LAB1 Report

Pin-Jing, Li (111511015 *ouo.ee11@nycu.edu.tw)*
Jing-Kai, Huang
Duan-Kai, Wu

September 20, 2025

In lab 1 We explored the basic configuration of e^2 studio, ultrasound module and the basic signal processing flow of the wireless transmitted signal.

1 Hardware configuration

1 AIK-RA8D1 Development Board

The core MCU of this experiment, responsible for control, signal acquisition, and data processing. Provides ADC pins (e.g., AN121, AN000, AN001) for analog signal sampling and transfers data to PC via USB CDC. Supports interfaces such as I^2C and UART for peripheral communication.

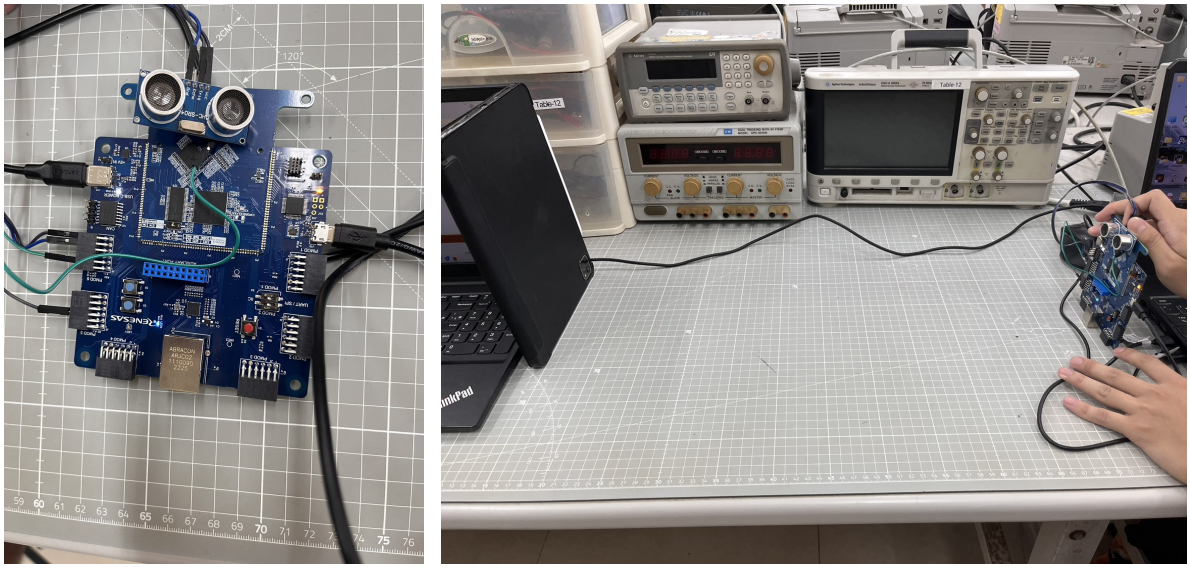
2 HC-SR04 Ultrasonic Module

Transmitter (Tx): Triggered by a high-level signal (<10 μs), it emits 8 pulses at 40 kHz. Receiver (Rx): Captures the reflected signal, which is sampled by the MCU's ADC.

Distance is calculated through time-of-flight (ToF) measurement, considering the speed of sound in air V=331+0.6t m/s.

3 DA14531 BLE Module

Provides Bluetooth Low Energy (BLE) communication capability. Enables UART $\rightleftarrows$ BLE data transmission between the MCU and mobile applications (e.g., GATTBrowser).



(a) sticking the cheap ultrasouond module onto the 10k board

(b) measurement experiments setups. We use our tablet as the reflection board, and the distance is referenced with the table pad.

Figure 1: hardware configurations

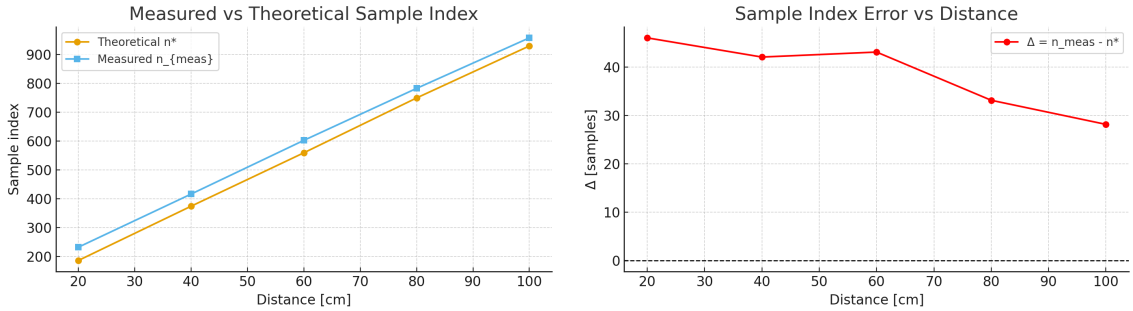
2 Error Source Analysis

We analyzed the measured peak indices across multiple distances and compared them with the theoretical sample indices n^* computed from the speed of sound at $T = 25°\text{C}$. This allowed us to examine the trend of the measurement error and construct a model for calibration.

