1. Source Code Billing Game PlayStation

a. Form Login

```
Imports System.IO.Ports
Imports System.Threading

Public Class Form1


    Private Sub Form1_Load(sender As System.Object, e
As System.EventArgs) Handles MyBase.Load
        ProgressBar1.Hide()

    End Sub

    Private Sub Button1_Click(sender As System.Object,
e As System.EventArgs) Handles Button1.Click
        If TextBox1.Text = "12" And TextBox2.Text =
"12" Then
            Timer5.Start()
            ProgressBar1.Show()
        Else
            If TextBox1.Text = "" And TextBox2.Text =
"" Then
                MsgBox(" Masukkan Nama & Password
Anda", MsgBoxStyle.Critical, "Error")
            Else
                If TextBox1.Text = "" Then
                    MsgBox("Masukkan Nama Anda",
MsgBoxStyle.Critical, "Error")
                Else
                    If TextBox2.Text = "" Then
                        MsgBox("Masukkan Password
Anda", MsgBoxStyle.Critical, "Error")
                    Else
                        MessageBox.Show("Nama Atau
Pasword Anda Salah Broo", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
                    End If
```

```
                End If
            End If
        End If

    End Sub

    Private Sub Timer5_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer5.Tick
        ProgressBar1.Increment(35)
        If ProgressBar1.Value = ProgressBar1.Maximum
Then
            Timer5.Stop()
            SerialPort1.PortName = ("com12") 'change
com port to match your Arduino portSerialPort1.Close()
            SerialPort1.BaudRate = 9600
            SerialPort1.DataBits = 8
            SerialPort1.Parity = Parity.None
            SerialPort1.StopBits = StopBits.One
            SerialPort1.Handshake = Handshake.None
            SerialPort1.Encoding =
System.Text.Encoding.Default 'very important!

            MsgBox("Selamat datang_Di My PS Control")
            Form2.Show

        End If
    End Sub

    Private Sub Button2_Click(sender As System.Object,
e As System.EventArgs) Handles Button2.Click
        Close()
    End Sub

    Private Sub Timer4_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer4.Tick


    End Sub
```

```
    Private Sub Label4_Click(sender As System.Object, e
As System.EventArgs) Handles Label4.Click


    End Sub

    Private Sub Label3_Click(sender As System.Object, e
As System.EventArgs) Handles Label3.Click

    End Sub

    Private Sub Label6_Click(sender As System.Object, e
As System.EventArgs) Handles Label6.Click

    End Sub
End Class
```

