
Audio Recording and Playback Using Direct Memory Access and Core Independent Peripherals

Introduction

Author: Max Prasad, Microchip Technology Inc.

This application note describes how to record, store and playback a mono audio sound using Direct Memory Access (DMA) and various Core Independent Peripherals (CIP) present on the newer PIC18 devices.

One of the key challenges today is to ensure there is a fast and reliable method of data transfer between memories, both on-chip and external, that does not affect the throughput of the system CPU. In this application, multiple DMA channels are used to transfer data between RAM and Flash memory. This allows for higher data transfer speed and ease of use while freeing the CPU to perform other tasks. The typical applications for use are data loggers, answering machines, voice recorders and toys.

This application note is based on the PIC18F57Q43 Curiosity Nano platform. It details the usage of ADC for sound recording, SPI for interfacing with the external memory module, DAC for playback, DMA for data transfers, and the use of timers to define sampling frequency. The application also demonstrates the use of Timer2 HLT and CLC for switch debouncing and the Comparator and PWM for Class D power amplifier drive.

Features

- 8-Bit Recording and Playback
- High-Speed Data Transfers Using DMA
- Excellent for Recording and Playing Back Speech
- Configurable Sampling Rate for High-Quality Sound Recording or Longer Record Time
- Very Small Code Size
- Internal and External Data Storage Options Available
- CIPs Used: ADC, DAC, DMA, SPI, NVM Controller, CMP, PWM, TMR2 and CLC
- Code Easily Ported into Other PIC® Devices

Figure 1. System Block Diagram

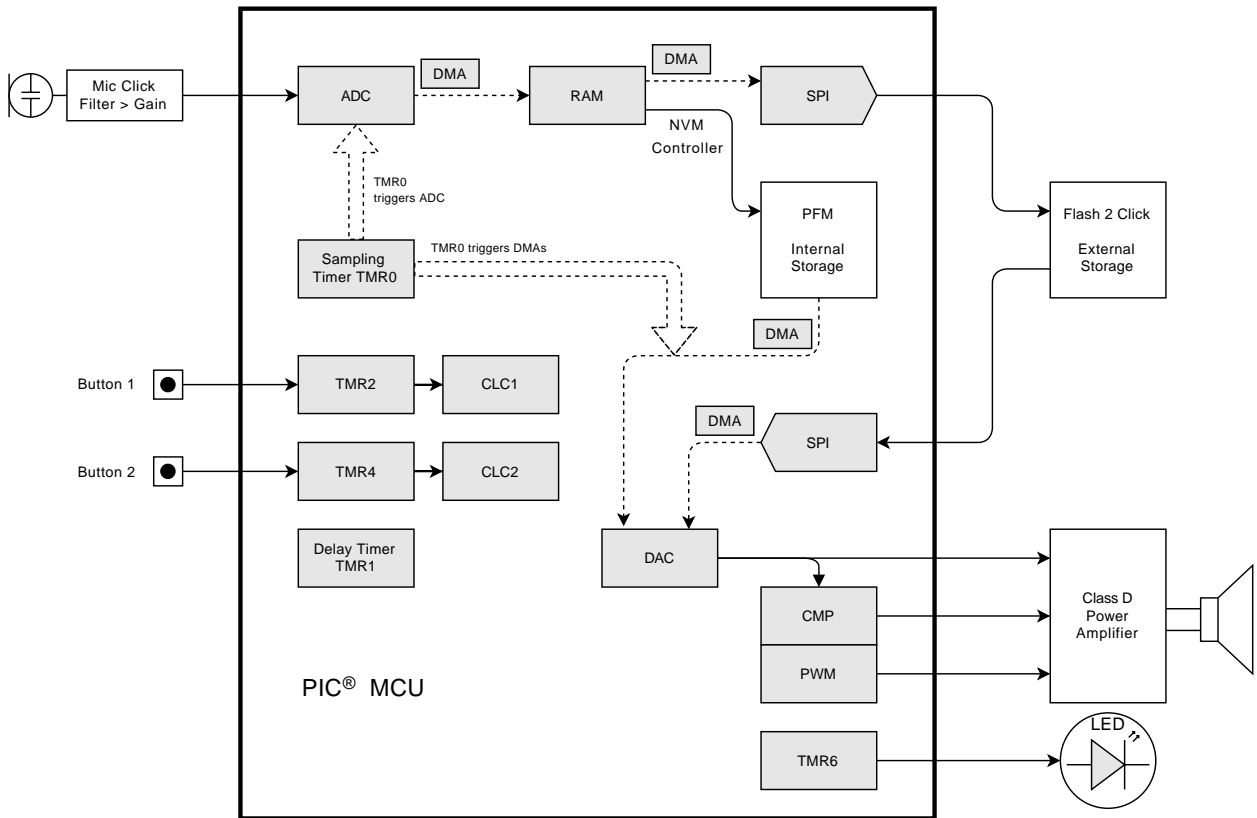


Table of Contents

Introduction.....	1
1. Features.....	1
1. Theory of Operation.....	5
1.1. Analog-to-Digital Conversion (ADC).....	5
1.2. Digital-to-Analog Conversion (DAC).....	6
2. Memory Organization.....	8
2.1. Data Memory Organization.....	8
2.2. Internal Program Memory Organization.....	9
2.3. External Memory Organization.....	9
3. Data Transfer Operations.....	11
3.1. Data Transfer from ADC to Ping-Pong Buffers.....	12
3.2. Data Transfer from Ping-Pong Buffers to External Memory.....	14
3.3. Data Transfer from Ping-Pong Buffers to Internal Memory.....	15
3.4. Data Transfer from External Memory to DAC.....	16
3.5. Data Transfer from Internal Memory to DAC.....	16
3.6. Data Transfer from ADC to DAC (Passthrough).....	17
4. Switch Debouncing.....	19
5. Application Setup.....	20
5.1. CPU Setup.....	20
5.2. TMR0 Setup.....	20
5.3. ADCC Setup.....	20
5.4. DAC1 Setup.....	21
5.5. SPI1 Setup.....	21
5.6. DMA Setup.....	21
5.7. PWM1 Setup.....	21
5.8. CMP1 Setup.....	22
5.9. TMR2/4 Setup.....	22
5.10. CLC1/2 Setup.....	22
5.11. TMR6 Setup.....	23
5.12. TMR1 Setup.....	24
5.13. UART1 Setup.....	24
6. Software Implementation.....	25
6.1. Setup, Main Loop and Idle Subroutine.....	25
6.2. Record Subroutine.....	27
6.3. Playback Subroutine.....	29
6.4. Erase Subroutine.....	30
6.5. Passthrough Subroutine.....	30
7. Hardware Implementation.....	32
7.1. Hardware Setup Using Curiosity Nano Evaluation Kit.....	32
7.2. Mic Click Board™ and Microphone Circuit.....	32

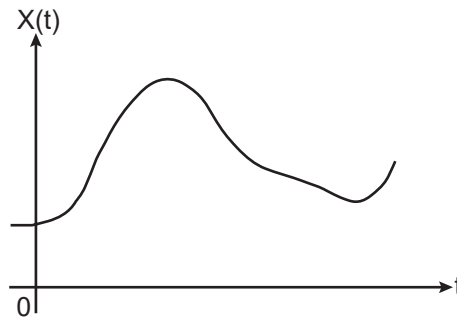
7.3. Power Amplifier.....	33
8. Conclusion.....	38
9. References.....	39
10. Revision History.....	40
The Microchip Website.....	41
Product Change Notification Service.....	41
Customer Support.....	41
Microchip Devices Code Protection Feature.....	41
Legal Notice.....	42
Trademarks.....	42
Quality Management System.....	43
Worldwide Sales and Service.....	44

1. Theory of Operation

1.1 Analog-to-Digital Conversion (ADC)

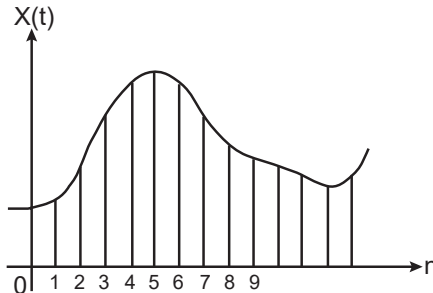
Before an analog voice signal can be stored digitally, it must be converted to a digital signal. This is done in multiple steps.

Figure 1-1. Example Analog Signal



First, the analog signal (an example shown in [Figure 1-1](#)), is converted to a time-discrete signal by taking periodic samples, as shown in [Figure 1-2](#). The time interval between two samples is called the sampling period and its reciprocal is called the sampling frequency. According to the Nyquist-Shannon sampling theorem, the sampling frequency has to be at least double the maximum frequency component present in the sampled signal. Otherwise, the periodic continuation of the signal in the frequency domain will result in spectral overlap, called aliasing. An aliased signal can not be uniquely reconstructed from its samples.

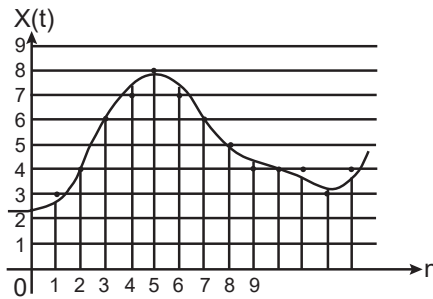
Figure 1-2. Discrete Time Signal



A speech signal contains its major information below 3 kHz; therefore a low-pass filter can be used to band-limit the signal. For an ideal low-pass filter with a cut-off frequency of 3 kHz, the sampling frequency must be 6 kHz or more. Depending on the filter, the filter slope is more or less steep. For a first-order filter like the RC filter used in this application, it is particularly necessary to choose a much higher sampling frequency. The upper limit is set by the features of the Analog-to-Digital Converter (ADC). See [1.1.1 ADC Data Acquisition Configuration](#) for details on how this has been implemented in the application.

Determining the digital values that represent the analog samples taken at this sampling frequency is called quantization. The analog signal is quantized by assigning an analog value to the nearest allowed digital value, as shown in [Figure 1-3](#). The number of available digital values is called resolution and is always limited to the resolution of the ADC being used (for example, an 8-bit ADC can have up to 256 values). Therefore, the quantization of the analog signals always results in a loss of information. This quantization error is inversely proportional to the resolution of the digital signal. It is also inversely proportional to the dynamic range of the signal, or the range between minimum and maximum values. The conversion range of the ADC can be adjusted to the dynamic range of the signal by setting the voltage reference to a maximum value suitable for the application.

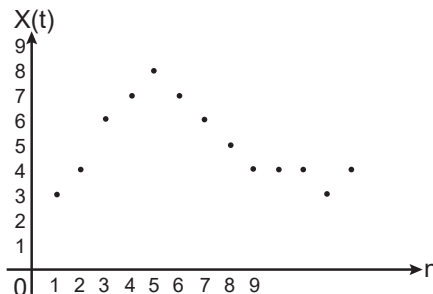
Figure 1-3. Quantized Signal



Alternatively, a microphone amplifier can be designed to cover the ADC dynamic range. Both methods reduce the quantization error.

Figure 1-4 shows the digital values that represent the analog signal. These are the values that are read as ADC conversion results and can be stored in memory.

Figure 1-4. Digital Signal



1.1.1 ADC Data Acquisition Configuration

In this application, the following configuration is used for ADC data acquisition:

- Sampling Frequency (F_s) = 10 kHz
- Sampling Interval ($T_s = 1/F_s$) = 100 μ s
- Resolution = 8-bit⁽¹⁾

This is implemented by using TMR0 to generate a 10 kHz timer signal to trigger the ADC. See [5.2 TMR0 Setup](#) and [5.3 ADCC Setup](#) to set up TMR0 and ADC with Computation (ADCC).



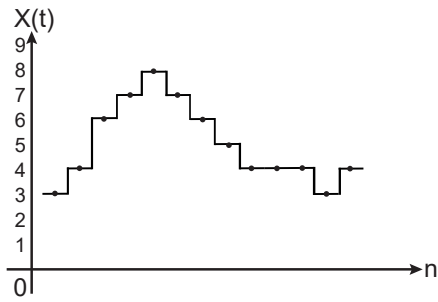
Important:

1. While most of the newer PIC18 devices have a 10-bit or 12-bit ADC, only 8-bit ADC values are used to keep the application simple. This is because the DAC used to playback the audio signal has a resolution of 8 bits.

1.2 Digital-to-Analog Conversion (DAC)

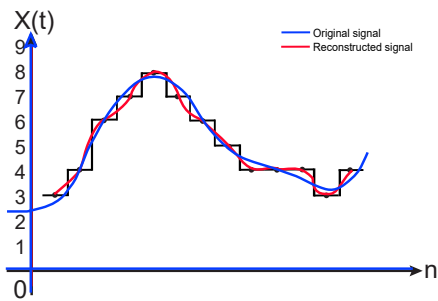
The Digital-to-Analog Converter (DAC) peripheral can be used to convert the digitally stored values to an analog output. By setting the voltage reference correctly to define maximum output value and use a timer to replicate sampling frequency as output frequency, the original signal is reconstructed. Each sample is held on the output for one output period (corresponding to the time in between the samples being taken). The example signal will look similar to the one shown in [Figure 1-5](#).

