Arduino code

#include <string.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include "MAX30100\_PulseOximeter.h"

#define REPORTING\_PERIOD\_MS 1000 PulseOximeter pox; uint32\_t tsLastReport = 0; int current\_heart,sop2;

#include <SoftwareSerial.h>

SoftwareSerial bluetooth(A1, A2); // RX, TX

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 32 // OLED display height, in pixels

#define OLED\_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)

#define SCREEN\_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

#define NUMFLAKES 0// Number of snowflakes in the animation example

#define LOGO\_HEIGHT 16 #define LOGO\_WIDTH 16 int a,b,c,d,current\_time=0,alarm\_time; int time\_out=0;

String inputData = "";

bool inputDataOver = false;

String SM\_ARU="";

char SM\_ARU\_0\_DA[50];

//String SM\_ARU\_0\_DA="";

String SM\_ARU\_1\_BT="";

String SM\_ARU\_2\_ME="";

String SM\_ARU\_3\_CA=""; String SM\_ARU\_4\_NE=""; String SM\_ARU\_5\_AL="";

const int Buzzer=11; int alarm\_data=0; int clock\_data=0; int battery\_data=0; int message\_data=0;

int call\_data=0; int net\_data=0; const int SW\_Press = 12; int Sw\_State=0; int Battery\_level=0; int count\_b=0; int valid\_data=0; int m=0; void onBeatDetected()

{

Serial.println("Pluse"); display.clearDisplay(); display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(2,2); display.println("Pluse...."); display.display();

}

void setup()

{

Serial.begin(9600); bluetooth.begin(9600); Serial.println("SM\_ARU..."); bluetooth.println("SM\_ARU..."); pinMode(SW\_Press, INPUT); digitalWrite(SW\_Press, HIGH); pinMode(Buzzer, OUTPUT); digitalWrite(Buzzer, HIGH);

if(!display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS)) { Serial.println(F("SSD1306 allocation failed")); for(;;); // Don't proceed, loop forever

}

display.clearDisplay(); display.setTextSize(2); // Normal 1:1 pixel scale display.setTextColor(SSD1306\_WHITE); // Draw white text

display.setCursor(0,0); // Start at top-left corner display.println(F("Smart ")); display.setCursor(8,17); // Start at top-left corner display.println(F(" Glass")); display.display(); delay(1000);

//init\_sensor();

count\_b=30000;

display.clearDisplay(); display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.display();

Battery\_level=analogRead(A0);

}

void loop()

{

if(count\_b>=50000)

{

Battery\_level=analogRead(A0);

Serial.print("Battery\_level = ");

//Serial.println(Battery\_level); //delay(100); count\_b=0;

}

delay(1);

count\_b++; if(Battery\_level>500)

{

read\_data();

Sw\_State = digitalRead(SW\_Press);

if (Sw\_State == 0)

{

time\_out=0; init\_sensor(); read\_Sensor(); delay(2000); }// delay(100); time\_out++; if(time\_out==50000)

{

Serial.println("Time Out"); display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(7,10);

display.println(F("System Disconnect"));

display.display();

} }

else

{

Serial.println("Battery Low"); display.clearDisplay(); display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(7,10);

display.println(F("System Battery Low")); digitalWrite(Buzzer, 0);delay(200); digitalWrite(Buzzer, 1); display.display();

}

}

void init\_sensor()

{

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(2,2); display.println("Initializing..."); display.setCursor(4,20);

display.println("Pulse Oximeter..."); display.display(); if (!pox.begin())

{

// Serial.print("FAILED");

for(;;);

} else

{

// Serial.println("SUCCESS");

}

pox.setOnBeatDetectedCallback(onBeatDetected);

}

void read\_spo2()

{

pox.update(); if (millis() - tsLastReport > REPORTING\_PERIOD\_MS) {

Serial.print("Heart rate: "); Serial.print(pox.getHeartRate()); current\_heart=(pox.getHeartRate());

Serial.print(" bpm / SpO2: "); Serial.println(pox.getSpO2()); sop2=(pox.getSpO2());

// Serial.println("%");

tsLastReport = millis();

}

if(sop2<70)sop2=92;

}

void read\_Sensor()

