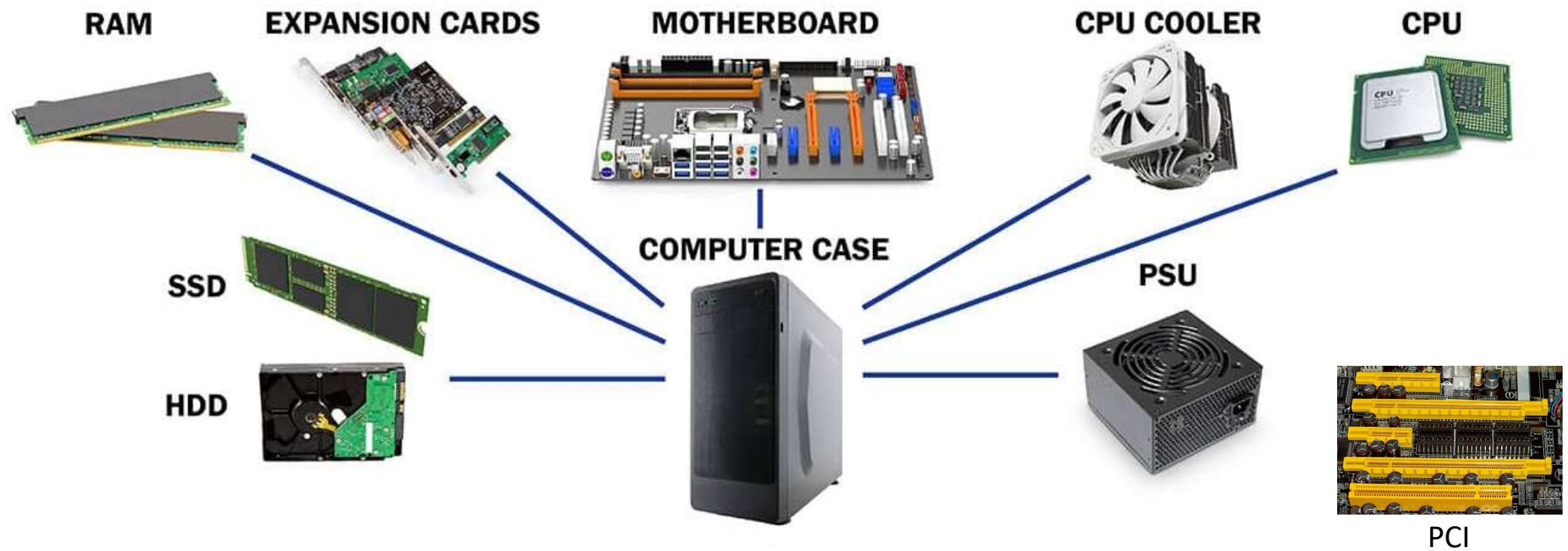




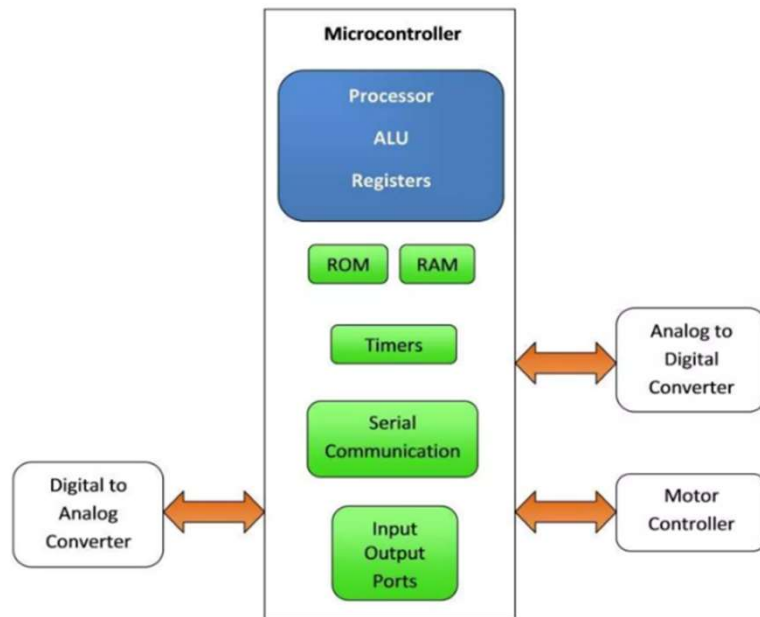
Real Time with STM32

אילו רכיבי חומרה מרכיבים מחשב?

