

פרויקט סמסטרלי בנושא מערכת השקיה

סמסטר א'

שנה"ל תשע"ח

תאריך הגשה : 28.2.2018

מגישים
דקל משה, ת.ז. 200862142, הנדסת תוכנה דקל תמם, ת.ז. 305242125, הנדסת תוכנה

תוכן העניינים

3	הצעת פרויקט	0
4	רקע הטכני ותאוריה	1
4	מבוא	1.1
5	I2C	1.2
7	PWM	1.3
8	UART	1.4
10	ADC	1.5
11	MQTT	1.6
12	תכן המודול	2
12	מודל התנהגות	2.1
12	תרשים פעילות 1	2.1.1
13	טבלת tasks and interrupts	2.2
14	הסבר על טבלת tasks and interrupt	2.3
15	יישום הפרויקט	3
16	מדריך למשתמש	4
16	חיבורים במטריצה ו-CC3220S-LAUNCHXL	4.1
17	סביבת הפיתוח – Code Composer Studio (CSS)	4.2
17	שימוש במודול	4.3
19	סיכום אישי	5
20	ביבליוגרפיה	6

0 הצעת פרויקט

שם הפרויקט : מערכת השקיה

תמצית הפרויקט :

בפרויקט זה נבנה מערכת זמן אמת אשר תתריע ללקוח מתי צריך להשקות את עציץ הנוי הביתי. ה-board ימוקם בתוך העציץ הביתי על מנת לדגום את הלחות.

חיישן הלחות ידגום את רמת הלחות בעציץ וכאשר רמת הלחות תרד מתחת לרף מסוים ויהיה צורך להשקות את העציץ, נורת הLED תשנה את העוצמה על מנת להתריע ללקוח שנדרש להשקות את העציץ. באמצעות חיישן התאורה נבדוק אם יש אור בחדר ונדליק את הנורה (הנורה הצהובה של ה-board) רק כאשר יהיה חושך בחדר על מנת לחסוך בחשמל, נשתמש בכפתורים מובנים של המערכת אחד ידליק ויכבה נורת LED והשני יעביר ויחזיר את המערכת למצב חסכוני שבו החיישנים לא דוגמים.

הרכיבים שבהם נשתמש הינם :

- TI SimpleLink CC3220S LAUNCHXL – זהו ה-board שיהווה את לב המערכת.
- חיישן לחות
- TSL 2561 – חיישן אור
- LED – נורת LED
- Button

באמצעות מערכת זו, אנשים יוכלו לדעת מתי צריך להשקות את העציץ ובאיזה כמות. במידה והם ישכחו לעשות זאת יופיע להם התרעה ממשיית על מנת שלא יישכחו.

סטודנטים המבצעים : דקל משה, דקל תמם

1 רקע הטכני ותאוריה

פרק זה מכיל רקע טכני ותאוריה עבור הפרויקט מערכת השקיה. בו כלול מידע על כל הממשקים הקשורים לתפקוד הנכון של תוצר הפרויקט.

1.1 מבוא

במסגרת הפרויקט החלטנו לפתח מערכת embedded אשר מתריעה ללקוח כאשר נדרש להשקות את עציץ הנוי הביתי, על מנת שהעציץ לא ינבול וימות.

תחילה יש למקם את המערכת בתוך העציץ כאשר חיישן האור חשוף וממקום בצורה גלויה על מנת לקלוט את האור בחדר, את חיישן הלחות יש למקם באדמה.

המערכת פועלת בצורה הבאה:

- לאחר שהמערכת הותקנה כראוי היא תכנס והחיישנים יכנסו לפעולה ויבצעו דגימות של הסביבה על מנת לבצע פלט מתאים.
 - חיישן הלחות ידגום את מצב הלחות, כאשר רמת הלחות תהיה מתחת לסף שנקבע נורת ה-LED תכנס לפעולה ותבצע אור משתנה על מנת להתריע שנדרש להשקות את העציץ. לאחר שהעציץ יקבל מים והחיישן לחות יבצע דגימה, נורת ה-LED תכבה.
 - חיישן האור (TSL 2561) יבצע דגימות של האור בחדר, כאשר החדר יהיה חשוך תדלק נורה (הנורה הצהובה של ה-board) על מנת שיהיה אפשר לראות את העציץ. כאשר החדר מואר הנורה תכבה על מנת לחסוך בחשמל.
 - לחיצה על כפתור SW2 של הבורד ידליק/יכבה את הנורה D6.
 - לחיצה על כפתור SW3 מעביר/מחזיר את המערכת למצב חסכוני שבו החיישנים מפסיקים לדגום.
- מערכת זו תיתן מענה עיקרי לבעיה שאנשים שוכחים באופן קבוע להשקות את העציץ. ובאמצעות שימוש במערכת יוכלו לפתור את הבעיה בקלות וביעילות מאוד גבוהה.

I2C 1.2

I2C הינו פרוטוקול סדרתי למעגלים משולבים. הארכיטקטורה שלו היא של master ו slave. הפרוטוקול מאפשר להרבה slave לתקשר עם ה-master. I2C דורש רק שני חוטי האות על מנת להחליף מידע.

רוב מכשירי ה-I2C יכולים לתקשר ב-100kHz או 400kHz. עבור כל 8 סיביות של נתונים שנשלחים צריך להעביר עוד נתון אחד של meta data (ACK/NACK).