Pair (cm)	n^*_{theory}	n_{meas}	$\Delta n^*_{\text{theory}}$	Δn_{meas}	$\frac{\Delta n_{\text{meas}}}{\Delta n^*_{\text{theory}}}$	Relative Error (%)
20→40	184.97→369.94	231→412	184.97	181	0.979	−2.1
40→60	369.94→554.91	412→598	184.97	186	1.006	+0.6
60→80	554.91→739.88	598→773	184.97	175	0.946	−5.4
80→100	739.88→924.86	773→953	184.97	180	0.973	−2.7

Table 1: Relative Distance Comparison (Measured vs. Theoretical)

The measured peak indices n_{meas} were aligned relative to the start of transmission, determined by the maximum slope of the Tx waveform. We then plotted n_{meas} against n^*, as well as the corresponding sample index error $\Delta n = n_{\text{meas}} - n^*$, as shown below.



(a) Measured peak index and theoretical n^* at $T = 25°\text{C}$

(b) Sample index error vs. distance

Figure 2: Comparison between measured peak indices and theoretical sample indices.

From Fig. 2b, we observe that the error Δn decreases approximately linearly as distance increases, consistent with a fixed system delay plus a small distance-dependent effect. Instead of the usual choice of a constant model, we found out that the error is related to the distance. We therefore model the error as

$$\Delta n = a + b\, n_{\text{meas}},$$

and use linear regression to estimate $a \simeq 53.14$ and $b \simeq -0.02469$. Applying this model, we compute corrected sample indices $\hat{n} = n_{\text{meas}} - \Delta n$, yielding the results summarized below.

True Distance (cm)	n_{meas}	$\hat{n}$ (Corrected)	n^* (Theory)	Error After Correction (cm)
20	231	183.56	184.97	−0.15
40	412	369.04	369.94	−0.10
60	598	559.39	554.91	+0.48
80	773	738.65	739.88	−0.13
100	953	922.23	924.86	−0.28

Table 2: Corrected Sample Index and Residual Distance Error (Linear Model)

The corrected indices significantly reduce the bias, with all residual distance errors falling within ±0.5 cm, demonstrating that the linear error model is effective.

To translate the measured sample index back into a physical distance, we use the relation between the sample index n, the sampling frequency F_s, and the speed of sound v. Each sample corresponds to a time increment of

$$T_s = \frac{1}{F_s},$$

so the round-trip time corresponding to n samples is

$$t = n\,T_s = \frac{n}{F_s}.$$

Assuming a two-way (Tx–Rx) propagation path, the one-way distance is then given by

$$\hat{d} = \frac{v\,t}{2} = \frac{v}{2F_s}\,n.$$

For our experiment at $T = 25°\text{C}$, we use $v = 331 + 0.6T \approx 346$ m/s $= 34{,}600$ cm/s and $F_s = 160$ kHz, which yields the conversion factor

$$k = \frac{v}{2F_s} = \frac{34{,}600}{2 \times 160{,}000} \simeq 0.1081 \ \frac{\text{cm}}{\text{sample}}.$$

Thus, the estimated distance from a measured sample index n_{meas} (or its corrected version $\hat{n}$) is

$$\hat{d} = k\,n.$$

This linear relation allows us to directly convert the corrected sample indices from Table 2 into absolute distance estimates and to compute residual errors in centimeters for performance evaluation.

3 Signal Processing

Our complete receiver design is summarized as follows.

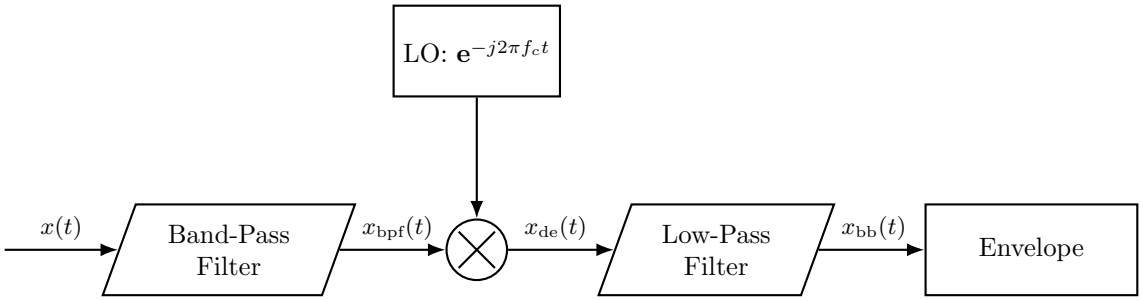


Figure 3: block diagram: signal → BPF → demod (mixer) → LPF → envelope.

