

# PMC1805 (upgrade : PMC1402)

Hardware Version 1.1

## User's manual

Document Revision 1.4



Piezoelectric Technology Co., Ltd.

Piezoelectric Technology Co., Ltd.

<http://www.piezo-tech.com>

## Revision History

Date	H/W Ver.	S/W Ver.	Revision	Description
2018/6	1.0	1.0	1.0	First release
2018/8	1.0	1.0	1.1	Add ADC, limit status
2018/9	1.1	1.2	1.3	Add select TTL,RS-422 on encoder
2020/8	1.1	1.2	1.4	Add PUMR spec , dimension.

## Table of Contents

1. Notes on safety
2. Technical specifications
3. Block Diagram
4. Remote Mode
5. Command
6. PC Monitor Software

Appendix 1 PMC1805

Appendix 2 How to install virtual communication port

Appendix 3 How to Install OCX

Appendix 4 PUMR40 dimension.

Appendix 5 PUMR60 dimension.

Appendix 6 PUMR40/60 specification.

Terms and Abbreviations

## 1. Notes on safety

### 1.1 General information

Make sure to familiarize thoroughly with the contents of these instructions before starting up the PMC1804 which stands for "Piezoelectric ultrasonic motor of Rotary type controller". We will provide you with supplementary information. Please refer to the relevant instructions for PMC1805. From now, the "PMC1805" means the controller of PUMR40, PUMR60 motor.

### 1.2 Notes on use

When connecting the motor cable to the driver, you must be careful of the direction of the connector. The "PMC1805" and PUMR40/60 motor must be matched each other. For example, PUMR60 motor cannot be used with any other controller except PMC1805-02. Please confirm this. If not, the motor and driver must be damaged permanently.

The operation frequency range of the "PMC1805" is fixed at factory. Before power on, please confirm the DIP switch in "PMC1805" which must be matched with PUMR40/60 USM respectively. For details, refer to the appendix.

In the case that the "PMC1805" is malfunctioned, please contact us.

### 1.3 Notes on storage

The "PMC1805" may be subjected to a high humidity. Any moisture that may be present in the body may damage the unit. Therefore, it is important that the interior of the "PMC1805" and PUMR40/60 must be dry before running the "PMC1805".

### 1.4 Notes on maintenance

Modification and repairs of the "PMC1805" and PUMR40/60 may only be carried out by the manufacturer or appropriately authorized persons. The manufacturer is not liable for damages caused by unauthorized handling of the "PMC1805" and PUMR40/60. All warranty claims are forfeited by unauthorized handling.

## 2. Technical specifications

### 2.1 Unique features

- Stop and Start
- CW,CCW changeable( or left, right )
- Speed control
- Position control (encoder type)

### 2.2 Electric and mechanical specification

SPECIFICATIONS	PMC1805-01	PMC1805-02
Size (W x H x D)	95 x 25 x 90	
Input power	DC 12V / 2A	DC 24V / 2A
Driving voltage(Vrms)	120	
Speed adjustment	Frequency Control / VR (0V ~ 5V analog input)	
Operating temperature	-10°C ~ +50°C	

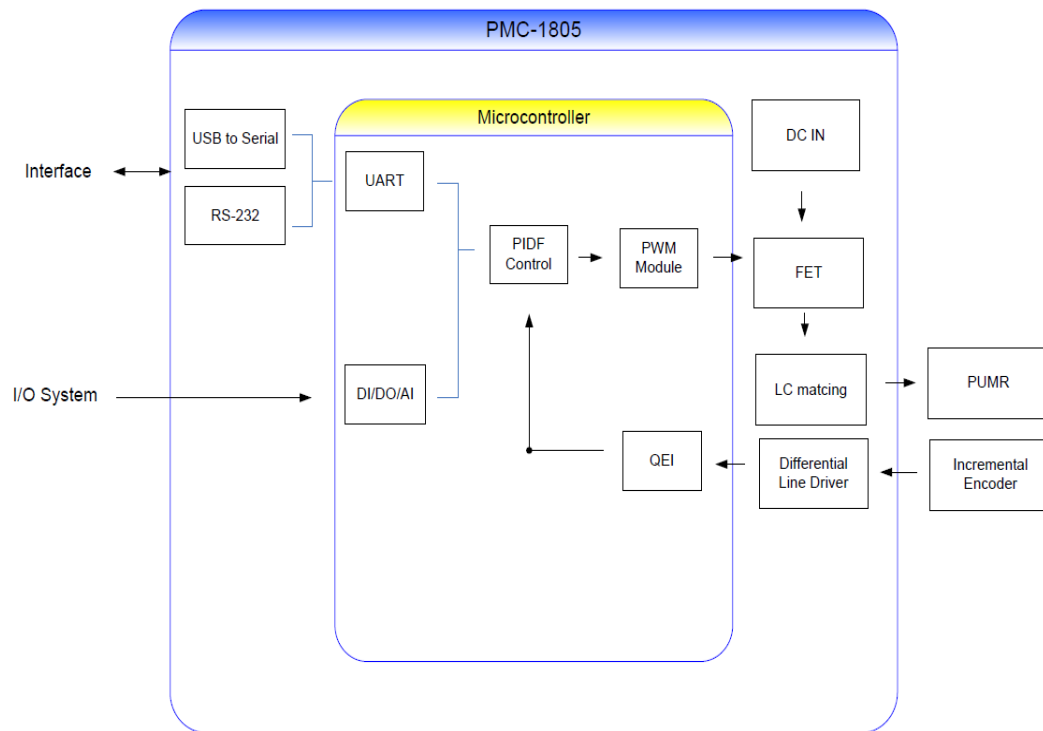
### 2.3 Serial communication specification

User can manipulate the “PMC1805” in the command level. Please refer to “chapter 5” for more information of the command-list. In case of PC as a upper-controller, please refer to APPENDIX 2.

