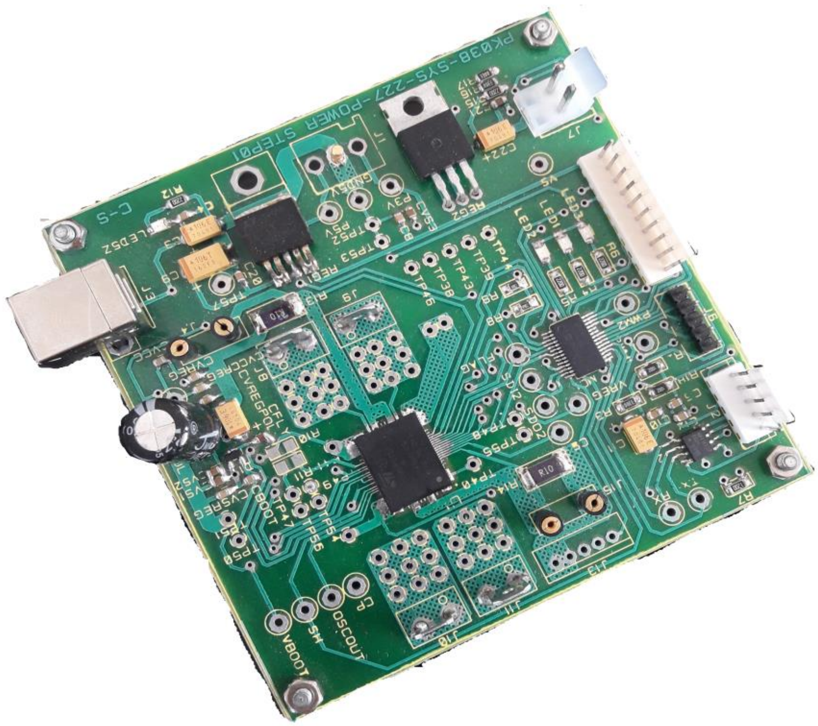
Stepper Motor Driver



summary

This is a stepper motor driver circuit based on the POWERSTEP01 IC and the PIC18F27J53 microcontroller in terms of hardware architecture and messaging frameworks, but it does not replace the reference to the technical datasheet of the IC.

1. Specifications

- Working voltage: 24V. It can be adjusted to work within the range of 7.5V-85V (as required).
- Maximum current: 2A .
- Working modes: The circuit supports two working modes (voltage mode and current mode).
- Driving modes: The circuit supports two driving modes (command mode and driving mode via external square signal).
- Maximum accuracy: 1/128 microstep.
- Maximum speed: 15625 steps/sec.
- Transmission protocol: RS422 .
- Data transmission frequency: 38400 bit/s.
- Saving protections: Over temperature and over current protection.
- Driving programs: flexible computer program.
- Additional features: Stall Detection and the ability to know the circuit status.
- Circuit dimensions: 9.3x9.2 cm.

2. Applications

- Bipolar stepper motor.

3. Description

This circuit is a bipolar stepper motor drive circuit based on the POWERSTEP01 integrated circuit which is the only integrated circuit in the market that provides the user with two modes of stepper motor drive: voltage control and current control in a wide range of voltages (up to 85V). The protocol used for communication with the circuit is RS422. The circuit responds to the investor's commands through a set of frames sent through the UART port, where the investor can change the value of the integrated circuit registers through these commands, allowing him to adjust the parameters (speed, acceleration, deceleration, accuracy, etc.) to suit him. In addition the circuit alerts the user when any error occurs by sending a message, where the type of this error can be determined by reading the circuit status (Get Status). There are a large number of parameters to control, so the circuit allows the investor to save the final values of the registers to suit his application, in addition to the ability to return to the initial state (factory setting) through the resetcommand. One of the most important features of this circuit is that it allows the investor to drive the motor at high speeds (up to 15625 steps/second) in addition to the ability to drive it with very high precision (up to 1/128 microstep). We provide the investor with an assistant demo program that allows him to adjust and control the circuit and identify the frames that correspond to each command.