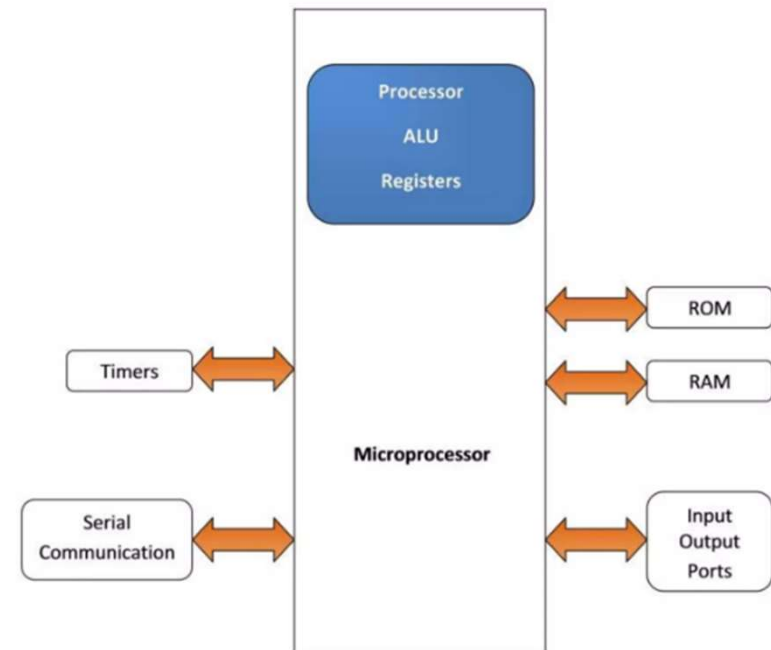


CPU vs MCU

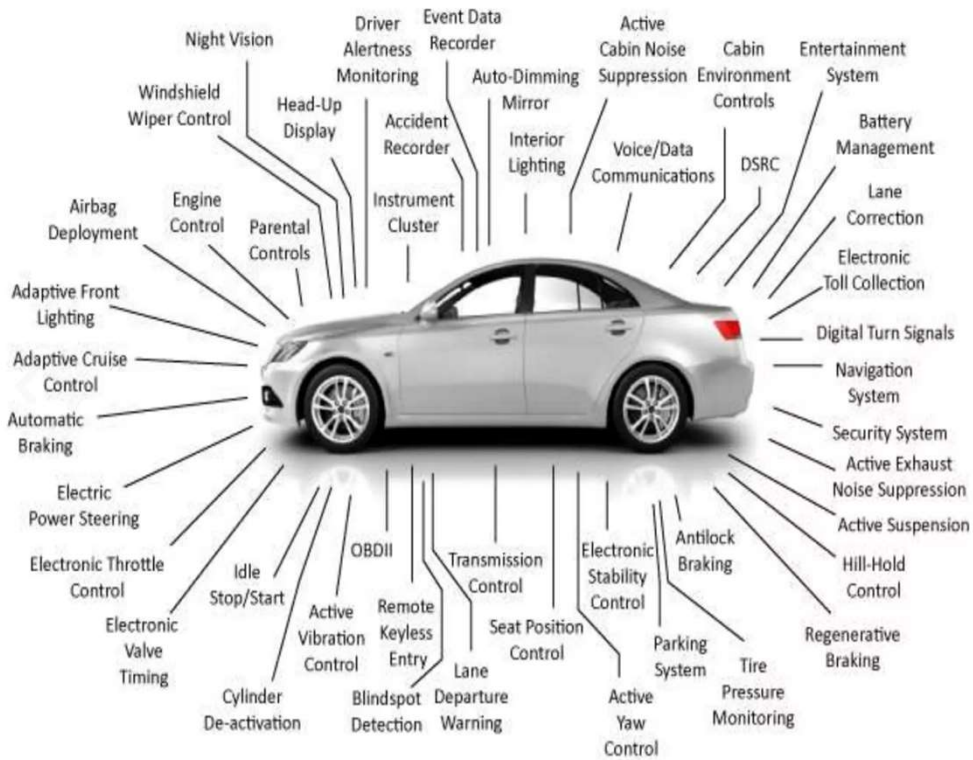
לעומת מיקרו בקר הוא מערכת מחשב המובנית על שבב יחיד המכיל בתוכו זיכרון מובנה, טיימרים ועוד.



מיקרו מעבד הוא CPU המכיל בתוכו יחידת עיבוד וכמות קטנה של זיכרון והוא מתממשק עם זיכרון חיצוני, טיימרים ועוד, המובנים בשבבים חיצוניים



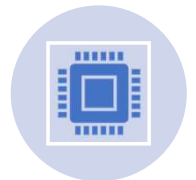
CPU vs MCU



לעומת CPU שהוא יחידת עיבוד חזקה המסוגלת לבצע מגוון רב של משימות ומיועדת למטרות כלליות.



המאפיין העיקרי של MCU הוא שהוא מיוצר לשימוש ספציפי ומתאים במיוחד לו.



מיקרו בקר הוא כמו מערכת של מחשב על שבב בודד. הוא מכיל בתוכו שבבים נוספים ומשמש בעיקר למערכות משובצות (EMBEDDED)



