

Interfacing a DC Motor with ARM Cortex M3 Microcontroller LPC1768

Md. Moyeed Abrar, Assistant Professor, Department of Computer Science & Engineering, Khaja Banda Nawaz College of Engineering, Kalaburagi, Karnataka, India, moyeed.abrar@gmail.com

Abstract- The ARM cortex M3 processor is the need of the hour as it has many salient features which enable the user to use in a wide range of applications in the embedded domain. The ARM cortex M3 processor is ideally suited for low cost microcontrollers which are used most commonly in consumer products such as electrical appliances, toys, ovens and mobile phones. Its lower power consumption, high performance and ease of use advantages enables the embedded developers to switch towards the 32 bit systems and develop quality based products with the flexible ARM architecture. The state of the art presented in the paper is the interfacing of DC motor with Arm cortex M3 microcontroller LPC1768. The result of this interfacing is the direction control of a DC motor, which means that it is enabled to rotate in clockwise and counterclockwise direction using the ARM cortex M3 microcontroller LPC1768.

Keywords — Arm cortex M3 microcontroller LPC1768, 32 bit systems, DC motor, direction control, high performance, interfacing, low power consumption.

I. INTRODUCTION

ARM cortex M3 processor based microcontrollers, programming can be done with ease using the C language and have a well-established architecture due to which application code can be ported and easily reused, which in turn reduces the development time and testing costs. ARM series of processors in the last few years have developed new processors and system blocks. Various architecture versions of ARM family include ARMv4T, ARMv5E and the ARMv6. Recently, it has been seen that the ARM has extended its product portfolio by diversifying its CPU development, which resulted in the architecture version 7 or v7. In this version the architecture design is divided into three profiles namely A profile, R profile and the M profile.

- The A profile is designed for high performance open application platforms.
- The R profile is designed for high end embedded systems which requires real time performance.
- The M profile is designed for deeply embedded micro controller type systems.

The ARM cortex M3 processor has ARMv7 architecture. With reference to programming as far as the instruction set is concerned the cortex M3 processor supports only the Thumb 2 (and traditional Thumb) instruction set. Instead of using ARM instructions for some operations as in traditional ARM processors, it uses the Thumb 2 instruction set for all operations. Thus it can be said that the cortex M3 processor is not backward compatible with traditional ARM processors [1], [2]. The cortex M3 processors can execute

almost all the 16 bit Thumb instructions including all 16 bit Thumb instructions supported on ARM 7 family processors making application porting easy. With support for both 16 bit and 32 bit instructions in the Thumb 2 instruction set, there is no need to switch the processor between Thumb state (16 bit instructions) and ARM state (32 bit instructions). Hence, in the cortex M3 processor, one can mix the 32 bit instructions with the 16 bit instructions without switching state, getting high code density and high performance with no extra complexity [1], [2]. The thumb 2 instruction is considered as one of the important feature of the ARMv7 architecture. The cortex M3 instruction set has a wide range of new features such as the hardware divide instruction and a number of multiply instructions are also available on the cortex M3 processor to improve data crunching performance. The cortex M3 processor also supports unaligned data accesses, a feature previously available only in high end processors.

In this paper the interfacing of a DC motor with Arm cortex M3 Microcontroller LPC1768 is presented. The rest of the paper is organized into sections as follows: section II describes the overview of DC motor. Section III focuses on the system design. Results and discussion are reported in section IV. Finally section V summarizes the paper and presents the concluding remark.

II. DC MOTOR OVERVIEW

A DC motor is an Electrical machine which converts Electrical energy into Mechanical energy. It works on the principle that whenever a current carrying conductor is placed in a magnetic field, it experiences a mechanical force

whose magnitude is given by

$$F = BIL \text{ Newton.} \quad (1)$$

Where

B is the Magnetic flux density in wb/m^2

I= current flowing through the conductor in Amperes

L= Length of the conductor in meters

The direction of the mechanical force experienced is given by Fleming's left hand rule, which states that when the thumb, forefinger and middle finger of the left hand are outstretched such that they are mutually perpendicular to each other, then the forefinger points in the direction of the magnetic field, the middle finger in the direction of the current, then the thumb indicates the direction of the force experienced by the conductor. The working of a DC motor is illustrated in figure 1.

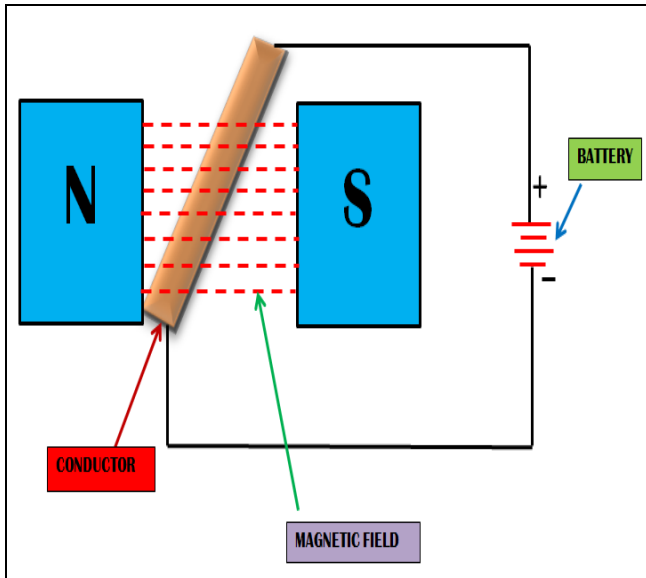


Figure 1 Working principle of a DC motor

The voltage equation of a DC motor is given by

$$V = E_b + I_a R_a \quad (2)$$

Where V is the applied voltage, E_b is the back emf, I_a is the armature current and R_a is the armature resistance.

In a DC motor, the electrical input (supply) voltage is the main cause of armature current and the motoring action and hence this induced emf (electromotive force) opposes the supply voltage. This emf tries to set up the current through the armature which is in opposite direction to it. As this emf always opposes the applied voltage it is also called as back emf or counter emf.

