# Nano as I2C Master



# // PROJECT  :NanoAsI2CMasterTMP36

// PURPOSE  :AVRs (Nano & ATtiny85) as I2C Slaves

// AUTHOR   :C. D'Arcy

// DATE     :Created: 2024 11 30. Updated:2025 03 01

// MCU      :Nano (328p) as Master. Nano (328p) and ATtiny85 as Slaves

// COURSE   :ICS3U-E

// STATUS   :Working!

// REFERENCE:https://forum.arduino.cc/t/arduino-as-i2c-slave/38507/4

// NOTES    :Prior to running this upload NanoAsI2CSlaveTMP36.ino to slave Nano (0x02)

//          :and ATTiny85AsI2CSlaveTMP36.ino to ATtiny85 as slave (0x03)

#include <Wire.h>

#define NANOSLAVEADDRESS  0x02  //arbitrary (but coordinated) address

#define TINY85ADDRESS     0x03  //arbitrary (but coordinated) address

#define PACE 2000               //arbitrary delay...

#define DEBUG true              //optional Serial Monitor support

//See: https://www.tutorialspoint.com/cprogramming/c\_unions.htm

union b2f {      //matches union in I2CSlaveTMP36.ino

  uint8\_t b[4];  //multiple variables share the same memory location

  float f;       //billed as better than typecasting

} data;

void setup() {

  Wire.begin();       //join i2c bus (address optional for master)

  Serial.begin(9600); //start serial for output

  while (!Serial);    //wait for it...

}

void loop() {

  Wire.requestFrom(NANOSLAVEADDRESS, 4); //request 4 bytes from slave

//  Wire.requestFrom(TINY85ADDRESS, 4); //same

  while (!Wire.available());          //wait for 4 bytes of TMP36 float

  data.b[3] = Wire.read();            //read the bytes

  data.b[2] = Wire.read();

  data.b[1] = Wire.read();

  data.b[0] = Wire.read();

  if (DEBUG)

    for (int8\_t i = 3; i >= 0; i--) {

      displayBinary(data.b[i]);

      Serial.print(' ');

    }

  Serial.println(data.f); //union to reassemble the byte into float

  delay(PACE); //pause to confirm

}

//utility function to present all 8 bits in a byte

void displayBinary(uint8\_t n) {

  for (int8\_t i = 7; i >= 0; i--)

    Serial.print(n & (1 << i) ? 1 : 0);

}



# Nano as I2C Slave

// PROJECT  :NanoAsI2CSlaveTMP36

// PURPOSE  : Slave Responds with celsius float reading as 4-byte stream

// COURSE   :ICS3U-E

// AUTHOR   :C. D'Arcy

// DATE     :Created: 2025 02 28

// MCU      :328P

// STATUS   :Working

// REFERENCE:https://forum.arduino.cc/t/arduino-as-i2c-slave/38507/4

// REFERENCE:http://darcy.rsgc.on.ca/ACES/Datasheets/TMP35\_36\_37.pdf

// NOTE     :Responds to a Request from NanoAsI2CMasterTMP36.ino

#include <Wire.h>

#define NANOADDRESS 0x02  //arbitrary address

#define TMP36VCC    A2

#define TMP36       A1

#define TMP36GND    A0

#define VREF        5.0

#define OFFSET      550   //calibrate for better accuracy

#define DEBUG       false

//See: https://www.tutorialspoint.com/cprogramming/c\_unions.htm

union b2f {

  uint8\_t b[4];

  float f;

} data;

void setup() {

  pinMode(TMP36VCC, OUTPUT);  //prepare non-I2C TMP36 analog sensor

  pinMode(TMP36GND, OUTPUT);

  digitalWrite(TMP36VCC, HIGH);

  digitalWrite(TMP36GND, LOW);

  Wire.begin(NANOADDRESS);       //Nano joins I2C Bus

  Wire.onRequest(requestEvent);  //register response handler

  if (DEBUG) {

    Serial.begin(9600);

    while (!Serial);

    requestEvent();

  }

}

// function that executes whenever a request is made

// from the master. This function is registered as an event, see setup()

void requestEvent() {

  uint16\_t reading = analogRead(TMP36);     //obtain TMP36 sensor reading

  float voltage = reading \* VREF / 1024;    //obtain voltage

  data.f = (voltage \* 1000 - OFFSET) / 10;  //convert voltage to celsius

  Wire.write(data.b[3]);                    //exploit the union mapping

  Wire.write(data.b[2]);                    //to transmit 4 bytes of float

  Wire.write(data.b[1]);                    //

  Wire.write(data.b[0]);                    //

  if (DEBUG)

    Serial.println(data.f);

}

void displayBinary(uint8\_t n) {

  for (int8\_t i = 7; i >= 0; i--)

    Serial.print(n & (1 << i) ? 1 : 0);

}

void loop() {}

# ATtiny85 as I2C Slave

// PROJECT  :ATtiny85AsI2CSlaveTMP36

// PURPOSE  :ATtiny85 Slave Responds with celsius float as a 4-byte stream

// COURSE   :\*

// AUTHOR   :C. D'Arcy

// DATE     :Created: 2025 03 01

// MCU      :328P

// STATUS   :Working!

// REFERENCE:https://forum.arduino.cc/t/arduino-as-i2c-slave/38507/4

// REFERENCE:http://darcy.rsgc.on.ca/ACES/Datasheets/TMP35\_36\_37.pdf

// NOTE     :Responds to a Request from NanoAsI2CMasterTMP36.ino

#include <TinyWireS.h>

#define TINYADDRESS 0x03  //arbirtrary address

#define TMP36       A3

#define VREF        5.0

#define OFFSET      550   //calibrate for better accuracy

#define DEBUG       false

//See: https://www.tutorialspoint.com/cprogramming/c\_unions.htm

union b2f {

  uint8\_t b[4];

  float f;

} data;

void setup() {

  TinyWireS.begin(TINYADDRESS);       //ATtiny85 joins I2C Bus

  TinyWireS.onRequest(requestEvent);  //register response handler

  if (DEBUG) {

    Serial.begin(9600);

    while (!Serial);

    requestEvent();

  }

}

// function that executes whenever a request is made

// from the master. This function is registered as an event, see setup()

void requestEvent() {

  uint16\_t reading = analogRead(TMP36);     //obtain TMP36 sensor reading

  float voltage = reading \* VREF / 1024;    //obtain voltage

  data.f = (voltage \* 1000 - OFFSET) / 10;  //convert voltage to celsius

  TinyWireS.send(data.b[3]);   //exploit the union mapping

  TinyWireS.send(data.b[2]);   //to transmit the 4 bytes of the float

  TinyWireS.send(data.b[1]);   //

  TinyWireS.send(data.b[0]);   //

  if (DEBUG)

    Serial.println(data.f);

}

void displayBinary(uint8\_t n) {

  for (int8\_t i = 7; i >= 0; i--)

    Serial.print(n & (1 << i) ? 1 : 0);

}

void loop() {}