Figure 1-5. Reconstructed Signal



The output filter smoothens the signal from the DAC. In the example used, the output signal will end up similar to [Figure 1-6](#). The signal is very similar to the original analog input signal except for the error from quantization, which is significant in the example displayed since the 3-bit samples have only eight possible values.

Figure 1-6. Output Signal After Filtering



2. Memory Organization

As part of this application, the following on-chip and external memories will be accessed:

- On-chip Data Memory (RAM)
- Internal Programmable Flash Memory (PFM)
- External Flash Memory

2.1 Data Memory Organization

Data memory in PIC18F57Q43 comprises of three sections:

- Special Function Registers (SFR space) - Banks 0 through 4
- General Purpose Registers (GPR space) - Banks 5 through 36
- Buffer RAM - Bank 37

Each bank contains 256 bytes of memory. Refer to the **“Memory Organization”** chapter of the [PIC18F57Q43 device data sheet](#) for more details about how the memory is organized.

In this application, two GPR banks are used as ping-pong buffers and buffer RAM is used to write audio samples to either the internal program Flash memory or an external Flash memory.

2.1.1 Buffer RAM

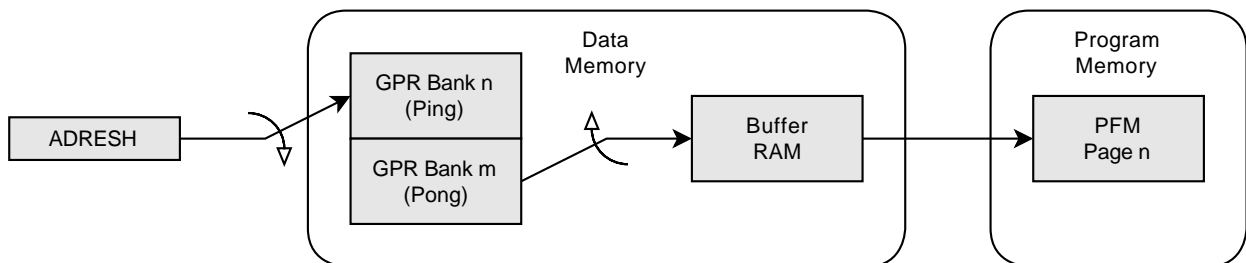
The Non-Volatile Memory (NVM), like the Programmable Flash Memory (PFM) in the PIC18F57Q43, is accessed using an NVM Controller. This uses a special bank at the end of the GPR space to act as a buffer for memory transfers and is called Buffer RAM. All page read/write operations on NVM use buffer RAM to transfer data. Refer to the device data sheet for more information.

2.1.2 Ping-Pong Buffers

In order to keep up with the sampling rate and prevent loss of data, two GPR banks are used as ping-pong buffers to write to buffer RAM. A ping-pong buffer is a double-buffer, which increases the throughput of the system by speeding up the read/write process. In the application, GPR banks 11 and 12 are used as ping-pong buffers.

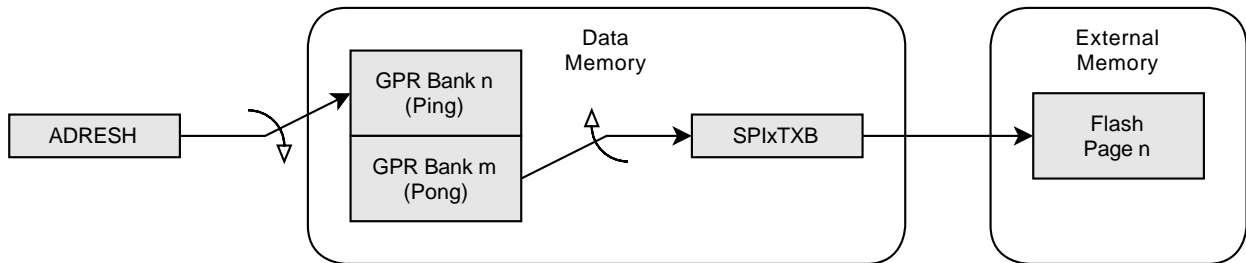
In [Figure 2-1](#), GPR banks n and m are arranged together as a ping-pong buffer. When the DMA is writing data to GPR Bank n (ping), another DMA is busy transferring data out of GPR bank m (pong) into the buffer RAM. The NVM Controller then writes the contents of the buffer RAM into the selected PFM page. See [3.3 Data Transfer from Ping-Pong Buffers to Internal Memory](#) for details on how to write to internal memory.

Figure 2-1. Ping-Pong Buffer Transfer to Internal Memory



If the data is being transferred to external Flash memory, then another DMA can be configured to write to the SPI transfer buffer instead. The SPI must be preconfigured to write to the external memory. See [3.2 Data Transfer from Ping-Pong Buffers to External Memory](#) for details on how to write to external memory.

Figure 2-2. Ping-Pong Buffer Transfer to External Memory



Tip: It is recommended for the two GPR banks for ping and pong buffers to reside adjacent to each other in memory. This allows one single DMA to service the entire ping buffer and pong buffer spaces, without the need to reconfigure DMA when transitioning between buffers. If the ping buffer and pong buffer are not adjacent, then the DMA will need to be reconfigured whenever buffer transition is required.

2.2 Internal Program Memory Organization

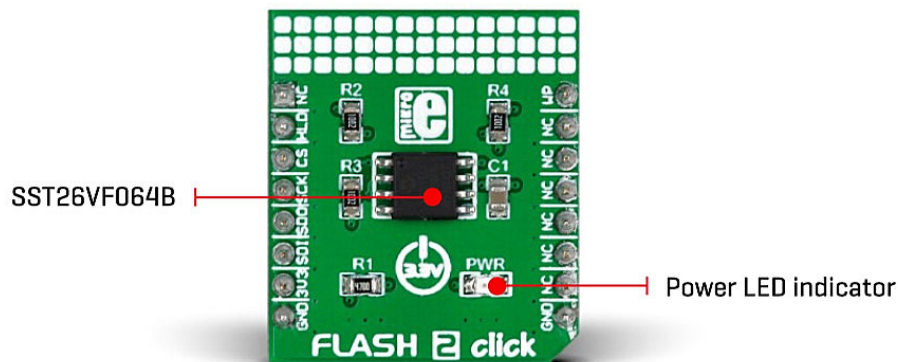
PIC18F57Q43's Programmable Flash Memory (PFM) is divided into 512 128-word pages (512 pages x 128 words = 65,536 words = 131,072 bytes). A read from program memory is executed either one word, one byte or one 128-word page at a time. A program memory erase is executed on a 128-word page at a time. A write to program memory can be executed as either 1 or 128 words at a time.

See [3.3 Data Transfer from Ping-Pong Buffers to Internal Memory](#) and [3.5 Data Transfer from Internal Memory to DAC](#) for details on how to write to and read from the internal memory.

2.3 External Memory Organization

The Flash 2 Click is used as the external memory module for this application. The Flash memory module used on this Click Board™ is the [SST26VF064B](#), a 64 Mbit serial Flash memory from Microchip Technology. SST26VF064B contains 67,108,864 bits organized in units of 8 bits, which gives 8,388,608 bytes of data memory. This memory module contains 2048 sectors of 4 KB each. Furthermore, the memory is organized in pages. One page holds 256 bytes and there are 32,768 pages (32,768 pages x 256 bytes = 8,388,608 bytes total). The SST26VF064B module communicates to the microcontroller via SPI interface.

Figure 2-3. Flash 2 Click (Image Courtesy of MikroElektronika)



The SST26VF064B can be written one page at a time. The `Page Program` instruction allows up to 256 bytes to be written during one write cycle. Data can be read by the `Read Data Bytes` instruction. While reading, the address increment is automatically executed, making it possible to read the entire memory by a single `Read Data` command. The `Page Program` instruction can only reset the bits to 0. Therefore, a segment of memory needs to be erased prior to programming or, in this case, filled with 1s (0xFF). The SST26VF064B allows erasing one sector at a time (`Sector Erase` instruction), one block at a time (`Block Erase` instruction) and the entire memory (using the `Chip Erase` instruction).

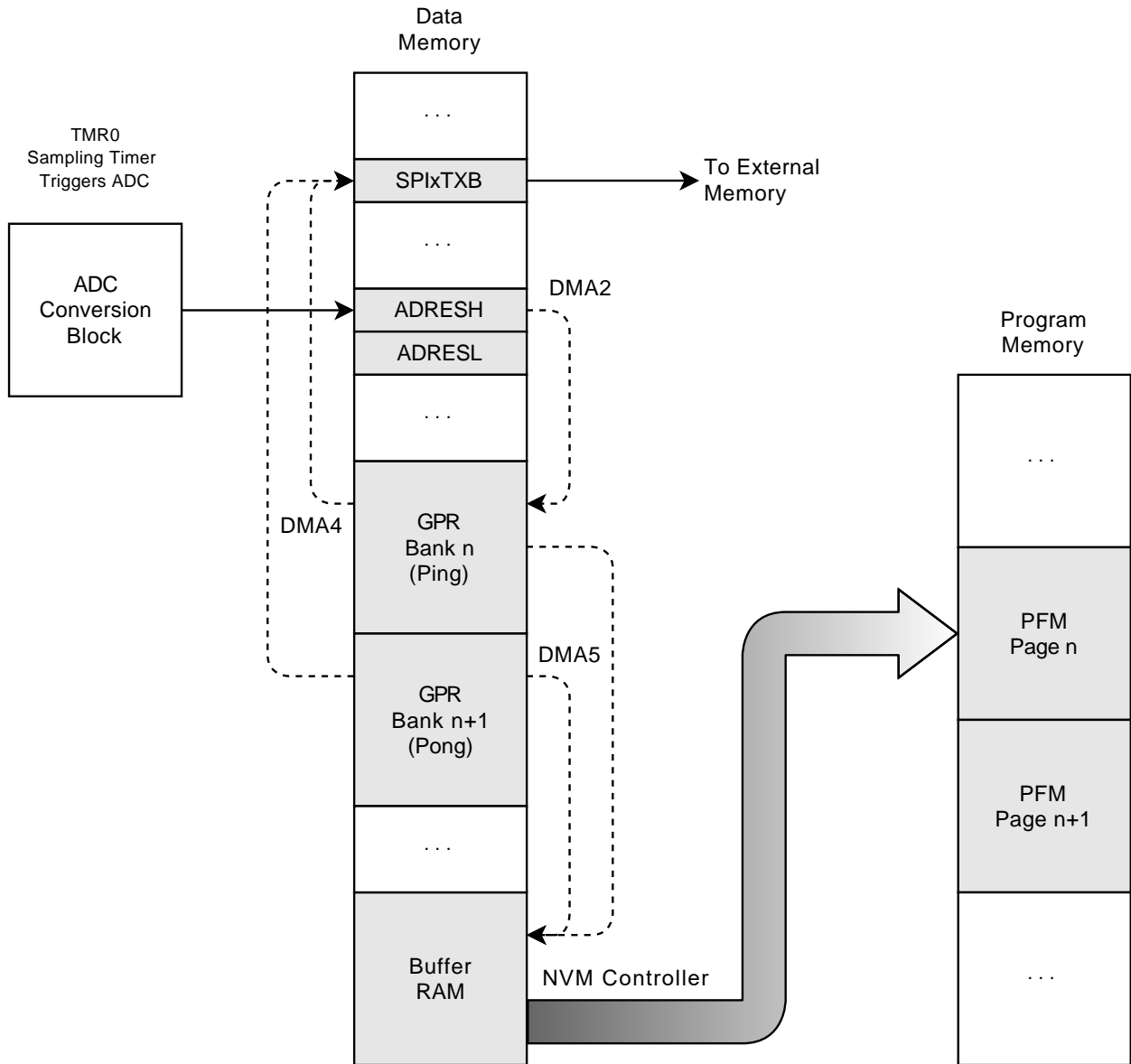
See [3.2 Data Transfer from Ping-Pong Buffers to External Memory](#) and [3.4 Data Transfer from External Memory to DAC](#) for details on how to write to and read from the external memory.

