

Getting Started Firmware V1003

The DF 3 USB gets power from the 5V supply of the AutomationDirect PLC. In the package is the DF 3 USB and an 18" 6-pin cable. The cable can be plugged in with either end to the logger and PLC. If datalogging from Port 2 of the DL06 or DL250/260 then order the DF15TO6 adapter. This adapter will get power from Port2 and allow serial communication to the PLC using the 6 pin cable enclosed.

Attaching the DF 3 USB to the PLC

Here is a table of possible ports that can be used by the DF 3 USB.

PLC	PORT 1	PORT 2	Communication
DL105	NO	NA	NA
DL05	YES	YES	DIRECTNET
DL06	YES	YES (with DF15TO6 adapter)	DIRECTNET
DL250	YES	YES (with DF15TO6 adapter)	DIRECTNET
DL250-1	YES	YES (with DF15TO6 adapter)	DIRECTNET
DL260	YES	YES (with DF15TO6 adapter)	DIRECTNET
Click	YES	YES	MODBUS

Before attempting to log using the DF 3 USB, read the next section to get the USB thumb drive ready for logging.

DF 3 USB Logging with DirectLogic Series PLC



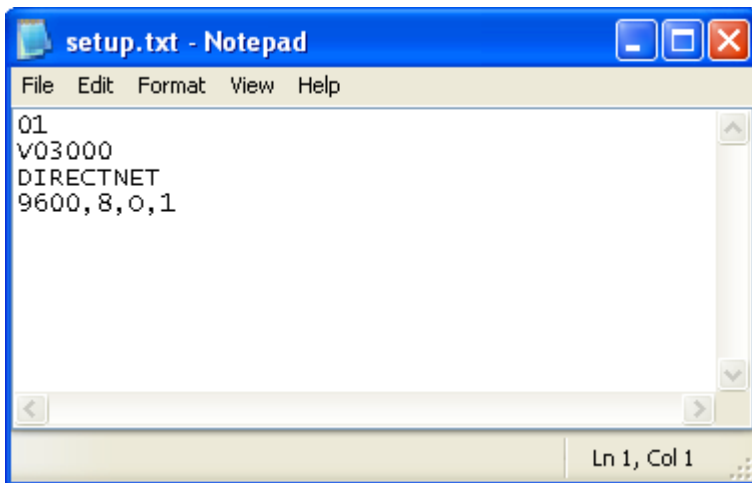
To begin logging to an AutomationDirect PLC, the USB thumb drive must have a **SETUP . TXT** file saved to the USB drive and either a **DATLOG1 . TXT** file, **DATLOG2 . TXT** file or both.

- SETUP . TXT** This file is used to set operating parameters.
- DATLOG1 . TXT** This file is used to log by time interval.
- DATLOG2 . TXT** This file is used to log by an event that occurs in the PLC.

SETUP . TXT Example. This example will be used for the rest of this document. V03000 was used for the Base location. You may use any valid Vmemory address that is LESS THAN V10000 for your base location.

Location	Definition
BASE+0	File I/O function plc writes data here
BASE+1	I/O function results usb logger writes here
BASE+2	File write low memory location plc writes here
BASE+3	File write high memory location plc writes here
BASE+4	Firmware version (read only)
BASE+5	1 = datlog1.txt exists (read only)
BASE+6	Datlog1 time delay (read only)
BASE+7	Datlog1 file size double word decimal (read only)
BASE+9	1 = datlog2.txt exists (read only)
BASE+10	Datlog2 memory location (read only)
BASE+11	Datlog2 value (read only)
BASE+12	Datlog2 file size double word decimal (read only)
BASE+14	Number of datlog1 backup files (read only)
BASE+15	Number of datlog2 backup files (read only)
BASE+16	Chip version = 2511 (read only)

RESERVE 17 VMEMORY LOCATIONS FROM V03000 TO V03020. DOUBLE CHECK NONE OF THESE LOCATION ARE IN USE IN THE LADDER LOGIC. THE USB LOGGER WRITES DATA TO MOST OF THESE LOCATIONS!



Use Notepad in Windows to create the file shown above. Save as **SETUP.TXT** to the USB thumb drive. Make sure a carriage return is on each line, even the last line.

Line	Example	Definition	Value Ranges
Top Line	01	PLC Address	01 to 99
Second Line	V03000	Base Vmem Address	02000 to 05000
Third Line	DIRECTNET	Communication Type	DIRECTNET or MODBUS
Bottom Line	9600,8,0,1	Baud rate	See Note Below

Baud rate options are:

4800, 9600, 19200, 38400

Parity options are only O and N.

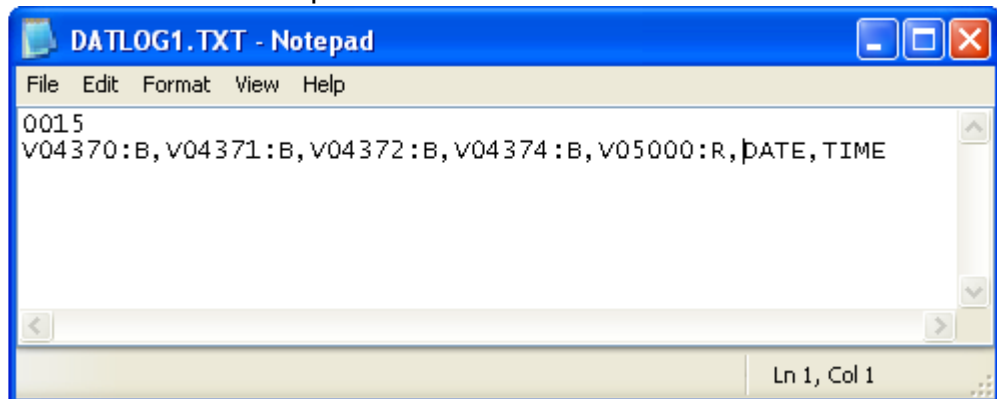
Only 8 data bits and 1 stop bit is supported.

If no baud rate is specified, Modbus will default to 38400,8,O,1 and DirectNet will default to 9600,8,O,1.

If lines 3 and 4 are omitted (similar to the DFII SD Logger **SETUP.TXT** file) then the default communication will be Directnet at 9600,8,O,1.

