**1) COLOR RECOGNITION:**

Color REcognition Sensor code  
  
/\*  
// TCS230 color recognition sensor  
// Sensor connection pins to Arduino are shown in comments  
  
Color Sensor      Arduino  
-----------      --------  
 VCC               5V  
 GND               GND  
 s0                8  
 s1                9  
 s2                12  
 s3                11  
 OUT               10  
 OE                GND  
\*/  
const int s0 = 8;    
const int s1 = 9;    
const int s2 = 12;    
const int s3 = 11;    
const int out = 10;    
// LED pins connected to Arduino  
int redLed = 2;    
int greenLed = 3;    
int blueLed = 4;  
// Variables    
int red = 0;    
int green = 0;    
int blue = 0;    
     
void setup()    
{    
  Serial.begin(9600);  
  pinMode(s0, OUTPUT);    
  pinMode(s1, OUTPUT);    
  pinMode(s2, OUTPUT);    
  pinMode(s3, OUTPUT);    
  pinMode(out, INPUT);    
  pinMode(redLed, OUTPUT);    
  pinMode(greenLed, OUTPUT);    
  pinMode(blueLed, OUTPUT);    
  digitalWrite(s0, HIGH);    
  digitalWrite(s1, HIGH);    
}    
     
void loop()  
{    
  color();  
  Serial.print("R Intensity:");    
  Serial.print(red, DEC);    
  Serial.print(" G Intensity: ");    
  Serial.print(green, DEC);    
  Serial.print(" B Intensity : ");    
  Serial.print(blue, DEC);    
  //Serial.println();    
  
  if (red < blue && red < green )  
  {    
   Serial.println(" - (Red Color)");    
   digitalWrite(redLed, HIGH); // Turn RED LED ON  
   digitalWrite(greenLed, LOW);    
   digitalWrite(blueLed, LOW);    
  }    
  
  else if (blue < red && blue < green)    
  {    
   Serial.println(" - (Blue Color)");    
   digitalWrite(redLed, LOW);    
   digitalWrite(greenLed, LOW);    
   digitalWrite(blueLed, HIGH); // Turn BLUE LED ON    
  }    
  
  else if (green < red && green < blue)    
  {    
   Serial.println(" - (Green Color)");    
   digitalWrite(redLed, LOW);    
   digitalWrite(greenLed, HIGH); // Turn GREEN LED ON  
   digitalWrite(blueLed, LOW);    
  }    
  else{  
  Serial.println();    
  }  
  delay(300);    
  digitalWrite(redLed, LOW);    
  digitalWrite(greenLed, LOW);    
  digitalWrite(blueLed, LOW);    
 }    
     
void color()    
{      
  digitalWrite(s2, LOW);    
  digitalWrite(s3, LOW);    
  //count OUT, pRed, RED    
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);    
  digitalWrite(s3, HIGH);    
  //count OUT, pBLUE, BLUE    
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);    
  digitalWrite(s2, HIGH);    
  //count OUT, pGreen, GREEN    
  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);    
}

**2) BLUETOOTH**

BT Controlled mobile:

//Tx of Bluetooth to 0(Rx)

//Rx of Blueooth to 1(Tx)  
  
#define ledPin 13  
int state = 0;  
void setup() {  
  pinMode(ledPin, OUTPUT);  
  digitalWrite(ledPin, LOW);  
  Serial.begin(9600); // Default communication rate of the Bluetooth module  
}  
void loop() {  
  if(Serial.available() > 0){ // Checks whether data is comming from the serial port  
    state = Serial.read(); // Reads the data from the serial port  
    Serial.print(state);  
 }  
 if (state == '0') {  
  digitalWrite(ledPin, LOW); // Turn LED OFF  
  Serial.println("LED: OFF"); // Send back, to the phone, the String "LED: ON"  
  state = 0;  
 }  
 else if (state == '1') {  
  digitalWrite(ledPin, HIGH);  
  Serial.println("LED: ON");;  
  state = 0;  
 }  
}

BT in command mode  
  
  
  
  
//connect tx=7, rx=8  
  
  
#include <SoftwareSerial.h>  
  
SoftwareSerial BTSerial(7, 8); // RX-CONNECT TXD | TX  
  
void setup()  
{  
  
  Serial.begin(9600);  
  Serial.println("Enter AT commands:");  
  BTSerial.begin(38400);  // HC-05 default speed in AT command more  
}  
  
void loop()  
{  
  
  // Keep reading from HC-05 and send to Arduino Serial Monitor  
  if (BTSerial.available())  
    Serial.write(BTSerial.read());  
  
