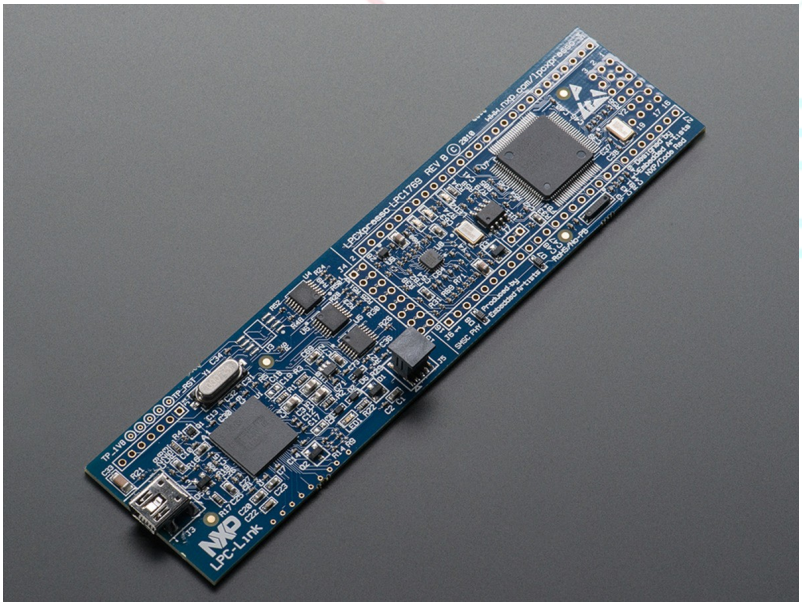
TENET TECHNETRONICS

2016

Interfacing DC motor with NXP LPC1769 using LPCXpresso



Author: Gurudatta Palankar

Reviewers:

Version: 1.0

9/3, 2nd floor, SreeLaksmi Complex, opp, to Vivekananda Park, Girinagar, Bangalore - 560085,

Email: info@tenettech.com, Phone: 080 - 26722726

Introduction:

LPCXpresso™ is a new, low-cost development platform available from NXP supporting NXP's ARM-based microcontrollers. The platform is comprised of a simplified Eclipse-based IDE and low-cost target boards which include an attached JTAG debugger. LPCXpresso™ is an end-to-end solution enabling engineers to develop their applications from initial evaluation to final production.

Step 1: Open **LPCXpresso** IDE

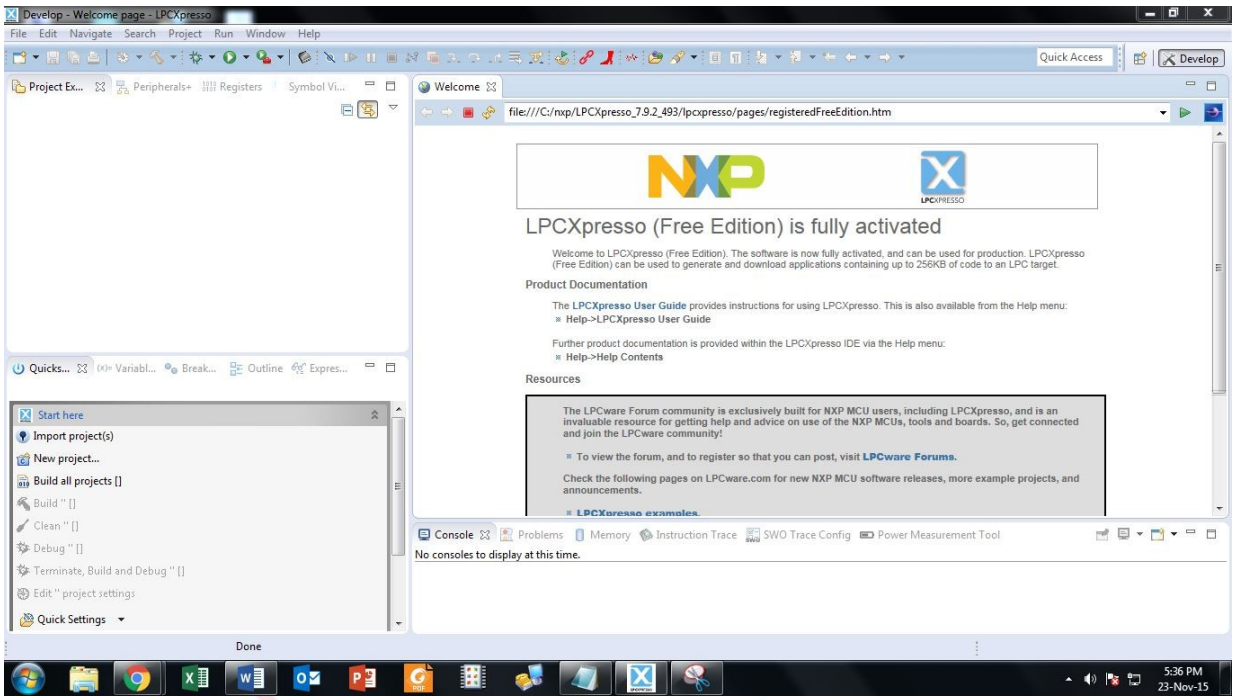


Figure 1

Step 2: Before writing a code, we have to Import some Library Files to the Workspace. Click on **Import projects** on Quickstart Panel on the bottom left of the window.

