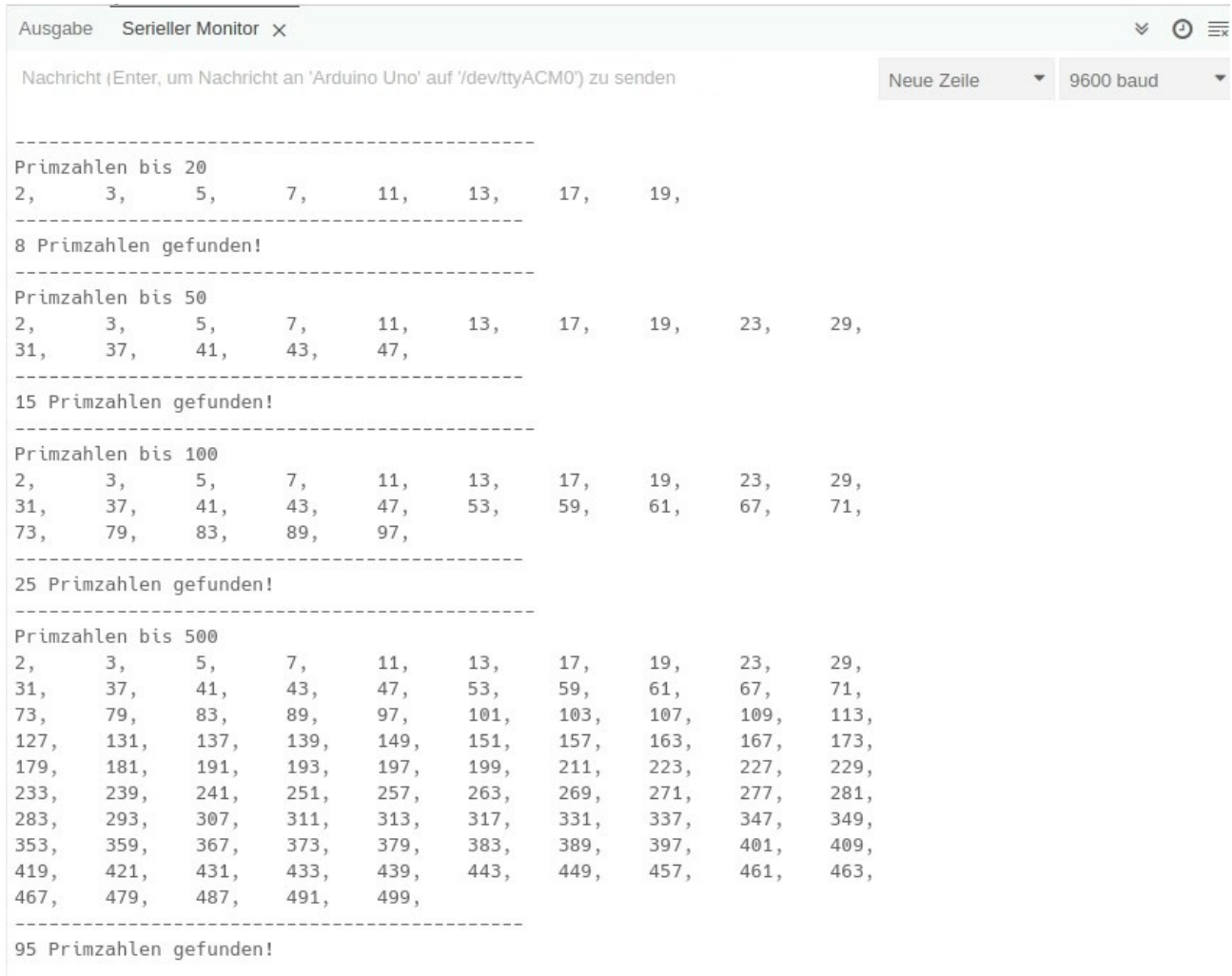


Das Programm soll die Primzahlen von 2 bis zu der eingegebenen Zahl ermitteln und im Seriellen Monitor anzeigen. Die zu untersuchende Zahl wird im Seriellen Monitor eingegeben. Außerdem wird eine Obergrenze für die eingegebene Zahl festgelegt. Die Variable int ist auf 32768 begrenzt und die Untersuchung größerer Zahlen benötigt sehr viel Rechenzeit.

