The playground is a publicly-editable wiki about Arduino.

Manuals and Curriculum
Installing Arduino on Linux
Board Setup and Configuration
Development Tools
Interfacing With Hardware
- Output
- Input
- User Interface
- Storage
- Communication
- Power supplies
- General

Interfacing with Software
User Code Library
- Snippets and Sketches
- Libraries
- Tutorials

Suggestions & Bugs
Electronics Technique
Sources for Electronic Parts
Related Hardware and Initiatives
Arduino People/Groups & Sites
Exhibition
Project Ideas
Languages

PARTICIPATE
- Suggestions
- Formatting guidelines
- All recent changes

I2C (master and slave) on the ATtiny85

Overview:

The ATtiny85 microprocessor is an 8 pin chip with 6 (max!) I/O ports. Using an I2C bus greatly expands the possibilities of what you can do with this chip.

The ATtiny85 (and it's cousins) does not have I2C (or SPI) "built in". Instead it has a Universal Serial Interface (USI) that can be used to facilitate I2C and SPI.

After a bit of searching, I found 2 sets of code that use the USI for I2C - one for I2C master and the other for I2C slave. I wrapped each in a separate class and had them follow a similar usage as the Wire lib.

Status & Usage:

I don't consider myself a "sharp guy" when it comes to coding, and this was my first experience with making a class, so go easy. As far as I'm concerned, these libraries are done. However it would be nice to see the master and slave libraries combined like the Wire library is. The slave lib is also missing the "onReceive" and "onRequest" functions - however it is still usable.

I've only tested these libraries with an ATtiny85. They may work on an ATtiny45 and others that use USI with easy mods to the include files.

Testing was done using the the ATtiny85 core files from "high-low tech" http://hlt.media.mit.edu/wiki/pmwiki.php?n=Main.ArduinoATtiny4585 and also the core files from http://code.google.com/p/arduino-tiny/ and uploaded with the ArduinoISP or the USBtinyISP.

Both the 1MHz and 8MHz clock speeds were tested.

By default the I2C master library (TinyWireM) is set to run at 1MHz. To run at 8MHz, #defines in USI_TWI_Master.h / .cpp must be changed. No

- PmWiki
- WikiSandBox
training
- Basic Editing
- Cookbook (addons)
- Documentation
index
- Italian Tutorial

edit SideBar

changes are necessary for the I2C slave library (TinyWireS).

Each library includes examples, and there is documentation in the headers of these, and more in the associated .h files.

Be sure to use pullups! (4.7K on 5V). I2C with these libs is not as forgiving as the Wire lib. (Maybe the internal pullups are not set.)

I2C Master:

The base code for this lib came from 'jkl' at CMU - see .h for link. I've tested send and receive on severial I2C devices. You can download the lib here: TinyWireM.zip

```
USAGE is modeled after the standard Wire library . . .
 Put in setup():
       TinyWireM.begin(){                                // initialize I2C lib
 To Send:
       TinyWireM.beginTransmission(uint8_t slaveAddr){  // setup slave's address (7 bit addres
       TinyWireM.send(uint8_t data){                    // buffer up bytes to send - can be ca
       someByte = TinyWireM.endTransmission(){          // actually send the bytes in the buff
                                                        // returns (optional) 0 = sucess or se
 To Receive:
       someByte = TinyWireM.requestFrom(uint8_t slaveAddr, uint8_t numBytes){      // reads 'n
                                                        // (usage optional) returns 0= success
       someByte = TinyWireM.receive(){                  // returns the next byte in the receiv
       someByte = TinyWireM.available(){                // returns the number of unread bytes
```

There are 3 example sketches in the "examples" folder.

I2C Slave:

The base code for this lib came from Don Blake at AVR freaks - see .h for link. I've tested receive and respond while connected to a "master send a receive" sketch on the Arduino. You can download it here: TinyWireS.zip

```
USAGE is modeled after the standard Wire library . . .
 Put in setup():
       TinyWireS.begin(I2C_SLAVE_ADDR);              // initialize I2C lib & setup slave's add

 To Receive:
   someByte = TinyWireS.available(){                // returns the number of bytes in the rece
   someByte = TinyWireS.receive(){                  // returns the next byte in the received b

 To Send:
       TinyWireS.send(uint8_t data){                // sends a requested byte to master
```

There's an example sketch in the "examples" folder.

Modified LiquidCrystal_I2C for the ATtiny:

After becoming quite tired of counting flashing LEDs on the ATtiny, I made a small mod to Mario H's LiquidCrystal_I2C library that will use TinyWireM instead of Wire when compiling for an ATtiny85. It's pretty cool to see a 2x16 LCD

display running from that tiny chip! I added an example "HelloWorld_Tiny" to his examples. You can download the modified LiquidCrystal_I2C library for the ATtiny here: LiquidCrystal_I2C_85.zip

Here is the same library based off the newer LiquidCrystal_I2C library for the v1+ IDE: LiquidCrystal_I2C_85V1.zip

This lib still supports the ATmega as it always had, so you can safely replace your LiquidCrystal_I2C lib with this one.

I hope you find these libraries fun and useful.

BroHogan

(Blog with I2C/85 project - http://brohogan.blogspot.com/search/label/ATtiny85)

Share |