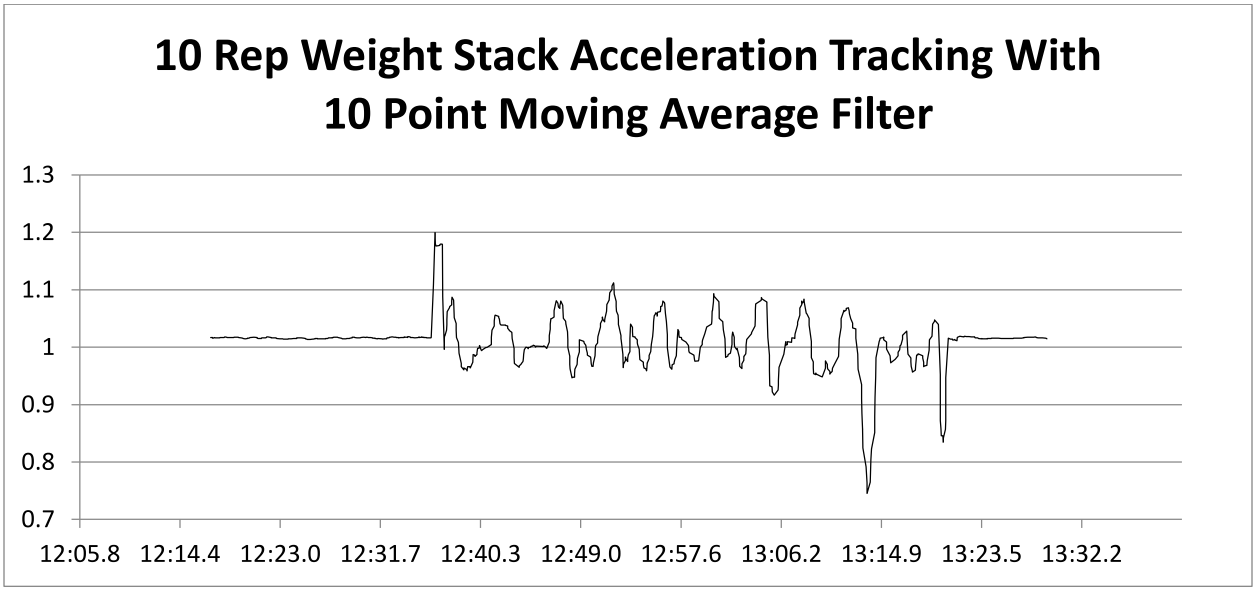
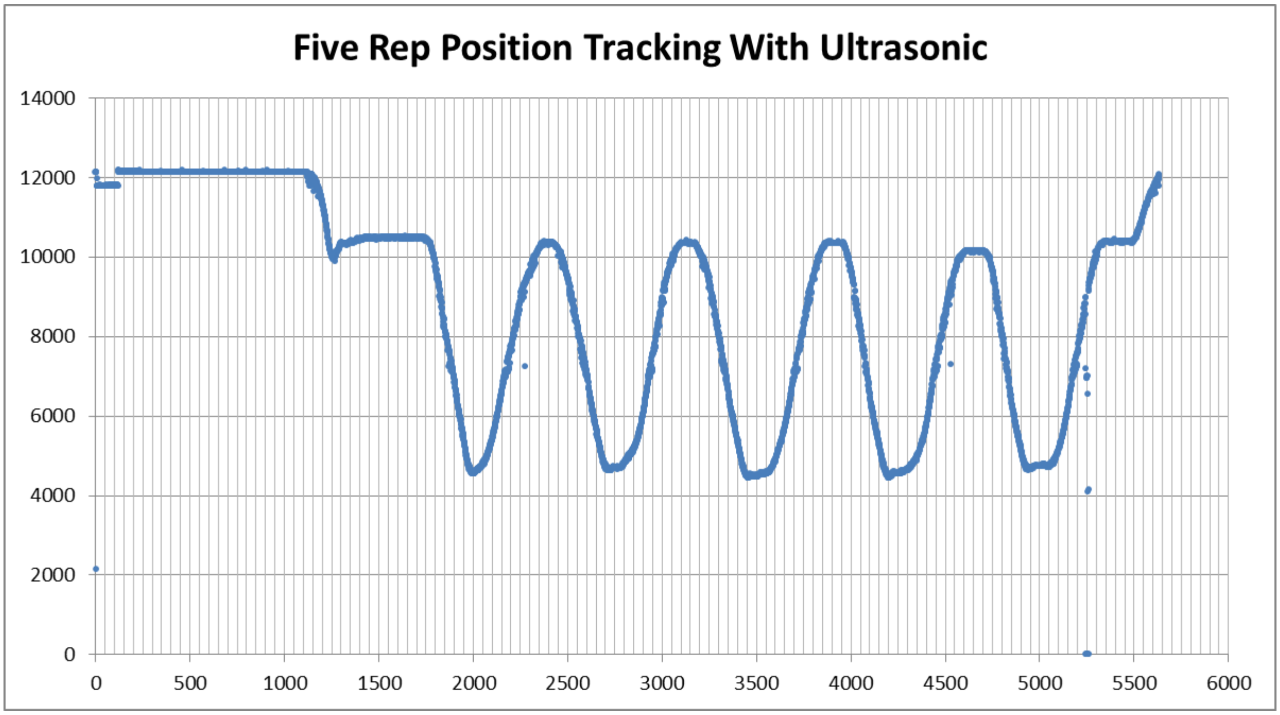
Section E: Appendices

