

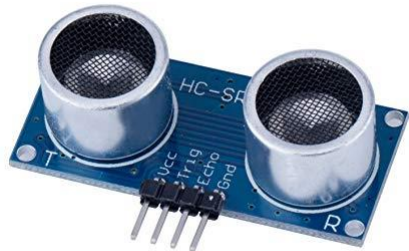
# Arduino Workshop – Intermediate Level

## *Day 2*

# Περιεχόμενο 2<sup>ης</sup> ημέρας – Συνοπτική αναφορά

*Ο εξοπλισμός που θα χρησιμοποιήσουμε*

## Ultrasonic



## Serial Communication



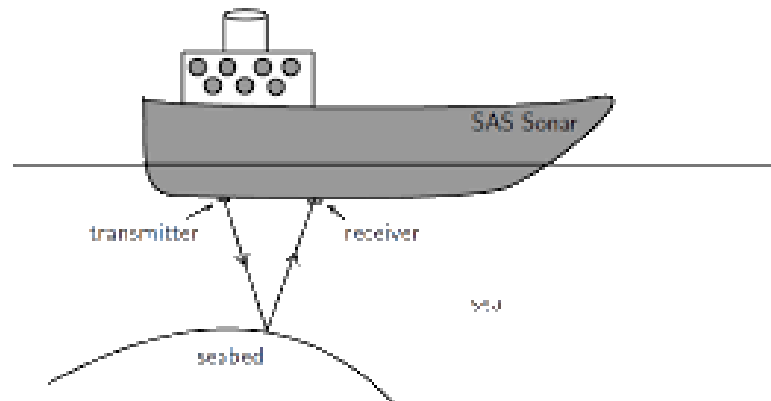
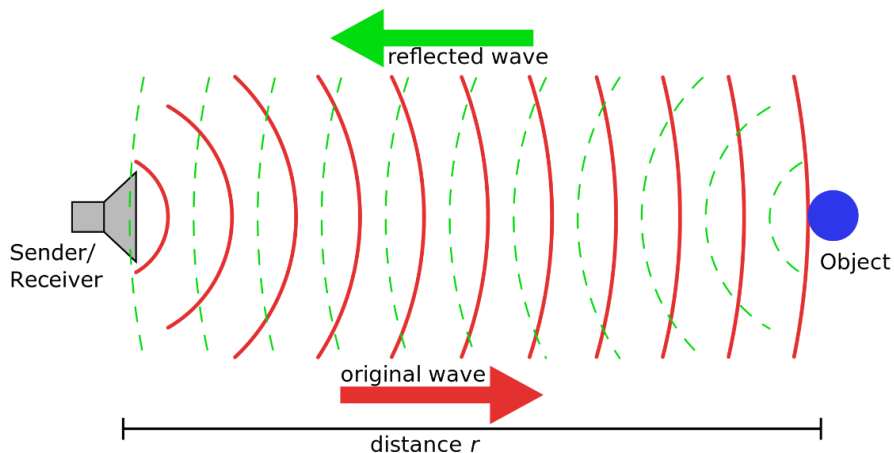
## Servo Motor



# 0 Ultrasonic Sensor

*Τι είναι και που χρησιμοποιείται;*

- Ως αισθητήρας, η δουλειά του είναι να στέλνει και λαμβάνει ένα υπερηχητικό κύμα, μετρώντας απόσταση βάσει του χρόνου επιστροφής.



- Χρησιμοποιείται όπου απαιτείται ο υπολογισμός απόστασης:
  - Σε κινητά ρομπότ για ανίχνευση εμποδίων.
  - Για ανίχνευση στάθμης νερού.
  - Σε αισθητήρες στάθμευσης.

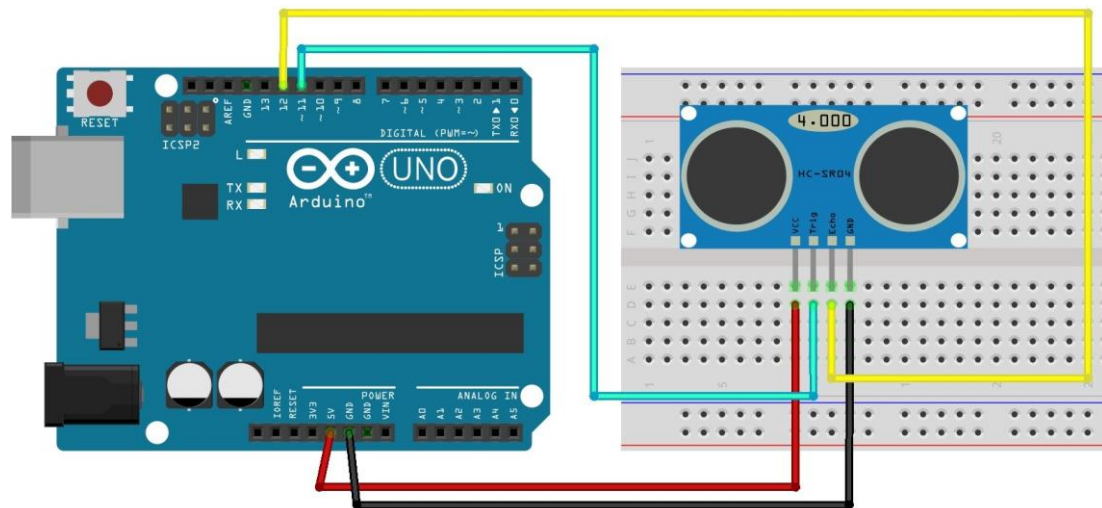
# Μια Γρήγορη Επανάληψη

## *Ultrasonic Sensor – Hardware*

➤ Διαθέτει 4 άκρα:

- **Vcc** (5V)
- **Trigger** (digital pin)
- **Echo** (digital pin)
- **Ground**

➤ Μέγιστη απόσταση: 2 μέτρα.



