2DP4 MICROPROCESSOR PROJECT: DATA ACQUISITION AND DISPLAY

1. Introduction

At some point in an engineer's career they will need to acquire data. There are numerous systems available to do this, but they can be expensive, limited in capability, and/or too complex for your requirements. As a result, data acquisition becomes too complex or expensive. Ultimately, engineers seek to measure physical phenomenon accurately at as low a cost possible.

The experience with this project will also give the student insight into how the commercial/industrial/medical data acquisition systems operate.

One of the primary objectives of 2DP4 is that the student leave with the capability of collecting analog data using the microcontroller and then be able to process and communicate that data. This ability should be directly applicable to most senior capstone projects, and certainly beyond.

2. Overview

Regardless of the specific analog signal to be measured, the interface process remains the same:

: **Quantify the analog signal** - range of amplitude, frequency, source, impedance [continuous signal].

: **Build/Select the appropriate transducer** - pressure, sound, temperature, etc.

: **Precondition signal** - amplification, filtering, and/or level shift to conform to ADC design.

: **Analog-to-Digital Conversion (ADC)** - determine voltage range (min, max), resolution, sampling frequency [discrete data].

: **Data processing** - read data from ADC and store/process/transmit under time constrains to return for next ADC.

: **Control/Communicate** - micro controllers are implemented with specific purpose for acquired data, implement an algorithm that meets the objective with hardware and timing constraints.

3. Technical Requirements

The overall design of an interface project is well described by Huang when introducing the Analog-to-Digital Converter module, illustrated in figure 2.

The technical requirements of this project are to build and embedded system as a single channel data acquisition system for various alternating signals with the following properties:

(1) Analog signal: AC centred about 0V (+/- 2V).

(2) Transducer: Measured across a fixed resistance

(3) Precondition: Amplify/buffer and/or level shift to ADC requirements

(4) ADC: 1 channel (student to determine voltage range, resolution, sampling frequency)

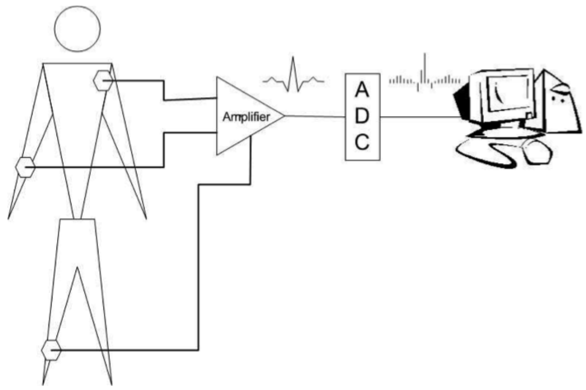

FIGURE 1. Example data acquisitions system

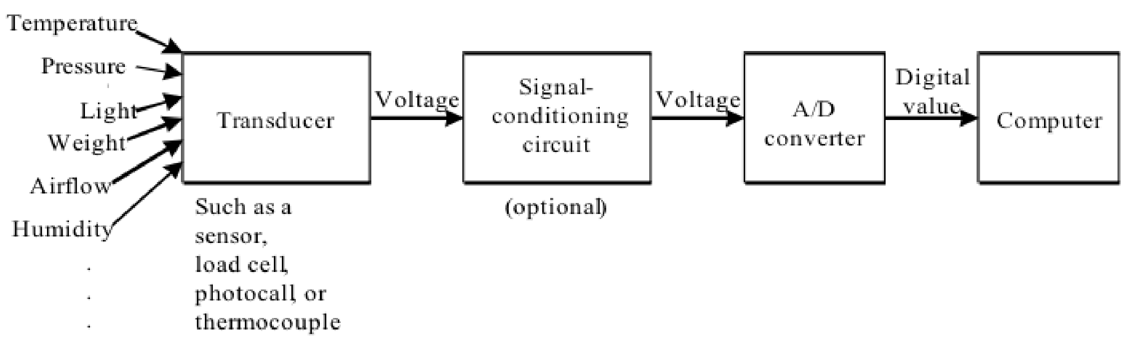

FIGURE 2. The ADC conversion process (from Huang)

(5) Data processing: polling or interrupt design

(6) Control/Communicate: serially transmit (built-in FTDI USB interface) data to PC application to graphically view data (Student choice: Labview, Matlab, Java, etc.)

