```c
/*
Monday, 19 May 2014
ATS Program  - Button Controlled.
uC: PIC16F628A
Fosc: 4MHz
Author: N. Chitiyo (nchitiyo@sirdc.ac.zw)
NOTE: use COFF file for debugging in Proteus ISIS
Software Rev: 0.2.2f (ATS)
Button Software Version: 0.1

Software Revisions:
0.0.1    : ATS. Auto only
0.0.2    : ATS. With Cooldown Timer
0.0.3    : ATS with key-override control
0.1.0    : software v0.0.3 for board version 0.1 (Split board design)
0.2.0    : software v0.1.0 for board version 0.2T(Split-board - Cable connected)
0.2.1    : software v0.2.0 for board version 0.2U (ULN driver version)
0.2.2    : ATS Software Cleaned up. for Documentation Purposes.
0.2.2f   : v0.2.2 implementing Feedback

Hardware Version: v0.2U (With ULN)
Hardware Revisions:
0.0      : ATS Board - Debug Version Prototype. uni-Board, Transistor Driven
0.1      : Split version. dual-board, joined by headers. Transistor Driven
0.2      : Split version dual board, joined by ribbon cable/UTP
0.2U     : dual, ULN-driven, Changeover Relay on board 1 with gen feedback


*NOTE: There's need to Edit the CoolDown Routine!! introduce a cooldown Poll
routine to fix this problem: to be looked into
*NOTE: there is need for a pre-changeover poll to verify the absence of ZESA for
 one more time, before switching GENENRATOR power to LOAD
*/

// ALL Important Functions defined and/or declared in "Initializers.c"

unsigned int ProgTimer, RealTimer, RunTimer, CoolDownTimer;
bit Auto_Flag, Run_Flag, CoolDown_Flag;
bit GenFeedFlag;         //feedback indirect register
unsigned int RunTime, CoolDownTime;
unsigned short RBValue, CrankTrials;

/*Input-Output Table:
     *  PIN  |  I/O  |  Assign                   |     Notes
     *-------------------------------------------|---------------
     *  RA0  |Output |  SQOUT/ N/A               |    Clock Count
     *  RA1  |Input  |  N/A                      |
     *  RA2  |Input  |  ZESA Sense               |
     *  RA3  |Output |  N/A                      |
     *  RA4  |Input  |  Gen Feedback             |Sense that Gen is ON
     *  RB0  |Input  |  OFF   Button             |Manual GenSTOP interrupt en.
     *  RB1  |Input  |  Start                    |Manual GenStart
     *  RB2  |Input  |  Auto                     |GenAuto
     *  RB3  |Input  |  ON                       |GenON
     *  RB4  |Output |  GenSTOP Control (NC!!!)  |
     *  RB5  |Output |  ChangeOver Control (NO)  |
     *  RB6  |Output |  GenStart Control (NO)    | crank.
     *  RB7  |Output |  GenOn Control (N.O)      |
     *-------------------------------------------|---------------
*/
sbit SQOUT at RA0_Bit;
sbit ZESA at RA2_Bit;
sbit feedback at RA4_bit;
sbit GenStop at RB4_Bit;
sbit ChangeOver at RB5_Bit;
sbit GenStart at RB6_Bit;
sbit GenOn at RB7_Bit;

// ///////////Declare Functions //////////////
void crank();
void Poll();
/////////////////////////////////////////////


void interrupt() {       //TMR0 Interrupt Handler
        GIE_Bit = 0;
        if (T0IF_Bit) {

           T0IF_Bit = 0;
           TMR0 = 0;
           ProgTimer++;
           if (progTimer == 1953)
           {
               RealTimer++;

               //Check Run and CoolDown Status
               if (RunTimer == RunTime) {
                  RunTimer = 0;
                  Run_flag = 0;
                  CoolDown_Flag = 1;
               }

               if (CoolDownTimer == CoolDownTime) {
                  CoolDownTimer = 0;
                  //Run_flag = 1;   //The Run Flag Should Be Started Elsewhere
                  CoolDown_Flag = 0;
           }

           if (Run_Flag && Feedback) RunTimer++;
           if (CoolDown_Flag) CoolDownTimer++;
           SQOUT=~SQOUT;
           progTimer = 0;
           }
        }
        if (INTF_Bit) {           //Pressing "Stop" forced Stop, Turn Off all
           INTF_Bit = 0;
           PORTB = 0;
           CrankTrials = 6;      //just to kick it out of the crank...
           RBValue = 19;          //so that it falls into "Default"
           Auto_Flag = 0;
           Run_Flag = 0;
        }
        GIE_Bit = 1;
}


void main() {

//set the runtime (4 hours) and cooldown time (1 hour)

      //RunTime = 20;         //Testing Purposes
      RunTime = 14388;   //4 hours - 12 seconds run time
      //CoolDownTime = 20;    //Testing Puroses
      CoolDownTime = 3599;   //1 hour (- 1 second) cooldown time

      PORTA = 0;
      TRISA = 0b10110;               //following the Table Above
      PORTB = 0;
```

```
          TRISB = 0b00001111;            //Following the Table Above
          CMCON = 0x0F;                  //PORTA all digital

          T0IE_Bit = 1;           //Enable TMR0 Interrupt
          INTE_Bit = 1;          //Enable RB0 Interrupt...
