Computer Applications in Electrical Engineering
Vol. 12 *2014*

New, three dimensional desktop printer control system based on STM32F4 microcontroller with extended interfaces

Bogdan Fabiański
Poznań University of Technology
60-965 Poznan, ul. Piotrowo 3a, e-mail: bogdan.fabianski@put.poznan.pl

The article presents the design of a new three-dimensional desktop printer control system developed with use of experience in the operation of equipment for the mass market. Paper introduces in printing technology basis and outlines its historical context. Solutions to improve operational stability and functionality compared to the original design are proposed. New design bases on the achievements in the domain of open source software. Control over the printing process through the use of feedback signals to enable detection of movement errors is increased. With support for up to seven parallel stepper motor drives, it became possible to use multi-colored head with up to four colors. The new design and dedicated embedded program for control system based on a modern and efficient STM32F4 microprocessor family gives a substantial contribution to the development of a new branch of the three-dimensional printing. The article describes a first known attempt to use 32-bit STM32F4 unit with popular open-source project called Marlin and Teacup for 3D printer. Multiple communication interfaces support integration with industrial automation at IT and process-automation level. Descriptions of the functional implementation are verified by laboratory tests.

KEYWORDS: 3D desktop printer, embedded system, control system, STM32F4, RepRap

1. Introduction

The idea of the spatial (three-dimensional) printing, understood as the process of creating real objects of arbitrary shape based on a computer model is not new. The technology was developed already in 1980 leading to the establishment of the first operating structure in 1984 by 3D Systems company [1]. There are several methods of 3D printing. The more popular include: FDM - ***F****used* ***D****eposition* ***M****odeling* and DMLS - ***D****irect* ***M****etal* ***L****aser* ***S****intering*. Suitable examples of the FDM and DMLS printing are presented in Figure 1. The first method (FDM) based on the extraction of plasticized material by following the head by path coordinated with extrusion process. A layer-by-layer, reference model geometry of the spatial object is created.

FDM technology requiring the use of the nozzle what limits the use of materials with high melting points (e.g. metals), but are widely used with various kinds of plastics and its composites. The movable die is replaced in DMLS technology by controlled laser beam directed to a powdered raw material, leading to the weld. Reference beam trajectory in conjunction with the successive layers application of powder material results in the desired print object.

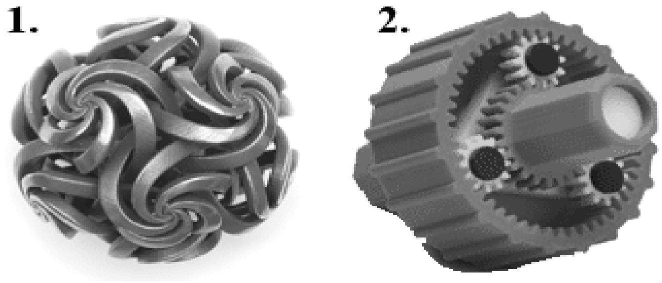

Fig. 1. An examples of most popular 3D printing results : 1- in DMLS technology; 2- FDM technology

With the opening of the technology (caused by international patents expiring), the idea of universal, self-replicating machines is born. The expression of that idea is three-dimensional printer. Around this idea, the *RepRap* (shortcut for ***Rep**licating **Rap**id Prototyper*) [2] project is created. *RepRap* basis on open-source domain and GNU GPL (from: ***G**eneral **P**ublic **L**icense*). The relatively low cost of using FDM technology (also called FFF from ***F**used **F**ilament **F**abrication*) etched into the domain of the project. *RepRap* is not only open source software, but also the openness of the hardware design, components and models documentation leading to the complete printer design. The basics of the theory of 3D printing can be found in [3] (in the context of bio-printing).

