How to install Espruino SW on STM32F4-Discovery

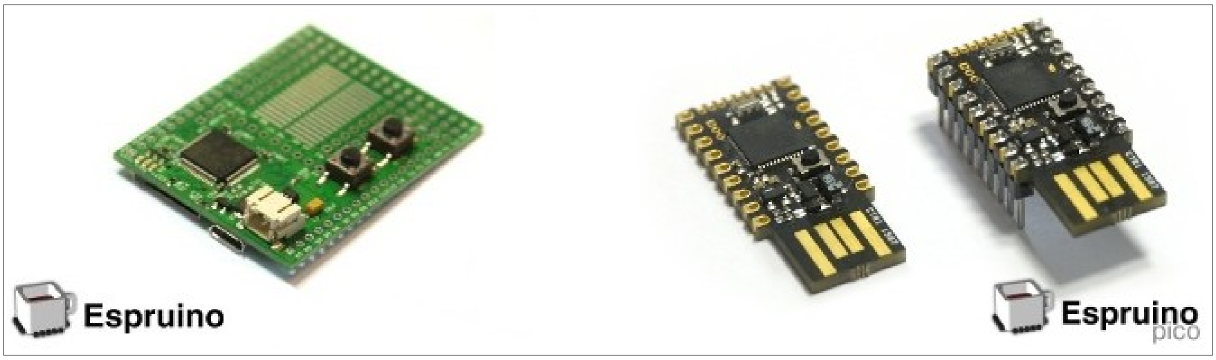
Introduction

The **Espruino** is a **JavaScript interpreter** (is probably about 95% JavaScript compatible) developed for STM32 mcu by Gordon Williams.
I like:

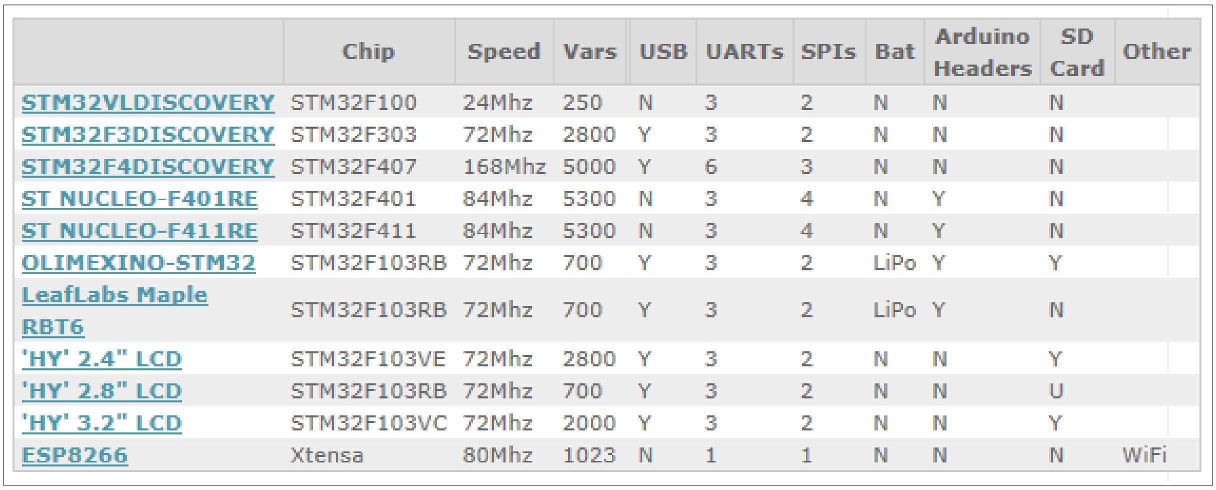
- The attention of the consumption, in fact the Espruino interpreter can put itself to sleep when it knows no action is required.
- Using JavaScript is that it opens up the world of microcontrollers to web developers.
- The interactive nature of the interpreter makes it easier to interface to different devices (LCDs, accelerometers etc) because commands can be tested one at a time.
  Espruino's interactive console allows users to enter code (such as '**analogRead(LIGHT_SENSOR)**') and have it executed, and the result returned immediately.

For more detail see the **Espruino web site**.

Espruino official boards are shown below.



Is it possible install the Espruino SW on a loot of STM32 boards, see **here**.
The BOARDS that Espruino works on, are shown below (December 2015).

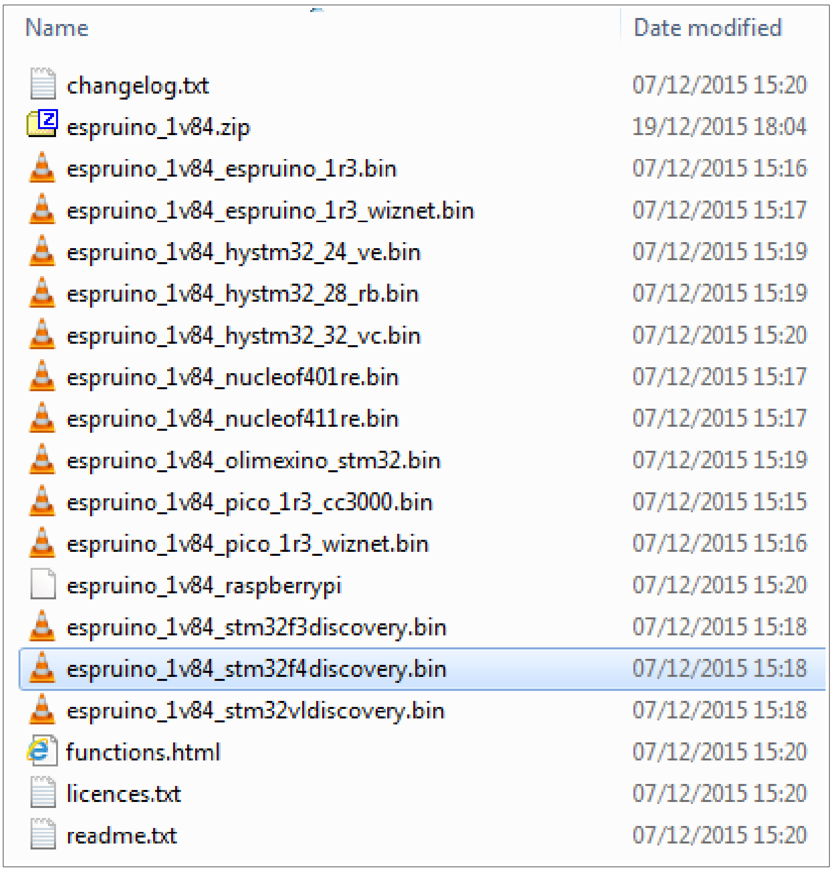


	Chip	Speed	Vars	USB	UARTs	SPIs	Bat	Arduino Headers	SD Card	Other
STM32VLDISCOVERY	STM32F100	24Mhz	250	N	3	2	N	N	N	
STM32F3DISCOVERY	STM32F303	72Mhz	2800	Y	3	2	N	N	N	
STM32F4DISCOVERY	STM32F407	168Mhz	5000	Y	6	3	N	N	N	
ST NUCLEO-F401RE	STM32F401	84Mhz	5300	N	3	4	N	Y	N	
ST NUCLEO-F411RE	STM32F411	84Mhz	5300	N	3	4	N	Y	N	
OLIMEXINO-STM32	STM32F103RB	72Mhz	700	Y	3	2	LiPo	Y	Y	
LeafLabs Maple RBT6	STM32F103RB	72Mhz	700	Y	3	2	LiPo	Y	N	
'HY' 2.4" LCD	STM32F103VE	72Mhz	2800	Y	3	2	N	N	Y	
'HY' 2.8" LCD	STM32F103RB	72Mhz	700	Y	3	2	N	N	U	
'HY' 3.2" LCD	STM32F103VC	72Mhz	2000	Y	3	2	N	N	Y	
ESP8266	Xtensa	80Mhz	1023	N	1	1	N	N	N	WiFi

