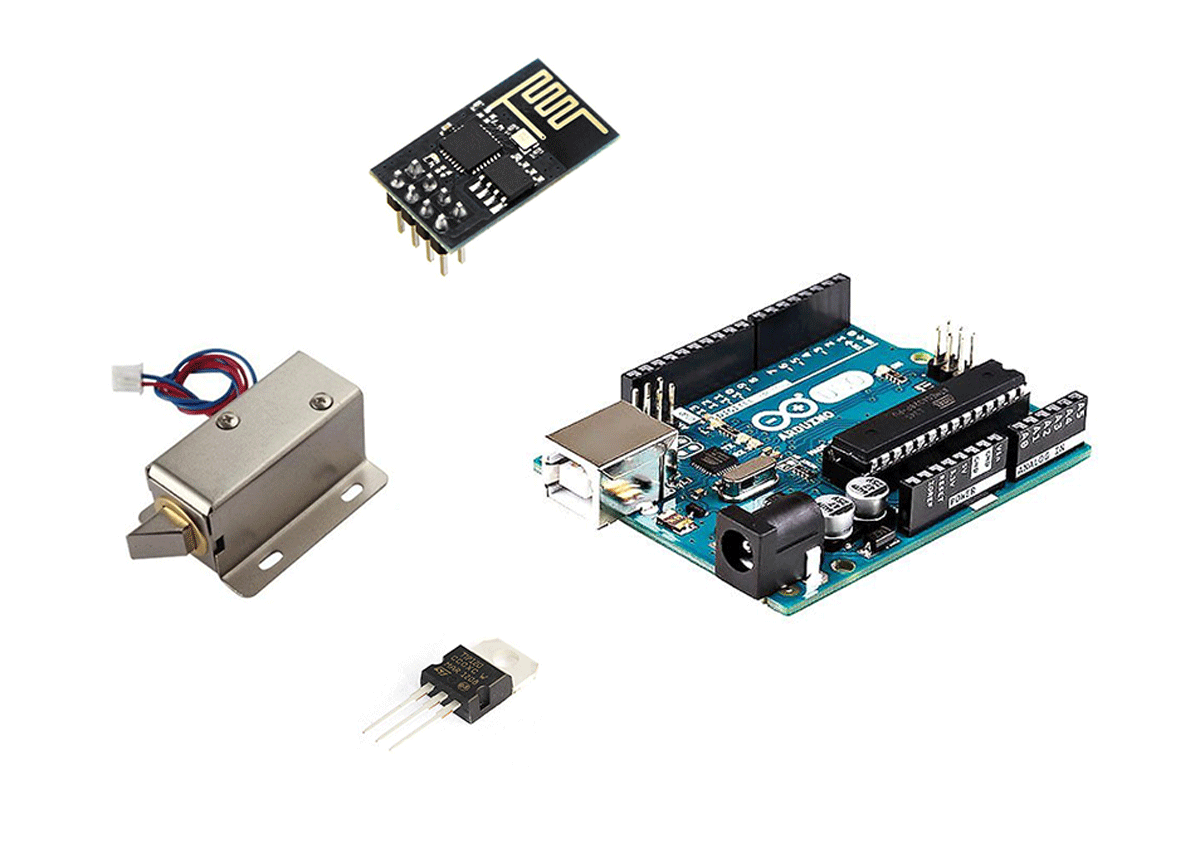
Door locks Using arduino

Component name :