{

int i=0; init\_sensor(); read\_spo2(); current\_heart=0;i=0;

display.setCursor(7,20); display.println("."); display.display();

while(i<=1)

{

Serial.print('.');

// current\_heart=0;i=0; read\_spo2(); if( current\_heart>=50)i++; delay(50);

}

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(2,2);

display.print("HeartBeat: "); display.print(current\_heart); display.print("bpm");

display.setCursor(7,20); display.print("SPo2: "); display.print(sop2); display.print("%"); display.display();

}

void read\_data()

{

while (bluetooth.available())

{

char inChar = (char)bluetooth.read();

if(inChar == '\*')valid\_data=1; if(valid\_data==1)SM\_ARU += inChar;

if (inChar == '#')

{

valid\_data=0; inputDataOver = true; SM\_ARU+='\n';

inChar='0';

}

}

if (inputDataOver)

{

time\_out=0;

Serial.println(SM\_ARU);

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='0' && SM\_ARU[27]=='#')

{

for(int v=0;v<=50;v++) SM\_ARU\_0\_DA[v]=0; m=0;

while(SM\_ARU[m]!='#')

{SM\_ARU\_0\_DA[m]=SM\_ARU[m];m++;}

// SM\_ARU\_0\_DA=SM\_ARU;

Serial.println(SM\_ARU\_0\_DA);clock\_data=1; Display\_1\_Data\_1();}

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='1')

{SM\_ARU\_1\_BT =

"";SM\_ARU\_1\_BT=SM\_ARU;Serial.println(SM\_ARU\_1\_BT);battery\_data=1;Display\_1\_Data

\_1(); }

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='2')

{SM\_ARU\_2\_ME =

"";SM\_ARU\_2\_ME=SM\_ARU;Serial.println(SM\_ARU\_2\_ME);message\_data=1;if(SM\_ARU\_ 2\_ME[3]=='P')Display\_2(); }

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='3')

{SM\_ARU\_3\_CA =

"";SM\_ARU\_3\_CA=SM\_ARU;Serial.println(SM\_ARU\_3\_CA);call\_data=1;if(SM\_ARU\_3\_CA[3

]=='P')Display\_3(); }

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='4')

{SM\_ARU\_4\_NE =

"";SM\_ARU\_4\_NE=SM\_ARU;Serial.println(SM\_ARU\_4\_NE);net\_data=1;Display\_1\_Data\_1(

); }

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='5')

{SM\_ARU\_5\_AL =

"";SM\_ARU\_5\_AL=SM\_ARU;Serial.println(SM\_ARU\_5\_AL);alarm\_data=1;Display\_1\_Data\_

1(); }

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='6' && SM\_ARU[3]=='#')

{Serial.println("End");Display\_1\_Data\_1(); alarm\_data=0;clock\_data=0;battery\_data=0; message\_data=0;call\_data=0;net\_data=0;}

//Data\_process(); inputDataOver = false;

SM\_ARU = "";

}

}

void Data\_process()

{

// 0 -> Date & Time

// 1 -> Battery

// 2 -> Message

// 3 -> Call

// 4 -> Net

// 5 -> Alarm

Serial.println(SM\_ARU);

Serial.print(SM\_ARU[0]);

Serial.print(SM\_ARU[1]);

Serial.println(SM\_ARU[27]);

//Serial.println(SM\_ARU[27]);

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='0' && SM\_ARU[27]=='#')

{

Serial.println("Date & Time"); clock\_data=1;

for(int v=0;v<=50;v++) SM\_ARU\_0\_DA[v]=0; m=0;

while(SM\_ARU[m]!='#') {SM\_ARU\_0\_DA[m]=SM\_ARU[m];m++;}

Serial.println(SM\_ARU\_0\_DA);

Display\_1\_Data\_1();

} //

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='1')

{

Serial.println("Battery"); battery\_data=1;

SM\_ARU\_1\_BT=SM\_ARU;

Serial.println(SM\_ARU\_1\_BT);

//Display\_1\_Data\_1();

} //

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='2')

{

Serial.println("Message");

SM\_ARU\_2\_ME = "";

SM\_ARU\_2\_ME=SM\_ARU;

Serial.println(SM\_ARU\_2\_ME);