I2C פותח במקור בשנת 1982 ע"י Philips. המפרט המקורי מאפשר רק עבור חיבור של 100kHz וסיפק רק עבור 7 סיביות של הכתובות דבר אשר הגביל את מספר המכשירים על הערוץ ל-112 בלבד.

כל ערוץ I2C מורכב משני אותות – SCL ו SDA. SCL הוא האות של השעון וה-SDA הוא האות של הנתונים. האות של השעון נוצר תמיד ע"י מנהל הערוץ הנוכחי. התקני slave מסוימים עשויים לאלץ את ה-master לבצע השהייה לשליחת המידע.

האיתות חייב לדבוק בפרוטוקול בטוח בשביל שהמכשירים על הערוץ ידעו שזהו תקשורת I2C תקינה. הודעות מחולקות לשני סוגים של frame. Address frame. כאשר ה-master מורה ל-slave איזה הודעה נשלחת. Data frame – הינו 8 סיביות של מידע המועבר מה-master ל-slave או להיפך.

כדי לאתחל את ה-address frame ה-master משאיר את ה-SCL גבוה ומושך את ה-SDA נמוך. זה גורם לכל מכשירי ה-slave לדעת שהשידור עומד להתחיל. אם שני מכשירי master רוצים לקחת בעלות על הערוץ בו זמנית, המכשיר אשר מושך את ה-SDA נמוך ראשון, מנצח במרוץ ומקבל בעלות על הערוץ.

Address frame תמיד ראשון בכל קצף תקשורת חדש. עבור כתובת של 7 סיביות, הכתובת מוציאה את הסיבית הכי חשובה בעקבותיו מגיעה סיבית R/W שמורה אם זו פעולה של כתיבה או קריאה.

הסיבית ה-תשיעית של ה-frame הינו NACK/ACK. ברגע ש-8 סיביות נשלחות, המכשיר שמקבל את ההודעה מקבל שליטה על ה-SDA. אם המכשיר המקבל לא אינו מושך את הקו הנמוך של ה-SDA לפני הפולס התשיעי של השעון, ניתן להסיק שהמכשיר המקבל לא קיבל את הנתונים או לא ידע כיצד לנתח את ההודעה. במקרה זה, העברה נעצרת, וזה תלוי ב-master של המערכת להחליט איך להמשיך.

לאחר ש-address frame נשלחה, הנתונים יכולים להתחיל להיות מועברים. ה-master ימשיך לייצר פולסי שעון במרווחים קבועים, והנתונים ימוקמו על ה-SDA ע"י ה-master או ה-slave, תלוי אם הסיבית של ה-R/W מצביעה על פעולת קריאה או כתיבה. מספר ה-data frames הוא שרירותי, ורוב מכשירי ה-slave יבצעו הגדלה אוטומטית לרישום הפנימי, מה שאומר שהקריאה או הכתיבה הבאה יבואו מה-register הבא בתור.

ברגע שה-data frame נשלח, ה-master ייצור תנאי עצירה. תנאי עצירה מוגדר ע"י מעבר מנמוך לגבוה על ה-SDA אחרי המעבר על ה-SCL. במהלך פעולת כתיבה רגילה, הערך ב-SDA צריך לא להשתנות כאשר SCL גבוה, כדי להימנע מתנאי עצירה שקרי [1].

החיישנים בפרויקט המדברים בפרוטוקול I2C הם:

חיישן TSL2561 מחובר לפינים מספר 1 ו-2, חיישן הממיר קלט אור לפלט דיגיטלי.
כדי לעבוד עם חיישן זה נדרש לשלוח אליו פקודת הפעלה (0x03) אל רגיסטר השליטה שלו (0x00)
ולקרוא נתונים מרגיסטר העבד (0x39) בשני ערוצים, בדף הנתונים של החיישן צורפה פונקציה אשר
באמצעותה אנחנו ממירים את המידע של הערוצים מהחיישן ליחידות Lux שמציינות את כמות
האור הנקלט [2].
באמצעות הפלט, אנחנו בודקים את הסביבה שבה המערכת פועלת מוארת או חשוכה, במקרה והיא
חשוכה אנו רוצים לחסוך בחשמל ולמנוע הדלקה חסרת טעם של מנורת התאורה.

PWM 1.3

PWM - Pulse Width Modulation הינו מונח לתיאור סוג של אות דיגיטלי ומשמש לקידוד הודעה לתוך אות פועם. למרות שטכניקה זו יכולה לשמש כדי לקודד מידע עבור השידור השימוש העיקרי שלה הינו לאפשר את השיטה של הכוח המסופק למכשירים החשמליים. בנוסף PWM הוא אחד משני האלגוריתמים העיקריים המשמשים מטענים סולאריים [3].

לכל ערוץ יש חוט יחיד משלו, לכן אם יש שמונה ערוצים ב-receiver נצטרך לחבר 8 חוטים כדי לקרוא את הקלט לתוך הבקר. כמעט לכל receiver יש יציאה של WPM והוא מאוד פופולרי אך הבעיה העיקרית היא שצריך לחבר חוט לכל ערוץ וזה עלול להוביל בקלות לטעויות [4].

PWM Signal הינה שיטה ליצירת אות אנלוגי באמצעות מקור דיגיטלי. PWM signal מורכב משני רכיבים עיקריים המגדירים את התנהגותו: מחזור עבודה ותדירות. מחזור העבודה מתאר את משך הזמן שהאות נמצא במצב גבוה (on) כאחוז מהזמן הכולל שנדרש להשלמה של מחזור אחד. התדירות קובעת כמה מהר ה-PWM ישלים מחזור, כלומר כמה מהר הוא עובר ממצב גבוה למצב נמוך. ע"י מחזור של כיבוי והדלקה של אותות דיגיטליים בקצב מספיק מהיר, ועם מחזור עבודה מסוים, הפלט ייראה כמתח אנלוגי קבוע בעת האספקה של החשמל למכשירים. התדר של ה-PWM signal צריך להיות מוגדר ותלוי ביישום ובזמן התגובה של המערכת המופעלת [5].