b. Form Billing

```
Imports System.IO.Ports
Imports System.Threading
Public Class Form2

    Shared _continue As Boolean
    Shared _serialPort As SerialPort
    Dim tspn As New TimeSpan
    Dim pinout4 As Boolean = True
    Dim pinout5 As Boolean = True
    Dim pinout6 As Boolean = True
    Dim pinout7 As Boolean = True
    Dim ts As TimeSpan
    Dim result As String
    Dim Write As MsgBoxResult



    Private Sub Mulai1_Click(sender As System.Object, e
As System.EventArgs) Handles Mulai1.Click
        If IsNumeric(TextBox1.Text) AndAlso
CInt(TextBox1.Text) > 0 Then
            tspn = New TimeSpan(0, CInt(TextBox1.Text),
0)
            Timer1.Enabled = True
            Label1.Enabled = True
            Pause1.Enabled = True
            Stop1.Enabled = True
            SerialPort1.Open()
            If pinout4 = True Then
                SerialPort1.Write("1")
                SerialPort1.Close()
                Timer4.Enabled = True
            End If

        Else : MessageBox.Show("Set waktunya dulu
Broo", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
```

```
        End If
    End Sub

    Private Sub Stop1_Click(sender As System.Object, e
As System.EventArgs) Handles Stop1.Click
        Timer1.Enabled = False
        Label1.Enabled = False
        Pause1.Enabled = False
        SerialPort1.Open()
        If pinout4 = True Then
            SerialPort1.Write("5")
            SerialPort1.Close()
            Stop1.Enabled = False
        End If
    End Sub

    Private Sub Stop2_Click(sender As System.Object, e
As System.EventArgs) Handles Stop2.Click
        Timer2.Enabled = False
        Label2.Enabled = True
        Pause2.Enabled = False
        SerialPort1.Open()
        If pinout5 = True Then
            SerialPort1.Write("6")
            SerialPort1.Close()
            Stop2.Enabled = False
        End If
    End Sub

    Private Sub Mulai3_Click(sender As System.Object, e
As System.EventArgs) Handles Mulai3.Click
        If IsNumeric(TextBox3.Text) AndAlso
CInt(TextBox3.Text) > 0 Then
            tspm = New TimeSpan(0, CInt(TextBox3.Text),
0)
            Timer3.Enabled = True
            Label3.Enabled = True
            Pause3.Enabled = True
            Stop3.Enabled = True
```

```
            SerialPort1.Open()
            If pinout6 = True Then
                SerialPort1.Write("3")
                SerialPort1.Close()
            End If
        Else : MessageBox.Show("Set waktunya dulu
Broo", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)

            Timer4.Enabled = True
        End If
    End Sub

    Private Sub Stop3_Click(sender As System.Object, e
As System.EventArgs) Handles Stop3.Click
        Timer3.Enabled = False
        Label3.Enabled = False
        Pause3.Enabled = False
        SerialPort1.Open()
        If pinout6 = True Then
            SerialPort1.Write("7")
            SerialPort1.Close()
            Stop3.Enabled = False
        End If
    End Sub

    Private Sub Pause1_Click(sender As System.Object, e
As System.EventArgs) Handles Pause1.Click
        Timer1.Enabled = False
        Lanjut1.Enabled = True
        SerialPort1.Open()
        If pinout4 = True Then
            SerialPort1.Write("5")
            SerialPort1.Close()
            Pause1.Enabled = False

        End If
    End Sub
```

```
    Private Sub Lanjut1_Click(sender As System.Object,
e As System.EventArgs) Handles Lanjut1.Click
        Timer1.Enabled = True
        Pause1.Enabled = True
        SerialPort1.Open()
        If pinout4 = True Then
            SerialPort1.Write("1")
            SerialPort1.Close()
            Lanjut1.Enabled = False
        End If
    End Sub

    Private Sub Pause2_Click(sender As System.Object, e
As System.EventArgs) Handles Pause2.Click
        Timer2.Enabled = False
        Lanjut2.Enabled = True
        SerialPort1.Open()
        If pinout5 = True Then
            SerialPort1.Write("6")
            SerialPort1.Close()
            Pause2.Enabled = False
        End If
    End Sub

    Private Sub Lanjut2_Click(sender As System.Object,
e As System.EventArgs) Handles Lanjut2.Click
        Timer2.Enabled = True
        Pause2.Enabled = True
        SerialPort1.Open()
        If pinout5 = True Then
            SerialPort1.Write("2")
            SerialPort1.Close()
            Lanjut2.Enabled = False
        End If
    End Sub

    Private Sub Pause3_Click(sender As System.Object, e
As System.EventArgs) Handles Pause3.Click