DownLoad the Espruino SW

First download the last release of the Espruino SW from here:
http://www.espruino.com/Download

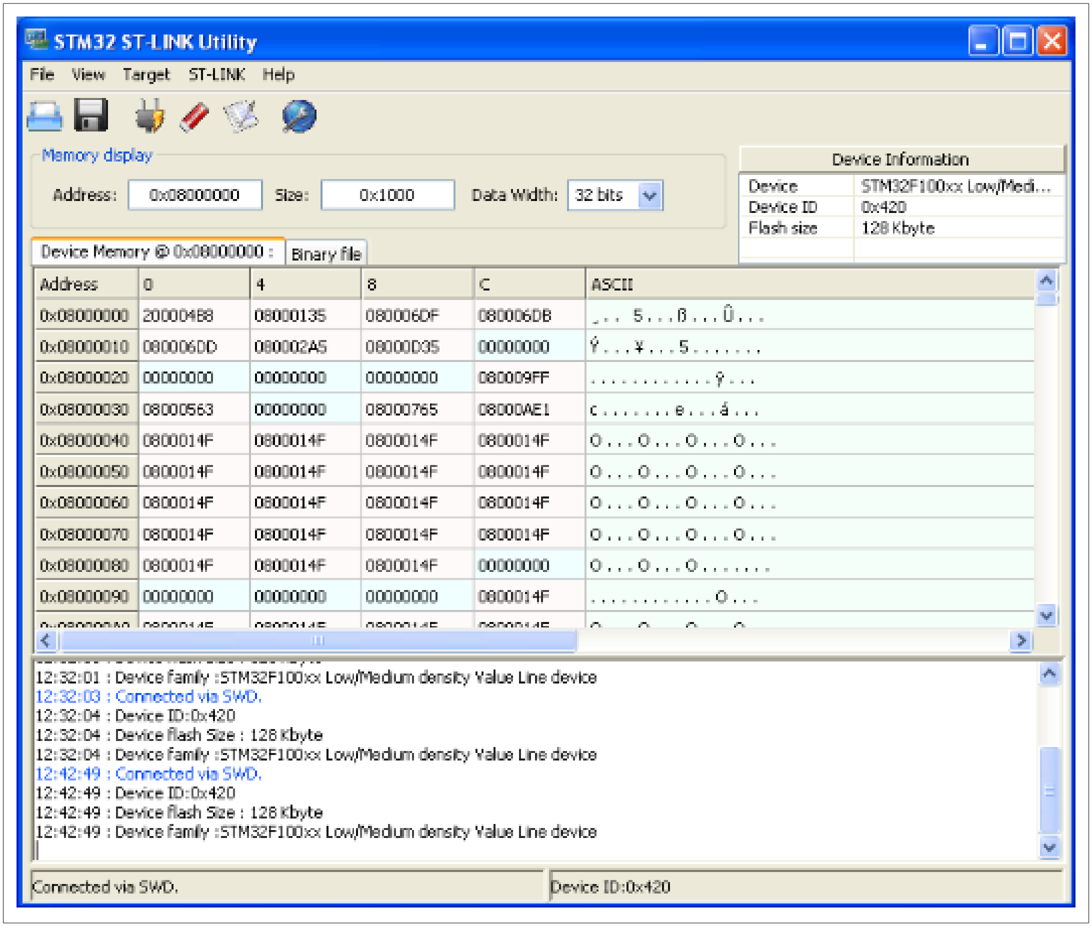
Unzip the SW and choose the right bin file for your board.
Below there is the list of bin files, up to now available, for some STM32 boards



Name	Date modified
changelog.txt	07/12/2015 15:20
espruino_1v84.zip	19/12/2015 18:04
espruino_1v84_espruino_1r3.bin	07/12/2015 15:16
espruino_1v84_espruino_1r3_wiznet.bin	07/12/2015 15:17
espruino_1v84_hystm32_24_ve.bin	07/12/2015 15:19
espruino_1v84_hystm32_28_rb.bin	07/12/2015 15:19
espruino_1v84_hystm32_32_vc.bin	07/12/2015 15:20
espruino_1v84_nucleof401re.bin	07/12/2015 15:17
espruino_1v84_nucleof411re.bin	07/12/2015 15:17
espruino_1v84_olimexino_stm32.bin	07/12/2015 15:19
espruino_1v84_pico_1r3_cc3000.bin	07/12/2015 15:15
espruino_1v84_pico_1r3_wiznet.bin	07/12/2015 15:16
espruino_1v84_raspberrypi	07/12/2015 15:20
espruino_1v84_stm32f3discovery.bin	07/12/2015 15:18
espruino_1v84_stm32f4discovery.bin	07/12/2015 15:18
espruino_1v84_stm32vldiscovery.bin	07/12/2015 15:18
functions.html	07/12/2015 15:20
licences.txt	07/12/2015 15:20
readme.txt	07/12/2015 15:20

Install the ST-LINLK-UTILITY

For install the Espruino SW on your STM32 board you must have installed on your PC the **ST-LINK-UTILITY**.



Address: 0x08000000 Size: 0x1000 Data Width: 32 bits

Device Information	
Device	STM32F100xx Low/Medi...
Device ID	0x420
Flash size	128 Kbyte

Address	0	4	8	C	ASCII
0x08000000	200004B8	08000135	080006DF	080006DB	¸... 5...ß...Û...
0x08000010	080006DD	080002A5	08000D35	00000000	Ý...¥...5.......
0x08000020	00000000	00000000	00000000	080009FF	............ÿ...
0x08000030	08000563	00000000	08000765	08000AE1	c.......e...á...
0x08000040	0800014F	0800014F	0800014F	0800014F	O...O...O...O...
0x08000050	0800014F	0800014F	0800014F	0800014F	O...O...O...O...
0x08000060	0800014F	0800014F	0800014F	0800014F	O...O...O...O...
0x08000070	0800014F	0800014F	0800014F	0800014F	O...O...O...O...
0x08000080	0800014F	0800014F	0800014F	00000000	O...O...O.......
0x08000090	00000000	00000000	00000000	0800014F	............O...