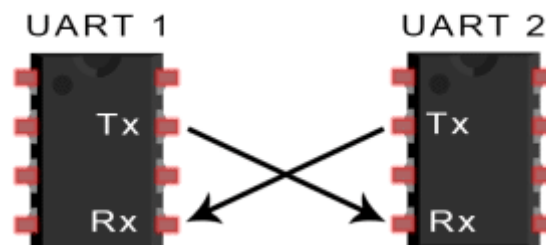
אנו משתמשים ב-PWM בפרויקט על מנת לשלוט בעוצמת התאורה של מנורת הLED המחוברת לפין הLED D7 (LED0) פין 64, על ידי שליטה במתח של הנורה אנו יכולים מורים על מצב הלחות הנקלט באמצעות חיישן הלחות (חיישן ADC) [6].

אנו מתריעים כך על מצב הלחות הנקלט בצורה ויזואלית, כאשר הלחות תקינה האור מכובה וככל שהלחות יורדת אנו מגבירים את עוצמת הLED וכאשר הלחות במצב קריטי אנו מהבהבים את הנורות.

UART 1.4

UART - universal asynchronous receiver-transmitter הוא התקן חומרה לתקשורת טורית אסינכרונית שבו פורמט הנתונים ומהירויות השידור ניתנים לשינויים. רמות ושיטות האותות החשמליים מטופלים על ידי מעגל חיצוני UART. UART הוא בדרי"כ מעגל משולב יחיד (IC) המשמש לתקשורת טורית על מחשב או יציאה טורית על התקן הקיפי. UART כיום נכללים ב-microcontroller [7].

בתקשורת UART, שני UART מתקשרים ישירות אחד עם השני. ה-UART שמשדר ממיר נתונים מקבילים ממכשיר שליטה כמו CPU, לטופס סדרתי, ומעביר אותו ברצף ל-UART המקבל והוא ממיר את הנתונים הטוריים בחזרה לנתונים מקביליים עבור ההתקן המקבל. נדרשים רק שני חוטים בשביל להעביר נתונים בין שני UARTים. נתונים זורמים מהפין של ה-Tx מה-UART השולח, לפין של ה-Rx ל-UART המקבל.



איור 1: תקשורת UART

העברת נתונים ב-UART מתבצעת בצורה אסינכרונית, כלומר אין אות שעון אשר יבצע סנכרון על הסיביות של הפלט מה-UART המעביר ל-UART המקבל אשר דוגם את הסיביות. במקום האות של השעון, ה-UART המשדר מוסיף סיביות של התחלה/סוף למנה המועברת. סיביות אלו מגדירות את ההתחלה ואת הסיום של המנה המועברת כך שה-UART המקבל ידע מתי להתחיל לקרוא את הסיביות.

כאשר ה-UART המקבל מזהה את הסיביות של ההתחלה הוא מתחיל לקרוא את המנה בתדר מסוים המכונה בשם קצב השידור (baud rate). קצב השידור הוא מדד למהירות העברת הנתונים, הנמדד ב-ביט לשנייה (bps). שני ה-UART חייב לפעול בערך באותו קצב שידור. קצב השידור בין ה-UART השולח למקבל יכול להשתנות רק ב-10%. שני ה-UART חייבים להיות מוגדרים לשליחה וקבלה לאותו מבנה של מנה.

כיצד ה-UART עובד: UART שהולך להעביר נתונים, מקבל את נתונים מהערוץ הנתונים. ערוץ הנתונים משמש לשליחת נתונים ל-UART באמצעות התקן אחר כמו CPU או microcontroller. הנתונים מועברים מערוץ הנתונים ל-UART השולח בצורה מקבילית. לאחר העברה, ה-UART מקבל את הנתונים המקביליים מערוץ הנתונים, הוא מוסיף את הסיביות התחלה, סיביות זוגיות/אי-זוגיות

(parity), וסיביות עצירה, ובעצם יוצר את המנה. לאחר מכן הפלט של המנה הינו סדרתי, סיביות לאחר סיביות לפין ה-Rx. ה-UART המקבל ממיר את הנתונים בחזרה לצורה המקבילית ומסיר את הסיביות ההתחלה, הזוגיות, הסוף. לבסוף ה-UART המקבל מעביר את המנה בצורה מקבילית לערוץ הנתונים. כל מנה מכילה סיביות 1 של התחלה, 5-9 סיביות נתונים, סיביות אופציונאליות של זוגיות 1 או 2 סיביות של עצירה [8].

אין רכיב אשר משתמש בפרוטוקול זה בפרויקט אך על מנת לענות על הדרישות השתמשנו בו על מנת לבצע הדפסה לטרמינל אשר מתריעה כאשר המערכת עוברת למצב חיסכון, חוזרת ממנו וכשיש הרבה אור. לביצוע הדפסות אלו בנינו מחלקת עזר שניתן לקרוא לשיטותיה מכל מחלקה אחרת, כדי להשתמש בה נדרש לאתחל אותה ואת UART לאחר שזה נעשה באמצעות שיטה להדפסה קוראים להדפסה מכל מקום אחר.