The spec. of serial is as follows

- Baud-rate: 115200 bps
- Data bits: 8 bit
- Stop bit: 1
- Parity: None
- Flow-control: None

### 3. Block Diagram



## 4. Remote mode

In the remote-mode, "PMC1805" can be manipulated by the upper-controller like PC. Before using this mode, the virtual com-port driver of "CDM 2.08.28.exe" must be installed in your PC. "PMC1805" use the RS232C over USB chip of FT232R of FTDI. Please refer to Appendix 2.

In order to enter the remote-mode, please follow the next procedures.

Step1: Open the proper com-port in your program

Step2: Send ">remote\r" command to "PMC1805" : Refer to chapter 5

Step3: Send any commands you need as Step2

.....

Finally, to exit the remote-mode, then send ">local\r" command to external PCB



## 5. COMMAND

- All commands are lower-case.
- Using the plain "C" syntax, we are explaining the usage of a command.
- The return-values of command are always the ASCII character string.
- When running, sometimes, the informative character-strings are sent

### 5-1 Format

STX + command + sp + parameter + terminator

- STX :            '>'
- command:        low case character string
- sp:              spaces as separator(=0x20)
- parameter:      Numeric-string which is optional
- terminator:     '\r' that is, carriage return(= 0xd)

### 5-2. command list

All command must be response after send the command within 100ms.

The response sentence is "<ok" or "<ng".

## encoder

### Description

Set the Encoder type.

Format '>'+ "encoder"+parameter+'\r'

Parameter: 0 : Hall Sensor 1 : ABZ phase 2 : AB phase 3 : No Encoder  
4: ABZ used Line Driver

Example:

```
printf(">encoder 4\r");                    //set ABZ phase used Line Driver
```

## home

### Description

Move to Home position and set position to zero.

Format `'>'+ "home" + '\r'`

Example:

```
printf(">home\r") //
```

Return string:

## **moveto**

Description

Move to specified position by degree ( unit:0.001degree ) .

Format `'>'+ "moveto" + parameter + '\r'`

Parameter: range degree (-2147483648~2147483647)

Example:

```
printf(">moveto 90000\r") //move to the 90 degree
```

## **ontime**

Description

Set the running-time which is only applicable to "run" command

Format `'>' + "ontime" + parameter + '\r'`

Parameter: time in mili-second ( 0 ~ 2000ms)

## **pause**

Description

Set the pause-time between the operation which is only applicable to "run" command

Format `'>' + "pause" + parameter + '\r'`

Parameter: time in mili-second ( 0 ~ 2000ms)

Example:

```
printf(">pause 0\r"); // no pause-time
```

```
printf(">pause 1000\r");           // 1000ms pause between operations
```

## **posi**

### Description

Get the current position and speed

Format ' >' + "posi" + ' \r'

Example:

```
printf(">posi\r") //
```

Return value:

1st numeric character string is the current position by degree.

2nd numeric character string is the speed by RPM.

## **pulse**

### Description

Set the pulse count for 1 revolution.

Format ' >' + "pulse" + parameter + ' \r'

Parameter: The pulse count for 1 revolution (36~60000)

Example:

```
printf(">pulse 4000\r");           // set the pulse count to 4000
```

## **remote**

### Description

Set the remote-mode. the "local S/W", "knob of VR" are not functioned anymore.

Format ' >' + "remote" + ' \r'

Example:

```
printf(">remote\r");           //
```

## **reset**

### Description

Warm-start, it is same as the cold-start.

Format `'>'+ "reset" + '\r'`

Example:

```
printf(">reset\r") //
```

## **run**

### Description

Run USM continuously. The direction of USM alternates between CW and CCW during set ontime.

Format `'>' + "run" + '\r'`

Example:

```
printf(">run\r"); // run USM during on alternating direction during set ontime.
```

## **runccw**

### Description

Run USM in the direction of CCW. After expiring set the on-time, USM stop.

Format `'>' + "runccw" + '\r'`

Example:

```
printf(">runccw\r"); // run USM during set the ontime in the direction of
```

CCW

## **runcw**

### Description

Run USM in the direction of CW. After expiration of the on-time, USM stop.

Format '>' + "runcw" + '\r'

Example:

```
printf(">runcw\r"); // run USM during set the ontime in the direction of CW
```

## **save**

Description

Save the current condition to the internal flash memory permanently.

After "save", the rebooting or ">reset" command is needed. Normally there is no need to use this command in end-user side.

Format '>'+ "save" + '\r'

Example:

```
printf(">save\r") //
```

## **speed**

Description

Set the speed of USM by the unit of percent( unit:0.1% )

Format '>' + "speed" + parameter + '\r'

Parameter: percentage ( 0 ~ 999)

Example:

```
printf(">speed 999\r"); // set USM speed to the fastest speed
```

## status

### Description

Get the results of the previous command. The decimal-string which represents the alarm-status bit by bit must be returned.

