50A DC brushed motor PWM speed controller

DATASHEET FOR: Motor PWM HL4050 **v1.22**

TECHNICAL SPECIFICATIONS:

Supply voltage:	10-50V
Output current:	70A peak 50A continuous at 25°C **PCB version needs additional heatsink!**
Minimal duty cycle:	0%
Maximal duty cycle:	97%
Protections:	Over current protection, Output short protection with minimal load inductance
Switching frequency:	20kHz
Power connection:	M5

PIN DESCRIPTION:

Motor+: The motor negative connection.
Motor-: The motor positive connection.
Suppy+: The positive power supply connection 10-50V maximal voltage.
Suppy-: The negative supply connection.
Speed GND: Ground for the speed potentiometer. Keep it isolated from the Supply GND.
Speed:
Speed potentiometer input. Use 1-10 kOhm linear potentiometer or 0-5V controll voltage.

- Potetniometer mode: 0V = 0% duty cycle
  4,95V = 97% duty cycle
- Throttle mode: 0.8V = 0% duty cycle
  4,2V = 97% duty cycle
- *USART mode: Analoge input.

+5V: +5V reference voltage for the speed potentiometer and for small external circuits. Max load current 50mA.

Brake/PWM input:

- Potentiometer or throttle mode: brake switch input.
- PWM mode1: 0-100% PWM: You can use your own PWM controll signal from your PLC or microprocessor. The minimum controll PWM frequency is 100Hz. The step resolution 1us. The acceptable voltage level: 3,3-10V.
- PWM mode2: RC wirelles receiver input 1,5msec pulse for the 0% duty cycle and 2,5msec pulse for the 97% duty cycle.
- PWM, logical and pulse counter input in USART mode

Mode switch:
1OFF-2OFF: Hall throtle mode
1OFF-2ON: Potentiometer mode
1ON-2OFF: PWM input mode
1ON-2ON: RC-receiver PWM mode
3OFF: Autoregen brake OFF (let the motor to spin without gas)
3ON: Autoregen ON

Controll potentiometers:
P1: Curren limit potentiometer 0-50A (if the current limit is zero the motor won't start)
P2: Acceleration limit potentiometer 0-41sec
P3: Regenerative current limit potentiometer

LED light on when:
Current limit is active output voltage and current is limited
Supply voltage is too high.
while USART communication

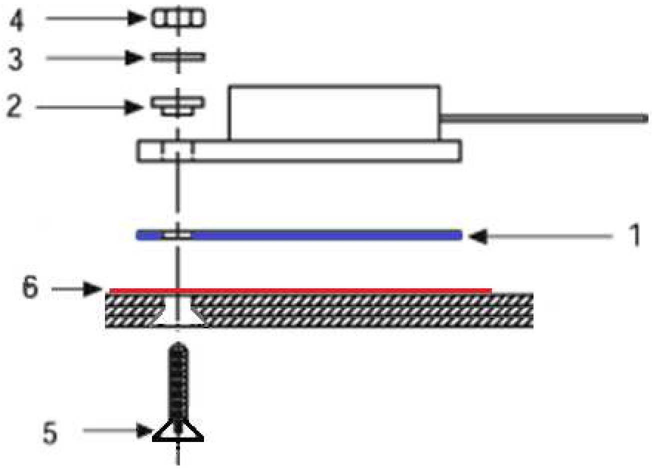
Istructions for the heatsink:

The PCB version of the controller needs an additional heatsink or aluminium case to cool the power transistors. The dimensions of the heatsink can change depending on the supply voltage and the output current and of cource the ambient temperature and the convection. Normally use a 100mmx100mx25mm (4"x4"x 1") aluminium heatsink or equvalend aluminium case.

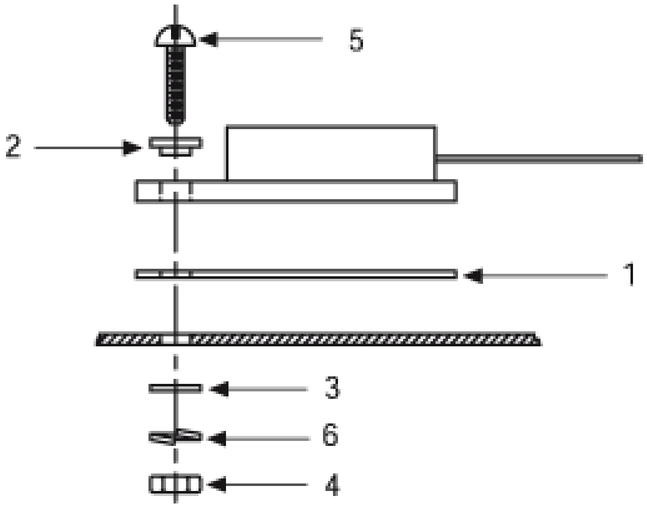
Step by step:

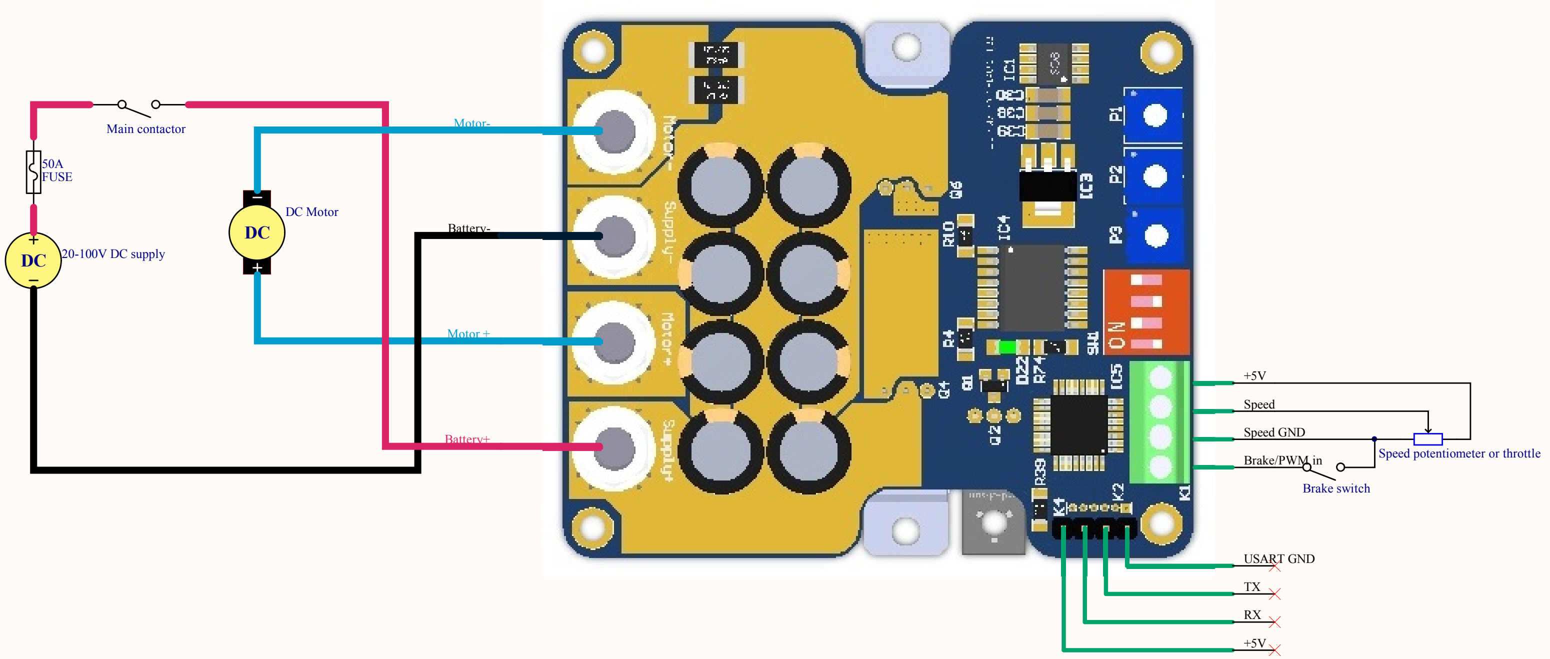
1. In the right positions drill M3 holes for the power transistors and for the fastening screws.
2. The hole's edges needs top be clear and burr free (small burr under the transistors can harm and penetrate the isolators causing short between the transistors and the heatsink).
3. Place heat conductor paste (6) under the isolator.Quanity is not importand it need to be only a thin layer.
4. Place the silicone isolators in place. (1)
5. Make sure all the transistors flags are parallel with the heatsink. Having gap under the transisotrs flag's can cause bad thermal contact ovetheating and damage.
5. Place the board and the transistors on the heatsink and fix it with the screws (5). Use the small white plastic isolator rings (2) under the screws to avoid the short between the metal flags and the screws.
6. Make sure all the crews are tight enogh to have the best thermal contact.
7. Check the isolation between the transostors flag's and the heatsink with digital ohm meter

Fixing from down side:



Fixing from up side:





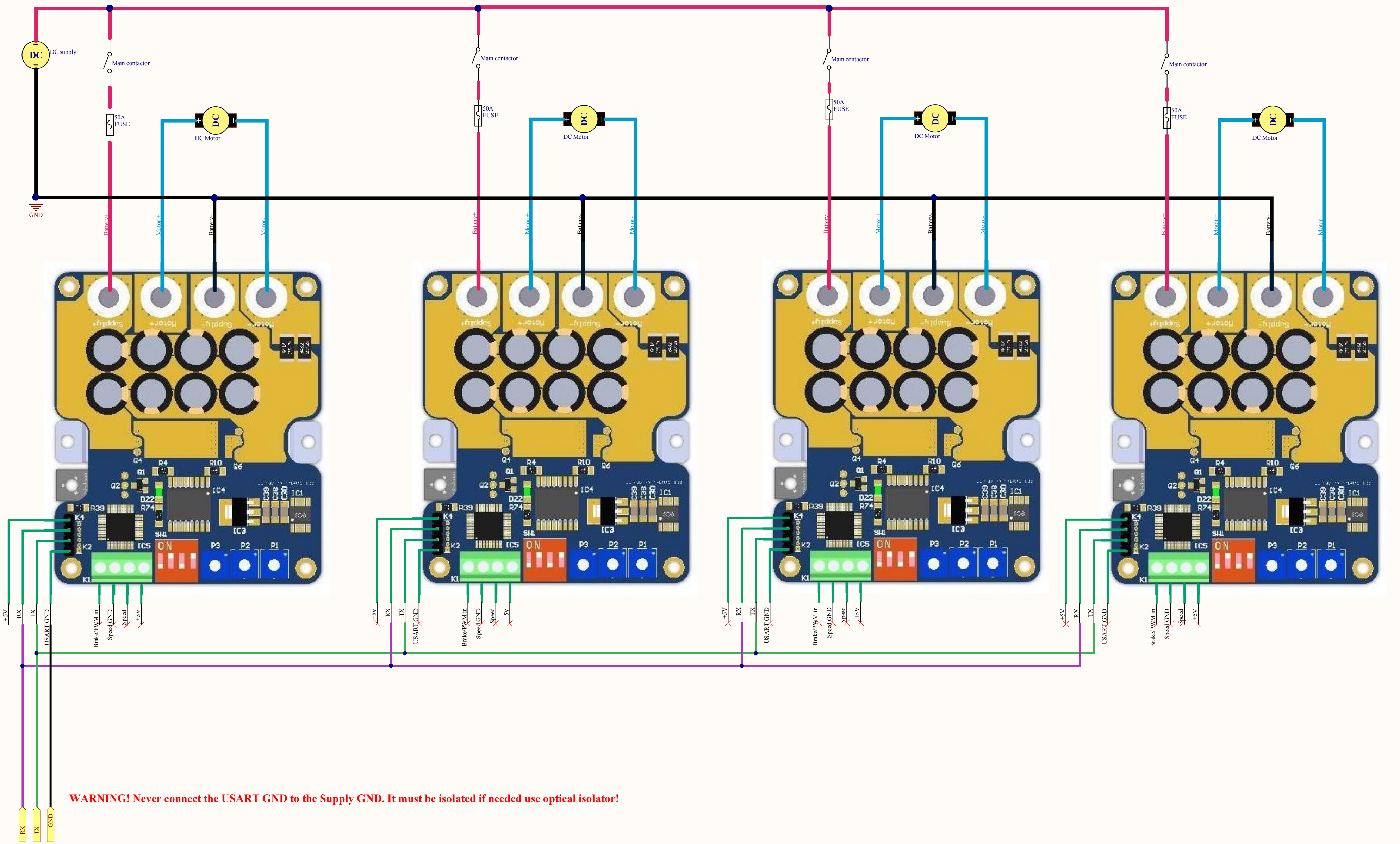
Title

Standard potentiometer or throttle + brake switch

Size	Number	Revision
A3	HL4050	Rev1

Date: 2017.07.28. Sheet of

File: E:\Projektek\..\Wir1.SchDoc Drawn By:

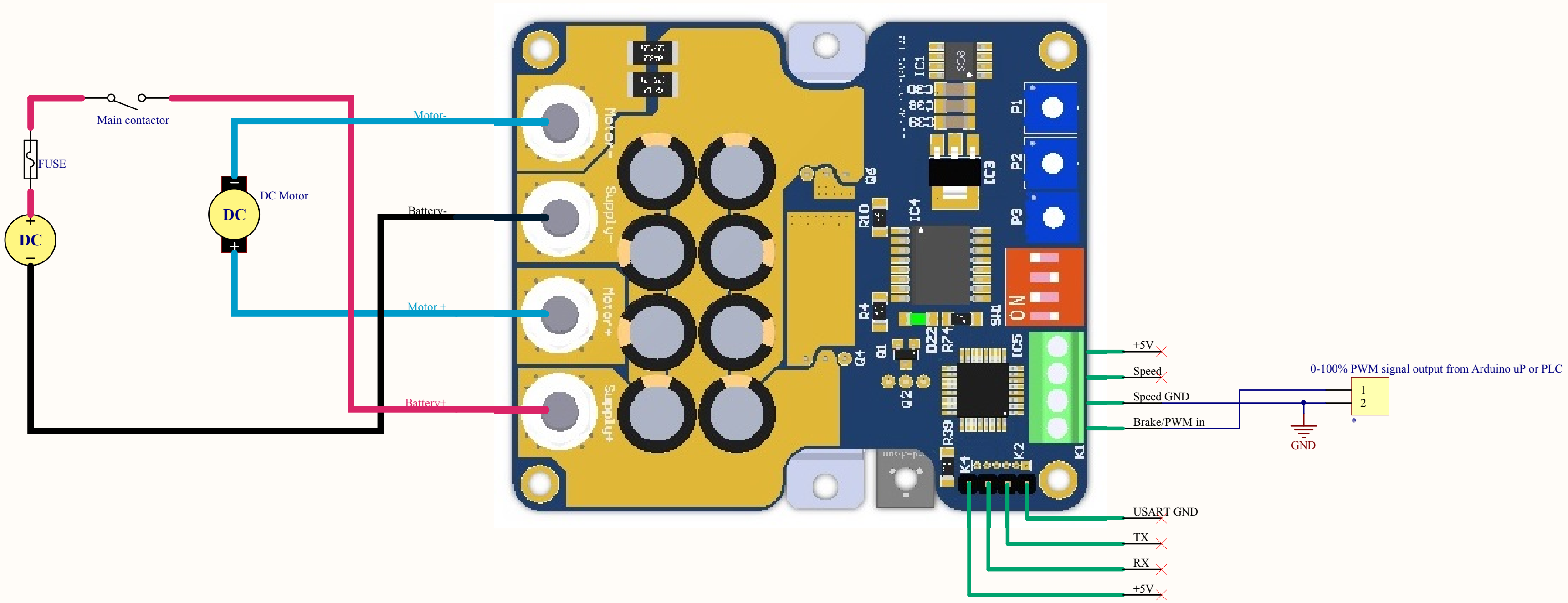


WARNING! Never connect the USART GND to the Supply GND. It must be isolated if needed use optical isolator!

Title
Connecting several units on one RS232 line

Size A2 | Number HL4050 | Revision Rev1

Date: 2017.07.27. | Sheet of
File: E:\Projektek\..\Wir2.SchDoc | Drawn By:



WARNING! Never connect the SPEED GND to the Supply GND. If you want to connect the same PWM for more controller use only one SPEED GND input.