ADC 1.5

ADC – Analog to Digital Converter מתרגמים אותות אנלוגיים, אותות מהעולם האמיתי כמו טמפרטורה, לחץ, מתח, זרם, מרחק או עוצמת קול לייצוג דיגיטלי של אות זה. ייצוג דיגיטלי זה יכול להיות מעובד, מחושב, מועבר או מאוחסן. במקרים רבים, תהליך ההמרה מאנלוגי לדיגיטלי הוא רק צעד אחד בתוך לולאה גדולה יותר של מדידה ובקרה שבה מעובד המידע הדיגיטלי ולאחר מכן מוחזר חזרה לאות אנלוגי.

ADC דוגם גלים אנלוגיים במרווחי זמן אחידים ונותן לכל דגימה ערך דיגיטלי בהתאמה. הערך הדיגיטלי מופיע בפלט של ה-converter בפורמט בינארי. הערך מתקבל ע"י חלוקת מתח הקלט האנלוגי שנדגם במתח הייחוס והכפלה במספר הקודים הדיגיטליים. הרזולוציה של ה-converter נקבעת ע"י מספר הביטים הבינאריים בפלט של הקוד [9].

פותחו דרכים רבות כדי להמיר אות אנלוגי, כל אחד עם הנקודות החזקות והחלשות שלו. הבחירה של ADC מסוים, עבור יישום מסוים בדרי"כ מוגדר לפי הדרישות שיש ליישום. אם צריך מהירות עבור היישום, עדיף לבחור ב-ADC מהיר. אם צריך דיוק אז עדיף להשתמש ב-ADC מדויק. אם אתה מוגבל בחלל אז עדיף להשתמש ב-ADC קומפקטי. כל ה-ADC עובדים תחת אותו עיקרון: הם צריכים להמיר אות למספר של סיביות. רצף של סיביות מייצג את המספר והמשקל של כל ביט הינו כפול מהמשקל של הביט שבא אחריו, החל מה-MSB עד ל-LSB. המטרה בסופו של דבר היא למצוא את הרצף של הסיביות אשר מייצגות את הערך האנלוגי [10].

החיישנים בפרויקט המדברים בפרוטוקול ADC הם:

חיישן לחות המחובר לפין 59 בבורד, באמצעות חיישן זה אנחנו בודקים את רמת הלחות וכך יודעים את כמות המים.

החיישן מודד את רמת ההתנגדות בין המזלגות שלו, ככל שיש יותר לחות ביניהם רמת ההתנגדות יורדת. [11]

לפי הקריאות של כמות המים אנחנו קובעים את התנהגות הלד באמצעות פרוטוקול PWM.

MQTT 1.6

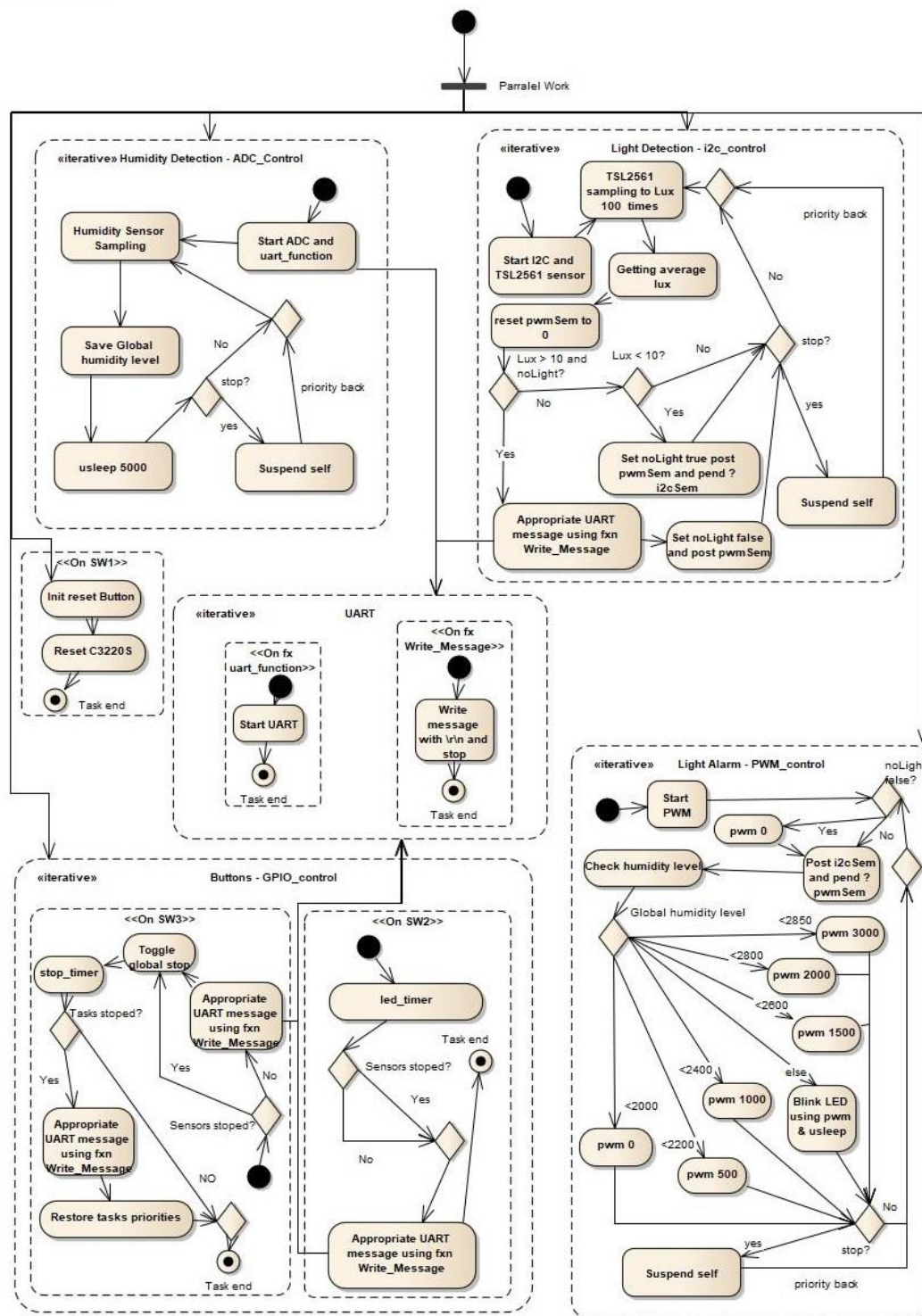
MQTT הינו פרוטוקול להעברת הודעות המספק משאבי רשת מוגבלים ללקוחות עם דרך פשוטה להפצת מידע טלמטריה (העברת נתוני מדידה ממרחק רב). הפרוטוקול משתמש בתקשורת publish/subscribe, אשר משמש עבור תקשורת בין מכונה למכונה (M2M) וממלא תפקיד חשוב באינטרנט של הדברים (IoT). MQTT מאפשר למכשירי אינטרנט של הדברים לשלוח או לפרסם מידע לגבי נושא מסוים לשרת המתפקד כמתווך הודעות MQTT (broker). ה-broker מעביר את המידע אל ה-clients הרשומים לנושא. לקוחות יכולים להירשם למספר נושאים שונים. פרוטוקול MQTT הוא בחירה טובה עבור רשתות אלחוטיות אשר חווים זמני ההמתנה שונים כתוצאה מרוחב פס שונה או מחיבור לא אמין לאינטרנט. אם החיבור בין ה-client ל-broker ייפסק, ה-broker ישמור את ההודעות המיועדות ל-client ויעביר לו אותם כאשר הוא יתחבר לאינטרנט שוב. אם החיבור ביניהם ייפסק באופן לא תקין, וללא הודעה מוקדמת ה-broker יסגור את החיבור.