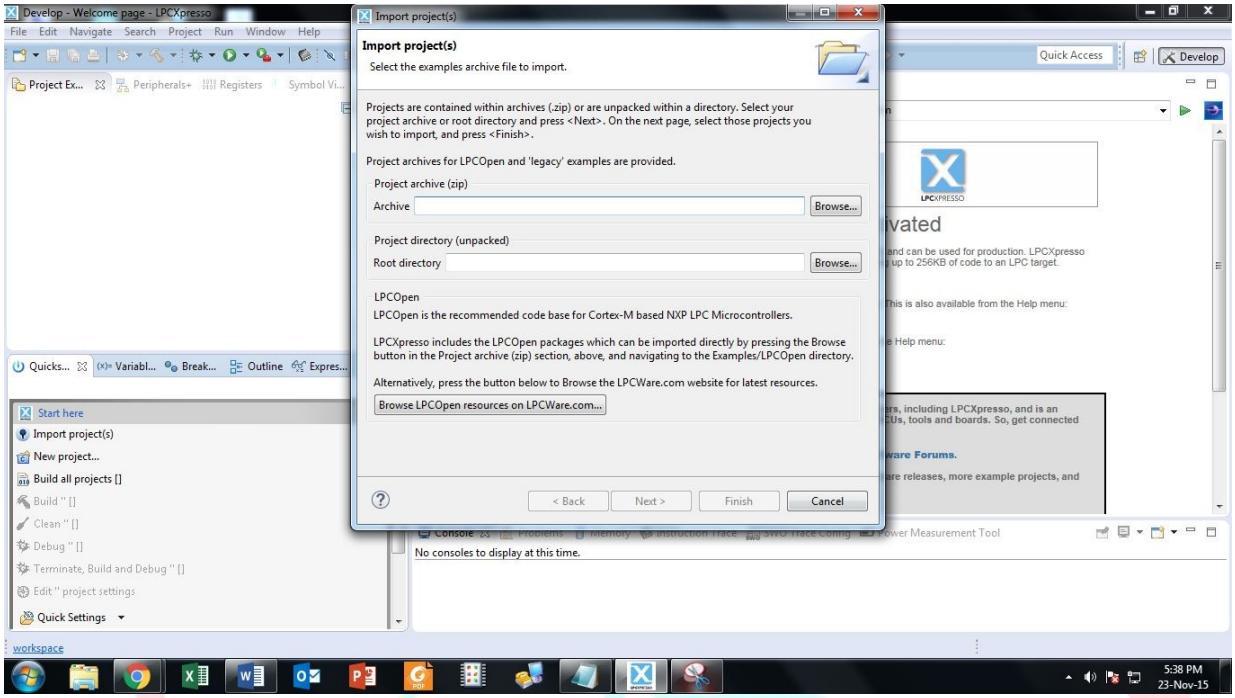


Figure 2

Step 3: Browse file, open the LPC1000 folder.

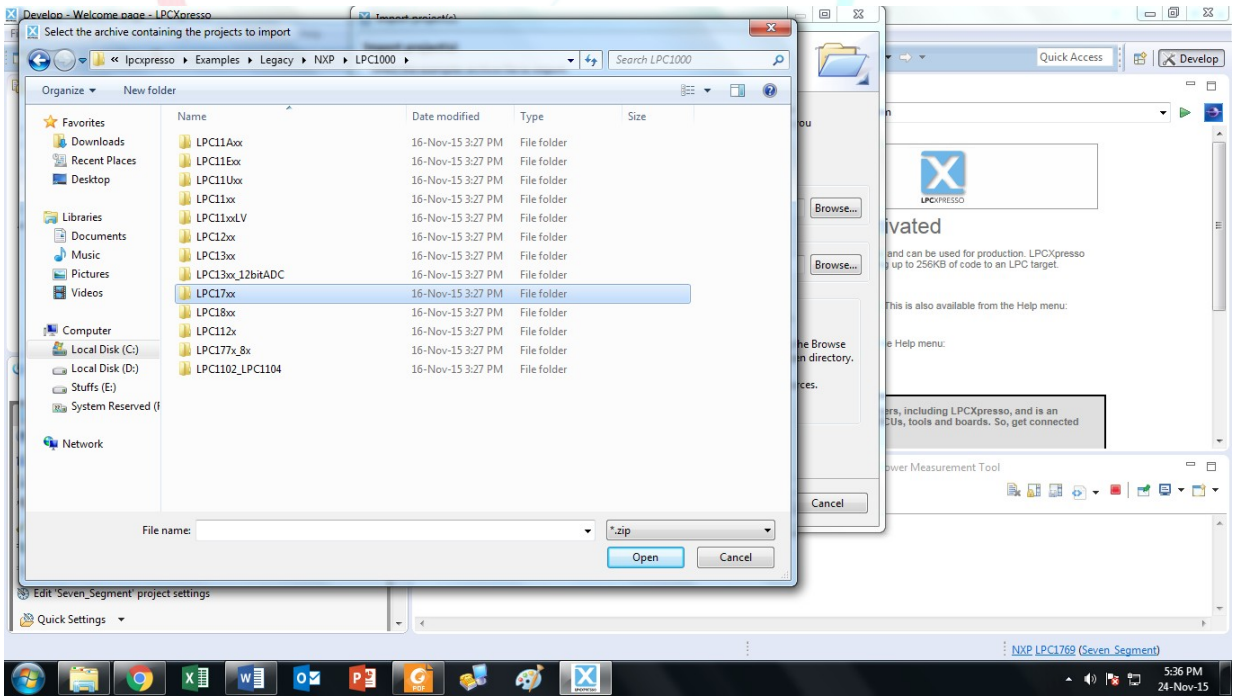


Figure 3

Step 4: Select the appropriate archive file. Let us select LPCXpresso176x_cmsis2. We can select CMSIS CORE library that include LPC17xx.h header file.

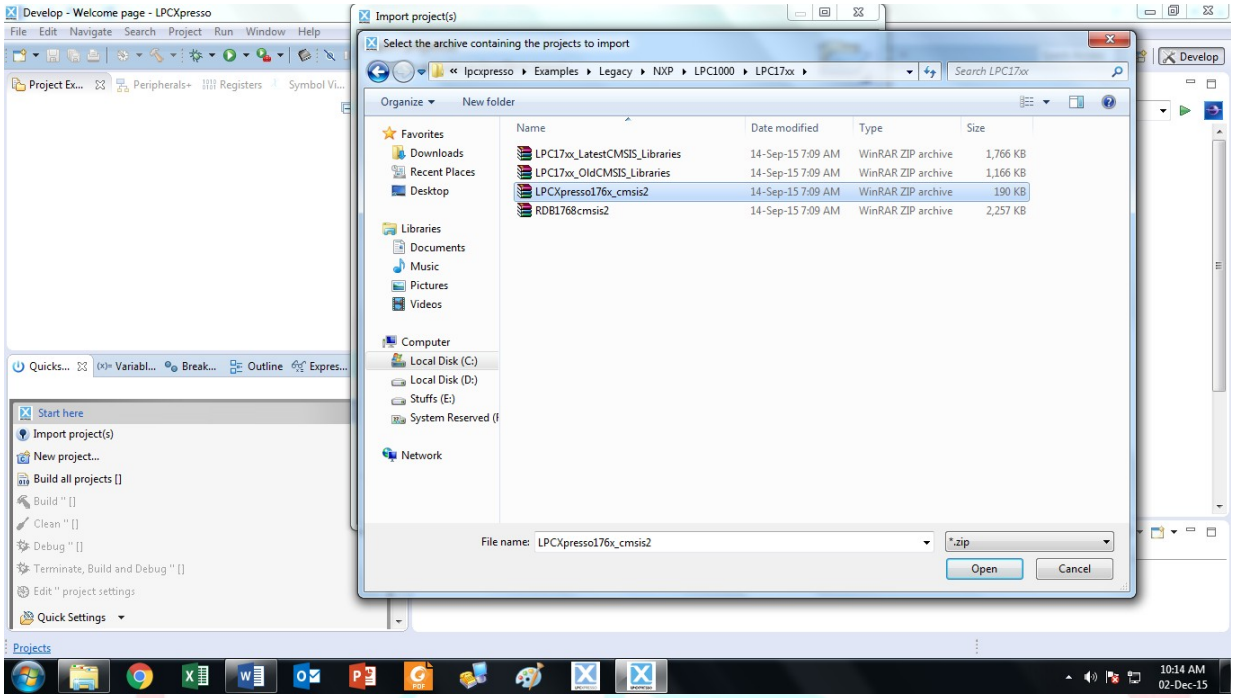


Figure 4

Step 5: After selecting you will be able to see the following libraries files. Let us select specific one.