From the voltage equation, the armature current can be calculated as follows

$$I_a = (V - E_b) / R_a \quad (3)$$

The main output of the DC motor is torque. It is defined as the turning moment about an axis. It is also defined as the product of force and perpendicular distance.

The torque equation of a DC motor is given by

$$T_a = (1/2\pi) \times \Phi \times Z \times (P/A) I_a \quad (4)$$

Where T_a is the armature torque or gross torque developed by the motor, Φ is the useful flux per pole in webers, Z is the total number of armature conductors of a DC motor, P is the number of poles and A is the number of parallel paths.

The speed equation of a DC motor is given by

$$N = V / K \Phi \quad (5)$$

Where K is a constant.

In many applications that uses DC motor, the speed control is of utmost importance and it is essential where the precision and protection is of prime essence. A DC motor's speed can be controlled over a wide range, by using either a variable supply voltage or by changing the strength of current in its field windings [4], [5]. But in this paper the direction control of DC motor is presented.

The direction control of a DC motor, which is making the DC motor to rotate in clockwise and anticlockwise direction by 'N' steps, is implemented using the advanced microcontroller the ARM cortex M3 LPC1768. The DC motor which is used for the proposed research is illustrated by the photographic view as shown in figure 2.



Figure 2 Photographic view of DC motor

III. SYSTEM DESIGN

3.1 Board Specifications of the ARM cortex M3 Microcontroller LPC1768

The LPC1768 is ARM cortex M3 based 32 bit Microcontroller. It has the following features

- 512KB flash memory and 64KB SRAM. In-System Programming (ISP) and In-Application Programming (IAP) capabilities.
- It has Single 3.3 V power supply (2.4 V to 3.6 V).
- There are 70 General Purpose I/O (GPIO) pins with configurable pull-up/down resistors, open drain mode, and repeater mode.
- Possesses a 12 bit Analog to Digital converter (ADC) and up to 8 analog channels.
- It has 10-bit Digital-to-Analog Converter (DAC) with dedicated conversion timer.
- Easy communication set up with the availability of 12 MHz crystal.
- Possesses a Reset push-button for resetting the controller.
- Four general purpose timers/counters are available with a total of 8 capture inputs and 10 compare

outputs.

- Possesses 3 I2C bus interfaces.
- There is a serial peripheral interface (SPI) controller with synchronous, serial and full duplex communication.
- Sleep mode, deep sleep mode, power down mode and deep power down mode are the four reduced power modes available.
- Real time clock (RTC) with a separate power domain is also available.
- Presence of an onboard voltage regulator for generating 3.3 volts. Input to this is through the external +5volt DC power supply through a 9 pin DSUB connector.
- Piggy back module is available which contains the LPC1768 controller.
- Presence of one RS232 interface circuit with 9 pin DSUB connector.
- Possesses DC motor interface for direction and speed control.
- Availability of stepper motor interface with direction and speed control.
- It has an 8 bit DAC interface.
- There is a 16x2 alphanumeric display for LCD interfacing.
- Presence of a 4x4 matrix keyboard for keyboard interface which is connected to the port lines of the controller.
- Possesses a two digit multiplexed seven segment display interface.
- On board buzzer, relay and LED indication controlled through push button interface circuits are also available.
- It has a 2 channel ADC IC with POT and temperature sensor.
- Standard 26 pin FRC connectors are available to connect to on board interface.

3.2 System Specifications

The system specifications are illustrated in table.

Sl. No	System Specifications
1.	Domain: Embedded systems Microprocessors and Microcontrollers
2.	ARM cortex M3 Microcontroller: LPC1768 32-bit RISC microcontroller from NXP founded by Philips.
3.	DC Motor : 5 volts, 100 mA
4.	Desktop computer: Pentium 4, 1 GB RAM, Processor speed 2.5 GHz.
5.	Programming language: Embedded C
6.	Port line: P.124 & P0. 26
7.	Software: Keil μ vision-4
8.	Adapter cable, USB cable

9.	In-system Programming (ISP): Flash magic software can be used to download the HEX files to the flash magic of the controller.
10.	Serial communication: RS 232 cross cable connections required for establishing communication between the evaluation board and a display terminal/host computer.
11.	Applications: Rotation of DC motor in clockwise and anticlockwise direction.

3.3 System set up

The system set up was done in Embedded system and Microcontroller laboratory. The system consists of a Desktop computer, ARM cortex M3 LPC1768 evaluation board, and a DC motor. The complete interfacing is illustrated by the photographic view in figure 3.



Figure 3 Photographic view of the system

The required hardware connections were done by connecting the adapter cable to the socket and the adapter pin was connected to the ARM cortex M3 LPC1768 evaluation board. Apart from this, the USB cable was connected to the USB port of the desktop computer and the other end of the USB cable was connected to the female connector of the RS232 cable and the male connector of the RS232 cable was connected to the female connector provided on the ARM cortex M3 LPC1768 evaluation board. The DC motor was interfaced to the board by connecting it to the reliamate RM4 provided on the board. Embedded C program was written for the interfacing of DC motor to run it in clockwise and anticlockwise direction using Keil μ vision-4 software. This software was opened by double clicking on the icon of the software located on the desktop computer. A new window appeared where project option was clicked and then new μ vision project was chosen as shown in figure 4.

