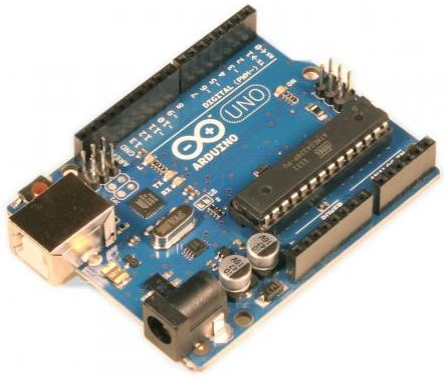
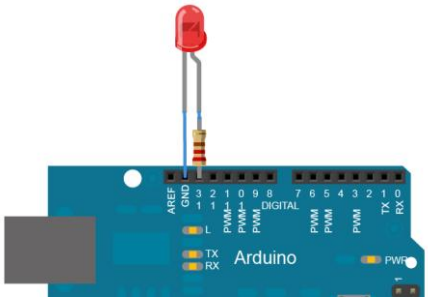
Docklight Application Note: Arduino Serial Communication

Docklight Application Note: Arduino Serial Communication 1
1 Arduino Board – Serial Communication and LED Output 1
2 Quick Start – Blinking LED Example 1
3 Advanced Example: Creating Morse Codes Using Docklight Scripting 4
4 Appendix A: ArduinoSerialLED.ino Program Code 5
5 Appendix B: Docklight-Arduino-MorseCode.pts script code 6
6 License and Copyright 7
7 References 7

1 Arduino Board – Serial Communication and LED Output



We have tested the Arduino serial communication described in this Application Note using an Arduino UNO. It should also work on other Arduino boards [3].



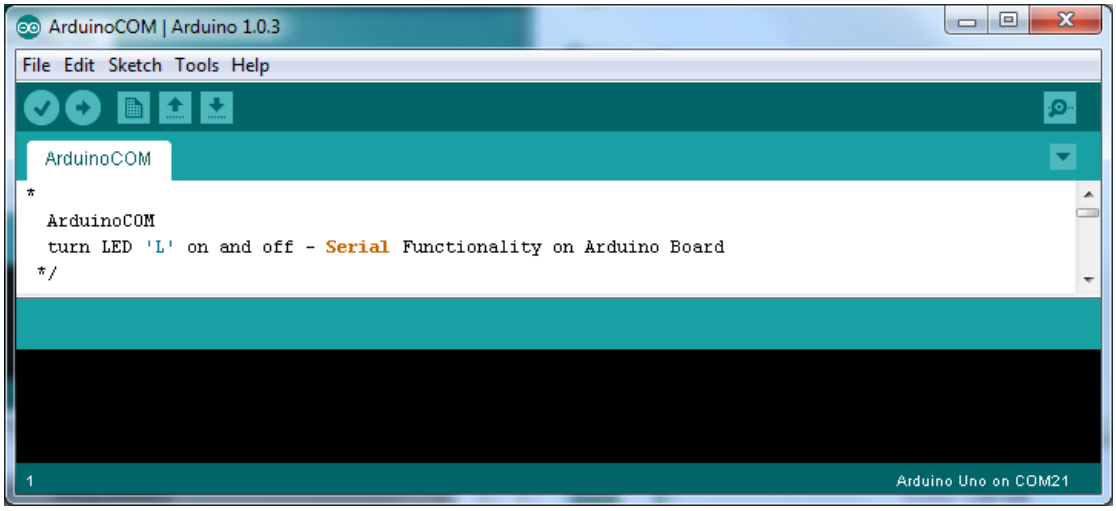
Most Arduino boards already have an on-board LED ‘L’ wired to the output pin no 13. If you run this example with no hardware attached to the Arduino, you should see this on-board LED ‘L’ blink.