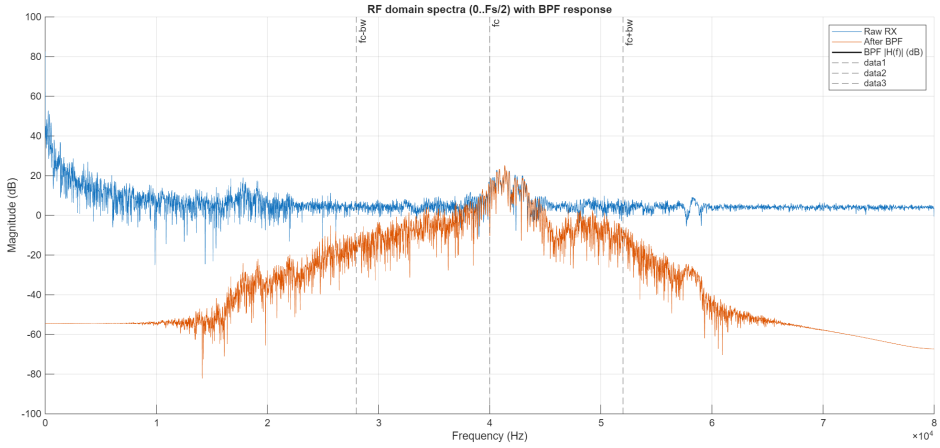


Figure 4: the Band-pass filter at the first stage. notice the FFT is in dB scale so it does not look exactly the same as what TAs have.

Bandpass Filtering

Given the sampled received signal $x(t)$, we first apply a bandpass filter to isolate the signal components near the carrier frequency. The bandpass-filtered signal is

$$x_{\text{bpf}}(t) = x(t) * h_{\text{bpf}}(t) \tag{1}$$

where $*$ denotes convolution and $h_{\text{bpf}}(t)$ is the impulse response of the bandpass filter.

Filter Design. We choose a 6^{th}-order IIR Butterworth bandpass filter with half-power frequencies

$$f_{\text{bp},1} = f_c - 2f_w, \qquad f_{\text{bp},2} = f_c + 2f_w,$$

where the carrier frequency $f_c = 40$ kHz and the signal bandwidth is

$$f_w = \frac{1}{T_{\text{burst}}} \approx 5 \text{ kHz}.$$

MATLAB Implementation. The filter is implemented and applied using MATLAB as follows:

```
bp_bw = max(2*fw, 12e3); % Bandwidth selection
bp_f1 = max(10, fc - bp_bw); % Lower cutoff frequency
bp_f2 = min(Fs/2-10, fc + bp_bw); % Upper cutoff frequency

dbp = designfilt('bandpassiir','FilterOrder',6, ...
    'HalfPowerFrequency1', bp_f1, ...
    'HalfPowerFrequency2', bp_f2, ...
    'SampleRate', Fs);

rx_bp = filtfilt(dbp, rx_dc); % Zero-phase filtering
```

This produces the zero-phase bandpass-filtered signal $x_{\text{bpf}}(t)$.

Demodulation

After bandpass filtering, we demodulate the signal by multiplying it with a complex exponential at the carrier frequency f_c:

$$x_{\text{de}}(t) = x_{\text{bpf}}(t) \cdot e^{-j2\pi f_c t} \tag{2}$$

In the frequency domain, this shifts the bandpass signal to baseband, centering its spectrum around 0 Hz.

```
w0 = 2*pi*fc/Fs; % Normalized carrier frequency (rad/sample)
lo = exp(-1j*w0*n); % Complex exponential for downconversion
bb = rx_bp .* lo; % Complex baseband signal
```

