

Low Cost Full Configurable Vibration Meter Based on Accelerometers

Marcelo Romanssini, Felipe Quirino and Alessandro Girardi
Computer Architecture and Microelectronics Group
Federal University of Pampa - UNIPAMPA
Alegrete, RS, Brazil
Email: marcelorman03@gmail.com

Abstract—The study and understanding of physical phenomena of vibration and oscillation is an important field of engineering. From this study it is possible to develop new materials and optimize construction models, as well as to improve safety in buildings and machines. The availability of good instrumentation, in this case, is essential for capturing data and validating analytic models. This paper describes the implementation of a system for the measurement of vibration and oscillation based on low-cost, good precision accelerometers. The proposed system is composed of six accelerometers, one microcontroller, two communications interfaces, one RTC module and one Bluetooth module. The control module is implemented with a customized microcontroller synthesized in FPGA. The implemented system is fully configurable, allowing the customization the number of active sensors and on the resolution of acquired data.

Index Terms—accelerometer, sensor, microcontroller, vibration.

I. INTRODUCTION

Understanding certain physical phenomena, such as oscillation and vibration, is of great importance, especially in the area of engineering [1]. It allows the development of new materials and construction systems that contribute to optimizing safety in engineering. A vibration meter is an instrument used in civil engineering to measure the vibration of structures such as buildings, roads and bridges. In addition, specialized vibration meter devices can be used to measure the vibration of the human body. Although there are already equipment and systems capable of performing vibration and oscillation measurements in the market, they are very expensive. Furthermore, they are in general available in a closed form, where the user has limited chance to configure the parameters and adapt for the target application. The implementation of a new system to measure vibration and oscillation aims to offer an alternative tool, providing low cost, full customization and adequate precision. This paper describes the implementation of such a system for the measurement of vibration acceleration and oscillation based on low cost, good precision accelerometers.

The proposed system is composed of six three-axis accelerometers, in which it is possible to measure acceleration in x, y and z coordinates. Each sensor can be activated or deactivated according to the needs of each application. The sensors are read and configured by a microcontroller, which processes the linear acceleration information delivered by the accelerometers and the correspondent measured time

information provided by a Real Time Clock (RTC) module, and send them to a computer. On the computer side, we use a MATLAB interface for processing and saving data.

The implemented system is fully configurable and includes a dedicated PAMPIUM microcontroller [2]. This allows an easy adaptation to most applications. The communication with the computer takes place via Bluetooth, facilitating the connection and data transmission.

II. SYSTEM IMPLEMENTATION

The implemented system consists basically of six accelerometers, a microcontroller, two communication interfaces, a RTC module and a Bluetooth module. Fig. 1 depicts the block diagram of implemented system.

The acceleration data are collected through six MPU 6050 accelerometers, which provide linear acceleration on the x, y and z axes. The six sensors can be turned on or off according to the need of each application. PAMPIUM microcontroller is used for system control and configuration. It is also responsible for transmitting acceleration time data for a computer. The communication between the sensors and PAMPIUM is made through the I2C communication protocol. The communication between the PAMPIUM and the Bluetooth interface is implemented by the RS-232 communication protocol. Finally, the communication with the computer happens through the Bluetooth protocol [3].

Each MPU 6050 sensor has a three-axis accelerometer, responsible for measuring the linear acceleration on the x, y and z-axes, as well as a three-axis gyroscope that measures the angular acceleration over the x, y and z-axes. However for the implemented system we use only the linear acceleration data obtained through the accelerometers. The precision of the accelerometer is 16 bits. The initial calibration tolerance is $\pm 3\%$ and the sensitivity change in relation to temperature is $\pm 0.02 \%/^{\circ}\text{C}$ [3].

TABLE I
MPU 6050 ACCELEROMETER SPECIFICATION.

| Scale Range | Sensitivity Scale Factor |
|------------------|--------------------------|
| $\pm 2\text{g}$ | 16,384 LSB/g |
| $\pm 4\text{g}$ | 8,192 LSB/g |
| $\pm 8\text{g}$ | 4,906 LSB/g |
| $\pm 16\text{g}$ | 2,048 LSB/g |