4. Hardware structure

Figure (1) shows the circuit's outputs and inputs.

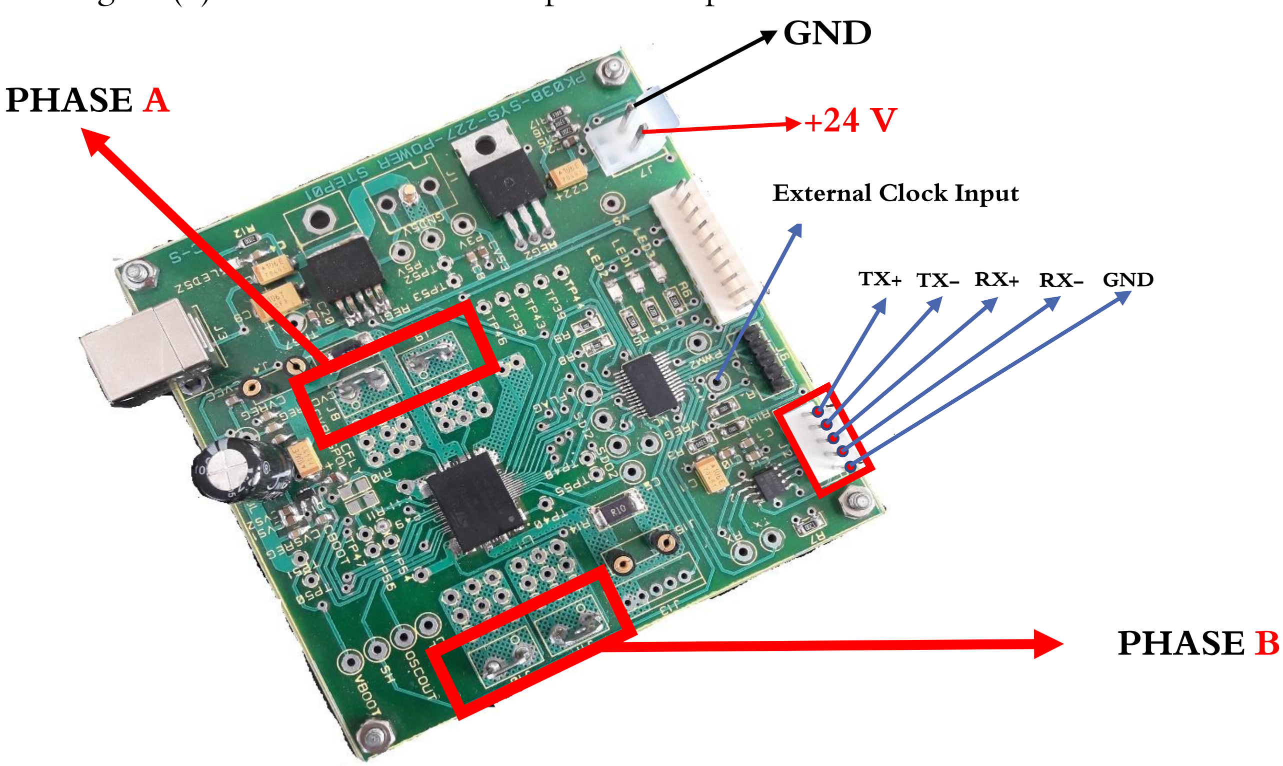


Figure (1): Outputs and inputs of the control circuit.

Note: The transmission outputs and inputs are labeled for the circuit and not for the RS422 to USB circuit .

warning : Please be careful when feeding the circuit as it is not protected from reverse polarity.

5. POWERSTEP01 IC REGISTERS

Table (1) shows the registers of the integrated circuit used.