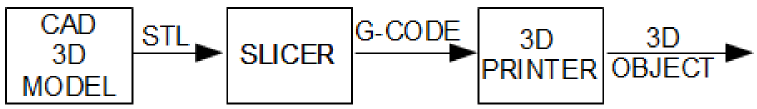

Fig. 2. 3D model design path with CAD/CAM systems usage

In the process of trajectory generation for the printing head, the machine language G-Code known well from CNC machines is used. The process of generating this code has been automated following the domain of rapid prototyping. In the chain of actions needed to achieve a given print follows each other (Figure 2): the design of the spatial object by using any CAD program for 3D editing that can save to *STL* format (file format created by *3D Systems* for printing spatial stereolithography method [4]). Next, the program (generally called *Slicer*) to generate individual layers of input model is applied. Slicer results in machine G-Code for direct use in 3D printers (some differences in G-

Code analysis and supported functionality are included into *Slicers* for different printers on the market). Proprietary software project realizes the task of processing the 3D model for spatial printing presented in [5] and [6].

2. Design

2.1. Reference control system

Implementing some of the basic assumptions about the traffic RepRap in the way of universal access to printer components, many projects control system constructions base on a relatively low-cost development boards [7]. The reference model uses the popular *Arduino* base plate with an 8-bit microcontroller family *ATmega* and *Pololu* extension board [8]. The basic and necessary tasks implemented in the controller can be divided into the following categories:

– motion/extrusion control,
– temperature control,
– printer head zero-position detection,
– fan/cooling control,
– communication with host and/or,
– internal memory use for standalone G-Code execution.

In the motion control domain, reference drivers have serious disadvantages and functional limitations. That is a main reason to develop a new control system constituting a significant contribution to the development of 3D printing technology.

The relatively small size of the reference stepper motor driver *A4988* housing [9], in practice, leads to substantial heating of the die and intermittent turning off when no noisy, efficient active cooling in association with a prominent heat sink applied. In this system there is a need to manually set the RMS current / torque by setting appropriate potentiometer output value. In reference system, there is also a lack of feedback from the process what implies that any step skip causes error invisible to the system disqualifying printing result. Furthermore, temperature sensors in the form of NTC (Negative Temperature Coefficient) have significant non-linearity characteristics of the resistance-temperature R(T) function, thus requiring its correction algorithms. USB interface via the FTDI limits the potential bandwidth of USB standard to microcontroller UART channel capabilities and maximum throughput of the FTDI at 1 [Mbps]. The central unit based on 8-bit architecture and a maximum operating frequency of 16 [MHz] (16 [MIPS]) [10] limits the maximum speed of the device and further complication of control algorithms (most assumed operations on 32-bit integer or floating point types of variables). On the Figure 3 reference control system main components are presented, while on Figure 4 final assembly of the controller.

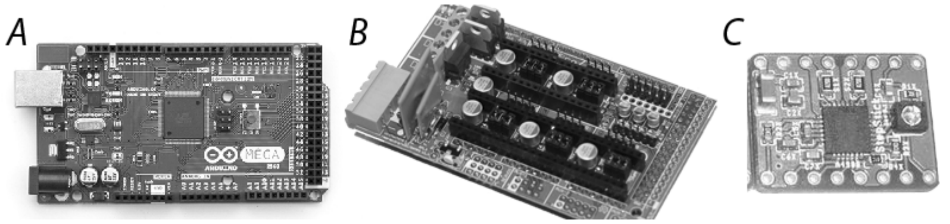

Fig. 3. Reference control system based on *Arduino ATMega* board (A), *Pololu* extension board (B) A4988 stepper motor driver (C)