MQTT session מחולקת לארבעה שלבים: חיבור, אימות, תקשורת וסיום. ה-client מתחיל ע"י יצירת חיבור TCP/IP ל-broker באמצעות port סטנדרטי או port מותאם אישית המוגדר ע"י ה-broker. בעת יצירת החיבור השרת עלול להמשיך שיחה ישנה אם הלקוח כבר היה מחובר אליו [12]. לא נעשה שימוש בפרוטוקול זה בפרויקט.

2 תכן המודול

2.1 מודל התנהגות

2.1.1 תרשים פעילות 1



איור 1: תרשים פעילות של מערכת השקיה

2.2 טבלת tasks and interrupts

טבלה 1 : tasks and interrupts

Number	HWI	Swi function	Thread type	Priority	Interrupt number
1.		ADCTask_func	Task	10	
2.		GPIO_function	Task	1	
3.		i2c_function	Task	4	8
4.		PWM_function	Task	7	24 (Pin64 uses Timer2B)
5.	Timer_led	led_toggle	swi	-2	19
6.	Timer_Stop	stop_toggle	swi	-2	19
7.	GPIO_BUTTON0	button_0_fxn	hwi	-2	2
8.	GPIO_BUTTON1	button_1_fxn	hwi	-2	2
9.	Write_Message	UART_writePolling	hwi	-2	5

על פי המדריך למשתמש ל-CS32xx [13].

Interrupt Number (Bit in Interrupt Registers)	Vector Address or Offset	Description	Type
0	0x0000.0040	GPIO Port 0 (GPIO 0-7)	
1	0x0000.0044	GPIO Port A1 (GPIO 8-15)	
2	0x0000.0048	GPIO Port A2 (GPIO 16-23)	
3	0x0000.004C	GPIO Port A3 (GPIO 24-31)	
5	0x0000.0054	UART0	
6	0x0000.0058	UART1	
8	0x0000.0060	I2C	
14	0x0000.0078	ADC Channel-0	
15	0x0000.007C	ADC Channel-1	
16	0x0000.0080	ADC Channel-2	
17	0x0000.0084	ADC Channel-3	
18	0x0000.0088	WDT	
19	0x0000.008C	16/32-Bit Timer A0A	
20	0x0000.0090	16/32-Bit Timer A0B	
21	0x0000.0094	16/32-Bit Timer A1A	
22	0x0000.0098	16/32-Bit Timer A1B	
23	0x0000.009C	16/32-Bit Timer A2A	
24	0x0000.00A0	16/32-Bit Timer A2B	
35	0x0000.00CC	16/32-Bit Timer A3A	
36	0x0000.00D0	16/32-Bit Timer A3B	
46	0x0000.00F8	uDMA Software Intr	
47	0x0000.00FC	uDMA Error Intr	
161	0x0000.02C4	I2S	
163	0x0000.02CC	Camera	
168	0x0000.02E0	RAM WR Error	
171	0x0000.02EC	Network Intr	
176	0x0000.0300	SPI	

