```
#include <Arduino.h>

/*********************************************************************

    Name    : LCD Button Shield Menu
    Author  : Jonathan Jenkins
    Created : June 14, 2016
    Last Modified: June 14, 2016
    Version : 1.0

Key Information:
  - Revision 1.0
  - Board : Arduino nano

 Notes   : This code is for use with an Arduino Uno and LCD/button shield. The
              intent is for anyone to use this program to give them a starting
              program with a fully functional menu with minimal modifications
              required by the user.
*********************************************************************/


 // ***************** INCLUDE NECESSARY LIBRARIES *****************

#include <SPI.h>
#include <Wire.h>
#include <Servo.h>




// ***************** GENERAL VARIABLES *****************

// INITIATE CONSTANT INTEGERS
// constants won't change. They're used here to set pin numbers:
  const int fsrPin = A2; //  the FSR and 10K pulldown
  const int HALLpin = A1; // input pullup
  const int Temppin = A3; // input pullup
  const int pixel = 4 ;



 // INITIATE VARIABLES (GLOBAL)
 // variables will change:
  int SerialSensorMore = true; // PUSH BUTTON 1 --- ARM
  int forceMeasured = 0; // PUSH BUTTON 1 --- ARM
```

```
  int tempmeasured = 0; // PUSH BUTTON 1 --- ARM
  int force2measured= 0; // PUSH BUTTON 1 --- ARM
  int force1measured = 0; // PUSH BUTTON 1 --- ARM
   int pixelLT = 0;
    int pixelRT = 0 ;



// ****************SERVO *****************

Servo myservo1;  // create servo object to control a servo
Servo myservo2;  // create servo object to control a servo
Servo myservo3;  // create servo object to control a servo
Servo myservo4;  // create servo object to control a servo

// INITIATE CONSTANT INTEGERS
// constants won't change. They're used here to set pin numbers:
  const int servo1 = 13; //  the FSR and 10K pulldown
  const int servo2 = 12; // input pullup
  const int servo3 = 11; // input pullup
  const int servo4 = 3;

 // INITIATE VARIABLES
 // variables will change:
int pos = 0;    // variable to store the servo position
int servooldState=0;
int servonewState=0;
int servotask=0;



// ****************Menu *****************


// You can have up to 10 menu items in the menuItems[] array below without having
to change the base programming at all. Name them however you'd like. Beyond 10
items, you will have to add additional "cases" in the switch/case
// section of the operateMainMenu() function below. You will also have to add
additional void functions (i.e. menuItem11, menuItem12, etc.) to the program.
String menuItems[] = {"ITEM 1", "ITEM 2", "finger tip", "Med Wide", "Large Wide",
"L GRIP", "T GRIP", "L DIAMETER", "S DIAMETER", "PINCH"};
String submenuItems[] = {"Sub Menu 1", "Sub Menu 2", "Sub Menu 3", "Sub Menu 4",
"Sub Menu 5", "Sub Menu 6", "Sub Menu 7", "Sub Menu 8", "Sub Menu 9", "Sub Menu
10"};
```

```
// Navigation button variables
int readKey;

// Menu control variables
int menuPage = 0;
int maxMenuPages = 8 ; // amount f menu items -2
int cursorPosition = 0;

// Creates 3 custom characters for the menu display
byte downArrow[8] = {
  0b00100, //   *
  0b00100, //   *
  0b00100, //   *
  0b00100, //   *
  0b00100, //   *
  0b10101, // * * *
  0b01110, //  ***
  0b00100  //   *
};

byte upArrow[8] = {
  0b00100, //   *
  0b01110, //  ***
  0b10101, // * * *
  0b00100, //   *
  0b00100, //   *
  0b00100, //   *
  0b00100, //   *
  0b00100  //   *
};

byte menuCursor[8] = {
  B01000, //  *
  B00100, //   *
  B00010, //    *
  B00001, //     *
  B00010, //    *
  B00100, //   *
  B01000, //  *
  B00000  //
};

#include <Wire.h>
#include <LiquidCrystal.h>
```

```
// Setting the LCD shields pins
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
  int activepage=0;
    int activeButton;
    int button=0;
  int buttonlimit;
 int lcdbutton=1000;









// ****************SETUP *****************
void setup() {
  Serial.begin(9600);
  Serial.println();
  Serial.println("Starting...");
  menusetup();
  servosetup();
  Serial.println("READY");
  delay(100);
}
// ****************End SETUP ***********************




// ****************  Main Loop *****************

void loop() {
  menu();   // function that controls changing arm state
  servo();
  delay(100);
}

// **************** End Main Loop *****************
```

```
/*********************************************************************************
******
 *
    Notes   : This code is for use with an Arduino Uno and LCD/button shield. The
              intent is for anyone to use this program to give them a starting
              program with a fully functional menu with minimal modifications
              required by the user.