# Μια Γρήγορη Επανάληψη

## *Ultrasonic Sensor – Software*

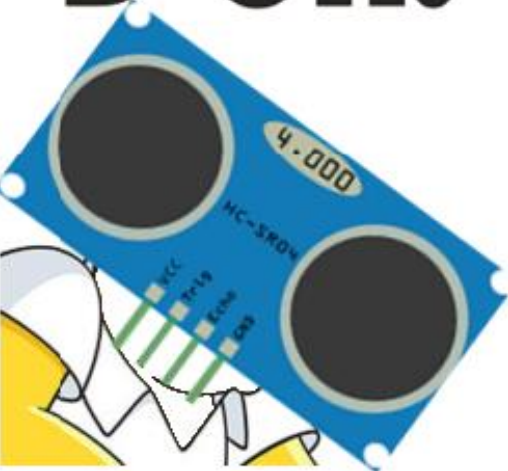
```
1 // Library
2 #include <NewPing.h>
3
4 // Object of class
5 NewPing sonar(TRIGGER, ECHO, MAX_DISTANCE);
6
7 // Getting distance
8 sonar.ping_cm();
```

# The Ultrasonic Problem

*Ανάθεμα τα ηχητικά κύματα!*

Πολλές φορές ο αισθητήρας λαμβάνει μη-αληθείς τιμές απόστασης λόγω ανακλώμενων (από άλλες πηγές) ηχητικών κυμάτων, ηλεκτρικού θορύβου ή η μετρούμενη απόσταση γίνεται (αιφνίδια και βραχυπρόθεσμα) πολύ μεγάλη.

# D'OH!



## ? Quiz:

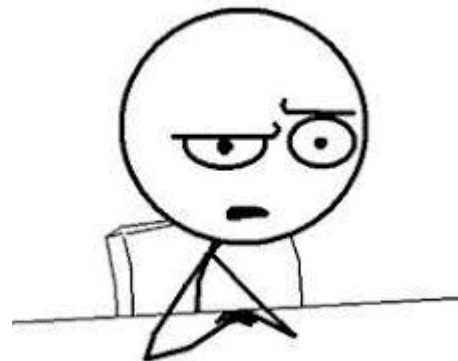
Πως μπορούμε να λύσουμε προγραμματιστικά το πρόβλημα?

# Time Triggers

Πάλι *millis()* αντί για *delay()*?!?

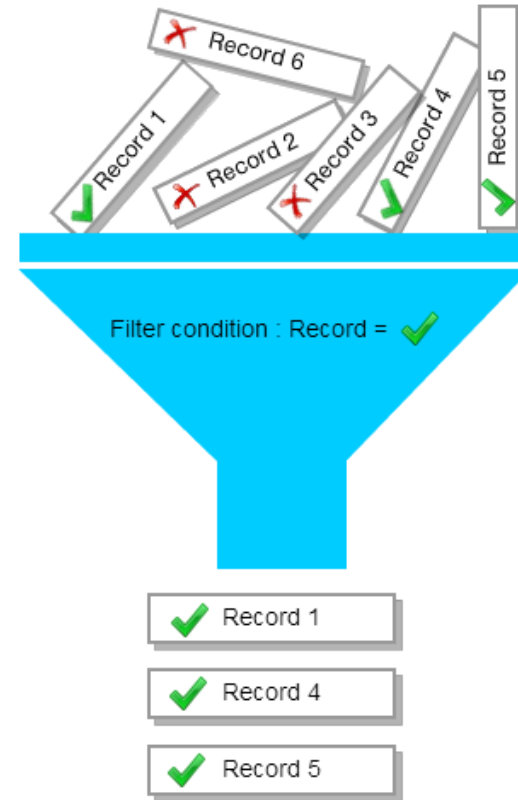
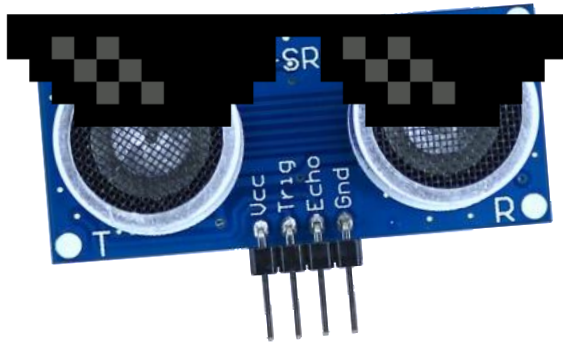
- ▶ Έστω ότι θέλουμε να βάλουμε *delay(100)*, αηλιά χωρίς το *delay*...
- ▶ Ακολουθούμε την παρακάτω μεθοδολογία:

```
unsigned long previousTime = 0;
unsigned long currentTime = millis();
if ((currentTime - previousTime) > 100){
    previousTime = currentTime;
    // More code
}
```



# Project 1 – {U L T R A} Sonic

*Φιλτράρετε τις τιμές του ultrasonic ώστε να απορρίπτονται πολύ απότομες μεταβολές (spikes) στις τιμές που επιστρέφει.*





# Project 1 – {U L T R A} Sonic

*Φιλτράρετε τις τιμές του ultrasonic ώστε να απορρίπτονται πολύ απότομες μεταβολές (spikes) στις τιμές που επιστρέφει*

```
1 #include <NewPing.h>
2 NewPing sonar(12,13,200);
3
4 // Ultrasonic Values
5 int new_reading = 0;
6 int new_trust = 0;
7 int current_reading = 0;
8 bool current_trust = 0;
9 int last_reading = 0;
10 bool last_trust = 0;
11 int second_last_reading = 0;
12 bool second_last_trust = 0;
```

```
13
14 void setup() {
15     Serial.begin(9600);
16 }
17
18 void loop() {
19     getReading();
20     int dist = getDistance();
21 }
22
```



# Project 1 – {U L T R A} Sonic

*Φιλτράρετε τις τιμές του ultrasonic ώστε να απορρίπτονται πολύ απότομες μεταβολές (spikes) στις τιμές που επιστρέφει*

```
23 // Time Trigger Variables
24 unsigned long previousTime = 0;
25 unsigned long currentTime = 0;
26 void getReading() {
27     currentTime = millis();
28     if ((currentTime - previousTime) > 50) {
29         previousTime = currentTime;
30         new_reading = sonar.ping_cm();
31         new_trust = 0;
32         applyFilters();
33         Serial.println(getDistance());
34     }
35 }
```

```
37 int getDistance() {
38     int sum = 0;
39     int counter = 0;
40     if (current_trust == 1) {
41         sum += current_reading;
42         counter++;
43     }
44     if (last_trust == 1) {
45         sum += last_reading;
46         counter++;
47     }
48     if (second_last_trust == 1) {
49         sum += second_last_reading;
50         counter++;
51     }
52     int avg = sum/counter;
53     return avg;
54 }
```

