

```

//
// DAC Control Program
//
// this file = DAC_Sketch_1.ino
//
// Uses Arduino Pro Trinket 3.3v (FTDI).
// Also uses USB to FTDI adapter to load program
// and for realtime control via serial monitor
// interface at 1200 baud.
//
// Arduino pins:
// Analog 4 = SDA
// Analog 5 = SCL
// Analog 2,3 = Control input Pot(s) wiper, CW = 3.3v, CCW=gnd
// Analog 1 = momentary pushbutton, n.o. = 3.3v, n.c. = gnd
// Digital 8 = I2C bus seizure relay, 1 = on (seize), 0 = off (release)
//
// Tested serial monitors include Arduino Serial Monitor
// and the freeware 'putty' terminal emulator
//
// To find serial port number to use with putty, for
// Windows open ControlPanel>DeviceManager then
// select 'show hidden devices' from the DeviceManager
// 'View' menu. Serial interface should appear under
// 'Ports (COM & LPT)'
//
// After locating the FTDI based COM port, the default
// COM port number can be used to open Trinket program
// user interface in putty. Also, COM port number for the FTDI Com
// device can be changed in Properties>Port Settings>Advanced
//
// Note: Used good working order below of #define, #include, and
// global type declarations section before start of setup()
//

#define VERSION "1.3"
#define USEEEPROM 1
#define I2C_HARDWARE 1

#include <ctype.h>
#include <stdio.h>
#include <string.h>
#include <EEPROM.h>

// constants for I2C interface
#define I2C_TIMEOUT 100
#define I2C_PULLUP 0
#define SDA_PORT PORTC
#define SDA_PIN 4 // = C4
#define SCL_PORT PORTC
#define SCL_PIN 5 // = C5

#include <SoftI2CMaster.h>

void setup() {
    pinMode(8, OUTPUT);
    i2c_init();
    Serial.begin(1200);
    Serial.println("\n\n*** DAC Control & Configuration ***");
}

```

```

}

void loop() {

int C2r;          // variable to hold C2 read value
int C2w;          // variable to hold C2 write value
int C3r;
int C3w;
int choice;       // menu selection choice
byte RegAddL;     // address of least significant byte
byte RegAddH;     // address of most significant byte
byte DevAdd;      // write address of I2C bus device

FlushSerialInput();
choice = 0;
Serial.println();
Serial.print("1=seize bus ");
Serial.print("2=release bus ");
Serial.println();
Serial.print("3=read C2 ");          //HarmonicAdj
Serial.print("4=write C2 ");
Serial.println();
Serial.print("5=read C3 ");          //HarmonicAdj
Serial.print("6=write C3 ");
Serial.println();
Serial.print("7=select register ");
Serial.print("8=read selected ");
Serial.println();
Serial.print("9=write selected ");
Serial.print("10=set volume level ");
Serial.println();

FlushSerialInput();
Serial.println("Select menu option and hit return");
Serial.print(">");
choice = SerialReadInt();

switch (choice)
{

case 1 :
    digitalWrite(8, HIGH);
    break;
//
case 2 :
    digitalWrite(8, LOW);
    break;
//
case 3 :
    C2r = I2CSignedRead16(0x90, 0x16, 0x17);
    Serial.print ("C2 = ");
    Serial.println (C2r);

    FlushSerialInput();
    Serial.println("Hit Enter for menu");
    Serial.print(">");
    SerialReadInt();
    break;
//
case 4:

```