```
#define BLYNK_PRINT Serial              // blynk
#include <ESP8266_Lib.h>                // blynk
#include <BlynkSimpleShieldEsp8266.h>   // blynk
#define EspSerial Serial2               // blynk
#define ESP8266_BAUD 115200             // blynk
ESP8266 wifi(&EspSerial);               // blynk
const char WIFI_AUTH[] = "e25c2bec146c4ffb842df4216e8ea372";
const char WIFI_SSID[] = "RadekV";
const char WIFI_PASSWORD[] = "Hotspot76";
BlynkTimer timer;
WidgetLCD lcdPhone(V15);
int bR=0;                       //variables to control blynk buttons
int bG=0;
int bB=0;
int bClear=0;
int bFinish=0;

#include <SPI.h>                        // WIFI
#include "RF24.h"                       // WIFI
#include "U8glib.h"                             // LCD library
U8GLIB_SSD1306_128X64 myOled(U8G_I2C_OPT_NONE); // LCD
unsigned long time;                             // variable for timer
const int b1Pin = 4;   //button pins
const int b2Pin = 5;
const int b3Pin = 6;
const int rPin = 13;    //light pins
const int oPin = 8;
const int gPin = 9;
bool b1;              //variables for reading button states
bool b2;
bool b3;
```

```
unsigned long stoptime;     // variable for timer
int S=1;     // variable for automaton and its initial state 1
int val=0;   // variable for testing wifi transfer
int cR=0;    //count of recieved messages
int cT=0;    //count of transmited messages
int pwd=156;

RF24 myRadio(49,53);                    //WIFI  pins
byte addresses[][6] = {"0"};            //WIFI
struct package {
  char text[20]="command";
  int pwd;
  int tx;
  int rx;
  int Int1;
  int Int2;
  float Float1;
  int Array1[5]={0,0,0,0,0};
};
typedef struct package Package;  // WIFI
Package dataRecieve;
Package dataTransmit;


void setup()
{ Serial.begin(9600);
  pinMode(b1Pin, INPUT_PULLUP);    //config of buttons
  pinMode(b2Pin, INPUT_PULLUP);
  pinMode(b3Pin, INPUT_PULLUP);
  pinMode(rPin, OUTPUT);           //config of lights
  pinMode(oPin, OUTPUT);
```

```
  pinMode(gPin, OUTPUT);


  myRadio.begin();                    //WIFI
  myRadio.setChannel(0x61);
  myRadio.setPALevel(RF24_PA_MIN);
  //myRadio.setDataRate( RF24_250KBPS );
  myRadio.openReadingPipe(1, addresses[0]); //automaton-action listen after RESET
  myRadio.startListening();
  myOled.firstPage();                 //LCD
  myOled.setFont(u8g_font_unifont);
  myOledUpdate();

  EspSerial.begin(ESP8266_BAUD);      //blynk
  delay(1000);
  Blynk.begin(WIFI_AUTH, wifi, WIFI_SSID, WIFI_PASSWORD);
  //timer.setInterval(1000L, sendSensor); //timer of blynk


}

void loop()
{ Blynk.run();    //timer.run();   //timer of blynk
Serial.println(bR);


  if (S==1){ while (myRadio.available()) {myRadio.read(&dataRecieve, sizeof(dataRecieve));}
          if (dataRecieve.pwd==pwd) {val=dataRecieve.val;
                                     cR=cR+1;
                                     myOledUpdate();
                                    }
```

```
        b1=!digitalRead(b1Pin);
        b2=!digitalRead(b2Pin);
        b3=!digitalRead(b3Pin);
        if (b1){digitalWrite(rPin,1);
                cT=cT+1;
                val=val+2;
                sendWifi();
                S=2;
                myOledUpdate();
               }
        if (b2){digitalWrite(oPin,1);
                S=3;
                myOledUpdate();
               }
        if (b3){digitalWrite(gPin,1);
                S=4;
                myOledUpdate();
               }
      }
if (S==2){ b1=!digitalRead(b1Pin);
        if (!b1){digitalWrite(rPin,0);
                 S=1;
                 myOledUpdate();
                }
      }
if (S==3){ b2=!digitalRead(b2Pin);
        if (!b2){digitalWrite(oPin,0);
                 S=1;
                 myOledUpdate();
                }
      }
```

```
  if (S==4){ b3=!digitalRead(b3Pin);
             if (!b3){digitalWrite(gPin,0);
                      S=1;
                      myOledUpdate();
                     }
           }
}

void myOledUpdate(void)
{ myOled.firstPage();                //LCD
  myOled.setFont(u8g_font_unifont);
 do {myOled.setPrintPos(0,12);
      myOled.print("Control Unit");
      myOled.setPrintPos(0,27);
      myOled.print("Hello Bala!");
      myOled.setPrintPos(0,42);
      myOled.print("val");
      myOled.print(val);
      myOled.print(" cR");
      myOled.print(cR);
      myOled.print(" cT");
      myOled.print(cT);
      myOled.setPrintPos(0,57);
      myOled.print("S");
      myOled.print(S);
      myOled.print(" ");
      myOled.print(bR);
      myOled.print(bG);
      myOled.print(bB);
      myOled.print(" ");
```

```
    } while(myOled.nextPage());
}

void readCodeFromWifi(void) {         // listens wifi and puts recieved code to dataRecieve
  while (myRadio.available()) {
    myRadio.read(&dataRecieve, sizeof(dataRecieve));
  }
}

void sendWifi(void) {
  myRadio.stopListening();            //WIFI NRF TRANSMIT
  dataTransmit.pwd=pwd;
  dataTransmit.tx=1;
  dataTransmit.rx=1;
  dataTransmit.val=val;
  myRadio.openWritingPipe(addresses[0]);
  myRadio.write(&dataTransmit, sizeof(dataTransmit));
  myRadio.openReadingPipe(1, addresses[0]);
  myRadio.startListening();
}
//    sync with Blynk
BLYNK_WRITE(V0) {
  bR = param.asInt();
}
BLYNK_WRITE(V1) {
  bG = param.asInt();
}
BLYNK_WRITE(V2) {
  bB = param.asInt();
}
BLYNK_WRITE(V3) {
```

```
  bClear = param.asInt();
}
BLYNK_WRITE(V4) {
  bFinish  = param.asInt();
}
```