

# NucleoPlay

The NucleoPlay is a simple touchscreen MP3 player. It plays a playlist stored in memory. It supports play/pause and stop functionality. It also has the ability to skip to the next or previous song in the playlist.

## Features

### ARM Cortex M4 Processor

At the heart of the NucleoPlay is a ARM Cortex M4 processor. It is hosted on the STM32F401RE MCU and interfaced via the Nucleo Development Board.

### MP3 Audio

MP3 audio is decoded via a VS1053 chip hosted on the Adafruit “Music Maker” MP3 Shield for Arduino. it communicates with the STM32 MCU via a SPI interface with a clock speed of 100 MHz.

### Reactive Touchscreen Display

The NucleoPlay interface is brought to life using the 2.8” TFT Capacitive Touch Shield. Display information is sent to the shield via a SPI interface with a clock speed of 100 MHz, while capacitive touch data is transferred via an I2C interface operating at 100 KHz.

## Operation

### Overview

The NucleoPlay makes use of uCOS-II, a real-time operating system (RTOS). Multiple tasks running in concurrency allow for the responsive touchscreen and audio functionality. For the purpose of this specification, the tasks are named as follows:

- MP3 Task

- Display Task
- Touch Task
- Command Task

Each task has a in depth description found in the NucleoPlay DOCs.

## Start Up

Upon powering the NucleoPlay, the RTOS will initialize all channels of communication between the operational tasks. This includes:

- Touch-Command Queue
- Command-MP3 Queue
- MP3-Display Mailbox

The RTOS will then initialize and start all operational tasks. Once all tasks have been started, the Display Task will present the user interface, the MP3 Task will be pending for a command to begin playback, the Command Task will pend for a touch notification, and the Touch Task will be polling the touchscreen continuously for user interaction.

## Task Interactions

A functional diagram of task interactions can be found in the NucleoPlay DOCs.

The general flow of task communication begins with the Touch Task continuously polling the touch driver to see if a touchscreen event has occurred (i.e. the play button has been pressed). If a touch was detected, the location of the touch point (received via I2C) is checked to determine if a button has been pressed. If so, a command is issued to the Command Task via the Touch-Command Queue.

The Command Task resumes from its pended state after receiving a message in the Touch-Command Queue. Based on the given command, the Command Task will then send one or more messages to the MP3 Task via the Command-MP3 Queue.

The MP3 player periodically checks the Command-MP3 queue to see if a command has been issued. If so, the MP3 Task then updates its state based on the received command. A change in state may or may not result in a message being sent from the MP3 Task to the Display Task via the MP3-Display Mailbox. If the MP3 Task is in the playback state, a chunk of MP3 data is read from ROM and sent to the MP3 decoder chip via the Nucleo Board's SPI interface to be played out of the speaker.

The Display Task spends most of its time in the pended state. Upon receiving a message in the MP3-Display mailbox, a quick update to the LCD screen is made via the SPI interface. The update to the display may or may not require reading MP3 metadata from ROM. The Display Task then re-entering the pended state and waits for another update.