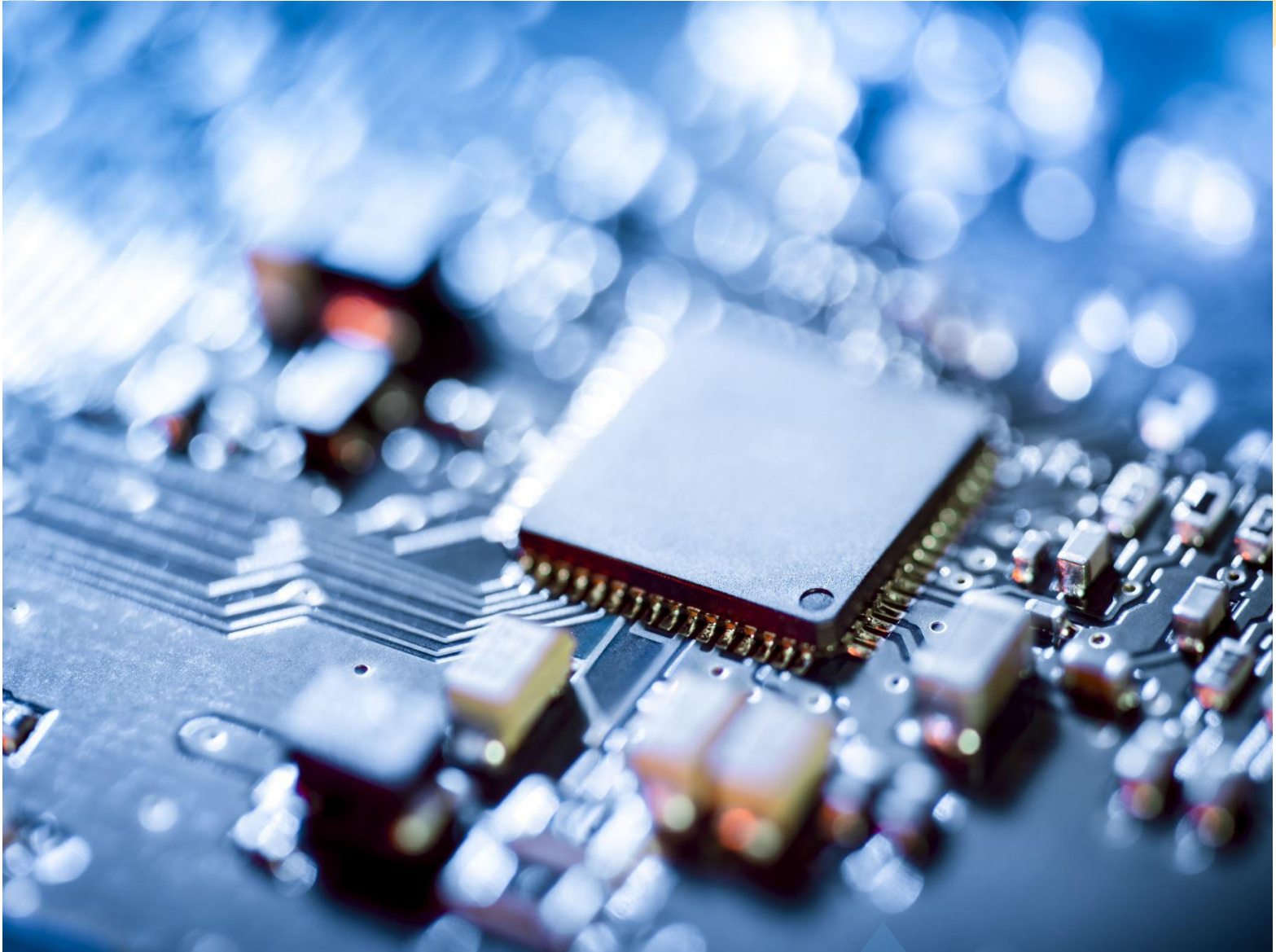
