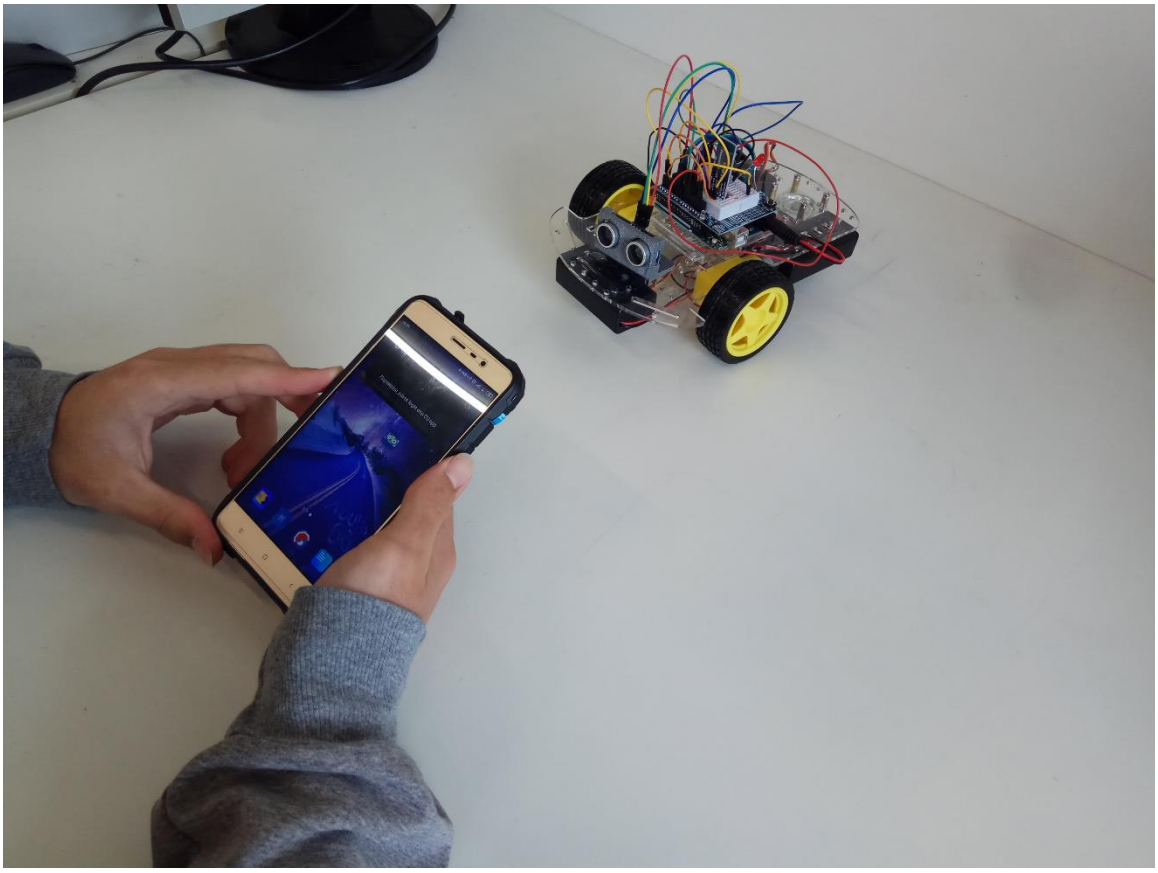
ΔΙΠΛΗ ΛΕΙΤΟΥΡΓΙΑ ΡΟΜΠΟΤ (Auto & Manual)



Το επόμενο βήμα είναι ο συνδυασμός των δύο προηγούμενων. Δηλαδή να προγραμματίσουμε το ρομπότ ώστε να μπορεί να μεταβεί από την Auto στην Manual λειτουργία με το πάτημα ενός κουμπιού στην android εφαρμογή.

Αρχικά είναι επιλεγμένη η manual λειτουργία ώστε ενεργοποιώντας την Bluetooth επικοινωνία, να έχουμε τον έλεγχο του ρομπότ.