# Project 1 – {U L T R A} Sonic

*Φιλτράρετε τις τιμές του ultrasonic ώστε να απορρίπτονται πολύ απότομες μεταβολές (spikes) στις τιμές που επιστρέφει*

```
56 void rotateValues() {  
57     second_last_reading = last_reading;  
58     second_last_trust = last_trust;  
59     last_reading = current_reading;  
60     last_trust = current_trust;  
61     current_reading = new_reading;  
62 }
```

```
64 void changeTrustFactor() {  
65     if (current_trust==0 && last_trust==0  
66         && second_last_trust==0) {  
67         current_trust = 1;  
68         last_trust = 1;  
69         second_last_trust = 1;  
70     }  
71 }
```

# Project 1 – {U L T R A} Sonic

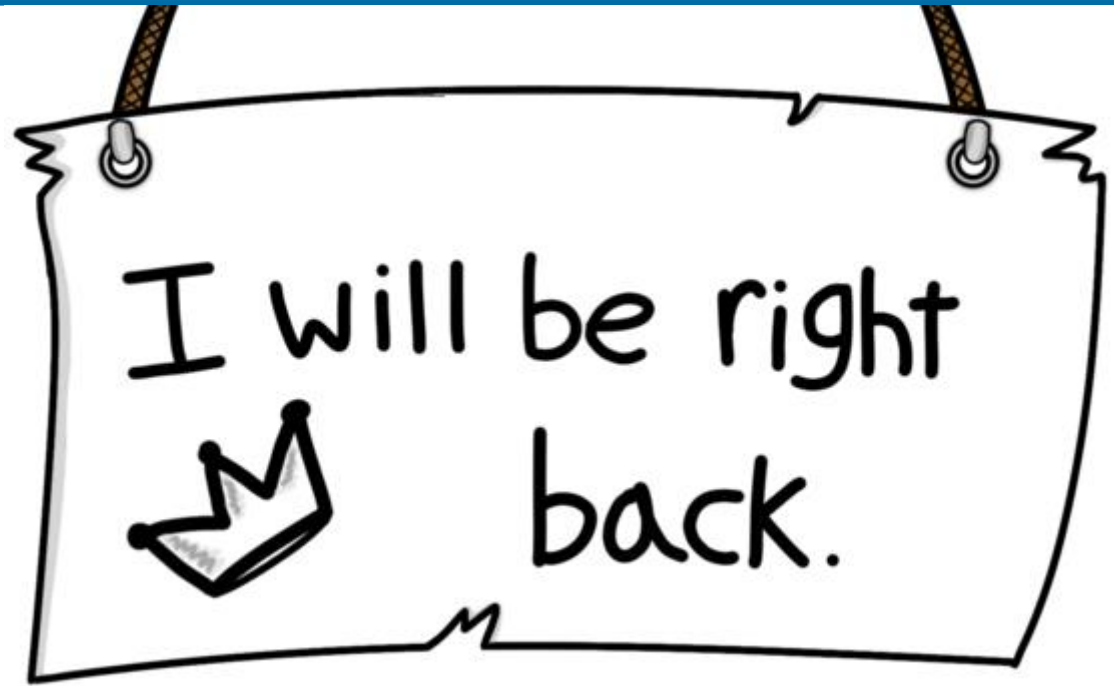
*Φιλτράρετε τις τιμές του ultrasonic ώστε να απορρίπτονται πολύ απότομες μεταβολές (spikes) στις τιμές που επιστρέφει*

```
1 #define tolerance 10
2 bool filter(int value, int timee){
3     int x = new_reading - value;
4     if (abs(x) < (tolerance*timee)){
5         new_trust++;
6         return 0;
7     } else {
8         new_trust--;
9         return 1;
10    }
11 }
12
```

```
13 void applyFilters(){
14     changeTrustFactor();
15     if (current_trust == 1){
16         bool flag1 = filter(current_reading,1);
17     }
18     if (last_trust == 1){
19         bool flag2 = filter(last_reading,2);
20     }
21     if (second_last_trust == 1){
22         bool flag3 = filter(second_last_reading,3);
23     }
24     if (new_trust > 0){
25         rotateValues();
26         current_trust = 1;
27     } else if (new_trust < 0){
28         rotateValues();
29         current_trust = 0;
30     }
31     changeTrustFactor();
32 }
```

