

אוניברסיטת תל אביב

הפקולטה להנדסת חשמל



מטע"ד אלחוטי לרחפן לדגימת מצלמה וחיישן אינרציאלי בזמן אמת

Real Time Wireless Drone's Payload for Sampling Images and IMU data

פרויקט גמר לתואר שני בהנדסת חשמל

ע"י

דן בלנארו

מנחה הפרויקט : מר יהונתן מנדל

המעבדה לרחפנים אוטונומיים באוניברסיטת תל-אביב

דצמבר 2020

1 תוכן עניינים

1	תוכן עניינים	1
2	מבוא	1
3	מטרות הפרויקט	2
4	איפיון דרישות המערכת	3
4	בחירת הרכיבים	3.1
5	סקר שוק	3.2
6	מצלמה	3.3
6	חיישן IMU	3.4
6	המודול האלחוטי	3.5
7	הבקר	3.6
7	חיבורים ואספקות מתחים	3.7
8	סקירת רכיבי הפרויקט	3.8
9	תכן מערכת	4
9	תרשים בלוקים:	4.1
10	הבקר - STMICRO STM32H743	4.2
11	BSP והקצאת GPIOs - BOARD SUPPORT PACKAGE	4.2.1
13	מפת חלוקת שעון מרכזי	4.2.2
14	פריפריות בשימוש	4.2.3
19	מצלמה - ON SEMICONDUCTOR MT9V034	4.3
20	מודול ה - WIFI - MICROCHIP ATSINC1500	4.4
21	החיישן האינרציאלי - IMU - INVENSENSE MPU6050	4.5
22	ארכיטקטורת תוכנה	5
22	ארכיטקטורה	5.1
22	מבנה שכבות הקוד	5.1.1
25	תכנת תחנה קרקעית	5.2
26	ממשקי DEBUG, צריבה ושימוש במערכת	5.3
27	תוצאות	6
27	תצורת המדידה	6.1
27	תוצאות	6.2
28	סיכום	7
28	סיכום	7.1
28	עבודות המשך	7.2
29	נספחים	8
29	תמונת נוספות של המערכת	8.1
30	ביבליוגרפיה	8.2

רחפנים כיום מנווטים באמצעות עיבוד החיישנים שמותקנים עליהם. אם עבור רחפנים בסביבות חיצוניות (outdoor) משתמשים בחיישנים כגון GPS, הרי שלרחפנים בסביבות סגורות (Indoor) לא ניתן לשתמש בחיישן כזה.

הפרויקט עוסק במימוש מטע"ד (payload) אלחוטי וקומפקטי לרחפן, למטרות חישה ודגימה של רכיבים לטיוב הניווט באמצעות אלגוריתם SLAM.

לטובת הנ"ל, ניתן להשתמש בחיישני עזר נוספים, כגון מצלמה - להתמצאות במרחב, ובחיישנים אינרציאליים (IMU) - שיכול להשלים מידע לאלגוריתמי עיבוד תמונה בנוגע למהירות התנועה וקצב התנועה של הרחפן. באמצעות היתוך מידע בין 2 החיישנים, ניתן להשיג מידע חזותי-אינרציאלי ולבצע מיפוי והתמצאות של המרחב (SLAM) ^[1].

רחפנים שמסוגלים לנווט בסביבה סגורה יכולים להביא תועלת רבה במקום שבה רחפנים לסביבות חיצוניות מביאים – לדוגמא ליישומי בנייה, חקלאות, רפואי, מסחרי, צבאי, קולנוע וגם לפנאי.

העבודה כאן מציגה שיתוף פעולה במעבדה לרחפנים אוטונומיים באוניברסיטת תל אביב TAU-ADL, על מנת לממש את מטרות הפרויקט

- סקר ספרות
- הבנת דרישות המערכת
- תכן על
- סקר שוק ובחירת רכיבים
- מימוש החומרה והתכנה לפרויקט
- חקר ביצועים
- סיכום

2 מטרות הפרויקט

1. גודל קטן
2. צריכת זרם מינימלית
3. תקשורת באמצעים סטנדרטיים לתחנת קרקע – Wifi, BT, 900MHz dongles etc.
4. שליחת המידע עם חותמות זמן ברזולוציה של כ 1msec
- 4.1. מדיד אינרצילי IMU ב 6 צירים - תאוצה קווית בצירים x, y, z, תאוצה זוויתית בצירים x, y, z
- 4.2. תמונה ממצלמה (Grayscale / RGB) – בקצב 20FPS
- 4.3. המידע יישלח בשיהוי של כ 150msec
5. אפשרות להריץ את תחנת הקרקע ע"י PC עם מערכת הפעלה Ubuntu 18.04
6. שמירת המידע בתיקה ייעודית על מנת לחבר אח"כ את תוצרי המערכת לסביבת ROS

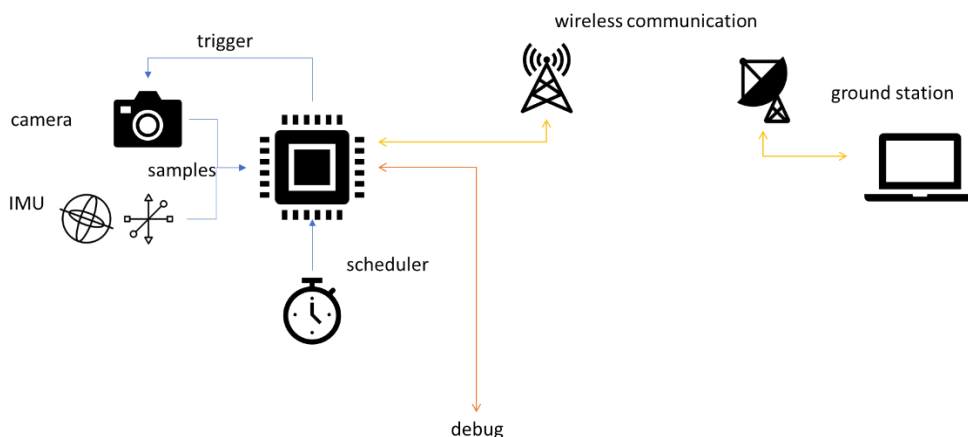
3 איפיון דרישות המערכת

3.1 בחירת הרכיבים

עלינו לבחור ארכיטקטורה שתאפשר מימוש גמיש ויעיל מבחינת משאבים של דרישות הפרויקט. המערכת תורכב מתחנת קרקע, וממטע"ד שיוטקן על הרחפן. בצד הבית לרשותנו מחשב PC, עם תקשורת אלחוטיות סטנדרטיות (Bluetooth, Wifi). אילוץ התקשורת הנחה בבחירת הרכיבים למטע"ד.

למטע"ד נבחר יחידת עיבוד מרכזית (CPU) שתנהל את לוגיקת דגימת החיישנים – מצלמה, מדיד אינרצילי, תזמון את דגימתם ותנהל את לוגיקת התקשורת האלחוטית.

הדרישה ליחידות המטע"ד הינה שימוש ברכיבים סטנדרטיים מבחינת Time To Market, היתכנות מענה לאילוצי המערכת, יכולות debug גבוהות ותמיכה בקהילות פיתוח עם עדיפות לתשתיות קוד-פתוח.



3.2 סקר שוק

טרם תהליך הפיתוח ערכתי סקר שוק על תשתיות ומוצרים שיתנו מענה לדרישות הפרויקט, אסכם כאן את החומרות שמצאתי ואת יתרונן וחסרונן למימוש המשימה.

לסיכום, נבחרה חומרת OpenMV H7, שאפרט את תהליך התאמת החומרה למשימה בהמשך.

model	features	Pros	Cons
Bitcraze Crazyflie 2.1	Small scale drone hardware with wireless communication and programming capabilities. Contains IMU on board. STM32F405 MCU (Cortex-M4, 168MHz, 192kb SRAM)	Small scale factor, open-source community	Not exporting I/F for external camera. The Bluetooth comm might be a little bit slow for our application.
3DR Pixhawk	3 IMU on board, wireless communication is based on ESP32 module (Wifi) STM32F427 MCU (Cortex-M4, 180MHz, 256kb SRAM)	The hardware is versatile, open-source repo is well maintained	No JTAG interface for real time debug No dedicated digital camera I/F. Not compact.
Beagle Bone Blue	Board containing an cortex a8, comm interfaces alongside Wifi & BT OSD3358 (TI Cortex-A8 512MB DDR). No IMU on board.	Versatile, open-source, heavy lifter. No dedicated I/F for camera – a USB camera can be used with proper drivers	The use of an embedded Linux OS might be a little
ESP32 Camera Module Development Board (OV2640)	An ESP8266 based board (160MHz, 160KB RAM) Has Wifi integrated! But the debug interfaces might be difficult!	Open-source, the wifi is built into the MCU.	No dedicated camera port, utilizing an SPI might be slow for this application. The use of SPI camera might be a limitation for future design. Debug interface is not convenient for real-time debug
OpenMV H7	Open-source small scale factor hardware and firmware. A product dedicated for image processing applications. User can utilize the hardware as the product is built as shields (IMU, Wireless communication, Camera)	Contains versatile hardware and cameras. Well maintained user forums and repo	Not all MCU's GPIOs are available to the user via standard headers.

