```
#include <avr/io.h>
#include <avr/portpins.h>
/*
Program to test the 8 Channel Analog-to-Digital Converter using a Microchip
PIC
as found on Marc Bertrand's website at
http://marcpic.com/Projects/8CHADCwithPIC/ADC8CHWithPIC.htm
Channel Selection Table
CHS3  CHS2  CHS1  CHS0   ADC Channel
-------------------------------------
 0     0     0     0    0 (LSB value)
 1     0     0     0    0 (MSB value)
 0     0     0     1    1 (LSB value)
 1     0     0     1    1 (MSB value)
 0     0     1     0    2 (LSB value)
 1     0     1     0    2 (MSB value)
 0     0     1     1    3 (LSB value)
 1     0     1     1    3 (MSB value)
 0     1     0     0    4 (LSB value)
 1     1     0     0    4 (MSB value)
 0     1     0     1    5 (LSB value)
 1     1     0     1    5 (MSB value)
 0     1     1     0    6 (LSB value)
 1     1     1     0    6 (MSB value)
 0     1     1     1    7 (LSB value)
 1     1     1     1    7 (MSB value)
-------------------------------------
Connect the Data Byte Output PORTC of PIC16F767 to UNO pins D2 to D9.
*/
const byte CHS0 = A0;
const byte CHS1 = A1;
const byte CHS2 = A2;
const byte CHS3 = A3;
const byte CS_not = A4; //Chip Select - active LOW
const byte OE_not = A5; //Output Enable - active LOW
unsigned int ADCvalue;  //Holds value of A/D conversion
byte Temp;

void setup() {
  for (int i=2;i<=9;i++){
    pinMode(i,INPUT_PULLUP);
  }
 pinMode(CHS0,OUTPUT);
```

```
 pinMode(CHS1,OUTPUT);
 pinMode(CHS2,OUTPUT);
 pinMode(CHS3,OUTPUT);
 pinMode(CS_not,OUTPUT);
 pinMode(OE_not,OUTPUT);
 digitalWrite(CS_not,HIGH);
 digitalWrite(OE_not,HIGH);
 Serial.begin(115200);
}

void loop() {
  Serial.println("Microchip ADC Test");
 Serial.println("------------------");
  Serial.println("0 - CH0");
  Serial.println("1 - CH1");
  Serial.println("2 - CH2");
  Serial.println("3 - CH3");
  Serial.println("4 - CH4");
  Serial.println("5 - CH5");
  Serial.println("6 - CH6");
  Serial.println("7 - CH7");
 Serial.println("?>");
  while (!Serial.available());  //Wait for channel #
  byte ChNum=Serial.read();
  switch (ChNum){
    case '1':
      digitalWrite(CHS0,HIGH);
      digitalWrite(CHS1,LOW);
      digitalWrite(CHS2,LOW);
      CommonForAllCH();
      Serial.print("CH1=");
      Serial.println(ADCvalue);
       break;
    case '2':
      digitalWrite(CHS0,LOW);
      digitalWrite(CHS1,HIGH);
      digitalWrite(CHS2,LOW);
      CommonForAllCH();
      Serial.print("CH2=");
      Serial.println(ADCvalue);
       break;
    case '3':
      digitalWrite(CHS0,HIGH);
```

```
  digitalWrite(CHS1,HIGH);
  digitalWrite(CHS2,LOW);
  CommonForAllCH();
  Serial.print("CH3=");
  Serial.println(ADCvalue);
   break;
case '4':
  digitalWrite(CHS0,LOW);
  digitalWrite(CHS1,LOW);
  digitalWrite(CHS2,HIGH);
  CommonForAllCH();
  Serial.print("CH4=");
  Serial.println(ADCvalue);
   break;
case '5':
  digitalWrite(CHS0,HIGH);
  digitalWrite(CHS1,LOW);
  digitalWrite(CHS2,HIGH);
  CommonForAllCH();
  Serial.print("CH5=");
  Serial.println(ADCvalue);
   break;
case '6':
  digitalWrite(CHS0,LOW);
  digitalWrite(CHS1,HIGH);
  digitalWrite(CHS2,HIGH);
  CommonForAllCH();
  Serial.print("CH6=");
  Serial.println(ADCvalue);
   break;
case '7':
  digitalWrite(CHS0,HIGH);
  digitalWrite(CHS1,HIGH);
  digitalWrite(CHS2,HIGH);
  CommonForAllCH();
  Serial.print("CH7=");
  Serial.println(ADCvalue);
   break;
default:
  digitalWrite(CHS0,LOW);
  digitalWrite(CHS1,LOW);
  digitalWrite(CHS2,LOW);
  CommonForAllCH();
```

```
     Serial.print("CH0=");
     Serial.println(ADCvalue);
  }
}

void CommonForAllCH(){
  digitalWrite(CHS3,HIGH);  //Get MSB value first
  digitalWrite(CS_not,LOW); //Strobe or signal ADC ready to read A/D value
 digitalWrite(CS_not,HIGH);
  digitalWrite(OE_not,LOW); //Read the A/D value
 delayMicroseconds(50);
  Temp=(PINB << 6)+(PIND >> 2);
 ADCvalue=Temp*256;
 digitalWrite(OE_not,HIGH);
  digitalWrite(CHS3,LOW);   //Get LSB value
  digitalWrite(CS_not,LOW); //Strobe/signal ADC ready to read A/D value
 digitalWrite(CS_not,HIGH);
  digitalWrite(OE_not,LOW); //Read the A/D value
 delayMicroseconds(50);
  Temp=(PINB << 6)+(PIND >> 2);
 ADCvalue+=Temp;
 digitalWrite(OE_not,HIGH);
}
```