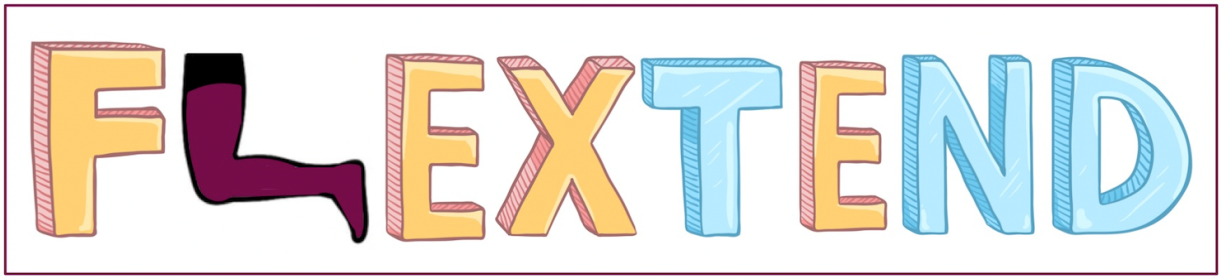
Department of Electrical & Computer Engineering

Boston University

EC 464 Senior Design Project II

Second Prototype Test Report



Written By
Team 24: Flextend

Team Members
Thomas Scrivanich
Carmen Hurtado
Sohaib Ansari
Jack Halberian
Murat Sencan

1.0 Required Materials

1.1 Hardware

1. Two ADXL345 Digital Accelerometer
2. Arduino Nano 33 IoT
3. 9V Battery
4. 200 Ohm Resistor
5. 10k Ohm Resistors
6. 1 Switch (on/off)
7. 2 Buttons

1.2 Software

1. Flextend Application
    a. iOS operation (iPhone simulator/iPhone)
    b. Android operation (Android emulator/Android)
2. Google Firebase

2.0 Set Up

2.1 The setup for the testing is in two parts: Arduino board with two Gyroscope sensors for hardware and React Native mobile application, both Android and iOS, for software. The HW and SW are now connected via Bluetooth Low Energy (BLE), and part of the test will show this functionality. The mobile device is connected to a computer via usb cable and the application will be run through there. We will be testing login and registration authentication with *Google Firebase*. In the login screen we will be testing that as a phone number is entered the login is successful (navigates to home screen) and that the phone number gets stored in Google Firebase. After navigation to the home screen, we will navigate to the previous result screen where any values for the login user that is stored in the database will be displayed on screen (flexion and extension). We will also be testing the User

Profile, where the user can track their progress and set calendar events, and so on. Next step will be to test the BLE functionality that it's implemented using *react-native-ble-plx* module. To do this we will show that measurement readings for the Gyroscopes display properly on the "Begin Measuring" page. On the Arduino side, we have the two Gyroscopes connected to calibrate and measure the degrees of extension and flexion. This data is sent to the Nano and then to the mobile application.

Circuit Pinout Connection	
1602 LCD Display Module	**Arduino Nano 33 IoT**
RS	D12
E (Enable)	D11
DB4	D5
DB5	D4
DB6	D3
DB7	D2
R/W	GND
VSS	GND
VCC	+3V
LED+	+3V (Through 200 Ohm R)
LED-	GND
V0	+3V and GND (Through 10K Ohm Potentiometer)

Circuit Pinout Connection	
ADXL345 Digital Accelerometer 1	**Arduino Nano 33 IoT**
GND	GND
Vin	+3V
SDA	SDA (A4)
SCL	SCL (A5)
SDO	+3V
CS	+3V

Circuit Pinout Connection	
ADXL345 Digital Accelerometer 2	**Arduino Nano 33 IoT**
GND	GND
Vin	+3V
SDA	SDA (A4)
SCL	SCL (A5)
SDO	GND
CS	+3V

3.0 Pre-testing Procedure

3.1 Hardware

1. Ensure the 9V battery is connected to power in the breadboard and switch is on.
2. Make sure Nano 33 IoT lights on.
3. Upload measurement + BLE code to Arduino (if not done already)

4. Once the mobile application is started, connect to Arduino and press the “calibrate” button to zero the extension and flexion readings.
5. Press the “measurement” button to begin sending measurements to the app.

