

## REC&PLAY CLICK

PID: MIKROE-3345 Weight: 30 g

**Rec&Play Click** is a digital voice recorder on a Click board™. It is based on the ISD3900, a multi-message record and playback device. It features the ChipCorder® technology, offering digital sound compression, smart message management, digitally configurable signal path, and more. Several audio inputs and outputs are available to the user. A small omnidirectional electret mic, as well as a miniature 700mW neodymium speaker are included on the Click board™, making it a complete playback & recording solution, capable of capturing up to 32 minutes of audio. It also offers numerous connectivity options.

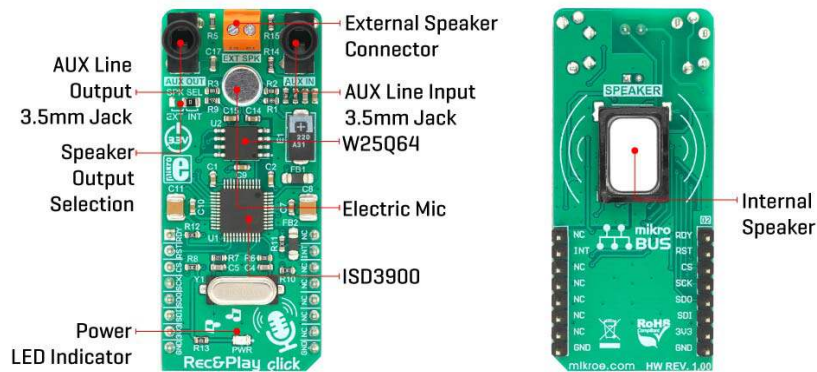
Rec&Play click is supported by a mikroSDK compliant library, which includes functions that simplify software development. This Click board™ comes as a fully tested product, ready to be used on a system equipped with the mikroBUS™ socket.

Offering the complete message playback and recording platform that offloads heavy compression calculations from the host MCU, Rec&Play click allows rapid development

of a wide range of sound-based applications, greatly cutting the time to market. It is an ideal solution for POS terminals, toys, home appliances, digital voice recorders, and all other applications that require a simple interface for sound notifications.

## HOW DOES IT WORK?

The main component of the Rec&Play click is the [ISD3900](#), a multi-message record and playback device with ChipCorder® technology, from [Nuvoton](#). Thanks to the many useful features it offers, this IC gained its popularity very quickly. Besides several compression and de-compression CODEC, it features a fully programmable signal path with the support for several different signal types, allowing a very simple design for any kind of message-based application. Sound notification from a POS terminal, a home appliance that uses voice to notify about some event, even a Dictaphone-like device, it can all be implemented very quickly with this Click board™. Instead of using memory locations, ISD3900 allows using message indices: once stored, a message can be invoked using its index. Building of macro scripts (Voice Macros) is also possible.



To allow the out-of-box functionality, the Click board™ is equipped with a small omnidirectional electret microphone, along with a miniature 8Ω speaker.

The microphone is can capture a sound or voice recording. The recording can be reinforced by applying the Automatic Gain Control (AGC), which dynamically changes the input gain, helping to obtain the most appropriate signal level. The AGC function, along with many other functions, can be configured over the SPI interface. The miniature neodymium speaker (internal speaker) measures only 15 x 11mm, and it is designed to output up to 0.7W of continuous power. It is driven by the ISD3900 IC, which integrates a small 0.5W class D amplifier. The internal speaker has an adhesive on its rims, allowing it to be easily mounted inside a small resonant box, which can reinforce its output. However, this speaker is good only when used indoors.

2-pole terminal labeled as EXT SPK allows an external passive speaker box to be connected. When the SPK SEL is moved to the EXT position, the output of the internal class D amplifier becomes routed to this connector, instead of the internal speaker. A

differential BTL (Bridge-Tied Load) output is also available over the EXT SPK connector, allowing an external amplifier with differential inputs to be used.

AUX IN and AUX OUT are 3.5mm jack connectors, used to connect line-level input and output signals. The AUX OUT connector offers a single-sided line-level output, compatible with most of the commercial amplifiers, active desktop speakers, sound monitors, etc. The AUX IN is a single-sided line-level input, allowing to capture sound from a home PC, TV set, MP3 player, and any other device equipped with the line-out connector.

The audio sample rate is derived from an external crystal oscillator, rated at 4.096MHz. This crystal oscillator allows the master sample rate up to 32kHz. Lower sample rates can also be used, resulting in reduced sound quality. Besides the sample rate, the sound quality is also affected by the used compression algorithm. ISD3900 supports many different CODECs. For more information, please refer to the ISD3900 IC datasheet.

