

SEED TECHNOLOGY INC (SEEDUINO)

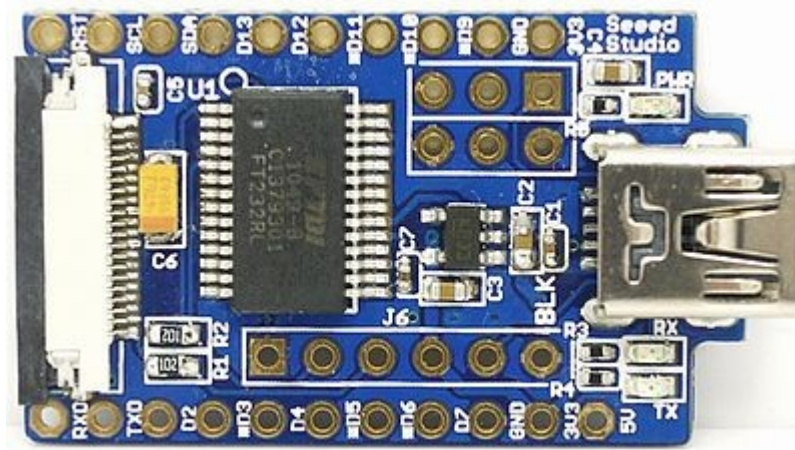
Seeeduino Frame - UartSBs

Model: PGM31058P

Introduction

UartSB Frame is a USB based programming adapter for [Seeeduino Film](#) equipped with a FPC socket. The integrated USB UART IC **FT232RL** can be used for programming or communicating with MCUs with 3.3V logic. It also doubles as a "FT232RL" Low Voltage(3.3V) Breakout board. **UartSB Frame** provides breakouts for the bit-bang mode pins of **FT232RL** as well. This Bit-bang mode pins (8 I/O pins) can be used as a replacement for applications involving PC parallel port which is scarce now a days.

Model: [PGM31058P](#)



Features

- USB 2.0 compatible Serial Interface.
- 20 Pin FPC connector for attaching [Seeeduino Film](#)
- 3.3V compatible I/Os.
- 3.3V and 5V dual power outputs.
- Bit-Bang mode ready (8 Serial I/Os or as SPI).
- LEDs for Tx, Rx and Power.

