



FUNCTION GENERATION AND VIRTUALIZATION ON RASPBERRY PI

Romain Alcesilas^{*1}, Remington Phelipe da Silva Correa², Victor Acrani Ruivo³, Renato Santos Carrijo³,
Josué Silva de Moraes³ e Aniel Silva de Moraes³

¹Grenoble INP - Esisar

²LASEC - Universidade Federal de Uberlândia

³FEELT - Universidade Federal de Uberlândia

Abstract - In some projects, it is needed to generate periodical functions, generally sinusoidal, square or triangle form. For this, a device that transforms a constant voltage into a periodic signal is used. In a classroom, sometimes it can be difficult to have one equipment per student, so virtualization can be used to share one equipment to all the computers in the classroom. The platform Raspberry Pi and the AD9833 are a cheap way to generate signals and they can be used for educational purposes like this project.

Keywords - Electronics, Function Generation, Raspberry Pi, Virtualization

I. INTRODUCTION

In university, to develop a project it is necessary to follow some constraints and one of them is the financial constraint. To illustrate it the following example is used: An electronical project will be developed to be applied in a course, but it is expensive to have one device per student. One of the solution for the financial constraint is virtualization. Instead of having one device per student, only one is necessary. The device is plugged into a server and shared over the network to the entire classroom. Then, all the computers can use the device as if it was plugged directly into it. In some projects, it is needed to generate a periodic signal at a specific frequency or form. Usually, a programmable waveform generator is used to do that. This device consists on a little chip that convert constant tension into periodic signal. It has registers that contains all the necessary data to configure the output signal. Moreover, this kind of device is generally inexpensive, and the generated signal has a satisfying quality. Also, the Raspberry Pi is a good platform for doing electronics. Originally, it was developed to facilitate educational projects. This micro-controller is inexpensive and can be used as a computer. Thus, it is possible use it to create an entire classroom equipped with Raspberry Pi as a computer. In that way, for educational projects, it is possible to use this kind of device and share it over the network. Combined to the Raspberry Pi, a program can be developed to overwrite the registers and generate easily a periodic signal.

Every computer on the classroom can change the settings of the generator.

II. DEVICES USED AND GENERAL INFORMATIONS

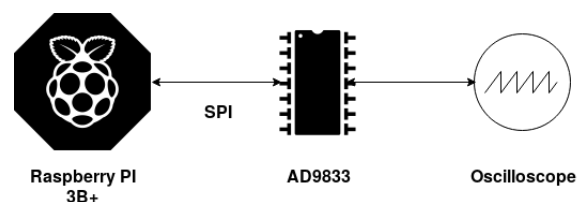
To do the function generator with a Raspberry Pi some choices about software and hardware were made. It is necessary to define the wave form generator (which force a protocol of communication) and the programming language to manipulate the registers of the generator. To understand how the system works, it is also necessary to understand some general concepts.

A. Raspberry Pi

The Raspberry Pi is a nano-computer based on ARM processor, which is used in many projects in different domains as automation or electronics, mainly because it is an inexpensive and easy-to-use device. It is possible to do embed program in the Raspberry Pi and use its GPIO ports to communicate with other devices. One of its biggest advantage is the possibility of programming in different languages and use it as a computer, or as a micro-controller. This project is based on Raspberry Pi 3B+, but it can easily be adapted over the other models of Raspberry Pi.

B. AD9833

The AD9833 is a low power programmable waveform generator, which can be bought at a price inferior than 5\$. This device can generate triangle, sinusoidal and square form signals. The frequency and the phase of the signal can be adjusted. The parametrization is based on registers which can be overwritten by sending a 16 bits word.



*romain.alcesilas@etu.esisar.grenoble-inp.fr

Fig. 1 : Global working of the signal generation

C. Serial Peripheral Interface

The AD9833 uses SPI to communicate. This full-duplex interface is based on a slave-master relation, which communicate thanks to four signals: a clock signal, a data output from master (MOSI) signal, a data output from slave (MISO) and a slave select (SS) signal. This interface is used for example on Raspberry Pi or on Arduino. An example of a communication between two Arduinos with SPI is shown below. The left one is the master, the right one is the slave.

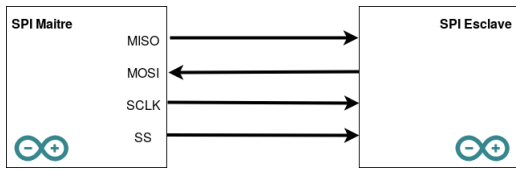


Fig. 2 : Communication between two Arduinos using SPI

D. Python

Python is a high-level interpreted programming language and it was chosen to do this work because it is easy to use and powerful. In addition, Python is natively implemented on Raspberry Pi libraries to use SPI with the GPIO ports and create graphical interfaces. Executable files which runs in different operating systems as Windows or Linux distributions can be created in Python.

E. Virtualization

The choice of the virtualization software was difficult. Indeed, the goal here is to share a USB device over the network. There are few software available, and just one is open-source: USB/IP. It works very well on Linux but it is not updated since a while. Then, the Windows driver does not work with the recent versions of the Linux kernel.

