Abstract
This application note examines BLDC motor control and offers a solution for setting up a BLDC motor for PWM testing using Texas Instruments' DRV8301 control kit and Piccolo TMS320F28069M microcontroller. Using these components creates a conceptually simple design while saving space and making integration easier to implement. The solution proposed uses the hardware mentioned as well as TI’s controlSUITE and control algorithms to simplify the software aspect as well. This application note will allow the user to gain knowledge on BLDC motor theory, BLDC motor control theory and how to set up testing for PWM for a BLDC motor for use in a RC car drone.

Keywords
RC car, drone, BLDC motor, motor control, DRV8301, controlSUITE, Piccolo controlCARD
Contents
Introduction .......................................................................................................................... 2
BLDC Motor Theory ............................................................................................................. 2
Control Theory .................................................................................................................... 3
Test Setup ............................................................................................................................. 4
  Hardware Setup ................................................................................................................... 4
    1. Integrating the motor ................................................................................................... 4
    2. Integrating the Microcontroller ................................................................................. 4
Software Setup .................................................................................................................... 6
  1. Code Composer Studio ................................................................................................. 6
  2. ControlSUITE ............................................................................................................. 6
Conclusion ............................................................................................................................. 7
References ............................................................................................................................ 8

Introduction
In order to create a functional RC car drone, multiple components need to be integrated together and
one main component is the motor. The motor selected depends on size of the vehicle, control board
used, microcontroller selected and control algorithms, among other things. BLDC motors are the best
choice for RC cars due to their efficiency versus brushed motors and the availability of such motors for
RC cars. In order to properly control a BLDC motor, some theoretical knowledge is necessary.

BLDC Motor Theory
Brushless DC motors are synchronous motors that are powered by a switching DC power supply. A
standard BLDC motor stator has three coils with three elements in series. These include an inductance, a
resistance and a back electromotive force or induced voltage[1]. The rotor has multiple permanent
magnets attached that are responsible for setting up the magnetic field in the motor. The multiple
switches between windings in the stator develop a revolving field in the air gap. The rotor is constantly
following the electromagnetic poles on the stator and this develops torque in the motor.

Synchronous motors have two main types of back-electromotive force profiles and they can be
simplified to either producing trapezoidal induced voltages or sinusoidal induced voltages. Sinusoidal
motors have less vibration and suffer from less mechanical stress but also have higher switching
losses[2]. Typically, these motors are called permanent magnet AC motors. In a trapezoidal motor,
current only flows through 2 of the 3 windings while sinusoidal motors may have all three phases on at
the same time. These types are exclusively called BLDC motors.
Control Theory

The voltage induced in a BLDC motor is proportional to motor speed and the torque being produced. In trapezoidal BLCD motors, the phase windings are distributed where only two phases conduct at any given time as mentioned in the BLDC theory section. Texas Instrument’s DRV 8301 control kit includes 3 half bridge drivers that are capable of driving two N-type MOSFETS and acts as the gate driver for the motor. A simplified application schematic can be found below[3].

![DRV8301 Application Schematic](image)

The switching is done by utilizing the MOSFETS to control two phases at a time and modulate a signal. The motor controller from the diagram determines which transistors must be driven bases on the position information coming from the sensors in the motor(or using sensorless techniques in this case). This results in an output similar to figure 2 below.

![Trapezoidal BEMF Waveform](image)

It can be observed that that by allowing current to flow in two of the three phases and combining with a trapezoidal BEMF, that it is theoretically possible to have constant torque.
Test Setup

Hardware Setup

1. Integrating the motor

As can be observed from Figure 1: DRV8310 Application Schematic, the motor must be attached to the 3-phase motor output of the DRV8301 control board so that the board can drive the motor. A typical motor will come supplied with 3 wires for the 3 phases and a ground plate. The example motor here is the HP RM1-5677 used for color inkjet printers. This motor is also suitable for RC cars. The DRV8301 labels each phase and ground to make wiring straightforward. An example can be found as figure 3.

![Figure 3: Wiring of BLDC Motor](image)

These steps should be followed to connect a motor:

- Solder extra wire to each phase wire of the motor until satisfactory length is achieved
- Wire each phase wire (single wire attached to motor) to OUT_A, OUT_B and OUT_C
- Solder a wire to the bottom plate of the motor (GND)
- Connect this wire to the GND terminal

2. Integrating the Microcontroller

The DRV8301 kit used came equipped with the Piccolo TMS320F28069M microcontroller. The advantage here lies in the fact that the instaSPIN-FOC and InstaSPIN-MOTION software is located in the ROM. The C200 family of devices all possess the desired computation power to execute these algorithms. This saves time and effort versus developing digital control algorithms individually. The microcontroller has a standard 100-pin DIMM interface and is designed to interface with the DRV8301. To properly setup the controlCARD, the user must make sure the DIP switch settings match figure 4.
Once this step is completed, the card must be snapped onto J1 of the control board as indicated in figure 5. The controlCARD has a USB cable connection also labeled J1. This is used to load code on to the microcontroller for testing purposes. It is important to note that J2 should also get installed and is outlined in figure 5.