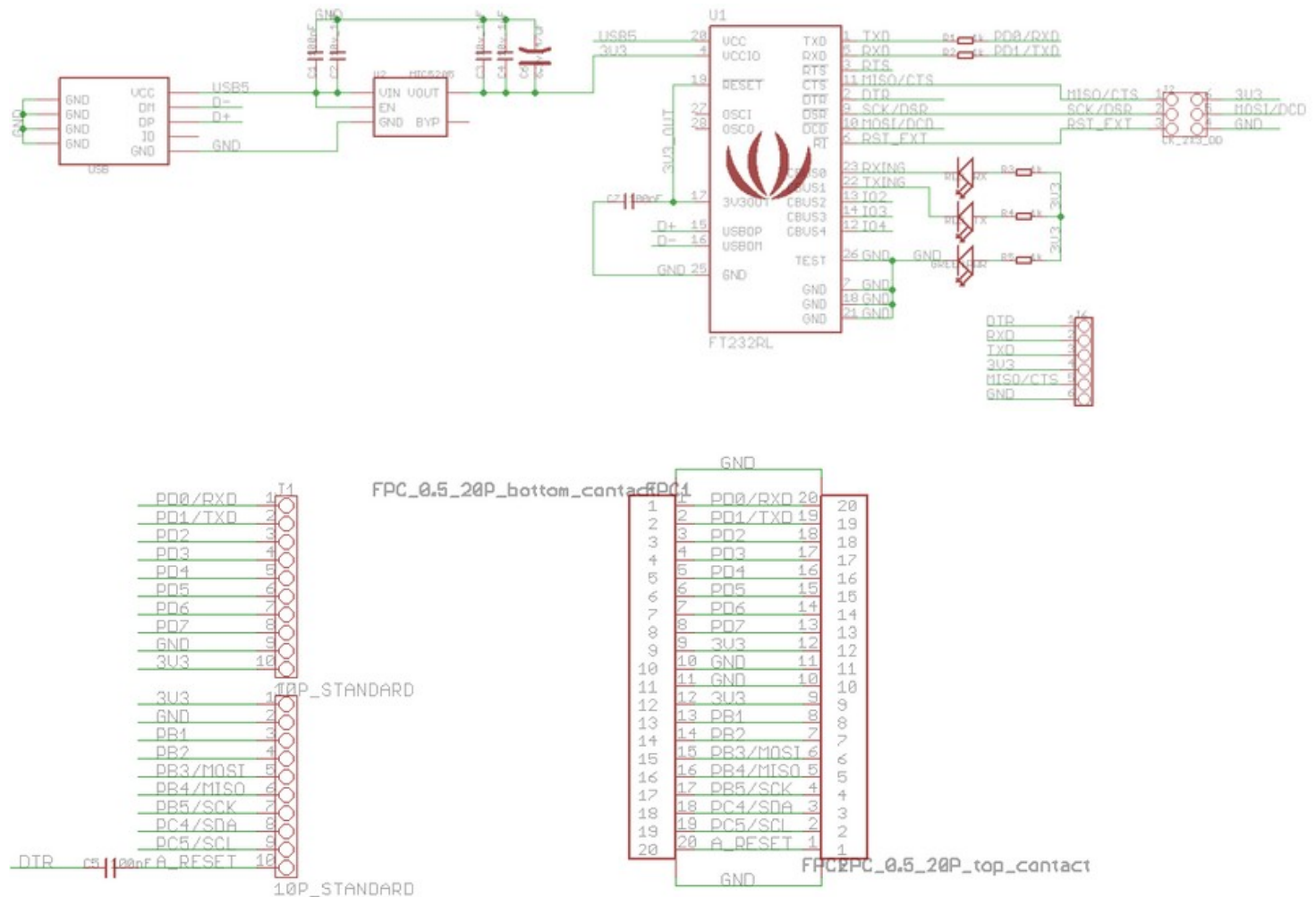
Applications Ideas

- Programmer for [Seeeduino Film](#) and 3.3V compatible boards.
- USB to Serial adapter for communicating with 3.3V level Serial devices.
- Programmer for micro-controllers / CPLDs using ISP (in-circuit-serial-programming).
- 3.3V / 5V power source for a breadboard MCU applications.
- The 5V supply can be used to charge the Seeeduino Film battery. (Do not use FPC connection for this application)
- Endless possibilities as a USB based parallel device in FT232RL bit-bang mode - Known to work as AVR-ISP, Slowspeed JTAG with OpenOCD and as I2C

Cautions

- **Connect UartSB Frame to Seeeduino Film only as shown in [Seeeduino Film Programmer](#) section.**
- **Do not use FFC (Flexible Flat Cable) to directly connect UartSB Frame to Seeeduino Film. This would reverse the 20 pin FPC connection.**
- **Handle the 20 pin FPC connector with care.**
- **Use UartSB Frame only with 3.3v I/Os.**

