InvenSense

MoveaTV™ RF4CE Remote User Guide

INTRODUCTION

MoveaTV remote user guide is a reference document for developers to help them evaluating the use of our motion technology for interactive TV applications, and investigate the integration of motion into their product designs.

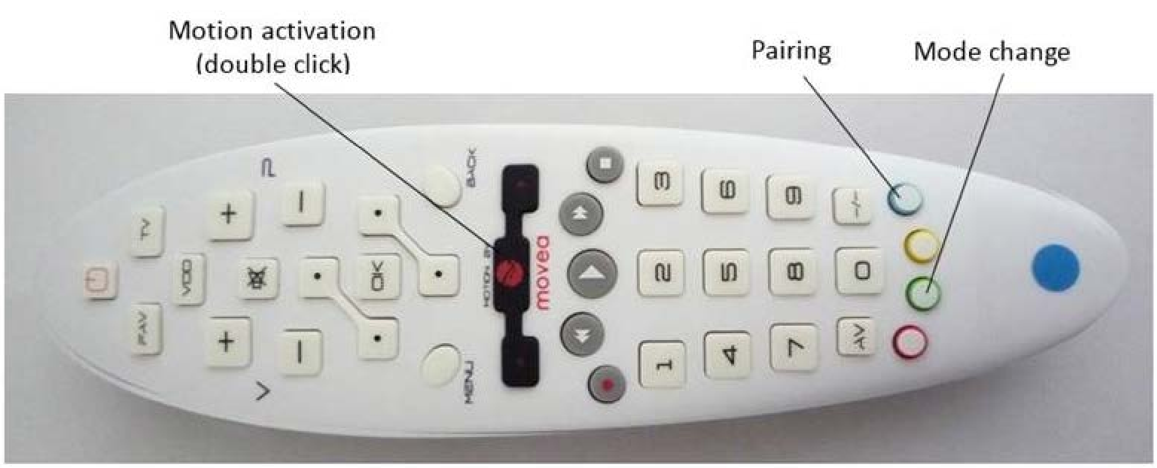
This document also explains the remote behavior.

REMOTE FUNCTIONALITIES

MoveaTV remote Reference Kit has been designed to show extended capabilities of motion technology.

Including:

- **Embedded or hosted accurate in-air pointing**
- **Embedded or hosted accurate in-air gaming**
- **Embedded or advanced hosted gesture recognition**
- **Static configuration depending on available sensors**
- **Real-time Gyroscope bias calibration**
- **Dynamic activation of roll compensation**
- **Dynamic activation of flip detection feature**
- **Dynamic configuration of gaming parameters**



InvenSense

TABLE OF CONTENTS

Introduction 1
Remote functionalities 1
TABLE OF CONTENTS 2
TABLE OF FIGURES 4
Features 5
Pointing 5
GAMING 5
Roll-Compensation 6
EasyClick 6
Calibration 6
Gesture recognition 7
Flip detection 7
IAR projects description 8
Database 8
IAR #define 9
Upgrade method 10
With IAR 10
Hex file with SmartRF Flash Programmer 11
OAD (over the air download) 12
Nano dongle Upgrade via Serial Bootloader 15
Special Key Sequence 16
Pointing test 17
Load firmware 17
Pair remote 17
Button press test 18
calibrate remote 18
Pointing test 19
Roll compensation test 20
Swipes test 20
Gaming test 21
Load firmware 21
Pair remote 21
Game controller test 22
MoveaTV Remote Tool 24
Annexes 25
Code size 25

TOOLS USED 25
FRAMES FORMAT 26
BATTERY LIFETIME APPROXIMATION 27
USB CONFIGURATION 28

InvenSense

TABLE OF FIGURES

FIGURE 1: SMARTRF FLASH PROGRAMMER WINDOW 11
FIGURE 2: TARGET EMULATOR WINDOW 13
FIGURE 3: OAD WINDOW 14
FIGURE 4: SERIAL BOOTLOADER..... 15
FIGURE 5: DONGLE 17
FIGURE 6: BLUE BUTTON TO PAIR..... 17
FIGURE 7: WINDOW SOUND 18
FIGURE 8: REMOTE MOVEA BUTTON..... 18
FIGURE 9: POINTING TEST..... 19
FIGURE 10: POINTING MOVEMENT..... 19
FIGURE 11: POINTING WITH ROLL COMPENSATION 20
FIGURE 12: REMOTE BUTTON SWIPE 20
FIGURE 13: DONGLE 21
FIGURE 15: GAMING TEST 22
FIGURE 16: HOW TO HOLD GAMEPAD..... 22
FIGURE 17: SWITCH JOYSTICK MAX ANGLE 23
FIGURE 18: MOVEATV REMOTE TOOL 24

FEATURES

POINTING

MoveaTV remote reference kit computes dx/dy mouse data, thanks to the Air Motion Library (embedded) or by sending raw data to the SmartMotion Server (hosted). It provides pixel-level accuracy, smooth displacement and tremor cancelation for best-in-class performance and ease of use.

The pointing computation can be performed from 2-axis Gyro, 2-axis Gyro + 3-axis Acc, or 3-axis Gyro + 3-axis Acc sensors data fusion.

The raw sensor feature is compatible with any sensors combination, up to 9-axis sensors (3-axis Gyro, Acc and Mag) data fusion.

Note: You can enable 'AirMotionLib' with **Green + 7** key sequences (see 6 for more information.)

GAMING

MoveaTV remote reference kit computes x/y joystick position, thanks to the Air Gaming Library (embedded) or by sending raw data to the SmartMotion Server (hosted). It provides high-accuracy, high-reactivity and smooth movement for best-in-class performance and ease of use.

The gaming can be performed with 3-axis Acc + Gyro sensors data fusion.

Note: You can enable 'AirGamingLib' with **Green + 8** key sequences (see 6 for more information).

ROLL-COMPENSATION

MoveaTV remote reference kit implements the roll compensation feature, which allows user's movement to be reliably reproduced on the screen, independently from the device roll rotation.

For a semi-dynamic roll-compensation, 3-axis Acc + 2-axis Gyro are necessary. For a full-dynamic roll-compensation, 3-axis Acc + 3-axis Gyro are necessary.

The computation is either made by the Air Motion Library (embedded) or by sending raw data to the SmartMotion Server (hosted).

