

```
// src/codec/textEncoder.ts
```

```
import {  
  getWavelengthForLetter,  
  getLetterForWavelength,  
  ALPHABET_MAP,  
} from "../protocol/wavelengthMap";
```

```
/**  
 * Normalize text:  
 * - Uppercase  
 * - Filter to A–Z only (non-letters are dropped in v1.0)  
 */  
export function encodeTextToLetters(text: string): string[] {  
  const normalized = text.toUpperCase();  
  const letters: string[] = [];  
  for (let i = 0; i < normalized.length; i++) {  
    const ch = normalized[i];  
    if (ch >= "A" && ch <= "Z") {  
      letters.push(ch);  
    }  
    // Future versions: handle spaces, punctuation via special  
    codes.  
  }  
  return letters;  
}  
  
/**  
 * Convert letters to wavelength sequence (nm).  
 * Letters that cannot be mapped are skipped.
```

```
*/  
export function encodeLettersToWavelengths(letters:  
string[]): number[] {  
  const wavelengths: number[] = [];  
  for (const letter of letters) {  
    const wl = getWavelengthForLetter(letter);  
    if (typeof wl === "number") {  
      wavelengths.push(wl);  
    }  
  }  
  return wavelengths;  
}
```

```
/**  
 * Convenience: directly encode text into wavelength  
sequence.  
*/  
export function encodeTextToWavelengthSequence(text:  
string): number[] {  
  const letters = encodeTextToLetters(text);  
  return encodeLettersToWavelengths(letters);  
}
```

```
/**  
 * Decoding utilities  
*/
```

```
/**  
 * Convert a sequence of wavelengths back into letters via  
nearest-neighbour mapping.
```

```
*/  
export function decodeWavelengthsToLetters(wavelengths:  
number[]): string[] {  
    return wavelengths  
        .map((wl) => getLetterForWavelength(wl))  
        .filter((l): l is string => !!l);  
}  
  
/**  
 * Convert letters to a reconstructed text string.  
 * Currently simply joins letters without spaces.  
 */  
export function decodeLettersToText(letters: string[]): string {  
    return letters.join("");  
}  
  
/**  
 * Helper to expose the alphabet mapping for UI purposes.  
 */  
export function getAlphabetMap() {  
    return ALPHABET_MAP;  
}
```