

## ארדואינו – פונקציות התקשורת הטורית

### א. כללי

התקשורת הטורית משמשת לחיבור בין כרטיס הארדואינו למחשב או רכיבים אחרים. לכל כרטיסי הארדואינו יש לפחות פורט טורי אחד ( הנקרא UART או Serial). הוא מתחבר בעזרת הדקים דיגיטאליים RX0 ו TX0 של המיקרו בקר אל מחבר ה USB של המחשב כאשר ביניהם יש רכיב תיווך המעביר מ TTL ל USB ולהיפך.

לתקשורת הטורית יש חוצץ ( buffer ) טורי שאליו נכנסים הנתונים שנקלטים בתקשורת טורית. החוצץ הוא זיכרון - בדרך כלל בגודל של 64 בתים אם כי ניתן להרחיב אותו ( לא מוסבר בפרק הזה).

ניתן להשתמש בתוכנית ה "מוניטור הטורי" שבתוכנת סביבת הפיתוח של הארדואינו . יש להקיש על צלמית (ICON) המוניטור (צד ימין למעלה) או דרך תפריט ה toolbar ובחירת serial monitor במסך התפריט שהתקבל. יש לדאוג שקצב התקשורת –baud rate– במסך המוניטור שנקבל יהיה זהה לזה שהגדרנו בתוכנית שלנו בפונקציית ה ( ) setup בעזרת הפונקציה ; ( קצב תקשורת ) Serial.begin .

בארדואינו מגה–Arduino Mega - יש 3 פורטי תקשורת טורית נוספים. תקשורת טורית 1 היא בהדקים 19 ( RX ) ובהדק 18 ( TX ) . תקשורת טורית 2 בהדקים 17 ( RX ) ו 16 ( TX ) ותקשורת טורית 3 בהדקים 15 ( RX ) ו 14 ( TX ) . כדי להשתמש בהדקים אלו לתקשורת עם המחשב האישי יש צורך במתאם מ USB ל SERIAL נוסף. כדי להשתמש בהם להתחברות לרכיבי TTL טוריים, יש לחבר את הדק TX שלנו להדק RX של הרכיב ואת הדק RX שלנו להדק TX של הרכיב ואת הדק האדמה שלנו להדק האדמה של הרכיב. לא לחבר הדקים אלו ישירות להדקי רכיבי RS232 כי הם עובדים עם  $\pm 12V$  וייגרמו לנזק לכרטיס הארדואינו.

גם לארדואינו דיו –Arduino Due– יש 3 הדקי תקשורת טורית נוספים להתחברות לרכיבי TTL של 3.3V. תקשורת טורית 1 היא בהדקים 19 ( RX ) ובהדק 18 ( TX ) . תקשורת טורית 2 בהדקים 17 ( RX ) ו 16 ( TX ) ותקשורת טורית 3 בהדקים 15 ( RX ) ו 14 ( TX ) . הדקים 0 ו 1 מתחברים גם הם אל רכיב ATmega 16U2 שבכרטיס. הוא משמש כרכיב המתאם בין USB לתקשורת טורית המתחבר אל פורט ה USB של ה debug . בנוסף יש פורט USB – serial – ברכיב SAM3X .

כרטיס ארדואינו לאונרדו - Arduino Leonardo - משתמש בתקשורת טורית 1 להתקשר דרך TTL ( 5V ) בהדקים 0 ( RX ) ו 1 ( TX ). התקשורת טורית נשמרת עבור USB CDC .

