Application Note: AN00156

How to use the I2C master library

Required tools and libraries

The code in this application note is known to work on version 14.1.1 of the xTIMEcomposer tools suite, it may work on other versions.

The application depends on the following libraries:

- lib_logging
- lib_i2c

Required hardware

Prerequisites

- This document assumes familiarity with the XMOS xCORE architecture, the Universal Serial Bus 2.0 Specification (and related specifications, the XMOS tool chain and the xC language. Documentation related to these aspects which are not specific to this application note are linked to in the references appendix.
- For descriptions of XMOS related terms found in this document please see the XMOS Glossary [#]_.

1 Overview

1.1 Introduction

1.2 Block diagram

2 How to use I2C master

APPENDIX A - Demo Hardware Setup

APPENDIX B - Launching the demo application

APPENDIX C - References

XMOS Tools User Guide

http://www.xmos.com/published/xtimecomposer-user-guide

XMOS xCORE Programming Guide

http://www.xmos.com/published/xmos-programming-guide

XMOS xCORE-USB Device Library:

http://www.xmos.com/published/xuddg

XMOS USB Device Design Guide:

http://www.xmos.com/published/xmos-usb-device-design-guide

USB HID Class Specification, USB.org:

http://www.usb.org/developers/devclass_docs/HID1_11.pdf

USB 2.0 Specification

http://www.usb.org/developers/docs/usb20_docs/usb_20_081114.zip

APPENDIX D - Full source code listing

D.1 Source code for endpoint0.xc

D.2 Source code for main.xc

```
// Copyright (c) 2015, XMOS Ltd, All rights reserved
#include <stdio.h>
#include <xs1.h>
#include "debug_print.h"
#include "i2c.h"

port p_scl = XS1_PORT_4C;
port p_sda = XS1_PORT_1G;

void demo(client interface i2c_master_if i2c)
{
  i2c_regop_res_t result;
  result = i2c.write_reg(0x45, 0x07, 0x12);
  if (result != I2C_REGOP_SUCCESS) {
    debug_printf("Write reg failed!\n");
  }
  result = i2c.write_reg(0x45, 0x08, 0x78);
  if (result != I2C_REGOP_SUCCESS) {
    debug_printf("Write reg failed!\n");
  }
  unsigned char data = i2c.read_reg(0x45, 0x07, result);
  if (result != I2C_REGOP_SUCCESS) {
    debug_printf("Read reg failed!\n");
  }
  debug_printf("Read data %x from addr 0x90,0x07 (should be 0x12)\n", data);
}

int main(void) {
  i2c_master_if i2c[1];
  par {
    i2c_master(i2c, 1, p_scl, p_sda, 100);
    demo(i2c[0]);
  }
  return 0;
}
```

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the “Information”) and is providing it to you “AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.