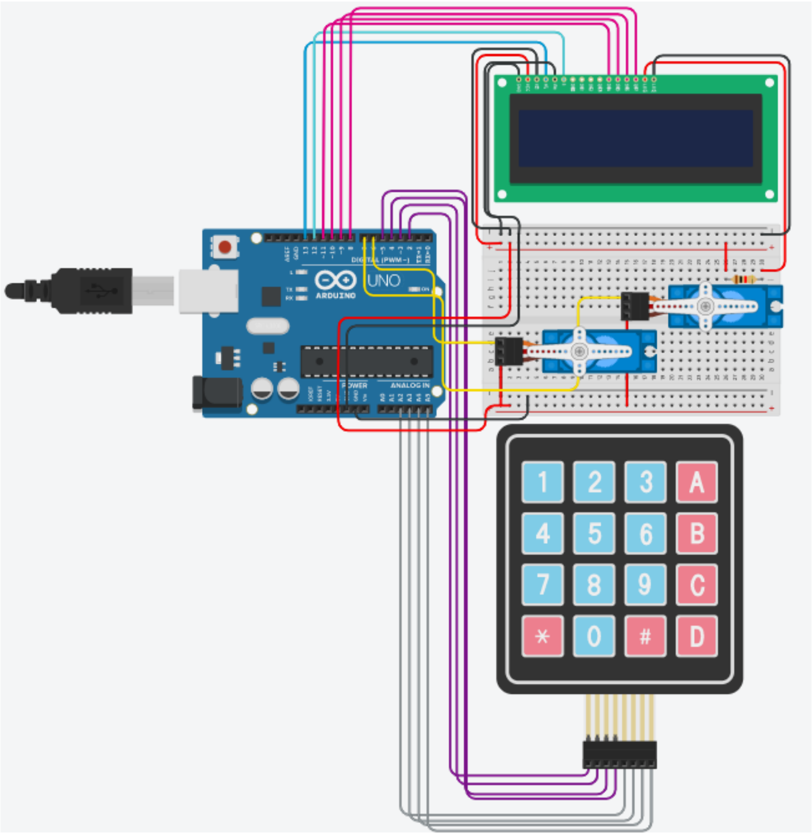
Robot Arm (2DoF)

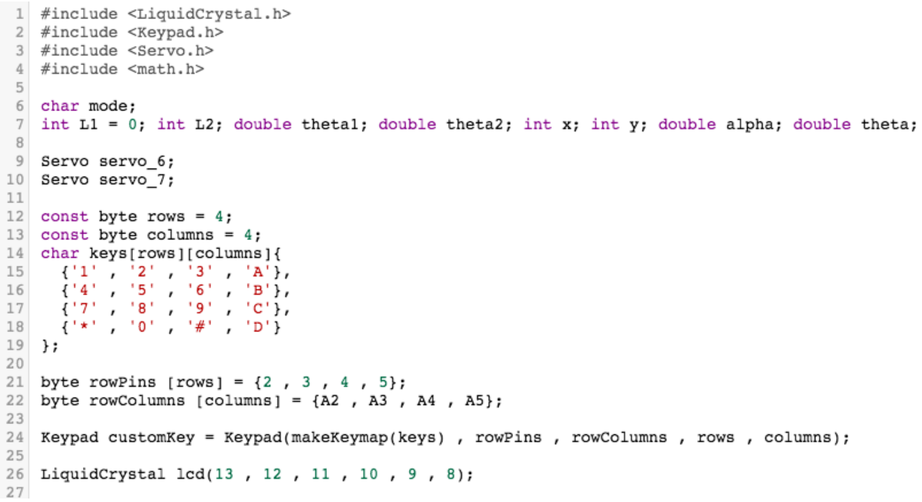
Circuit:



Code:

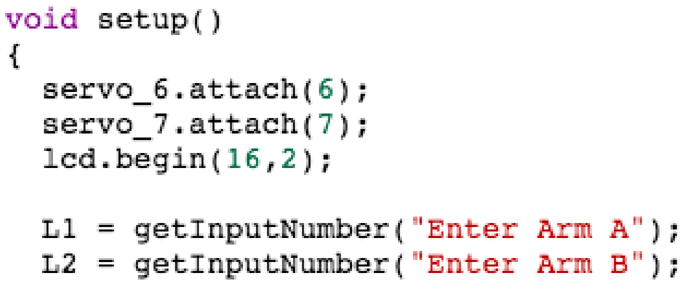
The steps taken to achieve this code were the following

1- Defining the variables needed and importing necessary libraries for the robot arm circuit.



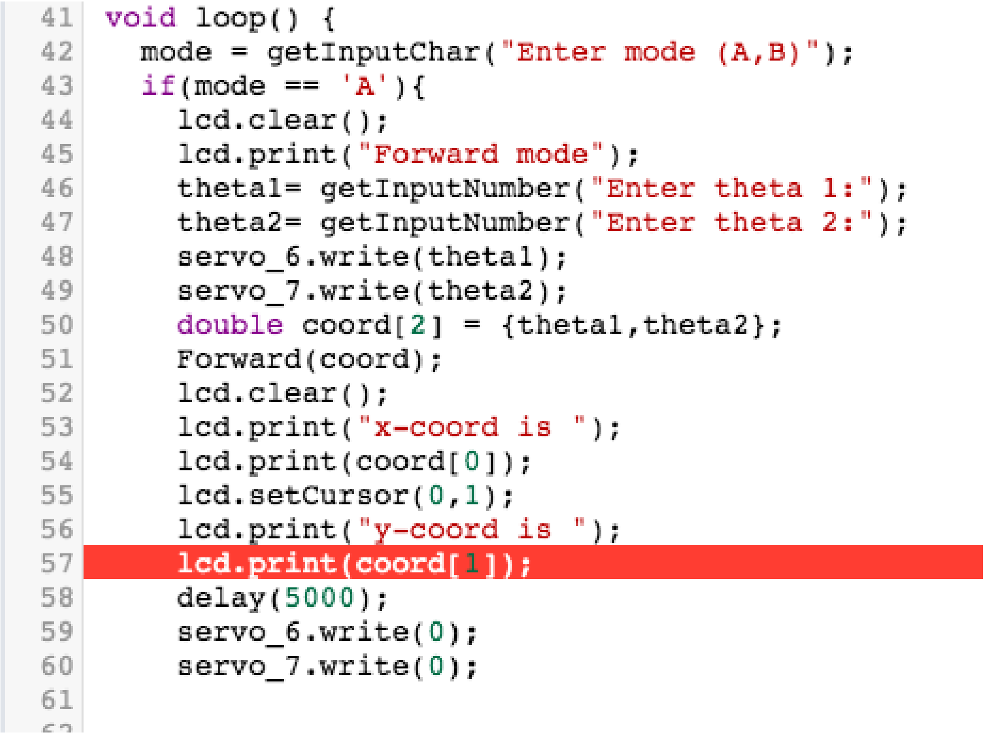
```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Servo.h>
#include <math.h>

char mode;
int L1 = 0; int L2; double theta1; double theta2; int x; int y; double alpha; double theta;

Servo servo_6;
Servo servo_7;

const byte rows = 4;
const byte columns = 4;
char keys[rows][columns]{
  {'1' , '2' , '3' , 'A'},
  {'4' , '5' , '6' , 'B'},
  {'7' , '8' , '9' , 'C'},
  {'*' , '0' , '#' , 'D'}
};

byte rowPins [rows] = {2 , 3 , 4 , 5};
byte rowColumns [columns] = {A2 , A3 , A4 , A5};

Keypad customKey = Keypad(makeKeymap(keys) , rowPins , rowColumns , rows , columns);

LiquidCrystal lcd(13 , 12 , 11 , 10 , 9 , 8);

```

2- In setup function, the first and second motor were linked to their corresponding digital pins and the length of Arm A and Arm B were given from the user by using the function getInputNumber().

