
Ultra-cheap SDR Digital Television Transmission: ISDB-T with an osmo-fl2k and an RTL-SDR

Federico Larroca
Pablo Flores-Guridi
David Artenstein
Lucas Inglés
Gastón Morales

FLARROCA@FING.EDU.UY
PABLOF@FING.EDU.UY
DAVID.ARTENSTEIN@FING.EDU.UY
LUCASI@FING.EDU.UY
GMORALES@FING.EDU.UY

Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, Montevideo, Uruguay

Abstract

We present the implementation in GNU Radio of an ISDB-T transmitter, the digital television standard used in most South American and some Asian and Central American countries. All the configurable parameters of the standard (such as mode, time interleaver's depth or coding rate) are accessible from the block's parameters, and they were thoroughly tested using commercial TVs in over-the-air transmissions (as well as other SDRs). We have also tested different SDR models as transmitters. In particular, we have successfully used an osmo-fl2k (a 15\$ USB3-to-VGA adapter repurposed as an SDR) as an ISDB-T transmitter. This, together with the 1-seg feature of the standard (i.e. a signal of less than 500 kHz of bandwidth that can be decoded on its own) makes it ideal for an ultra-cheap video transmission setup since, as we show, even an RTL-SDR is capable of decoding the signal. The complete code is available at <https://github.com/git-artes/gr-isdbt>.

1. Introduction

The importance of terrestrial television, although decreasing in popularity, is yet significant. For instance, according to a study carried out in 2019 by Uruguay's *Instituto Nacional de Estadística* (National Institute of Statistics), 96.2% of the households had at least a TV set, but only 69.3% had an internet connection (any internet connection, not necessarily broadband) and 65.4% had paid TV installed (either satellite or cable) (*Instituto Nacional de Estadística*, 2019). The presence of TV is even more important than radio, with about 76% of households having one.

With similar figures in most South American countries, and the digital transition still halfway, a complete understanding of the technology is, maybe more than ever, the key to successfully completing the analogical blackout and obtain its expected benefits. The availability of receivers and transmitters that are both inexpensive and open will only increase the trend towards embracing this new technology.

Moreover, the region has the particularity that when the digital TV standard was chosen around 2010, and under the lead of the Brazilian government, a technology appropriation was performed. Based on Japan's ISDB standard, they developed the SBTVD (Sistema Brasileiro de Televisão Digital-Terrestre). The ensuing improvements over ISDB were later incorporated into the original standard, resulting in the so-called "ISDB-T International" (or ISDB-Tb). The latter was then adopted by most South American countries, in addition to some Central American and Asian ones. See the next section for further details.

In this context, and back in 2016, we presented *gr-isdbt* (Larroca et al., 2016), an open-source ISDB-T receiver implemented as an Out-of-Tree (OOT) Module for GNU Radio (*gnu*). The present article presents what may be considered the completion of the *gr-isdbt* project: an open-source ISDB-T transmitter implemented on GNU Radio.

The implementation of the transmitter was simplified by the existing SDR-based receiver, which allowed us to debug the received signal, both in simulated (i.e. all software) as well as over-the-air transmissions, without resorting to expensive ISDB-T signal analyzers. In this sense, all the possible configurable parameters of the standard (accessible from the GNU Radio Companion block's configuration) were tested using our complete transceiver first, and then with commercial TV sets, ensuring its correct implementation.

We have also tested different SDR hardware models (e.g. USRP B200), and were particularly interested in checking whether the osmo-fl2k (Markgraf, 2018) (basically a 15\$

USB3-to-VGA adapter repurposed as an SDR transmitter, see Sec. 4) was capable of transmitting an ISDB-T signal. Since ISDB-T includes the possibility of transmitting a small bandwidth signal along with the HD one and virtually independent of it (the so-called 1-seg feature; cf. Sec. 2) a possible ultra-cheap setup is to use the osmo-fl2k along with a cheap RTL-SDR receiver (see for instance (rtl)).

The complete code, including several examples, is available at <https://github.com/git-artes/gr-isdbt>. The rest of the article is structured as follows. After discussing with more detail ISDB-T in the next section, Sec. 3 presents the implemented transmitter. Osmo-fl2k is analyzed in Sec. 4, and the evaluation of the transmitter is reported in Sec. 5, including the experiments with an osmo-fl2k and an RTL-SDR. Conclusions and future work are included in Sec. 6.