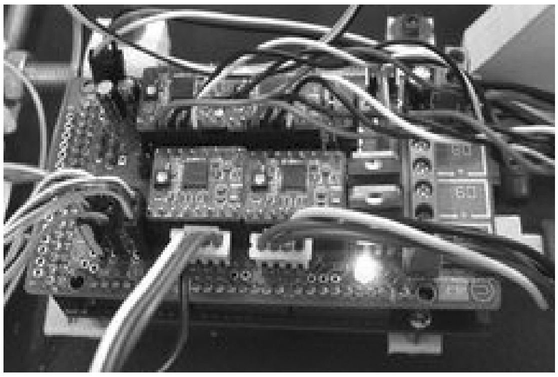

Fig. 4. Final assembly of the reference 3D printer controller

2.2. New control system

The main goal of the new project is elimination of limitations pointed out in previous chapter by:

- use of much more efficient microcontroller family STM32F4 [11] with maximum computing power at 210 [MIPS] and hardware floating point operations support,
- use of L6474 drivers with lower output stage transistors conduction resistance $R_{DS(On)}$ (180/370 [mΩ] with 320/320 [mΩ] in the driver A4988 in low/high side of the output bridge), lower thermal resistance (32 [K/W] reduced to 12 [K/W]), a higher value of effective continuous phase current (3[A] vs 2[A] in reference design),
- use of SPI digital interface with motor drivers with remote configurable current amplitude and availability of the feedback signal from the detected errors [12],
- use of temperature sensors based on PT1000-behaved linear response R(T) dependence and higher accuracy using an appropriate analog conversion path,

- storage interface implementation based on the SDIO (native interface of SD cards),
- implementation of the built-in Full Speed (FS) USB interface using the CDC device class operating directly with build in MS Windows driver (12 [Mbps] vs 1[Mbps] with FDI chip usage),
- implementation of the Ethernet/WiFi interface as an novel alternative for USB in desktop printers.

As shown on Figure 5, the controller consists of three separable PCBs (***P****rinted* ***C****ircuit* ***B****oards*):

- power electronics,
- microprocessor,
- wireless communication - WiFi board (optional).

For wireless communication a WizFi630 module is used. This module is chosen because of the user interface IEE802.11b/g/n providing full bus bandwidth and the ability to communicate via wired Ethernet.

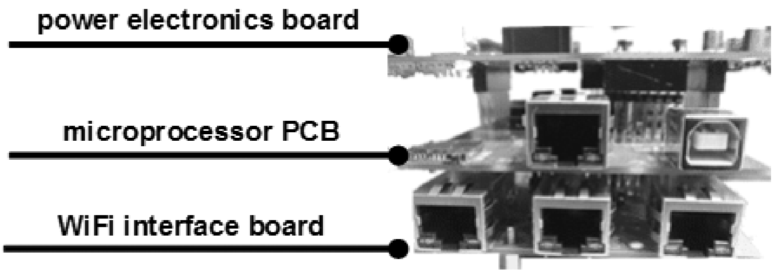

Fig. 5. PCBs of the new control system

Power electronics is described in Figure 6. It provides the necessary interfaces to control the printer by going far beyond the capabilities of reference driver.

The heater controller is based on the power transistor *AP9962* (40 [V]/32 [A]) in a compact *DPAK* with intermediate power stage consisting of a complementary *MOSFETs*. Smooth operation of the fans for the input signal PWM type provided through the active low-pass filter of the 1st order (RC configuration of the operational amplifier input) and the output buffer based on the pair of complementary bipolar transistors *BCP55* and *BCP52* in *SOT223* housing (maximum power approx. 12 [W] on channel).

Temperature measuring circuit based on PT1000 sensors is based on a dual operational amplifier, in which one is working in the configuration of current source (current at level approx. 1.3 [mA] to achieve wide voltage response) and the other as a differential amplifier.

