Motor Control Closed-loop System REFERENCE GUIDE

RD022-RGUIDE-01-E

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Table of Contents

1. INTRODUCTION	4
2. FEATURES	5
3. SPECIFICATIONS	5
4. EXTERNAL VIEW	6
5. WIRING CONNECTION	7
6. SWITCHES	8
6.1. Potentiometer	8
7. MCD+MCU MODE	9
7.1. Block Diagram in Case of MCD+MCU Mode	9
7.2. Hardware Connection	9
7.3. Jumper Setting	10
7.4. Arduino Connection	11
7.5. LCD Shield	12
7.6. Usage	13
7.7. Software on Arduino UNO	14
7.8. Arduino Sketch	14
7.9. Connection with Arduino UNO	16
7.10. Code Uploading to Arduino UNO	17
7.11. Serial Monitor of Arduino IDE	18
7.12. How to Change Kp in Serial Monitor	19
7.13. Selected LCD Shield	20
7.14. Motor Speed Control for MCD + MCU	21
8. MCD+OP-AMP MODE	24
8.1. Block Diagram In Case of MCD+Op-AMP	25
8.2. Usage	26
8.3. OPERATING PROCEDURE	27

RD022-RGUIDE-01-E

8.4. Speed Control Part	28
8.5. Auto Lead Angle Setting	29
9. ATTACHED BRUSHLESS MOTOR PARAMETERS	30

1. INTRODUCTION

This reference guide describes the usage of Motor Control Closed-loop system incorporating Toshiba's BLDC motor driver IC TB6605FTG. The board is designed as a plug-in board (Shield) for Arduino UNO platform. This document provides guidelines to quickly setup the hardware and software for BLDC motor driver IC performance evaluation.

This system can be used as two types:

- 1. MCD + MCU mode
- 2. MCD + Op-Amp mode

Motor Control Closed-loop System evaluation kit includes:

- Motor Control Reference Board (with Arduino connector)
- * Arduino is not attached.
- Brushless Motor (attached)



Figure 1-1 Motor Control Closed-loop system

In MCD+MCU mode, the following items are needed for evaluation.

-MCU Arduino (Not attached, Commercial goods) -LCD module (Commercial goods)

2. FEATURES

- Support three-Hall sensor BLDC motor driver
- Motor control function: Start, Brake, CW/CCW function
- Tune motor rotation speed with potentiometer
- A reference motor is attached for quick start
- Replaceable resistors and capacitors are used to adopt other motors
- Two work modes are available:
 - MCD+MCU mode
 Designed as plug-in Shield for Arduino UNO platform
 Speed closed-loop control is implemented based on PID control in software.
 Arduino example program (Sketch) is provided.
 - MCD + Op-Amp mode
 Speed closed-loop control is implemented using Op-Amp on the board.

3. SPECIFICATIONS

Parameter	Description
TB6605FTG	Sine Wave BLDC motor controller
Power supply voltage	10V-28V DC
Attached motor specifications	Middle speed motor 10V-24V/ 0.5A-10A/
	Output 26W/ 4000rpm
Motor driver output + MosFET	N ch+ Nch MosFET (Toshiba TK20P04M1)
Main ICs	TB6605FTG, TC75S51F, TK20P04M1
	(Arduino UNO is needed in MCD+MCU mode.)
MosFET specifications and	TK20P04M1: NMOS, 40V/20A, DPAK, Rdson=19m Ω
package information	

Table 3-1 Specifications

4. External View



Motor switch



5. Wiring Connection



Figure 5-1 Wiring Connection

If an attached motor does not rotate, check if the current is lower than 0.02A.

Note:

Phase U/V/W line of motor should be connected with U/V/W on reference model. Do not mix Hall U/V/W line and Phase U/V/W line.

6. Switches

The Closed-loop system includes:

- Start switch for starting or stopping motor rotation
- Brake switch for stopping motor rotation in an emergency
- CW/CCW switch for setting rotation direction
- Speed control by potentiometer (see 6.1)

The settings are shown in below:

START	BRAKE	Mode
L	Н	Active/Normal
L	L	Active/Brake
Н	H/L	Standby

	Н	L
CW/CCW	CCW	CW
OVP	Sine wave	Square wave



3100RPM CW

3100RPM S

С

Figure 6-1 Switches

6.1. Potentiometer

Motor speed can be controlled using potentiometer.