Schematic



Specifications

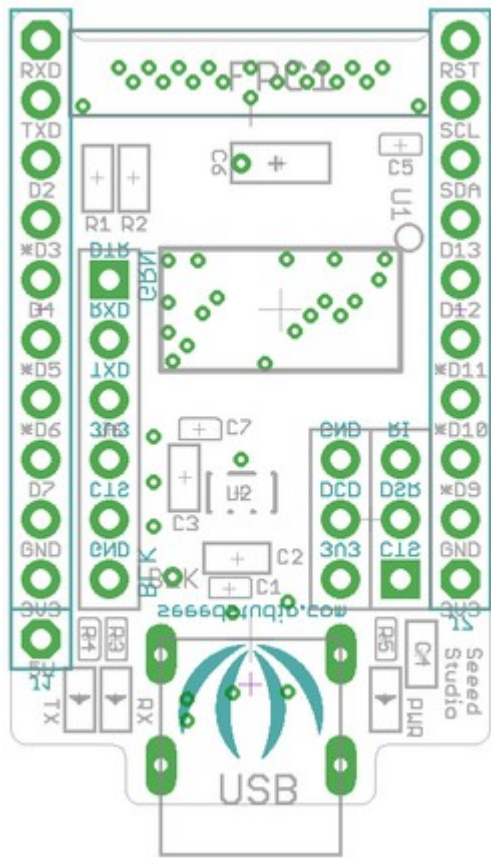
Key Specifications

Microprocessor	FT232RL
PCB size	35mm x 20mm
Indicators	POWER - Green LED, RX and TX - Red LEDs
Power supply	3.3V and 5V DC
Interface	Mini-B USB, 2.54mm pitch pin header
Seeeduino Programming Port	20 Pin FPC socket
Connectivity	USB
Communication Protocol	UART, Bit Bang I/O, SPI

Electrical Characteristics

Specification	Minimum	Typical	Maximum	Unit
Input voltage	-	5	5	Vdc
Current Consumption	-	-	500	mA
Output voltage	3.3	-	5	Vdc

Pin definition and Rating

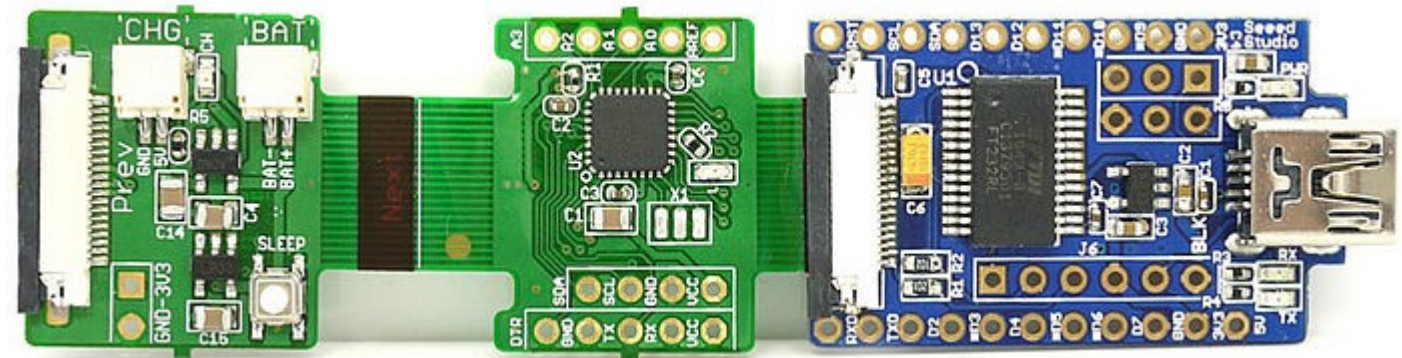


Pad Name	Description
GND	Ground pin
3v3	3.3V Power Supply
5V	5V Power supply from USB Port
RXD	UART Data input
TXD	UART Data output
CTS	UART Clear to Send Signal
RTS	UART Request to Send Signal
DTR	UART Data Terminal Ready Signal
DCD	UART Data Carrier Detect Signal
DSR	UART Data Set Ready Signal