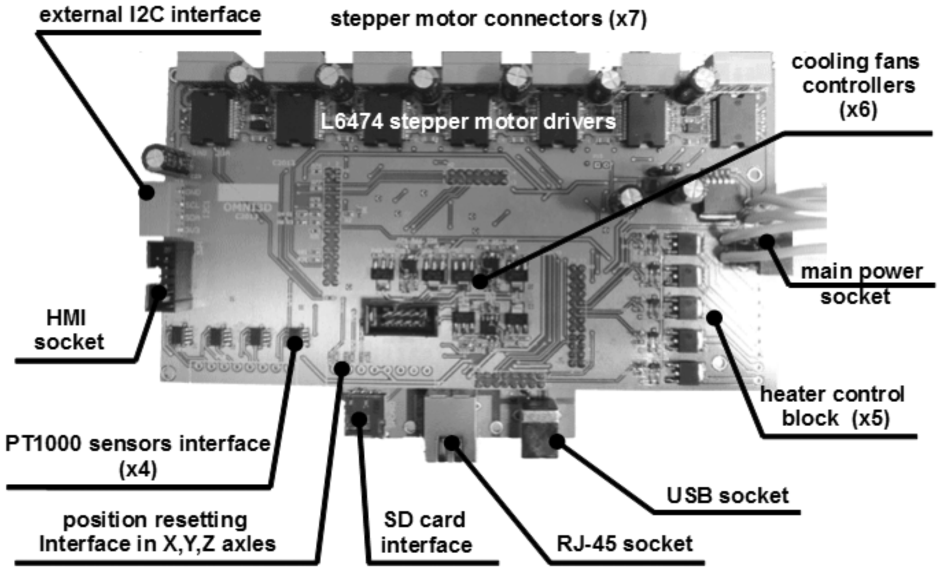

Fig. 6. Top view on the new control system (power electronics)

2.3. Interfaces

The very important aspect of the new design is to introduce Ethernet interface as an alternative to the USB. The USB interface at FS (full speed) mode uses standard STM32F4 peripheral. Embedded system driver is developed to comply with CDC device class [13]. Furthermore, after taking into account the *Windows* system requirements [14] it is possible to use standard USB driver with no additional installation requirement.

An Ethernet interface uses Ethernet STM32F4 peripheral and DP83848 MAC driver with reference design [15]. Wi-Fi communication uses wired Ethernet channel of WizFi360 module [16]. As the Wi-Fi module is transparent to the system there is nothing more to discuss. To make the Ethernet interface usable and compatible with *Pronterface* (printer PC host), software need to be recompiled in Windows environment (it is possible using the instruction from [17] web page, installing the *py2exe* extension and use of the supported compilation script *setup.py* file). Nowadays, native Python 2.7 IDE could be setup on Visual Studio. The Windows compilation of the newest *Pronterface* version is necessary because of the *TCP Stream* option availability from configuration menu. *TCP Stream* needs to be off while using Ethernet connection in the synchronous mode comparable to the USB (COM emulation) interface. Figure 7 presents *Pronterface* compiled under Windows from source binaries (Python language).

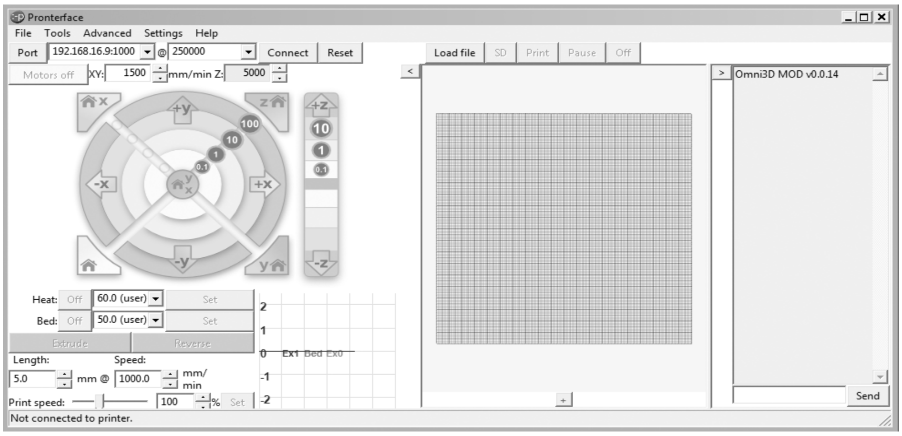

Fig. 7. Newest *Pronterface* version (2014.08.01) screenshot running from Windows after Python script compilation using py2exe tool

