A Low Cost Open-Controller for Interactive Robotic System

Juan Manuel Jacinto Villegas, Carlo Alberto Avizzano, Emanuele Ruffaldi and Massimo Bergamasco
Perceptual Robotics Lab (PERCRO), TeCIP Institute
Scuola Superiore Sant'Anna
Pisa, ITALY
n.lastname@sssup.it

***Abstract*— This paper presents the design and development of a new low cost device that allows real-time control application of a robotic system using novel methodologies of component based design: processor in the loop tuning and low/high level control. The particular design of this electronic board allows to control up to three permanent magnet (PM) DC motors per board that can be attached to magnetic or optical encoders. A triple USB connection can be used to program, debug and control simultaneously the different features of the board. Moreover, the generation of a new Simulink library allows debug for process analysis using Matlab/Simulink external mode as well as traditional code analysis protocol within the developer toolchain. The controller board has been developed for academic activities, but has also proven to be valid and robust in prototype application without requiring knowledge of device internals. The paper describes the design and hardware system, the development toolchain and evaluation test is presented and discussed.**

Keywords- Circuits; Sensors and Devices; Robot Control; Tools for Rapid prototyping; Printed Circuit Board (PCB).

I. INTRODUCTION

Traditional robot systems require stable laboratory support and large equipment plants in order to run smoothly. The design of new portable or wearable haptic devices, requires high computing power and motor driving system being built in small compact electronics like in literature [1], [2].

There are several challenges such as: miniaturizing the power supply and minimizing the power consumption; realizing small encumbrance electronics; optimize EM compatibility, etc. Most of these challenges conflicts with the usual approach of robot controllers based on professional boards/components takes from industrial automation applications.

However recently the trend has changed. Arduino based systems [3] helped to spread academic and homebuilt electronic and to familiarize a huge population of newbies into embedded system programming. For instance A. Soriano in [4] presents a robotic platform based on open hardware and oriented to educational teaching. André Araujo et al. in [5] present the integration of a compact educational mobile robotic platform that use Arduino boards integrated with Robotic Operating System (ROS) to provide hardware abstraction and intuitive operation mode. In [6] Francisco M. and Federico Cuesta performed and interesting educational platform based on Android and Arduino with Local Area Network (LAN) and Internet connection capabilities, to be used as an educational tool.

Since 2012, with the introduction of the ARM Cortex M4 processor [7] these low cost devices benefit also of high internal computing power combined with integrated single precision floating point support, an important feature for embedded control systems.

Several manufacturers foresaw the high potential offered by this architecture and delivered low cost prototyping boards (similar to Arduino) to enable designers developing their own home-built systems. For instance: Discovery F4 from ST-Microelectronics; XMC-4500 Relax lite from Infineon; Tiva-C Launchpad from Texas Instruments; LPCXpresso from NXP; or the FRDM platform from Freescale.

This hardware progress has simultaneously been accompanied by an increasing software support that allows new developers to design and integrate firmware on these devices at almost no cost. Among these the gcc compiler has been made available at no cost from several developers free GUI are available from independent developers (e.g. CooCox CoIDE), as well as from the respective hardware manufacturers.

It has always been hard to teach the related topics in an academic class due to fast pace of hardware/software development that increased the cost of ownership for the didactic material. Several authors have presented in the past years low cost robot controllers. Kilobot [8] for instance is a very simple robot vibrating motor which can cost less than 15 dollars (in an x1000 volume production).

In [9] we developed a training controller based on a networked processor and using a gentle donation of the TI University Program. However, with these premises, it becomes easier to develop this type of system by your own.

For universities and low volume production however the volume production to reduce cost proposed by Kilobot is not feasible and the approach followed by other researchers may be still too expensive [10].

Hence, our approach has followed a different principle. First, we gather minimum input-output and performance specifications from high-end project we developed for professional projects (e.g., the haptic gearshift developed in [11]). In particular we considered: type of sensing and actuation; continuous and maximum currents; number of I/O; loop rate for control; communication rate and interface type.

