

```
/* boost converter guide at reibot.org
```

```
feedback connected to analog 1.  
pwm pin is on digital 11
```

```
V 1 for the arduino pid library  
*/
```

```
#include <PID_v1.h>
```

```
//Define Variables we'll be connecting to  
double Setpoint, Input, Output;
```

```
//Specify the links and initial tuning parameters  
PID myPID(&Input, &Output, &Setpoint, .8, .1, .01, DIRECT);
```

```
int targetVoltage = 100; //up to 200v if you used same parts from the guide
```

```
void setup(){
```

```
  Serial.begin(115200);
```

```
  pwmSetup(); //set the pwm to 31khz
```

```
  myPID.SetMode(AUTOMATIC);
```

```
  myPID.SetOutputLimits(0, 100);
```

```
  //how often you want the pid to update in ms
```

```
  myPID.SetSampleTime(10); //changing this will require pid to be retuned
```

```
}
```

```
void loop(){
```

```
  PidUpdate();
```

```
  delay(10);
```

```
  //call PidUpdate(); at least every 10ms.
```

```
  //Otherwise set OCR2A = 0; while the arduino is doing something time consuming
```

```
}
```

```
void PidUpdate(){
```

```
  //using resistive voltage divider, same R1 and R2 on wikipedia "voltage divider"
```

```
  //R1 is connected to high voltage
```

```
  //R2 is connected to ground
```

```
  double R2 = 200000;
```

```
  double R1 = 10000000;
```

```
  double voltage = (analogRead(A1)/1023.)*4.9*(R2+R1)/R2;
```

```
  Input = voltage;
```

```
  Setpoint = targetVoltage;
```

```
myPID.Compute();
```

```
if(vol tage > targetVol tage + 10){
```

```
    OCR2A = 0;
```

```
}else if(vol tage <= targetVol tage){
```

```
    OCR2A = Output;
```

```
}
```

```
    //have it output the vol tage
```

```
/*
```

```
Serial.print("V: ");
```

```
Serial.print(vol tage);
```

```
Serial.print("\tADC vol tage: ");
```

```
Serial.print((analogRead(A1)/1023.)*4.9);
```

```
Serial.print("\tduty: ");
```

```
Serial.println((int)OCR2A);
```

```
delay(10);
```

```
*/
```

```
}
```

```
void pwmSetup(){//just run once at setup
```

```
pinMode(3, OUTPUT); //OCR2B 3 and 11 are pwm channels
```

```
pinMode(11, OUTPUT); //OCR2A
```

```
TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM20); //phase correct pwm 31250hz
```

```
TCCR2B = _BV(CS20); //change this as datasheet says to mainly get different pwm frequencies
```

```
OCR2A = 0;
```

```
OCR2B = 0;
```

```
}
```