3. Experimental results

3.1. Temperature measurements

The challenge for analog-to-digital converting channels in the controller is running wires transmitting these signals parallel with motor and heater power cables. Heater current has almost rectangular form with value up to12 [A] and even more in large printers. The values results from the direct measurement of the analog-to-digital converters are subjected to filtration using various algorithms. The implementation effects of one of them shows Figure 8, in comparison with nominal measurement. Digital filter algorithm resulting from Figure 8 (B) bases on the number of M consecutive samples saving in memory.

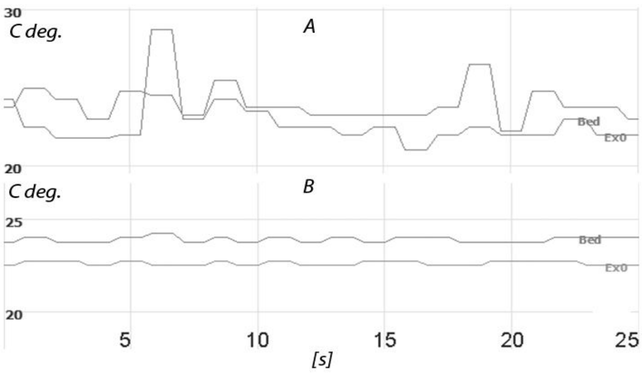

Fig. 8. Temperature acquisition process of the bed (Bed) and printer head (Ex0). Standard waveforms (without digital filtering) - (A) and with digital (median based) filtering (B)

Next, system sorts the record from the smallest to the largest values (using *bubble algorithm*), rejecting N extreme values (highest and lowest), and averaging the remaining number of L = M-2N samples. This algorithm shows significant immunity to transient disturbances. Delay due to the sampling frequency compared to the time constant of the heating elements is small enough to achieve good temperature control loop dynamic performance.

3.2. Motor drives

As can be seen from stacked waveforms in Figure 9, motor drives are characterized by similar waveforms of phase currents at the control mode of 1/16 microstepping. However, differences are visible in the way of keying both units (Figure 10), which difference is externalized in generated noise during operation.

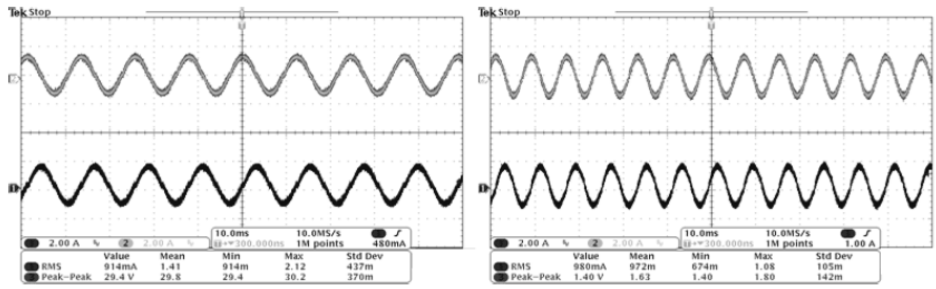

Fig. 9. Phase currents waveforms – for L6484 (new) on the left, A4988 (reference) on the right side