איור 3: CC32xx Application Processor Interrupts

2.3 הסבר על טבלת tasks and interrupt

1. ADCTask_func בעדיפות 10 מכיוון שזה הפעולה החשובה והמהירה ביותר של המערכת, ונרצה לתעדף אותה על פני שאר הפעולות. פעולה זו היא בעצם הדגימה של חיישן הלחות ונרצה שפעולה זו תהיה בעדיפות הגבוהה ביותר.
2. GPIO_function בעדיפות 1 מכיוון שזוהי לא פעולה קריטית, שמטרתה הינה להפסיק את פעילות המערכת ולכן אין בעיה שתהיה בעדיפות נמוכה.
3. i2c_function בעדיפות 4 מכיוון שזוהי פעולה לנוחות של המשתמש בלבד ולא פעולה קריטית שמטרתה לבדוק אם יש אור או חושך בחדר.
4. PWM_function בעדיפות 7 מכיוון שזוהי פעולה חיונית למערכת ומטרתה הינה להתריע למשתמש כאשר יש צורך להשקות את הצמח ולכן בחרנו בעדיפות גבוהה.
5. Timer_led בעדיפות 2- על פי המדריך של TI [13].
6. Timer_Stop בעדיפות 2- על פי המדריך של TI [13].
7. GPIO_BUTTON0 בעדיפות 2- על פי המדריך של TI [13].
8. GPIO_BUTTON1 בעדיפות 2- על פי המדריך של TI [13].

3 יישום הפרויקט

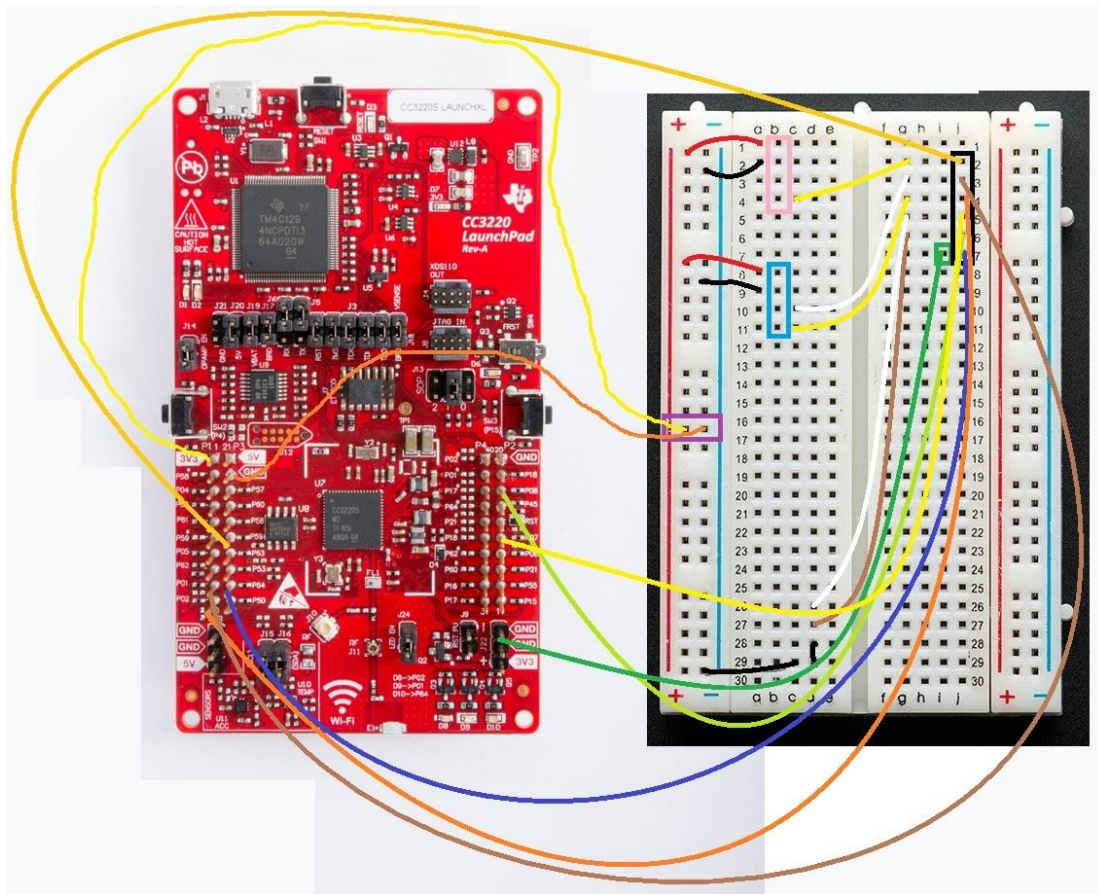
קוד המלא של הפרויקט, כולל הנחיות הידור (compilation) והתקנה נמצא ב dropbox :