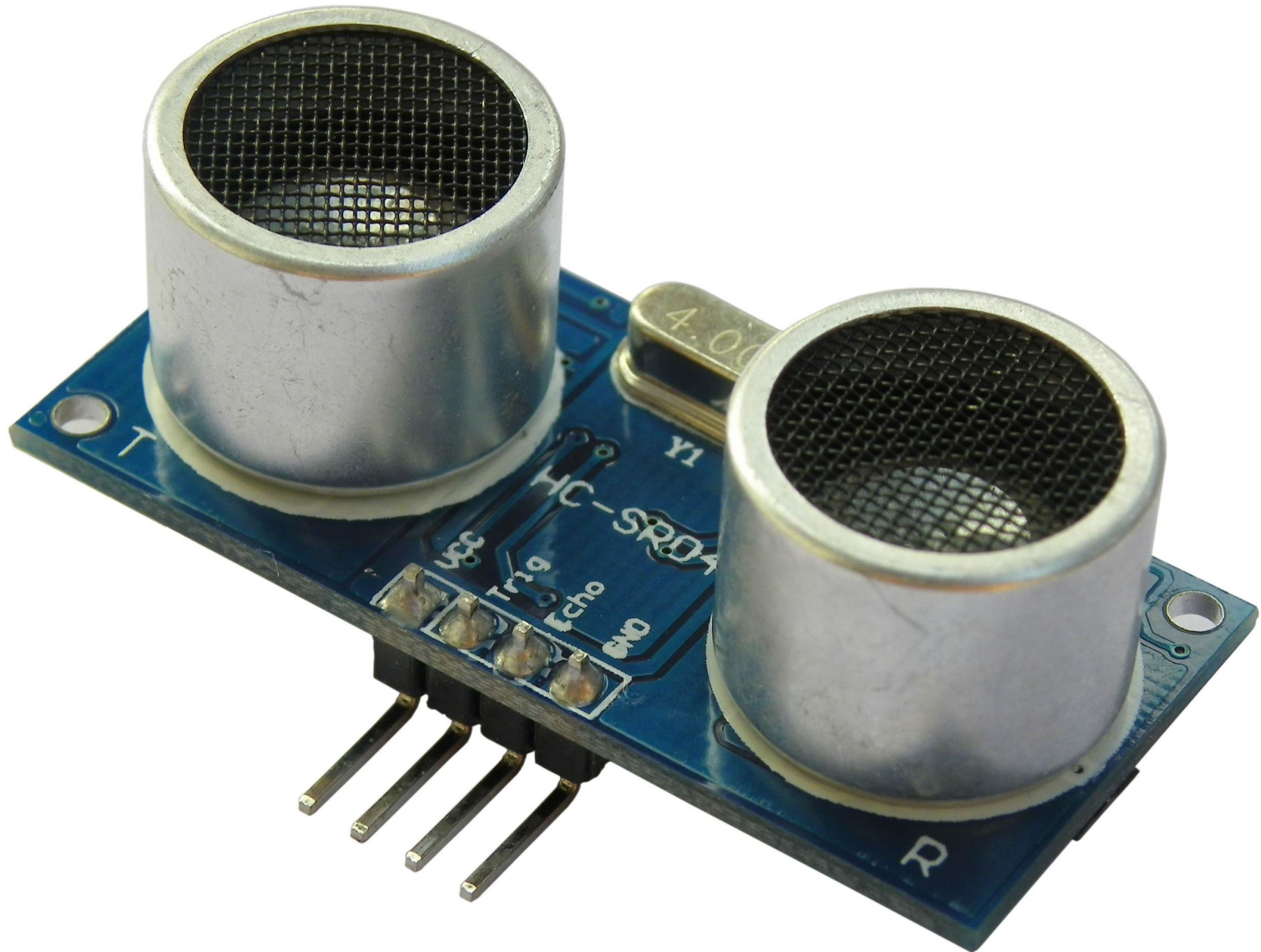
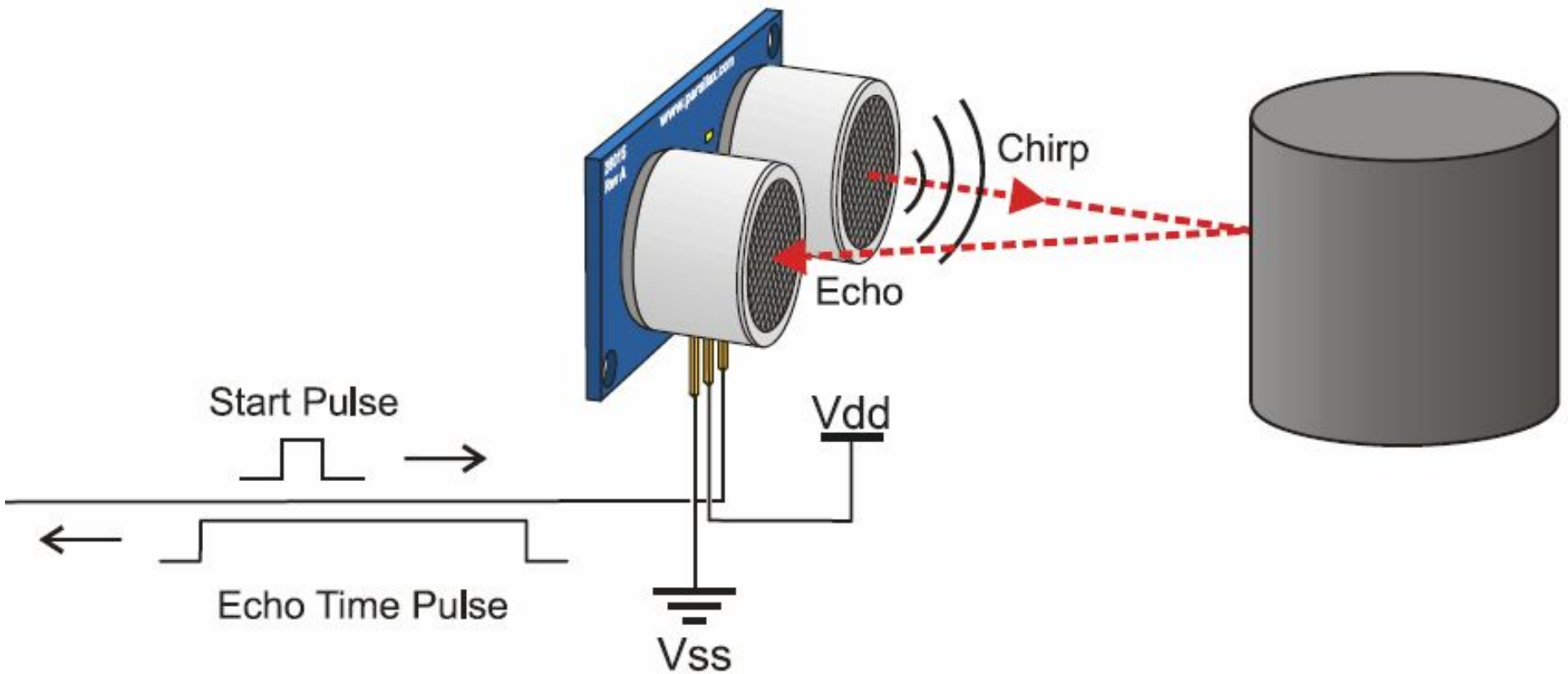