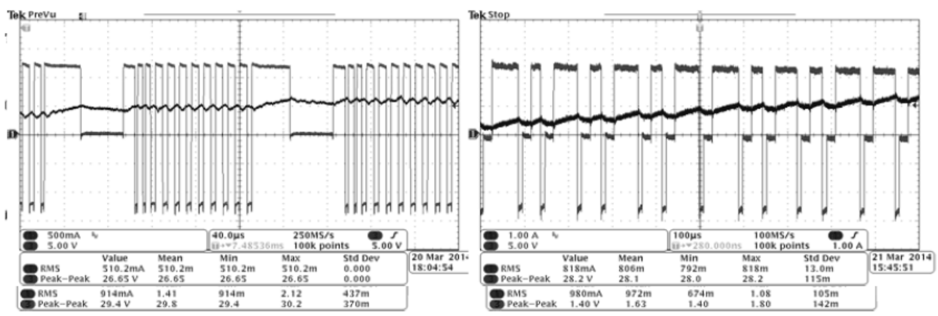

Fig. 10. Comparison of keying strategies of motor phases; left image - for the new driver L6484; right image - the reference A4988

As can be found from Figure 11 the voltage slopes are comparable and similar to the catalog parameters (180 [V/μs]), although the new controller allows to change the slope values (decrease from 180 [V/μs]) and reducing switching losses at the expense of increased emissions of electromagnetic interference.

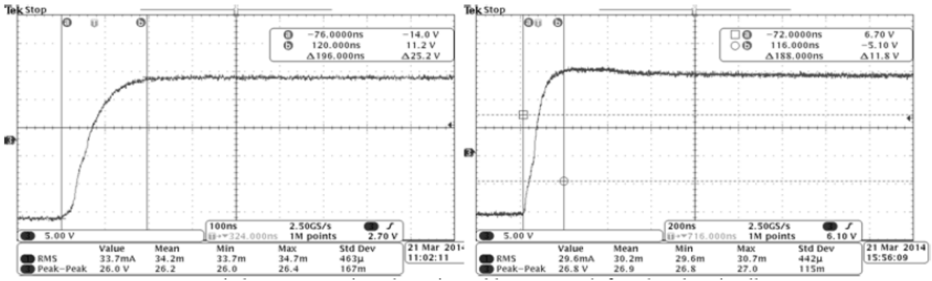

Fig. 11. Steepness of the slopes of the rising tensions in the keyed phases; Left - for the new driver L6484; Right - the reference A4988

Figure 12 shows representative image from the infrared camera, which can be seen compared to drivers operating under identical operating points. L6474 controller reaches a temperature of about 30% lower than the A4988 structure thereby minimizing the risk of overheating of the structure. The larger surface of L6474 driver gives active cooling more effective comparing to A4988. The grater airflow (voltage applied to the fan) the grater relative temperature differences (Figure 13).

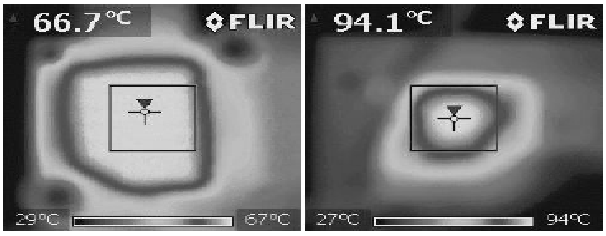

Fig. 12. Thermal imaging - surface temperature of the motor driver with phase current amplitude of 1[A] at steady-state, passive cooling; new driver L6474 on the left; reference A4988 on the right

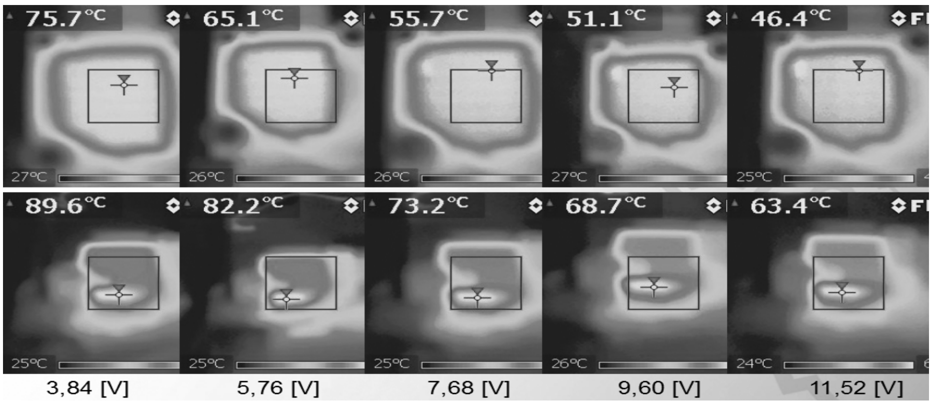

Fig. 13. Thermal imaging - surface temperature of the motor driver at different air flow (active cooling by fan); top: new L6474; bottom - old A4988