## ב. פונקציות תוכנה של הארדואינו

### 1. if(Serial)

הפונקציה בודקת ומדווחת האם הפורט הטורי (שמצוין בין הסוגריים) מוכן לתקשורת. בארדואינו ליאונרדו היא מדווחת האם החיבור הטורי של ה USB CDC פתוח. בכל הכרטיסים האחרים כולל if(serial1) בלאונרדו זה תמיד יחזיר true. התחביר בכל הכרטיסים השונים הוא: if(Serial). בלאונרדו if(Serial1) ובארדואינו מגה אפשר: if(Serial1) או if(Serial2) או if(Serial3). לא שולחים פרמטר לפונקציה והיא מחזירה ערך בוליאני boolean. אם ניתן לגשת אל הפורט כי יש נתון חדש בחוצץ הטורי היא מחזירה true. דוגמא:

```
void setup() {
  Serial.begin(9600); // אתחול תקשורת טורית ל 9600
  while (!Serial) { // המתנה שהפורט הטורי יתחבר
    // Leonardo only
  }
}

void loop() {
  // התכנית הרצויה
}
```

### 2. Serial.available()

מקבל את כמות הבתים (תווים) לקריאה מהפורט הטורי. אלו נתונים שכבר הגיעו ומאוחסנים בחוצץ של התקשורת הטורית (המכיל עד 64 בתים). הפונקציה היא תורשה מהמחלקה stream. התחביר:

```
Serial.available()
```

*Arduino Mega only:*

```
Serial1.available()
```

```
Serial2.available()
```

```
Serial3.available()
```

הפונקציה לא מקבלת פרמטרים ומחזירה את כמות הבתים האפשריים לקריאה מהחוצץ. דוגמא:

```
Int incomingByte=0; // לנתונים שייקראו מהחוצץ
```

```
void setup(){
```

```
Serial.begin(9600); // פתיחת הפורט הטורי לקצב 9600 ביטים בשנייה  
}
```

```
voidloop(){
```

```
    if(Serial.available()>0) // שליחת נתון רק כשקולטים נתון  
    {  
        incomingByte=Serial.read(); // קרא את הביית הנכנס  
        Serial.print("I received: "); // " קלטתי : "  
        Serial.println(incomingByte,DEC);  
    }  
}
```

דוגמא לאדואינו מגה :

```
voidsetup(){
```

```
    Serial.begin(9600); // פתיחת הפורט הטורי לקצב 9600 ביטים בשנייה  
}
```

```
voidloop(){
```

```
    if(Serial.available()) // האם יש נתונים בחוצץ  
    {  
        intinByte=Serial.read(); // קריאת הנתון שבחוצץ  
        Serial1.print(inByte,BYTE); // הדפסת הנתון למוניטור  
    }  
    // קריאה מפורט טורי 1 ושליחה מ 0  
    if(Serial1.available()) // קריאה מפורט טורי מספר 1  
    {  
        intinByte=Serial1.read();  
        Serial.print(inByte,BYTE);  
    }  
}
```

### 3. Serial.begin( קצב התקשורת )

הפונקציה קובעת את את קצב התקשורת בביטים לשנייה ( baud ) לתקשורת הטורית. לתקשורת עם המחשב יש לקבוע את אחד מהקצבים: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200. ניתן לרשום מספר אחר אם עובדים עם רכיב שעובד בקצב אחר. ניתן לקבוע גם את כמות הביטים, זוגיות וביטים של סיום. ברירת המחדל היא 8 ביט ללא זוגיות עם ביט סיום אחד.

```
Serial.begin(speed)
```

התחביר :

```
Serial.begin(speed, config)
```

עבור ארדואינו מגה בלבד:

```
Serial1.begin(speed);
```

```
Serial2.begin(speed);
```

```
Serial3.begin(speed);  
Serial1.begin(speed, config);  
Serial2.begin(speed, config);  
Serial3.begin(speed, config);
```

speed הוא קצב התקשורת בביטים לשנייה. זהו משתנה מטיפוס long.  
config הוא כמות הביטים בנתון, הזוגיות ומספר הביטים של הסיום. דוגמאות:

- SERIAL\_5N1 // 5 ביטים בנתון, אין זוגיות וביט סיום 1
- SERIAL\_8N1 // ברירת המחזל 8 ביטים ללא זוגיות וביט סיום אחד
- SERIAL\_5N2

הפונקציה איננה מחזירה ערך.

