Demokit RSB1-Z and Smart Player v1.1

Quick start

Document reference	UG_RSB1-Z_SP_Software
Date	17. Jan. 2018
Revision	Rev 0.2

Written by	Kevin Toutain	17. Jan. 2018
Approved by	Rémi Nieuwjaer	2. Feb. 2021

Table of Contents

Abbreviation table

OS	Operating System
RPi	Raspberry Pi
NUC	Non Uniformity Correction

1 Requirements

The RSB1-Z digi board can be provided with a Raspberry Pi 3 and a Touch screen with the RSB1-Z Smart Player software included.

If the demo kit was sent by FullScale, the following materials has to be included:

- 1 Raspberry Pi 3 Model B
- 1 Micro SD card 16Go (with Linux Raspbian OS installed)
- 1 Touchscreen 7” 800*480 pixels
- 1 Power Supply for Raspberry Pi 5V / 2.5A
- 1 RSB1-Z digi board (1 emitter and 1 receiver)
- 1 K035 proxy board (with ULIS Micro80 Gen2 sensor)
- 1 RJ45 câble

Illustration 1: RSB1-Z Demo kit with Raspberry Pi

2 Powering on the demo kit

To power up the RPi, the power supply can be connected directly on the RPi or on the touchscreen. The powering and tactile option will be shared between the two cards using the red, yellow, green and black wires.

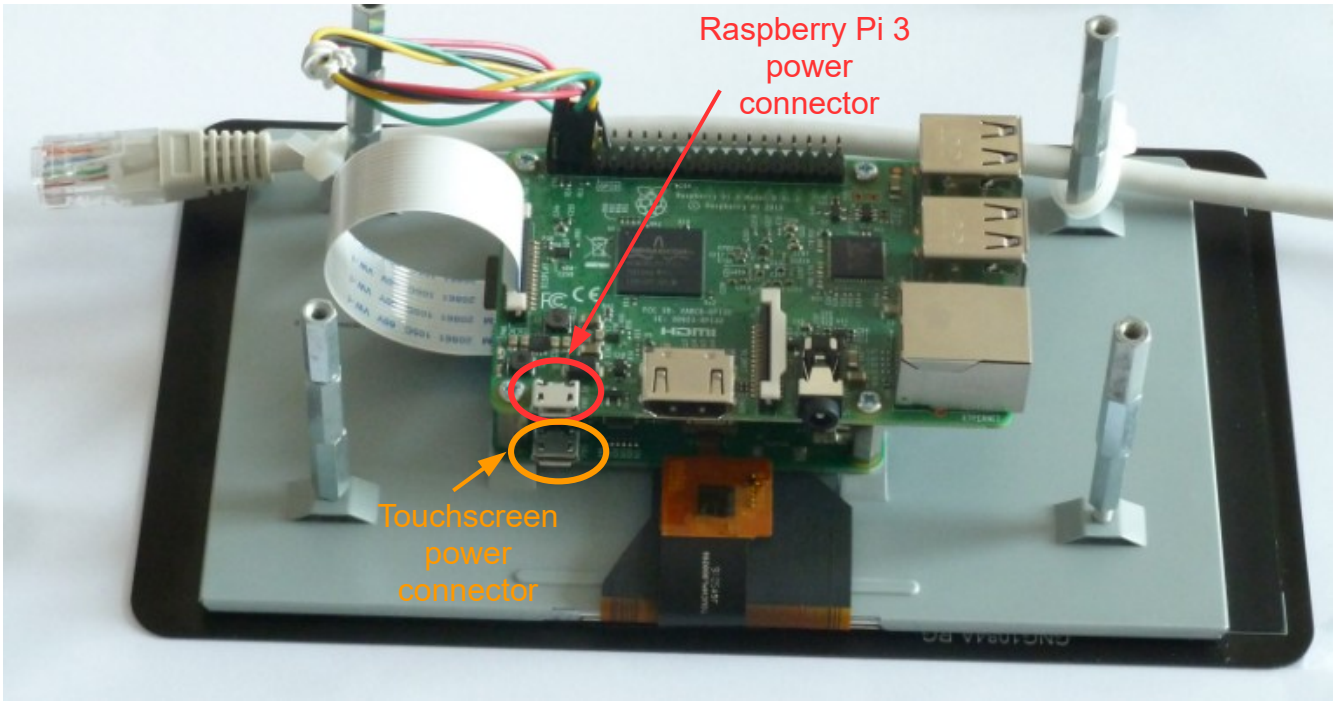

Illustration 2: Raspberry Pi 3 and touchscreen card

CAUTION: The RPi needs at least a power supply which deliver 2A to work correctly.

Once the Raspberry Pi has correctly booted, the following screen has to be display.