978-1-5090-0206-1/15 $31.00 © 2015 IEEE
DOI 10.1109/EMS.2015.75

Second we extracted from existing provider a set of general purpose boards (controllers, drivers, communication interfaces) that can be provided to each user at very low cost;

Third we designed low-cost PCBs (printed circuit boards) that can be integrated with these modules thus minimizing the overall cost.

Considering these factors we wanted to make a device dedicated for complex robot control (i.e. much higher performance, more I/Os, more peripherals, debugger, and memory) using low-cost components and easy programming by means of Matlab/Simulink blocks.

In the following of this paper we will highlight the component choice, the design rational, the electronics development, the software development, and we will show the application to a practical control case take from the research along an EU project REMEDI [12].

II. COTS COMPONENTS

As described above to reduce the overall components we made large use of Commercial off-the-shelf (COTS) components. There are three major components selected for the controller board: the computing node, the driver module and the communication module.

For the computing node we choose to use the STM32F407 microcontroller which comes integrated into a low cost development board (14USD) from ST Microelectronics, namely the Discovery F4. This board implements on board the following features:

- Integrated programmers through USB device;
- 168MHz cortexM4F with single precision floating point support;
- 1MB on chip flash memory with 192KB RAM;
- All device pins routed to two external headers;
- A service USB-OTG port at user disposal;
- Up to 6 (six) differential encoder inputs (Timers 1,2,3,4,5,8);
- Several PWM outputs.
- More than 16 ADC-12bit resolution;
- DMA operation for all devices;
- Two independent DAC output;
- Six independent UART port.

The board I/O has been almost fully exploited. Each motor requires at least eight I/O ports: three Encoder lines (CHA, CHB, Index), and five Motor control lines (Enable, INA, INB, PWM, Current Sense). Considering the internal multiplexer limitation we decided to implements three motors per controller for a total of 24 pins.

In addition to the above we also reserved six+six pins for general purpose digital and analog I/O; four pins for implementing two serial port communication; four pins for the USB OTG; eight pins for the Ethernet RMII interface, and two pins for I2C devices. For a total of 56 out of the 80 pins freely available at the connectors (see figure 1). In order to achieve this high number of available output, it has been necessary to remove from the board three chips: the audio DAC, the accelerometer and the microphone.

Using the discovery board the user already benefits of the onboard programmer device (STlinkv2), the required resonator crystals, voltage regulator for 3 and 5 volts, a service USB connector and few control buttons and LEDs.

Motor drivers were implemented using the ST VNH5019. These Integrated Circuits (IC) may operate over a large supply range while delivering high continuous (12A) and peak (30A) currents. The chip is powered with a dual supply and isolates the “logic power (2.5-5V)” from the “motor power (5.5-24V)”.

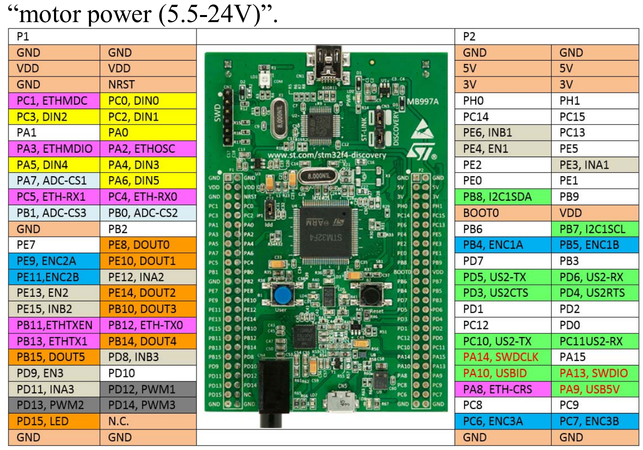

Figure 1. Discovery F4 pinout. Colors are associated to port type: Yellow Digital Input; pink Ethernet; grey motor driver; orange digital output; green communication devices (usb, usart, i2c); light gray encoder inputs.