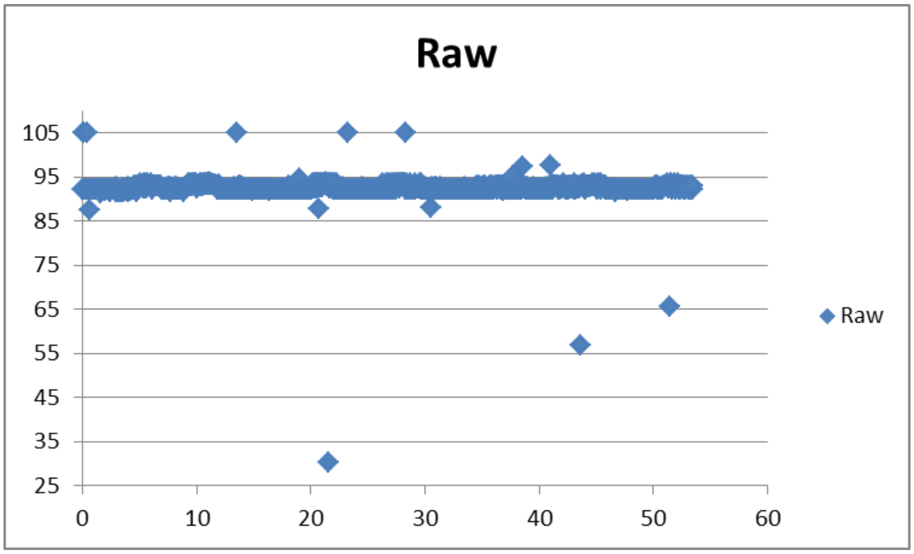
Weight Stack Acceleration From iPhone Accelerometer

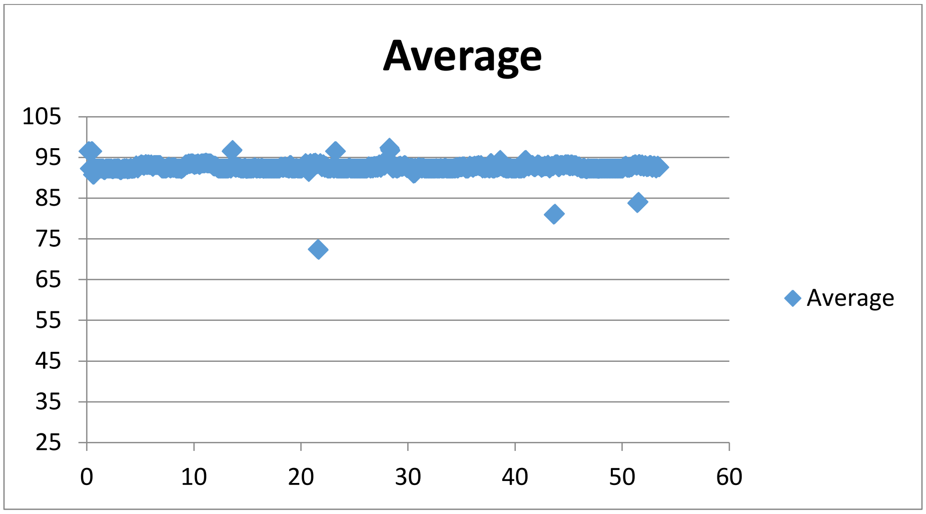


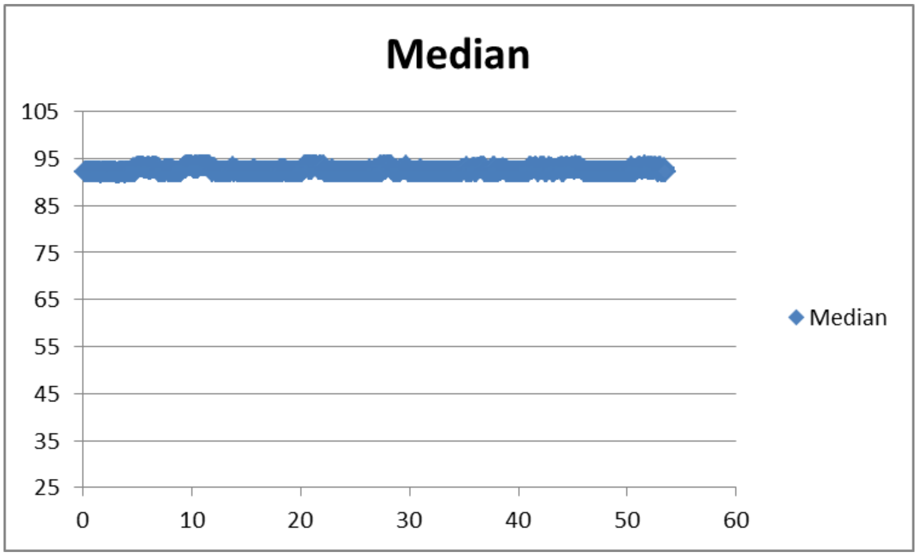
Experimental Tracking of Weight Stack Position with Ultrasonic Sensor



HC-SR04 Distance Data Filter Results







Resistance Calculations for Weight Stack Resistors and LightBlue Bean

Weight #	Bounds	Target	Percentage	Target(V)	R2	Total Resistance	Current (mA)	Resistor Used	Battery Voltage	3
	1024								R1	555
1										
	969									
2		943	92.09%	2.7627	6461	7016	0.4	6800	2.509803922	
	918									
3		892	87.11%	2.6133	3750	4305	0.7	3900	3.509803922	
	867									
4		841	82.13%	2.4639	2551	3106	1.0	2700	4.509803922	
	816									
5		790	77.15%	2.3145	1874	2429	1.2		5.509803922	
	765									
6		739	72.17%	2.1650	1439	1994	1.5		6.509803922	
	714									
7		688	67.19%	2.0156	1136	1691	1.8		7.509803922	
	663									
8		637	62.21%	1.8662	914	1469	2.0		8.509803922	
	612									
9		586	57.23%	1.7168	743	1298	2.3		9.509803922	
	561									
10		535	52.25%	1.5674	607	1162	2.6		10.50980392	
	510									
11		484	47.27%	1.4180	497	1052	2.9		11.50980392	
	459									
12		433	42.29%	1.2686	407	962	3.1		12.50980392	
	408									
13		382	37.30%	1.1191	330	885	3.4		13.50980392	
	357									
14		331	32.32%	0.9697	265	820	3.7		14.50980392	
	306									
15		280	27.34%	0.8203	209	764	3.9		15.50980392	
	255									
16		229	22.36%	0.6709	160	715	4.2		16.50980392	
	204									
17		178	17.38%	0.5215	117	672	4.5		17.50980392	
	153									
18		127	12.40%	0.3721	79	634	4.7		18.50980392	
	102									
19		76	7.42%	0.2227	44	599	5.0	47	19.50980392	
	51									
20		28	2.73%	0.0820	16	571	5.3	15	20.45098039	
	0									