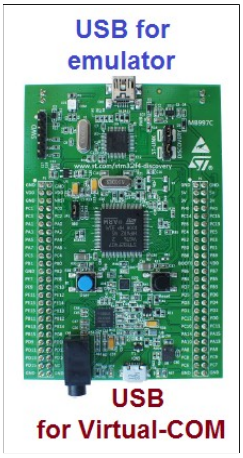
```
12:32:01 : Device family :STM32F100xx Low/Medium density Value Line device
12:32:03 : Connected via SWD.
12:32:04 : Device ID:0x420
12:32:04 : Device flash Size : 128 Kbyte
12:32:04 : Device family :STM32F100xx Low/Medium density Value Line device
12:42:49 : Connected via SWD.
12:42:49 : Device ID:0x420
12:42:49 : Device flash Size : 128 Kbyte
12:42:49 : Device family :STM32F100xx Low/Medium density Value Line device
```

Connected via SWD. Device ID:0x420

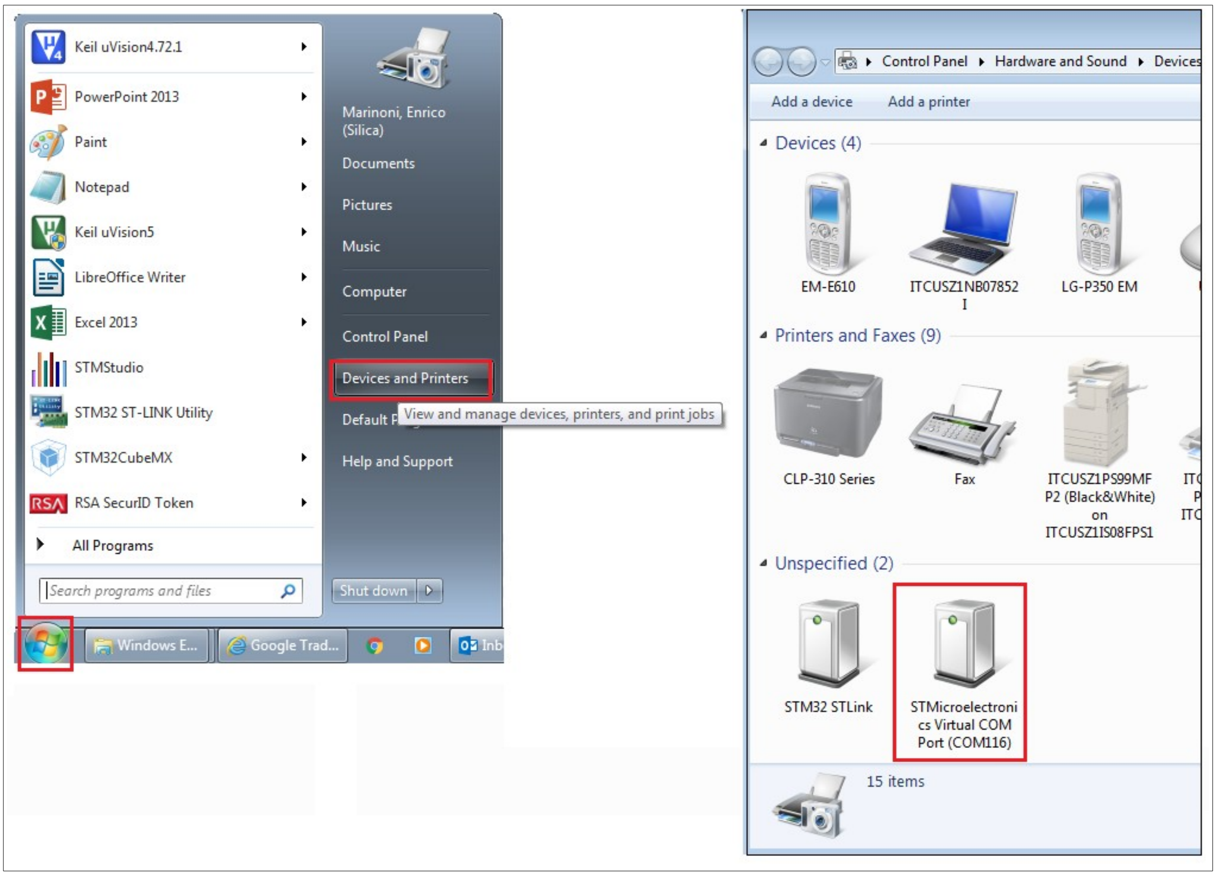
Install on the STM32F4-Discovery the Espruino SW

We decided to use the **STM32F4-Discovery** for testing the Espruino SW for this reason is necessary download on the STM32F4-Discovery this bin file:
espruino_1v84_stm32f4discovery.bin
for do this you must use the **ST-LINK-UTILITY**. See: **DownLoad the Espruino SW**

