

Bridging the Gap between an Amateur and Robotics by Developing Interactive Embedded Software to Automate Microcontroller Boards

Siddharth Shrotriya¹ and Animesh Pandey^{2 +}

¹ Department of Electronics and Communication, Jaypee Institute of Information Technology, Noida - 201307, India

² Department of Information and Technology, Jaypee Institute of Information Technology, Noida -201307, India

Abstract. We all know that every student uses AVR or Arduino for their micro-controller project either in college and schools all around the globe. For that purpose, students buy an arduino board and download the Arduino software for coding and then connect both of them. We have tried to make this interface more user-friendly and robust. A user interface which can be used by a 4th standard student as well as a college graduate. It is implemented to the AVR board built on the schematic of Arduino Uno which has a 28 pin Atmel IC. We have made the board in such a way that it is interfaced to a double amount of analog and digital peripherals as compared to the original arduino board. The UI gives the functionality of easily coding the use of peripherals as well as the internal clocks and timers of the board. The UI is built on .Net platform and it generates the code for the micro-controller in embedded c language and creates the hex file according to the selected boot-loader. The code generation is done by using the concepts of Artificial Intelligence which consists of understanding the choice of users by dynamically changing the style of the UI. The prototypes of the UI are tested using the heuristic evaluation, an evaluation technique of HCI (Human Computer Interaction). The schematic of the board has been built on the eagle platform and it can be used with any AVR and PIC boot-loader. Both the hardware and software are open source, platform- independent, robust and user friendly.

Keywords: Artificial Intelligence, AVR Architecture, Code Generation, Dot Net, Human Computer Interaction, Micro-controller, PIC Architecture and User Interface.

1. Introduction

User interfaces became a very important part of software development at the commencement of the 21st century. Software compatibility with the target user becomes as necessary as with an operating system or an application platform. We have implemented the same thing for robotics. It is very hard for an amateur student or a young enthusiastic learner to start developing his desired robots. He has to gain knowledge of required hardware and then the complexity of coding in the required software. To decrease the effort of learning for a beginner, we have developed interactive embedded software which will be used to generate the code and upload it to the desired hardware. The software is compatible with development boards which have micro-controllers with integrated Arduino Decimilla, Uno or Mega Boot-loaders. It supports micro-controllers which have Atmel 28 pin DIP. These types of chips have limited input output pins for interfacing peripherals. We have made use of the learning capability of the software to predict the usage of the target user and can be interfaced to double amount of peripherals. The user interface is built in such a way that it attracts and interacts with the user to assist them in developing the code for their robots. The UI changes its

⁺ Siddharth Shrotriya. Tel.: +91 8800723941
E-mail address: siddharth@ieee.org
Animesh Pandey. Tel.: +91 9953504244
E-mail address: animeshpandey@acm.org

style as per the preference and usage pattern of the user. It also reflects the peripherals used or available for the current application and their corresponding interfaced pins on the hardware.

The software is built on .Net framework using C# language and Visual Studio 2010 with the help of Arduino IDE. A compatible hardware board is also made which contains all the interfaced peripherals and the Atmel micro-controller with ISP programmer and debugger. The hardware schematic is built on Eagle platform and Microsoft Robotics Developer Studio. A user can select the input sensors, output electronic devices and the desired controlling pattern in the software. A tree is generated which will display the order and features of the selected components. User can generate the code in embedded C language and can directly upload to the hardware using USB of the computer. The hardware will perform the desired functions selected by the user in attempt of making their own versatile robots.

2. Description of the Software

2.1. Development of the Software

For developing the software we follow the Waterfall Model[1][3]. Software modeling consist of many steps, first of which is known as prototyping. In this, we make numerous designs and then evaluate them according to the specifications. One of our prototypes is shown in fig1. The main steps in prototyping are as follows:

- Setting the components
- Sharing with people involved
- Environment testing
- Task being accomplished
- Sequence of the items
- What steps are involved?
- What leads someone to use the app?
- What task is being illustrated?
- Satisfaction of the customer
- What's motivates people to use this system?
- What does it enable people to accomplish?

An important feature to be noticed is Holistic focus which helps to emphasize how an interface accomplishes the given task.

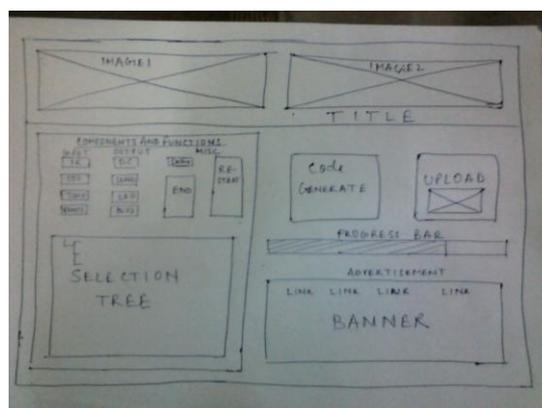


Fig. 1: Sample Prototype

We have shared our prototypes with many students in our college and then take their feedbacks to include the missing features of the model. Fig2 shows how our interpretation gets generalised by the comments of the students. We changed our designs according to the modifications suggested by the assessing people.