The ICs have full distinct half bridge control for the bridge enable (ENA/ENB) and the bridge direction (INA, INB). A unique PWM control open and closes all switches together. The ICs also provide onboard current sensing and basic internal protection against reverse-voltage, over-voltage, under-voltage, over-temperature, thermal limits and over-current. An internal switching delay time (close to 110nS), allow these circuits to operate well at 25 kHz with 12bit resolution.

As for the STM32F407 also the mentioned drivers comes integrated into a COTS package from the Pololu distributor. It comes at low cost for volume sell (16 USD). There are several benefits in using this module including: certified thermal analysis at high currents, pin headers to reduce cost; motor connectors comes with the module; polarization and filter circuit to convert current sensing into any given voltage range.

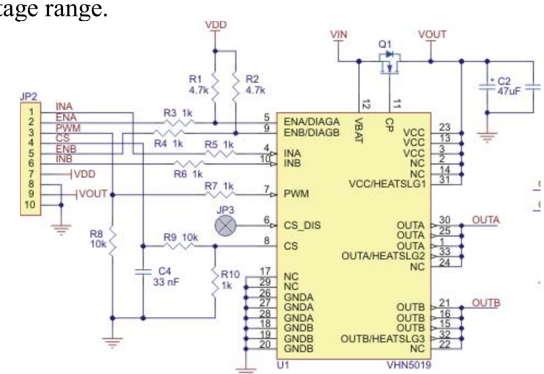

Figure 2. The electronic circuit of the POLOLU-VNH5019 Module. The circuit includes an RC filter network to convert the current output of the current-sense into a voltage.

The approximate current gain of the current sense is I_{out}/I_{mot}=7000, and a proper mapping of the RC filter (See figure 2) should be designed in order to have adequate current sensing on the ADC conversion. This map depends on the controller frequency, the maximum current and the ADC conversion voltage (which is fixed to 3V). The following table describes a typical setup:

TABLE I. COMPONENT SELECTION ACCORDING TO PERFORMANCE AND DESIGN ISSUES.

R10	R9	C4	A/V (Amax)	Fcut (Hz)
1kΩ	**10kΩ**	**33nF**	**7.1 (20)**	**484**
1kΩ	10kΩ	10nF	7.1 (20)	1580
3.3kΩ	33kΩ	10nF	2.15 (6.5)	1580
3.3kΩ	33kΩ	2.2nF	2.15 (6.5)	2200
3.3kΩ	3.3kΩ	3.3nF	2.15 (6.5)	1460
4.7kΩ	47kΩ	2.2nF	1.50 (4.5)	1540

Motor encoders, both differential and single ended (SE) can be directly connected to the electronics (in SE) configuration, since the board is 5Volt input tolerant no voltage divider is required. On the PCB, we provided space to insert two series resistor (or shorts) that protects encoder output from improper grounding during programming.

The last component of the board is the high speed remote communication (shown in figure 3). We implemented high-speed communication using a native high speed USB2Serial device such as the FTDI 232H. This device implements communication protocols at very high speed (480Mbit/s) and manage packet handshaking using the micro-frame policy (125μS per frame) which is much faster than the traditional frame policy (1mS per frame) adopted in full and low speed devices. Since USB is a host driven protocol, using high-speed USB (with 0 latency programmed) the user can achieve round trip time frequencies for the closed loop as higher as 2.66 kHz.

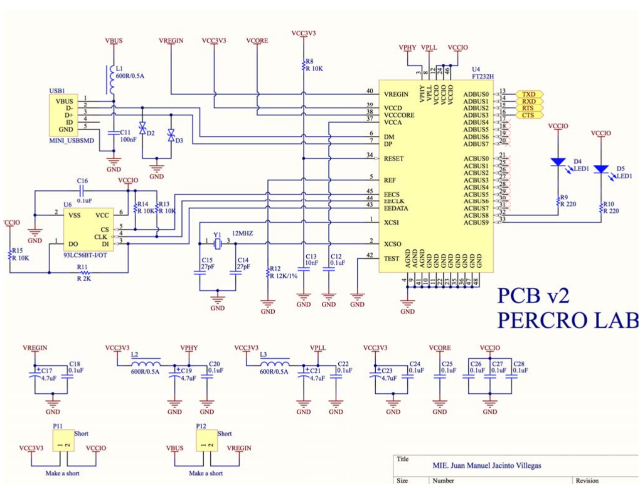

Figure 3. The FTDI area of the electronic control.

