Pinch Dynamometer Endurance Gaming Device

The device is a pinch battle system. It measures the pinch strength of two opponents in a two point or lateral pinch battle and tells the winner based on the person with the hardest pinch at the point a third party, the referee, decides to stop the game. The setup of the game consists of two force resistive sensors (FSR) that each opponent can pinch and a buzzer that the referee presses to either end or begin each round of the game. The game consists of three rounds. After each round, the winner is indicated by an asterisk underneath his/her handle. At the end of the game, the player that wins two of three rounds is shown as the winner and the game starts again. If there is no winner, the game indicates tie and starts again.

The main sensor used in this game is FSR, a sensor that interprets the force exerted on it as a resistance. A FSR is made of 2 layers (Semiconductor and Electrodes) separated by a spacer. The design of FSRs enables their resistance to decrease as the force exerted on the layers increase, allowing the layers to contact each other. This project measures the varying resistance from the FSR by connecting one end of the FSR to Power and the other to a pull-down resistor fixed to ground. Thus, the point between the pulldown resistor and the variable FSR resistor is connected to an analog input pin of the MSP430. To visualize the effect of the FSR, its reading could be used to modulate the brightness of an LED. In this device, the LED brightness was an extra indicator to the referee that each player was pinching the FSR with reasonable force.

A **B** **C**

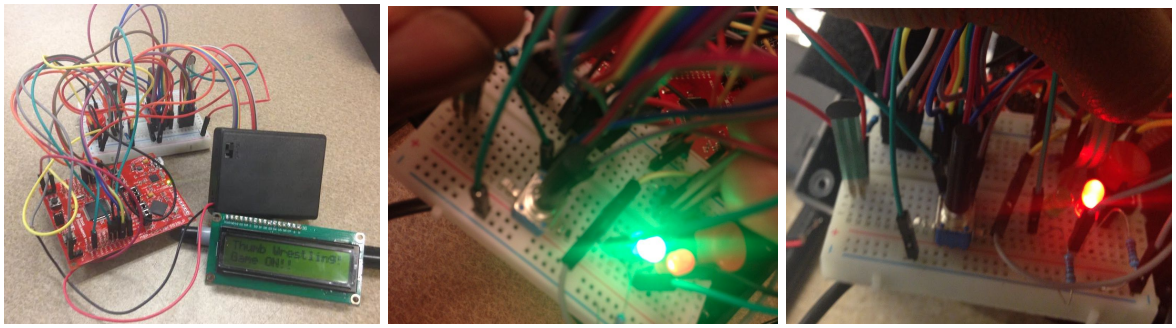


Figure 1. A shows the setup of the device (lcd, MSP-430, Breadboard, and battery box). B and C show the leds that light up during each round in response to the player's pinch during an active round. Player 1 lights the red LED while player 2 lights up green LED. The intensity of the LED's brightness corresponds to how hard the player pinches.

To operate, the referee switches on the device and the screen is preloaded with the text "Thumb Wrestling! Game on!!" The two opponents pick their alias as either player one (red) or player two (green).

Round 1

The referee starts the game by pressing a button to initiate the game. This is indicated by a change in screen from figure 2A to 2B. This signifies the beginning of round one. During this time, the players' LEDs brighten as they pinch harder (analog output of PWM pin). Also, an array of FSR readings are collected and averaged continuously during this time. The players

pinch hard until the referee presses the button to end the round. Once the button is pressed, the current average scores for both players goes through a set of conditionals to help determine the winner and assign points that would be accumulated at the end of the game. Also, the screen changes for about a second to indicate termination of the first round, then it changes to show the winner of the round (figures 2C, 2D).

