CE-2812, Lab Week 8, YATA (Yet Another Timer Application) – Analog Signal Capture

1 PURPOSE

The purpose of this lab is to explore another peripheral (DAC) and optionally DMA.

2 PREREQUISITES

- The Nucleo-F446RE board had been mounted onto the Computer Engineering Development Board.

3 ACTIVITIES

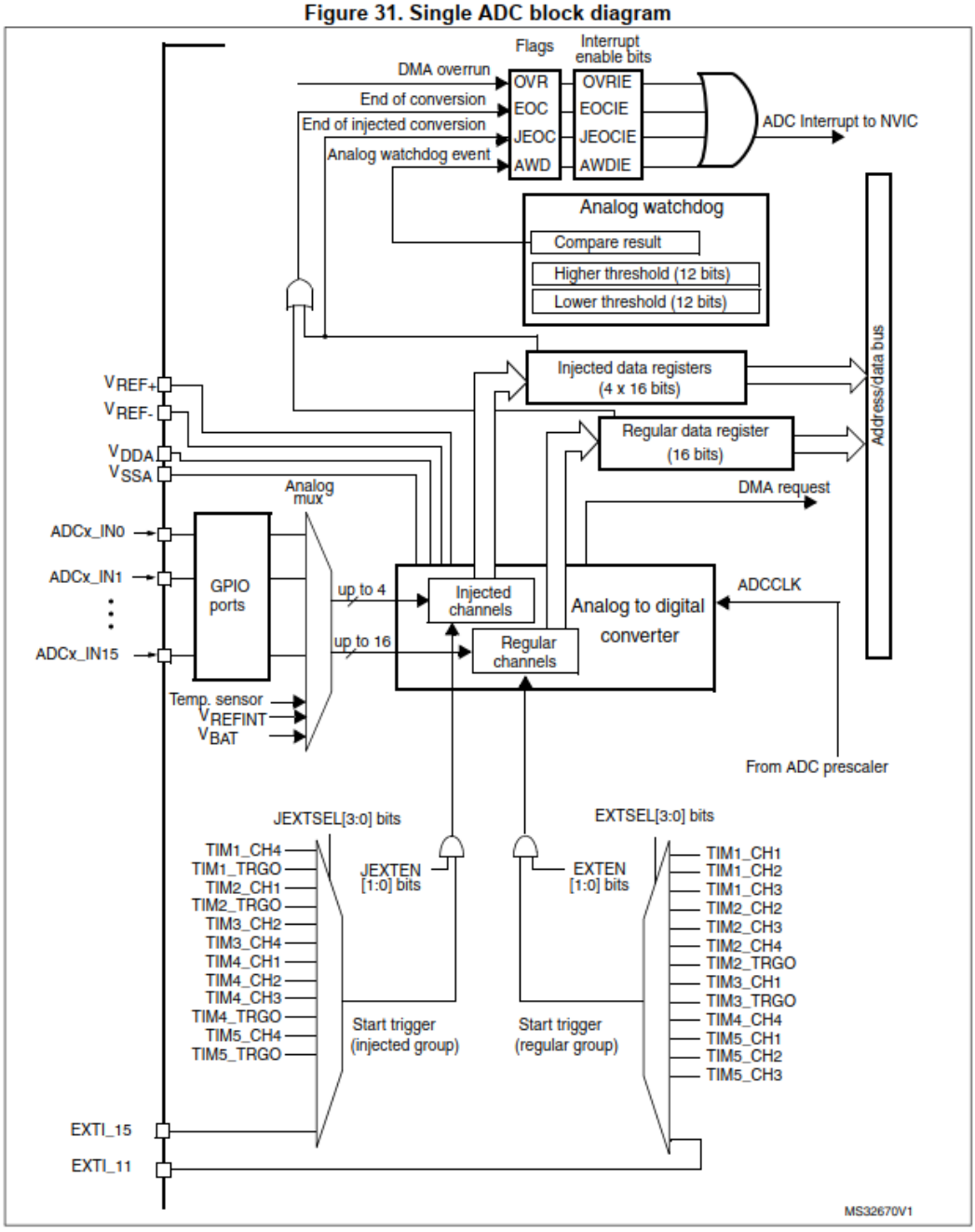
3.1 BACKGROUND

3.1.1 Analog to Digital Conversion

An analog to digital converter is a very common peripheral in the world embedded computing. Since embedded computer often interface to the real world, ADCs provide a means to collect information from that world through sensors. Our microcontroller, the STMF411 has a single converter, but also includes a multiplexor so that it can actually connect to a bunch of sensors; although, it can only convert one channel at a time.