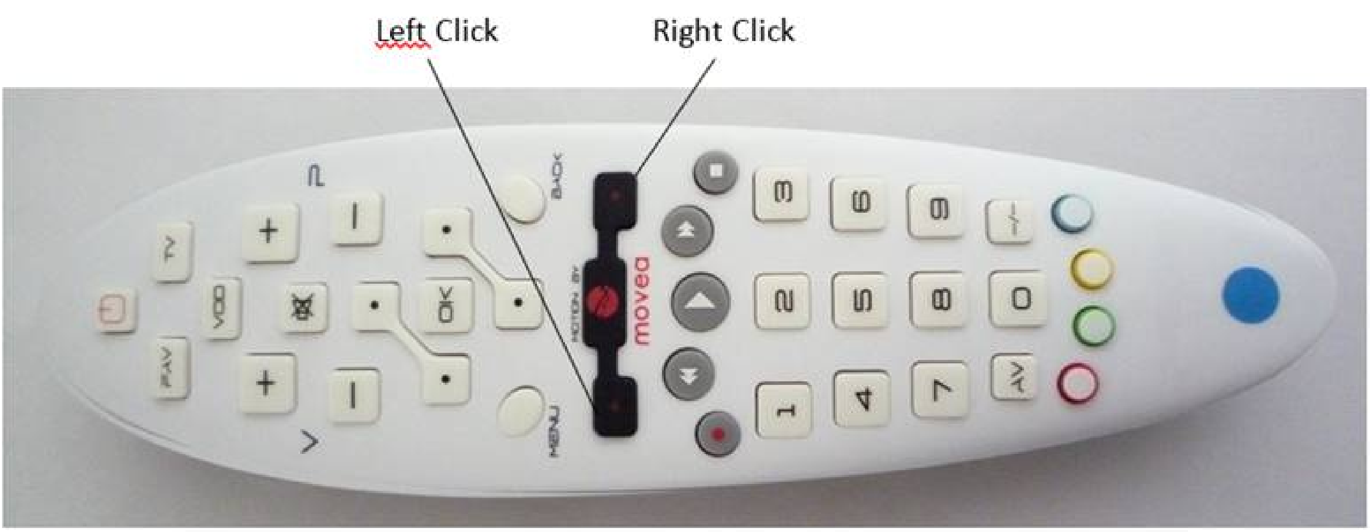
Note: You can switch on or off the roll compensation feature with **Green + 2** key sequence (see 6 for more information).

EASYCLICK

This feature allows user to perform more accurate and stable mouse clicks for a better experience.

On click press or release, the algorithm will freeze the pointer (i.e. force dX and dY values to zero) for a pre-defined amount of time to remove undesired movement (button pressing side-effect).

To avoid freezing pointer when not needed (e.g. while performing drag n' drop operations), device movement quantity is monitored: if it moves enough after a click, pointer will be released before specified timeout.



CALIBRATION

MoveaTV remote reference kit implements a calibration routine which computes gyroscope offsets values in real time.

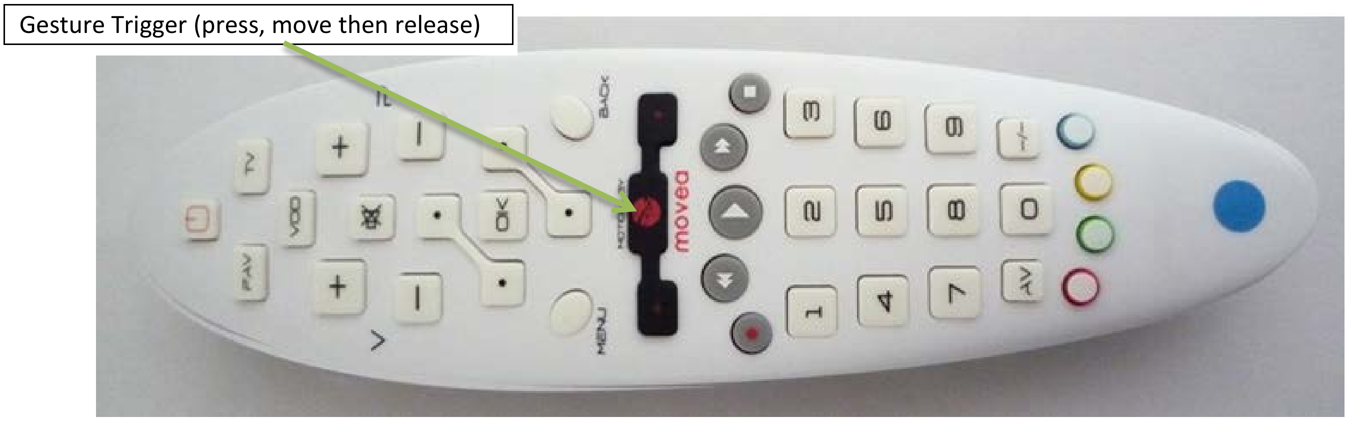
If the device is considered static for a certain amount of time (4 seconds), new gyroscope offsets values are computed and a buzzer will ring to signal the remote is calibrated.

The user has 15 seconds to calibrate the remote and if time is up he hears 3 short beeps. Time is reset if the Green + 1 key sequence is performed.

Note: The new offsets will be saved into the EEPROM only if the **Green + 1** key sequence is performed (see 6 for more information) and successful calibration.

GESTURE RECOGNITION

Gesture recognition is processed by the Air Motion Library (swipes), or by the SmartMotion Server (advanced gesture recognition).



FLIP DETECTION

The flip detection algorithm is used to detect a flip from a 3- axis accelerometer motion sensor. The library can detect if the device lays on its front side (buttons toward the sky) or back side.

When in pointing mode, if flip detection is enabled, and if remote lays on its back side, then remote stops sending mouse data until remote is flipped back.

When in raw sensor mode, if flip detection is enabled, and if remote lays on its back side, then remote sends special code inside frame “key press” field to indicate remote state, but it keeps on sending data unlike pointing mode. This is up to host application to interpret “key press” field and discard the frame or not whenever remote is flipped.

<u>Note:</u> The library can be turned on or off dynamically by pressing the **Green + 5** key sequence (see 6 for more information.)

IAR PROJECTS DESCRIPTION

DATABASE

All IAR projects are in *"\RemoTI 1.3"* directory. The projects are divided in several parts:

Components

Common for all projects, content HAL, OSAL and lib RF4CE

Projects\RemoTI\OadBoot

Remote bootloader

Projects\RemoTI\BasicRemote

Remote firmware with pointing, keyboard…

Binaries location:

"Projects\RemoTI\BasicRemote\CC2530RC\<PROJECT NAME>\Exe"

Project name description:

CC2533F96_* : hardware target (used by remote TI)
| _LIB_POINTING_V03 : with libPointing for 3A3G
| _LIB_AGL_V02 : with libGaming for 3A3G
| _LIB_POINTING_9AXIS_V03 : with libPointing for 3A3G based on 3A3G3M device (MPU9150)
| _LIB_POINTING_V03_AGL_V02 : with libPointing and libGaming for 3A3G

CC2533F128_* : hardware target (used by remote MOOS)
| _OAD_LIB_POINTING_V03 : with bootloader over the air and libPointing

Projects\RemoTI\ZidDongle

ZID dongle firmware

Project name description:

CC2531F256 : dongle firmware
CC2531F256_SB_NANO : nano dongle firmware

Projects\RemoTI\RNP

Serial dongle, use with over the air upgrade software. Don't work with remote pointing

