* from 0 to n-1

random(10) → 0 to 9

(float)random(10) -4.5 → -4.5 to 4.5

(float) (random(10) -4.5) / 9 → -0.5 to 0.5

The **round**() function converts a *float* to an *int*.