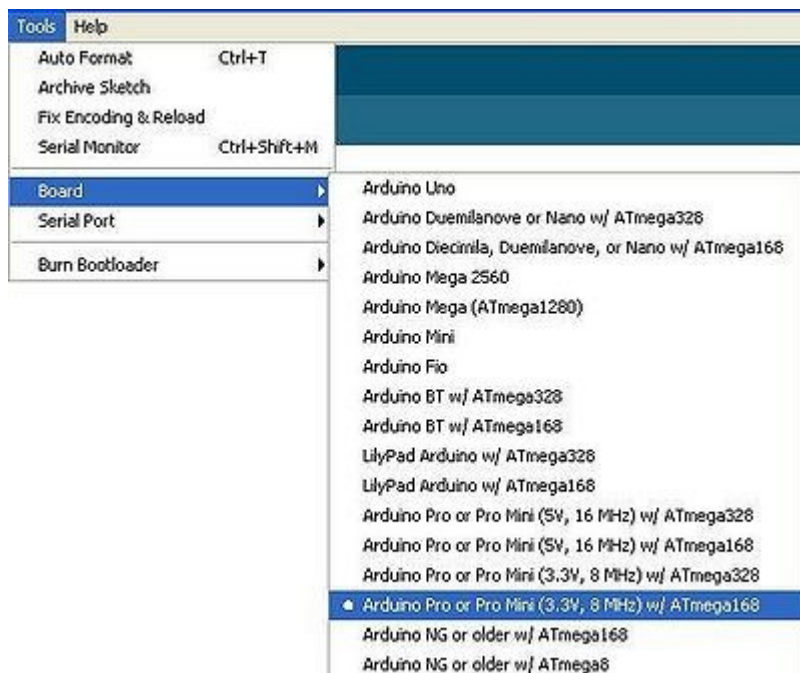
Usage**Seeeduino Film Programmer**

UartSB Frame is designed exclusively for flashing [Seeeduino Film](#). It works as a USB to Serial Port (COM port) interface when connected to PC. This kind of configuration can be used to communicate with a MCU serial port. Also useful to program a MCU which support UART based ISP.

- If the FT232RL drivers were not installed, follow the procedure in next section (configuring UartSB Frame).
- Connect the Seeeduino Film to UartSB Frame as shown below :



- Set the board type to Arduino Pro or Pro Mini (3.3V, 8MHz) w/ATMega168 in Arduino IDE



- Upload the sketch.

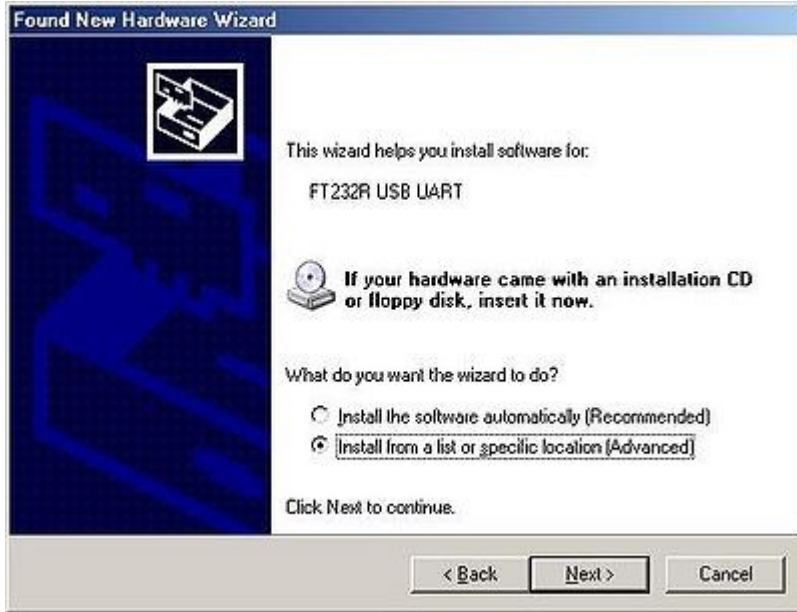
Configuring UartSB Frame in Windows

- In Windows OS, the first time you plug in the device, you might be asked for the driver.

