

**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**

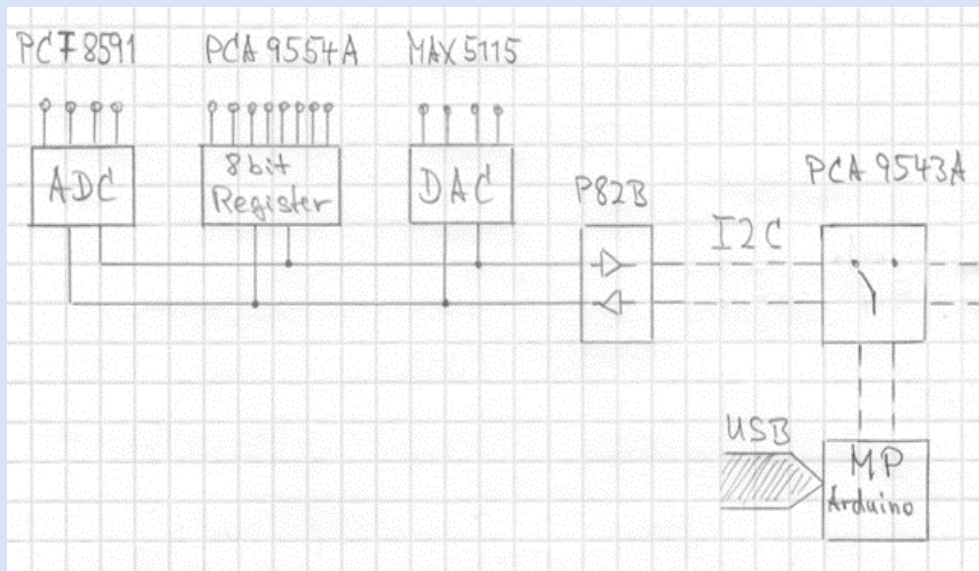
# Outline

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

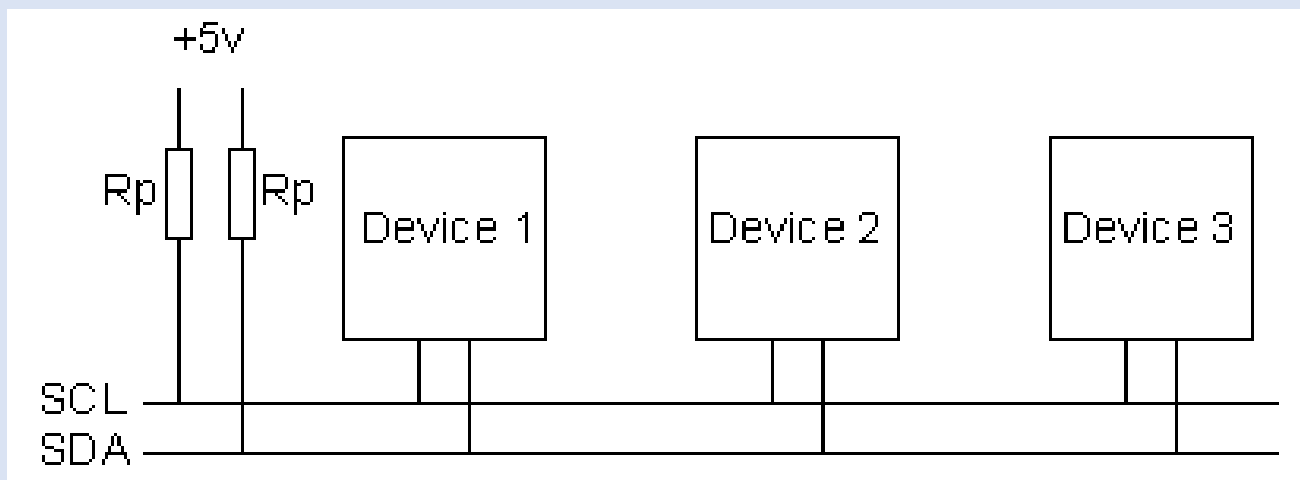
- PCA9543A (Switch)
- P82B96 (Bus Buffer)
- MAX5115 (DAC)
- PCA9554A (Register)
- PCF8591T (ADC)