Conversion takes time and is paced by hardware. So, there are a variety of ways to interact with the ADC. You can set it up to convert continuously and just go grab the latest measurement when you want it, you can invoke a single conversion and poll a status register to know when it is finished then grab the result, or, an interrupt can be generated when conversion is finished.

Figure 31. Single ADC block diagram



In the figure above, taken from the reference manual, you can see the analog mux on left side which can connect to a variety of GPIO pins (when in analog mode), along with an internal temperature sensor and a couple of internal voltages.

At the bottom of the diagram you can see a bunch of additional connections from mostly timers. These are for triggers. This makes sense as if you are digitizing a waveform you would want those samples to be timed very regularly.

3.2 General Requirements

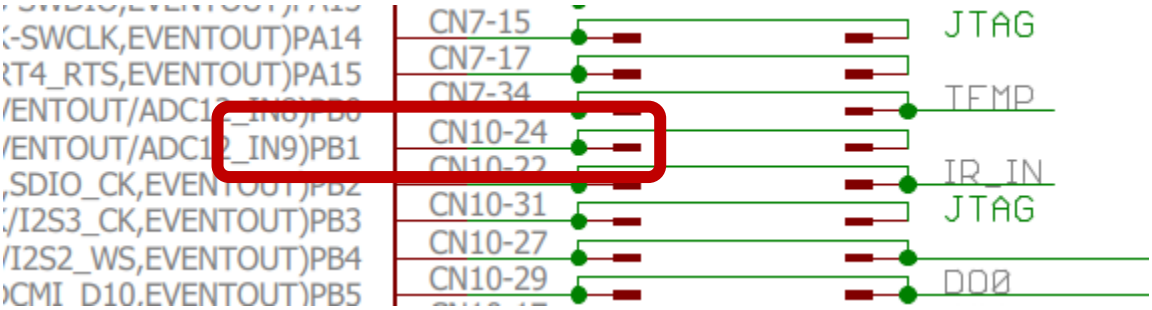
The goal of this application will be to digitize a waveform and present the data to the console from which the text can be captured and pasted into Excel for plotting.

This application will be added to your existing console program. It will operate in sort of a hybrid mode where the data collection will happen in the background via interrupts and there will have to be a second command to retrieve the collected waveform when collection is finished.

- Add option to "digitize waveform." This command will take arguments to set sampling rate and number of samples to be collected. Ideally, this will be in command form as illustrated below.
- Once invoked, the "digitize waveform" function should:
  - Use malloc() to dynamically allocate memory to store the requested number of the samples
  - Configure ADC
  - Configure a timer to pace analog conversion
  - Allow ISR to collect samples and place in dynamic memory
  - When ISR is done collecting data, a flag should be set so subsequent command know collection is finished
- You should be able to invoke other menu commands while signal collection continues in background. This should include background music and measuring frequencies as well as the original memory r/w/dump commands.
- You will need a second command that retrieves the samples and prints them to the console. A recommended format is shown below. You should free the dynamic memory after printing the collection to the console.
- Handling dynamic memory - you will need to allocate (malloc) memory when you start a collection and then free that memory after the samples are printed. What if you start a second collection without printing the previous collection? You may need a couple of additional "flag" variables to determine if memory has been allocated or freed and if your system is ready for collection or not.