3.1. **Core Components.** The project must sample analog signals, condition the signal for input to the ESDX's analog-to-digital converter module, encode analog signal to digital, process the data, serially transmit the data to a PC, and graphically display this data. Similar to the display on an oscilloscope, the amplitude axis must be labelled with the measured value and the time axis should have an appropriate time scale. The student must be able to start and stop data acquisition using either a physical button or a software command from the PC. Presented data must be updated periodically.

3.1.1. *Transducer.* You will be implementing a transducer to convert various electrical signals into signals suitable for your data acquisition system (e.g., an electret microphone would convert sound into an electrical signal, a thermistor converts temperature into an electrical signal, a photodiode converts light to an electrical signal, etc.). While transducers are typically a device that converts a physical phenomenon into an electrical signal, we will be using a resistor arrangement as our transducer to measure the changing voltage signal.

3.1.2. *Precondition.* While your textbook refers to the signal conditioning stage as optional, in practice it is normally not optional. Pre/conditioning is the stage where in the transducer signal is buffered, amplified, filtered, and/or level-shifted. It is common to have multiple conditioning stages for more complex signals and/or filtering.

For this project, buffering/amplification and level shifting stages are not necessary if you use the signal generator properly. However, you may design your own buffering/amplification and level shifting stages, research existing circuits (with proper attribution), or you may choose to use/adapt the circuit presented in figure 3. The circuit is from Horowitz and Hill, *The Art of Electronics, 2ed.* and described as:

> Figure [3] shows a typical single-supply noninverting amplifier to amplify an input signal of known positive polarity. The input, output, and positive supply are all referenced to ground, which is the negative supply voltage for the op-amp. The output "pulldown" resistor may be needed with type-1 amplifiers to ensure output swing all the way to ground; the feedback network or load itself could perform this function. An important point: remember that the output cannot go negative; this you cannot use this amplifier with, say, ac-coupled audio signals.
> Single-supply op-amps are indispensable in battery-operated equipment.

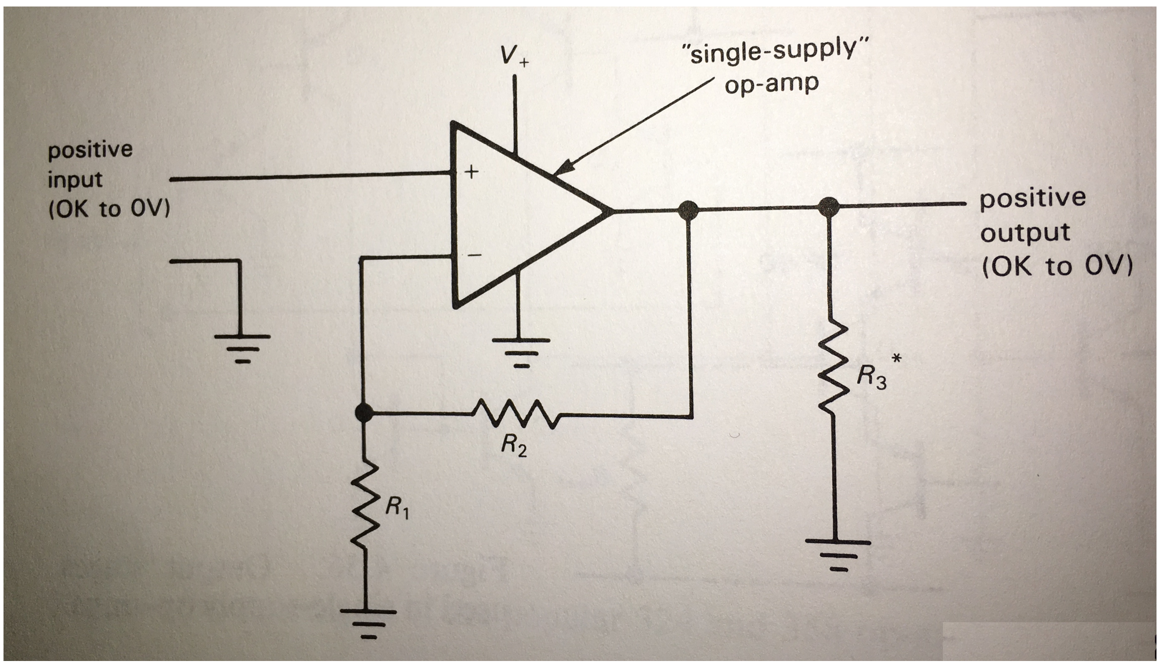

FIGURE 3. A typical single-supply noninverting amplifier circuit (0-5V)