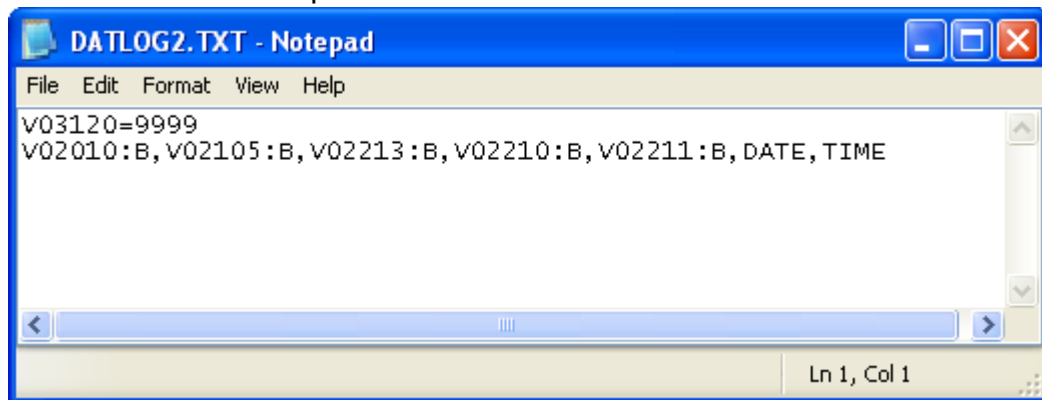
Communication options are DIRECTNET and MODBUS. Modbus is used for Click PLC operation. See the Click section below for more information.

DATLOG1.TXT Example:



When the DF 3 USB powers up and finds this file, it will be set to log the 5 vmemory locations listed plus DATE and TIME every 15 seconds.

DATLOG2.TXT Example:



When the DF 3 USB powers up and finds this file, it will be set to log the 5 vmemory location listed plus DATE and TIME when V03120 is equal to 9999. The DF 3 USB will then write a 0 to V03120 and log the values.

Both **DATLOG1.TXT** and **DATLOG2.TXT** can exist on the card at the same time. Both files will be used for logging.

The DF 3 USB logs directly to the files created above. No extra files are created.

DATLOG1.TXT and **DATLOG2.TXT** are APPENDED each time a write occurs. The files will grow continuously unless a backup is made. See the **BACKING UP LOG FILES** section below.

The DF 3 USB uses 17 V memory locations to indicate file status and other information to the PLC. 13 of the locations should be viewed as BCD WORD. V03007 and V03014 should be viewed as Decimal, DWORD.

Data Types Supported:

When creating a **DATLOG1.TXT** or **DATLOG2.TXT** use the following table to indicate the data type to use when reading the PLC and saving the values to the USB Drive.

Abbreviation	Data Type Recorded
:B	Records 4 digit BCD value
:b	Records 8 digit BCD value
:R	Records the Real (floating point) Value.
:D	records the Unsigned Decimal Value.
:d	Records the Signed Decimal Value.
%XX	Records XX characters of Text

Ex:

V04370:B will record values between 0000 and 9999 and will also record hex characters as well (00FF, A00A, etc...)

V04544:b will record values between 0 and 99999999 and will record Hex chars as well

V04440:d will record values between -32767 and +32767

V04555:D will record values between 0 and 65535

V10000%08 will record a TEXT string located at V10000 and is 8 characters.

V10000%16 will record a TEXT string located at V10000 and is 16 characters.

V07007%00 will read V07007 for the number of characters and record a TEXT string located at V07010.

Commas are not allowed and will be converted to a dash. Only **64** Characters are supported.

From a DataView in DirectSoft 4 or 5:

DF-3-USB Logging DirectNet to AutomationDirect PLC			
Element	Status	Edits	
V3000	0		File USB IO functions written from PLC to this location defined in SETUP.TXT
V3001	1011		USB status values written from USB logger to here.
V3002	600		Start location of Vmemory in Hex when writing data files.
V3003	620		End location of Vmemory in Hex when writing data files.
V3004	1002		USB logger firmware version
V3005	1		1 = DATLOG1.TXT exists and USB Logger is logging to this file by TIME Interval
V3006	15		This is the time interval being used for DATLOG1 logging.
V3007	1426094		DATLOG1.TXT File size (Must be view as Decimal DWORD)
V3011	1		1 = DATLOG2.TXT exists and USB Logger is logging by EVENT
V3012	650		This is the Vmemory location monitored by the USB logger for DATLOG2.TXT
V3013	9999		This is the Vmemory Value being monitored to trigger a DATLOG2.TXT log event.
V3014	186		DATLOG2.TXT File size (Must be view as Decimal DWORD)
V3016	2		Number of DATLOG1.TXT backup files on USB Drive.
V3017	0		Number of DATLOG2.TXT backup files on USB Drive.
V3020	2511		Chip Version of USB Logger.

(Note, V03012 = 650(Hex) which is 3120 Octal. **DATLOG2 . TXT** in our example aboved used V03120 for event logging.)

V03000: This is the base memory area defined in **SETUP . TXT** . You may use any value between V2000 and V5000 for your base location. RESERVE 17 VMEMORY LOCATIONS FROM VXXXX TO VXXXX + 16. DOUBLE CHECK NONE OF THESE LOCATION ARE IN USE IN THE LADDER LOGIC. THE USB LOGGER WRITES DATA TO MOST OF THESE LOCATIONS! The PLC write values to this register to cause the DF 3 USB to execute certain tasks defined below.

V03001 The DF 3 USB writes status indication values back to this location to show progress and current USB status.

- 0x1010 is written if no USB drive in present.
- 0x1011 is written when a new line it written to **DATLOG1 . TXT**.
- 0x1012 is written when a new line it written to **DATLOG2 . TXT**.
- 0x1000 is written when when the USB drive is present at power up or insertion only.

V03002 to **V03020** should be treated as read only by the PLC. Only the DF 3 USB should write values to these locations. The plc should not try to write to these memory areas.

V03004 to **V03020** are written to 0 by the DF 3 USB at power up and each time the USB Drive is inserted.

Backing up log file

Both the **DATLOG1 . TXT** and **DATLOG2 . TXT** file can be periodically backed up using ladder logic. By writing B001 to V03000, the DF 3 USB will create a backup copy of the current contents of **DATLOG1 . TXT** and rename it **DATLOG1A . TXT** up to **DATLOG1Z . TXT**. This means 26 backups can exist on the card before the backup command no longer functions. Look at V03016 to see how many backups exist on the card. By writing B002 to V03000, you can backup **DATLOG2 . TXT** and the same is true, you get 26 backups. When a backup occurs, the file is copied, renamed and a new blank **DATLOG1 . TXT** file is created with the top 2 lines copied over so the exact same logging will take place as before.