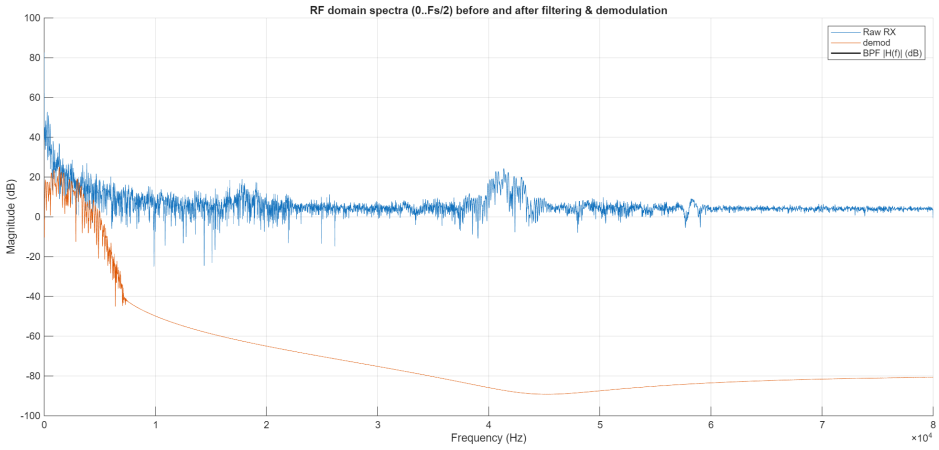


Figure 5: the final Demodulated signal after all the filtering

—

Low-Pass Filtering

The downconverted signal still contains high-frequency components due to the product term. We pass $x_{\text{de}}(t)$ through a low-pass filter to retain only the baseband component.

Filter Design. We use a FIR low-pass filter with passband edge at $f_{\mathrm{LP}} = f_{\mathrm{c}}$ and stopband at $1.6 f_{\mathrm{LP}}$:

```
% FIR: passband up to lp_fc, stopband starts at 1.6*lp_fc
dlp = designfilt('lowpassfir', ...
    'PassbandFrequency', lp_fc, ...
    'StopbandFrequency', 1.6*lp_fc, ...
    'PassbandRipple', 0.1, ...
    'StopbandAttenuation', 70, ...
    'SampleRate', Fs);

bb_f = filtfilt(dlp, real(bb)) + 1j*filtfilt(dlp, imag(bb));
```

This yields the complex baseband signal $x_{\mathrm{bb}}(t)$ with high-frequency components removed.

Envelope Detection

Since the original signal is carried by f_c, it can be expressed as

$$x(t) = A\cos(2\pi f_c t + \phi).$$

$$\mathrm{Re}\{x_{\mathrm{de}}(t)\} = x(t)\cos(2\pi f_c t) = \frac{A}{2}\big[\cos(\phi) + \cos(4\pi f_c t + \phi)\big].$$

Thus, the envelope amplitude is halved. We compensate for this loss by multiplying by

```
env = 2 * abs(bb_f);
```

This produces the envelope of the baseband signal, scaled back to its original amplitude.

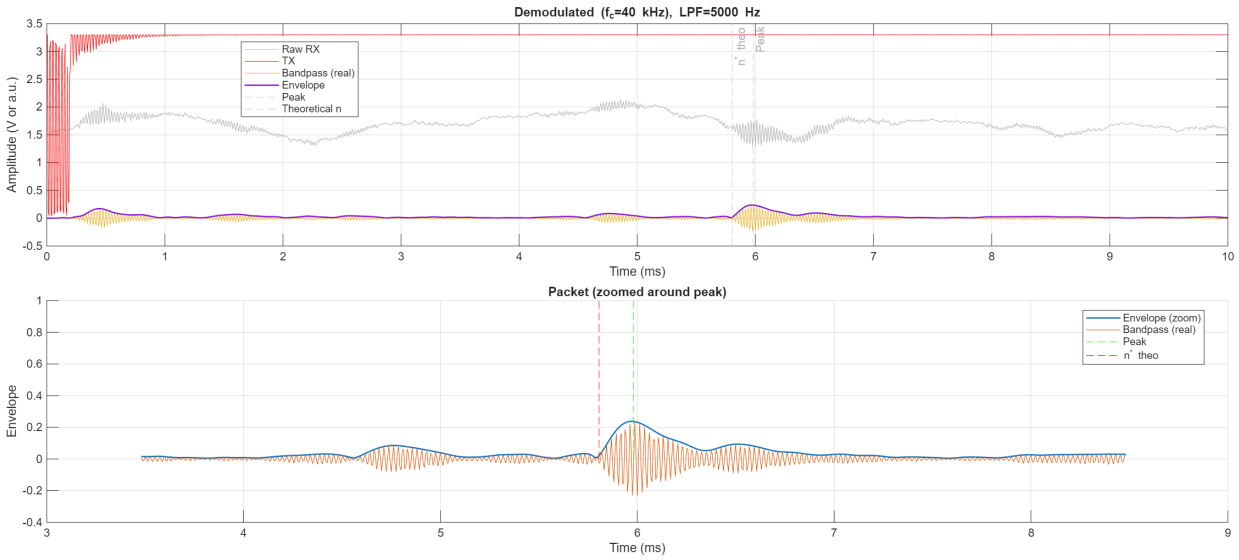


Figure 6: Time domain signal before and after filtering & demodulation

References

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.