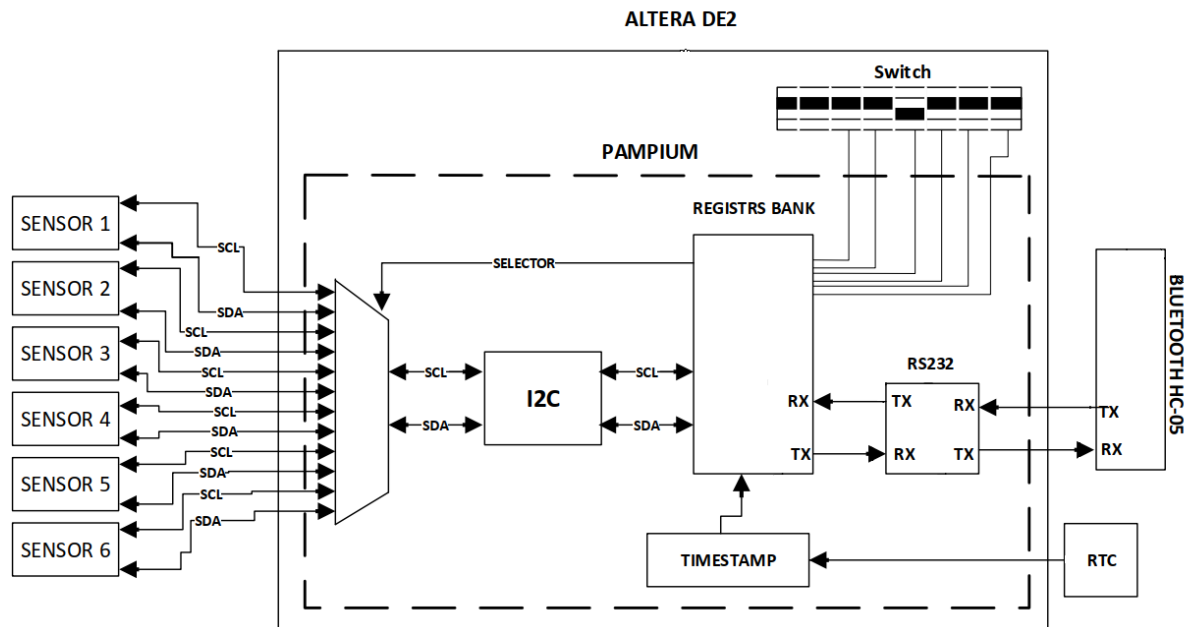


Fig. 1. Block diagram of the vibration meter based on accelerometers.

The accelerometer is capable of measuring acceleration on the three axes separately up to 16g and has four programmable ranges and four sensitivity settings, as show in Table I. The unit g refers to the value of the Earth gravitational acceleration. By default, MPU 6050 is configured with a scale range of $\pm 2g$. This is the largest sensitivity scale range, which allows the measurement of small movements with good accuracy [3].

The MPU 6050 sensor digitizes the x, y and z axes values collected from the accelerometers by means of three A/D (analog/Digital) converters. The read value for each axis is stored in two internal 8-bit registers. Communication through the I2C standard occurs at a maximum rate of 400 kHz and operates in a supply voltage range from 2.37 V to 3.46 V [3].

PAMPIUM is a fully configurable microcontroller developed by our group at UNIPAMPA. It consists of a general purpose microcontroller with 16-bit RISC architecture described in System Verilog. The circuit description is free and open source. Because it is configurable, it is possible to scale all the internal modules according to the applications needs, preventing the waste of area and power consumption while increasing the performance [2]. As the description of the circuit is free and all PAMPIUM operations are performed between registers, we connect the communication interfaces as well as the activation keys of the sensors directly of the register bank, facilitating the implementation of control protocols.

The I2C interface is a two-wire serial communications protocol usually used for the connection of low complexity devices. Communication is done between a master and a slave in 8-bit packets, where the master commands the bus and the slave responds to the commands sent by the master. The communication between master and slave is performed

through the SDA (serial data) pin, which is responsible for transmitting data, and SCL (serial clock), which performs the synchronization. We implemented the I2C communication interface so that its control is exercised directly by the PAMPIUM registers. For this, we allocate some bits of a specific register and connected them to the I2C interface. The communication interface connects the two input/output pins (SCL and SDA) to the sensors via a multiplexer/demultiplexer. For the connection between PAMPIUM and sensors, we use the external connection pins in the FPGA [4].

The RS-232 interface is a standard for serial data communication which uses negative voltage signals for high logic level "1" and positive for logic low level "0", ranging from $\pm 5V$ to $\pm 25V$, depending on the application. This protocol presents two separate RX and TX communication channels, which makes it possible to send and receive data in parallel. We implemented the RS-232 interface in a similar way to the I2C interface, in which all control pins are connected directly to the register bank. For this, we allocated the bits of the control interface to dedicated registers in the register bank. For the communication between the interface and the Bluetooth module, we used two pins (RX and TX) [4].

