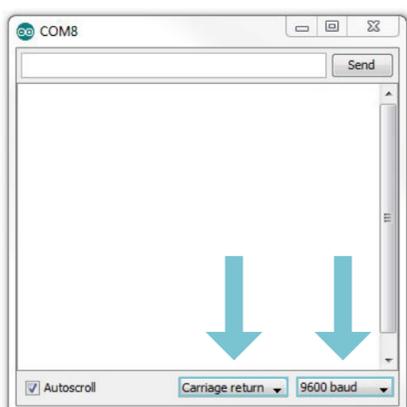
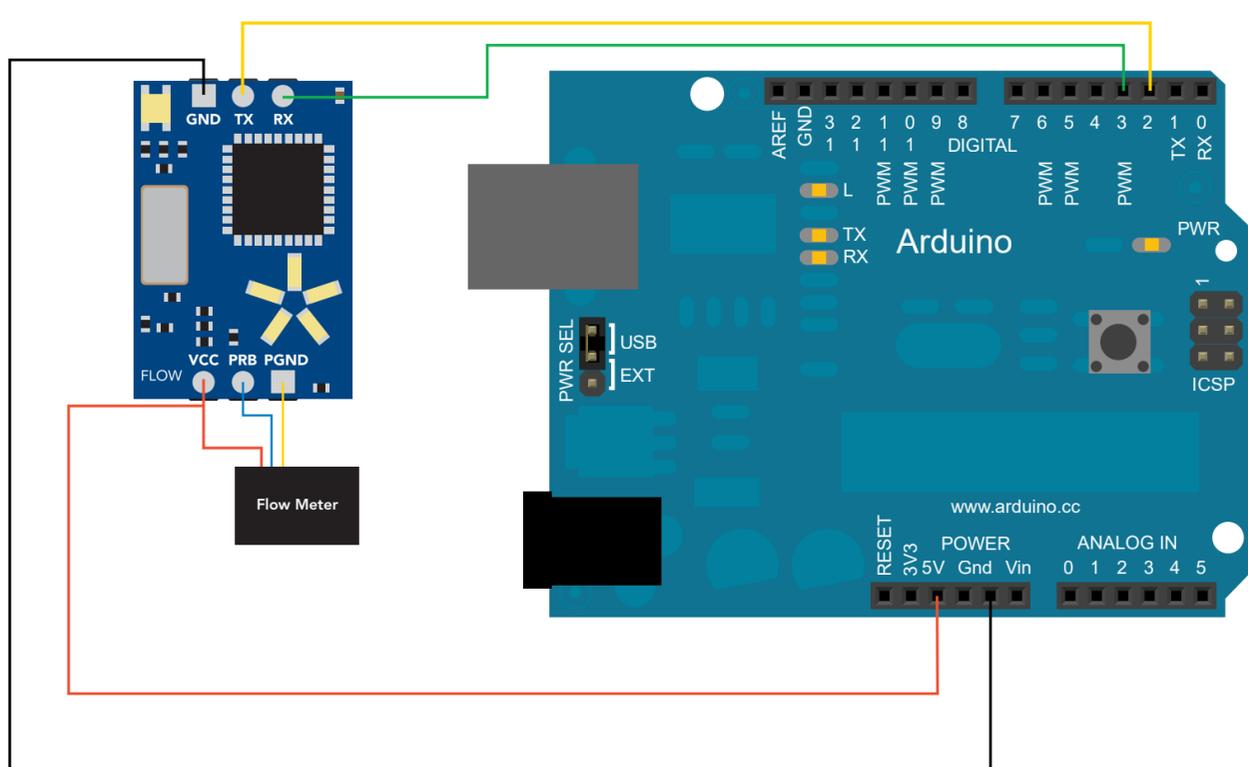


