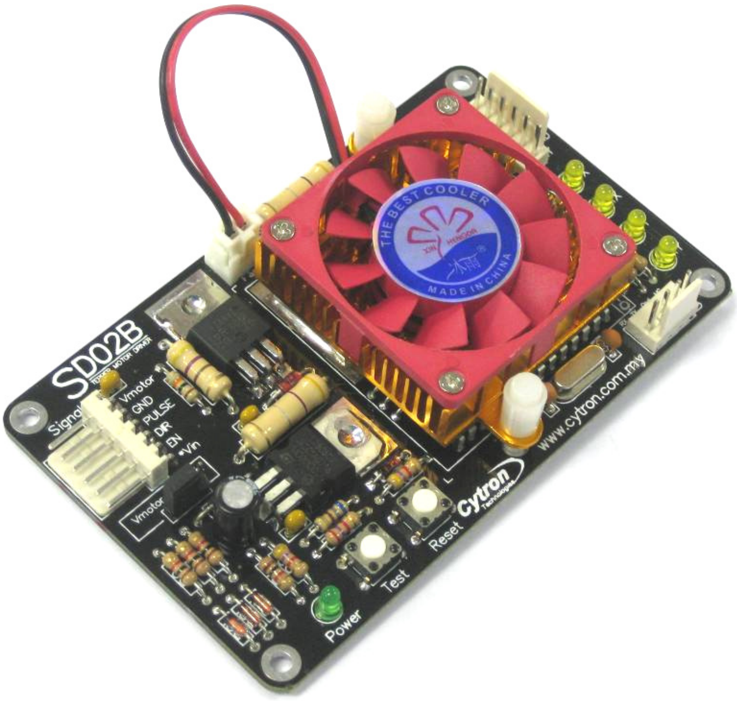
Cytron Technologies

SD02B
2A Stepper Motor Driver



User's Manual

V1.2

July 2013

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Index

1. INTRODUCTION AND OVERVIEW

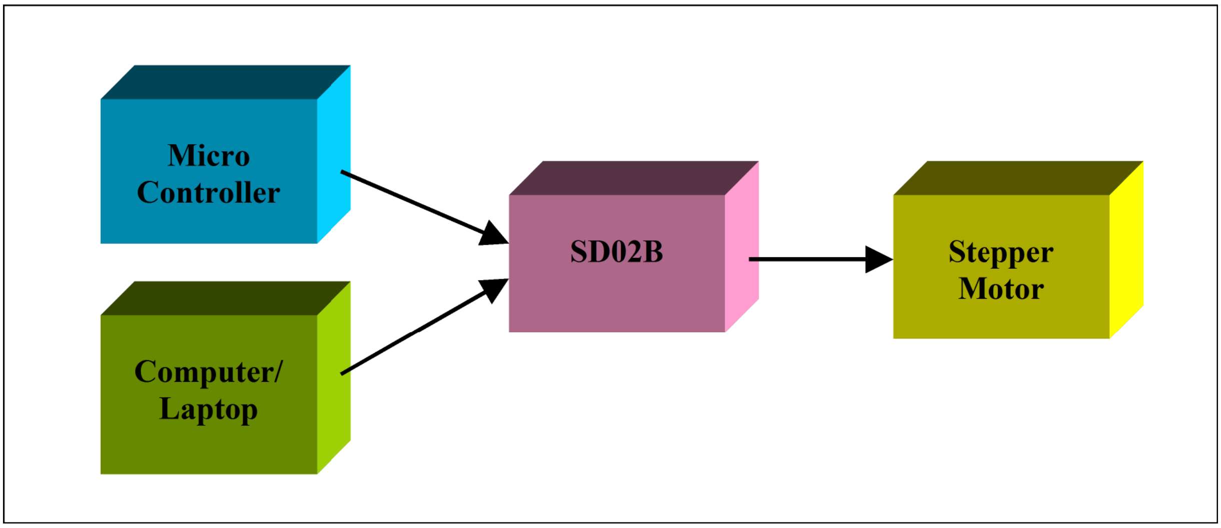
1.0 Introduction of SD02B

SD02B is the enhanced version from SD02A. SD02B is designed to drive unipolar or bipolar stepper motor. The board incorporates most of the components of the typical applications. With minimum interface, the board is ready to Plug and Play. Simply add in power and a few push buttons, this driver is ready to drive unipolar or bipolar stepper motor. SD02B will actually drive stepper motor in bipolar method. However, since unipolar stepper motor can also be used as bi-polar stepper motor, thus this driver can be used to drive both unipolar and bipolar stepper motor. This stepper motor driver has been designed with capabilities and features of:

- New! Now comes with UART interface for easier communication between the user's circuit (or PC) and SD02B. By using the new UART control, user can
  - On/Off, Run/Brake and change motor's rotation direction
  - Set motor speed
  - Request for encoder value
  - Track an encoder value and brake the motor
  - Set new baud rate for the driver
  - ✓ Not familiar with UART communication? Fret not, sample source code for UART control is provided for PIC18F4520 (C language), Arduino and VB.net!
- Support up to 1A per phase.
- Smoother stepper motor rotation with 2, 5 and 10 micro-stepping.
- Able to drive stepper motor from 3V to 40V.
- 8V to 24V compatible for driver circuit supply voltage.
- 5V logic level compatible inputs.
- Detect up to 20kHz pulses at PULSE input pin.
  However, the maximum speed a stepper motor can step is also depending on the motor's performance. For stepper motors available in Cytron store, the max stepping speed can be up to 2-4kHz without load (free run).
- Enable/Disable pin for low power consumption mode.
- Heat sink with fan for fast thermal release.
- UART baud rate and micro-stepping mode are saved in the non-volatile flash memory (Factory default is baud rate = 9600, micro-stepping mode = 10).

Dimension 10cm x 6cm

1.1 System Overview



SD02B is the enhanced version from SD02A. One of the additional features for SD02B compare to SD02A is the UART interface. UART is provided in SD02B for easier communication between user's circuit (microcontroller/Arduino) and SD02B. Besides user's circuit, computer (PC or Laptop) is ready be used for UART communication with SD02B.

By using UART function in SD02B, user is able to have more extensive control of the motor driver and stepper motor. User can leave most part of the motor control algorithm to SD02B and let it takes care of the encoder supervisory job. It saves much software resources on the host.

For example by sending some commands, SD02B is able to accelerate the stepper motor from the specified initial speed to final speed at desired acceleration rate. While the stepper motor is accelerating, the host may take care of other routines. In order to know the stepper motor has been accelerated to the final speed, the host can send another command to check the status of SD02B. Besides, host can brake the stepper motor at the desired step by just sending a command together with a 16-bit encoder value (send in two 8-bit values) to SD02B and let it counts the steps for you.

Besides that, the UART baud rate and micro-stepping mode are stored in the non-volatile flash memory. Those settings only need to be configured once and will remain the same even though the power has been turned off. Please refer to section 7.3, SD02B UART Protocol for the UART commands and descriptions.

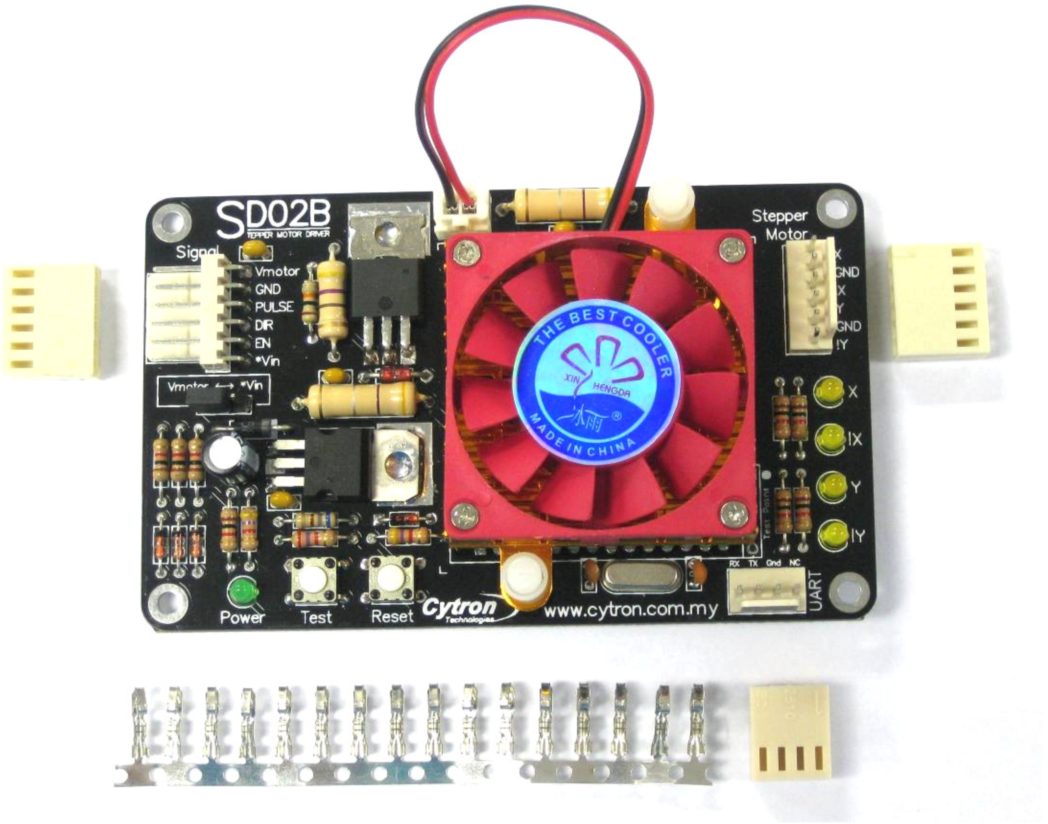
1.2 General Description

Driving stepper motor is common necessity in most robotic project. A stepper motor is a brushless, synchronous electric motor that can drive a full rotation into a large number of steps. Stepper motor is ideally suited for precision control. This motor can operate in forward/reverse with controllable speed from a microcontroller through a transistor driver circuit. There are various kinds of stepper motor. Some example are variable reluctant stepper motor, permanent magnet stepper motor, bipolar/unipolar stepper motor, bifilar stepper motor and hybrid stepper motor. For more description of stepper motor, user is advice to refer to Wikipedia at: http://en.wikipedia.org/wiki/Stepper_motor. This driver, SD02B is to drive stepper motor in bipolar method.

SD02B is designed to drive stepper in 2, 5 or 10 micro steps. This will offer smoother rotation of the stepper motor. Of course, the step angle will depend on the specification of stepper motor being drive. As an example, if the stepper motor being drive is 1.8° per step and use 1/10 micro-stepping, the smallest step (1 pulse) is 0.18°. This will offer much better resolution (°/step).

2. PACKING LIST

Please check the parts and components according to the packing list. If there are any parts missing, please contact us at sales@cytron.com.my immediately.



SD02B comes with:

- 1 x SD02B board with every component is soldered properly and tested before board is shipped.
- 16 x 2510 iron pin
- 2 x 2510-06 female connector
- 1 x 2510-04 female connector
- User's Manual and sample source codes (in VB.net and PIC C language) can be downloaded from www.cytron.com.my

3. PRODUCT SPECIFICATION AND LIMITATIONS

Signal Input Pins Function Description

Label	Definition	Function
Vmotor	Motor Supply Voltage	Vmotor is one of SD02B power sources. If users choose Vmotor as power source for circuit (in Single Supply Mode), Vmotor will supplies power to both stepper motor and SD02B circuit. Although SD02B provides protection against wrong polarity for this input, user must ensure the voltage and polarity of connection are correct before providing the power so that SD02B can function correctly.
GND	System Ground	Common ground for both logic operation and stepper motor power source.
PULSE	Stepping Pulse Input Pin	Input to drive the stepper motor. This pin is TTL/CMOS logic (5V and 0V). Each pulse (logic change from 0 to 1) will drive the stepper motor 1 step. SD02B has 2/5/10 micro stepping.

DIR	Stepper Motor Rotating Direction Input Pin	Input for stepper motor to rotate CW (clockwise) or CCW (counterclockwise). This pin is TTL/CMOS logic (5V and 0V). The direction is depends on the connection sequence of stepper motor wires to SD02B. For example:
		DIR = 0V → CW
		DIR = 5V → CCW

EN	Stepper Driver Enable Pin	Input pin to enable SD02B. This pin is TTL/CMOS logic (5V and 0V). 5V will enable the motor driver further hold the shaft of stepper motor while 0V will disable the motor driver and release the shaft of the stepper motor. By default, the driver is disabled for power saving.
***Vin**	Circuit Supply Voltage	*Vin is another power source for SD02B. If users choose *Vin as power source for circuit (in Dual Supply Mode), *Vin will supplies power to SD02B circuit and Vmotor will supplies power to stepper motor. Although SD02B provides protection against wrong polarity for this input, user need to ensure the voltage and polarity of connection are correct before providing the power so that SD02B can function correctly.

Single Supply Mode and Dual Supply Mode are designed to support stepper motor with wider range of supplies. If the stepper motor being used is rated between 8 to 20V, Single Supply Mode can be used. If the stepper motor is rated below 8V or above 20V, Dual Supply Mode must be chosen. However, stepper motor with ratings beyond 3 to 40V is not support in any mode. Please refer to Absolute Maximum Rating tables in the next page for further information.

UART Pins Function Description

Label	Definition	Function
RX	SD02B UART Receive Pin	This is SD02B's UART receive pin. It should be interfaced to 5V logic UART. It should be connected to host's transmit pin.
TX	SD02B UART Transmit Pin	This is SD02B's UART transmit pin. It should be interfaced to 5V logic UART. It should be connected to host's receive pin.
GND	System Ground	Common ground for both logic operation and also stepper power source.
NC	Not Connected	Leave this pin disconnected.

