Beverly Yee

u0770041

Prelab 06

1. What is hysteresis and how does it help prevent bad behavior on digital inputs?

**A type of voltage thresholding, it changes the voltage threshold depending on the detected digital state.**

**It makes it impossible for a signal to consistently hang around the trigger point by moving the threshold up if the signal is in a low state or moving it down if the signal is high.**

1. What is quantization?

**The process of mapping a high-resolution signal to a manageable lower-resolution one. Basically flattens pieces of an input signal to make it representable in a certain number range.**

1. What does Nyquist theory explain? What is the problem with sampling a signal too slowly?

**Nyquist theory explains the relationship between how often an input signal should be sampled and whether or not the data is represented properly afterwards.**

**Sampling too slow will decrease the accuracy of the representation of the signal. The best way to sample a signal is to sample at a rate that is at least twice as fast as the fastest signal.**

1. The maximum resolution of the ADC is 12-bits. How many quantization steps/values does this give us?

**Since 8 bits equates to 256 steps, 12-bits is 4096**

1. What are the steps to perform an ADC calibration?

**Calibration can only be performed when the peripheral is stopped.**

1. **Ensure that ADEN = 0 and DMAEN = 0**
2. **Set ADCAL = 1**
3. **Wait until ADCAL = 0**
4. **The calibration factor can be read from bits 6:0 of ADC\_DR**
5. What’s the difference between right and left-aligned data in the DAC registers?

**The left is used for selecting the upper bits of a 16-bit number. The DAC can act on 16-bit data without any conversion or shifting this way.**

1. What DAC register would you use to write 8-bit to right-aligned data? (use the peripheral reference manual)

**DAC\_DHR8Rx[7:0] bits**

**DAC\_DHRx[11:4] bits**

1. Name something you found confusing or unclear in the lab manual. If everything was clear, simply answer that you didn’t have any issues.

**Following intuition is how I got my answer for #4 but am not entirely sure that is correct. Does the number of bits correlate to the number of quantization steps given?**

ADC Peripheral Registers

The ADC can output data in either 12-bit, 10-bit, 8-bit or 6-bit resolution. As the output resolution decreases, the amount of time required for each analog conversion drops. Because there are delays between starting a conversion cycle and getting an output, the ADC uses status flags throughout all its registers to notify the user when it is ready to start a new conversion, when it is busy, what input in a conversion sequence is currently active and more.

ADC Interrupt and Status Register (ADC\_ISR)

The interrupt and status register contains interrupt flags as well as a few high-level status bit which indicate when the ADC is ready for use.

ADC Control Register (ADC\_CR)

The ADC control register is used to startup, calibrate, and shutdown the ADC peripheral. To minimize errors due to voltage offset and drift, it is important that you allow the ADC to perform a self-calibration process every time you first turn it on. Additionally the ADC has a warm-up period after you enable it before you can start a conversion process.

! One common source of confusion when first using the ADC is that many of the bits in the control register are not only used to control the peripheral, but are also modified by hardware as status flags.

An example of this is the self-calibration. Once the user triggers a calibration cycle by setting the appropriate bit, the ADC will clear that same bit when calibration is complete. Most of the bits in the control register have strict requirements for when the ADC will allow you to set them. You will need to read the bit descriptions to know what you are allowed to configure before/after each step.

ADC Configuration Register 1 (ADC\_CFGR1)

The configuration register sets the operational mode, data resolution, and trigger source of the ADC.

ADC Channel Selection Register (ADC\_CHSELR)

Each input to the ADC has a conversion channel associated with it. The channel selection register controls which of these channels are read by the ADC when operating. You will need to enable the channel associated with the input pin you choose in the lab assignment.

ADC Data Register (ADC\_DR)

The data register holds the most recently converted value. You will read the quantized value of the

ADC input pin from this location.

DAC Peripheral Registers

In contrast to the ADC, the digital-to-analog converter is one of the simplest peripherals to configure on the STM32F0. It can have either one or two output channels (the STM32F072 has two) with either 8 or

12-bit operation.