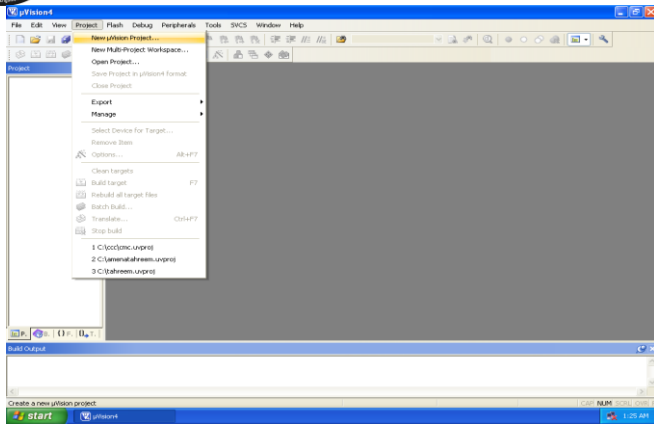


Figure 4 Choosing the new μ vision project

A new window appeared on the screen named Create New Project [6]. A separate folder named DC MOTOR was created on the desktop as shown in figure 5

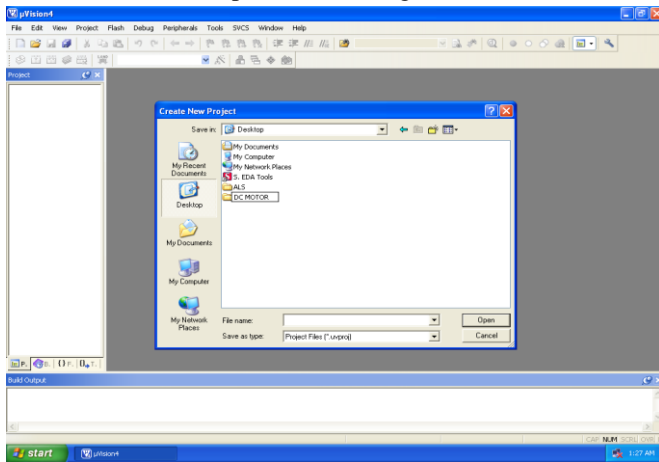


Figure 5 Creating the folder DC motor on the desktop

A file named CONTROL was saved in this folder as depicted in figure 6

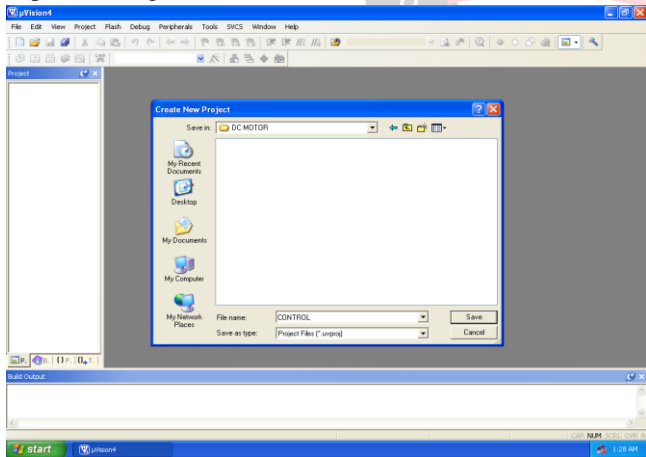


Figure 6 Naming the file CONTROL

Another new window named Select Device for Target 'Target 1' appeared on the screen and in its search column nxp was written to select the microcontroller of the nxp series founded by Philips [6]. The LPC1768 ARM cortex M3 microcontroller was selected and the features of this microcontroller was displayed and then ok option was clicked as shown in figure 7.

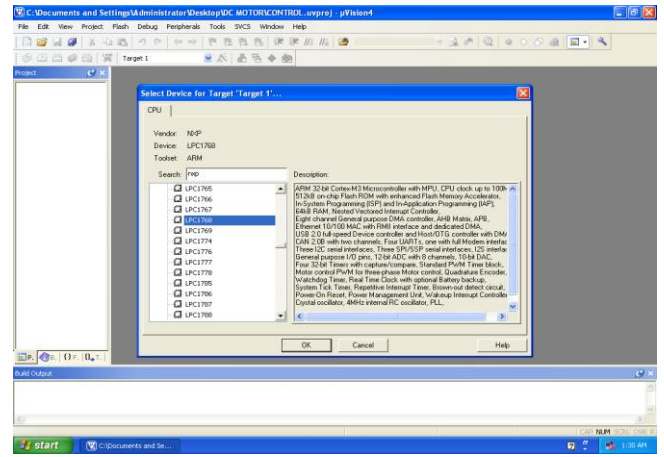


Figure 7 Choosing the ARM cortex M3 microcontroller LPC1768

As soon as ok option was clicked a new window named μ vision appeared with two options Yes and No and then the option YES was chosen to add the startup_LPC17xx.s as illustrated in figure 8.

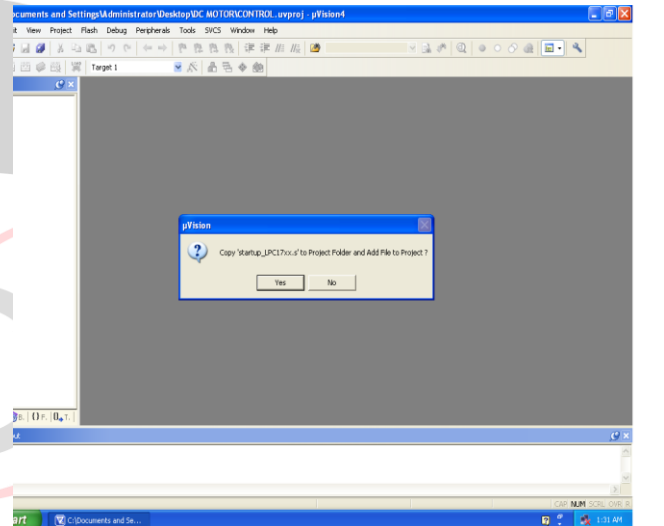


Figure 8 Selecting the Yes option

The target was created as 'Target 1' in the project window. The + sign of target was clicked as result of which the source group was shown below the target as illustrated in figure 9.