# 3. I2C Bus Communication Protocol

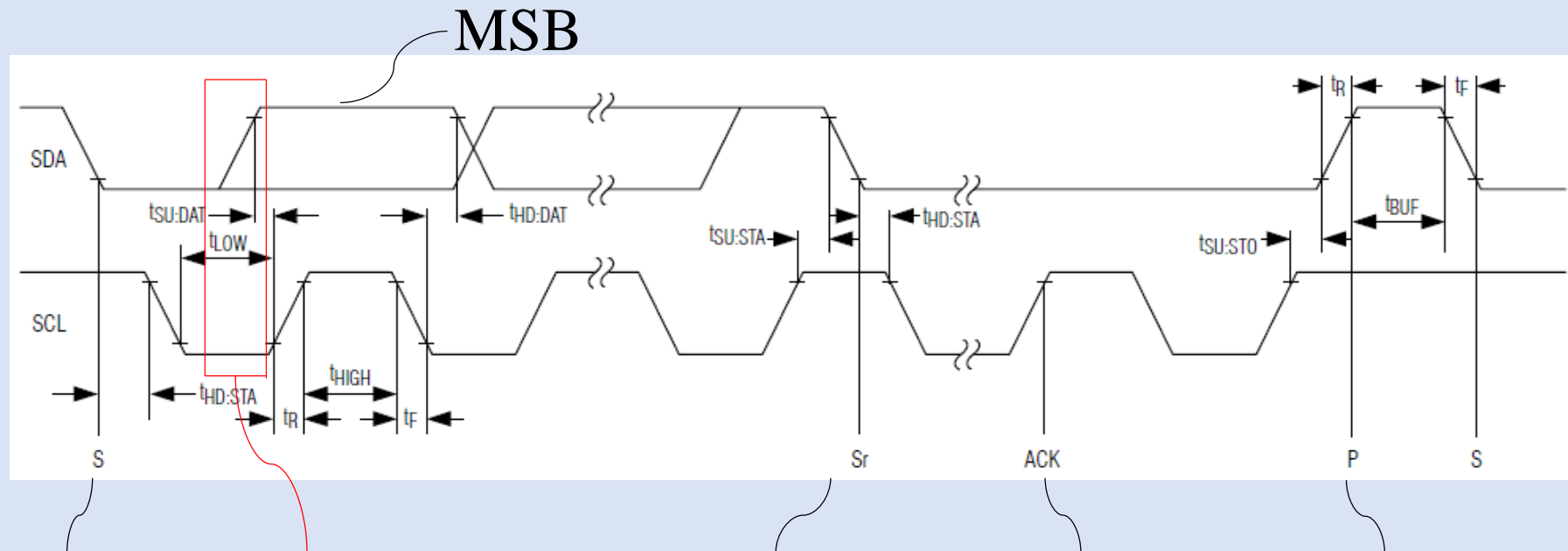
## I2C Bus

- 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  
start

Acknowledge  
bit

Stop

# 3. I2C Communication Protocol

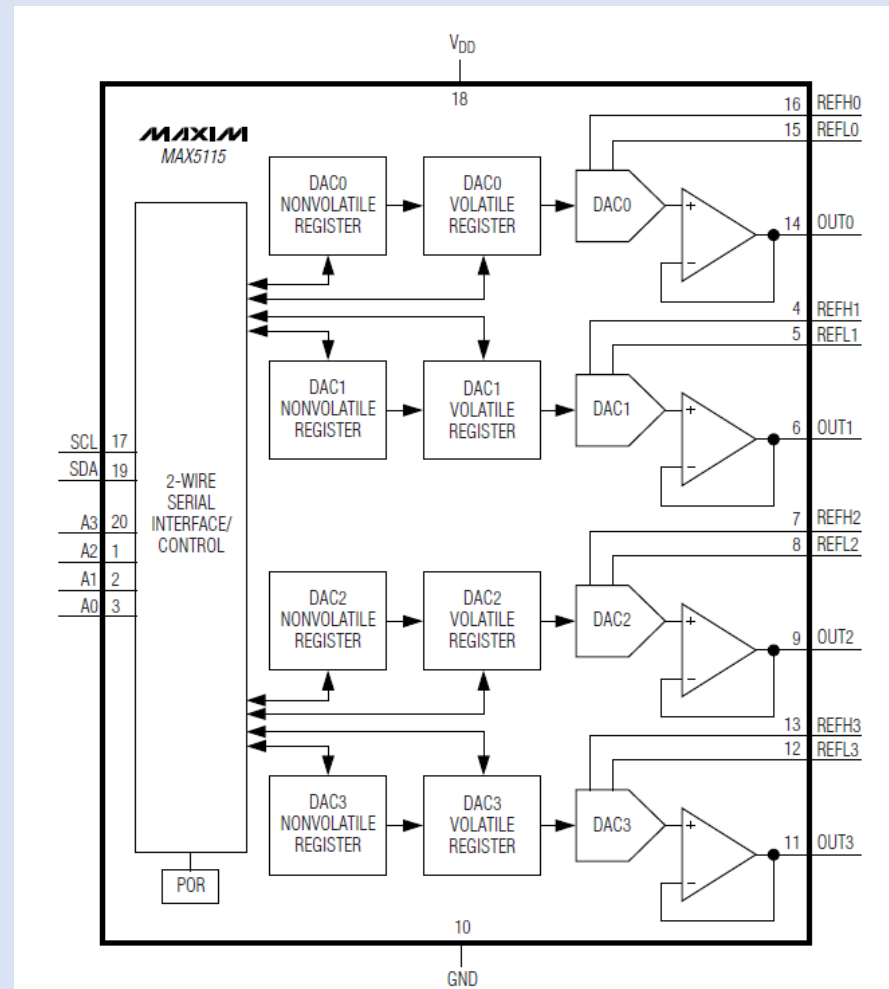
## An example of Write Operation (MAX5115)

	START	ADDRESS BYTE								R/W	COMMAND BYTE									DATA BYTE								STOP				
											C7	C6	C5	C4	C3	C2	C1	C0		D7	D6	D5	D4	D3	D2	D1	D0					
Master SDA	S	0	1	0	A3	A2	A1	A0	0	C7	C6	C5	C4	C3	C2	C1	C0	D7-D0								P						
Slave SDA										A	C	K						A	C	K									A	C	K	

- 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.

<b>C</b> <b>7</b>	<b>C</b> <b>6</b>	<b>C</b> <b>5</b>	<b>C</b> <b>4</b>	<b>C</b> <b>3</b>	<b>C</b> <b>2</b>	<b>C</b> <b>1</b>	<b>C</b> <b>0</b>
0	0	0	1	R 3	R 2	R 1	R 0

<b>NONVOLATILE (NV)</b>	<b>VOLATILE (V)</b>	<b>FUNCTION</b>
0	0	Transfer data from NVREG_ to VREG_
0	1	Write to VREG_
1	0	Write to NVREG_
1	1	Write to NVREG and VREG_

<b>R3</b>	<b>R2</b>	<b>R1</b>	<b>R0</b>	<b>FUNCTION</b>
0	0	0	0	DAC0
0	0	0	1	DAC1
0	0	1	0	DAC2
0	0	1	1	DAC3
1	1	1	1	All DACs*

# 3. I2C Communication Protocol

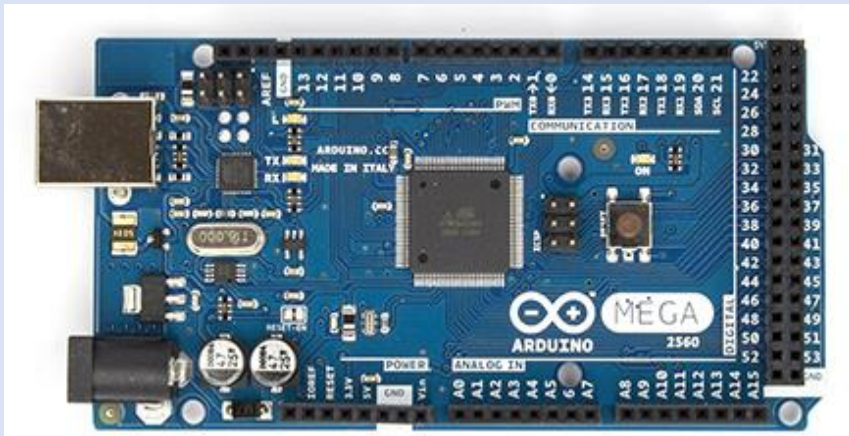
## An example of Write Operation (MAX5115)