In MCD + Op-Amp mode: Target speed increases in a counter-clockwise direction

In MCD+MCU mode: Target speed increases by 100RPM in a counter-clockwise direction. LCD displays target speed is shown as below:





4

7. MCD+MCU Mode

7.1. Block Diagram in Case of MCD+MCU Mode



Figure 7-1 Block Diagram in MCD+MCU Mode

7.2. Hardware Connection



LCD shield Arduino UN TOP BOTTOM MCD shield MIDDLE



7.3. Jumper Setting



Figure 7-3 Jumper Setting

Jumper	Setting	Function
JP1	Open	VREG(reserved)
JP2	Open	IDC: no connection with MCU (for future use)
JP3	Short pin2 and pin3	PWM-IN: input from MCU for speed control
JP4	Short pin2 and pin3	HP: hall sensor pulse output to MCU for detecting rotation speed
JP5	Short pin2 and pin3	DIF-IN: GND
JP6	Open	LA: not used (for future use)
JP7	Open	IDC: no connection with MCU (for future use)

7.4. Arduino Connection

	발명::::::::::::::::::::::::::::::::::::
9	
8	
+ ⊙√ Genui	104
	VIVITOR IN VIVITOR IN VIVITO
	6

Analog	oins/Digital pins	
Pin	Function	I/O
A0	Key(LCD)	1
A1	VSP*	1
A2	IDC	I.
A3		
A4	SDA(LCD)	I/O
A5	SCL(LCD)	0

Pin	Function	I/O
D0	Rx(Serial)	I
D1	Tx(Serial)	0
D2	HP	i i
D3	PWM IN	0
D4	CW/CCW	0
D5		
D6		0
D7	START	0
D8	BRAKE	0
D9		
D10		
D11	LA(PWM)	0
D12		
D13		

A0, A4, A5: for connection with LCD shield D0, D1: for serial Tx/Rx, debug log out and set target speed

xxx: used for motor control IDC, LA are not used in the current solution.

Figure 7-4	Arduino	Connection
------------	---------	------------

7.5. LCD Shield



Figure 7-5 LCD Shield

7.6. Usage

Perform steps (1) to (11) in order. This system includes a 42BLF BLDC motor manufactured by ACT motor. Motors must be a Hall sensor BLDC type for normal working.

Table 7-2 Usage for MCD+MCU mode

Step	Item
(1)	Check the R/B exterior, set the switch correctly as described in 6.
(2)	Connect the motor to the board. (see 5)
(3)	Insert the R/B board to Arduino and insert the LCD board to the R/B board.
(4)	Connect the stabilized power supply to the R/B board.
(5)	Check the motor control switch. Set the start switch to "H". Set the brake switch to "H". Set the OVP switch as desired ("L": square wave, "H": sine wave). Set the CW/CCW switch as desired rotation direction. ("L": CW, "H": CCW)
(6)	Supply power to the R/B board.
(7)	Connect PC with Arduino board via USB cable. Upload the attached sketch to Arduino. (See 7.4)
(8)	Tune potentiometer on R/B board to set target speed which is displayed on LCD.
(9)	Start motor rotation by changing the start switch to "L". Check if motor rotation speed displayed on LCD can achieve target speed.
(10)	Tune potentiometer on R/B board to change target speed which is displayed on LCD. Check if motor rotation speed displayed on LCD can achieve to the target speed.
(11)	Stop motor rotation by changing Start switch to "H".

To rotate the motor, shift the start switch of the shield to "L". Current rotation speed is displayed as "C" parameter of the LCD module.

After rotation is started, rotation speed is controlled to maintain the current speed even power supply voltage is changed.

7.7. Software on Arduino UNO

The latest Arduino Software (IDE) can be downloaded here:

https://www.arduino.cc/en/Main/Software

Read the appropriate procedure for your system. Please download and install it. Run the Arduino IDE and open the provided example sketches.



