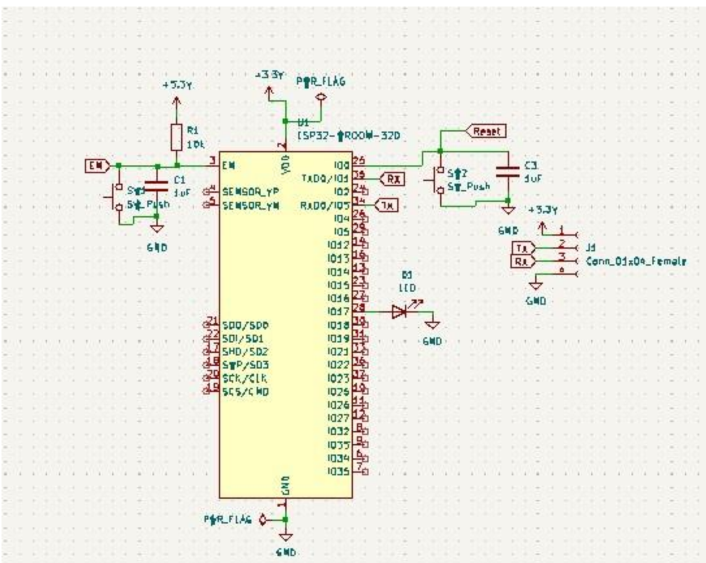
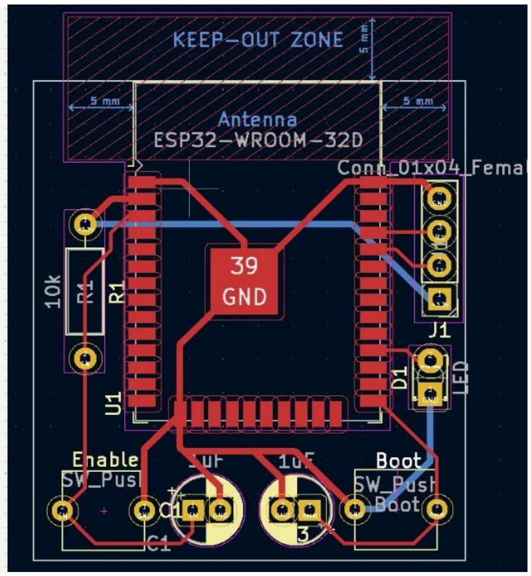
Vocalization recorder PCB for birds

The aim of this report is to display the design process of a PCB to be mounted on birds as a vocalization recorder.

The First Design

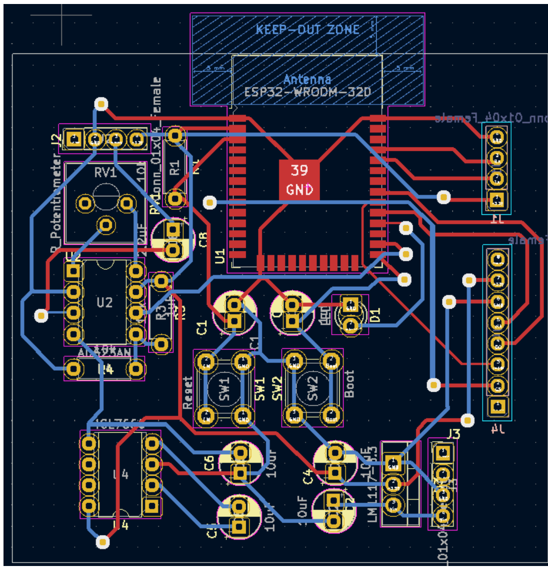
The first iteration of the design aims to validate the power up of the ESP32 and ensure the boot up sequence necessary to upload code onto the chip.