DAC CODE	OUTPUT VOLTAGE (V)
1111 1111	$\frac{255 \times (V_{REFH\_} - V_{REFL\_})}{256} + V_{REFL\_}$
1000 0000	$\frac{128 \times (V_{REFH\_} - V_{REFL\_})}{256} + V_{REFL\_}$
0000 0001	$\frac{(V_{REFH\_} - V_{REFL\_})}{256} + V_{REFL\_}$
0000 0000	$V_{REFL\_}$

Unipolar Code Output Voltage

# 4. Arduino Platform

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



```
adiosrvDAC | Arduino 1.0.5-r2
Datei Bearbeiten Sketch Tools Hilfe
adiosrvDAC
/* Analog and Digital Input and Output Server for MATLAB */
/* Giampiero Campa, Copyright 2009 The MathWorks, Inc */

/* This file is meant to be used with the MATLAB arduino IO
package, however, it can be used from the IDE environment
(or any other serial terminal) by typing commands like:

0e0 : assigns digital pin #4 (e) as input
0f1 : assigns digital pin #5 (f) as output
0n1 : assigns digital pin #13 (n) as output

1c : reads digital pin #2 (c)
1e : reads digital pin #4 (e)
2n0 : sets digital pin #13 (n) low
2n1 : sets digital pin #13 (n) high
2f1 : sets digital pin #5 (f) high
2f0 : sets digital pin #5 (f) low
4j2 : sets digital pin #9 (j) to 50=ascii(2) over 255
...

622 Arduino Mega 2560 or Mega ADK on COM4
```

# 4. Arduino Platform

## MATLAB Support Package for Arduino

- `adiosrv.pde`

(Firmware uploaded to the Arduino Board)

- `aduno.m`

(Matlab class definition file, object-oriented)

- They have to be modified for the I2C Bus

(`adiosrvDAC.pde` and `adunoDAC.m`)

