21M.370 Digital Instrument Design

Ultrasonic tutorial

Ultra sonic sensors are an easy way of detecting distance when working with microcontrollers. They are effective at sensing distance from a range of about 6 cm to 3 m.

They function on of the principle of sending out a ping and timing how long it takes for the ping to be reflected off the surface. Most ultrasonic sensors will have two pins, a trigger pin and an echo pin. The trigger pin is used to tell the sensor to send out a very short ultrasonic ping. The sensor also contains a microphone which is sensitive to the ultrasonic frequency, which enables the center to measure the time interval between when it sent out the pin and when the pin was reflected and received back at the microphone.

The sensor does not directly send the echo through the echo pin, rather the echo pin contains a pulse whose length is equivalent to the amount of time between sending and receiving the pin. The microcontroller measures the length of this pulse and returns that value in microseconds.

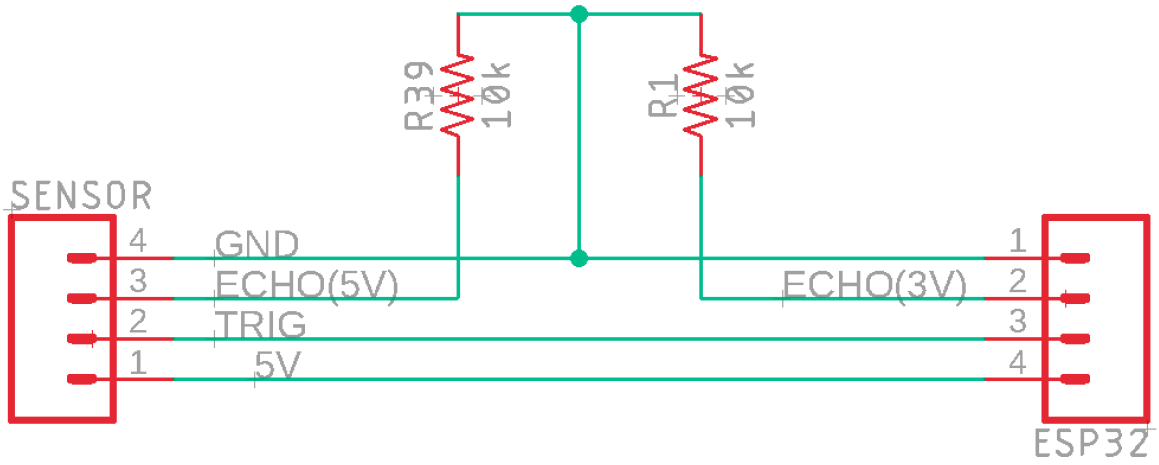
Unfortunately, the ultrasonic sensor that we have does not operate on 3.3v, which is the voltage that our system is running at. Instead, the sensor requires 5v. Luckily our board does have a 5v pin available, which is just the voltage that comes through the USB cable.

There are two things that we need to do in order to use this sensor with our board:

1. Power the sensor from 5v.

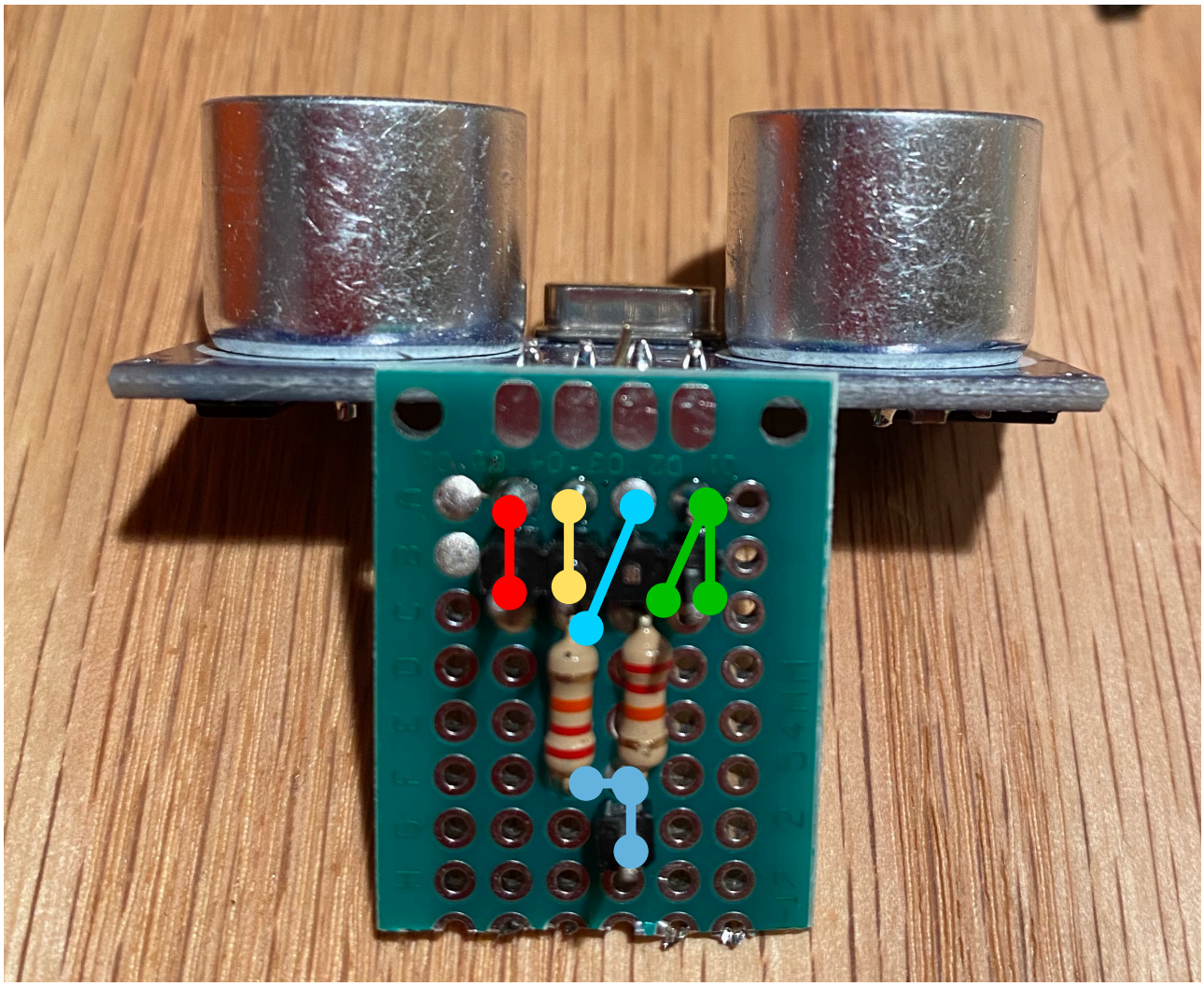
2. Use a voltage divider so that the signal coming from the Echo pen (which is a 5v signal) is reduced to a voltage lower than our board's voltage. Sending signals with voltages higher than the operating voltage of our board can damage the input pins, so we want to avoid it. - Any value of resistor will do for this voltage divider as long as they are both the same. This will create a 2.5v echo signal which our board can read.
3. The trigger input of the sensor is able to accept 3.3v signals sent from our board, so we don't need to do anything special for that.

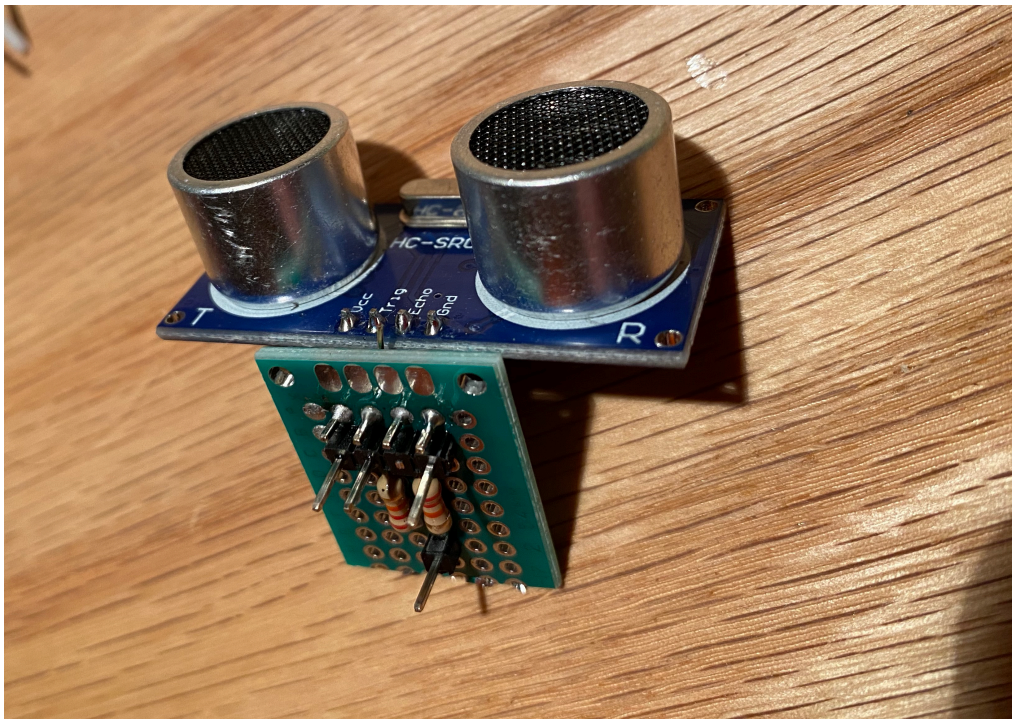
Here is a schematic of what we need to create:



Below is an example of the circuit put together on one of the prototyping boards that we have. The redline is 5 V, the yellow line is trigger, the light blue line is the 5v echo signal and the darker blue line is the 2.5v echo signal, and the green line is ground.