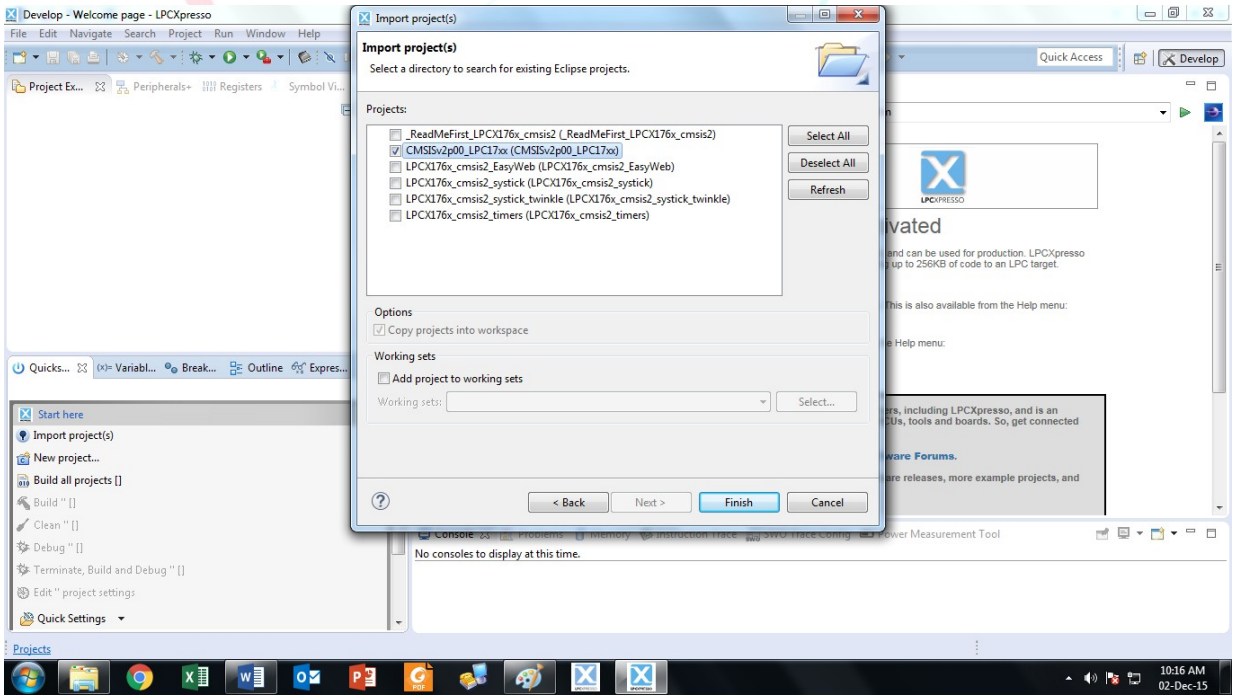


Figure 5

Step 6: Now we will be able to see those libraries in the workspace.

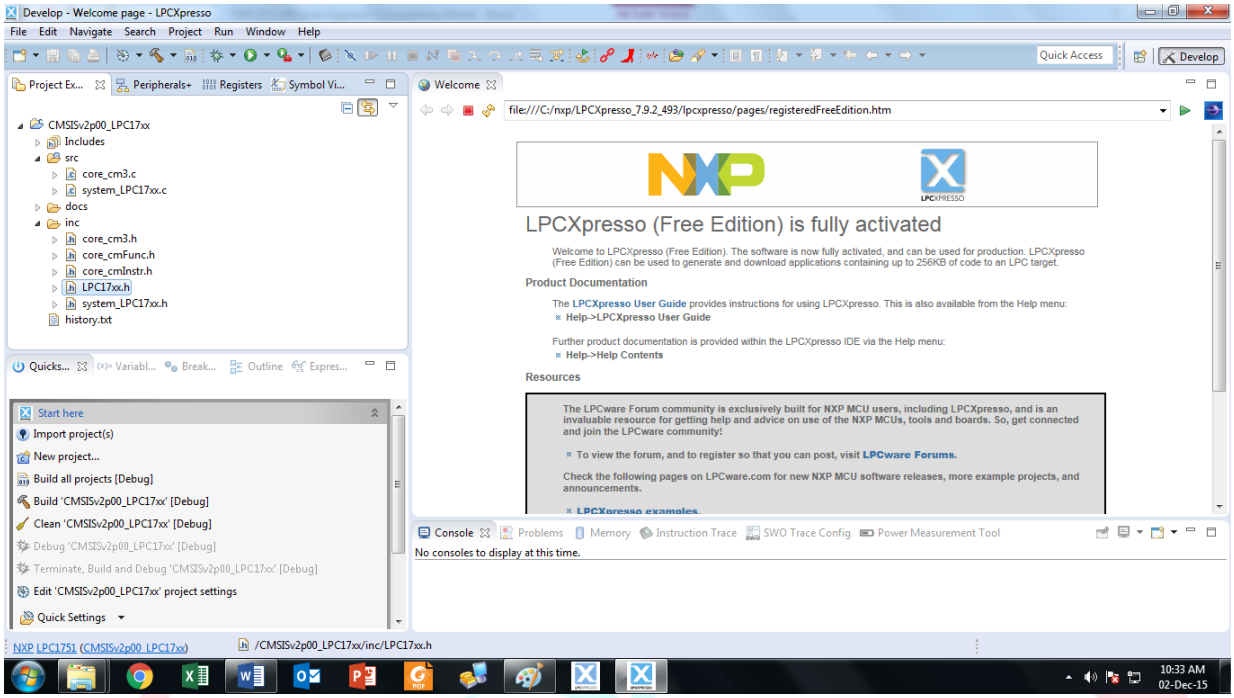


Figure 6

Step 7: Now we can start creating our new project. Goto **File** >> **New** >> **Project.** Select **LPCXpresso C project.**