Number of bits	Record work	Record Name	Record Title
20	Current Speed	SPEED	04H
12	Acceleration	ACC	05H
12	Deceleration	DEC	06H
10	Maximum Speed	MAX_SPEED	07H
12	Minimum Speed	MIN_SPEED	08H
5	OCD Threshold	OCD_TH	13H
11	Full-Step Speed	FS_SPD	15H
8	Step Mode	STEP_MODE	16H
8	Alarm Enables	ALARM_EN	17H
16	Status	STATUS	1BH
16	IC Configuration	CONFIG	1AH
Voltage Mode Configuration			
8	Holding KVAL	KVAL_HOLD	09H
8	Constant Speed KVAL	KVAL_RUN	0AH
8	Acceleration Starting KVAL	KVAL_ACC	0BH
8	Deceleration Starting KVAL	KVAL_DEC	0CH
14	Intersect Speed	INT_SPEED	0DH
8	Start Slope	ST_SLP	0EH

8	Acceleration Final Slope	FN_SLP_ACC	0FH
8	Deceleration Final Slope	FN_SLP_DEC	10H
4	Thermal Compensation Factor	K_THERM	11H
5	Stall Threshold	STALL_TH	14H
Current Mode Configuration			
8	Holding Reference Voltage	TVAL_HOLD	09H
8	Constant Speed Reference Voltage	TVAL_RUN	0AH
8	Acceleration Reference Voltage	TVAL_ACC	0BH
8	Deceleration Reference Voltage	TVAL_DEC	0CH
8	Fast Decay Setting	T_FAST	0EH
8	Minimum ON Time	TON_MIN	0FH
8	Minimum OFF Time	TOFF_MIN	10H

:Table (1)POWERSTEP01 integrated circuit registers .

For more details on the IC registers, **see page 51 of the technicaldatasheet**.

6. Messaging Commands

Communication with the circuit is done by sending frames consisting of 9 bytes. Figure (2) shows the general shape of the frame.

H1	H2	NBR	CMD	B0	B1	B2	B3	CRC

Figure (2): General shape of the messaging frame.

where:

H1: A fixed byte assigned the value :0x55 .(hexadecimal representation).

H2: Also a constant byte assigned the value :0xAA (hexadecimal representation).

NBR: Circuit number , which takes the value 0 in our case.

CMD: command :

B0…B3: Four bytes used to store data :B0 represents the lowest byte and B3 represents the highest byte.

CRC: This byte is used to verify the validity of the frame at the receiving end, where : it is assigned the value resulting from the XOR operation between the following bytes: CMD, B0, B1, B2, B3.

Below are the messaging commands, with all bytes written in hexadecimal notation:

1.6. Assign Record Value Command

The value of a register in the integrated circuit POWERSTEP01 is assigned by sending two frames: the first specifies the address of the register to which the value is to be assigned (the addresses are shown in Table (1)), and the second is for sending the desired value.

The register address is sent in a frame with a CMD byte value of 6 and an assignment frame with a CMD byte value of 7. For example, let's say we want to assign the value to the 1032 INT_SPEED register. The address of this register is 0x0D and the hexadecimal representation of 1032 is 0x0408, so the following two frames are sent

Firstly:

55	AA	00	06	0D	00	00	00	0B

Figure (3): INT_SPEED register address setting frame .

secondly:

55	AA	00	07	08	04	00	00	0B

Figure (4): INT_SPEED register value assignment frame.

2.6. Get Status command

This command is sent when you want to know the status of the circuit (request the value of the STATUS register. The value of the CMD byte in this frame is 2. When this command is sent, the drive circuit sends the value of the STATUS register (two bytes), where the highest byte is sent and then the lowest byte in order. This command is useful in knowing the status of the drive circuit and is very important in case of any error. When any problem occurs in the drive circuit such as high temperature or a drop in the voltage of driving the gate GATE of at least one of the drive bridge transistors or any other problem, the drive circuit stops working and does not return to normal working condition until the problem that caused the stop is solved and then the Get Status command is sent. When any problem occurs, the drive circuit sends the word (ERROR) in capital letters and to know the error, the Get Status command can be sent to read the value of the status register and determine the cause of the problem. After determining the cause of the problem and solving it, the command must be sent again as mentioned above, otherwise the circuit will not work again.