Title PWM input mode		
Size A3	Number HL4050	Revision Rev1
Date: 2017.07.27.		Sheet of
File: E:\Projektek\..\Wir3.SchDoc		Drawn By:

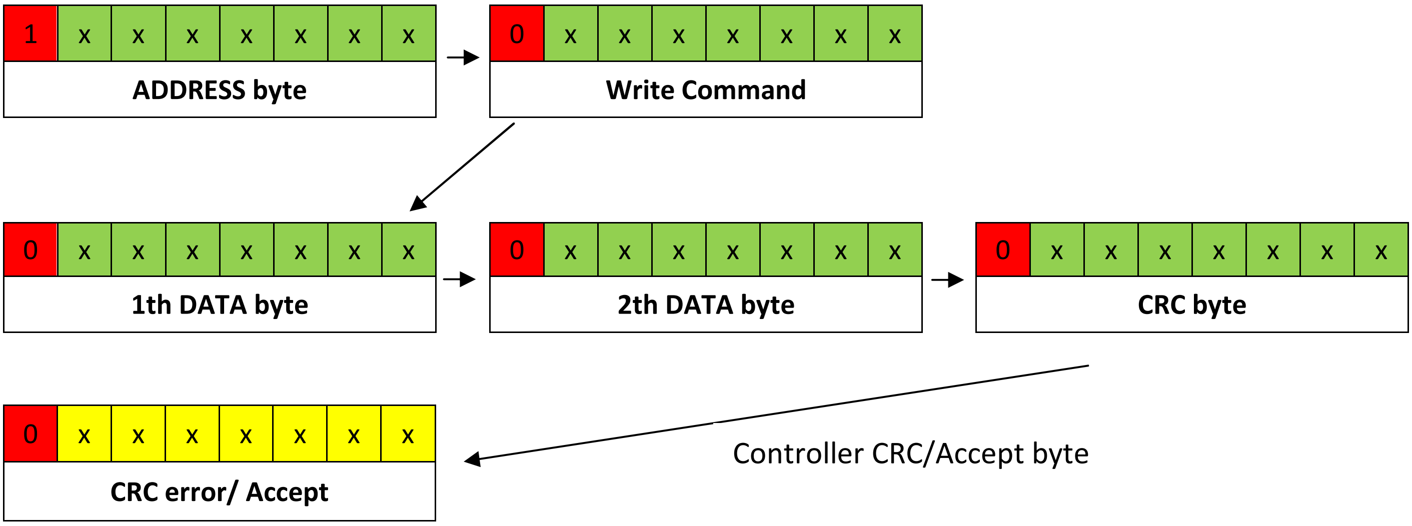
USART Data transmission:

revision: 1.   USART firmware v1.0
Supported baud rate: 38400 baud/sec
Supported bit rate: 1 start 8 data bit 1 stop and no parity bit.

Sending data:

The controllers are working in SLAVE mode USART communication. It means that every commands are addressable and only the controller which has the same address will accept the command otherwise it goes into stanby mode. When an address match occurs the the controller wakes up and accept the data. As long as the controller does not get other address which is not its own address remains in active mode.

Sending data from master device must be 5 bytes:

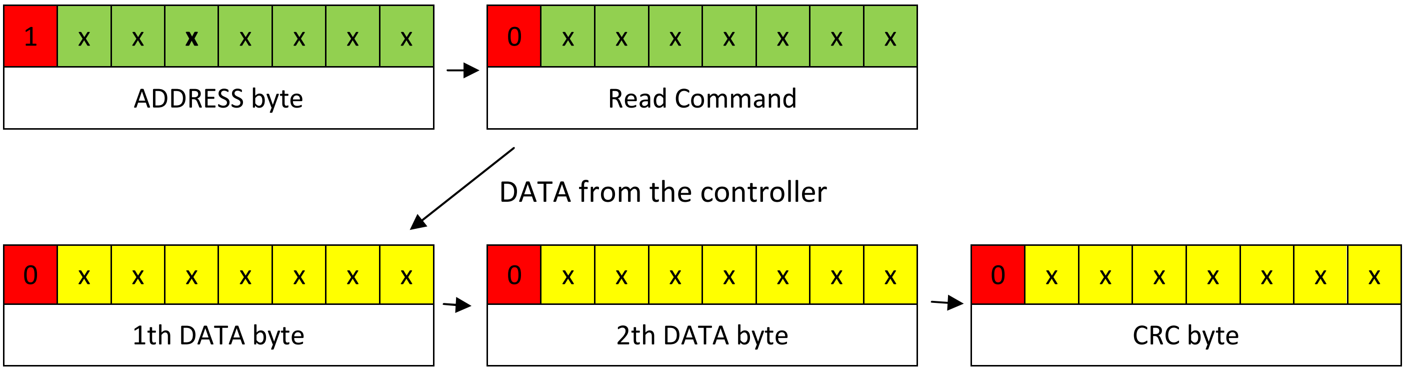


Every ADDRESS byte MSB bit must be "1" any other which is not an ADDRESS byte must be "0".
For robust and errorless communication CRC byte can be used to check the data transmission errors.
CRC ckeck byte calculated: ((Write command byte + 1th DATA byte + 2th DATA byte) & 0x7F)

Receiving successfully the write command and the DATA bytes the controller send a confirmation byte with his USART address (0x00 | address) or CRC error message with his address (0x80 | adddress).You can use this byte to know when to send the next data.