דוגמא:

```
voidsetup(){  
  Serial.begin(9600// ביטים בשנייה 9600 לקצב הטורי  
}
```

```
voidloop(){}
```

דוגמאות עם ארדואינו מגה : לדוגמא נשתמש בכל 4 פורטי התקשורת הטורית בקצבים משתנים :

```
voidsetup(){  
  Serial.begin(9600);  
  Serial1.begin(38400);  
  Serial2.begin(19200);  
  Serial3.begin(4800);  
  
  Serial.println("Hello Computer");  
  Serial1.println("Hello Serial 1");  
  Serial2.println("Hello Serial 2");  
  Serial3.println("Hello Serial 3");  
}
```

```
voidloop(){}
```

#### 4. Serial.end()

הפונקציה לא מאפשרת את התקשורת הטורית ומאפשרת להשתמש בהדקי RX ו TX כהדקים לכניסה או יציאה רגילים. כדי לחזור ולאפשר את הפורט יש לקרוא ל Serial.begin () .

התחביר : Serial.end()

לארדואינו מגה בלבד:

Serial1.end()

Serial2.end()

Serial3.end()

הפונקציה איננה מקבלת פרמטרים ואיננה מחזירה ערך.

### **Serial.find() .5**

הפונקציה קוראת נתונים מהחוצץ הטורי עד שהמחרוזת הנתונה נמצאה. הפונקציה מחזירה true אם נמצאה המחרוזת או false אם הסתיים הזמן. גם היא תורשה של המחלקה stream. התחביר: Serial.find(target) כאשר target היא המחרוזת שצריך לחפש (מטיפוס char). הפונקציה מחזירה ערך בוליאני true או false.

### **Serial.findUntil() .6**

הפונקציה קוראת נתונים מהחוצץ הטורי עד שהתגלתה המחרוזת הרצויה או הסתיימה המחרוזת. הפונקציה מחזירה true אם נמצאה המחרוזת או false אם לא. הפונקציה היא תורשה של המחלקה stream.

התחביר :

Serial.findUntil(target, terminal)

הפרמטרים שהפונקציה מקבלת הם target – המחרוזת אותה נחפש (char) ו terminal היא מחרוזת היציאה (מטיפוס char). הפונקציה מחזירה ערך בוליאני true או false.

### **flush () .7**

הפונקציה מחכה לסיום התקשורת היוצאת בפורט הטורי. גם היא תורשה של המחלקה stream. הפונקציה לא מקבלת פרמטרים ולא מחזירה ערך.

התחביר : Serial.flush()

בארדואינו מגה :

Serial1.flush()

Serial2.flush()

Serial3.flush()

### **Serial.parseFloat() .8**

הפונקציה מחזירה את המספר הממשי הראשון ( נקודה צפה ) מהחוצץ הטורי. קופצים על תווים שאינם ספרות . הפונקציה מסתיימת בתו הראשון שאיננו מספר בנקודה צפה. גם היא תורשה מהמחלקה .stream  
התחביר : Serial.parseFloat() . הפונקציה לא מקבלת פרמטרים ומחזירה ערך מטיפוס ממשי float.

## 9. Serial.parseInt()

הפונקציה לא מקבלת ערכים ומחזירה את המספר השלם שהגיע בתקשורת הטורית. אם לא הגיע מספר שלם בתוך 1 שנייה מוחזר ערך של 0 מהפונקציה . (זמן ההמתנה ניתן לשנוי בפונקציה )  
Serial.setTimeout ( ) . הפונקציה יורשת מהמחלקה Stream . בארדואינו מגה שבו 4 תקשורות טוריות ניתן לרשום :

Serial1.parseInt()

Serial2.parseInt()

Serial3.parseInt()

## 10. Serial.peek()

הפונקציה איננה מקבלת ערכים ומחזירה את הבייט הבא ( תו ) המגיע בתקשורת הטורית מבלי שתסלק אותו מחוצץ ( buffer ) התקשורת הטורית הפנימי. אם אין נתון היא תחזיר -1 . קריאה מוצלחת לפונקציה ( ) peek תחזיר את אותו התו כמו הקריאה לפונקציה ( ) read (שבסעיפים הבאים) . היא יורשת מהמחלקה Stream .