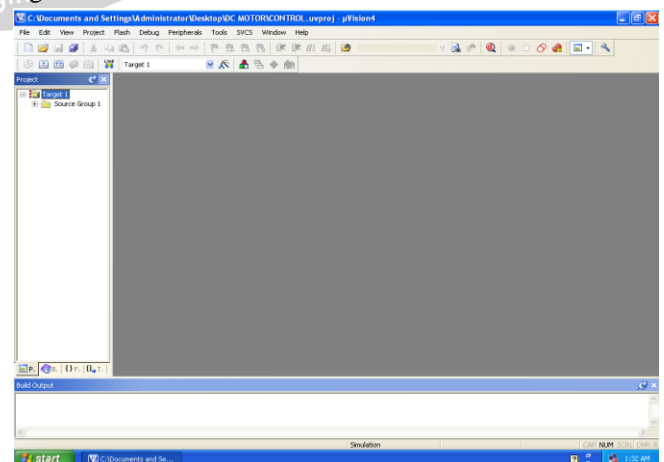


Figure 9 Creation of source group 1

After the creation of project a new window appeared where the file option was chosen and new option was selected as a result of which editor window was opened as shown in figure 10

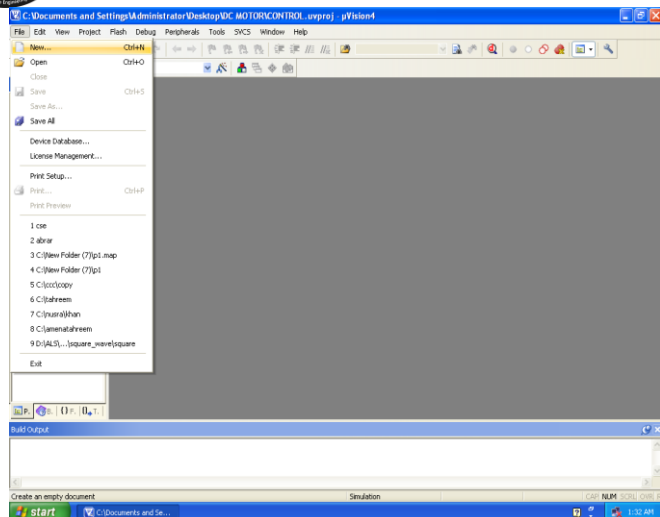


Figure 10 Choosing the new file

The program for the interfacing of DC motor to rotate the motor in clockwise and anticlockwise direction was written in embedded C language and then it was saved by clicking on the save icon as illustrated in figure 11.

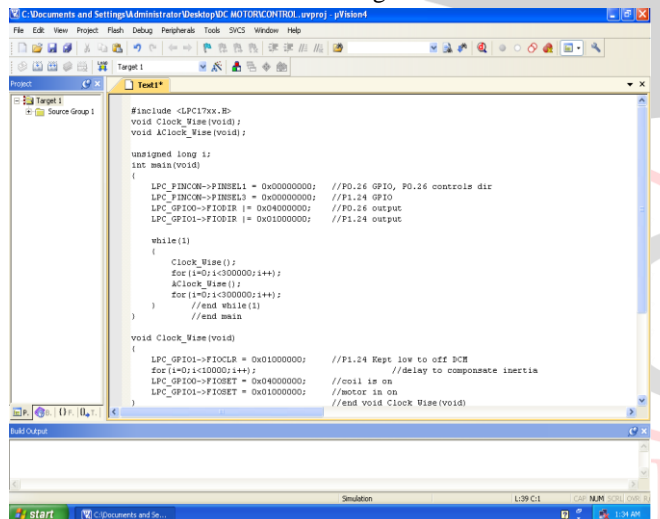


Figure 11 Saving the program

Another window named Save As appeared on the screen and the file was saved as CONTROL.c. The file extension .c is mandatory. This is shown in figure 12.

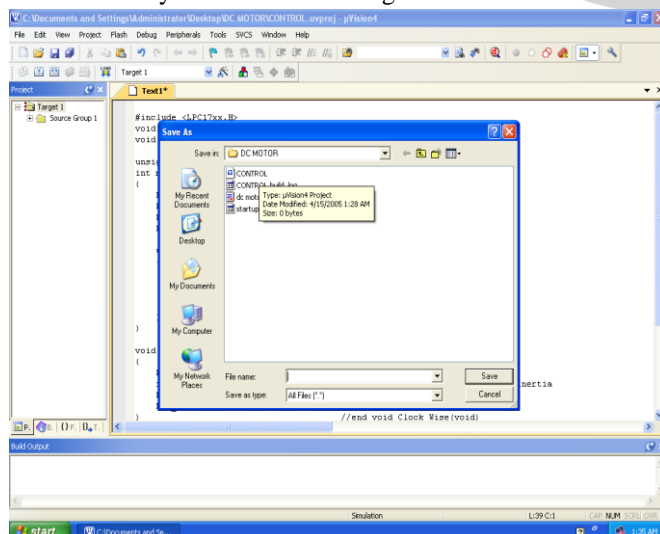


Figure 12 Naming the file as CONTROL.c

As the file CONTROL.c was saved color syntax

highlighting was enabled as shown in figure 13

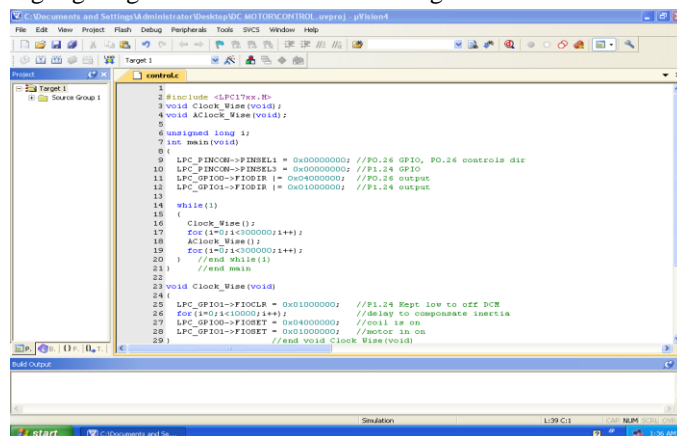


Figure 13 Color syntax highlighting.