4. Conclusions

The paper presents the design of a new three-dimensional printer control system developed on the basis of experience in the current operation of equipment for the desktop market. 3D printing technology basics and its history were introduced.

Solutions that improved operational stability and functionality compared to the original design based on the achievements in the domain of open source software were proposed. Increased control over the printing process was described as the error feedback signal from motor drivers was utilized in the system. Thanks to the use up to seven stepper motor drives it was possible to use multi-colored head. New stepper motor driver gave better thermal response during current load what increased system stability. The use of multiple communication interfaces – including native USB (with serial COM emulation on the Windows OS side), Ethernet and WiFi - increased flexibility and usage of printer that could be accessed from compact devices - not only PCs - and from almost everywhere. Furthermore, the access to the bus used in the mass production systems opens up new possibilities for integration with industrial automation.

Cooperation with *Pronterface*, a popular 3D printer PC host application in the field of different interfaces was shown, as the compilation process from binaries in Windows and Python IDE for Visual Studio deployment.

References

[1] *3D printing*, http://en.wikipedia.org/wiki/3D_printing, 2014.

[2] *RepRapproject* , http://en.wikipedia.org/wiki/RepRap_Project, 2013.

[3] Anastasiou A., Tsirmpas C., Rompas A., Giokas K., Koutsouris D., *3D printing: Basic concepts mathematics and technologies*, IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE), p.1-4, 2013

[4] Brown A.C., de Beer D., *Development of a stereolithography (STL) slicing and G- code generation algorithm for an entry level 3-D printer*, p. 1-5, IEEE AFRICON conference, 2013.

[5] Schmidt, R., Ratto, M., *Design-to-Fabricate: Maker Hardware Requires Maker Software,* IEEE Computer Graphics and Applications, v.33, i. 6, p.26-34, 2013.

[6] Valero-Gomez A., Gonzalez-Gomez J., Almagro M., Salichs, M.A., *Boosting mechanical design with the C++ OOML and open source 3D printers*, Global Engineering Education Conference (EDUCON), p.1-7, 2012.

[7] Kentzer J., Koch B. M., Jones R.W., Villumsen E., *An open source hardware-based mechatronics project: The replicating rapid 3-D printer*, 4th International Conference on Mechatronics (ICOM), p.1-8, 2011.

[8] *Pololu project*: http://reprap.org/wiki/Arduino_Mega_Pololu_Shield, 2012.

[9] *A4988- Microstepping Driver*, datasheet , Allegro Microsystems, 2011.

[10] *ATmega1280*, datasheet, Atmel Corporation, 2007.

[11] *STM32F407XX* , datasheet, STMicroelectronics Corporation, 2013.

[12] *L6474*, datasheet, STMicroelectronics Corporation., 2011.
[13] *CDC device class specification*, http://www.usb.org, USB org, 2012.
[14] *USB device class drivers included in Windows,* http://msdn.microsoft.com/en-us/library/windows/hardware/ff538820(v=vs.85).aspx, 2014.
[15] *AN-1405 DP83848 Single 10/100 Mb/s Ethernet Transceiver,* datasheet, Texas Instruments, 2013.
[16] *WizFi630 User Manual*, Wiznet Corp, www.wiznet.co.kr, 2012.
[17] *Printrun open-source project,* http://reprap.org/wiki/Printrun, 2014.