COMPUTER INTERFACING APPLICATION BRIEF

TOLEDO® CONTINUOUS FORMAT

TOLEDO

AP CONTIN 100

NOTICE!

This application brief is provided to illustrate a method of interfacing the Toledo Scale continuous mode output to a PC compatible computer. The program examples in this application brief **DO NOT** constitute a complete, working program suitable for use as a commercial application.

TOLEDO SCALE DOES NOT ACCEPT RESPONSIBILITY FOR ANY LIABILITY ARISING FROM ANY USE OF THE PROGRAM EXAMPLES DESCRIBED IN THIS APPLICATION BRIEF.

TOLEDO SCALE MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR ANY OTHER EXPRESSED OR IMPLIED WARRANTY.

TOLEDO SCALE COMPUTER INTERFACING APPLICATION BRIEF

SUBJECT: USING TOLEDO SCALE CONTINUOUS MODE DATA OUTPUT WITH A PC COMPATIBLE COMPUTER SERIAL PORT

1.0 CONTINUOUS MODE FORMAT DESCRIPTION

Toledo Scale Indicators, Models 8132, 8140, 8142, 8510, 8520, 8525, and 8530 supply the Toledo Scale continuous mode data output. The continuous mode output provides status information, displayed weight and tare weight data in a fixed format output every display update (approximately 4 to 16 updates per second depending on indicator model used). The continuous mode data output has several advantages for computer interfacing when compared to the demand mode output. Some of the continuous mode output features are:

- **Real Time Data:** The continuous mode output is transmitted every display update. The data output occurs even when the scale is in a "motion" condition.

- **Status Information:** The continuous mode output includes three status word bytes that contain individual bits for motion, sign, overcapacity, net/gross mode, lb/kg weight units, individual setpoint cutoff bits, decimal point location, print request and whether the scale has been zeroed after powerup.

- **Fixed Format:** The continuous mode output is consistent within the Toledo Scale family of indicators. The same program written to read weight data from the earliest Model 8132 Indicator is able to read weight data from the latest Model 8530 DigiTOL® Indicator. This continuity of support for the continuous mode output prevents your software investment from becoming obsolete as new models are introduced.

  Note: There are some minor differences in status word bit definitions from one model indicator to another as not all indicators support setpoint operation.

1.1 CONTINUOUS MODE DATA FORMAT

Data	STX	SWA	SWB	SWC	MSD	-	-	-	-	LSD	MSD	-	-	-	-	LSD	CR	CKS
Note	1	2			3						4						5	6

CONTINUOUS DATA FORMAT NOTES:

1 - <STX> ASCII Start of Text character, Hex 02.
2 - <SWA>, <SWB>, <SWC> Status Word A, B, C. Refer to Status Word A, B and C bit definitions, following.
3 - Displayed weight. Six digits, no decimal point or sign. Nonsignificant leading zeros are replaced with spaces in the lb weight unit mode.
4 - Tare weight. Six digits, no decimal point or sign. Nonsignificant leading zeros are replaced with spaces in the lb weight unit mode.
5 - <CR> ASCII Carriage Return, Hex 0D.
6 - <CKS> Optional checksum character. Checksum is defined as the 2's complement of the 7 low order bits of the binary sum of all characters on a line, preceding the checksum including the <STX> and <CR>. Checksum can be disabled in setup.

1.2 Standard (Nonsetpoint) Version Status Word Bit Definitions

Function	Selection	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Decimal Point or Dummy Zero	X00	A	A	*		0	0	0
	X0	L	L			0	0	1
	X	W	W			0	1	0
	0.X	A	A			0	1	1
	0.0X	Y	Y			1	0	0
	0.00X	S	S			1	0	1
	0.000X					1	1	0
	0.0000X	A	A			1	1	1
Increment Size	X = 1	0	1	0	1	*		
	X = 2			1	0			
	X = 5			1	1			

Status Word A Bit Definitions
(*) Bits Not Applicable to Function

Function	Bit
Gross/Net: Net = 1	0
Under Zero: Negative = 1	1
Out of Range = 1	2
Motion = 1	3
lb/kg: kg = 1	4
Always a 1	5
Powerup: Not Zeroed = 1	6

Status Word B Bit Definitions

Function	Bit
Always a 0	0
Always a 0	1
Always a 0	2
Print Request = 1	3
Expanded Weight = 1	4
Always a 1	5
Manual tare in kg = 1	6

Status Word C Bit Definitions

NONSETPOINT VERSION STATUS WORD BIT DEFINITION NOTES:

1 - The out of range bit in Status Word B indicates that the scale display is either blanked overcapacity or blanked under. Weight data and status bit information in the continuous transmission is invalid when the out of range bit is true and must be ignored. Most Toledo Scale indicators transmit spaces in place of the weight data when an out of range condition occurs.