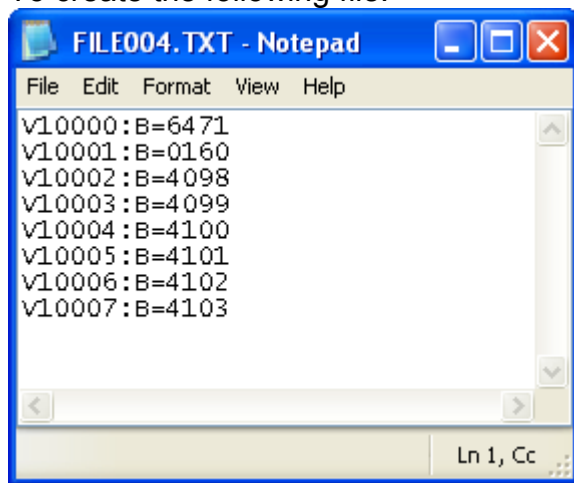
File Size

At power up, USB insertion and each time the DATLOGx file increases in size, the current file size is written to V03007 for **DATLOG1.TXT** and V03014 for **DATLOG2.TXT**. These are written to the PLC as Double Word Decimal numbers. To view these registers in a Dataview screen, highlight V03007 and use the pull downs to change to Decimal and DWORD.

Using the File I/O routines

It is very easy to create text-base files that contain the current values of contiguous vmemory locations. Only the BCD value is written. Decimal and Real are not supported in the File IO routines.

To create the following file:



Will require the following sequence:

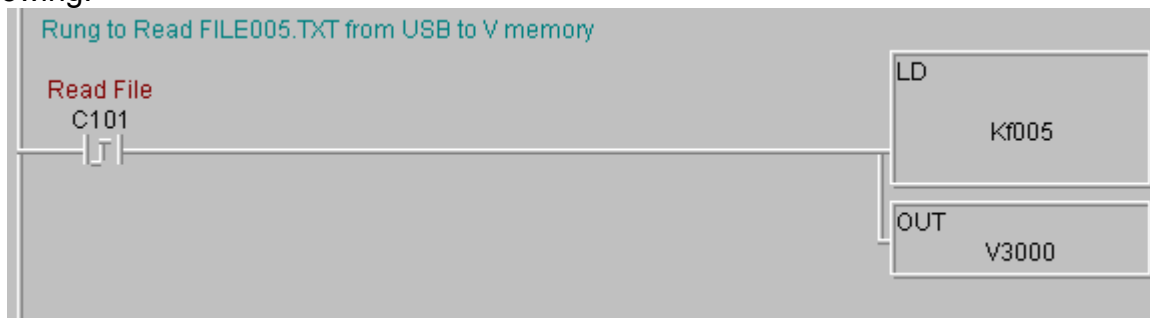


That is it. The DF 3 USB will take care of the saving the data to the USB drive.

Reading a **FILEXXX.TXT** file into Vmemory

Element	Status	Edits
V7500	1	
V7501	2	
V7502	3	
V7503	4	
V7504	1234	
V7505	999	

To read a file named **FILE005.TXT** and write those vmemory values to the PLC requires the following:



That is it. The DF 3 USB will take care of the reading the data from the USB drive and saving the values to the vmemory indicated.

The following is a list of values your PLC program can write to V03000.
V03000 Function:

V03000=	USB Function
Fxxx	Upload (F)ile if it exists from USB to PLC.
Exxx	Check to see if the file (E)xists on the USB Drive.
Dxxx	(D)elele File from the USB drive if it exists.
Cxxx	(C)reate File on the USB and overwrite any file with same name.
Bxxx	(B)ackup Datlog1.txt or Datlog2.txt.
Axxx	(A)ppend to existing USB file. If it does not exist, the file is created anyway.

XXX Can equal any value from 000 to 999 and include hex characters such as FFF or 9A9 or 00F etc...

The DF 3 USB will write back a status value to V03001.

V03001 Values:

If V03000 = Fxxx

- 1000 is written to V03001 while checking for file

- 1001 if file uploaded from flash ok.
- 1002 if file did not exist
- 1003 if busy uploading the file then will change to 1 if successful

If V03000 = Exxx

- 1000 is written to V03001 while checking for file
- 1006 if file does exist
- 1007 if file does not exist

If V03000 = Dxxx 0000 while busy checking for file

- 1008 is written to V03001 if file was delete successfully
- 1009 if file did not exist

If V03000 = Cxxx

- 1000 is written to V03001 while checking for file
- 1003 if busy creating the file then will change to
- 1002 if there was an error
- 1001 if successful

If V03000 = Bxxx

- 1000 is written to V03001 while checking for file
- 1003 if error creating backup
- 1002 if backup failed because all 26 names used
- 1001 if successful
- V03016 or V03017 will increment
- V03007 or V03014 will change to new file size (smaller)

If V03000 = Axxx

- 1000 is written to V03001 while checking for file
- 1004 if busy appending to the file then will change to
- 1002 if there was an error
- 1001 if successful

V03002 Values: Written by PLC

The hex start address for the Data you want to save or append

V03003 Values: Written by PLC

The hex end address for the Data you want to save or append

To get the Hex address use the LDA command from DirectSoft. This will load the Vmemory hex address equivalent. Ex: [LDA O3000] will load the accumulator with the value 600 hex. Then [OUT V03002] to save 600 Hex to V03002. Do the same for V03003. Be sure to load V03002 and V03003 values before you write Cxxx or Axxx to V03000. Do Not write any values to V03001.

Power Up Files

There are 2 files that can exist on the card and are read into the plc at powerup. They are called **POWUP.TXT** and **POWUPD.TXT**. They are written exactly like the example above (**FILE005.TXT**) but they are only looked at during the power up cycle.

If **POWUP.TXT** exists on the drive at power up, the contents of the file are written to the v memory EACH time the logger is powered.

If **POWUPD.TXT** exists on the drive at power up, the contents of the file are written to the v memory and the file is then deleted from the USB drive to prevent the write from happening again.