2. ISDB-T

ISDB-T stands for Integrated Services Digital Broadcasting - Terrestrial, and is the Japanese digital television standard. It is based on DVB-T, and was ratified in the early 2000's, after the European and the American standards were adopted. As we mentioned in the last section, it was later adopted and adapted by Brazil: most importantly a new interactivity middleware named *Ginga* was defined (instead of the original *BML*) and MPEG-4 replaced MPEG-2 for source coding. The new version of the standard was named ISDB-Tb, and was later adopted by most South American countries, as well as some Central American and Asian countries.

Figure 1 shows the entire block diagram of the ISDB-T transmission system. We now briefly discuss the specifications of each of the blocks, paying special attention to those aspects particular to ISDB-T. For more details, the interested reader should consult (Association of Radio Industries and Businesses (ARIB), 2014).

Let us begin by the **modulation scheme**. Orthogonal Frequency Division Multiplexing (OFDM) is used over a 6 MHz bandwidth channel. After the OFDM modulation, a cyclic prefix (CP) is added which length is expressed as a fraction of the active symbol's length, T_s . There are four possible values: $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ or $\frac{1}{32}$.

Regarding the number of carriers in one OFDM symbol, it can be either 2^{11} , 2^{12} or 2^{13} . However, the sampling rate (termed f_{FFT} in the standard) is always equal to $512/63 \approx 8.126$ MHz. This means that keeping the total data rate constant, the operator may choose to use more carriers but slower symbols in order to immunize the radio signal from multipath propagation effects, or less carriers but faster symbols in order to immunize the signal from Doppler effect. This choice is termed **Transmission Mode**,

and may be either 1, 2 or 3, although generally mode 3 is used.

Focusing on mode 3 for the rest of the article for the sake of clarity, not all the 8192 carriers are used, but rather 5617, enough to meet the bit-rate and bandwidth requirements (where a guard interval and zero-padding is used in the rest of the carriers). This useful spectrum is in turn sub-divided into 13 sub-bands named **segments**, of 432 carriers each. These 13 segments may be used independently from one another, a feature first implemented in ISDB-T and called *Band-Segmented Transmission OFDM* (BST-OFDM). In particular, in this case, they can be combined in up to three so-called **hierarchical layers** (A, B and C), which transmit different Transport Streams (TSs). Moreover, these groups of segments can be configured to use different forward error correction (FEC) rates, time interleaving lengths and modulation schemes.

Since up to three different TSs may be thus transmitted in the same channel, the typical configuration keeps a single segment with very robust transmission parameters. The idea is to allow visualization by cellphones, a market which particularly interested the ISDB-T designers. To this end, this single-segment layer is completely independent of the other layers, as it is frequency and time interleaved separately from the rest.

Thus, handheld receivers (or RTL-SDRs as in our case) need only tune and demodulate this single segment, making it possible to work with lower sample and data rates, and thus, less CPU requirements. This feature is known as **1-segment** (or 1-seg for short). Figure 2 illustrates this segment-based spectrum division (and the numbering used by the standard) and the typical 1-seg configuration.

From the 5617 active carriers, there are several which are used as **pilots** to assist the receiver in the equalization process. These so-called scattered pilots (SP) change position from symbol to symbol to avoid pathological situations (such as a permanent deep fade in the spectrum), although their payload is known (which in turn depends on their position).

In addition to these pilots, several carriers are devoted to transmitting the modulation parameters at use. There are a total of 203 bits to be transmitted: the so-called **TMCC** (Transmission Multiplexing Configuration Control). They occupy fixed carriers in the OFDM symbol, each of them corresponding to the same TMCC's bit, and DBPSK modulation scheme is used. Thus, 204 OFDM symbols are required in order to receive the complete TMCC, completing a so-called **OFDM frame**.

The rest of the blocks, except for the TS Remux which we explain next, are somewhat standard: frequency and a configurable time interleaving of the complex symbols;

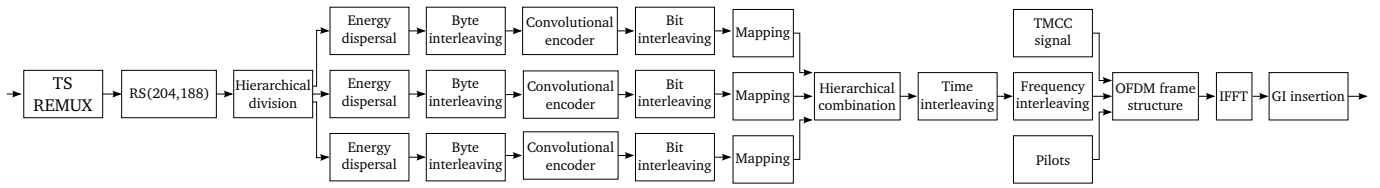


Figure 1. ISDB-T transmission system block diagram.