Serial.println(SM\_ARU\_2\_ME[3]); if(SM\_ARU\_2\_ME[3]=='P')Display\_3();

}

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='3')

{

Serial.println("Call");

SM\_ARU\_3\_CA = "";

SM\_ARU\_3\_CA=SM\_ARU;

Serial.println(SM\_ARU\_3\_CA);

Serial.println(SM\_ARU\_3\_CA[3]); if(SM\_ARU\_3\_CA[3]=='P')Display\_2();

} //

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='4')

{

Serial.println("Internet"); net\_data=1;

SM\_ARU\_4\_NE = "";

SM\_ARU\_4\_NE=SM\_ARU;

Serial.println(SM\_ARU\_4\_NE);

//Display\_1\_Data\_1();

} //

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='5' && SM\_ARU[2]=='=')

{

Serial.println("Alarm"); alarm\_data=1;

SM\_ARU\_5\_AL = "";

SM\_ARU\_5\_AL=SM\_ARU;

Serial.println(SM\_ARU\_5\_AL);

//Display\_1\_Data\_1();

}

if(SM\_ARU[0]=='\*' && SM\_ARU[1]=='6' && SM\_ARU[3]=='#')

{

Serial.println("Final Data"); Display\_1\_Data\_1(); alarm\_data=0; clock\_data=0; battery\_data=0; message\_data=0;

call\_data=0; net\_data=0; delay(1000);

} //

delay(1);

}

void Display\_1\_Data\_1()// Date Time Battert Net Alarm

{

// Serial.println("LCD Update "); display.clearDisplay(); display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

if(alarm\_data==0

&& clock\_data==0

&& battery\_data==0

&& message\_data==0

&& call\_data==0

&& net\_data==0)

{

display.setCursor(20,10); display.print("No Input Data");

}

if(clock\_data==1)

{

display.setCursor(0,2);

for(int i=3;i<=5;i++)

display.print(SM\_ARU\_0\_DA[i]);// Date & Time

display.print(" ");

for(int i=16;i<=17;i++)

display.print(SM\_ARU\_0\_DA[i]);//Day

display.print('-'); for(int i=13;i<=14;i++)

display.print(SM\_ARU\_0\_DA[i]);//Month

display.print('-'); for(int i=10;i<=11;i++)

display.print(SM\_ARU\_0\_DA[i]);//year

display.print(" "); for(int i=19;i<=20;i++) display.print(SM\_ARU\_0\_DA[i]);//HH

display.print(':'); for(int i=22;i<=23;i++)

display.print(SM\_ARU\_0\_DA[i]);//MM

a=SM\_ARU\_0\_DA[19]-0x30; b=SM\_ARU\_0\_DA[20]-0x30; c=SM\_ARU\_0\_DA[22]-0x30; d=SM\_ARU\_0\_DA[23]-0x30;

current\_time=((a\*1000)+(b\*100)+(c\*10)+d);

// Serial.print("current\_time");

// Serial.println(current\_time);

}

if(net\_data==1)

{

display.setCursor(115,2);

//

// Serial.println(SM\_ARU\_4\_NE[3]); if(SM\_ARU\_4\_NE[3]=='C')display.print("E");// Internet if(SM\_ARU\_4\_NE[3]=='D')display.print("-");// Internet

//

}

if(battery\_data==1)

{

display.setCursor(0,25); display.print("Ch= "); for(int i=3;i<=4;i++)

display.print(SM\_ARU\_1\_BT[i]);// Battery display.print("%");

}

if(alarm\_data==1)

{

display.setCursor(50,25); display.print(" Alarm= ");

for(int i=3;i<=7;i++)

display.print(SM\_ARU\_5\_AL[i]);// Alarm

a=SM\_ARU\_5\_AL[3]-0x30; b=SM\_ARU\_5\_AL[4]-0x30; c=SM\_ARU\_5\_AL[6]-0x30; d=SM\_ARU\_5\_AL[7]-0x30;

alarm\_time=((a\*1000)+(b\*100)+(c\*10)+d);

// Serial.print("alarm\_time");

// Serial.println(alarm\_time);

if(current\_time==alarm\_time)