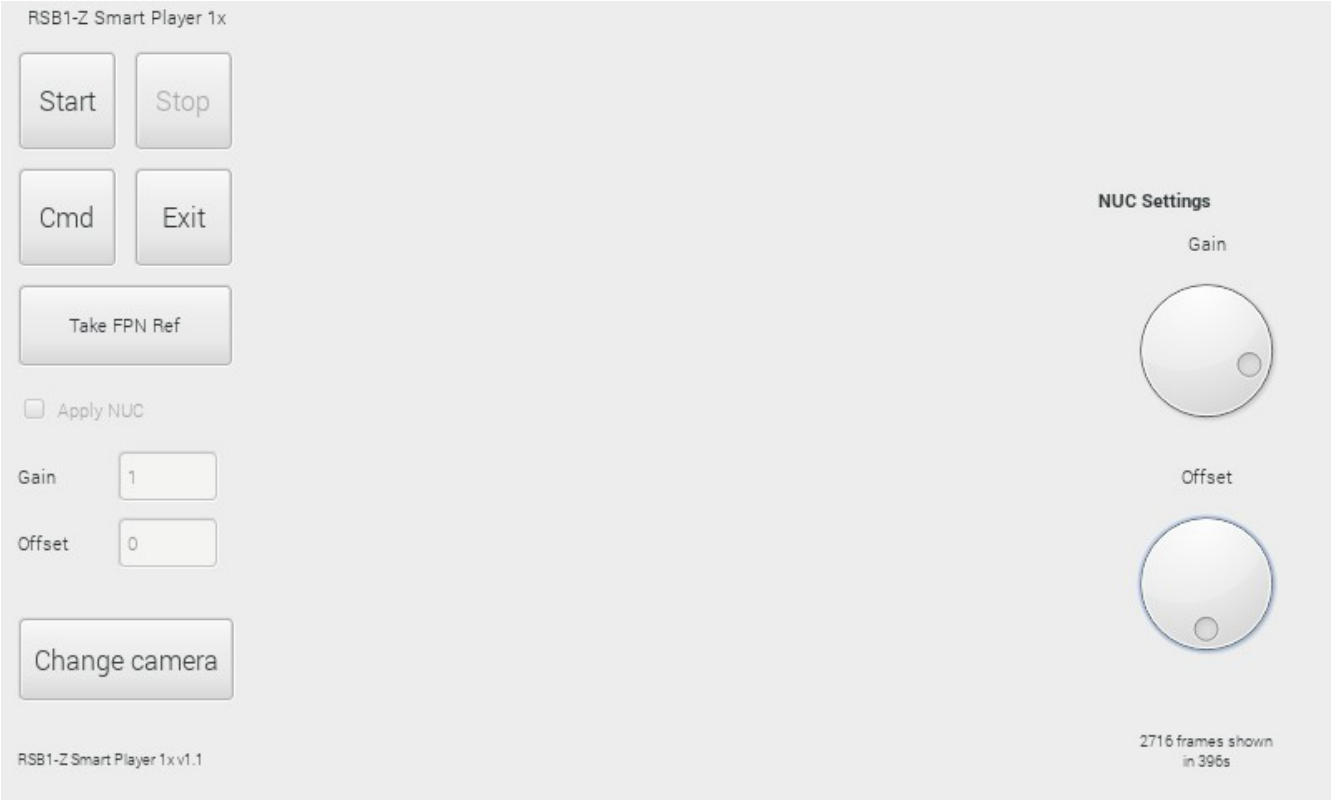

Illustration 3: Raspberry Pi 3 desktop

3 Emitter board

3.1 Description

The Emitter board samples data from the proxy board, and outputs them through the RJ45 cable (using RS485 protocol) to the computer.

CAREFUL: Do not use the Emitter board as an Ethernet peripheral.

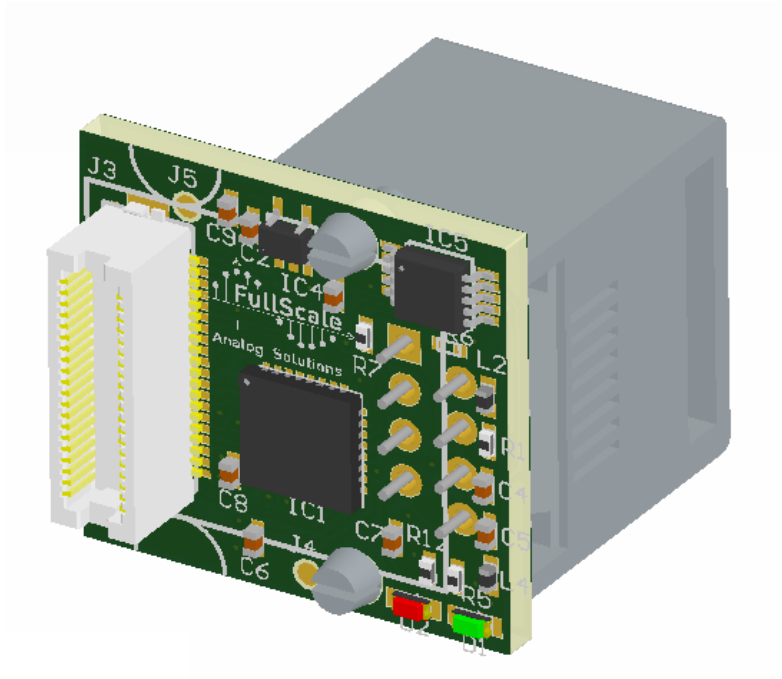

Illustration 4: RSB1-Z Emitter board 3D model

When this board is powered on, the green LED lights up.

3.2 Pinout

The RJ45 connector is used to send the data through the RS485 protocol. The RJ45 connection is wired as the following:

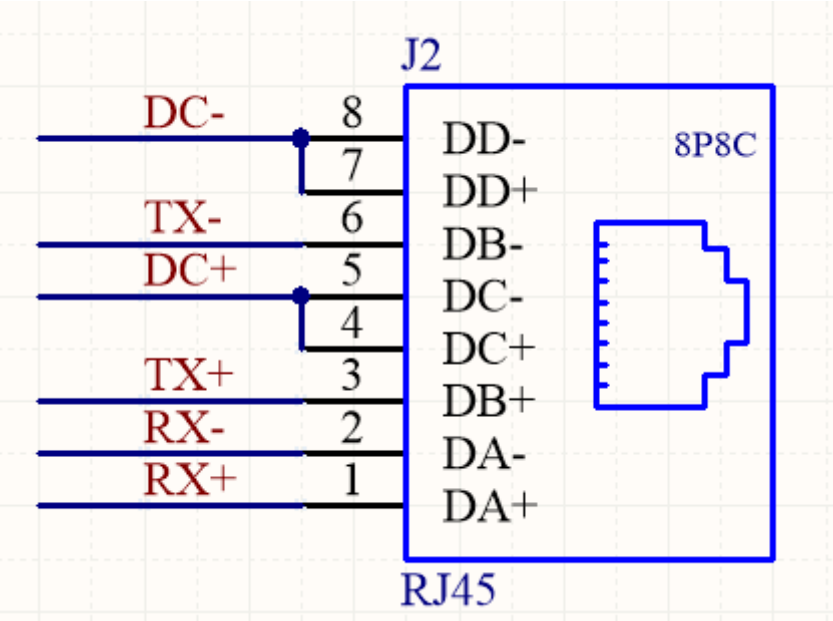

Illustration 5: RJ45 pinout on the Emitter board

The Emitter is supplied by the Receiver board using the RJ45 pins 8-7 and 5-4.

3.3 RS485 configuration

The Emitter is configured to work with the RS485 protocol using the following parameters:

- Speed: 921,600 bauds
- Parity: None
- Data bits: 8 bits
- Stop bit(s): 1 bit
- Flow control: None

3.4 Sending data protocol

The Emitter transfer 80*84 pixels frames with a maximum frame rate of 6.8FPS.

Two types of data are sent from the Emitter: Commands and Pixel data

3.4.1 Command data

Commands are used to indicates when a frame starts, when it ends and if a SPI command has been received.

The command format is the following one:

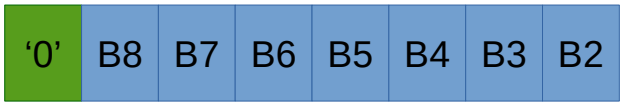

Illustration 6: Command data format

The MSB is always set to '0' when a command is sent.

The following list shows the commands used:

- "Start of Frame" = 0x02;
- "End of Frame" = 0x03;
- "Acknowledge" = 0x06

The "Acknowledge" command is optional and is sent only if a SPI command has been received and processed.

3.4.2 Pixel data

The Emitter manages the 14 MSB of each pixels. Those bits are transferred to the Receiver through the RJ45 cable.