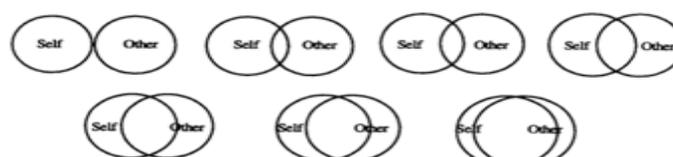


Fig. 2: Demonstrating the Process of Model Generalisation

After the assessment of all the prototypes, we finalise our model^[4] and build the software using the windows application package of Visual Studio 2010 and the dynamic behaviour is incorporated using C# language and many inbuilt libraries. We tried our best to include as much functionality as we can. The main features of the software are listed in the below mentioned software requirement specifications (SRS). A SRS is a specification for a software system – is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software.

2.2. Characteristics of the Software

Software Requirement Specifications^[4]

- Introduction
 - Purpose: To develop embedded software for beginners and aspirants of robotics.
 - System overview: It is a Windows application capable of running on Windows XP, 2000, Vista and Windows 7 platform.
- Overall description
 - Product perspective: Developed for beginners so it is informal in appearance.
 - Product functions: It allows user to select input and output peripherals in desired format and then generate the code and upload directly to the hardware.
 - Constraints: It can be used for incorporating maximum of 20 functions per peripheral interfaced with hardware and displayed in the software.
- Specific requirements
 - External interface requirements: Dot Net 2.0 framework and Windows OS platform.
 - Functional requirements: Working electronic components.
 - Performance requirements: To be refreshed after every successful run.
 - Design constraints: Less graphical but faster than any other software.
 - Other requirements: Arduino compatible micro-controller boards.

After incorporating all the required features, the application is build and tested on various platforms. Fig3 shows the main window of the final software.



Fig. 3: Fundamental Window of the Software

The software consist of the following list of windows for peripheral interfacing: CDS sensor Dialog Box, IR Sensor Dialog Box (Fig4), Touch Sensor Dialog Box, Remote Control Box, DC Motor Box (Fig5), Servo Motor Box, LED Dialog Box, Buzzer Box and Delay box. It has buttons to end the current program, start a new program, generate the code and Upload the code to the hardware via USB.

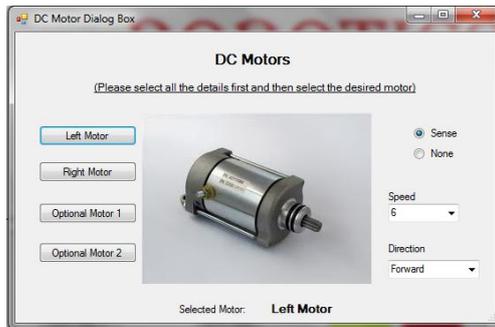


Fig. 4: DC Motor Dialog Box

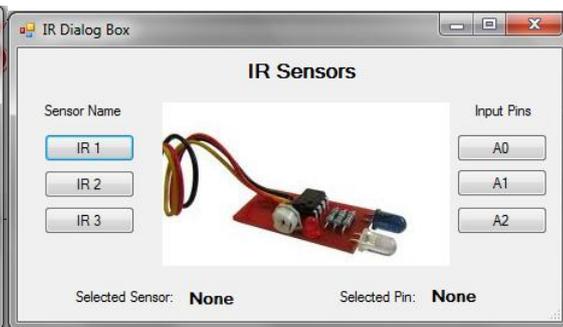


Fig. 5: IR Dialog Box

It allows user to interface 4 DC Motors, 4 Servo Motors, 2 LEDs, 1 Buzzer as output peripherals to the micro-controller. The input components which can be interfaced are 3 IR Analog Sensor, 2 CDS Sensor, 2 Touch Sensors and a RF based Wireless Remote Module with 8 buttons. The I/O pins used are shown separately and can be reused for other components. This feature is added using a repetitive analysis of the usage of the software and an application of artificial intelligence is shown which helps the feature to be incorporated in the software. User can add delay component in milliseconds, timer function using the internal 8 or 16 bit auto reload timers of the microcontroller and looping functions for functionality repetition in the program whose code will be generated in the embedded C language. After program completion, a graphical tree (Fig6) consisting of the components used and their functionality is shown to the user for analysing the work to be done by the Robot built using the microcontroller as an artificial brain. The code is then verified, converted to .hex file and uploaded to the hardware using the USB port of the PC and a USB cable. The ISP debugger of board interprets the code, converts it in machine language and tells the microcontroller what to do with the connected peripherals.