Besides the ISD3900 this Click board™ contains another IC: it is a non-volatile memory module labeled as W25Q64, a 64 Mbit serial Flash memory from Winbond. This memory module has a 64 Mbit density arranged in 8-bit words, allowing up to 32 minutes of audio to be stored.

The ISD3900 has a set of registers that control every aspect of the device: the entire sound path can be configured by writing to the proper config registers over the SPI interface. The ISD3900 also handles Flash-related commands, such as formatting, erasing, reading, writing... Although simplified, programming the ISD3900 can seem complex, but when used within the MIKROE environment, it is very easy to operate the Click board™. It is supported by the mikroSDK compatible library, which contains functions for accelerated software development. Several pre-configured sound path settings are available, saving valuable time.

The Click board™ is designed to work with 3.3V only. When using it with MCUs that use 5V levels for their communication, a proper level translation circuit should be used.


## SPECIFICATIONS

Type	Audio
Applications	It is an ideal solution for POS terminals, toys, home appliances, digital voice recorders, and all other applications that require a simple interface for sound notifications.

<b>On-board modules</b>	ISD3900, a multi-message record and playback device with ChipCorder® technology, from Nuvoton; W25Q64, a 64 Mbit serial Flash memory from Winbond.
<b>Key Features</b>	An integrated solution with the ChipCorder® technology, offering digital compression, smart message management, digitally configurable signal path, and more. Integrated speaker and microphone offer out-of-box functionality. Offers numerous connectivity options.
<b>Interface</b>	GPIO,SPI
<b>Input Voltage</b>	3.3V
<b>Click board size</b>	L (57.15 x 25.4 mm)

## PINOUT DIAGRAM

This table shows how the pinout on **Rec&Play Click** corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin					Pin	Notes
SPI Ready Status	<b>RDY</b>	1	AN	PWM	16	NC	
Reset	<b>RST</b>	2	RST	INT	15	<b>INT</b>	Interrupt
SPI Chip Select	<b>CS</b>	3	CS	RX	14	NC	
SPI Clock	<b>SCK</b>	4	SCK	TX	13	NC	
SPI Data OUT	<b>SDO</b>	5	MISO	SCL	12	NC	
SPI Data IN	<b>SDI</b>	6	MOSI	SDA	11	NC	

Power Supply	3V3	7	3.3V	5V	10	NC	
Ground	GND	8	GND	GND	9	GND	Ground

## ONBOARD SETTINGS AND INDICATORS

Label	Name	Default	Description
LD1	PWR	-	Power LED indicator
JP1	SPK SEL	Right	Speaker output selection: left position external speaker (EXT), right position internal speaker (INT)
AUX OUT	CN1	-	Auxiliary line-level output
AUX IN	CN2	-	Auxiliary line-level input
EXT SPK	CN3	-	External Speaker / BTL output connector

## SOFTWARE SUPPORT

We provide a library for the **Rec&Play Click** on our [LibStock](#) page, as well as a demo application (example), developed using MikroElektronika [compilers](#). The demo can run on all the main MikroElektronika [development boards](#).

### Library Description

Library offers a choice to perform control of the REC&PLAY Click board by sending commands to the ISD3900 via SPI interface. Commands include message recording, recorded message playing, sending or receiving compressed audio data, digital reading and writing, device configuration, device status and interrupt checking... All functions return device status byte, so on that way user can check status of the device and SPI interface after each sent command. For more details check documentation.

## Key functions:

- `T_RECPLAY_RETVAL recplay_recordMsg( void )` - Function initiates a managed record at first available location in memory.
- `T_RECPLAY_RETVAL recplay_playMsg( uint32_t memAddr, uint16_t sectorOffset )` - Function initiates a play of a recorded message starting at a specified address, with a specified sector offset.
- `T_RECPLAY_RETVAL recplay_stop( void )` - Function stops any current audio command.
- `T_RECPLAY_RETVAL recplay_eraseMsg( uint32_t memAddr )` - Function erases the message starting at the specified address.