3. Data Transfer Operations

The following data transfers take place when recording audio data:

- ADC to ping-pong buffers
- Ping-pong buffers to external memory
- Ping-pong buffers to internal memory

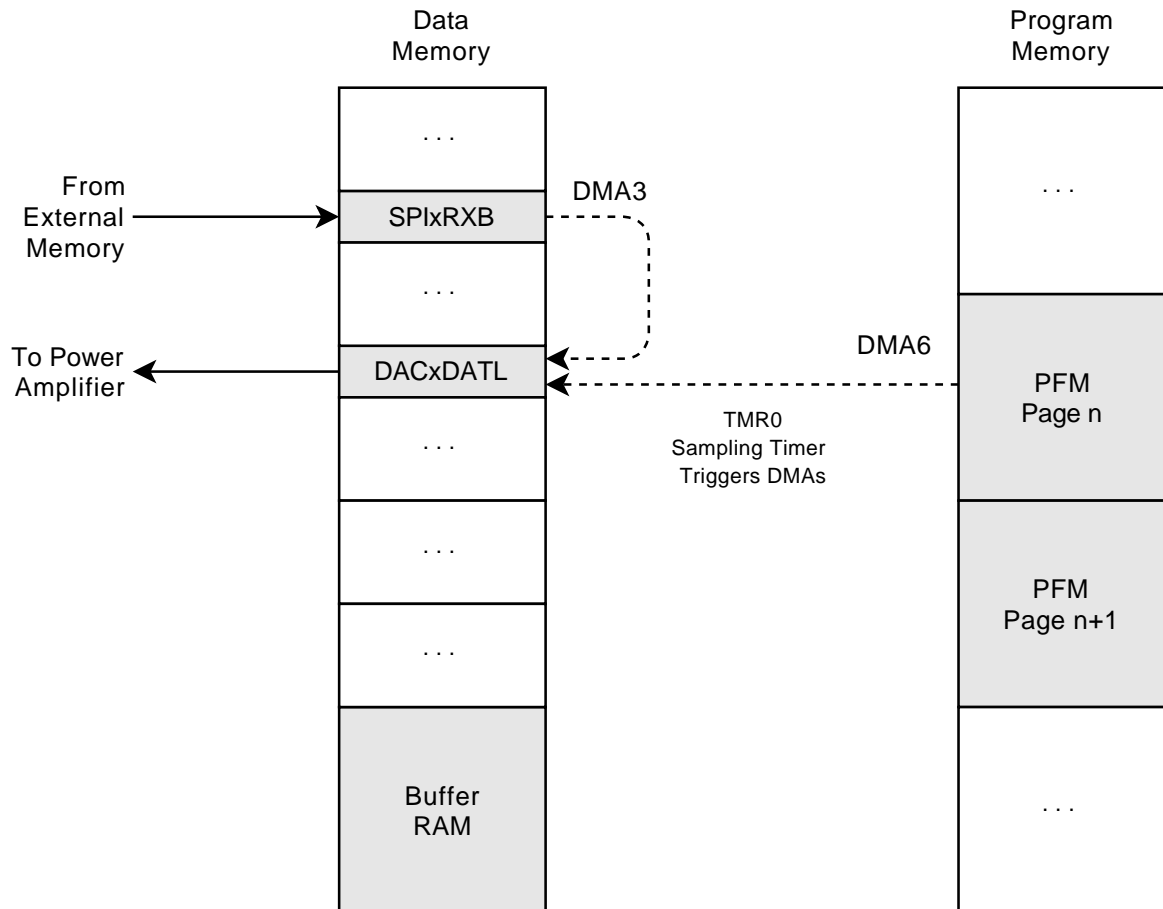
Figure 3-1. Data Transfers for Audio Recording



The following data transfers take place when playing back audio data:

- External memory to DAC
- Internal memory to DAC

Figure 3-2. Data Transfers for Audio Playback



The following data transfer takes place in an audio passthrough channel:

- ADC to DAC (passthrough)

3.1 Data Transfer from ADC to Ping-Pong Buffers

The ADC is auto-triggered from a sampling timer. See [1.1 Analog-to-Digital Conversion \(ADC\)](#) for details about ADC sampling and acquisition. The 12-bit conversion result from the ADC is available in the ADRESH:L registers. The ADC result can be left-aligned so that 8-bit values stored in ADRESH register can be transferred directly to the ping-pong buffers using DMA⁽¹⁾. The DMA setup configurations are listed in [Table 3-1](#).

Table 3-1. DMA Setup Configuration (DMA2)

DMA2 Setup		
SSA	Source Address	&ADRESH
SSZ	Source Size	1
SMR	Source Memory Region	SFR
SMODE	Source Address Mode	SPTR remains unchanged

.....continued		
DMA2 Setup		
DSA	Destination Address	Beginning of ping buffer
DSZ	Destination Size	512 (2 banks for ping and pong buffers)
DMODE	Destination Address Mode	DPTR is incremented
SIRQEN	Start Trigger Enable	Yes
SIRQ	Start Trigger	AD (ADC conversion)
SSTP	Source Counter Reload Stop	No - SIRQEN not cleared
DSTP	Destination Counter Reload Stop	No - SIRQEN not cleared
AIRQEN	Abort Trigger Enable	No (stops when SCNT reloads)
AIRQ	Abort Trigger	N/A

How It Works:

1. The DMA is triggered as soon as an ADC conversion is complete (ADIF trigger) and the content of the ADRESH register is transferred to the first address in the ping buffer. DPTR is incremented and now points to the next address in the ping buffer. Because SSZ = 1, the DMA stops after the first transfer and waits for the next trigger.
2. After 256 such iterations, the ping buffer is full and a timer interrupt sends a signal to the software to initiate data transfer from the ping buffer to memory. Meanwhile, DPTR increments and enters the pong buffer range⁽²⁾.
3. For the next few iterations, while the pong buffer is filling up, the software/hardware is busy transferring data out of the ping buffer.
4. After another 256 iterations (once the pong buffer is full), another timer interrupt sends another signal to the software to transfer data from the pong buffer to memory. Meanwhile DPTR reaches the end of the allocated buffer size and rolls back to the starting DSA address, which is the beginning of the ping buffer⁽²⁾.
5. This process continues as long as the recording is in progress.

**Important:**

1. While most of the newer PIC18 devices have a 10-bit or 12-bit ADC, only 8-bit ADC values are used to keep the application simple. This is because the DAC used to playback the audio signal has a resolution of 8 bits.
2. In this application, the ping and pong buffers are adjacent to each other in data memory. If ping and pong buffers are separated in the data memory, then SSA needs to be reconfigured after every 256 iterations.
3. The DMA servicing the ADC has the highest priority among all the DMAs used. This is to avoid the other DMAs from blocking the sampling and storage of new audio samples.

3.2 Data Transfer from Ping-Pong Buffers to External Memory

Data stored in the ping-pong buffers by the ADC can be seamlessly transferred to external memory using DMA and SPI. SPI can be configured to operate at an appropriate frequency (about 1 MHz) and in mode 0 (active-high clock, active to idle clock edge, input data sampled at the end). The DMA setup configurations are listed in [Table 3-2](#).

Table 3-2. DMA Setup Configuration (DMA4)

DMA4 Setup		
SSA	Source Address	Beginning of ping buffer OR pong buffer (reconfigured in software)
SSZ	Source Size	256 (1 bank for ping buffer OR pong buffer)
SMR	Source Memory Region	GPR
SMODE	Source Address Mode	SPTR is incremented
DSA	Destination Address	&SPI1TXB
DSZ	Destination Size	1
DMODE	Destination Address Mode	DPTR remains unchanged
SIRQEN	Start Trigger Enable	Yes
SIRQ	Start Trigger	SPI1TX (SPI1 TX FIFO is not full)
SSTP	Source Counter Reload Stop	Yes - SIRQEN is cleared when source counter reloads
DSTP	Destination Counter Reload Stop	No - SIRQEN not cleared
AIRQEN	Abort Trigger Enable	No (stops when DCNT reloads)
AIRQ	Abort Trigger	N/A

How It Works:

1. Before starting the DMA transfer, ensure that the microcontroller and the external memory are ready to transmit and receive data. To prepare the PIC18F57Q43 microcontroller and SST26VF064B Flash module on Flash 2 Click Board™, the following steps are required (refer to [7.1 Hardware Setup Using Curiosity Nano Evaluation Kit](#) for hardware connection between the microcontroller and the Flash module):
 - 1.1. Send the `write enable` command to the Flash module.
 - 1.2. Pull the `chip select` line low and send the `Page Program` command to the Flash module.
 - 1.3. Send the write address (the address to write to in the external memory) to the Flash module.
 - 1.4. Set SPI to operate in Transmit-only mode in PIC18F57Q43. This allows the device to send bursts of data without waiting for any data in return.
 - 1.5. Set SPI transmit counter to 256.
2. Set SSA to the beginning of the ping buffer (or pong buffer, whichever is appropriate).
3. Enable the DMA and SIRQEN. This will kick off DMA transfers until the SPI transmit counter becomes zero. At the same time, the SCNT reloads and SIRQEN is cleared by the hardware, preventing further transfers.
4. After the entire ping (or pong) buffer is transferred out through SPI, the write operation can be closed by following these steps:
 - 4.1. Reset SPI to operate in Full-duplex mode.
 - 4.2. Update the next source register to pong (or ping). This ensures the SSA is loaded with the correct buffer address for the next transaction.
 - 4.3. Release the `chip select` line.
 - 4.4. Update write address location and number of pages written in the metadata.
5. Wait for another buffer full signal in software to transfer another buffer.

3.3 Data Transfer from Ping-Pong Buffers to Internal Memory

Data stored in the ping-pong buffers by the ADC can be transferred to the internal Program Flash Memory (PFM) in a two-step process. First, the contents of the ping-pong buffers are transferred to buffer RAM using DMA. Next, the NVM Controller is used to write the buffer RAM contents to a designated page in the PFM. The DMA setup configurations are listed in [Table 3-3](#).

Table 3-3. DMA Setup Configuration (DMA5)

DMA5 Setup		
SSA	Source Address	Beginning of ping buffer
SSZ	Source Size	512 (2 banks for ping and pong buffers)
SMR	Source Memory Region	GPR
SMODE	Source Address Mode	SPTR is incremented
DSA	Destination Address	Beginning of buffer RAM
DSZ	Destination Size	256 (1 bank for buffer RAM)
DMODE	Destination Address Mode	DPTR is incremented
SIRQEN	Start Trigger Enable	No (stops when DCNT reloads)
SIRQ	Start Trigger	N/A
SSTP	Source Counter Reload Stop	N/A
DSTP	Destination Counter Reload Stop	N/A
AIRQEN	Abort Trigger Enable	No (stops when DCNT reloads)
AIRQ	Abort Trigger	N/A

How It Works:

1. Transfer the contents of the ping buffer to buffer RAM by manually starting the DMA transfer. The DMA stops as soon as the buffer RAM is full (DCNT reloads). Then, proceed to the next step to write to PFM. In the next iteration (when the pong buffer is ready to be transferred), simply start the DMA again manually. SPTR continues into the pong buffer region, whereas DPTR rolls back into buffer RAM again. When the buffer RAM is full and DMA stops again (DCNT reloads), the SPTR rolls back into the ping buffer region⁽¹⁾.
2. The buffer RAM contents can be written to PFM word-by-word at every sampling timer interrupt so that the CPU stalls are minimized^(2, 3). The transfer is performed using the write operation in the NVM Controller in PIC18F57Q43.
3. Update write address location and number of pages written in the metadata.



Important:

1. In this application, the ping and the pong buffers are adjacent to each other in data memory. If ping and pong buffers are separated in the data memory, then SSA needs to be reconfigured after every 256 iterations.
2. To write to PFM, word write is used instead of page write because page write stalls the CPU for typically 10 ms. This seriously affects the data sampling rate of ADC. However, a word write typically takes about 50 μ s, which can work with a 10 kHz ADC sampling rate.
3. PFM word writes stall the CPU for typically 50 μ s, which is about 50% time between two ADC samples (100 μ s sampling period). Therefore, an interrupt from the sampling timer is used to ensure that PFM writes occur after the ADC has finished sampling and storing data.

3.4 Data Transfer from External Memory to DAC

Data stored in the external memory can be seamlessly transferred to DAC using DMA and SPI. The SPI can be configured to operate at an appropriate frequency (about 1 MHz) and in mode 0 (active high clock, active to idle clock edge, input data sampled at the end). The DMA setup configurations are listed in [Table 3-4](#).

