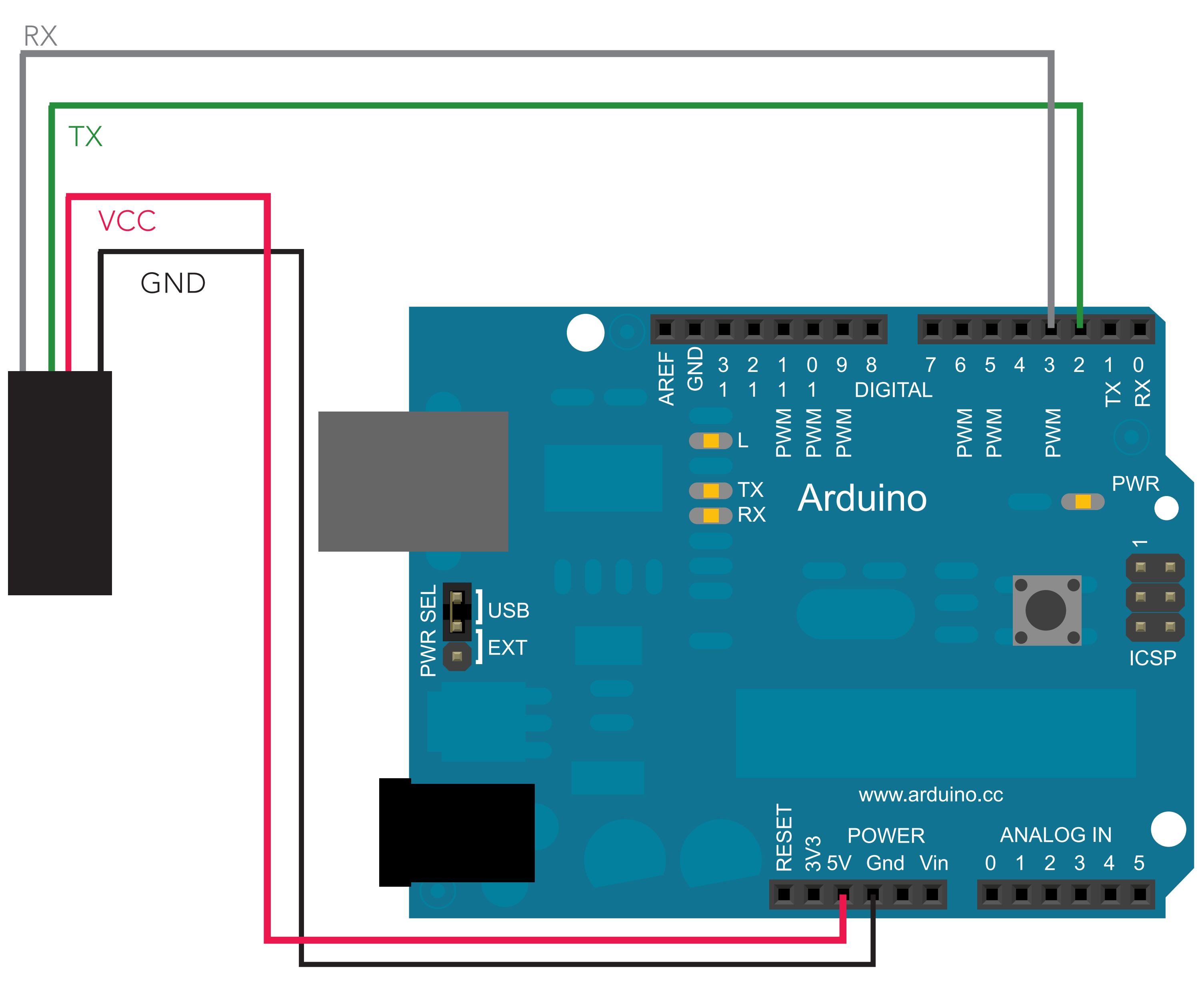
AtlasScientific
Biology•Technology

ENV-TMP-D
Arduino Sample Code

```
//This code has intentionally has been written to be overly lengthy and includes unnecessary steps.
//Many parts of this code can be truncated. Easy of understanding was the primary focus of this code.
//Code efficiency was not considered. Modify this code as you see fit.
//This code will output data to the arduino serial monitor.
//set the var arduino_only to =1 to watch the Arduino take over control of the ENV-TMP-D.
```



```
#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 tmp_data[20];                          //we make a 20 byte character array to hold incoming data from the ENV-TMP-D.
char computerdata[20];                      //we make a 20 byte character array to hold incoming data from a pc/mac/other.
byte pc_debug=1;                            //if you would like to debug the ENV-TMP-D through the serial monitor(pc/mac/other).
                                            //if not set this to 0.
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 arduino_only=0;                        //if you would like to operate the ENV-TMP-D with the arduino only and not use the
                                            //serial monitor to send it commands set this to 1. The data will still come out on the
                                            //serial monitor, so you can see it working.
byte startup=0;                             //used to make sure the arduino takes over control of the ENV-TMP-D properly.
float float_tmp=0;                          //used to hold a floating point number that is the ENV-TMP-D.
byte string_received=0;                     //used to identify when we have received a string from the ENV-TMP-D.


void setup(){
     Serial.begin(38400);                   //enable the hardware serial port
     myserial.begin(38400);                 //enable the hardware serial port
      }


void serialEvent(){                                                    //this interrupt will trigger when the data coming from
                                                                       //the serial monitor(pc/mac/other) is received.
       if(pc_debug==1){                                                //if pc debug is set to 0 this function will be bypassed.
       received_from_computer=Serial.readBytesUntil(13,computerdata,20); //we read the data sent from the serial monitor
                                                                       //(pc/mac/other) until we see a <CR>.
                                                                       //We also count how many characters have been received.
       computerdata[received_from_computer]=0;                         //we add a 0 to the spot in the array just after the last
                                                                       //character we recived. This will stop us from transmiting
                                                                       //incorrect data that may have been left in the buffer.
       myserial.print(computerdata);                                   //we transmit the data received from the serial monitor
                                                                       //(pc/mac/other) through the soft serial port to the ENV-TMP-D.
       myserial.print('\r');                                            //all data sent to the ENV-TMP-D must end with a <CR>.
      }
  }


void loop(){