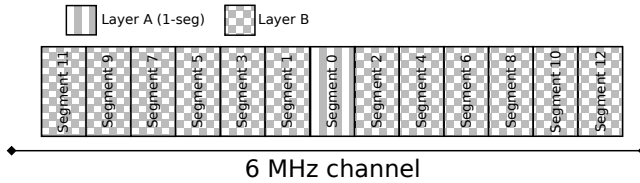


Figure 2. ISDB-T segmented spectrum and the 1-seg layered configuration.

mapping, bit interleaving and convolutional encoder are applied to bits; byte interleaver, energy dispersal and Reed-Solomon encoding may be regarded as applied to bytes. It is important to highlight that most of these algorithms are applied separately to each layer (thus the three parallel paths in Fig. 1), and each layer may use its own set of parameters.

Readers familiar with MPEG Transport Streams may notice that in order to send three independent TSs at the same time (one through each hierarchical layer, with its own channel coding and modulation scheme), the ISDB-T transmission subsystem should have up to three independent inputs. This is because the MPEG TS standard (ISO 13818-1:2000, E) does not provide any hierarchical information, so there is no way to separate one TS from another once all of them have been multiplexed together. In other words, the MPEG TS as it is, cannot be applied to hierarchical transmission (Uehara, 2006). To solve this problem, ISDB-T defined the Broadcast Transport Stream (BTS), which is created in the TS Remux block by combining up to three different TS inputs. During this process, 16 extra bytes are added at the end of each TS packet: the first 8 bytes are the ISDB-T Information, which indicates among other things, to which hierarchical layer the current TS packet corresponds; and the last 8 bytes are an optional Reed Solomon error correction code.

Following the block diagram on Fig. 1, after the BTS arrives to the hierarchical divider, each TS packet is routed to its corresponding layer and each layer is then processed and sent independently by using the already presented BST-OFDM. However, it should be seen that the information on the sequential order of the BTS may get lost in this serial-

to-parallel process, so receivers may not be able to recover the BTS as it was originally transmitted. One important thing to consider at this point is that if the BTS changes its original packets' order, large Program Clock Reference (PCR) jitter events may occur (Tektronix), and this is inadmissible for video and audio decoding. It is then necessary to recover the BTS in reception just as it was transmitted. To that end, ISDB-T defines a particular and well-known order for every BTS packet (which in turn will be different for different transmitter configurations) that can be recovered in reception by using some particular algorithm, defined in the standard and explained in detail in (Flores-Guridi & Larroca, 2017).

Table 1 summarizes what was described in this section and adds some other extra information. The parameters used in each layer for the convolutional encoding, time interleaving and modulation scheme for any particular transmission are all specified in the TMCC.

3. gr-isdbt

In this section we present the transmitter side of `gr-isdbt`. For details on the receiver the reader should consult (Larroca et al., 2016). The complete GNU Radio Companion (GRC) flowgraph is shown in Fig. 3. Note that some of the blocks are part of GNU Radio, such as the Reed-Solomon and convolutional encoders, as well as the OFDM cyclic prefixer.

The rest of the blocks mostly correspond to blocks appearing in Fig. 1. In particular, the Hierarchical Divider takes a BTS at its input and outputs the corresponding number of Transport Streams. Note that a BTS file is not actually necessary to use the transmitter, as the MPEG-2 Transport Streams may be directly fed into the Reed-Solomon encoder (previously passing it through a Stream to Vector block of length 188 bytes). From our experience, however, this configuration may make some commercial TV sets experience playback instabilities, or even failure. Our understanding is that those particular TV sets are trying to recover the full BTS before demuxing each transmitted Transport Stream, instead of tuning just the corresponding hierarchical layer. This is not a problem for single-layer or 1-seg transmissions, which can

Ultra-cheap SDR Digital Television Transmission: ISDB-T with an osmo-fl2k and an RTL-SDR

