5-26-19

Moisture Analyzer Manual

V1.00

Table of Contents

ABEtronics engineering services

Project Description:

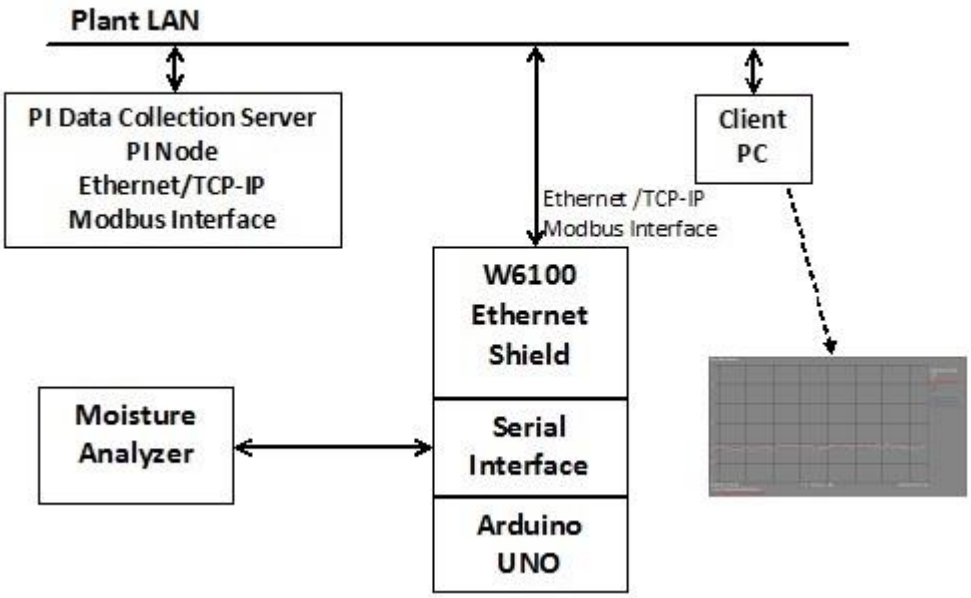
This project will interface with a moisture analyzer and a data collection system located on our plant local area network.

A moisture analyzer is used to measure % moisture content in a product. Previously, the readings were manually recorded by an operator after the measurement was complete. These measurements are critical to the process stability and performance.

The interface is built around the W6100 ethernet shield which allows communication over our plant local area network. This allows the % moisture readings to be historized on a PI data collection system. The PI data collection system node (ethernet TCP-IP Modbus interface) is used to communicate with the W6100 ethernet shield .

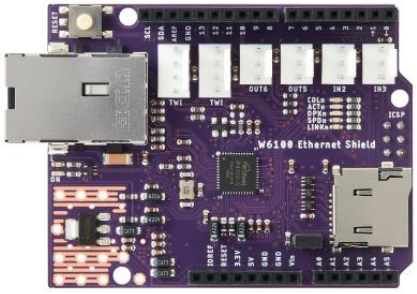
The data can now be easily analyzed by engineers for trending and making calculations on the data. The data is now useful for making improvements to the process and troubleshooting process problems.

Block Diagram:

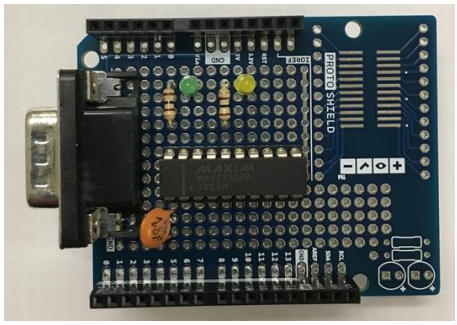


System Components:

W6100 Ethernet Shield:



Arduino UNO and Serial Interface Board:



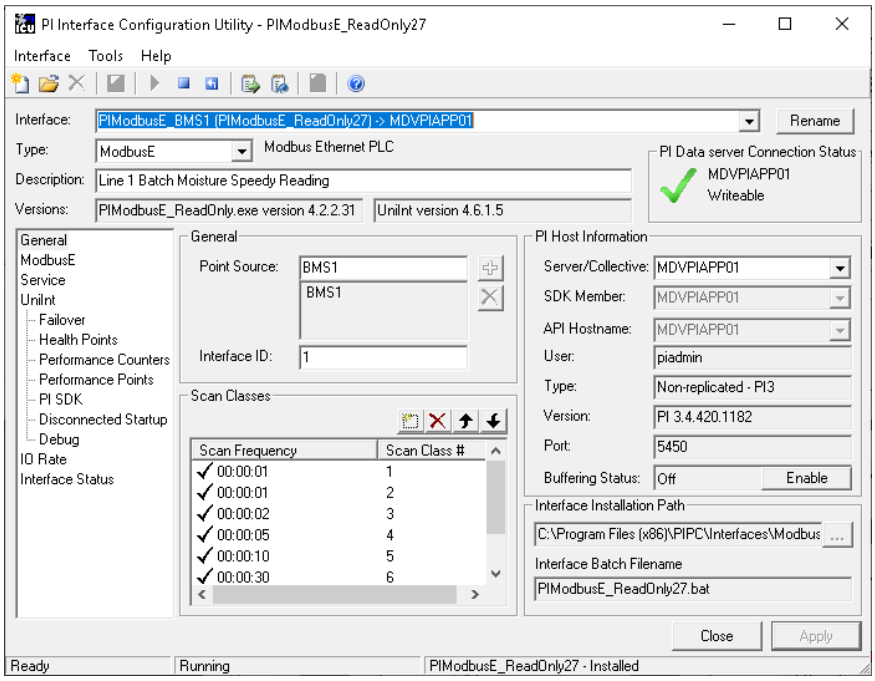
MF-50 Moisture Analyzer:

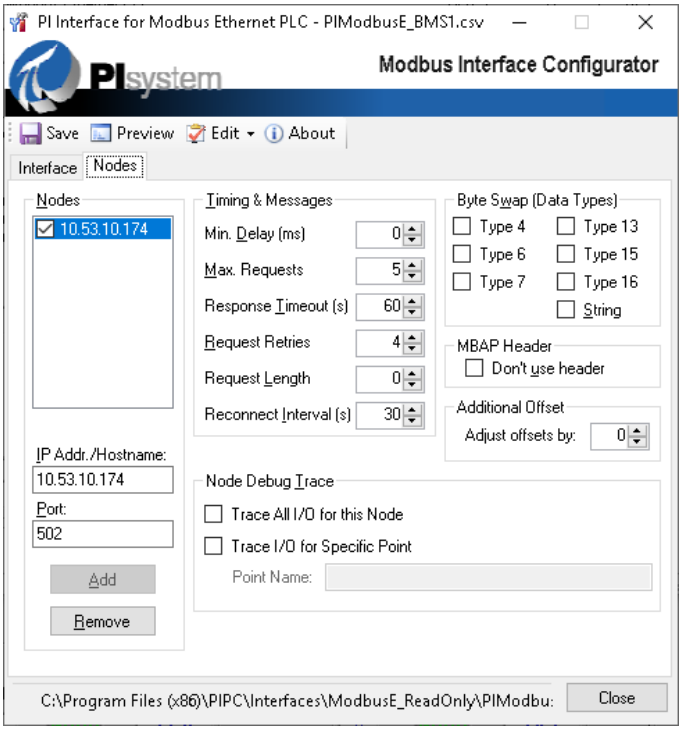
Ethernet TCP-IP/Modbus interface software: PI Node

Interface set-up: (PI Interface Configuration Utility (ICU).

Enter point source, scan classes, IP address.

Set-up as a service.





Ethernet TCP-IP/Modbus interface software: Arduino UNO/ W6100 ethernet shield

Set-Up Ethernet interface:

```
#include <SPI.h>
#include <Ethernet.h>

//--------------------------------------------------------------------------
// Ethernet settings (depending on MAC and Local network)
//--------------------------------------------------------------------------
  byte mac[] = {0x90, 0xA2, 0xDA, 0x0E, 0x94, 0xB5 };
  IPAddress ip(10,53,10,166);
  IPAddress myDns(10,60,11,130);
  IPAddress gateway(10,53,11,254);
  IPAddress subnet(255,255,254,0);

//-----------------------------------------------------------------
// initialize the ethernet device
//-----------------------------------------------------------------
 Ethernet.begin(mac, ip, myDns, gateway, subnet);   // start ethernet interface
```

Set-Up Modbus interface:

```
#include "MgsModbus.h"  // modbus TCP IP interface
//--------------------------------------------------------------------------
MgsModbus Mb;
//--------------------------------------------------------------------------
//--------------------------------------------------------------------------
// Moisture data to send to PI.
//--------------------------------------------------------------------------
 void Send_Data_to_PI()
 {
   bmi00 = inByte1.toInt();  // int part
   bmf00 = inByte2.toInt();  // fractional part
   Mb.MbData[0] = (bmi00*100)+bmf00;  // make an integer to send, /100 in PI.
   Mb.MbsRun(); // need to run as often as possible...otherwise get i/o timeout.
 }
```