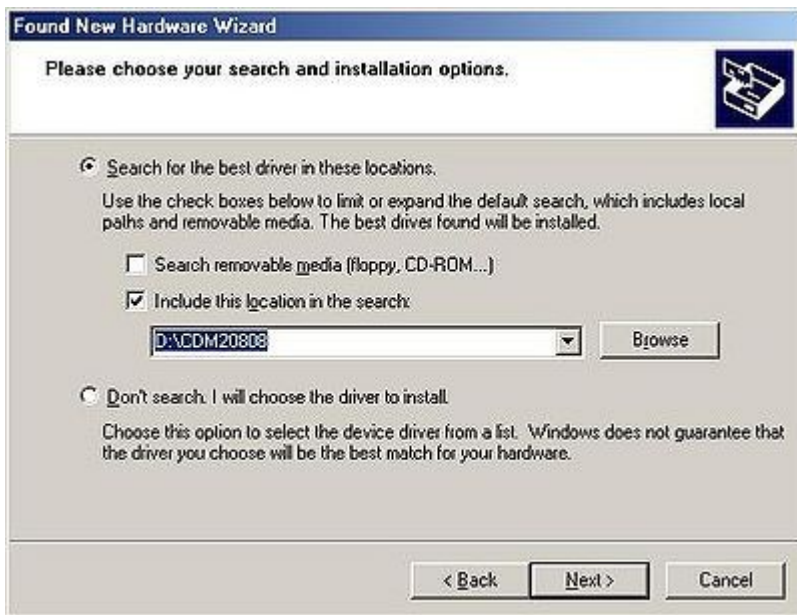


Download and install the **Virtual COM port** driver from FTDI website :

- A wizard opens for installing the driver. Choose "Install from a list or specific location"



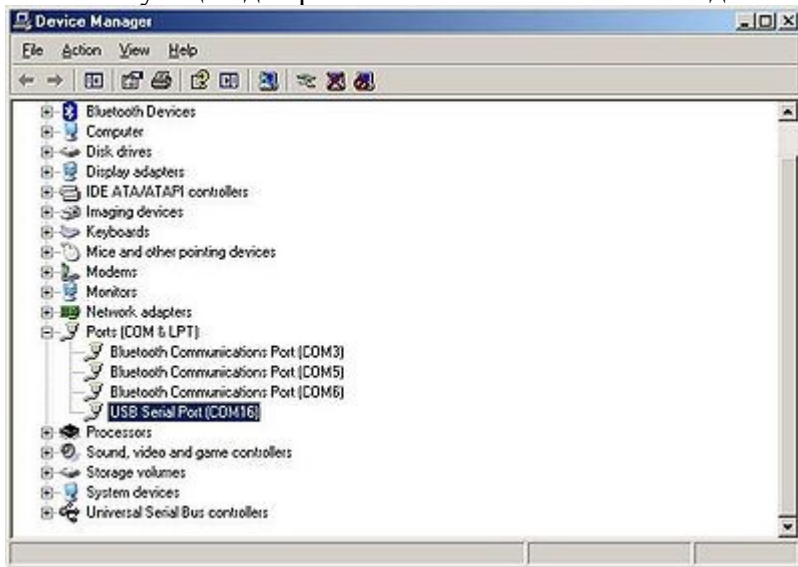
- Choose the downloaded driver path



- If you have downloaded unsigned driver, the following Window appears. Just click "Continue Anyway"



- **UartSB Frame** driver is successfully installed. Windows assigns a **COM** port name to **FT232RL** like **COM10**, **COM11** etc... Please check the exact name in Device Manager. In this case "**COM16**" is assigned for UartSB Frame



Configuring UartSB Frame in GNU/Linux

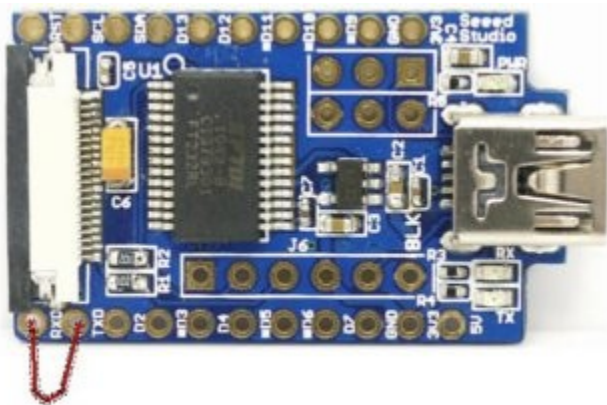
All modern **GNU/Linux OS** comes with FT232RL drivers. To check if UartSB Frame is detected, Issue a **lsusb** command. An output similar to below should appear.

```
File Edit View Terminal Help
@-desktop:~$ lsusb
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 003: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 002: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth Dongle (HCI mode)
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 002: ID 06a2:0033 Toppro Technology, Inc. USB Mouse
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
@-desktop:~$
```