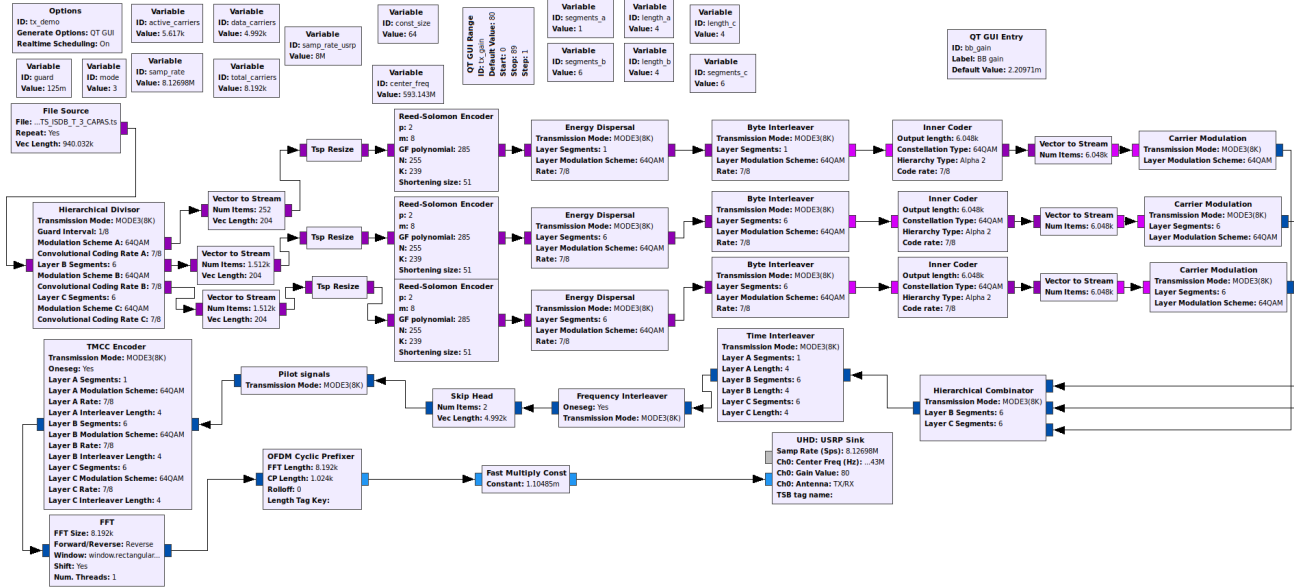


Figure 3. The complete transmitter's GRC flowgraph of `gr-isdbt`.

always be treated independently as they are transmitted by sending only one TS over a single layer, consisting of just the segment in the center of the spectrum.

Another block worth highlighting is the `Skip Head` with length 2. The effect is to drop the first two OFDM symbols. The byte interleaver, bit interleaver (included in the `Carrier Modulation` block) and time interleaver generate a transient in the symbols' flow up to the `Skip Head` in the diagram. Since the duration of this transient depends on the chosen parameters, the standard includes a certain extra delay in every block so that the total transient is always equal to two OFDM symbols. The `Skip Head` block is used precisely to drop these useless symbols before encoding the pilots and TMCC symbols.

A not-so-obvious question is how to generate Transport Streams that may be used by the system. Although we provide some samples in the project's website¹ that are meant to be used with the example flowgraphs, transmitting an arbitrary video or the webcam is desirable.

To this end, `ffmpeg` ([FFmpeg development team](https://ffmpeg.org/)) may be used as follows. Firstly, given the configured parameters of the corresponding layer, the resulting bitrate of a single segment may be obtained from Table 3-3 of the standard ([Association of Radio Industries and Businesses \(ARIB\), 2014](https://www.arib.or.jp/english/html/contents.htm)). Then, in order to re-encode a video file (say `video.mp4` into `video4isdbt.ts`, usable as input to the flowgraph), the following command may be used:

¹<https://iie.fing.edu.uy/investigacion/grupos/artes/projects/gr-isdbt/>

Table 1. ISDB-T transmission system parameters available.

Parameters	Values
Total Bandwidth	6 MHz
Number of segments	13
Segments bandwidth	$6000/14 \approx 428.57 \text{ kHz}$
Number of active carriers	1405 (mode 1) 2809 (mode 2) 5617 (mode 3)
Active symbol duration	$252 \mu\text{s}$ (mode 1) $504 \mu\text{s}$ (mode 2) $1004 \mu\text{s}$ (mode 3)
Guard interval duration	1/4, 1/8, 1/16, 1/32 (of active symbol duration)
Convolutional Code Rate	1/2, 2/3, 3/4, 5/6, 7/8
Time interleaving depth	0, 1, 2, 4 (mode 1) 0, 2, 4, 8 (mode 2) 0, 4, 8, 16 (mode 3)
Modulation schemes	DQPSK, QPSK, 16QAM, 64 QAM


```
ffmpeg -i video.mp4 -c:v libx264 -b:v
2M -r ntsc -s 720x480 -pix_fmt yuv420p
-profile:v main -level:v 3 -c:a aac
-b:a 64k -profile:a aa_c_low -f mpegts
-mpegts_service_id 0x70e0 -metadata
service_name="ISDBT Channel" -streamid
0:50 -streamid 1:52 -muxrate 5727280
video4isdbt.ts
```