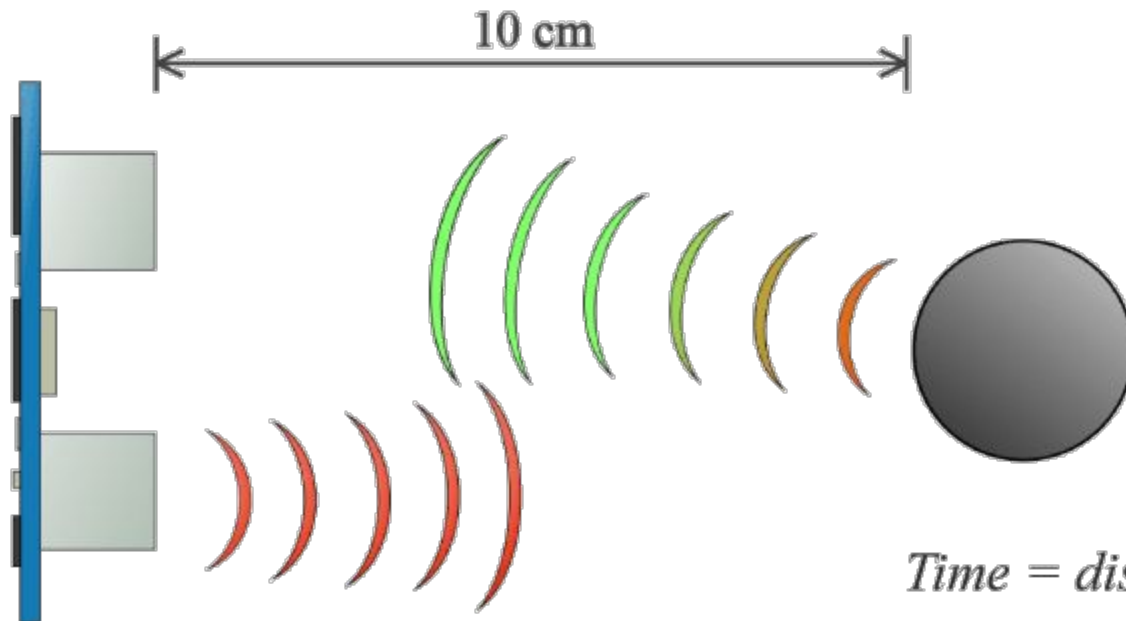