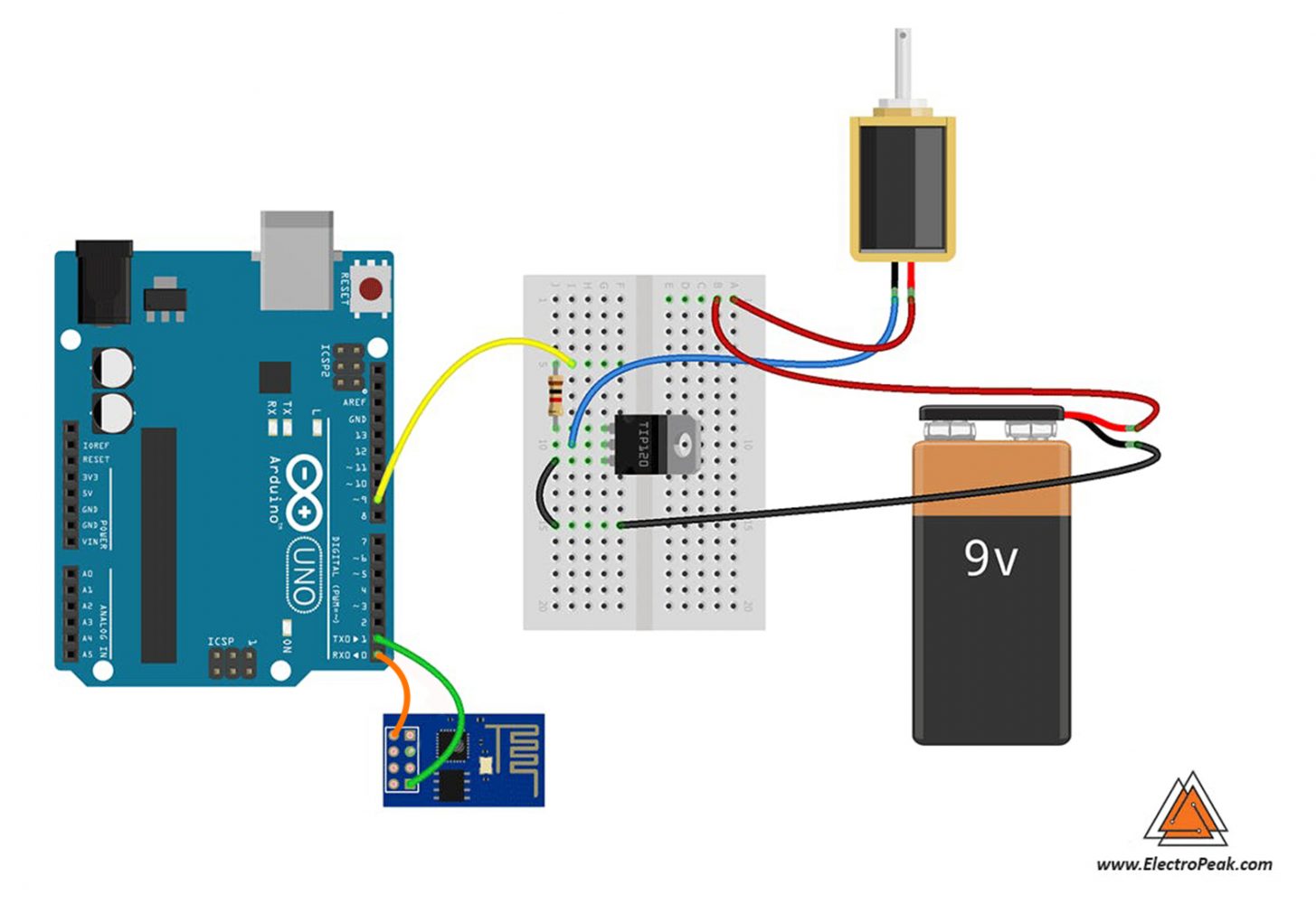
Arduino UNO R3

ESP8266 WiFi Module with PCB Antenna

TIP120 Power Darlington Transistors

Lock-style Solenoid

Daigram component:



Component from :

TIP120 Power Darlington Transistors

<https://store.fut-electronics.com/search?type=product&q=tip120>

Lock-style Solenoid

<https://egypt.souq.com/eg-en/dc-12v-cabinet-door-lock-electriclock-assembly-solenoid-code-ab102-39117756/i/?gclid=Cj0KCQjwjoH0BRD6ARIsAEWO9Du5jZQ4u-9UdNr-NDoPuYKeHzwVZzC5mq9Ca0lspWUwscl5k9KwSsUaAsgaEALw_wcB>

Code

First we write a code for ESP-01 to make a log-in page and receive password from user then send it to Arduino by the Serial port. Then we write another code for Arduino to get data from ESP-01 and control the servo motor. You can use Arduino IDE to compile both codes and Upload them to boards.

