Atmel

APPLICATION NOTE

AT02594: Smart Reduced Power Consumption Techniques

Atmel MCU Wireless

Features

- Supported in Atmel® AT86RF233, ATmega2564/1284/644RFR2, and ATmega256/128/64RFR2 devices
- PES - PLL Energy Saving
- SRT - Smart Receiving Technology
- ERD - Extended Receiving Desensitizing
- TPH - Automated TX Power Handling
- PAM - PAN Address Match Recognition
- Miscellaneous Power Reduction Functions
  - Dynamic Frame Buffer Protection period power save
  - Random back-off period power save
  - TX_ARET and RX_AACK wait time

Description

Reduced Power Consumption (RPC) is a set of self-contained, self-calibrating, and adaptive power reduction scheme available in Atmel transceivers which helps in reducing the device power consumption further below the normal power consumption. This application note describes about the various RPC modes available in megaRFR2 [1] and AT86RF233 [2] devices. The document also provides a detailed profile showcasing the power consumption, application scenarios to use RPC mode, and benefits of RPC feature available in Atmel transceivers.

Table of Contents

1 Introduction

RPC is an extended feature available in AT86RF233 [2] transceivers and megaRFR2 [1] which help to reduce power consumption further below the normal power consumed by typical IEEE®802.15.4 transceivers. The RPC feature along with extended operating modes makes AT86RF233 [2] and megaRFR2 [1] devices an ideal choice for IEEE802.15.4 based wireless products.

RPC offers a variety of independent techniques and methods to significantly reduce the power consumption over various scenarios. Table 1-1 shows the various RPC modes available in AT86RF233 [2] transceivers and megaRFR2 [1] SoC's.

Table 1-1. Reduced Power Consumption (RPC) Modes

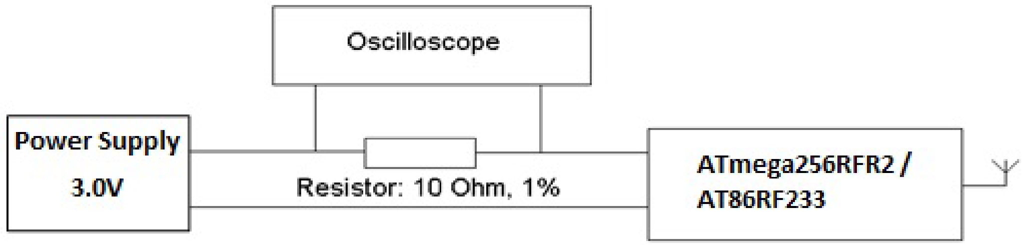
RPC mode	AT86RF233	megaRFR2
2.1 PES – PLL Energy Saving Mode	✓	✓
2.2 SRT – Smart Receiving Technology	✓	✓
2.3 ERD – Extended Receiving Desensitizing	✓	✓
2.4 TPH – Automated TX Power Handling	✓	✗
2.5 PAM – PAN Address Match Recognition	✓	✓
2.6.1 Dynamic Frame Buffer Protection Period Power Save	✓	✗
2.6.2 Random Back-Off Period Power Save	✓	✓
2.6.3 TX_ARET and RX_AACK Wait Time	✓	✓

The focus of this application note is to detail the RPC modes available, their usage, and implementation. This document also provides power profile for usage scenarios, with and without RPC to showcase the achievable power consumption reduction with individual modes.

1.1 Measurement Setup

- Hardware: REB233SMAD-EK kit [3]/ATmega256RFR2 Xplained Pro [4]
- Test setup:
  - Tektronix MSO 4054 Mixed signal oscilloscope (500MHz 2.5GS/s)
  - With 10Ω 1% metal film resistor as shunt resistor
    - REB233SMAD-EK [3]: Across JP1 in REB233SMAD transceiver board
    - ATmega256RFR2-XPRO [4]: Across J101 (VCC_Target and VCC_MCU)
  - External power supply is connected to VTG and GND pins in REB233SMAD-EK kit [3]/J100 - VCC_P3V3 and GND pins in ATmega256RFR2 Xplained Pro board [4]

Figure 1-2. Oscilloscope Setup



Oscilloscope measurements are expected to have offset of around 0.2mA. This error was calculated by measuring a constant current using both multimeter and oscilloscope.

Note: All measurement results are typical values.

2 RPC Modes

The different mode of the RPC can be enabled or disabled by configuring the register `TRX_RPC`. Figure 2-1 shows the `TRX_RPC` register configuration.

Figure 2-1. TRX_RPC Register

7 6	5	4	3	2	1	0
`RX_RPC_CTRL`	`RX_RPC_EN`	`PDT_RPC_EN`	`PLL_RPC_EN`	`XAH_TX_RPC_EN`	`IPAN_RPC_EN`	Reserved
Activate max / min power saving behavior in Smart Receiving mode	Enable/ Disable Smart Receiving mode	Enable / Disable Extended re-ceiver desensi-tizing	Enable / Disable PLL Energy saving	Enable / Disable Automated TX Power handling	Enable / Disable PAN Address match recognition	Should be always set to 1
Applicable States						
`RX_ON` state, `RX_AACK_ON` state and `TX_ARET` mode, when waiting for an ACK		`RX_ON` state, `RX_AACK_ON` state and `TX_ARET` mode, when waiting for an ACK	`PLL_ON` state and `TX_ARET_ON` state	`RX_AACK_ON` state	`RX_AACK_ON` state	

The maximum power saving in RPC can be enabled by setting 0xFF to `TRX_RPC` register. The RPC features can be disabled by setting `TRX_RPC` register to 0xC1 or 0x01. The reduced power saving in miscellaneous mode is enabled by default and it is not possible disable the power saving in these modes.

The code snippet below shows the ASF – AVR®2025 TAL [5] (component) API used for enabling the maximum power saving in RPC. Though the code snippet is provided for Atmel IEEE 802.15.4 MAC SDK, it is possible to implement in similar way in other stacks as well.

Code Snippet

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC, 0xFF); /* Enable all RPC features. */
```

2.1 PES – PLL Energy Saving Mode

PLL energy saving mode helps to reduce the current consumption of the device by automatically entering into power save mode immediately after the PLL calibration. A state change towards PLL_ON, TX_ARET_ON, RX_ON/RX_AACK_ON, and channel switch causes a PLL wake up and starts PLL calibration run. After finishing such PLL calibration, the PLL automatically enters to power save mode if PES mode is enabled. The PES mode can be enabled by setting the 'PLL_RPC_EN' bit in TRX_RPC register and is applicable to PLL_ON and TRX_ARET_ON state.

With PES mode enabled, the power consumed in PLL_ON state reduces from 5.2mA typically to 450mA. After the initial PLL run within the selected channel, the PLL will be disabled on PLL locked channel. The PES mode when enabled save power by automatically entering into power save mode.

2.1.1 Current Profile

Figure 2-2 and Figure 2-3 shows the comparison of power consumption when there is a state change from TRX_OFF state to PLL_ON state without and with PES mode enabled.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for setting the PLL_RPC_EN bit in TRX_RPC register and for changing the state of the transceiver.

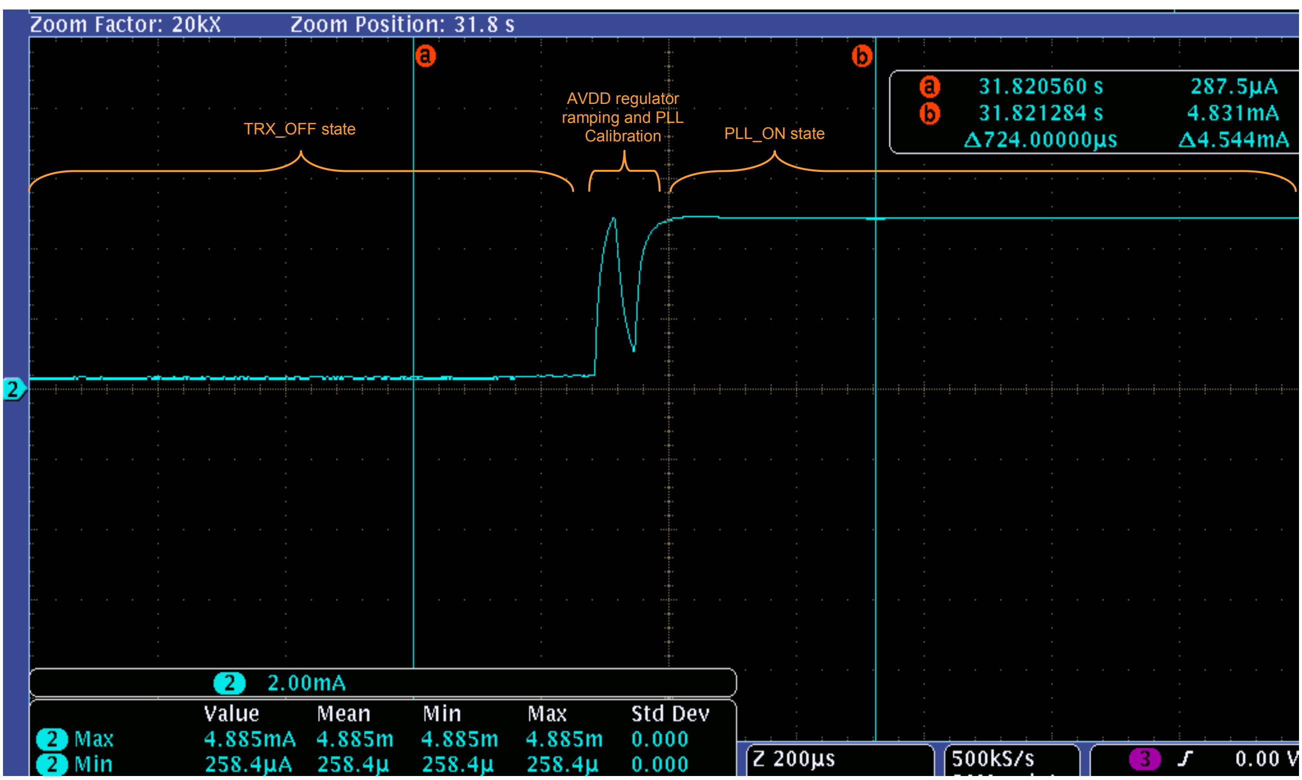
Code Snippet

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0xC9); /* Enable PES mode in RPC. */
set_trx_state(CMD_PLL_ON);
```

- **TRX_OFF to PLL_ON state**

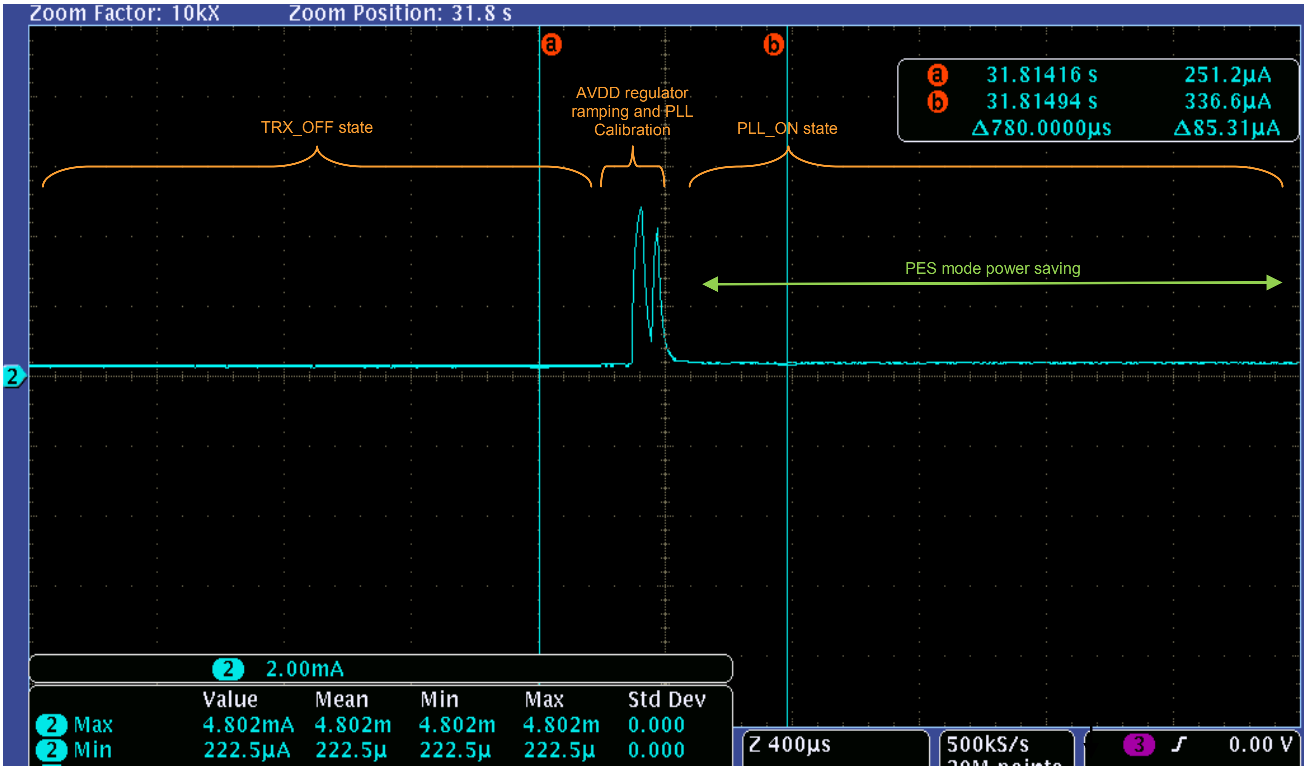
Figure 2-2 shows the power consumption of the device when the device changes its state from TRX_OFF to PLL_ON state. Here the device is consuming 5mA (4.83mA + Oscilloscope_Offset (200µA)) in PLL_ON state.