Reading data:

Requesting data from master device must be 2 bytes when the controller received the 2th byte it will send the answer 3 bytes (1th DATA byte ; 2th DATA byte ; CRC)
CRC ckeck byte calculated: ((Read command byte + 1th DATA byte + 2th DATA byte) & 0x7F)



Controll Registers MAP and value ranges:

Model: **HL4050 v1.22**

Register name	Command	Type	Min Value	Max value	Value Range	Remark
Speed	**0x00**	**Write**	**0x0000**	**0x0FFF**	**97% duty**	***Other factors may change the output duty internally.***
Current Limit	0x01	Write	0x0000	0x0FFF	60,3A	*This value is internally limited to 50A.*
Regeneraive Current Limit	0x02	Write	0x0000	0x0FFF	60,3A	*This value is internally limited to 50A.*
Acceleration Limit	0x03	Write	0x0000	0x0FFF	41 sec	*Minimum=0x0001 (10msec)*
Deceleration limit	0x04	Write	0x0000	0x0FFF	41 sec	*Minimum=0x0001 (10msec)*
Turning off minimum voltage	0x05	Write	0x0000	0x0FFF	75.9V	
Turning on minimum voltage	0x06	Write	0x0000	0x0FFF	75.9V	*Tipically: 10.0V*
Turning on maximum voltage	0x07	Write	0x0000	0x0FFF	75.9V	*Defaul valuet: 2600 (48,2V)*
Turning off maximum voltage	0x08	Write	0x0000	0x0FFF	75.9V	*Internally limited on 2800 (51,8V)*
I x R compensation	0x09	Write	0x0000	0x0FFF		**NOT Released!**
Lock Variables	0x20	Write				
Unlock Variables	0x21	Write				
Save	0x22	Write				
USART Address	0x26	Write	0x0000	0x007F		*Default address 0x00*
Current measurement calibration	0x27	Write	0x0000	0x3FFF		**NOT Released!**
Set Configuration bits	0x30	Write				
RESET Configuration bits	0x31	Write				
Speed	0x40	Read	0x0000	0x0FFF	97% duty	
Current Limit	0x41	Read	0x0000	0x0FFF	60,3A	*This value is internally limited to 50A.*
Regeneraive Current Limit	0x42	Read	0x0000	0x0FFF	60,3A	*This value is internally limited to 50A.*
Acceleration Limit	0x43	Read	0x0000	0x0FFF	41 sec	*Minimum=0x0001 (10msec)*
Deceleration limit	0x44	Read	0x0000	0x0FFF	41 sec	*Minimum=0x0001 (10msec)*
Turning off minimum voltage	0x45	Read	0x0000	0x0FFF	75.9V	
Turning on minimum voltage	0x46	Read	0x0000	0x0FFF	75.9V	*Tipically: 10.0V*
Turning on maximum voltage	0x47	Read	0x0000	0x0FFF	75.9V	*Defaul valuet: 2600 (48,2V)*
Turning off maximum voltage	0x48	Read	0x0000	0x0FFF	75.9V	*Internally limited on 2800 (51,8V)*
Heatsink Temperature	0x49	Read	0x0000	0x0FFF		**NOT Released!**
Mainboard Temperature	0x4A	Read	0x0000	0x0FFF		**NOT Released!**
Supply Voltage	0x4B	Read	0x0000	0x0FFF	75.9V	
Speed ADC Input	0x4C	Read	0x0000	0x0FFF	4,85V	
Load Current	0x4D	Read	0x0000	0x0FFF	+/- 120A	*0A=0x07FF **
Input Frequency Period Register	0x50	Read	0x0000	0x03FF		**NOT Released!**
Input Pulse Width	0x51	Read	0x0000	0x03FF		**NOT Released!**
Input Pulse Counter	0x52	Read	0x0000	0x03FF		**NOT Released!**
Status Register	0x60	Read				

* Current measurement accuracy is +/-3% before current measurement calibration

Speed Register:

Type: Write/Read
Write command: 0x00
Read command: 0x40

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	SPEED[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Speed register gives the base value of the output voltage (duty cycle). The final value depending some other factors. The final duty cycle depending on several other factors (current limits, regen modes..)

With default settings the speed register equal to the controller speed input ADC value modified by the acceleration/deceleration settings. Writing into this register this will be the new base value of the output voltage as long as the Speed input ADC has the same value as it was before the overwrite. When the Speed input voltage changes (turning up/down the speed potentiometer) the controller going to overwite the register with the value of the potentiometer.

*It is possible to lock the register value (speed potentiometer can't overwrite the speed register) for this feature use the **Lock Variable Register**.*

*The register set and cleared by USART write or Speed potentiometer input and loses it's value after restarting the controller. Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **SPEED [11:0]:** Speed register value
Max value is 4095 it is the 97% duty cycle.
Min value is 0 it is the 0% duty cycle.

Note: final duty cycle may change depending on other factors.

Current Limit Register:

Type: Write / Raed
Write command: 0x01
Read command: 0x41

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	CURRENT_LIMIT[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Current Limit Register sets the value of the current limitation. With default settings the current limit register is equal to the controller's current limit potentiometer setting. Writing into this register overwites the value of the potentiometer and holds the value as long as the potentiometer has the same value as it was before the overwrite. When the potentiometer changes (turning up/down the potentiometer) the controller going to overwite the register with the value of the potentiometer.