Format '>' + "status" + '\r'

Example:

```
printf(">status\r") // read the status
```

Return string:

"<S hexadecimal-string"

The meanings of "hexadecimal string" are as follows.

Bit-position	Alarm Name	Description
8000	-	
4000	Command Err	1 = The command was improperly formatted
2000	Parameter Err	1 = The parameter is out of range
1000	-	
800	Motor Connect Err	1 = Motor cable Not connect
400	Temp warning	1 = Controller Temperature high ( >60°C )
200	-	
100	Encoder Err-	1 = Encoder ABZ input have some trouble
80	-	
40	-	
20	Position Err	1 = Encoder position is missing
10	-	
8	Home Err	1 = Home position is not known
4	Time Over	1 = Motor is not arrived to target position
2	-	
1	Motor Run	1 = Motor is running
0		Controller wait to CMD

## **stop**

### Description

Stop running immediately.

Format '>'+”stop”+' \r'

Example:

```
printf(">stop\r") //
```

## **clearstatus**

### Description

Clear the bit of Error Status.

Format '>'+”clearstatus”+' \r'

Example:

```
printf(">clearstatus\r") //
```

Return String

“<S hexadecimal-string”

## **statusadc**

### Description

Read of ADC1,2,3 value of J7 3,4,5

Format '>'+”statusadc”+' \r'

Example:

```
printf(">statusadc\r") //
```

Return string:

“<A decimal-string1 decimal-string2 decimal-string3”

Decimal-string 1 : ADC1

Decimal-string2 : ADC2

Decimal-string3 : ADC3

## **statuslimit**

### Description

Get the status limit of J5 5,6

Format '>'+ "statuslimit"+ ' \r'

Example:

```
printf(">statuslimit\r") //
```

Return string:

"<L decimal-string"

0 : Not limit position

1: CW Limit position

2: CCW Limit position

## **timeover**

### Description

Set the time of USM malfunction or stuck( unit:ms )

Format '>' + "timeover" + parameter + '\r'

Parameter: percentage ( 0 ~ 9999999)

Example:

```
printf(">timeover 500\r"); // set USM auto-stop time for some
```

trouble

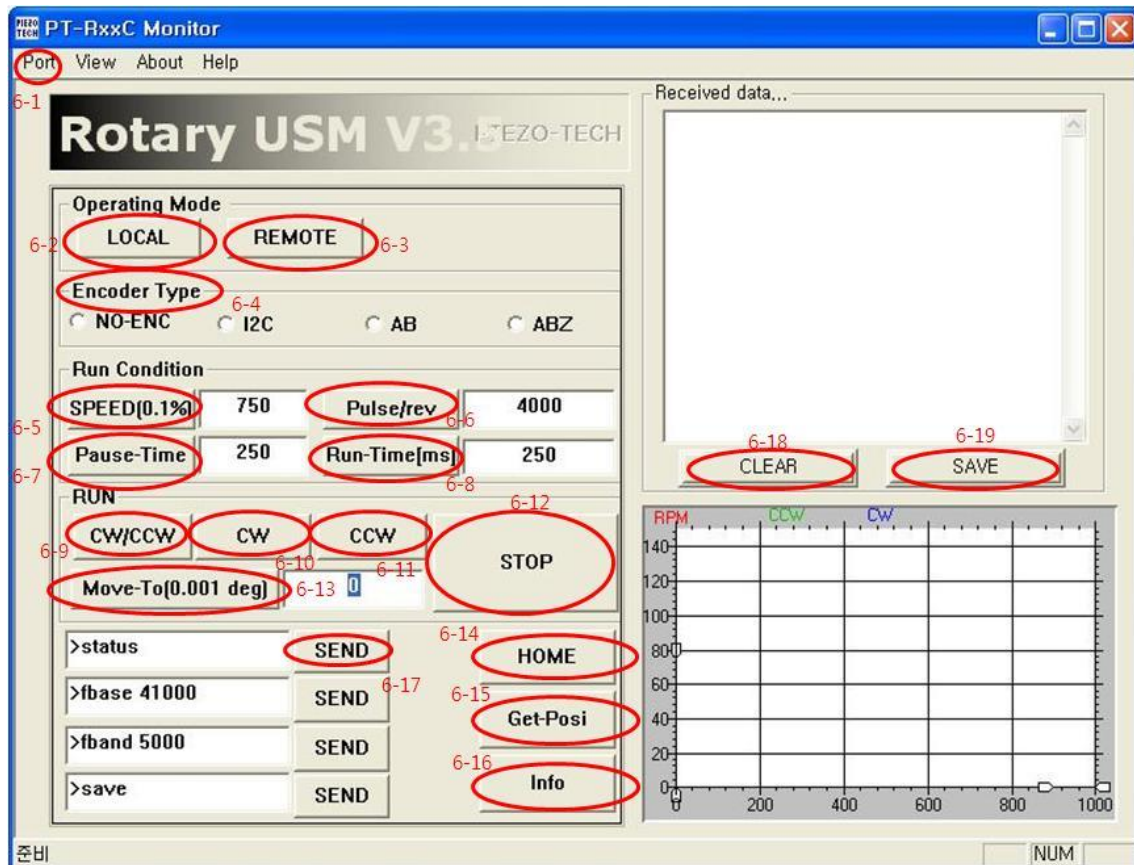


## 6. PC MONITOR software

In this chapter, we explain the usage of "PC MONITOR" program. Before executing, the virtual com-port driver and OCX must be installed. Refer to APPENDIX 2, 3

Figure 6-1 shows the UI window of "PT-Rx.exe".