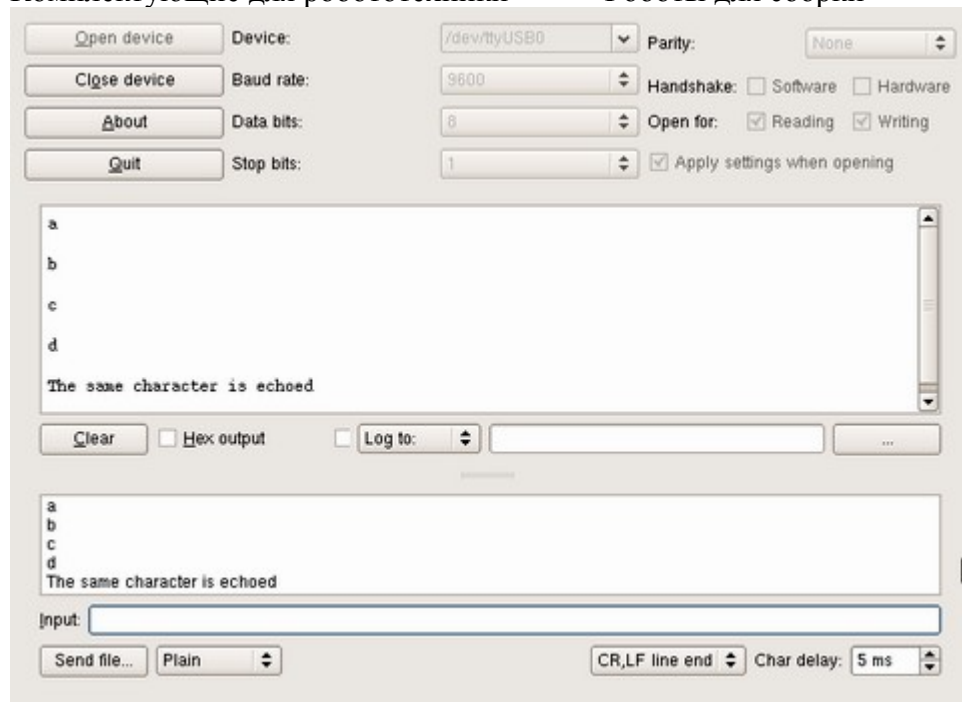
GNU/Linux assigns **/dev/ttyUSB0**, **/dev/ttyUSB1** etc... as device name.

To verify the working of serial port connect **TxD** and **RxD** pins of UartSB Frame and use a terminal application like **cutecom** to configure the device parameters as shown below.

Baudrate:9600, Data bits:8, Stop bits:None and no Handshake



Any character typed in terminal would be echoed back as shown.

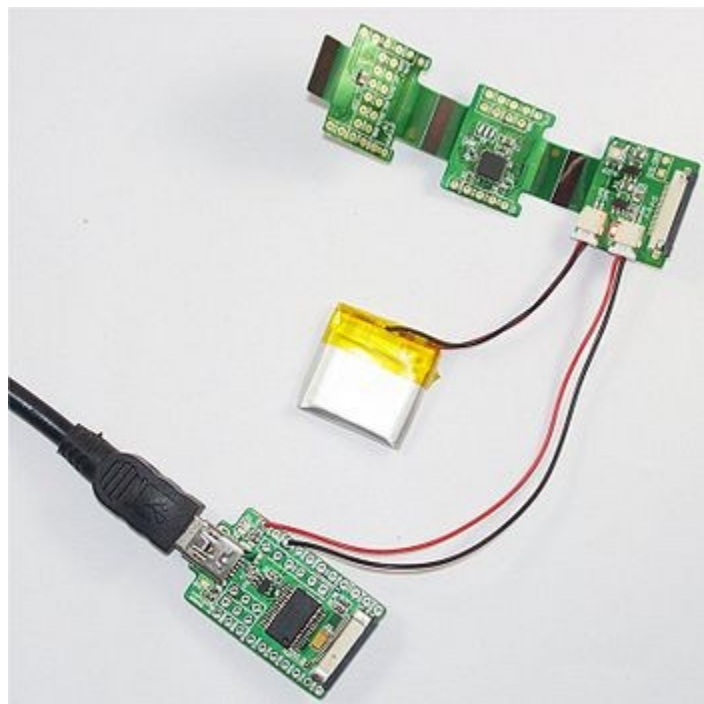


The same functionality can be verified in **Windows - Hyperterminal** as well.