You must add the library and then upload the code. If it is the first time you run an Arduino board, don’t worry. Just follow these steps:

* Go to www.arduino.cc/en/Main/Software and download the software of your OS. Install the IDE software as instructed.
* Run the Arduino IDE and clear the text editor and copy the following code in the text editor.
* Choose the board in tools and boards, select your Arduino Board.
* Connect the Arduino to your PC and set the COM port in tools and port.
* Press the Upload (Arrow sign) button.
* You are all set!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | String inputString = "";         // a String to hold incoming data  bool stringComplete = false;  // whether the string is complete    void setup() {    // initialize serial:    Serial.begin(115200);    // reserve 200 bytes for the inputString:    inputString.reserve(200);      pinMode(9,OUTPUT);  }    void loop() {    // print the string when a newline arrives:    if (stringComplete) {      if (inputString=="your\_password")     {         digitalWrite(9,HIGH);         delay(300);         digitalWrite(9,LOW);           Serial.println(inputString);         // clear the string:         inputString = "";        stringComplete = false;      }    }  }      void serialEvent() {    while (Serial.available()) {      // get the new byte:      char inChar = (char)Serial.read();      // add it to the inputString:      inputString += inChar;      // if the incoming character is a newline, set a flag so the main loop can      // do something about it:      if (inChar == '\n') {        stringComplete = true;      }    }  }  Now its time to upload the ESP-01 code. You should use the Arduino IDE to upload the sketch to ESP. Before uploading the code, you should select ESP board for IDE.  Go to File>Preferences and add http://arduino.esp8266.com/stable/package\_esp8266com\_index.json in the additional boards. Then download and install it. Now you can see the ESP boards in Tools>Board. Select “Generic ESP8266 Module” and copy the code in a new sketch. Then you should set USB to TTL Converter as Uploader hardware. Just plug the converter in and set the right port in Tools>Port. It’s ready to Upload. |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76 | #include <ESP8266WiFi.h>  #include <ESP8266WebServer.h>  // WiFi network  const char\* ssid     = "YourSSID";  const char\* password = "YourPASSWORD";  ESP8266WebServer server ( 80 );  char htmlResponse[3000];  void handleRoot() {  snprintf ( htmlResponse, 3000,  "<!DOCTYPE html>\  <html lang=\"en\">\  <head>\  <style>\  body {background-color: rgb(160, 0, 53);}\  h3   {color: white;text-align:center;}\  p    {color: white; text-align:center;}\  div  {color: white; text-align:center;}\  ID {text-align:center;}\  input {text-align:center;}\  </style>\     <meta charset=\"utf-8\">\     <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\  </head>\  <body>\           <h3>\<canter>Electropeak Smart Security Door</canter>\</h3>\           <p>\<canter>Please type your ID</canter>\</p>\           <div>ID: <input type='text' name='pass\_word' id='pass\_word' align='center' size=10 autofocus></div> \           <div>\           <br><button id=\"save\_button\">Log In</button>\           </div>\     <script src=\"https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js\"></script>\     <script>\       var pass;\       $('#save\_button').click(function(e){\         e.preventDefault();\         pass = $('#pass\_word').val();\         $.get('/save?pass=' + pass, function(data){\           console.log(data);\         });\       });\     </script>\  </body>\  </html>");    server.send ( 200, "text/html", htmlResponse );  }  void handleSave() {  if (server.arg("pass")!= ""){     Serial.println(server.arg("pass"));  }  }  void setup() {  // Start serial  Serial.begin(115200);  delay(10);  // Connecting to a WiFi network  Serial.println();  Serial.println();  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {     delay(500);     Serial.print(".");  }  Serial.println("");  Serial.println("WiFi connected");  Serial.println("IP address: ");  Serial.println(WiFi.localIP());  server.on ( "/", handleRoot );  server.on ("/save", handleSave);  server.begin();  Serial.println ( "HTTP server started" );  }  void loop() {  server.handleClient();  } |

After uploading the code, Open Serial monitor of Arduino IDE and get your IP. Now if you type the IP in a browser URL bar, You can see UI of the web server. Type your ID in the text box and if you do it correctly, Arduino will activate the lock. Notice that you have to connect to a common wifi router with ESP8266.

Link

<https://electropeak.com/learn/smart-door-lock-w-wifi-login-page-by-arduino-esp8266/>