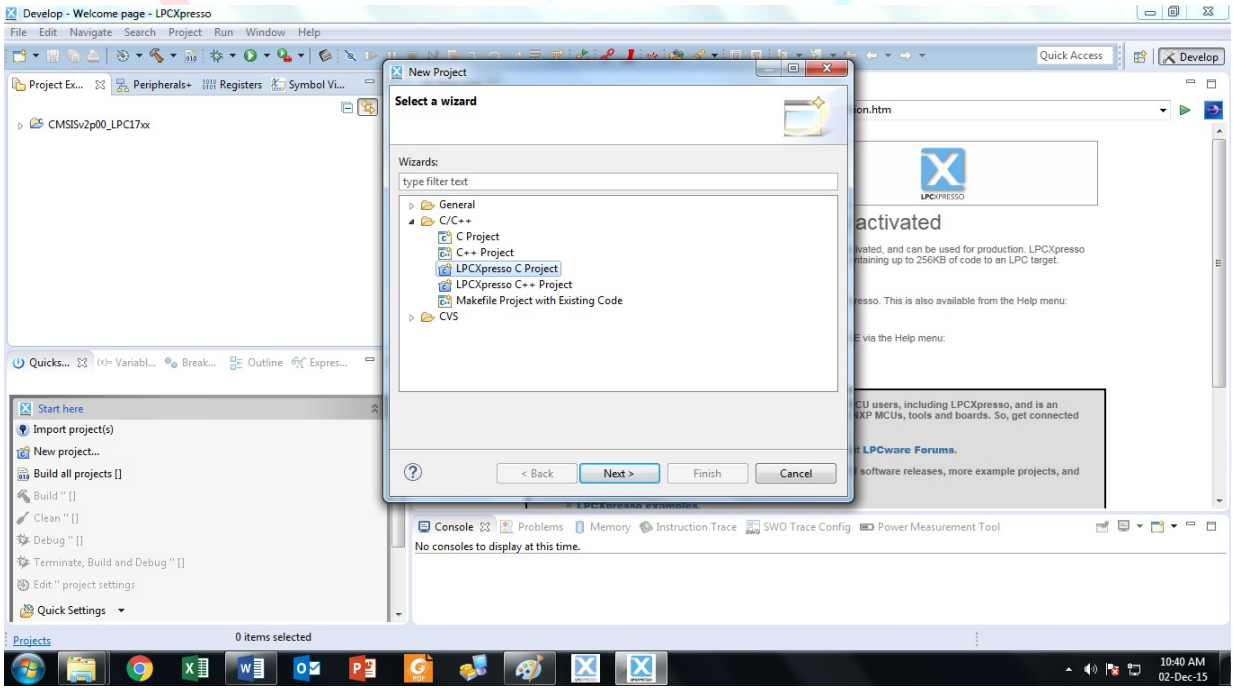


Figure 7

Step 8: Select LPC1769, **C Project** and give name to your project. Select target MCU as LPC1769.

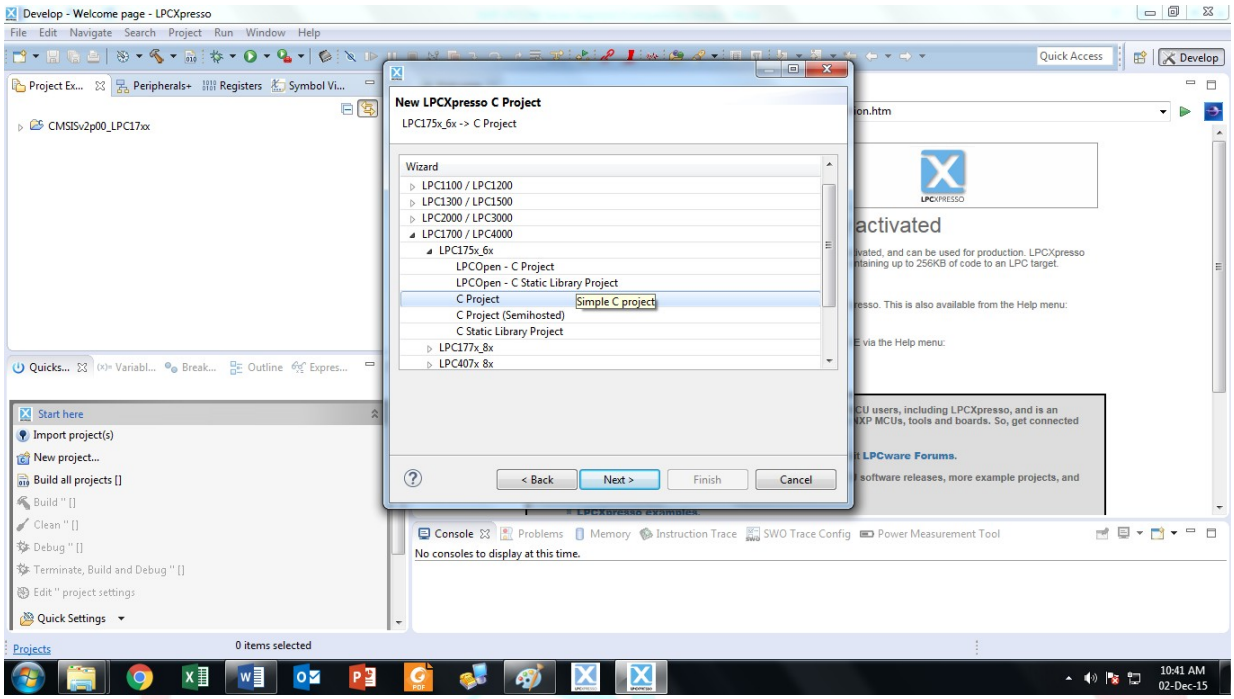


Figure 5

Step 9: Now select CMSIS Core library. Click on Next and keep all the other configurations as default and Finish.

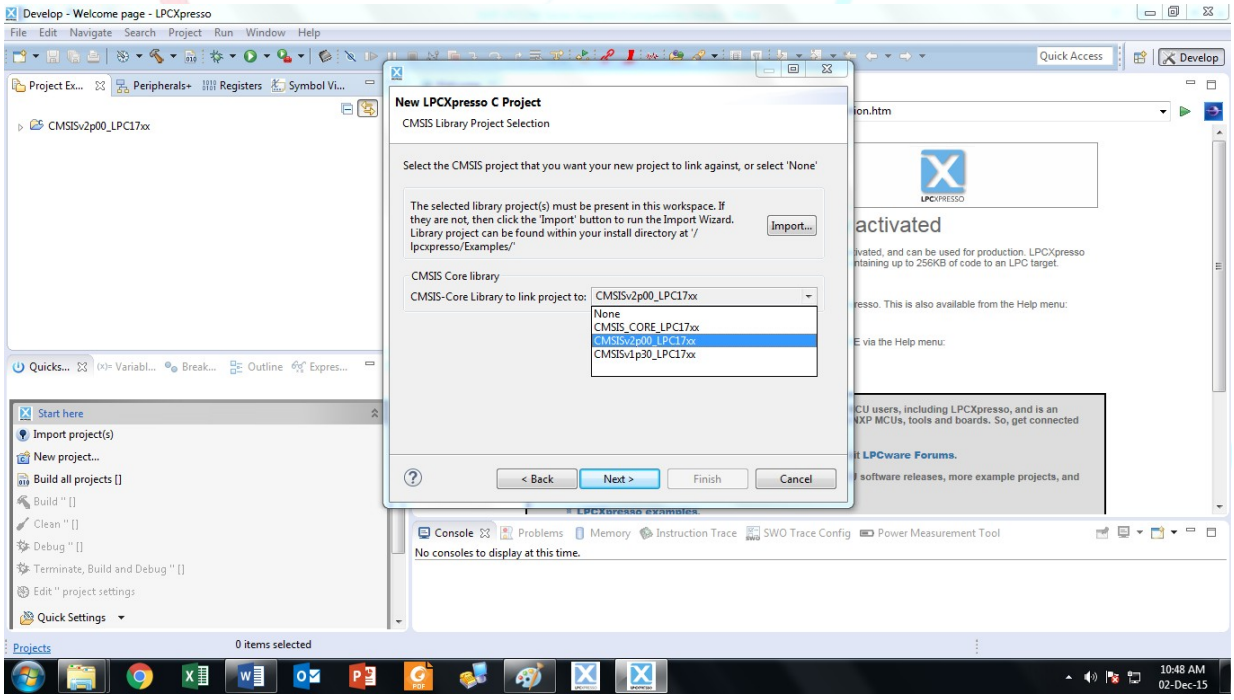


Figure 9

Step 10: Now we can see our project onto the workspace. Now by double clicking on DC_Motor.c file, we can start writing code.