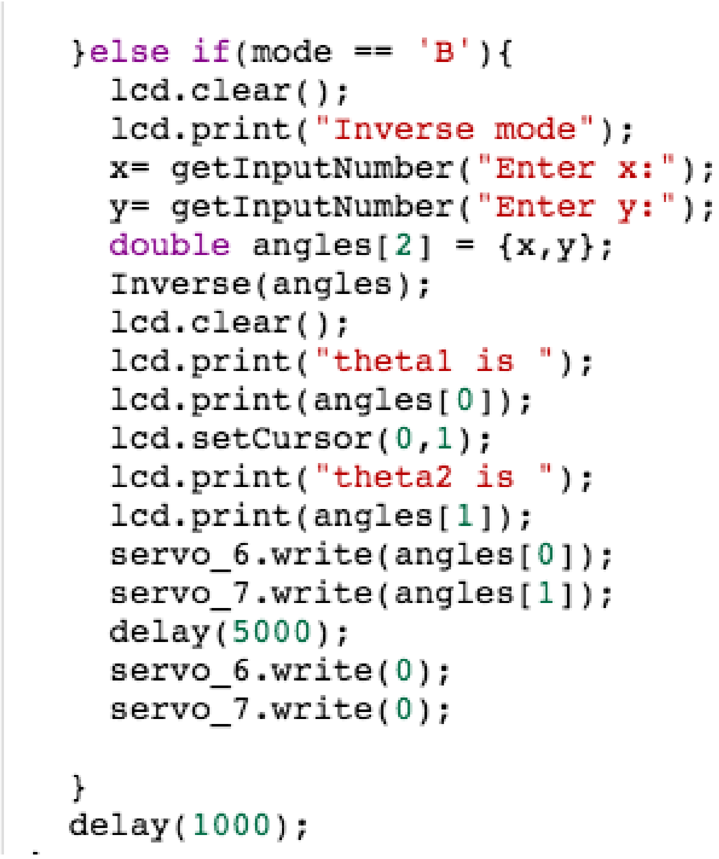


```
void setup()
{
  servo_6.attach(6);
  servo_7.attach(7);
  lcd.begin(16,2);

  L1 = getInputNumber("Enter Arm A");
  L2 = getInputNumber("Enter Arm B");
```

3- In loop function, the function getInputChar() was used to only allow the user to enter A for Forward mode or B for Inverse Mode. After that, if command was used to give values to theta1 and theta2 as well as adjust the motors position and use Forward() function and output the x-y coordinates if the user entered "A". If the user entered "B", then x-y coordinates will be given as input values and the Inverse() function will calculate theta1 and theta2 and the motors positions will be adjusted.