3.1.3. *Analog to Digital Conversion.* Many embedded systems need to deal with realtime analog phenomena: weight, humidity, pressure, weight, mass or airflow, temperature, light intensity, and speed. Analog quantities must be converted into digital format so that they can be processed by the computer. An ADC can only deal with electric voltage. Physical phenomena are converted to electrical signals using a transducer. The signal from the transducer is then conditioned to make it suitable for the ADC. The ADC converts the analog signal into a digital representation that can then be digitally processed or communicated.

3.1.4. *Serial Communication.* We discussed the benefits of serial communication in lecture; however, your textbook (Huang) presents a good summary, which I have reproduced here:

> The need to exchange data between the MCU and peripheral devices can be satisfied by using parallel data transfer (multiple bits in one transfer operation). However, there are a few drawbacks.
>
> - Parallel data transfer requires many I/O pins. This requirement prevents the microcontroller from interfacing with as many devices as desired in the application.

- Many I/O devices do not have a high enough data rate to justify the use of parallel data transfer.
- Data synchronization for parallel transfer is difficult to achieve over a long distance. This requirement is one of the reasons that data communications always use serial transfer.
- Higher cost.

The serial communication interface (SCI) was designed to transfer data in an asynchronous mode that utilizes the industrial standard TIA-232 protocol. The TIA-232 was originally called RS-232 because it was a recommended standard. You have been using this interface to communicate with and download programs onto the demo board for execution. Only two wires are used by the SCI function.

The ESDX is equipped with a built-in FTDI USB interface + microB USB connector. When appropriately configured this will serve as the serial communication link between your ESDX and a USB-enabled peripheral, such as your personal computer. The benefit is a simplified communication interface using the now common USB. Not so long ago this was all done using RS-232, level shifting, etc.

You have been provided serial communication code for lab and this project.

3.1.5. *Start/Stop.* To facilitate data collection, the project must offer a start capability. You may choose to do this using a push-button or through a bidirectional software interface.

Due to a difference in instruction between TA and Prof, the Stop functionality is not required.

3.1.6. *PC Data Display.* The student is free to choose the application for display. It is suggested that Matlab, LabView, Python or Java would be good choices. Matlab and Tabview are available on lab computers. This is expected to be automatically graphed – no user intervention.

3.2. **Student Specific Requirements.** Based upon student number each will be assigned different operational parameters. The example student number 1234567 (ABCDEFG) will be used to illustrate each assigned parameter (written as a string of letters to refer to digit position instead of digit value).

3.2.1. *ADC Channel.*

Least Significant Digit	Assigned Channel
0	AN3
1	AN4
2	AN5
3	AN6
4	AN3
5	AN4
6	AN5
7	AN6
8	AN3
9	AN4

Thus for our example student, her LSD is 7 (position G), so her ADC channel is 6.

3.2.2. *ADC Resolution.*

Least Significant Digit + 1	Assigned Resolution
0	10
1	12
2	8
3	10
4	12
5	8
6	10
7	12
8	8
9	10

Thus for our example student, her LSD+1 is 6 (position F), so her ADC resolution is 10-bits.

3.2.3. *Bus Speed.*

Least Significant Digit+2	Assigned Bus Speed (MHz)
0	2
1	4
2	6
3	2
4	4
5	6
6	2
7	4
8	6
9	2

Thus for our example student, her LSD+2 is 5 (position E), so her bus speed 6 MHz.

3.2.4. *Sampling Frequency.*

Least Significant Digit+3	Assigned Sampling Frequency (Hz)
1	420
2	300
3	500
4	440
5	360
6	560
7	480
8	340
9	320
0	400

Thus for our example student, her LSD+3 is 4 (position D), so her bus speed 440 Hz.