Table 3-4. DMA Setup Configuration (DMA3)

DMA3 Setup		
SSA	Source Address	&SPI1RXB
SSZ	Source Size	256 (1 page data from external memory)
SMR	Source Memory Region	SFR
S.MODE	Source Address Mode	SPTR remains unchanged
DSA	Destination Address	&DAC1DATL
DSZ	Destination Size	1
D.MODE	Destination Address Mode	DPTR remains unchanged
SIRQEN	Start Trigger Enable	Yes
SIRQ	Start Trigger	TMR0 (sampling timer)
SSTP	Source Counter Reload Stop	Yes - SIRQEN is cleared when source counter reloads
DSTP	Destination Counter Reload Stop	No - SIRQEN not cleared
AIRQEN	Abort Trigger Enable	No (stops when DCNT reloads)
AIRQ	Abort Trigger	N/A

How It Works:

- Before starting the DMA transfer, ensure the microcontroller and external memory are ready to receive and transmit data. To prepare the PIC18F57Q43 microcontroller and SST26VF064B Flash module on Flash 2 Click Board™, the following steps are required (refer to [7.1 Hardware Setup Using Curiosity Nano Evaluation Kit](#) for hardware connection between the microcontroller and the Flash module):
 - Pull the `chip select` line low and send the `Read` command to the Flash module.
 - Send the read address (the address to read from the external memory) to the Flash module.
 - Set SPI to operate in Receive-only mode in PIC18F57Q43. This allows the device to receive bursts of data without sending any data in return.
- Set SPI transmit counter to 256. This immediately kicks off the SPI. Enable SIRQEN to kick off DMA at the next timer overflow. The DMA and SPI transmit stop when SPI transmit counter depletes to zero (and coincidentally SCNT reloads, clearing SIRQEN). The address pointer in the external memory module is incremented automatically. Repeat this step to read as many pages of stored audio data.
- To close out the read operation, restore SPI to Full-duplex mode and release the `chip select` line.



Important: In Receive-only mode, SPI operations are suspended when the RXFIFO becomes full. In this application, the SPI operation resumes only when RXFIFO is read by the DMA (at the next TMR0 overflow event). This enables the playback and the data transfer rate from external memory to match the sampling rate.

3.5 Data Transfer from Internal Memory to DAC

Data stored in the internal Program Flash Memory (PFM) can be seamlessly transferred to DAC using DMA. The DMA setup configurations are listed in [Table 3-5](#).

Table 3-5. DMA Setup Configuration (DMA6)

DMA6 Setup		
SSA	Source Address	Beginning of space allocated for audio storage
SSZ	Source Size	256 (1 page data from PFM)
SMR	Source Memory Region	PFM
SMODE	Source Address Mode	SPTR is incremented
DSA	Destination Address	&DAC1DATL
DSZ	Destination Size	1
DMODE	Destination Address Mode	DPTR remains unchanged
SIRQEN	Start Trigger Enable	Yes
SIRQ	Start Trigger	TMR0 (sampling timer)
SSTP	Source Counter Reload Stop	Yes - SIRQEN is cleared when source counter reloads
DSTP	Destination Counter Reload Stop	No - SIRQEN not cleared
AIRQEN	Abort Trigger Enable	No (stops when DCNT reloads)
AIRQ	Abort Trigger	N/A

How It Works:

1. Enable SIRQEN to kick off DMA at the next timer overflow. This allows the DMA data transfer rate to match the sampling timer rate.
2. The DMA stops when an entire page of audio data has been read.
3. To read the next PFM page, SSA can be updated with the new address.
4. Repeat to read as many pages of stored audio data.

3.6 Data Transfer from ADC to DAC (Passthrough)

The left-aligned 8-bit ADC result in ADRESH register can be transferred directly to the DAC using DMA, creating an audio passthrough channel⁽¹⁾. The DMA setup configurations are listed in [Table 3-6](#).

Table 3-6. DMA Setup Configuration (DMA1)

DMA1 Setup		
SSA	Source Address	&ADRESH
SSZ	Source Size	1
SMR	Source Memory Region	SFR
SMODE	Source Address Mode	SPTR remains unchanged
DSA	Destination Address	&DAC1DATL
DSZ	Destination Size	1
DMODE	Destination Address Mode	DPTR remains unchanged
SIRQEN	Start Trigger Enable	Yes
SIRQ	Start Trigger	AD (ADC conversion)
SSTP	Source Counter Reload Stop	No - SIRQEN not cleared
DSTP	Destination Counter Reload Stop	No - SIRQEN not cleared

.....continued

DMA1 Setup

AIRQEN	Abort Trigger Enable	No (stops when SCNT/DCNT reloads)
AIRQ	Abort Trigger	N/A

How It Works:

1. The DMA is triggered as soon as an ADC conversion is complete (ADIF trigger) and the content of ADRESH register is transferred to the DAC1DATL register directly. Because SSZ = 1, the DMA stops after the first transfer and waits for the next trigger.
2. This process continues as long as the ADC and the DMA are in operation.

**Important:**

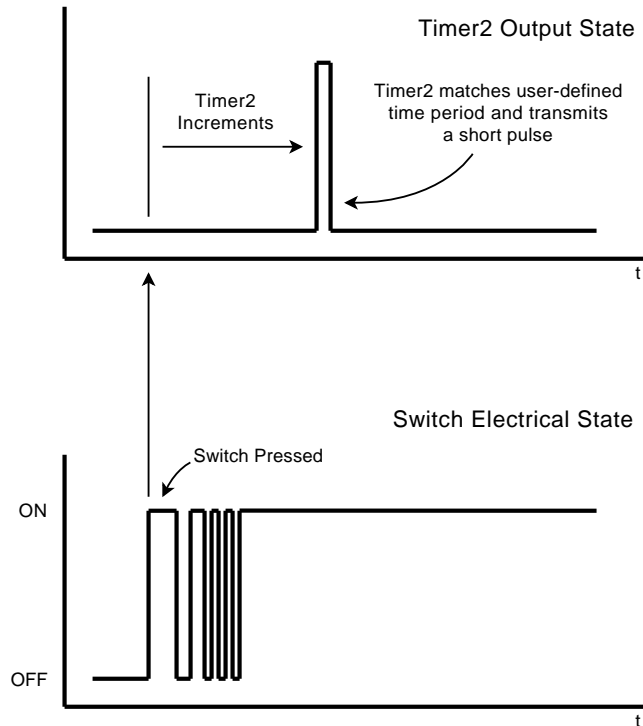
1. While most of the newer PIC18 devices have a 10-bit or 12-bit ADC, only 8-bit ADC values are used to keep the application simple. This is because the DAC used to playback the audio signal has a resolution of 8 bits.
-

4. Switch Debouncing

In this application, Timer2 and CLC are used to create a hardware based code-free switch debouncer (Figure 4-1). The buttons in the hardware are used to control the flow of the application as described in 6.1 Setup, Main Loop and Idle Subroutine.

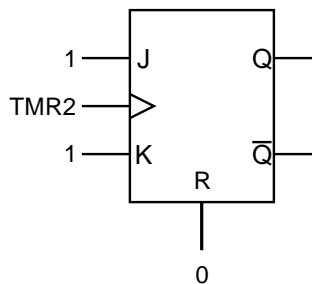
Timer2 automates the debouncing process using its monostable mode of operation. In this mode, the first switch activation is used to start the timer counting, ignoring any subsequent bouncing. Once the timer count reaches a predetermined value, the timer peripheral will produce a signaling event that can be used to indicate a valid switch activation has been detected. See 5.9 TMR2/4 Setup for details on how to set up Timer2.

Figure 4-1. Switch Electrical State vs. Timer2 Output State



The TMR2 output signal is used in conjunction with CLC to detect key presses and releases. The CLC operates as a toggle flip-flop in JK Flip-Flop with R mode of operation (Figure 4-2) where the TMR2 output acts as the clock input. Every output pulse from TMR2 toggles the output of the JK Flip-Flop from LOW-to-HIGH when the switch is pressed and from HIGH-to-LOW when the switch is released, as per the Start/Reset option configured earlier. See 5.10 CLC1/2 Setup for details on how to set up CLC.

Figure 4-2. Toggle Flip-Flop Using JK Flip-Flop



5. Application Setup

This section describes the setup of different Core Independent Peripherals (CIPs) used in the application. These settings can be used to easily configure the peripherals using MPLAB® Code Configurator (MCC).

5.1 CPU Setup

Table 5-1. CPU Setup

Parameter	Value
F _{osc}	64 MHz

5.2 TMR0 Setup

TMR0 is used to generate 10 kHz sampling timer. See [1.1.1 ADC Data Acquisition Configuration](#) for details.

Table 5-2. TMR0 Setup

Parameter	Value
Prescaler	1:1
Postscaler	1:1
Mode	16-bit
Clock Source	F _{osc} /4
Period	100 μs
Interrupt	Enabled

5.3 ADCC Setup

ADC with Computation (ADCC) is used for sampling and quantization of analog audio signal. See [1.1.1 ADC Data Acquisition Configuration](#) for details.

Table 5-3. ADCC Setup

Parameter	Value
Operation	Basic mode
ADC Clock	ADCRC
Result Alignment	Left
Positive Reference	V _{DD}
Negative Reference	V _{SS}
Auto-Conversion Trigger	TMR0
Input Channel	Application dependent

5.4 DAC1 Setup

DAC is used to output the analog audio signal.

Table 5-4. DAC1 Setup

Parameter	Value
Positive Reference	V _{DD}
Negative Reference	V _{SS}
DAC Output	Application dependent

5.5 SPI1 Setup

SPI1 is used to communicate with external memory module. See [3.2 Data Transfer from Ping-Pong Buffers to External Memory](#) and [3.4 Data Transfer from External Memory to DAC](#) for details.

Table 5-5. SPI1 Setup

Parameter	Value
SPI Clock	1 MHz
Input Data Sampled	End
SPI Mode	0
Clock Polarity	Idle: Low; Active: High
Clock Edge	Active to Idle

5.6 DMA Setup

See [3. Data Transfer Operations](#) for DMA setups for various data transfer operations.

5.7 PWM1 Setup

PWM1 is used to create a sawtooth waveform for the audio amplifier. See [7.3 Power Amplifier](#) for details.

Table 5-6. PWM1 Setup

Parameter	Value
Clock	F _{OSC}
Prescaler	None
Mode	Left-Aligned
Output Frequency	300 kHz
Output Duty Cycle	95%
Output Pin	Open Drain

5.8 CMP1 Setup

CMP1 is used to compare sawtooth waveform with DC-offset DAC output to produce PWM signal to drive the audio speaker. See [7.3 Power Amplifier](#) for details.

Table 5-7. CMP1 Setup

Parameter	Value
Positive Input	DACOUT
Negative Input	CIN1- (or application dependent)
Mode	Asynchronous
Hysteresis	Disabled
Output Polarity	Not Inverted

5.9 TMR2/4 Setup

TMR2/4, along with CLC1/2, are used to debounce buttons. See [4. Switch Debouncing](#) for details.

Table 5-8. TMR2/4 Setup

Parameter	Value
Control Mode	Monostable
ERS	T2/T4CKIPPS pins (buttons connected to these pins)
Start/Reset	Starts on rising/falling ERS edge
Clock Source	MFINTOSC 31.25 kHz
Prescaler	1:4
Postscaler	1:1
Timer Period	30 ms

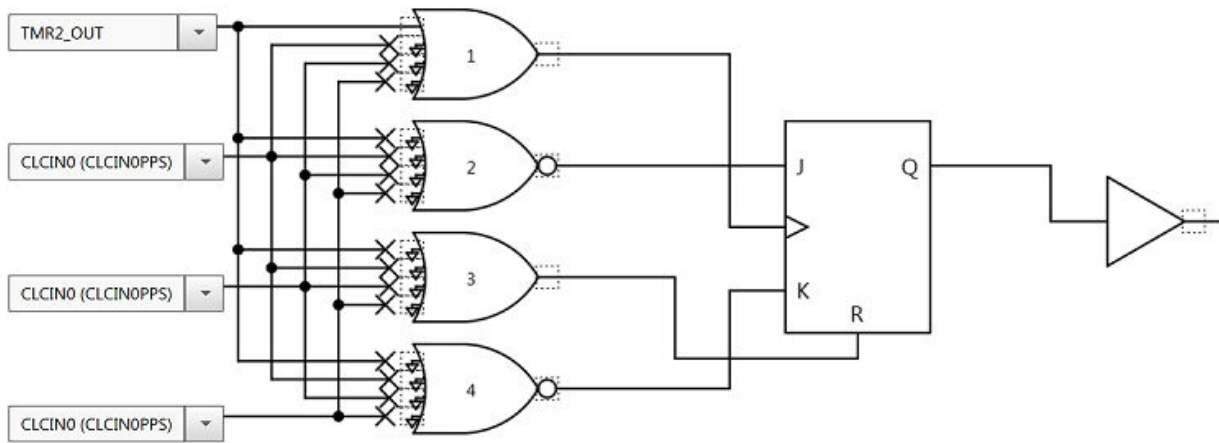
5.10 CLC1/2 Setup

CLC1/2, along with TMR2/4, are used to debounce buttons. See [4. Switch Debouncing](#) for details.