Standard Use Test Case

Step	Action	Result	Notes
1	Select 20 lbs on the weight stack and plug audio jack into appropriate box.	When bar is pulled down 20 lbs is lifted up	
2	Use "User 87" Card to log into machine	Log in LED changes from red to green. Distance Measurements begin. LED strip changes to red (BottomOfRep).	
3	Pull bar down to quarter rep postion	LED strip changes to blue and gets brighter as the rep progresses. MovingUp state is active.	
4	Pull bar down to half rep postion	LED strip changes to green to indicate TopOfRep state. Weight (20lbs) is retrieved from GymtronBean. numReps incremented for active set (set 0).	
5	Allow bar to return to starting postion (one rep completed)	LED strip changes from green to blue and gets dimmer. Once at starting postion LED strip changes to red (BottomOfRep).	
6	Pull bar down to half rep postion	LED strip changes from red to blue and gets brighter as the weight stack moves upwards. Once at half rep, LED strip changes to green (TopOfRep). numReps for current set is	
7	Allow bar to return to quarter rep postion, and return to half rep postion without returning to starting postion.	LED strip changes from green to blue and gets dimmer as quarter rep postion is reached. LED strip gets brighter as half rep position is achieved but it does not turn green when half rep position is reached. numReps for current set is NOT incremented.	
8	Return bar to starting postion and complete a full rep.	LED strip changes from red to blue to green. numReps is incremented for current set. LED strip changes from green to blue to red.	
9	Complete step 8 three more times	Set 0 should have 5 reps at 20 lbs	

10	Press the Set Complete button	Logged in LED (green) on box flashes a minimum of three times. The current set is incremented to set	
11	Change the selected weight to 30 lbs and complete 3 full reps.	Set 1 records 3 reps at 30lbs.	
12	Wait MAX_TIME_BETWEEN_REPS ms after the top of the last rep has been completed, then complete another full rep.	The set should automatically timeout. Set 1 should remain at 3 reps and Set 2 should have 1 rep at 30lbs.	
13	Press the logout button.	Log in LED changes from green to red. IDLE state becomes active.	
14	Log in to website and confirm User 87's most recent workout matches the workout just completed in this test case: set 0: 5 reps 20 lbs set 1: 3 reps at 30 lbs set2: 1 rep at 30 lbs	User's most recent workout on website matches expected data.	

Website Navigation Test Case

Step	Action	Result	Notes
1	Navigate to gymtron webpage via internet browser	User is presented with a form prompting login credentials	
2	Enter User 187 in login form and press enter	User is redirected to login page as credentials are invalid	
3	Enter User 87 in login form and press enter	User credentials are valid, and user is redirected to the workouts page	
4	User is directed to the workouts page	All previous workouts ar visible and sorted in descending order	
5	Navigate to the 'choose workout' form and enter a workout numbered 1 higher than the greatest workout viewable and submit	User is redirected to the details page and 0 results are displayed	
6	Select the 'return to workouts' link	User is redirected to the workouts page	
7	Navigate to the 'choose workout' form and enter the highest workout number and submit	User is redirected to the details page and all sets in the most recent workout are displayed	
8	Select the 'logout' link	User is logged out and redirected to the login page	

Auto Logout Test case

Step	Action	Result	Notes
1	Swipe card to log into machine	Login LED changes from red to green.	
2	Wait for longer than AUTO_LOGOUT_TIME milliseconds without completeing a rep (leave machine stationary)	After AUTO_LOGOUT_TIME milliseconds Login LED changes from green to red and user is automatically logged out.	

GymtronBean Communication Unavailable Test Case

Step	Action	Result	Notes
1	Remove the battery from the GymtronBean	GymtronBean cannot communicate with Raspberry Pi	
2	Run Gymtron executable file on Raspberry Pi	Program starts, then hangs for ~40 seconds while communication to bean is attempted. After this time message is printed indicating that the temperature and weight are unavailable.	
3	After ~40 second delay log into machine using NFC card	Workout session starts and executes as per the standard use case with with two exceptions: 1) Weight is set to -1 and is not updated for each new set 2) DEFAULT_TEMPERATURE is used instead of an actual value measured by GymtronBean. Communication with Gymtron bean will not be attempted again until the gymtron executable is restarted.	
4	Stop the gymtron executable and reinsert the battery into the GymtronBean. Restart the gymtron executable.	The executable starts and runs as per the standard use case.	
5	Remove the battery from the GymtronBean to simulate battery failure	Workout session continues as per the standard use case with with two exceptions: 1) Weight is set to -1 and is not updated for each new set 2) DEFAULT_TEMPERATURE is used instead of an actual value measured by GymtronBean. Communication with GymtronBean will not be attempted again until executable is restarted	