3.2 Software

1. Plug in the iOS or Android device to the host computer (if testing iOS this must be a Mac, otherwise we will use a Windows PC)
2. Start the app on the host computer and begin testing on the mobile device.
3. Login to the Firebase project on computer and open the database.

4.0 Testing Procedure

4.1 Arduino and Electronics

1. Place Nano breadboard onto measuring paper
2. Calibrate both gyroscopes on a flat surface.
3. Press the measure button to start measuring.
4. Tilt breadboard to angle on paper and observe accuracy
5. Confirm that the Arduino is efficiently outputting measured rotation data in degrees to the computer or mobile app.
6. Repeat steps for each angle and calibration modes.

4.2 Mobile Application

1. User registration: enter user information and confirm, redirects to Login Screen
2. On the login screen enter a valid phone number of an unregistered user with the correct format (+1 234 567 8910).
3. This action will fail as all users need to register before accessing the application’s home screen.
4. Enter a valid phone number of the user we previously registered. This action will succeed after authentication of the confirmation code and the user is redirected to the Home screen.

5. Verify that the registered phone number is stored in Firebase database (Through computer screen inside Firebase project)
6. Upon entering the home screen, verify that the Arduino has successfully auto-connected to the application via the debugging console.
7. Start the device.
8. Navigate to the Measuring Screen. Start measuring and recording data from the device. Having a protractor in front of the device confirms that the values in the screen are accurate. Values should be changing as the protractor moves to show live measurement functionality.
9. Navigate to the Previous Result screen and verify that it shows information for the logged user.
10. Confirm that it is the same information stored in the Firebase database.
11. Navigate to Profile Screen. Users can change profile photos. Demonstrate this.
12. Pop up modal for Body Metrics. Users can enter their information and it is passed to another screen.
13. Pop up modal for Reminders will have two options: add a new event and edit the event.
14. Confirm that these both work by creating an event, then go to the phone's calendar and make sure it is there.
15. Edit this event from the app and check that the changes are reflected in the phone's calendar.
16. Navigate to the Progress screen and see past measurements for the logged in user.
17. Confirm that these values are correct by looking at the Firestore DB.
18. Navigate to Guide view and confirm functionality.

5.0 Measurable Criteria

5.1 Hardware

1. The gyroscopes successfully capture movement of the circuit board in degrees.
2. The Arduino accurately transfers measurements in real time through bluetooth.
3. Calibration of sensors work effectively.

4. Measuring button consistently measures for 20 seconds.
5. Buzzer gives sound feedback on the measuring and calibration process.

5.2 Software

1. The user is able to Register to the application with their phone number and name.
2. The user is able to log into the application by providing their phone number as well as the country code.
3. Google Firebase records any phone number used to sign and register into the application.
4. The user navigates the application successfully from any given screen.
5. The hardware device can successfully connect to the application and send data over this channel.
6. The user can see their individual past measurement on the Past Results Screen. Google Firebase Firestore database contains all specific user metrics for knee flexion and extension with timestamps. The application successfully gets data and pulls data from Firestore.
7. Users can successfully select and change their profile photo.
8. Users can successfully add and edit calendar events for reminders.
9. Users can see their past progress in the form of a graph with legible and organized values.

6.0 Score Sheet

6.1 Hardware Power Switch

Device Power On Via Switch (Y/N)
Y

6.2 Measurement Test

Buzzer? (Y/N)	Measuring? (Y/N)	20 Seconds? (Y/N)
Y	Y	Y

6.3 Hardware Score Sheet (Two Gyroscope)

Tests	Theoretical Value (Degrees)	Experimental Value Displayed on Serial Monitor (Degrees)	Within $\pm$ 3°? (Y/N)
Test #1	0° (1)	2°	Y
Test #2	30° (1)	31°	Y
Test #3	60° (1)	62°	Y
Test #4	30° difference (2)	31° difference	Y
Test #5	60° difference (2)	64° difference	N

6.4 Calibration Test (Calibrated at 30°)

Tests	**Degree Placed At**	**Theoretical Value (Degrees)**	**Experimental Value Displayed on Serial Monitor (Degrees)**	**Within ± 3°? (Y/N)**
Test #1	30°	0°	3°	Y
Test #2	0°	-30°	-28°	Y
Test #2	60°	30°	30°	Y

