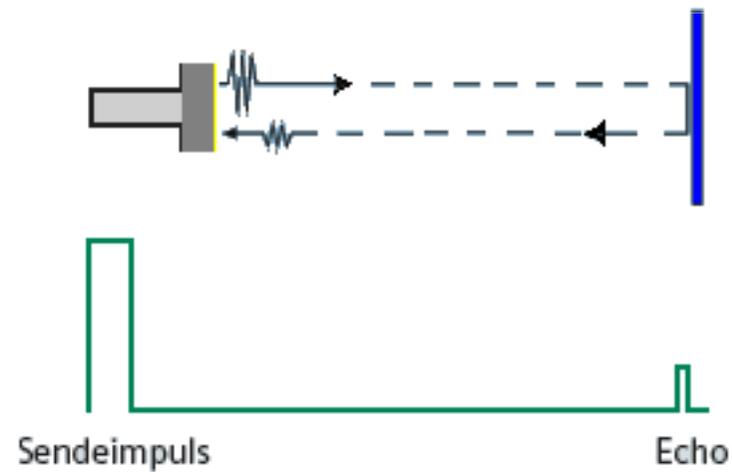


Supersonic

# Funktionsweise

Ein Ultraschallsensor sendet Schallwellen aus und misst den Zeitintervall bis zum Empfang des Echos.



# Anwendung

- Distanzmessgerät
- Detektion von Objekten
- Annäherung / Positionierung
- Automatisierungstechnik
- Kollisionsschutz
- Durchflussmessung

# Messfehler

- Unterschiedliche Schallgeschwindigkeiten bei unterschiedlichen Temperaturen
- Spiegel verfälschen das Ergebnis
- Bei flachem Winkel keine Reflexion!



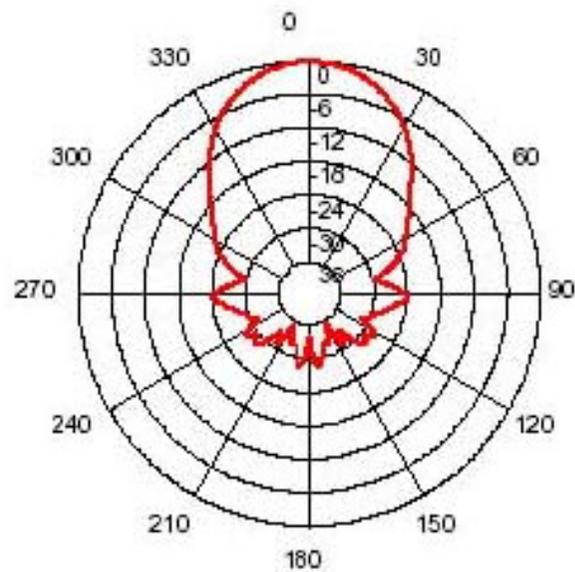
# Vorkommen in der Natur

- Delphin (Navigation, Kommunikation, Lokalisierung von Fischen)
- Fledermaus (Navigation, Lokalisierung von Beute etc.)