Set Length Timing Test Case

Step	Action	Result	Notes
This test requires two people. One person to perform the workout the other to time each set in the workout.			
1	Log into machine using NFC Card	Log in is successsful as per standard use case	
2	Complete 1 set of 5 reps using any weight. At the top of the first rep the stop watch operator should start the time. After the last rep the set complete button should be pressed. The stop watch operator shall stop the time at the top of the last rep.	Record the time taken for the first set in seconds.	
3	Complete a second set with 10 reps using any weight following the same procedure as setp 2.	Record the time taken for the second set in seconds.	
4	Navigate to the appropriate user profile on the website and verify set durations.	The webiste set durations should match the recorded results or be within reassonable error given the stopwatch timing procedure.	

```c
// To compile this simple example:
// $ gcc -o quick_start_example1 quick_start_example1.c -lnfc

#include <stdlib.h>
#include <nfc/nfc.h>

static void
print_hex(const uint8_t *pbtData, const size_t szBytes)
{
  size_t  szPos;

  for (szPos = 0; szPos < szBytes; szPos++) {
    printf("%02x  ", pbtData[szPos]);
  }
  printf("\n");
}

int
main(int argc, const char *argv[])
{
  nfc_device *pnd;
  nfc_target nt;

  // Allocate only a pointer to nfc_context
  nfc_context *context;

  // Initialize libnfc and set the nfc_context
  nfc_init(&context);
  if (context == NULL) {
    printf("Unable to init libnfc (malloc)\n");
    exit(EXIT_FAILURE);
  }

  // Display libnfc version
  const char *acLibnfcVersion = nfc_version();
  (void)argc;
  printf("%s uses libnfc %s\n", argv[0], acLibnfcVersion);

  // Open, using the first available NFC device which can be in order of selection:
  //   - default device specified using environment variable or
  //   - first specified device in libnfc.conf (/etc/nfc) or
  //   - first specified device in device-configuration directory (/etc/nfc/devices.d) or
  //   - first auto-detected (if feature is not disabled in libnfc.conf) device
  pnd = nfc_open(context, NULL);

  if (pnd == NULL) {
    printf("ERROR: %s\n", "Unable to open NFC device.");
    exit(EXIT_FAILURE);
  }
  // Set opened NFC device to initiator mode
  if (nfc_initiator_init(pnd) < 0) {
    nfc_perror(pnd, "nfc_initiator_init");
    exit(EXIT_FAILURE);
  }

  printf("NFC reader: %s opened\n", nfc_device_get_name(pnd));

  // Poll for a ISO14443A (MIFARE) tag
  const nfc_modulation nmMifare = {
    .nmt = NMT_ISO14443A,
    .nbr = NBR_106,
  };
  if (nfc_initiator_select_passive_target(pnd, nmMifare, NULL, 0, &nt) > 0) {
    printf("The following (NFC) ISO14443A tag was found:\n");
    printf("    ATQA (SENS_RES): ");
    print_hex(nt.nti.nai.abtAtqa, 2);
    printf("       UID (NFCID%c): ", (nt.nti.nai.abtUid[0] == 0x08 ? '3' : '1'));
    print_hex(nt.nti.nai.abtUid, nt.nti.nai.szUidLen);
    printf("      SAK (SEL_RES): ");
    print_hex(&nt.nti.nai.btSak, 1);
    if (nt.nti.nai.szAtsLen) {
      printf("          ATS (ATR): ");
      print_hex(nt.nti.nai.abtAts, nt.nti.nai.szAtsLen);
    }
  }
  // Close NFC device
  nfc_close(pnd);
  // Release the context
  nfc_exit(context);
  exit(EXIT_SUCCESS);
}
```