2 - The decimal point location is coded into Status Word A, bits 0, 1 and 2 only. The displayed weight and tare weight fields do not include a decimal point. Dummy zeroes will be included in the weight fields transmitted if the scale build uses dummy zeroes.

3 - These bit definitions are for indicators that do not provide setpoint operation, or for the JN Port of the Model 8142 Indicator. Refer to the indicator technical manual to determine the precise status word bit definition of that indicator.

1.3 Setpoint Version Status Word Bit Definitions

Function	Decimal Point Selection	Bit 2	Bit 1	Bit 0
Decimal Point or Dummy Zero	X00	0	0	0
	X0	0	0	1
	X	0	1	0
	0.X	0	1	1
	0.0X	1	0	0
	0.00X	1	0	1
	0.000X	1	1	0
	0.0000X	1	1	1

Function	Bit
Setpoint 1: Feeding = 0	3
Setpoint 2: Feeding = 0	4
Always a 1	5
Setpoint 3, Setpoint 1 Fast: Feeding = 0	6

Status Word A Bit Definitions

Function	Bit
Gross/Net: Net = 1	0
Under Zero: Negative = 1	1
Out of Range = 1	2
Motion = 1	3
lb/kg: kg = 1	4
Always a 1	5
Tolerance 1: In Tolerance = 0	6

Status Word B Bit Definitions

Function	Bit
Always a 0	0
Always a 0	1
Always a 0	2
Print Request = 1	3
Setpoint 4, Setpoint 2 Fast: Feeding = 0	4
Always a 1	5
Tolerance 2: In Tolerance = 0	6

Status Word C Bit Definitions

SETPOINT VERSION STATUS WORD BIT DEFINITION NOTES:

1 - The out of range bit in Status Word B indicates that the scale display is either blanked overcapacity or blanked under. Weight data, status bit and setpoint information in the continuous transmission is invalid when the out of range bit is true and must be ignored. Most Toledo Scale indicators transmit spaces in place of the weight data when an out of range condition occurs.

2 - The decimal point location is coded into Status Word A, bits 0, 1 and 2 only. The displayed weight and tare weight fields do not include a decimal point. Dummy zeroes will be included in the weight fields transmitted if the scale build uses dummy zeroes.

3 - The status word bit definitions listed are for the Model 8142 Indicator which provides either four, single speed setpoint operation or two, dual speed setpoint operation. Not all indicators that support setpoint operation do so in the same manner. For example the 8530 Indicator has four, single speed setpoint operation. The 8510-2001 Indicator has two, single speed setpoint operation. Refer to the indicator technical manual to determine the precise status word bit definition of that indicator.

2.0 HARDWARE INTERFACING

2.1 Serial Interconnect Pinout Table

PC compatible computers normally use either a female, twenty five pin (DB-25F) connector or a female, nine pin (DE-9F) connector for the RS-232C serial port, addressed as either COM1 or COM2. Refer to the Serial Interconnect Table for interconnect cable pinout.

Serial Interconnect Table

Description	Indicators					PC	
	8510-1001 TB2	8510-2001 Printer	8520 TB1	All other Indicators		DB-25F	DE-9F
				Desk Version DB-25M	Wall Version Twist Lock		
Shield	Stud	Shell	Stud	Chassis	Chassis	N.C.	N.C.
TxD (RS-232)	2	3	1	2	B	3	2
RxD (RS-232)	3	2	2	3	C	2	3
Logic Ground	1	5	6	7	G	7	5
Jumpers	None	None	W1(1-2) W3(1-2)	⌜4 ⌞5	⌜D ⌞E	⌜4 ⌞5	⌜7 ⌞8

SERIAL INTERCONNECT TABLE NOTES:

1 - Toledo Scale Indicator's serial I/O connectors often contain non RS-232C signals (such as 20 mA current loop, RS-422, or analog signals. Erratic operation of the indicator can result if any pins other than those listed in the Serial Interconnect Table are connected.

2 - The heading [All Other Indicators] in the Serial Interconnect Table, refers to Toledo Scale Models 8132 (With EIA Option), 8140 (With Printer Interface Option), 8142, 8525 (with Data I/O option) and 8530.