Students should start with a fixed DC input (0 Hz) and then attempt sinusoid, square, and sawtooth signals.

You are required to sample at your specified rate. If you cannot sample exactly at the assigned value, it must be clearly documented why with calculations demonstrating the reason(s).

4. Questions

Address the following questions in your discussion:

(1) Calculate your maximum quantization error.

(2) Based upon your assigned bus speed, what is the maximum standard serial communication rate you can implement.

(3) Reviewing the entire system, which element is the primary limitation on speed? How did you test this?

(4) Based upon the Nyquist Rate, what is the maximum frequency of the analog signal you can effectively reproduce? What happens when your input signal exceeds this frequency?

(5) Are input signals with sharp transitions (e.g., square, sawtooth, etc.) accurately reproduced? Justify your answer.

5. Report Format

Your report should be structured as outlined here:

(1) Title Page

(2) Table of Contents

(3) Introduction & Background (include a real-life example of ADC)

(4) Design Methodology

  (a) Quantify Signal Properties

  (b) Transducer

  (c) Precondition/Amplification/Buffer

  (d) ADC

  (e) Data Processing (Algorithm Flowchart)

  (f) Control/Communicate (Algorithm Flowchart)

  (g) Full System Block Diagram

  (h) Full System Circuit Schematic

(5) Results (include data/images representing the testing of each of the 4b-4f + entire system)

(6) Discussion (answer questions)

(7) Conclusion

(8) Appendix of neatly formatted code

In addition, a .zip file of all code (commented and templated). Use the .zip compression. Test your compressed file, if it cannot be opened, we will not accept another submission.

6. Bonus 1

After meeting all above core design requirements assigned to the student, a 10% will be given to any student that implements all core objectives on the microcontroller using **solely an interrupt based approach**. Students are cautioned to ensure they meet the above requirements first, and then focus on bonus marks.

7. Bonus 2

After meeting all above core design requirements AND Bonus 1, a 10% will be given to any student that implements all core objectives on the microcontroller with the addition of wireless transmission of sampled data to an alternate display. This bonus will require the student to purchase a wireless module or potentially use the module planned for lab 5.

8. Alternate Projects

Any student requesting a special project must book an interview with the instructor prior to March 7 to discuss. Such requests must make a written submission on how their own project meets these technical requirements and be interviewed on their knowledge of the material. Making such a request comes with the acknowledgement that the project will be graded against the same rubric as the rest of the class.

9. Submission Requirements

Each student will be interviewed and be required to demonstrate their final project (schedule to be posted). Some projects will be retained for further inspection. Interviews may be recorded by audio, video, and/or photo.

Any project that is not accompanied by a complete final report will be assigned a 0 for the entire project.

Bring the following items to the demonstration/interview:

(1) Printed copy - full algorithm flowchart

(2) Software copy - Technical report (same structure as lab reports)

(3) Printed copy - Brief user's guide (1-page maximum, can be included as section Technical report)

(4) Functioning prototype - fully working prototype (be prepared to leave your prototype for further testing)

Students should expect to be asked specific questions about their design choices and their implementation of the project. If you cannot answer or defend a design choice then you will not be awarded the grades for it. You should be prepared to have your prototype recorded via video or photograph(s). Some prototypes will be retained for a short period for further review.

Submit the following two items to Avenue (separately) at the end of the demonstration/interview:

(1) Software copy - Technical report (same structure as lab reports), that includes full algorithm flowchart and brief user's guide. Must be in PDF format.

(2) Zipped file of all code (commented and templated). Must use the .zip compression. Test your compressed file, if it cannot be opened by us, we will not accept another submission.

Any student choosing to not submit the a complete report or full set of code will be considered a late submission. Once penalties accumulate to 100% of the entire project the work cannot be accepted for evaluation. Similarly, an interview/demonstration that is MSAFed will be rescheduled; however, given this project will have been posted for several weeks, the associated report and files must be submitted by the end of the originally scheduled interview. The final grade in the course will be INC until the interview is completed. Should a rescheduled interview be missed, the entire project will be assigned a 0. You must be able to demonstrate and defend your project design AND submit all files (report and code) for our review.

10. Changelog

The following changes were recorded:

(1) Released February 2, 2017