IAR #DEFINE

POWER_SAVING: Accept µC low power mode

REMOTE_TI: use remote TI key matrix

REMOTE_MOOS: use remote Moos key matrix

USE_POINTING_LIB: use air motion lib
USE_POINTING_SENSOR: use raw sensor
USE_GAMING_AGL: use air gaming lib

USE_L3G4200D: use L3G4200D gyroscope (SPI only)
USE_IMU3000: use IMU3000 gyroscope (I2C only)
USE_LIS331: use LIS331 accelerometer (SPI only)
USE_KXTF9: use KXTF9 accelerometer (I2C only)
USE_MPU9150: use MPU9150 accelerometer and gyroscope (I2C only)
USE_MAGNETO: use MPU9150 magnetometer for 3A3G3M (I2C only)

USE_SENSOR_OVER_SAMPLING: oversampling sensor data reading, to work around a bug in L3G4200D which was updating data when reading multi bytes.
MOTION_GYRO_SLEEP: enable gyro sleep functionality, only with Moos remote.

USE_FLIP_DETECTION: use flip detection

USE_TRIGGER_FOR_POINTING: use trigger for pointing only in the use case “Pointing Off”, gesture is disabled

UPGRADE METHOD

WITH IAR

Dongle:

- Plug the probe
- Press Reset key on the probe. The LED should turn green.
- Load the firmware with IAR

Remote:

- Plug the probe
- Press Reset key on the probe. The LED should turn green.
- If the firmware needs a bootloader and if the bootloader was never loaded in target Load the bootloader with SmartRF Flash Programmer (see chapter 5.2) Firmware *"Projects\RemoTI\OadBoot\CC2530RC_Src\CC2530F128\Exe\ OadBoot.hex"*
- Load the firmware with IAR

HEX FILE WITH SMARTRF FLASH PROGRAMMER

<u>Remote / Dongle:</u>

- Start SmartRF Flash Programmer (TI software)
- Plug the probe
- Select « Program CCxxxx SoC or MSP430 » (default option)
- In Flash image, select your hex file
- Perform actions

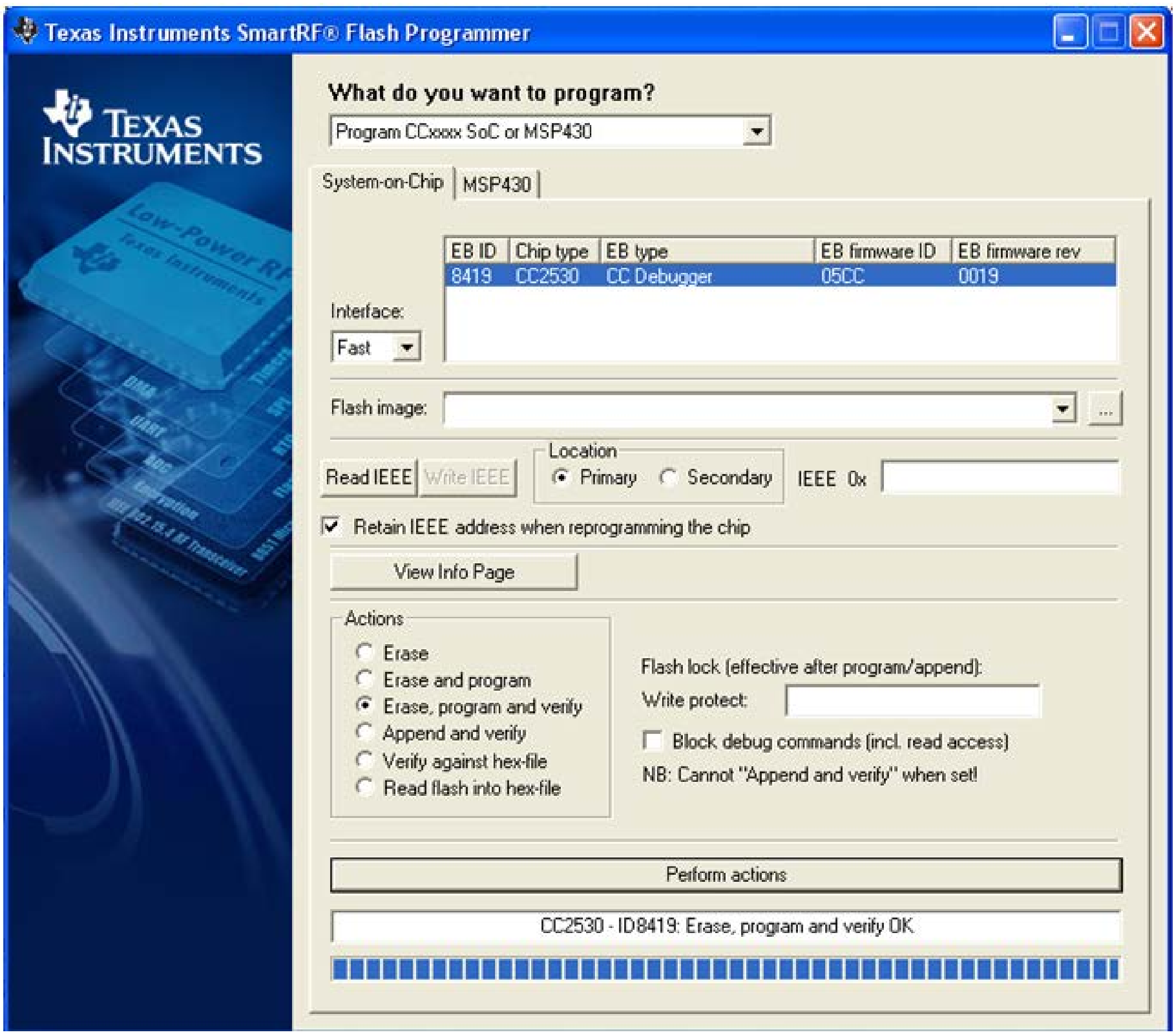


Figure 1: SmartRF Flash Programmer Window

OAD (OVER THE AIR DOWNLOAD)

Remote:

- Plug a dongle with RNP firmware
- Launch TargetEmulator.exe (TI Software)
- Menu Target->Start Target Emulator select Serial Port (remember COM port used)
- Pair remote with serial dongle via TargetEmulator and remote blue button
- Close TargetEmulator
- Launch RemoTI_OadDemo.exe (TI Software)
- Enter COM port number and click on « Connect »
- Select bin file firmware and your serial dongle
- Click on « Download Image »
- Press remote yellow button to accept upgrade

