Control of Integrated Circuits through I2C-Bus using USB interface and GUI in Matlab

Guansong Ji

Supervised by
Prof. Dr.-Ing. Klaus Solbach

Institute of Microwave and RF-Technology
University of Duisburg-Essen
2014-10-23
Outline

1. Motivation
2. System Overview
3. I2C Bus Communication Protocol
4. Arduino Platform
5. PCB Design
6. Matlab GUI
7. Test of Matlab GUI and ICs
8. Conclusion
1. Motivation

• Power Amplifier Project for 7T MRI.

• ICs require digital data input for the control

• Functions of the ICs need to be tested on a demonstrator PCB.

• Matlab GUI needs to be designed and tested.
2. System Overview

Integrated Circuits
- PCA9543A (Switch)
- P82B96 (Bus Buffer)
- MAX5115 (DAC)
- PCA9554A (Register)
- PCF8591T (ADC)
3. I2C Bus Communication Protocol

I2C Bus

• Inter-Integrated Circuit (I2C, Trademark)
• Two-Wire Interface (TWI, simpler than SPI)
• Two Signal Wires: Serial Data (SDA) and Serial Clock (SCL)
• Serial Communication (slower, cheaper and easier)
• Byte-oriented Transfer (8 bits)
3. I2C Communication Protocol

An Example of communication protocol

Start  SCL Low  SDA Change  Repeated  Acknowledge  Stop
        start  bit
3. I2C Communication Protocol

An example of Write Operation (MAX5115)

<table>
<thead>
<tr>
<th>START</th>
<th>ADDRESS BYTE</th>
<th>COMMAND BYTE</th>
<th>DATA BYTE</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master SDA</td>
<td>S 0 1 0 A 3 A 2 A 1 A 0 0</td>
<td>C 7 C 6 C 5 C 4 C 3 C 2 C 1 C 0</td>
<td>D 7 D 6 D 5 D 4 D 3 D 2 D 1 D 0</td>
<td>P</td>
</tr>
<tr>
<td>Slave SDA</td>
<td>ACK</td>
<td>ACK</td>
<td>D7-D0</td>
<td>ACK</td>
</tr>
</tbody>
</table>

- 7-bit slave address followed by one Read/Write bit.
- High is Read, Low is Write.
- The Acknowledge bit is sent by the slave device.
- How are the command and data defined?
3. I2C Communication Protocol

An example of Write Operation (MAX5115)
3. I2C Communication Protocol
An example of Write Operation (MAX5115)

- For Write Operation, C7 and C6 should be 0.
- C5 and C4 are set to select NV or V.
- C3-C0 are set to select the respective DAC register.

<table>
<thead>
<tr>
<th>C7</th>
<th>C6</th>
<th>C5</th>
<th>C4</th>
<th>C3</th>
<th>C2</th>
<th>C1</th>
<th>C0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>

**FUNCTION**
- Transfer data from NVREG_ to VREG_
- Write to VREG_
- Write to NVREG_
- Write to NVREG and VREG_

<table>
<thead>
<tr>
<th>R3</th>
<th>R2</th>
<th>R1</th>
<th>R0</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>DAC0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>DAC1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>DAC2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>DAC3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>All DACs*</td>
</tr>
</tbody>
</table>
### 3. I2C Communication Protocol

An example of Write Operation (MAX5115)

<table>
<thead>
<tr>
<th>DAC CODE</th>
<th>OUTPUT VOLTAGE (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111 1111</td>
<td>$\frac{255 \times (V_{\text{REFH}} - V_{\text{REFL}})}{256} + V_{\text{REFL}}$</td>
</tr>
<tr>
<td>1000 0000</td>
<td>$\frac{128 \times (V_{\text{REFH}} - V_{\text{REFL}})}{256} + V_{\text{REFL}}$</td>
</tr>
<tr>
<td>0000 0001</td>
<td>$\frac{(V_{\text{REFH}} - V_{\text{REFL}})}{256} + V_{\text{REFL}}$</td>
</tr>
<tr>
<td>0000 0000</td>
<td>$V_{\text{REFL}}$</td>
</tr>
</tbody>
</table>

**Unipolar Code Output Voltage**
4. Arduino Platform

- Arduino Mega 2560
- Arduino IDE (Integrated Development Environment)
- Support Package (by MathWorks)
4. Arduino Platform

MATLAB Support Package for Arduino

- adiosrv.pde
  (Firmware uploaded to the Arduino Board)

- aduino.m
  (Matlab class definition file, object-oriented)

- They have to be modified for the I2C Bus
  (adiosrvDAC.pde and aduinoDAC.m)
4. Arduino Platform

Firmware uploaded to Arduino Board

• Wire Library

• Function setup()
  • Serial.begin(115200);

• Function Loop()
  • If statement (Input)
  • Switch statement
  • Initial state s is set to be -1
  • Cases (Process and Output)

• State Machine
4. Arduino Platform

Matlab class definition file

• Object-oriented Programming Approach
• Classdef arduinoDAC < handle
• Class Properties and Class Methods
• The class defines an Arduino Object
• Functions used in the I2C Communications
  • checkI2Caddress (a, addr);
  • i2cWriteCommand (a, address, command);
  • i2cWriteData (a, address, command, data);
  • i2cReadCommand (a, address, command);
  • i2cReadData (a, address)
5. PCB Design

Finished Board Design of Software Eagle
5. PCB Design

The assembly of PCB

- PCA9543A
- PCA9554A
- MAX5115
- PCF8591T
- P82B
6. Matlab GUI

MultiWindow GUI

global ComPort
contents = cellstr(get(hObject,'String'));
ComPort = contents{get(hObject,'Value')};

clear a; % Clear the variable a
global a ComPort;
clc; % Clear Command Window
a = arduinoDAC(ComPort); %connect Matlab with the board and create an arduino object
6. Matlab GUI
MultiWindow GUI

PCA9543A

I2C address (A1-A0):
Set I2C address
Channel 1
Channel 2

MAX5115

I2C address (A3-A2-A1-A0): 0000

Channel
0

Phase
151

Store
Read

Graph
6. Matlab GUI

MultiWindow GUI

PCA9554A

PCF8591T
6. Matlab GUI

16 cassettes GUI

Main Panel
6. Matlab GUI

16 cassetes GUI

Control Panel
7. Test of Matlab GUI and ICs
8. Conclusion

• PCB has been designed and soldered
• Two versions of Matlab GUI has been designed
• The Matlab GUI and ICs have been tested
• New version of Support Package for Arduino
• GUI could be more convenient
Thank you for your attention!