        Timer3.Enabled = False
        Lanjut3.Enabled = True
```

```
        SerialPort1.Open()
        If pinout6 = True Then
            SerialPort1.Write("7")
            SerialPort1.Close()
            Pause3.Enabled = False
        End If
    End Sub

    Private Sub Lanjut3_Click(sender As System.Object,
e As System.EventArgs) Handles Lanjut3.Click
        Timer3.Enabled = True
        Pause3.Enabled = True
        SerialPort1.Open()
        If pinout6 = True Then
            SerialPort1.Write("3")
            SerialPort1.Close()
            Lanjut3.Enabled = False
        End If
    End Sub

    Private Sub Form2_Load(sender As System.Object, e
As System.EventArgs) Handles MyBase.Load
        SerialPort1.PortName = ("com12") 'change com
port to match your Arduino portSerialPort1.Close()
        SerialPort1.BaudRate = 9600
        SerialPort1.DataBits = 8
        SerialPort1.Parity = Parity.None
        SerialPort1.StopBits = StopBits.One
        SerialPort1.Handshake = Handshake.None
        SerialPort1.Encoding =
System.Text.Encoding.Default 'very important!
        Timer4.Enabled = True
        Label4.Text = TimeOfDay
    End Sub

    Private Sub Mulai2_Click_1(sender As System.Object,
e As System.EventArgs) Handles Mulai2.Click
        If IsNumeric(TextBox2.Text) AndAlso
CInt(TextBox2.Text) > 0 Then
```

```
            ts = New TimeSpan(0, CInt(TextBox2.Text),
0)
            Timer2.Enabled = True
            Label2.Enabled = True
            Pause2.Enabled = True
            Stop2.Enabled = True
            SerialPort1.Open()
            If pinout5 = True Then
                SerialPort1.Write("2")
                SerialPort1.Close()
            End If
        Else : MessageBox.Show("Set waktunya dulu
Broo", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)

            Timer4.Enabled = True
        End If
    End Sub

    Private Sub Timer4_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer4.Tick
        Label4.Text = TimeOfDay
        Mulai1.Enabled = True
        Mulai2.Enabled = True
        Mulai3.Enabled = True
    End Sub

    Private Sub Timer1_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer1.Tick
        tspn = tspn.Subtract(New TimeSpan(0, 0, 1))
        Label1.Text = String.Format(" {0} Mins : {1}
Secs", tspn.Minutes, tspn.Seconds)
        If tspn.Minutes = 0 AndAlso tspn.Seconds = 0
Then
            If tspn.Seconds = 0 < 1 Then
                Timer1.Enabled = False
                SerialPort1.Open()
                If pinout4 = True Then
                    Timer1.Enabled = False
                    SerialPort1.Write("5")
```

```
                    SerialPort1.Close()
                    MsgBox("   P S 1 SELESAI   ")
                End If
            End If
        End If
    End Sub

    Private Sub Timer2_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer2.Tick
        ts = ts.Subtract(New TimeSpan(0, 0, 1))
        Label2.Text = String.Format(" {0} Mins : {1}
Secs", ts.Minutes, ts.Seconds)
        If ts.Minutes = 0 AndAlso ts.Seconds = 0 Then
            If ts.Seconds = 0 < 1 Then
                Timer2.Enabled = False
                SerialPort1.Open()
                If pinout5 = True Then
                    SerialPort1.Write("6")
                    Timer2.Enabled = False
                    SerialPort1.Close()
                    MsgBox("   P S 2 SELESAI   ")
                End If
            End If
        End If
    End Sub

    Private Sub Timer3_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer3.Tick
        tspm = tspm.Subtract(New TimeSpan(0, 0, 1))
        Label3.Text = String.Format(" {0} Mins : {1}
Secs", tspm.Minutes, tspm.Seconds)
        If tspm.Minutes = 0 AndAlso tspm.Seconds = 0
Then
            If tspm.Seconds = 0 < 1 Then
                Timer3.Enabled = False
                SerialPort1.Open()
                If pinout6 = True Then
                    SerialPort1.Write("7")
                    Timer3.Enabled = False
                    SerialPort1.Close()
```

```
                    MsgBox("    P S 3   SELESAI   ")
                End If
            End If
        End If
    End Sub

    Private Sub Timer5_Tick(sender As System.Object, e
As System.EventArgs) Handles Timer5.Tick
        Label5.Text = Label5.TextAlign

    End Sub

    Private Sub Label5_Click(sender As System.Object, e
As System.EventArgs) Handles Label5.Click

    End Sub
End Class
```