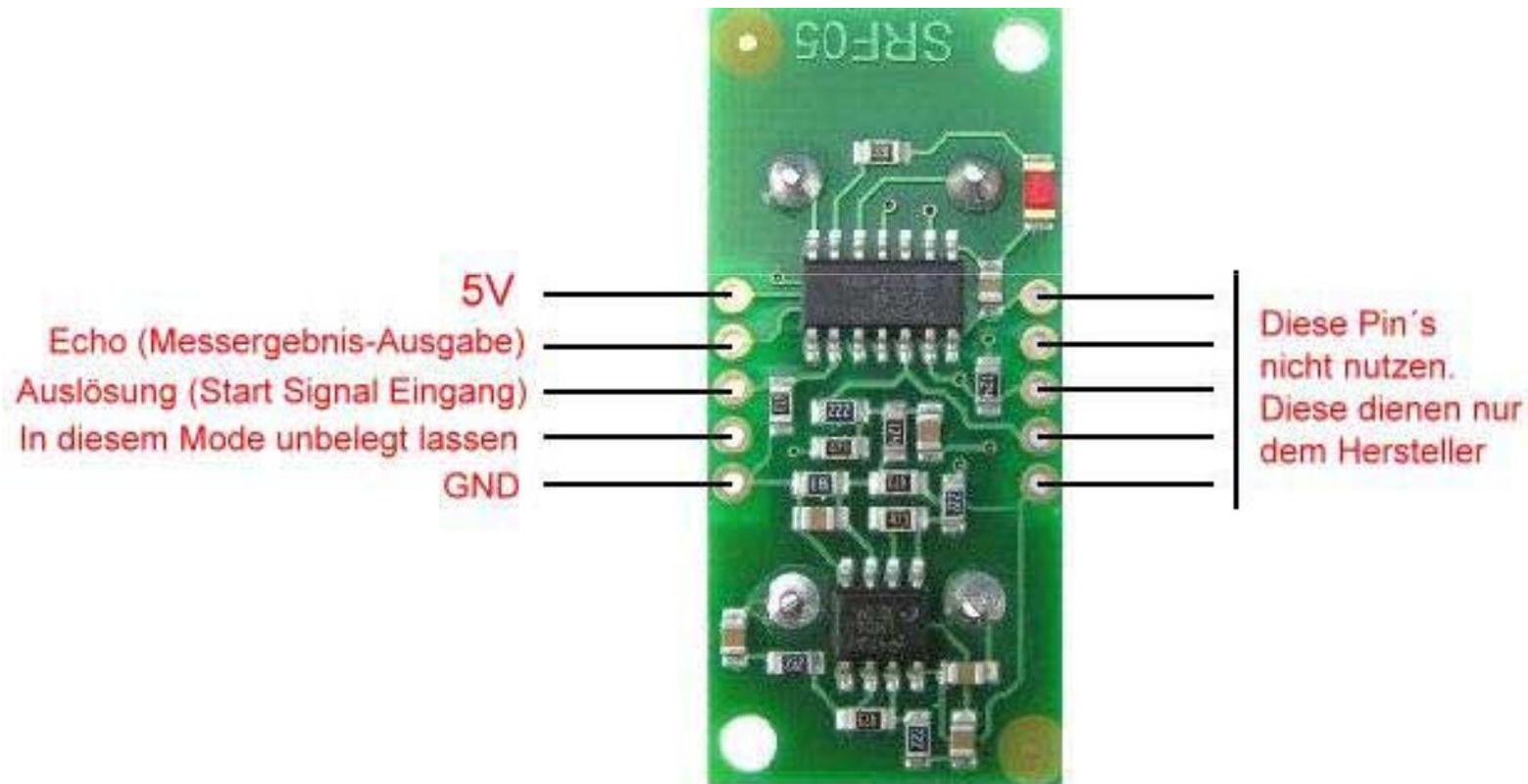


# SRF05 Datenblatt

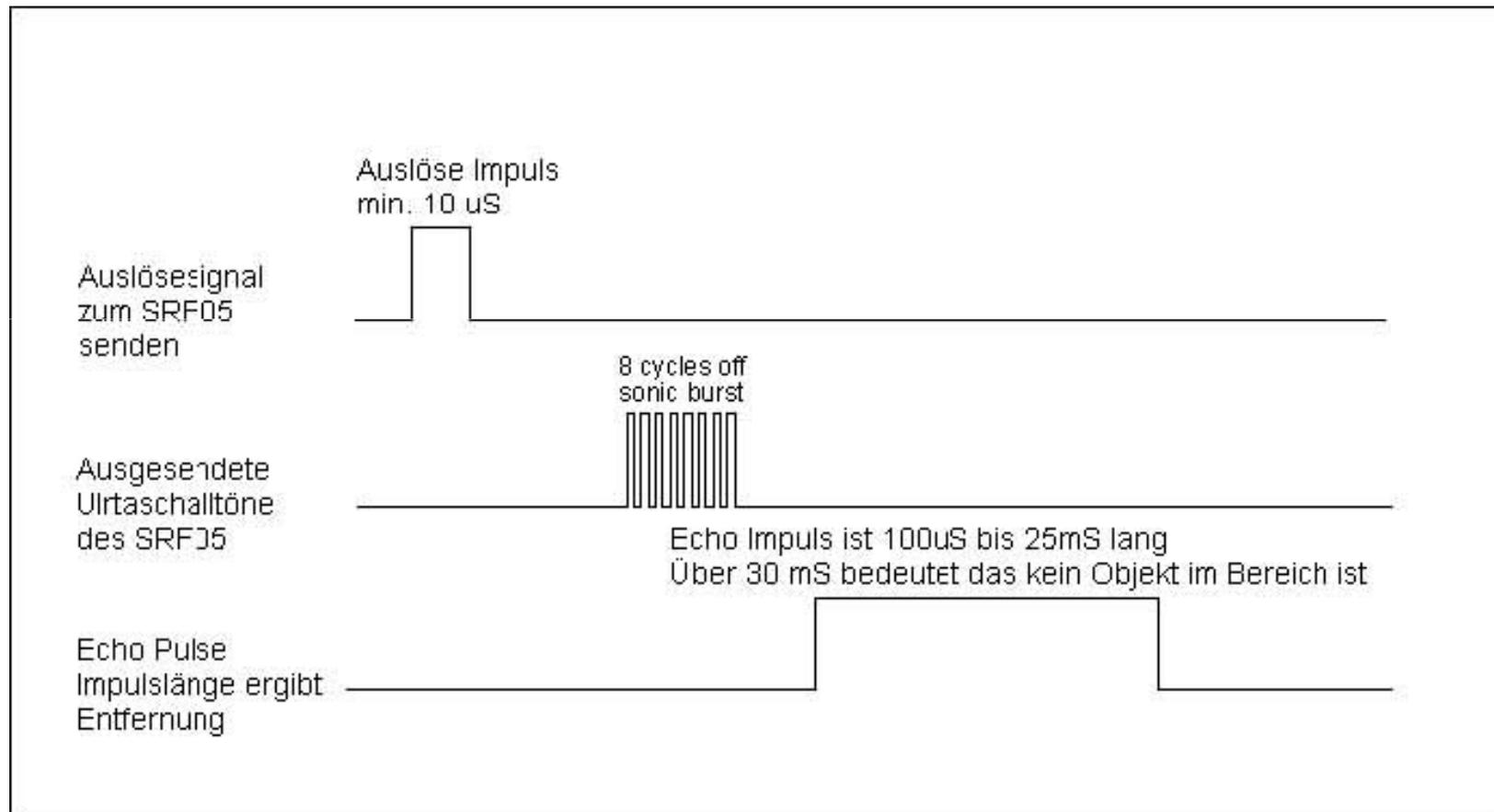
- SRF05 ist ein LowCost Ultraschallsensor
- 4m Reichweite
- Impulsmessung (SRF10 funktioniert mit I2C)