Figure 2-2. State Change from TRX_OFF to PLL_ON with RPC Disabled



[Figure 2-3](#) shows the same with `PLL_RPC_EN` bit in `TRX_RPC` register enabled. On enabling PES mode, the `PLL_ON` state current consumption reduces to 450µA.

Figure 2-3. State Change from TRX_OFF to PLL_ON with PES Mode Only Enabled



- **TRX_OFF state to RX_ON/RX_AACK_ON**

[Figure 2-4](#) and [Figure 2-5](#) show the comparison of power consumption when there is a state change from `TRX_OFF` state to `RX_ON/RX_AACK_ON` state without and with PES mode enabled.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for setting the `PLL_RPC_EN` bit in `TRX_RPC` register and for changing the state of the transceiver.

Code Snippet

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0xC9); /* Enable PES mode in RPC. */
set_trx_state(CMD_RX_ON);
```

[Figure 2-4](#) shows the power consumption of the device when the device changes its state from `TRX_OFF` to `RX_ON/RX_AACK_ON` state.

Figure 2-4. State Change from TRX_OFF to RX_ON with RPC Disabled

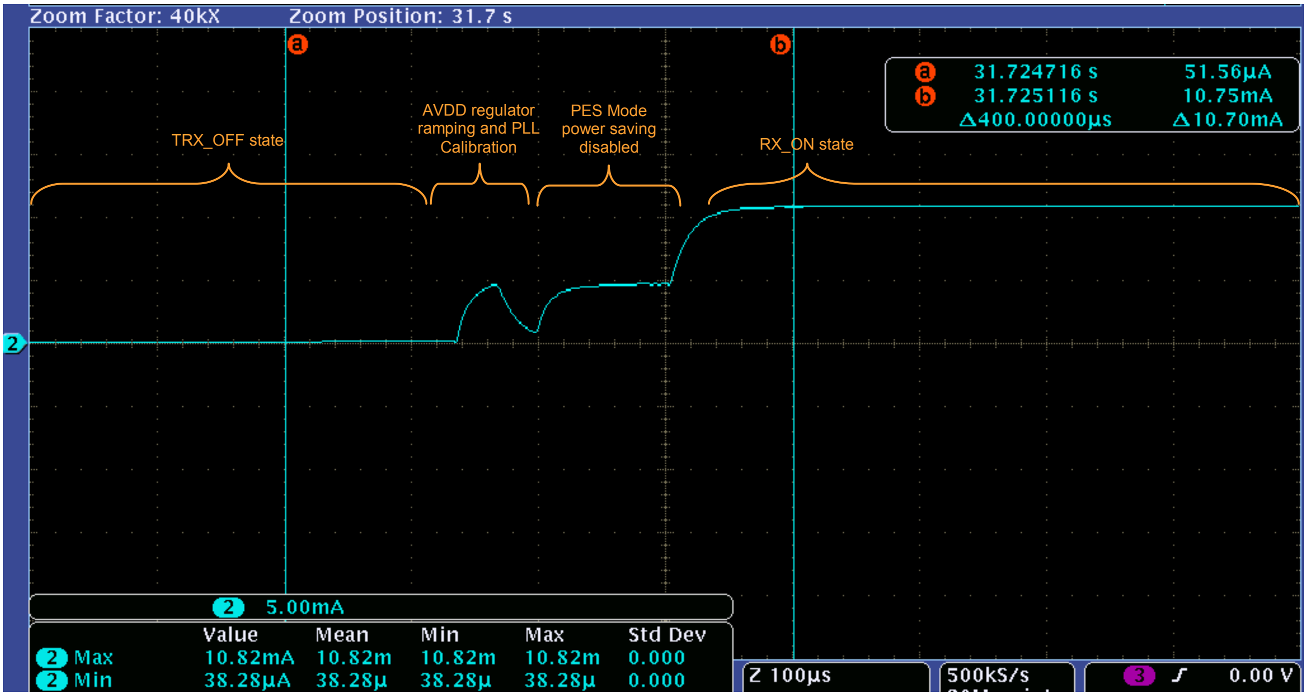
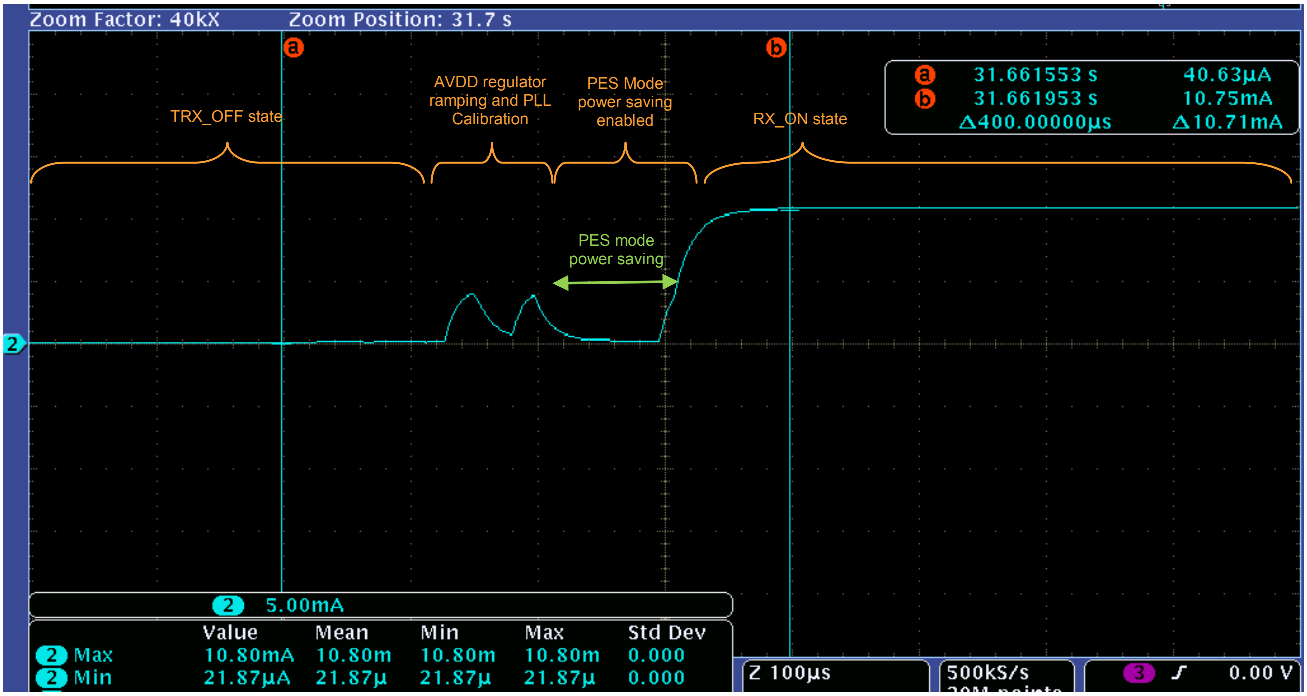


Figure 2-5 shows the same with `PLL_RPC_EN` bit in `TRX_RPC` register enabled. The device goes to power save mode during the PLL calibration period.

Figure 2-5. State Change from TRX_OFF to RX_ON with PES Mode Only Enabled



Note: In ASF-MAC SDK, the state change from `TRX_OFF` to `RX_ON` using the API set_trx_state will first set the transceiver to `PLL_ON` state and then to `RX_ON`/`RX_AACK_ON` state.

- **TRX_OFF state to TX_ARET_ON**

Figure 2-6 and Figure 2-7 show the comparison of power consumption when there is a state change from TRX_OFF state to TX_ARET_ON state without and with PES mode enabled.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API is used for setting the PLL_RPC_EN bit in TRX_RPC register and for changing the state of the transceiver.

Code Snippet

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0xC9); /* Enable PES mode in RPC. */
set_trx_state(CMD_RX_ON);
```

Figure 2-6 shows the power consumption of the device when the device changes its state from TRX_OFF to TX_ARET_ON state.

Figure 2-6. State Change from TRX_OFF to TX_ARET_ON with RPC Disabled

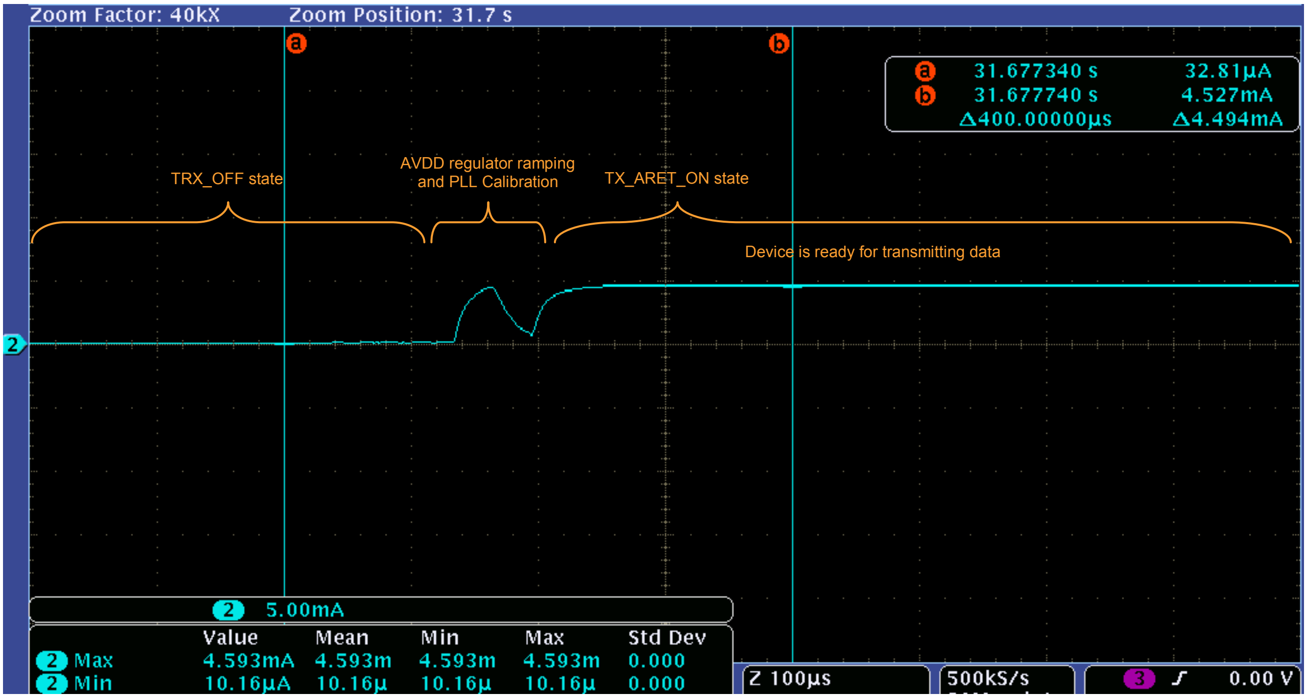
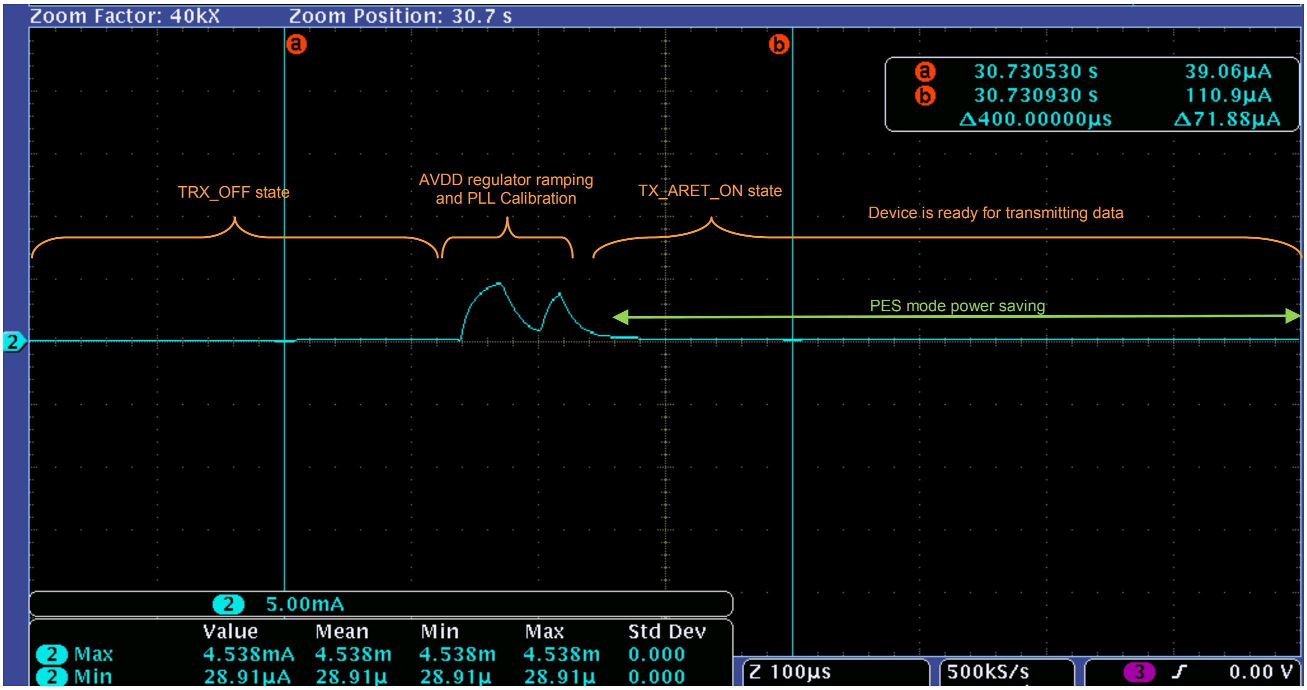


Figure 2-7 shows the same with PLL_RPC_EN bit in TRX_RPC register enabled.

