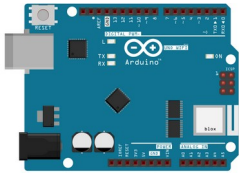
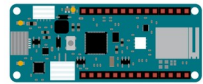


In einem Webformular können Lottozahlen ermittelt und angezeigt werden. Der Arduino „spielt“ für dich 6 Zahlen. Anschließend werden die Lottozahlen gezogen, aufsteigend sortiert und auf die Anzahl der übereinstimmenden Zahlen untersucht. Zusätzlich werden Datum und Zeit der Ziehung mit Hilfe des NTP-Protokolls angezeigt. NTP (Network Time Protocol) ist ein Protokoll zur **Synchronisierung von Uhren** in Computersystemen.

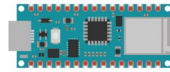
Für diese Anleitung benötigst du einen Arduino mit WiFi:



Arduino WiFi Rev 2



Arduino MKR WiFi 1010



Arduino Nano 33 IoT

So sieht es aus:

← → ↺ 🏠 🛡️ 192.168.1.160/WiFi.localIP() ☆

Lottozahlen mit WiFi ziehen (6 aus 49)

Montag, 13.05.2024 Uhrzeit: 08:04:04

neue Ziehung

Gespielte Zahlen: 8 11 14 16 33 36

Gezogene Zahlen: 11 22 23 33 35 39

2 Treffer: 11 33

Eigene IP: 192.168.1.12
IP Adresse Arduino: 192.168.1.160

Nach dem Start des Programms werden die IP und die aktuelle Zeit im Seriellen Monitor angezeigt. Weil die IP über DHCP vergeben wird, kann sie jedes Mal eine andere sein.

Ausgabe Serieller Monitor x

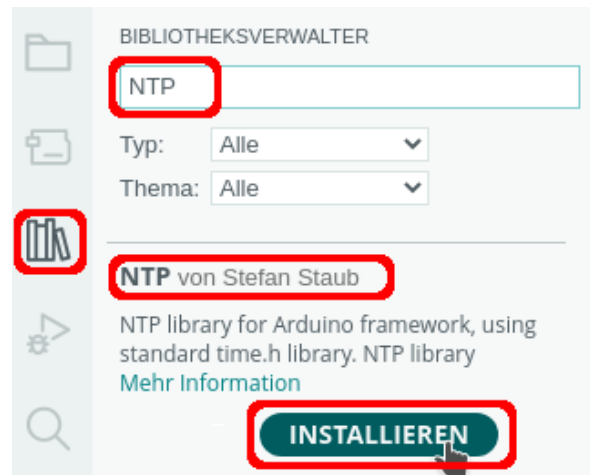
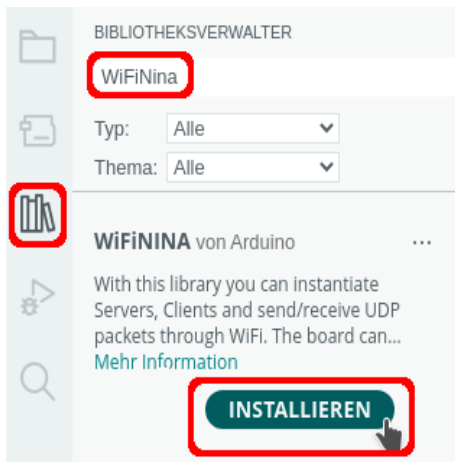
Nachricht (Enter um Nachricht für 'Arduino MKR WiFi 1010' zu senden) Sowohl NL als auch CR 9600 baud

```

Verbindung aufbauen mit FRITZ!Box 7590 LB
Server: FRITZ!Box 7590 LB
IP Adresse Arduino DHCP: 192.168.1.160
02.07.2023 Uhrzeit: 09:48:54

```

Benötigte Bibliotheken:



Binde die benötigte Bibliothek ein und definiere die Variablen.

```
# include "WiFinINA.h"
# include "NTP.h"

// Minimum/Maximum der Zufallszahlen
const byte Minimum = 1;
const byte Maximum = 49;

// Anzahl der zu ziehenden Zahlen
const byte Anzahl = 6;

// Array für die gezogenen Zahlen
int LottoZahl[Anzahl];

// Array für die angekreuzten Zahlen
int SpielZahl[Anzahl];

// SSID und Passwort des Routers
char Router[] = "FRITZ!Box 7590 LB";
char Passwort[] = "anea1246";

int Status = WL_IDLE_STATUS;
WiFiServer WiFiServer(80);
WiFiClient WiFiClient;

WiFiUDP wifiUdp;
NTP ntp(wifiUdp);
```

Der setup-Teil:

```
void setup()
{
  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial) { ; }

  // Verbindung aufbauen
  if (WiFi.status() == WL_NO_MODULE)
  {
    Serial.println(F("Verbindungsaufbau gescheitert!"));
  }

  Serial.print("Verbindung aufbauen mit ");
  Serial.println(Router);
  while (Status != WL_CONNECTED)
  {
    Status = WiFi.begin(Router, Passwort);

    // Zeit für den Verbindungsaufbau
    // wenn die Verbindung nicht zustandekommt -> Zeit vergrößern
    delay(500);
  }

  // Webserver starten
  WiFiServer.begin();

  // IP des Servers/des verbunden Computers anzeigen
  Serial.print("Server: ");
  Serial.println(WiFi.SSID());

  // IP des Arduinos anzeigen
  Serial.print("IP Adresse Arduino DHCP: ");
  Serial.println(WiFi.localIP());

  /*
  Zeitzone
  CEST: Central European Summertime
  Beginn europäische Sommerzeit letzter Sonntag im März 2 Uhr GMT + 2 Stunden
  */
  ntp.ruleDST("CEST", Last, Sun, Mar, 2, 120);

  // CET: Central European Time
  // Beginn Normalzeit letzter Sonntag im Oktober 3 Uhr GMT + 1 Stunde
  ntp.ruleSTD("CET", Last, Sun, Oct, 3, 60);

  // ntp starten
  ntp.begin();

  // Zeit holen
  ntp.update();
}
```

```
// Zeit mit formattedTime() anzeigen:  
// %d = Tag, %m = Monat, %Y = Jahr, %T = Zeit in Stunden, Minuten, Sekunden  
Serial.println(ntp.formattedTime("%d.%m.%Y Uhrzeit: %T"));  
  
// Zufallsgenerator mit dem Signal an A0 starten  
randomSeed(analogRead(A0));  
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()  
{  
  // Variable "leeren"/auf 0 setzen  
  String AnzahlTreffer = "";  
  String GespielteZahlen = "";  
  String GezogeneZahlen = "";  
  byte Treffer = 0;  
  
  // auf WiFiClienten warten ...  
  WiFiClient = WiFiServer.available();  
  if (WiFiClient)  
  {  
    // Seite aufbauen wenn SeiteAufbauen true ist  
    boolean SeiteAufbauen = true;  
  
    // solange der WiFiClient verbunden ist ...  
    while (WiFiClient.connected())  
    {  
      if (WiFiClient.available())  
      {  
        // Anforderung vom WiFiClienten lesen ...  
        char Zeichen = WiFiClient.read();  
  
        // wenn als Zeichen neue Zeile (\n) übergeben wird  
        // SeiteAufbauen true ist -> Seite anzeigen  
        if (Zeichen == '\n' && SeiteAufbauen)  
        {  
          ntp.update();  
  
          /*  
            HTML-Seite aufbauen  
            die folgenden Anweisungen müssen  
            mit print oder println gesendet werden  
            println "verschönert" den Quelltext  
            (erzeugt einen Zeilenumbruch im Quelltext)  
            " müssen mit \ maskiert werden " -> \  
          */  
          WiFiClient.println("HTTP/1.1 200 OK");  
          WiFiClient.println("Content-type:text/html");  
  
          // Leerzeile zwingend erforderlich  
          WiFiClient.println();  
          WiFiClient.println("<!doctype html>");  
          WiFiClient.println("<html>");  
          WiFiClient.println("<body>");  
        }  
      }  
    }  
  }  
}
```

```

WiFiClient.println("<h2>Lottozahlen mit WiFi ziehen (6 aus " + String(Maximum) + ")");
WiFiClient.print("</h2>");
WiFiClient.println("<hr />");
WiFiClient.println("<b>");

// formattedTime() zeigt Wochentage in englischer Sprache
// -> müssen einzeln abgefragt werden
switch (ntp.weekDay())
{
  case 0:
    WiFiClient.print("Sonntag");
    break;
  case 1:
    WiFiClient.print("Montag");
    break;
  case 2:
    WiFiClient.print("Dienstag");
    break;
  case 3:
    WiFiClient.print("Mittwoch");
    break;
  case 4:
    WiFiClient.print("Donnerstag");
    break;
  case 5:
    WiFiClient.print("Freitag");
    break;
  case 6:
    WiFiClient.print("Samstag");
    break;
}
WiFiClient.print(", ");
WiFiClient.print(ntp.formattedTime("%d.%m.%Y Uhrzeit: %T</b>"));

WiFiClient.println("<hr />");

WiFiClient.println("<table><tr>");

// Button neue Ziehung
WiFiClient.print("<td><input style='font-size:14pt;");
WiFiClient.print("font-weight:bold;");
WiFiClient.print(" background-color:#7eff78;");
WiFiClient.print(" width:200px; cursor:pointer;");
WiFiClient.print(" border-radius:5px;border: 2px solid black;");
WiFiClient.print(" type='button'");
WiFiClient.println(" onClick=\"location.href='WiFi.localIP()'\"");
WiFiClient.println(" value='neue Ziehung'>");
WiFiClient.println("</td></tr></table>");
WiFiClient.println("<hr />");

// Ziehung der "angekreuzten" Zahlen
Ziehung();
ZahlenSortieren(LottoZahl, 6);

```

```
// String GespielteZahlen aus den Elementen des Arrays LottoZahl
// zusammensetzen
for (int i = 0; i < Anzahl; i++)
{
    GespielteZahlen = GespielteZahlen + LottoZahl[i] + " ";
    SpielZahl[i] = LottoZahl[i];
}

// Ziehung der Lottozahlen
Ziehung();

// Zahlen sortieren
ZahlenSortieren(LottoZahl, 6);

// Vergleichen der Elemente der Arrays SpielZahl und LottoZahl
for (int i = 0; i < Anzahl; i++)
{
    for (int ii = 0; ii < Anzahl; ii++)
    {
        // Übereinstimmung gefunden -> Treffer um 1 erhöhen
        // AnzahlTreffer: String für übereinstimmende Zahlen erzeugen
        if (SpielZahl[i] == LottoZahl[ii])
        {
            AnzahlTreffer = AnzahlTreffer + SpielZahl[i] + " ";
            Treffer++;
        }
    }
}

// String GezogeneZahlen aus den Elementen des Arrays LottoZahl
// zusammensetzen
for (int i = 0; i < Anzahl; i++)
{
    GezogeneZahlen = GezogeneZahlen + LottoZahl[i] + " ";
}
WiFiClient.println("<h3 style='background-color:lightyellow'>");
WiFiClient.println("Gespielte Zahlen: ");
WiFiClient.println(GespielteZahlen);
WiFiClient.println("</h3>");

WiFiClient.println("<h3 style='background-color:lightcyan'>");

WiFiClient.println("Gezogene Zahlen: ");
WiFiClient.println(GezogeneZahlen);
WiFiClient.println("</h3>");

if (Treffer == 0) WiFiClient.println("<br><b>keine Treffer!</b>");

// getroffene Zahlen anzeigen
else
{
    WiFiClient.print("<br><b>" + String(Treffer));
    WiFiClient.print(" Treffer: </b>");
    WiFiClient.print(AnzahlTreffer);
}
```

```

        WiFiClient.println("<hr>");

        // IPs anzeigen
        WiFiClient.print("<b>Eigene IP: ");
        WiFiClient.print(WiFiClient.remoteIP());
        WiFiClient.print("</b>");
        WiFiClient.print("<br><b>IP Adresse Arduino: ");
        WiFiClient.print(WiFi.localIP());
        WiFiClient.print("</b>");
        WiFiClient.println("</body>");
        WiFiClient.println("</html>");

        // HTTP-Antwort endet mit neuer Zeile
        WiFiClient.println();

        // Seite vollständig geladen -> loop verlassen
        break;
    }
    // neue Zeile -> SeiteAufbauen auf true setzen
    if (Zeichen == '\n') SeiteAufbauen = true;

    // die aktuelle Zeile enthält ein Zeichen
    else if (Zeichen != '\r') SeiteAufbauen = false;
}
}
delay(1);

WiFiClient.stop();
}
}