It is important that the `mux_rate` is equal to the number of segments used by the layer (in the example above 13) times the value obtained from Table 3-3 in the standard (in this example 440.56 kbps, corresponding to QPSK, 2/3 coding rate and a 1/16 guard interval). Moreover, the sum of the `-b:v` (video bitrate) and `-b:a` (audio bitrate) parameters should be lower than `mux_rate`. We have empirically found that values 30% lower work fine. Note that the standard also supports `mpeg2video` as the video codec (instead of `libx264`). Examples of how to transmit the webcam are available at the project's website too, although it is very similar to the command above (and further interesting modifications, such as transmitting the desktop, are also possible).

Finally, note that our repository also includes equivalent blocks for 1-seg transmission only. These may be used in conjunction with the 1-seg only reception examples, also available. More details in Sec. 5.

4. Osmo-fl2k

Osmo-fl2k is a project by Osmocom to use USB3-to-VGA adapters based on the Fresco Logic FL2000 chip as a transmitter SDR. We discuss here how this is achieved, as it will help us understand how to use it in order to transmit ISDB-T.

The basic idea in osmo-fl2k is the same as the one behind TEMPEST, the name by which spying on unintentional emanations by VGA cables is known (see for example (Van Eck, 1985; Kuhn, 2003; Marinov, 2014)). Let us denote by x_k the intensity of the red component of the k -th pixel (and thus with $0 \leq x_k \leq 0.7$ V). The actual voltage in the red pin will be a Pulse Amplitude Modulated (PAM) signal of the form:

$$x(t) = \sum_k x_k p(t - kT_p), \quad (1)$$

where $p(t)$ is the pulse used by the VGA card, which as an approximation we will assume it is a square signal of duration T_p (which is the pixel time). If we analyze the spectrum of $x(t)$ we obtain the following expression for its Fourier transform:

$$X(f) = P(f) \sum_k x_k e^{-j2\pi f T_p} = P(f) X_s(f). \quad (2)$$

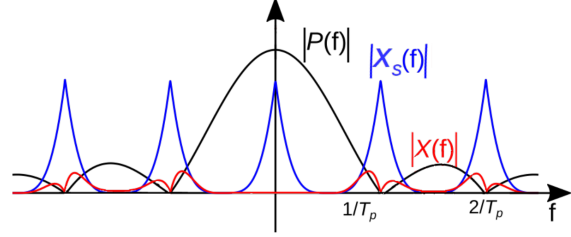


Figure 4. The spectrum of the VGA signal (cf. (1)). Please note that the baseband signal is not depicted as it will not propagate.

That is to say, the Fourier transform of the pulse $p(t)$ multiplied by the discrete time Fourier transform (DTFT) of the pixel intensities evaluated at $2\pi f T_p$. Defining this last term as $X_s(f)$, (2) is graphically depicted in Fig. 4.

The idea in TEMPEST is to try to recover the values of x_k by tuning into one of the harmonics of $1/T_p$ (where the typical VGA signal will be concentrated around as depicted in Fig. 4). In Osmo-fl2k the idea is the opposite: generate the sequence of pixels x_k such that in baseband it corresponds to the samples of what we are trying to transmit (in this case an ISDB-T signal), but such that the harmonics of $X_s(f)$ are not on the nulls of $P(f)$.

However, this idea is hindered by the existence of so-called blanking intervals. These are zeros in the sequence x_k inserted by all VGA adapters before and after each horizontal and vertical synchronization signals. We have thus only partial control over the sequence x_k , and the resulting signal is significantly altered.

The important feature of Fresco Logic's FL2000 chip is that the complete sequence x_k (including blankings) is all set by the host computer, and sent over the USB to the VGA adapter. The osmo-fl2k team reverse-engineered the USB protocol, disabled both horizontal and vertical blanking and provide the necessary software to send arbitrary samples to the USB adapter (in particular using the red component).

Thus, by connecting an antenna to the corresponding pin, we could transmit an arbitrary signal. For example, if we wanted to send our ISDB-T signal over a osmo-fl2k at a central frequency $f_c = 473$ MHz (channel 14), we could generate the signal with the flowgraph in Fig. 3. However, instead of sending it over to a USRP, we should upsample it to $1/T_p = f_s = 138$ MHz (which we will now denote as sampling rate)², multiply it by a complex exponential with a frequency of $f_{\text{offset}} = 59$ MHz, take the real part

²Although osmo-fl2k accepts any sample rate, it actually sets f_s as the nearest rational multiplier of 160 MHz, with interpolation from 3 to 6 and decimation up to 63. However, the actual bottleneck is the xHCI controller and the host PC, resulting in a maximum f_s which is generally below 150 MHz (and may be obtained by using the `fl2k_test` tool).

of the resulting complex and send the resulting samples (x_k) to the osmo-fl2k. The third harmonic plus f_c (precisely 473 MHz) will now contain a (somewhat distorted by $P(f)$) copy of our baseband complex ISDB-T signal.