Figure 2-7. State Change from TRX_OFF to TX_ARET_ON with PES Mode Only Enabled



2.2 SRT – Smart Receiving Technology

Smart receiving technology when enabled reduces power consumption by periodically enabling and disabling the transceiver while listening for incoming frame. Depending on the channel noise, traffic, environmental conditions, the SRT mode helps in saving up to 50% of the current consumption in `RX_ON` and `RX_AACK_ON` modes. The SRT mode is enabled by setting the `RX_RPC_EN` bit in `TRX_RPC` register. When SRT is enabled, the SRT is also active in `TX_ARET` mode when waiting for a requested acknowledgment. Setting the `RX_RPC_CTRL` bits activates the maximum power saving behavior of the SRT mode.

In this mode 1dB sensitivity loss is expected. The application must disable the SRT mode when performing the following actions

- RSSI Measurement
- Random Number Generation

During CCA and ED Scan, the transceiver automatically disables the SRT mode. If antenna diversity is enabled, the SRT mode cannot achieve the power saving in receiving mode as expected. SRT mode helps in reducing power consumption of the devices significantly which has '`Receiver On When Idle`' mode enabled.

2.2.1 Current Profile

Figure 2-8 and Figure 2-9 shows the comparison of power consumption in `RX_ON`/`RX_AACK_ON` mode without and with SRT enabled.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for setting the `RX_RPC_EN` and `RX_RPC_CTRL` bits and to set `RX_AACK_ON` state.

Code Snippet

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0xE1);   /* Enable SRT mode in RPC. */
set_trx_state(CMD_RX_AACK_ON);
```

Figure 2-8 shows the power consumption of the device in RX_ON/RX_AACK_ON mode with SRT disabled. Here the device consumes 11mA of current during RX_ON/RX_AACK_ON state.

Figure 2-8. RX_ON/AACK_ON Mode with RPC Disabled

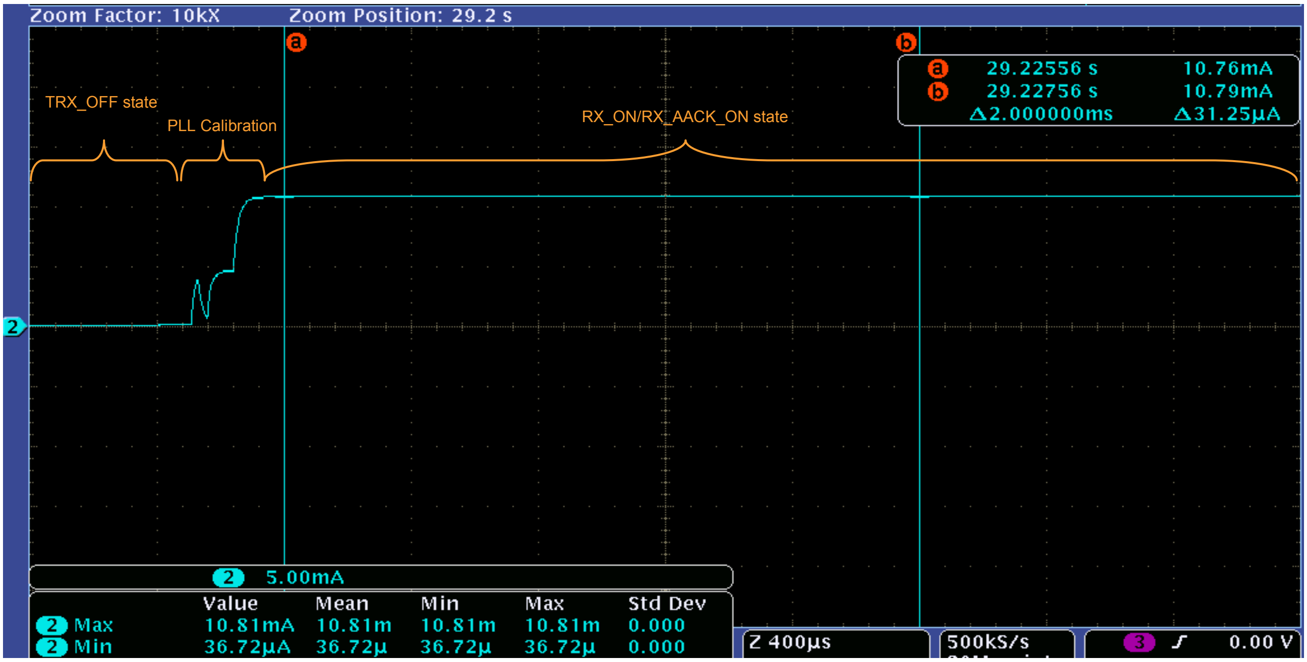


Figure 2-9 shows the power consumption of the same with SRT enabled. Here, when the device is on RX_ON/RX_AACK_ON state, the SRT mode reduces power consumption up to 50% by periodically enabling and disabling the transceiver while listening for incoming frames.

Figure 2-9. RX_ON/RX_AACK_ON Mode with SRT Mode Only Enabled

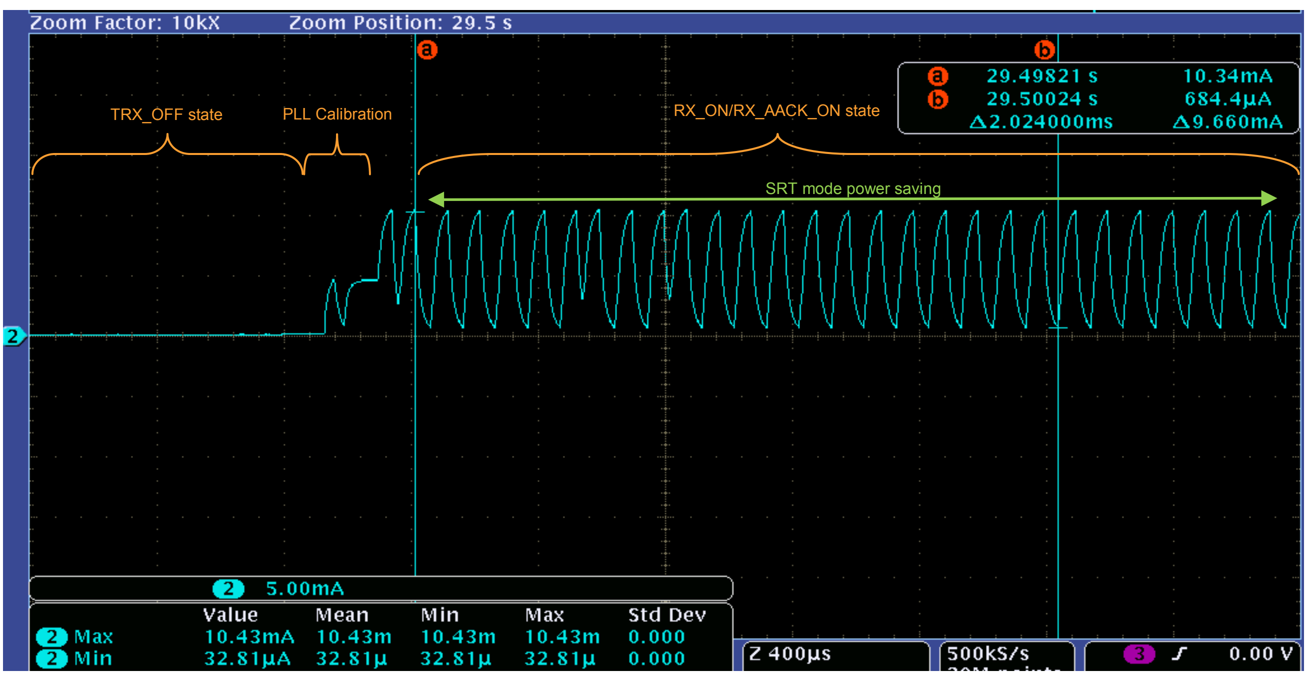


Figure 2-10 and Figure 2-11 shows the minimum and maximum power saving behavior of the transceiver when SRT is enabled. As mentioned above, the power saving in SRT mode is dependent on channel noise, traffic, environmental conditions, etc. Figure 2-10 shows the minimum power saving behavior of SRT and the code snippet below shows the ASF – AVR2025 TAL [5] (component) API for enabling SRT with minimum power saving by setting the register bits `RX_RPC_EN` bit and `RX_RPC_CTRL` bits.

Code Snippet: Enabling SRT Mode with Minimum Power Saving

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0x21);  /*Enable SRT mode with min power saving in RPC.*/
set_trx_state(CMD_RX_AACK_ON);
```

Figure 2-10. SRT Enabled with Minimum Power Saving Activated

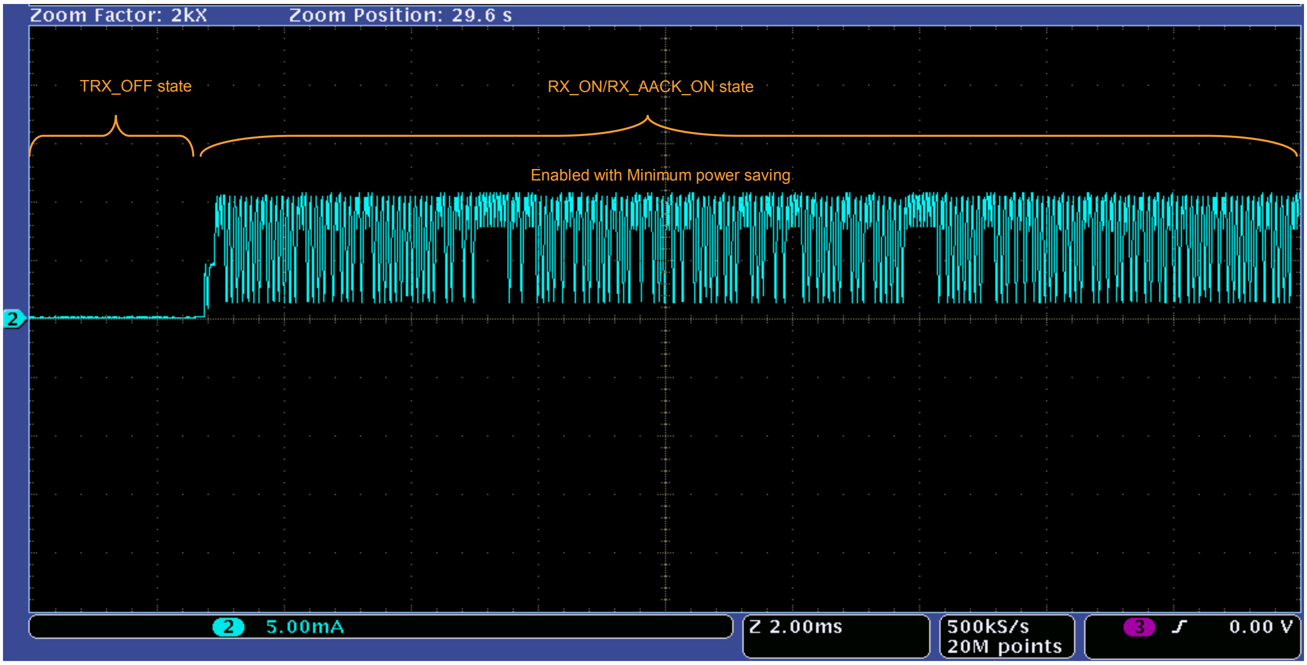


Figure 2-11 shows the maximum power saving behavior of SRT and the code snippet below shows the ASF – AVR2025 TAL [5] (component) API for enabling SRT with maximum power saving.