Absolute Maximum Rating

No	Parameters	Max	Typical	Min	Unit
1	Output Current per phase, I_{phase}	2*	1	-	A
2	Case Operating Temperature, T_C	70	-	0	°C
3	Storage Temperature, T_{STG}	80	-	0	°C

* For momentary peak current. Driving at this condition may trigger overheat or over-current cut-off.

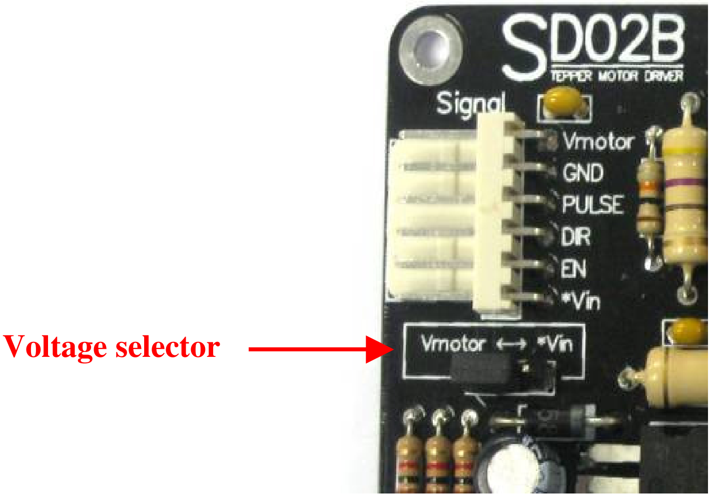
No	Parameters	Max	Typical	Min	Unit
	***Single Supply Mode (Vmotor)**				
4	Vmotor Voltage	20	-	8	V
5	Vmotor Current	2.53	-	0.11	A
No	**Parameters**	**Max**	**Typical**	**Min**	**Unit**
	****Dual Supply Mode (Vmotor & *Vin)**				
6	Vmotor Voltage	40	-	3	V
7	*Vin Voltage	20	-	8	V
8	Vmotor Current	2.35	-	0.11	A
9	*Vin Current	0.16	-	0.11	A

* Single Supply Mode (Vmotor) = Vmotor supplies power to both stepper motor and SD02B circuit.

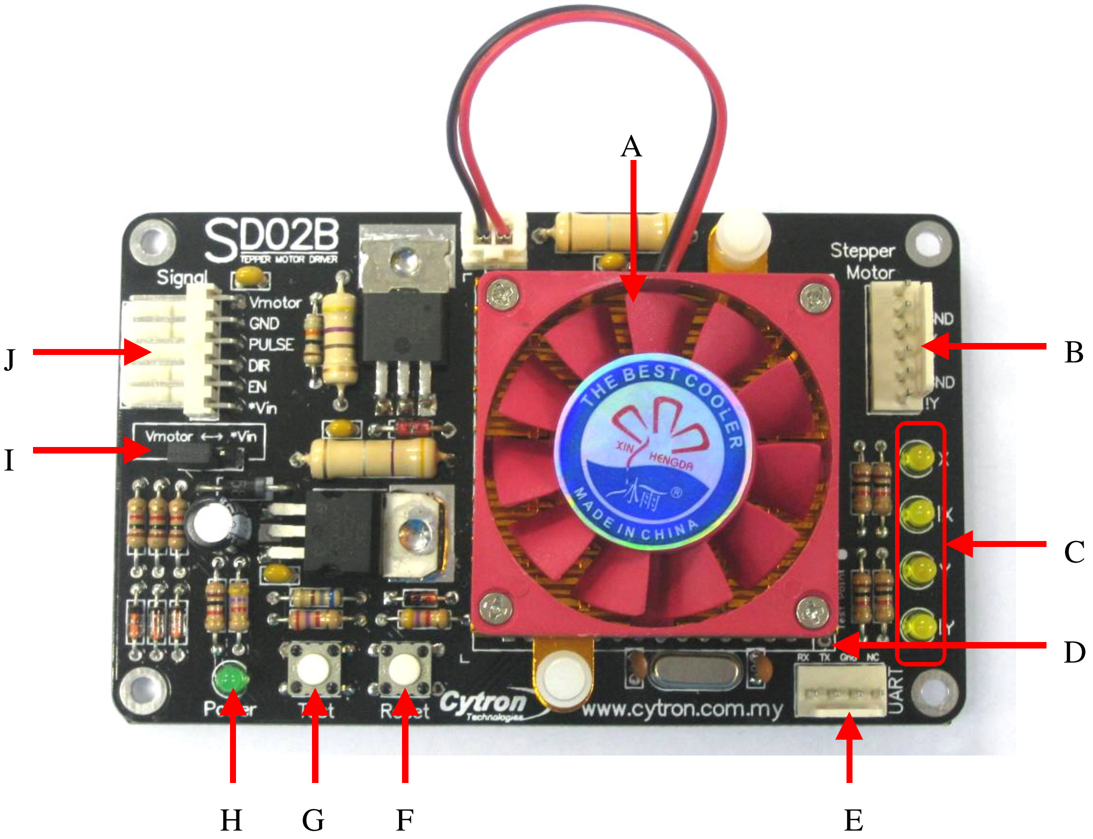
** Dual Supply Mode (Vmotor and *Vin) = Vmotor supplies power to stepper motor and *Vin supplies power to SD02B circuit.

Single Supply Mode and Dual Supply Mode

Voltage Selector is provided to choose either Single Supply Mode (Vmotor) or Dual Supply Mode (Vmotor and *Vin). If user selects Vmotor, Vmotor will supply power to both stepper motor and SD02B circuit. Otherwise if *Vin is selected, Vmotor will supply power to stepper motor and *Vin will supply power to SD02B circuit.



4. BOARD LAYOUT



Label	Function	Label	Function
A	Heat sink with fan	**F**	Push button to reset SD02B
B	Connector to stepper motor	**G**	Push button to test SD02B
C	Stepper motor coil power indicator (4 LEDs)	**H**	Small green LED as VCC indicator
D	Manufacturing test point	**I**	Voltage selector
E	UART communication	**J**	Connector to Host (Signal)

A – A heat sink with fan to dissipate the heat generated by SD02B.

B – A 2510 6 ways connector for user to connect particular stepper motor. Please solder the wire properly by referring to chapter 5, Hardware Installation.

C – 4 yellow LEDs acting as indicator for stepper motor coil. These LEDs will indicate which coil is being powered at a moment.

D – Manufacturing test point for SD02B. The header pin is not soldered since it is meant for production test point. Please DO NOT connect or shorted any of these pins.

E – UART communication between SD02B and microcontroller circuit or PC's Serial Communication Interface (SCI).

F – A push button acting as reset button to SD02B. If this button is pressed, SD02B will be reset to initial stage. Please DO NOT pressed this button during operation.

G – A push button to activate self test on SD02B. When it is pressed and hold, SD02B will start to drive stepper motor. If power supply is connected and voltage selector is selected correctly, LEDs at C will illuminate sequentially.

H – A small green LED to indicate status of power. If power is connected and voltage selector is selected correctly, this LED will light ON.

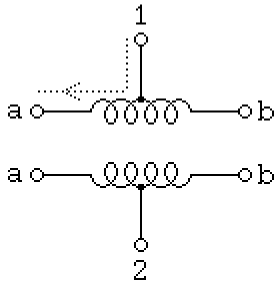
I – Voltage selector to select either Single Supply Mode or Dual Supply Mode for the power source.

J – **A** 2510 6 ways connector to power source and/or host. Please refer to chapter 5, Hardware installation for details.

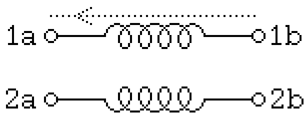
5. INSTALLATION (HARDWARE)

5.1 Connecting Driver to a Stepper Motor

Most unipolar stepper motor comes with 5 or 6 wires. As shown in the following left figure, there will be 2 wires (1 and 2) which act as the common terminal for coil a and coil b. If these 2 wires (1 and 2) are not used, it will become bipolar stepper motor as shown in figure on the right.



Unipolar stepper motor



Bipolar stepper motor

Before connection can be made, please determine the correct wiring for stepper motor. In general, 2-phase stepper motors can have 4, 5 or 6-wire leads, not including any (optional) encoder lines. The best solution is to obtain the pin out from the motor manufacturer. If user does not have access to the pin out, then the following procedure will help in determine wiring of the 2-phases stepper motor.

Some stepper motors have a motor case ground that can be tied to the ground of the system. It is usually a black wire, and it will add one additional wire to the overall count (4 coil wires + 1 casing ground = 5 wires total).

- If the stepper has <u>four coil wires</u>:

  a. Each of the two phases (X and Y) should have the same resistance when measured with a multi-meter. When measuring the resistance across one wire from each of the two phases (between X and Y), the resistance should be infinite because the circuit is open. Locate the two pairs of wires that represent the two phases, and both pairs of wires will have similar internal resistance.
  b. Connect each phase to SD02B and ignore the polarity (+/-), for now. User has a 50% chance of guessing right.
  c. Press Test button on SD02B to rotate the motor. It should rotate in CCW (counterclockwise) by looking at the motor shaft. If the motor rotates in the wrong direction, switch either phase X and !X or Y and !Y (effectively reversing directions).

- If the stepper has <u>six coil wires</u>, then each phase has a center tap wire:

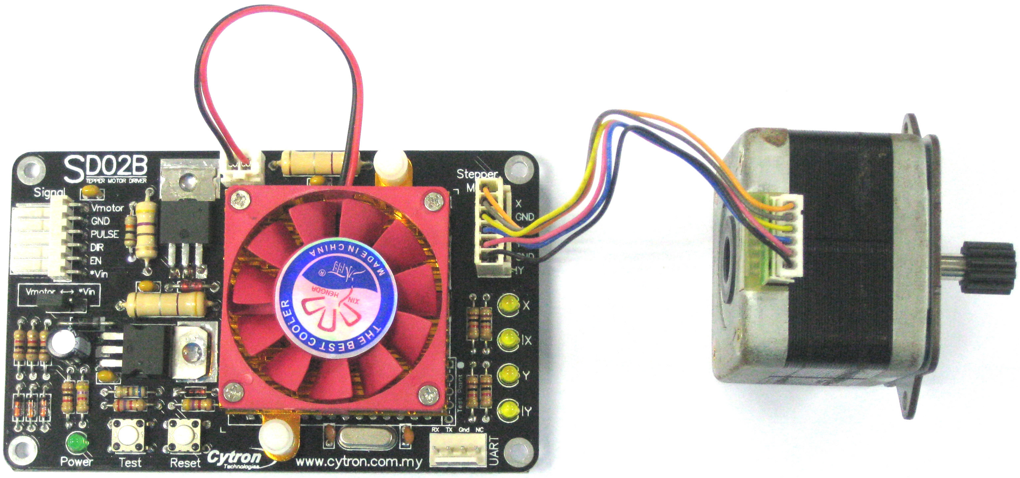
  a. The center tap wire should have half the internal resistance of the full phase. The easiest option is to use a multimeter to find the two pairs of wires that have the maximum resistance.

b. Connect each phase to the SD02B, and ignore the polarity (+/-) for now. User has a 50% chance of guessing right.

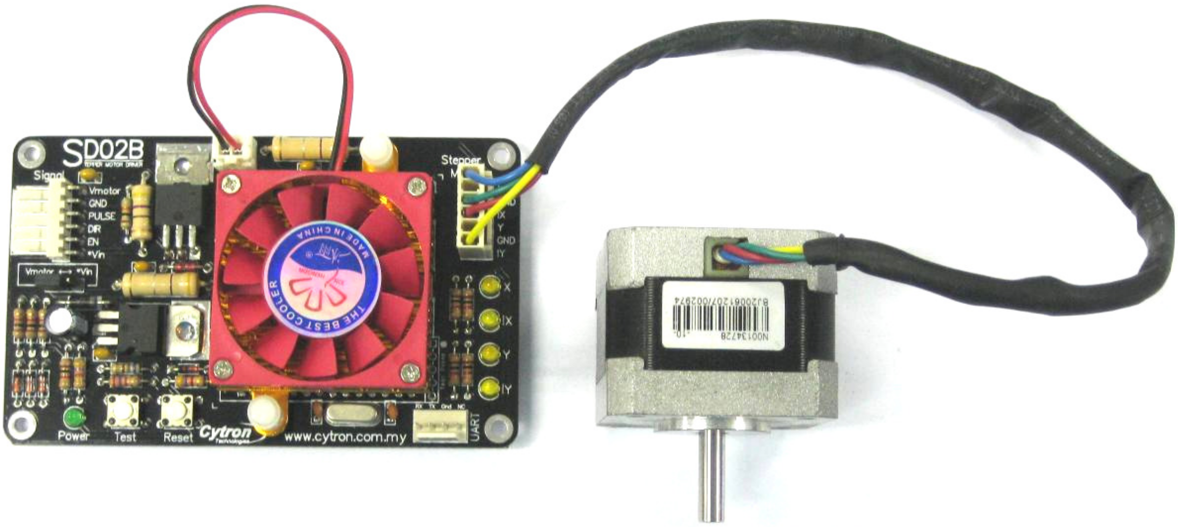
c. Press Test button on SD02B. The stepper motor should rotate in CCW (counterclockwise) by looking at the motor shaft. If it is in the opposite direction, switch either phase X and !X or Y and !Y (effectively reversing directions).

d. In case the stepper come with five coil wires, then each phase share the same common wire.

SD02B will only drive 2 coils which are 1a to 1b and 2a to 2b. Following pictures show typical connection from SD02B to Unipolar and Bipolar stepper motors.