*It is possible to lock the register value (potentiometer can't overwrite the register) for this feature use the **Lock Variable Register**.*

*The register set and cleared by USART write or the current limit potentiometer and loses it's value after restarting the controller. Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **CURRENT_LIMIT [11:0]:** Curretn limit register value

Max value is 4095

Min value is 0

*Note: for the corresponding curret value check the **Register Value Range** table.*

Regeneraive Current Limit Register:

Type: Write / Raed

Write command: 0x02

Read command: 0x42

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	REGEN_LIMIT[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Regenerative Current Limit Register sets the value of the current limitation. With default settings the current limit register is equal to the controller's potentiometer setting. Writing into this register overwites the value of the potentiometer and holds the value as long as the potentiometer has the same value as it was before the overwrite. When the potentiometer changes (turning up/down the potentiometer) the controller going to overwite the register with the value of the potentiometer.

*It is possible to lock the register value (potentiometer can't overwrite the register) for this feature use the **Lock Variable Register**.*

*The register set and cleared by USART write or the potentiometer and loses it's value after restarting the controller. Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **CURRENT_LIMIT [11:0]:** Regenerative current limit register value

Max value is 4095

Min value is 0

*Note: for the corresponding curret value check the **Register Value Range** table.*

Acceleration Limit Register:

Type: Write / Raed
Write command: 0x03
Read command: 0x43

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	ACCELERATION_LIMIT[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Acceleration Limit Register sets the time how fast the output follows the speed register's value under raising condition. With default settings it is equal to the controller's acceleration (RAMP) potentiometer's setting. Writing into this register overwites the value of the potentiometer and holds the value as long as the potentiometer has the same value as it was before the overwrite. When the potentiometer changes (turning up/down the potentiometer) the controller going to overwite the register with the value of the potentiometer.

*It is possible to lock the register value (potentiometer can't overwrite the register) for this feature use the **Lock Variable Register**.*

*The register set and cleared by USART write or the potentiometer and loses it's value after restarting the controller. Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **CURRENT_LIMIT [11:0]:** Acceleration limit register value
Max value is 4095
Min value is 0

*Note: for the corresponding time value check the **Register Value Range** table.*

Deceleration limit register:

Type: Write / Raed
Write command: 0x04
Read command: 0x44

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	DECELERATION_LIMIT[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Deceleration Limit Register sets the time how fast the output follows the speed register's value under falling condition. With default settings it is disabled in the controller.

*Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **DECELERATION_LIMIT [11:0]:** Deceleration limit register value

Max value is 4095
Min value is 0

Note: for the corresponding time value check the ***Register Value Range*** *table. The minimum value is 1.*

Turning off minimum voltage register:

Type: Write / Raed
Write command: 0x05
Read command: 0x45

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	TURN_OFF_MIN[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Turning Off Minimum voltage register sets the voltage level where the controller will shut down if the supply voltage is lower than the register value. With default settings it is disabled in the controller.Writing and locking this register activates the function. When the controller shuting down because of the low supply voltage level the corresponding status register gets an update about the low supply voltage.
Saving the value into the FLASH use the ***Save Register****.*

Bit 11:0 **TURN_OFF_MIN [11:0]:**
Max value is 4095
Min value is 0

Note: for the corresponding voltage level check the ***Register Value Range*** *table.*

Turning on minimum voltage register:

Type: Write / Raed
Write command: 0x06
Read command: 0x46

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	TURN_ON_MIN[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Turning On Minimum voltage register sets the voltage level where the controller will turn back from low supply status if the supply voltage is higher than the register value. With default settings it is disabled in the controller.Writing and locking this register activates the function.

*Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **TURN_ON_MIN [11:0]:**
Max value is 4095
Min value is 0

*Note: for the corresponding voltage level check the **Register Value Range** table.*

Turning on maximum voltage register:

Type: Write / Raed
Write command: 0x07
Read command: 0x47

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	TURN_ON_MAX[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Turning On maximum voltage register sets the voltage level where the controller will turn back from high supply voltage status if the supply voltage is lower than the register value. With default settings it is disabled in the controller.Writing and locking this register activates the function.
*Saving the value into the FLASH use the **Save Register**.*

Bit 11:0 **TURN_ON_MAX [11:0]:**
Max value is 4095
Min value is 0

*Note: for the corresponding voltage level check the **Register Value Range** table.*

Turning off maximum voltage register:

Type: Write / Raed
Write command: 0x08
Read command: 0x48

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	TURN_OFF_MAX[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Turning Off Maximum voltage register sets the voltage level where the controller will shut down if the supply voltage is higher than the register value. With default settings it is disabled in the controller.The internal maximum voltage settings may rewrite this register. Writing and locking this register activates the function. When the controller shuting down because of the high supply voltage level the corresponding status register gets an update about the high supply voltage.

Saving the value into the FLASH use the ***Save Register****.*

Bit 11:0 **TURN_OFF_MAX [11:0]:**

Max value is 4095

Min value is 0

Note: for the corresponding voltage level check the ***Register Value Range*** *table.*

~~I x R compensation register:~~

Command: 0x09

Not released!

Unlock Variables Register:

Type: Only write

Write command: 0x21

	15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
2th DATA byte									0	x	x	x	x	x	x	x
			Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ACC	REG	CUR	SPD
1th DATA byte		0	x	x	x	x	x	x	x							

Bit 3 **ACC:** Writing "1" clears the ALK bit in the Lock Variables Register.

Bit 2 **REG:** Writing "1" clears the RLK bit in the Lock Variables Register

Bit 1 **CUR:** Writing "1" clears the CLK bit in the Lock Variables Register

Bit 0 **SPD:** Writing "1" clears the SLK bit in the Lock Variables Register

Lock Variables Register:

Type: Only write
Write command: 0x20

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ACC	REG	CUR	SPD
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Locking any register means only USART update can overwrite the value. The on board potentiometers can't.

Bit 3 **ACC:** Writing "1" locks the Speed Register.