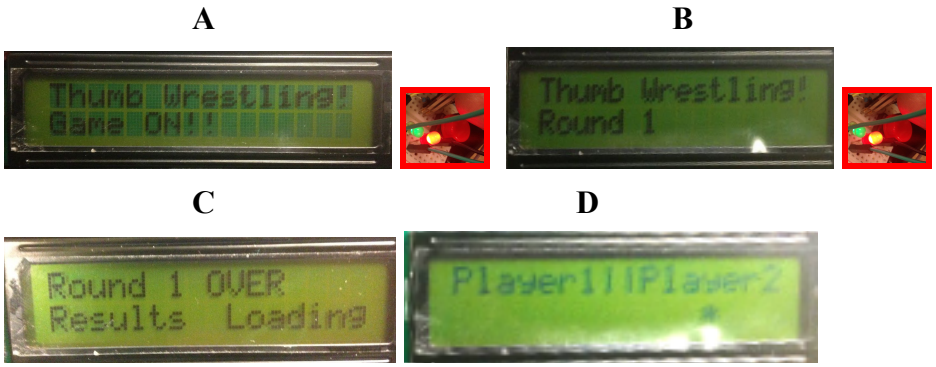


*Figure 2. Figure showing the screen changing process during round 1. The RED box represents button presses. The screen change order is from A to D. The winner is indicated by * under the player's handle.*

Round 2

The referee begins the second round with a button press and the screen changes, indicating the start of the second round. During this time, the players' LEDs brighten as they pinch harder. Also FSR readings are collected and averaged. Again, the players pinch hard until the referee presses the button to end the round (Figure 3B). Once the button is pressed, the current average scores for both players goes through a set of conditionals to help determine the winner of the round and store points that would be accumulated at the end of the game. Afterwhich, the screen changes for about a second to indicate termination of the second round, then it changes to show the winner of the round, indicated by asterisks beneath the players handle.

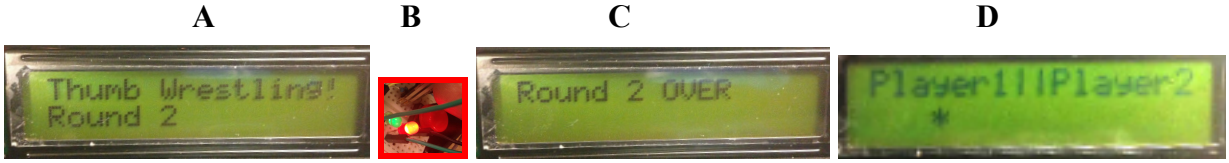


*Figure 3. Figure showing the screen changing process during round 1. The RED box represents button presses. The screen change order is from A to D. The winner is indicated by * under the player's handle.*

Round 3

The referee begins the third and last round with a button press and the screen changes indicating start of the second round. As usual, the players' LEDs brighten as they pinch harder. Also FSR readings are collected and averaged. Again, the players pinch hard until the referee presses the button to end the round (Figure 4B). Once the button is pressed, the screen changes for about a second to indicate termination of the first round, then it holds off the results of both players for

another second in suspense (Figure 4D). The game finally displays the winner for 3 seconds and restarts.