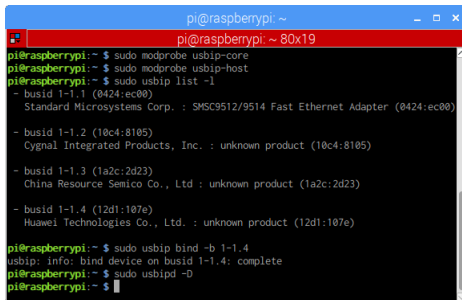


Fig. 3 : Example of sharing device using USB/IP

Another software available is VirtualHere. It works well on both OS but it is not open-source and limited to one device. Both softwares works by sharing USB device using IP protocol. They encapsulate the data into IP packets and send them into the network. As the software USB/IP is not compatible yet with Linux distributions, this work uses the software VirtualHere.

III. IMPLEMENTATION

This section focus on the electrical circuit to generate periodic signals and the Python program as the interface between the user and the generator. This makes easier the writing on the registers.

A. Electrical circuit

The wiring of the AD9833 on the Raspberry Pi is described below :

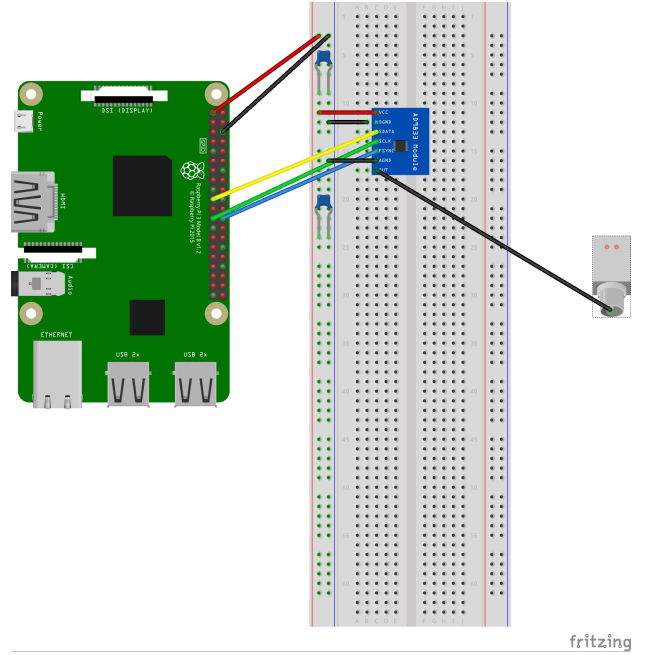


Fig. 4 : Electrical circuit

To use the generator, a constant voltage at the entry is needed. The voltage is fixed at 3.3V and generated by the Raspberry Pi. Three cables are used for the communication between the two devices using SPI. There is no MISO cable because the waveform generator did not send data to the Raspberry Pi, the communication between them is unilateral. There are also two cables for the ground (one analogic and one digital) which are connected to the ground of the Raspberry Pi. The last cable is used as output. It was used a probe to see the signal generated by the AD9833.

It is also necessary to uncouple the power supply using two capacitors of $0.1 \mu F$ and $10 \mu F$. Those capacitors are used to reduce the noise in the output signal.

B. Python program

To change the characteristics of the output signal, it is needed to send 16-bits words to the AD9833. These words contains the parameters needed to the manipulation of the registers. However, perform this task is not straightforward. For example, to change the frequency of the signal, it is necessary to send three words: one as a control word, which is used as a message to the AD9833 to say that is wanted to change the frequency register. The two last words contains the 28 bits of the frequency value, splitted as the 14 MSBs and the 14 LSBs.

Thus, a Python program was written in order to make these modifications on the output signal easier. Firstly, it was working only in console, but the second version of it was written and added a graphical interface and allow the support of CSV file (we can load a CSV file and save a configuration in this format). When launching the program, it is possible to choose the working mode. The graphical interface is presented below.

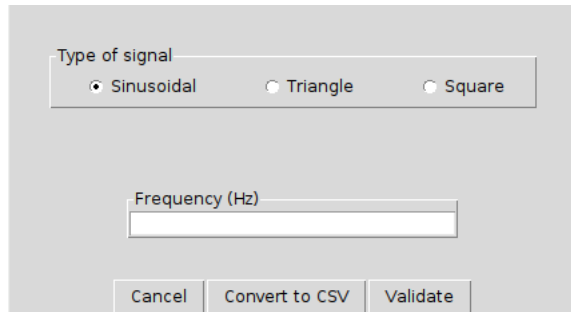


Fig. 5 : Graphical interface

The interface is simple: there are just two parameters, the type of signal and the frequency. These parameters can be easily changed without relaunching the program or unplugging the generator. It is just needed to pay attention to the GPIO ports: the program just load one SPI interface (more precisely, the SPI0 interface). If the cables are not connected to the good ports, the messages will not be send.