Ο κώδικας θα παρατηρήσετε ότι είναι η σύνδεση των δύο προηγούμενων.

Το βιντεάκι μπορείτε να το δείτε στον παρακάτω σύνδεσμο: https://youtu.be/HesnH83gSKE

Φύλλο εργασίας

ΔΙΠΛΗ ΛΕΙΤΟΥΡΓΙΑ ΡΟΜΠΟΤ (Auto & Manual)

Το επόμενο βήμα είναι ο συνδυασμός των δύο προηγούμενων. Δηλαδή να προγραμματίσετε το ρομπότ ώστε να μπορεί να μεταβεί από την Auto στην Manual λειτουργία με το πάτημα ενός κουμπιού στην android εφαρμογή.

Αρχικά να είναι επιλεγμένη η manual λειτουργία ώστε ενεργοποιώντας την Bluetooth επικοινωνία, να έχουμε τον έλεγχο του ρομπότ.

Ο κώδικας θα πρέπει να είναι η σύνδεση των δύο προηγούμενων βημάτων: της Auto και της Manual λειτουργίας.

Προγραμματίζοντας:

- Να γίνει ο συνδυασμός των προγραμμάτων, που χρησιμοποιήσατε στα δύο προηγούμενα βήματα
- Εισάγετε μια επιπλέον μεταβλητή τύπου χαρακτήρα (char) για να δηλώνετε το mode του ρομπότ. Για παράδειγμα:

if (state == 'A'){mode = 'a'; } **για Auto λειτουργία**

if (state == 'M'){ mode = 'm'} **για manual λειτουργία**

θα πρέπει λοιπόν στη συνέχεια να αναλύσετε χρησιμοποιώντας τη συνάρτηση **if,** τι θα συμβαίνει σε κάθε mode.

Ο κώδικας

```
//Σχετικά με τα μοτέρ:

//Motor A

const int motorR1  = 9;  //ΔΕΞΙ ΜΠΡΟΣΤΑ

const int motorR2  = 10;//ΔΕΞΙ ΠΙΣΩ

//Motor B

const int motorL1  = 5; //ΑΡΙΣΤΕΡΟ ΜΠΡΟΣΤΑ

const int motorL2  = 6;//ΑΡΙΣΤΕΡΟ ΠΙΣΩ

 //Σχετικά με τον αισθητήρα απόστασης:

 #include <Ultrasonic.h>

 //Ultrasonic      (Trig,Echo)
```

```
 Ultrasonic sensor (A0,A1);
 // μεταβλητές για να αποθηκεύεται η απόσταση που μετρά ο αισθητήρας απόστασης
 int  distance=100;
 int leftDistance;
 int rightDistance;
//Σχετικά με τον servo που χρησιμοποιούμε ως "λαιμό" του ρομπότ
 #include <Servo.h>
 Servo myservo1;  // create servo object to control a servo
 int pos1 ;   // μεταβλητή για να αποθηκεύεται η θέση του  servo
//Led
const int led = 12;
//Speaker
const int speaker = 13;
int i=0;
int j=0;
//Μεταβλητές για το bluetooth
int state;
 char mode='m';
 void setup(){
  //Ξεκινά η σειριακή επικοινωνία,Start serial communication
 Serial.begin(9600);
   //θέτω ως εξόδους τους παρακάτω ακροδέκτες(Set pins as outputs)
   pinMode(motorR1, OUTPUT);
   pinMode(motorR2, OUTPUT);
   pinMode(motorL1, OUTPUT);
   pinMode(motorL2, OUTPUT);
```

```
  pinMode(led, OUTPUT);
  pinMode(speaker, OUTPUT);

 //Σχετικά με την αρχική θέση του servo1(positioning servo1)
 myservo1.attach(8);  // συνδέεται ο servo 1  με το pin 8 του arduino
 myservo1.write(90);// ο servo 1 κοιτάζει μπροστά

 // Χωρίς ήχο αρχικά(no sound in the begining)
 noTone(speaker);

 //Διάρκεια αρχικοποίησης 1sec, time for the setup is 1sec
 delay(1000);   }

void loop(){
 //αν η σειριακή επικοινωνία είναι δυνατή
if (Serial.available() > 0) {
 // read the incoming byte:
 state = Serial.read();
 if (state == 'A'){mode = 'a'; }
 if (state == 'M'){ mode = 'm';
  stop();
  pos1=90;
   myservo1.write(pos1);
   delay(100);     }

   //Bluetooth mode
   if (mode == 'm'){
```

```
if (state=='F'){ forward();    }
else if (state=='B'){      backward();    }
else if (state=='R'){ right(); }
else if (state=='L'){left(); }
else if (state=='S'){stop(); }

//If state is equal with letter 'W', turn leds on or of off
else if (state == 'W') {if (i==0){ digitalWrite(led, HIGH); i=1; }
  else { digitalWrite(led, LOW); i=0;}
 state='n'; }

 //If state is equal with letter 'V', play (or stop) horn sound
 else if (state == 'V'){if (j==0){ tone(speaker, 1000);//Speaker on
    j=1;}
  else { noTone(speaker);    //Speaker off
    j=0; }
  state='n'; }
  }
```