3 - Jumpers shown are located in the cable connector at either the indicator end or the PC end of the cable as indicated in the Serial Interconnect Table.

2.2 Interconnect Cables

Scale Connector	PC Connector	Toledo Scale Part Number
JN, DB-25M Desk and Rack Enclosure Connector (8140, 8142, 8525, 8530)	DB-25F	136053 00A
	DE-9F	136054 00A
DE-9M (8510-2001)	DB-25F	136055 00A
	DE-9F	136056 00A
JN, Twist-Lock Washdown Enclosure Connector (8140, 8142, 8530)	DB-25F	136057 00A
	DE-9F	136058 00A

Serial Interconnect Cables are available from your local authorized Toledo Scale Distributor.

3.0 SOFTWARE INTERFACING

The Toledo Scale continuous mode output can be read as needed by the PC or used to constantly update a display. In this discussion the PC reads displayed weight, as needed, from a Toledo® Indicator programmed to output continuous mode data at **4800 baud, even parity, one stop bit and with checksum disabled.**

SOFTWARE INTERFACING NOTES:

1 - The code fragments in this application were tested with MicroSoft® Quick Basic® Version 4.5. The code fragments shown may require modification to operate correctly if used with other Basic Compilers or Interpreters.

2 - The code fragments in this example are provided to illustrate a method of interfacing the Toledo Scale continuous mode output to a PC compatible computer and **DO NOT** constitute a complete, working program suitable for use as a commercial application.
TOLEDO SCALE DOES NOT ACCEPT RESPONSIBILITY FOR ANY LIABILITY ARISING FROM ANY USE OF THE CODE FRAGMENTS DESCRIBED IN THIS EXAMPLE.

3.1 Open the Serial Port

This line opens serial port COM1 for bidirectional communication at 4800 baud, even parity, 7 data bits, 1 stop bit. The **",CS,DS,CD,LF"** are required by MicroSoft® Quick Basic® to setup the serial port handshaking. This statement must be executed before calling the **SCANREAD** subroutine.

```
OPEN "COM1:4800,E,7,1,CS,DS,CD,LF" FOR RANDOM AS #1
```

3.2 Read Data From Serial Port

When reading data into a serial port it is recommended that a time out abort be used to prevent the program from "locking up" if data is not received within a preset period of time. The following code fragment sets up a one second timeout and waits for data to enter the serial port. If the timer times out without data being received, the error message is printed and a **RETURN** statement is executed.

The Start of Text <STX> character transmitted at the beginning of each continuous data output is used by the program to synchronize data reception. Data received into the serial port is examined one character at a time until the <STX> character is detected, at this time one complete continuous format message is read in and saved to the string variable **SCALEDAT$**.

```
SCANREAD: REM Subroutine to read in one continuous format message.
  START! = TIMER: REM Initialize timer.
SCANWAIT:
  IF (TIMER - START!) > 1 THEN
    CLOSE #1
    PRINT "ERROR NO SCALE COMMUNICATION"
    RETURN
    END IF
GRABCHAR: REM Check input buffer, if buffer is empty then go to SCANWAIT.
  IF EOF(1) THEN GOTO SCANWAIT
  Q$ = INPUT$(1,1): REM Read one character from input buffer.
  IF Q$ < > CHR$(2) THEN GOTO GRABCHAR: REM If character is not STX then goto GRABCHAR
    LINE INPUT #1, SCALEDAT$: REM Read one line of data into string variable SCALEDAT$
    RETURN:
```

Note: The serial port must be closed after data is read in to prevent an buffer overflow error. Execute a **CLOSE #1** statement after calling the **SCANREAD** subroutine.

3.3 Decode Status

The following code fragment decodes the status bits in status words B and C and stores the results in integer variables NET%, MINUS%, OVERCAP%, MOTION%, KG%, PRINTREQUEST%. The status variables are set to a zero to indicate a false condition, or are set to a nonzero value to indicate a true condition.

```
READSTATUS:
   SWB% = ASC(MID$(SCALEDAT$, 2, 1))
   NET% = SWB% AND 1
   MINUS% = SWB% AND 2
   OVERCAP% = SWB% AND 4
   MOTION% = SWB% AND 8
   KG% = SWB% AND 16
   SWC% = ASC(MID$(SCALEDAT$, 3, 1))
   PRINTREQUEST% = SWC% AND 8
   RETURN
```