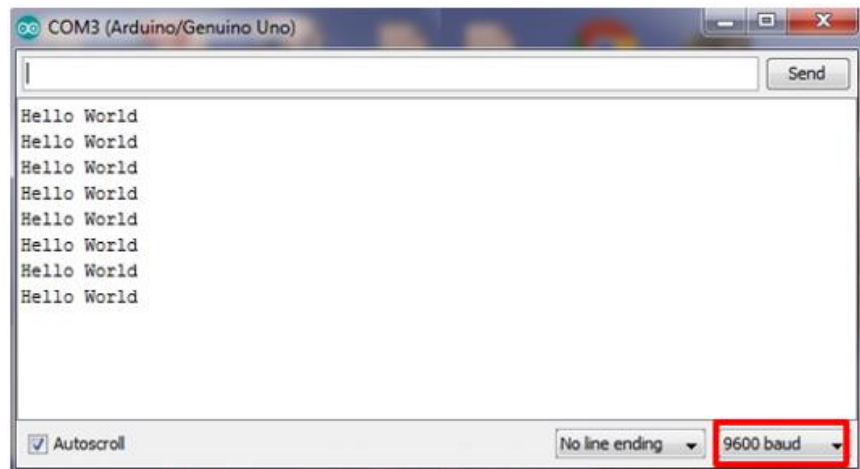
# Τέλος Α' Μέρους

*Καλό διάλειμμα!*



# Serial Communication στο Περιβάλλον IDE

*Για επικοινωνία Arduino - υπολογιστή*



# Μια Γρήγορη Επανάληψη

*Σειριακή επικοινωνία – πως δουλεύει;*

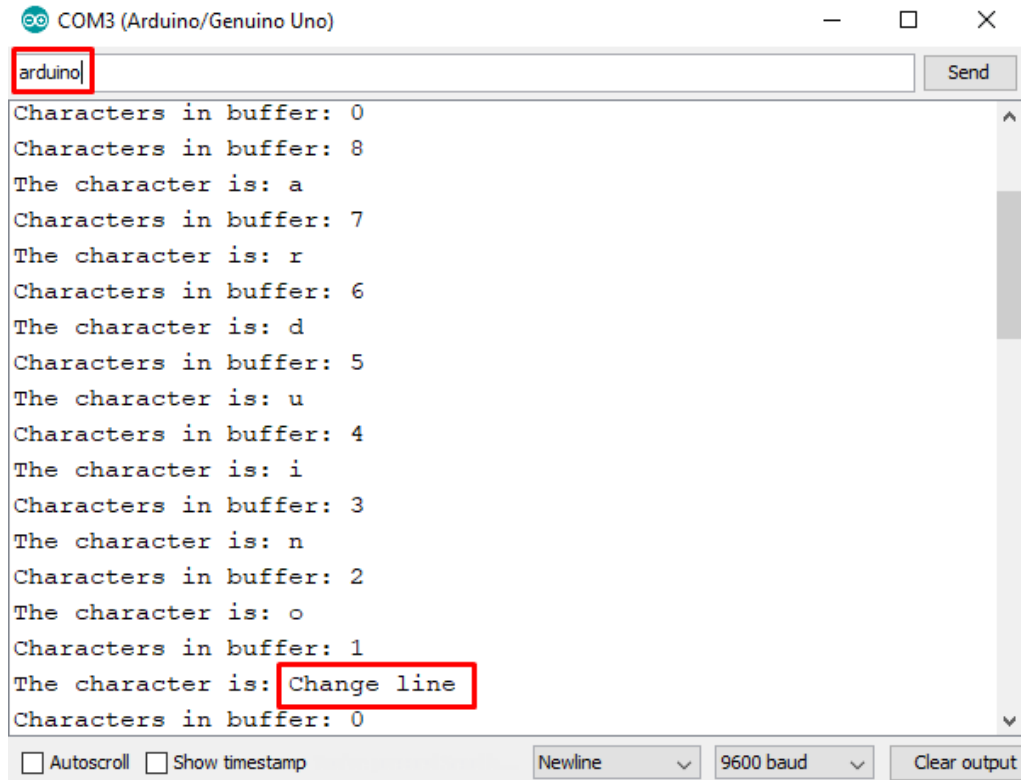
- ▶ Ορίζουμε στη `setup()` την εκκίνηση της σειριακής επικοινωνίας με την *επιθυμητή ταχύτητα*.
- ▶ Η `Serial.available()`; επιστρέφει τον αριθμό των χαρακτήρων που περιμένουν να αναγνωστούν **συν ένας**.
- ▶ Η `Serial.read()`; επιστρέφει τον **χαρακτήρα** του οποίου είναι η σειρά να αναγνωστεί.

```
1 void setup() {  
2     Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6     char c;  
7     if (Serial.available()) {  
8         c = Serial.read();  
9         // Commands  
10    }  
11 }
```

# Μια Γρήγορη Επανάληψη

*Ο χαρακτήρας αλλαγής σειράς.*

```
5 void loop() {  
6   char c;  
7   int x = Serial.available();  
8   Serial.print("Characters in buffer: ");  
9   Serial.println(x);  
10  if (x){  
11    c = Serial.read();  
12    Serial.print("The character is: ");  
13    if (c == '\n'){  
14      Serial.println("Change line");  
15    }else{  
16      Serial.println(c);  
17    }  
18  }  
19  delay(1000);  
20 }
```



COM3 (Arduino/Genuino Uno)

arduino Send

Characters in buffer: 0  
Characters in buffer: 8  
The character is: a  
Characters in buffer: 7  
The character is: r  
Characters in buffer: 6  
The character is: d  
Characters in buffer: 5  
The character is: u  
Characters in buffer: 4  
The character is: i  
Characters in buffer: 3  
The character is: n  
Characters in buffer: 2  
The character is: o  
Characters in buffer: 1  
The character is: Change line  
Characters in buffer: 0

☐ Autoscroll ☐ Show timestamp Newline 9600 baud Clear output



# Μια Γρήγορη Επανάληψη

*Ο χαρακτήρας αλληλαγής σειράς.*

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   char c;  
7   if (Serial.available()){  
8     c = Serial.read();  
9     if (c == '\n'){  
10      // Do nothing  
11    } else {  
12      // Commands  
13    }  
14  }  
15 }
```

## ➤ Getting input (fixed):

- Πρέπει να απορρίψουμε τον επιπλέον χαρακτήρα κατά την διαδικασία της ανάγνωσης
- Άρα θα εκτελέσουμε τις εντολές μας μόνο αν αυτά που διαβάσαμε δεν είναι η αλληλαγή γραμμής

## ? Quiz:

Πώς μπορούμε να διαβάσουμε μια ολόκληρη λέξη (ή αριθμό με 2+ ψηφία) αντί για 1 – 1 χαρακτήρα?

# Μια Γρήγορη Επανάληψη

*Μετατροπή χαρακτήρων εισόδου σε αριθμούς*

## Πως μετατρέπουμε το '1' σε 1:

- Ο κάθε χαρακτήρας αντιστοιχεί σε μια αριθμητική τιμή
- Οι χαρακτήρες των νούμερων είναι κωδικοποιημένοι σε διαδοχικούς αριθμούς, άρα αφαιρούμε από τον χαρακτήρα '1' τον χαρακτήρα '0' και παίρνουμε καθαρή (ακέραια) τιμή 1

COM3 (Arduino/Genuino Uno)