3.3 מצלמה

בחרנו מצלמת CMOS עם חשיפה של Global Shutter, על מנת להבטיח חשיפה אחידה של כל שורות הפיקסלים. השיקול נבע מהתאמת המצלמה לרחפן - פלטפורמה שזזה במהירות, והימנעות ממריחות בתמונה בזמן החשיפה, כמו הדוגמא בתמונה הבאה:



Figure 1 : rolling vs. global shutter

למצלמה יש חיישן בגודל 1/3inch עם רזולוציה מקסימלית של עד 752x480 פיקסלים, יכולת להפיק 60fps ברזולוציה מקסימלית, עם רזולוציה של 10bit לפיקסל בגווני אפור. כמו כן, המצלמה מנגישה רגל strobe על מנת לאפשר לבקר לתזמן את רגע לכידת התמונה ובכך להבטיח את עדכניות התמונה.

המצלמה מאפשרת אפשרות לחשיפה אוטומטית, וזה המצב שבו עבדנו, עם קצב שיעון מקסימלי של 27MHz.

נפחי פלט התמונה ישמשו אותנו בבחירת תצורת המערכת, מבחינת רזולוציה ו fps מובאים להלן:

Resolution	width	height	fps	Mbit/sec
Full resolution	752	480	20	57.8
QVGA	320	240	20	12.3

3.4 חיישן IMU

נבחר רכיב לדגימת תאוצות 6 צירים (3 רכיבי תאוצה בצירים ליניאריים ו3 רכיבי תאוצה בכיוון סיבובי). רכיבי התאוצה יאפשרו לאלגוריתם העיבוד על תחנת הקרקע, להבין תנועות יחסיות של הרחפן, ולטייב את עקיבת התמונה.

רזולוציות דגימה אופייניות של הרחפן תהיה כ 250dps לתאוצה סיבובית, וכ $\pm 2g$ לתאוצה קווית עבור כל ציר.

3.5 המודול האלחוטי

על המודול האלחוטי להיות בעל פרוטוקול תקשורת סטנדרטי ל PC (Wifi / Bluetooth), עם ממשק תקשורת סריאלי לבקר המנחה.

נבחר לממש את התקשורת ע"י Wifi בגלל יתרונות רוחב הסרט, פשטות החיבור אליו והיכולת העתידית לאגור מידע ממספר מטע"דים בו-זמנית ע"י התחברות של מספר מטע"דים לאותו Access Point.

המודול האלחוטי צריך להתחבר לתחנה קרקעית ע"י PC, לקבל ולשלוח את המידע מצד המטע"ד.

3.6 הבקר

לבקר יש מספר משימות באפליקציה שלנו :

- ניהול החיישנים והגדרותיהם
 - סנכרון זמן דגימת החיישנים
 - ניהול פרוטוקול תקשורת שמוצמדת אליו חותמת זמן
 - ניהול תקשורת אלחוטית עם תחנת הקרקע
 - ריכוז המידע מהחיישנים, ודחיסתו
 - קיום ממשקי תקשורת ס ממשקי תקשורת SPI/ I2C לחיישנים, UART לצורך הודעות debug למשתמש
 - ממשק SWD/JTAG לצריבה ודיבוג של התכנה בזמן אמת
 - יש יתרון לבקר עם ממשק תקשורת סטנדרטי למצלמה, דוגמת LVDS או ממשק מקבילי
- לבקר צריכים להיות ממשקי תקשורת שמתאימים לרכיבים שנבחרו, ויכולת לתת חותמת זמן לכל מידע שמגיע מהחיישנים, לטובת עיבוד המשך בתחנת הקרקע.
- מכיוון שמערכת הפעלה מוסיפה עוד תקורה בנפח הקוד, בריצה ובזמן ניהול האירועים שלה, ועל מנת לאפשר פיתוח מהיר ואמין, עם מעטפת ביצועים real time, נבחר לממש את האפליקציה ללא מערכת הפעלה, אלא בכתיבה של התכנה bare-metal לבקר.

3.7 חיבורים ואספקות מתחים





למערכת צריכה להיות אפשרות להיות מוזנת ממתחים סטנדרטיים, לדוגמת סוללת Lipo 3.7V, או מחיבור USB של 5V.

3.8 סקירת רכיבי הפרויקט

מודול ה-OpenMV H7 נותן לנו מענה חומרתי לרוב דרישות הפרויקט, ובפרק הבא נסקור את השינויים שבוצעו במודולים הקיימים על מנת לתת מענה מיטבי לפרויקט.

כל רכיבי המערכת עמידים לעבודה בטמפרטורת חדר, (-20 to $+70$ °C). ניתן לראות גם לפי התמונות שהם בנויים בצורת shields שיאפשרו ממשק נוח לחיבור בין החלקים השונים.

לני"ל שידכנו צורב מדגם stlink v3 של חברת STmicro – שנתן לנו ממשק JTAG וחיבור UART למטרות דיבאג במערכת.

Part	Model	Description	Size	Weight
MCU 	STmicro STM32H743	Implements micropython, includes DCMI I/F, RGB LED, up to 480MHz Idd = 170mA @ 3.3V Regulator – up to 5.5v, 1A	45x36x30mm Incl. camera	19gr
Camera 	ON MT9V034	HFOV = 70.8° VFOV = 55.6° Idd = 200mA max	35x25x28mm	10gr
IMU 	Invensense MPU6050	Idd <4mA	20x25x7mm	?
Wireless Communication 	Microchip AT-winc1500	Up to 12.5Mbps Idd =100 mA @ 3.3V	36x27x20mm	7g

נציג את בחירת הפרמטרים ונקודות העבודה לכל מודול כמו גם הממשקים בין הרכיבים השונים שמביאים נקודת עבודה אופטימלית לבעית הפרויקט.

4.1 תרשים בלוקים:

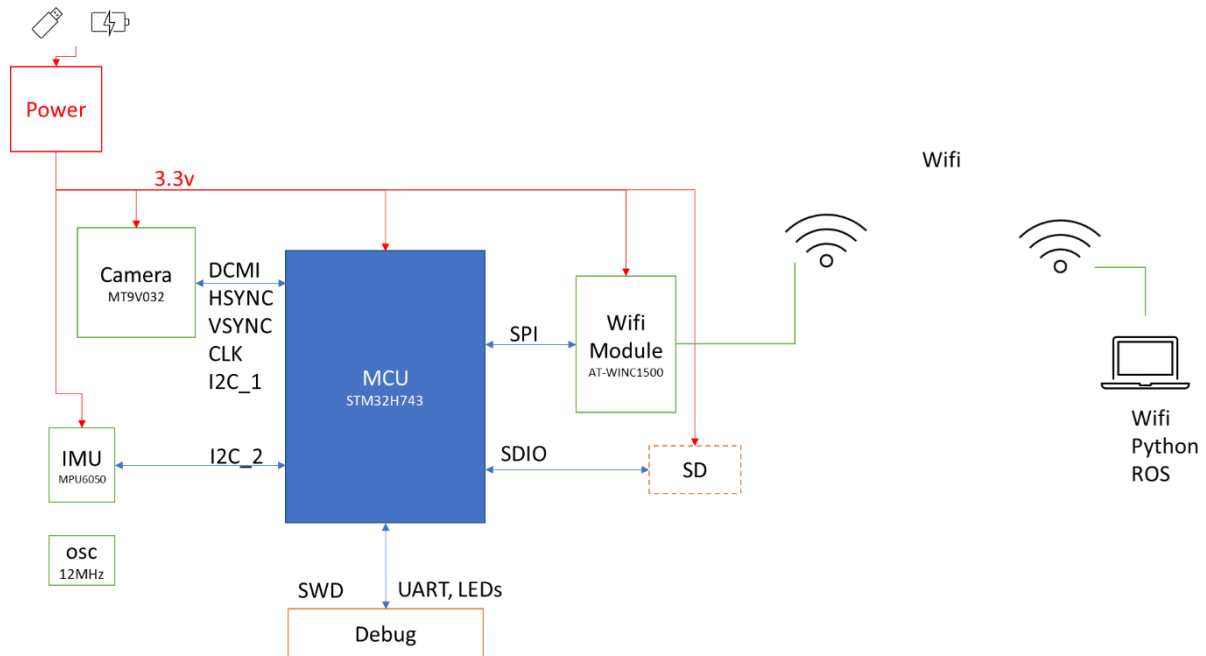


Figure: 2 block diagram

יחידת אספקת המתחים מייצבת מתח 3.3V לצרכנים במעגל. ניתן להזין את המעגל באמצעות USB או סוללה חיצונית.

הבקר מקבל שעון ממתנד חיצוני של 12MHz לצורך הריצה שלו. כמו כן, הבקר מתחבר דרך ממשקי תקשורת שונים לצרכנים במעגל-

למודול ה Wifi ממומש חיבור תקשורת SPI תקני (MISO, MOSI, CLK, CS).

למודול ה IMU מתחברים באמצעות I2C/

הממשק למצלמה כולל bus - ממשק מקבילי ברוחב 8 ביט, שנקרא DCMI, כמו גם קווי סינכרון לתחילת קליטה של שורה/עמודה חדשה, אות שעון למצלמה, וממשק I2C לקינפוג ושינוי פרמטרים במצלמה.

4.2 הבקר - STMICRO STM32H743

הבקר משמש אותנו להגדרת הפריפריות במעגל לנקי העבודה הרלוונטית, התקשורת ביניהן, תזמון האירועים, ניהול הפרוטוקול וניהול של המידע הנאגר.