ARDUINO 1.8.7

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the <u>Getting Started</u> page for Installation instructions. Windows Installer, for Windows XP and up Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits Linux 64 bits Linux ARM

Release Notes Source Code Checksums (sha512)

7.8. Arduino Sketch

A sketch is the name that Arduino uses for a program. This system needs a sketch "closed_loop_mcd.ino" to implement closed-loop system.

Figure 7-6 Arduino IDE

closed_loop_mcd.ino includes 3rd party libraries:

- #include <LiquidCrystal_l2C.h>
- #include <TimerOne.h>
- * Refer to next page for how to install 3rd party libraries.

LiquidCrystal_I2C: Display characters on LCD shield TimerOne: Use hardware Timer 1 for running an periodic interrupt function

7.8.1. How to Install

- 1. Unzip LiquidCrystal_I2C.zip and TimerOne.zip
- 2. Copy folder LiquidCrystal_I2C and TimerOne to "...\Documents\Arduino\libraries"
- 3. Check if library is installed:
 - Arduino IDE->Sketch->Include Library-> LiquidCrystall2C Arduino IDE->Sketch->Include Library-> TimerOne

💿 clo	closed_loop_mcd Arduino 1.8.5					
File Edit Sketch Tools Help						
	Ð	Verify/Compile	Ctrl+R			
		Upload	Ctrl+U			
clo	sed	Upload Using Programmer	Ctrl+Shift+U			
1	/*	Export compiled Binary	Ctrl+Alt+S			
2	1	Show Sketch Folder	Ctrl+K	osed-loop speed control implemented on Arduino UNO board		
3		Include Library	۱.	and a LCD shield is stacked shows MCD shield		
5		Add File		land, and a LOD Shield is stacked above mob Shield.		
6	Use	r can do following thin	g by pressing	y key on MCD board.		
7	1	. Start, stop or brake m	otor	,,		
8	2.	. Change the direction	of motor rot:	ation (CW or CCW)		
9						
10	Use	r can understand curren	t status by n	reading info display on LCD.		
11	1.	. Iarget speed (tuned by	y potentiomet	er)		
12	2.	. Current speed detecte	d by HP			
13	3.	. Motor status: start,	stop or brake			
14	4	. The direction of moto	r rotation ((;W/CCW)		
15	5. Kp value					
16						
17	A p	otentiometer located on	MCD shield :	is connected to analog input 1.		
18	The	higher the potentiomet	er value, the	faster of the motor speed.		
19						
20	Cre	ated on 6 Mar. 2018				
21	1 Modified on 20 Jun. 2018					
22	*/					
23	#4	uda (LianidCanatal T2	C b			
25	24 Finclude (Liquidurystal_120.h)					
26	6					
27	20 27 LionidCrystal T2C lcd(0x27 16 2)					
28			,,			
29	9 //Digital port					
30	0 #define PORI_HP 2 //D2					
31	1 #define PORI_PWMIN 3 //D3					
32	32 #define PORI_DIR 4 //D4					
33	3 #define PORI_LA 6 //D6					
34	4 #define PORI_SIARI 7 //D7					
35	#defin	ne PORI BRAKE 8 //D8				

Figure 7-7 Install

7.9. Connection with Arduino UNO

- 1.
- Open closed_loop_mcd.ino using Arduino IDE Connect the PC with Arduino board with USB cable 2.
- 3. Select Board and Port