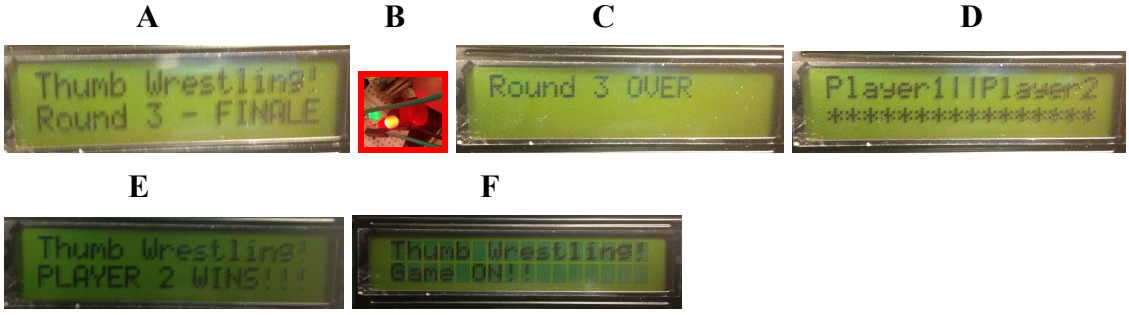
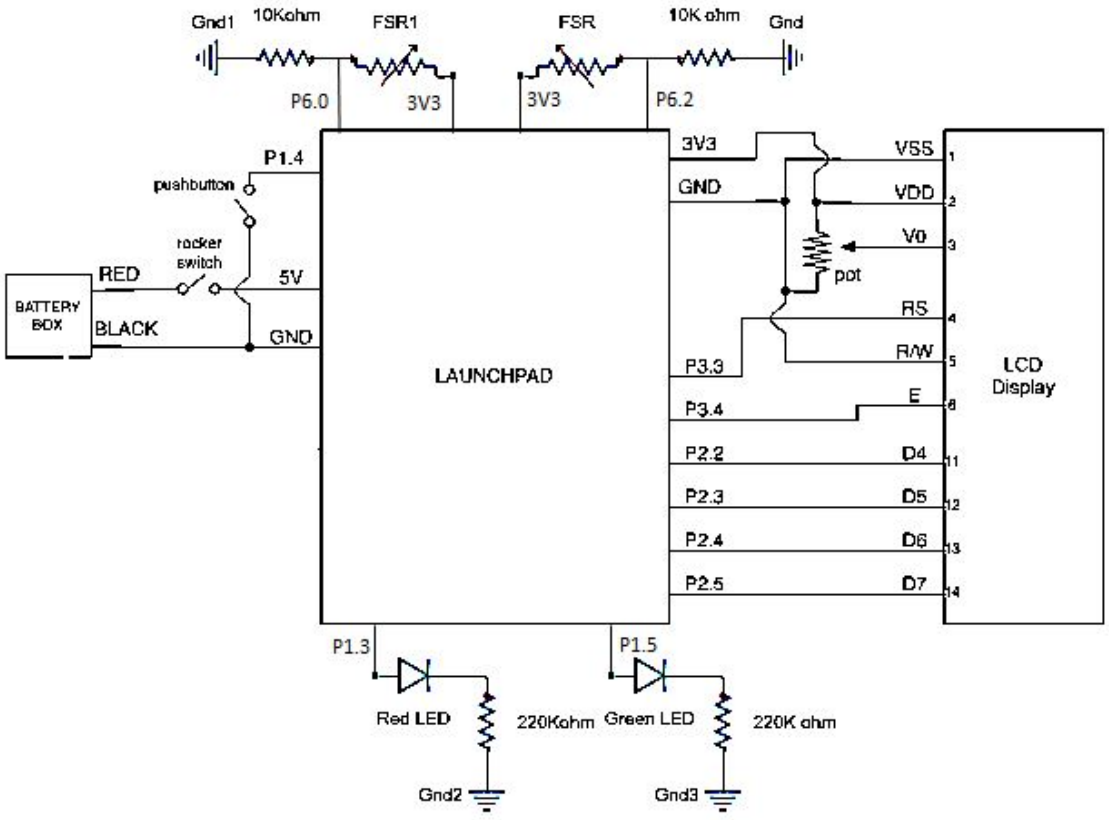


*Figure 4. Figure showing the screen changing process during round 1. The RED box represents button presses. The screen change order is from A to F. The winner is indicated by * under the player's handle.*

