

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

Artikel ini ditujukan untuk Anda yang ingin menghubungkan mikrokontroler Arduino dengan aplikasi desktop menggunakan Visual Basic (.NET), melalui komunikasi serial (serial port communication). Perangkat utama yang saya gunakan adalah papan Arduino Mega dan Visual Studio Express 2012. Kita akan membuat program di kedua sisi, yaitu antara mikrokontroler dan komputer akan saling berkomunikasi, jika CommandButton di aplikasi komputer kita klik, maka lampu pada Arduino akan menyala, demikian juga sebaliknya, jika salah satu port di Arduino kita tekan, maka akan muncul informasi di layar komputer.

### Pemrograman serial port pada Arduino

Hubungkan Arduino ke PC Anda menggunakan kabel USB, lalu buka software Arduino. Program ini untuk memastikan bahwa koneksi antara Arduino dan PC tidak ada kendala. Ketik program seperti di bawah ini lalu upload ke Arduino.

```
int inByte=0; // variable to receive data from serial port
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    inByte=Serial.read();
    Serial.write(inByte);
  }
}
```

Setelah itu buka **Tools > Serial Monitor** pada aplikasi Arduino, untuk melihat interaksi antara PC dan mikrokontroler. Jika kita mengirim sembarang data, maka Arduino akan membalas data yang sama. Jangan lupa periksa kecepatan komunikasi, keduanya harus sama. Dalam

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

hal ini 9600 bit per detik.

Pada bagian setup, kita mengaktifkan fungsi serial port pada Arduino (yang terkoneksi langsung dengan port USB) dengan perintah:

```
Serial.begin(9600);
```

Perintah ini hanya dilakukan sekali saja, pada saat Arduino diaktifkan atau di-reset. Jika kita ingin kecepatan lebih tinggi, ganti dengan 115000. Jika kita ingin menggunakan serial port yang lain, gunakan Serial1, Serial2 atau Serial3. (berlaku untuk mikro yang serial portnya lebih dari satu, contohnya Arduino Mega, terletak pada pin 14 sampai 19). Jika Anda menggunakan Uno, maka tidak ada pilihan lain.

Fungsi **Serial.available()** memberikan nilai BENAR jika ada data yang masuk. Sedangkan fungsi **Serial.read()** untuk membaca data yang masuk sebesar 1 byte. Perintah untuk mengirim data dari mikro ke luar adalah

**Serial.write**

```
(  
data_yg_ditulis  
);
```

Dengan melihat program di atas, maka mikrokontroler hanya akan mengirim ke luar jika ada masukan data serial, dan pengirimannya akan diulang terus sebanyak berapa byte data yang masuk.

### Mengendalikan LED lewat serial port

Sekarang kita akan coba menyalakan atau mematikan LED (pin 13), melalui serial port. Kita buat ketentuan yang sederhana sebagai berikut: jika menerima karakter 'A' maka lampu akan menyala, jika menerima karakter 'B' maka lampu akan mati.

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

```
// variable to receive data from serial port
int inByte=0;
// Pin 13 has an LED connected on most Arduino boards.
int led = 13;
void setup()
(
// initialize UART at 9600bps
Serial.begin(9600);
// initialize the digital pin as an output.
pinMode(led, OUTPUT);
}

void loop()
{
if (Serial.available() > 0) {
inByte=Serial.read();
Serial.write(inByte);

if (inByte==65)
{
digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
}
if (inByte==66)
{
digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
}

}
}
```

Ketik program di atas, upload dan jalankan Serial Monitor. Kirim karakter 'A' atau 'B' untuk menyalakan dan mematikan LED yang ada di papan Arduino. Catatan: tentu saja Anda bisa mengganti karakter tersebut dengan yang lain, sesuai keinginan. ('A' = ASCII nomor 65 dan 'B' = ASCII nomor 66).

