

A Low-Cost Educational Project for teaching Embedded Systems

Albert Fazakas^{*}, Ali-Mostakhdemin Hosseini^{}, Jari Mustajärvi^{***}**

^{}Technical University of Cluj-Napoca,
26-28 Gh. Baritiu str, 400027 Cluj-Napoca, Romania,
tel +40-264-591340, fax +40-264-591340
Albert.Fazakas@bel.utcluj.ro*

*^{**}Innofors Research Centre,
Kehräämä, Wahreninkatu 11, 30100 Forssa, Finland,
tel: +358-3-646-2663 fax: +358-3-646-2848
amir@hamk.fi*

*^{***}Häme Polytechnic, Forssa Unit
Kehräämä, Wahreninkatu 11, 30100 Forssa, Finland,
tel: +358-3- 646 2653 fax: +358-3-646-2848
Jari.Mustajarvi@hamk.fi*

Abstract: In this paper, an educational project involving a microcontroller-based toy car is presented. The project is meant for students having basic programming courses passed, but being beginners in Embedded Systems. The meaning of the project is to provide a hardware and software background for the target group. The project is done through introductory exercises and given assignments to several student teams. Competition between different teams is stimulated. A set of laboratory exercises are also supported to make the learning process experiment-based.

Keywords: Toy car, Educational Project, Microcontroller, Embedded Systems

1. INTRODUCTION

It has been found that humans assimilate 10% of the knowledge from seeing, 20% from hearing, 50% from seeing and hearing, and 80% from seeing, hearing and practicing.[1]. Therefore, the key idea is the practice. Of course, to practice designing with embedded systems can be done on development or evaluation boards provided by manufacturers. But these development boards can be quite expensive. To insure a large variety of applications, these boards often use numerous peripherals and shared connections etc. This makes the schematics of the board for beginners difficult to understand.

In this paper another approach is proposed: At the beginning, the students do programming examples on a simple development board on software level, handling peripheral devices just as simple variables. The key idea is to increase their confidence to interact with the hardware boards, trying simple applications and seeing that “it works”. In the followings, the students are familiarized with the board schematic. After

that, they will do laboratory exercises involving intermediate level peripheral handling like using interrupts.

The project presented in this paper is one of the implementations of the “Small mobile robot” educational project, being very popular in many universities across Europe [2]. Almost each university has its own implementation. This particularization is due to the curricula differences between the various universities, different mechanical structures and different hardware implementations: The project basically consists in a toy car, usually, having two electrical motors. The motors are electronically controlled by a decision system, with several possible implementations, for example microcontrollers, DSP-s, FPGA-s etc. The decision system is running a program which drives the toy car according to some specific rules. One example is to follow a track highlighted on the floor, basing on the information provided by reflective infrared (IR) sensors.

The students involved in learning will do, as a project, a program which drives the toy car on the track to move as fast as possible. This approach stimulates competition and teamwork; on the other hand, using a toy car makes their learning sessions more enjoyable, increasing their level of interest in the material.

In this paper a microcontroller-based implementation is presented. However, the peripheral board can be used with other implementations also, for example FPGA-based or DSP-based ones, interfacing of the peripheral board is done with standard logic signals.

The project is to be applied to students at the Computer Science section of the Håme Polytechnic, Forssa Unit and to students at the Electronics and Telecommunications section of the Technical University of Cluj-Napoca. The students are assumed to be passed basic Analog and Digital Electronics courses and also C programming course, but they are beginners in Embedded Systems. The project is to be supported by presentations of microcontrollers, including facts about the CPU and the embedded peripherals. The student’s role is to apply the theory learned at courses to the car project, following the laboratory exercises. Their final assessment is to make the program and download it to the embedded microcontroller. The program algorithm has to follow the rules established by the teacher. An example is the competition approach, making car races between students or student teams.

2. DESCRIPTION OF THE DESIGN

The schematic block of the design is presented in Figure 1. A toy car having two DC motors, from which one is used for movement (forward and backward) and the other one for turning the front wheels (left or right) is assumed.