Ultrasonic Sensor









speed of sound:

$$v = 340 \text{ m/s}$$

$$v = 0,034 \text{ cm}/\mu\text{s}$$

Time = distance / speed:

$$t = s / v = 10 / 0,034 = 294 \mu\text{s}$$

Distance:

$$s = t \cdot 0,034 / 2$$

Trigger the start of pulse



```
digitalWrite(trigPin,  
HIGH);
```



```
delay (10);
```



```
digitalWrite(trigPin,  
LOW);
```


pulseIn()

[Advanced I/O]

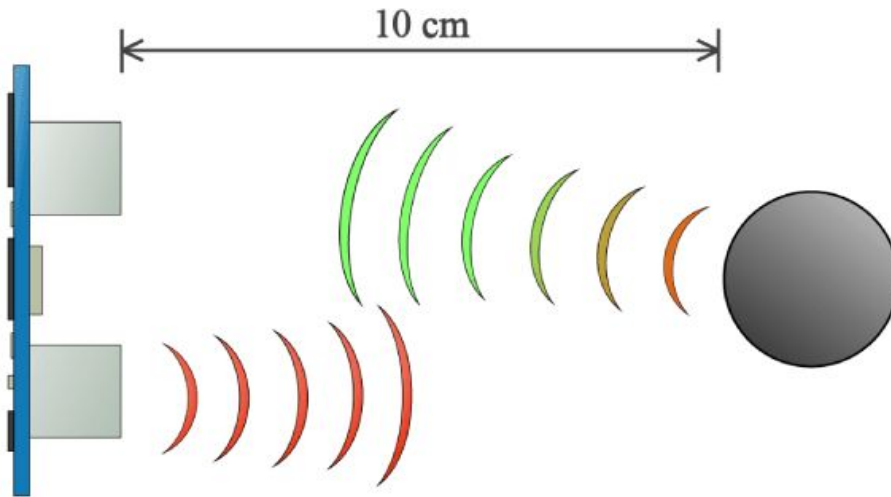
Description

Reads a pulse (either **HIGH** or **LOW**) on a pin. For example, if **value** is **HIGH**, **pulseIn()** waits for the pin to go from **LOW** to **HIGH**, starts timing, then waits for the pin to go **LOW** and stops timing. Returns the length of the pulse in microseconds or gives up and returns 0 if no complete pulse was received within the timeout.

The timing of this function has been determined empirically and will probably show errors in longer pulses. Works on pulses from 10 microseconds to 3 minutes in length.