## Membaca status input melalui serial port

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

Tahap berikutnya adalah pembacaan status input pada Arduino melalui serial port. Ketentuan yang kita buat adalah jika salah satu pin digital pada Arduino bernilai 0/low maka Arduino akan mengirim karakter 'L' dan jika bernilai 1/high maka akan mengirim karakter 'H'. Untuk memudahkan pengamatan, masing-masing karakter diikuti oleh karakter 'CR' atau 'n'.

Di contoh ini, kita gunakan pin nomor 24 yang dihubungkan ke sebuah tombol. Kaki satu lagi dihubungkan ke ground.

```
// variable to receive data from serial port
int inByte=0;

// Pin 13 has an LED connected on most Arduino boards.
int led = 13;

// Push button is connected to pin 24
int buttonPin = 24;
int buttonState=0;

void setup()
{
// initialize UART at 9600bps
Serial.begin(9600);
// initialize the digital pin as an output.
pinMode(led, OUTPUT);
pinMode(buttonPin, INPUT);
}
void loop()
{
if (Serial.available() > 0) {
inByte=Serial.read();
Serial.write(inByte);

if (inByte==65)
{
digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
}
if (inByte==66)
{
digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
}
}
buttonState=digitalRead(buttonPin);
```

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

```
if (buttonState==LOW)
{
Serial.write("Ln"); // send 'L' + CR
}
if (buttonState==HIGH)
{
Serial.write("Hn"); // send 'H' + CR
}
}
```

Ketik program di atas, lalu upload. Buka Serial Monitor, lalu amati data yang diterima. Cobalah juga menekan/melepas tombol sambil mengamati perubahan yang terjadi antara 'L' dan 'H'. Di percobaan ini kita bisa amati, bahwa Arduino mengirimkan data secara terus menerus baik L ataupun H, meskipun tombol ditekan atau dilepas lama. Ingat bahwa hal ini karena instruksi ini terletak di dalam program utama/main loop.

Sekarang kita akan coba modifikasi program, sehingga yang terkirim hanya satu kali saja saat terjadi perubahan dari L ke H atau H ke L. Untuk itu, kita harus membandingkan kondisi nilai tombol saat ini dengan kondisi pada loop sebelumnya. Variabel yang digunakan **lastButtonState** . Jika tidak

sama dengan

**buttonState**

, maka baru kita proses apa yang kita inginkan yaitu mengirimkan data.

```
// variable to receive data from serial port
int inByte=0;
```

```
// Pin 13 has an LED connected on most Arduino boards.
int led = 13;
```

```
// Push button is connected to pin 24
int buttonPin = 24;
int buttonState=0;
int lastButtonState=0;
```

```
void setup()
{
// initialize UART at 9600bps
Serial.begin(9600);
// initialize the digital pin as an output.
```

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

```
pinMode(led, OUTPUT);
pinMode(buttonPin, INPUT);
}

void loop()
{
  if (Serial.available() > 0)
  {
    inByte=Serial.read();
    Serial.write(inByte);

    if (inByte==65)
    {
      digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    }
    if (inByte==66)
    {
      digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    }
  }

  buttonState=digitalRead(buttonPin);
  if (buttonState != lastButtonState)
  {
    if (buttonState==LOW)
    {
      Serial.write("Ln");
    }
    else
    {
      Serial.write("Hn");
    }
  }
  lastButtonState=buttonState;
}
```

Ketik program di atas, lalu upload, kemudian amati di Serial Monitor. Tekan tombol yang agak lama sambil amati karakter yang terkirim. Arduino hanya akan mengirimkan satu kali saja saat ada perubahan **L**  $\square$  **H** atau **H**  $\square$  **L**.

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

Sampai di sini program di sisi Arduino sudah selesai, berikutnya kita masuk ke pemrograman serial port menggunakan Visual Studio .NET dalam hal ini kita coba memakai Visual Basic versi 2012.

---

```
(adsbygoogle = window.adsbygoogle || []).push({});
```

---

Seperti yang saya sampaikan sebelumnya, kita akan kendalikan lampu melalui Visual Basic application dan membaca status sebuah tombol pada Arduino.