Insert Files

There are 2 files that can exist on the card and are read into the plc when the USB drive is inserted while power is applied. They are called **INSERT.TXT** and **INSERTD.TXT**. They are written exactly like the example above (**FILE005.TXT**) but they are only looked at when a USB drive is inserted into it's socket.

If **INSERT.TXT** exists on the drive at USB insertion, the contents of the file are written to the v memory EACH time the USB drive is inserted or the logger is powered. .

If **INSERTD.TXT** exists on the drive at power up, the contents of the file are written to the v memory and the file is then deleted from the USB drive.

DF 3 USB LOGGING with Click Series PLC

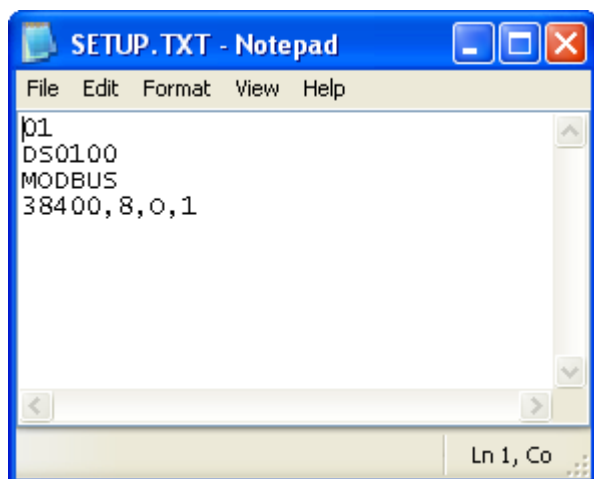


The procedure for logging the Click plc is very similar to the DirectLogic series. The main difference is how the **SETUP.TXT**, **DATLOG1.TXT**, **DATLOG2.TXT** and other power up and insertion files are written.

Communication with the Click must be set to Modbus. The default baud rate for Port 1 is 38400,8,O,1 and cannot be changed. Port2 can be changed but will default to 38400,8,O,1. The click plc will supply 5V dc to the DF 3 USB through the cable provided.

ONLY THE *DS* AND *DF* AREAS OF MEMORY IN THE CLICK PLC CAN BE ACCESSED BY THE DF 3 USB.

SETUP . TXT

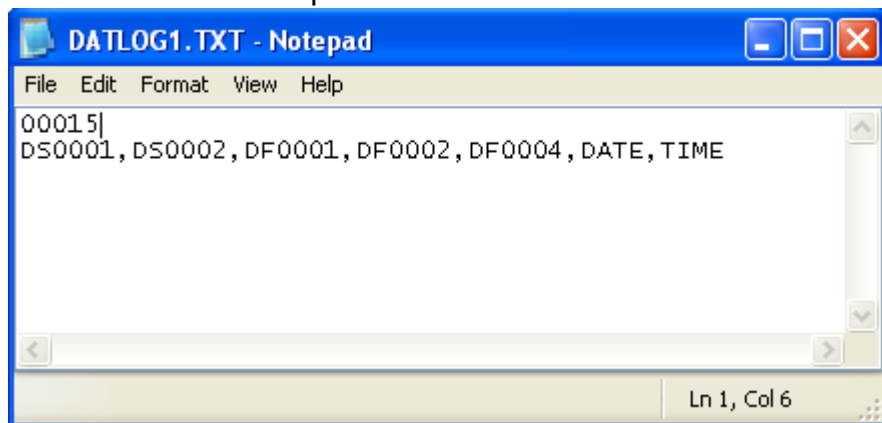


```
File Edit Format View Help
|01
DS0100
MODBUS
38400,8,0,1
Ln 1, Co
```

DS0100 is the BASE address for I/O functions. You may use DS0000 to DS4483. The BASE location must be in the DS area of memory.

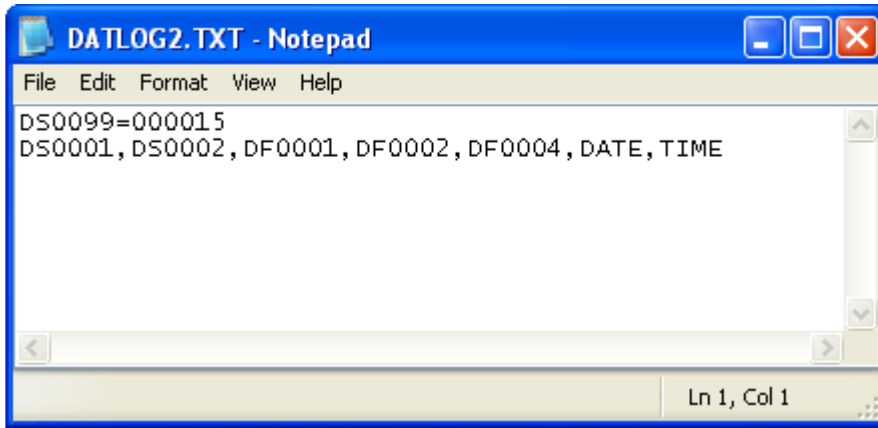
RESERVE 17 DS LOCATIONS FROM DS0100 TO DS0116. DOUBLE CHECK NONE OF THESE LOCATION ARE IN USE IN THE LADDER LOGIC. THE USB LOGGER WRITES DATA TO ALL BUT 2 OF THESE 17 LOCATIONS!

DATLOG1 . TXT Example:



```
File Edit Format View Help
00015|
DS0001, DS0002, DF0001, DF0002, DF0004, DATE, TIME
Ln 1, Col 6
```

This **DATLOG1 . TXT** file will log DS0001, DS0002 as Signed Decimal and DF0001,DF0002 and DF0004 as Floating point.