Circuit



```
#define _ACTIVATESERIAL
#define _USEINTERRUPT

#define PIN_BUTTON P1_4     // the number of the pushbutton pin

int FSR_Pin = P6_0;   // analog pin for 1st fsr
int LED = P1_3;       // Red LED that changes brightness player1 (PWM pin)
int LEDbrightness;    // storage for red LED brightness map ie, duty cycle
int FSR_Pinn = P6_2;  // analog pin for 2nd fsr
int LEDD = P1_5;      // Green LED change brightness player2 (PWM pin)
int LEDbrightnesss;   // storage for green LED brightness map ie, duty cycle
int array1[1000];     // store fsr values from player1
int array2[1000];     // store fsr values from player2
int Play1;            // average values from array1
int Play2;            // average values from array2
int winarray1[3];     // array that stores the points when player1 wins a round
int winarray2[3];     // array that stores the points when player2 wins a round
#include <LiquidCrystal.h> // include lcd library

#define DEBOUNCE 300 //debounce

LiquidCrystal lcd(P3_3, P3_4, P2_2, P2_3, P2_4, P2_5); //define lcd

long int now ;
long int lastpress ;

int presscount ;
volatile int update_count ;


void setup() {
  // put your setup code here, to run once:
    lcd.begin(16, 2);

  #ifdef _ACTIVATESERIAL
     Serial.begin(9600);
  #endif

    pinMode(PIN_BUTTON, INPUT_PULLUP) ;
    pinMode(LED, OUTPUT);

    lastpress=0;
    presscount=0;
    update_count=0;
```

```
  #ifdef _USEINTERRUPT
    attachInterrupt(PIN_BUTTON, inc_button_count, FALLING) ;
  #endif

  for (int i =0; i < 1000; i++)
    {
     array1[i] = 0; // array for collecting player one's FSR
     array2[i] = 0; // array for collecting player two's FSR
    }

    Play1 = 0; //average FSR initial value
    Play2 = 0; // average FSR initial value

   for (int i =0; i < 3; i++)
    {
     winarray1[i] = 0; //array for counting points accumulated each round by
player 1
    }

    for (int i =0; i < 3; i++)
    {
     winarray2[i] = 0; // array for counting points accumulated each round by
player 2
    }
}

void loop() {
  // reset winarrays to 0 before going through the game
   for (int i =0; i < 3; i++)
    {
     winarray1[i] = 0; //array for counting points accumulated each round by
player 1
    }

    for (int i =0; i < 3; i++)
    {
     winarray2[i] = 0; // array for counting points accumulated each round by
player 2
    }
  //Initially show this before button is pressed... Indicating start of game
    if (presscount == 0)
        {
        lcd.setCursor(0, 0);
        lcd.print("Thumb Wrestling!");
        lcd.setCursor(0,1);
```

```
        lcd.print("Game ON!!       ");
        }
     now = millis() ;

  //Button stuff
    #ifndef _USEINTERRUPT
      if (digitalRead(PIN_BUTTON) == LOW) update_count=1 ;
    #endif
    if (update_count>0)
    {
       update_count=0 ;

       if ((now-lastpress)>DEBOUNCE)
       {
         presscount++ ; // count the number button presses
         lcd.clear();
         lastpress=now ;
       }
    }

  ////////////Round 1\\\\\\\\\\\\\\\
  // printing Game stuff
   if (presscount == 1) //Upon one buttonpress, print round 1
    {
        lcd.setCursor(0,0);
        lcd.print("Thumb Wrestling!");
        lcd.setCursor(0,1);
        lcd.print("Round 1         ");
        Time = millis()/1000;

    //obtain fsr for game round 1

     //array1
        for (int i=0; i<1000; i++)
           {
              array1[i] = analogRead(FSR_Pin); //obtain a set of analog values
for player 1 (red led)
           }
        long int sum1 = 0; //intialize sum1
        for (int i=0; i<1000; i++)
           {
             sum1 = sum1 + array1[i]; //sum of array for player 1
           }
            Play1 = sum1/1000; //average of array for player a
```

```
      //array2
         for (int i=0; i<1000; i++)
             {
               array2[i] = analogRead(FSR_Pinn); //Player 2 (greenled) obtain a
set of analog values 1000 for robustness ish
             }
         long int sum2 = 0; //intialize sum2
         for (int i = 0; i < 1000; i++)
             {
                sum2 = sum2 + array2[i]; //sum of array for player 2
             }
               Play2 = sum2/1000; //average of array for player 2

      // Just for kicks
           LEDbrightness = map(Play1, 0, 4095, 0, 255);   // used by analogWrite
(0-255) with map, Energia maxes at 4095
           analogWrite(LED, LEDbrightness);  // LED gets brighter the harder you
press
