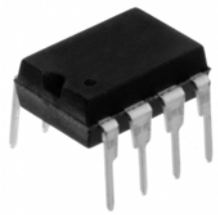
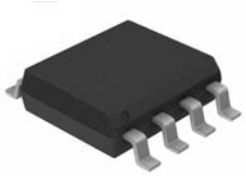
JP Serial Frequency and Counter Module



PDIP



SOIC

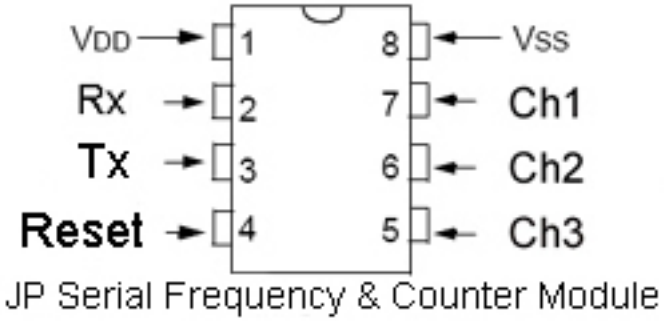
JP serial frequency and counter module is a simple and cost effective interface controller module. It will work with any microcontroller capable of accepting asynchronous serial data.

SPECS:

Power:	5V VDC
Number of channels:	3
Frequency:	From 1 Hz Up to 50,000Hz
Counter:	0 – 4294967295
I/O Voltage:	0V and 5V
Package:	PDIP (300 mil) and SOIC (3.9mm)
Baud Speed:	19200

PIN FUNCTIONS:

Pin1: Vdd
Pin2: Rx
Pin3: Tx
Pin4: Reset
Pin5: Ch3
Pin6: Ch2
Pin7: Ch1
Pin8: Vss



JP Serial Frequency & Counter Module

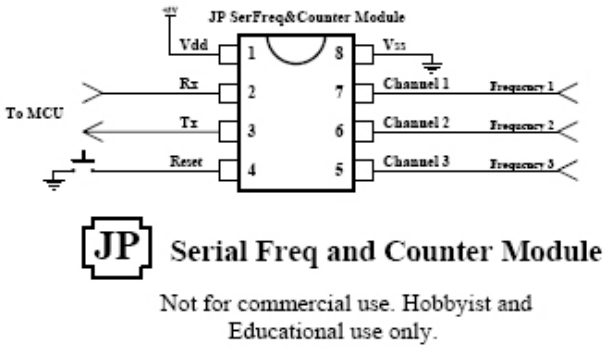
JP Serial Frequency and Counter Module Operate Commands:

(Its one byte command, data is ten bytes string)

Command(hex)	Command (Dec)	Description	Parameter
$30	48 (0)	Request Frequency from Ch1	None
$31	49 (1)	Request Frequency from Ch2	None
$32	50 (2)	Request Frequency from Ch3	None
$33	51 (3)	Request Counter number from Ch1	None
$34	52 (4)	Request Counter number from Ch2	None
$35	53 (5)	Request Counter number from Ch3	None
$36	54 (6)	Request Frequency (with zero) from Ch1	None
$37	55 (7)	Request Frequency (with zero) from Ch2	None
$38	56 (8)	Request Frequency (with zero) from Ch3	None
$39	57 9)	Request Counter number (with zero) from Ch1	None
$3A	58(:)	Request Counter number (with zero) from Ch2	None
$3B	59(;)	Request Counter number (with zero) from Ch3	None
$62	98 (b)	Calibration frequency +	None
$63	99 (c)	Calibration frequency -	None
$64	100(d)	Set Ch1 input positive edge	None
$65	101 (e)	Set Ch2 input positive edge	None
$66	102 (f)	Set Ch3 input positive edge	None
$67	103 (g)	Set Reset input positive edge	None
$68	104 (h)	Set Ch1 input negative edge	None
$69	105 (i)	Set Ch2 input negative edge	None
$6A	106 (j)	Set Ch3 input negative edge	None
$6B	107 (k)	Set Reset input negative edge	None

Note: With zero: 12345 = 0000012345. No zero: 0 = space

JP Serial Frequency and Counter Module Schematic



JP Serial Freq and Counter Module

Not for commercial use. Hobbyist and Educational use only.

Note:

When you send command and data to the module, it needs a pause to process them without missing the next command and data.

JP Serial Frequency and Counter Module integrated commands to a pic12F1822. User can use it very easy. Please read Pic12F1822 Datasheet first when you want to buy it.(www.microchip.com)

LIABILITY WARNING

This device should be used only for experimental purposes. It has NOT gone through extensive testing and it could erase or corrupt some or all data on media cards that are inside the device. You assume to take your own risk when you purchase this device, and release the responsibility and liability from the manufacturer with no harm.

REGULATORY WARNING

This device is intended solely for experimental purpose; it is not in finished product form and is NOT FCC approved. If you wish to install these modules into non-experimental final finished products, you will be responsible to have the modules approved by the FCC at your own cost.