Table 5-9. CLC1/2 Setup

Parameter	Value
Interrupt	Enabled for both positive and negative edge
Mode	JK Flip-Flop with R
Input	TMR2/4 Output

Figure 5-1. CLC Setup in MCC



5.11 TMR6 Setup

TMR6 is used as a time base to blink the on-board LED. See [6.1 Setup, Main Loop and Idle Subroutine](#) for details.

Table 5-10. TMR6 Setup

Parameter	Value
Control Mode	Roll over pulse
Start/Reset	Software Control
Clock Source	MFINTOSC 31.25 kHz
Prescaler	1:16
Postscaler	1:1
Timer Period	80 ms
Enable Timer Interrupt	Yes
Callback Function Rate	1

5.12 TMR1 Setup

TMR1 is used as a time base to generate delay to distinguish between short and long button presses. See [6.1 Setup, Main Loop and Idle Subroutine](#) for details.

Table 5-11. TMR1 Setup

Parameter	Value
Clock Source	MFINTOSC 31.25 kHz
Prescaler	1:2
Enable Synchronization	Yes
Timer Period	500 ms (for single/double presses); 3 s (for long press)
Enable 16-bit Read	No
Enable Gate	No

5.13 UART1 Setup

UART1 is used for sending and receiving debug messages.

Table 5-12. UART1 Setup

Parameter	Value
Baud Rate	115200 bps
Transmit	Enabled
Receive	Enabled (optional)
Redirect STDIO	Yes (optional)

6. Software Implementation

The entire software can be modularized into the following subroutines:

- Setup and Main Loop - Initializes the program and sets up the main loop.
- Idle Subroutine - Used to decode single, double and long button presses to enter the various modes of operation.
- Record Subroutine - Used for audio recording and data storage.
- Playback Subroutine - Used for data retrieval and playback.
- Erase Subroutine - Used to erase storage media.
- Passthrough Subroutine - Used to create an audio passthrough without any data storage.



View Code Example on GitHub
Click to browse repository

6.1 Setup, Main Loop and Idle Subroutine

When the program is started, the setup of the memory and internal peripherals is initiated. This involves the setup of the following peripherals:

- ADCC for sampling of the microphone
- DAC for output to the speaker
- SPI for interfacing external memory
- PORT settings for button interfacing
- TMR2 and CLC for button debouncing
- TMR0 for correct sampling and playback frequency
- DMA for data transfer between peripherals and memory
- PWM and Comparator for power amplifier
- TMR6 as time base for LED blinking
- TMR1 as time base for button press delays
- UART for debugging

In addition to the peripherals, external memory is also initialized. If the external memory can not be identified, then the program defaults to using the internal program memory as storage.

This is followed by loading the appropriate audio metadata from the EEPROM. The audio metadata contains the memory location where existing audio is stored and the size of the audio. If no audio is stored, the audio size is zero bytes.

The setup and main loop functionality is shown in [Figure 6-1](#).

In the PIC18F57Q43 Curiosity Nano, there is only one button available (SW0 button). In this case, a single button press enters Playback mode, a double press enters Record mode, and a long-press erases the memory. There is a special button (SW1 button) available on the custom-made PROTO Click Board™ (see [Figure 7-6](#)), along with the power amplifier. A long press of this button enters into Passthrough mode. This functionality is depicted in [Figure 6-2](#).

Figure 6-1. Setup and Main Loop

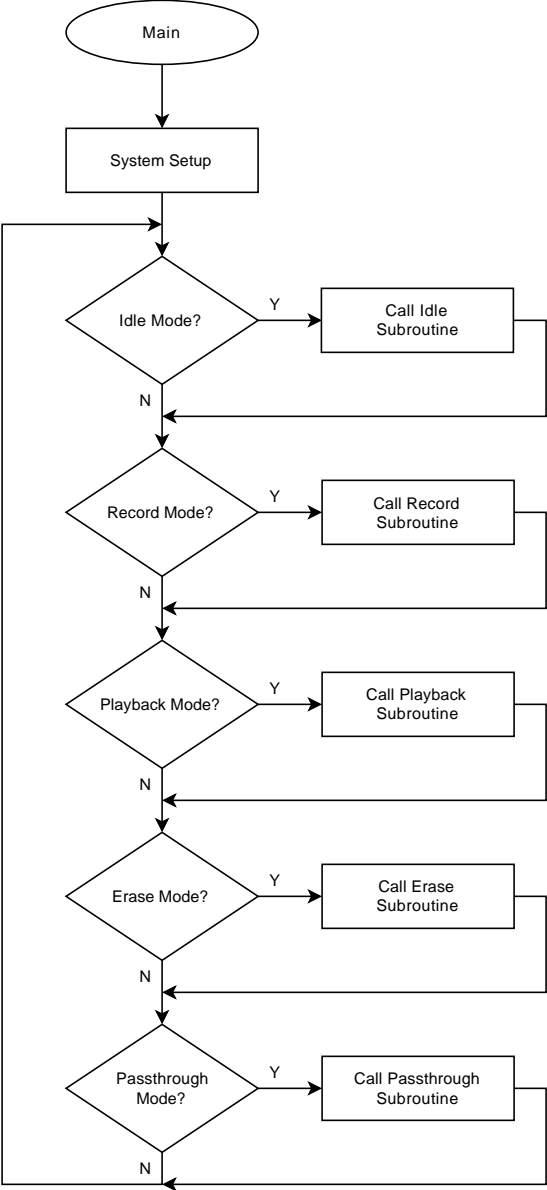
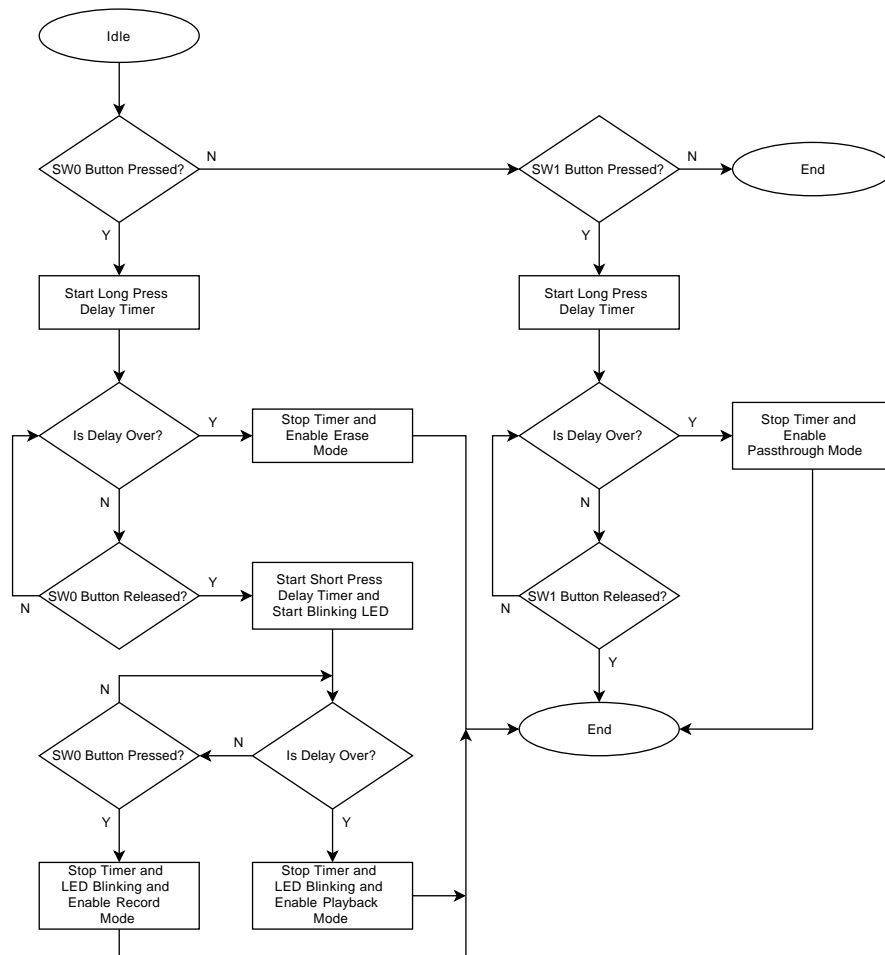


Figure 6-2. Idle Subroutine

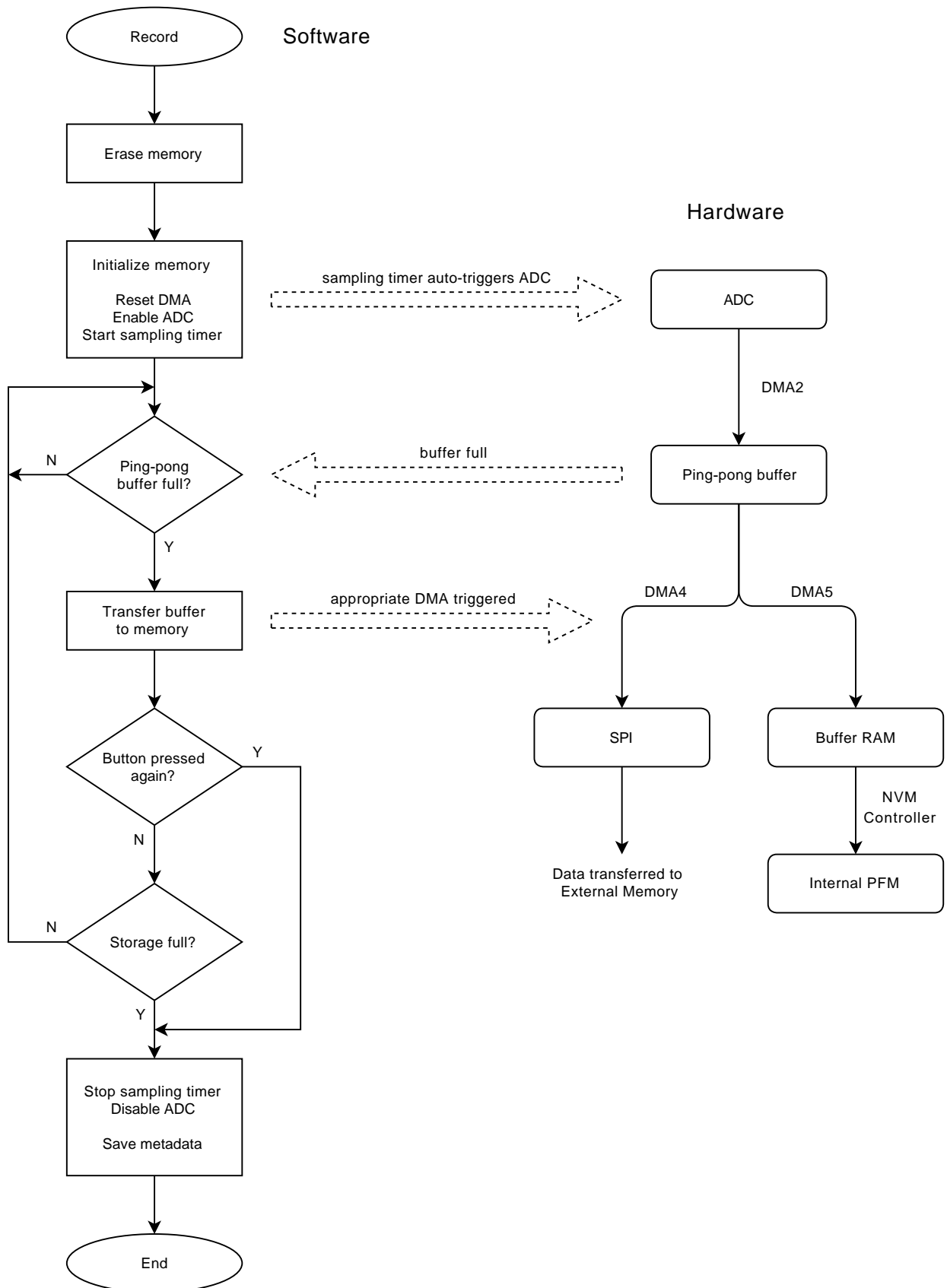


6.2 Record Subroutine

The record subroutine is shown in [Figure 6-3](#). The flowchart on the left indicates the functionality in software while the one on the right indicates hardware functionality. The record subroutine consists of the following steps:

1. As soon as the program enters Record mode, it erases the entire allocated memory to start a fresh recording.
2. The target memory is prepared to perform write operation. The ADC is turned on. The DMAs used to record and store audio are reset. The sampling timer is turned on.
3. Once the sampling timer is turned on, with every tick an ADC conversion is performed and stored in the ping-pong buffer via DMA.
4. When one of the ping-pong buffers is full, it signals the software to transfer the buffer into the target memory. Refer to [2.1.2 Ping-Pong Buffers](#) for details on ping-pong buffer transfer.
5. While software is busy transferring one of the ping-pong buffer's content into memory, the other ping-pong buffer is being populated by the ADC and DMA. Refer to [2.1.2 Ping-Pong Buffers](#) for details on ping-pong buffer transfer. Refer to [3.1 Data Transfer from ADC to Ping-Pong Buffers](#), [3.2 Data Transfer from Ping-Pong Buffers to External Memory](#) and [3.3 Data Transfer from Ping-Pong Buffers to Internal Memory](#) for details on memory operations.
6. The recording is stopped when the button is pressed again or the memory becomes full. To close the recording session, the sampling timer is stopped, ADC is disabled and the audio metadata is saved in the EEPROM.