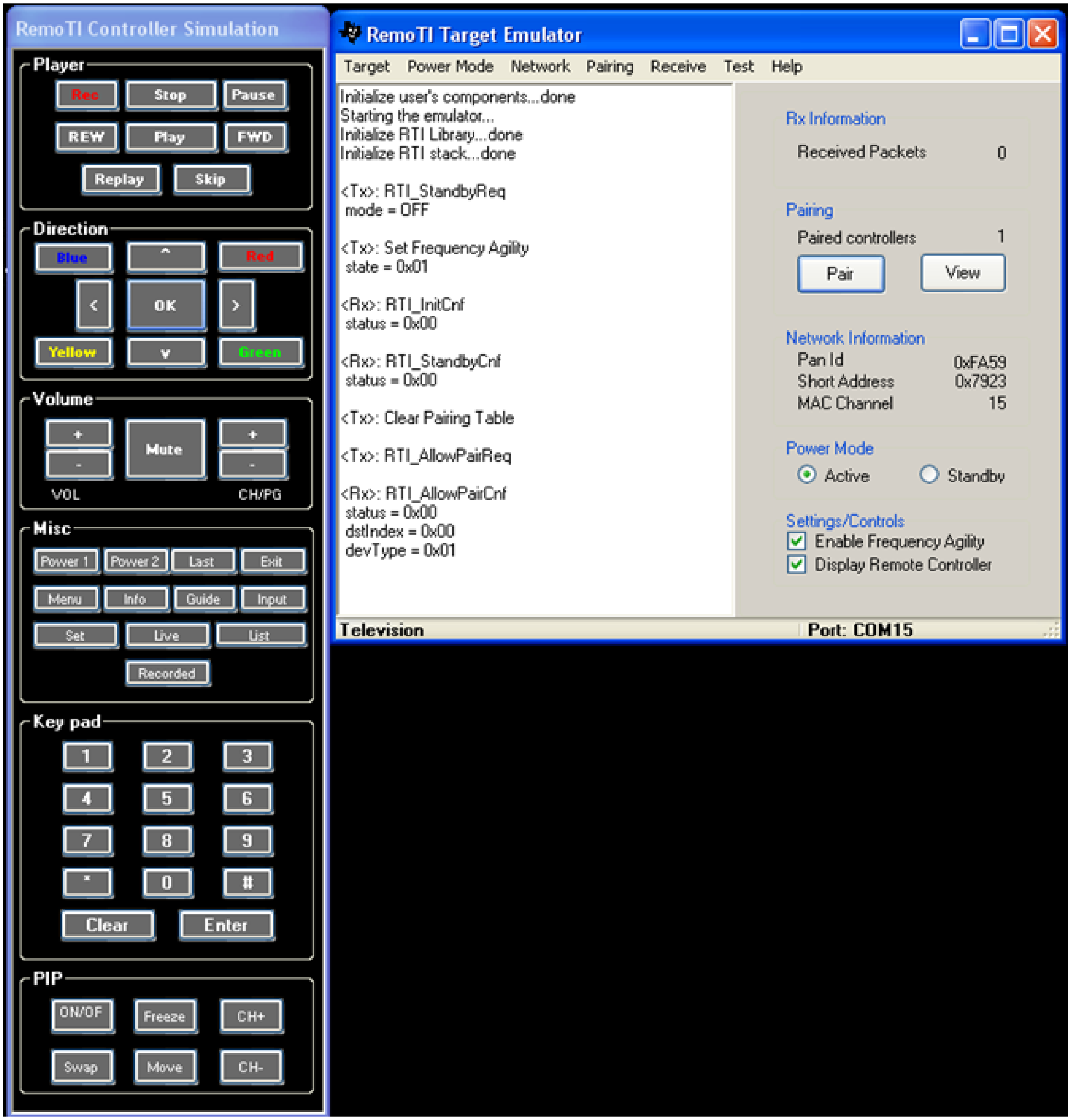


Figure 2: Target Emulator Window

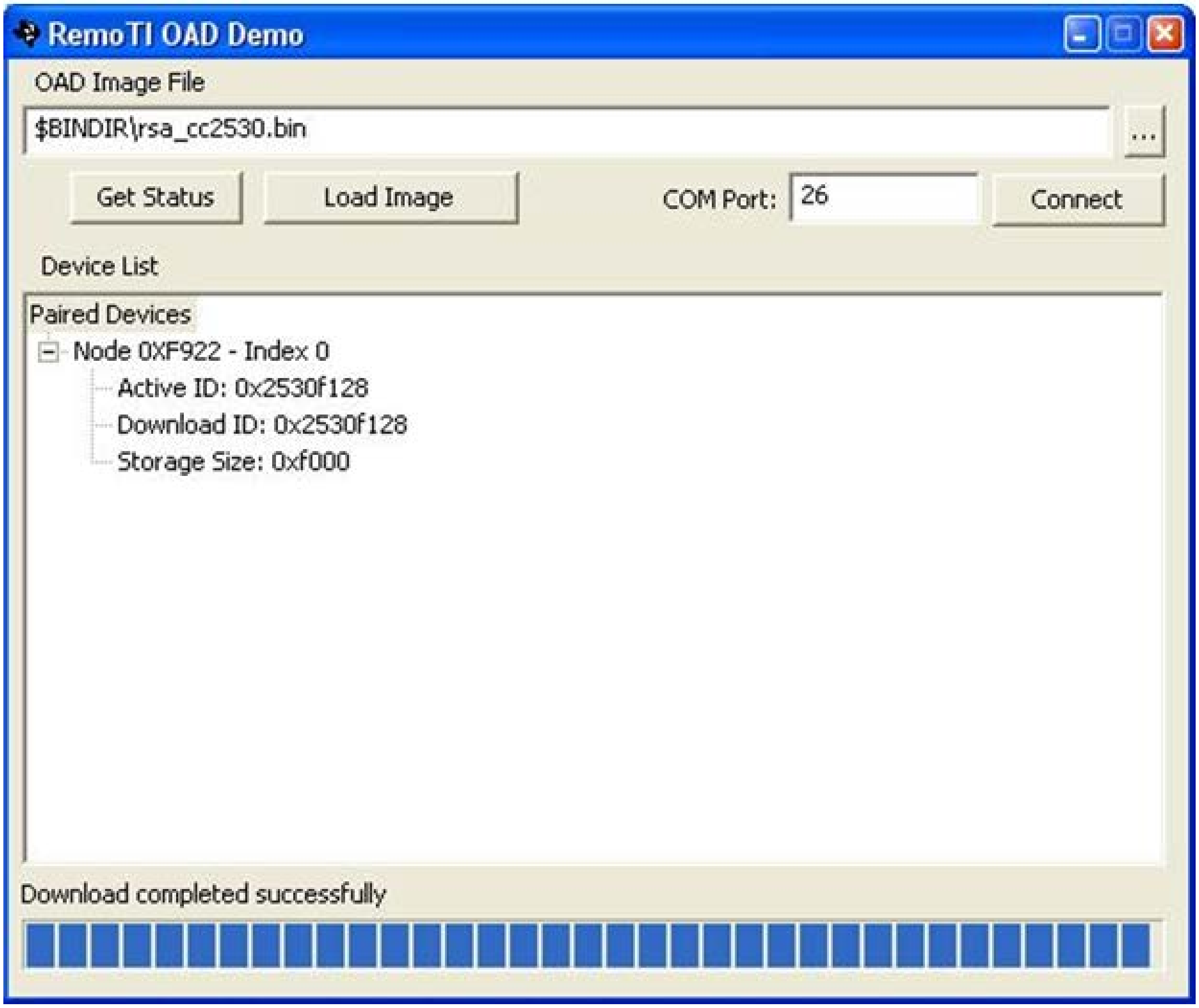


Figure 3: OAD Window

NANO DONGLE UPGRADE VIA SERIAL BOOTLOADER

Nano dongle have a USB bootloader code programmed into the device for loading application code onto the device.