*********************************************************************************
*****/









// ****************   SETUP *****************
//Call in main setup void
void menusetup() {
  // Initializes and clears the LCD screen
  lcd.begin(16, 2);
  lcd.clear();

  // Creates the byte for the 3 custom characters
  lcd.createChar(0, menuCursor);
  lcd.createChar(1, upArrow);
  lcd.createChar(2, downArrow);
}
// ****************  SETUP END *****************







// ****************  Main Loop *****************

void menu() {

   input();     // function that obtains input value from keypad
```

```
/*********************************************************************************
******
 *
    Notes   : This code controls the animation of the servo.
              Servo 1-4 connected to PWM Pins


*********************************************************************************
*****/


// ****************   SETUP *****************
//Call in main setup void
void servosetup() {
  myservo1.attach(servo1);  // attaches the servo on pin 9 to the servo object
  myservo2.attach(servo2);  // attaches the servo on pin 9 to the servo object
  myservo3.attach(servo3);  // attaches the servo on pin 9 to the servo object
  myservo4.attach(servo4);  // attaches the servo on pin 9 to the servo object
  opengrip();


}
// ****************  SETUP END *****************



// ****************  Main Void *****************

void servo() {
   servonewState = servotask;      // Get current button state.

if (servooldState != servonewState) {    // Check if state changed from high to
low (button press).

    switch (servotask) {                    // The case that is selected here is
dependent on which menu page you are on and where the cursor is.
          case 0:
            opengrip();
            break;
          case 1:
            grip1();
            break;
          case 2:
            grip2();
```

```
            break;
          case 3:
            grip3();
            break;

        }
        // Set the last-read button state to the old state.
  servooldState = servonewState;
     }

}




  // ****************  Sub Void ******************

  void opengrip(){


     myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);                       // waits 15 ms for the servo to reach the
position

      myservo1.write(50);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(70);              // Servo 2 ( Base Close)
      delay(500);                       // waits 15 ms for the servo to reach the
position

      myservo3.write(100);              // Servo 3 ( Fingertip Open)
      myservo4.write(100);              // Servo 4 ( Base Open)


  }

  void grip1(){               // water bottle
     myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
```

```
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);                       // waits 15 ms for the servo to reach the
position


      myservo1.write(25);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(60);              // Servo 2 ( Base Close)

     delay(500);                       // waits 15 ms for the servo to reach the
position

      myservo3.write(50);              // Servo 3 ( Fingertip Open)
      myservo4.write(80);              // Servo 4 ( Base Open)
   }

  void grip2(){

      myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);



      myservo1.write(60);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(60);              // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(60);              // Servo 3 ( Fingertip Open)
      myservo4.write(60);              // Servo 4 ( Base Open)
  }

  void grip3(){
     myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)
```

```
      myservo3.write(70);             // Servo 3 ( Fingertip Open)
      myservo4.write(70);             // Servo 4 ( Base Open)

     delay(500);

      myservo1.write(10);             // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(30);             // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(70);             // Servo 3 ( Fingertip Open)
      myservo4.write(90);             // Servo 4 ( Base Open)
  }


  void grip4(){
       myservo1.write(80);             // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);             // Servo 2 ( Base Close)

      myservo3.write(70);             // Servo 3 ( Fingertip Open)
      myservo4.write(70);             // Servo 4 ( Base Open)

     delay(500);
       myservo1.write(20);             // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(45);             // Servo 2 ( Base Close)
      delay(500);                      // waits 15 ms for the servo to reach the
position

      myservo3.write(75);             // Servo 3 ( Fingertip Open)
      myservo4.write(95);             // Servo 4 ( Base Open)

  }


  void grip5(){
       myservo1.write(80);             // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);             // Servo 2 ( Base Close)

      myservo3.write(70);             // Servo 3 ( Fingertip Open)
      myservo4.write(70);             // Servo 4 ( Base Open)

     delay(500);
```

```
       myservo1.write(20);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(55);              // Servo 2 ( Base Close)
      delay(500);                       // waits 15 ms for the servo to reach the
position

      myservo3.write(85);              // Servo 3 ( Fingertip Open)
      myservo4.write(95);              // Servo 4 ( Base Open)
  }



  void grip6(){
       myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);
      myservo1.write(15);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(45);              // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(90);              // Servo 4 ( Base Open)
  }



  void grip7(){
       myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);
 myservo1.write(2);               // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(30);              // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(70);              // Servo 3 ( Fingertip Open)
```

```
      myservo4.write(90);              // Servo 4 ( Base Open)
  }




  void grip8(){
       myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);
       myservo1.write(20);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(40);              // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)
  }




  void grip9(){
       myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);
myservo1.write(25);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(15);              // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(90);              // Servo 3 ( Fingertip Open)
      myservo4.write(80);              // Servo 4 ( Base Open)
  }
```

```
  void grip10(){
       myservo1.write(80);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(80);              // Servo 2 ( Base Close)

      myservo3.write(70);              // Servo 3 ( Fingertip Open)
      myservo4.write(70);              // Servo 4 ( Base Open)

     delay(500);

myservo1.write(30);              // Servo 1 ( Fingertip Close) .   Closen
<----90-----> Open
      myservo2.write(15);              // Servo 2 ( Base Close)
      delay(500);
      myservo3.write(90);              // Servo 3 ( Fingertip Open)
      myservo4.write(80);              // Servo 4 ( Base Open)
  }
```

```
   switch (activepage) { // The case that is selected here is dependent on which
menu page you are on and where the cursor is.