55	AA	00	02	00	00	00	00	02

Figure (5): Circuit status frame.

For more information on the contents of this record, please see page 71 of the technicaldatasheet.

3.6. Record value request command

This is done by assigning 5 to the CMD command byte and the register address value to byte B0. Let's say we request the register value (KVAL_DEC) then the following frame is sent:

55	AA	00	05	0C	00	00	00	09

Figure (6): KVAL_DEC register value request frame.

The circuit sends the value of the record to the investor, and the number of bytes returned varies according to the dimension of the record, noting that the bytes are sent starting with the most significant byte and ending with the least significant byte.

4.6. Switch to external signal mode

We mentioned earlier that the drive circuit supports the drive mode from an external square signal, where one microstep is moved at each ascending front of the external square signal. Switching to this mode is done by sending the following frame if you want to move in the forward direction :

55	AA	00	0D	01	00	00	00	0C

Figure (7): Frame for switching to driving mode via an external signal in the direct direction.

In the opposite case, the following frame is sent:

55	AA	00	0D	00	00	00	00	0D

Figure (8): Frame for switching to driving mode via an external signal in the opposite direction.

5.6. Speed Send Command

A speed command is sent by assigning 8 to the CMD command byte and the speed register value to the data byte. The relationship between the speed register value and the speed insteps/s is given by:

$$\frac{Step}{s} = \frac{SPEED.2^{-28}}{tick}$$

$$tick = 250ns$$

Suppose we want the motor to rotate at 4000 steps/s then the value of the speed register is:

$$\text{SPEED} = \frac{4000 \times 250 \times 10^{-9}}{2^{-28}} = 268435$$

The hexadecimal representation of this number is 0x41893 and therefore the frame sent is:

55	AA	00	08	93	18	04	00	87

Figure (9): Speed transmission frame with a value of 4000 steps/second.

a. SoftStop Order

55	AA	00	09	00	00	00	00	09

Figure (10): Soft stop frame.

7.6. Hard Stop

55	AA	00	0A	00	00	00	00	0A

Figure (11): Sudden stop frame.

8.6. Acceleration division assignment order

As mentioned earlier: When you want to change the value of any of the registers, you first send a frame to determine the address of the register by assigning the value 6 to the CMD byte then send the value of the register in the next frame in which you , assign the value 7 to the CMD byte .

The relationship between the register value and the acceleration is given by :

$$\frac{Step}{s^2} = \frac{ACC.2^{-40}}{tick^2}$$

$$tick = 250ns$$

Suppose we want to assign 5000 to the acceleration, then the value of the registeris :

$$ACC = \frac{5000 \times (250 \times 10^{-9})^2}{2^{-40}} = 343$$

The representation of this number in hexadecimalis 0x0157 Therefore, the frames . sent are as follows:

55	AA	00	06	05	00	00	00	03

Figure (12): Frame for determining the address of the acceleration register.

55	AA	00	07	57	01	00	00	51

Figure (13): Reference frame for the value 5000 for the acceleration register.

For the deceleration register the equation is the same as for the acceleration, but what changes is the value of the register address in the first frame instead of being 0x05H it becomes 0x06H as shown in Table (1).

9.6. Set minimum speed value

The relationship between the minimum speed and theregister value is given by:

$$\frac{Step}{s} = \frac{SPEED.2^{-24}}{tick}$$

$$tick = 250ns$$

Suppose we want the minimum speed to be 10 steps/second then the value of the register can be calculated as follows:

$$\mathrm{MIN}_{\mathrm{SPEED}} = \frac{10 \times 250 \times 10^{-9}}{2^{-24}} = 41$$

The representation of this number in hexadecimalis 0x29 Therefore, the frames sent are as follows:

55	AA	00	06	08	00	00	00	0E

Figure (14): Minimum speed record address setting frame.