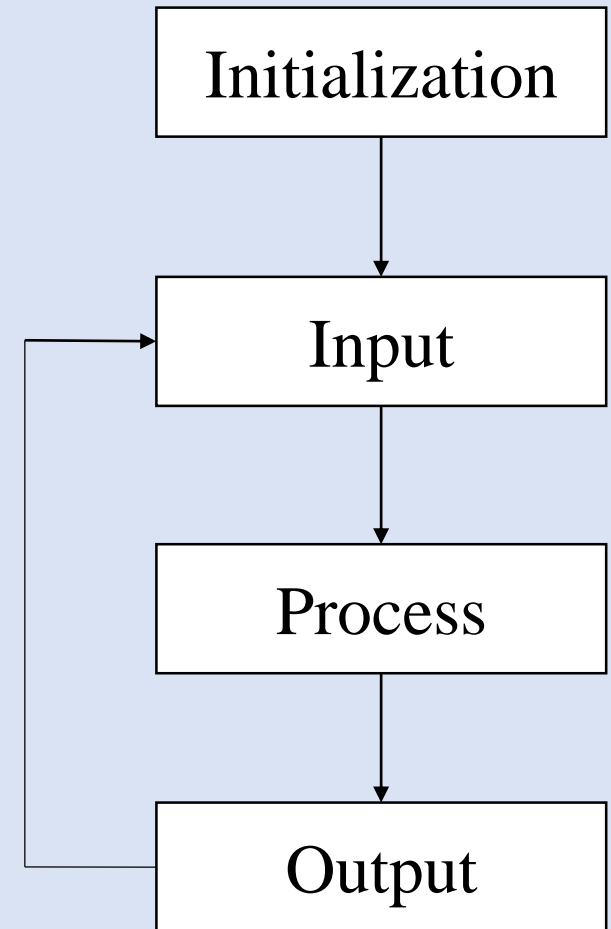
# 4. Arduino Platform

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

Initialization

Infinite Loop



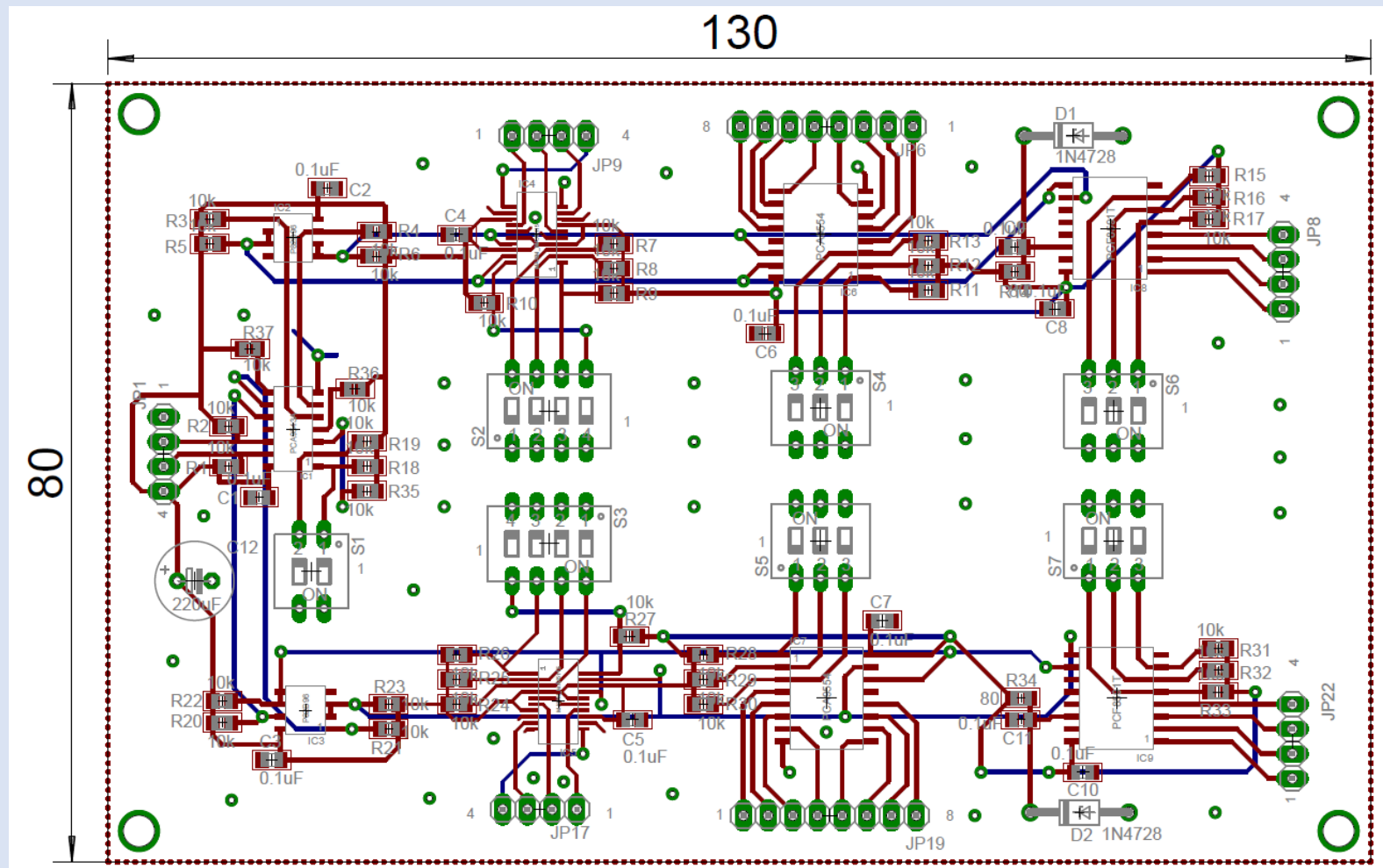
# 4. Arduino Platform

## 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)

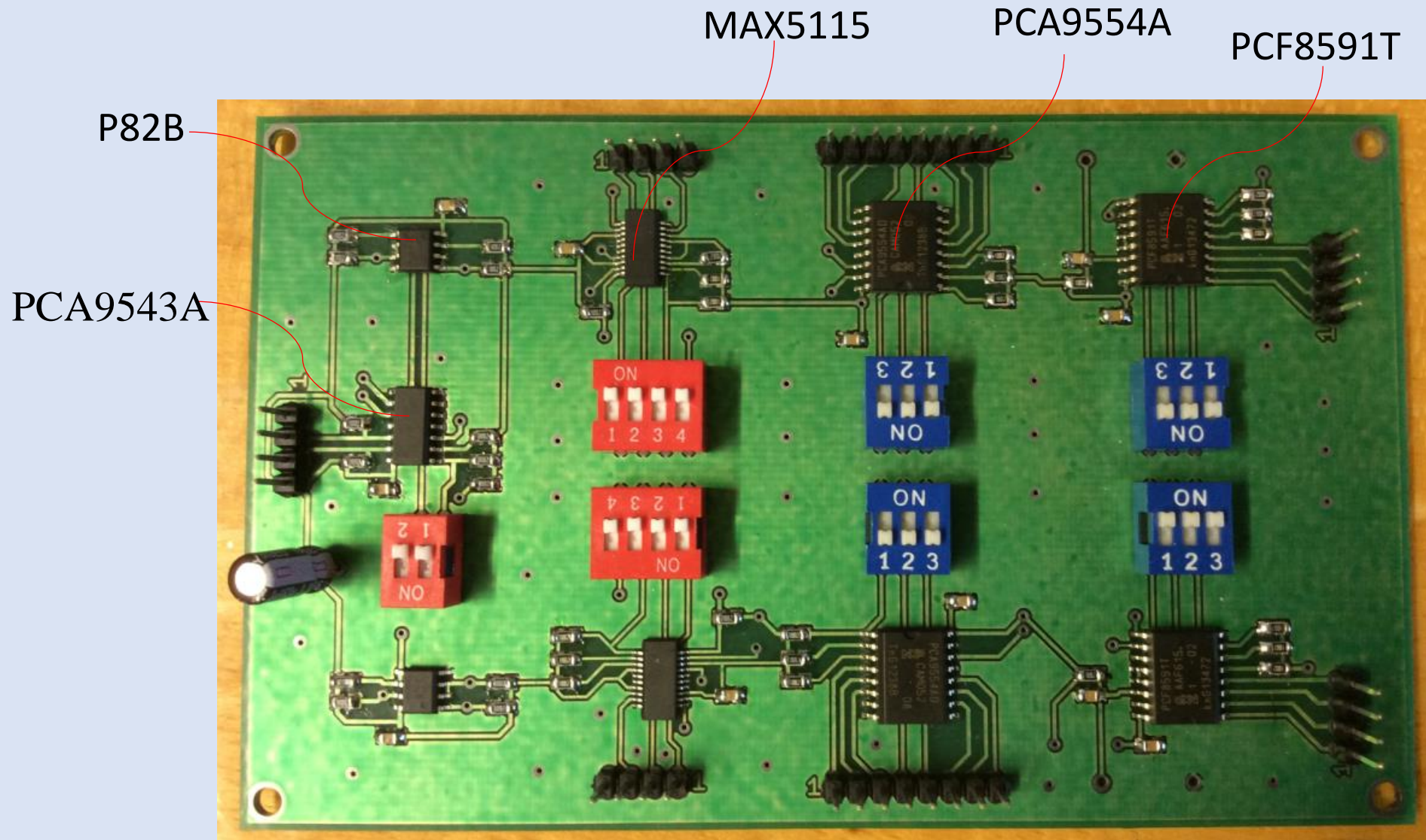
