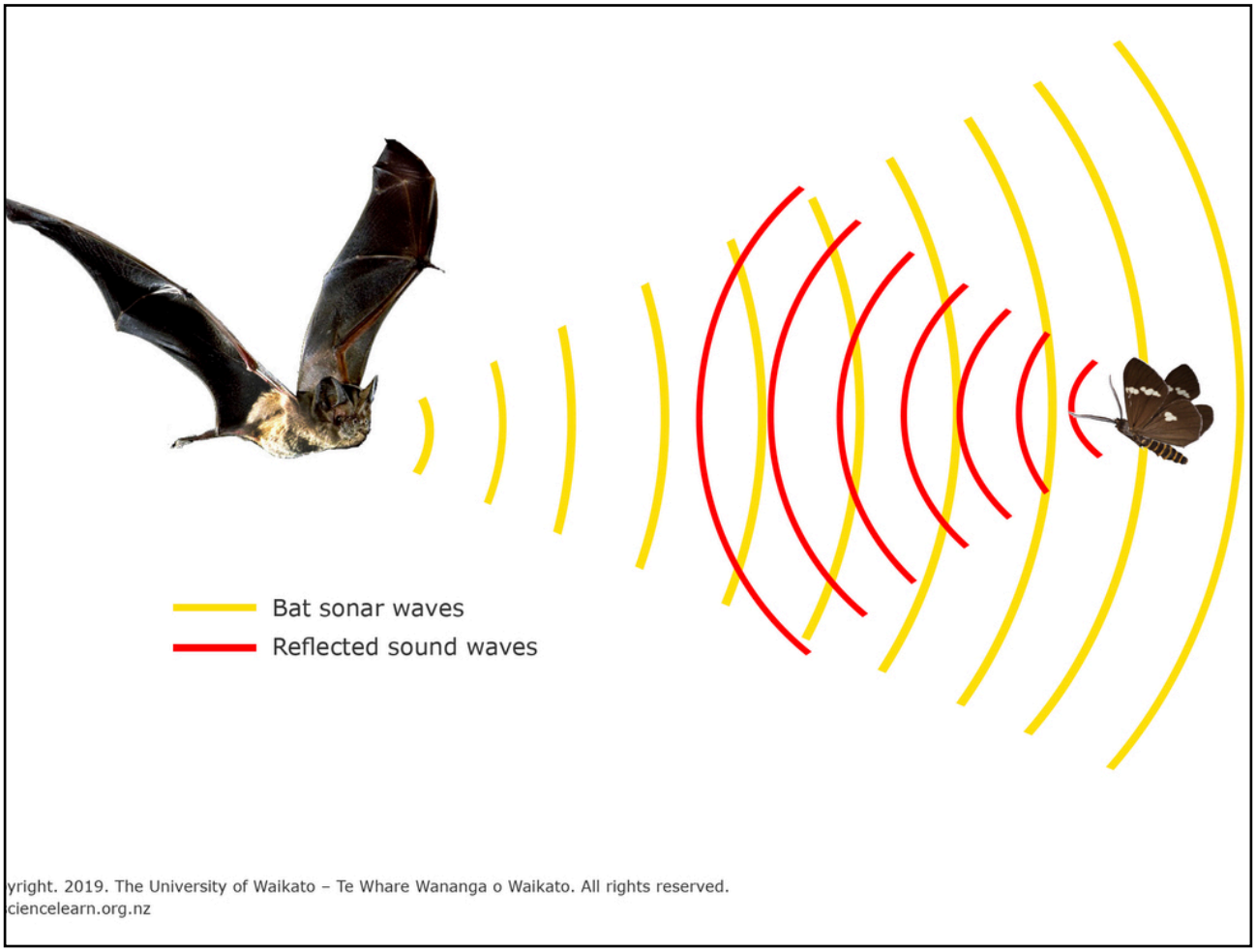

Basteltruppe
6
You're
breathtaking!

Measuring Distance

How would you measure distance?

How does Nature measure distance?