## 11. Serial.print()

הפונקציה מדפיסה לפורט הטורי טקסט אסקיי, שאנחנו יכולים לקרא. ניתן לראות את ההדפסה על המוניטור הטורי של התוכנה של הארדואינו ( serial monitor ) . לפקודה צורות שונות. מספרים מודפסים תוך שימוש בתווי האסקיי לכל ספרה. כך גם מספרים ממשיים כשברירת המחדל היא 2 ספרות אחרי הנקודה. בתים נשלחים כמו תו בודד. תווים ומחרוזות נשלחים כמות שהם. לדוגמא :

- Serial.print(78); // נותן "78"
- Serial.print(1.23456); // נותן "1.23"
- Serial.print('N'); // נותן "N"
- Serial.print("Hello world."); // נותן "Hello world."

אופציות נוספות הן לציין – פורמט - באיזה בסיס רוצים להשתמש . האפשרויות : BIN - בינארי .  
OCT -אוקטאלי ( בסיס 8 ) , HEX – בסיס 16 –הקסה דצימאלי . למספרים ממשיים float , המספר  
מציין כמה ספרות אחרי הנקודה רוצים לקבל. דוגמאות :

- Serial.print(78, BIN); // ( 78 = 4eh) - נותן "1001110"
- Serial.print(78, OCT); // נותן "116"
- Serial.print(78, DEC); // נותן "78"
- Serial.print(78, HEX); // נותן "4E"
- Serial.println(1.23456, 0); // 0 היא הנקודה היא 0
- Serial.println(1.23456, 2); // נותן "1.23"
- Serial.println(1.23456, 4); // נותן 4 ספרות אחרי הנקודה "1.2346"

ניתן להעביר מחרוזות עם הפונקציה ועם התו F . לדוגמא :

```
Serial.print(F("Hello World")); // Hello World נקבל
```

התחביר : ; ( ערך מכל טיפוס ) Serial.print( או ; (פורמט (בסיס הספירה), ערך Serial.print( הפונקציה מחזירה ערך מטיפוס long שמציין כמה בתים נכתבו ( זו אופציה - לא חובה לקרא זאת).

כדי לשלוח בית יחיד כדאי להשתמש בפונקציה Serial.write ( ) .

דוגמא :

```
/*  
נשתמש בלולאת for להדפסת מספר בפורמטים שונים  
*/  
int x =0; // המשתנה שנדפיס את ערכו  
  
voidsetup(){  
Serial.begin(9600); // פתיחת התקשורת הטורית לקצב 9600 ביטים בשנייה  
}  
  
voidloop(){  
// לסדר את ההדפסה לפי כותרות הדפסת תוויות  
Serial.print("NO FORMAT"); // הדפסת התווית  
Serial.print("\t"); // הדפסת tab והעברת הסמן לשדה ההדפסה הבא  
  
Serial.print("DEC");// הדפסת המחרוזת DEC  
Serial.print("\t"); // מעבר לשדה ההדפסה הבא  
  
Serial.print("HEX");// הדפסת המחרוזת HEX  
Serial.print("\t");// מעבר לשדה ההדפסה הבא  
  
Serial.print("OCT");// הדפסת המחרוזת OCT  
Serial.print("\t");// מעבר לשדה ההדפסה הבא
```

```
Serial.print("BIN"); // הדפסת המחרוזת BIN
Serial.print("\t");// מעבר לשדה ההדפסה הבא

for(x=0; x<64; x++){ // מתחילים להדפיס את המספרים מ 0 ועד 64 בפורמטים שונים

    Serial.print(x); // ( ללא פורמט - מקבלים בעשרוני )
    Serial.print("\t"); // מעבר לשדה ההדפסה הבא

    Serial.print(x,DEC); // הדפסה עשרונית
    Serial.print("\t"); // מעבר לשדה ההדפסה הבא

    Serial.print(x,HEX); // הדפסת ערך ההקסה דצימאלי של המספר
    Serial.print("\t"); // מעבר לשדה ההדפסה הבא

    Serial.print(x,OCT); // הדפסת הערך האוקטאלי
    Serial.print("\t"); // מעבר לשדה ההדפסה הבא

    Serial.println(x,BIN); // הדפסת הערך הבינארי של המספר והורדת הסמן לתחילת השורה הבאה

    // הפונקציה Serial.println ( ) תוסבר בסעיף הבא
    delay(200); // השהייה של 200 מילי שניות
}
Serial.println(""); // הורדת הסמן לתחילת השורה הבאה פעם נוספת
}
```