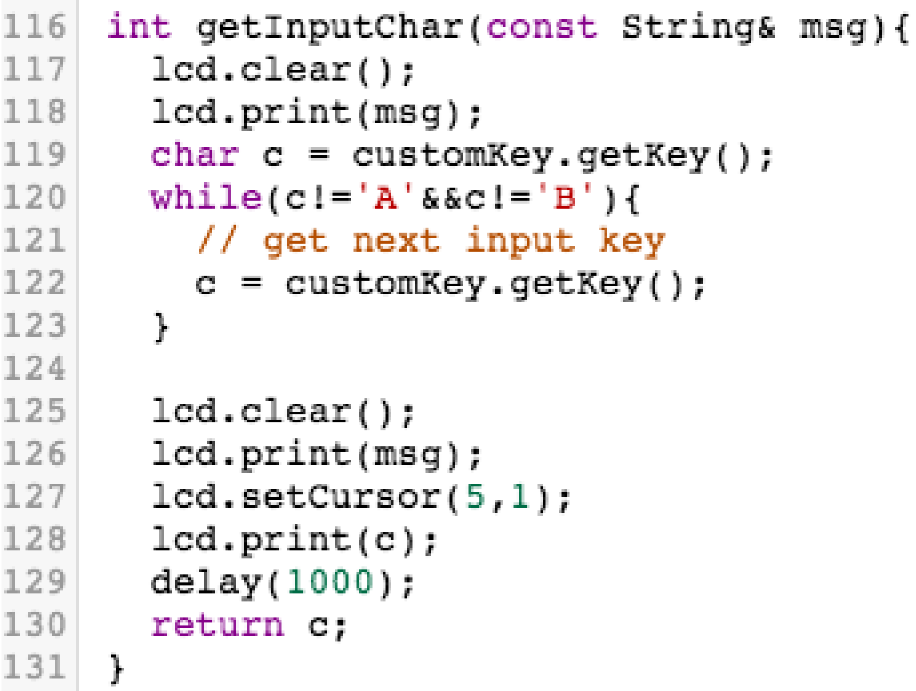
```
void loop() {
  mode = getInputChar("Enter mode (A,B)");
  if(mode == 'A'){
    lcd.clear();
    lcd.print("Forward mode");
    theta1= getInputNumber("Enter theta 1:");
    theta2= getInputNumber("Enter theta 2:");
    servo_6.write(theta1);
    servo_7.write(theta2);
    double coord[2] = {theta1,theta2};
    Forward(coord);
    lcd.clear();
    lcd.print("x-coord is ");
    lcd.print(coord[0]);
    lcd.setCursor(0,1);
    lcd.print("y-coord is ");
    lcd.print(coord[1]);
    delay(5000);
    servo_6.write(0);
    servo_7.write(0);
```



```
}else if(mode == 'B'){
  lcd.clear();
  lcd.print("Inverse mode");
  x= getInputNumber("Enter x:");
  y= getInputNumber("Enter y:");
  double angles[2] = {x,y};
  Inverse(angles);
  lcd.clear();
  lcd.print("theta1 is ");
  lcd.print(angles[0]);
  lcd.setCursor(0,1);
  lcd.print("theta2 is ");
  lcd.print(angles[1]);
  servo_6.write(angles[0]);
  servo_7.write(angles[1]);
  delay(5000);
  servo_6.write(0);
  servo_7.write(0);

}
delay(1000);
```

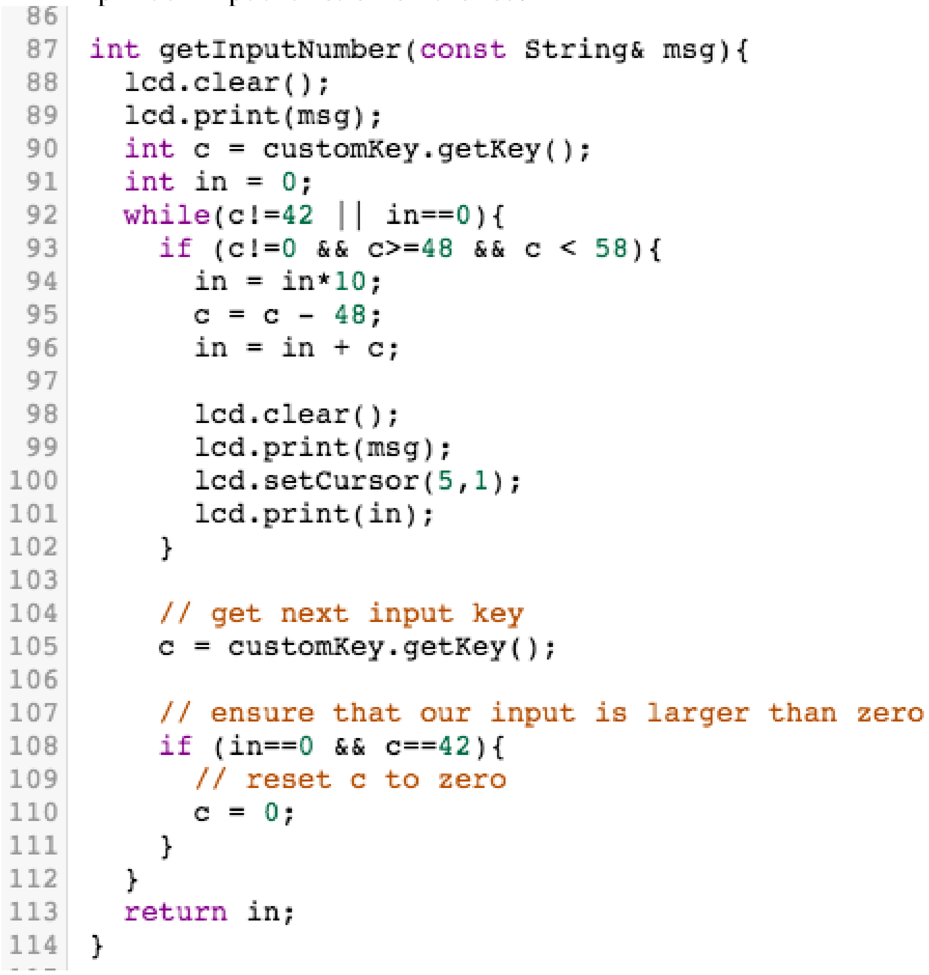
4- Creating getInputChar() and getInputNumber() functions:

The getInputChar() function was used to limit the used to either press “A” or “B” on the keypad as well as print an input string on the lcd.



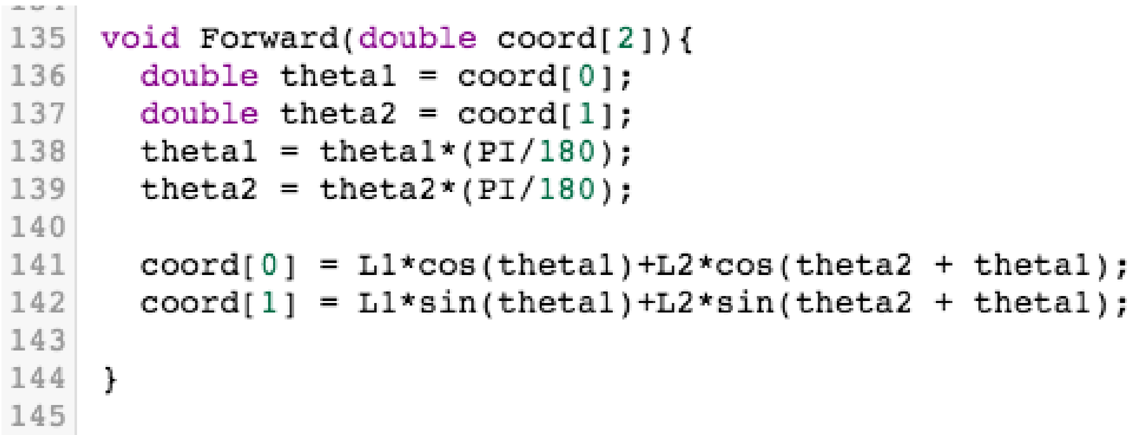
```
int getInputChar(const String& msg){
  lcd.clear();
  lcd.print(msg);
  char c = customKey.getKey();
  while(c!='A'&&c!='B'){
    // get next input key
    c = customKey.getKey();
  }

  lcd.clear();
  lcd.print(msg);
  lcd.setCursor(5,1);
  lcd.print(c);
  delay(1000);
  return c;
}
```

The getInputNumber() function was created to allow the user to enter any number using the keypad and input the final number by pressing “*” on the keypad as well as print an input function on the lcd.