# 5. PCB Design

## Finished Board Design of Software Eagle



# 5. PCB Design

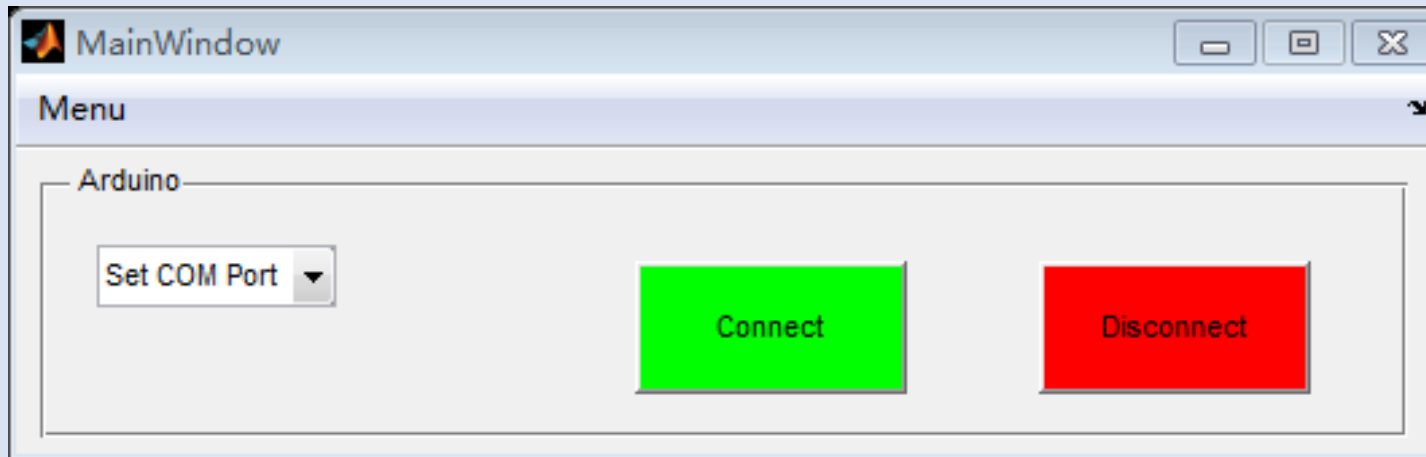
## The assembly of PCB





# 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;
```

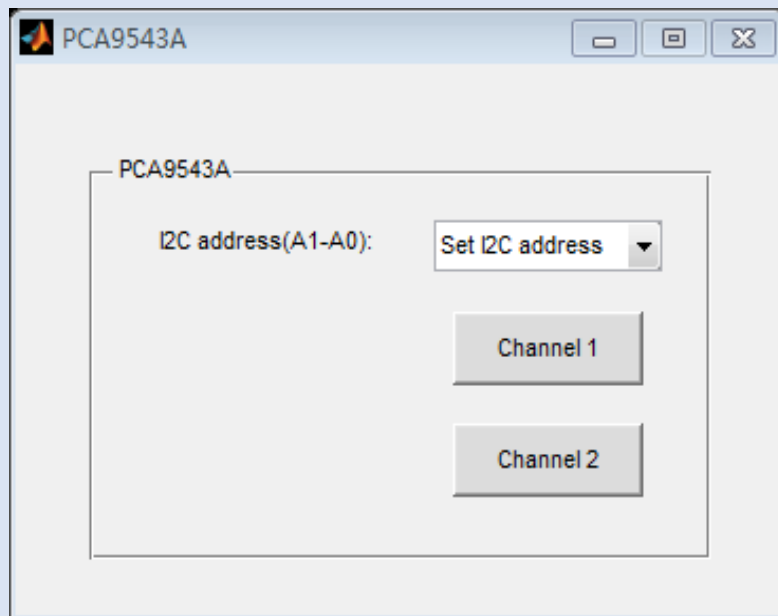
```
clc; % Clear Command Window
```