JP Serial Frequency and Counter Module Test Code:

```
/*============================================================
 File........JP Serial Frequency & Counter Module Test code for Arduino
 Purpose.....This test code for JP Serial Frequency & Counter Module
 MCU.........Arduino Mega 2560
 Author......Jianping Sun
 Website.....www.JianpingUSA.com
 Email.......JP@JianpingUSA.com
 Started.....July, 7, 2012
 Updated.....
 Note: The Freq & Counter module Rx <===> Tx1 of Arduino
       The Freq & Counter module Tx <===> Rx1 of Arduino
       The SerLCD module Rx <===> Tx0 of Arduino
============================================================*/
int x = 0;    // variable
int i = 0;
int j = 0;
int Ch_Line;

int F_Data[10];  // Frequency Data Variable

void setup()
{
  Serial.begin(4800);   // For JP SerLCD Module
```

```
  Serial1.begin(19200);  // For JP Serial Frequency & Counter Module

  delay(1000);
  First_Screen();
  delay(1000);
  //Three_Phase_Freq();
  Three_Phase_Counter();
  delay(1000);
}
void loop(){
//x = 48;           // First Channel Frequency (no zero)
//x = 54;           // First Channel Frequency (with zero)
//x = 51;           // First Channel counter (no zero)
x = 57;           // First Channel counter (with zero)
Req_Data();       // Request Frequency Data
delay(10);
Rec_Data();       // Receive Frequency Data
delay(10);
Ch_Line = 196;
//Display_Freq_Data();    // Display Frequency Data
Display_Counter_Data();    // Display counter Data
delay(10);
//x = 49;           // Second Channel Frequency (no zero)
//x = 55;           // Second Channel Frequency (with zero)
//x = 52;           // Second Channel counter (no zero)
x = 58;           // Second Channel counter (with zero)
Req_Data();       // Request Frequency Data
delay(10);
Rec_Data();       // Receive Frequency Data
delay(10);
Ch_Line = 148;
//Display_Freq_Data();    // Display Frequency Data
Display_Counter_Data();    // Display counter Data
delay(10);
//x = 50;           // Third Channel Frequency (no zero)
//x = 56;           // Third Channel Frequency (with zero)
//x = 53;           // Third Channel counter (no zero)
x = 59;           // Third Channel counter (with zero)
Req_Data();       // Request Frequency Data
delay(10);
Rec_Data();       // Receive Frequency Data
delay(10);
Ch_Line = 212;
//Display_Freq_Data();    // Display Frequency Data
Display_Counter_Data();    // Display counter Data
delay(10);
```

```
}

//==================== Channel 1 servo Test =====================
void First_Screen() {

/*============================================================
                First Screen Test

============================================================*/
  Serial.write(17);              // Clear screen
  Serial.write(18);              // Move cursor to home
  delay(10);
  Serial.print("  JianpingUSA  ");
  Serial.write(192);             // 128+64,Move sursor to 2nd row
  delay(10);
  Serial.print("JP Ser FREQNENCY");
  Serial.write(144);             // 128+16,Move sursor to 3rd row
  delay(10);
  Serial.print("   and COUNTER  ");
  Serial.write(208);             // 128+80,Move sursor to 4th row
  delay(10);
  Serial.print(" Module Testing ");
  delay(1000);
}
//============================================================
=
void Three_Phase_Freq()  {
  Serial.write(17);              // Clear screen
  Serial.write(18);              // Move cursor to home
  delay(10);
  Serial.print("  3 Phases FREQ  ");
  Serial.write(192);             // 128+64,Move sursor to 2nd row
  delay(10);
  Serial.print("P1=          Hz");
  Serial.write(144);             // 128+16,Move sursor to 3rd row
  delay(10);
  Serial.print("P2=          Hz");
  Serial.write(208);             // 128+80,Move sursor to 4th row
  delay(10);
  Serial.print("P3=          Hz");
  }
void  Three_Phase_Counter()  {
  Serial.write(17);              // Clear screen
  Serial.write(18);              // Move cursor to home
  delay(10);
  Serial.print("3 Phases Counter");
```

```
  Serial.write(192);              // 128+64,Move sursor to 2nd row
  delay(10);
  Serial.print("C1=");
  Serial.write(144);              // 128+16,Move sursor to 3rd row
  delay(10);
  Serial.print("C2=");
  Serial.write(208);              // 128+80,Move sursor to 4th row
  delay(10);
  Serial.print("C3=");
  }
//=================== Request Frequency Data ======================
void Req_Data()
{
Serial1.write(x);
}
//=================== Receive Frequency Data ======================
void Rec_Data()   {
  j = 0;
while ( j < 10) {
     if (Serial1.available() > 0) {
           // get incoming byte:
        F_Data[j] = Serial1.read();
        j++;
     }
  }
}
//================ Dispaly Frequency Data (6 digi) ================
void Display_Freq_Data() {
  for (int i = 4; i < 10; i++){
  Serial.write(Ch_Line + i - 4);
  Serial.write(F_Data[i]);
  delay(10);
  }
}
//=============== Dispaly Counter Data (10 digi) ==================
void Display_Counter_Data() {
  for (int i = 0; i < 10; i++){
  Serial.write(Ch_Line + i);
  Serial.write(F_Data[i]);
  delay(10);
  }
}
```

Jianping Electronics
jp@jianpingusa.com
www.jianpingusa.com