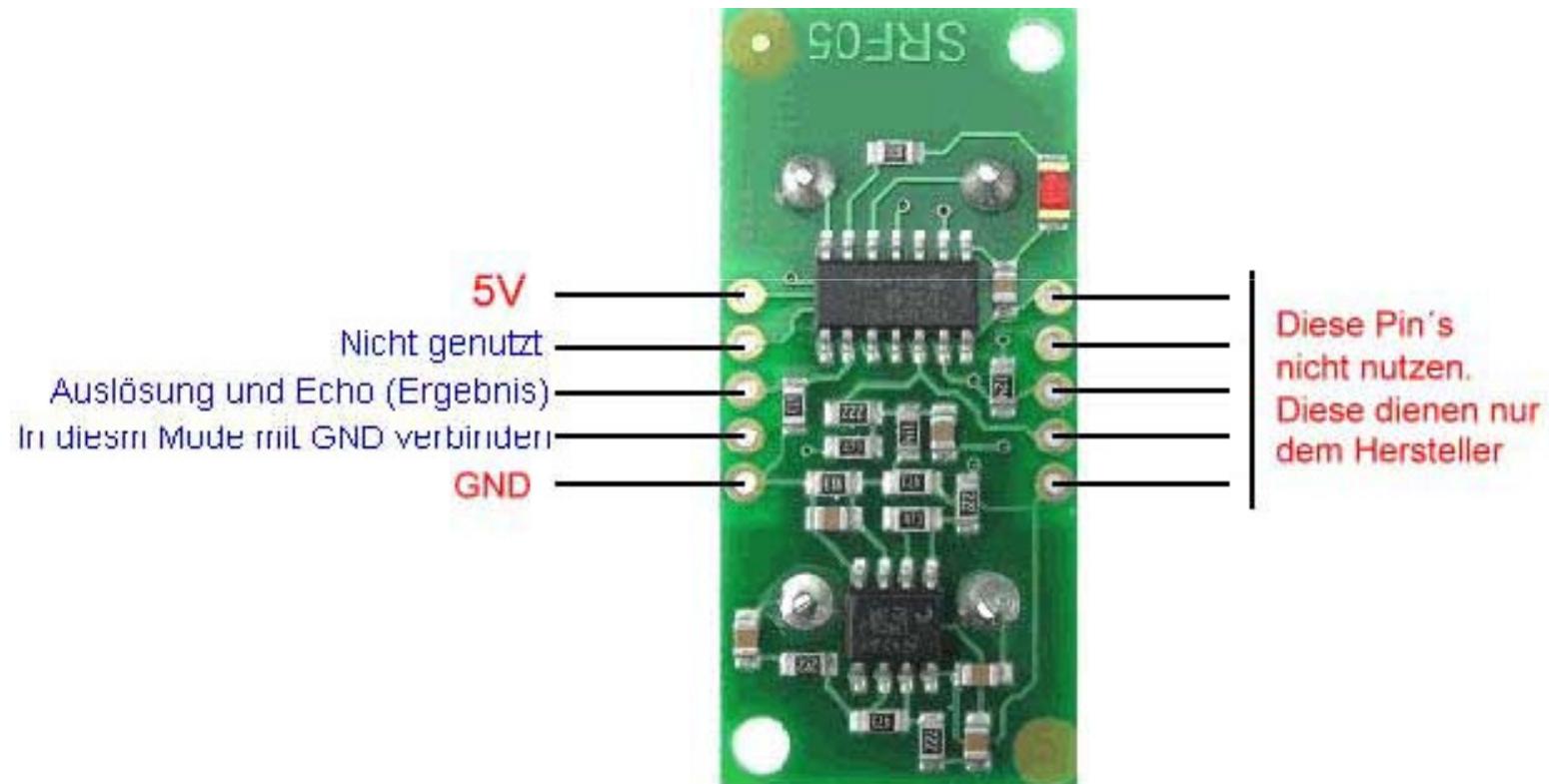
# Mode 1: Pinbelegung



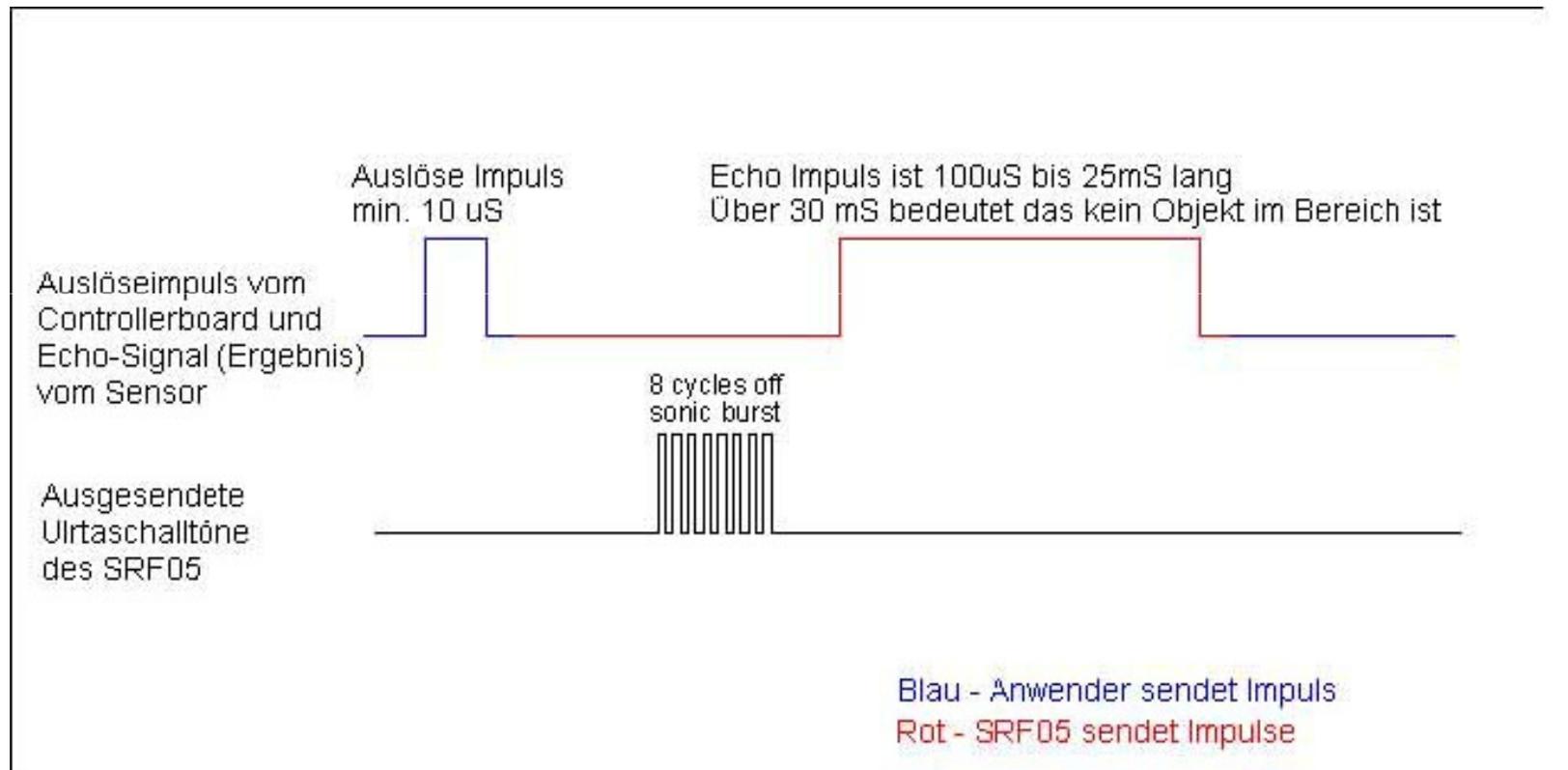
# Mode 1: Timing Diagram



## Mode 2: Pinbelegung



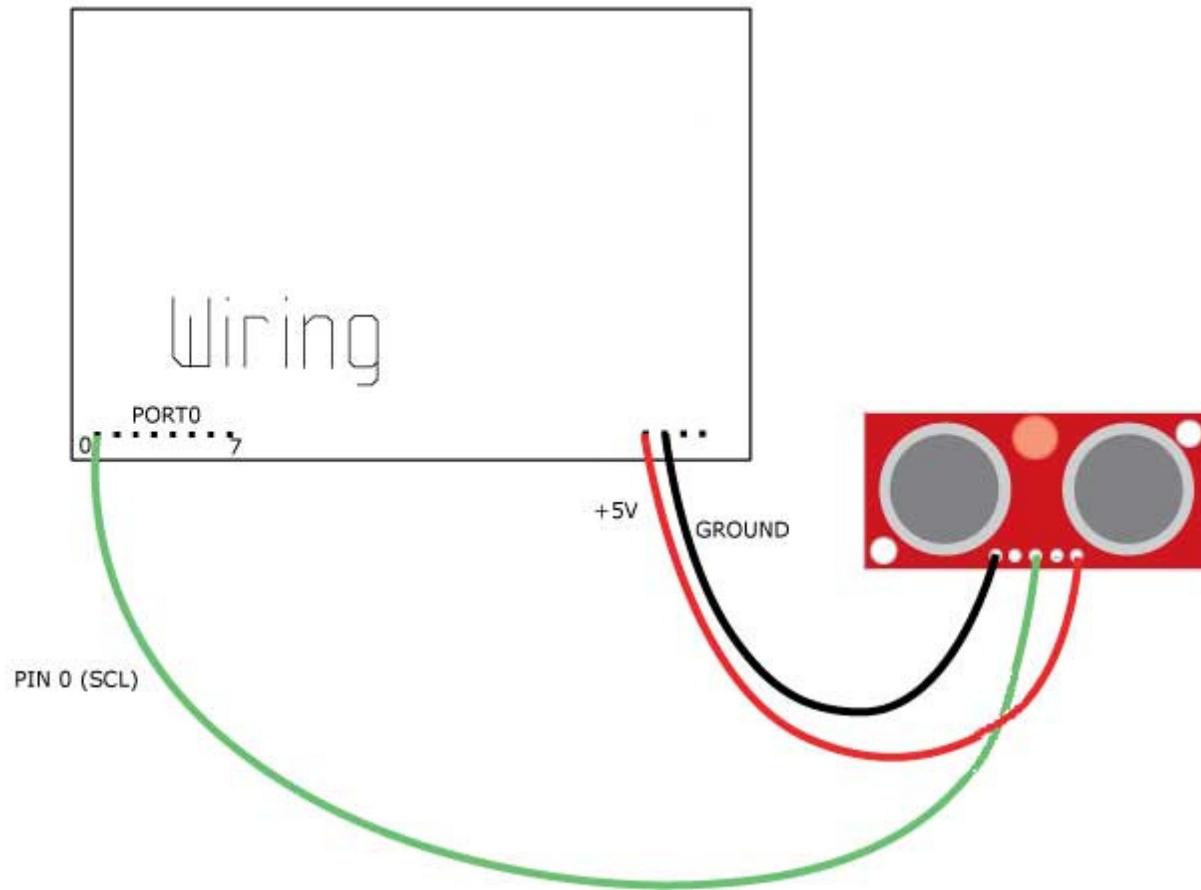
# Mode 2: Timing Diagram





## Wiring-Beispiel

# Verdrahtung



# Wiring - Impulsmessung

- Pulse Input (polled)  
**pulseIn ()**
- Pulse output generation  
pulseOut ()  
pulseRunning ()  
pulseStop ()

# Wiring – pulseIn()

Name	<b>pulseIn()</b>						
Examples	<pre>int inpin = 0; int val = 0;  void setup() {   pinMode(inpin, INPUT); }  void loop() {   val = pulseIn(inpin, HIGH); }</pre>						
Description	The <b>pulseIn()</b> method returns the length in microseconds on a digital input pin for a short period of time from 10 milliseconds to 3 minutes. It can be used to read sensors that return a pulse train like some accelerometers or rangars. The user can specify the counting on HIGH or LOW transitions. The pulseIn command stops all activity while executing. A timeout of 1 second (1000000 microseconds) prevents the command to wait forever. A specific timeout in microseconds can also be specified.						