```
a = arduinoDAC(ComPort); %connect Matlab with the  
board and create an arduino object
```

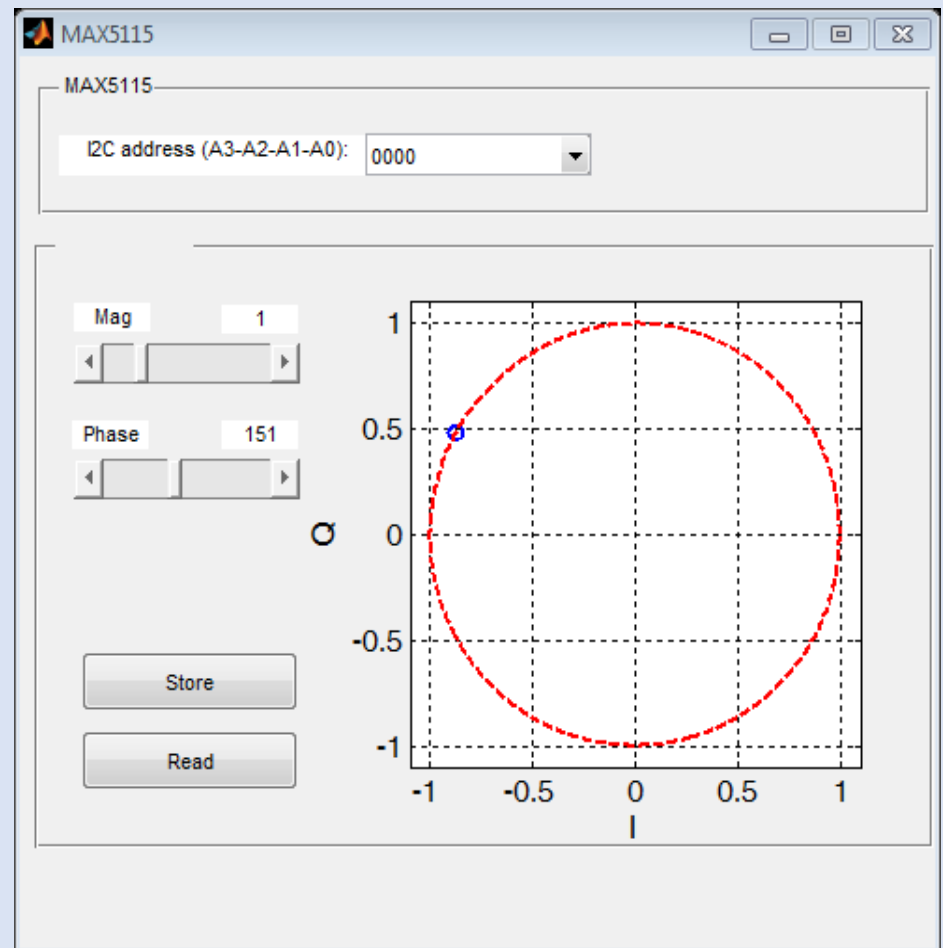
# 6. Matlab GUI

## MultiWindow GUI

### PCA9543A

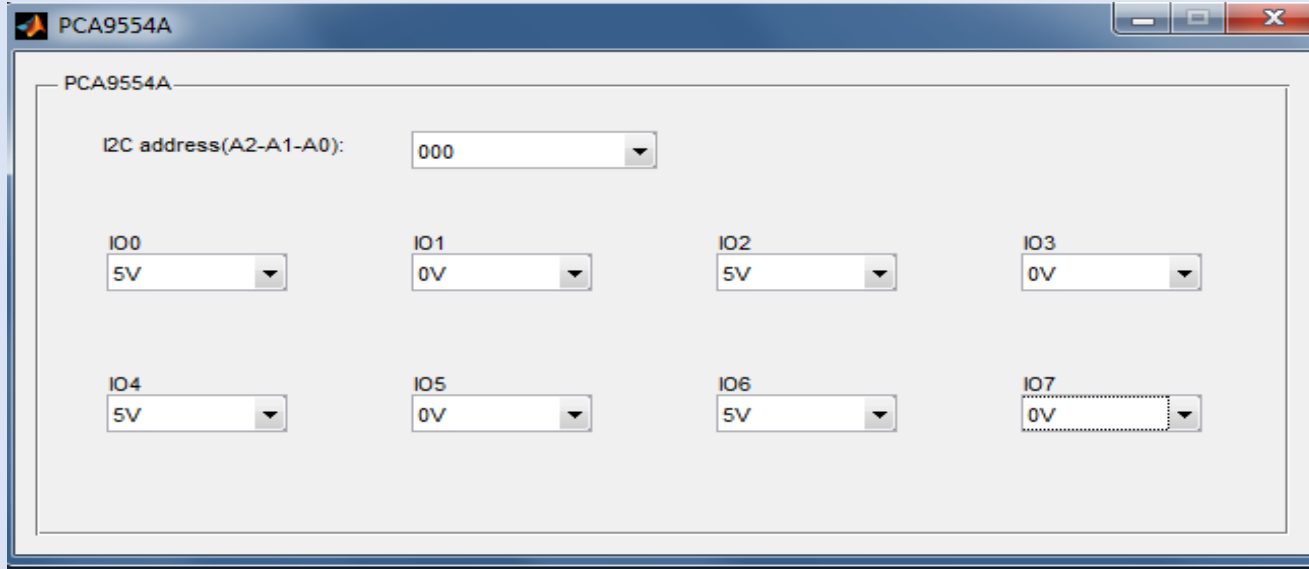