Bit 2 **REG** Writing "1" locks the Current Limit Register.

Bit 1 **CUR:** Writing "1" locks the Regeneration Limit Regitster.

Bit 0 **SPD:** Writing "1" locks the Acceleration Limit Register.

Save Register:

Type: Write
Write command: 0x22

								2th DATA byte							
								0	x	x	x	x	x	x	x
		CONF2	Res	Res	Res	Res	OFF_H	ON_H	OFF_L	ON_L	DEC	ACC	RLIM	CLIM	SPD
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

These bits starts a memory saving operation into the controller's FLASH memory. Writing into this register as a repeated operation is not recommended**. The FLASH memory rewritable life cycle is limited about 10.000 saving operation! Under the save operation the PWM controller stops and no other operation can be used!** When a memory saving occurs the data will be reloaded from the memory after every restart. Saving a register data automatically locks the particular register and only USRAT update can rewrite it.

Bit 13 **CONF2:** Configuration 2 Register.
0: no saving
1: Save the given value into memory.

Bit 8 **OFF_H:** Save turning off maximum value.
0: no saving
1: Save the given value into memory.

Bit 7 **ON_H:** Save turning on maximum value.
Refer to the Bit 8 OFH description.

Bit 6 **OFF_L:** Save turning off minimum value.
Refer to the Bit 8 OFH description.

Bit 5 **ONL:** Save turning on minimum value.
Refer to the Bit 8 OFH description.

Bit 4 **DEC:** Save deceleration value.
Refer to the Bit 8 OFH description.

Bit 3 **ACC:** Save acceleration value.
Refer to the Bit 8 OFH description.

Bit 2 **REG:** Save regenerative current limit value.
Refer to the Bit 8 OFH description.

Bit 1 **CUR:** Save current limit value.
Refer to the Bit 8 OFH description.

Bit 0 **SPD:** Save speed value.
Refer to the Bit 8 OFH description.

USART Address Register:

Type: Write

Write command: 0x26

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	Res	Res	Res	Res	Res	USART ADDRESS [6:0]						
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

USART Address bits is a unique 7bit address which marks the given controller. It allows to connect several controllers or even other USART devices to connect to the same RX TX channel. The dafault USART address is 0x00. If you want to controll two or more controllers throught the USART you have to change the address first one by one. After this it is possible to connect more devices to the same USART channel. USART Address will be saved into the FLASH memory and reloaded from the memory after every restart.

The controllers sends out their unique USART address right after the system start. This helps to recover and recognise a forgotten address.

Bit 11:0 **USART ADDRESS [6:0]:** Acceleration limit register value

Max value is 0x7F

Default value: 0x00

~~Current measurement calibration register:~~

Command: 0x27

Not released!

Set Configuration 1 register:

Type: Write

Write Command: 0x30

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	Res	Res	Res	Res	Res	Res	Res	DM	BM	SCR [1:0]		UC
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Bit 4 **DM:** Selecting operation mode

This bit set and cleared by USART write and cleared by hardware after restart.

0: Drive mode.

1: Brake mode or free running mode depending on **BM** bit.

Bit 3 **BM:** Braking mode
This bit set and cleared by USART write and cleared by hardware after restart.
0: Manual brake (free running mode)
1: Auto brake (automatic regeneration)

Note: *Manual brake let the motor spinning free after turning down the speed or sending 0 speed command. Auto brake start automatic current regeneration into the supply when the motor internal voltage is higher than the speed.*

Bit 2:1 **SRC[1:0]:** Speed reset controll
This bit set and cleared by USART write and cleared by hardware after restart.
00: No speed reset after no USART speed update.
01: Speed reset after 50msec without new USART speed update.
10: Speed reset after 500msec without new USART speed update.
11: Speed reset after 2800msec without new USART speed update.

Bit 0 **UC:** USART controll bit
This bit set and cleared by USART write and cleared by hardware after restart.
0: Disable the configuration bits
1: Enable the configuration bits

RESET Configuration 1 register:

Type: Write
Write Command: 0x31

							2th DATA byte								
							0	x	x	x	x	x	x	x	
		Res	Res	Res	Res	Res	Res	Res	Res	Res	DMC	BMC	SCRC [1:0]		UCC
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Note: writing 1 clears the given bit writing 0 makes no changes.

Bit 4 **DMC:** Clears DM configuration bit. .

Bit 3 **BMC:** Clears BM configuration bit.

Bit 2:1 **CSRC[1:0]:** Clears SRC configuration bits.

Bit 0 **UCC:** Clears UC configuration bit.

Set Configuration 2 register:

Type: Write
Write Command: 0x32

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	Res	Res	Res	Res	Res	PMRE	Res	Res	Res	USART_SPD		Res
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Bit 7 **PMRE:** Pulse measurement resolution

This bit sets the resolution of the pulse(PWM) input.

0: High resolution for measuring high frequency signal

1: Low resolution

Note: for the corresponding times check the ***Register Value Range*** *table.*

Bit 2:1 **SRC[1:0]:** USART speed controll

These bits set the USART baud/sec rate.

00: 38400 Baud/sec

01:

10:

11:

Load Current Register:

Type: Read
Read Command: 0x4D

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	LOAD_CURRENT [11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

This register holds the actual otput load current.

Bit 11:0 **LOAD CURRENT [11:0]:**

Max value is 4095

Min value is 0

Note: for the corresponding curret value check the ***Register Value Range*** *table.*

Supply Voltage Register:

Type: Read
Read Command: 0x4B

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	SUPPLY_VOLTAGE[11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

This register holds the actual supply voltage value.

