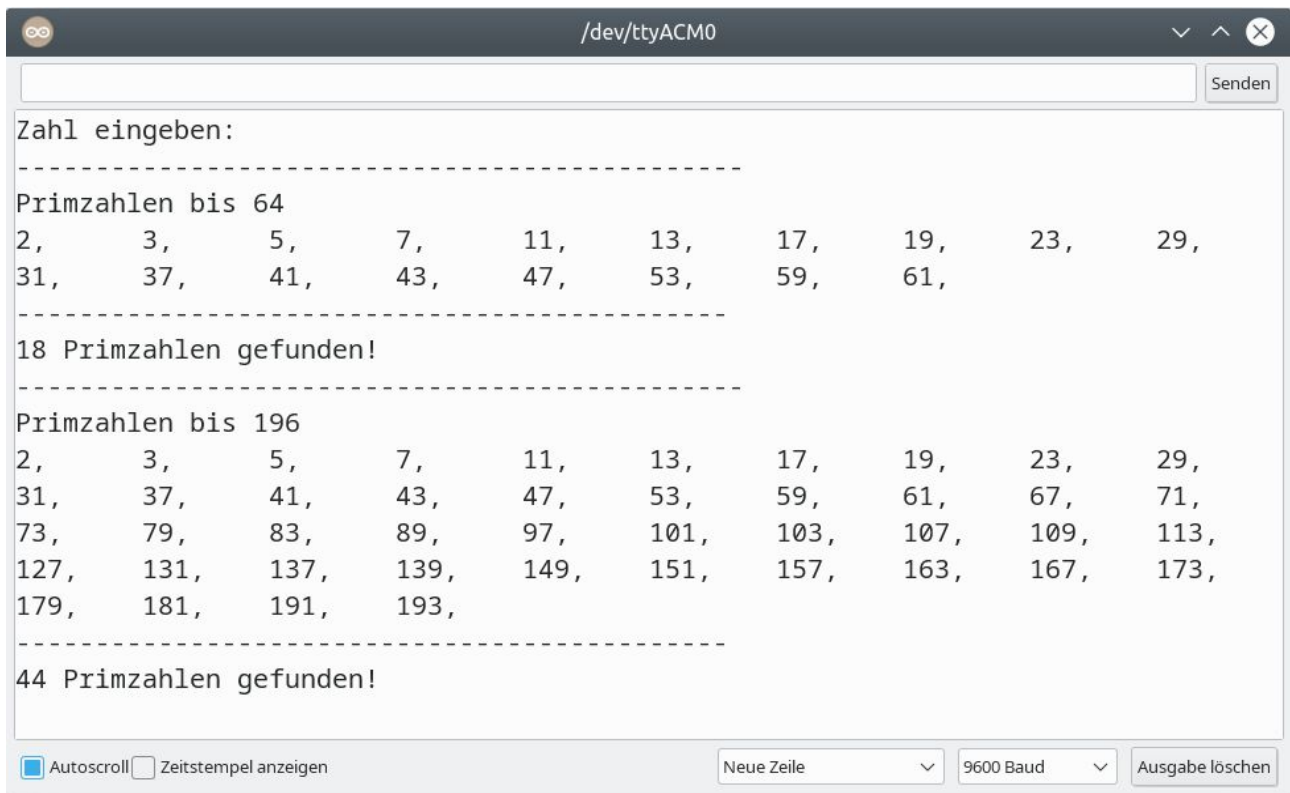


Primzahlen ermitteln

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:



The screenshot shows the Serial Monitor window for the device `/dev/ttyACM0`. It displays the output of a program that finds prime numbers. The first input is 64, resulting in 18 prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61. The second input is 196, resulting in 44 prime numbers: 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.

```

Zahl eingeben:
-----
Primzahlen bis 64
2,    3,    5,    7,    11,   13,   17,   19,   23,   29,
31,   37,   41,   43,   47,   53,   59,   61,
-----
18 Primzahlen gefunden!
-----
Primzahlen bis 196
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,
-----
44 Primzahlen gefunden!
  
```

Für diese Aufgabe 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;
}
```