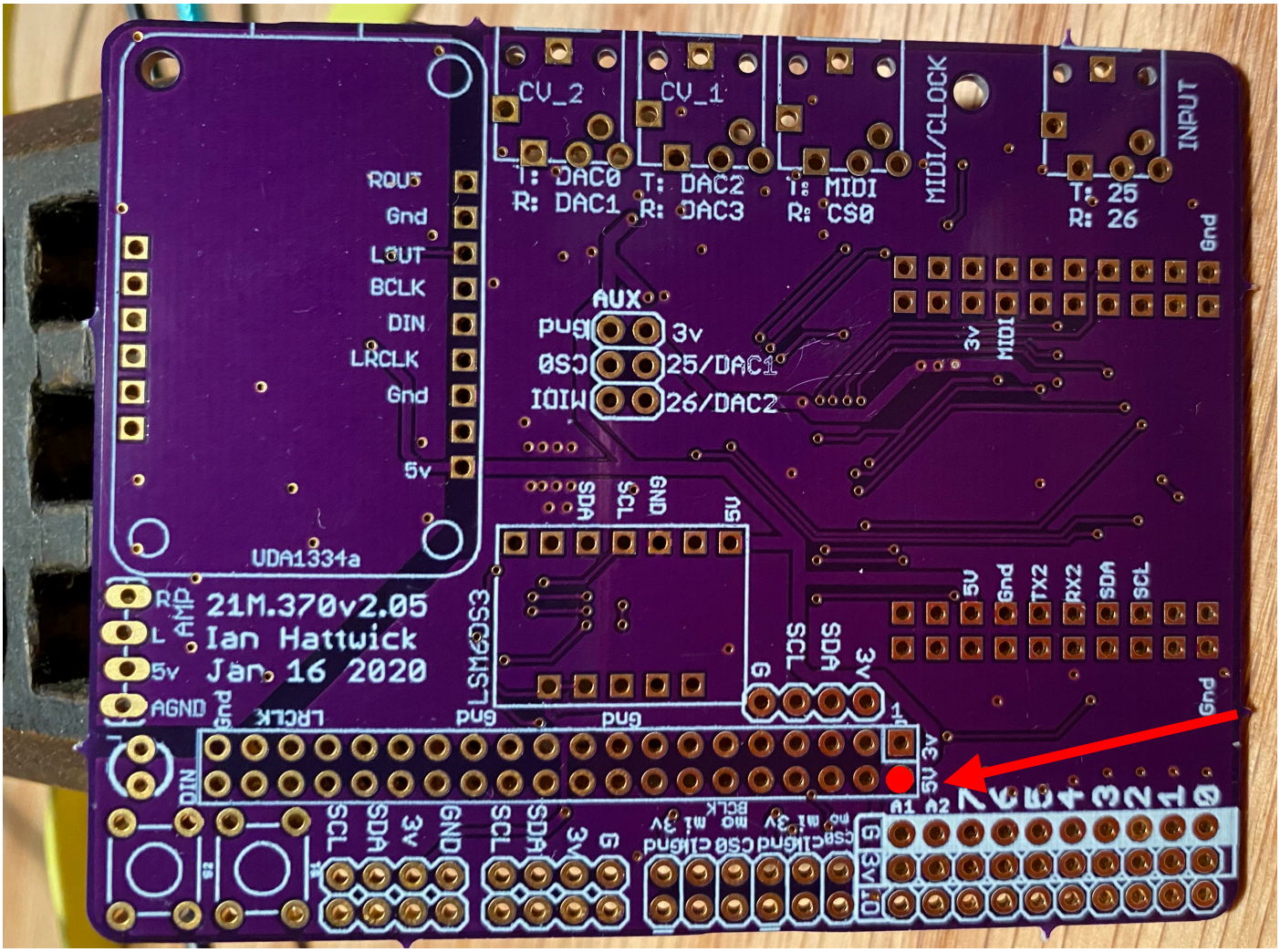
To make the circuit easier I added a header at the bottom of the PCB for the 2.5v header signal.




T
R
Vcc
Trig
Echo
Gnd

On the 370 PCB, here are the connections:

1. GND - any ground pin
2. Trigger and Echo - any analog GPIO pin
3. 5v - you'll need to solder an additional header here:



Software

ESP32

I added support for the ultrasonic sensor to our existing firmware. Now the most current firmware will always be called "M370_PRIMARY", and older

firmware versions will be in /NIME/ESP32/Dev.

Note that I ran into a problem where having the ESP32 was crashing when the MPR121 was enabled but not connected. For now, I commented out the MPR121 setup function - you can uncomment it if you want to use the MPR121.

Python

Example script: 370_ultrasonicExample.py
- method below works on any python script which includes a ‘mode’ parameter for OSC_ADDRESSES

For the python script you can enable the ultrasonic sensor just by setting the mode of two analog pins to be ‘TRIG” and “ECHO”:

```
OSC_ADDRESSES = {
    27:{ 'address':'/analog0', 'enable': 1, 'rate':100, 'mode':'TRIG' },
    33:{ 'address':'/analog1', 'enable': 1, 'rate':100, 'mode':'ECHO' },
```

Pins set to TRIG won’t send data to Python (since that pin is telling the ultrasonic sensor to send out a pulse). The ECHO pin returns the microseconds between sending the pulse and it being reflected back to the sensor.