Code Snippet: Enabling SRT Mode with Maximum Power Saving

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0xE1);  /* Enable SRT mode with max power saving in RPC. */
set_trx_state(CMD_RX_AACK_ON);
```

Figure 2-11. SRT Enabled with Maximum Power Saving Activated

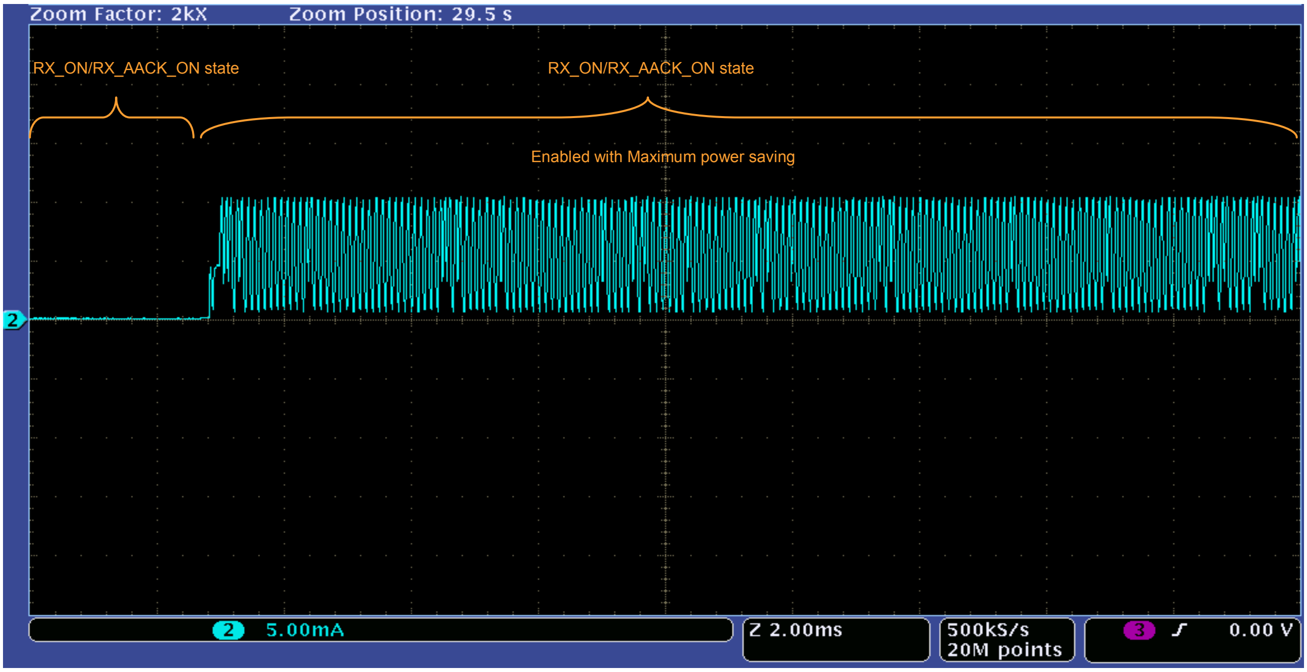
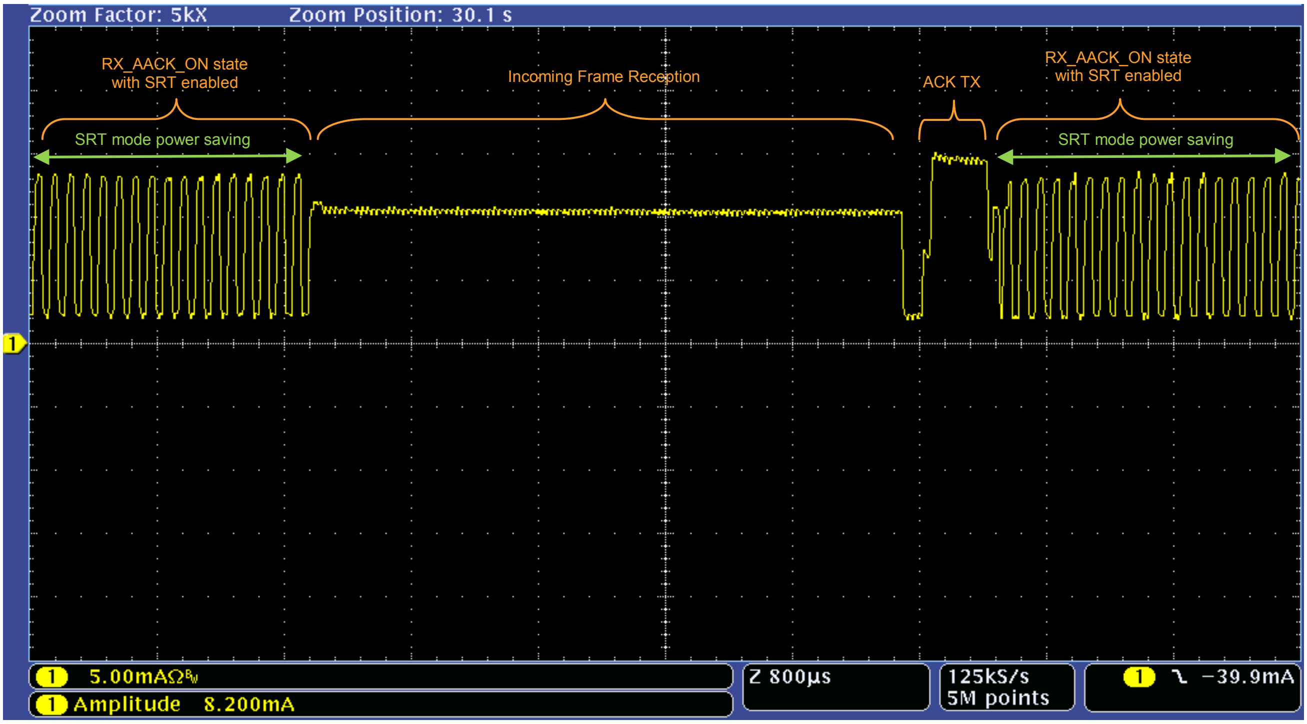


Figure 2-12 shows the current consumption scope plot when the transceiver is receiving an incoming frame and sending the acknowledgement in `RX_AACK_ON` mode.

Figure 2-12. Frame Reception and Automatic ACK Transmission in RX_AACK_ON Mode with SRT Enabled



2.3 ERD – Extended Receiving Desensitizing

AT86RF233 [2] transceivers and megaRFR2 [1] SoC's have an outstanding sensitivity performance of -101dBm. For certain Applications, Environmental condition or for High Data Rate modes it may be useful to decrease the receiver sensitivity. This can be achieved by adjusting the receiver sensitivity threshold using register bits `RX_PDT_LEVEL` in `RX_SYN` Register (0x15). The Extended Receiver Desensitizing mode (ERD) in RPC helps in reducing the power consumption of the device when receiver sensitivity reduced by configuring `RX_PDT_LEVEL` bits. The ERD can be enabled by setting the `PDT_RPC_EN` bit in `TRX_RPC` register and is applicable to `RX_ON`, `RX_AACK_ON`, and `TX_ARET` states.

Table 2-1 shows the `RX_PDT_LEVEL` register value and the receiver desensitization threshold value (dBm). The receiver sensitivity threshold value is calculated by the equation:

- `RX_THRES > RSSI_BASE_VAL+3*(RX_PDT_LEVEL-1), for RX_PDT_LEVEL>0`

Setting `RX_PDT_LEVEL = 0x08` requires special attention. In contrast to the above equation, `RX_PDT_LEVEL = 0x08` will reduce the sensitivity to -80dBm. Without ERD enabled, `RX_PDT_LEVEL` bits set to greater than zero will reduce the current consumption by 500µA. With ERD enabled, the current consumption reduces by 0.5mA for `RX_PDT_LEVEL` values from 0x01 to 0x07, 2mA for `RX_PDT_LEVEL` equal to 0x08 and 2.5mA for `RX_PDT_LEVEL` from 0x09 to 0x0F.

Table 2-1. Receiver Desensitization vs. Power Consumption

RX_PDT_LEVEL Register Value	Receiver Desensitization Threshold Level in dBm		Current consumption with ERD disabled (mA)	Current consumption with ERD enabled (mA)
	AT86RF233	megaRFR2		
0x00	Maximum Rx sensitivity (RSSI value not considered)		11	11
0x01	-91	-90	10.6	10.5
0x02	-88	-87	10.6	10.6
0x03	-85	-84	10.6	10.5
0x04	-82	-81	10.6	10.6
0x05	-79	-78	10.6	10.6
0x06	-76	-75	10.6	10.5
0x07	-73	-72	10.6	10.5
0x08	**-80**	**-80**	10.6	**9**
0x09	-67	-66	10.6	8.5
0x0A	-64	-63	10.6	8.5
0x0B	-61	-60	10.6	8.5
0x0C	-58	-57	10.6	8.5
0x0D	-55	-54	10.6	8.5
0x0E	-52	-51	10.6	8.5
0x0F	-49	-48	10.6	8.5

Note: The above characterization is taken with ERD enabled and SRT disabled. By enabling the SRT mode, the current consumption can be further reduced up to 50%. (i.e. when `RX_PDT_LEVEL` set to 0x08 with SRT and ERD enabled, the average current consumption will reduce up to 4.5mA, typically 5.1mA to 6.1mA).

In AT86RF233 [2], ERD mode enabled with `RX_PDT_LEVEL` set to 0x08 will reduce the power consumption in receive mode to 9mA for receiver sensitivity of -80dBm. The current consumption can be further reduced by enabling the SRT mode. With `RX_PDT_LEVEL` > 8, ERD and SRT enabled will reduce the effective power consumption to 5.1mA in receive mode. This configuration will reduce power consumption significantly in FFD's which has '`Receiver On When Idle`' mode enabled.

Note: In this configuration; the receiver sensitivity reduces to -79dBm. During CCA and ED scan, the ERD will be disabled automatically.

2.3.2 Current Profile

Figure 2-13 and Figure 2-14 shows the comparison of power consumption in `RX_ON/RX_AACK_ON` mode without and with ERD enabled.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for enabling ERD mode and setting `RX_PDT_LEVEL` to 0x08.

Code Snippet: Enabling ERD Mode and Setting `RX_PDT_LEVEL` to 0x08