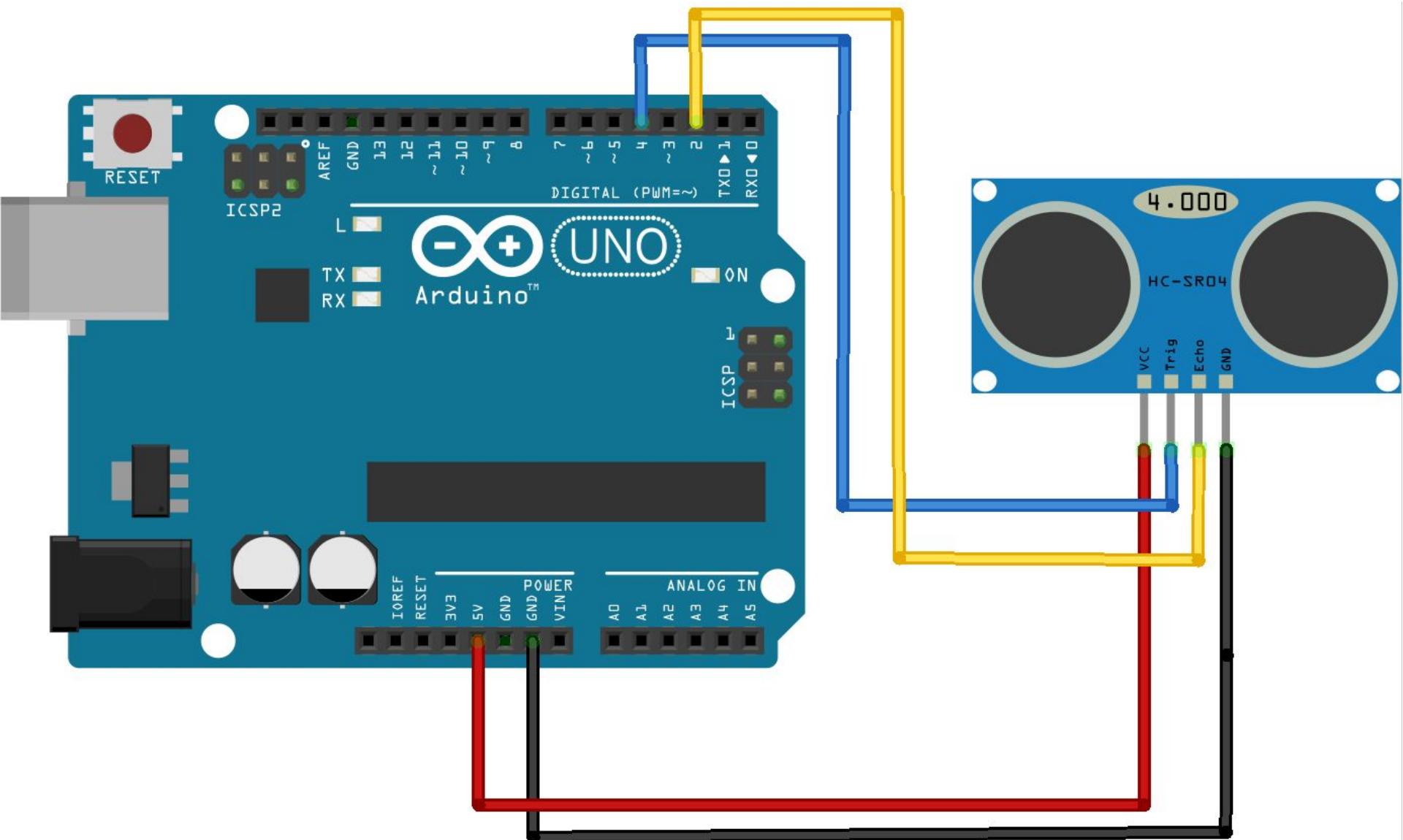
Syntax

```
pulseIn(pin, value)  
pulseIn(pin, value, timeout)
```



```
duration = pulseIn(echoPin,  
HIGH);  
distance= duration*0.034/2;
```