Serial interface software: Moisture Analyzer

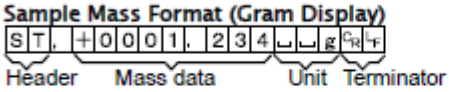
10.2. Output Format

In Case of Format omitted Temperature Data (Function Table 5-d 0)

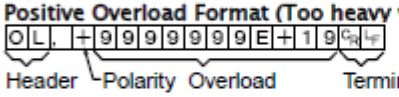
- The format consists of fifteen characters except the terminator.
- A polarity sign is placed before the data with the leading zeros. If the data is zero, the plus sign is used.
- The unit is ␣␣g or ␣␣%.
- The position of decimal point and minimum display are changed by models.
- Sign of ASCII code

CR	0Dh	Carriage return
LF	0Ah	Line feed
␣	20h	Space

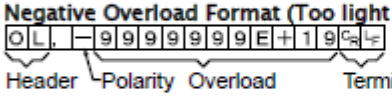
Sample Mass Format (Gram Display)



Positive Overload Format (Too heavy weighing, E display)

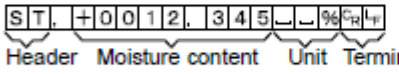


Negative Overload Format (Too light weighing, -E display)

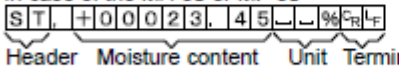


Moisture Content (during weighing or after weighing)

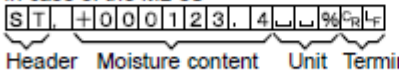
In case of the MS-70



In case of the MX-50 or MF-50



In case of the ML-50



Serial interface software: Moisture Analyzer

10.3. Command

- The analyzer can be controlled by the following commands from the computer.
  Add a terminator CR LF (0Dh、0Ah) to each command.

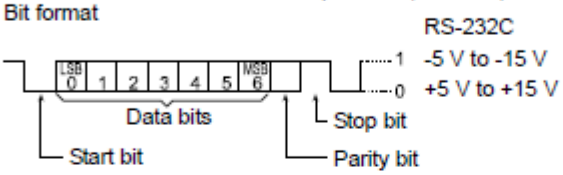
Command	Description
Q	Outputs the current data.
SIR	Outputs data continuously
C	Stops data output by SIR command.
QM	Outputs the data during measurement. (In other mode, QM can not use.)
START	Same as the START key
STOP	Same as the STOP key
RESET	Same as the RESET key
ENTER	Same as the ENTER key
SELECT	Same as the SELECT key
DOWN	Same as the ↓ key
UP	Same as the ↑ key
PROGRAM	Same as the PROGRAM key

Serial interface hardware: Moisture Analyzer

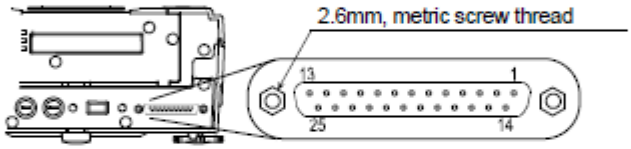
10.1. RS-232C Serial Interface

RS-232C Serial Interface

- Transmission system: EIA RS-232C
- Transmission form: Asynchronous, bi-directional, half duplex
- Data format: Baud rate 2400bps
  - Data bits: 7bits
  - Parity: EVEN
  - Stop bit: 1bit
  - Code: ASCII
  - Terminator: CR LF (CR: 0Dh, LF: 0Ah)
  - Bit format



Pin Connections



	MX-50 and MF-50 (DCE)		Direction	Computer (DTE)
Pin No.	Signal Name *2	Description		Signal Name
1	FG	Frame ground	–	FG
2	RXD	Receive data	←	TXD
3	TXD	Transmit data	→	RXD
4	RTS	Ready to send *3	←	RTS
5	CTS	Clear to send *3	→	CTS
6	DSR	Data set ready	→	DSR
7	SG	Signal ground	–	SG
16, 18, 19, 21, 23	Internal use		Do not connect *1	
Other	Not used			

*1: Normal DOS/V cables do not use these terminals.
*2: Signal names of the analyzer side are the same as the DTE side with TXD and RXD reversed.
*3: RTS and CTS control are not used. CTS output is HI always.