To implement the communication module we took inspiration from the exiting commercial module (UM232H), but this time we implemented all components on the PCB for three reasons: there is no commercial advantage in using the module prototype (which is quite expensive); high speed communication lines require minimum path length and electric disturbances; we may benefit of power supply and other electronic components which were already foreseen in the other boards of the PCB. Volume cost for the chip against the module is (3USD vs. 20USD).

III. PCB DESIGN AND INTEGRATION

The Printed Circuit Board (PCB) was designed using a dual layer design. All the components have been mounted on a single side. An overall layout of the board is shown in figure 4. The components have been arranged such that the motor connections and power signal are all available on the left side, while all control and communication signals are routed to the right/down side. This choice allows us to minimize the interference among component and to augment the distance between high voltage/power lines and the digital control lines.

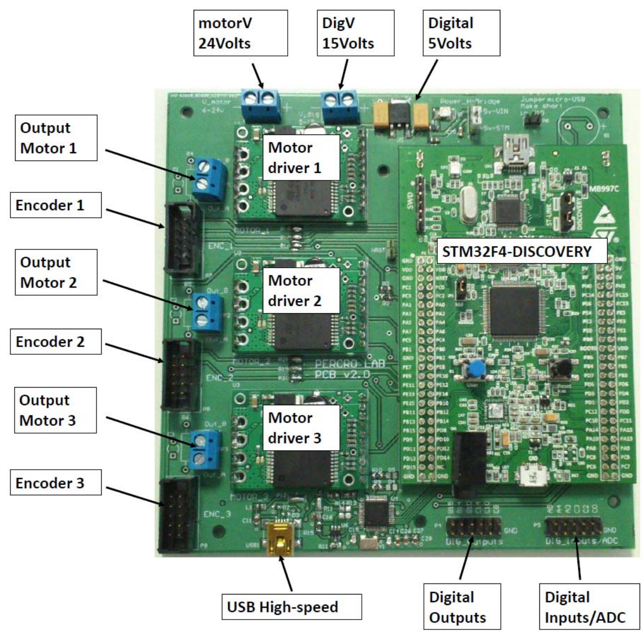

Figure 4. The PCB board version 2 description.

There are two versions of the PCB. The first version was dedicated to host an external Ethernet module (DP83848) to make the device operable through TCP-IP and a RS232 level shifter to allow serial communication through a standard DB9 connector. The second version mounts in the top area a FTDI 232H chip to make the device operable through high-speed USB.

On the TOP of the device (figure 4) we have the input for the high power supply. The "control logic" supply can also be provided through a 5V Voltage regulator, or using the USB supply directly. A couple of jumpers placed in the top allows the user to device which supply should be routed to the Discovery board and to the motor encoders.

The integration with user software is being performed using the recent STMcubeMX32 configurator which allows to setup the role of the chip pins, selecting the peripheral to use and for each of them defining the operational configuration. An example of this integration is shown in [13]. Figure 5 shows how we assigned pin-out to the STM32F407 chip mounted on the discovery board.

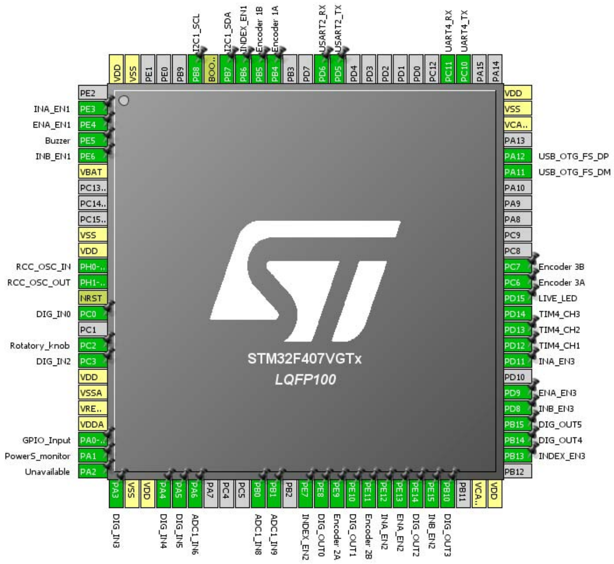

Figure 5. The assigned pin-map for the STM32F407 LQFP100. To ease interpretation each used pin (green color) has been labeled with its functional assignment.