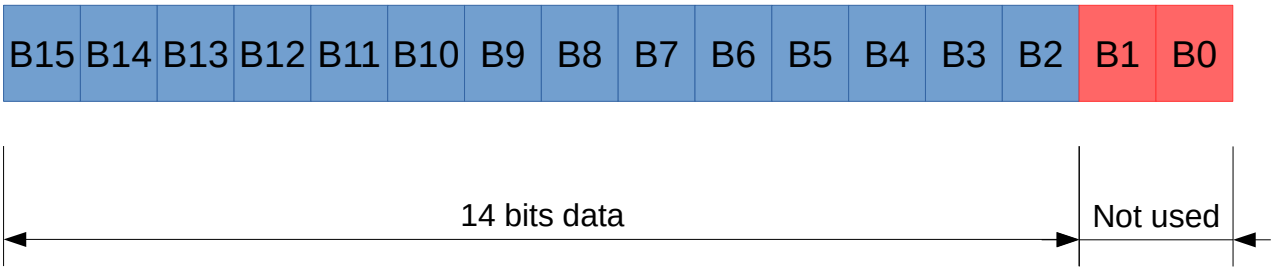

Illustration 7: Pixel data bits used by the RSB1-Z

Data sent from the Emitter are in the following format:

Each pixel data is sent in two bytes (The first one for the MSB (15 → 9) and the second for the LSB (8 → 2))

Each pixel data has its MSB set to '1'.

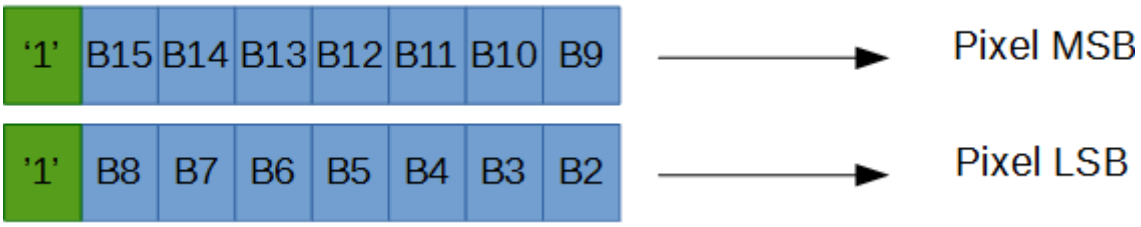

Illustration 8: Pixel data bits arrangement

3.4.3 Checksum data

To check the data integrity, the RSB1-Z calculates a checksum and send it after the “End of Frame” character.

The checksum is stored on 32 bits and is equal to:

$$CKSUM = Pix\,0_{MSB} + Pix\,0_{LSB} + Pix\,1_{MSB} + Pix\,1_{LSB} + \ldots + Pix\,6719_{MSB} + Pix\,6719_{LSB}$$

Information:

Only the 7 pixel data bits are used to calculate the checksum (the bits that indicates it is a pixel data is not included in the formula). A mask with 0x7F has to be use to get the right value.

On the illustration below, the checksum bits arrangement is shown:

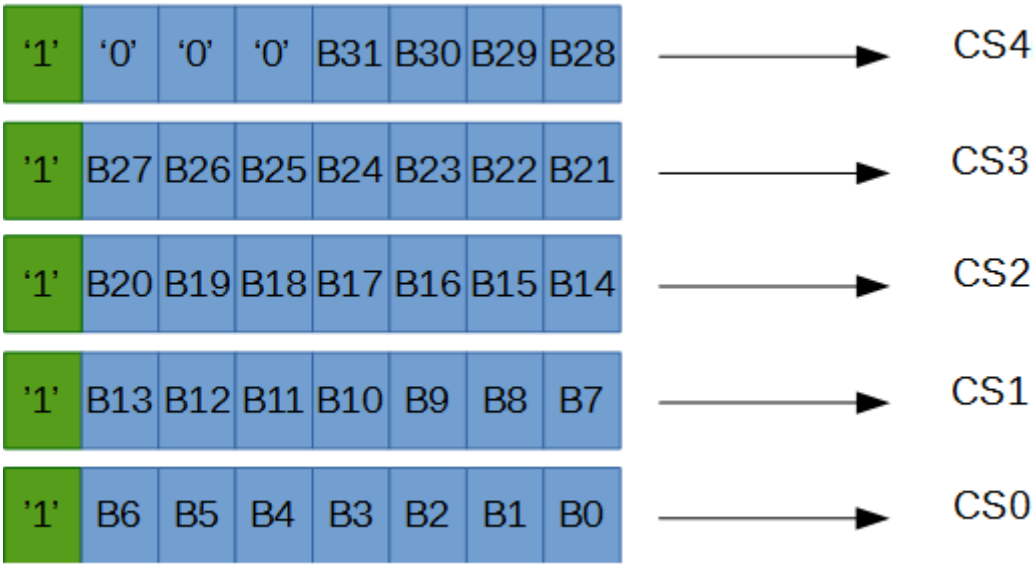

Illustration 9: Checksum bits arrangement

3.4.4 Frame format example

All frames start with a “Start of Frame” character (0x02) and end with an “End of Frame” character (0x03). Pixel data are stored between these commands.

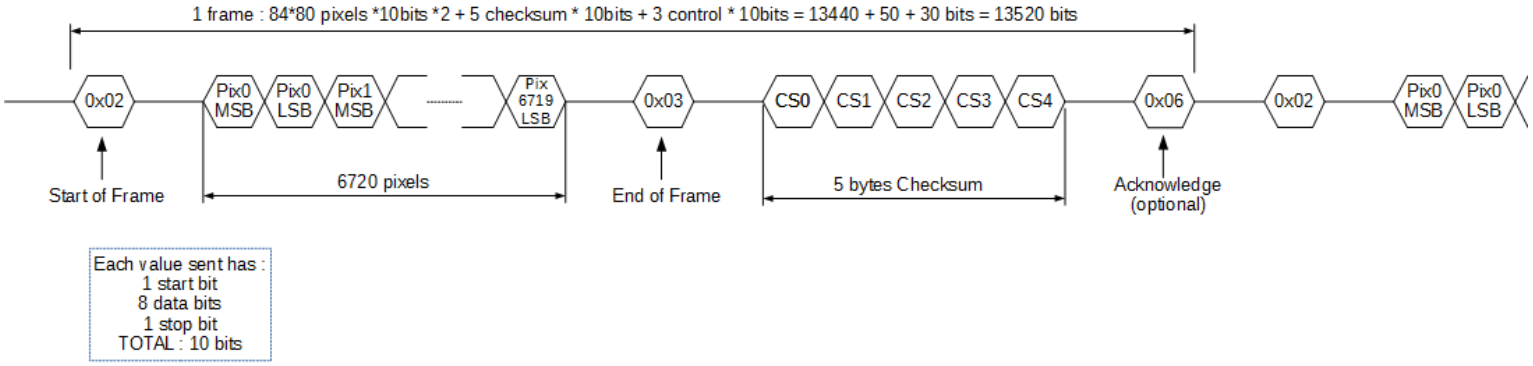

Illustration 10: Frame data over RS485 protocol

3.5 SPI communication

As the others FullScale Digi Board, some commands can be send to the Emitter to configure the proxy board or the sensor.