In the project window, the source group 1 located below the target 1 was right clicked and the option Add existing file to group 'source group 1' was selected to add the .c source file to the group as illustrated in figure 14.

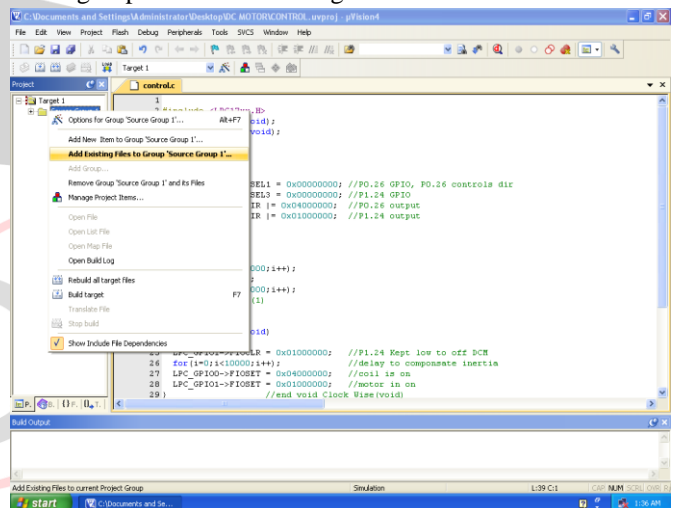


Figure 14 Adding the existing files to group source group 1

A new window named Add files to group 'source group 1' appeared where the file CONTROL.c was selected and then the option Add and Close was clicked one after the other as shown in figure 15

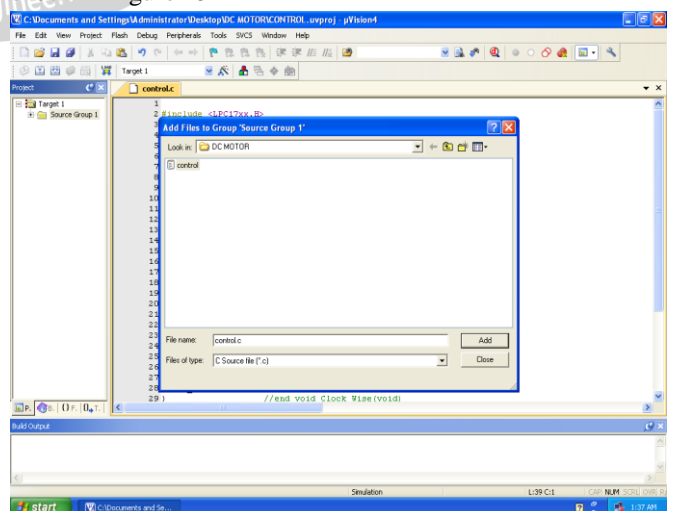


Figure 15 Selecting the file CONTROL.c

The window Add files to group 'source group 1' reappeared. Here the startup folder was chosen and in it NXP was chosen and then LPC17xx folder was opened and

finally system_LPC17xx was selected as shown in figure 16, 17 and 18 respectively.

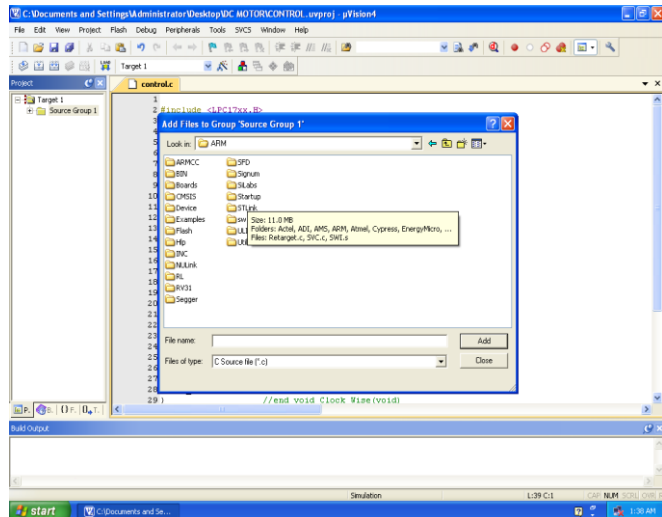


Figure 16 Selecting the start up folder

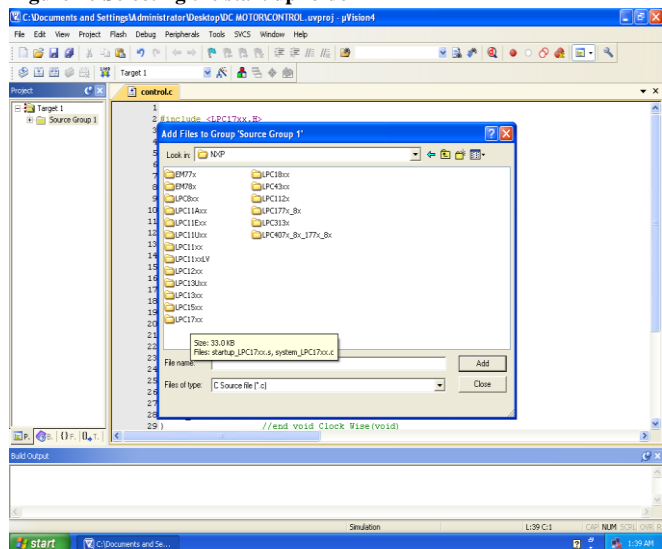


Figure 17 Opening the folder LPC17xx

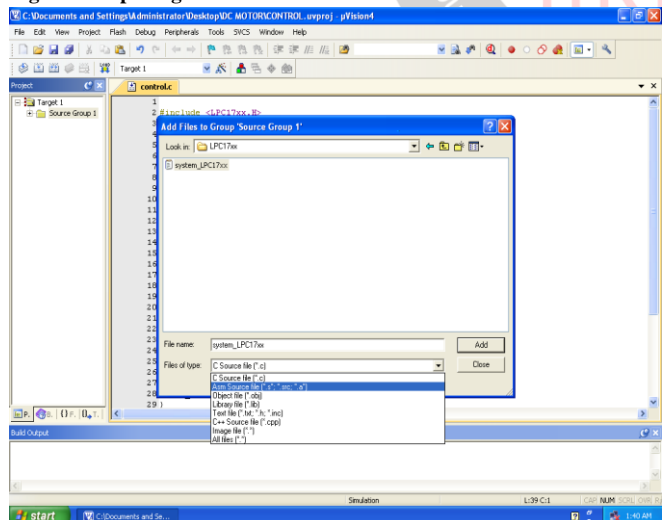


Figure 18 Selecting system_LPC17xx

The compilation of the files was the next important step which was done by clicking on the build icon as illustrated in figure 19