Compute the distance



Sketch to compute distance

```
// Sets the trigPin on HIGH state for 10  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);
```

```
// Reads the echoPin, returns the sound  
wave travel time in microseconds  
duration = pulseIn(echoPin, HIGH);
```

```
// Calculating the distance  
distance= duration*0.034/2;
```

```

const int trigPin = 4; // defining the pins

const int echoPin = 2;

// defining variables

long duration;

int distance;


void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600); // Starts the serial communication

}


void loop() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance= duration*0.034/2;

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.println(distance);

}

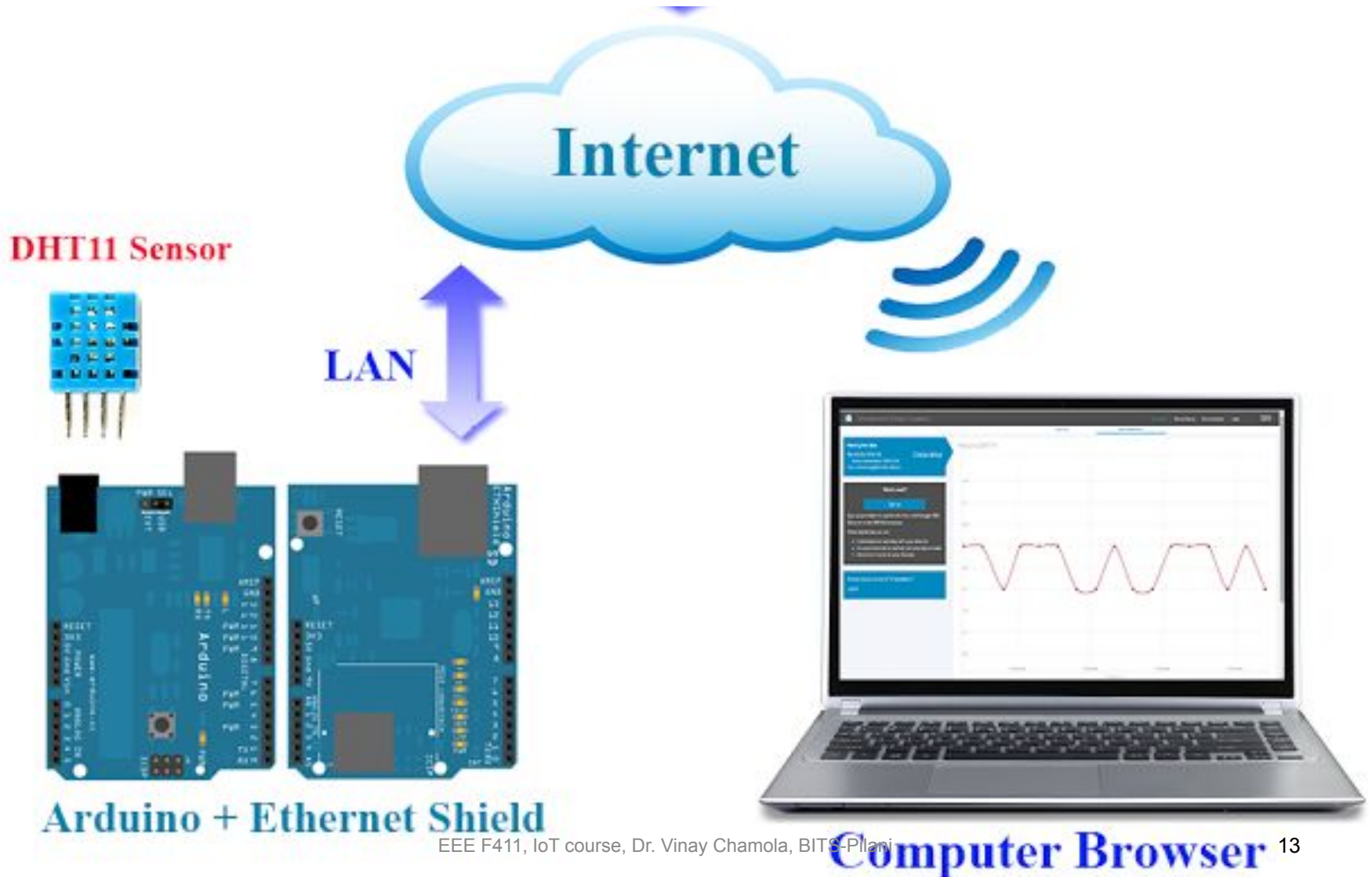
```

