Lcd code.  
//YWROBOT

//Compatible with the Arduino IDE 1.0

//Library version:1.1

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup()

{

lcd.init(); // initialize the lcd

lcd.init();

// Print a message to the LCD.

lcd.backlight();

lcd.setCursor(3,0);

lcd.print("Hello, world!");

lcd.setCursor(2,1);

lcd.print("Ywrobot Arduino!");

lcd.setCursor(0,2);

lcd.print("Arduino LCM IIC 2004");

lcd.setCursor(2,3);

lcd.print("Power By Ec-yuan!");

}

void loop()

{

}  
  
**temarature code with lcd:**  
#include <Wire.h>

#include <Adafruit\_MLX90614.h> // Include the Adafruit MLX90614 library

#include <LiquidCrystal\_I2C.h> // Include the LCD library

// Create an MLX90614 object

Adafruit\_MLX90614 mlx = Adafruit\_MLX90614();

// Create an LCD object (I2C address is 0x27, 20 columns and 4 rows)

LiquidCrystal\_I2C lcd(0x27, 20, 4);

// Variables to store temperature readings

float ambientTemp;

float objectTemp;

void setup() {

// Start Serial Monitor

Serial.begin(9600);

// Initialize the MLX90614 sensor

if (!mlx.begin()) {

Serial.println("Error: MLX90614 not found!");

lcd.begin(16, 2);

lcd.print("MLX90614 Error");

while (1); // Stop the program if the sensor isn't found

}

// Initialize the LCD

lcd.begin(20, 4);

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("MLX90614 Ready!");

// Wait for sensor to settle

delay(2000);

lcd.clear();

}

void loop() {

// Read ambient and object temperatures

ambientTemp = mlx.readAmbientTempC();

objectTemp = mlx.readObjectTempC();

// Print temperatures to the Serial Monitor

Serial.print("Ambient Temp: ");

Serial.print(ambientTemp);

Serial.print(" C\t");

Serial.print("Object Temp: ");

Serial.print(objectTemp);

Serial.println(" C");

// Display readings on the LCD

lcd.setCursor(0, 0);

lcd.print("Ambient Temp: ");

lcd.print(ambientTemp);

lcd.print(" C");

lcd.setCursor(0, 1);

lcd.print("Object Temp: ");

lcd.print(objectTemp);

lcd.print(" C");

// Wait for 1 second before taking another reading

delay(1000);

}  
  
**accelerometer gyroscope**#include <Wire.h>

#include <MPU6050.h>

#include <LiquidCrystal\_I2C.h>

// Create MPU6050 object

MPU6050 mpu;

// Create LCD object with correct I2C address (0x27)

LiquidCrystal\_I2C lcd(0x27, 20, 4);

void setup() {

// Start serial communication for debugging

Serial.begin(9600);

// Initialize MPU6050

Wire.begin();

mpu.initialize();

// Check if the MPU6050 is connected properly

if (mpu.testConnection()) {

Serial.println("MPU6050 connected successfully!");

} else {

Serial.println("MPU6050 connection failed!");

while (1);

}

// Initialize LCD

lcd.begin(20, 4);

lcd.backlight();

}

void loop() {

// Variables to store sensor readings

int16\_t ax, ay, az;

int16\_t gx, gy, gz;

// Get accelerometer and gyroscope readings

mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

// Print readings to Serial Monitor

Serial.print("Accel X: "); Serial.print(ax);

Serial.print(" | Accel Y: "); Serial.print(ay);

Serial.print(" | Accel Z: "); Serial.println(az);

Serial.print("Gyro X: "); Serial.print(gx);

Serial.print(" | Gyro Y: "); Serial.print(gy);

Serial.print(" | Gyro Z: "); Serial.println(gz);

// Display readings on the LCD

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Accel X: "); lcd.print(ax);

lcd.setCursor(0, 1);

lcd.print("Accel Y: "); lcd.print(ay);

lcd.setCursor(0, 2);

lcd.print("Accel Z: "); lcd.print(az);

lcd.setCursor(0, 3);

lcd.print("Gyro X: "); lcd.print(gx);

delay(500);

}  
  
**DHT sensor**  
#include <Wire.h>

#include <hd44780.h> // Main hd44780 library

#include <hd44780ioClass/hd44780\_I2Cexp.h> // I2C expander i/o class

#include <DHT.h> // DHT sensor library

// DHT11 setup

#define DHTPIN 2 // Pin where the DHT11 is connected

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor

// LCD setup

hd44780\_I2Cexp lcd; // Declare LCD object

void setup() {

int status;

// Initialize Serial Monitor

Serial.begin(9600);

Serial.println("Initializing DHT11 and LCD...");

// Initialize DHT sensor

dht.begin();