3.3 Hardware Considerations

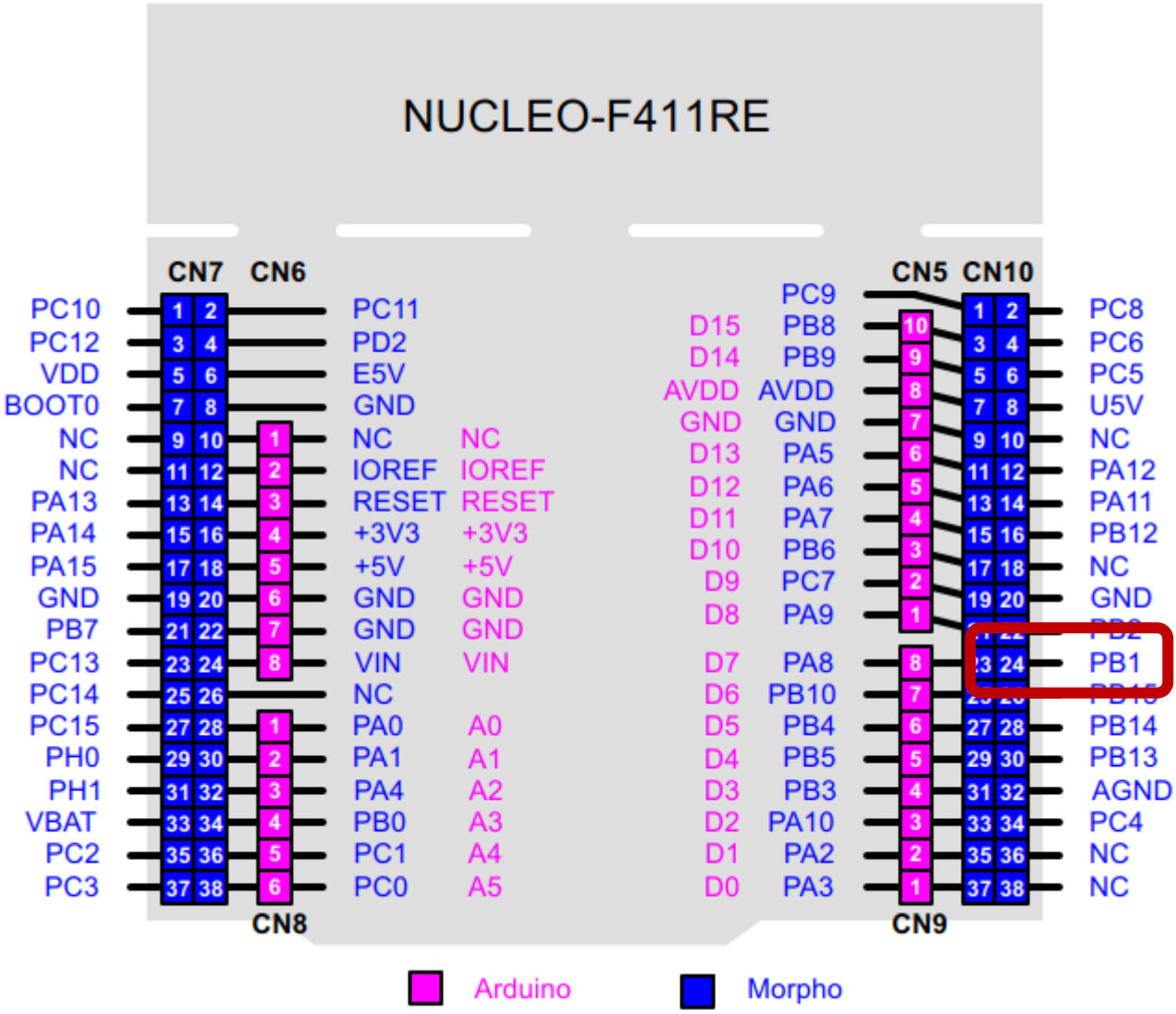
Many, but not all, I/O pins can connect to the ADC. One caveat, the dev board schematic depicts the F446 which actually has three ADC units. So, we will need to refer to the F411 datasheet as well. We would want to avoid other onboard peripherals. It looks like PB1 will be a good choice.



A glance at the datasheet confirms that PB1 is Channel 9 on the ADC:

PB1	I/O	FT	-	TIM1_CH3N, TIM3_CH4, SPI5_NSS/I2S5_WS, EVENTOUT	ADC1_9

This will be accessible on CN10, pin 24.