Even working on graphical mode, console mode or by importing a CSV file, the program finally call a function, which transform the data into binary words and then send them. The function is separated two cases: the first case is the sinusoidal and triangle one, and the second is the square one. That is because this type of signal is controlled by two bits in the control register, which named OPBITEN and MODE (bits 5 and 1). When the OPBITEN bit is at 0, the output is fixed by the value of the MODE bit. Conversely, when the OPBITEN is at 1, the output of ADC¹ is not connected to the output of the generator, but this is the value of the MSB. This generates a square wave, which is useful as a coarse clock.

| OPBITEN | MODE | OUPUT |
|---------|------|-----------------|
| 0 | 0 | Sinusoidal wave |
| 0 | 1 | Triangle wave |
| 1 | 0 | Square wave |
| 1 | 1 | Square wave |

Fig. 6 : Summary table of the OPBITEN and MODE bits.

These two cases create two different words, which are called "Control words", then the program will send two other binary words which set the frequency by changing the value of the frequency register. There are two frequency registers on the AD9833, but using the other one is not necessary in this case. Moreover, changing the frequency register can be done just by changing the value of the bit 11, the FSELECT bit, in the control word. The frequency word is calculated using the following equation:

$$word = \frac{freq}{f_{clk}} \times 2^{28}$$

¹Analog to Digital Converter

Where $freq$ is the frequency of the output signal, and f_{clk} the frequency of the clock. The factor 2^{28} is because the frequency is transformed into a word of 28 bits.

IV. RESULTS AND POSSIBLE IMPROVEMENTS

After setting up the electrical circuit, as it is described in the Fig. 4, some tests were done by observing the output signal on an oscilloscope.

A. Output waves

Activate the GPIO ports and the SPI protocol on the Raspberry Pi settings is necessary here, whitout them the generator will not work. Then some tests were done with the different modes of the program. Examples of the generated waves are presented below.

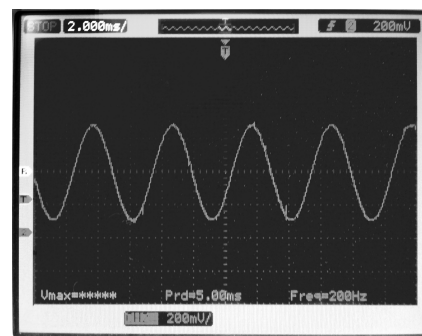


Fig. 7 : Sinusoidal output

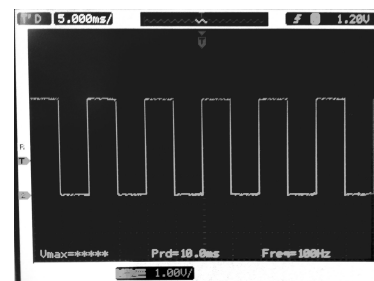


Fig. 8 : Square output

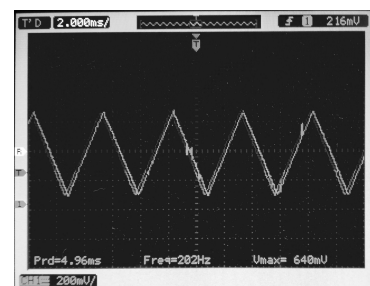


Fig. 9 : Triangle output

The results show that the generated signals are noisy. The following subsection shows ways to reduce the noise.

B. Improvements

This subsection shows some improvements that can be made on the system, both in hardware and software.

Generated signal

As it was said in the subsection IV.A, there is noise on the signal, but it can be reduced by using filtering methods. Indeed, the form of the output signal is known, the theoretical signal.

Python program

The Python program can be also upgraded: changes on the graphical interface to make it more complete, by adding management of the phase of the signal for example. For the console mode, a more convenient interface can be created. On the second version, to change the settings of the output signal it is necessary to restart the entire program. So, it is possible to create an interface similar to the graphical interface but working in the terminal. Finally, it is possible to add features to change every setting of the AD9833 and use it with its full potential.

Virtualization

The main problem here is that the device cannot be shared over the network. It is connected to the Raspberry Pi using GPIO ports, but not USB. Another micro-controller can be added as an intermediary between the Raspberry Pi (or the server) and the generator. That micro-controller will be connected to the USB port, and then it can be shared over the network. For example, a STM32 or an Arduino can be used. Instead of sharing the generator directly, the micro-controller

is shared. However, it is needed to make drastic changes in the program to make it work on the micro-controller. The other solution is to use a SPI-USB interface. This kind of device is cheaper than a micro-controller and the changes on the Python program will be less important.

V. CONCLUSION

In conclusion, the AD9833 offers a cheap way to generate different types of periodic signals. Combined to a Raspberry Pi, educational projects can be done easily. To make a proper conclusion, it is necessary to wait the implementation into a real classroom and the Windows driver of the USB/IP software.

REFERÊNCIAS

- [1] Analog devices *AD9833 : Low Power, 12.65 mW, 2.3 V to 5.5 V, Programmable Waveform Generator Datasheet*, Rev. F, <http://www.analog.com/media/en/technical-documentation/data-sheets/ad9833.pdf>.
- [2] Apprendre-Python *Interface graphique Tkinter python*, <http://apprendre-python.com/page-tkinter-interface-graphique-python-tutoriel>
- [3] Stephen Caudle *Python Spidev Documentation*, <https://github.com/doceme/py-spidev>, Feb 23, 2017