```
set_trx_state(CMD_TRX_OFF);
pal_trx_reg_write(RG_TRX_RPC,0xD1);          //enable ERD
tal_set_rx_sensitivity_level(0x08);          //RX_PDT_LEVEL set to 0x08
set_trx_state(CMD_RX_AACK_ON);
```

Figure 2-13 shows the power consumption of the device in `RX_ON/RX_AACK_ON` mode with ERD disabled and `RX_PDT_LEVEL` set to 0x00. Here the device consumes 11mA of current during `RX_ON/RX_AACK_ON` state.

Figure 2-13. RX_ON Mode with ERD Disabled and RX_PDT_LEVEL = 0x00

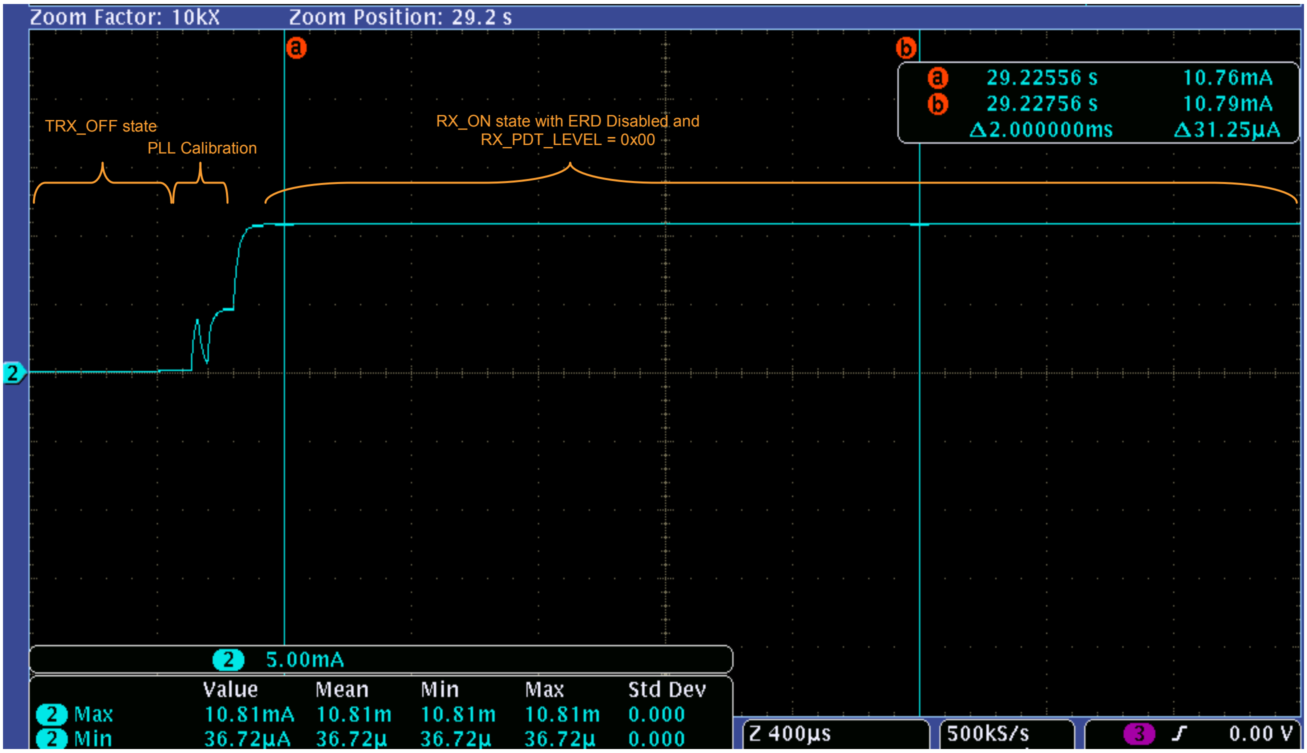


Figure 2-14 shows the power consumption of the same with ERD only enabled and `RX_PDT_LEVEL` set to 0x08. Here, when the device is on `RX_ON/RX_AACK_ON` state, the ERD mode reduces power consumption 9mA.

Figure 2-14. RX_ON Mode with ERD Enabled and RX_PDT_LEVEL = 0x08

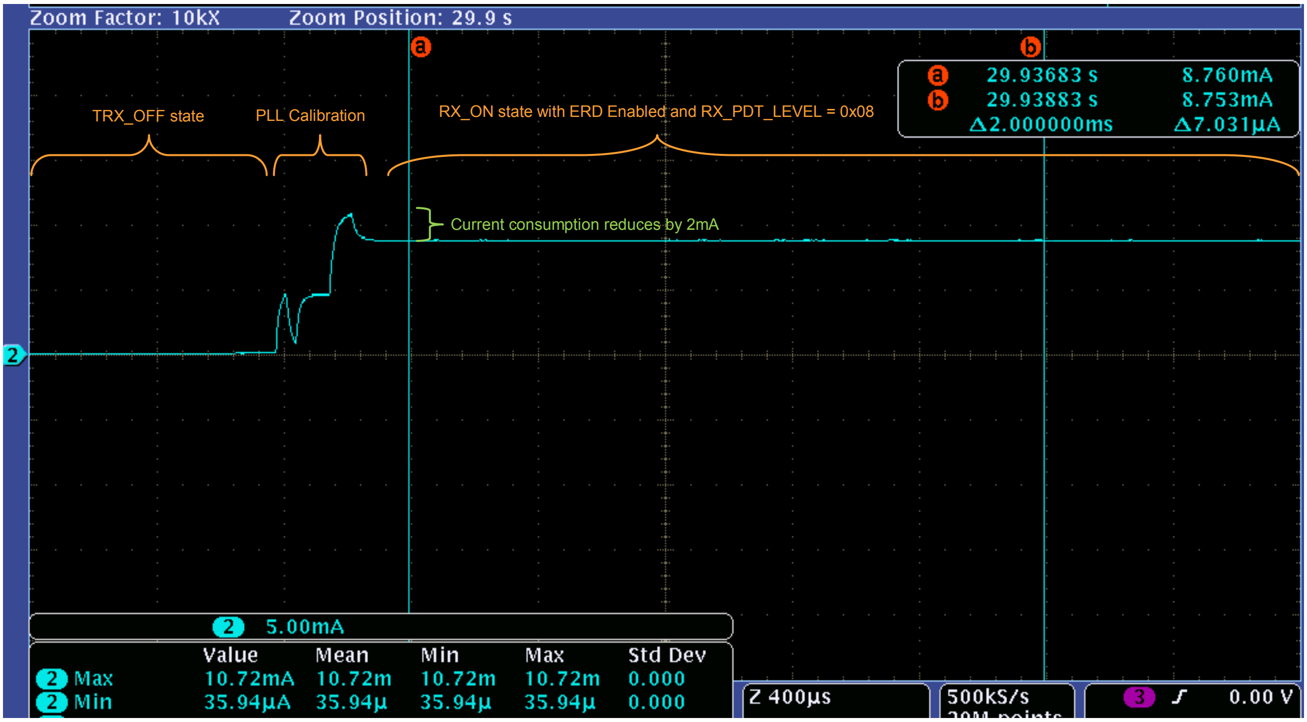
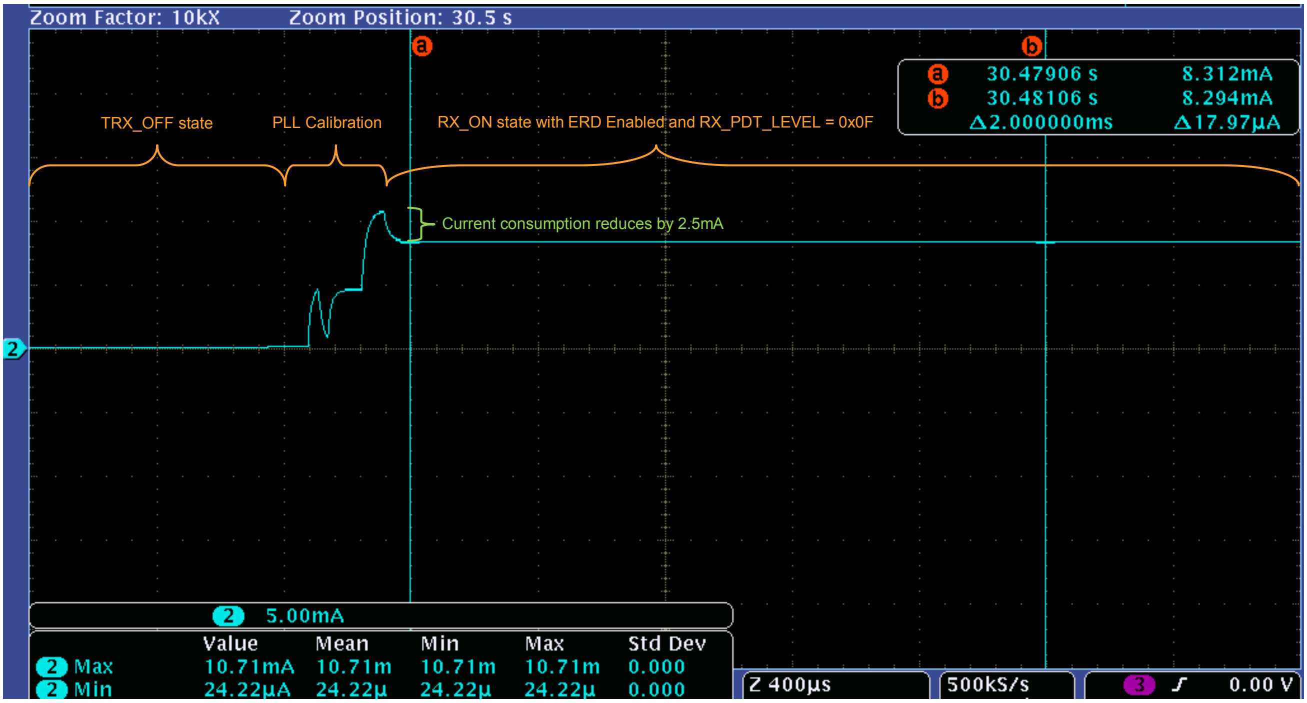


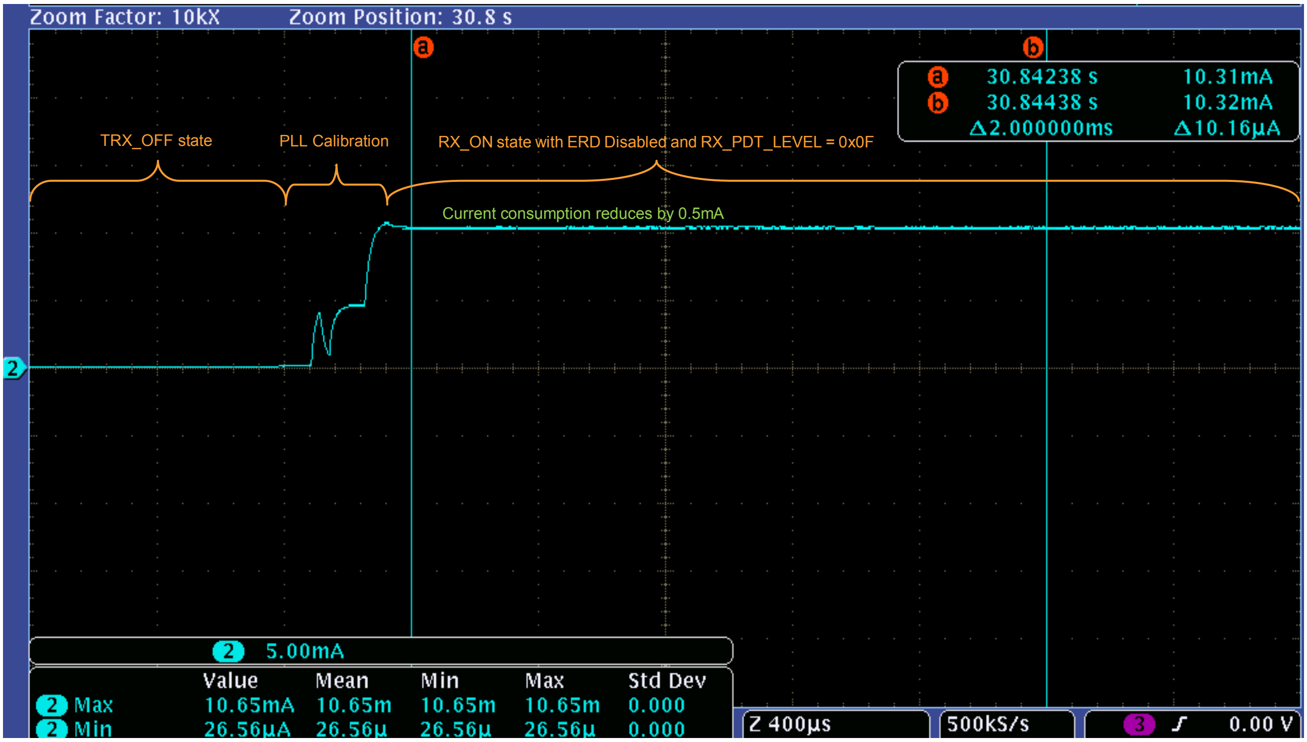
Figure 2-15 shows the power consumption with ERD only enabled and `RX_PDT_LEVEL` set to 0x0F. Here, when the device is on `RX_ON`/`RX_AACK_ON` state, the ERD mode reduces power consumption 8.5mA.

Figure 2-15. RX_ON Mode with ERD Enabled and RX_PDT_LEVEL = 0x0F



[Figure 2-16](#) shows the power consumption with ERD disabled and `RX_PDT_LEVEL` set to 0x0F. Without ERD enabled, `RX_PDT_LEVEL` bits set to greater than zero will reduce the current consumption by 0.5mA. Here, when the device is on `RX_ON/RX_AACK_ON` state and reducing the receiver sensitivity reduces the power consumption by 0.5mA.

Figure 2-16. RX_ON Mode with ERD Disabled and RX_PDT_LEVEL = 0x0F



2.4 TPH – Automated TX Power Handling

Automated Tx Power handling mode helps to reduce the current consumption of device by automatically adapting the transmission power according to the combination of received frame ED and LQI values. This feature is present only in AT86RF233 [2] and not available in megaRFR2 [1] devices. When a frame is received successfully with ED > -77dBm and LQI > 224, the Tx output power is automatically adeptly reduced. The minimum transmit power is -17dBm (ED > -45dBm and LQI > 224) whereas the maximum transmit power will be the value which was configured in `TX_PWR` register.

If the first frame transmission fails with the reduced Tx output power as set by the TPH, the next frame retry starts with the maximum Tx output power.

2.4.1 Current Profile

Figure 2-17, Figure 2-18, and Figure 2-19 show the power consumption comparison of the Tx power handling mode disabled, enabled with nodes kept nearby and far away. Here the device is configured to transmit at maximum power.

Figure 2-17 shows that the device is transmitting the acknowledgement at maximum power when TPH is disabled whereas Figure 2-18 shows that device automatically reduces the power of ACK frame since the nodes are kept nearby and Figure 2-19 shows that the ACK frame is transmitted at full power since the nodes are kept far away.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for setting the device short address, pan id and enabling TPH mode.

Code Snippet: Enabling TPH Mode