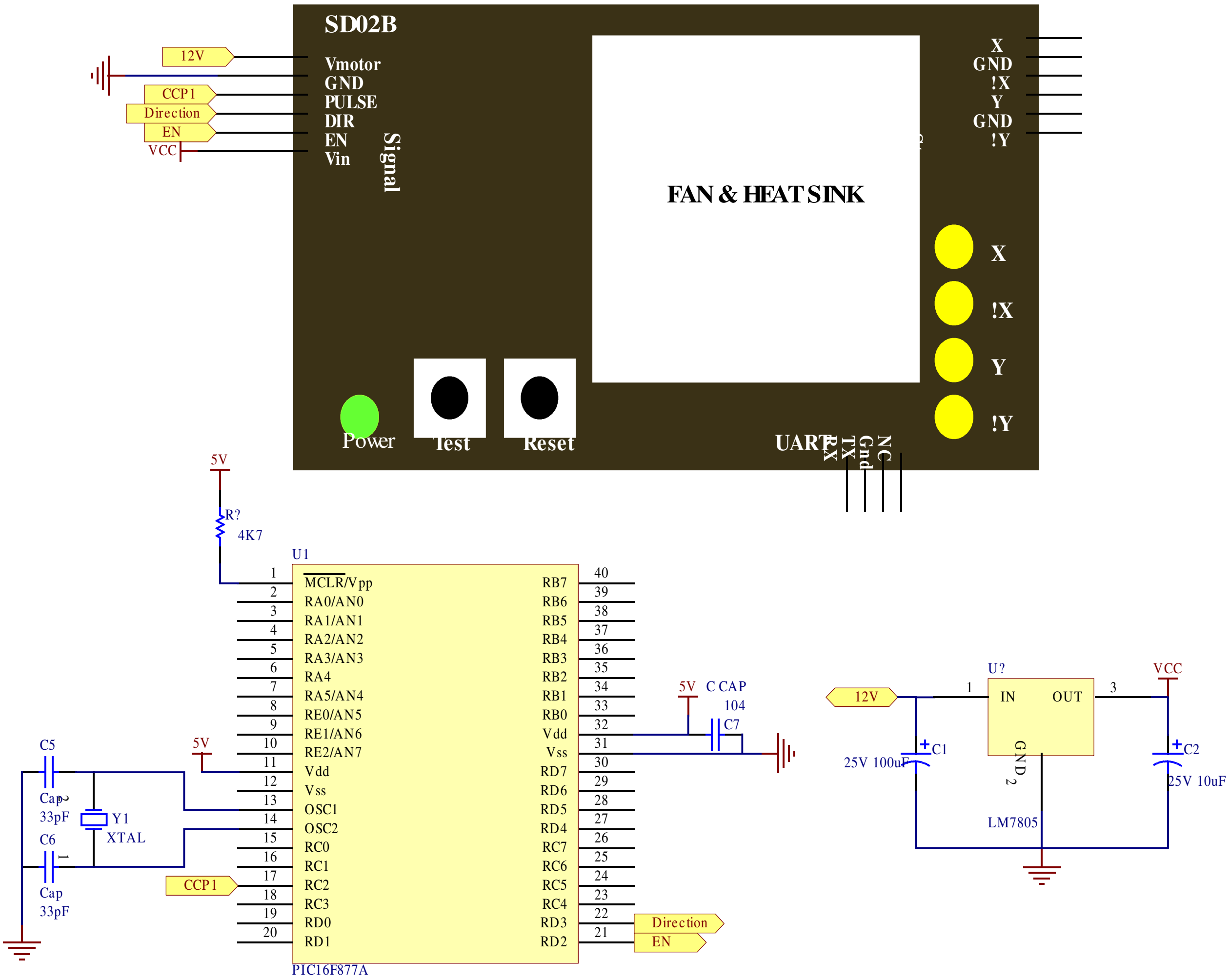
Unipolar Stepper motor (6 wires)



Bipolar Stepper motor (4 wires)

5.2 Connecting to Microcontroller using Signal Input Pins

Typical application would require a microcontroller to generate pulses and control the direction. Following figure shows an example of SD02B connected to microcontroller. The above example shows that the PULSE pin is being connected to RC2 of PIC16F877A.

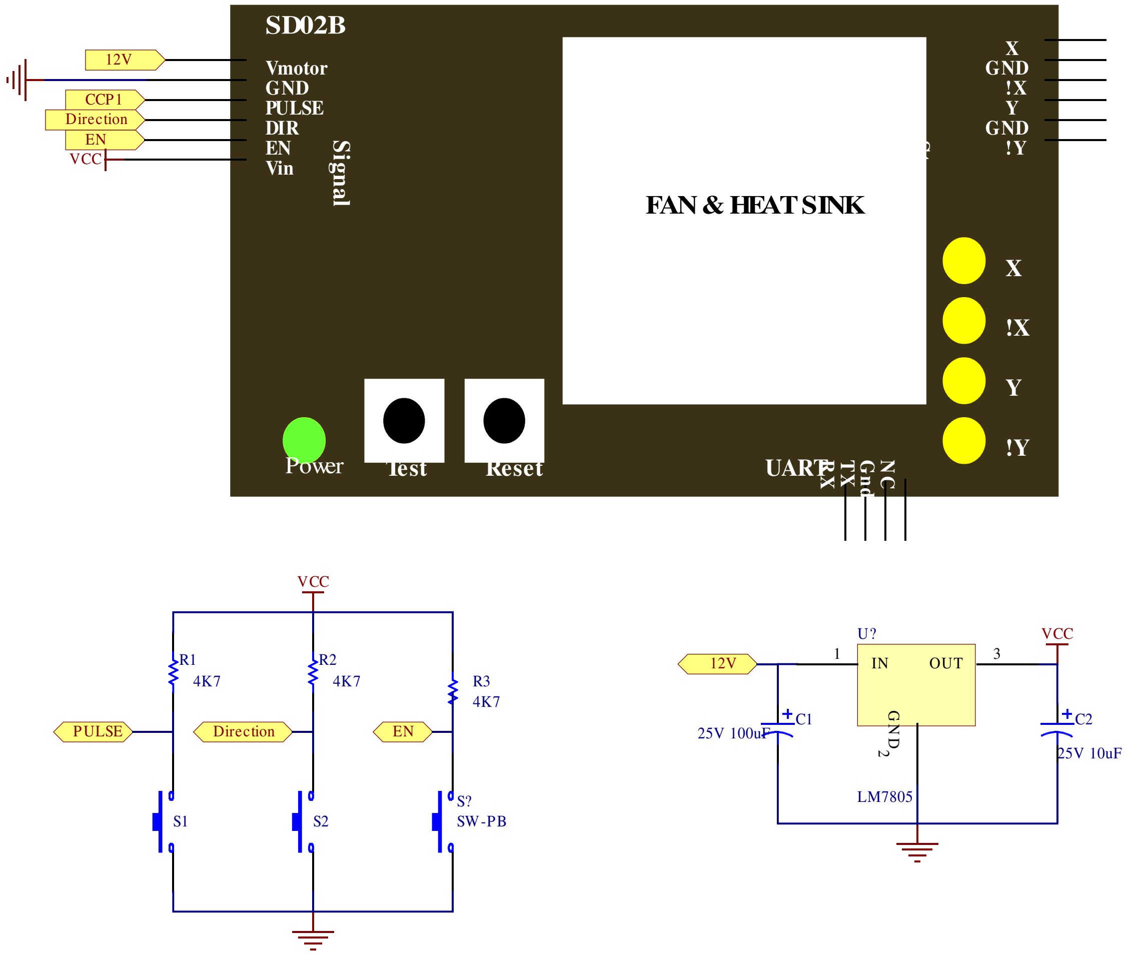


This pin has the function to generate pulses using Compare mode. Please refer to source code of PR7 from Cytron Technologies website for details of using Compare mode. Please note that if you are using PR7 to control SD02B, the voltage selector for SD02B should select Vmotor because the Vcc pin for PR7 only supplies 5V to *Vin. This is lower than the minimum voltage required for *Vin, which is 8V. Please refer to chapter 3 (Product Specification and Limitations) for details. The link to PR7 DIY Project is:
http://www.cytron.com.my/PR7.asp

Direction and EN pin can be connected to any digital I/O of PIC16F877A. However, both these pins must be configured as output from the controller.

5.3 Connecting to Switches (without microcontroller)

User may also use switches to generate pulses and control the direction of stepper motor. Here is an example of SD02B connected to switches.



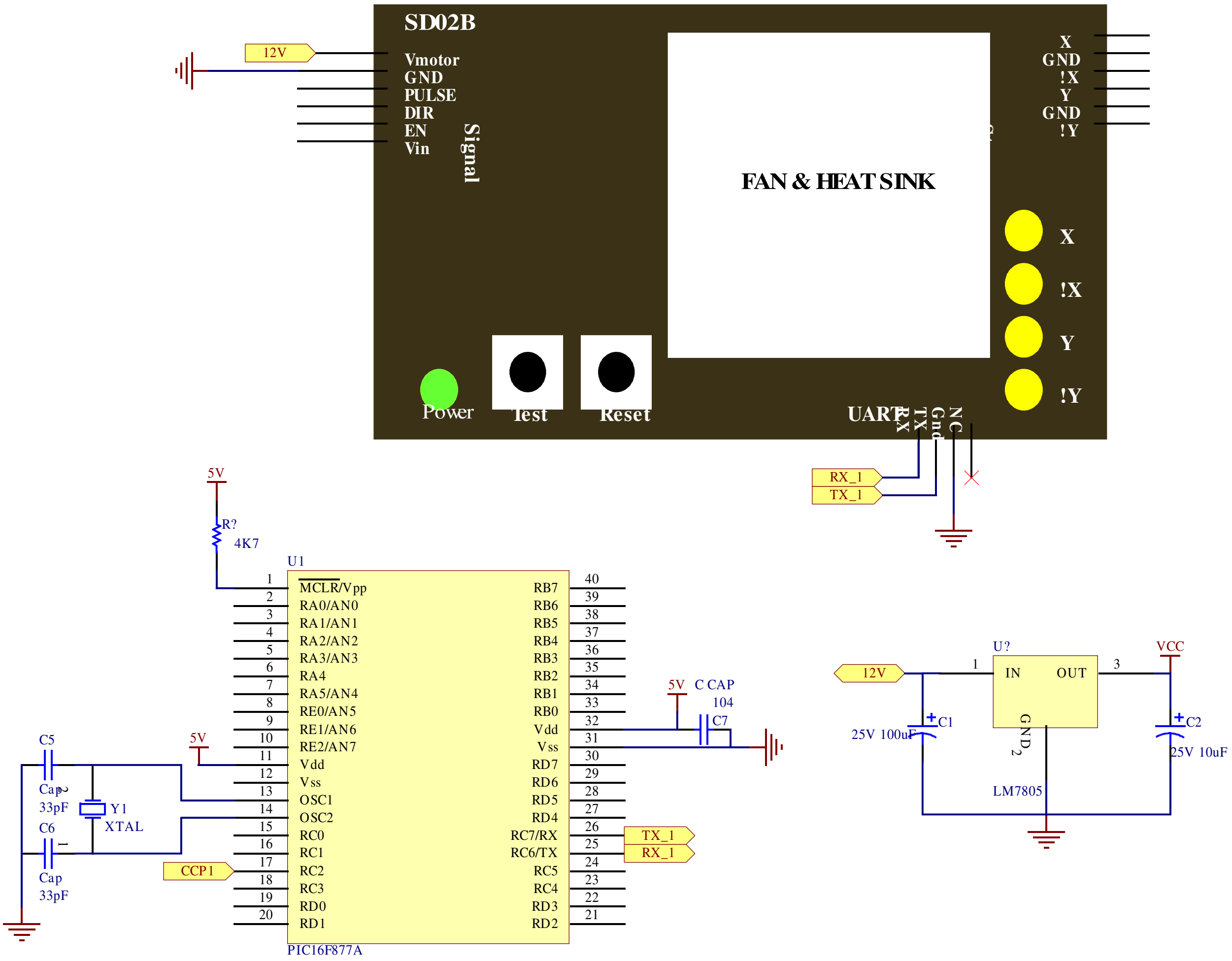
Above figure shows the sample connection using switches to control SD02B and drive a stepper motor. S1 act as pulses generator, S2 will control the direction of stepper motor while S3 will disable SD02B if pressed.

5.4 Connecting to Microcontroller through UART

A new feature for SD02B is the UART interface for easier communication between a host (user's circuit or PC) and SD02B. By using the UART control, user can on/off, run/brake and change motor rotation direction in a simple commands. Beside that user can request for encoder value, tracks the encoder value and brakes the motor at desired steps count, accelerate the motor from a specified initial speed to a final speed at desired acceleration rate. Here is an example connection of SD02B to Microchip PIC16F877A through UART.

5.4.1 Connecting SD02B to Microcontroller

For microcontroller to interface with SD02B, only 3 connections are needed. They are TTL UART (Universal Asynchronous Receiver and Transmitter) transmit (TX) and receive (RX) pins, together with the system ground (Gnd).



a. Sample schematic on above shows a PIC16F877A interface to SD02B. No restriction to what type or brand of microcontroller can be used, as long as it has UART peripheral.

b. User needs to connect the RX and TX pin to microcontroller. These two pins should be cross connected to microcontroller. In other words, RX on SD02B should be connected to microcontroller's transmitter pin (TX), while TX on SD02B should be connected to microcontroller's receiver pin (RX). For details connection, please refer to sample schematic above.

c. To begin, users have to choose either *Vin or Vmotor power supply at voltage selector. Then, connect Vmotor and/or *Vin and Gnd to power supply at a voltage according to the specification of the stepper motor used and make sure the voltage is within the voltage range of SD02B Vmotor and/or *Vin. Please refer to section 3, Product Specification and Limitations for details.

d. Power up the microcontroller board and SD02B. Once the power is supplied to SD02B, the 3mm green LED should light ON

e. The hardware interface for SD02B to microcontroller is ready! However, in order to control stepper motor from microcontroller, user is required to write a program for the microcontroller to send commands through its UART according to SD02B UART protocol.

f. User may refer to PIC18F4520 sample program provided to control two SD02B (thus, two stepper motors) through one hardware UART and one software UART using Microchip MPLAB C18 C Compiler Libraries. Since most midrange PICs only have one hardware UART, so one software UART is used for the second UART interface. This sample code shows how to generate an additional UART using software approach.

