

```

//libraries
#include <ezButton.h>
#include <LiquidCrystal.h>
#include <Servo.h>

#define VRX_PIN A0 // Arduino pin connected to VRX pin
#define VRY_PIN A1 // Arduino pin connected to VRY pin
#define SW_PIN 2 // Arduino pin connected to SW pin

ezButton button(SW_PIN); // uses the ezButton library on pin 2
Servo myservo; //creates an object that helps control servo

//established pins for lcd
const int RS = 7, EN = 8, D4 = 3, D5 = 4, D6 = 5, D7 = 6;
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);

int xValue = 0; // To store value of the X axis
int yValue = 0; // To store value of the Y axis
int bValue = 0; // To store value of the button
int trigPin = 10; // TRIG pin
int echoPin = 9; // ECHO pin
float duration_cm; // variable to hold value for ultrasonic sensor

int ledPins[] = { 13, 12, 11 }; //array that holds led pins

// arrays of chars that hold the messages for the lcd screen
char *initialMessage[] = { "great ", "okay ", "horrible" };
char *goodMessages0[] = { "That's good to", "Keep up the", "I'm here when", "Have a nice day"
};
char *goodMessages1[] = { "hear, Mark! ", "amazing work!", "you need me.", "Mark :)" };

char *okayMessages0[] = { "That's entirely", "It's fine to not", "Would you like", "Try to
take it", "Take a break ", "do one thing to", "I'm here when" };
char *okayMessages1[] = { "okay Mark... ", "always feel good", "general meds?", "easy
today.", "from work and...", "bring you joy.", "you need me." };

char *badMessages0[] = { "That's totally", "Some days we do", "And that's okay.", "Breathing
skills", "Breathe along", "I hope that", "Have some meds", "Take it slowly", "And I hope you"
};
char *badMessages1[] = { "valid, Mark.", "not feel good...", " ", "could help.", "with the
arrows.", "helped a little.", "to help out.", "today.", "feel better soon." };

char *yesNo[] = { "Yes", "No" };

int currentPin = -1; //variable to track the mood chosen

```

```

int count = 0; // count to traverse arrays

int start = 0; //variable to establish the start of if statements

int end = 0; //used to determine if the last section of each mood has been reached

int medication = 0; //yes is zero and no is 1 for medication

int pressState = 0; //variable that holds the press state of the joystick button

int initialDone = 0; //determines if first question has been posed

int inRange = 0; //hold value if ultrasonic sensor has picked anything up within a given
range

byte customChar1[8] = { //upward arrow custom character
  0b00100,
  0b01110,
  0b11111,
  0b00100,
  0b00100,
  0b00100,
  0b00100,
  0b00100
};

byte customChar2[8] = { //downward arrow custom character
  0b00100,
  0b00100,
  0b00100,
  0b00100,
  0b00100,
  0b11111,
  0b01110,
  0b00100
};

void setup() {
  lcd.begin(16, 2); // set up number of columns and rows
  lcd.createChar(1, customChar1); // create a new custom character (index 1)
  lcd.createChar(2, customChar2); // create a new custom character (index 2)
  myservo.attach(A4); // attaches the servo on pin A4 to the
servo object

```

```

    lcd.setCursor(0, 0); // move cursor to (0, 0)
    Serial.begin(9600); // turns on the serial monitor
    button.setDebounceTime(50); // set debounce time to 50 milliseconds for
the joystick button
    for (int thisPin = 0; thisPin < 4; thisPin++) { //traverses an array to establish pins for
leds
        pinMode(ledPins[thisPin], OUTPUT);
    }
    pinMode(trigPin, OUTPUT); // established trigger pin as output
    pinMode(echoPin, INPUT); // establishes echo pin as input
}

void loop() {

    pressState = button.isPressed();
    button.loop(); // MUST call the loop() function first
    // read analog X and Y analog values, and the button state
    xValue = analogRead(VRX_PIN);
    yValue = analogRead(VRY_PIN);
    bValue = button.getState();
    digitalWrite(trigPin, HIGH); //sends out signal from sonar sensor
    delayMicroseconds(10); //delay for 10 milliseconds
    digitalWrite(trigPin, LOW); //turns of the signal

    // measure duration of pulse from ECHO pin
    duration_cm = (pulseIn(echoPin, HIGH)) * .017; //mesures distances from object that relects
signal
    delay(250);
    if (duration_cm <= 15) { //if the duration is less than or equal to 15cm,
        // then the object is within reach, so rest of code begins
        inRange++;
    }

    if (inRange > 0) {
        if (start == 0 && initialDone == 0) { //displays first question
            lcd.setCursor(0, 0);
            lcd.print("How are you ");
            lcd.setCursor(0, 1);
            lcd.print("feeling today?");
            delay(2000);
            lcd.clear();
            start++;
        }
    }
}

