לימוד של פרוטוקול ע"י האזנה אליו ב-WireShark,   
וכתיבת תוכנת לקוח חדשה

שם הסטודנט:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , ת"ז:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

בכור בדימונה פועלות זוג תוכנות שרת לקוח. תוכנת השרת מותקנת במחשב החיישנים של הכור, תוכנת הלקוח מותקנת במחשב עמדת השליטה של הכור. בעזרת תוכנת הלקוח הטכנאי בעמדת השליטה של הכור בודק את ערכיהם של החיישנים השונים.

ממשק המשתמש הישן של תוכנת הלקוח, שיושבת במחשב בעמדת השליטה, מאפשרת לקרוא את החיישנים, אך רק עפ"י דרישה, ובכל פעם חיישן אחד.

כדי להתאים את הכור לדרישות הבטיחות המודרניות, נדרש לכתוב תוכנת לקוח חדשה למחשב השליטה שתדבר עם תוכנת השרת שבמחשב החיישנים, תקבל קריאות מהחיישנים, ותציג אותם ברצף על המסך, למפעילי הכור שבעמדת השליטה. (יש לכתוב תוכנת לקוח חדשה בלבד ולא תוכנת שרת)

זוג התוכנות לא עודכנו מאז הקמת הכור. בגלל הזמן הרב שעבר אין שום תיעוד או את קוד התוכניות. הדרך היחידה כרגע כדיי לשחזר את הפרוטוקול בין מחשב השליטה למחשב החיישנים היא בעזרת לכידת התקשורת ביניהם.

במעבדה זו נלכוד את התקשורת בין תוכנת הלקוח לשרת ונלמד את הפרוטוקול. בעזרת הפרוטוקול שנלמד נכתוב תוכנת לקוח חדשה בהתאם לדרישות החדשות.

1. הפעלת תוכנת השרת והלקוח  
   התוכנה: DimonaServer.exe היא תוכנת השרת שרצה על מחשב החיישנים.  
   התוכנה: DimonaClient.exe היא תוכנת הלקוח שרצה על מחשב השליטה.  
   (בהפעלה ב-VDI בהפעלה הראשונה יקפוץ חלון כיוון שה-firewall חוסם את הגישה לפורט. ביחרו cancel וזה אמור להסתדר. אם זה לא הסתדר פיתחו את פרויקט ה-VS מחדש או הפעילו מחדש את קובץ ה-exe)  
   הפעלת תוכנת השרת (מחשב החיישנים)  
   כדי להפעיל את תוכנת השרת עליך פשוט להקיש פעמיים על קובץ ההרצה DimonaServer.exe.  
   אם אתה מפעיל את תוכנת השרת ממחשב אחר מתוכנת הלקוח, בדוק את כתובת ה-ip של המחשב. אם מחשב השרת הוא בעל כתובת פרטית, על תוכנת הלקוח לרוץ על אותה הרשת. (אפשר שלא אך זה דורש קינפוגים בראוטר)  
   הפעלת תוכנת הלקוח (מחשב השליטה)  
   כדי להתחבר למחשב החיישנים, מקבלת תוכנת מחשב השליטה פרמטר אחד והוא כתובת ה-IP של מחשב החיישנים. אם לא תקיש פרמטר התוכנה תשתמש בערך ברירת המחדל והוא הכתובת 127.0.0.1 שהיא כתובת loopback כך שהתוכנה תחפש את השרת במחשב הנוכחי.  
   אם אתה מפעיל את תוכנת השרת והלקוח מאותו המחשב, כדי להפעיל את תוכנת הלקוח אתה יכול להקיש פעמיים על הקובץ DimonaClient.exe.  
   אם הפעלת את תוכנת השרת ממחשב אחר עליך להפעיל את תוכנת הלקוח מחלון הקונסול ושלוח כפרמטר את כתובת מחשב השרת. לדוגמה אם ה-ip של המחשב עליו רץ השרת הוא: 192.168.1.47, הפעל את התוכנה כך: DimonaServer 192.168.1.47.
2. למד את דרך פעולתה של תוכנת השליטה.
3. כדיי ללמוד את הפרוטוקול נאזין לתקשורת, בין תוכנת הלקוח לשרת, בעזרת WireShark.  
   אם אתה מריץ את תוכנת השרת והלקוח מאותו המחשב אז התקשורת ביניהם היא מסוג loopback. כדי להאזין לתקשורת מסוג loopback יש לפתוח ב-WireShark תתת מסוג loopback (זה קיים בגרסאות החדשות, בישנות זה בעייתי: <https://wiki.wireshark.org/CaptureSetup/Loopback>).   
   אם אתה מתקשה להאזין ב-WireShark לתקשורת, מצורף קובץ מוקלט של התקשורות שאתה יכול להשתמש בו במקום להאזין.  
   בקובץ המוקלט:  
   פריים 1 מהווה את הפקודה 1, ופריים 2 את התשובה, ערך הסנסור שמוצג למשתמש עבור פקודה זו הוא 74.   
   פריים 25 מהווה את הפקודה 2, ופריים 26 את התשובה, ערך הסנסור שמוצג למשתמש עבור פקודה זו הוא 991.   
   פריים 33 מהווה את הפקודה 3, ופריים 34 את התשובה, ערך הסנסור שמוצג למשתמש עבור פקודה זו הוא 200.  
   פתח את WireShark במחשבך, ועקוב אחרי החבילות.  
   פרוטוקולים רבים של שכבת האפליקציה מקודדים את התקשורת שלהם פשוט כטקסט. גם הפרוטוקול הזה מקודד כטקסט, כדי לראות את תוכן ההודעה מתורגם מקוד האסקי שלו לתוים תוכלו לעמוד על ההודעה של הפרוטוקול (על קוד האסקי שאתם רוצים לתרגם), ללחוץ על לחצן ימני ו-Show Pocket Bytes…. אתם אמורים לראות הודעה שהיא צירוף של מילים ומספרים שמגדירים את הפקודה והתשובה באופן ברור.   
   מה היררכיית הפרוטוקולים של חבילה שנשלחת בין השרת ללקוח או בין הלקוח לשרת:  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Ethernet II->IPv4->UDP**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   מה port היעד של חבילה שנשלחת ממחשבך למחשב השרת?  
   \_\_\_\_\_\_\_**destination port: 27015**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   בדוק מה הוא מבנה הפרוטוקול של שכבת האפליקציה לקבלת ערכי חיישנים.  
   מה מבנה הודעת הפקודה לקבלת ערך מחיישן, שמועברת ממחשב השליטה (לקוח) למחשב החיישנים (שרת). כתוב כמה סוגים של פקודות יש, ומה המשמעות של כל פקודה:

**מבנה הפקודה: SENSOR GET <Command\_number>**

**יש 3 סוגי פקודות, ומשמעותן:**

**SENSOR GET 1 – בקשה לטמפ' המים מהשרת**

**SENSOR GET 2 – בקשה לטמפ' הכור הגרעיני**

**SENSOR GET 3 – בקשה לעוצמת הזרימה במשאבה**  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
מה מבנה התשובה, שמועברת ממחשב החיישנים (שרת) למחשב השליטה (לקוח), כתגובה לפקודה: \_**מבנה התשובה הינו SENSOR DATA <Returned value>**  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. ברשותך פרוייקטים של visual studio שמכילים קוד C שמממש תוכנות דוגמה לשרת ולקוח UDP. הלקוח שבפרויקט הדוגמה מקבל פקודה מהמשתמש שולח אותה לשרת ומציג את התשובה שהתקבלה. השרת מקבל פקודה מהלקוח ומחזיר את הפקודה כתשובה כלומר מבצע echo.  
   אתה יכול להשתמש בפרויקט הדוגמה כדי לבדוק את מבנה הפקודות שמצאת. הפעל את השרת של דימונה, והרץ את הלקוח של פרויקט הדוגמה, ושלח בעזרתו את הפקודות לפי הפרוטוקול שמצאת, ובדוק את התשובה שהתקבלה.
2. העזר בפרויקט הדוגמה של תוכנת הלקוח של UDP כדי לכתוב תוכנת לקוח חדשה למחשב שליטה בכור בדימונה, התוכנה תקרא כל שניה את הערכים של שלושת החיישנים ותציג אותו.  
   יש להגיש את הקובץ עליו אתה עובד עכשיו (קובץ WORD) ואת הקובץ הראשי של התוכנה (socket.cpp) שכתבת שמכיל את הפונקציה main.  
   דגשים ופונקציות שימושיות –
   1. ראשית כדאי שתבדוק בעזרת תוכנת הדוגמה של הלקוח שאתה מבין את הפרוטוקול ויודע לשלוח פקודות ולקבל תשובות מהשרת. במידה שהפקודה ששלחת אינה מוכרת לשרת תקבל את התשובה: "ILLEGAL COMMAND".
   2. Sleep(1000) נחה ל-1000 מילי שניות.
   3. sscanf() לקריאה ממחרוזת.
   4. בתוכנית הדוגמה port השרת מוגדר ב-define. אל תשכח לשנות את port השרת שבתוכנית ל-port שמצאת ב-WireShark. (פורט השרת פה אינו well known port, פשוט משום שאין פורט שמוקדש לפרוטוקול הזה שיצרו בכור בדימונה ולכן השרת נאלץ להשתמש בפורט שאינו well known port)
   5. לתוכנית הלקוח מועבר כפרמטר, כתובת ה-IP של מחשב השרת. בהרצה ב-VS ובמקרה שלא מועבר פרמטר, אז כתובת ה-IP שמועברת היא כתובת loopback, כלומר כתובת מיוחדת שכאילו מחברת לאותו המחשב. אם השרת והלקוח נמצאים במחשבים שונים, אז יש לשנות את הפרמטר שמועבר לתוכנית.  
      שינוי הפרמטר שמועבר לתוכנית ב-debug – לחצן ימני על הפרויקט, properties, ב-Configuration Properties ב-Debugging, ב-Command Arguments. הפרמטר שמועבר לתוכנית הלקוח הוא כתובת ה-ip של השרת.