לבקר יש אילוץ זכרון RAM, כאשר בהתמודדות עם תפיסה של מטריצות פיקסלים גדולות, הופכים להיות מגבלת מערכת שיש להתמודד איתה בהיבט האפליקציה. כמו כן מניתוח של בעית זמן השליחה, רוחב הסרט הנתון והקצבים שהתבקשו לעמוד בהם, הבנו שעלינו לממש דחיסה לתמונה, לכן מימשנו כאן דוחס jpeg כפי שיפורט בפרקי המימוש בהמשך.

תכן הלוגיקה והעיבוד של הבקר, צריכים להתכנס מבחינת ציר הזמן לצילום, דגימה, עיבוד ושליחה לתחנה הקרקעית – על מנת לעמוד בדרישות הפרויקט של צילום ושליחה בreal time.

לצד כל חלק נסקור את הזמן האופייני שלוקח לכל חלק ונחשב את תקציב הזמנים.

t_{img_acq} - משך הזמן לחשיפת חיישן המצלמה והעברת התמונה לבקר

t_{img_comp} - משך הזמן לדחיסת התמונה

t_{imu} - משך הזמן לקבלת מידע מהחיישן האינרציאלי

t_{comm} - משך הזמן לשליחת המידע

להלן דיאגרמת בלוקים של פריפריות הבקר, גישות הזכרון והממשקים האפשריים לבSP.

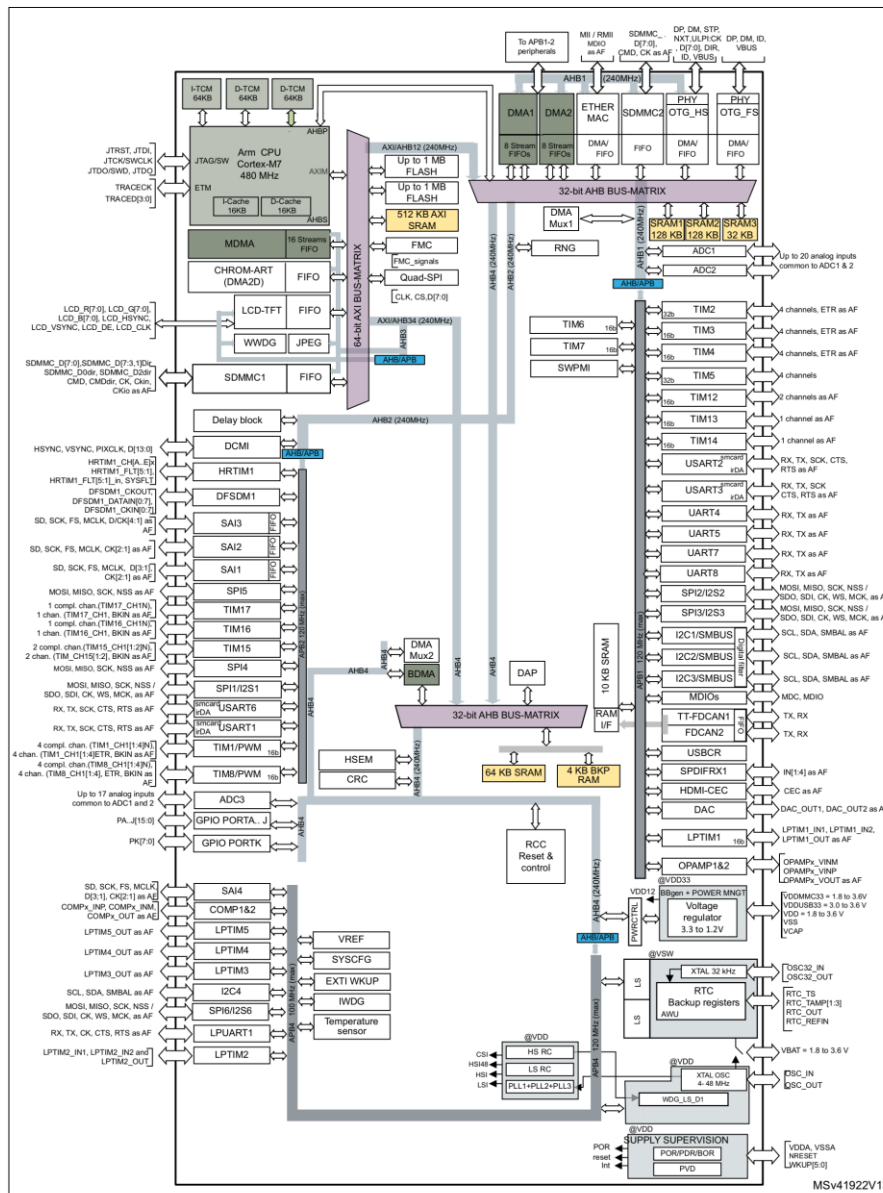


Figure : 3 STM32H743 block diagram

Board Support Package – GPIOs והקצאת BSP 4.2.1

להלן pinout שבחרנו למימוש הבקר ולהקצאת ה GPIO – General Purpose Input/Output.

הממשקים החשמליים (אספקות ואדמות)- הם אילוץ במימוש, ושאר ה pinout אליו צריכים להתכנס אליו לפי שרטוטי schematics של חומרת ה OpenMV, מפורט בתמונה הבאה.

כמו כן, התוספת של חיישן ה IMU דרך חיבורי header, אילצה אותנו לפרוס קווים נוספים לטובת UART, באמצעות הלחמה של maget wire ע"י רגלי הבקר, באופן שיאפשר שימוש במקביל גם בממשק ה debug של ה UART וגם על מנת לתקשר עם רכיב ה IMU. הלחמת קווי התקשורת בוצעה על רגלי הבקר (23 PA1, 22 PA0), בנוסף לחיבור קו אדמה לטובת התקשורת.



Figure : 4 Debug UART wiring

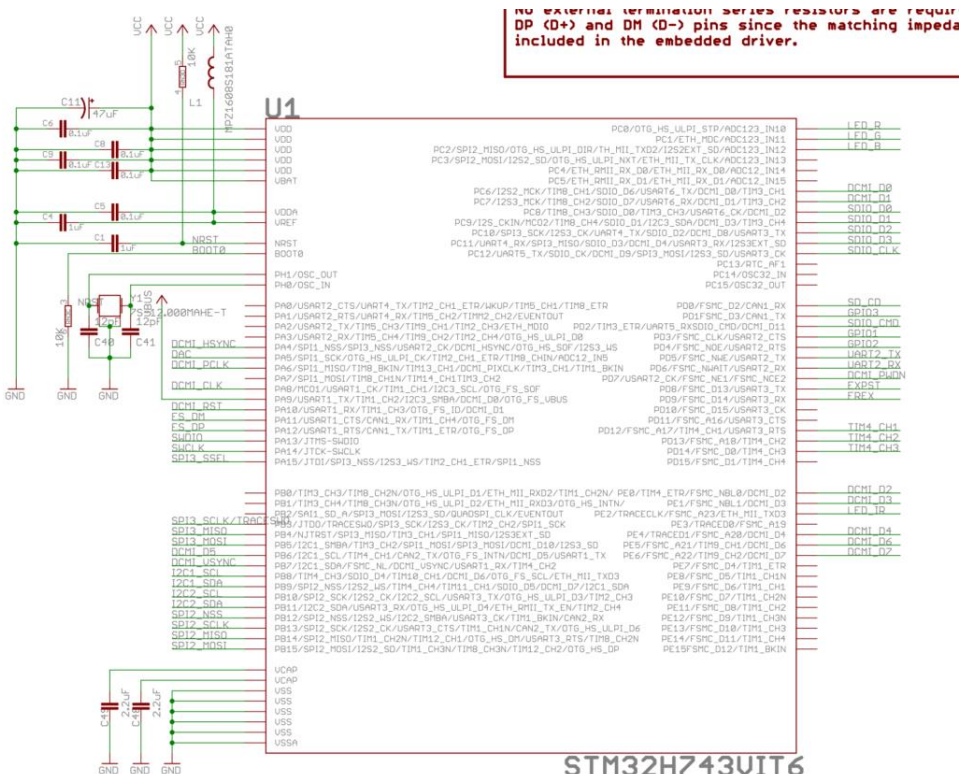


Figure5 : OpenMV H7 (c) schematics

4.2.2 מפת חלוקת שעון מרכזי

באמצעות סביבת ה CubeMx הגדרנו את חלוקת השעונים לחלקי הבקר. בגלל סיבות שיפורטו בהמשך, השתמשנו במכפילי שעון ומחלקים כדי לעבוד בקצב העבודה המקסימלי שהבקר מאפשר – 480MHz.

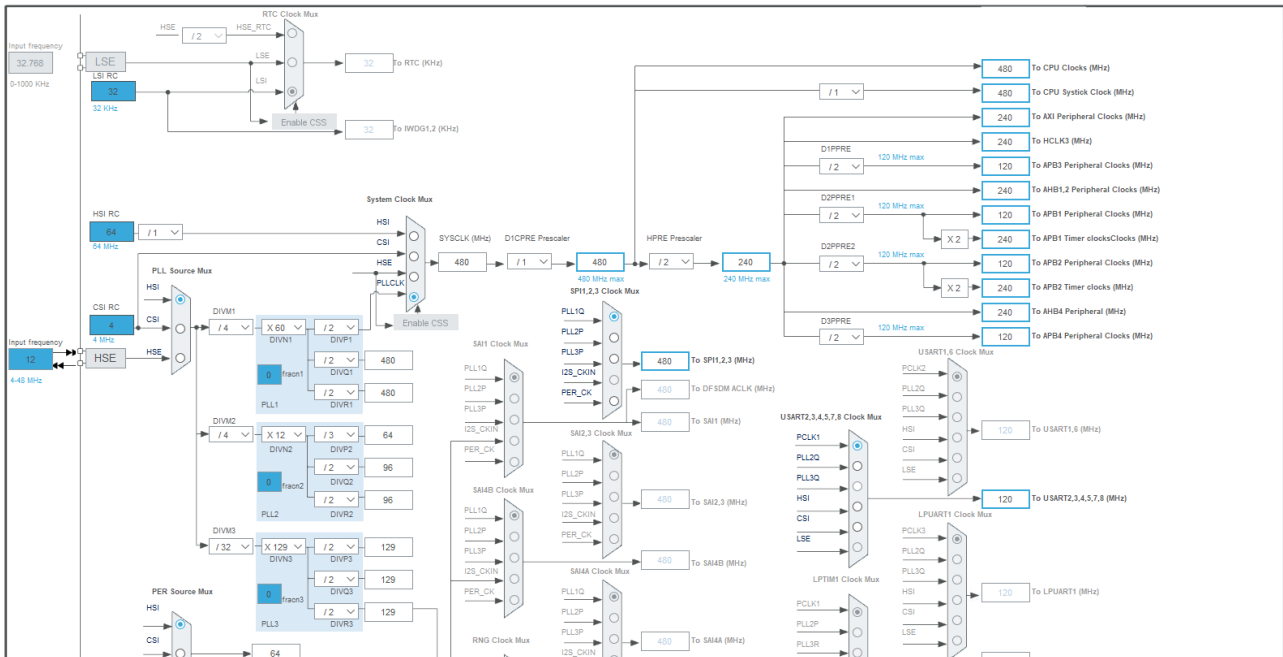


Figure 7: CPU clocks regime

4.2.3 פריפריית שימוש להלן ניצול משאבי הבקר



Figure 8: CPU peripherals

4.2.3.1 MDMA /DMA

זוהי פריפריית שמאפשרת גישת זכרון ישירה בין אזורי זכרון שונים בבקר, עם העברת מידע בכלי חומרת, שאינו מעכב את ריצת התכנה. באפליקציה שלנו - מוגדר כגישה ישירה בין זכרון פריפריית תפיסת המצלמה (DCMI) לזכרון ה-RAM, כמו כן בין ה-RAM לפריפריית דוחס התמונות.

היכולת הזו מאפשרת לבקר לתקשר במהירות וביעילות ולקבל מידע מהפריפריית השונות שמפורטות כאן, מבלי לעכב באופן משמעותי את התכנה.

לדוגמא – דחיסת התמונה לפורמט JPEG ממומשת אצלנו ע"י העברת 2 שורות בכל פעם לפריפריית הדחיסה חומרתית והמתנה לביצוע עיבוד הפריפרייה. כאשר הפריפרייה מסיימת את החישוב והדחיסה על 2 השורות, היא מעבירה באמצעות ה-DMA את המידע למרחב הזכרון ב-RAM (Random Access Memory) שהוגדר בבקר להמשך העבודה, הבקר מקבל פסיקה (interrupt) ובכך מקבל טריגר להמשיך להעביר את נתח התמונה הבא לעיבוד הפריפרייה החומרתית וכך חוזר חלילה, עד לסיום הדחיסה.

עוד שימוש ל-DMA אצלנו, הוא להעברת המידע מהמצלמה לזכרון הבקר.

4.2.3.2 Cache and MPU




לטובת תקשורת והעברת מידע בין אזורי זכרון שונים, הוגדרה גישה מותרת לאזורי זכרון בהתאם להגדרות ה-DCMI וה-JPEG לפי אילוצי הבקר.

4.2.3.3 JPEG Encoder

כפי שתואר בפרקים הקודמים, דרישות הפרויקט שמבקשות תמונה ברזולוציה טובה דיה (נניח 320x240), בקצב של 20fps, מאלצת רוחב סרט גדול מאוד של כ-12Mbps. מאוחר יותר, התברר שהרוחב סרט האפקטיבי שמצליחים להוציא מרכיב התקשורת האלחוטית הוא 3Mbps, עובדה שהפכה את השימוש בדוחס להכרח.

הפרמטרים לדוחס התמונה מאפשרים שימוש במרחב צבעים שונה להלן דוגמא של תמונה שנדחסה באמצעות פריפריית הדוחס החומרתית שבבקר, תחת אילוצי השימוש שבה:

Parameter	Value
JPEG_RGB_FORMAT	JPEG_RGB888
JPEG_CHROMA_SAMPLING	JPEG_420_SUBSAMPLING
JPEG_YCBCR_COLORSPACE	JPEG_COLOR_SPACE
JPEG_IMAGE_QUALITY	variable

			
W	400	400	400
H	240	240	240
Size	281KB	10.8KB	64.2KB
Q	-	75	100
Size factor	1	0.0384	0.228

ניתן לראות שאפשר להקטין את התמונה, מבלי לאבד איכות באופן ניכר, בפקטור ~0.04 ביחס לנפח המקורי. הקטנת נפח התמונה, תאפשר להעביר תמונה ב latency נמוך ומבלי לפתוח הרבה נפח.

מבחינת לוח הזמנים לעיבוד, עבור גודל תמונה 640x480, לפי דפי הנתונים יש לקחת $t_{img_comp} = 62msec$ בתקציב הזמנים.

להלן תיאור תהליך דחיסת התמונה בפריפרייה החומרתית שבבקר :

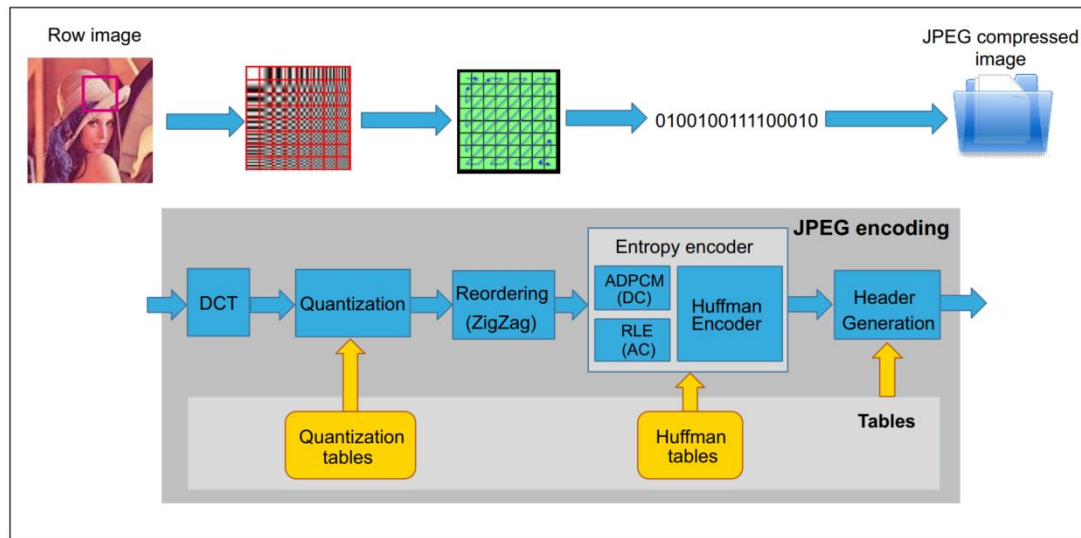


Figure 9: JPEG encoding flow

מהלך הדחיסה בנוי מקבלת מפת ביטים שלמה של התמונה (לדוג' 320×240 פיקסלים), ולאחר מכן קריצה של חלקים מהתמונה והעברתם לדוחס – 16 שורות בכל פעם, המרתם למרחב צבעים YCbCr וליחידות המרה מינימליות (MCU) לטובת דחיסה (DCT, קוונטיזציה, קידוד אנטרופיה, קידוד Huffman), וייצוג קובץ jpeg המתאים.

תהליך הקריצה, הגדרת הפרמטרים לדוחס, וטיפול הפלט הדוחס הם בצד התכנה. תהליך הדחיסה עצמו וההמרה מתבצע באופן חומרתי.

Timers 4.2.3.4

TIM1 – שעון המצלמה, במצב עבודה של OC (מייחצן אות שעון ע"ג רגל מוגדרת בבקר, כקלט למצלמה) לתדר 25MHz.

TIM2 – שעון זמן הדגימה לחיישן ה IMU - מכוון זמן מחזור 1msec (1kHz).

TIM6 – שעון זמן ניהול השליחה למודול ה Wifi - מכוון זמן מחזור 0.25msec (4kHz).

UART 4.2.3.5

משמש אותנו למטרת תקשורת סריאלית בין המחשב ול debug של המערכת - ממשק הודעות טקסטואליות שמוצמט לחותמת זמן לטובת חקר ביצועים של המערכת. מוגדר להיות בקצב 115,200bps. ממשק ה debug יומחש בפרק ביצועי המערכת.

SPI 4.2.3.6

SPI2 – למודול ה Wifi, בקצב 60MHz [הערה: הקצב הזה חורג מהקצב המומלץ של היצרן, שהוא עד 48MHz]

4.2.3.7 Digital Camera Interface – DCMII

הבקר מכיל ממשק למצלמה על מנת לתקשר עם מצלמות חיישני CMOS באמצעות bus ברוחב 8-14bit על מנת לקבל וידאו. הפריפריה מאחדת את הממשק והפרוטוקול לתזמון הפיסלים ביחס לאות השעון PIXCLK שיוצא מהחיישן, כמו גם סנכרון לאותות סימון השורות-עמודות של frame, שקרויים VSYNC ו- HSYNC בהתאמה.