File Edit Sketch Tools Help Auto Format Ctrl+T Archive Sketch Closed_loop_n 1 /* 2 Ioshiba 3 4 A MCD s 5 A BLDC 6 User ca Board: "Arduino/Genuino Uno" Board: "Arduino/Genuino Uno" Ctrl+Shift+L Control implemented on Arduino UNO board shield is stacked above MCD shield. rd.	a closed loop mod LArdvino 185						
Auto Format Ctrl+T Closed_loop_n 1 /* 2 Ioshiba 3 4 A MCD s 5 A BLDC 6 User ca 6 User ca 1 /* CM2 (Arduino/Genuino Uno" 5 A BLDC							
Auto Format Ctrl+1 Archive Sketch Fix Encoding & Reload Serial Monitor Ctrl+Shift+M Serial Plotter Ctrl+Shift+L 4 A MCD s 5 A BLDC 6 User ca Board: "Arduino/Genuino Uno" 6 User ca Board: "Arduino/Genuino Uno"							
closed_loop_n Archive Sketch 1 /* Serial Monitor Ctrl+Shift+M 2 Ioshiba Serial Plotter Ctrl+Shift+L 3 WiFi101 Firmware Updater control implemented on Arduino UNO board 5 A BLDC Board: "Arduino/Genuino Uno" initial is stacked above MCD shield. 6 User ca Board: "Arduino/Genuino Uno" initial is stacked above MCD shield.							
Initial Security Control Fix Encoding & Reload 1 /* 2 Ioshiba 3 Serial Monitor 4 A MCD s 5 A BLDC 6 User ca 8 Rett "Comparison" (Seriar Control implemented on Arduino UNO board shield is stacked above MCD shield.							
1 /* Serial Monitor Ctrl+Shift+M 2 Ioshiba Serial Plotter Ctrl+Shift+L 3 WiFi101 Firmware Updater control implemented on Arduino UNO board 4 A MCD s WiFi101 Firmware Updater 5 A BLDC Board: "Arduino/Genuino Uno" 6 User ca Board: "Arduino/Genuino Uno"							
2 Ioshiba Serial Plotter Ctrl+Shift+L control implemented on Arduino UNO board 3 A MCD s WiFi101 Firmware Updater shield is stacked above MCD shield. 5 A BLDC Board: "Arduino/Genuino Uno" rd. 6 User ca Board: "Arduino/Genuino Uno" rd.							
3 WiFi101 Firmware Updater shield is stacked above MCD shield. 5 A BLDC Board: "Arduino/Genuino Uno" 6 User ca Board: "COM2 (Arduino/Genuino Uno"							
4 A MCD s wirld rinnware opdater shield is stacked above MCD shield. 5 A BLDC Board: "Arduino/Genuino Uno" 6 User ca Bott "COM2 (Arduino/Genuino Uno"							
6 User ca Rott "COM2 (Arduino/Genuino Uno"							
6 User ca Port "COM2 (Arduino (Genuino Uno)"							
For COM2 (Arduno/ Genuno Ono)							
7 1. St Get Board Info							
8 2. Cn							
Programmer: "AVRISP mkII"							
10 User ca Burn Bootloader pray on LCD.							
1 1. larget speed (tuted by potentioneter)							
12 2. Conferr spece detected by In 13 3 Motor status: start start or nor brake							
4 The direction of motor rotation (CW/CCW)							
15 5 Kn value	4. Ine alrection of motor rotation (UW/UUW)						
e of varme							
17 A potentiometer located on MCD shield is connected to analog input 1.							
18 The higher the potentiometer value, the faster of the motor speed.							
19							
20 Created on 6 Mar. 2018	0 Created on 6 Mar. 2018						
21 Modified on 20 Jun. 2018							
22 */							
23							
24 #include <liquidcrystal_i2c.h></liquidcrystal_i2c.h>							
25 #include <timerone.h></timerone.h>							
26							
<pre>17 LiquidCrystal_I2C lcd(0x27, 16, 2);</pre>							
.8							
//Digital port							
) #define PORI_HP 2 //D2							
1 #define PORI_PWMIN 3 //D3							
# define FUKI_DIR 4 //D4							
/ +ucline Forn_LA 0 //D0							
35 #define PORT RRAYE & //D&							

Figure 7-8 Connection with Arduino UNO

7.10. Code Uploading to Arduino UNO

- 1. Verify/compile code till no errors reported
- 2. Upload program to Arduino board by USB cable