One advantage of using the CubeMX configurator is the opportunity to generate proper configuration code that will initialize the whole device in accordance with the configuration chosen in the graphical user interface.

With a couple of Matlab script we manage to convert the software projects generated by the CubeMX configurator into a library which may compile with the existing free distribution of the GCC ARM Embedded. In such a way it is possible to use the code generated as a module available for being integrated into Simulink schematics.

IV. FIRMWARE DEVELOPMENT

ST microelectronics already distributes a Simulink package (STM-MAT) for generating code on the STM32F4xx processor line. However we decided to re-implement this process by ourselves for a certain amount of reasons: first we would like to use free compiler toolchain instead of those officially supported by the STM; second we considered the approach through CubeMX more versatile since libraries are more frequently updated and the same project can be retargeted to a different processor without waiting for a future support in this sense by STM; in addition at a certain level of development the use has the need to control with high level of detail how to combine different resources (e.g. DMA and ADC) to achieve a joint conversion. In our case, for instance we would like to run the ADC by oversampling and averaging inputs without increasing the basic schematic rate. This choice was not available in basic STM-MAT package. Finally we also required to enable a series of special scheduling features which cannot be available in the basic package, among these: preemptive scheduling, idle task functions, timer driven asynchronous functions. Even if these features may be achieved partially with the ongoing development of a target RTOS [14], they cannot achieve the fine granularity (well below 1mS) which is requires in most robotic control applications.

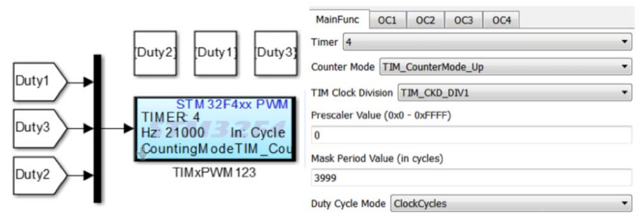

Figure 6. An example of the PWM control block.

However, since the STM cube, combined with the HAL libraries, already offered a very high level of hardware abstraction, re-implementing Simulink block-I/O for the associated device was almost easy.

In our implementation we re-implemented block-IO for almost all the device used, including some not originally provided by ST, among these: ADC block with oversampling; DAC block; digital I/O; UART controller; High level UART (with protocol and checksum similar to Ethernet); encoders, timers and PWM as distinctive blocks; scheduler block; USB-serial block; Interrupt and Idle blocks; I2C blocks (See example in fig. 6).

In addition to the above we also developed few debug and communication blocks that allows to transport C-STDIO (input output pipes) through serial or USB protocol, and/or to use this feature to combine with a shared memory block, to allow (real-time) external mode monitor simply playing schemes in Simulink while they are running on the remote target.

The Simulink(R) build procedure has been modified in order to generate file using the GCC-ARM toolchain and to generate a standard CoIDE project (http://www.coocox.org/). In such a way the developers can achieve different debug modalities:

- Traditional code debug (useful to verify user defined code);
- Protocol I/O debug (useful to verify overall system performances);
- External Mode debug on selected I/O (useful to monitor individual control component behavior when processing in the loop);
- Data logging and processing (useful to monitor long sequences of data);