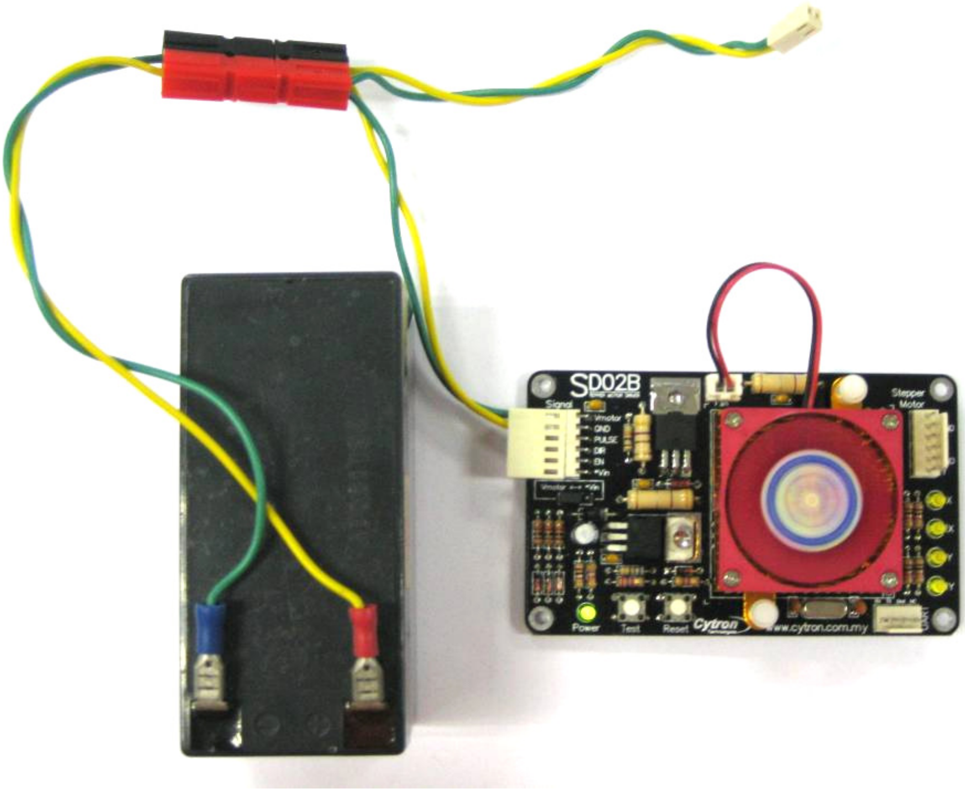
5.4.2 Connecting SD02B to Computer

Another concern in controlling stepper motor using a computer is the hard work needed to get started. However, with SD02B, interfacing with a computer is as easy as 1 2 3. Normally, user will need to develop a RS232 level shifter for communication to serial port. This generates extra work. Furthermore, laptop and computer nowadays have phase-out the serial port and replaced it with USB. With these reasons, an USB to UART converter, UC00A, have been developed by Cytron Technologies. Simply plug SD02B to UC00A and USB port of computer (PC or Laptop), install the driver (for the first time only) and it will create a virtual COM port in your computer and is ready to communicate with SD02B.

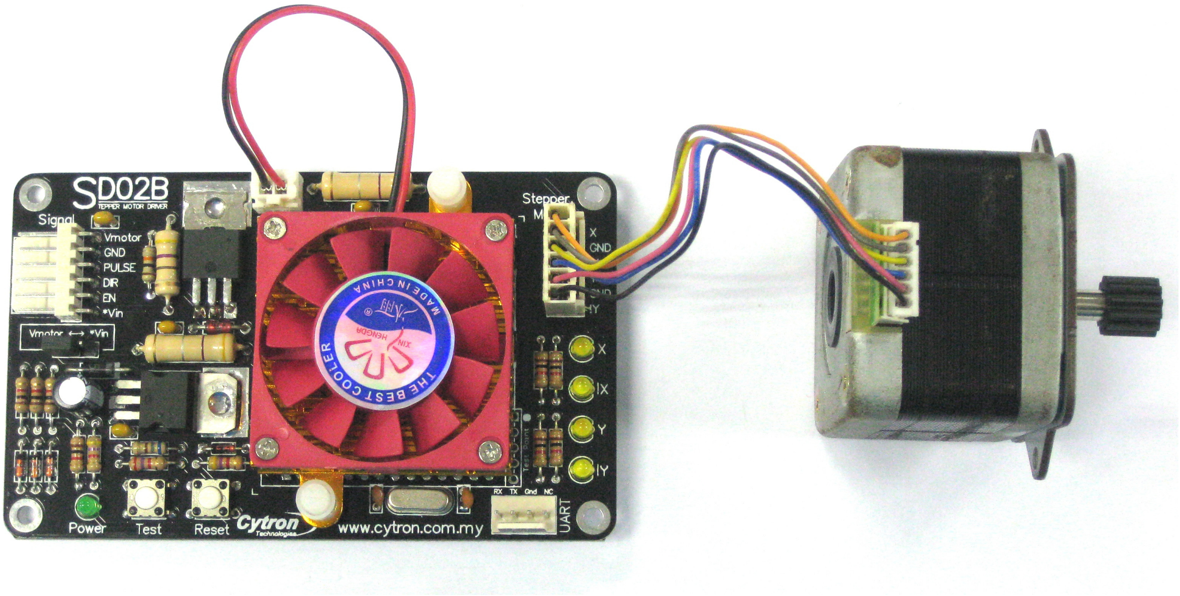
User may refer to UC00A User's Manual for hardware and software installation. So, no extra work is needed to control SD02B using a computer. And, of cause, you may connect more than one UC00A to your computer and control more than one SD02B at the same time.

A Graphical User's Interface (GUI) comes together with the source code is provided for user to implement stepper motor control using a computer.

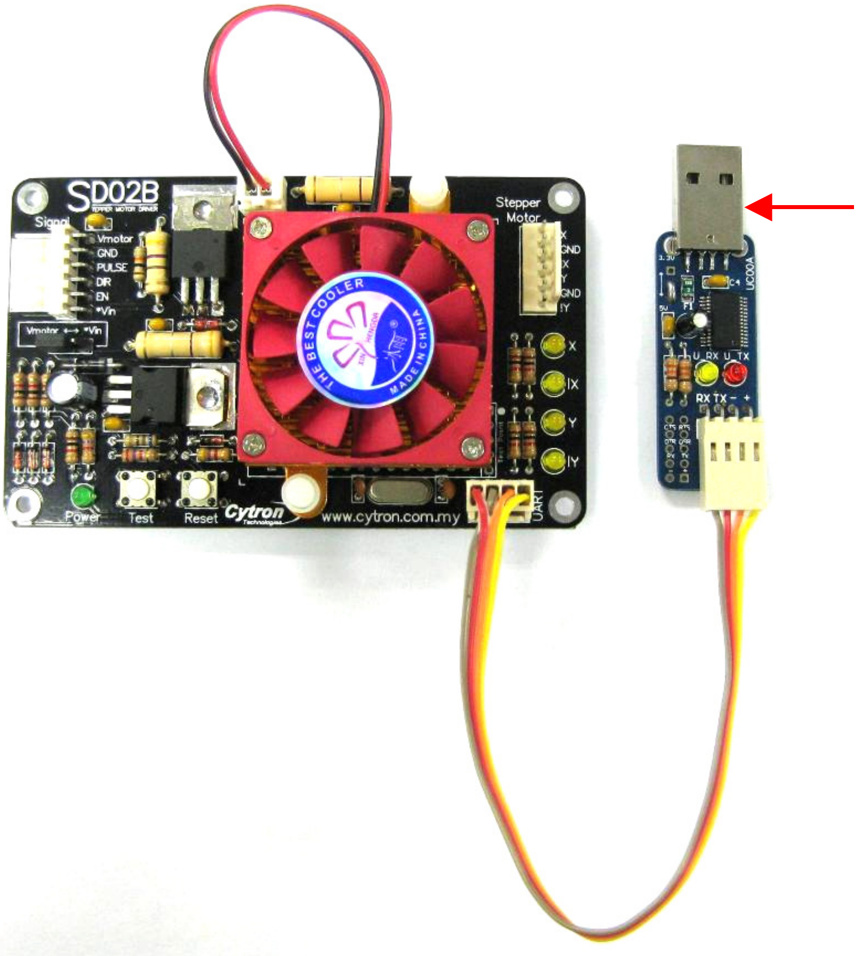
a. To begin, users have to choose either *Vin or Vmotor power supply at voltage selector. Then connect Vmotor and/or *Vin and Gnd to power source. Please refer to chapter 3, Product Specification and Limitations, for details. Figure below shows that SD02B is connected to a 12V Sealed Lead-Acid battery and Vmotor is choosen as the power source.



b. Figure below shows the setup of SD02B to a stepper motor.



c. To interface with a computer, simply connect UC00A 4-pin connector to SD02B and the USB end to PC as shown in the following figure. Please note that the TX and RX pins between SD02B and UC00A are cross-connected.



Connect to PC/Laptop

d. Please refer to documents named “USB Driver Installation Guide” included in UC00A package for driver installation.

e. After plugged in UC00A to computer and installed the driver, user is ready to test the functionality of SD02B using SD02B GUI.

5.5 Enable and Disable the driver

Current consumed by the stepper motor may raises to over 2A at certain conditions. It is depending on the stepper motor's ratings, stepping speed, micro-stepping mode and at which step the motor brakes. As stated in the Absolute Maximum Rating in chapter 3 (Product Specification and Limitations), the typical current per phase is rated at around 1A. Running at any value above 1A continuously may cause the driver to be overheated and automatically cut-off the circuit for protection. Hence it is more advisable to turn off the motor, instead of leaving it in braking condition (“stall state”).

For controlling SD02B using signal input pins, providing a low logic (0V) to EN pin allows user to “shut off” the driver and get into power saving mode. When stepper motor is idle, it can be in “stall state” (provide torque and draw current) or “off state” (free, do not draw current). By default, this pin is disabled.

For controlling SD02B through UART, the ‘OFF’ command will do the same job. Just send an ‘F’ to SD02B and it will turn off the motor and get into power saving mode. Please refer to section 7.3, SD02B UART Protocol for details.

6. INSTALLATION SOFTWARE

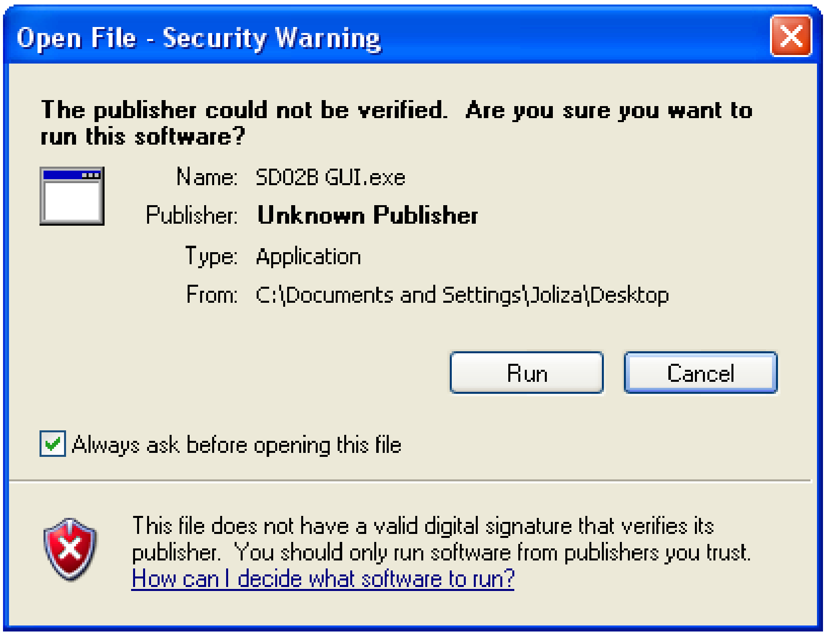
There are 2 options of sample software being provided for SD02B. The simplest way is to use the sample GUI program which can be run without the need to be installed on the computer, while another option is the sample program for PIC18F4520 to control two stepper motors through two SD02Bs.

6.1 Sample Program for Computer

To ease user in using SD02B, Cytron Technologies has developed a SD02B GUI using VB.net for user to control SD02B from computer through a COM Port. If your computer doesn't have a COM Port, you may use Cytron's UC00A (USB to UART Converter) for the interfacing between SD02B and your computer's USB port. The following steps will guide user to run SD02B GUI and use it.

