```
#include <ModbusRtu.h>

// data array for modbus network sharing
uint16_t au16data[16] = {
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, -1 };

//This sets up the MODBUS
Modbus slave(1,0,0); // this is slave @1 and RS-232 or USB-FTDI

void setup() {
  slave.begin( 19200 ); // baud-rate at 19200 for modbus
  pinMode(72, OUTPUT); //sets pin as output for stepper motor
  pinMode(73, OUTPUT); //sets pin as output for stepper motor
  pinMode(74, OUTPUT); //sets pin as output for stepper motor
  pinMode(75, OUTPUT); //sets pin as output for stepper motor
  pinMode(76, OUTPUT); //sets pin as output for stepper motor
  pinMode(77, OUTPUT); //sets pin as output for stepper motor
  pinMode(78, OUTPUT); //sets pin as output for stepper motor
  pinMode(79, OUTPUT); //sets pin as output for stepper motor
  pinMode(80, OUTPUT); //sets pin as output for relay (generator off)

  digitalWrite(73, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(75, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(72, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(74, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(77, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(79, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(76, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(78, LOW); //ensures pin is set to low so that stepper motor can be manually spun
  digitalWrite(80, LOW); //ensures pin is set to low so that generator doesn't get turned off on
accident
}

void loop() {
  int Vup = 0; //integer for copying positive voltage change commands
  int Vdown = 0; //integer for copying negative voltage change commands
  int Fup = 0; //integer for copying positive frequency change commands
  int Fdown = 0; //integer for copying negative frequency change commands


  while(1){
    slave.poll( au16data, 16 ); //polls modbus registers
    if(au16data[6]!=0){ //checks if stop! command has been sent and then clears out rest of
registers if it has been
      au16data[0] = 0;
      au16data[1] = 0;
      au16data[2] = 0;
      au16data[3] = 0;
      au16data[4] = 0;
      au16data[5] = 0;
    }
    /*if (au16data[0]==1){
      GenOn();
      au16data[0]=0;
    }*/
    //This code removed because Gen On was done with RTAC
    if (au16data[1]==1){ //checks for generator off command
      GenOff();
      au16data[1]=0;
    }
```

```
    if(au16data[2]!=0){ //checks for voltage up value
      Vup = au16data[2]; //sets Vup value for voltage up change
      VStepperControl(1, Vup); //calls funtion for stepper motor and passes direction and voltage
change value
      au16data[2] = 0; //clears voltage up change MODBUS register
    }
    if(au16data[3]!=0){ //checks for voltage down value
      Vdown = au16data[3]; //sets Vdown value for voltage down change
      VStepperControl(0, Vdown); //calls funtion for stepper motor and passes direction and voltage
change value
      au16data[3] = 0; //clears voltage down change MODBUS register
    }
    if(au16data[4]!=0){ //checks for frequency up value
      Fup = au16data[4]; //sets Fup value for frequency up change
      FStepperControl(1, Fup); //calls funtion for stepper motor and passes direction and frequency
change value
      au16data[4] = 0; //clears frequency up change MODBUS register
    }
    if(au16data[5]!=0){ //checks for frequency down value
      Fdown = au16data[5]; //sets Fdown value for frequency down change
      FStepperControl(0, Fdown); //calls funtion for stepper motor and passes direction and
frequency change value
      au16data[5] = 0; //clears frequency down change MODBUS register
    }
  }
}

//function for controlling the stepper motor that controls voltage
void VStepperControl(int VDir, int VoltValue){
  int VDelay;
  int VC = 3; //voltage constant. change this to change the rate the voltage change value is
amplified by
  int turn = VC*VoltValue;
  if (VoltValue <= 5){
    VDelay = 25;
  }
  else if (VoltValue <= 10){
    VDelay = 20;
  }
  else if (VoltValue <= 15){
    VDelay = 15;