           LEDbrightnesss = map(Play2, 0, 4095, 0, 255);   // used by analogWrite
(0-255) with map, Energia maxes at 4095
           analogWrite(LEDD, LEDbrightnesss);  // LED gets brighter the harder you
press
      }

     if (presscount == 2) //press the button once again to end round
        {
          //print out stuff for user to know the end of the round is here
          lcd.setCursor(0,0);
          lcd.print("Round 1 OVER    ");
          lcd.setCursor(0,1);
          lcd.print("Results  Loading");
          delay(1000);
        }

          while (presscount == 2) //show results for round 1
          {
            long int Now = millis(); //Just so the interupt can work to release
from the while loop
            if (Play1>Play2) // if player 1 wins
              {
                lcd.setCursor(0,0);
                lcd.print("Player1||Player2");
                lcd.setCursor(0,1);
                lcd.print("   *            ");
                winarray1[0] = 1; // points to be stored if player wins
```

```
            }
           else if  (Play1<Play2)  //if player 2 wins
           {
             lcd.setCursor(0,0);
             lcd.print("Player1||Player2");
             lcd.setCursor(0,1);
             lcd.print("            *   ");
             winarray2[0] = 1; // points to be stored if player wins
           }
           else if (Play1 == Play2); // if they tie
           {
           lcd.setCursor(0,0);
           lcd.print("Player1||Player2");
           lcd.setCursor(0,1);
           lcd.print("   *        *   ");
           }
       attachInterrupt(PIN_BUTTON, inc_button_count, FALLING) ;
        if (update_count>0)
           {
              update_count=0 ;
              if ((Now-lastpress)>DEBOUNCE)
              {
                presscount++ ;
                lcd.clear();
                lastpress=Now ;
              }
           }
       } // end of while loop

/////////////////Round 2\\\\\\\\\\\\\\\\\\\
    //print out stuff for user to know the start of round 2 is here
     if (presscount == 3)
      {
          lcd.setCursor(0, 0);
          lcd.print("Thumb Wrestling!");
          lcd.setCursor(0,1);
          lcd.print("Round 2         ");

      //obtainfsr to start game
            //array1
        for (int i=0; i<1000; i++)
           {
              array1[i] = analogRead(FSR_Pin); //obtain a set of analog values
for player 1 (red led)
           }
```

```
          long int sum1 = 0; //intialize sum1
        for (int i=0; i<1000; i++)
           {
              sum1 = sum1 + array1[i]; //sum of array for player 1
           }
            Play1 = sum1/1000; //average of array for player a

     //array2
        for (int i=0; i<1000; i++)
            {
              array2[i] = analogRead(FSR_Pinn); //Player 2 (greenled) obtain a
set of analog values 1000 for robustness ish
            }
          long int sum2 = 0; //intialize sum2
        for (int i = 0; i < 1000; i++)
            {
               sum2 = sum2 + array2[i]; //sum of array for player 2
            }
              Play2 = sum2/1000; //average of array for player 2
              // Just for kicks
         LEDbrightness = map(Play1, 0, 4095, 0, 255);   // used by analogWrite
(0-255) with map, Energia maxes at 4095
          analogWrite(LED, LEDbrightness);  // LED gets brighter the harder you
press
          LEDbrightnesss = map(Play2, 0, 4095, 0, 255);   // used by analogWrite
(0-255) with map, Energia maxes at 4095
          analogWrite(LEDD, LEDbrightnesss);  // LED gets brighter the harder you
press
      }

    if (presscount == 4) //press the button once again to end round 2
    //print out stuff for user to know the end of the round is here
      {
        lcd.setCursor(0,0);
        lcd.print("Round 2 OVER    ");
        delay(1000);
      }
       while (presscount == 4)
       {
              long int Now = millis();
              if (Play1>Play2)
              {
              lcd.setCursor(0,0);
              lcd.print("Player1||Player2");
              lcd.setCursor(0,1);
```

```
            lcd.print("   *            ");
            winarray1[1] = 1; // points to be stored if player wins
            }
            else if  (Play1<Play2)
            {
              lcd.setCursor(0,0);
              lcd.print("Player1||Player2");
              lcd.setCursor(0,1);
              lcd.print("            *   ");
              winarray2[1] = 1; // points to be stored if player wins
