Development of a time-variable nuclear pulser for half life measurements

Guilherme S. Zahn, Claudio Domienikan, Roberto P. M. Carvalhaes and Frederico A. Genezini

Instituto de Pesquisas Energéticas e Nucleares - IPEN-CNEN/SP. P.O. Box 11049, São Paulo, 05422-970, Brazil.

Abstract. In this work a time-variable pulser system with an exponentially-decaying pulse frequency is presented, which was developed using the low-cost, open-source Arduino microcontroler plataform. In this system, the microcontroller produces a TTL signal in the selected rate and a pulse shaper board adjusts it to be entered in an amplifier as a conventional pulser signal; both the decay constant and the initial pulse rate can be adjusted using a user-friendly control software, and the pulse amplitude can be adjusted using a potentiometer in the pulse shaper board. The pulser was tested using several combinations of initial pulse rate and decay constant, and the results show that the system is stable and reliable, and is suitable to be used in half-life measurements.

Keywords: time-variable pulser; half life
PACS: 29.30.Kv

INTRODUCTION

In half-life measurements of short-lived nuclides, it is important to make proper dead time corrections, since this dead time varies rapidly over time. The choice of methods to do this correction precisely is quite broad, going from special data acquisition systems to simple mathematical corrections [1]. Among the most reliable methods is the use of a pulser to determine the true livetime of a measurement [2, 3], but even this method has the limitation that the pulse rate is constant with time, adding different contributions to dead time during the measurements.

A solution proposed elsewhere [4] is to employ a pulser whose pulse frequency decays with a half-life very similar to the expected half-life of the isotope to be studied, so that the contribution of the pulsed to the dead time will decay with time together with the contribution of the radiation from the analysed source.

THE PULSER SYSTEM

The Arduino Uno board

In order to implement a very low-cost time-variable pulser, we opted for an Arduino Uno [5] microcontroller board. The Arduino is an open-source microcontroller board platform, frequently used in robotics projects, composed by an 8-bit RISC-based microcontroller chip and complementary systems intended to allow the programming of

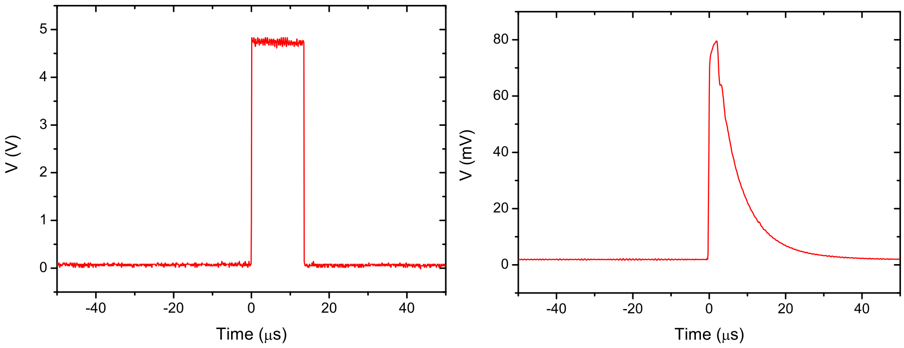

FIGURE 1. Sample output signals from the Arduino board (left) and the final output of the pulser (right).

the controller and the communication with other systems [6]. In the case of the Arduino Uno, the microcontroller used is an Atmel ATmega328P that runs at 16MHz and a very stable time resolution, with a nominal time jitter of about 0.0015% [7]. The Arduino is programmable using the Arduino IDE application, which uses the C/C++ language; the time pulser is implemmented using either the *delayMicroseconds()* routine (for delays up to 16383 microsseconds, precision better than 3μs) or the *delay()* one (which goes up to more than 4 billion milliseconds and is said to have an accuracy of 4-10μs).

The implemented software has an output pulse width of 13μs, and allows for the choice of the initial pulse separation (dt_0) and of the time decay constant (λ) so that at any time t the delay between two consecutive pulses (dt) is given by eq. 1; the minimum value for dt_0 is 1ms (which means a maximum pulse rate of 1000 pulses/second) and the minimum value for λ is $10^{-5}s^{-1}$ (which is equivalent to a half-life of approximately 19.2 hours).

$$dt = dt_0 \cdot e^{\lambda \cdot t} \tag{1}$$

The output of this system is shown the left side of Fig 1.

Pulse shaper board

In order to convert the 13μs 5V logical output of the Arduino board in a pulse that can be entered together with an HPGe output signal in a regular spectroscopy amplifier, a simple electronic system was developed that consists of a 74HC123 monostable IC which enlarges the pulse to 450μs and a CA3140 operational amplifier that is responsible for the integration and diferentiation of the pulse, thus resulting in a pulse whose shape is compatible with the expected output of a regular HPGe detector, as seen in the right side of Fig 1. Finally, a potentiometer was added to the end of the system to allow the adjustment of the final pulse height. The schematics of this board is presented in Fig. 2.