STM32F4-Discovery

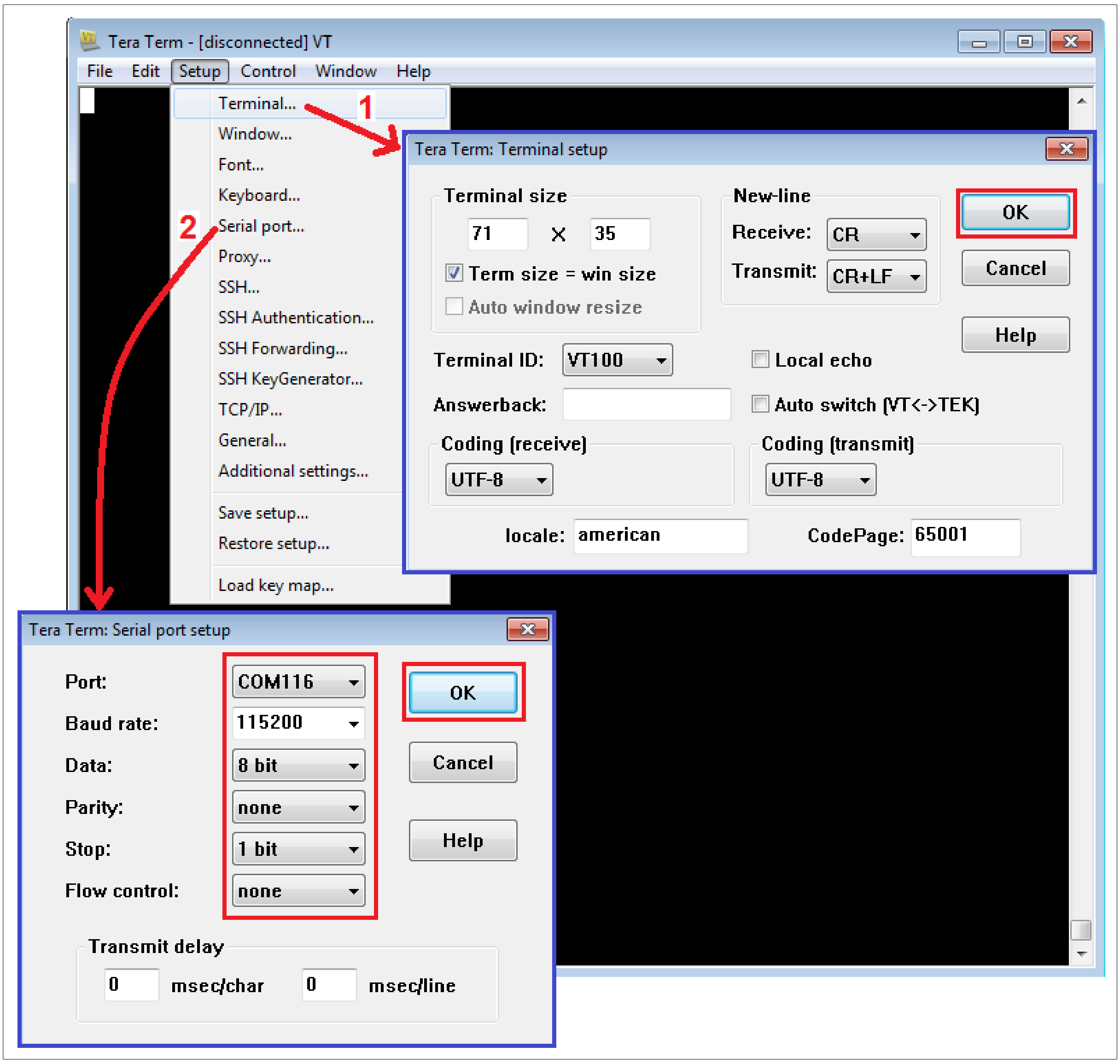


After the installation, connect a USB cable form: **USB for Virtual COM** to your **PC**.
You must see something like below.



Now is necessary a program for use the Virtual COM on the PC.
For Windows7, I suggest to install **Tera Term** (http://en.wikipedia.org/wiki/Tera_Term) download it from this link: http://ttssh2.sourceforge.jp/index.html.en

Run TeraTerm and configure it as shown below.



Tera Term: Terminal setup

Terminal size: 71 X 35
☑ Term size = win size
□ Auto window resize

New-line
Receive: CR
Transmit: CR+LF

Terminal ID: VT100
Answerback:
□ Local echo
□ Auto switch (VT<->TEK)

Coding (receive): UTF-8
Coding (transmit): UTF-8

locale: american
CodePage: 65001

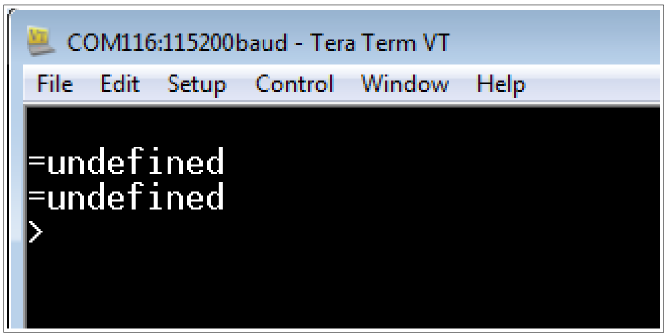
Tera Term: Serial port setup

Port: COM116
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char
0 msec/line

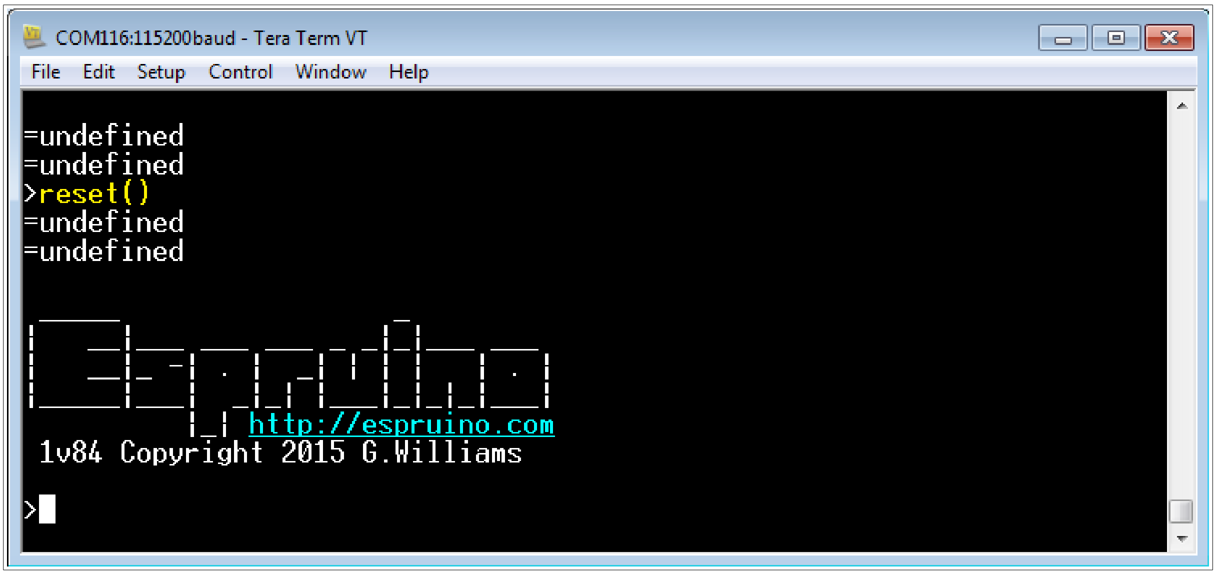
After the TeraTerm configuration you are connect with the STM32F4-Discovery.

For check if the connection is OK just simply press ENTER on PC keyboard.
You must see something like below.


