UBER Hardware Challenge:

Test Plan and Measurements required for the various hardware components:

Hardware Component	Test Plan with Required Measurements
Microphone	- Frequency response - Sensitivity - Distortion - Signal-to-noise ratio S/N - Detection of audible imperfections - Directivity, polar plot
Microprocessor	- ADC readings from the microphone - Clock reset events - Power Consumption when GPS is ON and OFF - Current measurement during peak power modes - Voltage Rail Stability - Power-on Transients - Voltage on various Button/ On-Off switches. - Deep sleep current measurements
Power Supply	- Measure relative ripple (as low as possible) - Buck/Boost Voltage Outputs - Measure optimal efficient point - Synchronization of switching regulators to prevent intermodulation
Amplifier	- Measure SNR - Check if output signal is not being clamped to the voltage rails - Gain Bandwidth Product - Differential voltage accuracy(low as possible) - Measure power rejection ratio - Measure Output Noise at different frequencies
GPS	- Tx and Rx Tests - Sensitivity Measurements - Benchmarking tests - Calibration - Model Error sources and perform accuracy analysis through simulations -Data propagation
Antenna	- RF Transmit/Receive performance - Tx tone peak measurement in dBm across channels Rx sensitivity measurement in dBm across channel - Spurious emissions using Spectrum Analyzer - Oscillator frequency accuracy

	- S11 Measurements using Vector Network Analyzer with pigtails - For a given frequency measure S11(dB), Reflection coefficient, Mismatch(dB), Reflected power(%) - Impedance matching and simulation on ADS - Check SWR (Standing Wave Ratio)
Serial Communication	- Measure Baud Rate - Check CRC (Error checking) on packets - Clocks and data rates - Send UART packet and find bit failures using logic analyzer and oscilloscope - UART stress-test scripts to identify fail conditions and verify fixes - Correct I/O register configuration settings - JTAG, I2C, SPI, UART, USB communication protocols and measurements
Battery Module (Additional Hardware)	- Implement Battery Management System - Fail/Safe events for various temperature - Max Current Discharge - Fuel Gauge
Compass (Future Hardware Addition) - Provides a reference frame to measure the angle of attack of the events	- Within 5 V - I2C or SPI interface - Max degree of heading resolution

Software Block:

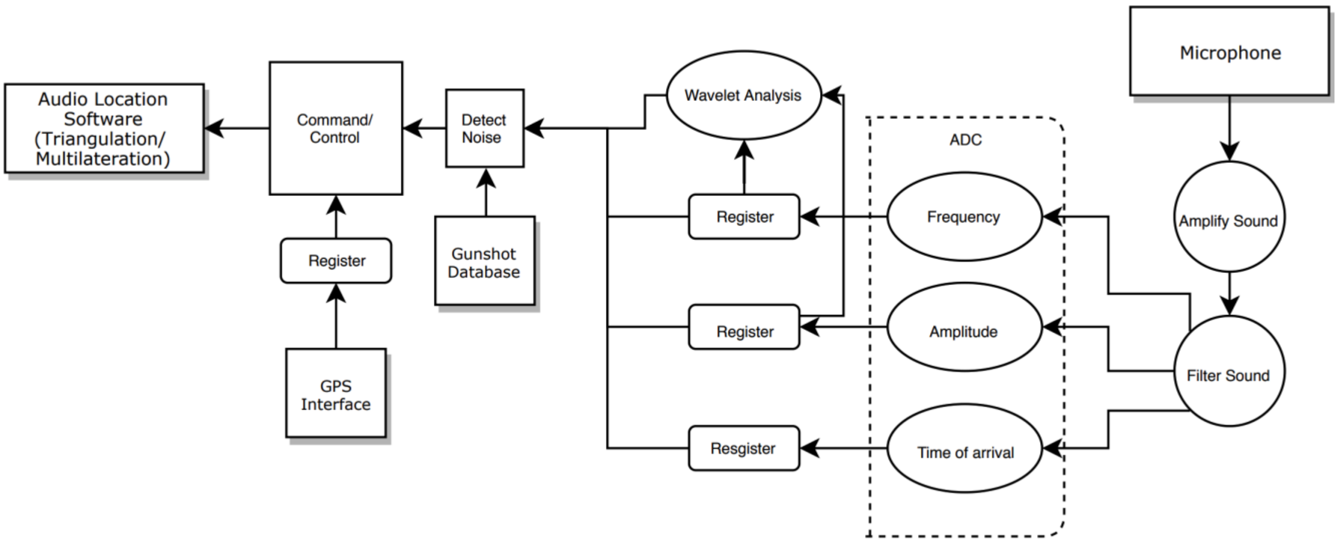

Figure 1: Software Block Diagram

High Level Software Overview:

Figure 1 above is a software flow diagram, showing the storage and movement of data for gunshot detection system. Initially, sound data is received by the microphone and is amplified. Next, the amplified sound is filtered to eliminate any sounds with a frequency outside of the frequency range of a normal gunshot. Functions will be called in the software that will extract the amplitude, frequency and time of arrival of the sound wave. Amplitude and frequency is then used to perform wavelet analysis and calculate normalized coefficients which will be used to detect whether the sound is a gunshot by comparing against coefficients of other gunshots from local database. If the sound is a gunshot, the time of arrival information along with the GPS coordinates will be sent to the command/control centre where a multilateration algorithm will be used to calculate the location of the sonic event.

Microcontroller Software:

The microcontroller will sample from the microphone and save the time and amplitude of the signal. When combined the saved signals and times will make a complete reconstruction of the wave. In reality the unit will save as many samples as possible. The microcontroller needs to sample at least twice the Nyquist frequency, which is the frequency of the source. So the microcontroller should save at least 6000 samples because the sound waves that are considered as events are in the 300 - 3000 Hz range. The Nyquist sampling theorem states a complete wave can be constructed by sampling at about twice the frequency of the source. The microcontroller should sample as fast as possible to produce the most accurate times of arrival and thus the most accurate GPS event coordinates.

Furthermore, the microcontroller will be able to determine the audio event type whether it is an explosion or gunshot. This is based on wavelets calculated from the sound wave form. Wavelet analysis will be performed by the mobile unit to determine the sound type since they are localized waves with their energy concentrated in time and space. Wavelets are used to determine the characteristic of the gunshot rather than Fourier analysis since wavelets are used for non-periodic waveforms, and are used ideally for representing sharp peaks. Discrete Wavelet Transform (DWT) will be implemented on the unit for fast processing. The DWT uses high pass and low pass filtering on the signal with down sampling used in the process. With this, half the frequencies of the signal are removed and similarly half the samples can be discarded as per the Nyquist rule. After this process there will be many signals, but each signal will match to a certain frequency range. This process will then be repeated multiple times, each time giving it a different level. This way, we only need to store the coefficients (frequency range samples) at each decomposition level. Hence, we are only storing half the coefficients in the unit (the rest is redundant). The wavelet coefficients is the energy of the signal in the time and frequency domain. Essentially, we are only keeping the central frequencies from each band which can be used to reconstruct the original time signal using deconvolution.

In order to find the different kind of gunshots, we need a local database on the microcontroller that stores the various normalized coefficients from Discrete Wavelet Transform from the various sounds of gunshots. Once the live gunshot event occurs, the coefficients are found from DWT and can be added to the database to compare the event gunshots with the database of stored coefficients to output the best match for the gun that was fired. Once the gunshot sound is detected and confirmed, the microcontroller will send the corresponding times of arrival and relay GPS coordinates to an outside source, likely to an nearby command/control station.

Command / Control Software:

The multilateration algorithm is the audio location algorithm that will find the location of the gunshot. This method requires using multiple microphones in an array like fashion to triangulate the location. Using the time difference of arrival of a sound wave between multiple microphones, the possible location of the sound source is found where the intersection of the microphones occur represented by their respective mathematical hyperbolic functions. Hence, the more mobile units deployed in the field, the more microphones will be used to find accurate location of the gunshot. Multilateration is triangulating signals based on time, rate and distance of the sound wave.