cl	osec	l_loop	p_mcd Arduino 1.8.5		
ile	Edit	Ske	tch Tools Help		
	0		Verify/Compile	Ctrl+R	
-	-		Upload	Ctrl+U	
clo	sec	1	Upload Using Programmer	Ctrl+Shift+U	
1	/*		Export compiled Binary	Ctrl+Alt+S	
2		1	Show Skatab Falder	Chilly K	
3			Include Library	CIII+K	
4		A	Add Eile		
0		H	and following thin	a bu prossir	
7		1.	Start. stop or brake m	g by piessii otor	
8		2.	Change the direction	of motor rot	
9					
10		User	can understand curren	t status by	
11		1.	Target speed (tuned by	y potentiome	
12	12 2. Current speed detected by HP				
13	3 3. Motor status: start, stop or brake				
14		4.	The direction of moto	r rotation	
15		5.	Kp value		
16					
17	7 A potentiometer located on MCD shield is connected to analog input 1.				
18		The !	higher the potentiomet	er value, tł	
19			4. J 6 M 2019		
20		Modi [.]	fied on 20 Tun 2018		
22	*/	mour	rica on 20 juli. 2010		
23					
24	#i	nclu	de < LiquidCrystal_I2	C.h>	
25	#i	nclu	de < TimerOne.h >		
26					
27	Li	qui	dCrystal_I2C lcd(0x27	, 16, 2);	
28					
29	//1	Digi	tal port		
30	#d	efin	e PORI_HP 2 //D2		
31	#d	efin	e PORI_PWMIN 3 //D3		
32	#d	efin	e PORT_DIR 4 //D4		
33	#d	efin	e PURI_LA 6 //D6		
34	#d	efin efir	<pre>e FURI_SIARI (//D) </pre>		

Figure 7-9 Code Uploading to Arduino UNO

7.11. Serial Monitor of Arduino IDE

Monitor Serial output for debugging

💿 closed Joop_mcd Arduino 1.8.5							
File Edit Sketch Tools Help							
	Auto Format	Ctrl+T					
	Archive Sketch						
closed_loop	-n Fix Encoding & Reload						
1 /*	Serial Monitor	Ctrl+Shift+M					
2 Ioshi	ba Serial Plotter	Ctrl+Shift+L cont	trol implemented on Arduino UNO board				
3 4 A MCD	s WiFi101 Firmware Updater	shie	eld is stacked above MCD shield.				
5 A BLD	Board: "Arduino/Genuino Uno"	•					
6 User	ca Port: "COM2 (Arduino/Genuino Uno)	" • rd.					
7 1.	St Get Board Info						
8 2.	Ch						
9 10 User	Programmer: "AVRISP mkII"	• nlaw	on ICD				
11 1.	Burn Bootloader	pray	on Lob.				
12 2.	Current speed detected by HP						
13 3.	Motor status: start, stop or brake						
© COM2 (Arduino/Genuino Uno)	OM2 (Arduino/Genuino Uno)						
			g input 1.				
== Input Kp:(1-9):			speed.				
V Autoscroll	No line ending 👻 9600 baud	✓ Clear output					

Figure 7-10 Serial Monitor

7.12. How to Change Kp in Serial Monitor

A Kp value can be changed using Serial Monitor for PI control.

1. Input Kp value	2. Click Send button ^{3.}	Kp is changed
COM2 (Arduino/Genuino Uno)	🕞 💷 🔁 💿 сом2	Arduino/Genuino Uno)
5	Send	Send
== Input Kp:(1-9):	== Input Kp: 5	t Kp: (1-9):
V Autoscroll No line ending	10 baud V Clear output	croll No line ending 🔹 [9600 baud 🔹 Clear output

When a Kp value is changed, upload the revised sketch to Arduino again.

Figure 7-11 Changing Kp value

7.13. Selected LCD Shield

– HW

Use three signal pins of Arduino connectors.

– SW

Library "LiquidCrystall2C" is dedicated for the LCD shield in this R/B.

7.13.1. Other Type Shield

– HW

May use more signal pins of Arduino connectors. Please refer to pin assignment in 7.9. Do not use pins for motor and serial.

– SW

Use a library provided by vender of your LCD shield and modify the software accordingly.

7.14. Motor Speed Control for MCD + MCU

PID control is used to implement closed-loop control through motor speed feedback to achieve target speed in case MCD controlled by Arduino UNO.

7.14.1. PID Algorithm

Apply incremental PI control, and D is not used for standard PID calculation.

 $CV(t) = Kp*E(t)+Ki* \int E(t)dt+Kd*(dE(t)/dt)$ simplified incremental PI calculation

 $CV(t) = CV(t-1) + Kp^{*}(E(t)-E(t-1)) + Ki^{*}E(t)$ E(t) = Set point - Input

- Input Process variable
 Motor rotation speed detected by hall sensor (HP signal)
- Output Control variable
 PWM duty (0-255, low active) (PWM IN signal)
- Set point
 Target speed set by tuning potentiometer.

7.14.2. PID Control

P (Proportional term): $CV(t) = Kp^*E(t)$

If Proportion Gain Kp is fixed, controller output (CV) is proportional to error E (t). If error is bigger, CV will become bigger automatically. If Kp is too big, the system will become unstable, on the other, if Kp is too small, the system will become less responsive.

Because the control output (CV) of Proportional term depends on error between SP and PV, a steady-state error (droop) is exist if use Proportional control only. To mitigate the droop, a compensating bias term (called offset) need to be set in advance. The best measures is add an Integral term into system.







If Kp is increasing:

- Rise time: decrease
- Overshoot: increase
- Settling time: small change
- Droop: decrease
- Stability: degrade



Figure 7-13 Increase of Kp Value

If Ki is increasing:

- Rise time: decrease
- Overshoot: increase
- Settling time: increase
- Droop: Eliminate
- Stability: degrade



Figure 7-14 Increase of Ki Value

7.14.3. PI Tuning Procedure

- 1. Keep Ki and Kd values to zero at first, increase the Kp until the output of the loop oscillates. Then the Kp should be set to about half of that value for a "quarter amplitude decay" type response.
- 2. Then increase Ki until any offset is corrected in sufficient time for the process. However, too much Ki will cause instability.
- 3. The tuning is a repeat process to approach the balance between the response speed and system stability.

7.14.4. Source code

```
err = rotateSpeedHp - detectedHp;
                                       //error = set value - actual value
  if (detectedHp == 0) {
    bias = Kp s * (err - err1); // do once only at startup when speed is zero
  }
  else
  {
    bias = Kp * (err - err1) + Ki * err;
    if (err < 2 \&\& err > -2)
    {
       pidOver_U = pidOver2 U;
       pidOver_D = pidOver2_D;
    }
    else
    {
       pidOver U = pidOver1 U;
       pidOver_D = pidOver1_D;
    }
    if (bias > pidOver U)
       bias = pidOver_U;
    else if (bias < pidOver_D)
       bias = pidOver D;
  }
```

8. MCD+Op-Amp Mode

In MCD + Op-Amp mode, the speed control part realizes the closed loop by using PI control.

- The VSP is used to determine the target speed.
- The LA voltage is used as a feedback signal to represent current speed.
- The Dif-in is used to adjust output duty to control motor speed.
- PWM-in should be connected to GND.



Figure 8-1 MCD + Op-Amp Mode

8.1. Block Diagram In Case of MCD+Op-AMP



Figure 8-2 Block Diagram In Case of MCD+Op-Amp

8.2. Usage

Perform steps (1) to (8) in order. The product includes a 42BLF BLDC motor manufactured by ACT motor. A motor must be a Hall sensor BLDC type for normal working.

Table 8-1 Usage for MCD + Op-Amp mode

Step	ltem			
(1)	Check the R/B exterior, set switch correctly as described in 6.			
(2)	Connect the motor to the board. (see 5)			
(3)	Connect the stabilized power supply to MCD board.			
(4)	Check the motor control switch. Set the start switch to "H". Set the brake switch to "H". Set the OVP switch as desired. ("L": square wave, "H": sine wave). Set the CW/CCW switch as desired rotation direction. ("L": CW, "H": CCW)			
(5)	Supply power and check if the power LED is ON.			
(6)	Set the start switch to "L" to start rotating the motor.			
(7)	Tune potentiometer to change a motor rotation speed.			
	Check how motor rotation speed changes by monitoring Test point "HP".			
(8)	Set the start switch to "H" to stop rotating the motor.			

If an Error Occurs

- Power supply LED no "lighting": Check power supply cable and voltage.
- Motor no rotation: Check motor connection and Motor Control switch set.

8.3. OPERATING PROCEDURE

8.3.1. HW setting for MCD + Op-Amp

8.3.2. Jumper Setting



Figure 8-3 Jumper Setting

Table 8-2 Jumper Setting

Jumper	setting	Function
JP1	Open	VREG(reserved)
JP2	Open	IDC: no connection with MCU
JP3	Short pin1 and pin2	PWM-IN: GND
JP4	Open or short pin1 and pin2	HP: not connection with MCU
JP5	Short pin1 and pin2	DIF-IN: OpAmp output for speed control
JP6	Short pin1 and pin2	LA: OpAmp input
JP7	Open	IDC: not connection with MCU

8.4. Speed Control Part

Use a rail-to-rail Op-Amp.