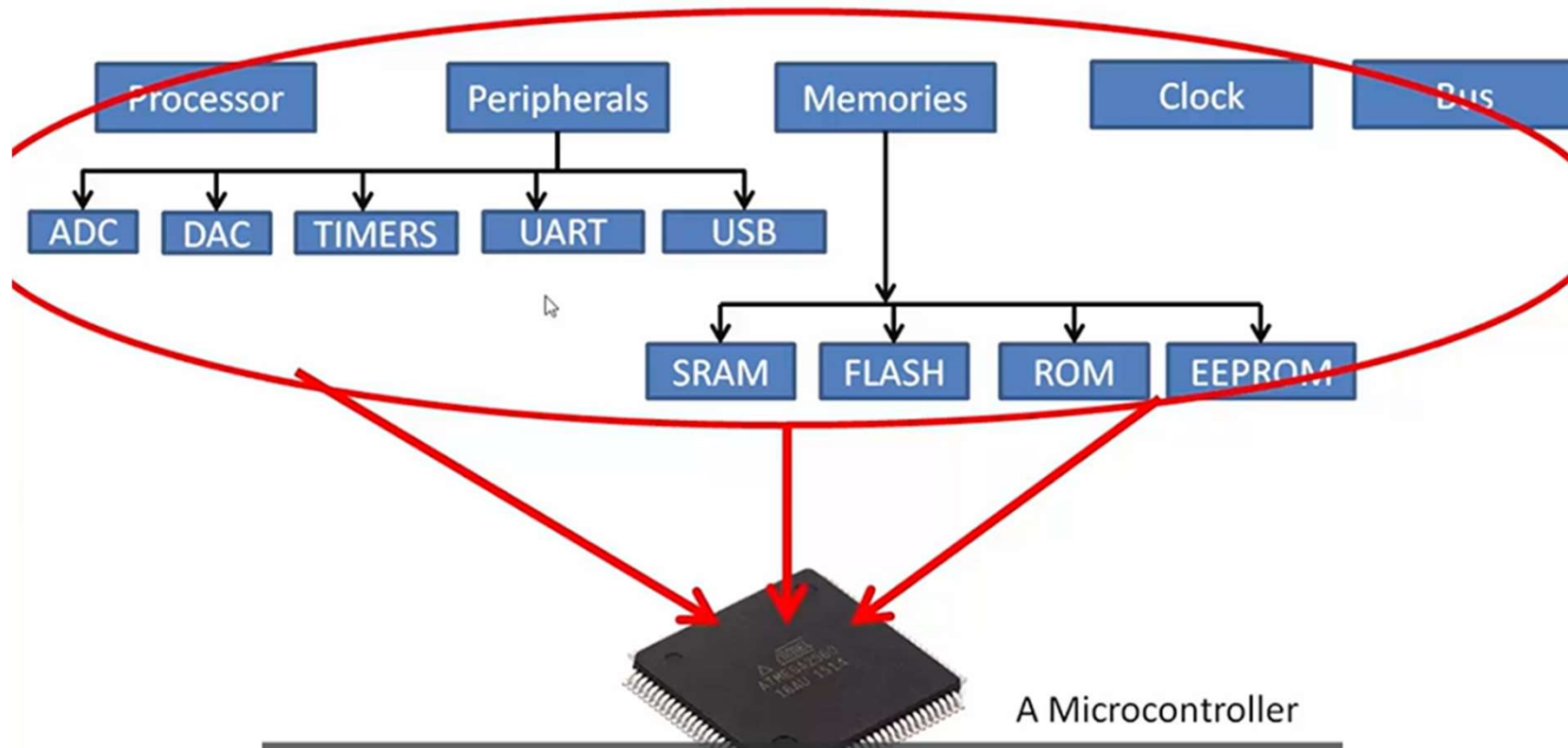
מיקרו-בקרים מצויים כמעט בכל מוצר אלקטרוני/חשמלי כלשהו, כגון כלי רכב, מכונות ביתיות ומשרדיות ואפילו צעצועים. (ומתאים במיוחד למכשירים בעלי צריכה חשמל נמוכה כיון שאינו צורך בהרבה POWER)



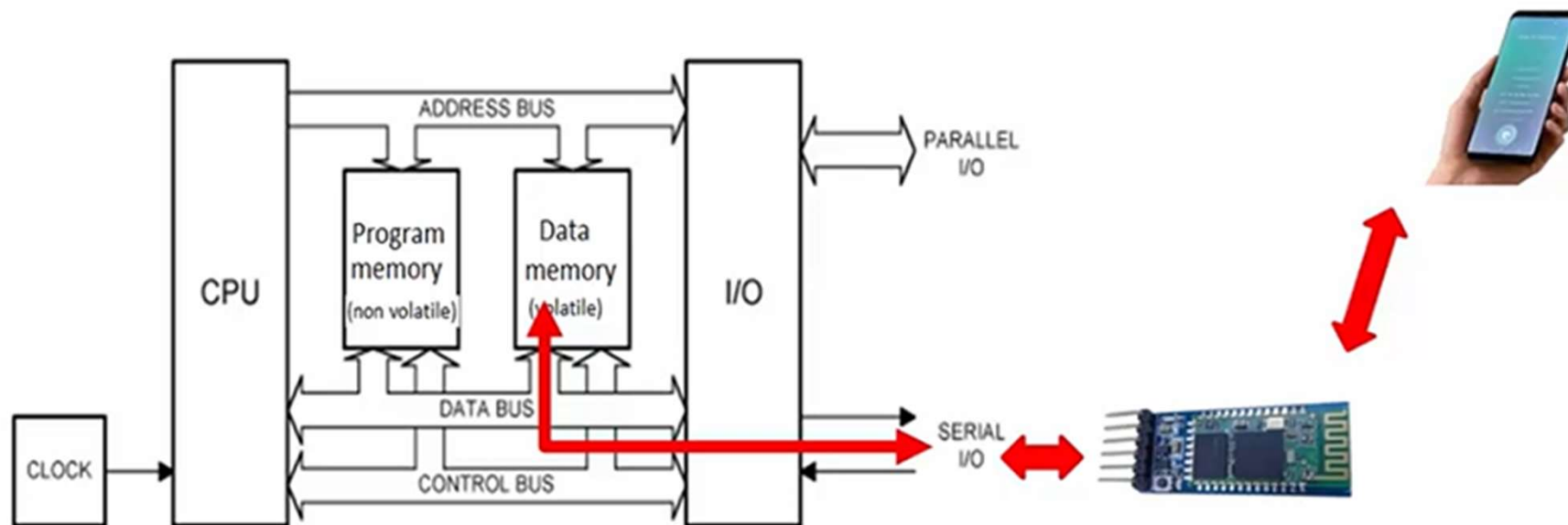
ה MCU פחות עוצמתי מ CPU ובעל משאבים מוגבלים (זיכרון, מהירות, מתח וכו') וממילא גם זול יותר ובדרך"כ קומפקטי יותר.