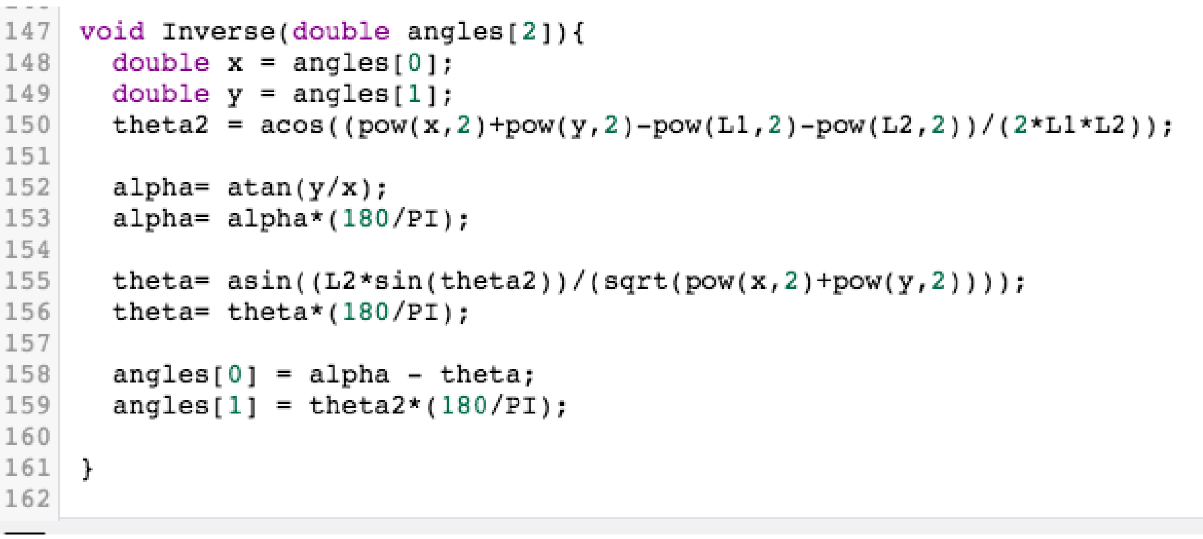
```
int getInputNumber(const String& msg){
  lcd.clear();
  lcd.print(msg);
  int c = customKey.getKey();
  int in = 0;
  while(c!=42 || in==0){
    if (c!=0 && c>=48 && c < 58){
      in = in*10;
      c = c - 48;
      in = in + c;

      lcd.clear();
      lcd.print(msg);
      lcd.setCursor(5,1);
      lcd.print(in);
    }

    // get next input key
    c = customKey.getKey();

    // ensure that our input is larger than zero
    if (in==0 && c==42){
      // reset c to zero
      c = 0;
    }
  }
  return in;
}
```

5- Creating Forward() and Inverse() function:
The forward() function was created to output the x-y coordinates from theta1 and theta2 inputs. This functions takes an array of two elements and changes the elements’ values theta1 and theta2 to x and y, respectively.



```
void Forward(double coord[2]){
  double theta1 = coord[0];
  double theta2 = coord[1];
  theta1 = theta1*(PI/180);
  theta2 = theta2*(PI/180);

  coord[0] = L1*cos(theta1)+L2*cos(theta2 + theta1);
  coord[1] = L1*sin(theta1)+L2*sin(theta2 + theta1);

}
```

The Inverse() function was created to output theta1 and theta2 for x and y inputs. This function takes an array of two elements and changes the elements’ values x and y to theta1 and theta2, respectively.



```
void Inverse(double angles[2]){
  double x = angles[0];
  double y = angles[1];
  theta2 = acos((pow(x,2)+pow(y,2)-pow(L1,2)-pow(L2,2))/(2*L1*L2));

  alpha= atan(y/x);
  alpha= alpha*(180/PI);

  theta= asin((L2*sin(theta2))/(sqrt(pow(x,2)+pow(y,2))));
  theta= theta*(180/PI);

  angles[0] = alpha - theta;
  angles[1] = theta2*(180/PI);

}
```