Some comments are in order:

1. Note that f_{offset} acts as a sort of intermediate frequency, and is used to generate a first real pass-band signal from the complex baseband one, as the osmocomb2k naturally accepts only real samples. Moreover, it is also used to avoid the nulls of $P(f)$ at the multiples of f_s . Even if $p(t)$ is not necessarily a perfect square signal, important fades in $P(f)$ are to be expected at those frequencies.
2. Although strictly speaking $P(f)$ will distort our signal, the signal's bandwidth (6 MHz if transmitting all segments in our case) is significantly smaller than f_s (138 MHz in this case).
3. In addition to our intended transmission, copies centered around $nf_s \pm f_{\text{offset}}$ (with $n \in \mathbb{N}$) will also be present, although the larger the n , the more attenuated the resulting signal. Thus small values of n should be favored. This is arguably the biggest disadvantage of osmo-fl2k: important interference is generated, and proper filtering is mandatory.
4. Since the signal had to be upsampled to $f_s = 138$ MHz, it means that online/real-time transmission is out of the question except for high-end PCs or simple waveforms (such as FM). The alternative is to save the samples to a file, and feed this file to osmo-fl2k afterwards. Although smaller values of f_s are possible, it would require higher values of n , and thus result in an excessively attenuated signal.

The following section presents our results using osmo-fl2k as an ISDB-T transmitter.

5. Experiments

5.1. Full-seg experiments

As we mentioned in Sec. 1, once the transmitter was completed the first tests were performed all in software using our already implemented and tested receiver (Larroca et al., 2016). Such flowgraph, which simply put is a simulation of a complete OFDM-based communication system, is included in the examples folder of the repository. It has proved particularly useful in classroom to demonstrate channel effects over OFDM signals, although several other non-academic uses may be easily envisioned.

Regarding over-the-air transmissions, we have first used a USRP B200 as a transmitter (Ettus Research) connected to

a Ikusi Flash HD Nano antenna (Ikusi Multimedia). We have first used a USRP B100 (now discontinued) as a receiver, along with a simple dipole antenna. This helped us finish debugging the transmitter (in particular, the delays discussed in Sec. 3), and set the normalization factors and gains to be used by the B200. Again the previously existing receiver helped enormously, as there was no need for expensive DTV measurement equipment.

The final tests were carried out using commercial TVs as receivers. In this case, we noticed significant differences between the TV models we tested. In some of them, blank screens would appear if the slightest sample was dropped, whereas other models would show no noticeable degradation. In any case, a laptop with a i7-4712MQ and 16 GB of RAM was enough to re-encode the webcam and transmit it over ISDB-T for several hours without problems, but note that the host PC is very important if commercial TVs are the target receivers.

Let us now focus on our tests using osmo-fl2k. Firstly, in order to comfortably connect an antenna to the VGA adapter, we have used a VGA to BNC breakout (although simply repurposing a spare VGA cable is also possible). In order to ensure the largest emitted power, we used a sampling frequency of $f_s = 127.14$ MHz and $f_{\text{offset}} = 80$ MHz, and tuned a B200 used as an ISDB-T receiver to this frequency. Note that we are thus using $n = 0$ (cf. Sec. 4). Given that the idea was to reach an ultra-cheap setup, we have decided to use a simple dipole antenna also at the transmitter.

Results are remarkably good. Using QPSK for a transmission using a single layer occupying the whole 13 segments the signal is decoded from several meters apart. For instance, the received constellation about 8 meters away from the transmitter is shown in the upper portion of Fig. 5.

Naturally, the performance when compared to the USRP B200 is inferior, but quite interestingly not significantly so. For instance, the MER (Modulation Error Ratio) when transmitting with the osmo-fl2k was only about 3-4 dB lower than when transmitting with the USRP B200 (in both cases the receiver was the USRP B100). However, this was only very near the transmitter, and significant differences appear as the distance increases.

Note that we are not using any amplifiers on the transmitter (just the output signal from the VGA adapter) and that its usage will improve its range. For instance, again at about 8 meters and using the same setup except for a 15 dB amplifier, the result is the lower constellation in Fig. 5. The amplifier even allows the usage of higher order constellations, although naturally the range decreases significantly. For instance, 64QAM was usable only from less than a meter.