המודול נערך והועלה לשמירה בתאריך 28.2.201 ב-15

סרטון וידאו המראה את פונקצינליות הפרויקט נמצא ב - https://youtu.be/xdu5FkYna_8

4 מדריך למשתמש

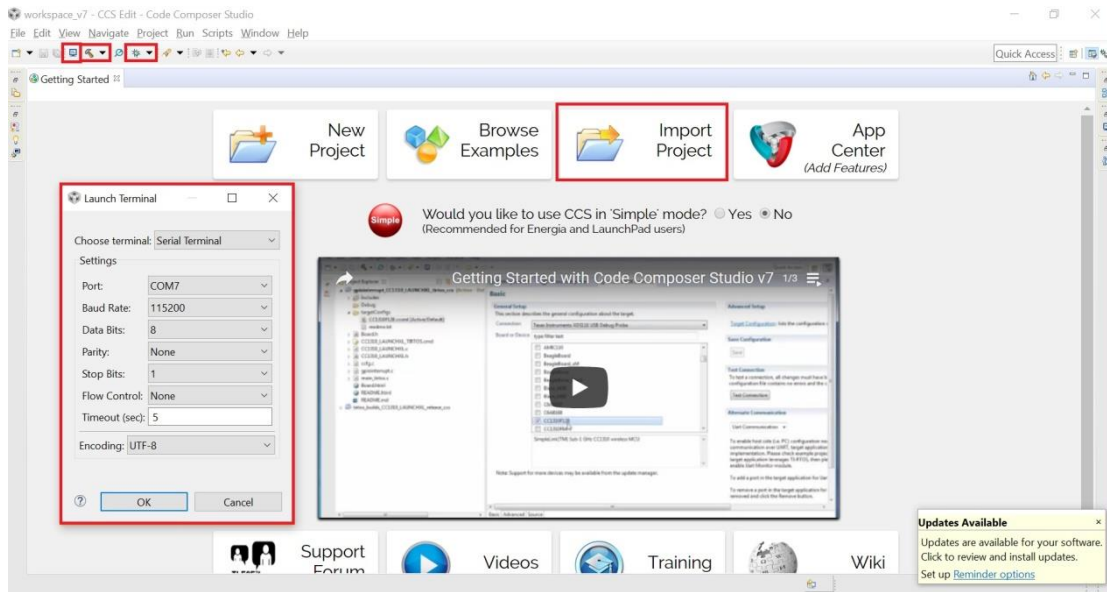
4.1 חיבורים במטריצה ו-CC3220S-LAUNCHXL



איור 4: חיבור למטריצה וללוח CC3220S-LAUNCHXL

- חיישן אור TSL2561 מתחבר לאזור הכחול בהיר במטריצה על פי איור 4 בצורה זאת:
- VCC מתחבר אל b8, GND מתחבר אל b9, SCL מתחבר אל b10, SDA מתחבר אל b11.
- חיישן לחות 土壤湿度检测 מתחבר לאזור הוורוד במטריצה על פי איור 4 בצורה זאת:
- VCC מתחבר אל b1, GND מתחבר אל b2, DO מתחבר אל b3, AO מתחבר אל b4.
- הלד מתחבר בחוט אחד לאזור הירוק במטריצה ובחוט השני אל GND בלוח על פי איור 4.
- חיבור החשמל מהלוח אל המטריצה מתחבר לאזור הסגול במטריצה והחוטים על פי איור 4.
- חיבור פניי הלוח אל המטריצה מתחבר לאזור השחור במטריצה והחוטים על פי איור 4.
- צריך לחבר חוטים נוספים העוברים במטריצה על פי איור 4.

4.2 סביבת הפיתוח – Code Composer Studio (CSS)



איור 5 : המחשת CCS

1. הורד את CCS והתקן אותה על המחשב.
2. התקן את ה SDK המתאים לCC32xx בתיקיה בה התקנת את CSS.
3. פתח את CSS
4. ייבא את הפרויקט Watering_Project ממסך ההתחלה
(View→Getting Started→Import Project)
5. בנה את הפרויקט באמצעות לחיצה על צלמית הפטיש.
6. בצע ניפוי באגים לפרויקט באמצעות לחיצה על צלמית הגיוק.
7. הרץ את הפרויקט על ידי לחיצה על צלמית החץ הירוק (דומה לצלמית ניגון סרט בVCR).

- כדי לראות פלט מהלוח נדרש לפתוח טרמינל באמצעות לחיצה על צלמית מסך המחשב ובחירת הפורט המתאים – שאר האפשרויות נדרשות להיבחר בדומה לאיור 5

4.3 שימוש במודול

כאשר המודול רץ החיישנים (לחות, אור) דוגמים נתונים מהסביבה על פי הדגימות ניתן יהיה לראות שינוי בנורת הלבד.

כאשר חיישן האור דוגם ערכים גבוהים (אור חזק) נורת הלבד תכבה על מנת להראות יכולת חיסכון בחשמל, למען ההדגמה אנו כיוונו את תגובה זאת לאור חזק על מנת שיהיה ניתן להראות את יכולת המערכת בסביבה בהירה.

כאשר חיישן האור אינו דוגם ערכים גבוהים נורת הלבד תפעל על פי הדגימות מחיישן הלחות.

דגימות חיישן הלחות עולות ככל שיש פחות מים בין שני המזלגות שלו (מודד התנגדות), כך שכאשר חיישן הלחות מחוץ למים המדידות גבוהות ונורת הלבד תתריע על מצב קריטי לצמח שדורש מים ותהבהב במהירות.

כאשר חיישן הלחות נמצא במים נורת הלבד תגביר את עוצמתה ככל שכמות המים יורדת, כאשר ישנה כמות מספקת של מים הנורה תישאר כבויה וככל שכמות המים תרד מתחת לסף מסוים עצמותה תתגבר עד אשר תגיע למצב הקריטי של ההבהוב.

על מנת להפסיק את פעולת המודל נדרש ללחוץ על כפתור SW3, לחיצה זאת תפסיק את פעולת החיישנים על מנת לחסוך בחשמל ותשאיר את המודול זמין אך ורק ללחיצת כפתור נוספת אשר תחזיר את החיישנים לפעולה.

המודול ידפיס בעזרת UART לטרמינל הודעות על מצב פעולתו בהתאם להדלקה וכיבוי החיישנים, וכאשר נקלט הרבה אור.

