

Marek JASKUŁA, Mirosław ŁAZORYSZCZAK, Sebastian PERYT

WEST POMERANIAN UNIVERSITY OF TECHNOLOGY SZCZECIN, FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
52 Żołnierska Str., 71-210 Szczecin

Fast MEMS application prototyping using Arduino/LabView pair

Abstract

In this paper, the authors present selected examples of rapid prototyping environment consisting of a popular hardware platform like Arduino and software tools, like graphical language LabView. As a case study the MEMS (Micro Electro-Mechanical Systems) applications are described. First, the initial requirements regarding software and hardware are introduced. Next, some basic configuration procedures are presented. The main application example is a MEMS sensor data acquisition system. It shows the possibility of using LabView and non-dedicated hardware with a large number of Arduino compatible modules. The communication between Arduino and LabView is realized by a serial wire connection. Further examples show wireless communication with Arduino using Bluetooth link. The presented applications are very common in the embedded system environment and their implementation can be done with minimal effort on creating software code.

Keywords: Arduino, embedded systems, rapid prototyping.

1. Introduction

The prevalence of embedded systems with different capabilities and levels of complexity has contributed to the development of several hardware platforms that increase particular popularity among users. These include a number of Arduino varieties. Arduino is an easy-to-use open-source physical computing platform and a development environment created for building digital devices and interactive objects that are able to control the physical world. It is intended for a wide range of users starting from beginners who want to learn basics of electronics and embedded systems designing to professionals using Arduino for fast system prototyping. Even though software for Arduino can be written using C or C++, most of the sketches are written in special Arduino IDE using Arduino programming language. Currently there are about twenty official boards which vary in terms of used components and size. Almost all of them are controlled by different microcontrollers from ATmega family. The only difference is that Arduino Due contains a microcontroller based on Atmel SAM3X8E ARM Cortex-M3 CPU [2].

Due to the fact the Arduino environment (board, IDE, language) is open source, there are a lot of Arduino-like boards where most of them can be programmed with Arduino programming language and Arduino IDE. Additionally, developers can use special extensions called shields that add some extra functionalities to the design without manual wiring of components. Examples of such shields are GPS, GPRS, Ethernet, motor control modules etc.

The number of engineering scientific applications shows the popularity of Arduino. Some use commercial software environment and Arduino as an real world interface. In [6], the authors describe the BasketBallBot controlled by the Arduino platform supported by LabView and Matlab realizing fuzzy control. There are even several application for signal processing like scrambling algorithms for speech signals [4] or education fostering with Matlab. In [8], there are presented the detailed examples of Arduino and Matlab configuration as well as specific applications like an FM sound synthesis or a vocoder.

The one of the most important features is the large number of board manufacturers and suppliers providing ready to use software libraries. These libraries are mostly written in C or C++ and make developing of a system using such a component very easy and robust without the need of documentation reading or controlling all internal microcontroller registers.

2. LabView connectivity

LabVIEW software is a graphical system-design platform and development environment created by National Instruments [7]. It is mainly used for data acquisition, instrument control and industrial automation. Software created in LabVIEW is mostly built up from graphical blocks representing various actions, data structures and formulas. All blocks are grouped in packages that can be added using built-in VI Package Manager. One of such packages is called Linx.

Linx contains a set of VIs providing easy integration with common embedded platforms like Arduino, chipKIT and myRIO [5]. Based on the provided VIs, the user can easily access pins on a board as well as some communication interfaces like SPI, UART or I²C. This toolbox allows also speeding up the system development for data acquisition from some popular sensors like accelerometers (e.g. ADXL345, MMA8452), ultrasonic distance sensors or photocells, thermometers, real time clocks, controlling servo or many others.

To be able to use Linx blocks for control, the special firmware has to be uploaded into the Arduino board. It is done using firmware wizard directly from the LabVIEW environment. One of the main task of this firmware is to upload a special sketch which makes the board able to understand commands sent from the LabVIEW environment, interpret them and respond in the same manner.