a. Script Tombol Mulai

```
Private Sub Mulai1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles mulai1.Click
        If IsNumeric(TextBox1.Text) AndAlso
CInt(TextBox1.Text) > 0 Then
        tspn = New TimeSpan(0, CInt(TextBox1.Text), 0)
        Timer1.Enabled = True
      Else : MessageBox.Show("Set waktunya dulu Broo", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        SerialPort1.Open()
        If pinout4 = True Then
          SerialPort1.Write("4 on")
          SerialPort1.Close()
          Timer4.Enabled = True
        End If
      End If
  End Sub
```

b. Script Tombol Stop

```
Private Sub Stop1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles stop1.Click
    Timer1.Enabled = False
    stop1.Enabled = True
    Label1.Visible = True
    SerialPort1.Open()
    If pinout4 = True Then
       SerialPort1.Write("4 off")
       SerialPort1.Close()
    End If
  End Sub
```

c. Script Tombol Pause

```
Private Sub Pause1_Click(sender As System.Object, e As
System.EventArgs) Handles Pause1.Click
    Timer1.Enabled = False
    Lanjut1.Enabled = True
    SerialPort1.Open()
    If pinout4 = True Then
        SerialPort1.Write("5")
        SerialPort1.Close()
        Pause1.Enabled = False
    End If
End Sub
```

d. Script Tombol Lanjut

```
    Private Sub Lanjut1_Click(sender As System.Object, e As
    System.EventArgs) Handles Lanjut1.Click
Timer1.Enabled = True
Pause1.Enabled = True
SerialPort1.Open()
If pinout4 = True Then
   SerialPort1.Write("1")
   SerialPort1.Close()
   Lanjut1.Enabled = False
End If
End Sub
```

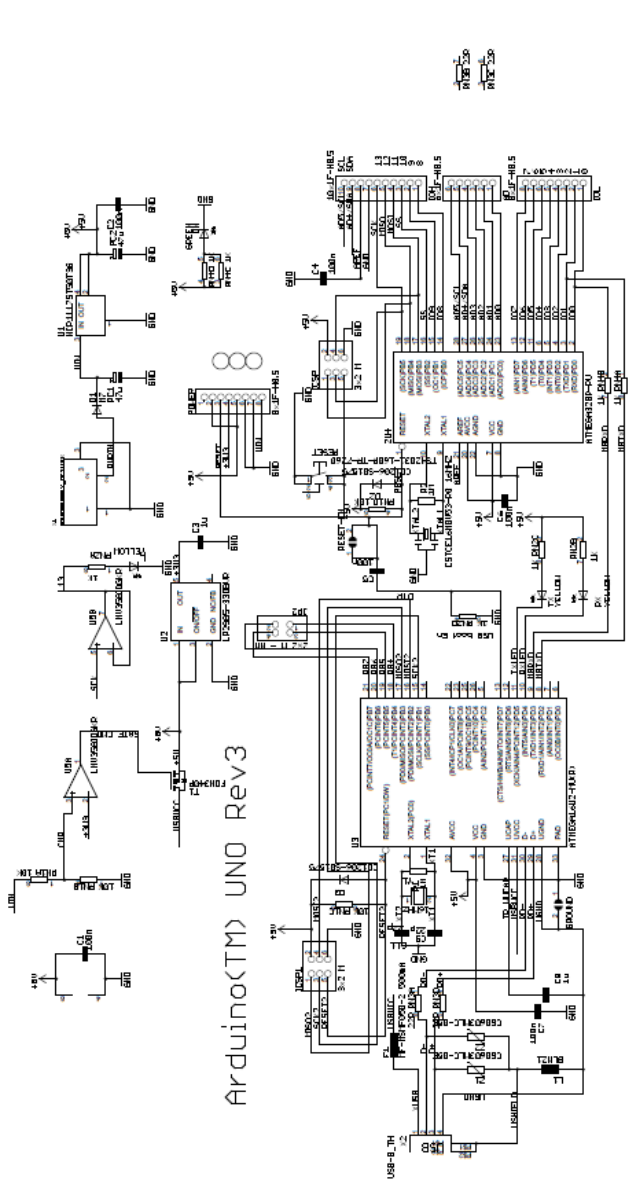
e. Script Code For Arduino

```
//BY Agung
//
int led3=3;
int led4=4;
int led5=5;
int led6=6;
{
 Serial.begin(9600);
 pinMode(led3,OUTPUT);
 pinMode(led4,OUTPUT);
 pinMode(led5,OUTPUT);
 pinMode(led6,OUTPUT);

}
void loop(){
 while(Serial.available()==0);
 int val=Serial.read()-'0';

 if(val==1){
  Serial.println("4, ON");digitalWrite(led4, HIGH);
 }
  if(val==2){
   Serial.println("5, ON");digitalWrite(led5, HIGH);
  }
   if(val==3){
   Serial.println("6, ON");digitalWrite(led6, HIGH);
   }
   if(val==4){
   Serial.println("3, ON");digitalWrite(led3, HIGH);
    }
  else if(val==5)
```

```
{
  Serial.println("4, OFF");digitalWrite(led4, LOW);
}
else if(val==6)
{
  Serial.println("5, OFF");digitalWrite(led5, LOW);
}
else if(val==7)
{
  Serial.println("6, OFF");digitalWrite(led6, LOW);
}
else if(val==8)
{
  Serial.println("3, OFF");digitalWrite(led3, LOW);
}
}
```