Battery Charger for Seeeduino Film

The USB 5v supply available in **UartSB Frame** breakout can be used to charge the battery of [Seeeduino Film](#).

- Solder the [Seeeduino Film](#) Power-Connector to 5V and GND of **UartSB Frame**.
- Connect Power connector to **CHG** jack.
- Connect the Battery to Seeeduino Film **BAT** jack.
- Power on the USB.



BitBang Mode

An nice feature of UartSB Frame compared to other FT232RL based USB-Serial devices is that all the Bit-Bang I/Os are brought to the header pins.

- Asynchronous Bit-Bang mode

Any data written to the device is clocked to the configured output pins. The rate of data transfer is configured based on baud rate generator. In this mode any of 8 I/O lines can be configured as input or ouput.

- Synchronous Bit-Bang mode

In this mode data is sent synchronously. An input is read before an output byte is sent to the device. Hence to read input, a write operation must be performed.

- CBUS Bit-Bang Mode

This is a special mode which requires reprogramming of FT232RL EEPROM. This uses signals **C0 - C3**.

Bit-Bang Mode of **FT232RL** is well documented in the application note[\[1\]](#)

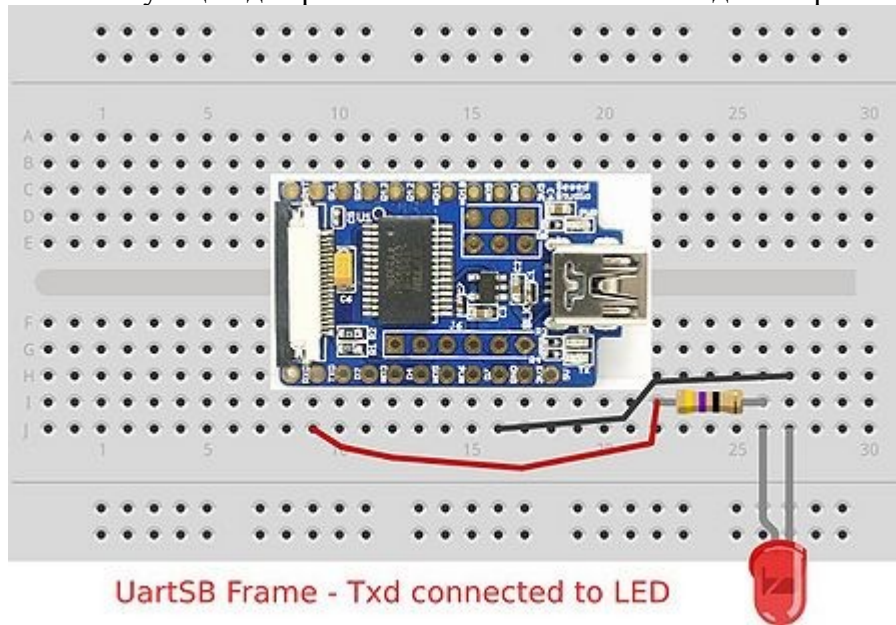
Table: Bit-Bang I/O Mapping

UartSB Frame Signal	BitBang I/O Signal
TxD	D0
RxD	D1
RTS	D2
CTS	D3
DTR	D4
DSR	D5
DCD	D6
RI	D7

BitBang Mode Operation:

A simple Asynchronous Bit-Bang mode operation is demonstrated in the below breadboard arrangement in which DTR (D4) pin is connected to an LED. The LED blinking rate is controlled by the PC side application program.

