**סטאטוס שבועי**

שבוע 2:

כחלק מתהליך בחירת סביבת העבודה וההכנה שלה, בדקנו מספר גורמים - מערכת ההפעלה, שפות תכנות רלוונטיות ומערכת לניהול גרסאות.

1. מערכת ההפעלה - התקנו Linux, Ubuntu בשלב א׳ כמכונה וירטואלית על מנת ללמוד את מערכת ההפעלה ולראות האם באמת קיימים יתרונות על פני עבודה בסביבת OS המבוססת גם על Unix.
2. שפות תכנות - החלטנו לאמץ את ההחלטה של תום ואביב, לעשות את העיבוד המידע הראשוני ע״י Python בשל הנוחות היחסית של קריאת packets  המועברים בפרוטוקול UDP ע״י שפה זו. כשאת תהליך זיהוי עצמים, ניתוח המידע והצגתו נעשה בסביבת MatLab. כחלק מהחלטה זו השוואה בין מספר סביבות עבודה של Python, בסוף בחרנו להשתמש גם ב- Spyder בשל האורינטציה המדעית של סביבה זו. לבסוף מצאנו מקור למידה מקצועי ל- Python  קורס דרך אתר  [linda.com](http://linda.com) שמסופק באדיבות אינטל .
3. מערכת לניהול גרסאות ושמירת גיבויים - בדקנו מה זה Git והאם יש לנו צורך בכך, החלטנו לנסות להשתמש במערכת זו לניהול גרסאות בפרויקט. נתקין אותה שבוע כבר מחר (שני 6/11).
4. ביצענו השוואה בין חיישני מרחק שונים כדי להבין יותר לעומק את נתוני החיישן איתו נעבוד לעומת מוצרים אחרים שקיימים בשוק. סיכום השוואה זו, וכן הסבר קצר על אופן מדידת המרחק (TOF) נמצאים בתיקייה "ספר פרויקט".

מטרתנו השבוע היא להתמקד בלמידה יותר מעמיקה של קוד הפייתון, ע"מ שנוכל בקרוב מאוד כבר לחלץ נתונים מהחיישן בכוחות עצמנו.

שבוע 3:

1. כתיבת דפי מידע מפורטים אודות המידע הקיים בחיישן ה- vlp16 איתו נעבוד. דפי המידע יכילו פירוט אודות כל הנתונים אותם אנו מתכוונים להפיק ולעבד מתוך הפלט של החיישן וכמו כן הסברים על האופן בו נעבד ונשתמש בהם.

בסופו של תהליך הכוונה שדפי מידע אלו יצורפו אל ספר הפרויקט.

1. המשך עבודה על החיישן כאשר המטרה העיקרית היא לקבל תצוגה ויזואלית בדמות קבצי טקסט של הפלט של החיישן. נכון לרגע זה, אנו מנסים להבין את הבעייתיות שנתקלנו בה בקוד הפייתון שאחראי על פענוח המידע מהחיישן. יצירת תקשורת מול החיישן בפרוטוקול UDP בוצעה.
2. נפגשנו עם תום (מהפרויקט הקודם שבוצע עם החיישן הספציפי במעבדה) על מנת להבין ממנו את דרכי הפעולה הכדאיים עם החיישן לפחות בתחילת הדרך, על מנת לחסוך זמן וטעויות מיותרות.