Implementing A Low Cost Data Acquisition System for Engineering Education Programs in Universities

Silviu REI
“Lucian Blaga” University, Sibiu, Romania
silviu.rei@ulbsibiu.ro

ABSTRACT

For many educational engineering projects running in Universities, a data acquisition system is a key component. Buying a system from the market is a viable option but many times the cost of it is prohibitive for the low budgets of University projects. This paper investigates the idea of implementing an own data acquisition system with use in small educational projects. Together with the advantage of cost reduction, an own implementation is an engineering exercise in itself, providing learning opportunities for the student. The paper presents first the results of a market scan for development platforms which can be used and then an example implementation of a simple data acquisition system using a low cost development platform. The performance of the implemented system is measured and presented. The main purpose of this paper is to encourage students and young researchers to implement their own tools, using this learning experiences along the way of meeting the project targets.

Keywords: data acquisition, education, engineering, market scan.

1. INTRODUCTION

In modern sciences and engineering, data acquisition is a fundamental and powerful tool. Many activities involve recording of time series representing evolution of various key parameters for a process or system, from temperature, humidity, forces affecting a certain material in a mechanical system, voltages in an electrical system, heartbeats per minute, number of cars passing a checkpoint and the list of examples can continue, being limited only by imagination. Capturing and recording the key parameters is done using data acquisition systems, which basically have the role of transforming the interesting parameters into digital time series, which are then available for processing with various algorithms on various computing platforms. Modern technology includes such data acquisition systems in many of the devices we use daily, from the smartphones, coffee machines, refrigerators or ovens, to cars, trains or airplanes. Even the nowadays common smartphone contains already a complete set of sensors: a gyroscope, compass, barometer, accelerometer, Hall Effect sensor, ambient light sensor, GPS, and so on. Some mobile application developers already developed tools to use all these sensors for recording data from the smartphone, allowing data export into real projects (Murphy, 2016).

The market offers various options for buying such systems. As example, National Instruments, a well established vendor of data acquisition systems, offers such systems in the price range starting at approx. 100 EUR for an 8 channel, 10kS/s/c, 12 bit system and reaching around 6000 EUR for and 8 channel, 2Ms/s/c, 16 bit system (National Instruments, 2017)

This paper will focus on investigating the options one has when trying to develop and implement on his own a data acquisition system, instead of purchasing one. In the first step we will briefly describe the key parameters we used for evaluating the performance of a data acquisition system. As a second step, we performed a local market scan for development platforms to be used as starting point for the implementation. The third part presents an example of implementation and an analysis of its performance.
2. SIMPLIFIED DATA ACQUISITION SYSTEMS PARAMETERS

For performing the market scan and evaluating the performance of a data acquisition system we used a simplified set of parameters. One important parameter is the Analog Digital Convertor representation. This parameter shows the amount of bits of the converted value. This parameter, together with the maximum voltage which can be measured, usually denoted as “reference voltage”, indicates the sensitivity of the system. Table 1 shows the minimum detectable variation in relation to the representation and reference voltage.

Table 1: Minimum variation detectable in relation to ADC representation and the reference voltage

| Reference Voltage (V) | ADC Representation (bit) | Minimum detectable variation (mV) |
|-----------------------|--------------------------|----------------------------------|
| 5                     | 10                       | 4.9                              |
| 1.1                   | 10                       | 1.1                              |
| 5                     | 8                        | 19.5                             |
| 1.1                   | 8                        | 4.2                              |

It is obvious that the more bits in the ADC representation, the more accurate the system.

The next interesting parameter is the sampling rate. This parameter represents the number of measurements that the system can take in the unit of time, which is taken here as one second.

Another relevant parameter considered is the acquisition time, which measures the capacity of the system for storing the time series at the given sample rate. We took this parameter into account for a specific reason. The development platforms which we considered are based on microcontrollers, with limited RAM memory. Each sample measured occupies memory space and leads to a limit of the time series which can be stored in RAM. Transfer of the time series from RAM in real time is usually problematic and can lead to quality issues for the recorded data so we considered for the analysis only the situation when the acquisition system does not transfer data in real time, but stores data in RAM at runtime and transfers it only after the acquisition is done.

With these parameters, certain basic theoretical estimations with regard to the performance of the system can be already done. For example, if we have a development platform limited to a specific ADC representation, we can estimate for a given acquisition frequency and knowing the amount of RAM, what would be the maximum acquisition time we can obtain. Table 2 shows some examples.