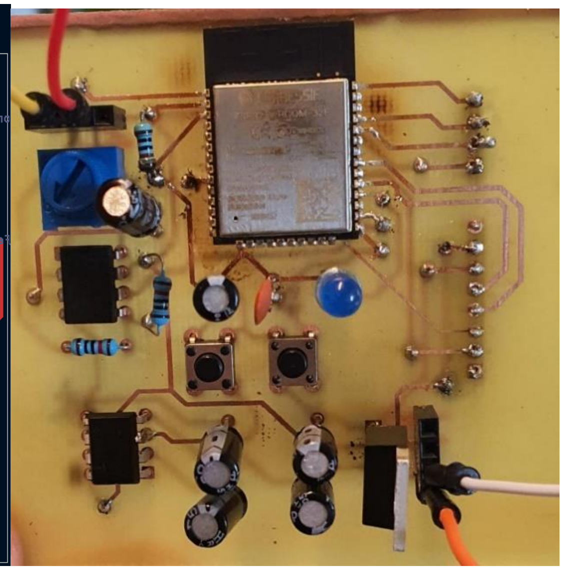


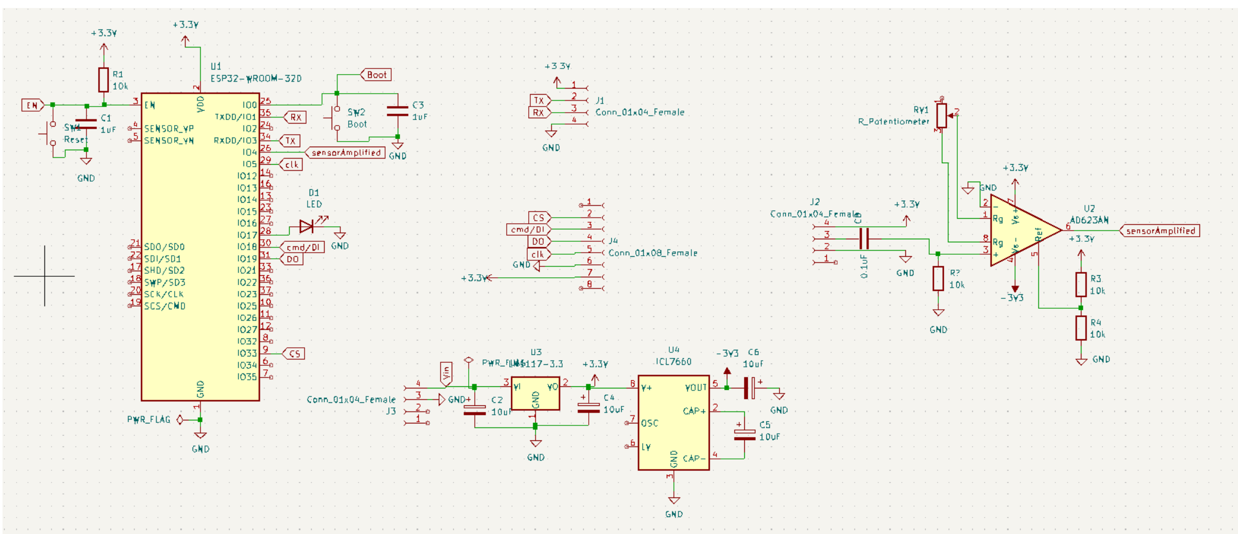


- The EN pin needs to be pulled high to turn on the ESP32. To restart the ESP32, a button was added to pull the EN pin to ground and a capacitor is added for debounce purposes.
- GPIO 0 is internally pulled high. A button connecting it to ground is sufficient to boot the code when uploading.
- The LED was used to confirm the functionality of the code.
- The External 4 pin connector is meant to connect to an FTDI connector (CP2102 chip) that converts the code sent from a USB serial connection to a UART serial connection with 2 main pins, the TX and RX pins. The TX pin of the CP2102 is connected to the RX pin of the ESP32 and the RX pin of the CP2102 is connected to the TX pin of the ESP32.
- The boot up sequence goes as follows:
    - Press the boot button on the esp32 (GPIO 0) and keep the button pressed.
    - Press the upload button on the computer to upload the code.
    - When the term “connecting ……” appears on the screen, press the enable button to reset and then release both buttons before connecting ends.
- Note the ESP32 was powered by the 3.3V output of the FTDI connection that’s connected to the laptop.
- To turn on the ESP32 after uploading the code, simply press the reset (EN) button.

The Second Design







The second design is a continuation of the first design adding the sensor connection, the amplifier, the power regulation circuit, and the SD card connector for an SD card port.

Amplifier Circuit: Based on previous research, the amplifier chosen is an instrumental amplifier (inamp) AD623AN due to its low noise capability.

- Potentiometer and Gain: The gain of this amplifier is determined by a single resistor connected at the terminals Rg-Rg. However in our situation, a potentiometer is used to adjust the gain according to the signal we are trying to amplify, taking into account the clipping bounds of the amplifier determined by V+ and V- . The voltages applied at V+ and V- affect the maximum gain the amplifier can provide. Initially V- was connected to ground, and the maximum gain was around 2, therefore a charge pump was used to provide a negative rail (-3.3V) for the inamp. Using this charge pump we were able to achieve larger gains up to a 100.