### MAX5115

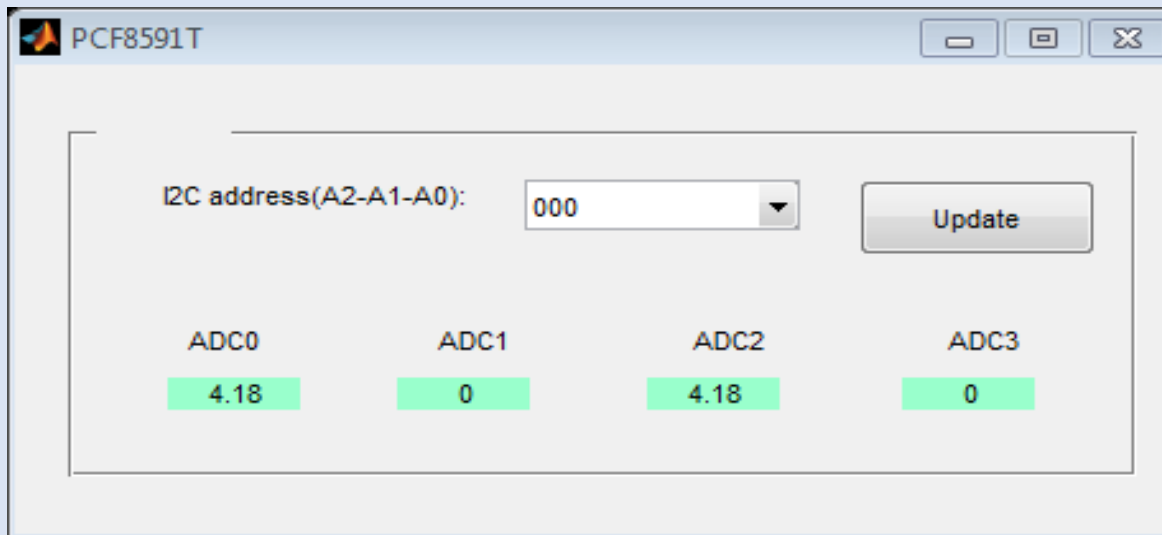


# 6. Matlab GUI

## MultiWindow GUI



PCA9554A

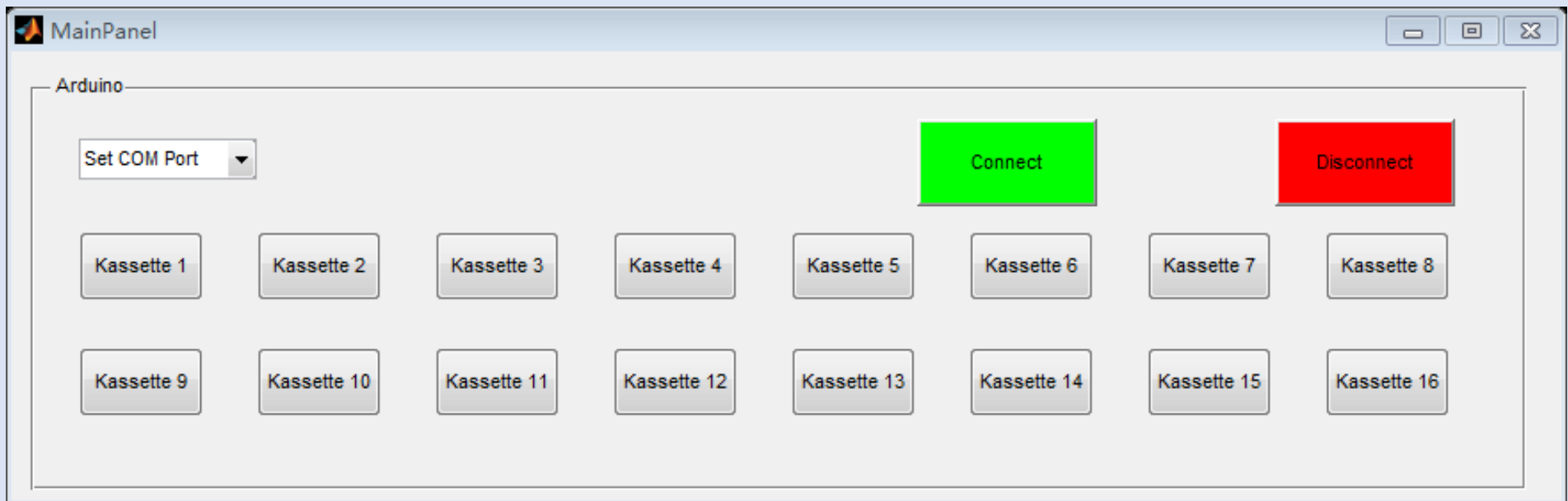


PCF8591T

# 6. Matlab GUI

## 16 cassettes GUI