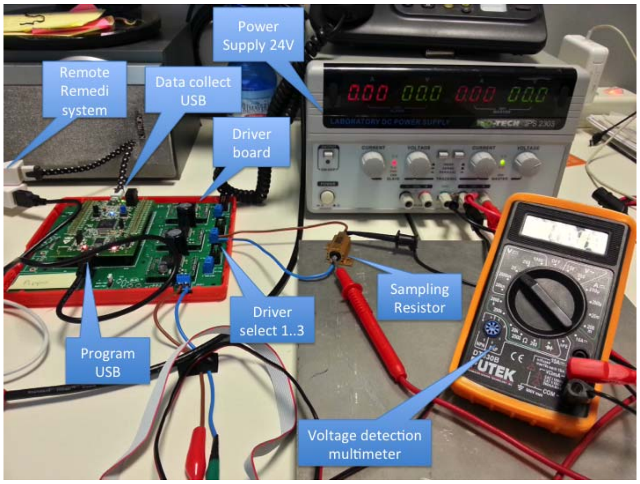

Figure 7. The board is programmed for ramped step on each driver. ADC values and motor currents are jointly monitored to asses the driving quality.

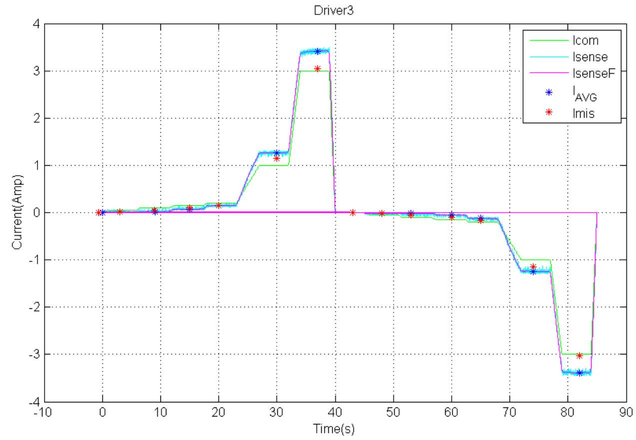

Figure 8. Driver information collection. The graphs shows the current profile acquired with different instruments and aligned with a batch procedure. Values on the x-axis represent the experiment time (in S); while data on the y-axis represent the measured current (in Amp).

V. Validation

In order to validate the control electronics we setup a joint test with some additional external equipment (see figure 7). During the test condition we connected in sequence each driver with a sampling resistor with known resistance (2Ω). We measured the voltage at the resistor leads through a shunted multi-meter.

For each test, we stimulated one of the three drivers with a given current profile. The current profile is a sequence of ramps and steady voltages to solicit the whole range of the motor operation (±3A). The collected information include both continuous data as well as sampled data from the multimeter. The collected information are: Commanded current (Icom); Current sensed on the respective driver (Isense); Filtered current with a zero lag smoother (IsenseF), achieved in post processing; Sampled current during steady state phases (Iavg); Measured/estimated current during the steady state phases (Imis). A sample data collection is shown in figure 8. To collect data we connected the service USB port. This port streams data from the microcontroller to a remote simulink scheme; the last provide to store samples into matlab variables. To reduce noise, we finally filtered Isense data (sampled with the microcontroller ADC) into Iavg whose estimation required offline-smoothing filter on the signals. During tests the device was supplied with a 24V generator. While the current driving is well shaped at large currents, we found that small currents (below 100mA) may have some profile distortion. This distortions however is almost static and regular.

Since it happens when the average power supply (and therefore the duty cycle) is close to 0.5-1% of the overall current range, we concluded that the non-linearity was induce by the MOSFET switching time. At 25 kHz PWM, a 1% duty pulse has an overall duration that is lesser than 500nS that is close to the rise time limit provided by the drivers.

Since it is not advisable to reduce the PWM switching frequency, to prevent this effect there are two opportunities: 1) reducing the driver motor supply will consequently reduce the current non linearity range; 2) modeling the non linearity shape can reduce this range by a factor of 4. In our test we decided the second solution which allowed us to diminish the non linearity while preserving the maximum speed/torque of the device. An example of the calibration procedure is shown in figure 9.