Arduino IoT

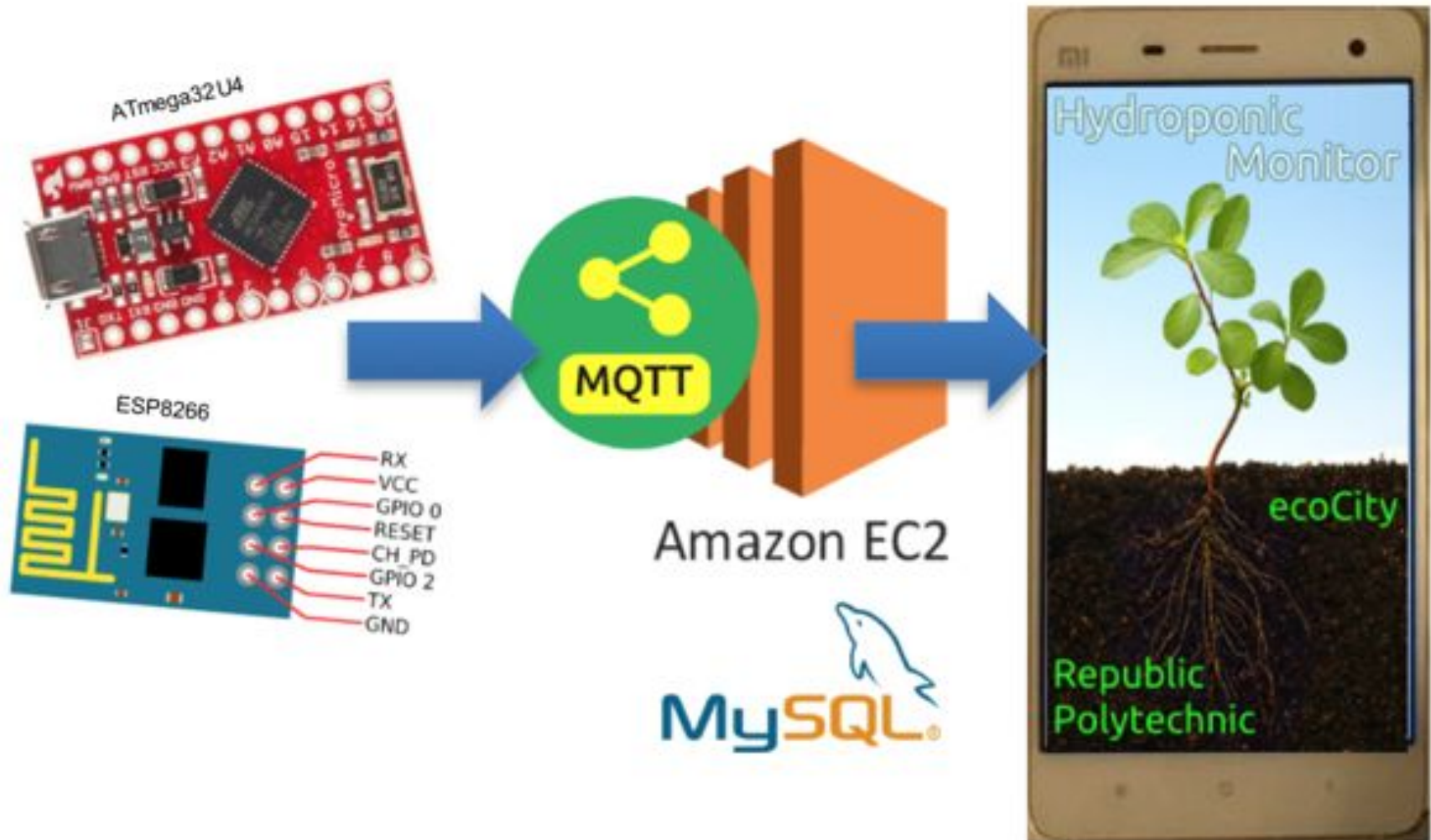
Arduino IoT

- Arduino + Ethernet shield
- Arduino + WiFi shield
- Arduino + WiFi module
- Arduino Yun
- Arduino Xbee Bluetooth to Raspberry Pi

Arduino IoT



Arduino IoT



Resources

- Arduino Guide <http://www.arduino.cc/guide>
- Arduino IDE <http://www.arduino.cc/ide>
- Arduino Troubleshooting <http://www.arduino.cc/trouble>
- Arduino Tutorial <http://www.arduino.cc/tutorial>
- Arduino Examples <http://www.arduino.cc/examples>