Arduino Uno R3

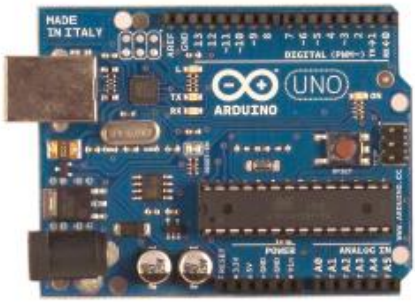
A. Schematic



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.
ARDUINO is a registered trademark.
Use of the ARDUINO name must be compliant with http://www.arduino.cc/en/Main/Policy

B. DataSheet Arduino Uno R3

Arduino UNO



CE

Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the Index of Arduino boards.

Index

RS radiospares RADIONICS

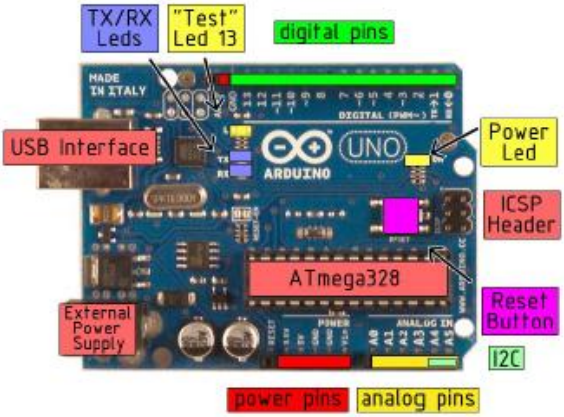
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



radiospares RADIONICS

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an ".inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

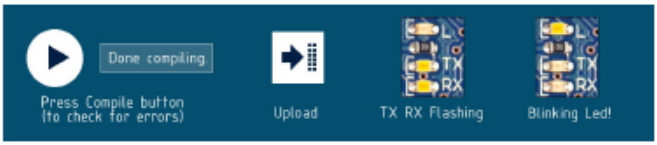
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

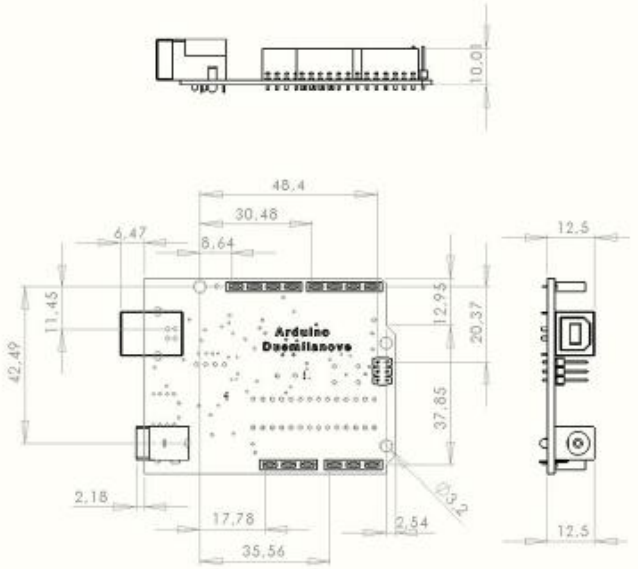
Once you have your skecth you'll see something very close to the screenshot on the right.