```
The character is: 1  
The numeric value of '1' is: 49  
The numeric value of '0' is: 48  
'1' minus '0' is: 1
```

☒ Autoscroll ☐ Show timestamp

# Μια Γρήγορη Επανάληψη

*Μετατροπή χαρακτήρων εισόδου σε αριθμούς*

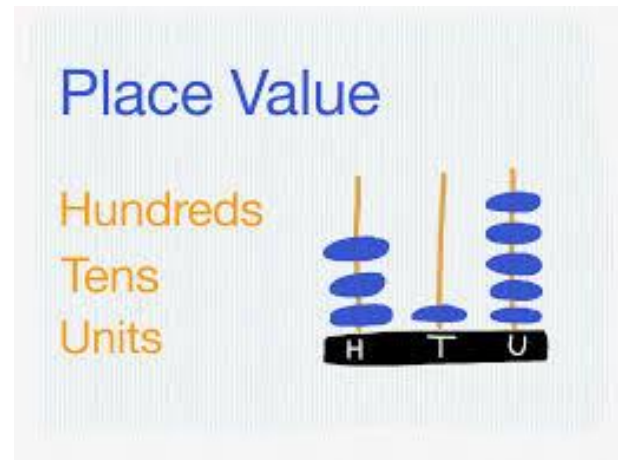
```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   int x;  
7   if (Serial.available()) {  
8     char c = Serial.read();  
9     if (c == '\n') {  
10      // Do nothing  
11    } else if ((c >= '0') && (c <= '9')) {  
12      x = c - '0';  
13      // Commands using integer x  
14    }  
15  }  
16 }
```

# Μια Γρήγορη Επανάληψη

*Για αριθμούς με πάνω από 1 ψηφία*

## Πως μετατρέπουμε το “180” σε 180:

- Πρέπει να διαβάσουμε 3 χαρακτήρες: ‘1’ , ‘8’ , ‘0’ και να τους ενώσουμε στο νούμερο 180 κατά την διαδικασία της ανάγνωσης
- Άρα πριν διαβαστεί ο επόμενος χαρακτήρας κάθε φορά, ο προηγούμενος πολλαπλασιάζεται με το 10
- Όταν δεν υπάρχουν άλλοι χαρακτήρες που περιμένουν να διαβαστούν, η διαδικασία σταματάει



## Project 2 – Hear me and Obey

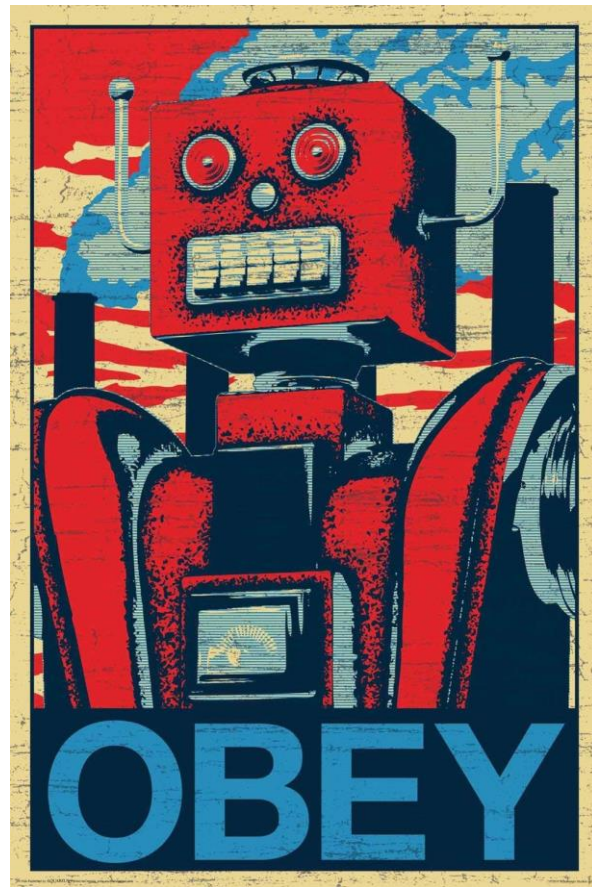
*Προγραμματίστε στο Arduino τις συναρτήσεις  
readSerial(); , read\_Num(); και read\_Text(); .*

*Οι 2 τελευταίες θα καλούνται μέσα στην  
readSerial(); αναλόγως με το τι είναι ο  
επόμενος χαρακτήρας.*

*Η ανάγνωση μίας εντολής θα τελειώνει με τον  
χαρακτήρα ';' .*

### ! Hint:

Μην είστε εξ' αρχής αυστηροί με τις λάθος εισόδους. Διορθώνεται και μετά, αρκεί να δουλέψει πρώτα ;)



# Project 2 – Hear me and Obey

```
1 int number_out = 0;
2 String text_out = "";
3
4 void setup() {
5     Serial.begin(9600);
6 }
7
8 void loop() {
9     if (readSerial()){
10         Serial.println("A command was read!");
11         execute();
12     }
13 }
```

```
15 bool readSerial(){
16     if (Serial.available()){
17         char char_in = Serial.read();
18         if (char_in == '\n'){
19             // Ignore
20         } else if ((char_in>='0' && char_in<='9') || char_in=='-'){
21             read_Num(char_in);
22         } else if (char_in == ';'){
23             set_Command();
24             return true;
25         } else {
26             read_Text(char_in);
27         }
28     }
29 }
30 return false;
31 }
```

# Project 2 – Hear me and Obey

```
1 int number_in = 0;
2 int sign = 1;
3 void read_Num(char c){
4     if (c == '-') {
5         sign = -1;
6     } else {
7         number_in = number_in*10 + (c - '0');
8     }
9 }
10
```

```
11 String text_in = "";
12 void read_Text(char c){
13     text_in += ((String)c);
14 }
15
16 void set_Command(){
17     number_out = number_in * sign;
18     number_in = 0;
19     sign = 1;
20     text_out = text_in;
21     text_in = "";
22 }
```

# Project 2 – Hear me and Obey