The basic ASCII message is a 6 characters message including a start marker “ ! ” like the following:

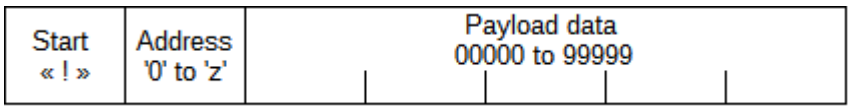

Illustration 11: ASCII message format

- Each message should start with a “ ! ” character as a start of message marker.
- The Address field is only one char long (please refer to the table register map here after).
- The Payload field is 4 char long in which a 12bit value perfectly fits (0 to 4095). This filed has a fixed length: leading zeros should be used.

ASCII Address	Register function	Target
'0'	Not used	
'1'	Proxy board integration time	Proxy board SPI address 0x01
'2'	Proxy DAC ADD02 setting (0000 to 4095)	Proxy board SPI address 0x02
'3'	Proxy DAC ADD03 setting (0000 to 4095)	Proxy board SPI address 0x03
'4'	Reserved – CKDIV system register	Proxy board SPI address 0x04
'5'	GMS register Gain/Mirroring/Windowing	Proxy board SPI address 0x05
'E'	Sensor I2C address	Proxy board SPI address 0x0E
'F'	Sensor I2C value	Proxy board SPI address 0x0F

Illustration 12: RSB1-Z register map

SPI messages are processed after a frame was sent to the Receiver.

When a message is sent to the Emitter, it sends back an “Acknowledge” (0x06) to inform that it processed the received message. The “Acknowledge” is sent between the “End of Frame” and “Start of Frame” characters,

3.6 Starting configuration

When powered up, the Emitter send the following configuration to the proxy / sensor:

- Proxy MASTERCLOCK = 2.88MHz
- TINT = 10
- GSK = 1250
- CKDIV = 3 (SensorClock = 360kHz)

The Emitter board is configured for the K035 proxy board with a Micro80Gen2 sensor running @23FPS.

4 Receiver board

The Receiver board convert the RS485 data coming from the Emitter to a Virtual COM Port.

When the Receiver is connected to a computer, then, it appears as “COMx” on Microsoft Windows or as “/dev/ttyUSBx” on a Linux platform (with x the number of serial port).

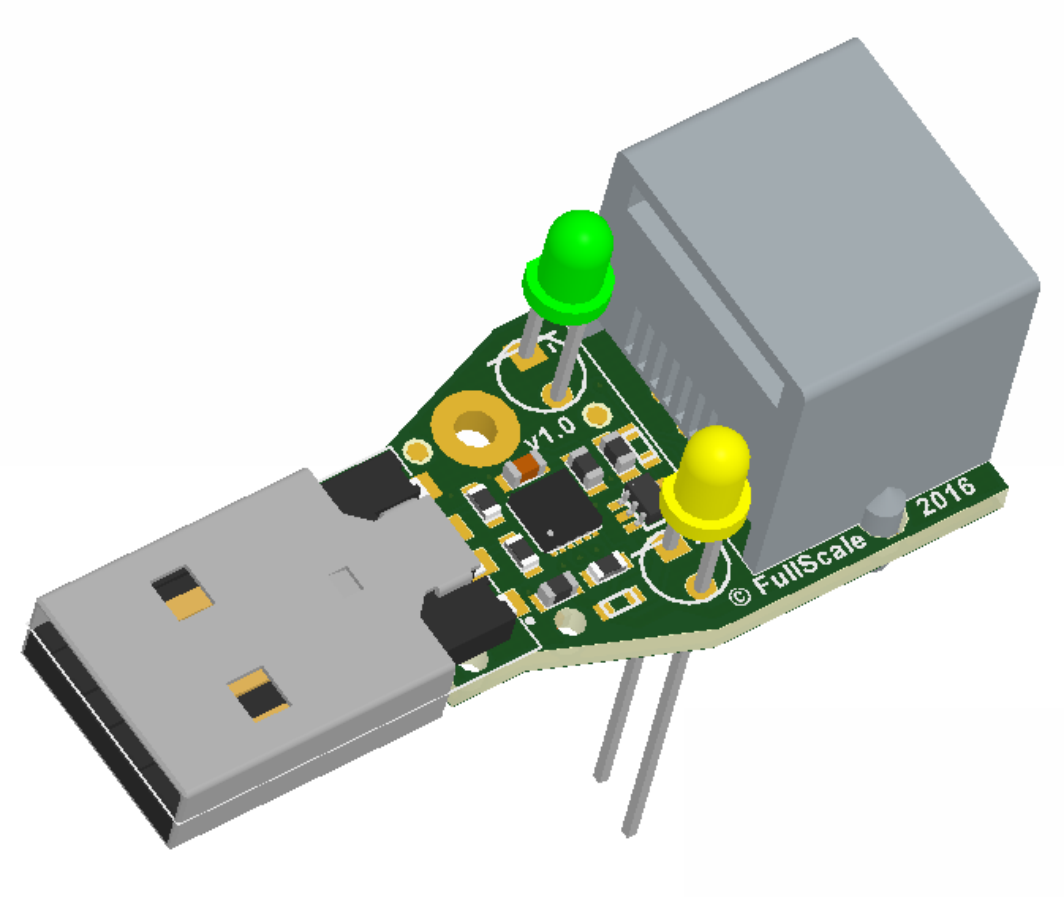

Illustration 13: RSB1-Z Receiver board 3D model

The green LED lights up when the board is powered (connected to a USB port). The orange LED lights up when data pass through the Receiver board.

This board is USB powered and transmit the power supply to the Emitter board through the RJ45 cable.

5 Connecting the RSB1-Z digi board

The RSB1-Z is composed by one Receiver (with the USB connector) and one Emitter. These boards has to be linked using a RJ45 cable.

Once the two boards are connected. Plug the USB connector of the RSB1-Z to one of the 4 USB port of the Raspberry Pi 3 /4.

A green LED has to be light on the Emitter and on the Receiver.

CAREFUL: Do not connect the RSB1-Z to the Ethernet port of the Raspberry using the RJ45 cable.

Now, the software can be launched.