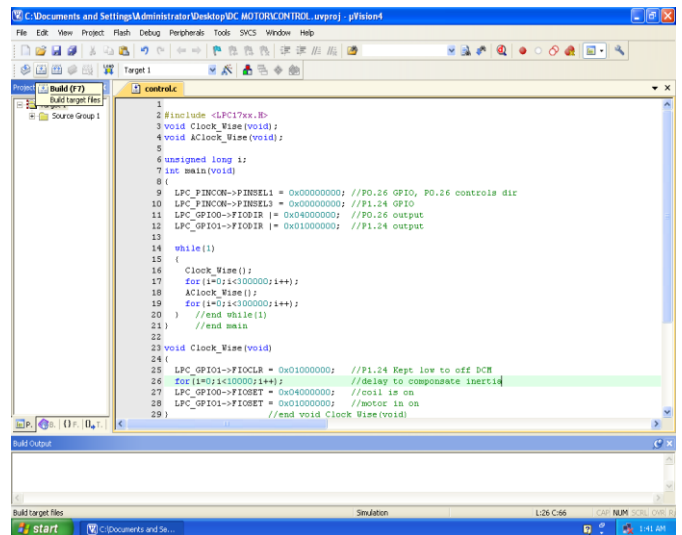


Figure 19 Choosing the build icon for compilation

After compilation the build output window was checked, where the message CONTROL.axf 0 Error(s), 0 warning(s) was displayed which ensured that the program was error free. This is shown in figure 20.

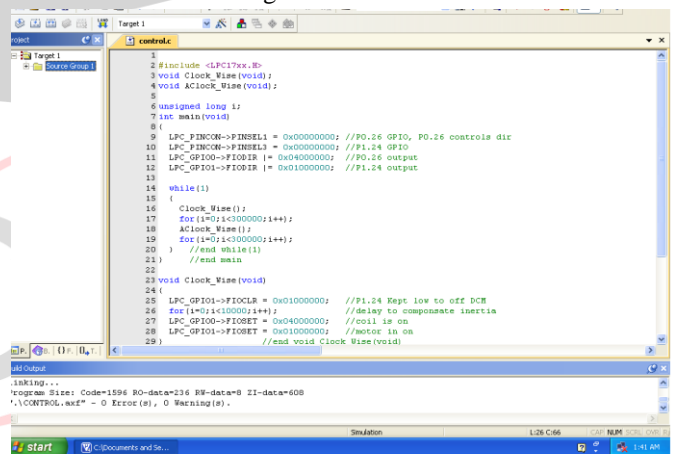


Figure 20 Obtaining the error free program with 0 errors and 0 warnings

In the project window, the option 'Target 1' was right clicked and then options for Target 'Target 1' was chosen as illustrated in figure 21.

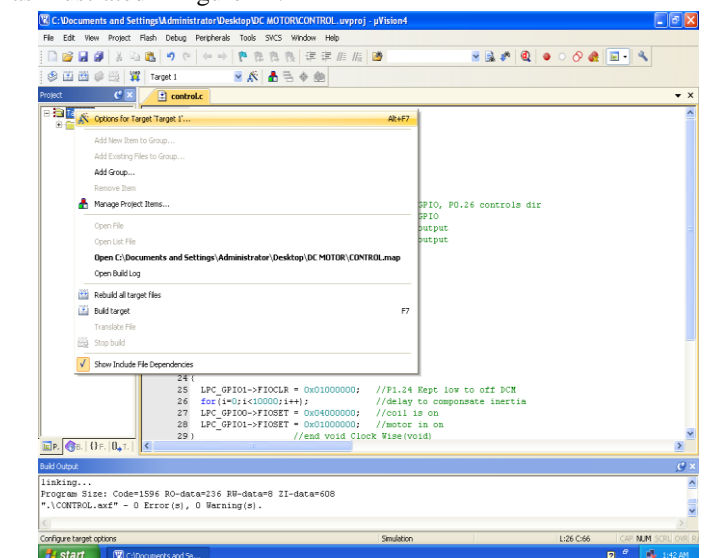


Figure 21 choosing options for Target Target 1

The options for Target 'Target 1' window appeared where output option was selected and then the option create hex file was enabled as depicted in figure 22

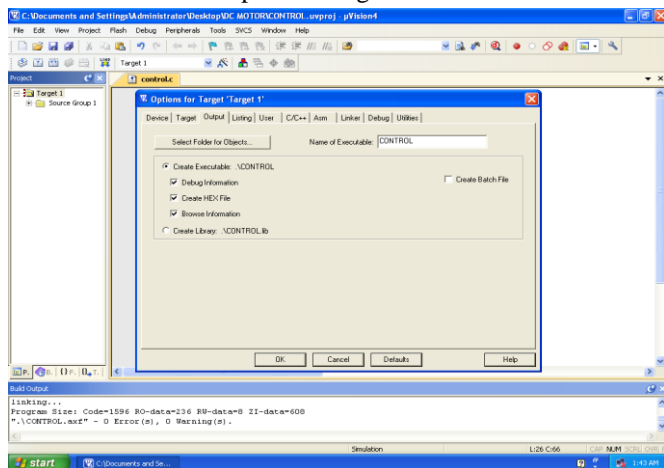


Figure 22 create hex file option enabled

In the same window the linker option was chosen and then the option Use Memory Layout from Target Dialog was enabled as shown in figure 23