```
set_trx_state(CMD_TRX_OFF);
/* Set short address. */
uint16_t temp = SOURCE_ADDRESS;
tal_pib_set(macShortAddress, (pib_value_t *)&temp);
/* Set PAN ID. */
temp = DEFAULT_PAN_ID;
tal_pib_set(macPANId, (pib_value_t *)&temp);
pal_trx_reg_write(RG_TRX_RPC,0xE5);  //enable SRT and TPH modes
set_trx_state(CMD_RX_AACK_ON);
```

Figure 2-17 shows the current consumption of the device with TPH disabled where both transmitting and receiving nodes are kept nearby.

Figure 2-17. TPH Mode Disabled (SRT Enabled) with Nodes kept Nearby

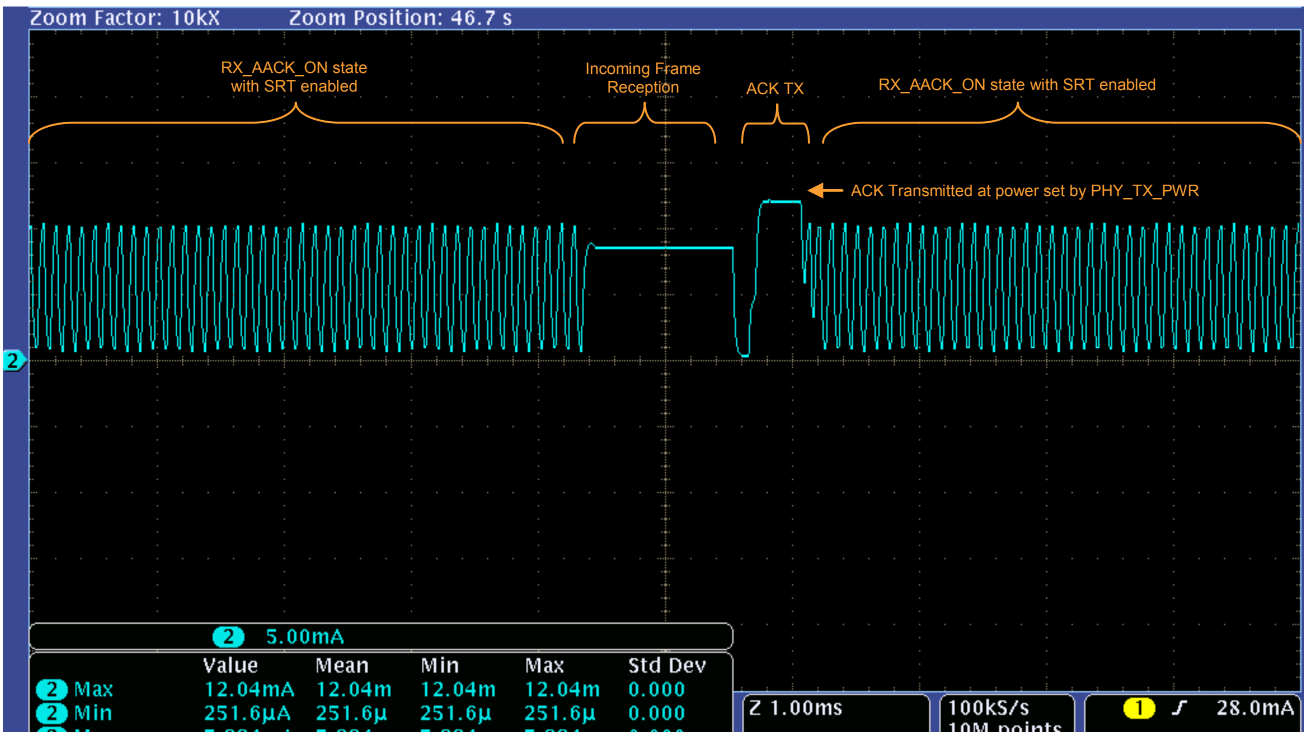


Figure 2-18 shows that device automatically reduces the power of ACK frame; here the nodes are kept nearby.

Figure 2-18. TPH Mode Enabled (SRT Enabled) with Nodes kept Nearby

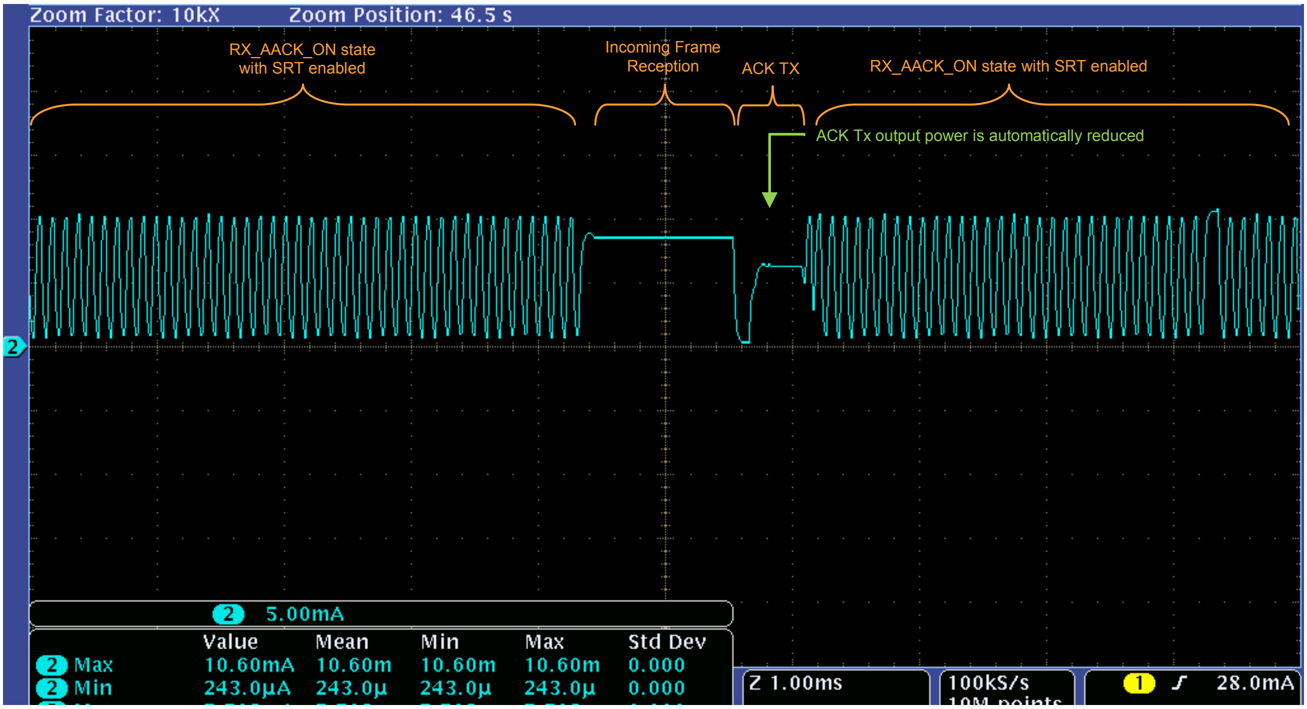
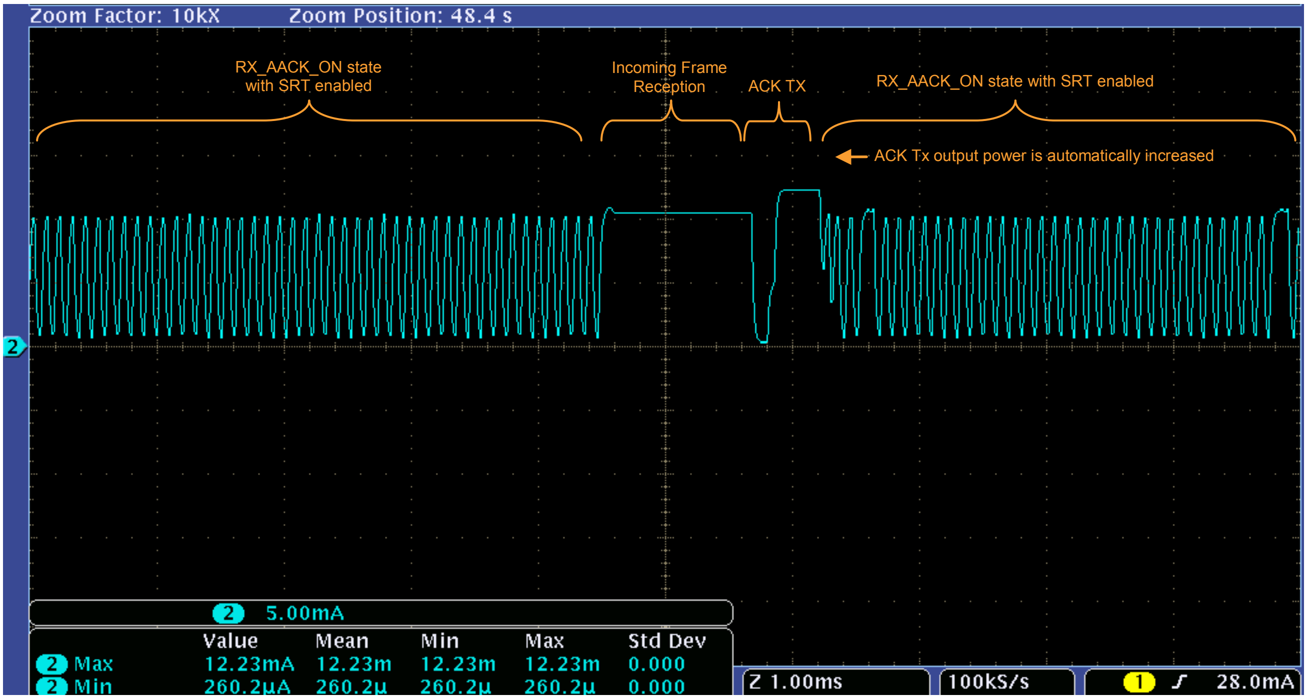


Figure 2-19 shows that the ACK frame is transmitted at full power; here the nodes are kept far away.

Figure 2-19. TPH Mode Enabled (SRT Enabled) with Nodes kept Far Away



2.5 PAM – PAN Address Match Recognition

PAN Address Match Recognition mode (PAM) when enabled reduces power consumption by automatically entering the power save mode for the remaining frame and ACK period, if the PAN Address matches and destination address does not matches. If PAN address does not match, a new listen period starts immediately. This feature is helpful in big network with lots of nodes.

2.5.1 Current Profile

Figure 2-20 and Figure 2-21 show the power consumption comparison of the PAM mode during frame reception with same PAN and different destination.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for setting the device short address, pan id, and enabling PAM and SRT mode.

Code Snippet: Enabling PAM Mode