          case 0:
            mainMenuDraw();
            drawCursor();
            operateMainMenu();
            break;
          case 1:
            menuItem1();
            break;
          case 2:
            menuItem2();
            break;
          case 3:
            menuItem3();
            break;
          case 4:
            menuItem4();
            break;
          case 5:
            menuItem5();
            break;
          case 6:
            menuItem6();
            break;
          case 7:
            menuItem7();
            break;
          case 8:
            menuItem8();
            break;
          case 9:
            menuItem9();
            break;
          case 10:
            menuItem10();
            break;
    }

    delay(10);
}
// ****************  Main Loop *****************
```

```
// ****************  Functions *****************



void input(){


readKey = analogRead(0); // UPDATE CURRENT STATE OF BUTTON 1
  if(readKey <= 800 && buttonlimit == LOW){                  // PRESS INITIATED
     lcdbutton = readKey;                                    // Store the value
of the input detected
     buttonlimit = HIGH;                                     // logic bit that
identify that the button has been recorded and to limit repeated commands
    }

 if (readKey >= 800 && buttonlimit == HIGH ){
     buttonlimit = LOW;                                      // reset limit bit
      if (lcdbutton < 50) {                                  // determain desired
command pressed
        button = 1; // right
      } else if (lcdbutton < 200) {
        button = 2; // up
      } else if (lcdbutton < 400) {
        button = 3; // down
      } else if (lcdbutton < 600) {
        button = 4; // left
      }else if (lcdbutton < 800) {
        button = 5; // select
      }
       activeButton = 1;                          // button value has been
updated and will be used to execude command
       delay(10);
      }
}
```

```
// ***************  VOIDS *****************


// This function will generate the 2 menu items that can fit on the screen. They
will change as you scroll through your menu. Up and down arrows will indicate
your current menu position.
void mainMenuDraw() {
  //Serial.print(menuPage);
  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print(menuItems[menuPage]);
  lcd.setCursor(1, 1);
  lcd.print(menuItems[menuPage + 1]);
  if (menuPage == 0) {
    lcd.setCursor(15, 1);
    lcd.write(byte(2));
  } else if (menuPage > 0 and menuPage < maxMenuPages) {
    lcd.setCursor(15, 1);
    lcd.write(byte(2));
    lcd.setCursor(15, 0);
    lcd.write(byte(1));
  } else if (menuPage == maxMenuPages) {
    lcd.setCursor(15, 0);
    lcd.write(byte(1));
  }
}



// When called, this function will erase the current cursor and redraw it based
on the cursorPosition and menuPage variables.
void drawCursor() {
  for (int x = 0; x < 2; x++) {     // Erases current cursor
    lcd.setCursor(0, x);
    lcd.print(" ");
  }

  // The menu is set up to be progressive (menuPage 0 = Item 1 & Item 2, menuPage
1 = Item 2 & Item 3, menuPage 2 = Item 3 & Item 4), so
  // in order to determine where the cursor should be you need to see if you are
at an odd or even menu page and an odd or even cursor position.
  if (menuPage % 2 == 0) {
    if (cursorPosition % 2 == 0) {  // If the menu page is even and the cursor
position is even that means the cursor should be on line 1
```

```
      lcd.setCursor(0, 0);
      lcd.write(byte(0));
    }
    if (cursorPosition % 2 != 0) {  // If the menu page is even and the cursor
position is odd that means the cursor should be on line 2
      lcd.setCursor(0, 1);
      lcd.write(byte(0));
    }
  }
  if (menuPage % 2 != 0) {
    if (cursorPosition % 2 == 0) {  // If the menu page is odd and the cursor
position is even that means the cursor should be on line 2
      lcd.setCursor(0, 1);
      lcd.write(byte(0));
    }
    if (cursorPosition % 2 != 0) {  // If the menu page is odd and the cursor
position is odd that means the cursor should be on line 1
      lcd.setCursor(0, 0);
      lcd.write(byte(0));
    }
  }
}




void operateMainMenu() {

    if (activeButton == 1){
        activeButton = 0;

switch (button) {

      case 0: // When button returns as 0 there is no action taken
        activeButton = 0;
        button = 0;
        break;

      case 5:  // This case will execute if the "select" button is pressed
        activeButton = 0;
        button = 0;

        switch (cursorPosition) { // The case that is selected here is dependent
on which menu page you are on and where the cursor is.