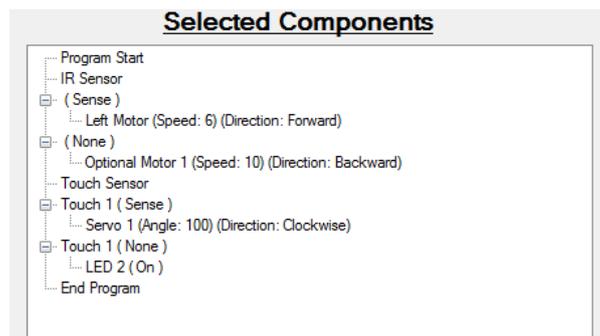


Fig. 6: Graphical Tree of the selected components

3. Description of the Hardware

3.1. Development of the Hardware

The hardware circuit is built in the eagle platform using the features of its capture development. It is designed in such a way that it becomes compatible with the code generated through the software. The schematic is then converted to a brd file which has layout of the circuit to be incorporated in PCB^{[6][7]} fabrication of the board. The routing process is initiated by the auto routing feature of the eagle software. The errors are then corrected by manual routing of each layer of the layout to give accurate spacing for every component, wires, switches and headings. The final brd file is shown in Fig7 which is converted to the gerber file. Gerber is the file extension used for the final image extraction to be imprinted on the PCB.

The circuit of the board consist of Atmel AVR 28 pin microcontroller IC, I/O holders for peripherals to be connected to the microcontroller, holder for the motor driving L293d IC, F232 IC used as debugger, power and reset switch, 3.3 V, 5V and 12V power supply sockets, USB holder and connecting wires. The brd file consists of the circuit components placed in a sequential order. The high voltage connections are made strong and there is spacing for headings and mounting holes. The placing is done in such a way that the wiring does not intersect and the input and output holder are placed in one collection. This helps to decrease the PCB dimension to 3.5'' X 2.5''. The PCB is turned into Printed Circuit Assembly by photographic

process followed by hole-drilling, solder plating and lamination. The final board is connected to the required peripherals to be used by cod developed in the software.

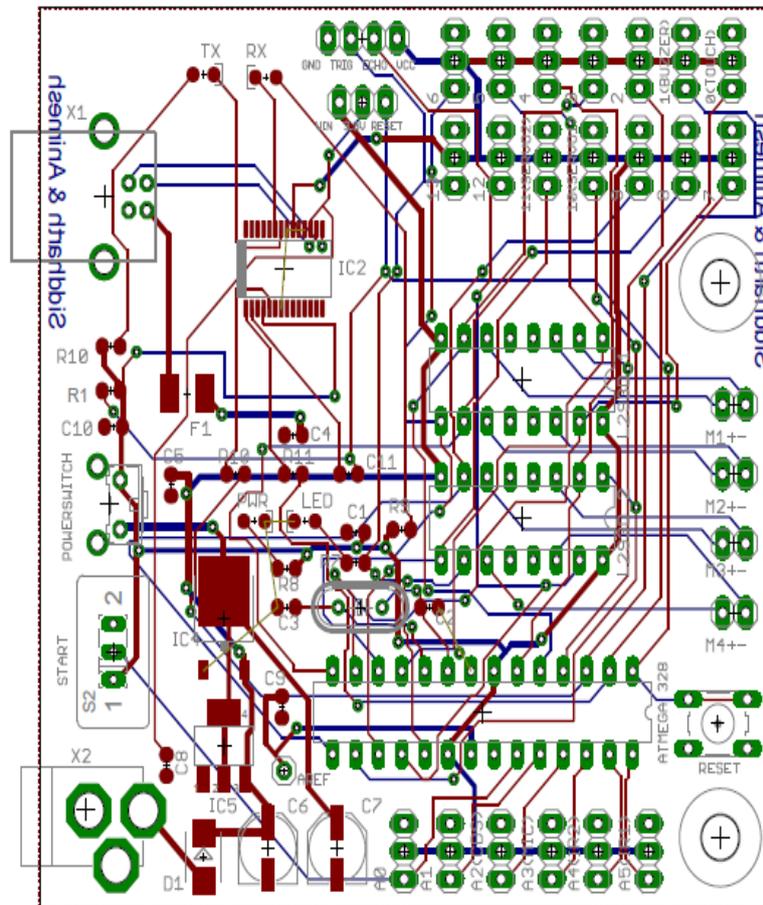


Fig. 7: Layout of the hardware schematic

4. Conclusion

This implementation of software and hardware is done specially for the beginners of Robotics who has very little knowledge of embedded systems or coding in embedded c language. We hope students find the software interactive enough to use it every time for robot development. We also hope that hardware is compact and robust enough to be used with the software for connecting the peripheral components for the development of good and interesting embedded applications.

5. References

- [1] Jonathan Arnowitz, Michael Arent, Nevin Berger “*Effective Prototyping for Software Makers*”
- [2] Debbie Stone, Caroline Jarrett, Mark Woodroffe “*User Interface Design and Evaluation*”
- [3] John M. Carroll “*HCI Models, Theories and Frameworks: Towards a Multidisciplinary Science*”
- [4] Hassan Gomaa “*Software Modeling and Design*”, 2010
- [5] Tamara Taylor and Thomas A. Standish “*Initial thoughts on Rapid prototyping techniques*”, 1982
- [6] R.S. Khandpur “*Printed Circuit Boards*”, 2005
- [7] Robertson, Christopher T. “*Printed Circuit Board Designer’s Reference: Basics*”, Prentice Hall, 2003