```

Die Funktion ZahlenSortieren sortiert die Lottozahlen aufsteigend der Größe nach.

```

void ZahlenSortieren(int Zahlen[], int Groesse)
{
    // unsortierten Bereich des Arrays durchlaufen
    for (int i = 0; i < (Groesse - 1); i++)
    {
        // bei jedem Durchlauf wird das jeweils letzte Element weggelassen
        for (int ii = 0; ii < (Groesse - (i + 1)); ii++)
        {
            /*
                wenn die aktuelle Zahl größer als die nachfolgende Zahl ist
                -> aktuelle Zahl temporär speichern
                -> Zahlen vertauschen
                -> temporäre Zahl der nachfolgenden Zahl zuweisen
            */
            if (Zahlen[ii] > Zahlen[ii + 1])
            {
                int zwischengespeicherteZahl = Zahlen[ii];
                Zahlen[ii] = Zahlen[ii + 1];
                Zahlen[ii + 1] = zwischengespeicherteZahl;
            }
        }
    }
}
}

```

Die Funktion Ziehung sorgt für die zufällige Auswahl der Zahlen.

```
int Ziehung()
{
  for (int i = 0; i < Anzahl; i ++)
  {
    /*
     * die Lottozahl gilt solange als bereits gezogen
     * bis in der for-Schleife nachgewiesen wird, dass sie neu ist
     * und BereitsGezogen den Wert false hat
     */
    bool BereitsGezogen = true;

    while (BereitsGezogen)
    {
      // Zahl ziehen
      LottoZahl[i] = random(Minimum, Maximum);
      BereitsGezogen = false;
      /*
       * Zufallszahl mit den bereits gezogenen Zahlen vergleichen
       * i wird in der ersten for-Schleife hochgezählt
       * alle bisherigen Zahlen (ii) werden mit den bisher gezogenen
       * (außer der gerade gezogenen) verglichen (ii < i)
       * ist sie identisch, erhält BereitsGezogen den Wert true
       * es wird erneut eine Zufallszahl bestimmt
       * wenn die Zahl neu ist, (BereitsGezogen hat noch den Wert false)
       * wird die while-Schleife verlassen und die nächste Zahl gezogen
       */
      for (int ii = 0; ii < i; ii ++)
      {
        if (LottoZahl[i] == LottoZahl[ii]) BereitsGezogen = true;
      }
    }
  }

  /*
   * das Array mit den Zahlen wird an das Hauptprogramm zurückgegeben
   * beim ersten Aufruf sind es die gespielten Zahlen
   * beim zweiten Aufruf die gezogenen Lottozahlen
   */
  return LottoZahl[Anzahl];
}
```