Syntax	<pre>pulseIn(pin,transition) pulseIn(pin,transition,timeout)</pre>						
Parameters	<table><tr><td>pin</td><td>int: the pin used for reading the pulse</td></tr><tr><td>transition</td><td>HIGH or LOW</td></tr><tr><td>timeout</td><td>int: a timeout value in microseconds</td></tr></table>	pin	int: the pin used for reading the pulse	transition	HIGH or LOW	timeout	int: a timeout value in microseconds
pin	int: the pin used for reading the pulse						
transition	HIGH or LOW						
timeout	int: a timeout value in microseconds						
Returns	int						

# Wiring – setup ()

```
// Const

static int digitalPin = 0;

// Setup

void setup() {

    //LED (2 * blink)
    pinMode(48, OUTPUT);
    digitalWrite(48, HIGH);
    delay(200);
    digitalWrite(48, LOW);
    delay(200);
    digitalWrite(48, HIGH);

    // Initialize Serial
    Serial.begin(9600);

    // Set Pin-Mode for digital Pin
    pinMode(digitalPin, INPUT);
}
```

I

# Wiring – draw ()

```
// Loop

void loop() {

    // Trigger pulse
    digitalWrite(digitalPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(digitalPin, LOW);

    // Measure echo pulse
    int imp = pulseIn(digitalPin, HIGH);

    // Serial
    Serial.print(imp);
    Serial.print(" -> ");
    Serial.print(imp / 58);
    Serial.print(".");
    Serial.print(imp % 58 * 100 / 58);
    Serial.println("cm");

    // Wait
    delay(50);

}
```

## Zu beachten

- Messen eines High Impulses.
- Timeout von 1 Sekunde funktioniert nicht immer.
- Die Impulslänge muss im Fall des SRF05 durch 58 geteilt werden um ‚cm‘ zu erhalten.
- `pulseIn ()` ist in Verschiedenen Versionen von Wiring unterschiedlich implementiert! -> Falls ein Beispiel nach einer gewissen Zeit nicht mehr funktioniert, sollte man mit älteren Versionen von Wiring testen.

# Download

[www.embedded.arch.ethz.ch](http://www.embedded.arch.ethz.ch)

-> Examples -> Sensorboard Examples