  }
  else if (VoltValue <= 20){
    VDelay = 10;
  }
  else if (VoltValue <= 25){
    VDelay = 5;
  }
  else{
    turn = VC*25;
    VDelay = 5;
  }
  for (int i=turn; i>0; i--){
    slave.poll( au16data, 16 );
    if(au16data[6]!=0){
      au16data[0] = 0;
      au16data[1] = 0;
      au16data[2] = 0;
      au16data[3] = 0;
      au16data[4] = 0;
```

```
      au16data[5] = 0;
      return;
    }
    if (VDir == 1){
      digitalWrite(73, LOW);
      digitalWrite(75, LOW);
      digitalWrite(72, HIGH);
      digitalWrite(74, HIGH);
      delay(VDelay);
      digitalWrite(72, LOW);
      digitalWrite(75, LOW);
      digitalWrite(73, HIGH);
      digitalWrite(74, HIGH);
      delay(VDelay);
      digitalWrite(72, LOW);
      digitalWrite(74, LOW);
      digitalWrite(73, HIGH);
      digitalWrite(75, HIGH);
      delay(VDelay);
      digitalWrite(73, LOW);
      digitalWrite(74, LOW);
      digitalWrite(72, HIGH);
      digitalWrite(75, HIGH);
      delay(VDelay);
    }
    else{
      digitalWrite(73, LOW);
      digitalWrite(74, LOW);
      digitalWrite(72, HIGH);
      digitalWrite(75, HIGH);
      delay(VDelay);
      digitalWrite(72, LOW);
      digitalWrite(74, LOW);
      digitalWrite(73, HIGH);
      digitalWrite(75, HIGH);
      delay(VDelay);
      digitalWrite(72, LOW);
      digitalWrite(75, LOW);
      digitalWrite(73, HIGH);
      digitalWrite(74, HIGH);
      delay(VDelay);
      digitalWrite(73, LOW);
      digitalWrite(75, LOW);
      digitalWrite(72, HIGH);
      digitalWrite(74, HIGH);
      delay(VDelay);
    }
  }
  digitalWrite(73, LOW);
  digitalWrite(75, LOW);
  digitalWrite(72, LOW);
  digitalWrite(74, LOW);
  return;
}

//function for controlling the stepper motor that controls frequency
void FStepperControl(int FDir, int FreqValue){
  int FDelay;
  int FC = 6;  //frequency constant. change this to change the rate the frequency change value is
amplified by
```

```
int turn = FC*FreqValue;
if (FreqValue <= 10){
  FDelay = 25;
}
else if (FreqValue <= 20){
  FDelay = 20;
}
else if (FreqValue <= 30){
  FDelay = 15;
}
else if (FreqValue <= 40){
  FDelay = 10;
}
else if (FreqValue <= 50){
  FDelay = 5;
}
else{
  turn = FC*50;
  FDelay = 5;
}
for (int i=turn; i>0; i--){
  slave.poll( au16data, 16 );
  if(au16data[6]!=0){
    au16data[0] = 0;
    au16data[1] = 0;
    au16data[2] = 0;
    au16data[3] = 0;
    au16data[4] = 0;
    au16data[5] = 0;
    return;
  }
  if (FDir == 1){
    digitalWrite(77, LOW);
    digitalWrite(79, LOW);
    digitalWrite(76, HIGH);
    digitalWrite(78, HIGH);
    delay(FDelay);
    digitalWrite(76, LOW);
    digitalWrite(79, LOW);
    digitalWrite(77, HIGH);
    digitalWrite(78, HIGH);
    delay(FDelay);
    digitalWrite(76, LOW);
    digitalWrite(78, LOW);
    digitalWrite(77, HIGH);
    digitalWrite(79, HIGH);
    delay(FDelay);
    digitalWrite(77, LOW);
    digitalWrite(78, LOW);
    digitalWrite(76, HIGH);
    digitalWrite(79, HIGH);
    delay(FDelay);
  }
  else{
    digitalWrite(77, LOW);
    digitalWrite(78, LOW);
    digitalWrite(76, HIGH);
    digitalWrite(79, HIGH);
    delay(FDelay);
    digitalWrite(76, LOW);
```

```
      digitalWrite(78, LOW);
      digitalWrite(77, HIGH);
      digitalWrite(79, HIGH);
      delay(FDelay);
      digitalWrite(76, LOW);
      digitalWrite(79, LOW);
      digitalWrite(77, HIGH);
      digitalWrite(78, HIGH);
      delay(FDelay);
      digitalWrite(77, LOW);
      digitalWrite(79, LOW);
      digitalWrite(76, HIGH);
      digitalWrite(78, HIGH);
      delay(FDelay);
    }
  }
  digitalWrite(77, LOW);
  digitalWrite(79, LOW);
  digitalWrite(76, LOW);
  digitalWrite(78, LOW);
  return;
}

/*void GenOn(void){
  delay(250);
  digitalWrite(80, HIGH);
  delay(2000);
  digitalWrite(80, LOW);
  delay(250);
  return;
}*/
//This code removed because Generator On was sent through the RTAC

//this code turns off the generator
void GenOff(void){
  delay(250);
  digitalWrite(80, HIGH);
  delay(4000);
  digitalWrite(80, LOW);
  delay(250);
  return;
}
```