Table 2: Maximum acquisition time for an ADC with a known representation, sampling rate and RAM

| ADC Representation (Bits) | Sampling Rate (Samples / Second) | RAM (kBytes) | Maximum Acquisition Time (Seconds) |
|--------------------------|---------------------------------|-------------|-----------------------------------|
| 8                        | 20000                           | 256         | 13.1                              |
| 12                       | 20000                           | 256         | 8.8                               |
| 12                       | 1000                            | 256         | 174.8                             |
| 12                       | 20000                           | 8           | 0.3                               |
| 8                        | 1000                            | 256         | 262.1                             |

Similar estimations can be done using different perspectives: calculating the sampling rate which can be obtained for a known combination of representation, RAM size and time series length or calculating the amount of RAM needed for a known combination of ADC representation, sampling rate and maximum acquisition time.
3. MARKET SCAN FOR DEVELOPMENT PLATFORMS

During the next step we performed a market scan or survey, analyzing a set of 60 platforms available on local market (as of January 2017). The basis for the analysis was the following criteria, which include the above mentioned simplified parameters: development environment, type of processor used, ADC representation, ADC number of channels, RAM size, connectivity and price. Table 3 shows a subset of the market survey, a selection of 7 development platforms, filtered from the general market scan by having more than 10 bits representation and more than 128 kB RAM size.

Table 3: Selection of development platforms from the market scan

| Platform                  | Development Environment | Processor                                | ADC bits | ADC Channels | RAM (kB) | Price (EUR) |
|--------------------------|-------------------------|------------------------------------------|----------|--------------|----------|-------------|
| Arduino YUN (Chao J., et.al., 2017) | Arduino IDE            | ATmega32 u4 16 MHz Atheros AR9331 400MHz | 10       | 12           | 2.5 + 64000 | 85.56       |
| Arduino Industrial       | Arduino IDE            | ATmega32 u4 16 MHz Atheros AR9331 400MHz | 10       | 12           | 2.5 + 64000 | 85.56       |
| Teensy (LC, 3.2, 3.5, 3.6) (Yingwei L. et. al. 2016) | Arduino IDE            | 32 bit ARM Cortex                       | 13       | 12           | 8-256     | 22          |
| Particle Photon / Red Board Photon (Aayush B. 2017) | WebIDE (C Based)       | BCM43362 STM32F20 5 12MHz ARM Cortex M3 | 12       | 24           | 128       | 36.2222     |
| WiPy                     | MicroPython            | Texas Instrument CC3200 Cortex-M4        | 12       | 3            | 256       | 57.7778     |
| ESP32 Thing              | phant.io Arduino IDE (ESP32 Arduino Core) | Dual Core Tensilica LX6                | 12       | 18           | 520       | 28.89       |
This market survey revealed the fact that at the time this was performed, there were at least 20 promising platforms that would allow implementing useful data acquisition systems in a reasonable cost. The price range for these platforms is from 6 to 100 EUR and for most of the projects there is nothing else really needed for the implementation, besides the platform itself.

There is no platform which is the best, as the choice of it depends on the specific needs of the project. Implementation of a generic data acquisition system is not recommended, as it might increase the cost artificially. Our recommendation is to select one platform based on the specific project needs and implement the data acquisition system in the most efficient way reducing the cost to a minimum.

4. IMPLEMENTATION EXAMPLE

As a next step we implemented a basic data acquisition system using as starting point the Arduino Uno development platform. This platform was selected due to its wide use and availability, together with very good documentation and support and not necessary for its performance. The Arduino Uno is based on the microcontroller ATmega328, having an analog-digital converter with a representation of 10 bits, 6 analog channels but only a limited 2kB of RAM.

The quality of the ADC conversion is affected by the clock frequency. Figure 1 shows this dependency.

![Figure 1: ADC Conversion Quality Dependency on Clock Frequency](image)

Considering the small amount of RAM the Arduino Uno has, this data acquisition system can store only short time series. For a sampling rate of 1000 samples/second, with a representation of 10 bits, the system could store 1.6 seconds of data only.

When the sampling rate required by the project is low, in the range 10-500 Hz, the system can be extended with an SD card for storing runtime samples, as the transfer speed to the SD card does not affect the quality of the acquisition. For higher sampling rates, the SD card transfer rate could affect the quality and storing runtime data is not recommended.
As our acquisition sample rate requirements were in the range 10-500 Hz, our implementation used an additional SD shield connected to the basic development platform for runtime storing of data.

The system was implemented using two main modes for the sampling rate: free mode, running at maximum speed, based on the clock frequency, and fixed mode, running at a configurable fixed sampling rate. In addition, transfer of data was implemented in two configurable modes: runtime transfer on serial interface and storing of data on SD card.

The total cost of the system was around 40 EUR.

The implementation effort for the code was around 16 hours with some previous experience on the development environment.

The system was tested with a sinusoid waveform. The quality of the acquisition was good, and this makes the system suitable for being used in real projects.

