CODING:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(4, 3); // RX, TX

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0X27, 16, 2);

int pulse_count = 0;
unsigned long start_time = 0;
unsigned long end_time = 5000000;

int sound = 2;
int sound_vls = 0;
int urine = 7;
int urine_vls = 0;
int buzzer = A2;
float temp_vls = 0;
int pulse_vls = 0;
int mp3_count = 0;
#include "DFRobotDFPlayerMini.h"
#include <AltSoftSerial.h>
// Use pins 2 and 3 to communicate with DFPlayer Mini
static const uint8_t PIN_MP3_TX = 8; // Connects to module's RX
static const uint8_t PIN_MP3_RX = 9; // Connects to module's TX
//SoftwareSerial softwareSerial(PIN_MP3_RX, PIN_MP3_TX);
AltSoftSerial  AltsoftwareSerial;
// Create the Player object
DFRobotDFPlayerMini player;

void setup() {
Serial.begin(9600);
mySerial.begin(4800);
pinMode(sound, INPUT);
pinMode(urine, INPUT);
pinMode(buzzer, OUTPUT);
AltsoftwareSerial.begin(9600);

lcd.begin();
// Turn on the blacklight and print a message.
lcd.backlight();
lcd.print("     Baby   ");
lcd.setCursor(1, 1);
lcd.print("  Monitoring   ");
delay(2000);
}


void loop() {

//player.stop();
display_fun();
pulse_fun();
//delay(2000);
sound_fun();
urine_fun();
temp_fun();
buzzer_fun_ON();
buzzer_fun_OFF();


//  display();
}
```

```
void display_fun() {

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Tmp :");
lcd.setCursor(7, 0);
lcd.print(temp_vls);



lcd.setCursor(0, 1);
lcd.print("Pulse :");
lcd.setCursor(7, 1);
lcd.print(pulse_vls);



mySerial.print("\@");
mySerial.print("\n");

mySerial.print(temp_vls);
mySerial.print("\n");

mySerial.print(pulse_vls);
mySerial.print("\n");

mySerial.print(urine_vls);
mySerial.print("\n");
delay(1000);

}
```

```
void temp_fun() {
int sensorValue = analogRead(A0);
float voltage = sensorValue * (5.0 / 1023.0);
float temperature = voltage * 100;
Serial.println("Voltage : ");
Serial.println(voltage);
temp_vls = temperature;
Serial.println("Temperature : ");
Serial.println(temperature);
//delay(500);
}

void buzzer_fun_ON() {

digitalWrite(buzzer, HIGH);
lcd.clear();
lcd.setCursor(4, 0);
lcd.print("Urine Alert ");
//delay(2000);
}

void buzzer_fun_OFF() {
digitalWrite(buzzer, LOW);
}

void urine_fun() {
urine_vls = digitalRead(urine);
if (urine_vls == 0) {
Serial.println("Urine alert");
digitalWrite(buzzer, HIGH);
buzzer_fun_ON();
```

```
}
else {
digitalWrite(buzzer, LOW);
buzzer_fun_OFF();
}
}

void sound_fun() {
sound_vls = digitalRead(sound);
Serial.println("sound :");
Serial.println(sound_vls);
if (sound_vls == 1) {
// play songs
Serial.println("Play Song");

mp3_player();

}
}



void mp3_player() {
if (player.begin(AltsoftwareSerial)) {
Serial.println("OK");
// Set volume to maximum (0 to 30).
player.volume(30);
player.stop();
delay(1000);
player.play(1);
delay(10000);
player.stop();
//    player.stop();
//    player.stop();
Serial.println("Play Song");
delay(1000);


//    lcd.clear();
//    lcd.setCursor(4, 0);
//    lcd.print("Play Song ");
} else {
Serial.println("Connecting to DFPlayer Mini failed!");
//        lcd.clear();
//        lcd.setCursor(0, 0);
//        lcd.print("Song failed!");
mp3_player();
}



}


void pulse_fun() {

pulse_count = 0;
start_time = micros();
end_time = start_time + 5000000;
while (start_time < end_time) {
sound_fun();
urine_fun();
temp_fun();
```

```
start_time = micros();
if (mySerial.available()) {
Serial.println(mySerial.read());
pulse_count++;
}
}

Serial.println("-----------------------");

Serial.println(pulse_count);
Serial.println("-----------------------");

if (pulse_count > 5) {
pulse_vls = pulse_count * 12;
} else {
pulse_vls = 0;
}

}
```