55	AA	00	07	29	00	00	00	2E

Figure (15): Sent frame for value 10 for the minimum speed .

10.6. Set the maximum speed value

The relationship between theregister value and the max speed is given by :

$$\frac{Step}{s} = \frac{SPEED.2^{-18}}{tick}$$

$$tick = 250ns$$

Suppose we want the maximum speed to be 15000 steps/second then the value of the register can be calculated as follows:

$$\mathrm{MIN_{SPEED}} = \frac{15000 \times 250 \times 10^{-9}}{2^{-18}} = 983$$

The hexadecimal representation of this number is 0X03D7 Therefore, the frames sent are as follows:

55	AA	00	06	07	00	00	00	01

Figure (14): Minimum speed record address setting frame.

55	AA	00	07	D7	03	00	00	D3

Figure (15): Reference frame for value 10 for the minimum speed register.

11.6. .Determine the full step speed value

Full Step speed , where the circuit automatically switches to the Full Step mode when this speed is reached, i.e. the Micro Steps are ignored if the settings are set to one of the states1/2, 1/8, 1/16,…1/128.

Theregister value is related to the speed (step/second) by the following relationship :

$$\frac{Step}{s} = \frac{(FS_{SPEED} + 0.5)2^{-18}}{tick}$$

$$tick = 250ns$$

Suppose we want to assign the value 2000 steps/second to this speed, then the value of the register can be calculated as follows:

$$FS_{SPEED} = \frac{2000 \times 250 \times 10^{-9}}{2^{-18}} - 0.5 = 130$$

The representation of this number in hexadecimalis 0x82 Therefore, the frames . sent are as follows:

55	AA	00	06	15	00	00	00	13

Figure (16): Frame for specifying the address of the interval speed record.

55	AA	00	07	82	00	00	00	85

Figure (17): Reference frame for the value 2000 for the interval speed value.

12.6. Parameter saving Command

Parameters can be saved when they are changed by sending the following command :

55	AA	00	0B	00	00	00	00	0B

Figure (18): SaveParameters command .

13.6. Reset command

The drive circuit can be reset to its default state (factory reset) by sending a reset command. The parameters can then be saved by sending a save command.

55	AA	00	0C	00	00	00	00	0C

Figure (19): The replacement order.

The initial state of the circuit is set to current mode and 1/16 microsteps and the values of theT_HOLD ,T_RUN ,T_ACC andT_DEC registers are 41, so the reference current is approximately 3.2 A.

The circuit comes with a demo program that allows the user to monitor the commands sent.

7. Circuit operating modes

The circuit operates in two modes: voltage mode and current mode. The voltage mode has high precision (1/128 microstep) but the speed that can be achieved is low, while the current mode has lower precision (1/16 microstep) but the speed that can be achieved is much higher. The sensitive coefficients in voltage mode are K_HOLD ,K_RUN ,K_ACC ,K_DEC in addition to the microstep value, while the sensitive coefficients in current mode areT_HOLD ,T_RUN ,T_ACC ,T_DEC which express the reference current value in addition to the microstep value as well. It is recommended to change these values in case the circuit is not working and keep the rest of the coefficients at their default values. Often, equal values are assigned to these coefficients in each case. For example, these values in current mode express the reference current value when accelerating, decelerating, stopping and turning, so the same values can be assigned to them to avoid complexity. For more details, please refer to the technicaldatasheet of the POWERSTEP01 IC.

8. computer program

The computer interface consists of four main sections: the drive section, the general setting section, the current mode setting section, and the voltage mode setting section. The general setting section specifies the operating mode, the number of microsteps, the identification of error, as well as the deceleration, acceleration, maximum and minimum speeds, and the full step speed, as shown in Figure 20. Each page contains the circuit status requested by the Get Status command When any parameter is . changed it can be assigned by pressing the Set button Each page also contains a . section to display the sent commands and received data.