Microprocessor (μ P)	Microcontroller (μ C)
מיועד ליישומי מחשב למטרות כלליות	מיועד ליישומי מערכת משובצים ספציפיים
CPU עם זיכרון מובנה מינימלי, ציוד היקפי וממשקי קלט/פלט	מערכת מחשב בעלת שבב יחיד עם זיכרון, ציוד היקפי וממשקי קלט/פלט מובנים
מעבד + שבבים נוספים לתמיכה	שבב יחיד
צריכה חשמל גבוהה	צריכת חשמל נמוכה
מערכת הוראות גמישה יותר וניתנת לתכנות	מערכת הוראות קבועה
יציאות ק.פ בודדות	כמה שיותר יציאות לק.פ
ציוד היקפי חיצוני	ציוד היקפי מובנה
עלות גבוהה	עלות נמוכה
General-purpose	Embedded systems
כלי פיתוח סטנדרטיים ושפות כגון C, ++C ואסמבלי	סביבת פיתוח המסופקת על ידי יצרנים, עם שפות תכנות מיוחדות וכלים
מהירות שעון גבוהה יותר, בדרך כלל גדולה מ-1GHz	מהירות שעון נמוכה יותר, בדרך כלל פחות מ-100MHz

הרכיבים העיקריים של הבקר

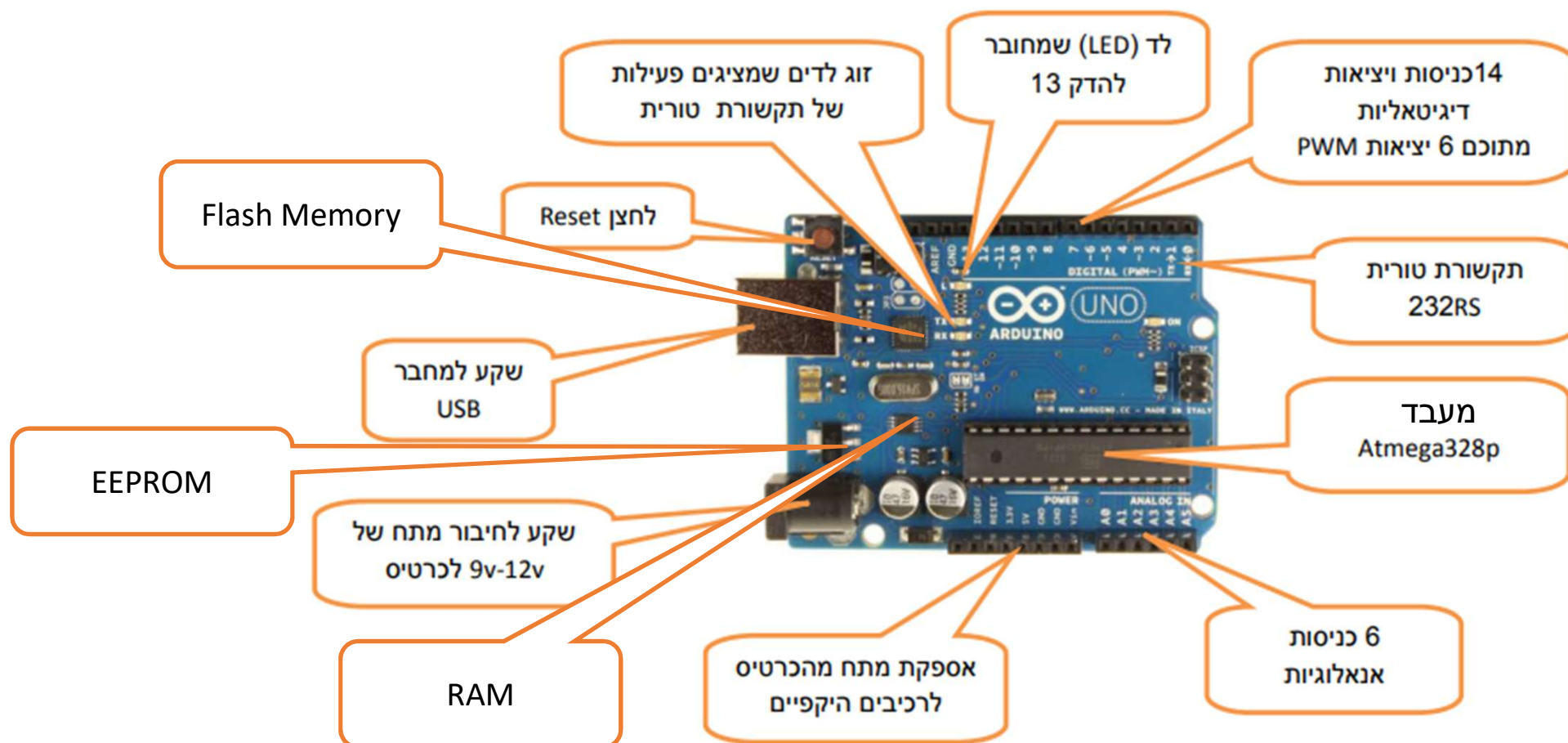


הקשר בין הרכיבים במיקרו-בקר

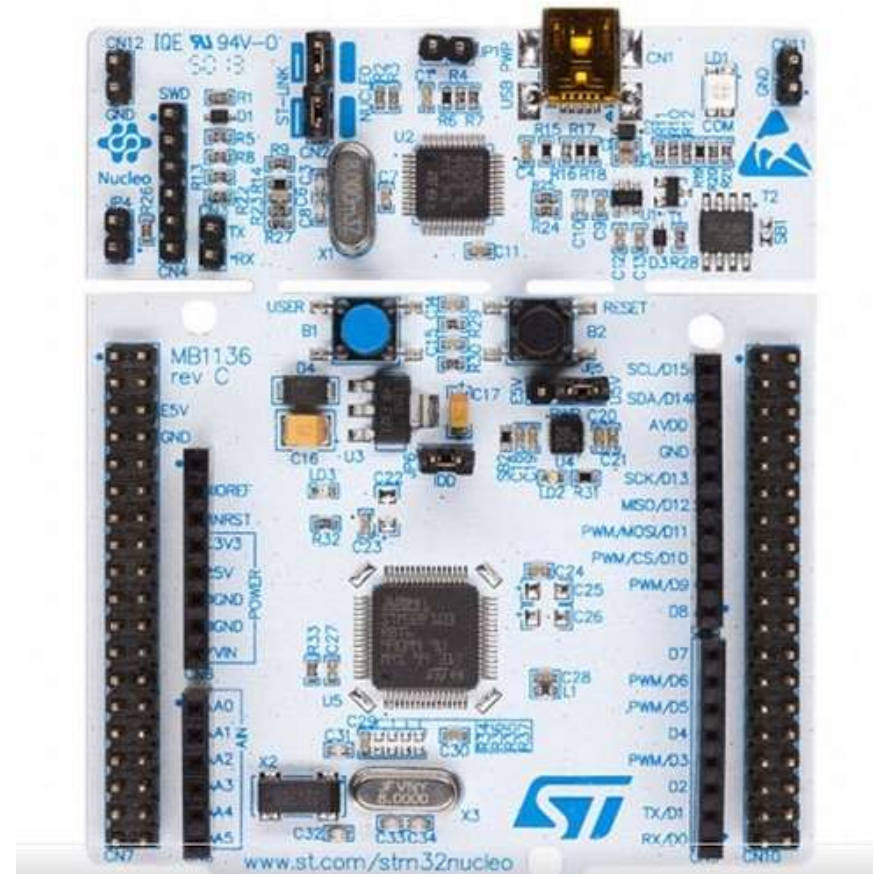


MCU receiving data from Bluetooth and storing in data memory

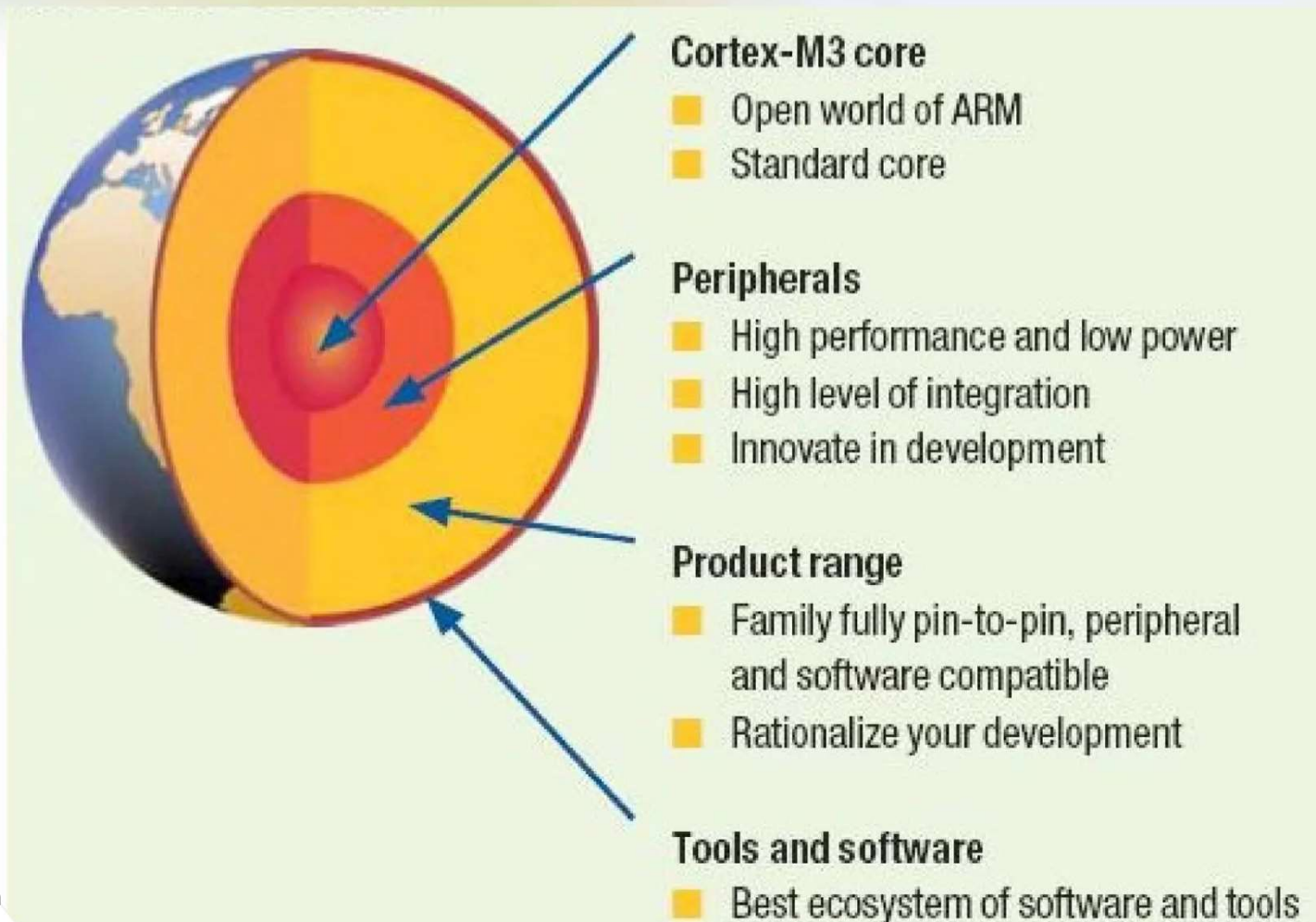
רכיבי הבקר- בקר ארדואינו



- התמונה מציגה את הרכיבים השונים של לוח STM32 Nucleo:
- MCU: שבב המרכזי של הלוח, הממוקם במרכז הלוח.
 - זיכרון: זיכרון ROM, RAM, הממוקם בצד השמאלי של הלוח.
 - יציאות: יציאות UART, SPI, I2C, GPIO, הממוקמות בצד הימני של הלוח.
 - כניסות: כניסות ADC, GPIO, DAC, הממוקמות בצד השמאלי של הלוח.
 - שקע USB: שקע USB, הממוקם בצד הימני העליון של הלוח.
 - מתכנת: ST-LINK/V2-1, מתכנת ומאתר באגים, הממוקם בצד השמאלי העליון של הלוח.
 - כותרות: ST Morpho, כותרות הרחבה, הממוקמות בצד הימני התחתון של הלוח.



The World of STM32



שימושים של STM32

סדרת מיקרו-בקרים מבוססי ARM32 המיוצרת על ידי STMicroelectronics.

- מספק ביצועים גבוהים בצריכת חשמל נמוכה.
- מציע מגוון רחב של תכונות, כולל פריפריות תקשורת, טיימרים ומעבד גרפי.
- זמין במגוון רחב של מודולים, מה שמקל על התאמת המוצר הספציפי לצרכים שלך.
- עלות נמוכה

שימושים של STM32 בתעשייה

- בקרת מכונות: משמש לשלוט במכונות תעשייתיות, כגון רובוטיקה, מכונות ייצור ומערכות בקרת איכות.
- בקרת תהליכים: משמש לשלוט בתהליכים תעשייתיים כגון התקני מודולים, מערכות IoT ומערכות בקרת תעבורה.

דוגמאות

- חברת General Electric משתמשת ב STM32-כדי לשלוט במנועים שלה.
- חברת Siemens משתמשת ב STM32-כדי לשלוט במערכות ייצור שלה.
- חברת Bosch משתמשת ב STM32-כדי לשלוט במערכות תעופה שלה.



תכונות עיקריות STM32

• מעבד 32 ARM-ביט Cortex-M3 עם פלאש ו-SRAM מובנים

• עד 512KB זיכרון פלאש

• עד 64KB זיכרון SRAM

• ניהול שעון, איפוס ואספקת חשמל

• פריפריות תקשורת מרובות

• I2C, USART, SPI

• מצב צריכת חשמל נמוכה של שינה,

עצירה ומצב המתנה

• טיימר 16-ביט כפול

• שעון Real Time

• בקר DMA

• 12 DAC סיביות

• פיתוח קל, זמן יציאה לשוק מהיר

מעבד 32 ARM-ביט Cortex-M3: המעבד הוא הלב של המערכת ומבצע את כל החישובים.

עד 512KB זיכרון פלאש: זיכרון זה משמש לאחסון תוכנה, כגון מערכת הפעלה ותוכניות.

עד 64KB זיכרון SRAM: זיכרון זה משמש לאחסון נתונים, כגון ערכי משתנה ונתונים פנימיים.