```
1 void execute() {  
2     if (text_out == "") {  
3         Serial.println("Empty command!");  
4     } else {  
5         Serial.print("The command was: ");  
6         Serial.print(text_out);  
7         Serial.print(" ");  
8         Serial.println(number_out);  
9     }  
10 }
```



# Servo Motor

*Τι είναι και πού χρησιμοποιείται;*

- Είναι πρακτικά κινητήρας, αλλιά έχει περιορισμένη γωνία περιστροφής (0 – 180 μοίρες) και ανάλογα με την είσοδο που δέχεται ακινητοποιείται στην αντίστοιχη θέση.



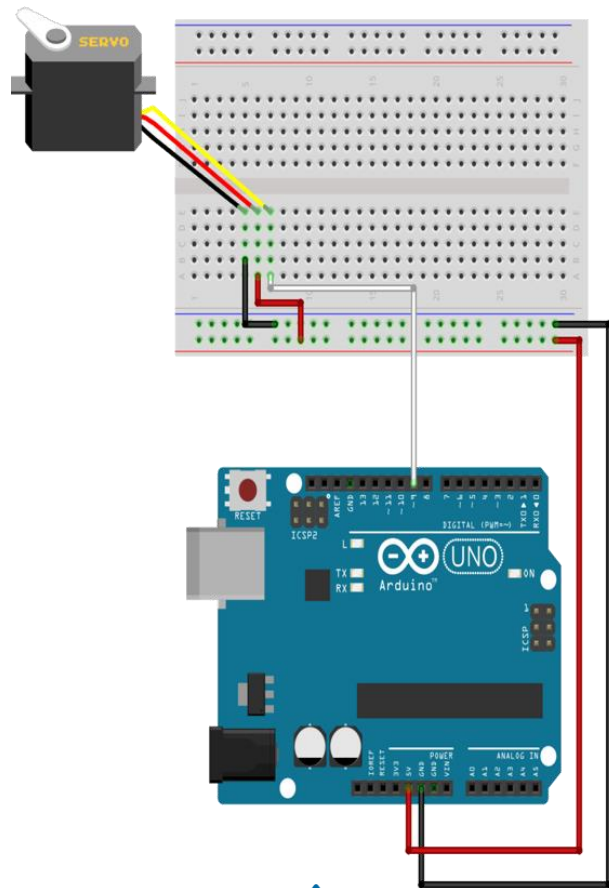
- Χρησιμοποιούνται σε:
  - Αρθρώσεις (βραχίονες)
  - όπου απαιτούνται μικρές ελεγχόμενες κινήσεις (στροφή μπροστινών τροχών αυτοκινήτων)
  - διάφορους άλλους αυτοματισμούς (περιστροφή solar panel)

# Μια Γρήγορη Επανάληψη

## *Servo motor – Hardware*

Τα Servo έχουν 3 σημεία σύνδεσης:

- **Κόκκινο** (μεσαίο) καλώδιο:
  - Τροφοδοσία (5V)
- **Μαύρο** ή **καφέ** καλώδιο:
  - Γείωση (GND)
- **Κίτρινο** ή **πορτοκαλί** καλώδιο:
  - Για τον έλεγχο του Servo
  - Σε οποιοδήποτε pin του Arduino



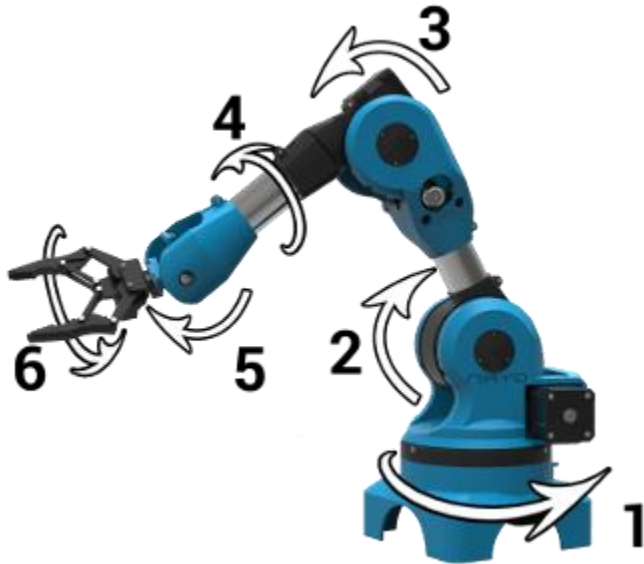
# Μια Γρήγορη Επανάληψη

## *Servo motor – Software*