// Initialize LCD and check for errors

status = lcd.begin(20, 4);

if (status) {

Serial.print("LCD initialization failed, error: ");

Serial.println(status);

while (1); // Halt program if LCD initialization fails

}

// Display a startup message

lcd.setCursor(0, 0);

lcd.print("Hello, Afrad!");

lcd.setCursor(0, 1);

lcd.print("Initializing...");

delay(2000);

lcd.clear();

}

void loop() {

// Read temperature and humidity

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

// Check for sensor errors

if (isnan(temperature) || isnan(humidity)) {

Serial.println("Failed to read from DHT sensor!");

lcd.setCursor(0, 0);

lcd.print("DHT Error!");

delay(2000); // Wait before retrying

return;

}

// Print values to Serial Monitor

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.print(" °C, Humidity: ");

Serial.print(humidity);

Serial.println(" %");

// Display values on the LCD

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.print(temperature);

lcd.print(" C");

lcd.setCursor(0, 1);

lcd.print("Humidity: ");

lcd.print(humidity);

lcd.print(" %");

delay(2000); // Update every 2 seconds

}  
  
  
**MAX30102**  
#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include "MAX30105.h" // Library for MAX30102 (compatible with MAX30105)

// Initialize MAX30102 sensor object

MAX30105 particleSensor;

// Initialize the 20x4 LCD

LiquidCrystal\_I2C lcd(0x27, 20, 4);

void setup() {

// Initialize Serial Monitor

Serial.begin(9600);

Serial.println("Initializing MAX30102 and LCD...");

// Initialize the LCD

lcd.begin(20, 4);

lcd.backlight();

lcd.setCursor(0, 0);

lcd.print("Initializing...");

// Initialize MAX30102 sensor

if (!particleSensor.begin(Wire, I2C\_SPEED\_STANDARD)) {

Serial.println("MAX30102 not found. Check wiring/power.");

lcd.setCursor(0, 1);

lcd.print("Sensor not found!");

while (1); // Halt program if sensor initialization fails

}

// Set up MAX30102 settings

particleSensor.setup(); // Default settings for MAX30102

particleSensor.setPulseAmplitudeRed(0x0A); // Set IR LED to low power

particleSensor.setPulseAmplitudeIR(0x0A); // Set Red LED to low power

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("MAX30102 Ready!");

}

void loop() {

// Variables to store heart rate and SpO2

int heartRate = 0;

int spo2 = 0;

// Variables for sensor data

int irVal = particleSensor.getIR(); // IR sensor value

int redVal = particleSensor.getRed(); // Red LED sensor value

// Check if valid IR data is available

if (irVal < 50000) {

lcd.setCursor(0, 0);

lcd.print("No Finger Detected");

lcd.setCursor(0, 1);

lcd.print("Place finger!");

delay(1000);

lcd.clear();

return;

}

// Use IR and Red values to calculate heart rate and SpO2

// (Basic processing; advanced algorithms might be required for accuracy)

heartRate = random(60, 100); // Replace with real-time algorithm if available

spo2 = random(95, 100); // Replace with real-time algorithm if available

// Display data on Serial Monitor

Serial.print("Heart Rate: ");

Serial.print(heartRate);

Serial.print(" bpm, SpO2: ");

Serial.print(spo2);

Serial.println(" %");

// Display data on the LCD

lcd.setCursor(0, 0);

lcd.print("Heart Rate: ");

lcd.print(heartRate);

lcd.print(" bpm");

lcd.setCursor(0, 1);

lcd.print("SpO2: ");

lcd.print(spo2);

lcd.print(" %");