Figure 6-1

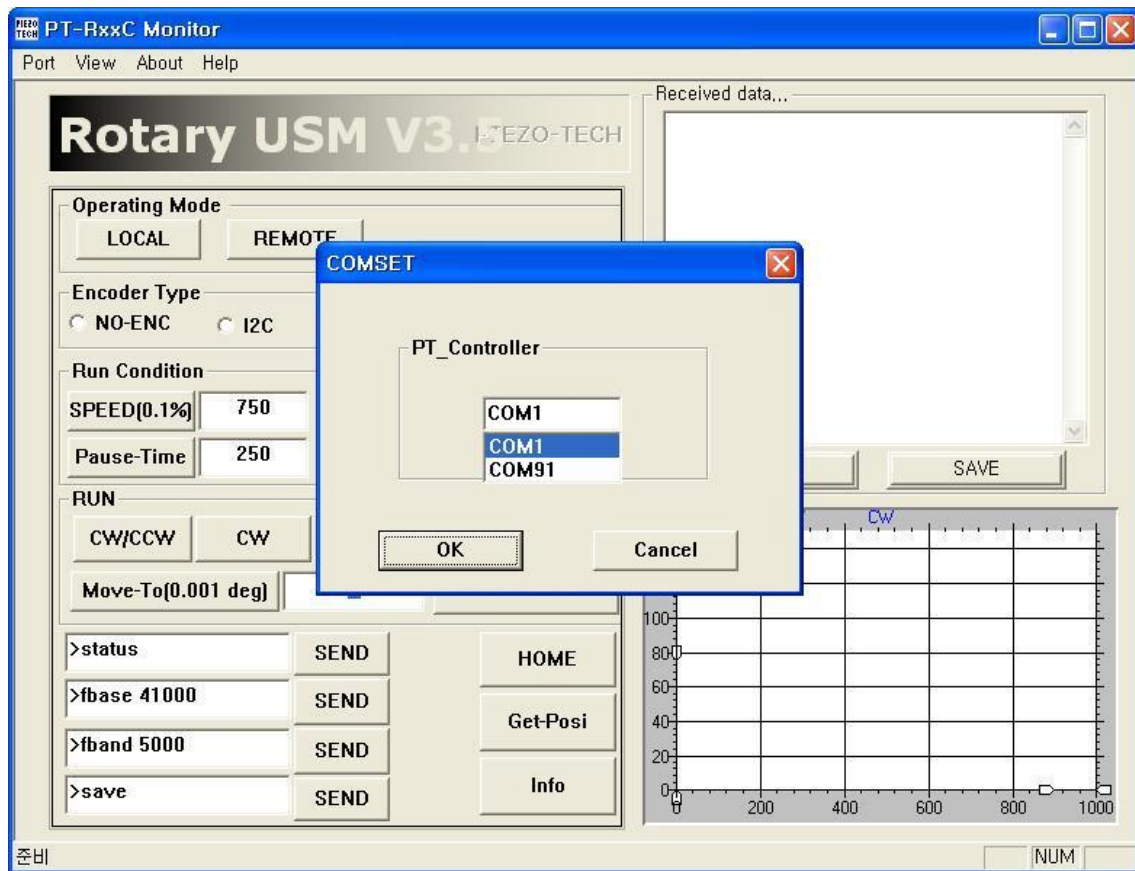


## 6-1 Menu button of [Port]

In order to connect PC with “PMC1805” over USB port of PC, at first, upon clicking the button named “Port” and select the proper COM-port. Please confirm “ok” message.

Figure 6-2 shows the window in question.

Figure 6-2



## 6-2 [Local] Button

This button set “PMC1805” to the local mode. In this mode, the SW, VR knob of “PMC1805” are available.

## 6-3 [Remote] Button

This button set “PMC1805” to the remote mode. In this mode, the SW, VR knob of “PMC1805” are not available. The upper-controller can control “PMC1805” through RS232C over USB in

command level.

#### 6-4 [Encoder Type] Button

This button is same as “encoder type” command in chapter 5.

#### 6-5 [SPEED] Button

This button is same as “speed” command in chapter 5.

#### 6-6 [Pulse/rev] Button

This button is same as “pulse” command in chapter 5.

#### 6-7 [Pause-Time] Button

This button is same as “pause” command in chapter 5.

#### 6-8 [Run-Time] Button

This button is same as parameter of “run”, “runccw”, “runccw” command in chapter 5.

#### 6-9 [CW/CCW] Button

This button is same as “run” command in chapter 5.

#### 6-10 [CW] Button

This button is same as “runccw” command in chapter 5.

#### 6-11 [CCW] Button

This button is same as “runccw” command in chapter 5.

#### 6-12 [Move - to [0.001 deg]] Button

This button is same as “moveto” command in chapter 5.

#### 6-13 [STOP] Button

This button stop running of PUMR40/60 USM

#### 6-14 [HOME] Button

This button is same as “home” command in chapter 5.

#### 6-15 [Get-posi] Button

This button is same as “posi” command in chapter 5.

#### 6-16 [Info] Button

This button is to get the information of “PMC1805” internal setting.

#### 6-17 [SEND] Button

Internal use only.