## 12. Serial.println ( )

כמו הפונקציה שבסעיף הקודם אבל בתוספת של הורדת הסמן לתחילת השורה הבאה. (חיבור הדפסה של הערך 13 או \r קיצור של return והערך 10 שהוא גם \n- קיצור של new line) .

## 13. Serial.read()

הפונקציה קוראת את הנתון הטורי. בארדואינו מגה נציין מהו מספר ערוץ התקשורת הטורית :

Serial1.read()

Serial2.read()

Serial3.read()

הפונקציה איננה מקבלת ערכים ומחזירה את הנתון הטורי שהתקבל. אם לא התקבל נתון היא מחזירה את הערך -1 . דוגמא :



```
IntincomingByte=0;//משתנה שיקבל את הנתון הנקרא בתקשורת טורית
voidsetup(){
Serial.begin(9600);//פתיחת התקשורת הטורית ל 9600 ביטים בשנייה
}

voidloop(){

// שליחת נתון רק שהתקבל נתון
if(Serial.available(>0)){
    incomingByte=Serial.read();//קריאת הביית שהתקבל
    Serial.print("I received: ");// מדפיס "קלטתי "
    Serial.println(incomingByte,DEC); // הערך העשרוני
}
}
```

#### 14 . Serial.readBytes()

הפונקציה קוראת תווים מהפורט הטורי לתוך החוצץ (buffer) . הפונקציה מסתיימת כאשר נקראו כמות התווים הרצויה או שחלף הזמן שנקבע בעזרת הפונקציה Serial.setTimeout( ) (שתוסבר בהמשך) . הפונקציה מחזירה את כמות הבתים שהוכנסו לחוצץ. אם הוחזר 0 זה אומר שאין נתון בחוצץ. היא תורשה של Stream .

התחביר : Serial.readBytes ( buffer , length ) . החוצץ –buffer הוא החוצץ לשמירה של הבתים ב char[ ] או byte[ ] . האורך - length (מטיפוס int) אומר כמה בתים יש לקרא. הפונקציה מחזירה ערך מטיפוס byte שהוא כמות הבתים שנכנסה לחוצץ.

#### 15 . Serial.readBytesUntil()

הפונקציה קוראת תווים מהחוצץ הטורי ומכניסה אל מערך . הפונקציה מסתיימת כאשר התגלה התו הרצוי או שנקראו כמות התווים הרצויה או שהסתיים הזמן שנקבע בפונקציה Serial.setTimeout( ) (שתוסבר בהמשך) . הפונקציה מחזירה את כמות התווים שנקראה לחוצץ. אם מוחזר 0 זה אומר שאין נתון בחוצץ. גם היא ירושה של המחלקה Stream .

התחביר : Serial.readBytesUntil(character, buffer, length) .

character – הוא התו שאותו מחפשים . החוצץ –buffer הוא החוצץ לשמירה של הבתים ב char[ ] או byte[ ] . האורך - length (מטיפוס int) אומר כמה בתים יש לקרא. הפונקציה מחזירה ערך מטיפוס byte שהוא כמות הבתים שנכנסה לחוצץ.

## 16. Serial.setTimeout()

הפונקציה קובעת את מקסימום הזמן ( במילי שניות ) שיש להמתין לנתון טורי כאשר משתמשים בפונקציות Serial.readBytes() או Serial.readBytesUntil(). ברירת המחדל היא 1000 מילי שניות ( 1 שנייה ) . הפונקציה היא תורשה של המחלקה Stream . התחביר : Serial.setTimeout(time) כאשר time הוא הזמן במילי שניות ( מספר טיפוס long ).