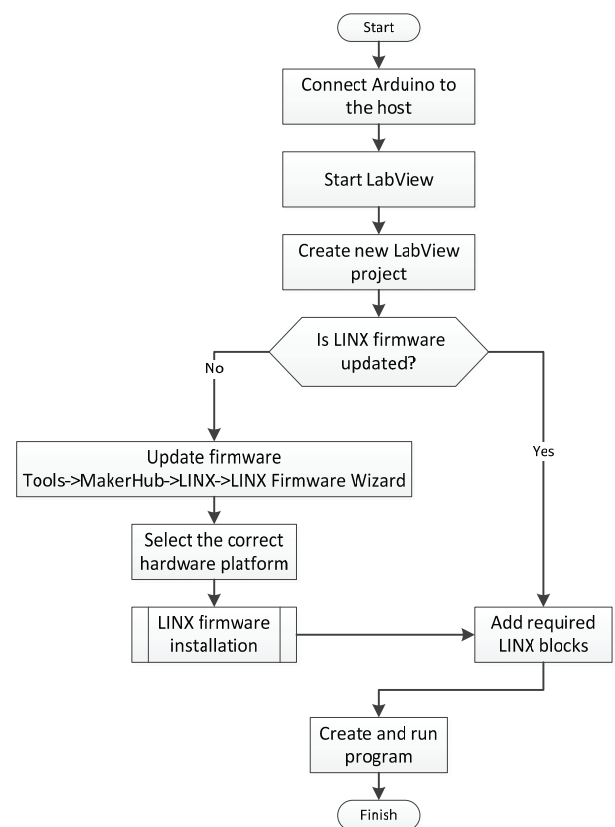


Fig. 1. The sequence of required steps while creating a new LabView/Arduino project

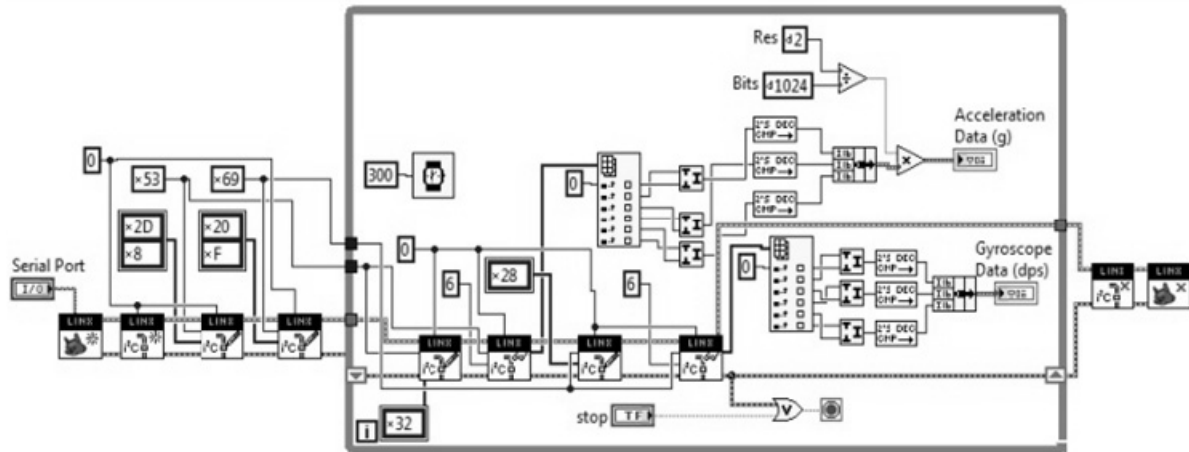


Fig. 2. MEMS sensor data acquisition program using Linx via USB. The program simple display 3D acc and gyro data

Communication between Arduino and LabVIEW can be achieved using serial communication by UART or USB connection. At the beginning, the user should select the appropriate board connection from the provided list of available serial ports. Linx itself does not upload the created software to the board. In fact, it only sends the control data using serial communication and then the uploaded firmware on Arduino takes care of proper analyzing and responding to received commands. The common flow-chart ensuring prerequisite compliance is depicted in Fig. 1.

The graphical programming language and Linx package allow the development of complex systems using Arduino; some peripherals can be done in minutes without writing any line of code.

