לאחר שתכנתנו את החיישנים של Arduino לשדר לנו נתונים בטווח שיהיה אמין ויעיל לצורך ניתוח הנתונים, את נתוני החיישנים נקלוט באמצעות הbluetooth של הטלפון ובעזרת התכנה. התכנה מתבססת על open-source שמפורסם ב github ונקרא BlueTerm , ויודע לקלוט את נתוני הbluetooth ולהציג אותם בterminal. לקחתי את התצוגה ודאגתי שנתוני החיישנים יעברו דרך מערך חיישנים שיוצגו במקומם המתאים על המסך לפי מפת החיישנים שיצרתי על התצוגה שיצרתי.

1. מאתחלים רשימה של 3 חיישנים, שכל אחד מהם הוא מסוג TextView , שאותו נשלח אחר כך לתצוגה של המסך ועליו נשים את הנתונים המגיעים מהחיישנים עצמם. מספר החיישנים (כרגע) הוא סטטי, נקבע רגע לפני יצירת הרשימה עצמה של החיישנים, וכך קובע את גודלה. (בהמשך מספר החיישנים יקבע בתוך האפליקציה, או לחלופין ע"י בדיקה של מספר החיישנים הקיימים שהתכנה תצליח לזהות, עדיין לא יודע אם זה אפשרי)
2. המתודה onCreate היא בעצם המתודה שמאתחלת את האפליקציה ואת כל מה שצריך להיות מאותחל כדי שהיא תוכל לרוץ (החל מקריאה לפונקציית האב שבעצם מאתחלת את האפליקציה שתוכל לרוץ, דרך הlogs , כל ההעדפות והמשאבים שנחוצים לקביעת הגדרות של האפליקציה, וידוא של תקשורת הנתונים עם הbluetooth , וקביעת התצוגה (layout) הנוכחית.
3. בתוך מתודה זו אנו מקשרים את מערך החיישנים לתצוגה שלהם, ע"י מציאת כל חיישן לפי הID שלו

\*בחישוב ניתוח (parsing) של המידע של החיישנים ניסיתי בהתחלה לצבור "שורה" של data מהחיישנים ורק אז לחשב את הממוצע. משהו בחישוב הסתבך בקריאת הנתונים ולכן כשלב ביניים לקבל נתונים כלשהם - חיברתי כל פעם את הנתון הקיים ואת הנתון המתקבל וחילקתי ב2. כך קיבלתי מעין "ממוצע זמני" של הנתונים, כדי שנוכל לראות נתונים. אחר כך אנסה לצבור יותר נתונים ולראות איך אני מחשב אותם יותר במדויק.

\*\* לאחר ניסיון\בדיקה קצרה – עדכון התצוגה של ערך החיישן לא קורה בכל קריאה, אלא רק אחת לכמה קריאות, ולכן יוצא שהנתון יהיה יותר קרוב לממוצע מאחר וזה בעצם קורה רק לאחר קריאות מרובות ולא תוצאה רק של 2 קריאות. בנוסף – הוספתי בדיקה שלא תאפשר הכנסת ערכים נמוכים מידי וכן בדיקה שאם עדיין אין ערכים לא תתבצע החלוקה ב2, אלא רק לאחר שהכנסנו ערך נבצע חלוקה ל2 הערכים וכן על זו הדרך.

**TODO** – בשלב הבא – חישוב ע"פ נתוני החיישנים מהו **אחוז הלחץ** בכל חיישן