Figure 6-3. Record Subroutine

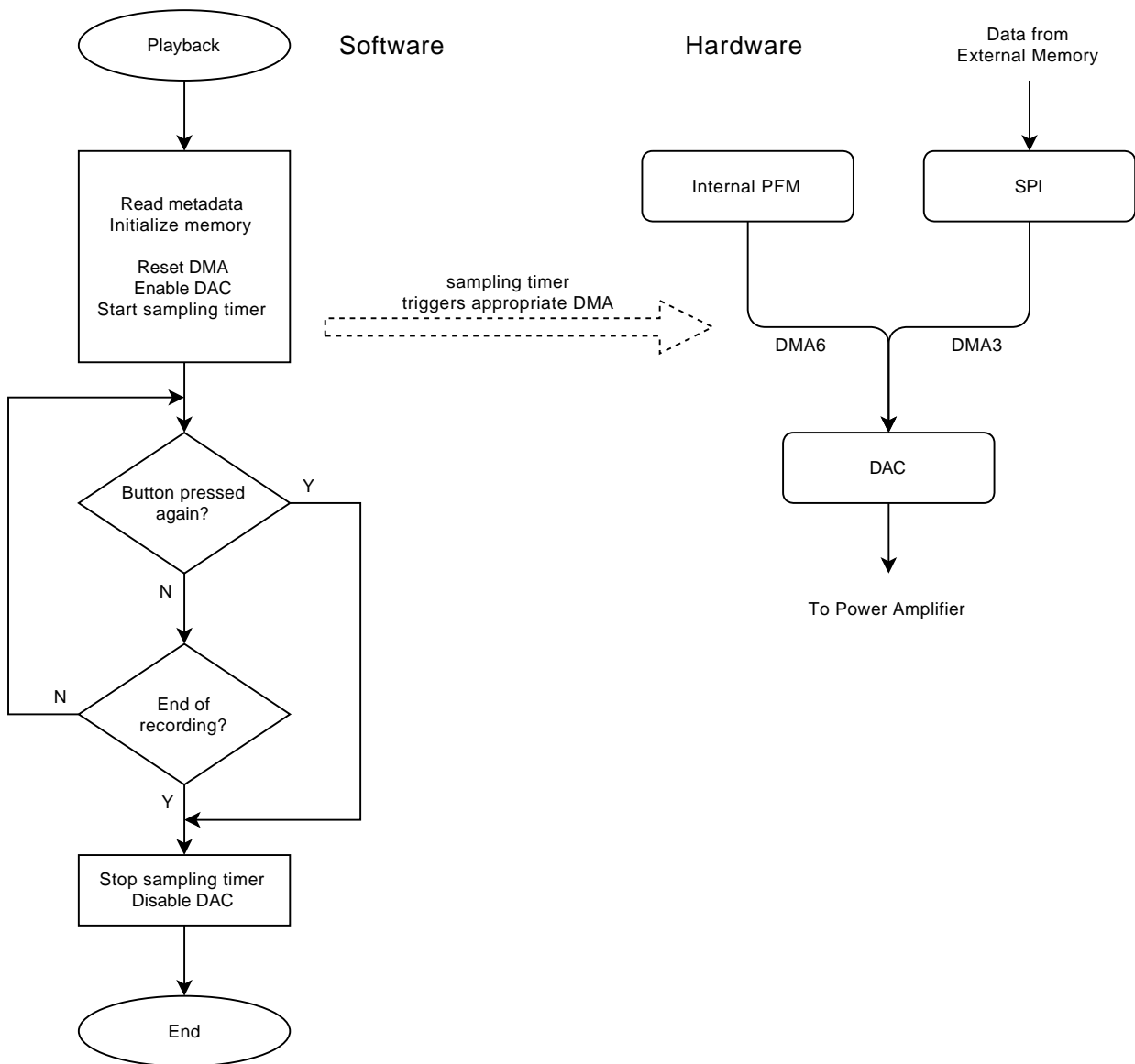


6.3 Playback Subroutine

The playback subroutine is shown in [Figure 6-4](#). The flowchart on the left indicates the functionality in software while the one on the right indicates hardware functionality. The playback subroutine consists of the following steps:

1. As soon as the program enters Playback mode, the audio metadata is read from the EEPROM.
2. The target memory is prepared to perform read operation. The DAC is turned on. The DMAs used to playback audio are reset. The sampling timer is turned on.
3. Once the sampling timer is turned on, with every tick a DMA is triggered to transfer data from the memory to DAC. This data transfer takes place one page at a time. Refer to [3.4 Data Transfer from External Memory to DAC](#) and [3.5 Data Transfer from Internal Memory to DAC](#) for details on memory operations.
4. The playback is stopped when the button is pressed again or the entire allocated memory is read. To close the playback session, the sampling timer is stopped and the DAC is disabled.

Figure 6-4. Playback Subroutine

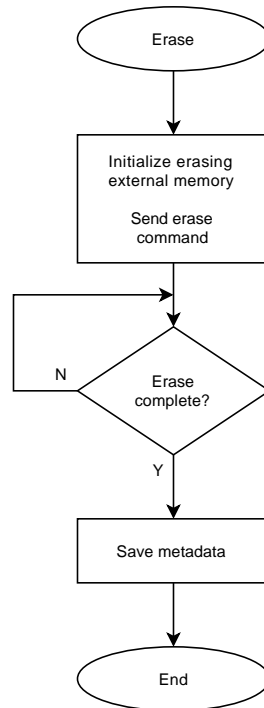


6.4 Erase Subroutine

The erase subroutine is shown in [Figure 6-5](#). The erase subroutine consists of the following steps:

1. Once the program enters Erase mode, the target memory is initialized to perform an erase operation.
2. An erase command is sent. If internal memory is used, the erase command needs to be sent for each page to be erased.
3. When the erase is complete, the audio metadata is updated in the EEPROM.

Figure 6-5. Erase Subroutine

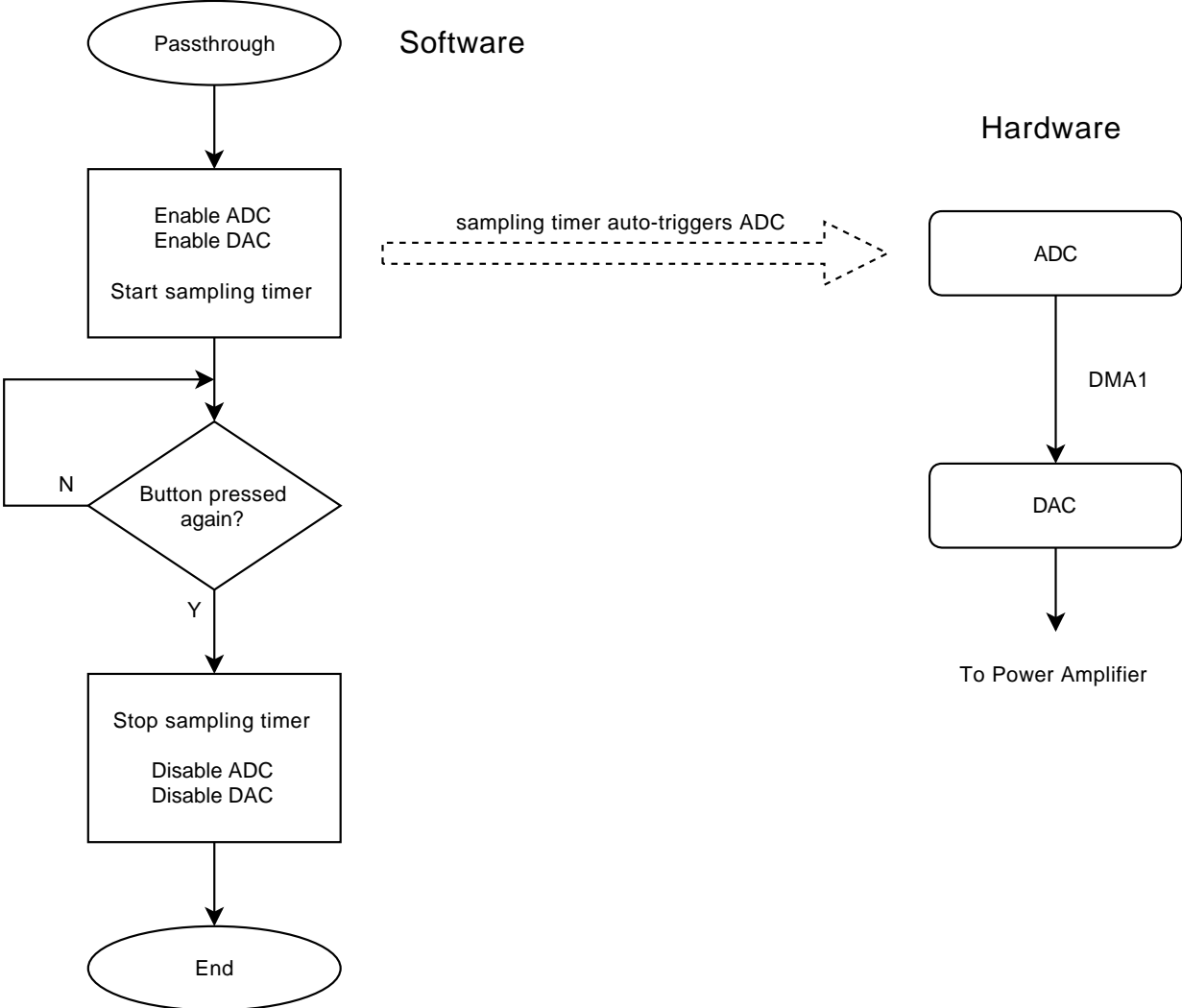


6.5 Passthrough Subroutine

The passthrough subroutine is shown in [Figure 6-6](#). The flowchart on the left indicates the functionality in software while the one on the right indicates hardware functionality. The passthrough subroutine consists of the following steps:

1. Once the program enters Passthrough mode, the ADC, DAC and sampling timer are turned on.
2. With each tick of the sampling timer, an ADC conversion is performed and the data passes directly through to the DAC via DMA.
3. The passthrough is stopped when the button is pressed again. To close the passthrough session, the sampling timer is stopped and the ADC and DAC are disabled.

Figure 6-6. Passthrough Subroutine



7. Hardware Implementation

For this application note, the following hardware is used:

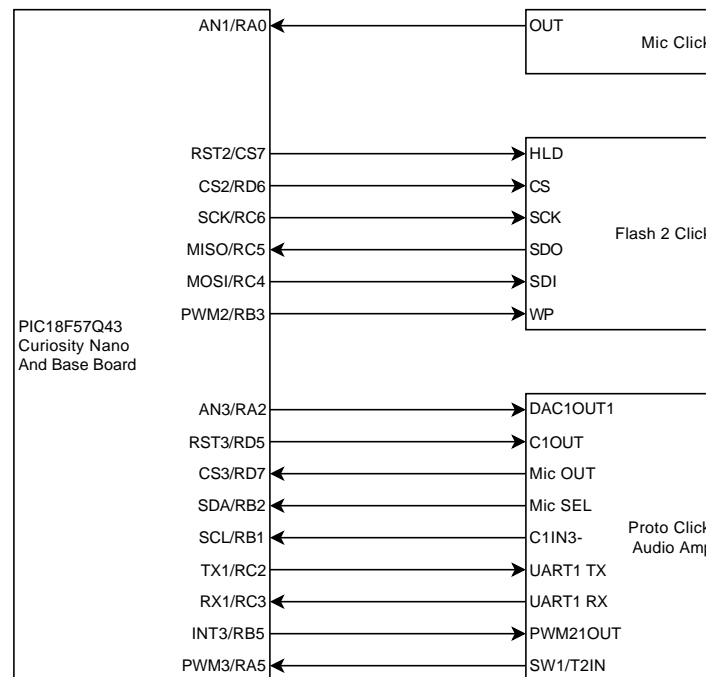
- PIC18F57Q43 Curiosity Nano Evaluation Kit (DM164150)
- Curiosity Nano Base for Click boards™ (AC164162)
- Mic Click Board™ (MIKROE-2563)
- Flash 2 Click Board™ (MIKROE-2267)
- Power amplifier implemented on a PROTO Click Board™ (MIKROE-1507)

7.1 Hardware Setup Using Curiosity Nano Evaluation Kit

The necessary connections between the PIC18F57Q43 Curiosity Nano board and the required click boards are shown in Figure 7-1. The full system setup is shown in Figure 7-7.