DATLOG2.TXT - Notepad

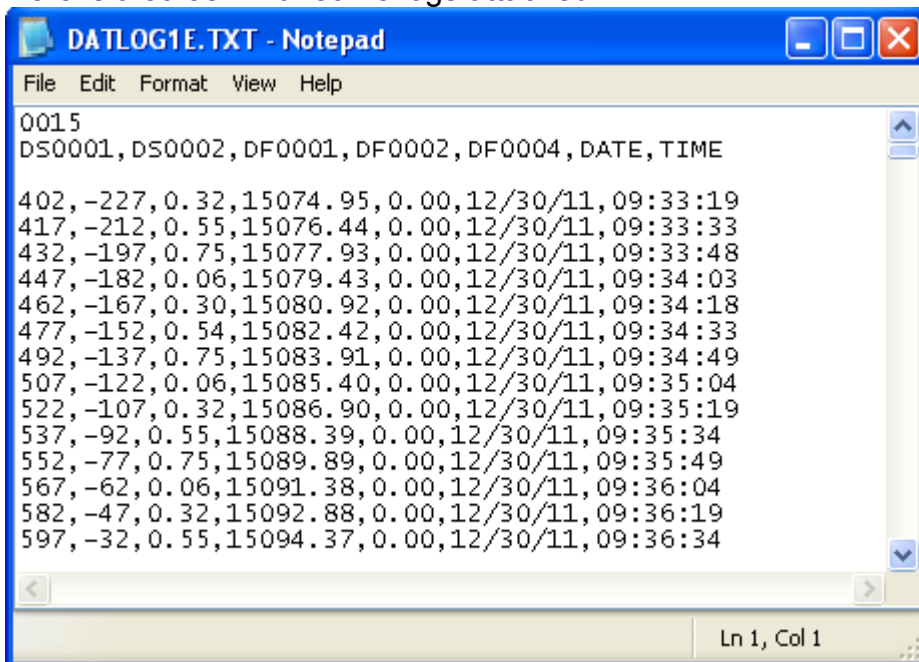
```
File Edit Format View Help
DS0099=000015
DS0001, DS0002, DF0001, DF0002, DF0004, DATE, TIME
```

Ln 1, Col 1

This DATLOG2 . TXT file will log DS0001, DS0002 as Signed Decimal and DF0001,DF0002 and DF0004 as Floating point. It will log when DS0099 = 15.

ALL LEADING ZEROES IN ALL THE .TXT FILES ARE IMPORTANT AND MUST BE TYPED IN THE FORMAT SHOWN!

Here is a screen with some logs attached.



DATLOG1E.TXT - Notepad

```
File Edit Format View Help
0015
DS0001, DS0002, DF0001, DF0002, DF0004, DATE, TIME

402, -227, 0.32, 15074.95, 0.00, 12/30/11, 09:33:19
417, -212, 0.55, 15076.44, 0.00, 12/30/11, 09:33:33
432, -197, 0.75, 15077.93, 0.00, 12/30/11, 09:33:48
447, -182, 0.06, 15079.43, 0.00, 12/30/11, 09:34:03
462, -167, 0.30, 15080.92, 0.00, 12/30/11, 09:34:18
477, -152, 0.54, 15082.42, 0.00, 12/30/11, 09:34:33
492, -137, 0.75, 15083.91, 0.00, 12/30/11, 09:34:49
507, -122, 0.06, 15085.40, 0.00, 12/30/11, 09:35:04
522, -107, 0.32, 15086.90, 0.00, 12/30/11, 09:35:19
537, -92, 0.55, 15088.39, 0.00, 12/30/11, 09:35:34
552, -77, 0.75, 15089.89, 0.00, 12/30/11, 09:35:49
567, -62, 0.06, 15091.38, 0.00, 12/30/11, 09:36:04
582, -47, 0.32, 15092.88, 0.00, 12/30/11, 09:36:19
597, -32, 0.55, 15094.37, 0.00, 12/30/11, 09:36:34
```

Ln 1, Col 1

The screenshot shows a software window titled "Data View [DataView1]". At the top, there are buttons for "Edit", "Fill Down", and "Write All New Values". To the right, there is a "View Override" section with a checkbox and two buttons labeled "ON" and "OFF". The main area contains a table with the following columns: "No.", "Address", "Nickname", "Current Value", "New Value", "Write", and "Viewing Format". The table lists 18 rows, with the first 17 rows having data. The "No." column contains numbers from 001 to 018. The "Address" column contains addresses from DS100 to DS116. The "Current Value" column contains various integers. The "Viewing Format" column for all rows is "Integer". At the bottom of the window, there are buttons for "Export", "Close", and "Help".

No.	Address	Nickname	Current Value	New Value	Write	Viewing Format
001	DS100		0			Integer
002	DS101		1011			Integer
003	DS102		100			Integer
004	DS103		119			Integer
005	DS104		1002			Integer
006	DS105		1			Integer
007	DS106		15			Integer
008	DS107		0			Integer
009	DS108		12247			Integer
010	DS109		1			Integer
011	DS110		0			Integer
012	DS111		0			Integer
013	DS112		61			Integer
014	DS113		0			Integer
015	DS114		4			Integer
016	DS115		3			Integer
017	DS116		2511			Integer
018						

DS0100: This is the base memory area defined in **SETUP.TXT**. You may use any value between DS0000 and DS4483 for your base location. The PLC writes values to this register to cause the DF 3 USB to execute certain tasks defined below.

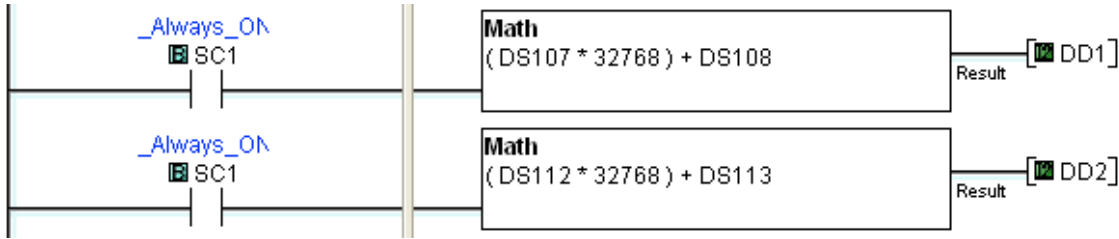
DS0101 (BASE + 1) The DF 3 USB writes status indication values back to this location to show progress and current USB status.

- 1010 is written if no USB drive is present.
- 1011 is written when a new line is written to **DATLOG1.TXT**.
- 1012 is written when a new line is written to **DATLOG2.TXT**.
- 1000 is written when the USB drive is present at power up or insertion only.

DS0102 to **DS0116** (BASE + 2 to BASE + 16) should be treated as read only by the PLC. Only the DF 3 USB should write values to these locations. The PLC should not try to write to these memory areas.

DS0104 to **DS0116** are written to 0 by the DF 3 USB at power up and each time the USB Drive is inserted.