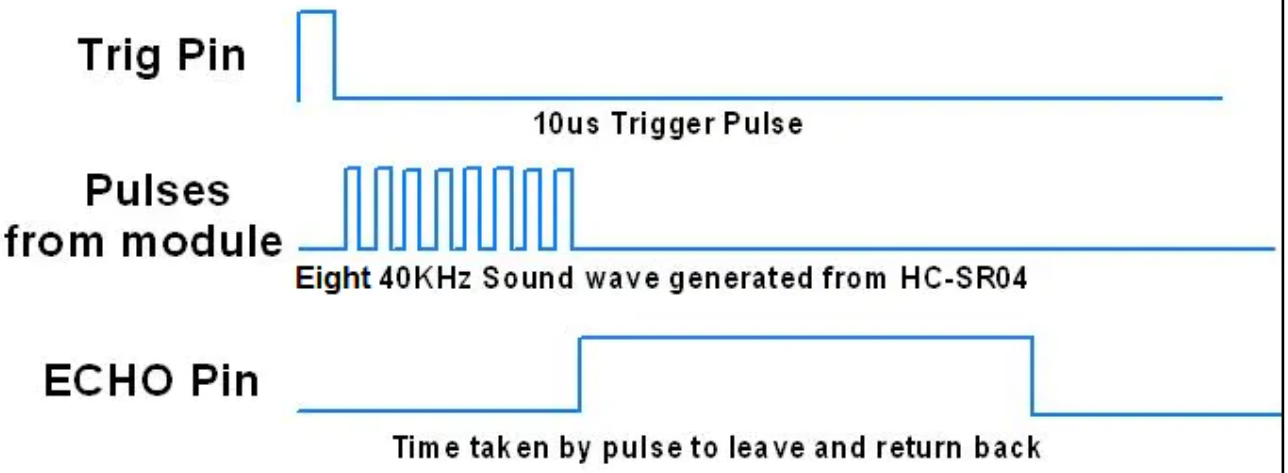

Ultrasonic



Imitation time!


HC-SR04
Vcc
Trig
Echo
Gnd
T
R

Ultrasonic HC-SR04 moduleTiming Diagram



- Set trigger pin low for 2 microseconds
- Set trigger pin high for 10 microseconds
- Use pulseIn(echoPin, HIGH) to measure how long until echoPin is HIGH
- Derive distance from Duration
- (Sound travels at 340 m/s)

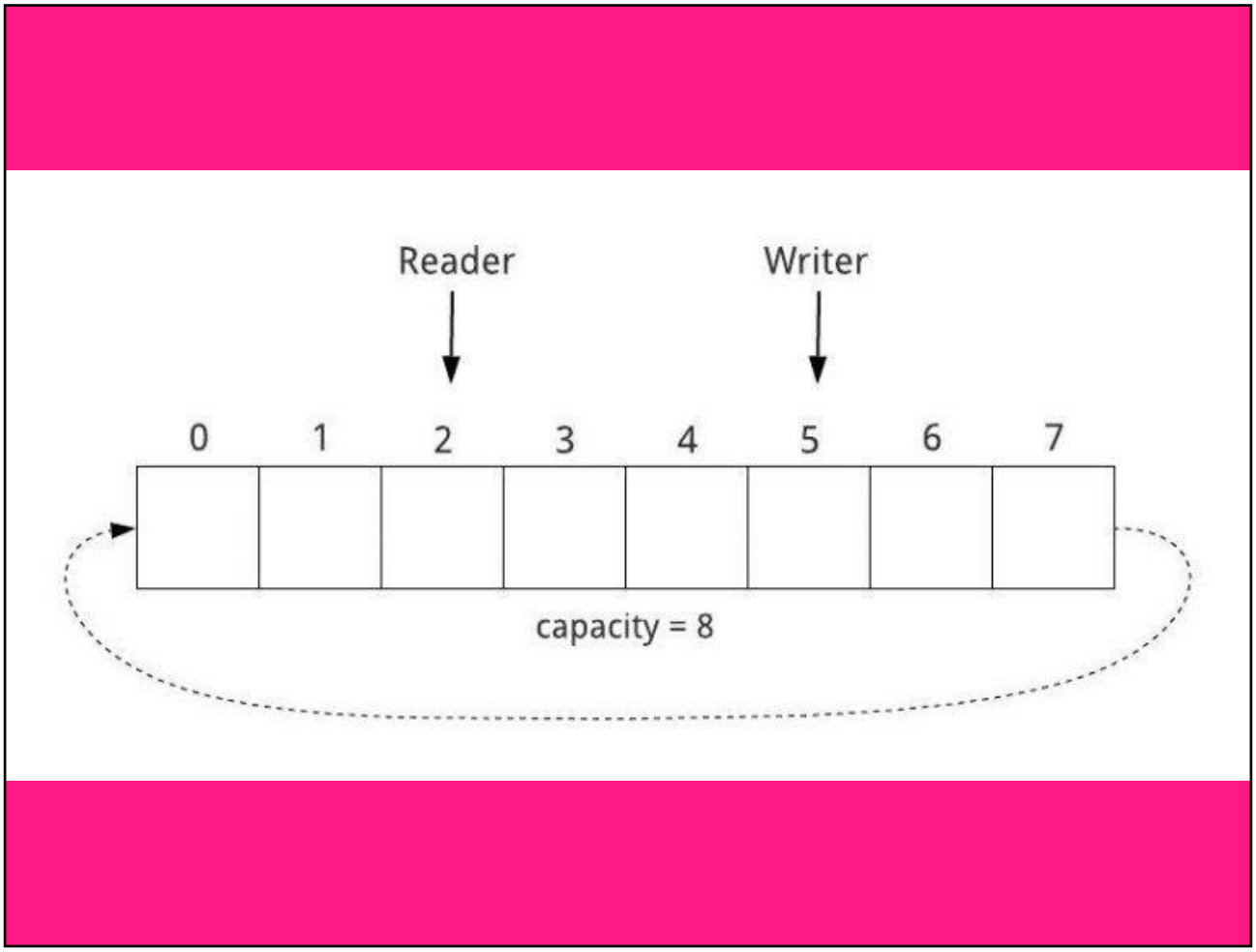
- Don't forget to set the pinModes of the trigger and Echo pins!
- Set low for 2 microseconds: LOW -> Wait 2 Microseconds
- Set high for 10 microseconds HIGH -> Wait 10 Microseconds -> LOW
- pulseIn also allows for timeouts :-)