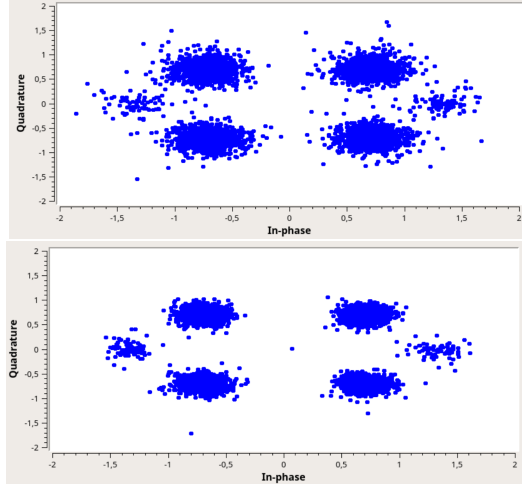


Figure 5. The received constellation about 8 meters away when transmitting the complete 6 MHz signal using the osmo-fl2k. The lower figure corresponds to the case when a 15 dB amplifier is used at the transmitter.

5.2. 1-seg experiments

Let us finally focus on arguably the least expensive possible transmission setup: a 15\$ osmo-fl2k as transmitter and a 25\$ RTL-SDR as a receiver. Note that the main author of the osmo-fl2k project has posted several very interesting examples on his github repository ([Markgraf](#)), including a DVB-T transmission that may be received by an RTL-SDR. However, in order to achieve this, the RTL-SDR has to be used as a DVB-T USB dongle (its original purpose) and not an SDR (recall that the RTL-SDR has a maximum bandwidth of about 3 MHz, and a DVB-T channel is at least 5 MHz wide).

We are interested in using the RTL-SDR as an SDR device and receive and decode the ISDB-T signal. This enables all the possibilities brought by an SDR-based system. In our particular case we were interested in using it in the classroom (where each student may use a cheap dongle), but again, several more non-academic uses are possible. This is where the 1-seg feature of the standard comes in handy: a single segment spanning about 428 kHz of bandwidth (comprising 432 sub-carriers in mode 3), thus fitting perfectly into the RTL-SDR capabilities, and which may be decoded on its own.

To this end, we have implemented the equivalent of all the receiver's blocks for a 1-seg only reception. In particular, the sample rate we used was $f_{IFFT}^{1seg} = 64/63$ MHz (1/8th of the full-seg f_{IFFT}), followed by a 250 kHz bandwidth low-pass filter, and we used an FFT with size 1024 (and thus expect 592 null sub-carriers). Although theoretically a 32/63 MHz sampling rate would be possible, the RTL-SDR does not support sampling rates between 300

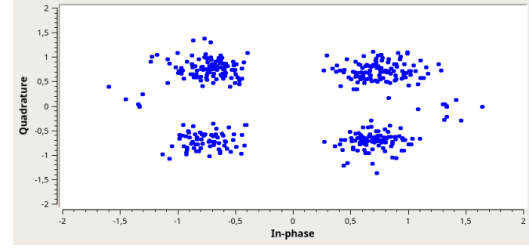


Figure 6. The received constellation about 20 meters away when transmitting the 1-seg signal using the osmo-fl2k (no amplifier).

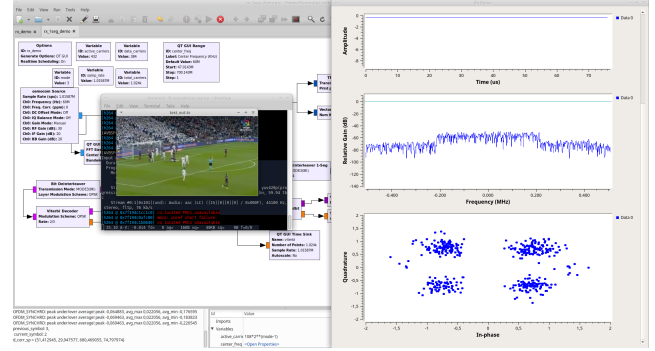


Figure 7. The complete 1-seg receiver at work, including the video decoding using `ffplay`.

and 900 kHz .

Moreover, as we are only interested in the 1-seg signal, we have also implemented all of the transmitter's block for a 1-seg only transmission. The idea is to minimize processing on the transmitter's host PC, but specially focus what little power the osmo-fl2k can transmit into the 428 kHz of bandwidth that interest us here. This results in a range that is significantly larger than in the previous subsection. For instance, using the same parameters as before (but without any amplification), the received constellation at about 20 m from the transmitter is shown in Fig. 6 (we have thus roughly doubled the range). The complete system at work, including the video decoding using `ffplay`, may be seen in Fig. 7.