- Start RemoTI_SBDemo.exe
- Select the the port us by nano and the nano dongle firmware
- Plug nano dongle and click on load Image

For more information on Texas Instruments wiki: http://processors.wiki.ti.com/index.php/Nano-USB

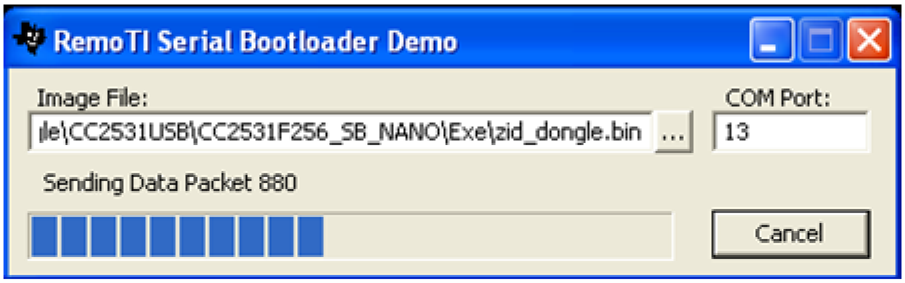


Figure 4: Serial Bootloader

SPECIAL KEY SEQUENCE

Green + 1: Sensor calibration (libPointing), stored in Flash.
Green + 2: Enable/disable Roll-Compensation (libPointing). Default is ‘Enabled’.
Green + 3: Change from AirMotionLib to AirGamingLib to RAW sensor (choice saved in Flash).
Green + 4: Enable AirMotionLib extra info frame (see 10.3), disable pointing.
Green + 5: Enable/disable FlipDetectionLib. Default is ‘Disabled’.
Green + 6: Switch AirGamingLib joystick max angle between $^{+}/_{-}$ 45° and 90°. Default is ‘45°’.
Green + 7: Enable AirMotionLib mode (choice saved in Flash).
Green + 8: Enable AirGamingLib mode (choice saved in Flash).
Green + 9: Enable RAW data mode (choice saved in Flash).

Green + Forward + Backward + Yellow:	Reset Default Setting (calibration offsets, roll compensation, flip detection, angle AGL)
Rec + Back + Menu + Stop:	The time of “Sleep Mode” increase of 5 minutes after this sequence performed. Time possibility : 5/10 minutes
Stop + Menu + Back + Rec:	Reset the time of “Sleep Mode”. Default is 15 seconds

POINTING TEST

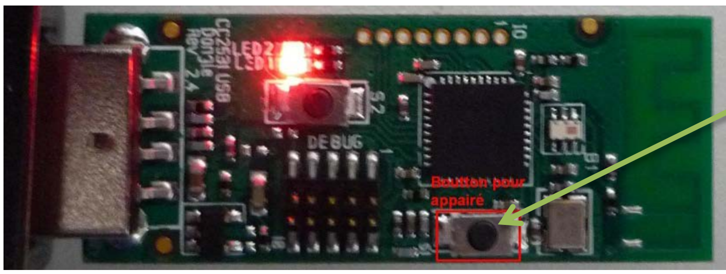
LOAD FIRMWARE

See chapter 5 for details.

PAIR REMOTE

Press right button dongle, green LED start blinking

Press remote blue button. If dongle green LED stop blinking and stay green, the remote is paired.



PRESS BUTTON TO PAIR THE REMOTE

Figure 5: Dongle

PRESS BLUE BUTTON TO PAIR THE REMOTE

Figure 6: Blue button to pair

BUTTON PRESS TEST

Click “Start->configuration Panel->sound and audio device”

Press remote mute button and check on screen if mute is check

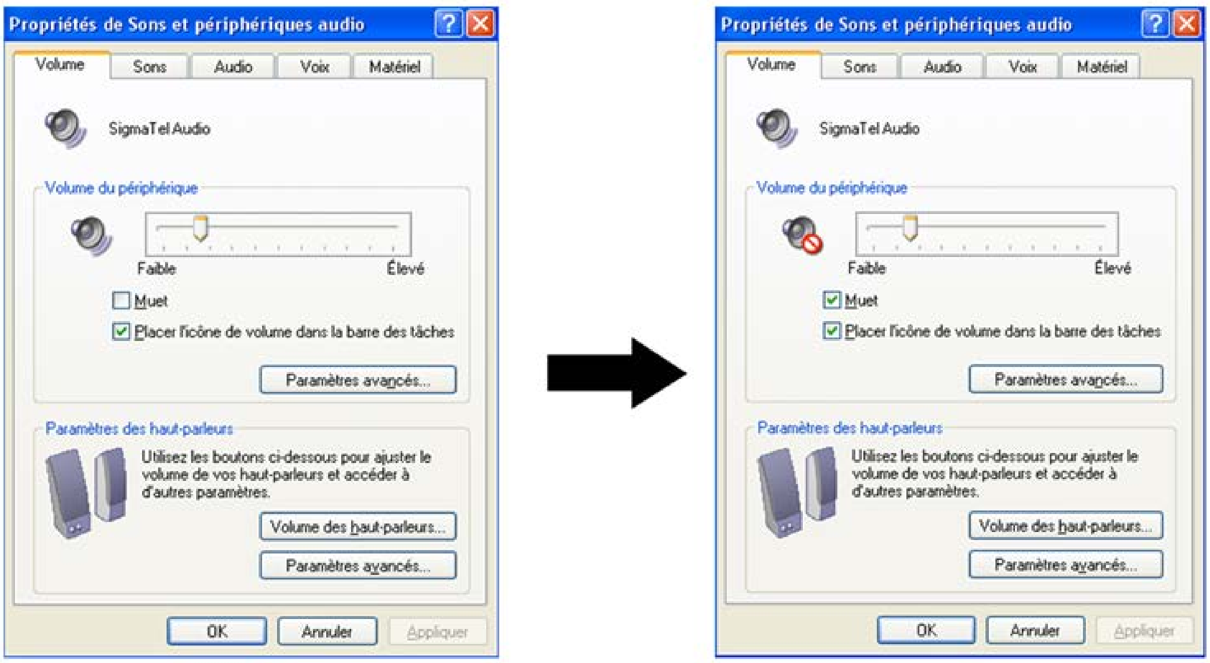


Figure 7: Window sound

CALIBRATE REMOTE

Double click on Movea button to enable the pointing or data transfer.
Put the remote flat on a table. The remote should be still to update the gyroscope offsets.
The calibration should not take more than 5 seconds. You can press **Green + 1** to save the offsets.

