

```

// *****
// * 20/5/15 *
// * Wiper control software wipers_8 *
// * Controls PWM to motor in response to *
// * the switch and a current overload sensor. *
// * wipers_8 *
// * by Julian Rogers *
// *****

//Program checks overload conditions. With a mild overload, the program attempts to
//park the wipers and waits for the stalk to be turned to "off" to reset.
//With a heavy overload the wipers stop wherever they are until reset as above.
//It is important to measure average current rather than take a single instantaneous
//value (except I don't follow this for really heavy current once the motor has started)
//(It's also important to give the motor a chance to get going and not try to measure
//the starting current which will always be very high)

//The system measures the position of a six way switch. Position 1 is normal
//ie. Off, slow, fast, single from the stalk. Positions 2 - 5 give delayed action on
//stalk set to slow. Off and fast over ride these. Position 6 is an experimental auto
//position whereby the wipers run at slow until current measurement detects a drying
//screen when a five second delay is introduced. Further drying causes the wipers
//to be parked until reset as above. The "drying" current value is set by the parameter
//potentiometer

boolean flag = false; // excess current flag
int maxcurr; // max current in fn switch pos 6
long totcurr = 0;
long avcurrent = 0;
int pass6 = 0;
int pass = 0; // counts passes through loop
int valD1; //slow switch status - LOW on
int valD2; //fast switch status - LOW on
int count = 0; //counts passes when current is excessive
int valPark = LOW; //park switch status - HIGH is in parked position
int motorPin = 10; // motor connected to digital pin 10
int currentPin = 0; // current sensor connected to pin A0
int paramPin = 1; // pot connected to A1
int parkPin = 13; // park switch
int stopPin = 12; // stop crowbar mosfet
int d2Pin = 9; // stalk switch
int d1Pin = 8; // stalk switch
int overvoltPin = 11; // pwm drive for 18 volt
// pins for 6-way switch are 2,3,4,5,6,7

long timeThen;
long timeNow;
long timeSince;

int current; // motor current value - goes down as current goes up!!
int param; //parameter from pot

//-----

void park() //parks the wipers
{
  valPark = digitalRead(parkPin);
  if(valPark == LOW)
  {
    analogWrite(motorPin, 180); //start motor in slow

    timeThen = millis();
    do
    {
      current = analogRead(currentPin);
      if (current < 200){
        analogWrite(motorPin, 0); //stop motor immediately ic current really high
        Serial.println("stopped in park");
        flag = true;
      }
    }
  }
}

```

```

    }
    valPark = digitalRead(parkPin);
    timeNow = millis();
    timeSince = timeNow - timeThen;
}while(valPark == LOW && timeSince < 2000);
delay(100);
analogWrite(motorPin, 0);    // if park position reached or 2 sec have passed
                             // stop motor and apply brake

    digitalWrite(stopPin, HIGH);
    delay(500);
    digitalWrite(stopPin, LOW);
}
}
}

//-----

void wiperDelay(int delayParam)    // initiates 1 sweep of wipers then waits for
                                   // delay period. Fast or stop over-rides this
//delayParam is delay between wipes in secs.
{
    delay(500);    //get wipers off stop switch and on wipe
    park();
                                   //wait during delay period or until fast selected
    timeThen = millis();
    do {
        valD1 = digitalRead(d1Pin);
        valD2 = digitalRead(d2Pin);
        timeNow = millis();
        timeSince = timeNow - timeThen;
    }while(timeSince < delayParam*1000 && valD2 == HIGH && valD1 == LOW);
}

//-----

void setup()
{
    pinMode(parkPin, INPUT);
    pinMode(motorPin, OUTPUT);
    pinMode(overvoltPin, OUTPUT);
    pinMode(stopPin, OUTPUT);
    pinMode(d2Pin, INPUT);
    pinMode(d1Pin, INPUT);
    pinMode(7, INPUT);
    pinMode(6, INPUT);
    pinMode(5, INPUT);
    pinMode(4, INPUT);
    pinMode(3, INPUT);
    pinMode(2, INPUT);

    analogWrite(overvoltPin, 128);    // generates overvoltage
    analogWrite(motorPin, 0);    // wiper motor off
    digitalWrite(stopPin, LOW);    // stop mosfet off
    Serial.begin(9600);    // starts serial comms for debugging
}

//-----

void loop()
{
    int param = analogRead(paramPin);
    maxcurr = map(param, 0, 1023, 300, 500);
    Serial.print("max current ");
    Serial.println(maxcurr);
    // get values from switches
    int valD2 = digitalRead(d2Pin);
    int valD1 = digitalRead(d1Pin);
    int sixWay1 = digitalRead(2);
    int sixWay2 = digitalRead(3);
    int sixWay3 = digitalRead(4);
    int sixWay4 = digitalRead(5);
}

```

```

int sixWay5 = digitalRead(6);
int sixWay6 = digitalRead(7);

pass++;
//Serial.print(current);
//Serial.print(" ... ");
current = analogRead(currentPin);

//Serial.print("main loop current ");
//Serial.println(current);

if (current < 300){ //count every time current is above a certain value
count++;
if(pass == 1000)
{
count = 0;
pass = 0;
}
Serial.println(count);
if(count > 99){ //if it happens 100 times the park!
park();
flag = true; //set flag to indicate overload has taken place
count = 0;
Serial.println("stopped in main loop");
}
}

if(valD2 == LOW && flag == false) // switch on high, no overcurrent detected
{
analogWrite(motorPin, 255);
}

else if(valD1 == LOW && flag == false) // switch on low, no overcurrent detected
{
analogWrite(motorPin, 180); //run wipers slow

// detect position of fn switch (only on slow)

if(sixWay1 == LOW){ // pos 1, continuous
}
else if(sixWay2 == LOW){ // pos 2 to 5, increasing delays
wiperDelay(1);
}

else if(sixWay3 == LOW){
wiperDelay(2);
}

else if(sixWay4 == LOW){
wiperDelay(5);
}

else if(sixWay5 == LOW){
wiperDelay(12);
}

else if(sixWay6 == LOW){ // "auto" function on pos 6
//Serial.println("%");
pass6++;
current = analogRead(currentPin);
totcurr = totcurr + current;
//delay(500);
if(pass6 > 299)
{
avcurrent = totcurr / pass6; // calculate average current every 300 passes
Serial.print("average current...");
Serial.println(avcurrent);
pass6 = 0;
totcurr = 0;
}
}

```

```
    }
    if(avcurrent < maxcurr && avcurrent > 0){ // 5 sec delay if overcurrent
        // overcurrent threshold set by pot
        avcurrent = 0;
        wiperDelay(5);
    }
}
}
if(valD2 == HIGH && valD1 == HIGH)
{
    flag = false; //reset if stalk switch is off so wipers can be turned on again
    count = 0;
    pass = 0;
    park();
}
}
```