Arduino based Controller for the Smart Assistive Mobility Hardware

P. Lengvenis, R. Maskeliunas, V. Raudonis
Department of Process Control, Kaunas University of Technology,
Studentų St. 48–327, Kaunas LT-51367, Lithuania, phone: +370 682 40371
pauliuslengvenis@gmail.com

***Abstract*—The main goal of this work is the development of Arduino microcontroller based integral solution for the smart, assistive mobility hardware. The paper presents schematics of device control and electrics, the control logic, the power regulation chain model and affiliated calculate efficiency parameters. The overall experimental investigation of real-life performance of our Arduino model proved quite successful. The added power regulator allowed achieving the stable outputs (±0.001V) to mimic the original controller. The deviation in trajectory compared to the ideal model was small (< 10 %) and can be further reduced by doing micro corrections.**

***Index Terms*—Microcontrollers, smart sensors, mobile robot sensing systems, modelling.**

I. INTRODUCTION

Assistive technologies typically allow individuals to accomplish tasks they might not be able to do otherwise and can be considered with an emphasis on potential barriers, driving forces and the potential role of the government funding [[1]]. However most of these are: not affordable, not easy to obtain, have poor performance, are designed without consulting the targeted user's needs [[2]].

As controllers for powered mobility vehicles become more and more technologically sophisticated more people are provided the means of independent mobility. A quote of Cooper's intelligent control still remains important designing such systems [[3]]: "*Tunable controllers have had a significant impact, but they only allude to the possibilities for improved mobility. Shared control proves promising for expanding the number and variety of users. Integrated control will increase the independence of power wheelchair users in all realms of assistive technology. Fault-tolerant control can guide one towards safer and more reliable power wheelchairs*".

In this article we are proposing our own very effective low cost solution for the control of standard commercial mobility vehicle based on a low powered (~50mA) 16 MHz Arduino MEGA ADK microcontroller [4]–[9] capable of 54 Digital I/O (14 of them can be used as PWM outputs) and 16 Analog Inputs.

II. THE REALM OF MICROCONTROLLER

Microcontrollers are becoming more and more common in smart home conceptions due to a low cost of production and reasonably simple implementation. Most of them are capable of typical automation and control tasks [[4]]. Small microcontrollers are also often used in hierarchical control systems for the initial processing of information in lower layers of implementations and then passing a result to the higher layers. Another interesting example is the design of a smart home system capable of wireless data mining. The end nodes collect data and send it back to the base station where an 8051 microcontroller combined with an Arduino board interprets the information allowing the home monitoring features from a smartphone or remote computer [5]. Other popular implementation is the Intelligent Home Navigation Systems [6] offering a combination of a wheelchair, a voice module and a navigation module capable of letting an elderly or physically challenged people to move inside their living environments. A system can be further enhanced by implementing a microcontroller, for controlling the device by voice through speech processing for example using Hawkboard (OMAP processor) [7]. Another design for controlling the powered wheelchair was developed using Infineon XC886 microcontroller and Infineon BTS7970B power module connected to the motor and drive system electronics [8]. This system was controlled by these chips by comparing the spoken voice commands to the ones trained and sending affiliated control signals to the microcontroller which in turn powered on or the other wheel.

III. SYSTEM ARCHITECTURE

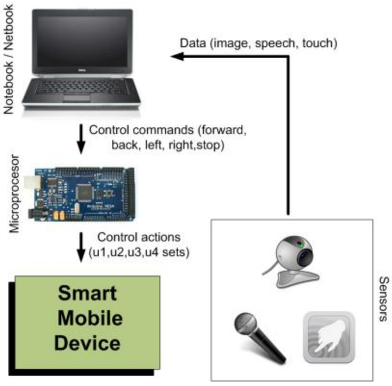
Notebook / Netbook
Data (image, speech, touch)
Control commands (forward, back, left, right,stop)
Microprocesor
Control actions (u1,u2,u3,u4 sets)
Smart Mobile Device
Sensors

Fig. 1. Schematics of system architecture.