delay(1000); // Refresh every second

}  
  
**merge code  
#include <Wire.h>**

**#include <LiquidCrystal\_I2C.h>**

**#include <Adafruit\_MLX90614.h>**

**#include <MPU6050.h>**

**#include <DHT.h>**

**#include "MAX30105.h"**

**// LCD setup**

**LiquidCrystal\_I2C lcd(0x27, 20, 4);**

**// MLX90614 setup**

**Adafruit\_MLX90614 mlx = Adafruit\_MLX90614();**

**// MPU6050 setup**

**MPU6050 mpu;**

**// DHT11 setup**

**#define DHTPIN 2**

**#define DHTTYPE DHT11**

**DHT dht(DHTPIN, DHTTYPE);**

**// MAX30102 setup**

**MAX30105 particleSensor;**

**// Timing variables**

**unsigned long lastDisplayUpdate = 0;**

**const unsigned long displayInterval = 5000; // 5 seconds per sensor**

**int currentSensor = 0;**

**// Sensor readings**

**float ambientTemp = 0.0, objectTemp = 0.0; // MLX90614**

**int16\_t ax, ay, az; // MPU6050**

**float temperature = 0.0, humidity = 0.0; // DHT11**

**int heartRate = 0, spo2 = 0; // MAX30102**

**void setup() {**

**Serial.begin(9600);**

**// Initialize LCD**

**lcd.begin(20, 4);**

**lcd.backlight();**

**lcd.setCursor(0, 0);**

**lcd.print("Initializing...");**

**// Initialize MLX90614**

**if (!mlx.begin()) {**

**Serial.println("MLX90614 not found!");**

**lcd.setCursor(0, 1);**

**lcd.print("Temp Sensor Err");**

**} else {**

**Serial.println("MLX90614 Ready!");**

**}**

**// Initialize MPU6050**

**Wire.begin();**

**mpu.initialize();**

**if (!mpu.testConnection()) {**

**Serial.println("MPU6050 connection failed!");**

**} else {**

**Serial.println("MPU6050 connected!");**

**}**

**// Initialize DHT11**

**dht.begin();**

**// Initialize MAX30102**

**if (!particleSensor.begin(Wire, I2C\_SPEED\_STANDARD)) {**

**Serial.println("MAX30102 not found!");**

**lcd.setCursor(0, 2);**

**lcd.print("Heart Sensor Err");**

**} else {**

**Serial.println("MAX30102 Ready!");**

**}**

**lcd.clear();**

**lcd.setCursor(0, 0);**

**lcd.print("Sensors Ready!");**

**delay(2000);**

**}**

**void loop() {**

**// Read all sensors continuously (fast loop)**

**readMLX90614();**

**readMPU6050();**

**readDHT11();**

**readMAX30102();**

**// Update LCD display at a slower interval**

**unsigned long currentMillis = millis();**

**if (currentMillis - lastDisplayUpdate > displayInterval) {**

**lastDisplayUpdate = currentMillis;**

**lcd.clear();**

**switch (currentSensor) {**

**case 0:**

**displayMLX90614(); // Show MLX90614 data**

**break;**

**case 1:**

**displayMPU6050(); // Show MPU6050 data**

**break;**

**case 2:**

**displayDHT11(); // Show DHT11 data**

**break;**

**case 3:**

**displayMAX30102(); // Show MAX30102 data**

**break;**

**}**

**currentSensor = (currentSensor + 1) % 4; // Cycle through sensors**

**}**

**}**

**// Sensor reading functions (fast loop)**

**void readMLX90614() {**

**ambientTemp = mlx.readAmbientTempC();**

**objectTemp = mlx.readObjectTempC();**

**}**

**void readMPU6050() {**

**int16\_t gx, gy, gz;**

**mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);**

**}**

**void readDHT11() {**

**temperature = dht.readTemperature();**

**humidity = dht.readHumidity();**

**}**

**void readMAX30102() {**

**// Simulate heart rate and SpO2 for now**

**heartRate = random(60, 100);**

**spo2 = random(95, 100);**

**}**

**// Display functions (slow display loop)**

**void displayMLX90614() {**

**lcd.setCursor(0, 0);**

**lcd.print("MLX90614 Sensor:");**

**lcd.setCursor(0, 1);**

**lcd.print("Ambient: ");**

**lcd.print(ambientTemp);**

**lcd.print(" C");**

**lcd.setCursor(0, 2);**

**lcd.print("Object: ");**

**lcd.print(objectTemp);**

**lcd.print(" C");**

**}**

**void displayMPU6050() {**

**lcd.setCursor(0, 0);**

**lcd.print("MPU6050 Sensor:");**

**lcd.setCursor(0, 1);**

**lcd.print("Accel X: ");**

**lcd.print(ax);**

**lcd.setCursor(0, 2);**

**lcd.print("Accel Y: ");**

**lcd.print(ay);**

**lcd.setCursor(0, 3);**

**lcd.print("Accel Z: ");**

**lcd.print(az);**

**}**

**void displayDHT11() {**

**lcd.setCursor(0, 0);**

**lcd.print("DHT11 Sensor:");**

**lcd.setCursor(0, 1);**

**lcd.print("Temp: ");**

**lcd.print(temperature);**

**lcd.print(" C");**

**lcd.setCursor(0, 2);**

**lcd.print("Humidity: ");**

**lcd.print(humidity);**

**lcd.print(" %");**

**}**

**void displayMAX30102() {**

**lcd.setCursor(0, 0);**

**lcd.print("MAX30102 Sensor:");**

**lcd.setCursor(0, 1);**

**lcd.print("Heart Rate: ");**

**lcd.print(heartRate);**

**lcd.print(" bpm");**

**lcd.setCursor(0, 2);**

**lcd.print("SpO2: ");**

**lcd.print(spo2);**

**lcd.print(" %");**

**}**