Figure 8-4 Speed Control Part

8.4.1. Parameter adjustment

R1 and R2: if VSP=VMAX represents max speed, and LA=VLAMAX at max speed.

Adjust R1 and R2 so that *LAMAX=VMAX×R2/(R1+R2)*

R3: The factor of proportional control Kp is related to R3. Increase R3 to increase Kp.

C1: The factor of integral control Ki is related to C1. Increase C1 to increase Ki.

R5 and R6 : Adjust R5 and R6 to ensure Dif-in will not exceed 3V.

If Dif-in=VMIN makes motor run at min speed, it is recommended to set R5 and R6 so that $MIN<5\times R6/(R5+R6)<3$.

Total resistant of R5 and R6 should be around 50k to 100k.

8.5. Auto Lead Angle Setting

For high efficiency, TB6605FTG needs to be set a lead angle parameter. When you use a different motor, change a lead angle parameter if necessary. And also change the TCR parameter for good lead angle value.



Figure 8-5 Auto Lead Angle Setting

Usage Notes

- To ensure that power supply is at the range of 10V to 28V, and current is under 2A (limited by Motor spec).
- At high current working states, motor and control board may have a high temperature.
- Some of the unused pins of the product have not been processed in any way.
 Make sure to process these pins as appropriate for your system.
- Resistors and capacitors in pin sockets can be changed if necessary when other than the attached motor is used.

9. ATTACHED BRUSHLESS MOTOR PARAMETERS



Terms of use

This terms of use is made between Toshiba Electronic Devices and Storage Corporation ("We") and customers who use documents and data that are consulted to design electronics applications on which our semiconductor devices are mounted ("this Reference Design"). Customers shall comply with this terms of use. Please note that it is assumed that customers agree to any and all this terms of use if customers download this Reference Design. We may, at its sole and exclusive discretion, change, alter, modify, add, and/or remove any part of this terms of use at any time without any prior notice. We may terminate this terms of use at any time and for any reason. Upon termination of this terms of use, customers shall destroy this Reference Design. In the event of any breach thereof by customers, customers shall destroy this Reference Design. customers shall destroy this Reference Design, and furnish us a written confirmation to prove such destruction.

1. Restrictions on usage

- This Reference Design is provided solely as reference data for designing electronics 1. applications. Customers shall not use this Reference Design for any other purpose, including without limitation, verification of reliability. This Reference Design is for customer's own use and not for sale, lease or other transfer.
- 2. Customers shall not use this Reference Design for evaluation in high or low temperature, 3. high humidity, or high electromagnetic environments.
- This Reference Design shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations. 4.

5 2. Limitations

- 1.
- We reserve the right to make changes to this Reference Design without notice. This Reference Design should be treated as a reference only. We are not responsible for 2. any incorrect or incomplete data and information.
- Semiconductor devices can malfunction or fail. When designing electronics applications 3. by referring to this Reference Design, customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of semiconductor devices could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Customers must also refer to and comply with the latest versions of all relevant our information, including without limitation, specifications, data sheets and application notes for semiconductor devices, as well as the precautions and conditions set forth in the "Semiconductor Reliability Handbook".
- the precautions and conditions set forth in the "Semiconductor Reliability Handbook".
 When designing electronics applications by referring to this Reference Design, customers must evaluate the whole system adequately. Customers are solely responsible for all aspects of their own product design or applications. WE ASSUME NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.
 No responsibility is assumed by us for any infringement of patents or any other intellectual property rights of third parties that may result from the use of this Reference Design.
- Design. No license to any intellectual property right is granted by this terms of use,
- Design. No license to any intellectual property right is granted by this terms of use, whether express or implied, by estoppel or otherwise. THIS REFERENCE DESIGN IS PROVIDED "AS IS". WE (a) ASSUME NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (b) DISCLAIM ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO THIS REFERENCE DESIGN, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABLI ITY FITNESS FOR A PARTICULAR PURPOSE 6. CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.