Manuscript received March 12, 2012; accepted May 12, 2012.
This research was funded by a grant (No. MIP-037/2011) from the Research Council of Lithuania.

The system architecture is shown in Fig. 1. The Arduino microcontroller platform was used for data mining and signal forming. The input signals (either video, audio, or gestures) were formed and processed on a standard computer and sent via USB connection to Arduino board, which then formed the physical control curves and outputted specific voltages via PWM outs to the control chip of our smart mobility vehicle.

IV. CONTROL OVERTAKE MODEL

On a standard mobility device the movement was controlled by a magnetic contactor joystick manipulating 4 different output voltage values and changing the associated electromagnetic fields whose parameters are read by a built-in microcontroller at all times. The overtaking of control was necessary in order to allow the input from our own separate multimodal controller. So a SIMULINK model (Fig. 2) was created simulating the overtaking of control and outputting the necessary voltage values for the control of electromagnetic field parameters.

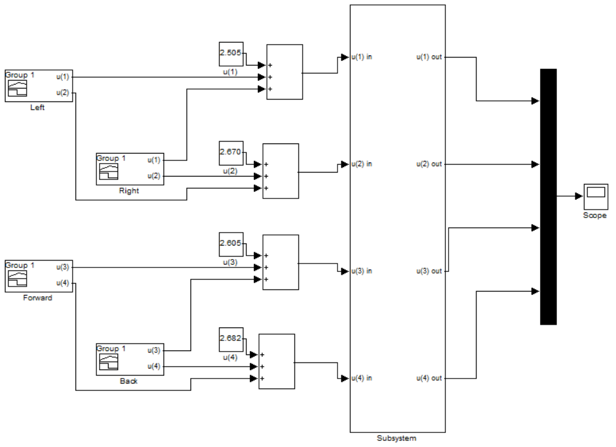

Fig. 2. Device control model in MATLAB Simulink.

The four main driving directions were modeled as voltage ramps (the initial values were measured manually). The average control values at max. speed are offered in Table I.

TABLE I. THE AVERAGE CONTROL VALUES MOVING AT MAX. SPEED.

Action/Output (V)	u(1)	u(2)	u(3)	u(4)
STOP	2.505	2.670	2.605	2.682
Forward	2.417	2.620	3.489	3.312
Back	2.534	2.684	1.812	2.114
Right	1.442	1.711	2.550	2.713
Left	3.311	3.297	2.610	2.640

TABLE II. THE AVERAGE CONTROL VALUES WHILE STOPPED.

Action/Output difference (V)	u(1)	u(2)	u(3)	u(4)
Forward	~ 0	~ 0	0.884	0.63
Back	~ 0	~ 0	-0.792	-0.568
Right	-1.063	-0.959	~ 0	~ 0
Left	0.806	0.627	~ 0	~ 0

Further experimentations proved that specific almost constant voltage sets were outputted in stopped mode as well (shown in Table II). This had to be accounted in our model, noting that while moving in one direction only one set of voltages varied significantly (either positively or negatively), the other sets varied insignificantly.

V. IMPLEMENTATION

The above control overtake model was implemented to the Arduino controller which was then wired to the hardware of our mobility device. The schematic of our implementation is presented in Fig. 3.