## Examples description

The application is composed of the three sections :

- System Initialization - Initializes peripherals and pins.
- Application Initialization - Initializes SPI interface in properly mode and performs the all necessary commands to put device in properly working mode (chip reset, chip power up, chip erasing, clock configuration).
- Application Task - (code snippet) - Performs the chip configuration for recording message via microphone, then records a message for 8 seconds to specified memory location. After that reads the recorded message address with message length and then plays a recorded message. When playback is done erases a recorded message from memory. Repeats the all operation every 10 seconds. Note : The ISD3900 must be properly configured to work in record mode every time when user wants to record a message. When user wants to play a recorded message, then ISD3900 must be properly configured, but now to work in play mode.

```
void applicationTask()
{
    mikrobus_logWrite( "Preparing to record a message...", _LOG_LINE );
    for (i = 0; i < 32; i++)
    {
        if ((i != _RECPLAY_CFG0A_REG) && (i != _RECPLAY_CFG1C_REG) && (i != _RECPLAY_CFG1E_REG))
        {
            waitReady();
            tempVar = configRecMic[ i ];
            statusByte = recplay_writeCnfgReg( i, &tempVar, 1 );
        }
    }
    waitReady();
    Delay_ms( 2000 );

    mikrobus_logWrite( "Message Recording", _LOG_TEXT );
    statusByte = recplay_recordMsgAddr( 0x12000 );
}
```

```

timeRecord( 8 );
statusByte = recplay_stop();
waitCmdFin();
mikrobus_logWrite( "End of recording", _LOG_LINE );

mikrobus_logWrite( "-----", _LOG_LINE );
statusByte = recplay_readMsgAddr( &messageAddr, &messageLength );
LongWordToHex( messageAddr, text );
mikrobus_logWrite( "Message Address: 0x", _LOG_TEXT );
mikrobus_logWrite( text, _LOG_LINE );
WordToStr( messageLength, text );
mikrobus_logWrite( "Message Length: ", _LOG_TEXT );
mikrobus_logWrite( text, _LOG_LINE );
mikrobus_logWrite( "-----", _LOG_LINE );
Delay_ms( 1000 );

mikrobus_logWrite( "Preparing to play a message...", _LOG_LINE );
setVolume( 100 );
for (i = 0; i < 32; i++)
{
    if ((i != _RECPLAY_CFG0A_REG) && (i != _RECPLAY_CFG1C_REG) && (i != _RECPLAY_CFG1E_REG
))
    {
        waitReady();
        if (i == _RECPLAY_CFG03_REG)
        {
            tempVar = volume;
        }
        else
        {
            tempVar = configPlayPwmSpk[ i ];
        }
        statusByte = recplay_writeCnfgReg( i, &tempVar, 1 );
    }
}
waitReady();
Delay_ms( 2000 );

```

```

mikrobus_logWrite( "Message is playing...", _LOG_LINE );
statusByte = replay_playMsg( 0x12000, 0 );
waitCmdFin();
mikrobus_logWrite( "End of playing", _LOG_LINE );

mikrobus_logWrite( "-----", _LOG_LINE );
ByteToHex( statusByte, text );
mikrobus_logWrite( "Status Byte: 0x", _LOG_TEXT );
mikrobus_logWrite( text, _LOG_LINE );
ByteToHex( interrByte, text );
mikrobus_logWrite( "Interrupt Byte: 0x", _LOG_TEXT );
mikrobus_logWrite( text, _LOG_LINE );
mikrobus_logWrite( "-----", _LOG_LINE );
Delay_ms( 1000 );

mikrobus_logWrite( "Message Erasing...", _LOG_LINE );
statusByte = replay_eraseMsg( 0x12000 );
waitCmdFin();
mikrobus_logWrite( "End of erasing", _LOG_LINE );

mikrobus_logWrite( "*****", _LOG_LINE );
mikrobus_logWrite( "", _LOG_LINE );
Delay_ms( 10000 );
}

```

### Additional Functions :

- `void waitCmdFin()` - Waits until current command has properly finished.
- `void waitReady()` - Waits until SPI interface was ready for new command.
- `void waitPowerUp()` - Waits until ISD3900 has powered up.
- `void timeRecord( uint32_t secondsTime )` - Performs the desired time in seconds for message recording.
- `void setVolume( uint8_t volumeSel )` - Sets a volume on the desired value (0 - 100%) for a message playing.

The full application code, and ready to use projects can be found on our [LibStock](#) page.  
Other mikroE Libraries used in the example:

- [Conversions](#)
- [SPI](#)
- [UART](#)



## Additional notes and informations

Depending on the development board you are using, you may need [USB UART click](#), [USB UART 2 click](#) or [RS232 click](#) to connect to your PC, for development systems with no UART to USB interface available on the board. The terminal available in all MikroElektronika [compilers](#), or any other terminal application of your choice, can be used to read the message.

## MIKROSDK

This click board is supported with [mikroSDK](#) - MikroElektronika Software Development Kit. To ensure proper operation of mikroSDK compliant click board demo applications, mikroSDK should be downloaded from the [LibStock](#) and installed for the compiler you are using.

For more information about mikroSDK, visit the [official page](#).