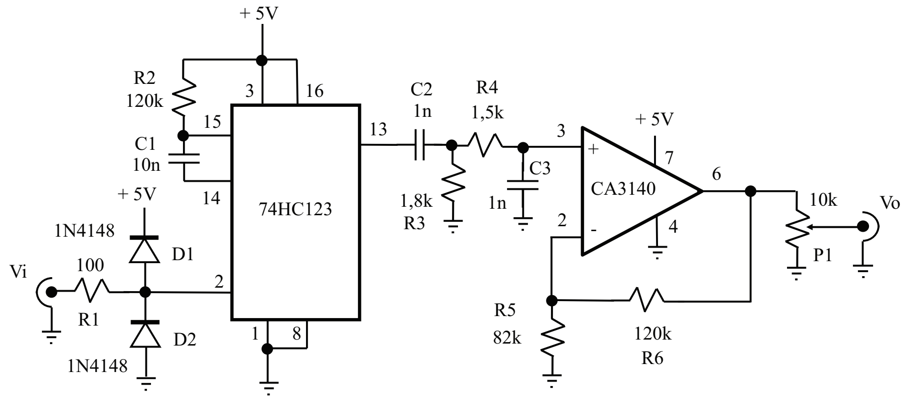

FIGURE 2. Schematics of the pulse shaper board.

EXPERIMENTAL PERFORMANCE

In order to check the experimental performance of the pulser, the output of the system was entered in a Ortec 572 spectroscopy amplifier and then in a Ortec 920-8 Multichannel Analyser coupled to a PC computer. Data acquisition was performed using the Ortec Maestro software in batch mode, so that several consecutive acquisitions of a fixed time were performed (the number of acquisitions and the duration of each were chosen according to the time decay constant selected); the spectra were then analysed using the IDeFix software [8] and then an exponential decay function was fitted to the resulting data – when an initial pulse separation time of 1μs was used, the non-paralyzable dead time correction [9] was added to the decay function. The experimentally-determined half-life was then compared to the nominal half-life chosen in the Arduino controller software to verify the pulser's precision; several combinations of initial pulse separation and decay constant were used, and long ($\sim$ 10h) measurements were also performed to asses the stability of the pulse height.

RESULTS

The results obtained in all the individual measurements are shown in Table 1; all measurements resulted in an uncertainty below 8% and the Z-Scores were all between -3 and 3, thus showing that the pulser is exact within a 99% confidence interval. The best results (best compromise between low uncertainty and low Z-Score) were generally achieved using a lower pulse separation time (i.e., a larger pulse rate) and a half-life between 20 and 140 seconds (at least, as there is a lack of points with that initial pulse rate in the interval up to 693 seconds). The larger uncertainties obtained for larger half-life values may be due to the use of the less precise *delay()* function for pulse separation times larger than 16μs, as well as to possible non-optimal choices of spectrum counting time/number of repetitions.

Two 10h-long measurements were made (with spectra collected each half hour) and the centroids of the peaks didn't change more than one channel (in a 8192-channel

TABLE 1. Results obtained in each individual measurement, with the respective initial pulse separation time, the Z-Score when compared to the nominal value and the relative uncertainty of the obtained result.

Nominal Half Life (s)	Measured Half Life (s)	Pulse separation (μs)	Z-Score	Relative Uncertainty (%)
6.9	6.1 (3)	1	-2.7	5.3
13.9	13.3 (5)	1	-1.2	3.5
23.1	22.1 (5)	1	-2.2	2.0
34.7	33.7 (6)	1	-1.7	1.6
46.2	46.1 (5)	1	-0.2	1.1
69.3	69.4 (6)	1	0.1	0.9
138.6	140.4 (13)	1	1.4	0.9
138.6	135.8 (12)	10	-2.4	0.9
277	274 (10)	10	-0.3	3.6
693	647 (46)	1	-1.0	7.1
693	714 (8)	10	2.6	1.1
1386	1433 (40)	10	1.2	2.8
3466	3536 (75)	10	0.9	2.1
6931	7543 (324)	1	1.9	4.3
6931	7173 (96)	10	2.5	1.3
6931	7072 (509)	50	0.3	7.2
6931	6866 (195)	100	-0.3	2.8
11552	11498 (56)	100	-1.0	0.5
69315	68759 (1216)	100	-0.5	1.8

spectrum), showing that the system's overall gain is very stable.

CONCLUSIONS

The time-variable nuclear pulser developed effectively delivered an exponentially-decaying pulse rate that followed the chosen decay constant within a 99% confidence interval and proved to be very stable with time. The results obtained show that in its present form the pulser can be used in the larger puse rate setting (10^3 pulses/second) for measurements of half-lifes between 20 seconds and 140 seconds with a relative uncertanty below 2% – further investigation is needed to asses its usability in the region up to 10 minutes. Nevertheless, the pulser software must be improved to work with lower pulse rates and/or for larger half-lives, as the *delay()* function seems to be a lot less reliable than the 16μs-limited *delayMicroseconds()* one.

REFERENCES

1. M. Wiernik, *Nucl. Instrum. Meth.* **95**, 13–18 (1971).
2. M. G. Strauss, L. L. Sifter, F. R. Lenkzus, and R. Brenner, *IEEE Trans. Nucl. Sci.* **15**, 3, 518–530 (1968).
3. O. U. Anders, *Nucl. Instrum. Meth.* **68**, 205–208 (1969).
4. H. H, Bolotin, M. G. Strauss, and D. A. McClure, *Nucl. Instrum. Meth.* **83**, 1–12 (1970).
5. Arduino Team, website at http://arduino.cc/en/Main/ArduinoBoardUno
6. Wikipedia, *Arduino – Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/w/index.php?title=Arduino.

7. Atmel Corporation, website at http://www.atmel.com/devices/atmega328p.aspx
8. P. Gouffon, *Manual do Programa IDeFix*, IFUSP, São Paulo (1983).
9. G. F. Knoll, *Radiation Detection and Measurement, 3rd Edition*, Wiley, New York, 1999, pp. 119–127.