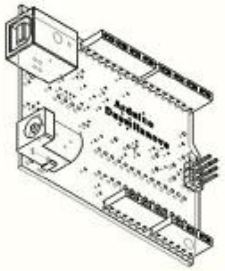
In **Tools>Board** select

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Dimensioned Drawing





RS radiospares RADIONICS

Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Enviromental Policies

The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.It. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

IMPATTO ZERO® www.lifegate.it LIFEGATE

RS radiospares RADIONICS

Quad bilateral switches 74HC/HCT4066

FEATURES

- Very low "ON" resistance:
  50 Ω (typ.) at V_{CC} = 4.5 V
  45 Ω (typ.) at V_{CC} = 6.0 V
  35 Ω (typ.) at V_{CC} = 9.0 V
- Output capability: non-standard
- I_{CC} category: SSI.

GENERAL DESCRIPTION

The 74HC/HCT4066 are high-speed Si-gate CMOS devices and are pin compatible with the "4066" of the "4000B" series. They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT4066 have four independent analog switches. Each switch has two input/output terminals (nY, nZ) and an active HIGH enable input (nE). When nE is LOW the belonging analog switch is turned off.

The "4066" is pin compatible with the "4016" but exhibits a much lower "ON" resistance. In addition, the "ON" resistance is relatively constant over the full input signal range.

QUICK REFERENCE DATA

GND = 0 V; T_{amb} = 25 °C; $t_r = t_f$ = 6 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PZH}/t_{PZL}	turn-on time nE to V_{os}	C_L = 15 pF; R_L = 1 kΩ; V_{CC} = 5 V	11	12	ns
t_{PHZ}/t_{PLZ}	turn-off time nE to V_{os}		13	16	ns
C_I	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per switch	notes 1 and 2	11	12	pF
C_S	max. switch capacitance		8	8	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in µW):
   a) $P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum \{(C_L + C_S) \times V_{CC}^2 \times f_o\}$ where:
   b) f_i = input frequency in MHz
   c) f_o = output frequency in MHz
   d) $\sum \{(C_L + C_S) \times V_{CC}^2 \times f_o\}$ = sum of outputs
   e) C_L = output load capacitance in pF
   f) C_S = maximum switch capacitance in pF
   g) V_{CC} = supply voltage in V
2. For HC the condition is V_I = GND to V_{CC}
   For HCT the condition is V_I = GND to V_{CC} − 1.5 V

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
74HC4066	DIP14	plastic dual in-line package; 14 leads (300 mil)	SOT27-1
74HC4066	SO14	plastic small outline package; 14 leads; body width 3.9 mm	SOT108-1
74HC4066	SSOP14	plastic shrink small outline package; 14 leads; body width 5.3 mm	SOT337-1
74HC4066	TSSOP14	plastic thin shrink small outline package; 14 leads; body width 4.4 mm	SOT402-1
74HCT4066	DIP14	plastic dual in-line package; 14 leads (300 mil)	SOT27-1
74HCT4066	SO14	plastic small outline package; 14 leads; body width 3.9 mm	SOT108-1
74HCT4066	SSOP14	plastic shrink small outline package; 14 leads; body width 5.3 mm	SOT337-1
74HCT4066	TSSOP14	plastic thin shrink small outline package; 14 leads; body width 4.4 mm	SOT402-1

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 4, 8, 11	1Y to 4Y	independent inputs/outputs
2, 3, 9, 10	1Z to 4Z	independent inputs/outputs
7	GND	ground (0 V)
13, 5, 6, 12	1E to 4E	enable inputs (active HIGH)
14	V_{CC}	positive supply voltage

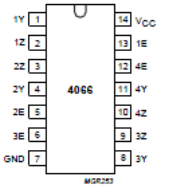


Fig.1 Pin configuration.