LED Blinky Circuit



This above image is made with Fritzing.org

Bit Bang I/O Pins (ISP header) shown at Bottom Layer



Bit-Bang mode requires special FTDI direct driver called [D2XX](#). This driver needs to be installed after removing the Virtual COM Port driver of FT232RL chip. In GNU/Linux this driver runs in kernel mode. As an alternative to D2XX an Free Open Source driver [libFTDI](#) is available. This works in Windows, GNU/Linux and Mac OS. It runs in user mode in GNU/Linux. Hence no need to remove the existing FT232RL driver.

libFTDI

<http://www.intra2net.com/en/developer/libftdi/>

libUSB libFTDI required libusb which can be downloaded from

<http://www.libusb.org/>

The below example code can be compiled in the similar way as that of libFTDI example files. An easy way is to copy the contents of the below code to one of the existing example .c file and build the whole driver using

```
./configure
```

```
make
```

Code Example

```
/*
Blinky.C: UartSBee v3.1 (FT232RL) Bit-Bang mode - Blinky.

Circuit:
Connect TXD to Anode of LED, Connect one end of resistor to GND and other end
to Cathode of the LED
*/

#ifdef __WIN32__
#define sleep(x) Sleep(x)
#endif

// 8 bit pin mask for I/O pin
#define TXD 0x01
#define RXD 0x02
#define RTS 0x04
#define CTS 0x08
#define DTR 0x10
#define DSR 0x20
#define DCD 0x40
#define RI 0x80

#include <stdio.h>;
#include <ftdi.h>;

int main()
{
    unsigned char ouputState = 0;
    struct ftdi_context ftdic;

    /* 1. Initialize ftdi device context */
    ftdi_init(&ftdic);

    /* 2. Open the device based of VID/PID pair */

    if(ftdi_usb_open(&ftdic, 0x0403, 0x6001) &lt; 0)
    {
        printf("Unable to open UartSBs Frame");
        return 1;
    }

    /* 3. Enable Bit-Bang mode with for DTR line */
    ftdi_set_bitmode(&ftdic, TXD, BITMODE_BITBANG);

    /* 4. Blink LED every 1 second */
    while(1) {
        ouputState ^= TXD;
        ftdi_write_data(&ftdic, &ouputState, 1);
        sleep(1);
    }
}
```

FT232RL Bit-Bang mode can be used to construct AVR ISP, JTAG, SPI and I2C Ports. Refer the external links.

Image Gallery

UartSB Frame Programming Seeeduino Film (Do not connect Battery)

FAQ

Please list your question here:

Support

If you have questions or other better design ideas, you can go to our [forum](#) or [wish](#) to discuss.

Document Version Tracker

Revision	Description	Editor	Release Date
V0.9b	Detailed Application Notes	Visweswara R	22 Feb 2010

Hardware Version Tracker

Revision	Description	Release Date
UartSBs Frame	0.9b	Feb 22, 2010

Bug Tracker

Bug Tracker is the place you can publish any bugs you think you might have found during use. Please write down what you have to say, your answers will help us improve our products.

Additional Idea

The Additional Idea is the place to write your project ideas about this product, or other usages you've found. Or you can write them on Projects page.

Resources

- [Schematic and Board Files in Eagle](#)
- [PDF Schematic](#)

How to buy

Click here to buy [UartSB Frame](#)

See Also

- [Frame Series](#)
- [UartSBee](#)

Licensing

This documentation is licensed under the Creative Commons [Attribution-ShareAlike License 3.0](#). Source code and libraries are licensed under [GPL/LGPL](#), see source code files for details.

External Links

- [\[2\]FTDI FT232RL product Page](#)
- [\http://www.ftdichip.com/Drivers/VCP.htm FTDI Virtual COM Port (VCP) drivers
- [FTDI D2XX drivers](#)
- [FTDI Bit-Bang mode application note](#)

Open Source Drivers

- [libFTDI](#)
- [libUSB](#)

Other Sources of FT232RL application information

- [Hackaday - Introduction to bit-bang mode](#)
- [FT232R JTAG implementation with OpenOCD](#)
- [FT232R SPI Bitbang Mode example](#)
- [Flashing Arduino with FT232R bitbang mode](#)