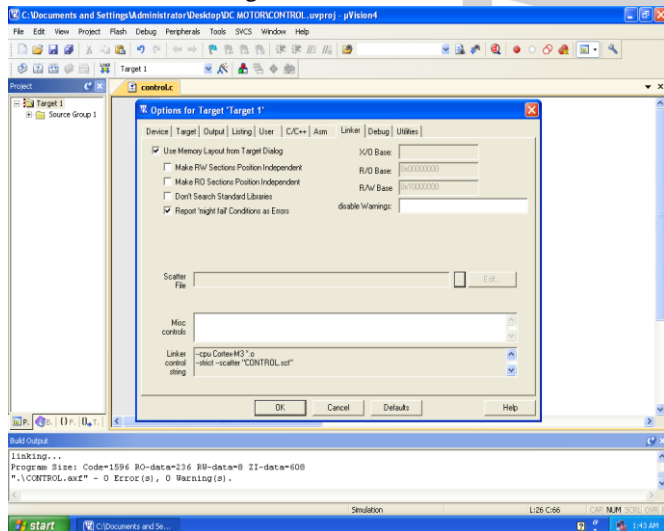


Figure 23 Use memory layout from target dialog option enabled

Finally to come out of this window ok option was clicked. Lastly the rebuild icon was clicked to rebuild all the target files as shown in figure 24

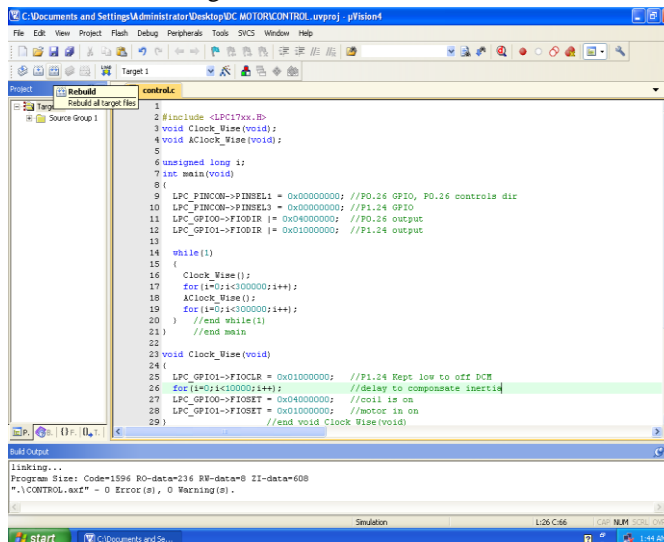


Figure 24 Selecting the Rebuild icon to build target files.

This created the hex file with 0 Error(s) and 0 warning(s) as displayed in the build output window. This is illustrated in figure 25.

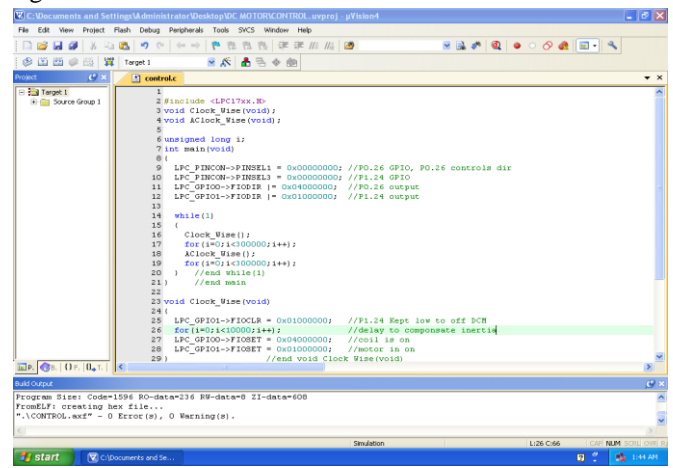


Figure 25 Creation of Hex file.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The required output in the form of rotation of DC motor in clockwise and anticlockwise direction was observed. The rotation of DC motor was continuous in loop. Flash magic software played a vital role in obtaining the final output. The flash magic software located on the desktop computer was double clicked. Five important steps were performed in order to get the desired output.

STEP 1: COMMUNICATION

The following selections were done in this step.

- Device: LPC1768
- Com port: COM 3 (as the USB cable is connected to this port).
- Baud rate: 19200
- Interface: None (ISP)
- Oscillator (MHz) : 12

STEP 2: ERASE

In step 2, the option erase blocks used by the hex file was selected by enabling it.

STEP 3: HEX FILE

In this step, browse option was clicked so as to download the hex file. In the proposed system the hex file with the name CONTROL.hex was contained in a folder named DC MOTOR on the desktop.

STEP 4: OPTIONS

Verify after programming option was selected by enabling it.

STEP 5: START

In this step, start option was clicked in order to download the hex file to the controller on the ARM cortex M3 LPC1768 evaluation board.

As step 5 was completed it was observed that the DC motor rotated in clockwise and anticlockwise direction.

The steps 1 to 4 are illustrated as shown in figure 26 and the last step that is step 5 is depicted as shown in figure 27.

