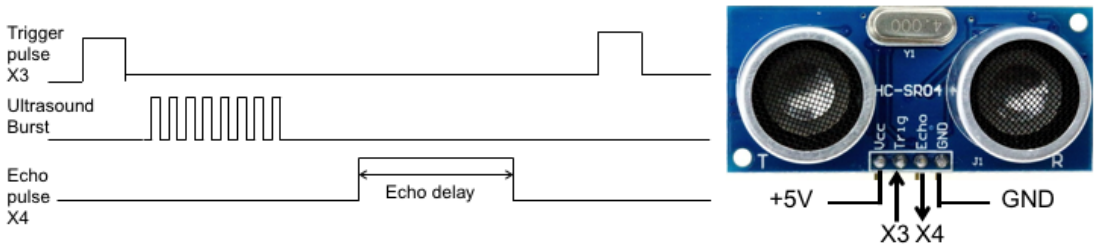
Ultrasound distance senor (1)

- Connect the ultrasound module as shown here.
- The sensor works as follows:
  1. Send a positive pulse on the Trigger input lasting at least 10μsec.
  2. The sensor will send out a burst of 8 pulses at 40kHz via the transducer.
  3. The second transducer will receive the echo signal, and produce an Echo output pulse. The width of the Echo pulse is the time it takes for echo to return to the receiving transducer.
- From this time delay, we can computer the distance as follows:

  Distance (in cm) = echo delay*0.5/speed of sound
- Create a file task5.py and enter the program shown on the next slide.
- Test the sensor by putting your hand in front of it and see how the distance measure changes.



PYKC 26 May 2016 EA1.3 - Electronics Lab 4 Slide 1

This another SENSE task. Now you want to measure distance of obstacle through echo location. The HC-SR04 module is a low cost device that uses ultrasound to detect distance – very much like what a bat does.

You need to send a trigger signal, a high pulse lasting for 10 μsec or longer. This will produce a burst of 8 ultrasound pulses at 40kHz. The second transducer will receive the return echo and the onboard electronics will provide an echo pulse signal. The width of the pulse is the echo delay.

Note that because this is a low-cost device, it may have errors. That is, the receiver some times does not produce the echo pulse at all. This is particular true if you have fast moving objects in front of the transducers.

Ultrasound distance senor (2)

```
import pyb
from pyb import Pin, Timer
print('Task 5: Test the Ultrasound distance sensor')

Trigger = Pin('X3', Pin.OUT_PP)
Echo = Pin('X4',Pin.IN)

# Create a microseconds counter.
micros = pyb.Timer(2, prescaler=83, period=0x3fffffff)
micros.counter(0)
start = 0               # timestamp at rising edge of echo
end = 0                 # timestamp at falling edge of echo

while True:
    # Send a 20usec pulse every 10ms
    Trigger.high()
    pyb.udelay(20)
    Trigger.low()

    # Wait until echo pulse goes from low to high
    while Echo.value() == 0:
        start = micros.counter()    # record start time of pulse

    # Wait until echo pulse goes from high to low
    while Echo.value() == 1:   # do nothing
        end = micros.counter()      # record end time of pulse

    # Calculate distance from delay duration
    distance = int(((end - start) / 2) / 29)
    print('Distance: ', distance, ' cm')
    pyb.delay(500)
```

- The clear new feature here is the use of timer 2 to count microseconds.
- The echo pulse may be disturbed if the obstacle moves quickly. This may hang the program.
- A better modification is to include some form of timeout in the inner while-loops. However, we are not going to border with that in this experiment.

PYKC 26 May 2016 EA1.3 - Electronics Lab 4 Slide 2

The program to measure the distance via this ultrasound transducer is quite long.

The code here is quite simple. The interesting part is where we program Timer_2 to provide a counter that counts microseconds. The value in the counter register accessed through the function micros.counter() stores the number of microseconds elapsed.