הממשק למצלמה באופן כללי, מורכב מ:

- קווי שליטה למצלמה (טריגרים, reset)
- קווי מידע ותקשורת מחיישן המצלמה (המידע מחיישן ה-ADC שדוגם את הפיקסלים). הסבר על הפרוטוקול מפורט בפרק המצלמה.
- קווי תקשורת להגדרת פרמטרים לעבודת המצלמה (זמן חשיפה, רזולוציה)

אצלנו הממשק מוגדר ברוחב 8bit, עם סנכרון מצד הבקר. עם מימוש העברת frame באמצעות DMA2.

השימוש ב-DMA מאפשר לבקר להעביר את המידע מפריפריית DCMII למרחב הזכרון שמבצע עיבוד על התמונה. פריפריית DCMII נמצאת שייך ל-AHB2, יחד עם DMA1 + DMA2. על מנת להעביר את המידע לאזור הזכרון בו רצה האפליקציה ב- AXI-SRAM, עלינו להגדיר MPU (Memory Protection Unit) עם הרשאה לגישה בין אזורי הזכרון הנ"ל. זמן החשיפה האופייני של מצלמה (~16msec) וזמן השליפה של המידע מהמצלמה הוא זניח ומסתכם בכ- $t_{img_acq} \sim 16.7msec$.

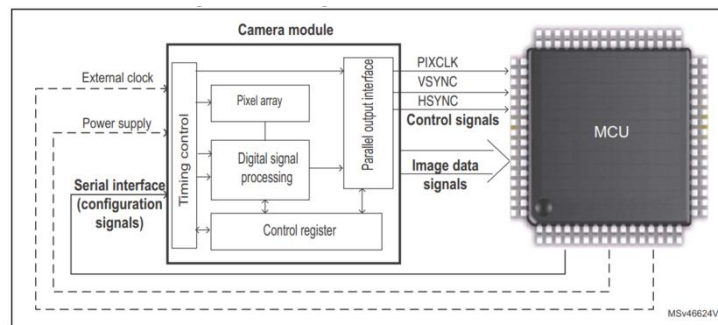


Figure10: MCU-Camera DCMII I/F

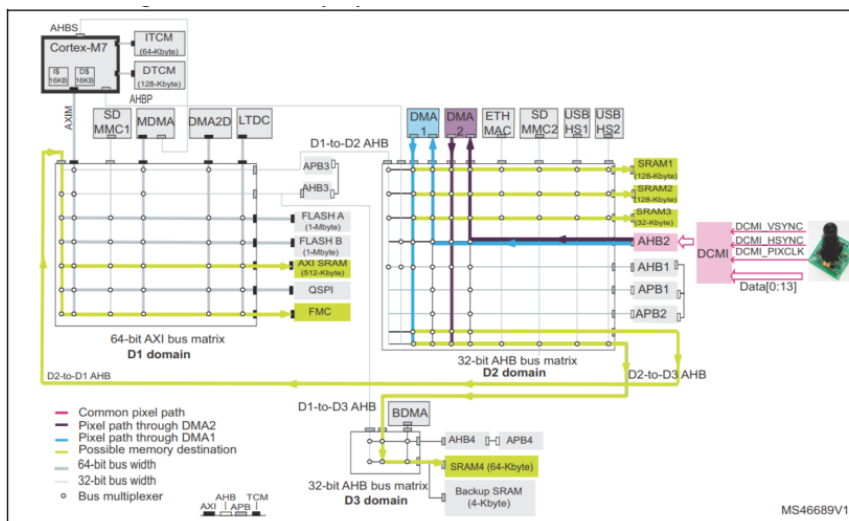


Figure11: DCMII memory access

פלט התמונה שנשמר מהמצלמה, הינו מפת ביטים (bitmap), לפי גודל התמונה שהוגדר, כאשר כל פיקסל בגוון אפור בגודל 8bit לפיקסל. התמונה תישמר בבאפר במרחב ה-AXI_SRAM שיוקצה מראש לגודל התמונה הרצוי. המצלמה אצלנו הוגדרה לרזולוציית תמונה של 320X240 פיקסלים.

4.2.3.8 I2C

I2C1 – עבור מודול המצלמה, בקצב 400KHz

I2C2 – לחיישן IMU, בקצב 400KHz

4.2.3.9 SD + FATFS

למטרות debug וחקר ביצועים, הושמש ממשק לשמירת פלטי המצלמה והדחיסה ע"ג כרטיס SD במערכת קבצים FATFS. השימוש בפיצ'ר הני"ל אפשרי באמצעות הדלקת דגל בקימפול הקוד – בקובץ `camera_mgr.h` - אפשר ה `define` של `SAVE_INPUT_IMG_ON_SD` ו `SAVE_OUTPUT_IMG_ON_SD` יאפשרו את שמירת תמונה הקלט והפלט בהתאמה על כרטיס הSD. לשים לב שהפעולות הני"ל יכולות להשפיע על קצב עיבוד ושליחת הנתונים, לכן יש לעדכןם בהתאם. כמו כן, יש לפתוח את פלט ה `bmp` שנשמר באמצעות סקריפט קצר ב `raw_2plot.py`.

4.3 מצלמה – ON SEMICONDUCTOR MT9V034

המצלמה שבשימוש אצלנו היא מצלמת Global Shutter עם חיישן CMOS בגודל 1/3inch. כמו כן, תת הדגם שאנחנו עובדים איתו מכיל רזולוציה של עד 480x752 בגוויי אפור עם רזולוציה של 10bit דגימת ADC לפיקסל.

ממש המצלמה מובנה באמצעות פריסת המעגל של OpenMV וממומש לפי הדרישה בדפי הנתונים.

רגלי SCLK/SDATA מחווטות לרגלי I2C1 של הבקר.

רגלי ה SYSCLK מחווטות לאות שעון שהבקר מייצר (בתדר 25MHz), כאשר המצלמה תומכת באות שעון עד 27MHz.

ממשק ה Exposure מאפשר שליטה על חשיפת החיישן של המצלמה, אולם לא מימשנו אותו כאן במערכת (ניתן לממש אותו ע"י הגדרת רגל PB4 בבקר וניהול החשיפה ביחס לתפיסת התמונה).

רזולוציית המצלמה נקבעה מטעמי זכרון וקצבים – להיות על (QVGA) 320x240.

הממשק המקבילי של פלט המצלמה – Dout (9:0) נכנס לרגלי ה DCMI שהוגדרו בבקר כאשר מטעמי חומרה ואינטגרציה, ממומשים בפועל רק 8bit של התקשורת, קרי (9:2) Dout. פרוטוקול העברת המידע בין החיישן למצלמה מפורט בתמונה בהמשך.

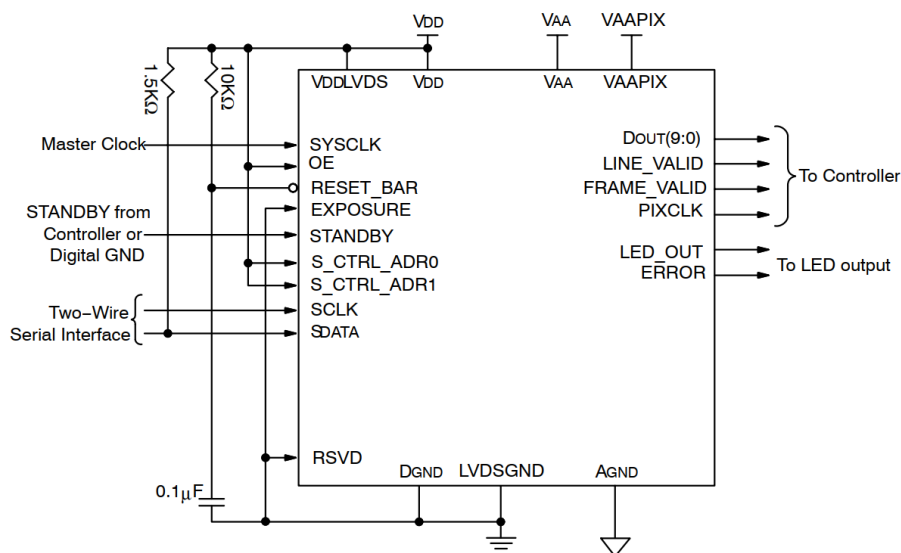


Figure 12 : MT9034V pinout

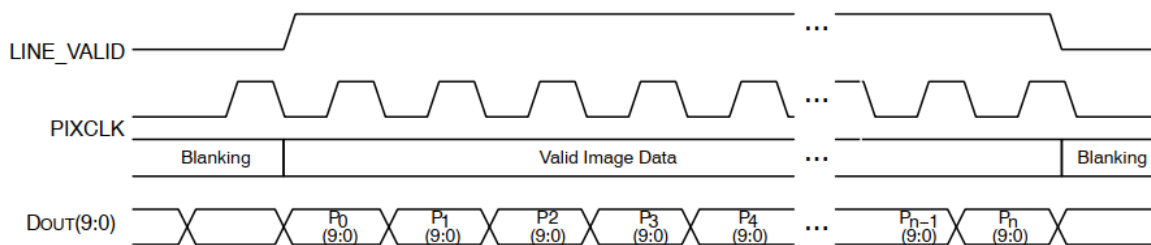


Figure 13 : Pixel data protocol

4.4 מודול ה-WIFI - MICROCHIP ATSINC1500

מודול ה-Wifi מאפשר לנו לקיים תקשורת עם כל מודם Wifi סטנדרטי, במצבי עבודה של Client (STA) או Host (AP).