ניהול שעון, איפוס ואספקת חשמל: תכונות אלו מאפשרות לשלוט בצריכת החשמל של המעבד.

פריפריות תקשורת מרובות: פריפריות אלו מאפשרות לך לתקשר עם מכשירים אחרים.

מצב צריכת חשמל נמוכה של שינה, עצירה ומצב המתנה: מצבים אלו מאפשרים להפחית את צריכת החשמל של המעבד כאשר הוא אינו בשימוש פעיל.

טיימר 16-ביט: טיימרים אלו משמשים למטרות שונות, כגון מדידת זמן, יצירת פולסים והגנה על מערכת מפני עומס יתר.

שעון זמן אמת: שעון זה מספק אות זמן אמיתי מדויק.

בקר DMA: בקר DMA מאפשר לך להעביר נתונים בין זיכרון לפריפריות מבלי להסתמך על המעבד.

12 DAC סיביות: DAC זה מאפשר לך להמיר נתונים אנלוגיים לדיגיטליים.

```
#define PIN_TRIG 7
#define PIN_ECHO 6

void setup() {
  Serial.begin(115200);
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
}

void loop() {
  // Start a new measurement:
  digitalWrite(PIN_TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);

  // Read the result:
  int duration = pulseIn(PIN_ECHO, HIGH);
  Serial.print("Distance in CM: ");
  Serial.println(duration / 58);
  Serial.print("Distance in inches: ");
  Serial.println(duration / 148);

  delay(1000);
}
```

<https://wokwi.com/projects/387931197716196353>

©rachel.sch8825@gmail.com

סוגי תקשורת

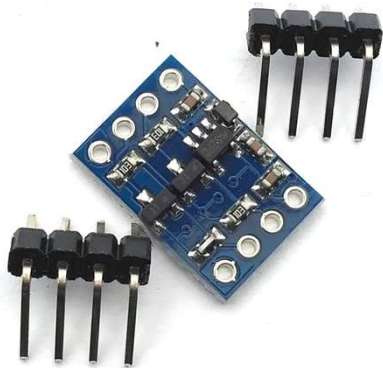
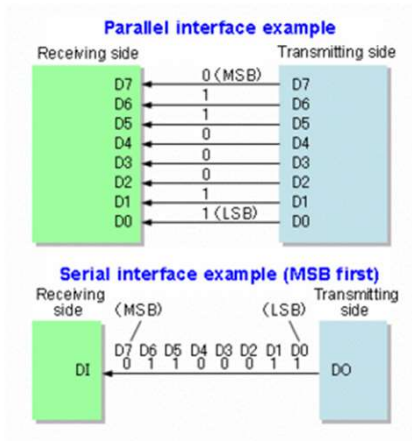
תקשורת סריאלית

בתקשורת סריאלית, המידע מועבר בין שני התקנים בצורה של ביטים, אחד אחרי השני. הביט הראשון הוא הביט המשמעותי ביותר, והביט האחרון הוא הביט הפחות משמעותי.

תקשורת מקבילית

בתקשורת מקבילית, המידע מועבר בין שני התקנים בצורה של ביטים רבים (תלוי ברוחב הפס), בבת אחת. מספר הביטים המועברים בבת אחת נקבע על ידי מספר הנתיביים (wires) במערכת התקשורת.

בדרך כלל, תקשורת סריאלית היא הבחירה הטובה יותר עבור יישומים שבהם יש צורך להעביר כמות קטנה של נתונים, כגון תקשורת בין מחשב למקלדת או בין מחשב לעכבר. תקשורת מקבילית היא הבחירה הטובה יותר עבור יישומים שבהם יש צורך להעביר כמות גדולה של נתונים במהירות גבוהה, כגון תקשורת בין מעבד לזיכרון או בין מעבד להתקן אחסון.



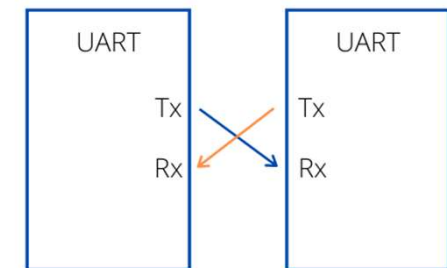
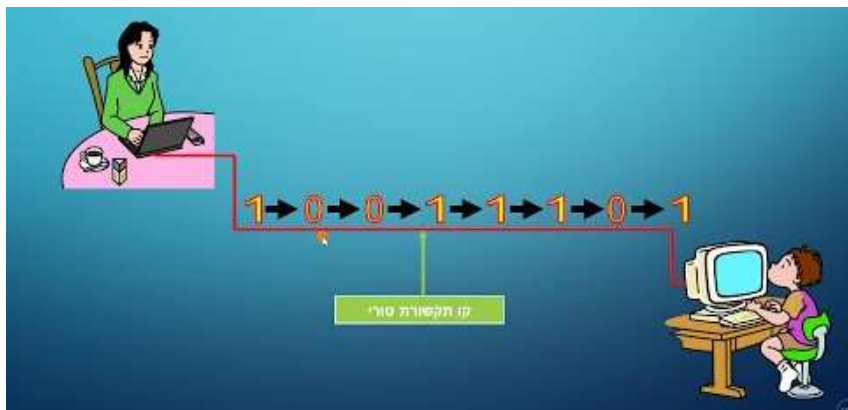
Serial Communication vs Parallel Communication