#### 6-18 [CLEAR] Button

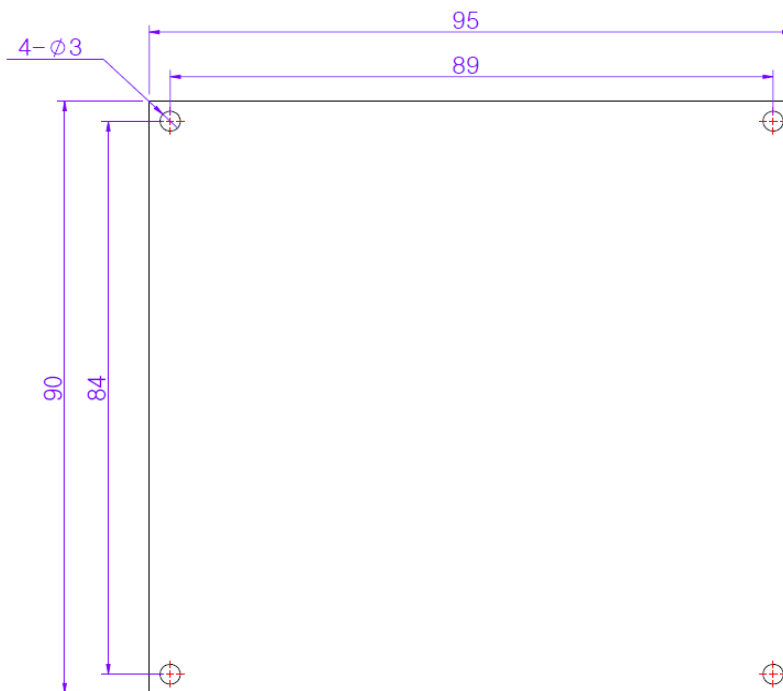
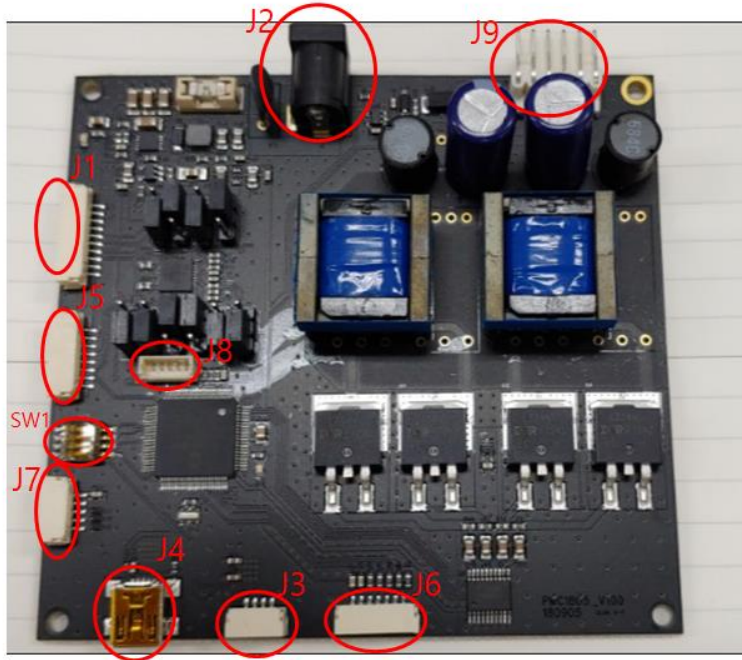
This button is to clear the information window.

#### 6-19 [SAVE] Button

This button is to save the contents of the information window.

# APPENDIX 1. PMC 1805

## Connector and Jumper (PMC-1805)

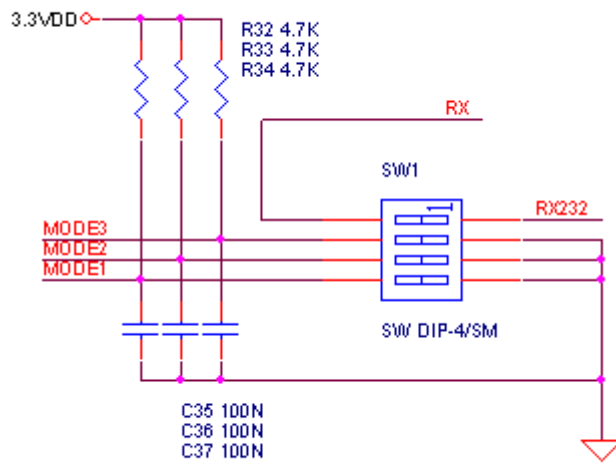


### J3 - RS232C port

The Figure A-1 shows J6 pin-map. In order to use J3, Pin NO.1 of SW1 must be jumped at proper position.