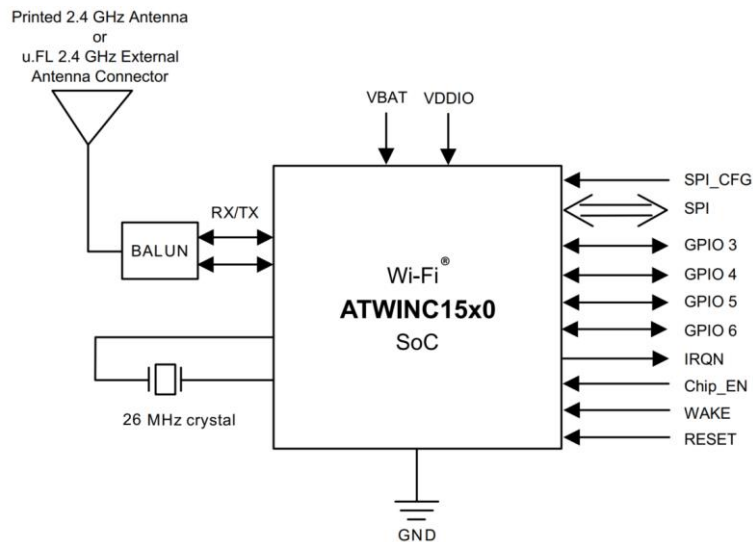


Figure 14: ATWinc1500 diagram

Feature	Value
WLAN standard	802.11b/g/n
Host interface	SPI, 28MHz max
Data rate	802.11b: 1, 2, 5.5, 11Mbps 802.11g: 6, 9, 12, 18, 24, 36, 48, 54Mbps
Sensitivity	-76dBm @ 802.11g 48Mbps OFDM
Output power	18.5dBm max @802.11b 11Mbps
Power consumption	I _v batt = 268mA, I _v ddio = 22mA max @802.11g

4.4.1.1 ביצועי המודול

מהעבודה של השותף שלי לחלק השני של הפרויקט, נמדד כי הביצועים האפקטיביים של המודול הינם כ-3Mbps. עובדה שהפכה את עבודה הדחיסה להכרח, כמו כן, אילצה את גודל התמונה האפקטיבי שאנחנו יכולים לעבוד איתו, להספיק מבחינת תקציב הזמן לדחוס ולשלוח, כאמור 320X240 פיקסלים.

4.5 החיישן האינרציאלי – IMU – INVENSENSE MPU6050

החיישן נועד להקליט תאוצות לינאריות וסיבוביות של הרחפן. החיישן הזה מצטיג ביצועים טובים מבחינת טווח דגימה וצריכות זרמים. כמו כן, חיפשנו רכיב עם דרייברים ותמיכה ברשת, שאכן הרכיב הזה עומד פרמטר.

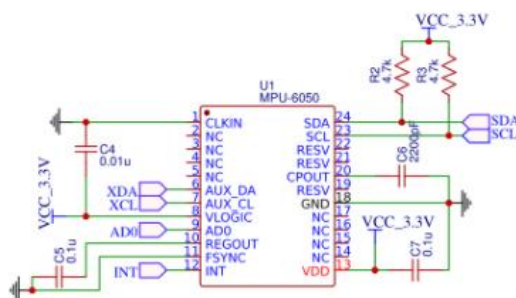


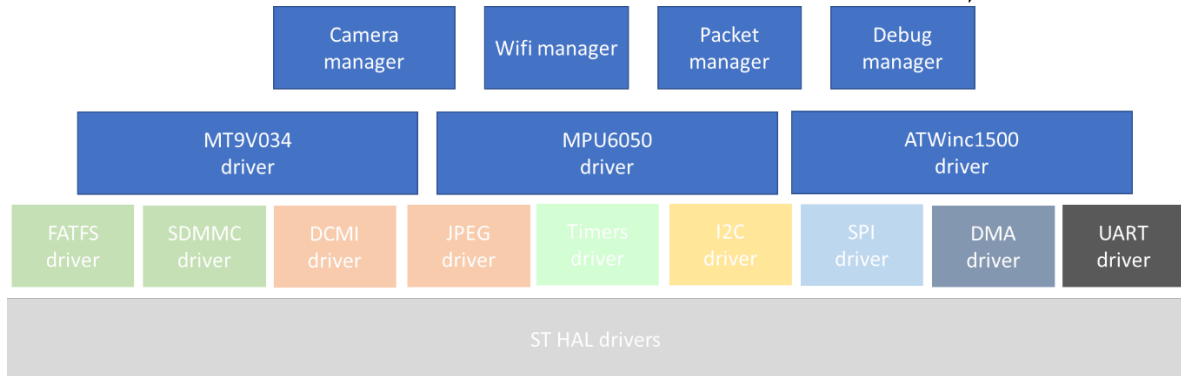
Figure15: MPU 6050 breakout schematic

Feature	Value
Gyro fullscale [dps]	±250 ±500 ±1000 ±2000
Gyro sensitivity [LSB/dps]	131 65.5 32.8 16.4
Gyro output data rate	8KHz max
Accelerometer fullscale [g]	±2 ±4 ±8 ±16
Accelerometer sensitivity [LSB/g]	16,384 8192 4096 2048
Accelerometer output data rate	1KHz max
Digital Output	I ² C
Shock resistant	10,000g
Power consumption	I = 3.8mA, V = 2.375v-3.46v

את החיישן יש להגדיר (config) לאחר עליית המתח לרכיב. יש להגדיר רזולוציה (full scale) עבור gyro ועבור accelerometer, כמו כן יש להגדיר את קצב רענון הנתונים – output data rate. לאחר מכן, ניתן לקרוא באופן יזום בכל רגע נתון את דגימות החיישן, ולאחסן אותן בבקר לצורך שליחה.

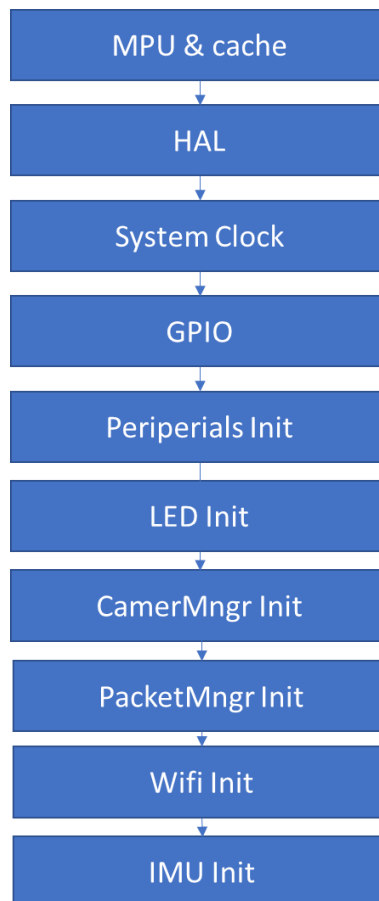
5.1 ארכיטקטורה

5.1.1 מבנה שכבות הקוד



5.1.1.1 שלב האתחול של הבקר

עליית הקוד צריכה להבטיח הדלקת הפריפריות, הקצאת הזכרונות, הגדרות השעונים ואתחול כל מודול חומרתי לפי הפרמטרים שפורטו בפרק הקודם. הקוד מתחיל מהגדרת מרחבי הזכרון בתחילת ריצת הקוד, לאחר מכן אתחול ה-HAL (Hardware Abstraction Layer - שכבת קוד שחברת STMicro מפרסמת כהפשטה של שכבת החומרה בבקר, לדרייברים תכנתיים פשוטים יותר), הגדרות שעונים, ולאחר מכן אתחול של כל שאר הפריפריות והמודולים שלנו לפי השעונים והזכרונות שהוקצו להם.



להלן מצורף תרשים הזמן על תפיסת התמונה, דחיסתה והכנתה לשליחה, בכל פרק זמן של אירוע שליחה שמוגדר אחת ל 60msec כפי ששותפי לעבודה על המוצר הגדיר בקובץ `packet_mng.h`.

