

PyBoard – מדריך למתקנים ומשתמשים במעבדה ב'

תאריך עריכה אחרון: 03.01.2020, עריכה אחרונה ע"י: אגם אנקורי, גרסה ראשונה ע"י: אגם אנקורי

"Computers are incredibly fast, accurate and stupid. On the other hand, a well-trained operator compared to a computer is incredibly slow, inaccurate and brilliant." - Someone, Somewhere, But not Einstein.

1. הקדמה

1.1 מבוא

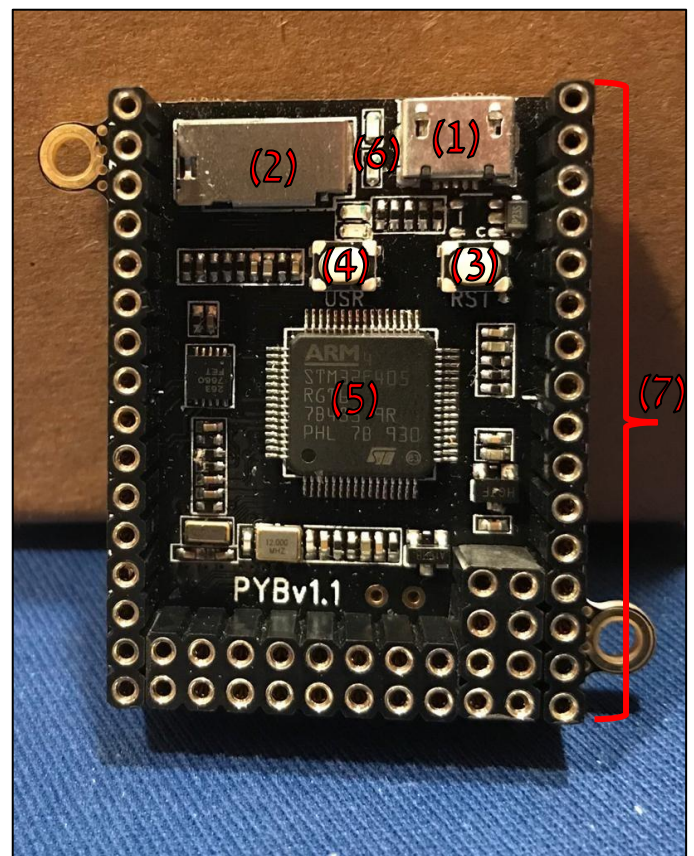
PyBoard הוא Microcontroller (או בהפשטה – מחשב קטן מאוד) שמריץ Python כמערכת הפעלה. אתם יותר ממוזמנים לקרוא בויקיפדיה על הפרויקט ועל פרויקט האב שלו (MicroPython), כמו גם להעמיק בחומרה עצמה (מיקרו-קונטרולר מסוג STM32 F4).

PyBoard (שמעתה נכנה בשם "בורד") יאפשר לכם לבצע אוטומטית מגוון ניסויים באמצעות חיישנים שונים דוגמת טמפרטורה, לחץ וקירבה, שליטה ברכיבים דוגמת מנועים ולייזרים וביצוע דגימות אוטומטיות, גם כשאתם לא בחדר הניסוי!

1.2 מבנה הבורד והחיבורים

איך זה עובד? כמו שאמרנו, על הבורד מותקנת מערכת הפעלה מבוססת Python. בבורד מותקנים מספר חיישנים Built-In (דוגמת מד תאוצה) והוא שופע חיבורים שמאפשרים ממשיך לחיישנים נוספים.

הבורד מאפשר לנו במספר שורות קוד קצרות (שנראה בהמשך) לשלוט ברכיבים הנוספים (לעיתים נכנה אותם "פריפריאליים"), לקבל מהם נתונים ולשמור אותם על כרטיס הזכרון. בנוסף, נוכל לשלוח פקודות למנועים השונים על מנת לשנות את סביבת הניסוי וכך לערוך ניסויים באופן אוטומטי ומדויק יותר.