```
1 // Library
2 #include <Servo.h>
3
4 // Object of class
5 Servo myservo;
6
7 // Defining control pin
8 myservo.attach(8);
9
10 // Setting angle
11 myservo.write(90);
```

# Servo και βαθμοί ελευθερίας

*Βραχίονες και συντεταγμένες*



- Στην μηχανική, κάθε servo συνεπάγεται και ένα βαθμό ελευθερίας
- Με 3 servo έχουμε 3 βαθμούς ελευθερίας, με άλλα λόγια ένα “mini” βραχίονα που μπορεί να κινείται στον χώρο.
- Αυτή η κίνηση όμως προϋποθέτει είσοδο με συντεταγμένες ( $x - y - z$ ), που θα μεταφράζεται σε μοίρες από τον κώδικα.

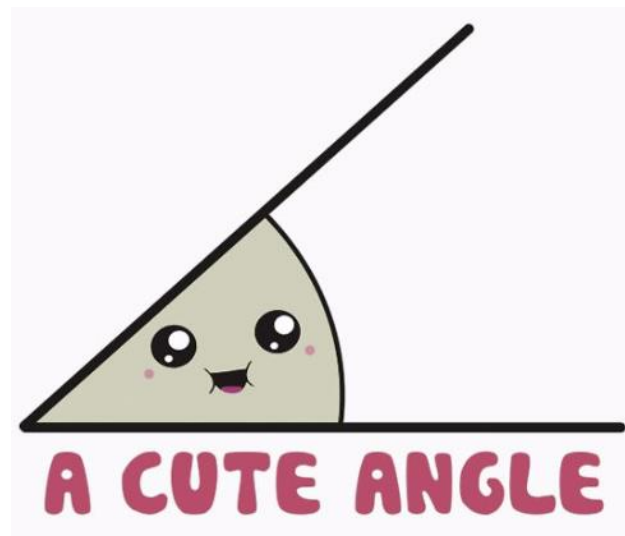
## Project 3 – Angles are fine, coordinates are better

*Προγραμματίστε τον βραχίονα έτσι ώστε να πηγαίνει(η άκρη του) στις συντεταγμένες που θα του δίνουμε μέσω σειριακής επικοινωνίας. Αν το σημείο είναι εκτός της εμβέλειάς του να τυπώνει το αντίστοιχο μήνυμα λάθους στο Serial Monitor*

### ! Hint:

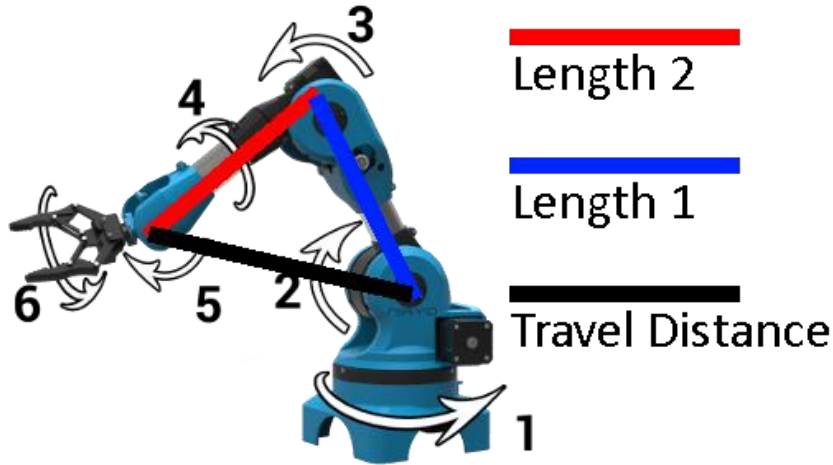
Η μετατροπή απαιτεί τριγωνομετρία.  
Μπορείτε να ανατρέξετε στον παρακάτω σύνδεσμο για βοήθεια:

<https://www.calculator.net/triangle-calculator.html>



# Υπολογισμός γωνιών από τις πλευρές του τριγώνου

Ξέρουμε το μήκος του κάθε μπράτσου του βραχίονα, αλλιώς και την συνολική απόσταση που θέλουμε να φτάσει



Length 2

Length 1

Travel Distance

Side a = 4

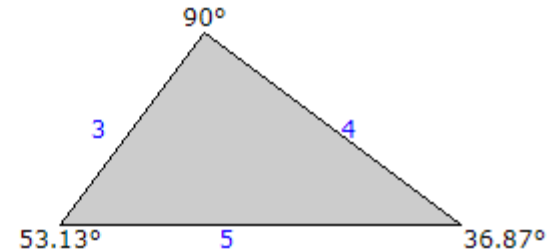
Side b = 3

Side c = 5

Angle  $\angle A = 53.13^\circ$

Angle  $\angle B = 36.87^\circ$

Angle  $\angle C = 90^\circ$



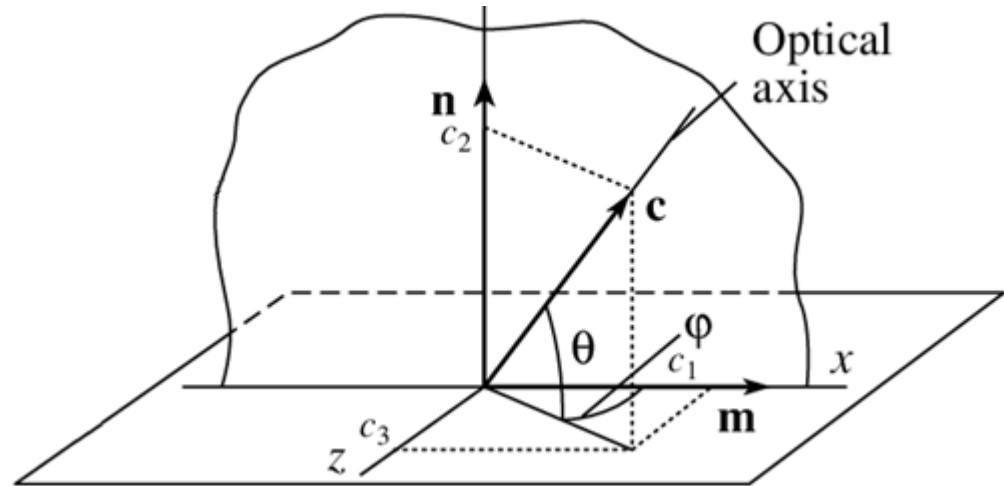
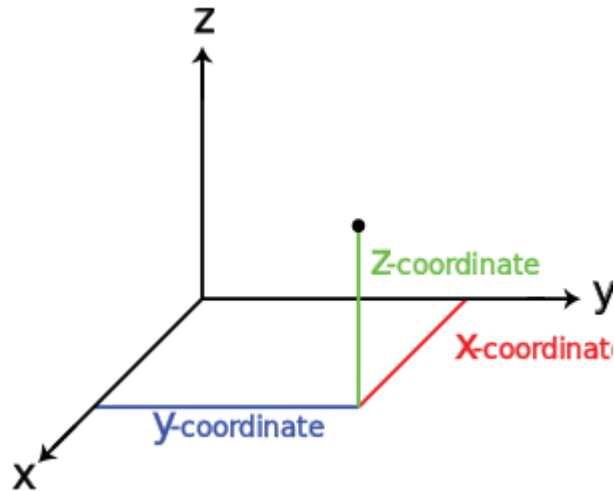
$$\angle A = \arccos\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$

$$\angle B = \arccos\left(\frac{a^2 + c^2 - b^2}{2ac}\right)$$

$$\angle C = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

# Υπολογισμός γωνιών από τις πλευρές του τριγώνου

Αν το *travel distance* είναι το  $c$ , τότε μόλις υπολογίσουμε τις προηγούμενες γωνίες προσθέτουμε την  $\phi$  στο *servo1* και την  $\theta$  στο *servo2* και τελειώσαμε!



# Project 3 – Angles are fine, coordinates are better

```
1 #include <Servo.h>
2 Servo Servo1, Servo2, Servo3;
3 // Servo pins
4 #define pin1 3
5 #define pin2 4
6 #define pin3 5
7 // Servo angles
8 double angle1 = 90;
9 double angle2 = 90;
10 double angle3 = 90;
11 // Arm lengths
12 long length1 = 55; // mm
13 long length2 = 65; // mm
14
15 // Cartecian Values
16 long x = 120;
17 bool x_flag = 0;
18 long y = 0;
19 bool y_flag = 0;
20 long z = 0;
21 bool z_flag = 0;
22
23 void setup() {
24     // Initializing Servos
25     Servo1.attach(pin1, 550, 2450);
26     Servo2.attach(pin2, 550, 2450);
27     Servo3.attach(pin3, 550, 2450);
28     // Initial arm position
29     Servo1.write((int)angle1);
30     Servo2.write((int)angle2);
31     Servo3.write((int)angle3);
32     // Serial communication
33     Serial.begin(9600);
34 }
35
```



# Project 3 – Angles are fine, coordinates are better