3.4 Decode Decimal Point Location and Extract Weight Fields

The following code fragment decodes the decimal point information from status word A and stores the result in integer variable DP%. The displayed weight is then extracted from SCALEDAT$ and stored in string variable WT$. The subroutine DADJUST is then called to adjust the decimal point in the weight field. The process is then repeated for the tare weight.

```
READWEIGHT:
   SWA% = ASC(LEFT$(SCALEDAT$, 1))
   DP% = SWA% AND 7
   WT$ = MID$(SCALEDAT$, 4, 6)
   GOSUB DADJUST
   WEIGHT$ = RESULT$
   WT$ = MID$(SCALEDAT$, 10, 6)
   GOSUB DADJUST
   TARE$ = RESULT$
   RETURN

DADJUST:
   IF DP% > 1 THEN
   RESULT$ = LEFT$(WT$, -1 * DP%+8) + CHR$(46) + RIGHT$(WT$, DP% - 2)
      ELSE
      RESULT$ = WT$
      END IF
   IF RIGHT$(RESULT$, 1) = CHR$(46) THEN RESULT$ = WT$
   RETURN
```

3.5 Error Handling

It is recommended that an error handling routine be included in your program to prevent the program from aborting if a run time error occurs. Run time errors could be caused by selecting a serial port that does not exist, connecting the serial port to a device that is not powered up or errors in the program. The error handling routine listed below has error messages for some of the more common errors that can occur and will display the basic error code for any run time error that occurs. Refer to the MicroSoft® Quick Basic® documentation for error code descriptions.

The code fragment listing assumes that there is an ON ERROR GOTO HANDLER statement following the RESTART: line label at the beginning of the program. The error handler routine is placed at the end of the program after your subroutines.

```
RESTART:
   ON ERROR GOTO HANDLER

   (BODY OF PROGRAM)

HANDLER:
   CLOSE #1
   LOCATE 12, 20
   PRINT "ERROR #"; ERR;
   IF ERR = 52 THEN PRINT "BAD FILE NUMBER";
   IF ERR = 69 THEN PRINT "COMMUNICATION BUFFER OVERFLOW";
   IF ERR = 25 THEN PRINT "DEVICE FAULT;"
   IF ERR = 57 THEN PRINT "CHECK SCALE, DEVICE I/O ERROR";
   IF ERR = 68 THEN PRINT "CAN'T OPEN COM PORT";
   IF ERR = 24 THEN PRINT "DEVICE TIMEOUT";
   IF ERR = 53 THEN PRINT "FILE ("; FILENAME$; ") NOT FOUND."
   START! = TIMER
   DO WHILE TIMER - START! < 2
      LOOP
   CLEAR
   GOTO RESTART
```

Note: Place a REM statement at the beginning of the ON ERROR statement during program development to permit the MicroSoft® Quick Basic smart editor to assist you in debugging your program. Once your program is running without errors, remove the REM statement from the ON ERROR statement before final compilation.