6 RSB1-Z Smart Player software

6.1 On Linux Operating System

To start the software, double click on the folder available on the desktop named “RSB1-Z_Smart_Player_1x_v1.1”. Then, go launch the executable file on “release-Linux/RSB1-Z_Smart_Player”

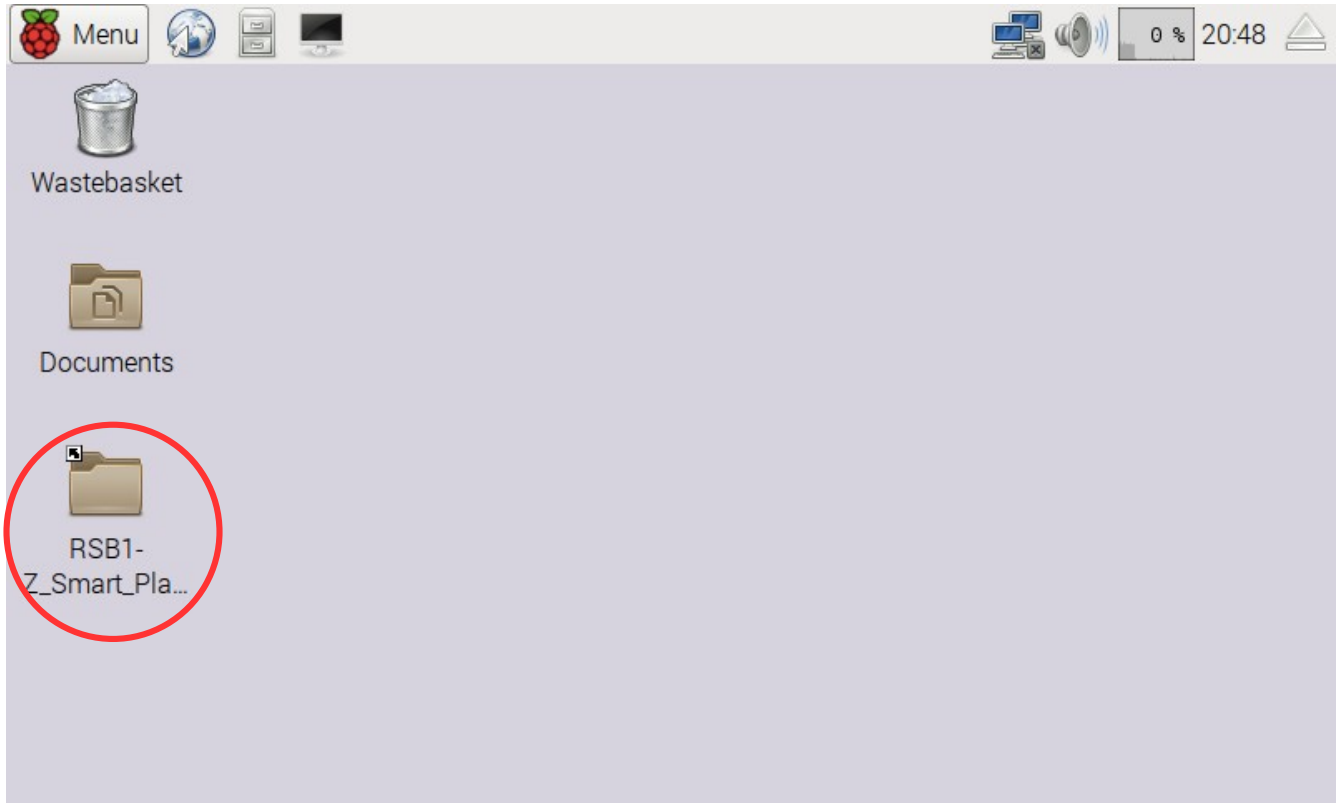

Illustration 14: Software shortcut

When opening this file, a window will ask if the software has to be execute into a terminal or not. Click on “Execute”.

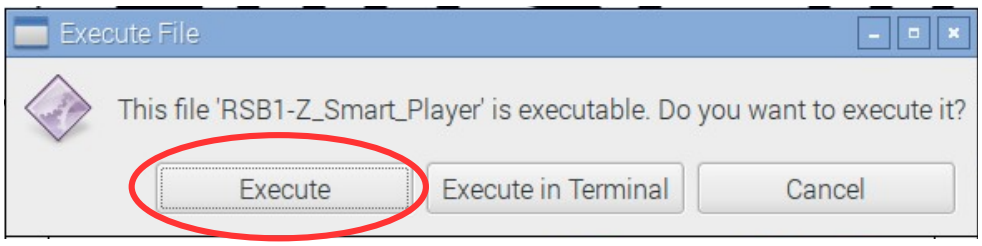

Illustration 15: Do not execute in terminal

Now, the software will be display.