          INTEDG_Bit = 1;        //...on rising edge of RB0
          GIE_bit = 1;

          // /////////Timer Configuration ////////////////////

          T0CS_Bit = 0;    //Select Timer Mode. Timer Starts Now
          TMR0 = 0;        //reset the TMR0 Register
          PSA_Bit = 0;     //Assign Prescaler from WDT to Timer0 when value = 0
          OPTION_REG &=248;         //Clear Previous Prescaler Values
          OPTION_REG |=0;     //set Prescaler to TimeSet  (1:2)

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


    while(1) {                           //Main Endless Loop

         delay_ms(100);                     //delay for latency (To allow system
                                            //to tolerate key debounces)

         //collect the last three values input from the ignition switch:
         if (RBValue != 19) RBValue = PORTB & 0x07; //19 is the fallback value


         if (!Run_flag && CoolDown_flag) RBValue = 19;      //it's time to cool down

         /* NOTE: When the Gen Cools down, you have to Start
            it manually after coolddown.
         */

         /* RBValue Mode Coding Table
                         |Auto |Start |Off |
                         |-----|------|----|--------------|
            * Condition  |RB2  |RB1   |RB0 | Hex | Dec    |
            *------------|-----|------|----|-----|--------|
            *  Auto      | 1   | x    | 0  | 0x04| 4,6    |
            *  Start     | 0   | 1    | 0  | 0x02| 2      |
            *  OFF       | x   | x    | 1  | 0x01|1,3,5,7 |
            *  Maintain  | 0   | 0    | 0  | 0x00| 0      |
            *----------------------------------------------
          */

         switch(RBValue) {

         // /////////////Auto Mode/////////////////
         case 4:                                   //Gen Auto Mode Selected
              Auto_Flag = 1;                     // Gen is in Auto Mode
              RBValue = 0;
              if (Auto_Flag) Poll();
              break;

         case 6:                                   //Gen Auto Mode Selected
              Auto_Flag = 1;                     // Gen is in Auto Mode
              RBValue = 0;
              if (Auto_Flag) Poll();
              break;

         // /////////////////System Start/Crank ////////////////////

         case 2:                                   //Start Button Pressed
```

```
          Auto_Flag = 0;                     //Gen is in Manual Mode
          GenStop = 1;                         //Turn Off GEN_OFF signal
          GenOn = 1;
          crank();                             //crank till there's feedback
          RBValue = 0;
          break;

     // ////////////////Maintain Scenario ///////////////

     case 0:
          Auto_Flag = 0;
          RBValue = 0;
          break;

     // ///////////////Off and Default Scenario//////////

     default:
          Auto_Flag = 0;
          PORTB = 0;       //Turn the Gen Off if none of the conditions are met
          delay_ms(5000);
          if (!feedback) Run_Flag = 0; //Ensure Gen is off
          RBValue = 0;
          break;         //We need an ERROR Condition Here
     } //switch


 }//While

}

// ///Crank Part is Working!!! //////////////////////////////////////

void crank() { //Crank till there's feedback. 5X2 seconds before failing.
GenFeedFlag = feedback;
CrankTrials = 0;
while (!GenFeedFlag && CrankTrials <5) {
   GenStart = 1;
   delay_ms(2000);
   CrankTrials++;
   GenFeedFlag = feedback;
   if (CrankTrials == 5) {
      CrankTrials = 0;
      GenFeedFlag = 1;
      PORTB = 0;
      Run_Flag = 0;
      RBValue = 19;  //Falls to "Default
      //break;
      }
   else Run_Flag = 1;
GenStart = 0;      //Stop Cranking
} //while
}//crank

// //////Polling Function ///////////////////
void Poll() {
     if (Auto_Flag && ZESA) {
        ChangeOver = 0;    // make sure you revert to ZESA Supply
        delay_ms(2000);
        RBValue = 19;       //Fall to Default
        }
        else {             //If ZESA is not there,
          delay_ms(1000);    //Just wait...
           if (!Run_flag) {  //if the Gen is running, leave and Poll
              if (!CoolDown_Flag) {
                GenStop = 1;
                GenOn = 1;
                crank();   //...and run the Generator Start Routine.
                Delay_ms(5000);     //Stabilize
                if (RBValue ==4 || RBValue == 6 && !ZESA) {
                   ChangeOver = 1;
                   Run_Flag = 1;
                   }//if AutoMode
                }//if !CoolDown_Flag
              } // if !Run Flag
           }//else ZESA is not there
} //Poll
```