  if(myserial.available() > 0){                                 //if we see that the ENV-TMP-D has sent a character.
    received_from_sensor=myserial.readBytesUntil(13,tmp_data,20); //we read the data sent from ENV-TMP-D untill we see a <CR>.
                                                                //We also count how many character have been recived.
    tmp_data[received_from_sensor]=0;                           //we add a 0 to the spot in the array just after the last character
                                                                //we recived. This will stop us from transmiting incorrect data that
                                                                //may have been left in the buffer.
    string_received=1;                                          //a flag used when the arduino is controlling the ENV-TMP-D
                                                                //to let us know that a complete string has been received.
    Serial.println(tmp_data);                                   //lets transmit that data received to the serial monitor.
    }


  if(arduino_only==1){                      //if you set arduino_only to = 1.
    if (startup==0){                        //if the arduino just booted up, we need to set some things up first.
       pc_debug=0;                          //make sure pc_debug is set to 0. You will no longer be able to write commands
                                            //to the ENV-TMP-D in the serial monitor.
        myserial.print("e\r");              //take the ENV-TMP-D out of continues mode.
        delay(50);                          //on start up sometimes the first command is missed.
        myserial.print("e\r");              //so, let’s send it twice.
        delay(50);                          //a short delay after the ENV-TMP-D was taken out of continues mode is used to make
                                            //sure we don’t over load it with commands.
        startup=1;                          //startup is completed, let's not do this again during normal operation.
     }


  delay(800);                                //we will take a reading ever 800ms. You can make this longer or shorter if you like.
   myserial.print("R\r");                    //send it the command to take a single reading.
   if(string_received==1){                   //did we get data back from the ENV-TMP-D?
    float_tmp=atof(tmp_data);                //many people ask us "how do I convert a sting into a float?" this is how...(pretty tough)
    if(float_tmp>=25){Serial.println("high\r");} //This is the proof that it has been converted into a string.
    if(float_tmp<25){Serial.println("low\r");}   //This is the proof that it has been converted into a string.
    string_received=0;}                      //reset the string received flag
  }

}


/*
here are some functions you might find useful

//set the tmp scale to Kelvin
void s_kel(){

// send the "sk" command to set the tmp scale to Kelvin
mySerial.print("sk\r"); }

//set the tmp scale to Fahrenheit
void s_far(){

// send the "sf" command to set the tmp scale to Fahrenheit
mySerial.print("sf\r"); }

// set the tmp scale to Celsius
void s_cel(){

// send the "sc" command to set the tmp scale to Celsius
mySerial.print("sc\r"); }

// enable the data logger Q 1 min.
void datta_legger_en(q)  //where q is an int (in this case set to 6)
{
mySerial.print("d6\r");
}

// send the "I" command to query the information
mySerial.print("I\r");

*/
```

Click here to download the *.ino file

Atlas-Scientific.com