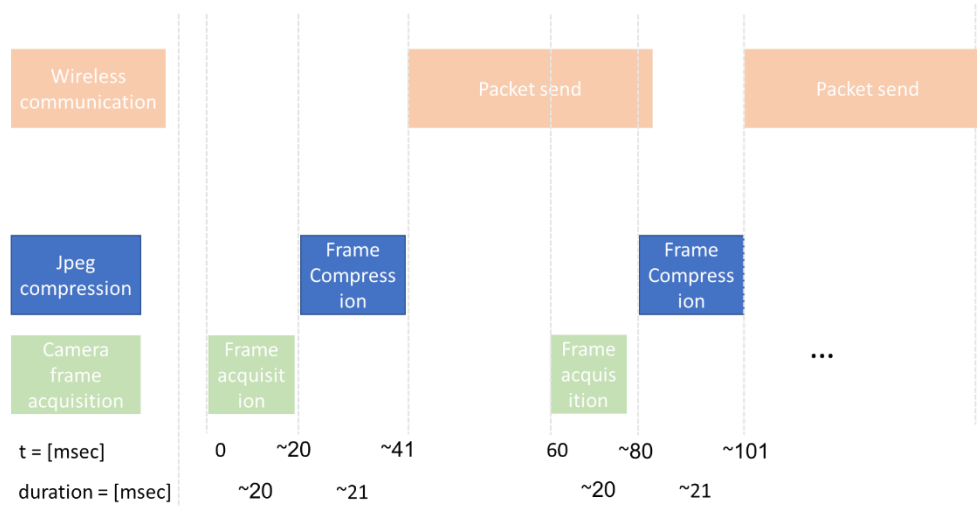


Figure 16 : compression timeline

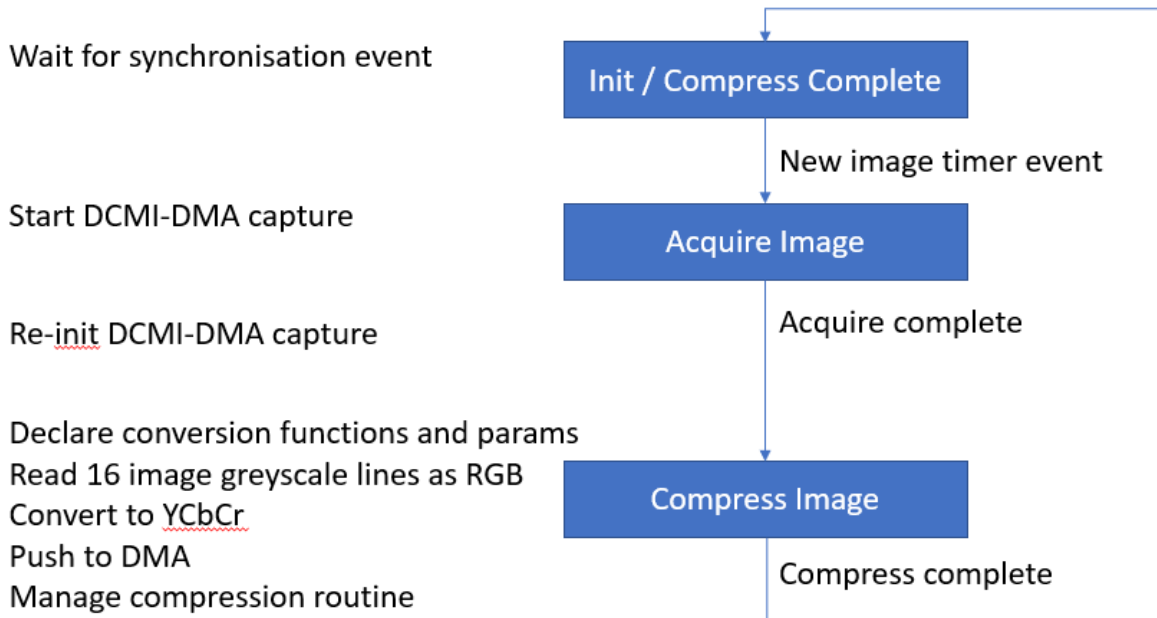


Figure 17 : Image handling routine

מובא כאן פלט debug בזמן אמת של תקשורת מול תחנת קרקע. זהו צילום מסך של תכנת Docklight שמגישה את תקשורת ה UART של הבקר. ליד כל שורה יש חותמת זמן ב msec מרגע עליית הבקר, ובטקסט את האירועים שקורים.

ניתן לראות שהזמן האופייני לתפיסת התמונה ודחיסתה תלוי במאפייני התמונה וכמה היא עשירה בגרדיאנטים זמן אופייני לתפיסת תמונה – עד כ 20msec. זמן אופייני לדחיסת תמונה – כ 21msec.

Parameter	Value
SOF	A message with IMAGE data is being sent to ground station
acq s	Acquisition of new frame from camera started
imu	A message with IMU data is being sent to ground station
acq dur X[msec]	Acquisition of frame took X msec
comp s	JPEG compression started
jout size = X[b], jcmp Y[msec]	JPEG compression took Y msec with output of size X bytes

```

11787<HT>    SOF<CR><LF>
11804<HT>    acq s<CR><LF>
11812<HT>    imu<CR><LF>
11820<HT>    acq dur 15[msec]<CR><LF>
11822<HT>    comp s<CR><LF>
11844<HT>    jout size = 4144 [b]<HT>    jcmp 21[msec]<CR>
<LF>
11848<HT>    SOF<CR><LF>
11864<HT>    acq s<CR><LF>
11872<HT>    imu<CR><LF>
11881<HT>    acq dur 16[msec]<CR><LF>
11883<HT>    comp s<CR><LF>
11905<HT>    jout size = 4144 [b]<HT>    jcmp 21[msec]<CR>
<LF>
11909<HT>    SOF<CR><LF>
11924<HT>    acq s<CR><LF>
11932<HT>    imu<CR><LF>
11942<HT>    acq dur 17[msec]<CR><LF>
11944<HT>    comp s<CR><LF>
11966<HT>    jout size = 4140 [b]<HT>    jcmp 20[msec]<CR>
<LF>
11970<HT>    SOF<CR><LF>
11984<HT>    acq s<CR><LF>
11992<HT>    imu<CR><LF>
12003<HT>    acq dur 18[msec]<CR><LF>
12005<HT>    comp s<CR><LF>
12028<HT>    jout size = 4144 [b]<HT>    jcmp 21[msec]<CR>

```

Figure 18: acquisition and compression debug output

5.2 תכנת תחנה קרקעית

לטובת חקר ביצועים ונוחות הפיתוח, פותח כלי להצגת התמונות בזמן אמת כתוספת לתכנת התחנה הקרקעית שפיתח שותפי לפרויקט.

התצוגה מאפשרת לקבל את התמונה העדכנית ומציגה את ההשהיות בקבלת התמונות.

התכנה, בסביבת פייתון מהווה את הניהול והטריגר לתחילת שליחת המידע מהמטע"ד והעברת המידע למחשב.

לציין שבגלל סיבה שאינה ידועה, החלק העליון והתחתון של התמונה יוצא מעוות, לכן בתקנת הקרקע אנו חותכים חלק מהצד העליון והתחתון של התמונה על מנת להביא פריים תקין לתצוגה.

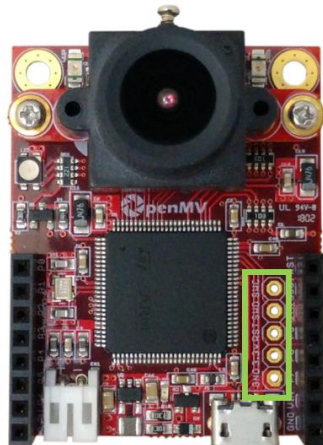


Figure19: benchmark and payload's live view

5.3 ממשקי DEBUG, צריבה ושימוש במערכת

לצורך פיתוח הקוד והאלגוריתם ולהשמשת המערכת השתמשנו בכלים הבאים :

- הזנת מתח, אפשרית באחת מ2 השיטות הבאות :
 - מתח USB באמצעות מחבר micro-USB על הכרטיס (השיטה המומלצת)
 - מתח 3.7v מסוללת LiPo באמצעות מחבר ייעודי ע"ג הכרטיס.
- STM32Cube IDE - compiler ו- debugger מבוסס eclipse לבקרי STM32.
- STLINK V3 – צורב לבקר STM32, מאפשר צריבה דרך JTAG או SWD. בפרויקט הזה השתמשנו בצריבה באמצעות SWD, שאפשר צריבה ע"ג 2 רגליים (SWDIO, SWCLK), ב header ייעודי ע"ג הOpenMV. הממשק לצורב מצריך גם מתן קו לוגי גבוה 3.3V, קו אדמה משותף GND, ואצלנו מושמש גם קו RESET חיצוני.



- ממיר תקשורת USB to UART – לצורך הצגת הודעות מצב מהמערכת. החיבור מורכב מקו אדמה משותף GND, רגל Tx מהבקר, ורגל Rx מהבקר (לא מושמש תכניתית ממשק Rx).

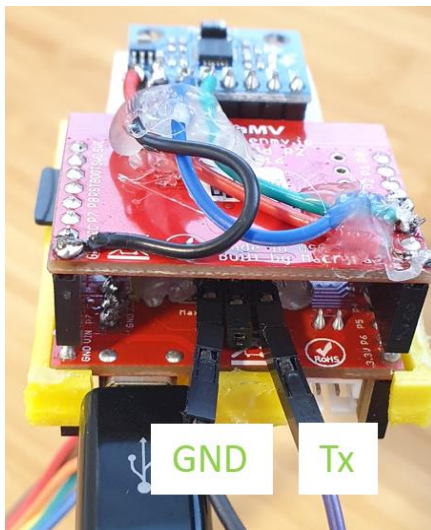


Figure 21 : UART setup wiring



Figure 20: Debug Setup

6.1 תצורת המדידה

בוצעו מדידות להבנת מעטפת הביצועים של המוצר. מערך המדידה כלל חיבור של מצלמת המט"ד לתחנה קרקעית, הקרנת שעון ע"ג התחנה הקרקעית, ושידור במסך סמוך. הסטאפ הזה הוקלט. לאחר מכן יש תהליך של מיצוי המידע, שהוא מעבר frame-by-frame על הסרטון המוקלט באמצעות תכנה דוגמת VLC player, תיעוד השעון של התחנה הקרקעית (ground truth שלנו), ותיעוד תמונה שהמטע"ד שלח. באמצעות כך יש לנו וקטור של frames עוקבים של המטע"ד והשיהויים בין כל פריים.