          case 0:
```

```
      activepage=1;
      Serial.println("submenu 1");

      break;
    case 1:
      activepage=2;
      Serial.println("submenu 2");
      break;
    case 2:
      activepage=3;
      Serial.println("submenu 3");
      break;
    case 3:
      activepage=4;
      Serial.println("submenu 4");
      break;
    case 4:
      activepage=5;
      Serial.println("submenu 5");
      break;
    case 5:
      activepage=6;
      Serial.println("submenu 6");
      break;
    case 6:
      activepage=7;
      Serial.println("submenu 7");
      break;
    case 7:
      activepage=8;
      Serial.println("submenu 8");
      break;
    case 8:
      activepage=9;
      Serial.println("submenu 9");
      break;
    case 9:
      activepage=10;
      Serial.println("submenu 10");
      break;
  }

  break;

case 2:    // This case will execute if the "up" button is pressed
```

```
        activeButton = 0;
        button = 0;
        if (menuPage == 0) {
          cursorPosition = cursorPosition - 1;
          cursorPosition = constrain(cursorPosition, 0, ((sizeof(menuItems) /
sizeof(String)) - 1));
        }
        if (menuPage % 2 == 0 and cursorPosition % 2 == 0) {
          menuPage = menuPage - 1;
          menuPage = constrain(menuPage, 0, maxMenuPages);
        }

        if (menuPage % 2 != 0 and cursorPosition % 2 != 0) {
          menuPage = menuPage - 1;
          menuPage = constrain(menuPage, 0, maxMenuPages);
        }

        cursorPosition = cursorPosition - 1;
        cursorPosition = constrain(cursorPosition, 0, ((sizeof(menuItems) /
sizeof(String)) - 1));

        mainMenuDraw();
        drawCursor();
        break;


      case 3: // This case will execute if the "down" button is pressed
        activeButton = 0;
        button = 0;
        if (menuPage % 2 == 0 and cursorPosition % 2 != 0) {
          menuPage = menuPage + 1;
          menuPage = constrain(menuPage, 0, maxMenuPages);
        }

        if (menuPage % 2 != 0 and cursorPosition % 2 == 0) {
          menuPage = menuPage + 1;
          menuPage = constrain(menuPage, 0, maxMenuPages);
        }

        cursorPosition = cursorPosition + 1;
        cursorPosition = constrain(cursorPosition, 0, ((sizeof(menuItems) /
sizeof(String)) - 1));
        mainMenuDraw();
        drawCursor();
        break;
```

```
    }
  }

}









// ****************  VOIDS *****************



void menuItem1() { // Function executes when you select the 1st item from main
menu

     lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Sub Menu 1");

  if (activeButton == 1){
       activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
         grip1();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
```

```
        activepage=0;
        break;
    }


  }
  }
```

```
void menuItem2() { // Function executes when you select the 2nd item from main
menu
  lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Sub Menu 2");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip2();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }


  }
  }
```

```
void menuItem3() { // Function executes when you select the 3rd item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Finger tip");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip3();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }


  }
}

void menuItem4() { // Function executes when you select the 4th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Med Wide");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {
```

```
      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip4();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }


  }
}

void menuItem5() { // Function executes when you select the 5th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Large Wide");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip5();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
```

```
       Serial.println("back");
        activepage=0;
        break;
    }


  }
}

void menuItem6() { // Function executes when you select the 6th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("loose grip");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip6();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }


  }
}

void menuItem7() { // Function executes when you select the 7th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
```

```
    lcd.print("Tight Grip");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip7();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }


  }
}

void menuItem8() { // Function executes when you select the 8th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Large Diameter");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip8();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
```

```
        Serial.println("-");
        opengrip();
       delay(15);
       break;

     case 5:  // This case will execute if the "back" button is pressed
      Serial.println("back");
       activepage=0;
       break;
   }


  }
}



void menuItem9() { // Function executes when you select the 9th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Small diameter");

  if (activeButton == 1){
       activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip9();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }
```

```
  }
}
```

```
void menuItem10() { // Function executes when you select the 10th item from main
menu
lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("FLAT PINCH");

  if (activeButton == 1){
        activeButton = 0;

    switch (button) {

      case 1:  // This case will execute if the "right" button is pressed
        Serial.println("+");
        grip10();
        delay(15);
        break;
      case 4:  // This case will execute if the "left" button is pressed
         Serial.println("-");
         opengrip();
        delay(15);
        break;

      case 5:  // This case will execute if the "back" button is pressed
       Serial.println("back");
        activepage=0;
        break;
    }


  }
  }
```