COM116:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
=undefined
=undefined
>
```

Try this command: **reset()**
you must see something like below.

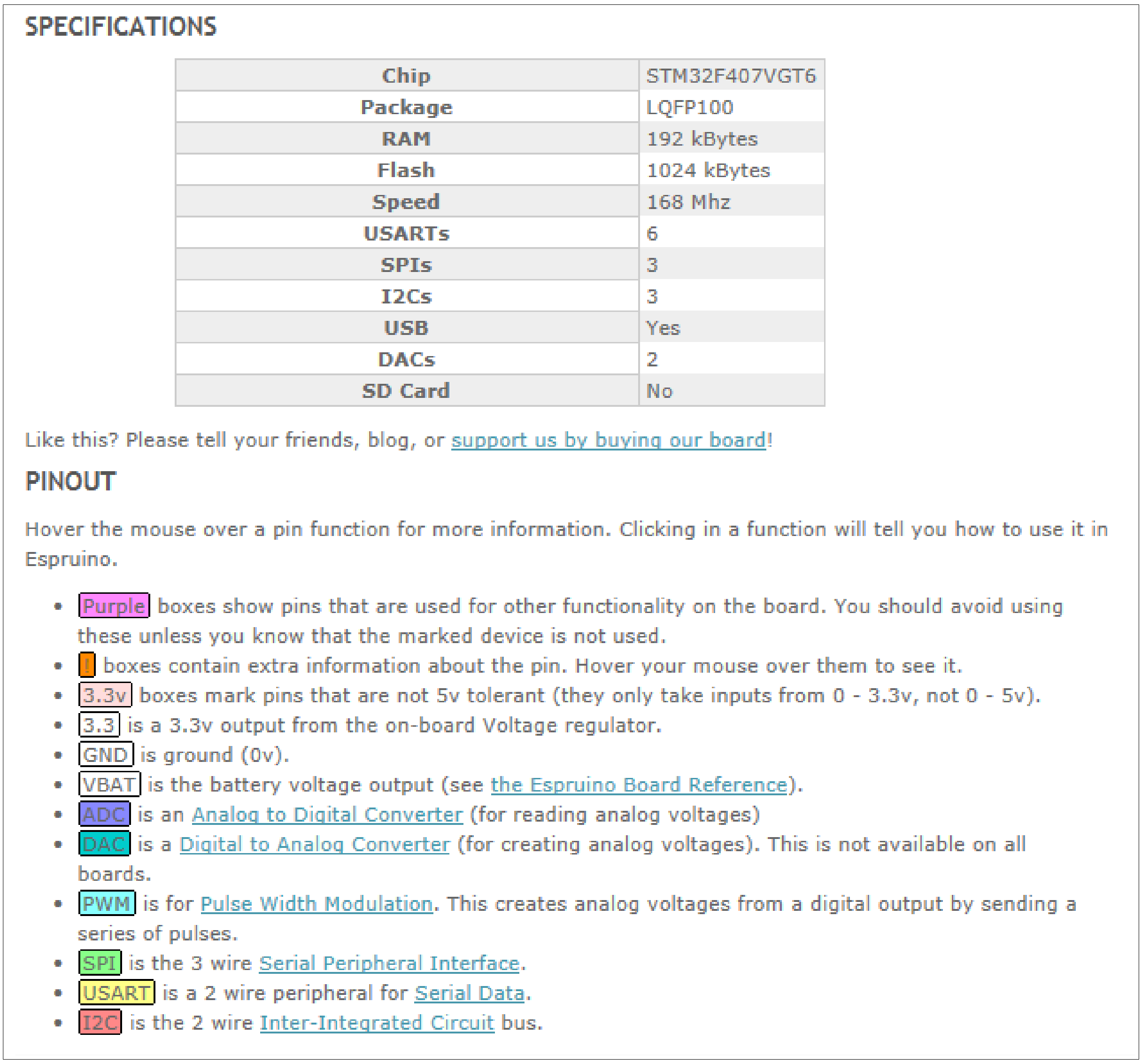

COM116:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
=undefined
=undefined
>reset()
=undefined
=undefined

 _____                 _
|   __|___ ___ ___ _ _|_|___ ___
|   __|_ -| . |  _| | | |   | . |
|_____|___|  _|_| |___|_|_|_|___|
          |_| http://espruino.com
 1v84 Copyright 2015 G.Williams

>
```

My first program

As I said before, now on the STM32F4-Discovery there is installed a **JavaScript interpreter**.
Every time you type a command and press enter, it will be executed immediately.
= will be displayed followed by the result.
If there is no result (for instance if you were executing a function that returned no value), **=undefined** is displayed.

Before to use our STM32F4-Discovery is necessary know the pin allocations.
All the boards supported from Espruino are **here**, we select the **STM32F4Discovery**.



SPECIFICATIONS

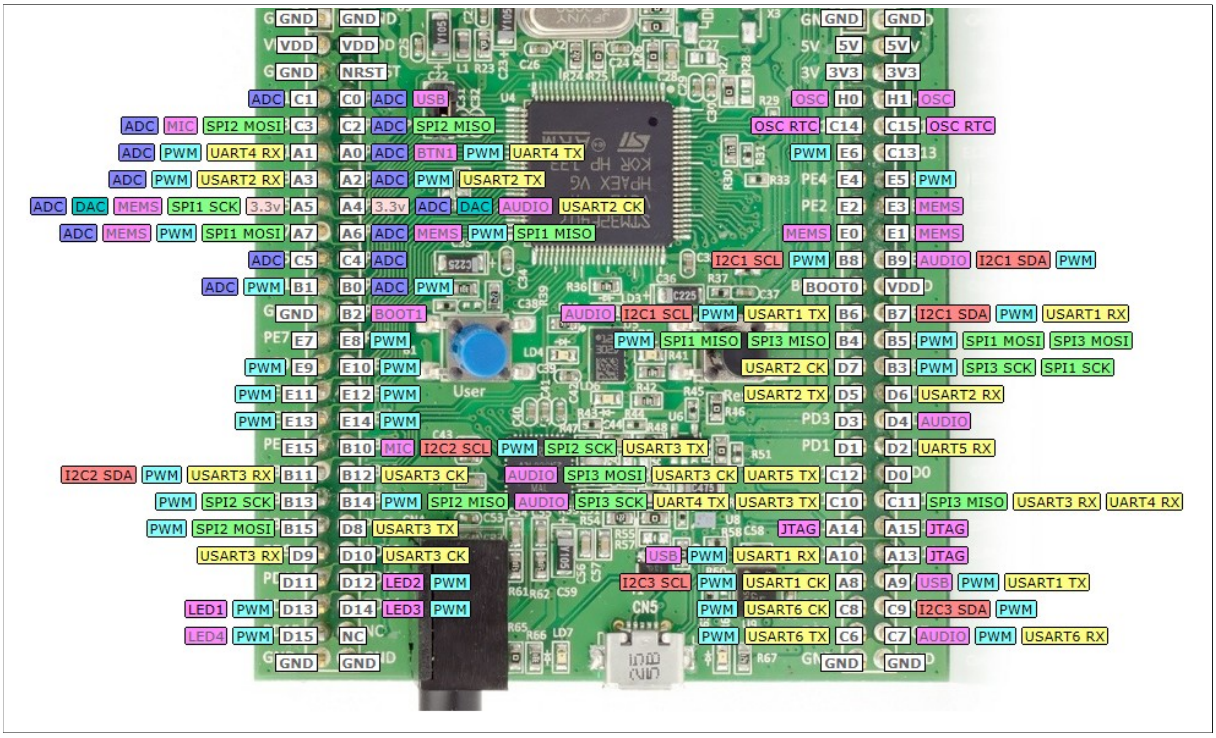
Chip	STM32F407VGT6
Package	LQFP100
RAM	192 kBytes
Flash	1024 kBytes
Speed	168 Mhz
USARTs	6
SPIs	3
I2Cs	3
USB	Yes
DACs	2
SD Card	No

Like this? Please tell your friends, blog, or support us by buying our board!