```
36 void loop() {  
37   if (readSerial()) {  
38     execute();  
39   }  
40   if (x_flag==1 && y_flag==1 && z_flag==1) {  
41     x_flag = 0;  
42     y_flag = 0;  
43     z_flag = 0;  
44     Serial.println("Moving arm to coordinates");  
45     MoveServos();  
46   }  
47 }
```

```
1 template <typename type>  
2 type sign(type value) {  
3   return type((value>0)-(value<0));  
4 }  
5
```

## Project 3

```
1 bool readSerial() {
2     if (Serial.available()) {
3         char char_in = Serial.read();
4         if (char_in == '\n') {
5             // Ignore
6         } else if ((char_in >= '0' && char_in <= '9')
7             || char_in == '-') {
8             read_Num(char_in);
9         } else if (char_in == ';') {
10            set_Command();
11            return true;
12        } else {
13            read_Text(char_in);
14        }
15    }
16    return false;
17 }
```

```
19 int number_in = 0;
20 int number_sign = 1;
21 void read_Num(char c) {
22     if (c == '-') {
23         number_sign = -1;
24     } else {
25         number_in = number_in * 10 + (c - '0');
26     }
27 }
```

# Project 3

```
29 String text_in = "";
30 void read_Text(char c){
31     text_in += ((String)c);
32 }
33
34 int number_out = 0;
35 String text_out = "";
36 void set_Command(){
37     number_out = number_in * number_sign;
38     number_in = 0;
39     number_sign = 1;
40     text_out = text_in;
41     text_in = "";
42 }
```

```
44 void execute() {
45     if (text_out == "x"){
46         x = long(number_out);
47         x_flag = 1;
48         Serial.print("x=");
49         Serial.println(x);
50     } else if (text_out == "y"){
51         y = long(number_out);
52         y_flag = 1;
53         Serial.print("y=");
54         Serial.println(y);
55     } else if (text_out == "z"){
56         z = long(number_out);
57         z_flag = 1;
58         Serial.print("z=");
59         Serial.println(z);
60     }
61 }
```

# Project 3 – Angles are fine, coordinates are better

```
14  if (y == 0L){
15      angle1 = 90;
16  } else if (x == 0L){
17      angle1 = -(sign(y)*90 - 90);
18  } else {
19      angle1 = acos((sq(xx)+sq(dxy)-sq(yy))/((double)(2*xx*dxy)))*4068/71;
20      angle1 = -(sign(y)*angle1 - 90);
21  }
22  Serial.print("angle1=");
23  Serial.println(angle1);
24  Servo1.write((int)angle1);
25  delay(1000);
```

```
6 void MoveServos() {
7     long xx = abs(x);
8     long yy = abs(y);
9     long zz = abs(z);
10    // Servo 1
11    long dxy = sqrt(sq(xx)+sq(yy));
12    Serial.print("dxy=");
13    Serial.println(dxy);
```

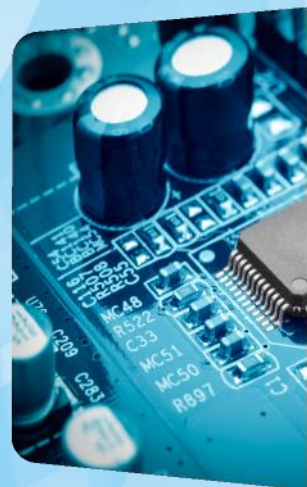
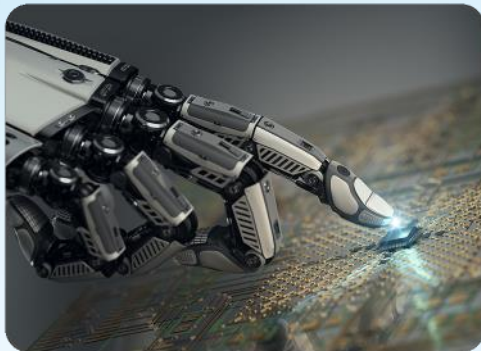
# Project 3 – Angles are fine, coordinates are better

```
27 // Servo 2,3
28 long dxyz = sqrt(sq(dxy)+sq(zz));
29 Serial.print("dxyz=");
30 Serial.println(dxyz);
31 double xyz_angle;
32 if (dxy == 0){
33     if (sign(z) < 0){
34         xyz_angle = -90;
35     } else if (sign(z) > 0){
36         xyz_angle = 90;
37     } else {
38         Serial.println("Zero coordinates.");
39         Serial.println("Arm not moving.");
40         return;
41     }
42 } else if (z == 0L){
43     xyz_angle = 0;
44 } else {
45     xyz_angle = acos((sq(dxy)+sq(dxyz)-sq(zz))
46                     /(((double)(2*dxy*dxyz))))*4068/71;
47     xyz_angle = xyz_angle*sign(z);
48 }
49 Serial.print("xyz_angle=");
50 Serial.println(xyz_angle);
```

```
52  if (dxyz > (length1+length2)){
53      Serial.println("Object is too far.");
54      Serial.println("Arm not moving.");
55      return;
56  } else {
57      angle2 = acos((sq(length1)+sq(dxyz)-sq(length2))
58          /((double)(2*length1*dxyz)))*4068/71;
59      Serial.print("triangle_angle2=");
60      Serial.println(angle2);
61      angle2 = 90 - xyz_angle - angle2 - 20; // - error
62      Serial.print("angle2=");
63      Serial.println(angle2);
64      angle3 = acos((sq(length1)+sq(length2)-sq(dxyz))
65          /(double)(2*length1*length2)))*4068/71;
66      Serial.print("triangle_angle3=");
67      Serial.println(angle3);
68      angle3 = - 90 + angle3 + (-10); // + error
69      Serial.print("angle3=");
70      Serial.println(angle3);
71  }
```

## Project 3 – Angles are fine, coordinates are better

```
72  if ((angle2<0) || (angle2>180)){
73      Serial.println("Servo2 can not go there.");
74      Serial.println("Arm not moving.");
75      return;
76  } else if ((angle3<0) || (angle3>180)){
77      Serial.println("Servo3 can not go there.");
78      Serial.println("Arm not moving.");
79      return;
80  } else {
81      Servo2.write((int)angle2);
82      delay(1000);
83      Servo3.write((int)angle3);
84      delay(1000);
85  }
86 }
```



# Arduino Workshop – Intermediate Level

## *Day 2 – The End*