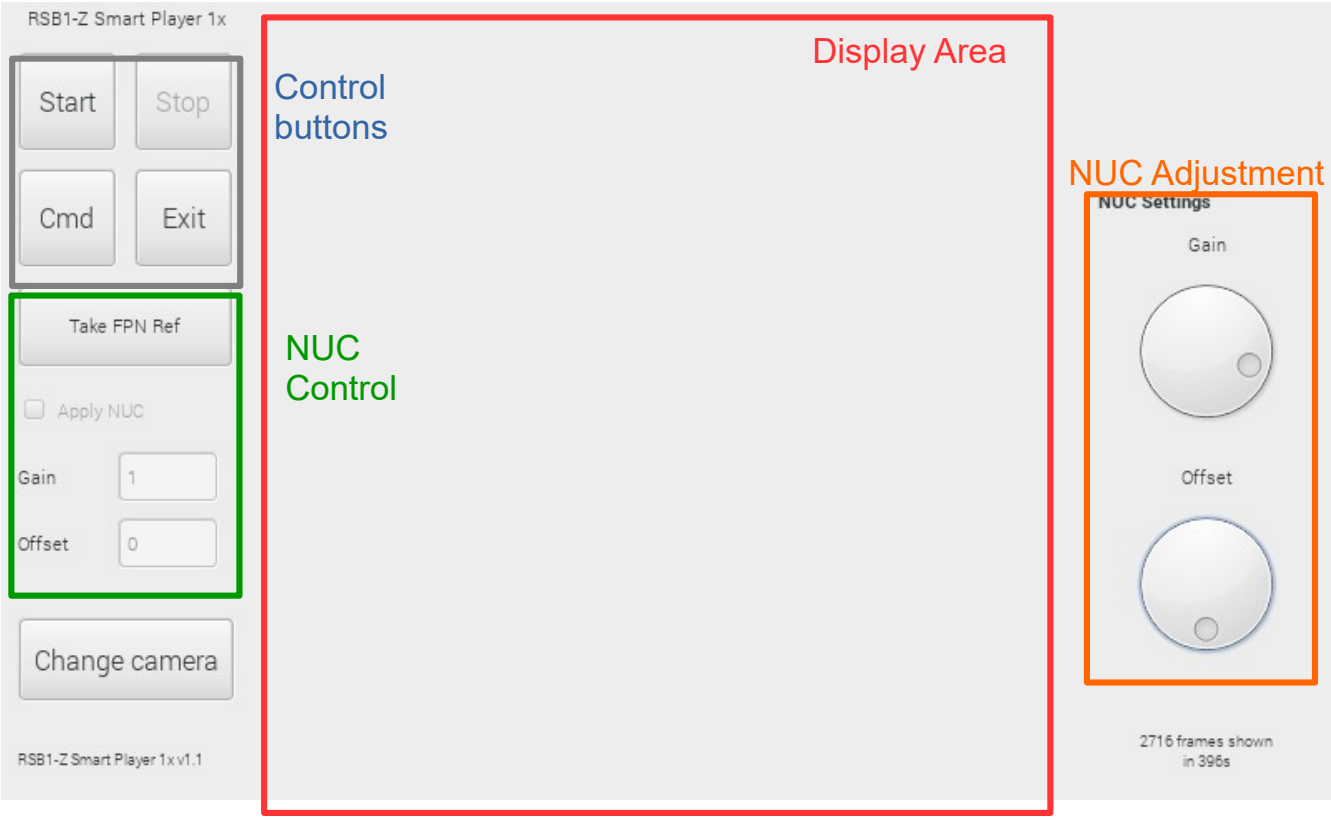

Illustration 16: RSB1-Z Smart Player startup

This software was made to be user-friendly and easily usable using the Raspberry Pi touchscreen.

You can handle the touchscreen in the hands and controlling the software using the fingers.

The Gain and Offset can be adjusted turning the fingers. And the buttons are actionable touching them.

6.1.1 Starting the display

When the RSB1-Z (with the proxy and the sensor) is connected to the Raspberry Pi, press “Start” button. The RAW image will be displayed.

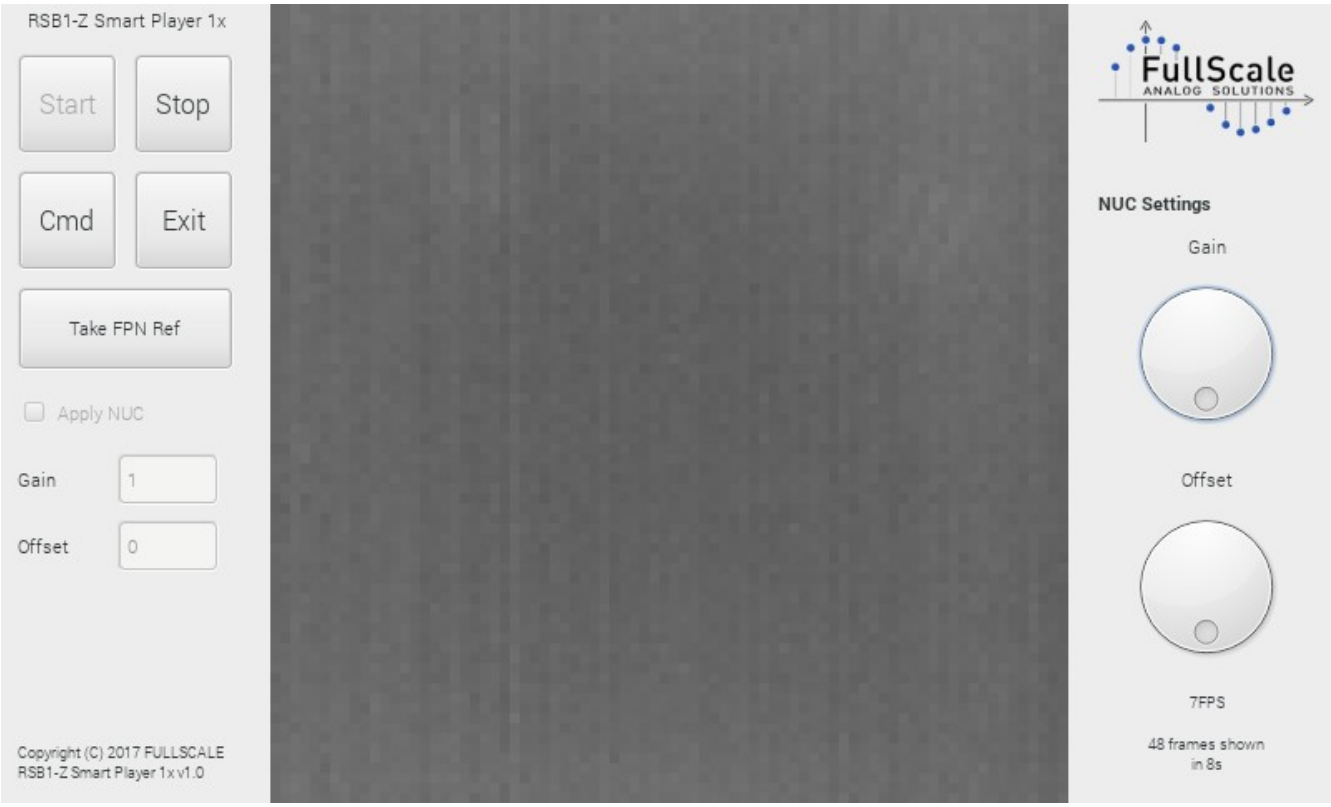

Illustration 17: RAW Image display starting the display

6.1.2 Non Uniformity Correction

The software can make a NUC on the image. To do it, place a shutter in front of the sensor and then, press “Take FPN Ref” button. The image will be saved as reference. Then, click on “Apply NUC” box to enable the correction.

The display will become black.

Illustration 18: NUC applied without Gain or Offset

Using your fingers, adjust the values of the Gain and the Offset.

Illustration 19: NUC with a correct gain and offset

6.1.3 Command to apply to the proxy / sensor

With the RSB1-Z, it is possible to send commands to the proxy or directly to the sensor.

To do it, press the “Cmd” button. A new window will appear.

Illustration 20: Send command window

To send some commands to the proxy it is mandatory to use a keyboard (or install a virtual Keyboard for Raspberry Pi).

To know the commands available, please refer to the RSB1-Z datasheet.

For the sensor, the address and values may be entered using a finger. Once the two fields are correct, press “Send” (on the bottom) to apply the command.

Addresses and values may corresponding to the ULIS Micro80Gen2 datasheet.

6.2 On Microsoft Windows Operating System

To start the software, launch the executable file on “RSB1-Z_Smart_Player.exe”

Now, the software will be display.

