

```

/* Automated Chicken Feeder:
   This device will be used to automatically dispense chicken feed
*/

#include <LiquidCrystal.h> // Need to include the LCD library
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Pins connecting the LCD to the Genuino

// Variables:
const int switchPin = 13; // Declaration of variable with pin
const int motorPin = 10; // Declaration of variable with pin
int switchState = 0; // Declaration of variable starting at 0
unsigned long currentTime = 0; // Declaration of variable starting at 0
unsigned long feedTime = 0; // Declaration of variable starting at 0
unsigned long dayTime = 0; // Declaration of variable starting at 0
const int sensorPin = A0; // Declaration of variable with pin
const float baselineTemp = 17.0; // Set room temp. according to serial monitor

void setup() {

  pinMode(switchPin, INPUT); // Initialize the pushbutton pin as an input
  pinMode(motorPin, OUTPUT); // Initialize the motor pin as an output
  Serial.begin(9600); // Initialize serial monitor for debug messages
  lcd.begin(16, 2); // Initialize the LCD
  lcd.setCursor(1, 0); // Sets position on the LCD for the title
  lcd.print("Chicken Feeder"); // Prints title Chicken Feeder to LCD
}

void loop() {

  int sensorVal = analogRead(sensorPin); // Read status of signal from sensor pin
  Serial.print(" Sensor Value: "); // Prints Sensor Value label to serial display
  Serial.print(sensorVal); // Prints the actual sensorVal number to serial display
  float voltage = (sensorVal / 1024.0) * 5.0; // Conversion factor to voltage
  Serial.print(" Volts: "); // Prints Volts label to serial display
  Serial.print(voltage); // Prints the actual voltage number to serial display
  Serial.print(" Temperature: "); // Prints Temperature label to serial display
  float temperature = (voltage - 0.5) * 100; // Conversion factor from V. to Temp.
  Serial.println(temperature); // Prints actual temperature # to serial display

  if (temperature < baselineTemp + 2) { // If Temp. < then initial set Temp:
    lcd.setCursor(2, 1); // Sets position on the LCD for the title
    lcd.print("Heat Chickens"); // Prints title Heat Chickens to LCD
  }
}

```

```

if (temperature > baselineTemp + 2) { // If Temp. > then initial set Temp:
  lcd.setCursor(2, 1); // Sets position on the LCD for the title
  lcd.print("          "); // Clears the display
}

digitalWrite(motorPin, LOW); // Turn motor off
switchState = digitalRead(switchPin); // Read status of signal from switch pin

if (switchState == HIGH) { // If the switch is reading high, do this:
  digitalWrite(motorPin, HIGH); // Turn motor on
  lcd.setCursor(1, 0); // Sets position on the LCD for the title
  lcd.print(" Dispensing "); // Prints title Dispensing to LCD
  delay(500); // Wait for 0.5 seconds
  dayTime = currentTime; // Resets timer for next dispence
}

digitalWrite(motorPin, LOW); // Turn motor off
currentTime = millis(); // Sets the start time in millis (1000 ms per s)
feedTime = currentTime - dayTime; // Sets feedtime

if (feedTime >= 19000) { // Variable for feed time (can be changed accordingly)
  dayTime = currentTime; // Current time is equal to day time
  digitalWrite(motorPin, HIGH); // Turn motor on
  lcd.setCursor(1, 0); // Sets position on the LCD for the title
  lcd.print(" Dispensing "); // prints title Dispensing to LCD
  delay(9000); // Wait for 9 seconds (motor will be on for this much time)
  digitalWrite(motorPin, LOW); // Turn motor off
}

lcd.setCursor(1, 0); // Sets position on the LCD for the title
lcd.print("Chicken Feeder"); // Print: Chicken Feeder to LCD out of any loop
}

```