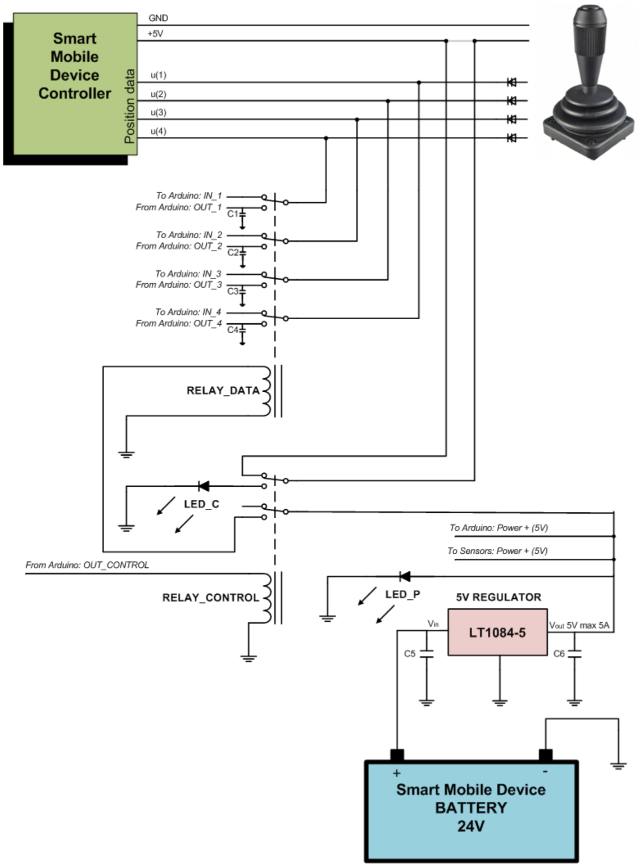

Fig. 3. Schematics of device control and electrics.

We have connected Arduino controller to mobility device's controller signal path to read (IN_1, IN_2, IN_3, IN_4) and set (OUT_1, OUT_2, OUT_3, OUT_4) voltages. Also in built-in joystick side we've added 4 diodes to disable the affect to other system components which were used on the original magnetic contactor board for a situation when we overtake a control and set output voltages from Arduino. Control overtake relay (RELAY_CONTROL) were set OFF in standby, so a power source (+5V) was connected to the joystick so we could also retain the control of a device by a mechanical manipulation of the original joystick. On the overtake condition RELAY_CONTROL goes on and RELAY_DATA connects Arduino outputs to the mobility device's controller while in the same time disconnecting a power source from the joystick.

When the system powers up the Arduino microprocessor reads values of the device's joystick and sets outputs with same values, then waits for a command from the computer and, if a movement command was received RELAY_CONTROL relay goes active and Arduino overtakes the control of the device. The schematics of the control logic are presented in Fig. 4.

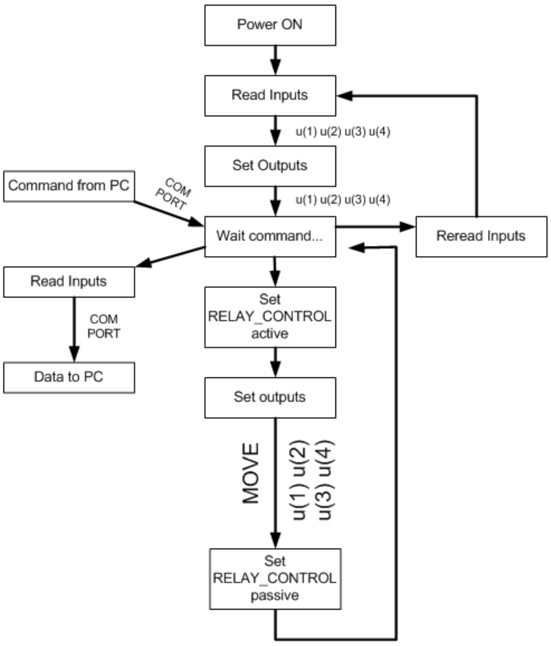

Fig. 4. Device control logic.

For power regulation we’ve used a Linear LT®1084 series positive adjustable regulator which was designed to provide a stable 5V 5A supply with a higher efficiency than the currently available devices from other makers. Our implementation is shown in Fig. 5.

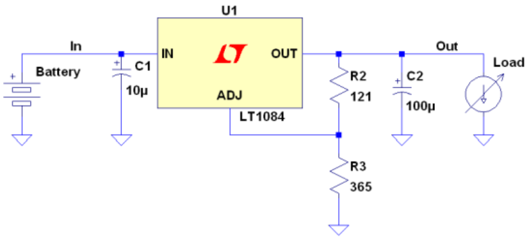

Fig. 5. Power regulator simulation model.