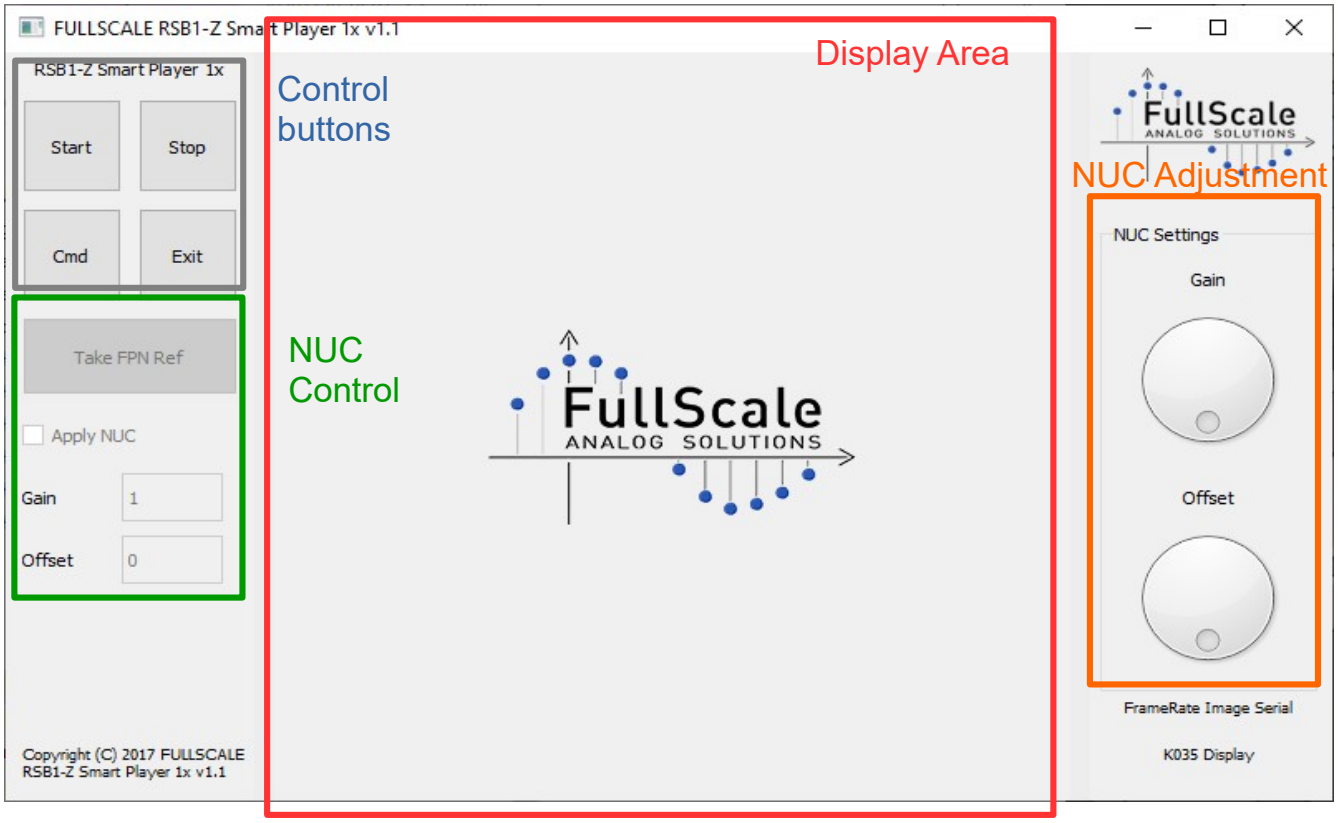

Illustration 21: RSB1-Z Smart Player startup

This software was made to be user friendly and easily usable using the Raspberry Pi touchscreen.

You can handle the touchscreen in the hands and controlling the software using the fingers.

The Gain and Offset can be adjusted turning the fingers. And the buttons are actionable touching them.

6.2.1 Starting the display

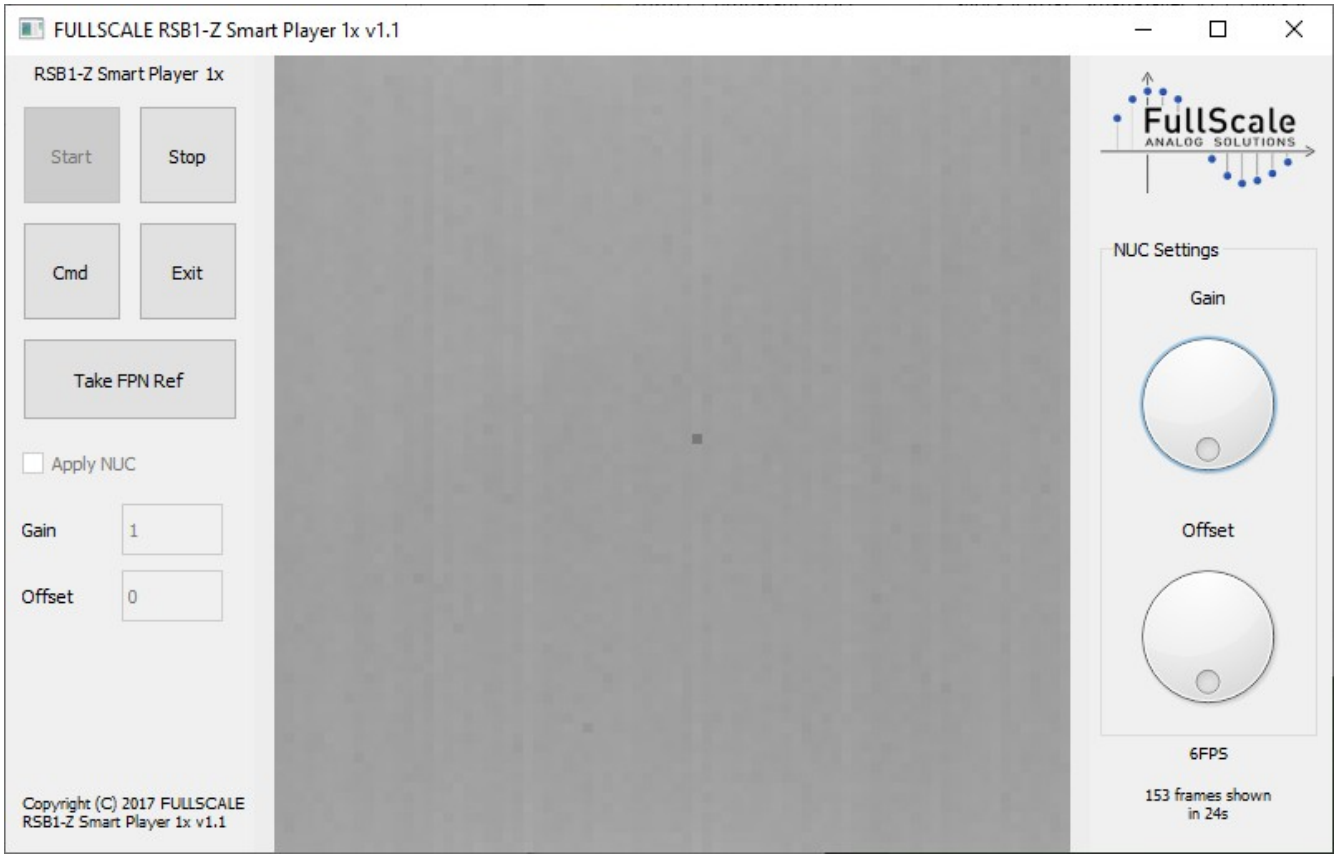

Illustration 22: RAW Image display starting the display

When the RSB1-Z (with the proxy and the sensor) is connected to the Raspberry Pi, press “Start” button. The RAW image will be displayed.