```

```

if (start != 0 && initialDone == 0) { //calls initial question function
    initialQuestion();
    if (bValue == 0) { //if the button is pressed, then initial is one, and the first part
is done
        initialDone = 1;
        lcd.clear();
        start = 0;
        if (currentPin >= 1) { //used to clear up difficulties on later if statements
            count = -1;
        }
    }
}

if (initialDone == 1) { //once the initial questioning is done, go into specific
statemens/questions
    if (currentPin == 0) { //good sequence
        if (start == 0) { //displays first messages
            lcd.setCursor(0, 0);
            lcd.print("So, you're");
            lcd.setCursor(0, 1);
            lcd.print("feeling "), initialMessage[currentPin];
            lcd.setCursor(8, 1);
            lcd.print(initialMessage[currentPin]);
            delay(3000);
            start++;
            lcd.clear();
        }
        if (start != 0) { //traverses good statement, using input from the joystick
            if (count >= 0) {
                Serial.println(count);
                //lcd.clear();
                lcd.setCursor(0, 0);
                lcd.print(goodMessages0[count]);
                lcd.setCursor(0, 1);
                lcd.print(goodMessages1[count]);
                delay(250);
                if (225 >= xValue <= 725 && yValue <= 10) { //the actual if statement that uses
the joystick
                    count++;
                    lcd.clear();
                    if (count == 4) {
                        count = 0;
                    }
                }
            }
        }
    }
}

```

```

    }
}
if (currentPin == 1) { //if the person chooses okay

    if (start == 0 && end == 0) { //traverse first half of the okay array, until count is
3
        if (225 >= xValue <= 725 && yValue <= 10) {
            lcd.clear();
            count++;
            if (count == 3) {
                start++;
                lcd.clear();
            }

            lcd.setCursor(0, 0);
            lcd.print(okayMessages0[count]);
            lcd.setCursor(0, 1);
            lcd.print(okayMessages1[count]);
        }
        delay(250);
    }
    if (start != 0 && end == 0) { // ask the individual if they would like medicatio with
yesNo array
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(yesNo[medication]);
        if (225 >= xValue <= 725 && yValue <= 10) {
            medication++;
            lcd.clear();
            if (medication == 2) {
                medication = 0;
            }
        }
        delay(250);
        if (bValue == 0 && medication == 0) { //if the individual would like medicine,
dispense using the dispense() fxn
            Serial.println(" meds");
            lcd.setCursor(0, 0);
            lcd.print("Dispensing meds...");
            delay(2000);
            dispense();
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Here, take these");
            lcd.setCursor(0, 1);

```

```

        lcd.print("...");
        delay(2000);
        end++;
        lcd.clear();
    }
    if (bValue == 0 && medication == 1) { //if the individual would not like medicine,
clear the lcd screen
        lcd.clear();
    }
}
if (end != 0) { //once the medication sequence is done, display the final messages,
using joystick input
    lcd.setCursor(0, 0);
    lcd.print(okayMessages0[count]);
    lcd.setCursor(0, 1);
    lcd.print(okayMessages1[count]);
    if (225 >= xValue <= 725 && yValue <= 10) {
        count++;
        lcd.clear();
        if (count == 7) {
            count = 4;
        }
    }
    delay(250);
}
}
if (currentPin == 2) { //if the person chooses horrible during first questions
    if (start == 0 && end == 0) { //displa the first five messages in the bad array
        if (225 >= xValue <= 725 && yValue <= 10) {
            lcd.clear();
            count++;
            if (count == 5) {
                start++;
                lcd.clear();
            }

            lcd.setCursor(0, 0);
            lcd.print(badMessages0[count]);
            lcd.setCursor(0, 1);
            lcd.print(badMessages1[count]);
        }
        delay(250);
    }
}
if (start != 0 && end == 0) { //if statement that holds breathing exercise
    lcd.clear();

```

```

    lcd.setCursor(0, 1);
    lcd.print("                ");
    for (int i = 0; i < 17; i++) {
        if ((i % 2) == 0) { //if i is even, then display an up arrow and delay for 5
seconds
            lcd.setCursor(i, 0);
            lcd.write((byte)1);
            delay(5000);
        } else { //if i is odd, then display a down arrow and delay for 3 seconds
            lcd.setCursor(i, 0);
            lcd.write((byte)2);
            delay(3000);
        }
    }
    end++;
}
if (end != 0) { //once the breathing exercises are over, display the last messages in
the bad array
    lcd.setCursor(0, 0);
    lcd.print(badMessages0[count]);
    lcd.setCursor(0, 1);
    lcd.print(badMessages1[count]);
    if (count == 7) {
        dispense();
    }
    if (225 >= xValue <= 725 && yValue <= 10) {
        count++;
        lcd.clear();
        if (count == 9) {
            count = 6;
        }
    }
    delay(250);
}
}
}
}
}

int initialQuestion() { //traverse an array holding the first messages, using the joystick to
filter through options for mood

    lcd.setCursor(0, 0);
    lcd.print("I'm feeling: ");
    if (225 >= xValue <= 725 && yValue <= 10) {

```

```
digitalWrite(ledPins[currentPin], LOW);
currentPin++;
if (currentPin > 2) {
  currentPin = 0;
}
digitalWrite(ledPins[currentPin], HIGH);
lcd.setCursor(0, 1);
lcd.print("-");
lcd.setCursor(1, 1);
lcd.print(initialMessage[currentPin]);
}
delay(250);
}

//says encouraging things, great, have an amazing day, keep at it...

int dispense() { //dispense medicine by turning the servo motor

  myservo.write(0); // tell servo to go to position in variable 'pos'
  delay(2000);
  myservo.write(360); // tell servo to go to position in variable 'pos'
  delay(3000);
}
```