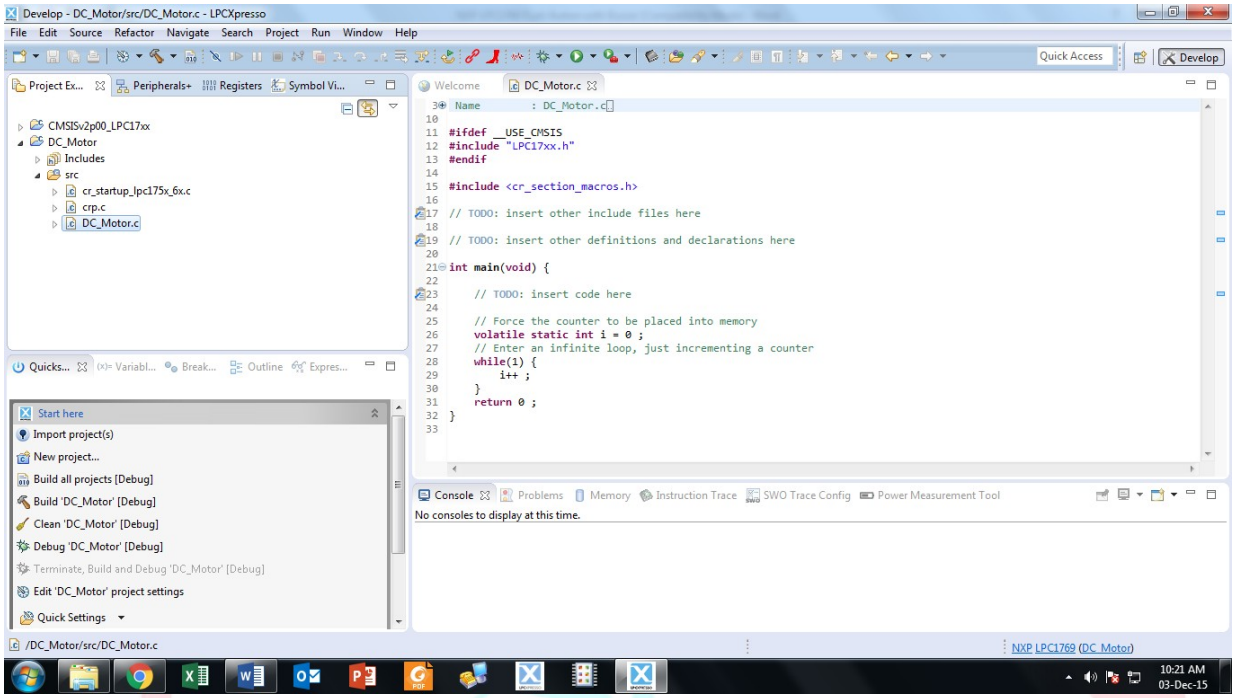


Figure 10

Step 11: Write a code as shown below.

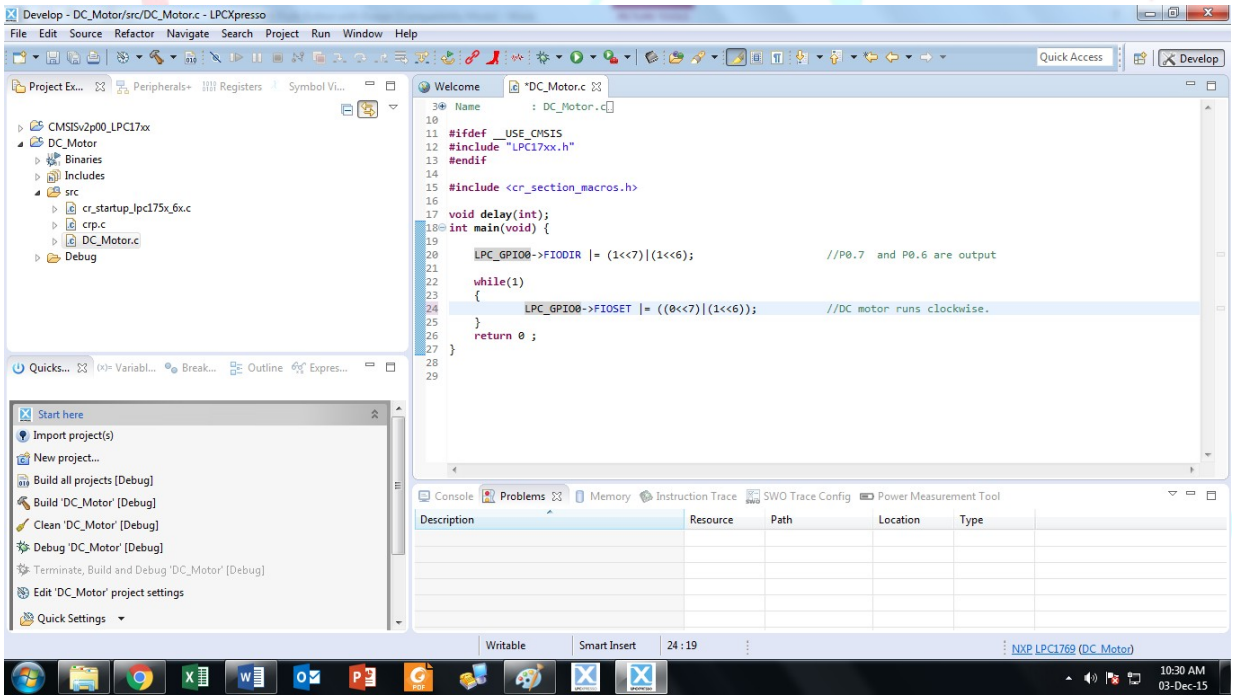


Figure 11

CODE:

```
#ifdef __USE_CMSIS
#include "LPC17xx.h"
#endif

int main(void)
{
    LPC_GPIO0->FIODIR |= (1<<7)|(1<<6);          //P0.7  and P0.6 are output
    while(1)
    {
        LPC_GPIO0->FIOSET |= (0<<7)|(1<<6);      //DC motor runs clockwise.
    }
    return 0 ;
}
```

Step 12: After writing code, Build the project by clicking on Build Buzzer on the Quickstart Panel on the bottom left of the window.