6.2.2 Non Uniformity Correction

The software can make a NUC on the image. To do it, place a shutter in front of the sensor and then, press “Take FPN Ref” button. The image will be saved as reference. Then, click on “Apply NUC” box to enable the correction.

The display will become black.

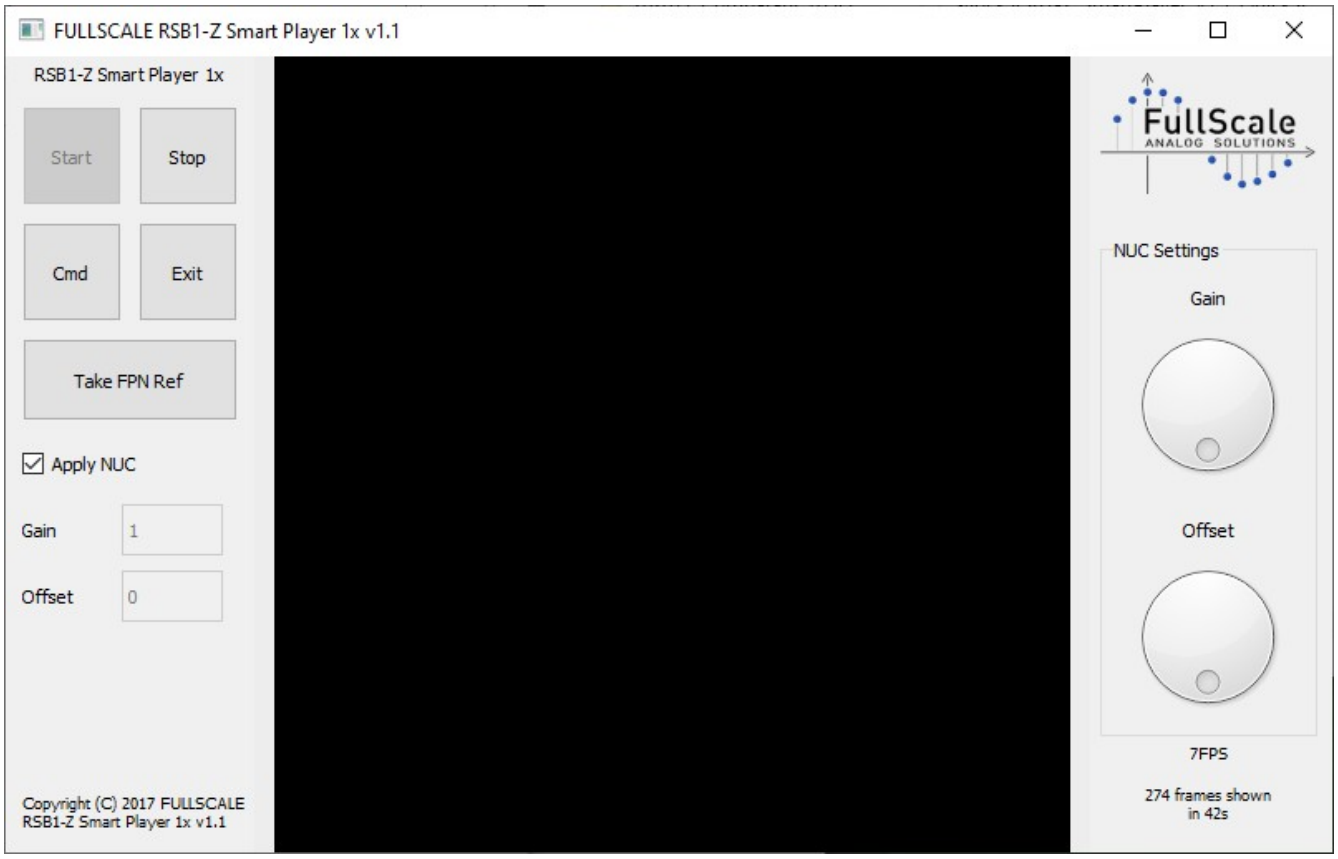

Illustration 23: NUC applied without Gain or Offset

Using your fingers, adjust the values of the Gain and the Offset.

Illustration 24: NUC with a correct gain and offset

6.2.3 Command to apply to the proxy / sensor

With the RSB1-Z, it is possible to send commands to the proxy or directly to the sensor.

To do it, press the “Cmd” button. A new window will appear.

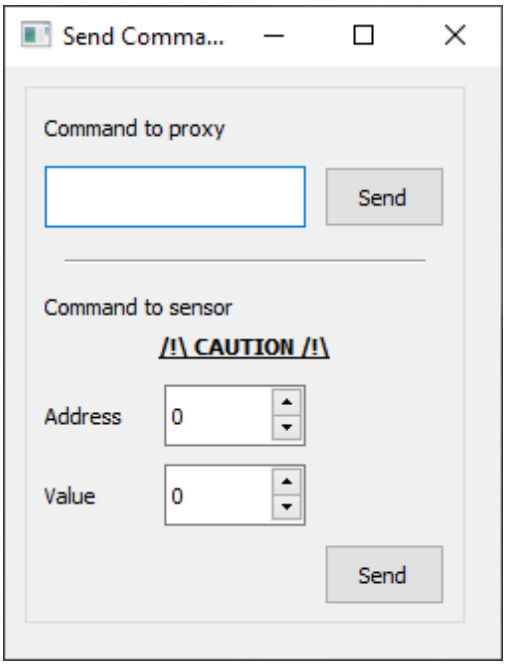

Illustration 25: Send command window

To send some commands to the proxy it is mandatory to use a keyboard (or install a virtual Keyboard for Raspberry Pi).

To know the commands available, please refer to the RSB1-Z datasheet.

For the sensor, the address and values may be entered using a finger. Once the two fields are correct, press “Send” (on the bottom) to apply the command.

Addresses and values may corresponding to the ULIS Micro80Gen2 datasheet.