Project Notes

**Words**

ISR = interrupt service routine

NVIC = nested vectored interrupt controller

EXTI = external interrupt/event controller

IRQ = an interrupt request

CSS = clock security system

AHB = advanced high performance bus

**Notes**

There are 15 standard exceptions for M7

Reset, NMI = high priority, Hard fault = generic software fault , memory management fault, bus fault, usage fault = trying to access illegal something, SVCCall = system interrupt for privileged access, debug monitor, PendSV lower priority system interrupt, SysTick uses a timer typically every 1ms.

We can have up to 240 interrupts in the M7

DMA

DMA controllers have two master ports, “peripheral” and “memory”. A slave port used by the processor to program it. Can have memory flow bi-directionally.

Only DMA2 can do memory to memory management

Certain channels are bound to certain DMA streams. Check which ones work for DAC and ADC. Loc 5267

MemInc: important for continuous memory saving. Memory increment.

Look at PeriphDataAlignment in the datasheet on DMA. May need this.

Mode: set to circular. Read up on it.

FIFOMode: allows burst mode. Could be good for using 3 ADCs. Loc 5303

Example using USART memory transfer from peripheral. Loc 5376

When using DMA in interrupt mode, it is important to set the other functions to null to avoid odd errors. Loc 5422

How to set up DMA interrupt mode. Loc 5422

See HAL documentation for using ADC and DAC in DMA mode.

Peripheral –to-peripheral is tough.

Example setup for peripheral to memory with circular. See loc 5490

Might want to use HAL\_DMAEx\_MultiBufferStart()

We may need to look at debouncing any user buttons to avoid transients.

**Clock Tree**

Words:

HSI = high speed internal clock source

HSE = high speed external clock source

LSE = low speed external

LSI = low speed internal

IWDT = Independent watchdog peripheral

RCC = reset and clock control

MSI = multispeed internal

MCO = master clock output

Notes:

There is a drift of 1% on the clock

This has phase-locked loops to help stabilize the clocks.

It may only be possible to reach maximum clock speed by using an external oscillator. Loc 5889

If we change the clock by hand, we need to call SystemCoreClockUpdate().

If the HSE causes a failure, the MCU will automatically switch to the HSI for safety.

**Timers**

Terms:

UEV = Update Event

CNT = counter register

EGR = event generator

UG = Update Generator

DWT = Data watchpoint and Tracing

CYCCNT = cycles performed by CPU

Notes

Basic, general purpose, and advanced timers

When the counter reaches a maximum number, it can set off an interrupt

Basic Timers only have 16 bits.

Basic timers are used with the DAC to trigger DMA request

General purpose timers can have 32 bits

Equation for calculating how often a timer overflows. Loc 6289

Must start the timer

Example of using timer loc 6345

The timer needs enough resolution to give an accurate distance measurement. That is a constraint. Each point must correspond to a reasonable amount of time.

Timers are not stopped when debugging unless \_\_HAL\_DBGMCU\_FREEZE\_TIMx() is called x = 1,2,3,... timer number. Also, \_\_HAL\_RCC\_DBGMCU\_CLK\_ENABLE() loc 8027

ADC

Terms:

SAR ADC = Successive Approximation Register ADC

Notes

For 12 bit, we have 2^12 – 1 = 4095

It can be configured to be driven by a clock with interrupts

Configuring DMA for ADC use instructions on loc 8794 continuous conversions in DMA mode

If the ADC register value is overwritten before being read, it can trigger an error interrupt which can be picked up with HAL\_ADC\_ErrorCallback()

The conversions may cease if they were not picked up. If we are reading three streams, we may run into this issue. See section 12.2.7 loc 8809 for handling this difficult to debug behavior.

We can configure the calibration of the ADC. It should be fine, but this can be done.

This should operate on single channel Continuous conversion

DAC

Terms:

Notes:

Vout = vref X D/2^n

The quality of the output signal is heavily dependent on the purity of the input analog power.

I can have an interrupt for when the dac conversion in DMA has finished. HAL\_DACEx\_ConvCpltCallbackChX() This could be used to set my reference time, or a timer that counts up to track how long the response will take.

Plans

Create routines that can be tested and shown to work individually.

Take a given PN code and convert it to an analog signal that is sampled at a given sampling frequency. Cause it to be sent based on a counter that is triggered after the user pushes the button. (we have two PN codes to send, one when user pushes the button, the second at x time after the peak was detected). Flash the light after the PN code is sent. If sent 3 times, flash 3 times.

Perform FFT convolution and detect a received signal.

FFT

To use the fft, include “arm\_math.h”

The memory is in Little Endian format. Pg 65/1671 on microchip long reference.

See here for documentation on the FFT and DSP functions: <https://www.keil.com/pack/doc/CMSIS/DSP/html/group__ComplexFFT.html>

<https://www.keil.com/pack/doc/CMSIS/DSP/html/group__ComplexFFT.html>

pins needed for board 001

Vdd

Gnd

MCO

BOOT0

STLK\_RX

STLK\_TX

TCK

Gnd

TMS

NRST

SWO

TDI