```
set_trx_state(CMD_TRX_OFF);
/* Set short address. */
uint16_t temp = SOURCE_ADDRESS;
tal_pib_set(macShortAddress, (pib_value_t *)&temp);
/* Set PAN ID. */
temp = DEFAULT_PAN_ID;
tal_pib_set(macPANId, (pib_value_t *)&temp);
pal_trx_reg_write(RG_TRX_RPC,0xE3);  //enable SRT and PAM modes
set_trx_state(CMD_RX_AACK_ON);
```

Figure 2-20 shows the current consumption of the device with PAM disabled, the device stays in receive mode during the frame reception period.

Figure 2-20. PAM Mode Disabled (SRT Enabled) Frame Reception with Same PAN and Different Destination Address

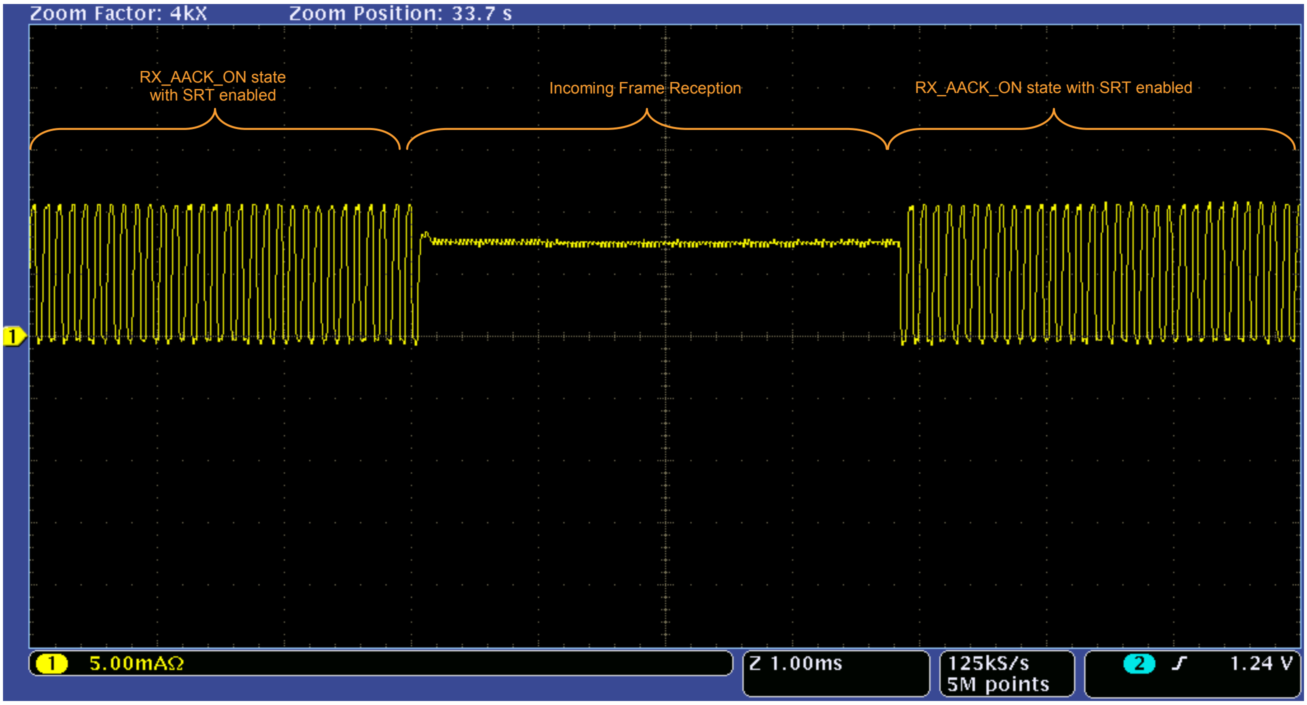


Figure 2-21 shows the current consumption with PAM enabled, the device automatically enter the power save mode for the remaining frame and ACK period, if the PAN Address matches and destination address does not matches.

Figure 2-21. PAM Mode Enabled (SRT Enabled) Frame Reception with Same PAN and Different Destination Address

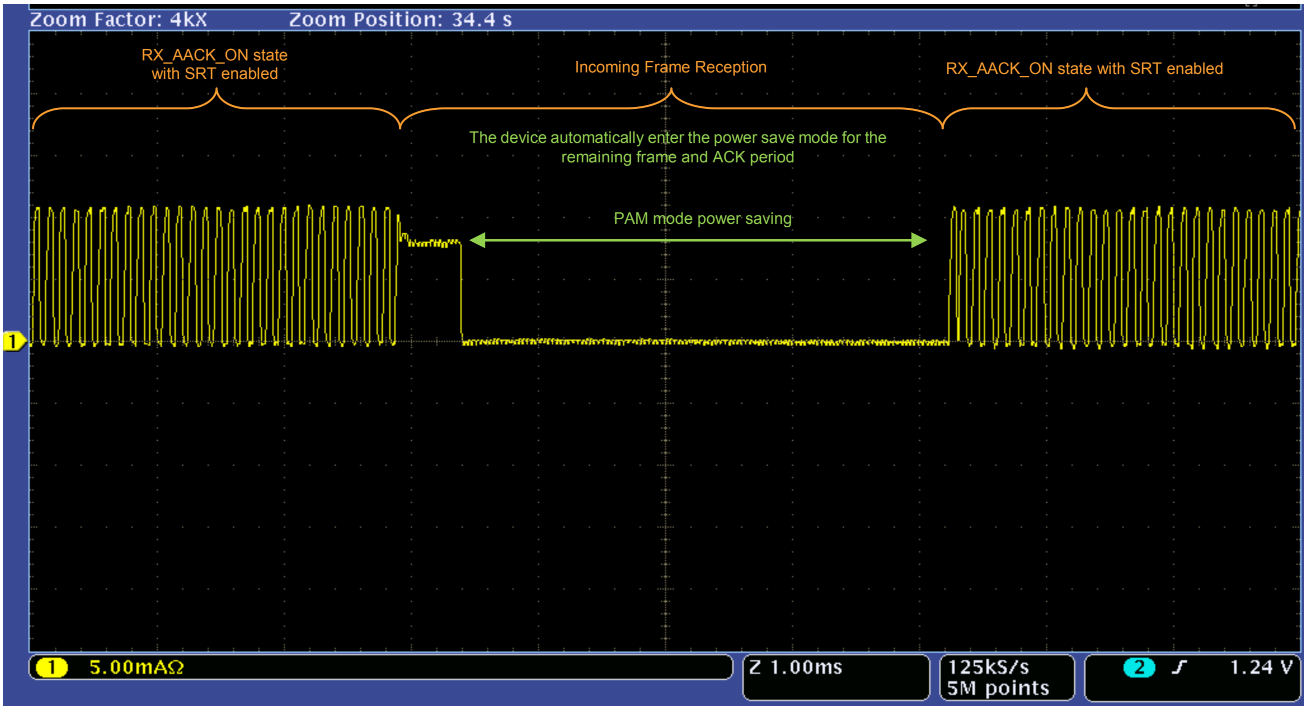
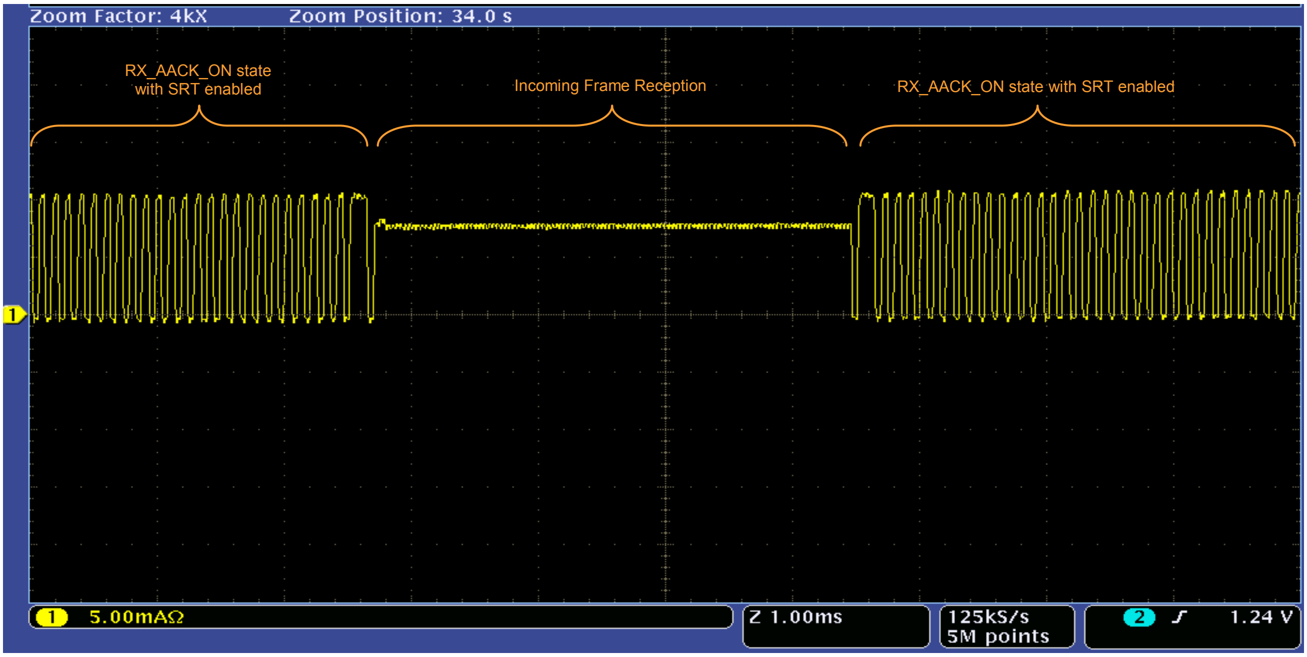


Figure 2-22 and Figure 2-23 shows the power consumption comparison of the PAM mode during frame reception with different PAN address.

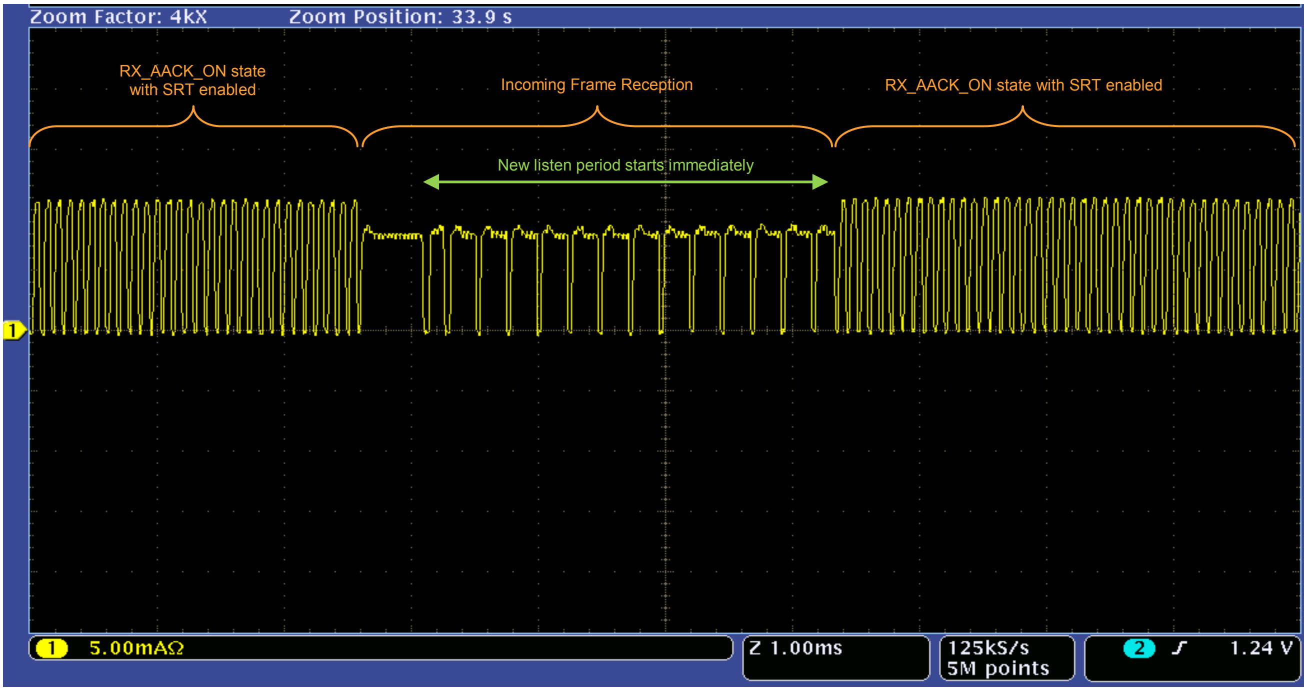
Figure 2-22 shows the current consumption of the device with PAM disabled, the device stays in receive mode during the frame reception period.

Figure 2-22. PAM Mode Disabled (SRT Enabled) Frame Reception with Different PAN Address



[Figure 2-23](#) shows the current consumption with PAM enabled, here the PAN address does not matches and a new listen period has started immediately. In this case, the device is not blocked to receive undesirable frame and is now ready to receive frames from the own network.

Figure 2-23. PAM Mode Enabled (SRT Enabled) Frame Reception with Different PAN Address



2.6 Miscellaneous Power Reduction Functions

In addition to the power saving techniques mentioned in Section 2.1 to 2.5, AT86RF233 and megaRFR2 devices can reduce power consumption further in following periods:

a) Dynamic Frame Buffer protection period (power save is only in AT86RF233).

b) Random back-off period.

c) TX_ARET and RX_AACK wait period.

The power save in these modes will get enabled automatically when these features are enabled and it is not possible to disable the feature alone (with feature enabled).

2.6.1 Dynamic Frame Buffer Protection Period Power Save

Dynamic frame buffer when enabled prevents a newly received valid frame passes to frame buffer until a frame buffer read has ended. This feature when enabled helps to relax timing requirement for a frame buffer read access. In AT86RF233 [2], when dynamic frame buffer protection enabled, the device automatically enters in power save mode until the frame buffer is read. The power save feature is not present in megaRFR2 [1] devices and the device will not enter power save mode.

The code snippet below shows the ASF – AVR2025 TAL [5] (component) API used for enabling Dynamic Frame protection and enabling all the RPC modes.

Code Snippet: Enabling PAM Mode

