# Assignment 5 Test Plans

## Test plan 1: Morse Decoder

- **Name**: Morse Decoder unit test
- **Rationale**: In Morse Code Decoder Project, the ability to interpret Morse code is the key component of this application. Therefore, it is necessary to test and ensure the correctness of the decode algorithm.
- **Instruction**: All unit testing methods can be called by command: *npm test*
- **Test case**:
    - Input: **SL_LSS&LLL&LLS_**
      (Note: Besides the standard Morse code, the **_** symbol stands for the gap between words, and & symbol stands for the gap between letters)
    - Expected output: A DOG
    - Test passes.

## Test plan 2: Morse Translator

- **Name**: Morse Translator unit test
- **Rationale:** In this project, Morse code is simulated by PIR Motion Sensor signals. Hence, our application needs the capability of receiving signals from the motion sensor and transmit them as Morse signals (dot or dash). This unit testing method will make sure our application can perform Morse code simulating successfully.
- **Instruction:** All unit testing methods can be called by command: *npm test*
- **Test case 1: Dot**
    - Input: 2 signals separated by 2s interval, first signal is motionstart, second signal is motionend.
    - Expected output: dot symbol “S”
    - Test passes.
- **Test case 2: Dash**

    - Input: 2 signals separated by 6s interval, first signal is motionstart, second signal is motionend.
    - Expected output: dot symbol “L”
    - Test passes.
- **Test case 3: Letter gap**
    - Input: 2 signals separated by 6s interval, first signal is motionend, second signal is motionstart.
    - Expected output: dot symbol “&”
    - Test passes.
- **Test case 4: Word gap**
    - Input: 2 signals separated by 2s interval, first signal is motionend, second signal is motionstart.
    - Expected output: dot symbol “_”
    - Test passes.