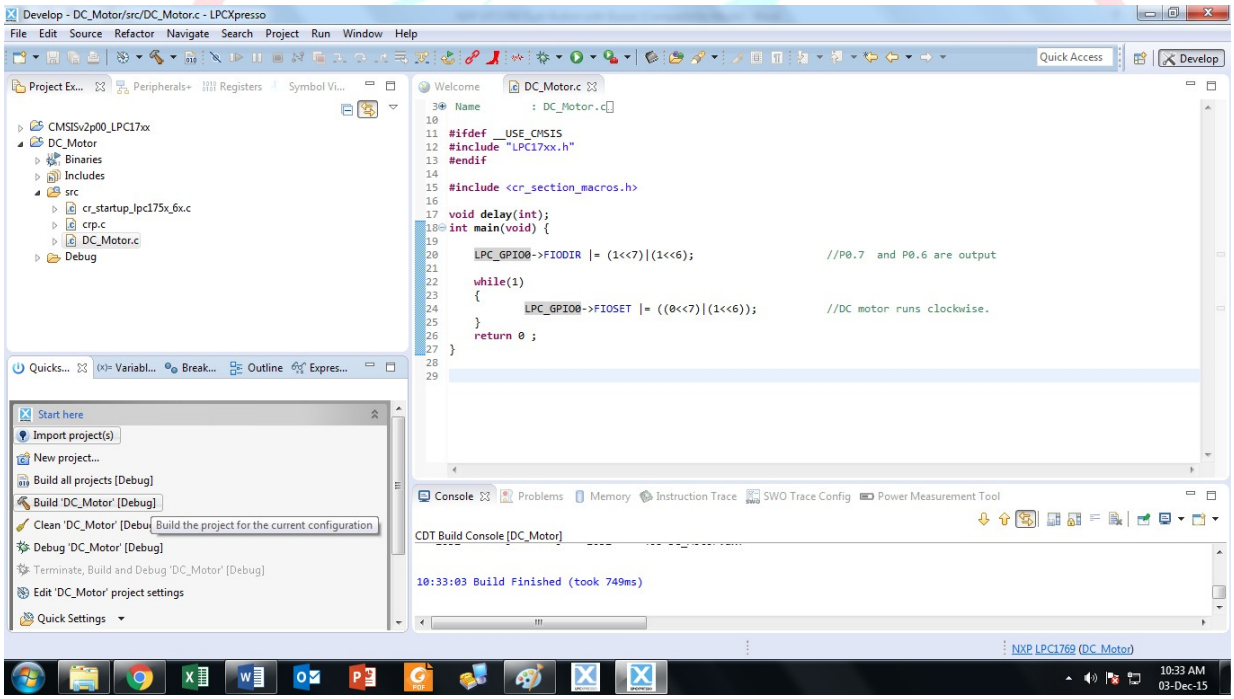


Figure 12

Step 13: Now, if all goes well connect the Micro B cable to LPC1769 and connect it to your computer. To upload the project file, click on the Program flash.

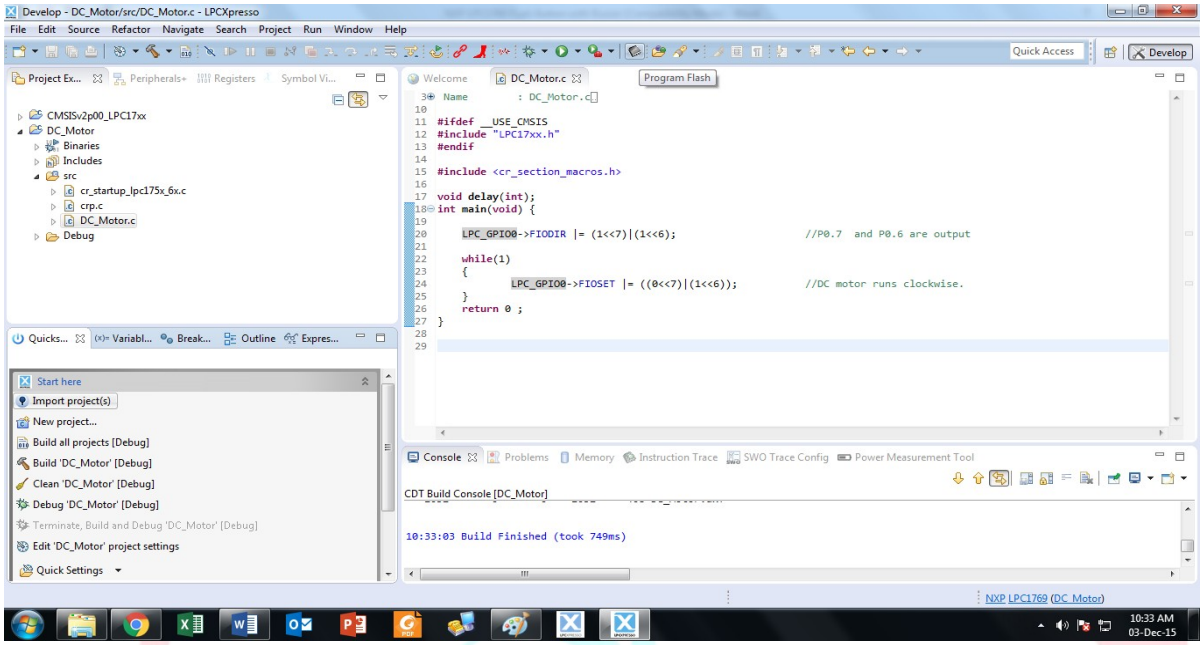


Figure 13

Step 14: Now select the Project file DC_Motor.axf. We can find it in our project folder.

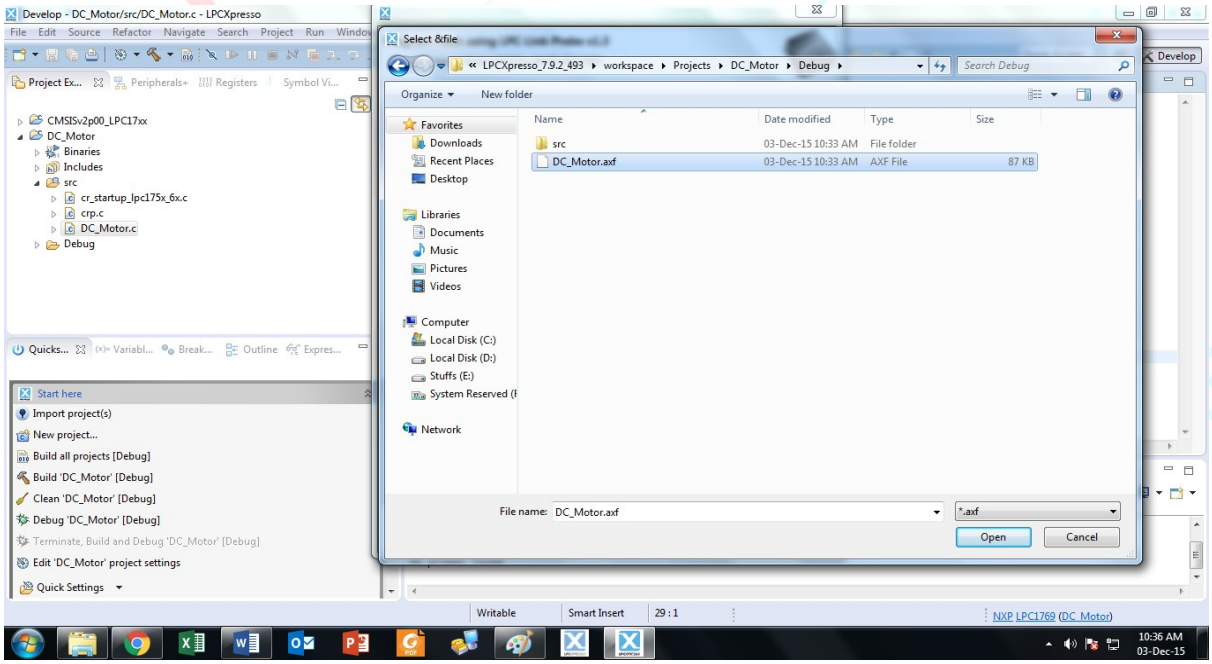


Figure 14

Step 15: Now this window shows we have finally dumped our project onto LPC1769.