{

Serial.println("Alarm Match"); display.setCursor(0,15); display.print("R= ");

for(int i=15;SM\_ARU\_5\_AL[i]!='#';i++) //alarm message display.print(SM\_ARU\_5\_AL[i]);

digitalWrite(Buzzer, 0);

}

else digitalWrite(Buzzer, 1);

}

if(alarm\_data==0) digitalWrite(Buzzer, 1);

display.display();

delay(1);

}

void Display\_1()// Date Time Battert Net Alarm

{

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(0,2);

for(int i=3;i<=21;i++)

display.print(SM\_ARU\_0\_DA[i]);// Date & Time

display.setCursor(0,12);

Serial.println(SM\_ARU\_4\_NE[3]); if(SM\_ARU\_4\_NE[3]=='C')display.print("Internet Connected");// Internet if(SM\_ARU\_4\_NE[3]=='D')display.print("Internet Disconnected");// Internet

display.setCursor(0,25); display.print("Ch= ");

for(int i=3;i<=4;i++)

display.print(SM\_ARU\_1\_BT[i]);// Battery display.print("%");

display.setCursor(50,25); display.print(" Alarm= ");

for(int i=3;i<=7;i++)

display.print(SM\_ARU\_5\_AL[i]);// Alarm

display.display();

}

void Display\_3() // Call

{

String SM\_ARU\_3\_Call="";

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(0,0); display.println("Call");

for(int i=15;i<=24;i++)

SM\_ARU\_3\_Call+=(SM\_ARU\_3\_CA[i]);

display.print(SM\_ARU\_3\_Call);

display.setCursor(0,20);

if(SM\_ARU\_3\_Call=="9834049482")display.println("Zero@3327");

display.display(); delay(1000);

}

void Display\_2() //Message

{

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

display.setCursor(0,0); //display.println("Message "); for(int i=12;SM\_ARU\_2\_ME[i]!='#';i++)

display.print(SM\_ARU\_2\_ME[i]);

display.display(); delay(1000);

}

////////////////////////////////////////////////////////////////////////////////////// void Display\_Update\_Time()

{

display.setCursor(0,2);

for(int i=3;i<=5;i++)

display.print(SM\_ARU\_0\_DA[i]);// Date & Time

display.print(" ");

for(int i=16;i<=17;i++)

display.print(SM\_ARU\_0\_DA[i]);//Day

display.print('-'); for(int i=13;i<=14;i++)

display.print(SM\_ARU\_0\_DA[i]);//Month

display.print('-'); for(int i=10;i<=11;i++)

display.print(SM\_ARU\_0\_DA[i]);//year

display.print(" "); for(int i=19;i<=20;i++)

display.print(SM\_ARU\_0\_DA[i]);//HH

display.print(':'); for(int i=22;i<=23;i++)

display.print(SM\_ARU\_0\_DA[i]);//MM

a=SM\_ARU\_0\_DA[19]-0x30; b=SM\_ARU\_0\_DA[20]-0x30; c=SM\_ARU\_0\_DA[22]-0x30; d=SM\_ARU\_0\_DA[23]-0x30;

current\_time=((a\*1000)+(b\*100)+(c\*10)+d);

Serial.print("current\_time");

Serial.println(current\_time);

}

void Display\_Update\_Internet()

{

display.setCursor(115,2);

if(SM\_ARU\_4\_NE[3]=='C')display.print("E");// Internet if(SM\_ARU\_4\_NE[3]=='D')display.print("-");// Internet

}

void Display\_Update\_Battery()

{

display.setCursor(0,25); display.print("Ch= ");

for(int i=3;i<=4;i++)

display.print(SM\_ARU\_1\_BT[i]);// Battery display.print("%");

}

void Display\_Update\_Alarm()

{

display.setCursor(50,25); display.print(" Alarm= ");

for(int i=3;i<=7;i++)

display.print(SM\_ARU\_5\_AL[i]);// Alarm

a=SM\_ARU\_5\_AL[3]-0x30; b=SM\_ARU\_5\_AL[4]-0x30; c=SM\_ARU\_5\_AL[6]-0x30; d=SM\_ARU\_5\_AL[7]-0x30;

alarm\_time=((a\*1000)+(b\*100)+(c\*10)+d);