            }
            else if (Play1 == Play2);
            {
              lcd.setCursor(0,0);
              lcd.print("Player1||Player2");
              lcd.setCursor(0,1);
              lcd.print("   *        *   ");
            }
             attachInterrupt(PIN_BUTTON, inc_button_count, FALLING) ;
             if (update_count>0)
                {
                   update_count=0 ;
                   if ((Now-lastpress)>DEBOUNCE)
                   {
                     presscount++ ;
                     lcd.clear();
                     lastpress=Now ;
                   }
                }
       }

/////////////Round 3\\\\\\\\\\\\\\\
    //print out stuff for user to know the start of round 3
      if (presscount == 5)
      {
          lcd.setCursor(0, 0);
          lcd.print("Thumb Wrestling!");
          lcd.setCursor(0,1);
          lcd.print("Round 3 - FINALE");

       //get fsr for round 3
             //array1
        for (int i=0; i<1000; i++)
           {
              array1[i] = analogRead(FSR_Pin); //obtain a set of analog values
```

```
for player 1 (red led)
            }
         long int sum1 = 0; //intialize sum1
         for (int i=0; i<1000; i++)
            {
              sum1 = sum1 + array1[i]; //sum of array for player 1
            }
             Play1 = sum1/1000; //average of array for player a

      //array2
         for (int i=0; i<1000; i++)
             {
               array2[i] = analogRead(FSR_Pinn); //Player 2 (greenled) obtain a
set of analog values 1000 for robustness ish
             }
         long int sum2 = 0; //intialize sum2
         for (int i = 0; i < 1000; i++)
             {
                sum2 = sum2 + array2[i]; //sum of array for player 2
             }
               Play2 = sum2/1000; //average of array for player 2

               // Just for kicks
          LEDbrightness = map(Play1, 0, 4095, 0, 255);   // used by analogWrite
(0-255) with map, Energia maxes at 4095
          analogWrite(LED, LEDbrightness);  // LED gets brighter the harder you
press
          LEDbrightnesss = map(Play2, 0, 4095, 0, 255);   // used by analogWrite
(0-255) with map, Energia maxes at 4095
          analogWrite(LEDD, LEDbrightnesss);  // LED gets brighter the harder you
press
      }

     if (presscount == 6) //press the button once again to end round 3
        // Indicate that the round 3 is over
        {
          lcd.setCursor(0,0);
          lcd.print("Round 3 OVER    ");
          delay(1000);
        }

     if (presscount == 6)
     {
            if (Play1>Play2)
            {
```

```
            lcd.setCursor(0,0);
            lcd.print("Player1||Player2");
            lcd.setCursor(0,1);
            lcd.print("   *            ");
            winarray1[2] = 1; // points to be stored if player wins
            }
            else if  (Play1<Play2)
            {
              lcd.setCursor(0,0);
              lcd.print("Player1||Player2");
              lcd.setCursor(0,1);
              lcd.print("            *   ");
              winarray2[2] = 1; // points to be stored if player wins
            }
            else if (Play1 == Play2);
            {
              winarray1[2] = 1; // points to be stored if player wins
              lcd.setCursor(0,0);
              lcd.print("Player1||Player2");
              lcd.setCursor(0,1);
              lcd.print("****************");
            }
          delay(1000);
          int sum = winarray1[2]+winarray1[1]+winarray1[0]; //sum of player 1
points accumulated per round
          int suma = winarray2[2]+winarray2[1]+winarray2[0]; //sum of player 2
points accumulated per round

          if (sum>suma)
          {
            lcd.setCursor(0, 0);
            lcd.print("Thumb Wrestling!");
            lcd.setCursor(0,1);
            lcd.print("PLAYER 1 WINS!!!");
          }
          else
          {
            lcd.setCursor(0, 0);
            lcd.print("Thumb Wrestling!");
            lcd.setCursor(0,1);
            lcd.print("PLAYER 2 WINS!!!");
          }
          else if (sum == suma)
          {
            lcd.setCursor(0, 0);
```

```
          lcd.print("Thumb Wrestling!");
          lcd.setCursor(0,1);
          lcd.print("TIE!!! Try Again");
        }
        delay(3000);
        presscount = 0; // reset presscount to zero and restart the loop
 }


}
void inc_button_count()
{
  update_count=1 ;
}
```