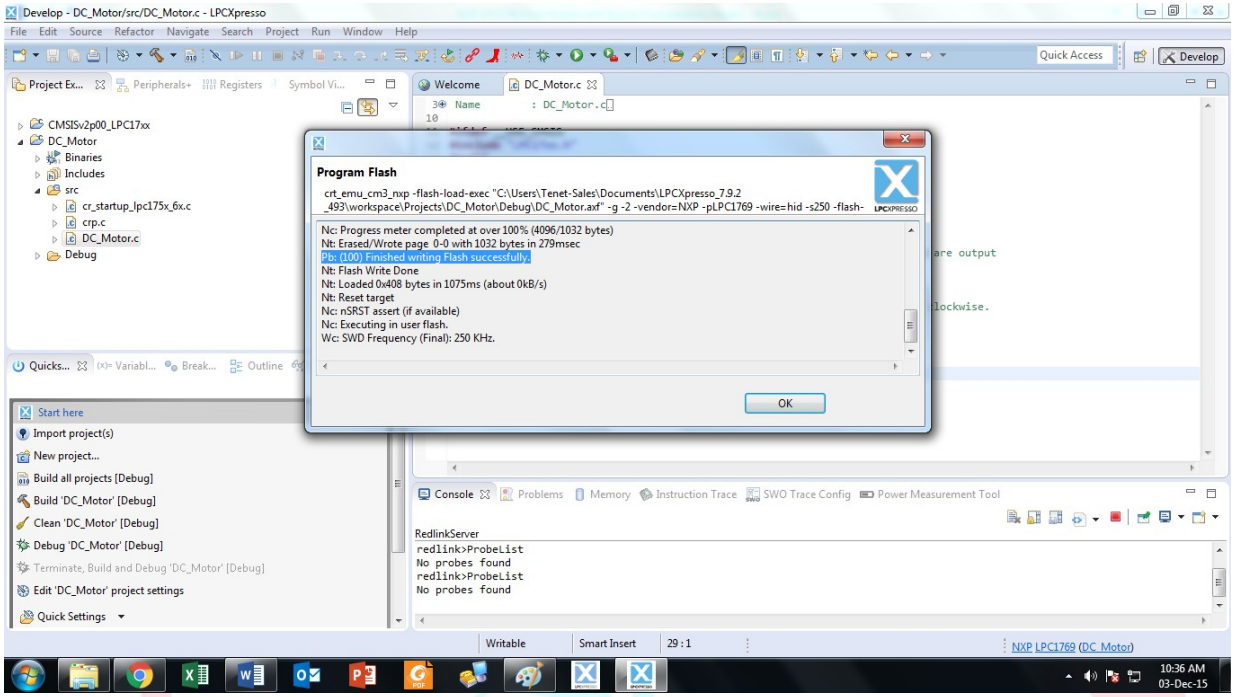


Figure 15

CIRCUIT EXPLANATION:

Interface DC motor using L293d motor driver

This example shows how to control dc motor in forward and reverse direction with LPC1769 using motor driver IC.

Hardware:

- LPC1769 Board
- DC motor
- L293d IC
- breadboard
- hook-up wire

L293D motor driver IC:

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The mode of operations of two motors can be

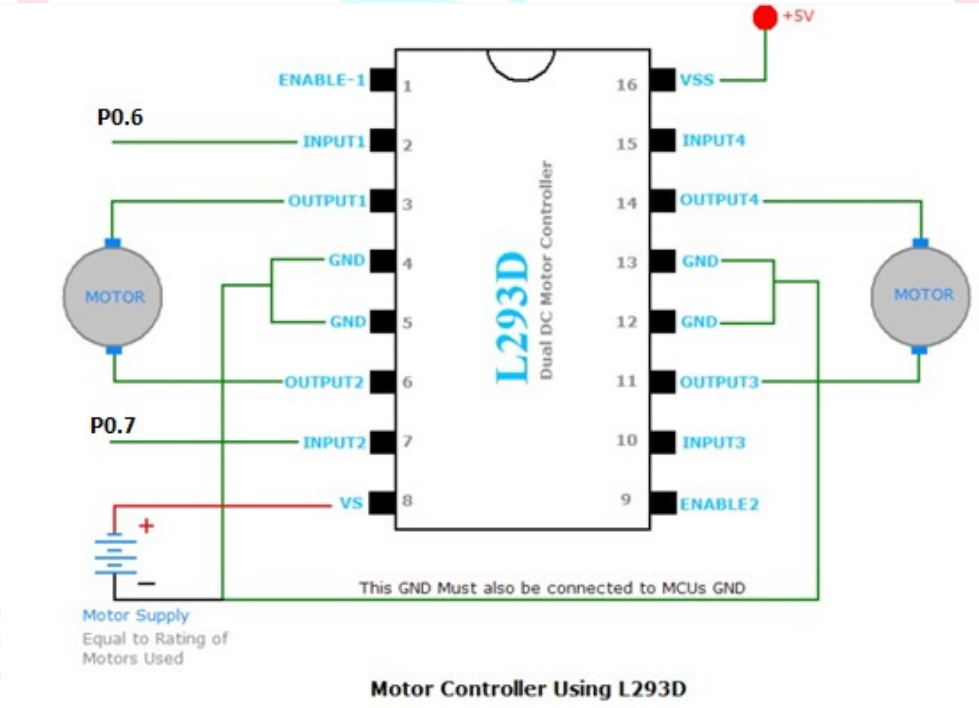
controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

Circuit:

The L293D has two +V pins (8 and 16). The pin '+Vmotor (8) provides the power for the motors, and +V (16) for the chip's logic. We have connected both of these to the 5V pin. However, if you were using a more powerful motor, or a higher voltage motor, you would provide the motor with a separate power supply using pin 8 connected to the positive power supply and the ground of the second power supply is connected to the ground of the LPC1769.

Remaining two input pins goes two PORT1 on the LPC1769 board and two output pins goes to the motor. The enable pin is connected to 3.3v on the LPC1769.