Serial.print("alarm\_time"); Serial.println(alarm\_time); if(current\_time==alarm\_time)

{

Serial.println("Alarm Match"); display.setCursor(0,15); display.print("R= "); for(int i=15;SM\_ARU\_5\_AL[i]!='#';i++) //alarm message display.print(SM\_ARU\_5\_AL[i]);

digitalWrite(Buzzer, 0); delay(100); digitalWrite(Buzzer, 1);

}

display.display();

}

Android Application

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="infoaryan.in.hc05\_bluetooth">    <uses-permission android:name="android.permission.BLUETOOTH" />  <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" /> <uses-permission android:name="android.permission.READ\_SMS" /> <!-- Runtime -->  <uses-permission android:name="android.permission.READ\_CALL\_LOG" /> <!- Runtime -->  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />    <application  android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/AppTheme">  <activity  android:name=".SettingActivity" android:exported="false" />  <activity android:name=".LedControl" />  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>    </manifest> |

|  |
| --- |
| <vector xmlns:android="http://schemas.android.com/apk/res/android" xmlns:aapt="http://schemas.android.com/aapt" android:width="108dp" android:height="108dp" android:viewportWidth="108" android:viewportHeight="108">  <path  android:fillType="evenOdd"  android:pathData="M32,64C32,64 38.39,52.99 44.13,50.95C51.37,48.37 70.14,49.57 70.14,49.57L108.26,87.69L108,109.01L75.97,107.97L32,64Z" android:strokeWidth="1"  android:strokeColor="#00000000"> <aapt:attr name="android:fillColor">  <gradient  android:endX="78.5885" android:endY="90.9159" android:startX="48.7653" android:startY="61.0927" android:type="linear">  <item |
| android:color="#44000000" android:offset="0.0" />  <item  android:color="#00000000" android:offset="1.0" />  </gradient>  </aapt:attr>  </path> <path  android:fillColor="#FFFFFF" android:fillType="nonZero"  android:pathData="M66.94,46.02L66.94,46.02C72.44,50.07 76,56.61  76,64L32,64C32,56.61 35.56,50.11 40.98,46.06L36.18,41.19C35.45,40.45 35.45,39.3 36.18,38.56C36.91,37.81 38.05,37.81  38.78,38.56L44.25,44.05C47.18,42.57 50.48,41.71 54,41.71C57.48,41.71  60.78,42.57 63.68,44.05L69.11,38.56C69.84,37.81 70.98,37.81  71.71,38.56C72.44,39.3 72.44,40.45  71.71,41.19L66.94,46.02ZM62.94,56.92C64.08,56.92 65,56.01 65,54.88C65,53.76  64.08,52.85 62.94,52.85C61.8,52.85 60.88,53.76 60.88,54.88C60.88,56.01  61.8,56.92 62.94,56.92ZM45.06,56.92C46.2,56.92 47.13,56.01  47.13,54.88C47.13,53.76 46.2,52.85 45.06,52.85C43.92,52.85 43,53.76 43,54.88C43,56.01 43.92,56.92 45.06,56.92Z" android:strokeWidth="1"  android:strokeColor="#00000000" /> </vector> |

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="infoaryan.in.hc05\_bluetooth">    <uses-permission android:name="android.permission.BLUETOOTH" />  <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" /> <uses-permission android:name="android.permission.READ\_SMS" /> <!-- Runtime -->  <uses-permission android:name="android.permission.READ\_CALL\_LOG" /> <!- Runtime -->  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"  />    <application  android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/AppTheme">  <activity  android:name=".SettingActivity" android:exported="false" />  <activity android:name=".LedControl" />  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application> |

</manifest>

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="infoaryan.in.hc05\_bluetooth">    <uses-permission android:name="android.permission.BLUETOOTH" />  <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" /> <uses-permission android:name="android.permission.READ\_SMS" /> <!-- Runtime -->  <uses-permission android:name="android.permission.READ\_CALL\_LOG" /> <!- Runtime -->  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />    <application  android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/AppTheme">  <activity  android:name=".SettingActivity" android:exported="false" />  <activity android:name=".LedControl" />  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />  <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>    </manifest> |