- High Pass Filter: A high pass filter was afterwards introduced using a 100nF capacitor and a 10Kohm resistor attenuating all frequencies below the cutoff frequency at 159Hz (1/(2pi x RC)).
- Vref Pin: The ESP32 ADC pin converts analog voltages from the scale 0 to 3.3V with nonlinearity clipping voltages at the boundaries. Therefore we can either add a dc bias to the input and that might cause common mode noise depending on the CMRR, or we can add the DC voltage required (3.3V/2 = 1.65V) to the Vref pin of the amplifier (which we opted to use).
- The Vref pin has an output impedance of 100Kohms. To be able to supply 1.65V , we need a low input impedance voltage source such as an opamp with unity gain however due to limiting space on the PCB, a voltage divider can work as long as the resistors are calculated properly. By using 2 10k ohm resistors, the circuit becomes equivalent to a 10k resistor in series with a 9k resistor (10k and 100k in parallel), and the corresponding Vref voltage based on this voltage divider is 1.57V. Note that the current dissipated using this method is 0.173mA.

Power Regulation: The ESP32 requires 3.3V to power up, therefore the LM1117 linear voltage regulator was chosen. Note that every linear voltage regulator has a dropout voltage that must be compensated from the voltage source. For the LM1117, the dropout voltage is around 1.2V, therefore the battery must have at least a voltage of 4.5V for normal operation of the regulator.

Charge Pump: The ICL7660 is used as a charge pump to provide the negative rail of the in-amp and requires 2 large electrolytic capacitors.

SD card Port: The SD card port was placed to initially store all the sensor data on the SD card before any WIFI implementation.

Wi-Fi Implementation

To remove the need of an SD card due to limited memory, the code was adjusted so that the data can be directly stored onto the computer. We setup 2 ESP32 devices, one of them acts as a transmitter and the other acts as a receiver using the ESP-Now communication protocol. The receiving ESP32 prints all the received data on the USB Serial monitor.

To be able to store the serial data, an open-source software was used called CoolTerm. CoolTerm records the serial monitor data and exports the data into a text file.

The PCB above uses one of the ADC2 pins to read the sensor data, however ADC2 is preoccupied by WIFI and therefore the sensor pin was changed to another pin of ADC1.

In order to setup the receiving ESP32, first run the script on the ESP32 to get its MAC address. Then run the script that prints on the terminal any receiving data.

To setup the transmitting ESP32, edit the code by placing the appropriate MAC address taken from the previous step and upload the code that transmits the data using ESP-Now.

Future Work and a Few Notes

- Make sure to adjust the cutoff frequency of the high pass filter to meet the needs of the signal if necessary.

- Understand the nature of the signal recorded and select an appropriate resistor for the gain of the in-amp without causing any clipping (can be done by adjusting the potentiometer and measuring the resistance across its terminals).
- The dimensions of the ICL7660 and its 2 electrolytic capacitors are large with respect to the PCB, other modern alternatives could be used (maybe trying the LM2776 that has 3 ceramic capacitors instead).
- Adjust the resistors of the voltage divider at the Vref pin, to center the signal at 1.65V and benefit from maximum gain without approaching the nonlinear boundaries of the ADC. To be on the safe side, make sure the amplified signal is between 0.3V and 3V rather than from 0 to 3.3V.
- Increase the sampling frequency of the code to avoid signal distortion.
- Select the appropriate battery voltage and capacity taking into account the dropout voltage of the linear voltage regulator and the current consumption of the device using Wi-Fi (which is around 150mA), a different approach would be to select another linear voltage regulator with a lower dropout voltage.
- Connect the sensor signal to a pin of ADC1 (such as SENSOR_VP) and remove the SD Card Port.
- When designing the final PCB to be manufactured abroad, take into account the manufacturing capabilities of the provider: minimum clearance, minimum via diameter, layer thickness, via to via clearance, minimum trace width, minimum trace spacing, minimum distance of trace to edge.
- The ESP32 used on the board is the ESP32-WROOM-32E. This chip has the crystal already built in, and no external pins are present for placing another 40MHz crystal. The only accessible pins for placing a crystal on the WROOM are meant for deep sleep mode and take a crystal value around 32KHz. The smaller chip that has all the pins exposed is referred to ESP32-D0WD-V3. (more details can be found in the datasheet).