1	TXD
2	RXD
3	GND
4	Not implement

Figure A-1



### SW1 DIP switch

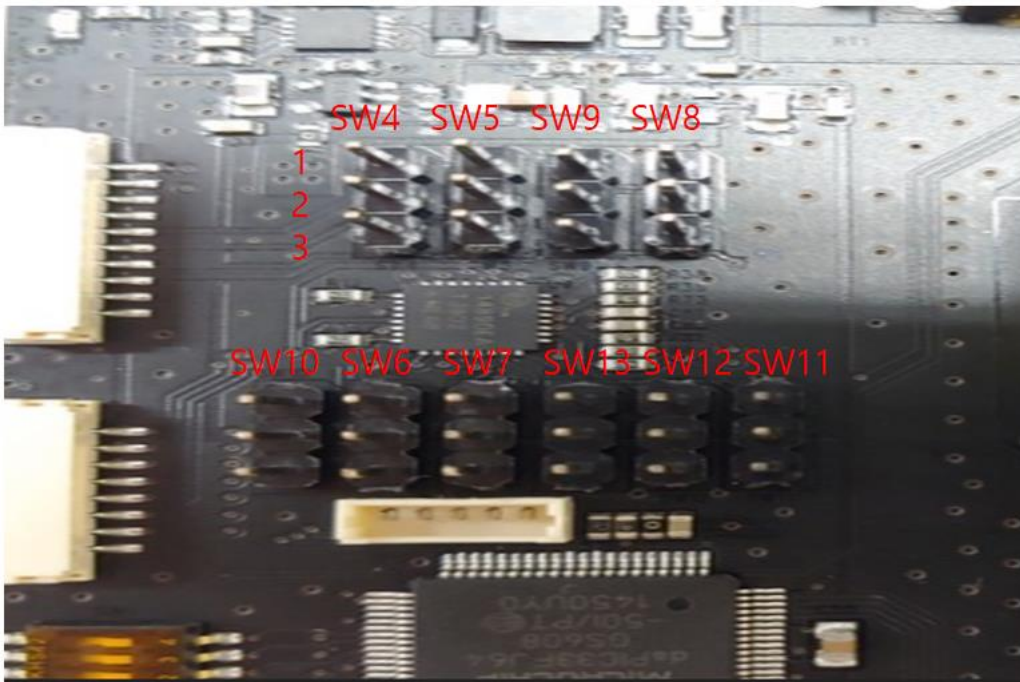
1	Select RS232 or USB Communication
2	Not implement
3	Not implement
4	Not implement

### J2 - DC Power Jack

PMC1805-01 requires DC 12V/2A and PMC1805-02 needs DC 24V/2A regulated.

#### J4 - USB Connector for RS232C communication

The virtual com-driver must be installed at your PC to communicate with "PMC1805".



#### SW 4, 5 Pin Header

Select the Jumper connector to use TTL or RS422 (refer J1 , J5 connector)

QEI_Encoder type	SW4	SW5
TTL	2-3	1-2
RS422	1-2	2-3

#### SW 6, 7 Pin Header

Select the Jumper connector to use TTL or RS422 (refer J1 , J5 connector)

QEI_Encoder type	SW6	SW7
TTL	2-3	1-2
RS422	1-2	2-3

### SW 8, 9 Pin Header

Select the Jumper connector to use TTL or RS422 (refer J1 , J5 connector)

QEI_Encoder type	SW8	SW9
TTL	2-3	1-2
RS422	1-2	2-3

### SW 10,11 Pin Header

Select the Jumper connector to use TTL or RS422 (refer J1 , J5 connector)

QEI_Encoder type	SW10	SW11
TTL	2-3	1-2
RS422	1-2	1-2

### SW 12,13 Pin Header

(Not implement ) must be differ SW12 than SW13

QEI_Encoder type	SW12	SW13
-	If ,2-3	Must be, 1-2
-	If, 1-2	Must be, 2-3

### J1. Encoder Connector

This port is option of PUMR. When using the encoder of RS422 .

Table A-2 describes the pin-map

1	VCC (DC 5V)
2	GND
3	QEI B+
4	QEI B-
5	QEI A+
6	QEI A-
7	QEI Z+
8	QEI Z-
9	GND
10	GND



## J5. Encoder Connector

This port is option of PUMR. When using the encoder of Single input Table A-2 describes the pin-map

Table A-2

1	VCC(DC 5V)
2	QEI A
3	QEI B
4	QEI Z
5	EXIO IN 1 (for CW limit )
6	EXIO IN 2 (for CCW limit)
7	-
8	GND

## J7. External Connector

Using the external ports of J7 User can control the “PMC1805” directly in the external analog and digital signal. Regarding the accurate position of jumper and connects, please refer to APPENDIX 1.

The “Figure 3-1” shows the pin map for the external analog signal. And the input-circuit of “PMC1805” side and Table 3-1 explains the pin assignment and functions.