The design is divided to two boards: a decision board and a peripheral board, in order to be able to replace the microcontroller implementation of the decision board with another one. The DC motors are driven with L293D power drivers from ST Microelectronics, in a bridge-connected configuration [3]. All of the signals connecting the two boards are 5V TTL compatible logic signals.

For the decision board, a simple 8-bit microcontroller from Motorola’s HC08 family, specifically a QY4 device was used. Basing on the Evaluation Board description from Motorola [4], the microcontroller is programmed to contain the internal User Mode Monitor Access [4] program. Using this feature, program downloading and online debugging can be done without any expensive external programmer [5], just a serial cable and an RS232 level shifter is needed. This reduces drastically the costs of

manufacturing more than one board, taking into account that several student teams are assumed to work simultaneously at the same project.

The interconnecting signals and their purpose are shown in Table 1.

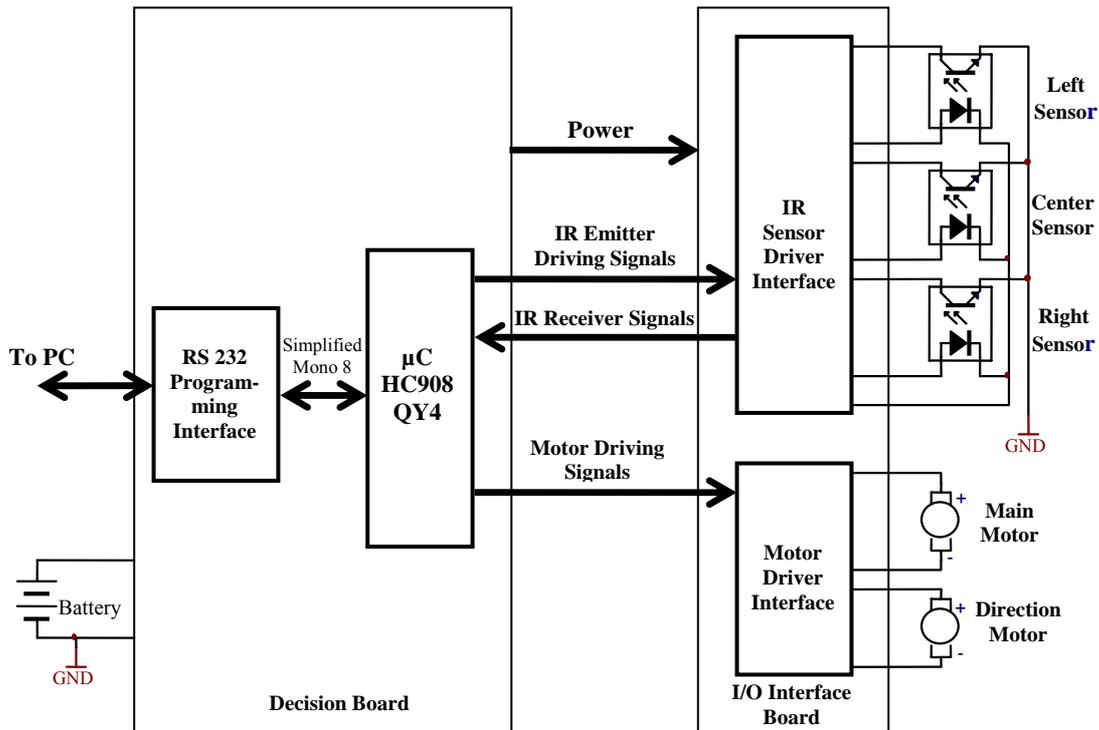


Figure 1. Schematic block of the design

Signal Group	Signal name	Signal type (I or O) and active level (viewed from the peripheral board)	Initial Status	Corresponding bit variable (denoted as in the development environment)
IR Emitter Driving Signals	P_EN	I, active-high	L (PD)	
	SUPPLY_P_EN	I, active-high	H (PU)	
	RL_P_IN	I, active-low	H (PU)	
	CL_P_IN	I, active-low	H (PU)	
	LL_P_IN	I, active-low	H (PU)	
	IR_SHDN	I, active-high	L (PD)	PTB_PT B4
IR Receiver Signals	RL_S	O, active-low		PTB_PT B7
	CL_S	O, active-low		PTB_PT B6
	LL_S	O, active-low		PTB_PT B5
Motor Driving Signals	MM_EN	I, active-high	L (PD)	PTA_PTA1
	MM_PT	I	H (PU)	PTB_PT B2
	MM_NT	I	H (PU)	PTB_PT B3
	DM_EN	I, active-high	L (PD)	
	DM_PT	I	H (PU)	PTB_PT B0
	DM_NT	I	H (PU)	PTB_PT B1

Table 1. The logic signals from the I/O board and the signal description and location. I and O means input and output, PU and PD means pulled-up and pulled-down, respectively

Note that the chosen microcontroller does not have enough number of pins to drive all of the signals. For example, the IR sensors can be driven in pulse-mode or in continuous current mode, but in this design only the continuous current mode is considered. That is the reason why the IR driving signals in pulse mode, P_EN, SUPPLY_P_EN, RL_P_IN, CL_P_IN, LL_P_IN and the MM_EN signals are unused by the decision board, their usage being left for further developments