Figure 8: Remote Movea Button

POINTING TEST

Double click on Movea button to enable pointing and **Green + 7** if necessary to enable Mouse mode.
Make square with your mouse pointer. Check if a horizontal displacement corresponds to dX pointer movement and if a vertical displacement corresponds to dY pointer movement.

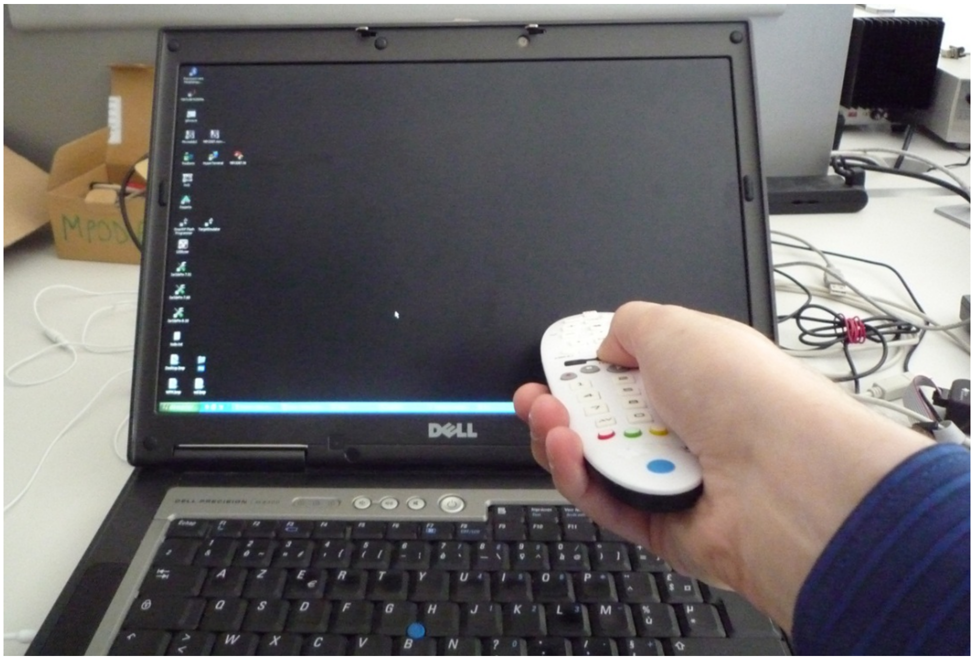


Figure 9: Pointing test

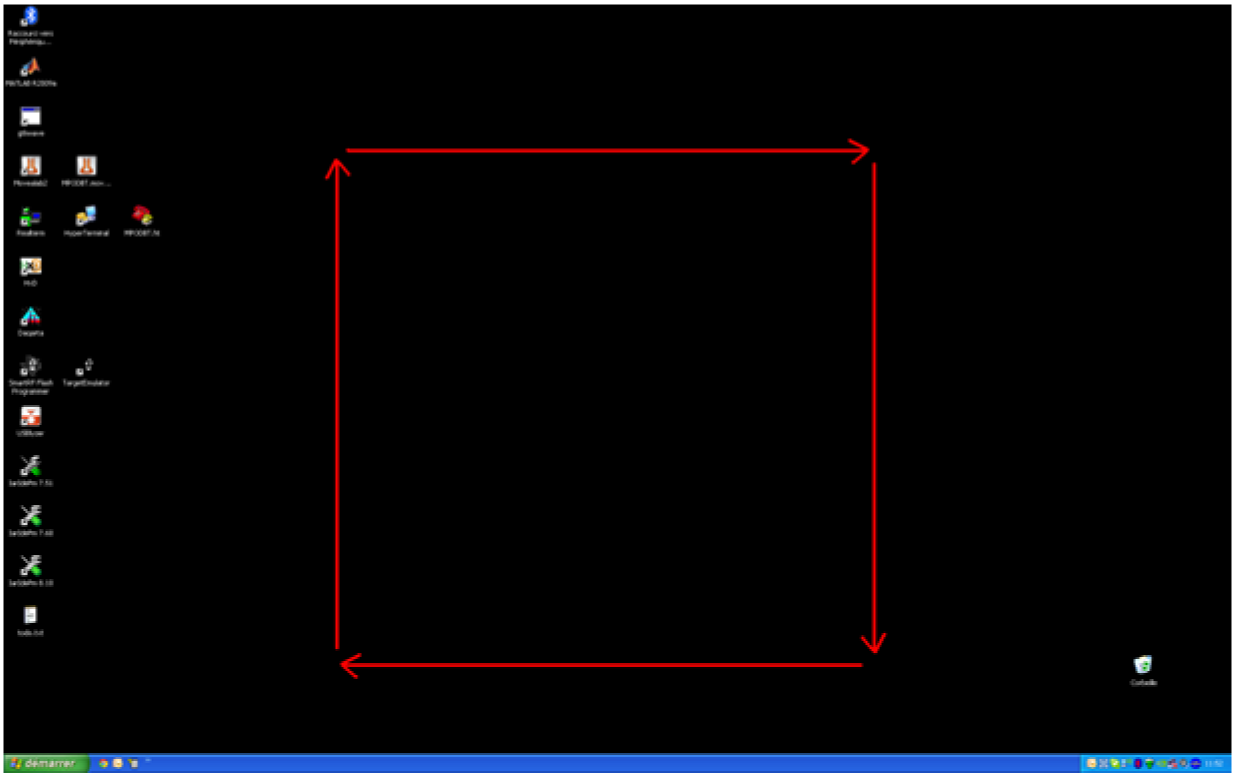


Figure 10: Pointing movement

When performing a square, the cursor should go back to its initial position.

ROLL COMPENSATION TEST

Tilt the remote to be 90 ° axis. And repeat the last step.