Figure 3-1

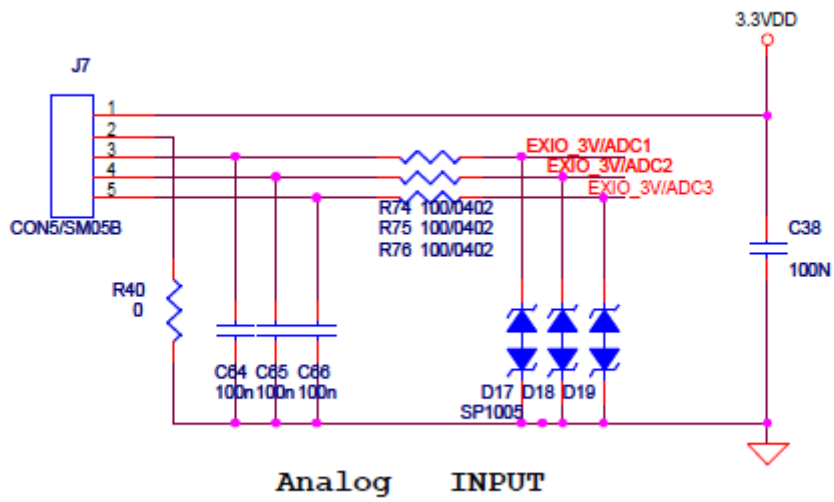


Table 3-1

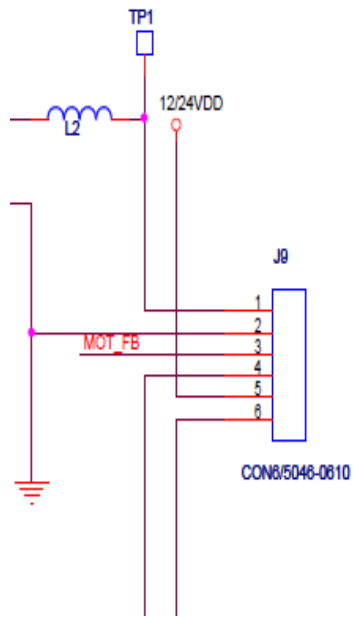
NO	Item	Function
1	3.3 VDD	3.3V Output Voltage
2	GND	Ground
3	EXIO1	External IO 1 (input level 3.3V)
4	EXIO2	External IO 2 (input level 3.3V)
5	EXIO3	External IO 3 (input level 3.3V)

## J9 MOTOR Connector

Figure A-4 shows "MINI DIN 6PIN" connector and Table A-2 describes the pin-map

Figure A-4

Table A-2

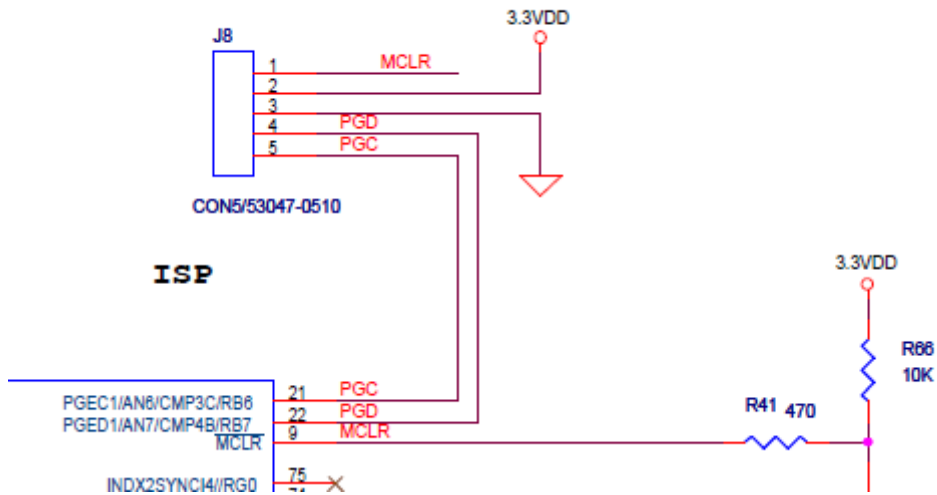


NO.	DESCRIPTION
1	Motor signal A (Cosine)
2	GND
3	Motor_Feedback signal(Not implement)
4	Motor signal B (Sine)
5	Connect Pin 6 of J9
6	Connect Pin 5 of J9

## J8. ISP Connector

This port is used to program MCU of “PMC1805”.

Figure A-5



## Inductor and Transformer

The differences between PMC1805-01 and PMC1805-02 are the transformer, inductor and the DC power input. Figure A-6 shows “PMC1805-01” and Figure A-7 shows “PMC1805-02” controller.

Figure A-6: PMC1805-01



Figure A-7: PMC1805-02



## APPENDIX 2.

### How to install virtual communication port

The virtual com-port driver must be installed named "CDM 2.08.28.exe" or latest version.

Website URL : <http://www.fdichip.com/drivers/vcp.htm>

## APPENDIX 3.

### How to Install OCX

#### 1. Windows XP

Step1: Unzip the enclosed "OCX.zip" in any directory

Step2: Run "install.bat"

#### 2. Windows 7, 8 32bit

Step1: Unzip the enclosed "OCX.zip" in any directory

Step2: Copy "StripS.ocx" to "C:\windows\system32

Step3: Run "install.bat" in administrator privileges mode which is selected using right-click of mouse

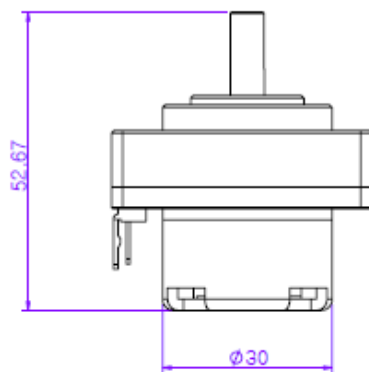
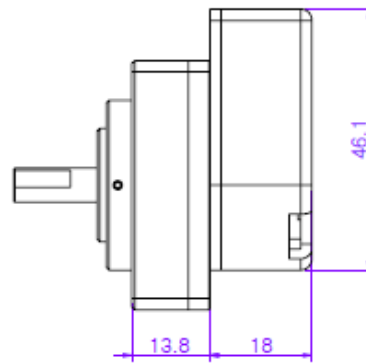
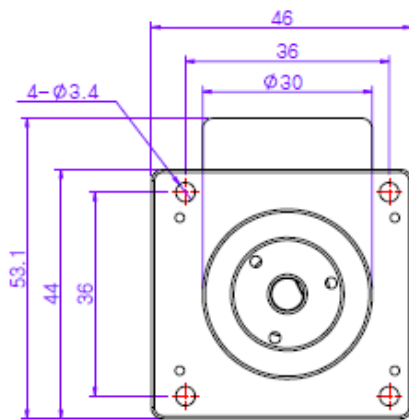
#### 3. Windows 7, 8 64bit