//Automatic mode

```
else if (mode == 'a'){
distance = sensor.Ranging(CM);
//If an object detected at 10cm, stop the robot and find a way out
if (distance <= 10){
 stop();
 tone(speaker, 1000);
 digitalWrite(led,HIGH);
```

```
delay(1000);
noTone(speaker);
digitalWrite(led,LOW);
//Look left
for (pos1 = 90; pos1 <= 180; pos1 += 1) { // goes from 0 degrees to 180 degrees
myservo1.write(pos1);              // tell servo to go to position in variable 'pos'
delay(15);                       // waits 15ms for the servo to reach the position
}
leftDistance = sensor.Ranging(CM);
delay(100);

 //Now look right
 for (pos1 = 180; pos1 >= 0; pos1 -= 1) { // goes from 180 degrees to 0 degrees
 myservo1.write(pos1);              // tell servo to go to position in variable 'pos'
 delay(15);                       // waits 15ms for the servo to reach the position
   }
rightDistance = sensor.Ranging(CM);
delay(100);
pos1 = 90; // look forward again
myservo1.write(pos1);

//Finally compare left and right distances and make the best turn decision
if (leftDistance > rightDistance){
left();
delay(250); // Change the time to make 90 deg. turn
   }
else if (leftDistance < rightDistance){
```

```
   right();
  delay(250);
      }
  else{ //that means that two distances are equal
  backward();
  delay(1000);// Go back for 1 sec
  left();   // and turn left
  delay(250);}
   }
   //All crear, move forward!
   else{ forward(); }
   }}}
```

//Movement functions

```
void forward(){
digitalWrite(motorR1, HIGH);
 digitalWrite(motorR2, LOW);
 digitalWrite(motorL1, HIGH);
 digitalWrite(motorL2, LOW);}

void backward(){
 digitalWrite(motorR1, LOW);
 digitalWrite(motorR2, HIGH);
 digitalWrite(motorL1, LOW);
 digitalWrite(motorL2, HIGH);}

void right(){
```

```
  digitalWrite(motorR1, LOW);
  digitalWrite(motorR2, HIGH);
  digitalWrite(motorL1, HIGH);
  digitalWrite(motorL2, LOW);  }

void left(){
  digitalWrite(motorR1, HIGH);
  digitalWrite(motorR2, LOW);
  digitalWrite(motorL1,LOW );
  digitalWrite(motorL2, HIGH);}

void stop(){
  digitalWrite(motorR1, LOW);
  digitalWrite(motorR2, LOW);
  digitalWrite(motorL1, LOW);
  digitalWrite(motorL2, LOW);}
```