Laboratory log book Robotics

Joris Stork: 6185320 Lucas Swartsenburg: 6174388
Sander van Veen: 6167969

January 12, 2011

1 I^2C and the CSS Compiler

1.1 Objective

Our objective is to implement an I^2C bus using: a 3-state word generator, an I/O expander and a microcontroller (these are described below). The I^2C bus is a two-wire serial bus used for data transport between integrated circuits.

1.2 Method

Our implementation of the I^2C bus involves transporting 8 parallel bits of data - generated by a human operator via the interface of a word generator - over from the I/O expander using the 2 bit I^2C protocol to the microcontroller, and back again. We programmed the microcontroller in a C-like language and used the provided C API (functions `i2c_read` and `i2c_write`) to act as the master on the I^2C bus, with the I/O expander as slave (see the appendix for code). We connected the microcontroller's C4 and C3 pins with the I/O expander's SDA (serial data line) and SCL (serial clock line) pins respectively. The I/O expander's eight left-hand side pins were connected to the corresponding eight pins of the word generator. The program loaded onto the microcontroller initiates the transfer on the bus, first setting the SDA signal to "low", then indicating the transmission direction (from slave to master) and the address of the slave (corresponding to the left-hand pins of the I/O expander) from which to send the data. Once the microcontroller had received the data, the program initiated a second transfer in the opposite direction, sending the same data it had received back to the I/O expander for display on the I/O expander's right-hand LEDs.

1.2.1 Hardware

- PCF8574 I^2C I/O expander integrated circuit (IC). Each of this I/O expander's eight pins can be used as an input or output. In order to use a pin as input, a 1 has to be written to that pin's register.
- 16F876 microcontroller: this is an 8 bit microcontroller that we load with programs written in mplab ...
- 74LS244 word (8 bit) generator. Buttons toggle each of the eight pins representing the 8 bits of output.

1.2.2 Software

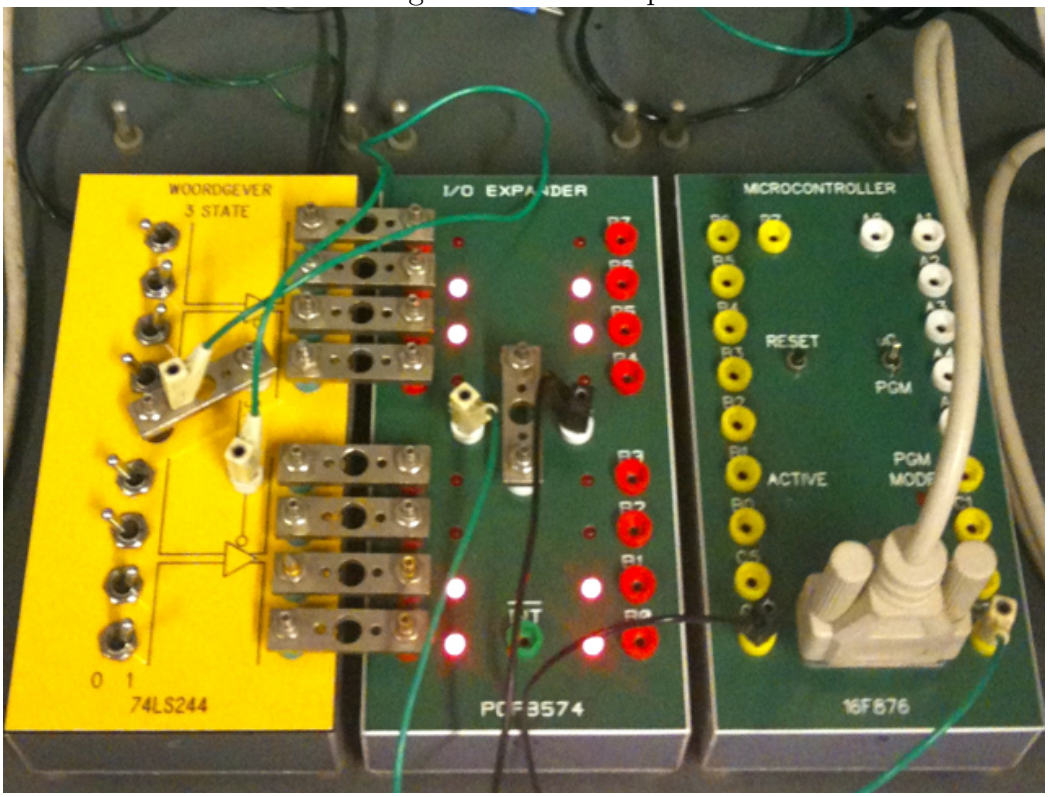
The follows software runs on an i86 Windows XP machine which is connected to the microcontroller using a serial port cable.

- Tera Term Web 3.1 : used as a terminal console for the microcontroller.
- MPLAB IDE v6.30
- PIDAC 876 programmer: Loads the compiled program onto the microcontroller using the serial port.

1.3 results

The experiment worked as expected. The `0b11000110` was transmitted and returned successfully, as shown in the image below:

Figure 1: The set-up



A source: i^{2}c

```
////////////////////////////////////////////////////////////////////////
//
// Filename  : Test876.c
// Revision  : 1.0
// Created   : 19-3-2001
// Revised :  26-11-2003 by Benb

// Project   : Pidac876
// Device : PIC16F876
// Development : MPLAB / CCS PCM
// Author : E. Steffens
// Department : Faculty of science
// Copyright : Universiteit van Amsterdam
//Description : Testing serial connection with PC
////////////////////////////////////////////////////////////////////////

#include <C:\Program Files\PICC\Devices\16F876.H>
#include <C:\Program Files\PICC\Drivers\CTYPE.H>

// Inform the compiler the clock frequency is 8 MHz
#use delay(clock=8000000)

// Setup the RS232 communication
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7,  bits=8)

//Setup the I2C bus
#use I2C (Master, SDA=PIN_C4, SCL=PIN_C3, SLOW)

int main(){
int Getal = 0;
    int Getal_1 = 0;
    printf("Assignment 4 \n\r");

//Voorbeeld: Byte van Master ? Slave met adres 1

while(1){
//Voorbeeld: Byte van Slave met adres 0 ? Master
i2c_start();
i2c_write( 0x41 );//write address (01000001)
```

```
Getal = i2c_read(0); // acknowledge
i2c_stop();

i2c_start();
i2c_write( 0x42 );//write address (01000010)
i2c_write( Getal ); //write byte
i2c_stop();
if(Getal!=Getal_1){
        printf("%d \n\r", Getal);
        Getal_1 = Getal;
    }
    }
  return 0;
}
```

B source: sar adc

C source: servo system