Press **Green + 2** to enable/disable roll compensation. Default is ‘Enabled

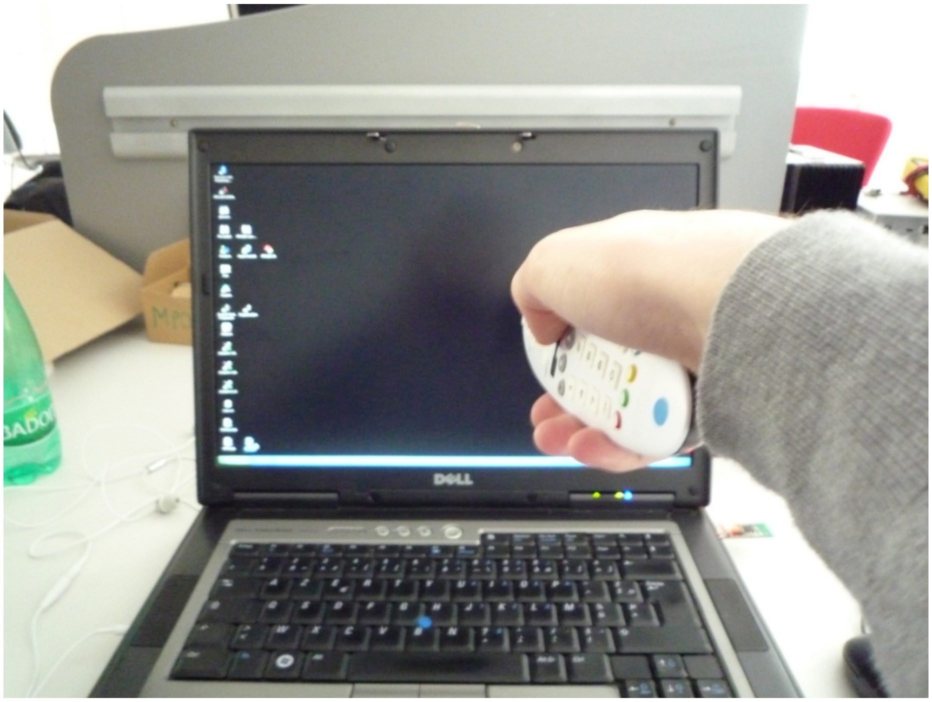


Figure 11: Pointing with roll compensation

SWIPES TEST

Swipes gestures are dedicated to web browser. Open Windows Xhrome web browser for example.

Figure 12: Remote Button swipe

To perform a swipe: Press the Movea button, move the remote to describe one of the following gestures, and then release the Movea button. (You can test them in a web browser for example).

Gesture	Action
Left	Previous page
Right	Next page
Up	Page up
Down	Page down
Roll clockwise	Zoom +
Roll counter clockwise	Zoom -

GAMING TEST

LOAD FIRMWARE

See chapter 5 for details.

PAIR REMOTE

Press right button dongle, green LED start blinking.
Press remote blue button. If dongle green LED stop blinking and stay green, the remote is paired.

PRESS BUTTON TO PAIR THE REMOTE

Figure 13: Dongle

PRESS BLUE BUTTON TO PAIR THE REMOTE

GAME CONTROLLER TEST

Double click on Movea button to enable gaming, and **Green + 8** if necessary to enable Joystick mode.
Run Game Controller/Properties/Test Windows application.

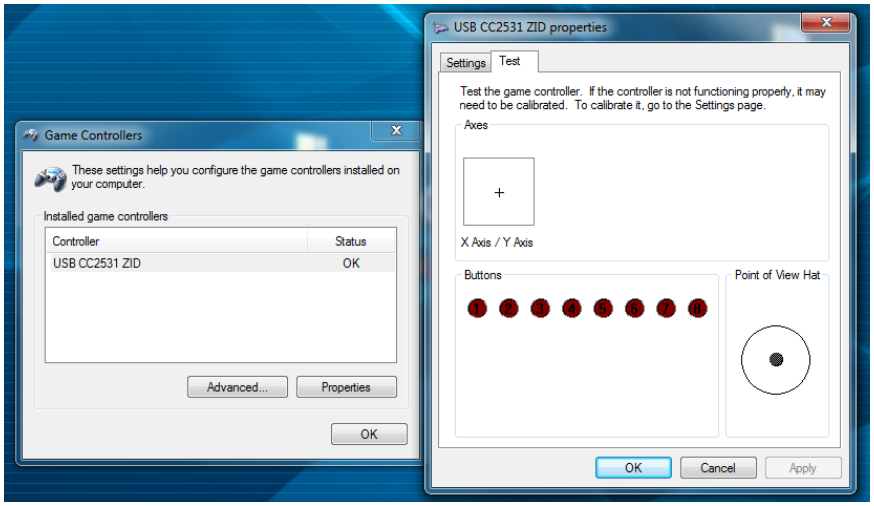


Figure 14: Gaming test

Take and roll remote like a joystick to check X and Y axis gamepad behavior.

Figure 15: How to hold gamepad

Press **Green + 6** to switch maximum angle detected for joystick between 45 and 90°.
Default is 45°. Repeat rolling operation to check the different sensitivities.

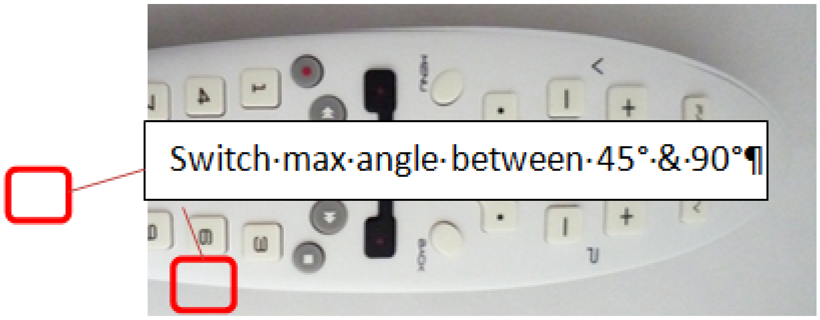


Figure 16: Switch joystick max angle

MOVEATV REMOTE TOOL

MoveaTV Remote Tool is a diagnosis tool which is able to monitor raw data sent by the remote.

Open ***MoveaTV Remote Tool v07***.
Click on "Connect remote"

<u>Notes:</u>

- This tool only works with raw data. Please switch to raw data mode with **Green + 9** key sequence if required.
- You can monitor the trigger state.
  The Flip state is monitored only if the flip detection library is enabled (**Green + 5** key sequence)
- MoveaTV Remote Tool detect if you use 6 Axis or 9 Axis remote.

Figure 17: MoveaTV Remote Tool

ANNEXES

Note: Information in this section is for reference only. Figures may be out of date or inaccurate.