- The Click 1 expansion connects to the Mic Click Board™. This uses the AN1 pin to transfer analog voice signal to the ADC on-board the PIC18F57Q43. The Mic Click Board™ has a signal conditioning and filtering circuit on board.
- The Click 2 expansion connects to the Flash 2 Click Board™. This board has the SST26VF064B Flash memory module by Microchip and uses SPI to communicate with the microcontroller.
- The Click 3 expansion connects to a custom-made Class D power amplifier circuit on a PROTO Click Board™. An additional circuitry is included to support audio input from a headphone mic attached to the 3.5 mm audio connector.

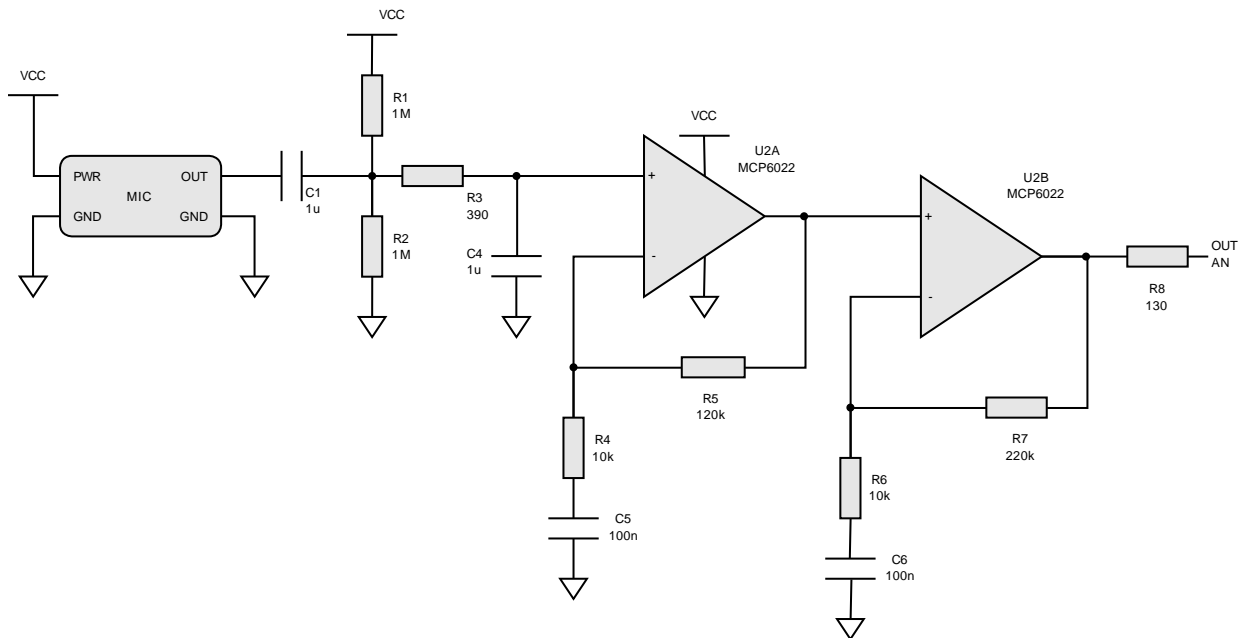
Figure 7-1. Curiosity Nano Board Connections



7.2 Mic Click Board™ and Microphone Circuit

The microphone circuit that comes with the Mic Click Board™ is modified to increase the signal gain and improve filtering. The modified microphone circuit is shown in Figure 7-2.

Figure 7-2. Mic Click Board™ – Modified Circuit



7.3 Power Amplifier

This application uses a custom-made Class D power amplifier (Figure 7-3) to connect an external speaker to the DAC output. A 300 kHz, 95% duty cycle PWM signal is used to generate a sawtooth waveform. This sawtooth waveform, along with the DC-offset DAC signal, is fed into a comparator to generate a PWM waveform. This PWM waveform passes through a MOSFET driver to drive an externally connected speaker. Refer to the [TC4426/TC4427/TC4428 data sheet](#) for more information about the MOSFET driver. This Class D power amplifier is implemented as a custom-made circuit on a PROTO Click Board™ (see Figure 7-5 and Figure 7-6).

In addition to the power amplifier, the PROTO Click Board™ also contains the following additional features:

- An additional switch SW1 to access Passthrough mode.
- Jumper J1 to select the audio input source between Mic Click and Headphone Mic (1, 2).
- Jumper J2 to select the audio output between external speaker and 3.5 mm audio connector. The 3.5 mm audio connector can be used to connect to an external speaker or a headphone.
- Signal conditioning circuit for headphone microphone input. See Figure 7-4.



Important:

1. The user is advised to select Mic Click as audio input source when a headphone microphone is not connected.
2. It is recommended to avoid selecting headphones for both audio output and mic input together in Passthrough mode to avoid noise induction.