```
set_trx_state(CMD_TRX_OFF);
pal_trx_bit_write(SR_RX_SAFE_MODE, RX_SAFE_MODE_ENABLE); /* Enable buffer protection mode */
pal_trx_reg_write(RG_TRX_RPC, 0xFF); /* RPC feature configuration. */
set_trx_state(CMD_RX_AACK_ON);
```

Note: It is recommended to disable/enable dynamic frame buffer protection in TRX_OFF state and then switch to RX_ON/RX_AACK_ON state to re-activate SRT.

2.6.1.1 Current Profile

To understand the current consumption in this mode, a long delay is inserted in ASF – AVR2025 TAL [5] (component) API component before the frame buffer is read. Figure 2-24 shows the power consumption in AT86RF233 [2] device. Here the device enters into a long delay routine before the frame is read and the device automatically switches to power save mode until the frame buffer is read.

Figure 2-24. RPC On with Frame Buffer Read After a Delay in AT86RF233

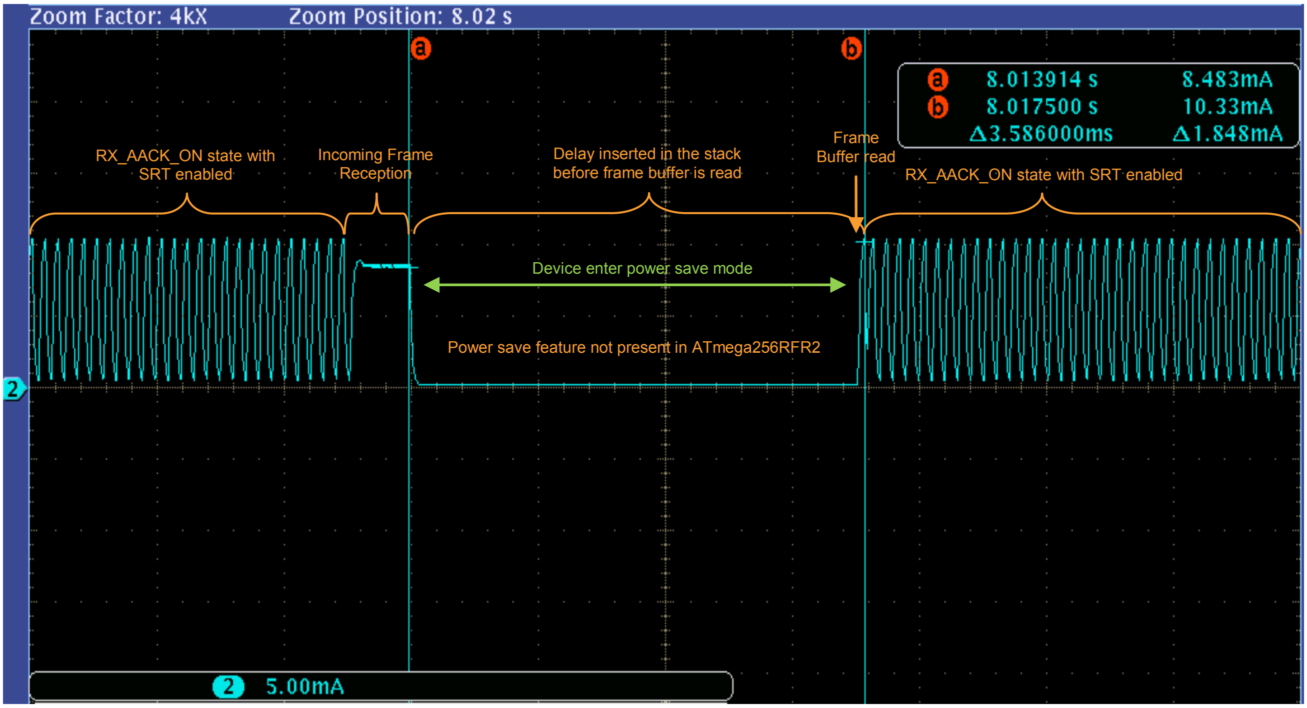
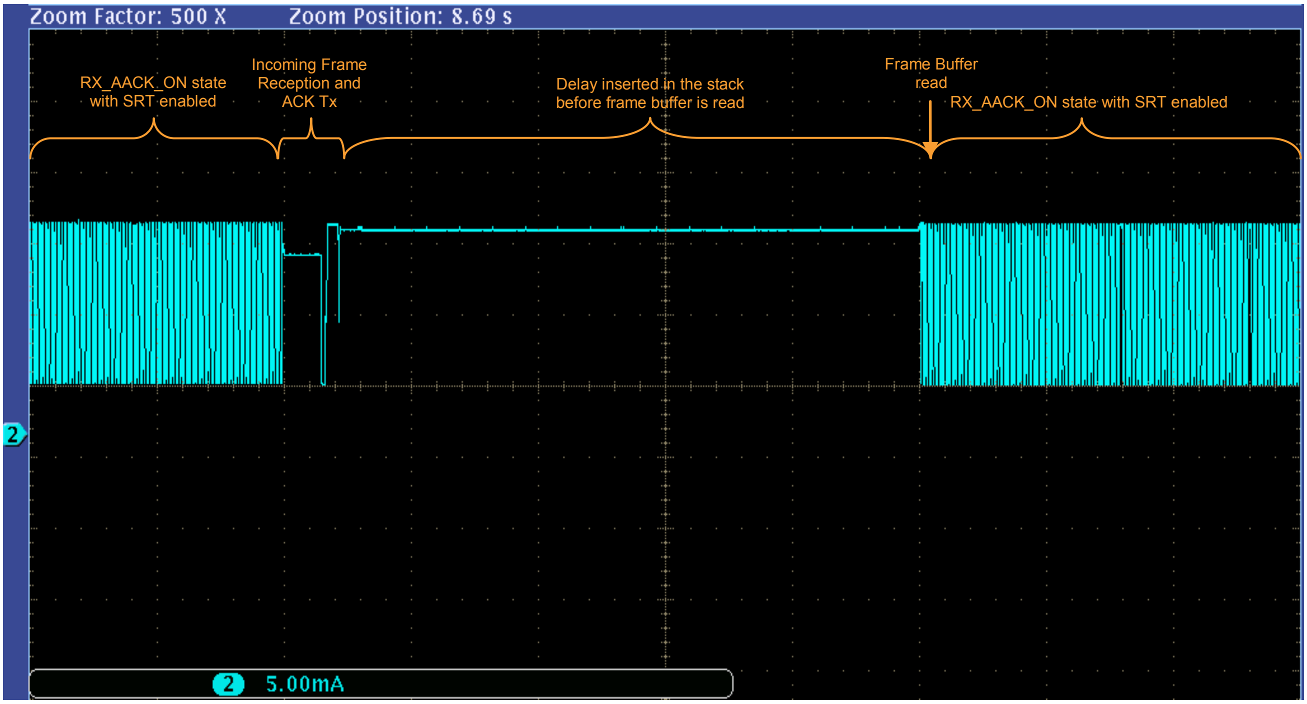


Figure 2-25 shows the power consumption of megaRFR2 [1] devices. Here, these devices does not support power save feature in during dynamic frame buffer protection period and the device does not go to power save mode until the frame buffer is read.

Figure 2-25. RPC On with Frame Buffer Read After a Delay in ATmega256RFR2



2.6.2 Random Back-Off Period Power Save

In TX_ARET (Transmit with automatic frame retransmission and CSMA-CA retry) mode, the transceiver will perform CSMA-CA algorithm as defined by IEEE 802.15.4-2006 which performs random back-off period. During this period, the transceiver automatically enters into power save mode.

2.6.2.1 Current Profile

Figure 2-26 shows the power save mode during random back-off period when frame transmission is initiated from TRX_OFF state.

Figure 2-26. Power Save in Random Back-Off Period

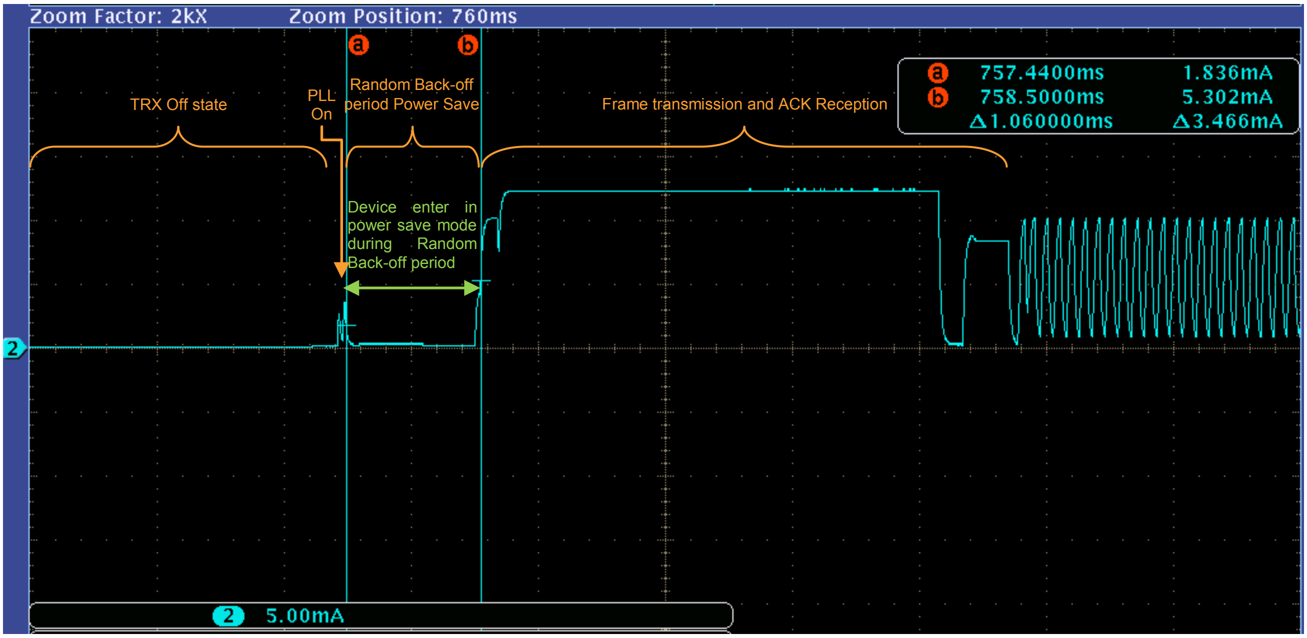
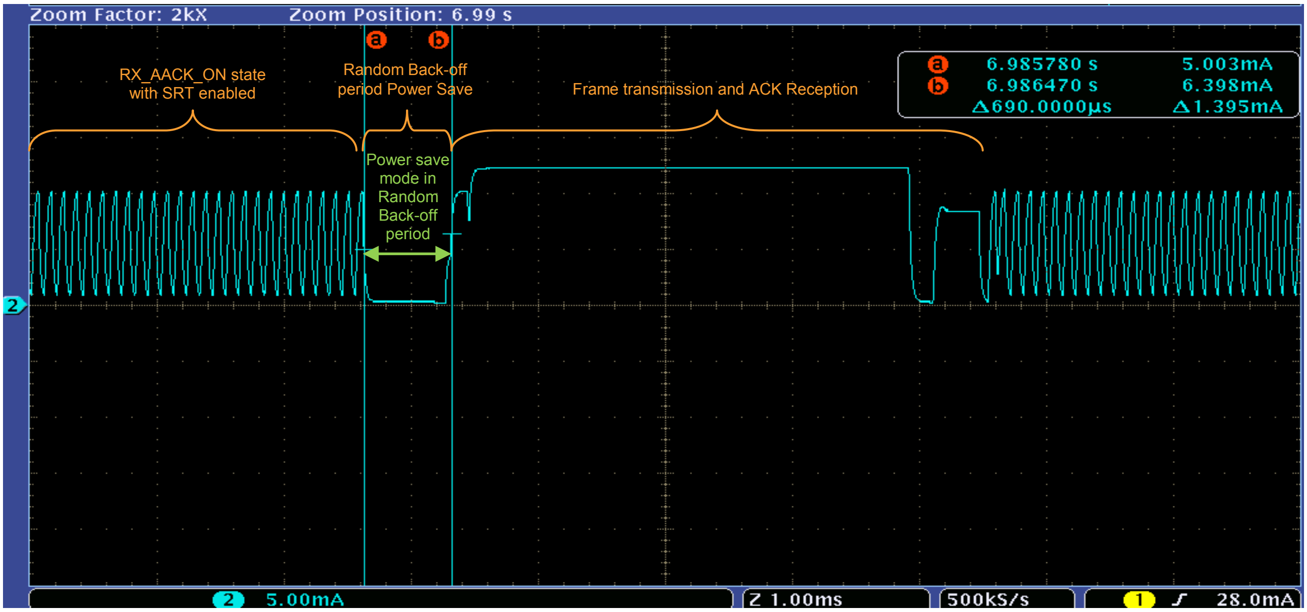


Figure 2-27 shows the power save mode during random back-off period when frame transmission is initiated from RX_ON/RX_AACK_ON state.

Figure 2-27. Power Save in Random Back-Off Period



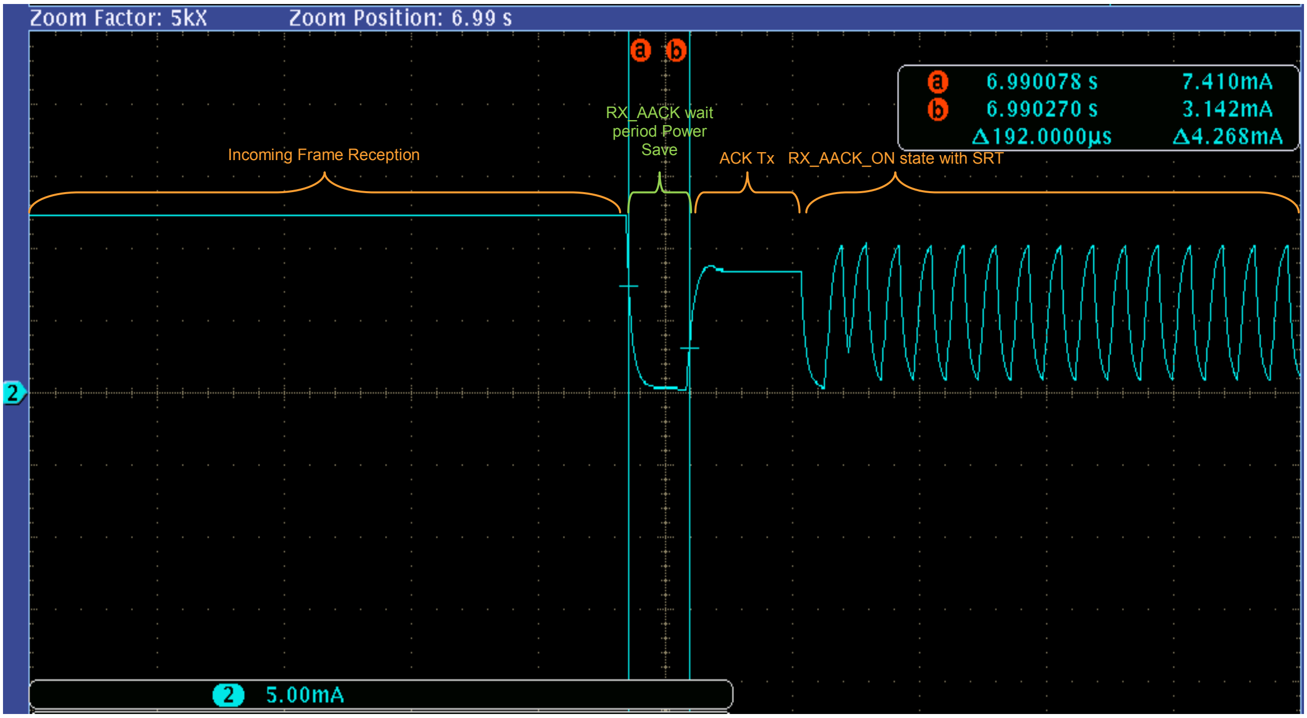
2.6.3 TX_ARET and RX_AACK Wait Time

As per IEEE 802.15.4-2006, transceiver should wait for a minimum turnaround time (12 symbol period) when changing from Tx-to-Rx or Rx-to-Tx. The Atmel transceivers will automatically transmit acknowledgement frame in `TX_ARET_ON` and `RX_AACK_ON` mode after 12 symbols period (192µSec for 2.4GHz) of the reception of last symbol of data/command frame. During this wait time, the transceiver automatically enters to power save mode.

2.6.3.1 Current Profile

Figure 2-28 shows the power saving during the turnaround time after reception of a frame to sending of acknowledge frame. As per IEEE 802.15.4-2006 specification the turnaround time is 12 symbols period and Figure 2-28 shows that the device enters in power save mode during the automatic acknowledge transmission in `RX_AACK_ON` mode.

Figure 2-28. Power Save in RX AACK Waits Period



3 Reference

[1] Atmel ATmega256RFR2 Datasheet

[2] Atmel AT86RF233 Datasheet

[3] Atmel AVR2162: REB233SMAD – Hardware User Manual

[4] Atmel ATmega256RFR2 Xplained Pro User Guide

[5] ASF – AVR2025 TAL (component) API - version 3.18.0

4 Revision History

Doc Rev.	Date	Comments
42356A	08/2014	Initial document release.