4.0 DEMO PROGRAM LISTING

```
REM ************************************************************************
REM *  TOLEDO SCALE CONTINUOUS FORMAT COMMUNICATION DEMO PROGRAM
REM *  WRITTEN BY PHILIP OUELLETTE      QUICK BASIC VERSION 4.5
REM *  VERSION 1                    DATE: 2/13/1991
REM * THIS PROGRAM IS FURNISHED WITH THE UNDERSTANDING THAT THE ESSENCE
REM * THEREOF WILL NOT BE REPRODUCED IN WHOLE OR IN PART WITHOUT THE
REM * EXPRESS WRITTEN PERMISSION OF TOLEDO SCALE.
REM * COPYRIGHT: TOLEDO SCALE CO. 1991
REM ************************************************************************

REM INDICATOR IS PROGRAMMED FOR CONTINUOUS FORMAT DATA OUTPUT AT 4800 BAUD
REM SEVEN DATA BITS, EVEN PARITY, ONE STOP BIT.  CHECKSUM IS DISABLED.

RESTART:
 ON ERROR GOTO HANDLER
 CLS
 LOCATE 1, 25
 PRINT "TOLEDO SCALE CONTINUOUS DEMO PROGRAM"
 LOCATE 4, 30
 PRINT "PRESS [Esc] KEY TO EXIT PROGRAM"
 LOCATE 5, 23
 PRINT "PRESS ANY OTHER KEY TO READ SCALE WEIGHT"
 VIEW PRINT 13 TO 17

MAIN:
 Q$ = UCASE$(INKEY$)
 IF Q$ = "" THEN GOTO MAIN
 IF Q$ = CHR$(27) THEN
   VIEW PRINT
   CLS
   END
   END IF
 OPEN "COM1:4800,E,7,1,CS,DS,CD,LF" FOR RANDOM AS #1
 GOSUB SCANREAD
 CLOSE #1
 GOSUB READSTATUS
 GOSUB READWEIGHT
 CLS 2
 LOCATE 13, 20
 PRINT "DISPLAYED WEIGHT: "; WEIGHT$; "    TARE WEIGHT: "; TARE$
 IF MOTION% > 0 THEN
   LOCATE 14, 20
   PRINT "SCALE IS IN MOTION "
   END IF
 IF OVERCAP% > 0 THEN
   LOCATE 14, 40
   PRINT "SCALE IS OUT OF RANGE"
   END IF
 GOTO MAIN

REM   END OF MAIN PROGRAM
REM ************************************************************************
```

```
REM ************************************************************************
REM   SUBROUTINE TO READ IN ONE CONTINUOUS FORMAT MESSAGE
SCANREAD:
  START! = TIMER
SCANWAIT:
  IF (TIMER - START!) > 1 THEN
    CLOSE #1
    PRINT "ERROR NO SCALE COMMUNICATION"
    RETURN
    END IF
GRABCHAR:
  IF EOF(1) THEN GOTO SCANWAIT
  Q$ = INPUT$(1, 1)
  IF Q$ < > CHR$(2) THEN GOTO GRABCHAR
REM   READ IN ONE LINE OF DATA, TERMINATED BY A <CR>
  LINE INPUT #1, SCALEDAT$
  RETURN


REM ************************************************************************
REM   SUBROUTINE TO DECODE STATUS WORDS B AND C
READSTATUS:
  SWB% = ASC(MID$(SCALEDAT$, 2, 1))
  NET% = SWB% AND 1
  MINUS% = SWB% AND 2
  IF MINUS% = 0 THEN
    SIGN$ = " "
    ELSE
    SIGN$ = "-"
    END IF
  OVERCAP% = SWB% AND 4
  MOTION% = SWB% AND 8
  KG% = SWB% AND 16
  SWC% = ASC(MID$(SCALEDAT$, 3, 1))
  PRINTREQUEST% = SWC% AND 8
  RETURN


REM ************************************************************************
REM   SUBROUTINE TO EXTRACT WEIGHT DATA
READWEIGHT:
  SWA% = ASC(LEFT$(SCALEDAT$, 1))
  DP% = SWA% AND 7
  WT$ = MID$(SCALEDAT$, 4, 6)
  GOSUB DADJUST
  WEIGHT$ = SIGN$ + RESULT$
  WT$ = MID$(SCALEDAT$, 10, 6)
  GOSUB DADJUST
  TARE$ = RESULT$
  RETURN
```

```
REM ************************************************************************
REM   SUBROUTINE TO ADJUST DECIMAL POINT IN WEIGHT DATA
DADJUST:
  IF DP% > 1 THEN
    RESULT$ = LEFT$(WT$, -1 * DP% + 8) + CHR$(46) + RIGHT$(WT$, DP% - 2)
    ELSE
    RESULT$ = WT$
    END IF
  IF RIGHT$(RESULT$, 1) = CHR$(46) THEN RESULT$ = WT$
  RETURN


REM ************************************************************************
REM   ERROR HANDLING ROUTINE
HANDLER:
  CLOSE #1
  LOCATE 12, 20
  PRINT "ERROR #"; ERR;
  IF ERR = 52 THEN PRINT "BAD FILE NUMBER";
  IF ERR = 69 THEN PRINT "COMMUNICATION BUFFER OVERFLOW";
  IF ERR = 25 THEN PRINT "DEVICE FAULT;"
  IF ERR = 57 THEN PRINT "CHECK SCALE, DEVICE I/O ERROR";
  IF ERR = 68 THEN PRINT "CAN'T OPEN COM PORT";
  IF ERR = 24 THEN PRINT "DEVICE TIMEOUT";
  IF ERR = 53 THEN PRINT "FILE ("; FILENAME$; ") NOT FOUND."
  START! = TIMER
  DO WHILE TIMER - START! < 2
    LOOP
    CLEAR
  GOTO RESTART


REM   END OF SUBROUTINES
REM ************************************************************************
```