Table 4 shows the performance obtained with this system, in the different working modes implemented.

| ADC Mode          | Acquisition Mode | Transfer Mode | Maximum Sampling Rate (Samples/Second) |
|-------------------|------------------|---------------|----------------------------------------|
| Free              | Live             | SD            | 1500                                   |
| Fixed Frequency   | Live             | SD            | 1500                                   |
| Free              | Live             | Serial        | 500                                    |
| Fixed Frequency   | Live             | Serial        | 500                                    |
| Fixed Frequency   | Snapshot         | RAM           | 33000                                  |
| Free              | Snapshot         | RAM           | 84210                                  |

It can be seen that the system is able to perform data acquisition with a sampling rate up to even 84 kHz, but this frequency is limited to a small time series, and the limitation is caused by the small amount of RAM. On normal functioning, with transfer of data on SD card, the system can go up to 1500 Hz sampling rate, which is sufficient for many real projects.

Figures 2 and 3 illustrate two examples of sinusoid waveform acquisition done with the system.

![Figure 2](image1.png)  
**Figure 2:** Example of data acquisition for a sinus wave form, real time operating mode, transfer of data on serial port, sampling rate = 50 Hz

Figure 4 illustrates another example of data acquisition, from a real system: the blitz from an Iphone 4s camera. It can be seen that the data acquisition system was able to detect the first pulse of red eye reduction, which normally is triggered by the IPhone software prior to the real blitz, to reduce the opening of the eye iris in order to reduce the red eye effect on the captured image.

![Figure 4](image2.png)
Figure 3: Example of data acquisition for a sinus wave form, real time operating mode, transfer of data on serial port, sampling rate = 250 Hz

Figure 4: Example of data acquisition for a “blitz” effect from an IPhone 4s camera. Sampling rate = 200 Hz.

The system described in this work was successfully used in acquiring time seria during a simple Dynamic Light Scattering experiment, with results published in (Chicea, D., & Rei, S., 2015).

5. CONCLUSIONS

Starting from the assumption that for most educational projects running in Universities, the performance required from a data acquisition system is a modest one, we investigated the idea of implementing a simple, own-designed data acquisition system instead of purchasing an expensive one from the market. After a selection of parameters to be used for the study, a market scan was performed, revealing a range of 60 development platforms which can be used for development, out of which at least 20 are of sufficient quality for the requirements of an educational project. The price range for these platforms was 6 – 100 EUR.

In addition, we successfully implemented a basic data acquisition system using a development platform selected from the market scan. The implementation was done initially as a proof of concept but came out to be sufficient to be used in a real scientific investigation (Chicea, D., & Rei, S., 2015).
The conclusion is that for educational projects running in Universities, a viable option is implementing a self-designed data acquisition system instead of purchasing a ready-made one from the market. In addition to the cost reduction, the involved developers will have plenty of opportunities for learning and this contributes to the overall purpose of the original educational project.

REFERENCES

Aayush Bista, *Smart Home Using Particle Photon*, Technical Report (2017), DOI: 10.13140/RG.2.2.23584.58885

Chao Jiang, Jun Wu, Yun Tu Yu Hu, *Design of fitness information recording and network monitoring system based on BDS/GPS and Arduino Yun*, (2017), DOI: 10.1201/9781315116242-91, Proceedings of the International Conference on Civil, Architecture and Environmental Engineering (ICCAE2016).

Chicea, D., & Rei, S. (2015). Time Series Space Phase Qualitative Analysis and a Possible Application. *Academic Journal of Manufacturing Engineering*. Vol.13 Issue 2, p148-153.

David Murphy (2016). Start Experimenting With Google’s ‘Science Journal’ App. PCMag, [http://www.pcmag.com/news/344653/start-experimenting-with-googles-science-journal-app](http://www.pcmag.com/news/344653/start-experimenting-with-googles-science-journal-app)

Enzo Mastinu, Bo Håkansson, Max Ortiz-Catalan, *Low-cost, open source bioelectric signal acquisition system*, (2017), Proc. Conference: 14th International Conference on Wearable and Implantable Body Sensor Networks, At High Tech Campus - Eindhoven, THE NETHERLANDS

National Instruments Low-Cost Data Acquisition Family. [http://www.ni.com/low-cost-daq/](http://www.ni.com/low-cost-daq/)

Yingwei Li, Yunjie Tong, Kimberly P. Lindsey, Kenroy R. Cayetano, Blaise deB. Frederick, *A low cost multichannel NIRS spectrometer for monitoring global physiological hemodynamic fluctuations*, (2016), Proc. Conference: Society for Functional Near Infrared Spectroscopy Biennial Meeting, At Université Paris Descartes, Paris, France.