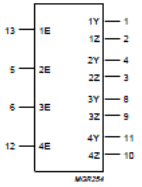
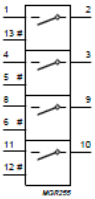
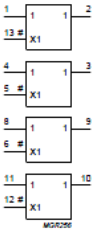


Fig.2 Logic symbol.



a.



b.

Fig.3 IEC logic symbol.

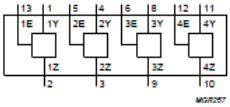


Fig.4 Functional diagram.

FUNCTION TABLE

INPUT NE	SWITCH
L	off
H	on

Note

1. H = HIGH voltage level; L = LOW voltage level.

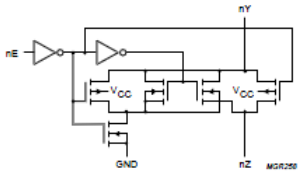


Fig.5 Schematic diagram (one switch).

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134) Voltages are referenced to GND (GND = 0 V)

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	CONDITIONS
V_{CC}	DC supply voltage	−0.5	+11.0	V	
$\pm I_{IK}$	DC digital input diode current		20	mA	for $V_I < -0.5$ V or $V_I > V_{CC} + 0.5$ V
$\pm I_{SK}$	DC switch diode current		20	mA	for $V_S < -0.5$ V or $V_S > V_{CC} + 0.5$ V
$\pm I_{IS}$	DC switch current		25	mA	for -0.5 V $< V_S < V_{CC} + 0.5$ V
$\pm I_{CC;}$ $\pm I_{GND}$	DC V_{CC} or GND current		50	mA	
T_{stg}	storage temperature range	−65	+150	°C	
P_{tot}	power dissipation per package				for temperature range: −40 to +125 °C 74HC/HCT
	plastic DIL		750	mW	above +70 °C: derate linearly with 12 mW/K
	plastic mini-pack (SO)		500	mW	above +70 °C: derate linearly with 8 mW/K
P_S	power dissipation per switch		100	mW	

Note

1. To avoid drawing V_{CC} current out of terminal nZ, when switch current flows in terminal nY, the voltage drop across the bidirectional switch must not exceed 0.4 V. If the switch current flows into terminal nZ, no V_{CC} current will flow out of terminal nY. In this case there is no limit for the voltage drop across the switch, but the voltages at nY and nZ may not exceed V_{CC} or GND.

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	74HC			74HCT			UNIT	CONDITIONS
		min.	typ.	max.	min.	typ.	max.		
V_{CC}	DC supply voltage	2.0	5.0	10.0	4.5	5.0	5.5	V	
V_I	DC input voltage range	GND		V_{CC}	GND		V_{CC}	V	
V_S	DC switch voltage range	GND		V_{CC}	GND		V_{CC}	V	
T_{amb}	operating ambient temperature range	−40		+85	−40		+85	°C	see DC and AC CHARACTERISTICS
T_{amb}	operating ambient temperature range	−40		+125	−40		+125	°C	
t_r, t_f	input rise and fall times		6.0	1000 500 400 250		6.0	500	ns	V_{CC} = 2.0 V V_{CC} = 4.5 V V_{CC} = 6.0 V V_{CC} = 10.0 V