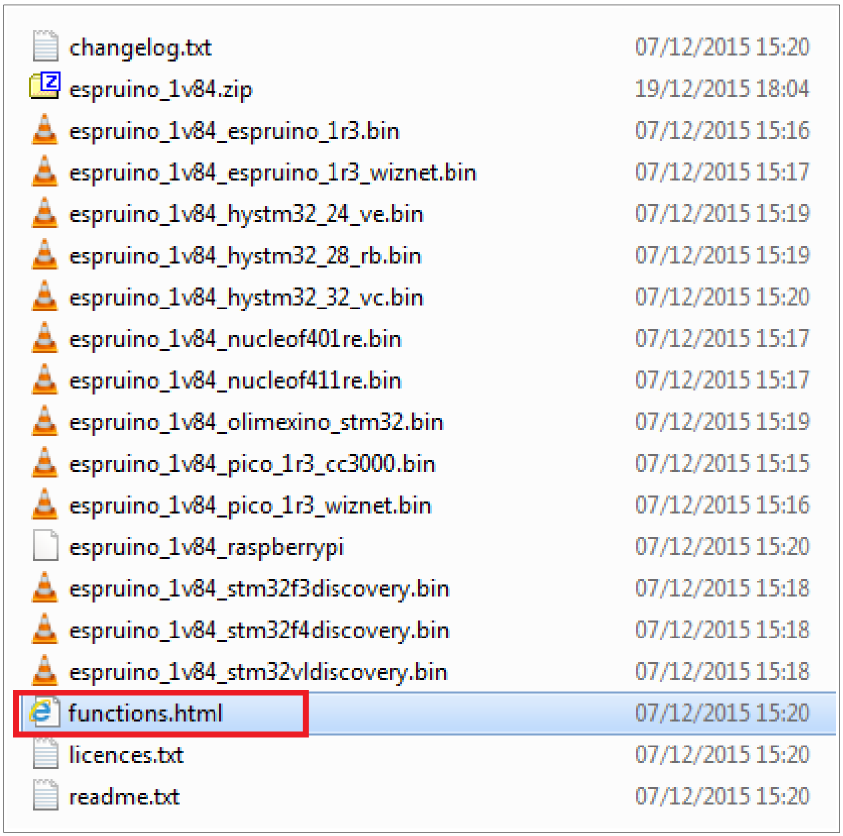
PINOUT

Hover the mouse over a pin function for more information. Clicking in a function will tell you how to use it in Espruino.

- Purple boxes show pins that are used for other functionality on the board. You should avoid using these unless you know that the marked device is not used.
- ! boxes contain extra information about the pin. Hover your mouse over them to see it.
- 3.3v boxes mark pins that are not 5v tolerant (they only take inputs from 0 - 3.3v, not 0 - 5v).
- 3.3 is a 3.3v output from the on-board Voltage regulator.
- GND is ground (0v).
- VBAT is the battery voltage output (see the Espruino Board Reference).
- ADC is an Analog to Digital Converter (for reading analog voltages)
- DAC is a Digital to Analog Converter (for creating analog voltages). This is not available on all boards.
- PWM is for Pulse Width Modulation. This creates analog voltages from a digital output by sending a series of pulses.
- SPI is the 3 wire Serial Peripheral Interface.
- USART is a 2 wire peripheral for Serial Data.
- I2C is the 2 wire Inter-Integrated Circuit bus.


GND GND GND GND
VDD VDD 5V 5V
GND NRST 3V3 3V3
ADC C1 C0 ADC USB OSC H0 H1 OSC
ADC MIC SPI2 MOSI C3 C2 ADC SPI2 MISO OSC RTC C14 C15 OSC RTC
ADC PWM UART4 RX A1 A0 ADC BTN1 PWM UART4 TX PWM E6 C13
ADC PWM USART2 RX A3 A2 ADC PWM USART2 TX E4 E5 PWM
ADC DAC MEMS SPI1 SCK 3.3v A5 A4 3.3v ADC DAC AUDIO USART2 CK E2 E3 MEMS
ADC MEMS PWM SPI1 MOSI A7 A6 ADC MEMS PWM SPI1 MISO MEMS E0 E1 MEMS
ADC C5 C4 ADC I2C1 SCL PWM B8 B9 AUDIO I2C1 SDA PWM
ADC PWM B1 B0 ADC PWM BOOT0 VDD
GND B2 BOOT1 AUDIO I2C1 SCL PWM USART1 TX B6 B7 I2C1 SDA PWM USART1 RX
E7 E8 PWM PWM SPI1 MISO SPI3 MISO B4 B5 PWM SPI1 MOSI SPI3 MOSI
PWM E9 E10 PWM USART2 CK D7 B3 PWM SPI3 SCK SPI1 SCK
PWM E11 E12 PWM USART2 TX D5 D6 USART2 RX
PWM E13 E14 PWM D3 D4 AUDIO
E15 B10 MIC I2C2 SCL PWM SPI2 SCK USART3 TX D1 D2 UART5 RX
I2C2 SDA PWM USART3 RX B11 B12 USART3 CK AUDIO SPI3 MOSI USART3 CK UART5 TX C12 D0
PWM SPI2 SCK B13 B14 PWM SPI2 MISO AUDIO SPI3 SCK UART4 TX USART3 TX C10 C11 SPI3 MISO USART3 RX UART4 RX
PWM SPI2 MOSI B15 D8 USART3 TX JTAG A14 A15 JTAG
USART3 RX D9 D10 USART3 CK USB PWM USART1 RX A10 A13 JTAG
D11 D12 LED2 PWM I2C3 SCL PWM USART1 CK A8 A9 USB PWM USART1 TX
LED1 PWM D13 D14 LED3 PWM PWM USART6 CK C8 C9 I2C3 SDA PWM
LED4 PWM D15 NC PWM USART6 TX C6 C7 AUDIO PWM USART6 RX
GND GND GND GND
User

The manual of the commands is inside the download that we did before, see below.

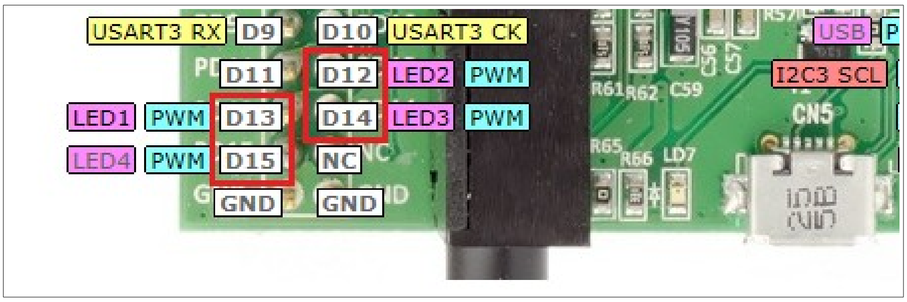


changelog.txt	07/12/2015 15:20
espruino_1v84.zip	19/12/2015 18:04
espruino_1v84_espruino_1r3.bin	07/12/2015 15:16
espruino_1v84_espruino_1r3_wiznet.bin	07/12/2015 15:17
espruino_1v84_hystm32_24_ve.bin	07/12/2015 15:19
espruino_1v84_hystm32_28_rb.bin	07/12/2015 15:19
espruino_1v84_hystm32_32_vc.bin	07/12/2015 15:20
espruino_1v84_nucleof401re.bin	07/12/2015 15:17
espruino_1v84_nucleof411re.bin	07/12/2015 15:17
espruino_1v84_olimexino_stm32.bin	07/12/2015 15:19
espruino_1v84_pico_1r3_cc3000.bin	07/12/2015 15:15
espruino_1v84_pico_1r3_wiznet.bin	07/12/2015 15:16
espruino_1v84_raspberrypi	07/12/2015 15:20
espruino_1v84_stm32f3discovery.bin	07/12/2015 15:18
espruino_1v84_stm32f4discovery.bin	07/12/2015 15:18
espruino_1v84_stm32vldiscovery.bin	07/12/2015 15:18
functions.html	07/12/2015 15:20
licences.txt	07/12/2015 15:20
readme.txt	07/12/2015 15:20

Now we need to blink the green LED.
The LEDs on the STM32F4-Discovery are connected as shown below (see the **manual** of STM32F4-Discovery.