Problem Unreliable Sensor

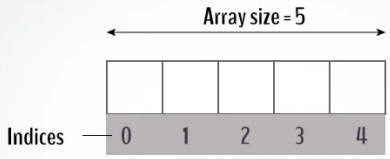
You want an average of the last X values

How would you do this?

Circular Buffer


Reader
Writer
0
1
2
3
4
5
6
7
capacity = 8

Storing Multiples of something



C Arrays

Oh No, you need more C :-(

```
uint16_t measurements [3];
measurements[0] = 10;
measurements[1] = 5;
measurements[2] = 8;
uint16_t average = 0;
for(int i = 0; i < 3; i++){
    average += measurements[i];
}
average /= 3;
```

- First you create an array with the size 3
- Then you assign 3 values to the various indices
- To calculate the average you first make a temporary value in which to store the sum
- You iterate over all values and add them to the sum
- Finally you divide by the number of elements

Ring Buffer

```
typedef struct RingBuffer{
    uint16_t index;
    uint16_t length;
    uint16_t values[10];
} RingBuffer;
```

- You create a struct for your ringbuffer to store associated information such as the current index and the length

Write to Circular Buffer

```
RingBuffer writeToBuffer(Ringbuffer buff,
uint16_t value){
    buff.values[buff.index] = value;
    buff.index++;
    if(buff.index == buff.length){
        buff.index = 0;
    }
        return buff;
}
```

- To write to the ring buffer you assume that the index always writes to the next place to write to
- First you write your value
- Then you increase the index such that the next value fits there
- If the index is at maximum, i.e. the length of your array, you reset it to 0
- Finally, you return your array!
- Currently you are using call by value, its quite inefficient but we haven't had pointers yet, so just use that one for now

Get Average

```
uint16_t getAverage(Ringbuffer
buff){
    uint32_t avg = 0;
    for(uint16_t i = 0; i <
buff.length; i++){
        avg += buff.values[i];
    }
    return avg/buff.length;
}
```

- To get the average you first have a temporary value again
- In a for loop, iterate over all values
- Divide by the number of elements

Exercises

- Measure the distance with your Arduino!
- Use a Ringbuffer to get a more reliable distance estimation
- How does the size of the Ring buffer affect your results?
- Try the median instead of the average, what does change?