In order to get the regulator efficiently calculations and model the loads impacting the output voltage stability, we have used Linear LTspiceIV software [10]. Fig. 6 and Fig. 7 show the simulation running for 10 seconds and, at the second simulation second we simulate two seconds of movement from the Arduino microcontroller witch affected the load and device’s battery voltage changes.

From the simulation data we can describe the regulator’s efficiency in idle mode

$$\text{efficiency} = \frac{\text{usefull\_power\_output}}{\text{total\_power\_input}} = \frac{50.00 \ \text{mA}}{60.00 \ \text{mA}} \cdot 100\ \% = 83.33\% \ . \quad (1)$$

As well as the regulator’s efficiency on system control mode

$$\text{efficiency} = \frac{\text{usefull\_power\_output}}{\text{total\_power\_input}} = \frac{150.21 \ \text{mA}}{159.93 \ \text{mA}} \cdot 100\ \% = 93.92\ \% \ . \quad (2)$$

The final voltage regulator’s stability was really high and, on the input voltage change from 24V to 22V and load current change from 50mA to 150mA we got as low as 0.001V range disturbances.

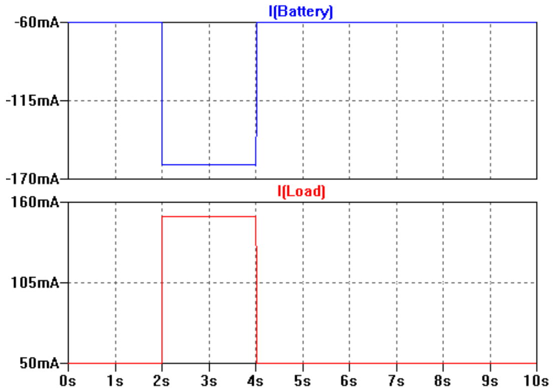

Fig. 6. Power regulator current drop characteristics.

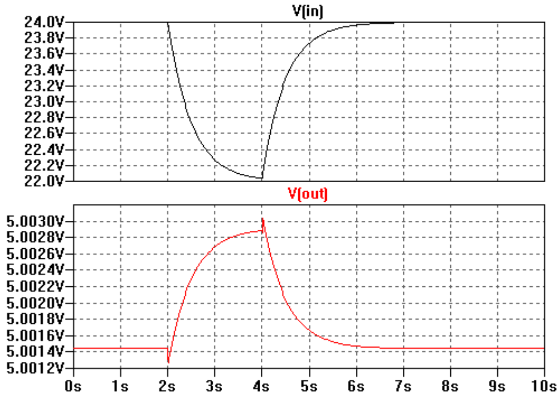

Fig. 7. Power regulator voltage characteristics on system load.

VI. Experimental Setup and Evaluation

The experimental investigation of our control method was done in indoors to overcome the variations in outdoor environmental surfaces. We have used 4 different loads (60 kg, 70 kg, 80 kg, 100 kg) to mimic the weight of real users. To measure the accuracy of the movement a trajectory was tapped on the floor and the device was driven via built-in controls as well as via Arduino and this data was checked versus the simulated trajectory from our model.

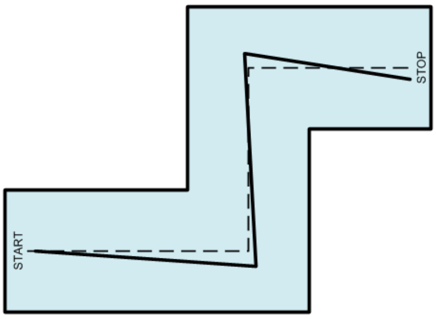

Fig. 8. Simulated and real output trajectory.