1. Extract the zipped filed named "SD02B GUI.zip". This file can be downloaded from Cytron website (www.cytron.com.my).

2. Double click on "SD02B GUI.exe" to run the SD02B GUI. The window shown as below may prompt out, click Run.



Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

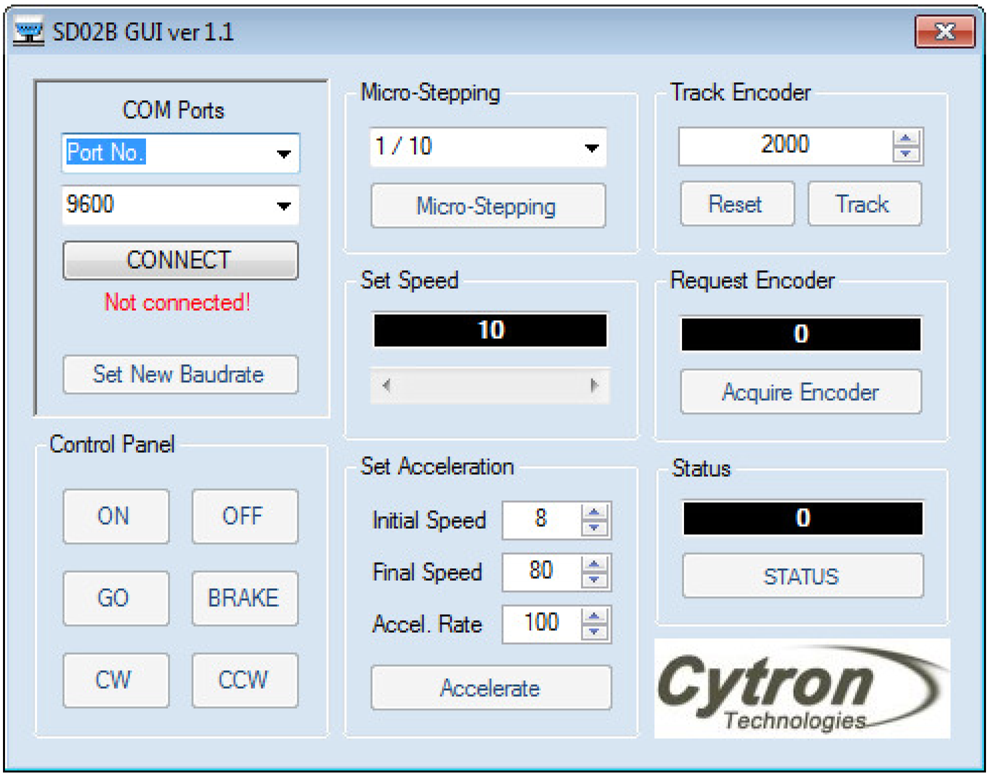
Name: SD02B GUI.exe
Publisher: **Unknown Publisher**
Type: Application
From: C:\Documents and Settings\Joliza\Desktop

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.
How can I decide what software to run?

3. If the program is unable to be executed, the latest .NET Framework may be needed to be installed. It is downloadable from Microsoft official website.

4. A window as shown in the figure below pops up.



SD02B GUI ver 1.1

COM Ports: Port No., 9600 — Not connected!

Micro-Stepping: 1 / 10

Track Encoder: 2000

Set Speed: 10

Request Encoder: 0

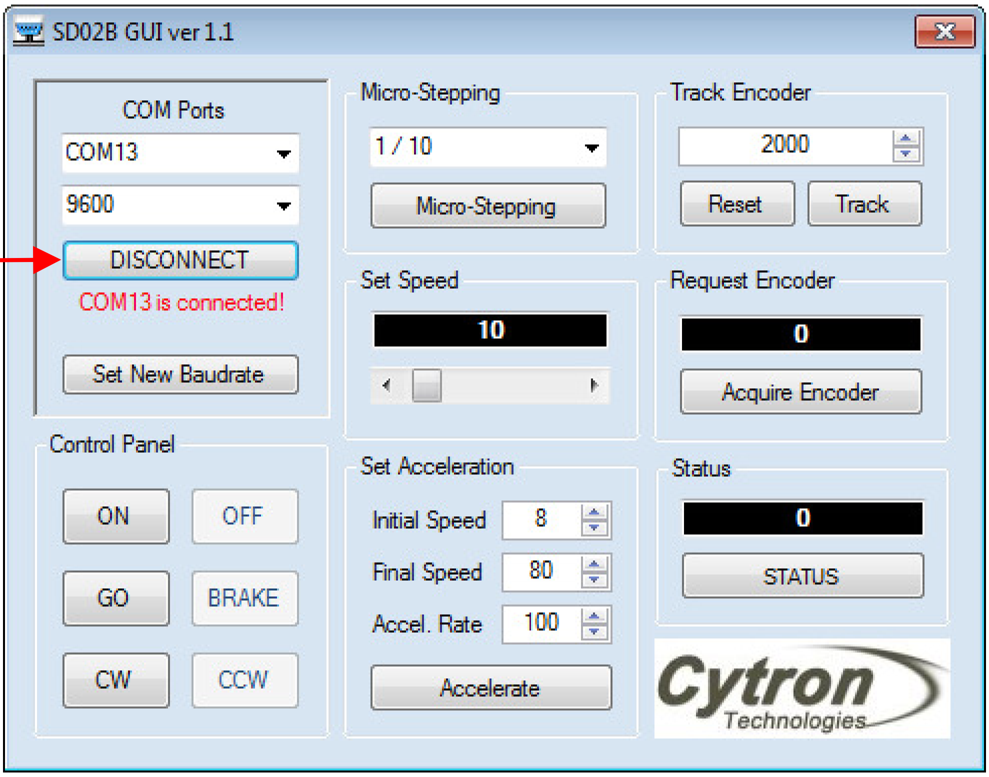
Set Acceleration: Initial Speed 8, Final Speed 80, Accel. Rate 100

Status: 0

5. Before any setup can be done, please ensure hardware installation of SD02B is being setup correctly and power to stepper motor is connected. If you are using UC00A, UC00A's driver must be installed.

6. After hardware installation, choose the correct COM Port. Normally, the extra virtual COM port will be the largest number port after driver installation. Select it. The default baud rate for SD02B is 9600.

7. Click the "Connect" button and SD02B GUI will show "COMxx is connected".



COM Ports: COM13, 9600

COM13 is connected!

Micro-Stepping: 1 / 10

Track Encoder: 2000

Set Speed: 10

Request Encoder: 0

Initial Speed: 8

Final Speed: 80

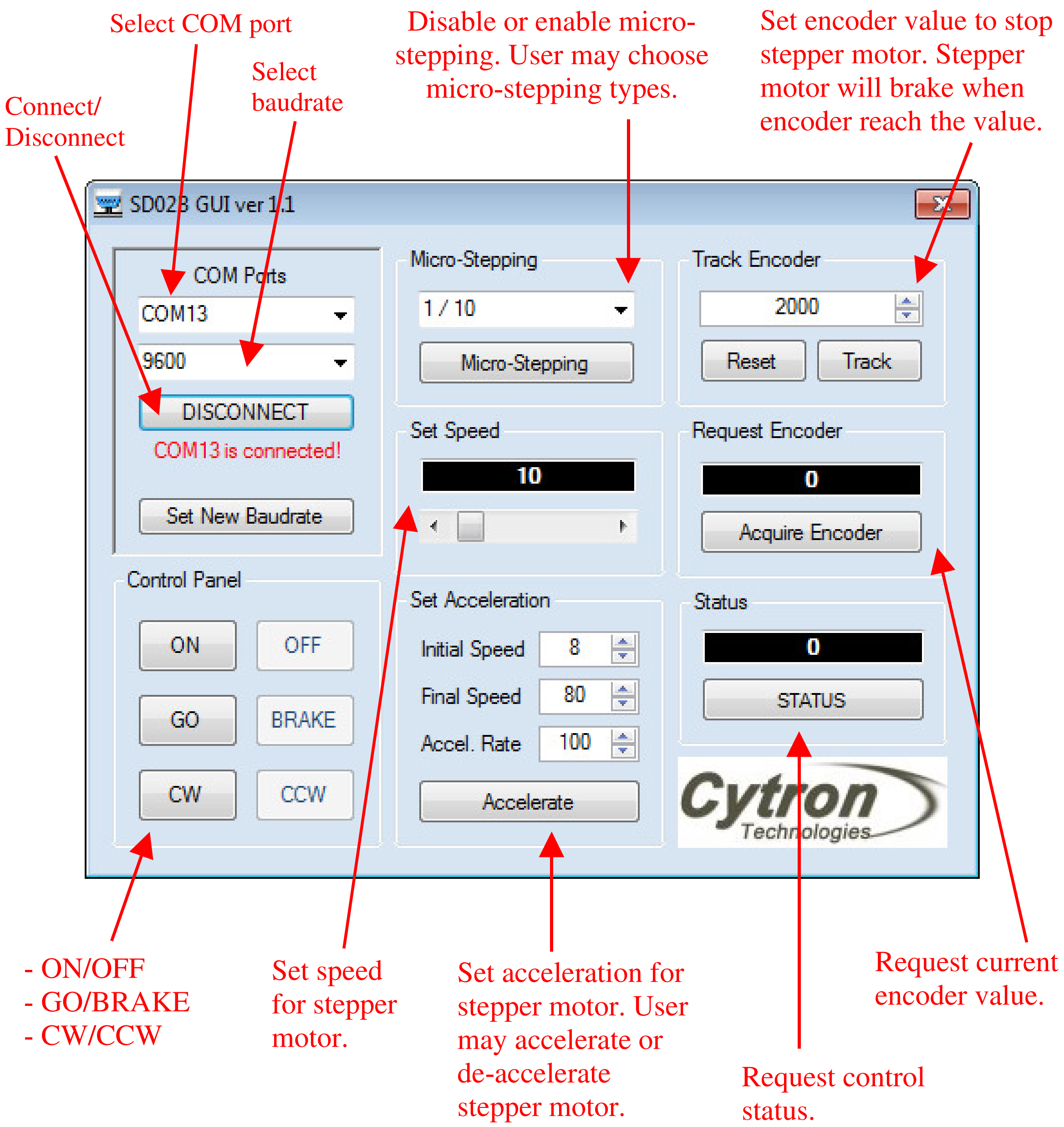
Accel. Rate: 100

Status: 0

CONNECT button has changed to DISCONNECT. Click to disconnect.

8. Now, SD02B GUI is ready to control the stepper motor using SD02B. User may on/off the motor, run/brake the motor, set the stepping speed, change direction, reset/track/request the encoder value (number of steps), accelerate the stepper motor from a specified initial speed to a final speed at desired acceleration rate and choose among 2/5/10 microstepping mode or non-microstepping mode using this GUI.

SD02B GUI ver1.1 Description



SD02B GUI ver 1.1

COM Ports: COM13

9600

COM13 is connected!

Micro-Stepping: 1 / 10

Track Encoder: 2000

Set Speed: 10

Request Encoder: 0

Set Acceleration

Initial Speed: 8

Final Speed: 80

Accel. Rate: 100

Status: 0

Select COM port

Select baudrate

Connect/ Disconnect

Disable or enable micro-stepping. User may choose micro-stepping types.

Set encoder value to stop stepper motor. Stepper motor will brake when encoder reach the value.

- ON/OFF
- GO/BRAKE
- CW/CCW

Set speed for stepper motor.

Set acceleration for stepper motor. User may accelerate or de-accelerate stepper motor.

Request control status.

Request current encoder value.

Note: The encoder value and the actual distance will depend on the diameter of the wheel. For example, if user use 1/10 micro-stepping with motor 1.8° per step, it needs 360° /0.18° = 2000 steps for 1 full 360° rotation. If the diameter of the wheel is 5cm, so the distance for stepper motor to travel in 1 full 360° rotation is 2п(5cm/2) = 15.71cm. Means, if we track encoder by sending 2000 to SD02B, it will stop the motor when it traveled 15.71cm. For wheel with different size and stepper motor with different resolution (°/step), the distance for the wheel to travel is different at 2000 encoder value. User need to calculate own encoder value by referring to the formulae given above.

6.2 Sample Program for PIC18F4520

Driving stepper motor is common necessity in most robotic project based on the embedded system which uses microcontroller. Considering this reason, Cytron Technologies has also developed the sample source code for PIC18F microcontroller using MPLAB C18 C Compiler. This sample program is developed for PIC18F4520. If user uses different PIC microcontroller, user may need to modify the sample program based on that particular microcontroller by referring to its datasheet released by Microchip Technology Inc.

The sample program will control two SD02Bs. Since this PIC only has one hardware UART, we have to generate another additional UART using software approach to control the second SD02B.

a. The sample program can be downloaded from Cytron website, (www.cytron.com.my), in SD02B product page.

b. Extract the file named "Dual UART sample program.zip".

c. There are 4 files available for the user; **18F sample program.mcp** (MPLAB project file)., **18F sample program.mcw** (MPLAB workspace file), **18F4520-dual uart sample.c** (source code in C language) and **18F sample program.hex** (HEX file). For immediate testing, user may load 18F sample program.hex to PIC18F4520 and plug into microcontroller board which was being completed in step c.

d. The sample program is based on SK40B startup kit. We will discuss about how to control two SD02Bs and two stepper motor using a SK40B through UART interface.

e. Ensure the hardware interface for microcontroller to SD02B is setup properly. Besides, the power for stepper motor is also needed. Please refer to section 5.4 for example connection of SD02B to microcontroller through UART interface.

f. Power indicator LED on SD02B will turn ON when power is supply to microcontroller board.

g. Please refer to **18F4520-dual uart sample.c** for details on method to send/receive command to SD02B.

h. User must install MPLAB and C18 Compiler if user wants to modify and compile this sample program. The free MPLAB and C18 Compiler for PIC18 MCU can be obtained from Microchip official website.

7. GETTING STARTED

Please refer to PR7, DIY project from Cytron website for details example of using SD02B. This DIY project also shows the method to write a program to control SD02B through the Signal Input pins on SD02B. The link to PR7 DIY project is:
http://www.cytron.com.my/PR7.asp

This section will show how to get started with SD02B and connect to UART communication. 1st example is using SD02B with computer and 2nd example is using SD02B with microcontroller.