2 Quick Start – Blinking LED Example

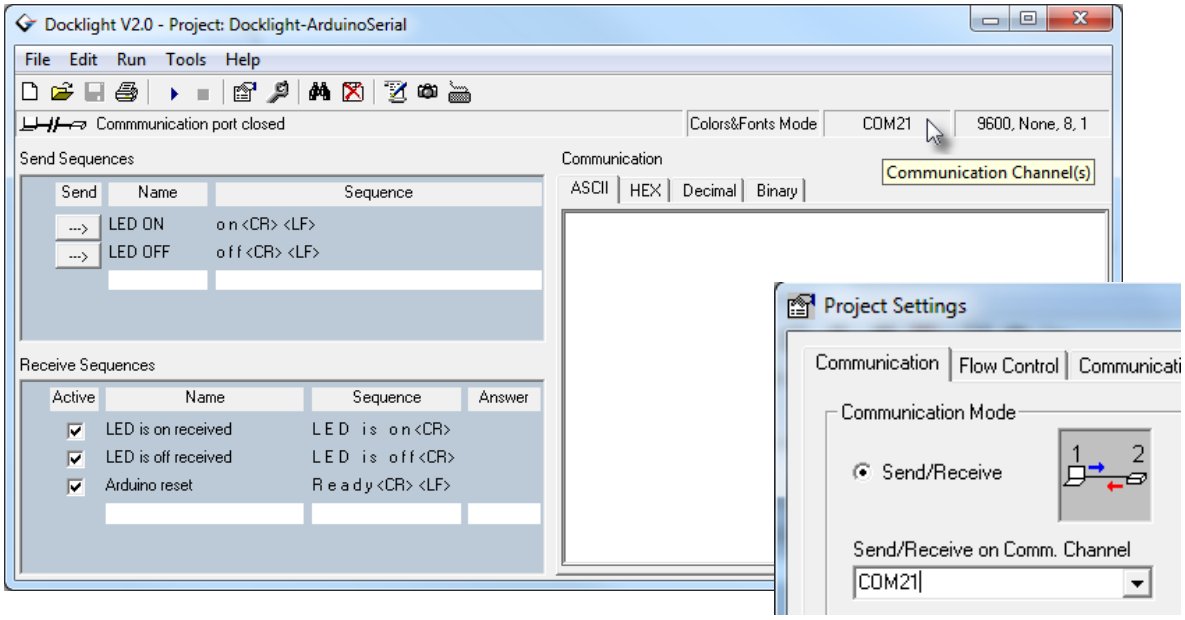
This is a very basic Arduino communication and programming example using Docklight. It builds on what is described as the “The "Hello World!" of Physical Computing” [6] on the Arduino web site.

- Download and install Docklight V2.0 (or higher) from http://www.docklight.de/download_en.htm
- Download and install the latest Arduino environment from http://arduino.cc/en/Main/Software [4]
- Download and extract the Arduino/Docklight example projects: https://docklight.de/exampleFiles/devices/ArduinoExamples.zip [11]

- Connect your Arduino board to your PC via USB.
- run **arduino.exe**
- Open the Arduino Sketch file **ArduinoCOM.ino**



- Upload the Sketch to your Arduino board, e.g. by using the arrow right toolbar .
- Start **Docklight** and open the project file **Docklight-ArduinoSerial.ptp.**
- Double click on the **Communication Channel(s)** area that shows COM21 by default, or use menu **Tools -> Project Settings -> Communication** to adjust the COM port, if necessary.



NOTE: The correct COM port to use is shown in the right lower corner of the **Arduino** environment (**COM21** in this example).

You can also find out using the **Windows Device Manager**: Use the Windows **Start** menu and type **Device Manager** in the search box. In the Device Manager dialog, check the section **Ports (COM & LPT)** for the Arduino entry.

TIP: If the COM port used for your Arduino Hardware is not shown in Docklight Project Settings drop down list, just type in the correct COM port in the text box, e.g. COM21.

- Press **Start Communication (F5)** in Docklight.

You should see this message in the Docklight Communication Window (**ASCII**)

```
2013-02-18 20:07:09.024 [RX] - Ready
<CR><LF>
 Arduino communication established
```

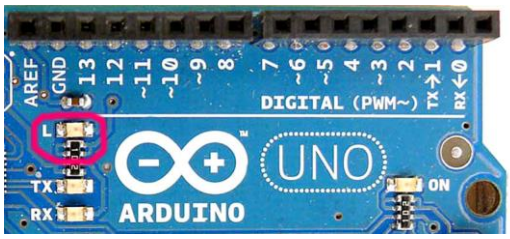
- Using the Docklight **Send -->** buttons, you can now transmit the **LED ON** and **LED OFF** ASCII commands to your Arduino board.

ASCII Commands and confirmation messages from the Arduino are shown in the Docklight Communication Window (ASCII):

```
2013-02-18 20:08:44.297 [TX] - on<CR><LF>
2013-02-18 20:08:44.304 [RX] - LED is on<CR><LF>
Arduino reports: LED on

2013-02-18 20:08:46.031 [TX] - off<CR><LF>
2013-02-18 20:08:46.041 [RX] - LED is off<CR><LF>
Arduino reports: LED off
```

Of course you will now also see the Arduino board LED 'L' or an external PIN13 LED turned off or on.