Motor Controller Using L293D

Figure 16

CONNECTION DIAGRAM:

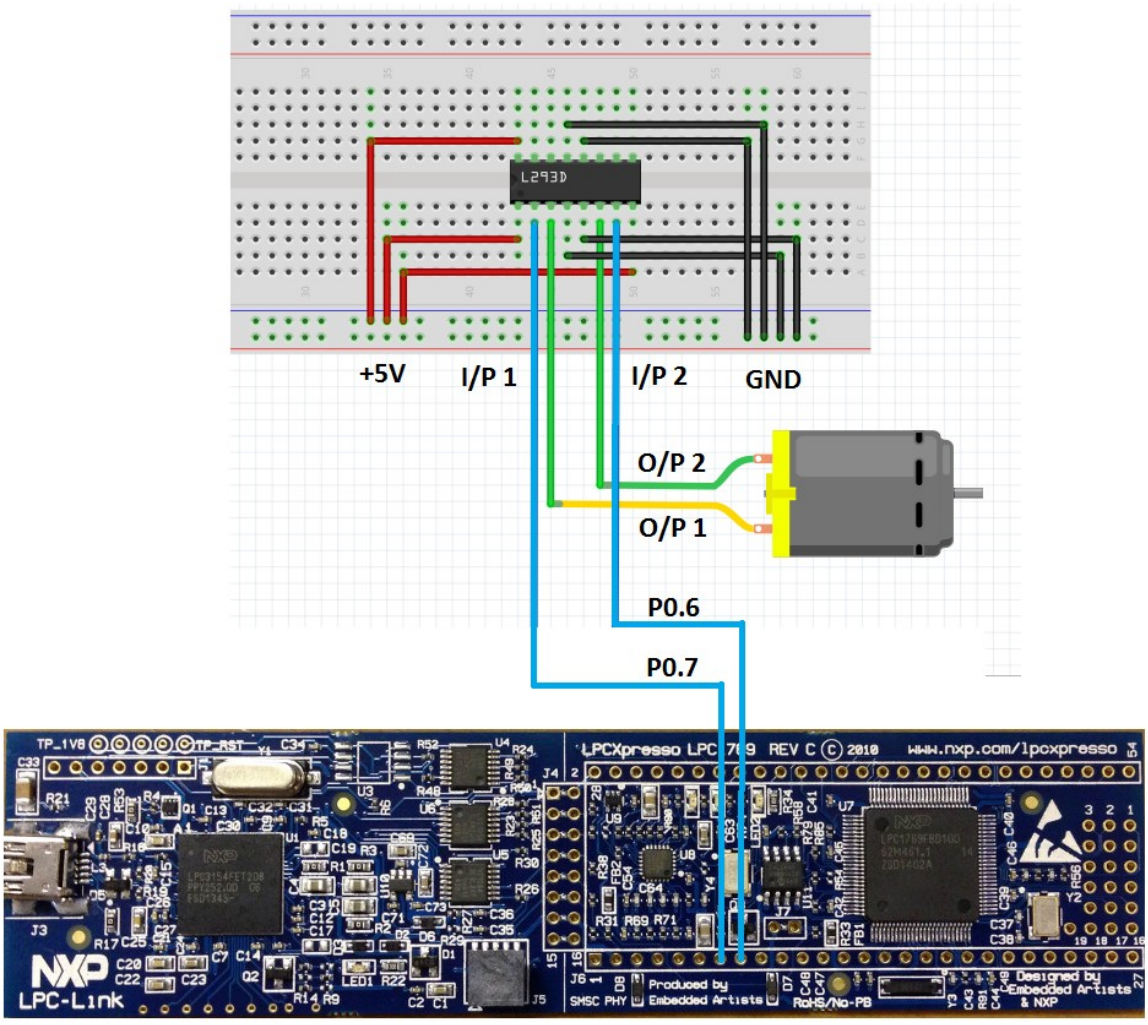


Figure 17

TABLE:

Output A	Output B	Description
0	0	Motor stops
0	1	Motor runs clockwise
1	0	Motor runs anti-clockwise
1	1	Motor stops

OUTPUT:

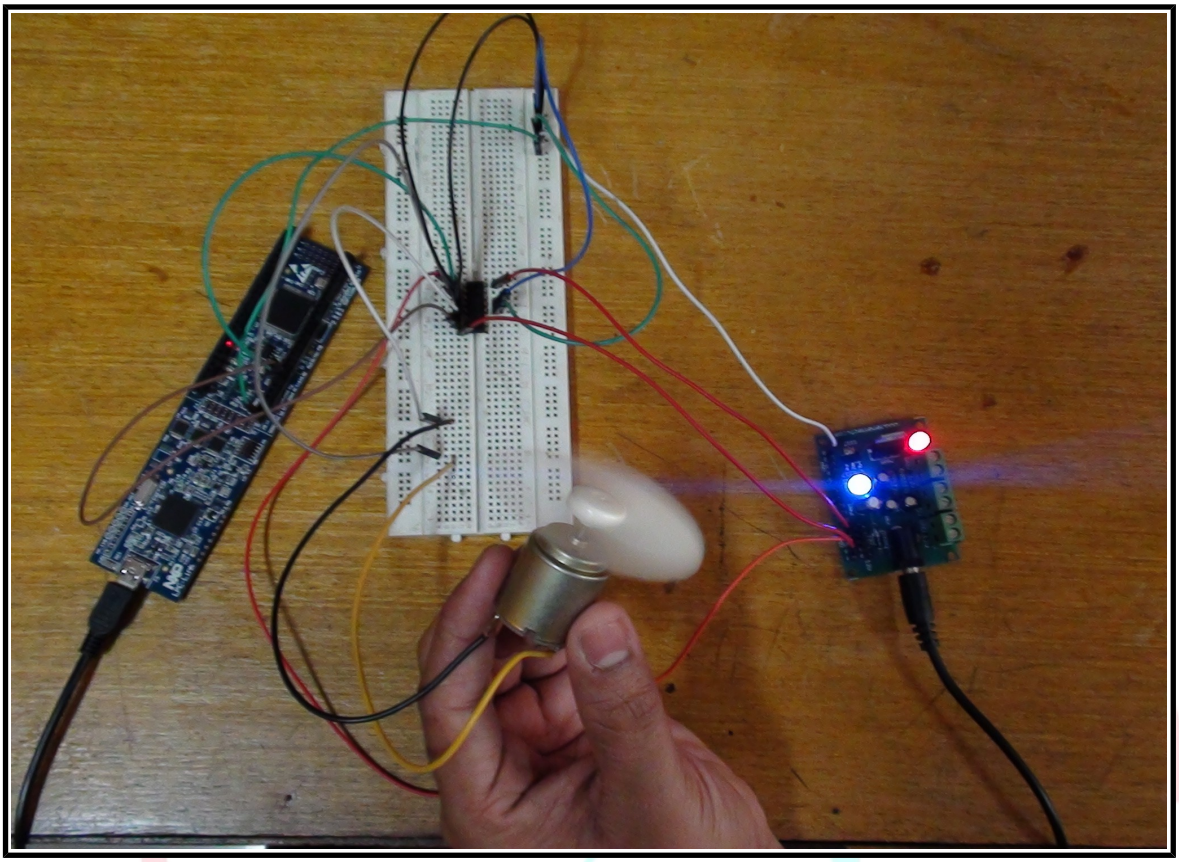


Figure 18

For product link:

1. **http://www.tenettech.com/product/1548/lpc1769-lpcxpresso-board**
2. **http://www.tenettech.com/product/3725/l293d-motor-driver-breakout-board**
3. **http://tenettech.com/product/6655/universal-gpio-board**
4. **http://www.tenettech.com/product/2846/power-supply-breakout**

For more information please visit: www.tenettech.com

For technical query please send an e-mail: info@tenettech.com