The DS3231 is a serial RTC driven by a temperature compensated $32kHz$ crystal oscillator. The TCXO (temperature compensated crystal oscillator) provides a stable and accurate reference clock, and maintains the RTC to within ± 2 minutes per year accuracy from $-40^{\circ}C$ to $+85^{\circ}C$. The TXCO frequency output is available at the $32kHz$ pin. The DS3231 has internal registers that can be accessed through an I2C bus interface. Typical oscillator startup time is less than one second. Approximately 2 seconds after supply voltage

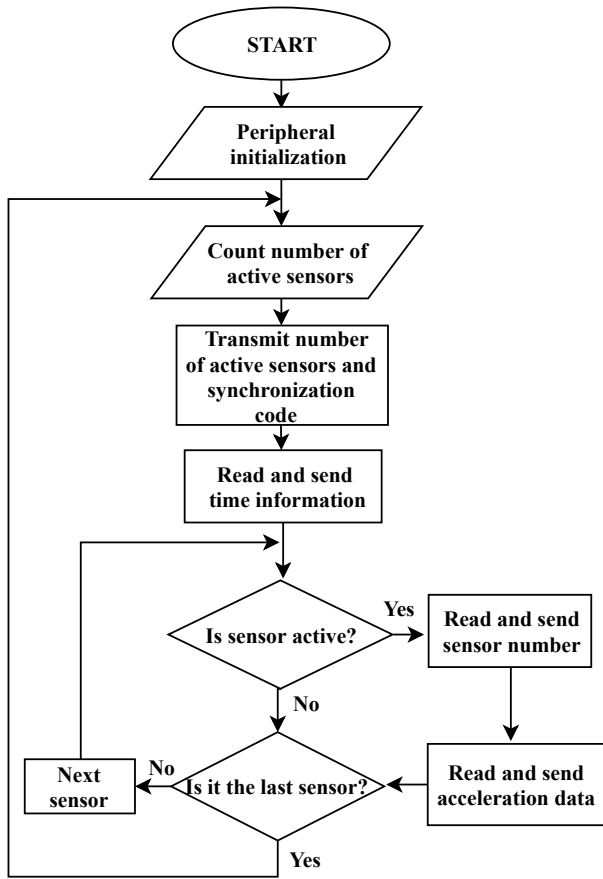


Fig. 2. Flowchart of the firmware stored in the PAMPIUM program memory.

is applied, the device makes a temperature measurement and applies the calculated correction to the oscillator. Once the oscillator is running, it continues to run as long as a valid power source is available, and the device continues to measure the temperature and correct the oscillator frequency every 64 seconds [5].

The clock/calendar of RTC module provides seconds, minutes, hours, day, date, month, and year information. The RTC module does not provide millisecond information, so we implemented a counter (Timestamp) for precise time count compatible to the acceleration transmitting rate. The Timestamp module is connected directly to the PAMPIUM register bank, thus facilitating reading and sending time information [5].

For transmitting data to a computer, we use the HC05 chip, which implements a Bluetooth SPP (Serial Port Protocol) module. It is designed for transparent wireless serial communication, making it easy to interface with the computer.

A. Firmware operation flow

We implemented a dedicated firmware in the hardware device. The internal program stored in the PAMPIUM microcontroller is shown in the flowchart of Fig. 2.

The start condition occurs automatically when the PAMPIUM is powered with 3.3V, when the peripherals are

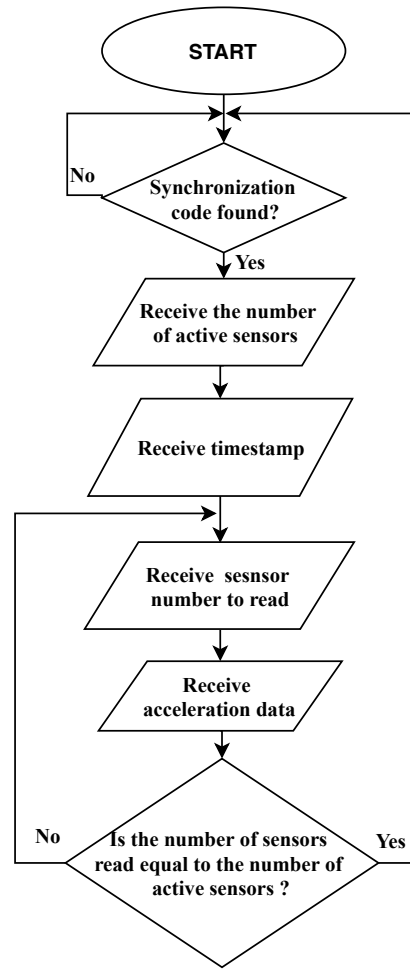


Fig. 3. Flowchart of the interface algorithm implemented in Matlab.

initialized (MPU6050 sensors, DS321 RTC and HC05 Bluetooth). The program checks how many sensors are active for reading. For this, it tests the bit values that were connected to the switches. Then, the synchronization code and the information of how many sensors are active for reading are sent to the computer through the RS-232 interface. The synchronization code consists of two bytes that are set to 255 and 127 in decimal. After this, the current time information provided by the RTC module is sent. A test routine is started, where each sensor is checked if it is active or not. This test is done for all sensors, in sequence, starting with sensor one. If the sensor is not active, the program tests if the current sensor is the last one. If true, it returns to the step of counting active sensors; if false, it tests the next sensor. In the case that the sensor is active, it sends the respective sensor number to the computer. Next, the acceleration data is read from the sensor and sent to the computer. At the end of this process, it tests if the current sensor is the last one and continues as previously described. This instruction flow is repeated as long as the system remains on.