## 17. Serial.write()

הפונקציה כותבת נתונים בינאריים לפורט הטורי. הנתון הוא ביית או סדרת בתים. לשליחת תווים מציינים את הספרות של מספר יש להשתמש בפונקציה Serial.print(). התחביר : ( 3 אפשרויות )

Serial.write(val)

Serial.write(str)

Serial.write(buf, len)

הארדואינו מנהל תומך גם ב : Serial1, Serial2, Serial3 במקום Serial . val - הערך שרוצים לשלוח כביית בודד . str – המחזורות שרוצים לשלוח ( סדרה של בתים ) . buf - מערך לשליחה המורכב מסדרת בתים . len - אורך החוצץ . הפונקציה מחזירה ביית המציין כמה בתים נכתבו לחוצץ ( אופציונאלי - לא חובה לקרא את הערך הזה ). דוגמא:

```
voidsetup(){
```

```
  Serial.begin(9600);
```

```
}
```

```
voidloop(){
```

```
  Serial.write(45);// שלח את הערך 45
```

```
  intbytesSent=Serial.write("hello");
```

```
  // bytesSent שליחת המחזורות "hello" וקבלת כמות הבתים שנשלחה למשתנה
```

```
}
```

## 18. serialEvent()

זוהי פונקציה שאותה אנחנו כותבים ואין צורך לקרא לה בתכנית. את הפונקציה רושמים מחוץ לפונקציית ה loop () של תוכנת הארדואינו. בכל פעם שהפונקציה loop() מגיעה לשורה האחרונה שלה וחוזרת לתחילתה התכנית עוברת על הפונקציה SerialEvent(). הפונקציה מראה האם התקבל נתון. אם כן נוכל לקבל את הנתון בעזרת הפונקציה Serial.read(). הפונקציה אין כרגע תאימות עם Esplora, Leonardo, Micro התחביר :

```
void serialEvent(){  
  //פקודות של התכנית  
}
```

לארדואינו מגה :

```
void serialEvent1(){  
  //פקודות של התכנית  
}
```

```
void serialEvent2(){  
  //פקודות של התכנית  
}
```

```
void serialEvent3(){  
  //פקודות של התכנית  
}
```

דוגמא :

```
/*  
  כאשר מגיע נתון טורי, נוסף את הנתון למחרוזת. כאשר יגיע התו newline ('\n') נדפיס את המחרוזת  
  הנקלטת ונבנה את המחרוזת. ניתן למצוא את הדוגמא ב public domain :  
  http://www.arduino.cc/en/Tutorial/SerialEvent  
*/  
string inputString = ""; // הגדרת אובייקט שאליו נכניס את המחרוזת שנקלוט //  
boolean stringComplete = false; // משתנה בוליאני שיראה האם הושלמה קליטת המחרוזת //  
void setup() {  
  Serial.begin(9600); // אתחול התקשורת הטורית //  
  inputString.reserve(200); // קריאה לפונקציה שיודעת לשמור 200 בתים עבור המחרוזת //  
}  
void loop() {  
  if (stringComplete) // המשתנה הזה יקבל 1 שנסיים לקלוט את המחרוזת ואז נדפיס אותה //
```

```
{  
Serial.println(inputString); // הדפסת המחרוזת שקלטנו  
inputString = ""; // ניקוי המחרוזת  
stringComplete = false;  
}  
}
```

```
/*
```

הפונקציה SerialEvent רצה בין כל ריצת הפונקציה loop . היא קורית כאשר יש נתון חדש בכניסת החומרה של הדק RX . אם נשתמש בפונקציה השהייה בתכנית ה loop זה יגרום גם להשהייה בתגובה של הפונקציה שלנו. ניתן לקבל בתים רבים בעזרת הפונקציה הזו.

```
*/
```

```
void serialEvent() // הגדרה של הפונקציה  
{  
while (Serial.available()) {  
char inChar = (char)Serial.read(); // קבלת נתון חדש  
inputString += inChar; //inputString באובייקט הוספת הנתון החדש  
if (inChar == '\n') // האם הגיע התו ? newline  
{  
stringComplete = true;  
}  
}  
}
```