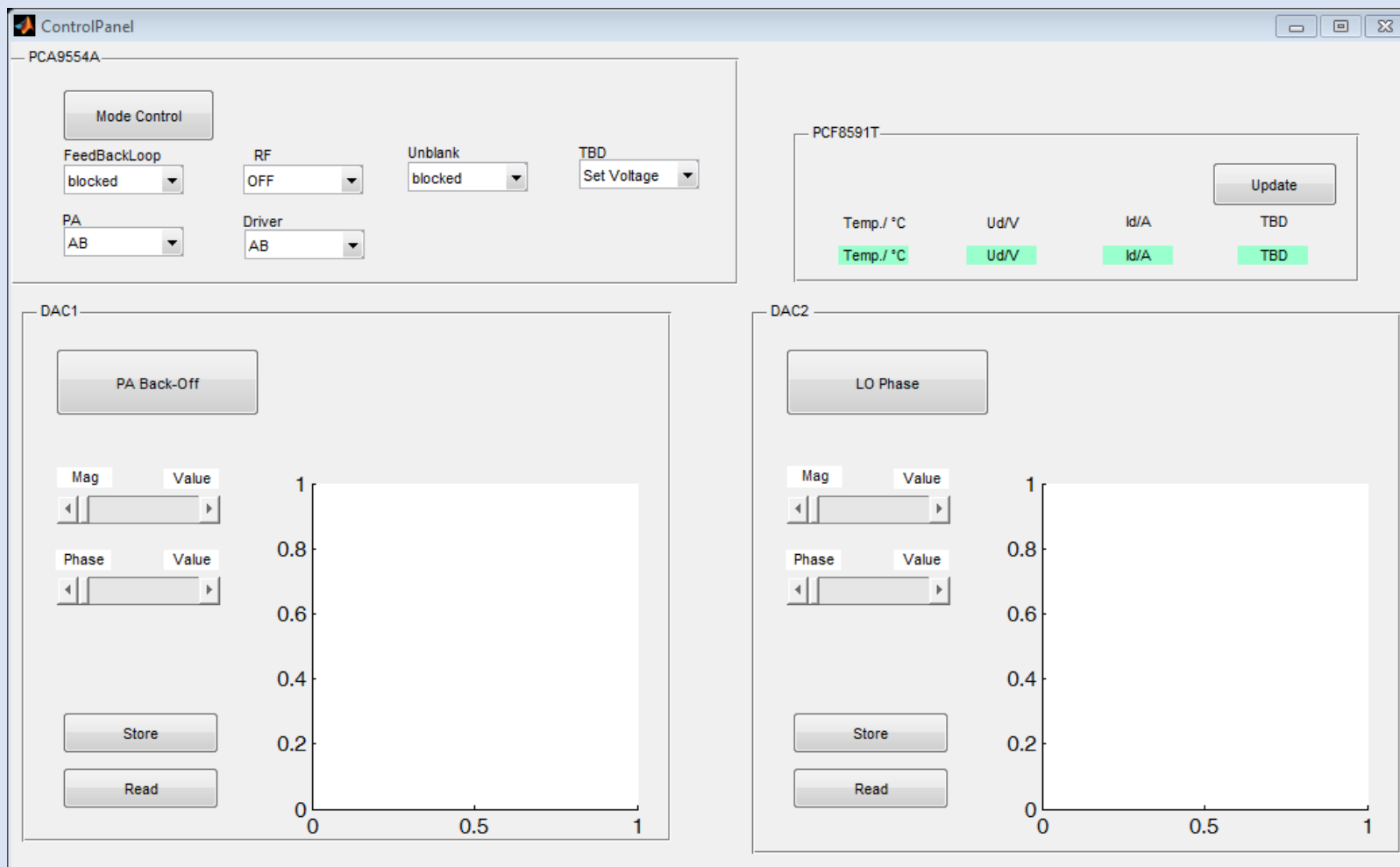
### Main Panel



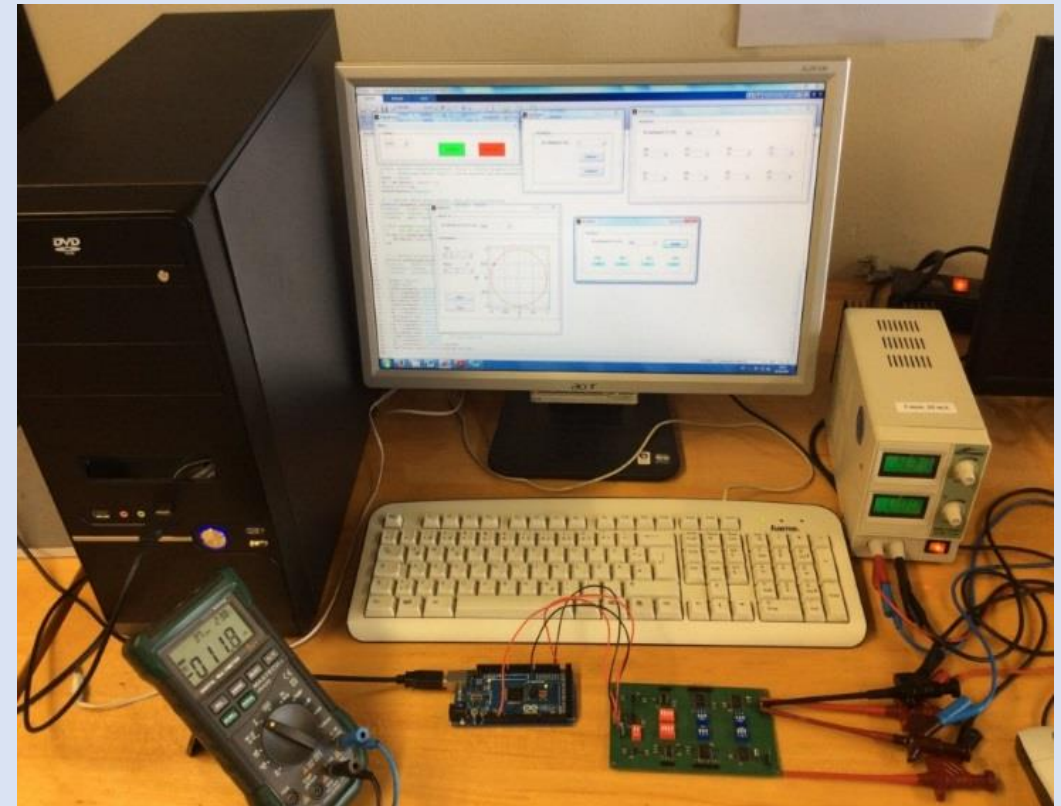
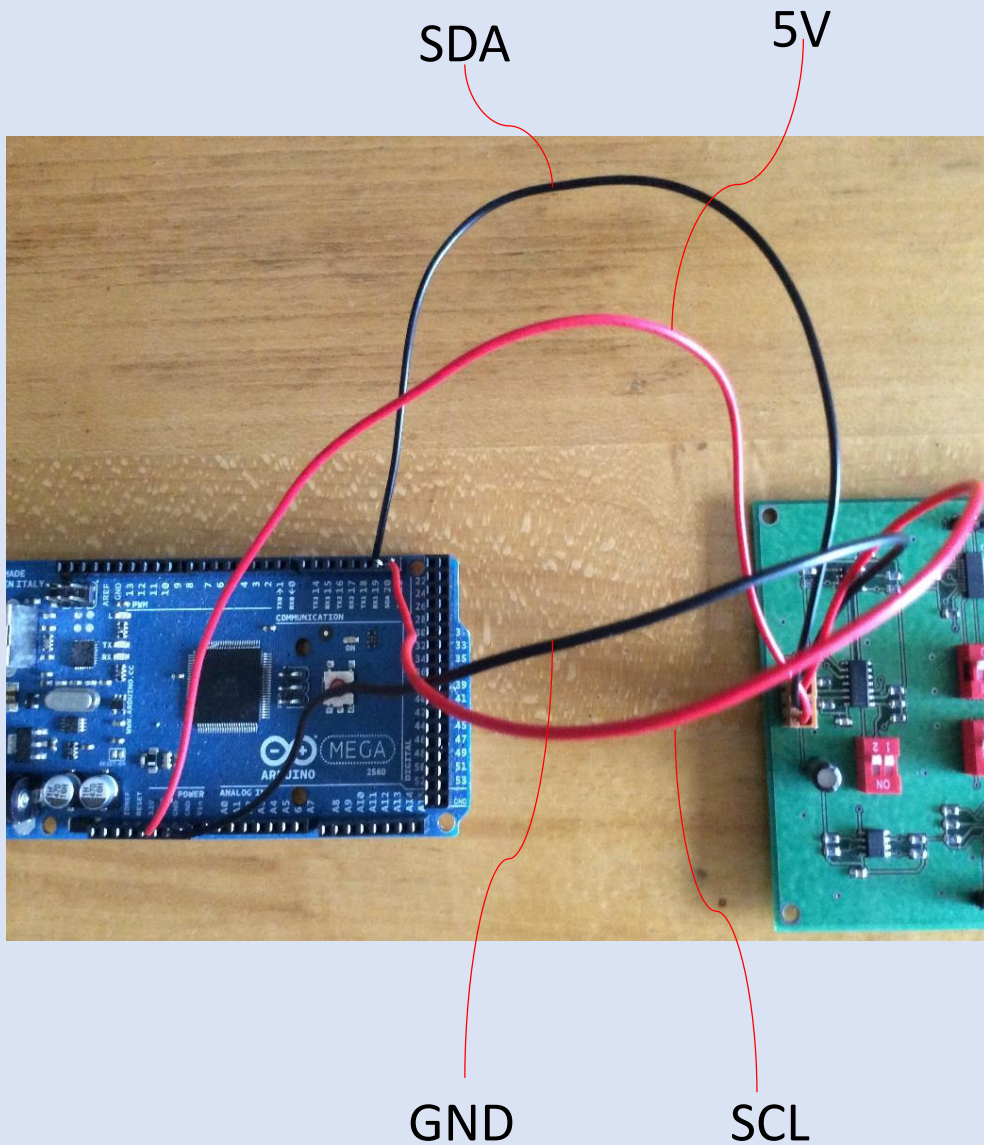
# 6. Matlab GUI

## 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!