The last step for hardware involves connecting a power supply. The board is located on the vehicle and is powered by batteries (60V max) to the PVDD and GND terminals. For the purposes of testing code and practicing safety during testing, this application note requires powering the board through a power supply. This is done by connecting a 8-60 V supply to PVDD. Once properly connected, the controlCARD LED lights up and LED1 and LED3 on the control board also light up.

A summary of this step is:

- Open packaging of Piccolo CC2803 microcontroller
- Move SW3 to DOWN to turn ON as indicated in figure 4
- Move the SW2 switches in the UP positions to ON as indicated in figure 4
- Place controlCARD in the connector slot J1 on the control board and secure by pushing the controlCARD down and snap the controlCARD with white clickers
- Connect power supply wire to PVDD terminal and power supply ground to GND using the two connections closest to the large capacitor between them (PVDD and GND of the board are next to the motor connections)

Software Setup

1. Code Composer Studio

In order to program the Piccolo controlCARD, the user must install code composer studio that can be found on TI’s website. This can be found at [http://www.ti.com/tool/ccstudio](http://www.ti.com/tool/ccstudio) and downloaded for free as the last order option. A user application note can be found the ECE: 480 Fall 2013; Team 8: RC car drone webpage. The page URL is [http://www.egr.msu.edu/classes/ece480/capstone/fall13/group08/](http://www.egr.msu.edu/classes/ece480/capstone/fall13/group08/) and can be found under the documents section and is called *Code Composer*.

2. ControlSUITE

Once Code Composer is installed, the user can install controlSUITE which is found in the accompanying flash drive that came with the DRV8301 kit. ControlSUITE centralizes the location for all C200 software and organizes it in a simple file structure[4]. The device package here is the Piccolo F2803. In order to setup the controlSUITE software, the installer must be run. There are several dialogs that require user input. The steps that should be taken are:

- Accept End User License Agreement
- Select an installation directory. (C:/TI) is the default option and the preferred location
- Select the corresponding device support package to match the one that was bought (Piccolo 2803 in this case)
- Select extra libraries to install if desired

Please refer to the “DRV830x-HC-C2-KIT” How to Run Guide for help with the controlSUITE setup[5]. This guide explains the steps necessary for running the DRV8301 EVM with the software supplied through controlSUITE. It covers some points that this application note will not. The main focus is to use a setup similar to Code Composer but for controlSUITE. This file should be reviewed prior to performing the next step.

Once Code Composer is opened with controlSUITE installed for the appropriate device (TMS320F28035), the user should perform these steps:

- Connect computer to controlCARD with USB connection (Connect USB to J1 on Piccolo controlCARD)
- Open the correct directory (This was set to C:\ti\controlSUITE\development_kits\DRV830x-HC-C2-KIT_v14 in this example and is the default setting)
- Open and select Project Explorer tab
- Select InstaSPIN_BLDC as the active project
- Set build level to 1 and right click project name and select “Rebuild Project”
- Launch a debug session to load the code into the controller once project is rebuilt. This can be found in the upper left hand corner and looks like a green bug
- Open a watch window and add the variables shown in figure 6 [6]
More information on where the watch expressions tab and debug are found can be examined in the application note *Code Composer* found on the team website. This is the final step in the test setup and covers this application note.

**Conclusion**

Once the previous section is successfully completed, the motor is now available for testing with programming code. The testing phase can effectively be broken down into 8 different phases to test 11 different modules. This is covered in detail by the application note *Sensorless Trapezoidal Control of BLDC Motors using BEMF Integration (InstaSPINE-BLDC)*[6]. The test set up described in this application note covers all the necessary steps to make sure that the Piccolo MCU can be programmed and send the proper signals to drive the motor through the inverter located on the DRV board.

With this application note the user will be successful in learning the background material needed for controlling BLDC motors, setting up the hardware to test the motor and setting up the software while eliminating the need to search various application notes and user guides while presenting information other user guides/application notes have missed. The application note offers extended details on different parts by presenting other application notes that can be found in the references as well. If you have completed this application note to the letter, you have successfully set up testing of a BLDC motor for PWM using the DRV8301 EVM.
References