3. Sample applications

There are three main possibilities of fast application prototyping with Arduino:

- using LabView and the arlier mentioned library Linx (wired serial connection),
- with BlueTerm project (Bluetooth connection),
- with ArduDroid project (Bluetooth connection).

Each of the listed options has advantages and disadvantages and the use of the selected one depends upon the user's experience with developing application and project assumptions and requirements. Next, the two sample applications will be described presenting LabView and BlueTerm capabilities.

The first presented example is MEMS application with Arduino. The used module is 10 DOF Inertial Measurement Unit (IMU) combining:

- three axis gyroscope L3G4200D,
- three axis acceleration sensor ADXL345,
- three axis magnetic field sensor HMC5883L,
- pressure sensor BMP085.

All parts of IMU module communicate with the board using I²C protocol.

3.1. Arduino/LabView connection using UART

The sample project is shown in Fig. 2. From the Linx point of view, the project creating starts with an Initialize block. The only requirement is adding a controlling block to determine PC COM port. Next, the Linx resource signal is supplied to all proceeding blocks. After that the I²C Open block is required. All the Linx programs after initialization use the while loop to put all their reading/writing blocks inside for continuous work. In the described sample program, two sensors (gyroscope and accelerometer) are used and therefore two I²C Write and Read blocks are used creating one couple per sensor.

At this step, all the required commands for proper communication are set together with addresses of sensors and from now on the user can easily read data for further analysis. It can be plotted on a graph or more advanced processing can be applied like filtering or other sophisticated techniques. The last two components outside while loop required at the end are I²C Close and Close blocks.

3.2. Arduino/Android connection using Bluetooth

The next example Arduino communication application is done based on Bluetooth link. Because Arduino does not include built-in Bluetooth, the HC-05 module has been adopted. It is an inexpensive, readymade solution for wireless communication that converts board serial communication to Bluetooth one.

The application BlueTerm will be used in the project to receive data from DHT11 module connected to Arduino. DHT11 module is a humidity and temperature sensor with OneWire communication. Both modules are linked with Arduino as shown in Fig. 3. Because BlueTerm does not have any special firmware the user has to write all the Arduino code by himself or using converted examples. BlueTerm is an application installed on an Android device used for simulating a serial console with Bluetooth communication capabilities [3]. It is preprogrammed for using 9600 baud rate and the same settings should be set on the Arduino board. Next, all commands sent or received on the board's serial are the ones from Bluetooth messages. In this way the user can read/send data between Arduino and Android without any physical connection. The sample data sent by Arduino board are shown in a BlueTerm window (Fig. 4).

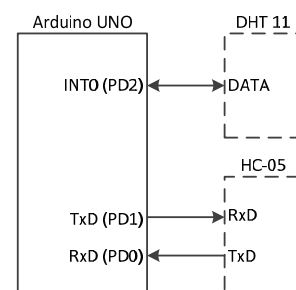


Fig. 3. Arduino with DHT11 and HC05 wiring

In the field of fast prototyping using Arduino/Android and Bluetooth connection, one more option is available through the open source project – ArduDroid. ArduDroid is a two-way Bluetooth-based Android controller for Arduino consisting of the

Arduino firmware and special Android application [1]. For communication the HC-05 module is recommended, too.

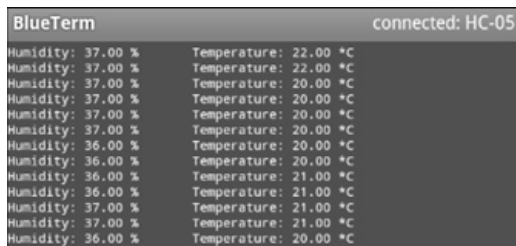


Fig. 4. A part of the main screen of BlueTerm application connected with Arduino via Bluetooth

After uploading the firmware onto the board, the user can send commands for all analog and digital pins. From the Android operating device, the user can toggle digital pins or modify values on PWM ones. As a feature there is also a basic serial read and write ability using one of the options: Send Data or Get Data (Fig. 5). The reading capability is hard coded in the provided firmware that means the user has to explicitly write before uploading it onto a board what message he wants to receive. Serial reading/writing is not proceeded like the previous one from BlueTerm in an infinite loop, instead the user has to update it manually by pressing the adequate button.