3 Advanced Example: Creating Morse Codes Using Docklight Scripting

This is a more advanced command sequence example using **Docklight Scripting**, an extended edition of Docklight that features a VBScript-based programming environment.

NOTE: VBScript/Docklight Scripting provide easy serial communication access and similar programming capabilities to the Arduino C/C++ language set, but the language syntax is different in detail. See also http://www.docklight.de/manual/vbscriptoverview.htm [2] and http://arduino.cc/en/Reference/Comparison [9].

How to run the Docklight Scripting example:

- Make sure the Arduino environment is set up and ready, as explained in the previous section 2: “Quick Start – Blinking LED Example”.
- Download and install Docklight Scripting V2.0 (or higher) from http://www.docklight.de/download_en.htm
- Download and extract the Arduino/Docklight example projects: https://docklight.de/exampleFiles/devices/ArduinoExamples.zip [11]
- Start Docklight Scripting and open the project file **Docklight-ArduinoSerial.ptp.** Then open the script **Docklight-Arduino-MorseCode.pts**, using menu **Scripting –> Open Script…**
- Run the script using menu **Scripting -> Run Script (Shift+F5)**.

You should see these messages in Docklight Communication Window (**Hex**)

```
2013-02-18 20:22:34.924 [RX] - 52 65 61 64 79 0D 0A  Arduino communication established

 Sending character 'S' = ...

2013-02-18 20:22:35.720 [TX] - 6F 6E 0D 0A
2013-02-18 20:22:35.727 [RX] - 4C 45 44 20 69 73 20 6F 6E 0D 0A  Arduino reports: LED
on

2013-02-18 20:22:36.041 [TX] - 6F 66 66 0D 0A
2013-02-18 20:22:36.050 [RX] - 4C 45 44 20 69 73 20 6F 66 66 0D 0A  Arduino reports:
LED off

2013-02-18 20:22:36.375 [TX] - 6F 6E 0D 0A
2013-02-18 20:22:36.382 [RX] - 4C 45 44 20 69 73 20 6F 6E 0D 0A  Arduino reports: LED
on

2013-02-18 20:22:36.716 [TX] - 6F 66 66 0D 0A
2013-02-18 20:22:36.726 [RX] - 4C 45 44 20 69 73 20 6F 66 66 0D 0A  Arduino reports:
LED off

2013-02-18 20:22:37.052 [TX] - 6F 6E 0D 0A
2013-02-18 20:22:37.058 [RX] - 4C 45 44 20 69 73 20 6F 6E 0D 0A  Arduino reports: LED
on

2013-02-18 20:22:37.395 [TX] - 6F 66 66 0D 0A
2013-02-18 20:22:37.402 [RX] - 4C 45 44 20 69 73 20 6F 66 66 0D 0A  Arduino reports:
LED off
```

And of course the Arduino LED should blink and send out that famous Morse code [10] sequence we all know from them sinking ships movies…

4 Appendix A: ArduinoSerialLED.ino Program Code

```
/*
  ArduinoSerialLED.ino
  turn LED 'L' on and off via 'Serial' Functionality on Arduino Board
 */

 // use output pin no. 13 / on board LED 'L'
 const unsigned int LED_PIN = 13;

 // use standard baud rate 9600 baud.
 const unsigned long BAUD_RATE = 9600;
 // Other baud rates are possible, too, e.g. 57600. Just make sure your
 // Docklight project settings are adjusted accordingly
 // (Docklight menu Tools -> Project Settings ... -> Communication).

 // init command buffer
 String command = "";
 boolean command_available = false;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication according to BAUD_RATE:
  Serial.begin(BAUD_RATE); // other settings are SERIAL_8N1, 8 data bits, no parity, one
stop bit
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  pinMode(LED_PIN, OUTPUT); // configure LED_PIN as output

  // Serial.available() < 0 would indicate an error on initialization
  if (Serial.available() >= 0)
     Serial.println("Ready");   // send Ready<CR><LF> at start/reset
}

// after completing setup(), this gets called consecutively
void loop() {
  if (command_available) {
    // commands received need to be terminated by <CR><LF> characters
     // (same as Serial.println does it)
    if (command == "on\r\n") {
      digitalWrite(LED_PIN, HIGH);
      Serial.println("LED is on");
    }
    else if (command == "off\r\n") {
      digitalWrite(LED_PIN, LOW);
      Serial.println("LED is off");
    }
    else {
      Serial.print("Unknown command: "+ command);
    }
    command = ""; // clear input buffer
    command_available = false;
  }
}

// event Handling - reading characters
void serialEvent(){
  while (Serial.available() && !command_available) {
    const char c = Serial.read();
    if (c == '\n') { // Linefeed character is end-of-command mark
      command += c;
      command_available = true;
```

```
    }
    else command += c;  // add character for receive string
  }
}
```