DC CHARACTERISTICS FOR 74HC/HCT

For 74HC: V_{CC} = 2.0, 4.5, 6.0 and 9.0 V; For 74HCT: V_{CC} = 4.5 V

SYMBOL	PARAMETER	T_{amb} (°C)							UNIT	TEST CONDITIONS			
		74HC/HCT											
		+25			−40 to +85		−40 to +125			V_{CC} (V)	I_S (μA)	V_{IS}	V_I
		min.	typ.	max.	min.	max.	min.	max.					
R_{ON}	ON-resistance (peak)		–	–		–		–	Ω	2.0	100	V_{CC} to GND	V_{IH} or V_{IL}
			54	95		118		142	Ω	4.5	1000		
			42	84		105		126	Ω	6.0	1000		
			32	70		88		105	Ω	9.0	1000		
R_{ON}	ON-resistance (rail)		80	–		–		–	Ω	2.0	100	GND	V_{IH} or V_{IL}
			35	75		95		115	Ω	4.5	1000		
			27	65		82		100	Ω	6.0	1000		
			20	55		70		85	Ω	9.0	1000		
R_{ON}	ON-resistance (rail)		100	–		–		–	Ω	2.0	100	V_{CC}	V_{IH} or V_{IL}
			42	80		106		128	Ω	4.5	1000		
			35	75		94		113	Ω	6.0	1000		
			27	60		78		95	Ω	9.0	1000		
ΔR_{ON}	maximum variation of ON-resistance between any two channels		–						Ω	2.0		V_{CC} to GND	V_{IH} or V_{IL}
			5						Ω	4.5			
			4						Ω	6.0			
			3						Ω	9.0			

Note

1. At supply voltages approaching 2 V, the analog switch ON-resistanoe becomes extremely non-linear. Therefore it is recommended that these devioes be used to transmit digital signals only, when using these supply voltages.

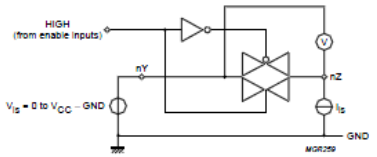


Fig.6 Test circuit for measuring ON-resistance (R_{ON}).

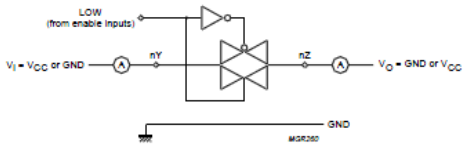


Fig.7 Test circuit for measuring OFF-state current.

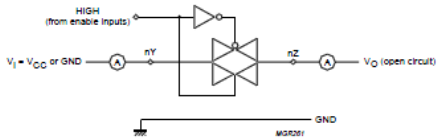


Fig.8 Test circuit for measuring ON-state current.

BIODATA DIRI

Agung Triawan yang akrab dipanggil Agung ini lahir di Surabaya, Jawa timur, 12 Juli 1992. Penulis lahir dari pasangan la bintang dan siti maryam ini merupakan anak ke tiga dari lima bersaudara. Penulis bersekolah pertama kali di TK Al-Ihsan Surabaya lalu menempuh pendidikan dasar di SD AL-Ihsan Surabaya. Tamat pendidikan dasar tahun 2004, bersama keluarganya penulis pindah ke Makassar Pada menyelesaikan sekolahnya di SMPN 1 MARIO-RIAWA Makasar pada tahun 2007. Setelah itu penulis kembali ke Surabaya dan bersekolah di SMA Wachid Hasyim 1 surabaya dan lulus pada tahun 2010.

Setelah tamat Sekolah Menengah Atas, penulis melanjutkan studinya ke universitas Muhammadiyah Surabaya dan lulus dengan gelar AMD (D3) program studi Teknik Komputer pada 23 Mei 2015.

UNIVERSITAS MUHAMMADIYAH SURABAYA
PUSAT BAHASA
Jl. Sutorejo 59 Surabaya 60113 Telp. 031-3811966, 3811967 Ext (130) Gd. A Lt 2
Email: pusba.umsby@gmail.com

ENDORSEMENT LETTER

094/PB-UMS/EL/V/2015

This letter is to certify that the abstract of the thesis below

Title	:	Design Systems of Switch-Timer Game of Play-Station on the Base of Arduino Uno R3.
Student's name	:	Agung Triawan
Reg. Number	:	20110335011
Department	:	D3 Teknik Komputer

has been endorsed by Pusat Bahasa *UMSurabaya* for further approval by the examining committee of the faculty.

Surabaya, 20 Mei 2015
Chair,

Sulton Dedi Wijaya, S.Pd