6. Conclusions

We presented the implementation in GNU Radio of a complete ISDB-T transmitter. All of the configurable parameters of the standard have been verified to work correctly both using an SDR receiver and commercial TVs, using for example an USRP B200 as transmitter.

We were particularly interested in testing whether a cheap osmo-fl2k was capable of transmitting the signal. Our results were successful, obtaining a range of about 10 m with-

out using any amplification and a simple dipole, although we had to use a relatively low carrier frequency of 80 MHz, as the transmitted power in larger harmonics (necessary to reach TV channels in the UHF band) was in our experience very low. We were successful transmitting to a TV using for instance channel 7 (centered around 173 MHz, and thus using the first harmonic), but the signal was already weak and the TV decoding very unstable. This means that the osmo-fl2k may be used with an SDR in mind as a receiver. In any case, this constitutes an extremely cheap setup, where for less than 50 \$ a full digital TV transceiver implemented in SDR may be executed.

As future work, since the project is built around GNU Radio 3.7, a migration to 3.8 is underway. However, ultimately the objective is to integrate the whole project into the GNU Radio core (as part of its gr-dtv module).

Acknowledgements

We would like to thank Pablo Belzarena, Gabriel Gómez-Sena, Víctor González-Barbone, Javier Henrández and Santiago Castro for working with us in previous versions of gr-isdbt.

References

- GNU Radio. The free & open software radio ecosystem. URL <http://gnuradio.org>.
- RTL-SDR (RTL2832U) and software defined radio news and projects. Also featuring Aircspy, HackRF, FCD, SDRplay and more. URL <http://www.rtl-sdr.com>.
- Association of Radio Industries and Businesses (ARIB). STD-B31. Transmission System for Digital Terrestrial Television Broadcasting., 2014. URL www.arib.or.jp/english/html/overview/doc/6-STD-B31v2_2-E1.pdf.
- Ettus Research. USRP B200 SDR Kit - Single Channel Transceiver (70 MHz - 6GHz). URL <https://www.ettus.com/all-products/ub200-kit/>.
- FFmpeg development team. FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. URL <http://ffmpeg.org/about.html>.
- Flores-Guridi, P. and Larroca, F. The isdb-t multiplex frame pattern explained. In *2017 IEEE URUCON*, pp. 1–4, 2017.
- Ikusi Multimedia. FLASHD HDT511V Datasheet. URL <https://www.ikusi.tv/en/products/flashd-hdt511v>.
- Instituto Nacional de Estadística. Microdatos de la encuesta continua de hogares 2019. 2019. URL http://www.inecub.uy/c/document_library/get_file?uuid=590504fd-5f0d-42ff-a94c-fba714fb3f65&groupId=10181.
- ISO 13818-1:2000(E). ISO/IEC 13818-1: Information Technology – Generic Coding of Moving Pictures and Associated Audio: Systems. Standard, International Organization for Standardization, Geneva, Switzerland, December 2000.
- Kuhn, Markus G. Compromising emanations: eavesdropping risks of computer displays. Technical Report UCAM-CL-TR-577, University of Cambridge, Computer Laboratory, December 2003. URL <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-577.pdf>.
- Larroca, Federico, Flores-Guridi, Pablo, Gómez, Gabriel, González-Barbone, Víctor, and Belzarena, Pablo. An open and free isdb-t full seg receiver implemented in gnu radio. In *Wireless Innovation Forum Conference on Wireless Communications Technologies and Software Defined Radio (WinnComm 16)*, Reston, Virginia, USA, 15-17 mar, pp. 1–10, 2016.
- Marinov, Martin. Remote Video Eavesdropping Using a Software-Defined Radio Platform. Master’s thesis, University of Cambridge, United Kingdom, 2014.
- Markgraf, Steve. Example flowgraphs for osmo-fl2k. URL <https://github.com/steve-m/fl2k-examples>.
- Markgraf, Steve. osmo-fl2k, 2018. URL <https://osmocom.org/projects/osmo-fl2k/wiki>.
- Tektronix. PCR Measurements. Technical document, Tektronix, 2003.
- Uehara, M. Application of MPEG-2 Systems to Terrestrial ISDB (ISDB-T). *Proceedings of the IEEE*, 94(1):261–268, Jan 2006. ISSN 0018-9219. doi: 10.1109/JPROC.2005.859695.
- Van Eck, Wim. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security*, 4(4):269–286, 1985.