6.5 Software Score Sheet (Registration Process)

Phone Number	**Display Name**	**Stored in Google Firebase? (Y/N)**	**Successful Login? (Y/N)**	**Successful Registration? (Y/N)**
+1 1111111111	Thomas Scrivanich	Y	Y	N
+1 2012801587	Thomas Scrivanich	Y	Y	N
+1 7862738812	null	N	N	Y
+1 1111111113	null	N	N	Y

6.6 Software Score Sheet (Navigation)

Navigation Errors? (Y/N)
N

6.7 Software Score Sheet (Bluetooth)

Successful Connection to App? (Y/N)	Data
Y	Degree value displayed on ‘Live Measurement’ Screen

6.8 Software Score Sheet (Previous Results/Progress)

Phone Number	Previous Results Match Google Firebase (Y/N)	Progress Results Match Google Firebase (Y/N)
+1 1111111111	Y	Y
+1 2012801587	Y	Y

6.9 Software Score Sheet (Calendar Reminder Event)

Event Created (Y/N)	Event Shows on Calendar (Y/N)	Event Can be Edited (Y/N)
Y	Y	Y

7.0 Testing Results

7.1 Hardware

The main components tested were the accuracy of angle measurement, ease of use, and the device design in general. With the circuit set up and sensors not attached to the acrylic frame of the design, the angle measurement is accurate with a +- 2 degree difference. After prototype testing day, we did some tests when the sensors were placed on the frame on a user's leg. This first design prototype showed some problems that are critical to our project:

1. The acrylic frame must be positioned in the correct way for the angle measurement to be accurate.
2. The velcros help keep the frame tight to the user's leg but at the same time suppress mobility.
3. Overall not comfortable for the user to wear.

The accuracy of measurement when the device is on a user's leg is in the range of +- 5 to 10 degrees. Moreover, in order for a user to use this design assistance is most likely needed.

With these points in mind we are shifting our design. We will try to implement a design that is either sturdier to keep the frame in the correct positioning even when user movement happens, or a design that relies heavily on mathematical operations to handle any movements or user error as measurements are being taken.

The last important point on the hardware that needs to be reimagined is calibration of the sensors. We did not take into account that a user could have limited mobility and their leg could not fully extend to achieve calibration. Our design will shift to find a simple calibration algorithm that can trigger during any position of a user's leg.

7.2 Software

On the software side, our team tested new components as well as existing components that were refined after the 1st Prototype Testing session. Our application's registration process, data handling, and calendar were new components tested in the 2nd Prototype Testing session.

The application navigation and bluetooth connectivity were existing features that the software team improved upon earlier in the semester.

The registration process passed all testing parameters. We tested user phone numbers that were recognized by Google Firebase and numbers that were not recognized by the system. The two phone numbers that were recognized by Google Firebase could be used to log in to the application from our login screen. However, these numbers could not be used to register since they are already associated with a user account. Any attempt to register would generate an alert for the user. The two phone numbers not recognized by Google Firebase could not be used to log into the application from the login screen. An alert would generate for the user directing them to the registration page. These numbers, however, can be used to register and create a new user account.

In addition to the registration process, our team also implemented two methods of showing results to the user using test data. These methods include:

1. Previous Results - Results of the most recent measurement; displayed via progress rings that show how close the flexion and extension values were to the best possible value (flexion - 135 degrees, extension - 0 degrees)
2. Progress over Time - Multiple entries showing the user their flexion and extension progress over time; displayed via bar graphs with time as the x-axis and flexion/extension value as the y-axis. Each entry has a timestamp in Google Firebase

The data handling tests passed all our our team's parameters. The graphs successfully displayed results stored in Google Firebase Firestore.

Finally, the last new component tested was the calendar. Our software team implemented calendar events allowing the user to setup reminders to use the Flextend device. The calendar event can be created through the application by navigating to the user profile. The user can see the newly created event by navigating to their mobile device's calendar app. In addition, the user can edit the calendar event. An issue was discovered during the 2nd Prototype Test session where the calendar GUI on Android made it very challenging to navigate and perform the event

operations. Going forward, our team plans to find a solution to improving the calendar experience on Android.

The application navigation was improved since the 1st Prototype testing session. Our team found no errors in regards to navigation. The bluetooth connection saw major improvements. The user can automatically connect to the Flextend device just by navigating to the live measurement screen. The user can calibrate the device by clicking a button in the live measurement screen and receive data from the device.