5 Appendix B: Docklight-Arduino-MorseCode.pts script code

```
' Docklight-Arduino-MorseCode.pts
' Author: MF
' Date: 2013-02-18
' Example script for Docklight Application Note
' Docklight_Application_Note_ArduinoSerial.pdf
'
' How to Use:
' - Open Docklight project "Docklight-ArduinoSerial.ptp"
' - cross-check COM port settings as indicated in the project file
'   and Application Note
' - Start Script: Scripting -> Run Script (Shift + F5)
'

' speed of the Morse code
unitDelayMSec = 200
' Morse code table from A-Z
strCodeList = ".-,-...,-.-.,-..,.,..-.,--.,....,..,.---,-.-,.-..,--,-.,---,.--.,--.-,.-
.,...,-,..-,...-,.--,-..-,-.--,--.."
morseArray = Split(strCodeList, ",")

DL.ClearCommWindows
DL.StartCommunication
Do
    sendMorseMessage "SOS "
Loop

Sub sendMorseMessage(ByVal strMessage)
    For i = 1 To Len(strMessage)
        nextLetter = Mid(strMessage, i, 1)
        If nextLetter = " " Then
            ' "the space between words is seven units"
            DL.Pause(4 * unitDelayMSec)  ' (we already had 3 units delay after completing
the last letter)
        Else
            sendMorse(Mid(strMessage, i, 1))
            ' "the space between letters is three units"
            DL.Pause(2 * unitDelayMSec) ' (we already had one unit delay within the For
loop)
        End If
    Next
End Sub

Sub sendMorse(ByVal letter)
    morseCode = getMorseCode(letter)
    DL.AddComment()
    DL.AddComment("Sending letter '" & letter & "' = " & morseCode)
    For j = 1 To Len(morseCode)
        DL.SendSequence("LED ON")
        If Mid(morseCode, j, 1) = "-" Then
            DL.Pause(3 * unitDelayMSec) ' "a dash is three units"
        Else
            DL.Pause(unitDelayMSec)  ' "the length of a dot is one unit"
        End If
        DL.SendSequence("LED OFF")
        ' "the space between parts of the same letter is one unit"
        DL.Pause(unitDelayMSec)
```

```
    Next
End Sub

Function getMorseCode(ByVal letter)
    arrayIndex = Asc(UCase(letter))
    If arrayIndex >= 65 And arrayIndex <= 90 Then
        getMorseCode = morseArray(arrayIndex - 65)
    Else
        DL.AddComment("No morse code for '" & letter & "'")
        DL.Quit()
    End If
End Function
```

6 License and Copyright

The text of this Application Note is licensed under a Creative Commons Attribution-ShareAlike 3.0 License [12]. Code samples are released into the public domain.

Contact

E-Mail Support: docklight@fuh-edv.de
Flachmann & Heggelbacher
Waldkirchbogen 27
D-82061 Neuried (Munich)
Germany
http://www.fuh-edv.de

7 References

[1] Docklight User Manual
http://www.docklight.de/pdf/docklight_manual.pdf

[2] Docklight Scripting User Manual – VBScript Basics
http://www.docklight.de/manual/vbscriptoverview.htm

[3] Arduino Products
http://arduino.cc/en/Products

[4] Arduino Software Environment
http://arduino.cc/en/Main/Software

[5] Arduino Language Reference
http://arduino.cc/en/Reference/HomePage

[6] Arduino - The "Hello World!" of Physical Computing
http://www.arduino.cc/en/Tutorial/BlinkingLED

[7] Arduino Reference – Serial
http://arduino.cc/en/Reference/serial

[8] Arduino Tutorial – SerialEvent
http://arduino.cc/en/Tutorial/SerialEvent

[9] Arduino/Processing Language Comparison
http://arduino.cc/en/Reference/Comparison

[10] Wikipedia: Morse Code
http://en.wikipedia.org/wiki/Morse_code

[11] Arduino and Docklight example projects/scripts as used in this Application Note.
https://docklight.de/exampleFiles/devices/ArduinoExamples.zip

[12] Creative Commons Attribution-ShareAlike 3.0 License
http://creativecommons.org/licenses/by-sa/3.0/