

TechNote #33:

Model 5300 Data Logging

The following QuickBuilder programming example utilizes variant registers to create a single row data table with a header. By activating input 1 (DT_On) a single row, 5 columns wide will be filled with either 1's or 0's depending on the state of input 3 (DT_Write). The test unit used with a similar program had a task timer variable defined to measure the time it takes to write a single row of data consisting of 80 elements. Timing stayed consistent at 4ms per 80-element row.

After writing the data to the data table, the program then creates a log and stores it to the directory RAMDISK/DataStore on the 5300 controller in a file named Log001.log. To examine the data within the log file, simply access the 5300 controller via an FTP session then view the file with any standard text editor or a Comma-Separated Variable (CSV) file type compatible program.

Before running the program, you must copy the "Startup.ini" Script file to the "Scripts" directory of the "_system" folder of the 5300 controller. The script file creates the virtual disk for storing the data logging file. For more information on scripts refer CTC Script Language Guide, document no. [951-520003](#). Once you copy the Startup.ini file, be sure to power cycle the 5300 controller to create the new virtual disk.

Startup.ini Script:

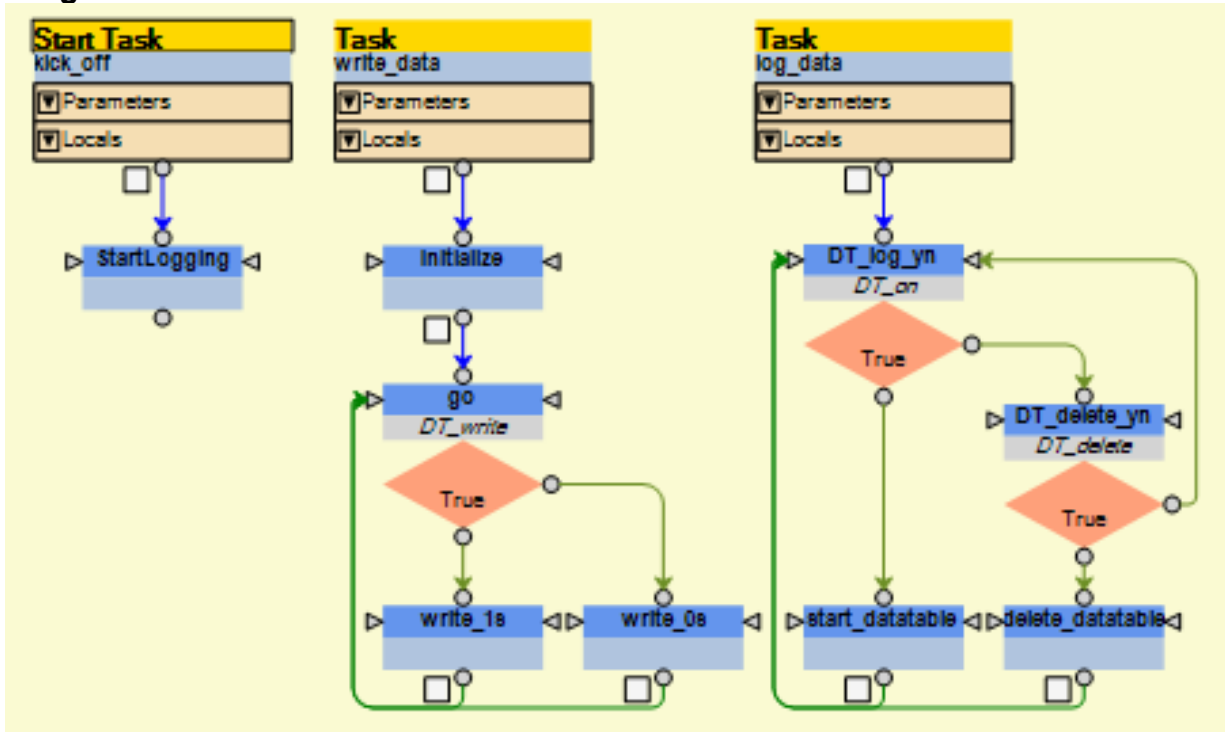
```
#create and mount the virtual disk for data logging
mkdir /RAMDISK/DataStore
mount "/_system/Messages/data" "/RAMDISK/DataStore"

#Delete Log001.log, if exist.
12328=1

#Perform data logging for record #1 in the log.ini file.
#Write to Log001.log file. This begins session. Any additional
12326=1

#logging will be appended to file.
12325=1
```

Program Overview:

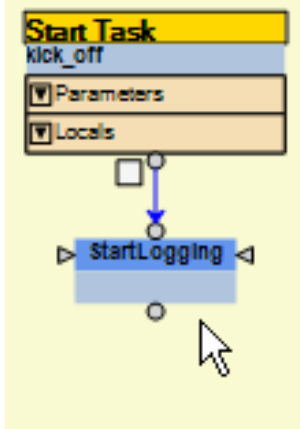


The program consists of three tasks:

1. The Start Task starts the “write_data” and “log_data” tasks.
2. The “write_data” Task initializes the log and sets up the log header information. It is then determined whether to write 1’s or zeros based on the state of digital input 3 (DT_write).
3. The “log_data” Task starts or stops the data logging routine based on the state of digital input 1 (DT_on). If digital input 2 (DT_delete) is active the data log is deleted.

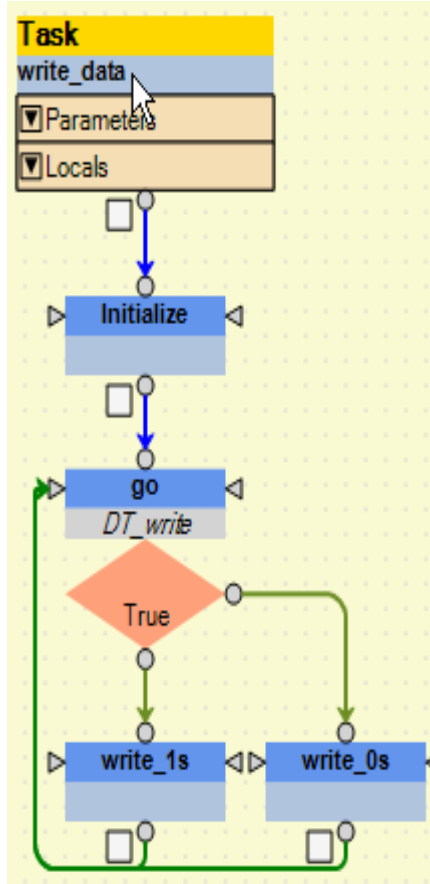
The following pages examine the QuickBuilder programming instructions contained within each step of the program and can be copied and pasted into your QuickBuilder program.

Start Task: kick_off



```
//Step- startLogging:
do (write_data log_data);
```

Task: write_data:



/*Step- Initialize:

On start up, the program will delete Log001.log and creates a new file. It will then write the header (Row 1) to the log file in preparation for data to be written to it.*/

```

store 1 to DelLog_12328;
store 1001 to log_execute_r12325;
DT_Header[0][0]=string("msTimer");
DT_Header[0][1]=string("C2");
DT_Header[0][2]=string("C3");
DT_Header[0][3]=string("C4");
DT_Header[0][4]=string("C5");
store addr(DT_Header) to log_string_xfer_r12326;
delay 1000 ms;
  
```

//Step- write_1s

```

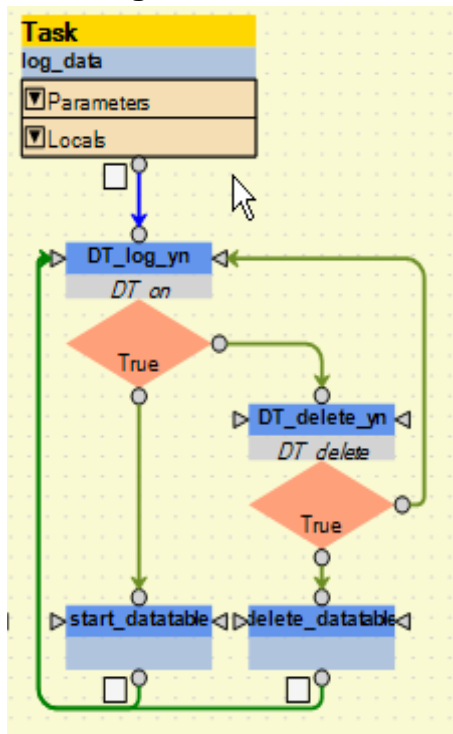
store MSTMR to TableOfData[0][0];
store 1 to TableOfData[0][1];
store 1 to TableOfData[0][2];
store 1 to TableOfData[0][3];
store 1 to TableOfData[0][4];
  
```

//Step- write_0s

```

store MSTMR to TableOfData[0][0];
store 0 to TableOfData[0][1];
store 0 to TableOfData[0][2];
store 0 to TableOfData[0][3];
store 0 to TableOfData[0][4];
  
```

Task: log_data:



/*Step- start_datatable

Start_Log- 12325 select the log file to write. 12326 select the Variant Register to write into the log file selected by r12325. Monitor the switch to be deactivated; or after one second, loop back to repeat.*/

```

store 1001 to log_execute_r12325;
store addr(TableOfData) to log_string_xfer_r12326;
delay 1000ms;
  
```

//Step- delete_datatable

Delete_log - Once switch is activated, the program will delete Log001.log and then create a new file. It will then write the header (Row 1) to the log file in preparation for data to be written to it.*/

```

store 1 to DelLog_12328;
store 1001 to log_execute_r12325;
DT_Header[0][0]=string("msTimer");
DT_Header[0][1]=string("C2");
DT_Header[0][2]=string("C3");
DT_Header[0][3]=string("C4");
DT_Header[0][4]=string("C5");
store addr(DT_Header) to log_string_xfer_r12326;
delay 1000 ms;
  
```

If there are questions regarding the functionality of this example or data logging, in general you can email <mailto:customersupport@ctc-control.com> or call 1- 800.282.5008