The **DATLOG1 . TXT** file size is still written to BASE + 7 (DS0007, DS0008), however with a Click plc, the file size is written into 2 consecutive memory locations. To calculate the true file size, use the following function:



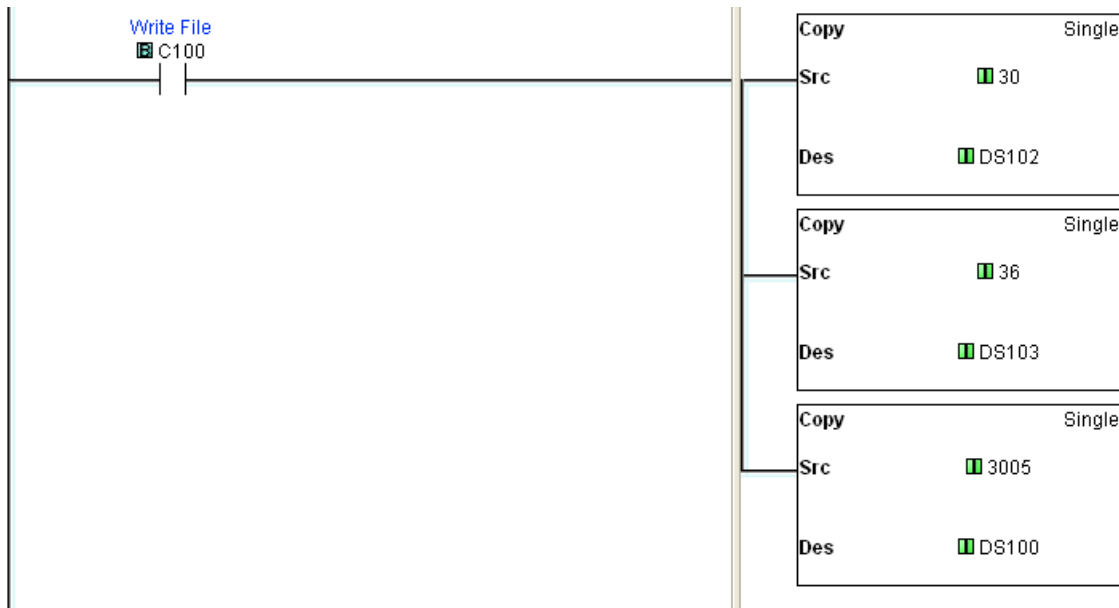
Now DD1 and DD2 will contain the double word value for the true file size of **DATLOG1 . TXT** and **DATLOG2 . TXT**

No.	Address	Nickname	Current Value	New Value	Write	Viewing Format
025						
026	DS30		1			Integer
027	DS31		2			Integer
028	DS32		3			Integer
029	DS33		4			Integer
030	DS34		1234			Integer
031	DS35		-4321			Integer
032	DS36		32767			Integer
033						
034						

```

FILE005.TXT - Notepad
File Edit Format View Help
DS0030= 00001
DS0031= 00002
DS0032= 00003
DS0033= 00004
DS0034= 01234
DS0035=-04321
DS0036= 32767
Ln 1, C
  
```

To create a file called **FILE005 . TXT** and save DS0030 to DS0036, use the following logic:



Make sure C100 is a ONE SHOT. 3005 is the same as C005 (Create a File) with the Click. See section below for more explanation.

The functions that are written to DS0100 are different than for the DirectNet plc. Since the DS values cannot have Hex characters, the thousands digit is used to specify the File I/O function:

DS0100=	USB Function
6xxx	Upload File if it exists from USB to PLC.
5xxx	Check to see if the file Exists on the USB Drive.
4xxx	Delete File from the USB drive if it exists.
3xxx	Create File on the USB and overwrite any file with same name.
200x	Backup Datlog1.txt or Datlog2.txt. (2001 or 2002 only allowed)
1xxx	Append to existing USB file. If it does not exist, the file is created anyway.

XXX Can equal any value from 000 to 999.

The above example wrote 3005 to DS0100 to write **FILE005.TXT**

The DF 3 USB will write back a status value to DS0101.
DS0101 Values:

If DS0100 = 6xxx (Upload File)

- 1000 is written to DS0001 while checking for file
- 1001 if file uploaded from flash ok.
- 1002 if file did not exist
- 1003 if busy uploading the file then will change to 1 if successful

If DS0100 = 5xxx (Check file Exists)

- 1000 is written to DS0001 while checking for file
- 1006 if file does exist
- 1007 if file does not exist

If DS0100 = 4xxx (Delete File)

- 0000 while busy checking for file
- 1008 is written to DS0001 if file was delete successfully
- 1009 if file did not exist

If DS0100 = 3xxx (Create File)

- 1000 is written to DS0001 while checking for file
- 1003 if busy creating the file then will change to
- 1002 if there was an error
- 1001 if successful

If DS0100 = 200x (Backup Datlog1 or Datlog2)

- 1000 is written to DS0001 while checking for file

- 1003 if error creating backup
- 1002 if backup failed because all 26 names used
- 1001 if successful
- DS0016 or DS0017 will increment
- DS0007 or DS0014 will change to new file size (smaller)

If DS0100 = 1xxx (Append to File)

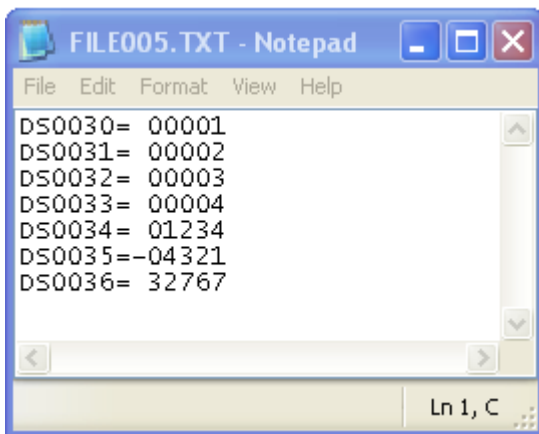
- 1000 is written to DS0001 while checking for file
- 1004 if busy appending to the file then will change to
- 1002 if there was an error
- 1001 if successful

DS0002 Values: Written by PLC

The start address for the Data you want to save or append

DS0003 Values: Written by PLC

The end address for the Data you want to save or append



```

FILE005.TXT - Notepad
File Edit Format View Help
DS0030= 00001
DS0031= 00002
DS0032= 00003
DS0033= 00004
DS0034= 01234
DS0035=-04321
DS0036= 32767
Ln 1, C

```

If you are creating FILEXXX.TXT with another application or manually typing in the data to be uploaded to the plc, make sure each line is 13 characters wide and each line has a CARRIAGE RETURN/LINE FEED at the end. After the "=" sign, either a space, " ", "-", or "0" must be entered. The number must be between -32,767 and 32,767

Using the DF3USB for serial PRINT logging

The DF3USB can accept serial data for logging and file creation. In this mode, any file name can be saved to the USB drive. The file must follow the 8.3 convention for file naming:

ABCDEF . TXT
12345678 . CSV

The USB drive that is inserted into the DF3USB should not contain the SETUP.TXT file or DATLOG1.TXT or DATLOG2.TXT. When these file are missing, the DF3USB automatically sets itself to accept serial print data. The baud rate is also set to 9600,8,N,1. Notice it is reset to NO PARITY.