Although just two signals are required for driving a DC motor in the two opposite directions, three signals were included to provide support for driving the motors using PWM. The main motor driving signal logic is presented in Table 2.

The active level for the IR receiver signals is considered for IR signal sensing, i.e. when IR is reflected The IR sensors could be so sensitive that they “see” any color of the floor except black. So the track color should be black on a light colored floor, or light colored track on a black floor. It can be used, for example, a duct tape. Considering black tape, if a sensor meets the track, will indicate logic level “1”, which means inactive level according to Table 1. This consideration has to be taken into account at the software programming.

MM_EN	MM_PT	MM_NT	Action taken	
0	X	X	The motor is stopped (free running mode)	
1	X	=MM_PT	The motor is stopped (break mode)	
1 or PWM	1	0	The motor moves forward	Applying PWM signal to MM_EN, the speed of the motor can be adjusted
1 or PWM	0	1	The motor moves backward	

Table 2. Truth table of the Main Motor driving signals

In order to drive and read the logic signals, the General purpose I/O ports were used in the specific microcontroller, except the MM_EN motor driving signal, where a shared I/O/PWM pin is used [MOTO_DATASHEET]. This enables PWM driving of the main motor. An image of the toy car running on the track is presented in Figure 2.

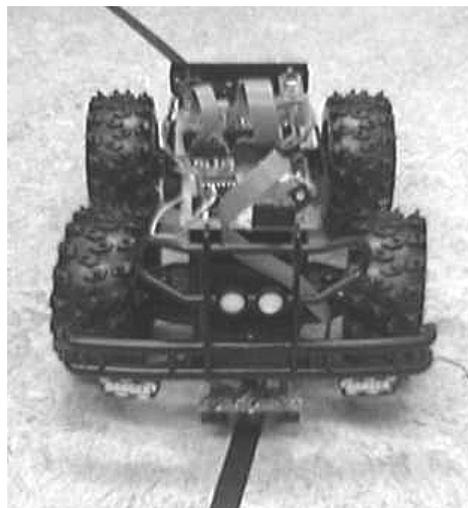


Figure 2. The toy car running on the track.

3. DEVELOPMENT SOFTWARE AND EDUCATIONAL CONSIDERATIONS

Concerning HC08 microcontrollers, several low-cost or free development tools are available on the Internet, [5], [6]. These software tools are limited, for example, to a maximum amount of 4Kbytes of program compiled from C language. Although C language is very popular for microcontroller application development [7], the students are encouraged to use projects combining both C and Assembly sources.

Several software examples will be provided to the students to understand step-by-step the way how they have to use the peripherals to move and turn the car, how to read the sensors and how to make as small as possible code. They will run the software applications both in debug and run-time mode, to get familiarized with debugging of a program made for Embedded Systems.