The “corridor” of the movement is shown in Fig. 8. The two lines mimic the trajectories. The one in dashes is the “ideal” (simulated) trajectory; the one in bold line is a measured, averaged trajectory. The achieved output control values are shown in Fig. 9. A less than 10 % deviation was noted in real-life movement and can be explained by a variation of user’s weight and also the position of front non-drive wheels.

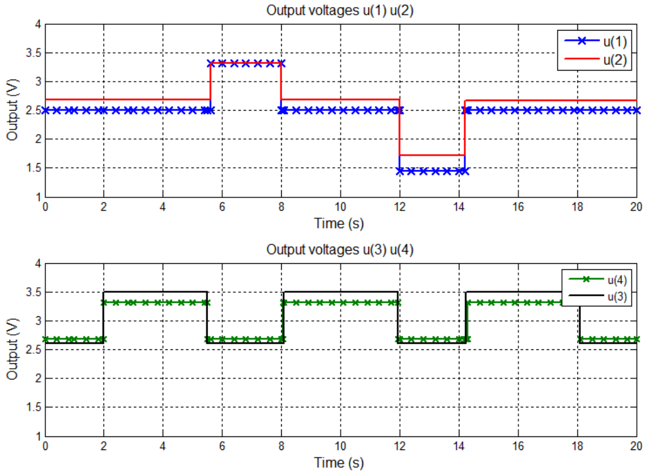

Fig. 9. Output control values.

VII. CONCLUSIONS

The solution offered in this work allows the simple and cost effective way of controlling the device's movement by adding an inexpensive microcontroller such as Arduino, allowing further expansion into the realm of automation and multimodal control.

An overall experimental investigation of real-life performance of our Arduino model proved quite successful. The added power regulator allowed achieving the stable outputs (±0.001V) to mimic the original controller. The deviation in trajectory compared to the ideal model was small (< 10 %) and can be further reduced by doing micro corrections, especially by adding the factors of human weight and checking the position of front wheels, and modifying the control values in real-time according to these parameters.

REFERENCES

[1] J. Pauwelussen, *Smart vehicles: for better and for worse.* Swets & Zeitlinger, 1995, pp. 451–462.

[2] A. Hurst, J. Tobias, "Empowering individuals with do-it-yourself assistive technology", in *Proc. of the 13th international ACM SIGACCESS conference on Computers and accessibility*, 2011, pp. 11–18. [Online]. Available: http://dx.doi.org/10.1145/2049536.2049541

[3] R. A. Cooper, "Intelligent control of power wheelchairs". *Engineering in Medicine and Biology Magazine*, no. 14, pp. 423–431, 1995.

[4] A. Liutkevičius, A. Vrubliauskas, E. Kazanavičius, D. Imbrasas, "Smart home services development, provisioning, and management framework", *Information technology and control*, vol. 2, no. 40, pp. 163–169, 2011.

[5] P. Killeen, "Developing a Smart Home System", in *Proc. of the International Conference on Embedded Systems and Applications*, 2011, pp. 148–152.

[6] K. Megalingam, N. Nair, M. Prakhya, "Automated voice based home navigation system for the elderly and the physically challenged", *Advanced Communication Technology (ICACT)*, pp. 603–608, 2011.

[7] S. Sangmeshwar, S. D. Joshi, N. Patil, S. D. Apte, "A voice activated multiprocessor embedded system to improve the control of a motorized wheelchair", *International Journal of Engineering Science and Technology*, vol. 2, no. 11, pp. 6812–6818, 2010.

[8] J. Z. Yi, Y. K. Tan, Z. R. Ang, S. K. Panda, "Microcontroller based voice-activated powered wheelchair control", in *Proc. of the 1st international convention on Rehabilitation engineering & assistive technology: in conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting*, *i-CREATe 07*, 2007, pp. 67–72.

[9] J. D. Warren, J. Adams, H. Molle, "Arduino for Robotics", *Arduino Robotics*, pp. 51–82, 2011. [Online]. Available: http://dx.doi.org/10.1007/978-1-4302-3184-4_2

[10] *Linear Software.* [Online]. Available: www.linear.com/designtools/software