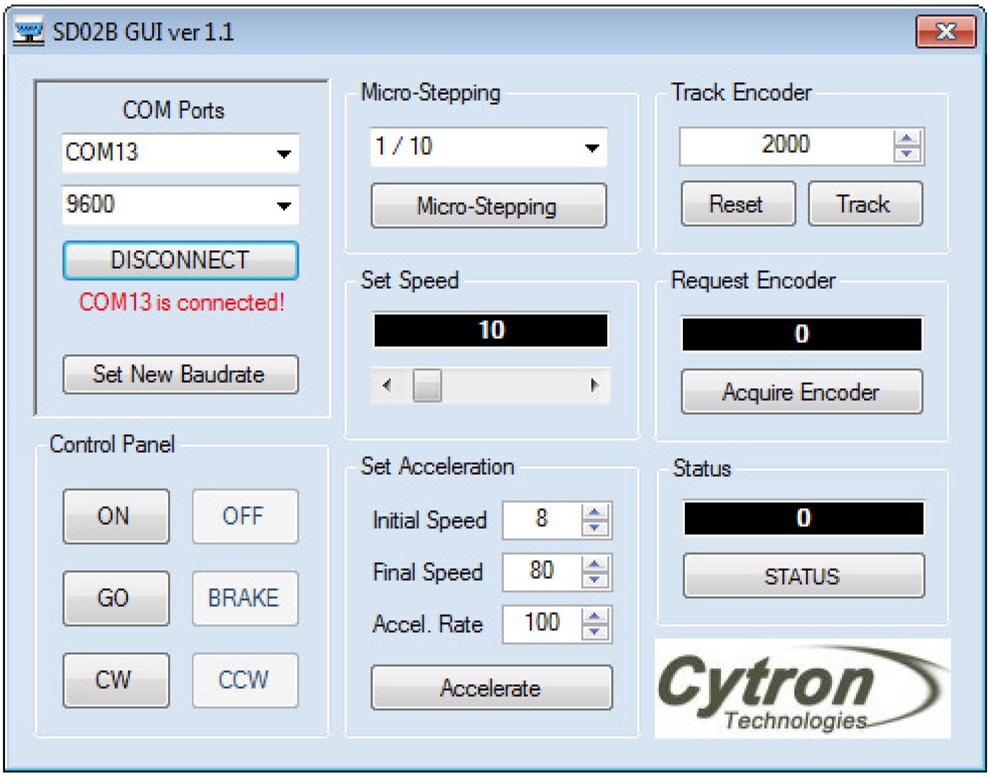
7.1 Using SD02B with computer

After installation of SD02B GUI and UC00A driver, user is ready to use SD02B with computer/laptop.

a. Setup SD02B according to steps in 5.4.2. Connect UC00A to SD02B, another end of UC00A to PC.

b. Power up SD02B. When power is connected, power indicator LED will turn ON.

c. Click SD02B GUI.exe to start control stepper motor. Before the SD02B GUI is able to control stepper motor on SD02B, connection must be made. Click on COM port, choose the last COM port, click connect.


SD02B GUI ver 1.1

COM Ports: COM13, 9600

COM13 is connected!

Micro-Stepping: 1 / 10

Track Encoder: 2000

Set Speed: 10

Request Encoder: 0

Set Acceleration: Initial Speed 8, Final Speed 80, Accel. Rate 100

Status: 0

d. Once the status is connected, user is free to control stepper motor using SD02B.

e. If you wish to communicate at higher baudrate, you may select the desired baudrate from the list and click on "Set New Baudrate" button. This will change the baudrate at both the GUI and SD02B. Press Reset button on SD02B will change the baudrate to its default value, which is 9600.

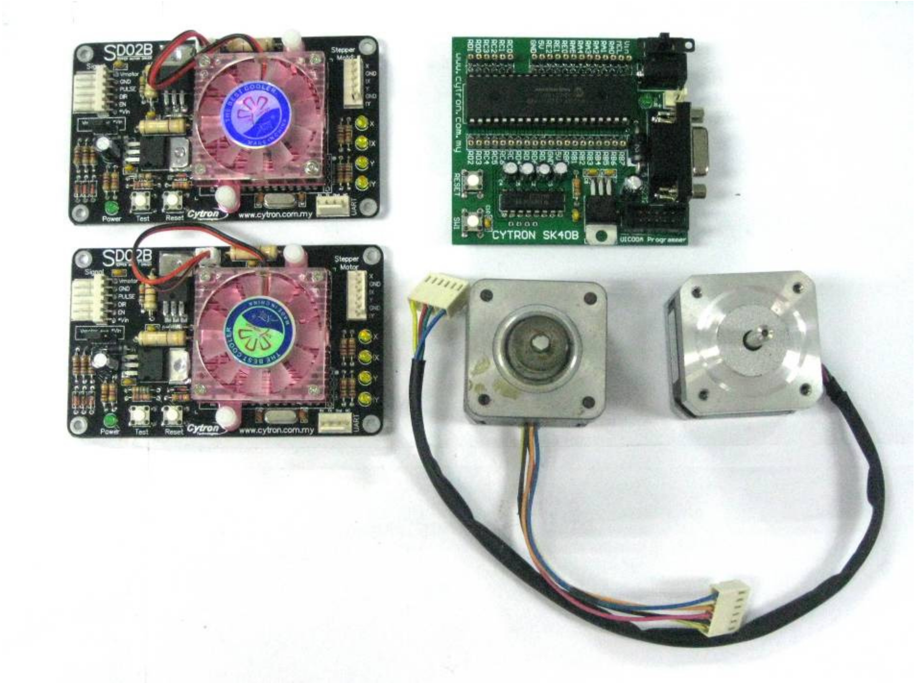
f. By using SD02B GUI, user may on/off the motor, run/brake the motor, set the stepping speed, change direction, reset/track/request the encoder value (number of steps), accelerate the stepper motor from a specified initial speed to a final speed at desired acceleration rate and choose among 2/5/10 microstepping mode or non-microstepping mode.

g. Since the GUI VB.net source code is provided, user can refer to the source code and modify it or write a new program to create own GUI for own application. The GUI is written with Microsoft Visual Basic 2008 Express Edition. Any version older than this may not be able to open the project file. If it is the case, user needs to install the newer version or recreate the VB.net project in older version.

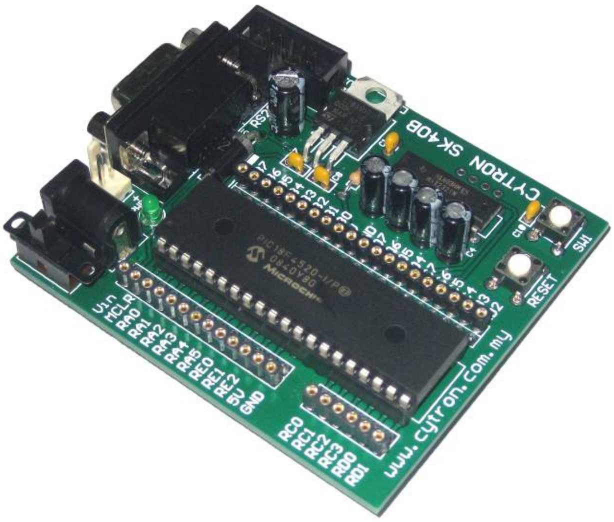
7.2 Using SD02B with PIC microcontroller

This section will show the example of using PIC18F4520 as host to control two SD02Bs through UART interface. The sample program is based on SK40B startup kit. This sample program is developed for PIC18F4520. If user uses different PIC microcontroller, user may need to modify the sample program based on that particular microcontroller by referring to its datasheet released by Microchip Technology Inc. This sample program will control 2 SD02B through 1 X hardware UART and 1 X software UART.

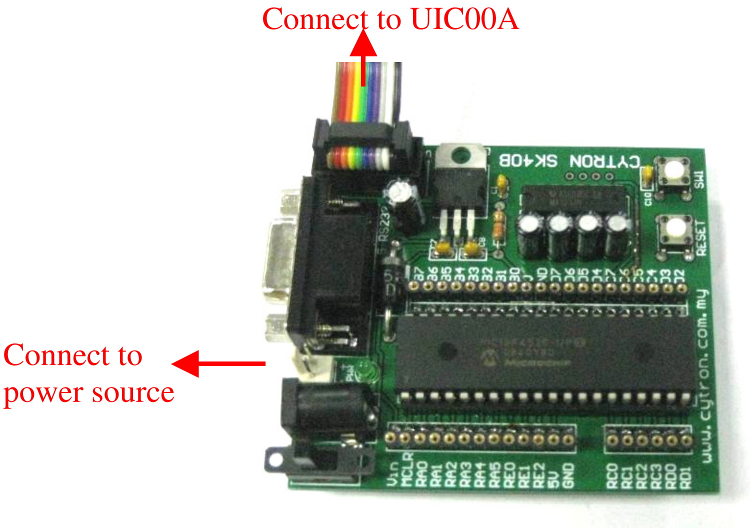
a. Since the sample program will control 2 SD02B, so two SD02Bs are used. Other devices are: 1 x SK40B, 2 x stepper motor.



b. Setup SK40B with PIC18F4520.

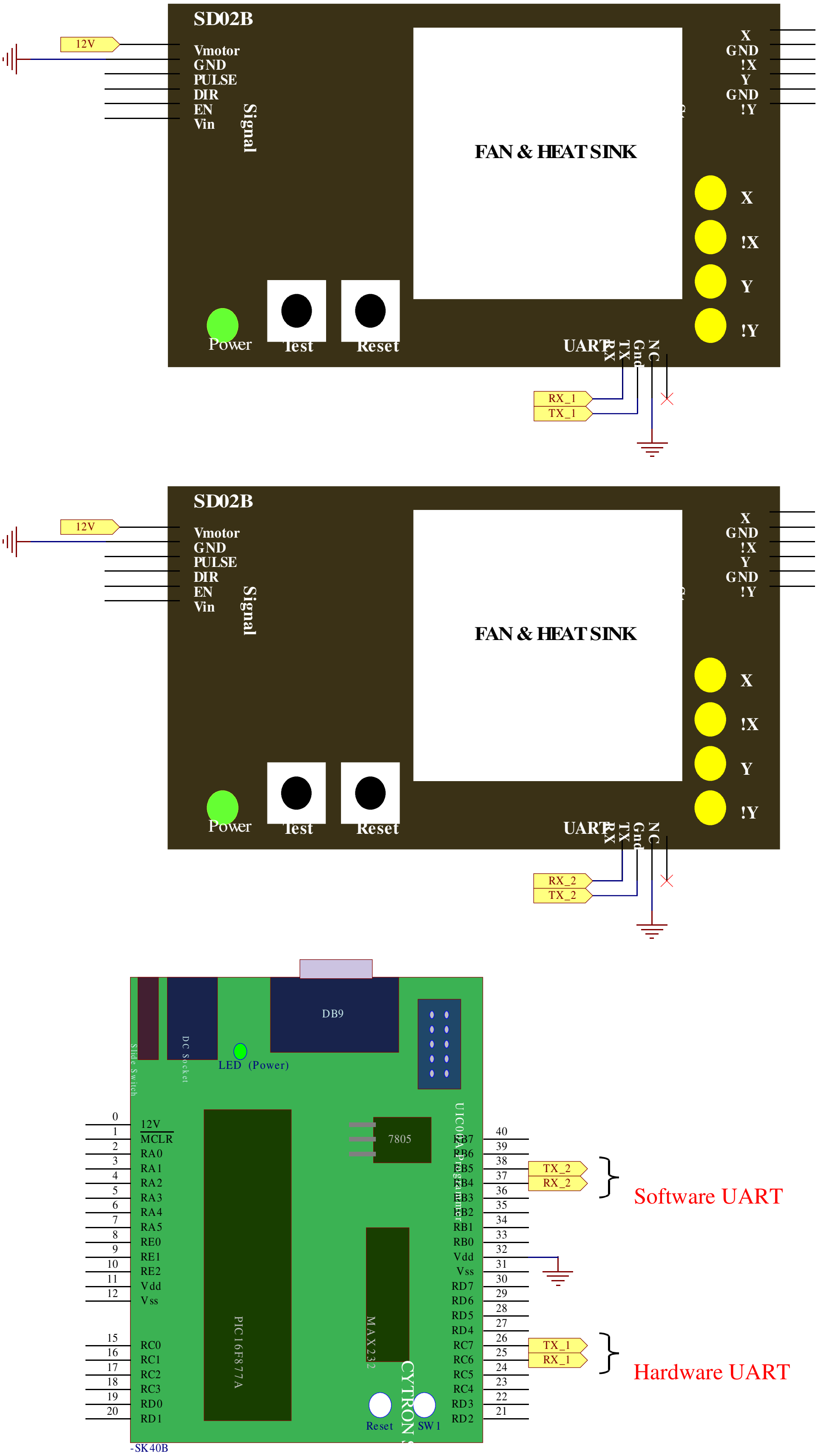


c. Load **18F sample program.hex** to PIC18F4520 using UIC00A. User may refer to UIC00A user's manual on how to load program into a PIC. User may use other programmer to load the program.