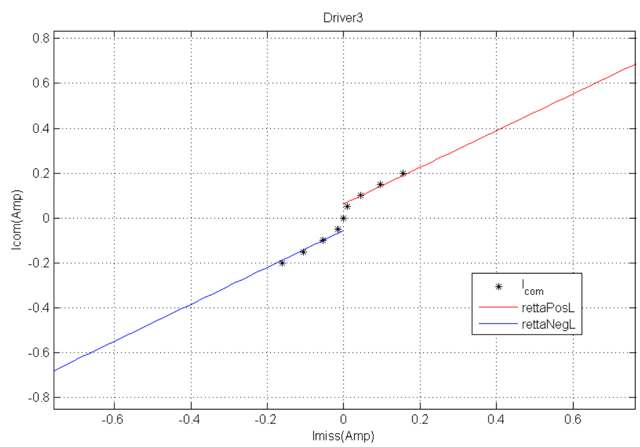

Figure 9. Estimated current distortion in the range 0-200 mA. The plot shows the measured currents (Imis) against the desired one (Icom). A piecewise linear interpolation has been implemented to compensate the distortion.

Other tests on the equipment included the encoder reading in the 4X modality; the digital I/O lines; the H-Bridge duty to PWM conversion; the serial and USB communication with remote PC device.

Once these tests were completed the controller system has been integrated with an existing Delta-like mechanism [15], [16] from the REMEDI project (http://www.remedi-project.eu/).

The controller firmware was integrated with all major features required, including: high frequency current regulator (5 kHz); joint torque, force, velocity and position controllers (1 kHz); direct and inverse kinematic; gravity and friction compensation; motor thermal models; initial calibration; plus a manager state machine that provides to failure check, component control and communication protocols. A video of the working system has been uploaded at this location (https://goo.gl/zTbPVU).

Using interaction with the HAL, most of the tasks were delegated to DMA controller who provides to acquire ADC information, manage USB and USART data, computing data checksum, without loading the CPU.

The computational load for the microcontroller to manage all the above tasks was below 20% of overall computing power (45% without compiler optimizations). This performance numbers have been determined by using an internal timer as a benchmark peripheral and monitoring processor time at start/stop of each Matlab loop.

An overall quality test of the computed information and the control system has been performed by controlling the device along a vertical trajectory, slow enough to solicit only friction and gravitational effects. In figure 10, we represented for each joint the model computed torque against the real torques required to provide the motion.

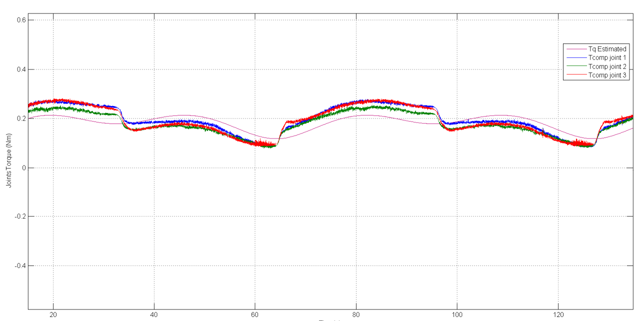

Figure 10. Extimated (Tq) vs real torques (Tcomp) when the device is slowly moved up down along the central vertical axis of the Delta kinematics (EE trayectory *XYZ* = [0, 0, 0.12 + 0.05*sin(t/10)]).

VI. CONCLUSION

We developed a microcontroller based open controller for robot that allows controlling user designed robot with a reduced cost of development (see table II).

When compared to existing similar solution the proposed system allows controlling high-end devices with a very flexible prototyping procedure. Most of the development can be done from within Matlab/Simulink without requiring knowledge of device internals. Nevertheless the development procedure allows users to modify the code generation and the device configuration in every details.

The support towards the HAL libraries and the self generated code, allows the user to enable all kinds of compiler optimization within the generated code without the need to insert any memory barrier or other execution order protection.

TABLE II. DESIGN COST.

Component	Est. Price (EUR)
Discovery F4	14
VNH 5019 X3	48
FTDI 232H	3
PCB	15
Misc. SMD	12
Total	92

The building and debugging procedure allows all kind of debug without compromising the real-time performances of the control process: process in the loop, batch debug, logging and I/O debug.

