dataTaker

dataTaker DT800 Range Code Examples

Updated 13/7/2020

ThermoFisher SCIENTIFIC

Table of Contents

Introduction

Example Programs are working examples produced with dataTaker 800 series loggers that can be downloaded and utilized as a starting point for your own projects or applications.

Hardware: DT800

For more information visit **thermofisher.com/datataker**

DT800 low power

This code allows the DT800 to wake up on from sleep mode when main power is turned on. A typical application is in the automotive industry. When the ignition key is turned on, the DT800 wakes up and starts logging. When the key is turned off, the DT800 powers down into sleep mode.

Note: The same can be achieved with the DT5xx and the DT8x by pulling the “Wake” terminal to ground.

Wiring: The + power is connected to the CT of the serial sensor port and to digital input 1.

```
BEGIN”Low_Pwr”
‘Parameter declarations
P3=2             ‘Set minimum sleep period to 2 seconds
P15=1            ‘Force low power operation
P17=5            ‘Set delay to low power mode to 5 seconds
RA”Sched_A”1E    ‘Trigger schedule A on either a rising or falling edge of digital input 1
   ALARM1(1DS>0.5/5S){[LOGONB GB /K]}      ‘If 1DS is high, power is on. Start schedules running.
   ALARM2(1DS<0.5){[LOGOFFB HB /k]}        ‘If 1DS is low, Powr is off. Stop schedules running.
RB1S
        ‘Put your program stuff here and / or following schedules.
END     ‘End of program
```

Working with GPS

GPS string handling DT800

This is an example of code using a Garmin GPS18 5Hz with a DT800. The default communications rate for this device is 19,200 instead of the usual 4800 baud.

- A 5 V regulator is available to power the GPS
- 12 V supply is tied to digital 8 i.e. when 12 V is available the logging starts and when the 12 V is removed the logging stops
- Heading and mps speed measurements are only made when the speed is greater than 42 CV

(Not syntax checked for a DT800 yet)

```
BEGIN”GPSDT800”
PS=19200,N,8,1,NOFC
SATTN;
42CV(“Spd Lmt”,W)=1
‘ Turn off the unnecessary GPS messages
1SERIAL(“{$PGRMO,GPRMC,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,GPGGA,1\\013\\010}”,W)
1SERIAL(“{$PGRMO,GPGSA,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,GPGSV,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,PGRME,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,GPGLL,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,GPVTG,1\\013\\010}”,W)
1SERIAL(“{$PGRMO,PGRMV,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,PGRMF,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,PGRMB,0\\013\\010}”,W)
1SERIAL(“{$PGRMO,PGRMT,0\\013\\010}”,W)
DELAY(W)=1000
RA”Report”200T  LOGONA GA
 1SERIAL(RS232,”$GPGGA,%f[10CV],%2f[11CV]%f[12CV],%1s[‘N’,’S’,43CV=-1],%3f[13CV]%f[14CV],%1s[‘E’,’W’,44CV=-
1],%f[30CV]”,=99CV,.1,W)
 IF(43CV><1,2){11CV(W)=-11CV}
 IF(44CV><1,2){13CV(W)=-13CV}
 IF(30CV>1){1SERIAL(RS232,”,%f[9CV],%f[16CV],%f[15CV],%14s[1$]$GPVTG,%f[17CV],T,%f[8CV],M,%f[7CV],N,%f[18CV]”,=99CV,.1,W)}
 ALARM(30CV>1){CATTN}
 ALARM(30CV><0,1){SATTN 16CV(W)=-1 15CV(W)=0 17CV(W)=0 8CV(W)=0 18CV(W)=0}
 1SERIAL(RS232,”\\e”,=99CV,.1,W)
 34CV(“TempSpd”,=35CV,W)=(18CV>42CV)*18CV*1000/3600
 35CV(“m”,IB,+=22CV,W)
 10CV(“GPSTime”)
 11CV(“LatD~Deg”,FF0) 12CV(“LatM~Min”,FF7)
 13CV(“LonD~Deg”,FF0) 14CV(“LonM~Min”,FF7)
 15CV(“Alt~m”)
 16CV(“HErr~m”,FF2)
 30CV(“GPS State”,FF0)
 27CV(“Head~deg”)=(18CV>42CV)*17CV+(18CV<=42CV)*27CV
 18CV(“Spd~kph”)
 20CV(“Spd~mps”,=21CV)=(18CV>42CV)*18CV*1000/3600
 21CV(“m”,IB,+=23CV,W)
 23CV(“Trip~km”,.001,FF4)
RD”GoHalt”8E LOGOFFD
 IF(8DS<0.5){HA}
 IF(8DS>0.5){GA}
END
```

Turning point analysis

This code read a wave form, extracts the maxima and minima, and saves the value of the maxima/minima and time. It also includes noise rejection.

```
BEGIN"TP"
'------------------------------------------------------------------------
'       Turning Point Analysis Routine for DT800
'
'               DataTaker Technical Support 6 Jan 2004
'               support@datataker.com.au
'
'       This code logs the turning points of any wave form.
'       The time of turning is recorded logged in 4CV.
'
'       Notes:  4CV holds the time since midnight in seconds.
'               This limits the time accuracy to 2 decimal places.
'
'------------------------------------------------------------------------
7CV(W)=10  '7CV hold the dead band for noise rejection
  'Any signal less that the current turning point
  '+/- 7CV will be rejected.
8..9CV(W)=0    'Minimum noise level and Maximum noise level respectivly.
10CV(W)=0    'Holds last turning point. Used for noise rejection.
'
' Schedule A is where the turning points are actually logged.
' Note: The X schedule must be before the fast schedule.
'
RAX LOGONA
   2CV("TP ~mV",FF7)
   4CV("Time ~Sec",FF7)
'
' Schedule B detects the turning point
'
RB,FAST
   2CV(W)=1CV        'Shift register for reading
   4CV(W)=3CV        'Shift register for time
   1V(=1CV,GL20V,W)    'Read Current value (Note: Gain lock to suit)
   T(=3CV,W)    'Read current time
   1CV(DF,=5CV,W)    'Read differance between readings
     'This boolean in the next bit of code returns 1 if the current reading is greater than the last
     '(Rising signal) and returns -1 if the current reading is less than the last (Falling signal)
     'By taking the differance (DF) 6CV holds 2 maxima turning point or -2 for minima turning point.
   6CV(DF,=6CV,W)=-1*(5CV<0)+(5CV>0)
     'Calculate Lower noise tollerance.
   8CV(W)=10CV-7CV
     'Calculate Upper noise tollerance.
   9CV(W)=10CV+7CV
     'Then check for turning point and noise then save turning point if valid.
IF(6CV<>-0.5,0.5)AND    'If a turning point AND
IF(1CV<>8CV,9CV){[10CV=2CV XA]}    'If the current signal is outside the noise dead band THEN
'Record new turning point and Log turning point data
END
```

DT800 burst mode, SMSX example

The following code will trigger a burst and store it if a particular measurement alarm is triggered. If the memory card becomes too full, an SMS message is sent to warn the operator. The program shows how a multi-line SMSX message can be structured.

```
BEGIN”DMM3A”
CATTN
‘Scalling
S1=0,200,0,5000”psi”
S2=0,15,0,5000”psia”
S3=0,5000,0,5000”psi”
P31=3
‘Global declarations
‘ Set the threshold for the Alarm SMS messages
10CV(“Alarm Memory”,W)=50
‘ Set the threshold for the FW
11CV(“Alarm PU FW “,W)=100
‘ Set the threashold for the Comp
12CV(“Alarm PU Comp”,W)=100
‘ Set an initial value for free memory available
33CV(“Free Memory”,W)=-1
 ‘schedule definition
‘ Run the A schedule while 1 is high (the key switch is on
RA5S:1W LOGONA  GA
 13CV(“Mag PU FW”)
 14CV(“Mag PU Comp”)
 3TJ(“Disch Temp”,=15CV)
 5V(S1,”Disch Press”,=16CV)
 5*V(S2,”Inlet Vac”,=17CV)
 6V(S3,”Feed Press”,=18CV)
 6*V(S3,”Rot Press”,=19CV)
 3*TJ(“Ambient”,=20CV)
 4TJ(“H Temp1”,=21CV)
 4*TJ(“H Temp2”,=22CV)
‘ Pole the Serial Gateway device for the appropriate measurements. If it times out set the value to -9999999
 1SERIAL(RS232,”\\e{RP 1 \\013}%f[102CV]\\010”,=103CV,.025,W)
 ALARMR(103CV><20,21){[102CV=-9999999]}
 1SERIAL(RS232,”\\e{RP 2 \\013}%f[104CV]\\010”,=105CV,.025,W)
 ALARMR(105CV><20,21){[104CV=-9999999]}
 1SERIAL(RS232,”\\e{RP 3 \\013}%f[106CV]\\010”,=107CV,.025,W)
 ALARMR(107CV><20,21){[106CV=-9999999]}
 1SERIAL(RS232,”\\e{RP 4 \\013}%f[108CV]\\010”,=109CV,.025,W)
 ALARMR(109CV><20,21){[108CV=-9999999]}
 ‘ Read the card memory available
 3SV(W,=30CV)
‘ Read the used memory
 4SV(W,=31CV)
 32CV(“Total Memory”,W)=30CV+31CV
‘ If there is a memory card inserted calc the PC used
 ALARM(32CV(“Memory Insert”)>1.){[33CV(W)=100*30CV/32CV]}
‘ If it is not installed set the value to -1
 ALARM(32CV(“No Memory”)<1.){[33CV(W)=-1]}
‘ Check the memory alarm send a message after one minute
 33CV(“Mem Ava”)
 ALARM(33CV<10CV/1M){[100CV(W)=1 X]}
```

```
 ALARM(33CV<10CV/1D){[100CV(W)=1 X]} ‘ one day
 ALARM(33CV<10CV/2D){[100CV(W)=1 X]} ‘ two days
 ALARM(33CV<10CV/3D){[100CV(W)=1 X]} ‘ Three Days
‘schedule definition
RB200T:1W LOGOFFB  GB
 1VAC(“Mag PU FW”,200,NS10000,=13CV,GL1V,W)
 2VAC(“Mag PU Comp”,200,NS10000,=14CV,GL1V,W)
 ALARMR(13CV>11CV){[XK]}
 ALARMR(14CV>12CV){[XK]}
‘Burst Data
RKX;BURST(32000,2000) LOGOFFK  GK
 1VNC(“Mag PU FW (DC)”,NR,GL1V)
 2VNC(“Mag PU Comp (DC”,NR,GL1V)
‘SMS message
RX LOGONX
 DELAY(W)=5000
 DO{[Q /e/Z/m/r]}
 DELAY(W)=1000
 DO’ET-SMS=0|’
 DO’EDD72’
 ALARMR(1DS<1)’^JNot Running’
 DO’^JCard Mem=?33F1^J’
 ALARMR(1DS>1)’DT=?15F1^J’
 ALARMR(1DS>1)’DP=?16F1^J’
 ALARMR(1DS>1)’IV=?17F1^J’
 ALARMR(1DS>1)’FP=?18F1^J’
 ALARMR(1DS>1)’RP=?19F1^J’
 ALARMR(1DS>1)’HT1=?21F1^J’
 ALARMR(1DS>1)’HT2=?22F1^J’
 ALARMR(100CV>1)’|P1|90^M’
 ALARMR(100CV<1)’|P0|90^M’
 100CV(W)=0
END
```

Thermocouple failure detection

Setting a pre-determined level for thermocouple output

This code will display a preset value if the resistance of a thermocouple goes Open circuit (over 100 Ohms.) (DT800 Only)

```
BEGIN”JOB1”

1CV=1000 ‘Displayed fail value.

RA10S LOGONA GA
  1R(2W,W,=2CV)    ‘read thermocouple resistance
  1TK(=3CV,W)      ‘read thermocouple temperature

 ‘If Ok display temperaure, Failed display fail value
  4CV(“Temperature~DegC)=3CV*(2CV<100)+1CV*(2CV>=100)
END
```

Using the Beeper and the Warning light to send a coded warning DT800 code

Operators and researchers frequently need to be alerted when conditions fall outside the desired parameters they are monitoring. The DT800 can be programmed to respond to alarms by creating a series of audio and visual warnings.

The DT800 provides two methods of operator feedback in its standard form. The first is the attention LED and the second is the beeper. These can be accessed using the 1WARN=1 to turn the beeper on and 1WARN=0 to turn the beeper off and 2WARN=1 and 2WARN=0 to turn on and off the attention LED. We can use both the beeper and the attention led to indicate a problem. However, if you wish to convey something of the nature of a problem to an operator you could provide a coded message from which an operator can determine a fault code. The beeps and flashes could indicate a particular fault code in the form of a series of long beeps/flashes and short beeps/flashes.

This code example uses a long beep to indicate tens and a short beep to indicate units. For example, long, long, short, short, short would represent 23, or short, short, short, short, short represents 5.

The following example checks that the connected thermocouple is not open circuit using a resistance measurement. A standard thermocouple measurement is subsequently made. If the resistance measurement is greater than 100 Ohms, it is considered that the thermocouple is open circuit.

Note: Because the resistance measurement is in an alarm statement without a number, it is not logged. An alarm is triggered and channel variable 50 is set to an appropriate value. Schedule K is programmed such that when 50 CV is not zero. This schedule will execute at half second intervals, sounding the appropriate number of flashes and beeps.

```
BEGIN”JOB1”
  RA10S LOGONA GA
   ALARMR(1R(2W)>100){[50CV(W)=1]}
   ALARMR(2R(2W)>100){[50CV(W)=2]}
   ALARMR(3R(2W)>100){[50CV(W)=3]}
   1TK
   2TK
   3TK
  RK500T:50CV LOGONK GK
   ALARMR(50CV><0.,10.){[1WARN=1 2WARN=1 DELAY=20 1WARN=0 2WARN=0 50CV(W)=50CV-1]}
   ALARMR(50CV<0.){[1WARN=1 2WARN=1 DELAY=50 1WARN=0 2WARN=0 50CV(W)=0]}
   ALARMR(50CV>10.){[1WARN=1 2WARN=1 DELAY=200 1WARN=0 2WARN=0 50CV(W)=50CV-10]}
END
```

In Australia:
For customer service, call 1300-735-292
To email an order, ordersau@thermofisher.com

In New Zealand:
For customer service, call 0800-933-966
To email an order, ordersnz@thermofisher.com

Find out more at **thermofisher.com/datataker**

ThermoFisher SCIENTIFIC