**Challenges:**

**1-install library of sensor MAX30100.**

**2-uses of each sensor.**

**3-pins of 21 ,22 of I2C connection between MAX30100 and LCD has problem in connection but we try**

**the chip of MAX30100 with serial monitor because if we connect lcd not I2C with the max that use I2C the pins will not be enough**

A screen shot of a computer

Description automatically generated

**4- we connect key pad in our project but it uses 8 pins so then we face lack of pins to connect the other components .so we remove it to be able to connect the other components**

**A screenshot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A computer screen shot of text

Description automatically generated**

**A computer screen shot of a program code

Description automatically generated**

**A screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A black screen with text

Description automatically generated**

**Setup Phase:**

1. **Start Serial Communication:**
   * The Serial.begin(115200); command initializes serial communication at a baud rate of 115200. This is used for debugging and displaying messages in the Serial Monitor.
2. **Initialize I2C and LCD:**
   * Wire.begin(21, 22); sets up I2C communication with specific GPIO pins for SDA and SCL.
   * lcd.begin(16, 2); initializes the LCD with 16 columns and 2 rows.
   * lcd.backlight(); turns on the backlight of the LCD for better visibility.
3. **Initialize Sensors:**
   * sensors.begin(); initializes the DS18B20 temperature sensor.
   * The Pulse Oximeter (MAX30100) is initialized with pox.begin(). If it fails, the program halts with an error message.
4. **Setup Pins:**
   * Define and configure GPIO pins for IR sensor, ECG sensor, pressure sensor, LEDs, buzzer, and servo motor.
   * Set LED and buzzer pins to OUTPUT mode and initialize the servo motor to 0 degrees.
5. **Configure Pulse Oximeter:**
   * Set the IR LED current and specify a callback function to handle heartbeat detection.

A computer screen shot of a program

Description automatically generated

A computer screen shot of a black background

Description automatically generated

**2. Main Process**

**Loop Phase:**

1. **Update Pulse Oximeter:**
   * pox.update(); checks and updates the Pulse Oximeter readings for SpO2 and heart rate.
2. **Read IR Sensor:**
   * bool humanDetected = digitalRead(IR\_SENSOR\_PIN) == LOW; reads the IR sensor to determine if a human is present.
3. **If Human Detected:**
   * **Timing Control:**
     + The millis() function provides the current time in milliseconds. It is used to manage the timing of sensor data updates without blocking the program.
     + Update sensor readings every 5 seconds using the previousMillis and interval variables.
   * **Display Sensor Data:**
     + Depending on the mode (displayMode), the program updates the LCD with data from one of the sensors:
       - **Mode 0:** Displays temperature from DS18B20.
       - **Mode 1:** Displays ECG values.
       - **Mode 2:** Displays pressure values.
       - **Mode 3:** Displays SpO2 and heart rate from MAX30100.
     + The servo.write() function moves the servo to different angles based on the mode.
4. **If No Human Detected:**
   * **Display Message:**
     + Show a message "No human detected" on the LCD indicating the system is in standby mode.
   * **Turn Off Alerts:**
     + Ensure the green LED, red LED, and buzzer are turned off.
   * **Reset Servo:**
     + Move the servo back to its initial position (0 degrees).

**3. Sensor Classification**

**Classification Functions:**

* **Temperature (classifyTemperature(float temperature)):**
  + Categorizes temperature readings as "Low", "Normal", or "High" based on predefined thresholds.
* **ECG (classifyECG(float ecgValue)):**
  + Determines whether ECG values are "Normal" or "High".
* **Pressure (classifyPressure(float pressureValue)):**
  + Classifies pressure readings as "Normal" or "High".
* **SpO2 (classifySpO2(float spO2Value)):**
  + Indicates if SpO2 levels are "Low" or "Normal".

**4. Alerts and Notifications**

* **Trigger Alerts (triggerAlert()):**
  + Activates the red LED, buzzer, and moves the servo to 180 degrees to signal an alert condition.
  + The alert persists for 2 minutes before the servo returns to 0 degrees, and the buzzer is turned off.
* **Heartbeat Detection (onBeatDetected()):**
  + A callback function that prints a message to the Serial Monitor whenever a heartbeat is detected by the Pulse Oximeter.