איור 1: צילום ה-PyBoard. (1) חיבור USB (2) תושבת כרטיס MicroSD (3) כפתור הפעלה מחדש (4) כפתור ניתן לתכנות (5) מעבד (6) LED הדלקה ו-LED ניתן לתכנות. (7) חיבור פריפריאליים (חיישנים ומנועים)

2. שימוש ראשון ב-PyBoard

2.1 התקנה – Putty, קבצים חשובים

כדי להשתמש בבורד עלינו תחילה לחבר אותו למחשב. ראשית, ודאו שבפורט המתאים בבורד (מסומן ב-(2) באיור 1) מחובר כרטיס זכרון. קחו כבל USB וחברו לפורט מתאים במחשב, ולאחר מכן חברו לפורט המתאים (מסומן ב-(1) באיור 1) בבורד.

על הכרטיס עותקים של כל הקבצים הדרושים לשימוש בו. אם זהו שימוש ראשון בבורד במחשב זה, המשיכו לקרוא, אם לא, תוכלו לדלג לסעיף 2.2.

הכנסו ל"מחשב זה" ("This Computer"), אחד הכוננים המופיעים יקרא SDCARD, זהו כרטיס הזכרון המחובר לבורד. הכנסו אליו ופתחו את התיקיה "Setup and Backup". התיקנו את התוכנות הבאות במידה ומי מהן לא מותקנת על המחשב:

1. Putty (התוכנה באמצעותה נתקשר עם הבורד)
2. Python (סביר שכבר מותקן על המחשב)
3. Git (יאפשר עדכון של הקוד המותקן על הבורד בהמשך)

לאחר סיום ההתקנות, מומלץ ללחוץ על כפתור ה-RST (מסומן ב-(3) באיור 1). הכנסו שוב לתיקית כרטיס הזכרון והפעילו את התוכנית Update.bat, לאחר מספר שניות התוכנית תסתיים ואנחנו מוכנים לעבודה. לחצו על connect_to_pyboard.py, לאחר כמה רגעים תפתח תוכנת ה-Putty ותקבלו שליטה על הבורד.

כפי שראינו בהקדמה, מערכת ההפעלה של הבורד היא פייתונית ולכן בחלון הפקודה שנפתח תוכלו לשלוט ישירות בבורד בפקודות פייתוניות. מוזמנים לוודא שהפקודות השונות שאתם מכירים מבוצעות כהלכה בבורד.

חשוב לזכור – החישוב מתבצעים על גבי הבורד ולא על המחשב שלכם!

2.2 בדיקת תקינות והפעלה ראשונה

נוודא שהבורד שלנו מתפקד כצפוי. על גבי כרטיס הזכרון נמצאת החבילה lab והיא מכילה את הפקודות השונות שיקלו עליכם

בשימוש בבורד ומומלץ להשתמש בה. כדי לטעון את החבילה, כתבו את השורה הבאה ולאחריה Enter:

```
1. import lab
```

כעת החבילה טעונה, נרצה להשתמש בפונקציה `disco()`. לפני שנפעיל אותה, נלמד מה היא עושה. בחבילה קיימת הפונקציה `what` המציגה עזרה על הפונקציות, לדוגמה כתבו:

```
1. lab.what('disco')
```

למסך תודפס פסקה קצרה המסבירה את הפונקציה `disco`. כעת נשתמש בה, כתבו את השורה הבאה והסתכלו על הבורד:

```
1. lab.disco()
```

ה-LEDים על הבורד ידלקו ויכבו, תוכלו להפסיק פעולה זו בכל זמן ע"י הקשת `Ctrl + C`.