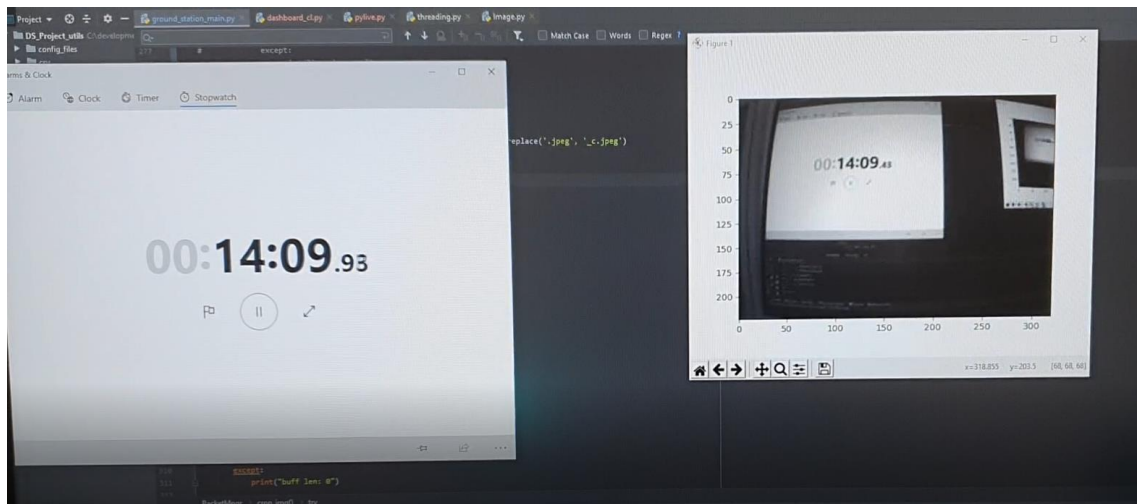


Figure 22 : Delay inspection measuring setup

6.2 תוצאות

delay				
frames counted	mean delay [msec]	stdev [msec]	max [msec]	min [msec]
16	237.50	27.73	270	200

ניתן לראות את ביצועי המערכת, עם שיהוי אופייני של כ 316msec בין כל פריים, עם קצב אופייני של 16fps

7.1 סיכום

הפרויקט מציג מטע"ד קטן וחדשני, ככלי עזר למחקר ולעבוד המשך, שמביא נתונים ויזואליים-אינרציאליים לתחנת קרקע. במעבדת הרחפנים האוטונומיים באוניברסיטת תל אביב, נעשה שימוש רב בניווט רחפנים באמצעות SLAM ע"י קבלת מידע חזותי של מיקום הרחפן. ההבנה שהיתוך מידע אינרציאלי יכול לשפר את ביצועי ה-SLAM, הביאו אותנו לחפש מוצרים שיכולים לתת מענה לבעיה הטכנולוגית. הפרויקט יצא לדרך מתוך הבנה שלא קיימת מערכת מדף כזו מוכנה לצרכי המעבדה, ועלינו לתכנן ולבנות כזו בעצמנו.

המעבדה סיפקה לנו את הצרכים ההנדסיים שנזדקקנו להם, החל מחומרות, כלי עבודה, סביבת עבודה והנחיה טכנולוגית. שלבי הפיתוח, מתוך עבודה בסביבת אילוצים נתונה, הביאו אותנו לבצע שיקולים הנדסיים כדוגמת שימוש נכון ברוחב הפס הנתון, עבודה בגודל תמונות וקצב נתונים מסוים.

הפרויקט המשותף של תחנת הקרקע והמטע"ד, שמורים ב repository של מעבדת הרחפנים:

<https://github.com/tau-adl/Drones-Navigation-Payload>

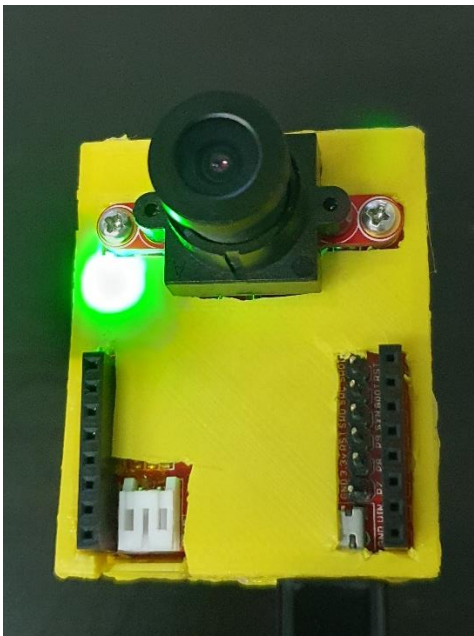
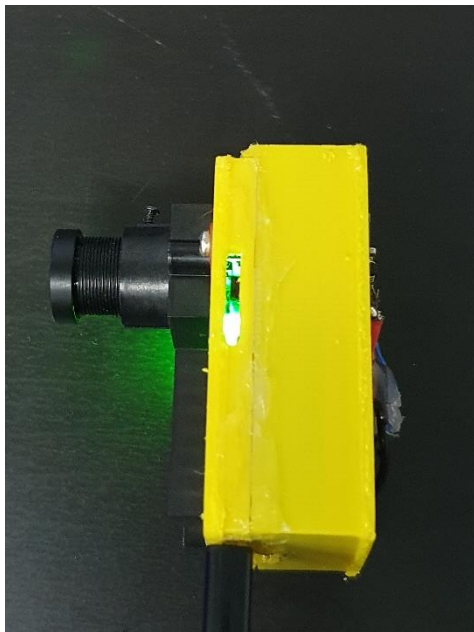
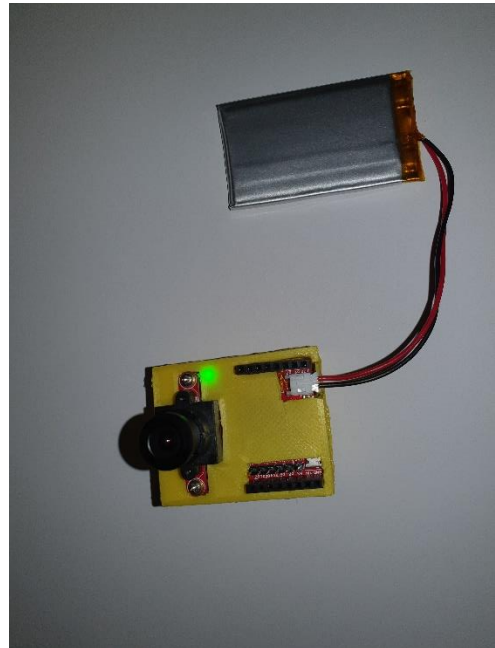
ברצוני להודות למר יוני מנדל - המנחה שלי בפרויקט הזה, ולשנהב עובדיה-השותף לבניית המערכת הזו.

7.2 עבודות המשך

הפרויקט הוכיח היתכנות מבחינת חומרה ותכנה להשלמת מטרת הפרויקט. אולם יש עוד מקום לשיפור - על מנת לתת ביצועים מספקים ואמינים לטובת השמשת אלגוריתמי ניווט ועיבוד post processing ערכי על תחנה קרקעית

1. השמשת מצבי שינה בבקר, או מציאת תדר עבודה אופטימלי על מנת להשיג צריכות זרמים אופטימליים למטע"ד שמוזן ממתח סוללה
2. עריכת הרכיבים ע"ג מעגל ייעודי – דבר שיפתור בעיות אלמ"ג, רובוסטיות של המערכת, יקטין גודל ויפחית משקל
3. בחינת שימוש במצלמה FLIR על מנת לתמוך באפליקציות ניווט לא בחדר סגור
4. התאמת קצב דגימת המצלמה להתמודדות עם תופעות ריצוד תאורת פלורוסנט בתמונות
5. בחינת התועלת בתוספת תאורה חיצונית לתמונה – דוגמאת מבזק או תאורה להצפת הסצינה
6. עבודה עם מודם המטע"ד בתור Client ולא בתור AP, יאפשר עבודה עם תחנה קרקעית אחת אל מול יותר ממטע"ד יחיד

8.1 תמונות נוספות של המערכת



8.2 ביבליוגרפיה

1. **VersaVIS: An Open Versatile Multi-Camera Visual-Inertial Sensor Suite**, Florian Tschopp , Michael Riner , Marius Fehr , Lukas Bernreiter , Fadri Furrer , Tonci Novkovic , Andreas Pfrunder , Cesar Cadena , Roland Siegwart , and Juan Nieto, Sensors 2020
2. **OpenMv** - <https://openmv.io/>
3. **Global Shutter vs. Rolling Shutter image (Figure 1)** - <https://andor.oxinst.com/learning/view/article/rolling-and-global-shutter>
4. **Digital Camera Interface** – and STmicro document - dm00373474
5. **MPU6050 breakout schematic** - <https://easyeda.com/replicagames/mpu6050>
6. **MPU6050 driver** - https://github.com/sinadarvi/SD_HAL_MPU6050
7. **OpenMV repository** - <https://github.com/openmv/openmv>
8. **Digital Camera Interface (DCMI) for STM32 MCUs** – AN5020
9. **On Semiconductor MT9V034** - Datasheet
10. **STM32H743xl/G** – Datasheet
11. **Invensense MPU6050** - Datasheet
12. **Microchip ATWINC1500** - Datasheet