The following is a ladder rung example of how to open a file and write data to the file:

15	C2 ┌ ─┐ └ ─┘	<pre> PRINT K2 "W\$RFILE1.TXT\$RHERE IS THE GUTS OF THE FILE.\$NBYTES WRITTEN: " V2002:B "\$R.\$R" </pre>
----	--------------------	--

Breaking this PRINT statement up:

W or w means Write to File.

\$RFILE1.TXT\$R Will write to FILE1.TXT

(text will be written to file like the following. \$N inserted a CR/LF)

```

HERE IS THE GUTS OF THE FILE
BYTES WRITTEN 0053

```

\$R. \$R (This must be written to close the file and wait for next string)

\$R. \$R will close the file and add a CR/LF to the end of the file

\$R:\$R will close the file only. The next write will append directly behind the last write.

The PRINT rule is only 128 characters can be sent in one print statement.

The DF3USB will respond with a 16 character ASCII string that indicates if the thumb drive is attached and how many bytes were written and the current file size:

Ex:

```

USB>000000000000 (USB> means USB drive is inserted no data written)
USB!000000000000 (USB! means USB drive was removed)
USB>003400000455 (USB> means usb is inserted, 34 bytes were written
and the file size is 455 bytes)

```

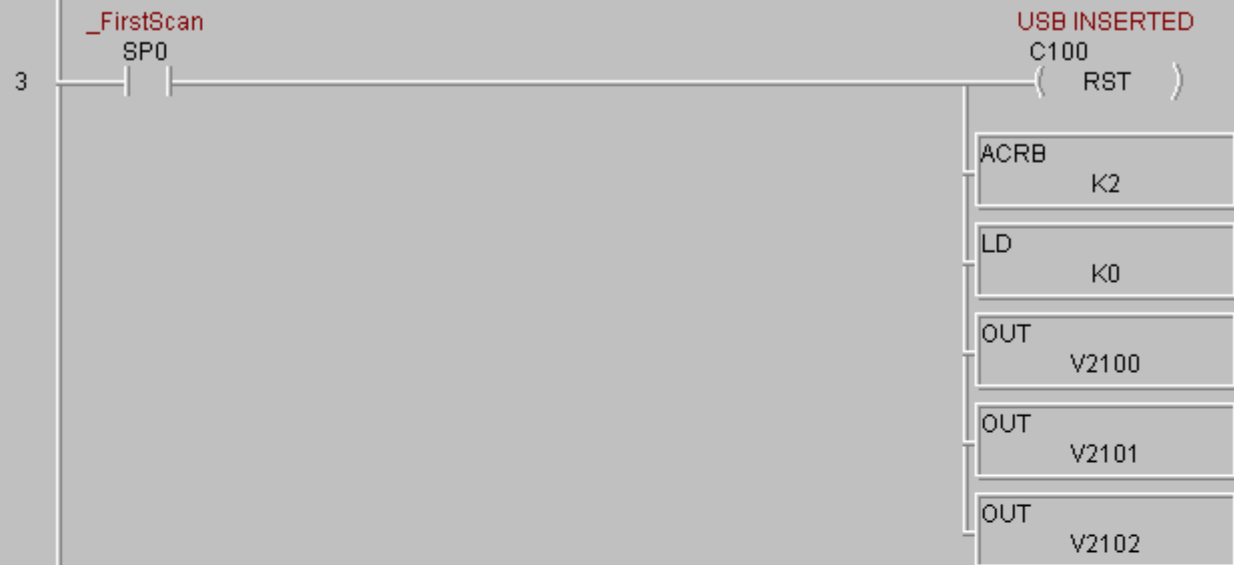
The following is a set of rungs you can use to extract the file size and bytes written.

```

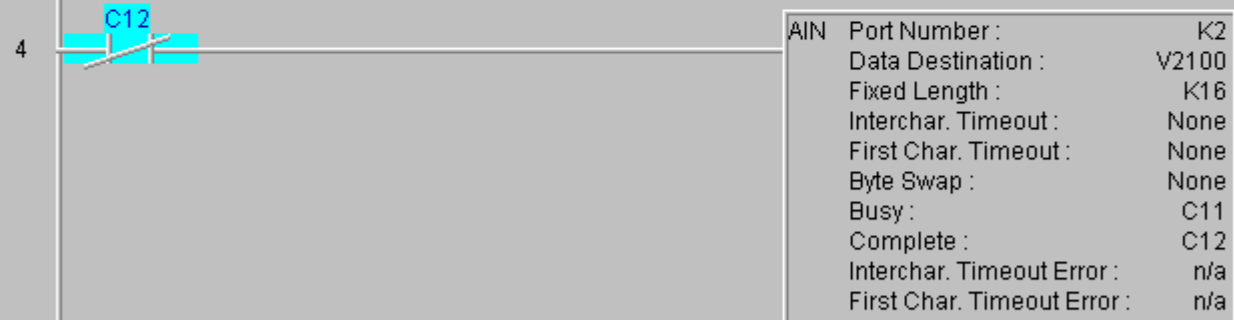
V2020 = bytes written (Single Word BCD)
V2021 = file size (Double Word BCD)
C100 = USB Inserted latch/unlatch bit.

```


ON FIRST SCAN,
 RESET USB INSERTED C100, CLEAR K2 BUFFER AND AIN DESTINATION FIRST 3
 ADDRESSES.