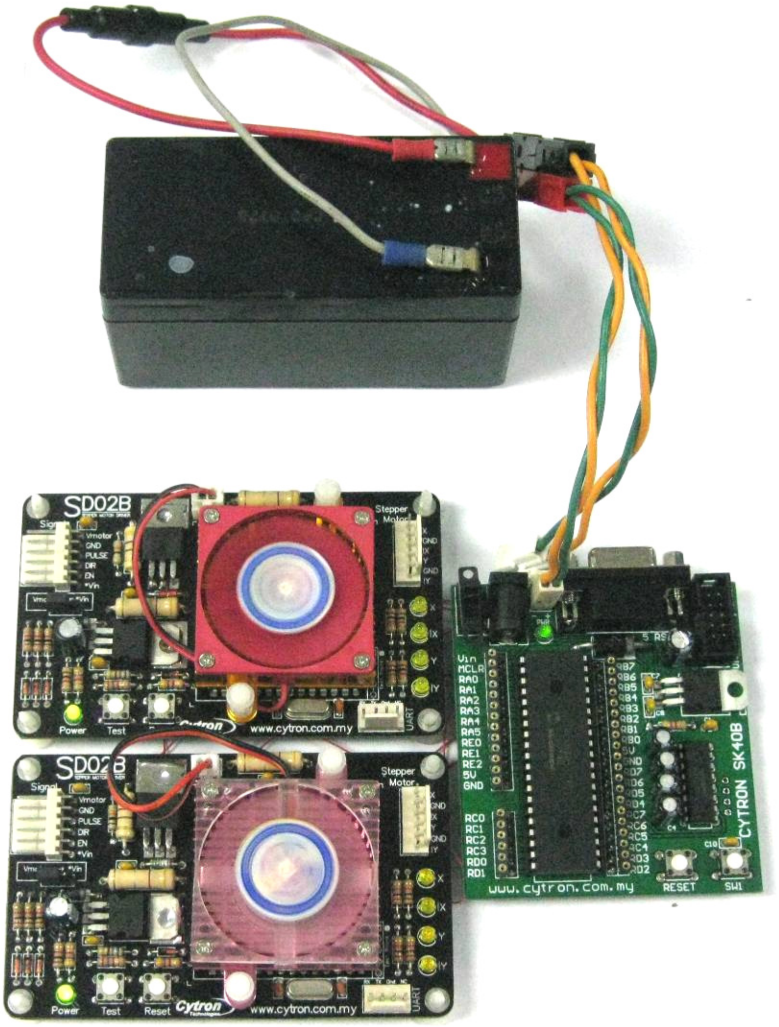
d. For communication between these two SD02Bs and SK40B, the following connections are required.

i. First SD02B **(RX)** to SK40B **(RC6 - TX)**
ii. First SD02B **(TX)** to SK40B **(RC7 - RX)**
iii. Second SD02B **(RX)** to SK40B **(RB4 - TX)**
iv. Second SD02B **(TX)** to SK40B **(RB5 - RX)**
v. Both SD02B **(GND)** to SK40B **(GND)**

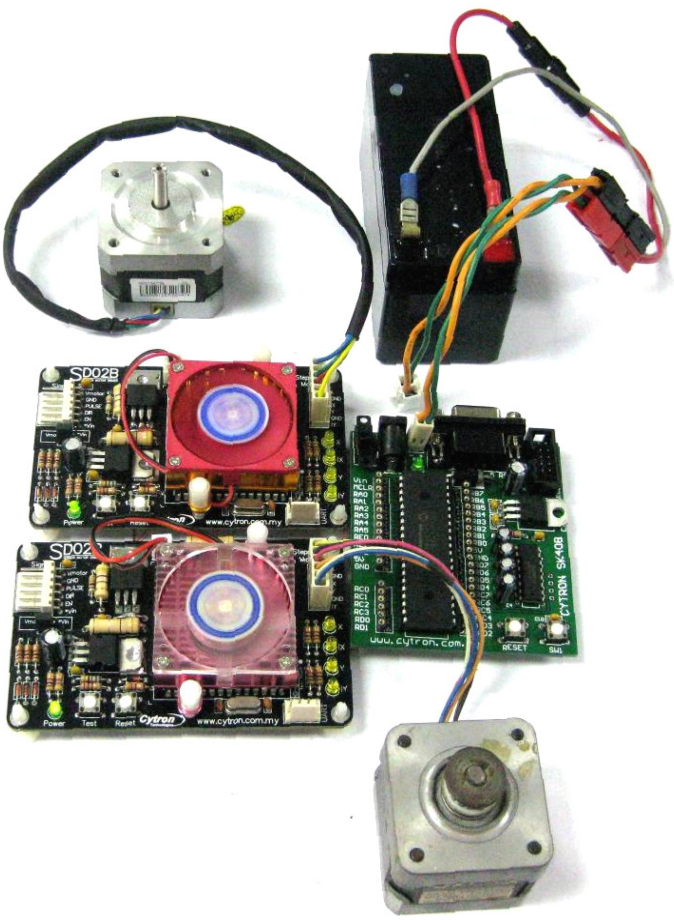


Sample connection to control 2 SD02B using software and hardware UART

e. Select Vmotor using the mini jumper at SD02B's voltage selector. To supply power to SD02B, user has to connect SD02B Vmotor to SK40B positive (+) power supply pin and GND to SK40B negative (-) power supply pin. In this case, we use a 12V Sealed Lead-Acid battery to supply SK40B. Once power is ON, power indicator LED on both SK40B and SD02B will turn ON.



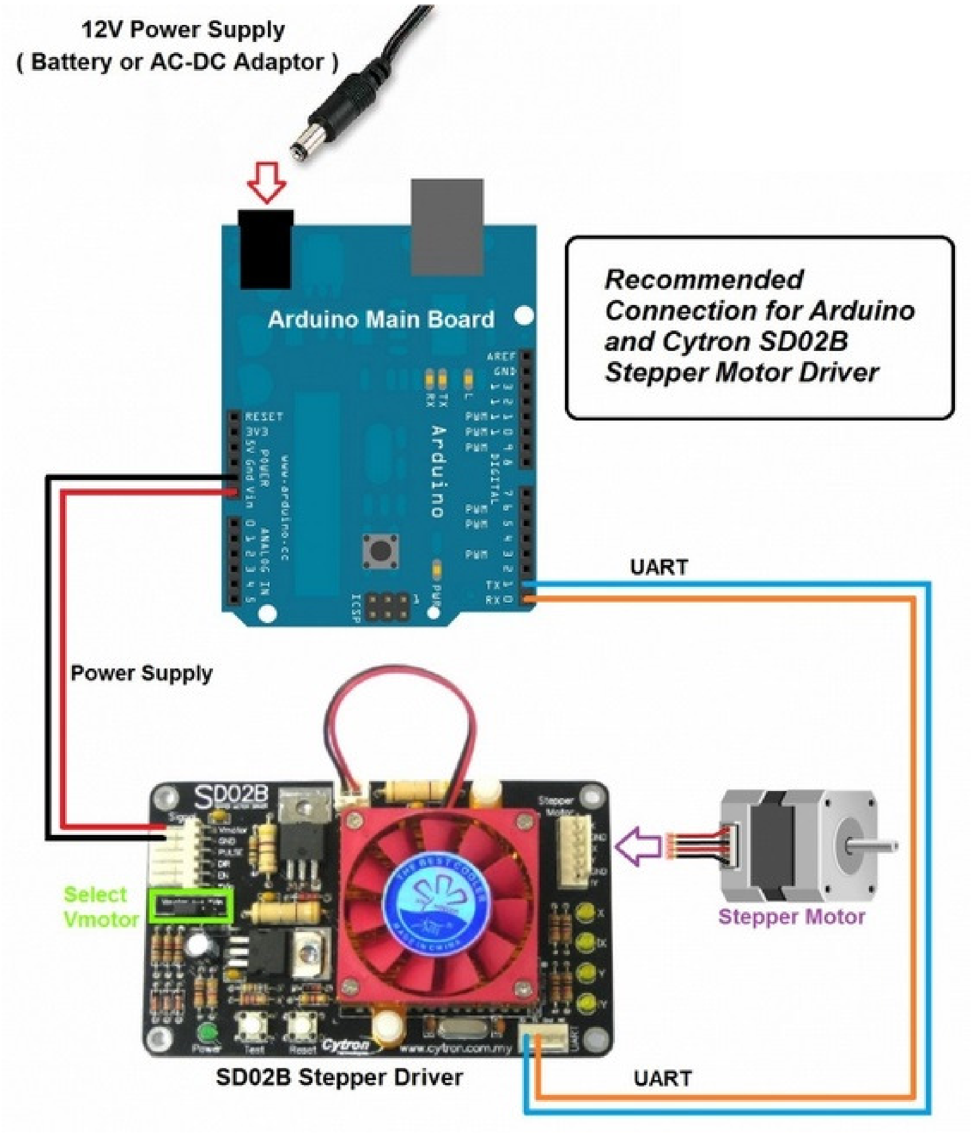
f. Next, connect the stepper motors to SD02Bs.



g. Now press the SW on SK40B to test the program. Every time the switch is pressed, SK40B sends some commands to either one SD02B or both SD02Bs. Please refer to sample program **18F4520-dual uart sample.c** file for the description.

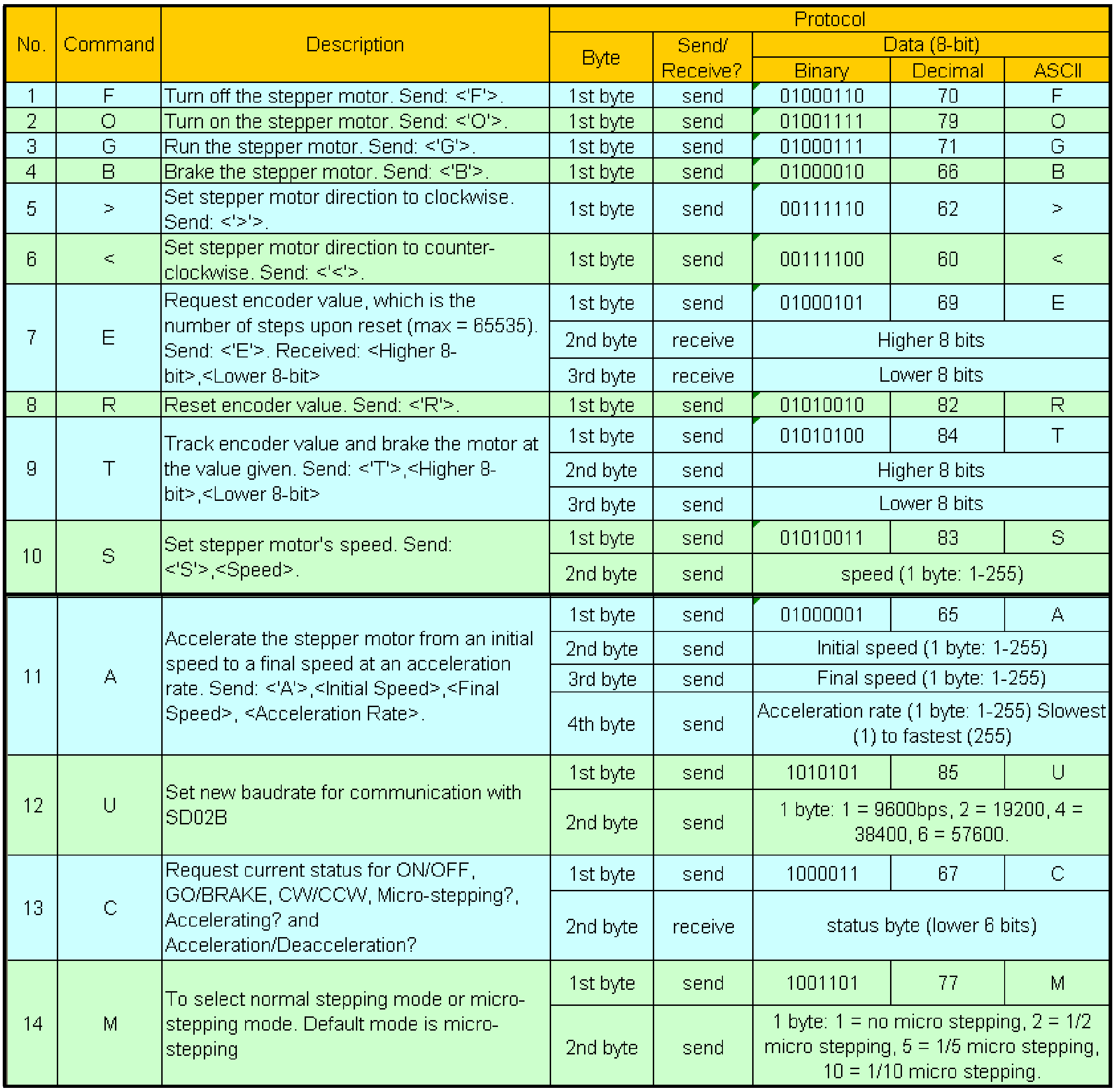
7.3 Using SD02B with Arduino

We have created a sample sketch for Arduino to communicate with SD02B through UART. The hardware connection diagram is shown in the figure below. The sample code can be downloaded from SD02B's product page. You can download Arduino Software (IDE) from http://www.arduino.cc


12V Power Supply
(Battery or AC-DC Adaptor)
Arduino Main Board
Recommended Connection for Arduino and Cytron SD02B Stepper Motor Driver
UART
Power Supply
Select Vmotor
Stepper Motor
SD02B Stepper Driver
UART

7.4 SD02B UART Protocol

If you are using the UART protocol to control SD02B, both computer and microcontroller are actually sending serial command to SD02B. This section explains the protocol for send/receive commands.



