| **Library 2 Test Plan**  Prepared by Connlaoi Fruit, Daniel Covaci |
| --- |
| 1. **Introduction**  A plan describing the processes and methods of testing for the Pull up/down function for the Library 2 Components Project for 4235 Embedded Systems II. It includes the resources used, scope, approaches, results, and issues discovered over the testing period. |
| 1. **Testing Resources**  Connlaoi Fruit and Daniel Covaci were the testers for these functions as designers of the E4235\_gpio\_pud function in Group 7.   Items needed: A RPi 4 with ARM8 Assembly |
| 1. **Scope of Testing** *In-Scope:*   The E4235\_gpio\_pud function  *Out-of-Scope:*  The E4235\_GPIOPWM and E4235\_MultiGPIO functions |
| 1. **Testing Approaches** *RPi4 Ping Method:*  The main method used was what we termed the “Ping” method. The command ‘raspi-gpio get’ outputs the current status of the Pi’s gpio pins and their pull up/down status. The method involves using this command to determine the current status of the pull up/down resistor, using the function to change it, and then using it to confirm whether or not it was changed. |
| 1. **Test Results**   E4235\_gpio\_pud was tested using the method described above, for a test GPIO of GPIO 24. GPIO 24 is initially in the DOWN PUD state and if the command E4235\_gpio\_pud(24, 1) is run, will become the UP PUD state, while all other GPIO states remain constant, indicating a successful test. |
| 1. **Risks & Issues**   The function is called in the form E4235\_gpio\_pud(int x, int y) where x is the desired GPIO #, limited to the external GPIOs 2 - 27 and y is the desired pull up/down limited to 1 (Pullup) or 2 (Pulldown). For values of x < 2, x > 27, y < 1, y > 2, an error message will print. |