  // Keep reading from Arduino Serial Monitor and send to HC-05  
  if (Serial.available())  
    BTSerial.write(Serial.read());  
}  
//0021:07:0016B7

**3) RFID:**

**Connection:**

**5V-Arduino 5V**

**GND-Arduino GND**

**Tx-pin 9**

**Code:**

#include<SoftwareSerial.h>

SoftwareSerial mySerial(9, 10);

int count = 0; // count = 0

char input[12]; // character array of size 12

boolean flag = 0; // flag =0

void setup()

{

Serial.begin(9600); // begin serial port with baud rate 9600bps

mySerial.begin(9600);

}

void loop()

{

if(mySerial.available())

{

count = 0;

while(mySerial.available() && count < 12) // Read 12 characters and store them in

input array

{

input[count] =mySerial.read();

count++;

delay(5);

}

Serial.print(input); // Print RFID tag number

}

}

CODE2:

#include<SoftwareSerial.h>

SoftwareSerial mySerial(9, 10);

#define LEDPIN 12

char tag[] ="3C0087D597F9"; // Replace with your own Tag ID

char input[12]; // A variable to store the Tag ID being presented

int count = 0; // A counter variable to navigate through the input[]

character array

boolean flag = 0; // A variable to store the Tag match status

void setup()

{

Serial.begin(9600); // Initialise Serial Communication with the Serial Monitor

mySerial.begin(9600);

pinMode(LEDPIN,OUTPUT); //WRONG TAG INDICATOR

}

void loop()

{

if(mySerial.available())// Check if there is incoming data in the RFID Reader Serial

Buffer.

{

count = 0; // Reset the counter to zero

/\* Keep reading Byte by Byte from the Buffer till the RFID Reader Buffer is

empty

or till 12 Bytes (the ID size of our Tag) is read \*/

while(mySerial.available() && count < 12)

{

input[count] = mySerial.read();

// Read 1 Byte of data and store it in the input[] variable

Serial.write(input[count]);

count++; // increment counter

delay(5);

}

/\* When the counter reaches 12 (the size of the ID) we stop and compare each

value

of the input[] to the corresponding stored value \*/

if(count == 12) //

{

count =0; // reset counter varibale to 0

flag = 1;

/\* Iterate through each value and compare till either the 12 values are

all matching or till the first mistmatch occurs \*/

while(count<12 && flag !=0)

{

if(input[count]==tag[count])

flag = 1; // everytime the values match, we set the flag variable

to 1

else

flag= 0;

/\* if the ID values don't match, set flag variable to 0 and

stop comparing by exiting the while loop \*/

count++; // increment i

}

}

if(flag == 1) // If flag variable is 1, then it means the tags match

{

Serial.println("Access Allowed!");

digitalWrite(LEDPIN,HIGH);

delay (2000);

digitalWrite (LEDPIN,LOW);

}

else

{

Serial.println("Access Denied"); // Incorrect Tag Message

digitalWrite(LEDPIN,LOW);

delay(2000);

}

/\* Fill the input variable array with a fixed value 'F' to overwrite

all values getting it empty for the next read cycle \*/

for(count=0; count<12; count++)

{

input[count]= 'F';