The succession for the laboratory exercises is presented below. Note that this is not an order of the laboratory session, for example, parts 1.) to 3.) can be done in a single session, but part 7.) can take two sessions

1.) Get familiarized with the software environment. Learn how to create a new project. Open an existing project and debug it using simulation

2.) Get familiarized with the block schematic of the board. Understand the purpose of the variables and how to set them to achieve the desired action.

3.) Learn to download a program to the board, i.e. to move the car forward.

4.) Get familiarized with the physical board schematics. Follow exercises like applying manually logic levels to the signals and following the movement of motors, or measuring logic levels of the sensors.

5.) In order to understand the timer and the way as PWM signal is generated, analyze, download and run an application which makes use of the timer, like blinking the onboard LED when the center sensor sees the track.

6.) Debug an application having errors (intentionally left)

7.) Create an application which makes the car to move in different ways, like move forward during 3s, after that turn to left during 2S etc, to learn how to use the timer and create state machines using programming

8.) Assessment: Basing on the knowledge above, create, simulate and debug a program which drives the car along the track, following a very simple basic algorithm (which makes just decisions of turning to the left or right basing on the information provided by the IR sensors).

9.) Assessment as teamwork: Improve the basic algorithm above, in order to solve the following problems (assume that the shape of the track is unknown by the students at the moment of programming):

- Some curves on the track are having small rays which cannot be followed by the mechanics of the car. The car will loose the track in these curves. Solve the problem from software to make the car able to return to the track.

- Reduce the speed at curves to avoid slipping out of the car from the track.

- Avoid or reduce oscillations of the car when moving among the track as much as it is possible

- Keep logging information about the track using the internal RAM in order to “learn” the most difficult parts of the track

- A contest will be organized between teams. The winner is the team which makes the car following the track respecting the rules above in the smallest time.

Obviously, succession and content of the exercises can be changed according to the type of the implementation, which is not limited just to a HC08 microcontroller. On the other hand, the content can be changed easily to be in accordance with the theoretical presentations also.

4. CONCLUSIONS

In this paper, a low-cost educational project for teaching embedded systems was presented. The cost of the project is low due to the use of an embedded programming interface and free software. An implementation example was also presented, using a HC08 microcontroller from Motorola. Obviously, this is not the only possible implementation of the design, the toy car and the peripheral board is able to support different type of embedded systems.

Parts of the project were experimentally applied to two of the exchange students at the Håme Polytechnic, Forssa. The students found the project more attractive than the exercises on a development board.

In further development of the project, as a next level step, a more complex design is considered, having two microcontroller boards communicating to each other, to make more use of the teamwork. Standard communication interfaces like SCI, SPI and I2C are also considered, to make the students familiarized with these interfaces. Obviously, microcontroller-FPGA or FPGA-FPGA combinations can be included. The number of sensors providing information about the car will be increased, sensors like speed meters and shock sensors are also considered. The further target is a multidisciplinary project, involving a wireless communication interface with the car like BlueTooth™.

4. REFERENCES

1. Eleftherios A. Kayafas, Albert Fazakas, (1998), *Education in Electronics – A Multimedia Implementation*, Proceedings of the 6th International Conference on Optimization of Electrical and Electronic Equipments OPTIM '98, Braşov, May 14-15, 1998, pp. 589-594
2. Université de Savoie, (2003), *Concours Robotique GEII*, <http://www.geii.univ-savoie.fr/geii/robots/robots.htm>
3. ST Microelectronics, (2003), *L293D Datasheet*, ST Microelectronics, <http://www.st.com>.
4. Jim Sibigroth, (2003) *User Mode Monitor Access for M68HC908QY/QT Series MCUs*, Application note AN2305, Rev 1, 3/2003, Motorola Semiconductors
5. P&E Microcomputer Systems Products, <http://www.pemicro.com>
6. Metrowerks Products, <http://www.metrowerks.com>
7. Javier Gutierrez, (2001), *HC08-The C-Language ready MCU*, Presentation, Motorola Semiconductors, <http://www.motorola.com>