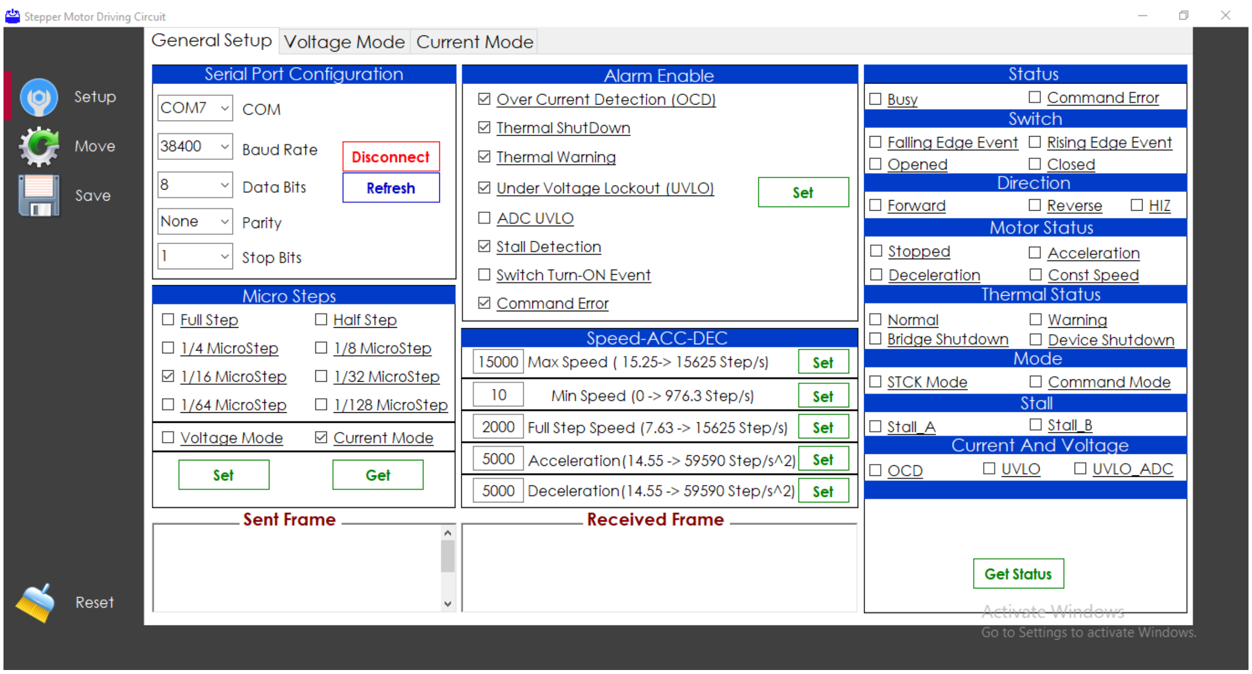


Figure (20): General Control Section.

The Voltage and Current pages contain the parameters for each mode that can be changed as desired by the user. We have already mentioned the most sensitive parameters. These two pages also contain a field to adjust the registers in the Voltage Mode Configuration and Current Mode Configuration sections , as shown in Figures 21 and 22. You can refer to the technical datasheet for more details about this registers.

In the movement section, you can change the movement settings such as deceleration nd acceleration, in addition to changing the speed in both modes, moving by a certain number of steps, and changing the work mode to the external signal mode, as shown in Figure (23).

It should be noted that the voltage and current operating modes share the same CONFIG register so the register value must be assigned to the mode considered , when switching between them.