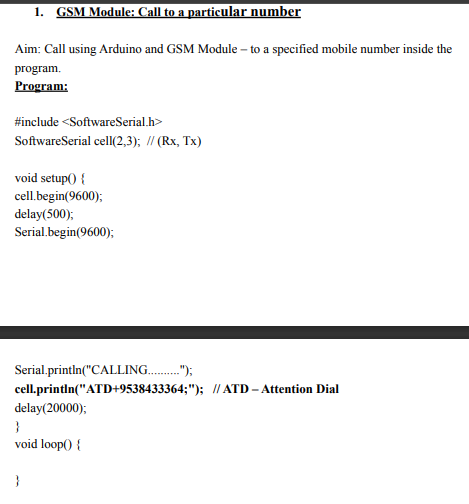
}

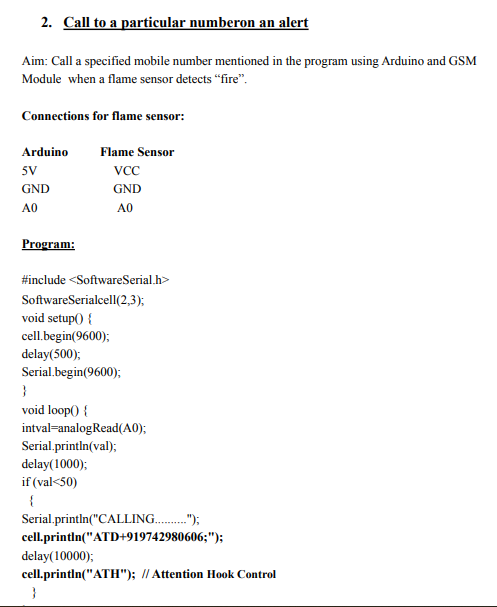
count = 0; // Reset counter variable

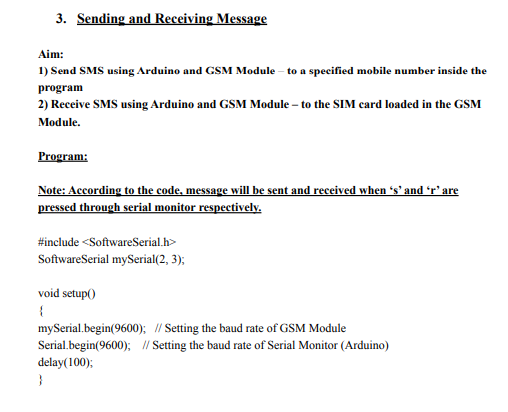
}

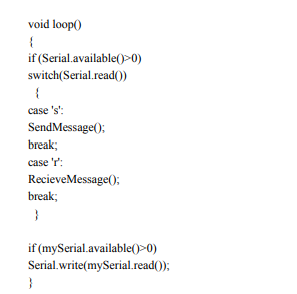
}

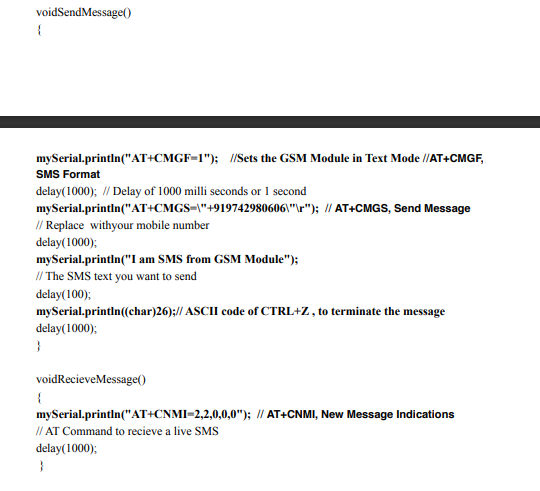
**4)GSM**

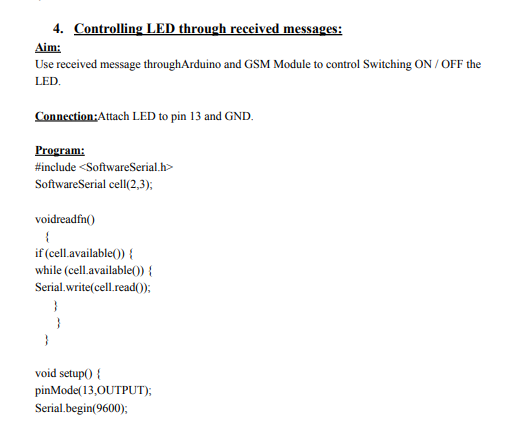
****

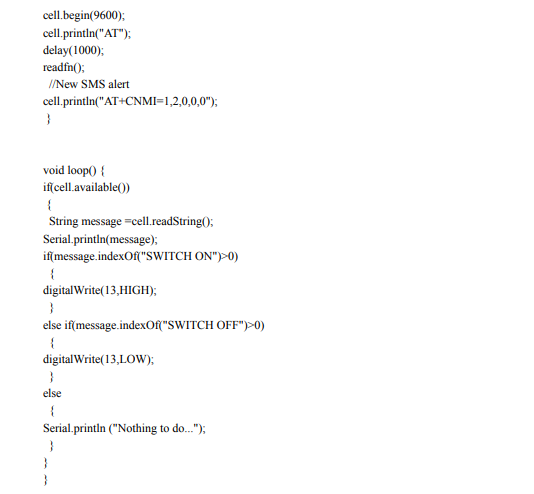
****

****

****

****

****

****