מאפיין	תקשורת סריאלית	תקשורת מקבילית
מספר סיביות המועברות בבת אחת	אחת	רבות
צורת השליחה והקליטה	ביט אחר ביט	ביטים רבים בבת אחת
זמן העברת הנתונים	איטי יותר	מהיר יותר
עלות החומרה	נמוכה יותר	גבוהה יותר
טווח השימוש	רחב יותר	מוגבל יותר

סוגי תקשורת סריאלית

קיימים שני סוגים עיקריים של תקשורת סריאלית:

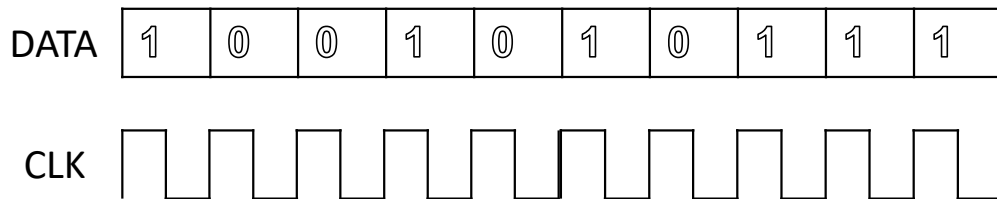
- תקשורת סריאלית סינכרונית (Synchronous Serial Communication): בשיטת תקשורת זו, שני ההתקנים משתמשים באותו שעון כדי לסנכרן את השליחה והקליטה של הביטים.
- תקשורת סריאלית א-סינכרונית (Asynchronous Serial Communication): בשיטת תקשורת זו, שני ההתקנים אינם משתמשים בשעון לסנכרון. הפרוטוקול הנפוץ ביותר לתקשורת א-סינכרונית היא תקשורת UART.



תקשורת טורית - סריאלית

סינכרונית – עם CLK
לדוגמא SPI

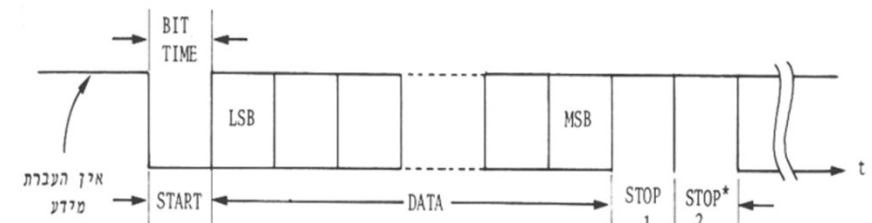
יש 2 קווי נתונים אחד לשעון ואחד לדאטה
(לפעמים יש יותר מ2 לשימושים נוספים)



בכל עליית שעון מגיע ביט נוסף

אסינכרונית – ללא CLK
לדוגמא UART (RS232)

שני הרכיבים שמתקשרים ביניהם מחליטים מראש:
bits per second = BPS (בדור"כ 9600)
כמה ביטים במילה (בדור"כ 8)
כמה סטופ ביט 1-2 (1)



* לא תמיד קיים.

- כדי להעביר מילה בקו טורי עלינו לשדר לפי סדר
- א. START BIT
 - ב. סיביות המילה מתחילים עם הLSB
 - ג. STOP BIT או אחד שניים

UART (Universal Asynchronous Receiver Transmitter)

פרוטוקול UART הוא פרוטוקול א-סינכרוני, מה שאומר שההתקנים המעורבים בתקשורת אינם מסונכרנים על ידי שעון משותף. במקום זאת, כל התקן משתמש בתזמון שלו כדי להעביר ולקלוט נתונים. התהליך הזה חוזר על עצמו עד להעברת כל הנתונים.

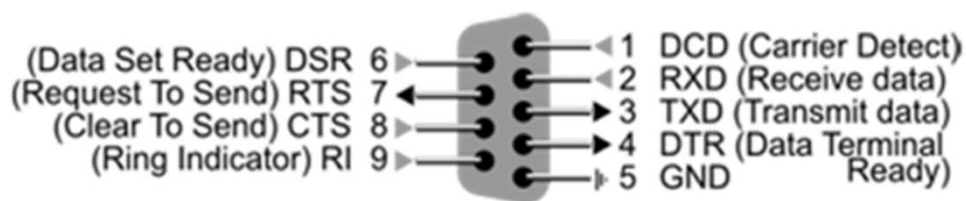
פרוטוקול UART הוא פרוטוקול פשוט וקל ליישום, אך הוא אינו מאפשר העברת נתונים במהירות גבוהה כמו פרוטוקול SPI. והוא משמש במגוון רחב של יישומים

יתרונות:

- פשוט ליישום.
- משתמש בשני קווים פיזיים בלבד.
- תומך במגוון רחב של מהירויות העברה.

חסרונות:

- מהירות העברה נמוכה יחסית.
- לא מתאים להעברת כמות גדולה של נתונים בזמן.



בפרוטוקולים בסיסיים ישתמשו בדר"כ רק ב 3 יציאות: RX, TX, GND