User **LD3**: **orange** LED connected to the I/O **PD13**
User **LD4**: **green** LED connected to the I/O **PD12**
User **LD5**: **red** LED connected to the I/O **PD14**
User **LD6**: **blue** LED connected to the I/O **PD15**

The same LEDs are mapped with different names on the Espruino but, of course, are connected to the same I/O. See below



Try to type the commands below and look the **Blue Led** on the STM32F4-Discovery

```
var on = false
function toggle() { on = !on; digitalWrite(D15, on);}
var i = setInterval(toggle, 500)
```

You must see the Blue Led that flashing.

For stop the flash type the command below:

```
reset()
```

The **reset()** command, **start completely from scratch and wipe out everything you have done**.

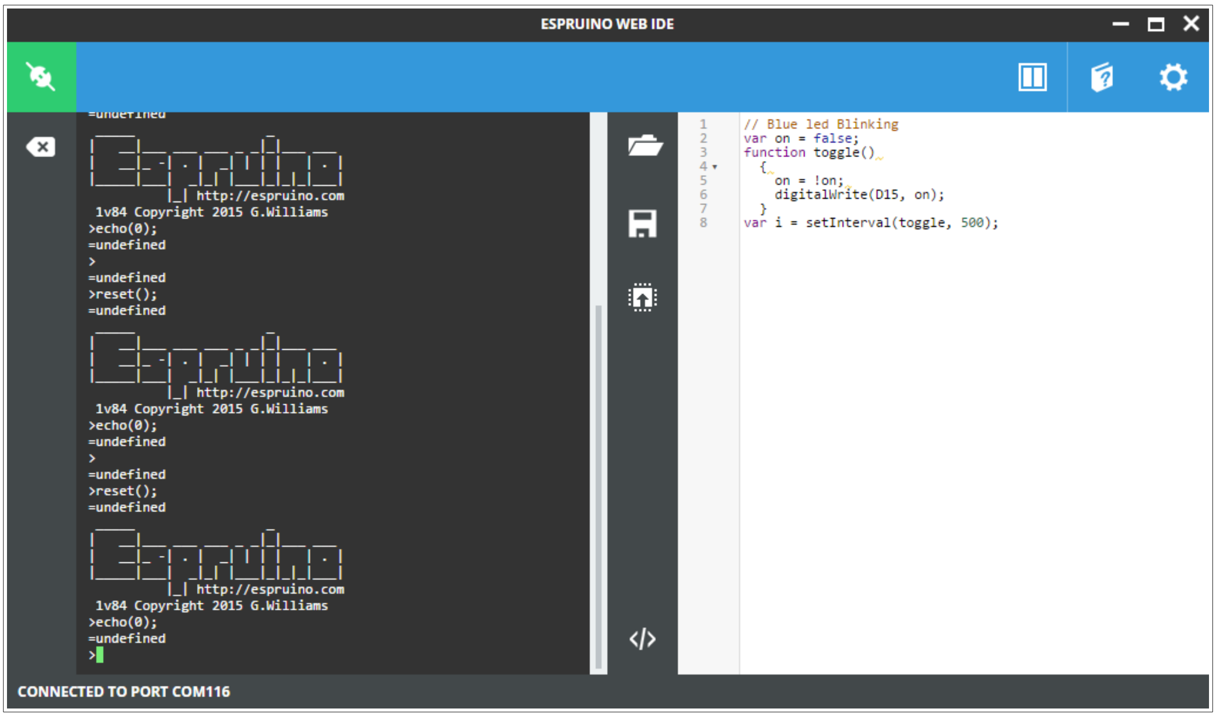
If you want to **execute some code when Espruino starts**, you can use the init event **E.on**
For example this bit of code lights the blue LED each time Espruino starts up:

```
E.on('init', function() { digitalWrite([D15, 2);
setTimeout("digitalWrite(D15, 1);", 1000);
setTimeout("digitalWrite(D15, 0);", 2000);});
```

For more info see the **Quick Start Guide** that is **here**.

IDE for develop on Espruino

For develop SW for Espruino we suggest:
Chrome Web App (it has a bunch of extra features, including firmware updates).
See below.
For more info see: **Quick Start Guide** that is **here**.



```
=undefined
 _____                 _
|   __|___ ___ ___ _ _|_|___ ___
|   __|_ -| . |  _| | | |   | . |
|_____|___|  _|_| |___|_|_|_|___|
          |_| http://espruino.com
 1v84 Copyright 2015 G.Williams
>echo(0);
=undefined
>
=undefined
>reset();
=undefined
 _____                 _
|   __|___ ___ ___ _ _|_|___ ___
|   __|_ -| . |  _| | | |   | . |
|_____|___|  _|_| |___|_|_|_|___|
          |_| http://espruino.com
 1v84 Copyright 2015 G.Williams
>echo(0);
=undefined
>
=undefined
>reset();
=undefined
 _____                 _
|   __|___ ___ ___ _ _|_|___ ___
|   __|_ -| . |  _| | | |   | . |
|_____|___|  _|_| |___|_|_|_|___|
          |_| http://espruino.com
 1v84 Copyright 2015 G.Williams
>echo(0);
=undefined
>
```

```
// Blue led Blinking
var on = false;
function toggle()
  {
    on = !on;
    digitalWrite(D15, on);
  }
var i = setInterval(toggle, 500);
```

CONNECTED TO PORT COM116

The blinking Led see before, now is in this format, more easy to read.

```
var on = false;
function toggle()
  {
    on = !on;
    digitalWrite(D15, on);
  }
var i = setInterval(toggle, 500);
```

How to save & remove a program on flash

Last but not least, is the command to save in flash your SW.
For do this use the command: **save()**

For erase all the content of flash use the command: **reset()**

An advanced example

This example is ready to use on **STM32F4-Discovery** and does this:

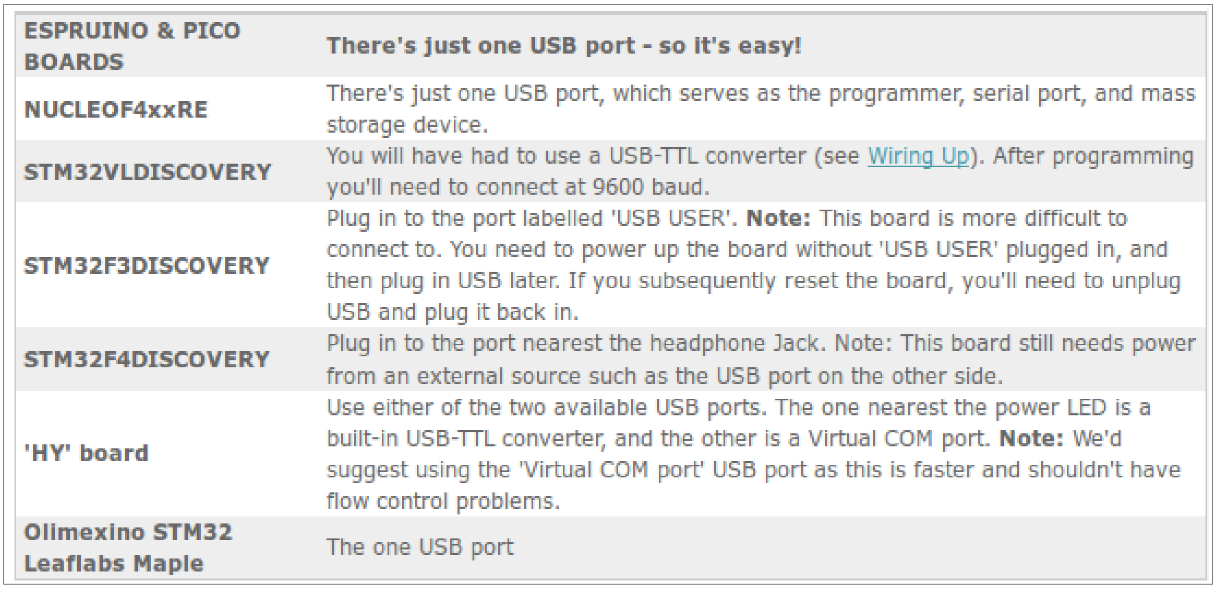
- Drive the LEDs
- Change the LEDs status by press on Blue button.
  The Blue button is under Interrupt on rising edge and has a debounce (30mS).
- There is a TimeOut that reset the LEDs after 5sec
- Send to PC the status of the LEDs

```
// SetUp the LEDs
LED1.write(0);
LED2.write(0);
LED3.write(0);
LED4.write(0);

// keep track of the next LED
var next_LED = 1;

// keep track of the ID, see later
var timeout_ID;

function swap()
  {
  // remove the timeout to turn of all LEDs when the user pressed the
button
  if (timeout_ID !== undefined)
    {
      clearTimeout(timeout_ID);
    }
  // determine which LED to turn on/off
  switch(next_LED) {
    case 1:
      print("LED Orange is ON");
      LED4.write(0);
      LED1.write(1);
      LED2.write(0);
      LED3.write(0);
      break;
    case 2:
      print("LED Green is ON");
      LED2.write(1);
      LED1.write(0);
      break;
    case 3:
      print("LED Red is ON");
      LED3.write(1);
      LED2.write(0);
      break;
    case 4:
      print("LED Blue is ON");
      LED4.write(1);
      LED3.write(0);
      break;
  }

  // determine the next LED to turn on
  next_LED = Math.wrap(next_LED, 4) + 1;

  // prepare a timeout to turn off all LEDs after a while
  // we capture the ID here, so that we can use it in a next call to this
function
  timeout_ID = setTimeout(function () { LED1.write(0); LED2.write(0);
LED3.write(0); LED4.write(0); print("All LEDs are OFF"); timeout_ID =
undefined; }, 5000);
}

// Monitor the status of the Blue Button
setWatch(swap, BTN1, {repeat:true, edge:"rising", debounce:30});
```

Wiring Up for the boards that don't have USB connector

Every supported board except the **STM32VLDISCOVERY** has a USB serial port, so most users can skip this step.
If you have an STM32VLDISCOVERY, or you want to use Bluetooth, or to connect to a Raspberry Pi using serial (rather than USB), please see the **Wiring Up page**.

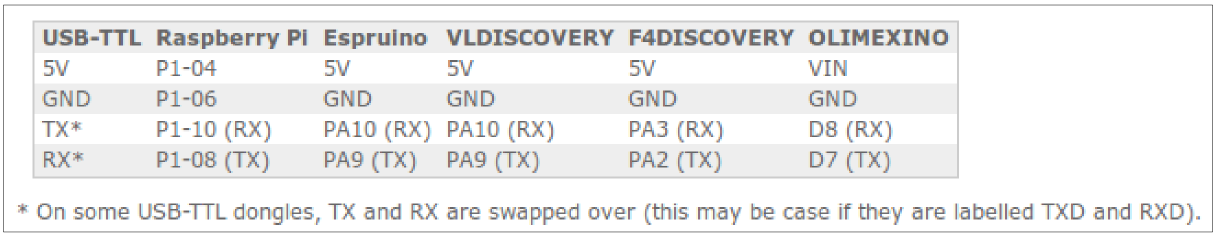
You communicate with Espruino using a **Terminal Emulator** over a **Serial port**.
Most Espruino devices can emulate a Serial port over USB, so when you plug these in to your PC or Mac the Operating System will automatically detect them. All you need to do is find out what the Operating System has 'called' the serial port that has been created.
For more info see this **page**.



ESPRUINO & PICO BOARDS	**There's just one USB port - so it's easy!**
NUCLEOF4xxRE	There's just one USB port, which serves as the programmer, serial port, and mass storage device.
STM32VLDISCOVERY	You will have had to use a USB-TTL converter (see Wiring Up). After programming you'll need to connect at 9600 baud.
STM32F3DISCOVERY	Plug in to the port labelled 'USB USER'. **Note:** This board is more difficult to connect to. You need to power up the board without 'USB USER' plugged in, and then plug in USB later. If you subsequently reset the board, you'll need to unplug USB and plug it back in.
STM32F4DISCOVERY	Plug in to the port nearest the headphone Jack. Note: This board still needs power from an external source such as the USB port on the other side.
'HY' board	Use either of the two available USB ports. The one nearest the power LED is a built-in USB-TTL converter, and the other is a Virtual COM port. **Note:** We'd suggest using the 'Virtual COM port' USB port as this is faster and shouldn't have flow control problems.
Olimexino STM32 Leaflabs Maple	The one USB port

The USB-SerialPort converter recommended is this:
USB to TTL Converter Module with Built-in CP2102

To connect any of the boards up via serial, the pins you need are:



USB-TTL	Raspberry Pi	Espruino	VLDISCOVERY	F4DISCOVERY	OLIMEXINO
5V	P1-04	5V	5V	5V	VIN
GND	P1-06	GND	GND	GND	GND
TX*	P1-10 (RX)	PA10 (RX)	PA10 (RX)	PA3 (RX)	D8 (RX)
RX*	P1-08 (TX)	PA9 (TX)	PA9 (TX)	PA2 (TX)	D7 (TX)

* On some USB-TTL dongles, TX and RX are swapped over (this may be case if they are labelled TXD and RXD).

Examples

- Quick Start Guide is **here**.
- How to install Espruino on the STM32F4DISCOVERY board (video) is **here**.
- You can see a lot of examples **here**.
- Espruino IDE (video) is **here**.