Buka Visual Studio Express 2012, lalu buat proyek baru. Klik File > New Project. Pilih template Visual Basic > Windows Form Application, lalu beri nama SerialArduinoControl (atau terserah Anda). Tentukan lokasi folder lalu klik OK.

Anda dapat lihat sebuah form window kosong dalam modus design. Komponen standar yang kita gunakan adalah CommandButton, TextBox, ListBox, Label, OvalShape dan jangan lupa **SerialPort**

. Atur komponen tersebut dan beri nama seperti yang saya contohkan.

```
Public Class Form1
```

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

```
Dim ports As String() = IO.Ports.SerialPort.GetPortNames()  
' Display each port name to ListBox1  
Dim port As String  
For Each port In ports  
    ListBox1.Items.Add(port)  
Next port
```

```
OvalShapeLED.FillStyle = PowerPacks.FillStyle.Solid  
End Sub
```

```
Private Sub ButtonConnect_Click(sender As Object, e As EventArgs) Handles  
    ButtonConnect.Click  
    Try  
        SerialPort1.PortName = ListBox1.SelectedItem  
        SerialPort1.BaudRate = 9600  
        SerialPort1.Open()  
        LabelConnectionStatus.Text = ListBox1.SelectedItem & " is connected."  
    Catch ex As Exception  
        LabelConnectionStatus.Text = ListBox1.SelectedItem & " is not connected."  
    End Try  
End Sub
```

```
Private Sub ButtonLEDON_Click(sender As Object, e As EventArgs) Handles  
    ButtonLEDON.Click  
    If SerialPort1.IsOpen Then  
        SerialPort1.WriteLine("A")  
    Else  
        MsgBox("Serialport is not connected.")  
    End If  
End Sub
```

```
Private Sub ButtonLEDOFF_Click(sender As Object, e As EventArgs) Handles  
    ButtonLEDOFF.Click  
    If SerialPort1.IsOpen Then  
        SerialPort1.WriteLine("B")  
    Else  
        MsgBox("Serialport is not connected.")  
    End If  
End Sub
```

```
Private Sub SerialPort1_DataReceived(sender As Object, e As  
    IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived  
    TextBoxReceiver.Invoke(New myDelegate(AddressOf updateTextBox), New Object() {})  
End Sub
```



## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

```
Public Delegate Sub myDelegate()  
Public Sub updateTextBox()
```

```
    TextBoxReceiver.Text = SerialPort1.ReadLine  
    If TextBoxReceiver.Text.Length > 3 Then  
        TextBoxReceiver.Text = Mid(TextBoxReceiver.Text, 1, 3)  
    End If  
    If TextBoxReceiver.Text = "L" Then  
        OvalShapeLED.FillColor = Color.Blue  
    End If  
    If TextBoxReceiver.Text = "H" Then  
        OvalShapeLED.FillColor = Color.Red  
    End If  
End Sub  
  
End Class
```

Supaya tidak repot mencoba program di atas, jangan lakukan Copy-Paste seluruh program, tapi silakan copy-paste satu per satu sub program. Nama program dan parameternya jangan Anda ketik, tapi pilihlah method dan event pada editor atau lakukan double click pada object yang bersangkutan saat di modus design.

Catatan mengenai komunikasi serial:

pada contoh pengiriman data pada program ini, kita menggunakan karakter dengan nomor ASCII di bawah 128. Jika Anda ingin menerapkan pengiriman data yang lebih kompleks, dan membutuhkan karakter dengan nomor ASCII sampai 255, maka kita harus mengkonfigurasi SerialPort dengan menambahkan kode berikut di atas perintah **SerialPort.Open()**

```
SerialPort1.Encoding = System.Text.Encoding.GetEncoding(1252)
```

## Komunikasi Arduino dengan VB.NET (serial port)

Written by Mada Jimmy

Monday, 31 March 2014 19:49 - Last Updated Sunday, 04 May 2014 07:44

---

Selamat mencoba.