Figure 7-3. Class D Power Amplifier

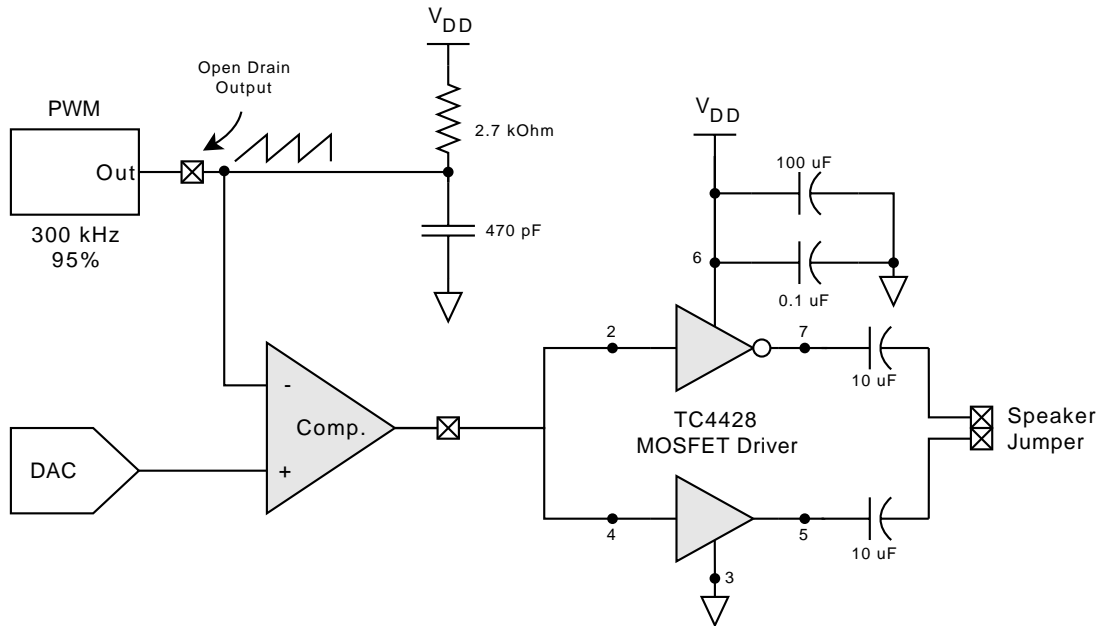


Figure 7-4. Headphone Mic Circuit

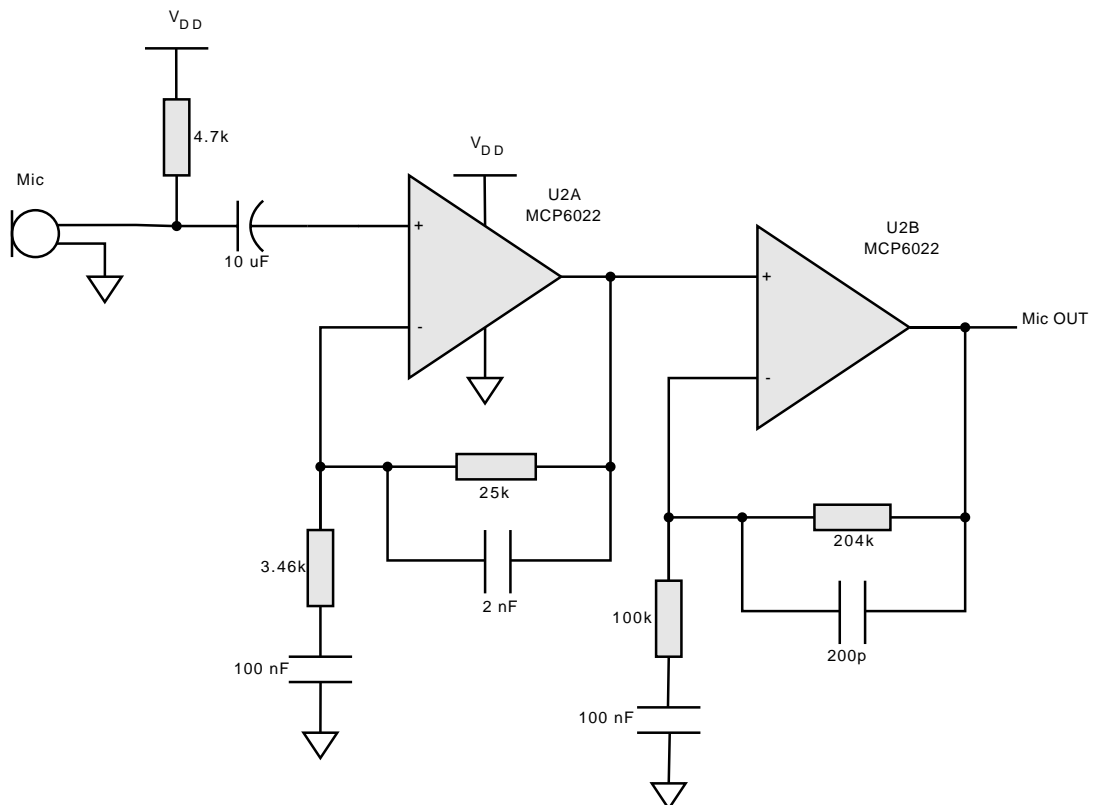


Figure 7-5. PROTO Click Board™ Circuit

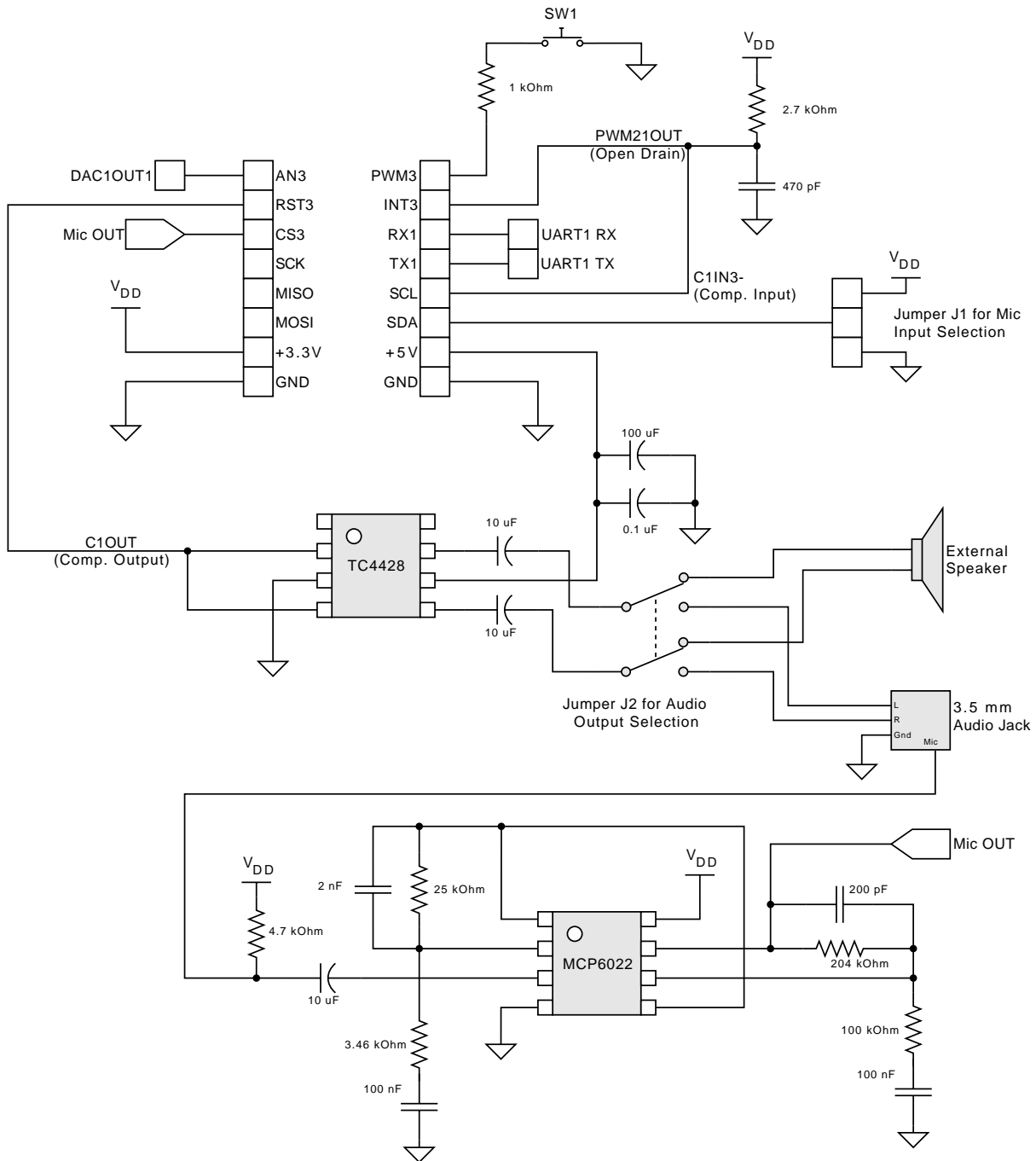


Figure 7-6. Custom-Made PROTO Click Board™

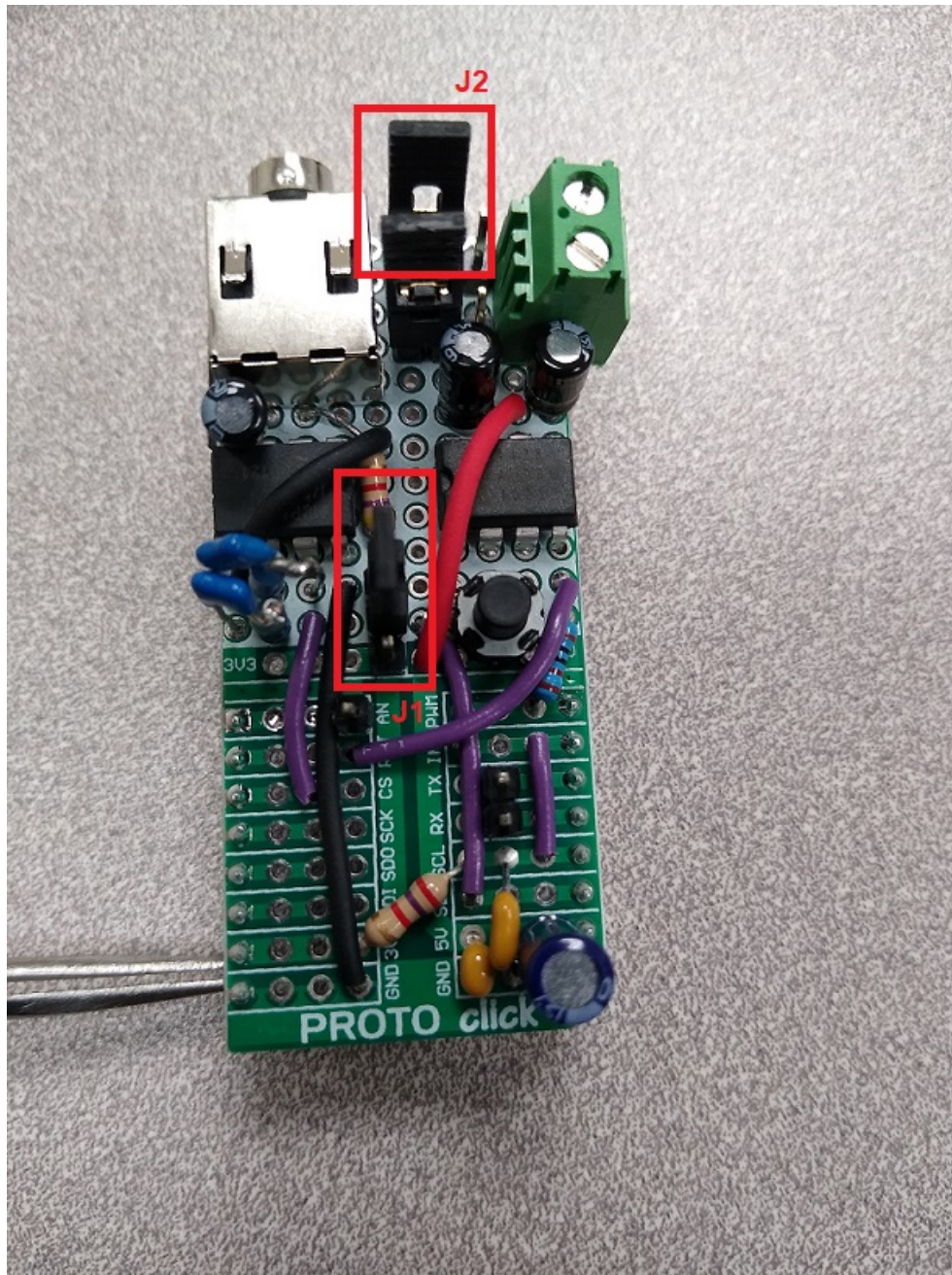
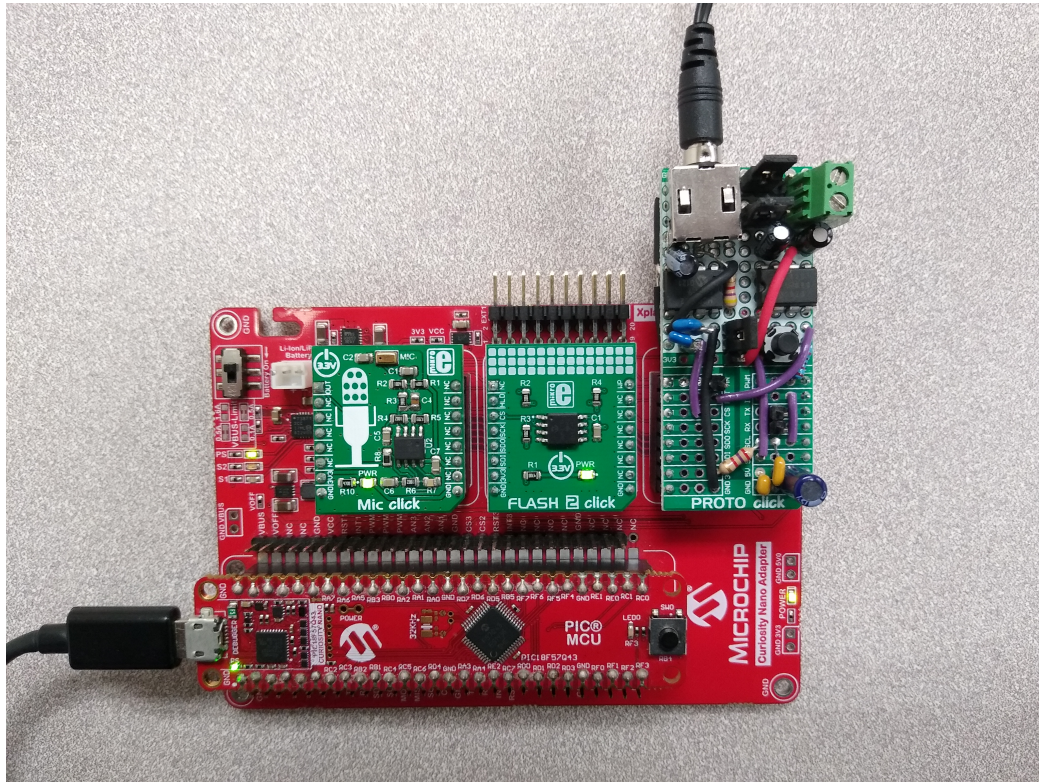


Figure 7-7. Full System Setup



8. Conclusion

Using the DMA is a great way to transfer data between SFRs, GPRs and PFM in different configurations. It allows for speed and ease of use, as well as for freeing up the CPU to perform other tasks.

The DAC allows improved reproduction of sampled audio signals because of the higher 8-bit resolution. The buffered output of the DAC forms a great connection to different peripherals directly and without degrading the signal. This can be used along with the PWM and Comparator to form an excellent Class D power amplifier.

Timer2 HLT can be combined with CLC for an excellent code-free switch debouncing technique.

9. References

Refer to the following for additional information:

- [Code Free Switch Debounce using TMR2 with HLT](#)
- [Application Note AN2547: Digital Sound Recorder Using DAC with the tinyAVR[®] 1-series \(DS00002547\)](#)

10. Revision History

Doc. Rev.	Date	Comments
B	10/2020	Updated sections 6.1, 7.1, 7.2, and 7.3. Updated figures 7-3, 7-5, and 7-7. Added figures 7-4 and 7-6. Corrected various typos.
A	07/2020	Initial document release.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2020, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-6822-6

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
<p>Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Tel: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com</p> <p>Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455</p> <p>Austin, TX Tel: 512-257-3370</p> <p>Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088</p> <p>Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075</p> <p>Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924</p> <p>Detroit Novi, MI Tel: 248-848-4000</p> <p>Houston, TX Tel: 281-894-5983</p> <p>Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380</p> <p>Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800</p> <p>Raleigh, NC Tel: 919-844-7510</p> <p>New York, NY Tel: 631-435-6000</p> <p>San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270</p> <p>Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078</p>	<p>Australia - Sydney Tel: 61-2-9868-6733</p> <p>China - Beijing Tel: 86-10-8569-7000</p> <p>China - Chengdu Tel: 86-28-8665-5511</p> <p>China - Chongqing Tel: 86-23-8980-9588</p> <p>China - Dongguan Tel: 86-769-8702-9880</p> <p>China - Guangzhou Tel: 86-20-8755-8029</p> <p>China - Hangzhou Tel: 86-571-8792-8115</p> <p>China - Hong Kong SAR Tel: 852-2943-5100</p> <p>China - Nanjing Tel: 86-25-8473-2460</p> <p>China - Qingdao Tel: 86-532-8502-7355</p> <p>China - Shanghai Tel: 86-21-3326-8000</p> <p>China - Shenyang Tel: 86-24-2334-2829</p> <p>China - Shenzhen Tel: 86-755-8864-2200</p> <p>China - Suzhou Tel: 86-186-6233-1526</p> <p>China - Wuhan Tel: 86-27-5980-5300</p> <p>China - Xian Tel: 86-29-8833-7252</p> <p>China - Xiamen Tel: 86-592-2388138</p> <p>China - Zhuhai Tel: 86-756-3210040</p>	<p>India - Bangalore Tel: 91-80-3090-4444</p> <p>India - New Delhi Tel: 91-11-4160-8631</p> <p>India - Pune Tel: 91-20-4121-0141</p> <p>Japan - Osaka Tel: 81-6-6152-7160</p> <p>Japan - Tokyo Tel: 81-3-6880-3770</p> <p>Korea - Daegu Tel: 82-53-744-4301</p> <p>Korea - Seoul Tel: 82-2-554-7200</p> <p>Malaysia - Kuala Lumpur Tel: 60-3-7651-7906</p> <p>Malaysia - Penang Tel: 60-4-227-8870</p> <p>Philippines - Manila Tel: 63-2-634-9065</p> <p>Singapore Tel: 65-6334-8870</p> <p>Taiwan - Hsin Chu Tel: 886-3-577-8366</p> <p>Taiwan - Kaohsiung Tel: 886-7-213-7830</p> <p>Taiwan - Taipei Tel: 886-2-2508-8600</p> <p>Thailand - Bangkok Tel: 66-2-694-1351</p> <p>Vietnam - Ho Chi Minh Tel: 84-28-5448-2100</p>	<p>Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393</p> <p>Denmark - Copenhagen Tel: 45-4485-5910 Fax: 45-4485-2829</p> <p>Finland - Espoo Tel: 358-9-4520-820</p> <p>France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79</p> <p>Germany - Garching Tel: 49-8931-9700</p> <p>Germany - Haan Tel: 49-2129-3766400</p> <p>Germany - Heilbronn Tel: 49-7131-72400</p> <p>Germany - Karlsruhe Tel: 49-721-625370</p> <p>Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44</p> <p>Germany - Rosenheim Tel: 49-8031-354-560</p> <p>Israel - Ra'anana Tel: 972-9-744-7705</p> <p>Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781</p> <p>Italy - Padova Tel: 39-049-7625286</p> <p>Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340</p> <p>Norway - Trondheim Tel: 47-72884388</p> <p>Poland - Warsaw Tel: 48-22-3325737</p> <p>Romania - Bucharest Tel: 40-21-407-87-50</p> <p>Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91</p> <p>Sweden - Gothenberg Tel: 46-31-704-60-40</p> <p>Sweden - Stockholm Tel: 46-8-5090-4654</p> <p>UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820</p>