Bit 11:0 **SUPPLY VOLTAGE [11:0]:**
Max value is 4095
Min value is 0

Note: for the corresponding voltage value check the ***Register Value Range*** *table.*

~~Heatsink Temperature Register:~~

Not released!

Type: Read
Read Command: 0x49

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	HEATSINK_TEMPERATURE [11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

This register holds the actual temperature value.

Bit 11:0 **HEATSINK TEMPERATURE [11:0]:**
Max value is 4095
Min value is 0

Note: for the corresponding temperature value check the ***Register Value Range*** *table.*

~~Mainboard Temperature Register:~~

Not released!

Type: Read
Read Command: 0x4A

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	MAINBOARD_TEMPERATURE [11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

This register holds the actual temperature value.

Bit 11:0 **MAINBOARD TEMPERATURE [11:0]:**
Max value is 4095
Min value is 0

*Note: for the corresponding temperature value check the **Register Value Range** table.*

Speed ADC Input Register:

Type: Read
Read Command: 0x4C

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	SPEED_ADC_INPUT [11:0]											
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Controlling and locking the controller with USART Speed commands the Speed ADC input can serve other task like measuring a sensor or anything else.

Using the Speed ADC input for other task and not for the speed controll function the Speed Register Lock bit must be enabled.

Bit 11:0 **SPEED ADC INPUT [11:0]:**
Max value is 4095
Min value is 0

*Note: for the corresponding voltage value check the **Register Value Range** table.*

~~Input Frequency Period Register:~~

Not released!

Type: Read
Read Command: 0x50

								2th DATA byte							
								0	x	x	x	x	x	x	x
		INPUT_FREQUENCY__PERIOD_REGISTER [13:0]Res													
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

The controller PWM/brake input can be used for measuring pulses and other purposes. Like hall sensors encoders etc. Input Frequency Register holds the period's of the measured signal.

Bit 13:0 **INPUT FREQUENCY PERIOD REGISTER [13:0]:**
Max value is 16383
Min value is 0

*Note: for the corresponding time value check the **Register Value Range** table.*

~~Input Pulse Width Register:~~

Not released!

Type: Read
Read Command: 0x51

								2th DATA byte							
								0	x	x	x	x	x	x	x
		INPUT_PULSE_WIDTH_REGISTER [13:0]Res													
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

The controller PWM/brake input can be used for measuring pulses and other purposes. Like hall sensors encoders etc. Input Pulse Width Register holds the width of the captured pulse of the measured signal.

Bit 13:0 **INPUT PULSE WIDTH REGISTER [13:0]:**

Max value is 16383
Min value is 0

*Note: for the corresponding time value check the **Register Value Range** table.*

~~Input Pulse Counter Register:~~

Not released!

Type: Read
Read Command: 0x52

								2th DATA byte							
								0	x	x	x	x	x	x	x
		INPUT_PULSE_COUNTER_REGISTER [13:0]Res													
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

The controller PWM/brake input can be used for measuring pulses and other purposes. Like hall sensors encoders etc. Input Pulse Width Register holds the width of the captured pulse of the measured signal.

Bit 13:0 **INPUT PULSE WIDTH REGISTER [13:0]:**

Max value is 16383
Min value is 0

*Note: for the corresponding time value check the **Register Value Range** table.*

Status Register:

Type: Read
Read Command: 0x60

								2th DATA byte							
								0	x	x	x	x	x	x	x
		Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OV	UV
15. bit	14. bit	13. bit	12. bit	11. bit	10. bit	9. bit	8. bit	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
	0	x	x	x	x	x	x	x							
	1th DATA byte														

Bit 1 **OV:** Over voltage status register.
0:
1: Supply voltage is higher than the Turn off maximum register value.

Bit 0 **UV:** Under voltage status register.
0:
1: Supply voltage is lower than the Turn on minimum register value.

Examples:

Example 1: Calling 0x02 address device and writing DATA_to send: 0x05C5 value into the Speed register:

Sent bytes:

0x82	*0x00*	*0x0B*	*0x45*	*0x50*

C conversion code before sending out the bytes:

```
ADDRESS=UNIT_ADDRESS | 0x80;
COMMAND= for this check the write commands in the register table
DATA_1th= (DATA_to_send>>7) & 0x7F;
DATA_2th=DATA_to_send  & 0x7F;
CRC=( COMMAND + DATA_1th + DATA_2th) & 0x7F;
```

Example 2: Calling 0x02 address device and locking Speed variable:

Sent bytes:

0x82	0x21	0x00	0x01	0x22

C conversion code before sending out the bytes:

Refer to the Example 1.

Example 3: Calling 0x02 address device to change the regenerative brkaking mode to "auto regen" from "free running mode"and initiating regeneraitive braking with 0x062C current:

For this you need to execute 2 write comands. The first set the regenerative current limit.

Sent bytes:

0x82	0x02	0x0C	0x2C	0x3A

C conversion code before sending out the bytes:

Refer to the Example 1.

Second to set the Configuration bit:.

Sent bytes:

0x82	0x30	0x00	0x19	0x1C

C conversion code before sending out the bytes:

*Refer to the Example 1. and **Set Configuration bits register.***

Example 4: Calling 0x02 address device to read the load_current_register:

Sent bytes:

0x82	0x44

(read command)

Received bytes form the controller:

0x13	0x36	0x0D

C conversion code for the received bytes:

CRC check:

```
if(((received_byte_1th + received_byte_2th + Read_command) & 0x7F)== received_byte_3th)
        {
                load_current_register=( received_byte_1th <<7) | (received_byte_2th & 0x7F);
        }
```