# Arduino Sample Code

Revised 6/3/15



*//This code has intentionally has been written to be overly lengthy and includes unnecessary steps.  
 //Many parts of this code can be truncated. This code was written to be easy to understand.  
 //Code efficiency can not be considered. Modify this code as you see fit.  
 //This code will output data to the Arduino serial monitor.  
 //Type commands into the Arduino serial monitor to control the Flow Meter Totalizer.*



```
#include <SoftwareSerial.h>           //we have to include the SoftwareSerial library, or else we can't use it.
#define rx 2                          //define what pin rx is going to be.
#define tx 3                          //define what pin tx is going to be.

SoftwareSerial myserial(rx, tx);      //define how the soft serial port is going to work.

char flow_data[48];                  //we make a 48 byte character array to hold incoming data from the Flow Meter Totalizer.
char computerdata[20];               //we make a 20 byte character array to hold incoming data from a pc/mac/other.
byte received_from_computer=0;       //we need to know how many characters have been received.
byte received_from_sensor=0;         //we need to know how many characters have been received.
byte string_received=0;              //used to identify when we have received a string from the Flow Meter Totalizer.

float total_flow=0;                  //used to hold a floating point number that is the total volume flow.
float flow_per_time ;                //used to hold a floating point number that is the flow rate per X time [hour, min, sec]

char *total;                          //char pointer used in string parsing
char *FPT;                             //char pointer used in string parsing [FPT= flow per time]

void setup(){
  Serial.begin(9600);                 //enable the hardware serial port
  myserial.begin(9600);               //enable the hardware serial port
}

void serialEvent(){
  received_from_computer=Serial.readBytesUntil(13,computerdata,20); //this interrupt will trigger when the data coming from
  computerdata[received_from_computer]=0; //the serial monitor(pc/mac/other) is received.
  myserial.print(computerdata); //we read the data sent from the serial monitor
  myserial.print("\r"); //we add a 0 to the spot in the array just after the last
  //character we received. This will stop us from transmitting
  //incorrect data that may have been left in the buffer.
  //we transmit the data received from the serial monitor
  //(pc/mac/other) through the soft serial port to the Flow Meter Totalizer.
  //all data sent to the Flow Meter Totalizer must end with a <CR>.
}

void loop(){
  if(myserial.available() > 0){
    received_from_sensor=myserial.readBytesUntil(13,flow_data,48); //if we see that the Flow Meter Totalizer has sent a character.
    flow_data[received_from_sensor]=0; //we read the data sent from Flow Meter Totalizer until we see a <CR>.
    //We also count how many character have been received.
    //we add a 0 to the spot in the array just after the last
    //character we received. This will stop us from transmitting incorrect data
    //that may have been left in the buffer.
    //if ec_data[0] is a digit and not a letter

    if((flow_data[0] >= 48) && (flow_data[0] <=57)){
      pars_data();
    }
    else
      Serial.println(flow_data); //if the data from the Flow Meter Totalizer does not start with a number
  } //transmit that data to the serial monitor.//transmit that data to the
  //serial monitor.
}

void pars_data(){
  total=strtok(flow_data, ","); //let's pars the string at each comma.
  FPT=strtok(NULL, ","); //let's pars the string at each comma.

  Serial.print("total_flow="); //We now print each value we parsed separately.
  Serial.println(total); //this is the total_flow.

  Serial.print("FPT="); //We now print each value we parsed separately.
  Serial.println(FPT); //this is the the flow rate per X time.

  Serial.println(); //this just makes the output easier to read.
}

//here are some functions you might find useful
//these functions are not enabled
/*

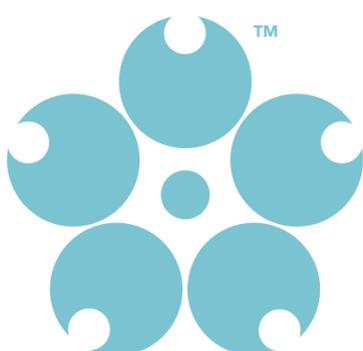
void flowFactoryDefault(){ myserial.print("X\r");} //factory defaults the Flow Meter Totalizer
myserial.print("factory\r"); //send the "factory" command to factory reset the device

void read_info(){ //get device info
  myserial.print("I\r"); //send the "I" command to query the information

void sleep(){ //send the "sleep" command to put theFlow Meter Totalizer in a low power state
  myserial.print("sleep\r");}

void flowSetLEDs(byte enabled) //turn the LEDs on or off
{
  if(enabled) //if enabled is > 0
    myserial.print("L,1\r"); //the LED's will turn ON
  else //if enabled is 0
    myserial.print("L,0\r"); //the LED's will turn OFF
}

*/
```



[Click here to download the \\*.ino file](#)