DAC Control Register (DAC\_CR)

The DAC control register selects the trigger source and enables/disables the output channels. The trigger source of the DAC determines whether software (writing to a data holding register) causes the DAC to update its output, or if the peripheral waits on a signal from a peripheral such as a timer. Data Holding Registers

The DAC has three different types of data holding registers used depending on the desired data format. When triggered, the DAC moves the data from the appropriate holding register into the read-only output registers.

The three possible operation modes are:

• DAC\_DHR8Rx – 8-bit right-aligned (data in bits 0-7) • DAC\_DHR12Rx – 12-bit right-aligned (data in bits 0-11) • DAC\_DHR12Lx – 12-bit left-aligned (data in bits 15-4)

The left-aligned mode is typically used for selecting the upper bits of a 16-bit number, allowing the DAC to act on 16-bit data without any conversion or shifting. (with some minor loss in precision provided by low-order bits) You can read more about the DAC’s data formats in section 14.5 of the

peripheral reference manual.

Lab Assignment

1. Initialize the LED pins to output.
2. Select a GPIO pin to use as the ADC output.
   1. Look in the Discovery datasheet and find a pin that has “ADC\_INx” in their additional/analog function column.
      1. PA1-7, PC4-5, PB0-1
   2. Configure the pin to analog mode, no pull-up/down resistors.
   3. Connect the center pin (output) of a potentiometer to the input pin.
   4. Connect the other two pins of the potentiometer to 3V and GND.
3. Enable the ADC1 in the RCC peripheral.
4. Configure the ADC to 8-bit resolution, continuous conversion mode, hardware triggers disabled (software trigger only).
5. Select/enable the input pin’s channel for ADC conversion.
6. Perform a self-calibration, enable, and start the ADC.
   1. The lab manual describes the basic procedure without mentioning the actual flags and conditions for advancement.
   2. Use sections 13.4.1 (Calibration) and 13.4.2 (ADC on-off control) in the peripheral reference manual.
7. In the main application loop, read the ADC data register and turn on/off LEDs depending on the value.
   1. Use four increasing threshold values, each LED should have a minimum ADC value/voltage to turn on.
   2. As the voltage on the input pin increases, the LEDs should light one-by-one.
   3. If the pin voltage decreases below the threshold for a LED, it should turn off.
8. Select a GPIO pin to use as the DAC output.
   1. APB bus, 0x40007400 – 0x400077FF

Initialize the ADC

1. Set the desired operating mode, data resolution, and trigger source.

• Single conversion mode performs one measurement every time the ADC is triggered. Continuous conversion mode repeatedly measures and updates the data register.

• The trigger source selects whether software or a hardware signal starts a new conversion

process.

2. Start the ADC calibration

• Calibration can only be performed when the peripheral is stopped, don’t set any enable/start

bits (other than in the RCC peripheral) before attempting to start a calibration process.

3. Wait for the hardware to signal that the calibration has completed.

• The ADC will clear the calibration start/request bit when the process has completed. 4. Set the peripheral enable.

5. Wait until the ADC ready flag is set. 6. Start the ADC conversion.

• Starting a conversion sequence is separate from enabling the peripheral. • Since the lab assignment uses continuous conversion mode, this bit only needs to be set

once at the beginning.

Initialize the DAC

1. Set the trigger source for the channel/output update. 2. Enable the channel used for output.

Beverly Yee

u0770041

Postlab Questions

1. Consider a system where the DAC is updated every 4us (250 kHz) with a value from a 200element wave table containing a single cycle of a waveform. What would be the frequency of the output wave?
2. Consider that the ADC in 12-bit mode divides the input voltage range (0-3V) into 4096 steps (where 0V is 0, and 3V is 4095).
   1. What is the voltage/measurement resolution (how much does the voltage change per bit) of the ADC?
   2. What would be the ADC output value (nearest integer) if the input voltage was 1.75V?