So soll es aussehen:



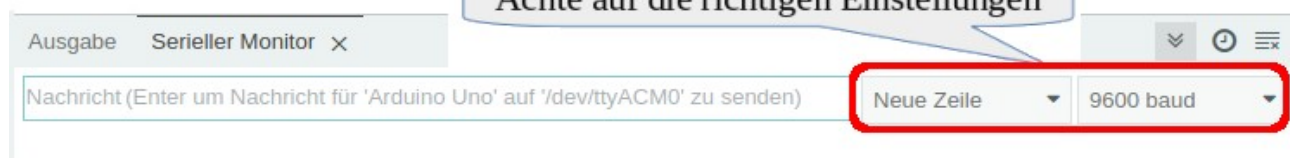
```

Ausgabe  Serieller Monitor x
Nachricht (Enter, um Nachricht an 'Arduino Uno' auf '/dev/ttyACM0' zu senden)  Neue Zeile  9600 baud

-----
Primzahlen bis 20
2, 3, 5, 7, 11, 13, 17, 19,
-----
8 Primzahlen gefunden!
-----
Primzahlen bis 50
2, 3, 5, 7, 11, 13, 17, 19, 23, 29,
31, 37, 41, 43, 47,
-----
15 Primzahlen gefunden!
-----
Primzahlen bis 100
2, 3, 5, 7, 11, 13, 17, 19, 23, 29,
31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
73, 79, 83, 89, 97,
-----
25 Primzahlen gefunden!
-----
Primzahlen bis 500
2, 3, 5, 7, 11, 13, 17, 19, 23, 29,
31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
73, 79, 83, 89, 97, 101, 103, 107, 109, 113,
127, 131, 137, 139, 149, 151, 157, 163, 167, 173,
179, 181, 191, 193, 197, 199, 211, 223, 227, 229,
233, 239, 241, 251, 257, 263, 269, 271, 277, 281,
283, 293, 307, 311, 313, 317, 331, 337, 347, 349,
353, 359, 367, 373, 379, 383, 389, 397, 401, 409,
419, 421, 431, 433, 439, 443, 449, 457, 461, 463,
467, 479, 487, 491, 499,
-----
95 Primzahlen gefunden!

```

Achte auf die richtigen Einstellungen



```

Ausgabe  Serieller Monitor x
Nachricht (Enter, um Nachricht für 'Arduino Uno' auf '/dev/ttyACM0' zu senden)  Neue Zeile  9600 baud

```

Für diese Anleitung brauchst du nur den Arduino.

Definiere die Variablen:

```
int Zahl;  
int EingabeMax = 1000;
```

Der setup-Teil startet den Seriellen Monitor und zeigt die Aufforderung an, eine Zahl einzugeben.

```
void setup()  
{  
  Serial.begin(9600);  
  Serial.println("Zahl eingeben:");  
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()  
{  
  String Eingabe;  
  
  while (Serial.available() > 0)  
  {  
    // solange lesen, bis return \n = return eingegeben wurde  
    Eingabe = Serial.readStringUntil("\n");  
  
    // das letzte Zeichen ist return = \n → soll entfernt werden  
    Eingabe = Eingabe.substring(0, Eingabe.length() - 1);  
  
    // eingegebene Eingabe zu int umwandeln  
    Zahl = Eingabe.toInt();  
  
    // nur ausführen wenn Zahl < 1000, ansonsten ist die Rechenzeit zu lang  
    if (Zahl < EingabeMax)  
    {  
      int ZaehlerZeile = 0;  
  
      Serial.println("-----");  
      Serial.println("Primzahlen bis " + String(Eingabe)) ;  
      for (int i = 2 ; i < Zahl ; i++)  
      {  
        // Primzahl suchen -> Funktion aufrufen  
        bool Suche = Primzahl(i);  
  
        // Primzahl gefunden -> Suche = true  
        if (Suche)  
        {  
          ZaehlerZeile++;  
  
          // 10 Eingaben pro Zeile  
          if (ZaehlerZeile % 10 == 0) Serial.println(String(i) + "," ) ;  
          else Serial.print(String(i) + "," + '\t') ;  
        }  
      }  
    }  
  }  
}
```

```
Serial.println("\n-----");
Serial.println(String(ZaehlerZeile) + " Primzahlen gefunden!");
}

// \u00df -> ß
else Serial.println("Die Eingabe ist zu gro\u00df!");
}
}
```

Der Funktion Primzahl wird die zu untersuchende Zahl übergeben. Weil Teiler, die größer als die Hälfte der Zahl sind, nicht möglich sind, wird auch nur bis zur Hälfte geprüft.

Die Teilbarkeit wird mit modulo ermittelt.

```
bool Primzahl(int Zahl)
{
    // nur bis zur Hälfte der Eingabe prüfen
    for (int i = 2 ; i <= Zahl / 2 ; i++)
    {
        // Teilbarkeit mit modulo testen
        // wenn kein Rest → Eingabe ist teilbar → false zurückgeben
        if (Zahl % i == 0) return false;
    }

    // keine Teiler gefunden → PrimEingabe → true zurückgeben
    return true;
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 15.05.23