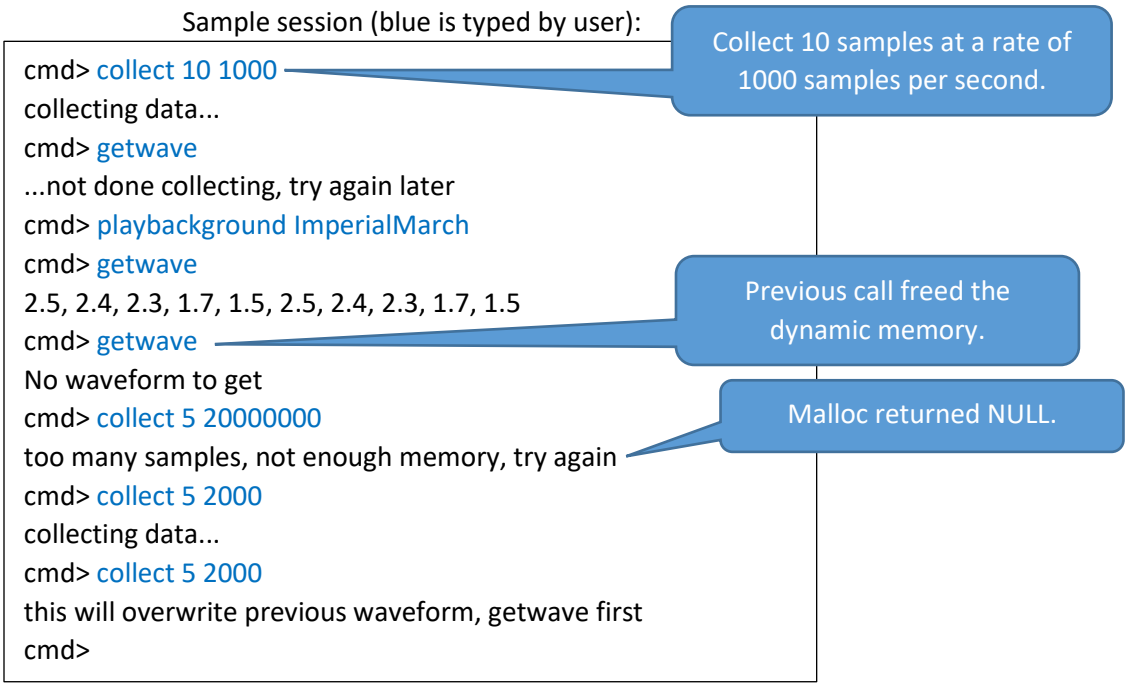


You will need to connect an Analog Discovery to generate a signal. Since we are sampling regularly, you should be able to digitize sine waves, triangle waves, square waves, etc. You could also sample complex waveforms. Do note, however, that the input range will be positive voltages only and in the range of the board's power supply, so 0 to 3.3 volts max only.

To use the ADC, you will need to "connect" PB1 to the ADC, configure the ADC's collection mode, configure a timer to invoke regular sampling, and use an ISR to retrieve the samples and store into [dynamic] memory. Exact details of each of these steps will be provided in lecture.

3.4 Command Structure

- The command issued to generate a signal should be a single line command and not a series of prompts.
- You need to support a range of sampling rates. Perhaps a good range would be on the order of 10 samples per second to 10,000 samples per second.
- While you might consider a minimum number of samples (maybe 10 or 100), the upper range should be dictated by available memory. If a user requests too many samples, malloc will fail (return NULL) and you should refuse to complete the collection.
- The ADC has a configurable resolution. The maximum resolution is 12-bits. It would be silly to use a 32-bit integer to store a 12-bit number. Make sure you are efficient with memory and choose the best type for your storage.
- Per the previous point, it makes the most sense to store the samples as an integer, however, when you print to console, you should print as a voltage. This will require converting the integer "code" to a voltage.



4 DELIVERABLES

When completed:

1. Submit to Canvas a **single pdf** printout of your completed source code to Canvas. **Include in a comment block at the top of your code a summary of your experience with this project.**
2. Ask to demo your lab to instructor. You can do this via writing your name on the whiteboard.
    a. If you demo during lab in Week 8, you will earn a 10% bonus on this lab.
    b. If you demo during lab in Week 9, you will be eligible for full credit.
    c. You demo must be made with a signal generator (Analog Discovery) and the collected waveform must be plotted in Excel (or Google Sheets). You will need to have this prepared prior to calling instructor over for demo.

- Demos are ONLY accepted during lab periods. If you are unable to demo by the end of lab in Week 8, you lose the 10% of the assignment attributed to the demo (per syllabus).

- Demos must be ready a reasonable amount of time before the end of the lab period. If you write your name on the board at 9:45 and lab ends at 9:50, and there are five names in front of yours, you will be unlikely to complete your demo by the end of lab and hence lose the bonus or demo points.

4.1 GRADING CRITERIA

For full credit, your solution must:

- Minor errors usually result in a deduction of ~ 3 points (three such errors results in ~ a letter grade reduction)
- Major errors, such as not achieving a requirement, usually result in a deduction of 5 to 10 points.