No.	Command	Description	Protocol				
			Byte	Send/ Receive?	Data (8-bit)		
					Binary	Decimal	ASCII
1	F	Turn off the stepper motor. Send: <'F'>.	1st byte	send	01000110	70	F
2	O	Turn on the stepper motor. Send: <'O'>.	1st byte	send	01001111	79	O
3	G	Run the stepper motor. Send: <'G'>.	1st byte	send	01000111	71	G
4	B	Brake the stepper motor. Send: <'B'>.	1st byte	send	01000010	66	B
5	>	Set stepper motor direction to clockwise. Send: <'>'>.	1st byte	send	00111110	62	>
6	<	Set stepper motor direction to counter-clockwise. Send: <'<'>.	1st byte	send	00111100	60	<
7	E	Request encoder value, which is the number of steps upon reset (max = 65535). Send: <'E'>. Received: <Higher 8-bit>,<Lower 8-bit>	1st byte	send	01000101	69	E
			2nd byte	receive	Higher 8 bits		
			3rd byte	receive	Lower 8 bits		
8	R	Reset encoder value. Send: <'R'>.	1st byte	send	01010010	82	R
9	T	Track encoder value and brake the motor at the value given. Send: <'T'>,<Higher 8-bit>,<Lower 8-bit>	1st byte	send	01010100	84	T
			2nd byte	send	Higher 8 bits		
			3rd byte	send	Lower 8 bits		
10	S	Set stepper motor's speed. Send: <'S'>,<Speed>.	1st byte	send	01010011	83	S
			2nd byte	send	speed (1 byte: 1-255)		
11	A	Accelerate the stepper motor from an initial speed to a final speed at an acceleration rate. Send: <'A'>,<Initial Speed>,<Final Speed>, <Acceleration Rate>.	1st byte	send	01000001	65	A
			2nd byte	send	Initial speed (1 byte: 1-255)		
			3rd byte	send	Final speed (1 byte: 1-255)		
			4th byte	send	Acceleration rate (1 byte: 1-255) Slowest (1) to fastest (255)		
12	U	Set new baudrate for communication with SD02B	1st byte	send	1010101	85	U
			2nd byte	send	1 byte: 1 = 9600bps, 2 = 19200, 4 = 38400, 6 = 57600.		
13	C	Request current status for ON/OFF, GO/BRAKE, CW/CCW, Micro-stepping?, Accelerating? and Acceleration/Deacceleration?	1st byte	send	1000011	67	C
			2nd byte	receive	status byte (lower 6 bits)		
14	M	To select normal stepping mode or micro-stepping mode. Default mode is micro-stepping	1st byte	send	1001101	77	M
			2nd byte	send	1 byte: 1 = no micro stepping, 2 = 1/2 micro stepping, 5 = 1/5 micro stepping, 10 = 1/10 micro stepping.		

On/Off, Run/Brake, CW/CCW and Reset Encoder Value

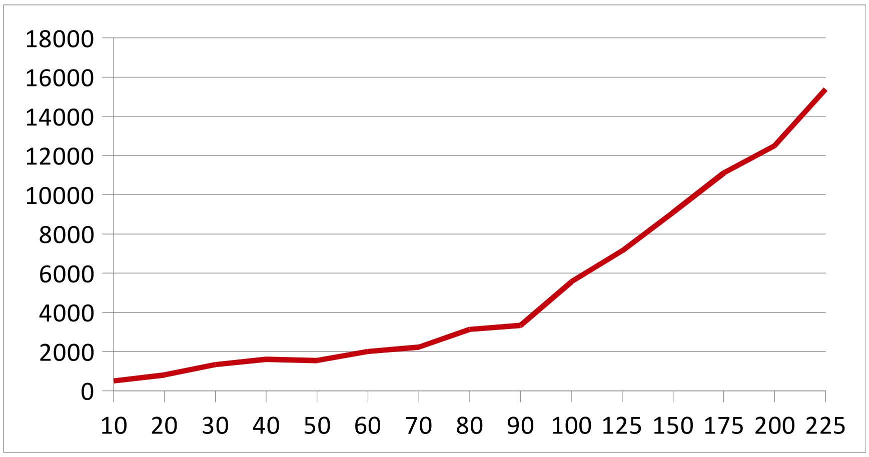
Host sends 1 byte command to SD02B to On/Off, Run/Brake, CW/CCW and reset encoder value.

Set Stepper Motor Speed

a. Host sends 'S' to SD02B to set stepper motor's speed.

b. After sent the first command, host need to send 8 bit speed value to SD02B. The speed value is from 1 to 255. If SD02B receives '0', the speed is assumed as 1. However, the speed for each stepper motor is different. It is depend on the resolution (°/step) of stepper motor.

c. Please note that the relationship between the speed value and actual stepping frequency is non-linear. (X-axis: Stepping Frequecy (Hz), Y-axis: Speed Value.)



Stepping Frequency (Hz) versus Speed Value

Accelerate stepper motor

a. Host needs to send a packet of data. Each packet consists of 4 bytes. The data need to be sent 1 byte by 1 byte.

   i. 1st byte: Send command 'A'
   ii. 2nd byte: Send initial speed
   iii. 3rd byte: Send final speed
   iv. 4th byte: Send acceleration rate

b. After sent 'A', host needs to send 1 byte of initial speed, followed by 1 byte of final speed and 1 byte of acceleration rate.

c. All initial speed, final speed and acceleration rate range from 1 to 255. If SD02B receive '0', the speed is assumed as 1.

d. Initial speed and final speed received by SD02B will be compared to determine which value is greater than the other one. If initial speed < final speed, SD02B will perform acceleration and if initial speed > final speed, SD02B will perform deacceleration.

e. For acceleration rate, 1 is the slowest rate and 255 is the fastest rate. This means when SD02B received 1 as acceleration rate, stepper motor will accelerate/deaccelerate very slow and if 255 is received as the acceleration rate, stepper motor will accelerate/deaccelerate very fast.

Request Encoder Value

a. Host sends command 'E' (1 byte) to SD02B to request encoder value. Encoder value is the number of steps upon reset.

b. After sent command to request encoder value, SD02B will send 1 byte for higher 8 bits current encoder value and another 1 byte for lower 8 bits current encoder value to the host. For example, host receives 11010110_2 for higher 8 bits current encoder value and 01110110_2 for lower current encoder value:

1	1	0	1	0	1	1	0	1st: Receive higher 8 bits
0	1	1	1	0	1	1	0	2nd: Receive lower 8 bits

c. To get higher 8 bits in 16 bits current encoder value, save the higher 8 bits into a 16 bits variable and rotate/shift 8 times.

x	x	x	x	x	x	x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	Save the first received 8 bits in 16 bits variable
x	x	x	x	x	x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	shift 1st time
x	x	x	x	x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	shift 2nd times
x	x	x	x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	x	shift 3rd times
x	x	x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	x	x	shift 4th times
x	x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	x	x	x	shift 5th times
x	x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	x	x	x	x	shift 6th times
x	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	x	x	x	x	x	shift 7th times
1	**1**	**0**	**1**	**0**	**1**	**1**	**0**	x	x	x	x	x	x	x	x	shift 8th times

Higher 8 bits

c. After rotated/shifted to the left for 8 times, user will get first received 8 bits at the 8 most significant bits (MSB).

1	1	0	1	0	1	1	0	x	x	x	x	x	x	x	x

x = don't care

d. AND the 16 bits variable with 0xFF00

$11010110XXXXXXXX_2$ AND 1111111100000000_2
$= 1101011000000000_2$

e. After received higher 8 bits encoder value, host will receive the lower 8 bits encoder value from SD02B. Save the lower 8 bits encoder value to another 16 bits variable.

x	x	x	x	x	x	x	x	0	1	1	1	0	1	1	0

f. AND the 16 bits variable with 0x00FF

$XXXXXXXX01110110_2$ AND 0000000011111111_2
$= 00000000001110110_2$

g. To get 16 bits encoder value, Exclusive-OR the two 16 bits variables and store it to a 16 bits variable.

Encoder value = 1101011000000000_2 XOR 0000000001110110_2
= 1101011001110110_2
= 54902 (16 bits encoder value)

Track encoder value

a. Host needs to send command 3 bytes of commands to SD02B to track encoder value. When the steps counted are equal to the value sent by host, SD02B will brake the stepper motor.

i. 1st byte: Send command 'T' to track encoder and brake motor
ii. 2nd byte: Send higher 8 bits encoder value
iii. 3rd byte: Send lower 8 bits encoder value

b. 2nd byte and 3rd byte sent to SD02B are 16 bits encoder value for encoder tracking (number of steps to go before SD02B brakes the stepper motor). For example, if user wants stepper motor to brake when encoder value equals to 54902. Host needs to send this value (1101011001110110_2) to SD02B. Higher 8 bits is to be sent before the lower 8 bits encoder value.

Higher 8 bits encoder value Lower 8 bits encoder value

1	1	0	1	0	1	1	0	0	1	1	1	0	1	1	0

16 bits encoder value

c. To get higher 8 bits, rotate/shift the 16 bits value 8 times to the right.

1	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	1	1	0	1	1	0	
0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	1	1	0	1	1	shift 1st times
1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	1	1	0	1	shift 2nd times
1	1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	1	1	0	shift 3rd times
0	1	1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	1	1	shift 4th times
1	0	1	1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	1	shift 5th times
1	1	0	1	1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	1	shift 6th times
1	1	1	0	1	1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	0	shift 7th times
0	1	1	1	0	1	1	0	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	shift 8th times

d. After rotated/shifted 8 times, AND the value with 0x00FF and save it into an 8 bits variable.

0111011011010110_2 AND 0000000011111111_2
= 0000000011010110_2
= 11010110 (higher 8 bits)

d. To get lower 8 bits, AND the same 16 bits encoder value with 0x00FF and save it into another 8 bits variable.

1101011001110110_2 AND 0000000011111111_2
= 0000000001110110_2
= 01110110_2 (lower 8 bits)

e. Send 'T' to SD02B first. Then send the higher 8 bits, followed by the lower 8 bits.

Set UART Baudrate

a. The default baudrate for SD02B is 9600. However user can define new baudrate for UART communication after the host and SD02B is connected at 9600. Host needs to send 'U' to set new baudrate for communication with SD02B.

b. The standard baudrates for SD02B are 9600bps, 19200bps, 38400bps and 57600bps. After sent the first command, host has to send the 8 bit value to SD02B according to the explanation below. The values are the multipliers for 9600.

1 x 9600bps = 9600bps
2 x 9600bps = 19200bps
4 x 9600bps = 38400bps
6 x 9600bps = 57600bps

Send this 8 bit value to SD02B to set the new desired baudrate.

c. However, it is possible for user to operate SD02B at a baudrate other than the standard baudrates given, but the success of the attempt is not guaranteed. For example, you can try to send '3' to SD02B to get the baudrate of 9600 x 3 = 28800bps.

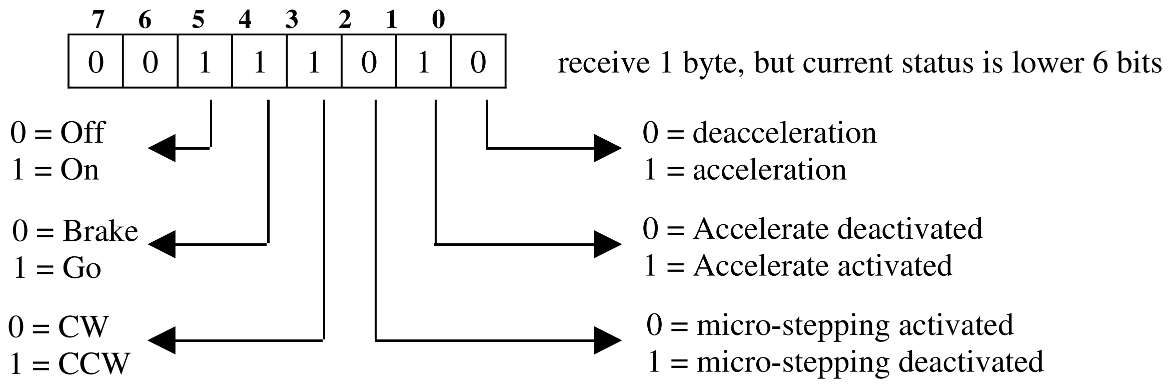
Request current status

a. Host sends 'C' to request current status for On/Off, Go/Brake, CW/CCW, micro-stepping and acceleration.

b. Host will receive 1 byte current status. From the 8 bit (1byte) received, only the lower 6 bits is used to indicate the current status. The status bits and its definition are shown below.

7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	1

send command 'C' to request current status



7	6	5	4	3	2	1	0
0	0	1	1	1	0	1	0

receive 1 byte, but current status is lower 6 bits

0 = Off
1 = On

0 = Brake
1 = Go

0 = CW
1 = CCW

0 = deacceleration
1 = acceleration

0 = Accelerate deactivated
1 = Accelerate activated

0 = micro-stepping activated
1 = micro-stepping deactivated

1/10, 1/5 or 1/2 Micro-stepping and Non-micro-stepping

a. Host needs to send 'M' and followed by 1 byte value which indicates the stepping mode to SD02B. Any values other than the given ones will be ignored by SD02B.

1	= no micro-stepping
2	= 1/2 micro-stepping
5	= 1/5 micro-stepping
10	= 1/10 micro-stepping

Send this 8 bit value to SD02B to choose the stepping mode.

b. 1/10 micro-stepping means every step of the stepper motor is further divided into 10 micro-steps. For a stepper motor with 1.8°/step, 1 micro-step will rotate the motor shaft 1.8° / 10 = 0.18° if 1/10 micro-stepping mode is chosen. It is goes the same for 1/5 and 1/2 micro-stepping. Non-micro-stepping mode will step the motor shaft at its original value which is 1.8°.

8. WARRANTY

- Product warranty is valid for 6 months.
- Warranty only applies to manufacturing defect.
- Damage caused by miss-use is not covered under warranty.
- Warranty does not cover freight cost for both ways.

Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: *+607-521 3178*
Fax: *+607-521 1861*

URL: www.cytron.com.my
Email: support@cytron.com.my
sales@cytron.com.my