CODE FOR ULTRASONIC SENSOR AND BUZZER

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 13;

// defines variables
long duration;
int distance;
int safetyDistance;

void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(buzzer, OUTPUT);
pinMode(ledPin, OUTPUT);
Serial.begin(9600); // Starts the serial communication
}

void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

```
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance= duration*0.034/2;

safetyDistance = distance;
if (safetyDistance <= 10){
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
else{
  digitalWrite(buzzer, LOW);
  digitalWrite(ledPin, LOW);
}

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
}
```

CODE FOR CAMERA MODULE

OV7670 connections:

VSYNC - PIN2
XCLCK - PIN3 (must be level shifted from 5V -> 3.3V)
PCLCK - PIN12
SIOD - A4 (I2C data) - 10K resistor to 3.3V
SIOC - A5 (I2C clock) - 10K resistor to 3.3V
D0..D3 - A0..A3 (pixel data bits 0..3)
D4..D7 - PIN4..PIN7 (pixel data bits 4..7)
3.3V - 3.3V
RESET - 3.3V
GND - GND
PWDN - GND

1.8" TFT connections:

DC - PIN8 (5V -> 3.3V)
CS - PIN9 (5V -> 3.3V)
RESET - PIN10 (5V -> 3.3V)
SPI data - PIN11 (5V -> 3.3V)
SPI clock - PIN13 (5V -> 3.3V)
VCC - 5V/3.3V (depending on jumper position on the TFT board)
BL - 3.3V
GND - GND

CODE FOR FINGERPRINT SENSOR

```
#include <Adafruit_Fingerprint.h>

// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white

// uncomment this line:

// #define mySerial Serial1

// For UNO and others without hardware serial, we must use software serial...

// pin #2 is IN from sensor (GREEN wire)

// pin #3 is OUT from arduino (WHITE wire)

// comment these two lines if using hardware serial

#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint  finger  = Adafruit_Fingerprint(&mySerial);

void setup()

{

pinMode(4,OUTPUT);

pinMode(5,OUTPUT);

Serial.begin(9600);

while (!Serial); // For Yun/Leo/Micro/Zero/...

delay(100);

Serial.println("\n\nAdafruit finger detect test");

// set the data rate for the sensor serial port

finger.begin(57600);

```

```
23)  if (finger.verifyPassword()) {
24)  Serial.println("Found fingerprint sensor!");
25)  } else {
26)  Serial.println("Did not find fingerprint sensor :(");
27)  while (1) { delay(1); }
28)  }
29)  finger.getTemplateCount();
30)  Serial.print("Sensor contains "); Serial.print(finger.templateCount);
Serial.println(" templates");
31)  Serial.println("Waiting for valid finger...");
32)  }
33)  void loop()  // run over and over again
34)  {
35)  getFingerprintIDez();
36)  delay(50);  //don't ned to run this at full speed.
37)  digitalWrite(5,HIGH);
38)  }
39)  uint8_t getFingerprintID() {
40)  uint8_t p = finger.getImage();
41)  switch (p) {
42)  case FINGERPRINT_OK:
43)  Serial.println("Image taken");
44)  break;
45)  }
46)  // OK success!
47)  p = finger.image2Tz();
48)  switch (p) {
49)  case FINGERPRINT_OK:
50)  Serial.println("Image converted");
51)  break;
52)  // found a match!
```

```
53)     digitalWrite(5,LOW);
54)     digitalWrite(4,HIGH);
55)     delay(1000);
56)     digitalWrite(4,LOW);
57)
58)     Serial.print("Found ID #"); Serial.print(finger.fingerID);
59)     Serial.print(" with confidence of "); Serial.println(finger.confidence);
60)     return finger.fingerID;
61)     }
```

CODE FOR LOAD CELL

```
#include <HX711_ADC.h> // need to install
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // need to install

HX711_ADC LoadCell(6, 7); // parameters: dt pin 6, sck pin 7;
LiquidCrystal_I2C lcd(0x27, 16,2); // 0x27 is the i2c address might different;you can check with Scanner
void setup()
{
  LoadCell.begin(); // start connection to HX711
  LoadCell.start(2000); // load cells gets 2000ms of time to stabilize
  LoadCell.setCalFactor(1000.0); // calibration factor for load cell => dependent on your individual setup
 lcd.init();
lcd.backlight();
const int buzzerPin = 10;
const int buttonPin = 8;
pinMode(8,OUTPUT);
pinMode(buzzerPin, OUTPUT);
```

```
bool buzzerState = false;
}
void loop()
{
 LoadCell.update(); // retrieves data from the load cell
float i = LoadCell.getData(); // get output value
lcd.setCursor(0, 0); // set cursor to first row
lcd.print("Weight[g]:"); // print out to LCD
lcd.setCursor(0, 1); // set cursor to second row
lcd.print(i); // print out the retrieved value to the second row
if(i>20)
{
digitalWrite(8,HIGH);
digitalWrite(10,HIGH);
buzzerState = true;
} else
{
digitalWrite(8, LOW);
digitalWrite(buzzerPin, LOW);
buzzerState = false;
}
}
```