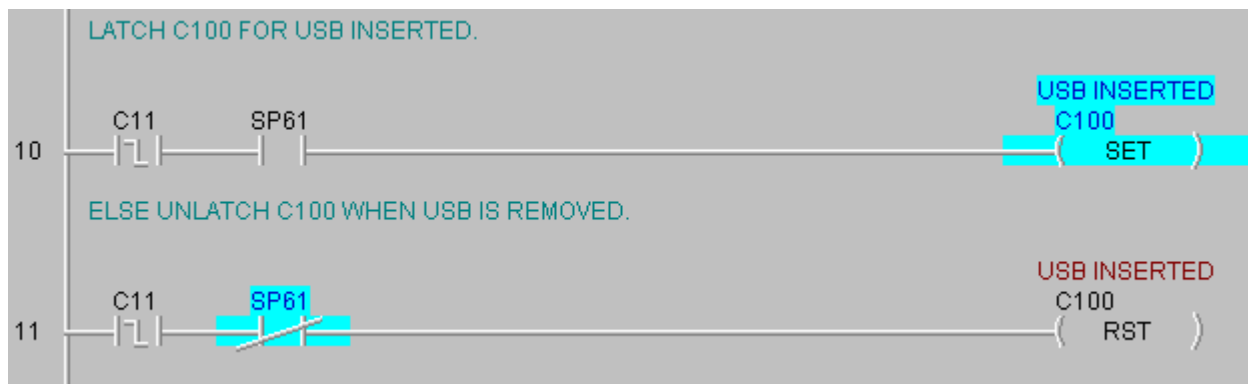
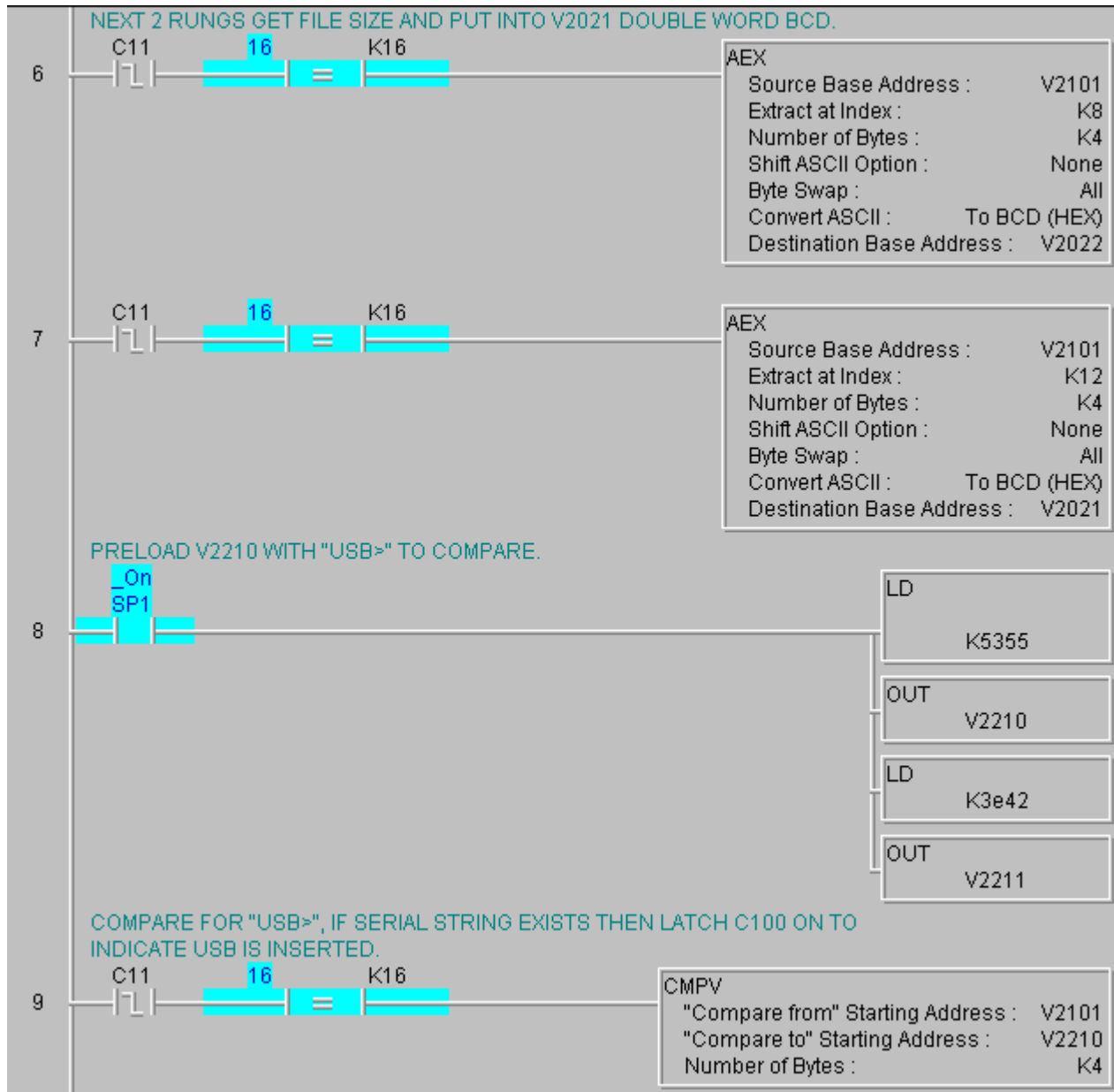


SETUP TO RECEIVE 16 CHARACTERS FROM DF3USB AND PUT CHARACTERS IN
 V2100. USB!0000000000000000 MEANS USB DRIVE WAS REMOVED.
 USB>001200001234 MEANS USB IS INSERTED (USB>) 12 BYTES WERE WRITTEN
 TO THE FILE AND THE FILE SIZE IS 1234 BYTES.



EXTRACT THE NUMBER OF BYTES WRITTEN AND PUT INTO V2020 SINGLE
 WORD BCD.





To set the time so the file saved has a current time/date stamp, send the following during a write to the file.



This sends the current clock settings as ASCII text to the DF3USB. Send all 7 registers.

The plc will send:

\$T0012003400100004000300040012\$T

The DF3USB will extract the date and time out of this string and then set the file date and time accordingly. If no date/time string is sent, then the default date/time of the file will be

01/01/2012 12:00:00 PM

The data between \$T...\$T will not be written to the file. If you truly want to send a TAB, the just send \$T and 5 spaces will be saved to the open file.

BE CAREFUL NOT TO SEND **\$T (28character) \$T** or the system may setup an erroneous date/time for time stamping.