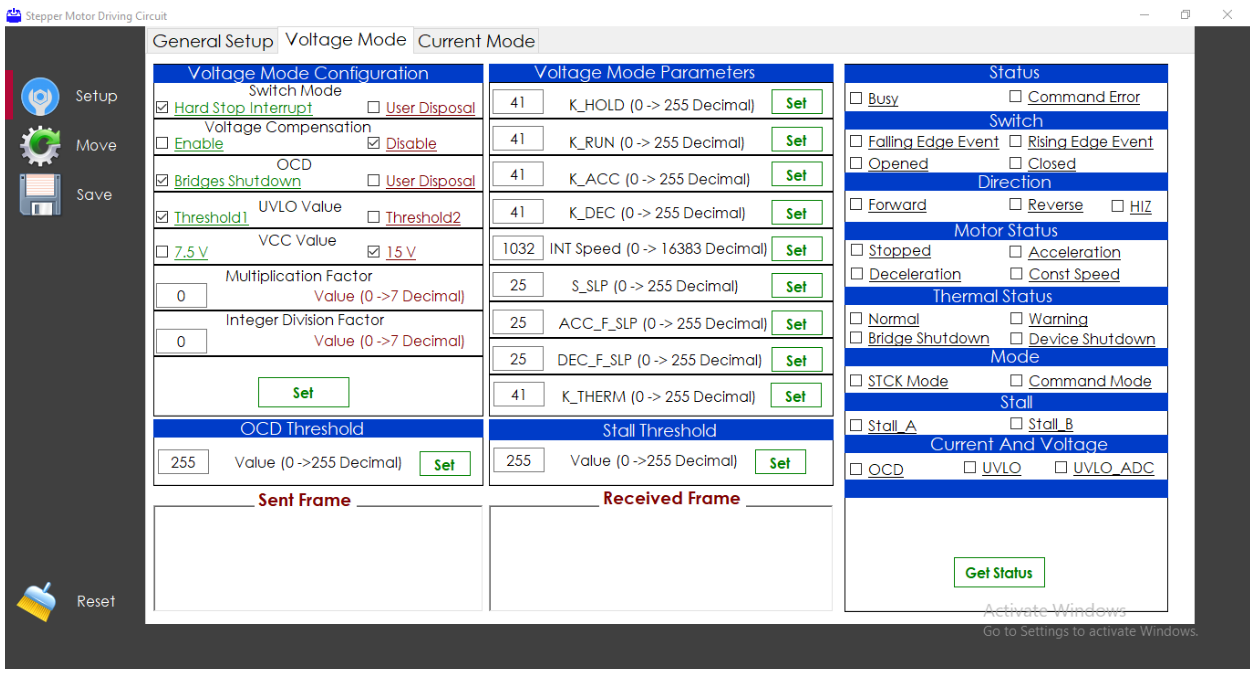
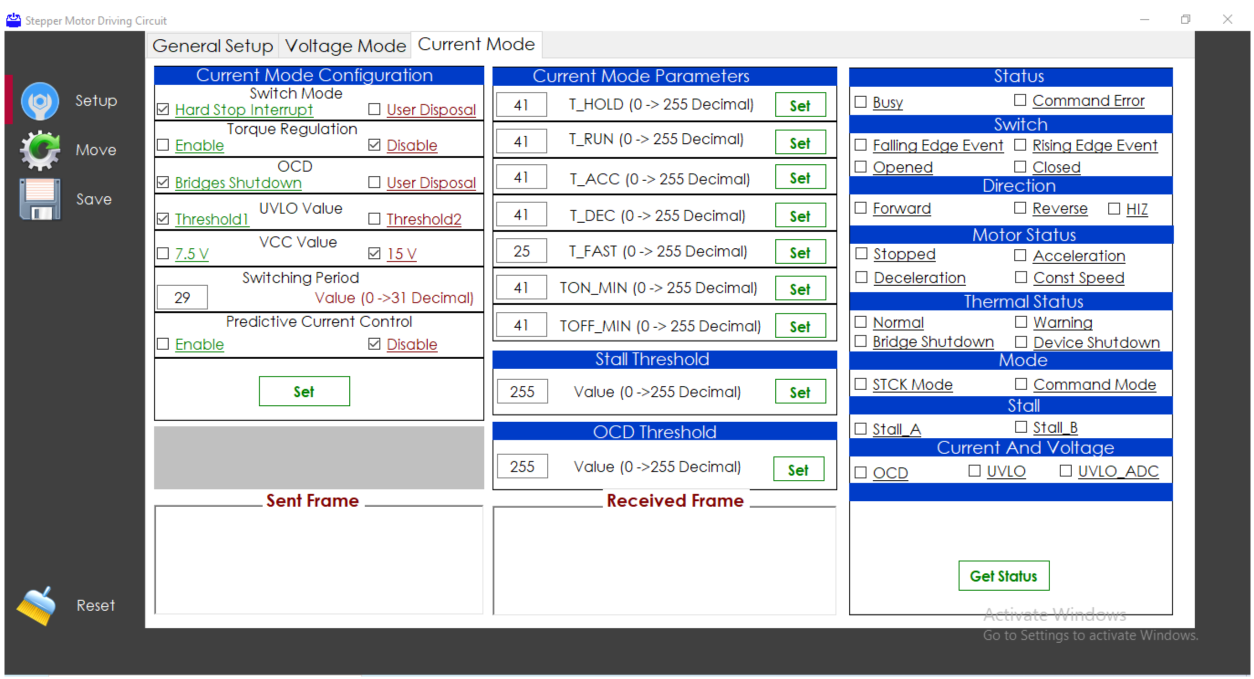
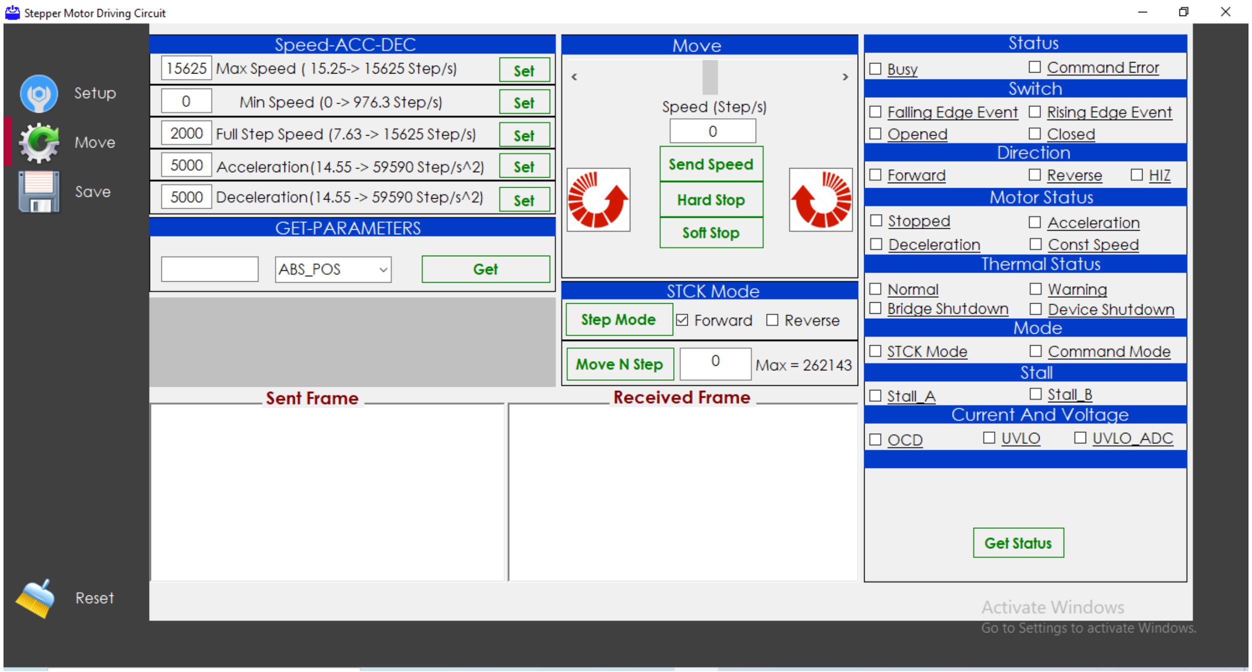


Figure (21): Voltage pattern section.



Section (22): Current Pattern Section.



:Figure (23)Movement section.