NYU Tandon School of Engineering

RT Embedded Challenge 2018

Objective: This project involves navigating a vehicle through a minefield by using audio beacons of fixed frequencies located throughout the field. The vehicle will start at a fixed location in the field. The mission is to locate the "next" audio beacon and steer the vehicle toward that beacon while monitoring for the next beacon. The vehicle should continue to search each consecutive beacon until the final beacon is located, indicating the vehicle has exited the minefield. During the journey, the vehicle may not collide with any of the beacons and must steer clear.

You and your team consisting of 4 or 5 students will use the vehicle described below and a microphone sensor/amplifier to measure the audio signal from the beacons. You may use other approved sensors for collision avoidances well. Please clear any additional sensors with me or the TAs.

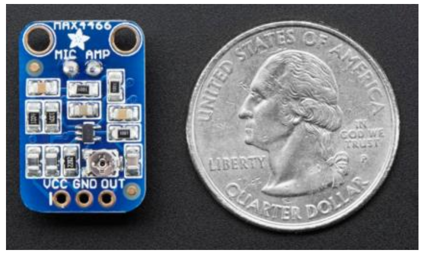
The details regarding the vehicle, motors, PWM drivers, and microphone/amplifier subsystem are provided below, however you are responsible for purchasing the parts. The total cost for all parts are approximately $40, or less than $10 per person. Web site references are provided for each part for purchasing, but you may purchase from a different vendor as well.

Vehicle Details: The vehicle consists of an aluminum chassis with three wheels (one front center wheel, two rear wheels). Each rear wheel has a dedicated DC motor used to motivate each wheel independently. Both DC motors are connected to a PWM motor driver unit that controls the RPM of each motor independently based on a 2 PWM signals. The details of the PWM motor driver are included in the appendix.

Microcontroller: Teensy 3.2 Real Time processor: This microcontroller will interface the PWM motor drivers on the vehicle and the microphone sensor (and any other sensor you may include). Plan to power the vehicle separately from the microcontroller. A four bank AA battery pack is sufficient for the vehicle, while a separate power source may be used for the controller. Do not power the motors through the microcontroller; it is not capable of providing sufficient current. Battery packs may be purchased from sparkfun.com or adafruit.com.

Sensor Details: The microphone/amplifier assembly is described below.

https://www.adafruit.com/product/1063



Description (from web): *Add an ear to your project with this well-designed electret microphone amplifier. This fully assembled and tested board comes with a 20-20KHz electret microphone soldered on. For the amplification, we use the Maxim MAX4466, an op-amp specifically designed for this delicate task! The amplifier has excellent power supply noise rejection, so this amplifier sounds really good and isn't nearly as noisy or scratchy as other mic amp breakouts we've tried!*

This breakout is best used for projects such as voice changers, audio recording/sampling, and audio-reactive projects that use FFT. On the back, we include a small trimmer pot to adjust the gain. You can set the gain from 25x to 125x. That's down to be about 200mVpp (for normal speaking volume about 6" away) which is good for attaching to something that expects 'line level' input without clipping, or up to about 1Vpp, ideal for reading from a microcontroller ADC. The output is rail-to-rail so if the sounds gets loud, the output can go up to 5Vpp!

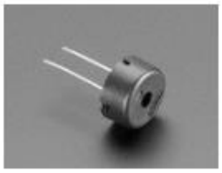
Using it is simple: connect GND to ground, VCC to 2.4-5VDC. For the best performance, use the "quietest" supply available (on an Arduino, this would be the 3.3V supply). The audio waveform will come out of the OUT pin. The output will have a DC bias of VCC/2 so when its perfectly quiet, the voltage will be a steady VCC/2 volts (it is DC coupled). If the audio equipment you're using requires AC coupled audio, place a 100uF capacitor between the output pin and the input of your device. If you're connecting to an audio amplifier that has differential inputs or includes decoupling capacitors, the 100uF cap is not required.

The output pin is not designed to drive speakers or anything but the smallest in-ear headphones - you'll need an audio amplifier (such as our 3.7W stereo amp) if you want to connect the amp directly to speakers. If you're connecting to a microcontroller pin, you don't need an amplifier or decoupling capacitor - connect the OUT pin directly to the microcontroller ADC pin.

For audio-reactive Arduino projects, we suggest using an FFT driver library (such as the one in this library) which can take the audio input and 'translate' it into frequencies. Also, check out this awesome Voice Changer project that uses this mic and an Adafruit Wave Shield!

Beacon Information: Each beacon is a piezo buzzer tuned to a particular frequency. Details of the buzzer are below:

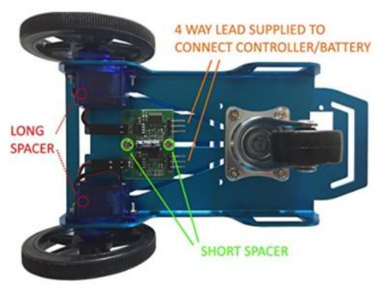
https://www.adafruit.com/product/160 (Datasheet available as well)



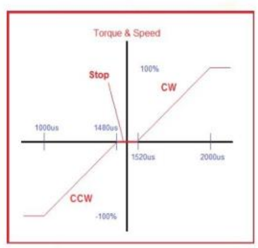


Description (from web): *Piezo buzzers are used for making beeps, tones and alerts. This one is petite but loud! Drive it with 3-30V peak-to-peak square wave. To use, connect one pin to ground (either one) and the other pin to a square wave out from a timer or microcontroller. For the loudest tones, stay around 4 KHz, but works quite well from 2KHz to 10KHz. For extra loudness, you can connect both pins to a microcontroller and swap which pin is high or low ('differential drive') for double the volume.*

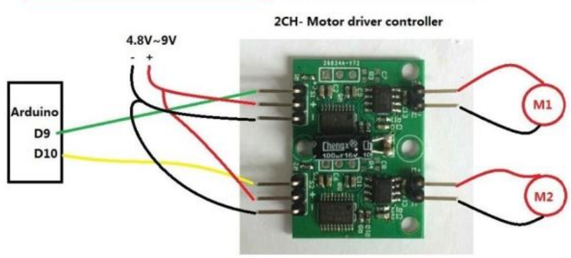
Vehicle Details and motor driver details:
https://www.amazon.com/gp/product/B06XZC2XDV/ref=oh_aui_detailpage_o02_s00?ie=UTF8&psc=1

FT-SMC-2CH MAKES MOTORS INTO CONT. SERVOS



2CH Digital Motor Driver
the speed and direction control by PWM signal
The PWM control protocol
(
Stop deadband -- 1500us(+ - 50us) 50Hz
CCW -- 1450us(LOW speed) ~ 1000us(MAX speed)
CW -- 1550us(LOW speed) ~ 2000us (MAX speed)
)



Rules:

1. You may use preassembled libraries for PWM signal generation and audio processing.
2. One group will be tested at a time. The vehicle will be placed at the starting point, and the beacons will be turned on. There should be a "Go" button on the vehicle to start the journey to the finish line as well as a visual indication (LED) that the journey has ended.
3. The time calculation will begin when the team pressed the "Go" button and will conclude when the visual indication of completion is visible. The time recorded for each team is marked after any part of the vehicle passes the first edge of the finish line.
4. Each team will have 2 tries to complete the task.
5. All source code must be submitted.

Grading:

This term project represents 40% of your final grade. The grading rubric for the project is as follows:

40% Performance (ranking/overall time)

20% Coding Efficiency

20% Repeatability/Consistency

20% Control smoothness