CODE SIZE

CC2530F128		With OAD	
_OAD_LIB_POINTING_V03	4.0.0	Flash	RAM
		59 867	4590
CC2533F96		Without OAD	
_LIB_POINTING_V03	4.0.0	Flash	RAM
		67710	4185
_LIB_POINTING_9AXIS_V03	4.0.0	Flash	RAM
		79574	4197
_LIB_AGL_V02	4.0.0	Flash	RAM
		66912	3955
_LIB_POINTING_V03_AGL_V02	4.0.0	Flash	RAM
		79084	4415
_LIB_POINTING_9AXIS_V03_AGL_V02	4.0.0	Flash	RAM
		91827	4427
RNP		Flash	RAM
		61096	5815
OADBoot		Flash	RAM
		1869	2 400
ZID Dongle		Flash	RAM
		73667	5921

*in Bytes

TOOLS USED

- IAR version 8.10
- SmartRF Flash Programmer
- TargetEmulator.exe (TI Software) (for OTA upgrade)
- RemoTI_OadDemo.exe (TI Software) (for OTA upgrade)

FRAMES FORMAT

RF4CE frame format:

Octets: 1	4	0/1	0/2	Variable	0/4
Frame control	Frame counter	Profile identifier	Vendor identifier	Frame payload	Message integrity code
Header				Payload	Footer

- Frame control: control information for the frame
- Frame counter: incrementing counter to detect duplicates and prevent replay attacks (security)
- Profile identifier: the application frame format being transported
- Vendor identifier: to allow vendor extensions
- Frame payload: contains the application frame
- Message integrity code: to provide authentication (security)

Source: https://docs.zigbee.org/zigbee-docs/dcn/09-5231.PDF

Payload frame format for raw sensor 6 axis (Movea specific):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ID = 1	Frame counter	Ax LSB	Ax MSB	Ay LSB	Ay MSB	Az LSB	Az MSB	Gx LSB	Gx MSB	Gy LSB	Gy MSB	Gz LSB	Gz MSB	Key Press

Payload frame format for raw sensor 9 axis (Movea specific):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ID = 2	Frame counter	Ax LSB	Ax MSB	Ay LSB	Ay MSB	Az LSB	Az MSB	Gx LSB	Gx MSB	Gy LSB	Gy MSB	Gz LSB	Gz MSB	Mx LSB	Mx MSB	My LSB	My MSB	Mz LSB	Mz MSB	Key Press

Payload frame format for air motion lib extra info (Movea specific)

0	1	2	3	4-5	6-7	8-9	10	11-13	14
ID = 1	Frame counter	dX	dY	AML version	Algo Duration in µs	µC Sleep Duration in µs	Air Motion Status	Remote FW reference kit version	Key Press

Payload frame format for air gaming lib extra info (Movea specific)

0	1	2	3	4-5	6-7	8-9	10	11-13	14
ID = 1	Frame counter	x	y	AGL version	Algo Duration in µs	µC Sleep Duration in µs	Air Gaming Status	Remote FW reference kit version	Key Press

BATTERY LIFETIME APPROXIMATION

The usage model we employed is based upon available customer usage data, with the average consumer watching TV about 8.25 hours per day. The average consumer is assumed to be navigating menus such as the program guide 1.8 times per hour and will watch more half-hour shows but occasionally watching longer programs such as movies or sports throughout their viewing day. The number of commands transmitted during menu navigation varies based on whether the remote is a button-only remote or an advanced remote with mouse-like navigation.

Our usage model also assumes a certain number of button presses for other functions. For example, the model assumes 10 button presses per hour related to TV volume, including mute. Additionally, a remote associated with a set-top box that has a digital video recorder (DVR) function will experience more usage as the user will take advantage of various features such as pause and skip forward.

ontrol will affect battery life. The assumption is that the remote uses AAA alkaline cells wired serially and that the remote ICs can consume 1200 milliamp hour (mA) from the batteries before reaching their lower voltage limit and ceasing to function.

The remote control's processor is assumed to draw 7.5mA when active and 5 microamperes (µA) when idle. When sending ZigBee Remote Control commands, the remote's transceiver is assumed to draw 27mA and 28mA in receive mode when waiting for acknowledgements. This model assumes all ZigBee messages are acknowledged. In idle/sleep mode the transceiver is assumed to draw 2µA. These currents are typical for the parts that are available today for use in remote controls.

For IR transmission, there is a wide range of current draws as various remote manufacturers try to balance IR communication robustness with battery draw. Additionally, the duty cycles for various IR codes and commands vary. We assumed that the IR current draw is 250mA peak with new batteries and with a duty cycle that is 15 percent on, 85 percent off. These values are in the typical range for IR remote controls. Although the average IR current appears to be similar to RF, there is a significant difference in the total time involved. With ZigBee Remote Control, the transceiver will typically be on for ~1millisecond (msec). Allowing for occasional retries of unacknowledged commands, the average power on time for a ZigBee Remote Control transceiver is <<10msec. In the case of a typical IR command, the IR transmission will take >100msec.

	Non-DVR Remote	DVR Remote
IR Only	596	381
ZigBee - STB, IR - TV	904	688
ZigBee Only	1044	766

Table 2 - Calculated Battery Life for a Remote With a Pointing Device in Days

	Non-DVR Remote	DVR Remote
IR Only	200	168
ZigBee - STB, IR - TV	348	310
ZRC Only	367	325

Table 3 - Calculated Battery Life for a Remote Also Used For Internet Activity in Days

Source: https://docs.zigbee.org/zigbee-docs/dcn/11-0009.pdf

Warning: these numbers don't take into account the gyroscope consumption, for example.

USB CONFIGURATION

Dongle is configured as an USB2.0 device of HID class.
There are 5 interfaces supported :

- **Endpoint0** used for control transfer. The size of the packet in Endpoint 0 is 32 bytes.

- **Endpoint1** used for generic **keyboard** configured as IN interrupt with keyboard protocol.

- **Endpoint2** used for **consumer electronics** control configured as IN interrupt.

- **Endpoint3** used for **raw** sensor data transfer configured as IN interrupt. The size of the packet is :

    - either 15 bytes for 6-axis sensor as defined in Payload frame format for raw sensor 6 axis and can be identified with report ID 0x1
    - or 21 bytes for 9-axis sensor as defined in Payload frame format for raw sensor 9 axis and can be identified with report ID 0x2
    - or 15 bytes for air motion lib extra information as defined in Payload frame format for air motion lib extra info, which is activated through **Green + 4** and can be identified with report ID 0x1

- **Endpoint4** used for **mouse** configured as IN interrupt with mouse protocol.

- **Endpoint5** used for **joystick** configured as IN interrupt with gamepad control including hat switch and 7 buttons.

This information furnished by InvenSense is believed to be accurate and reliable. However, no responsibility is assumed by InvenSense for its use, or for any infringements of patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. InvenSense reserves the right to make changes to this product, including its circuits and software, in order to improve its design and/or performance, without prior notice. InvenSense makes no warranties, neither expressed nor implied, regarding the information and specifications contained in this document. InvenSense assumes no responsibility for any claims or damages arising from information contained in this document, or from the use of products and services detailed therein. This includes, but is not limited to, claims or damages based on the infringement of patents, copyrights, mask work and/or other intellectual property rights.

Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by implication or otherwise under any patent or patent rights of InvenSense. This publication supersedes and replaces all information previously supplied. Trademarks that are registered trademarks are the property of their respective companies. InvenSense sensors should not be used or sold in the development, storage, production or utilization of any conventional or mass-destructive weapons or for any other weapons or life threatening applications, as well as in any other life critical applications such as medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, disaster prevention and crime prevention equipment.