The system has been applied in replacement of professional equipment without loss of control quality and with an overall cost reduction in the control equipment of about 75:1.

ACKNOWLEDGMENT

This work has been carried out within the framework of the following projects: the EU project REMEDI, grant number 610902, The Tuscany Regional Project TAUM and the private cooperation with Yanmar Research Europe. The authors are grateful to the sponsor for the financial support received for the described development. Also the author wish to acknowledge the project partners and the laboratory staff who kindly supported the conceptual discussion of the design and the technical work of development.

REFERENCES

[1] H. Liu, P. Meusel, N. Seitz, B. Willberg, G. Hirzinger, M. Jin, Y. Liu, R. Wei, and Z. Xie, "The modular multisensory dlr-hit-hand," *Mechanism and Machine Theory*, vol. 42, no. 5, pp. 612–625, 2007.

[2] C. Avizzano, M. Satler, G. Cappiello, A. Scoglio, E. Ruffaldi, M. Bergamasco *et al.*, "Motore: A mobile haptic interface for neurorehabilitation," in *RO-MAN, 2011 IEEE. IEEE*, 2011, pp. 383–388.

[3] O. Pineño, "Arduipod box: A low-cost and open-source skinner box using an ipod touch and an arduino microcontroller," *Behavior research methods*, vol. 46, no. 1, pp. 196–205, 2014.

[4] A. Soriano, L. Marın, M. Valles, A. Valera, and P. Albertos, "Low cost platform for automatic control education based on open hardware.," in *World Congress Cape Town, South Africa*, vol. 19, 2014, pp. 9044–9050.

[5] A. Araùjo, D. Portugal, M. S. Couceiro, and R. P. Rocha, "Integrating arduino-based educational mobile robots in ros," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 2, pp. 281–298, 2015.

[6] F. M. L´opez-Rodr´ıguez and F. Cuesta, "Andruino-a1: Low-cost educational mobile robot based on android and arduino," *Journal of Intelligent & Robotic Systems*, pp. 1–14.

[7] J. Yiu, *The Definitive Guide to ARMR CortexR -M3 and CortexR -M4 Processors*. Newnes, 2013.

[8] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3293–3298.

[9] C. A. Avizzano, E. Ruffaldi, D. Leonardis, and M. Bergamasco, "A networked haptic embedded controller," in *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*. IEEE, 2014, pp. 52–57.

[10] R. LeGrand, K. Machulis, D. P. Miller, R. Sargent, and A. Wright, "The xbc: a modern low-cost mobile robot controller," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3896–3900.

[11] M. Solazzi, D. Pellegrinetti, P. Tripicchio, A. Frisoli, and M. Bergamasco, "Dynamics modeling of an encountered haptic interface for ball catching and impact tasks simulation," in *Haptics: Perception, Devices, Mobility, and Communication*. Springer, 2012, pp. 540-551.

[12] E. Ruffaldi, A. Filippeschi, F. Brizzi, J. M. Jacinto, and C. A. Avizzano, "Encountered haptic augmented reality interface for remote examination," in *3D User Interfaces (3DUI), 2015 IEEE Symposium on.* IEEE, 2015, pp. 179–180.

[13] K. Holman and Z. Szabo, "Microcontroller based application prototyping using domain specific modeling," in *Applied Machine Intelligence and Informatics (SAMI), 2015 IEEE 13th International Symposium on*. IEEE, 2015, pp. 199–202.

[14] D. P. Renaux and F. Pottker, "Applicability of the cmsis-rtos standard to the internet of things," in *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2014 IEEE 17th international Symposium on*. IEEE, 2014, pp. 284–291.

[15] F. Pierrot, C. Reynaud, and A. Fournier, "Delta: a simple and efficient parallel robot," *Robotica*, vol. 8, no. 02, pp. 105–109, 1990.

[16] S. Grange, F. Conti, P. Rouiller, P. Helmer, and C. Baur, "The delta haptic device," Tech. Rep., 2001.