כדיי ללמוד אודות הפונקציות הנוספות הקיימות בחבילה, תוכלו להשתמש בפונקציה `what` מבלי להזין שם פונקציה ספציפי:

```
1. lab.what()
```

אודות חבילות נוספות והרחבה על יכולות הבורד, תוכלו להסתכל במדריך הרשמי של הבורד (ראו סעיף 4.1).

3. ניסוי הדגמה – מדידת XXX

3.1 רכיבים וחיבורים

לטובת ניסוי ההדגמה, תזדקקו לרכיבים הבאים:

1. חיישן XXX
2. שלושה חוטים
3. XXXX

חברו את המערכת באופן הבא:

איור XXX: סכימת חיבור הניסוי

מה עשינו כאן?

ראשית, סיפקנו מתח לחיישן XXX באמצעות החוט XXX. שנית, הארקנו את החיבור באמצעות החוט XXX. לבסוף, איפשרנו מעבר של נתוני הדגימה באמצעות החוט XXX.

3.2 מדידת XXX

כעת נרצה להשתמש בתוצאות הדגימה. ראשית, נתחבר לבורד באותו אופן שעשינו קודם. כעת, נשתמש בפונקציה `lab.start_continuous_measurement()`. כמקודם, השתמשו בפונקציה `lab.what()` כדי ללמוד איך להשתמש בה.

לאחר שתסיימו את המדידה, קובץ המדידות יישמר על הבורד, תחת התיקיה `Results`. כדי להגיע אליה, תאלצו לכבות ולהדליק את הבורד (באמצעות לחיצה על כפתור ה-RST, ראו איור 1).

פתחו את קובץ המדידה והתבוננו בתוצאות. שימו לב, התוצאות שקיבלתם אינן מאופסות או ביחידות של XXX! חישבו כיצד לנרמל את התוצאות וחזרו על הניסוי כדי לקבל תוצאות ביחידות מתאימות.

3.3 טיפים נוספים

נוסף לפונקציה `lab.what()` המאפשרת הבנה של הפונקציות בתוך החבילה שנכתבה במעבדה, תוכלו גם להשתמש בפקודה `help()` שתספק לכם מידע נוסף אודות הפונקציות הפייתוניות הקיימות בבורד וקיצורי המקלדת.

4. העמקה – לא חובה

4.1 חומרי קריאה נוספים

כפי שראינו, מערכת ההפעלה של הבורד מבוססת פייתון ומסוגלת לבצע (כמעט) כל פקודה פייתונית המוכרת לכם - לולאות For ו-List Comprehension, While, שמירת משתנים, כתיבה וקריאה של קבצים ועוד ועוד. כדי שתוכלו לנצל את כל המגוון הזה, כמובן שתצטרכו לדעת להשתמש בפייתון. מומלץ לעשות לעצמכם רענון קצר, לדוגמה באמצעות הספר החינמי הבא:

https://en.wikibooks.org/wiki/Python_Programming

להעמקה נוספת, מומלץ לקרוא את המדריך הרשמי לשימוש ב-MicroPython. זוכרים שאמרנו שהמערכת שלנו "מבוססת" פייתון? זה נכון כיוון שכדי להתאים אותה לריצה על הבורד (שהוא דל משאבים ופרמיטיבי יחסית למחשב שלכם), היה צורך לבצע התאמה לשפה. כך נולדה MicroPython והיא קצת שונה (אבל לא מאוד) מהשפה המוכרת לכם. תוכלו לקרוא עוד כאן:

<https://en.wikipedia.org/wiki/MicroPython>

<https://docs.micropython.org/en/latest/index.html>

לבסוף, שימו לב שהרכיבים השונים שתחברו לבורד דורשים לעיתים צורת חיבור מיוחדת או קריאה מסוימת של המידע. בנוסף, לא תמיד רשום על הרכיב מה הוא עושה אלא רק קוד דגם כלשהו. כדי לפענח כל רכיב, מומלץ לחפש את קוד הדגם באינטרנט. ניתן כמובן להעזר באלקטרונאי המעבדה ולהשתמש בספר הבא המפרט על רוב הרכיבים ואופן החיבור שלהם:

<https://images-eu.ssl-images-amazon.com/images/I/C1lrpIfADaS.pdf>

4.2 למתכנתים

כמו שראינו, כדי לעטוף בצורה נוחה חלק מהפונקציות שמגיעות עם הבורד, טענו את החבילה `lab`. חבילה זו נבנתה ע"י סטודנטים מהאוניברסיטה והיא פתוחה לעריכות ושינויים. למעשה, בכל פעם שתפעילו את `Update.bat`, אתם מתחברים באמצעות `git` לאתר החבילה ומעדכנים אותה. ה-`git` עצמו נמצא כאן:

https://github.com/stormage2/HUJI_PyBoard

אתם יותר ממוזמנים לפתוח branch-ים חדשים ולכתוב `pull requests` או פשוט לעשות `Push` לפונקציות חדשות שכתבתם. בשלב זה לא ברור מי יתחזק את החבילה, אפשר לדבר על זה ב-`github`.

כמה דגשים לכתיבת פונקציות חדשות:

1. אין בבורד משאבים! חישבו היטב מה אתם מעלים לזכרון, מתי אתם עושים `flush` ומתי אתם גורמים ל-`block`!
2. השתמשו בקובץ `__init__.py` שנמצא על הבורד (או בגיט) כדי להבין את פורמט `docstring` בחבילה. MicroPython מתנהגת קצת שונה מפייתון "רגיל" ולכן יש לנסח את הדוקסטרין בצורה שהפונקציה `what` תוכל לפענח אותו.
3. בדקו שהפונקציות שרשמתם גנריות מספיק. אין טעם לממש פונקציה שמתחברת ל-Pin יחיד (כי רובם זהים) ואין סיבה לכתוב פונקציה שיודעת לעבוד עם פריפריאלי יחיד (אלא אם הוא -מאוד- ייחודי).
4. לא לשכוח להוסיף קומנטים לפושים! פוש עם קומנט גנרי לא ברור לא יעבור.

דבר אחרון למיטיבי לכת – כמו שאמרנו, הבורד הוא מחשב עצמאי לחלוטין ולא באמת צריך אתכם/המחשב שלכם באיזור כדי להמשיך בחישוב/מדידה. אם תחברו את הבורד למחשב ללא כרטיס זכרון תראו שעל הבורד עצמו נמצאים שני קבצי סקריפט: boot.py ו-main.py. הקובץ boot.py הוא הקובץ הראשון שרץ כשהבורד **מקבל חשמל**. שימו לב שאין שום חובה לחבר אותו למחשב לטובת זה – תוכלו לחבר אותו עם ממיר מתח לכל שקע חשמלי.

מה אפשר לעשות עם זה? נניח ואתם רוצים למדוד את הטמפרטורה בחדר בלי להשאיר את המחשב דולק (ישפיע על הטמפרטורה בחדר). תוכלו לכתוב סקריפט שמתחבר לחיישן טמפרטורה, לשמור אותו כקובץ py. על הבורד עצמו (שימו לב – לא כרטיס הזכרון!) ולהוסיף ל-boot.py את השורה הבאה:

```
1. import your_script_name
```

נתקו את הבורד מהמחשב וחברו אותו לשקע החשמלי (ואת החיישן הרלוונטי בהתאם לסקריפט שלכם). הסקריפט ירוץ מרגע שהבורד יסיים את תהליך הboot ועד שתנתקו אותו מהחשמל.

תהליך כזה כמובן דורש מכם קצת יותר מאמץ כי תצטרכו לממש בעצמכם את הליך הדגימה ולוודא שאתם שומרים את הנתונים דווקא לכרטיס הזכרון, אבל אם הגעתם עד כאן כנראה שאתם מסוגלים לזה.

בהצלחה!