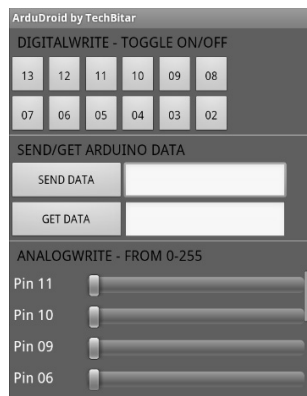


Fig. 5. Main screen of ArduDroid application connected with Arduino via Bluetooth

The main drawback of using the provided firmware is that this method allows only one-way communication with a board of most available features (while not as announced by the manufacturer two-way). It means the user is able to send digital signals (output value, PWM width or serial data) while reading is limited to serial one.

4. Summary

The Arduino board was created both for beginners but it is often used by professionals, too. Using the described techniques the development process can be substantially improved and some additional features can be added without any effort. However, the right technique is determined by application requirements.

The most advanced fast prototyping environment is LabView combined with Linx. This tool allows the virtually limitless development of an Arduino software with further sophisticated analysis from LabView. Thanks to this tool, a developer can also incorporate different functionalities like GUI or various forms of communication mechanisms without any additional applications.

ArduDroid can be used for simple applications like remote engine control or simple on/off capabilities. While it has built in serial communication it has to be up front programmed what makes it not so useful if the user wants to read various messages.

The simplest method is BlueTerm that is only a mobile Bluetooth terminal. While the user has to write all the Arduino source code by himself, the mobile terminal gives him unlimited possibilities of communication because no wires are required and only properties of radio link of the used module can determine the range of efficient communication.

5. References

- [1] ArduDroid project, <http://www.techbitar.com/ardudroid-simple-bluetooth-control-for-arduino-and-android.html>, (access Aug. 2015).
- [2] Arduino website, <http://www.arduino.cc>, (access August 2015).
- [3] BlueTerm repository, <https://github.com/johnhowe/BlueTerm>, (access August 2015).
- [4] Enache F., Deparateanu D., Oroian T., Popescu F., Vizitiu I.: Theoretical and Practical Implementation of Scrambling Algorithms for Speech Signals. 7th International Conference on Electronics, Computers and Artificial Intelligence, Bucharest, Romania, 2015.
- [5] LINX library, <https://www.labviewmakerhub.com/doku.php?id=libraries:linx:start>, (access Aug. 2015).
- [6] McPherson D. L., Ofoli A. R., Loveless T. D.: BasketballBot: Developing an Intelligent Controls Teaching Platform using LabView, MATLAB, and Arduino. Proceedings of the Southeast Conf., Fort Lauderdale, USA, 2015.
- [7] National Instruments LabView product page. <http://www.ni.com/labview/>, (access Aug. 2015).
- [8] Silva S., Soares S., Valente A., Marcelino S. T.: Digital Sound Processing using Arduino and Matlab. Science and Information Conference, London, UK, 2015.

Received: 02.10.2015

Paper reviewed

Accepted: 03.11.2015

Marek JASKUŁA, PhD

Marek Jaskuła received his MSc degree in Electronics Engineering and the PhD degree from the Faculty of Electrical Engineering of the Szczecin University of Technology in 1994 and 1999, respectively. Currently he is an lecture in West Pomeranian University of Technology, Szczecin, Faculty of Computer Science And Information Technology. His research activity focuses on signal processing and measurement in speech, acoustic and biomedical applications. He is the author or co-author of about 50 publications.

e-mail: Marek.Jaskula@zut.edu.pl



Mirosław LAZORYSZCZAK, PhD

Mirosław Łazoryszczak is an assistant professor at the West Pomeranian University of Technology Szczecin, Faculty of Computer Science and Information Technology. He received his PhD in 2006. His research interests include embedded systems, and application of programmable devices.

e-mail: mlazoryszczak@wi.zut.edu.pl



Sebastian PERYT, BSc

Sebastian Peryt is a 5th year student of Computer Science at the West Pomeranian University of Technology Szczecin, Faculty of Computer Science and Information Technology. His research interests include embedded systems and signal processing.

e-mail: speryt@wi.zut.edu.pl