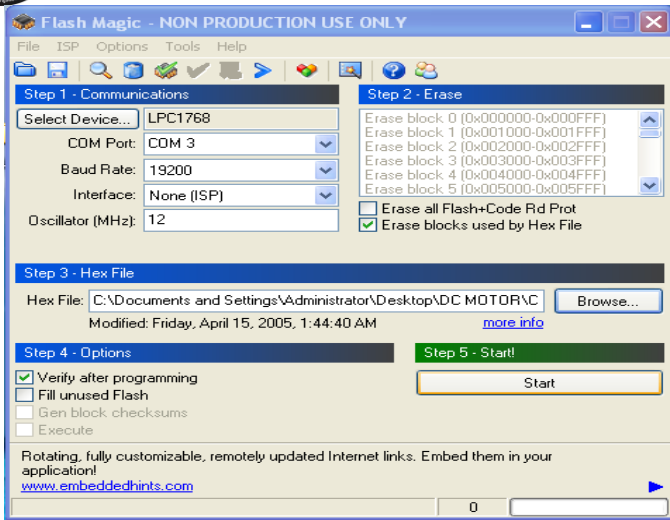


Figure 26 Photographic view of steps 1 to 4

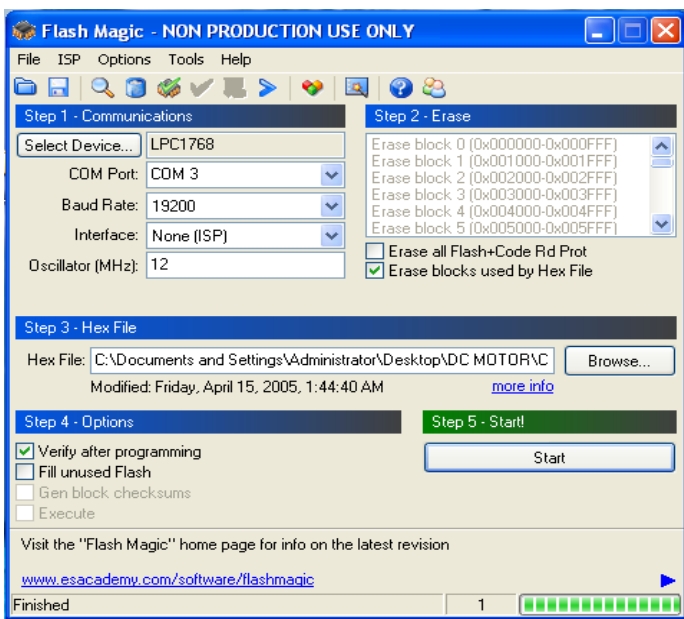


Figure 27 Photographic view of step 5

V. CONCLUSION

The DC motor was interfaced with the ARM cortex M3 microcontroller LPC1768. The interfacing was done using ALS-SDA ARM cortex M3-06 board, desktop computer and a DC motor. Keil μ -vision 4 and flash magic software were used in the system. The entire system is simple, reliable, easy to implement and is user friendly. Less cost is one of the major advantages of the system. The DC motor rotated in clockwise and anticlockwise direction and this process was continuous in loop. It is also possible to rotate the DC motor in only clockwise direction and by making code change the DC motor can also be rotated only in anticlockwise direction. In a similar way the speed of rotation of DC motor can be controlled.

ACKNOWLEDGMENT

First of all I would like to thank Almighty Allah by the grace of whom I reached the stage of completion of this work. This avenue has been a turning point in my career to

mold me into a thorough and dynamic Professional. My sincere thanks to the Management of esteemed institution Khaja Banda Nawaz College of Engineering for their inspiration and support.

REFERENCES

- [1] Joseph Yiu, *The Definitive Guide to the ARM cortex-M3*, second edition, Elsevier: USA Texas instruments incorporated, 2010, pp. 1-9.
- [2] Andrew N Sloss, Dominic Symes, Chris Wright, *ARM System Developer's guide Designing and optimizing system software*. USA: Morgan Kaufman publications, 2004.
- [3] Datasheet LPC1769/68/67/66/65/64/63 32-bit ARM cortex M3 microcontroller, NXP semiconductors, 4 May 2018, pp. 1-3.
- [4] D.C Kulshreshtha, *Basic Electrical Engineering*, Revised first edition, Mc Graw Hill education India private limited 2012, pp.535-538.
- [5] B.L. Theraja, A.K Theraja, *A Text book of Electrical Technology- AC and DC machines*, 23rd revised edition, vol.2, S.chand publishers 2015, pp.995-999.
- [6] A manual on Keil tools by ARM: Getting started Creating applications with μ -vision pp. 55-85, 1997-2009, www.keil.com
- [7] L.E Davis, H.H. Moghbelli, A.Ahmed, "Microprocessor control of DC motor", *IEEE Industry Applications Society Annual Meeting*, Houston, TX, USA, 1992,
- [8] M.V Ramesh, Gorantla. S Rao, J. Amarnath, S. Kamakshaiyah, B. Jawaharlal, "Speed Torque Characteristics of Brushless DC motor in Either Direction on Load using ARM controller", *IEEE conferences PES Innovative Smart Grid Technologies-India*, 2011, pp. 217-222.
- [9] Gururaj Mulay, Akshay Yembarwar, Surabhi Rajee, "A DC motor driver consisting of a single MOSFET with capability of speed and direction control". *IEEE 6th India international conference on Power Electronics (IICPE)*, 2014 Kurukshetra India.
- [10] Tazwar Muttaqi, Thomas Baldwin, S Hossein Mousavinezhad, "DC Motor Cruise Control by Interfacing IR Sensor Tachometer and SIMULINK, Regardless Motor Specifications and change of shaft load", *IEEE International conference on Electro/Information Technology*, Rochester, MI, USA 2018.
- [11] Aruna Kommu, Naveen Kumar Uttarkar, Raghavendra Rao Kanchi, "Design and development of sensor based mini projects for embedded system laboratory using ARM cortex-M3 (LPC1768)" *International conference on Information Communication and Embedded Systems (ICICES)*, Chennai India 2014.
- [12] Md. Moyeed Abrar, "Interfacing a Stepper Motor with ARM Controller LPC2148", Volume 4 Issue 3, *International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRSCE)*, pp. 72 – 80, March 18.
- [13] Jianxin Tang, Real time DC motor control using the MATLAB interfaced TMS320C31 digital signal processing starter kit (DSK), *Proceedings of the IEEE 1999 International Conference on Power Electronics and Drive Systems. PEDS'99*, vol. 1 Hong kong 1999.