כמו כן לחיצה על כפתור SW2 ידפיס לטרמינל את מצב המשימות (פועלות/מושקות).

5 סיכום אישי

הפרויקט היה עבורנו חוויה מאוד מאתגרת ומלמדת. זו הפעם הראשונה שבה התמודדנו עם פרויקט שהתחלנו אותו מאפס. נהנינו מאוד מהעבודה עם הרכיבים ברמה הנמוכה ביותר ולמדנו מכך המון דבר אשר בוודאות יעזור לנו בהמשך הדרך. הפרויקט הצריך מאתנו לשבת ולקרוא את כל ה- Data sheet על מנת להבין כיצד להפעיל כל חיישן וחיישן והצלחנו להתמודד עם זה בהצלחה רבה. כתיבת התאוריה של כל פרוטוקול גרמה לנו להבין לעומק את הפרוטוקולים ולדעת כיצד להשתמש בהם, וכך הבנו איזה קלט ופלט כל אחד מהם נדרש לקבל. היה לנו קשה מאוד לקרוא נתונים מחיישן האור TSL2561 אך לאחר קריאה מעמיקה ב-Data sheet שלו לבסוף הצלחנו. החלק המורכב והקשה עבורנו היה לגרום לכל החלקים הקטנים שבנינו לעבוד כמערכת אחת גדולה ויציבה, נתקלנו בהמון קשיים גם מכיוון שרמת שפת c איננה ברמה מספיק טובה אצלנו וגם כי אין הרבה חומר באינטרנט על הבורד אך לאחר עבודה קשה הצלחנו בהצלחה רבה. היה חסר לנו מאוד חיישן שמדבר UART, מכיוון שכן השתמשנו בפרוטוקול אך ללא חיווי פיסי ממשי.

היה מעניין ללמוד על ה UART ולגלות שאפשר באמצעות שליחה של פקודה דרכו לנקות את הטרמינל. על מנת לשפר את הפרויקט בעתיד לדעתנו כדאי לתת יותר תרגולים בכיתה ובמיוחד תרגולים עם חיישנים חיצוניים. להשתמש בבורד שיש עליו יותר חומר באינטרנט.

6 ביבליוגרפיה

- [1] sparkfun, "I2C," [Online]. Available:
<https://learn.sparkfun.com/tutorials/i2c>. [Accessed 21 February 2018].
- [2] TAOS059N, TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER,
TEXAS: TAOS, 2009.
- [3] Wikipedia, "Pulse-width modulation," 9 January 2018. [Online]. Available:
https://en.wikipedia.org/wiki/Pulse-width_modulation. [Accessed 22
February 2018].
- [4] A. "RC radio control protocols explained: PWM, PPM, PCM, SBUS, IBUS
DSMX, DSM2," 22 December 2017. [Online]. Available:
[https://www.dronetrest.com/t/rc-radio-control-protocols-explained-pwm-
ppm-pcm-sbus-ibus-dsmx-dsm2/1357](https://www.dronetrest.com/t/rc-radio-control-protocols-explained-pwm-ppm-pcm-sbus-ibus-dsmx-dsm2/1357). [Accessed 22 February 2018].
- [5] National Instruments, "What is a Pulse Width Modulation (PWM) Signal and
What is it Used For?," 12 February 2018. [Online]. Available:
[https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019Ok
FSAU](https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019OkFSAU). [Accessed 22 February 2018].
- [6] Texas Instruments, "README," 2017. [Online]. Available:
[http://dev.ti.com/tirex/content/simplelink_cc32xx_sdk_1_60_00_04/examp
les/nortos/CC3220S_LAUNCHXL/drivers/pwmlcd/README.html](http://dev.ti.com/tirex/content/simplelink_cc32xx_sdk_1_60_00_04/examples/nortos/CC3220S_LAUNCHXL/drivers/pwmlcd/README.html).
[Accessed 25 February 2018].
- [7] Wikipedia, "Universal asynchronous receiver-transmitter," November 2010.
[Online]. Available:
https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter.
[Accessed 22 February 2018].
- [8] Circuit Basics, "BASICS OF UART COMMUNICATION," [Online].
Available: <http://www.circuitbasics.com/basics-uart-communication/>.
[Accessed 22 February 2018].

- [9] d. "Chapter 20: Analog to Digital Conversion," 5 September 2013. [Online]. Available:
<https://wiki.analog.com/university/courses/electronics/text/chapter-20>.
[Accessed 22 February 2018].

- [10] onmyphd, "Analog-Digital Converters," [Online]. Available:
<http://www.onmyphd.com/?p=analog.digital.converter>. [Accessed 22 February 2018].

- [11] "FC-28 moisture sensor datasheet".

- [12] P. Waher, "MQTT (MQ Telemetry Transport)," February 2018. [Online]. Available: <http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport>. [Accessed 24 February 2018].

- [13] TEXAS INSTRUMENTS, Technical Reference Manual, 2017.

- [14] X. Gouchet, "Ted (Text Editor)," 21 November 2012. [Online]. Available: <https://play.google.com/store/apps/details?id=fr.xgouchet.texteditor&hl=en>.

- [15] M. Backes, S. Bugiel, E. Derr, P. McDaniel, D. Ocateau and S. Weisgerber, "On Demystifying the Android Application Framework: Re-Visiting Android Permission Specification Analysis," in *25th USENIX Security Symposium (USENIX Security 16)*, Austin, TX, 2016.