Step1: Unzip the enclosed "OCX.zip" in any directory

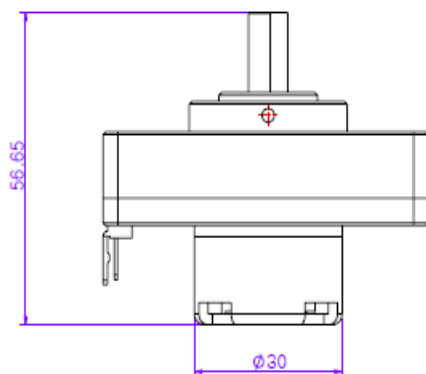
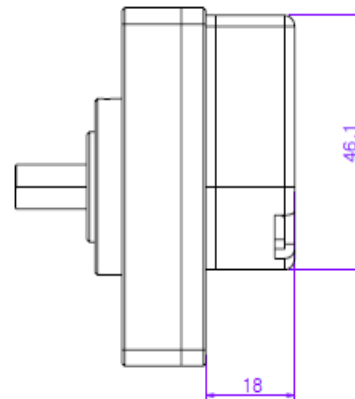
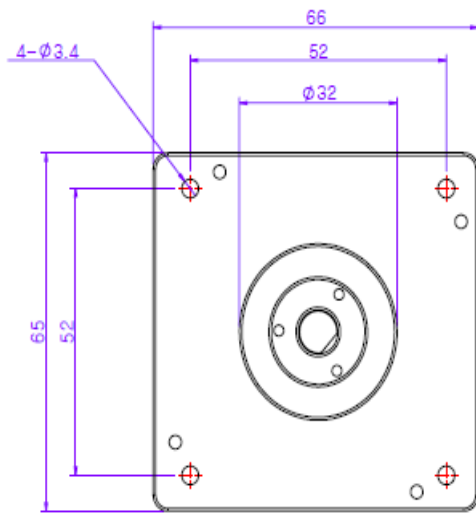
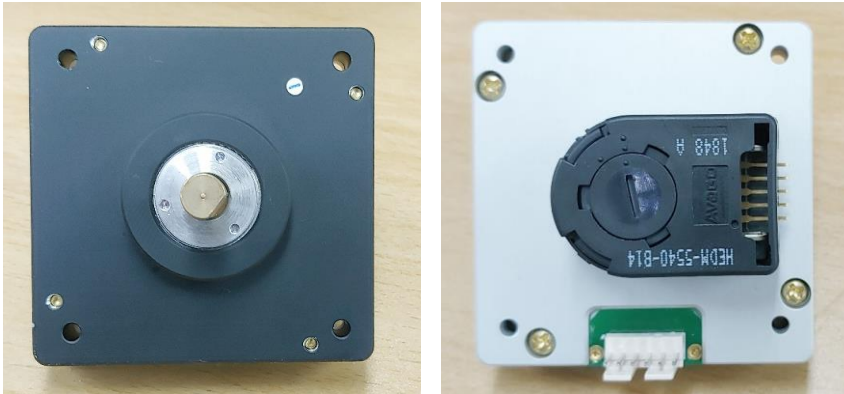
Step2: Copy "StripS.ocx" to "C:\windows\syswow64

Step3: Run "install.bat" in administrator privileges mode which is selected using right-click of mouse

## APPENDIX 4. PUMR40 dimension



## APPENDIX 5. PUMR60 dimension



## APPENDIX 6 . PUMR 40/60 specifications

MOTOR	PUMR40	PUMR60
Shaft diameter (mm)	Φ6	Φ8
Dimension (mm <sup>3</sup> )	53 × 44 × 53	66 × 65 × 56
Weight (g)	91	270
Operating frequency (kHz)	34 ~ 38	38 ~ 42
Operating voltage (Vrms)	120	120
Rated rotational speed (RPM)	60	70
Rated torque (kgf·cm)	0.8	2.5
No load speed (RPM)	≥120	≥120
Maximum torque (kgf·cm)	≥ 1.6	≥ 5
Holding torque (kgf·cm)	2.5	10.2
Response time (msec)	≤ 0.1	≤ 0.1
Encoder	4000 pulse	
Allowable temperature (°C)	(On surface of the motor) 55 °C	
Operating temperature (°C)	-10 ~ 50	-10 ~ 50
Storage temperature (°C)	-20 ~ 60	-20 ~ 60
Life time (hrs)	3,000	3,000



## Terms and Abbreviations

USM	Ultrasonic Motor
CW	Clockwise
CCW	Counter-clockwise
VR	Variable Resistor
PUMR40	Model name of USM which diameter is 40mm.
PUMR60	Model name of USM which diameter is 60mm.
PMC1805-01	Model name of USM controller for PUMR40
PMC1805-02	Model name of USM controller for PUMR60
PMC1805	Means PMC1805-01 and PMC1805-02 together.
CDM 2.08.28.exe	FTDI device driver

Web Site URL: <http://www.ftdichip.com>