Serial interface software: Arduino UNO/ W6100 ethernet shield

```
// Moisture content = ST,+00000.00spsp%crlf
// MF-50 Commands:
   // Q   = Output the current data.
   // SIR = Output data continuously.
   // C   = Stop data output by SIR command.
   // QM  = Output data during measurement only.
//-----------------------------------------------------------------
// Serial setup for MF-50
//-----------------------------------------------------------------
 Serial.begin(2400,SERIAL_7E1);   // To Analyzer RS-232 port.

 //-----------------------------------------------------------------
 // Check if Measuring Moisture. (Send data only when measuring)
 // Send command to analyzer to check if measuring batch moisture.
 //-----------------------------------------------------------------
   Serial.println("C"); //Stop
   Serial.println("QM"); // print QM with CR and LF.

 //-----------------------------------------------------------------
 // Check if Measuring Moisture
 //-----------------------------------------------------------------
  void Chk_Start_Measuring()
  {
   charcnt=0, charcnt1=0, charflag=0;
   if (start_measure==0)
     {
       Serial.println("QM");
       digitalWrite(3, LOW);      // Measuring LED off
       measure_time = 0, measure_count = 0;
     }

   if (start_measure==1)
     {
       digitalWrite(3, HIGH);     // Measuring LED on
       if (cont_read==0)          // Measurement Done.
         {
           Serial.println("QM");
         }

       if (cont_read==1)
         {
           measure_time = measure_time + 1;
           if (measure_time<=3)
             {
               Serial.println("Q");
             }
           else
             {
               start_measure=0;
               cont_read=0;
               Serial.println("QM");
               gotdata=3; // measure end.
             }
         }
     }
  }
```

```
//--------------------------------------------------------------------------
// Send command to analyzer to check if measuring batch moisture.
// Check if analyzer is sending Measurements
//--------------------------------------------------------------------------
 void Get_Measurement_Readings()
 {
  if(Serial.available())
   {
//--------------------------------------------------------------------------
// keep reading from Serial until there are bytes in the serial buffer
//                        123456789012 3  4  5 6  7
// Check for Data Stream: ST,+00000.00 sp sp % cr lf
//--------------------------------------------------------------------------
  while (Serial.available()>0)
        {
           reading=1, reading_done=0, start_measure=1;
           inByte = Serial.read();

           if (inByte == 'S' and charflag==0)
               {
                  charflag=1;
               }
           // Get reading
           if (charflag==1)
               {
                  charcnt=charcnt+1;
                  if (charcnt>7 and charcnt<13)
                  {
                      charcnt1=charcnt1+1;
                      chr[charcnt1] = inByte; // get current reading
                  }
               }
         }//while
           measure_count = measure_count + 2;
    }//if
   else // no bytes in serial buffer.
    {
     //Mb.MbsRun();
     if (reading==1) // check if measurement done.
     {
       reading_done=reading_done+1;
       if (reading_done>=4)
       {
         cont_read=1;
         reading=0;
         reading_done=0;
       }
     }
```

```
//-------------------------------------------------------------------------------
// Get Batch Moisture data (0-10%)
// Measurement Complete!
//-------------------------------------------------------------------------------
        if (gotdata==3)
          {
            start_measure=0, in_data=0, gotdata=0;
            //-----------------------------------------------------
            // Save Final Moisture Reading.
            //-----------------------------------------------------
            inByte1 ="";
            inByte1 += chr[1];        // x0.00
            inByte1 += chr[2];        // 0x.00
          //inByte1 += chr[3];
            inByte2 ="";
            inByte2 += chr[4];        // 00.x0
            inByte2 += chr[5];        // 00.0x

            if (measure_count>75 and inByte1 != "00")
              {
                bmi00 = inByte1.toInt();  // int part
                bmf00 = inByte2.toInt();  // fractional part
                EEPROM.update(1,bmi00);
                EEPROM.update(2,bmf00);
              }
          }
    }//else
 }
```

Hardware Parts List:

1 – W6100 ethernet Shield
1 – Arduino UNO
1 – Arduino prototype board
1 – MAX233 chip
1 – 20 pin dip socket
1 – 0.1 uF capacitor
2 – 100 ohm resistors
2 – LEDs (yellow and green)
1 – 12vdc power supply or 5vdc power supply to plug into USB connector
1 – CAT5 ethernet cable

Hardware Diagram: