The AWSON Well Firmware
for the
ARDUINO Pro-Mini

```
/ --------------   Ray/Godfrey Please see line  #777  to enter new SIM PIN entry  -----------------------------------------------------------------

// FIRST type in the SIM-PIN code of the SIM card YOU are using by doing the below......
// Find:      String pin = "   ";    way down below and enter your four numbers between the  "   " to YOUR four SIM-PIN numbers say "5678" keeping the "
" etc all the same.

// Please take care not to change anything else.
// Look between the double hash lines #############################
//                                    #############################
// see ROW 777 below
// it will be SIMILAR to the handfull of ROWS below... Good luck . . .
/*
//####################################################################################################################################################

  String pin = "3488";                                        // Please just change the four numbers ONLY.
                                                              // Please leave all the quotation marks, semi-colon - thanks and good luck!

//####################################################################################################################################################
*/

// SECOND, and only if necessary,  change just the           "AWSOM#xxxx         number on row  535  see EXAMPLE below (row 26) to the number supplied by
Andrew  +447549871528

// ------------------------------------- sprintf(textMAXtemp,"AWSOM#3456 MxTmp = %d.%00d:", (int)temperatureMax, abs((int)(temperatureMax * 100) % 100));
// Thanks  Andrew
```

```
/*
This software is licensed under Creative Commons BY-SA 4.0, http://creativecommons.org/licenses/by-sa/4.0/ International (CC BY-SA 4.0)
This is a human-readable summary of (and not a substitute for) the license.

You are free to:
Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material for any purpose, even commercially.
The licensor cannot revoke these freedoms as long as you follow the license terms.
Under the following terms:
under Creative Commons BY-SA 4.0, http://creativecommons.org/licenses/by-sa/4.0/ International (CC BY-SA 4.0)

No warranties are given. The license may not give you all of the permissions necessary for your intended use.
For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   Main code by Andrew WS Ainger BSc. CEng. FIET.
   Significant support from:
   Maurice Smith BSc. MBE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*/
//________________________________________________________________________________________________________________________

//    SYSTEM OVERVIEW
//    WHEN USED WITH THE AWSOM WELL HARDWARE THIS SKETCH SENDS A TEXT AND PRINTS TO THE SERIAL USB PRINTOUT THE:

//       THE Max-Temperature (NB: Resets after EVERY TEXT)
//       THE NiMH-Battery voltage (an instantaneous reading at the time of the TEXT)(+/-0.1v)
//       THE PUMP-COUNT (CUMULATIVE) in activity-units of 30 secs (+/-4 %).
//       THE WORK-DAY (CUMULATIVE, number of days or 'bright-times', the unit has been used since last RESET)
//       THE Powered-Time between TEXTs  ( The number of 30second blocks the unit has been used just that day/bright-time)
//
//     NB:Use OLD BOOTLOADER 'ARDUINO NANO' for Arduino Mini-Pro from low-cost overseas suppliers!
//     Also, When uploading sketch make sure Vcc on Arduino is not over 5v
//    Thanks & good luck
//    Andrew  G1ZYJ
//__________________________________________________________________________________________________________________________

//  This sketch can be used with the AWSOM configured hardware.
//  However, there are several items that are recomended to be adjusted for each Unit as below:
//  See between lines marked ////////////////////////////////////////////////////////////////////////////////////////////////////////////

//  CURRENTLY SET UP FOR UNIT NUMBER:- see row 540-ish
// 1 The Phone numbers you want it to text to. (about lines #68+ below)
// 2 The versionNumber of the AWSOM sketch (about line #97 below)
// 3 The sleepTimeSLOWLOOP should be calibrated to make total loop take 30secs. +/-250ms (about line #167)
// 4 The Well ID, really just a unique number in each Arduino-Pro-mini, and now a sequential number starting from #3887
// 5 See textMAXtemp on about line 460 to carefully enter WELL ID# "xxxx")
// 6 The TEMPERATURE CALIBRATION: See main-loop circa 200ish. Other variables are optional.
//

///////////////////////////////////   below: Numbers   /////////////////////////////////////////////////////////////////////////////////////
//
//------------------------- USER INPUT HERE (change NUMBERS ONLY please) --------------------------------------------------
//                          (If uncertain please leave all settings as-is)
//
//  ENTER the MOBILE number you want the WELL to TEXT you on. (Change the numbers ONLY please)

int andrew =         "+447549871xxx"   ;               // For Andrew's phone "+44754xxxxx"
int baseStation =    "+447399243xxx"   ;               // AAs new base station SIM card #3882 "+447399243x"
int mike =           "+447781136xxx"   ;               // Mike's "+447781136994"
int godfrey =        "+255762603xxx"   ;               // check number as was "+2557447xxxx"  and was "0762603xxx"
int ray =            "+255755760xxx"   ;               // check number "+255755xxx"
int maurice =        "+447788952xxx"   ;               // Maurice's mobile used in uk tests

int SMSfrequency =  1                  ;               // This is normally set to 1 so that there is an SMSs every 1 night. (Set TO 7 for a WEEKLY Text)
char versionNumber[] = "v319"          ;               // Put sketch ID here so it prints out in the USB printout as a reference
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
// Save this sketch as eg AWSOMsketch318-3887 (sketch version 318 and it's for Mino-Pro computer in Tanzania #3887 )
// NB The WELL ID# eg "xxxx"  (NB NO " = " SIGN remember, as it messes up auto analysis) see about LINE #462
// is to record in the text message as it WAS the last 4 digits of the Well Mobile SIM card for easy ID when forwarding BUT NOW ITS THE ARDUINO PINO-PRO
NUMBER for the AWSOM well unit!

// That's all the 'input' required here. Well done!

//////////////////////////////////above: phone &  versionNumbers ////////////////////////////////////////////////////////////////////////////

// We now have to load this software onto the AWSOM WELL MAINTENANCE UNIT so now:-
// First, unplug the green solar and battery plugs in the AWSOM unit then
// Make sure you have the Arduino IDE installed on your PC and Plug in the 4 wires: (Gnd, Vcc, Rx, Tx) of the WELL MAINTENANCE UNIT to your PC via the
Converter, then
// select the TOOLS  menu by going upto just ABOVE the GREEN-BLUE bar, (above-left) and
// then left-click and slect BOARD and then go right and down and select "PRO-MINI" or, if that doesnt load, try "ARDUINO NANO". (yes, nano) Good, now,
// Select TOOLS again and select PORT: (NB: The microcomputer must be plugged in at this stage or it will be Greyed-Out) and
// select the active COM Port number (normally the one at the bottom of the list - eg: COM3 )

// Okay, now go FILE and SAVE-AS and rename this file to your first name followed by a number (eg: 1).
// Now go up to the top blue-green bar and left-click on the only ARROW 'pointing to the RIGHT'  This will start to load the software onto the
// AWSOM WELL UNIT but as soon as you see the word 'Uploading' on the BOTTOM BLUE-GREEN bar (just above the bottom black band) IMMEDIATELY press
// the RESET button on the Arduino Nano. (if you miss it just wait a few mins, press the right ARROW AGAIN PRESSING THE RESET BUTTON ON
// THE ARDUINO AGAIN ) THE BLUE-GREEN BAR should SAY, eventually, 'Done uploading'
// That's it, all finished, now you can test and install your Well AWSOM unit.

// NOTE:
// Remember, if you want to be informed as to what this little AWSOM unit is doing as it steps through its program, click on the
// small 'magnifying-glass' icon near the top-RIGHT of this window and a new window will appear on your screen, just close it down when you have
// finished.

// Well done.
// Many thanks & good luck.
// Andrew
```

```
//------------------------------------------------------------------------------------------------------------------------
/*   This AWSOM WELL UNIT sends you a TEXT which shows:

     1. How much TIME (called the PUMP-COUNT) the Well has been active, (The Count is in blocks of 30 secs) it
     2. Sends the TEXT every setnumber of 'DAYS' ('Dark-Times') or, if you want, a set number (>120) which will be in units of 30sec time periods
     3. It also sends some engineering data.

    The INPUTS required are:

    (1) The mobile number to send the texts to.
    (2) The number of 'DAYS' (or more accurately, Dark-Times) between each TEXT (say fron 1 up to  99 - normally set to 1, = every day)
    (3) Remember, you should make COMMENTS  and enter them into the bar charts of the AWSOM SPREADSHEET

    The software below (is called a 'sketch') and after checking its own battery voltage and setting itself up, the AWSOM unit starts the DAY-COUNT in
    the suntimeCountFunction.  This suntimeCountFunction then opens the PUMP-COUNT (ie the clunkCountFunction) and it increments cumulatively for every
30
    seconds the Well has activity around it. This gives the total PUMP-COUNT the Well has been ACTIVE. Then, after a number of SuntimeCounts, ('DAYS'),
it
    sends an SMS TEXT at DUSK (at the SMSfrequency) of the totalised PUMP-COUNT together with other engineering data.

    This 'other data' shows the Voltage of the battery and the
    maximum temperature the AWSOM Well Maintenance Unit has reached for the period since the LAST text/SMS.
    The battery voltage must be between 4.60 volts and 5.90 volts. If below 4.60v the Unit will turn itself off and recharge itself for about 4 hours
    and then it will try starting again all by itself.

    The battery measurements are +/- 0.15v so care should be taken when interpertating the battery voltage readings from the SMS messages.
    If outside these voltages for several days the solar cells should be cleaned and the battery should be checked & recharged (NB: it should be replaced
    every 4 or 5 years or so anyway) The temperature  measurements are within a couple or degrees or so. The main issue here is if the max temperature is
    over 50 Centegrade for several hours on a regular basis this could degrade or degrade the unit's components.


    When the SMS has been sent, the Powered-time is reset BUT the PUMP-COUNT (or clunkCount)is NOT, it keeps incrementing until the MAXIMUM value of
999,998
```

```
    is reached when it is recomended to press the RESET button (about every 24 months). When this maxclunkCount is reached the unit needs a manual reset
    (recommended). This resets the Well PUMP-COUNT ready to start again. No other User action is required.

    If large black clouds obscure the sun  for about >30 Days the battry will go flat. This is OK as the Unit will charge itself up next time it sees the
sun.
    If there are many passing dark clouds especially at dawn or at dusk then then the number of texts sent will increase as
    a large dark cloud at this time can be interpreted by the Unit as a short night-time! (The Arduino computer is only small and not very smart is it?)
Anyway, it
    then waits for about 8 mins sleep between text-batches plus 22.5 mins (45 loop counts) whilst, during the 45 loop cycles only, still counting water
activity.

    The PUMP-COUNT numbers from the TEXTS should be entered into the spreadsheet (even if a '1' is sent) as the spreadsheet will adjust the totals
automatically.

    (Just as a test, Pin 13 is sent HIGH for 150 milli-seconds each time around the main loop of the sketch, (this is about every 30secs). If necessary
    Pin13 can be measured with a multimeter.
    Also, if using an Arduino Mini v5, the Serial port can be connected at ANY time as this action does NOT reset the counters.)

    And finally, if you want a TEXT every hour (+-3%) then replace the current number of timeTxtWanted (normally 15000) and enter 108 into the
timeTxtWanted
    field below. This will make the Unit send a text every hour in addition to the TEXTs at night.

    Thanks and good luck.

    Andrew WS Ainger BSc. CEng. FIET.
    2020
*/

//---------------------- ENGINEERING INPUT ONLY BELOW THIS LINE  -------------------------------------------------------------
//               ( Change NUMBERS ONLY  If uncertain please leave as-is)

unsigned long maxclunkCount = 999998;                // Normally 999998 (yes 8) Maximum value of clunkCounts (up to about 2 years worth) before manually
reset.
                                                     // Dont make this too low (ie: <EngText) as the autotrigger for the EngText will never be reached.
```

```
//////////////////////////////////////////below to calibrate main loop to 30 secs ////////////////////////////////////////////////
unsigned long sleepTimeSLOWLOOP = 6100;              // Calibreation of Loop-Count-Time. This ia an 'unsigned long' This (2,000mS to 12,000mS ca) is
adjusted
                                                     // to give a total loop time of 30 secs. (This should be calibrated for every new unit)
                                                     // It saves power whilst still complying with Shannan-Hartly Theroum (Average bucket fill-time
about
                                                     // 90sec for a full 20-litre water bottle)  An additional
                                                     // 22sec delay is below too.  (#3888 = 6252   #3887 = 6250 or 6100)
/////////////////////////////////////////above to calibrate main loop to 30 secs ////////////////////////////////////////////////

int ENGsmsST = 1;                                    // Number of Sun-Times/ Days (sunrises) between each SMS (Typically every day, (Range between 1 &
99)
                                                     // (depends upon passing dark clouds - default = 1))

int timeTxtWanted = 15000;                           // Set to 138 for 1.25 hour, or to 15000 for a day or multiples of day-texts. (138 in 30 sec blocks
+
                                                     // 6 min for txt)  y=mx-c. (c=6mins) (If left in a light office should text every 5 days with 15000
tbc)
                                                     // DO NOT SET LOWER THAN 125 otherwise battery may never recharge in high-voltage trap.
                                                     // This counter is reset each evening or Dark-Time anyway so will auto-reset its count then but
will
                                                     // also reset after each text sent/attempted.

int reTxtCount = 45;                                 // This is the minimum ammount of time (in number of 30 second units) to the next text (normally
set to 45)(plus about 6 mins as now 3 texts)
                                                     // NB This must be LOWER than the  txtPowerCount  or it wont send first start up text.
                                                     // (We dont want it to send the first text upon reset as the battery may be flat and it will get no
                                                     // time to charge up - care to be taken here)

int stopChargeFor = 120;                             // 120 Number of Main Loops (in 30sec lots) system STOPS charging the battery whilst consuming an
                                                     // extra 100mA for the relay and the SIM unit. (Should be between 120 and 240) when battery too
high.
                                                     // MUST NOT BE TOO SMALL AS STRESSES RELAY'S 50,000 CYCLE LIFE.

///////////////////////////////////////below to calibrate voltage reference///////////////////////////////////////////////////
```

```
const float measuredVcc = 5.12;                    // CALIBRATION: Used in Batt VOLTAGE calculations as it uses: Measured Vcc by multimeter divided by
                                                   // reported Vcc to calibrate unit?? need to check this in next version awsa-nov19
const float reportedVcc = 5.16;                    // CALIBRATION: In big 3888  5.01/4.94;     In big 3887   5.03/5.19     In big #0003    5.0/5.10

int calibrateTEMP =   0.8;                         // CALIBRATION: TEMPERATURE MANUAL CALIBRATION: From -1.4 to about 0.8 FOR UNIT 3888 & for #3887v2

///////////////////////////////////////above to calibrate voltage reference ////////////////////////////////////////////////////

/*     AWSOM unit defaults are:
       Set up for 1 DAY/NIGHT transition between texts (or 15000 cycles ie about 5 days if left on 24/7)
       Set up for 4 hours charge when flat
       Set up for 1 hour discharge when over charged
       Set for 22.5 mins (45 Count) + 6 mins delay for retext period and
       Resets its internal txtPowerCount every 0 counts (100 Days as just 12 active hours/day!)??
*/
//                 (If uncertain in anything above please leave as-is)

//----------------------------------------------------------------------------------------------------------------------
// This is the rest of the sketch/software for the AWSOM Well Maintenance Unit.
// Please change it with great care! It took me over TWO years of part-time activity to get this far!
// I know its not perfect but it works (so far!)- thanks, Andrew.

// Libraries & definitions:
#include <Vcc.h>                                         // Library to measure its own Vcc (& lower, compared to its internal 1.1v band gap referance.
                                                         // GREAT!  https://github.com/Yveaux/Arduino_Vcc/tree/master/examples
#include <Sim800L.h>                                     // This is one of the libraries for the great and low-cost little phone SIM800l txt unit.
#include <SoftwareSerial.h>                              // Required to send messages to the PC (when connected) a really usefull function.
#include <Sleep_n0m1.h>                                  // I can use ONE SLEEP library but not two such as NARCOILEPTIC as well.
Sleep sleep;                                             // Used in SLEEP LIBRARY!
#define RX  10                                           // The RX receive of the Sim800l goes to Pin 11 eleven of the Adduino
#define TX  11                                           // The TX transmit of the Sim800l goes to Pin 10 ten of the Adduino
Sim800L GSM(RX, TX);                                     // Used to send the texts to phones, GSM800l use only. Must have defined RX & TX FIRST
ie:'ABOVE'
                                                         // this line uses 8%SRAM
```

```
char* number;                    // In older version I had char number[11]=""; //phone number to send message to but now
defined
                                 // separately
bool error;                      // To catch the response of sendSms?
int suntimePinState;             // Used in conting the Days in below proceedure
volatile int suntimePin = 3;     // Pin 3 is an inturrupt input triggered by the  light sensor (solar cells) to count the
                                 // 'DAYS(ish) (it actually increments at dusk)
int clunkCountPin = 9;           // pin 9 for the PUMP-COUNT.(clunk-count)  Radar unit's Output is fed to this pin
byte mosfetPower = 4;            // NOW A RELAY but this mosfetPower PIN 4, powers the 4.1v regulator that supplies the
SIM800l
                                 // module that sends the TEXTS
byte radarPower = 5;             // Radar Power PIN 5, powers the vibration sensor only when it's daytime (A byte stores an
                                 // 8-bit number, from 0 to 255 inclusive)
unsigned long suntimeCount;      // This gives us 4 bytes of data 4 million, safer place in RAM for the suntimeCount rather
than 'file'
unsigned long suntimeLastCount;  // This gives us 4 bytes of data 4 million, safer place in RAM for the suntimeLastCount
rather than
                                 //'file'
unsigned long clunkCount;        // A variable to hold the number of Times (in 30-sec increments) the Sensor has detected
there has
                                 // been activity around the Well.
int clunkCountPinState;          // The State of Pin 9 is recorded here (ie: radar's output)
int Pin13 = 13;                  // Just used as a quick reference for an AVO on Pin 13 to see, if you want to, if it goes
high for
                                 // a short time every time round the main loop (now attached to pin 13 LED)
float batVoltage;                // Using new voltage library and changed to battVoltage from v
                                 // https://github.com/Yveaux/Arduino_Vcc/tree/master/examples/VccSimple  and
                                 //  https://github.com/Yveaux/Arduino_Vcc/blob/master/examples/VccSimple/VccSimple.ino
const int temperaturePin = A0;   // Put the temp sensor Ooutput on this analogue pin
float temperatureNow;            // Register to store the CURRENT  Temperature straight from the tempPin A0
float temperatureMax;            // Register to store the MAXIMUM internal Tenperature inside the cork-lined housing
int ENGsmsCOUNT = 0;             // Value of current Eng SMS Count (how many times it has been sent so far since the last time
it
                                 //was zeroed)
```

```
int ENGnextSMS = 0;                                        // Record value of SunTime for comparision with  int ENGsmsST  which will trigger next eng
text.
int ENGlastSMS = 0;                                        // Record value of SunTime for comparision with  int ENGsmsST  which will trigger next eng
text.
int loopCount = 0;                                         // This is used to reset the main Void Loop counter to one. The number of times the main void
                                                           // loop is gone through in multiples of 30 seconds
int chargeLoopCount = 0;                                   //  This is to track loopCount before start charging battery again.
int overVoltageLoop = 0;                                   // Only used in the OverVoltage section of code

unsigned long txtPowerCount = 2;                          // This is used to Count the main Void Loops (say up to 150,000 to stop multiple texts in the
                                                           // evening by having to wait for say 45 loops (22.5 mins) )
unsigned long txtedPowerCount = 1;                        // This is used to determine the minimum time between texts (about 22.5 mins or 45
loopcounts).
                                                           // Records the loopCount since last Text.

char textMAXtemp[30];                                      // temperature
char CHARMAINE[125];                                       // text for SMS. This is the MAIN CHARacter set ready to send the sms Charmain ;-)
const float VccMin   = 4.60;                               // Minimum expected Vcc level, in Volts.
const float VccMax   = 5.45;                               // Maximum expected Vcc level, in Volts.
const float VccCorrection = (measuredVcc)/(reportedVcc);   // Measured Vcc by multimeter divided by reported Vcc. SEE ROW 187  188 TO CHANGE THIS
VALUES.
                                                           // (For OLD REF:-  tiny 5.20/5.12;    In big 3888  5.42/5.20;     In big 3887 5.13/5.28
                                                           // (for loughbrough 3888 settings 5.06/5.22  now 4.91/4.79 )
Vcc vcc(VccCorrection);
const int analogPin = A2;                                  // Trying to measure the HIGH battery voltage directly now with no library (as won't work
over 5v)
                                                           // but using a divide by 2 resistor chain on A2
int startChargeCount = 0;                                  // transitory holder of info later below

//------------------------------------------------------------------------------------------------------------------------------------
```

```
// Now Setting everything up - the below happens once every couple of years or so on power-up and on RESET only (I hope)  ;-)

void setup() {                                          // NB the activation of WDT created chaos with some Chinese Arduinos, so I dont use the
WDT.h library anymore

                                                        // https://www.youtube.com/watch?v=hT24hOTrbEI deactivate early as could lock up Arduino
                                                        //for ever! it was removed.
  pinMode(temperaturePin, INPUT);                       // Pin A0  Three-pin temperature sensor
  pinMode(suntimePin, INPUT);                           // PIN 3  this is used as an interrupt. When the sun comes up and the solar cell wakes,
                                                        // pin3 kickes off the main loop again till night time
  pinMode(clunkCountPin, INPUT);                        // pin 9, the pin from the Well 'radar/movement' sensor.
  pinMode(mosfetPower, OUTPUT);                         // Pin 4  now a relay, powers the SIM800 phone module ( the MOSFET was too slow starting
                                                        // up and getting over 3.8v for the 4.1v regulator.)
                                                        // https://www.arduino.cc/en/Reference/DigitalWrite
  digitalWrite(mosfetPower, HIGH);                      // setting pin 4 to be HIGH to initially turn OFF the PHONE module (NB its a RELAY now)
  pinMode(radarPower, OUTPUT);                          // Pin 5  powers the vibration-sensor/'Radar' module.
  digitalWrite(radarPower, LOW);                        // setting pin 5 to be LOW to initially turn OFF the 'Radar' module
  pinMode(Pin13, OUTPUT);                               // Used to flash for about 150mS the LED every 30 seconds so we can see its working
                                                        // (but not when its asleep)
  digitalWrite(Pin13, LOW);                             // Turns off the LED on pin 13
  Serial.begin(9600);                                   // Starts internal serial print to PC via USB port
  while (!Serial) {                                     // Wait for serial port to connect, (for native USB port only)
    ;                                                   // Wait for serial port to connect, (for native USB port only)
  }                                                     // Wait for serial port to connect, (for native USB port only)
  temperatureNow = 0;                                   // Setting up TemperatureNow register to zero
  temperatureMax = 0;                                   // Setting up TemperatureMax register to zero
  ENGnextSMS = ENGsmsST;                                // Sets up the first time the SMS is sent
  suntimeCount = 0;                                     // This sets the suntimeCount to 0 ready for the next 2 years or so after the manual
                                                        // max clunkCount RESET.
  suntimeLastCount = 0;                                 // This sets the suntimeLastCount to zero ready for the FIRST suntime calculation.
  batVoltage = 9.99;                                    // Set this high to pass the first voltage test and to make it obvious when its shown
                                                        // in USB.    see : https://github.com/Yveaux/Arduino_Vcc/tree/master/examples/VccSimple
  Serial.println("");                                   // The below sets up the USB print formatted message below . .
  Serial.println("");                                   //
  delay(100);                                              //
}
//---end setup ------------------------------------------------------------------------------------------------------------------
```

```
//-----------------------------------------------------------------------------------------------------------------------------------
void loop() {                                              // Main Loop of the sketch, set now to 30 secs loop time to save power.

  // This Main Loop calls the suntimeCount (day-count) function. When the suntimeCount (day-count) goes over a set value (default 1) it sends an SMS of
the
  // Well's PUMP ACTIVITY and resets things.
  // So: when  PIN 3 (I/P) is LOW, (ie: when it goes dark, in the evening and is pulled down by the 100kohm jic) it then increments the int sunTimeCount
by 1
  // when the numbwer of DAYS is = to or over the set number, (normally 1) then an SMS is sent.

  // Also, if the battery is lower than 4.6 volts it will switch itself off and wait for the sun to recharge it for a few hours and then it will try
again.
  // Finally, when the Well PUMP-COUNT (clunkcount) gets too large (about every 24 months ie >=999998 clunks) then a manual RESET is recommended. This
  // keeps the pumpcount value low for the spreadsheet manual update process.
  // It is recomended that this PUMP-COUNT figure is limited to just six figures as if it is any more it will be difficult to manually read and
transcribe
  // error-free into the spreadsheet.

  interrupts();                                            // Just to ensure interrupts are ON not only by default but by design too.
  Serial.println();
  delay(50);                                               //
  int sensorValue = analogRead(A2);                        // Setting up pin A2 as the raw battery voltage


/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  float newVoltageBatt = sensorValue * ((4.91 * 2.00) / 1023.0); //Use real Vcc reading from voltmeeter 4.91v x 2.00 as using a divide by 2 resistor. SO,
Change 4.91 for each AWSOM unit

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

  // chain on Pin 2 CALIBRATIONISH.  ( see  https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage )
  delay(100);
  // INCREMENT LOOP COUNT
  loopCount = ++loopCount;                          // This increases the Powered-time or loopCount (LC) each time round the mainLoop but resets
                                                    // each TEST
  txtPowerCount = ++txtPowerCount;                  // This increases the Powered-time or loopCount (LC) each time round the mainLoop but resets
                                                    // every 150,000 Counts


  if(txtPowerCount > 150000) {                      // This limits the txtLoopCount so about every 100 days (150000 loops) txtPowerCount will be
                                                    // set to 0 and start count again (limits data corruption ?)
    txtPowerCount=65;                               // This figure is HIGHER than  reTxtCount so, when SET, it won't send a first text
immediately?
                                                    // (waits for battery to charge)
    txtedPowerCount=0;                              //
  }
  delay (100);                                      // JIC

  // CATCHES SUB 1-DAY TEXT REQUESTS
  if (loopCount > timeTxtWanted) {                  // This establishes when the loopCount is greater than the timeTxtWanted and
    loopCount = 1;                                  // Resets the loopCount to 1 ready to count up to the next text and
    sendENGsms();                                   // We send the sms
    delay (500);                                    // Ths waits for it to all settle down again before continuing with the Radar ETC stuff
  }

  // FLASH THE LED ON PIN 13
  digitalWrite(Pin13, HIGH);                        // Just to act as a TEST point on Pin13 for sub-second LED flash.
  sleep.pwrDownMode();                              // Sets the sleeptime Mode here.
  sleep.sleepDelay(150);                            // Power saving as in main loop.
  digitalWrite(Pin13, LOW);                         // Turns Pin13 OFF here

  // SLOW LOOP DELAY FOR 30-SEC LOOP TOTAL
  //(NB-includes radar time delay)
  sleep.pwrDownMode();                              // Sets the sleeptime Mode here.
  sleep.sleepDelay(sleepTimeSLOWLOOP);              // Sleep delay here to SAVE POWER and to act as a slowing-down device to read the serial
                                                    // monitor and calibrate the 30 sec main loop.
                                                    // Can't be too long as need to maintain Shannan-Hartly Theroum
  // PRINTS to USB "AWSOM START-UP  " and the Sketch's Version number
  Serial.println("  ");                                    // Formating the serial output to the USB port
 //
  Serial.print("AWSOM START  ");                         // Serial progress report to USB
  Serial.print(versionNumber);                             // Serial progress report to USB
  //Serial.println();
  delay(100);                                              // Delay to let Serial port finish 'printing'

  // prints to USB progress only - very usefull . . .
  // can un-dump below to test then redump as short of SRAM ////////////////////////////////////////////////////////////////////////////////////////////
/*
  Serial.println("newVoltageBatt,  batVoltage ,  Vcc");     //
  delay(100);                                              //
  Serial.print("");                                        //
  Serial.print(newVoltageBatt);
  Serial.print("             ");                           //
  delay(100);
  Serial.print(batVoltage);
  Serial.print("          ");                              //
*/
  delay(100);
  float v = vcc.Read_Volts();
//  Serial.print(v);
//  delay(100);
//  Serial.println();
  //can un-dump above to test then redump as short of SRAM
/////////////////////////////////////////////////////////////////////////////////////////////////

  // THIS SERIALLY PRINTS & FORMATS A TEXT WHEN BATTERY IS FLAT IE: <4.60v THEN SLEEPS
  if (batVoltage < 4.60) {                                 // 4.60  If BATTERY is less than 4.60 volts (ie: Vcc<4.40v) then it will wait for a time so
                                                           // solar cells can charge battery before wakeing up again and powering up etc.
                                                           // (Arduino worked down to 4.3v but not g'teed!)
    String textBLANK = ("");                               // Why does this work? But it does on the Uno too so don't argue . .
    char textBatt[30];
    sprintf(textBatt, "    : %d.%00d", (int)batVoltage, abs((int)(batVoltage * 100) % 100));  // This sends a FLOAT as text see
```

```
                                                    // http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html
                                                    // at last. (NB: Does not go negative)
    delay(100);
    Serial.println(textBLANK + " Battery low " + batVoltage + " volts");
    delay(100);
    Serial.println("Charging Battery for 4.5 Hrs");        // Serial progress report to USB
    delay(100);
    digitalWrite(mosfetPower, HIGH);                       // As Need to turn off everything for a few hours so battery can charge this Powers OFF
                                                    // the 4.1v regulator needed for the SIM8001 module
    digitalWrite(radarPower, LOW);                         // Powers OFF the radarPower module. Turned on again in suntimeCountFunction.
    sleep.pwrDownMode();                                   // Sets the sleeptime Mode here.
    sleep.sleepDelay(15000000);                            // 15000000 = 4 hour 10 mins Sleep delay here to CHARGE BATTERY first thing in morning
                                                    // just in case it's nearly flat  (4 x 60mins x 60secs x 1000 mS) 15000000 (or, to cancel,
                                                    // just press RESET on the Arduino)
  // NEED TO GET BATTVOLTAGE HIGH othewise it will start charging again as it remembers the last low voltage?
    newVoltageBatt = 4.80;                                 // to make sure it does NOT re-enter the flat-loop again AND FORCES A 30 sec RE-TEST.
                                                    // This has to be set ABOVE batVoltage above as THE SEQUENCE goes through the loop twice to
    delay(100);                                            // update the newVoltageBatt, AND LOWER than V max of 5.90v
    Serial.println(batVoltage);                            // batt voltage ok but script reads Vcc plus 0.2v
    delay(100);
    Serial.println(v);                                     // reads Vcc
    delay(100);
    Serial.println();                                      // This is set 7 lines above to make sure that it comes out of the LOW VOLTAGE loop.
                                                    // It should be just higher than the IF STATEMENT 20 lines above so it does not repeat
    delay(100);                                            // the LOW BATTERY LOOP again whilst it catches up. It forces a 30sec re-test of the battery
voltage
  }

  // STOPS OVER-CHARGING THE BATTERY WHEN THE VOLTAGE IS > 5.90 VOLTS AND SERIALLY PRINTS MESSAGE   --------------------------------------
  if (overVoltageLoop < startChargeCount) {                // If battery is OVER indicated 5.90v volts then it will switch the relay and stop
                                                    //charging fo a period but will continue counting activity, it also dumps power in the
sim8001
                                                    // THE ABOVE CHECKS IF BATTERY IS ALREADY OVER VOLTAGE, IF SO IT SKIPS THE VOLTAGE TEST AND
                                                    // THE 'RESSETTING' SECTION.
    Serial.println ("  STOPPED CHARGING Bat High");
    delay (100);
    Serial.print("  Will start Charging in   ");            // Serial progress report to USB.
    Serial.print(startChargeCount - overVoltageLoop - 1);   // Serial progress report to USB.
    Serial.println();                                       //
    delay(100);                                             //
    digitalWrite(mosfetPower, LOW);                         // Turn OFF charging by turning ON relay and SIM ON too.
    overVoltageLoop = ++overVoltageLoop;                    //

  } else {
    overVoltageLoop = 0;                                    // Resets this overvoltage count as needed
    startChargeCount = 0;                                   // THIS line resets the startChargeCount when the relay turns OFF (& STARTS CHARGING AGAIN)
                                                            // As must stop any high value being held in memory over TEXT transitions.

    if (newVoltageBatt >= 5.80) {                           // 5.80 USING resistor chain Voltage calcs
      startChargeCount = stopChargeFor ;                    // Want to put 60mins = 120 at no-charge before testing again. MUST NOT BE TOO SMALL AS
                                                            // STRESSES RELAYS LIFE. But cant be too big because LOOP-COUNT MAY never reach the value.
    }
  }

  if (overVoltageLoop >= startChargeCount) {                // Sets the Powered-Count time when the Unit will start charging again.
    digitalWrite(mosfetPower, HIGH);                        // Turn ON charging by turning OFF relay   Serial.print("Relay off");
    delay(100);
  }
  // THE ABOVE STOPS OVER-CHARGING THE BATTERY FOR AN HOUR WHEN THE VOLTAGE IS OVER A SET VOLTAGE AND SERIALLY PRINTS MESSAGE (
  // NB. should have same delay in each if-loop tbc)

  // CALLS THE SUNTIMECOUNT PROCEEDURE HERE  -----------------------------------------------------------------------------------------------
  suntimeCountFunction();                                   // A Main proceedure call here that counts the Days, or more accurately the sunsets,
                                                            // well actually to be precise, it counts the NIGHT_TIMES as dark clouds and camera flash
                                                            // guns can confuse it!
  delay(100);                                               // Gives the processor time to send the data to the USB serial port.

/////////////////////////////////////////////nb-calibration-below  //////////////////////////////////////////////////////////////////////////////
// TEMPERATURE AND VOLTAGE TO USB SERIAL LINE
```

```
  int a = analogRead(temperaturePin);                          // Read the voltage applied to the temperaturePin of the Analogue To Digital Voltage 
Converter
                                                               // (pin  ‘A0’ ). (my toungue gets it to 34C not bad really)
  float volts = a / 205.0;                                     // Divides the number (0 to 1023) by 205 to convert to a voltage and stores the result into a 
                                                               //  ‘floating point’  variable so that we can have a resolution to the right of the decimal 
point.
  float temperatureNow = ((volts - 0.5) * 100) - (calibrateTEMP); // Takes the voltage and converts to a temperature in degrees Celsius;
                                                               //SUBTRACT 1 FOR UNIT 3887 and SUBTRACT 0.8 FOR UNIT 3888 ///////- because I think its the
                                                               // voltdrop in the power line.
                                                               // the offset from the TP35 is 0.75 volts at 25 ° C   a change of 0.01 volt per ° C, 
therefore
                                                               // at 0° C we subtract 25 x 0.01V (i.e. 0.25) from the offset of 0.75 BUT my rough 
calibration
                                                               // shows its 2C+ too high so I subtract 2
  if (temperatureNow > temperatureMax) {                       // (i.e. 0.75-0.25=0.5) and because the change per ° C is 0.01 volts we multiply by 100
                                                               // (how about Serial.println(((analogRead(Temp)/204)-0.5) * 100))??
    temperatureMax = temperatureNow;
  }
/////////////////////////////////////////////nb-calibration-
above/////////////////////////////////////////////////////////////////////////////////////




//////////////////////////////////////////////VIP-
below/////////////////////////////////////////////////////////////////////////////////////////
// remember to change the WELL ID eg:
//                     AWSOM#4567   below...

  sprintf(textMAXtemp,"AWSOM#3888 MxTmp = %d.%00d:", (int)temperatureMax, abs((int)(temperatureMax * 100) % 100));  // This sends a FLOAT as text. 
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
                                                               // http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html
                                                               // works, at last. (NB Does not go negative though)
//////////////////////////////////////////////above////////////////////////////////////////////////////////////////////////////////////
///
```

```
  delay (100);                                            // JIC
  String textBLANK = ("");                                // it needs it, don't know why, but it works this way...
  delay (100);                                            // JIC
  char textBatt[30];

  // SELECTS THE VOLTAGE REFERENCE FOR THE BATTERY VOLTAGE TEXT  -------------------------------------------------------------------

  // This IF statement selects which voltage referance to use before it displays the battery voltage.
  // IF Vcc is under 4.85 volts then we have to use the 1.1v internal band-gap referance, but if Vcc is >4.85 then we have to use the regulated Vcc as
  // the reference source and use the resistor divider to determin the HIGHER battery voltages.
  // BUT there is the regulator volt-drop to consider, this valuse is about 200mV for measuring the lower voltages <4.9v
  // (Watch out for the UPDATE 60Sec DELAY)

  batVoltage = v;                                         // not v really its Vcc at this level
  if (v < 4.85) {                                         // Must be BELOW full output of regulator (5.0v) or it will never switch over to the
potential
                                                          // diveded battery voltage measurement system?
    batVoltage = v + 0.20;                                // The 200mV compensates for the 200mV volt-drop across the Vcc regulator to give the BATTERY
                                                          // Voltage
  } else {                                                // IF GREATER THAN 4.85V THEN USE RESISTOR DIVIDER ANDvCC AS REFERENCE
    batVoltage = newVoltageBatt;
  }
  // THE ABOVE IS THE BATTERY REFERENCE SELECTION  ------------------------------------------------------------

  sprintf(textBatt, "    \rNi-Battery = %d.%02d", (int)batVoltage, abs((int)(batVoltage * 100) % 100)); // This sends a FLOAT as text
                                                          // http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html  at
last!
                                                          // (Does not go negative though)
  delay (200);                                            // JIC
  int newPercent = ((batVoltage - 4.6) * 100) / 1.4;      // Taking  the two-decimal point float and make it an int so we just get no decimal places
                                                          // after the % of battery capacity
  String text18 = String(textMAXtemp + textBLANK + "   \rNi-Battery = " + batVoltage + " Volts (" + newPercent + "%):  \rPUMP-COUNT = " + clunkCount + ":
\rWork-day = " +  suntimeCount + ":  \rPowered-time = " + loopCount + ":");
                                                            //  do I need textBLANK? dont forget the  r  too.    xxxxxxxxxxx r

  delay(250);                                               // time VIP ADDED 100mS here jic
  Serial.println(text18);                                   // Send to serial USB ONLY all the data above, MAX temp, Batt voltage, PUMP-COUNT, Work-day #
,
                                                            // (NOW Temp)  . This is Brill!
  text18.toCharArray(CHARMAINE, 175);                       // Could not think of another way to get text18 into the engineering sms, so used this global
                                                            // char.. (Main Char  :-)
  delay(200);

}

// End of main loop  ------------------------------------------------------------------------------------------------------------
// SUNTIME COUNT PROCEEDURE ------------------------------------------------------------------------------------

// WHEN the solar cells are illuminated then pin 3 is HIGH. (NB: only when the sun goes IN and it gets DARK does the suntimeCount increase by one)
// Anyway, when pin 3 goes HIGH in sunlight the radar sensor is turned ON via Pin5 (radarPower) and gets to go HIGH so that any Well-movement-time can be
// counted in 'about 30 sec' blocks by  the Radar's output on Pin 9.

void suntimeCountFunction() {
  suntimePinState = digitalRead(suntimePin);                // Pin 3. Code checked with
                                                            // https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/
                                                            // Need code here to stop clouds in the morning triggering a new day count. but no time to
                                                            // test it so leaving for version 2 in 2019?
                                                            // ( something like (int set1; if suntimepinstate ==high, int set1 =1, delay 5000, suntime
                                                            // pinstate = digitalread(suntimepin etc...
  if (suntimePinState == HIGH) {                            // If daylight then enter pumpcount
    delay(550);                                             // make this big so jic (this could mean taking 500 milli-seconds out of the main loop.
    clunkCountFunction();                                   // Main call to the clunkCountFunction which counts the cumulative number of time-periods
                                                            // (30sec) the Well has been used
    delay(550);                                             //
  }
  else {
    digitalWrite(radarPower, LOW);                          // Turns off the 'radar' sensor at night and
    suntimeCount = ++suntimeCount;                          // Increments the suntimeCount, the DayCount when the 'Day' is over
    loopCount = 1;                                          // Resets the main loopCount every night (see start of main Loop above)
    Serial.println();                                       //
    delay(100);                                             //

    // SMS TRIGGER REPORTS
    // This will call a Process to send an SMS if Sun-Time Count is greater than that specified
    // NOTE, the Sun Time Count (DAY COUNT) initiates the TEXT

    if ((suntimeCount - suntimeLastCount) >= SMSfrequency){ // The almost ever-increasing Well PUMP-COUNT or ClunkCount.
      suntimeLastCount = suntimeCount;                      // This lets the suntimeCount increment, for the Eng SMS, until maxclunkcount and emergency
                                                            // reset whilst tracking the 'Days' to the next text message

      delay(100);
    }

    // SMS SEND (note this is the only text sent)
    ENGsmsCOUNT = ++ENGsmsCOUNT;                            // Increments the ENGsmsCOUNT by one every time a new 'DAY' passes (ie each night/dark time)
    delay(100);;
    if (ENGsmsCOUNT >= ENGnextSMS) {                        // Call to send ENG SMS
      sendENGsms();                                         // Starts up the SEND-ENG-SMS Function
      delay(200);
      ENGlastSMS = ENGnextSMS;
      ENGnextSMS = ENGlastSMS + ENGsmsST;
      temperatureMax = 1;                                   // Resets the Maximum Temperature after SMS has been sent according to SMSfrequency.
                                                            // (This instruction re-sets the Maximum Temperature record ready for the next 'Day Period')
    }
    sleepFunctionST();                                      // THIS WORKS now.  Do NOT put any WDT in this bit 'cos it will lose the sunCount information
  }
}
// End of suntimeCountFunction (its actually a Proceedure - but anyway. . .) --------------------------------
// CLUNKCOUNT PROCEEDURE  --------------------------------------------------------------------------------------------

void clunkCountFunction() {                          // clunkCountFunction is called & incremented each time Pin 3 is SEEN to go LOW, it then
sleeps
                                                     // for a few secs, (but only if it catches it high!)
                                                     // Put the loop in to catch HIGH but now find I dont need it!? so its set to 1
 // RADAR ON: NEEDS TIME TO RE-SENSATISE
  digitalWrite(radarPower, HIGH);                    // Turns ON radarPower after it sleept in the sleepTimeSLOWLOOP
  sleep.pwrDownMode();                               // Sets the sleeptime Mode here.
  sleep.sleepDelay(22000);                           // Power saving when main loop. Also calibrated loop counter and gives 12 secs for Radar
output
                                                     // to settle & resensatise after turn-on.
  Serial.println("  ");                              // Formatting Serial USB port
  delay(100);

  int clunkCountPinState = digitalRead(clunkCountPin);   // This is to get & record Pin 9 status . .
  if (clunkCountPinState == HIGH) {                  // Dont forget, if HIGH it will be left HIGH untill its made LOW. But this happens only once
                                                     // UNDER THE IF STATEMENT every Main-Look (30secs) so
    clunkCount = ++clunkCount;                       // Increment clunkCount by one
  }
  delay(100);                                        // jic

  // CALCULATES THE NIGHTS TO NEXT ENG TEST & SENDS TO USB
  ENGnextSMS = ENGlastSMS + ENGsmsST;                // This expression makes the ENGnextSMS value equeal to the sum of the last one (ENGlastST)
                                                     // plus the ENGstCOUNT in User Input. (not fully tested as set now to 1 day) fix in version
2?
  delay(100);

  // SENDS EMERGENCY RESETING TEXT
  if (clunkCount >= maxclunkCount) {                 // This calls CC SMS Function to send sms if count is greater than that specified (eg:
999,998)
    digitalWrite(radarPower, LOW);                   // This turns off Radar whilst SIM800 is powering up
    Serial.println();
    Serial.println("Perform manual RESET asap");               // Sketch serial progress report only
    Serial.println("Sending RESET Text");                      // Could put on one line but looks better in serial on two
    delay(100);
    sendENGsms();                                              // it sends the eng sms
    delay(5000);                                               // jic
    clunkCount = 1;
    digitalWrite(radarPower, HIGH);                            // This turns back on Radar Power when the sms and the emergency reset txt has been sent
                                                               // (if it is sent - needs further checking) v2
    delay(30000);                                              //

    Serial.println("Manual RESET needed");                     // Sketch serial progress report
    delay(100);
  }
  delay(100);                                                  // Sketch serial progress report & jic only
}
// End clunkCountFunction  -----------------------------------------------------------------------------------------------
// SEND ENGINEERING TEXT PROCEEDURE -----------------------------------------------------------------------------------------

void sendENGsms() {                                          // sends SMS for sendEngText based on clunkcount value

  Serial.println("preTxt-if:  if(txtPowerCount-txtedPowerCount < reTxtCount then skip txt"); //
  Serial.print("");                                          //
  Serial.print(txtPowerCount);                               //
  Serial.print(", ");                                        //
  Serial.print(txtedPowerCount);                             //
  Serial.print(", ");                                        //
  Serial.print(reTxtCount);                                  //
  Serial.println(".");                                       //
  delay (100);

if(txtPowerCount-txtedPowerCount < reTxtCount) {             // So, when a TEXT is about to be sent it checks that the loopCount is more than several mins
since
                                                             // the previous text. if( CURRENTloopCount-loopCountTexted AS SET IN LIB AND DEFINITIONS IS <
reTxtCount)
  delay (50);                                                // This is just to occupy the ELSE space. it does nothing really, its just easier for me to
read
                                                             // the sketch using this technique.
} else {

  digitalWrite(mosfetPower, LOW);                            // Now a RELAY, but this action now POWERS-UP the 4.1v regulator needed for the SIM800L
module.
                                                             // GIVES SIM800L TIME TO SETTLE
  delay(250);                                                //
  Serial.println(CHARMAINE);                                 // The Main Char ;-)
  delay(150);
  GSM.begin(9600);                                           // do i need this line as its global? JIC
  delay(150);
  Serial.println();
  Serial.println();
  Serial.println();
  Serial.println();                                       // For USB spacing of eng test info
  Serial.println("   SENDING TEXT");                      // USB print out
  delay(500);                                             //

  int i;
  for (i = 0; i < 6; i++) {                               // this FOR loop sets the time (number of FOR-LOOPS before droping out) in this case LOOP is
                                                          // 5000 milli secs and FOR is<6 but this includes 0 so is 6 really so 6*5000 ms = 30 secs
delay here
    sleep.pwrDownMode();                                  // Sets the sleeptime Mode
    sleep.sleepDelay(5000);                               // Sleep delay to save power AND to give the SMS unit to sync with the mobile network say 30
secs
                                                          // (= 30x1000ms)
  }
  delay(100);                                             // Time to stabalise
  Serial.println(CHARMAINE);                              // The file that contains the text data from 'text18'
  delay(200);                                             // what?
  sprintf(textMAXtemp, "Max-Temperature = %d.%00d", (int)temperatureMax, abs((int)(temperatureMax * 100) % 100));
                                                          // This sends a FLOAT as text
                                                          // http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html
                                                          // Works at last! (NB Does not go negative though)
  String textBLANK = ("");
  char textBatt[30];
  sprintf(textBatt, "        Ni-Battery = %d.%02d", (int)batVoltage, abs((int)(batVoltage * 100) % 100));
                                                          // This sends a FLOAT as text
                                                          // http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html
                                                          // at last. (Does not go negative though)

//////////////////////////////////  The below three should send to Ray and Godfrey & Mike -
////////////////////////////////////////////////////////////////////////////////////////////////////////

  delay(30000);                                           // To give the SMS unit time to sync with the mobile network, say 30 secs, on the FIRST txt

//This 11 lines of code below auto-enter the SIM PIN (called the  String pin ) when needed -(I Hope)
```

```
  Serial.begin(9600);                                   //connect to GSM network usng a pin number
  delay(5000);                                          //was no delay either but seems to send text now
  GSM.begin(9600);                                      //was 4800 but seems to send text now?
  delay(20000);                                         //was no delay either but seems to send text now



//##########################################################################################################
####

                                                        // Hi Both,
String pin = "0000";                                    // Please ONLY change the four numbers (digits)on the LEFT between the quotation marks   "
"
                                                        // leaving all the quotation marks, semi-colon and spaces - thanks and good luck!

//##########################################################################################################
####



  delay(5000);                                          //
  if (GSM.setPIN(pin)){ Serial.println("Pin set");      //
  delay(5000);                                          // jic
  }else {Serial.println("SIM check");
  delay(5000);  }                                       // jic


  delay(500);                                           // BELOW
  error = GSM.sendSms(mike, CHARMAINE);                 //   The argument here used to be "07549871528" Just // 'comment out' to stop sending texts.
Soon to go to mike
  delay (60000);

//  delay(500);                                           // BELOW
//  error = GSM.sendSms(andrew, CHARMAINE);               // Just 'comment out' to stop sending texts
//  delay (60000);
```

```
  delay(500);                                                   // BELOW
  error = GSM.sendSms(ray, CHARMAINE);                          //  comment this out in final version
  delay (60000);

  delay(200);                                                   // BELOW
  error = GSM.sendSms(godfrey, CHARMAINE);                      //  Just 'un-comment out' to send text
  delay (60000);                                                //



////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////


  int j;                                                        // not necessary, but not fully tested with it removed so leave to version 2
  for (j = 0; j < 5; j++) {
    sleep.pwrDownMode();                                        // Sets the sleeptime Mode
    sleep.sleepDelay(5000);                                     // Sleep delay to save power AND to give the SMS unit to send over the mobile network say 30
secs
                                                                // (= 30x1000ms) (again?)

  }
  startChargeCount = 0;                                         // This is to make sure that after each TEXT the startChargeCount is low as it could be HIGH
and
                                                                // never start charging the battery ever again!.
  digitalWrite(mosfetPower, HIGH);                              // Powers DOWN the 4.1v regulator for the SIM8001 module so turns it off. THIS is set at HIGH
AS
                                                                // HIGH IS OFF FOR THE GSM via the RELAY
  delay(100);                                                   // JIC

  txtedPowerCount = txtPowerCount;                              // Sets the txtedPowerCount to the current txtPowerCount value for comparison above (ED.watch
spelling!)

  Serial.println(" txtPowerCount, txtedPowerCount, reTxtCount"); //
  Serial.print(" ");                                            //
  Serial.print(txtPowerCount);                              //
  Serial.print(", ");                                       //
  Serial.print(txtedPowerCount);                            //
  Serial.print(", ");                                       //
  Serial.print(reTxtCount);                                 //
  Serial.print(".");                                        //
  delay (100);


  // This stops any rapid repeat sms as it slowly goes to dusk
  sleep.pwrDownMode();                                      // Sets the sleeptime Mode
  sleep.sleepDelay(300000);                                 // Sleeps for 5 mins to stop any repeat sms just as it slowly goes dark
  delay(100);                                               // JIC
}
}
// End sendENGsms  -----------------------------------------------------------------------------------------------------
// SLEEP SUN-TIME PROCEEDURE  ------------------------------------------------------------------------------------------

void sleepFunctionST() {
  int preSleepPinState1;                                  // New interger defined here. This is to catch and stop the time a dark cloud appears at
                                                          // night and and goes away again which used to lock up the Unit for a day
  preSleepPinState1 = digitalRead(suntimePin);            // Code checked with   \www.arduino.cc\en\Reference\DigitalRead.html
  if (preSleepPinState1 == LOW) {
    Serial.println( "");                                  //
    Serial.println( "");                                  //
    delay(100);                                           // Need this to ensure serial port stays active long enough to 'print' the message
    sleep.pwrDownMode();                                  // Sets the sleeptime Mode
    sleep.sleepDelay(20000);                              // Sleep delay to save power in DUSK DARK-CLOUD FILTER. (also using this it does not lock
                                                          // up the system for a day as before should be say 20 secs? SHANNON-HARTLEY THEOREM
    delay(100);                                           // Need this to ensure serial port stays active long enough to wake-up
  }
  int preSleepPinState2 = digitalRead(suntimePin);        // New interger defined here. This is to catch and stop the time a dark cloud appears at
night
                                                          // and goes away again which used to lock up the Unit for a day
  if (preSleepPinState2 == LOW) {                         // This checks as the last moment before going to sleep if the Sun is out again and if so
then
                                                          // it skips the sleep bit but
                                                          // it still sends the TEXTS. So, it is counted as a 'DAY' but does not lose info or lock up!
                                                          // brill(at last)
    Serial.println("Sleeping until SUN-Rise");
    delay(100);                                           //delay to allow serial to fully print
    Serial.println("");
    Serial.println("");                                   //
    Serial.println("");                                   //
    Serial.println("");                                   //
    delay(200);                                           //delay to allow serial to fully print before sleep
    sleep.pwrDownMode();                                  //set sleep mode.  Sleep till interrupt pin equals a particular state.
    sleep.sleepPinInterrupt(suntimePin, RISING);          //(interrupt Pin Number, interrupt State) (was HIGH & seemed OK but RISING is recomended)
    delay(100);                                            // JIC

    //   This stops LIGHTENING triggering unit and waking it up. BUT ONLY ONCE!
    delay(2000);                                          // lightening delay?
    int postSleepState1;                                  // New interger defined here.
    postSleepState1 = digitalRead(suntimePin);            // Code checked with    \www.arduino.cc\en\Reference\DigitalRead.html
    if (postSleepState1 == LOW) {
      Serial.println( "D3=LOW still");                    // Just sends confirmation to serial port that suntimePin D3 is HIGH now so will not sleep.
                                                          // Needs to skip sleep if HIGH hence ELSE below.
      delay(100);                                         // Need this to ensure serial port stays active long enough to 'print' the message
      sleep.pwrDownMode();                                // Sets the sleeptime Mode
      sleep.sleepPinInterrupt(suntimePin, RISING);        //(interrupt Pin Number, interrupt State) (was HIGH & seemed OK but RISING is recomended)
      delay(100);                                         // Need this to ensure serial port stays active long enough to wake-up
    }
    //   end of LIGHTENING trigger
  } else {                                                // NOTE This relates to the TOP 'IF' statement (I know, not good but no time to resolve just
now)
    delay(100);                                           // But now it just checks its not a cloud by checking again after this new sleepdelay
    Serial.println();
    Serial.println( "D3=HIGH");                           // Indicates that the suntimePin D3 is HIGH so the unit will go into main loop again
    delay(100);                                           // Need this to ensure serial port stays active long enough to 'print' the message
  }
}
// END sleepFunctionST

//----------  SKETCH END  ------------------------------------------------------------------------------------------------




/*

  COMMENTS & lESSONS LEARNT
```

This is only my second shetch. After the ubiquitous BLINK sketch I launched straight here.
Wow! Have learnt a lot, still lots to learn.
Sorry about the format, it's not very conventional but I am used to it now.
The next version (yours?) will be better.

```
  lESSONS IDENTIFIED:
  1. Combining text and intergers use sprintf, it takes 1k memory but is simple to use this command
  2. The serial/sms code now allows you to upload whilst the SIM800l is connected!
  3. If the SIM800L is connected and the serial window is open then the text is sent twice when the sketch is uploaded,
   (I think once on start up and once when it realises the serial window is open so it restarts quickly)
  4. CHECK SPELLLING! eg    suntimePinState:   is different to  suntimepinState;   in TWO ways!  Yes 2.   VIP
  5. But first how about stopping interuppts in the interupt? YES it's automatic as the compiler does it!
  6. Bring ALL  "NEEDS USER INPUT HERE"  lines up-top......` DONE - almost!                                          VIP
  7. Narcoleptic.delay needs an ordinary delay(100) infront of it for printing, but it does NOT work with SLEEP.h  SO DO NOT USE ANYMORE
  8. It's VIP where in the sequence you put instructions. EG the RESET 1 count has to be at the END of void setup()
  9. Dont need to create clunkcount text as this only occurs never!
  10 LOOK FOR CORRECT PLACING OF { & }
  11 stringThree = stringOne + stringTwo;
  12 Check wires! They become loose.                                                                    VIP
  13 Make strings as SHORT as possible! (only 2k SRAM)                                                  VIP
  14 COMMENT OUT STRINGS IN PRODUCTION VERSION
  15 Can't use just suntime Delay sleep must use inturrupt otherwise DayCount will increase after every daytime delay loop.
  16 Need modified sprintf to combine float (see voltage sketch)
  17 serial.end(); shuts off serial because it uses interrupts!?
  18 If I use an IF statement there must be a reason to have the associated ELSE statement too - deep thought please  VIPish
  19 Write psudo sketch logic first                                                                     VIP
  20 Review what calls what when                                                                        VIP
  21 GSM800l VERY sensative to voltage keep at 4.1v can't even use RS Ammpmeter to measure it due to long leads voltage drop making it loose GSM
synhronisation
  22 Take GREAT care if using more than 60% of Dynmic RAM as it can cause erratic behaviour! This is      VIP
  23 Use VERTICLE space to separate Serial USB display and NOT lines eg ------------------ as it gobbles RAM
  24 pinMode(pin, INPUT);           // set pin to input
  24 digitalWrite(pin, HIGH);       // turn on pullup resistors  WRONG !!
  25 START with a small sketch, compile, load and run. Then expand the sketch to include other bits!!     VIP
```

26 NOTE the radar unit needs >11 secs to stabalise after turning on. At switch-on O/P goes HIGH for <3sece THEN GOES LOW for <3secs then starts to detect for real with no gaps. VIP
27 If the battery voltage is higher than the PC USB voltage by over 200mV then the Sketch will not load. (just leave the AWSOM unit in the shade for a couple of hours & wait for the battery to didchange)
28 If the battery voltage is unknown then disconnect it and disconnect the phone module and use the PCs USB voltage whilst loading the sketch.
29 Don't forget to select, under TOOLS the PROCESSOR. More likley to be Bootloader(old)
30 Don't LOAD SKETCH when Vcc is higher than USB voltage because it fails to load.
31 The zener to keep the voltage below 5.5 volts is soft Not used anymore.
32 NB, The seral/USB connector takes >40mA when powered by the Arduino and not the associated PC! (not anymore)
33 NB Only upload sketch whem BATTERY voltage is 5.0v (otherwise it fails to load!)
34 USE (select from drop-down) old bootloader
35 CHARGING via A 5.5V SOLAR CELL GIVES Vcc as <5.1V, IT LASTed about 3 HRS OF NO DIRECT SUN SO:
36 NEED TO CHARGE UP MORE, TO Vcc of 5.4v via a 6v solar cell and then stop charging switch when Vcc too high 5.85 now
37 With old ENLOOP batteries it lasts just 25 hrs (but watch out for loose psu connections) y = -0.0005x + 5.3976
38 PUT CAP ACROSS ARDUINO PINS as Cap near the battery makes little difference to apparent voltage drop reported by engtxt message
39 REMEMBER TO CHECK THE PUMP-COUNT CAN GO UP TO 6 DIGITS (NOT JUST 2)
40 PIN13 LED FLASHES TOO FAST SOMETIMES WHEN A TEXT IS BEING ATTEMPTED, WHY? COS IT RESETS DUE TO VOLTAGE DIP CREATED BY CURRENT TAKEN FOR 250MS by SIM800L
41 REMOVED ARDUINO power led & ZENER CURRENT DUMP to save power
42 Removed the constant current/voltage charging unit as not needed as solar cells cant overcharge batterys (i think)
43 very slow when waking up! remember it takes 8 mins to stop multiple text batches
44 Remember that the VOLTAGE divider cct can show voltages that are HIGHER than Vcc but not lower,
45 This is because the 5v regulator drops down with the battery feeding it and the analogue pin will always show Vcc when Vcc is LOW as it will always be
half what ever the Vcc is..
46 DON'T make extensive changes before recompiling and testing, even if it ony formatting the sketch, as you will make unintentional and difficult to detect errors! (as i did - several times!)
47 DO NOT Use more than 66% SRAM as crazy and illogical things happen! (best to keep it 63% or below. (this sketch is 61%, just remove USB PRINT lines to reduce it)

DESIGN NOTES #2

KNOWN BUGS!

A. Will gain a Days Count when it comes light when its going dark due to a transient dark cloud. (so what, spreadsheet captures it all with dates so it "Saul Goodman"
B. When at night and Unit is reset for what ever reason the Day count Starts at #1 But it's logical as it counts NIGHTS after LIGHT-TIME, after that it sorts itself out and its 'Saul Goodman!
C. Temperature calibraation looks ok +/- 2C to the dry tongue test
D. Watch out 'cos when voltage low it could send 1000 texts/day! Think i have stopped this now.
F. Warning, removed wdt as got rapid flashing of pin 13 LED when it was activitated. Gave mega problems loading new shetch!!!!
E. Stress releive of fixing holes. no tight screws to pre-stress vero board

---

ACTION LIST:

1. version 2 tidy up and test blind routes 2020...
2.
3.
4.

DESIGN REFERENCE SECTION

WatchDog reset	https://www.youtube.com/watch?v=hT24hOTrbEI possibly the best way via WatchDogTimer keep kicking by wdt_reset(); DOES NOT WORK reliably with arduino mini pro SO HAVE REMOVED IT!!!
also see	https://playground.arduino.cc/Main/ArduinoReset or 47mF and relay
RESET	https://www.youtube.com/watch?v=vt9C-hMa1Ok
and	https://www.codeproject.com/Articles/1012319/Arduino-Software-Reset
sleep	https://github.com/n0m1/Sleep_n0m1/tree/master/examples/Sleep_n0m1_simple
sleep inturrupt	https://github.com/n0m1/Sleep_n0m1/blob/master/examples/Sleep_n0m1_Interrupt/Sleep_n0m1_Interrupt.ino
INTERRUPTS	https://www.allaboutcircuits.com/technical-articles/using-interrupts-on-arduino/
	https://www.arduino.cc/en/Reference/AttachInterrupt = attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);
	https://www.arduino.cc/en/Reference/Interrupts = noInterrupts(); and interrupts();
Bounce	https://github.com/maykon/ButtonDebounce can't understand this yet so use capacitor!
STRINGs	https://www.arduino.cc/en/Tutorial/StringAdditionOperator to do eng sms battery voltage each month?
	STRING TO CHAR https://forum.arduino.cc/index.php?topic=401150.0
	STRING TUTORIAL http://www.tutorialspoint.com/c_standard_library/string_h.htm
Voltage	https://github.com/Yveaux/Arduino_Vcc/tree/master/examples

```
                        https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage
                        https://jeelabs.org/2012/05/04/measuring-vcc-via-the-bandgap/
  SIM800l               https://github.com/VittorioEsposito/Sim800L-Arduino-Library-revised/blob/master/examples/sendSms/sendSms.ino
                        https://www.youtube.com/watch?v=-okAX7ZoGDk  best one & reserve https://www.youtube.com/watch?v=9BUDri6Qi9o and
                        https://github.com/VittorioEsposito/Sim800L-Arduino-Library-revised/issues/2
                        https://stackoverflow.com/questions/42158230/sim800l-string-concatenation    and  https://cristiansteib.github.io/Sim800l
                        and commands  https://www.exploreembedded.com/wiki/Setting_up_GPRS_with_SIM800L
                        http://www.ayomaonline.com/programming/quickstart-sim800-sim800l-with-arduino/  and  https://www.youtube.com/watch?v=XqoeqG2j-ik
                        and https://www.youtube.com/watch?v=-okAX7ZoGDk
                        https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/ just  found 28/03/2020
                        https://developer.gemalto.com/threads/unlock-sim-require-pin-code    just  found 28/03/2020
  sprintf               http://forum.arduino.cc/index.php?topic=46884.0   https://gist.github.com/philippbosch/5395696
                        https://liudr.wordpress.com/2012/01/16/sprintf/
  sprintf               http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html with FLOAT yea!
                        https://dereenigne.org/arduino/arduino-float-to-string/
                        sprintf(combinedText, "Current temperature is %i degrees.", temperature);
  temperature           https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage   and
                        https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v32/experiment-7-reading-a-temperature-sensor
                        http://yaab-arduino.blogspot.co.uk/2015/12/how-to-sprintf-float-with-arduino.html
  CONVERT               INTERGER into a Character from http://forum.arduino.cc/index.php?topic=46884.0
  Temperature           Consider using internal temp of chip as no external components
                        https://playground.arduino.cc/Main/InternalTemperatureSensor
  Text formatting       Remember that \r gives a caridge return in a text massage    see \rPUMP-COUNT = " + clunkCount + ":

*/

// Andrew WS Ainger BSc. CEng. FIET.
// October 2020
//-----------------------------------------------------------------------------------------------------------------------------------------------
```